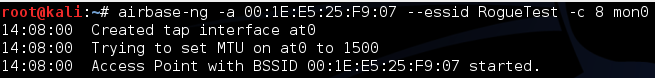
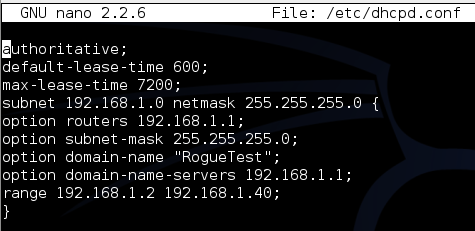
Ryan Stearns  
Lab 8 – Cont’d  
CNA 432

**Lab 8**

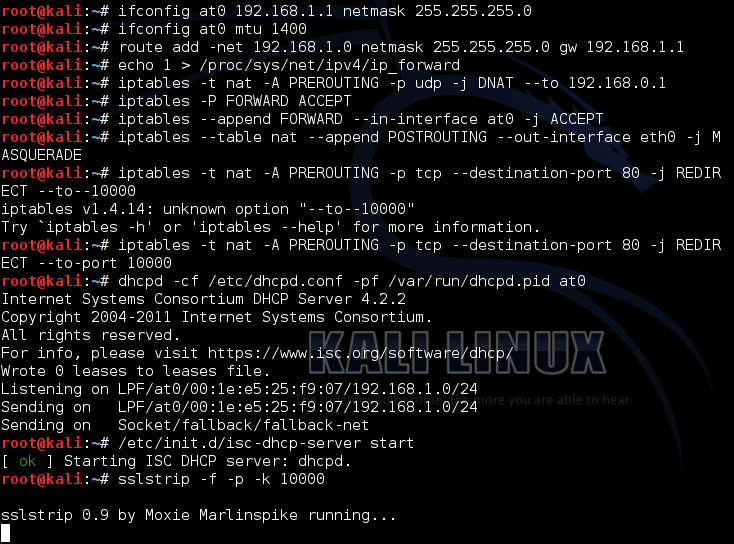
My second go at SSLStripping from a man-in-the-middle attack included a couple different techniques than my first time implementing it in the lab. I created an access point from my wireless adapter using the airbase-ng package.

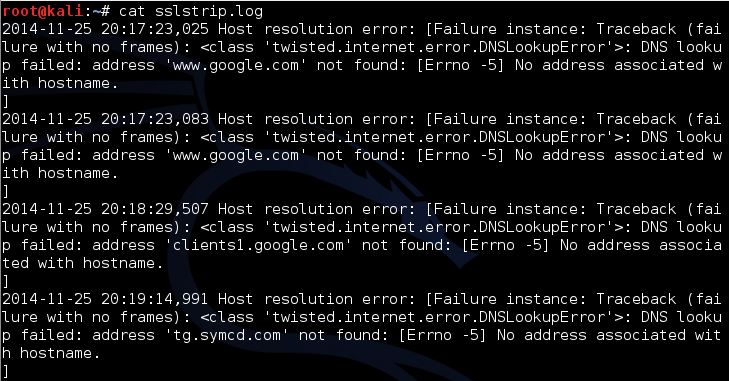


I then installed dhcp3-server and made a configuration file for my network.



From there, I configured the at0 interface created by airbase-ng, enabled IP forwarding, and forwarded the necessary ports to 10000. From there I started the DHCP server, and started sslstrip, as well as ettercap.





Although sslstrip was running, I was unable to capture any usernames or passwords with ettercap. This was caused by the inability to open an SSL-enabled website using just HTTP. Every time HTTP was used on an SSL site, I would automatically be redirected to the HTTPS version. This happened to sites I have never been to before as well. I tried disabling the browser.urlbar.autofill option on FireFox, but I was still automatically redirected to HTTPS. After trying different browsers, I decided to research the cause of this.

According to Mozilla, “HTTP Strict Transport Security (often abbreviated as HSTS) is a security feature that lets a web site tell browsers that it should only be communicated with using HTTPS, instead of using HTTP.” [1] HTTP Strict Transport Security was defined and published in 2012 as RFC 6797. According to OWASP, browser support for HTTP Strict Transport Security is included in Internet Explorer, Firefox, Opera, Safari, and Google Chrome. [2] Also, “HSTS does not allow a user to override the invalid certificate message.” [2]

HSTS works by having an HTTP response header field with the name Strict-Transport-Security. Broadly, this header has a max-age that the end client must abide by, and only access the site using HTTPS. “When the expiration time specified by the Strict-Transport-Security header elapses, the next attempt to load the site via HTTP will proceed as normal instead of automatically using HTTPS. Whenever the STS Header is delivered to the browser, it will update the expiration time for that site, so sites can refresh this information and prevent the timeout from expiring”

HSTS prevents SSLStripping, Downgrade Attacks, Cookie Hijacking and much more. According to the RFC 6797, the Threats Addressed include: Passive Network Attackers, Active Network Attackers, and Web Site Development and Deployment Bugs. Threats Not Addressed by RFC 6797 are Phishing, and Malware and Browser Vulnerabilities. [3] In conclusion, HTTP Strict Transport Security made it impossible to access a HTTPS website using just HTTP, and thus, Ettercap was not able to capture any login information.

**Resources**

[1] "HTTP Strict Transport Security." *Mozilla Developer Network*. N.p., n.d. Web. 25 Nov. 2014.

<https://developer.mozilla.org/en-US/docs/Web/Security/HTTP\_strict\_transport\_security>

[2] "HTTP Strict Transport Security." *- OWASP*. N.p., n.d. Web. 25 Nov. 2014.  
 <https://www.owasp.org/index.php/HTTP\_Strict\_Transport\_Security>  
  
[3] Hodges, Jeff, Collin Jackson, and Adam Barth. "Http strict transport security (hsts)."

< http://tools.ietf.org/html/draft-ietf-websec-strict-transport-sec-04(2012).>