COMP2207 Network Report

Huw Jones
27618153

October 29, 2016

1 Scan Results

1.1 HTTP

The HTTP service is the **Hyper Text Transfer Protocol**. It is the application layer that drives the majority of websites worldwide. It can use any port, but the most common port is `80/tcp`; the next most common port is `8080/TCP`.

There are a plethora of application servers that run the HTT Protocol. The most notable are Apache (httpd), NGINX (nginx), Internet Information Services (IIS - Microsoft). More recently, Javascript based services (Node.js) have joined the HTTP scene.

There are 25 hosts listening on `80/tcp` and 1 host listening on `8080/tcp`. The host listening on `8080/tcp` is also listening on `80/tcp`. Please note I've excluded hosts running HTTP on `443/tcp` as this setup is normally to inform users connecting via HTTP that they are connecting to an HTTPS port, so use HTTPS.

1.2 HTTPS

The HTTPS service is the **Hyper Text Transfer Protocol - Secure**. It is a **Secure Socket Layer** (SSL - now deprecated)/**TLS** (Transport Layer Security) wrapper for the original HTTP protocol. Both SSL and TLS use symmetric encryption to secure the communication. This provides more security for sensitive information (e.g.: passwords/bank details), whilst also increasing data integrity by effectively eliminating MITM attacks (see below). HTTPS normally uses `443/tcp` for servers to listen on.

Most HTTP servers include the capability to run HTTPS out of the box. However, HTTPS requires the use of public key authentication to validate the identity of the server party. Public key authentication is also used to mitigate Man-In-The-Middle (MITM) attacks where a 3rd party attempts to intercept a secure communication channel in order to capture sensitive data, cookies, or to send false requests whilst pretending to be the client.

HTTPS therefore requires the server to have a public key certificate to validate the private key of the server. It is the responsibility of the server admin to install the public certificate. Most organisations will charge for public certificates, however a group called letsencrypt.org are promoting the use of widespread HTTPS by offering free certificates. These certificates are much more secure than Self-Signed certificates, however they do not validate the server party as well as an enterprise-grade certificate from a reputable company (e.g.:) VeriSign.

Overall, there are 8 hosts listening on `443/tcp` running an HTTPS server. All 8 hosts are also listening on `80/tcp` running an HTTP server.

1.3 VOIP Phones (H.323)

Many VOIP Phones use `1720/tcp` for H.323. H.323 is an ITU (International Telecommunications Union) standard for Packet Based VOIP Phone systems.

There are 464 hosts listening on `1720/tcp`.

1.4 SSH

SSH stands for **Secure SHell**. SSH is a method for securely interacting with a remote host. The most common use of SSH is to login and manage hosts (mainly servers). It also has the ability to act as a network bridge by bridging ports between the two hosts. An extension to SSH, SCP (Secure CoPy), allows files to be transferred between the two hosts via an SSH tunnel.

SSH tunnels and port bridging can be used to secure network communications from external activity. For example, a SQL server could be behind a firewall that blocks the SQL port. The remote user could use an open SSH server behind the firewall as a jump box to forward SQL traffic between the SQL server and the remote user's computer.

Out of the /23 subnet scanned, only 1 host had `22/tcp` open and was listening for SSH connections.

2 Analysis

2.1 HTTP/HTTPS

2.1.1 Apache (httpd)

The most stable release of Apache is the 2.4.x branch. Only 9 out of 21 hosts running Apache are on the stable release. This means that is highly doubtful that any of the vulnerabilities in previous versions have been patched.

The oldest release of Apache in use is 2.2.14. There are 27 publicly listed vulnerabilities for this version, most of which would cause the remote host to go down (Denial of Service). Other vulnerabilities allow easy Cross Site Scripting (XSS).

However, most vulnerabilities in Apache are to do with the shear amount of modules that are bundled in the package.

2.1.2 NGINX

The most recent (stable) release of NGINX is 1.10.2. One host is running version 1.1.19, and another host is running 1.4.6.

Version 1.1.19 was released in April 2012 and version 1.4.6 was released March 2014. There are 11 publicly known vulnerabilities of NGINX v1.1.19. Of those 11, some are also applicable to v1.4.6

A majority of the vulnerabilities are Denial of Service attacks. There also seems to be a flaw in the NGINX DNS Resolver in versions prior to 1.9.10 that allows specially crafted DNS CNAME responses to DOS the server due to a buggy parser/handler. Some vulnerabilities allow arbitrary code to be executed on the host via specially crafted packets (using the `SPDY` handler). One vulnerability allows remote clients to bypass some restrictions via an unescaped space character in a URI.

2.2 SSH

The nmap scan indicated that the one host listening on `22/tcp` was running OpenSSH 5.3 and listening for SSH 2.0. The most recent release of OpenSSH is 7.3 (released August 2016).

CVE details list 8 vulnerabilities for OpenSSH v5.3. A majority of the vulnerabilities are Denial of Service attacks by exploiting buffer overflows, or creating enough new TCP connections to overload the service. Other vulnerabilities make use of specially crafted packets in order to get the OpenSSH server to send sensitive information stored in memory.

3 Shortcomings

Network port scans are an easy way to produce a list of hosts and their potential flaws. But, the cannot show the entire picture of an internal network's security. In this report, several terms will be used, they are defined as follows. **Network Penetration Test** (Pen Test), the act of assessing a network for potential flaws and vulnerabilities. **Network Port Scan** (NPS), a scan of host(s) by inspecting open/closed ports and seeing the responses from those ports. **Compromised Host**, a host which is not performing in its intended use/the way it was configured.

An NPS can be restricted by firewalls. Many network contain firewalls and routers. This allows the network to be segmented into subnets which in turn allow easy, granular access control lists to be set up. In combination with `802.1Q` (VLAN tagging) and `802.1X` (Network Authentication), these subnets can be easily managed and grant granular access controls (ACLs).

By placing a router/firewall (possibly in combination on one physical machine) between subnets, the traffic type can be shaped and restricted. This also allows for a stateful firewall that can detect unusual behaviour and react accordingly (e.g.: terminating the existing connection, or dropping packets from that host).

Although useful for, say, controlling access to different servers for different departments (e.g.: Accounts and Sales), an NPS from both the Accounts and Sales subnet could yield completely different results. In this case, there are effectively have two separate networks that each have their own vulnerabilities, but, since they are interconnected, they can be abused and manipulated to effectively negate the protection that the firewall/routing is providing.

Due to network segregation, an NPS in one subnet may not reveal anything of interest in other subnets (due to firewall/routing ACLs). For a naive tester, this could mean a lot of potential issues would be missed, thereby leaving the network still at risk.

An NPS cannot always show if a machine is compromised. An NPS can indicate that a host has been compromised by a remote attacker (e.g.: DOS). If the target port is scanned and open, then 5 minutes later scanned and is down - if nothing has changed in terms of firewalls, then it could be assumed that the host is being DOSed.

On the other hand, a host that was brute for dictionary scanned by an attacker for admin credentials could show next to no signs of compromise from an NPS. If the attacker did not tweak the host firewall configuration, or didn't open any new ports, then an NPS would not show any changes.

An NPS cannot show if the network has been infected by a Trojan Human A suit, a clipboard, a legitimate looking ID badge, and a very convincing sell could all that could be needed to physically gain access to a network. Through the use of social engineering, a good trojan could quite easily gain access to the server/data room. If they do, it's game over.

An NPS could never show this attack until it is too late.

An NPS should only ever be to form part of a Pen Test. As shown by the examples above, an NPS cannot show the full picture of a network's health. Yes, it can be used to detect services failing (from DOS/hardware failure/software failure). However in the complex environment of an `802.1Q/802.1.X` environment, an NPS would need to be performed from different angles across all routers and firewalls in order to be able to build a full picture of the network topology.

Overall, a Network Scan cannot show the full picture when it comes to the health of a network.