Expanding Response: Deeper Analysis for Incident Handlers

Russ McRee

Expanding Response: Deeper Analysis for Incident Handlers

GCIH Gold Certification

Author: Russ McRee, holisticinfosec@gmail.com

Adviser: John Bambenek

Accepted: September 30, 2008

Table of Contents

1 Introduction

Most incident handlers likely have a toolkit they're fond of that, in all probability, contains tools most handlers are familiar with. It is the intent of this paper to expand common horizons and discuss tools that may not readily appear in a typical toolkit.

To create commonality amongst results, this paper will utilize a single malicious binary and two packet captures.

The binary utilized is named fireworks.exe, MD5: d7d350e34809adc4a55e592b58f9d4ad, also known as W32/Nuwar aka Storm.

Two packet captures will be utilized and referred to as fireworks.pcap and camda.pcap. Fireworks.pcap is 340 frames starting at 5:37:34 pm and ending at 5:39:34 pm on July 3rd, 2008. In this two minute capture the infected host connected to 142 unique hosts. Subtracting good traffic to NTP servers, gateway and broadcast, multicast, and local NetBIOS, 136 of these hosts were bot peers. Camda.pcap is 979 frames starting at 2:51:42 and ending at 2:54:42. This three minute capture identifies two malicious hosts, including an IRC server and malicious web server.

Further, all analysis for Expanding Response was conducted on hosts running Ubuntu Linux and VMWare Server 1.7 or Windows XP/Vista. Installation discussions will conform to those standards.

The perspective embraced for this discussion is that of an analyst who is working a process to determine the exact nature of malicious software on his network. He is in receipt of the above mentioned .exe and .pcap files and seeks to further his understanding with the use of less typical tools. She begins the process with the network capture, and then takes a closer look at the binary to see what can be learned and what the impacts of an outbreak on her network might be.

2 Network & Packet Analysis

2.1 Argus-3.0.0

Argus – Auditing Network Activity http://qosient.com/argus/index.htm

Prerequisites

```
*nix OS with bison, flex, and libpcap
Argus-clients-3.0 to utilize racluster
Graphviz to work with neato for AfterGlow output
RRDtool (RRDs.pm) for ragraph
```

Introduction

Argus is the network Audit Record Generation and Utilization System, a Real Time Flow Monitor that is designed to perform comprehensive data network traffic auditing. The brainchild of Carter Bullard of QoSient, "The Argus Open Project is focused on developing network activity audit strategies that can do real work for the network architect, administrator and network user." As a longtime advocate for good network security monitoring (NSM) tactics, I was first exposed to Argus via Bejtlich's *The Tao of Network Security Monitoring*. Consider this essential reading, if you haven't already read it. Where Argus shines for yours truly is, of course, security assurance. Bejtlich considers Argus "the single most important tool in the emergency NSM arsenal." Forced to choose one application in an incident response scenario, it would be Argus.[i]

When reading up on Argus, refer directly to its website at http://qosient.com/argus/index.htm. Web searches for Argus will also yield another project unfortunately of the same name that has nothing to do with this fine NSM offering.

Project Details

First iterations of Argus are traced back to 1990 when Bullard put it to use for investigative purposes while a grad student at Georgia Tech. A few years later Cisco's NetFlow debuted, but with one notable difference. Where NetFlow is unidirectional, Argus is a bi-directional flow modeler, matching network responses to any network traffic that is sent. In adhering to the IETF's Framework for IP Performance Metrics, Argus matches multiple identifier/descriptors in various flow models in Layers 2 through 5 of the OSI. Updated regularly since its inception, and with a strong, supportive community, Argus is a 3.0 pre-release state. Although version 2.06 is stable and readily available, I utilize 3.0 pre-release in order to take advantage of new client features described in the **Clients** section. This release, slated for availability by year's end, will offer IPv6 capabilities, arbitrary encapsulation parsing (VLAN, MPLS, GRE, IPnIP) and can be quickly adapted to new protocols, sometimes providing basic metrics without extension.

According to the NSMWiki, you'll find universities using Argus to record both internal traffic flows, as well as flows outside the DMZ to detect infected or compromised machines, in real time. Argus is in use at US government facilities to provide more extensive network forensics, and is the focus of research in control-plane network non-repudiation. The new release provides the parts to build a distributed network activity audit system for the complete enterprise; Argus as a data source installed in the end-system and as a network-based flow monitor, and the client program radium, which is a flow data collection and distribution system. Argus client programs can also read and convert NetFlow data, so you don't have to ignore that as a flow data source. Finally, network research labs have used Argus to test network performance of unique protocols, such as Infiniband over IPv6. [ii]

Bullard describes Argus on his website as well suited to security assurance as it "enables the establishment of a comprehensive audit trail of all network activity, either for a single network element or for an entire network segment." Opportunities abound, including:

- Non-Repudiation
- Incident Response
- Policy Enforcement Validation
- protocol validation
- network behavioral base lining
- intrusion detection
- discovery detection

- network asset inventory

If we agree that the definition of incident response is "the practice of detecting a problem, determining its cause, minimizing the damage it causes, resolving the problem, and documenting each step of the response for future reference"[iii], then incorporating Argus is the embodiment of this endeavor.

Installation

Argus is simple to install on an Ubuntu host.

sudo apt-get install argus-server argus-client

You can also grab source and *./configure*, *make,* and *make install*, assuming your dependencies are met (bison, flex, libpcap), and has been ported to most platforms, including Cygwin, and OpenWrt.

Server use

My use of Argus is often capturing via a SPAN port, but any "network tapping strategy that captures all the packets destined to and from the target(s)"[iv] will suffice. Keep in mind though that a SPAN port has limitations and that you are well advised to utilize a well placed tap. Lots of conventional wisdom is available to you if you search *span versus tap*.

Keep in mind, Argus can be deployed directly on a server of interest to measure performance itself.

I'll show you a few specific use scenarios, but I'd also refer you to an excellent resource that provides precise details on Argus use. *Structured Traffic Analysis*, by Richard Bejtlich, can be found in (IN)SECURE Magazine Issue 4. Grab it here: www.net-security.org/dl/insecure/INSECURE-Mag-4.pdf. Richard covers an entire process for network incident response (NIR) that includes other useful tools in addition to Argus, but provides an ideal play by play of Argus use as well.

There is a configuration file (most often /etc/argus.conf) wherein you can daemonize it, set a PID, manage instances, and set the listening port, IP, and interface. All settings have command line equivalents, conveniently explained in the configuration sample, as well as the man pages, and *argus -h* will always come through for you.

If I'm running the server, I typically set the Argus daemon to run simply. I don't use it to measure performance, so I don't pass parameters like -R for response time data, but remember Argus excels in this capacity. I just want it to listen on eth1 and write to an out file; thus I pass *argus -I eth1 -w capture.out* at the prompt. Remember, Argus is also excellent for static analysis as detailed below, but first a not on security considerations.

Security Considerations

If you set Argus up for remote connectivity, you have some security considerations to attend to. Access control can be managed by tcp_wrappers where you can specify what hosts can access Argus, or you can incorporate the Simple Authentication and Security Layer (SASL).

SASL provides strong authentication and confidentiality protection for Argus data on the wire, deemed "very important stuff when accessing remote real-time Argus data."[v]

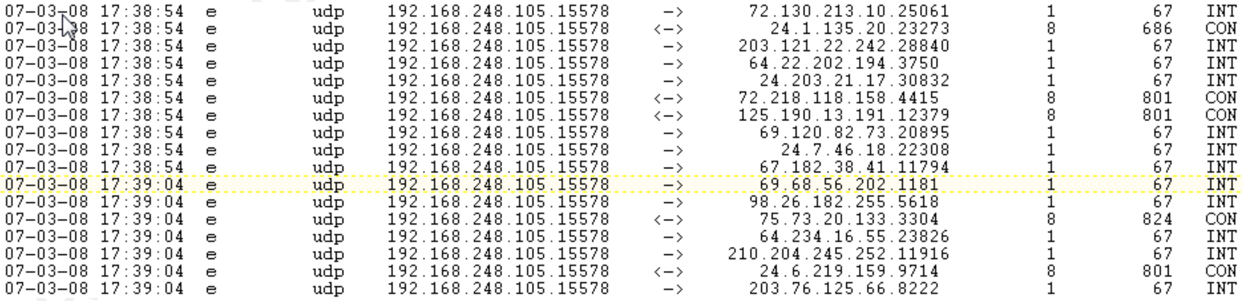
Client Use

Utilizing fireworks.pcap, a capture taken during analysis of Storm analysis I conducted on the 4th of July, 2008, I have a number of client options to run the data through. With the client package installed as described earlier, you should have tools like ra, ragrep, racount, rasort, and ragraph at your disposal.

Before conducting any analysis via the client tools I converted camda.pcap for use via Argus as follows:

argus -mAJZRU 512 -r fireworks.pcap -w camda.arg3

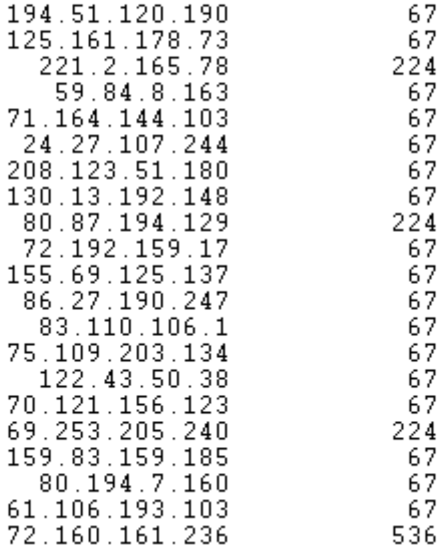
First, I ran *ra -r fireworks.arg3 – udp* to see what kind of UDP was being generated and spotted a plethora of high port UDP to multiple hosts; very typical of Storm.



```
07-03-08 17:38:54  e     udp   192.168.248.105.15578   ->     72.130.213.10.25061      1      67  INT
07-03-08 17:38:54  e     udp   192.168.248.105.15578  <->      24.1.135.20.23273       8     686  CON
07-03-08 17:38:54  e     udp   192.168.248.105.15578   ->    203.121.22.242.28840      1      67  INT
07-03-08 17:38:54  e     udp   192.168.248.105.15578   ->     64.22.202.194.3750       1      67  INT
07-03-08 17:38:54  e     udp   192.168.248.105.15578   ->      24.203.21.17.30832      1      67  INT
07-03-08 17:38:54  e     udp   192.168.248.105.15578  <->    72.218.118.158.4415       8     801  CON
07-03-08 17:38:54  e     udp   192.168.248.105.15578  <->    125.190.13.191.12379      8     801  CON
07-03-08 17:38:54  e     udp   192.168.248.105.15578   ->      69.120.82.73.20895      1      67  INT
07-03-08 17:38:54  e     udp   192.168.248.105.15578   ->        24.7.46.18.22308      1      67  INT
07-03-08 17:38:54  e     udp   192.168.248.105.15578   ->      67.182.38.41.11794      1      67  INT
07-03-08 17:39:04  e     udp   192.168.248.105.15578   ->      69.68.56.202.1181       1      67  INT
07-03-08 17:39:04  e     udp   192.168.248.105.15578   ->     98.26.182.255.5618       1      67  INT
07-03-08 17:39:04  e     udp   192.168.248.105.15578  <->      75.73.20.133.3304       8     824  CON
07-03-08 17:39:04  e     udp   192.168.248.105.15578   ->      64.234.16.55.23826      1      67  INT
07-03-08 17:39:04  e     udp   192.168.248.105.15578   ->   210.204.245.252.11916      1      67  INT
07-03-08 17:39:04  e     udp   192.168.248.105.15578  <->     24.6.219.159.9714        8     801  CON
07-03-08 17:39:04  e     udp   192.168.248.105.15578   ->     203.76.125.66.8222       1      67  INT
```

Figure 1

We see rapid connections to multiple hosts, with occasional conversations thrown in for good measure. Drilling in further for just destination addresses and src to dst byte quantities we can the hosts with whom more extensive communication occurred; command and control message perhaps. Executing ra -r fireworks.arg3 - udp -s daddr sbytes gives us a clearer picture.



```
194.51.120.190          67
125.161.178.73          67
  221.2.165.78         224
   59.84.8.163          67
71.164.144.103          67
 24.27.107.244          67
208.123.51.180          67
130.13.192.148          67
 80.87.194.129         224
 72.192.159.17          67
155.69.125.137          67
 86.27.190.247          67
  83.110.106.1          67
75.109.203.134          67
  122.43.50.38          67
70.121.156.123          67
69.253.205.240         224
159.83.159.185          67
  80.194.7.160          67
61.106.193.103          67
72.160.161.236         536
```

Figure 2: ra - daddr & sbytes output

If you make use of RRDtool, ragraph will offer you a graphical representation of Argus output as well. Issuing *ragraph dbytes daddr -M 1s -fill -stack -r fireworks.arg3 - udp and dst bytes gt 67* will produce the following.

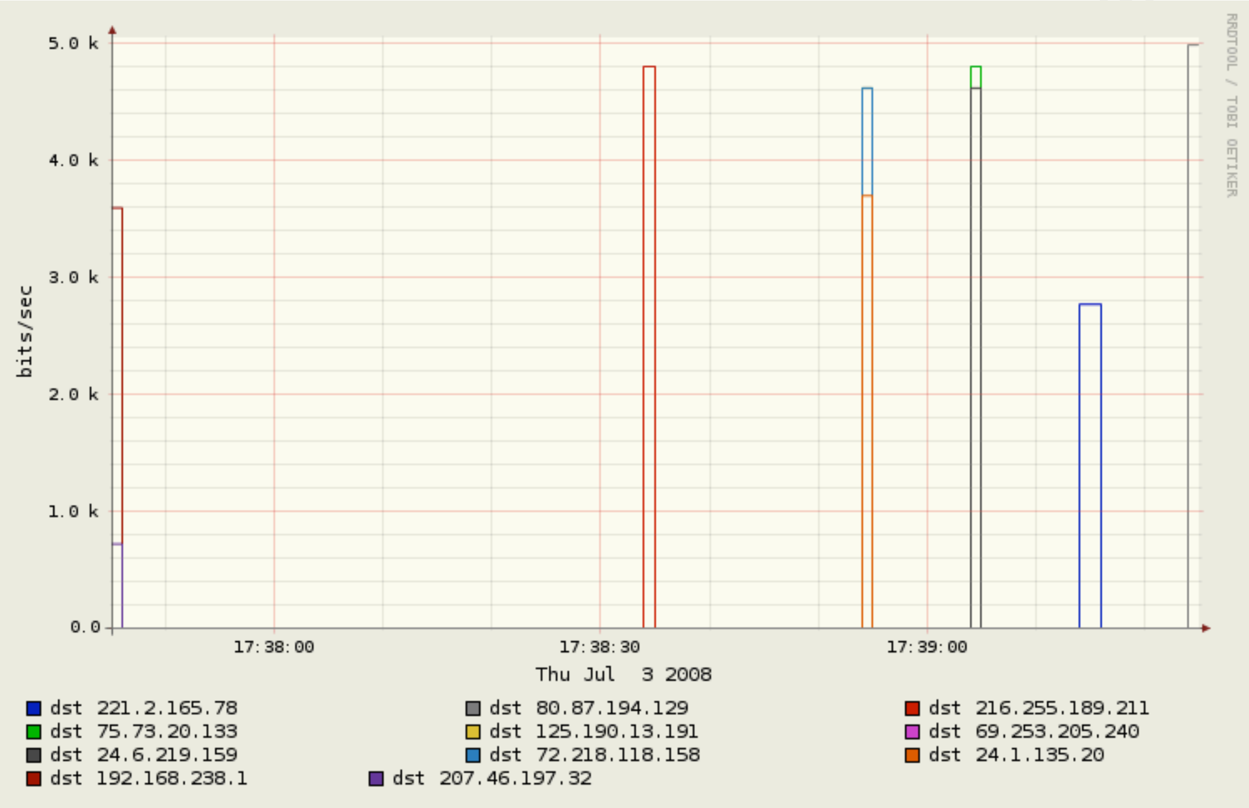


Figure 3: ragraph - dbytes daddr

I've asked only for results where destination bytes exceed 67; the graphic shows only destination addresses with larger byte counts, as seen in Figure 2. Visualizing Argus results in this fashion, utilizing the plethora of parameter available, may assist the analyst with data discovery that may be elusive when reviewing raw output.

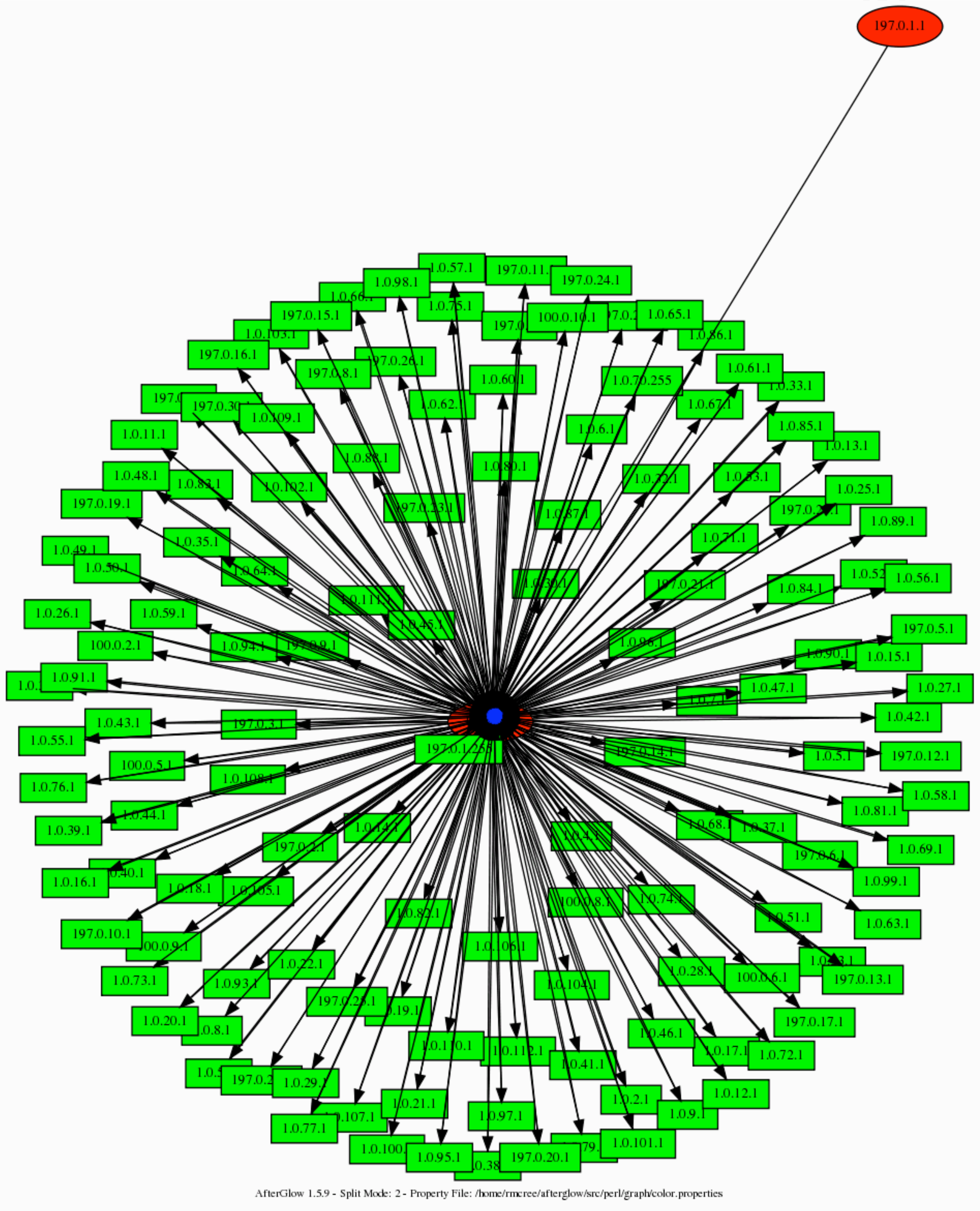
For an interesting read on tracking specific traffic, like bot-infected hosts, check out http://www.rawpacket.org/anonymous/papers/Argus-PracticalBotNetDetection.pdf.

Additional visualization opportunities

Taking the visual focus to the next level, there are certain visualization projects that can work with Argus data as well. If security visualization is of interest to you, check out secviz.org. be sure to read Greg Conti's *Security Data Visualization* and Raffael Marty's *Applied Security Visualization*.

Argus output can also be rendered by certain visualization projects, in particular AfterGlow *http://afterglow.sourceforge.net.* Argus 3.0 includes a number of new client features that aid in the visualization process, including direct output to CSV (needed for AfterGlow), ranonymize, which will change your IPs in the Argus output to create privacy, and racluster, an aggregator, both of which can be utilized for AfterGlow visualizations. The fireworks.pcap, randomized for privacy, can be rendered as follows, resulting in Figure 4:

```
ranonymize -r fireworks.arg3 -w - | racluster -r - -m saddr daddr ttl -c, -s
saddr daddr - 'udp' | /home/rmcree/afterglow/src/perl/graph/afterglow.pl -c
/home/rmcree/afterglow/src/perl/graph/color.properties -p 2 | neato -Tgif -o
fireworks.gif
```



AfterGlow 1.5.9 - Split Mode: 2 - Property File: /home/rmcree/afterglow/src/perl/graph/color.properties

Figure 4: ranonymize, racluster, AfterGlow output

SecViz.org is a great resource with content regarding combining use of Argus, AfterGlow, and Neato.[vi]

Benefits and Drawbacks

The only time I could imagine a drawback when using Argus might be in a scenario where a heretofore unmonitored user LAN is graced with a first look from an NSM implementation including Argus. The resulting horror of seeing what previously unmonitored users are up to would count as a drawback, given the probable heart attack for the analyst.

Conversely, the same information would be considered highly beneficial as it would aid the enterprise in question in the process of improving its security posture. You cannot fix what you cannot see. Risk reduced is confidence gained.

Conclusion

It has been said the Argus is easy to use but hard to master, and you may find that an honest assessment, but the references included at the end of this discussion will quickly lead you to further discovery. Regardless, consider Argus essential as part of your situational awareness arsenal, in both performance and security capacities.

2.2 HeX System

Prerequisites

This LiveCD distribution requires only a system capable of booting from an optical drive. Two nics may be beneficial, depending on your usage. 512MB RAM minimum is recommended.

Introduction

While I just discussed conducting incident analysis with Argus, an excellent stand alone tool, it is considered, in certain circles, to be part of a larger family of tools useful for Network Security Monitoring, or NSM. There have been numerous articles, books, and seminars on Snort and Wireshark and perhaps you've heard of tools like Etherape and Netdude. These tools all serve under the common goal of good NSM practice, but often they require dedicated systems or individual efforts to implement or run. That process can be made a bit easier on you by gathering many invaluable NSM tools, in one LiveCD/LiveUSB offering. Enter HeX, from rawpacket.org. All hail the Packet Monkey!

HeX categorizes its tools into unique subsets designed to aid you with specific efforts like NSM, network based forensics (NBF), network visualization, capture editing, a network toolkit for packet manipulation, as well as pentest (Metasploit) and forensics (Sleuthkit) toolkits. One somewhat atypical element noteworthy with this distribution is the fact that the OS is FreeBSD 6.2, rather than a Linux variant.

This project, under the direction of C.S.Lee, has an extensive roadmap and a wide range of influences. This is a project under constant development; the development team invites feedback and contribution should you be so inclined.

Installation

No real installation challenges with this LiveCD, other than development maturity and some hardware detection issues. If it runs oddly on one machine, boot it on another. I had an issue with video card detection on one of my lab systems (yes, X is included).

The development team included dedicated workspaces that are both logical and humorous.

- WorkaholiC - Normal working environment for web browsing, email and rss reading plus other common daily tasks.
- AnalyzT - Workspace to perform security analysis, all the NSM based tools will be loaded on this workspace.
- HackeR - Workspace to perform network hacking, all the network hacking tools will be launched at this workspace and you can learn about packet crafting here.
- WankeR - Do whatever you want in this workspace, usually instant messaging programs will be launched here.

Remember, if you're running HeX while listening to network traffic, do so via a tap or SPAN port as switched traffic will yield very limited results.

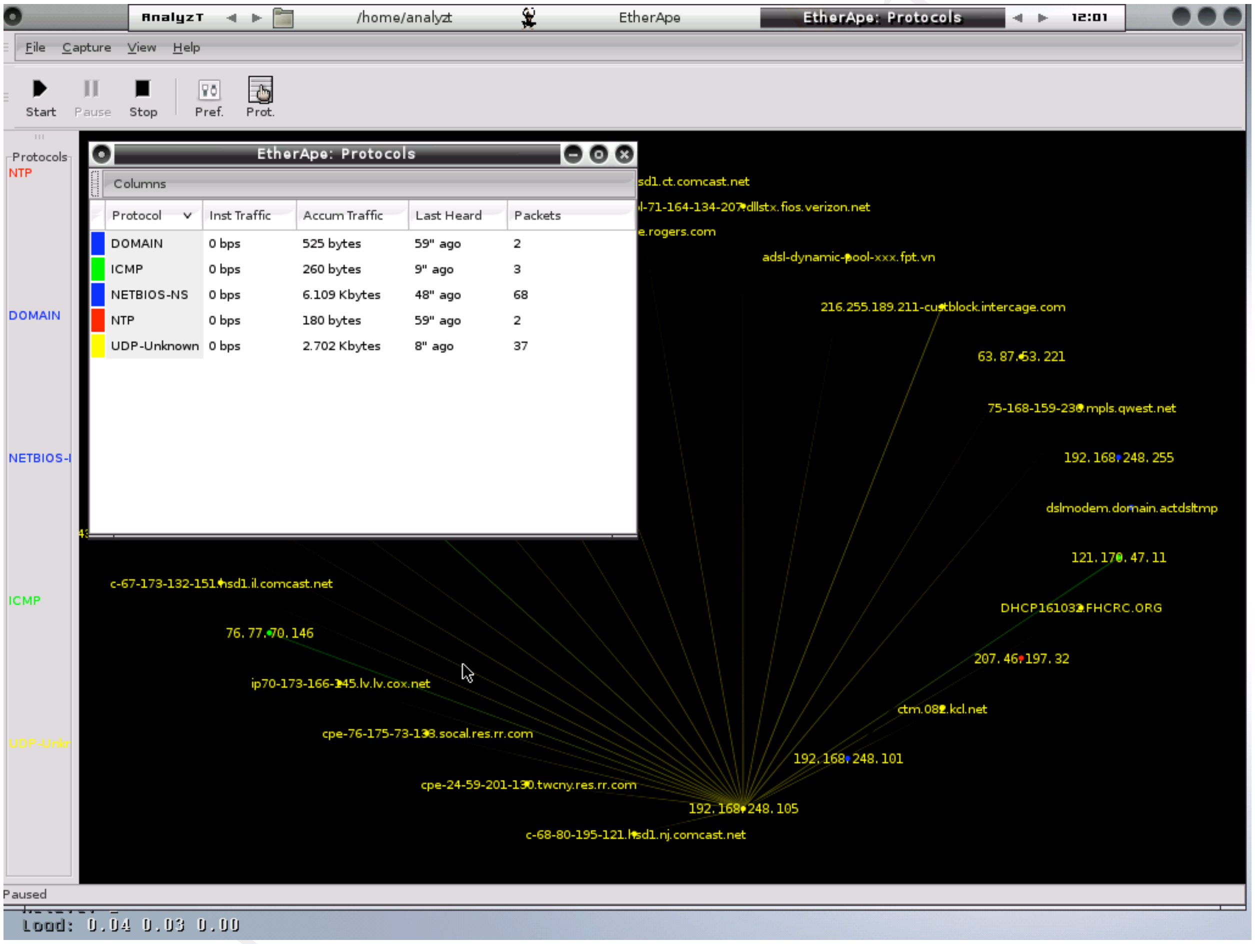
Usage

The variety of tools available on the HeX distribution is so extensive; you'll not likely run out of opportunities for discovery. To present an array of just such opportunity I chose three strong contributors to this distribution's strengths; namely, Etherape, Chaosreader, and NSM-Console.

Etherape

With HeX running as a LiveCD/USB, or as a VMWare virtual machine (my preferred use), engaging Etherape is as simple as right-clicking in desktop empty space, selecting Net-Visual, then Etherape. One of the features most useful in Etherape is default name resolution. Rather than purely IP results, where possible, Etherape displays the hostname after resolving it. This very feature in my use of Etherape via HeX with the fireworks.pcap led me to two discoveries that I'd not noted prior, as I hadn't analyzed the capture in great detail; rather, I had spent time feeding it to AfterGlow for visualization purposes. The discoveries were not shocking or of great merit, but timely and of community service. First, one of the hosts in the P2P mesh inherent to the fireworks.pcap and visualized by Etherape was 216.255.189.211-custblock.intercage.com. Atrivo/Intercage is (was) likely the most reviled ISP in history, long guilty of letting malware run amok globally.[vii] At the time this paper was being written, Intercage was under serious duress and likely meeting its demise. Second, one of the additional hosts that Etherape conveniently identified, by name, was DHCP161032.FHCRC.ORG. As a Seattle area resident, this jumped out at me as I know FHCRC.ORG to be the Fred Hutchinson Cancer Research

Center. As this is an organization with a noble and vital charter, I was immediately concerned to find bot traffic emanating from their network. I contacted their security team moments after the discovery and provided all the details necessary for them to remediate.


AnalyzT
/home/analyzt
EtherApe
EtherApe: Protocols
12:01
File Capture View Help
Start
Pause
Stop
Pref.
Prot.
Protocols
NTP
DOMAIN
NETBIOS-I
ICMP
UDP-Unkn
sd1.ct.comcast.net
l-71-164-134-207.dllstx.fios.verizon.net
e.rogers.com
adsl-dynamic-pool-xxx.fpt.vn
216.255.189.211-custblock.intercage.com
63.87.63.221
75-168-159-230.mpls.qwest.net
192.168.248.255
dslmodem.domain.actdsltmp
121.170.47.11
DHCP161032.FHCRC.ORG
207.46.197.32
ctm.082.kcl.net
192.168.248.101
192.168.248.105
c-67-173-132-151.hsd1.il.comcast.net
76.77.70.146
ip70-173-166-145.lv.lv.cox.net
cpe-76-175-73-138.socal.res.rr.com
cpe-24-59-201-130.twcny.res.rr.com
c-68-80-195-121.hsd1.nj.comcast.net
Paused
Load: 0.04 0.03 0.00

EtherApe: Protocols

Columns

	Protocol	Inst Traffic	Accum Traffic	Last Heard	Packets
🟦	DOMAIN	0 bps	525 bytes	59" ago	2
🟩	ICMP	0 bps	260 bytes	9" ago	3
🟦	NETBIOS-NS	0 bps	6.109 Kbytes	48" ago	68
🟥	NTP	0 bps	180 bytes	59" ago	2
🟨	UDP-Unknown	0 bps	2.702 Kbytes	8" ago	37

Figure 5: Etherape

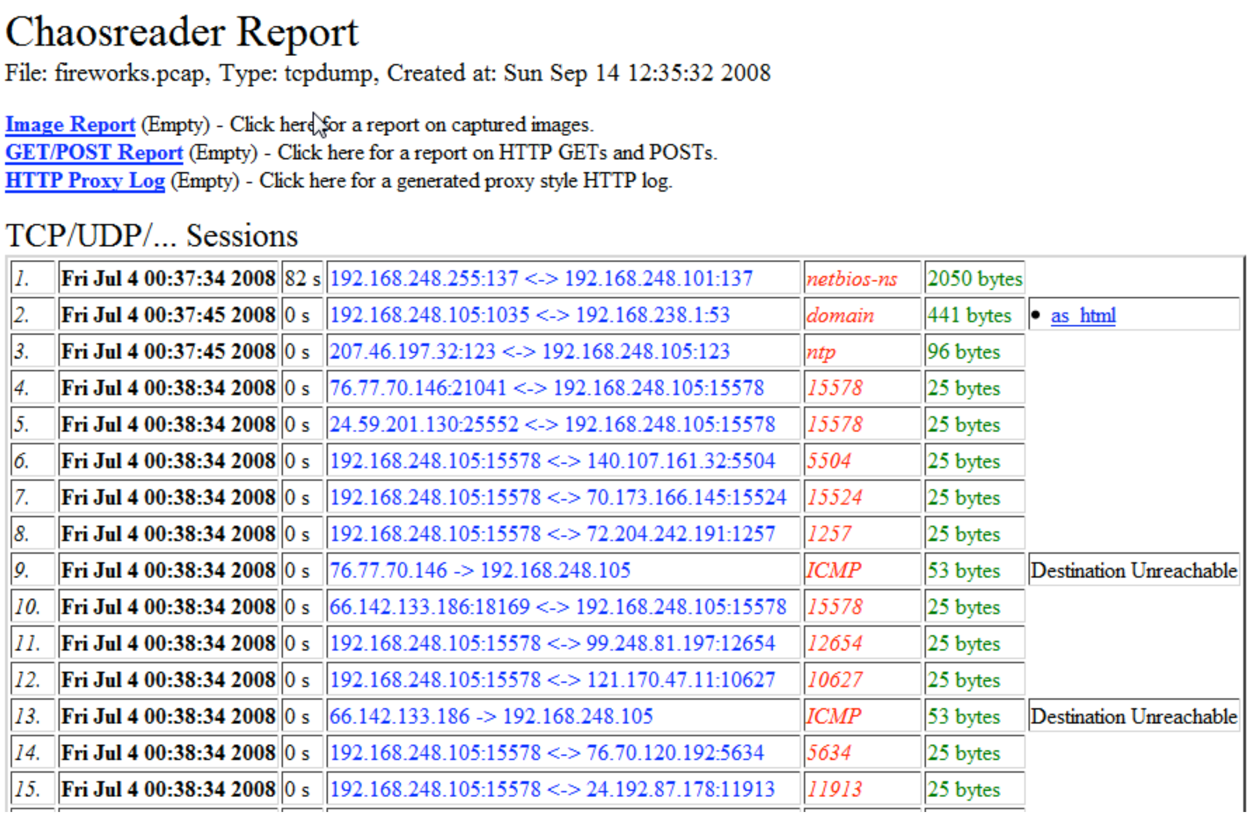
The visualization tweaks are endless. Under Preferences, I typically increase Node Radius Multiplier and Link Width Multiplier as they render node size and link width dependent on the amount of traffic generated.

Chaosreader

In HeX's NBF-Toolkit you'll discover Chaosreader, which can trace various sessions and fetch application data from tcpdump or snoop logs. Like an "any-snarf" program, it will fetch telnet sessions, FTP files, HTTP transfers (HTML, GIF, JPEG, etc.), SMTP, etc. from the captured data. It creates an html index file that links to all the session details, including real-time replay for telnet, rlogin or IRC sessions. Additionally, chaosreader reports images and HTTP GET/POST content. Check out the chaosreader website for more details.[viii]

From the HeX menu choose NBF-Toolkit, then Chaosreader.

Next, run *chaosreader <capture file>*, then browse the resulting *index.html* file. I prefer chaosreader largely for reporting purposes as it generates such report friendly human readable content.



Chaosreader Report

File: fireworks.pcap, Type: tcpdump, Created at: Sun Sep 14 12:35:32 2008

Image Report (Empty) - Click here for a report on captured images.
GET/POST Report (Empty) - Click here for a report on HTTP GETs and POSTs.
HTTP Proxy Log (Empty) - Click here for a generated proxy style HTTP log.

TCP/UDP/... Sessions

1.	Fri Jul 4 00:37:34 2008	82 s	192.168.248.255:137 <-> 192.168.248.101:137	netbios-ns	2050 bytes	
2.	Fri Jul 4 00:37:45 2008	0 s	192.168.248.105:1035 <-> 192.168.238.1:53	domain	441 bytes	• as_html
3.	Fri Jul 4 00:37:45 2008	0 s	207.46.197.32:123 <-> 192.168.248.105:123	ntp	96 bytes	
4.	Fri Jul 4 00:38:34 2008	0 s	76.77.70.146:21041 <-> 192.168.248.105:15578	15578	25 bytes	
5.	Fri Jul 4 00:38:34 2008	0 s	24.59.201.130:25552 <-> 192.168.248.105:15578	15578	25 bytes	
6.	Fri Jul 4 00:38:34 2008	0 s	192.168.248.105:15578 <-> 140.107.161.32:5504	5504	25 bytes	
7.	Fri Jul 4 00:38:34 2008	0 s	192.168.248.105:15578 <-> 70.173.166.145:15524	15524	25 bytes	
8.	Fri Jul 4 00:38:34 2008	0 s	192.168.248.105:15578 <-> 72.204.242.191:1257	1257	25 bytes	
9.	Fri Jul 4 00:38:34 2008	0 s	76.77.70.146 -> 192.168.248.105	ICMP	53 bytes	Destination Unreachable
10.	Fri Jul 4 00:38:34 2008	0 s	66.142.133.186:18169 <-> 192.168.248.105:15578	15578	25 bytes	
11.	Fri Jul 4 00:38:34 2008	0 s	192.168.248.105:15578 <-> 99.248.81.197:12654	12654	25 bytes	
12.	Fri Jul 4 00:38:34 2008	0 s	192.168.248.105:15578 <-> 121.170.47.11:10627	10627	25 bytes	
13.	Fri Jul 4 00:38:34 2008	0 s	66.142.133.186 -> 192.168.248.105	ICMP	53 bytes	Destination Unreachable
14.	Fri Jul 4 00:38:34 2008	0 s	192.168.248.105:15578 <-> 76.70.120.192:5634	5634	25 bytes	
15.	Fri Jul 4 00:38:34 2008	0 s	192.168.248.105:15578 <-> 24.192.87.178:11913	11913	25 bytes	

Figure 6: Chaosreader

The examples in Figure 6 show a mere pittance of the possible output, what you don't see is the all the possible session data, or the IP, TCP port, UDP port, IP protocol, and Ethernet type counts, let alone images, GET/POST requests ,and proxy logs. As this traffic was generated by bot chatter it largely displays high port UDP results.

NSM-Console

Last but not least, and written specifically for HeX by Matthew Lee Hinman, is the NSM-Console, found in the NSM-Toolkit category. The closest comparison, drawn by the project developer, is this: what Metasploit is to exploit modules, NSM-Console is to packet analysis modules. Written in Ruby with the serious packet analyst in mind, NSM-

Console is a framework to run numerous NSM modules against pcap files. The framework will allow you to toggle the modules based on categories like flow, forensics, nsm, and statistics, or you can easily add your own categories. You can also enable/disable modules at your discretion. The NSM-Console version included in version 1.0.3 of HeX is 0.6-DEVEL version, but you can download 0.7-stable (at the time of writing) if you wish at the project website, and take advantage of no less than 29 modules.[ix] NSM-console includes basic versions of tools I've already discussed as standalone or included in HeX, including Argus and Chaosreader. Lee's also done a great screen cast which I highly recommend viewing; you'll find links to it on the project site as well.

Let's run through a quick example of the NSM-Console at work. You can run NSM-Console against single pcaps or a directory with many files in one fell swoop.

After setting the file option, you'll need to choose modules; you can return all available modules and categories by passing *list*. You can also learn more about modules at any time by passing *nsm> info <module>* for more details. Change global options by passing *nsm> options* or change options on a specific module via *nsm> options <module>*. Options might include output, base file, host lists, or logging defaults. You can also set entire categories of modules, including statistics, IDS, nsm, flow, and forensics.

From the HeX menu I chose NSM-Toolkit, then NSM Console.

I left *options* set to default for this example and set my favorite modules as active, then executed *run*:

```
nsm> file /home/analyzt/fireworks.pcap (specify the source file)
nsm> options (set global options)
nsm> output /home/analyzt (define output directory)
nsm> toggle iploc (determines location of all traffic)
nsm> toggle harimau (checks IPs against the harimau blacklist)
nsm> toggle snort (generates snort alerts)
nsm> toggle ip2asn (returns the ASNs for all IPs)
nsm> toggle tshark (analyzes network traffic)
nsm> toggle hash (hashes the pcap file)
nsm> toggle capinfos (extract general pcap info)
nsm> toggle tcpdstat (extracts pcap statistics)
nsm> run (you get the point)
```

Note: I put each module toggle on its own line so as to include the description, but the same can be done in one command as follows:

```
nsm> toggle iploc harimau snort ip2asn tshark has capinfos tcpdstat
```

The results from our run are as follows:

capinfos

```
File name: /home/analyzt/fireworks.pcap
File type: Wireshark/tcpdump/... - libpcap
Number of packets: 340
File size: 34085 bytes
Data size: 28621 bytes
Capture duration: 119.946069 seconds
Start time: Fri Jul  4 00:37:34 2008
End time: Fri Jul  4 00:39:34 2008
Data rate: 238.62 bytes/s
Data rate: 1908.92 bits/s
Average packet size: 84.18 bytes
```

tcpdstat

```
Id: 200807040037
StartTime: Fri Jul  4 00:37:34 2008
EndTime:   Fri Jul  4 00:39:34 2008
TotalTime: 119.95 seconds
TotalCapSize: 0.03MB  CapLen: 506 bytes
# of packets: 340 (27.95KB)
AvgRate: 3.22Kbps  stddev:4.73K

### IP flow (unique src/dst pair) Information ###
# of flows: 162  (avg. 2.10 pkts/flow)
Top 10 big flow size (bytes/total in %):
 29.9%  3.0%  2.7%  2.4%  2.2%  2.1%  2.1%  2.1%  2.0%  2.0%

### IP address Information ###
# of IPv4 addresses: 142
Top 10 bandwidth usage (bytes/total in %):
 69.9% 29.9% 29.9%  3.2%  3.2%  3.0%  2.9%  2.9%  2.9%  2.8%
### Packet Size Distribution (including MAC headers) ###
<<<<
 [   32-   63]:         38
 [   64-  127]:        292
 [  256-  511]:         10
>>>>

### Protocol Breakdown ###
<<<<
     protocol           packets                 bytes       bytes/pkt
------------------------------------------------------------------------
[0] total              340 (100.00%)            28621 (100.00%)     84.18
[1] ip                 339 ( 99.71%)            28579 ( 99.85%)     84.30
[2]  udp               310 ( 91.18%)            26009 ( 90.87%)     83.90
[3]   dns                2 (  0.59%)              525 (  1.83%)    262.50
[3]   other            308 ( 90.59%)            25484 ( 89.04%)     82.74
[2]  icmp               28 (  8.24%)             2510 (  8.77%)     89.64
[2]  igmp                1 (  0.29%)               60 (  0.21%)     60.00
>>>>
```

hash

```
MD5 (/home/analyzt/fireworks.pcap) = bc06b6b4ee79d7cd43a6cf21b95e056d
SHA256 (/home/analyzt/fireworks.pcap) =
128bb169c4db494ef3f2b3e828af629a373e666464b9a09b170fc7c6ededc949
```

harimau

```
no records
```

tshark

```
===================================================================
Protocol Hierarchy Statistics
Filter: frame

frame                                    frames:340 bytes:28621
  eth                                    frames:340 bytes:28621
    ip                                   frames:339 bytes:28579
      udp                                frames:310 bytes:26009
        nbns                             frames:93 bytes:8556
        dns                              frames:2 bytes:525
        ntp                              frames:2 bytes:180
        data                             frames:212 bytes:16681
        udpencap                         frames:1 bytes:67
          esp                            frames:1 bytes:67
      icmp                               frames:28 bytes:2510
      igmp                               frames:1 bytes:60
    arp                                  frames:1 bytes:42
   ===================================================================
```

ip2asn (small snapshot of results)

```
Bulk mode; whois.cymru.com [2008-09-15 03:29:06 +0000]
11530   | 69.68.56.202      | EMBARQ-MNFD - Embarq Corporation
17858   | 125.190.13.191    | KRNIC-ASBLOCK-AP KRNIC
4732    | 210.198.227.155   | DION KDDI CORPORATION
3462    | 59.125.103.124    | HINET Data Communication Business Group
3215    | 90.57.73.32       | AS3215 France Telecom - Orange
6128    | 69.120.82.73      | CABLE-NET-1 - Cablevision Systems Corp.
11456   | 209.177.224.14    | NUVOX - NuVox Communications, Inc.
4515    | 210.177.92.126    | ERX-STAR Star Internet Services Ltd.
4134    | 61.185.220.249    | CHINANET-BACKBONE No.31,Jin-rong Street
19262   | 71.164.144.103    | VZGNI-TRANSIT - Verizon Internet Services Inc.
19262   | 71.172.28.127     | VZGNI-TRANSIT - Verizon Internet Services Inc.
27595   | 216.255.189.211   | INTERCAGE - InterCage, Inc.
22773   | 72.218.118.158    | CCINET-2 - Cox Communications Inc.
17864   | 61.106.193.103    | HANVITIAB-AS-KR Hanvit I&B
6079    | 216.164.142.151   | RCN-AS - RCN Corporation
```

iploc

```
Inbound Addresses:
24.6.219.159,UNITED STATES (US),(Unknown city),,,4
80.33.231.40,SPAIN (ES),(Unknown city),,,1
59.162.52.130,INDIA (IN),(Unknown city),,,1
221.2.165.78,CHINA (CN),(Unknown city),,,4
24.1.135.20,UNITED STATES (US),(Unknown city),,,4
222.254.80.57,VIET NAM (VN),(Unknown city),,,2
121.170.47.11,(Unknown Country?) (XX),(Unknown City?),,,1
216.255.189.211,UNITED STATES (US),Concord CA,37.9733,-122,4
207.46.197.32,UNITED STATES (US),Redmond WA,47.6742,-122.115,1
192.168.248.1,(Private Address) (XX),(Private Address),,,1
200.138.197.250,BRAZIL (BR),(Unknown city),,,8
72.218.118.158,(Unknown Country?) (XX),(Unknown City?),,,4
69.70.90.134,CANADA (CA),Montreal,45.5167,-73.5667,1
66.142.133.186,UNITED STATES (US),Plano TX,33.0462,-96.7467,1
192.168.238.1,(Private Address) (XX),(Private Address),,,1
192.168.248.101,(Private Address) (XX),(Private Address),,,93
```

```
202.132.45.14,TAIWAN (TW),Taipei,25.05,121.517,1
69.253.205.240,(Unknown Country?) (XX),(Unknown City?),,,4
80.87.194.129,RUSSIAN FEDERATION (RU),(Unknown city),,,4
75.73.20.133,UNITED STATES (US),Vallejo CA,38.1075,-122.264,4
222.252.168.218,VIET NAM (VN),(Unknown city),,,1
125.190.13.191,(Unknown Country?) (XX),(Unknown City?),,,4
76.77.70.146,(Unknown Country?) (XX),(Unknown City?),,,1
210.177.92.126,HONG KONG (HK),(Unknown city),,,1
192.168.248.105,(Private Address) (XX),(Private Address),,,179
59.125.103.124,(Unknown Country?) (XX),(Unknown City?),,,9

Outbound Addresses:
192.168.238.1,(Private Address) (XX),(Private Address),,,1
192.168.248.255,(Private Address) (XX),(Private Address),,,1
224.0.0.1,UNITED STATES (US),Sacramento CA,38.5668,-121.467,1
192.168.248.105,(Private Address) (XX),(Private Address),,,23
```

snort

```
[**] [1:2007634:1] BLEEDING-EDGE TROJAN Storm Worm Encrypted Traffic Outbound
- Likely Search by md5 [**]
[Classification: A Network Trojan was detected] [Priority: 1]
07/04/08-00:38:44.375934 192.168.248.105:15578 -> 69.70.90.134:33419
UDP TTL:128 TOS:0x0 ID:472 IpLen:20 DgmLen:53
Len: 25

[**] [1:2007634:1] BLEEDING-EDGE TROJAN Storm Worm Encrypted Traffic Outbound
- Likely Search by md5 [**]
[Classification: A Network Trojan was detected] [Priority: 1]
07/04/08-00:39:04.363678 192.168.248.105:15578 -> 96.33.86.211:20092
UDP TTL:128 TOS:0x0 ID:518 IpLen:20 DgmLen:53
Len: 25

[**] [1:2007634:1] BLEEDING-EDGE TROJAN Storm Worm Encrypted Traffic Outbound
- Likely Search by md5 [**]
[Classification: A Network Trojan was detected] [Priority: 1]
07/04/08-00:39:14.876621 192.168.248.105:15578 -> 125.161.178.73:21238
UDP TTL:128 TOS:0x0 ID:562 IpLen:20 DgmLen:53
Len: 25
```

The Snort results from NSM-console give us the final clue in our HeX use. Remember, I'm assuming the role of the analyst who is not yet clear on the devil in the details. If NSM-console hasn't pulled it together from a packet capture perspective, nothing will.

Now imagine using NSM-Console against entire directories of pcaps, with modules most useful to your investigation selected. You just saved a ton of time on your analysis by switching to NSM-console.

To quote Peyton Manning, “You’re feeling me. You love it.”

This framework really represents an aggregation of the best packet analysis tools, and rather than reinventing the wheel, it defines packet monkey efficiency.

Benefits and Drawbacks

The HeX project is growing, with lots of community involvement, and is extremely well intended and headed in the right direction, with a development team and leadership working in earnest to keep it current and relevant. The roadmap includes a 2.0 release in the near future (at the time of this writing).

For users with no *nix skills, this distribution may present some challenges. While purists will tout the strength of FreeBSD until they no longer draw breath, fans of desktop friendly Linux distros may face some challenges here. HeX does present a great opportunity to strengthen your chops with an OS you may be less familiar with.

Conclusion

For NSM practitioners, this offering is a dream come true. For packet analysts you’ll find no better gathering of discipline specific tools in one distribution.

2.3 NetworkMiner

Prerequisites

Winpcap
Windows XP recommended, but works well for static analysis on Vista.

Introduction

I had the pleasure of participating in the 20th Annual FIRST Conference in Vancouver, B.C., as a speaker and attendee, just before beginning the process of my initial study of NetworkMiner. Given that FIRST is the Forum of Incident Response and Security Teams, I was hopeful that I would discover some relevant tools for research and I wasn't disappointed. That said, the most influential conversation I enjoyed at the conference was a chat with Richard Bejtlich (The Tao of Network Security Monitoring) and Raffael Marty (Applied Security Visualization) where the discussion's focus was largely on new ways to interpret network data captures. Having long embraced network security monitoring and more recently, security data visualization, I went searching for new tools that bring a different element to traffic analysis. Enter NetworkMiner 0.85, the strong results of Erik Hjelmvik's development efforts.

Erik was kind enough to provide me with a number of details regarding NetworkMiner. For instant gratification though, you can find almost everything you need on the NetworkMiner wiki at http://networkminer.wiki.sourceforge.net/NetworkMiner.

Erik’s goal is for NetworkMiner to become a full blown Network Forensic Analysis Tool (NFAT), available for free as an open source application. NetworkMiner is focused on the extraction of relevant events and information about hosts and users on a network, and providing that information in an intuitive user interface. Further focus is on analyzing and parsing PCAP files rather than on performing live sniffing with NetworkMiner. Simply, there are several other applications that are better at sniffing packets like Wireshark or tcpdump. Erik goes so far as to not recommend the use of a Windows OS if hoping to perform packet sniffing properly on a high speed network. That being said, NetworkMiner can be used to sniff data, either by using WinPcap or by using Raw Sockets. NetworkMiner is an excellent compliment to network security monitoring systems as a tool for attack investigation, and it can also be used to conduct behavior analysis of a compromised machine, potential rogue host or malicious user.

Some of the things planned for future implementation are:

- A proper reporting tool
- Faster parsing of large PCAP files
- Implement even more protocols
- Statistical methods to do protocol identification (protocol fingerprinting) of a TCP session or UDP data, or identifying the correct protocol based on the TCP/UDP packet content rather than port number, eliminating non-standard port identification failures. See Bejtlich’s discussion regarding PIPI (Port Independent Protocol Identification) and Dynamic Application-Layer Protocol Analysis.[x]

Erik maintains a list of more minor features he’s planning to add to NetworkMiner at:

http://sourceforge.net/tracker/?group_id=189429&atid=929293

If you'd like to get a look at new upcoming versions of NetworkMiner, as well as have access to a large amount of PCAP files, apply for a membership to the private NetworkMiner beta testers mailing list at:

https://lists.sourceforge.net/lists/listinfo/networkminer-betatesters

Before detailing a bit of NetworkMiner usage, allow me to highlight its capabilities as a forensic data collector.

1. OS Fingerprinting
    a. TCP SYN and SYN+ACK using OS fingerprinting databases from p0f and Ettercap
    b. DHCP via the Satori OS fingerprinting database from FingerBank.
    c. The MAC-vendor list from Nmap.
2. File extraction via PCAP parsing with supported protocols including FTP, HTTP, and SMB.
3. Credentials grabbing from supported protocols.
4. Clear text parsing inclusive of keyword search functionality.
5. Wireless sniffing and parsing with AirPcap adapters.[xi]

Using NetworkMiner

One negative, and I'm more prone to blaming Windows Vista than anything else, but NetworkMiner on Windows Vista gets pretty hosed up looking for wpcap.dll, even if you *Run as Administrator*. This issue really only affects conducting captures from the Vista PC which, like Erik, I recommend against. You can use Raw Sockets but reliability suffers. Instead, grab your PCAPs from a *nix host and conduct static analysis on the Windows machine running NetworkMiner.

NetworkMiner certainly lives up to its name. The UI is incredibly simple, but there are some subtleties that, once uncovered, will leave you smiling at the realization of the tool’s usefulness. The UI will offer you a network adapter drop-down menu, and nine tabs giving you scads of data on hosts, frames, files, images, credentials, parameters, keywords, clear text, and anomalies.

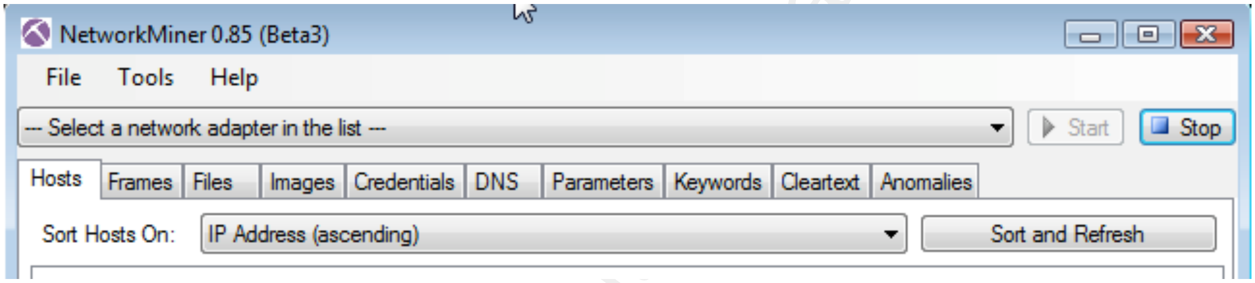
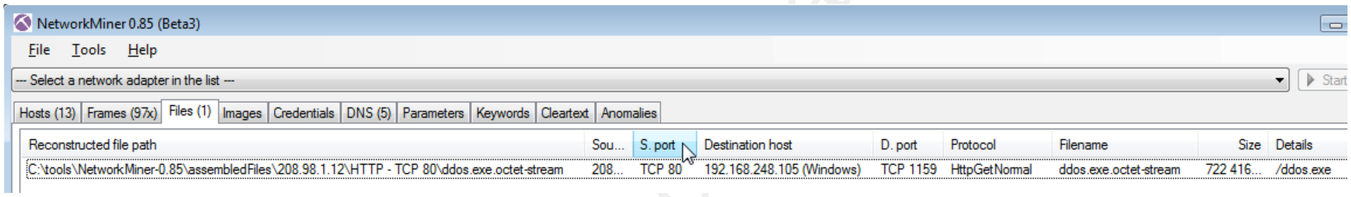


Figure 7: NetworkMiner UI

For NetworkMiner, in addition to fireworks.pcap, I utilized a pcap taken from sandbox execution of an IRC based Trojan with a binary referred to as camda.exe. The resulting camda.pcap taken during analysis allowed me to validate the strengths of NetworkMiner in a distinctive fashion. For your own testing, I’m offering up camda.pcap via email request only, at holisticinfosec@gmail.com. WARNING: The domain name you will see in the Hosts tab, while inactive, is both adult and hostile, with registrar originations in Turkey. Further, you will be reconstructing actual malware if you use this PCAP while testing NetworkMiner.

Starting with the *Files* view, I can immediately determine that I received a little present named ddos.exe from our friends at *<content removed for the children in the audience>*.info. Hmm…I wonder what ddos.exe does. NetworkMiner rebuilt it from the HTTP GET request and wrote ddos.exe.octet-stream to the assembled files directory in the default NetworkMiner hierarchy. If you feed the .exe.octet-stream to VirusTotal.com you will find that the payload was identified 24 out of 33 times here: http://www.virustotal.com/analisis/4e1dad92775925b844b397a0be133eae.

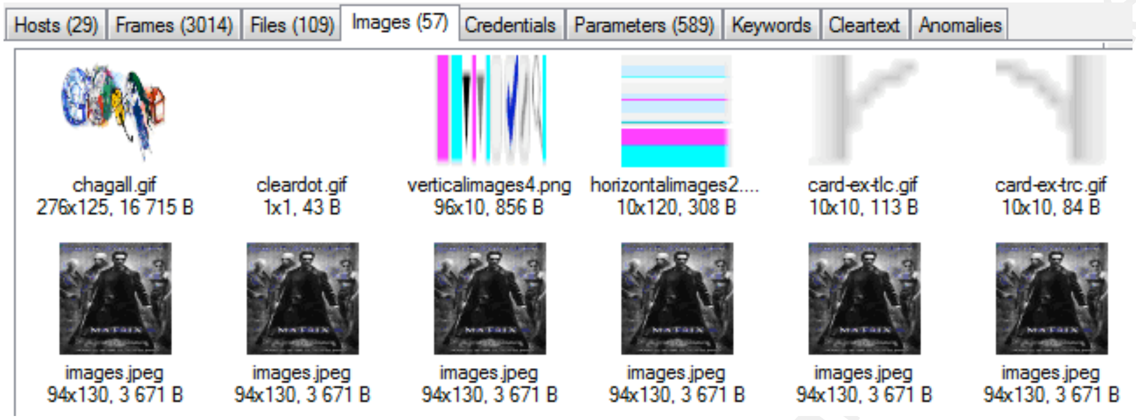


Reconstructed file path	Sou...	S. port	Destination host	D. port	Protocol	Filename	Size	Details
C:\tools\NetworkMiner-0.85\assembledFiles\208.98.1.12\HTTP - TCP 80\ddos.exe.octet-stream	208...	TCP 80	192.168.248.105 (Windows)	TCP 1159	HttpGetNormal	ddos.exe.octet-stream	722 416...	/ddos.exe

Figure 8: NetworkMiner reconstructs ddos.exe

NetworkMiner will grab certificates for you in a similar fashion, and is even useful for building media files from streams.

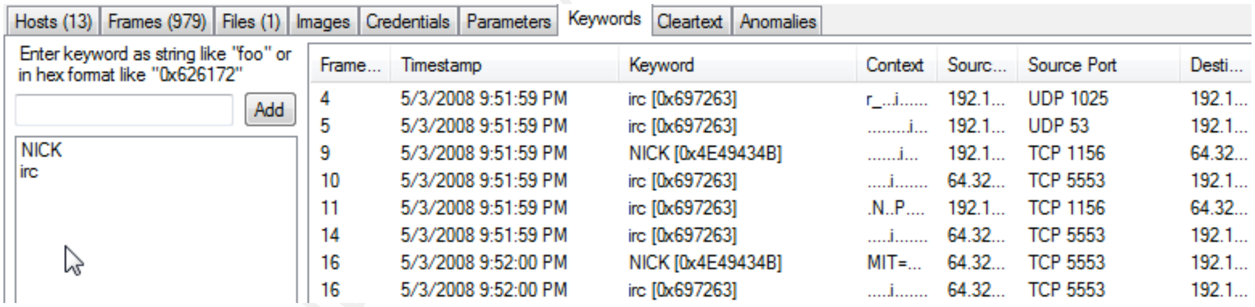
You'll likely find the *Images* tab interesting as well. I utilized a generic capture for this to demonstrate the functionality; the *Images* feature is quite similar to Driftnet. I conducted a Google Image search for images from The Matrix (I know, really original).



chagall.gif 276x125, 16 715 B
cleardot.gif 1x1, 43 B
verticalimages4.png 96x10, 856 B
horizontalimages2.... 10x120, 308 B
card-ex-tlc.gif 10x10, 113 B
card-ex-trc.gif 10x10, 84 B
images.jpeg 94x130, 3 671 B
images.jpeg 94x130, 3 671 B
images.jpeg 94x130, 3 671 B
images.jpeg 94x130, 3 671 B
images.jpeg 94x130, 3 671 B
images.jpeg 94x130, 3 671 B

Figure 9: Images mined

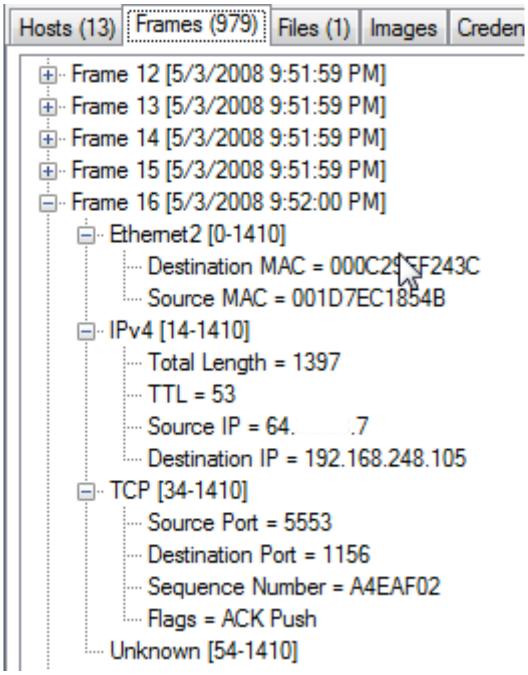
The *Keywords* feature is great for typical forensic discovery. Using *Tools-Reset Capture Data* before re-opening camda.pcap, I added the keywords irc and NICK, and then opened the capture.



Frame...	Timestamp	Keyword	Context	Sourc...	Source Port	Desti...
4	5/3/2008 9:51:59 PM	irc [0x697263]	r_..i......	192.1...	UDP 1025	192.1...
5	5/3/2008 9:51:59 PM	irc [0x697263]	.........i...	192.1...	UDP 53	192.1...
9	5/3/2008 9:51:59 PM	NICK [0x4E49434B]	.......i...	192.1...	TCP 1156	64.32...
10	5/3/2008 9:51:59 PM	irc [0x697263]	....i.......	64.32...	TCP 5553	192.1...
11	5/3/2008 9:51:59 PM	irc [0x697263]	.N..P....	192.1...	TCP 1156	64.32...
14	5/3/2008 9:51:59 PM	irc [0x697263]	....i.......	64.32...	TCP 5553	192.1...
16	5/3/2008 9:52:00 PM	NICK [0x4E49434B]	MIT=...	64.32...	TCP 5553	192.1...
16	5/3/2008 9:52:00 PM	irc [0x697263]	....i.......	64.32...	TCP 5553	192.1...

Figure 10: Typical IRC chatter

The bonus of the keyword search is the fact that it points me to the associated frame, seen in the *Frames* tab.



```
⊞ Frame 12 [5/3/2008 9:51:59 PM]
⊞ Frame 13 [5/3/2008 9:51:59 PM]
⊞ Frame 14 [5/3/2008 9:51:59 PM]
⊞ Frame 15 [5/3/2008 9:51:59 PM]
⊟ Frame 16 [5/3/2008 9:52:00 PM]
    ⊟ Ethernet2 [0-1410]
        Destination MAC = 000C29 F243C
        Source MAC = 001D7EC1854B
    ⊟ IPv4 [14-1410]
        Total Length = 1397
        TTL = 53
        Source IP = 64.      .7
        Destination IP = 192.168.248.105
    ⊟ TCP [34-1410]
        Source Port = 5553
        Destination Port = 1156
        Sequence Number = A4EAF02
        Flags = ACK Push
    Unknown [54-1410]
```

Figure 11: Frames - Where's the bad guy?

Frames kindly offers up the fact that the IRC chatter is occurring with 64.x.y.7 (hope it's not yours). Taking a quick peek in the *Hosts* tab will confirm our suspicion; 64.x.y.7 is indeed an IRC server.

The *Credentials* tab should be obvious; if you're passing credentials in the clear your goodies are up for grabs. Don't forget to check on *Anomalies* for errant behavior, and the *Cleartext* tab will definitely give up some likely data to narrow down via *Keywords*. The *Parameter* tabs will even satisfy the web crawler in you, identifying exactly what you imagined: input variable/parameters and the strings passed to them.

Benefits and Drawbacks

The benefits from NetworkMiner use are endless. The almost instantaneous forensic discovery the tool allows simply speaks for itself. If I force myself to find a drawback, it might be the fact that you'll likely uncover too much information and may have to review your privacy policies before proceeding. It is a young tool, and there are numerous functionality enhancements pending, but suffice it to say that if Erik brings them all to light, I'm willing to go right out on a limb here and recommend it for Fyodor's Top 100 Network Security Tools.[xii]

In Conclusion

There are certain tools incident responders should always have in their toolkits. NetworkMiner is one of those tools. It's easy to use, you'll be underway in no time, and the resulting data will be of assistance no matter the forensic circumstance. I'm certain you will find immediate use for it; if not for you, for someone on your team.

3 Malcode Analysis

3.1 Malcode Analysis Software Tools

Prerequisites

Windows 2000 or XP

Introduction

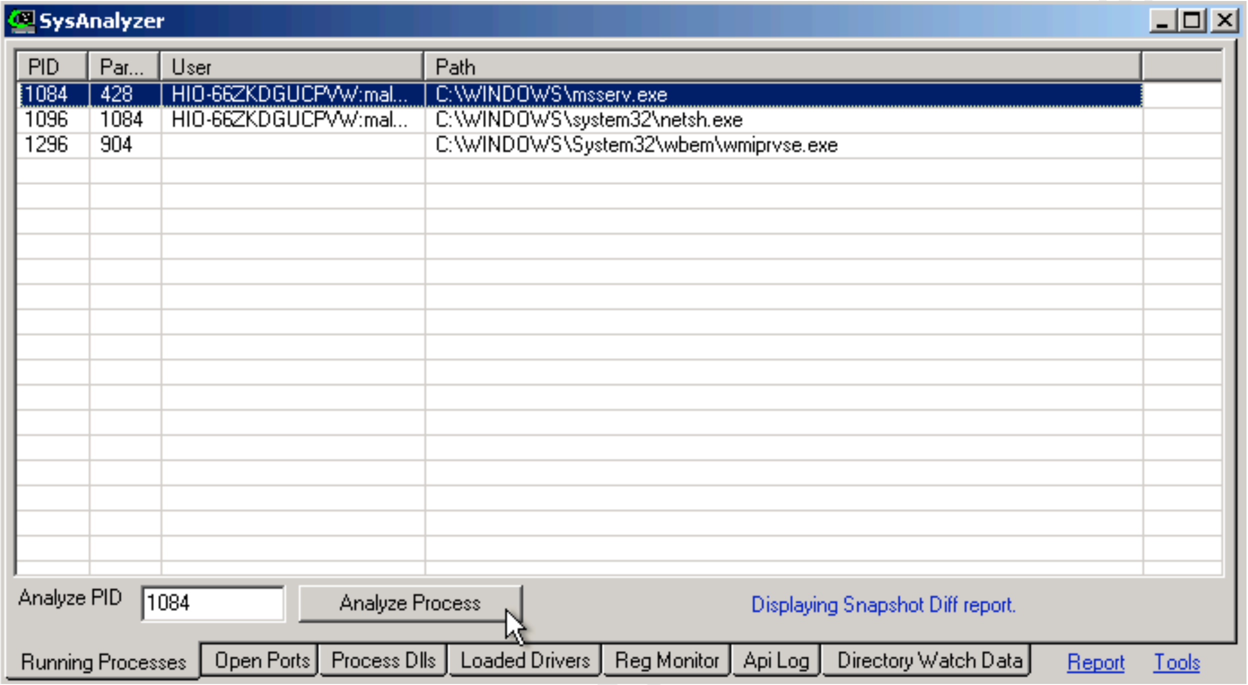
For bug hunters there are a great many tools available, from simple command line essentials such as *strings* or *netstat*, to root kit detectors like Helios, or Joe Stewart's Truman. From iDefense Labs (a VeriSign division) you'll find an excellent set of tools for malcode analysis on Windows PCs that provide detailed discovery. For further reading on malware analysis give Joe Stewart's work a read on the Secureworks blog at http://www.secureworks.com/research/threats/ and take a close look at Lenny Zeltser's paper on reverse-engineering malware at http://www.zeltser.com/reverse-malware-paper/. There are four malcode analysis offerings on the iDefense site but for this effort I'll cover three, specifically SysAnalyzer, Malcode Analysis Pack, and MultiPot.

SysAnalyzer

SysAnalyzer is described on the iDefense Labs site as "an automated malcode run time analysis application that monitors various aspects of system and process states." From the SysAnalyzer overview comes one **critical note:** SysAnalyzer is not a sandboxing utility. Target executables are run in a fully live test on the system. Thus, if you are testing malicious code, your test system will be infected.[xiii] The simplest method to test malware under these circumstances is using one of the free VMWare solutions like VMWare Server or VMWare Fusion for Mac, where you can take a *Snapshot*, then *Revert* when your research is complete.

In just such an environment I fired up a Windows XP victim and fed SysAnalyzer fireworks.exe. SysAnalyzer’s initial UI is an efficient little wizard that offers additional options to use Sniff Hit, API Logger, and Directory Watcher. I selected API Logger and Directory Watcher and clicked Start.

The SysAnalyzer view, after execution, includes *Running Processes, Open Ports, Process Dlls, Loaded Drivers, Reg Monitor,* and *Directory Watch Data*. *Running Processes* immediately advised of a new process called msserv.exe. When viewing the *Running Processes* tab you’re afforded the additional opportunity to *Analyze Process* which will spawn Process Analyzer, where you can right-click a PID of your choice, 1084 as seen in Figure 7, representing msserv.exe. This will, in turn, *List Data* on the specific process including MD5, packer, and file properties as well as run it through some basic exploit signatures in an attempt to identify the malcode.



PID	Par...	User	Path
1084	428	HIO-66ZKDGUCPVW:mal...	C:\WINDOWS\msserv.exe
1096	1084	HIO-66ZKDGUCPVW:mal...	C:\WINDOWS\system32\netsh.exe
1296	904		C:\WINDOWS\System32\wbem\wmiprvse.exe

Analyze PID 1084

Displaying Snapshot Diff report.

Figure 12: SysAnalyzer

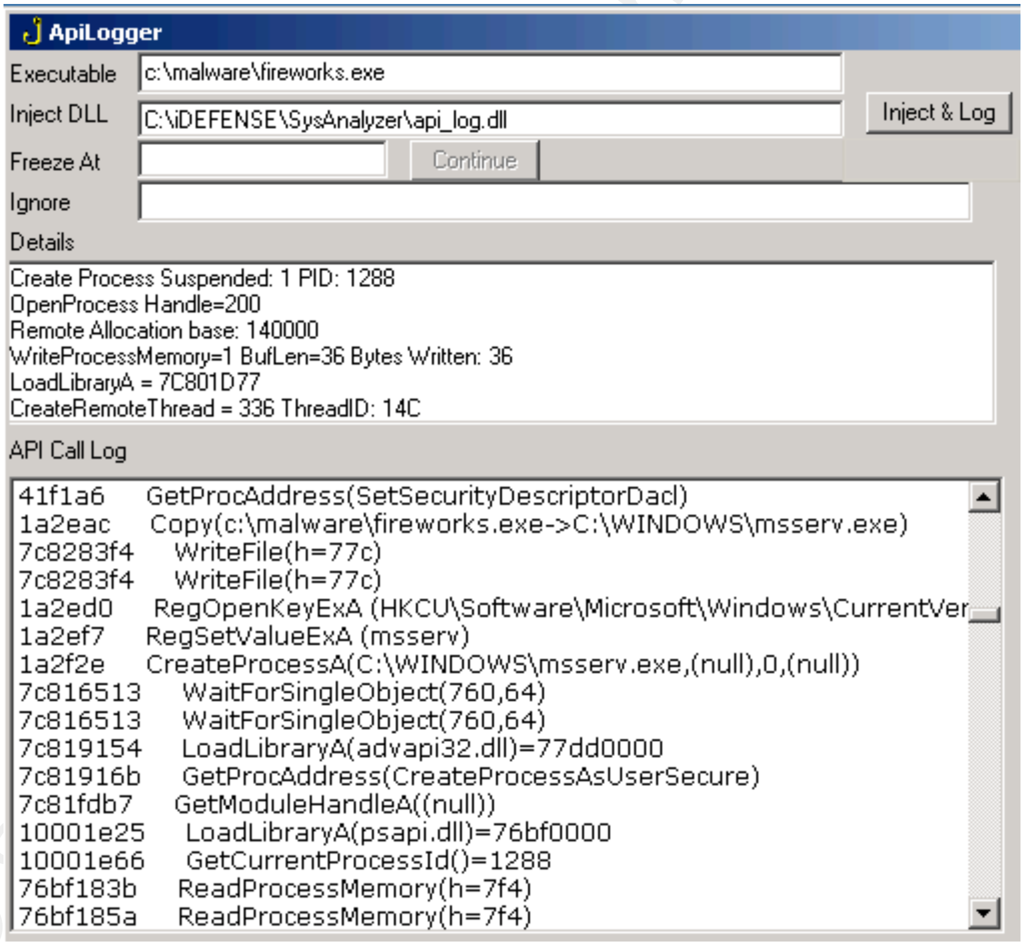
I found an entry in *Directory Watch Data* that also indicated *Created: C:\WINDOWS\msserv.exe*, along with *Created: C:\WINDOWS\msserv.config*, the typical peer list Storm utilizes to define the victim ID and port as well as create its P2P mesh.

Reg Monitor let me know that an entry had been made at HKCU\Software\Microsoft\Windows\CurrentVersion\Run for msserv.exe.

SysAnalyzer will also present you with yet more *Tools* including *Snapshot* options and the ability to create a *Known File DB*. Snapshot capabilities are useful for comparing snapshots over a time interval of your choosing.

I chose to run *ApiLogger* in standalone mode (in the SysAnalyzer menu) with our fireworks.exe sample. ApiLogger adds real-time API logging to the analysis output by injecting a dll into the target process. Once loaded, the dll will insert a series of detour-style hooks into specific api calls. When these APIs are accessed by any code in the process, they will trigger a notification message, which is sent to the main SysAnalyzer interface, or the API Call Log in standalone mode.[xiv]

The results at the end the *Inject & Log* process, outlined a couple of malware indicative traits. At address 1a2f2e we see the msserv.exe process created.



Executable: c:\malware\fireworks.exe

Inject DLL: C:\iDEFENSE\SysAnalyzer\api_log.dll

Details

```
Create Process Suspended: 1 PID: 1288
OpenProcess Handle=200
Remote Allocation base: 140000
WriteProcessMemory=1 BufLen=36 Bytes Written: 36
LoadLibraryA = 7C801D77
CreateRemoteThread = 336 ThreadID: 14C
```

API Call Log

```
41f1a6    GetProcAddress(SetSecurityDescriptorDacl)
1a2eac    Copy(c:\malware\fireworks.exe->C:\WINDOWS\msserv.exe)
7c8283f4    WriteFile(h=77c)
7c8283f4    WriteFile(h=77c)
1a2ed0    RegOpenKeyExA (HKCU\Software\Microsoft\Windows\CurrentVer
1a2ef7    RegSetValueExA (msserv)
1a2f2e    CreateProcessA(C:\WINDOWS\msserv.exe,(null),0,(null))
7c816513    WaitForSingleObject(760,64)
7c816513    WaitForSingleObject(760,64)
7c819154    LoadLibraryA(advapi32.dll)=77dd0000
7c81916b    GetProcAddress(CreateProcessAsUserSecure)
7c81fdb7    GetModuleHandleA((null))
10001e25    LoadLibraryA(psapi.dll)=76bf0000
10001e66    GetCurrentProcessId()=1288
76bf183b    ReadProcessMemory(h=7f4)
76bf185a    ReadProcessMemory(h=7f4)
```

Figure 13: ApiLogger

At address 4a2f72 we see an oldie but goodie where msserv.exe sets itself as allowed through the Windows Firewall via *netsh firewall set allowedprogram "C:\WINDOWS\msserv.exe"*, followed immediately by the *run on startup* registry key.

Finally, starting at address 4a2bce we see the peer list created, followed by *accept, bind, closesocker, connect, getpeername, getsockname, getsockopt*, etc., typical of Storm as it connects to its peers.

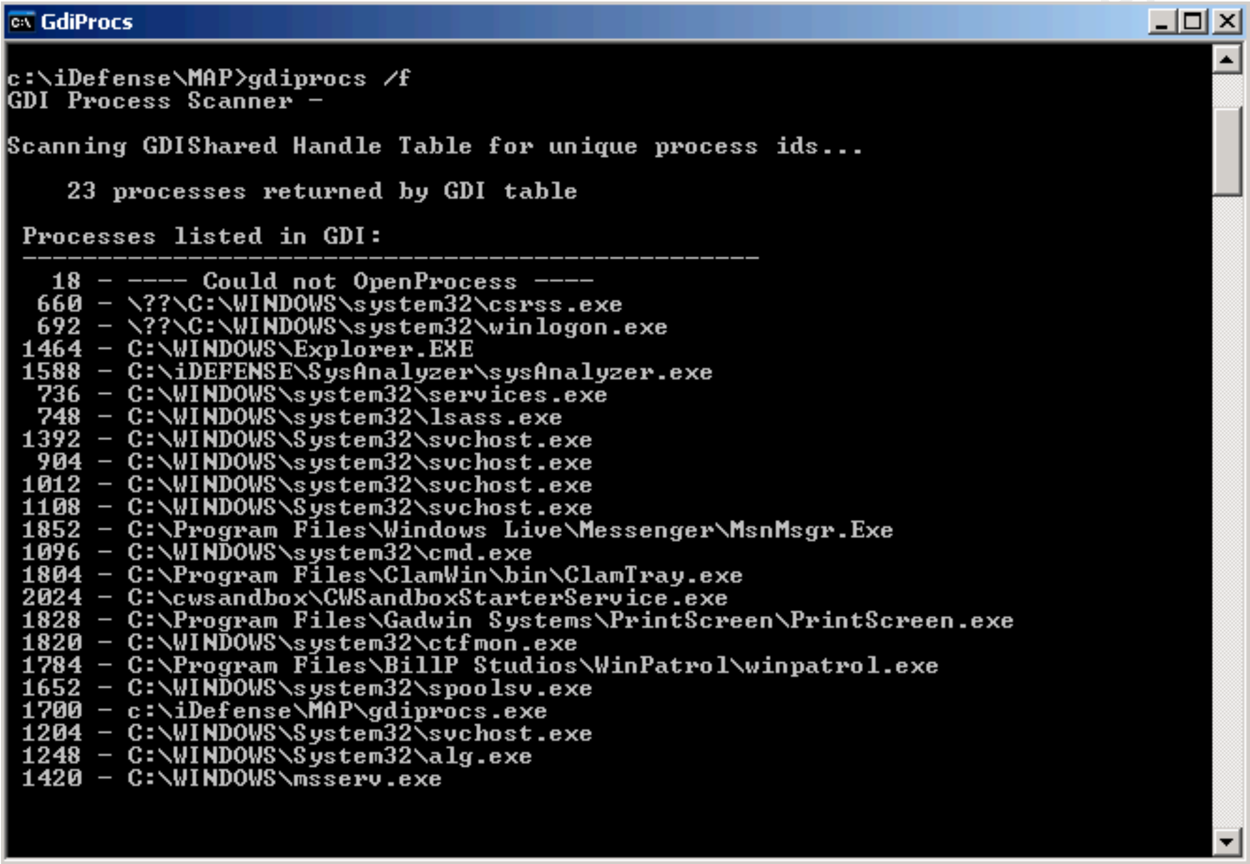
SysAnalyzer is an excellent framework in which to "quickly collect, compare, and report on the actions" taken by malware on a system.

Malcode Analysis Pack

The Malcode Analysis Pack, or MAP, offers an extensive list of features with which to investigate malware samples:

- ShellExt - 4 explorer shell extensions
- socketTool - manual TCP Client for probing functionality.
- MailPot - mail server capture pot
- fakeDNS - spoofs dns responses to controlled ip's
- sniff_hit - HTTP, IRC, and DNS sniffer
- sclog - Shellcode research and analysis application
- IDCDumpFix - aids in quick RE of packed applications
- Shellcode2Exe - embeds multiple shellcode formats in exe husk
- GdiProcs - detect hidden processes

GDIProcs, run with the /f switch to show the full path of all visible processes, immediately took note of msserv.exe in C:\WINDOWS, although this is not exactly a revelation as msserv.exe does nothing to hide itself.

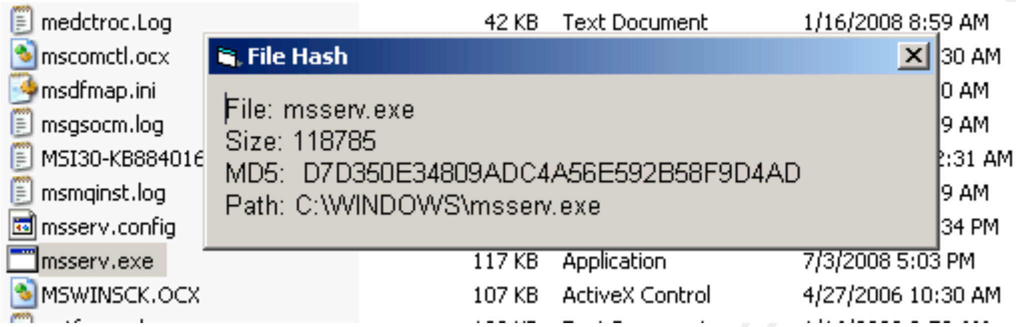


```
c:\iDefense\MAP>gdiprocs /f
GDI Process Scanner -

Scanning GDIShared Handle Table for unique process ids...

    23 processes returned by GDI table

 Processes listed in GDI:
 -------------------------------------------------
   18 - ---- Could not OpenProcess ----
  660 - \??\C:\WINDOWS\system32\csrss.exe
  692 - \??\C:\WINDOWS\system32\winlogon.exe
 1464 - C:\WINDOWS\Explorer.EXE
 1588 - C:\iDEFENSE\SysAnalyzer\sysAnalyzer.exe
  736 - C:\WINDOWS\system32\services.exe
  748 - C:\WINDOWS\system32\lsass.exe
 1392 - C:\WINDOWS\System32\svchost.exe
  904 - C:\WINDOWS\system32\svchost.exe
 1012 - C:\WINDOWS\system32\svchost.exe
 1108 - C:\WINDOWS\System32\svchost.exe
 1852 - C:\Program Files\Windows Live\Messenger\MsnMsgr.Exe
 1096 - C:\WINDOWS\system32\cmd.exe
 1804 - C:\Program Files\ClamWin\bin\ClamTray.exe
 2024 - C:\cwsandbox\CWSandboxStarterService.exe
 1828 - C:\Program Files\Gadwin Systems\PrintScreen\PrintScreen.exe
 1820 - C:\WINDOWS\system32\ctfmon.exe
 1784 - C:\Program Files\BillP Studios\WinPatrol\winpatrol.exe
 1652 - C:\WINDOWS\system32\spoolsv.exe
 1700 - c:\iDefense\MAP\gdiprocs.exe
 1204 - C:\WINDOWS\System32\svchost.exe
 1248 - C:\WINDOWS\System32\alg.exe
 1420 - C:\WINDOWS\msserv.exe
```

Figure 14: GDI Process Scanner

When Storm includes a mass mailer it provides ample fodder for MailPot which captures email sent out by trojans and mass mailers. If the malware uses Outlook automation you can configure your Outlook client to use MailPot or if it connect to an open relay by domain name use MailPot with fakeDNS to redirect it.[xv]

The Shell Extensions are context menu gems, including *Strings* and *MD5 Hash*, with right-click convenience. *MD5 Hash* lists the target file name, size in bytes and MD5 hash. This is always an indispensable method of identification. Using a search engine to query the hash of your sample is almost always results in a likely source of information.



```
File: msserv.exe
Size: 118785
MD5:  D7D350E34809ADC4A56E592B58F9D4AD
Path: C:\WINDOWS\msserv.exe
```

Figure 15: MAP – MD5 Hash

Strings provides invaluable information about certain behavioral attributes of malware as it extracts all ASCII and Unicode strings from the specified file and displays the results. The MAP stings will also pull the MD5 for you. Sometimes as you're reviewing the output you may be offered something useful. Strings run against fireworks.exe wasn't all that revealing with the exception of the very last bit of output that included the text reference *sfdbee*. This stood out to me as unique and possible useful having seen it roll by during API logging. I searched it, and sure enough, the results immediately referred to msserv.exe and Peacomm (a Storm moniker). Theoretically then, and analyst could have reached the conclusion that this was a Storm variant with little more than strings and an md5 hash.

MultiPot

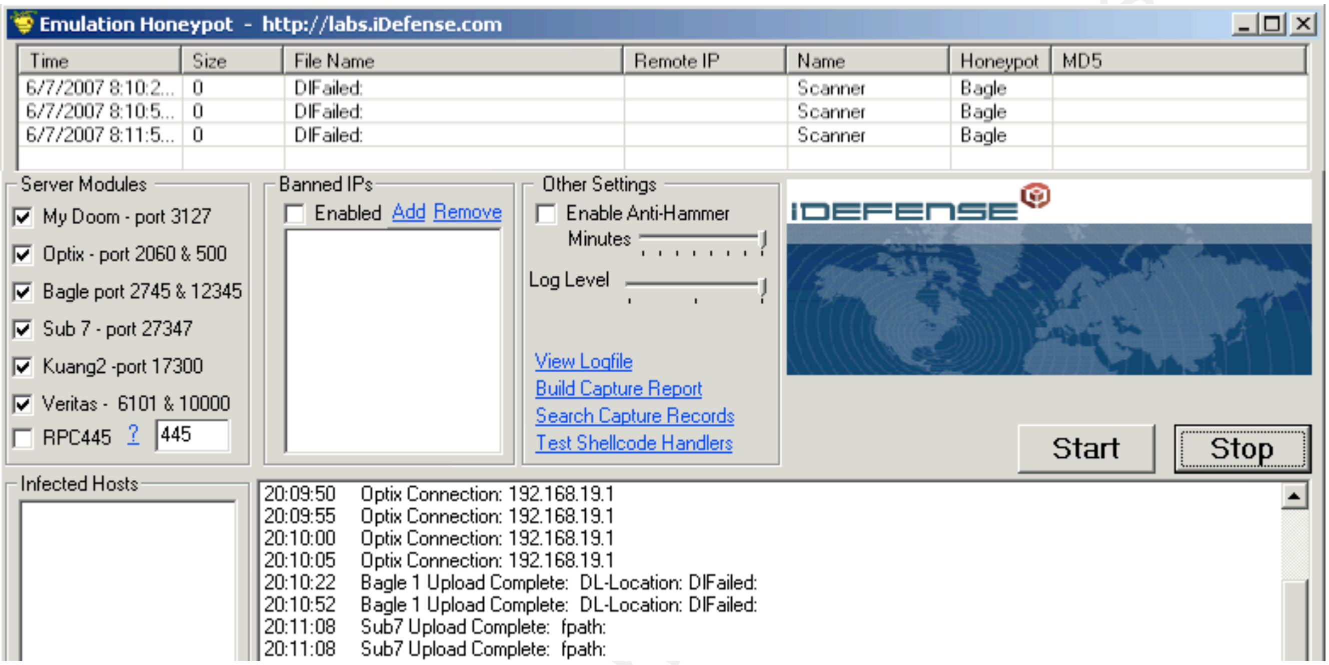
Consider MultiPot a sidebar as it did not aid in my study of the Storm sample, but it is interesting none the less as part of the Malcode Analysis Software Tools family. MultiPot is an emulation-based honeypot designed to capture malicious code as it spreads via various exploits across the net. The captures are such that the host machine requires minimal supervision and is not itself at risk of infection. It was designed specifically to emulate exploitable services in order to safely collect malicious code.

You might find MultiPot useful as an ISP monitoring your network or as corporate security personnel watching for outbreaks. It might also be useful to security and virus researchers to build statistics or collect samples.[xvi]

MultiPot is very simple to setup, and is offered under the GPL, so you can craft your own handlers or modify those included. In fact, all these tools exist under the GPL, leaving additional opportunities to experiment. Source code is available in the installation or on the iDefense website. MultiPot includes protective measures to avoid disk flooding and the frequency of uploads and for shellcoders, it includes five shellcode handlers which represents the most commonly seen shellcodes at the time this app was created. Each of these handlers can be tested individually.

MultiPot was last updated in 2005, so server modules are quite dated, but it is worthy of experimenting with.

While writing this, I didn’t actually expose MultiPot to the internet, but I did pseudo-fake it out with nmap. The results are seen in Figure 11.



Time	Size	File Name	Remote IP	Name	Honeypot	MD5
6/7/2007 8:10:2...	0	DlFailed:		Scanner	Bagle	
6/7/2007 8:10:5...	0	DlFailed:		Scanner	Bagle	
6/7/2007 8:11:5...	0	DlFailed:		Scanner	Bagle	

Server Modules
- ☑ My Doom - port 3127
- ☑ Optix - port 2060 & 500
- ☑ Bagle port 2745 & 12345
- ☑ Sub 7 - port 27347
- ☑ Kuang2 -port 17300
- ☑ Veritas - 6101 & 10000
- □ RPC445 ? 445

Banned IPs
- □ Enabled Add Remove

Other Settings
- □ Enable Anti-Hammer
- Minutes
- Log Level

```
20:09:50   Optix Connection: 192.168.19.1
20:09:55   Optix Connection: 192.168.19.1
20:10:00   Optix Connection: 192.168.19.1
20:10:05   Optix Connection: 192.168.19.1
20:10:22   Bagle 1 Upload Complete:  DL-Location: DlFailed:
20:10:52   Bagle 1 Upload Complete:  DL-Location: DlFailed:
20:11:08   Sub7 Upload Complete:  fpath:
20:11:08   Sub7 Upload Complete:  fpath:
```

Figure 16: Multipot

Benefits and Drawbacks

Cost to use these tools from iDefense Labs? Zero dollars.

Cost to buy similar commercial offerings? Hundreds or thousands of dollars.

Value of the knowledge gained from using MAP or SysAnalyzer? Priceless.

There are no drawbacks other than the normal malware investigation caution flags (no researching malware in production environments).

In Conclusion

Studying malware like Storm is an endless and evolving process but tools like MAP and SysAnalyzer offer significant aid in that process. They represent an ample framework for experimentation and research.

When taking a closer look at malware, as an incident handler, or as a system administrator, these tools will serve you well.

3.2 Mandiant Red Curtain

Prerequisites

Windows XP or higher console use
Windows 2000 or higher for agent use
Microsoft .NET 2.0 framework for the MRC console
http://www.microsoft.com/downloads/details.aspx?familyid=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en
PsTools for remote agent deployment
http://www.microsoft.com/technet/sysinternals/Utilities/PsTools.mspx
Helix for a trusted toolkit http://e-fense.com/helix/

Introduction

MANDIANT Red Curtain (MRC) is free software for Incident Responders that takes analysis of malware to a different level; beyond expected norms you might say. MRC examines executable files to establish how suspicious they are based on a set of criteria, including review of multiple aspects of an executable for things such as entropy (more on this below), indications of packing, compiler and packing signatures, the presence of digital signatures, and other characteristics used to generate a threat "score." This score is then used to determine whether files are worthy of further investigation.[xvii]

MRC includes an analysis engine and a data presentation layer. The engine reads in the file to be analyzed, using the data within the file to calculate the Shannon Entropy across a series of overlapping windows of file segments. The engine also reviews additional aspects of the file, such as the permissions associated with various sections of the file, and whether or not the file has a valid, trusted digital signature applied to it. This data is then fed to the presentation layer, which organizes it for display to the user. It also takes the various elements of analysis generated by the analysis engine and calculates an aggregate threat score based on that data.

MRC’s road map includes incremental improvements such as refining the threat scoring algorithms, usability improvements, and perhaps the potential for expanding the criteria investigated. Mandiant is happy to receive community feedback to help set the direction of their next release.

MRC techniques are inherent in a commercial product, as part of the IR/acquisition features in Mandiant Intelligent Response.

MRC can be used under a free license but there are restrictions on reverse engineering or extending the product and resale. Mandiant has considered some open source models for some of their work, but haven't yet taken that direction.[xviii]

Entropy

Analysis of entropy, the measure of disorder and randomness, as it relates to malware, is a focus seemingly unique to MRC. Malware often makes use of encrypted, compressed, or obfuscated (depending on the method of obfuscation) data, and as such, its entropy tends to be higher than that of "structured" data, such as user-generated documents and well known computer programs. MRC approaches the identification of these attributes as follows, using Shannon Entropy[xix]:

1. A file is opened and the bytes read in to calculate a global entropy value for the entire file.

2. MRC then divides the file into overlapping samples and calculates the entropy across them. For arguments sake, assume a file of size X is divided into n samples of size Y.

3. The mean and standard deviation of all entropy values from all samples is calculated. The overall entropy for the input file is derived by taking the mean and adding one standard deviation to it. This value is referred to as the Sample Source Entropy.

4. Sample Source Entropy and Global Entropy are compared to a threshold. This threshold is an empirically derived value between 0 and 1. If either entropy value is greater than the threshold, the data block is determined to be entropic, and therefore potentially interesting.[xx]

Simplifying, entropic theory is applicable across many scientific practices, not just computer science, and is best explained by Dr. Thomas Schneider in *Information Is Not Entropy, Information Is Not Uncertainty!*, as it pertains to biology.

"Shannon called his measure not only the entropy but also the "uncertainty". I prefer this term because it does not have physical units associated with it. If you correlate information with uncertainty, then you get into deep trouble. Suppose that *information ~ uncertainty*, but since they have almost identical formulae, *uncertainty ~ physical entropy*, so *information ~ physical entropy* BUT as a system gets more random, its entropy goes up, *randomness ~ physical entropy*, so *information ~ physical randomness*

How could that be? Information is the very opposite of randomness!

The confusion comes from neglecting to do a subtraction:

Information is always a measure of the decrease of uncertainty at a receiver (or molecular machine)."[xxi]

If you subscribe to the claim that "information is always a measure of the decrease of uncertainty" you will not only grasp the concept that drives MRC's methodology, but one of the underlying fundamentals of malware research, or for all intents and purposes, incident handling in general. Eliminate uncertainty and you will be more readily able to build an effective response.

Installation

MRC installation is point and click so long as .NET 2.0 is already installed, but if you don't have on board, the installer will assist in its installation.

Usage

I typically use MRC in two situations. The first is as part of my live response toolkit for analysis of suspect hosts. The second is as part of my malware research sandbox, installed on Windows virtual machines destined for intentional infection with a variety of malware. As with any malware analysis under sandbox conditions, ensure you are operating in confirmed isolation where you'll do no harm to production or critical systems.

If reviewing live suspect hosts, there are some recommended steps to include as part of your procedure. If we assume prescribed methodology remember your goals include steps to:

- Identify & Analyze
- Contain
- Eradicate
- Recover
- Prevent

Incident handlers aren't likely to benefit from the same time allotment that may be afforded forensic investigators, given that information must be acquired quickly in order to establish an enterprise response. Tools like MRC provide ample assistance in that endeavor.

MRC can be used directly on the suspect host, but remember the .NET 2.0 framework must be installed.

Assuming you have the appropriate permissions to do so, I suggest running the MRC agent on the suspect host remotely, and analyzing its output on your workstation. Building the agent package is very simple. *File* -> *New* -> *Deploy Scanning Agent* will prepare the files you need to copy to the suspect host you're investigating.

A quick tip to consider as part of your incident response repertoire: only rely on trusted tools. If a system has been compromised, what guarantees do you have that it hasn’t been rooted or that common system executables haven’t been replaced? This is most easily overcome via reliance on a trusted toolkit like the Helix distribution.

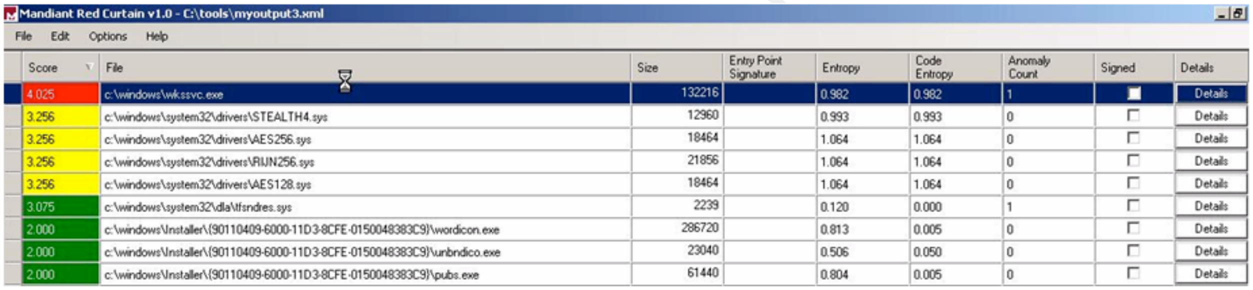
To deploy and execute the scanning with a trusted cmd.exe from your Helix distribution, make use of PsExec from SysInternals and do as follows:

1. Create scanning agent files with MRC.
2. Copy scanning agent files to suspect host.
3. Share your local CD drive as **cdrom**.
4. *psexec -u <admin acct> -p <password> \\<victim host ip> net use x: \\ <localhost ip\cdrom>*
5. *psexec –w x: \IR\xp -u <admin acct> -p <password> \\<victim host ip> x: \IR\xp\cmd.exe*
6. Now on victim host, issue *MRCAgent.exe epcompilersigs.dat eppackersigs.dat roamingsigs -r c:\windows output.xml*
7. Copy output.xml back to your workstation and open output.xml in the MRC console.

I make an assumption in my execution of MRCAgent, specifically the likely location of a malicious file on a suspect Windows host. Scan C:\WINDOWS if you want to cover the vast majority of probable locations and not risk missing anything, but you’ll note that the scan time is a lot longer than if you specified just C:\WINDOWS\system32 or C:\WINDOWS\system (common playgrounds for evil).

Analysis

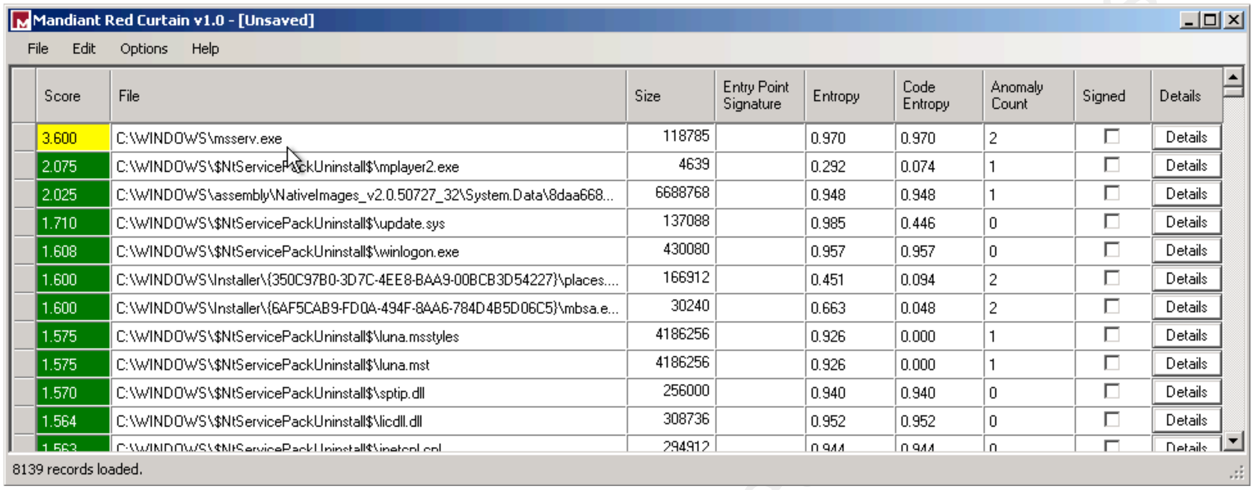
While, for the sake of this research, I’ve been working with a known malicious executable, I’ll treat out use of MRC as if I am seeking the culprit with no prior knowledge, as would be typical of an on-scene incident handler. Such is a scene where MRC's benefits really come to play. In order to exhibit what one might consider a "no-brainer" courtesy of MRC, the first example (non-Storm) shows an immediate and obvious response, where the findings are clearly delineated by a high entropy score for wkssvc.exe. Thanks to instant gratification from MRC, I grabbed wkssvc.exe out of C:\WINDOWS, fed it to Virustotal, and quickly determined that the suspect host had an SDBot variant onboard.



Score	File	Size	Entry Point Signature	Entropy	Code Entropy	Anomaly Count	Signed	Details
4.025	c:\windows\wkssvc.exe	132216		0.982	0.982	1	□	Details
3.256	c:\windows\system32\drivers\STEALTH4.sys	12960		0.993	0.993	0	□	Details
3.256	c:\windows\system32\drivers\AES256.sys	18464		1.064	1.064	0	□	Details
3.256	c:\windows\system32\drivers\RIJN256.sys	21856		1.064	1.064	0	□	Details
3.256	c:\windows\system32\drivers\AES128.sys	18464		1.064	1.064	0	□	Details
3.075	c:\windows\system32\dla\tfsndres.sys	2239		0.120	0.000	1	□	Details
2.000	c:\windows\Installer\{90110409-6000-11D3-8CFE-0150048383C9}\wordicon.exe	286720		0.813	0.005	0	□	Details
2.000	c:\windows\Installer\{90110409-6000-11D3-8CFE-0150048383C9}\unbndico.exe	23040		0.506	0.050	0	□	Details
2.000	c:\windows\Installer\{90110409-6000-11D3-8CFE-0150048383C9}\pubs.exe	61440		0.804	0.005	0	□	Details

Figure 17: wkssvs.exe stands out

However, the results from MRC output may not be as obvious as those seen in Figure 5, but paying close attention to details will still provide you with invaluable feedback if properly interpreted. Consider Figure 6.

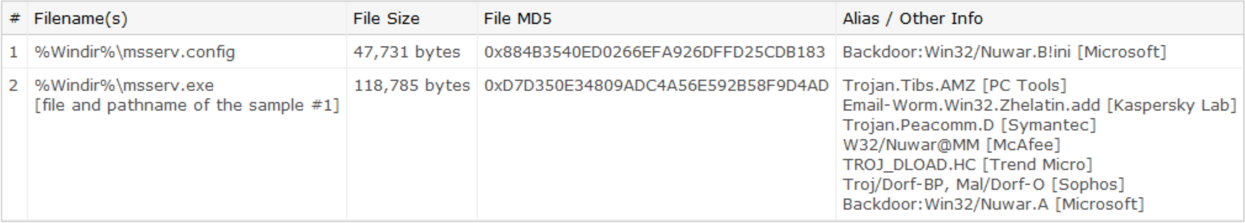


Score	File	Size	Entry Point Signature	Entropy	Code Entropy	Anomaly Count	Signed	Details
3.600	C:\WINDOWS\msserv.exe	118785		0.970	0.970	2	□	Details
2.075	C:\WINDOWS\$NtServicePackUninstall$\mplayer2.exe	4639		0.292	0.074	1	□	Details
2.025	C:\WINDOWS\assembly\NativeImages_v2.0.50727_32\System.Data\8daa668...	6688768		0.948	0.948	1	□	Details
1.710	C:\WINDOWS\$NtServicePackUninstall$\update.sys	137088		0.985	0.446	0	□	Details
1.608	C:\WINDOWS\$NtServicePackUninstall$\winlogon.exe	430080		0.957	0.957	0	□	Details
1.600	C:\WINDOWS\Installer\{350C97B0-3D7C-4EE8-BAA9-00BCB3D54227}\places....	166912		0.451	0.094	2	□	Details
1.600	C:\WINDOWS\Installer\{6AF5CAB9-FD0A-494F-8AA6-784D4B5D06C5}\mbsa.e...	30240		0.663	0.048	2	□	Details
1.575	C:\WINDOWS\$NtServicePackUninstall$\luna.msstyles	4186256		0.926	0.000	1	□	Details
1.575	C:\WINDOWS\$NtServicePackUninstall$\luna.mst	4186256		0.926	0.000	1	□	Details
1.570	C:\WINDOWS\$NtServicePackUninstall$\sptip.dll	256000		0.940	0.940	0	□	Details
1.564	C:\WINDOWS\$NtServicePackUninstall$\licdll.dll	308736		0.952	0.952	0	□	Details
1.563	C:\WINDOWS\$NtServicePackUninstall$\inetcpl.cpl	294912		0.944	0.944	0	□	Details

8139 records loaded.

Figure 18: less obvious, but no less evil

A pretty red alert with a high score didn't pop right to the top of my console, just a yellow, medium score. But, when sorting by Anomaly Count, msserv.exe (installed by fireworks.exe) stood out with a score of 2 and higher entropy counts in both categories. Msserv.exe, found in C:\WINDOWS showed two anomalies, including checksum_is_zero, contains_eof_data. There was no odd Entry Point Signature reference such as Borland Delphi, as opposed to MS Visual C++, but look for that detail as you utilize MRC. Finally, checking MAC times on msserv.exe (and the fact that I know it has no business in *C:\WINDOWS*) led me to a super-sleuth conclusion…msserv.exe must be malware! As I prove on a daily basis, one needn't be a genius to go bug hunting and find your quarry. ThreatExpert confirmed my deduction and advised that msserv.exe is indeed Storm.



#	Filename(s)	File Size	File MD5	Alias / Other Info
1	%Windir%\msserv.config	47,731 bytes	0x884B3540ED0266EFA926DFFD25CDB183	Backdoor:Win32/Nuwar.B!ini [Microsoft]
2	%Windir%\msserv.exe [file and pathname of the sample #1]	118,785 bytes	0xD7D350E34809ADC4A56E592B58F9D4AD	Trojan.Tibs.AMZ [PC Tools] Email-Worm.Win32.Zhelatin.add [Kaspersky Lab] Trojan.Peacomm.D [Symantec] W32/Nuwar@MM [McAfee] TROJ_DLOAD.HC [Trend Micro] Troj/Dorf-BP, Mal/Dorf-O [Sophos] Backdoor:Win32/Nuwar.A [Microsoft]

Figure 19: Storm confirmed

In all seriousness, MRC directly contributed to identifying msserv.exe as worthy of further investigation and likely halved my response and analysis time in this particular investigation.

Benefits and Drawbacks

The addition of software that readily aids in the identification of malware can only be seen as beneficial to your toolkit. MRC is just such an application.

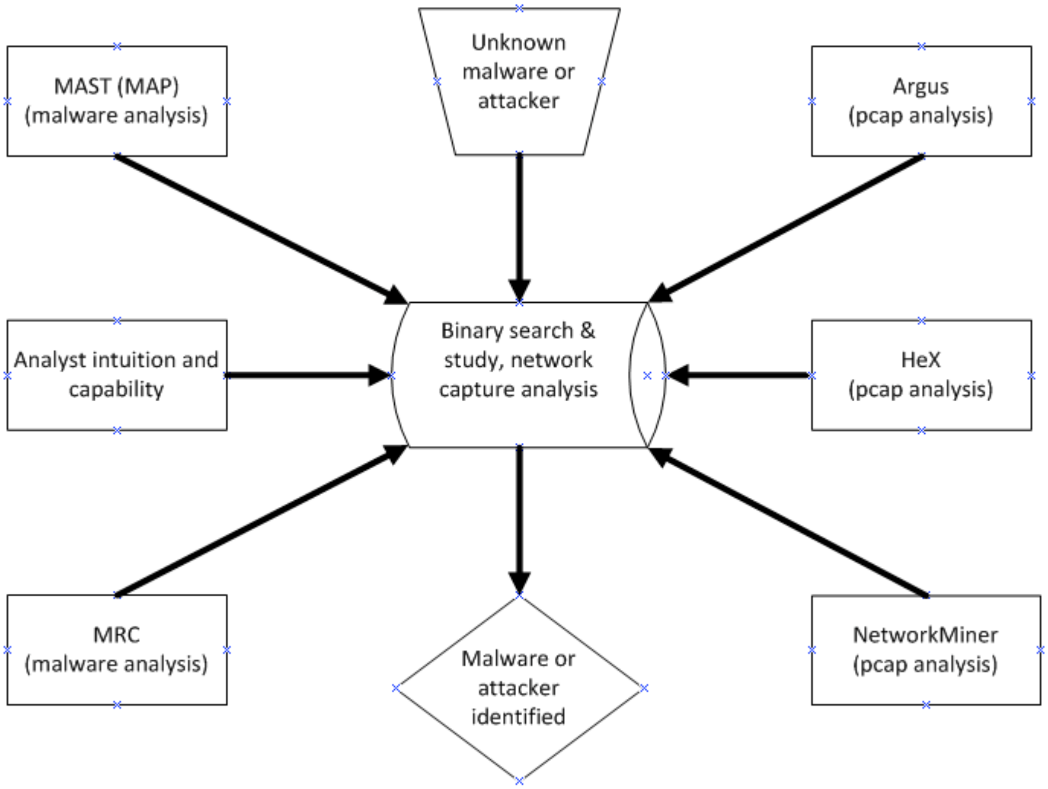
Results are not always obvious; remember to sort by Entry Point Signatures, Anomaly counts, and Entropy until you locate a candidate for further investigation. I have found malware that received an entropic green light, but had enough additional odd characteristics to stand out regardless.

Conclusion

Remember our discussion of trusted tools, and conduct malware analysis in an isolated environment as often as possible. Additionally, enhance, refine, and practice with your incident response toolkit. Your real response will be all the more successful when the time comes. The addition of MRC to your toolkit will further guarantee that success.

4 Summary

I've discussed five unique offerings, each with specific nuances and capabilities, but all valuable to the incident handler in delving deeper into discovery and investigation. I firmly adhere to the belief that incident analysis is best conducted following a spoke and wheel model. In this scenario we sought malicious code via binary search and study, as well as network capture analysis.


MAST (MAP) (malware analysis)
Unknown malware or attacker
Argus (pcap analysis)
Analyst intuition and capability
Binary search & study, network capture analysis
HeX (pcap analysis)
MRC (malware analysis)
Malware or attacker identified
NetworkMiner (pcap analysis)

Through the thoughtful addition of investigation expanding tools, the incident handler may better fulfill their duties, reach a precise conclusion, inclusive of extensive discovery data, allowing their respective organizations to respond and remediate effectively.

5 References

[i] Pg. 236, The Tao of Network Security Monitoring, Richard Bejtlich, Addison-Wesley, 2005

[ii] Argus-NSMWiki, http://www.vorant.com/nsmwiki/index.php?title=Argus

[iii] https://wiki.internet2.edu/confluence/display/secguide/Glossary

[iv] http://qosient.com/argus/how-to.htm#8

[v] http://qosient.com/argus/faq.htm#11.1

[vi] http://secviz.org/?q=node/74

[vii] http://sunbeltblog.blogspot.com/2008/09/dancing-in-streets-almost-intercage.html

[viii] http://www.brendangregg.com/chaosreader.html

[ix] http://writequit.org/projects/nsm-console/

[x] http://taosecurity.blogspot.com/2006/09/port-independent-protocol.html

[xi] http://networkminer.wiki.sourceforge.net/NetworkMiner

[xii] http://sectools.org

[xiii] http://labs.idefense.com/files/labs/releases/previews/SysAnalyzer/

[xiv] http://labs.idefense.com/files/labs/releases/previews/SysAnalyzer/

[xv] http://labs.idefense.com/files/labs/releases/previews/map/

[xvi] http://labs.idefense.com/files/labs/releases/previews/multipot/index.html

[xvii] http://www.mandiant.com/mrc

[xviii] Dave Merkel, MANDIANT

[xix] *A Mathematical Theory of Communication*, C.E. Shannon, http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

[xx] Mandiant Red Curtain User Guide, section 2.2 The Entropy of Evil

[xxi] Information Is Not Entropy, Information Is Not Uncertainty!, Dr. Thomas D. Schneider, http://www.lecb.ncifcrf.gov/~toms/information.is.not.uncertainty.html