Interested in learning more about security?

SANS Institute InfoSec Reading Room

Repurposing Network Tools to Inspect File Systems

It can be a laborious and manual process to perform computer forensics to identify files of interest on a host. Any tools that can be used to automate portions of this process will help to accelerate computer forensics and reduce the likelihood of human error. If a host is compromised it is likely that antivirus and other host- based security tools have failed to detect and/or remediate malware that caused the compromise. If a host contains a file of interest that is not specifically malware then host-based security to...

Repurposing Network Tools to Inspect File Systems

GIAC (GCFA) Gold Certification

Author: Andre Thibault, andre.SANS.paper@gmail.com
Advisor: Johannes Ullrich, Ph.D.

Accepted: MMM DD 2014
(Date your final draft is accepted by your advisor)

Abstract

It can be a laborious and manual process to perform computer forensics to identify files of interest on a host. Any tools that can be used to automate portions of this process will help to accelerate computer forensics and reduce the likelihood of human error. If a host is compromised it is likely that antivirus and other host-based security tools have failed to detect and/or remediate malware that caused the compromise. If a host contains a file of interest that is not specifically malware then host-based security tools will also have not detected the file. It may be beneficial to repurpose existing network based security tools and signatures to evaluate local files. Network based tools may be used as a "second set of eyes" to inspect the file system. A possible method to perform this inspection could be to combine dd with Ncat/Netcat and Snort. In essence, the file system could be sent "over the wire" in order to allow "traditional" network based IDS inspection to be used in a novel manner.

1. Introduction

Digital forensics can be a laborious and multi-step process. Some of the initial steps in digital forensics include: Data Reduction, Anti-Virus checks, and an Indicator of Compromise (IOC) search. One purpose of Data Reduction is to simplify and scope the search for the data of interest (Lee, 2012). The analyst may be searching to see if known data (i.e. a string, or a file) exists or may be searching for completely unknown data (i.e. "is there anything bad on this filesystem?"). The Anti-Virus check may be used to check against "known bad" lists (signatures). The analyst may also search for Indicators of Compromise (either evidence of malware files or artifacts that result from the effects of malware), (OpenIOC, 2011). A host-based or "local" approach is typically and logically used to assist in the search for local data of interest. Each of these steps focuses entirely on data that is resident on the system and it would seem obvious to define the scope of the analysis in this manner.

However, it may also be beneficial to consider that much of the local data on a system originated from outside that system. This locally stored data has frequently made its way on to the system via a network connection. When considering that this local information has likely transited the network at some point in time, it may be wise to broaden the search for data of interest to the network. It may be worthwhile to obtain Intrusion Detection System/Intrusion Prevention System (IDS/IPS) logs for review to see if some of those local files or data ever triggered IDS/IPS events when transiting the network on their way to the target system. However, the forensic analyst may not have access to the relevant IDS/IPS logs in a number of situations: (1) the analyst is performing digital forensics as a third party; (2) the relevant IDS/IPS logs have fallen out of the log retention period; or (3) the owners of the IDS/IPS logs are unwilling to share the logs (since the logs could contain sensitive data and/or data that may not pertain to the target system). In this scenario it is possible (in a manner of speaking) to "regenerate" the relevant IDS/IPS logs. By re-routing the filesystem over the network it is possible to then inspect the data with an IDS/IPS. This allows the analyst to leverage well-established IDS rule infrastructure. If an organization has custom IDS rules based on

policies that pertain that organization, then violation of these custom rules (and the corresponding alerts) are an ideal starting point to identify items of interest. Re-routing the filesystem over the network (to allow IDS inspection) can be accomplished with well-known network utilities/tools such as dd and Ncat/Netcat (Maurya, 2009). IDS inspection can be done with an open source IDS such as Snort. By combining these tools in this manner one may be able to achieve results that may not otherwise be easily obtained.

1.1. dd

The Unix utility dd has long been used to copy and convert files. For its primary functionality, dd simply takes an input and then provides an output (Carrier, 2005). The great power of dd (as with many other Unix based utilities) lies in its many options and flexibility. This utility is often used by forensics analysts when creating forensic copies/images of a target filesystem. In the process of acquiring a forensic image, dd can be used to copy the target system (input) to a remote system (output). The forensic analyst can specify the input/output block size (number of bytes). Specifying the block size can affect the speed of the copying process and can also affect how efficiently data can be recovered. Text encodings can be specified to account for data format and/or differences in filesystem formats. There may already be a plan to use dd during forensic imaging when performing network-based acquisition (Carrier, 2005). Since the analyst will be generating network traffic during the imaging process it is advocated that the analyst use a few additional tools to take advantage of this network traffic by performing IDS inspection upon it.

1.2. Ncat/Netcat

The utility Ncat (Gibson, 2013) is a rewritten and improved version of another long used UNIX utility: Netcat (Hobbit, 1996). Both Ncat and Netcat are used primarily to communicate across a network connection. Like dd, these utilities take an input and provide an output, and whereas dd is primarily focused on files, these utilities are primarily focused on network communications. As with dd, these utilities are extremely flexible. The original Netcat utility has often been thought of as a "Swiss army knife" for network communications.

Andre Thibault, andre.SANS.paper@gmail.com

1.3. Snort

The Snort IDS (Roesch, 2013) is one of the most popular IDSs in the world. A major benefit associated with Snort is the ease with which one can create custom rules/signatures for the IDS. As a result of this facility there are a large number of rules/signatures covering many categories of interest that are available to the public (Emerging Threats, 2013). In addition, there are also a large number of Snort rules/signatures that are commercially and/or privately available. This vast repository of available signatures is one of Snort's greatest advantages

1.4. Snort rule advantages

Snort rules (Snort Rules, 2013) are written in plain text. Being in plain text makes these rules relatively easy to create, review, share, and customize quickly. Since IDS rules can sometimes be prone to errors and/or be prone to "firing" excessively it is important to be able to easily modify and tune these rules.

1.4.1. Rule Selection

When performing computer forensics on a target system owned by an organization it could be useful to leverage the IDS rules that are currently used by that organization. An organization has likely gone through a vetting and Quality Assurance (QA) process for its IDS rules. These rules can serve as a starting point to identify violations of an organization's policies. Beyond policy violations, an analyst may also desire to search for "unknown unknowns"; this would indicate the use of a larger IDS ruleset. If the analyst has been directed to search for a category of malware, or a specific information category, then the corresponding IDS ruleset could be selected and focused upon.

1.4.2. Custom Rules

If the analyst has been made aware of a specific piece of information to search for, corresponding custom IDS rules could be created. Examples could include rules for any of the following: (1) proprietary or classified information (e.g. intellectual property, secrets); (2) a specific exploit; or (3) any specific words or characters, i.e. "strings".

2. File inspection using network tools

2.1. Configuration

In order to rapidly facilitate a testing environment, an existing security focused Linux distribution called Network Security Toolkit (NST) was selected (Henderson & Blankenbaker, 2013). NST was selected based upon its facile integration of a number of the necessary network security tools (Snort, Netcat, and Ncat) as well as the standard Linux utility dd.

Figure 1. Screenshot of NST boot screen

Snort, although preinstalled, still requires some amount of configuration. The Snort configuration file, snort.conf, was modified. In addition, select Snort rulesets were identified and referenced in the snort.conf file. A custom Snort ruleset was created. Specific Snort "runtime" options were selected; options were selected to favor human readable review of testing results (rather than favoring performance). In a managed production setting, different Snort options may be selected to improve performance when running searches against a large dataset.

In this section of snort.conf, specific rulesets are identified.

```
include classification.config
include reference.config


##################################################
# Step #7: Customize your rule set
# For more information, see Snort Manual, Writing Snort Rules
#
# NOTE: All categories are enabled in this conf file
##################################################

# site specific rules
include $RULE_PATH/local.rules

include $RULE_PATH/community.rules
include $RULE_PATH/andre.rules
```

Figure 2. Screenshot of a section of the Snort.conf file.

In this screenshot specific Snort switches are used. Switch meanings are listed below.

```
[root@probe-wlan0 sbin]# sudo snort -C -d -e -K ascii -l /var/log/snort -v -i wlan0
Running in packet logging mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Log directory = /var/log/snort
pcap DAQ configured to passive.
Acquiring network traffic from "wlan0".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.4.5 GRE (Build 71)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.3.0
           Using PCRE version: 8.31 2012-07-06
           Using ZLIB version: 1.2.7

Commencing packet processing (pid=570)
```

Figure 3. Snort switches.

-C (log payload data in ascii, rather than hex)

-d (log application layer data)

-e (log link layer packet headers)

-K (specify log format, ascii is used for ease of human review)

-l (output results to log specified log path/file)

-v (verbose)

-i (specify the interface to listen on)

Andre Thibault, andre.SANS.paper@gmail.com

The tool Ncat was then configured. A remote computer was used as a destination; this system served as a recipient of the forensic target system's copied files. A persistent Ncat "server" was setup on a specific port on the remote computer:

```
Mac:bin macuser$ ncat -l 12345 -k | dd of=/Users/macuser/Desktop/netcat-dd-dest/file
```

Figure 4. Ncat server.

The following Ncat switches are used above:

-l (listener port)

-k (persistence)

The data is piped to dd and is then sent to the specified filesystem location.

From the forensic target system an Ncat "client" is used to communicate with the remote computer's Ncat "Server" port. The utility dd is used to copy files over this Ncat connection. For the purpose of file inspection it is not actually necessary that the files even reach their destination, however if forensic network-based image acquisition were being performed simultaneously, accurate data transfer would be a requirement.

2.2. Testing

Entire drive images or folders could be transferred and inspected using the methods described above. For the sake of brevity, individual files will be transferred and inspected in the following examples.

2.2.1. Policy Violation

If an analyst was searching for Policy Violations, the analyst could use a specific Snort ruleset/rules to assist with the search. An example could include a rule to search for documents tagged with security classification or Intellectual Property (IP) markings (e.g. Confidential, Sensitive, or Company Proprietary). These rules could assist with Data Reduction.

Here is a Snort rule designed to look for a specific class of confidential documents:

```
alert tcp any any -> any any (msg:"POLICY confidential"; flow: established;
pcre:"/CONFIDENTIAL//NOAGENTS/ism"; classtype:policy-other; sid:9000001; rev:1;)
```

Figure 5. Snort rule #1.

This rule will be used to detect the following sample "confidential" document.

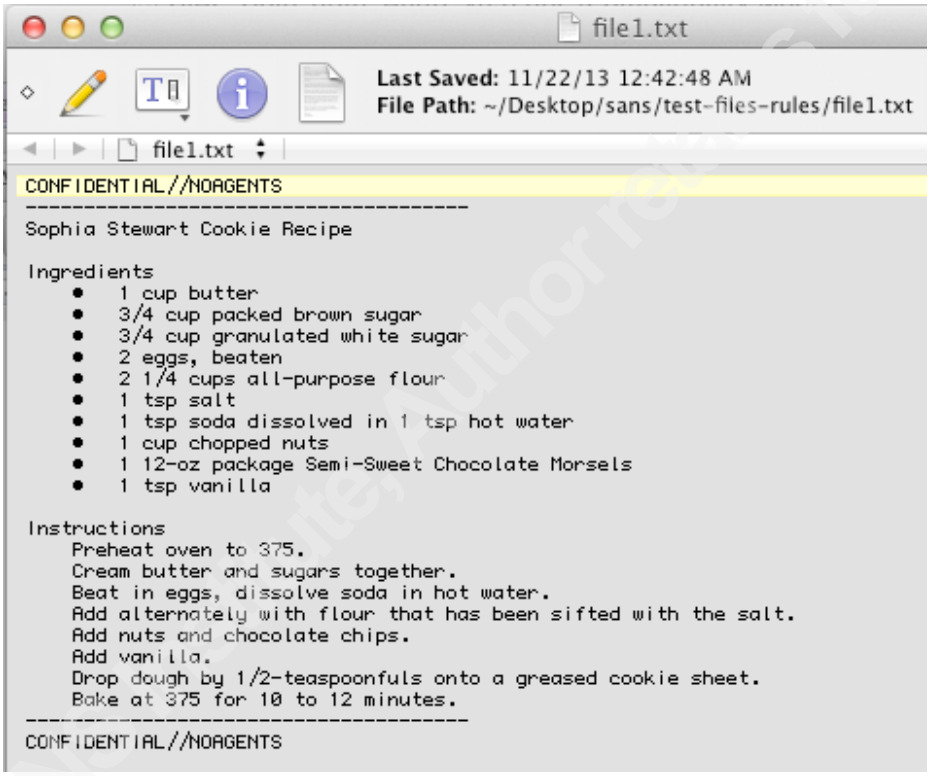

Figure 6. Screenshot of file1.txt.

The tool dd is used to copy a "confidential" file over Ncat.

```
[root@probe-wlan0 folder1]# dd if=/home/nst/folder1/file1.txt | ncat 10.0.1.2 12345
1+1 records in
1+1 records out
788 bytes (788 B) copied, 4.7352e-05 s, 16.6 MB/s
```

Figure 7. Transfer of file1.txt.

The Snort rule matches against the traffic and generates the log file TCP:35773-12345.

```
[root@probe-wlan0 folder1]# cd /var/log/snort/10.0.1.13
[root@probe-wlan0 10.0.1.13]# ls -l
total 32
-rw------- 1 root root 4464 Nov 24 22:18 TCP:35773-12345
-rw------- 1 root root 4464 Nov 24 22:18 TCP:35774-12345
-rw------- 1 root root 3780 Nov 24 22:18 TCP:35775-12345
-rw------- 1 root root 3789 Nov 24 22:19 TCP:35776-12345
-rw------- 1 root root 3330 Nov 24 22:19 TCP:35777-12345
```

Figure 8. Snort alert log file.

Here is the content of log file TCP:35773-12345.

```
TCP Options (8) => MSS: 1460 NOP WS: 3 NOP NOP TS: 250863658 363772805
TCP Options => SackOK EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:18:09.976540 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0x42
10.0.1.13:35773 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:51370 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x31D0732D  Ack: 0x4EF1EEA5  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363773022 250863658

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:18:09.976688 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0x356
10.0.1.13:35773 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:51371 IpLen:20 DgmLen:840 DF
***AP*** Seq: 0x31D0732D  Ack: 0x4EF1EEA5  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363773022 250863658
CONFIDENTIAL//NOAGENTS.---------------------------------------.So
phia Stewart Cookie Recipe..Ingredients......1 cup butter......3
/4 cup packed brown sugar......3/4 cup granulated white sugar...
...2 eggs, beaten......2 1/4 cups all-purpose flour......1 tsp s
alt......1 tsp soda dissolved in 1 tsp hot water......1 cup chop
ped nuts......1 12-oz package Semi-Sweet Chocolate Morsels .....
.1 tsp vanilla..Instructions..Preheat oven to 375. ..Cream butte
r and sugars together. ..Beat in eggs, dissolve soda in hot wate
r. ..Add alternately with flour that has been sifted with the sa
lt. ..Add nuts and chocolate chips. ..Add vanilla. ..Drop dough
by 1/2-teaspoonfuls onto a greased cookie sheet. ..Bake at 375 f
or 10 to 12 minutes..-------------------------------------.CONF
IDENTIAL//NOAGENTS..
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:18:09.976754 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0x42
10.0.1.13:35773 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:51372 IpLen:20 DgmLen:52 DF
***A***F Seq: 0x31D07641  Ack: 0x4EF1EEA5  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363773022 250863658

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 9. Snort rule successfully detects the confidential document.

Andre Thibault, andre.SANS.paper@gmail.com

2.2.2. Discover unspecified "bad" files

In a case where an analyst suspected that a host was compromised but did not know what malware caused the compromise, the analyst could leverage a set of Snort rules to search for Indicators of Compromise (IOC).

Here is an example of a modified Snort "Community Rule" (Community Rules, 2013) designed to detect a possible IOC.

```
alert tcp any any -> any any (msg:"INDICATOR-COMPROMISE config.inc.php in iframe";
flow:to_client,established; file_data; content:"<iframe"; content:"config.inc.php";
within:100; content:"</iframe>"; reference:url,blog.sucuri.net/2013/05/
auto-generated-iframes-to-blackhole-exploit-kit-following-the-cookie-trail.html;
classtype:malware-trojan; sid:26585; rev:9;)
```

Figure 10. Snort rule #2.

The above rule will be used to detect the following sample file containing an IOC.

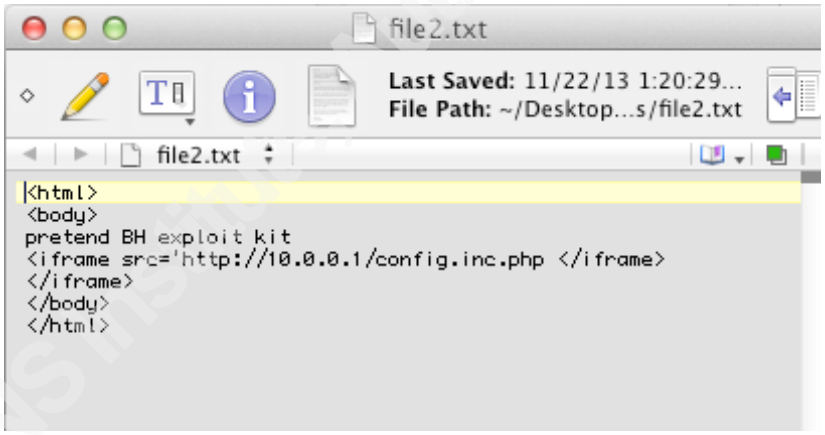

Figure 11. Screenshot of file2.txt.

The tool dd is used to copy a file containing an IOC over Ncat:

```
[root@probe-wlan0 folder1]# dd if=/home/nst/folder1/file2.txt | ncat 10.0.1.2 12345
0+1 records in
0+1 records out
116 bytes (116 B) copied, 0.00244018 s, 47.5 kB/s
```

Figure 12. Transfer of file2.txt.

The Snort rule matches against the traffic and generates the log file TCP:35775-12345.

```
[root@probe-wlan0 folder1]# cd /var/log/snort/10.0.1.13
[root@probe-wlan0 10.0.1.13]# ls -l
total 32
-rw------- 1 root root 4464 Nov 24 22:18 TCP:35773-12345
-rw------- 1 root root 4464 Nov 24 22:18 TCP:35774-12345
-rw------- 1 root root 3780 Nov 24 22:18 TCP:35775-12345
-rw------- 1 root root 3789 Nov 24 22:19 TCP:35776-12345
-rw------- 1 root root 3330 Nov 24 22:19 TCP:35777-12345
```

Figure 13. Snort alert log file

Here is the content of log file TCP:35775-12345.

```
11/24-22:18:55.637675 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0x4A
10.0.1.13:35775 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:54120 IpLen:20 DgmLen:60 DF
******S* Seq: 0xAB1BBDD3  Ack: 0x0  Win: 0x3908  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 363818683 0 NOP WS: 7

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:18:56.029868 B8:C7:5D:12:CA:CF -> 00:17:F2:44:07:79 type:0x800 len:0x4E
10.0.1.2:12345 -> 10.0.1.13:35775 TCP TTL:64 TOS:0x0 ID:8810 IpLen:20 DgmLen:64 DF
***A**S* Seq: 0x5E2B8F14  Ack: 0xAB1BBDD4  Win: 0xFFFF  TcpLen: 44
TCP Options (8) => MSS: 1460 NOP WS: 3 NOP NOP TS: 250909676 363818683
TCP Options => SackOK EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:18:56.029970 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0x42
10.0.1.13:35775 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:54121 IpLen:20 DgmLen:52 DF
***A**** Seq: 0xAB1BBDD4  Ack: 0x5E2B8F15  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363819075 250909676

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:18:56.030126 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0xB6
10.0.1.13:35775 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:54122 IpLen:20 DgmLen:168 DF
***AP*** Seq: 0xAB1BBDD4  Ack: 0x5E2B8F15  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363819076 250909676
<html>.<body>.pretend BH exploit kit.<iframe src='http://10.0.0.
1/config.inc.php </iframe>.</iframe>.</body>.</html>
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:18:56.030195 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0x42
10.0.1.13:35775 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:54123 IpLen:20 DgmLen:52 DF
***A***F Seq: 0xAB1BBE48  Ack: 0x5E2B8F15  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363819076 250909676

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 14. Snort rule successfully detects the IOC file.

Andre Thibault, andre.SANS.paper@gmail.com

2.2.3. Detect Specific type of Malware

If an analyst knew that a specific type of malware was involved but did not know which files on the system contained the malware, the analyst could leverage specific Snort rules or rulesets to narrow down the search. This search could assist with the Anti-Virus check.

This test will look for evidence of files that contain code that might be used to execute a "NOP sled" remote buffer overflow exploit (Honeynet Project Staff, 2002). Specifically, the search will look for the hex value for the NOP operation (0x90) in the x86 architecture. Here is an example rule intended to detect the buffer overflow exploit code (Lockhart, 2006).

```
alert tcp any any -> any any (msg:"Possible  exploit"; content:"|90|";)
```

Figure 15. Snort rule #3.

The above rule will be used to detect the following sample document containing a "buffer overflow" code snippet.

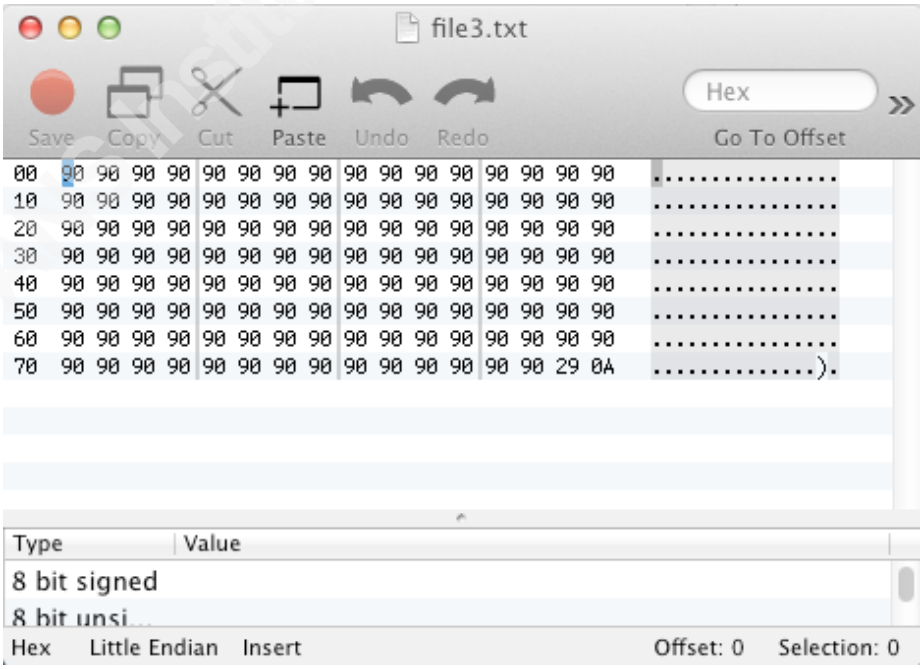

Figure 16. Screenshot of file3.txt.

The tool dd is used to copy a file containing "buffer overflow" code over Ncat:

```
[root@probe-wlan0 folder1]# dd if=/home/nst/folder1/file3.txt | ncat 10.0.1.2 12345
0+1 records in
0+1 records out
128 bytes (128 B) copied, 0.00238773 s, 53.6 kB/s
```

Figure 17. Transfer of file3.txt.

The Snort rule matches against the traffic and generates the log file TCP:35776-12345.

```
[root@probe-wlan0 folder1]# cd /var/log/snort/10.0.1.13
[root@probe-wlan0 10.0.1.13]# ls -l
total 32
-rw------- 1 root root 4464 Nov 24 22:18 TCP:35773-12345
-rw------- 1 root root 4464 Nov 24 22:18 TCP:35774-12345
-rw------- 1 root root 3780 Nov 24 22:18 TCP:35775-12345
-rw------- 1 root root 3789 Nov 24 22:19 TCP:35776-12345
-rw------- 1 root root 3330 Nov 24 22:19 TCP:35777-12345
```

Figure 18. Snort alert log file.

Here is the content of log file TCP:35776-12345.

```
11/24-22:19:03.402214 B8:C7:5D:12:CA:CF -> 00:17:F2:44:07:79 type:0x800 len:0x4E
10.0.1.2:12345 -> 10.0.1.13:35776 TCP TTL:64 TOS:0x0 ID:53351 IpLen:20 DgmLen:64 DF
***A**S* Seq: 0x2BCBEBF7  Ack: 0x96AC843C  Win: 0xFFFF  TcpLen: 44
TCP Options (8) => MSS: 1460 NOP WS: 3 NOP NOP TS: 250917039 363826197
TCP Options => SackOK EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:19:03.402306 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0x42
10.0.1.13:35776 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:2095 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x96AC843C  Ack: 0x2BCBEBF8  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363826448 250917039

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:19:03.402452 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0xC2
10.0.1.13:35776 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:2096 IpLen:20 DgmLen:180 DF
***AP*** Seq: 0x96AC843C  Ack: 0x2BCBEBF8  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363826448 250917039
................................................................
................................................................).

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:19:03.402517 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0x42
10.0.1.13:35776 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:2097 IpLen:20 DgmLen:52 DF
***A***F Seq: 0x96AC84BC  Ack: 0x2BCBEBF8  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363826448 250917039

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:19:03.405680 B8:C7:5D:12:CA:CF -> 00:17:F2:44:07:79 type:0x800 len:0x42
10.0.1.2:12345 -> 10.0.1.13:35776 TCP TTL:64 TOS:0x0 ID:60494 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x2BCBEBF8  Ack: 0x96AC843C  Win: 0xFFFF  TcpLen: 32
TCP Options (3) => NOP NOP TS: 250917042 363826448

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 19. Snort rule successfully detects the file containing "buffer overflow" code.

Andre Thibault, andre.SANS.paper@gmail.com

2.2.4. String Detection

If the analyst is looking for a specific piece of data, such as a character string, the analyst could select or create a rule (or rules) to search for this string. These rules could assist with the Data Reduction check. Below is a custom rule designed to detect a specific string. The section after "content:" contains the search string (in this case 999-999-9999). This test string is modeled on a Social Security Number (SSN). More complex Snort rules exist to detect validly formatted SSNs. Similar types of rules could be created or downloaded to detect validly formatted, unencrypted credit card numbers (McMillan, 2009).

Snort rule:

```
alert tcp any any -> any any (msg:"String Search 999-99-9999";content:"999-99-9999";)
```

Figure 20. Snort rule #4.

The above rule will be used to detect the following sample document containing the "string of interest".

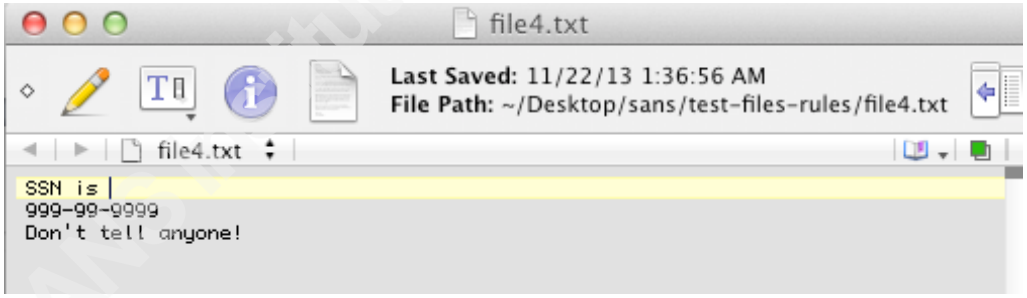

Figure 21. Screenshot of file4.txt.

The tool dd is used to copy a file containing the "string of interest" over Ncat.

```
[root@probe-wlan0 folder1]# dd if=/home/nst/folder1/file4.txt | ncat 10.0.1.2 12345
0+1 records in
0+1 records out
38 bytes (38 B) copied, 0.00268721 s, 14.1 kB/s
```

Figure 22. Transfer of file4.txt.

The Snort rule matches against the traffic and generates the log file TCP:35777-12345.

```
[root@probe-wlan0 folder1]# cd /var/log/snort/10.0.1.13
[root@probe-wlan0 10.0.1.13]# ls -l
total 32
-rw------- 1 root root 4464 Nov 24 22:18 TCP:35773-12345
-rw------- 1 root root 4464 Nov 24 22:18 TCP:35774-12345
-rw------- 1 root root 3780 Nov 24 22:18 TCP:35775-12345
-rw------- 1 root root 3789 Nov 24 22:19 TCP:35776-12345
-rw------- 1 root root 3330 Nov 24 22:19 TCP:35777-12345
```

Figure 23. Snort alert log file.

Here is the content of log file TCP:35777-12345.

```
11/24-22:19:10.165666 B8:C7:5D:12:CA:CF -> 00:17:F2:44:07:79 type:0x800 len:0x4E
10.0.1.2:12345 -> 10.0.1.13:35777 TCP TTL:64 TOS:0x0 ID:57924 IpLen:20 DgmLen:64 DF
***A**S* Seq: 0x8B9C10A  Ack: 0xCD823943  Win: 0xFFFF  TcpLen: 44
TCP Options (8) => MSS: 1460 NOP WS: 3 NOP NOP TS: 250923791 363832945
TCP Options => SackOK EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:19:10.165763 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0x42
10.0.1.13:35777 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:40756 IpLen:20 DgmLen:52 DF
***A**** Seq: 0xCD823943  Ack: 0x8B9C10B  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363833211 250923791

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:19:10.165979 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0x68
10.0.1.13:35777 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:40757 IpLen:20 DgmLen:90 DF
***AP*** Seq: 0xCD823943  Ack: 0x8B9C10B  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363833212 250923791
SSN is .999-99-9999.Don't tell anyone!
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:19:10.166052 00:17:F2:44:07:79 -> B8:C7:5D:12:CA:CF type:0x800 len:0x42
10.0.1.13:35777 -> 10.0.1.2:12345 TCP TTL:64 TOS:0x0 ID:40758 IpLen:20 DgmLen:52 DF
***A***F Seq: 0xCD823969  Ack: 0x8B9C10B  Win: 0x73  TcpLen: 32
TCP Options (3) => NOP NOP TS: 363833212 250923791

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/24-22:19:10.172954 B8:C7:5D:12:CA:CF -> 00:17:F2:44:07:79 type:0x800 len:0x42
10.0.1.2:12345 -> 10.0.1.13:35777 TCP TTL:64 TOS:0x0 ID:5644 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x8B9C10B  Ack: 0xCD823943  Win: 0xFFFF  TcpLen: 32
TCP Options (3) => NOP NOP TS: 250923800 363833211
```

Figure 24. Snort rule successfully detects the “string of interest”.

3. Conclusion

Digital forensic examination of a filesystem can at times be an arduous process. An analyst should leverage tools as much as possible to automate parts of the process and to reduce the likelihood of human error when performing repetitive tasks or when searching for "a needle in a haystack". Several network tools were combined and tested to evaluate for suitability with respect to inspection of files on a digital forensic "target" filesystem. A number of IDS rules were selected, modified, and/or created with the goal of detecting specific data of interest in sample files. Each rule successfully detected the data of interest in the sample files. It is concluded that repurposing these network tools can indeed assist with the automation of some aspects of the digital forensic examination of a filesystem.

Andre Thibault, andre.SANS.paper@gmail.com

4. References

Carrier, B. (2005). *File System Forensic Analysis*. (pp. 60-64). Boston, MA: Pearson Education, Inc.

Community Rules (2013). Download Snort Rules. Retrieved November 23, 2013, from Snort Web site: http://www.snort.org/snort-rules/

Emerging Threats (2013). Index of /open/snort-2.9.0/rules. Retrieved October 29, 2013, from Emerging Threats Web site: http://rules.emergingthreats.net/open/snort-2.9.0/rules/

Gibson, C. (2013). Ncat. Retrieved September 3, 2013, from Nmap Web site: http://nmap.org/ncat/

Henderson, R., & Blankenbaker, P. (2013). Network Security Toolkit. Retrieved July 16, 2013, from Network Security Toolkit Web site: http://networksecuritytoolkit.org/nst/index.html

Hobbit (1996). Netcat 1.10. Retrieved August 16, 2013, from Sourceforge Web site: http://nc110.sourceforge.net

Honeynet Project Staff. (2002). *Know Your Enemy*. (p. 62). Indianapolis, IN: Addison-Wesley Professional.

Lee, R. (2012). SANS Digital Forensics and Incident Response Poster. (p. 2). Retrieved January 11, 2014, from SANS Web site: https://blogs.sans.org/computer-forensics/files/2012/06/SANS-Digital-Forensics-and-Incident-Response-Poster-2012.pdf

Lockhart, A. (2006). *Network Security Hacks*. (2nd ed., p. 374). Sebastopol, CA: O'Reilly Media, Inc.

Maurya, A. (2009). Remote Mirroring Using nc and dd. Retrieved October 29 2013, from Linux Journal Web site: http://www.linuxjournal.com/content/tech-tip-remote-mirroring-using-nc-and-dd

McMillan, J. (2009). Using Snort to Detect Clear Text Credit Card Numbers. Retrieved November 23, 2013 from SANS Web site: http://www.sans.org/security-resources/idfaq/snort-detect-credit-card-numbers.php

OpenIOC (2011). An Introduction to OpenIOC. (p. 4). Retrieved January 11, 2014, from OpenIOC Web site: http://openioc.org/resources/An_Introduction_to_OpenIOC.pdf

Andre Thibault, andre.SANS.paper@gmail.com

Roesch, M. (2013). Snort homepage. Retrieved October 29 2013, from Snort Web site: http://www.snort.org

Snort Rules (2013). Open Source, Snort Rules. Retrieved November 23, 2013, from Sourcefire Web site: http://www.sourcefire.com/products/open-source

5. Appendix

```
#-----------------
# Snort rules used in SANS paper.  See body of paper for description and references.
#-----------------

alert tcp any any -> any any (msg:"POLICY confidential"; flow: established;
pcre:"/CONFIDENTIAL//NOAGENTS/ism"; classtype:policy-other; sid:9000001;
rev:1;)

alert tcp any any -> any any (msg:"INDICATOR-COMPROMISE config.inc.php in
iframe";
flow:to_client,established; file_data; content:"<iframe"; content:"config.inc.php";
within:100; content:"</iframe>"; reference:url,blog.sucuri.net/2013/05/
auto-generated-iframes-to-blackhole-exploit-kit-following-the-cookie-trail.html;
classtype:malware-trojan; sid:26585; rev:9;)

alert tcp any any -> any any (msg:"Possible  exploit"; content:"|90|";)

alert tcp any any -> any any (msg:"String Search 999-99-9999";content:"999-99-
9999";)
```

Andre Thibault, andre.SANS.paper@gmail.com

Last Updated: August 24th, 2017

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Tampa - Clearwater 2017	Clearwater, FLUS	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Network Security 2017	Las Vegas, NVUS	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS Dublin 2017	Dublin, IE	Sep 11, 2017 - Sep 16, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MDUS	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, DK	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, GB	Sep 25, 2017 - Sep 30, 2017	Live Event
Data Breach Summit & Training	Chicago, ILUS	Sep 25, 2017 - Oct 02, 2017	Live Event
Rocky Mountain Fall 2017	Denver, COUS	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS SEC504 at Cyber Security Week 2017	The Hague, NL	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Oslo Autumn 2017	Oslo, NO	Oct 02, 2017 - Oct 07, 2017	Live Event
SANS DFIR Prague 2017	Prague, CZ	Oct 02, 2017 - Oct 08, 2017	Live Event
SANS Phoenix-Mesa 2017	Mesa, AZUS	Oct 09, 2017 - Oct 14, 2017	Live Event
SANS October Singapore 2017	Singapore, SG	Oct 09, 2017 - Oct 28, 2017	Live Event
SANS AUD507 (GSNA) @ Canberra 2017	Canberra, AU	Oct 09, 2017 - Oct 14, 2017	Live Event
Secure DevOps Summit & Training	Denver, COUS	Oct 10, 2017 - Oct 17, 2017	Live Event
SANS Tysons Corner Fall 2017	McLean, VAUS	Oct 14, 2017 - Oct 21, 2017	Live Event
SANS Tokyo Autumn 2017	Tokyo, JP	Oct 16, 2017 - Oct 28, 2017	Live Event
SANS Brussels Autumn 2017	Brussels, BE	Oct 16, 2017 - Oct 21, 2017	Live Event
SANS Berlin 2017	Berlin, DE	Oct 23, 2017 - Oct 28, 2017	Live Event
SANS San Diego 2017	San Diego, CAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Gulf Region 2017	Dubai, AE	Nov 04, 2017 - Nov 16, 2017	Live Event
SANS Miami 2017	Miami, FLUS	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Amsterdam 2017	Amsterdam, NL	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Milan November 2017	Milan, IT	Nov 06, 2017 - Nov 11, 2017	Live Event
Pen Test Hackfest Summit & Training 2017	Bethesda, MDUS	Nov 13, 2017 - Nov 20, 2017	Live Event
SANS Paris November 2017	Paris, FR	Nov 13, 2017 - Nov 18, 2017	Live Event
SANS Sydney 2017	Sydney, AU	Nov 13, 2017 - Nov 25, 2017	Live Event
SANS San Francisco Fall 2017	OnlineCAUS	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced