**Vulnerability Assessment and Penetration Testing**

**A Minor Project Report Submitted to**



**Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal**

**Towards Partial Fulfillment for the Award of**

**Bachelor of Technology**

**(Computer Science and Engineering)**

**Submitted By:**

**Siddharth Jain(0827CS171211)**

**Shubham Malviya(0827CS171208)**

**Siddharth Choudhary (0827CS171210)**

**CS4 III YEAR**

**Supervised By: Guided by:**

**Mr. Ritesh Khedekar Mrs. Vandana Kate**

**Ms. Nupur Agrawal**

****

**Department of Computer Science and Engineering**

**Acropolis Institute of Technology and Research, Indore**

**July - Dec 2019**

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. NO.** | **TOPIC** | **PAGE NO.** | **REMARK** |
| 1. | Introduction |  |  |
|  | 1.1 RATIONALE |  |  |
|  | 1.2 problem DEFINTION & SOLUTION |  |  |
|  | 1.3 objectives |  |  |
|  | 1.4 SCOPE |  |  |
|  | 1.5 REPORT ORGANIZATION |  |  |
| 2 | LITERATURE SURVEY |  |  |
|  | 2.1 BACKGROUND |  |  |
|  | 2.2 RELATED WORK |  |  |
| 3 | SYSTEM REQUIREMENTS SPECIFICATION |  |  |
|  | 3.1 HARDWARE REQUIREMENTS |  |  |
|  | 3.2 SOFTWARE REQUIREMENTS |  |  |
| 4 | SYSTEM ANALYSIS & DESIGN |  |  |
|  | 4.1 SCOPING |  |  |
|  | 4.2 INFORMATION GATHERING |  |  |
|  | 4.3 VULNERABILITY SCANNING |  |  |
|  | 4.4 FALSE POSITIVE ANALYSIS |  |  |
|  | 4.5 VULNERABILITY EXPLOITATION |  |  |
|  | 4.6 REPORT GENERATION |  |  |
| 5 | TESTING |  |  |
|  | 5.1 TESTING TOOLS |  |  |
|  | 5.2 TESTING REPORT |  |  |
| 6 | FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS |  |  |
|  | 6.1 FUNCTIONAL REQUIREMENT |  |  |
|  | 6.2 NON FUNCTIONAL REQUIREMENT |  |  |
| 7 | CONCLUSION |  |  |

1. **INTRODUCTION**

Hacker Vs Ethical Hacker

* **Hacker**: A person who invades or interferes with another system with the intent to cause harm, without having any permission from the system owner.
* **Ethical hacker**: A professional hired by an organization to review its security posture from the eyes of the hacker. Ethical hackers test vulnerabilities of the systems.
  1. **RATIONALE**

A vulnerability assessment is the process of defining, identifying, classifying and prioritizing vulnerabilities in computer systems, applications and network infrastructures and providing the organization doing the assessment with the necessary knowledge, awareness and risk background to understand the threats to its environment and react appropriately.A vulnerability assessment process that is intended to identify threats and the risks they pose typically involves the use of automated testing tools, such as [network security scanners](https://searchsecurity.techtarget.com/definition/vulnerability-scanning), whose results are listed in a vulnerability assessment report.Organizations of any size, or even individuals who face an increased risk of cyberattacks, can benefit from some form of vulnerability assessment, but large enterprises and other types of organizations that are subject to ongoing attacks will benefit most from vulnerability analysis.Because security [vulnerabilities](https://whatis.techtarget.com/definition/vulnerability) can enable hackers to access IT systems and applications, it is essential for enterprises to identify and remediate weaknesses before they can be exploited. A comprehensive vulnerability assessment along with a management program can help companies improve the security of their systems.

A penetration test, also known as a pen test, is a simulated cyber-attack against your computer system to check for exploitable vulnerabilities. In the context of web application security, penetration testing is commonly used to augment a [web application firewall (WAF)](https://www.imperva.com/products/web-application-firewall-waf/).Pen testing can involve the attempted breaching of any

number of application systems, (e.g., application protocol interfaces (APIs), frontend/backend servers) to uncover vulnerabilities, such as unsanitized inputs that are susceptible to code injection attacks.Insights provided by the penetration test can be used to fine-tune your WAF security policies and patch detected vulnerabilities.

**1.2 PROBLEM DEFINITION AND PROPOSED SOLUTION**

Now, a day we can see Cyber crime increasing day by day. So it is necessary to secure the website site from following attacks, towards user or organization:

* **Keep Software update:** When website security holes are found in software, hackers are quick to attempt to abuse them.
* **SQL injection:** SQL injection attacks are when an attacker uses a web form field or URL parameter to gain access to or manipulate your database.
* **Protect against XSS attacks:** Cross-site scripting (XSS) attacks inject malicious JavaScript into your pages.
* **Beware of error and message:** Be careful with how much information you give away in your error messages.
* **Use HTTPS:** HTTPS is a protocol used to provide security over the Internet.

**Payment Gateway:** With the help of cookies, session, tampering payment gateway can be bypass easily.

A vulnerability assessment process that is intended to identify threats and the risks they pose typically involves the use of automated testing tools, such as network security scanners, whose results are listed in a vulnerability assessment report.

**1.3 OBJECTIVES**

1. The main objective of penetration testing is secure the organization from attacks like, SQL injection, XSS, DOS, data tampering, etc.
2. Information Security, Network Security and Application Security.

**1.4 SCOPES**

The scopes of this project are:

* Protect from cyber attacks.
* Identify the possible vulnerability for a website.
* To ensure the security of website.

**1.5 REPORT ORGANIZATION**

A vulnerability assessment is the process of defining, identifying, classifying and prioritizing vulnerabilities in computer systems, applications and network infrastructures and providing the organization doing the assessment with the necessary knowledge, awareness and risk background to understand the threats to its environment and react appropriately. A penetration test, also known as a pen test, is a simulated cyber-attack against your computer system to check for exploitable vulnerabilities.

1. The Platform will be secured after testing.
2. Managing the risk properly.
3. Detect and arrange security threat.

Protect customer loyalty and company image.

1. **LITERATURE SURVEY**

Stated here a vulnerability that has been discovered but is unpatched represents a security risk to a system. During the lifetime of a software system, new vulnerabilities are discovered over time. There are two opposing actors, the patch developers and the potential exploiters. An exploit can happen immediately after a disclosure, perhaps even before the disclosure if the discovery is made by a black-hat finder. Here, a framework for software risk evaluation with respect to the vulnerability lifecycle is proposed. Risk can be evaluated using the likelihood of a security breach and the impact of that adverse event on the system. The proposed approach models the vulnerability lifecycle as a stochastic process. Some of the CVSS metrics can be used to evaluate the impact of the breach. The model uses the information about the transition rates with the related distributions and can lead to simplified as well as detailed modeling methods. It allows a comparison of software systems in terms of the risk and potential approaches for optimization of remediation.

* 1. **Background**

penetration test is using a metaphor to give a better understanding of the problem. Imagine you have just bought an expensive brand new car and you want to prevent a theft. So you ask a friend, a policeman, to keep your car safe. He doesn’t know your model, so he has to examine the car and do it as if he were a thief. Before the test begins you must decide what you mean by “steal”, that is breaking into the car and drive it away,

even though is later returned. Now the goal becomes more specific and the tester asks himself: can someone who hasn’t a key drive your car away? The goal of the test isn’t to absolutely prevent the theft, being almost impossible, but at least make it harder, limiting the thief’s capacities, for example requiring more than three hours for stealing the car. Proved this, the tester must know as much information as the attackers know so that the test does not fail, and the best way to assure it is that he knows everything about the system. Here the penetration test starts, when the tester (the policeman) first has to gather information about the technical specification (for example the owner’s manual or mechanic’s guide) of the car and get information provided by you.

* 1. **Related work**

In this section we will see a real-life example of penetration testing that involved the civilian government agency FBI. The example is taken from an article in Computerworld that speaks about the penetration tester Chris Goggans that has been working as a penetration tester since 1991. One of his latest exploit was against the FBI. It only took him six hours to break into a crime database without permission. This is how he acted: he discovered a series of unpatched vulnerabilities in the civilian government agency's Web server, used a hole in the Web Server to pull down usernames and passwords that were reused on a host of enterprise systems,

therefore he got Windows domain administrator privileges gaining full access to almost all Windowsbased system in the enterprise, including workstations used by police officers. Finally, remotely controlling them he found programs on their desktops that

automatically connected the workstations to the FBI’s crime database. This vulnerability could have been eliminated through a clear separation of domains such as between the police network and the enterprise network.

1. **SYSTEM REQUIREMENT SPECIFICATION**

The system requirements of this project are as follows :

From user point of view is:-

1. Internet Connection.

2. Basic knowledge of hacking.

3. Expert in peneteration testing.

* 1. **HARDWARE REQUIREMENT**

1. **Minimum requirement**: Internal Memory (RAM)- 2.00 GB, Hard Disk Capacity (CPU)- 60.00GB, Processor- Intel Pentium 1.60GHZ, Monitor-17” Colored 32bit.

**3.2 SOFTWARE REQUIREMENT**

1. **Kali Linux or Windows system.**

1. **BurpSuite (PT)L**: **Burp** or **Burp Suite** is a graphical tool for testing Web application security. The tool is written in Java and developed by PortSwigger Web Security.

The company behind **Burp Suite** has also developed a mobile application containing similar tools compatible with iOS 8 and above.

1. **Metasploit(PT):** **Metasploit** is an exploit development framework that facilitates penetration testing of IT systems. This tool initially started off as a game and was taken over by Rapid 7 for maintenance and further development.
2. **Sublist3r** : **Sublist3r** is a python tool designed to enumerate subdomains of websites using OSINT. It helps penetration testers and bug hunters collect and gather subdomains for the domain they are targeting.
3. **Nmap** : **Nmap** (Network Mapper) is a free and open-source network scanner created by Gordon Lyon (also known by his pseudonym Fyodor Vaskovich). **Nmap** is used to discover hosts and services on a computer network by sending packets and analyzing the responses.
4. **Dirsearch** is a Python-based command-line website directory scanner designed to brute force site structure including directories and files in websites.
5. **Sqlmap** is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. Support to enumerate users, password hashes, privileges, roles, databases, tables and columns.
6. **SYSTEM ANALYSIS & DESIGN**

## **The life cycles of Vulnerability Assessment and Penetration Testing:**

1. Scoping
2. Information gathering
3. Vulnerability scanning
4. False positive analysis
5. Vulnerability exploitation (Penetration Testing)
6. Report generation

The following figure illustrates the different sequential stages recommended to follow for a Vulnerability Assessment or Penetration Testing:

## 

## **Stage 1 – Scoping**

Scoping is the primary step of any security assessment activity. In order to execute a VA or PenTest, the first step is to identify the scope of the assessment in terms of infrastructure against which the assessment is to be conducted, for example, servers, network devices, security devices, databases, and applications.

Scoping depends on the business objective of the Vulnerability Assessment. During the scoping, a scanning window should also be agreed upon. Also, the types of attacks that are permitted should be agreed upon. After deciding on the scope of assessment, this phase also includes planning and preparation for the test, which includes deciding on the team, date, and time of the test.

Another major factor that should be taken care of prior to beginning the engagement is signing a formal engagement agreement between the security tester and the party on whose infrastructure these tests will be performed.

Scoping should also include identifying the count of infrastructure elements to be tested.

Apart from the infrastructure scope and other program management modalities, the exact scope, the organization’s approach to the business objective, and the methodology of the assessment should be decided. For deciding on the business objective, the organization should identify the type of attack that it would like to get mimicked.

An example of an objective that a company might seek is: “To find out what an external attacker can achieve by targeting externally exposed infrastructure with only the knowledge of a publicly exposed IP address.” This type of requirement will be met through an external Black box penetration testing of infrastructure and applications, and the approach and the methodology should be in accordance with that.

Based on the accessibility of infrastructure from the Internet or intranet, the testing can be done from an external or internal network. Also, based on the type of details, the infrastructure testing can be Black box or Grey box. And depending on the type of infrastructure, the plugins or features of a vulnerability scanning tool should be enabled, aided by appropriate manual checks.

## **Stage 2 – Information gathering**

Information gathering is the second and most important stage of a VA-PT assessment. This stage includes finding out information about the target system using both technical (WhoIS) and nontechnical passive methods such as the search engine.

This step is critical as it helps in getting a better picture of the target infrastructure and its resources. As the timeline of the assessment is generally time bound, information captured during this phase helps in streamlining the effort of testing in the right direction by using the right tools and approach applicable to target systems.

This step becomes more important for a Black box assessment where very limited information about the target system is shared. Information gathering is followed by a more technical approach to map the target network using utilities such as pings and Telnet and using port scanners such as NMAP. The use of such tools would enable

assessors to find a live host, open services, operating systems, and other information.

The information gathered through network mapping will further validate information gathered through other passive means about the target infrastructure, which is important to configure the vulnerability scanning tool. This ensures that scanning is done more appropriately.

## **Stage 3 – Vulnerability scanning**

This stage involves the actual scanning of the target infrastructure to identify existing vulnerabilities of the system. This is done using vulnerability scanners such as Nessus. Prior to scanning, the tool should be configured optimally as per the target infrastructure information captured during the initial phases.

Care should also be taken that the tool is able to reach the target infrastructure by allowing access through relevant intermediate systems such as firewalls.

Such scanners perform protocol TCP, UDP, and ICMP scans to find open ports and services running on the target machine and match them to well- known published vulnerabilities updated regularly in the tool’s signature database if they exist in the target infrastructure. The output of this phase gives an overall view of what kind of vulnerabilities exist in the target infrastructure that if exploited can lead to system compromise.

## **Stage 4 – False positive analysis**

As an output of the scanning phase, one would obtain a list of vulnerabilities of the target infrastructure. One of the key activities to be performed with the output would be false positive analysis, that is, removing any vulnerability that is falsely reported by the tool and does not exist in reality. All scanning tools are prone to report false positives, and this analysis can be done using methods such as correlating vulnerabilities with each other and previously gathered information and scan reports, along with actually checking whether system access is available.

Vulnerability scanners give their own risk rating to the identified vulnerabilities; these can be revisited considering the actual criticality of the infrastructure element

(server or network device) to the network and impact of the vulnerability.

## **Stage 5 – Vulnerability exploitation (Penetration Testing)**

In case system owners require proof of existing vulnerabilities or exploits to understand the extent to which an attacker can compromise a vulnerable system, testers will be required to demonstrate exploits in a controlled environment without actually making the infrastructure unavailable, unless that’s a requirement.

Penetration Testing is the next step to Vulnerability Assessment aiming to penetrate the target system based on

exploits available for the identified vulnerabilities. For exploitation,

our own knowledge or publicly available exploits of well-known vulnerabilities can be utilized.

Penetration Testing or Vulnerability Exploitation can be broadly divided into phases such as pre exploitation, exploitation, and post exploitation.

Activities in the pre-exploitation phase are explained in phases 1 to 4 that is, enumerating the infrastructure and identifying the vulnerability.

Once any vulnerability is exploited to gain access to the system, the attacker should aim to further detail the network by sniffing traffic, mapping the internal network, and trying to obtain a higher privilege account to gain the maximum level of access to the system.

This will enable testers to launch further attacks on the network to further increase the scope of compromised systems.

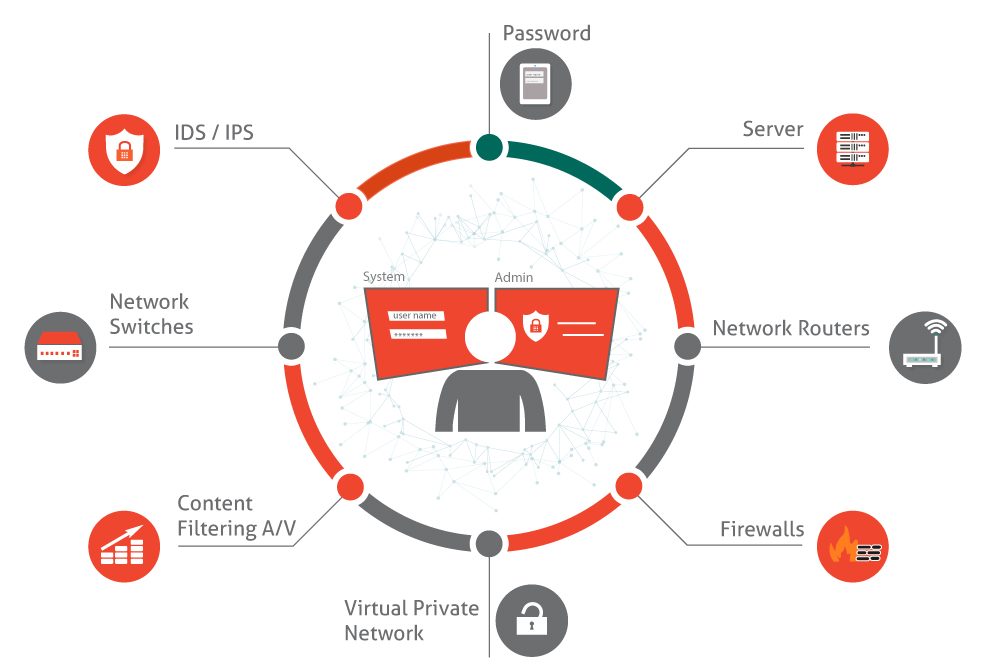
The post exploitation step will also involve clearing of tracks by conducting activities such as clearing logs and disabling antivirus. As a post-exploitation phase tester, you can demonstrate how an attacker can maintain access to the system through backdoors and rootkits.

## **Stage 6 – Report generation**

After completing the assessment as per the scope of work, final reporting needs to be

done covering the following key areas:

* A brief introduction about the assessment
* The scope of assessment
* The management/executive summary
* A synopsis of findings with risk severity
* Details about each finding with their impact and your recommendations to fix the vulnerability.



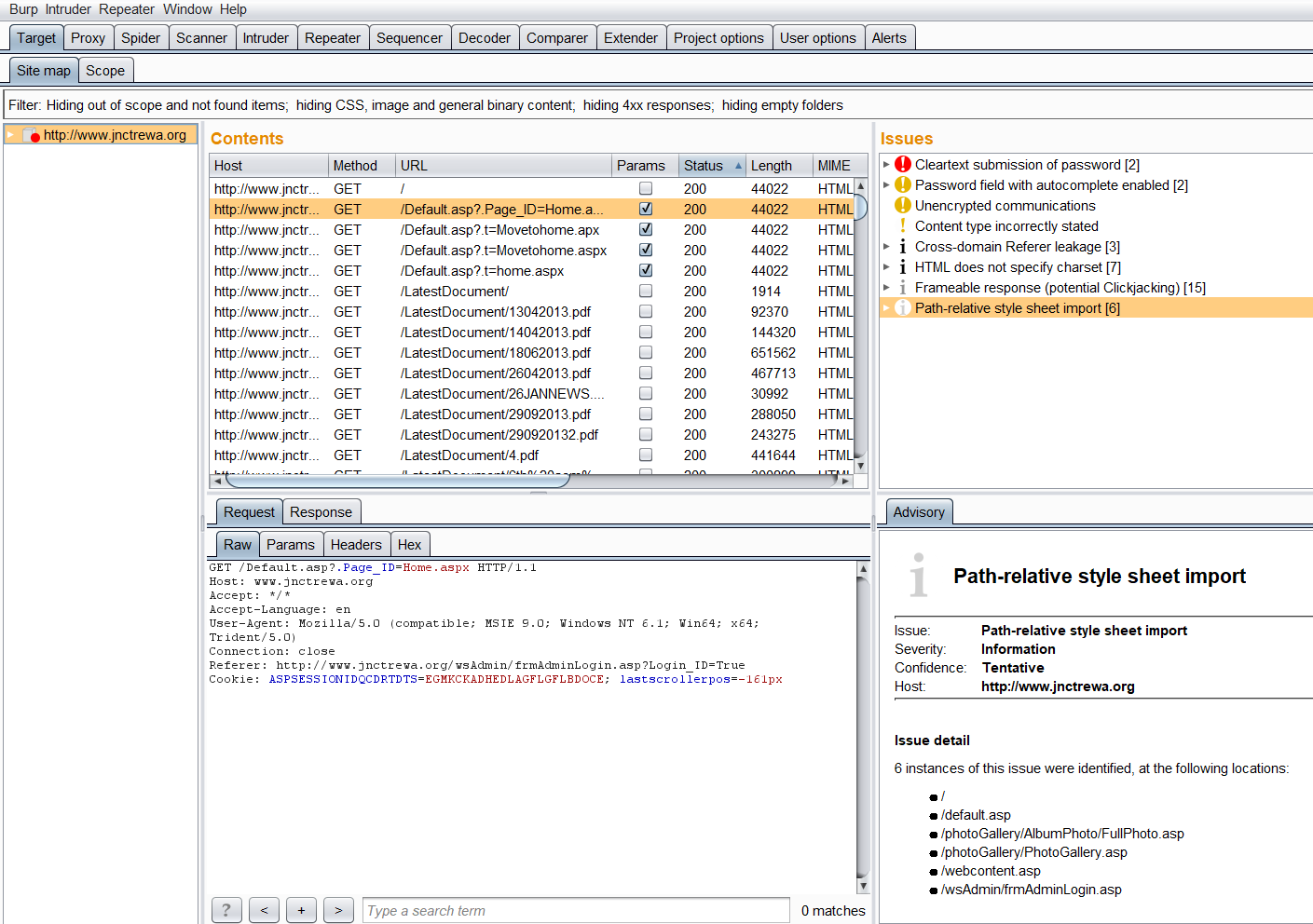
1. **TESTING**

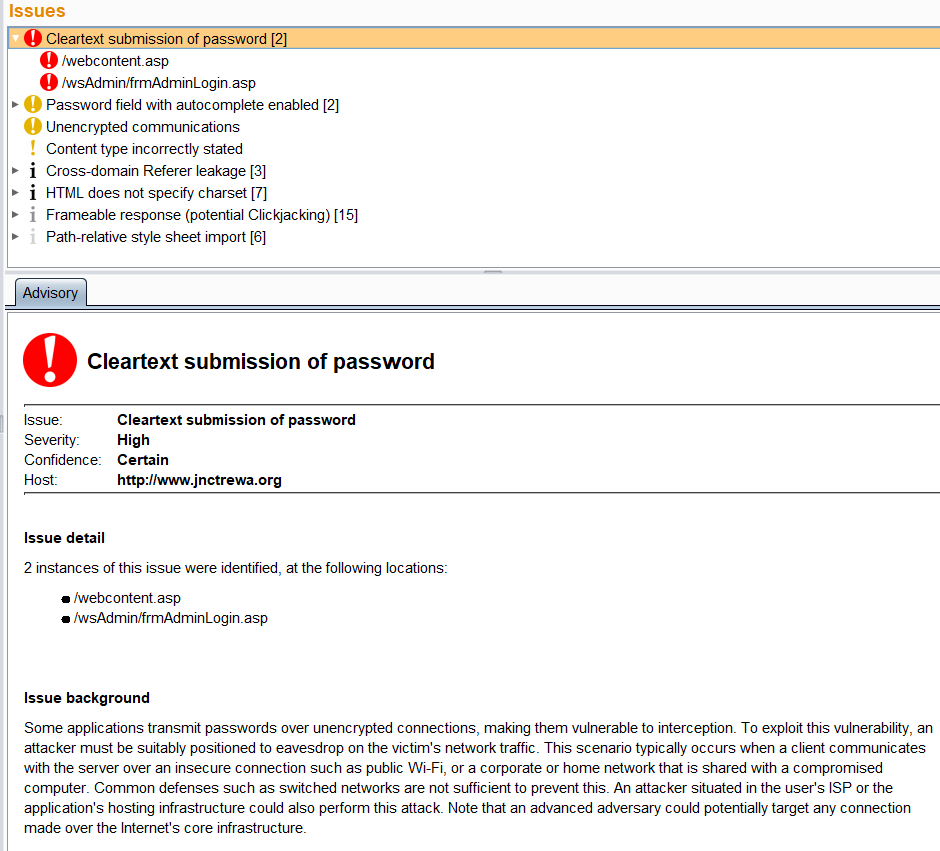
****

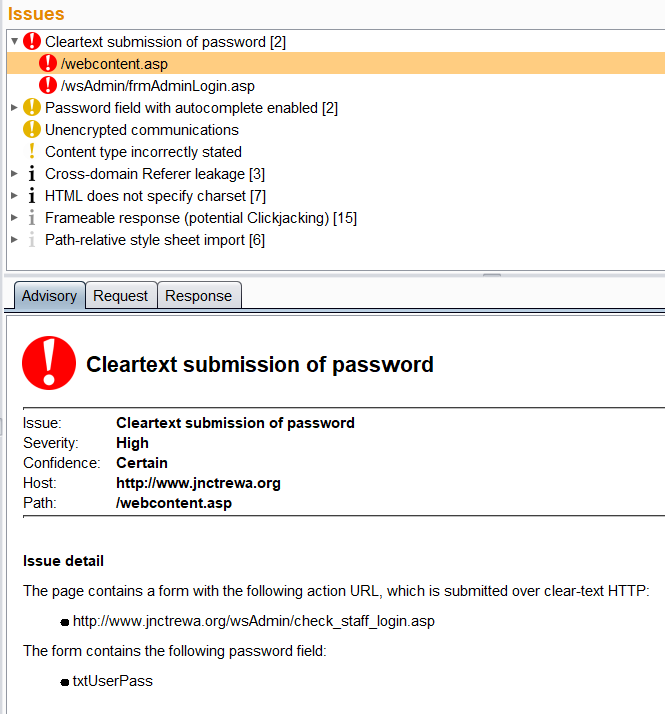
## **5.1 TESTING TOOLS**

**Burpsuite**: **Burp Suite** is a Java based [Web Penetration Testing](https://www.pentestgeek.com/web-penetration-testing) framework. It has become an industry standard suite of tools used by information security professionals. **Burp Suite** helps you identify vulnerabilities and verify attack vectors that are affecting web applications. Because of its popularity and breadth as well as depth of features, we have created this useful page as a collection of **Burp Suite** knowledge and information.

In its simplest form, **Burp Suite** can be classified as an Interception Proxy. While browsing their target application, a penetration tester can configure their internet browser to route traffic through the **Burp Suite** proxy server. **Burp Suite** then acts as a (sort of) [Man In The Middle](https://www.pentestgeek.com/man-in-the-middle) by capturing and analyzing each request to and from the target web application so that they can be analyzed. Penetration testers can pause, manipulate and replay individual HTTP requests in order to analyze potential parameters or injection points. Injection points can be specified for manual as well as automated fuzzing attacks to discover potentially unintended application behaviors, crashes and error messages.







## C:\Users\lenovo\Desktop\New folder (2)\b3.PNG

## C:\Users\lenovo\Desktop\New folder (2)\c1.PNG

## C:\Users\lenovo\Desktop\New folder (2)\p1.PNGC:\Users\lenovo\Desktop\New folder (2)\path.PNGC:\Users\lenovo\Desktop\New folder (2)\h1.PNGC:\Users\lenovo\Desktop\New folder (2)\f1.PNG

## **Sublist3r:** is a python tool designed to enumerate subdomains of websites using OSINT. It helps penetration testers and bug hunters collect and gather subdomains for the domain they are targeting. Subdomain enumeration is the process of finding valid (resolvable) subdomains for one or more domain(s). Unless the DNS server exposes a full DNS zone ([via AFXR](https://www.us-cert.gov/ncas/alerts/TA15-103A)), it is really hard to obtain a list of existing subdomains. The common practice is to use a dictionary of common names, trying to resolve them. While this method is effective in some cases, it doesn't include subdomains that have strange names. Another approach is to crawl the second-level domain in order to find links to subdomains (faster approach is to use a search engine directly).

## C:\Users\lenovo\Desktop\New folder (2)\sublist3r.PNG

## **DNSdumpster**.com is a FREE domain research tool that can discover hosts related to a domain. Finding visible hosts from the attackers perspective is an important part of the security assessment process. Footprinting and Reconnaissance

## Quickly mapping an organisations attack surface is an essential skill for network attackers (penetration testers, bug bounty hunters or Mr Robot) as well as those who are defending the network (network security folks, system administrators, blue teams etc).

## A detailed footprint of an organisations Internet facing systems is a tactical resource that can be used by both attackers and defenders. By developing an understanding of the attack surface skilled security analysts are able to quickly identify weak areas in the attack surface.

## Discovered assets such as old servers, custom web applications and forgotten services are often the first crumbs in a trail that leads to a compromise.

## C:\Users\lenovo\Desktop\New folder (2)\dns2.PNGC:\Users\lenovo\Desktop\New folder (2)\dns1.PNG

## **pentester-tools:** Penetration testing, also called pen testing or ethical hacking, is the practice of testing a computer system, network or web application to find security vulnerabilities that an attacker could exploit. Penetration testing can be automated with software applications or performed manually. Either way, the process involves gathering information about the target before the test, identifying possible entry points, attempting to break in -- either virtually or for real -- and reporting back the findings.

## The main objective of penetration testing is to identify security weaknesses. Penetration testing can also be used to test an organization's security policy, its adherence to compliance requirements, its employees' security awareness and the organization's ability to identify and respond to security incidents.

## C:\Users\lenovo\Desktop\New folder (2)\pentestTools.PNG

## **Dirsearch** is a Python-based command-line website directory scanner designed to brute force site structure including directories and files in websites.

## C:\Users\lenovo\Desktop\New folder (2)\dirsearch.PNG

## **Visualsitemapper:** A sitemap generator is a specific type of software that can automatically create a list of pages that are contained within a website or online application.

## C:\Users\lenovo\Desktop\New folder (2)\visualsite.PNG

## **5.2 Testing Report**

## VAPT Report on [http://www.jnctrewa.org](http://www.jnctrewa.org/):

We have discovered multiple security flaws in your website, please follow the below mentioned steps to reproduce the same.

Vulnerability Name:

Reflected XSS (cross site scripting)

Default credentials used in student gallery control panel Misconfigured SSL

Missing SPF records Information leakage

## Reflected XSS (cross site scripting):

Summary:

Reflected Cross Site Scripting (XSS) is a vulnerability in which attacker injects malicious codes into URL/Link and share this with victim. When any user will visit shared link, the code will execute and will result in Reflected Cross Site Scripting (XSS).

Severity:

High

Vulnerable Domain: <http://www.jnctrewa.org/>

Vulnerable Link:

[http://jnctrewa.org/webcontent.asp?.jPage=Sitesearch.aspx&search=XSS&Sub](http://jnctrewa.org/webcontent.asp?.jPage=Sitesearch.aspx&amp;search=XSS&amp;Sub) mit=Search

Steps:

1: Visit this mentioned website [http://www.jnctrewa.org/.](http://www.jnctrewa.org/) 2: Please enter the following code into search bar

<script>alert(“XSS”)</script>.

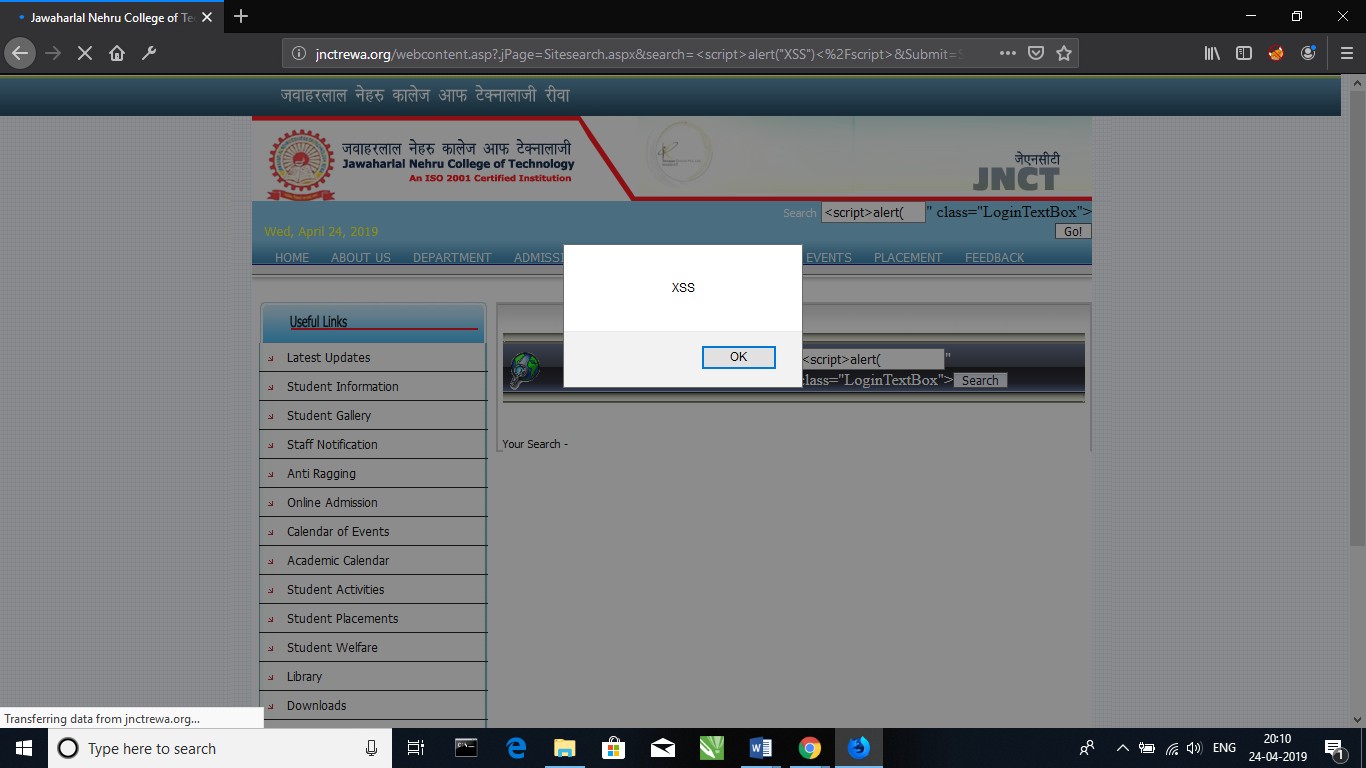
3: This will execute a XSS popup on your browser window.

Impact:

Attacker may use this vulnerability to craft and send legimate website link which contains crafted malicious payload, As soon as the victim will click on that link it'll execute on browser. Which may lead to some serious issues i.e Session Hijacking, Virtual Defacing.

Fix:

Having a strong validation of user input will fix this issue. Special character's like ></ should be blacklisted on priority.

Proof:

## Default credentials used in student gallery control panel:

Summary:

Default credentials used in student control panel is a vulnerability in which attacker login into your control panel by using default credentials and he can use the upload feature to upload any image or use delete feature to delete any image.

Severity:

Medium

Vulnerable Domain: <http://www.jnctrewa.org/>

Vulnerable link: <http://www.jnctrewa.org/photoGallery/AlbumPhoto.asp>

Steps:

1: Visit the mentioned website <http://www.jnctrewa.org/>.

2: Then go to this link <http://www.jnctrewa.org/login/>.

3: Then go to the Student gallery <http://www.jnctrewa.org/photoGallery/AlbumPhoto.asp>.

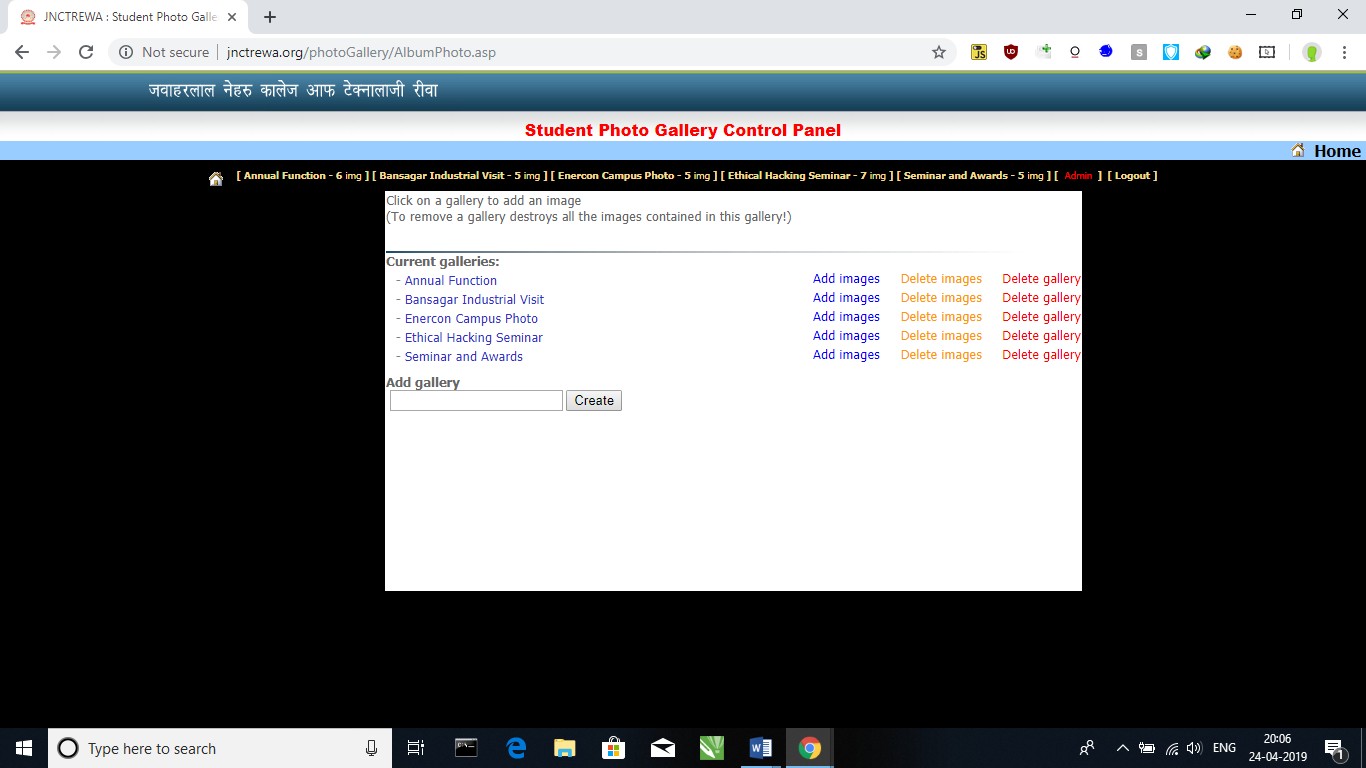
4: Then go to Admin menu and type Username=admin, password=admin

Impact:

Attacker may use this vulnerability to upload the web shell on server of the website by which he can get access to the server of the website.

Fix:

Change the default credential of student photo gallery control panel .

Proof:

## 

## Misconfigured SSL:

Summary:

Misconfigured SSL is a vulnerability in which attacker can perform MITM (man in the middle attack) to any victim by which attacker can sniff the login detail of the victim.

Severity:

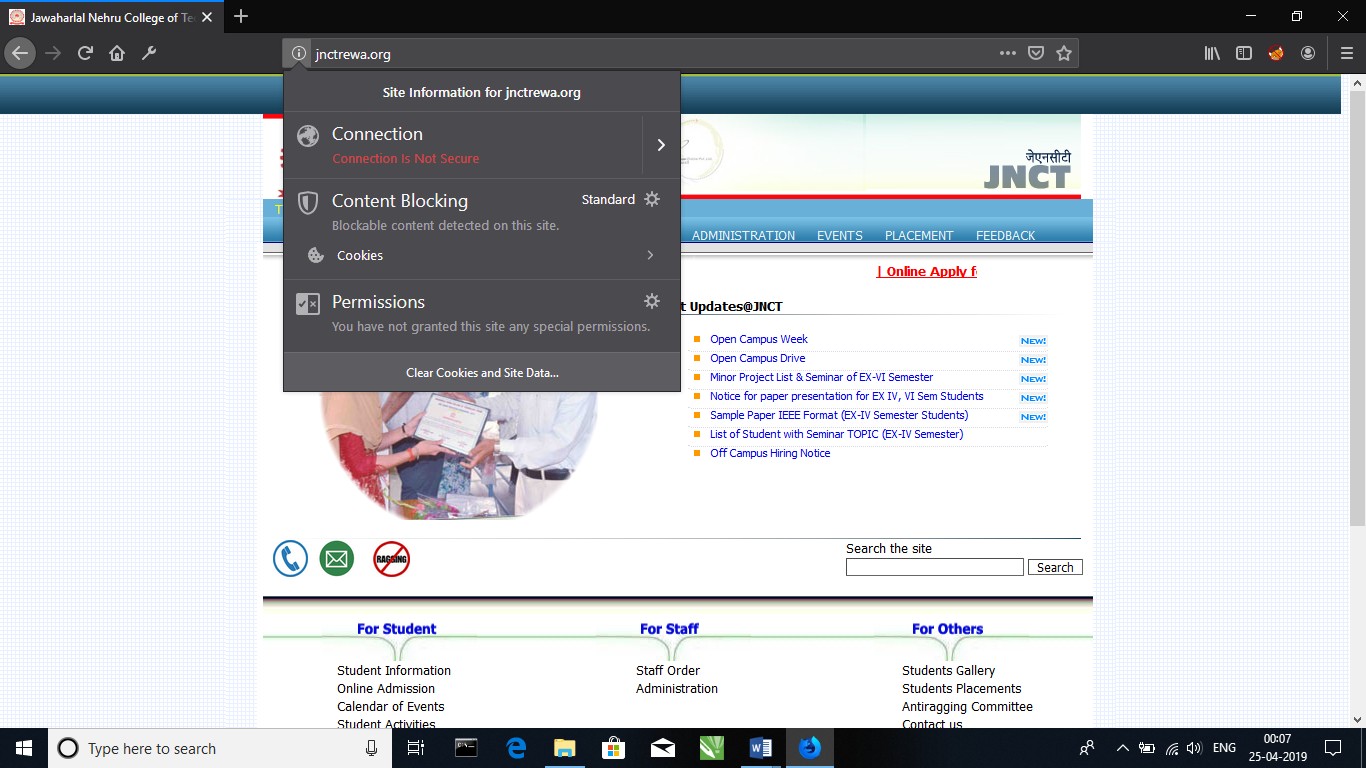
Medium

Vulnerable Domain: <http://www.jnctrewa.org/>

Solution:

Generate a new certificate for the server, expired certificates pose a security threat as they prevent the user accessing your site from being able to properly evaluate the safety of your SSL certificates.

Proof:



## Missing SPF Records:

Summary:

An SPF record is a type of Domain Name Service (DNS) record that identifies which mail servers are permitted to send email on behalf of your domain. The purpose of an SPF record is to prevent spammers from sending messages with forged from addresses at your domain.

Severity:

Medium

Vulnerable Domain: <http://www.jnctrewa.org/>

Steps to Check SPF Records on a website:-

Go to <https://www.kitterman.com/spf/validate.html> Enter Target Website <http://www.jnctrewa.org/> Hit Check SPF

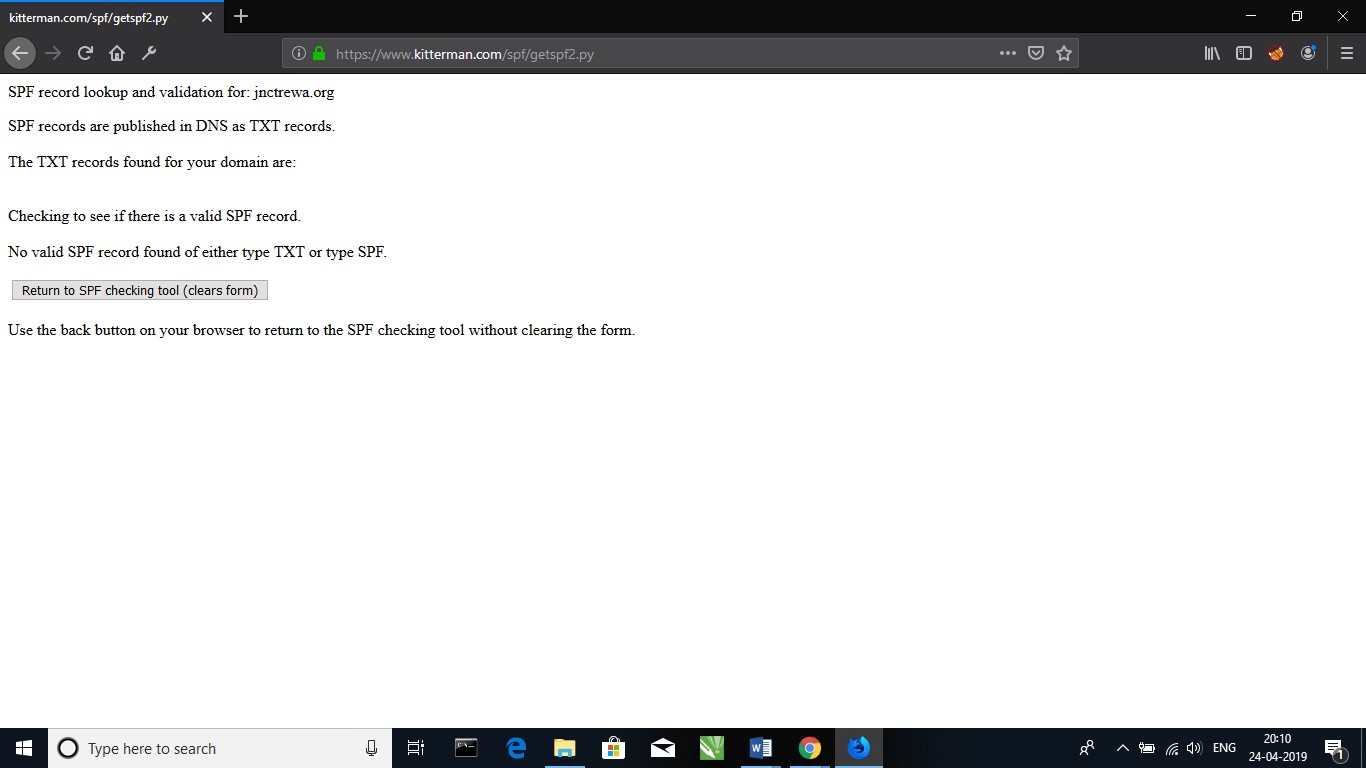
Impact:

Once There are no SPF Records. An Attacker Can Spoof Email Via any Fake Mailer Like Emkei.cz.An Attacker Can Send Email with Social Engineering Attack He Can Takeover User Account Let’s say Victim Knows the Phishing Attacks but When He Sees the Email from the Authorized Domain. He Gets tricked Easily.

Fix:

Set the Host field to the name of your domain; Fill the TXT Value field with your SPF record (i.e. “v=spf1 a mx include: [www.jnctrewa.org](http://www.jnctrewa.org/) ~all””); Specify the Time To Live (TTL), enter 3600 or leave the default; Click “Save” or “Add Record” to publish the SPF TXT record into your DNS.

Proof:



Information leakage:

Summary:

Information Leakage is different than the failure to protect sensitive information at rest and in-transit. This is a legitimate concern, and involves the exposure of any sensitive data stored and processed by the application. Information Leakage involves the exposure of information that would facilitate attacks on the application or other infrastructure, such as insight into the application design, deployment, or organizational details.

Severity:

Low

Vulnerable Domain: <http://www.jnctrewa.org/>

Steps to check leaked information:

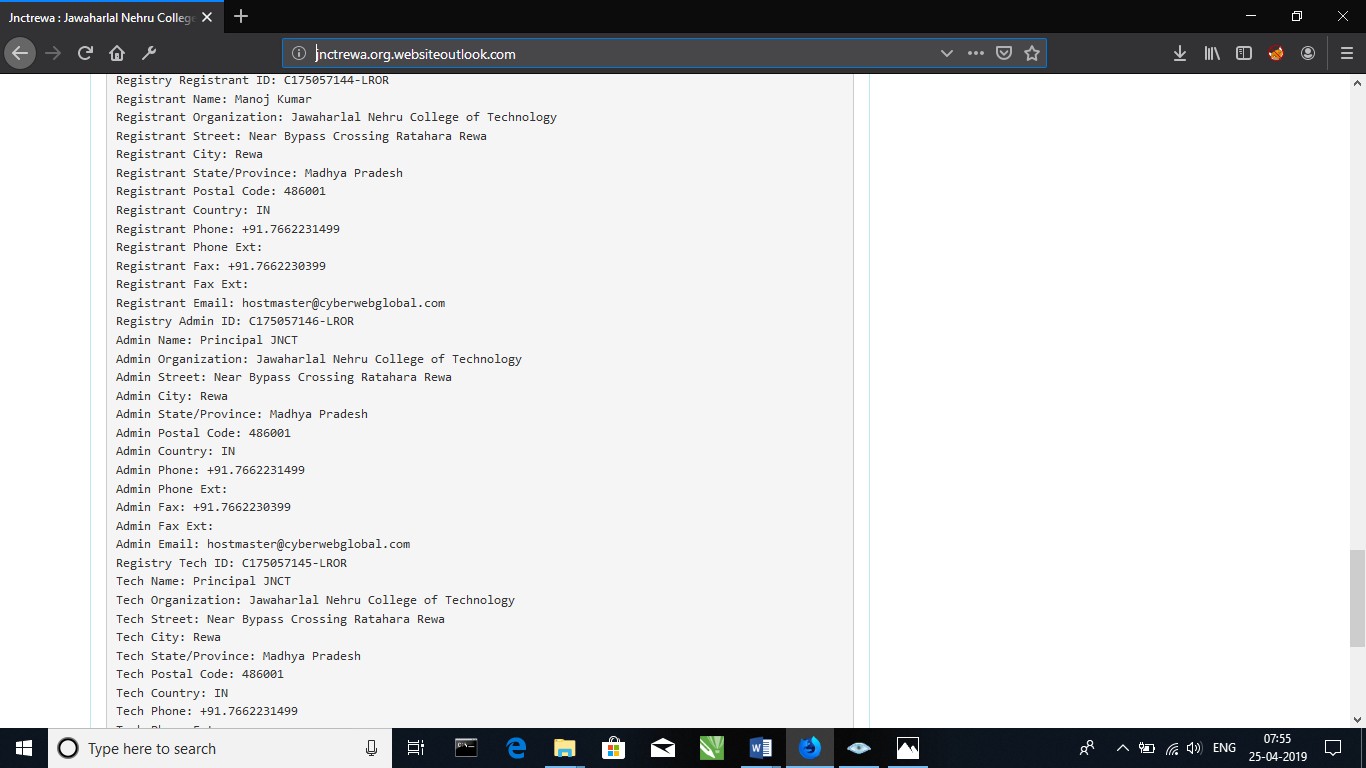
Got to <http://jnctrewa.org.websiteoutlook.com/>

Open Nmap tool type command: nmap -T4 -A -v [www.jnctrewa.org](http://www.jnctrewa.org/)

Impact:

Although such incidents pose the risk of identity theft or other serious consequences, in most cases there is no lasting damage; either the breach in security is remedied before the information is accessed by unscrupulous people, or the thief is only interested in the hardware stolen, not the data it contains.

Proof:



Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-25 00:50 India Standard Time

NSE: Loaded 148 scripts for scanning. NSE: Script Pre-scanning.

Initiating NSE at 00:50

Completed NSE at 00:50, 0.02s elapsed Initiating NSE at 00:50

Completed NSE at 00:50, 0.00s elapsed Initiating Ping Scan at 00:50

Scanning [www.jnctrewa.org](http://www.jnctrewa.org/) (50.62.169.113) [4 ports] Completed Ping Scan at 00:50, 1.69s elapsed (1 total hosts) Initiating Parallel DNS resolution of 1 host. at 00:50

Completed Parallel DNS resolution of 1 host. at 00:50, 0.09s elapsed Initiating SYN Stealth Scan at 00:50

Scanning [www.jnctrewa.org](http://www.jnctrewa.org/) (50.62.169.113) [1000 ports] Discovered open port 1723/tcp on 50.62.169.113 Discovered open port 554/tcp on 50.62.169.113 Discovered open port 21/tcp on 50.62.169.113 Discovered open port 443/tcp on 50.62.169.113 Discovered open port 80/tcp on 50.62.169.113 Discovered open port 49154/tcp on 50.62.169.113

Increasing send delay for 50.62.169.113 from 0 to 5 due to 24 out of 59 dropped probes since last increase.

Discovered open port 2107/tcp on 50.62.169.113 Discovered open port 2105/tcp on 50.62.169.113

Increasing send delay for 50.62.169.113 from 5 to 10 due to 143 out of 356 dropped probes since last increase.

SYN Stealth Scan Timing: About 44.98% done; ETC: 00:51 (0:00:38 remaining)

Discovered open port 8443/tcp on 50.62.169.113

Warning: 50.62.169.113 giving up on port because retransmission cap hit (6). Discovered open port 49155/tcp on 50.62.169.113

Discovered open port 49156/tcp on 50.62.169.113 Discovered open port 2103/tcp on 50.62.169.113

Completed SYN Stealth Scan at 00:52, 124.53s elapsed (1000 total ports) Initiating Service scan at 00:52

Scanning 12 services on [www.jnctrewa.org](http://www.jnctrewa.org/) (50.62.169.113)

Service scan Timing: About 41.67% done; ETC: 00:55 (0:01:23 remaining) Completed Service scan at 00:55, 153.34s elapsed (12 services on 1 host) Initiating OS detection (try #1) against [www.jnctrewa.org](http://www.jnctrewa.org/) (50.62.169.113) Retrying OS detection (try #2) against [www.jnctrewa.org](http://www.jnctrewa.org/) (50.62.169.113) Initiating Traceroute at 00:55

Completed Traceroute at 00:55, 3.08s elapsed Initiating Parallel DNS resolution of 5 hosts. at 00:55

Completed Parallel DNS resolution of 5 hosts. at 00:55, 0.11s elapsed NSE: Script scanning 50.62.169.113.

Initiating NSE at 00:55

Completed NSE at 00:55, 19.52s elapsed Initiating NSE at 00:55

Completed NSE at 00:55, 1.16s elapsed

Nmap scan report for [www.jnctrewa.org](http://www.jnctrewa.org/) (50.62.169.113) Host is up (0.096s latency).

rDNS record for 50.62.169.113: p3nwvpweb154.shr.prod.phx3.secureserver.net Not shown: 911 closed ports, 77 filtered ports

PORT STATE SERVICE VERSION

21/tcp open ftp Microsoft ftpd

| ftp-syst:

|\_ SYST: Windows\_NT

80/tcp open http Microsoft IIS httpd 8.5

| http-cookie-flags:

| /:

| ASPSESSIONIDCQRDDCQD:

|\_ httponly flag not set

|\_http-favicon: Unknown favicon MD5: B8F2338E1C83D5881ED2CD87508B9ECA

| http-methods:

| Supported Methods: OPTIONS TRACE GET HEAD POST

|\_ Potentially risky methods: TRACE

|\_http-server-header: Microsoft-IIS/8.5

|\_http-title: Jawaharlal Nehru College of Technology, Rewa 443/tcp open ssl/http Microsoft IIS httpd 8.5

| http-cookie-flags:

| /:

| ASPSESSIONIDAWSDCDTA:

|\_ httponly flag not set

| http-methods:

|\_ Supported Methods: HEAD POST OPTIONS

|\_http-title: Page Not Found

| ssl-cert: Subject: commonName=\*.shr.prod.phx3.secureserver.net

| Subject Alternative Name: DNS:\*.shr.prod.phx3.secureserver.net, DNS:shr.prod.phx3.secureserver.net

| Issuer: commonName=Starfield Secure Certificate Authority - G2/organizationName=Starfield Technologies, Inc./stateOrProvinceName=Arizona/countryName=US

| Public Key type: rsa

| Public Key bits: 2048

| Signature Algorithm: sha256WithRSAEncryption

| Not valid before: 2019-03-12T19:28:40

| Not valid after: 2021-03-12T19:28:40

| MD5: 5185 1a3d 90c3 e244 bd00 5218 599e 2372

|\_SHA-1: 3b53 f59d e22c 393d 3885 e89b d950 6882 ca33 325a

554/tcp open rtsp?

1723/tcp open pptp?

|\_pptp-version: ERROR: Script execution failed (use -d to debug) 2103/tcp open msrpc Microsoft Windows RPC

2105/tcp open msrpc Microsoft Windows RPC 2107/tcp open msrpc Microsoft Windows RPC 8443/tcp open ssl/http Microsoft IIS httpd 8.5

|\_http-favicon: Unknown favicon MD5: 1DB747255C64A30F9236E9D929E986CA

| http-methods:

|\_ Supported Methods: HEAD POST

| http-robots.txt: 1 disallowed entry

|\_/

|\_http-server-header: Microsoft-IIS/8.5

|\_http-title: Plesk Onyx 17.8.11

| ssl-cert: Subject: commonName=\*.shr.prod.phx3.secureserver.net

| Subject Alternative Name: DNS:\*.shr.prod.phx3.secureserver.net, DNS:shr.prod.phx3.secureserver.net

| Issuer: commonName=Starfield Secure Certificate Authority - G2/organizationName=Starfield Technologies, Inc./stateOrProvinceName=Arizona/countryName=US

| Public Key type: rsa

| Public Key bits: 2048

| Signature Algorithm: sha256WithRSAEncryption

| Not valid before: 2019-03-12T19:28:40

| Not valid after: 2021-03-12T19:28:40

| MD5: 5185 1a3d 90c3 e244 bd00 5218 599e 2372

|\_SHA-1: 3b53 f59d e22c 393d 3885 e89b d950 6882 ca33 325a

49154/tcp open msrpc Microsoft Windows RPC 49155/tcp open msrpc Microsoft Windows RPC 49156/tcp open msrpc Microsoft Windows RPC

OS fingerprint not ideal because: Didn't receive UDP response. Please try again with -sSU

No OS matches for host

Uptime guess: 15.692 days (since Tue Apr 09 08:19:26 2019) Network Distance: 6 hops

Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE (using port 80/tcp) HOP RTT ADDRESS

1 0.00 ms 192.168.43.1

3 63.00 ms 10.72.57.226

4 63.00 ms 172.25.110.53

5 63.00 ms 172.25.66.146

6 63.00 ms p3nwvpweb154.shr.prod.phx3.secureserver.net (50.62.169.113)

NSE: Script Post-scanning. Initiating NSE at 00:55

Completed NSE at 00:55, 0.00s elapsed Initiating NSE at 00:55

Completed NSE at 00:55, 0.00s elapsed

Read data files from: C:\Program Files (x86)\Nmap

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 324.27 seconds

Raw packets sent: 6584 (293.812KB) | Rcvd: 2425 (98.236KB)

## **Overall vulnerability Risk Classification:**

Throughout the document, each vulnerability or risk identified has been labeled as a Finding and

categorized as a High-Risk, Medium-Risk, or Low-

Risk. In addition, each supplemental testing note

is labeled as an Issue. These terms are defined below:

**Medium Risk:** These findings identify condition that do not immediately or directly result in the compromise or unauthorized access of a network, system, application or information but do provide a capability or information that could, in combination with other capabilities or information, result in the compromise or unauthorized access of a network, system, application or information. Examples – Unprotected system .files and services that could result in DoS on non-critical services that could be further exploited.

**High Risk**: These finding identify conditions that could that compromise or unauthorized access of a network, system, application or information. Example – No Password,

No encryption, Denial of Service, buffer overflows.

**Low Risk:** These finding identify condition that do not immediately or directly result in the compromise of a network.system,application,or information ,but do provide information that could be used in combination with other information to gain insight into how to compromise or gain unauthorized access to a network ,system, application or information .Low risk findings may also demonstrate an incomplete approach to or application of security measure within the environment .Examples of Low risk include cookies not marked secure ;concurrent session and revealing system banners

1. **FUNCTIONAL & NON FUNCTIONAL REQUIREMENTS**

**6.1 Various functional requirements that would be required in this application are as follows:**

1. Scoping

2. Information gathering

3. Vulnerability scanning

4. False positive analysis

5. Vulnerability exploitation (Penetration Testing)

6. Report generation

**6.2 Various non-functional requirements that would be required in this application are as follows:**

1. Performance -The application lies between light weight and heavy weight. Service provider are trustworthy.

2.Reliability- A customer can rely on the services provide by the providers as they are verified first.

3.Availability- The user should have Internet connection technology in the device.

4.Security- Security systems need database storage just like many other applications. All the feedbacks provided will be genuine as mail service is linked to it.

5.Maintainability- The main maintainability required is for the database .

6.Portability- The product is highly portable as it can be accessed from any device with any operating system.

**1.1 PERFORMANCE REQUIREMENT**

Performance requirements define how well the system performs certain functions under specific conditions. Examples are speed of response, throughput, execution time and storage capacity. The service levels comprising performance requirements are often based on supporting end-user tasks.The base prerequisites for performance

testing include understanding the application under test, identifying performance requirements such as response time, normal and peak load, common traffic patterns, and expected or required uptime.In order to assess the performance of a system the following must be clearly specified:  
• Response Time  
• Workload  
• Scalability  
• Platform

### 2.1 RELIABILITY

Application reliability is the probability of a piece of software operating without failure while in a specified environment over a set duration of time. In a perfect world, a reliable piece of software is completely defect free, does not create downtime, and performs correctly in every scenario. In reality, this is difficult to accomplish as an infrastructure becomes increasingly complex and developers must deal with strict time schedules for project completion.

Reliability is a factor of quality, but a distinct measurement for determining the probability of failure as programs are developed or enhanced. A robust application is one that can perform even when unexpected or unanticipated events occur. The development of secure, dependable, and robust software is the end goal for most organizations. Application reliability metrics aid in meeting this objective by providing insightful information about what areas of an application are causing or could cause potential problems.

**3.1 SCALABILITY**

Scalability is the ability of your app to handle a growing number of customers, clients and/or users. It also pertains to the ability of your current or future developer to maintain the app. Consider the possibility that your app isn’t currently scalable. If it’s not, be prepared to scramble to add necessary additional features without the flexible coding that keeps your app running smoothly.Your app needs to be scalable from the start, way before it’s launched to the public. And the only way to truly know if the app can handle rapid scaling is by extensive testing. The reason scalability testing is necessary is that you

need to know if your app can handle the enormous workload ahead while avoiding costly glitches and constant updates. The only way to truly know it’s ready is to consistently and comprehensively test it.

Scalability is crucial for your app’s success. When your app is scalable, it can accommodate growth, provide a good user experience for new users, and give you a better ROI. If you want to serve thousands or even millions with your app, you need to plan for it from the beginning.The ultimate purpose of scalability testing is to push the current app beyond its breaking point. Your team can then pinpoint that crucial moment, reverse engineer the problem, and make the app scalable and functional.

**4.1 SECURITY**

Application security is the use of software, hardware, and procedural methods to protect applications from external threats. Application security encompasses measures taken to improve the security of an application often by finding, fixing and preventing security vulnerabilities.

Application security can be enhanced by rigorously defining enterprise assets, identifying what each application does (or will do) with respect to these assets, creating a security profile for each application, identifying and prioritizing potential threats and documenting adverse events and the actions taken in each case. This process is known as [threat modeling](https://searchsecurity.techtarget.com/definition/threat-modeling). In this context, a threat is any potential or actual adverse event that can compromise the assets of an enterprise, including both malicious events, such as a denial-of-service ([DoS](https://searchsecurity.techtarget.com/definition/DOS)) attack, and unplanned events, such as the failure of a storage device.

Once an afterthought in software design, security is becoming an increasingly important concern during development as applications become more frequently accessible over networks and are, as a result, vulnerable to a wide variety of threats. Security measures built into applications and a sound application security routine minimize the likelihood that unauthorized code will be able to manipulate applications to access, steal, modify, or delete sensitive data.

**5.1 TESTABILITY**

Software testability is the degree to which a software artifact (i.e. a software system, software module, requirements- or design document) supports testing in a given test context. If the testability of the software artifact is high, then finding faults in the system (if it has any) by means of testing is easier.A testable product ensures complete execution of the test scripts. Assuming that good test coverage is applied, most of the defects will be uncovered and fixed before the product is released. This insures customers will report a minimum number of defects. A lot of money is spent on supporting and maintaining a product after its development. Testable products are easy and less costly to maintain. The chances of achieving customer satisfaction with such products is are much higher. Hence testability is an important attribute to the maintainability of any software product.

Therefore, testability is often thought of as an extrinsic property which results from interdependency of the software to be tested and the test goals, test methods used, and test resources (i.e., the test context). Even though testability cannot be measured directly (such as software size) it should be considered an intrinsic property of a software artifact because it is highly correlated with other key software qualities such as encapsulation, coupling, cohesion, and redundancy. Being able to test software, a piece of code or functionality, depends on what the user can see and control, known as observability and controllability.

### Observability enables a tester or user to see the external and internal of the software. When a user receives the correct expected output, but the internal or the background processes are not quite what was specified in the requirements, defects are often found elsewhere. This is more important in the case of unit and integration testing rather than a simple black box testing.

Controllability is a measure of how easily a tester or a user can create difficult scenarios to test the software under extreme circumstances. For example, behavior of an application cannot be tested very easily when the hard disk is full or table overflow conditions exist.

**7. CONCLUSION**

This is to conclude that the project that we undertook was worked upon with a sincere effort. Most of the requirements have been fulfilled up to the mark and the requirements which have been remaining, can be completed with a short extension. This project would definitely satisfy all the requirements of the college and would be beneficial for the students and the college staff. We find the vulnerabilities in the website of successfully with the help of above tools and it is very interesting to hack into the website and make report of it.

Signature of the Candidate Signature of the Guide Signature of the Supervisor