Packet Analysis with the HeX System

By Russ McRee – ISSA member, Puget Sound (Seattle), WA, USA chapter

Prerequisites

HeX LiveCD distribution requires only a system capable of booting from an optical drive.

Likewise, to utilize HeX liveUSB, the system must be able to boot from USB.

HeX Virtual Appliance runs on VMware, Qemu, Parallels, and Virtualbox.

Two nics may be beneficial, depending on your usage. 512MB RAM minimum is recommended.

Similar Projects

Knoppix-NSM[1]

NUbuntu[2]

NSM practitioners and packet analysts rejoice, the HeX System[3] is a project with your best interests at heart. We have recently discussed auditing network activity with Argus, the real time flow monitor. Argus, while an excellent stand alone tool, is considered, in certain circles, to be part of a larger family of tools useful for NSM, or Network Security Monitoring. Again, if you have not already made them part of your library, add Richard Bejtlich's books as required reading on all things NSM. You have likely read numerous discussions on Snort and Wireshark, and hopefully Sguil, and perhaps heard of 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. However, the subject of this month's topic hopes to make the process a bit easier on you by gathering the above mentioned, and many other invaluable NSM tools, in one offering. Enter the HeX System, 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.

Additionally, Fluxbox is the main window manager, used for its minimalistic interface and simplicity in order to free computer resources for better network traffic analysis capacity.

HeX is best utilized for offline packet analysis, where captures can be pulled down to the local system and dissected with the likes of Sguil or the NSM-Console.

This project, under the direction of C.S. Lee, has an extensive roadmap and a wide range of influences.

Pending releases will offer:

- Further analyst assistance
  - Browser bookmarks (whois, domain queries)
  - Security rss feeds
- More network traffic analysis scripts
- Analysis result output in html
- Unionfs integration

This is a project under constant development, and as with many we discuss in toolsmith, the development team invites feedback and contribution.

Installation

There are no real installation challenges with the HeX System, 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).

Remember, the HeX System is intended to be portable, so the Virtual Appliance and liveUSB versions are at your disposal, but I used the LiveCD version for this month's discussion.

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

[1] http://securixlive.com/knoppix-nsm/index.php.

[2] http://nubuntu.org.

[3] http://rawpacket.org.

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, the best way to run HeX is while listening to a network via a tap or SPAN port; switched traffic will yield very limited results.

Usage

Rather than make this month's discussion one of tools you may be familiar with, I would like to take you down the road less traveled, but no less useful.

NSM-Console

The NSM-Console, written specifically for HeX by Matthew Lee Hinman, is 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 stable version included in version 1.0.2 of HeX includes 12 modules, but you can quickly grab the 0.3-DEVEL version (at the time of writing) at the project website.[4] Expect more modules in the near future. Lee has done a great screencast which I highly recommend viewing; you will find links to it on the project site as well. The HeX Sytem project roadmap for the 2.0 release includes the default integration of NSM-Console as the standard shell.

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 will 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.

We will leave everything default for our example and simply set some modules as active; then run. Rather than toggle a whole category, I chose to:

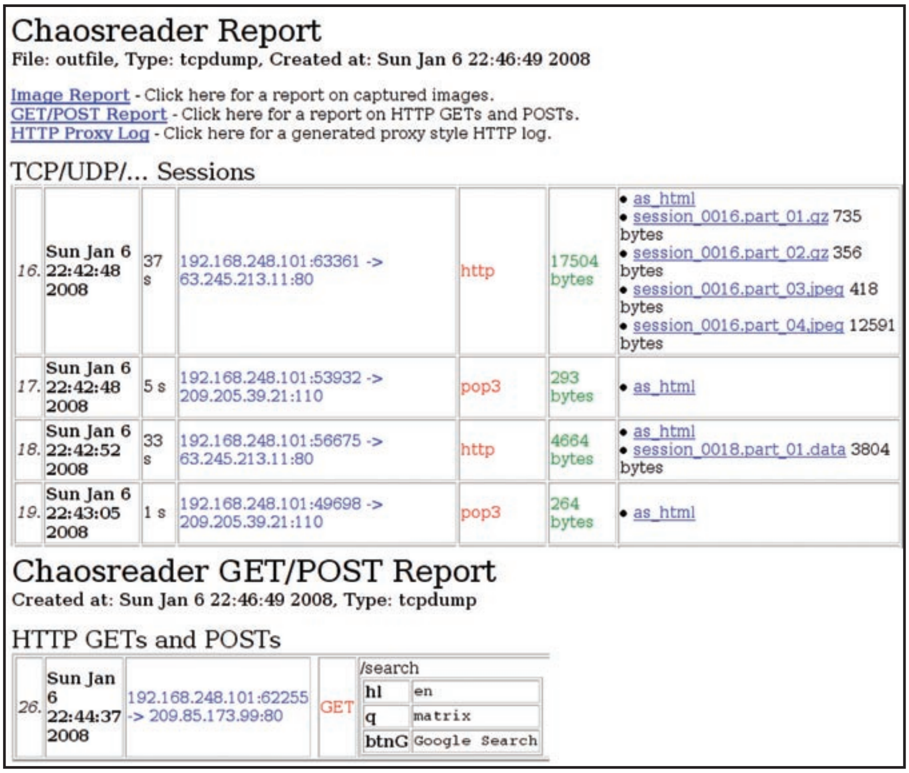
```
nsm> file /home/analyzt/toolsmith.pcap (specify the
source file)
nsm> options (set global options)
nsm> output /home/analyzt (define output directory)
nsm> toggle tcpdstat (extract tcp statistics about a
pcap)
nsm> toggle capinfos (print information about a pcap)
nsm>toggle  hash (hash a pcap)
nsm> run (you get the point)
```

The net results from our run are:

```
/home/analyzt/hash
MD5 (/home/analyzt/toolsmith.pcap) =
bfd6c78a0b6d9f41d4496ea1ed2d5d52
SHA256 (/home/analyzt/toolsmith.pcap) = e619ba3d83471b0e
3bf4878a7219bf3237c48cf4f29d6634e30b3b07d4b83e6f
 /home/analyzt/tcpdtstat
toolsmith.pcap.tcpdstat (abbreviated output)
DumpFile:  /home/analyzt/toolsmith.pcap
FileSize: 1.15MB
Id: 200801072234
StartTime: Mon Jan  7 22:34:59 2008
EndTime:   Mon Jan  7 22:36:55 2008
TotalTime: 115.45 seconds
TotalCapSize: 1.12MB  CapLen: 1514 bytes
# of packets: 2143 (1.12MB)
AvgRate: 108.84Kbps  stddev:197.69K
### Protocol Breakdown ###
<<<<
protocol    packets       bytes            bytes/pkt
-----------------------------------------------------
[0] total   2143 (100.00%)  1173519 (100.00%)  547.61
[1] ip      2143 (100.00%)  1173519 (100.00%)  547.61
[2] tcp     2007 (93.65%)   1153160 (98.27%)   574.57
[3] http(s) 1036 (48.34%)   1000215 (85.23%)   965.46
[3] http(c) 971  (45.31%)   152945  (13.03%)   157.51
[2] udp     135  (  6.30%)  20299  (  1.73%)   150.36
[3] dns     130  (  6.07%)  18889  (  1.61%)   145.30
[3] mcast   1  (  0.05%)    82  (  0.01%)      82.00
[3] other   4  (  0.19%)    1328  (  0.11%)    332.00
[2] igmp    1  (  0.05%)    60  (  0.01%)      60.00
>>>>
/home/analyzt/capinfos
toolsmith.pcap.capinfos
File name: /home/analyzt/toolsmith.pcap
File type: Wireshark/tcpdump/... - libpcap
Number of packets: 2143
File size: 1207831 bytes
Data size: 1173519 bytes
Capture duration: 115.445612 seconds
Start time: Mon Jan  7 22:34:59 2008
End time: Mon Jan  7 22:36:55 2008
Data rate: 10165.12 bytes/s
Data rate: 81320.99 bits/s
Average packet size: 547.61 bytes
File name: /home/analyzt/toolsmith.pcap
File type: Wireshark/tcpdump/... - libpcap
Number of packets: 2143
File size: 1207831 bytes
Data size: 1173519 bytes
Capture duration: 115.445612 seconds
Start time: Mon Jan  7 22:34:59 2008
End time: Mon Jan  7 22:36:55 2008
Data rate: 10165.12 bytes/s
Data rate: 81320.99 bits/s
Average packet size: 547.61 bytes
```

Now imagine using NSM-Console against numerous pcaps, with multiple modules selected. To quote Peyton Manning,

[4] http://thnetos.wordpress.com/nsm-console.



Chaosreader Report

File: outfile, Type: tcpdump, Created at: Sun Jan 6 22:46:49 2008

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

TCP/UDP/... Sessions

16.	Sun Jan 6 22:42:48 2008	37 s	192.168.248.101:63361 -> 63.245.213.11:80	http	17504 bytes	• as_html • session_0016.part_01.gz 735 bytes • session_0016.part_02.gz 356 bytes • session_0016.part_03.jpeg 418 bytes • session_0016.part_04.jpeg 12591 bytes
17.	Sun Jan 6 22:42:48 2008	5 s	192.168.248.101:53932 -> 209.205.39.21:110	pop3	293 bytes	• as_html
18.	Sun Jan 6 22:42:52 2008	33 s	192.168.248.101:56675 -> 63.245.213.11:80	http	4664 bytes	• as_html • session_0018.part_01.data 3804 bytes
19.	Sun Jan 6 22:43:05 2008	1 s	192.168.248.101:49698 -> 209.205.39.21:110	pop3	264 bytes	• as_html

Chaosreader GET/POST Report

Created at: Sun Jan 6 22:46:49 2008, Type: tcpdump

HTTP GETs and POSTs

26.	Sun Jan 6 22:44:37 2008	192.168.248.101:62255 -> 209.85.173.99:80	GET	/search hl en q matrix btnG Google Search

Figure 1 – Chaosreader reports

"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. Kudos, Lee!

Chaosreader

In the NBF-Toolkit you will discover Chaosreader (also a module in NSM-Console), 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.[5]

Run `sudo wireshark`, select the available interface, and leave it running for a few minutes, then save the capture file.

Next, run `chaosreader <capture file>`, then browse the resulting `index.html` file.

The examples in Figure 2 show a mere pittance of the possible output, what you do not see is the all the possible session data, or the IP, TCP port, UDP port, IP protocol, and Ethernet type counts.

Darkstat

So as not to leave real-time monitoring out entirely, in the *NSM-Toolkit > Statistical* category we find Darkstat, a handy little network monitor. Darkstat is an ntop-like packet sniffer that runs as a background process and gathers all sorts of statistics about network usage, serving them over HTTP.

Darkstat features include:

- Traffic graphs, reports per host, reports each port per host
- Embedded web-server with deflate compression
- Asynchronous reverse DNS resolution using a child process
- Small, single-threaded, efficient, and uncomplicated[6]

Run `sudo darkstat –i bge0 –p 8080`, using the appropriate interface nomenclature for the `–i` parameter (subject to change per system). The resulting output, before it drops into a background process, will give you a sense of how capable the tool's features really are.

```
darkstat 3.0.619 using libpcap 2.4
darkstat (01520): starting up
darkstat (01520): daemonizing to run in the background!
darkstat (01520): parent waiting
darkstat (01521): DNS child has PID 1522
darkstat (01521): caplen is 54
darkstat (01521): capturing in promiscuous mode
darkstat (01521): filtered out BPF writes
darkstat (01521): locked down BPF for security
darkstat (01521): listening on 0.0.0.0:8080
darkstat (01521): loaded 131 protos
darkstat (01521): loaded 1006 tcp and 971 udp servs,
from total 1982
darkstat (01521): local_ip update(bge0) = 192.168.248.101
darkstat (01521): entering main loop
darkstat (01520): parent done reading, calling waitpid
darkstat (01520): waitpid ret 0, status is -215823172
```

You can then view the output by browsing localhost:8080 with Firefox.

As the Darkstat website indicates, "Although it is possible to configure/complicate things further… darkstat will just work without much trouble on your part."

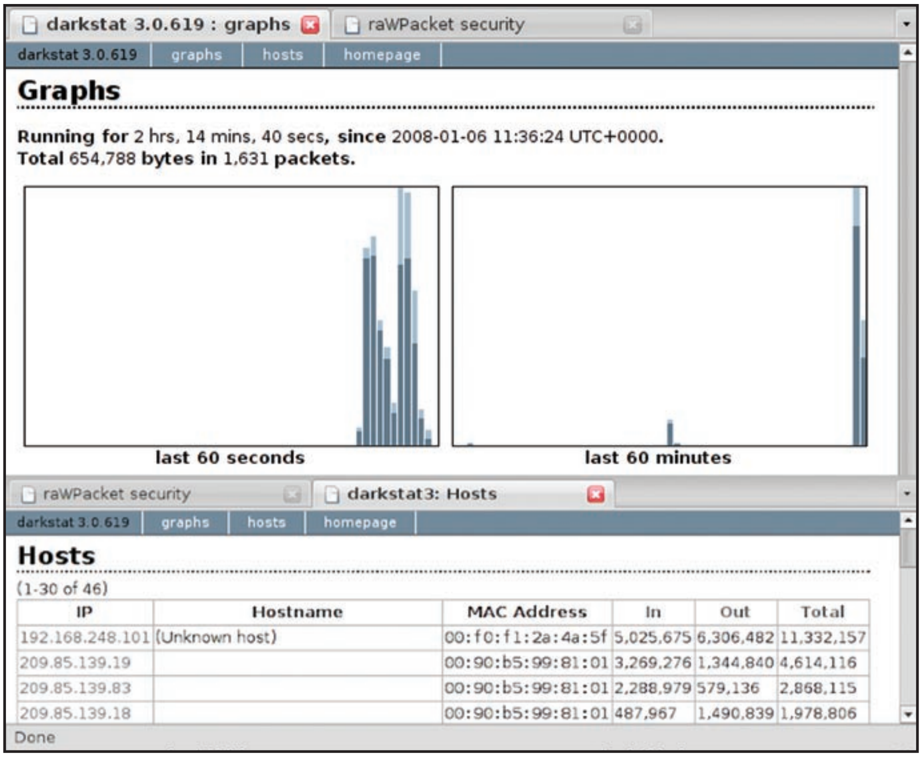
Benefits and Drawbacks

The HeX project is a young one, and therefore functionally immature in places, but it is extremely well intended and headed in the right direction with a development team and leadership working in earnest to improve it every opportunity. The roadmap satisfies any criticisms I may have, and I look forward to what the future holds for the Hex project.

For users with no *nix skills, this distribution may present some challenges. While purists will tout the strength of

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

[6] http://dmr.ath.cx/net/darkstat.



Graphs

Running for 2 hrs, 14 mins, 40 secs, **since** 2008-01-06 11:36:24 UTC+0000.
Total 654,788 **bytes in** 1,631 **packets.**

Hosts

(1-30 of 46)

IP	Hostname	MAC Address	In	Out	Total
192.168.248.101	(Unknown host)	00:f0:f1:2a:4a:5f	5,025,675	6,306,482	11,332,157
209.85.139.19		00:90:b5:99:81:01	3,269,276	1,344,840	4,614,116
209.85.139.83		00:90:b5:99:81:01	2,288,979	579,136	2,868,115
209.85.139.18		00:90:b5:99:81:01	487,967	1,490,839	1,978,806

Figure 2 – Darkstat output

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.

In Conclusion

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

Required online reading for NSM practitioners includes:

- taosecurity.blogspot.com
- geek00l.blogspot.com
- www.vorant.com/nsmwiki/Main_Page
- http://thnetos.wordpress.com

Cheers…until next month.

Acknowledgments

C.S. Lee, Kevin Foo, J.J. Cummings, Mathew Lee Hinman, and the Hex Development Team for their hard work and dedication.

About the Author

Russ McRee, GCIH, GCFA, CISSP, is a security analyst working in the Seattle area. As an advocate of a holistic approach to information security, Russ' website is holisticinfosec.org. Contact him at russ@holisticinfosec.org.