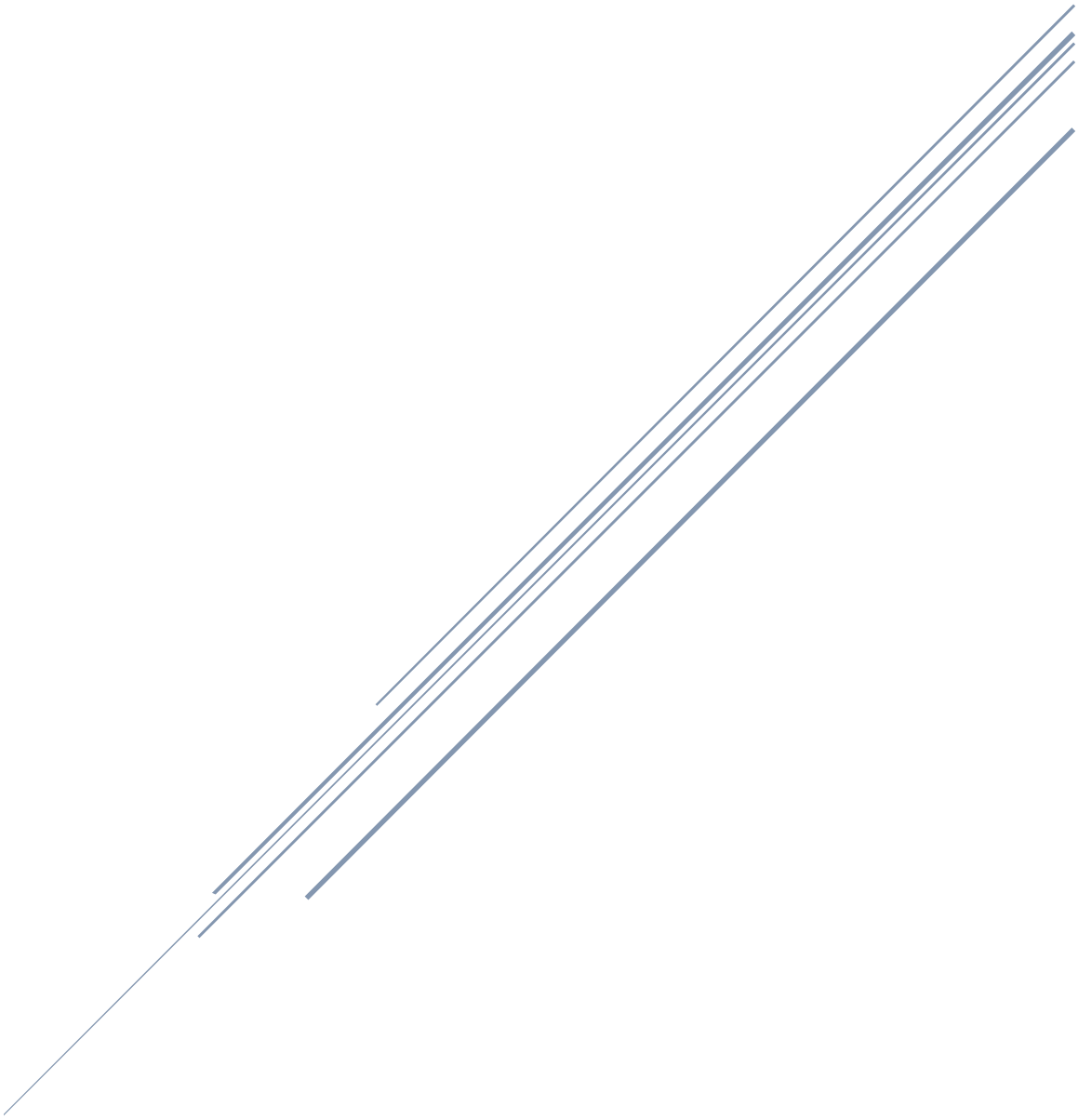
SERVER/VM NESSUS AND OPEN VAS REPORTS



Cole Baker

Table of Contents

Introduction

This report is to help us better understand how to use different vulnerability scanners and how they interact with different operating systems. As well as test the security of the servers created during this course and determine if any vulnerabilities exist on the servers. As well as understanding those vulnerabilities if any exist.

Task 1 – Vulnerability Scans

To conduct my scans, I choice to use NESSUS and OpenVAS as the two different vulnerability scanners for this report.

NESSUS

Windows Server

Screenshot of the NESSUS scan for the Windows Server 2016 after the scan was completed successfully, as well as a summary of the vulnerabilities found which can be found below in more detail.

Linux Server

Screenshot of the NESSUS scan for the Linux Ubuntu Server after the scan was completed successfully, as well as a summary of the vulnerabilities found which can be found below in more detail.

PFSense

Screenshot of the NESSUS scan for PFSense after the scan was completed successfully, as well as a summary of the vulnerabilities found which can be found below in more detail.

OpenVAS

Windows Server

I was unable to complete a scan of the Windows Server using OpenVAS due to an unknow reason of the Kali Linux VM being able to see the server.

Linux Server

Screenshot of the OpenVAS scan for PFSense after the scan was completed successfully as well as a summary of the vulnerabilities found which can be found below in more detail.

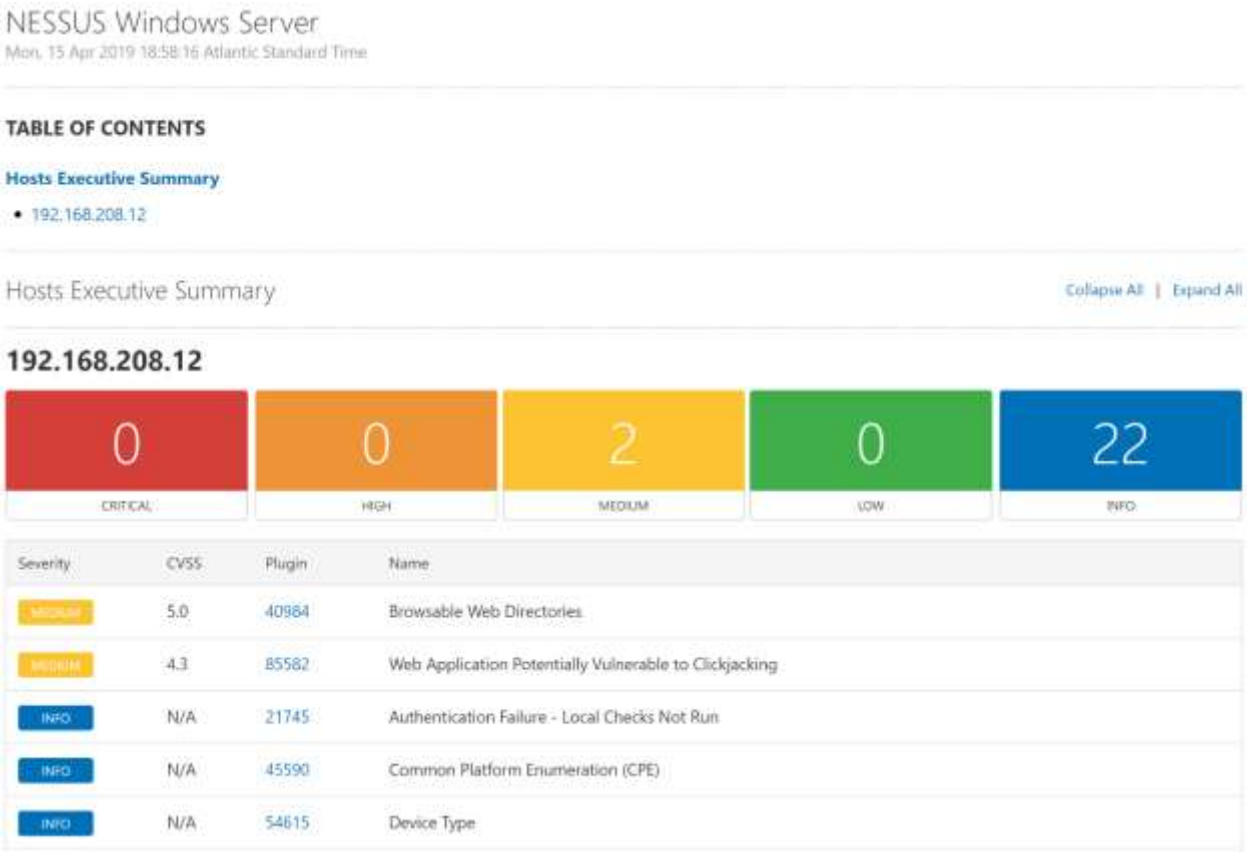
PFSense

Screenshot of the OpenVAS scan for PFSense after the scan was completed successfully as well as a summary of the vulnerabilities found which can be found below in more detail.

Task 2

NESSUS Final Vulnerability Reports

Windows Server



NESSUS Windows Server

Mon, 15 Apr 2019 18:58:16 Atlantic Standard Time

TABLE OF CONTENTS

Hosts Executive Summary

- 192.168.208.12

Hosts Executive Summary — Collapse All | Expand All

192.168.208.12

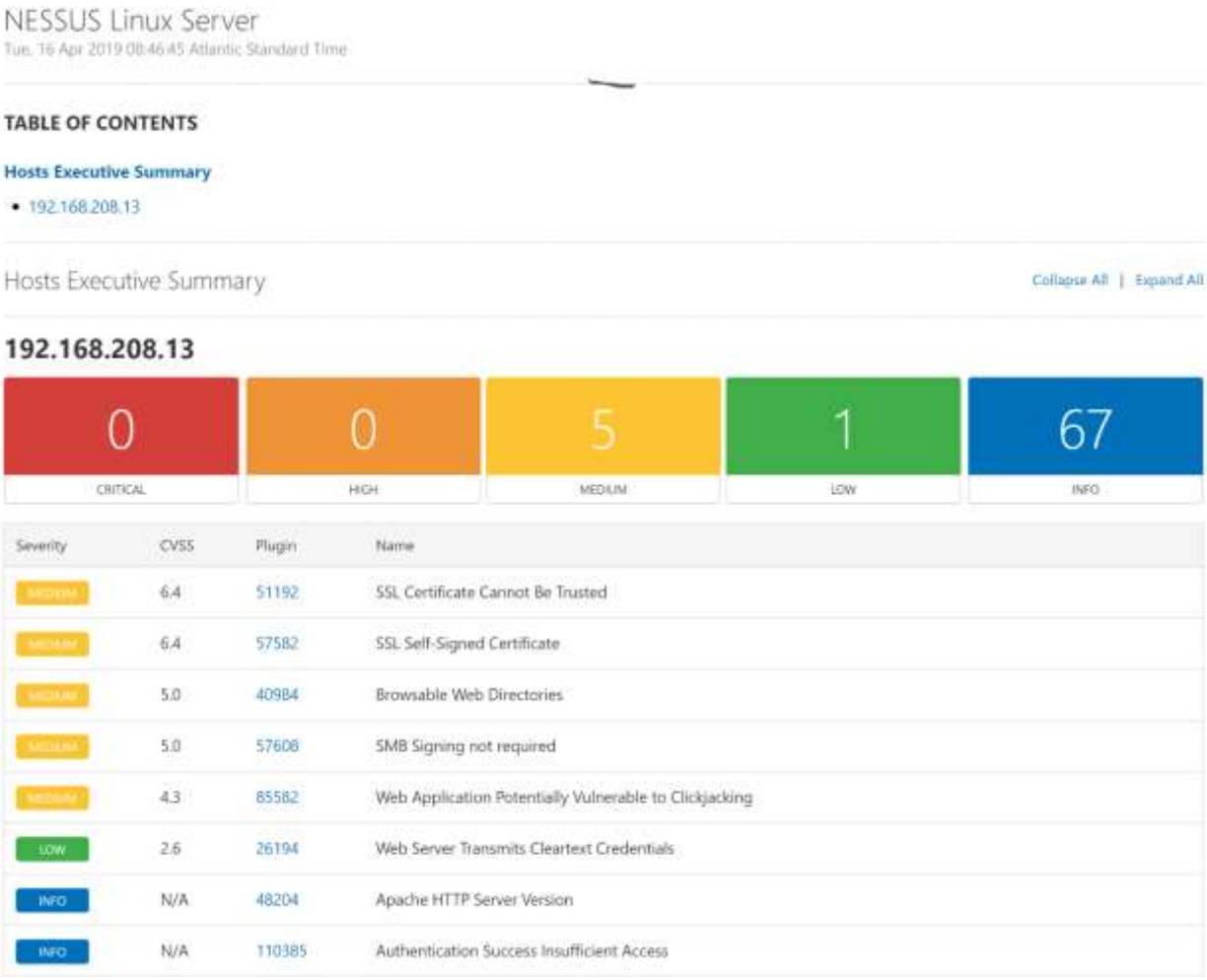
0	0	2	0	22
CRITICAL	HIGH	MEDIUM	LOW	INFO

Severity	CVSS	Plugin	Name
MEDIUM	5.0	40984	Browsable Web Directories
MEDIUM	4.3	85582	Web Application Potentially Vulnerable to Clickjacking
INFO	N/A	21745	Authentication Failure - Local Checks Not Run
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	54615	Device Type

The Windows Server contained 2 medium ranged vulnerabilities, Browsable Web Directories and Web Applications Potentially Vulnerable to Clickjacking. The first vulnerability would give the ability to browse different directories on the server through remote access. The only real solution to this vulnerability is to increase the restrictions on the directories as well as keep sensitive information out of those directories. This vulnerability was rated at a 5.0 meaning it would be more important to fix.

The second medium vulnerability found would give certain web applications the ability to exploit vulnerabilities. The server would not be able to mitigate certain vulnerabilities. The server does not set an X-Frame-Options response header or a Content-Security-Policy 'frame-ancestors' response header in all content responses. This could potentially expose the site to a clickjacking or UI redress attack. This can be easily mitigated by creating an X-Frame-Options so that the server will not respawn to the requests from these applications.

Linux Server



NESSUS Linux Server

Tue, 16 Apr 2019 08:46:45 Atlantic Standard Time

TABLE OF CONTENTS

Hosts Executive Summary

- 192.168.208.13

Hosts Executive Summary

Collapse All | Expand All

192.168.208.13

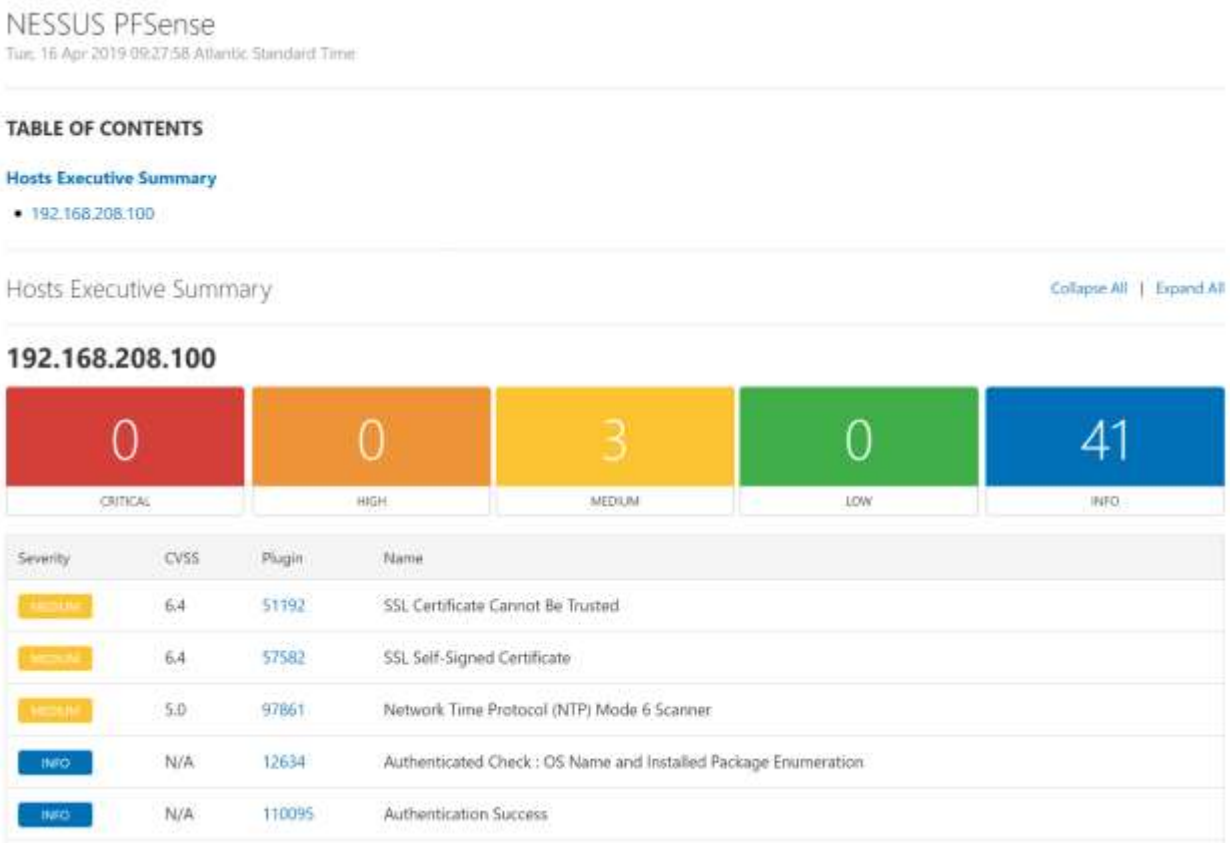
0	0	5	1	67
CRITICAL	HIGH	MEDIUM	LOW	INFO

Severity	CVSS	Plugin	Name
MEDIUM	6.4	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.4	57582	SSL Self-Signed Certificate
MEDIUM	5.0	40984	Browsable Web Directories
MEDIUM	5.0	57608	SMB Signing not required
MEDIUM	4.3	85582	Web Application Potentially Vulnerable to Clickjacking
LOW	2.6	26194	Web Server Transmits Cleartext Credentials
INFO	N/A	48204	Apache HTTP Server Version
INFO	N/A	110385	Authentication Success Insufficient Access

The Linux Server contained a larger amount of vulnerabilities, so I went through the higher ranked vulnerabilities, the two 6.4 due to there severity. This server also contained a few of the same vulnerabilities as the Windows Server. The first vulnerability is the SSL Certificate Cannot Be Trusted. Which directly correlates to the second vulnerability of the SSL Certificate being self-signed. Creating or purchasing a proper signed certificate would fix both of these vulnerabilities. With just a self-sign certificate, the browser to believe the website is unsecure and will warn the user to go back from the webpage.

Another vulnerability that would be worth noting, would be the low ranked vulnerability, this vulnerability allows the Web Server to transmit some credentials in cleartext. So, any attacker watching the network traffic may be able to intercept any credentials used over http. The best fix would be to only transmit sensitive credentials over https.

PFSense



NESSUS PFSense

Tue, 16 Apr 2019 09:27:58 Atlantic Standard Time

TABLE OF CONTENTS

Hosts Executive Summary

- 192.168.208.100

Hosts Executive Summary — Collapse All | Expand All

192.168.208.100

0	0	3	0	41
CRITICAL	HIGH	MEDIUM	LOW	INFO

Severity	CVSS	Plugin	Name
MEDIUM	6.4	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.4	57582	SSL Self-Signed Certificate
MEDIUM	5.0	97861	Network Time Protocol (NTP) Mode 6 Scanner
INFO	N/A	12634	Authenticated Check : OS Name and Installed Package Enumeration
INFO	N/A	110095	Authentication Success

The PFSense Server contains almost the same medium vulnerabilities as the Linux Server with the exception of the 5.0 Network Time Protocol (NTP) Mode 6 Scanner. As far as the two higher ranked vulnerabilities, they have the same reason for being considered a vulnerability and have the same mitigation methods.

The Network Time Protocol vulnerability can be used to respond to mode 6 queries. Servers that respond to these queries have the potential to be used in NTP amplification attacks. The only way to mitigate this vulnerability is to restrict NTP mode 6 queries to reduce the possibility of this being exploited by attackers.

OpenVAS

Windows Server

I was unable to complete a scan of the Windows Server using OpenVAS due to an unknow reason of the Kali Linux VM being able to see the server.

Linux Server

Host Summary

Host	Start	End	High	Medium	Low	Log	False Positive
192.168.208.13	Apr 19, 01:05:33	Apr 19, 01:33:56	0	0	1	0	0
Total: 1			0	0	1	0	0

Host Authentications

Host	Protocol	Result	Port/User
192.168.208.13	SSH	Success	Protocol SSH, Port 22, User cpawar
192.168.208.13	SMB	Success	Protocol SMB, Port 445, User

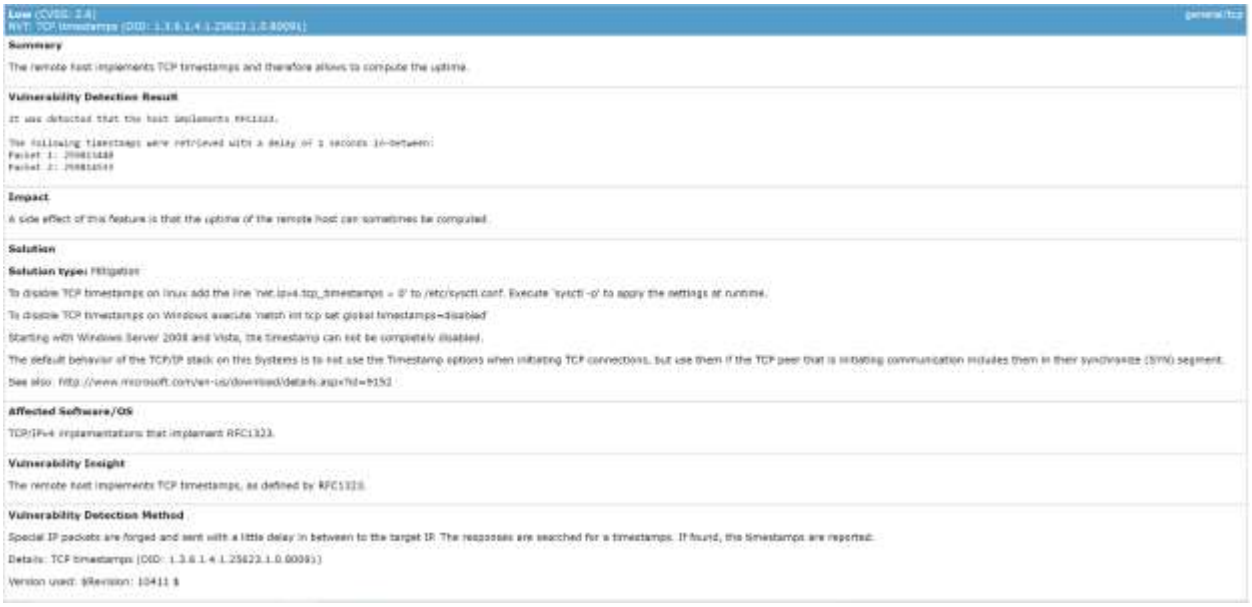
Results per Host

Host 192.168.208.13

Scanning of this host started at: Fri Apr 19 01:05:33 2019 UTC
Number of results: 1

Port Summary for Host 192.168.208.13

Service (Port)	Threat Level
general/tcp	Low



Low (CVSS: 2.6)
NVT: TCP timestamps (OID: 1.3.6.1.4.1.25623.1.0.80091)

Summary

The remote host implements TCP timestamps and therefore allows to compute the uptime.

Vulnerability Detection Result

It was detected that the host implements RFC1323.

The following timestamps were retrieved with a delay of 1 seconds in-between:
Packet 1: 2998513688
Packet 2: 2998513777

Impact

A side effect of this feature is that the uptime of the remote host can sometimes be computed.

Solution

Solution type: Mitigation

To disable TCP timestamps on linux add the line 'net.ipv4.tcp_timestamps = 0' to /etc/sysctl.conf. Execute 'sysctl -p' to apply the settings at runtime.

To disable TCP timestamps on Windows execute 'netsh int tcp set global timestamps=disabled'

Starting with Windows Server 2008 and Vista, the timestamp can not be completely disabled.

The default behavior of the TCP/IP stack on this Systems is to not use the Timestamp options when initiating TCP connections, but use them if the TCP peer that is initiating communication includes them in their synchronize (SYN) segment.

See also: http://www.microsoft.com/en-us/download/details.aspx?id=9152

Affected Software/OS

TCP/IPv4 implementations that implement RFC1323.

Vulnerability Insight

The remote host implements TCP timestamps, as defined by RFC1323.

Vulnerability Detection Method

Special IP packets are forged and sent with a little delay in between to the target IP. The responses are searched for a timestamps. If found, the timestamps are reported.

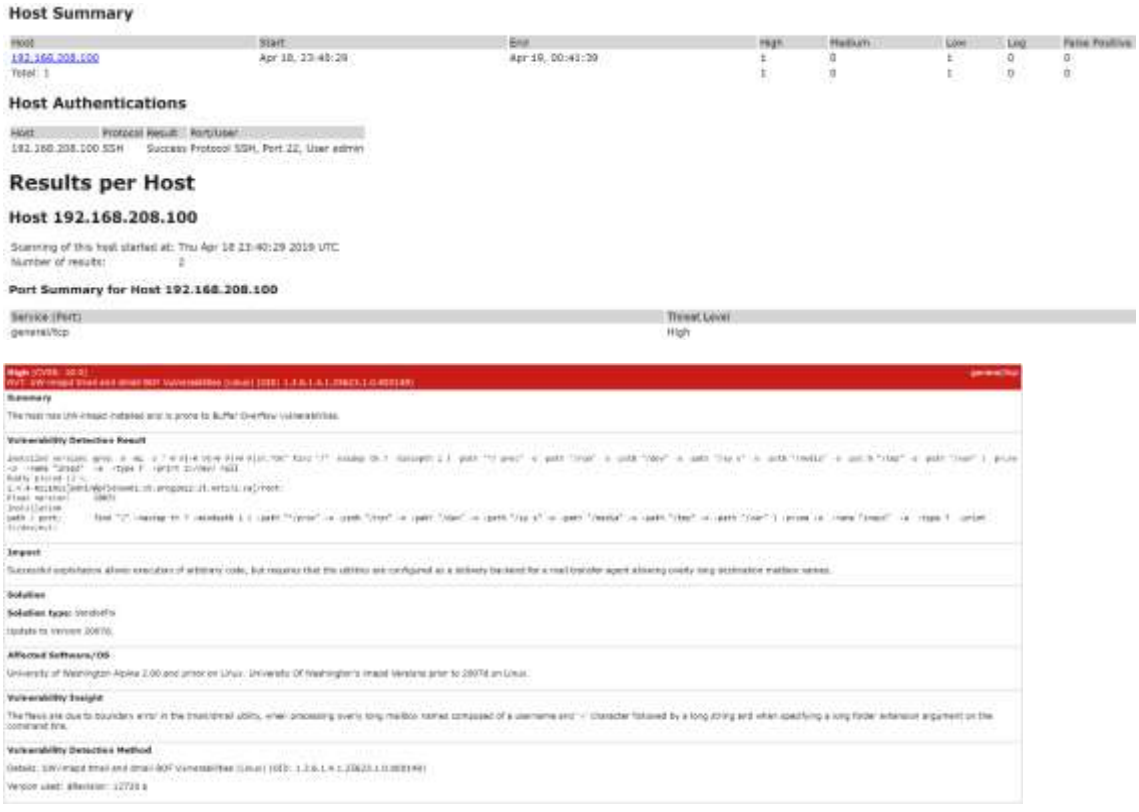
Details: TCP timestamps (OID: 1.3.6.1.4.1.25623.1.0.80091)

Version used: $Revision: 10411 $

During the scan of the Linux Server using OpenVAS, OpenVAS was only able to find one vulnerability even after running the scan multiple times and using different scan modes.

Even though only one vulnerability was found and was found to be low. This vulnerability can still be used by an attacker. The vulnerability is defined as TCP Timestamps. This vulnerability would allow an attacker to compute the uptime of a specific server. This could be used to tell when the server is being accessed or monitored which would help an attacker plan attack. This can be easily mitigated by disabling the TCP stamp on the server which would not effect the servers operations.

PFSense



Host Summary

Host	Start	End	High	Medium	Low	Log	False Positive
192.168.208.100	Apr 18, 23:40:29	Apr 19, 00:41:39	1	0	1	0	0
Total: 1			1	0	1	0	0

Host Authentications

Host	Protocol	Result	Port/User
192.168.208.100	SSH	Success	Protocol SSH, Port 22, User admin

Results per Host

Host 192.168.208.100

Scanning of this host started at: Thu Apr 18 23:40:29 2019 UTC
Number of results: 2

Port Summary for Host 192.168.208.100

Service (Port)	Threat Level
general/tcp	High

High (CVSS: 10.0)
NVT: SMTP/IMAP Email and Email-BOF Vulnerabilities (Linux) (OID: 1.3.6.1.4.1.25623.1.0.803185)

Summary

The host has SMTP/IMAP installed and is prone to Buffer Overflow vulnerabilities.

Vulnerability Detection Result

Impact

Successful exploitation allows execution of arbitrary code, but requires that the utilities are configured as a delivery backend for a mail/transfer agent allowing overly long destination mailbox names.

Solution

Solution type: Vendor Fix

Update to version 2007d.

Affected Software/OS

University of Washington Alpine 2.00 and prior on Linux. University Of Washington's imapd Versions prior to 2007d on Linux.

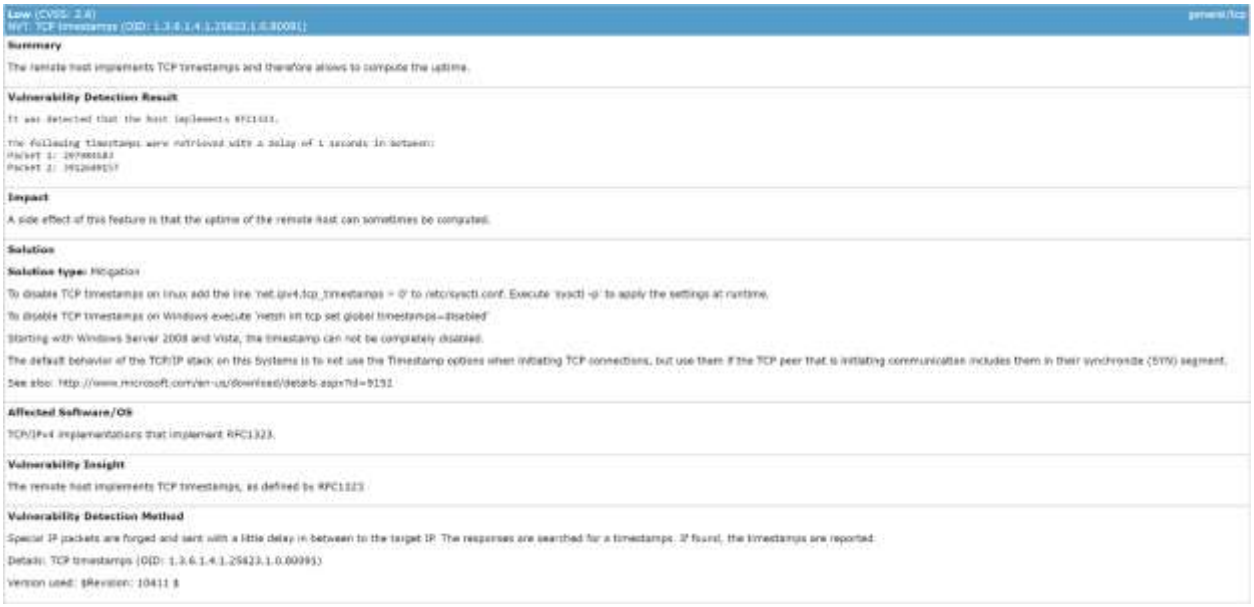
Vulnerability Insight

The flaws are due to boundary error in the tmail/dmail utility, when processing overly long mailbox names composed of a username and '+' character followed by a long string and when specifying a long folder extension argument on the command line.

Vulnerability Detection Method

Details: SMTP/IMAP Email and Email-BOF Vulnerabilities (Linux) (OID: 1.3.6.1.4.1.25623.1.0.803185)

Version used: $Revision: 12720 $



After running a scan on the PFSense Server with OpenVAS. The scan reported two main vulnerabilities, one high and one low. The low vulnerability is the same one found on the Linux Server as well as possesses the same mitigation. The higher ranked vulnerability is UW-imapd tmail and dmail BOF Vulnerabilities. Which an attacker used to implement a buffer overflow attack which can be exploited through connecting to the remote host and running code. This can be easily fixed by updating the service this vulnerability is found in to the newest version.

Task 3

Mitigation Summary

Throughout the creation of these servers, I have taken many steps to reduce the attack surface of each server and reduce the number of exploitable vulnerabilities. Starting with the Window Server 2016, the first step I took was to create different users, so that the server did not have to run using the admin account every time to reduce the unnecessary privileges, as well as created a backup admin account. Next was to make sure the system was up to date to patch any old vulnerabilities. The local security policies were set for passwords to keep passwords up to date and prevented from using old passwords. The system was also set to audit login activity which would help to mitigate against unauthorized logins and failed logins. Ports 137, 138 and 139 were disabled and blocked in the firewall to reduce the attack surface and also would prevent any vulnerabilities using the port from being exploited. The .NET Trust Levels were increased, if they had not been, a vulnerability for that service would have appeared in the vulnerability scans.

Once the Linux server was created, I started by changing all default passwords to avoid any exploit that would have used the default admin password to exploit something. The website created was then added to the server as well as a WordPress site.

The browser was set to clear history and passwords to prevent attackers from being able to exploit the browser and gain access to this information. Next SSL was installed on the server to create a self-signed certificate. Yes, self-signed certificates do possess some vulnerabilities, but they are more secure then having no certificate, so having one did mitigate some potential vulnerabilities.

After finishing with the Linux Server, the PFSense Server was configured to connect only through certain LANS and WANS. Snort was then installed, which is an intrusion prevention and detection system which would help monitor the known vulnerabilities and reduce them by preventing them from being exploited. Next OpenVPN was installed, which would help with any vulnerabilities that need to connect to the server remotely because the VPN would reroute their request. Then the Squid Proxy server was added to the server which allowed me to install and run a firewall-antivirus service to reduce the possibility of a vulnerability being exploited.

References

All references used were either from the reports provided by the final NESSUS and OpenVAS scans which contained information regarding each vulnerability. Or finished class assignments to reflect on how a certain vulnerability could have been mitigated and to reflect on what was learned.