William Paavolainen

Hackstructor
Studying platform for cybersecurity

Bachelor’s thesis
Information Technology / Game Programming

2017

South-Eastern Finland
University of Applied Sciences

Tekijä/Tekijät	**Tutkinto**	**Aika**
William Paavolainen	Insinööri (AMK)	Joulukuu 2017
Opinnäytetyön nimi Hackstructor Kyberturvallisuuden oppimisalusta		43 sivua
Toimeksiantaja Kaakkois-Suomen ammattikorkeakoulu Oy		
Ohjaaja Lehtori Marko Oras		

Tiivistelmä

Tämän opinnäytetyön tavoitteena oli ensisijaisesti luoda virtuaalinen opiskelijoille suunnattu kyberturvallisuuspainotteinen itseopiskelualusta. Itse alustan tuli olla selkeä, helposti navigoitava, sekä huomioon oli otettava myös jatkokehityksen mahdollistaminen. Toteutettua alustaa tullaan käyttämään Kaakkois-Suomen ammattikorkeakoulussa osana tieto- ja viestintätekniikan koulutuksen kyberturvallisuuden kurssitoteutuksia.

Työssä käsitellään koko Hackstructor-alustaa, johon sisältyy useita virtuaalikoneita, mutta pääpainona on kuitenkin alustassa oleva verkkopalvelin ja sen sisältö. Alustan verkkopalvelin sisältää useita erilaisia haavoittuvuuksia ja niiden hyväksikäyttöön tarkoitettuja työkaluja, niihin liittyvää yleistietoa sekä harjoitteluosioita, joissa alustan käyttäjät voivat harjoitella haavoittuvuuksien hyväksikäyttöä ja työkalujen hyödyntämistä lähes riskivapaassa virtualisoidussa ympäristössä.

Työn aloitusvaiheessa annetut tavoitteet on saavutettu onnistuneesti. Hackstructor-alusta on tuotantokäytössä Kaakkois-Suomen ammattikorkeakoulun virtuaalilaboratoriojärjestelmässä. Aikaisempia vastaavan kaltaisia alustoja ei Kaakkois-Suomen ammattikorkeakoulussa ole ollut käytössä, eikä jo olemassa olevia alustoja tai niiden resursseja ole hyödynnetty tehokkaasti. Vertailukelpoisiin alustoihin nähden Hackstructorin valttina on työkalujen oppimismateriaali esimerkkeineen ja harjoittelumahdollisuuksineen.

Työssä jäi paljon kehitettävää etenkin sisällön puolesta, mutta kokonaisuudessaan työ onnistui hyvin. Työssä esiintyvä teoriapohjainen tieto sekä käytännön harjoittelun helppous tarjoavat hyvän mahdollisuuden jatkokehityksen toteuttamiseen sekä vastaavan kaltaisten web-sovelluksien kehittämiseen.

Asiasanat

kyberturvallisuus, tietoturvallisuus, hakkerointi, web-sovellus

XAMK
South-Eastern Finland
University of Applied Sciences

Author (authors) William Paavolainen	**Degree** Bachelor of Engineering	**Time** December 2017
Thesis Title Hackstructor Studying Platform for Cybersecurity		43 pages
Commissioned by South-Eastern Finland University of Applied Sciences Ltd.		
Supervisor Marko Oras, Senior Lecturer		

Abstract

The primary objective of this thesis was to create a virtual cybersecurity-oriented self-study platform for students. The platform had to be clear, easy to navigate, and future development had to also be taken into consideration. The implemented platform will be used by South-Eastern Finland University of Applied Sciences as a part of the ICT degree programme's cybersecurity courses.

The work covers the Hackstructor platform, which includes some virtual machines, but the primary focus is on the web server and its contents. The platform's web server includes a variety of vulnerabilities and exploitation tools, general information about them, and practicing sections where users can practice exploitation inside a near risk-free environment.

The goals for the project have been successfully achieved. The Hackstructor platform is currently in production inside the virtual laboratory system in South-Eastern University of Applied Sciences. In South-Eastern University of Applied Sciences there have not been any similar platforms before, and comparable platforms or their resources have not been utilized efficiently. When compared to similar solutions, Hackstructor's advantage is that there is information about exploitation tools with clear examples and possibilities to practice the usage of them.

There is lots of room for development especially regarding to the content, but overall the project is a success. Theoretical knowledge and the ease of practical training inside the Hackstructor platform offer a good opportunity for further development and development of similar web applications.

Keywords

cybersecurity, information security, hacking, web application

CONTENTS

1 INTRODUCTION

In the world of cybersecurity, the frequency of security breaches and other cyberattacks is currently growing. Few of the most notorious ones have also gained global media attention. Good example of the impact these cyberattacks have against businesses and individuals would be the WannaCry ransomware attack. WannaCry infected over 200000 systems within just one day, affecting more than 150 different countries (Brenner 2017).

As the frequency in cyberattacks is growing, so is the projected percent change in employment in the cybersecurity field. There is a growing demand of cybersecurity professionals such as information security analysts, as many businesses have seen the massive impact of modern cyberattacks.

The project started in the summer of 2017 in an internship period in South-Eastern Finland University of Applied Sciences Ltd. The original idea was to create a platform for extra cybersecurity focused material for students interested in the field to read through, and later the idea reformed to include a possibility to practice exploitation of different kinds of vulnerabilities inside the platform.

1.1 XAMK CyberLab

CyberLab is a data center that has been built a couple of years ago in South-Eastern Finland University of Applied Sciences for education and project purposes. Currently CyberLab serves as a data center for running a virtual laboratory system as a cloud service. CyberLab data center consists of twelve servers, which in all have 3 terabytes of random access memory and over three hundred cores (Kettunen 2017).

1.2 Purpose of this thesis

This thesis is about a web application platform called Hackstructor, that has multiple different built-in vulnerabilities for educational purposes. It is to be used by students at South-eastern Finland University of Applied Sciences as a studying platform that is in a supporting role for cybersecurity related courses.

Hackstructor provides a "shooting ground" for offensive web application security testing in a legal and low-/no-risk environment, and provides an insight into web applications from the security standpoint as well as in general. There are also setup- and usage guides, cheat sheets, preconstructed payloads and plenty of extra resources included in order to further educate people in the field of cybersecurity.

Although it is not required, Hackstructor is originally meant to be used in collaboration with Kali Linux, which is maintained and funded by Offensive Security Limited. Kali Linux is a Linux distribution that is primarily developed for penetration testing and security auditing purposes. Kali also has almost all the tools already installed that you can use for the practice-sections in the Hackstructor platform. (Aharoni et al. 2017, 2-3.)

2 BACKGROUND SPECIFICATIONS

In the most basic sense, Hackstructor is a vulnerable network which includes multiple systems and devices. It is completely virtual and it is located inside a virtual laboratory system that is running inside XAMK CyberLab data center.

This section explains shortly what the virtual laboratory system is, describes the goals for the Hackstructor project and already available solutions that are similar to Hackstructor.

2.1 Virtual laboratory system

Virtual laboratory system is a virtualized environment created by Jaakko Nurmi which allows the addition of multiple virtual machines and virtualized networking equipment into a single ”scenario”. This environment gives the possibility to set up a virtualized network that includes multiple virtual machines.

Virtual laboratory system is currently used at least in advanced routing, network security equipment and penetration testing courses taught in South-Eastern Finland University of Applied Sciences (Nurmi 2016).

The virtual laboratory system can be accessed from the South-Eastern University of Applied Sciences’ internal network and it requires users to log in with the login information provided by the university, as shown in Figure 1.

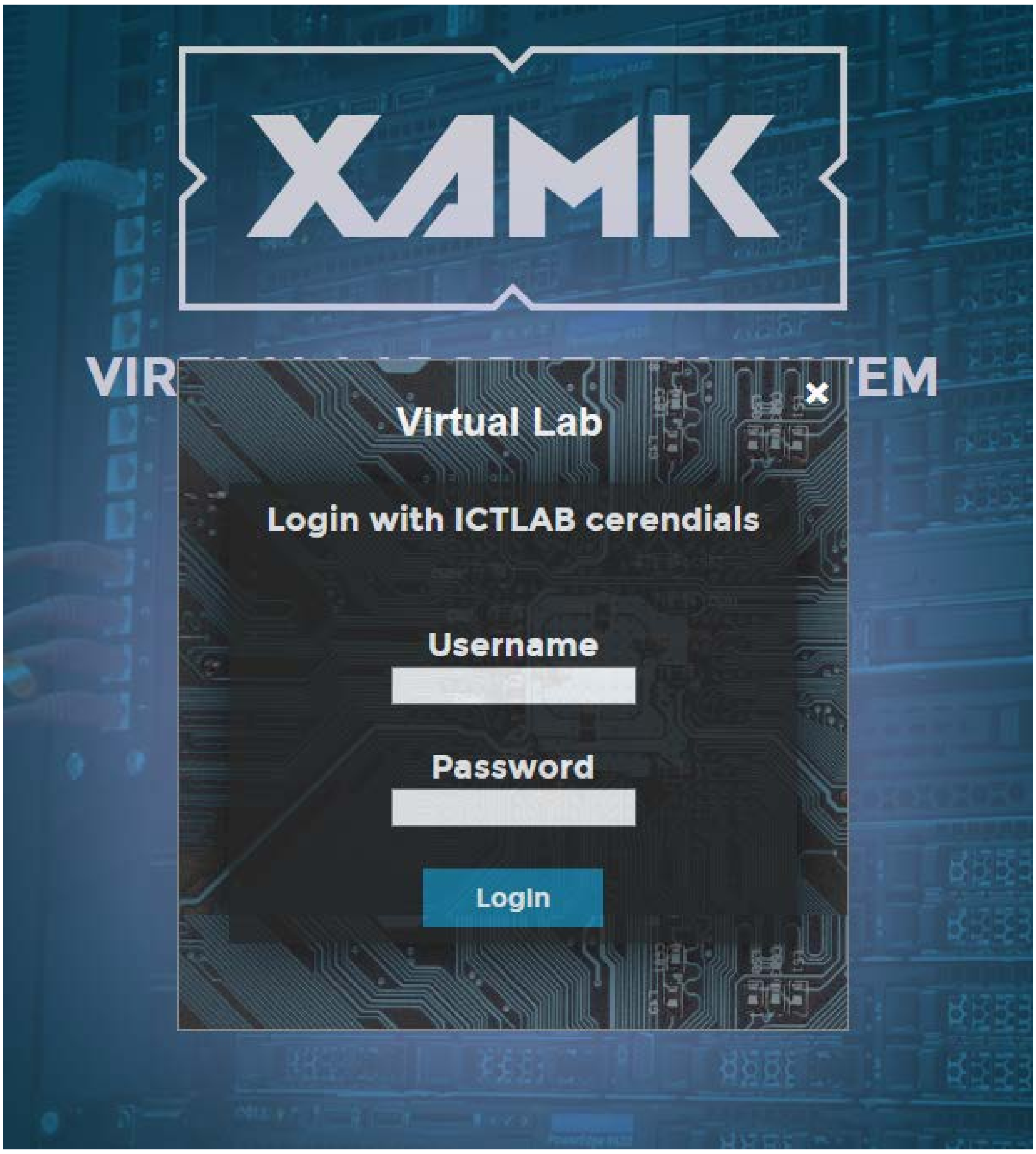


Figure 1. Virtual laboratory system's login page

All virtual machines and devices configured and used in the Hackstructor project are running inside the virtual laboratory system.

2.2 Hackstructor specifications

The primary goal for the Hackstructor project is to provide easily understandable and collective cybersecurity related studying material for users. This allows users to learn about cybersecurity and penetration testing inside a safe environment in their own phase.

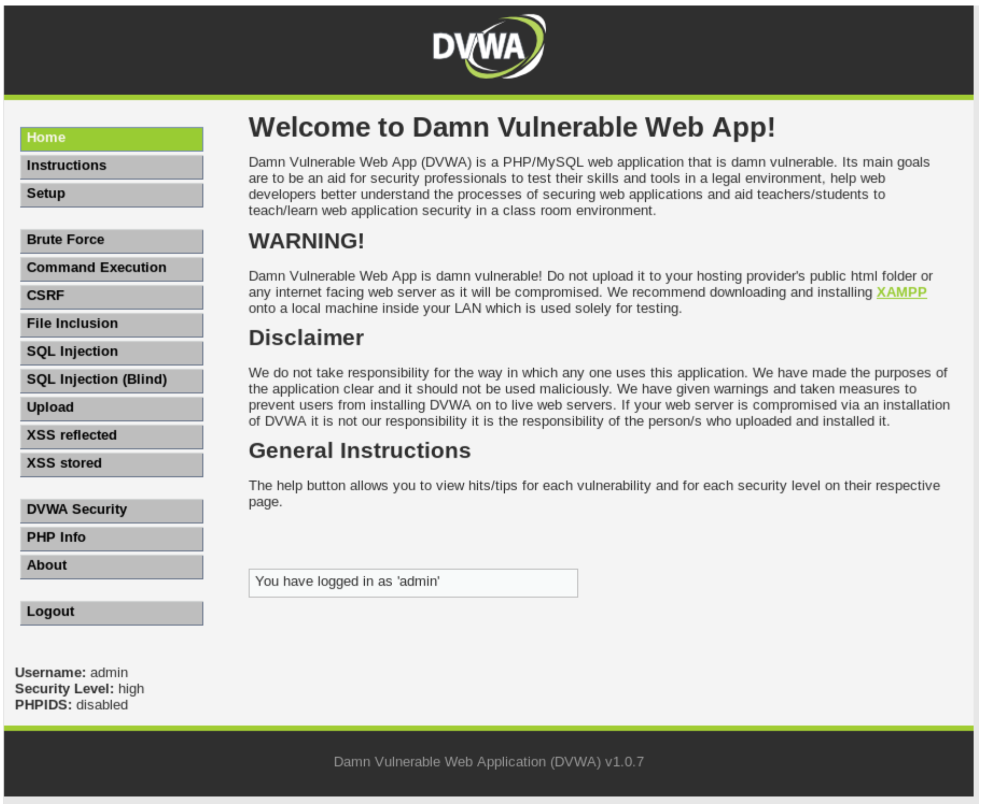
From the technical standpoint, a way to achieve the goal of this project is to provide easily understandable and accurate information about different kinds of vulnerabilities as well as exploitation tools, give working examples and extra resource material about them that provide a possibility for users to learn even more about specific subjects included in the platform.

2.3 Related solutions

There are a couple of notable solutions which all share similarities with Hackstructor. All the following solutions have had influence over Hackstructor, and providing general information about them will give some understanding about the starting point.

2.3.1 Damn Vulnerable Web Application

Damn Vulnerable Web Application (often referred as DVWA) shares probably the most similarities with Hackstructor at least as far as the web-based user interface is considered. The purpose of DVWA is also relatable to Hackstructor, as both are meant to help ethical hackers and other cybersecurity professionals on practicing the usage of various tools and techniques.
The home page and the navigation menu are simple and easy to operate, as can be seen in Figure 2 (DVWA 2017).



Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'admin'

Username: admin
Security Level: high
PHPIDS: disabled

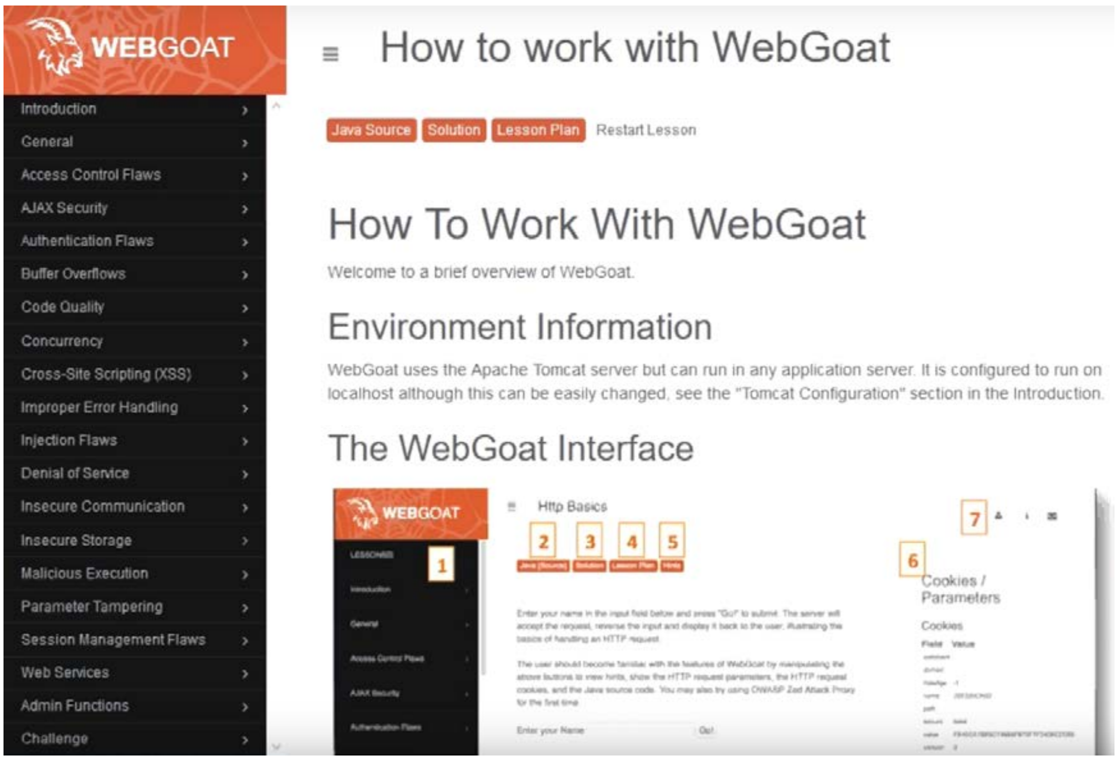
Damn Vulnerable Web Application (DVWA) v1.0.7

Figure 2. Damn Vulnerable Web App home page

For the Hackstructor project, DVWA was the most influential solution.

2.3.2 WebGoat

WebGoat is also a purposefully vulnerable web application, but the major difference compared to other similar web applications is that WebGoat is basically a collection of “lessons”. These lessons require users to exploit vulnerabilities while providing little tips for the user. The user can also see the solution for the relevant lesson if he/she so desires. WebGoat and DVWA have similar navigation panels as can be seen in Figure 3 (OWASP 2016a).



How to work with WebGoat

How To Work With WebGoat

Welcome to a brief overview of WebGoat.

Environment Information

WebGoat uses the Apache Tomcat server but can run in any application server. It is configured to run on localhost although this can be easily changed, see the "Tomcat Configuration" section in the Introduction.

The WebGoat Interface

Enter your name in the input field below and press "Go!" to submit. The server will accept the request, reverse the input and display it back to the user, illustrating the basics of handling an HTTP request.

The user should become familiar with the features of WebGoat by manipulating the above buttons to view hints, show the HTTP request parameters, the HTTP request cookies, and the Java source code. You may also try using OWASP Zed Attack Proxy for the first time.

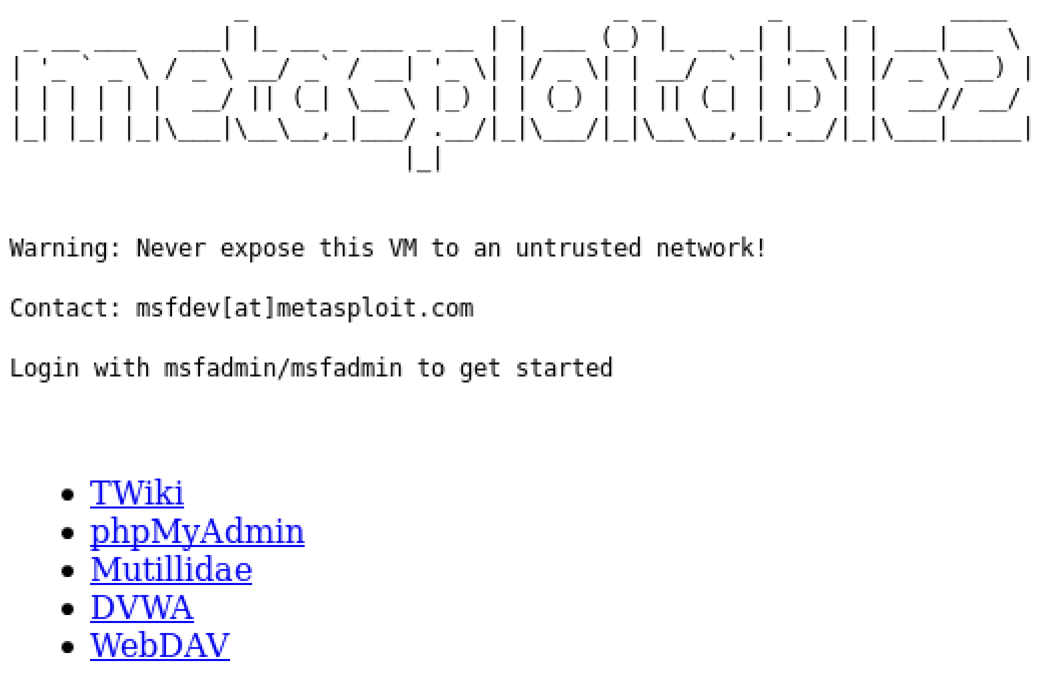
Figure 3. WebGoat home page

WebGoat's lesson system fits extremely well to a school environment, which was a good enough reason to include it in Hackstructor as well.

2.3.3 Metasploitable

Metasploitable is not just a vulnerable web application, but a whole vulnerable virtual machine built on GNU/LINUX. Metasploitable is a great target to practice different techniques and the usage of various tools against. There are no real lessons or in-depth explanations provided for the exploitable vulnerabilities, but the virtual machine serves as an excellent "punching bag" (Rapid7 s.a.).

There are a couple of vulnerable web applications included in Metasploitable, as can be seen in Figure 4.



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

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

Figure 4. Metasploitable2 website home page

Metasploitable gave the idea of a “Tools” section that later came to be included in the Hackstructor project.

3 PLATFORM DEVELOPMENT

This section covers the virtual machines that the Hackstructor platform requires, network topology, and explanations about the contents of Hackstructor’s web server.

3.1 Systems and devices

The most basic way to approach the Hackstructor platform’s requirements regarding the systems and devices is to think about the requirements from a user’s perspective.

Users need to be able to access the resources provided in the platform, which means that first we need a system where these resources are located and a way to access that system. In this case the system that has the resources included is an Ubuntu server 16.04 LTS, and the way to access it is through a Kali Linux desktop machine.

As some of the above-mentioned resources refer to a specific system, that system needs to be included as well. In this case that system is Microsoft Windows XP.

Ubuntu server 16.04 LTS

This server is the heart of the project. It will serve as a web server where the exploitation and vulnerability descriptions, examples, practice instructions and all other website contents are hosted. This is the system what the users most often interact with (excluding Kali Linux, since that is the system what users are using to access this one).

The reason why Ubuntu 16.04 LTS was chosen is that it is fast to setup, I have a fair understanding of the system and it allowed me to save time in the setup phase since I already had a fully ready image of a working Ubuntu server 16.04 LTS virtual machine. Hard requirements for this server are MySQL, PHP and SSH capability. The contents of this server will be discussed in more depth in a later chapter.

Kali Linux

Kali Linux is a popular Linux distribution that is primarily developed for penetration testing and security auditing purposes. Kali Linux fits perfectly with our platform inside the virtual environment and at the current state of development Kali has all the required tools for the Hackstructor web applications' practice sections. (Aharoni et al. 2017, 2-3.)

Windows XP

Windows XP is a good operating system for beginners to practice against. The support for Windows XP has ended in 8.4.2014, which means that Microsoft does not provide any security updates anymore for this operating system. Therefore, when new vulnerabilities are found, there will be no patches against them for Windows XP (Microsoft 2017).

Windows XP is a near perfect target for practicing the usage of Metasploit framework against, and there are plenty of guides and tutorials ready on the web that help users to exploit this operating system.

The setup requirements for Windows XP, excluding the virtual machine itself, are extremely minimal. In the current state of the Hackstructors' development, all this machine needs to have is an administrator user and a couple of "Easter eggs" for users to find.

3.2 Network

Currently there is no need for an outside network access. The outside network access would not provide almost any benefits for the Hackstructor platform, but also expose highly vulnerable systems to the public network and present the possibility of a user to accidentally use the hacking tools included in Kali Linux against an unauthorized network.

The basic networking requirements are that a user needs to be able to access the web server and Windows XP desktop machine through Kali Linux. This can be achieved by inserting each system into the same local area network inside the virtual laboratory system. As can be seen in Figure 5, the chosen network topology layout is a star network.

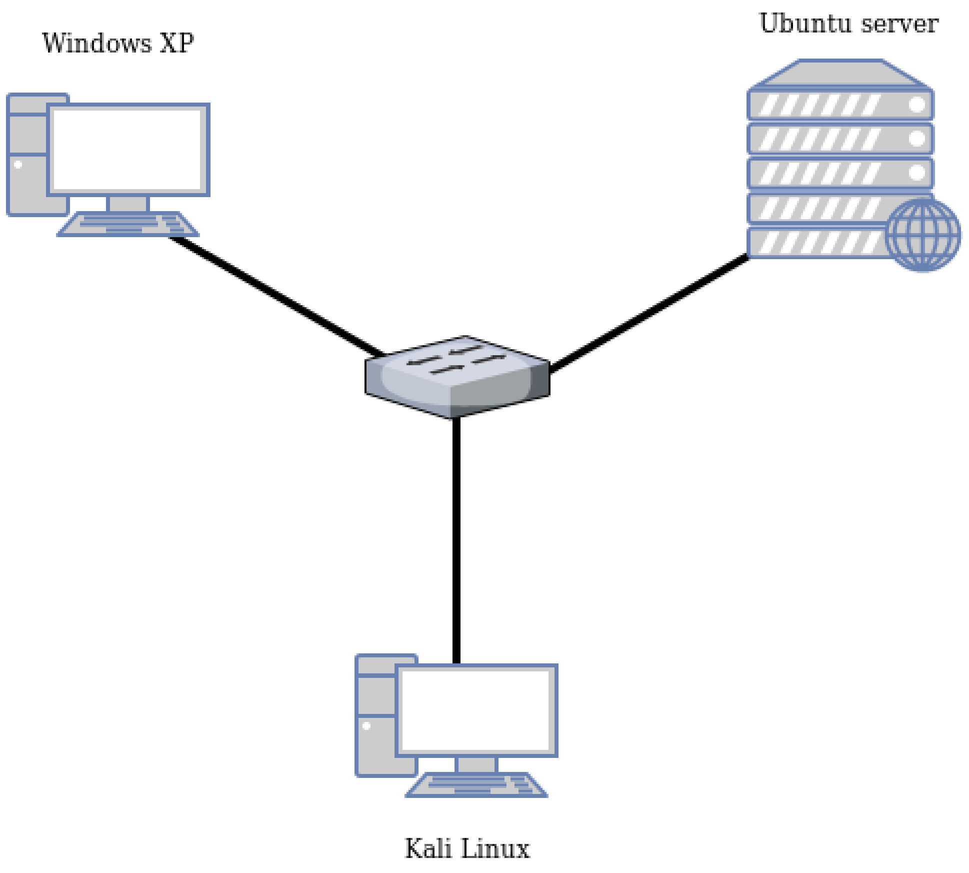


Figure 5. Hackstructor network topology

3.3 Server contents

This section covers the contents of the web server, which has the highest priority from development standpoint.

The whole content of the server is divided to four categories:

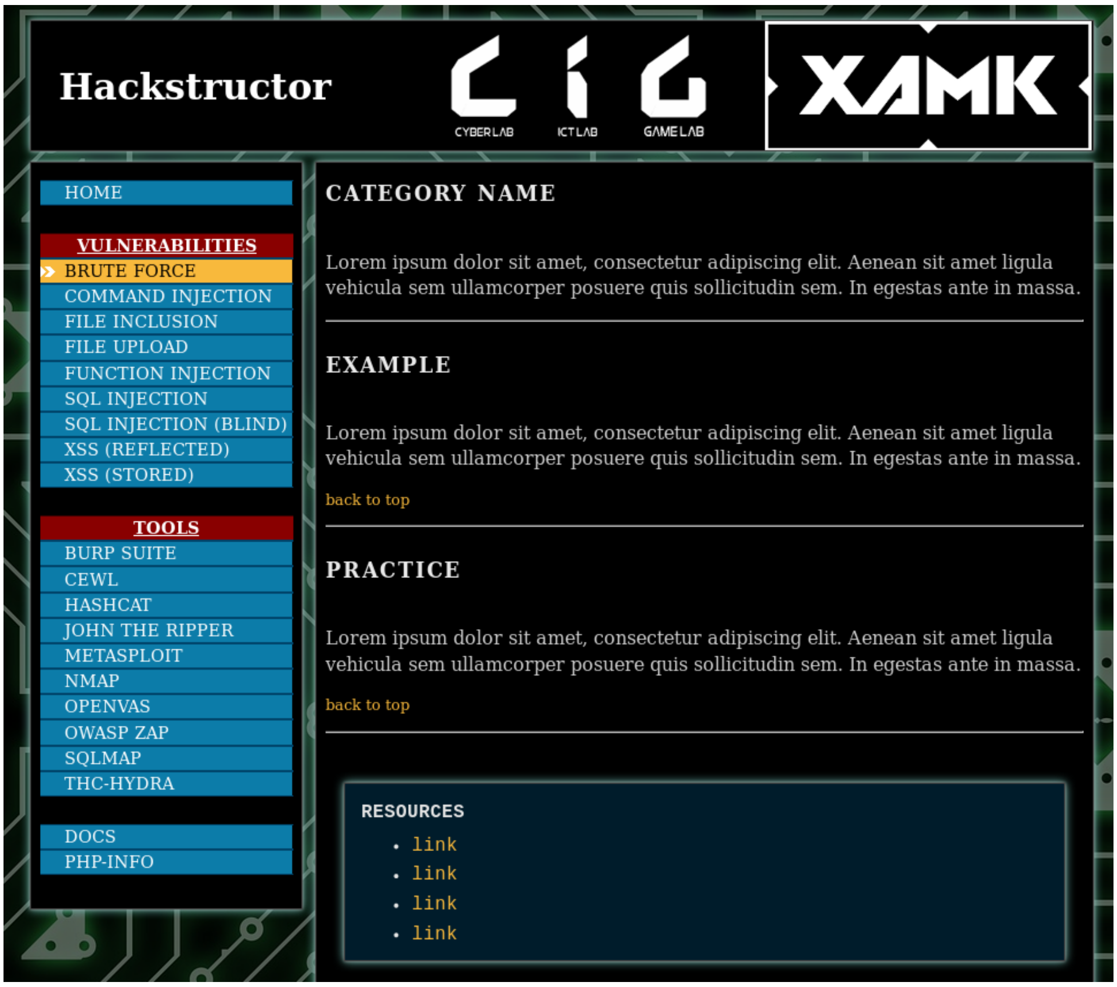
- Vulnerabilities and exploitation examples
- Exploitation tools
- Databases
- Other back end content

3.3.1 Vulnerabilities and exploitation examples

The Hackstructor platform contains multiple purposefully built-in vulnerabilities. At the current state of development, all vulnerabilities introduced in the web server are web application focused vulnerabilities.

Each vulnerability in the Hackstructor platform's web server has a short explanation about what the vulnerability is, how it works in theory, exploitation examples, section for practicing exploitation against the vulnerability, as well as extra resources and references for users to learn more about the vulnerability.

Figure 6 shows a template page for the Hackstructor.



CATEGORY NAME

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean sit amet ligula vehicula sem ullamcorper posuere quis sollicitudin sem. In egestas ante in massa.

EXAMPLE

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean sit amet ligula vehicula sem ullamcorper posuere quis sollicitudin sem. In egestas ante in massa.

back to top

PRACTICE

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean sit amet ligula vehicula sem ullamcorper posuere quis sollicitudin sem. In egestas ante in massa.

back to top

RESOURCES

- link
- link
- link
- link

Figure 6. Template page for Hackstructor

Brute Force

In short, brute force attack is the practice of guessing a value for a known parameter, for example a password, passphrase or even a directory. Brute force as a method tries every possible character combination until a right one is found, or when some other user defined requirement is met. This type of attack is systematic, and in theory, can decrypt any kind of encryption (OWASP 2016b).

Although brute force can theoretically crack anything and everything, it realistically cannot. The biggest weakness it has is long character strings. As an example, if someone would like to crack a hashed password that is 1 character long, on modern computers it would take only a fraction of a second to do so. But if you would like to do the same against a hashed password that has the length of 30 characters, the time required to calculate all possible character combinations would exceed a lifetime.

Command injection

This section about command injection is based on OWASP website's command injection web page (OWASP 2016c).

Command injection, also known as shell command injection, is a form of attack where an attacker exploits vulnerable and unprotected website applications to input their own commands within the service.

Command injection exists because of poor validation of input within the data supplied by the user to an application through forms, cookies and HTTP-headers for example. In short, applications and services can be exploited by executing system-wide commands through the web application.

This form of attack can be used to display information that is not supposed to be available for normal users, functioning similarly to SQL Injection. It can also be used to execute malicious programs or scripts to obtain passwords and other sensitive information as the end-user accesses a website.

File Inclusion

File inclusion vulnerability provides an opportunity for an attacker to include a file by exploiting the badly written code of a web application. The main cause for the existence of this type of vulnerability is improperly validated and/or unsanitized user input. File inclusion includes a couple of different vulnerability subcategories:

- Local File Inclusion (LFI)
- Remote File Inclusion (RFI)

Local file inclusion and remote file inclusion are both quite self-explanatory. Local file inclusion includes files located at the server, while remote file inclusion includes files which are located remotely (Usually included in the form of HTTP- or FTP URI) (Offensive security 2017).

Function injection

This section about function injection is based on OWASP website’s function injection web page (OWASP 2016d).

In exploitation via function injection, the attacker takes advantage of poor filtering of user input and untrusted data. This way the attacker can inject a malicious code, and the website application will falsely interpret it as something it is supposed to execute.

A more complex code will not only allow the attacker to pass any function to the application in question, but also define certain parameters to it. In theory, this would allow the attacker to remotely access the services, and pass their own commands as system-wide command functions.

Generally, web applications vulnerable to function injection are moderately hard to exploit, but if the exploitation is successful, the results can be devastating.

SQL injection

OWASP (2016e) states: *“A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system.”*

Just to clarify, SQL injection basically means the modification of a predefined SQL query via some sort of field that accepts user input. Unsanitized user input is the reason this vulnerability exists.

Blind-SQL injection

This section about blind SQL injection is based on OWASP website’s blind SQL injection web page (OWASP 2016f).

Blind SQL injection and normal SQL injection are almost identical when compared to each other, the only difference being the data retrieval method. When utilizing blind SQL injection attack method, you try to ask multiple true/false type questions from the database and inspect the response of the application.

Blind SQL injection should not be the first thing you try when you find a possibly vulnerable field accepting user input. It is quite time-consuming when compared to the alternatives, but sometimes you have no choice in the matter.

Blind SQL injection is often used when the application outputs just general error messages which do not provide you with enough information, or when the application does not provide error messages at all. Blind SQL injection is also more difficult than normal SQL injection, because you must do a lot more guesswork, but with some logical thinking and systematic approach it is still very much possible.

Reflected/client-side cross-site scripting

This section about reflected/client-side cross-site scripting is based on OWASP website's cross-site scripting web page (OWASP 2016g).

Cross-site scripting (XSS) attacks are one of the injection type attacks where an attacker can inject scripts (E.g. JavaScript) into websites, which are then executed inside another users' browser through the web application.

XSS has historically been, and still is, one of the most common vulnerabilities in web applications. OWASP Top Ten Project has rated XSS as the seventh most common vulnerability in 2017 (OWASP 2017a).

XSS attack is possible in a couple of different occasions:

- Web application gets data through an untrusted source
- User input data is not sanitized and/or content data sent to the end user is not validated properly

Malicious data that is sent by the attacker can include any type of code that the browser can execute (JavaScript, HTML, Flash, etc.). When a different user opens a webpage containing that malicious payload, the payload gets executed.

Anything you can code with JavaScript, you can execute in target user's browser session (website redirect, session hijacking (related to CSRF vulnerability), keylogging, opening shells, etc.).

Reflected XSS attacks occur when the attackers' injected script is reflected off the web server, for example an error message. In this case the attacker can deliver the malicious script via social engineering through emails or other websites. For example, a target user clicks a malicious link, the injected code gets sent to the website and is reflected to the target user's browser.

Stored/server-side cross-site scripting

This section about stored/server-side cross-site scripting is based on OWASP website's cross-site scripting web page (OWASP 2016g).

Stored cross-site scripting is fundamentally in the same category of attacks as reflected XSS; injection. The payloads are quite similar as well as the means of exploitation.

Stored XSS attacks occur when the attackers' injected script is stored on the server via a comment field for example. This means that each time any user goes to see the comments where the attacker has injected a script, that script will be executed.

From attackers' point of view, a stored XSS is more devastating than a reflected XSS. The reason being that the attacker can affect multiple target users at the same time without the need of direct interaction with the users.

Unrestricted file upload

This section about unrestricted file upload is based on OWASP website's unrestricted file upload web page (OWASP 2017b).

Many attackers are aiming towards their goal via two steps:

- Get a harmful code into target system
- Execute the harmful code

If attackers can upload files without proper sanitization, they can instantly skip through the first step and start trying to get the harmful code executed.

Think about the following scenario:
A hacker finds a possibly vulnerable upload-form from a local company's website. The upload-form is meant to give registered users the possibility to upload their own profile picture. The hacker now proceeds to upload a PHP-

shell, a php-file which gives file system- and shell-access to the attacker. Because the company’s upload-form is completely unsanitized, the PHP-shell gets uploaded into their system and the attacker now has the server under his/her command.

This is a very possible, but somewhat unlikely scenario. The almost instant exploitation could have been avoided with proper sanitization of files uploaded by users.

3.3.2 Exploitation tools

In the Hackstructors’ web server content, exploitation tools have a similar content structure as the vulnerabilities have, but with a few differences. First there is a short explanation about what the tool in question is and what it does, then there is a section for how to set the tool up for use, examples on how to use the tool, section for practicing the usage of it, and finally resources and references for additional information.

Burp suite

Burp Suite is a tool for web application penetration testing made with Java (Portswigger Ltd. 2017a). Burp Suite provides multiple functionalities, such as a proxy server, a web spider, a scanner, and more, as shown in Figure 7.

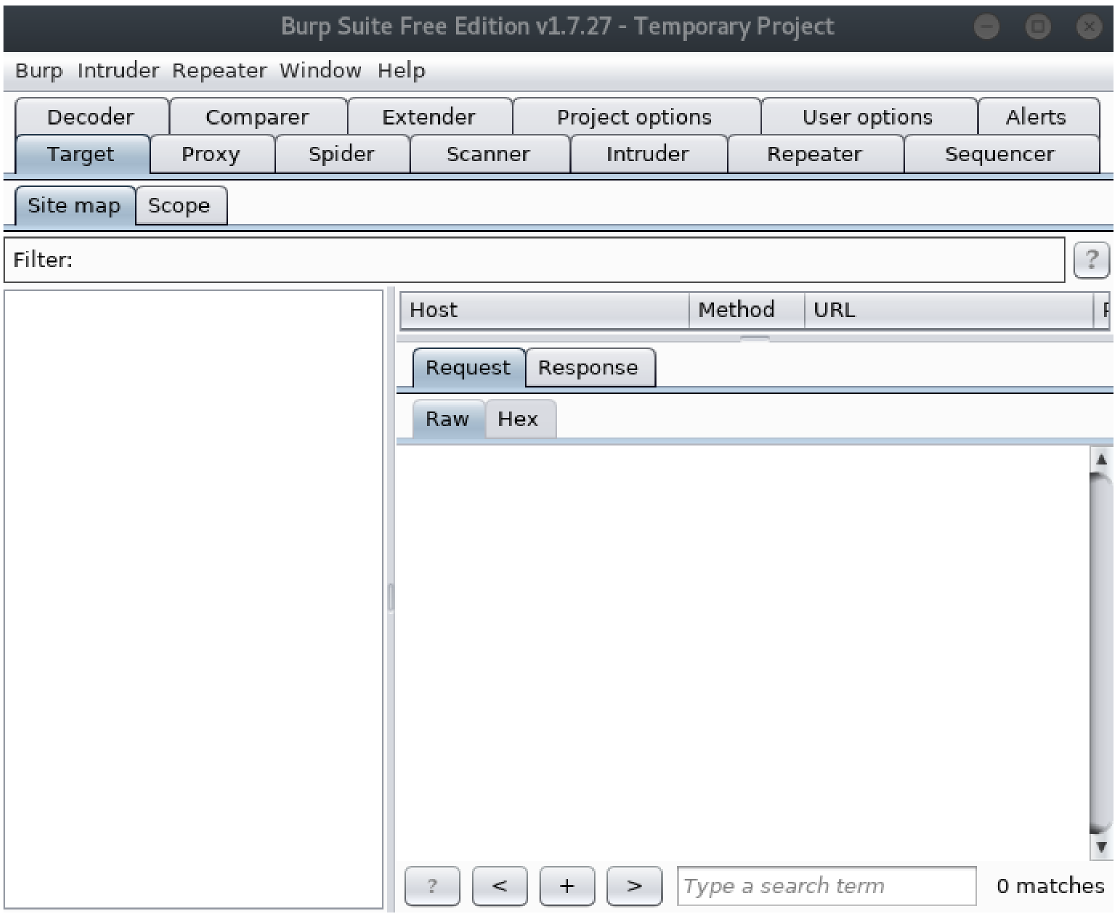

Burp Suite Free Edition v1.7.27 - Temporary Project
Burp Intruder Repeater Window Help
Decoder Comparer Extender Project options User options Alerts
Target Proxy Spider Scanner Intruder Repeater Sequencer
Site map Scope
Filter:
Host Method URL
Request Response
Raw Hex
Type a search term
0 matches

Figure 7. Empty temporary project in Burp Suite with different features shown

Burp Suite has currently two different editions available:

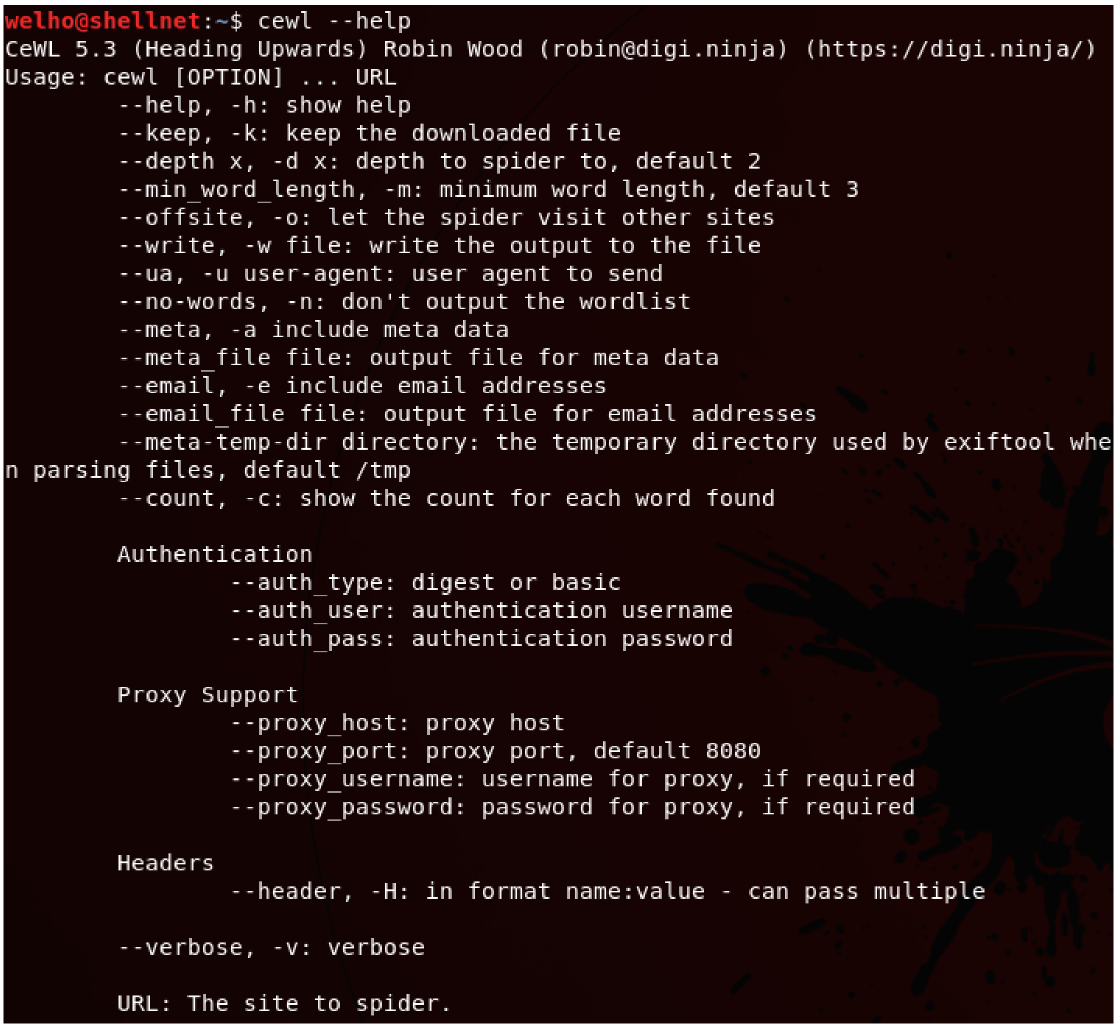
- Free Edition
- Professional Edition

The professional edition has more features available than the free edition, but for educational purposes and because Hackstructor is aimed at beginners in the cybersecurity field, the free edition has been chosen as the focused version (Portswigger Ltd. 2017b).

CeWL

CeWL is a custom word list generator that is used for website spidering and searching for unique words to create wordlists with ease. The user will be able to specify which URL CeWL will search, how long words it should search for, and whether it will include e-mail addresses in its search results. It is also capable of following links to external sites (Wood 2017).

CeWL is a tool that is extremely simple and easy to use. As can be seen in Figure 8, running the command “cewl --help” provides information about the flags you can assign to the command.

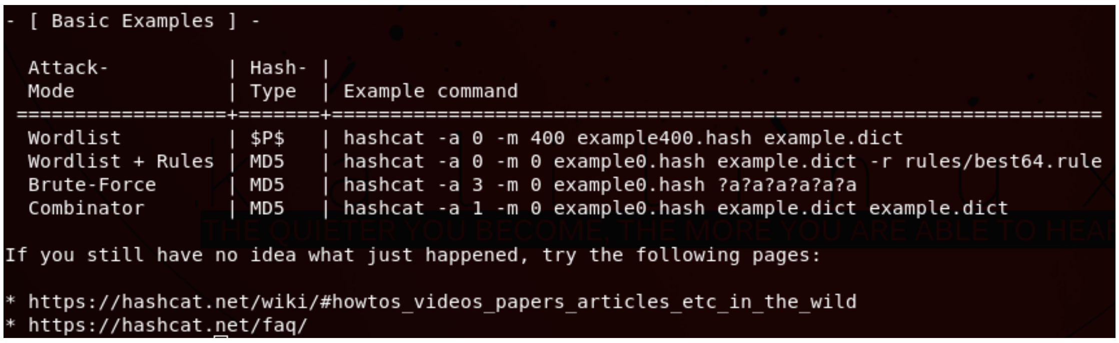


```
welho@shellnet:~$ cewl --help
CeWL 5.3 (Heading Upwards) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
Usage: cewl [OPTION] ... URL
        --help, -h: show help
        --keep, -k: keep the downloaded file
        --depth x, -d x: depth to spider to, default 2
        --min_word_length, -m: minimum word length, default 3
        --offsite, -o: let the spider visit other sites
        --write, -w file: write the output to the file
        --ua, -u user-agent: user agent to send
        --no-words, -n: don't output the wordlist
        --meta, -a include meta data
        --meta_file file: output file for meta data
        --email, -e include email addresses
        --email_file file: output file for email addresses
        --meta-temp-dir directory: the temporary directory used by exiftool whe
n parsing files, default /tmp
        --count, -c: show the count for each word found

        Authentication
                --auth_type: digest or basic
                --auth_user: authentication username
                --auth_pass: authentication password

        Proxy Support
                --proxy_host: proxy host
                --proxy_port: proxy port, default 8080
                --proxy_username: username for proxy, if required
                --proxy_password: password for proxy, if required

        Headers
                --header, -H: in format name:value - can pass multiple

        --verbose, -v: verbose

        URL: The site to spider.
```

Figure 8. CeWL command help

Hashcat

Hashcat is a "world's fastest and most advanced" (Hashcat. s.a. a) password recovery tool often used in dictionary/brute-force attacks. It has extremely extensive list of supported hash-types as well as options for customizing the attack to precisely fit the needs of the user. However, there are so many options to choose from and so many different values for those options that it might be somewhat overwhelming for new users. That in mind, there are examples of working commands so that even new users can get started with Hashcat, as shown in Figure 9.

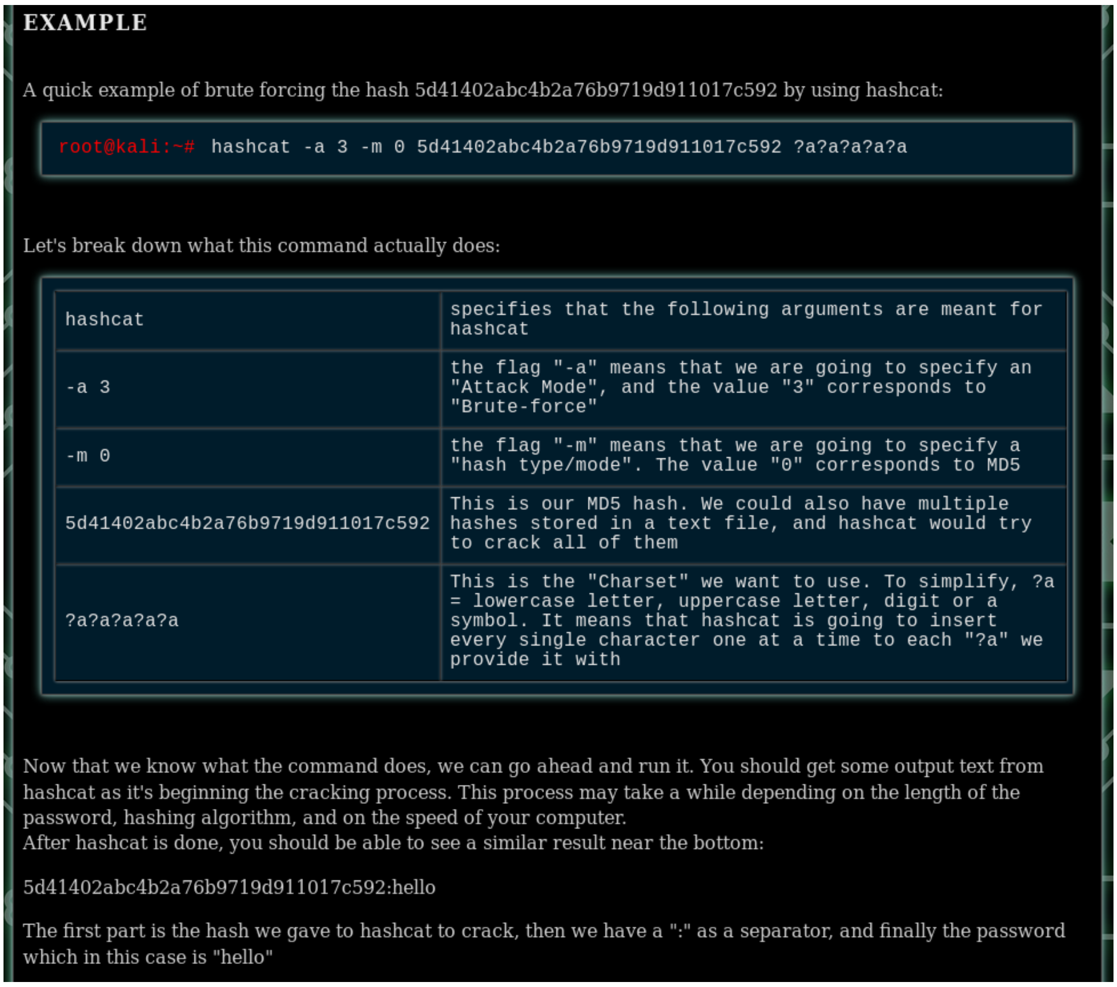


```
- [ Basic Examples ] -

  Attack-          | Hash- |
  Mode             | Type  | Example command
 ==================+=======+==================================================================
  Wordlist         | $P$   | hashcat -a 0 -m 400 example400.hash example.dict
  Wordlist + Rules | MD5   | hashcat -a 0 -m 0 example0.hash example.dict -r rules/best64.rule
  Brute-Force      | MD5   | hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a?a
  Combinator       | MD5   | hashcat -a 1 -m 0 example0.hash example.dict example.dict

If you still have no idea what just happened, try the following pages:

* https://hashcat.net/wiki/#howtos_videos_papers_articles_etc_in_the_wild
* https://hashcat.net/faq/
```

Figure 9. Hashcat command examples

There are also plenty of information regarding the usage of the tool on the Internet (Hashcat. s.a. b).

In the Hackstructor’s website under the menu entry for Hashcat can be found a usage example as shown in Figure 10.



EXAMPLE

A quick example of brute forcing the hash 5d41402abc4b2a76b9719d911017c592 by using hashcat:

```
root@kali:~# hashcat -a 3 -m 0 5d41402abc4b2a76b9719d911017c592 ?a?a?a?a?a
```

Let's break down what this command actually does:

hashcat	specifies that the following arguments are meant for hashcat
-a 3	the flag "-a" means that we are going to specify an "Attack Mode", and the value "3" corresponds to "Brute-force"
-m 0	the flag "-m" means that we are going to specify a "hash type/mode". The value "0" corresponds to MD5
5d41402abc4b2a76b9719d911017c592	This is our MD5 hash. We could also have multiple hashes stored in a text file, and hashcat would try to crack all of them
?a?a?a?a?a	This is the "Charset" we want to use. To simplify, ?a = lowercase letter, uppercase letter, digit or a symbol. It means that hashcat is going to insert every single character one at a time to each "?a" we provide it with

Now that we know what the command does, we can go ahead and run it. You should get some output text from hashcat as it's beginning the cracking process. This process may take a while depending on the length of the password, hashing algorithm, and on the speed of your computer.
After hashcat is done, you should be able to see a similar result near the bottom:

5d41402abc4b2a76b9719d911017c592:hello

The first part is the hash we gave to hashcat to crack, then we have a ":" as a separator, and finally the password which in this case is "hello"

Figure 10. Hashcat usage example in Hackstructor

John the Ripper

This section about John the Ripper is based on Openwall's official John the Ripper documentation (Openwall 2013).

John the Ripper is a tool used for password cracking, similar to Hashcat. It supports and automatically detects multiple different hash types, and it also has highly customizable cracking capabilities.

John the Ripper has a couple of different cracking modes included within itself:

- Wordlist mode
- Single crack mode
- Incremental mode
- External mode

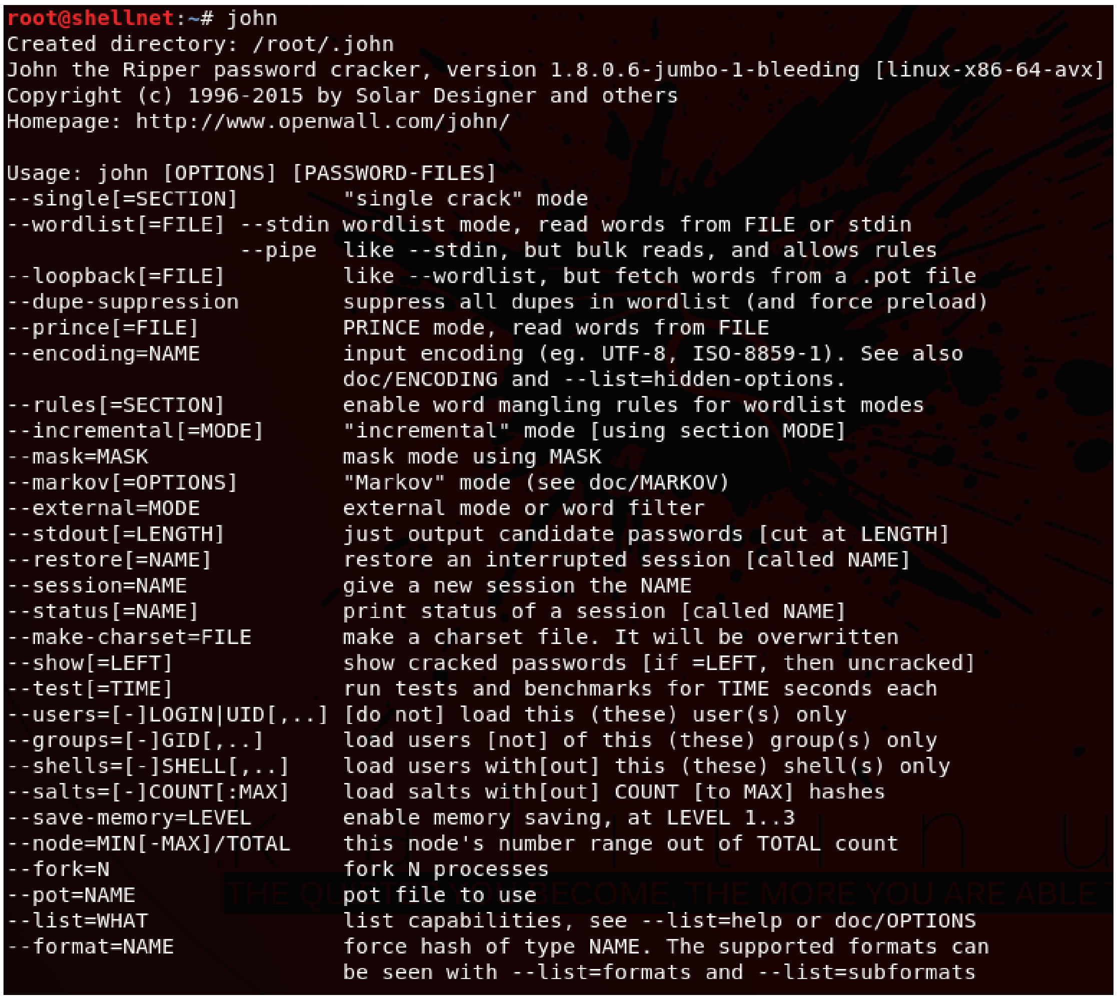
The wordlist mode is in my opinion probably the simplest cracking mode John the Ripper supports. It is basically a dictionary attack; you provide a wordlist along with a hash file, and John the Ripper tries all lines inside the wordlist against the hash file. Word mangling rules can also be used with the wordlist mode, and it is possible for users to even create their own rules that John will apply into the cracking process.

The single crack mode is kind of like the wordlist mode, but instead of using a separate wordlist, it constructs a list of words found in the so called GECOS fields which include a system user's full name, phone numbers and location of his/her office, provided that the user in question has entered this information when the system user was created (Cormany 2009). That means the "wordlist" in single crack mode will be substantially smaller, but John will apply a complete set of mangling rules it supports.

The incremental mode is basically the traditional brute forcing mode. With the incremental mode, John will try every possible combination of characters until the right one is found.

The external mode allows users to define/create an external cracking mode for John. Full utilization of this mode requires some programming experience, because external modes are created by using a subset of the C programming language.

As can be seen in Figure 11, running the program without any parameters triggers the help flag for it which displays various flags and their meanings in the terminal window.



```
root@shellnet:~# john
Created directory: /root/.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]
--make-charset=FILE       make a charset file. It will be overwritten
--show[=LEFT]             show cracked passwords [if =LEFT, then uncracked]
--test[=TIME]             run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..] [do not] load this (these) user(s) only
--groups=[-]GID[,..]      load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]    load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX]    load salts with[out] COUNT [to MAX] hashes
--save-memory=LEVEL       enable memory saving, at LEVEL 1..3
--node=MIN[-MAX]/TOTAL    this node's number range out of TOTAL count
--fork=N                  fork N processes
--pot=NAME                pot file to use
--list=WHAT               list capabilities, see --list=help or doc/OPTIONS
--format=NAME             force hash of type NAME. The supported formats can
                          be seen with --list=formats and --list=subformats
```

Figure 11. John the Ripper command help

Metasploit

Metasploit is a highly developed tool used in penetration testing. It is one of the most popular tools/frameworks in the penetration testing field, because of the exploitation possibilities it provides (Rapid7 metasploit s.a.).

Metasploit has various editions (Rapid7 s.a. b):

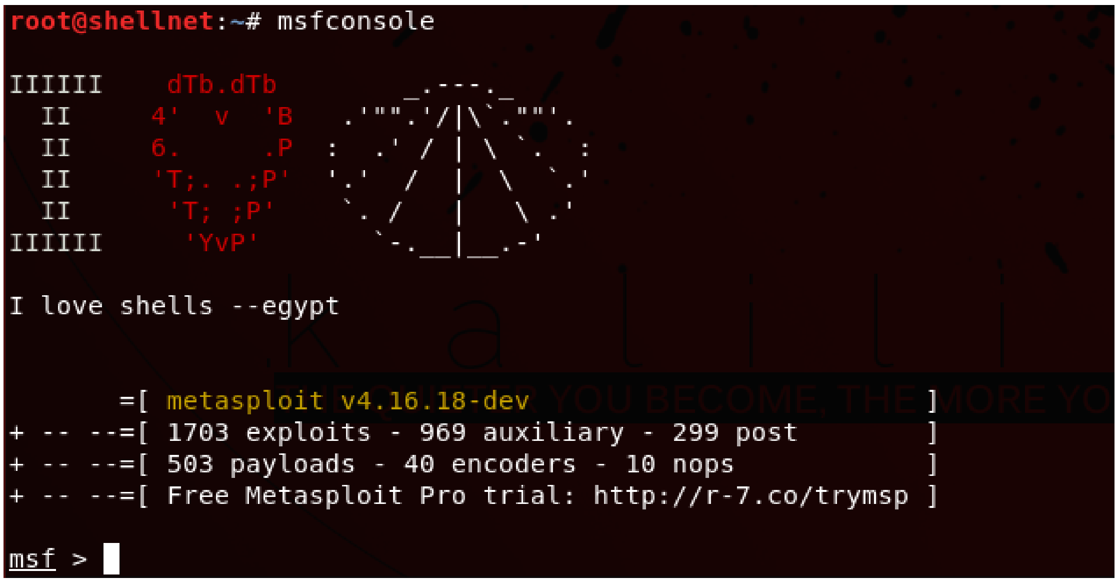
- **Metasploit Pro**: Commercial version that provides the most features. It also has automated exploitation capabilities.

- **Metasploit Express**: Another commercial version that is primarily meant for small and medium-sized businesses, as well as for IT generalists.
- **Metasploit Community**: Free version with less functionalities compared to the Metasploit Pro and Express. This version is primarily meant for students and small companies.
- **Metasploit Framework**: Free command-line version with all manual tasks (exploitation, third-party import, etc.)

Metasploit also offers multiple different user interfaces (Maldevel 2012):

- **Graphical User Interface (GUI)**: This is a user-friendly interface that allows you to set options by just clicking buttons.
- **Console Interface**: Console interface allows you access to all options Metasploit has to offer. It is the most popular interface, and we are going to work with this one.
- **Armitage**: Another GUI-style interface for Metasploit. The most notable features include recommendations for exploits and support for feature automation.

Running the command “msfconsole” opens up the console for Metasploit Framework as can be seen in Figure 12.



```
root@shellnet:~# msfconsole

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v4.16.18-dev                         ]
+ -- --=[ 1703 exploits - 969 auxiliary - 299 post        ]
+ -- --=[ 503 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

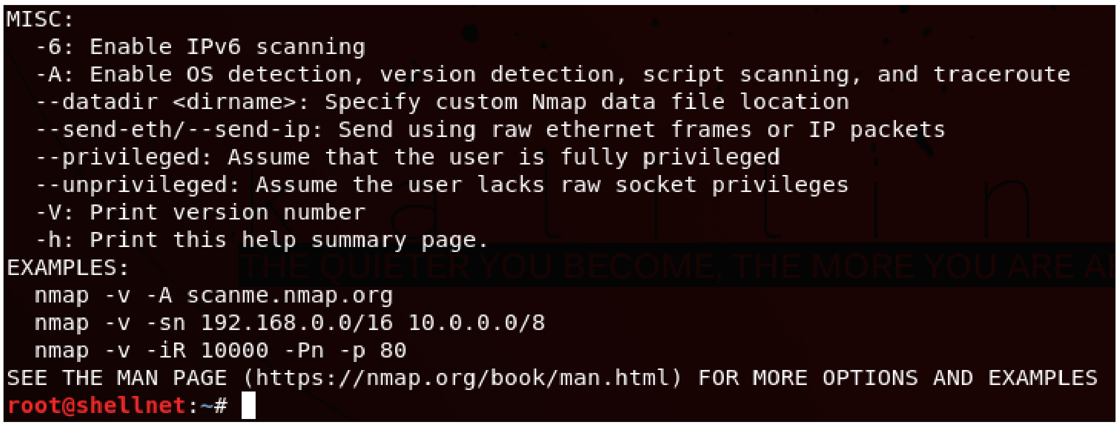
Figure 12. Metasploit Frameworks' msfconsole command

NMAP

Nmap (a.k.a. Network Mapper) is a top-tier tool used for both network discovery and security auditing purposes. Nmap can be used in many different phases of hacking, most notably in the reconnaissance- and scanning -phases. Nmap is most commonly used for security audits, but it is also used by many system and network administrators (Lyon, G. 2008).

With Nmap you can find out what hosts are available on the network, their services, operating systems, possible firewalls and/or packet filters on the network, and so on and so forth. The tool is also highly versatile in the sense that you can extend its capabilities with scripts (self-made and/or premade by others).

By running the program without any flags, it prints out the help-menu for Nmap, and in the bottom of that menu there are a few example commands as shown in Figure 13.



```
MISC:
  -6: Enable IPv6 scanning
  -A: Enable OS detection, version detection, script scanning, and traceroute
  --datadir <dirname>: Specify custom Nmap data file location
  --send-eth/--send-ip: Send using raw ethernet frames or IP packets
  --privileged: Assume that the user is fully privileged
  --unprivileged: Assume the user lacks raw socket privileges
  -V: Print version number
  -h: Print this help summary page.
EXAMPLES:
  nmap -v -A scanme.nmap.org
  nmap -v -sn 192.168.0.0/16 10.0.0.0/8
  nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
root@shellnet:~#
```

Figure 13. NMAP command examples

OpenVAS

OpenVAS is a free open source framework of tools that offer vulnerability scanning and solutions for vulnerability management. OpenVAS originally began as a fork of another scanning tool called Nessus which is currently a closed source vulnerability scanner (OpenVAS s.a.).

After OpenVAS has been installed and configured correctly, a web user interface can be opened for it directly through a browser. Through the web interface users can launch vulnerability scans against any target networks or systems they want. The home page of the web interface can be seen in Figure 14.

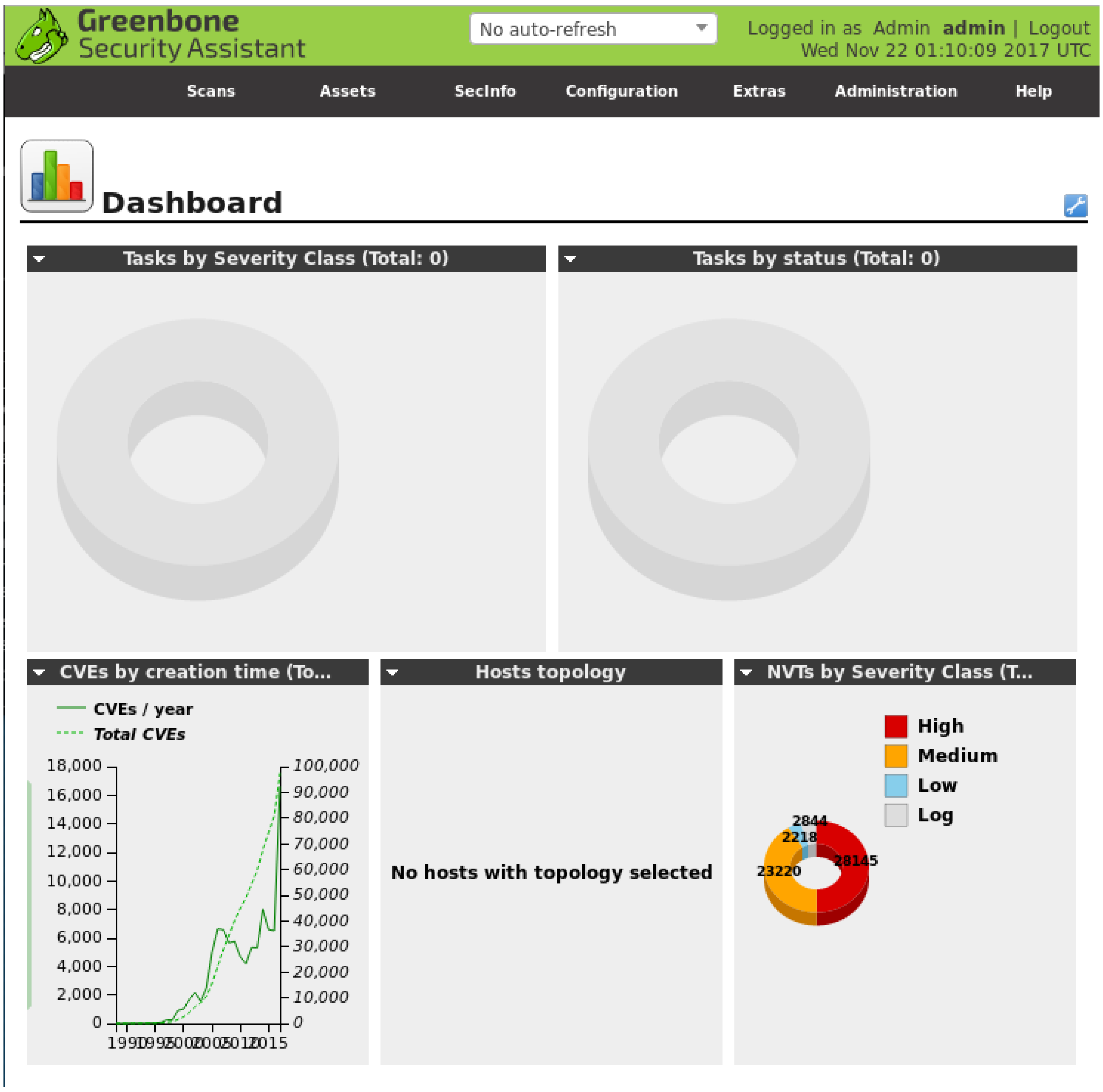
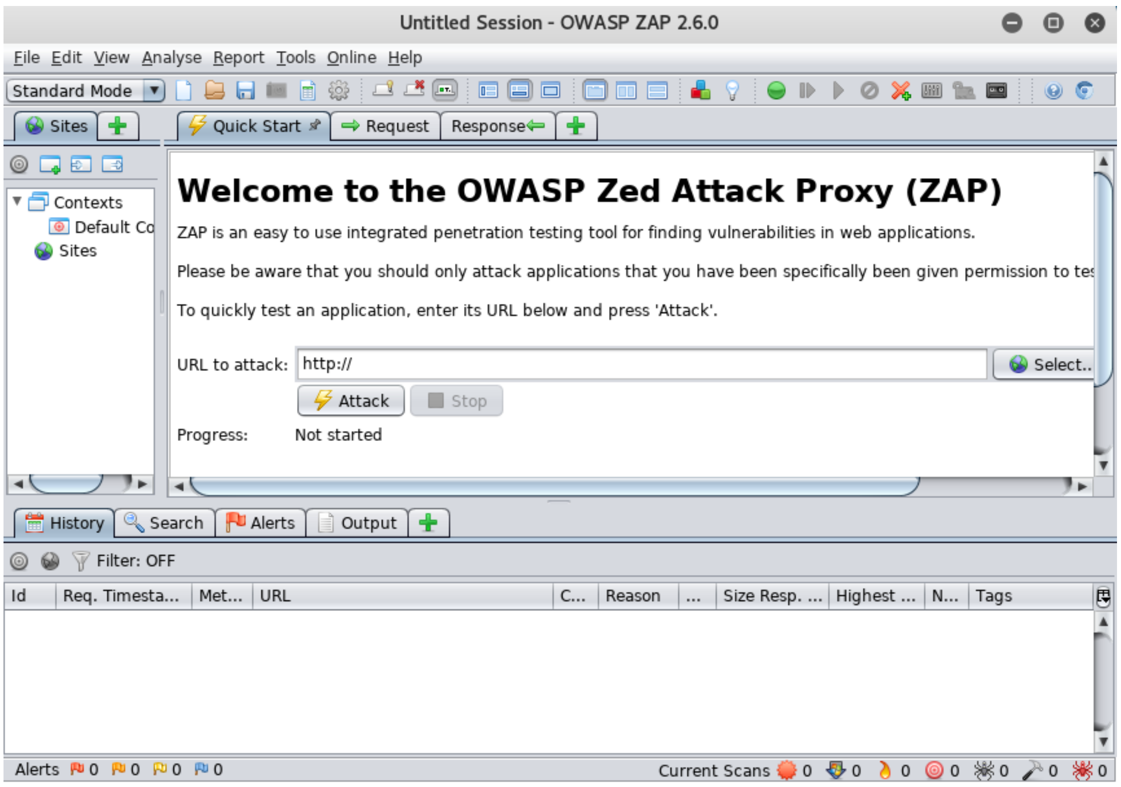

Greenbone Security Assistant
No auto-refresh
Logged in as Admin admin | Logout
Wed Nov 22 01:10:09 2017 UTC
Scans
Assets
SecInfo
Configuration
Extras
Administration
Help
Dashboard
Tasks by Severity Class (Total: 0)
Tasks by status (Total: 0)
CVEs by creation time (To...
CVEs / year
Total CVEs
Hosts topology
No hosts with topology selected
NVTs by Severity Class (T...
High
Medium
Low
Log
2844
2218
28145
23220

Figure 14. OpenVAS web user interface dashboard

OWASP-ZAP

OWASP ZAP (OWASP Zed Attack Proxy) is a free open-source web application security scanner, that is maintained mostly by a group of volunteers. The main function of OWASP-ZAP is to help the user find security deficiencies and vulnerabilities in web applications during development and testing phases. It is also a useful tool for manual security testing (OWASP 2017c).

OWASP-ZAP is a powerful tool that offers many features, but there is also an extremely simple “point and click” -style functionality that requires the user to only provide a target URL as can be seen in Figure 15.



Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given permission to tes

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http:// Select..

Attack Stop

Progress: Not started

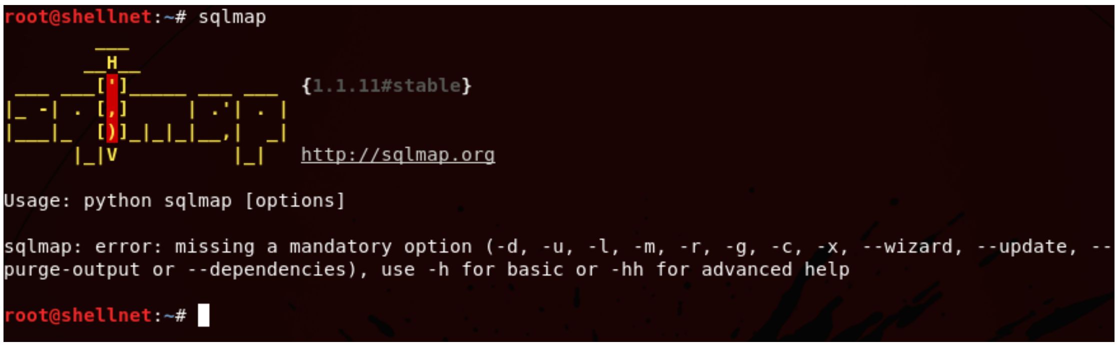
Figure 15. OWASP Zed Attack Proxy quick start

SQLMAP

Sqlmap is an open source penetration testing and auditing tool for automated detection and exploitation of numerous known SQL injection vulnerabilities in multiple database management systems in order to gain control of servers containing databases and the databases themselves (Guimaraes & Stampar 2017).

When used correctly against a vulnerable website application, Sqlmap will be able to perform multiple tasks, such as retrieving data from the database behind the web application and accessing the file system of the host machine.

After installation and configuration, the sqlmap command prints out a banner and short usage instruction, as shown in Figure 16.



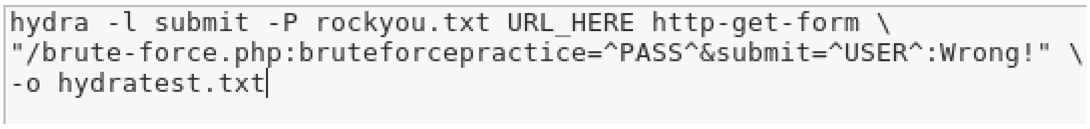
```
root@shellnet:~# sqlmap
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.1.11#stable}
|_ -| . [,]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

Usage: python sqlmap [options]

sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, -x, --wizard, --update, --
purge-output or --dependencies), use -h for basic or -hh for advanced help

root@shellnet:~# 
```

Figure 16. sqlmap command

THC-Hydra

THC-hydra is a free tool for cracking passwords, and it supports numerous different protocols. Typically, hydra is used to crack network logins via dictionary- and/or brute force -attack methods (Hauser 2017).

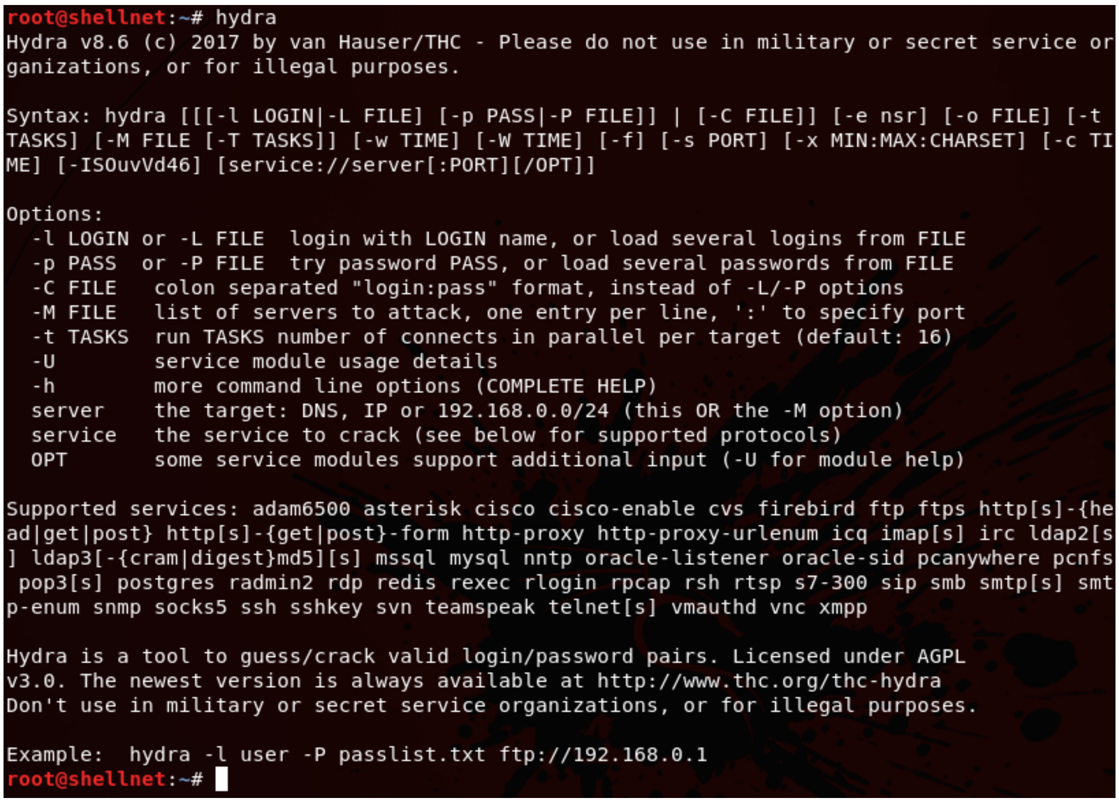
Hydra notoriously has quite a complex syntax for beginners and the fully constructed commands are often fairly long, but once the logic behind the tool is understood, it is not too hard to create a working command tailored specifically for the user's needs. An example of a more complex command can be seen in Figure 17.



```
hydra -l submit -P rockyou.txt URL_HERE http-get-form \
"/brute-force.php:bruteforcepractice=^PASS^&submit=^USER^:Wrong!" \
-o hydratest.txt
```

Figure 17. Command for dictionary attack with THC-Hydra

By running the program, it prints out a simplified version of the help menu. This menu contains the syntax hydra uses, some basic options, services it supports, licensing information and one simple example, as shown in Figure 18.



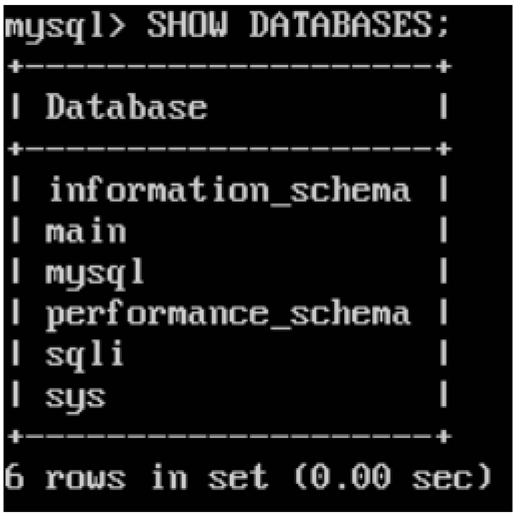
```
root@shellnet:~# hydra
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service or
ganizations, or for illegal purposes.

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t
TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TI
ME] [-ISOuvVd46] [service://server[:PORT][/OPT]]

Options:
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -t TASKS  run TASKS number of connects in parallel per target (default: 16)
  -U        service module usage details
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp ftps http[s]-{he
ad|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s
] ldap3[-{cram|digest}md5][s] mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs
 pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smt
p-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
v3.0. The newest version is always available at http://www.thc.org/thc-hydra
Don't use in military or secret service organizations, or for illegal purposes.

Example:  hydra -l user -P passlist.txt ftp://192.168.0.1
root@shellnet:~#
```

Figure 18. Hydra command

3.3.3 Databases

The database management system used is MySQL. The reasoning behind that is the ease of use, fast deployment and popularity. According to DB-engines (2017), as of November 2017 MySQL is ranked as the second most popular database management system. High popularity increases the chances of encountering more MySQL databases in the real world, which makes it a great candidate to study and practice exploitation against.

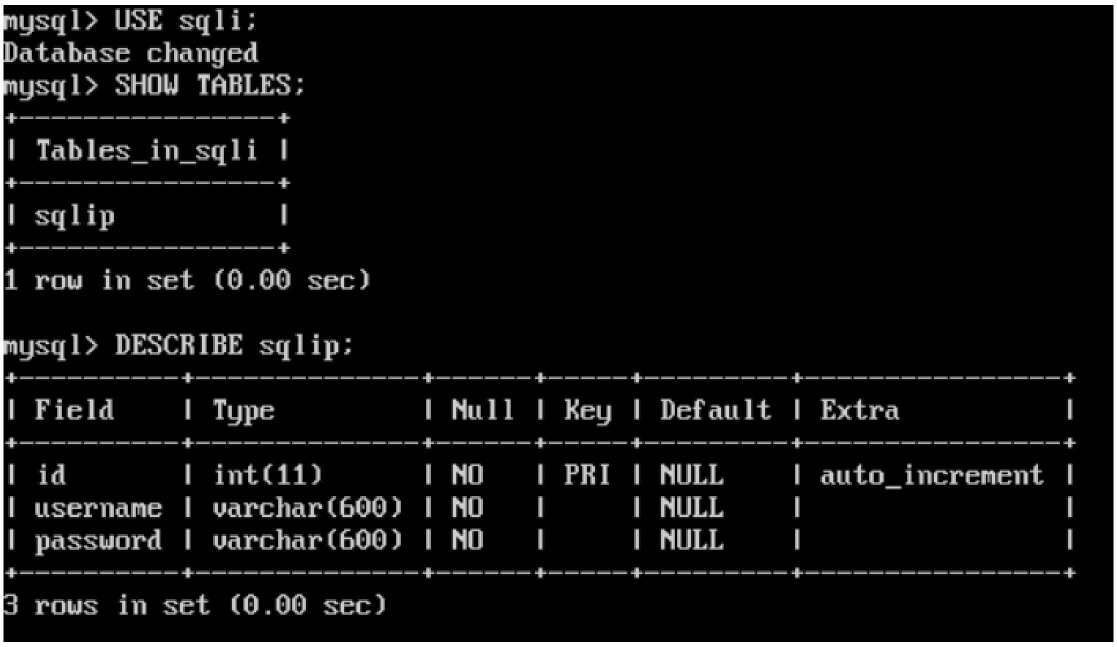
There are two different databases created for the Hackstructor’s Ubuntu web server. These databases are required for SQL injection- and stored XSS vulnerabilities to function properly. The names for the two databases are “main” and “sqli”, as shown in Figure 19.



```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| main               |
| mysql              |
| performance_schema |
| sqli               |
| sys                |
+--------------------+
6 rows in set (0.00 sec)
```

Figure 19. MySQL databases

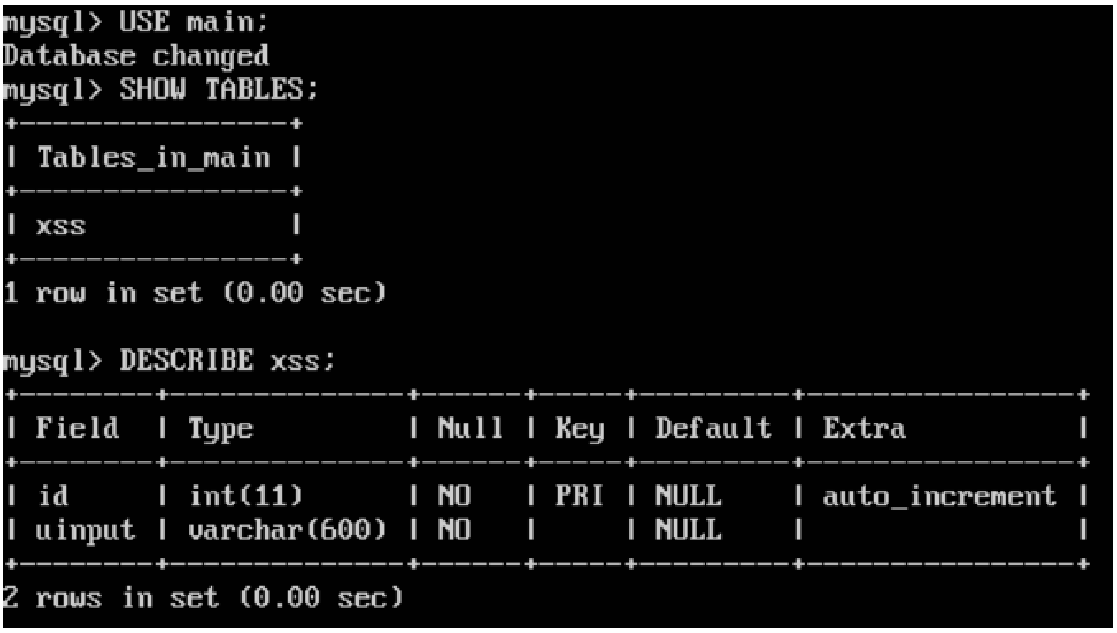
The sqli database has one table called “sqlip”, and that table has three fields called “id”, “username” and “password”. The function of this database is to store username and password combinations for the purpose of practicing SQL injection. The SQL injection- and blind SQL injection vulnerability pages rely on this database for their practice sections. Figure 20 shows the sqli database- and table structure.



```
mysql> USE sqli;
Database changed
mysql> SHOW TABLES;
+----------------+
| Tables_in_sqli |
+----------------+
| sqlip          |
+----------------+
1 row in set (0.00 sec)

mysql> DESCRIBE sqlip;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int(11)      | NO   | PRI | NULL    | auto_increment |
| username | varchar(600) | NO   |     | NULL    |                |
| password | varchar(600) | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

Figure 20. sqli database- and table structure

The database called “main” also has just one table. This table includes two fields called “id” and “uinput”. The purpose of this database is to store strings that a user has input. These strings are then printed out into the web server’s stored XSS vulnerability page in a message board -like manner. Keeping that in mind, the practice section of the XSS vulnerability page is the sole reason this database is required. The database- and table structure of *main* can be seen in Figure 21.

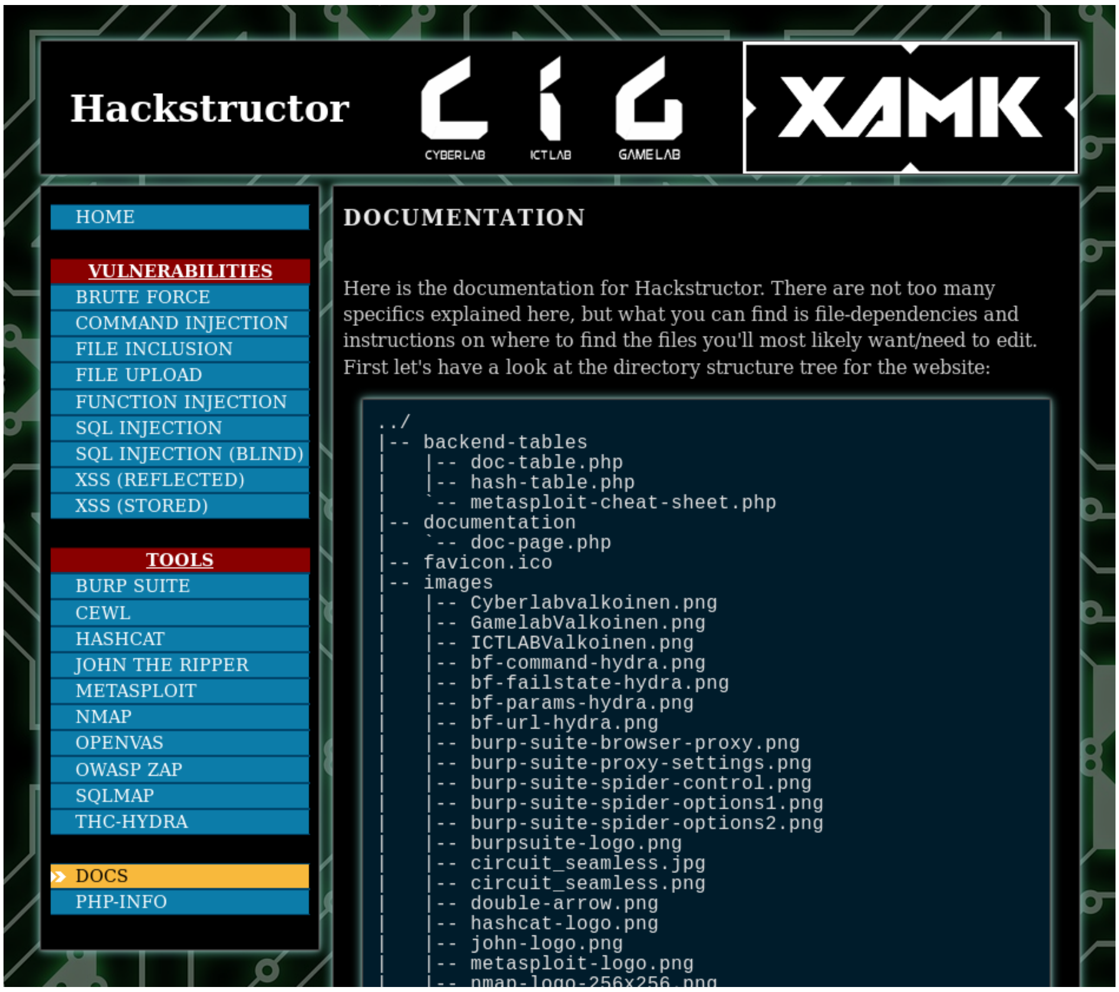


```
mysql> USE main;
Database changed
mysql> SHOW TABLES;
+----------------+
| Tables_in_main |
+----------------+
| xss            |
+----------------+
1 row in set (0.00 sec)

mysql> DESCRIBE xss;
+--------+--------------+------+-----+---------+----------------+
| Field  | Type         | Null | Key | Default | Extra          |
+--------+--------------+------+-----+---------+----------------+
| id     | int(11)      | NO   | PRI | NULL    | auto_increment |
| uinput | varchar(600) | NO   |     | NULL    |                |
+--------+--------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

Figure 21. main database- and table structure

3.3.4 Other back end content

There is some notable back-end related content in the Hackstructor’s web server that is worth mentioning. First there is a truncated documentation for possible future developers which lists all the databases, directories and files that are included in the web server, as well as what those files contain and what files should be modified if someone would like to create a new vulnerability listing for example. As can be seen in Figure 22, this is the web page that corresponds to the “DOCS” menu item in the side panel.



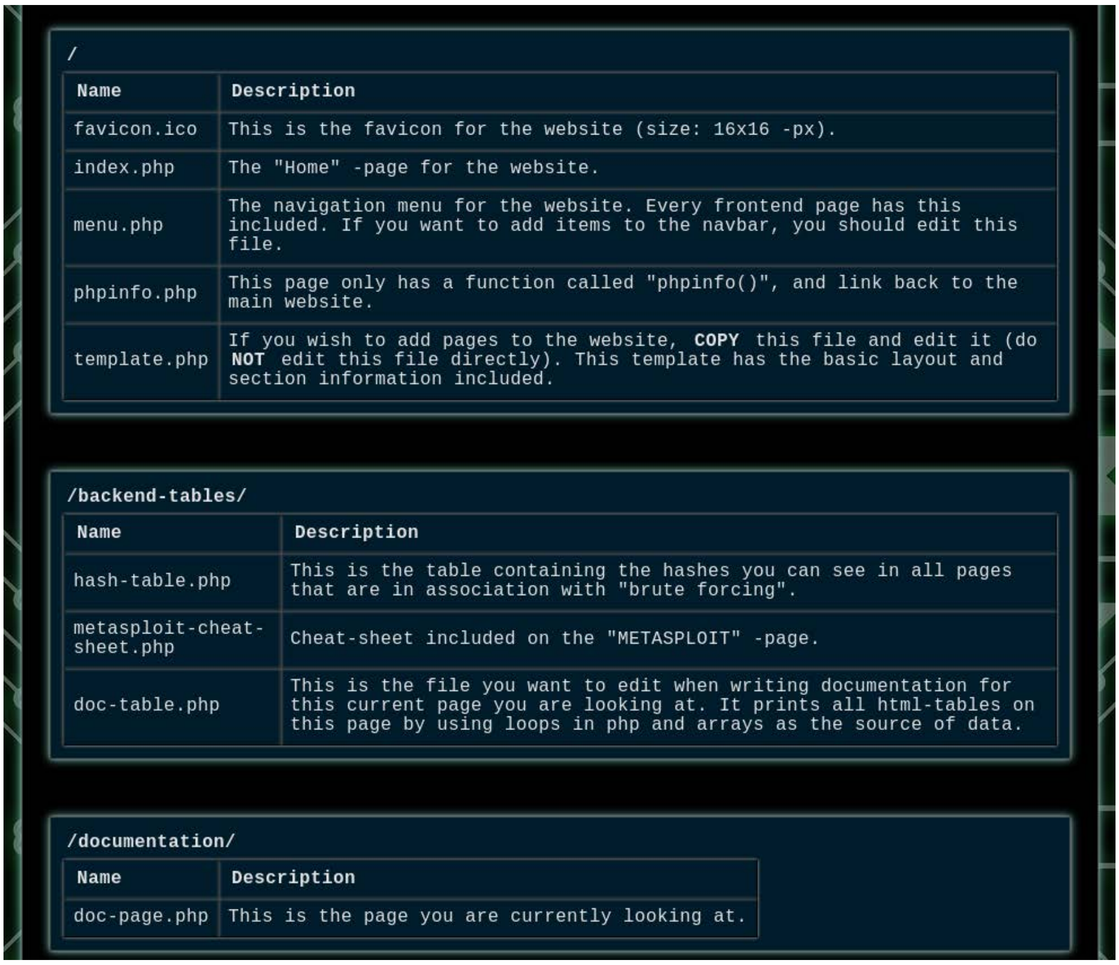
DOCUMENTATION

Here is the documentation for Hackstructor. There are not too many specifics explained here, but what you can find is file-dependencies and instructions on where to find the files you'll most likely want/need to edit. First let's have a look at the directory structure tree for the website:

```
../
|-- backend-tables
|    |-- doc-table.php
|    |-- hash-table.php
|    `-- metasploit-cheat-sheet.php
|-- documentation
|    `-- doc-page.php
|-- favicon.ico
|-- images
|    |-- Cyberlabvalkoinen.png
|    |-- GamelabValkoinen.png
|    |-- ICTLABValkoinen.png
|    |-- bf-command-hydra.png
|    |-- bf-failstate-hydra.png
|    |-- bf-params-hydra.png
|    |-- bf-url-hydra.png
|    |-- burp-suite-browser-proxy.png
|    |-- burp-suite-proxy-settings.png
|    |-- burp-suite-spider-control.png
|    |-- burp-suite-spider-options1.png
|    |-- burp-suite-spider-options2.png
|    |-- burpsuite-logo.png
|    |-- circuit_seamless.jpg
|    |-- circuit_seamless.png
|    |-- double-arrow.png
|    |-- hashcat-logo.png
|    |-- john-logo.png
|    |-- metasploit-logo.png
```

Figure 22. DOCS -page of Hackstructor

In Figure 23 there are shown some of the explanations for the files included in Hackstructor’s web server.



/

Name	Description
favicon.ico	This is the favicon for the website (size: 16x16 -px).
index.php	The "Home" -page for the website.
menu.php	The navigation menu for the website. Every frontend page has this included. If you want to add items to the navbar, you should edit this file.
phpinfo.php	This page only has a function called "phpinfo()", and link back to the main website.
template.php	If you wish to add pages to the website, **COPY** this file and edit it (do **NOT** edit this file directly). This template has the basic layout and section information included.

/backend-tables/

Name	Description
hash-table.php	This is the table containing the hashes you can see in all pages that are in association with "brute forcing".
metasploit-cheat-sheet.php	Cheat-sheet included on the "METASPLOIT" -page.
doc-table.php	This is the file you want to edit when writing documentation for this current page you are looking at. It prints all html-tables on this page by using loops in php and arrays as the source of data.

/documentation/

Name	Description
doc-page.php	This is the page you are currently looking at.

Figure 23. Files and their explanations in Hackstructor's DOCS -page

4 FUTURE DEVELOPMENT

There is a limited, but still a great number of possibilities for future development of the Hackstructor platform. Inclusion of new systems and devices, addition of new content or even new features would all be growing the size and magnitude of the platform even higher than it is now. In this chapter I will talk about some of the possibilities for future development.

4.1 Systems and devices

By including more systems and devices into the platform, it is possible to create even more content for the web application itself. For example, the inclusion of new computer systems behind badly configured switches or routers would add even more in-depth learning possibilities into the Hackstructor platform. There could also be firewall, SIEM, IDS or even IPS solutions included, although that would decrease the effectiveness of using the platform’s web server as a single modular virtual machine image.

4.2 Content

More content can be created simply by using the templates and instructions provided in the platform itself. There are enormous amounts of security focused tools available as well as vulnerabilities. Future content development could also include new examples, updates to the current content, new cheatsheets or even completely new categories for the web user interface.

Possible examples could include more in-depth versions regarding the usage of the current exploitation tools. As the development of different kinds of exploitation tools advances constantly, examples are required to be updated in order to keep the platform up to date.

A new category for the platform could be something related to social engineering for example. Social engineering is something that is not included at all in the Hackstructor platform, and it could open up completely new perspectives on how to deliver exploits and on vulnerability exploitation in general.

4.3 Features

In the case that the platform grows to include multiple new vulnerabilities, tools, and even completely new categories, it would be wise to add some features for at least to the web user interface to ensure scalability. For example, the ability to search via a search bar would be a great addition in case there seems to be too much information to navigate through with other methods. One more feature that could be worth implementing is the ability for users to change the difficulty of the practice sections, similarly to the DVWA (Damn Vulnerable Web Application).

There could also be practice exams for users to participate in, as well as “real” exams, which would mean that there should be scoring and grading systems implemented to the platform. With the implementation of these features and other future development ideas discussed earlier, Hackstructor could evolve into a complete platform for teachers to use in cybersecurity related courses; teachers could use Hackstructor as a course material, they could refer to it when students have cybersecurity related questions, and teachers could even utilize Hackstructor in exams for the courses they are teaching.

5 CONCLUSIONS

The constant development of new security tools and the rapid growth of information technology in general, means that the Hackstructor platform would have to be under constant development to be a relevant studying platform for longer periods of time. The topic of this thesis is very broad, as some of the vulnerabilities and tools that are mentioned inside Hackstructor, such as SQL injection and Metasploit Framework, could be their own topics for a thesis.

The project has required an extraordinary amount of research on vulnerabilities and security tools all the way up to networking and back end programming. The project has also been fairly time consuming, because in addition to the research, every page in the web interface of Hackstructor is aimed to be an informative essay.

In my opinion, the goals of this thesis were completed and Hackstructor is ready to be used by students in cybersecurity related courses. I am also positive that the Hackstructor platform can be further developed by teachers and students in the future.

REFERENCES

Aharoni, M., Hertzog, R. & O'Gorman, J. 2017. Kali Linux Revealed. 2.-3. Cornelius NC: Offsec Press.

Brenner, B. 2017. WannaCry: the ransomware worm that didn't arrive on a phishing hook. Available at: https://nakedsecurity.sophos.com/2017/05/17/wannacry-the-ransomware-worm-that-didnt-arrive-on-a-phishing-hook/ [Accessed 15 November 2017].

Cormany, A. 2009. AIX user and group administration. Available at: https://www.ibm.com/developerworks/aix/library/au-aixuseradmin/index.html [Accessed 28 November 2017].

DB-engines 2017. DB-Engines Ranking. Available at: https://db-engines.com/en/ranking [Accessed 26 November 2017].

DVWA 2017. README. Available at: https://github.com/ethicalhack3r/DVWA/blob/master/README.md [Accessed 12 November 2017].

Guimaraes, D. A. B. & Stampar, M. 2017. Sqlmap Automatic SQL injection and database takeover tool. Available at: http://sqlmap.org/ [Accessed 18 November 2017].

Hashcat s.a. a. Hashcat suite. Available at: https://hashcat.net/wiki/ [Accessed 11 November 2017].

Hashcat s.a. b. Hashcat Frequently Asked Questions. Available at: https://hashcat.net/wiki/doku.php?id=frequently_asked_questions [Accessed 14 November 2017].

Hauser, V. 2017. THC-Hydra. Available at: https://www.thc.org/thc-hydra/ [Accessed 25 November 2017].

Kettunen, M. 2017. XAMK Virtuaalilaboratorio. Available at: https://www.ictlab.fi/images/kyberturvallisuus/tekstit/XAMK-Virtuaalilaboratorio.pdf [Accessed 29 November 2017].

Lyon, G. 2008. NMAP Network Scanning. 1. Sunnyvale, CA. Insecure.Com LLC.

Maldevel 2012. Metasploit – Interfaces – Part 2. Available at: https://securityblog.gr/1042/metasploit-interfaces-part-2/ [Accessed 10 November 2017].

Microsoft 2017. Support for Windows XP ended. Available at: https://www.microsoft.com/en-us/windowsforbusiness/end-of-xp-support [Accessed 18 November 2017].

Nurmi, J. 2016. Implementation of Nested Virtual Laboratory System. Available at: http://urn.fi/URN:NBN:fi:amk-201604204658 [Accessed 16 November 2017].

Offensive security 2017. File Inclusion Vulnerabilities. Available at: https://www.offensive-security.com/metasploit-unleashed/file-inclusion-vulnerabilities/ [Accessed 13 November 2017].

OpenVAS s.a. The world's most advanced Open Source vulnerability scanner and manager. Available at: http://www.openvas.org/index.html [Accessed 23 November 2017].

Openwall 2013. John the Ripper's cracking modes. Available at: http://www.openwall.com/john/doc/MODES.shtml [Accessed 20 November 2017].

OWASP 2016a. OWASP WebGoat Project. Available at: https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project [Accessed 12 November 2017].

OWASP 2016b. Brute force attack. Available at: https://www.owasp.org/index.php/Brute_force_attack [Accessed 26 November 2017].

OWASP 2016c. Command Injection. Available at: https://www.owasp.org/index.php/Command_Injection [Accessed 21 November 2017].

OWASP 2016d. Function Injection. Available at: https://www.owasp.org/index.php/Function_Injection [Accessed 17 November 2017].

OWASP 2016e. SQL Injection. Available at: https://www.owasp.org/index.php/SQL_Injection [Accessed 17 November 2017].

OWASP 2016f. Blind SQL Injection. Available at: https://www.owasp.org/index.php/Blind_SQL_Injection [Accessed 17 November 2017].

OWASP 2016g. Cross-site Scripting (XSS). Available at: https://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29 [Accessed 12 November 2017].

OWASP 2017a. The Ten Most Critical Web Application Security Risks. Available at: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_(en).pdf.pdf [Accessed 29 November 2017].

OWASP 2017b. Unrestricted File Upload. Available at: https://www.owasp.org/index.php/Unrestricted_File_Upload [Accessed 25 November 2017].

OWASP 2017c. OWASP Zed Attack Proxy Project. Available at: https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project [Accessed 19 November 2017].

Portswigger Ltd. 2017a. Getting Started With Burp Suite. Available at: https://portswigger.net/burp/help/suite_gettingstarted [Accessed 25 November 2017].

Portswigger Ltd. 2017b. Burp Suite Editions. Available at: https://portswigger.net/burp [Accessed 21 November 2017].

Rapid7 metasploit s.a. The world’s most used penetration testing framework. Available at: https://www.metasploit.com/ [Accessed 27 November 2017].

Rapid7 s.a. a. Metasploitable 2 Exploitability Guide. Available at: https://metasploit.help.rapid7.com/v1.1/docs/metasploitable-2-exploitability-guide [Accessed 22 November 2017].

Rapid7 s.a. b. Metasploit Pen Testing Tool. Available at: https://www.rapid7.com/products/metasploit/download/editions/ [Accessed 26 November 2017].

Wood, R. 2017. CeWL – Custom Word List generator. Available at: https://github.com/digininja/CeWL/blob/master/README [Accessed 27 November 2017].

LIST OF FIGURES