Exploitation of Content Management System Vulnerabilities to Launch Large Scale Cyber Attacks

A PROJECT REPORT

Submitted by

MANOJ KUMAR SWAMI (13UECS0062)
ANKIT PANDEY (13UECS0011)

In fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

Department of Computer Science and Engineering

VeltechDr.RR& Dr.SR University
(Estd. u/s 3 of UGC Act, 1956)
Apr, 2017

Veltech Dr.RR & Dr.SR University
(Estd. u/s 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

This is to certify that the project entitled **“Exploitation of Content Management System Vulnerabilities to Launch Large Scale Cyber Attacks”** submitted by MANOJ KUMAR SWAMI (13UECS0062), ANKIT PANDEY (13UECS0011) in partial fulfillment for the requirements for the award of Bachelor of Technology Degree in Computer Science and Engineering is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the project report has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Signature of Supervisor

Mr. R. Vinoth Kumar, M.Tech.,
Asst. Professor,
Department of CSE,
Veltech Dr.RR & Dr.SR University,
Avadi, Chennai-600062

Signature of Head of the Department

Dr.N.Malarvizhi, Ph.D.,
Professor,
Department of CSE,
Veltech Dr.RR & Dr.SR University,
Avadi, Chennai-600062.

Submitted for the partial fulfillment for the award of the degree of Bachelor of Technology In Computer Science and Engineering from VELTECH Dr. RR & Dr.SR UNIVERSITY, 400 feet Outer Ring Road, Avadi, Chennai– 600062

CERTIFICATE OF EVALUATION

COLLEGE NAME :VELTECH Dr.RR& Dr.SR UNIVERSITY

BRANCH : COMPUTER SCIENCE AND ENGINEERING

SEMESTER : VIII

S. No.	NAME OF THE STUDENT	TITLE OF THE PROJECT	NAME OF THE PROJECT SUPERVISOR
1.	MANOJ KUMAR SWAMI	EXPLOITATION OF CONTENT MANAGEMENT SYSTEM VULNERABILITIES TO LAUNCH LARGE SCALE CYBER ATTACKS	Mr. R. VINOTH KUMAR, M.TECH
2.	ANKIT PANDEY		

The report of the Project submitted by the above student in partial fulfilment for the award of Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING of VELTECH DR.RR & DR.SR UNIVERSITY for the viva-voce examination held at VELTECH DR.RR & DR.SR UNIVERSITY on **_______________**, has been evaluated and confirmed to be reports of the work done by the above student.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO). DSc., Chancellor and Foundress President Dr. R. SAKUNTHALA RANGARAJAN M.B.B.S.**, Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. BEELA SATYANARAYANA B.E(MECH), M.E(MD), M.E(IE), M.Tech(CSE), Ph.D. (IIT, Delhi),** for providing us with an environment to complete our project successfully.

We are very thankful to our beloved **Pro-Vice-Chancellor Dr. U.CHANDRASEKHAR Ph.D.,** for providing us with an environment to complete our project successfully.

We obligated to our beloved **Registrar Dr.E.KANNAN M.E., Ph.D.,** for providing immense support in all our endeavors.

We thankful to our esteemed **Director Academics Dr. ANNE KOTESWARA RAO, Ph.D.,** for providing a wonderful environment to complete our project successfully.

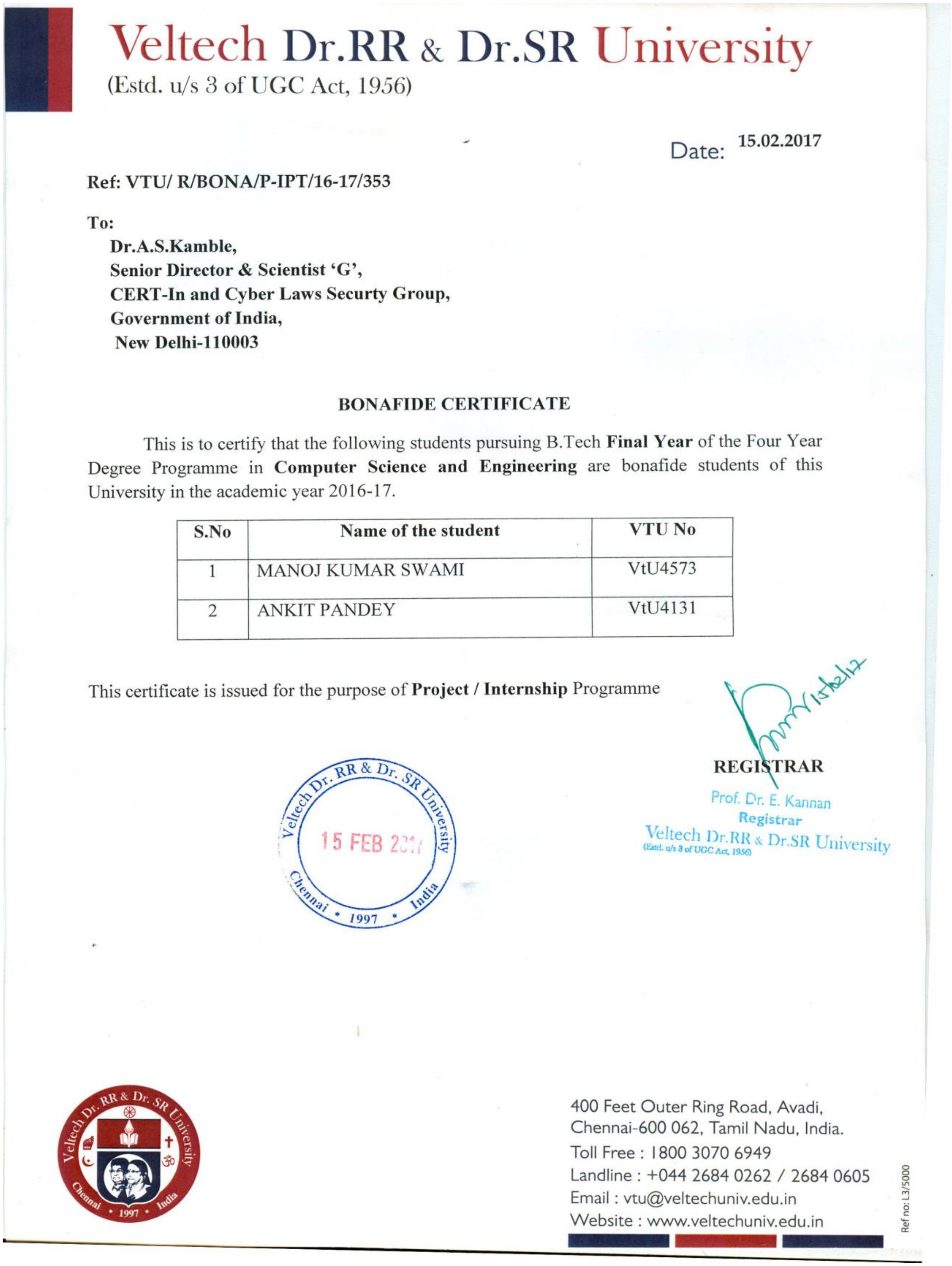
We extremely thankful and pay my gratitude to our **Dean Dr. M.M.Naidu Ph.D** for his valuable guidance and support on completion of this project in it presently.

We record indebtedness to our **Head of the Department. Dr. N.Malarvizhi Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

A special thanks to our **Project Coordinator Dr. Arthi Ph.D and Mr. N.Rajkumar M.E.,** for thier valuable guidance and support throughout the course of the project.

We also take this opportunity to express a deep sense of gratitude to Our **Internal Guide Mr. R. Vinoth Kumar, M.TECH,** for his cordial support, valuable information and guidance, He helped us in completing this project through various stages.

We thank our department faculty, supporting staffs, parents, and friends for their help and guidance to complete this project.



Veltech Dr.RR & Dr.SR University

(Estd. u/s 3 of UGC Act, 1956)

Date: 15.02.2017

Ref: VTU/ R/BONA/P-IPT/16-17/353

To:

Dr.A.S.Kamble,
Senior Director & Scientist 'G',
CERT-In and Cyber Laws Securty Group,
Government of India,
New Delhi-110003

BONAFIDE CERTIFICATE

This is to certify that the following students pursuing B.Tech **Final Year** of the Four Year Degree Programme in **Computer Science and Engineering** are bonafide students of this University in the academic year 2016-17.

S.No	Name of the student	VTU No
1	MANOJ KUMAR SWAMI	VtU4573
2	ANKIT PANDEY	VtU4131

This certificate is issued for the purpose of **Project / Internship** Programme

15/2/17

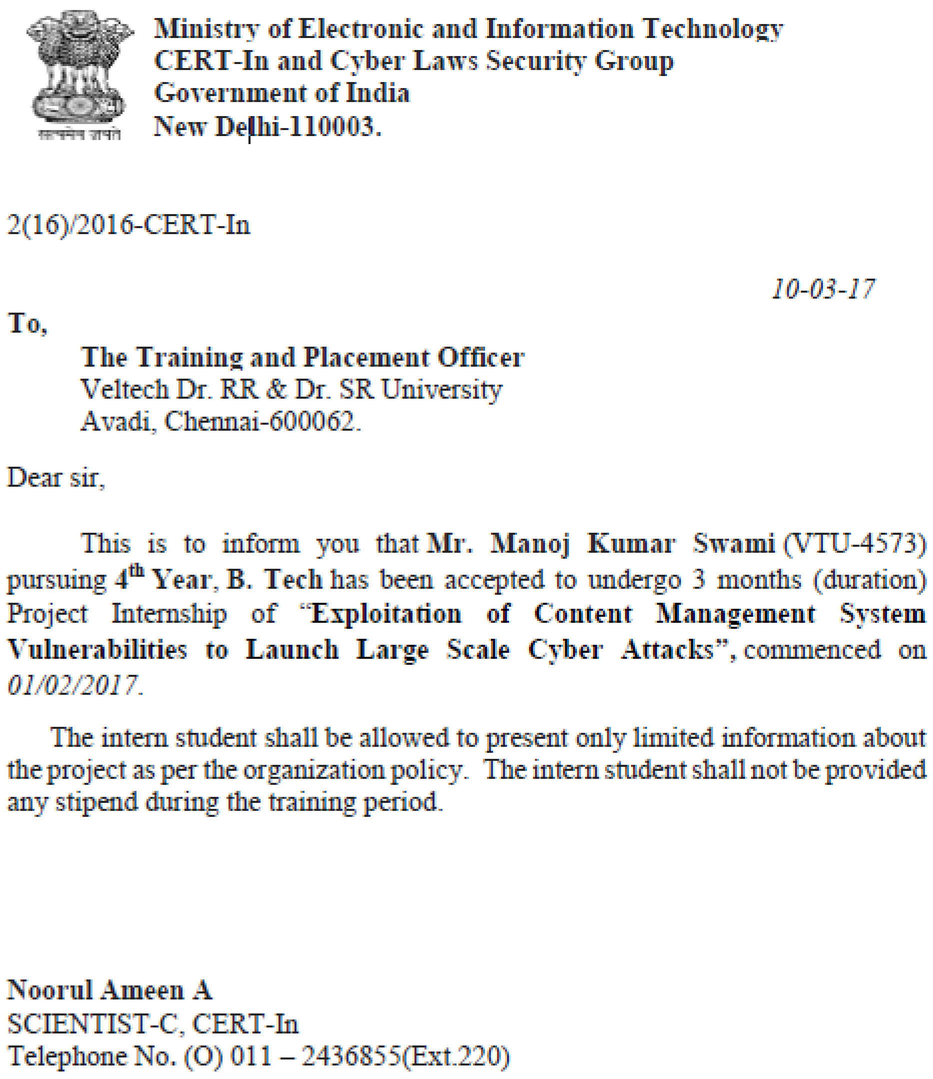
REGISTRAR

Prof. Dr. E. Kannan
Registrar
Veltech Dr.RR & Dr.SR University
(Estd. u/s 3 of UGC Act, 1956)

Veltech Dr. RR & Dr. SR University
15 FEB 2017
Chennai • 1997 • India

400 Feet Outer Ring Road, Avadi,
Chennai-600 062, Tamil Nadu, India.
Toll Free : 1800 3070 6949
Landline : +044 2684 0262 / 2684 0605
Email : vtu@veltechuniv.edu.in
Website : www.veltechuniv.edu.in

Ref no: L3/5000



Ministry of Electronic and Information Technology
CERT-In and Cyber Laws Security Group
Government of India
New Delhi-110003.

2(16)/2016-CERT-In

10-03-17

To,

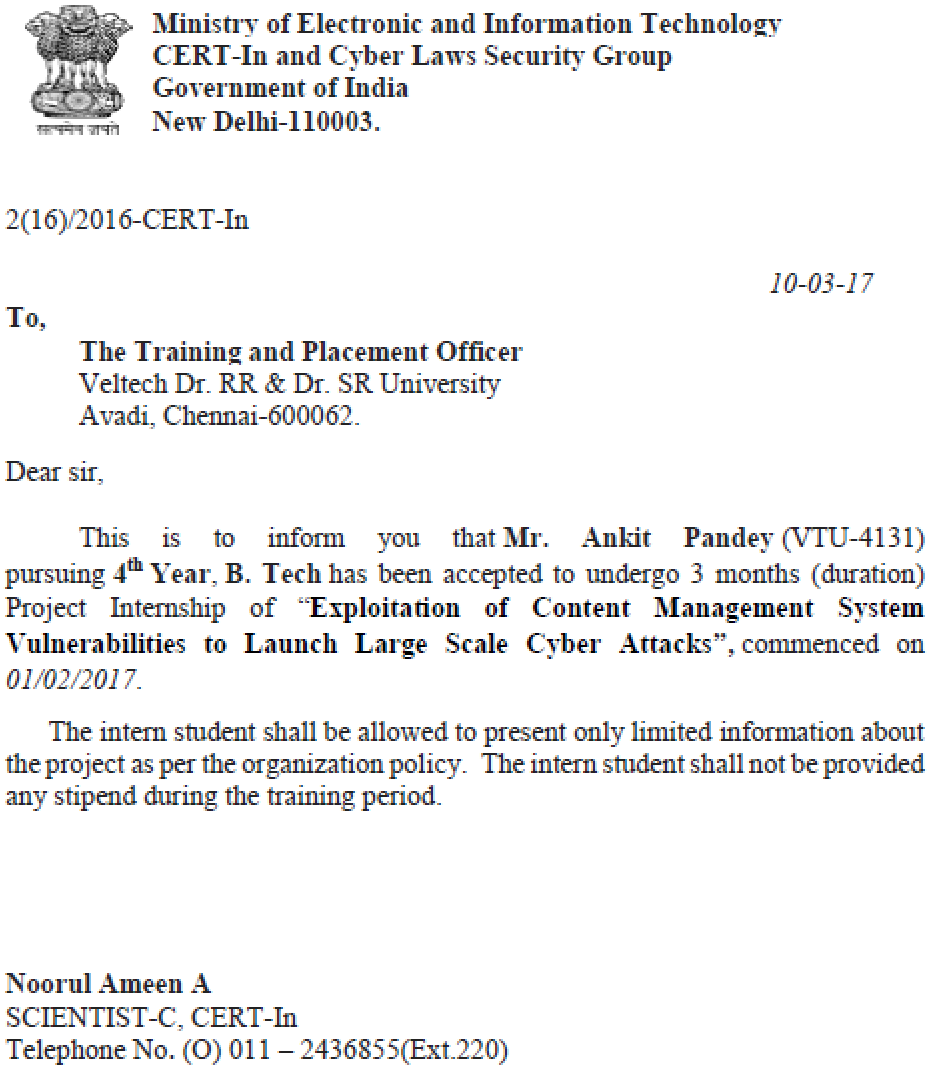
The Training and Placement Officer
Veltech Dr. RR & Dr. SR University
Avadi, Chennai-600062.

Dear sir,

This is to inform you that **Mr. Manoj Kumar Swami** (VTU-4573) pursuing **4th Year**, **B. Tech** has been accepted to undergo 3 months (duration) Project Internship of "**Exploitation of Content Management System Vulnerabilities to Launch Large Scale Cyber Attacks",** commenced on *01/02/2017*.

The intern student shall be allowed to present only limited information about the project as per the organization policy. The intern student shall not be provided any stipend during the training period.

Noorul Ameen A
SCIENTIST-C, CERT-In
Telephone No. (O) 011 – 2436855(Ext.220)



Ministry of Electronic and Information Technology
CERT-In and Cyber Laws Security Group
Government of India
New Delhi-110003.

2(16)/2016-CERT-In

10-03-17

To,

The Training and Placement Officer
Veltech Dr. RR & Dr. SR University
Avadi, Chennai-600062.

Dear sir,

This is to inform you that **Mr. Ankit Pandey** (VTU-4131) pursuing **4th Year**, **B. Tech** has been accepted to undergo 3 months (duration) Project Internship of "**Exploitation of Content Management System Vulnerabilities to Launch Large Scale Cyber Attacks",** commenced on *01/02/2017.*

The intern student shall be allowed to present only limited information about the project as per the organization policy. The intern student shall not be provided any stipend during the training period.

Noorul Ameen A
SCIENTIST-C, CERT-In
Telephone No. (O) 011 – 2436855(Ext.220)

ABSTRACT

Content Management Systems are mostly used for website development. Websites are the main targets of various malicious attackers, and therefore it is necessary to be knowledge about the security level of websites. This will describe some features of the well-known open source Content Management System platforms: Drupal, WordPress and Joomla. The security of such web applications depends as much on vulnerabilities found in plugins as it does in vulnerabilities in the application core. One of the biggest problems facing the IT security industry is the use of vulnerabilities in legitimate software to launch malware attacks. Malicious programs can use these vulnerabilities to infect a Content of Websites without attracting the attention of the user and in some cases, without triggering an alert from security software. The cyber-attacks, or more specifically denial of service attacks, were launched by the Cyber fighters of Izz Ad-Din Al Qassam also known as Qassam Cyber Fighters. Operation Ababil was one of a series of cyber-attacks starting in 2012, targeting various American financial institutions and carried out by a group of Izz Ad-Din Al Qassam.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

ACRONYM	ABBREVATION
IEEE	Institute of Electrical and Electronics Engineers
CMS	Content Management System
GPL	General Public Licence
PHP	Hypertext Pre-Processor
XSS	Cross-Site Scripting
CSRF	Cross-Site Request Forgery
SQL	Structured Query Language
UML	Unified Modelling Language
OWASP	Open Web Application Security Project
DOS	denial of service
NMAP	Network Mapper
DDOS	Distributed Denial of Service Attack
LOIC	Low Orbit Ion Canon
HOIC	High Orbit Ion Canon
HTTP	Hyper-Text Transfer Protocol
DVWA	Damn Vulnerable Web App

INDEX

CHAPTER-1

INTRODUCTION

1.1. Overview of the Project:

Content management systems (CMS) provide a customizable multi-user environment with a default set of functions that are needed to create, organize and maintain web content. One of the main features of CMSs is the modularity which allows users to add various functions that are appropriate for their needs. First CMSs appeared in late 1990s, but they have become popular in middle 2000s. There are over 200 CMSs developed in various programming languages such as PHP, Java, Perl, .NET and others, divided in open-source and proprietary CMSs, each one with its own supported databases (eg. MySQL, SQLite, Oracle, PostgreSQL).

The **Content Management System** is a Web-Based application that provides capabilities for multiple users with different permission levels to manage content, data or information of a website.

Examples:

- Create, Edit, Publish, Archive web pages
- Create, Edit, Publish, Archive articles
- Create, Edit, Publish, Archive press releases
- Create, Edit, Publish, Archive blogs
- Add / Edit events into an Event Calendar
- Add / Edit products, description, product specifications, prices, photos, etc.
- Enter, Edit, or View orders and print packing slips and invoices
- View reports and statistics site data
- Create and Edit system users which have different permission levels to different section(s) of the above administration

Content Management Systems (CMSs) are consistently targeted and leveraged to launch cyber-attacks. CMSs are software suites that allow site administrators to easily manage the design, functionality, and operation of websites with minimal technical expertise. In recent years, there has been an increase in the number of deployments of CMS software on the Internet. This has been fuelled by popular open source projects which are freely available under General Public License (GPL) model. Unfortunately, some CMS web server operators are not following security best practices, exposing them and others to cyber security risks such as compromise and denial of service.

Drupal and WordPress the most widely used CMSs in the world. It is PHP-based and allows rapid deployment of dynamic content on websites. It is recognized for its simplicity of deployment and usage while offering extensive features and plugins. However, like many other large software packages, Joomla! has been the subject of a number of vulnerabilities in recent years and, if left unpatched, can represent a risk for site owners, and any other Internet users. Content management systems are very relevant to knowledge management (KM) since they are responsible for the creation, management, and distribution of content on the intranet, extranet, or a website. Content management is a discipline in itself, so this section will be relatively brief, only outlining the basic considerations.

One of Drupal's most popular features is the Taxonomy module, a feature that allows for multiple levels and types of categories for content types. And you can find plenty of professional Drupal Themes, which are ready to be customized and worked with. You can also grab Drupal Plugins. Drupal also has a very active community powering it, and has excellent support for plugins and other general questions.

Some Popular websites using these CMSs are:

1. WordPress

The New York Times, CNN, Forbes, Reuters, UPS, Ebay, TechCrunch.

2. Drupal

The White House, Linux, University of Minnesota, Red Hat, Timex, The Economist, Georgetown University.

3. Joomla

Peugeot, Harvard University, The Hill, MTV Greece, The Fashion Spot.

1.2. Vulnerabilities:

If we take a look at the vulnerabilities distribution by type, as shown in Figure 2, then we see that cross-site scripting (XSS) accounts for the most common vulnerabilities with an average of 65%. However, code execution with an average of 41%, and SQL injection with an average of 34%, are also quite common.

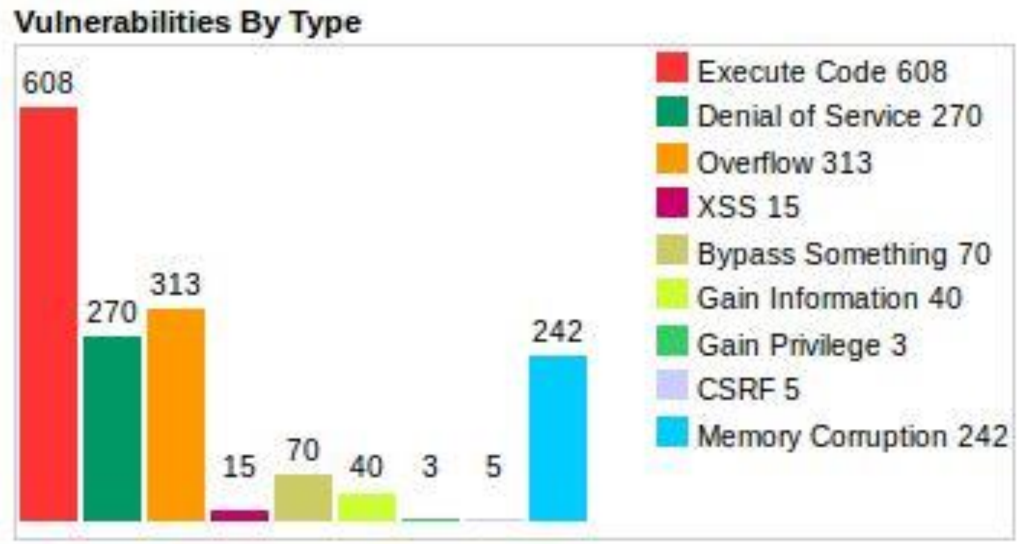


Figure 1. Vulnerabilities by Type

CMS should never be run in its default configuration and should be upgraded whenever newer versions become available. Sometimes changing the administration folder is enough to deceive your average script kiddies and prevent them from accessing the system as they are more likely to go after the low-hanging fruits that they can find with their scanners.

1.3. Problem Statement:

Most used CMS in world today are WordPress, Joomla and Drupal which are all open source software built on PHP with MySQL database support primarily. WordPress alone is used by more than 23.3% of the top 10 million websites as of January 2017 with around 60% of CMS market share and more than 60 million websites altogether. Joomla and Drupal together take around 10% of CMS market share. We'll analyse main causes of security problems in CMS in general and compare their security with enterprise CMS solutions. Since CMS security dominantly depends on security of underlying systems (server, database server and web server) we will briefly cover the comparison between three most used CMS (Linux server, MySQL or Maria DB as database servers and Apache as web servers) with underlying Microsoft technology for running CMS (Windows Server, MS SQL as database server and .NET based CMS running on IIS as web server). We'll give analysis of most abused and most severe cases of vulnerabilities and try to identify most common ways in which they are being exploited. Analyses importance of community in preventing widespread abuse of newly discovered vulnerabilities. Analyses effectiveness of CDN 2 and their ability to eliminate standard threats. Final goal of this project is to securing open source CMS.

1.4. Types of CMS:

Most popular Open Source Content Management System are:

1. **WordPress**
2. **Drupal**
3. **Joomla**

1. **WordPress:** Ideal for personal websites, blogs, small businesses, and those with the least technical and no desire to learn any programming skills, WordPress is a great place to start. It has the most available free themes (templates) and plugins available today. Originally developed as a blogging tool, it is now used commonly to create complicated websites. If you like to hack and code, WordPress can still indulge even the hardcore programmer in you.
2. **Drupal:** Drupal is another CMS that has a very large, active community. Instead of focusing on blogging as a platform, Drupal is more of a pure CMS. The White House ditched its proprietary CMS a few years ago and now uses Drupal. Other companies using Drupal include AT&T, McDonald's, Duke and Sandford Universities, Symantec, and Linux Foundation, etc.
3. **Joomla:** Olympus, Porsche, Sprint, and Vodafone are just a few major corporations that use Joomla. Unlike WordPress, Joomla was never designed as a blogging software, but works fine for blogs, too. Joomla takes a little bit more time to set up than WordPress but is overall more powerful and still easier than the mighty Drupal.

   There are plenty of options when it comes to picking a content management system for a development project. Depending on how advanced you need the CMS to be, what language it's built in, and who is going to be using it, it can be a nightmare trying to find the "perfect" CMS for a project.

However, some CMSs have a slight edge over the rest of the competition because of the usability of the software. Some are just easier to install, use and extend, thanks to some thoughtful planning by the lead developers.

1.5. Security Issues:

- **Why are CMS platforms so vulnerable?**

CMS are vulnerable by nature because they are built on open source frameworks. Such shared development environments offer several benefits but they also have their share of flaws, many of which arise form a lack of accountability. With no price tag, and with no one to take direct responsibility for potential problems, it’s no surprise when the final product has some security issues. Since the top CMS are so popular, these security vulnerabilities are actively sought after both by security researchers and members of the hacker community.

Once identified, these flaws can turn into a virtual gold mine for hackers, creating a much more efficient way for them to execute automated large-scale cyber-attacks.

Adding to the issue are website operators who use weak passwords, leaving their admin accounts vulnerable to automated brute force attacks.

Open-source CMS solutions such as WordPress, Joomla and Drupal, have grown very popular amongst many developers. Open-source CMS solutions are seen as affordable, flexible and quick solutions to building a web presence. Unfortunately, the security of this type of software is very average and can result in users having to rebuild their websites, patch up problems that may reoccur or even opt for the safer approach of using a closed or proprietary source CMS solution.

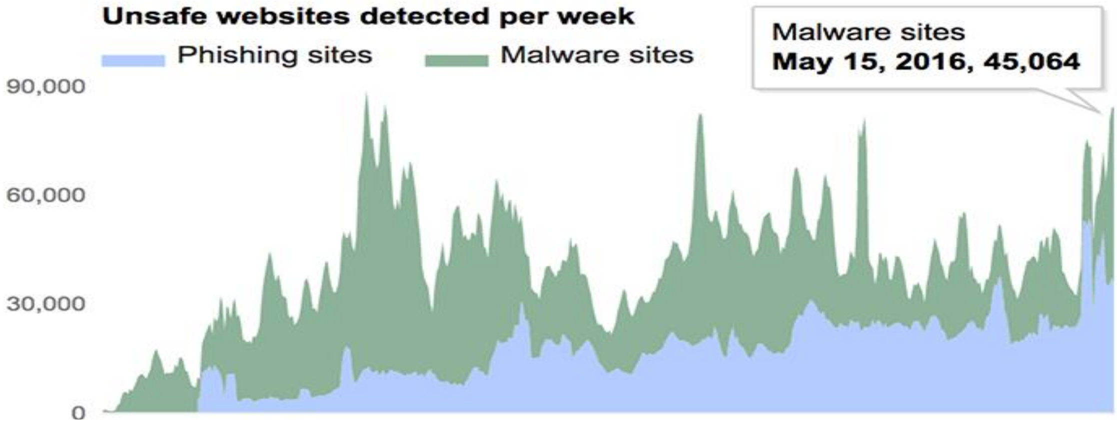


Figure 2. Malicious Sites

2.1. Existing System:

One of the biggest problems facing the IT security industry is the use of vulnerabilities in legitimate software to launch malware attacks. Professional hackers, either working on their own or employed by the government or military service, can find computer systems with vulnerabilities lacking the appropriate security software. Once found, they can infect systems with malicious code and then remotely control the system or computer by sending commands to view content or to disrupt other computers.

Vulnerability factor exploits how vulnerable an organization or government establishment is to cyber-attacks. An organization can be vulnerable to a denial of service attack, and a government establishment can be defaced on a web page.

2.2. Disadvantages of using a CMS

- Potential to break your websites look and feel if not used properly.
- Can require extra maintenance, backups, upgrades, security patches.
- The overhead on server resources.
- So many things can go wrong, formatting errors, incorrect preparation of images, no image compression, inconsistent resizing resulting in out of proportion photographs, breaking away from the 'style guide' of your website that you may have paid a designer a lot of money to create for you, thereby effecting the consistency of your brand. Most of this can be mitigated by proper training or indeed good advice on which CMS to use and as they say time heals all things. After some initial teething trouble, most of these issues can be ironed out eventually.
- You may not have the resource to update website regularly
- There is no point paying for a CMS to be implemented, that you don't intend to use very often and more importantly one that you don't have the resources; time, staff and skills to use effectively.
- Using a CSM effectively can require certain computer skills that you or your staff may not have Training will cost money so think about offsetting this against a retainer on your developer. Both options are well worth consideration.

2.3. Purposed System:

Most used Content Management System in world today are Word Press, Joomla and Drupal which are all open source software built on PHP with My SQL database support primarily. We'll analyze main causes of security problems in CMS in general and compare their security with enterprise CMS solutions. Since CMS security dominantly depends on security of underlying systems (server, database server and web server) we will briefly cover comparison between usual underlying technology of three most used CMS (My SQL, Maria DB or PostgreSQL as database servers and Apache (Linux server as web servers). It has been shown that the basic installations of the CMSs fail to ensure safety requirements against Cyber-attacks due to irresistance to some threats. Necessary software tools for the tested CMSs have been specified in order to ensure the resistance to threats which have not been provided in the basic installation of the tested CMSs.

2.4. Advantages of using CMS

- Ability to add or edit pages on your website yourself
- It is nice to have control over your investment, the feeling of empowerment to control your website is a good one. In particular, for an organization or business that is dynamic and needs things to happen fast.
- Not have to pay your developer monthly maintenance or hourly rate for changes Why pay someone to do something you can do yourself right?
- Useful in organizations, with many content contributors, that perhaps need to audit additions and changes to content being made Many CMSs offer the ability to delegate roles and cascade these throughout the organization with some people writing content and others giving the OK. A CMS can be ideal for this type of workflow.
- The ability to leverage excellent third party plugins

CHAPTER-3

FEASIBILITY STUDY:

The following feasibilities are considered for the project in order to ensure that the project is variable and it does not have any major obstructions. Feasibility study encompasses the following things:

1. Technical Feasibility
2. Economic Feasibility
3. Operational Feasibility

3.1. Technical Feasibility:

Our project is technically feasible. It requires following technologies: -

- Python and Python Exploit Files.
- Virtual Machine to the get the complete access of the content on another machine.
- R57.php, C99.txt, C100.txt, mturk.php etc. to get down the target server machine.
  Copy these files into the target content server and the target machine will automatically go down.
- **SQL Injection:**

  An attacker can use SQL Injection to bypass authentication or even impersonate specific users. One of SQL's primary functions is to select data based on a query and output the result of that query. An SQL Injection vulnerability could allow the complete disclosure of data residing on a database server.

3.2. Economic Feasibility

For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. This project is completely feasible. It does not require any type of financial investment with respect to time. And the cost of new system used in this project as it is more than the investment and expenditure. It need more development tools.

3.3. Operational Feasibility:

For many years, the design of efficient and robust feature extraction and feature selection, especially band selection, algorithms have been the most important issue addressed by the remote sensing community. Strong efforts have been devoted to elaborate new band selection algorithms and improve techniques used to reduce dimensionality.

CHAPTER-4

REQUIREMENTS ANALYSIS

4.1. SOFTWARE RQUIREMENTS:

Table 1. Software Requirements

S. No.	DESCRIPTION	TOOLS
1.	Operating System	Windows 8.1 & Windows XP
2.	Apache & MySQL	XAMPP
3.	CMS	Drupal *v7.2*
4.	Virtual Machine	VMWare
5.	DDos Attacking Tools	HOIC, LOIC, Pyloris
6.	SQL Injection	Havij *v1.17*
7.	Malicious Files	R57, C99.php
8.	Network Analysis	Wireshark, Omnipeek
9.	Network Scanner	Nmap *v6.47*
10.	Testing	DVWA

4.2. HARDWARE REQUIREMENTS:

Table 2. Hardware Requirements

S. No.	DESCRIPTION	TOOLS
1.	Processor	Intel Core i5
2.	Memory	50GB
3.	RAM	8GB

4.3. DOCUMENTATION:

Table 3. Documentation

S. No.	DESCRITION	TOOLS
1.	Report	MS Word 2016 & PPT
2.	Diagrams	Rational Rose & Online

CHAPTER-5

SYSTEM DESIGNING

5.1. So, what exactly is CMS?

A CMS or a 'Content Management System' allows you to control and manage the content within your web site - without technical training. Using this uncomplicated system, you can very easily add, delete images and edit text in your web site on the fly.

Many companies find it difficult to keep their web site content as up to date as they would like. Often there are delays getting new content online, the site users and your clients get to see outdated information. That's why so many companies are turning to CMS.

- **CMS General Problems:**

General security problems with CMS are: easy remote identification of system, poor programming practices while creating plugins, lack of oversight while submitting plugins, poor inspection of potential security issues with plugins and lack of auto-update of CMS installations and plugins which are leaving system open for knows CMS and plugins vulnerabilities.

Once a CMS has been compromised, adversaries can exploit their access to:

- Obtain access to authenticated and privileged areas of the site
- Upload malware to the web server to facilitate remote access, for example: upload C99, C100, R57.php files in the publish area and site will goes down.
- Inject malicious content into web pages. This could be used to serve exploits or malware to visitors or to facilitate remote access to the infrastructure.
- If you run a popular site, there is a very good chance that you will get hacked by DOS (Denial of Service) Attack. The vulnerability makes use of a well-known cyber-attack, DOS Attack. When executed, it has the capability to take down the whole website or server almost instantly, with the use of only a single machine.

5.2. Architecture Diagram:

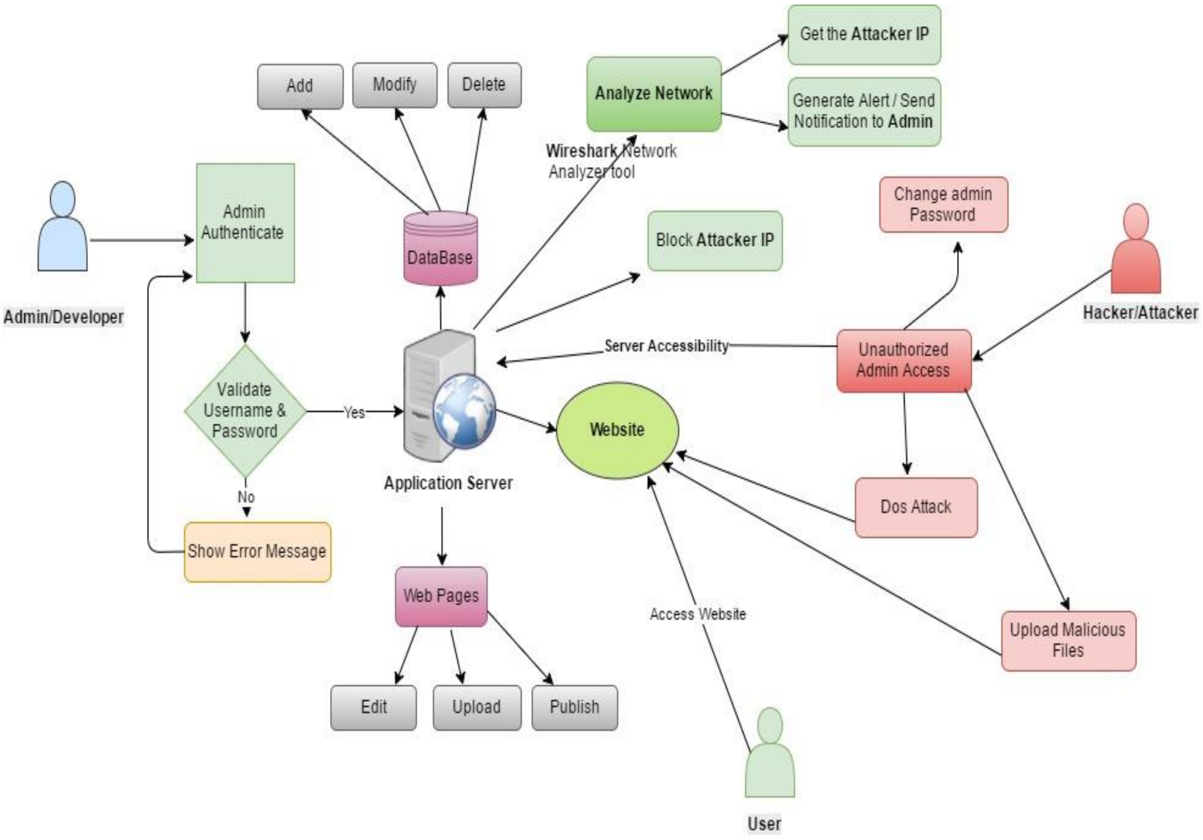


Figure 3. Architecture Diagram

UML Diagrams:

UML (Unified Modelling Language) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

Types of UML Diagrams:

- Use Case Diagram
- Class Diagram
- Activity Diagram
- Sequence Diagram

5.3. Use Case Diagram:

Use case diagrams are usually referred to as behaviour diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

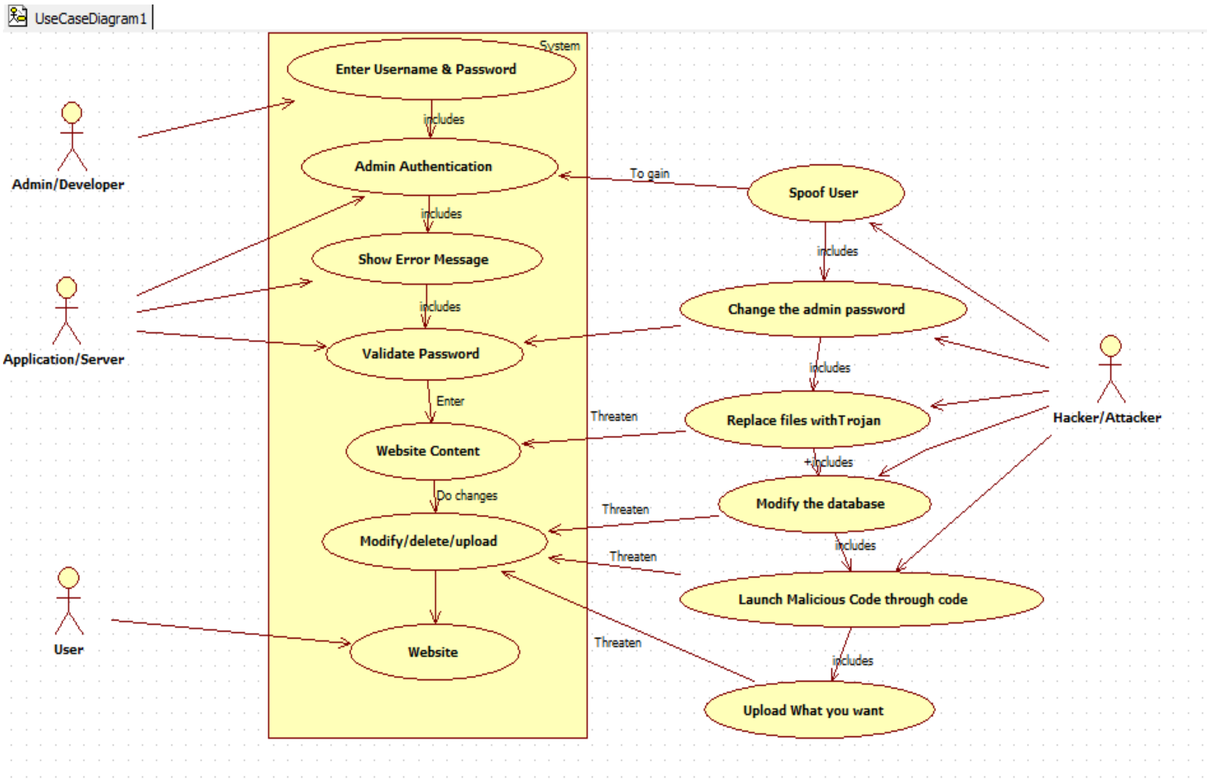


Figure 4. Use Case Diagram

5.4. Class Diagram:

Class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction.

So, the purpose of the class diagram can be summarized as:

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

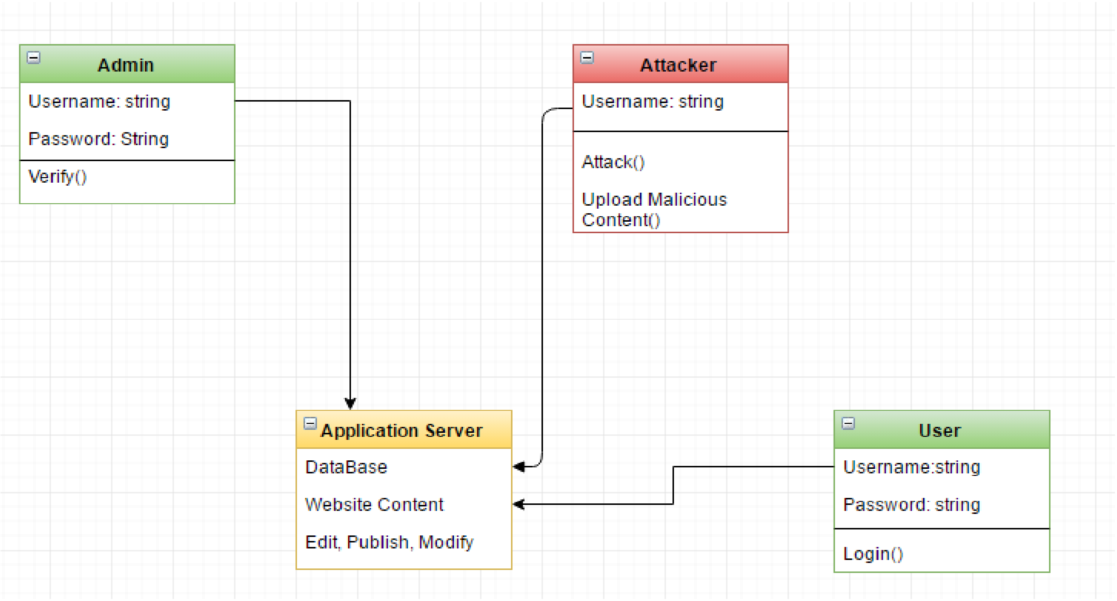


Figure 5. Class Diagram

5.5. Activity Diagram:

Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity Diagram has following elements:

- Activities
- Association
- Conditions
- Constraints

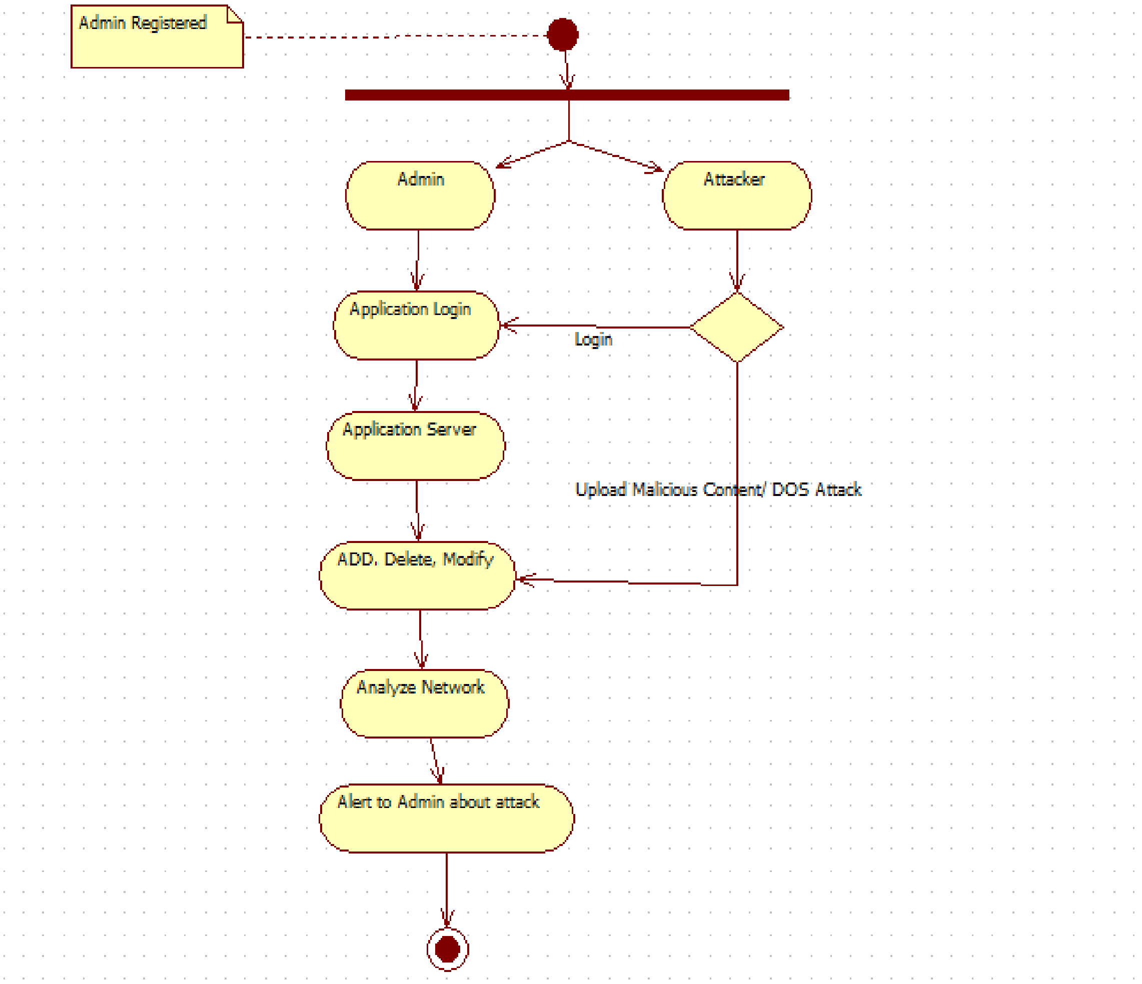


Figure 6. Activity Diagram

5.6. Sequence Diagram:

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

Sequence diagrams can be useful reference diagrams for businesses and other organizations.

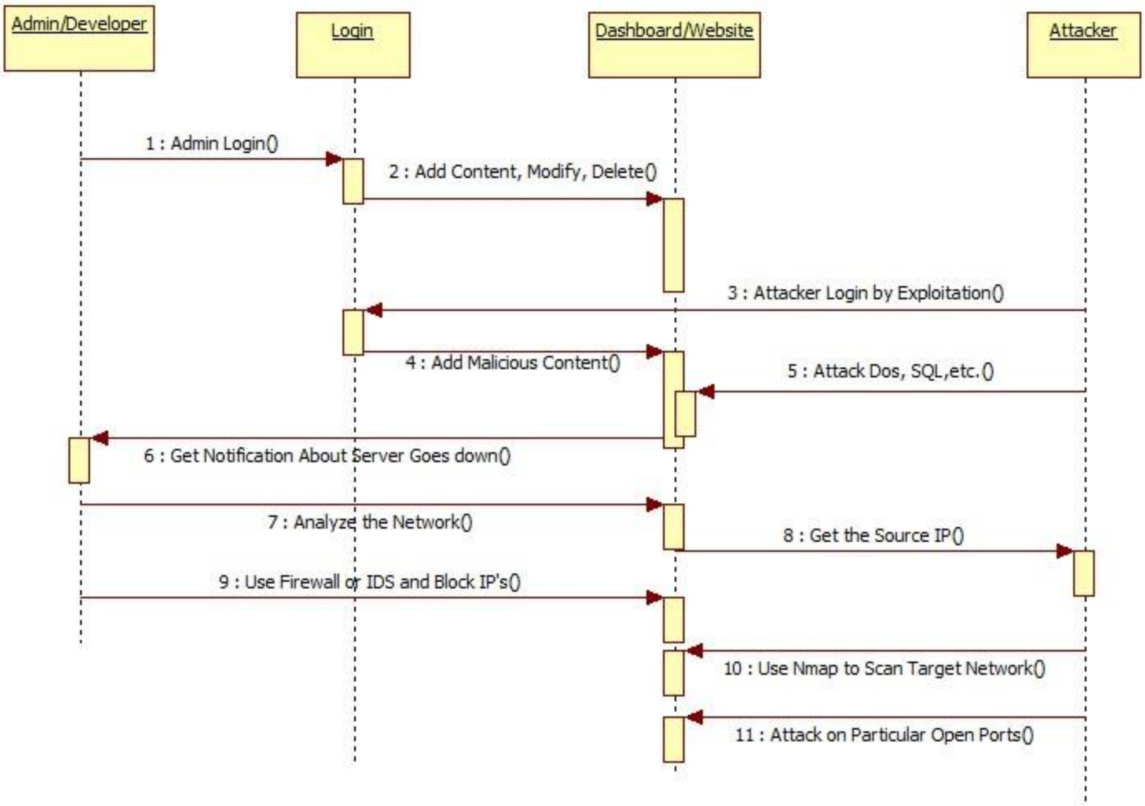

Admin/Developer
Login
Dashboard/Website
Attacker
1 : Admin Login()
2 : Add Content, Modify, Delete()
3 : Attacker Login by Exploitation()
4 : Add Malicious Content()
5 : Attack Dos, SQL,etc.()
6 : Get Notification About Server Goes down()
7 : Analyze the Network()
8 : Get the Source IP()
9 : Use Firewall or IDS and Block IP's()
10 : Use Nmap to Scan Target Network()
11 : Attack on Particular Open Ports()

Figure 7. Sequence Diagram

5.7. Data Flow Diagram:

A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

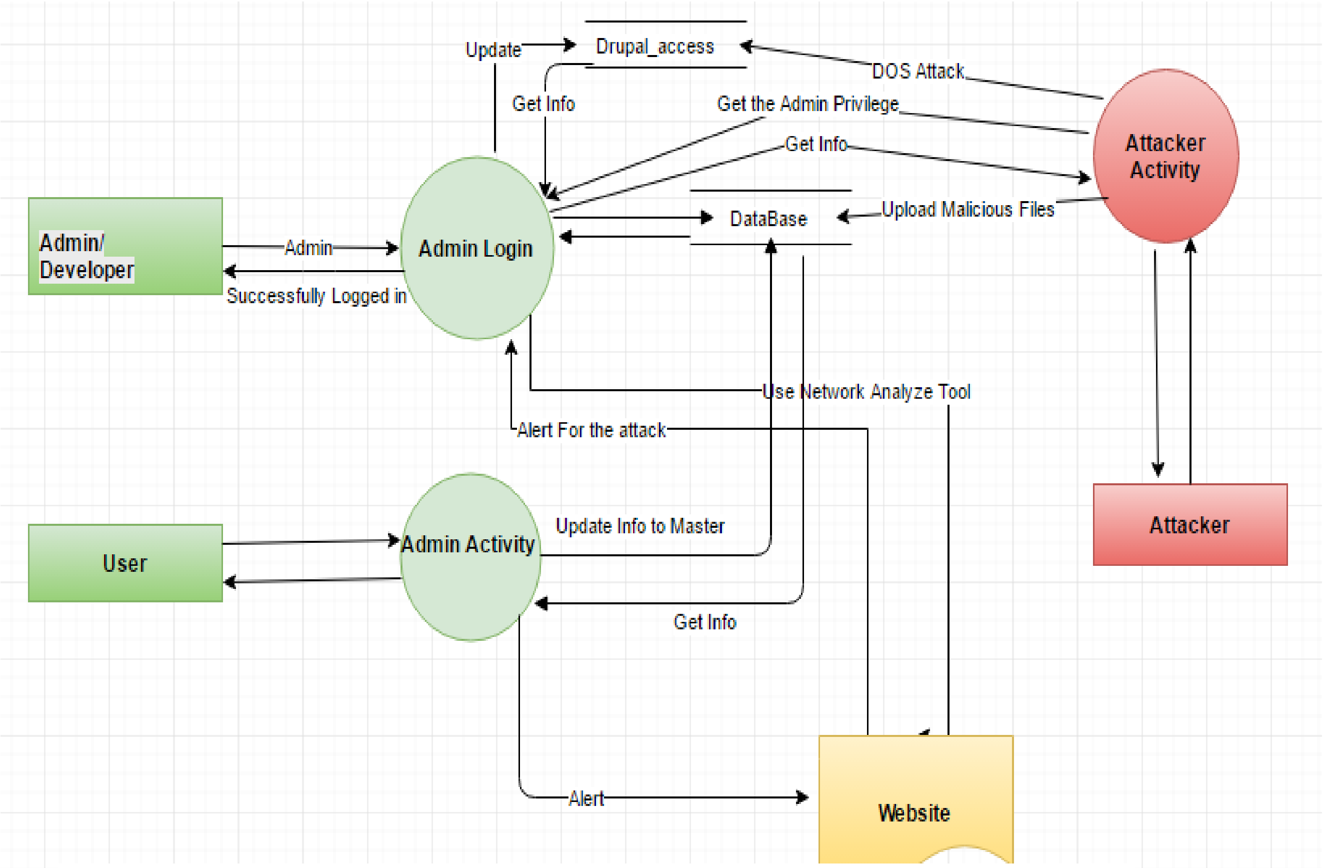


Figure 8. Data Flow Diagram

CHAPTER -6

ANALYSIS MODULES

6.1. System Model:

Web applications represent software solutions which provide accessibility to the user interface through a web browser, and are performed on a remote server by means of interpreter programming languages. Accordingly, and as explained in the previous chapter, CMSs represent the kind of web application that performs the function of content management. There are many organizations which deal with the security issues of computer systems and web applications. Some of them are: SANS Institute, Web Application Security Consortium12, **Computer Emergency Response Team (CERT)**, The Open Web Application Security Project (**OWASP),** Security Focus. In this paper, the recommendations of the OWASP organization are used. The representatives of the OWASP organization have created a document in which the ten most critical vulnerabilities of web applications are listed and explained (OWASP, 2016). The same organization has recommended certain tools for testing critical vulnerability of web applications.

Analysis of High Score CMS Vulnerabilities:

In order to propose adequate security procedures, we must analyse common vulnerabilities and common ways in which they are being exploited. There are many ways in which CMS vulnerabilities could be exploited. Besides manual methods, recently many automated solutions appeared in forms of web kits for exploiting common vulnerabilities. Analysis of vulnerability severity for Drupal.

SQL injection allows attacker injection of destructive SQL code in order to misuse database of CMS application. This is a code injection technique that exploits a security vulnerability occurring in the database layer of an application and service. This is most often found within web pages with dynamic content. In 2015, serious SQL injection vulnerability was discovered in Drupal. SQL injection vulnerability in the ***drupal/includes/post.php*** in Drupal. We can conclude that most frequently reported types of attacks are mainly using SQL injection vulnerabilities or CSRF in combination with SQL injection. Also, we can conclude that having organized and interactive community (like one present around WordPress development) and professional hosting is essential for security of CMS solution.

6.2. System Modules:

1. Vulnerabilities
2. Admin Access
3. Dos Attack
4. SQL Injection attack
5. Analyze Network
6. Network Scanning
7. Generate Alert

6.2.1. Vulnerabilities:

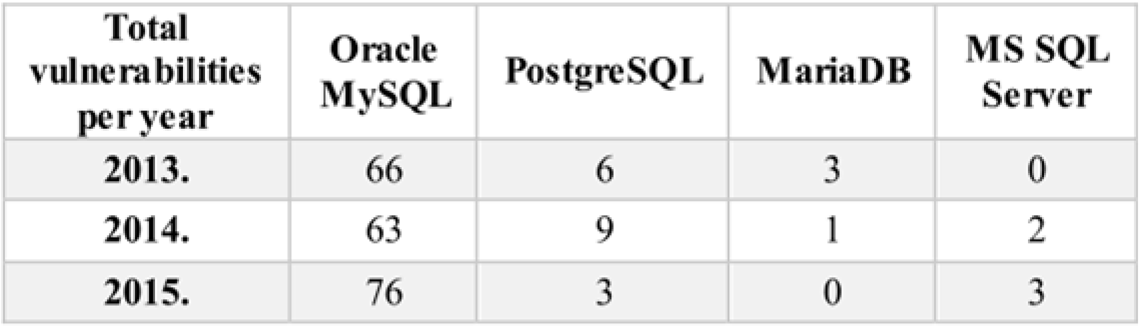
The common types of vulnerabilities in the Content Management System that are responsible for the problems occurring in the CMS platforms:

- Database Vulnerabilities
- Web Server Vulnerabilities
- CMS Vulnerabilities
- Server Vulnerabilities

6.2.1.1. Database Vulnerabilities:

Final goal of attacker is usually access to database so securing database is probably most important issue in overall security of web application. Most popular CMS usually are using MySQL databases but since Oracle acquired it many users are turning to Maria DB which is a community-developed fork of the MySQL.

Table 4. Total Vulnerabilities by Year

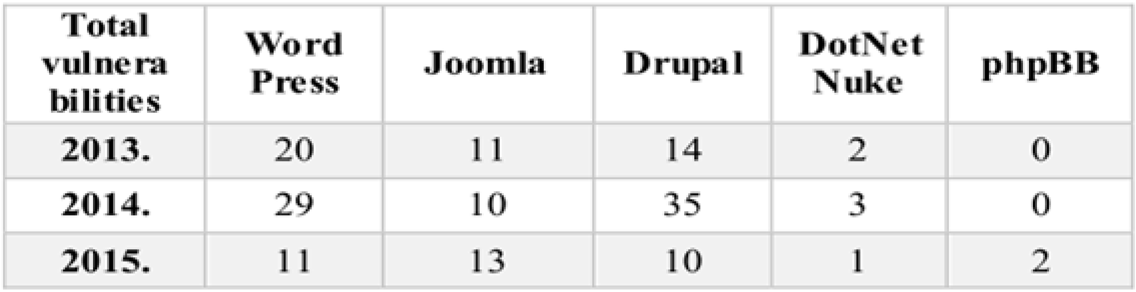


Total vulnerabilities per year	Oracle MySQL	PostgreSQL	MariaDB	MS SQL Server
2013.	66	6	3	0
2014.	63	9	1	2
2015.	76	3	0	3

6.2.1.2. Web Server Vulnerabilities:

Three major open source CMS have to be hosted on web servers that support PHP/MySQL. So, security of application greatly depends on security of web server. We will do comparison of general security of web servers commonly used to host three major CMS and other currently used web servers.

Table 5. Total Vulnerabilities by CMS Comparison

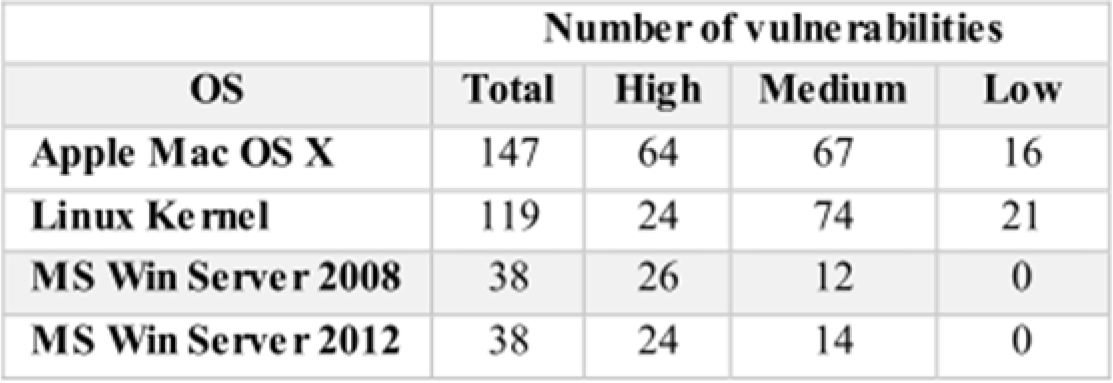


Total vulnerabilities	Word Press	Joomla	Drupal	DotNet Nuke	phpBB
2013.	20	11	14	2	0
2014.	29	10	35	3	0
2015.	11	13	10	1	2

6.2.1.3. Server Vulnerabilities:

Securing application itself is futile without server-side validation measures. So, we'll compare general security of Windows servers (which are most commonly used for running three most used CMS) with security of other servers (mainly Linux servers).

Table 6. Operating System Vulnerabilities



	Number of vulnerabilities			
OS	Total	High	Medium	Low
Apple Mac OS X	147	64	67	16
Linux Kernel	119	24	74	21
MS Win Server 2008	38	26	12	0
MS Win Server 2012	38	24	14	0

6.2.1.4. CMS Vulnerabilities:

Main cause of security problems is CMS application itself. We'll analyse security situation for main most used CMS: WordPress, Joomla and Drupal and compare that to DotNetNuke (.NET based CMS) Most dangerous types of attacks are related to SQL injection vulnerability because these attacks have higher availability impact on the system. Most dangerous types of attacks are related to SQL injection vulnerability because these attacks have higher availability impact on the system. Usually there is reduced performance or interruptions in resource availability.

General hacking probability of CMS is defined in *Fig*. Hackers dominantly focus on CMS with greater market share, than on existing CMS exploits and server vulnerabilities.

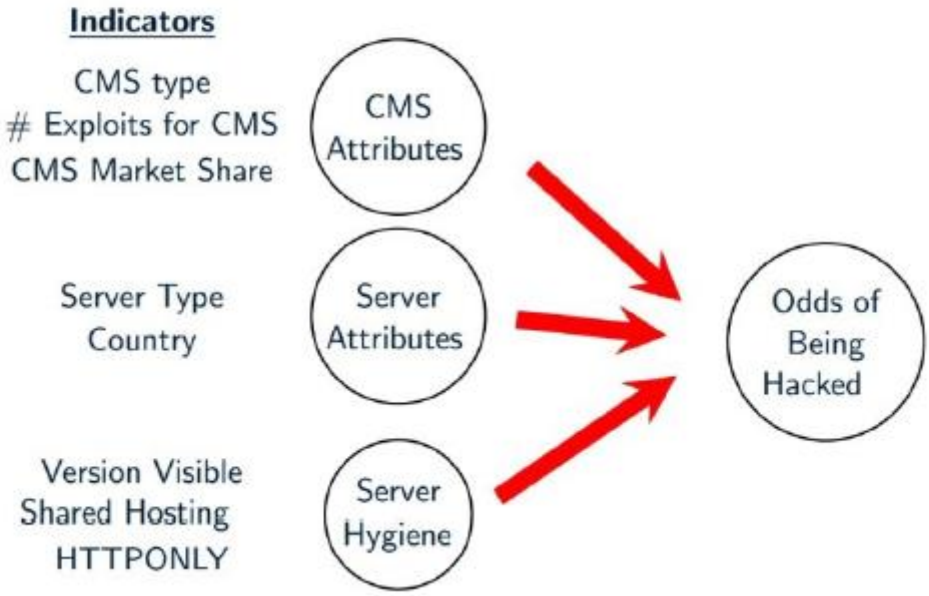


Figure 9. CMS Hacking Probabilities

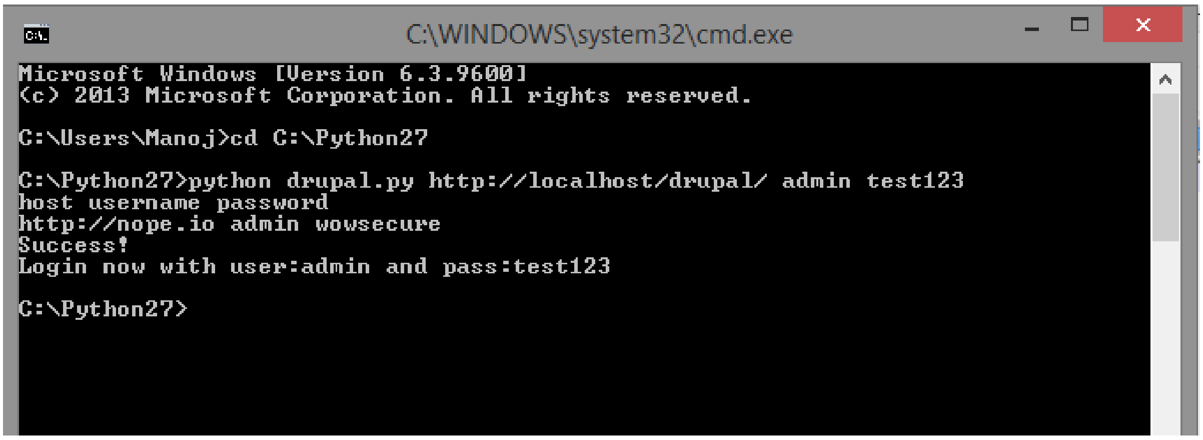
6.2.2. Admin Access:

With only standard user privileges, the attacker gets full access to the Administrator of the user account. This is enough to integrate malicious code into processes in order to remotely control the system, to modify the content in the browser, etc. Since most antivirus programs can control attempts to implement unknown code in the processes, attackers often use more secretive methods. Since the accounts of most corporate users are domain accounts, the domain authentication provides the user with access to the whole access of the website content. This access is often provided automatically without any additional verification of the username and password. As a result, if the infected user has access to the corporate database, attackers can easily take advantage of it.

In our case by using the python code we can change the administration password to get access of the whole content of the website. There are codes available in python called **Python Exploit** save them in the python installation directory and then run the command prompt and go to the python command directory: - "**C:\Python27**" and type the command:

C:\Python27>python drupal.py http://localhost/drupal/ admin test123.

And here(below) is the result: -



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Manoj>cd C:\Python27

C:\Python27>python drupal.py http://localhost/drupal/ admin test123
host username password
http://nope.io admin wowsecure
Success!
Login now with user:admin and pass:test123

C:\Python27>
```

Figure 10. Admin Access

6.2.3. Dos Attack:

What is a Denial of Service Attack?

A DOS attack is an attempt to make a system or server unavailable for legitimate users and, finally, to take the service down. This is achieved by flooding the server's request queue with fake requests. After this, server will not be able to handle the requests of legitimate users. The denial of service (DOS) attack is one of the most powerful attacks used by hackers to harm a company or organization. In past few years, the use of the attack has increased due to the availability of free tools.

6.2.4. SQL Injection:

SQL Injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements (also commonly referred to as a malicious payload) that control a web application's database server (also commonly referred to as a Relational Database Management System – RDBMS). Since an SQL Injection vulnerability could possibly affect any website or web application that makes use of an SQL-based database, the vulnerability is one of the oldest, most prevalent and most dangerous of web application vulnerabilities.

6.2.5. Analyse Network:

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. Wireshark includes filters, color-coding and other features that let you dig deep into network traffic and inspect individual packets. Wireshark has two filtering languages: One used when capturing packets, and one used when displaying packets.

6.2.6. Network Scanning:

Nmap:

Nmap (Network Mapper) is a security scanner to scan the network to find vulnerabilities in the target website. Open source tool Nmap is a popular choice amongst hackers and security pros alike for network mapping, port-scanning and testing for network vulnerabilities. As an alternative to the router-based device tracking, it is possible to directly scan the network for devices by using Nmap.

CHAPTER-7

IMPLEMENTATION

7.1. DDos Attack:

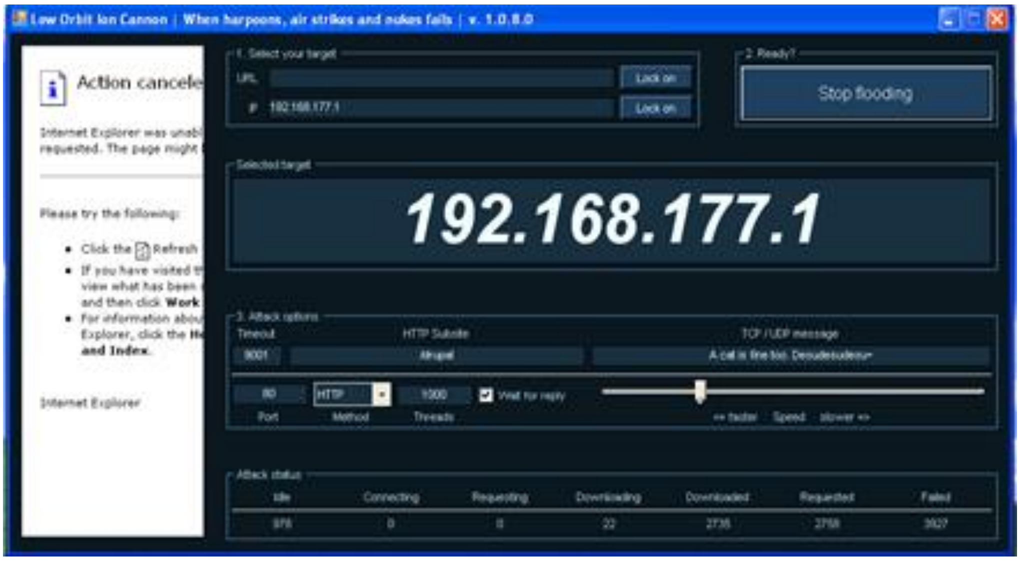
DDOS or Distributed Denial of Service Attack

This is the complicated but powerful version of DOS attack in which many attacking systems are involved. In DDOS attacks, many computers start performing DOS attacks on the same target server. As the DOS attack is distributed over large group of computers, it is known as a distributed denial of service attack.

To perform a DDOS attack, attackers use a zombie network, which is a group of infected computers on which the attacker has silently installed the DOS attacking tool. Whenever he wants to perform DDOS, he can use all the computers of ZOMBIE network to perform the attack. In simple words, when a server system is being flooded from **fake requests** coming from multiple sources, it is known as a DDOS attack. There are many tools available for free that can be used to flood a server and perform an attack.

Free DOS Attacking Tools:

- LOIC (Low Orbit Ion Canon)
- HOIC (High Orbit Ion Canon)
- PyLoris



IP 192.168.177.1

192.168.177.1

Timeout 9001 HTTP Subsite /drupal/ TCP / UDP message A cat is fine too. Desudesudesu~

Port 80 Method HTTP Threads 1000 ☑ Wait for reply

<< faster Speed slower >>

Idle	Connecting	Requesting	Downloading	Downloaded	Requested	Failed
978	0	0	22	2738	2758	3927

Internet Explorer was unabl... requested. The page might ...

Please try the following:

- Click the Refresh ...
- If you have visited th... view what has been ... and then click **Work** ...
- For information abou... Explorer, click the **He**... **and Index**.

Figure 11. LOIC Attack

Algorithm to Configure LOIC:

- Open attacker machine
- Open the DDos attack tool (**LOIC**)
- Enter the target IP
- Click on Lock on
- Then set the method as http
- And click on Flood

7.2.7. ATTACK CONFIGURATION:

- **UDP Attack:** To perform the UDP attack, select the method of attack as UDP. It has port 80 as the default option selected, but you can change this according to your need. Change the message string or leave it as the default.
- **TCP Attack:** This method is similar to UDP attack. Select the type of attack as TCP to use this.
- **HTTP Attack:** In this attack, the tool sends HTTP requests to the target server. A web application firewall can detect this type of attack easily.

How to use LOIC to perform a Dos attack: Just follow these simple steps to perform a DOS attack against a website.

Step 1: Run the tool.

Step 2: Enter the URL of the website in The URL field and click on Lock O. Then, select attack method (TCP, UDP or HTTP).

Step 3: Change other parameters per your choice or leave it to the default. Now click on the Big Button labelled as "**IMMA CHARGIN MAH LAZER**." You have just mounted an attack on the target.

After starting the attack, you will see some numbers in the Attack status fields. When the requested number stops increasing, restart the LOIC or change the IP. You can also give the UDP attack a try. We can also set the speed of the attack by the slider. It is set to faster as default but we can slow down it with the slider.

Here's the meaning of each field:

- **IDLE:** It shows the number of threads idle. It should be zero for higher efficiency of the attack.
- **Connecting:** This shows the number of threads that are trying to connect to the victim server.
- **Requesting:** This shows the number of threads that are requesting some information from the victim server.
- **Downloading:** This shows the number of threads that are initiating some download for some information from the server.
- **Downloaded:** This number shows how many times data downloading has been initiated from victim server on which you are attacking.
- **Requested:** This number shows how many times a data download has been requested from victim server.
- **Failed:** This number shows how many times the server did not respond to the request. A larger number in this field means the server is going down. The success of the attack can be measured by the number shown in this field.

Drawbacks of using LOIC:

The main drawback of LOIC as a DOS attack tool is that it is very easy to find the attacker. This tool does not take any precautions to hide IP address of the origin of the attack. Attacks generated by this tool are simple and expose the IP address of attacker in each request packet sent to victim server to flood the request queue. Attackers cannot use proxies in these attacks because our requests will hit the proxy server, not the target server. So, we will not be able to launch a DOS attack on the server effectively while using a proxy. But some analysts say that this can be used with a proxy server if the proxy is robust enough. According to them, all your request packets will be forwarded to the server system by proxy at the end.

How to prevent the attack of LOIC:

LOIC is available for free to download and use, and can be used effectively with very little hacking experience. Anyone that wants to can attack a website with this tool. As discussed above, the attack of this tool is simple and easy to identify. A well-configured firewall is enough to prevent the attack from being fully effective. And a server administrator can see the request logs to identify the IP and block the IP from the server. Every website owner or server administrators should monitor the traffic and all the activities being performed on the server. This can help well enough against the attack. But this will not help you when a network of LOIC clients will fire on the server system all at once. Protecting the server with a Firewall configured to filter the packets sent by the LOIC is the best way to protect against the attack.

7.3. NMAP:

Nmap (Network Mapper) is a security scanner to scan the network to find vulnerabilities in the target website. Open source tool Nmap is a popular choice amongst hackers and security pros alike for network mapping, port-scanning and testing for network vulnerabilities. As an alternative to the router-based device tracking, it is possible to directly scan the network for devices by using Nmap.

1. Nmap Target Selection

Scan a single IP	***nmap 192.168.1.1***
Scan a host	***nmap www.hostname.com***
Scan a range of IPs	***nmap 192.168.1.1-20***
Scan targets from a text file	***nmap -iL list-of-ips.txt***

2. Nmap Port Selection

Scan a single Port	***nmap -p 22 192.168.1.1***
Scan a range of ports	***nmap -p 1-100 192.168.1.1***
Scan 100 most common ports (Fast)	***nmap -F 192.168.1.1***
Scan all 65535 ports	***nmap -p- 192.168.1.1***

3. Nmap Port Scan types

Scan using TCP connect ***nmap -sT 192.168.1.1***

Scan using TCP SYN scan (default) ***nmap -sS 192.168.1.1***

Scan UDP ports ***nmap -sU -p 123,161,162 192.168.1.1***

Scan selected ports - ignore discovery ***nmap -Pn -F 192.168.1.1***

4. Service and OS Detection

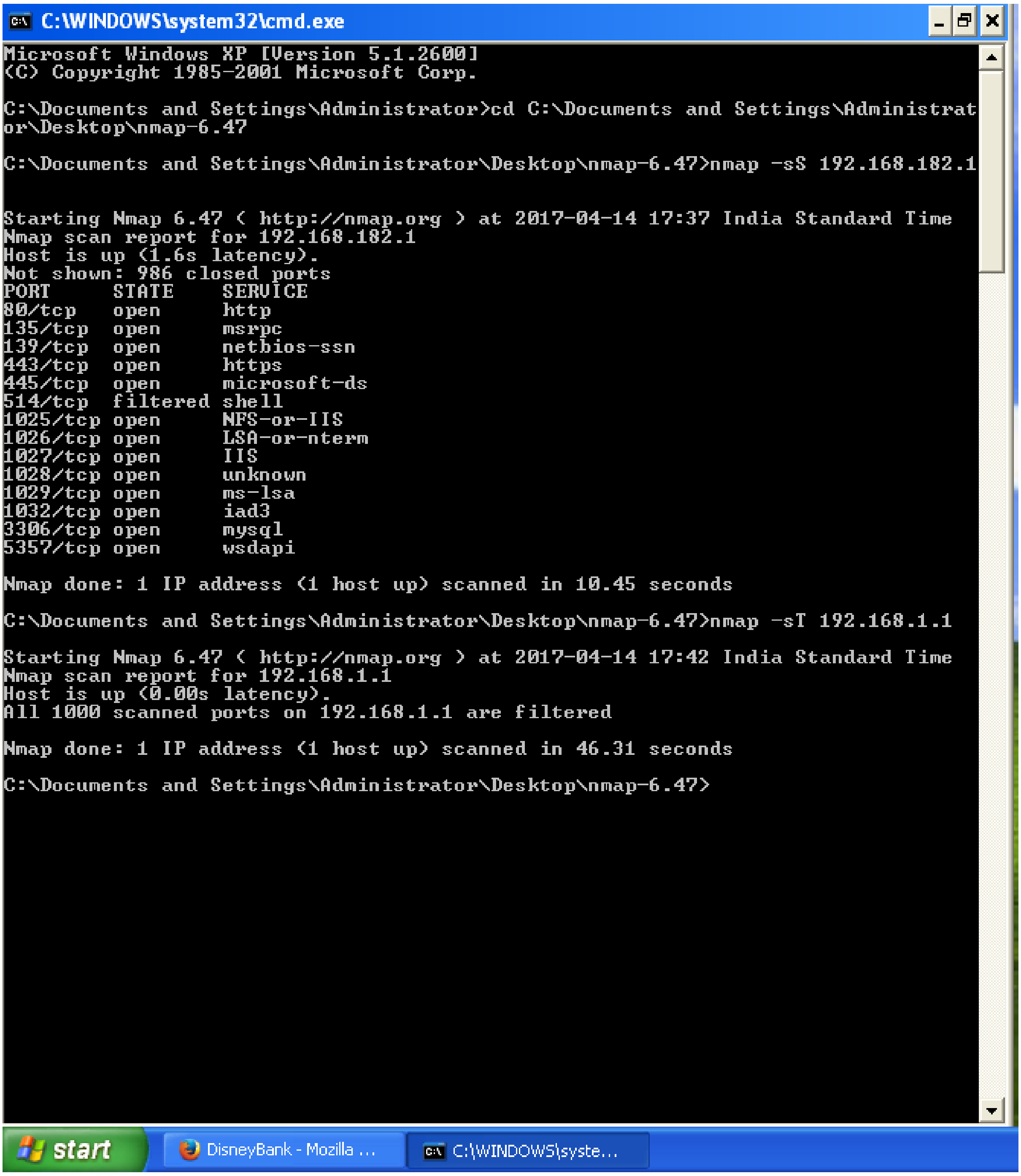
Detect OS and Services ***nmap -A 192.168.1.1***

Standard service detection ***nmap -sV 192.168.1.1***

More aggressive Service Detection ***nmap -sV --version-intensity 5*** 192.168.1.1

Lighter banner grabbing detection ***nmap -sV --version-intensity 0 192.168.1.1***

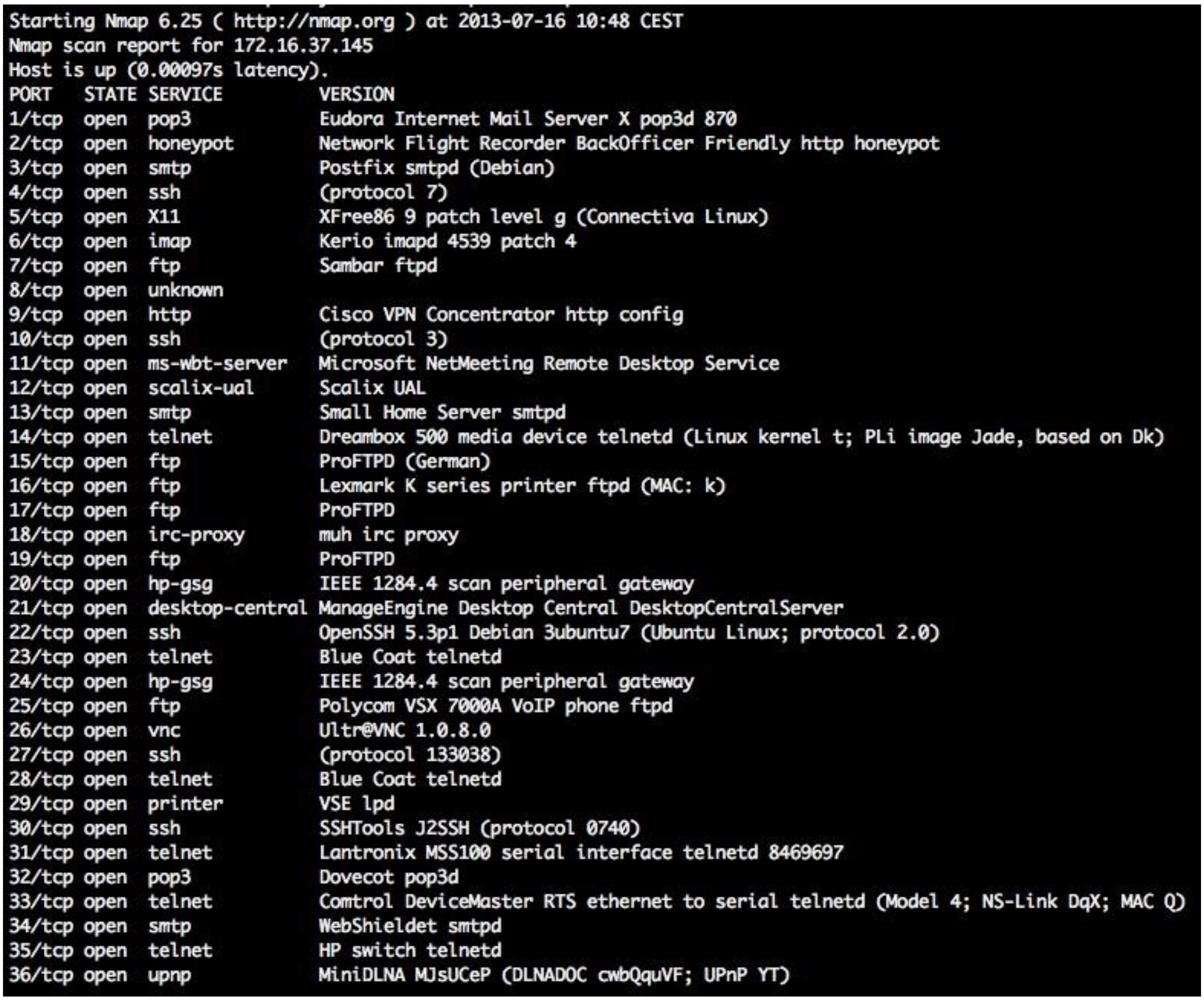
TCP Scan and Stealth Scan:



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>cd C:\Documents and Settings\Administrat
or\Desktop\nmap-6.47

C:\Documents and Settings\Administrator\Desktop\nmap-6.47>nmap -sS 192.168.182.1


Starting Nmap 6.47 ( http://nmap.org ) at 2017-04-14 17:37 India Standard Time
Nmap scan report for 192.168.182.1
Host is up (1.6s latency).
Not shown: 986 closed ports
PORT      STATE    SERVICE
80/tcp    open     http
135/tcp   open     msrpc
139/tcp   open     netbios-ssn
443/tcp   open     https
445/tcp   open     microsoft-ds
514/tcp   filtered shell
1025/tcp  open     NFS-or-IIS
1026/tcp  open     LSA-or-nterm
1027/tcp  open     IIS
1028/tcp  open     unknown
1029/tcp  open     ms-lsa
1032/tcp  open     iad3
3306/tcp  open     mysql
5357/tcp  open     wsdapi

Nmap done: 1 IP address (1 host up) scanned in 10.45 seconds

C:\Documents and Settings\Administrator\Desktop\nmap-6.47>nmap -sT 192.168.1.1

Starting Nmap 6.47 ( http://nmap.org ) at 2017-04-14 17:42 India Standard Time
Nmap scan report for 192.168.1.1
Host is up (0.00s latency).
All 1000 scanned ports on 192.168.1.1 are filtered

Nmap done: 1 IP address (1 host up) scanned in 46.31 seconds

C:\Documents and Settings\Administrator\Desktop\nmap-6.47>
```

Figure 12. Nmap TCP Port Scanning

Scanning All ports:



```
Starting Nmap 6.25 ( http://nmap.org ) at 2013-07-16 10:48 CEST
Nmap scan report for 172.16.37.145
Host is up (0.00097s latency).
PORT   STATE SERVICE         VERSION
1/tcp  open  pop3            Eudora Internet Mail Server X pop3d 870
2/tcp  open  honeypot        Network Flight Recorder BackOfficer Friendly http honeypot
3/tcp  open  smtp            Postfix smtpd (Debian)
4/tcp  open  ssh             (protocol 7)
5/tcp  open  X11             XFree86 9 patch level g (Connectiva Linux)
6/tcp  open  imap            Kerio imapd 4539 patch 4
7/tcp  open  ftp             Sambar ftpd
8/tcp  open  unknown
9/tcp  open  http            Cisco VPN Concentrator http config
10/tcp open  ssh             (protocol 3)
11/tcp open  ms-wbt-server   Microsoft NetMeeting Remote Desktop Service
12/tcp open  scalix-ual      Scalix UAL
13/tcp open  smtp            Small Home Server smtpd
14/tcp open  telnet          Dreambox 500 media device telnetd (Linux kernel t; PLi image Jade, based on Dk)
15/tcp open  ftp             ProFTPD (German)
16/tcp open  ftp             Lexmark K series printer ftpd (MAC: k)
17/tcp open  ftp             ProFTPD
18/tcp open  irc-proxy       muh irc proxy
19/tcp open  ftp             ProFTPD
20/tcp open  hp-gsg          IEEE 1284.4 scan peripheral gateway
21/tcp open  desktop-central ManageEngine Desktop Central DesktopCentralServer
22/tcp open  ssh             OpenSSH 5.3p1 Debian 3ubuntu7 (Ubuntu Linux; protocol 2.0)
23/tcp open  telnet          Blue Coat telnetd
24/tcp open  hp-gsg          IEEE 1284.4 scan peripheral gateway
25/tcp open  ftp             Polycom VSX 7000A VoIP phone ftpd
26/tcp open  vnc             Ultr@VNC 1.0.8.0
27/tcp open  ssh             (protocol 133038)
28/tcp open  telnet          Blue Coat telnetd
29/tcp open  printer         VSE lpd
30/tcp open  ssh             SSHTools J2SSH (protocol 0740)
31/tcp open  telnet          Lantronix MSS100 serial interface telnetd 8469697
32/tcp open  pop3            Dovecot pop3d
33/tcp open  telnet          Comtrol DeviceMaster RTS ethernet to serial telnetd (Model 4; NS-Link DqX; MAC Q)
34/tcp open  smtp            WebShieldet smtpd
35/tcp open  telnet          HP switch telnetd
36/tcp open  upnp            MiniDLNA MJsUCeP (DLNADOC cwbQquVF; UPnP YT)
```

Figure 13. Nmap all ports Scanning

The 3 way TCP handshake

First a bit of background, during communication with a TCP service, a single connection is established with the TCP 3 way handshake. This involves a SYN sent to an TCP open port that has a service bound to it, typical examples are HTTP (port 80), SMTP (port 25), POP3 (port 110) or SSH (port 22). The server side will see the SYN and respond with SYN ACK, with the client answering the SYN ACK with an ACK. This completes the set up and the data of the service protocol can now be communicated.

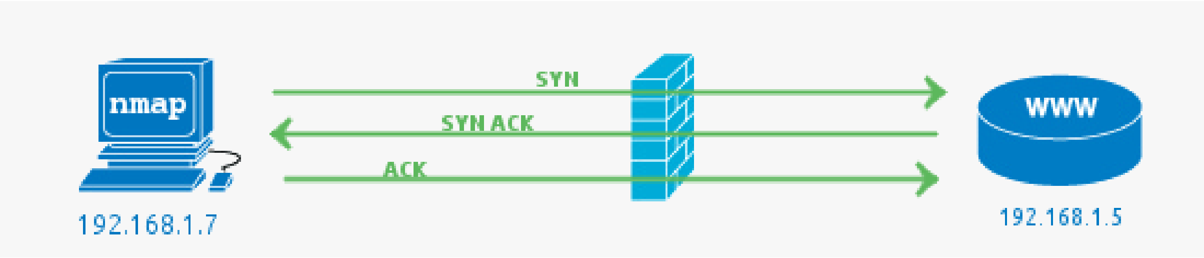


Figure 14. Nmap 3-way Handshake

Real time and display them in human-readable format. Wireshark includes filters, color-coding and other features that let you dig deep into network traffic and inspect individual packets. Wireshark has two filtering languages: One used when capturing packets, and one used when displaying packets.

7.5. FILTERING PACKETS:

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format.

Wireshark Features:

- Live capture, which allows sniffing data from a live network in real time.
- Support for offline protocol analysis.
- Enriched user interface (UI).
- Supports almost all network, transport and application protocols.
- Active development by enthusiasts across the globe.
- Compatibility with other products. Supports standard. pcap format which makes it compatible for use with other such tools.

Step 1: Start Wireshark!

To open Wireshark in Windows, open it after installation of the Wireshark setup.

Step 2: Wireshark GUI

Once the Wireshark GUI has opened, we will see that the dashboard has a left column box called ‘Interface List’. This list lets you know which devices and capture cards you can use. At the top of the application there is an option called ‘Capture Options’ which is exactly that, it allows you to modify and tweak how you would like to capture the packets of data that are being transmitted over your network.

Step 3. Wireshark Interface

If you have a look at your interface list (see Step 2 and the associated screen shot) you’ll see that one of your devices is actually sending and receiving packets. Options include promiscuous mode and capture mode etc. Have a play around with these and understand what each of these functions does – and you will rapidly learn how to effectively use Wireshark.

Step 4. Capture Interface Options

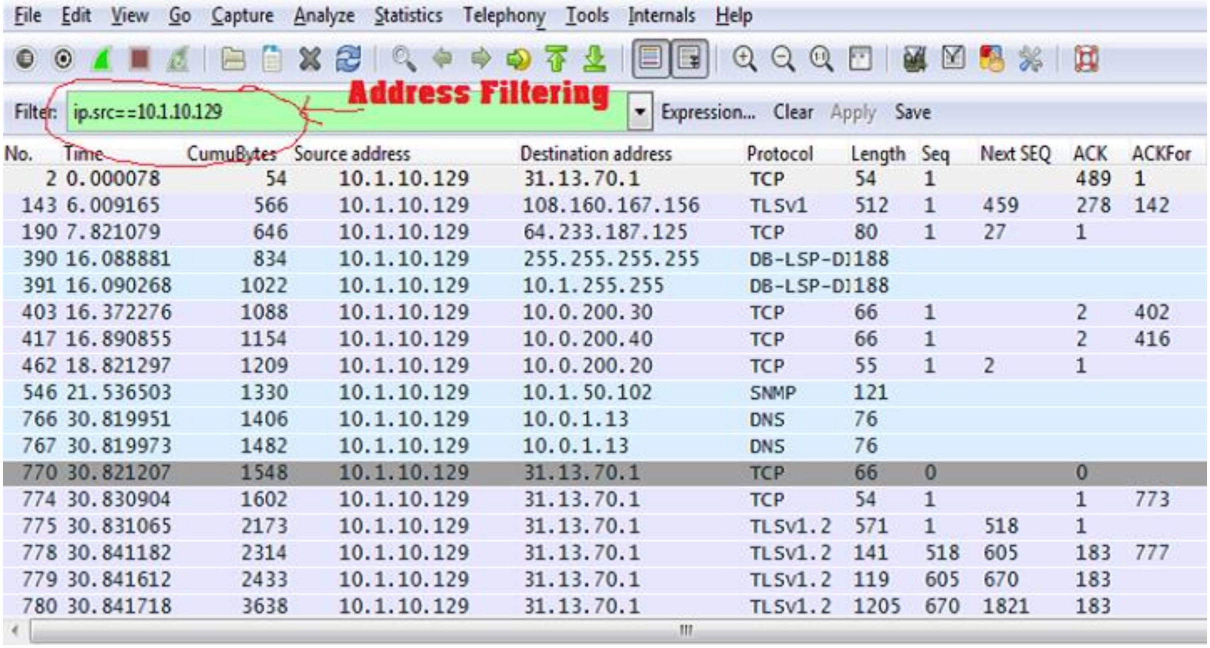
This screen shot shows the Wireshark capture interfaces, in other words, it shows what processes and platforms are receiving and sending data on your machine. If you have a wireless card, then it will show it.

Step 5. The Main Packets Panel

Once you are happy with the interface you’d like to use, go ahead and click ‘start’ and Wireshark will show all the packets that are being transmitted over your network. The whole point here is to find patterns or anything that looks suspicious. Taking the columns at the top of the Wireshark interface from left to right, the first number is the ‘packet number’. The second column shows how many seconds it has been since the start of the capture. The third column is the source IP Address and the fourth column shows the destination IP Address. The fifth column is the protocol that sent the packet, i.e. it could be DNS, TCP (Transmission Control Protocol) or even HTTP.

Filtering the packets is key when using Wireshark – done by using the search bar within the interface (top left). If we right click on a packet of interest you can ‘follow TCP stream’ and you get a ton of raw information.

Filter with the source IP address

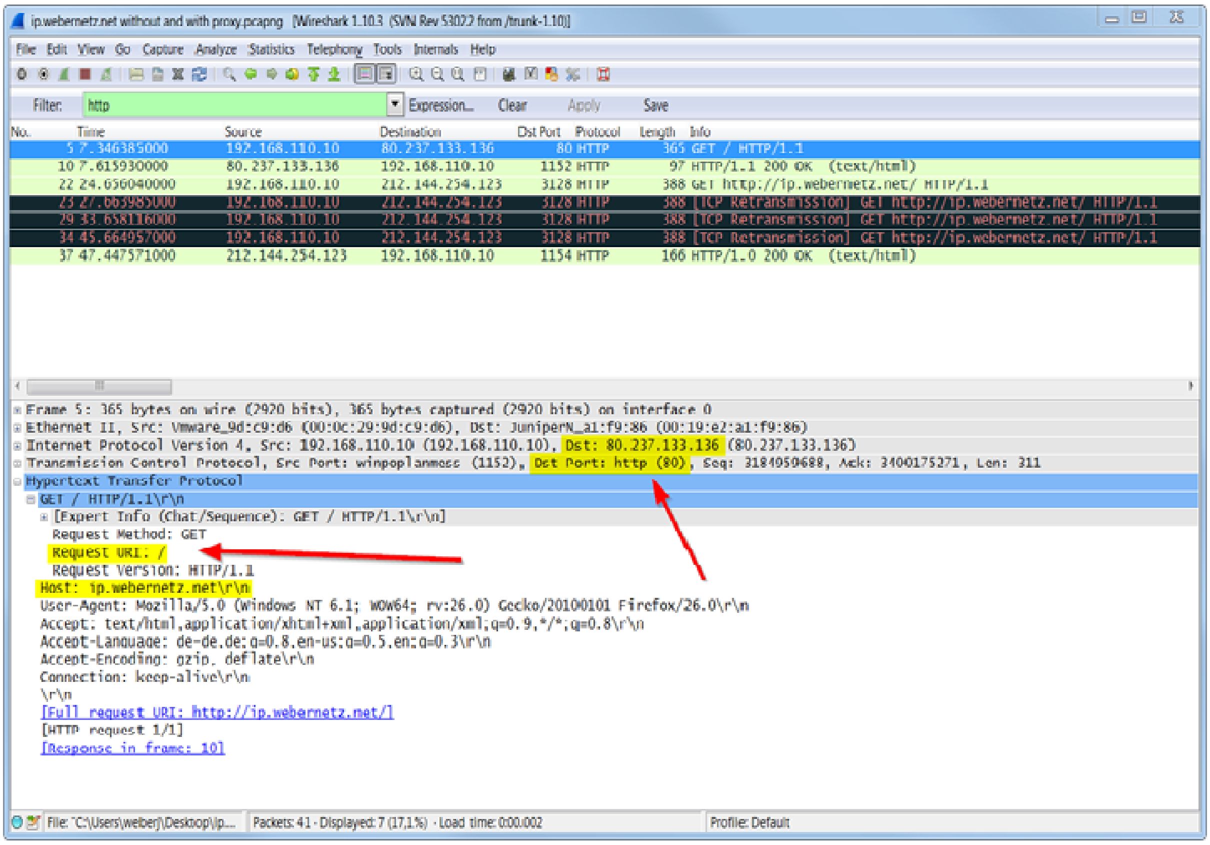


Filter: ip.src==10.1.10.129

No.	Time	CumuBytes	Source address	Destination address	Protocol	Length	Seq	Next SEQ	ACK	ACKFor
2	0.000078	54	10.1.10.129	31.13.70.1	TCP	54	1		489	1
143	6.009165	566	10.1.10.129	108.160.167.156	TLSv1	512	1	459	278	142
190	7.821079	646	10.1.10.129	64.233.187.125	TCP	80	1	27	1	
390	16.088881	834	10.1.10.129	255.255.255.255	DB-LSP-DI	188				
391	16.090268	1022	10.1.10.129	10.1.255.255	DB-LSP-DI	188				
403	16.372276	1088	10.1.10.129	10.0.200.30	TCP	66	1		2	402
417	16.890855	1154	10.1.10.129	10.0.200.40	TCP	66	1		2	416
462	18.821297	1209	10.1.10.129	10.0.200.20	TCP	55	1	2	1	
546	21.536503	1330	10.1.10.129	10.1.50.102	SNMP	121				
766	30.819951	1406	10.1.10.129	10.0.1.13	DNS	76				
767	30.819973	1482	10.1.10.129	10.0.1.13	DNS	76				
770	30.821207	1548	10.1.10.129	31.13.70.1	TCP	66	0		0	
774	30.830904	1602	10.1.10.129	31.13.70.1	TCP	54	1		1	773
775	30.831065	2173	10.1.10.129	31.13.70.1	TLSv1.2	571	1	518	1	
778	30.841182	2314	10.1.10.129	31.13.70.1	TLSv1.2	141	518	605	183	777
779	30.841612	2433	10.1.10.129	31.13.70.1	TLSv1.2	119	605	670	183	
780	30.841718	3638	10.1.10.129	31.13.70.1	TLSv1.2	1205	670	1821	183	

Figure 15. Wireshark IP search

Get the http requests:


ip.webernetz.net without and with proxy.pcapng [Wireshark 1.10.3 (SVN Rev 53022 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: http
Expression... Clear Apply Save

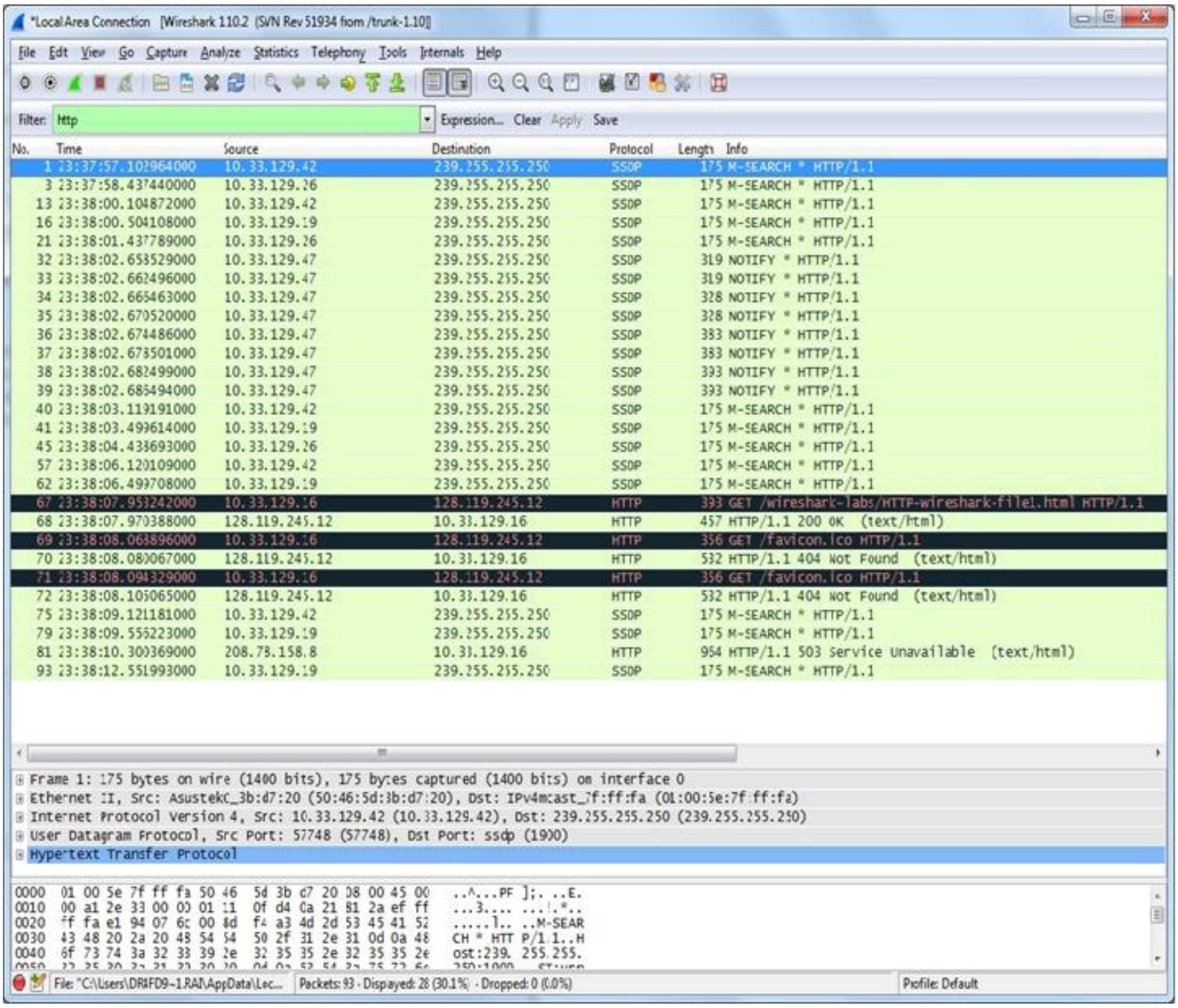
```
No.  Time          Source           Destination      Dst Port Protocol Length Info
   5 7.346385000   192.168.110.10   80.237.133.136     80 HTTP   365 GET / HTTP/1.1
  10 7.615930000   80.237.133.136   192.168.110.10   1152 HTTP    97 HTTP/1.1 200 OK  (text/html)
  22 24.656040000  192.168.110.10   212.144.254.123  3128 HTTP   388 GET http://ip.webernetz.net/ HTTP/1.1
  23 27.663985000  192.168.110.10   212.144.254.123  3128 HTTP   388 [TCP Retransmission] GET http://ip.webernetz.net/ HTTP/1.1
  29 33.658116000  192.168.110.10   212.144.254.123  3128 HTTP   388 [TCP Retransmission] GET http://ip.webernetz.net/ HTTP/1.1
  34 45.664957000  192.168.110.10   212.144.254.123  3128 HTTP   388 [TCP Retransmission] GET http://ip.webernetz.net/ HTTP/1.1
  37 47.447571000  212.144.254.123  192.168.110.10   1154 HTTP   166 HTTP/1.0 200 OK  (text/html)
```

```
⊞ Frame 5: 365 bytes on wire (2920 bits), 365 bytes captured (2920 bits) on interface 0
⊞ Ethernet II, Src: Vmware_9d:c9:d6 (00:0c:29:9d:c9:d6), Dst: JuniperN_a1:f9:86 (00:19:e2:a1:f9:86)
⊞ Internet Protocol Version 4, Src: 192.168.110.10 (192.168.110.10), Dst: 80.237.133.136 (80.237.133.136)
⊞ Transmission Control Protocol, Src Port: winpoplanmess (1152), Dst Port: http (80), Seq: 3184050688, Ack: 3400175271, Len: 311
⊟ Hypertext Transfer Protocol
  ⊟ GET / HTTP/1.1\r\n
    ⊞ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
    Host: ip.webernetz.net\r\n
    User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:26.0) Gecko/20100101 Firefox/26.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: de-de,de;q=0.8,en-us;q=0.5,en;q=0.3\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://ip.webernetz.net/]
    [HTTP request 1/1]
    [Response in frame: 10]
```

File: "C:\Users\weberj\Desktop\ip... Packets: 41 · Displayed: 7 (17,1%) · Load time: 0:00.002 Profile: Default

Figure 16. Wireshark http request

Filtering:


File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

*Local Area Connection [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]

Filter: http

No.	Time	Source	Destination	Protocol	Length	Info
1	23:37:57.102964000	10.33.129.42	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
3	23:37:58.437440000	10.33.129.26	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
13	23:38:00.104872000	10.33.129.42	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
16	23:38:00.504108000	10.33.129.19	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
21	23:38:01.437789000	10.33.129.26	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
32	23:38:02.658529000	10.33.129.47	239.255.255.250	SSDP	319	NOTIFY * HTTP/1.1
33	23:38:02.662496000	10.33.129.47	239.255.255.250	SSDP	319	NOTIFY * HTTP/1.1
34	23:38:02.665463000	10.33.129.47	239.255.255.250	SSDP	328	NOTIFY * HTTP/1.1
35	23:38:02.670520000	10.33.129.47	239.255.255.250	SSDP	328	NOTIFY * HTTP/1.1
36	23:38:02.674486000	10.33.129.47	239.255.255.250	SSDP	383	NOTIFY * HTTP/1.1
37	23:38:02.678501000	10.33.129.47	239.255.255.250	SSDP	383	NOTIFY * HTTP/1.1
38	23:38:02.682499000	10.33.129.47	239.255.255.250	SSDP	393	NOTIFY * HTTP/1.1
39	23:38:02.686494000	10.33.129.47	239.255.255.250	SSDP	393	NOTIFY * HTTP/1.1
40	23:38:03.119191000	10.33.129.42	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
41	23:38:03.499614000	10.33.129.19	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
45	23:38:04.433693000	10.33.129.26	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
57	23:38:06.120109000	10.33.129.42	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
62	23:38:06.499708000	10.33.129.19	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
67	23:38:07.953242000	10.33.129.16	128.119.245.12	HTTP	393	GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
68	23:38:07.970388000	128.119.245.12	10.33.129.16	HTTP	457	HTTP/1.1 200 OK (text/html)
69	23:38:08.068896000	10.33.129.16	128.119.245.12	HTTP	356	GET /favicon.ico HTTP/1.1
70	23:38:08.080067000	128.119.245.12	10.33.129.16	HTTP	532	HTTP/1.1 404 Not Found (text/html)
71	23:38:08.094329000	10.33.129.16	128.119.245.12	HTTP	356	GET /favicon.ico HTTP/1.1
72	23:38:08.105065000	128.119.245.12	10.33.129.16	HTTP	532	HTTP/1.1 404 Not Found (text/html)
75	23:38:09.121181000	10.33.129.42	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
79	23:38:09.555223000	10.33.129.19	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
81	23:38:10.300369000	208.73.158.8	10.33.129.16	HTTP	954	HTTP/1.1 503 Service Unavailable (text/html)
93	23:38:12.551993000	10.33.129.19	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1

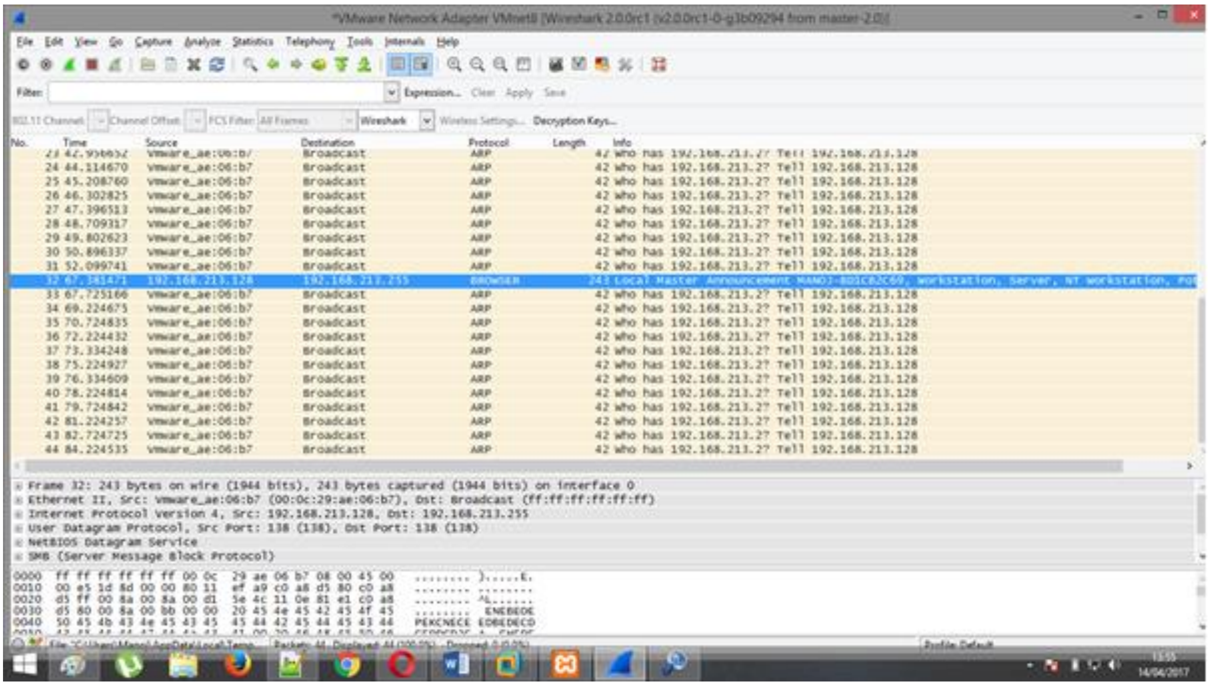
```
⊞ Frame 1: 175 bytes on wire (1400 bits), 175 bytes captured (1400 bits) on interface 0
⊞ Ethernet II, Src: AsustekC_3b:d7:20 (50:46:5d:3b:d7:20), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
⊞ Internet Protocol Version 4, Src: 10.33.129.42 (10.33.129.42), Dst: 239.255.255.250 (239.255.255.250)
⊞ User Datagram Protocol, Src Port: 57748 (57748), Dst Port: ssdp (1900)
⊞ Hypertext Transfer Protocol
```

```
0000  01 00 5e 7f ff fa 50 46  5d 3b d7 20 08 00 45 00   ..^...PF ];. ..E.
0010  00 a1 2e 33 00 00 01 11  0f d4 0a 21 81 2a ef ff   ...3.... ...!.*..
0020  ff fa e1 94 07 6c 00 8d  f4 a3 4d 2d 53 45 41 52   .....l.. ..M-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1..H
0040  6f 73 74 3a 32 33 39 2e  32 35 35 2e 32 35 35 2e   ost:239. 255.255.
```

File: "C:\Users\DR#FD9~1.RAI\AppData\Loc... Packets: 93 · Displayed: 28 (30.1%) · Dropped: 0 (0.0%) Profile: Default

Figure 17. Wireshark Packet Capturing

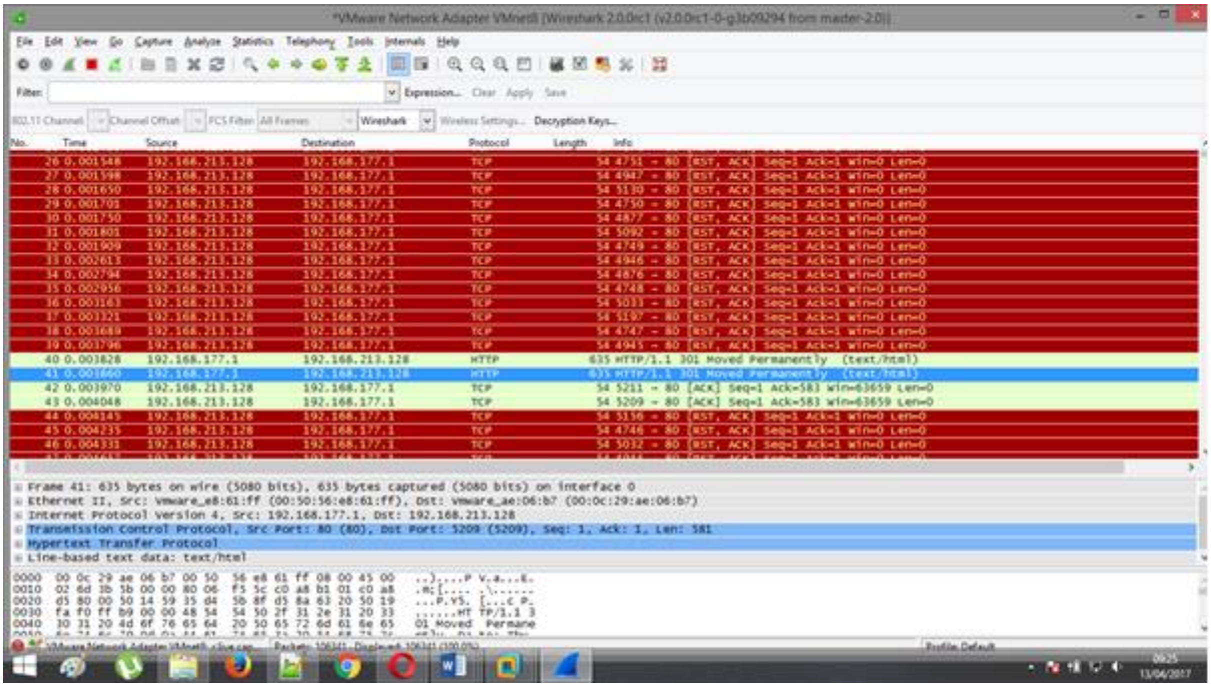
When the attack is happening on the website through **HOIC** that time the network analysis:


*VMware Network Adapter VMnet8 [Wireshark 2.0.0rc1 (v2.0.0rc1-0-g3b09294 from master-2.0)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...
13:55
14/04/2017

```
No.   Time          Source                Destination        Protocol  Length  Info
   24 44.114670     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   25 45.208760     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   26 46.302825     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   27 47.396513     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   28 48.709317     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   29 49.802623     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   30 50.896337     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   31 52.099741     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   32 67.381471     192.168.213.128       192.168.213.255    BROWSER    243 Local Master Announcement MANOJ-8D1C82C69, Workstation, Server, NT Workstation, Po
   33 67.725166     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   34 69.224675     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   35 70.724835     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   36 72.224432     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   37 73.334248     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   38 75.224927     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   39 76.334609     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   40 78.224814     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   41 79.724842     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   42 81.224257     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   43 82.724725     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128
   44 84.224535     Vmware_ae:06:b7       Broadcast          ARP         42 who has 192.168.213.2? Tell 192.168.213.128

⊞ Frame 32: 243 bytes on wire (1944 bits), 243 bytes captured (1944 bits) on interface 0
⊞ Ethernet II, Src: Vmware_ae:06:b7 (00:0c:29:ae:06:b7), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol Version 4, Src: 192.168.213.128, Dst: 192.168.213.255
⊞ User Datagram Protocol, Src Port: 138 (138), Dst Port: 138 (138)
⊞ NetBIOS Datagram Service
⊞ SMB (Server Message Block Protocol)

0000  ff ff ff ff ff ff 00 0c  29 ae 06 b7 08 00 45 00   ........ ).....E.
0010  00 e5 1d 8d 00 00 80 11  ef a9 c0 a8 d5 80 c0 a8   ........ ........
0020  d5 ff 00 8a 00 8a 00 d1  5e 4c 11 0e 81 e1 c0 a8   ........ ^L......
0030  d5 80 00 8a 00 bb 00 00  20 45 4e 45 42 45 4f 45   ........  ENEBEOE
0040  50 45 4b 43 4e 45 43 45  45 44 42 45 44 45 43 44   PEKCNECE EDBEDECD
```

Figure 18. HOIC Packet Capturing

When the **Dos** TCP attack is happening on website, that time the captured analysis look like:


*VMware Network Adapter VMnet8 [Wireshark 2.0.0rc1 (v2.0.0rc1-0-g3b09294 from master-2.0)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...
09:25
13/04/2017

```
No.   Time          Source                Destination        Protocol  Length  Info
   26 0.001548      192.168.213.128       192.168.177.1      TCP         54 4751 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   27 0.001598      192.168.213.128       192.168.177.1      TCP         54 4947 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   28 0.001650      192.168.213.128       192.168.177.1      TCP         54 5130 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   29 0.001701      192.168.213.128       192.168.177.1      TCP         54 4750 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   30 0.001750      192.168.213.128       192.168.177.1      TCP         54 4877 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   31 0.001801      192.168.213.128       192.168.177.1      TCP         54 5092 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   32 0.001909      192.168.213.128       192.168.177.1      TCP         54 4749 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   33 0.002613      192.168.213.128       192.168.177.1      TCP         54 4946 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   34 0.002794      192.168.213.128       192.168.177.1      TCP         54 4876 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   35 0.002956      192.168.213.128       192.168.177.1      TCP         54 4748 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   36 0.003163      192.168.213.128       192.168.177.1      TCP         54 5033 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   37 0.003321      192.168.213.128       192.168.177.1      TCP         54 5197 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   38 0.003689      192.168.213.128       192.168.177.1      TCP         54 4747 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   39 0.003796      192.168.213.128       192.168.177.1      TCP         54 4945 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   40 0.003828      192.168.177.1         192.168.213.128    HTTP       635 HTTP/1.1 301 Moved Permanently  (text/html)
   41 0.003860      192.168.177.1         192.168.213.128    HTTP       635 HTTP/1.1 301 Moved Permanently  (text/html)
   42 0.003970      192.168.213.128       192.168.177.1      TCP         54 5211 → 80 [ACK] Seq=1 Ack=583 Win=63659 Len=0
   43 0.004048      192.168.213.128       192.168.177.1      TCP         54 5209 → 80 [ACK] Seq=1 Ack=583 Win=63659 Len=0
   44 0.004145      192.168.213.128       192.168.177.1      TCP         54 5136 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   45 0.004235      192.168.213.128       192.168.177.1      TCP         54 4746 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   46 0.004331      192.168.213.128       192.168.177.1      TCP         54 5032 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

⊞ Frame 41: 635 bytes on wire (5080 bits), 635 bytes captured (5080 bits) on interface 0
⊞ Ethernet II, Src: Vmware_e8:61:ff (00:50:56:e8:61:ff), Dst: Vmware_ae:06:b7 (00:0c:29:ae:06:b7)
⊞ Internet Protocol Version 4, Src: 192.168.177.1, Dst: 192.168.213.128
⊞ Transmission Control Protocol, Src Port: 80 (80), Dst Port: 5209 (5209), Seq: 1, Ack: 1, Len: 581
⊞ Hypertext Transfer Protocol
⊞ Line-based text data: text/html

0000  00 0c 29 ae 06 b7 00 50  56 e8 61 ff 08 00 45 00   ..)....P V.a...E.
0010  02 6d 3b 5b 00 00 80 06  f5 5c c0 a8 b1 01 c0 a8   .m;[.... .\......
0020  d5 80 00 50 14 59 35 d4  5b 8f d5 8a 63 20 50 19   ...P.Y5. [...c P.
0030  fa f0 ff b9 00 00 48 54  54 50 2f 31 2e 31 20 33   ......HT TP/1.1 3
0040  30 31 20 4d 6f 76 65 64  20 50 65 72 6d 61 6e 65   01 Moved  Permane
```

Figure 19. Wireshark TCP Attack

Wireshark 3 way TCP Handshake:

```
((tcp.flags == 0x0002 || tcp.flags == 0x0012) && tcp.seq == 0) || (tcp.flags == 0x0010 && tcp.seq == 1 && tcp.ack <=1)
```

```
dumpfile={}

--Set filter to use as capture filter on next line

filter = "(tcp.flags == 0x02 && tcp.seq == 0) || (tcp.flags == 0x12 && tcp.seq == 0) || (tcp.flags == 0x10 && tcp.seq == 1)"  -- syn ack

-- tcp.flags

-- 0x10 = ack

-- 0x02 = syn

-- 0x12 = syn ack


--first frame

--syn, seq = 0

--tcp.flags = 0x02  tcp.seq = 0




--second frame

--syn ack, seq = 0

--tcp.flags = 0x12

--tcp.seq = 0
```

--third frame

--ack, seq = 1

--tcp.flags = 0x10

--tcp.seq = 1

-- Run tshark as shown on the following line

-- tshark -X lua_script:dumptofile_ack_packet.lua -i 4 -o tcp.relative_sequence_numbers:TRUE

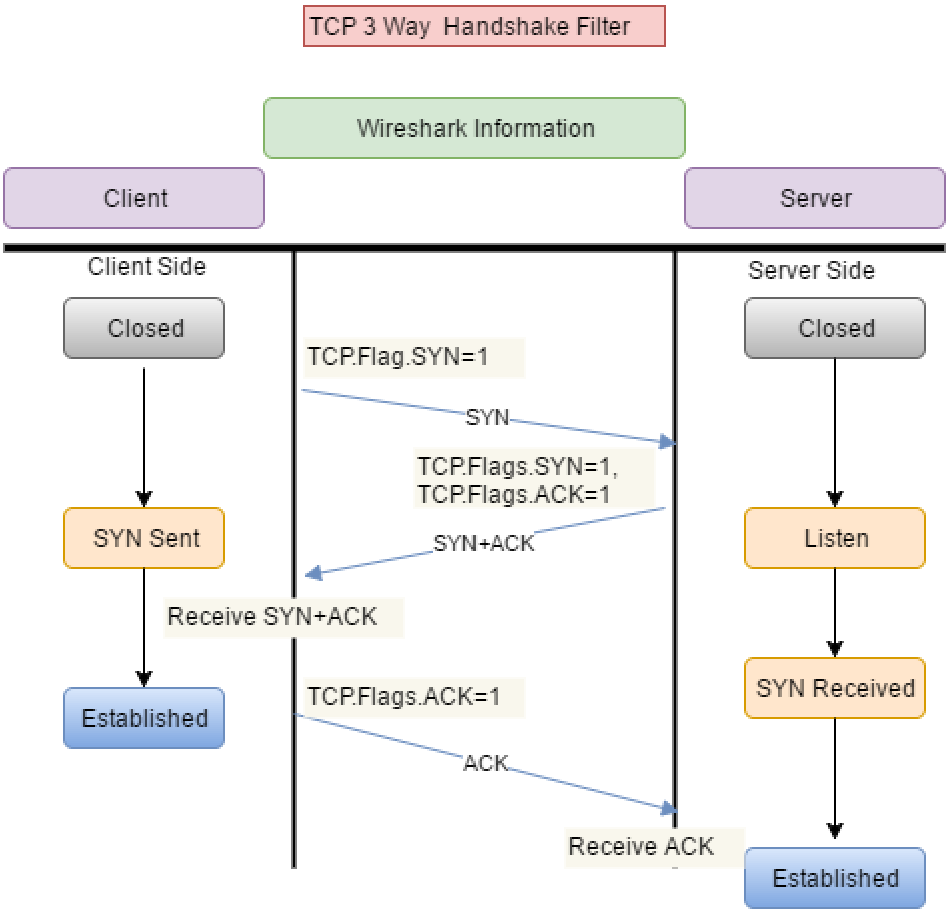
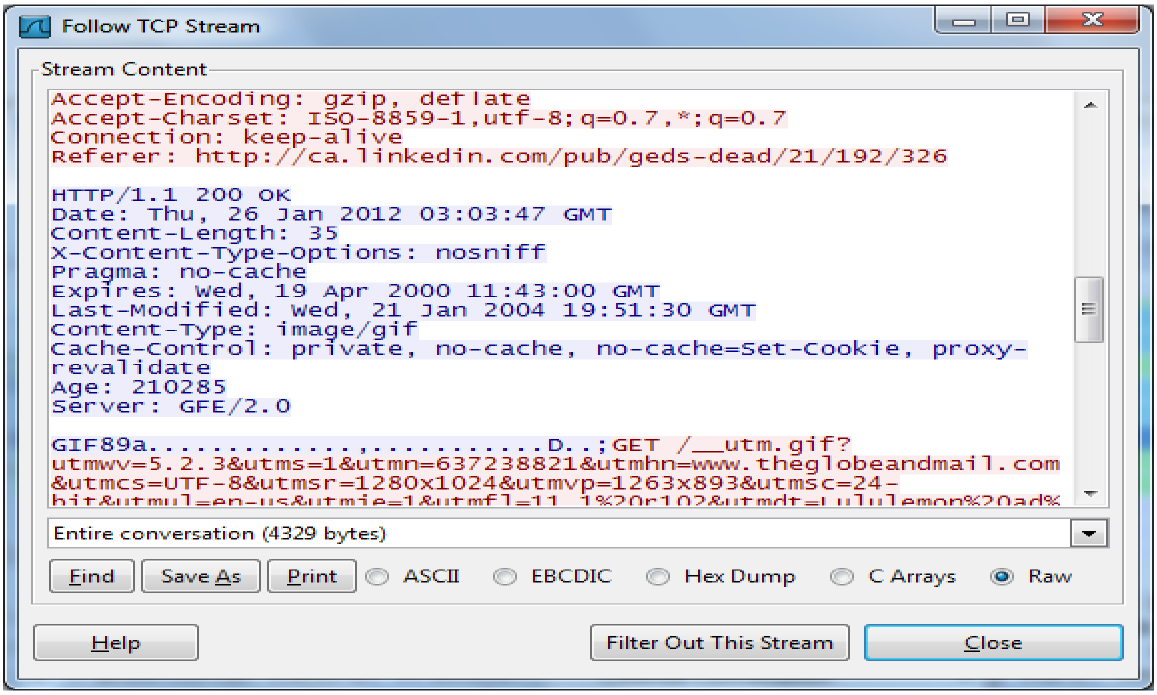


Figure 20. Wireshark 3-way handshake

Filter 1: tcp.flags.syn == 1 && tcp.flags.ack == 0

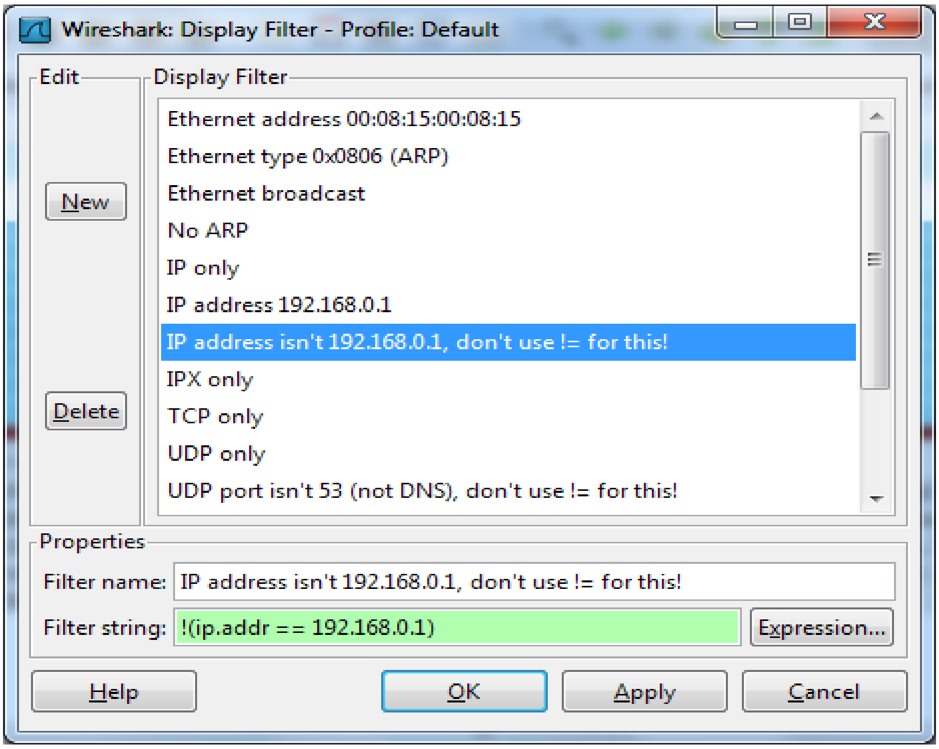
Filter 2: tcp.flags.syn == 1 && tcp.flags.ack == 1



```
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
Referer: http://ca.linkedin.com/pub/geds-dead/21/192/326

HTTP/1.1 200 OK
Date: Thu, 26 Jan 2012 03:03:47 GMT
Content-Length: 35
X-Content-Type-Options: nosniff
Pragma: no-cache
Expires: Wed, 19 Apr 2000 11:43:00 GMT
Last-Modified: Wed, 21 Jan 2004 19:51:30 GMT
Content-Type: image/gif
Cache-Control: private, no-cache, no-cache=Set-Cookie, proxy-
revalidate
Age: 210285
Server: GFE/2.0

GIF89a.............,...........D..;GET /__utm.gif?
utmwv=5.2.3&utms=1&utmn=637238821&utmhn=www.theglobeandmail.com
&utmcs=UTF-8&utmsr=1280x1024&utmvp=1263x893&utmsc=24-
bit&utmul=en-us&utmje=1&utmfl=11.1%20r102&utmdt=Lululemon%20ad%
```

Entire conversation (4329 bytes)

Figure 21. Wireshark TCP Stream Analysis



Ethernet address 00:08:15:00:08:15
Ethernet type 0x0806 (ARP)
Ethernet broadcast
No ARP
IP only
IP address 192.168.0.1
IP address isn't 192.168.0.1, don't use != for this!
IPX only
TCP only
UDP only
UDP port isn't 53 (not DNS), don't use != for this!

Properties

Filter name: IP address isn't 192.168.0.1, don't use != for this!

Filter string: !(ip.addr == 192.168.0.1)

Figure 22. Wireshark Display Filter

7.4. SQL Injection:

A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands.

The following server-side pseudo-code is used to authenticate users to the web application.

```
# Define POST variables

uname = request.POST['username']

passwd = request.POST['password']


# SQL query vulnerable to SQLi

sql = “SELECT id FROM users WHERE username=’” + uname + “’ AND password=’” + passwd + “’”


# Execute the SQL statement

database.execute(sql)
```

An attacker can also comment out the rest of the SQL statement to control the execution of the SQL query further.

-- MySQL, MSSQL, Oracle, PostgreSQL, SQLite

' OR '1'='1' --

' OR '1'='1' /*

-- ***MySQL***

' OR '1'='1' #

-- Access (using null characters)

' OR '1'='1' %00

' OR '1'='1' %16

Once the query executes, the result is returned to the application to be processed, resulting in an authentication bypass.

Threats from SQL Injection Attack:

- An attacker can use SQL Injection to bypass authentication users.
- One of SQL's primary functions is to select data based on a query and output the result of that query. An SQL Injection vulnerability could allow the complete disclosure of data residing on a database server.
- Since web applications use SQL to alter data within a database, an attacker could use SQL Injection to alter data stored in a database. Altering data affects data integrity and could cause repudiation issues, for instance, issues such as voiding transactions, altering balances and other records.
- SQL is used to delete records from a database. An attacker could use an SQL Injection vulnerability to delete data from a database. Even if an appropriate backup strategy is employed, deletion of data could affect an application's availability until the database is restored.
- Some database servers are configured (intentional or otherwise) to allow arbitrary execution of operating system commands on the database server. Given the right conditions, an attacker could use SQL Injection as the initial vector in an attack of an internal network that sits behind a firewall.

Threat Modelling:

- SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.
- SQL Injection is very common with PHP and ASP applications due to the prevalence of older functional interfaces. Due to the nature of programmatic interfaces available, J2EE and ASP.NET applications are less likely to have easily exploited SQL injections.
- The severity of SQL Injection attacks is limited by the attacker’s skill and imagination, and to a lesser extent, defence in depth countermeasures, such as low privilege connections to the database server and so on. In general, consider SQL Injection a high impact severity.

CHAPTER-8

SYSTEM TESTING

8.1. General:

Test your site to ensure that all pages display correctly, links go to the specified address, and images are not broken. It is important to test the site as you build it; do not wait until just before launch to begin testing. Also, test templates as you create them so any issues are resolved before creating other content based on the templates. Recruit as many people as possible to help you test.

Here are some general site testing guidelines:

- Test on the browsers and platforms your site visitors use
- Test on a variety of monitors (for example, LCD and CRT)
- View pages using different screen resolutions
- View pages using different colour settings
- Test all navigation and links
- Test items that can be downloaded (for example, PDF files)
- Test the search functionality
- Test site security
- If required, test for accessibility

8.2. Penetration Testing:

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

8.2.1. Command Execution:

What is Command Execution?

Command Execution is where a website application provides the ability to execute system commands.

What is a Command Injection Attack?

The purpose of the command injection attack is to inject and execute commands specified by the attacker in the vulnerable application. In situations like this, the application, which executes unwanted system commands, is like a pseudo system shell, and the attacker may use it as an authorized system user. Note, the commands are executed with the same privileges as the application and/or web server.

Command injection attacks are possible in most cases because of lack of correct input data validation, which can be manipulated by the attacker (forms, cookies, HTTP headers etc.).

What is Command Injection Harvesting?

Command Injection Harvesting is where a malicious user manipulates a website command execution application to render sensitive data.

If we browse to the" Command Execution" tab we are presented with a small PHP utility that allows us to ping remote machines. After a bit of fooling around I discovered you can make the utility execute multiple commands by chaining them together with the" &" character. Our end-game ploy in this demo is to remotely execute a PHP exploit. So, first of all we have to find a way to transfer our malicious payload to the remote machine. There are many ways to do this: ftp, tftp, inline transfer, web browser, … To get an idea of what we have to work with we can get a directory list of C:\WINDOWS\system32 which will contain binaries of the programs that are installed on the remote server. As we can see below we are in luck, tftp is installed on the remote machine (this is most practical transfer method for non-interactive command line execution).

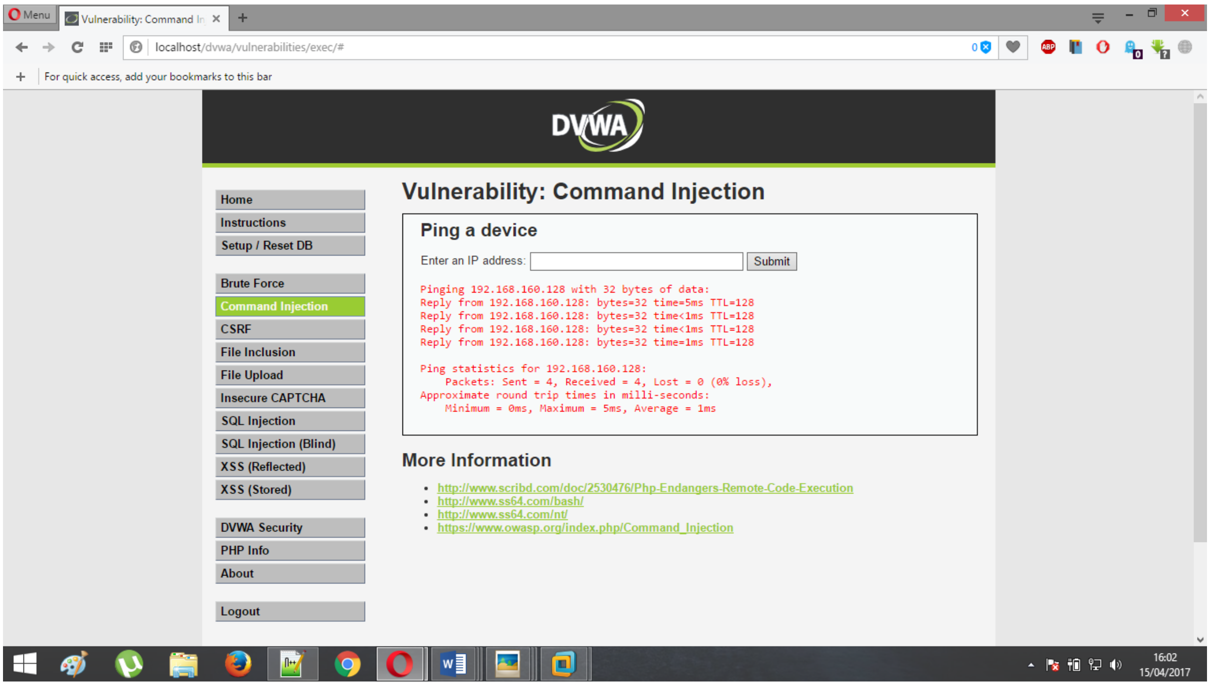
Instructions:

Click on Command Execution

Execute Ping

192.168.1.106

Click Submit



Vulnerability: Command Injection

Ping a device

Enter an IP address: [ ] Submit

```
Pinging 192.168.160.128 with 32 bytes of data:
Reply from 192.168.160.128: bytes=32 time=5ms TTL=128
Reply from 192.168.160.128: bytes=32 time<1ms TTL=128
Reply from 192.168.160.128: bytes=32 time<1ms TTL=128
Reply from 192.168.160.128: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.160.128:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 5ms, Average = 1ms
```

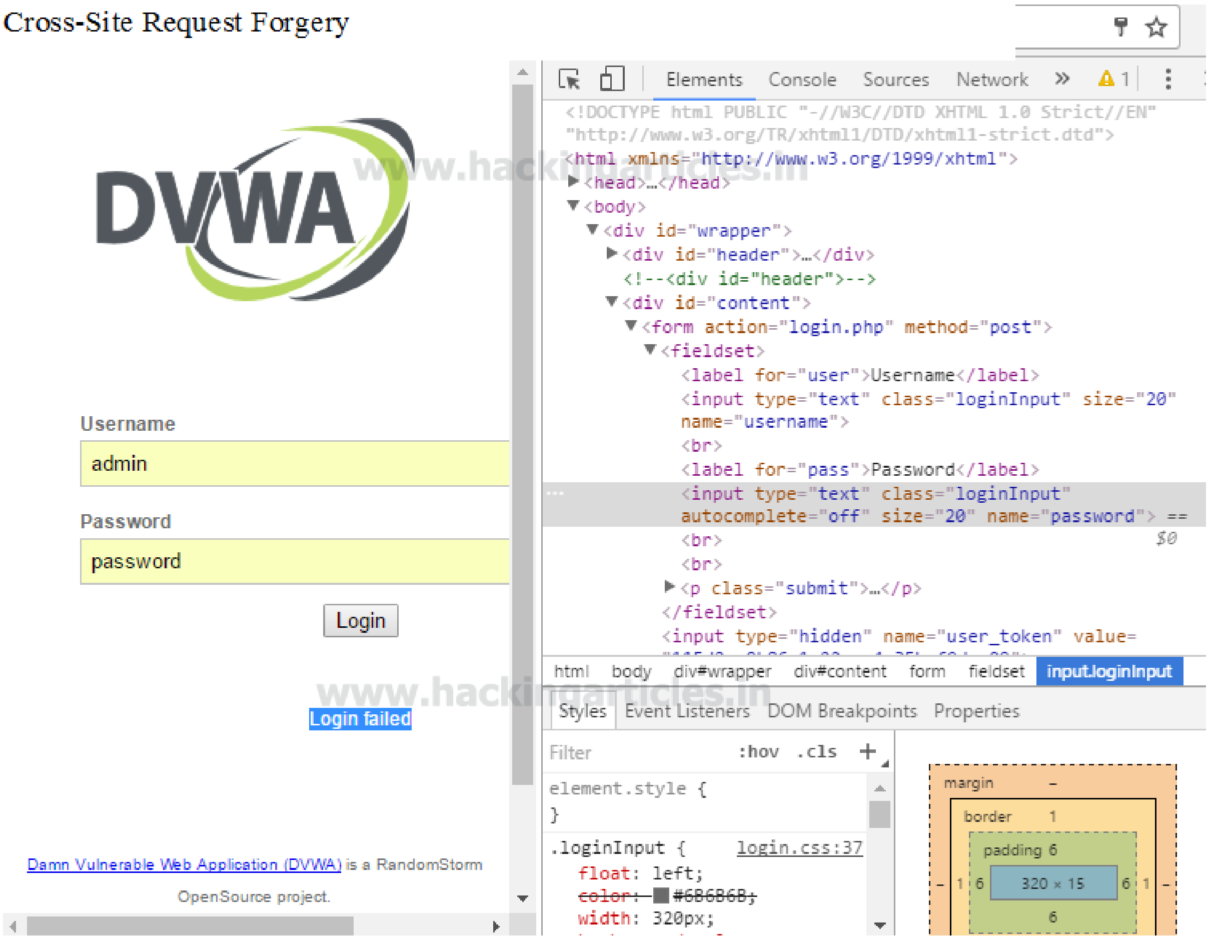
More Information

- http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://www.owasp.org/index.php/Command_Injection

Figure 23. DVWA Command Injection

8.2.2. CSRF (Cross-Site Request Forgery):

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.



Cross-Site Request Forgery

Username

admin

Password

password

Login

Login failed

Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 ▶<head>…</head>
 ▼<body>
   ▼<div id="wrapper">
     ▶<div id="header">…</div>
      <!--<div id="header">-->
     ▼<div id="content">
       ▼<form action="login.php" method="post">
         ▼<fieldset>
            <label for="user">Username</label>
            <input type="text" class="loginInput" size="20"
            name="username">
            <br>
            <label for="pass">Password</label>
            <input type="text" class="loginInput"
            autocomplete="off" size="20" name="password"> ==
            <br>                                             $0
            <br>
           ▶<p class="submit">…</p>
          </fieldset>
          <input type="hidden" name="user_token" value=
```

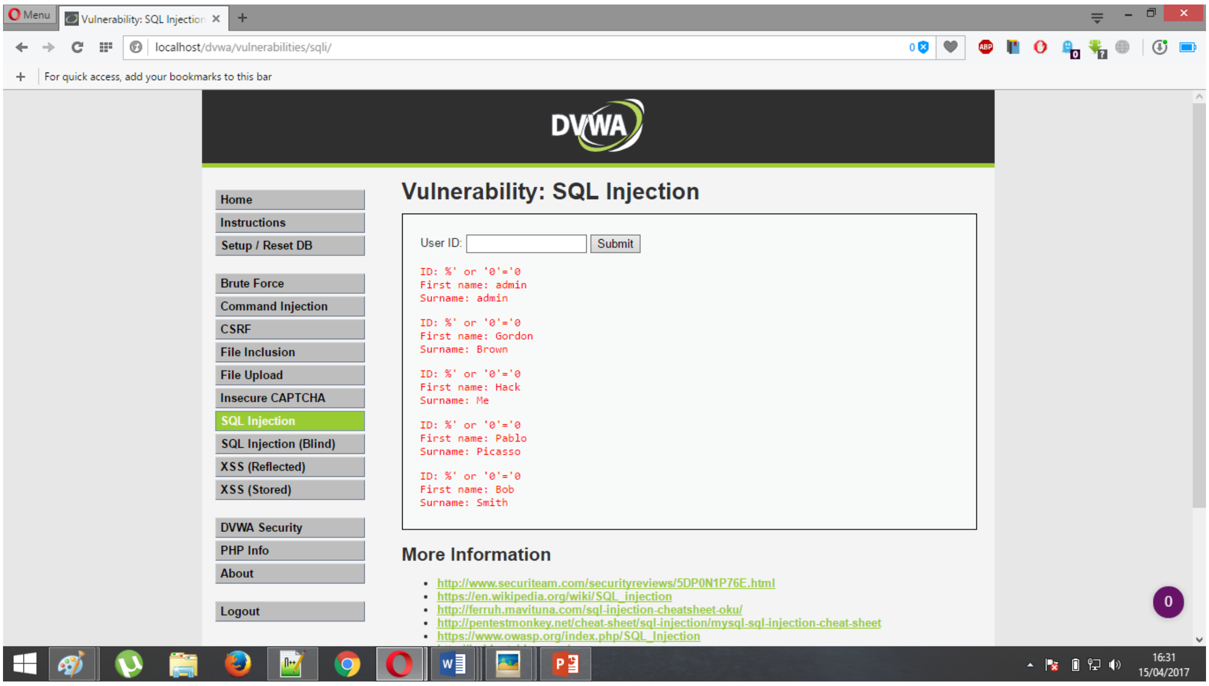
```
html  body  div#wrapper  div#content  form  fieldset  input.loginInput
```

```
element.style {
}
.loginInput {                    login.css:37
   float: left;
   ~~color: #6B6B6B;~~
   width: 320px;
```

Figure 24. DVWA CSRF

8.2.3. SQL Injection:

SQL injection (also known as SQL fishing) is a technique often used to attack data driven applications. This is done by including portions of SQL statements in an entry field in an attempt to get the website to pass a newly formed rogue SQL command to the database (e.g., dump the database contents to the attacker). SQL injection is a code injection technique that exploits a security vulnerability in an application's software. The vulnerability happens when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed.



Vulnerability: SQL Injection

```
ID: %' or '0'='0
First name: admin
Surname: admin

ID: %' or '0'='0
First name: Gordon
Surname: Brown

ID: %' or '0'='0
First name: Hack
Surname: Me

ID: %' or '0'='0
First name: Pablo
Surname: Picasso

ID: %' or '0'='0
First name: Bob
Surname: Smith
```

More Information

- http://www.securiteam.com/securityreviews/5DP0N1P76E.html
- https://en.wikipedia.org/wiki/SQL_injection
- http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
- http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
- https://www.owasp.org/index.php/SQL_Injection

Figure 25. SQL Injection

Instructions:

Input the below text into the User ID Textbox.

%' or '0'='0

Click Submit

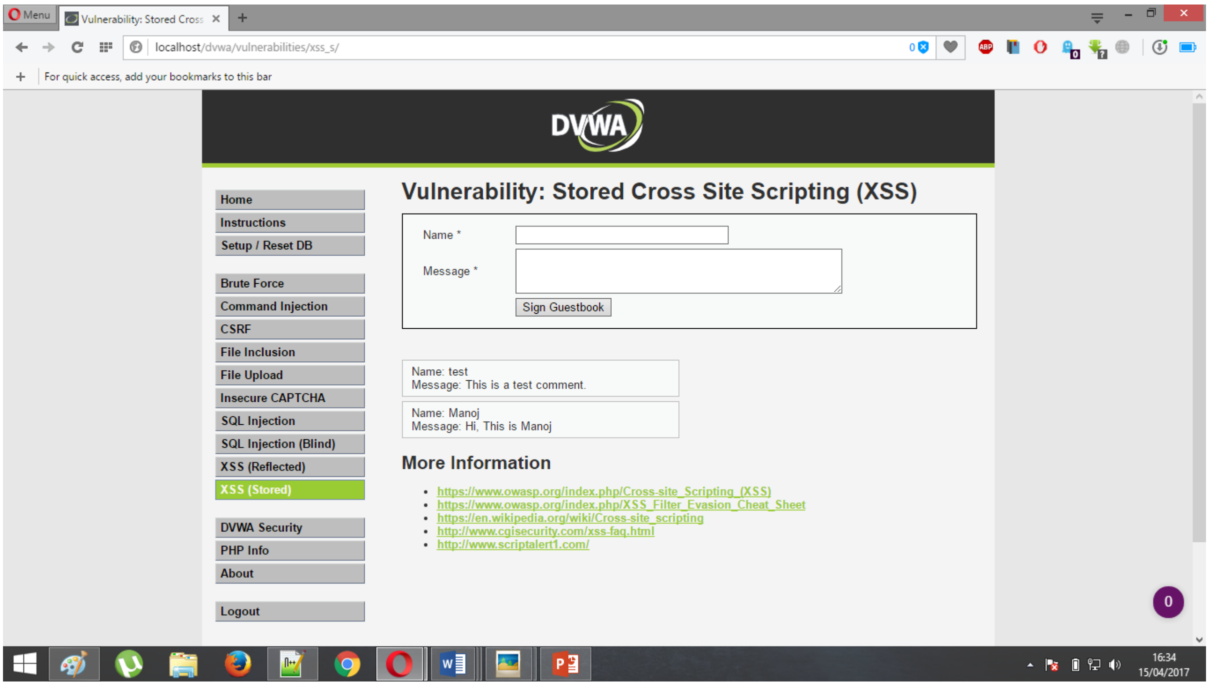
8.2.4. XSS (Cross Site Scripting):

What is Cross Site Scripting?

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in Web applications.

XSS enables attackers to inject client-side script into Web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same origin policy. In Addition, the attacker can send input (e.g., username, password, session ID, etc.) which can be later captured by an external script.

The victim's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site.



Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

Name: Manoj
Message: Hi, This is Manoj

More Information

- https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

Figure 26. DVWA XSS.

8.2.5. Upload:

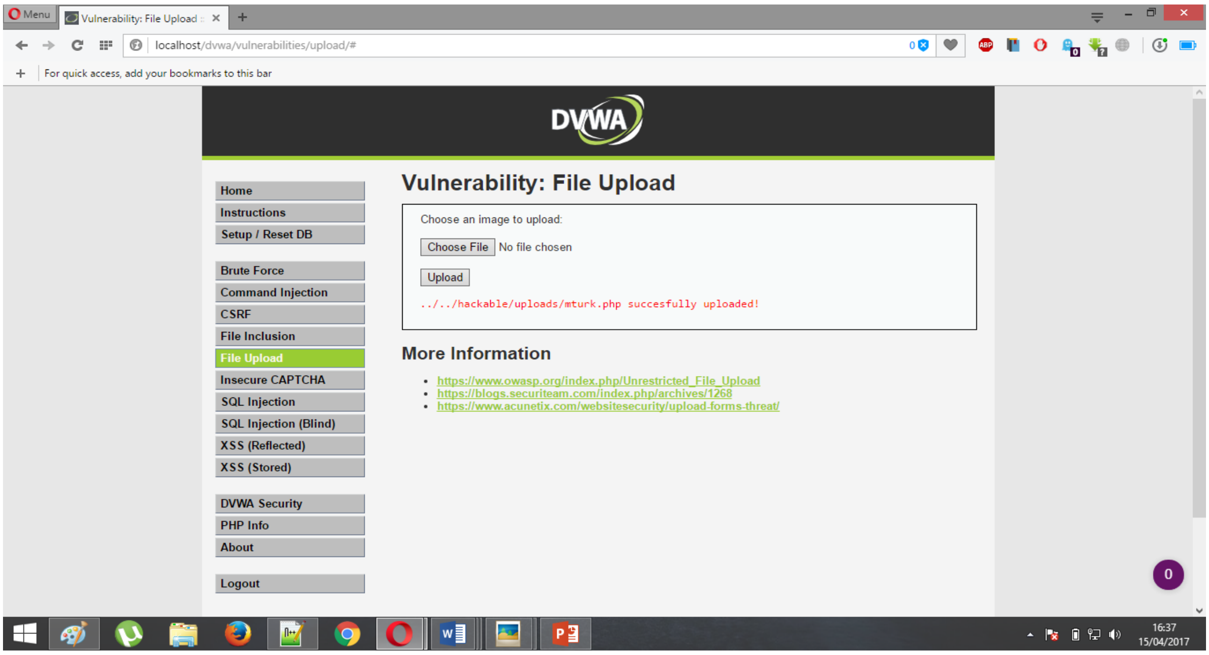
What is an Upload Attack Vector?

An Upload Attack Vector exists when a website application provides the ability to upload files. Uploaded files represent a significant risk to applications. The first step in many attacks is to get some code to the system to be attacked. Then the attack only needs to find a way to get the code executed. Using a file upload helps the attacker accomplish the first step. The consequences of unrestricted file upload can vary, including complete system takeover, an overloaded file system, forwarding attacks to backend systems, and simple defacement. It depends on what the application does with the uploaded file, including where it is stored.

What is c99.php?

The c99 PHP utility provides functionality for listing files, brute-forcing FTP passwords, updating itself, executing shell commands and PHP code. It also provides for connecting to

MySQL databases, and initiating a connect-back shell session. In many ways, it can be considered the web equivalent of the rootkits that successful attackers often download. In other ways, it is the malware equivalent of PHPShell itself. c99 is often one of the utility programs that is either downloaded if a web server is vulnerable due to being misconfigured.



Vulnerability: File Upload

Choose an image to upload:

Choose File No file chosen

Upload

```
../../hackable/uploads/mturk.php succesfully uploaded!
```

More Information

- https://www.owasp.org/index.php/Unrestricted_File_Upload
- https://blogs.securiteam.com/index.php/archives/1268
- https://www.acunetix.com/websitesecurity/upload-forms-threat/

Figure 27. DVWA File Upload

CHAPTER-9

CODING

Session File:

```
<?php
 include('config.php');
session_start();
$user_check = $_SESSION['login_user'];
$ses_sql = mysqli_query($db,"select username from admin where username = '$user_check' ");
 $row = mysqli_fetch_array($ses_sql,MYSQLI_ASSOC);
$login_session = $row['username'];
if(!isset($_SESSION['login_user'])){
header("location:login2.php");
  }
?>
```

Database Store:

```
<?php
define('DB_NAME','database');
define('DB_USER','root');
define('DB_PASSWORD',");
define('DB_HOST','localhost');
$link=mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
if(!$link){
```

```
die('could not connect:'.mysql_error());

}
$db_selected=mysql_select_db(DB_NAME, $link);
if(!$db_selected){
        die('can\'t use' . DB_NAME . ':' .mysql_error());

}
$value1=$_POST['Full_Name'];
$value2=$_POST['Emai_id'];
$value3=$_POST['mobile'];
$value4=$_POST['username'];
$value5=$_POST['passcode'];
$sql="INSERT INTO register  (Full_Name,Emai_id,mobile,username, passcode)
values ('$value1' , '$value2','$value3','$value4','$value5')";
if(!mysql_query($sql)){
        die('error:'.mysql_error());
}
echo'<html><title>Successfull signup</title><body bgcolor="yellow"><font
color="blue">Your account has been created <br><br><br>
Click here to login----> <a href="index.php">Login</a></font></body></html>';
mysql_close();
?>
```

Logout:

```
<?php
  session_start();

  if(session_destroy()) {
    header("Location: index.php");
  }
?>
```

Register:

```
<html><head>
<title>Registration</title>
<link rel="shortcut icon" href="swami.ico" type="image/x-icon"/>
</head>
<body bgcolor="lightgreen"></body><br><br>
<div align = "center">
        <div style = "width:400px; border: solid 2px blue; " align = "left">
<center>
<h1 style="color:red">Registration</h1><hr color="yellow">
<h2 style="color:blue">Fill your complete details</h2><hr color="yellow">

<form action="store.php" method="post">
<b>Full Name:&nbsp;&nbsp;<input type="text" name="Full_Name"><br><br>
Emai-id:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text"
name="Emai_id"><br><br>
Mobile No.&nbsp;:&nbsp;&nbsp;<input type="text" name="mobile"><br><br>
Username:&nbsp;&nbsp;&nbsp;<input type="text" name="username"><br><br>

Password:&nbsp;&nbsp;&nbsp;<input type="password" name="passcode"> <hr
color="yellow">

<input type="Submit" value="   Register   "
Style="color:blue">&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="   Reset
" Style="color:blue">  </b>

<center> </form> </div> </div> </div>
</html>
```

Successfully Login:

```
<?php
  include('session.php');
?>
<html">

  <head>
    <title>Welcome </title>
    <link rel="icon" href="swami.ico" type="image/x-icon">
  </head>

  <body bgcolor="lightgreen">
    <h3>Welcome <u><b><?php echo $login_session; ?><b></u> you have successfully loged in</h3>
    <a href = "logout.php">Sign Out</a>
  </body>

</html>
```

Configuration:

```
<?php
  define('DB_SERVER', 'localhost');
  define('DB_USERNAME', 'root');
  define('DB_PASSWORD', '');
  define('DB_DATABASE', 'database');
  $db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
?>
```

CHAPTER-10

10. SECURING CMS:

Based on this project we have defined list of best practices for securing CMS. Most important of them are:

- Regularly updating CMS, programming and database support, web server and server itself. If all of that is done regularly it’s almost impossible to hack major CMS applications,
- Limiting logging attempts, obscuring admin page and installation itself, enforcing strong passwords, use discreet error messages, regularly check for proper permissions, limit IP and country access if necessary.
- Deleting known IDs and names for administration access (like Admin) and use random IDs for admin accounts,
- Using professional hosting services with security teams specially dedicated to monitor security issues with CMS of a choice,
- Using professional CDN service with option to automatically add new and recommended firewall rules.
- Using web firewall and security plugins (like Themes Security or Wordfence) is not effective in case previously defined important factors are not covered. Experts demonstrated how fully patched systems with all security plugins in place can still be compromised. This could be countered if hosting is moved in cloud environment. Since attackers are using standardized exploit kits and standardized codes massively it’s easy to spot breach in cloud environment if cloud hosting service is hosting thousands of WordPress installations for example. If there’s communication with development team of CMS application than rules can be quickly added or changed easily. For specific most used web applications attacks are detected daily and rules are added immediately to prevent them.

CHAPTER-11

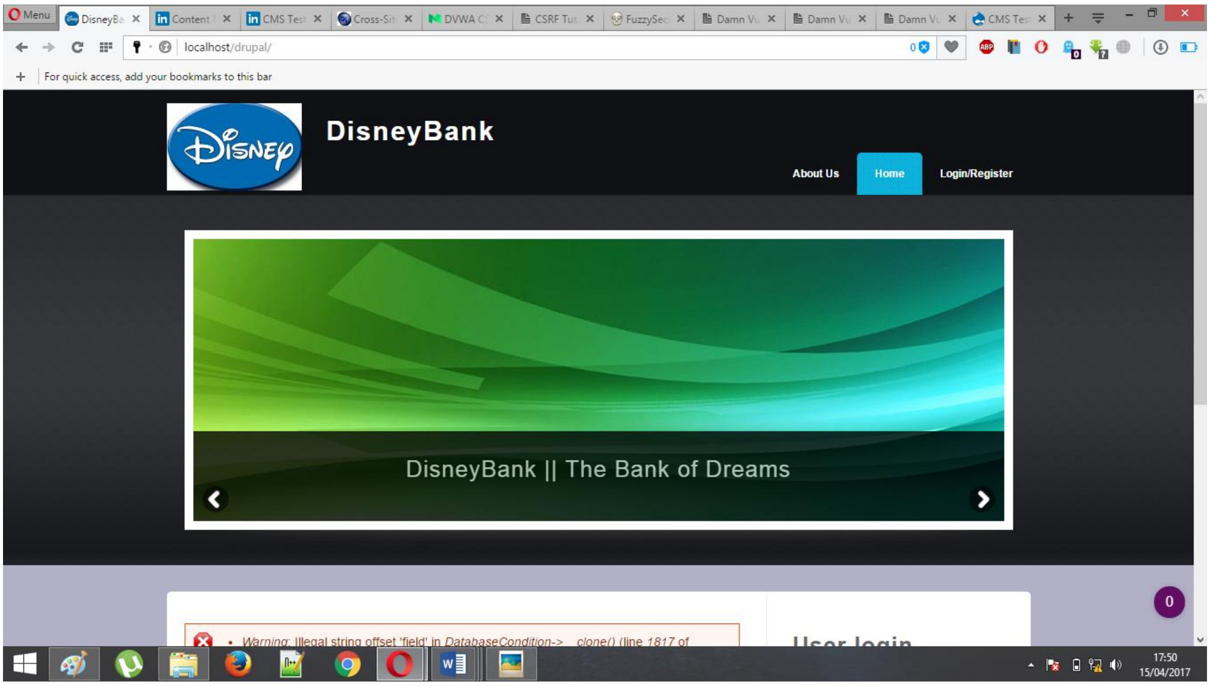
CONCLUSION

In this project, basic features of CMSs have been presented. Critical web vulnerabilities against web applications have been explained. Each vulnerability has been tested on the basic installation of the tested CMSs (Drupal) and possible security measures have been listed. The test results of the CMS systems have been presented as well as the list of threats against which the tested CMS systems were resistant in their basic installations. Based on findings the vulnerabilities we can conclude that servers, web servers and database servers used to host most popular CMS explained in project are not less secure than those used to host other CMS if proposed alternatives are used. We also concluded that number of discovered CMS vulnerabilities in three major CMS is correlated with their market share. Analysis of most frequently reported types of attacks showed that attacks on three major CMS are mainly using SQL injection vulnerabilities or CSRF in combination with SQL injection. WordPress development team is good example of such community where automated security updates are created and deployed as soon as threat is discovered. Current project confirms that hosting CMS without cloud web protection is becoming increasingly vulnerable even if all other measures are taken. The resistance of the tested CMSs to each of the ten analysed critical threats against web applications may be achieved by installing add-ons (plugin, extension) and the web server and network resource settings.

APPENDICES

APPENDIX:

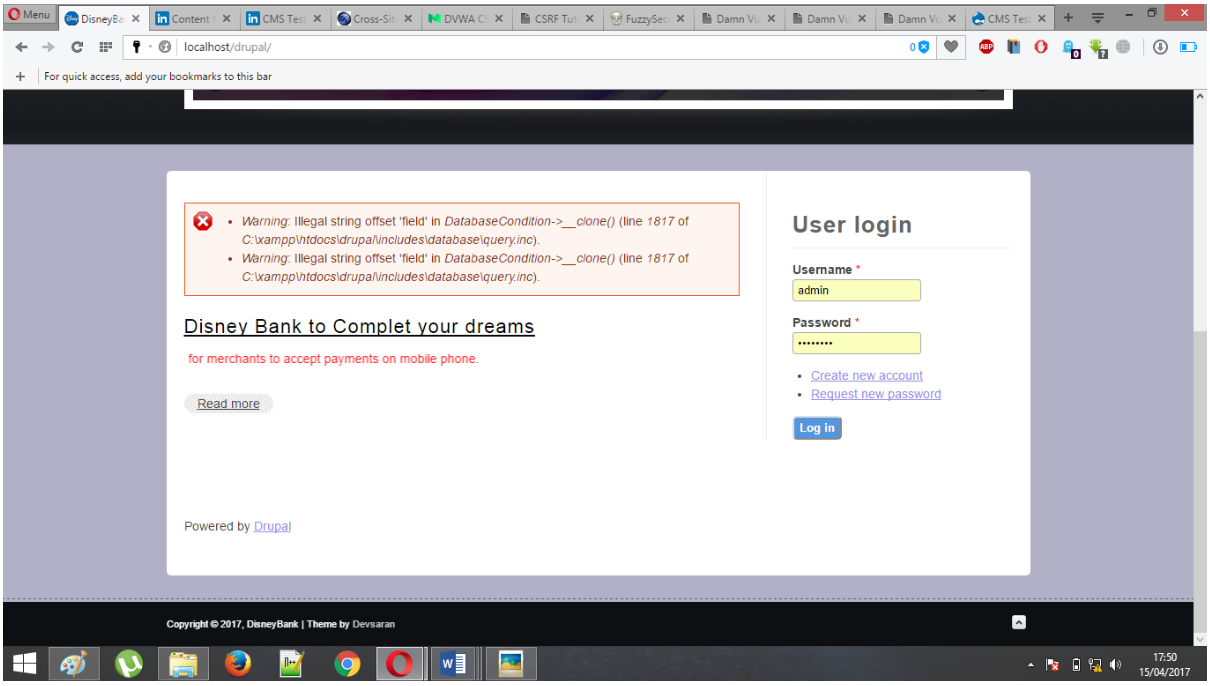
Drupal website:



- *Warning*: Illegal string offset 'field' in *DatabaseCondition->__clone()* (line *1817* of

Figure 28. Drupal Website Main

Drupal Login:



- *Warning*: Illegal string offset 'field' in *DatabaseCondition->__clone()* (line *1817* of *C:\xampp\htdocs\drupal\includes\database\query.inc*).
- *Warning*: Illegal string offset 'field' in *DatabaseCondition->__clone()* (line *1817* of *C:\xampp\htdocs\drupal\includes\database\query.inc*).

Disney Bank to Complet your dreams

for merchants to accept payments on mobile phone.

Username * admin

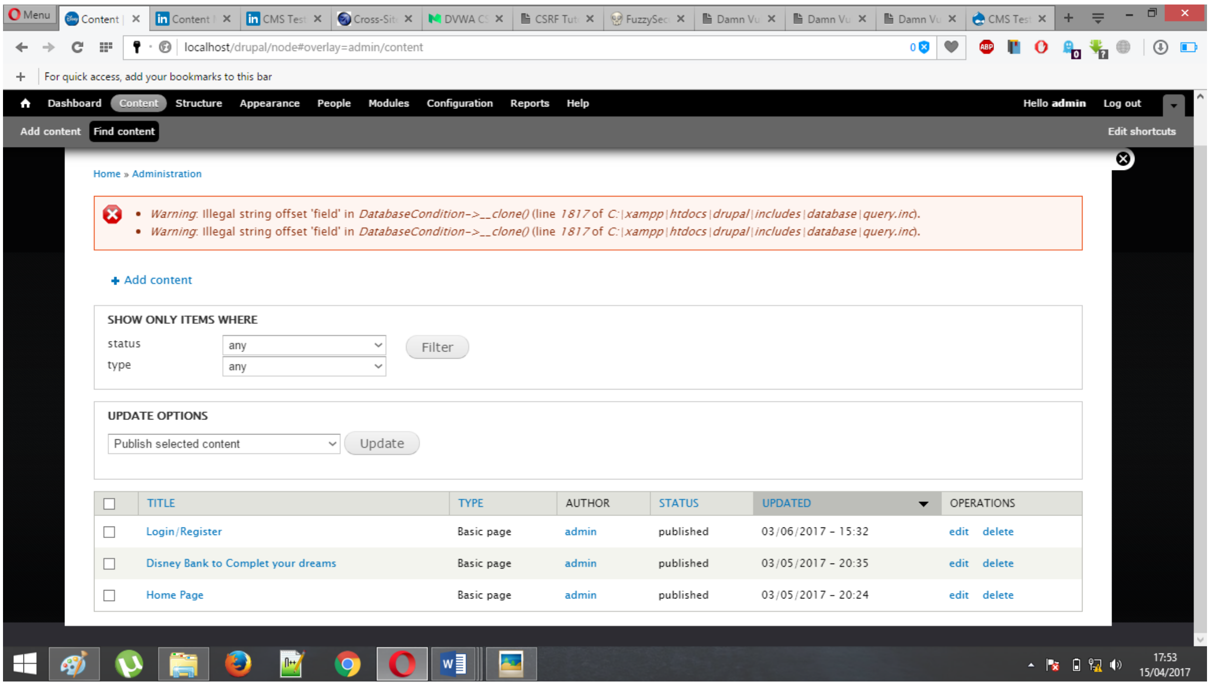
Password * ••••••••

Powered by Drupal

Copyright © 2017, DisneyBank | Theme by Devsaran

Figure 29. Drupal Login

Drupal Content Dashboard:



- *Warning*: Illegal string offset 'field' in *DatabaseCondition->__clone()* (line *1817* of *C:|xampp|htdocs|drupal|includes|database|query.inc*).
- *Warning*: Illegal string offset 'field' in *DatabaseCondition->__clone()* (line *1817* of *C:|xampp|htdocs|drupal|includes|database|query.inc*).

SHOW ONLY ITEMS WHERE

status: any

type: any

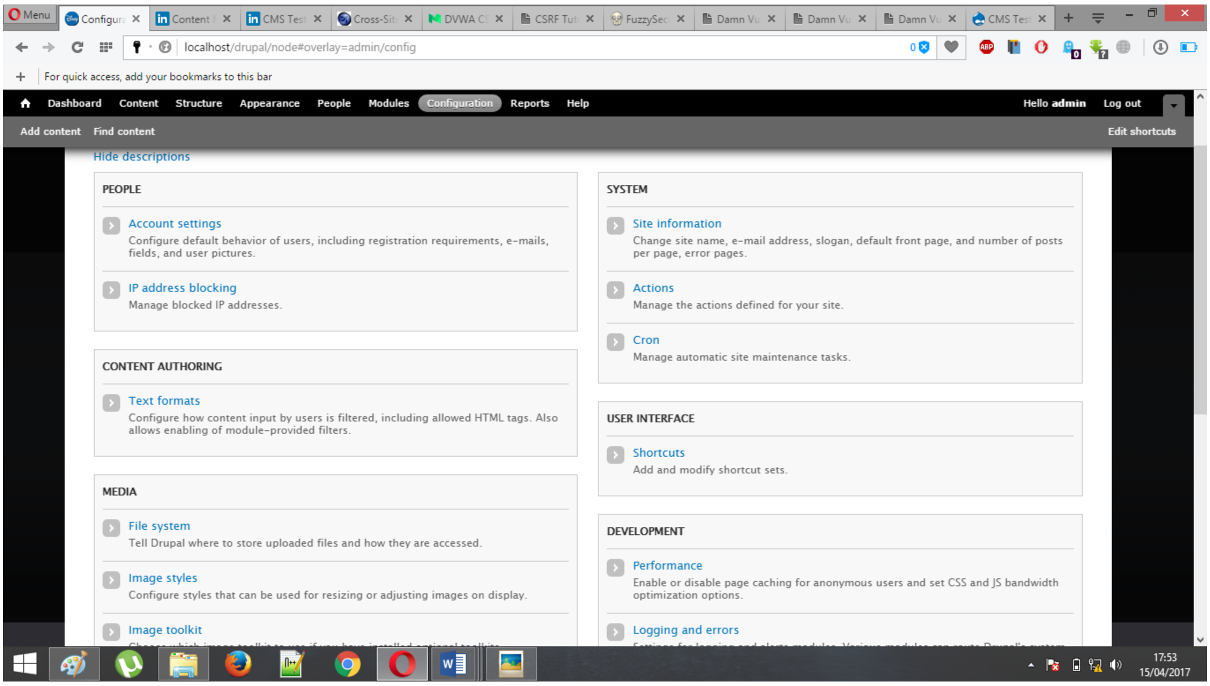
UPDATE OPTIONS

Publish selected content

□	TITLE	TYPE	AUTHOR	STATUS	UPDATED	OPERATIONS
□	Login/Register	Basic page	admin	published	03/06/2017 - 15:32	edit delete
□	Disney Bank to Complet your dreams	Basic page	admin	published	03/05/2017 - 20:35	edit delete
□	Home Page	Basic page	admin	published	03/05/2017 - 20:24	edit delete

Figure 30. Drupal Dashboard

Drupal Configuration:



PEOPLE

Account settings
Configure default behavior of users, including registration requirements, e-mails, fields, and user pictures.

IP address blocking
Manage blocked IP addresses.

CONTENT AUTHORING

Text formats
Configure how content input by users is filtered, including allowed HTML tags. Also allows enabling of module-provided filters.

MEDIA

File system
Tell Drupal where to store uploaded files and how they are accessed.

Image styles
Configure styles that can be used for resizing or adjusting images on display.

Image toolkit

SYSTEM

Site information
Change site name, e-mail address, slogan, default front page, and number of posts per page, error pages.

Actions
Manage the actions defined for your site.

Cron
Manage automatic site maintenance tasks.

USER INTERFACE

Shortcuts
Add and modify shortcut sets.

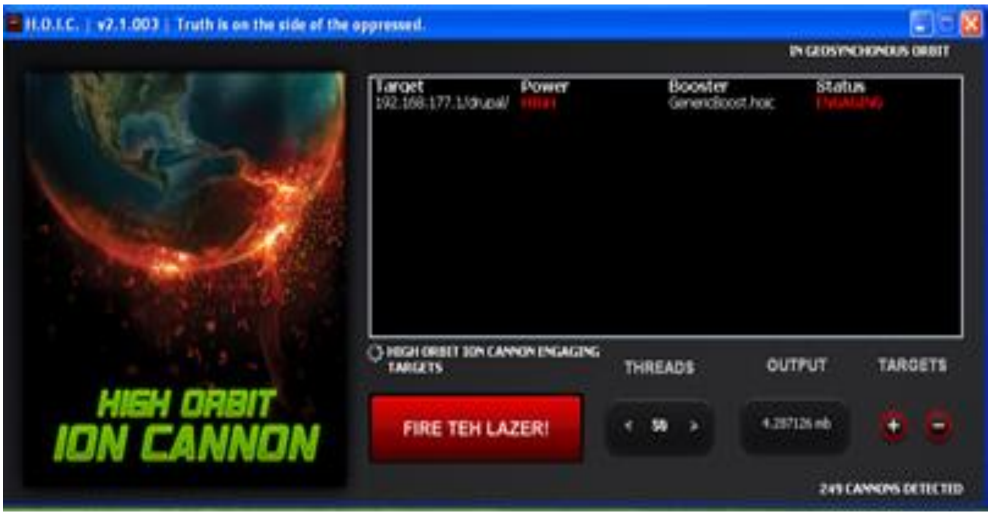
DEVELOPMENT

Performance
Enable or disable page caching for anonymous users and set CSS and JS bandwidth optimization options.

Logging and errors

Figure 31. Drupal Configuration

HOIC Attack:



H.O.I.C. | v2.1.003 | Truth is on the side of the oppressed.

Target	Power	Booster	Status
192.168.177.1/drupal/	HIGH	GenericBoost.hoic	ENGAGING

HIGH ORBIT ION CANNON ENGAGING TARGETS

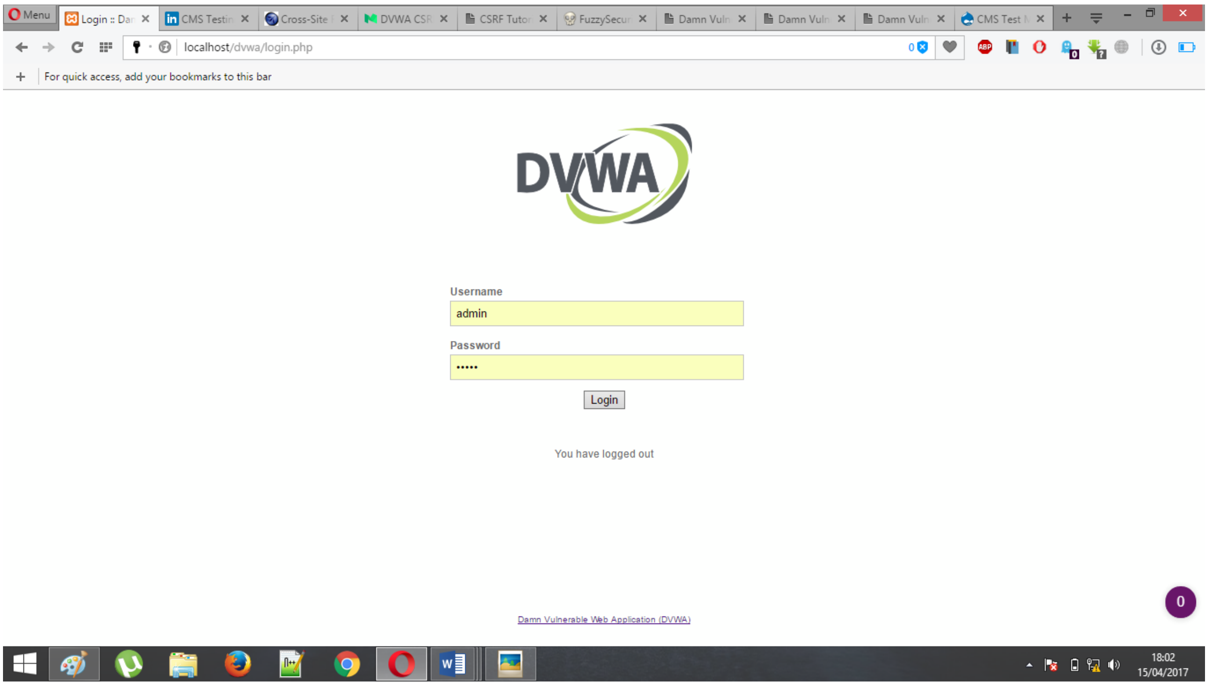
THREADS: 50

OUTPUT: 4.287126 mb

249 CANNONS DETECTED

Figure 32. HOIC Attack

DVWA Login:



localhost/dvwa/login.php

Username: admin

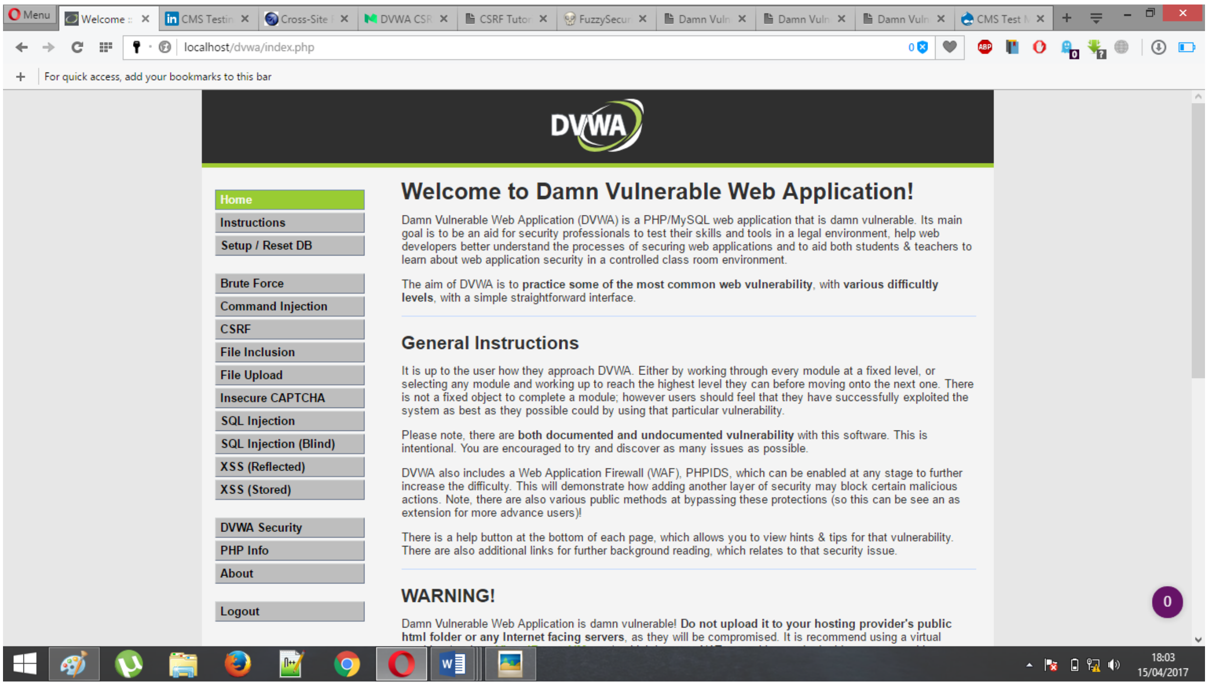
Password: •••••

You have logged out

Damn Vulnerable Web Application (DVWA)

Figure 33. DVWA Login

DVWA Dashboard:



Welcome to Damn Vulnerable Web Application!

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

The aim of DVWA is to **practice some of the most common web vulnerability**, with **various difficultly levels**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be see an as extension for more advance users)!

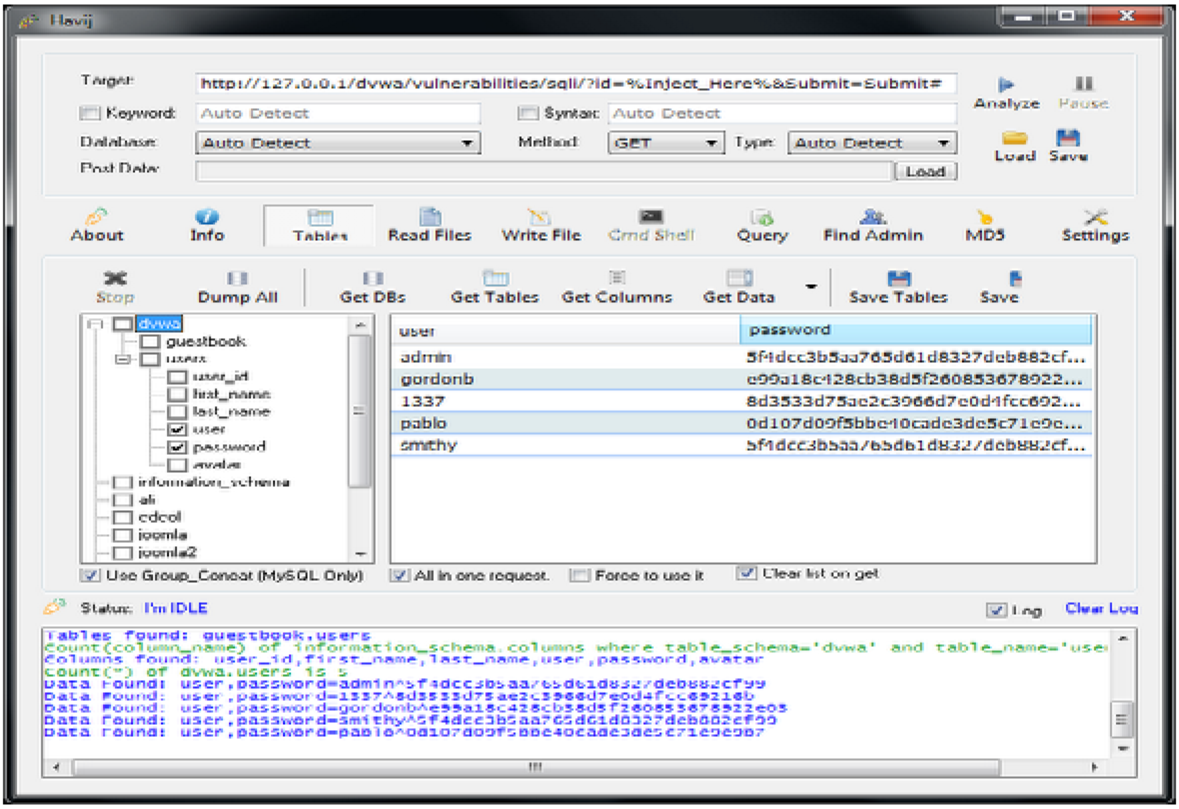
There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual

Figure 34. DVWA Dashboard

Havij:



Target: http://127.0.0.1/dvwa/vulnerabilities/sqli/?id=%Inject_Here%&Submit=Submit#

□ Keyword: Auto Detect

□ Syntax: Auto Detect

Database: Auto Detect

Method: GET

Type: Auto Detect

Post Data:

□ dvwa
- □ guestbook
- □ users
  - □ user_id
  - □ first_name
  - □ last_name
  - ☑ user
  - ☑ password
  - □ avatar
- □ information_schema
- □ ali
- □ cdcol
- □ joomla
- □ joomla2

user	password
admin	5f4dcc3b5aa765d61d8327deb882cf...
gordonb	e99a18c428cb38d5f260853678922...
1337	8d3533d75ae2c3966d7e0d4fcc692...
pablo	0d107d09f5bbe40cade3de5c71e9e...
smithy	5f4dcc3b5aa765d61d8327deb882cf...

☑ Use Group_Concat (MySQL Only) ☑ All in one request. □ Force to use it ☑ Clear list on get

Status: I'm IDLE ☑ Log

```
Tables found: guestbook,users
count(column_name) of information_schema.columns where table_schema='dvwa' and table_name='use
Columns found: user_id,first_name,last_name,user,password,avatar
count(*) of dvwa.users is 5
Data Found: user,password=admin^5f4dcc3b5aa765d61d8327deb882cf99
Data Found: user,password=1337^8d3533d75ae2c3966d7e0d4fcc69216b
Data Found: user,password=gordonb^e99a18c428cb38d5f260853678922e03
Data Found: user,password=smithy^5f4dcc3b5aa765d61d8327deb882cf99
Data Found: user,password=pablo^0d107d09f5bbe40cade3de5c71e9e9b7
```

Figure 36. SQL Injection Attack

REFERENCES:

1. MIPRO 2016, May 30 - June 3, 2016, Opatija, Croatia, “Securing web content and services in open source content management systems” H. Jerković, P. Vranešić, S. Dadić.
2. M. Kaluža, B. Vukelic, T. Rojko: “Content management system security” Zbornik Veleucilišta u Rijeci, Vol. 4 (2016.), No.1, pp. 29-44
3. J. Ji, J. Kim, Y. Kim, S. Jung, C. Lee, D. Kim, et al., "Design and Implementation of the Korean Style Plug-In using the WordPress. "S. a. t. i. t. u. o. c. m. s.-. W3Techs. (2016). Usage Statistics and Market Share of Content Management Systems for Websites
4. S. K. Patel, V. R. Rathod, and J. B. Prajapati,"Comparative analysis of web security in open source content management system," in Intelligent Systems and Signal Processing (ISSP), 2013 International Conference on, 2013, pp. 344-349.
5. A. Onishi, "Security and Performance," in Pro WordPress Theme Development, published: Springer, 2013, pp. 297-332.
6. R. Ismailova, "Web site accessibility, usability and security: a survey of government web sites in Kyrgyz Republic," Universal Access in the Information Society, pp. 1-8, 2015.
7. S. Mansfield-Devine, "Taking responsibility for security," Computer Fraud & Security, vol. 2015, pp. 15-18, 2015.
8. J. C. Coelho Martins da Fonseca and M. P. Amorim Vieira, "A Practical Experience on the Impact of Plugins in Web Security," in Reliable Distributed Systems (SRDS), 2014 IEEE 33rd International Symposium on, 2014, pp. 21-30.
9. T. Koskinen, P. Ihantola, and V. Karavirta, "Quality of WordPress plug-ins: an overview of security and user ratings," in Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom), 2012, pp. 834-837.
10. A. Algarni and Y. Malaiya, "Software vulnerability markets: Discoverers and buyers," International Journal of Computer, Information Science and Engineering, vol. 8, pp. 71-81, 2014.
11. A. Mirdha, A. Jain, and K. Shah, "Comparative analysis of open source content management systems," in Computational Intelligence and Computing Research (ICCIC), 2014 IEEE International Conference on, 2014, pp. 1-4.
12. P. J. Costa Nunes, J. Fonseca, and M. Vieira, "phpSAFE: A Security Analysis Tool for OOP 1406.