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 containing a few tools he found useful when using `sslstrip` on a different linux host that didn't have all of the necessary python 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 contained in `sslstrip3.tgz`. If you un-tar that file, you can execute it by issuing `python3 sslstrip.py` from that same directory. Make sure to read the readme file carefully before trying to use the program as an attack tool and make sure either know the default log filename or create a different log file. Also make sure to use the logging level that will provide the most output.

   The attack described in the `README.md` file in `sslstrip3` must be run on the host where you will mount your `sslstrip` attack–not on your `kali` host. That host should be on a layer 2 network where a host making sslstrippable http requests resides.

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. You can only be sure you have identified this opportunity if you find a web page that is reachable

by http that has an https link. Explain identification of an sslstripping opportunity to Hank 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. If you are setting NAT rules, you can list them by executing

```
iptables -L -t NAT
```

If you install a broken NAT rule, you can remove it with

```
iptables -F -t NAT
```

If you have multiple conflicting NAT rules, rules are followed in order of insertion in time until the packet has been sent somewhere. This usually means that earlier bad rules are applied by later good rules. Don't make the mistake of installing multiple NAT rules. Remove the bad ones!

(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 and a web server were running on the machine you are using, you 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 whatever other port `sslstrip` wille be listening on.

(f) Use `arpspoof` to convince the target and its gateway to route all traffic through devbox. Pay close attention to the `arpspoof` parameters. The `-i` parameter specifies the interface that is on the layer 2 network where you want the spoofing to occur.

(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. If you see anything that could give you access to a web site, follow up on it! Penetration testers are paid to be curious!

8. Write a partial penetration test report with any findings you have and submit it in Canvas. The finding should include any vulnerability and also an attack narrative containing you TTP(s) explaining, among other things, how you initially identified the problem.