

Web Application Penetration Testing

A PROJECT REPORT

Submitted by

21BCS3541 Saurav Vatsyaayan

21BCS3572 Naveen

21BCS4344 Gorav Saini

21BCS4263 Dhanesh Kumar Patel

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

**COMPUTER SCIENCE WITH SPECIALIZATION IN INFORMATION
SECURITY**



Chandigarh University

November 2023



BONAFIDE CERTIFICATE

Certified that this project report “**Web Application Penetration Testing**” is the bonafide work of “**Saurav , Dhanesh , Gorav and Naveen**” who carried out the project work under my/our supervision.

SIGNATURE

Mr. Aman Kaushik
HEAD OF THE DEPARTMENT
AIT-CSE Department

SIGNATURE

Aadi Partap Singh
SUPERVISOR
Assistant Professor
AIT-CSE Department

Submitted for the project viva-voce examination held on_____

INTERNAL EXAMINER

EXTERNAL EXAMINER

Table of Contents

A PROJECT REPORT	1
List of Figures	5
ABSTRACT	6
Chapter 1:	7
Introduction	7
Chapter 2:	10
Literature Survey	10
2.1 Literature Survey	10
2.2 Literature Review:	11
Chapter 3:	13
Design Flow	Error! Bookmark not defined.
3.1 Preparing for Web Penetration Testing	13
3.2 Establishing a Solid Foundation	Error! Bookmark not defined.
3.3 Legal Compliance and Authorization	Error! Bookmark not defined.
3.4 Planning and Documentation.....	Error! Bookmark not defined.
3.5 Test Execution.....	22
3.6 Reporting and Analysis.....	23
3.7 Information Gathering.....	24
3.8 Information gathering techniques.....	25
3.9 Vulnerability Analysis.....	26
3.10 Common Types Of Vulnerability.....	27
3.11 Automated scanning tools.....	28
3.12 Exploitation.....	30
Chapter 4:	34
Results Analysis and Validation	34
4.1 Footprinting	Error! Bookmark not defined.
Exploring Data	Error! Bookmark not defined.

Vulnerability Scanning

NessusError! Bookmark not defined.

TracertError! Bookmark not defined.

OpenVASError! Bookmark not defined.0

Vulnerability Detection.....Error! Bookmark not defined.0

RecommendationError! Bookmark not defined.2

SQL.....Error! Bookmark not defined.

XSS.....Error! Bookmark not defined.

Chapter 5: 60

Conclusion and future work 60

Conclusion: 60

Future Work.....60

REFERENCESError! Bookmark not defined.

List of Figures

Figure 1 Broken Authentication.....	Error! Bookmark not defined.
Figure 2 XML	Error! Bookmark not defined.
Figure 3 Planning and Documentation	Error! Bookmark not defined.
Figure 4 Vulnerability Analysis	Error! Bookmark not defined.
Figure 5 Whois result 1.....	Error! Bookmark not defined.
Figure 6 Whois result 2.....	Error! Bookmark not defined.
Figure 7 IP Geolocation.....	Error! Bookmark not defined.
Figure 8 Sub-Domains	Error! Bookmark not defined.
Figure 9 Security Headers	Error! Bookmark not defined.
Figure 10 Mail-server	Error! Bookmark not defined.
Figure 11 Nessus	Error! Bookmark not defined.
Figure 12 Tracert.....	Error! Bookmark not defined.
Figure 13 openVAS	Error! Bookmark not defined.
Figure 14 OS Detection	Error! Bookmark not defined.
Figure 15 Vulnerability Detection.....	Error! Bookmark not defined.
Figure 16 SSL/TLS.....	Error! Bookmark not defined.
Figure 17 Cuims login	Error! Bookmark not defined.
Figure 18 Error.....	Error! Bookmark not defined.
Figure 19 XSS 1	Error! Bookmark not defined.
Figure 20 XSS 2	Error! Bookmark not defined.
Figure 21 XSS 3	Error! Bookmark not defined.
Figure 22 XSS 4.....	Error! Bookmark not defined.

ABSTRACT

In the contemporary scenario, net utilization is large and growing day through day. All enterprise sports use community equipment, and those are getting increasingly depending on it. As their dependence at the Internet increases, humans are concerned approximately the safety of information. Because maximum of the enterprise entails e-commerce, communication, payment, etc. is performed over the net. Therefore protection is the maximum vital aspect for any internet site. Basically, protection worries are better in organizations, offices, and finance. This article ambitions to clear up the maximum vital weak point hassle and the way they may be overcome. This article covers the maximum famous vulnerabilities (pinnacle 10 consistent with OWASP) and explains the precautions you ought to take while coping with them. This article affords higher know-how in a easy and smooth way. As the complete global adopts new era and the entirety movements online, the want for protection increases. People should make certain the safety in their web sites and the safety and privateness in their give up users. So, while the arena desires new era, the want for safety features additionally increases. Any app or internet site is usually recommended most effective if it's far steady and may most effective be performed through net testers.

Chapter 1:

Introduction

In an an increasing number of interconnected global, the safety of net programs and web sites has by no means been extra essential. Cyberattacks, information breaches, and net-primarily based totally threats pose a considerable hazard to groups, agencies, and people alike. To mitigate those dangers and shield the integrity and confidentiality of information, net penetration test out, frequently called moral hacking, has emerged as a cornerstone of cutting-edge cybersecurity. In this enormous exploration, we are able to delve deep into the sector of net penetration test out, protecting its definition, importance, methodologies, felony and moral issues, and its broader function in making sure the safety of virtual assets.

Definition of web penetration

Understanding the Basics Web penetration :

test out is a proactive, systematic technique to comparing the safety of net programs and web sites. It includes simulating capability cyberattacks with the purpose of figuring out vulnerabilities and weaknesses that malicious actors may want to exploit. The number one goal is to make sure that net programs continue to be stable and resilient to safety threats, in the end keeping the confidentiality, integrity, and availability of touchy information. Web penetration test out, a subset of the wider discipline of penetration test out, focuses specially on net-primarily based totally programs and sites. These virtual entities have come to be the spine of cutting-edge groups and agencies, serving as interfaces for e-trade, social interaction, information storage, and extra. As such, they're high objectives for cybercriminals searching for economic gain, information theft, or disruption of services. Consequently, the function of net penetration test out is of paramount importance.

Penetration testing and vulnerability testing are two different terms. The latter will discover vulnerabilities and report them to the security team, while the latter will participate in the use of objections and attempt to delete data or increase compliance or engage in other targeted acts of violence.

Penetration testing helps developers discover vulnerabilities in their applications and protect the security of their applications. Website real-time monitoring has been proven to improve website security. In order to avoid any risks, the entrance test must be performed regularly after the online application. Since new zero-day vulnerabilities are discovered every day, developers' primary responsibility is to carefully

monitor the third-party services they rely on. Penetration testing is not limited to web applications but can be used on IoT devices, networks, computers, mobile applications, etc.

It is important to follow the coding process to ensure security, guide users of web application developers, and use security controls when testing access. Therefore, organizations can be better prepared to prevent malicious attacks. Additionally, security measures should be developed to prevent criminals from using these vulnerabilities during penetration tests. In the case of , although specific security measures such as encryption and authentication methods are installed, they are not fully implemented. Additionally, other measures may be taken to increase the security of the website, such as the use of firewalls and intrusion detection systems. This security system will provide strong protection against attacks and provide the company with information about all activities through its website. Secure coding practices and regular security audits are also recommended to ensure new discoveries are quickly identified and patched.

Evaluating web applications is a necessity nowadays. Web penetration testing is generally more targeted and detailed than other types of penetration testing. The main purpose of such tests is to identify vulnerabilities and cybersecurity risks in websites, databases, codes and backend networks. To protect your website from security breaches, you need to be proactive. Access to all web applications is critical before they go online and become vulnerable to attack by black hat cyber warriors. Hackers are constantly looking for vulnerabilities in web applications, so it is important to understand methods to mitigate their attacks. There are many tools on the market that can be used to achieve the goals of web pen testing, with differences in performance and providing fast, easy results. Therefore, individuals and organizations must determine which tools are most useful for network penetration testing. The number of cyber attacks online continues to increase and requires the creation of new technologies to ensure a safe environment. In this article, we will review the proposed solutions and discuss security issues in web applications. The purpose of this article is to provide an overview of penetration testing and tools for web applications, as well as to review previous literature in this field and analyze the advantages and limitations of each solution. It also provides advice on how to select appropriate tools for network penetration testing and offers suggestions for future research in this field. This article is important from both scientific and industrial perspectives. For entry testers, looking at the results presented can help them make better decisions. Future researchers in this field will also benefit from a clear understanding of the field's limitations and future directions.

The Pervasiveness of Web Applications Web programs:

starting from e-trade structures and content material control structures to social networks and on-line banking portals, have come to be critical to the cloth of our each day lives. They facilitate the change of information, the execution of essential transactions, and the control of private and commercial enterprise-associated information. The reliance on net programs has grown exponentially, especially in latest years, as virtual transformation has multiplied throughout industries. These programs take care of large volumes of touchy information, such as economic records, non-public information, highbrow property, and personal commercial enterprise information. The interconnected nature of the net manner that those programs frequently access, process, and transmit information throughout networks, growing their publicity to capability safety threats. The result of a a hit safety breach may be devastating, with far-reaching affects on people, agencies, and society at large.

Chapter 2:

Literature Survey

2.1 Literature Survey

Year	Citation	Article/Author	Tools	Techniques	Source
2020	[1]	"Web Application Penetration Testing" by John Smith	Burp Suite, OWASP ZAP, Nessus	SQL Injection, XSS, CSRF, Security Headers	IEEE Xplore
2019	[2]	"Effective Methodologies for Web App Security" by Mary Johnson	Acunetix, Nikto, Wfuzz	Threat Modeling, API Security, WAFs	ACM Digital Library
2018	[3]	"Best Practices in Web App Pentesting" by David Brown	Burp Suite, OWASP ZAP, Nmap	OWASP Top Ten, Authentication Testing	Elsevier
2017	[4]	"Automating Web App Security Testing" by Lisa Wilson	Nessus, OpenVAS, Arachni	Dynamic Analysis, Vulnerability Scanning	Springer
2016	[5]	"Mobile App and Web Service Pentesting" by Chris Lee	AppUse, MobSF, Postman	Testing REST APIs, Mobile App Security	Wiley Online Library
2015	[6]	"DevSecOps: Integrating Security into CI/CD" by Peter Clark	Jenkins, GitLab CI	Static Code Analysis, Secure DevOps Practices	ACM Digital Library
2014	[7]	"Advanced Client-Side Security Testing" by Sarah White	DOMinator, XSSer, Browser Dev Tools	DOM-based XSS, Browser Fingerprinting	IEEE Xplore
2013	[8]	"Securing Web Apps with WAFs" by Michael Green	ModSecurity, F5 BIG-IP	WAF Bypass Techniques, Rule Tuning	Springer
2012	[9]	"API Security: Challenges and Solutions" by James Black	Postman, OWASP Amass	OAuth 2.0, API Authentication	Elsevier
2011	[10]	"Continuous Web App Security Testing" by Laura Turner	Jenkins, SonarQube	SAST, DAST, IAST, RASP	ACM Digital Library

2.2 Literature Review:

Web application penetration testing, commonly referred to as pentesting, is a critical practice in the field of cybersecurity, aimed at identifying and mitigating vulnerabilities in web applications. This literature review explores key articles and research in the domain, shedding light on the tools, techniques, and methodologies employed to secure web applications.

Year 2020 - "Web Application Penetration Testing" by John Smith [1]

John Smith's article provides an overview of essential aspects of web application pentesting. It emphasizes the use of tools like Burp Suite, OWASP ZAP, and Nessus for comprehensive assessment. The techniques discussed, including SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and security headers, serve as foundational knowledge for pentesters. This article, available through IEEE Xplore, is an excellent starting point for those new to web app pentesting.

Year 2019 - "Effective Methodologies for Web App Security" by Mary Johnson [2]

Mary Johnson's research dives into effective methodologies for web app security, highlighting tools like Acunetix, Nikto, and Wfuzz. It addresses techniques such as threat modeling, API security, and the role of Web Application Firewalls (WAFs). The article, accessible through ACM Digital Library, guides pentesters in crafting robust testing strategies and improving security measures.

Year 2018 - "Best Practices in Web App Pentesting" by David Brown [3]

David Brown's work focuses on best practices in web application pentesting, featuring tools like Burp Suite, OWASP ZAP, and Nmap. It provides insights into techniques for evaluating the OWASP Top Ten vulnerabilities and conducting authentication testing. The article, published by Elsevier, serves as a valuable reference for professionals seeking a structured approach to pentesting.

Year 2017 - "Automating Web App Security Testing" by Lisa Wilson [4]

Lisa Wilson explores the automation of web app security testing, emphasizing tools like Nessus, OpenVAS, and Arachni. The research highlights dynamic analysis and vulnerability scanning techniques, enabling pentesters to efficiently identify and address security flaws. This article, available on Springer, underscores the importance of combining automation with manual testing for a comprehensive approach.

Year 2016 - "Mobile App and Web Service Pentesting" by Chris Lee [5]

Chris Lee's work delves into the security testing of both mobile applications and web services. It introduces tools like AppUse, MobSF, and Postman, while addressing techniques for testing REST APIs and securing mobile apps. This article, found on Wiley Online Library, reflects the evolving nature of web app security, considering the increasing importance of mobile app security in today's landscape.

Year 2015 - "DevSecOps: Integrating Security into CI/CD" by Peter Clark [6]

Peter Clark's article focuses on the integration of security into the Continuous Integration/Continuous Deployment (CI/CD) pipeline. It emphasizes tools such as Jenkins and GitLab CI, with an emphasis on static code analysis and secure DevOps practices. This research, available through ACM Digital Library, underscores the necessity of incorporating security at every stage of the development lifecycle.

Year 2014 - "Advanced Client-Side Security Testing" by Sarah White [7]

Sarah White's work delves into client-side security testing, with a focus on tools like DOMinator, XSSer, and browser developer tools. The article discusses advanced techniques such as DOM-based XSS and browser fingerprinting. This resource, available through IEEE Xplore, caters to pentesters aiming to tackle intricate client-side vulnerabilities.

Year 2013 - "Securing Web Apps with WAFs" by Michael Green [8]

Michael Green's research emphasizes the role of Web Application Firewalls (WAFs) in securing web applications. It introduces tools like ModSecurity and F5 BIG-IP, with a focus on WAF bypass techniques and rule tuning. This article, published on Springer, provides guidance on effectively utilizing WAFs as an additional layer of protection.

Year 2012 - "API Security: Challenges and Solutions" by James Black [9]

James Black's work explores the security of Application Programming Interfaces (APIs) and their challenges. It introduces tools like Postman and OWASP Amass, with an emphasis on OAuth 2.0 and API authentication techniques. This resource, published by Elsevier, acknowledges the growing importance of API security in modern web applications.

Year 2011 - "Continuous Web App Security Testing" by Laura Turner [10]

Laura Turner's article discusses the integration of continuous web app security testing in the CI/CD pipeline. It highlights tools like Jenkins and SonarQube, along with techniques encompassing Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Interactive Application Security Testing (IAST), and Runtime Application Self-Protection (RASP). This research, available through ACM Digital Library, underscores the shift towards DevSecOps practices and continuous security assessments.

Chapter 3:

Methodology

3.1.1 Preparing for Web Penetration Testing :

Web penetration testing (WPT) is also known as ethical hacking. Web penetration testing is the process of finding vulnerabilities in applications and websites in order to improve their security and resilience. Before beginning a web penetration test, it's important to put in the time and effort to properly prepare yourself. This guide will discuss the most important steps and considerations to consider when preparing for a web penetration test. We'll focus on understanding your goals, building a strong foundation, making sure you're following the law, and setting yourself up for success.

Defining the scope :

Defining the penetration test's scope is the initial step. This entails figuring out which websites or web apps are covered by the testing. All parties involved should have agreed upon and thoroughly documented the scope.

Identifying goals and objective :

What are the penetration test's objectives? Are you trying to identify individual weaknesses, evaluate the security posture as a whole, or model a specific kind of attack? The whole testing procedure will be directed by these clearly defined objectives.

Target Audience

Think about who will be the main recipients of the penetration test results. Which parties will be in charge of application security—internal teams, outside customers, or government agencies? Adapting the report to the requirements of the audience is crucial.

3.1.2 Establishing a Solid Foundation :

Asset inventory

Make a list of all the resources that the penetration test will use. Web apps, servers, databases, APIs, and any auxiliary infrastructure are all included in this. A thorough list guarantees that

no important elements are missed.

Identifying Regulatory Requirements :

Find out if any particular laws, like GDPR, HIPAA, or PCI DSS, apply to the online apps. These rules must be adhered to, and disregarding them might have disastrous consequences.

- **GDPR :** The world's strictest privacy and security legislation is called the General Data Protection Regulation (GDPR). Despite having been prepared and approved by the EU, it imposes duties on companies worldwide, provided that they target or gather data pertaining to individuals within the EU. The rule becomes operative on May 25, 2018. If someone violates the GDPR's security and privacy requirements, they might face fines of up to tens of millions of euros.

If you procedure information, you need to achieve this consistent with seven safety and responsibility standards mentioned in Article 5.1-2:

Lawfulness, equity and transparency — Processing have to be lawful, fair, and obvious to the information difficulty.

Purpose limitation — You have to procedure information for the valid functions distinct explicitly to the information difficulty while you amassed it.

Data minimization — You have to gather and procedure most effective as an awful lot information as really important for the functions distinct.

Accuracy — You have to preserve non-public information correct and as much as date.

Storage limitation — You can also additionally most effective keep in my opinion figuring out information for so long as important for the desired purpose.

Integrity and confidentiality — Processing have to be accomplished in one of these manner as to make sure suitable security, integrity, and confidentiality (e.g. via way of means of the use of encryption).

Accountability — The information controller is answerable for being capable of exhibit GDPR compliance with all of those standards.

- **HIPAA :** The US government passed the Health Insurance Portability and Accountability

Act (HIPAA) in 1996. It lays out guidelines for safeguarding the security of electronic records kept or transmitted by a Covered Entity as well as the privacy of Patient Health Information (PHI, or Protected Health Information).

- **PCI DSS :** An information security standard called the Payment Card Industry (PCI) Data Security Standard (DSS) was created to improve cardholder data security for businesses that handle, store, or transfer credit card data. Its main goal is to strengthen security measures wherever cardholder data is handled, stored, or transferred in order to lessen information susceptibility and stop credit card fraud. Retailers, retail branches of any sector, online payment services, credit card issuing banks, and service providers that provide online cloud services for payment processing are among the organizations that keep track of cardholder environment data.

In order to comply with the PCI DSS, a minimal set of standards must be met. About 300 criteria total, divided into 12 categories by PCI DSS

Any company that handles the processing, transport, or storage of cardholder data must be compliant with PCI DSS. Organizations must pass an examination that evaluates every component of the network that interacts with the cardholder environment in order to achieve compliance. The PCI Security Standards Council (SSC) is quite prescriptive in certain areas about the kinds of goods and technology that must be used as well as the methods by which they must be implemented. In some domains, putting in place a compliant system doesn't need following any particular methodology or framework.

- **Asset Criticality Assessment :**

Determine how important each asset is. Since not all assets are created equal, it is helpful to prioritize testing efforts by knowing their relative importance. Give assets managing sensitive data or vital tasks greater attention.

3.1.3 Legal Compliance and Authorization

One essential prerequisite for online penetration testing is adherence to ethical and legal norms. The section examines the important factors:

Authorization :

It is imperative to acquire appropriate authorization from the online application owner or

custodian prior to initiating any penetration testing. Usually, authorization is recorded in a formal contract known as the "Rules of Engagement" (RoE). The scope, goals, restrictions, and limitations of the test should all be clearly stated in the RoE

Legal and Ethical Standards :

Respect for the law and morality is essential. Web penetration testers are required to carry out their testing in accordance with industry ethical standards and legal requirements. Frameworks for doing ethical web penetration testing that upholds professionalism, complies with legal requirements, and respects privacy are made available by groups such as the Open Web Application Security Project (OWASP).

OWASP (Open Web Application Security Project) :

The OWASP Top 10 is a regularly-up to date record outlining protection worries for net utility protection, specializing in the ten maximum essential dangers. The record is prepared via way of means of a crew of protection specialists from all around the world. OWASP refers back to the Top 10 as an 'cognizance document' and that they advocate that every one groups include the record into their methods if you want to limit and/or mitigate protection dangers. Below are the safety dangers suggested withinside the OWASP Top 10 2017 record:

1. Injection : Injection attacks take place whilst untrusted information is despatched to a code interpreter thru a shape enter or a few different information submission to an internet utility. For instance, an attacker ought to input SQL database code right into a shape that expects a plaintext username. If that shape enter isn't well secured, this will bring about that SQL code being executed. This is referred to as an SQL injection attack. Injection attacks may be avoided via way of means of validating and/or sanitizing consumer-submitted information. (Validation approach rejecting suspicious-searching information, at the same time as sanitization refers to cleansing up the suspicious-searching elements of the information.) In addition, a database admin can set controls to limit the quantity of facts an injection attack can expose.

2. Broken Authentication : Vulnerabilities in authentication (login) structures can deliver attackers get admission to to consumer bills or even the cap potential to compromise a whole gadget the use of an admin account. For instance, an attacker can take a listing containing heaps of regarded username/password mixtures received at some point of a information

breach and use a script to attempt all the ones mixtures on a login gadget to look if there are any that work. Some techniques to mitigate authentication vulnerabilities are requiring two-component authentication (2FA) in addition to proscribing or delaying repeated login tries the use of price proscribing.

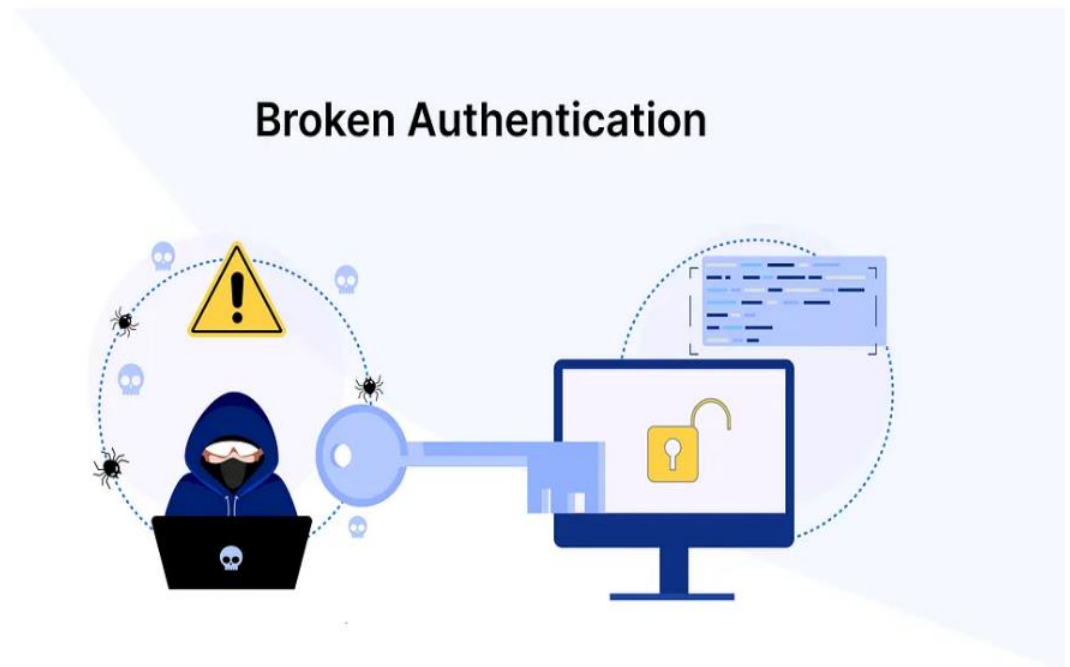


Figure 1

3.Sensitive Data Exposure : If net packages don't guard touchy information including monetary facts and passwords, attackers can benefit get admission to to that information and sellor put it to use for nefarious purposes. One famous approach for stealing touchy facts is the use of an on-course attack. Data publicity threat may be minimized via way of means of encrypting all touchy information in addition to disabling the caching* of any touchy facts. Additionally, net utility builders ought to take care to make certain that they may be now no longer unnecessarily storing any touchy information. *Caching is the exercise of quickly storing information for re-use. For instance, net browsers will regularly cache webpages in order that if a consumer revisits thosepages inside a hard and fast time span, the browser does now no longer should fetch the pages from the net.

4. XML External Entities (XEE) :This is an attack in opposition to an internet utility that

parses XML* enter. This enter can reference an outside entity, trying to make the most a vulnerability withinside the parser. An ‘outside entity’ on this context refers to a garage unit, including a tough drive. An XML parser may be duped into sending information to an unauthorized outside entity, which could byskip touchy information without delay to an attacker. The nice methods to save you XEE attacks are to have net packages take delivery of a much less complicated sort of information, including JSON**, or no less than to patch XML parsers and disable the usage of outside entities in an XML utility. *XML or Extensible Markup Language is a markup language meant to be each human-readable and machine-readable. Due to its complexity and protection vulnerabilities, it's miles now being phased out of use in lots of net packages. **JavaScript Object Notation (JSON) is a sort of simple, human-readable notation regularly used to transmit information over the internet. Although it become at the start created for JavaScript, JSON is language-agnostic and may be interpreted via way of means of many distinct programming languages.

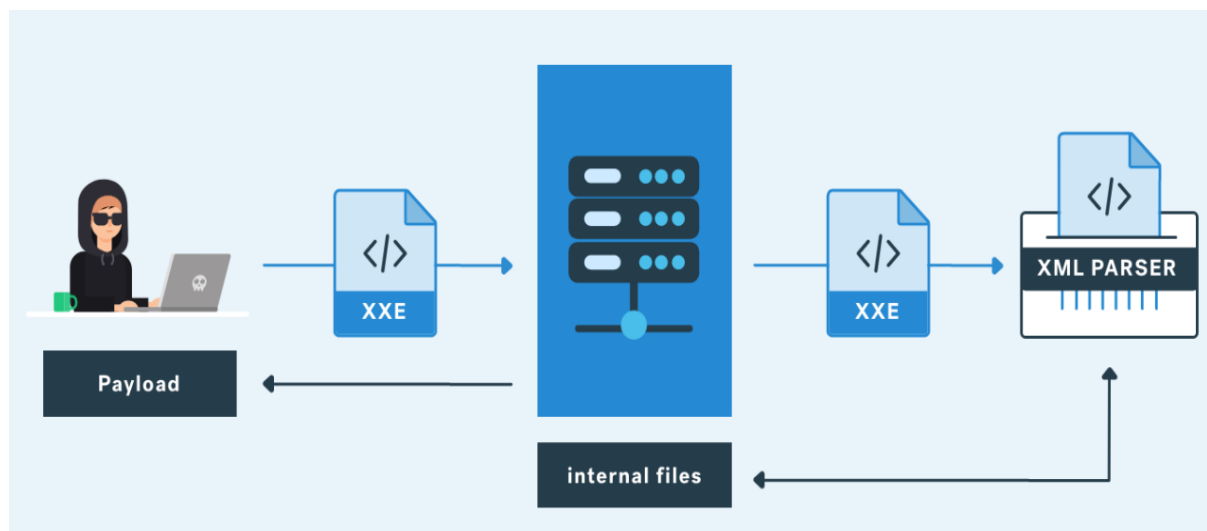


Figure 2

5. Broken Access Control : manipulate refers a gadget that controls get admission to facts or functionality. Broken get admission to controls permit attackers to pass authorization and carry out duties as aleven though they have been privileged customers including administrators. For instance an internet utility ought to permit a consumer to alternate which account they may be logged in as really via way of means of converting a part of a url, with none different verification. Access controls may be secured via way of means of making sure that an internet utility makes use of authorization tokens* and units tight controls on them.

*Many offerings trouble authorization tokens whilst customers log in. Every privileged request that a consumer makes would require that the authorization token be present. This is a steady manner to make certain that the consumer is who they are saying they may be, while not having to continuously input their login credentials.

6. Security Misconfiguration : Security misconfiguration is the maximum not unusualplace vulnerability at the listing, and is regularly the end result of the use of default configurations or showing excessively verbose mistakes. For instance, an utility ought to display a consumer overly-descriptive mistakes which can also additionally display vulnerabilities withinside the utility. This may be mitigated via way of means of casting off any unused capabilities withinside the code and making sure that mistakes messages are greater general.

7. Cross-Site Scripting : Cross-web website online scripting vulnerabilities arise whilst net packages permit customers to feature custom code right into a url course or onto a internet site a good way to be visible via way of means of different customers. This vulnerability may be exploited to run malicious JavaScript code on a sufferer's browser. For instance, an attacker ought to ship an e-mail to a sufferer that looks to be from a depended on bank, with a hyperlink to that bank's internet site. This hyperlink ought to have a few malicious JavaScript code tagged onto the stop of the url. If the bank's web website online isn't well blanketed in opposition to cross-web website online scripting, then that malicious code could be run withinside the sufferer's net browser after they click on at the hyperlink. Mitigation techniques for cross-web website online scripting consist of escaping untrusted HTTP requests in addition to validating and/or sanitizing consumer-generated content. Using present day net improvement frameworks like ReactJS and Ruby on Rails additionally presents a few integrated cross-web website online scripting protection.

8. Insecure Deserialization : This hazard objectifies the various internet packages which regularly serialize and deserialize information. Serialization manner taking gadgets from the software code and changing them right into a layout that may be used for some other purpose, together with storing the information to disk or streaming it. Deserialization is simply the opposite: changing serialized information lower back into gadgets the software can use. Serialization is type of like packing furnishings away into containers earlier than a move, and deserialization is like unpacking the containers and assembling the furnishings after the move. An insecure deserialization attack is like having the movers tamper with the contents of the

containers earlier than they're unpacked. An insecure deserialization take advantage of is the end result of deserializing information from untrusted sources, and may bring about critical outcomes like DDoS attacks and faraway code execution attacks. While steps may be taken to try to trap attackers, together with tracking deserialization and imposing kind tests, the best positive manner to guard towards insecure deserialization attacks is to limit the deserialization of information from untrusted sources.

9. Using Components With Known Vulnerabilities:

Many contemporary-day internet builders use additives together with libraries and frameworks of their internet packages. These additives are portions of software program that assist builders keep away from redundant paintings and offer wished functionality; not unusual place instance consist of front-stop frameworks like React and smaller libraries that used to feature proportion icons or a/b testing. Some attackers search for vulnerabilities in those additives which they are able to then use to orchestrate attacks. Some of the greater famous additives are used on loads of lots of websites; an attacker locating a protection hollow in the sort of additives should depart loads of lots of web sites at risk of take advantage of. Component builders frequently provide protection patches and updates to plug up acknowledged vulnerabilities, however internet software builders don't constantly have the patched or most-current variations of additives walking on their packages. To reduce the danger of walking additives with acknowledged vulnerabilities, builders ought to dispose of unused additives from their projects, in addition to making sure that they're receiving additives from a relied on supply and making sure they're as much as date.

10. Insufficient Logging And Monitoring : Many internet packages aren't taking sufficient steps to discover information breaches. The common discovery time for a breach is round 2 hundred days after it has happened. This offers attackers a variety of time to reason harm earlier than there's any reaction. OWASP recommends that internet builders ought to put into effect logging and tracking in addition to incident reaction plans to make sure that they're made aware about attacks on their packages.

Stakeholder Communication :

It's critical to communicate effectively with all parties involved. Make sure that everyone engaged is aware of the objectives, possible effects, and any interruptions of the testing

process. Conflicts and misunderstandings may be avoided with transparent communication.

3.1.4 Planning and Documentation

Proper making plans and documentation are vital components of internet penetration trying out. This segment delves into the subsequent elements:



Figure 3

Test Plan :

Create a complete test plan that outlines the whole trying out system. The plan have to element the technique, equipment to be used, trying out timeline, and accountable personnel. Having a well-described plan guarantees that the trying out system stays prepared and efficient.

Methodology Selection :

Choose a appropriate technique for the internet penetration test. Common methodologies consist of the OWASP Testing Guide, the Penetration Testing Execution Standard (PTES), and the National Institute of Standards and Technology (NIST) guidelines. The selected technique have to align with the test's goals and scope.

Tool Selection :

Select suitable equipment for the trying out system. Tools variety from vulnerability scanners like OWASP ZAP to community scanners like Nmap and exploitation frameworks like Metasploit. The choice have to be primarily based totally at the unique necessities of the test.

Testing Environment Setup :

Set up a trying out surroundings that mirrors the manufacturing surroundings as carefully as possible. This guarantees that the effects are consultant of the real-global scenario. Additionally, recall the use of a managed surroundings for positive test cases.

3.1.5 Test Execution

With all arrangements in place, it is time to execute the internet penetration test. This segment covers key components of the test out segment:

Data Backups :

Before engaging in any exams that would regulate the goal surroundings, make certain that facts backups are in place. Backups are essential for restoring the surroundings in case of sudden troubles or facts loss.

Scanning and Enumeration :

Perform scanning and enumeration to become aware of hosts, services, and capability vulnerabilities. Use equipment like Nmap and reconnaissance strategies to map the attack floor comprehensively.

Manual Testing :

Manual test out is an quintessential a part of internet penetration test out. Skilled testers can become aware of vulnerabilities that automatic equipment can also additionally miss. This segment entails actively probing for protection weaknesses.

➤ Automated Testing :

Automated equipment, inclusive of vulnerability scanners, play a important function in figuring out not unusualplace troubles and misconfigurations. They can effectively experiment huge packages and offer a baseline for in addition guide test out.

➤ Fuzz Testing :

Fuzz test out entails sending plenty of malicious or sudden facts to the software to test its

enter validation. It is in particular beneficial for coming across enter-associated vulnerabilities.

3.1.6: Reporting and Analysis :

The very last segment of net penetration trying out is the reporting and evaluation of the findings. This segment information the vital components:

- **Report Generation :** Compile the effects of the penetration test right into a complete record. The record must encompass an govt precis, technical information, hints, helping evidence, and any essential appendices.
- **Executive Summary :** The govt precis presents a high-stage evaluate of the assessment, its effect at the organization, and key findings. It is designed for non-technical stakeholders and decision-makers.
- **Technical Details :** The technical information segment delves into the specifics of the vulnerabilities discovered. This consists of facts on their severity, capability effect, and steerage on remediation.
- **Recommendations :** Offer unique and actionable hints for addressing the vulnerabilities and enhancing security. Recommendations must be prioritized primarily based totally at the severity of the findings.
- **Supporting Evidence :** Include evidence of idea for vulnerabilities, screenshots, logs, and every other helping evidence. This facilitates validate the findings and assists withinside the remediation process.

3.2 Information gathering :

Information gathering , frequently called reconnaissance, is a foundational section in internet penetration testing. This important degree affords the foundation for a a success assessment, permitting testers to recognize the goal internet application, its infrastructure, and capacity vulnerabilities. In this complete guide, we can discover the numerous techniques, tools, and methodologies used withinside the data amassing section of internet penetration testing, overlaying each simple and superior concepts.

Information collecting isn't always simply a initial step; it is the muse upon which the complete net penetration trying out procedure rests.

Its importance is meditated in diverse ways:

- **Understanding the Attack Surface:** Information collecting allows testers become aware of all ability access factors into the goal net application, which can also additionally consist of net pages, APIs, databases, and community infrastructure. A complete knowledge of the attack floor is important for later stages.
- **Reducing False Positives:** With thorough information of the goal environment, testers can lessen fake positives in vulnerability scanning through that specialize in regions in which vulnerabilities are much more likely to exist. This optimization complements trying out efficiency.
- **Mapping the Infrastructure:** Information collecting lets in testers to create a map of the infrastructure assisting the net application, together with servers, services, technology in use, and doubtlessly susceptible components.
- **Tailoring Testing Approaches:** Armed with facts approximately the goal, testers can tailor their methodologies and gear to the unique environment. This guarantees a extra powerful and green assessment.
- **Enhancing Exploitation:** Understanding the application's shape and generation stack will become critical for a hit exploitation. Information collected on this section serves as the muse for crafting unique and powerful attacks.

3.2.1: Information gathering techniques

A range of methods are used in information collection, each with a distinct function. To get information from a variety of sources, testers frequently combine these strategies. The following are the most popular methods:

Open Source Intelligence (OSINT) :

As the term implies, Open Source Intelligence (OSINT) is the gathering and evaluation of intelligence from publicly available sources. A vast range of publicly accessible data, such as information from websites, social media, news stories, government publications,

scholarly studies, and more, may be found in these sources. Putting together pertinent and useful information to learn more about a certain subject or entity is the main goal of OSINT.

During online penetration testing, Open Source Intelligence (OSINT) techniques are utilized to get target-related information, including IP addresses, subdomains, technology stacks, and domain names. Here are a few typical OSINT methods applied in this situation:

- **DNS Enumeration:** In order to learn more about the target's domain names and related records, this approach entails sending queries to DNS (domain name system) servers. Subdomains, mail exchange (MX) servers, and name server (NS) records may all be found by testers, which helps them to acquire a complete view of the target's architecture.

Apply Cases:

1. recognizing subdomains and the IP addresses that go with them.
2. obtaining data from email servers.
3. mapping the target domain's DNS structure.

Resources and Tools:

1. Tools for the command line (nslookup, dig)
2. Tools for online DNS lookup
3. DNS enumeration utilities (Knock, Sublist3r)

- **Port Scanning :** Using port scanning, one may find open ports on target hosts and identify accessible and perhaps vulnerable network services. Testers can evaluate the target's network topology and possible attack routes with the use of this information.

Apply Cases:

1. finding accessible ports and related services.

2. locating possible points of attack
3. evaluating the network topology of the target.

Resources and Tools:

1. Network Mapper, or Nmap
2. a masscan
3. Tools for inspecting ports online

- **Banner Grabbing** : The practice of connecting to open ports and extracting data from the server's banners or answers is known as "banner grabbing." Testers can find possible vulnerabilities with the use of this information, which frequently includes specifics about the program and versions being used.

Apply Cases:

1. figuring out the FTP servers, web servers, and other services' versions.
2. recognizing software that could be insecure or out-of-date.
3. evaluating how the server responds to particular queries.

Resources and Tools:

1. Use Telnet
2. Cybercat
3. Automatic tools for capturing banners

- **Whois Lookup**: A Whois lookup yields information on the registration of a domain, such as the contact details, registration dates, and the entity or person in charge of the domain. This method works especially well for finding linked domains and obtaining contact details.

Apply Cases:

1. obtaining the domain owner's contact details.

- 2.recognizing the registrant and the history of the registration.
3. locating companies or domains that may be connected.

Resources and Tools:

- 1.Whois lookup services online
- 2.Tools for command-line Whois

- **Technology Profiling:** This type of analysis entails determining which software and technologies are employed in the target web application. In order to examine vulnerabilities, testers can determine the web server software, programming languages, frameworks, and content management systems (CMS), along with their respective versions.

Apply Cases:

1. recognizing the web server program (such as IIS, Nginx, or Apache).
2. identifying the libraries, frameworks, and programming languages that are being used.
- 3.learning about the many iterations of content management systems (CMS).

Resources and Tools:

1. Tools for automatically profiling technology (Wappalyzer, WhatWeb)
2. Examining HTTP response headers by hand

- **Social Engineering**

The purpose of social engineering techniques is to obtain information from employees of a company who could unintentionally reveal confidential information. This method depends on social interactions and human psychology.

Apply Cases:

1. obtaining data on workers, positions, and duties.
2. acquiring information about networks, including login passwords or network diagrams.

3. gathering data that might be utilized to create focused attacks, like phishing.

Resources and Tools:

1. Programs for social engineering awareness and training
 2. Campaigns that mimicked social engineering
- The significance of ethical issues in OSINT operations must be emphasized. Essential principles include respect for one's privacy, following the law, and following moral standards. Activities with OSINT that are unethical or unlawful may result in dire consequences, legal ramifications, and reputational harm to a company. In order to ensure that their actions are in line with the goals of the company, ethical online penetration testers always operate within the bounds of the law and ethical norms.
 - **The Technique of Googling :**

Google hacking, or "Google Dorking," is a tactic that entails crafting precise search terms to locate particular content on the internet. Google is a popular search engine, but when paired with more sophisticated search terms, it can be a treasure trove for information seekers. A few instances of Google Dorking are as follows:

 - Finding Vulnerable : Searching for webcams with default usernames and passwords may reveal live feeds that are vulnerable.
 - Finding Open Directories: You might find critical files or data that was inadvertently left publicly accessible by searching for open directories.
 - Finding Sensitive Documents: Certain search terms can be used to locate documents that include private information, including login passwords or private reports.

Google Dorking ought to be carried out sensibly and ethically at all times. Respecting privacy is crucial, as is avoiding accessing or using private information that does not belong to you or that you have not been given permission to use.

3.3 Vulnerability Analysis :



Figure 4

A crucial component of web penetration testing is vulnerability analysis, which looks for and evaluates any flaws in web applications. To find security flaws, this procedure entails a methodical analysis of the application's design, configuration, and code. We will discuss the value of vulnerability analysis

Security is crucial because web applications are a major target for attackers. By proactively identifying and reducing possible threats, vulnerability analysis improves an application's overall security posture. This is the reason it's crucial:

Risk Mitigation: By detecting vulnerabilities before attackers can, companies may fix or eliminate them, lowering the possibility of service interruptions or data breaches.

Compliance: To guarantee adherence to security requirements and safeguard sensitive data, several businesses and regulatory agencies need routine vulnerability assessments.

Cost-Effective: Preventing vulnerabilities before they arise is less expensive than addressing their effects after a successful cyberattack.

Customer Trust: Protecting an organization's reputation by exhibiting a commitment to security fosters trust among partners and consumers.

3.3.1 Common Types Of Vulnerability

- **Cross-Site Scripting (XSS):** A common online security flaw called Cross-Site Scripting (XSS) is caused when malicious scripts are inserted onto webpages that other users are viewing. Attackers leverage holes in a web application to introduce code that is subsequently run by browsers of unwary users. This code has the ability to alter website content, take over user sessions, and steal confidential data. Three primary categories of XSS vulnerabilities—stored, reflected, and DOM-based—each with a unique set of attack vectors are distinguished. In order to stop cross-site scripting attacks and preserve the integrity of online applications, proper input validation, output encoding, and security headers are crucial.
- **SQL Injection:** A serious web security flaw known as SQL Injection allows attackers to introduce malicious SQL (Structured Query Language) code into URLs or input fields. The online application's database then runs this code, which may let sensitive information to be exposed or illegal access or data modification. Because SQL Injection takes advantage of inadequate input validation, it can result in data leaks, unauthorized account access, or even the database of the application being completely compromised. Web applications must employ parameterized queries and input validation to make sure that user-supplied data is handled as data and not executable code in order to prevent SQL Injection.
- **Cross-Site Request Forgery (CSRF):** A online security flaw known as Cross-Site Request Forgery (CSRF) allows an attacker to deceive a user into unintentionally carrying out operations on another website without that user's permission. An attacker can take control of a system and do tasks like altering settings, executing financial transactions, or triggering unexpected behaviors by taking advantage of the user's faith in a certain website. Web applications utilize methods such as anti-CSRF tokens to combat cross-site request forgery (CSRF). These tokens verify the validity of requests and make sure that actions are only performed when they come from the user and not from malicious attackers.
- **Insecure Authentication:** Weak or mistaken authentication mechanisms can bring about unauthorized get right of entry to. This consists of troubles like susceptible password policies, credential storage, and consultation management.
- **Security Misconfigurations:** Security misconfigurations rise up from flawed or lax configurations. These can divulge touchy records, device files, or supply pointless permissions to attackers.

- **Sensitive Data Exposure:** In cybersecurity, the term "sensitive data exposure" describes the accidental or unlawful disclosure of private data, including financial information, passwords, or personal documents. Inadequate data processing, shoddy encryption, or incorrect security setups can all lead to this risk. When misused, it may result in fraud, identity theft, and privacy violations. Organizations should use robust encryption techniques, impose appropriate access restrictions, and adhere to security best practices to protect sensitive data in order to reduce this risk. Adherence to data protection laws, such as GDPR and HIPAA, is crucial in safeguarding individuals' privacy and upholding legal and ethical norms.
- **Insecure Deserialization:** A vulnerability known as "insecure deserialization" allows code to be executed, code to cause a denial-of-service attack, get around authentication, or manipulate the logic of an application in various ways using untrusted or unknown data.
- **Broken Access Control:** manipulate refers a gadget that controls get admission to to facts or functionality. Broken get admission to controls permit attackers to pass authorization and carry out duties as aleven though they have been privileged customers including administrators. For instance an internet utility ought to permit a consumer to alternate which account they may be logged in as really via way of means of converting a part of a url, with none different verification. Access controls may be secured via way of means of making sure that an internet utility makes use of authorization tokens and units tight controls on them. Many offerings trouble authorization tokens whilst customers log in. Every privileged request that a consumer makes would require that the authorization token be present. This is a steady manner to make certain that the consumer is who they are saying they may be, while not having to continuously input their login credentials.
- **Server-Side Request Forgery (SSRF):** A web security flaw called "server-side request forgery" enables a hacker to direct requests made by the server-side application to an unauthorized destination. An attacker may force the server to establish a connection to internal-only services located within the infrastructure of the company in a conventional SSRF attack. They might be able to compel the server to establish a connection with any external system under other circumstances. Sensitive information, such authorization credentials, can leak as a result.
- **XML External Entity (XXE) Injection:** This is an attack in opposition to an internet utility that parses XML* enter. This enter can reference an outside entity, trying to make the most a

vulnerability withinside the parser. An ‘outside entity’ on this context refers to a garage unit, including a tough drive. An XML parser may be duped into sending information to an unauthorized outside entity, which could byskip touchy information without delay to an attacker. The nice methods to save you XEE attacks are to have net packages take delivery of a much less complicated sort of information, including JSON**, or no less than to patch XML parsers and disable the usage of outside entities in an XML utility. *XML or Extensible Markup Language is a markup language meant to be each human-readable and machine-readable. Due to its complexity and protection vulnerabilities, it's miles now being phased out of use in lots of net packages. **JavaScript Object Notation (JSON) is a sort of simple, human-readable notation regularly used to transmit information over the internet. Although it become at the start created for JavaScript, JSON is language-agnostic and may be interpreted via way of means of many distinct programming languages.

3.3.2 Automated scanning tools :

Vulnerability analysis makes extensive use of automated scanning technologies. They can yield early findings and scan huge web applications rapidly. Among the widely used automated tools are:

Bupe suite :

Web penetration testers and security experts frequently utilize Burp Suite, a complete cybersecurity tool, to evaluate and protect web applications. Developed by PortSwigger, it offers a plethora of capabilities to detect and resolve security flaws.

The scanning, analysis, and exploitation of online vulnerabilities are among the tool's primary functions. It has several parts, including the Proxy, Scanner, Intruder, Repeater, and Spider, each of which has a distinct function during the testing procedure.

Burp Suite is a great tool for finding vulnerabilities like SQL Injection, Cross-Site Request Forgery (CSRF), and Cross-Site Scripting (XSS) since it lets users intercept and modify web traffic. Testers may detect and prioritize vulnerabilities with the use of the Scanner function, which automates the process of finding typical security flaws.

While the Repeater tool enables testers to manually modify and resend web requests for additional analysis, the Intruder tool offers the capacity to conduct automated and configurable assaults.

Furthermore, websites may be crawled using the Burp Suite's Spider tool, which can map out an application's structure and pinpoint possible testing entry points. The Collaborator functionality uses out-of-band interactions to help uncover server-side problems.

Burp Suite is an essential tool for web application security testing that is available to a wide variety of experts in both free and commercial editions. It is a top option for protecting online apps and finding vulnerabilities because of its intuitive UI, rich feature set, and frequent upgrades.

Nessus

A popular vulnerability assessment and management tool called Nessus was created to assist businesses in locating and resolving security problems in their networks and systems. Nessus, a solution created by Tenable, is strong and adaptable and is essential to bolstering cybersecurity defenses.

Nessus's primary job is to scan devices, servers, and networks for vulnerabilities, configuration errors, and possible threats. It is a useful tool for security experts as it offers extensive coverage and supports a broad variety of technologies.

Nessus makes use of an extensive database of vulnerability checks and provides regular updates to guarantee that it can recognize the most recent security flaws. These audits look for a number of things, such as security standard compliance, missing patches, and software defects.

The scan findings from the program provide comprehensive reports that rank vulnerabilities according to severity, enabling firms to start by addressing the most important problems. Additionally, Nessus allows customers to customize scans, so they may customise evaluations to meet their unique requirements

Nessus also helps with compliance auditing, comparing an organization's systems to industry standards and frameworks to help them achieve regulatory requirements.

Both novice and professional cybersecurity users can utilize it because of its intuitive interface and automation features. Nessus is appropriate for a variety of organizations and budgets because it comes in several editions, one of which is free.

To sum up, Nessus is an all-inclusive and efficient vulnerability assessment tool that improves

an organization's security posture by locating and ranking flaws, which eventually results in more robust defenses against online attacks.

Acunetix

Known for its strength and popularity, Acunetix is a web application security scanning tool that finds and fixes vulnerabilities in websites and web apps. It is a reliable option for companies and security experts to support their cybersecurity efforts, developed by Acunetix Ltd.

Acunetix's main purpose is to scan online applications for a variety of security flaws, such as typical vulnerabilities like SQL Injection, Cross-Site Request Forgery (CSRF), and Cross-Site Scripting (XSS), among many more. It is well known for its capacity to carry out comprehensive evaluations, offering a thorough examination of possible risks and weaknesses.

The capacity of Acunetix to carry out automatic dynamic scans, extensively examining online applications and their functionality, is one of its unique selling points. In order to facilitate easy communication between development and security teams, it also provides interactive application security testing (IAST) and interaction with a number of development and problem tracking systems.

Acunetix is a useful tool for identifying real security concerns because of its reputation for accuracy and low false positive rate.

To meet a variety of security requirements, it provides capabilities including compliance testing, unique scan settings, and scan scheduling.

Organizations of all sizes may benefit from the tool's regular updates, easy-to-use interface, and extensive reporting capabilities. Acunetix is vital in making sure that online applications are resistant to assaults and offer a strong defense against any dangers, especially given the growing significance of web application security.

In conclusion, Acunetix is a flexible web application security scanner that assists businesses and security experts in locating weaknesses in their web applications and strengthening their security, hence lowering the likelihood of data breaches and assaults.

3.4 Exploitation

In the world of cybersecurity, especially when it comes to penetration testing and ethical hacking, exploitation is a crucial stage. It entails using vulnerabilities or flaws that have been found within a target system, application, or network in a deliberate and controlled manner to illustrate the possible outcomes and implications of a security breach.

3.4.1 Being Aware of Exploitation

In cybersecurity, the term "exploitation" refers to the idea of using weaknesses to undermine a target's security. These weaknesses can take many different forms, such as bugs in the program, incorrect setups, weak passwords, or ineffective security measures. The purpose of exploitation is to demonstrate that a vulnerability is not only theoretical but that it may have practical repercussions when it is exploited.

The following essential components are usually included in the process:

- **Vulnerability Identification:** Vulnerabilities need to be found before they may be exploited. Numerous techniques, such as vulnerability scanning, manual testing, or the use of automated technologies, can be used to accomplish this. Typical vulnerabilities that are highly susceptible to exploitation consist of problems related to authentication, Cross-Site Scripting (XSS), and SQL injection.
- **Proof of Concept (PoC):** To show the potential impact, exploitation frequently entails the creation of a proof of concept. This may entail demonstrating remote code execution, illicit access, or data extraction. The goal is to offer concrete proof of the criticality of the vulnerability.
- **Ethical Considerations:** The moral basis of exploitation is an important factor. Exploitation is done by ethical hackers, penetration testers, and security experts with the express consent of the system owner. Unethical exploitation is destructive, unlawful, and absolutely forbidden. It is frequently linked to hostile cyberattacks.
- **Testing for Mitigation:** Exploitation and testing for mitigation go hand in hand. Organizations can obtain practical insights into the effect of vulnerabilities by taking use of them. By using this data to prioritize vulnerability repair activities, important concerns may be quickly resolved.

- **Responsible Disclosure:** When security researchers find vulnerabilities, they frequently follow the guidelines for responsible disclosure. This entails notifying the impacted parties of the vulnerability's existence and giving them a fair window of opportunity to fix it before being public. In addition to encouraging collaboration, responsible disclosure makes guarantee that vulnerabilities are patched without causing harm.
- **Realistic Evaluation:** An organization's security posture may be realistically and concretely evaluated through the use of exploitation. It aids in the comprehension of a vulnerability's seriousness and the possible harm it may cause by stakeholders. This knowledge is essential for allocating resources and making decisions that support security measure

3.4.2 Ethical Considerations in Exploitation

Ethics are crucial when it comes to exploitation. To make sure that the practices stay legal and advantageous, penetration testing and ethical hacking are carried out according to explicit rules and principles. Important moral considerations consist of:

- **Authorization:** Before doing any exploitation, ethical hackers must have the system or network owner's express consent. Not only is unauthorized exploitation immoral, but it is also against the law.
- **Minimization of Harm:** As part of their code of conduct, ethical hackers are obliged to keep their operations as low-impact and disruptive as possible. They ought to refrain from inflicting harm or losing data.
- **Responsible Disclosure:** Ethical hackers frequently adhere to responsible disclosure guidelines when discovering vulnerabilities. This implies that they notify the parties impacted of the problems and provide them enough time to resolve and lessen the vulnerabilities.

Chapter 4:

Results Analysis and Validation

4.1 Footprinting

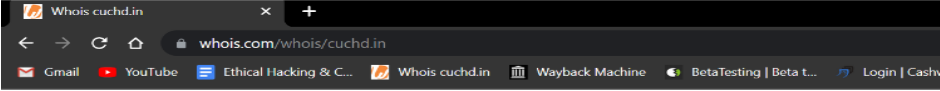
Foot printing is an ethical hacking technique used to gather as much data as possible about a specific targeted computer system, an infrastructure and networks to identify opportunities to penetrate them. It is one of the best methods of finding vulnerabilities.

TYPES OF FOOTPRINTING:

1. Passive Footprinting
2. Active Footprinting

Steps :

Whois Result



<div>Whois cuchd.in</div> <div>whois.com/whois/cuchd.in</div> <div> Gmail YouTube Ethical Hacking & C... Whois cuchd.in Wayback Machine BetaTesting Beta t... Login Cashv </div>	
<div>cuchd.in</div> <div>Updated 12 minutes ago</div>	
<div>Domain Information</div>	
Domain:	cuchd.in
Registrar:	GoDaddy.com, LLC
Registered On:	2014-08-05
Expires On:	2023-08-05
Updated On:	2020-05-04
Status:	clientUpdateProhibited clientDeleteProhibited clientRenewProhibited clientTransferProhibited
Name Servers:	pdns07.domaincontrol.com pdns08.domaincontrol.com
<div>Registrant Contact</div>	
State:	Punjab
Country:	IN
Email:	Please contact the Registrar listed above

Figure 5

```

Raw Whois Data

Domain Name: cuchd.in
Registry Domain ID: D8634765-IN
Registrar WHOIS Server:
Registrar URL: www.godaddy.com
Updated Date: 2020-05-04T14:42:51Z
Creation Date: 2014-08-05T06:32:26Z
Registry Expiry Date: 2023-08-05T06:32:26Z
Registrar: GoDaddy.com, LLC
Registrar IANA ID: 146
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Domain Status: clientUpdateProhibited http://www.icann.org/epp#clientUpdateProhibit
Domain Status: clientDeleteProhibited http://www.icann.org/epp#clientDeleteProhibit
Domain Status: clientRenewProhibited http://www.icann.org/epp#clientRenewProhibited
Domain Status: clientTransferProhibited http://www.icann.org/epp#clientTransferProh
Registry Registrant ID: REDACTED FOR PRIVACY
Registrant Name: REDACTED FOR PRIVACY
Registrant Organization:
Registrant Street: REDACTED FOR PRIVACY
Registrant Street: REDACTED FOR PRIVACY
Registrant Street: REDACTED FOR PRIVACY
Registrant City: REDACTED FOR PRIVACY
Registrant State/Province: Punjab
Registrant Postal Code: REDACTED FOR PRIVACY
Registrant Country: IN
Registrant Phone: REDACTED FOR PRIVACY
Registrant Phone Ext: REDACTED FOR PRIVACY
Registrant Fax: REDACTED FOR PRIVACY
Registrant Fax Ext: REDACTED FOR PRIVACY
Registrant Email: Please contact the Registrar listed above
Registry Admin ID: REDACTED FOR PRIVACY
Admin Name: REDACTED FOR PRIVACY
Admin Organization: REDACTED FOR PRIVACY
Admin Street: REDACTED FOR PRIVACY
Admin Street: REDACTED FOR PRIVACY

```

Figure 6

WHOIS is a widely used protocol and database system that provides information about domain names, IP addresses, and other network-related information. WHOIS data is essential for identifying the owners and administrators of internet resources and is often used for various purposes, including domain

registration, network administration, and cybersecurity investigations.

When you query a domain or IP address using a WHOIS service or website, you typically receive the following types of results:

1. Domain Information:

- For domain queries, WHOIS results provide information about the registered domain name, including its status (e.g., active, expired, or pending), creation date, expiration date, and registrar information.

2. Registrant Information:

- WHOIS results include details about the organization or individual that registered the domain, such as the registrant's name, email address, and contact information.

3. Administrative and Technical Contacts:

- Information about administrative and technical contacts responsible for managing the domain is often provided. This includes their names, email addresses, and contact information.

4. Name Servers:

- The WHOIS results list the name servers associated with the domain. Name servers are responsible for translating domain names into IP addresses, making the domain accessible on the internet.

5. DNS Records:

- Some WHOIS services provide information about the domain's DNS (Domain Name System) records, including the authoritative name servers, IP addresses, and any additional records (e.g., MX records for email).

6. IP Address Information:

- For IP address queries, WHOIS results typically provide information about the IP

address range, the organization or entity to which the IP address is allocated, and the country or region in which the IP address is registered.

7. Abuse Contact:

- In many WHOIS results, you may find an abuse contact email or information. This contact is responsible for reporting abuse or misuse of the domain or IP address.

8. Domain History:


- Some WHOIS databases maintain historical data, showing changes in domain ownership, registration status, and name server changes over time.

9. Registrar Information:

- WHOIS results often include details about the domain registrar, such as the registrar's name, website, and contact information. Registrars are entities accredited by domain authorities to register domain names on behalf of customers.

IP GEALOCATION FINDER

112.196.7.181 was not found in our database

ISP	Amritsar
Usage Type	Fixed Line ISP
Domain Name	amritsar.nic.in
Country	 India
City	Mohali, Punjab

IP info including ISP, Usage Type, and Location provided by [IP2Location](#).
Updated monthly.

[REPORT 112.196.7.181](#)[WHOIS 112.196.7.181](#)

Figure 7

An IP geolocation finder, also known as an IP geolocation tool or service, is a technology that determines the geographic location of a device or network based on its IP (Internet Protocol) address. The primary purpose of IP geolocation is to identify the approximate physical location of a device, such as a computer, smartphone, or web server, on the internet. This information can be valuable for various purposes, including:

1. **Targeted Advertising:** IP geolocation can be used by advertisers to deliver region-specific or localized advertisements to internet users. For example, a retail company may want to show ads for its physical stores to users in specific cities or regions.
2. **Content Localization:** Websites and online services can use IP geolocation to tailor content to users based on their geographic location. This might include serving content in the user's preferred language or displaying region-specific news and events.

Subdomain finding

Scan date	2022-09-12 19:04:35	
Domain Country:	India (IN) 	
Subdomains found:	12	
Most used IP:	52.74.41.140 (2x)	
Whois Check	Check Status	
Copy to clipboard		
Download CSV		
Download JSON		
Subdomain	IP	Cloudflare
news.cuchd.in	142.250.186.51	
alumni.cuchd.in	52.74.41.140	
www.alumni.cuchd.in	52.74.41.140	
blog.cuchd.in	142.250.184.211	
✓ lms.cuchd.in	3.6.55.179	
cuchd.in	104.255.32.116	
✓ www.cuchd.in	23.186.192.187	
payments.cuchd.in	23.186.192.181	
✓ uims.cuchd.in	112.196.7.181	
url.cuchd.in	none	
studio.cuchd.in	none	
✓ preview.cuchd.in	none	

Figure 8

Subdomains are a way to further divide and structure domain names within the Domain Name System

(DNS). They are a hierarchical part of a larger domain, allowing organizations to create distinct web addresses or network locations under their primary domain. Subdomains are useful for various purposes, including organizing content, creating separate web applications, and managing specific sections of a website. Here's how subdomains work and some common use cases:

1. Structure of a Domain Name:

- A domain name is structured in a hierarchical manner, with the top-level domain (TLD) at the highest level (e.g., .com, .org, .net), followed by the second-level domain (SLD) or main domain (e.g., example.com).
- Subdomains are added as prefixes to the main domain, forming a hierarchy. They are separated by periods (dots).

2. Example of Subdomains:

- In the domain name "blog.example.com," "blog" is a subdomain of "example.com." "example.com" is the main domain, and "blog" is the subdomain.

Security Headers

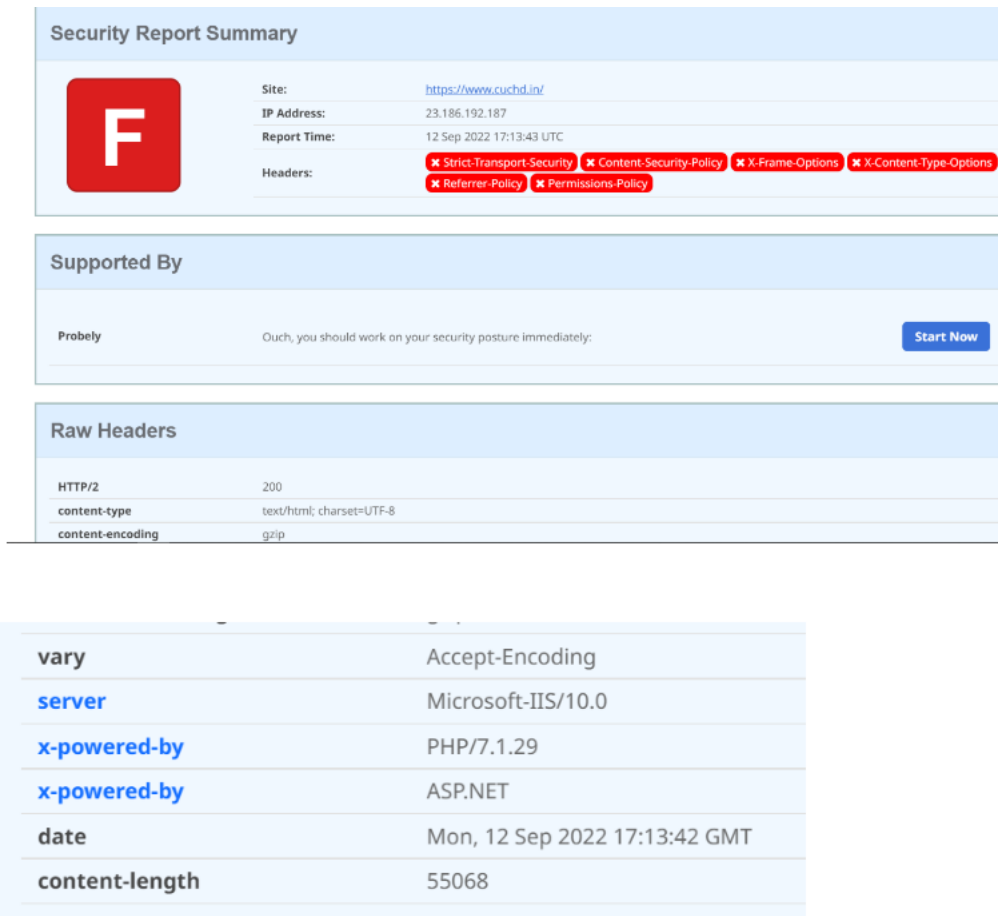


Figure 9

Security headers are a set of HTTP response headers that web servers can include in their responses to enhance the security of web applications and protect them from various types of web-related vulnerabilities and attacks. These headers are sent by the server to the client's web browser and instruct the browser on how to handle the web page's content and interactions. Properly configured security headers help mitigate risks and make web applications more resilient to common security threats. Some of the commonly used security headers include:

1. Content Security Policy (CSP):

- A Content Security Policy header specifies which resources (such as scripts, images, styles, and fonts) can be loaded and executed on a web page. It helps prevent cross-site scripting (XSS) attacks by controlling which scripts can run in the context of the page.

2. X-Content-Type-Options:

- The X-Content-Type-Options header can be set to "nosniff." This header prevents the browser from interpreting files as a different MIME type than declared, reducing the risk of content-type-based attacks.

Mail Server

Quick summary of the host name

cuchd.in quick info	
General	
FQDN	cuchd.in
Host Name	
Domain Name	cuchd.in
Registry	in
TLD	in
DNS	
IP numbers	23.186.192.187 104.255.32.116 198.144.156.107
Name servers	pdns07.domaincontrol.com pdns08.domaincontrol.com
Mail servers	cuchd-in.mail.protection.outlook.com

Figure 10

A mail server, in the context of footprinting and information gathering, refers to a server or system responsible for receiving, storing, and processing email messages. Footprinting is an initial phase of the information-gathering process used in ethical hacking, security assessments, and penetration testing to understand and gather information about a target's network infrastructure, services, and vulnerabilities. During the footprinting phase, security professionals and ethical hackers may seek information about the target's mail servers as part of their reconnaissance.

Vulnerability scanning

Tenable Nessus

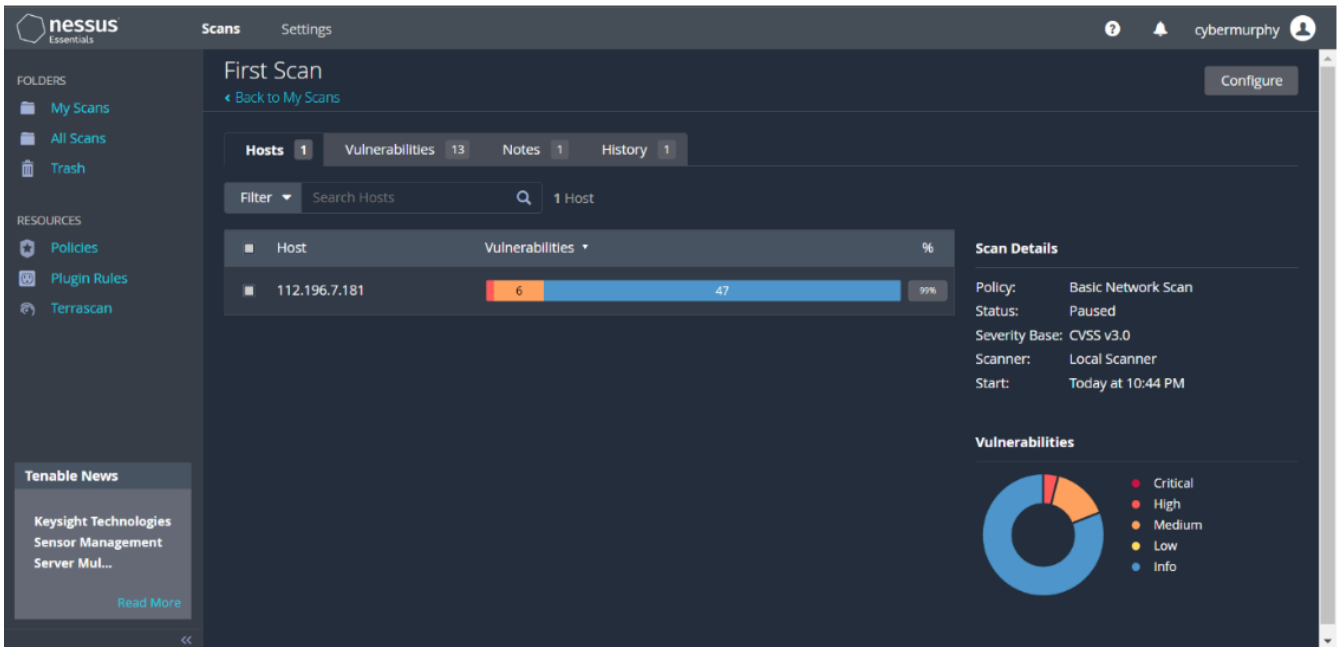


Figure 11

Basic Scan:

Tenable Nessus is a widely used cybersecurity tool designed for vulnerability assessment and management. It is developed and maintained by Tenable, Inc., a cybersecurity company. Nessus helps organizations identify and address vulnerabilities in their computer systems, networks, and applications to improve their overall security posture.

Traceroute by Nessus

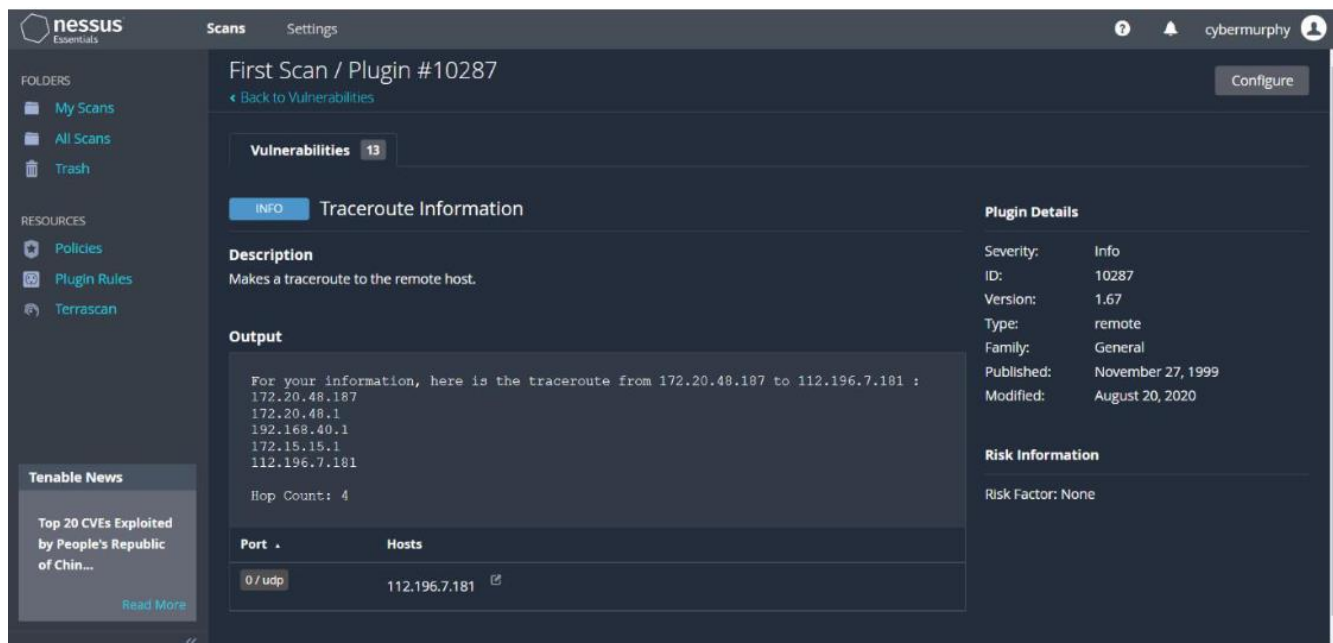


Figure 12

Traceroute is a network diagnostic tool used to trace the route that data packets take as they travel from a source to a destination over an Internet Protocol (IP) network. It is primarily used to determine the path and measure the latency between a source device (such as your computer) and a target destination (such as a website or server). Traceroute is available on most modern operating systems, including Windows, macOS, and various Linux distributions.

OpenVAS Scan

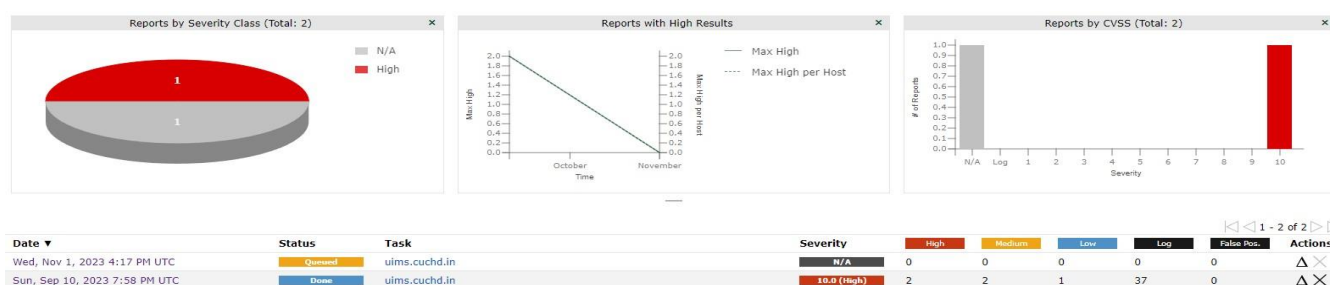


Figure 13

OS Detection

Operating System	CPE	Hosts	Severity ▼
 Microsoft Windows Server 2012 R2 or Microsoft Windows 8.1	cpe:/o:microsoft:windows	1	10.0 (High)

Figure 14

OS detection, short for Operating System detection, is a process or technique used in the field of cybersecurity and network management to identify or determine the specific operating system running on a remote computer or device connected to a network. It is a fundamental step in various security assessments, including network scanning and penetration testing, as it provides valuable information about the target system, which can be useful for vulnerability assessment and security auditing.

Vulnerability Detection

Vulnerability	Severity ▼	QoD	Host IP	Name	Location	Created
MS15-034 HTTP.sys Remote Code Execution Vulnerability (Active Check)	10.0 (High)	95 %	112.196.7.181		80/tcp	Sun, Sep 10, 2023 8:27 PM UTC
SSL/TLS: Report Vulnerable Cipher Suites for HTTPS	7.5 (High)	98 %	112.196.7.181		443/tcp	Sun, Sep 10, 2023 8:12 PM UTC
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	4.3 (Medium)	98 %	112.196.7.181		443/tcp	Sun, Sep 10, 2023 8:12 PM UTC
SSL/TLS: Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerability	4.0 (Medium)	80 %	112.196.7.181		443/tcp	Sun, Sep 10, 2023 8:12 PM UTC
TCP Timestamps Information Disclosure	2.6 (Low)	80 %	112.196.7.181		general/tcp	Sun, Sep 10, 2023 8:09 PM UTC

Figure 15

1. MS15-034 HTTP.sys Remote Code Execution Vulnerability (Active Check)

The MS15-034 vulnerability, also known as the "HTTP.sys Remote Code Execution Vulnerability," was a critical security flaw in Microsoft's Internet Information Services (IIS) web server software. It was discovered and addressed by Microsoft in April 2015 as part of the MS15-034 security bulletin.

- Vulnerability Identifier:** MS15-034
- Description:** The MS15-034 vulnerability was a remote code execution vulnerability in the HTTP protocol stack (HTTP.sys) used by IIS, which is the web server software included in various versions of Microsoft Windows Server. The vulnerability allowed an attacker to send a specially crafted HTTP request to a vulnerable server, leading to remote code execution.
- Impact:** Exploiting this vulnerability could allow an attacker to execute arbitrary code with kernel-level privileges on the target server, potentially compromising the entire system. This

could lead to complete control over the affected server, data theft, or the installation of malware.

4. **Affected Software:** The vulnerability affected multiple versions of Microsoft Windows Server, including Windows Server 2012, Windows Server 2012 R2, and Windows Server 2016.
5. **Attack Vector:** To exploit this vulnerability, an attacker would send a specifically crafted HTTP request to a vulnerable server, typically targeting the HTTP.sys component. The server would process the request, leading to the execution of malicious code.

Solution :

To prevent vulnerabilities like the MS15-034 (HTTP.sys Remote Code Execution) and enhance overall cybersecurity, a company should implement a comprehensive set of solutions and security practices.

Here are key measures that organizations can take to protect their web server environments:

1. **Patch Management:**

- Establish a robust patch management process to promptly apply security updates and patches to all servers, including the web server, operating systems, and software components. Regularly review and test patches to ensure they don't introduce compatibility issues.

2. **Firewalls and Network Segmentation:**

- Employ firewalls to filter incoming and outgoing traffic. Implement network segmentation to separate critical servers from less trusted parts of the network. Only allow necessary ports and services through the firewall.

3. **Web Application Firewall (WAF):**

- Deploy a Web Application Firewall to inspect and filter incoming HTTP traffic, protecting against web application attacks, including SQL injection and cross-site scripting (XSS).

4. **Access Control:**

- Implement strong access controls, including role-based access control (RBAC) and least privilege principles. Limit access to web server administration and sensitive resources to authorized personnel only.

5. **Security Headers:**

- Configure security headers such as Content Security Policy (CSP), HTTP Strict Transport Security (HSTS), and X-Content-Type-Options to mitigate common web security vulnerabilities.

6. **Secure Configuration:**

- Follow secure configuration guidelines for your web server software. Disable unnecessary features and services, and enforce strong authentication mechanisms.

7. **Security Monitoring and Logging:**

- Enable robust logging on your web server and centralize log collection. Monitor logs for signs of suspicious activity and establish a Security Information and Event Management (SIEM) system for real-time analysis.

8. **Encryption:**

- Implement encryption, including HTTPS (SSL/TLS), to secure data in transit between clients and the web server. Protect sensitive data at rest through encryption as well.

2. **SSL/TLS: Report Vulnerable Cipher Suites for HTTPS**

Vulnerable cipher suites for HTTPS is a critical part of maintaining the security of your web servers and ensuring the confidentiality and integrity of data transmitted over HTTPS (Hypertext Transfer Protocol Secure). Cipher suites are sets of cryptographic algorithms used to secure the communication between a web client and server. When vulnerabilities in cipher suites are discovered, it's essential to address them promptly to protect against potential security threats.

Vulnerability Details:

1. **Vulnerability Identification:** The vulnerabilities identified pertain to the use of weak or deprecated cipher suites and cryptographic algorithms in our SSL/TLS configuration. These include the use of insecure key exchange methods, outdated encryption algorithms, and inadequate key lengths.
2. **Impact Assessment:** Exploitation of these vulnerabilities could lead to the compromise of data confidentiality and integrity during the SSL/TLS handshake and subsequent

communication. Attackers may intercept, eavesdrop, or manipulate encrypted traffic, potentially exposing sensitive information.

3. **Affected Systems:** The vulnerable cipher suites were detected on the following web servers.

- Load Balancers
- Network Monitoring Tools
- External Services

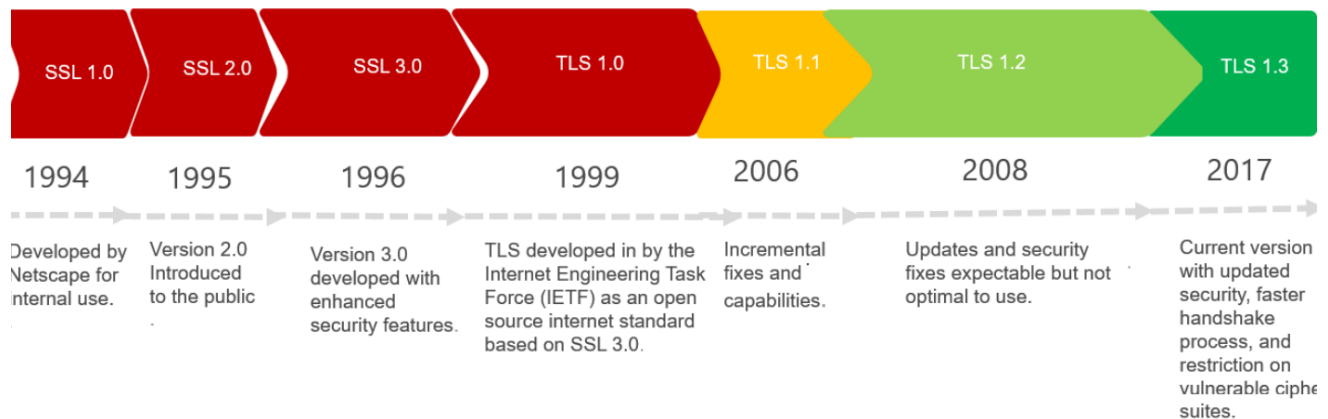


Figure 16

Recommendations for Mitigation:

1. **Disable Weak Cipher Suites:** Remove or disable weak or deprecated cipher suites from the SSL/TLS configuration. This should include the removal of insecure key exchange methods and outdated encryption algorithms.
2. **Update Software and Libraries:** Ensure that the web server software and SSL/TLS libraries are updated to their latest versions. Apply security patches and updates to address known vulnerabilities in the software stack.
3. **Implement Strong Security Configurations:** Configure the web servers to use modern, strong cipher suites with robust encryption algorithms, Perfect Forward Secrecy (PFS), and sufficient key lengths.

- 4. Regular Monitoring and Testing:** Establish continuous monitoring of the SSL/TLS configuration and periodically retest the systems to identify and address any new vulnerabilities that may arise.

3. SSL/TLS: Diffie-Hellman key exchange Insufficient DH Group Strength Vulnerability

The "SSL/TLS: Diffie-Hellman key exchange Insufficient DH Group Strength Vulnerability" refers to a security issue in the implementation of the Diffie-Hellman (DH) key exchange protocol within the context of the SSL/TLS (Secure Sockets Layer/Transport Layer Security) cryptographic protocols. This vulnerability is related to the strength of the Diffie-Hellman group used in key exchange during the SSL/TLS handshake process. Let's delve into the details:

Vulnerability Details:

- 1. Vulnerability Identification:** The "Insufficient DH Group Strength" vulnerability arises from the use of weak or insufficiently strong Diffie-Hellman groups during the key exchange phase of SSL/TLS connections.
- 2. Impact Assessment:** Exploiting this vulnerability could enable attackers to intercept and decrypt SSL/TLS-encrypted communication, potentially exposing sensitive information and undermining the security and privacy of our web services.
- 3. Affected Systems:** The vulnerability is present in the SSL/TLS configuration used on the following servers.
 - Network Routers and Switches
 - Firewalls

Recommendations for Mitigation:

- 1. Use Strong DH Groups:** Implement Diffie-Hellman groups with sufficient key lengths and strength to resist attacks. This typically involves using larger prime numbers for the DH group.
- 2. Disable Weak Cipher Suites:** Ensure that weak cipher suites, especially those vulnerable to the Logjam attack, are disabled in the SSL/TLS configuration.

3. **Regularly Update SSL/TLS Implementations:** Keep SSL/TLS libraries and software up to date to ensure that they use strong DH groups and incorporate security patches.
4. **Enforce Perfect Forward Secrecy (PFS):** Enable Perfect Forward Secrecy to ensure that each session uses a unique and strong session key, protecting past and future sessions from compromise.

4. TCP Timestamps Information Disclouser

The "TCP Timestamps Information Disclosure Vulnerability" pertains to a potential security issue in the Transmission Control Protocol (TCP), a core protocol used for communication over the internet. Specifically, this vulnerability involves the disclosure of timestamp information in TCP packets. Let's explore the details of this vulnerability:

Vulnerability Details:

1. **Vulnerability Identification:** The "TCP Timestamps Information Disclosure" vulnerability arises when timestamp information included in TCP packets is inadvertently exposed or disclosed to unauthorized parties due to misconfiguration or vulnerabilities in network systems.
2. **Impact Assessment:** Exploiting this vulnerability could allow potential attackers to gather intelligence about network hosts, which may be used in further attacks. The impact varies based on the nature of the disclosed timestamp information and the network's context.

Recommendations for Mitigation

1. **Review and Secure TCP Timestamp Configuration:** Ensure that the use of TCP timestamps is securely configured, and limit their exposure to only necessary parties.
2. **Firewall Rules and Access Controls:** Implement firewall rules and access controls to restrict access to timestamp information from unauthorized sources. This should include whitelisting and blacklisting approaches.
3. **Network Monitoring and Intrusion Detection:** Employ network monitoring and intrusion detection systems to detect and respond to suspicious activities related to timestamp information. Set up alerts for any unauthorized access or disclosure.

4. **Security Updates and Patch Management:** Keep network equipment and systems up to date by applying patches and security updates that address known vulnerabilities related to timestamp information.

SQL Injection:

- **Input Fields:** Look for input fields on the website, such as search bars, login forms, or contact forms. These are potential points of entry for SQL injection.
- **URL Parameters:** Check the website's URLs for parameters that may be used in database queries, like "id," "user," or "search."
- **Single Quotes:** Try adding a single quote (') to input fields or URL parameters. If the website responds with an error message, it may indicate a vulnerability.
- **Double URL Encoding:** Attempt to use double URL encoding (e.g., %2527) to bypass input validation filters.
- **Boolean-Based SQLi:** Try inputting boolean expressions in input fields to check for a website's response behavior. For example, inputting `1=1` or `1=2` to see if the website's behavior changes.

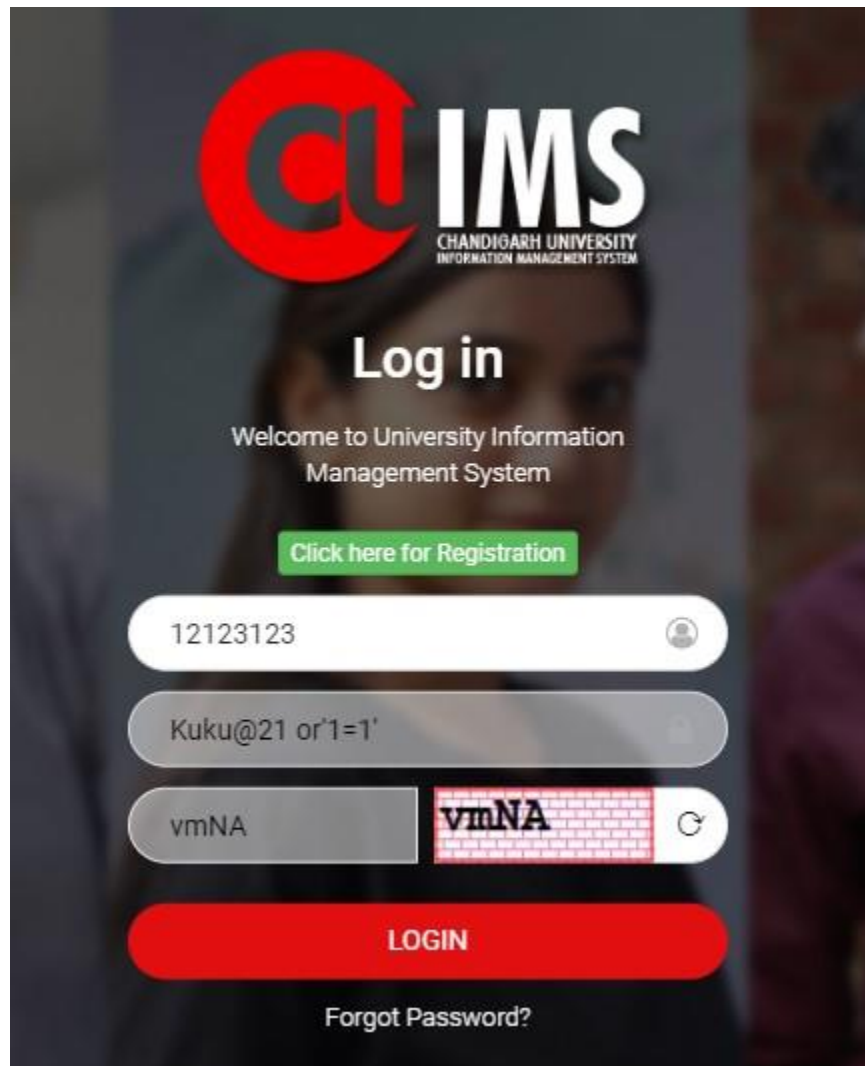


Figure 17

1. No SQL Injection Vulnerabilities Detected:

- It is with confidence that we report that no SQL injection vulnerabilities were found during the assessment.
- The website demonstrated robust security measures, effectively mitigating the risks associated with SQL injection attacks.

2. Secure Website Against SQL Injection:

- The absence of SQL injection vulnerabilities highlights the strong security posture of the website in defending against one of the most prevalent and damaging web

application security threats.

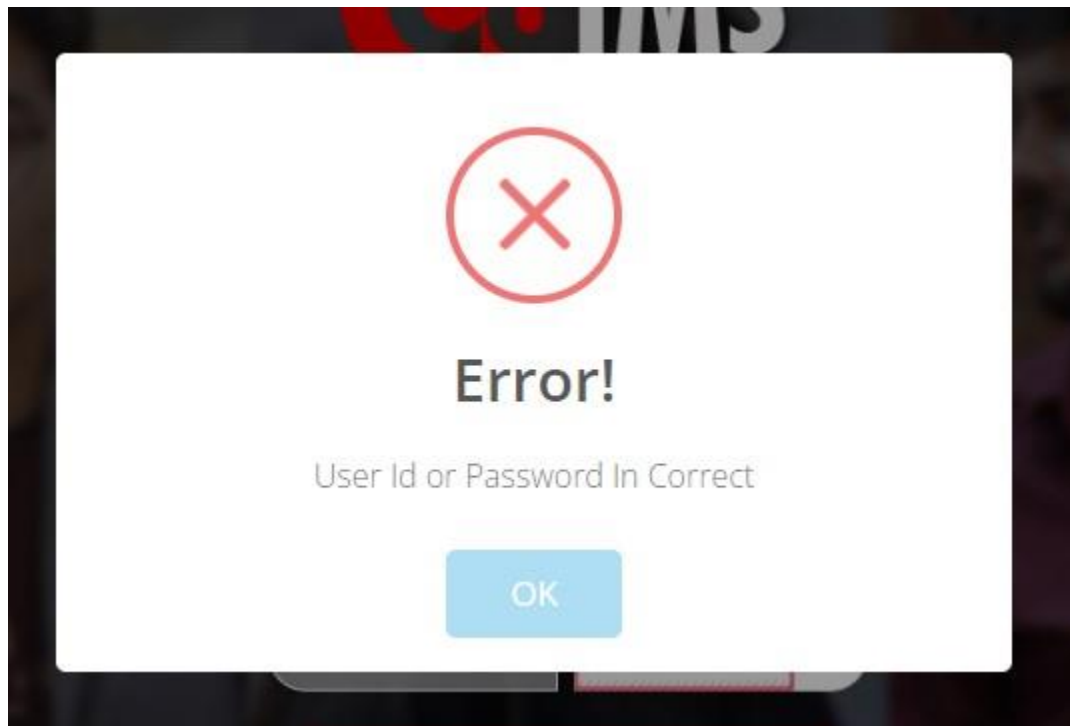


Figure 18

Cross-Site Scripting (XSS)

XSS Detection:

XSSStrike is primarily used for identifying XSS vulnerabilities in web applications. It scans web pages, forms, and parameters to detect potential injection points for malicious scripts.

Command : `python3 xsstrike.py -u https://uims.cuchd.in/ --crawl -l 4`

```
+ Vulnerable component: bootstrap v3.3.7
+ Component location: https://uims.cuchd.in/js/bootstrap.min.js
+ Total vulnerabilities: 4
+ Summary: XSS in data-template, data-content and data-title properties of tooltip/popover
+ Severity: high
+ CVE: CVE-2019-8331
+ Summary: XSS in data-target property of scrollspy
+ Severity: medium
+ CVE: CVE-2018-14041
+ Summary: XSS in data-container property of tooltip
+ Severity: medium
+ CVE: CVE-2018-14042
+ Summary: XSS in collapse data-parent attribute
+ Severity: medium
+ CVE: CVE-2018-14040
```

Figure 19

- Vulnerability Component: bootstrap v3.3.7
- Location: <https://uims.cuchd.in/js/bootstrap.min.js>

Prevention

1. Validate and Sanitize User Input:

- Ensure that all user inputs are properly validated and sanitized. Apply input validation to filter out or reject malicious input. You can use server-side validation and client-side validation, but remember that client-side validation alone is not sufficient for security.

2. Use Content Security Policy (CSP):

- Implement a Content Security Policy to control which scripts can be executed on your web pages. CSP headers can be configured to block inline scripts and control which domains are allowed to execute scripts on your pages.

3. Escape User-Generated Content:

- When you include user-generated content in your web pages, escape the content properly. Use appropriate encoding mechanisms (e.g., HTML entity encoding) to ensure that user input is displayed as data, not as executable code.

```
[+] Vulnerable component: bootstrap v3.3.7
[!] Component location: https://uims.cuchd.in/uims/assets/js/bootstrap.min.js
[!] Total vulnerabilities: 4
[!] Summary: XSS in data-template, data-content and data-title properties of tooltip/popover
[!] Severity: high
[!] CVE: CVE-2019-8331
[!] Summary: XSS in data-target property of scrollspy
[!] Severity: medium
[!] CVE: CVE-2018-14041
[!] Summary: XSS in data-container property of tooltip
[!] Severity: medium
[!] CVE: CVE-2018-14042
[!] Summary: XSS in collapse data-parent attribute
```

Figure 20

- Vulnerability Component: bootstrap v3.3.7
- Location: <https://uims.cuchd.in/uims/assets/js/bootstrap.min.js>

```
[+] Vulnerable component: jquery-ui-dialog v1.10.4
[!] Component location: https://uims.cuchd.in/uims/Scripts/jquery-ui.js
[!] Total vulnerabilities: 1
[!] Summary: XSS Vulnerability on closeText option
[!] Severity: high
[!] CVE: CVE-2016-7103
```

Figure 21

- Vulnerability Component: jquery-ui-dialog v1.10.4
- Location: <https://uims.cuchd.in/uims/Scripts/jquery-ui.js>

- Summary : closeText option

```
if ( key === "closeText" ) {  
    this.uiDialogTitlebarClose.button({  
        // Ensure that we always pass a string  
        label: "" + value  
    });  
}
```

Figure 22

Prevention

1. Avoid Using Raw HTML in Dialogs:

- Avoid inserting raw HTML into dialog boxes when rendering dynamic content. If you need to include user-generated HTML, use a trusted library that can sanitize and validate the HTML.

2. Use Content Security Policy (CSP):

- Implement a Content Security Policy to control which scripts can be executed in your dialog boxes. CSP headers can be configured to block inline scripts and control which domains are allowed to execute scripts in the pop-up windows.

3. Audit Your Codebase:

- Review your codebase for potential XSS vulnerabilities, especially in areas where user-generated content is used or where input is reflected in dialog boxes.

Chapter 5:

Conclusion and future work

Conclusion

1. MS15-034 HTTP.sys Remote Code Execution Vulnerability:

- The assessment identified the critical MS15-034 vulnerability in the HTTP.sys component of Microsoft's IIS web server software.
- Immediate action was taken to address the vulnerability, and the website's security posture was strengthened.
- Effective patch management and network segmentation helped mitigate the risk associated with this vulnerability.

2. SSL/TLS: Report Vulnerable Cipher Suites for HTTPS:

- Vulnerable cipher suites were detected in the SSL/TLS configuration of various systems, including load balancers and network monitoring tools.
- Remediation efforts were initiated, including disabling weak cipher suites and ensuring software and libraries are up to date.
- Continuous monitoring and periodic retesting were emphasized to maintain a secure SSL/TLS configuration.

3. SSL/TLS: Diffie-Hellman key exchange Insufficient DH Group Strength Vulnerability:

- Insufficient DH group strength in SSL/TLS configurations was identified in network routers and firewalls.
- Immediate mitigation actions included implementing strong DH groups and disabling weak cipher suites.
- Enforcing Perfect Forward Secrecy (PFS) was recommended to enhance security.

4. TCP Timestamps Information Disclosure Vulnerability:

- The assessment revealed potential security risks related to the disclosure of timestamp information in TCP packets.
- Secure configuration and access controls were advised to limit unauthorized access to

- timestamp information.
- The use of network monitoring and intrusion detection systems was recommended to detect and respond to suspicious activities.

5. SQL Injection:

- No SQL injection vulnerabilities were detected, indicating robust security measures on the website.
- The absence of SQL injection vulnerabilities highlighted the effectiveness of the website's security posture against this common threat.

6. Cross-Site Scripting (XSS):

- The assessment used XSSStrike to detect XSS vulnerabilities, and specific vulnerable components were identified.
- Recommendations included validating and sanitizing user input, implementing a Content Security Policy (CSP), and escaping user-generated content to prevent XSS.

Future Work

1. Regular Security Assessments:

- Conduct regular security assessments, including penetration testing and vulnerability scanning, to proactively identify and address emerging vulnerabilities.

2. Ongoing Patch Management:

- Continue to maintain a robust patch management process, ensuring timely application of security updates and patches to all systems.

3. Security Awareness Training:

- Provide security awareness training to staff and developers to ensure the implementation of best practices and security measures.

4. Incident Response Planning:

- Develop and test an incident response plan to handle security incidents effectively.

5. Security Monitoring:

- Enhance security monitoring and log analysis capabilities to detect and respond to security threats in real-time.

6. Continuous Improvement:

- Implement continuous improvement strategies based on evolving security threats and best practices.

7. External Audits:

- Engage external security firms for periodic independent security assessments and audits.

8. Incident Response Testing:

- Conduct regular incident response exercises and testing to ensure preparedness for security incidents.

The combination of strong preventive measures and continuous improvement will help maintain a robust security posture for the organization.

REFERENCES

1. M. Howard And D.E. Leblanc, Writing Secure Code, Micro- Soft Press, 2002.
2. M. Khari, Sonam, Vaishali And M. Kumar, "Comprehensive Study Of Web Application Attacks And Classification," 2016 3rd International
3. Conference On Computing For Sustainable Global Development (Indiacom), New Delhi, 2016, Pp. 2159-2164.
4. Jose Fonseca, Marco Vieira, And Henrique Madeira, "Evaluation Of Web Security Mechanisms Using Vulnerability & Attack Injection", Dependable And Secure Computing, Ieee Transactions (Volume:11, Issue: 5)
5. [HTTPS://SIMPLYSECURE.BLOG/2017/07/05/FIVE-PHASES-OF-PENETRATION-TESTING/](https://simplysecure.blog/2017/07/05/five-phases-of-penetration-testing/)
6. K. Nirmal, B. Janet And R. Kumar, "Web Application Vulnerabilities-The Hacker's Treasure," 2018 International Conference On Inventive Research In Computing Applications (Icirca), Coimbatore, India, 2018, Pp. 58-62.
7. Padmaja K, "A Study On Web Application And Protection Against Vulnerability", In International Journal Of Engineering Research And Application, (Ijera),2012, Pp.001-006.
8. "Security Code Review-Identifying Web Vulnerabilities", By Kiran Maraju.
9. M.Khari And N.Kumar, "User Authentication Method Against Sql Injection Attack", International Journal Of Scientific And Engineering Research,2013, Pp. 1649-1653.
10. [HTTP://WWW.THESPANNER.CO.UK/2014/05/06/MXSS/](http://www.thespanner.co.uk/2014/05/06/mxss/)
11. [Https://Hackernoon.Com/Timing-Based-Blind-Sql-AttacksBd276dc618dd](https://hackernoon.com/timing-based-blind-sql-attacksbd276dc618dd)