IMPLEMENTING A NETWORK SECURITY LABORATORY AT UNIVERSITY UNDERGRADUATE LEVEL

Umut GULEC[1], Valeriu Manuel IONESCU[2]
[1]University of Selcuk, Turkey
[2]University of Piteşti, Romania
umtglc99@gmail.com

Keywords: Network Security, SQL Injection, Mutillidae, Burp Suite, SQL Mapping, Cyber Security.

Abstract: *In universities that teach computer network courses, the main focus is placed on identifying the security threats and placing the correct countermeasures for preventing security attacks. There are only few universities that analyze how an attack is actually performed in order to allow the students to draw their conclusions on the resources, skills and motivation for making that attack successful. This paper presents the steps used in creating a hands-on security laboratory by using Mutillidae (as a vulnerable website), Burp Suite (as attacking tools) and SQLmap for attack automation, with accent on the SQL Injection method.*

1. INTRODUCTION

Network security undergraduate classes in universities [1] draw a lot of attention from students as this is an area with much visibility in recent years and gives, for a long time, the chance of obtaining a high paying job [13].

Network security policies try to prevent unauthorized access and usage of network resources. Understanding how an attack is performed is sometimes the key on protecting from those attacks. Some students are even attracted more by security penetration than security prevention measures. This can be used to attract students to study the network security area.

Universities themselves are targets of network security attacks especially because they rely on older technologies due to insufficient staff and funding [2]. Students can help improve the campus security, build positive experience and a good resume by sharing their undergraduate responsibilities.

This article shows presents the structure of a laboratory in network security that uses readily available tools for building vulnerable targets and creating successful attacks. Students are therefore taught network security by live practice. Finally the students are required to determine the implementation errors, correct them and develop a security policy that will mitigate apparent weaknesses.

In this paper we will be using Mutillidae as a vulnerable website and as the attacking tools we will be using two suites: SQLmap and Burp Suite. The focus of this lab will be SQL Injection and the target time will be four hours.

Mutillidae [3], developed by Open Web Application Security Project (OWASP) [4], is a free web application that includes vulnerabilities that can be targeted by security researchers and developers. It has an easy to use web interface and can be manually installed on a Linux or Windows system or can be directly downloaded as there are many pre-built virtual environments with tools and vulnerable apps. As an alternative there is the Metasploitable Framework (from Rapid 7 [5]) - a pre built Linux virtual machine with many vulnerabilities and Core Impact (by Core Security [5]) – a vulnerability assessment tool with a large corporate user base but with important parts available only as paid components. For a laboratory setup Mutillidae is free, easy to install on multiple operating systems and was chosen to be used in this paper.

Burp Suite has many alternatives, with similar functionality. Burp Suite is a freemium tool (meaning that some parts are free while the advanced parts need to be paid for); for a completely free alternative mitmproxy [6] can be used. However, Burp Suite's "Free" part will be sufficient for the purpose of this paper with the advantage of having a better GUI.

SQLmap [7] is an automated penetration testing tool that is used to perform SQL database attacks with multiple DB types supported. As this paper will target the SQL injection attacks, it is the second tool used to affect the security of the target systems. An alternative to SQLmap is for example Netsparker, a commercial security scanner used by Ford, Cisco and Starbucks. Due to the cost however, SQLmap is the preferred tool.

The reminder of the paper is organized as follows: The Mutillidae web application along with the XAMPP installation are presented in Chapter 2; a simple SQL injection attack is presented in Chapter 3; SQLmap installation is presented in Chapter 4; Burp suite install and integration with Firefox is presented in Chapter 5; the automation of SQL injection using SQLmap is presented in Chapter 6; finally conclusions are presented in Chapter 7.

2. INSTALLING MUTILLIDAE ON WINDOWS

Mutillidae can easily be installed on either Linux or Windows operating systems. In this paper we will install in Windows operating system. As Mutillidae is a web app, it is necessary to download and install a web server and a database, in this case we chose XAMPP (which stands for Apache, MySQL, PHP, and Perl while the X at the beginning indicates that this application is cross-platform). Here, Mutillidae will use:

-Apache as a web server,
-MySQL as a database,
-PHP as the programming language.

A. XAMPP installation

After downloading XAMPP from: https://www.apachefriends.org/download.html, and choosing the latest version from the list: 7.4.1, it is necessary to install the software.

We can continue installing XAMPP by opening he downloaded file, selecting Next button and in the next window, check all of the default components as seen in Fig. 1. Critical for Mutillidae operation are only Apache and MySQL.

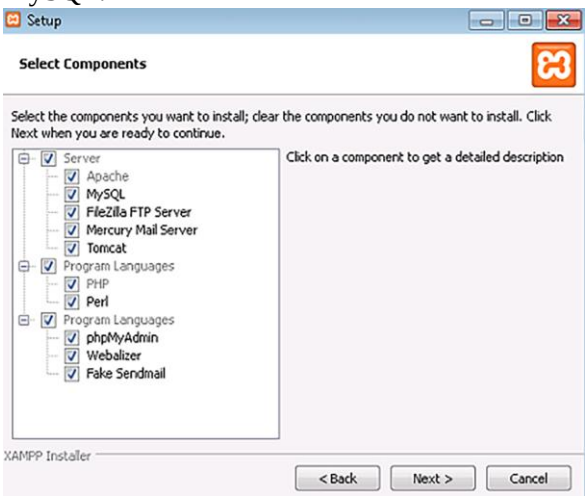

Fig. 1 XAMPP Components

In the next step, a default path (C:\xamp\htdocs) is proposed for the XAMPP installation that we can accept if it is unused or change it as necessary. In the reminder of the paper we will assume the default path. After the installation has completed, start the XAMPP Control Panel so we can launch the services needed to install Mutillidae.

Start the Apache and MySQL services by clicking on their Start buttons as seen in Fig 2.

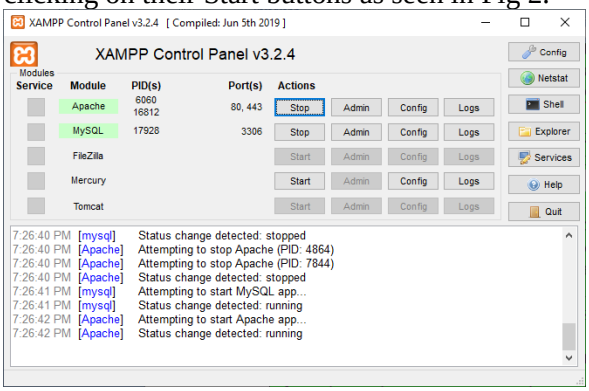

Fig. 2 XAMPP Control Panel with Apache and MySQL seervices started

B. MUTILLIDAE INSTALLATION

You can download Mutillidae from https://sourceforge.net/projects/mutillidae/. A possible problem for students is that Windows 10 makes compressed folders look like normal folders but sometimes the installer will not work

correctly if run from inside the compressed folder; therefore it is recommended that after downloading, to extract the compressed archive file to C:\xamp\htdocs folder.

In order to access the Mutillidae site from the LAN, we will need to adjust the configuration file ".htaccess". Open the Mutillidae folder that you just extract and open the .htaccess file as seen in the Fig. 3 using Notepad.

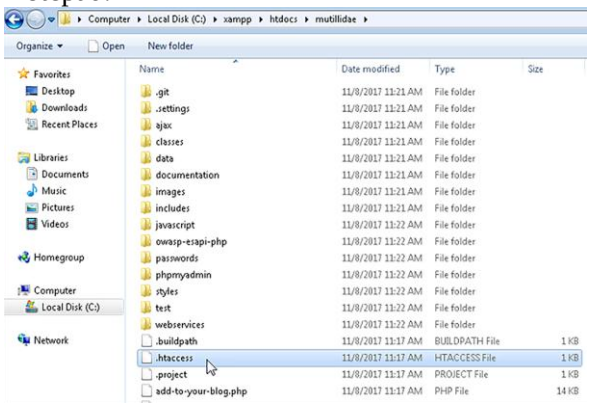

Fig. 3 The .htaccess file location

If you want to access Multillidae from the local network (in this case the network is 192.168.100.0/24 that is not in the default Multillidae list), you must Add the following line to allow section:

Allow from 192.168.100.0/24

Where *192.168.100.0/24* can be the network address, domain or a partial IP address (the first numbers of the IP address) [8]. An address can be added as a partial address for briefness "Allow from 192.168.100." in the allow section as in the Fig. 4 [10].

```
ErrorDocument 403 "By default, Mutillidae only allows access
from localhost (127.*.*.*). Edit the .htaccess file to change
this behavior (not recommended on a public network)."
Order Deny,Allow
Deny from all

## This allows access from localhost
Allow from 127.
Allow from localhost

## This is to allow access from other machines on Virtual Box
host-only networks.
Allow from 192.168.0.0/16
Allow from 192.168.100.
```

Fig. 4 changed .htaccess text file

Next, navigate to http://[your machine IP]/mutillidae (or localhost/mutillidae) with your browser. After the page loads, click on the setup/reset link to install the DB tables, and Mutillidae will be configured. If the process works correctly, you will be told that no errors were detected when resetting the database and you will be directed to the Mutillidae start page as seen in Fig. 5.

If there are messages displayed that the root user does not have access to install the Mutillidae database tables (as seen in Fig. 6, an error common in more recent versions of XAMPP), a simple fix is to add in the my.ini configuration file of the MySQL (found by pressing the *Config* button in the *XAMPP Control Panel*), the line *skip-grant-tables* after the [mysqld] line. This will allow execution without restrictions of the SQL commands.

To apply the changes it is necessary to Stop MySQL service, make the changes, then Start the MySQL service from the *XAMPP Control Panel* (Fig. 2).

Fig. 5 Mutillidae main page

Fig. 6 Mutillidae error due to MySQL configuration permissions

3. SQL INJECTION ATTACK TARGETING MUTILLIDAE

SQL injection is a technique used to extract data from a DB that the user would not normally have access to. This is based on the fact that many web sites need to get user data but they do not parse it correctly before entering it in a SQL query.

This gives the opportunity for an attacker to insert specially formatted text in the field passed to the server so that it will execute a different query and bypass the usual security checks (such as user rights or password).

The actual implementation consists of inserting an escape character into the web server

query followed by the actual query and finally by placing a comment in the final part (in order to avoid errors that could appear by running the reminder of the DB code). The data returned by the query will return some unauthorized information such as usernames, passwords, server information and other sensitive data.

In the following section will implement a SQL Injection targeting the Mutillidae page "User Info" page (Fig. 7).

The path to User Info Page is: OWASP 2017 => A1 - SQL Injection => SQLi - Extract Data => User Info (Fig. 8).

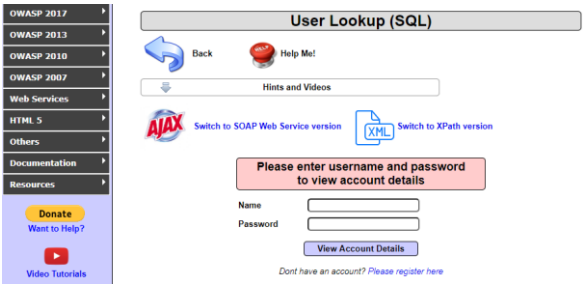

Fig. 7 The User Info page

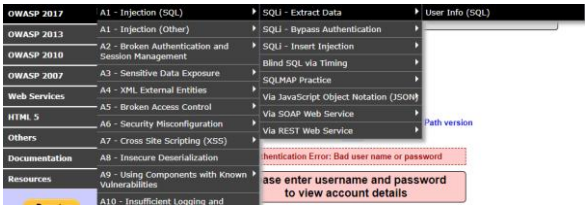

Fig. 8 Menu to reach The User Info page

First, we want to know how the query looks like to be able to manipulate it. Causing errors intentionally on the page can often reveal that information to us.

On the "name" field enter an apostrophe (') and submit the query. This will cause an error and give you an output looking like Fig. 9.

We can determine the SQL query that is used from the error table. In a proper login query, we are querying if there is any match with Username AND Password that we enter from database.

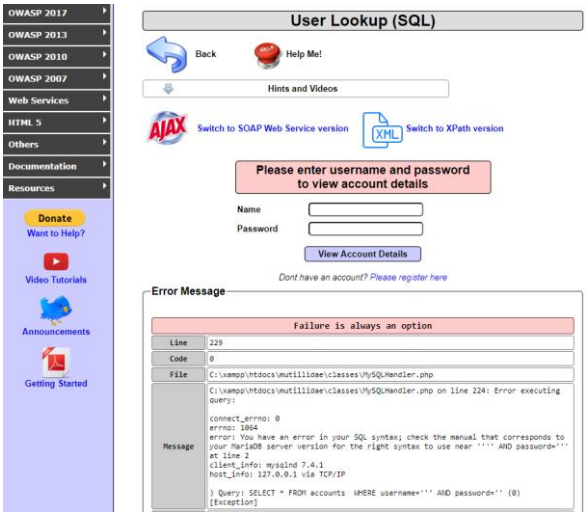

Fig. 9 Return of the input, an SQL error

If both are found and are correct (true), the query will return True (as 1 AND 1 gives TRUE) and the user will be able to log in:

*SELECT * FROM accounts WHERE username='yourUsername' AND password='yourPassword'*

This statement will only be true if you know both the username and password. However since we can use special characters, we can simply change the query as we wish. We can add comments (--) and an "OR" SQL operator to make the statement TRUE without needing the username and password. Such as:

*SELECT * FROM accounts WHERE username='' **OR 1=1** -- password=''*

Here, we are sending a blank username field which will return false but the use of the "OR" operator tells the statement that only one of the operators needs to be true in order for the query condition to be true, and "1=1" is a true condition. The "--" makes comment of the rest of the statement.

To make the query like this we will enter " ' OR 1=1 -- " as an input to Username field.

And the results should look like Fig. 10.

The next step is to automate this process.

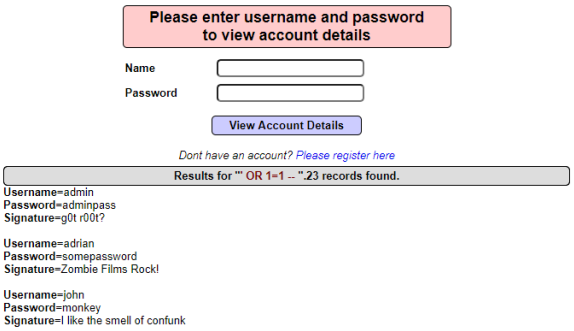

Fig. 10 Result of the modified statement

4. INSTALLING SQLMAP

Most of the time if you want to automate a process, writing a script is sufficient. In this paper SQLmap will be used to automate our SQL injection attacks and the script was written by Bernardo Damele A. G. and Miroslav Stampar. SQLMap is detecting and exploiting SQL injection flaws and takes over the databases.

Since SQLmap is written in Python, the first thing we need is the Python interpreter. Download and install the Python interpreter from python.org (it is recommended to use 2.6 or 2.7, not version 3 – however in 2019 Mutillidae support was improved for version 3 [14]), next download the SQLmap zip file from sqlmap.org. Extract the zip files to a directory (we chose c:\sqlmap). To test if you install it properly launch the DOS prompt and navigate to the directory where SQLmap was unzipped (in this case c:\sqlmap, where the file sqlmap.py is found).

cd C:\ sqlmap

Now run the sqlmap.py script with the python interpreter as follows:

python sqlmap.py

The result should give an error from executing the Python code, but this confirms that Python is installed correctly, as seen in Fig.11.

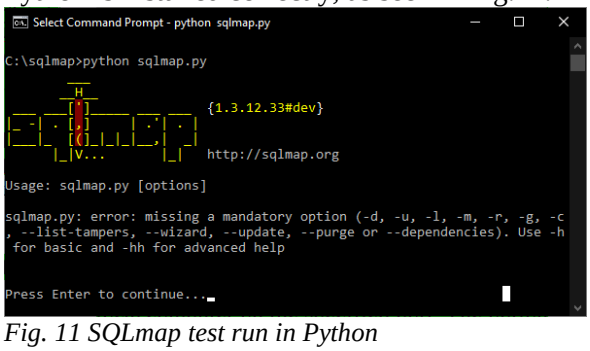

Fig. 11 SQLmap test run in Python

5. INSTALLING BURP SUITE

The first step is to download and install Burp Suite Community Edition from: https://portswigger.net/burp/communitydownload, in this case v2.1.07. After the install, create a Temporary project, with Default values.

Next, we must configure Firefox to work with Burp Suite [9] by going to the Firefox Menu by accessing “Preferences” / "Options" (altenate access method by pressing ALT key, then pressing *Tools* and afterwards *Options*). In the *General* section scroll to the “Network Proxy” settings (at the bottom of the page). Click on the "Settings" button as marked in Fig. 12.

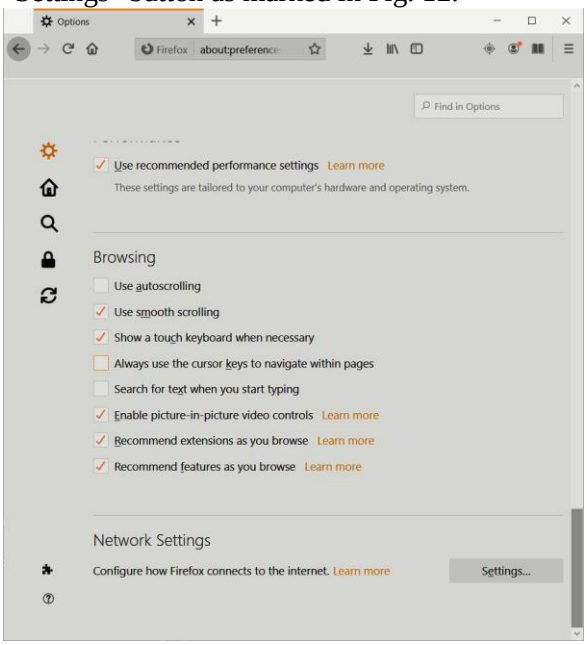

Fig. 12 Firefox proxy General setting

Here select "Manual proxy configuration" option, where we will specify the Burp Proxy listener address in the "HTTP Proxy" field (default set to 127.0.0.1) the port (default value 8080) and the option "Use this proxy server for all protocols" needs to be checked as seen in Fig. 13. This will make all the traffic from Firefox be diverted to the proxy found on the local computer on the specified port (where Burpsuite will be found).

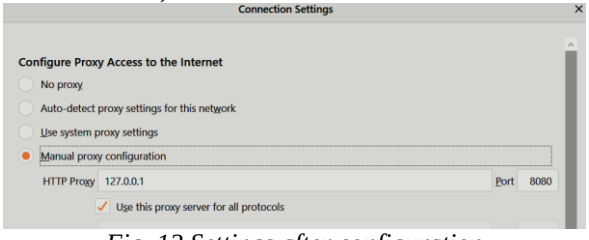

Fig. 13 Settings after configuration

Finally, as Burpsuite does work only with absolute URLs (Fig. 14), it is necessary to associate the *localhost* with a name on the local computer. This can be done by updating the hosts file (by default found in C:\Windows\System32\drivers\etc) with the line:

127.0.0.1 example.com

This means that all calls to the name *example.com* will end up in 127.0.0.0 network.

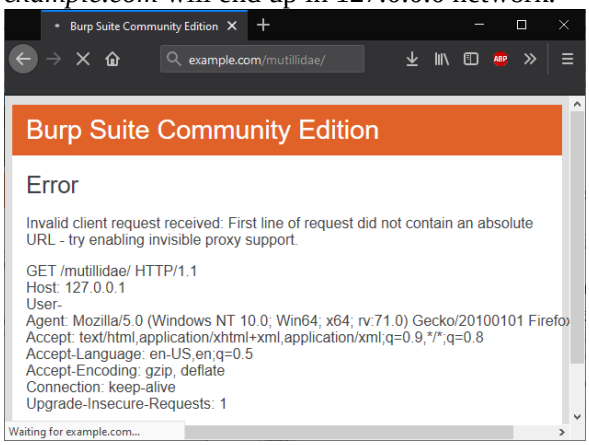

Fig. 14 Burp Suite proxy is functional for absolute URLs (in this case example.com)

6. AUTOMATE SQL INJECTION USING SQLMAP TO DUMP CREDIT CARDS TABLE IN MUTILLIDAE

We will target SQLmap against the Mutillidae view-blog-entries.php page [11, 12]. We will automate the SQL injection attack and will set up SQLmap by taking a request from Burp Suite and feeding it to SQLmap through the use of the -r (request) parameter. We will first find the names of the databases, then the tables, and finally dump the credit-cards table.

In the first step, in the Burp Suite, we go proxy tab, then intercept tab and make sure intercept is on like in Fig. 15.

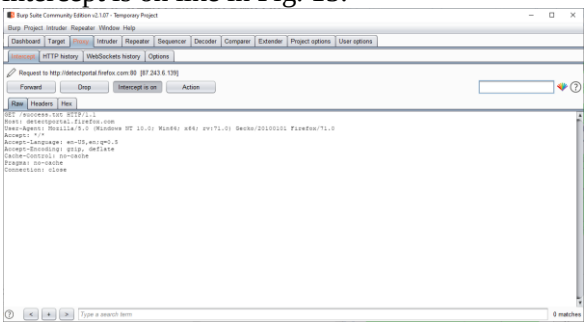

Fig. 15 Burp Suite intercept page

Finally in the *Options* tab of the *Proxy* section, the Intercept client requests needs to be configured as seen in Fig. 16:

- File extension – does not match the specified list (image formats, CSS files, etc.)
- AND HTTP method – matches GET or POST
- AND Domain name matches example.com

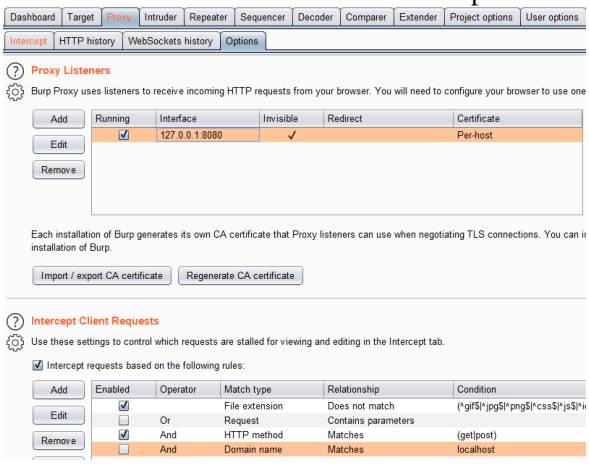

Fig. 16 Proxy Options configuration in Burp Suite

We tested before that the site is vulnerable to the SQL injection attacks, therefore we will need the request path that we will pass to SQLmap. We will intercept the request from http://localhost/mutillidae/index.php?page=view-someones-blog.php, as seen in Fig. 17.

In the Burp Suite, we go proxy tab and then intercept tab, make sure intercept is on. Then make request to view blog (select the View Log Entries button in the web page). As it needs to send data to the server, it will be a POST HTTP request. For all the queries that are not POST we will press the *Forward* button. When to the POST with view-blog-entries.php is found, we copy the content from the *Raw* tab into a file 1.txt placing it in *C:\sqlmap* install folder.

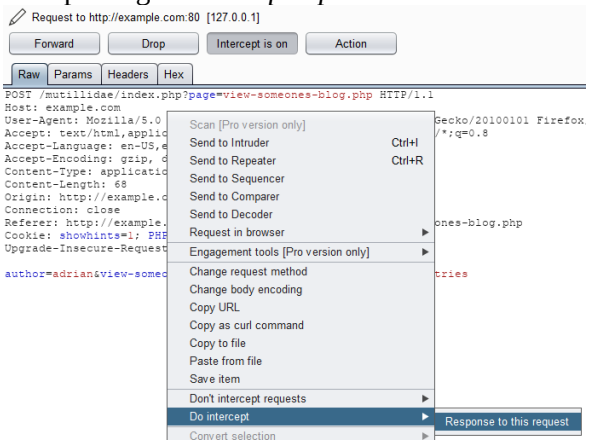

Fig. 17 Intercepted request

To discover the databases on the vulnerable site, we run the following command line in the SQLmaps's folder path:

python sqlmap.py -r 1.txt -threads=2 -dbs

After a few questions to the user reducing the investigation options, the tables in the database are identified as seen in Fig. 18.

Selecting the "mutillidae" database, we can identify its tables with the command in Fig. 19.

python sqlmap.py -r 1.txt -threads=2 -D mutillidae -tables

In here we can extract the data in CSV format from the table named *credit_cards* with the command in Fig. 20:

python sqlmap.py -r "the file path" –threads=2 -D owasp10 –T credit_cards --dump

The attack is now complete and the data is in the file *credit_cards.csv*

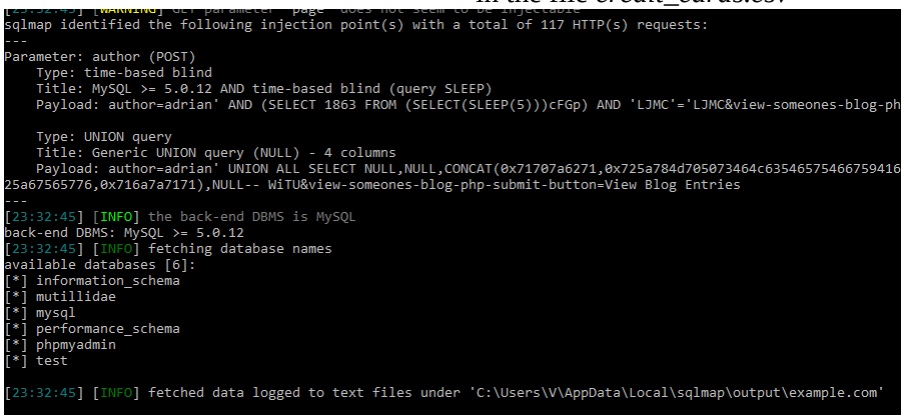

Fig. 18 Databases found by SQLmap on the vulnerable site

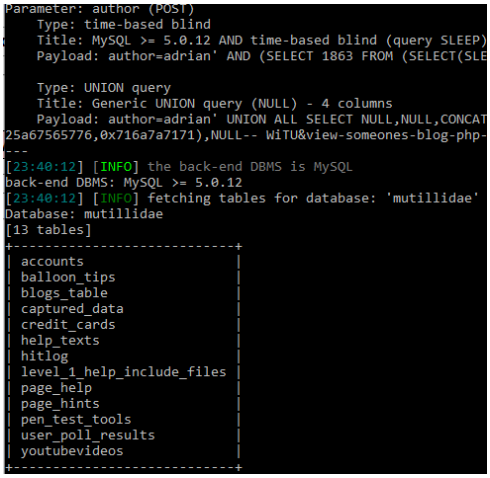

Fig. 19 Tables from mutillidae database

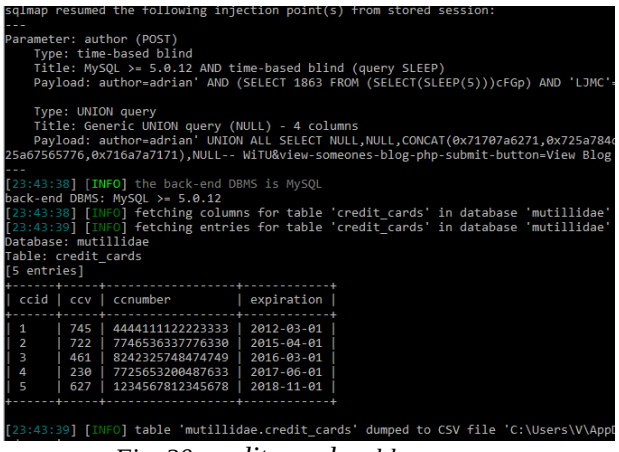

Fig. 20 credit_cards table contents

7. CONCLUSIONS

The paper presented the implementation of a network security laboratory for undergraduate students. Students from Computer Science and Computer Networks specializations are prime candidates for following this laboratory as they have the knowledge and understanding of the theoretical concepts involved.

The attacks presented include software installation, attack implementation and attack automation. These are intended to show the attack steps from simple attacks to more complex ones.

Problems were encountered and solved during the attack implementation and automation, as most existing guides found in the documentation are for older versions of the software, and do not discuss incompatibilities between different pieces of software. As security assessment tools are in constant development, in order to implement the most recent techniques, the students will need to learn that they will have to improvise and use additional knowledge in order to finalize an attack. Furthermore, with new software versions, older attack methods can become obsolete or non functional. Therefore the network and database security audit tools will have to be updated constantly for persons working in this domain.

The students will also have to consider techniques for countering these attacks such as strong typing variables, correctly parsing user data or limiting

Practicing an attack on a known vulnerable site will highlight the attack steps such as vulnerability detection, attack implementation, data extraction and process automation but will show that more complex attacks on unknown targets need better tools and a broader knowledge base from the attacker.

While the theory of implementing an SQL injection attack can be taught in several minutes the attack themselves take a longer time and involve the use of software suites that require even more time to master.

8. REFERENCES

[1] C. Irvine et al. Security Education and Critical Infrastructures, Ed. Springer Science+Business Media New York 2003, DOI: 10.1007/978-0-387-35694-5_2

[2] Don Carfagno. Why Is Higher Education the Target for Cyber Attacks?, June 24th, 2019, web: https://www.blackstratus.com/why-is-higher-education-the-target-for-cyber-attacks/, Accessed: 12.12.2019

[3] Jeremy Druin. OWASP Mutillidae 2 Project, web: https://www.owasp.org/index.php/OWASP_Mutillidae_2_Project, Accessed: 12.12.2019

[4] OWASP Foundation. Flagship Projects, 1 October 2019, web: https://www.owasp.org/, Accessed: 12.12.2019

[5] UpGuard. Core Security vs Rapid7 for Continuous Security, December 12, 2019, web: https://www.upguard.com/articles/core-security-vs-rapid7, Accessed: 12.12.2019

[6] Mitmproxy Project, "How mitmproxy works" web: https://docs.mitmproxy.org/stable/concepts-howmitmproxyworks/, Accessed: 12.12.2019

[7] Bernardo Damele A. G., Miroslav Stampar. sqlmap Automatic SQL injection and database takeover tool, web:http://sqlmap.org/, Accessed: 12.12.2019

[8] The Apache Software Foundation, "Apache Module mod_authz_host", web: http://httpd.apache.org/docs/2.2/mod/mod_authz_host.html#allow, Accessed: 12.12.2019

[9] PortSwigger Ltd., "Configuring Firefox to work with Burp", web: https://support.portswigger.net/customer/portal/articles/1783066-configuring-firefox-to-work-with-burp, Accessed: 12.12.2019

[10] Packt | Programming Books, eBooks & Videos for Developers, "Installing Mutillidae on Windows", accessed 13.12.2019: https://subscription.packtpub.com/book/networking_and_servers/9781788624039/1/ch01lvl1sec12/installing-mutillidae-on-windows

[11] ComputerSecurityStudent, "(Mutillidae: Lesson 12)", web: https://computersecuritystudent.com/SECURITY_TOOLS/MUTILLIDAE/MUTILLIDAE_2511/lesson12/index.html/, Accessed: 12.12.2019

[12] ComputerSecurityStudent, "(Mutillidae: Lesson 8)", web: https://www.computersecuritystudent.com/SECURITY_TOOLS/MUTILLIDAE/MUTILLIDAE_2511/lesson8/index.html/, Accessed: 12.12.2019

[13] Exabeam, "EXABEAM 2018 CYBER SECURITY PROFESSIONALS SALARY AND JOB REPORT", web: https://www.exabeam.com/wp-content/uploads/2018/05/EXA_Salary-Survey-Report_L1R6.pdf, Accessed: 12.12.2019

[14] Jeremy Druin. Commits, https://github.com/webpwnized/mutillidae/commits/master, Accessed: 26.12.2019