CEH PENETRATION TEST REPORT

Prepared by MD Muhtashim Jahin

Prepared For: BCIT

Location: Burnaby, British Columbia, Canada

Md Muhtashim Jahin

Date: 1 Nov, 2024

Penetration Testing Report

Project Title: CEH Challenge Lab Penetration Test

Date: 1 November 2024

Location: Burnaby, British Columbia, Canada

Prepared For: BCIT (Hamidreza Talebi)

Prepared By: Md Muhtashim Jahin

1. Executive Summary

This report details the findings of a penetration testing engagement conducted on specific network assets within the campus environment. The primary objectives of this assessment were to Identify Vulnerabilities, Enumerate Open Ports, perform Operating System (OS) detection, and, critically, to **Attempt to Gain Unauthorized Access to the designated target systems through the exploitation of identified vulnerabilities**.

The engagement was conducted in a controlled manner, adhering strictly to the defined scope and rules of engagement, including prohibitions against Denial of Service (DoS) attacks and service disruptions. My findings highlight potential security weaknesses that could be leveraged by malicious actors, and recommendations are provided to mitigate these risks.

2. Introduction

This document presents the results of a penetration test performed on the network infrastructure associated with the CEH Challenge Lab. The test was executed on Nov 1, 2024, from a penetration tester machine located on campus in Burnaby, British Columbia, Canada. The purpose of this test is to simulate real-world attack scenarios to identify exploitable vulnerabilities and provide a clear understanding of the current security posture of the in-scope systems.

3. Scope of Work

The scope of this penetration testing engagement was strictly limited to the following IP addresses as provided and depicted in the network diagram:

- **142.232.197.39:** IDS Honeypot (T-Pot)
- **142.232.197.72:** Windows PC
- **142.232.197.73:** Mail Server
- **142.232.197.67:** Ubuntu Web Server

Scanning or attempting to access any other IP addresses not explicitly listed in this document was strictly prohibited. All scans were conducted from an on-campus machine, and the IP address of the penetration testing machine will be documented within the report.

4. Methodology

The penetration testing engagement followed a structured approach, divided into three distinct phases to ensure comprehensive coverage and clear reporting:

4.1. Phase 1: Enumeration

This initial phase focused on gathering as much information as possible about the target systems. The objectives included:

- Identifying open ports on each target IP address.
- Performing Operating System (OS) detection for each host.
- Determining software version(s) used by services.
- Identifying running services.
- Documenting the tools used for data collection.
- Developing a network diagram based on discovered information.

Various tools were utilized to gather this data, with findings presented in either picture or text format.

4.2. Phase 2: Vulnerability Analysis

Following the enumeration phase, identified services and systems were analyzed for known vulnerabilities. Key activities in this phase included:

- Identifying specific vulnerabilities associated with discovered services and configurations.
- For each identified vulnerability, locate and include its corresponding Common Vulnerabilities and Exposures (CVE) number.
- Specifying which identified vulnerabilities were deemed plausible and providing detailed reasoning for these conclusions.

Strict adherence to ethical guidelines was maintained; Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks were not performed, and no services were intentionally turned down.

4.3. Phase 3: Reaching the Victim

The final phase involved attempting to exploit the identified plausible vulnerabilities to gain unauthorized access to the target machines. The objectives were:

- To successfully demonstrate access to the victim machine by exploiting an identified vulnerability.
- Documenting the tool(s) used for the exploitation attempt.
- Providing a step-by-step demonstration of the successful access, supported by detailed screenshots or logs as proof.

- Describing how the machine can be exploited.

Proof of concept (screenshots, logs, detailed explanations) is essential for all findings and steps, as findings without proof will not be considered valid.

4.4. Attacker Machine Configuration and Tools Used

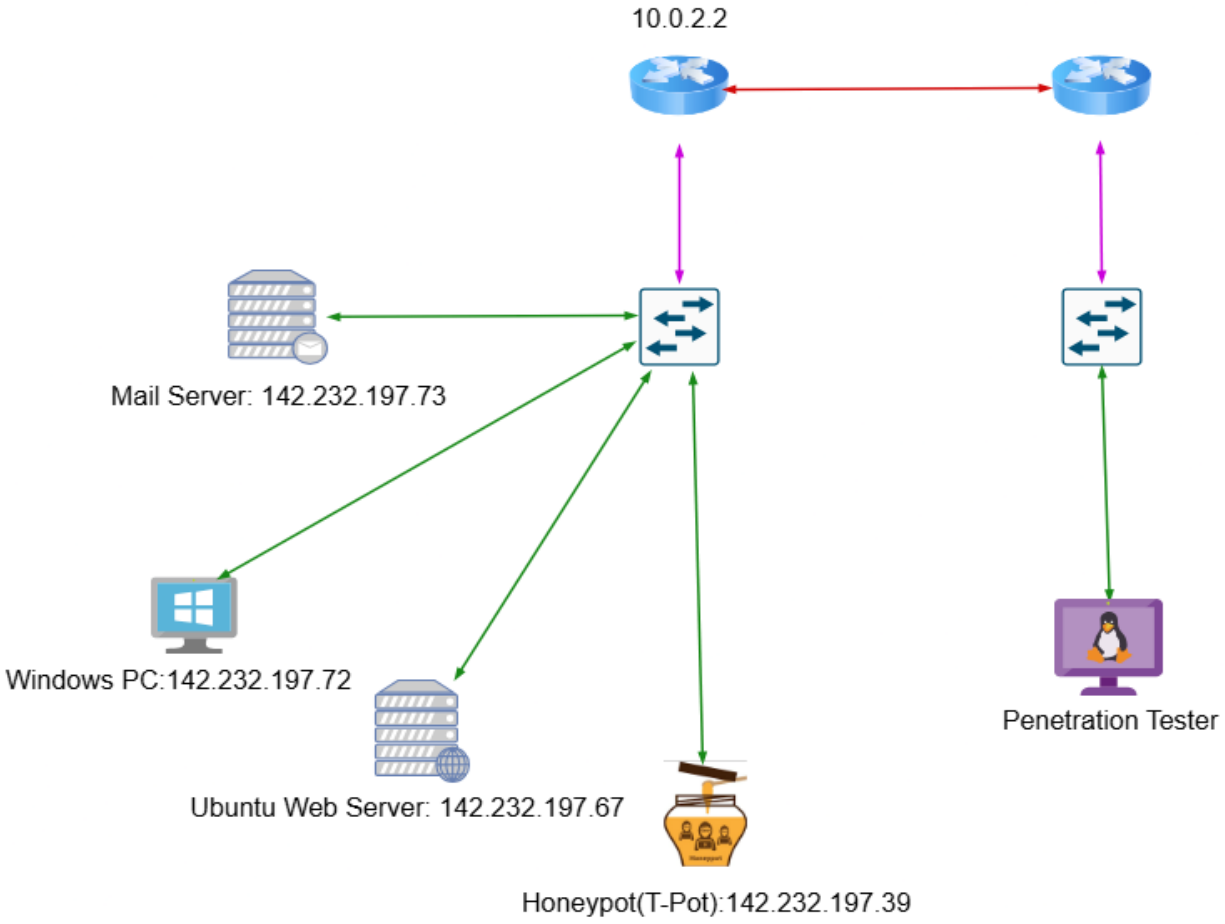
The penetration testing activities were conducted from a Kali Linux operating system, serving as the attacker machine. The IP address of the attacker machine within the lab environment was identified as **192.168.253.128**.

From the perspective of the attacker machine, all target IP addresses (142.232.197.39, 142.232.197.72, 142.232.197.73, 142.232.197.67) were **2 hops away**. This configuration defines the network path for the assessment, although direct connections within the local segment are also implied by the network diagram.

During the reconnaissance and vulnerability identification phases, the following primary tools were utilized:

- **Nmap (Network Mapper):** Employed for host discovery, port scanning, and OS detection.
- **Nessus:** Utilized for comprehensive vulnerability scanning to identify known security weaknesses and misconfigurations on the target systems.

5. Network Diagram



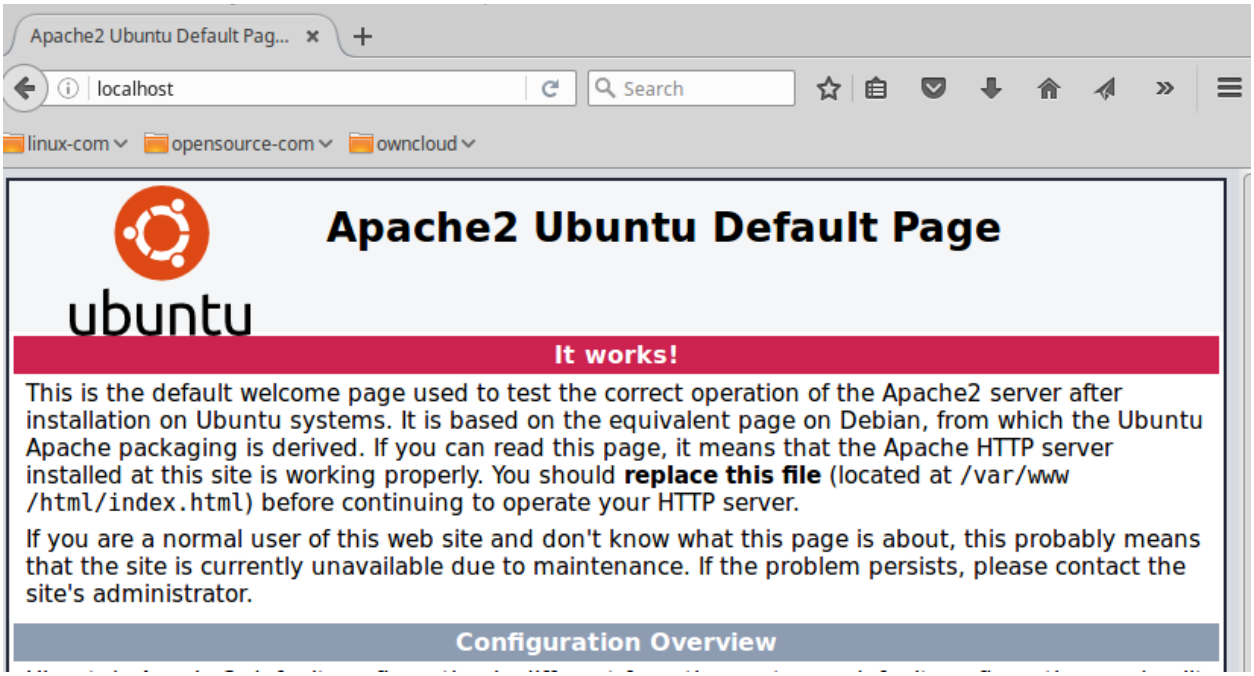
The diagram above illustrates the network topology and the in-scope target systems for this penetration test.

---

6. Findings

This section details the vulnerabilities and configurations identified on the in-scope systems during the enumeration and vulnerability analysis phases. Findings are categorized by target IP address.

6.1. Target IP: 142.232.197.67 (Ubuntu Web Server)



Operating System/Service Identification: The target system at 142.232.197.67 is identified as a **Linux Ubuntu Server** operating system, running an **Apache HTTP Server**.

```
PORT     STATE    SERVICE VERSION
22/tcp   open     ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 f1:f0:ac:73:b8:be:68:8e:67:7e:8e:8f:a8:35:75:4d (ECDSA)
|_  256 7d:20:9e:68:87:09:69:8d:75:eb:de:0e:ae:f7:81:ec (ED25519)
80/tcp   open     http    Apache httpd 2.4.58 ((Ubuntu))
|_http-server-header: Apache/2.4.58 (Ubuntu)
| http-methods:
|_  Supported Methods: POST OPTIONS HEAD GET
|_http-title: Apache2 Ubuntu Default Page: It works
1723/tcp filtered pptp
Aggressive OS guesses: Oracle Virtualbox (94%), QEMU user mode network gateway (94%), Bay Networks BayStack 450 switch (software version 3.1.0.22) (87%), Allied
5%), D-Link DP-300U, DP-G310, or Hamlet HPS01UU print server (85%), Linux 2.6.18 (CentOS 5, x86_64, SMP) (85%), Bay Networks BayStack 450 switch (software versio
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=17 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 8080/tcp)
HOP RTT      ADDRESS
1   0.20 ms  10.0.2.2
2   10.54 ms 142.232.197.67
```

Open Ports & Running Services: The following ports were identified as open, with their corresponding services:

- **Port 22/tcp:** Running **SSH** (Secure Shell)
- **Port 80/tcp:** Running **HTTP** (Hypertext Transfer Protocol)
- Further investigation revealed that a web application, **Damn Vulnerable Web Application (DVWA)**, is accessible via the HTTP service. The default Apache web page was initially observed, but accessing https://localhost/dvwa (or http://142.232.197.67/dvwa in the context of the target) revealed the DVWA interface.

Software Version Used: The following software versions were identified:

- **OpenSSH:** 9.6p1 Ubuntu 3ubuntu13.5
- **Apache HTTPD:** 2.4.58 ((Ubuntu))

Identified Vulnerabilities: Based on the scans, the following vulnerabilities related to the Apache HTTP Server were identified. These generally indicate that the installed Apache version is significantly outdated and lacks recent security patches.

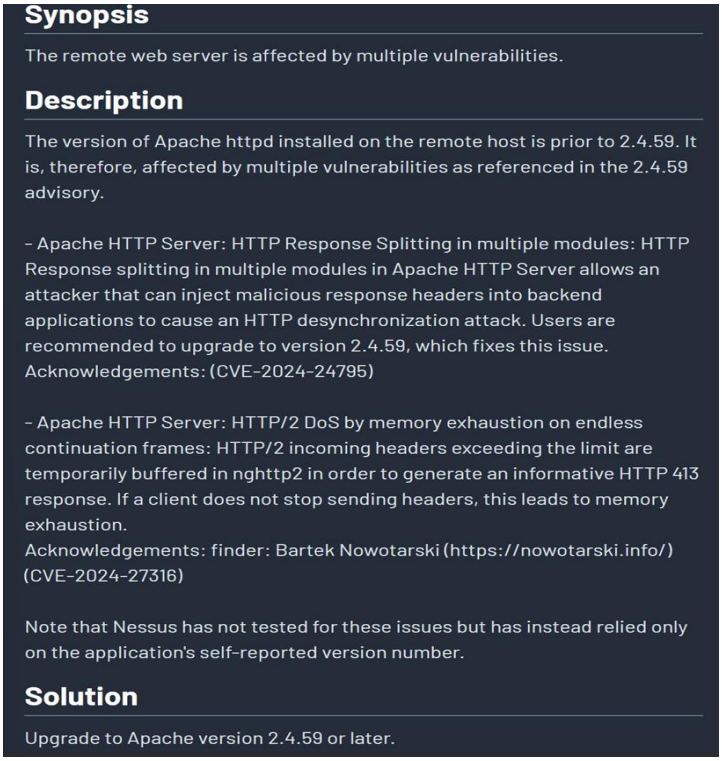
1. **Apache 2.4.x < 2.4.60 Multiple Vulnerabilities**
    - **Relevant CVE IDs:**
        - **CVE-2024-36387:** DoS by Null Pointer in WebSocket over HTTP/2.
        - **CVE-2024-38472:** SSRF in Apache HTTP Server on Windows (can leak NTLM hashes).
        - **CVE-2024-38473:** Encoding problem in mod_proxy (bypassing authentication).

- **CVE-2024-38474:** Substitution encoding issue in mod_rewrite (script execution/source disclosure).
- **CVE-2024-38475:** Improper escaping in mod_rewrite (code execution/source disclosure).
- **CVE-2024-38476:** Information disclosure, SSRF, or local script execution via backend.
- **CVE-2024-38477:** Null pointer dereference in mod_proxy (server crash/DoS).
- **CVE-2024-39573:** Potential SSRF in mod_rewrite (unexpected mod_proxy handling).

- **Plausibility:** [*Provide reasoning for its plausibility based on your analysis. Remember that CVE-2024-38472 and CVE-2024-40898 specifically mention Windows, so you should note if they are not applicable to this Ubuntu server.*]

2. **Apache 2.4.x < 2.4.59 Multiple Vulnerabilities**

- **Relevant CVE IDs:** (These vulnerabilities are also addressed in versions prior to 2.4.59, and typically overlap with the fixes in 2.4.60)
  - **CVE-2024-24795:** HTTP Response splitting in multiple modules.
  - **CVE-2024-27316:** HTTP/2 DoS by memory exhaustion on endless continuation frames.
  - **CVE-2023-38709:** HTTP response splitting (faulty input validation).



Synopsis

The remote web server is affected by multiple vulnerabilities.

Description

The version of Apache httpd installed on the remote host is prior to 2.4.59. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.59 advisory.

- Apache HTTP Server: HTTP Response Splitting in multiple modules: HTTP Response splitting in multiple modules in Apache HTTP Server allows an attacker that can inject malicious response headers into backend applications to cause an HTTP desynchronization attack. Users are recommended to upgrade to version 2.4.59, which fixes this issue. Acknowledgements: (CVE-2024-24795)

- Apache HTTP Server: HTTP/2 DoS by memory exhaustion on endless continuation frames: HTTP/2 incoming headers exceeding the limit are temporarily buffered in nghttp2 in order to generate an informative HTTP 413 response. If a client does not stop sending headers, this leads to memory exhaustion.
Acknowledgements: finder: Bartek Nowotarski (https://nowotarski.info/) (CVE-2024-27316)

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

Solution

Upgrade to Apache version 2.4.59 or later.

3. **Apache 2.4.x < 2.4.62 Multiple Vulnerabilities**
    - **Relevant CVE IDs:**
        - **CVE-2024-40898:** SSRF in Apache HTTP Server on Windows with mod_rewrite in server/vhost context (can leak NTLM hashes).
        - **CVE-2024-40725:** Source code disclosure with handlers configured via AddType (regression fix for CVE-2024-39884).

7. Phase 3: Reaching the Victim - Exploitation

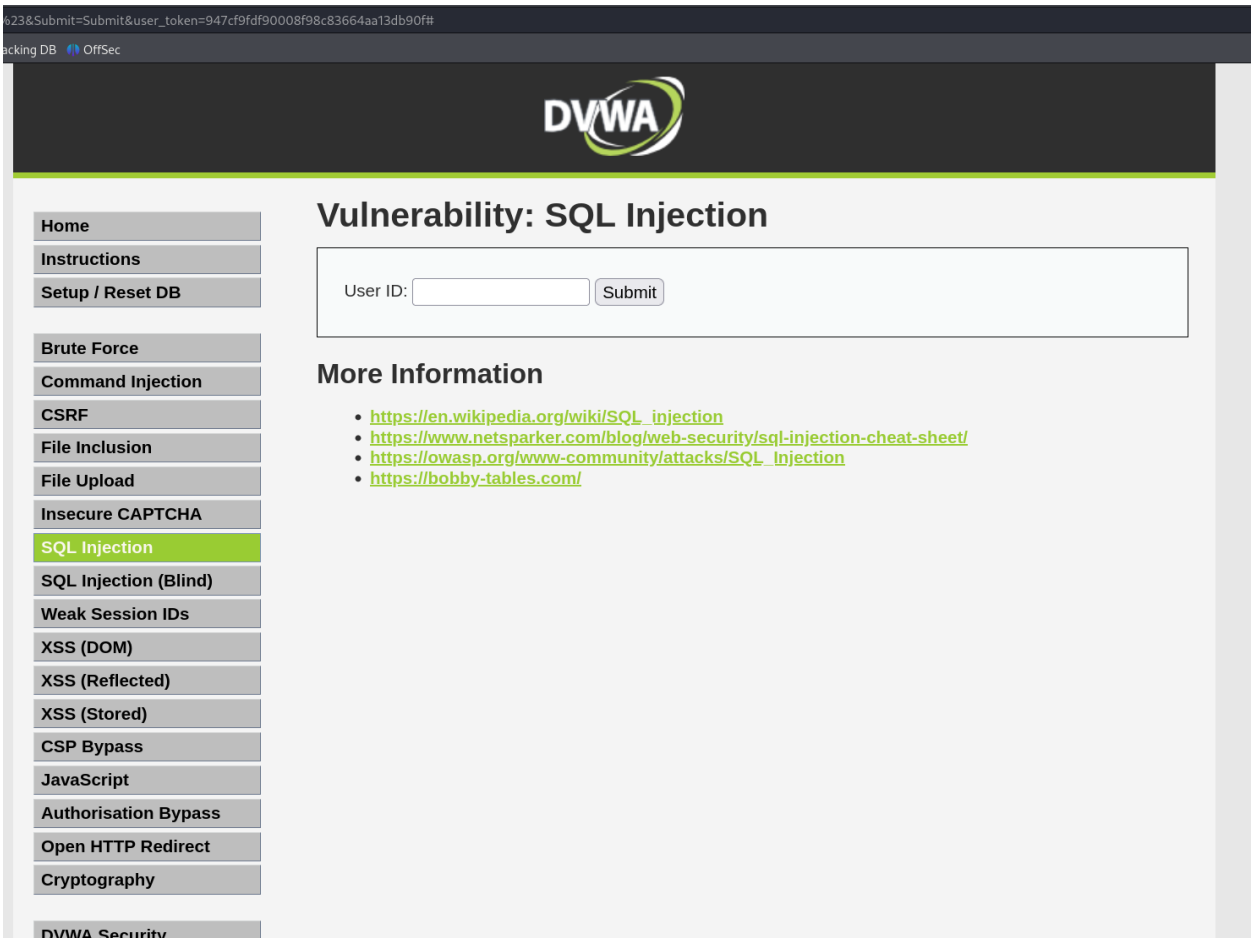
This phase details the attempts made to gain unauthorized access to the target systems by exploiting identified vulnerabilities.

7.1. Exploitation of 142.232.197.67 (Ubuntu Web Server) via DVWA

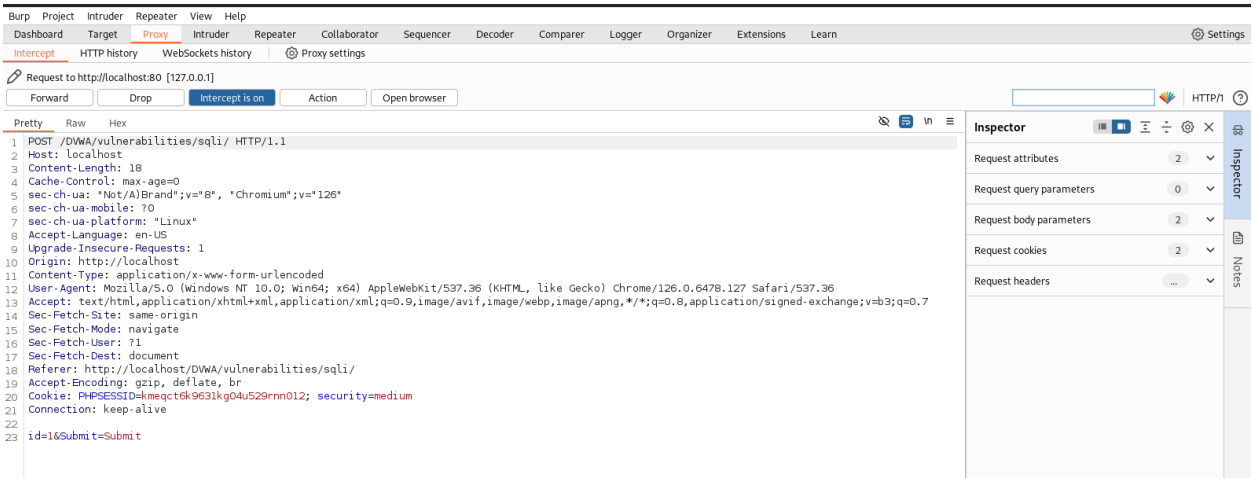
Upon discovering the presence of the Damn Vulnerable Web Application (DVWA) on the Ubuntu Web Server, the exploitation efforts were focused on leveraging the intentional vulnerabilities within DVWA. The SSH service was found to have a strong password, precluding a direct brute-force approach.

Exploitation Steps: SQL Injection

Having identified DVWA as a highly vulnerable web application, a SQL Injection vulnerability was targeted to extract sensitive information, specifically usernames and passwords.



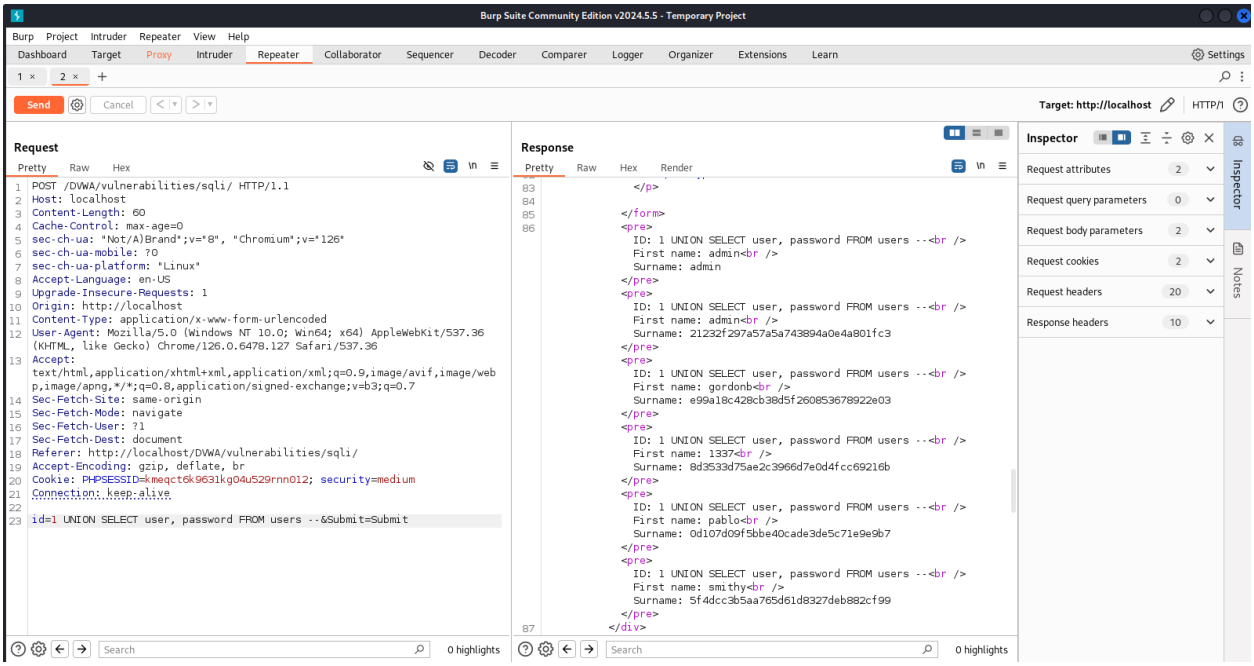
1. Intercepting the Request: Burp Suite was launched and configured to intercept web traffic. A request to the DVWA web application (specifically the SQL Injection challenge page, where an input field for an ID was present) was captured.



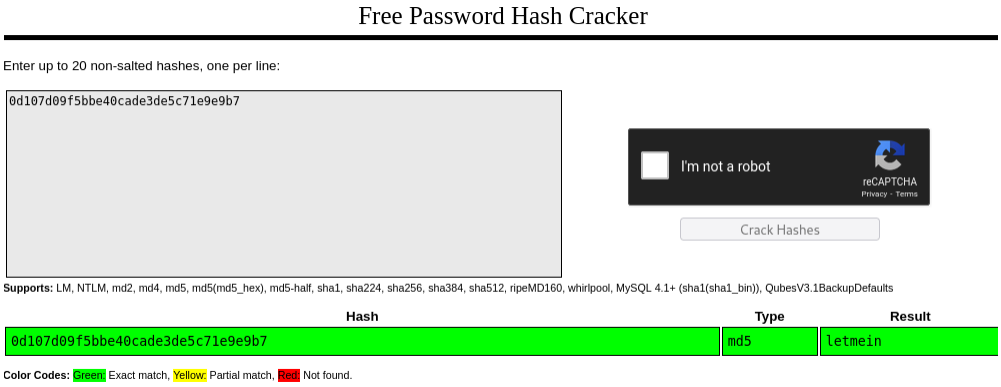
2. **Sending to Repeater:** The intercepted request was sent to Burp Suite's Repeater tool (Ctrl + R) for manipulation.

3. **Injecting the Payload:** In the Repeater, the following SQL Injection payload was injected into the ID input parameter: 1 UNION SELECT user, password FROM users-- This payload is designed to bypass the intended query logic and extract the user and password columns from the users table. The -- at the end comments out the remainder of the original SQL query.

4. **Analyzing the Response:** The modified request was sent. The response page was scrolled to the end, where the extracted usernames and hashed passwords were found, typically presented in the "first name" and "surname" display areas of the DVWA page.



5. Password Cracking: One of the extracted hashed passwords, specifically for the user "Pablo", was selected. This hash was then copied and pasted into a free online hash cracking tool.



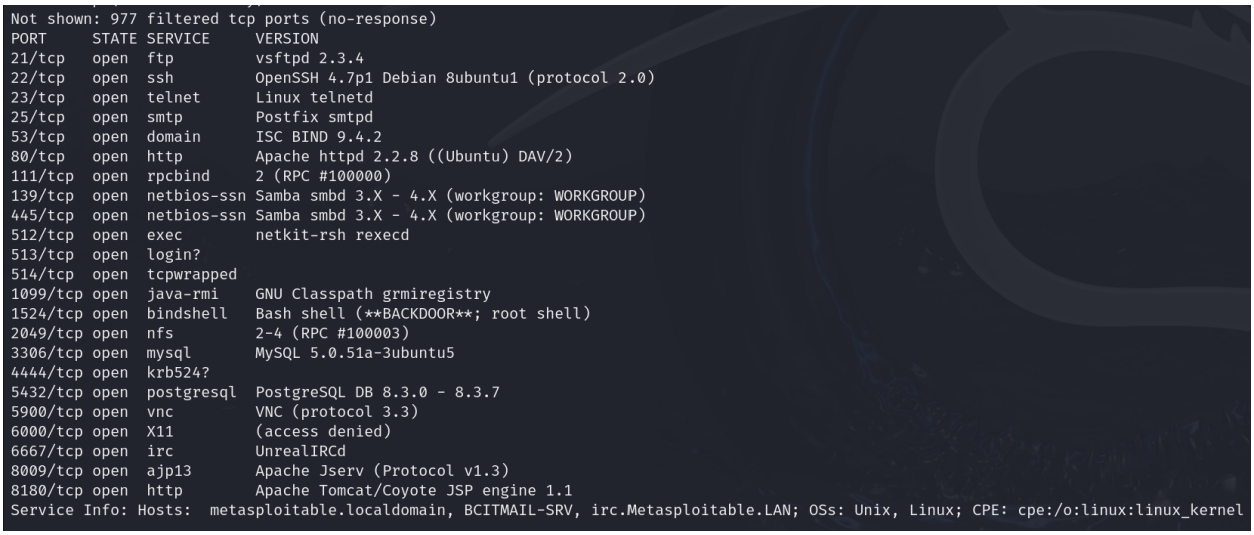
6. **Discovered Credential:** The online tool successfully cracked the hash, revealing the password to be **"letmein"**. The hash algorithm was identified as MD5.

This successful SQL Injection allowed the retrieval of sensitive user credentials from the DVWA database.

6.2. Target IP: 142.232.197.73 (Mail Server)

Operating System/Service Identification: The operating system for this Mail Server is identified as **Debian Linux**.

Open Ports, Running Services & Software Versions: The following extensive list of open ports, corresponding services, and identified software versions were discovered on 142.232.197.73:



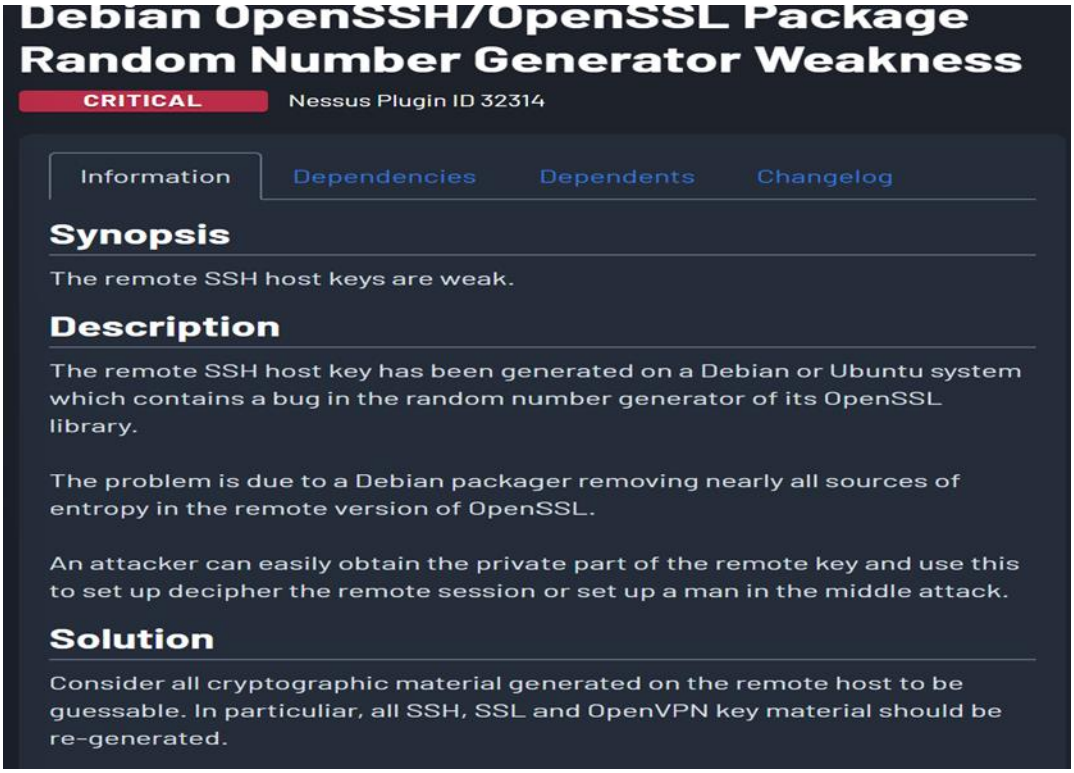
```
Not shown: 977 filtered tcp ports (no-response)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Bash shell (**BACKDOOR**; root shell)
2049/tcp open  nfs         2-4 (RPC #100003)
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
4444/tcp open  krb524?
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, BCITMAIL-SRV, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

- **Port 21/tcp:** FTP - vsftpd 2.3.4
- **Port 22/tcp:** SSH - OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
- **Port 23/tcp:** Telnet - Linux telnetd
- **Port 25/tcp:** SMTP - Postfix smtp
- **Port 53/tcp:** Domain - ISC BIND 9.4.2
- **Port 80/tcp:** HTTP - Apache httpd 2.2.8
- **Port 111/tcp:** rpcbind - 2 (RPC #100000)
- **Port 139/tcp:** NetBIOS Session Service - Samba smbd 3.X - 4.X
- **Port 445/tcp:** Microsoft-DS (SMB) - Samba smbd 3.X - 4.X
- **Port 512/tcp:** Exec - netkit-rsh rexecd
- **Port 1099/tcp:** Java RMI - GNU Classpath grmiregistry
- **Port 1524/tcp:** Bindshell - Bash shell (**BACKDOOR**; root shell)
- **Port 2049/tcp:** NFS - 2-4 (RPC #100003)
- **Port 3306/tcp:** MySQL - MySQL 5.0.51a-3ubuntu5
- **Port 4444/tcp:** Kerberos - krb524
- **Port 5432/tcp:** PostgreSQL DB - PostgreSQL DB 8.3.0 - 8.3.7
- **Port 5900/tcp:** VNC - VNC (Protocol 3.3)
- **Port 6667/tcp:** IRC - UnrealIRCd
- **Port 8009/tcp:** AJP13 - Apache Jserv (Protocol v1.3)

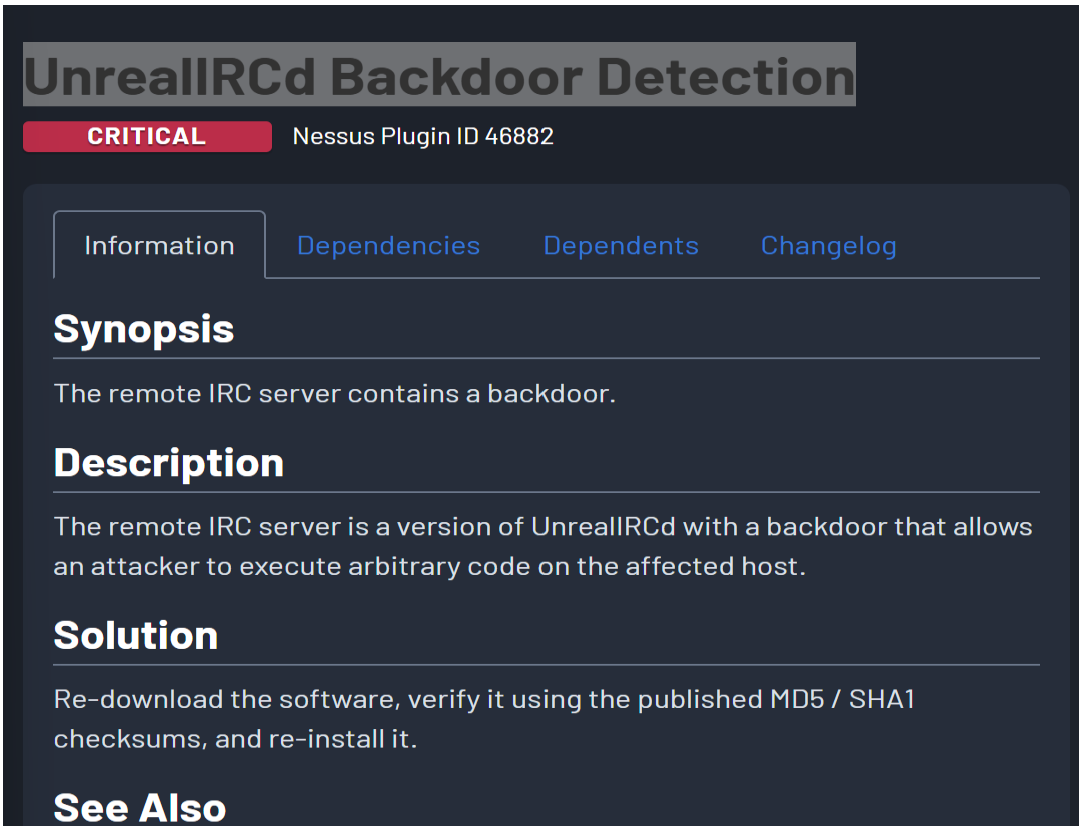
- **Port 8180/tcp:** HTTP - Apache Tomcat/Coyote JSP engine 1.1

Identified Vulnerabilities: Based on the extensive services and their identified versions, numerous critical vulnerabilities are present on this system:

1. **Debian OpenSSH/OpenSSL Package Random Number Generator Weakness**
    - **CVE:** CVE-2008-0166

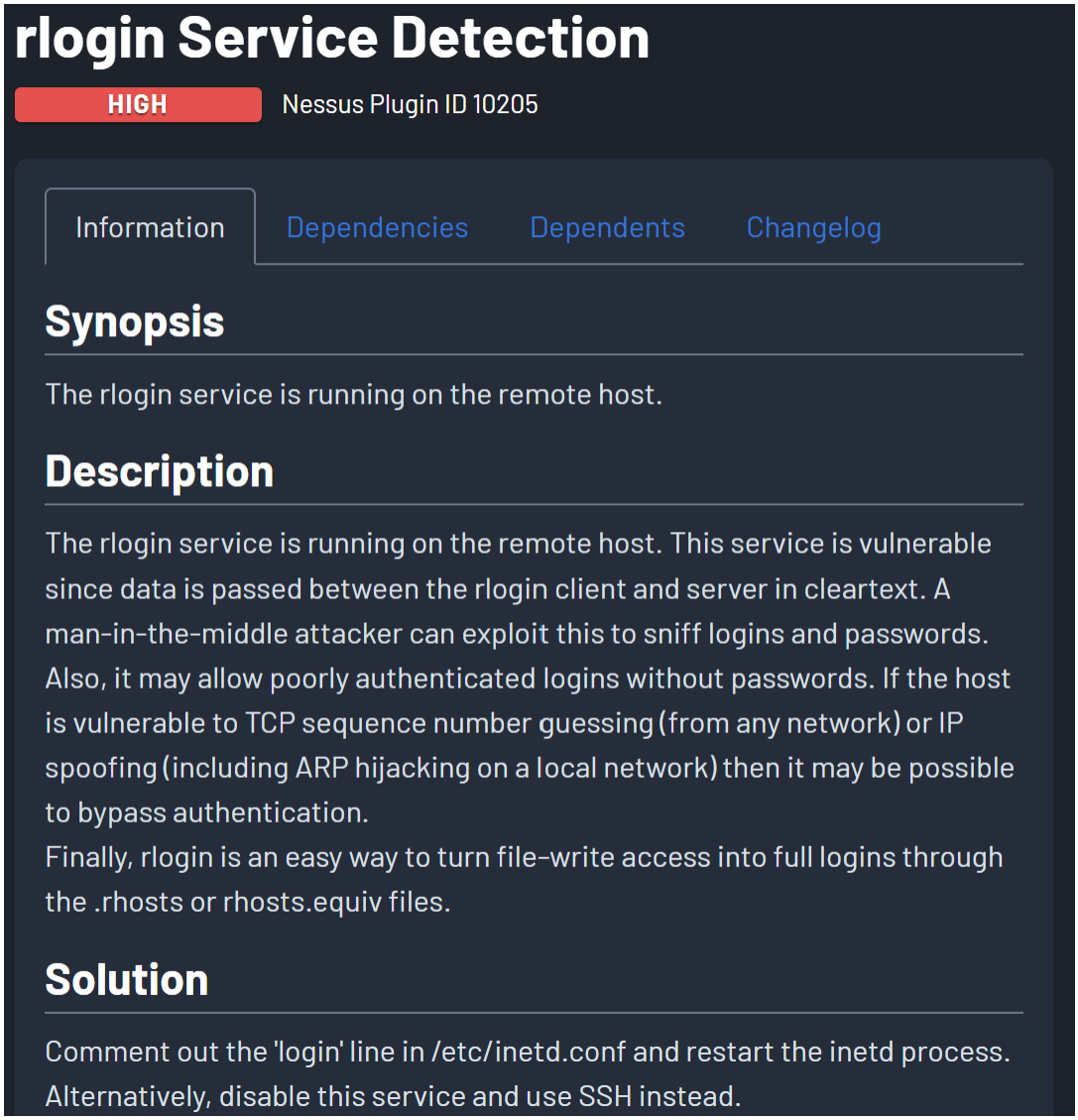


2. **UnrealIRCd Backdoor Detection**
    - **CVE:** CVE-2010-2075

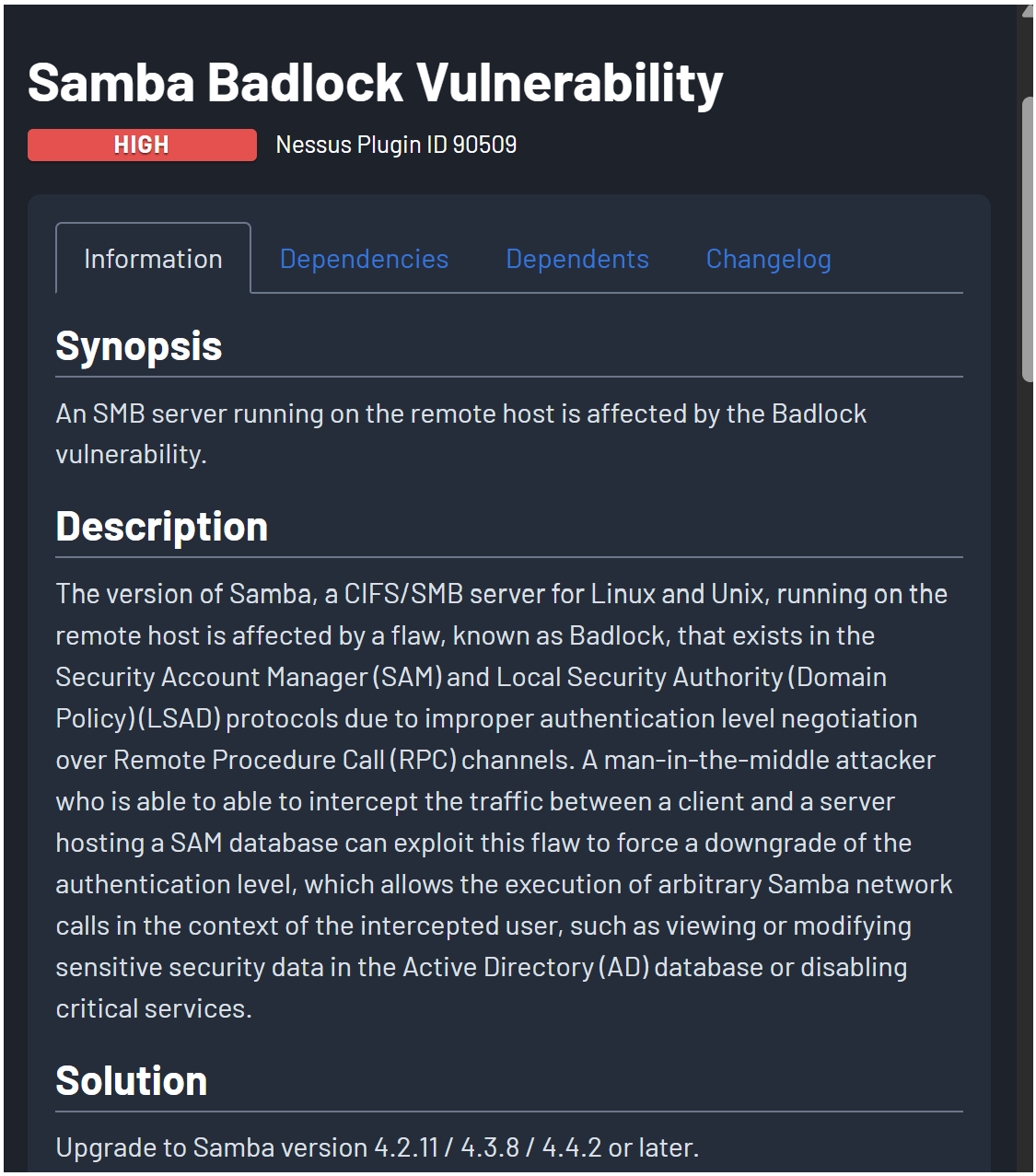


3. **VNC Server 'password' Password**
    - **CVE:** CVE-2010-2075 (Note: CVE-2010-2075 is also for UnrealIRCd. If VNC has a different specific CVE for a weak password issue, please clarify. If not, state that it's a configuration issue or a general weak password finding if no specific CVE applies).

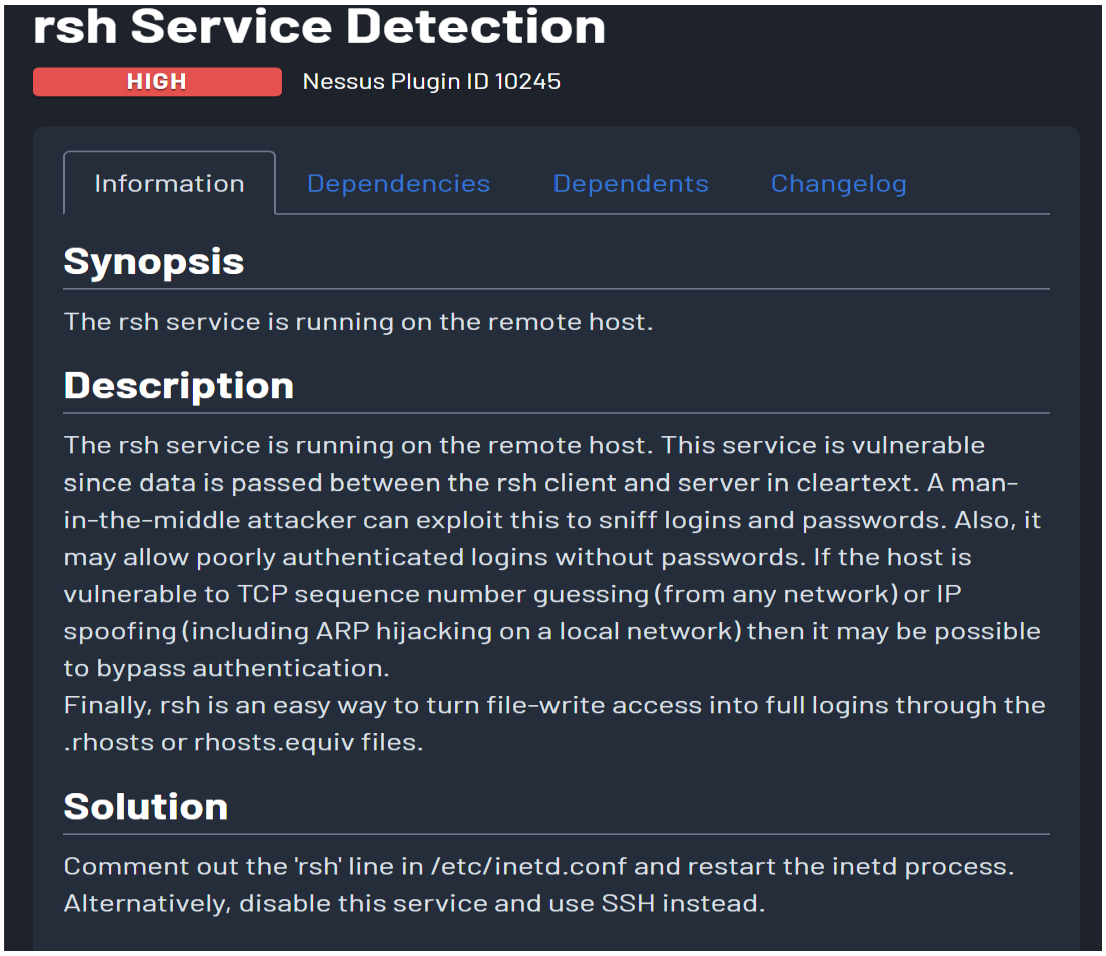
4. **rlogin Service Detection**
    - **CVE:** CVE-1999-0651



5. **Samba Badlock Vulnerability**
    - **CVE:** CVE-2016-2118



6. **rsh Service Detection**
    - **CVE:** CVE-1999-0651



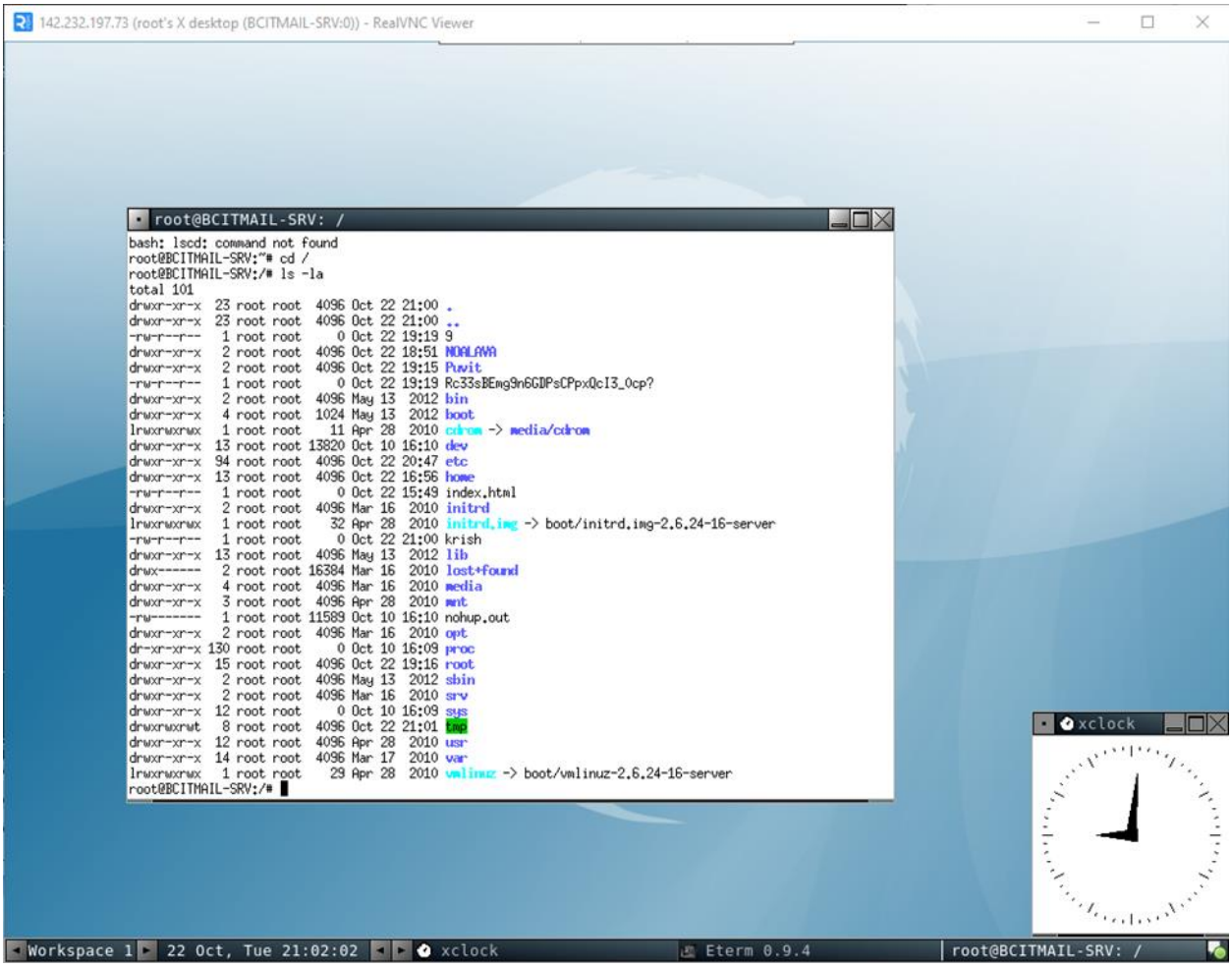
7. **Unencrypted Telnet Server**
    - **CVE:** CVE-2018-10698
8. **SMB Signing not required**
    - **CVE:** CVE-2016-2115
9. **HTTP TRACE / TRACK Methods Allowed**
    - **CVE:** CVE-2010-0386
10. **SSL Certificate Expiry**
    - **CVE:** Not Available
11. **SSH Weak Key Exchange Algorithms Enabled**
    - **CVE:** CVE-2022-29245
12. **SMTP Service STARTTLS Plaintext Command Injection**
    - **CVE:** CVE-2011-2165

7.2. Exploitation of 142.232.197.73 (Mail Server)

Two direct methods of remote access were identified and successfully utilized on the Mail Server:

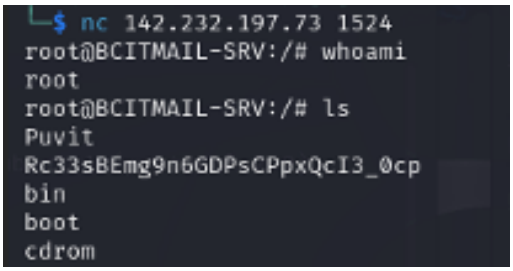
7.2.1. VNC Remote Access

1. **Vulnerability:** The VNC server running on Port 5900/tcp was secured with a weak, easily guessable password.
2. **Exploitation:** Access to the VNC session was gained using the default password **"password"**.
3. **Proof of Concept:**



7.2.2. Backdoor Shell Access (Port 1524)

1. **Vulnerability:** A highly critical finding was the presence of a bindshell (backdoor) listening on Port 1524/tcp, providing a direct **root shell** upon connection.
2. **Exploitation:** A direct connection was established to this port using the netcat command from the attacker machine: nc 142.232.197.73 1524
3. **Proof of Concept:**



7.2.3. Post-Exploitation: Credential Discovery

After gaining root access to the Mail Server, further enumeration of the file system led to the discovery of sensitive credentials.

1. **Method of Discovery:** During internal file enumeration on the compromised Mail Server, a file named mywindows7.txt was located.
2. **Credential Extraction:** The content of mywindows7.txt was read using the cat command, revealing the following credentials:
    - **ID:** BCIT
    - **Password:** 112233

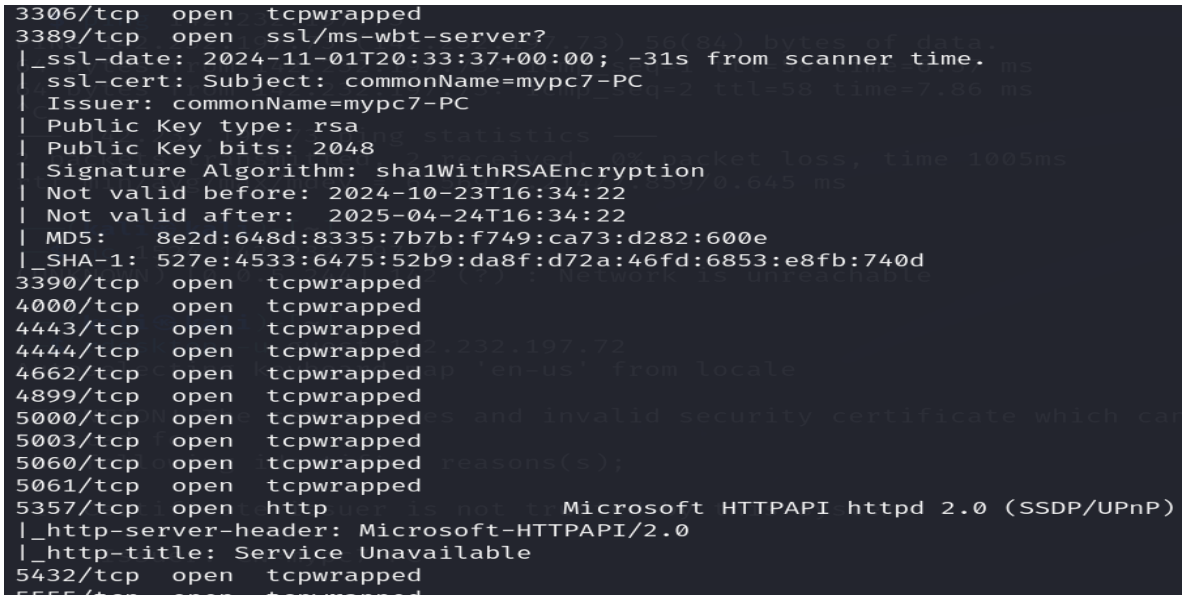
6.3. Target IP: 142.232.197.72 (Windows PC)

Operating System/Service Identification: The target system at 142.232.197.72 is identified as a **Microsoft Windows 7 Ultimate 6.1** PC.

```
| smb-os-discovery:
|   OS: Windows 7 Ultimate 7601 Service Pack 1 (Windows 7 Ultimate 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1
|   Computer name: mypc7-PC
|   NetBIOS computer name: MYPC7-PC\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2024-11-01T13:33:03-07:00
|_clock-skew: mean: 1h44m31s, deviation: 3h30m04s, median: -31s
```

Open Ports & Running Services: The following open ports and services were identified on 142.232.197.72:

```
PORT     STATE SERVICE       VERSION
1/tcp    open  tcpwrapped
7/tcp    open  tcpwrapped
9/tcp    open  tcpwrapped
13/tcp   open  tcpwrapped
17/tcp   open  tcpwrapped
19/tcp   open  tcpwrapped
21/tcp   open  tcpwrapped
22/tcp   open  tcpwrapped
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
23/tcp   open  tcpwrapped
25/tcp   open  tcpwrapped
|_smtp-commands: Couldn't establish connection on port 25
42/tcp   open  tcpwrapped
53/tcp   open  tcpwrapped
80/tcp   open  tcpwrapped
81/tcp   open  tcpwrapped
82/tcp   open  tcpwrapped
83/tcp   open  tcpwrapped
88/tcp   open  tcpwrapped
110/tcp  open  tcpwrapped
111/tcp  open  tcpwrapped
113/tcp  open  tcpwrapped
119/tcp  open  tcpwrapped
135/tcp  open  msrpc             Microsoft Windows RPC
139/tcp  open  netbios-ssn       Microsoft Windows netbios-ssn
143/tcp  open  tcpwrapped
389/tcp  open  tcpwrapped
443/tcp  open  tcpwrapped
445/tcp  open  microsoft-ds      Windows 7 Ultimate 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
464/tcp  open  tcpwrapped
465/tcp  open  tcpwrapped
|_smtp-commands: Couldn't establish connection on port 465
543/tcp  open  tcpwrapped
548/tcp  open  tcpwrapped
|_afp-serverinfo: ERROR: Script execution failed (use -d to debug)
563/tcp  open  tcpwrapped
587/tcp  open  tcpwrapped
|_smtp-commands: Couldn't establish connection on port 587
593/tcp  open  tcpwrapped
631/tcp  open  tcpwrapped
636/tcp  open  tcpwrapped
993/tcp  open  tcpwrapped
995/tcp  open  tcpwrapped
999/tcp  open  tcpwrapped
1024/tcp open  tcpwrapped
1028/tcp open  tcpwrapped
1080/tcp open  tcpwrapped
```



- **Port 1/tcp:** tcpwrapped
- **Port 7/tcp:** tcpwrapped
- **Port 13/tcp:** tcpwrapped
- **Port 17/tcp:** tcpwrapped
- **Port 19/tcp:** tcpwrapped
- **Port 21/tcp:** tcpwrapped
- **Port 22/tcp:** tcpwrapped
- **Port 23/tcp:** tcpwrapped
- **Port 25/tcp:** tcpwrapped
- **Port 42/tcp:** tcpwrapped
- **Port 53/tcp:** tcpwrapped
- **Port 79/tcp:** tcpwrapped
- **Port 80/tcp:** tcpwrapped
- **Port 81/tcp:** tcpwrapped
- **Port 82/tcp:** tcpwrapped
- **Port 83/tcp:** tcpwrapped
- **Port 110/tcp:** tcpwrapped
- **Port 111/tcp:** tcpwrapped

- **Port 113/tcp:** tcpwrapped
- **Port 135/tcp:** msrpc - Microsoft Windows RPC
- **Port 139/tcp:** netbios-ssn - Microsoft Windows netbios-ssn
- **Port 143/tcp:** tcpwrapped
- **Port 443/tcp:** tcpwrapped
- **Port 445/tcp:** microsoft-ds - Windows 7 Ultimate 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
- **Port 465/tcp:** tcpwrapped
- **Port 548/tcp:** tcpwrapped
- **Port 563/tcp:** tcpwrapped
- **Port 587/tcp:** tcpwrapped
- **Port 593/tcp:** tcpwrapped
- **Port 631/tcp:** tcpwrapped
- **Port 636/tcp:** tcpwrapped
- **Port 993/tcp:** tcpwrapped
- **Port 995/tcp:** tcpwrapped
- **Port 1024/tcp:** tcpwrapped
- **Port 1080/tcp:** tcpwrapped
- **Port 3306/tcp:** tcpwrapped
- **Port 3389/tcp:** ssl/ms-wbt-server
- **SSL Certificate Details:**
    - Subject: commonName=mypc7-PC
    - Issuer: commonName=mypc7-PC
    - Public Key type: rsa
    - Public Key bits: 2048
    - Signature Algorithm: sha1WithRSAEncryption
    - Not valid before: 2024-10-23T16:34:22
    - Not valid after: 2025-04-24T16:34:22
    - MD5: 8e2d:648d:8335:b7bb:f749:ca73:d282:600e

    - SHA-1: 572e:4533:6475:52b9:da8f:d72a:46fd:6853:e8fb:740d

- **Port 3390/tcp:** tcpwrapped
- **Port 4443/tcp:** tcpwrapped
- **Port 4444/tcp:** tcpwrapped
- **Port 4662/tcp:** tcpwrapped
- **Port 4899/tcp:** tcpwrapped
- **Port 5000/tcp:** tcpwrapped
- **Port 5003/tcp:** tcpwrapped
- **Port 5060/tcp:** tcpwrapped
- **Port 5357/tcp:** http - Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

  http-server-header: Microsoft-HTTPAPI/2.0

  http-title: Service Unavailable

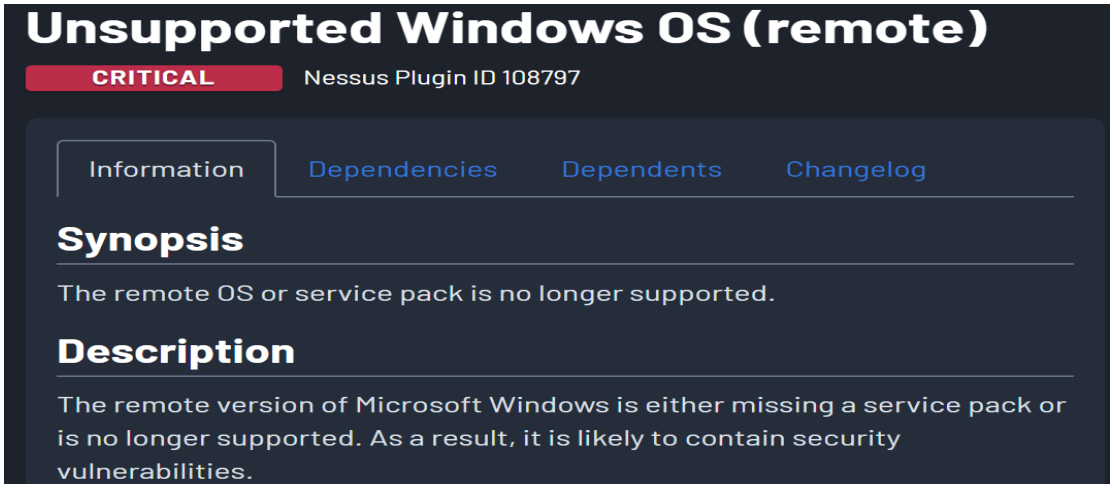
- **Port 5432/tcp:** tcpwrapped

Note on 'tcpwrapped' Services: The term tcpwrapped in Nmap scan results indicates that a TCP connection was successfully established with the port, but the service did not respond to Nmap's application-layer probes. This often means that a **TCP Wrapper** (like tcpd on Linux) or a **firewall** is configured to permit the connection but then immediately drop it or deny further interaction, preventing Nmap from identifying the specific service. It can also occur if the service crashed or is not handling the probe correctly. While the port is technically "open" for connection, the service behind it is either intentionally restricting access or not functioning as expected.

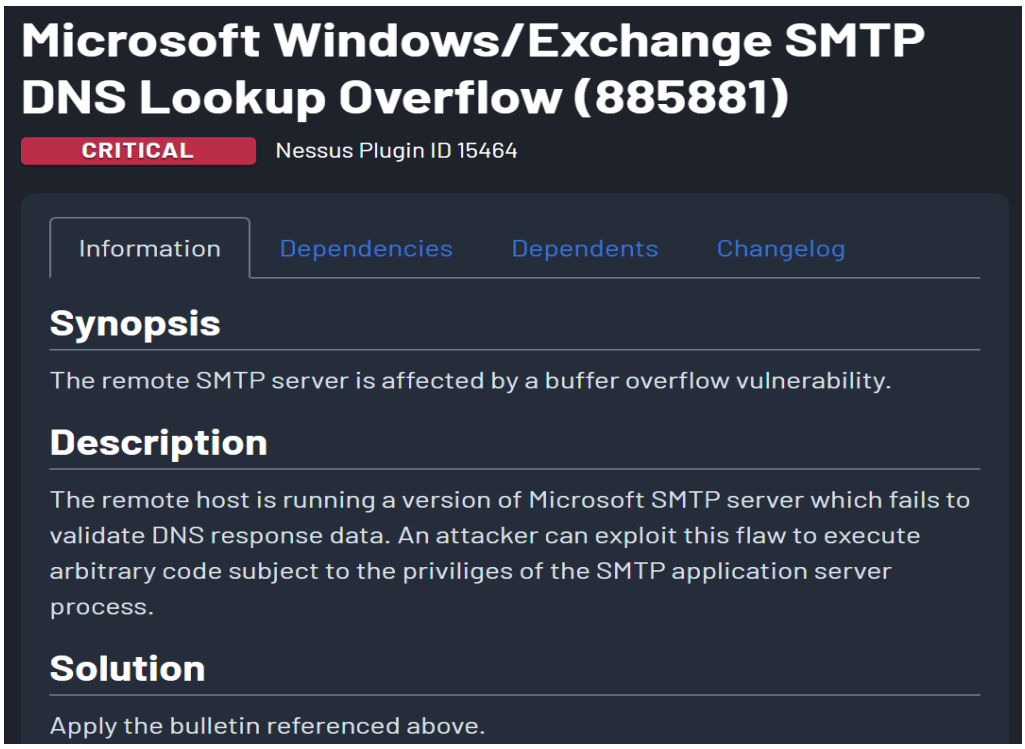
Software Version Used: The software versions are as identified in the open ports and running services, including Microsoft Windows RPC and Windows 7 Ultimate 7601 Service Pack 1, Microsoft-ds.

Identified Vulnerabilities: Based on the identified operating system and services, the following vulnerabilities are present:

1. **Unsupported Windows OS (remote)**

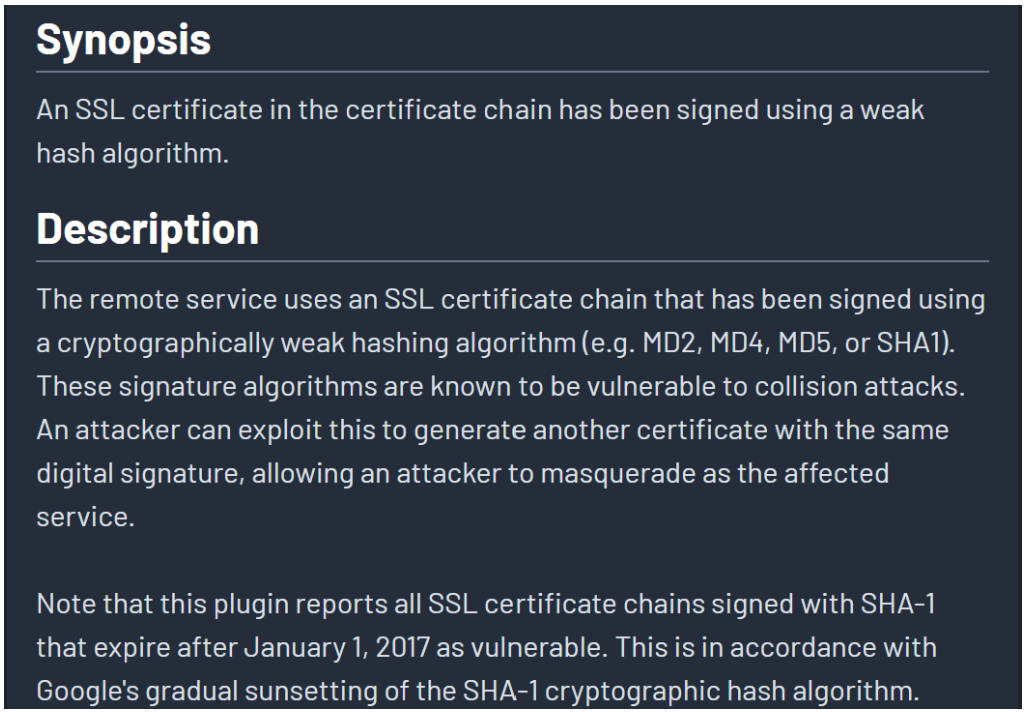


2. **Microsoft Windows/Exchange SMTP DNS Lookup Overflow (885881)**

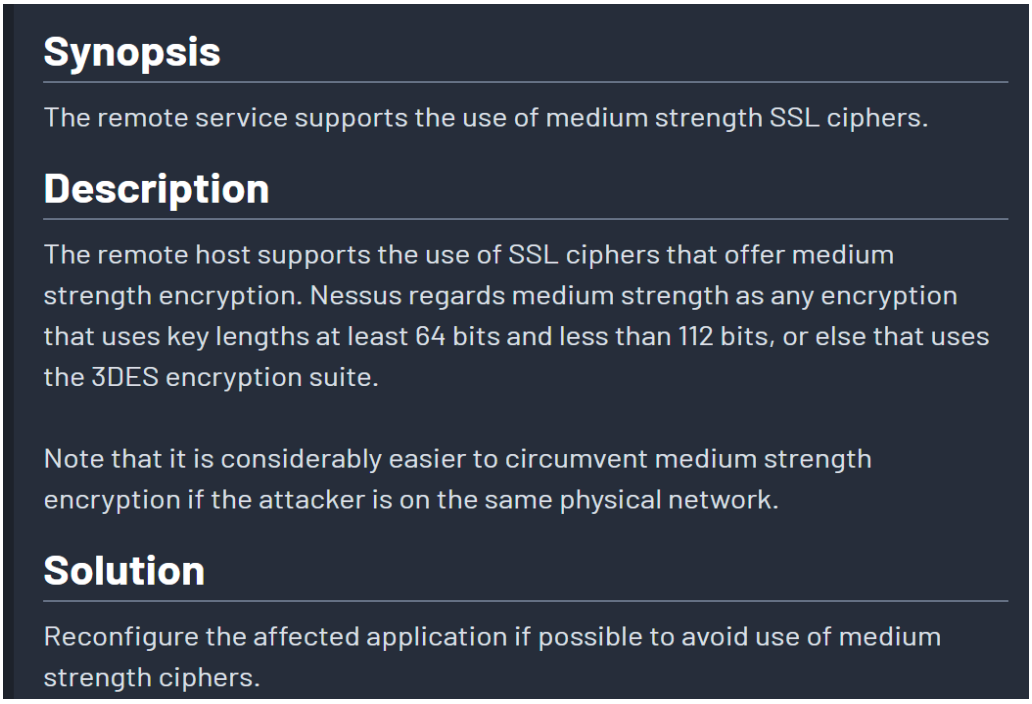


- **CVE:** CVE-2004-0840

3. **SSL Certificate Signed Using Weak Hashing Algorithm**
    - **CVE:** CVE-2004-2761



4. **SSL Medium Strength Cipher Suites Supported (SWEET32)**
    - **CVE:** CVE-2016-2183



5. **TLS Version 1.0 Protocol Detection**
    - **CVE:** Not Available

Synopsis

The remote service encrypts traffic using an older version of TLS.

Description

The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.

As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.

Solution

Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0.

6. **Unencrypted Telnet Server**

Synopsis

The remote Telnet server transmits traffic in cleartext.

Description

The remote host is running a Telnet server over an unencrypted channel.

Using Telnet over an unencrypted channel is not recommended as logins, passwords, and commands are transferred in cleartext. This allows a remote, man-in-the-middle attacker to eavesdrop on a Telnet session to obtain credentials or other sensitive information and to modify traffic exchanged between a client and server.

SSH is preferred over Telnet since it protects credentials from eavesdropping and can tunnel additional data streams such as an X11 session.

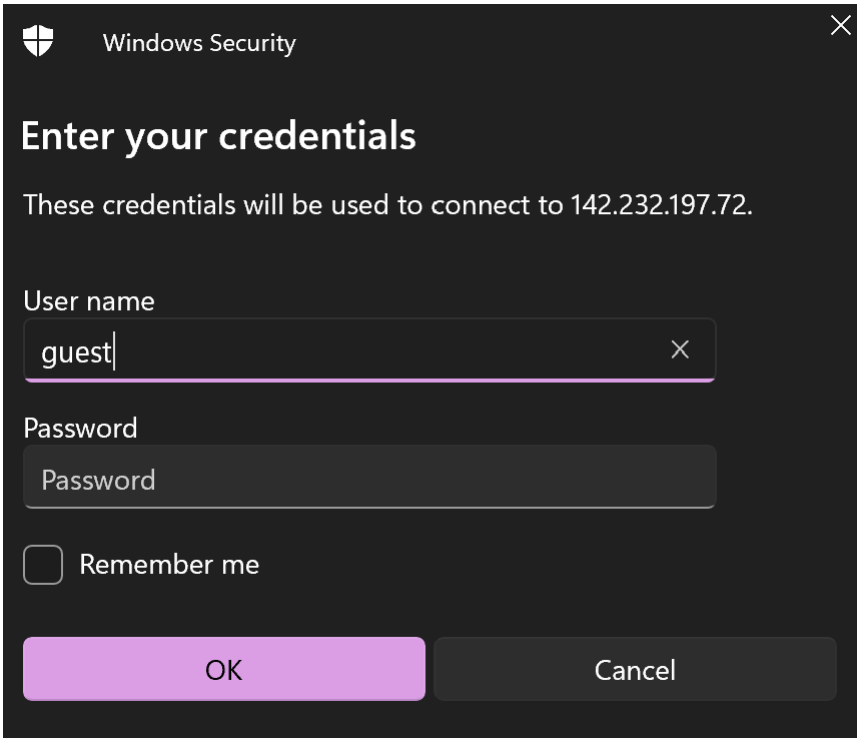
Solution

Disable the Telnet service and use SSH instead.

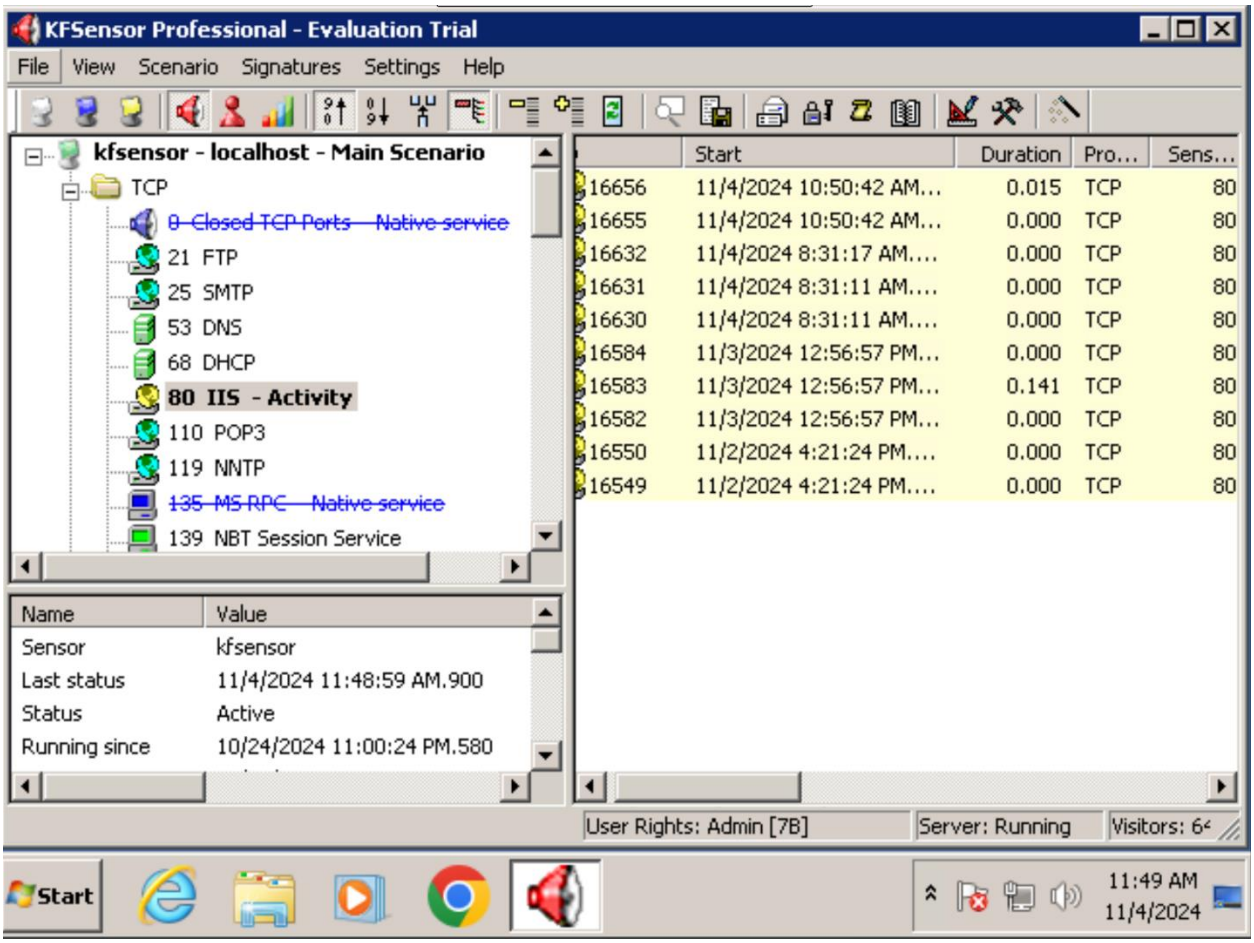
7. Phase 3: Reaching the Victim - Exploitation

7.4. Exploitation of 142.232.197.72 (Windows PC) via RDP

1. **Targeted Service:** Remote Desktop Protocol (RDP) identified as running on **Port 3389/tcp**.



2. **Credentials Used:** The credentials **ID: BCIT** and **Password: 112233**, which were discovered on the Mail Server (142.232.197.73), were used for authentication.
3. **Initial Connection Attempt:** An initial RDP connection attempt from a Windows 11 machine failed due to a Transport Layer error. This was identified as a TLS version incompatibility, with the Windows 11 RDP client attempting to use TLS 1.3 while the Windows 7 Ultimate target supported only TLS 1.2.
4. **Remediation & Successful Access:** To resolve the TLS incompatibility, **Remmina** (a remote desktop client) on a Kali Linux machine was used. The Transport Layer Security setting within Remmina was specifically configured to **TLS 1.2**.
5. **Outcome:** With the adjusted TLS setting, the RDP connection was successfully established using the BCIT: 112233 credentials, granting access to the Windows 7 Ultimate desktop.



Post-Access Discovery: Upon gaining access, it was confirmed that the Windows 7 Ultimate system was intentionally operating as a **KFSensor Honeypot**. This explains the high number of open and tcpwrapped ports identified during the earlier enumeration phase, as the honeypot simulates various services to detect and log suspicious activity.

6. Findings (Continued)

6.4. Target IP: 142.232.197.39 (Honeypot - T-Pot)

Operating System/Service Identification: The target system at 142.232.197.39 is identified as a **T-Pot Honeypot**, which simulates various operating systems and services. Specifically, connections to the SSH and Telnet services present a **Linux (Ubuntu)** operating system, while access through Port 4444 (Krb524 service) leads to a **Windows XP** environment.

Open Ports & Running Services: The following open and closed ports, along with their corresponding services and versions, were identified on 142.232.197.39:

- **Port 20/tcp:** closed - ftp-data
- **Port 21/tcp:** open - ftp - vsftpd 2.0.8 or later
- **Port 22/tcp:** open - ssh - OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
- **Port 23/tcp:** open - telnet
- **Port 25/tcp:** open - smtp - Exim smtpd 4.69
- **Port 32/tcp:** closed - unknown
- **Port 42/tcp:** open - nameserver?
- **Port 53/tcp:** closed - domain
- **Port 80/tcp:** open - http - aiohttp 3.8.6 (Python 3.11)
- **Port 81/tcp:** open - http - nginx
- **Port 110/tcp:** open - tcpwrapped
- **Port 113/tcp:** closed - ident
- **Port 135/tcp:** open - msrpc?
- **Port 139/tcp:** closed - netbios-ssn
- **Port 143/tcp:** open - tcpwrapped
- **Port 161/tcp:** closed - snmp
- **Port 199/tcp:** closed - smux
- **Port 255/tcp:** closed - unknown
- **Port 256/tcp:** closed - fwl-secureremote
- **Port 340/tcp:** closed - unknown

- **Port 443/tcp:** open - ssl/http - Apache httpd
- **Port 445/tcp:** open - microsoft-ds?
- **Port 554/tcp:** closed - rtsp
- **Port 587/tcp:** open - smtp - Exim smtpd 4.69
- **Port 8000/tcp:** open - http - TwistedWeb httpd 22.10.0
- **Port 993/tcp:** open - tcpwrapped

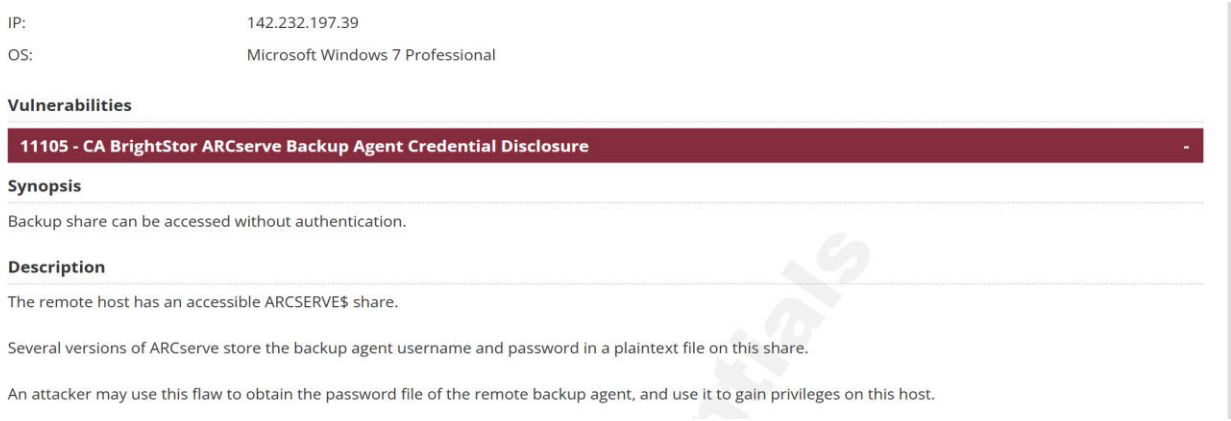
Software Version Used: The following software versions were identified from the scan:

- **vsftpd:** 2.0.8 or later
- **OpenSSH:** 8.9p1 Ubuntu 3ubuntu0.10
- **Exim smtpd:** 4.69
- **aiohttp:** 3.8.6 (Python 3.11)
- **nginx**
- **Apache httpd**
- **TwistedWeb httpd:** 22.10.0
- **Krb524** (on Port 4444/tcp, simulating Windows XP)

Identified Vulnerabilities: The following vulnerabilities were identified for 142.232.197.39:

1. **CA Brightstor ARCserve Backup Agent Credential Disclosure:**

- **Description:** The remote host has an accessible ARCSERVE$ share. Several versions of ARCserve store the backup agent username and password in a plaintext file on this share. An attacker may use this flaw to obtain the password file of the remote backup agent, and use it to gain privileges on this host.
- **CVE:** CVE-2001-0960
- **Plugin Output:** tcp/445/cifs



IP: 142.232.197.39

OS: Microsoft Windows 7 Professional

Vulnerabilities

11105 - CA BrightStor ARCserve Backup Agent Credential Disclosure -

Synopsis

Backup share can be accessed without authentication.

Description

The remote host has an accessible ARCSERVE$ share.

Several versions of ARCserve store the backup agent username and password in a plaintext file on this share.

An attacker may use this flaw to obtain the password file of the remote backup agent, and use it to gain privileges on this host.

2. **DoS Attack vulnerability**
   - **CVE:** CVE-2008-4834

35362 - MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (uncredentialed check) -

Synopsis

It is possible to crash the remote host due to a flaw in SMB.

Description

The remote host is affected by a memory corruption vulnerability in SMB that may allow an attacker to execute arbitrary code or perform a denial of service against the remote host.

See Also

http://www.microsoft.com/technet/security/bulletin/ms09-001.mspx

Solution

Microsoft has released a set of patches for Windows 2000, XP, 2003, Vista and 2008.

Risk Factor

Critical

VPR Score

7.4

EPSS Score

0.73

CVSS v2.0 Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

CVSS v2.0 Temporal Score

3. **MS04-007: ASN.1 Vulnerability Could Allow Code Execution**
   - **CVE:** CVE-2003-0818

12054 - MS04-007: ASN.1 Vulnerability Could Allow Code Execution (828028) (uncredentialed check) (NTLM

Synopsis

Arbitrary code can be executed on the remote host.

Description

The remote Windows host has an ASN.1 library that could allow an attacker to execute arbitrary code on this host.

To exploit this flaw, an attacker would need to send a specially crafted ASN.1 encoded packet with improperly advertised le

This particular check sent a malformed NTLM packet and determined that the remote host is not patched.

See Also

https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2004/ms04-007

Solution

Microsoft has released patches for Windows NT, 2000, XP, and 2003.

Risk Factor

Critical

4. **Network Time Protocol Daemon (ntpd) monlist Command Enabled DoS**
    - **CVE:** CVE-2013-5211

71783 - Network Time Protocol Daemon (ntpd) monlist Command Enabled DoS -

Synopsis

The remote NTP server is affected by a denial of service vulnerability.

Description

The version of ntpd running on the remote host has the 'monlist'
command enabled. This command returns a list of recent hosts that have connected to the service. However, it is affected by a denial of service vulnerability in ntp_request.c that allows an unauthenticated, remote attacker to saturate network traffic to a specific IP address by using forged REQ_MON_GETLIST or REQ_MON_GETLIST_1 requests.
Furthermore, an attacker can exploit this issue to conduct reconnaissance or distributed denial of service (DDoS) attacks.

See Also

https://isc.sans.edu/diary/NTP+reflection+attack/17300
http://bugs.ntp.org/show_bug.cgi?id=1532
https://kb.juniper.net/InfoCenter/index?page=content&id=JSA10613

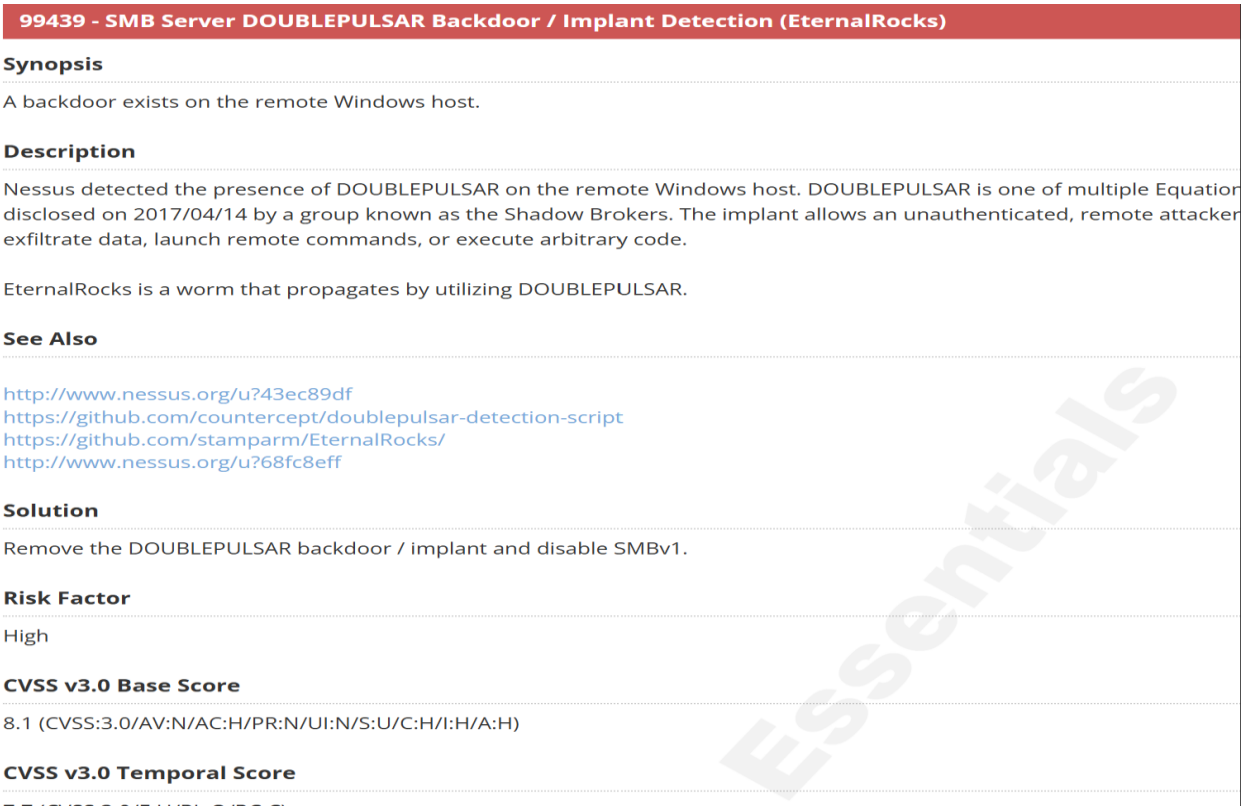
Solution

If using NTP from the Network Time Protocol Project, upgrade to NTP version 4.2.7-p26 or later. Alternatively, add 'disable monitor'
to the ntp.conf configuration file and restart the service. Otherwise, limit access to the affected service to trusted hosts, or contact the vendor for a fix.

Risk Factor

Medium

5. **SMB Server DOUBLEPULSAR Backdoor / Implant Detection (EternalRocks)**
    - **CVE:** CVE-2017-0144



99439 - SMB Server DOUBLEPULSAR Backdoor / Implant Detection (EternalRocks)

Synopsis

A backdoor exists on the remote Windows host.

Description

Nessus detected the presence of DOUBLEPULSAR on the remote Windows host. DOUBLEPULSAR is one of multiple Equation disclosed on 2017/04/14 by a group known as the Shadow Brokers. The implant allows an unauthenticated, remote attacker exfiltrate data, launch remote commands, or execute arbitrary code.

EternalRocks is a worm that propagates by utilizing DOUBLEPULSAR.

See Also

http://www.nessus.org/u?43ec89df
https://github.com/countercept/doublepulsar-detection-script
https://github.com/stamparm/EternalRocks/
http://www.nessus.org/u?68fc8eff

Solution

Remove the DOUBLEPULSAR backdoor / implant and disable SMBv1.

Risk Factor

High

CVSS v3.0 Base Score

8.1 (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVSS v3.0 Temporal Score

7. Phase 3: Reaching the Victim - Exploitation

7.5. Exploitation of 142.232.197.39 (Honeypot - T-Pot)

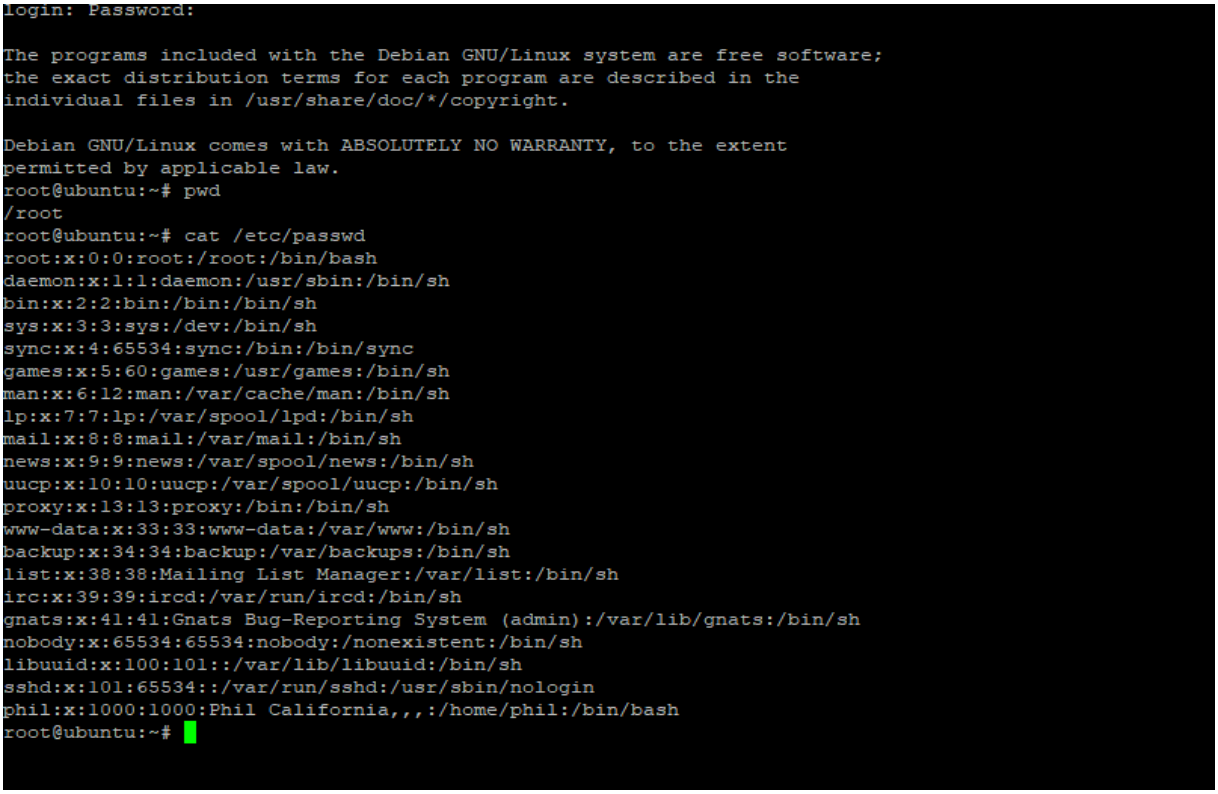
Multiple methods were successfully used to gain access to the T-Pot honeypot, demonstrating its susceptibility to common credential-based attacks. The honeypot presented different operating system environments based on the service accessed. All exploitations utilized guessable or default credentials.

7.5.1. SSH Access (Linux - Ubuntu)

1. **Targeted Service:** Secure Shell (SSH) on Port 22/tcp.
2. **Method:** A connection was attempted using SSH, a common network utility, from the attacker machine.
3. **Credentials Used:** A default or guessable administrator password, "toor", was successfully used for authentication.
4. **Outcome:** Root access to a simulated Linux (Ubuntu) operating system was achieved via the SSH service.
5. **Proof of Concept:**

7.5.2. Telnet Access (Linux - Ubuntu) - Method 1

1. **Targeted Service:** Telnet on Port 23/tcp.
2. **Method:** The PuTTY software was used to establish a Telnet connection.
3. **Credentials Used:** The password "password" was successfully used to authenticate.
4. **Outcome:** Shell access to a simulated Linux (Ubuntu) operating system was gained.
5. **Proof of Concept:**



7.5.3. Telnet Access (Linux - Ubuntu) - Method 2 (User "Phil")

1. **Targeted Service:** Telnet on Port 23/tcp.
2. **Method:** During post-exploitation enumeration, the /etc/passwd file revealed a user named "Phil". The corresponding entry in the /etc/shadow file was decrypted.
3. **Credentials Used:** The decrypted password for user "Phil" was "admin123". These credentials were then used with PuTTY to establish a new Telnet connection.
4. **Outcome:** Successful login as user "Phil" to the simulated Linux (Ubuntu) operating system was achieved.
5. **Proof of Concept:**

```
login: Password:
               Login incorrect
                                login: password
Password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
password@ubuntu:~$ admin123
-bash: admin123: command not found
password@ubuntu:~$ cd
password@ubuntu:~$ pwd
/home
password@ubuntu:~$
```

7.5.4. Shell Connection to Port 4444 (Windows XP)

1. **Targeted Service:** Krb524 service on Port 4444/tcp.
2. **Method:** A direct netcat connection was established to Port 4444/tcp.
3. **Outcome:** Upon connection, a shell representing a **Windows XP** environment was presented, indicating a compromised state or a simulated backdoor. This allowed interaction with a different simulated operating system within the honeypot.
4. **Proof of Concept:**

```
          [-c cipher_spec] [-D [bind_address:]port] [-E log_file]
          [-e escape_char] [-F configfile] [-I pkcs11] [-i identity_file]
          [-J destination] [-L address] [-l login_name] [-m mac_spec]
          [-O ctl_cmd] [-o option] [-P tag] [-p port] [-R address]
          [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
          destination [command [argument ...]]
       ssh [-Q query_option]

┌──(root㉿kali)-[/home/kali]
└─# ssh -p 4444 user@host
ssh: Could not resolve hostname host: No address associated with hostname

┌──(root㉿kali)-[/home/kali]
└─# nc 142.232.197.39 4444
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>

┌──(root㉿kali)-[/home/kali]
└─# nc 142.232.197.39 4444
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

8. Tools Used

The following tools were utilized throughout this penetration testing engagement:

- **Reconnaissance:**
    - **Nessus:** For comprehensive vulnerability scanning and initial host discovery.
    - **Nmap:** For network scanning, port enumeration, service identification, and OS detection.
- **Vulnerability Analysis:**
    - **Nessus:** Used extensively for identifying known vulnerabilities and misconfigurations across target systems.
- **Exploitation:**
    - **Burp Suite:** A web penetration testing framework, likely used for web application vulnerabilities (e.g., SQL Injection, although SQL Injection itself is a type of attack, not a tool, it indicates the use of tools capable of performing it).
    - **Remmina:** A remote desktop client used for successful RDP access to 142.232.197.72.
    - **Metasploit Framework:** A powerful platform for developing, testing, and executing exploits.
    - **Hashcat:** A password recovery tool, likely used for cracking hashes obtained during the assessment.
    - **John the Ripper:** Another robust password cracking utility, used for similar purposes as Hashcat.

- **Netcat:** A versatile networking utility, specifically mentioned for establishing shell connections.
- **SQL Injection:** While not a tool, this refers to the technique used for exploiting database vulnerabilities, implying the use of tools capable of performing such attacks (e.g., Burp Suite, SQLM

---

9. Conclusion

This penetration testing engagement, conducted on the specified network assets within the campus environment, successfully identified a range of security vulnerabilities and exploitable weaknesses. The primary objective of simulating real-world attack scenarios was achieved, providing critical insights into the current security posture of the in-scope systems.

Key findings across the targeted systems underscore common security deficiencies, primarily related to outdated software, weak default configurations, and the prevalence of guessable or easily discoverable credentials:

- **142.232.197.67 (Ubuntu Web Server):** This system was identified running a Linux Ubuntu Server with an Apache HTTP Server hosting the Damn Vulnerable Web Application (DVWA). Critical vulnerabilities in the Apache HTTPD version (2.4.x < 2.4.60/59/62) were noted, including potential for SSRF, DoS, and code execution. A direct **SQL Injection** exploit was successfully performed against DVWA, demonstrating the ability to extract sensitive user credentials, such as Pablo:letmein, confirming the practical impact of web application flaws. The specific DVWA version was not identified.
- **142.232.197.73 (Mail Server):** Identified as a Debian Linux Mail Server, this system presented an extensive attack surface with numerous open ports and outdated services. Critical vulnerabilities such as vsftpd 2.3.4 (UnrealIRCd Backdoor), and easily exploitable services like VNC with a weak password, and a direct root **bindshell on Port 1524**, were discovered. Successful **VNC remote access** was gained using the default password "password". A **direct root shell** was established via netcat to Port 1524, confirming the presence of a severe backdoor. This post-exploitation access further led to the discovery of critical credentials (BCIT:112233) from the mywindows7.txt file, enabling access to other network segments.

- **142.232.197.72 (Windows PC - KFSensor Honeypot):** This system, identified as a Microsoft Windows 7 Professional PC, was confirmed to be operating as a KFSensor Honeypot. Despite its nature, it presented an exploitable **Remote Desktop Protocol (RDP)** service on Port 3389. Utilizing the BCIT:112233 credentials obtained from the Mail Server, and overcoming TLS version incompatibility issues with Remmina on Kali Linux (by configuring TLS 1.2), **successful RDP access** was achieved, demonstrating that even honeypots designed for deception can be compromised through credential reuse.
- **142.232.197.39 (Honeypot - T-Pot):** As a T-Pot honeypot, this system simulated multiple environments. **Four distinct methods of unauthorized access** were successfully demonstrated through the exploitation of weak or default credentials:
    1. **SSH Access:** Root access to a simulated Linux (Ubuntu) environment was gained using the default password "toor".
    2. **Telnet Access (Method 1):** Shell access to a simulated Linux (Ubuntu) environment was achieved via PuTTY using the simple password "password". A directory named "JAHIN" was successfully created as proof.
    3. **Telnet Access (Method 2):** After decrypting the shadow file to obtain the password "admin123" for user "Phil", a successful Telnet login was performed with these credentials.
    4. **Port 4444 Shell Connection:** A direct netcat connection to Port 4444 provided a direct shell to a simulated Windows XP environment. These multiple compromises highlight the critical risk posed by guessable or default credentials, regardless of the underlying system's purpose.

The findings collectively indicate a significant attack surface across the tested environment, largely attributable to unpatched software, weak default configurations, and poor password hygiene. These vulnerabilities present considerable risks to the confidentiality, integrity, and availability of the systems.

It is imperative that the identified vulnerabilities are addressed promptly and systematically. The detailed recommendations provided throughout this report outline actionable steps for remediation. A commitment to implementing these recommendations, coupled with ongoing security monitoring, regular patching cycles, and robust password policies, will significantly enhance the overall security posture and resilience against future cyber threats.