# Security Ops

# Activity 3: Monitor the webserver for shellshock exploitation attempts

The frontline of defense is security operations. These employees monitor the servers for compromise & intervene in the event of compromise. Not every company is large enough to justify the cost of full time employees who monitor for evidence of compromise.

**Step 1:** shell into the server.

1. Open a terminal window.
2. Type ssh pentesterlab@192.168.64.210 (pwd is tcuser)

**Step 3:** Let’s monitor the web logs:

1. tail –f /usr/local/apache2/logs/access\_log



Tail is a way of viewing the entries at the end of a file. This makes it very helpful for viewing log files, which can be very large. The –f directive tells it to follow the file. If there is new data added to the end of the file, it will show that data in real time.

This file captures a log of all access attempts on the server. It shows

* the IP address of the client that connected
* the time they connected (down to the second)
* The HTTP method that was used
* The URI that was accessed
* The status code that the web server returned (200 means everything was ok! 404 means file not found).

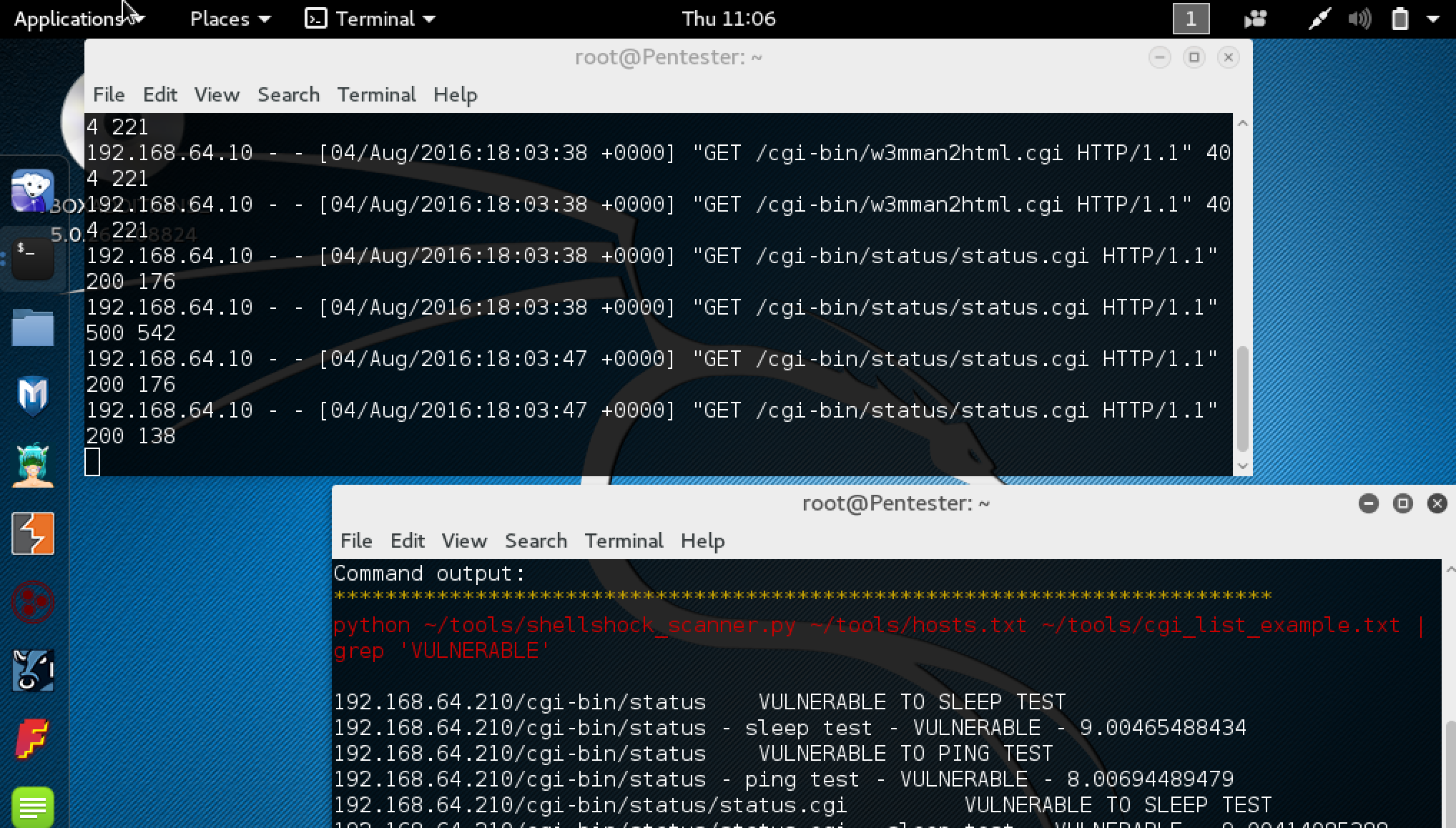
If you open a web browser on your pentesting VM & connect to the server entry, you’ll notice that new entries will get added to the log.

This type of log is on practically every web server on the planet. Every time you access a web page, a webserver is gathering this information about you.

**Step 3:** Watch access\_log during scan

Let’s open a new terminal on the pentester workstation and start our vulnerability scanning script & compare the entries for a normal browser connection versus our vulnerability scan:

* Open a second terminal window & resize them to be side by side. The one on the left should be your existing ssh session. The one on the right will be for launching attacks.
* We’re going to run only the shellshock vulnerability scan against the server. This is automated if you run the following command:
* *./vulntest.sh*



Notice that as you execute the scan, you can see log entries being added on the web server.

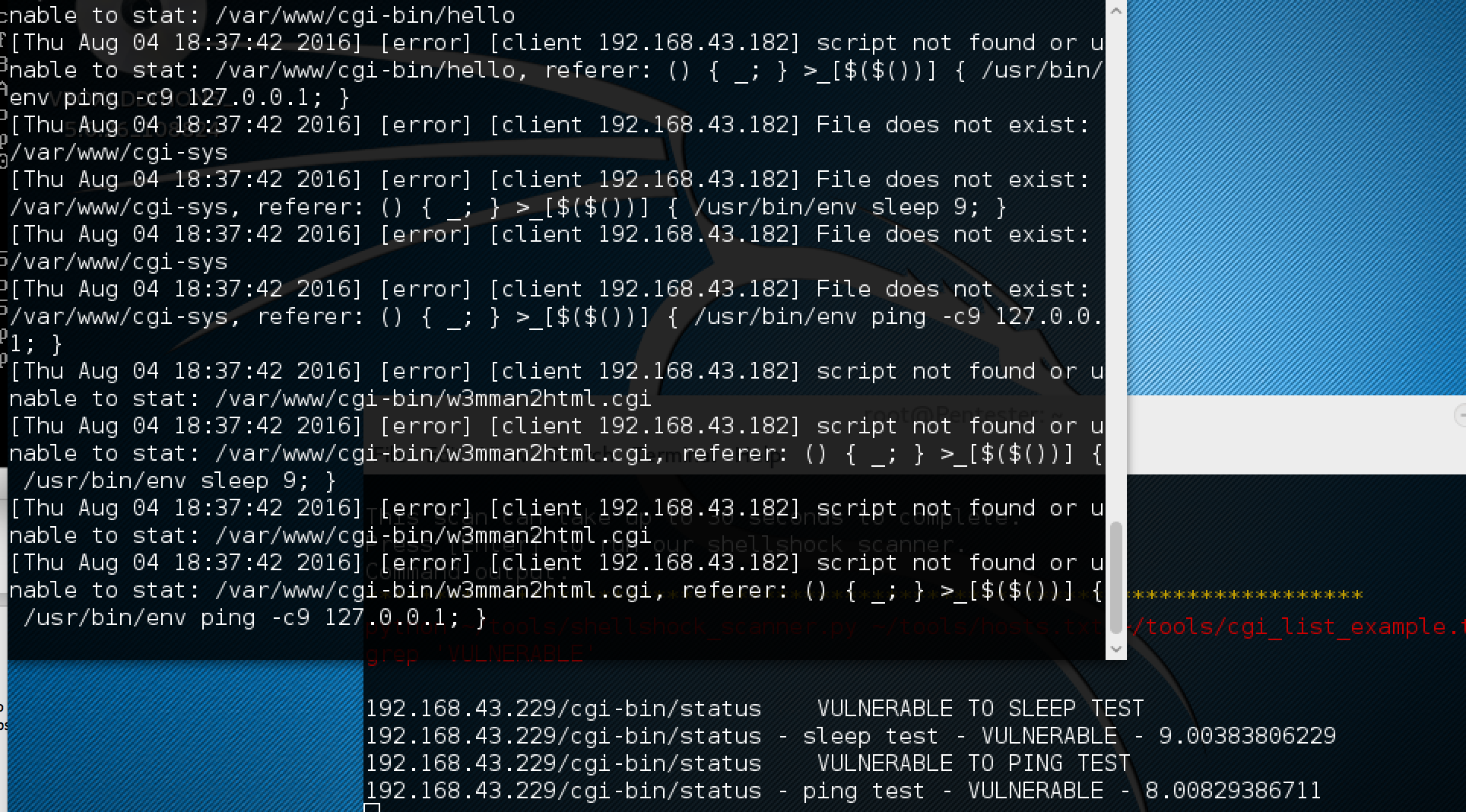
# Activity: Compare access\_log entries for normal traffic & scanning traffic

Scroll through the log entries & compare the scan entries to the earlier entries for regular traffic.

* Do you notice any differences?
* Is there any patterns that you see in the access log for the scan traffic?

**Step 4:** Watch error\_log during scan

1. Hit ctrl+c to close out the tail session of access logs
2. Type tail –f /usr/local/apache2/logs/error\_log
3. Go back to the scanner terminal and type vulntest.sh
4. Observe error logs as the scan runs



# Activity: Observe error\_log entries for normal traffic & scanning traffic

Scroll through the log entries & observe the numerous errors

* Do you notice any patterns?
* Do you recognize any strings from Part 1 or Part 2 in the new error log?

Based on the errors above, we can easily determine that the server underwent a scan for shellshock vulnerabilities. We can derive this because of the attempts to access a script that had the magic () { \_; } pattern from our previous shellshock research.

The logs do not prove vulnerability, however. They only show a scan. Scans happen to production servers on the Internet all the time, and it is the responsibility of the security operations team to make sure that the servers are patched from these vulnerabilities, or that they have other controls to prevent the system from being hacked.

# Concluding questions:

* Is browsing to a website private?
* How do security operations teams detect hackers?