PROJECT REPORT

ON

TIME TRAVEL FOR REAL: BREAKING HTTPS USING NTP

ELEG-697 ADVANCED CYBERSECURITY

UNIVERSITY OF DELAWARE

Submitted by:

Vishnu Bharadwaj

Kartik Khanna

Sarvanan Ramesh

CONTENTS

SUMMARY

The project focuses on understanding the Network Time Protocol and its vulnerabilities. We exploited one of its vulnebilities and expired the SSL certificate in the browser. We have shown that even the HSTS is also breakable.

INTRODUCTION

1. **HTTP Strict Transport Security (HSTS)** – It's a security feature that lets a web site tell browsers that it should only be communicated with using HTTPS, instead of using HTTP through a use of a specific response header. Once a supported browser receives this header that browser will prevent any communications from being sent over HTTP to the specified domain and will instead send all communications over HTTPS.

   An example scenario

   You log into a free WiFi access point at an airport and start surfing the web, visiting your online banking service to check your balance and pay a couple of bills. Unfortunately, the access point you're using is actually a hacker's laptop, and they're intercepting your original HTTP request and redirecting you to a clone of your bank's site instead of the real thing. Now your private data is exposed to the hacker.

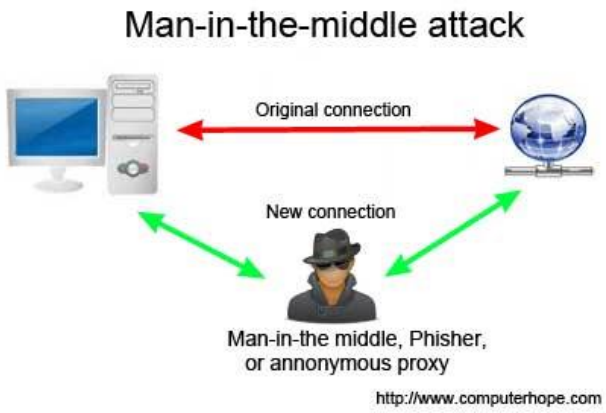
   Strict Transport Security resolves this problem; as long as you've accessed your bank's web site once using HTTPS, and the bank's web site uses Strict Transport Security, your browser will know to automatically use only HTTPS, which prevents hackers from performing this sort of man-in-the-middle attack.

2. **Network Time Protocol (NTP)** - NTP stands for *Network Time Protocol*, and it is an *Internet* protocol used to synchronize the clocks of computers to some time reference. NTP uses Coordinated Universal Time (UTC) to synchronize computer clock times to a millisecond, and sometimes to a fraction of a millisecond. UTC time is obtained using several different methods, including radio and satellite systems. Computers designated as *primary time servers* are outfitted with the receivers and they use protocols such as NTP to synchronize the clock times of networked computers. Degrees of separation from the UTC source are defined as strata. A radio clock (which receives true time from a dedicated transmitter or satellite navigation system) is stratum-0; a computer that is directly linked to the radio clock is stratum-1; a computer that receives its time from a stratum-1 computer is stratum-2, and so on.

Need

Time usually just advances. If you have communicating programs running on different computers, time still should even advance if you switch from one computer to another. Obviously if one system is ahead of the others, the others are behind that particular one. From the perspective of an external observer, switching between these systems would cause time to jump forward and back, a non-desirable effect.

3. **Man-in-the-middle attack (MITM)** - The man-in-the middle attack intercepts a communication between two systems. For example, in an http transaction the target is the TCP connection between client and server. Using different techniques, the attacker splits the original TCP connection into 2 new connections, one between the client and the attacker and the other between the attacker and the server, as shown in figure 1. Once the TCP connection is intercepted, the attacker acts as a proxy, being able to read, insert and modify the data in the intercepted communication.



PROCEDURE

For our project we will be using Kali-Linux since it has all the Networking tools install in it.

1. **Network Scanning** – In order to find the victim's ip address we will be running network scanner. Network Scanning is a procedure for sending data packets via the network to a specified subnet or computing system's specified service port numbers. All the system or ports that are active at that moment will send a reply back and will display their respective ip address along with what system it is. For this project we will be using Nmap scanner for network scanning.

   nmap –sP subnet_address

2. **Man-In-The-Middle** – We will be using ARP-Spoofing in order to implement this attack. ARP is an ethernet layer 2 address, network hardware address (MAC). The communication between nodes in the network established between source and destination through layer 2 address, i.e., MAC address, in the network the nodes share their MAC address through broadcast. Every node in the network maintains an ARP cache table which consists of L2 and L3 addressing (IP address and associated MAC address). In ARP spoofing, we "spoof" our computer to act as the wireless router. We can then have other devices on the network send their data to us, because they think that by doing so, it will be routed to the Internet. Of course, we'll make sure that everything is eventually sent back and forth from the real router, so that you don't notice anything different.
   Now that we've placed our computer "in the middle" of the Internet connection and everyone else, we get to look at all of the network traffic as it goes by. Since our system is acting as router now we first have to enable ip forwarding in it.

   echo 1 > /proc/sys/net/ipv4/ip_forward

   With the use of iptables we will redirect the coming request to the user defined port number. Iptables is taking traffic inbound to our machine which the destination is port 123 known for UDP-NTP traffic and redirecting the traffic to the port 123. We just want the NTP request to gateway through us.

   iptables -t nat -A PREROUTING -p udp --destination-port 123 -j REDIRECT --to-port 123

Now we can start ARP spoofing, or more specifically, poisoning the victim's ARP cache. On Linux, it's best to do this in two separate terminal windows since they need to run at the same time. Type each of these commands into a separate window:

arpspoof -t <target_ip> <gateway_ip>
arpspoof -t <gateway_ip> <target_ip>

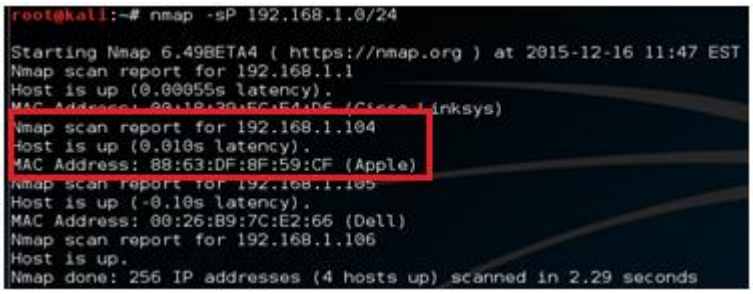
After a few seconds, your victim's traffic should be successfully redirecting to your computer. We can verify it by running Wireshark if all the traffic is coming and going through our ip address then it's successful.

3. **Delorean-** In order to perform NTP Man-in-the-Middle attacks, a new tool called 'Delorean' has been developed and it is available for download at Github. It's a python script based on the kimifly's tool 'ntpserver' through which we can jump to any desired time.

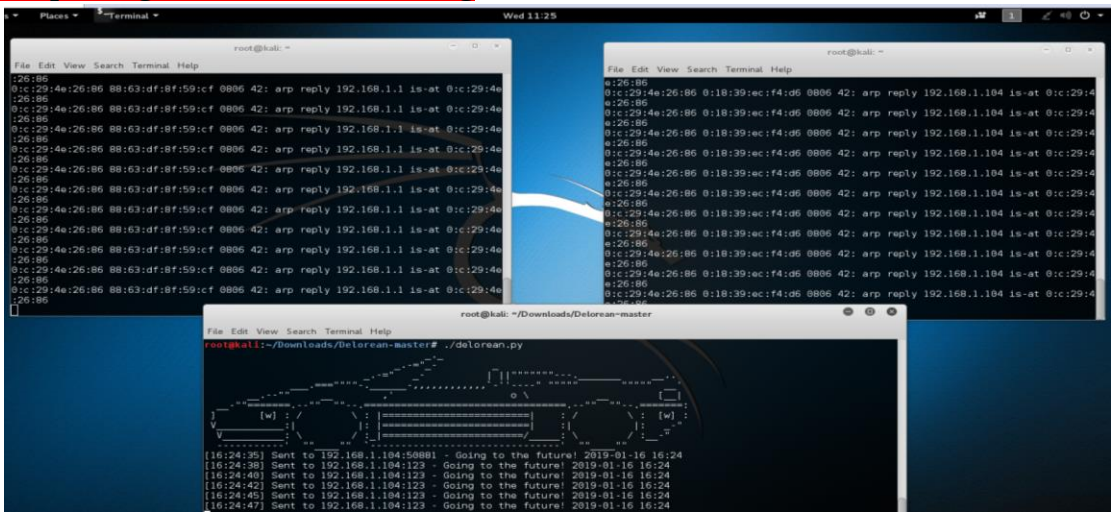
   - Since we know HSTS security relies on time and for reference it trust the OS's current time. Operating System uses NTP as Time Synchronization Protocol over the internet.
   - The Delorean code can be activated by a simple command: **./delorean.py.** This code mainly creates its own UDP time packet with the required time change.
   - It continuously watches for UDP time packets from port 123, the incoming packet is blocked and a modified time packet is sent to the same port instead. Since the port is already open for response it readily accepts the received NTP packet and the time changes.
   - For our project we are going to make the system go forward four years in future.

RESULTS

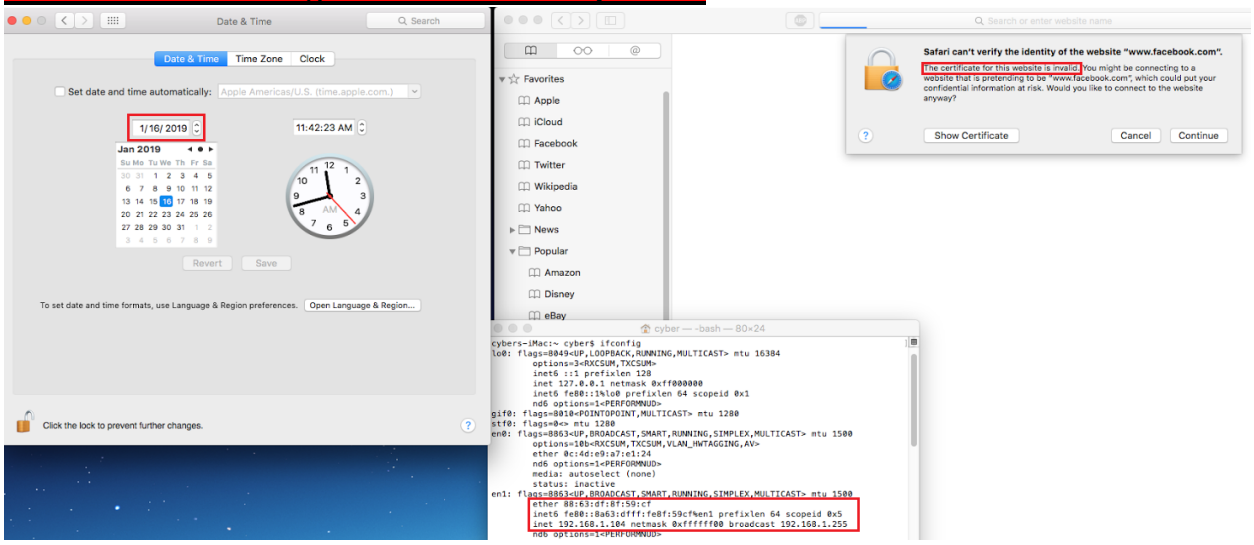
Victim's IP Address



ARP-Spoofing and Dolorean Running



Successful Time Change and Certificate Expiration



FUTURE WORK

When system time changes a lot of things can happen in a system. If it's in future many log files can become old and might get deleted. We have now shown that the browser warns us about the expired certificate but since we know Browsers these days have self-signed certs and we can create and inject one which may not show that the cert has expired We can make it to go back in time and can exploit the http version of a website which was not SSL secure at that time. We can explore which all applications go back to its previous version with respect to time and exploit that previous versions without any patches.

Since cookies too have a lifetime so with this we can expire a cookie. With a use of sslstrip and greasemonkey tool we can strip https and then can inject the new cookie which will take place of the expired one.

REFERENCES

- https://www.blackhat.com/docs/eu-14/materials/eu-14-Selvi-Bypassing-HTTP-Strict-Transport-Security-wp.pdf
- https://hackertarget.com/nmap-cheatsheet-a-quick-reference-guide/
- http://7riversys.com/howto-mitm-attack/
- https://github.com/PentesterES/Delorean/blob/master/delorean.py
- http://arstechnica.com/security/2015/10/new-attacks-on-network-time-protocol-can-defeat-https-and-create-chaos/
- http://dustint.com/post/12/cookie-injection-using-greasemonkey