Nothing but Network Forensics

Submission for the DFRWS 2009 Forensic Challenge

Author: Erik Hjelmvik <erik.hjelmvik[at]gmail.com>

Date: 07 June 2009

Used Software:

NetworkMiner 0.88
http://networkminer.sourceforge.net/

SPID
Algorithm

SPID Algorithm Proof-of-Concept 0.4
http://sourceforge.net/projects/spid/

Wireshark 0.99.8
http://www.wireshark.org/

1 Introduction

The 2009 forensic challenge of the Digital Forensics Research Conference provides data to be analyzed in the form of data at rest (two filesystem images), data in use (one physical memory dump) and data in transit (three network traces). This submission is, however, based only on the analysis of the captured network traffic. The goal is to show just how much relevant information that can easily be extracted from network captures in general, and by using NetworkMiner in particular.

Many of the questions in the challenge are left unanswered in this submission, which is fine considering the limited time I have spent on this submission. The analysis results provided herein can easily be extracted within hours with the listed tools without having expertise knowledge.

1.1 About NetworkMiner

NetworkMiner is a network forensic analysis tool that I have developed in order to facilitate the task of performing network forensic investigations as well as conducting incident response (Hjelmvik, 2008).

NetworkMiner is designed to collect data about hosts on a network rather than to collect data regarding the traffic on the network. It has a graphical user interface where the main view is host centric (information grouped per host) rather than packet centric (information showed as a list of packets/frames).

One of the most appreciated functions in NetworkMiner is the ability to easily extract files from captured network traffic in protocols such as HTTP, FTP, TFTP and SMB. NetworkMiner actually reassembles files to disk on the fly as it parses a PCAP file. A lot of other useful information like user credentials, transmitted parameters, operating systems, hostnames, server banners etcetera can also be extracted from network traffic with NetworkMiner. All of this is of course performed fully passive, so that no traffic is emitted to the network while performing the network forensic analysis.

NetworkMiner is available at SourceForge[1] as open source under a GPL license.

1.2 About the SPID Algorithm

The Statistical Protocol IDentification (SPID) Algorithm was designed by me with the purpose of identifying the application layer protocol in TCP sessions without relying on port numbers. The SPID algorithm performs protocol identification based on simple statistical measurements of various protocol attributes. These attributes can be defined by all sorts of packet and flow data, ranging from traditional statistical flow features to application level data measurements, such as byte frequencies and offsets for common byte-values (Hjelmvik and John, 2009).

I have made a proof-of-concept implementation of the SPID algorithm available on SourceForge[2] as open source under a GPL license.

---

[1] http://sourceforge.net/projects/networkminer

[2] http://sourceforge.net/projects/spid

1.3 Challenge Details

The first network trace is based on early surveillance of the suspect; this network trace is named nssal-capture-1.pcap.bz2. A second trace, nssal-capture-2.pcap.bz2, contains communication between "nssal" and another machine located at Johns Hopkins University. The network administrator in the lab at Johns Hopkins identified the machine as another PS3. This administrator regularly monitors communication and was able to provide a third network trace, jhuisi-capture-1.pcap.bz2, which contains traffic transmitted between the "nssal" PS3 and the PS3 in the Johns Hopkins lab.

1.4 Challenge Questions

- What relevant user activity can be reconstructed from the available forensic data and what does it show?
- Is there evidence of inappropriate or suspicious activity on the system?
- Is there evidence of collaboration with an outside party?
    - If so, what can be determined about the identity of the outside party?
    - How was any collaboration conducted?
- Is there evidence that illicit data (specifically, Mardi Gras images) was exchanged?
    - If so, what can be determined about that data and the manner of transfer?
- What data (if any) was provided by the Johns Hopkins PS3?
- The suspect claims that he was not responsible for any transfer of data. What evidence do you have to show that remote, unauthorized access to the system might have occurred, and does this evidence exonerate the suspect?

2 Results

A simple network diagram of a few involved hosts is shown in Figure 1. The diagram is built exclusively from information extracted with NetworkMiner from the provided packet capture files.

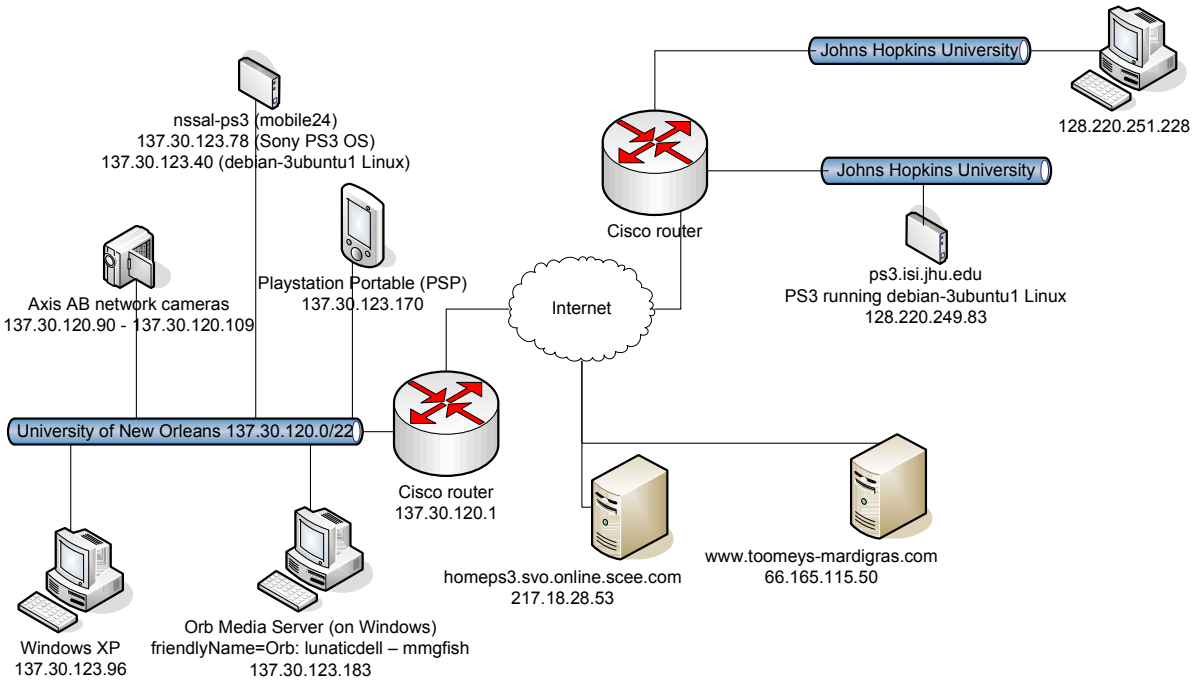

Figure 1 Simplified network diagram

The two most important hosts in the diagram are:

- nssal-ps3 (137.30.123.78 and 137.30.123.40)
- ps3.isi.jhu.edu (128.220.249.83)

NetworkMiner successfully identified the operating systems on the two PS3 devises (137.30.123.40 and 128.220.249.83) as Linux. There are, however, no operating system fingerprints available in NetworkMiner for the "Cell OS", which is the default operating system for Playstation 3 devices.

The web browser User-Agent banners from the PS3 machines do, however, reveal the Cell OS. Some User-Agent strings retrieved from nssal-ps3 are:

- Mozilla/5.0 (PLAYSTATION 3; 1.00)
- PS3FriendImUtil/2.6.0-000 libhttp/2.6.0-000 (CellOS)
- PS3Application libhttp/2.6.0-000 (CellOS)

2.1 User Activity

What relevant user activity can be reconstructed from the available forensic data and what does it show?

A good overview of the performed activities can be achieved by looking at the timeline during the time of the network traffic captures. All timestamps in this submission are provided in Central European Time (CET), even though that clearly not is the time zone

where the activities took place. The reference time used for all timestamps is that of the network monitoring machine used in the nssal packet captures[3].

Events from March 5 2009 (nssal-capture-1.pcap):

Timestamp	Event Description
21:19:38	User at nssal-ps3 searches for "mardi gras" at www.google.com
21:19:43	User at nssal-ps3 visits www.mardigrasday.com
21:21:05	User at nssal-ps3 visits mardigrasday.makesparties.com
21:22:10	User at nssal-ps3 visits www.mardigrasgalveston.com
21:22:36	User at nssal-ps3 searches for "mardi gras" at images.google.com
21:24:01	User at nssal-ps3 visits www.toomeys-mardigras.com
21:24:14	User at nssal-ps3 searches for "mardi gras pictures" at www.google.com
21:25:20	User at nssal-ps3 searches for "mardi gras pictures k00l" at www.google.com
21:32:54	User at nssal-ps3 searches for "mardi gras" at www.yahoo.com
21:38:07	User at nssal-ps3 logs in with the account name "nssal" and account ID "2692321" at homeps3.svo.online.scee.com (217.18.28.53)
21:38:28	User at nssal-ps3 retrieves her avatar image "Avatar-nssal.jpg" from homeps3.svo.online.scee.com

Table 1 Events from March 5 2009 between 21:18:52 and 21:42:53

Events from March 11 2009 (nssal-capture-2.pcap and jhuisi-capture-1.pcap):

Timestamp	Event Description
17:03:42	The user at nssal-ps3 logs in with the account name "nssal" and account ID "2692321" at homeps3.svo.online.scee.com (217.18.28.53)
17:04:16	The user at nssal-ps3 retrieves her avatar image "Avatar-nssal.jpg" from homeps3.svo.online.scee.com
17:05:36	The user at ps3.isi.jhu.edu (128.220.249.83) logs in with the account name "jhuisi" and account ID "2175478" at homeps3.svo.online.scee.com
17:06:02	The user at ps3.isi.jhu.edu retrieves his avatar image "Avatar-jhuisi.jpg" from homeps3.svo.online.scee.com
17:28:42	The ps3.isi.jhu.edu machine (running PS3 "Cell OS") is shut down
17:29:01	The nssal-ps3 (137.30.123.78) machine (running PS3 "Cell OS") is shut down
17:29:56	The ps3.isi.jhu.edu machine is rebooted, but this time running Ubuntu Linux
17:31:16	An SSH session is established from 128.220.251.228 (on the John Hopkins network) to ps3.isi.jhu.edu (128.220.249.83)
17:31:51	A second SSH session is established from 128.220.251.228 to ps3.isi.jhu.edu
17:42:06	The nssal-ps3 machine is booted with Ubuntu Linux on the new IP address 137.30.123.40

[3] The nssal monitoring machine is lagging 13 seconds behind the time given from ntp.ubuntu.com, the jhuisi monitoring machine on the other hand is 14 seconds ahead of the same NTP reference

17:42:15	An SSH session is established from nssal-ps3 to ps3.isi.jhu.edu
17:44:29	A second SSH session is established from nssal-ps3 to ps3.isi.jhu.edu[4]
17:44:42	A third SSH session is established from 128.220.251.228 to ps3.isi.jhu.edu
17:49:02	A third SSH session is established from nssal-ps3 to ps3.isi.jhu.edu
17:49:37	An SSH session is established from ps3.isi.jhu.edu to nssal-ps3 (opposite direction compared to previous SSH sessions)
17:51:57	A second SSH session is established from ps3.isi.jhu.edu to nssal-ps3[5]
17:56:06	A third SSH session is established from ps3.isi.jhu.edu to nssal-ps3[6]
18:02:23	A forth SSH session is established from ps3.isi.jhu.edu to nssal-ps3
18:09:59	A remote (unencrypted) shell is established from ps3.isi.jhu.edu to nssal-ps3 on TCP 45541. The user authenticates himself with username "jhuisi" and password "mac"[7]

Table 2 Events from March 11 2009 between 17:01:16 and 18:16:13

Image 1 Avatar-nssal.jpg

Image 2 Avatar-jhuisi.jpg

2.2 Suspicious Activity

Is there evidence of inappropriate or suspicious activity on the system?

None which I could easily be find with the tools used in this submission. The incoming remote shell session from ps3.isi.jhu.edu to nssal-ps3 on TCP 45541 is, however, slightly unusual.

---

[4] This session is not closed until 18:14:15

[5] Probably a file download: 4MB received in 7 seconds from nssal-ps3 to ps3.isi.jhu.edu

[6] Probably a file upload: 20kB sent in 3 seconds from ps3.isi.jhu.edu to nssal-ps3

[7] This backdoor behaves like "backd00r.c", which can be found here: http://www.packetstormsecurity.org/UNIX/penetration/rootkits/backd00r.c

2.3 Collaboration with an Outside Party

Is there evidence of collaboration with an outside party?

The two PS3 machines (nssal-ps3 and ps3.isi.jhu.edu) were both shut down to be rebooted with Ubuntu Linux at the same time (at approximately 17:29 on March 11). This would indicate some coordination between these two parties. There was no IP communication between these two machines prior to the rebooting of the PS3s.

2.3.1 Identity of the Outside Party

If so, what can be determined about the identity of the outside party?

The outside party uses the nickname "jhuisi", which is an acronym for "Johns Hopkins University Information Security Institute" (it can also be noted that nssal is an acronym for "Networking, Security, and Systems Administration Laboratory", which is a part of the University of New Orleans). The avatar image used by the jhuisi user is shown in Image 2 Avatar-jhuisi.jpg, which is a computer animated picture.

2.3.2 Collaboration Operation

How was any collaboration conducted?

After booting Linux a series of SSH sessions took place in both directions between these hosts. Several of the SSH sessions are expected to be file transfers due to the high bandwidth usage of the sessions. File transfers are expected to have been made in both directions between the two PS3 machines.

2.4 Mardi Gras Image Exchange

Is there evidence that illicit data (specifically, Mardi Gras images) was exchanged?

Several Mardi Gras images were viewed and/or downloaded by the user at nssal-ps3.

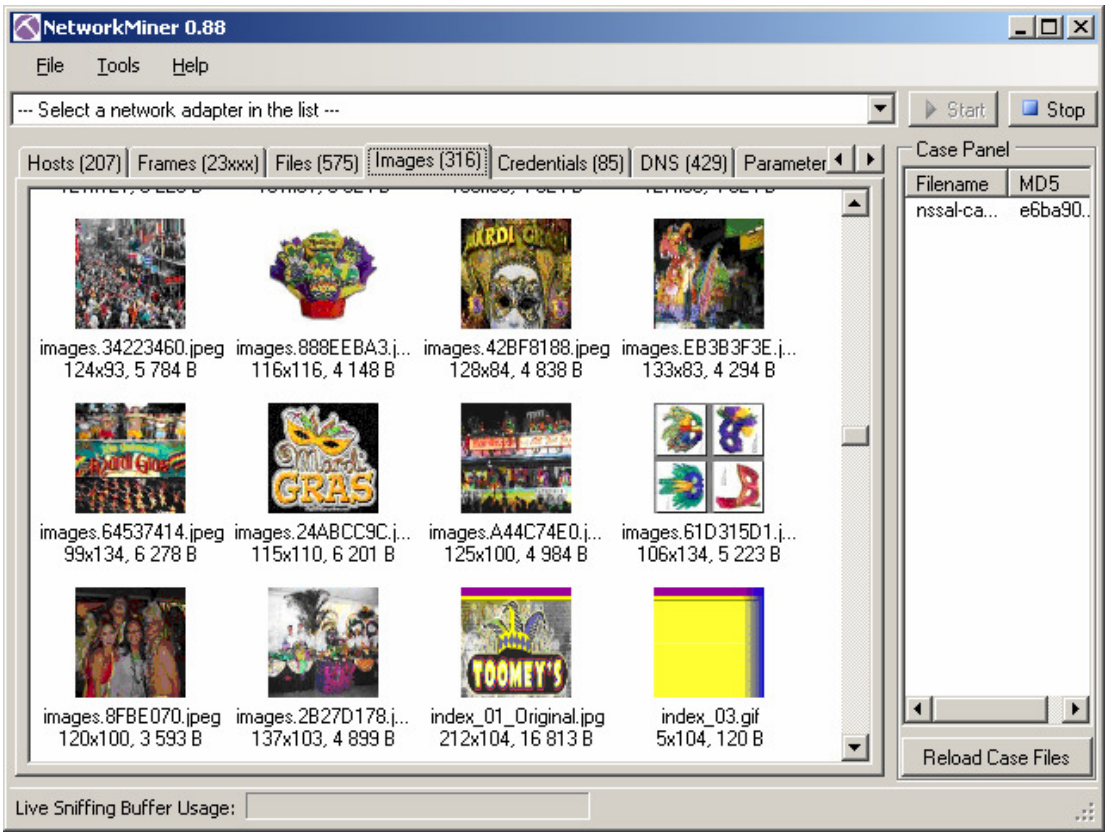

Image 3 NetworkMiner "Images" tab showing thumbnails of Mardi Gras images

No evidence of any Mardi Gras images being transferred to an outside party was discovered though.

The properties of the SSH connections between to ps3.isi.jhu.edu nssal-ps3 does, however, indicate that approximately 4MB of data was transferred from nssal-ps3 to ps3.isi.jhu.edu on March 11 17:51:57 by the user at ps3.isi.jhu.edu.

2.5 Data Provided by the John Hopkins PS3

What data (if any) was provided by the Johns Hopkins PS3?

SSH session properties of one of the SSH sessions established from ps3.isi.jhu.edu to nssal-ps3 indicate that approximately 20kB of data was uploaded to the nssal-ps3 machine. The contents of the suspected transfer are unknown since the transfer was encrypted.

2.6 Remote and Unauthorized Access

The suspect claims that he was not responsible for any transfer of data. What evidence do you have to show that remote, unauthorized access to the system might have occurred, and does this evidence exonerate the suspect?

There was one unencrypted incoming remote shell session to nssal-ps3 on TCP 45541[8] from ps3.isi.jhu.edu. The remote shell session behaved like "backd00r.c"[9], which is a bindshell backdoor.

[8] The port number 45541 is l33t speek for "assal", which is very close to "nssal"

The user at ps3.isi.jhu.edu authenticated himself with the words "jhuisi" and "mac" and then listed various data on the nssal-ps3 machine. The user at ps3.isi.jhu.edu did not attempt to remove, add or change any contents of the files on the nssal-ps3 machine while being connected through the backdoor.

No backdoor connection to the nssal-ps3 machine was active during the time of the assumed SSH file transfers between ps3.isi.jhu.edu and nssal-ps3, so the existence of this backdoor does not exonerate the suspect.

[9] Available at: http://www.packetstormsecurity.org/UNIX/penetration/rootkits/backd00r.c

3 Details

This chapter provides details on how the information in the previous chapter was retrieved from the packet capture files.

3.1 SPID Algorithm Proof-of-Concept Application

The SPID algorithm was used to determine the application layer protocols of the TCP sessions in the analyzed network capture files. The SPID Algorithm Proof-of-Concept-0.4[10] was used in this submission.

After launching the SPID PoC algorithm (which is a Microsoft Windows application based on the .NET framework) a PCAP file can be analyzed either by drag-and-dropping it onto SPID PoC, or by selecting "File > Open > Pcap File" from the menu.

The used SPID PoC version can only identify the following protocols:

- BitTorrent
- eDonkey
- FTP control
- HTTP
- IMAP
- IRC
- MSE (Message Stream Encryption, a.k.a. "Encrypted BitTorrent")
- MSN Messenger
- POP
- SMTP
- SSH
- SSL (including TLS)

Adding new protocols to SPID is, however, easy since all that is needed to create a new protocol model is a set of PCAP files to use as training data[11].

3.1.1 Application Layer Protocols in the Packet Captures

The TCP-based application layer protocols used in the captured network traffic are, according to SPID, mainly HTTP, SSL and SSH. The HTTP and SSL traffic stems from the time when the PS3 machines were running the default Cell OS, while the SSH sessions were not seen until after Linux was booted on the PS3 machines.

The TCP sessions that were not properly identified by SPID fall into three categories:

- False negatives, usually short HTTP sessions with limited data to perform the protocol matching on

[10] Available at http://sourceforge.net/projects/spid

[11] At least 10 TCP sessions of the application layer protocol is needed to build a reliable protocol model

- Encrypted sessions to hosts on the 217.18.16.0/20 network on TCP 10071, 10078, 10079 and 10375. Analysis in Wireshark reveals that the used protocol seems to be using public-key certificates with authentication of both the client and server.
- An unencrypted remote shell session from ps3.isi.jhu.edu to nssal-ps3 on TCP port 45541.

3.1.2 Protocols on Non-Standard Ports

Modern applications do often not follow the port number assignments made by IANA[12] (Moore and Papagiannaki, 2005); this is also the case with several TCP sessions in the analyzed captures. Reasons to avoid using standard port numbers can be to get traffic to pass through firewalls and in order to avoid detection by network monitoring tools.

The TCP-based application layer protocols, which do not use the proper IANA assigned ports, that were detected by SPID are:

- 137.30.120.90 to 137.30.120.109 are running HTTP servers on TCP 49152 and 49153
- 137.30.123.96 runs an HTTP server on TCP 2869
- 137.30.123.183 runs an HTTP server on TCP 2869 and 9500
- 198.107.156.132, 198.107.157.136 and 198.107.156.128 run SSL servers on TCP 5223
- 217.18.28.53 runs a HTTP server on TCP 10060 and an SSL server on TCP 10061

[12] Internet Assigned Numbers Authority

3.2 NetworkMiner Packet Analyzer

Most of the analysis made in this submission was made with a modified version of NetworkMiner-0.88[13]. The modification made to the network forensics tool is the TcpPacket.cs source file, which was modified to decode traffic to and from TCP ports 2869, 9500, 10060, 49152 and 49153 as HTTP and traffic to and from TCP ports 5223 and 10061 as SSL.

PCAP files can be analyzed in NetworkMiner by drag-and-droping them onto the application or by selecting "File > Open" in the menu. NetworkMiner can aggregate information from several PCAP files in one analysis session, and a list of the opened PCAP files and their MD5 sums are displayed in the "Case Panel" on the right-hand side of the user interface.

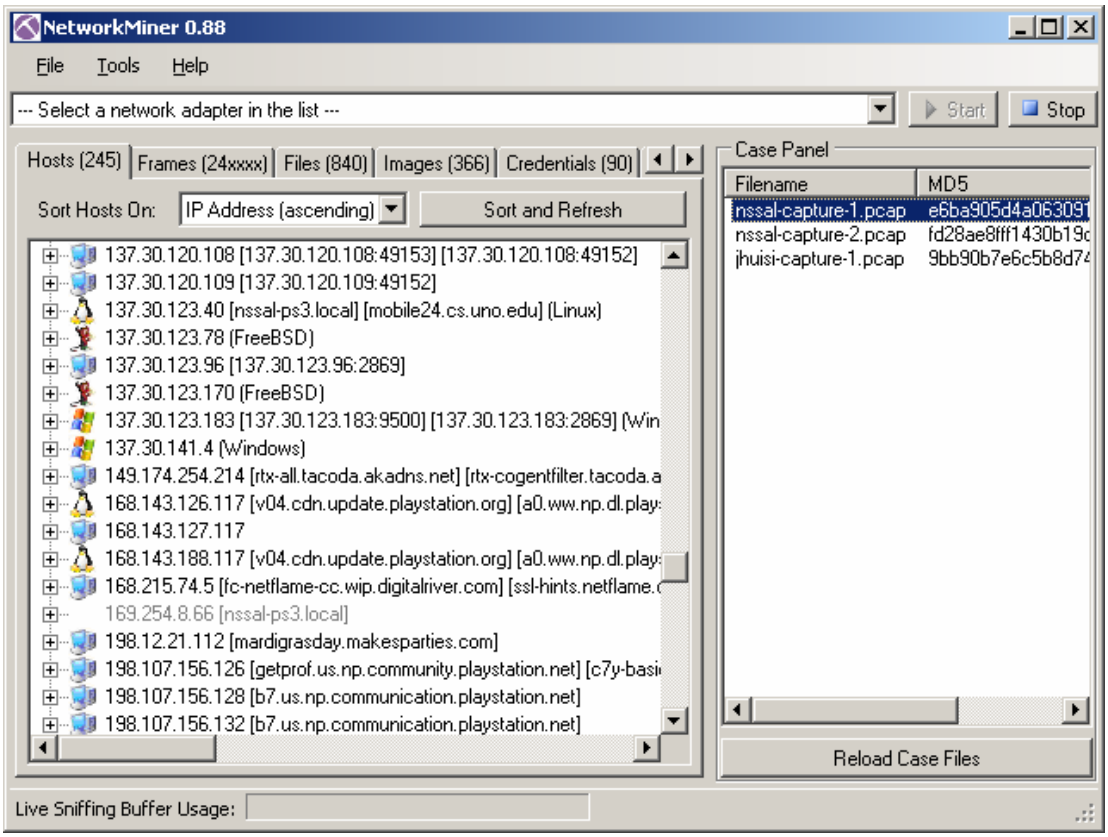

Image 4 NetworkMiner with all three case files loaded

[13] Available at: http://sourceforge.net/projects/networkminer

3.2.1 OS Fingerprinting

The OS fingerprinting functionality in NetworkMiner uses database files from p0f and Satori. The p0f database is built by Michal Zalewski and was designed for use with his tool p0f, which analyze the information in the IP and TCP headers in SYN and SYN+ACK packets in order to determine the operating system of a machine (Zalewski, 2005). The fact that the TCP and IP RFCs provide a fair amount of freedom in the implementation has rendered differences in the various TCP/IP stacks of different operating systems. Typical OS-specific differences are the IP time-to-live and the TCP maximum segment length values. The databases from Satori extend Zalewski's idea by also making use of fingerprints based on differences in DHCP implementations between various operating systems (Kollmann, 2007).

NetworkMiner displays the identified OS for each host in the "Hosts" tab by showing an icon for the OS next to the host in the tree-view. Each host can also be expanded, which enables the user to see a more detailed analysis of the matching OS fingerprints for that particular host.

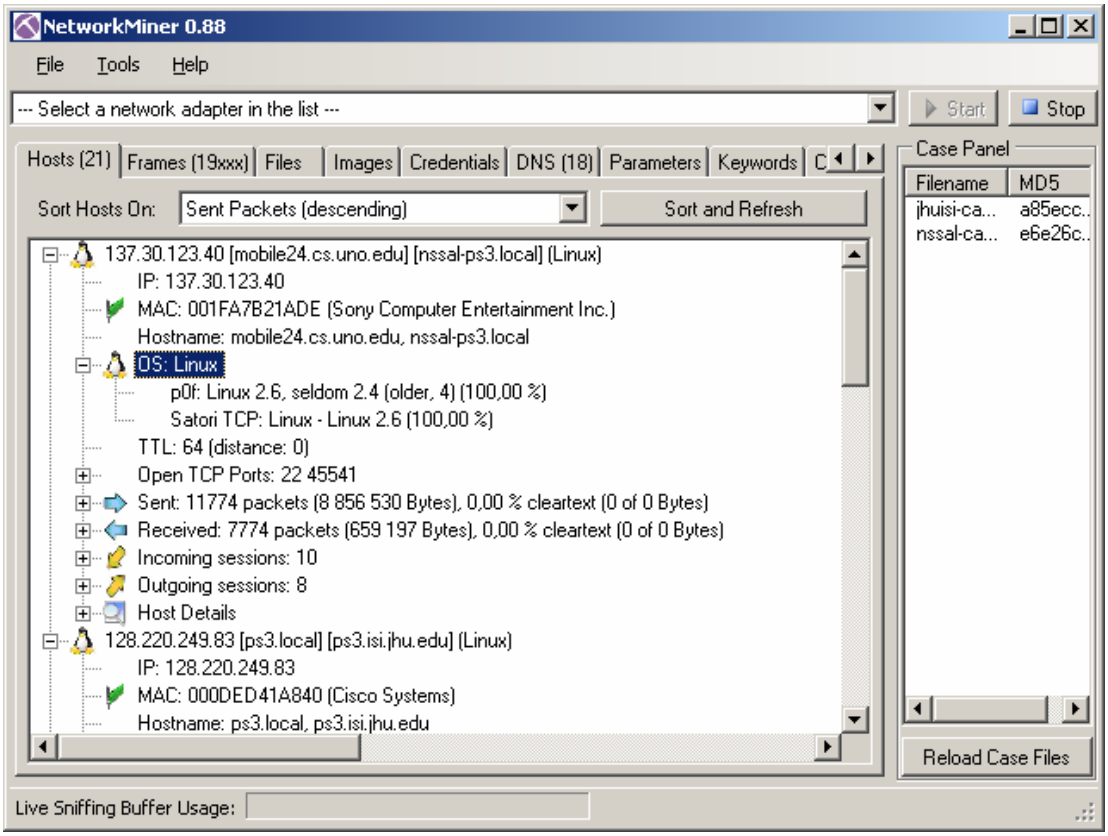

Image 5 Host tab showing OS fingerprint information for nssal-ps3

3.2.2 Web Browser User-Agents

NetworkMiner extracts the User-Agent information from web browsers and displays them on the "Hosts" tab under "Host Details". The User-Agent information can be used in order to identify the operating system of a machine.

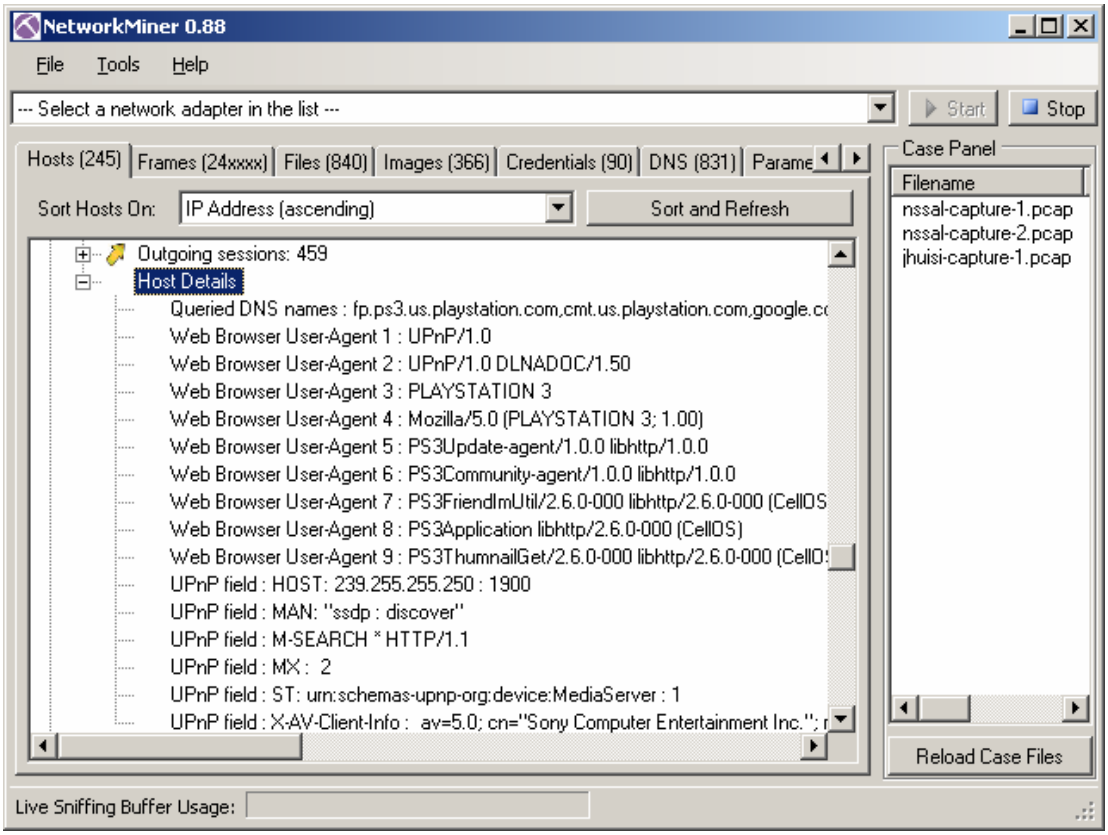

Image 6 Host Details for nssal-ps3 running the PS3 Cell OS

3.2.3 SSH Analysis

NetworkMiner parses the initial handshakes of the SSH sessions and extracts the SSH version and used SSH application. The SSH application banner from the two PS3 machines is "OpenSSH_5.1p1 Debian-3ubuntu1".

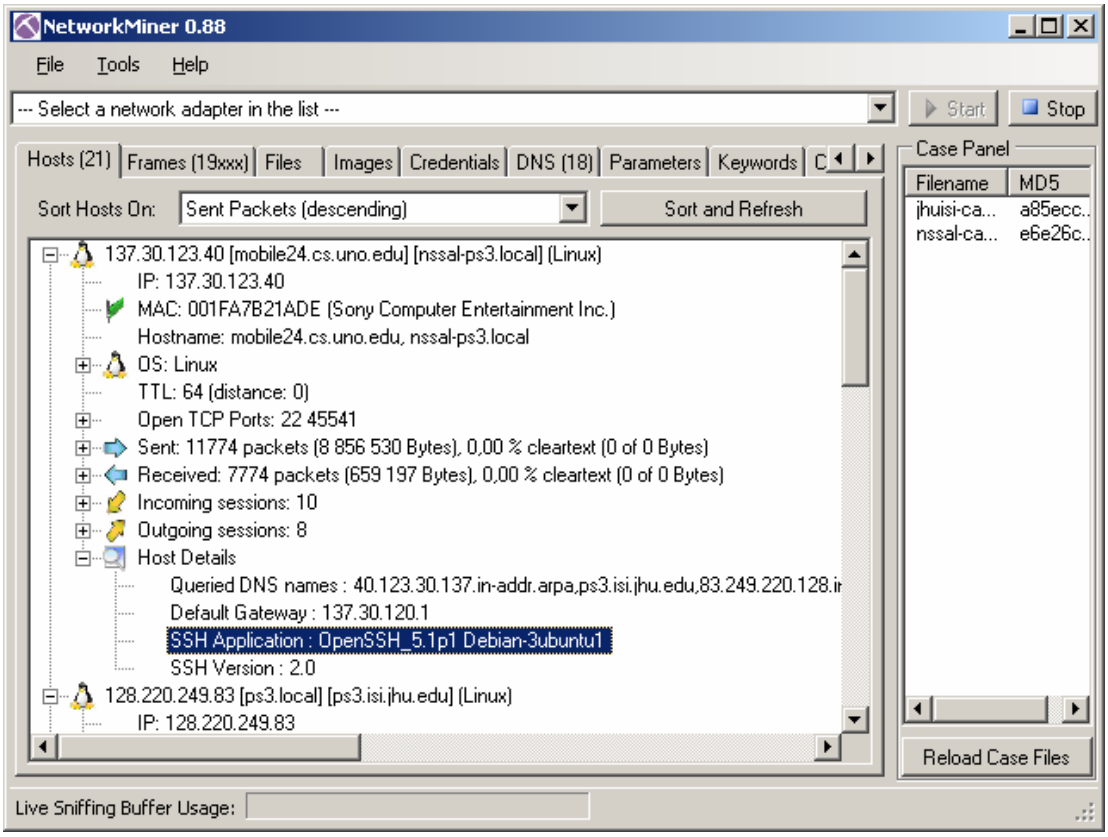

Image 7 Host details for nssal-ps3 showing the SSH banner

This information shows that there is a possibility that the Linux machines could be vulnerable to the well known Debian OpenSSL vulnerability (Wright, 2009). To investigate this further an analyst could download the ssh_kex_keygen plus ssh_decoder[14] and modify the code to produce big-endian[15] PIDs rather than little-endian ones.

Doing so does, however, lie outside the scope of my intentions with this submission since that would require a much more complex analysis that cannot be made with NetworkMiner.

3.2.4 Transferred Files

NetworkMiner has the ability to reassemble files transferred over protocols such as HTTP, FTP, TFTP and SMB. Unlike so-called file carving tools, which need a file format fingerprint for each file to extract, NetworkMiner performs file reassembly by performing deep packet inspection. This enables NetworkMiner to extract any file

[14] http://www.cr0.org/progs/sshfun/

[15] The PS3 is built on a PowerPC architecture, which normally runs in big-endian mode

format from the network and also enables extraction of files which are transferred using chunked, compressed or other encodings that obfuscate the data stream.

NetworkMiner automatically extracts all detected files to disk when a PCAP file is being loaded. The reassembled files are placed under separate folders for each host and network service. The directory structure of the reassembled files also mimics that of the server from where the files were retrieved.

A list of all reassembled files is show in the "Files" tab. Right-clicking the list opens a context menu that allows the user to open a file with the default application or to open the folder in which the file is located.

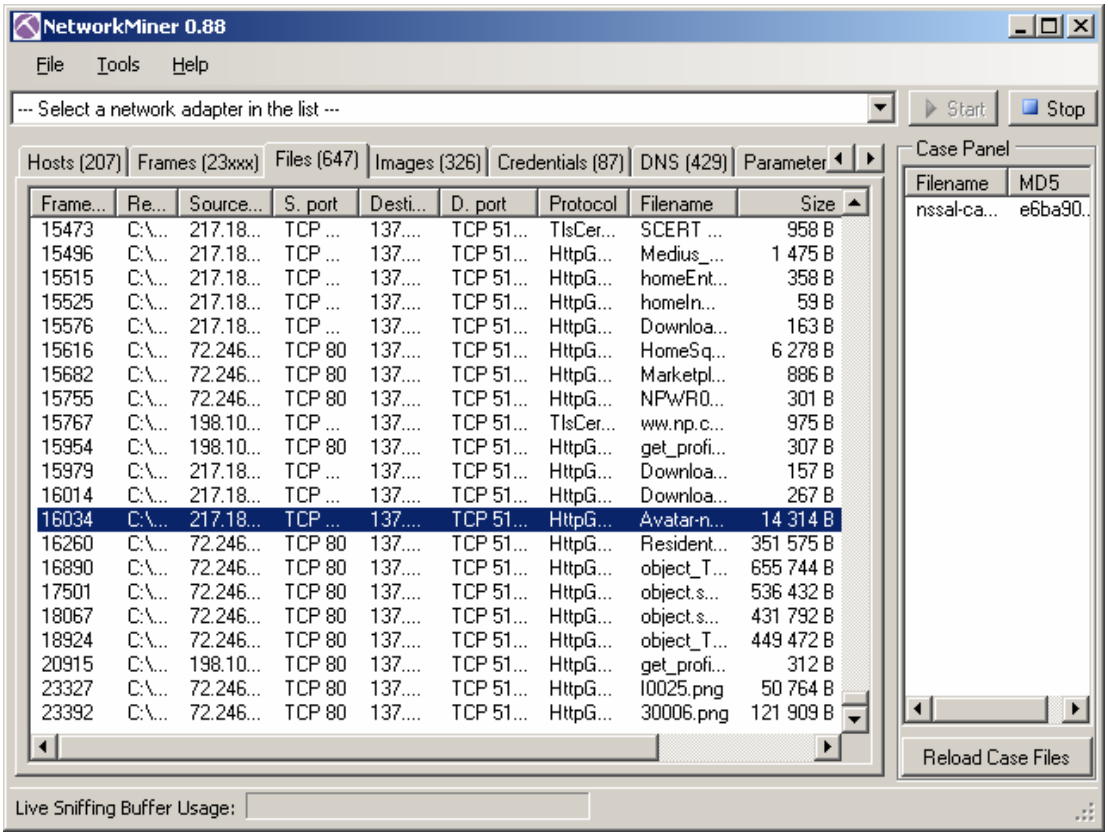

Image 8 Reassembled files from nssal-capture-1.pcap

Opening files directly is not recommended since there is a risk that the reassembled files contain malware. Malware files can also cause other problems when running NetworkMiner, since an Anti-Virus agent running on the analysis machine can choose to quarantine the infected file as NetworkMiner tries to write it to disk (Hull, 2009).

3.2.5 Browsing Images

Some of the reassembled files are images. NetworkMiner produces a thumbnail for each extracted image and displays it under the "Images" tab. The images tab gives a quick overview of the activity in a PCAP file, especially when a user is browsing websites.

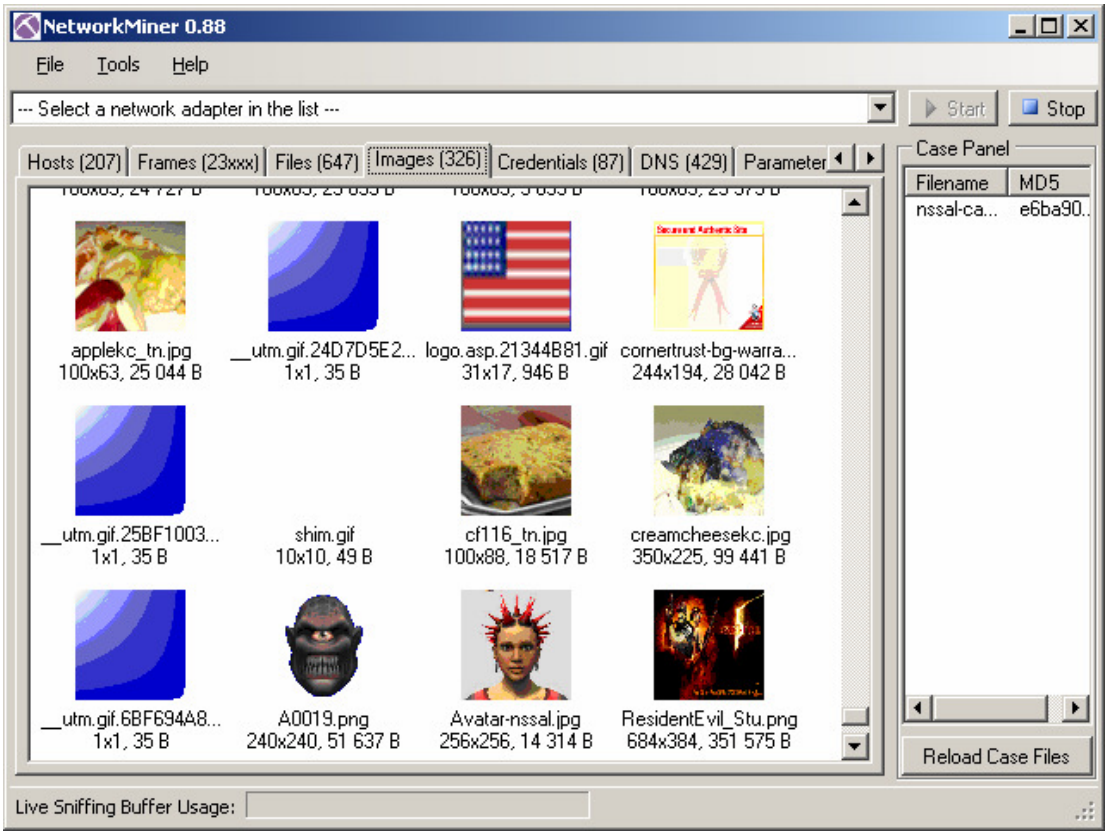

Image 9 Thumbnail view of images in nssal-capture-1.pcap

3.2.6 User Credentials

NetworkMiner extracts user credentials from several protocols including HTTP, FTP, NTLM, SMB, Tabular Data Stream (SQL) and the Spotify Key Exchange Protocol. All these types of user credentials are collected and displayed under the "Credentials" tab in NetworkMiner.

The credentials tab also contains HTTP cookies, which is how I retrieved the "nssal" and "jhuisi" login information to homeps3.svo.online.scee.com.

3.2.7 Search Engine Queries and other Parameters

All detected parameters sent as HTTP QueryStrings, in HTTP POSTs, FTP command sessions, SQL queries etcetera are all displayed in the "Parameters" tab in NetworkMiner.

The parameters tab comes in quite handy when analyzing search engine queries. By sorting the parameters data on "Parameter name" one can easily scroll down to all the parameters with a parameter name "q", which is the name used by several search engines (including Google) for the queries sent by the users (yahoo uses parameter name "p" for queries).

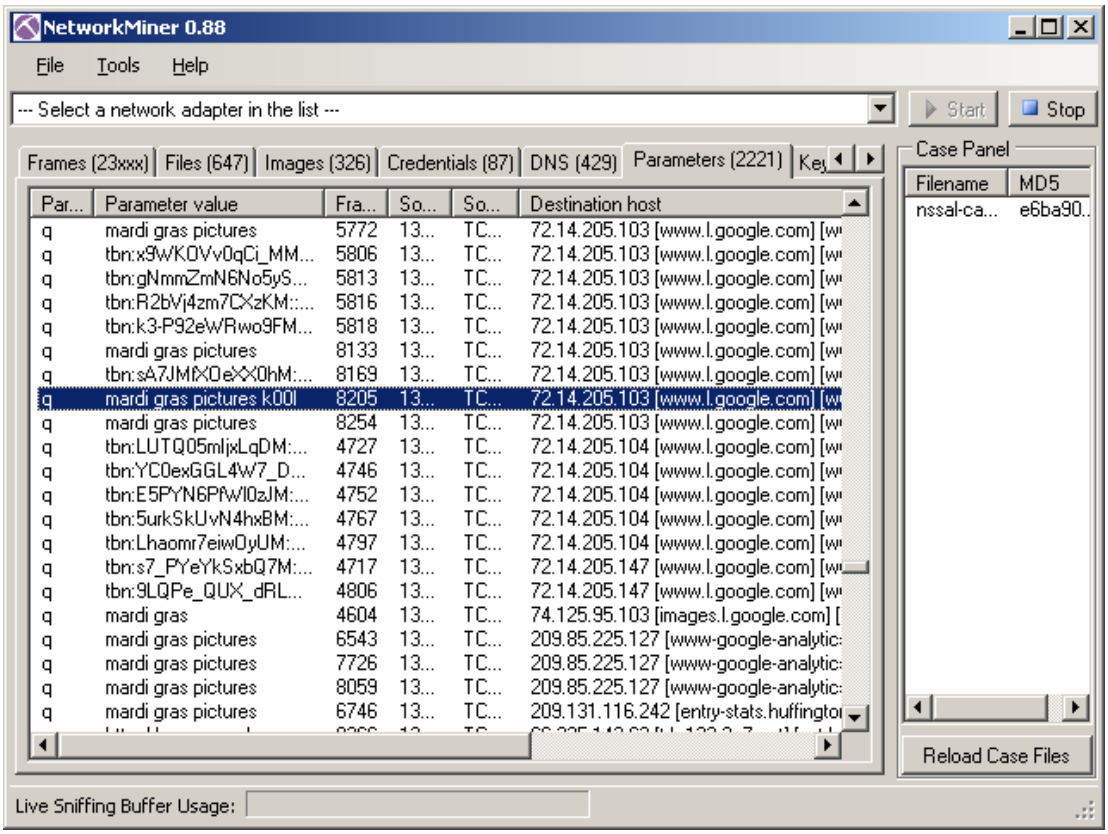

Image 10 Google queries listed under the Parameters tab

4 References

Hjelmvik E., "Passive Network Security Analysis with NetworkMiner". (IN)SECURE Magazine, Issue 18; 2008. p. 18-21.
http://www.net-security.org/dl/insecure/INSECURE-Mag-18.pdf

Hjelmvik E. and John W., "Statistical Protocol IDentification with SPID: Preliminary Results". In Proceedings of the 6:th Swedish National Computer Networking Workshop (SNCNW'09); 2009.
http://spid.sourceforge.net/sncnw09-hjelmvik_john-CR.pdf

Hull D., "NetworkMiner follow up". SANS Computer Forensics Blog; 2009.
https://blogs.sans.org/computer-forensics/2009/03/17/networkminer-follow-up/

Kollmann E., "Chatter on the Wire: A look at DHCP traffic"; 2007.
http://myweb.cableone.net/xnih/download/chatter-dhcp.pdf

Moore A. W. and Papagiannaki K., "Toward the accurate identification of network applications". In Proceedings of the Passive and Active Network Measurement: 6th International Workshop (PAM); 2005.

Wright J., "Decrypting Debian-Vulnerable SSH Traffic". Will Hack For SUSHI Blog; 2009.
http://www.willhackforsushi.com/Home/Entries/2009/2/3_Decrypting_DebIan-Vulnerable_SSH_Traffic.html

Zalewski M., "Silence on the Wire: A Field Guide to Passive Reconnaissance and Indirect Attacks". No Starch Press, ISBN:1593270461; 2005.