PROJECT REPORT ON

TOP 10 OWASP ATTACKS

&

WINDOWS FIREWALL

TABLE OF CONTENTS

TABLE OF CONTENTS

- Sensitive Data Exposure
    - Introduction
    - Demonstration
    - Impacts & Prevention
- Missing Function Level Access Control
    - Introduction
    - Demonstration
    - Impacts & Prevention
- Cross Site Request Forgery (CSRF)
    - Introduction
    - Demonstration
    - Impacts & Prevention
- Using Components with Known Vulnerability
    - Introduction
    - Working
    - Impacts & Prevention
- Unvalidated Redirects and Forwards
    - Introduction
    - Demonstration
    - Impacts & Prevention

TABLE OF CONTENTS

ABOUT THIS PAPER

Today we are all part of a smart-digital world where the majority of the population are using devices that are connected to various kinds of networks most commonly the Internet. The world has changed so much that, everyone online shares their personal data on the web, also the transactions systems seen nowadays are totally based on the internet, so we need to find ways to keep ourselves secure without the need to depend on others.

This Project is based on the top ten attacks of OWASP (Open Web Application Security Project), and also about the windows firewall. It includes the detailed description about all the top ten and the common attacks mentioned in the OWASP list including its Impact on Individual and Businesses, also the associated security fixes to stop those attacks and also about the advanced windows firewall features.

This Report is developed under the complete guidance of Mr. Rahul Tyagi, Vice President, Lucideus Tech, New Delhi. I really thank him for his kind help and guidance.

INTRODUCTION OWASP TOP 10

OWASP is an Open Web Application Security Project started on September 9, 2001 by Mark Curphey. It is an Organization focused on improving the security of software. They make software security visible so that the individuals and organizations worldwide can make informed decisions about true software security risks.

THE OWASP TOP 10

OWASP top ten is a list of the 10 most dangerous security flaws/vulnerability along with the effective methods of correcting those flaws/vulnerability that any Web applications on the network/internet can face.

LIST OF TOP 10 OWASP ATTACKS

1. Injection.
2. Broken Authentication and Session Management.
3. Cross Site Scripting.
4. Insecure Direct Object References.
5. Security Misconfiguration.
6. Sensitive Data Exposure.
7. Missing Function Level Access Control.
8. Cross Site Request Forgery (CSRF).
9. Using Components with Known Vulnerabilities.
10. Unvalidated Redirects and Forwards.

1] INJECTION

What is Injection?

Injection is an attacker's attempt to breach into an application by giving commands to it such that it will change the meaning in which the commands are interpreted and then change the course of execution into an unexpected manner. It is one of the most common security risks in web applications.

Some common types of Injections are: SQL injection, OS commanding, LDAP injection, XML injection, XPath injection, SSL injection, IMAP/SMTP Injection, Buffer Overflow and Code Injection.

Definition of Injection according to OWASP:-

Injection flaws such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. Using these injections, the attacker could trick the interpreter into executing unintended commands or accessing data without proper authorization. The most common type of injection used is SQL injection, so we will be focusing on SQL injection, its types and demonstration.

Most commonly used types of SQL Injections:-

In total there are approximately 42 types of SQL Injection, these are some of the most commonly used SQL Injections:

- Union Based Injection
- Error Based Injection
- Blind Injection
- Post Parameter Injection

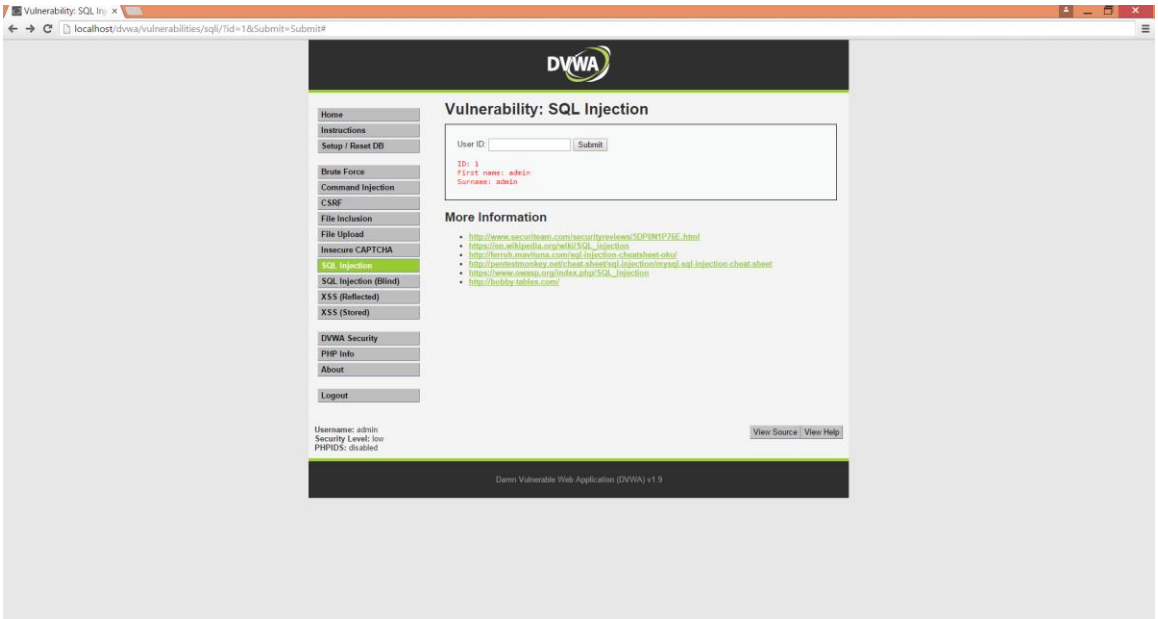
1) Union Based Injections:

Union based injection is a type of injection which is basically injected in dynamic websites based on php/mysql. It mainly targets the GET parameters used in the PHP based websites and poorly managed error reporting system of the MYSQL database. This attack is often used when the web application is not configured to show generic error messages.

Let's dig deep on how this attack works. The step by step demonstration will be shown below.

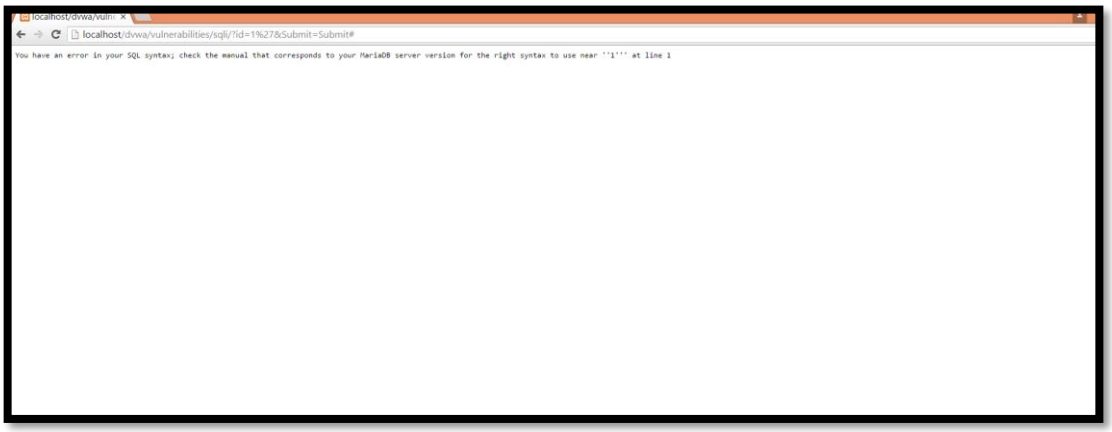
Steps for performing Union Based injection:-

1. **Find any Get Parameter in the any URL of the website:** It should look somewhat like
   Eg: http://www.examplewebsite.com/samplepage.php?id=1

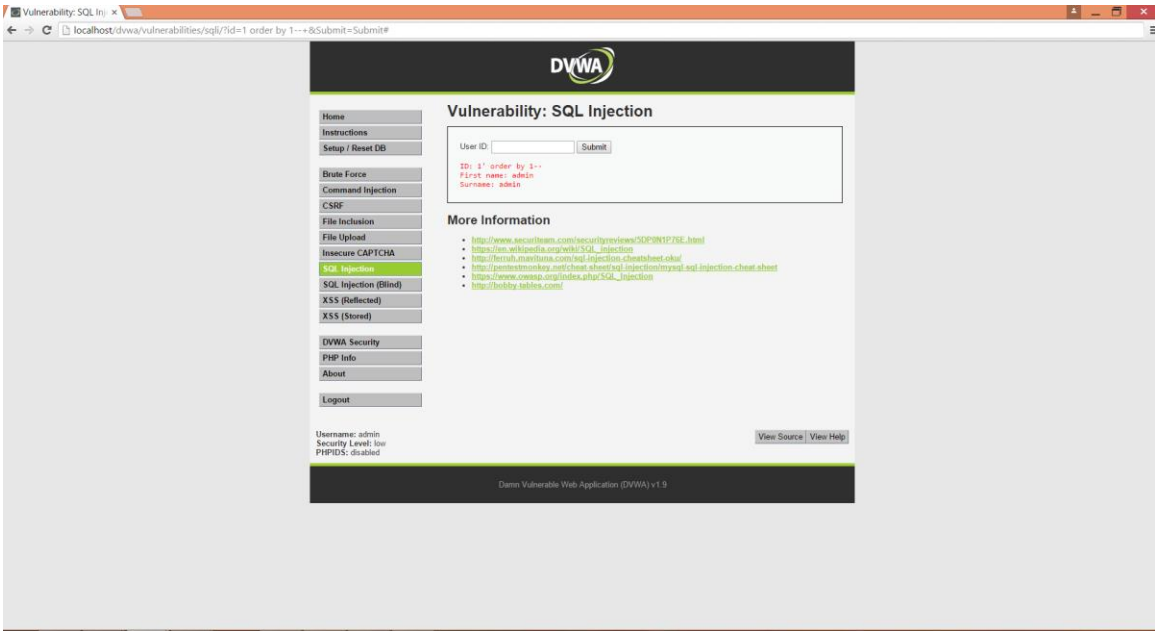


2. **Check the Error Handling of the website:** This is done by simply putting a "quote" i.e. (') at the end of the "id" parameter and press Enter. If by chance you land at any page which shows statements like: "You have an error in your sql syntax", it means that the website is vulnerable with this type of injection and now we can directly interact with the database of the website.

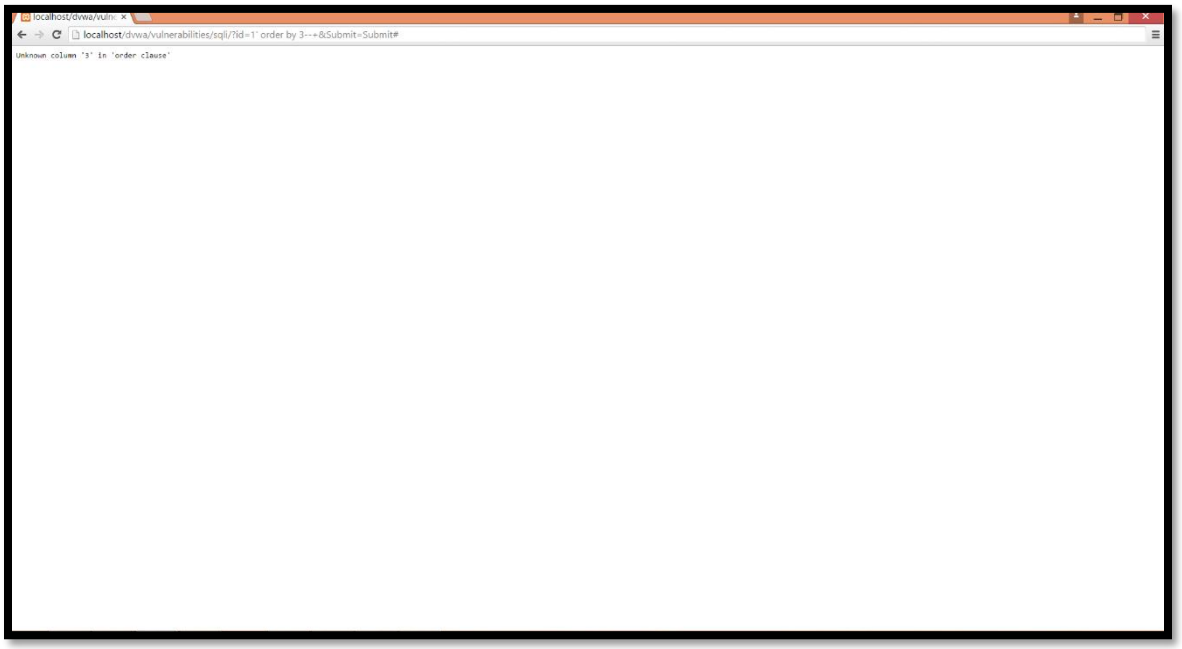
Eg: http://www.examplewebsite.com/samplepage.php?id=1'



3. **Find the total number of columns**: Now since we can interact with the database, our next task is to find the total number of columns in the respective URL's tables. To do so, add "order by 1--+" at the end of the URL you were having in the above step.
   Eg: http://www.examplewebsite.com/samplepage.php?id=1' order by 1--+

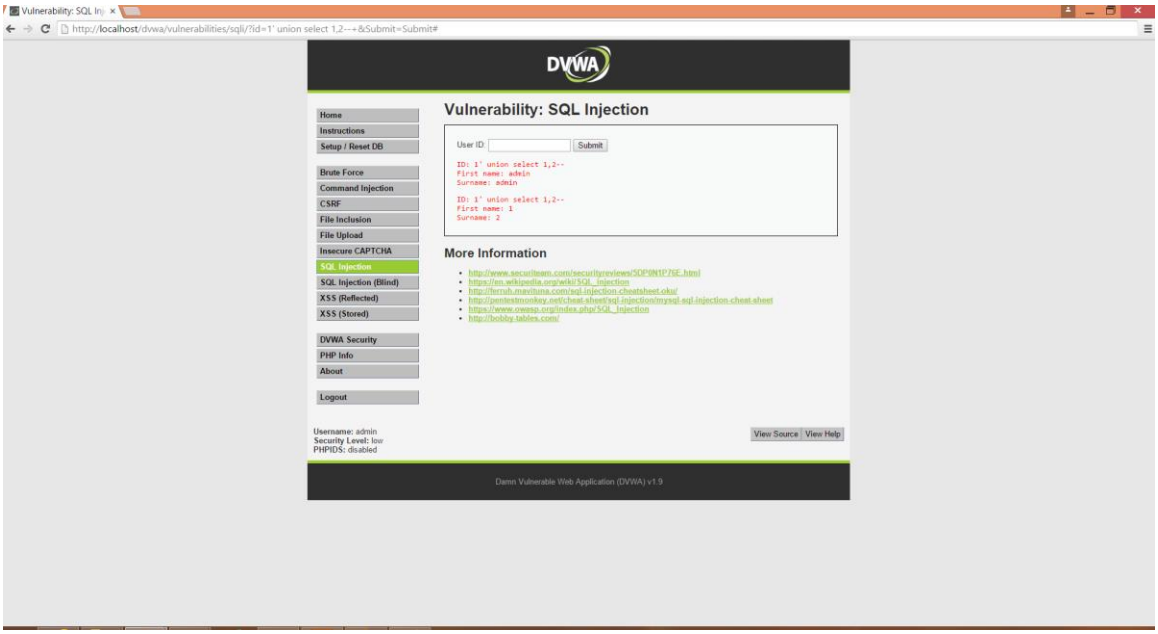


Now continue typing the numbers (2,3,4 etc..) until you get any error reporting for " unknown column in order clause".



4. **Use union statement to combine the results of 2 statements:** Dump the DDL (i.e. Data Definition Language) of the table's columns for custom query by using the UNION SELECT statement with the number of columns you found on the previous step.
   Eg: http://www.examplewebsite.com/samplepage.phpi?id=1 union select 1,2--+

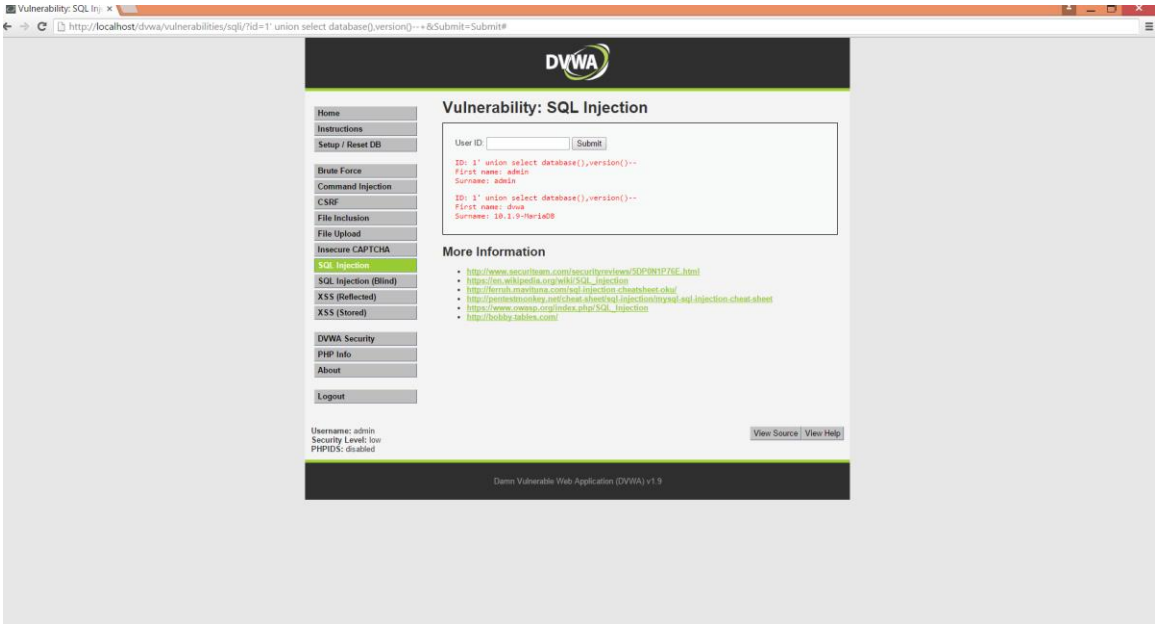
(In this example, there were 2 columns found in the previous step, it could be different in your case).



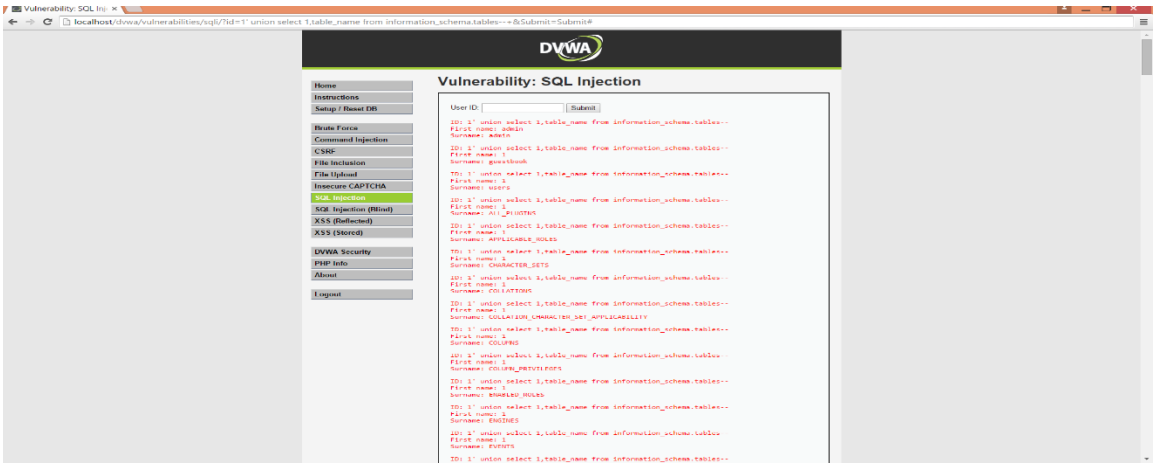
5. **Find out the Database and the Version of MY-SQL used in the website:** It can be done by using two common functions of mysql i.e. database() and version(). You need to replace the column numbers after the union select statement in the previous url with these functions.

   Eg : http://www.examplewebsite.com/samplepage.php?id=1' union select database(),version()--+

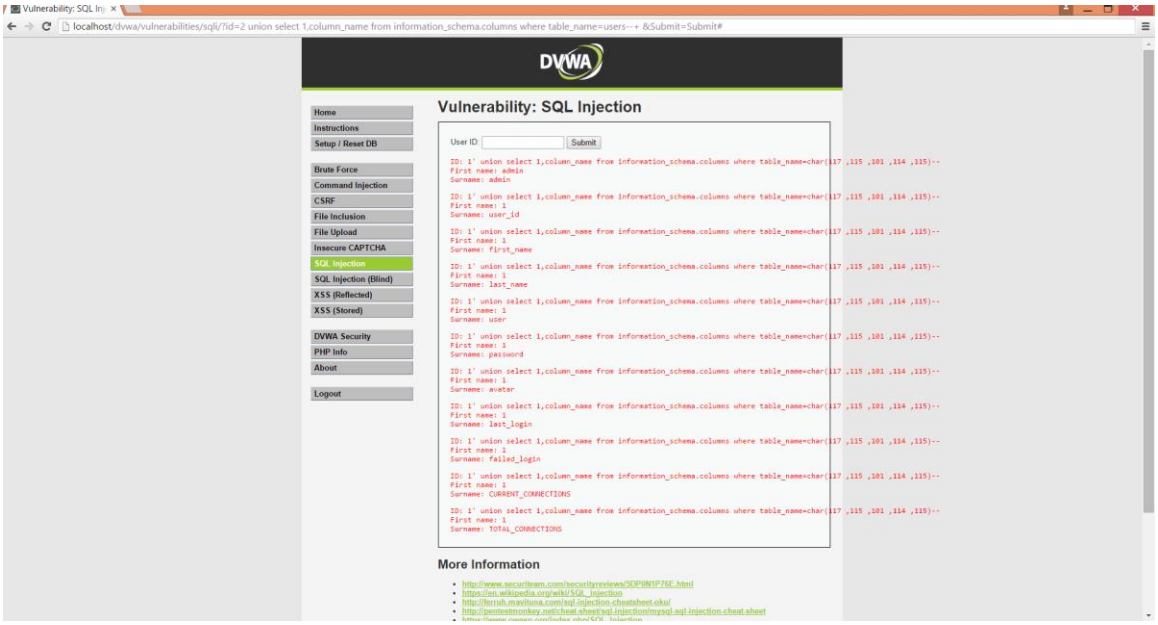
   This will return the database name and the version of the database used in the website. The database version should be greater than 5 if not this injection will not work.



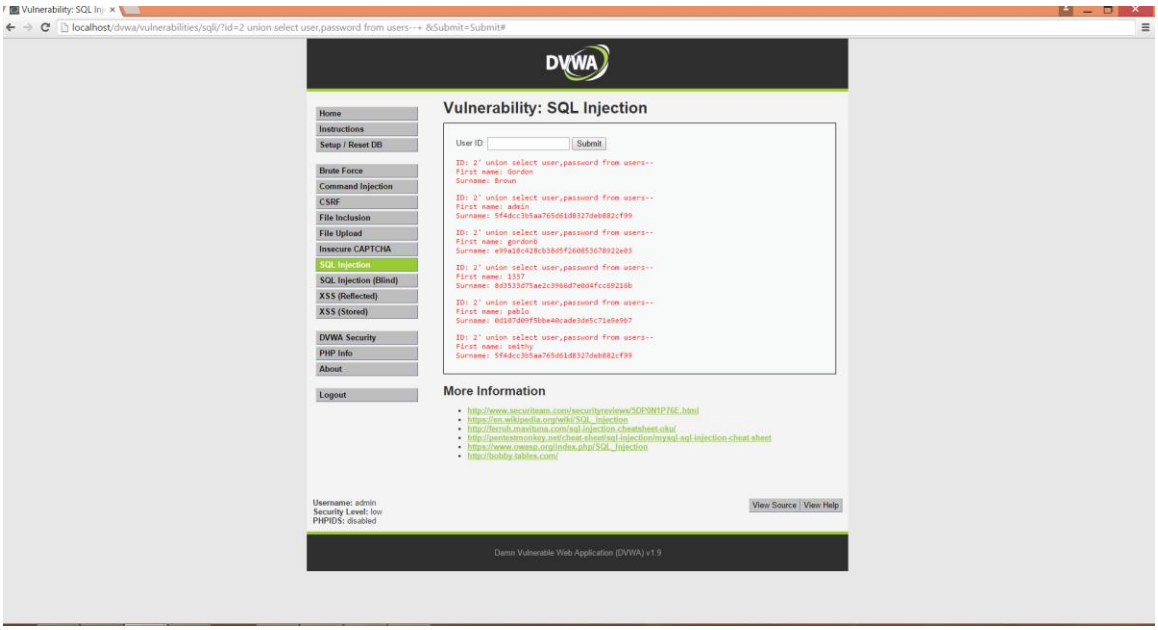
6. **Extract the entire list of tables in the database**: To extract the entire list of tables on the front end of the website we take help of the Information Schema. http://www.examplewebsite.com/samplepage.php?id=1' union select 1,table_name from information_schema.tables--+



7. **Extract the list of columns in our table:** Now we need to get the columns of the tables under which the possibilities of username and password are maximum in this case the table is "users". http://www.examplewebsite.com/samplepage.php?id=1' union select 1,column_name from information_schema.columns where table_name=users--+



8. **Extract the data from the column:** Now we need to extract the data stored in the columns which we found from the above step.
   In this case the columns are “user” and “password” and the table is “users”.



1. Blind SQL Injection

Blind SQL (Structured Query Language) injection is a type of SQL Injection that asks queries to the database with true or false questions and examines the answer based

on the application's response. This attack is often used when the web application is configured to show generic error messages, but has not mitigated the code that is vulnerable to SQL injection.

When an attacker exploits SQL injection, sometimes the web application displays error messages from the database complaining that the SQL Query's syntax is incorrect. Blind SQL injection is nearly identical to normal SQL Injection, the only difference being the way the data is retrieved from the database. When the database does not output data to the front end of the web page, the attacker forces the database management system to do so by asking a series of true or false questions. The attacker receives his desired results from the errors he receives.

Blind SQL injection are mainly of two types:-

1. CONTENT-BASED BLIND SQL INJECTION

In this type of attack, where attacker will try to verify if the database is susceptible to Blind SQL Injection by comparing the results of different queries which return TRUE or FALSE.

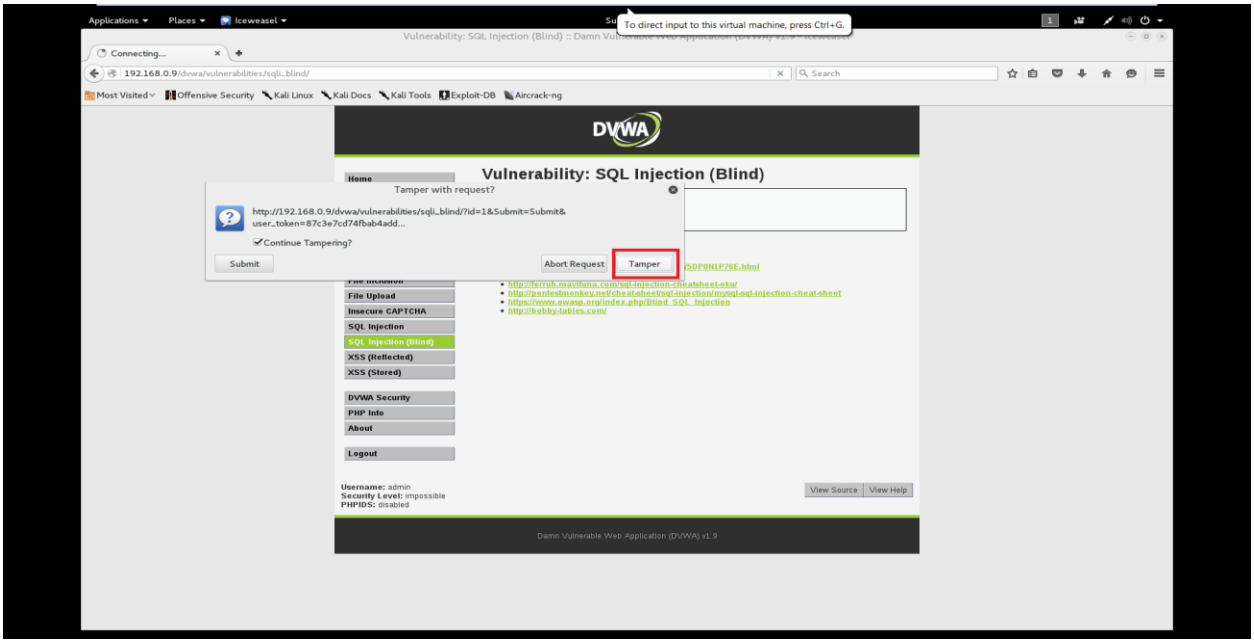
2. TIME-BASED BLIND SQL INJECTION

In Time-based attacks, the attacker needs to instruct the database to perform a time-intensive operation. If the web site does not return a response immediately, the web application is vulnerable to Blind SQL Injection. A popular time intensive operation is the sleep operation.

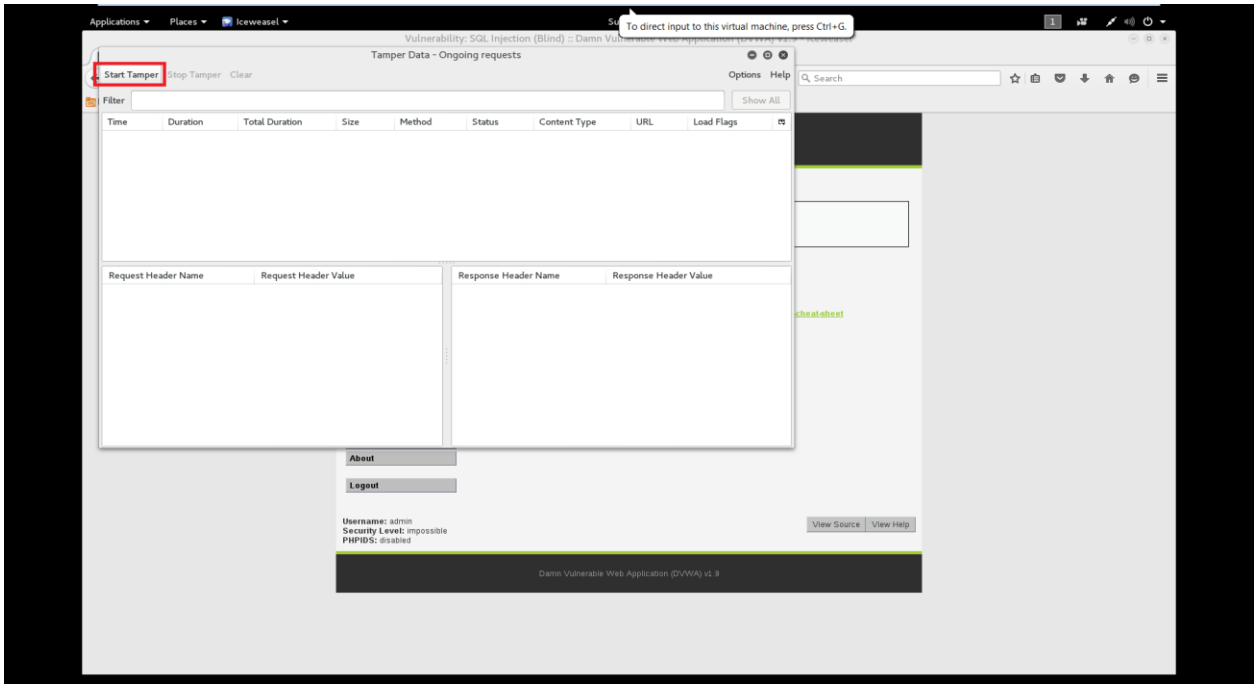
Performing a manual blind based injection is a very time taking process so for an easy demonstration of blind injection we will be using SQLMAP tool provided in Kali Linux to automate the injection in a very less time.

Steps for Performing a Basic Blind Injection using SQLMAP:-

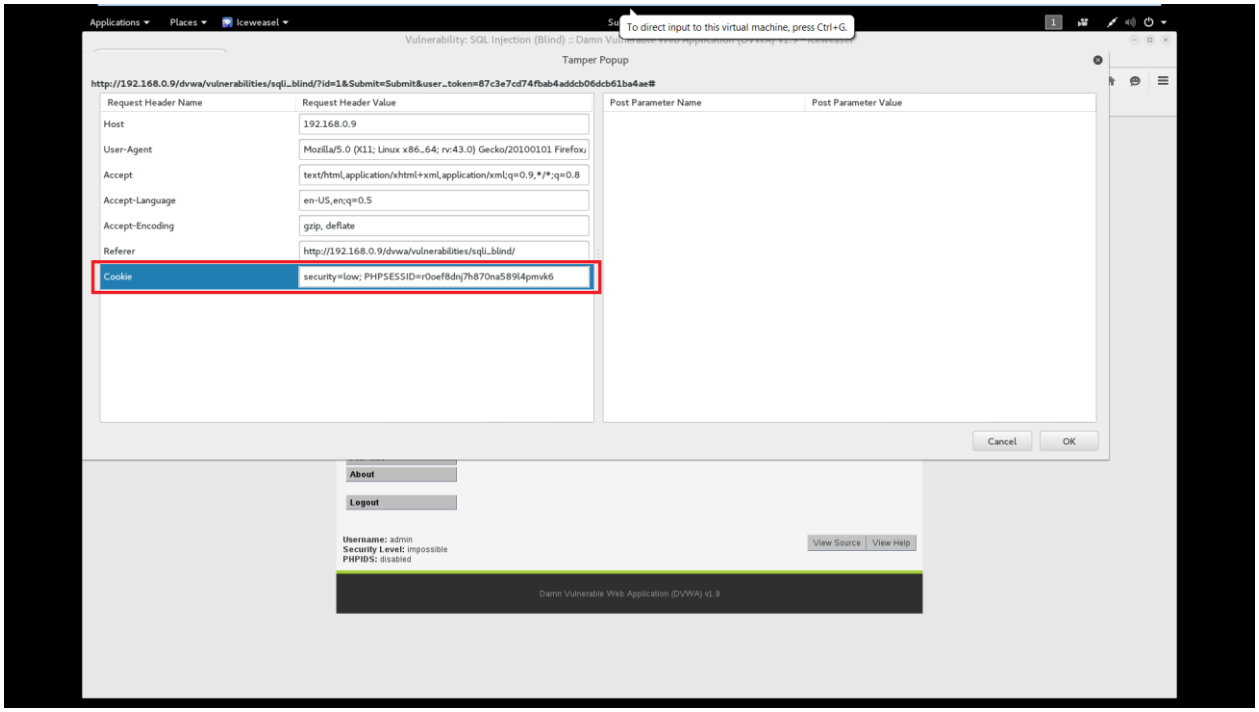
1. We are performing Blind SQL injection on DVWA web site as it is an open source and in designed to practice owasp attacks.
2. **Tamper the post data:** Now we need any tool that can tamper the POST data going out through the web browser, in our case we are using TAMPER DATA plugin for the browser, Press the "**tamper**" button.



3. Now open your website and **insert any number in the user id field** and press the submit button & copy the URL in the address bar of the browser, you will be provided with a dialogue box, press "**start tamper"** button to get the POST data details.



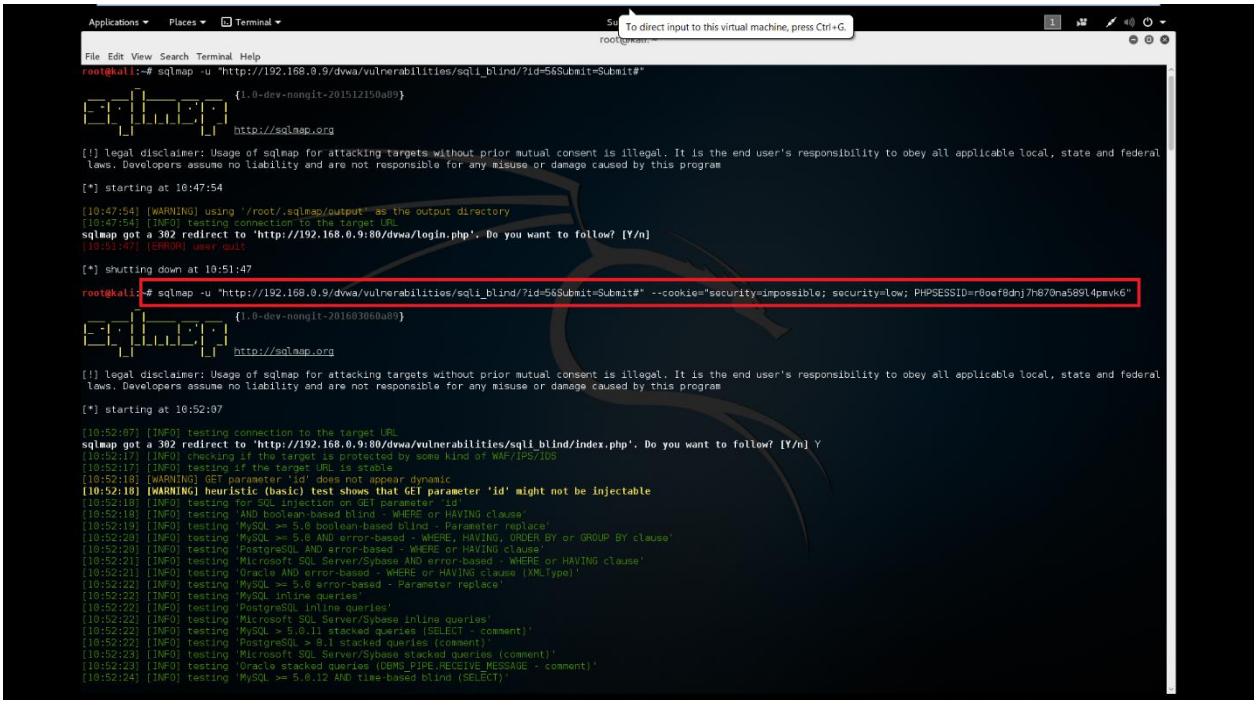
4. Now **copy the details in the "Cookie" tab**, we will need it when performing injection attack.



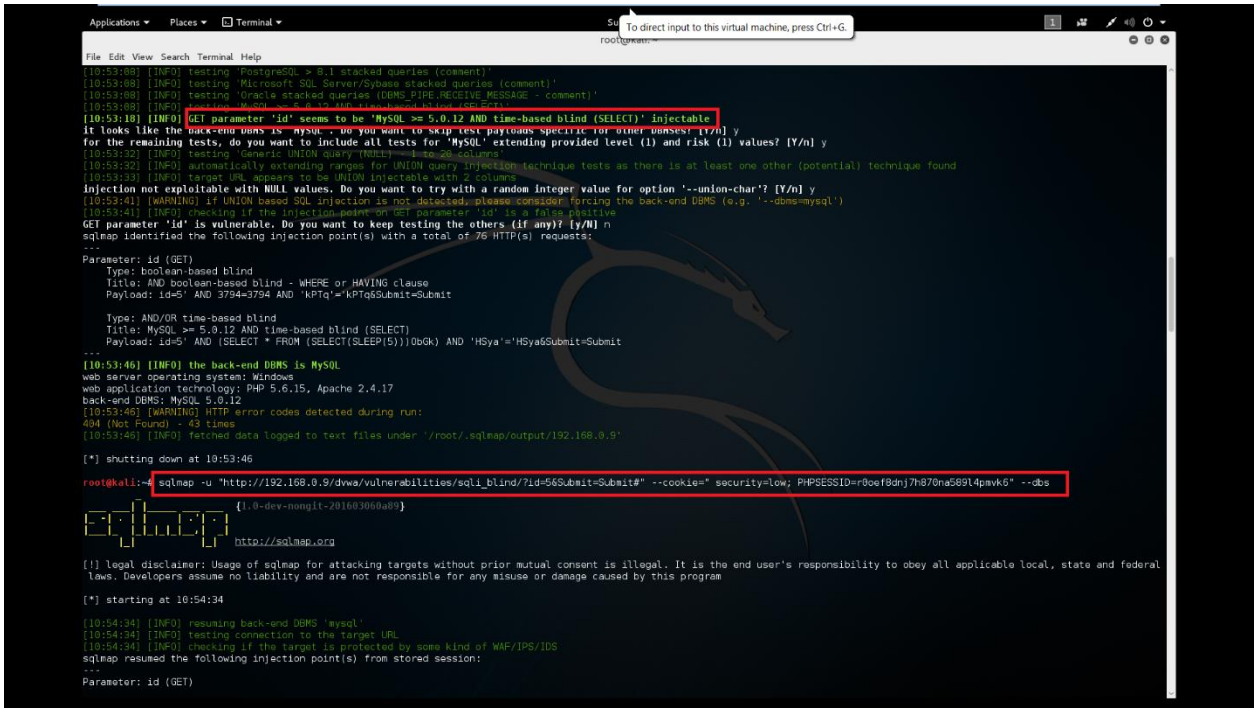
5. Now open the Terminal in Kali Linux and type the following command

   sqlmap –u "http://192.168.0.9/dvwa/vulnerabilities/sqli_blind/?id=5&Submit#" --cookie="security=impossible; security=low; PHPSESSID=r0cef8dnj7hb70na58914pmvk6 "

In the above command, replace the URL with the URL you copied in step 3 and cookie with the cookie you copied in step 4 and the hit enter on your keyboard.

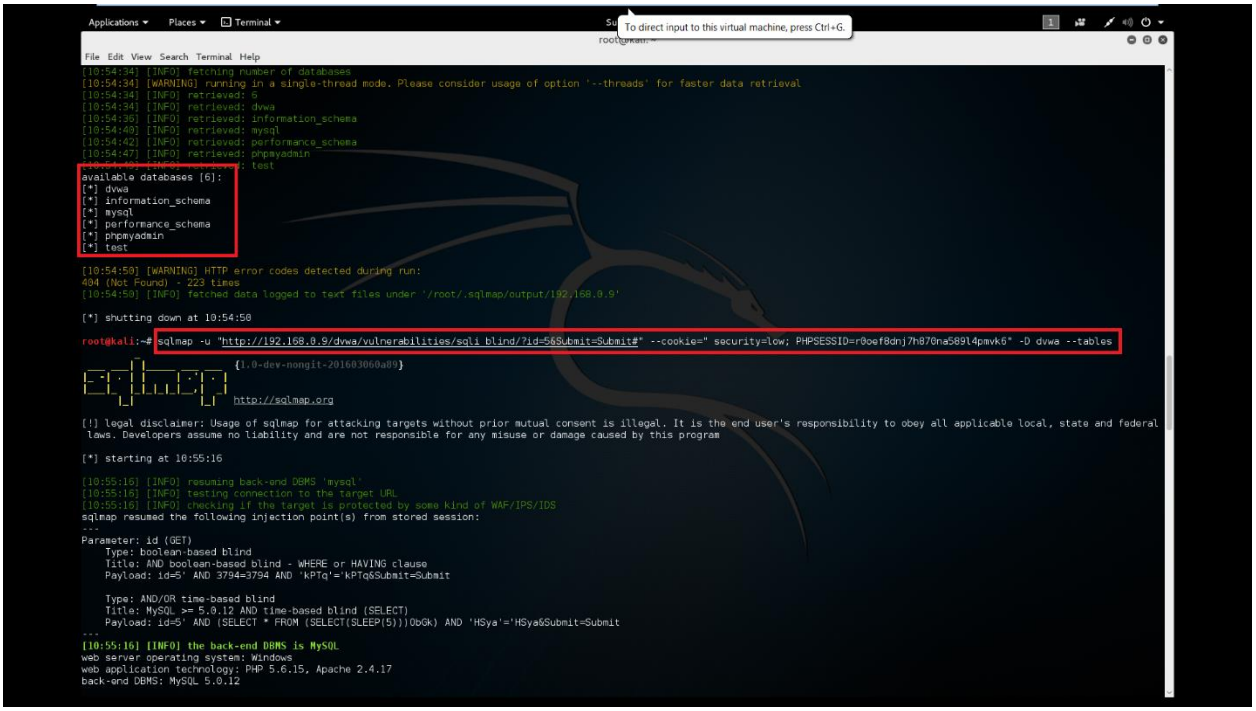


6. As we can see that the website is vulnerable to blind sql injection, we will find the database of the website by adding "**–-dbs"** command at the end of the previous command.

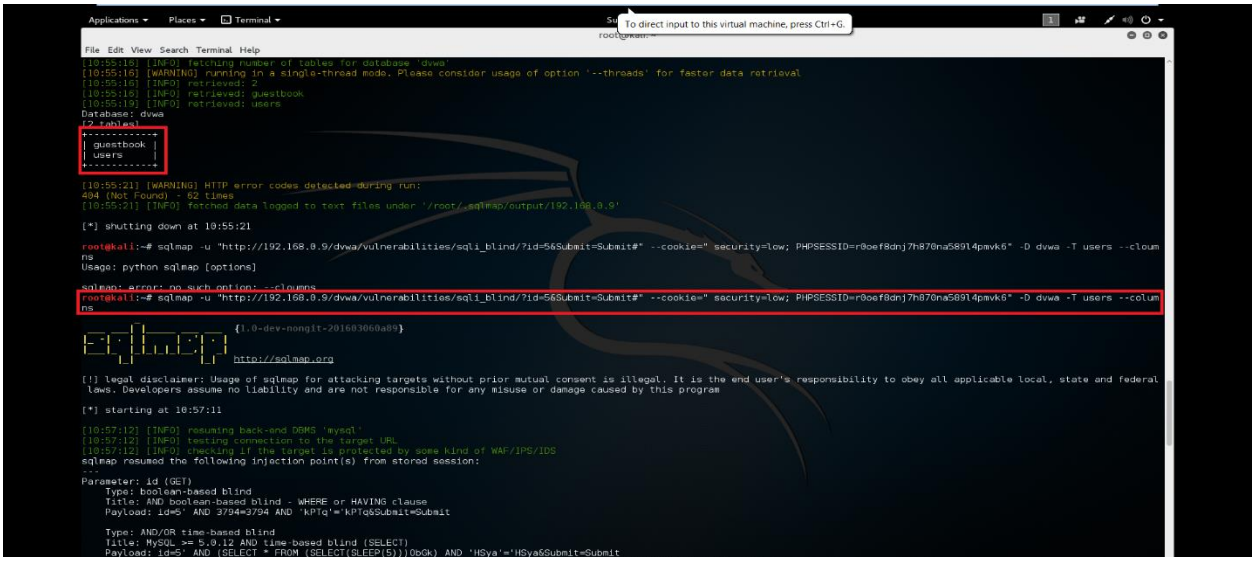


7. As you can see in the image below that we got the entire database of the website, now we will get all the tables under the database **"dvwa"**, the command for searching the tables in a database is :-

sqlmap –u "http://192.168.0.9/dvwa/vulnerabilities/sqli_blind/?id=5&Submit#" --cookie=" security=low; PHPSESSID=r0cef8dnj7hb70na58914pmvk6 " –D dvwa --tables



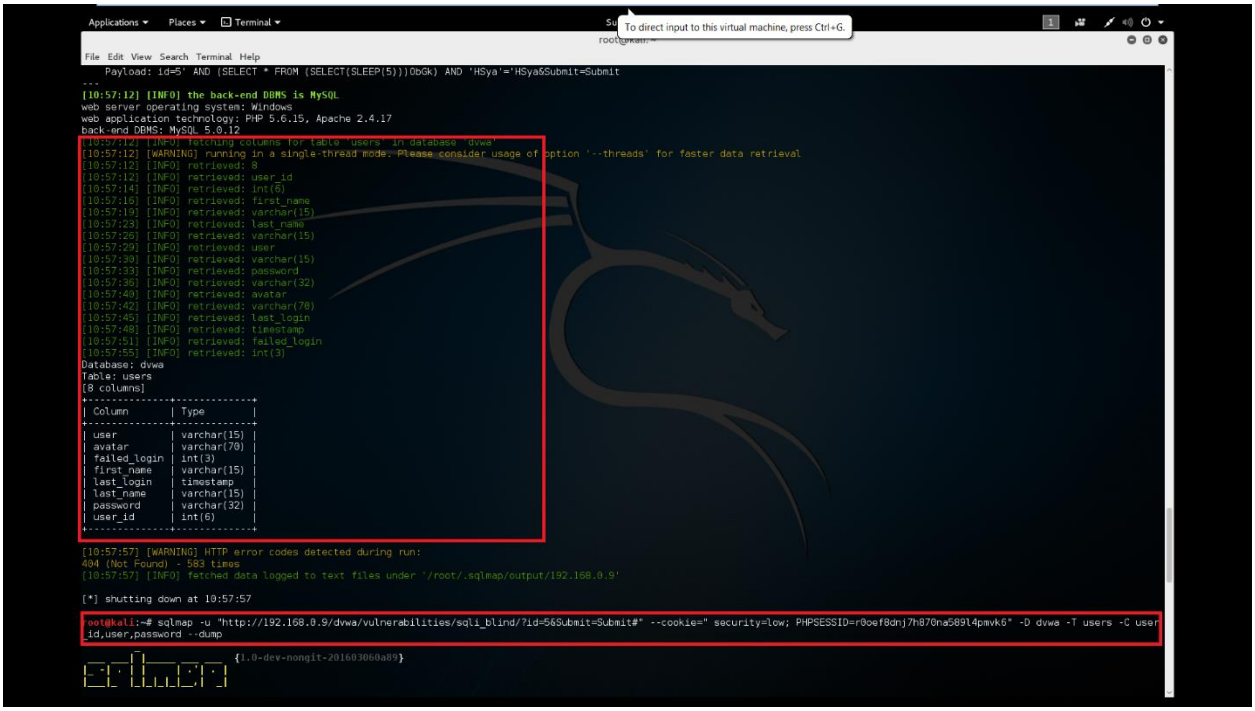
8. We got the tables, now we need the columns under the table "**users**" where we expect the user's credentials to be stored. To get columns we need to add " **–T users --columns**" after "**-D dvwa**" and removing "**--tables**" in the above command.



9. We got all the columns in the Table "users", now we will dump the data on the columns in which we expect the username and password to be. In our case its "**user**" and "**password**". It can be done by using the command below.

   Sqlmap -u "http://192.168.0.9/dvwa/vulnerabilities/sqli_blind/?id=5&Submit#" --cookie="security=impossible; security=low; PHPSESSID=r0cef8dnj7hb70na58914pmvk6 " –D dvwa –T users –C user,password –dump"

In the above command –D is the database name, -T is the tablename, -C is the columns whose details we need to dump.



10. So, we were successful in dumping the username and password from the database using sqlmap for a website vulnerable with blind-sql injection.

Technical and Business Impacts of Injection Attacks?

1. TECHNICAL IMPACTS: "SEVERE"

Injection attacks can result in data loss or corruption, lack of accountability, or denial of access. It can sometimes lead to complete host takeover.

2. BUSINESS IMPACTS: "APPLICATION / BUSINESS SPECIFIC"

For the business impacts of injection attacks consider the business value of the affected data and the platform running the interpreter. All data could be stolen, modified or deleted and could affect the business reputation.

How to prevent Injection attacks?

According to OWASP preventing injection requires keeping untrusted data separate from commands and queries.

1. The preferred option is to use a safe API which avoids the use of the interpreter entirely or provides a parameterized interface.

2. If a parameterized API is not available, you should carefully escape special characters using the specific escape syntax for that interpreter.

3. Positive or "white list" input validation is also recommended, but is not a complete defense as many applications require special characters in their input.

In simple terms,

- Reducing the use of GET methods
- Saving the Credentials in salted encryption form.
- Use of HTTPS instead of HTTP
- Proper validation on the database inputs from the end users.

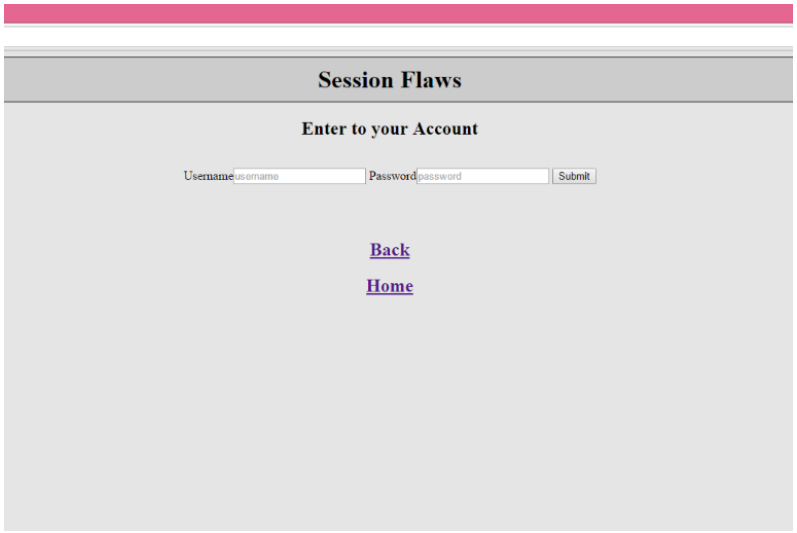
2] BROKEN AUTHENTICATION AND SESSION MANAGEMENT

As per the definition of OWASP, Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, session tokens, or to exploit other implementation flaws to assume other users' identities.

In simpler teams if any unauthorized person may be a client or any attacker is successful in accessing any page on the website or any other user's data without authentication by providing the direct reference to the page or by manipulating some id or value of URL the website lies under Broke Authentication and Session Management.

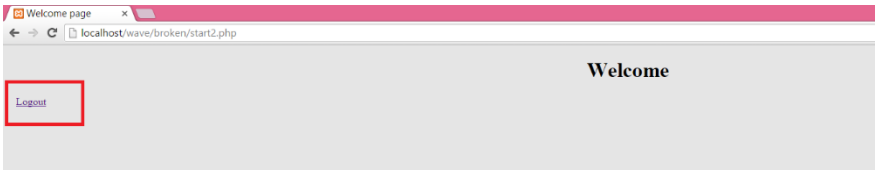
Let us take a look at how an attacker or any unauthorized user can gain access to home page of a login portal which is protected by authentication but its session and authentication system is not properly managed.

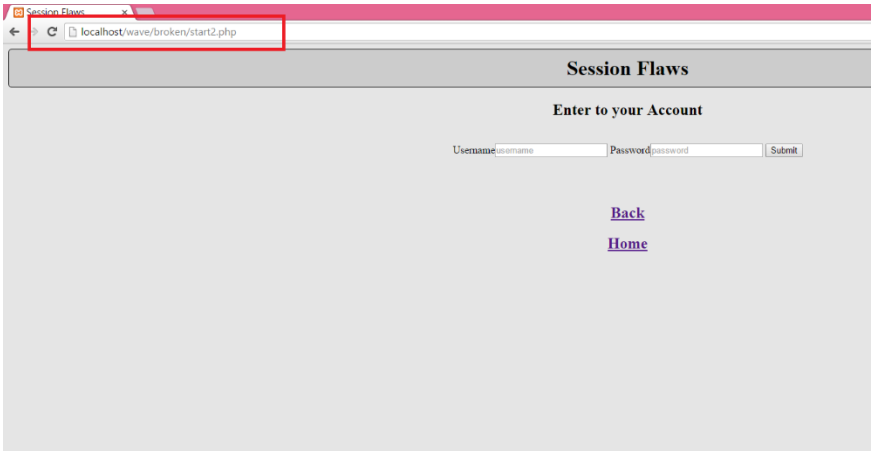
1. Given below is the webpage which requires authentication for accessing the data.



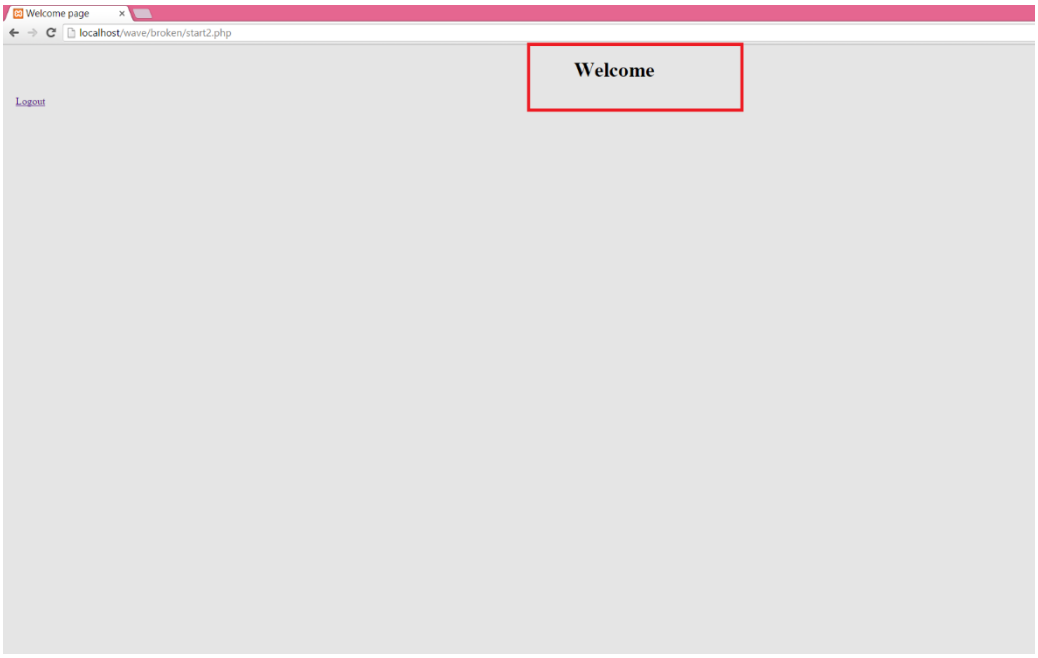
2. After providing the correct authentication details we were successfully landed to the home page named as "**start2.php**" displaying WELCOME message on the top.

3. Now we will logout and will try to directly access the "**start2.php**" page from the address bar of the web browser without entering any login credentials.





4. After giving the direct link of the "**start2.php**" page in the web browser and pressing enter without any login credentials we were able to land successfully at the home page displaying WELCOME message on the top.



5. Since we were able to access a restricted page without any login credentials this means that the authentication system and the sessions was not properly managed on the web site which can easily be used by any outside attacker.

Technical and Business Impacts of Broken Authentication and Session Management?

1. TECHNICAL IMPACTS: "SEVERE"

Such flaws may allow some or even all accounts to be attacked. Once successful the attacker can do anything the victim was allowed to do most frequently the Privileged accounts are targeted.

2. BUSINESS IMPACTS: "APPLICATION / BUSINESS SPECIFIC"

Consider the business value of the affected data or application functions. Also consider the business impact of public exposure of the vulnerability.

How to prevent broken authentication and session management?

1. Use strong authentication and session management controls.
2. Provide simple Interface for the developers to avoid common errors.
3. Session ID's should not be exposed in the URL.
4. Passwords, Session ID's, and other credentials should be encrypted when sent over networks.
5. Application's timeouts should be implemented and used properly.
6. Sessions and tokens should be used and destroyed properly in a given time stamp.

3] CROSS SITE SCRIPTING (XSS)

Cross Site Scripting also known as XSS is found mainly in **dynamic websites**. If any website or any web application takes any kind of executable input from any unauthorized visitor and sends it to a web browser without proper validation or escaping then we can say that the website or application is vulnerable to XSS attack. It allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface websites, manipulate the front end of the website, or redirect the user to malicious sites.

TWO TYPES OF XSS

1. Reflected XSS

This type of XSS has a temporary effect. It effects only a particular user's browser and does not remains permanently for every visitor of the website.

In this type of attack any attacker can add a form on the vulnerable link and send a Input form with details like Name, Credit Card no, Password , PINCODE etc.

2. Stored XSS

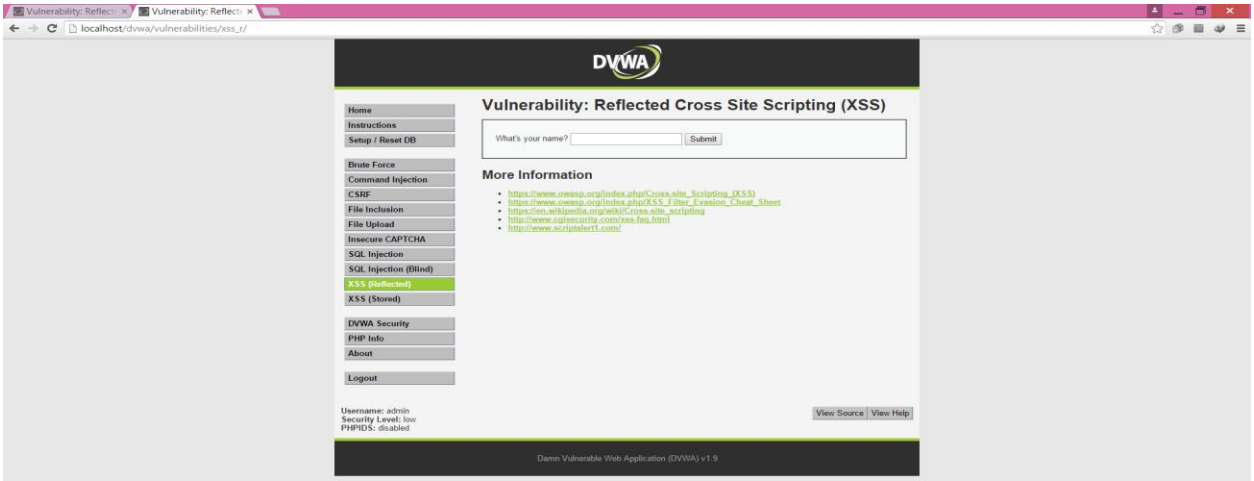
This type of XSS has a permanent effect. If the input field is attached to the database then we can say that it's a stored XSS.

In this type of attack an attacker inputs a malicious java script code to steal cookies of all the visitors of the website who visits for online shopping or banking etc.

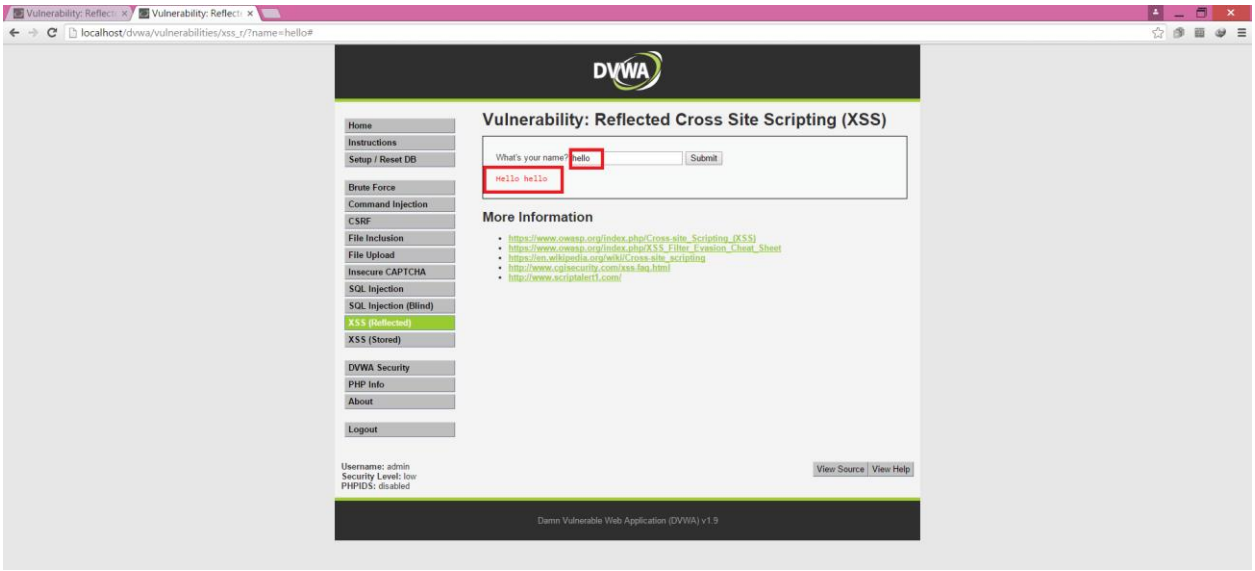
Given Below is a detailed working on XSS attacks on websites

- First let's take a look at how **Reflected XSS** attack works.

1. We are ready with the dvwa webpage on which we will be performing the Reflected XSS attack.



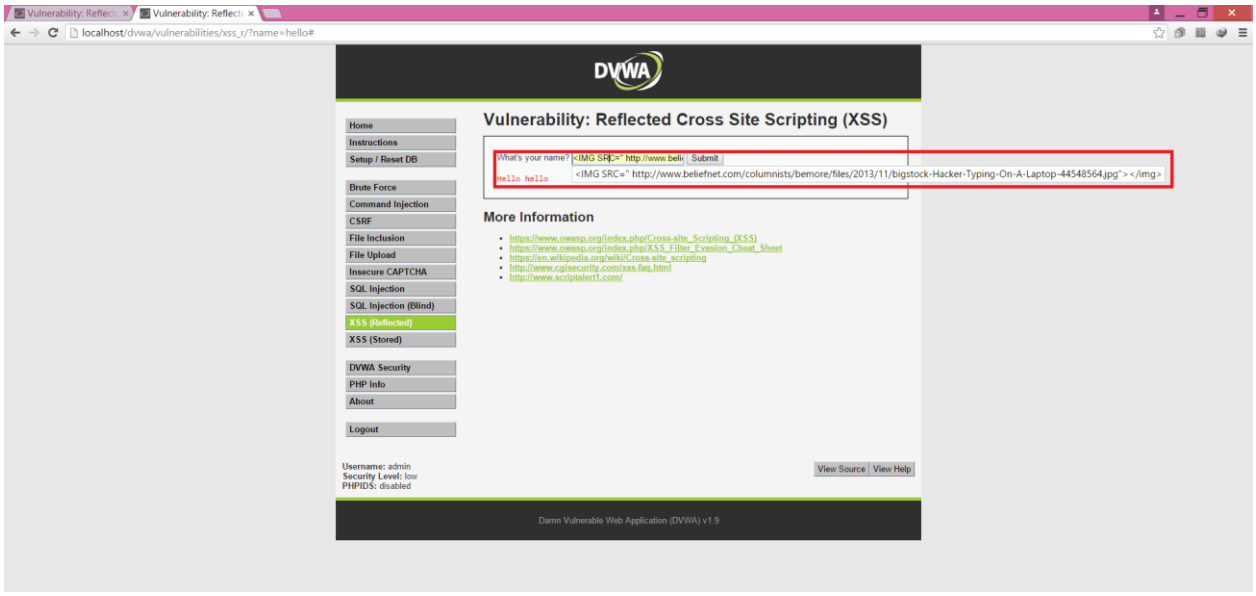
2. Now after entering any name in the input box in the dvwa page and hitting the submit button the webpage returns us the name like show in the image below.



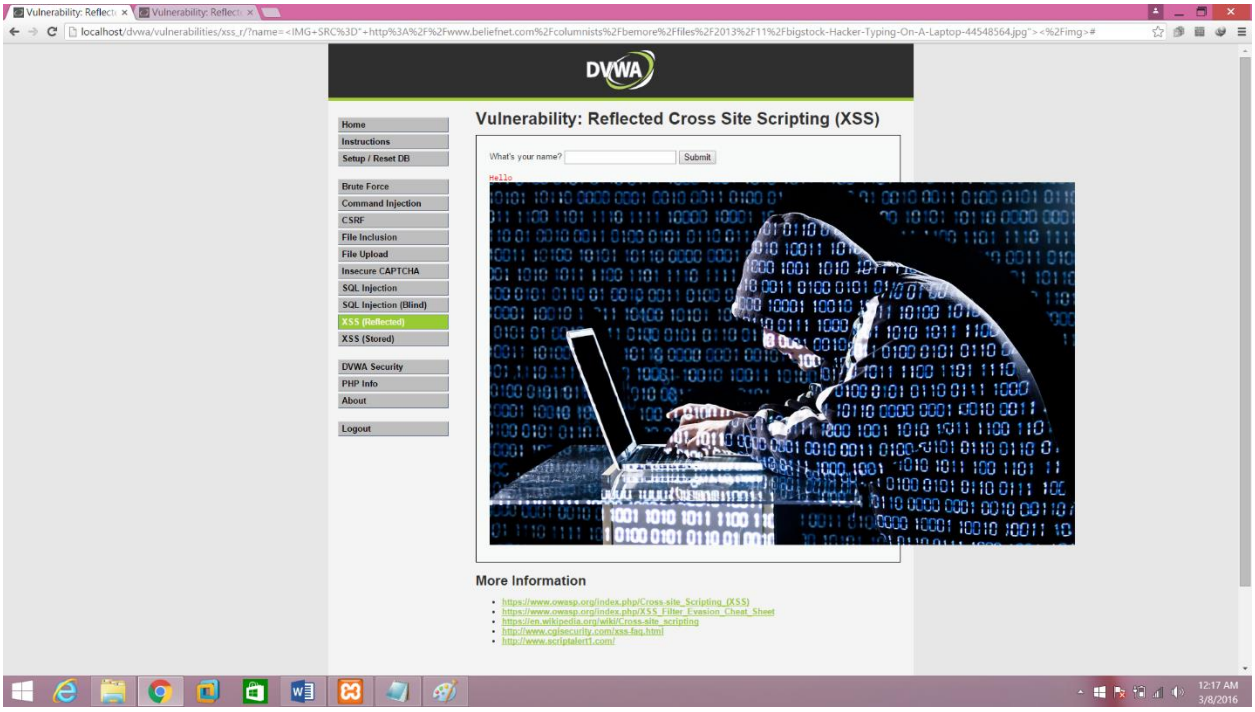
3. Since the website is displaying anything we are typing in the textbox let's try to insert any html code and see whether it executes or not.

   The HTML code used is:-
   <IMG SRC=" http://www.beliefnet.com/columnists/bemore/files/2013/11/bigstock-Hacker-Typing-On-A-Laptop-44548564.jpg"> </img>

   The above code just inserts an image from the given url.



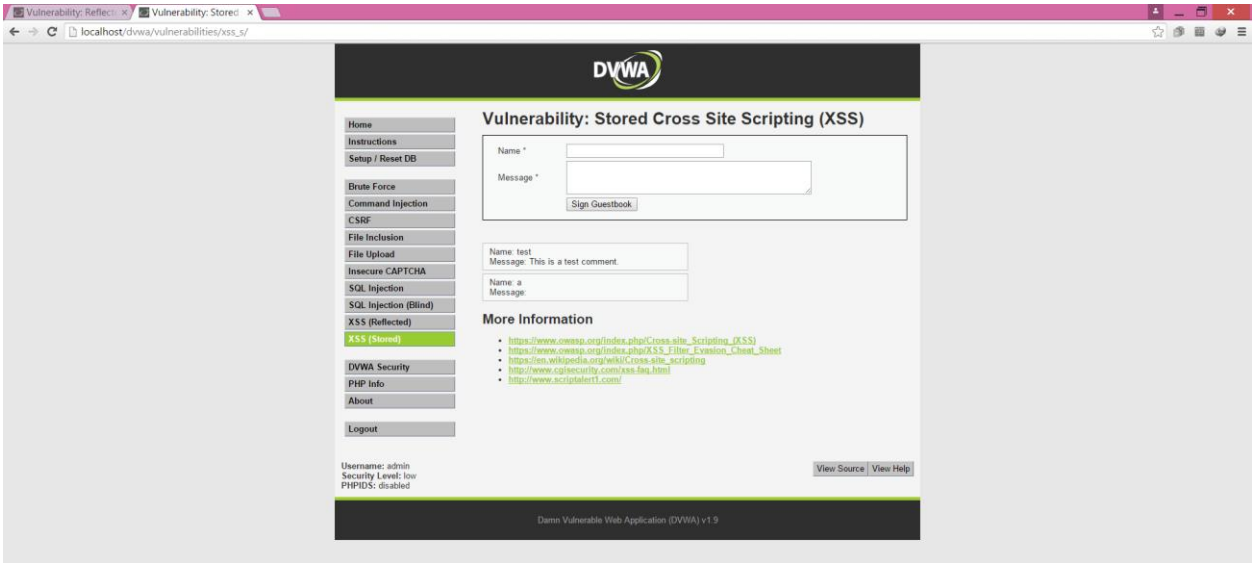
4. After inserting the code and pressing the submit button, we see that instead of name, the webpage is executing our html code and showing the image in the page which should not be allowed it means that we can execute any scripts in this page.



5. The above was a given Scenario of how the Reflected XSS works.

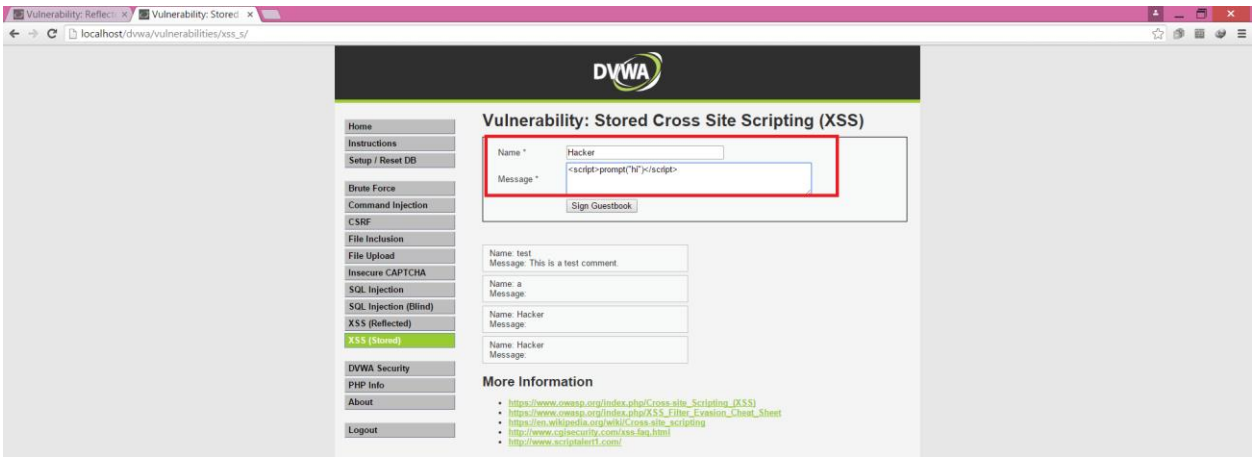
   Now let's take an insight on how **Stored XSS** works:-

1. In the previous attack the impact of our code was not permanent but in this attack our code on execution will update the database and whenever the site is loaded by any user our code gets executed.

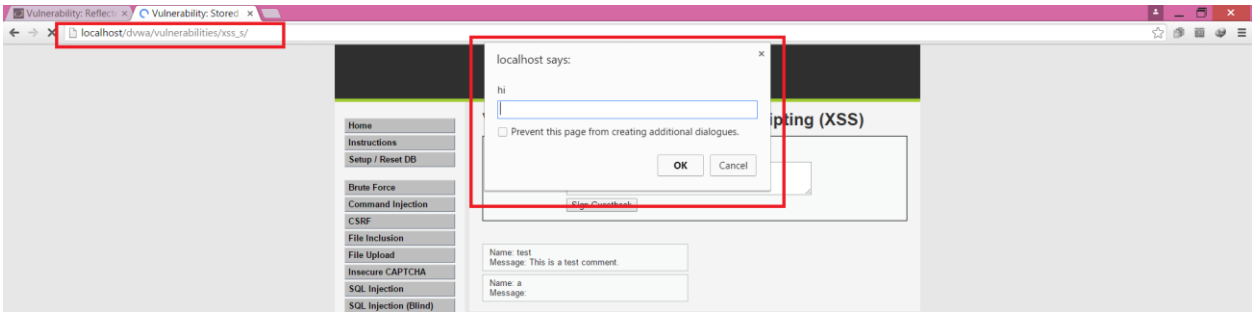


2. So let's take an example, in this dvwa page there are two input fields "**name**" and "**message**".

3. Now **enter any name in the name field and enter any executable code in the message section.** In our case we are using a JAVASCRIPT code that will just display a text "hi" and prompt for any input. After inserting the code hit Sign GuestBook button.
   Code used :
   <script>prompt("hi")</script>



4. After pressing the "Sign Guestbook" button our JavaScript code will be saved permanently in the website's database and whenever any user requests the webpage from the server our JavaScript code executes automatically.
   As you can see in the image below our code has excuted on reloading the webpage displaying the text "hi" in the prompt box of the web browser.



5. This was how Stored XSS works in real scenario.

Technical and Business Impacts of Cross Site Scripting (XSS)?

1. TECHNICAL IMPACTS: "MODERATE"

Attackers can execute scripts in a victim's browser to hijack user's sessions, deface websites, insert hostile content, redirect users to any fake page, hijack user's browser with the help of malwares etc.

2. BUSINESS IMPACTS: "APPLICATION / BUSINESS SPECIFIC"

Consider the business value of the affected system and the data it processes. Also consider the business impact of public exposure of the vulnerability.

How to prevent XSS?

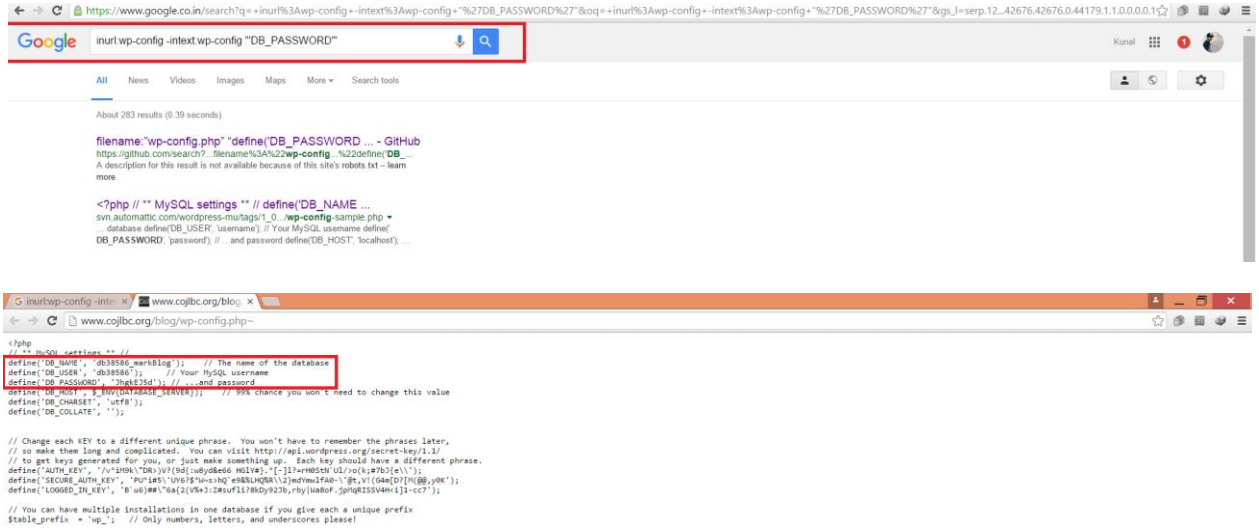
1. To prevent XSS attacks the best option is to properly escape all untrusted data based on the HTML context (body, attribute, JavaScript, CSS, or URL).
2. For rich content, consider auto-sanitization libraries.
3. Consider Content Security Policy (CSP) to defend against XSS across your entire site.
4. Whitelist input validation can also be used to get protection against XSS.

4] INSECURE DIRECT OBJECT REFERENCE

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate theses references to access unauthorized data.

Simply if the website or the server is not avoiding a direct reference to its database keys or any sensitive data by providing appropriate file permissions or by implementing firewall, the data and the website is vulnerable to insecure direct object reference.

Consider a website using inappropriate file permissions in its database keys or the configuration files by giving it the read write and execute permission to all users and groups, any attacker can take the advantage of this security flaw just by using a simple google dork string and can gain the access to the database configuration files of the affected websites.



The above images shows how an attacker can get access to some database credentials just by using google dork if the server or the website has read, write and execute permissions on its files.

Technical and Business Impacts of Insecure Direct Object Reference?

1. TECHNICAL IMPACTS: "MODERATE"

These flaws can compromise all the data that can be referenced by the vulnerable parameter. Until and unless the object references are unpredictable by the attacker, it's easy for an attacker to access all available data of that type.

2. BUSINESS IMPACTS: "APPLICATION / BUSINESS SPECIFIC"

Consider the business value of the exposed data, also consider the business impact of public exposure of the vulnerability.

How to prevent Direct Object Reference?

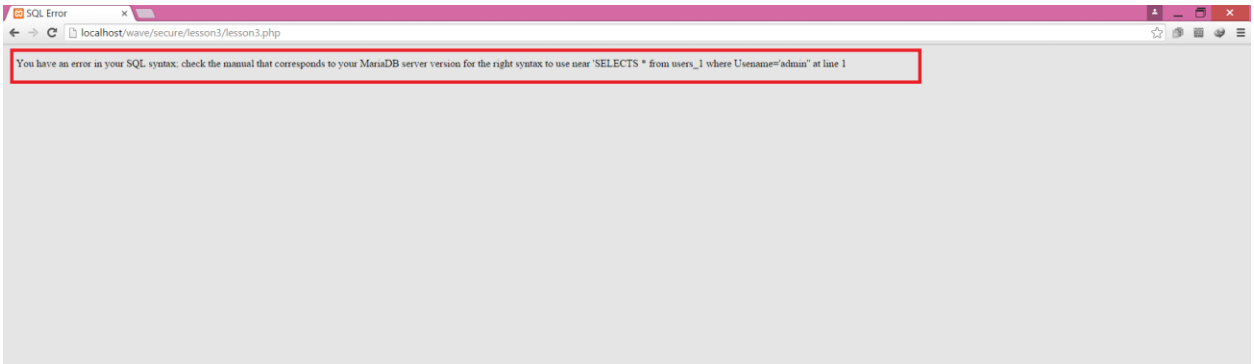
1. The values supplied by the user should be carefully examined through an access control function to ensure that the user is authorized for the requested object.
2. Use per user or session indirect object references, it prevents attackers from directly targeting unauthorized resources.

5] SECURITY MISCONFIGURATION

Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.

In a simple words any firewall or web application's security settings should be configured by the administrator itself as the default configuration is already know to every attacker along with its flaws, so using default configuration may put your web application to danger.

The image below is showing that the Error Reporting for the website and database has not been configured and left at the default state.



Looking at this any attacker can guess that the configurations of the website is at its default state, and also he can assume that the website can be vulnerable to SQL injection.

Hence the attacker can exploit the website easily.

Technical and Business Impacts of Security Misconfiguration?

1. TECHNICAL IMPACTS: "MODERATE"

The vulnerable system should be completely compromised without your knowledge. All the data in it could be stolen or modified.

2. BUSINESS IMPACTS: "APPLICATION / BUSINESS SPECIFIC"

The system could be completely compromised without the administrator's knowledge all of the data in it could be stolen or modified slowly over time.

Recovery costs could be expensive.

How to prevent Security Misconfiguration?

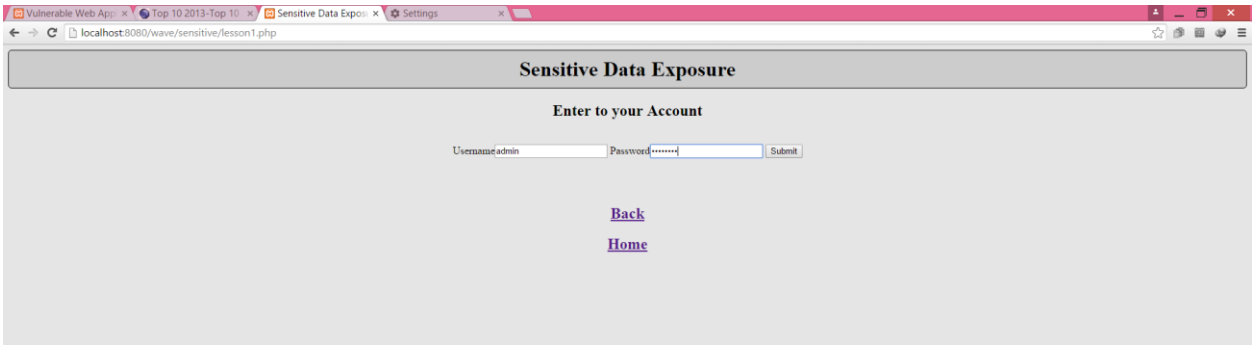
1. Firewall configuration should not be kept at its default state, it should be configured manually.
2. A strong application architecture that provides effective, secure separation between components should be established.
3. Periodic scans and audits should be done to detect future misconfiguration and missing patches.

6] SENSITIVE DATA EXPOSURE

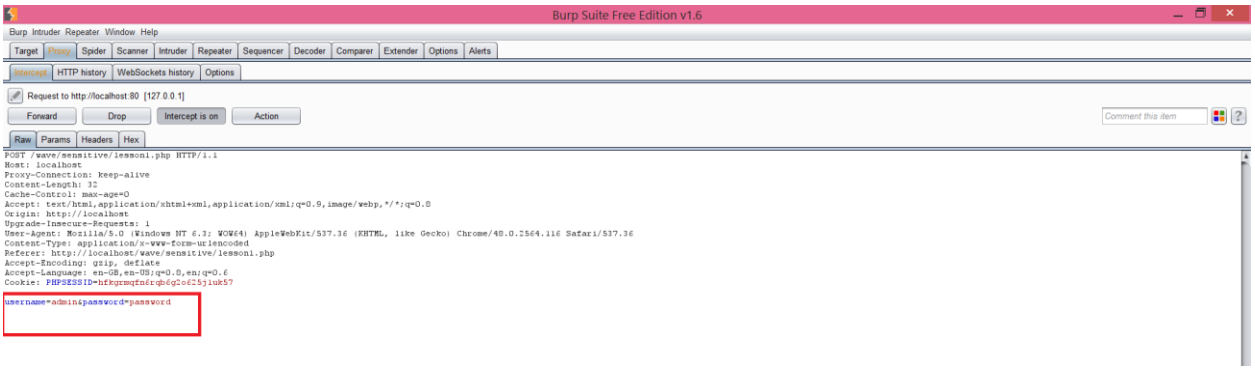
Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.

For this attack let's consider a scenario where an attacker sniffs all the outgoing traffic from the victim's system and get the credentials of the website transferred without any encryption.

1. In the image below assume that a user log's in using his username and password and the website transfers his credentials without any encryption.



1. When an attacker was sniffing the victim's all traffics he was able to get the username and password used for the authentication of the website.



Technical and Business Impacts of Sensitive Data Exposure?

1. TECHNICAL IMPACTS: "SEVERE"

Frequent failures compromises all the data that should have been protected. Typically these information includes sensitive data such as health records, credentials, personal data, credit card details, secret projects etc.

2. BUSINESS IMPACTS: "APPLICATION / BUSINESS SPECIFIC"

The business value of the lost data and impact to the organization's reputation should be considered. The liability of the organization should be kept in mind before the data is exposed.

How to prevent Sensitive Data Exposure?

For all sensitive data following the hints below can reduce the vulnerability to data exposure.

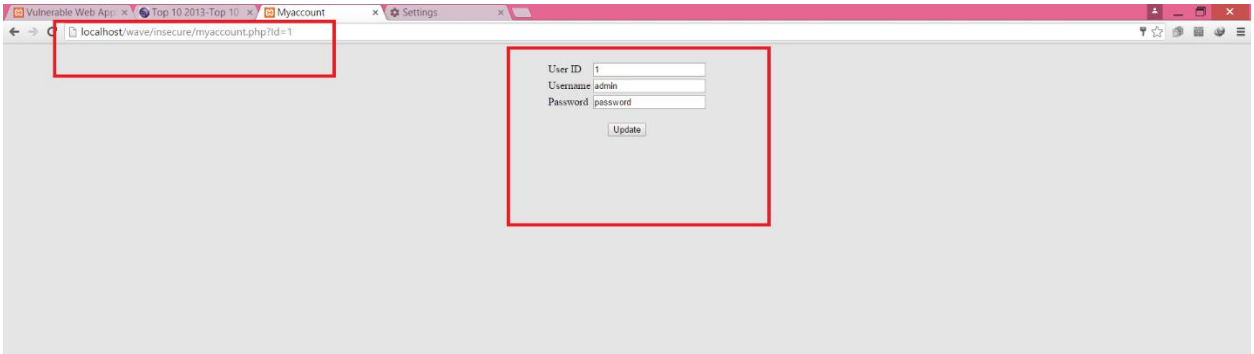
1. Use encryption for all the sensitive data at rest or in transit to defend against threats regarding data exposure.
2. Sensitive data should not be stored unnecessarily it should be discarded as soon as its work is over because data you do not possess cannot be stolen.
3. Passwords should be stored with specific password protection algorithm and also using salted algorithms (mixture of your own hashing algorithm and any standard password protection algorithm available).
4. Autocomplete functions on forms should be disabled in order to prevent them from collecting sensitive data also disable caching for pages that contain sensitive data.

7] MISSING FUNCTIONAL ACCESS CONTROL

Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.

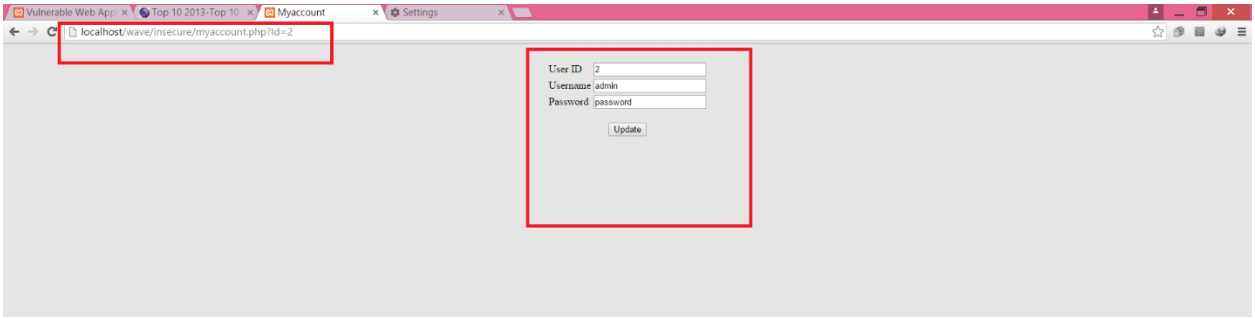
For example, if attacker can just change the value of the id or value in the URL it can lead it to access the other level control and can see the details of other users.

1. In the image below, a user with low level access is logged in and he can see his details in the screen i.e. “User ID”, “Username” & “Password”.



2. When this user tries to access some other user’s details by changing the “id” parameter in the URL and he was successful in doing so.

   You can see in the image below, the user with low level can see the details of other users by manipulating the parameter in the URL.



3. The low level user was able to do so because there was no authentication done at the server side to check whether the data is accessed by appropriate user.

Technical and Business Impacts of Missing Functional Level Access Control?

1. TECHNICAL IMPACTS: "MODERATE"

Such flaws allow attackers to access unauthorized functionality. Administrative functions are key targets for this type of attack.

2. BUSINESS IMPACTS: "APPLICATION / BUSINESS SPECIFIC"

The business value of the exposed functions and the data they process should be considered. Also the impact to the organization's reputation should be considered if this vulnerability becomes public.

How to prevent Missing Functional Level Access Control?

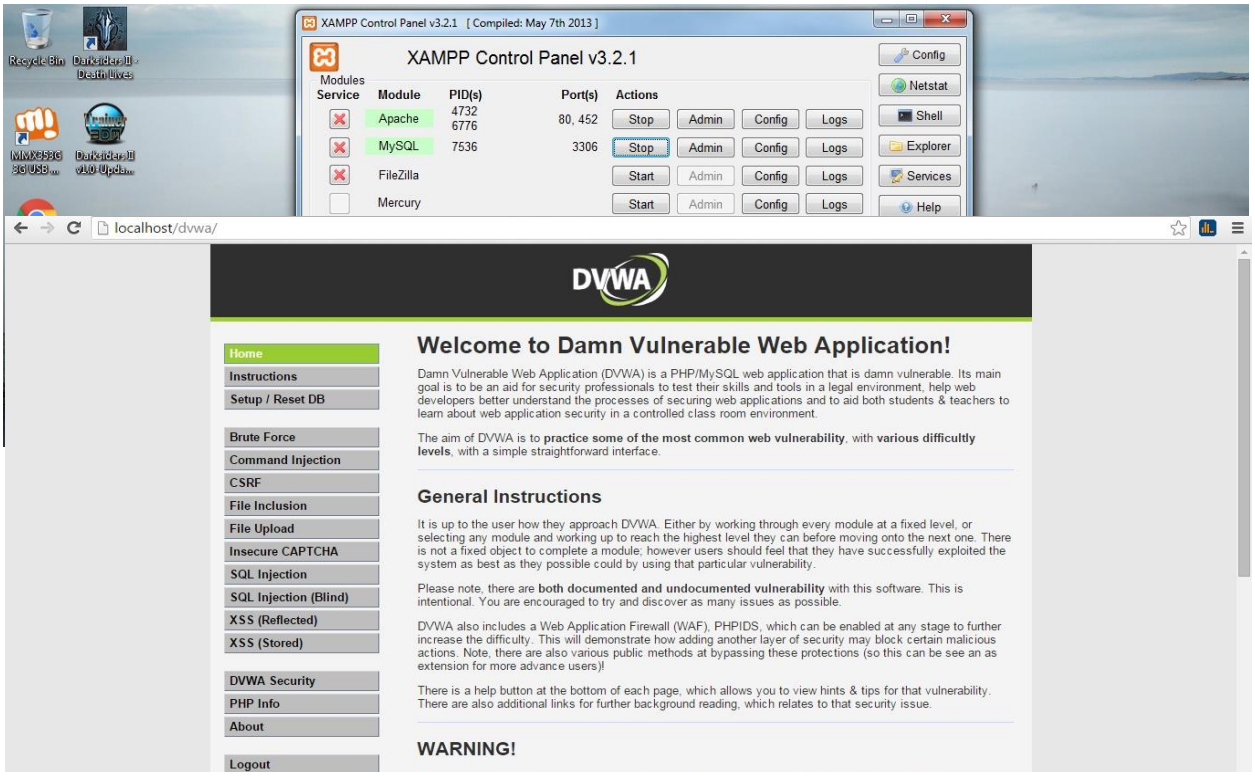
1. Implement checks on the controller or business logic because depending on presentation layer access control will not really help.
2. Use easy to analyze authorization module that is invoked from all of the business functions.
3. Do not hard code the process for managing entitlements and ensure that it can be updated and audited easily.
4. The enforcement mechanism should deny all access by default, requiring explicit grants to specific roles for access to every function.
5. Perform proper checks before allowing access to any function.

8] CROSS SITE REQUEST FORGERY (CSRF)

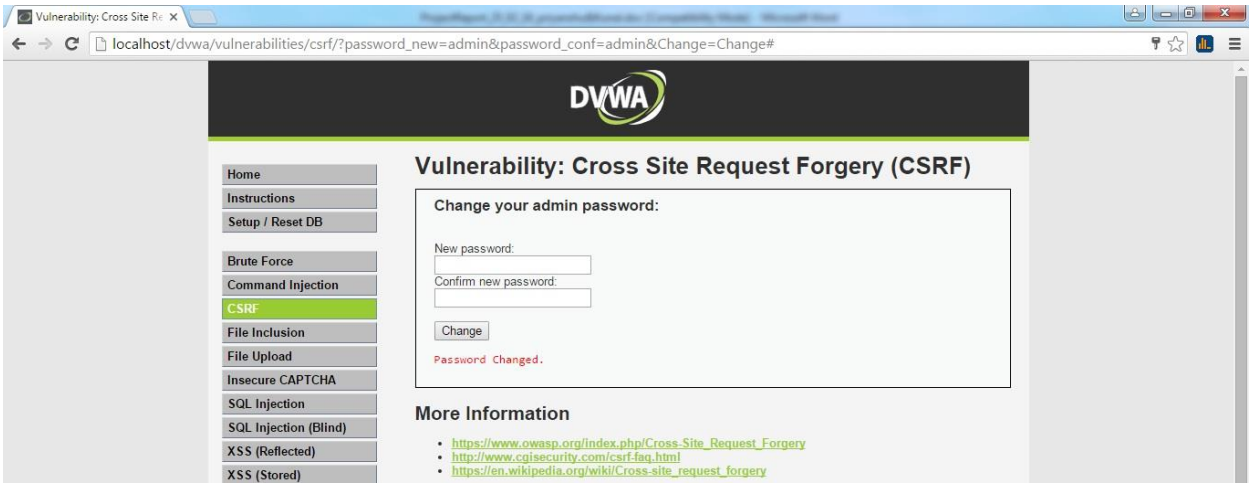
A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests that the vulnerable application thinks are legitimate requests from the victim.

It is a very dangerous attack as an attacker can use the victim in such a manner that the victim can't even predict or understand what is going on. For example, suppose a person is trying to transfer some money online using a bank website, the attacker can make up some code in which a popup will occur at victim's browser and on clicking, it can change the values of the text field of the browser completely by its own crafted values which can transfer any amount from victim's bank account to the attacker's and the serious part is that the victim will be unaware from what's happening.

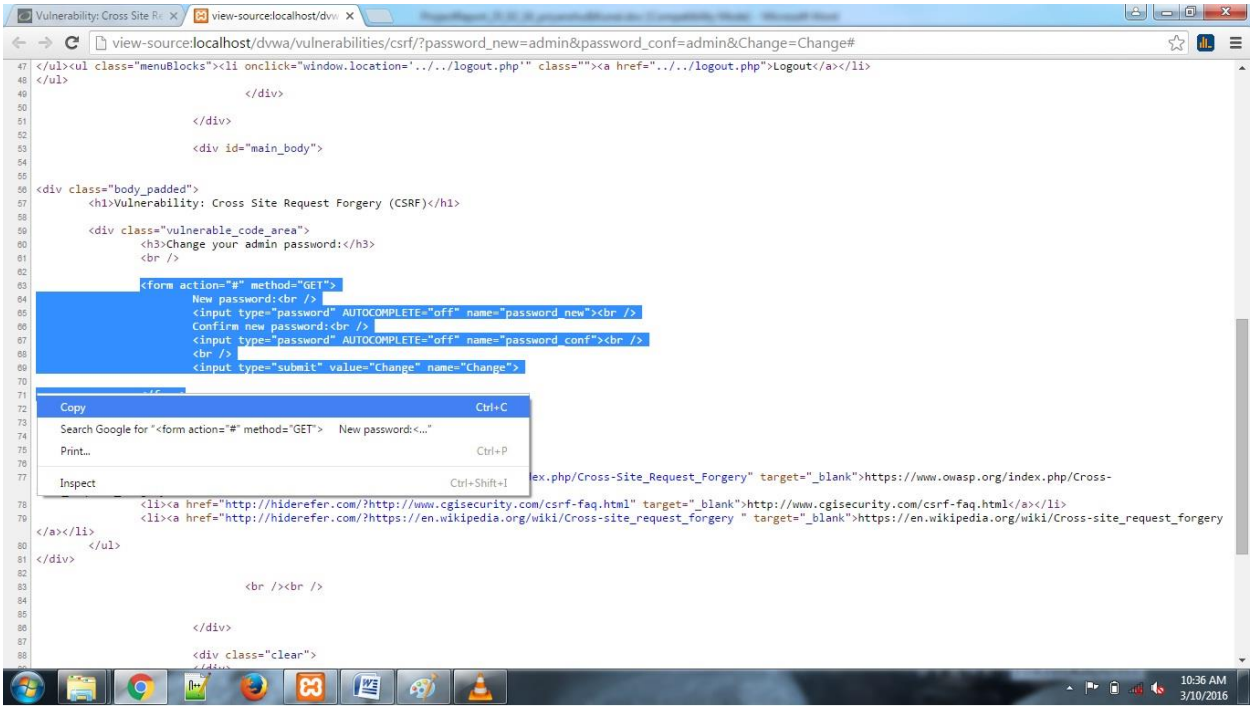
1. We are already set up with our xampp, apache and DVWA for getting a virtual environment to perform a CSRF attack.



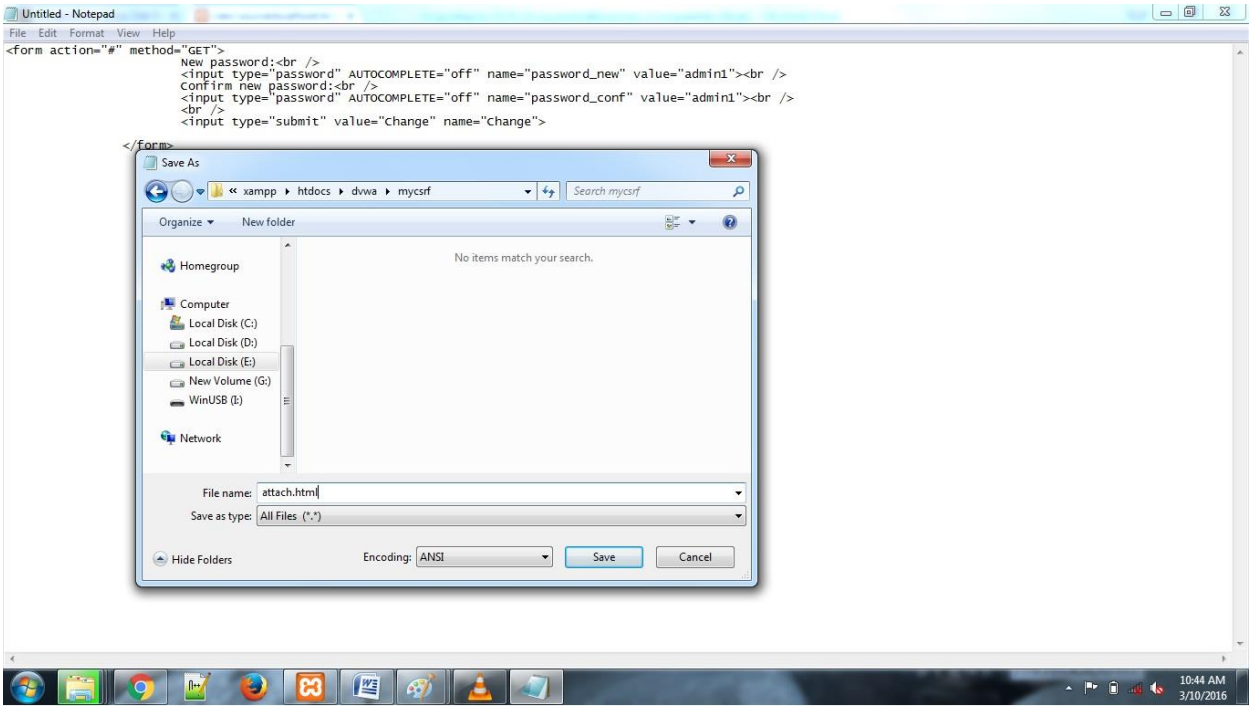
2. Now lets assume the that we need to change the password of any user ‘admin’.



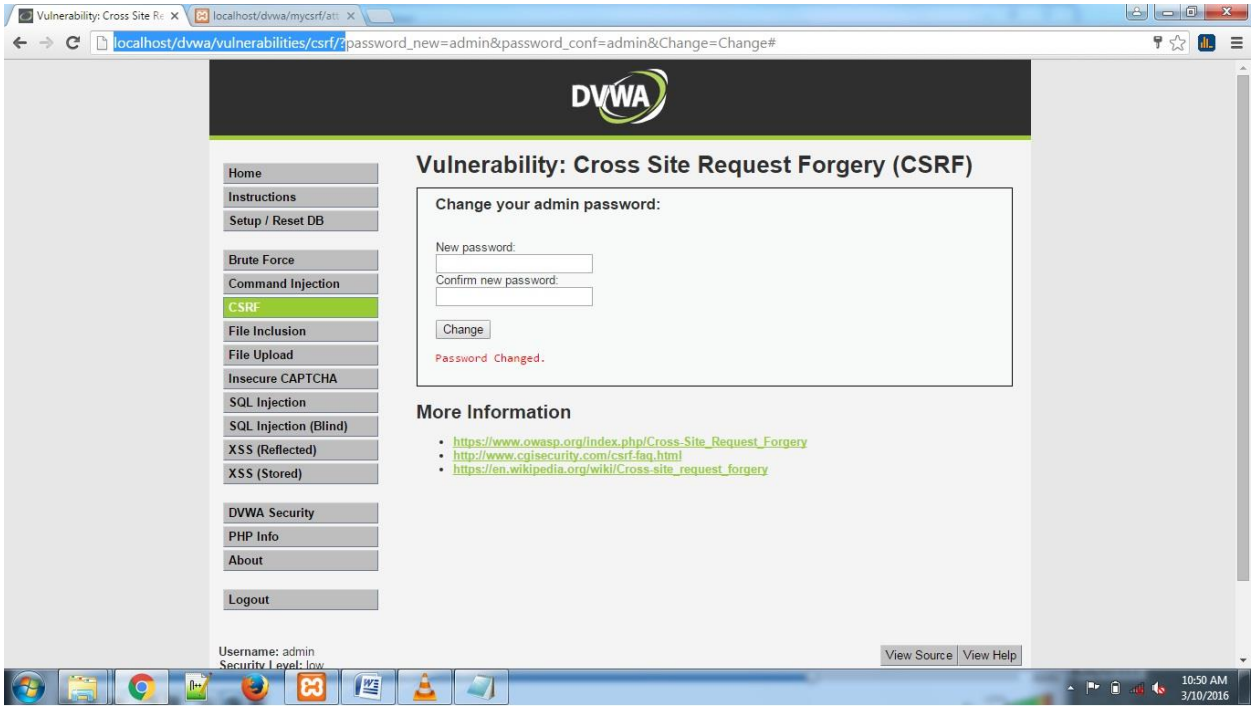
3. Now click on view source of the page to check the source code of the page and we will now do our attack by using some selective code from the page.



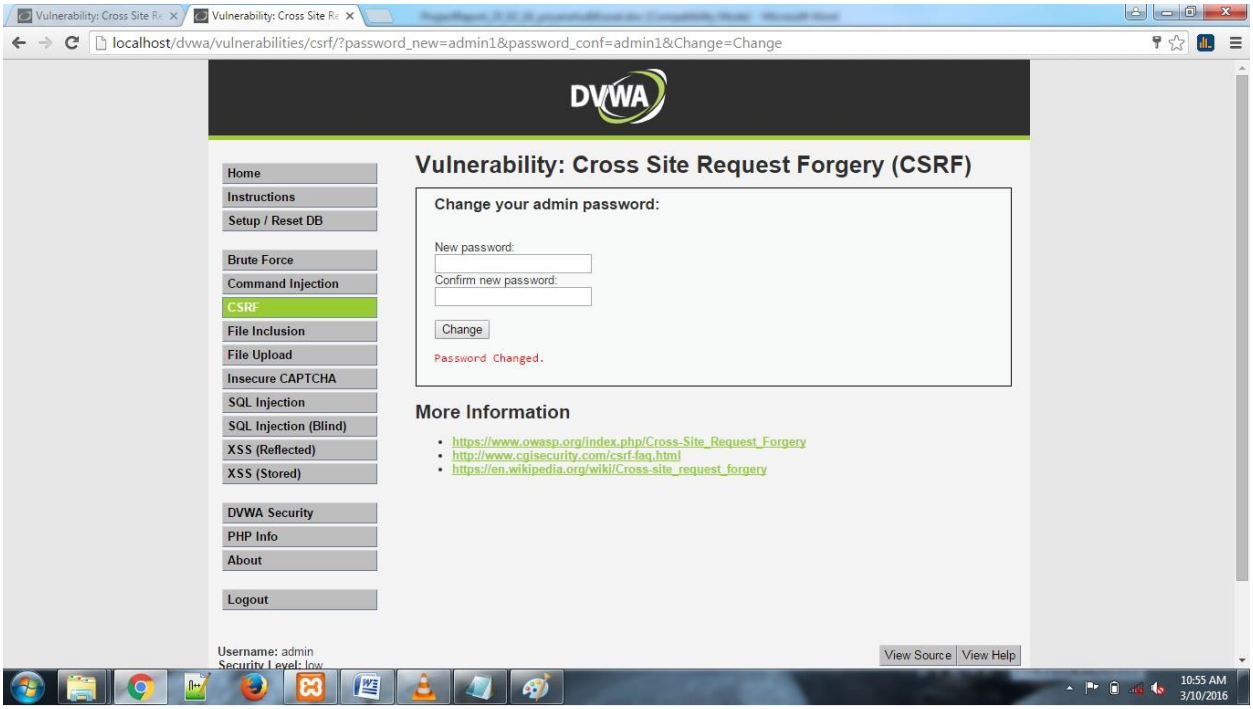
4. Create an html page with any name and paste the code we have selected above in it and save it in localhost folder.



5. Now copy the destination address of the victim from the url and paste in your from action of the page you have just created.



6. Now open the html page you just created, fill in ‘admin’ as the username and put any password in the form of your page and click on the submit button of your html page, as soon as you will click on change button the you will be redirected to the victim page with a message displaying “ password changed”.



7. So just by knowing the user’s name you changed his password without panicking the user and took over his account using cross site request forgery.

Technical and Business Impacts of Cross Site Request Forgery (CSRF)?

1. TECHNICAL IMPACTS: "MODERATE"

Attackers can easily trick victims into performing any state changing operation the victim is authorized to perform. E.g. updating account details, making purchases, logout and even login.

2. BUSINESS IMPACTS: "APPLICATION / BUSINESS SPECIFIC"

The business value for the affected data or application functions should be considered. Consider the impact on the organization's reputation.

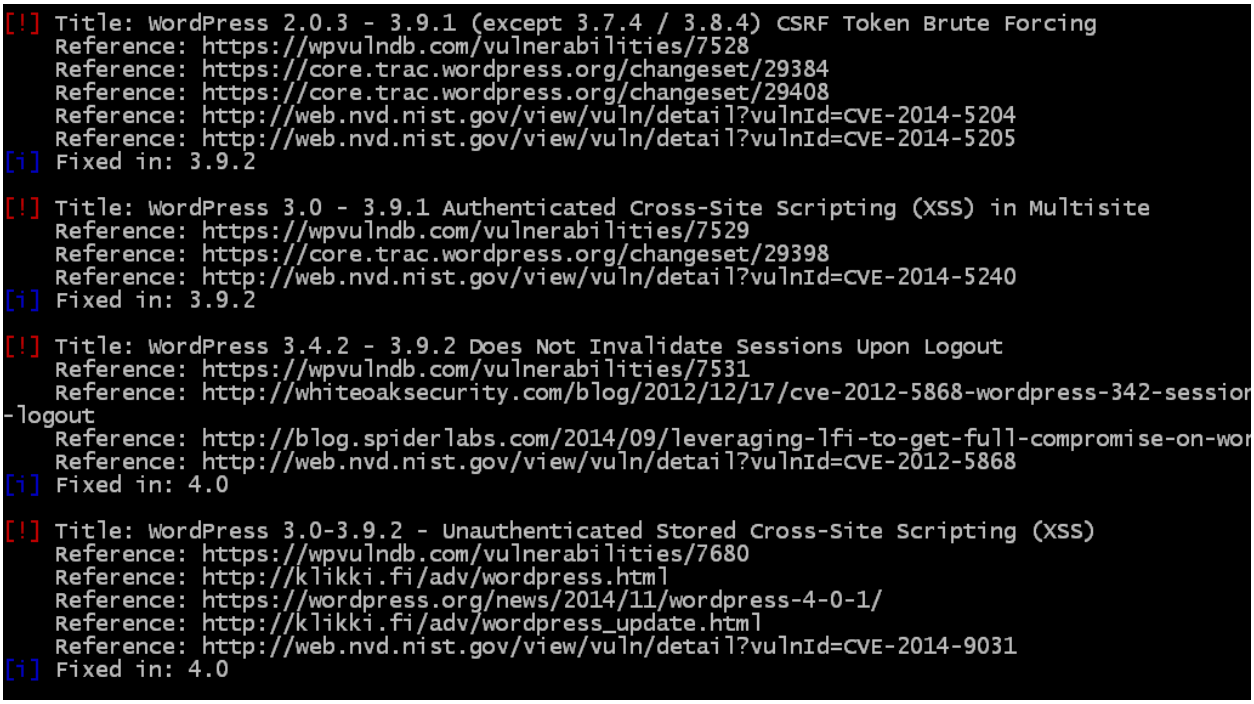
How to prevent Cross Site Request Forgery (CSRF)?

1. Use of tokens should be at a minimum and should be unique, one per user's session.
2. The tokens should be implemented in hidden fields in order to avoid its exposure and inclusion in the URL.
3. Re authentication for the user in order to prove that they are authentic user can also prevent CSRF attacks.

9] USING COMPONENTS WITH KNOWN VULNERABILITIES

Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

In the image below you will be able to find how any attacker can find the components (in this case its plugins providing extra add-on features) installed on a word press website with its vulnerabilities just by making a simple security scan.



After this scan any attacker can craft its attacks according to these results and target these out dated plugins in order to compromise this website's security even though the website is running a firewall these components can give the attacker a direct access to the server.

Technical and Business Impacts of Using Components with known Vulnerabilities?

1. TECHNICAL IMPACTS: "MODERATE"

All type of attacks are possible including injection, broken access control, XSS, etc. The impact could range from minimal to complete host takeover and data compromise.

2. BUSINESS IMPACTS: "APPLICATION / BUSINESS SPECIFIC"

Consider what each vulnerability might mean for the business controlled by the affected application. It could be trivial or it could mean complete compromise.

How to prevent 'using components with known vulnerabilities'?

The best option is not to use any components that are not developed by any trusted source but it's not very feasible. Vulnerabilities for the old versions of most of the components are not released by its organization so upgrade these components or use any other of similar kind which is having all of its security patches up-to-date.

The software development phase should include:-

1. Identification of all components and the versions that are used including all the dependencies.
2. The security of these components in public databases, project mailing lists, security mailing lists, should be monitored and kept updated.
3. Security policies governing the use of components should be implemented.

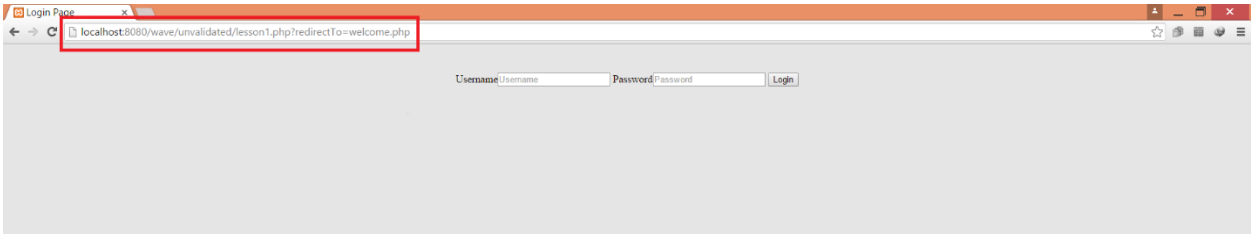
10] UNVALIDATED REDIRECTS AND FORWARDS

Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

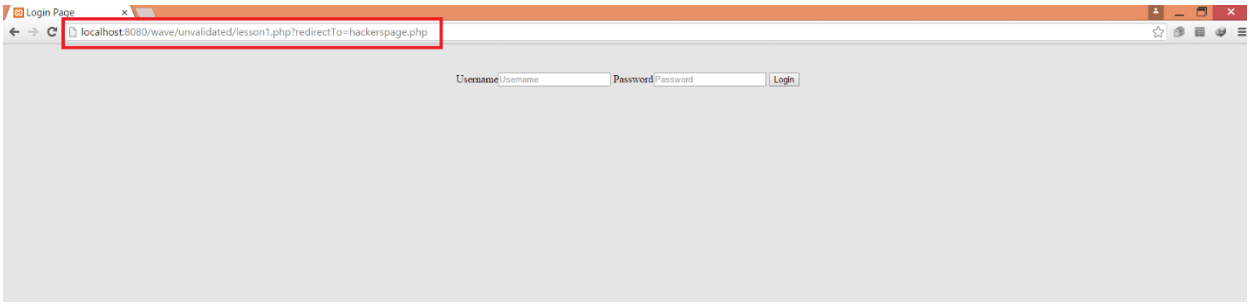
This type of attack are generally not found these days, in this attack the attacker uses the malicious code for forwarding or redirecting the web page to the place he wants the victim to be.

Let's see how an attacker can redirect a user of vulnerable to his own fake site which steals the victim's credentials without his knowledge.

1. Let us assume that the user is at some site which has unvalidated redirects and forwards in our case it's a web site set up by us to demonstrate this attack.



2. You can notice in the browser's address bar that this webpage is redirecting the user to a page named "welcome.php". Now assume that if by chance any attacker changes the redirection point to his own fake page named "hackerspage.php". The URL should look somewhat like shown in the image below.



3. Now if the victim enter the details and hit the “Login” button the instead of getting redirected to the home page of the website the victim will be redirected to a fake page created by an attacker to collect the credentials of the user or for any other intention.

   The victim will trust the webpage because it is from the original source i.e the original website.

4. So this was how an attacker can take the advantage of any unvalidated redirects and forwards in a website.

Technical and Business Impacts of Unvalidated Redirects and Forwards?

1. TECHNICAL IMPACTS: "MODERATE"

Such redirects may attempt to install malware or trick victims into disclosing passwords or other sensitive information. Unsafe forwards may allow access control bypass.

2. BUSINESS IMPACTS: "APPLICATION / BUSINESS SPECIFIC"

Consider the business value of retaining the users' trust also consider the impacts if they get owned by malware or if attackers get access to internal functions.

How to prevent Unvalidated Redirects and Forwards?

For safe redirects and forwards:-

1. Avoid using redirects and forwards unnecessarily.
2. Do not involve user parameters in calculating the destination if redirects and forwards are used.
3. If the destination parameters can't be avoided then ensure that the supplied value is authorized for the user by performing the security check on the full URL.

WINDOWS FIREWALL

1. Understanding Firewall
2. Understanding Windows Firewall with advanced security.
3. Understanding Inbound, Outbound & Connection Security Rules.
4. Monitoring Section.
5. Managing existing windows firewall rules.
6. Creation of outbound rules for windows firewall.
7. Creation of inbound rules for windows firewall.
8. Restoring windows firewall to its defaults.

WINDOWS FIREWALL

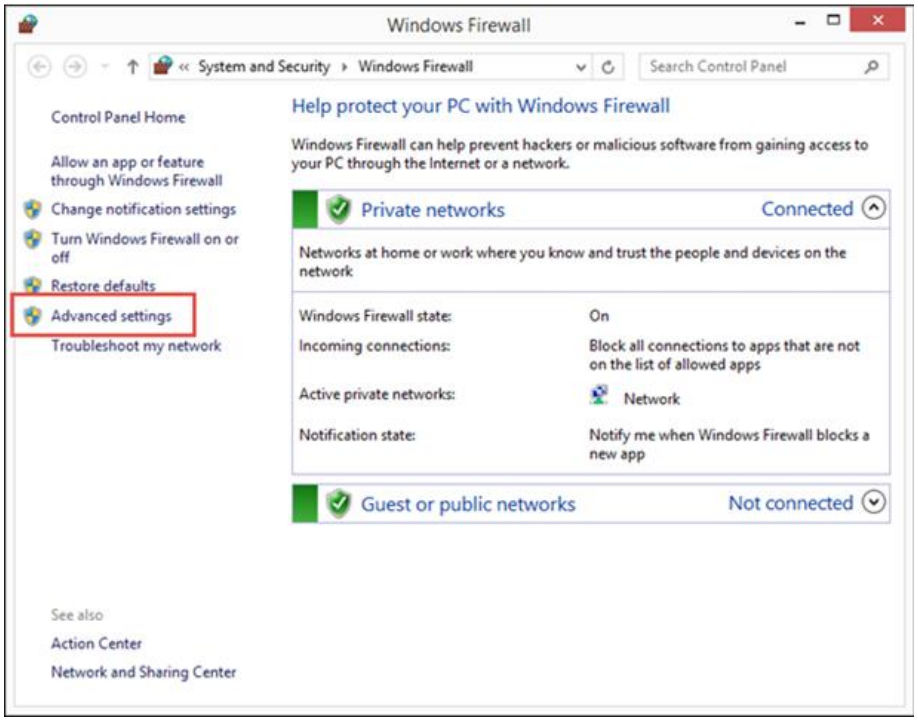
1] Understanding Firewall

- Firewalls can be an effective means of protecting a local system or network of systems from network-based security threats while at the same time affording access to the outside world via wide area networks and the Internet.
- A firewall forms a barrier through which the traffic going in each direction must pass. A firewall security policy dictates which traffic is authorized to pass in each direction.
- A firewall may be designed to operate as a filter at the level of IP packets, or may operate at a higher protocol layer.
- A trusted system is a computer and operating system that can be verified to implement a given security policy. Typically, the focus of a trusted system is access control. A policy is implemented that dictates what objects may be accessed by what subjects.

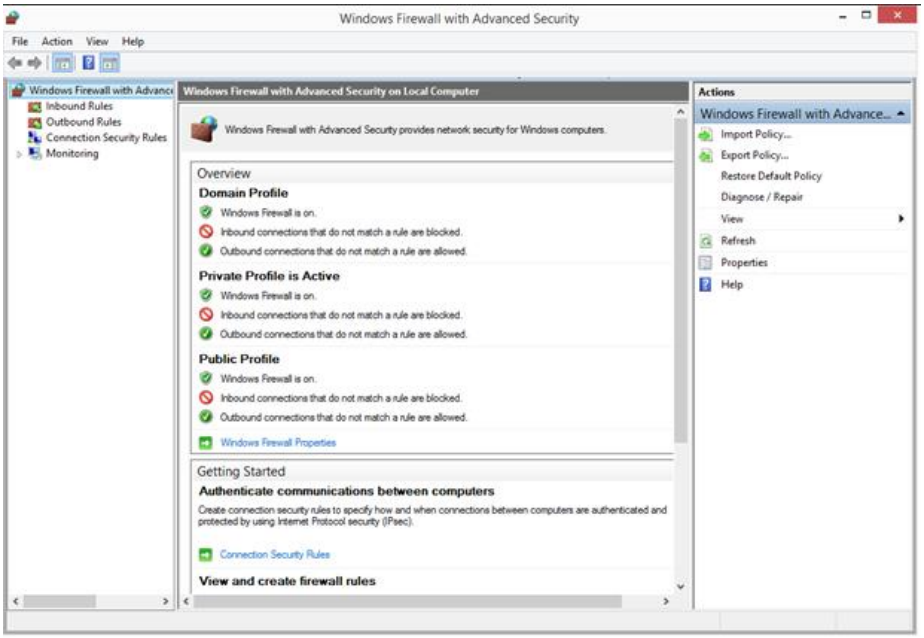
2] Understanding Windows Firewall with advanced security

Windows Firewall is a software component of the windows operating system that provides firewalling and packet filtering functions. The Windows Firewall with Advanced Security is a management work area for the Windows Firewall from which you can control the various sections of the firewall and can decide how the firewall is to work.

- In order to access it, you need to open the Windows Firewall as shown in the previous lesson and then click or tap the “Advanced settings” link on the column on the left.



This is where Windows Firewall stores all its rules at a very detailed level.



3] Understanding Inbound, Outbound & Connection Security Rules

Inbound Rules:

They are the rules applied to traffic that is coming from the network or the Internet to your Windows computer or device.

For example, if you are downloading a file through BitTorrent, the download of that file is filtered through an inbound rule.

Outbound Rules:

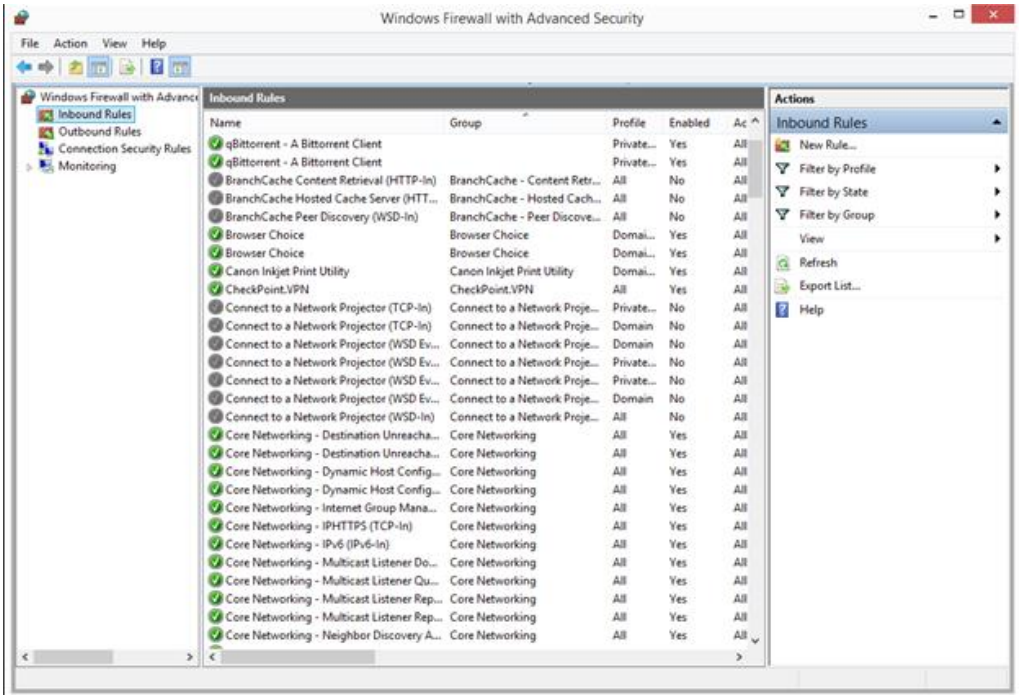
These are the rules applied to traffic that is originating from your computer and going to the network and the Internet.

For example, your request to load the Facebook website in your web browser is outbound traffic and it is filtered through an outbound rule.

Connection security rules:

These are the less common rules that are used to secure the traffic between two specific computers while it crosses the network. This type of rule is used in very controlled environments with special security requirements. Unlike inbound and outbound rules which are applied only to your computer or device, connection security rules require both computers involved in the communication to have the same rules applied. It uses Internet Protocol security (IPsec) to achieve connection security by using key exchange, authentication, data integrity, and, optionally, data encryption.

You can display the rules of a certain type by selecting the appropriate category in the column on the left. All of these rules can be configured so that they are specific to certain computers, user accounts, programs, apps, services etc.



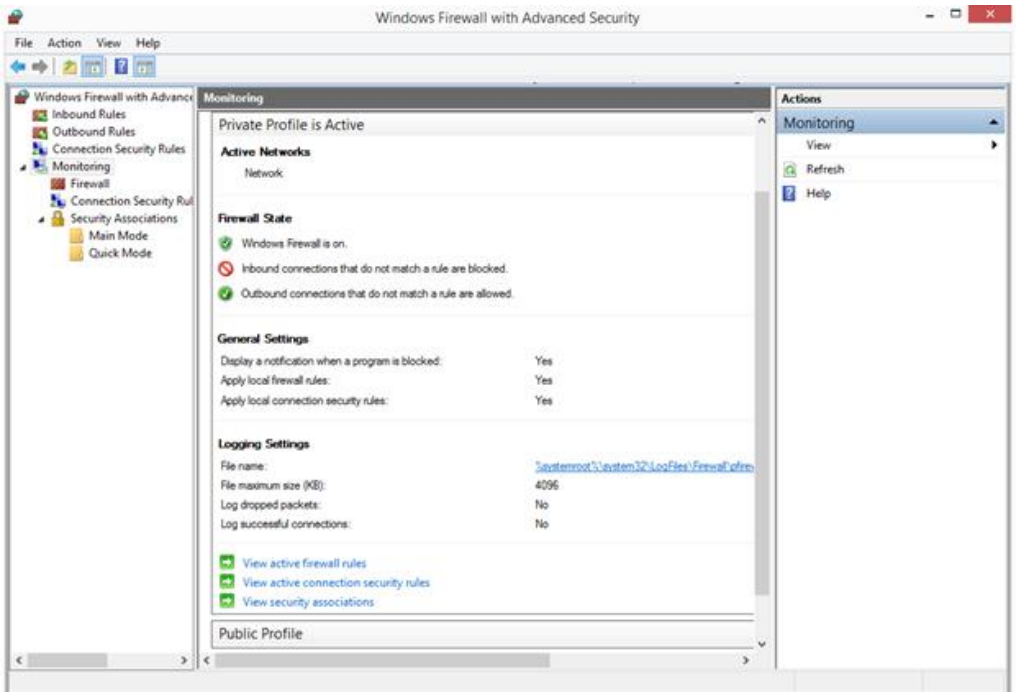
The rules with the green checkmark are enabled, meaning that they are used by Windows Firewall. Those with a gray checkmark are disabled and they are not used by Windows Firewall. The following are the parameters that can be edited:

- **Name**: The name of the rule you are viewing.
- **Group:** The group the rule belongs to. Generally, the group describes the app or the Windows feature the rule belongs to. For example, e.g. File and Printer Sharing, will have as a group name the feature they relate to.
- **Profile:** The network location/profile the rule is applied to: private, public, or domain (for business networks with network domains).
- **Enabled:** It tells you whether the rule is enabled and applied by Windows Firewall or not.
- **Action:** The action can "Allow" or "Block" based on what the rule is supposed to do.
- **Override:** Tells you whether that rule overrides an existing block rule. By default, all rules should have the value "No" for this parameter.
- **Program:** The desktop program the rule applies to.

- **Local address:** Tells you whether the rule is applied only when your computer has a specific IP address or not.
- **Remote address:** Tells you whether the rule is applied only when devices with specific IP addresses are connected or not.
- **Protocol**: Shares the network protocols for which the rule is applied.
- **Local port:** Tells you whether the rule is applied for connections made on specific local ports or not.
- **Remote port:** Tells you whether the rule is applied for connections made on specific remote ports or not.
- **Authorized users:** The user accounts for which the rule is applied (for inbound rules only).
- **Authorized computers:** Computers for which the rule is applied.
- **Authorized local principals:** The user accounts for which the rule is applied (for outbound rules only).
- **Local user owner:** The user account which is set as the owner/creator of the rule.
- **Application package:** This applies only to apps from the Windows Store and it shares the package name of the app the rule applies to.

4] MONITORING SECTION

Beneath the before mentioned 3 rules lies the monitoring section. On expanding it, you can view the active firewall rules, the active connection security rules, and view the active security associations.

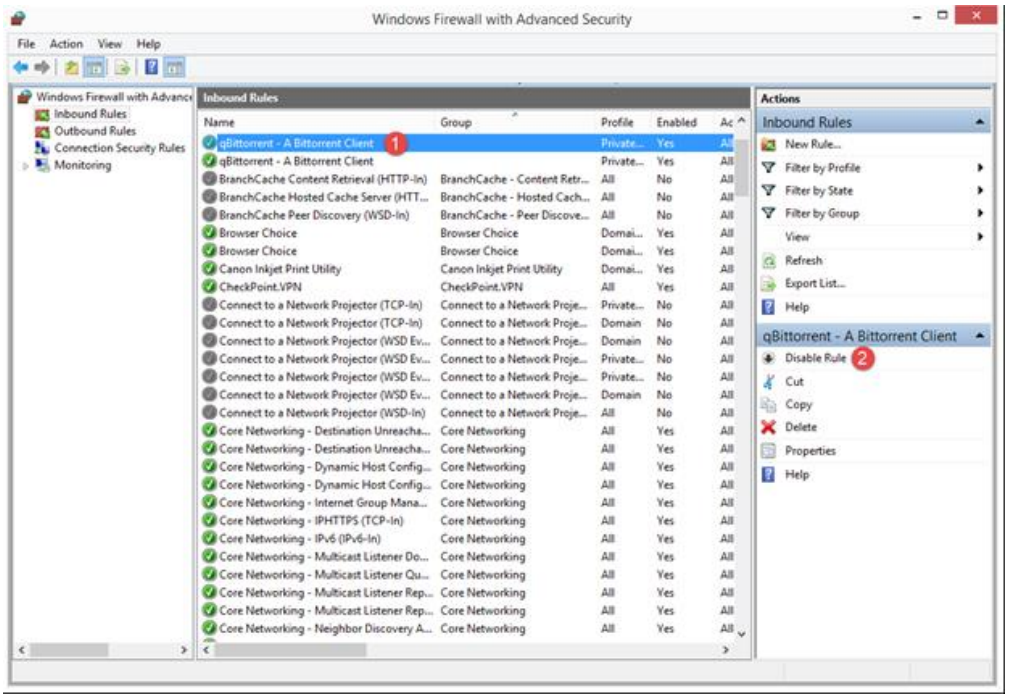


The monitoring section has the active networks, the state of our windows firewall and certain general and logging section. A security association is the information maintained about a secure encrypted channel on the local computer or device, so that this information can be used for future network traffic to a specific remote computer or device. It contains 2 sections namely Main mode and Quick mode. Here you can view which peers are currently connected to your computer and which protection suite was used by Windows to form the security association.

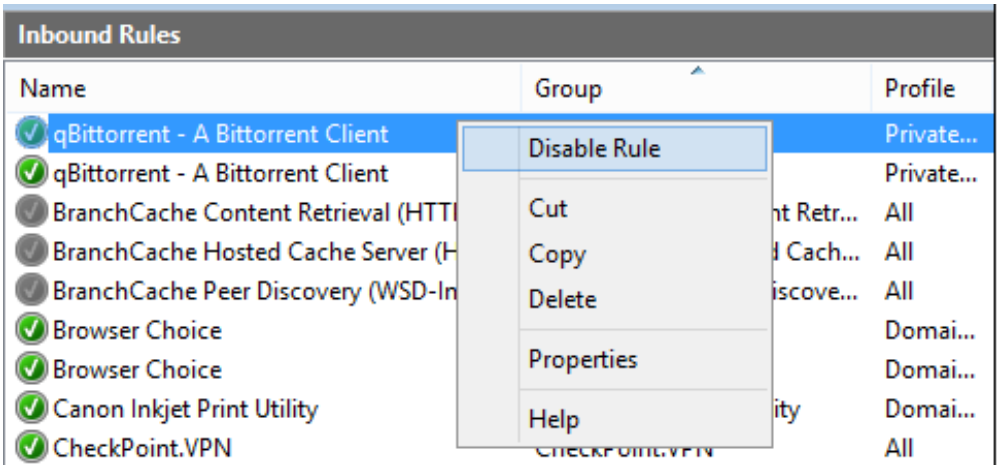
5] MANAGING EXISTING WINDOWS FIREWALL RULES

It is always better to disable a rule than delete it, so that if anything ill-advised happens, it could be reverted by re enabling the disabled rule.

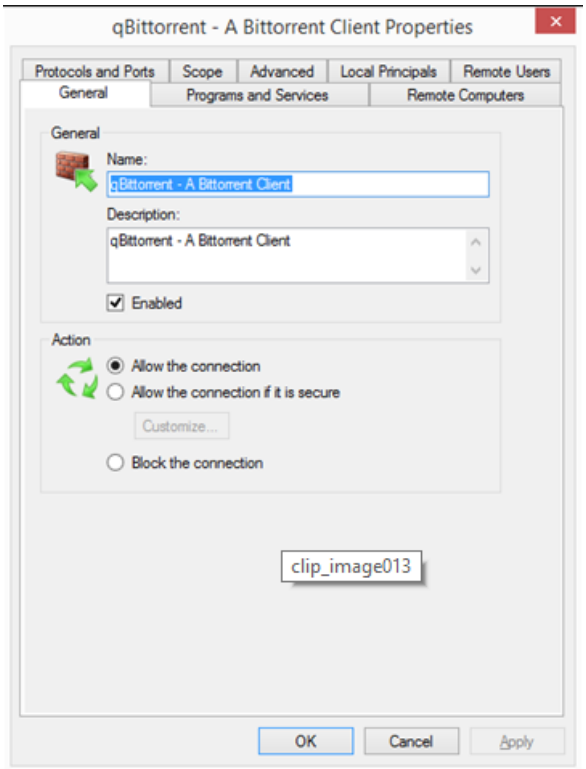
- To disable a rule, first select it and then press “Disable Rule” on the column on the right.



- The alternate method is, right click on the selected rule and then press "Disable Rule".



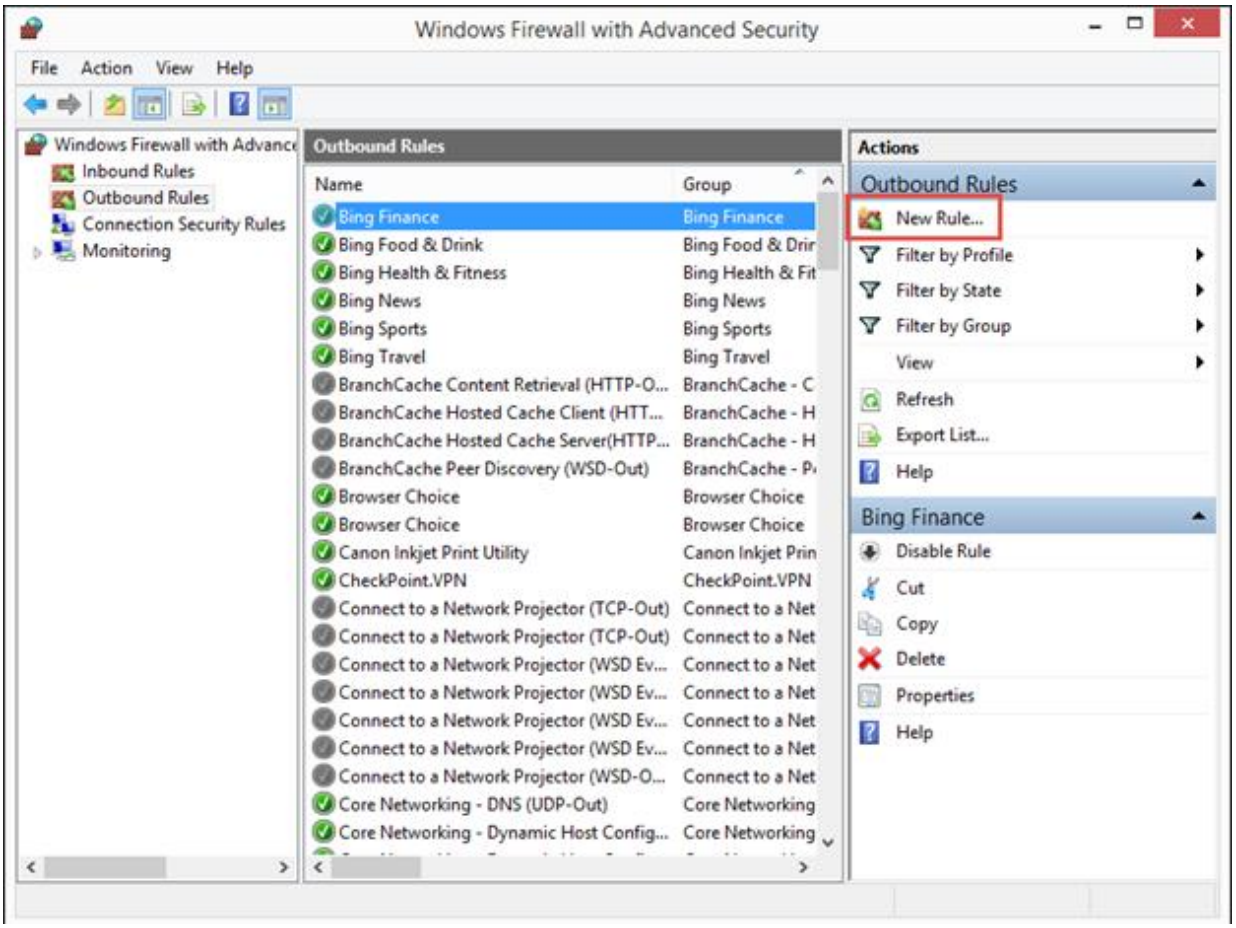
- If you want to edit a rule and the way it works, you can do so by double-clicking on it, selecting it, and then pressing "Properties" in the column on the right or right-clicking on it and selecting "Properties."



- All the parameters we have mentioned earlier can be modified in the "Properties" window of that rule. After making the required changes, click on OK.

6] CREATION OF OUTBOUND RULES FOR THE WINDOWS FIREWALL

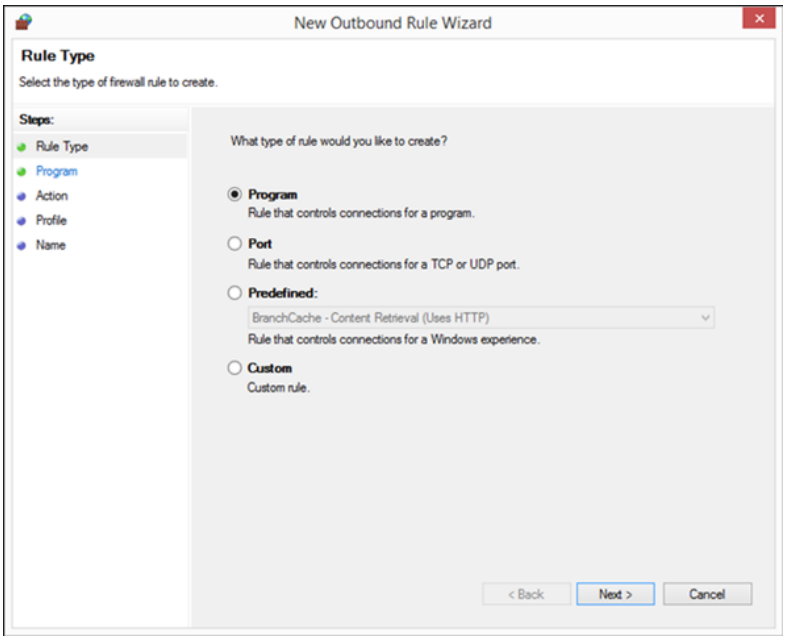
- To illustrate, let's create an outbound rule that blocks access to the network and the Internet for Skype, only when you are connected to untrusted public networks. To do this, go to "Outbound Rules" and press "New Rule" in the column on the right.



Now, you are asked to select the type of rule you want to create. Your choices are:

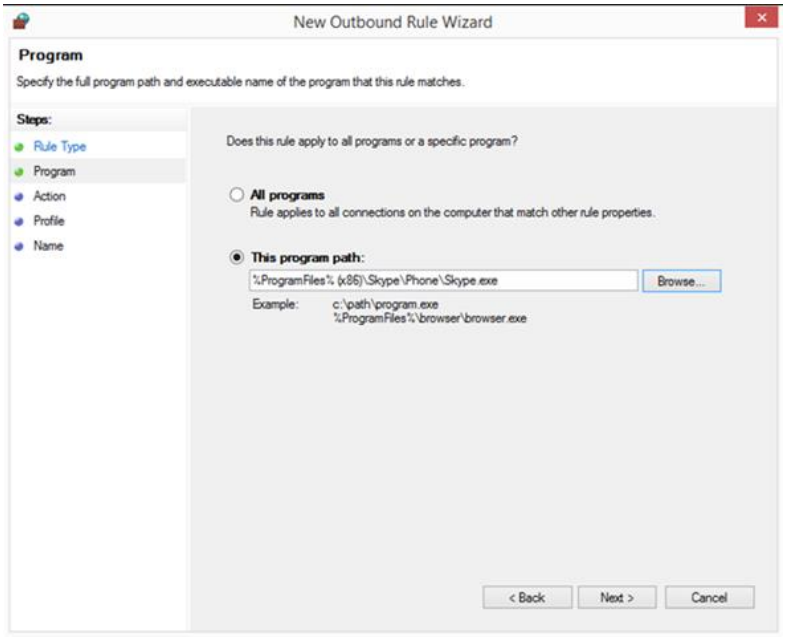
- **Program** – the rule applies to a specific program
- **Port** – the rule applies to the network traffic that is performed through a specific port
- **Predefined** – rule that controls the connections performed by a specific Windows service or feature
- **Custom** – a custom rule that can block both programs and ports or a specific combination of both.

For our example, we have selected “Program” and pressed “Next.”



Depending on what you have chosen at the previous step, you are now asked to select the program or the ports that you want to add to the rule.

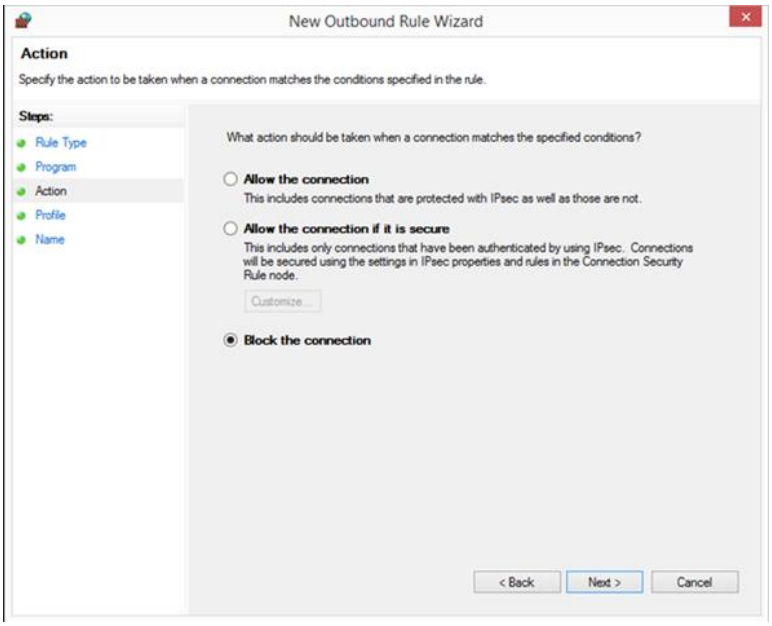
For our example, we have selected the executable of the program that we want to block – Skype.exe. When you've finished setting things up, press "Next."



Next, is to specify the action to be taken:

- **Allow the connection** – this includes both secure and insecure connections
- **Allow the connection if it is secure** – the connection is allowed only if it is made through a secure channel. You can specify the kind of authentication and encryption you want applied by pressing "Customize"
- **Block the connection** – blocks the connection, whether it is secure or not

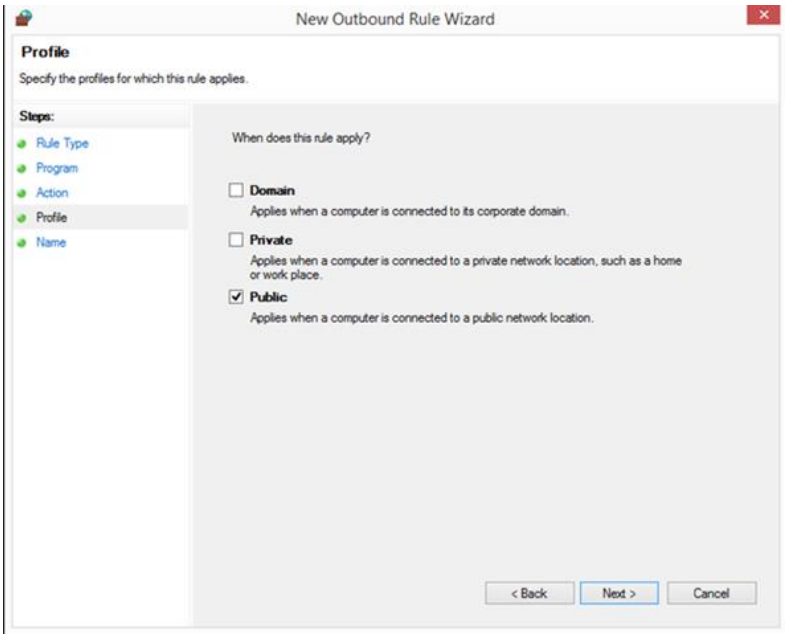
For our example we have selected "Block the connection" and pressed "Next."



Now you are asked to select when apply the rule. This means the network location when the rule is applied:

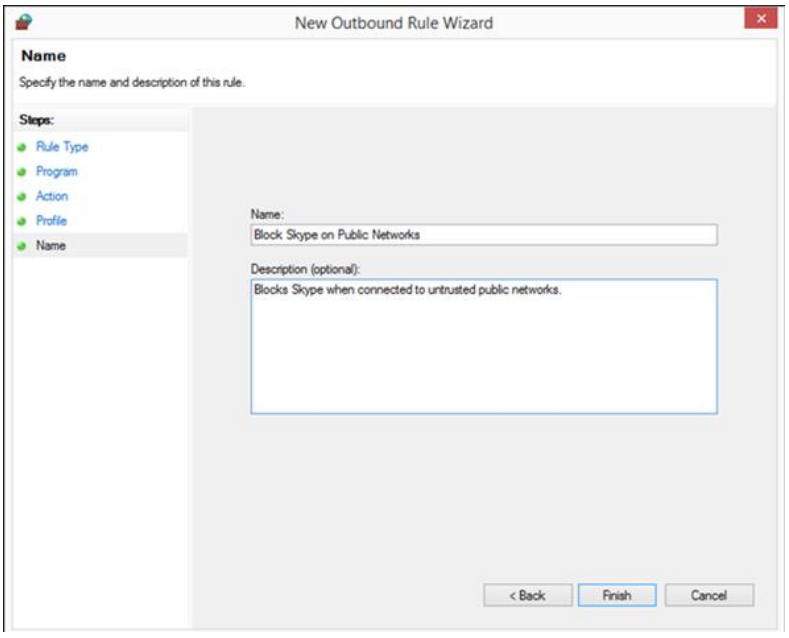
- **Domain** – the rule is applied only when the computer is connected to a network domain
- **Private** – the rule is applied only when the computer is connected to trusted private networks
- **Public** – the rule is applied only when the computer is connected to untrusted public networks

For our example we have chosen "Public" because we wanted to block access only when the computer is connected to untrusted public networks.



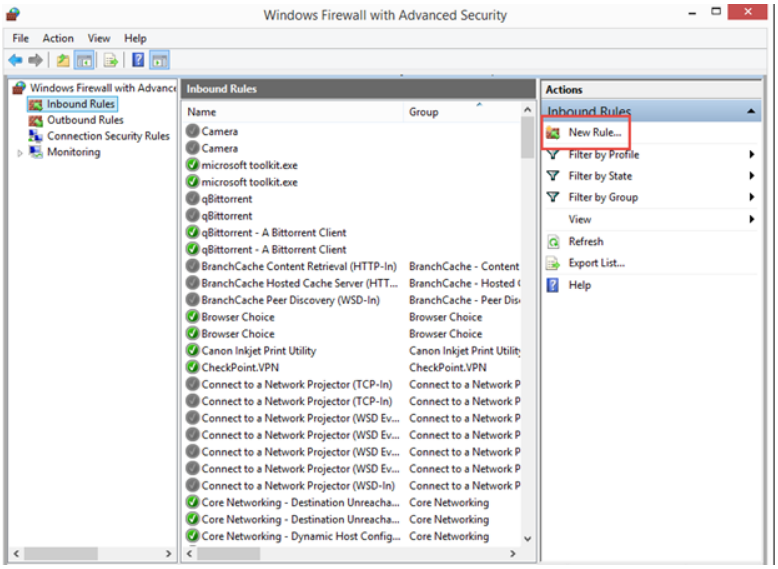
After clicking next, you are asked to enter a name and a description for the newly created rule. Always be sure to make this name as descriptive as possible, so that you can recognize the rule when you need to edit.

Press "Finish" and the rule is created and used by the Windows Firewall.

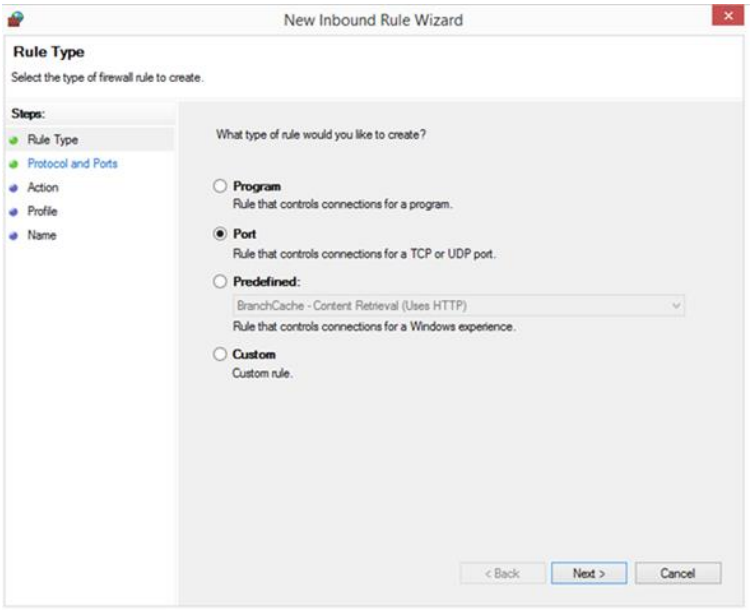


7] CREATION OF INBOUND RULES FOR THE WINDOWS FIREWALL

In Windows Firewall with Advanced Security, go to “Inbound Rules” and press “New Rule” in the column on the right.

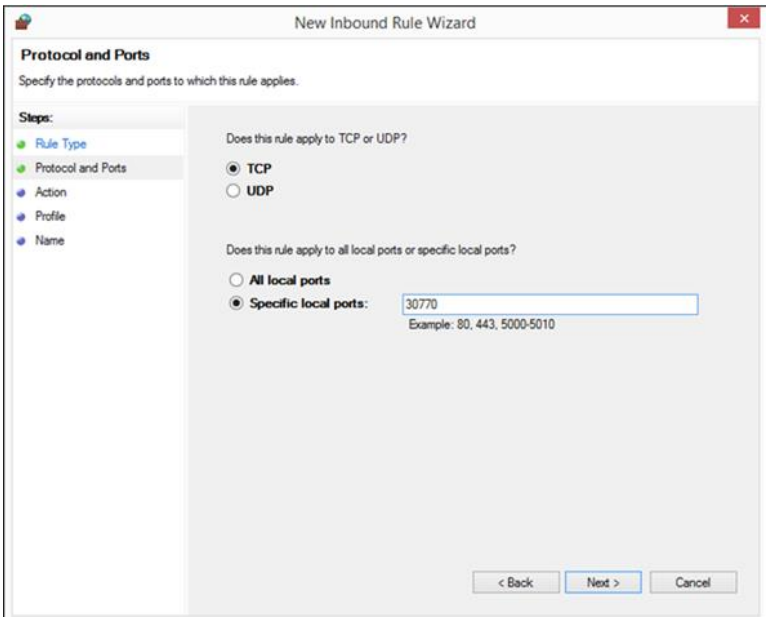


- This starts up the “The New Inbound Rule Wizard”. The options it displays are almost the same as the “New Outbound Rule Wizard”.



In this example, we have created a rule which blocks all inbound traffic made using the TCP protocol on the port 30770. At the first step we selected "Program" and pressed "Next."

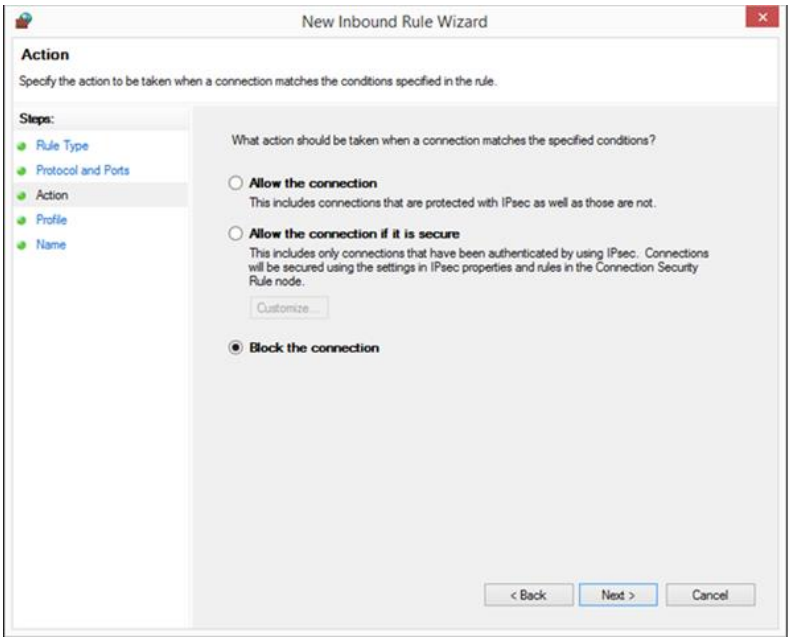
- We are asked to select the protocol for which the rule applies and the port. The choices for protocols are TCP and UDP. Then, we had the choice to block all ports or only specific ones. We selected "Specific local ports", entered "30770," and pressed "Next."



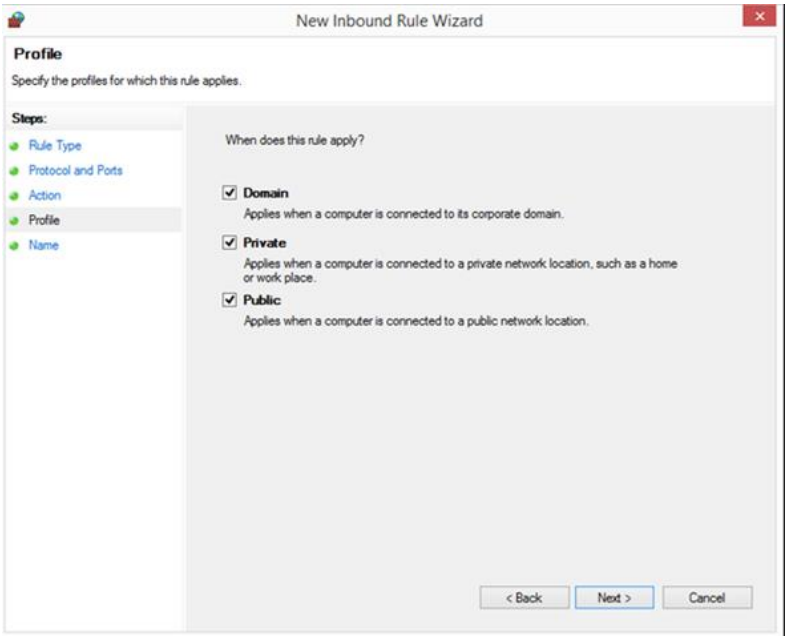
Now you are asked to select what action to take when a connection matches the conditions specified earlier. The 3 options that are available for selection are:

- **Allow the connection.**
- **Allow the connection if it is secure.**
- **Block the connection.**

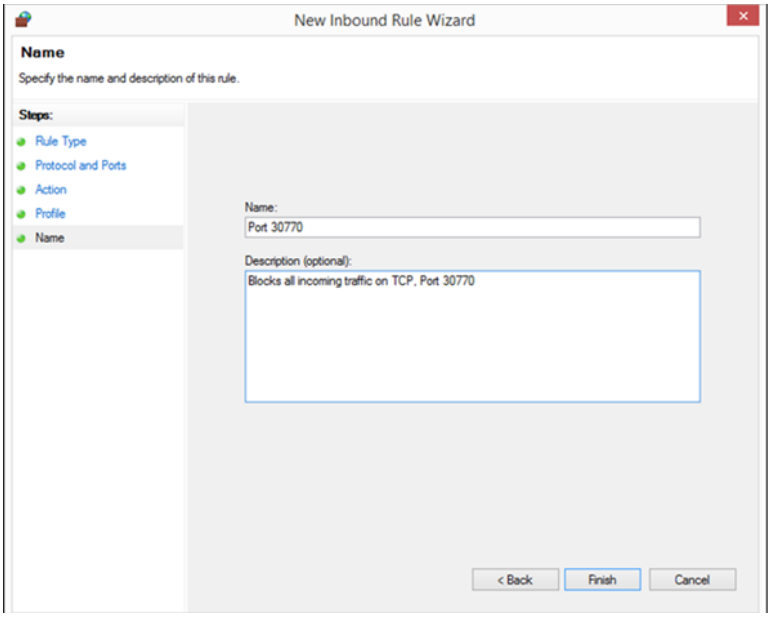
For our example, we have chosen "Block the connection" and pressed "Next."



Now you have to select the network locations for which the rule applies. Since we wanted to block all TCP traffic on port 30770, we selected all three locations and pressed "Next."



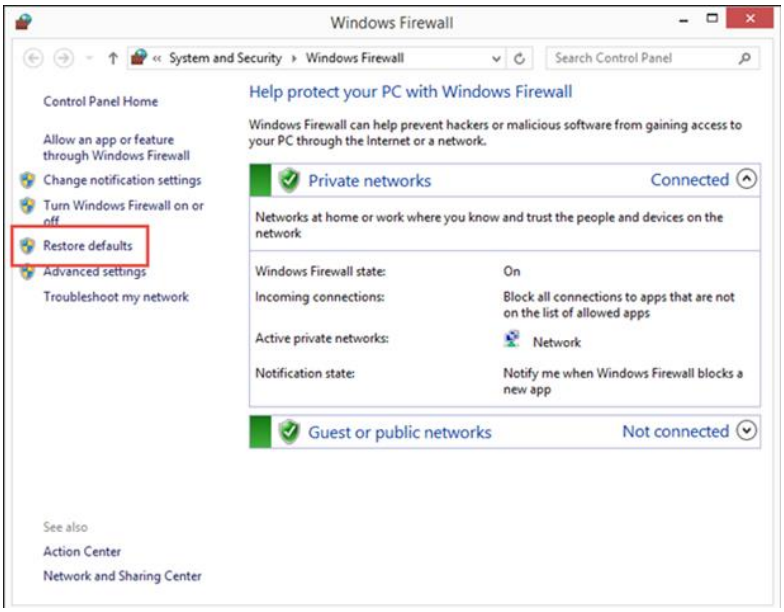
Finally, enter the name and the description for the newly created rule and press “Finish.”



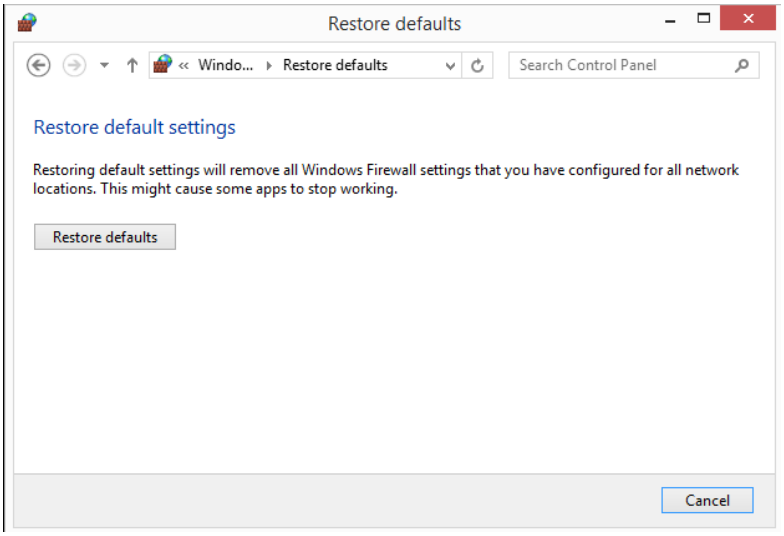
8] RESTORING WINDOWS FIREWALL TO ITS DEFAULT

If you have played too much with the rules in Windows Firewall and things have started to work incorrectly, you can easily undo all your settings and restore Windows Firewall to its defaults. This can be done only for an administrator account.

To do this, open the Windows Firewall and from the left column, click or tap “Restore defaults.”



You are now informed of what this resetting will do, when you're ready, press "Reset defaults."



Finally, you are asked to confirm that you are okay to go ahead with the reset. Click on yes and you are back to the default settings of the Windows Firewall.

REFERENCES

- https://www.owasp.org
- https://portswigger.net/burp/
- https://www.wikipedia.org/
- http://www.dvwa.co.uk/
- http://www.howtogeek.com/
- https://technet.microsoft.com/en-us/