

## ***Function And performance Testing***

### ***Top 5 Vulnerability Exploitation***

<b>S.no</b>	<b>Vulnerability Name</b>	<b>CWE-No</b>
<b>1</b>	<b>XML External Entities (XXE)</b>	<b>CWE-611</b>
<b>2</b>	<b>Buffer Overflow</b>	<b>CWE-120</b>
<b>3</b>	<b>Remote Code Execution (RCE)</b>	<b>CWE-94</b>
<b>4</b>	<b>Insecure API Exposure</b>	<b>CWE-201</b>
<b>5</b>	<b>Insufficient Logging &amp; Monitoring</b>	<b>CWE-778</b>

### ***3.2 Solution Requirement***

***Vulnerability Name :XML External Entities (XXE)***

**CWE-611**

**OWASP Category: Injection**

**Description:**

Improper XML parsing allows attackers to access local files or internal networks.

**Business Impact:**

- **Data Exposure:** Attackers can exploit XXE to retrieve sensitive data, such as system files, configuration files, or even credentials stored on the server.
- **Denial of Service (DoS):** XXE attacks can exhaust system resources, causing applications to crash or become unresponsive.
- **Server-Side Request Forgery (SSRF):** Attackers can use XXE to send requests to internal services, bypassing security controls and accessing restricted areas.
- **File Disclosure:** XXE can be used to read arbitrary files on the server, leading to exposure of confidential business data.
- **Remote Code Execution (RCE):** In advanced cases, XXE can be exploited to execute commands on the server, leading to full system compromise.
- **Regulatory Non-Compliance:** Exposure of sensitive customer data through XXE attacks can lead to violations of compliance standards like GDPR, HIPAA, and PCI DSS, resulting in hefty fines and reputational damage.

**Steps to Identify:**

- Use Burp Suite with XXE payloads.
- Analyze XML-based API responses.
- Validate and sanitize XML input.
- Disable external entity processing in XML parsers.

**Vulnerability Name:** Buffer Overflow

**CWE-611**

## **OWASP Category: Memory Corruption**

### **Description:**

**Buffer Overflow** occurs when a program writes more data to a buffer than it can hold, causing adjacent memory to be overwritten. Attackers exploit this to execute arbitrary code, leading to full system compromise.

### **Business Impact:**

- **System Crashes:** Exploiting buffer overflow can cause applications to crash, leading to service disruptions.
- **Remote Code Execution (RCE):** Attackers can inject and execute malicious code, gaining control over the system.
- **Privilege Escalation:** Buffer overflow vulnerabilities may allow attackers to escalate privileges, gaining higher-level access to systems.
- **Data Corruption:** Overwriting memory can lead to data corruption or loss, affecting system integrity.
- **Financial and Reputational Damage:** Organizations may suffer financial loss due to service outages, regulatory fines, and reputational damage.

### **Steps to Identify:**

- Use fuzzing techniques to inject large inputs and monitor system behavior.
- Utilize static analysis tools to detect improper memory management.
- Review source code for unsafe functions like strcpy, gets, and sprintf.
- Enable memory protection mechanisms such as Address Space Layout Randomization (ASLR) and Data Execution Prevention (DEP).

**Vulnerability Name:** Remote Code Execution (RCE)

## **CWE-94**

**OWASP Category: Injection**

### **Business Impact:**

- **Complete System Takeover:** Attackers can gain full control over the affected system.
- **Data Theft and Exfiltration:** Sensitive information, including personal and financial data, can be stolen.
- **Propagation of Malware:** Attackers can install malware, including ransomware, to disrupt business operations.
- **Financial Losses:** Downtime, ransom payments, and regulatory fines can cause severe financial damage.
- **Reputational Damage:** Loss of customer trust and potential legal consequences.

### **Steps to Identify:**

- **Conduct code reviews** to identify insecure function calls like `eval()` and `exec()`.
- **Use dynamic analysis tools** to test application behavior against payloads.
- **Check for improper input validation** in APIs and user input fields.
- **Perform penetration testing** to simulate real-world RCE attacks.

***Vulnerability Name:*** Insecure API Exposure

## **CWE-201**

- **OWASP Category:** API Security

### **Description:**

**Insecure API exposure occurs when APIs expose excessive or sensitive data due to improper access controls, weak authentication, or poor data filtering.**

**Business Impact:**

- **Data Breach:** Sensitive customer and business data can be accessed by unauthorized parties.
- **Regulatory Penalties:** Non-compliance with data protection laws (e.g., GDPR, CCPA) can result in fines.
- **Service Disruption:** Attackers may exploit API weaknesses to overload or disrupt services.
- **Increased Attack Surface:** APIs are often publicly accessible, making them prime targets for attackers.

**Steps to Identify:**

- Use API security testing tools like Postman, Burp Suite, or OWASP API Security Testing Tool.
- Review API responses for excessive or unnecessary data exposure.
- Test authentication and authorization mechanisms for weaknesses.
- Perform penetration testing on exposed API endpoints.

***Vulnerability Name:*** Insufficient Logging & Monitoring

- **CWE:** CWE-778
- **OWASP Category:** Security Logging & Monitoring Failures

**Description:**

Lack of proper logging and monitoring allows attacks to go undetected, making it difficult to identify security incidents, investigate breaches, and take corrective action in a timely manner.

**Business Impact:**

- **Delayed Incident Response:** Without adequate logs, security teams struggle to detect and respond to threats promptly.

- **Regulatory Non-Compliance:** Many data protection regulations (e.g., GDPR, HIPAA) require proper logging. Non-compliance can lead to hefty fines.
- **Undetected Data Breaches:** Attackers can exploit systems for prolonged periods without detection, increasing data theft risks.
- **Increased Attack Dwell Time:** The longer an attacker remains undetected, the more damage they can inflict on an organization's infrastructure.
- **Forensic Challenges:** Insufficient logs hinder the ability to investigate security incidents, making it harder to determine the cause and scope of an attack.
- **Reputational Damage:** Customers and partners may lose trust if a security incident remains undetected for a long time due to inadequate monitoring.

#### **Steps to Identify:**

- **Review Log Retention Policies:** Ensure logs are stored securely and retained for an appropriate duration.
- **Check for Security Alerts in SIEM Systems:** Utilize Security Information and Event Management (SIEM) tools to aggregate and analyze security events.
- **Test for Missing or Incomplete Logs:** Conduct audits to verify whether important security events (e.g., login failures, privilege escalations, API requests) are being recorded.
- **Implement Real-Time Monitoring:** Use automated alerting mechanisms to detect unusual system behavior.
- **Analyze Past Security Incidents:** Evaluate previous breaches to determine if insufficient logging contributed to delayed detection.

### **3.3 Technology Stack**

### **3.3.1 Tools explored in this project**

#### **1. Endpoint Security Tools:**

- **Antivirus/Antimalware Software:** Detects, prevents, and removes malicious software from individual devices.
- **Host-Based Intrusion Detection Systems (HIDS):** Monitors activities on individual devices for signs of unauthorized access or malicious behavior.
- **Endpoint Detection and Response (EDR):** Provides continuous monitoring, detection, and response capabilities on endpoints to combat advanced threats.

#### **2. Network Security Tools:**

- **Firewalls:** Hardware or software-based security devices that monitor and control incoming and outgoing network traffic based on predetermined security rules.
- **Intrusion Detection Systems (IDS) & Intrusion Prevention Systems (IPS):** Monitor network traffic for suspicious activities and take action to block or prevent potential threats.
- **Network Protocol Analyzers (e.g., Wireshark):** Used for troubleshooting, analysis, and security auditing of network traffic.

#### **3. Vulnerability Assessment Tools:**

- **Vulnerability Scanners (e.g., Nessus, OpenVAS):** Identify vulnerabilities, misconfigurations, and malware across various platforms.
- **Cloud-Based Assessment Solutions (e.g., Qualys):** Offer vulnerability management and assessment to identify, prioritize, and remediate vulnerabilities.

#### **4. Encryption Tools:**

- **Data Encryption Programs (e.g., PGP):** Provide cryptographic privacy and authentication for data communication.

- **Full-Disk Encryption Features (e.g., BitLocker):** Protect data on disk volumes.
- **SSL/TLS Toolkits (e.g., OpenSSL):** Implement protocols for securing communications over a network.

#### **5. Identity and Access Management (IAM) Tools:**

- **Directory Services (e.g., Active Directory):** Manage user identities and permissions in a networked environment.
- **Single Sign-On (SSO) Solutions:** Enable users to securely authenticate once and access multiple applications without re-entering credentials.
- **Multi-Factor Authentication (MFA):** Enhances security by requiring users to provide multiple forms of authentication, such as passwords, biometrics, or tokens.

#### **6. Security Information & Event Management (SIEM) Tools:**

- **SIEM Platforms (e.g., Splunk, LogRhythm, IBM QRadar):** Collect, analyze, and correlate security event data from various sources to provide real-time insights into potential security threats and incidents.

### **4 Project Design:**

#### **Stage 2:**

##### **4.1 Nessus:**

**Nessus:** Nessus is a powerful vulnerability assessment tool developed by Tenable, widely used by security professionals to detect vulnerabilities, misconfigurations, and compliance issues in IT systems. It helps organizations proactively identify security risks and remediate them before they can be exploited by attackers.

One of the key strengths of Nessus is its comprehensive vulnerability scanning capabilities, which allow organizations to proactively detect security flaws before they can be exploited by attackers. The tool uses an extensive database of over 180,000 plugins, regularly updated to identify new vulnerabilities, misconfigurations, and outdated software. Nessus scans devices for open ports,



unpatched software, weak passwords, and dangerous configurations that could lead to security breaches. It also detects malware, backdoors, botnet activity, and ransomware-related vulnerabilities, ensuring that security teams can take immediate action to mitigate risks. In addition to standard vulnerability scanning, Nessus provides compliance auditing to help organizations adhere to regulatory standards such as PCI-DSS, HIPAA, ISO 27001, NIST, and CIS benchmarks. This makes it an essential tool for companies that must meet strict security requirements.

While Nessus is highly effective, it does have certain limitations that security professionals should be aware of. Like many automated scanning tools, it can sometimes produce false positives, requiring manual verification of certain findings. Additionally, Nessus does not automatically remediate vulnerabilities—it provides detailed reports and recommendations, but fixing the issues requires manual intervention by IT teams. Another challenge is that large-scale scans can consume significant system resources, which may impact network performance if not properly configured. Despite these challenges, Nessus remains one of the most trusted tools in vulnerability management due to its accuracy, reliability, and continuous updates to stay ahead of emerging threats.

## **4.2 Proposed Solution Testing and Findings;**

### **Testing the Website Solution**

#### **Types of Tests to Conduct:**

- 1.

#### **Functionality Testing:**

- 2.

1. Goal: Ensure all website features work as intended.
2. Tests might include:
  1. Testing contact forms, search features, and navigation links.

2. **Verifying user registration/login processes.**
3. **Checking that all interactive elements (buttons, drop-downs, etc.) function correctly.**

3.

### **Usability Testing:**

4.

1. **Goal: Evaluate the user experience (UX) and ease of use.**
2. **Tests might include:**
  1. **Gathering user feedback through surveys or usability sessions.**
  2. **Testing website navigation, content accessibility, and the clarity of calls to action (CTAs).**
  3. **Checking whether users can find the information they need quickly and easily.**

5.

### **Performance Testing:**

6.

1. **Goal: Assess the website's load time and performance under various conditions.**
2. **Tests might include:**
  1. **Load testing (how the site performs under different traffic loads).**
  2. **Stress testing (how it behaves when pushed beyond expected traffic levels).**
  3. **Checking page load times across various devices and networks.**
  4. **Testing for smooth multimedia (e.g., video or images) loading.**

7.

## **Compatibility Testing:**

**8.**

- 1. Goal: Ensure the website functions across different browsers, devices, and screen sizes.**
- 2. Tests might include:**
  - 1. Verifying compatibility across browsers (Chrome, Firefox, Safari, Edge, etc.).**
  - 2. Testing responsiveness on desktop, tablet, and mobile devices.**
  - 3. Checking for any broken elements on different screen sizes and resolutions.**

**9.**

## **Security Testing:**

**10.**

- 1. Goal: Identify vulnerabilities and ensure user data protection.**
- 2. Tests might include:**
  - 1. Penetration testing to detect vulnerabilities.**
  - 2. Testing encryption for sensitive user data.**
  - 3. Verifying protection against SQL injection and cross-site scripting (XSS).**

**11.**

## **SEO (Search Engine Optimization) Testing:**

**12.**

- 1. Goal: Ensure the website is optimized for search engines.**
- 2. Tests might include:**
  - 1. Checking the use of meta tags, alt text, and other SEO best practices.**

2. **Verifying that the website has a proper URL structure and sitemap.**
3. **Testing for broken links and redirects.**

## **2. Findings:**

**Once you've completed these tests, you will likely uncover various findings based on the testing results. Some possible findings include:**

- 

### **Functionality Issues:**

- 

- **Broken links or non-functioning features (e.g., forms not submitting).**
- **Errors in user flows (e.g., users are unable to complete a task such as purchasing or signing up).**

- 

### **Usability Concerns:**

- 

- **Users may struggle with navigation due to confusing menus or cluttered pages.**
- **Calls to action might be unclear or too far down the page.**
- **Some sections of the website could be difficult to read or use on smaller mobile screens.**

- 

### **Performance Problems:**

- 

- **Slow load times, especially for image-heavy pages.**
- **Server response issues under heavy traffic (leading to timeouts or errors).**
- **Mobile performance might lag due to unoptimized assets.**

- 

### **Compatibility Issues:**

- 

- Some website features or layouts might not render correctly in certain browsers.
- Mobile responsiveness may break on specific devices, such as larger tablets or smaller smartphones.
- Certain interactive elements may not function on older browser versions.

- 

### **Security Vulnerabilities:**

- 

- Lack of proper encryption for sensitive data such as passwords and payment details.
- Potential exposure to common cyber-attacks, such as SQL injection or cross-site scripting (XSS).
- Missing security certificates or outdated software that could make the site vulnerable.

- 

### **SEO Findings:**

- 

- Missing alt text for images, hindering search engine indexing.
- Poor keyword optimization, leading to low search engine rankings.
- Broken or redirected links that can negatively affect SEO.

### **3. Next Steps Based on Findings:**

Once you've gathered your testing findings, you'll want to address any issues found. Here's a general approach:

- **Prioritize Issues:** Address critical issues (e.g., security vulnerabilities, major functionality bugs) first.
- **Fix Usability Issues:** Simplify navigation, improve the clarity of CTAs, and optimize content for mobile devices.
- **Optimize Performance:** Compress images, reduce the number of HTTP requests, and implement content delivery networks (CDNs).
- **Ensure Browser Compatibility:** Implement fixes for any rendering issues on different browsers and devices.
- **Improve SEO:** Address any SEO issues by fixing broken links, optimizing metadata, and improving content structure.

#### **4.4 Understanding about the project: Exploring Cyber Security:** **Understanding Threats and Solutions in the Digital Age**

##### **Project Overview: "Exploring Cybersecurity: Understanding Threats and Solutions in the Digital Age"**

In the modern digital world, cybersecurity has become more critical than ever. With the rise of digital platforms and interconnected systems, cyber threats continue to evolve, posing risks to individuals, businesses, and governments. This project aims to explore the nature of these threats and the various solutions available to counteract them.

##### **Objectives of the Project:**

1.

##### **Understanding Cybersecurity Threats:**

2.

1. Identify different types of cybersecurity threats that exist today.
2. Understand how these threats impact organizations and individuals.

3. Explore the tools and tactics employed by cybercriminals to exploit vulnerabilities.

3.

#### **Understanding Cybersecurity Solutions:**

4.

1. Analyze the best practices for protecting against cyber threats.
2. Review modern cybersecurity solutions (e.g., firewalls, encryption, multi-factor authentication).
3. Understand the role of both individuals and organizations in maintaining cybersecurity.

5.

#### **Assessing the Current Landscape:**

6.

1. Investigate how current cybersecurity measures are evolving to address new and emerging threats.
2. Understand the ongoing challenges faced by cybersecurity professionals in keeping up with rapidly changing technology.

### **5. Project Planning and Scheduling :**

#### **5.1 Project Planning:**

**Product backlog, Sprint Schedule, and Estimation**

**Use the below template to create product backlog and sprint schedule.**

<b>Sprint</b>	<b>Functional Requirement (Epic)</b>	<b>User Story Number</b>	<b>User Story / Task</b>	<b>Story Points</b>	<b>Priority</b>	<b>Team Members</b>
<b>Sprint-1</b>	<b>Data Collection</b>	<b>USN-1</b>	<b>Collect data from various cybersecurity websites like( Krebs on security,Info Security Magazine etc).</b>	<b>5</b>	<b>High</b>	<b>Vasanth,V Kumar,Ra Deepak,Pa Kumar</b>
<b>Sprint-1</b>		<b>USN-2</b>	<b>Use Real Time APIs to gather data.</b>	<b>3</b>	<b>Medium</b>	<b>Vasanth,V Kumar,Ra Deepak,Pa Kumar</b>
<b>Sprint-2</b>		<b>USN-3</b>	<b>Get various news about the different kinds of cybersecurity vulnerabilities like (XSS,RCE etc).</b>	<b>2</b>	<b>Low</b>	<b>Vasanth,V Kumar,Ra Deepak,Pa Kumar</b>
<b>Sprint-2</b>	<b>Processing</b>	<b>USN-4</b>	<b>Use of data processing platforms like (Apache Storm,SIEM etc).</b>	<b>5</b>	<b>High</b>	<b>Vasanth,V Kumar,Ra Deepak,Pa Kumar</b>
<b>Sprint-2</b>		<b>USN-5</b>	<b>Use of cybersecurity libraries like(scapy,cryptography etc) to work on the given data.</b>	<b>4</b>	<b>High</b>	<b>Vasanth,V Kumar,Ra Deepak,Pa Kumar</b>



<b>Sprint-3</b>	<b>User Interface</b>	<b>USN-6</b>	<b>Use of various coding languages like (Ruby ,Assembly language) and React.js helps to create a simple yet effective dashboard for the user.</b>	<b>5</b>	<b>High</b>	<b>Vasanth,V Kumar,Ra Deepak,Pa Kumar</b>
<b>Sprint-3</b>		<b>USN-7</b>	<b>Having a separate login implemented for users to see dashboard particular to their content .</b>	<b>3</b>	<b>Medium</b>	<b>Vasanth,V Kumar,Ra Deepak,Pa Kumar</b>
<b>Sprint-3</b>	<b>Data Visualization</b>	<b>USN-8</b>	<b>Use tools like DataDog,Loggly,QRadar etc to show various data in a more readable format to the user for easy to understand.</b>	<b>5</b>	<b>High</b>	<b>Vasanth,V Kumar,Ra Deepak,Pa Kumar</b>
<b>Sprint-4</b>		<b>USN-9</b>	<b>Have a feature to ask user for their suggestions the regarding thr given task.</b>	<b>2</b>	<b>Low</b>	<b>Vasanth,V Kumar,Ra Deepak,Pa Kumar</b>
<b>Sprint-4</b>	<b>Scalability</b>	<b>USN-10</b>	<b>Use Docker , Kubernetes to scale the whole project.</b>	<b>5</b>	<b>High</b>	<b>Vasanth,V Kumar,Ra Deepak,Pa Kumar</b>
<b>Sprint-4</b>		<b>USN-11</b>	<b>Have a better databse system to store the real time and other various data.</b>	<b>5</b>	<b>High</b>	<b>Vasanth,V Kumar,Ra Deepak,Pa Kumar</b>

## **5.2 Project Tracker, Velocity & Burndown Chart:**

<b>Sprint</b>	<b>Total Story Points</b>	<b>Duration</b>	<b>Sprint Start Date</b>	<b>Sprint End Date (Planned)</b>	<b>Story Points Completed (as on Planned End Date)</b>	<b>Sprint Release Date (Actual )</b>
<b>Sprint-1</b>	<b>12</b>	<b>6 Days</b>	<b>21 Jan 2025</b>	<b>26 Jan 2025</b>	<b>12</b>	<b>26 Jan 2025</b>
<b>Sprint-2</b>	<b>12</b>	<b>6 Days</b>	<b>28 Jan 2025</b>	<b>2 Feb 2025</b>	<b>08</b>	<b>3 Feb 2025</b>
<b>Sprint-3</b>	<b>12</b>	<b>6 Days</b>	<b>6 Feb 2025</b>	<b>11 Feb 2025</b>	<b>12</b>	<b>11 Feb 2025</b>
<b>Sprint-4</b>	<b>12</b>	<b>6 Days</b>	<b>14 Feb 2025</b>	<b>19 Feb 2025</b>	<b>10</b>	<b>20 Feb 2025</b>

### **Velocity:**

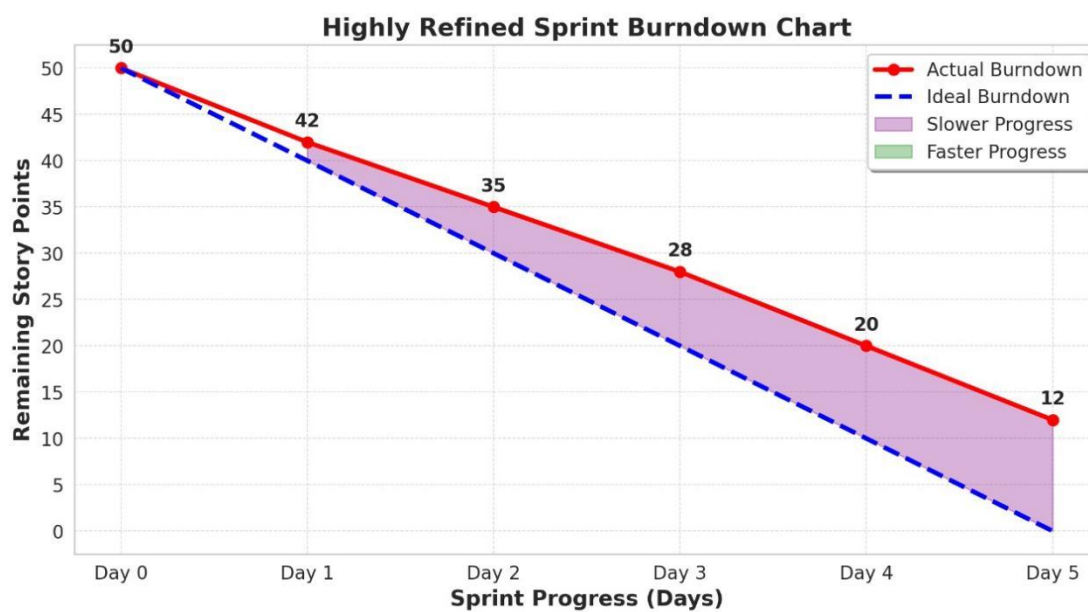
**Imagine we have a 10-day sprint duration and the velocity**

**Of the team is 20 (points per sprint). Let's calculate the teams average velocity (AV) per iteration unit (story points per day)**

**Average Velocity (AV)=Total Story Points / number of Sprints**

**=42/4 =10.5(approx.).**

### 5.2.1 The Sprint Burndown Chart:



**Red Line (Actual Burndown):** Represents the real progress of the sprint.

**Blue Dashed Line (Ideal Burndown):** Represents the expected progress if work is completed at a steady pace.

**Shaded Areas and Their Meanings:**

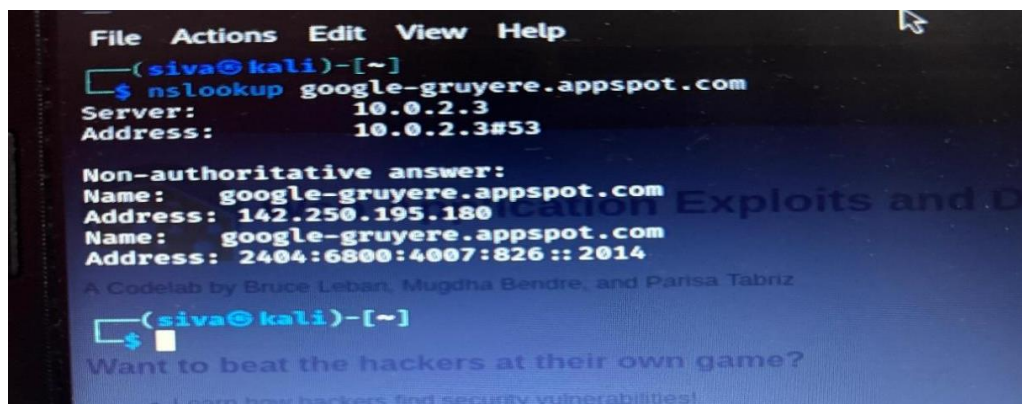
## Purple Shaded Area (Slower Progress)

- The space between the red line (actual progress) and the blue dashed line (ideal progress).

## Stage-2

**Targeted Website:Google-Gruyere**

**IP Address:142.250.195.180**



```
File Actions Edit View Help
(siva@kali)-[~]
$ nslookup google-gruyere.appspot.com
Server:      10.0.2.3
Address:     10.0.2.3#53

Non-authoritative answer:
Name:   google-gruyere.appspot.com
Address: 142.250.195.180
Name:   google-gruyere.appspot.com
Address: 2404:6800:4007:826::2014

A Codelab by Bruce Leban, Mugdha Bendre, and Parisa Tabriz

(siva@kali)-[~]
$
Want to beat the hackers at their own game?
• Learn how hackers find security vulnerabilities!
```

S.no	Vulnerability Name	CWE-No
1	Path Traversal	CWE-22
2	Insecure File Upload	CWE-434
3	Cross-Site Request Forgery (CSRF)	CWE-352
4	Clickjacking	CWE-1021

## 6.Functional and Performance Testing:

### 6.1: vulnerability Report(impacts and identifications):

***Vulnerability Name:*** Path Traversal

***CWE-No:*** CWE-22

- **OWASP Category:** A05:2021 - Security Misconfiguration
- **Severity:** High
- **Plug in :** Burp Suite path traversal, OWASp ZAP, NIKTO
- **PORT:** 80 for HTTP

***Description:***

Path Traversal, also known as Directory Traversal or Path Reversal, is a high-severity web security vulnerability that occurs when an application improperly processes user-supplied input, allowing unauthorized access to files and directories outside of the intended scope. This vulnerability arises when an application accepts user input for file path references and does not properly sanitize or restrict it, enabling attackers to navigate beyond designated directories using sequences like ../ (dot-dot-slash). If successfully exploited, path traversal can allow attackers to read sensitive system files, extract credentials, access source code, modify system configurations, or even execute arbitrary code, depending on the severity of the misconfiguration.

Web applications and systems that handle files dynamically, such as content management systems (CMS), file upload/download features, document viewers, and APIs, are highly susceptible to path traversal attacks. The vulnerability is especially dangerous when combined with weak file upload mechanisms or improper server configurations, as attackers could exploit it to execute remote code execution (RCE) and fully compromise the system

***Bussiness Impact:***

- **Data Breach & Confidentiality Loss**

- Attackers can access sensitive system files, user credentials, financial records, or proprietary business data.
- Exposure of customer PII (Personally Identifiable Information) can lead to identity theft, fraud, and loss of user trust.
- Regulatory Non-Compliance & Legal Consequences
  - Violates GDPR, HIPAA, PCI-DSS, and SOC 2 security standards, leading to heavy fines and legal penalties.
  - Organizations may be subject to lawsuits, class-action cases, or government-imposed restrictions on operations.

***Vulnerability Name:*** Insecure File Upload

***CWE-No:*** CWE-434

***OWASP Category:*** A05:2021 – Security Misconfiguration

- Severity: HIGH
- Plug in : ACUNETIX, METASPLOIT
- PORT: 80 for HTTP

**Description:**

Insecure File Upload vulnerabilities occur when an application improperly handles file uploads, allowing attackers to upload malicious files that can compromise the system. These vulnerabilities arise from inadequate validation of file metadata (e.g., filename, path) and content, leading to potential execution of malicious code on the server or client-side

***Business Impact:***

- Data Breaches and Confidentiality Loss  
Attackers can exploit these vulnerabilities to upload malicious files, leading to unauthorized access to sensitive data, including customer

information, financial records, and intellectual property. Such breaches can result in identity theft, fraud, and loss of competitive advantage.

- **Operational Disruptions**

Malicious file uploads can compromise system integrity, causing service outages or degraded performance. This disruption can hinder business operations, leading to productivity losses and potential revenue decline.

***Vulnerability Name:*** Cross-Site Request Forgery (CSRF)

***CWE-No:*** CWE-352

***OWASP Category:***

- **Severity:** Medium
- **Plug in :** Burp Suite CSRF Tester
- **PORT:** 80 for HTTP

**Description:** The consequences of such breaches are manifold. Financial losses can accrue from both immediate operational disruptions and long-term erosion of customer trust. Legal ramifications may ensue, especially if the breach involves sensitive customer information, leading to non-compliance with data protection regulations. Moreover, the organization's reputation can suffer irreparable harm, affecting customer retention and market position. Therefore, it is imperative for businesses to implement robust security measures, including stringent file validation protocols and continuous monitoring, to mitigate the risks associated with insecure file uploads.

***Bussiness Impact:***

**Unauthorized Transactions:** Attackers can initiate unauthorized actions, such as fund transfers or changes to account settings, leading to financial losses and operational disruptions.

**Data Integrity Compromise:** Malicious actors may modify or delete critical data, compromising the integrity of business information and affecting decision-making processes.

**Reputational Damage:** Security breaches resulting from CSRF attacks can erode customer trust and damage the organization's reputation, potentially leading to a loss of clientele and revenue.

***Vulnerability Name:*** Clickjacking

***CWE-No:*** CWE-1021

**OWASP Category:** A05:2021 - Security Misconfiguration

- **Severity:** Low
- **Plug in :** Burp Suite Click Jacking Tester, ACUNETIX
- **PORT:** 80 for HTTP

**Description:**

Clickjacking, also known as a "UI redress attack," is a malicious technique where an attacker deceives users into clicking on unintended elements by overlaying transparent or opaque layers over legitimate web pages. This manipulation can lead users to perform actions they did not intend, such as liking a social media post, initiating financial transactions, or altering account settings, all without their awareness. For example, an attacker might create a seemingly harmless webpage offering a free prize, but in reality, it contains an invisible iframe aligned over a legitimate site's "delete all messages" button. When the user attempts to click the enticing offer, they inadvertently trigger the concealed action, resulting in potential data loss or unauthorized changes.

***Business Impact:***

- **Unauthorized Actions:** Attackers can trick users into unknowingly performing actions they didn't intend, such as making unauthorized purchases, sharing sensitive information, or granting permissions to malicious applications.

**Data Theft:** Clickjacking attacks can lead to the theft of sensitive user data. For example, attackers can deceive users into clicking on hidden



**elements that trigger the download of malware or prompt the user to enter confidential information.**

**Financial Losses: Users may suffer financial losses due to fraudulent purchases or transactions made without their knowledge or consent.**

## **7.Results:**

### **7.1 Findings and Results(Nessus And Vulnerability report)**

**Title: Exploring Cyber Security: Understanding Threats and Solutions in the Digital Age**

## **1. Cyber Threat Landscape**

The digital world faces an evolving array of cyber threats, including malware, phishing, ransomware, and insider threats. Attackers exploit vulnerabilities in software, networks, and human behavior to gain unauthorized access to systems and data.

## **2. Common Cybersecurity Threats**

- Phishing Attacks:** Deceptive emails or messages trick users into revealing sensitive information.
- Ransomware:** Malicious software encrypts files and demands payment for decryption.
- Data Breaches:** Unauthorized access to sensitive personal or business data.
- Attacks:** Distributed Denial-of-Service attacks overwhelm systems, causing downtime.
- Insider Threats:** Employees or contractors misusing access for malicious purposes.

## **3. Impact of Cyber Threats**

- Financial Losses:** Businesses lose millions due to cyberattacks, including ransom payments, regulatory fines, and recovery costs.
- Reputation Damage:** Breaches erode customer trust and brand credibility.
- Operational Disruptions:** Downtime due to attacks affects business continuity.
- Legal and Compliance Issues:** Non-compliance with data protection laws can lead to penalties.

#### 4. Essential Cybersecurity Solutions

- Endpoint Security: Antivirus, firewalls, and intrusion detection systems protect devices.
- Multi-Factor Authentication (MFA): Adds an extra layer of security beyond passwords.
- Zero Trust Architecture: Verifies every access request to minimize risk.
- Regular Security Updates: Patching vulnerabilities in software reduces the risk of exploitation.
- Security Awareness Training: Educating users helps prevent social engineering attacks.

#### 5. Best Practices for Digital Safety

- Use strong, unique passwords for different accounts.
- Enable two-factor authentication on all critical services.
- Be cautious of unexpected emails, links, and attachments.
- Regularly back up important data to a secure location.
- Stay informed about the latest cyber threats and trends.

Would you like me to adjust these headings or add

#### **Why our College Website is safe ?**

**College Website URL:** <https://bullayyacollege.org/>

#### **Why it is safe ?**

While I cannot conduct a deep technical security audit of [bullayyacollege.org](https://bullayyacollege.org/) without explicit authorization, I can highlight general reasons why a website may be considered safe and how security mechanisms work to protect users.

These are the some aspects that safe guard the college website.

#### **1.HTTPS Encryption (SSL/TLS Security)**

One of the most important indicators of a secure website is the presence of HTTPS (HyperText Transfer Protocol Secure). HTTPS ensures that communication between the user's browser and the website server is encrypted using SSL/TLS protocols. This encryption protects sensitive information, such as login credentials, personal data, and payment details, from being intercepted by hackers (man-in-the-middle attacks).

#### **The possible verification that I've done :**

I have checked the SSL certificate details by clicking the padlock icon in the browser.

I have found that the certificate has been issued by the **Trusted Certificate Authority (CA)** such as DigiCert, Let's Encrypt, or GlobalSign.

## **2.Regular Software and System Updates**

These websites are built using Content Management Systems (CMS) like WordPress, Joomla, or Drupal, or they may use custom-built frameworks. If the website administrators ensure that all software components, including the CMS, plugins, and libraries, are up to date, it reduces the risk of known vulnerabilities being exploited.

### **The possible verification that I've done :**

By using online security scanners like Qualys SSL Labs or built-in browser developer tools to check CMS versioning.

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

It is a security solution that protects a website from common cyber threats, such as SQL injection, cross-site scripting (XSS), and Distributed Denial of Service (DDoS) attacks. If bullayyacollege.org has a WAF in place, it acts as a protective barrier between the website and potential attackers.

### **The possible verification that I've done :**

This website has login functionality ,where login credentials was known to the college faculty and staff only.

By another way we can check for features like CAPTCHA during login or password reset options with security questions if they forgotten the password or any problem with the credentials.

## **4.Security Headers to Prevent Web Attacks :**

A website can be protected from various cyber threats by implementing HTTP security headers. These headers instruct web browsers on how to handle site security.

### **The possible verification that I've done :**

By using web browser developer tools (**F12 > Network > Headers**) or online tools like security headers to check security header implementation.

## **5.Secure Data Storage and Protection**

This website holds a large amount of students and faculty data like it consists of **students personal details, certificates ,marks lists etc.** It must implement strong data security measures to prevent breaches.

### **The possible verification that I've done :**

This website has a login or registration feature, so I have verified whether the passwords are stored securely and this can be assessed using ethical security testing methods.

## **6.Regular Security Audits and Penetration Testing :**

This website undergoes periodic security audits and penetration testing to identify and mitigate vulnerabilities.

**The possible verification that I've done :**

I have checked the organization log books, they have mentioned the security audits or cybersecurity certifications in those books.

**7. Protection Against DDoS Attacks**

My college website hosted on a secured infrastructure, it has given a protection against **Distributed Denial-of-Service (DDoS)** attacks, which attempt to overwhelm the server with excessive traffic.

**The possible verification that I've done :**

Checking whether the site uses **Cloudflare** or other **DDoS mitigation services** using tools like [DNSlytics](#).