**Body of Knowledge (BOK)**

Mohammed Nasser Alshukaili.

ICT, Fontys University of Applied Sciences.

Cyber Security, Semester 4.

September 2022

Table of Content

[INTRODUCTION 3](#_Toc114523723)

[IT Background 3](#_Toc114523724)

[(BOK) Web Application Security (week 1-4) 4](#_Toc114523725)

[Exercises & Challenges 5](#_Toc114523726)

[Lesson 1.3 (Security Threat + Risk Analysis for the test Web Shop) 5](#_Toc114523727)

[Ransomware 5](#_Toc114523728)

[Phishing 6](#_Toc114523729)

[Viruses 7](#_Toc114523730)

[Lesson 2.2 (Host Intrusion Detection and Prevention (HIDS) 10](#_Toc114523731)

[Host Intrusion Detection and Prevention System (HIDS) 10](#_Toc114523732)

[Lesson 3.2 (Web Application Firewalls) 23](#_Toc114523733)

[DVWA (Damn Vulnerable Web Application) 27](#_Toc114523734)

[Path Traversal 28](#_Toc114523735)

[Brute Force 28](#_Toc114523736)

[Brute Force – Low level 30](#_Toc114523737)

[SQL injection 32](#_Toc114523738)

[SQLi – Low level 33](#_Toc114523739)

[SQLi – Medium level 34](#_Toc114523740)

[SQLi – Online Challenges 34](#_Toc114523741)

[Command Injection 38](#_Toc114523742)

[Command Injection – Low level 38](#_Toc114523743)

[Command Injection – Medium level 40](#_Toc114523744)

[CSRF (4.1) 42](#_Toc114523745)

[CSRF – Low Level 42](#_Toc114523746)

[XSS DOM (4.1) 43](#_Toc114523747)

[XSS – Low Level 43](#_Toc114523748)

[XSS – Medium Level 44](#_Toc114523749)

[XSS – High Level 45](#_Toc114523750)

[XSS Stored (4.1) 46](#_Toc114523751)

[Juice Shop 47](#_Toc114523752)

[(BOK) Network Security (week 5-8) 49](#_Toc114523753)

[(BOK) Security Concepts (week 9-10) 49](#_Toc114523754)

[References 49](#_Toc114523755)

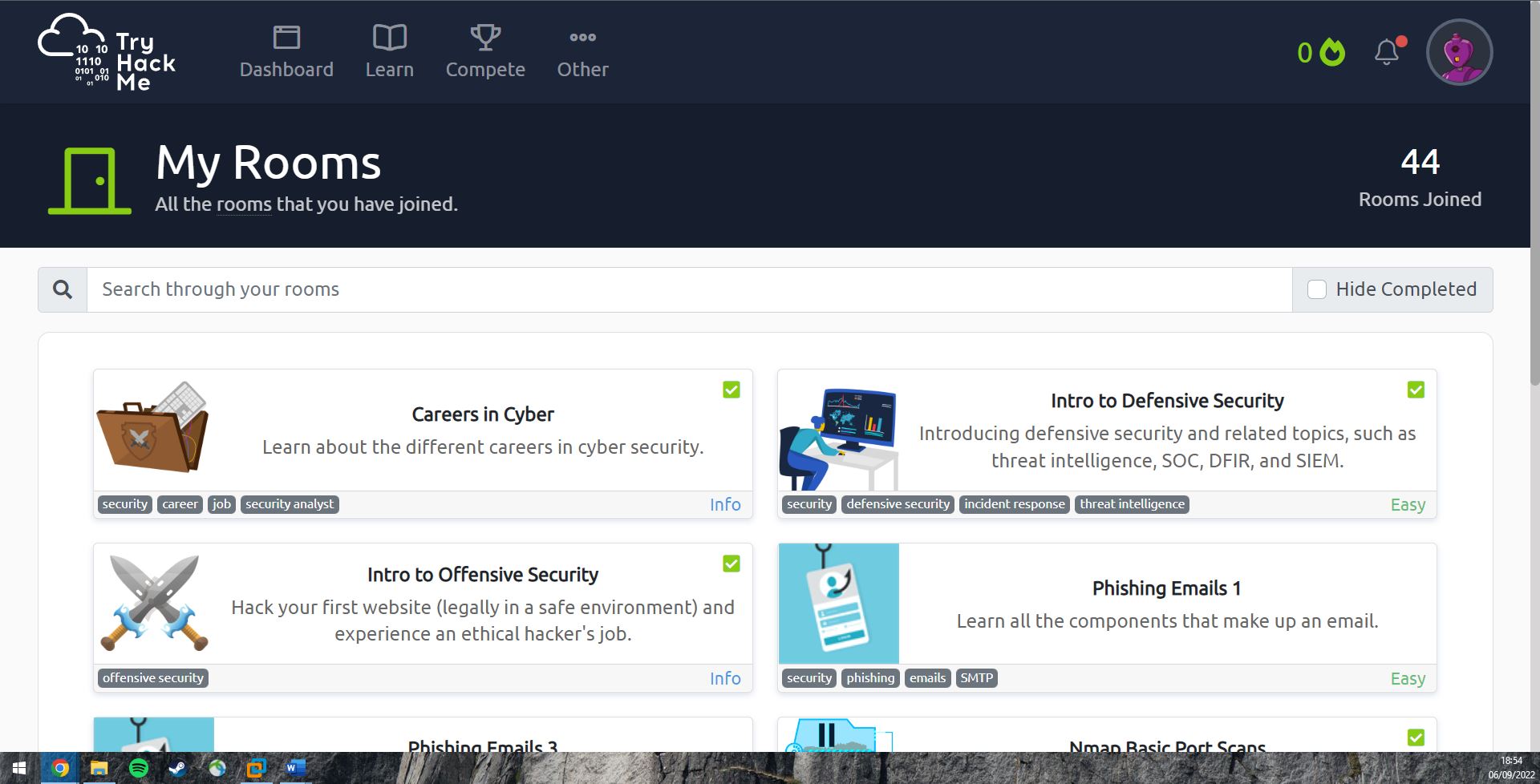
# INTRODUCTION

This report will take you through my learning process and semester progress for the Fontys, ICT, Cyber Security (S4) course.

My name is Mohammed Alshukaili. Although I come from media design background, my passion is to study cyber security. I consider myself an absolute beginner in this field. I will do whatever it takes to become a good student throughout the whole semester.

# IT Background

I only studied infrastructure engineering for 10 weeks in my first semester at Fontys. I felt like this was not going to be enough to become at a high-level student. Therefore, I decided to invest all my summer breaks studying new things that are related to infrastructure.

For instance, I subscribed to [www.tryhackme.com](http://www.tryhackme.com), a website that provides a huge amount of IT courses. I finished all the fundamentals for Linux, windows, and online databases. 

This picture shows some of the rooms that I enrolled for. I finished most of them with so much information gained.

I also like to learn by watching sometimes. I am subscribed to so many IT YouTube channels that provide basics for the IT industry. Some of my favorite YouTube channels are

* <https://www.youtube.com/c/PowerCertAnimatedVideos>
* <https://www.youtube.com/c/H>

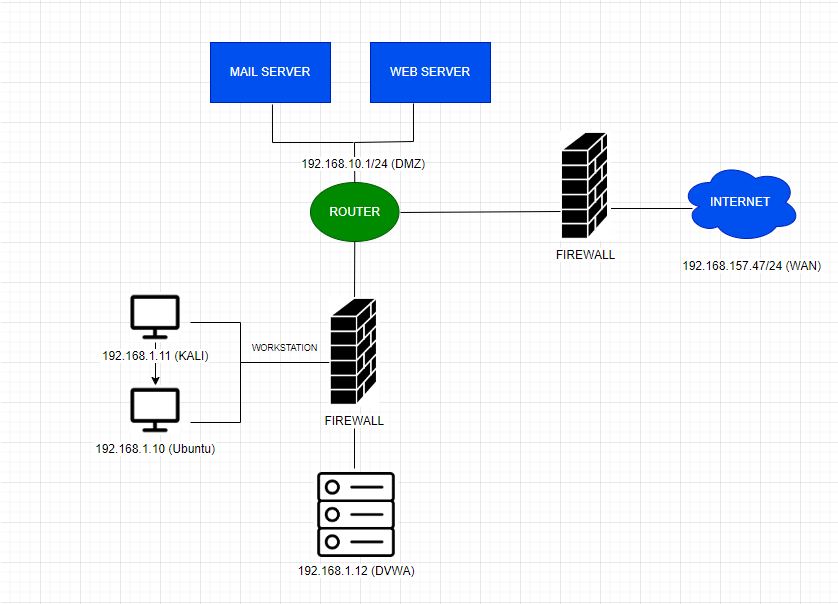
What I realized after all the tutorials that I have been to is that I prefer reading rather than watching tutorials. As the visual tutorials usually do not give me the chance to do things myself. On the other side, services like TryHackMe, Fontys are more practical that I found myself enjoy it more.

# (BOK) Web Application Security (week 1-4)

First day was a refresher to warm up ourselves. We made groups and competed by getting more points. The website name was picoCTF that gives you points by solving technical problems. The more you solve the more your group gets.

The second day, I attended the networking refresh. We learned about vSphere and how to set up a local virtual network. I thought that we would need to borrow some routers and devices to set up local network. However, the VLAN concept really changed the way I see internet. I managed to design a local network that includes 2 firewalls, workstation, DMZ, and the WAN to access the internet.

Here is a picture of my simple design:



I had to learn pfSense to set up this network and some of Docker basics.

Last class of the week, we had a theory about security and the different types of attacks.

# Exercises & Challenges

## Lesson 1.3 (Security Threat + Risk Analysis for the test Web Shop)

Task: To get a better understanding of these threats and attacks find articles on internet about any malware.

### Ransomware

Ransomware is one of the most popular cyber-attacks around the world. It is used to damage as many systems as possible. It encrypts the data in the infected system, this encourages the attacker to ask for money to decrypt the data. This malware spreads rapidly by itself without needing an attacker to manage it. Nobody can expect how fast it would spread to other systems.

#### WannaCry Attack

On May 12, 2017, the UK detected a ransomware called Wanacrypt0r. It affected the hospitals in the UK. After that, it went across the globe, infecting thousands of systems in at least 150 countries.

What did the attacker want?

The attacker promised to decrypt the data after receiving 300$ in Bitcoin. The hackers usually tend to use Bitcoin to get the money, to prevent the banks and governments to trace the money transfers.

What did the attacker damage?

The attack spread rapidly. It infected so many sectors. These included:

* Healthcare
* Emergency
* Security
* Telecom
* Gas
* Petrol
* Automotive
* Education
* Advertising

Who was responsible for WannaCry?

The United States officially blames North Korea for this attack. (What Was WannaCry? | WannaCry Ransomware, n.d.)

### Phishing

Phishing is one of the social engineering methods to conduct an attack.

Social engineering is a term used to describe the attacks that take place by tricking the human to do something that leads to a vulnerability. It is when the attacker aims to trick the user to click on a malicious link.

Phishing types:

* **General Phishing** is a simple attack where the target can be anybody that uses popular services like PayPal, Amazon, Steam, etc.
* **Spearphishing** targets small group. However, that group might have security experience because they know that they can be targeted anytime.
* **Whaling** is so specific that it only targets the big names in a company. Well designed and crafted emails are being used for in this type.

Let us take a deep look at the general phishing type which is the most popular.

1. The attacker sends out an email that contains a malicious code.
2. The user (victim) sees the email and thinks that they should log in and do something about it.
3. The website retrieves the user credentials and forward them to the attacker.
4. The attacker receives the credentials and access the site, thus taking over the victims’ accounts.

How to identify phishing emails?

* The users’ names are replaced with something like “Dear customer”
* The link in the email looks almost exactly like a trusted website but one letter was changed for example.
* When hovering on the link in the email, something suspicious appears on the left bottom instead of the trusted website.

Now that we know how the hacker might get to our system, how can we prevent this from happening? Here are some good steps to stay safe:

* Delete every suspicious email that you detect.
* Report any email that looks harmful.
* Do not open attachments from untrusted emails.
* Do not click on any image in the email as it might be a hidden button that sends you to the attacker’s website.
* If you work from home, make sure to set up a safe network and pay attention to all the emails received.

### Viruses

Computer virus is a type of malicious software that tends to harm the system in some way.

How would you know if your system is being infected by a virus?

* Viruses usually affects the performance of the system. Therefore, if you notice that your system is slower that normal, there is probably a malicious virus inside your system.
* Viruses can force the system to close itself randomly.
* Computer viruses can send out emails from your account without your permission.

What are the ways for the virus to spread?

* USB drivers used to be the most popular way for the virus to be spread. Nowadays, the chance of it being spread through the internet is much higher. Such as, suspicious emails, malicious websites, files that you downloaded to your system, etc.

Types of computer viruses:

* Resident Virus:

This type of malware is only hurting your system when the malicious file is open. When you close the files, it stops infecting.

* Multipartite Virus:

This virus remains in the computer’s memory and try to spread to the other hard drives.

* Direct Action:

This type is very dangerous as it can alter/delete all the files/folders that are in the autoexec.bat. It can also damage the memory and the system performance.

* Web Scripting Virus:

This virus attacks the websites and inject malicious codes in there which can change the normal behavior of the website.

* Network virus:

Network virus is too dangerous as it could affect any connected device in a single network. They are difficult to discover. They can hide in any devise and go to other connected device easily.

How to stay safe?

* Do not open attachments that come from untrusted emails.
* Install a good antivirus and keep it updated.
* Use malwares removal tools for your browser.
* Implement Host/Network Intrusion and Detecting/Preventing systems.
* Do not click Pop-ups on websites.

Torjan horse is a virus that fakes the purpose of it to get onto as many files as possible and infect the. Whereas worms are not a virus as they do not require human interaction to spread and execute.

|  |  |  |  |
| --- | --- | --- | --- |
| Dimension | Information Attribute | Threat | Relevance for Example Company |

|  |  |  |  |
| --- | --- | --- | --- |
| Confidentiality | Exclusiveness | Disclosure |  |
|  | Exclusiveness | Abuse |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Integrity | Correct | Tampering |  |
|  | Complete | Removal |  |
|  | Complete | Addition |  |
|  | Valid | Out of date |  |
|  | Authentic | Forgery |  |
|  | Indisputability | Denial |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Availability | Well timed | Delay |  |
|  | Continuity | Downtime |  |

## Lesson 2.2 (Host Intrusion Detection and Prevention (HIDS)

### Host Intrusion Detection and Prevention System (HIDS)

1. Explain the difference between NIDS and HIDS and IDS and IPS, and the meaning and relevance for your company.
2. Select (open source) HIDS software and deploy it in your network. Show that it triggers on (simulated) malicious activity.
3. Intrusion Detection System (IDS) is a method that strengthen the security against cyber crimes. It detects malicious activities entering/leaving the network and start sending alerts to do something about it.

Intrusion Preventing System (IPS) could do both, alerting and blocking the intruder from getting what they came for.

Host Intrusion Detection System (HIDS)

HIDS only monitors one device in a network. It keeps waiting for any malicious activity on a specific device, then sending alerts.

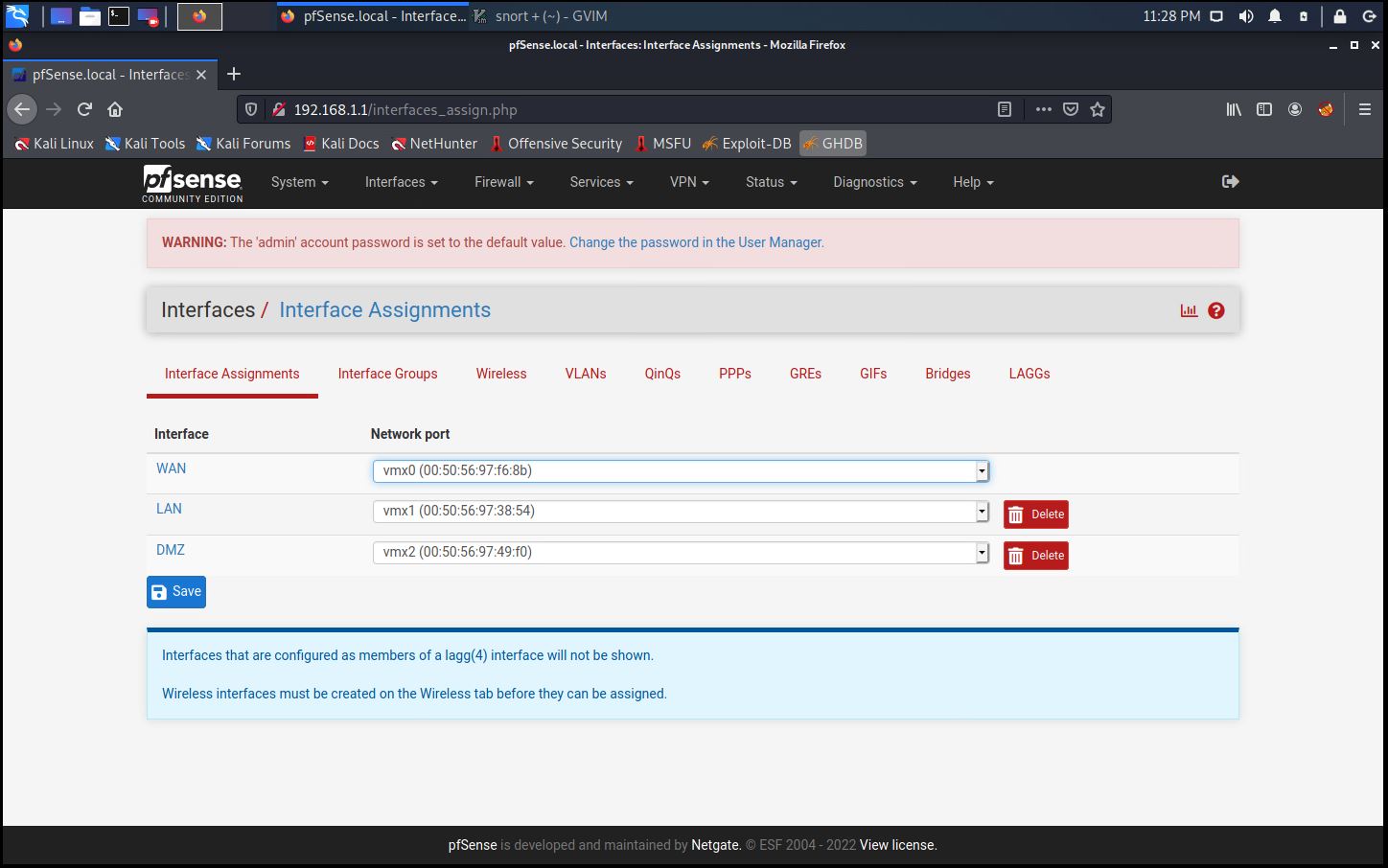
NIDS monitors a network segment. It keeps track of all the data entering and leaving a specific network.

Why would you need to use HIDS when you can just use NIDS to monitor all the devices in the network?

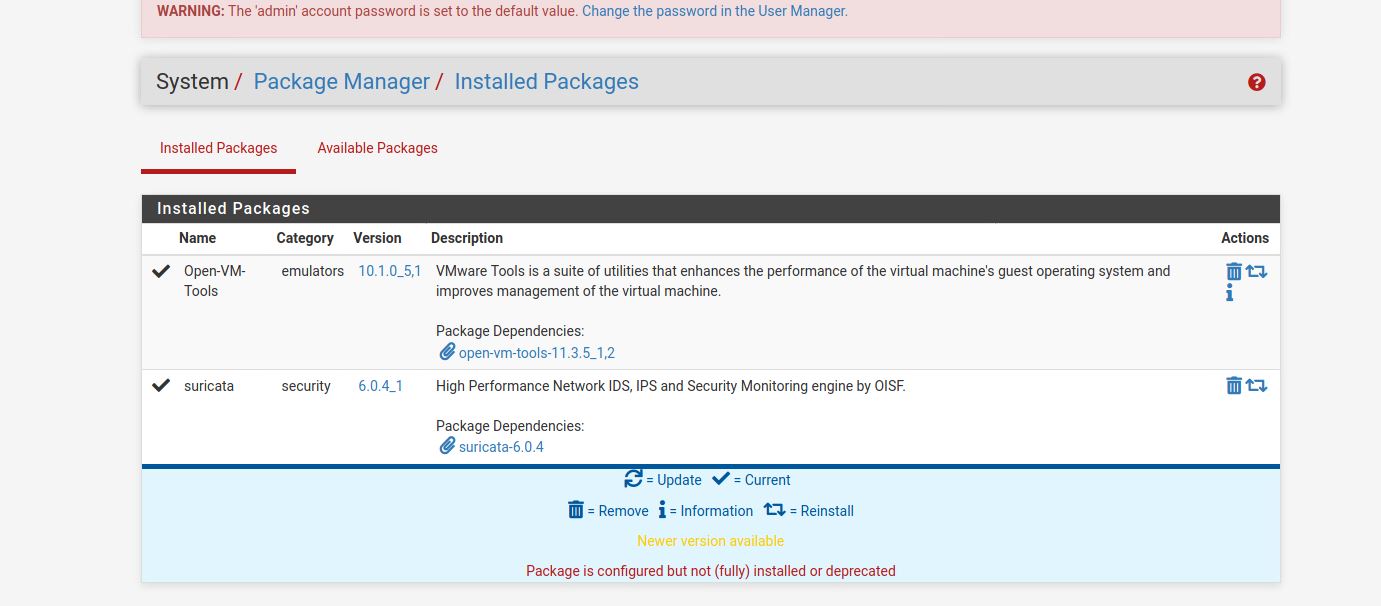
-Well, the intruder might find another path to a specific device and use that path to send malicious packets. Therefore, when having a very private device, HIDS is a better option.

1. Suricata

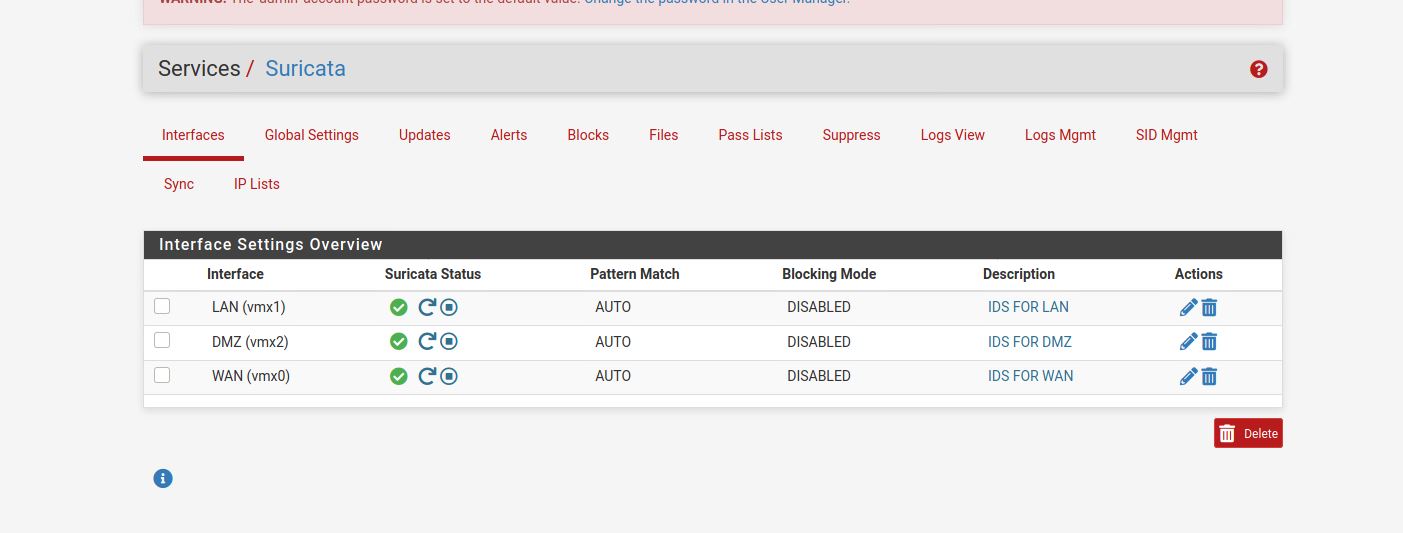
For network-IDS I decided to use Suricata package as it is available in pfSense and free to install.



In my pfSense, I set up 3 interfaces for my local network. I also set up 2 wirefalls. The next step is setting up IDS. I could find the Suricata package in pfSense-System-Package Manager- Available Packages. Then I searched for Suricata and I installed it.

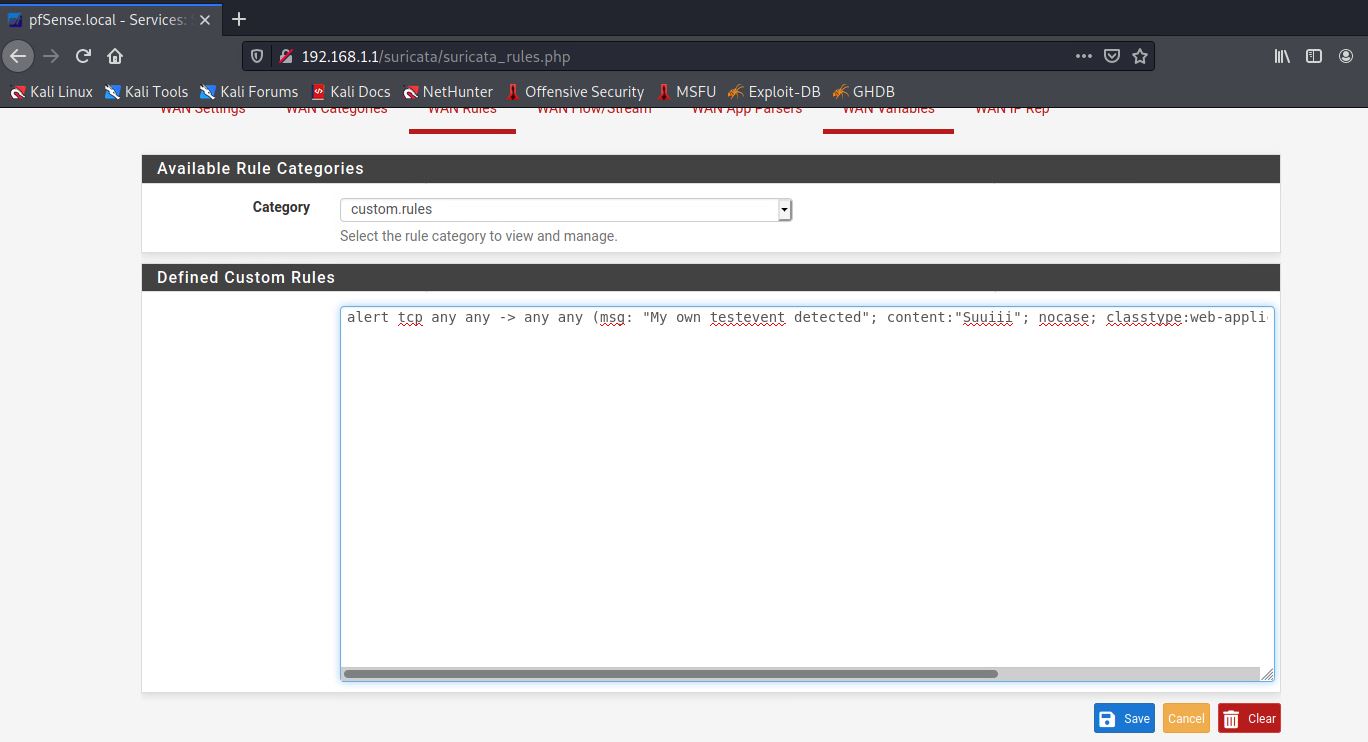


I setup 3 IDS for every interface. However, It says disabled so I still need to enable it.

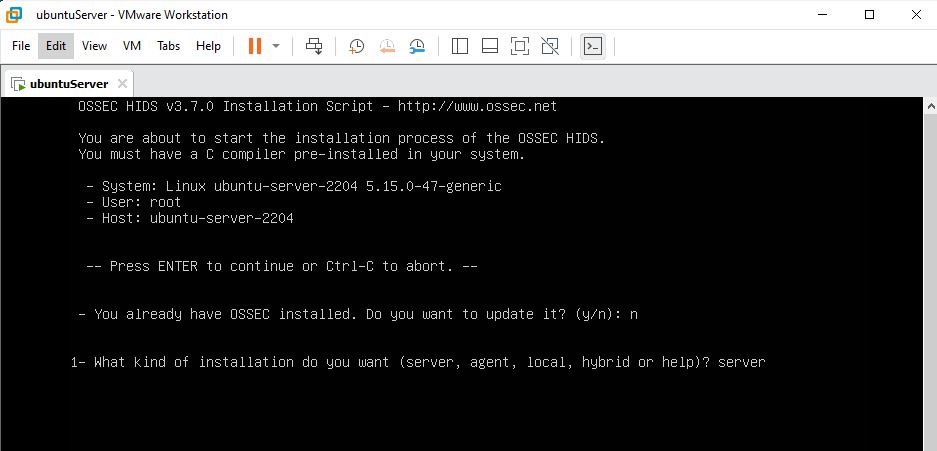


To set rules for the WAN IDS for example, we need to go to Services / Suricata / WAN Interface Settings / WAN – Categories. Then we can check all the rules that we want for the interface’s IDS.

We also can set up custom rules in WAN Rules section



* Setting up OSSEC:

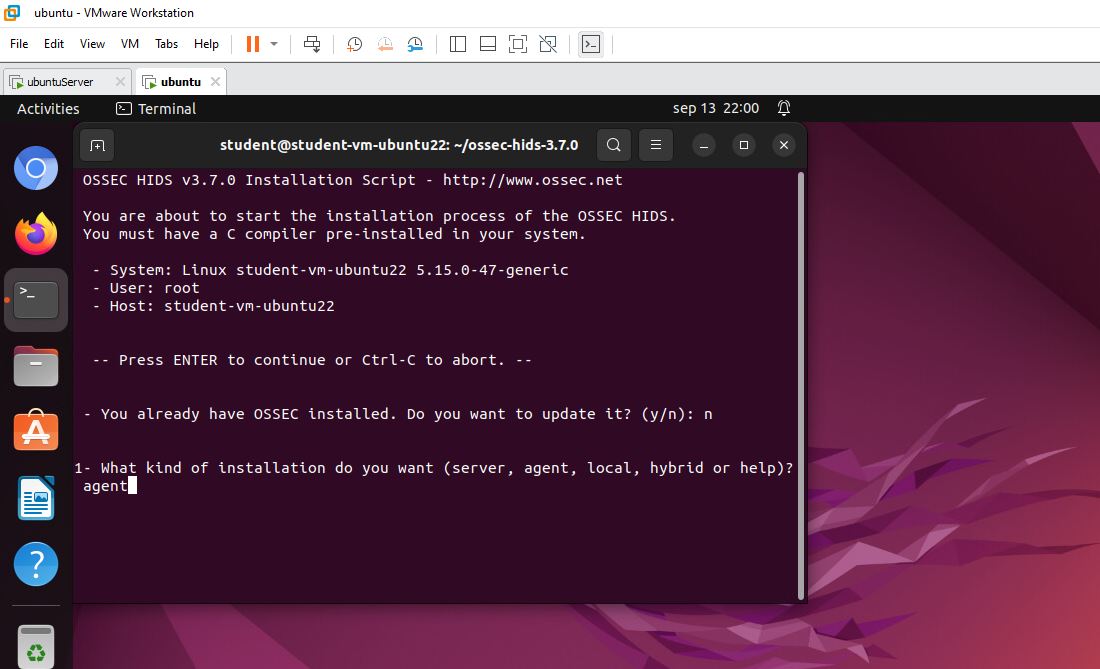


First, I git cloned the latest version of OSSEC from <https://github.com/ossec/ossec-hids>

Then, I unzip the file then I executed the file named install.sh using sudo.

The picture above shows that I am installing the server version because that is my server interface.

This server will need agents. Therefore, I will redo the same steps on my Ubuntu interface. However, when installing, I should install the agent version of ossec.



Now that I have set up OSSEC on server and agent. I could monitor all the requests that any agent sends.

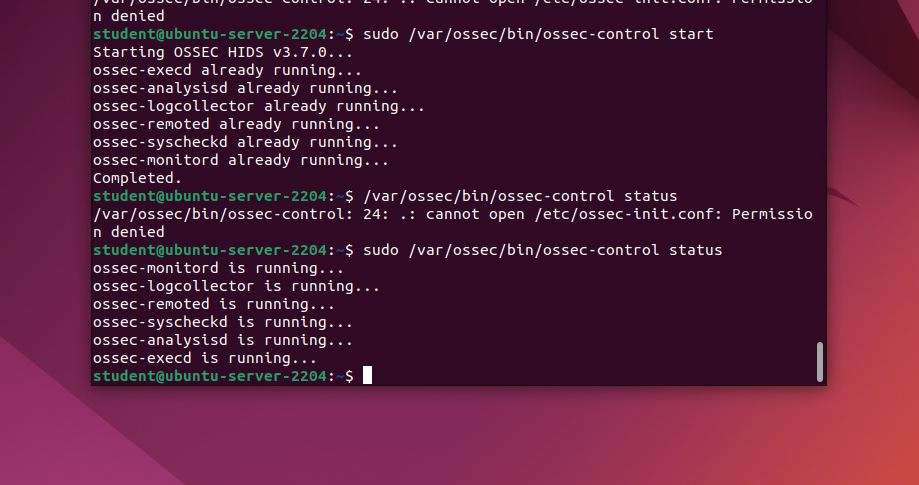
After setting up server OSSEC in the server, and agent OSSEC in Ubuntu. We need to link the agent to the server.

In the server, if we go to /var/ossec/bin/manage\_agents, we can add an agent. Let us add our Ubuntu client:



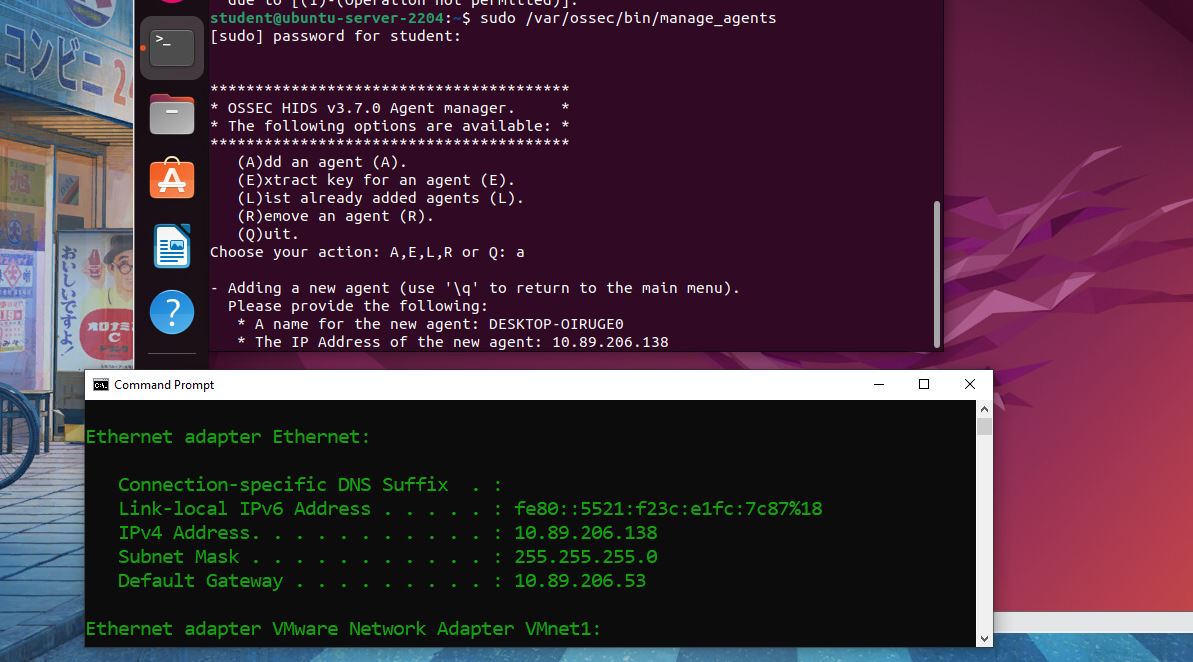
I added the Ubuntu agent and got the key that is unique for it.

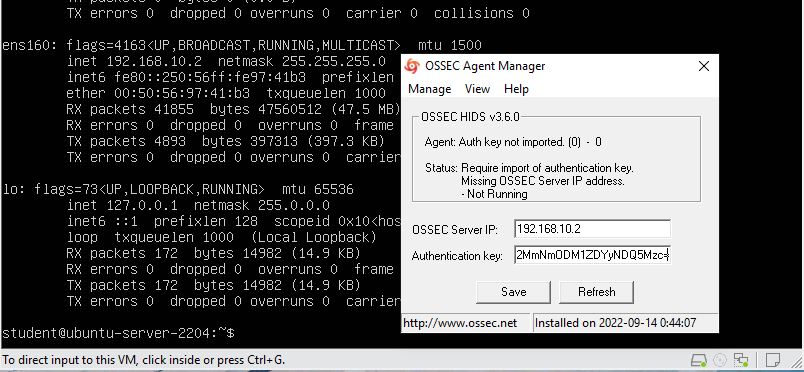
Now let us stop and start OSSEC



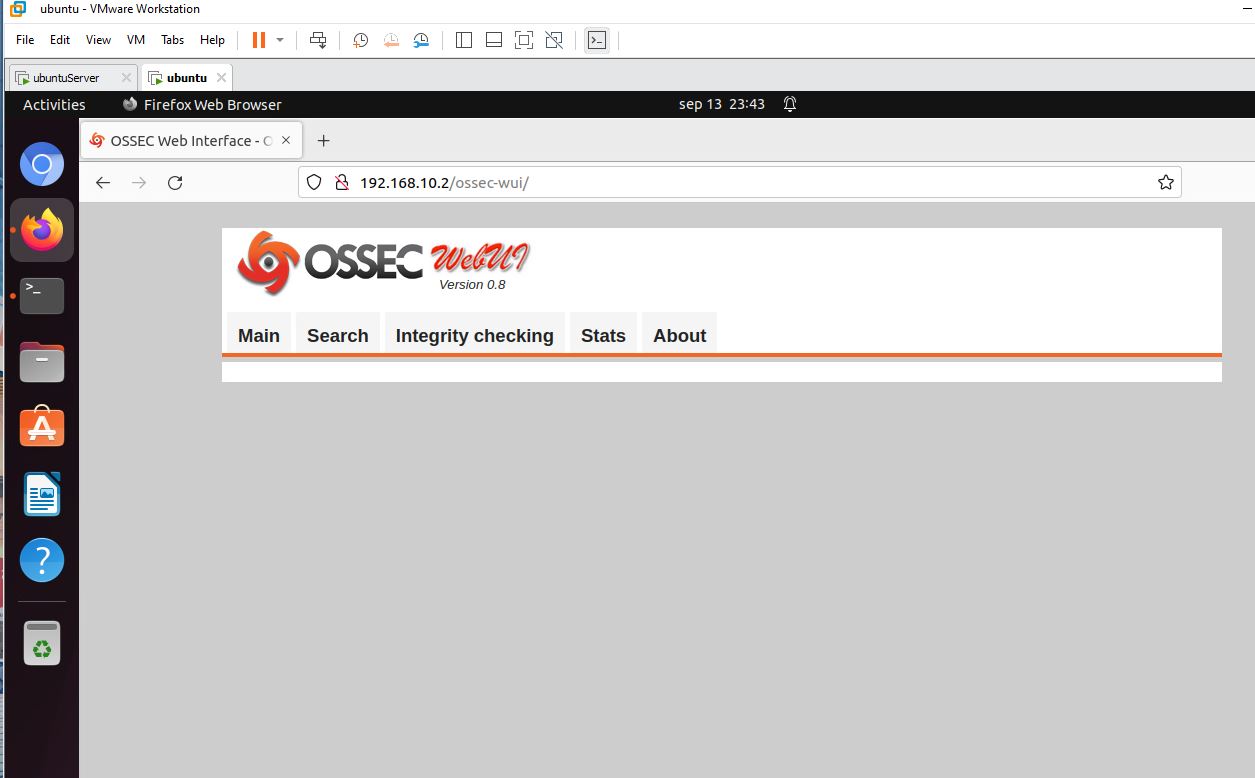
Now we can go to <http://192.168.10.2/ossec-wui> and see the dashboard for OSSEC.

I will do something optional which is installing OSSEC in windows. I could add agent like I did for Ubuntu. Then I enter the windows name and ip. I should get a key that is unique for windows. Install ossec and add the server ip and unique key in the interface.





The problem that I could not fix is that the OSSEC dashboard is empty. I spent hours trying to fix it and I looked for many websites, but nothing worked. However, I managed to make it work in the terminal only.



This is the link that I followed for the first time:

<https://kifarunix.com/install-ossec-hids-agent-on-ubuntu-20-04/>

I will format both, the server and Ubuntu and I will do the whole thing again with some changes.

First, I used this command to remove everything related to ossec from both machines:

/var/ossec/bin/ossec-control stop && rm -rf /var/ossec && rm /etc/init.d/\*ossec\* && rm /etc/ossec-init.conf

This time, I will follow this guide to install ossec.

https://hendgrow.com/2020/10/01/ossec-open-source-hids-with-web-user-interface-updated-for-ubuntu-20-04-ossec-3-6-0/

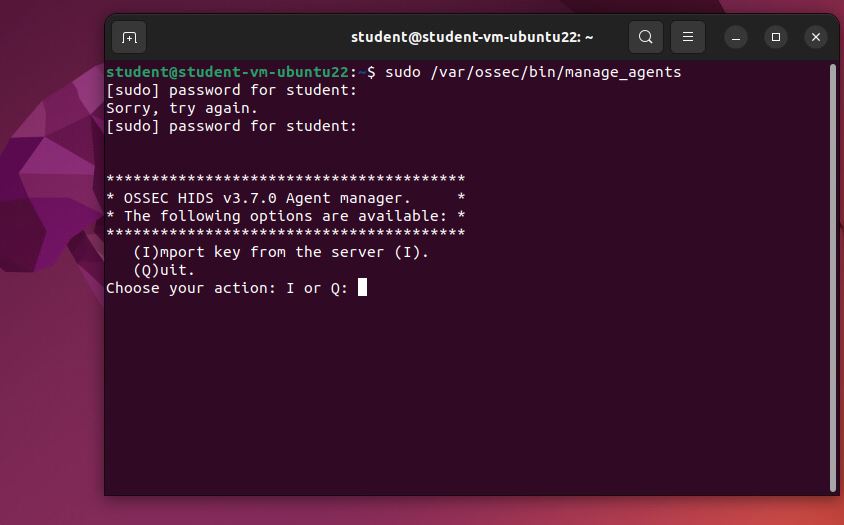
I just changed the version number since there is a newer version for ossec than 3.6.0.

I installed ossec on the server with the wget comman. Then, I unzipped it then I installed it like before. I made sure to choose server installation. On the other hand, I chose agent installation for ubuntu. I added the Ubuntu agent to the server through this command:

Sudo /var/ossec/bin/manage\_agents

I added the client, and I got a unique key for Ubuntu client.

With the same command but on Ubuntu, I got this dashboard:



I did import the key for the client and the client got added to the server.

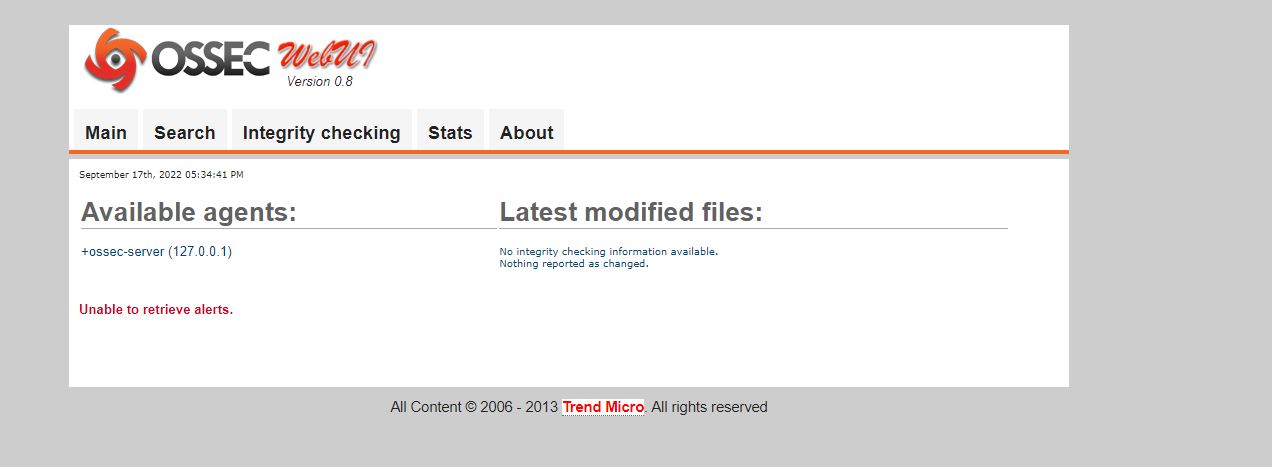
However, after doing all the steps I still have the same problem which is empty ossec dashboard.

I am starting to think that the problem is on the server. Therefore, I will redo the steps but on a new server.

#### Try 3:

I launched a new server and installed ossec with server installation on it.

I did the same exact steps as before and it worked this time. The ossec dashboard is no longer empty.



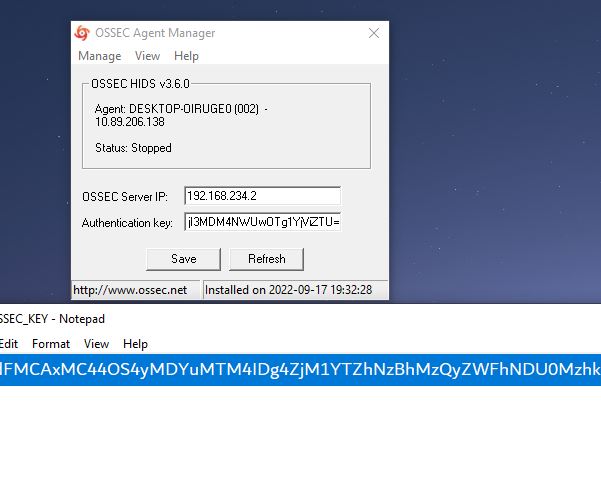
The problem was with the server.

Now let us add some agents:

I installed OSSEC agent for windows through this link:

<https://updates.atomicorp.com/channels/atomic/windows/ossec-agent-win32-3.6.0-12032.exe>

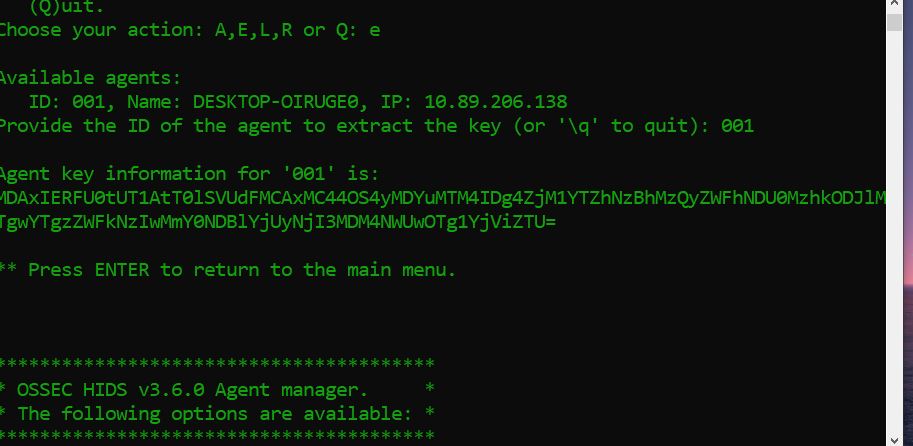
I opened it as administrator, and this is the user interface.



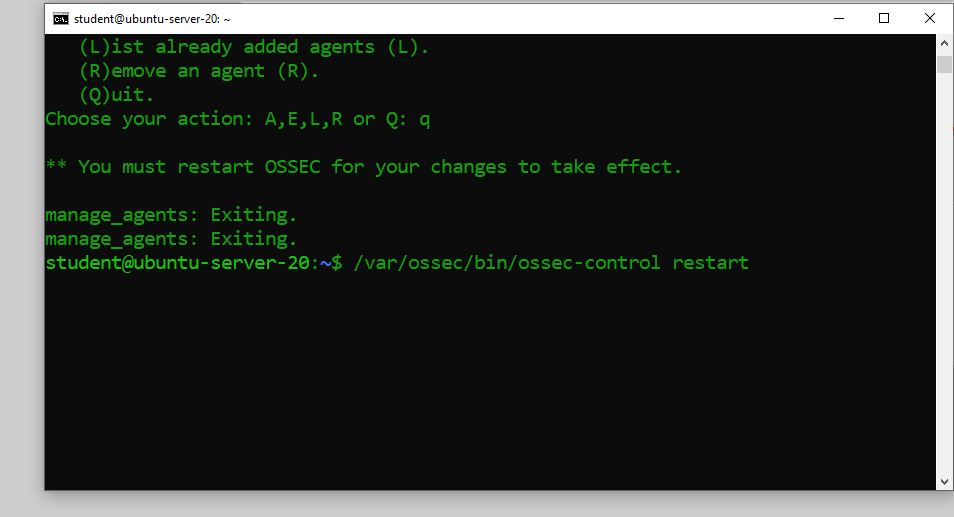
There are 2 inputs:

One for the ip address of the server which was: 192.168.234.2

And input for authentication key for the windows client.



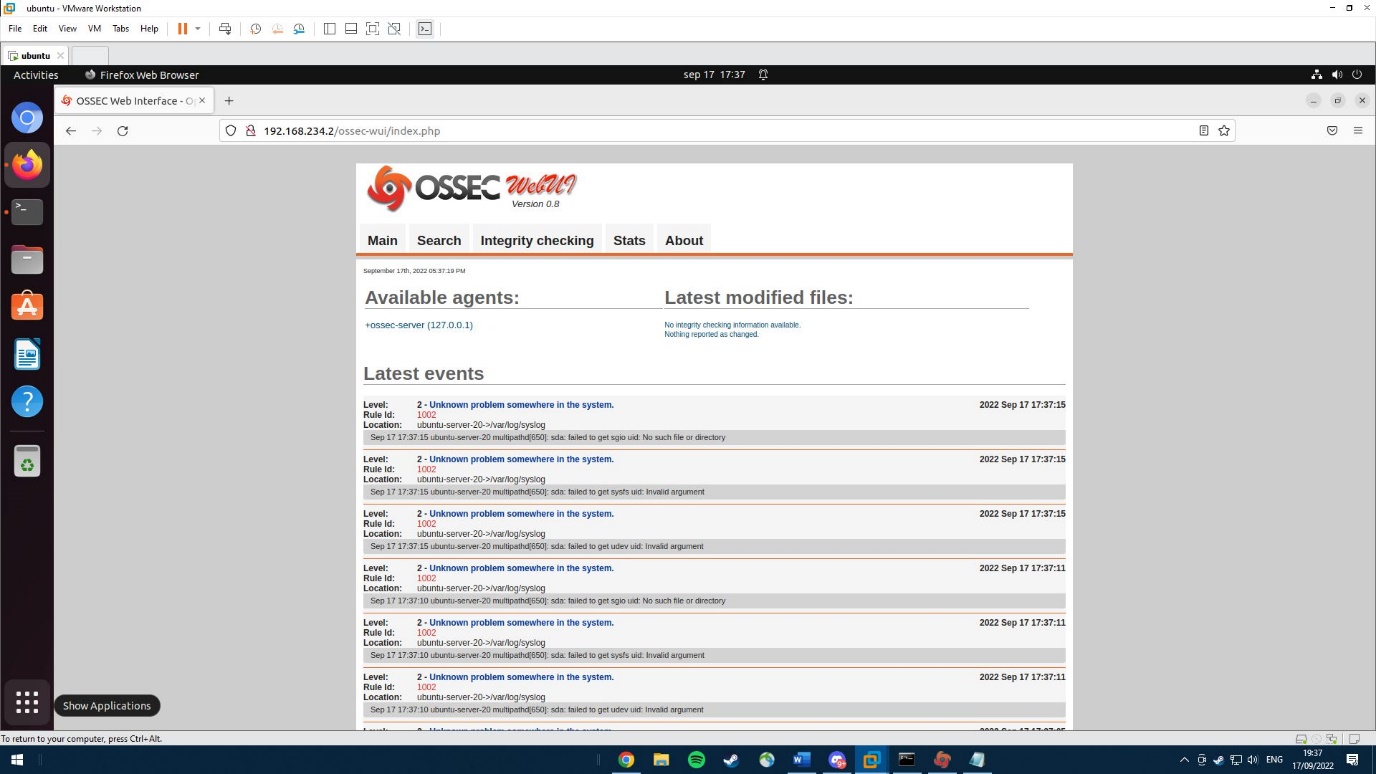
This is the key for the windows client.



I am using the server terminal on the cmd to be able to copy the key as on the server I would not be able to copy, I used this command to connect to the server terminal:

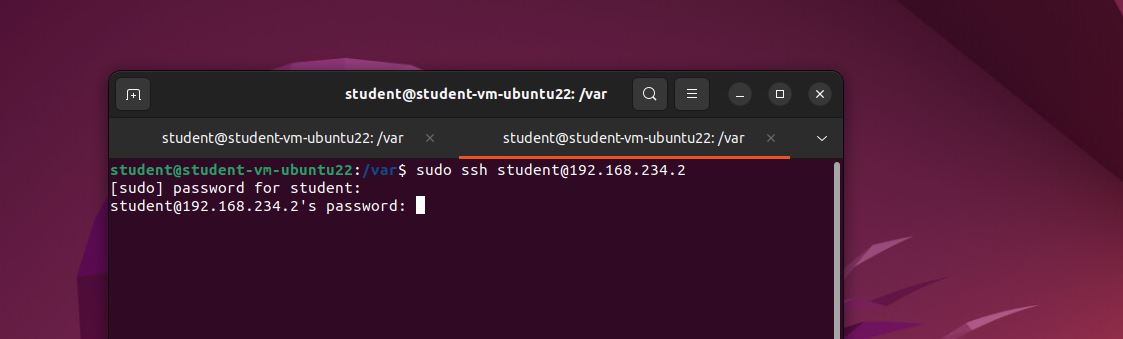
ssh [student@192.168.234.2](mailto:student@192.168.234.2)

Now that I added the client, let us take a look at the OSSEC-wui

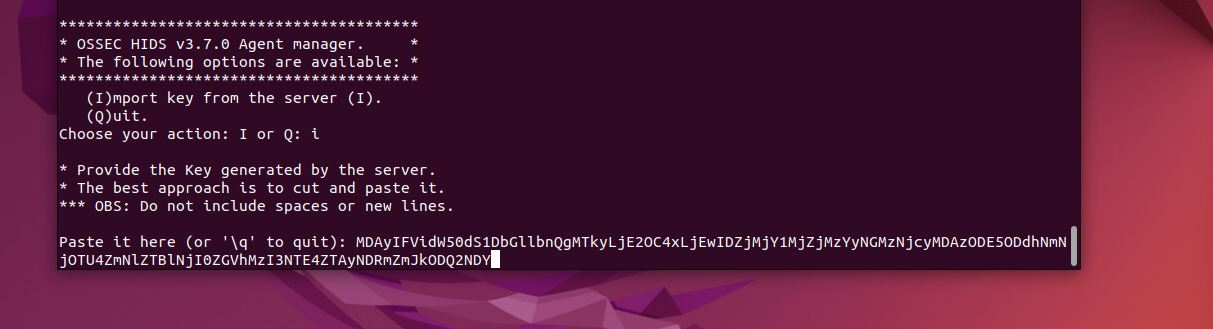
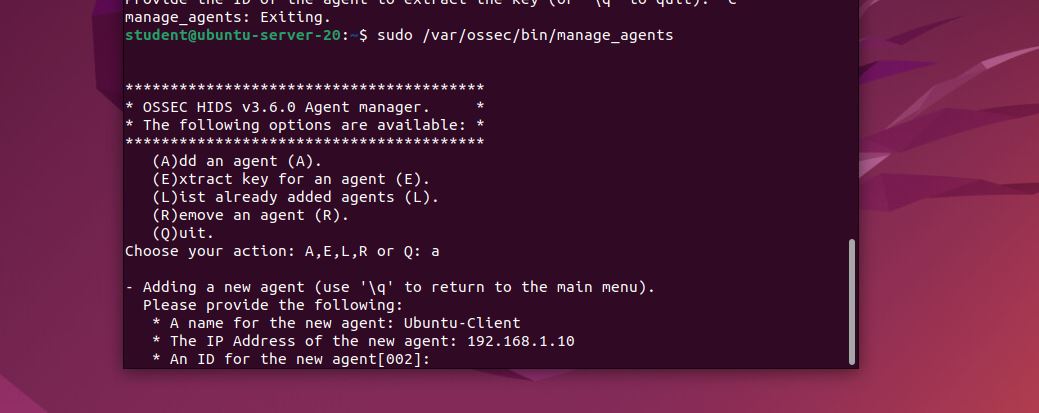


As you can see the dashboard is now full of information about the client.

I only added one client which is my windows PC, let me add another client which is the Ubuntu desktop as well.



First ssh to add the client and copy the key.

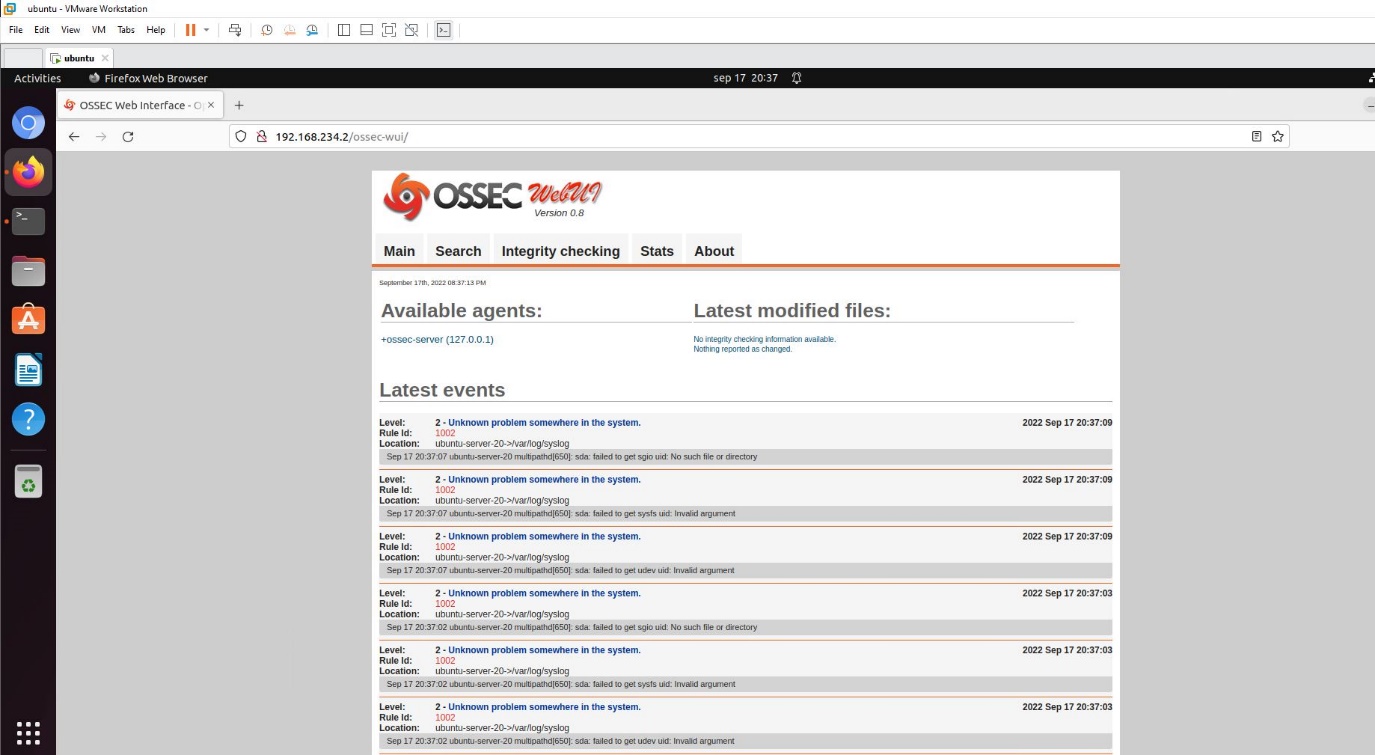


Now restart the server with:

Sudo systemctl restart apache2

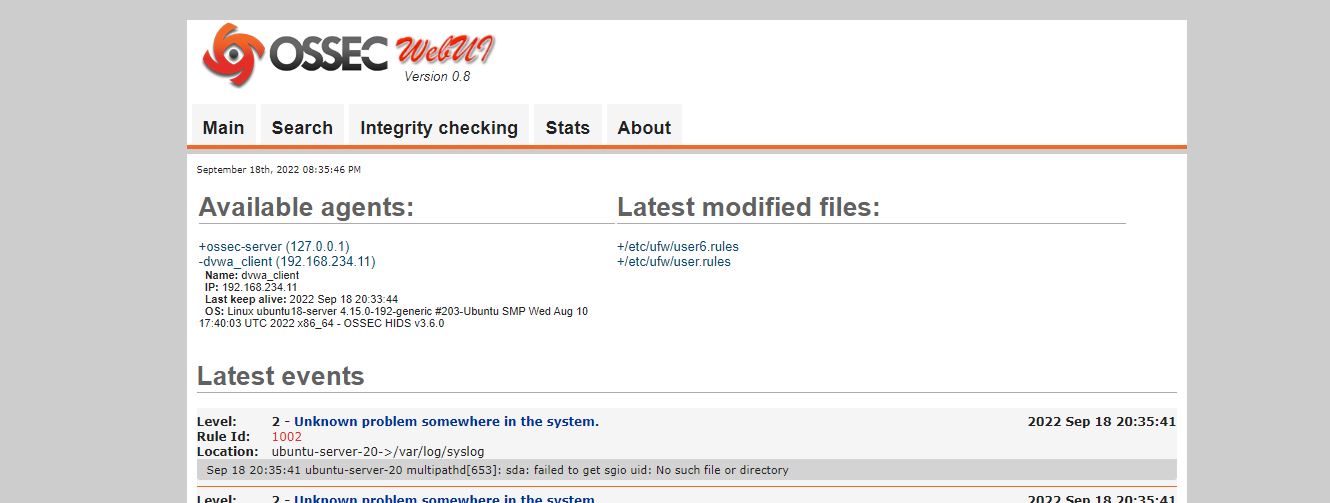
Sudo /var/ossec/bin/ossec-control restart

Finally, let us look at the web interface:

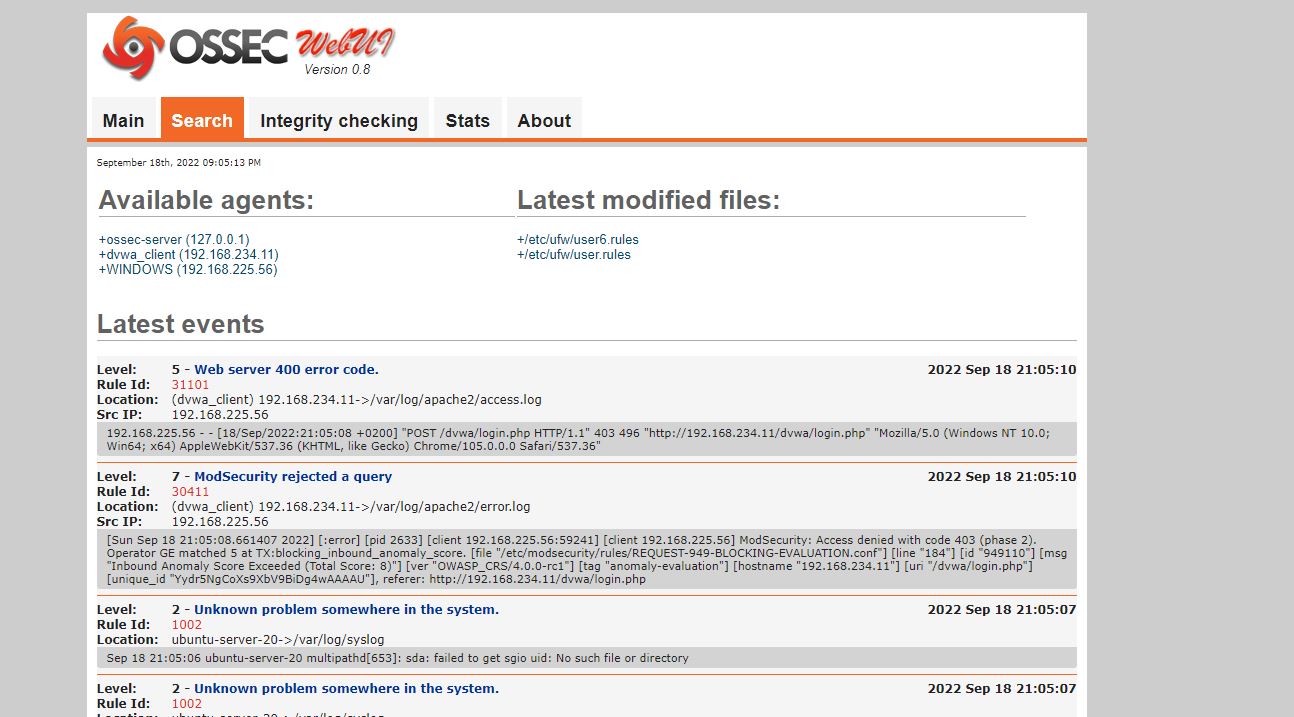


I want to add another client which is the server that I use for dvwa.

With the same previous steps, I added the server as a clients and got added to the dashboard as you can see:



Now I can see all the events that happen to the server. For instance, I tried to do sql injection on the dvwa log in and got this event:



It says that modSecurity rejected a query which is the purpose of HIDS.

## Lesson 3.2 (Web Application Firewalls)

Web Application Firewall.

I will deploy a WAF (mod security) to a dvwa server, then I will go to dvwa website to test whether the mod security is working or not.

After updating & upgrading the system. I need to install ModSecurity package using this command:

$ sudo apt install libapache2-mod-security2

Then I need to enable the module:

$ sudo a2enmod security2

To check whether I messed up the server or not, I will check localhost/dvwa between the steps to make sure the server is running without problems.

I can fine the package that I just installed in /etc folder.

Let me go there:

I need to copy the file modsecurity.conf-recommended and paste it on the same folder but without the word “recommended”.

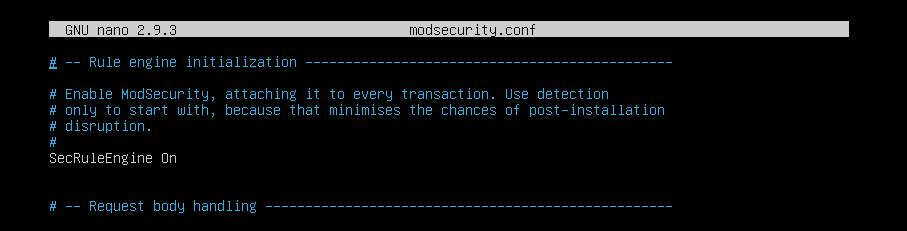
I do this to be able to apply some changes to the new file that I will use instead of the recommended one.

This command would copy it:

$ sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf

Now I can nano to that file and change its code.

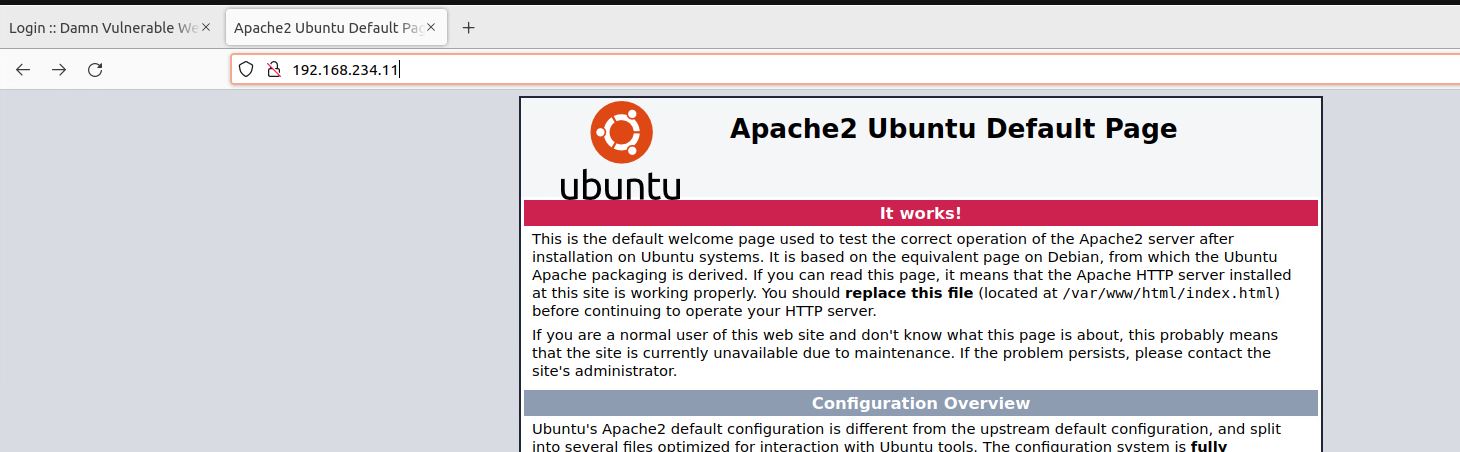
$ sudo nano /etc/modsecurity/modsecurity.conf



Save and quit.

Restart the server and check dvwa:

$ sudo systemctl restart apache2



Server works fine.

Now I need to install WAF rules for the server from a git repository.

$ git clone <https://github.com/coreruleset/coreruleset.git>

Then go inside the folder and rename crs-setup.conf.example to crs-setup.conf

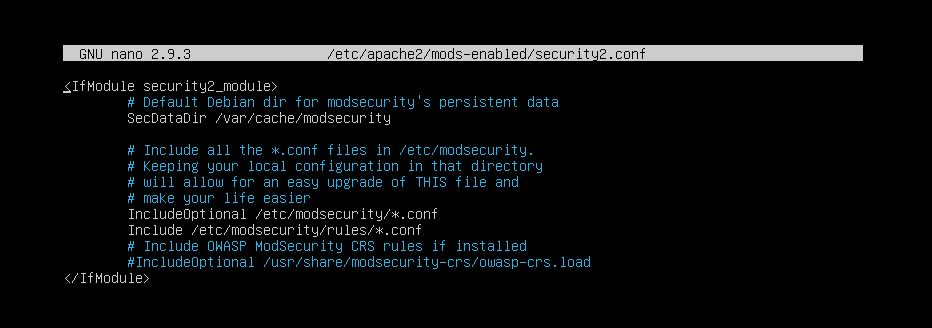
$ sudo mv crs-setup.conf.example /etc/modsecurity/crs-setup.conf

Also move the rules folder to /etc/modsecurity

$ sudo mv rules/ /etc/modsecurity/

Now I need to edit the security2.conf file to add the rules folder rule to it:

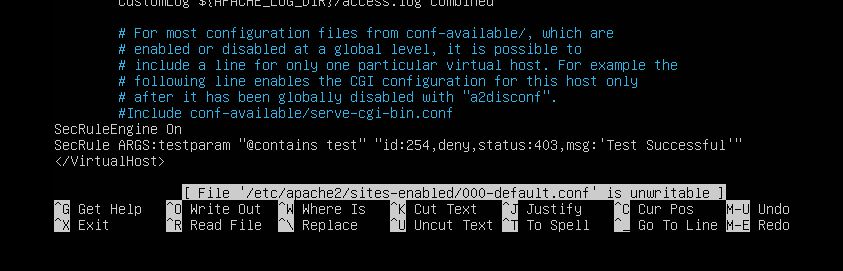
$ sudo nano /etc/apache2/mods-enabled/security2.conf



I added the second Include statement to add the rules that I just cloned from github.

Finally, I just need to make sure that ModSecurity can detect and block any suspicious activity. To achieve this, I need to add 2 lines of code to a file:

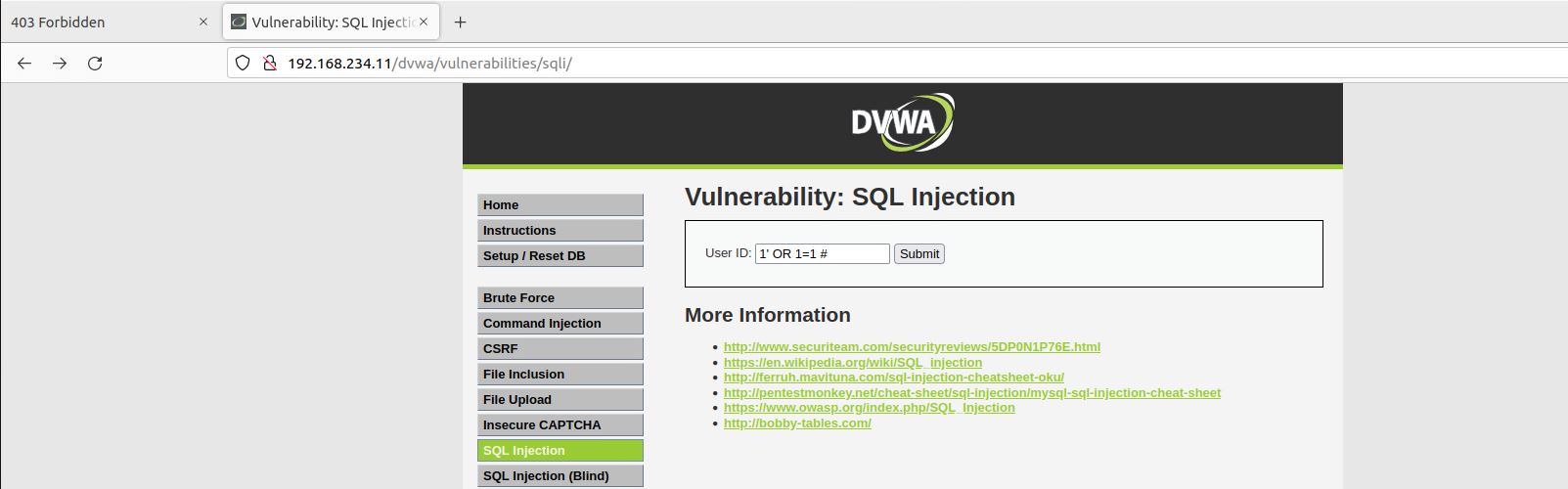
$ sudo nano /etc/apache2/sites-available/000-default.conf



Now everything is set up. I should restart the server:

$ sudo systemctl restart apache2

And go to dvwa to check whether the rules are working or not.

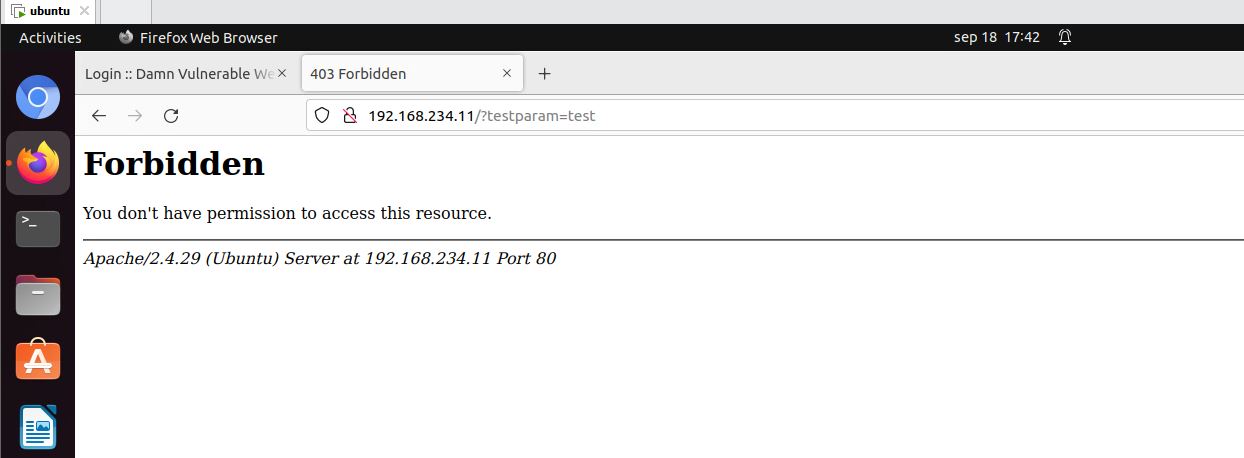
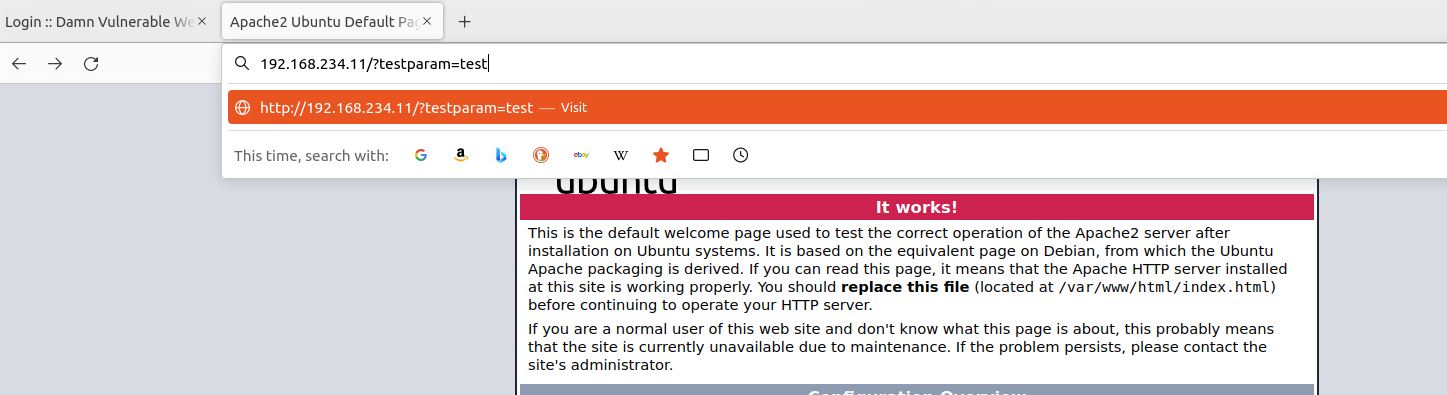


This line of code would return everything in the database. However, we just applied rules that prevent these tricks from happening. Therefore, let me try to run this command to see what would happen:



As you can see, the rules are working properly and prevented sql injection statement from running.

Let me try another trick which is adding parameters at the end of the url:



This trick also is being stopped.

# DVWA (Damn Vulnerable Web Application)

## Path Traversal

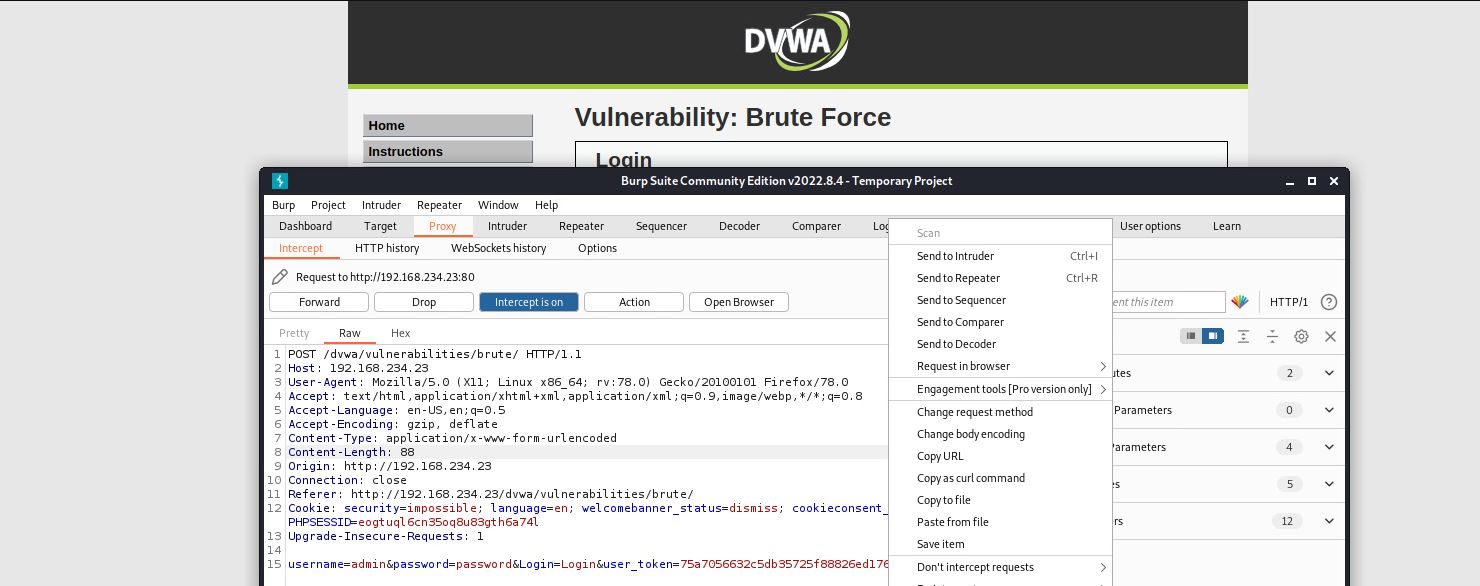
Moving to the real pen testing, we started the second week with a class for path traversal. It is when you study what is a specific system and then be able to know the file names and where sensitive data might be stored, you can navigate to the system files until you reach some private data.

Also, you could get to trick the terminal that you are someone else and execute commands on behalf of them.

## Brute Force

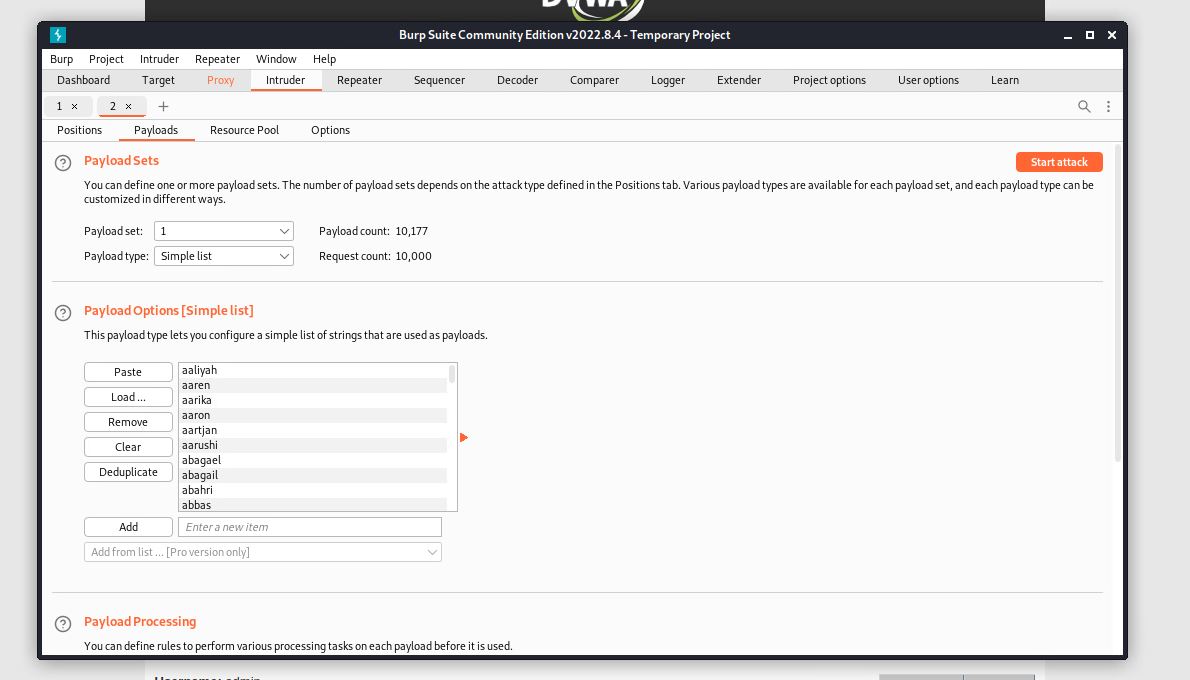
Brute force is when the attacker guesses the username/password to log into places they do not have access for. The way I would do this is by sending the POST request to Burp Suite to adjust the inputs as I desire. Burp Suite has many methods that have several purposes. For instance, Intruder is to specify a payload that you want to try so many inputs in that payload, and when burp suite returns a 200 status for the input (what the attacker guessed), means that the input was successful. Other way to know that the input is successful is by looking at the length as the correct input differs from the others.

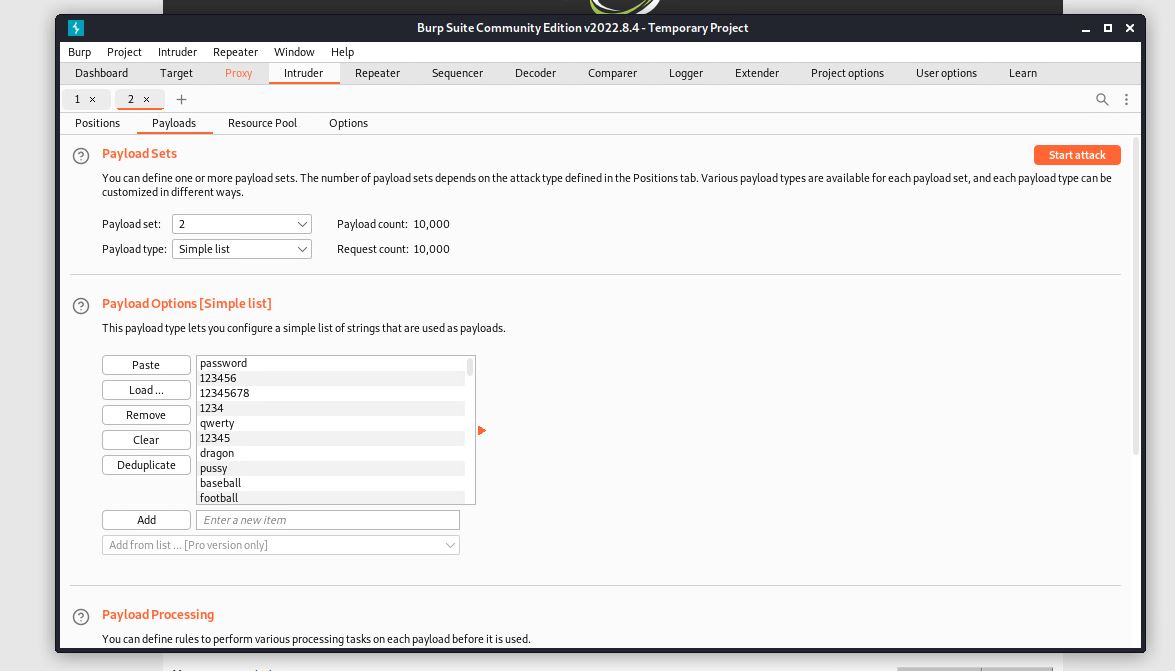
There are hundreds of username/password files that are available to make it easier for the pen tester to use instead of having to type the common username/passwords, you can just use these wordlists(/usr/share/secLists/), or you can find many other files on the internet.

Let’s get back to brute forcing, when capturing the request on Burp Suite, we should right-click and “Send to Intruder”. 

Then navigate to Intruder tab to specify the payloads that you want to test. For example, $username$(payload 1) and $password$(payload 2).

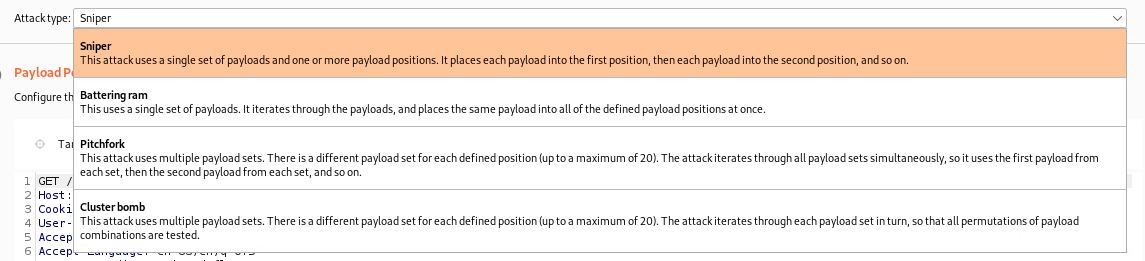
Start the attack and when there is a correct payload you would know by comparing to other inputs.





There are many Intruder methods. Such as, Sniper, Battering ram, Pitchfork, Cluster bomb.

You can choose one of them considering you situation.



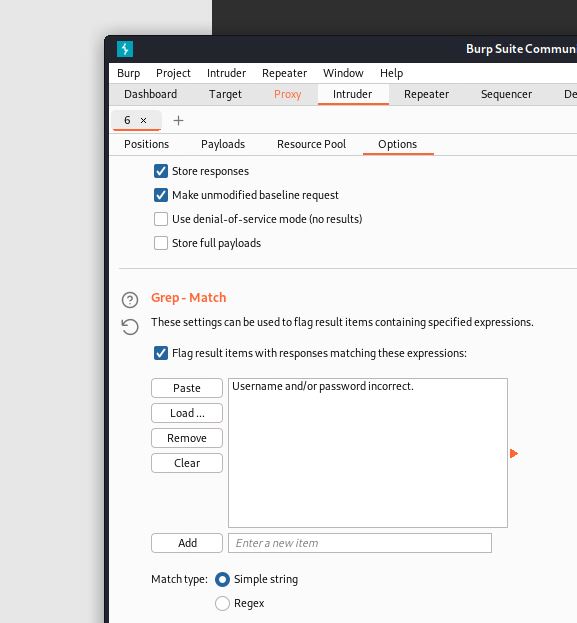
DVWA Exercise

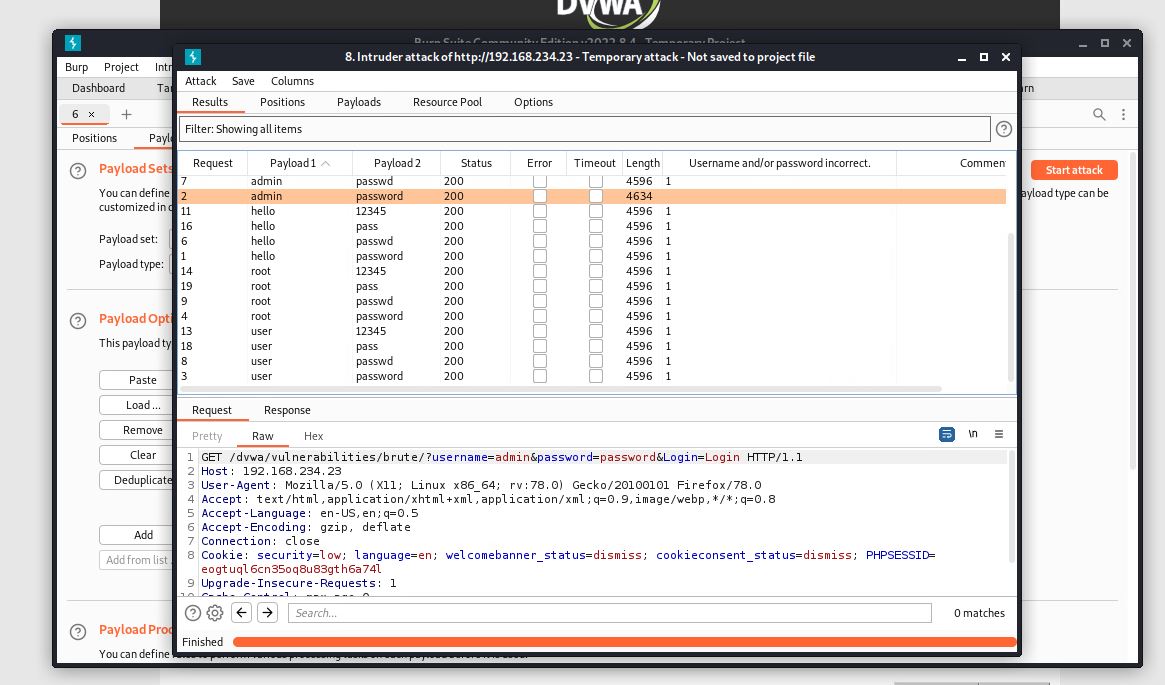
### Brute Force – Low level

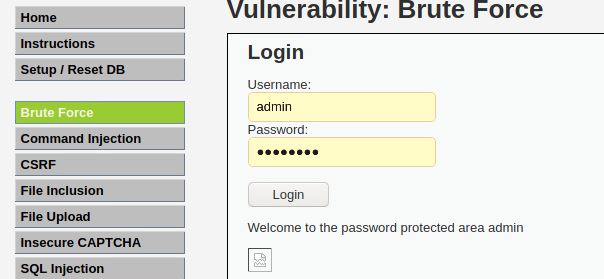
The correct answer is:

Username: admin

Password: password

You can notice it is the one that differs from others. I decided to filter the outputs as if they cause error and the correct answer is the only option that does not. 



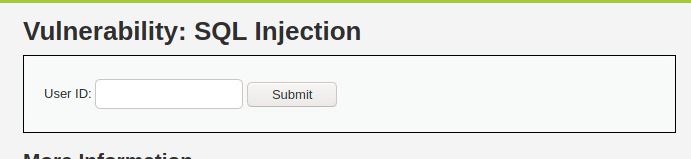


## SQL injection

This method attacks the database by tricking them to returning sensitive data. Websites have become more interactive than before. Therefore, the user is expected to manipulate the programming language code. For example, when the user is typing something in a web input, the website is ready to take that value to execute it to the website. This is a chance for the attacker to input malicious code that might affect the website in a bad way. One of the SQLi tricks is to force the database to return all the information by putting “1=1” in the sql query. 1=1 means that the query is always true, so it returns all the data.

DVWA Exercise:

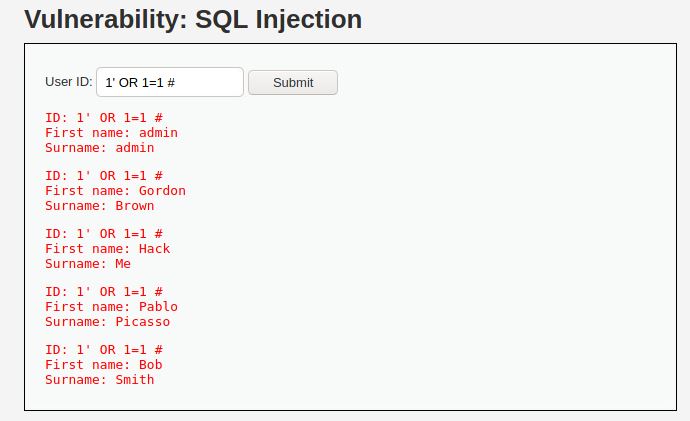
### SQLi – Low level



The website here expects a number as input to return data that are true to that specific number. However, we can manipulate the query but first we need to understand it.



Simple query that we can easily adjust by entering a number as it asks, then close the quotation mark then type “OR 1=1” as it would return everything as true. Then finish the query with # to comment anything comes after.



Indeed the database returned all the data because of the 1=1 logic.

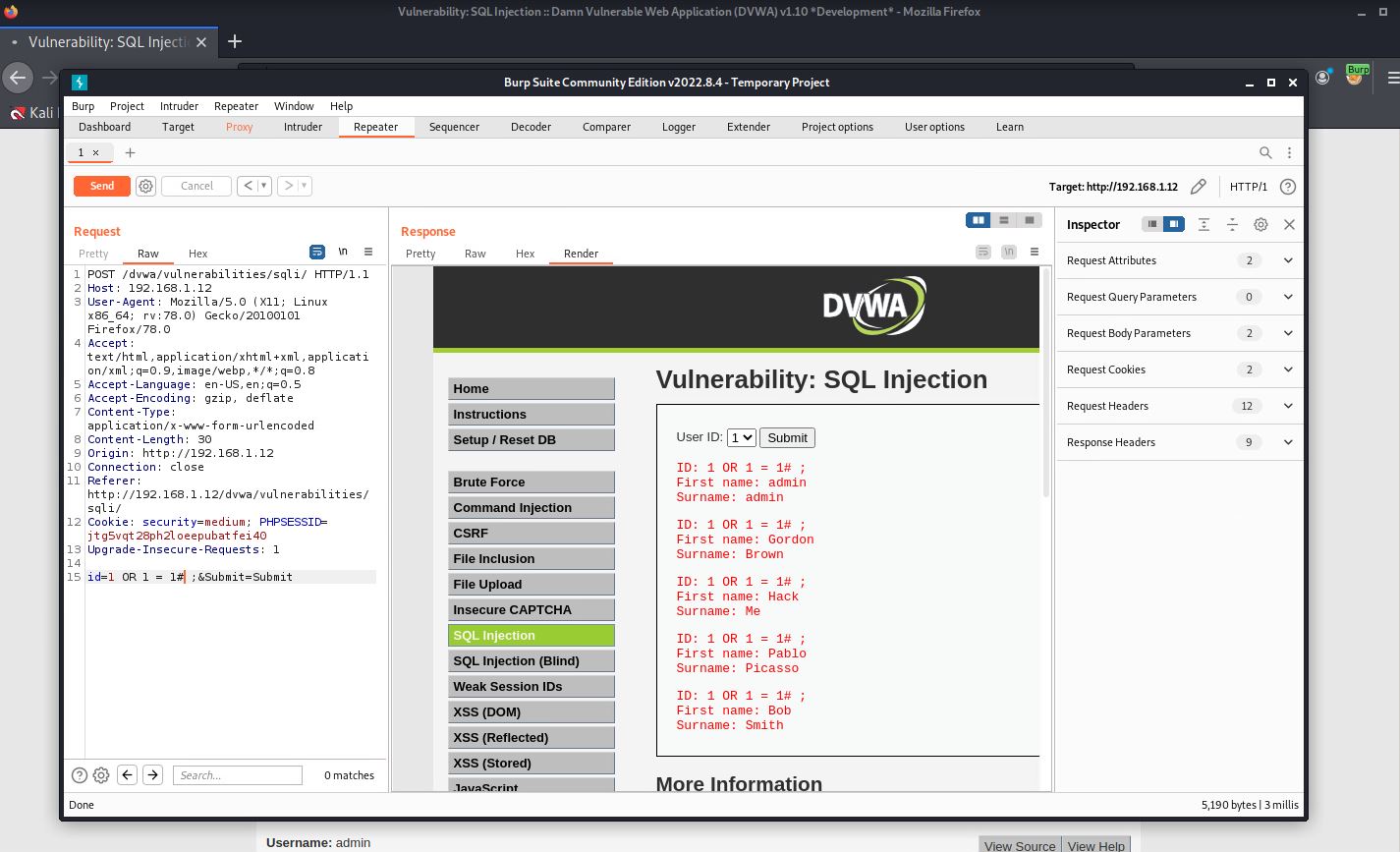
### SQLi – Medium level



The difference here is that we are not given input to enter true logic with the input. We are just asked to select the number from a drop down. The idea for how to do this is still exactly like the old one. However, we just need to find a different way to put the logic onto the database.

For this I would choose Burp Suite to help me manipulate the POST request.

I just need to send the POST request to Burp Suite then send it to repeater to execute my logic there.



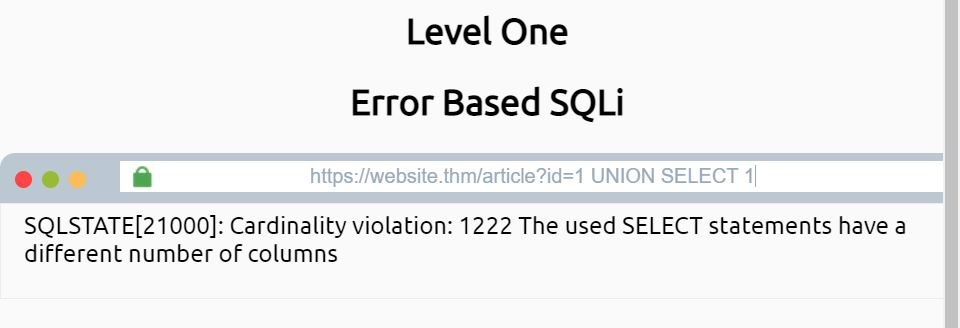
As you can see it is like the previous one, we only had to find a way to manipulate the POST request.

### SQLi – Online Challenges

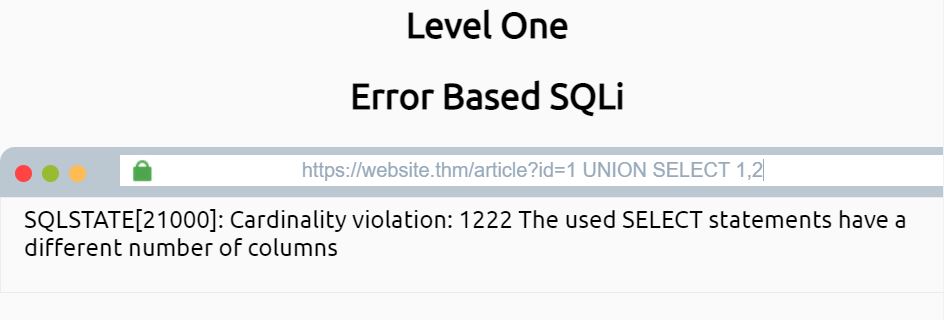
We can easily retrieve data from the database by just entering an expected number to the database. We can retrieve more data using the key word UNION.

How UNION works?

Union lets you trick the database by entering a number as expected, then after the number you type UNION to tell the database that you are not finished, and you demand more data.



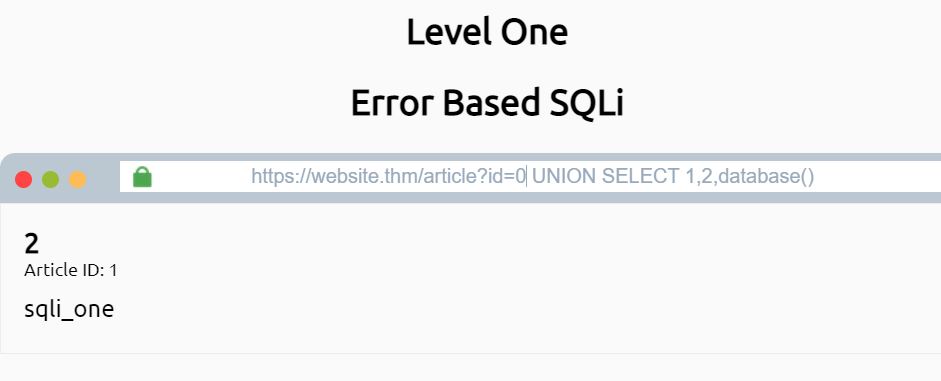
Getting error in SQLi is the first step towards breaking the database. The attacker should aim for getting error before trying to exploit. This error tells us that the second database has different columns than we entered. Therefore, let us try adding one column:



We still get the same error. Let us add one more column:

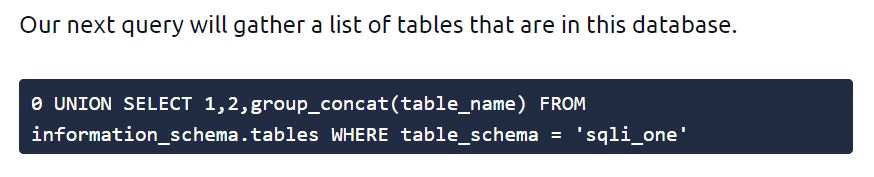


It worked. The second table has 3 columns. Let us now get the name of the database:

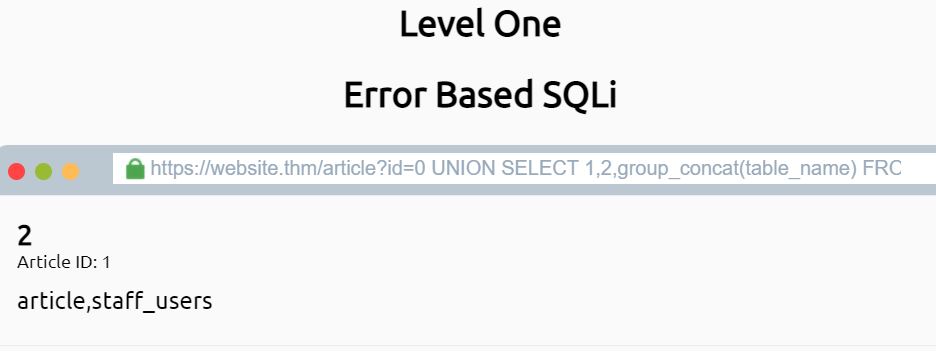


We replaced a column with “database()” to get the name of the database. Also you can notice that we changed the id value from 1 to 0, this is because we do not care about the return of the first part of the query, we care about the return of the UNUION SELECT statement.

Now we can ask for all the tables in that database.



And it returns:



This tells us that we have 2 tables: article and staff\_users, and we can access them using group\_concat().

admin123' UNION SELECT 1,2,3 where database() like 'D%';--

This query helps us to guess the database name. We keep entering values in the LIKE statement, as long as we get true as a respond, that means that the input letter/number is true. We keep guessing every letter until we get the whole name.

What is step 2?

admin123' UNION SELECT 1,2,3 FROM information\_schema.tables WHERE table\_schema = 'DATABASE\_NAME' and table\_name like 'u%';--

We repeat the same thing but for the table name this time.

Step 3:

admin123' UNION SELECT 1,2,3 FROM information\_schema.COLUMNS WHERE TABLE\_SCHEMA='DATABASE\_NAME' and TABLE\_NAME='users' and COLUMN\_NAME like 'a%';

We keep guessing the column name. After getting all the column names, we can retrieve the data inside that column using:

admin123' UNION SELECT 1,2,3 from users where username like 'a%

admin123' UNION SELECT 1,2,3 from users where password like '123%

OR

admin123' UNION SELECT 1,2,3 from users where username='admin' and password like '123%

With this technique, we can get to the username and the password then log in.

#### SQLi - TIME BASED

It has the same idea as the previous method. When the entered statement is true, the response should be delayed for some seconds.

admin123' UNION SELECT SLEEP(5),2;--

If the database has 2 columns(TRUE STATEMENT) the response should be delayed 5 seconds.

Then we can proceed:

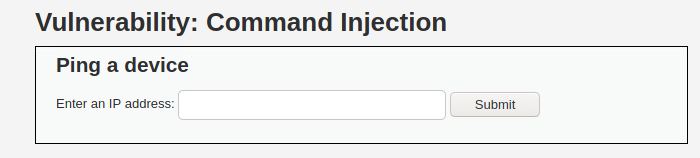
=admin123' UNION SELECT SLEEP(5),2 where database() like 'u%';--

(WE CAN USE BurpSuite to save time)

## Command Injection

### Command Injection – Low level

We are presented with an input to enter any ip address to PING.

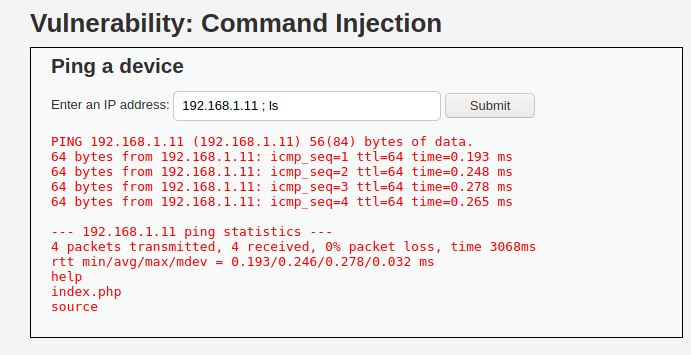


Let us see the code behind this function:



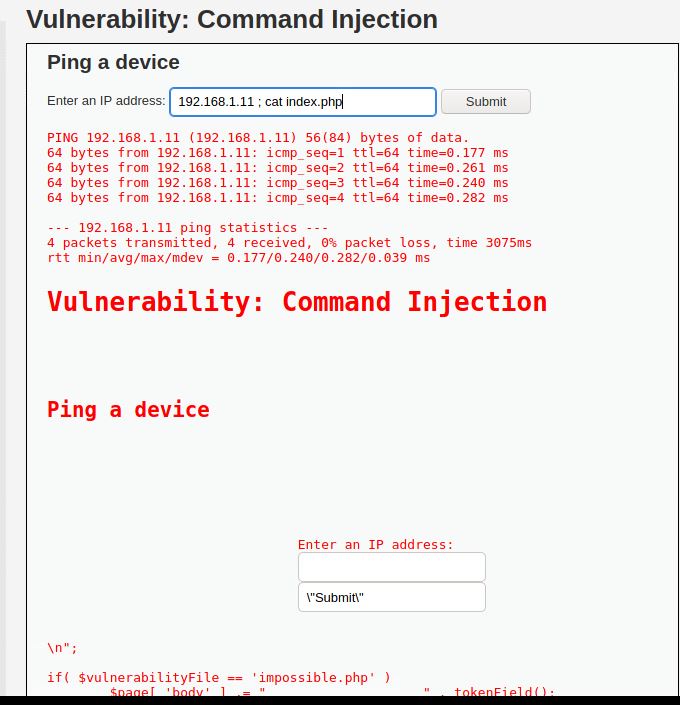
This function basically tells the program to go to the cmd and ping the target (entered ip).

It should be too easy to tell the input to do more commands. For example, we can enter any ip normally then we can finish the command by putting ; in the end. After that, we can enter more commands to get more details like (ls, pwd, etc).



As we ping the ip, we also managed to execute ls command which outputs files and folders in the system (help, index.php, source)

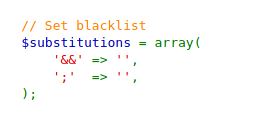
Let us see the content of index.php for example:



### Command Injection – Medium level

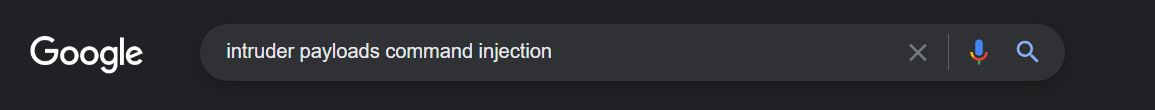
When trying to do the previous method here you would not get any response.

Why? Looking at the code:



We see that some symbols are not allowed anymore. Good but not good enough as the attacker can easily to adjust the POST request in Burp Suite to look for the allowed symbols.

First I went to google.com and looked up:

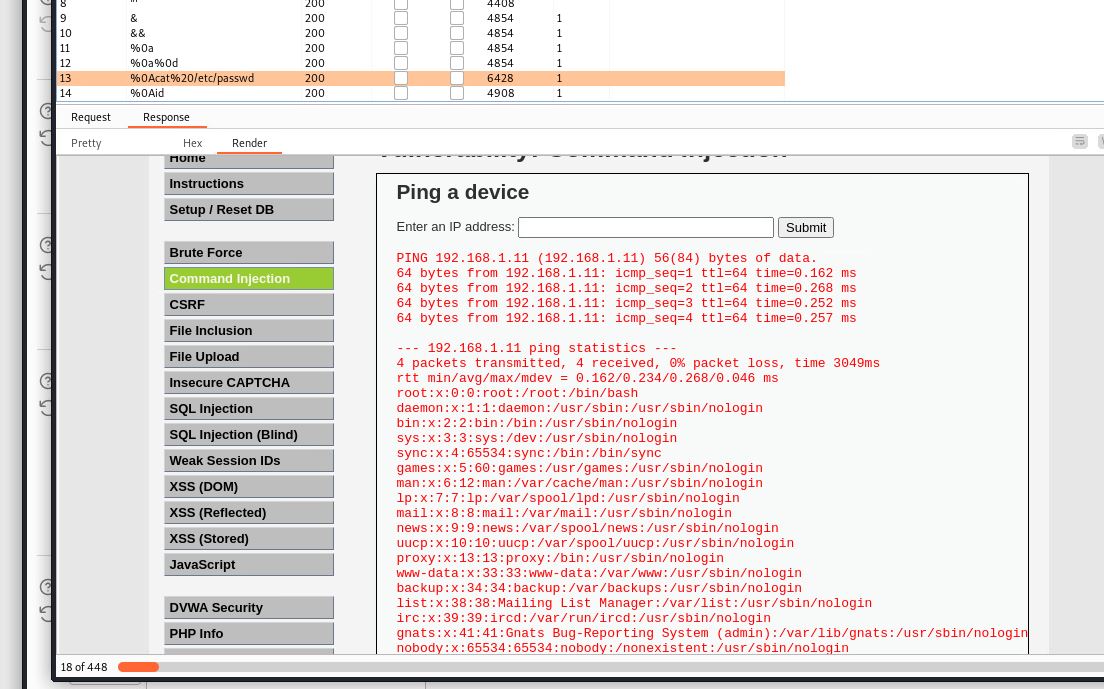


I found this page

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Command%20Injection/Intruder/command_exec.txt>

This github page has so many command injection payloads that I can try for the exercise.

I sent the request to Intruder. Sniper method helped me to load the words list and I filtered the output to packet loss as I would know the right answer If the packet loss check box was checked.



Eventually, I got to output the passwords file.

## CSRF (4.1)

A Cross Site Request Forgery is forcing the users to do things without their knowledge. For instance, they could click on normal website and their cookie was suddenly stolen. Or being asked to log in to Facebook but the attacker is receiving the information on beEF.

### CSRF – Low Level

The attacker can trick the user to click on a link that has a parameter like:

password\_new=hacker&password\_conf=hacker&Change=Change

Then the attacker can fake a confirmation page where behind the scenes they forced the user to change the password to one only the attacker knows.

## XSS DOM (4.1)

The Cross-Site Scripting in a method to inject JavaScript codes onto the user’s website.

It happens when the attacker modifies the DOM variable in the victim’s browser.

### XSS – Low Level

The website asks us to choose a language. After submitting, we notice that the URL look like this:

Graphical user interface

Description automatically generated

And the form looks like this:

Text, chat or text message

Description automatically generated

We can inject a script tag between the option tag, and it would be executed. How to inject the js code inside that tag?

Immediately after the selected language in the URL, we can add

<script>alert(document.cookie)</script>

This code would be executed since the website took it without any security.

We can do other things between the script tag, it depends on the attacker and the purpose of the attack.

### XSS – Medium Level

When trying the previous method here, it does not work. Why?

The developer knew about our trick and decided to put the word ‘script’ on the blacklist.

What can we do?

We can try one of these methods

<sscriptcript>alert(‘xss’)</sscriptcript>

<img src=a onerror=alert(‘xss’)>

The first one did not work. Neither the second one. However, the second one did not work because an image tag cannot be inside another tag. We should close the option & the select tags first, then we can add the image tag.

It would look like this:

</option></select><img src=a onerror=alert(‘xss’)>

Graphical user interface, website

Description automatically generated

### XSS – High Level

Text

Description automatically generated

The PHP code is written to secure the website. We can force the website to take our code through hiding the js code from the server.

We should not allow the website to send our script tag to the server because that would reject our request. We should only send the selected language to the server and after the word we could add # to prevent the website to send what comes after to the server.

Graphical user interface, website

Description automatically generated

## XSS Stored & Reflected (4.1)

XSS Stored is injecting code to the database that will remain in there permanently. The website developer would need to reset the database or remove the code manually to clean the website from suspicious activities.

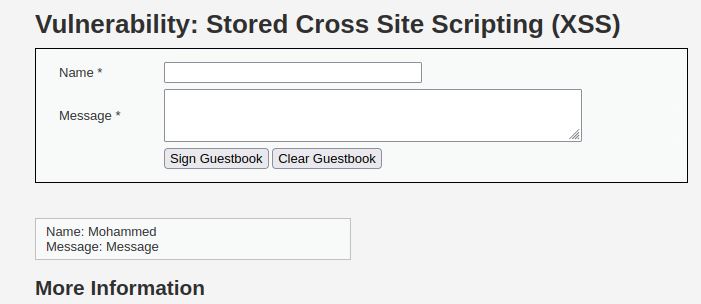
### XSS Stored & Reflected – Low Level

The objective is to redirect the users to another page immediately after loading the website.

For redirecting, I could use a js function which is window.location() to direct the user to another page.

I also need to find a way to force the code to stay in the database permanently or until someone manually decides to remove it.

DVWA offers me 2 inputs that I can POST:



The good thing that the messages load as the same time with the actual website. Meaning the messages will load permanently just like we needed.

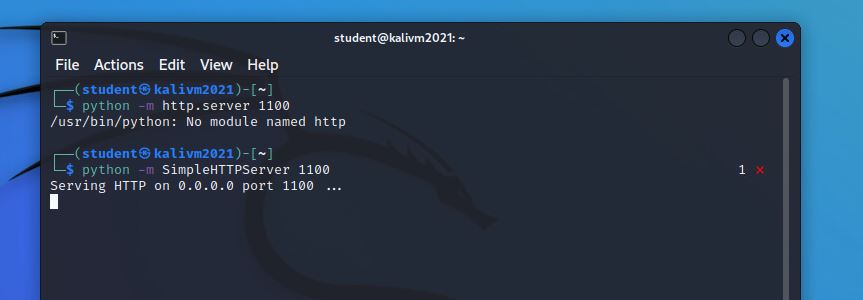
For instance, in the Message field, if I added the code:

<script>alert(‘Hello World’)</script>

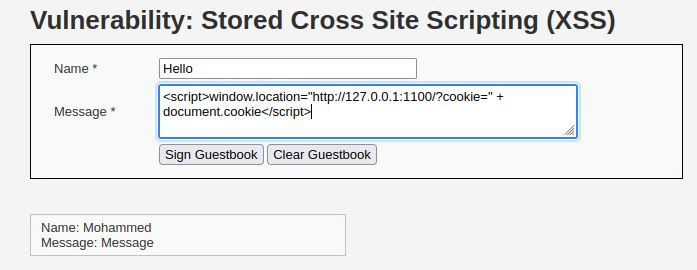
Every user that visits that website would get an alert says “Hello World” After the alert they can proceed to the website. The process would take place every time someone visits the website without any differences.

Anyways, the objective here is to redirect the user to different page.

I do not have any website on mind so I will just create a server that listens to a specific port.

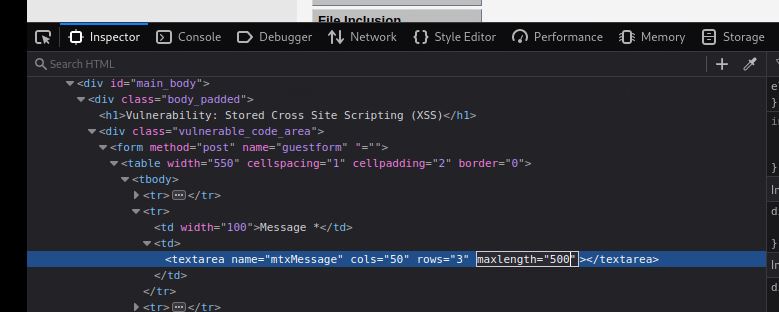


Now I can redirect the users to here with their cookies using this code.



Now I will get the cookies for all the users that visit the website.

**Note**: I had to adjust the maxlength for the Message input because it only allowed 50 characters.



**What did I just do?**

I just ruined the XSS (Stored) page, and I cannot continue working on the other tasks. Why?

Because I every time I click on that page it redirects me to another link. How can I fix that?

No idea yet.

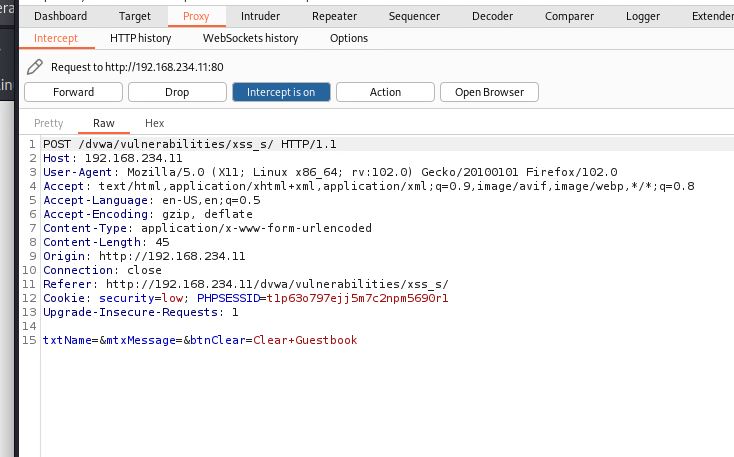
**Fixed**

I launched another DVWA server and could have forget about the broken DVWA and continue with the new one. Fortunately, I really wanted to solve that problem.

Here is what I did:

I launched a new DVWA server and went to XSS Stored and I clicked on “Clear Guestbook”.

I sent this POST request to Burp Suite and it looks like this:



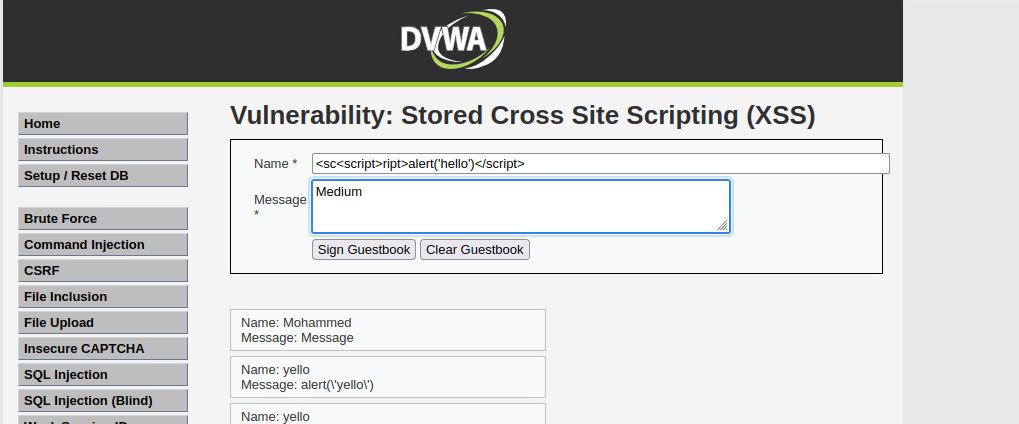
If I forward this, it will just delete all the messages for the new DVWA. However, that was not the point. I want to send the exact same POST request but for the old DVWA.

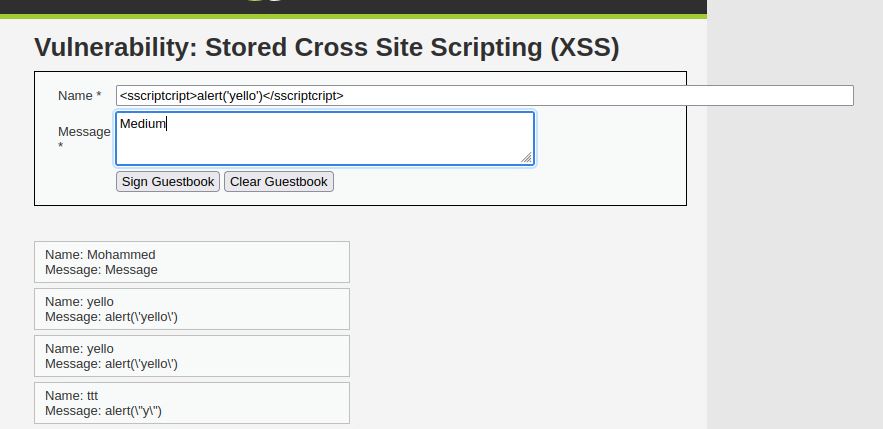
From the picture above, I replaced that ip address with the ip address of the old DVWA, and I also replaced the cookie with the old DVWA cookies, and I sent the request.

Now I can click the XSS stored without being redirected.

### XSS Stored & Reflected – Medium Level

The difference between Low level and Medium level in XSS is always the same. They just block the word “script” to prevent the attacker from injecting some js there. We can easily bypass this by duplicating the word script of putting some letter in upper case or put some space between the letters.

I tried these 2 tricks, and both worked:



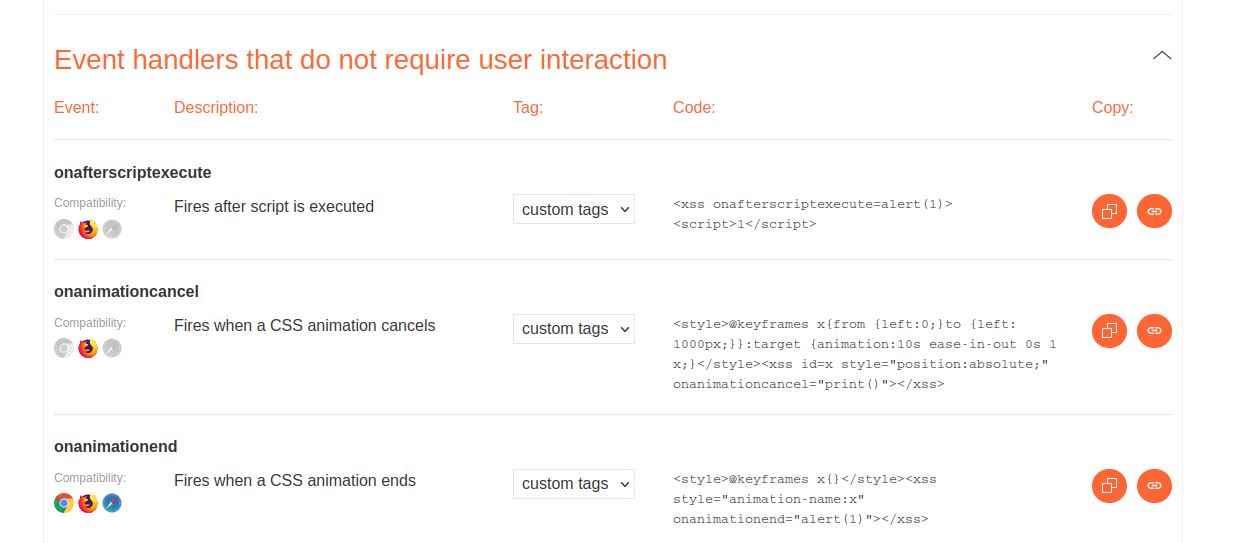
Yes, they blocked the word script but removing it would create another script tag.

### XSS Stored & Reflected – High Level

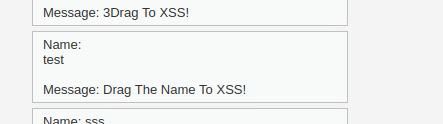
I went to this website

### <https://portswigger.net/web-security/cross-site-scripting/cheat-sheet>

and found too many XSS cheat sheet codes that most of them work.



I tried a weird one which is drag to execute and it worked.



When I drug the name, alert pops up.

# Juice Shop

Juice shop has the same purpose as DVWA. It is a website full of vulnerabilities that I can test. It has score board that indicates how far you have discovered all the vulnerabilities.

I will try to implement most of the pen testing skills that I have on it and check my score board.

Firstly, I like to test the cross-site scripting vulnerabilities. Therefore, let me look for a searching bar and execute some JavaScript code on it.

Graphical user interface, application

Description automatically generated

As you can see, I managed to find a XSS vulnerability. All I had to do was put this code on an input filed:

<img src=”xx” onError=”alert(‘xss’)”/ >

Other codes that might work:

<script>alert(‘xss’)</script>

This could work because it takes the js code and inject it inside the search function and it would look like the developer put it there on purpose.

Some developers know this trick and decide to block the word “script” from all the inputs.

The blocking process still executes the input but only deletes the word “script”. Therefore, we can do this trick which is typing the word twice:

<sscriptcript>alert(‘xss’)</sscriptcript>

Other vulnerability is being able to access the confidential document.

We notice on the page About Us, there is a highlighted text that when we hover on it, it indicates that it sends us to this link:

Text

Description automatically generated

I know that the ftp file should have more files that I care to look at. Therefore, I can copy the link and remove the last word because I do not care about that specific file.

Then a page full of files appear:

Graphical user interface, application

Description automatically generated

Now I can download and open any file I want.

Let us move to something more interesting.

I will go to the log in page, on the username input field, I can type this line to bypass the log in and be able to log in as an admin:

’ OR 1=1 #

The ‘ closes the sql query that the database expects.

The OR statement is for typing something that is true so the database return true to the website.

The # for commenting everything that comes after the true statement.

The database returns true, and I log in successfully.

# OWASP Top Ten

Open Web Application Security Project is an open-source software that aims to improve the security of web applications.

OWASP Top Ten refers to the top 10 critical security risks to web applications.

The current version of the list was updated in 2021 whereas the previous one was in 2017.

2021 added some new vulnerabilities. However, most of them remain on the list.



(Picture from <https://owasp.org/www-project-top-ten/> )

Let me talk about the top 10 security risks.

1. **Broken Access Control**

It is when the attacker gets inside the victim’s system as a privileged user. This problem sounds less dangerous than it really is. This risk deserves its rank as the top on the list, because giving the attacker all the permission to do anything might terminate the whole system.

How to help to prevent it?

* Log all the access control failures, and always alert admins.
* Disable web server directory listing.

1. **Cryptographic Failures**

It is the failures that are related to cryptography. This could leak sensitive and secret information that is being sent through the internet.

How to help to prevent it?

* Do not store any sensitive data if you do not need them.
* Encrypt all sensitive data.
* Disable caching for response with sensitive information.
* Always use authenticated encryption instead of just encryption.

1. **Injection**

Injection is when the attacker injects their malicious code into any program to force it to behave as wanted. (e.g., stealing authentication cookies, storing malicious code in the database, retrieving sensitive data from the database)

How to help to prevent it?

* Use blacklist to the inputs on the website to block any malicious words like the tag script.
* Use the LIMIT and other SQL controls within queries to not expose your sensitive data.
* Use ModSecurity on your server to block any malicious injections.

1. **Insecure Design**

Implementing the suitable design to the web applications is very important to block the way for the attacker to inject something.

How to help to prevent it?

* Use libraries of secure design patterns on your applications.
* Limit resource consumption.
* Do user testing to test the design of the application.

1. **Security Misconfiguration**

The security engineers must be aware of every device on their infrastructure and why is it there. The lack of infrastructure security could help the attacker to get into the system.

How to help to prevent it?

* Always check the configurations and setting in all the infrastructure environment.
* Keep all the systems updated.

1. **Vulnerable and Outdated Components**

When you use vulnerable operating systems, web servers, databases, you are giving the attacker the green light to get into your system. The attacker only needs one vulnerable system to compromise your infrastructure.

How to help to prevent it?

* Remove unused dependencies
* Only click on trusted links from official sources.

1. **Identification and Authentication Failures**

The passwords, cookies, and session tokens are unique for the user and should not be found by intruders.

How to help to prevent it?

* Implement multi-factor authentication.
* Implement weak password checks.
* Limit the requests coming from one ip address.

1. **Software and Data Integrity Failures**

Relates to code and poorly secured infrastructure that might allow for the intruder to get in the system. Also, it relates to servers that rely on plugins such as Wordpress.

How to help to prevent it?

* Make sure that your service only consumes trusted repositories.
* Ensure to apply security tools like OWASP to your service.

1. **Security Logging and Monitoring Failures**

Detecting and logging all the system failures is an integral part for the security. The security people need to be aware of every error that happen inside their system before the attacker gets in.

How to help to prevent it?

* Log every log in failure and wrong input in your application.
* Encode all the log data.

1. **Server-Side Request Forgery**

When the attacker adjusts the code to behave as they want. This could give them sensitive data about the user. Such as, cookies, log in credentials. Then the attacker can do anything that the user can do.

How to help to prevent it?

* Separate your networks as many as possible.
* Sanitize all the input data.
* Do not send raw responses to clients.

# (BOK) Network Security (week 5-8)

# (BOK) Security Concepts (week 9-10)

# References

* Jagatic, T. N., Johnson, N. A., Jakobsson, M., & Menczer, F. (2007, October). Social phishing. *Communications of the ACM*, *50*(10), 94–100. <https://doi.org/10.1145/1290958.1290968>
* Wallen, D. (2020, March 3). *Intrusion Detection Systems: A Deep Dive Into NIDS & HIDS*. Security Boulevard. Retrieved September 9, 2022, from <https://securityboulevard.com/2020/03/intrusion-detection-systems-a-deep-dive-into-nids-hids/>
* *What was WannaCry? | WannaCry Ransomware*. (n.d.). Malwarebytes. Retrieved September 19, 2022, from https://www.malwarebytes.com/wannacry
* Tudor, D. (2021, August 27). *What Is a Host Intrusion Detection System (HIDS) and How It Works*. Heimdal Security Blog. Retrieved September 9, 2022, from <https://heimdalsecurity.com/blog/host-intrusion-detection-system-hids/>
* Grow, H. (2021, July 7). *OSSEC Open Source HIDS with Web user interface. (updated for Ubuntu 20.04 & OSSEC 3.6.0)*. HendGrow. Retrieved September 17, 2022, from https://hendgrow.com/2020/10/01/ossec-open-source-hids-with-web-user-interface-updated-for-ubuntu-20-04-ossec-3-6-0/
* *What are Computer Viruses? | Definition & Types of Computer Viruses*. (n.d.). Fortinet. Retrieved September 18, 2022, from <https://www.fortinet.com/resources/cyberglossary/computer-virus#:%7E:text=A%20computer%20virus%20is%20a,in%20data%20loss%20and%20leakage>.
* *Cross-Site Scripting (XSS) Cheat Sheet - 2022 Edition | Web Security Academy*. (n.d.). Retrieved September 21, 2022, from https://portswigger.net/web-security/cross-site-scripting/cheat-sheet