PROJECT (HOME-LAB)

ON

VULNERABILITY ASSESSMENT AND MANAGEMENT USING TENABLE NESSUS

AUTHOR – PRATIK ERANDE

(pratikerande995@gmail.com)

(https://www.linkedin.com/in/pratik-erande-74b318279)

DATE – 20/01/2025

INTRODUCTION

VULNERABILITY ASSESSMENT AND MANAGEMENT - Vulnerability Assessment and Management (VAM) is a crucial component of an organization's overall cybersecurity strategy, focused on identifying, evaluating, prioritizing, and remediating security weaknesses across IT systems, applications, and networks. Vulnerabilities, which are security flaws or gaps in software, hardware, and systems, can be exploited by attackers to compromise the confidentiality, integrity or availability (CIA) of the targeted system or to gain unauthorized access, disrupt services, or steal sensitive information from a targeted systems or networks. Vulnerability assessment and management is the process of continuously identifying and mitigating these vulnerabilities to reduce the risk of a successful cyber attack and improve the security posture of the organization.

The primary objectives of vulnerability assessment and management includes:

1. **Identify Vulnerabilities**: Regularly scanning IT systems, applications, and networks to detect vulnerabilities, including those resulting from outdated software, misconfigurations, or unpatched systems.

2. **Assess and Prioritize Risks**: Evaluate the severity of vulnerabilities based on their potential impact and exploitability. This helps prioritize remediation efforts, ensuring the most critical vulnerabilities are addressed first.

3. **Mitigate and Remediate Vulnerabilities**: Apply patches, configuration changes, and other fixes to address identified vulnerabilities, reducing the likelihood of exploitation.

4. **Monitor and Reassess**: Continuously monitor the security environment to identify new vulnerabilities and re-assess the effectiveness of previous remediation efforts, ensuring ongoing protection.

5. **Ensure Compliance**: Support compliance with regulatory requirements (e.g., PCI-DSS, HIPAA, GDPR) by ensuring vulnerabilities are managed and remediated according to industry standards.

6. **Improve Security Posture**: Strengthen the organization's overall security framework by systematically addressing weaknesses and improving resilience against cyber threats.

VULNERABILITY ASSESSMENT PROCESS

(Stages of VA Process)

There are four stages involved in VA process including –

1. Vulnerability Identification:

This is the first step in the vulnerability assessment process. It involves scanning IT assets (like servers, networks, and applications) to detect potential security weaknesses. Tools and techniques such as automated vulnerability scanners, penetration testing, and manual reviews are used to identify vulnerabilities like outdated software, misconfigurations, and missing patches.

2. Vulnerability Assessment and Prioritization:

Once vulnerabilities are identified, they need to be evaluated based on their severity and potential impact on the organization. Vulnerabilities are categorized and scored using frameworks like CVSS (Common Vulnerability Scoring System) to assess their risk level. This helps in prioritizing which vulnerabilities to address first, focusing on those with the highest potential for exploitation or impact.

3. Vulnerability Remediation:

Remediation involves taking steps to fix the identified vulnerabilities. This can include applying patches, updating software, reconfiguring systems, or implementing additional security controls. The goal is to eliminate or mitigate the vulnerabilities to reduce the risk of exploitation. Timely and effective remediation is crucial to maintaining a strong security posture.

4. Verification and Continuous Monitoring:

After vulnerabilities are remediated, it's important to verify that the fixes have been successfully implemented and that the vulnerabilities no longer exist. This is done by re-scanning or testing the system. Continuous monitoring involves regularly checking for new vulnerabilities, scanning systems continuously, and reassessing the environment to stay ahead of emerging threats and ensure ongoing security.

PROJECT OVERVIEW

PROJECT PURPOSE

This project focuses on conducting vulnerability assessment using Nessus. Nessus is a essential tool to identify potential security weaknesses and vulnerabilities. We will perform scans on the vulnerable Windows 10 OS hosted using VMware. This will identify missing updates, misconfigurations, and other exploitable flaws that could be exploited by the malicious attacker to gain unauthorised access or compromise the security of the system. To make Windows 10 system vulnerable we will install some outdated softwares and make some misconfigurations.

The project aims to demonstrate the scans on vulnerable Windows 10 OS hosted in VMware. After performing scans, we'll assess those identified vulnerabilities based on their severity and their potential impact and perform patches of those vulnerabilities. After patching those vulnerabilities, we will do the verification of patched vulnerabilities and monitor for further findings.

PROJECT GOALS

1. Identify vulnerabilities in a Windows 10 VM using Nessus.
2. Prioritize identified vulnerabilities based on their severity and impact.
3. Apply patches to improve the system's security.
4. Demonstrate vulnerability management from detection to remediation.

REQUIREMENTS

1. **VMware Workstation Player**
   It provides a secure and isolated environment for conducting experiments and tests. By installing, it allows users to setup virtual machines to simulate a variety of operating environments while maintaining the security and integrity of the underlying system. The use of VMware ensures that any potential risks or consequences or consequences originating from vulnerability assessment activities are contained within virtual environment.

2. **Windows 10 OS**
   Setting up a vulnerable Windows 10 OS in VMware allows to simulate a real-world environment and will serve as the target machine where the vulnerabilities are identified and assessed. Additionally, to evaluate the security posture of a Windows-based OS.

3. **Nessus**
   Installing and configuring Nessus which is an open-source tool on local machine to perform vulnerability assessment is an important component of the project. Nessus is a remote security scanning tool which scans a computer and raises an alert if it discovers any weakness that malicious hackers could use to gain any computer connected to a network. Moreover, Nessus provides patching assistance to mitigate the detected vulnerabilities.

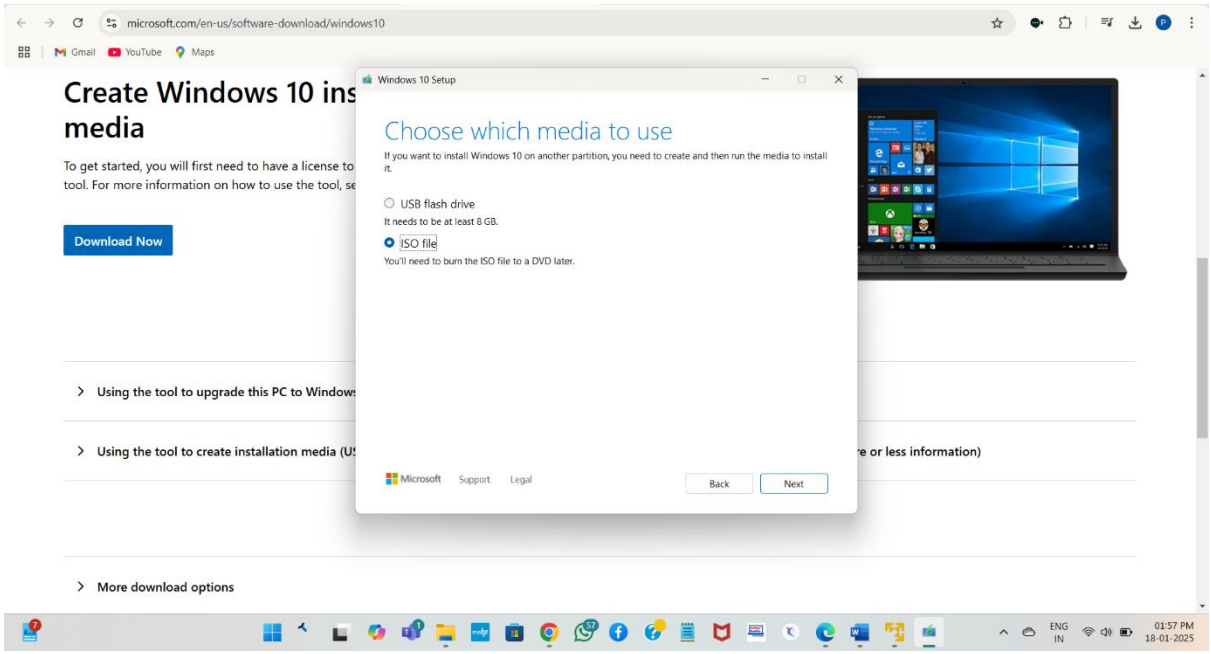
   We will install the free trial version of Tenable Nessus to demonstrate scans on our targeted Windows 10 OS.

IMPLEMENTATION

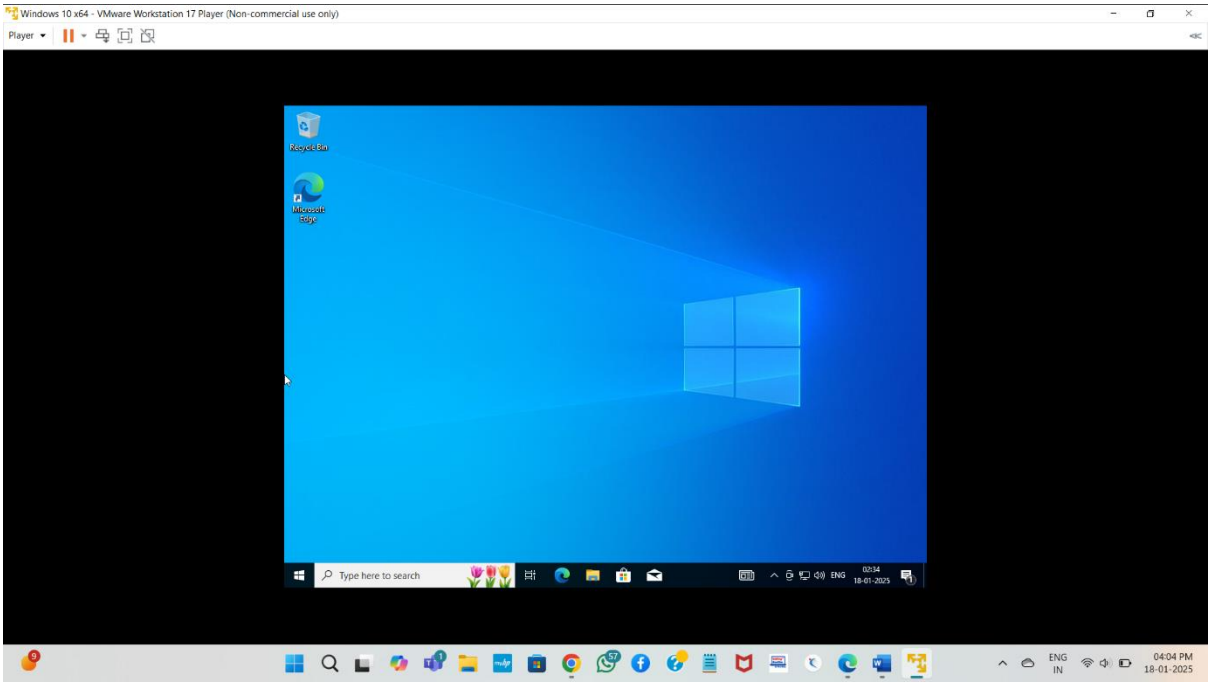
INSTALLING VMWARE WORKSTATION

Firstly, we will install VMware workstation to perform vulnerability scans in a sandbox or in controlled environment as well as to host Windows 10 OS. (Already installed on system)

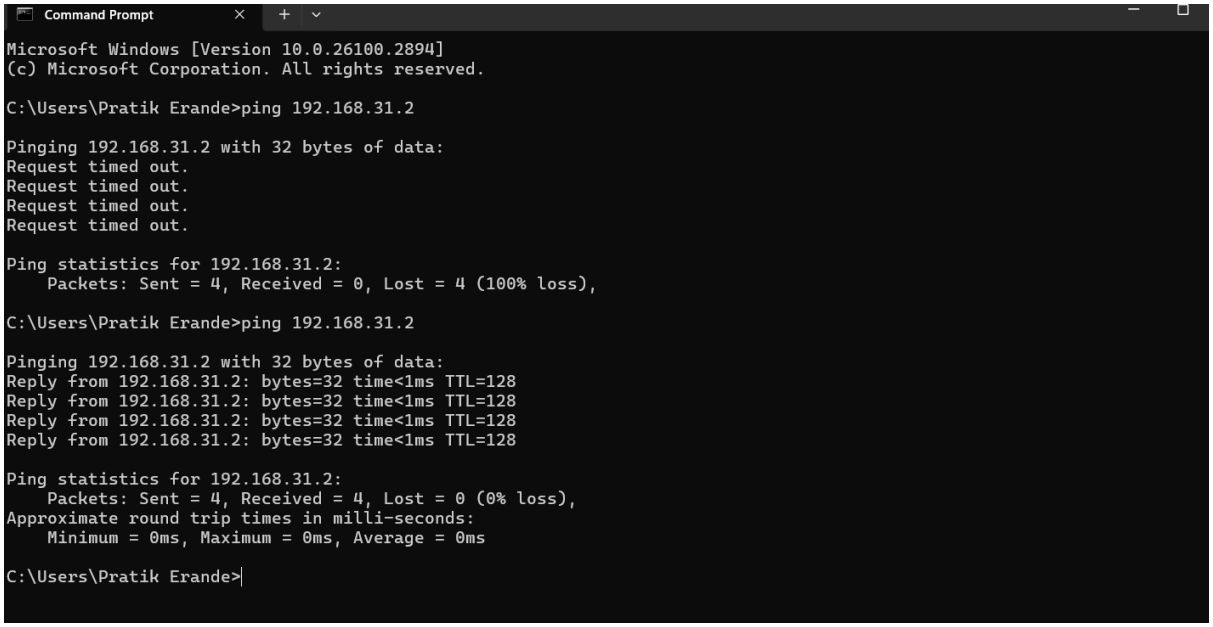
INSTALLING WINDOWS 10 ISO FILE



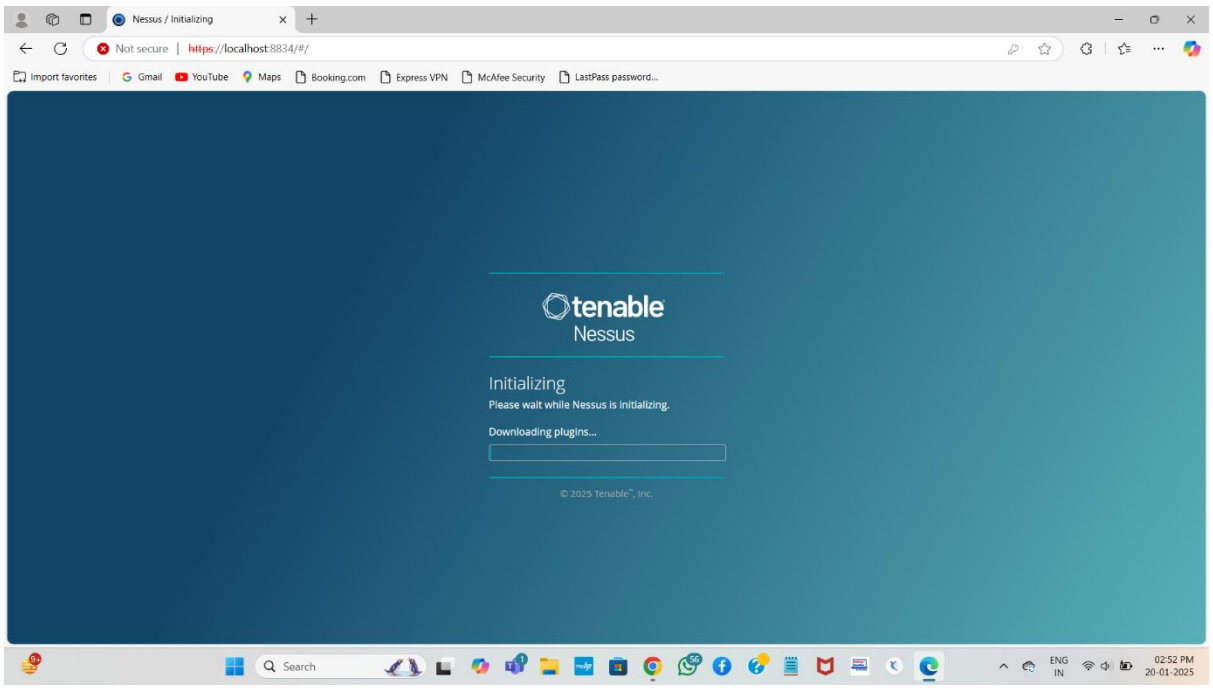
After installing Windows 10 ISO file we will install and setup it in VMware



Now, we have enabled connectivity between local machine and virtual machine (By disabling windows defender firewall of virtual machine) 192.168.31.2 is IP address of VM.

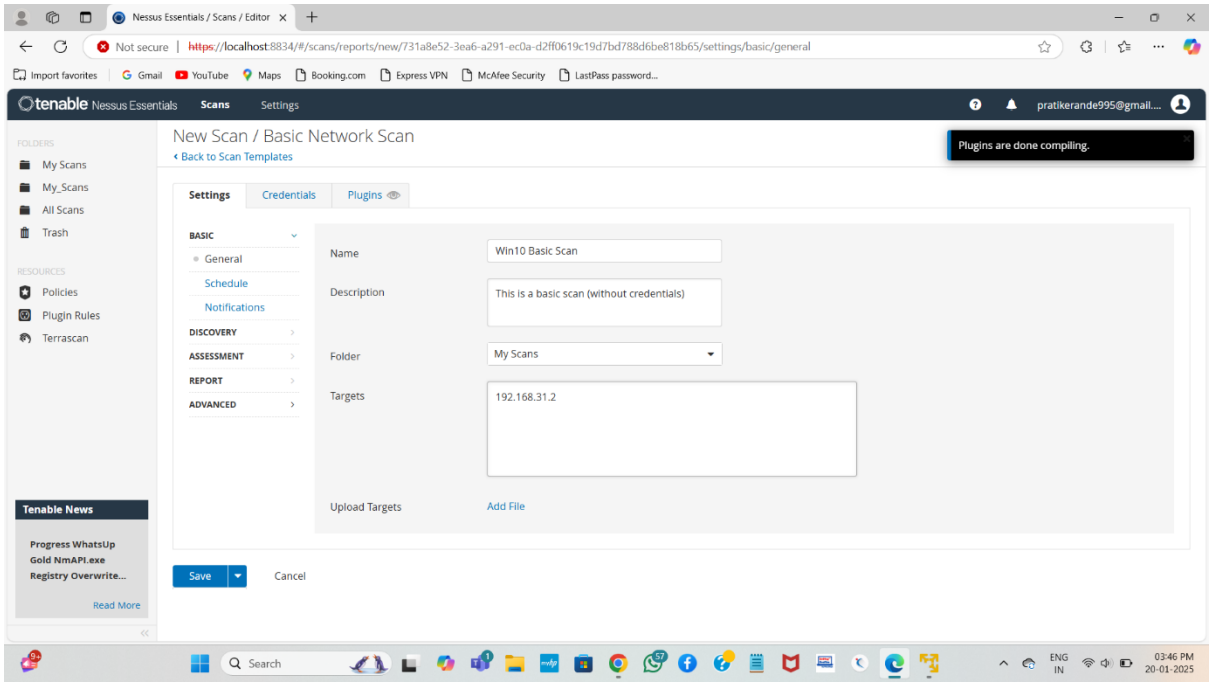


Now, we'll install and setup the trail version of Nessus

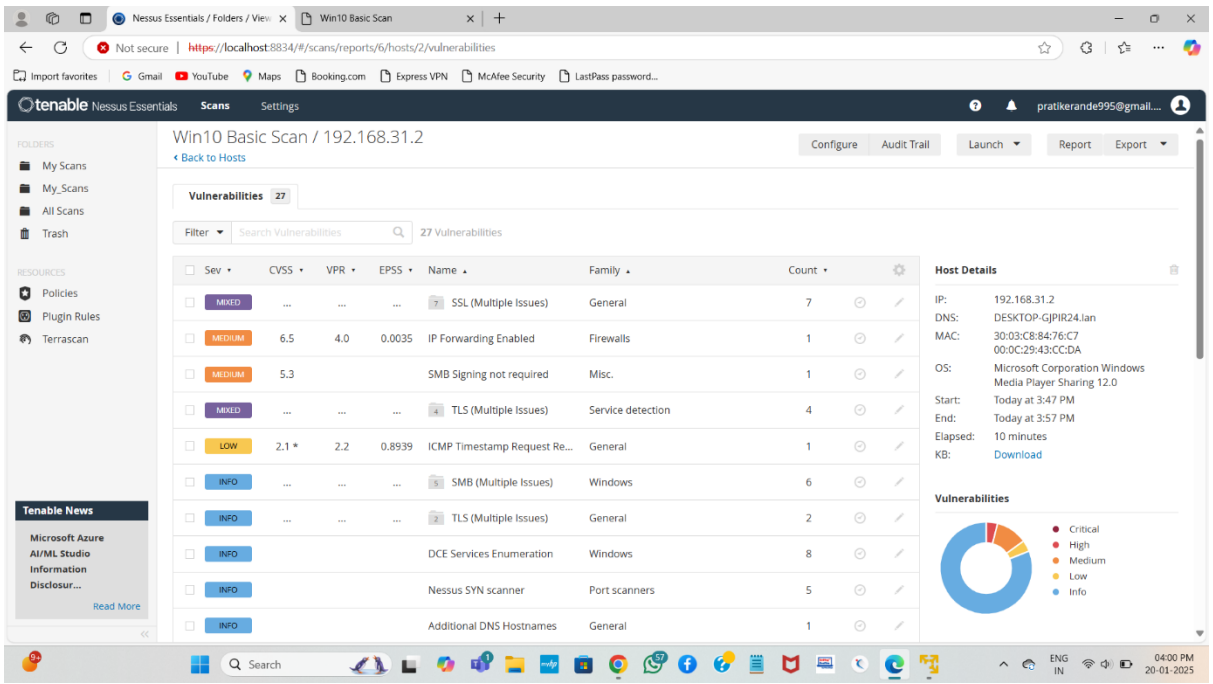


Now, at first we'll perform a basic scan (Non-Authenticated Scan) without giving target system credentials

Launch basic scan on targeted system



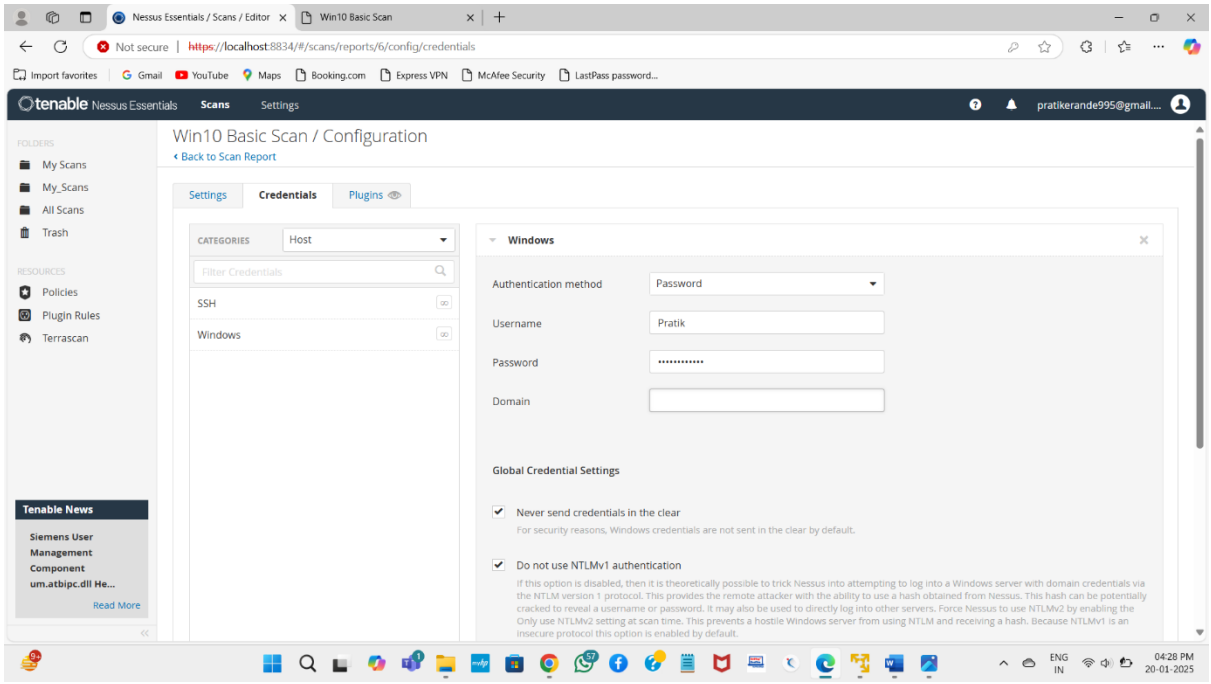
This basic scan gives few vulnerabilities as shown below



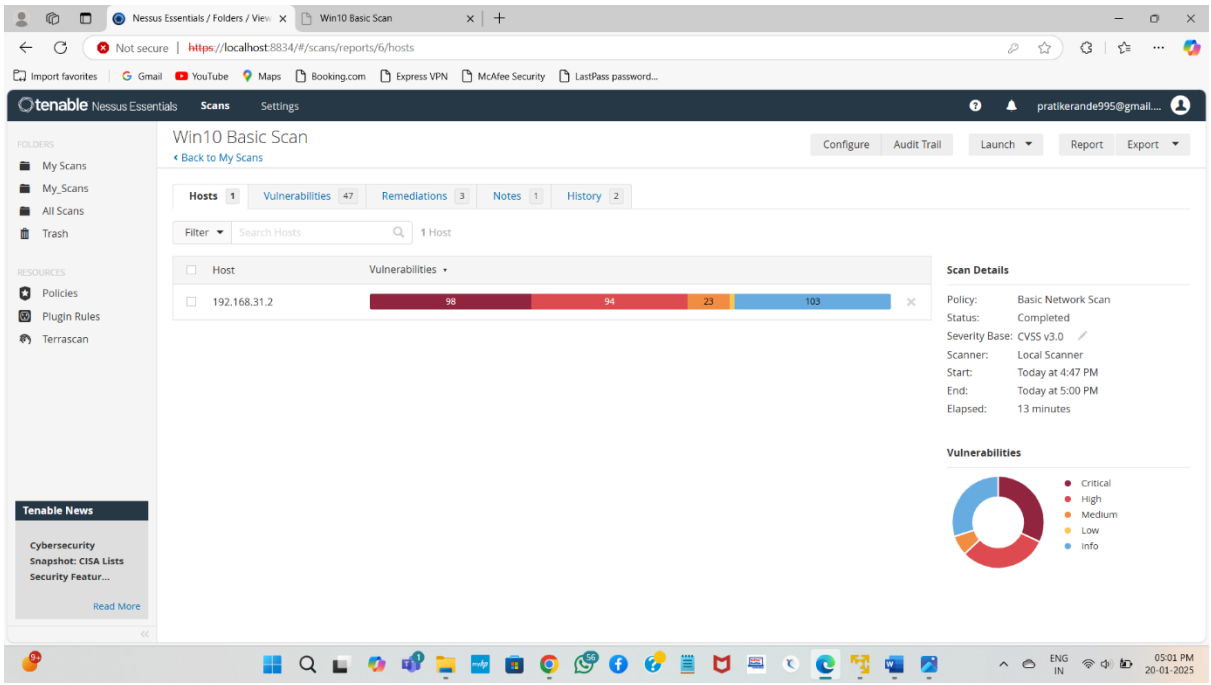
As per the report generated by Nessus, System includes 1 high, 6 medium and 1 low severity vulnerabilities. The high vulnerability has 7.5 CVSS score, 6 medium vulnerabilities have 6.5 CVSS score and low severity vulnerability has 2.1 CVSS score. CVSS (Common Vulnerability Scoring System) score defines the potential impact and criticality of the vulnerability. CVSS score helps to prioritize the high critical vulnerabilities.

Now, we'll install some outdated software on our targeted system (Windows 10) and make the system more vulnerable. Here we have installed older version of Mozilla Firefox.

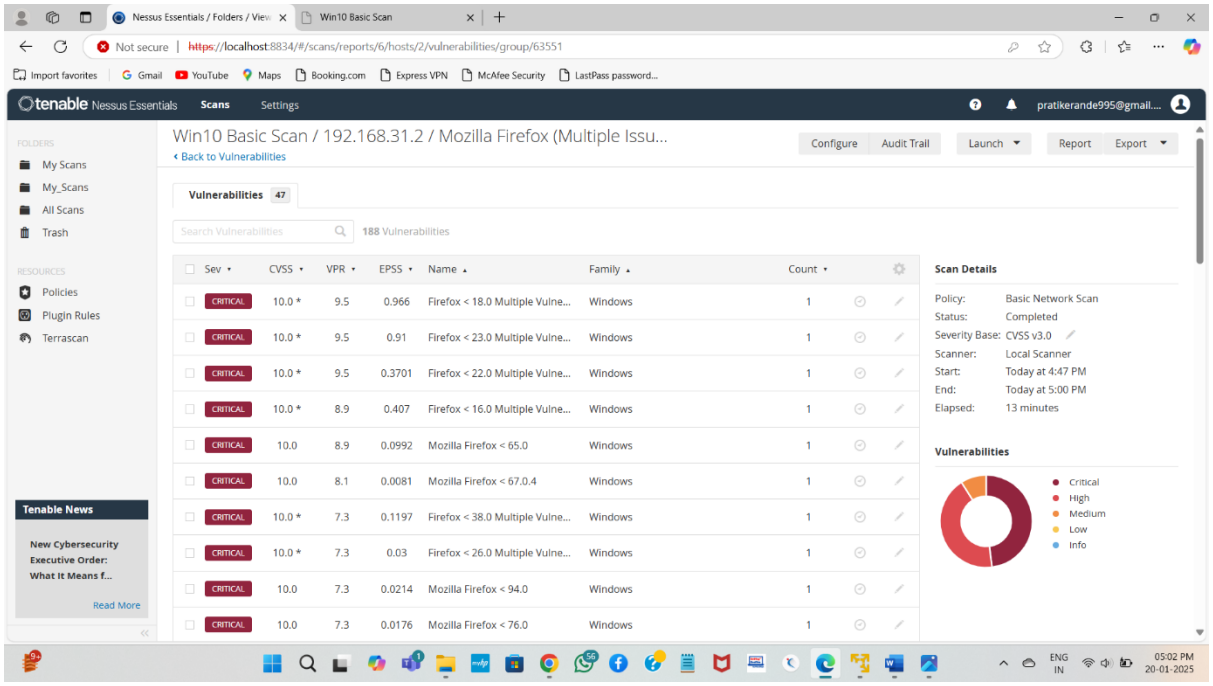
Now, we'll provide the credentials of out target machine to Nessus. An authenticated scan scans a system deeply by logging in.



After launching authenticated scan the no of vulnerabilities increased because we've installed outdated version of Firefox as well as authenticated scan scans the targeted system deeply by logging in.



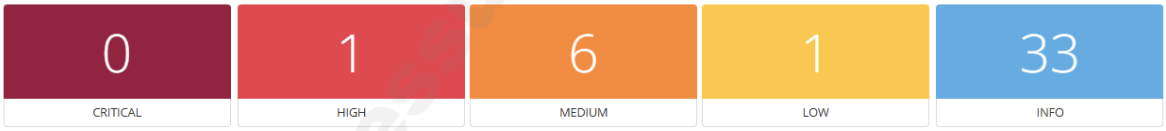
Most of the critical vulnerabilities are of Firefox


Nessus Essentials / Folders / View
Win10 Basic Scan
Not secure | https://localhost:8834/#/scans/reports/6/hosts/2/vulnerabilities/group/63551
Import favorites
Gmail
YouTube
Maps
Booking.com
Express VPN
McAfee Security
LastPass password...
tenable Nessus Essentials
Scans
Settings
pratikerande995@gmail....
FOLDERS
My Scans
My_Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Terrascan
Tenable News
New Cybersecurity Executive Order: What It Means f...
Read More
Win10 Basic Scan / 192.168.31.2 / Mozilla Firefox (Multiple Issu...
Back to Vulnerabilities
Configure
Audit Trail
Launch
Report
Export
Vulnerabilities 47
Search Vulnerabilities
188 Vulnerabilities
Sev
CVSS
VPR
EPSS
Name
Family
Count
CRITICAL 10.0 * 9.5 0.966 Firefox < 18.0 Multiple Vulne... Windows 1
CRITICAL 10.0 * 9.5 0.91 Firefox < 23.0 Multiple Vulne... Windows 1
CRITICAL 10.0 * 9.5 0.3701 Firefox < 22.0 Multiple Vulne... Windows 1
CRITICAL 10.0 * 8.9 0.407 Firefox < 16.0 Multiple Vulne... Windows 1
CRITICAL 10.0 8.9 0.0992 Mozilla Firefox < 65.0 Windows 1
CRITICAL 10.0 8.1 0.0081 Mozilla Firefox < 67.0.4 Windows 1
CRITICAL 10.0 * 7.3 0.1197 Firefox < 38.0 Multiple Vulne... Windows 1
CRITICAL 10.0 * 7.3 0.03 Firefox < 26.0 Multiple Vulne... Windows 1
CRITICAL 10.0 7.3 0.0214 Mozilla Firefox < 94.0 Windows 1
CRITICAL 10.0 7.3 0.0176 Mozilla Firefox < 76.0 Windows 1
Scan Details
Policy: Basic Network Scan
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: Today at 4:47 PM
End: Today at 5:00 PM
Elapsed: 13 minutes
Vulnerabilities
Critical
High
Medium
Low
Info
ENG IN
05:02 PM 20-01-2025

RESULTS AND FINDINGS

BASIC (NON-AUTHENTICATED) SCAN –

192.168.31.2



Severity	CVSS v3.0	VPR Score	EPSS Score	Plugin	Name
HIGH	7.5	5.1	0.0398	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
MEDIUM	6.5	4.0	0.0035	50686	IP Forwarding Enabled
MEDIUM	6.5	-	-	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.5	-	-	57582	SSL Self-Signed Certificate
MEDIUM	6.5	-	-	104743	TLS Version 1.0 Protocol Detection
MEDIUM	6.5	-	-	157288	TLS Version 1.1 Deprecated Protocol
MEDIUM	5.3	-	-	57608	SMB Signing not required
LOW	2.1*	2.2	0.8939	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	-	46180	Additional DNS Hostnames
INFO	N/A	-	-	45590	Common Platform Enumeration (CPE)

Generally, this is simple scan that we performed on our virtual Windows 10 OS. To establish uninterrupted connectivity between local machine and virtual Windows 10 we disable the Microsoft Windows Defender Firewall. This allows ICMP requests. The results and findings of basic scan is as below-

VULNERABILITIES (WITH SEVERITY) –

High – 01

Medium – 06

Low – 01

DESCRIPTION –

SSL Medium Strength Cipher Suites Supported (SWEET32) :

This vulnerability has 7.5 CVSS score. The vulnerability arises when a server supports SSL/TLS cipher suites that use medium-strength encryption. Medium-strength encryption refers to encryption algorithms with key lengths between 64 and 112 bits or the 3DES encryption suite. These ciphers are vulnerable to attacks such as SWEET32, which exploits the collision weaknesses of 64-bit block ciphers.

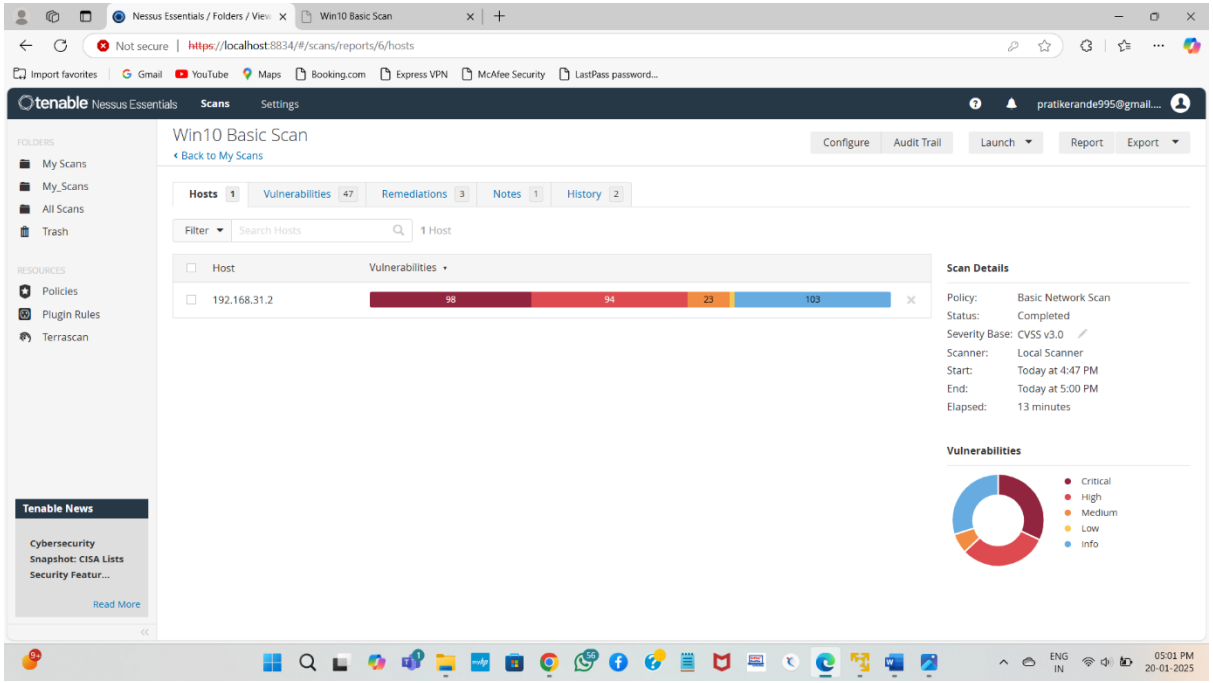
In SWEET32 attacks, an attacker may capture significant amounts of encrypted traffic to recover sensitive information. This risk increases if the attacker is on the same physical network as the victim, as they can more easily perform packet sniffing.

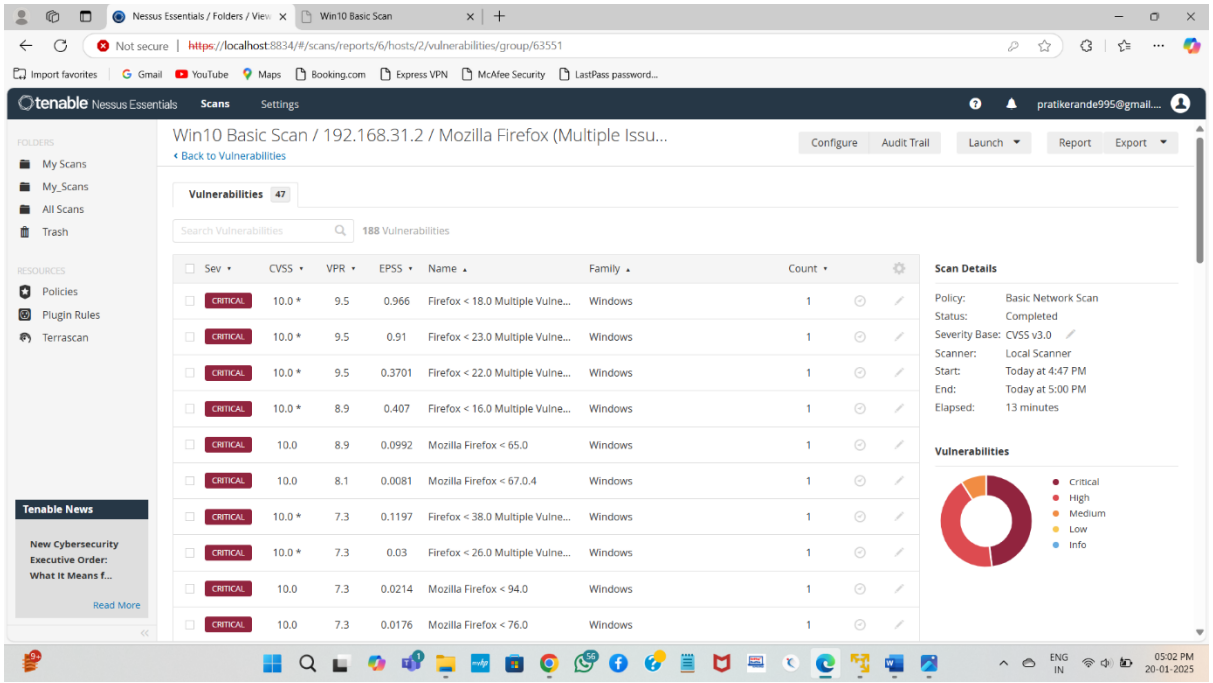
Solution

1. **Disable medium-strength cipher suites**, including:
    - Ciphers using key lengths below 112 bits.
    - Ciphers using 3DES (Triple DES) encryption.
2. **Enable strong ciphers**, such as:
    - AES with 128-bit or 256-bit key lengths (AES-GCM, AES-CCM, etc.).
    - Modern ciphers supporting forward secrecy (e.g., ECDHE-based ciphers).
3. **Update SSL/TLS settings**:
    - Use the latest version of TLS (e.g., TLS 1.2 or TLS 1.3).
    - Avoid deprecated protocols such as SSL 2.0, SSL 3.0, and TLS 1.0/1.1.

AUTHENTICATED SCAN

Generally, in this authenticated scan we provide the login credentials of our targeted system to Nessus. By providing these logging credentials Nessus will able to scan the system in-depth. With authenticated scan Nessus scans the registry, file system as well as all ports. This type of scan provides deeper insights into the system's security posture compared to unauthenticated scans because it can access and analyse system configurations, installed software, and other details that are not visible externally. We have installed outdated version of Mozilla Firefox during an authenticated scan. This will make the targeted system more vulnerable.





As compared to basic scan, in authenticated scan we have a lot of vulnerabilities. Vulnerabilities with critical severity are about outdated version of Mozilla Firefox that we've installed on our Windows 10 virtual machine.

VULNERABILITIES (WITH SEVERITY) –

CRITICAL – 98

HIGH – 94

MEDIUM – 23

LOW – 1

DESCRIPTION (CRITICAL VULNERABILITIES) –

Firefox < 44 Multiple Vulnerabilities :

The installed version of Firefox on the remote Windows host is outdated, with multiple vulnerabilities that pose security risks. These vulnerabilities can be exploited by remote attackers to compromise system integrity, execute arbitrary code, inject cookies, or trick users into visiting malicious websites. Below is a summary of the key issues:

Key Vulnerabilities

1. **Cookie Injection**:
    - Malicious control characters stored in cookies enable attackers to inject cookies.
      (CVE-2015-7208, CVE-2016-1939)
2. **Memory Corruption**:
    - Unspecified memory corruption issues can lead to remote code execution.
      (CVE-2016-1930, CVE-2016-1931, CVE-2016-1944)
3. **GIF Parsing Integer Overflow**:
    - Malformed GIF images cause denial of service or remote code execution.
      (CVE-2016-1933)
4. **WebGL Buffer Overflow**:
    - Cache out-of-memory errors allow attackers to execute arbitrary code. *(CVE-2016-1935)*
5. **Content Spoofing**:
    - Protocol handler flaws allow attackers to spoof content and trick users. *(CVE-2016-1937)*
6. **Cryptographic Weakness**:

    - Vulnerabilities in Network Security Services (NSS) weaken cryptographic operations. *(CVE-2016-1938, CVE-2016-1978)*

7. **URL Spoofing**:
    - Flaws in URL handling allow attackers to deceive users into visiting malicious sites. *(CVE-2016-1942)*
8. **Wild Pointer in ZIP Handling**:
    - Malformed ZIP files cause unspecified impacts due to memory issues. *(CVE-2016-1945)*
9. **MP4 Metadata Integer Overflow**:
    - Libstagefright library flaws allow arbitrary code execution via MP4 files. *(CVE-2016-1946)*
10. **Safe Browsing Flaw**:
    - Failure in the Application Reputation service lets attackers deliver malicious downloads unnoticed. *(CVE-2016-1947)*

IMPACT

Exploitation of these vulnerabilities can lead to:

- **System Compromise**: Remote code execution, memory corruption, or arbitrary file execution.
- **Data Exposure**: Injection of malicious cookies or interception of sensitive user data.
- **User Deception**: Content and URL spoofing to trick users into malicious actions.
- **Cryptographic Failures**: Weaknesses in encryption algorithms reducing security.

SOLUTION

Upgrade Firefox and install its latest version.

VULNERABILITIES REMIDIATION

Remediation Phase: Addressing Critical Vulnerabilities

The remediation phase involves systematically addressing identified critical vulnerabilities to reduce risk and strengthen the security posture. Key activities in this phase include:

1. **Prioritization**:
    - In this phase we will prioritize high severity vulnerabilities and which has potential impact. Focus on vulnerabilities classified as critical due to their potential impact and likelihood of exploitation.
2. **Patch Management**:
    - In this phase we will apply patches and updates to fix security flaws in software and systems.
3. **Configuration Adjustments**:
    - In this phase we'll reconfigure systems to remove insecure settings, such as disabling weak ciphers or enforcing secure protocols.
4. **Mitigation Measures**:
    - Deploy temporary safeguards, such as network segmentation or access restrictions, if immediate patching is not possible.
5. **Removal of Unsupported Systems**:
    - Remove outdated hardware or software that cannot be secured.

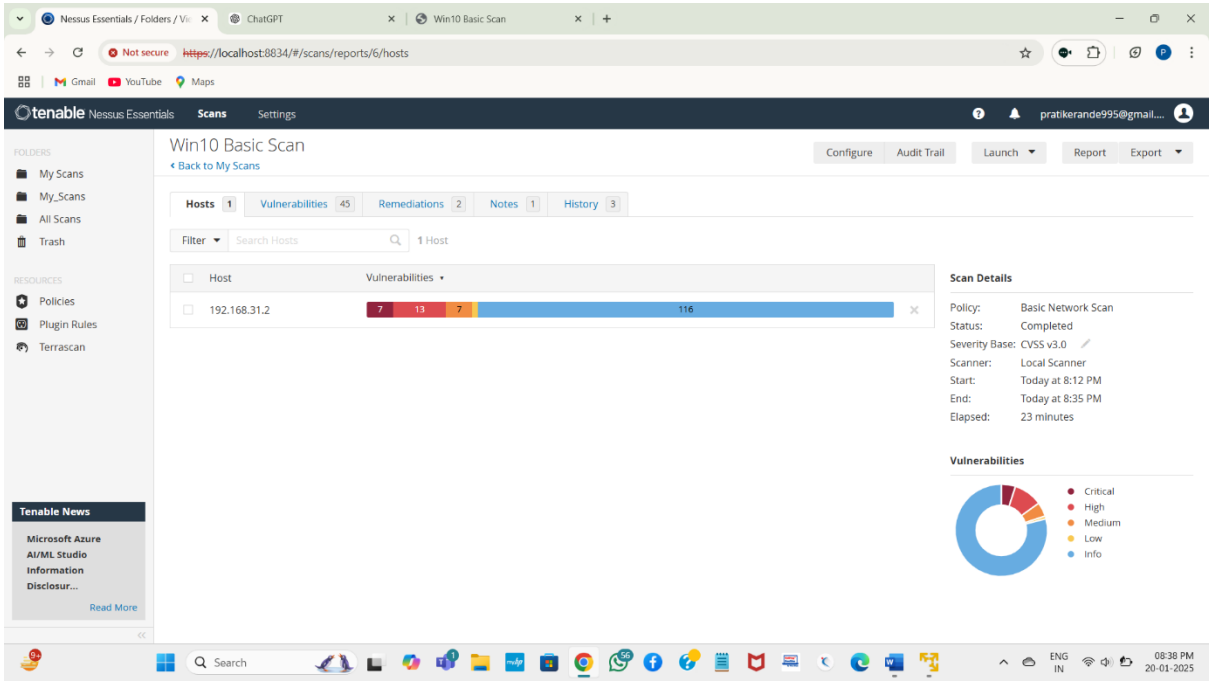
This phase ensures that critical vulnerabilities are effectively mitigated, reducing the attack surface and preparing the system for subsequent verification and continuous monitoring.

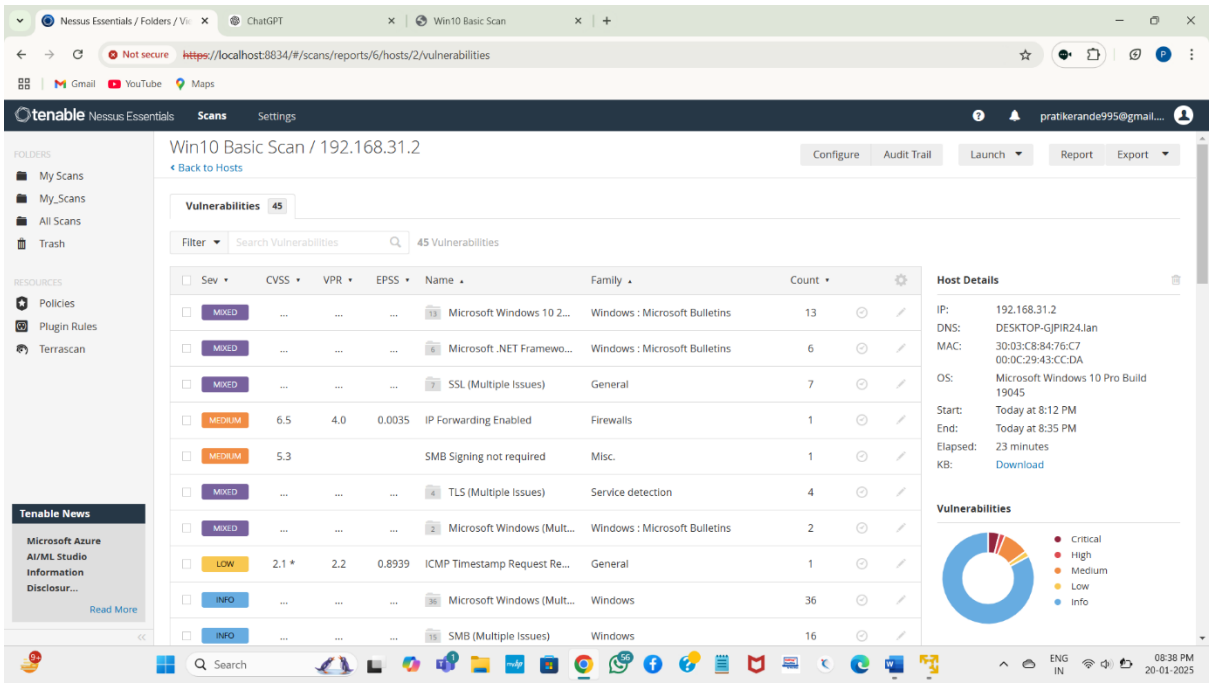
In this remediation phase we uninstalled our older and outdated Mozilla Firefox that was installed on Windows 10 virtual machine.

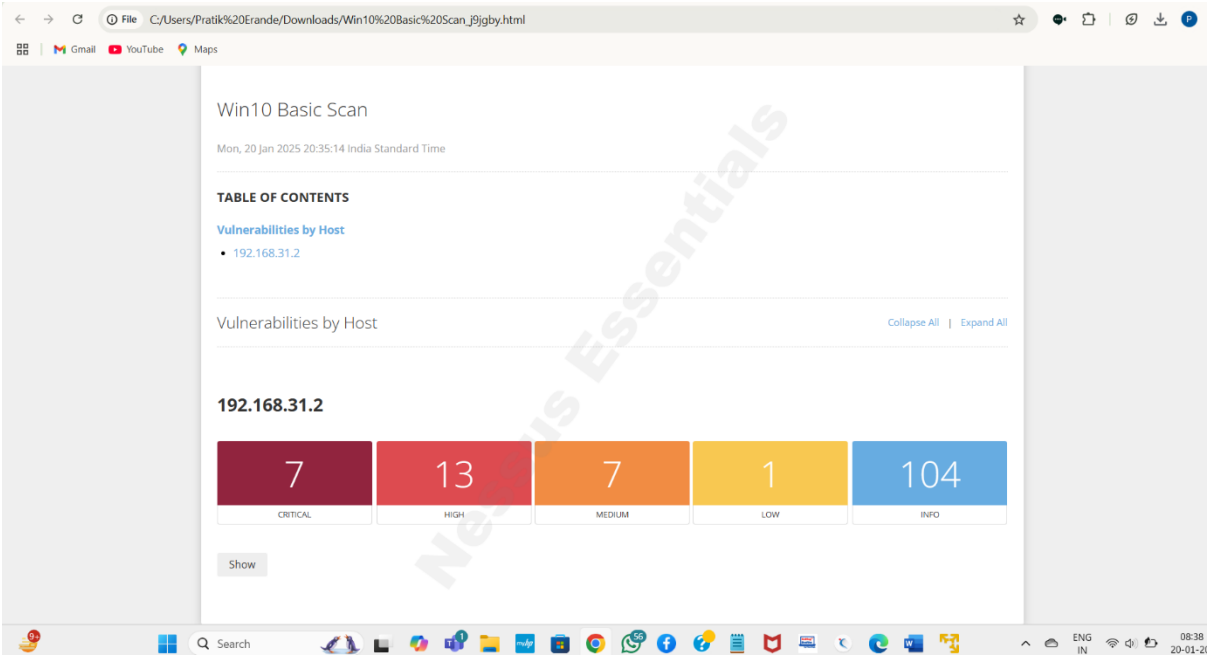
VERIFICATION & CONTINUOUS MONITORING

In verification and continuous monitoring phase, we will perform scans on Windows 10 that we have patched recently. We will rescan it to ensure that all the critical vulnerabilities was patched.

Here is the result of scanning







Post-Remediation Scan Results

The post-remediation scan was conducted to verify the status of previously identified vulnerabilities. The results are as follows:

1. **Patched Vulnerabilities**:
    - All critical vulnerabilities identified in the initial scan have been successfully patched or mitigated.
2. **Remaining Issues**:
    - [Windows 10 Security Update.]
3. **System Integrity**:
    - Validation confirms that the implemented patches and configuration changes have not introduced new vulnerabilities or impacted system performance.

Verification and **continuous monitoring** are essential components of a robust vulnerability management process. Their importance lies in:

1. **Verification**:
    - Ensures that remediation efforts were effective by confirming that vulnerabilities have been patched or mitigated.
    - Identifies any issues caused by remediation, such as misconfigurations or new vulnerabilities.
2. **Continuous Monitoring**:

- Detects emerging vulnerabilities, misconfigurations, or threats in real-time to maintain a secure environment.
- Ensures compliance with security policies and industry standards.
- Provides actionable insights for proactive risk management.

By combining verification and continuous monitoring, organizations maintain an adaptive security posture, ensuring ongoing protection against evolving threats.

CONCLUSION

In this **Vulnerability Assessment and Management Project using Nessus**, I successfully identified, prioritized, and remediated security vulnerabilities within the target environment. By using Nessus's powerful scanning capabilities, I was able to:

1. Detect critical vulnerabilities and evaluate their potential impact on the system.
2. Implement effective remediation measures through patching, configuration adjustments, and system updates.
3. Validate the effectiveness of the remediation by conducting post-remediation scans.

This project highlights the importance of a structured vulnerability management lifecycle, including assessment, remediation, verification, and monitoring. The implementation of these processes not only strengthens the organization's security posture but also ensures compliance with industry standards and best practices. Moving forward, the continuous monitoring phase will ensure that the environment remains resilient against evolving threats.