Case #3: Attacker's Kali System!

You have been called to analyze the system of a known threat actor who has been accused of breaking into the server of a smaller company. The company has now come to you asking for help figuring out if they had actually been exploited and how. You have been given an Image of the system used for the attack, but not the endpoint that was potentially exploited.
Find the different ways in which the threat actor gained access to the system and what modifications were made.

Deliverables:

1. What is the time range of exploitation?
2. How many exploits can you find evidence of? What can you not find evidence of and why?
3. Where could evidence of each exploit be found? (Multiple places for some, singular for others)
4. Did the attacker act maliciously on the target system?
5. Can evidence be found of NFS usage?

Outcomes:

At the end of this lab, you will have the required skills to deal with a compromised Linux system, were you will be capable of doing:

1. Understanding the locations and functions of a handful of Linux log files
2. Using linux's built in commands to quickly create and apply a timeline of user or system activity
3. Applying log parsing techniques
4. Searching through the FHS

Task #1: Finding Time Range of Attack

When did activity begin and end for the active user account?

$ last -F
$ last -F root
First two entries are going to be from the image release
Last -f

Find the files modified in this time range? (sample for proof of concept & command)

$ls -la /
$lsblk or $mount
#get getting mounted location of drive
$stat /lib
#getting inode number
$debugfs /dev/sda1
$debugfs <16>
#Inode number of /lib, a folder created at compile
$debugfs q
#close debugfs

This is checking for the time of compile to determine if that may 17th logon date is relevant to our timeline or not

$ find / -newermt "2019-09-06 18:30:00" -not -newermt "2019-09-08 00:15:00" > quicktimeline.txt

Trying to find activity from May 17th onward will end up with an extremely long running command due to this login date being the time of the version release being pushed out. This is not necessarily the case with other linux distributions though.

Task #2: Finding Exploits Used Against Target System

Did the attacker use metasploit on the victims machine? If they did what did they do?

We've found metasploit logs under /root/.msf4 on the system in the list of modified files we found. Looking at the history and logs/framework.log, we can find out how metasploit was used to exploit the system.

```
$ less /root/.msf4/history
```

#No timestamps but shows all commands run in metasploit by the attack. Similar to a .bash_history.

```
$ less /root/.msf4/logs/framework.log
```

#Find logs that are between the date range found in Task #1. This will show details on what was run in exploit during this time frame.

#By default this is only error logging for metasploit - so only failures will make entries here. It is unlikely this will be changed by an attacker to be more verbose, but it is theoretically possible.

Find out what other exploits were run on the system using our timeline.

#Also found that VNC was run. Let's see where the attacker was trying to connect to.

```
$ cat /root/.vnc/default.tigervnc
```

#This file shows the configuration of the last connection that was attempted by VNC. The timestamps of this file will correlate to the last attempted connection. We can see the same IP address that was targeted by metasploit in this file.

What can you not find evidence of? (This will be explained)

#using telnet does not create any logs due it being such an old and simple protocol. Specifying a log file is possible though.

#commands entered in metasploit. These are not dropped to the attacking system's logs by default. A switch can be turned on within the metasploit framework to log commands to the .msf4/logs/sessions/(session) location, which will record commands executed on the remote system, but this is then crushed at the end of a session as well.

Task #3: Finding Evidence of NFS Usage

Where can evidence of NFS usage be found on the system? What does it tell us?

Method 1:

$cat /var/log/syslog | grep -B 5 -A 5 "NFS" > nfs1.txt

this gives a before and after of five lines for anything matchine “NFS” with capital or lowercase matches. This provides some context around the matches as well to help fill in the story if needed.

#This is one of the few sources of nfs logging that is created on linux by default. Default syslog configurations dump debug NFS information to the syslog file, but can be configured to behave differently.

Method 2:

The more reliable method that doesn’t work in the default distribution of Kali!

$journalctl --output=short-full > journal.txt

#This will work for the case you are working on, but it will not normally. Let's examine why.

$cat /root/.bash_history

#config in /etc/systemd/journald.conf of systemd settings

Note: please use tables and screenshots to represent your results if needed. Like I usually say “**Screenshot or it didn’t happen!**”.