PenTest Lab Exercise Ex0e0 – `SSLStrip`

Goal

Identify a possible target for `sslstrip` and do what's required to use the exploit, get credentials, etc., and write your report.

Tasks

1. Use some method to gain access to `devbox.artstailor.com` in an Ex0e0 pod as a user with root privilege. I know this is tedious, but life is difficult. If you've taken good notes and paid attention, this doesn't take long.

2. Hank has prepared a directory full of tools he found useful when using `sslstrip` on a different linux host that didn't have all of the necessary python libraries and executables installed. The directory is `sslstrip-extras` in your `kali` home directory. You can use `scp -r` to copy this whole directory to the machine you are trying to use to mount an `sslstrip` attack.

   The `sslstrip.py` python executable depends on some modules that aren't usually installed on a linux server. You probably don't need to install those things to get it to work. The two `.whl` files in this directory are python *wheel* files which are installed by running

   ```
   sudo pip install filename.whl
   ```

   You will definitely need to install the sslstrip module by changing into the sslstrip subdirectory and running `sudo python setup.py install` there. You can read the README file.

   These commands must be run on the host where you will mount your `sslstrip` attack–not on your `kali` host.

3. Once on `devbox`, use `tcpdump` to capture packets on the network it's attached to. You'll want to capture packets uninterrupted for several minutes. Then you'll want to look for packets from hosts that might be making both http and https connections to the same server. These reveal potential opportunities for `sslstrip`. Make sure you visit the web page referenced in the http connection to determine whether, in fact, there is an opportunity for stripping ssl. Explain how you determined this fact in your report.

Here are some tips:

(a) You can find a tcpdump tutorial. Things to check out: `-i` parameter (to specify ethernet interface), `-w` parameter (to specify `.pcap` file to write to), `-r` parameter (to read from a `.pcap` file), Berkeley Packet Filter (BPF) arguments like `dst port ## or dst port ###`, and `-X` and `-XX` to send packet content to `stdout`.

(b) There are two fundamental ways to go: either you capture packets using `tcpdump` and use `BPF` and output to identify the packets that are being captured, or you dump packets to a file then `scp` that file back to your kali host to inspect using `wireshark`. (If you run `wireshark` at the command line, you can provide a `.pcap` file argument from which to read packets.)

(c) A `wireshark` filter you might want to use would be something like this:

```
tcp.port == ## or tcp.port == ###
```

4. Once you've found which host to attack, you need to plan an `sslstrip` attack. You'll need to consider these elements:

(a) Remember a MitM attack using `arpspoof` must be mounted on the local network of the attacked host!

(b) Set the `devbox` kernel to do IP forwarding.

(c) Start up `sslstrip`, *making sure to provide parameters that will guarantee that all information captured will be logged*.

(d) If a service is running on port 80, you must *s*top that service in order to mount an `sslstrip` attack. If this were a real penetration test, we would modify the web server and the `iptables` rules to make sure that the web server could serve up web pages. This is not necessary in your assignment.

(e) Insert an `iptables` rule to redirect packets from port 80 to port 10000 (or whatever other port `sslstrip` may be listening on).

(f) Use `arpspoof` to convince the target and its gateway to route all traffic through devbox.

(g) Be patient! (It may take a while for someone to get snagged by your trap.)

(h) Double check the logging you are doing with `sslstrip`. People often don't capture everything they need to capture.

5. After some time, you may notice that a number of `http` requests and responses will have been filtered by `sslstrip`. One way to see updates to a log file (named, say, `logfile`) is to execute

   ```
   tail -f logfile
   ```

   which will start up, show the last few lines in `logfile`, and continue to show more data as it is concatenated to the end of the file.

6. You will likely find some packets that use *Basic Authentication*. You can identify the encoded credentials that will have been captured by `sslstrip` and use an appropriate decoding program to decode these credentials as necessary. These would be automatically decoded in a `wireshark` session, but if they are delivered via https, then you won't see them.

7. All avenues that could lead to potential sensitive corporate information should be followed.

8. Write a partial penetration test report with any findings you have and submit it in Canvas. The finding should include the risk level, vulnerability explanation, confirmation method, and mitigation. Include an attack narrative in which you explain how you initially identified the problem.