Penetration Testing Project

Supervisor:
ENG/Mamdouh El-Tahairy

This presentation provides a comprehensive overview of the findings from a penetration testing engagement conducted on a web application.

Team Members

1. Youmna Ahmed El-Shennawi
2. Samir Walid Samir
3. Ahmed Mohammed Osama Alafy
4. Ibrahim abdelrazek abdelrazek Ibrahim
5. Muhamed Nashaat Elmorsy Aboelraiat
6. Taha Hamed Shosha

Project Overview

Tool Development

Our project leverages the power of Go programming language to develop a robust and efficient cybersecurity tool.

Penetration Testing Engagements

We conduct thorough penetration testing engagements to identify and mitigate vulnerabilities in your systems and applications.

Subdomain Enumeration Tool

A powerful tool for automated reconnaissance, expediting subdomain discovery and enhancing your cybersecurity arsenal.

Introduction

1 **Problem**

Gathering subdomain information can be time-consuming.

2 **Solution**

This tool automates subdomain enumeration, saving time and effort.

Key Features

Subdomain Sources

Enumerates from CRT.sh, URLScan.io, VirusTotal, and SecurityTrails.

Data Processing

Removes duplicates and sorts results, saving to a file.

External Dependencies

Utilizes curl, jq, httpx, and httprobe for comprehensive analysis.

Architecture Overview

1. ### Input

   Single domain or file with multiple domains.

2. ### API Integrations

   Leverages external services like VirusTotal and URLScan.

3. ### Output

   Sorted subdomain list stored in a text file.

How It Works

1

User Input

Specify a domain or file with domains.

2

API Calls

Gather data from CRT.sh, URLScan.io, VirusTotal, and SecurityTrails.

3

Data Processing

Clean, deduplicate, and sort the results before saving.

Code Highlights

API Integration

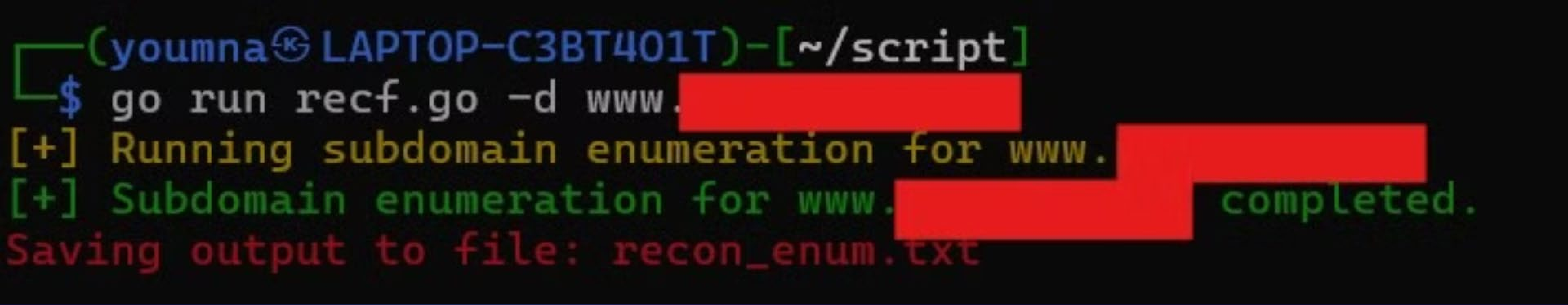
Seamless connectivity to various APIs for comprehensive data collection.

Data Processing

Efficient deduplication and sorting algorithms for clean output.

Error Handling

Robust error logging and failure recovery mechanisms.



```
┌──(youmna㉿LAPTOP-C3BT4O1T)-[~/script]
└─$ go run recf.go -d www.[redacted]
[+] Running subdomain enumeration for www.[redacted]
[+] Subdomain enumeration for www.[redacted] completed.
Saving output to file: recon_enum.txt
```

Live Demo

Domain Input

Enumerating subdomains for a test domain.

Output Displayed

Sorted subdomain list shown in real-time.

Made with Gamma

Penetration Testing Engagement

This presentation provides a comprehensive overview of the findings from a penetration testing engagement conducted on a web application.

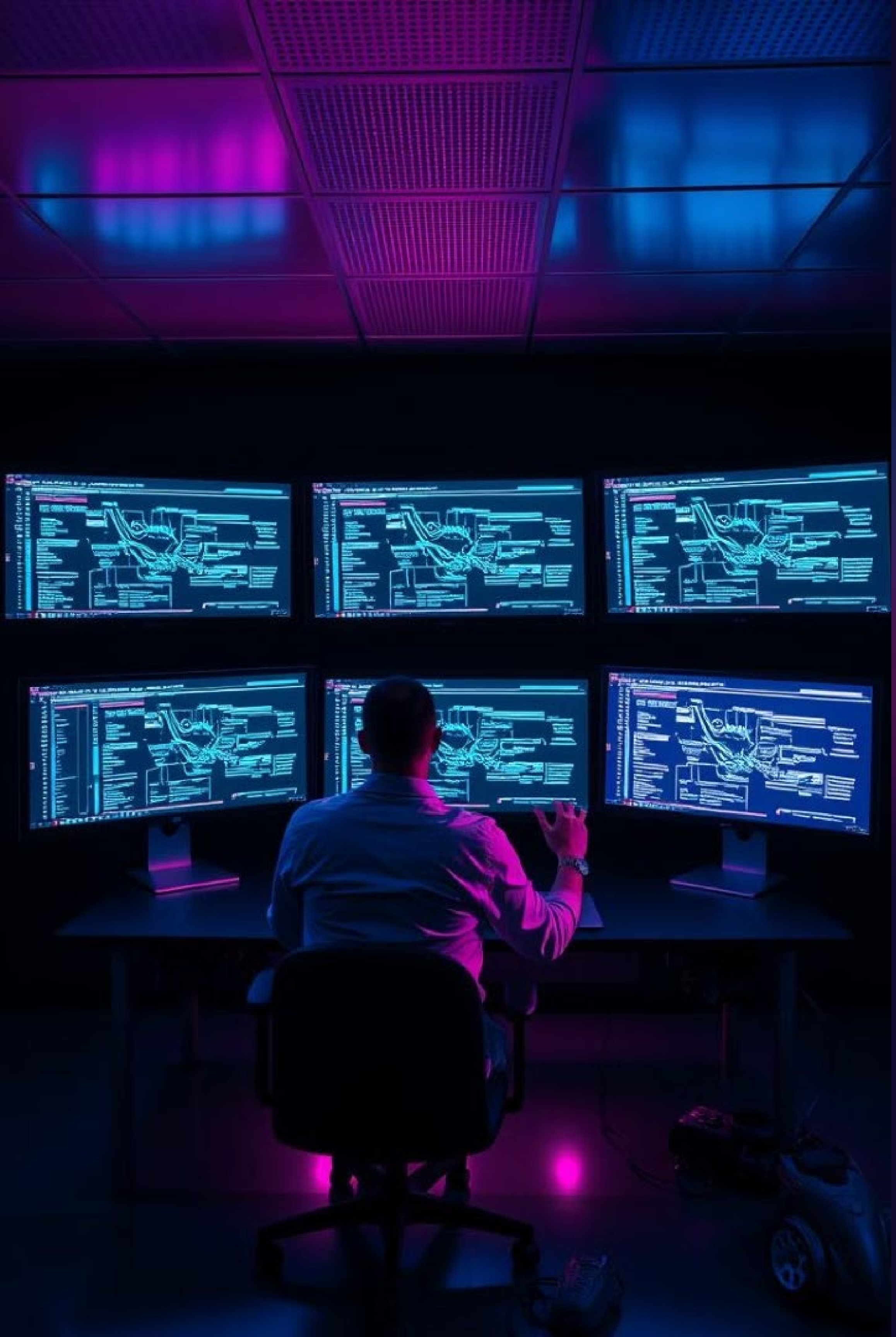
Engagement Scope & Methodology

Target	Testing Type	Framework
nahamstore.thm web application	Black-box penetration testing	OWASP Top 10 2023

Key Findings

Vulnerability	Description
Open Port (8000) & Weak Credentials	Exposed admin panel with vulnerable credentials.
LFI (Local File Inclusion)	Access to sensitive files.
CSRF	Unauthorized actions possible.
XSS	Injection of malicious scripts.
SQLi	Database manipulation.
RCE	Arbitrary code execution.

Let's start with recon phase



Reconnaissance Phase

1

Information Gathering

Collected publicly available information about NahamStore's digital footprint.

2

Network Mapping

Identified key infrastructure components and potential entry points.

3

Vulnerability Scanning

Conducted automated scans to identify known vulnerabilities in the target systems.

Subdomain Enumeration

Tool Used

Leveraged ffuf for efficient and thorough subdomain discovery.

Target Domain

Focused on nahamstore.thm as the primary domain for investigation.

Key Findings

Uncovered critical subdomains including stock, shop, marketing, and www.

Content Discovery

1

Tool Selection

Utilized Gobuster for its powerful directory and file enumeration capabilities.

2

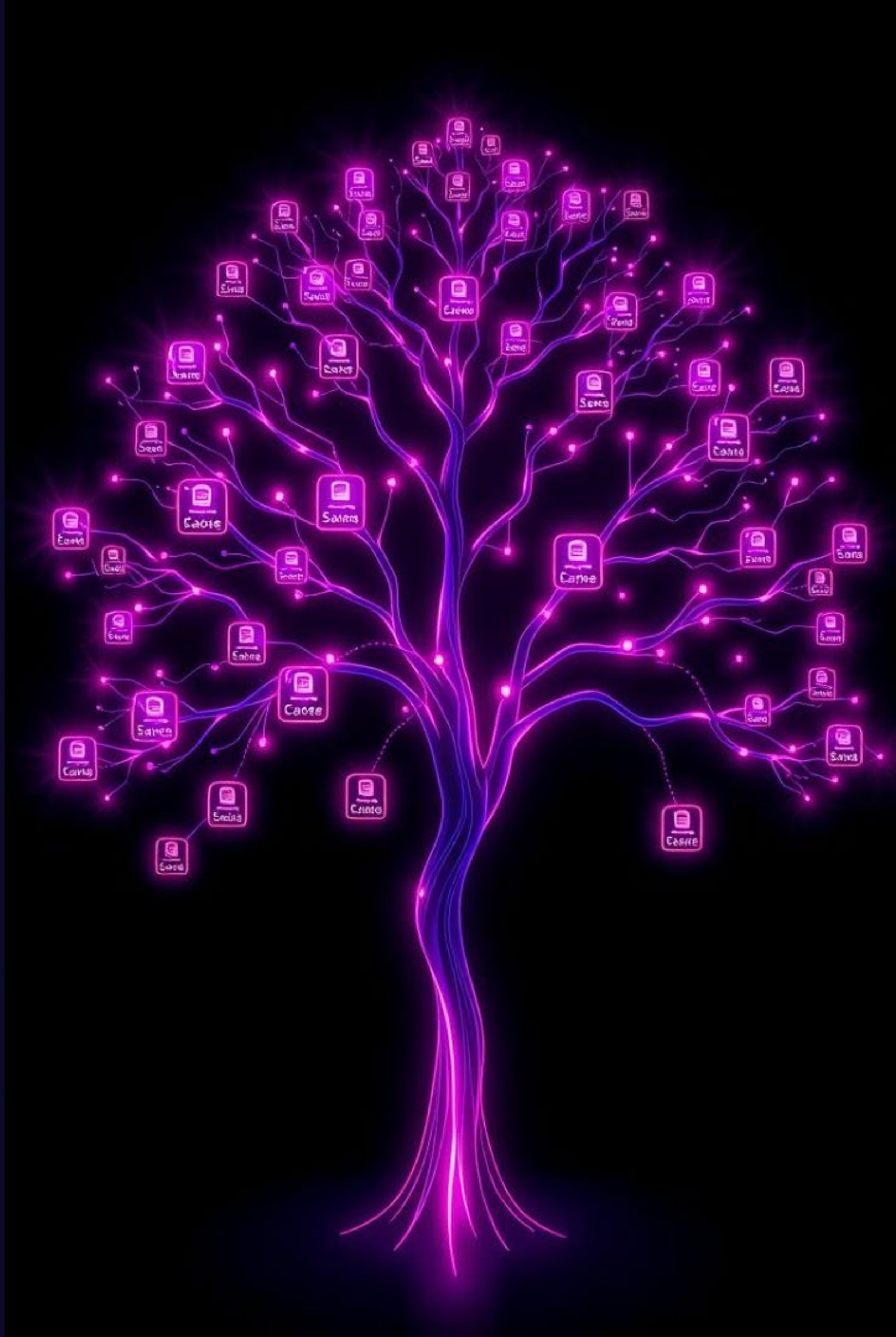
Execution

Ran Gobuster against nahamstore.thm using a comprehensive wordlist.

3

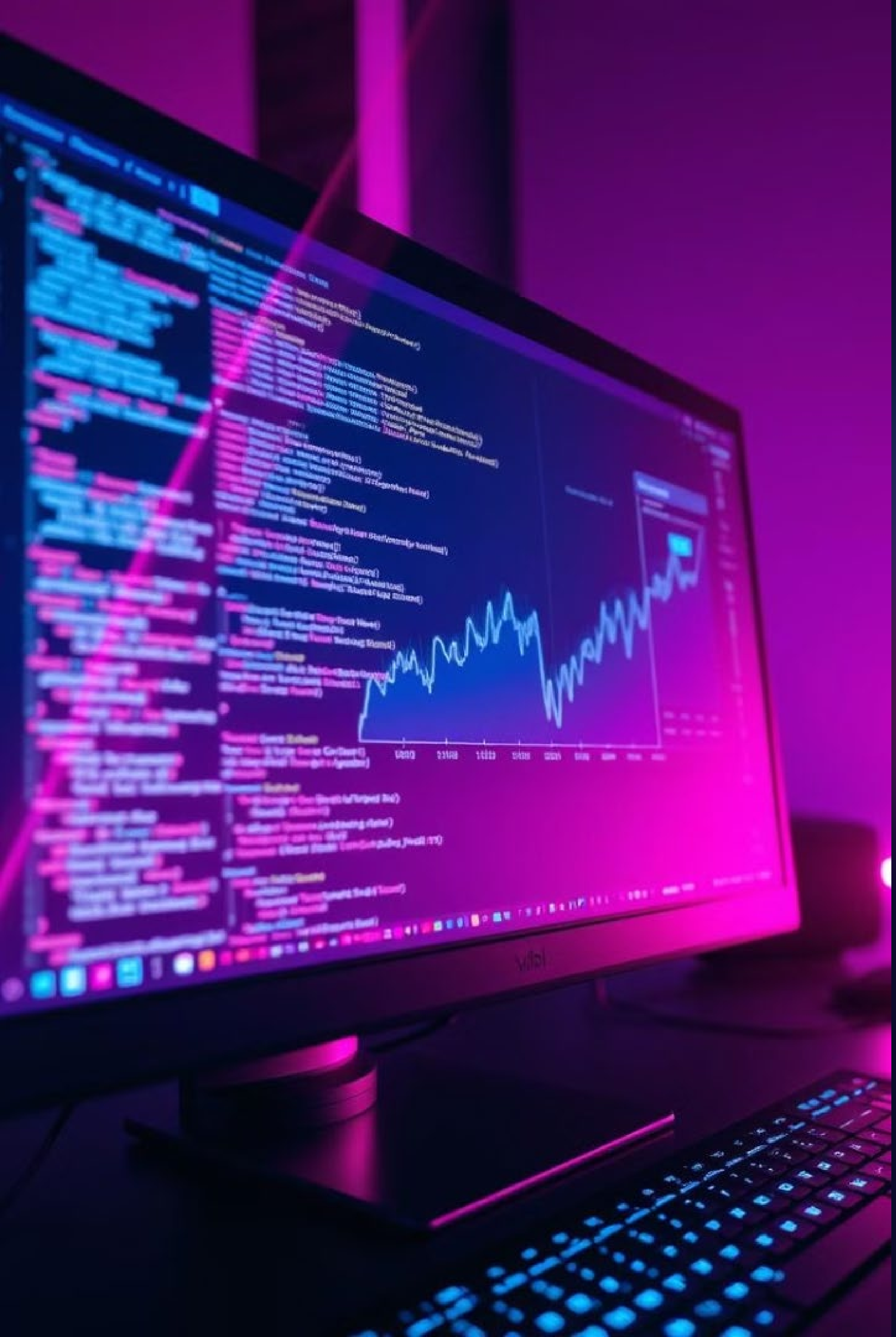
Results Analysis

Identified sensitive directories: /staff, /uploads, and /basket.

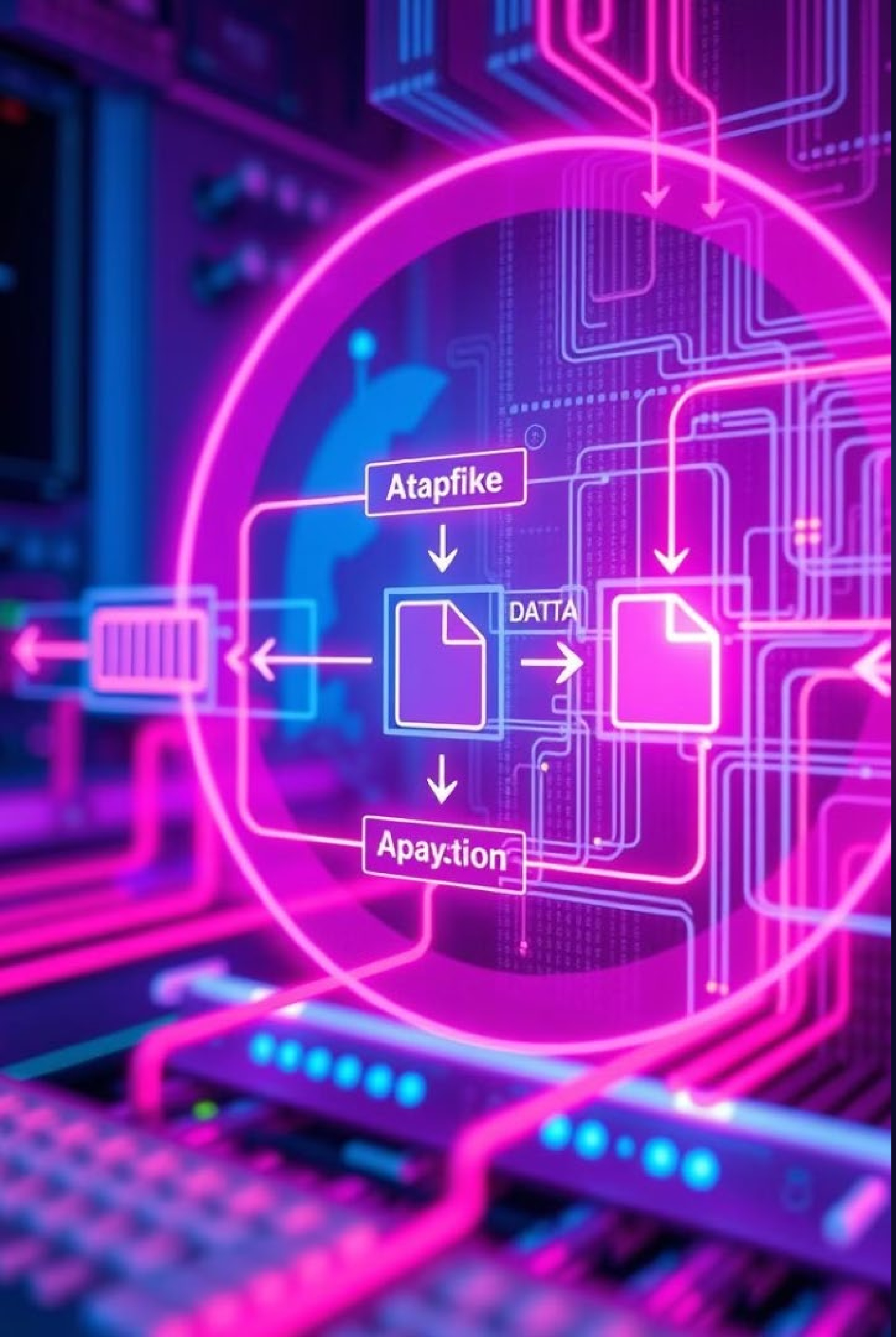


Network Vulnerability Assessment

Port	Service	Potential Risk
22	SSH	Brute Force Attacks
80	HTTP	Web Vulnerabilities
8000	Admin Panel	Unauthorized Access



Let's dive into the vulnerabilities



Local File Inclusion (LFI)

1 Vulnerability Type

Local File Inclusion (LFI) exploits weaknesses in web applications that allow attackers to access and read files on the server.

2 Exploitation

Attackers manipulate file parameters to gain access to sensitive files or execute commands through file inclusion.

3 Impact

Potential for data breaches, system compromise, and remote code execution, posing significant threats to website integrity and data security.

Mitigating LFI Vulnerabilities

Software Updates

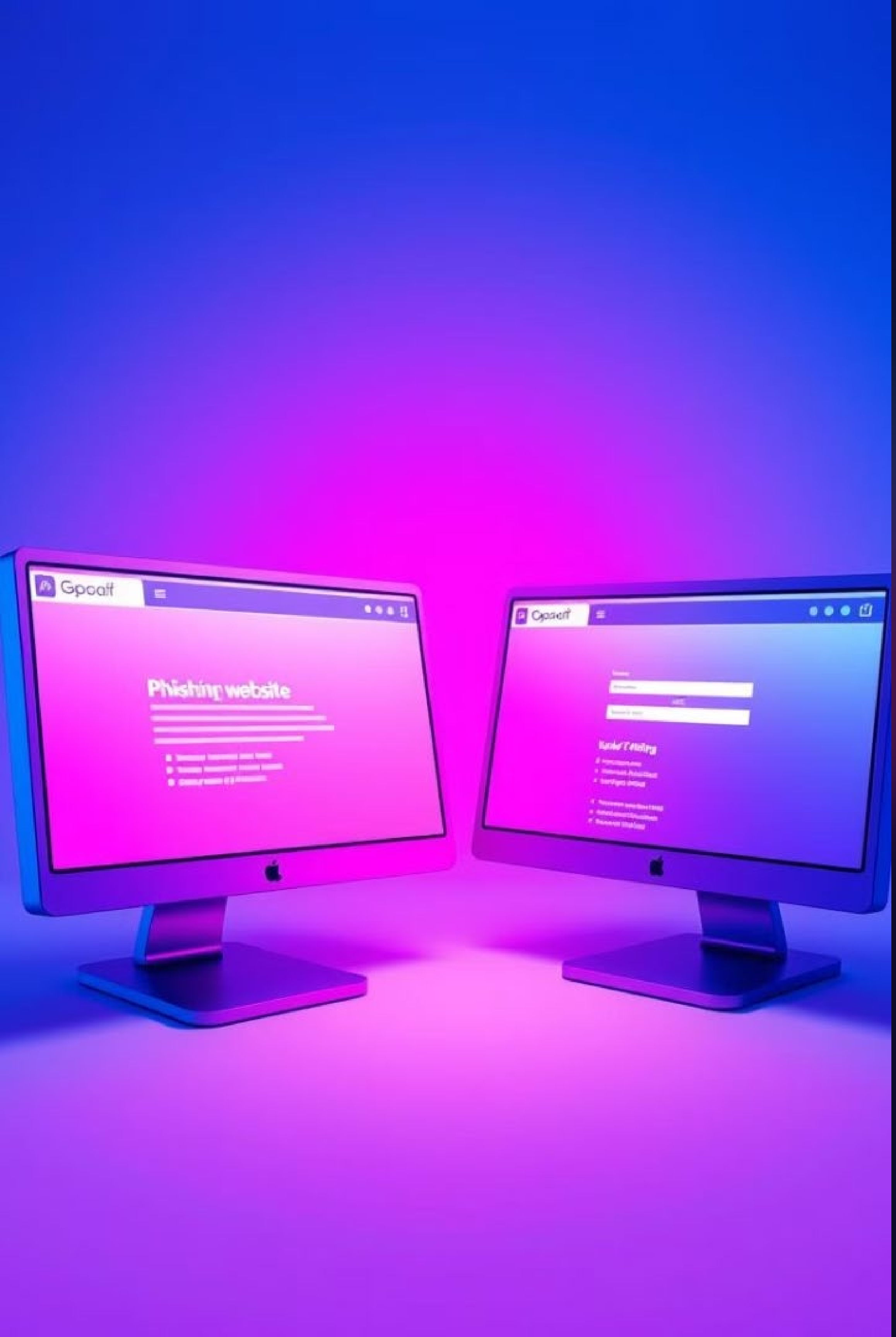
Regularly update software and dependencies to address known vulnerabilities and security patches.

Web Application Firewalls (WAFs)

Implement WAFs to block malicious requests and prevent unauthorized access to server resources.

Security Audits & Testing

Conduct regular security audits and penetration testing to identify vulnerabilities and address them proactively.



CSRF Vulnerability: Overview & Impact

Description

CSRF vulnerabilities exploit weaknesses in web applications that allow attackers to execute actions on behalf of unsuspecting users, bypassing their authorization.

Impact

Attackers can manipulate users into performing actions without their knowledge, leading to account takeovers and sensitive data manipulation.

Reproducing & Mitigating CSRF

1 Step 1

Login to the email change endpoint of the vulnerable application.

2 Step 2

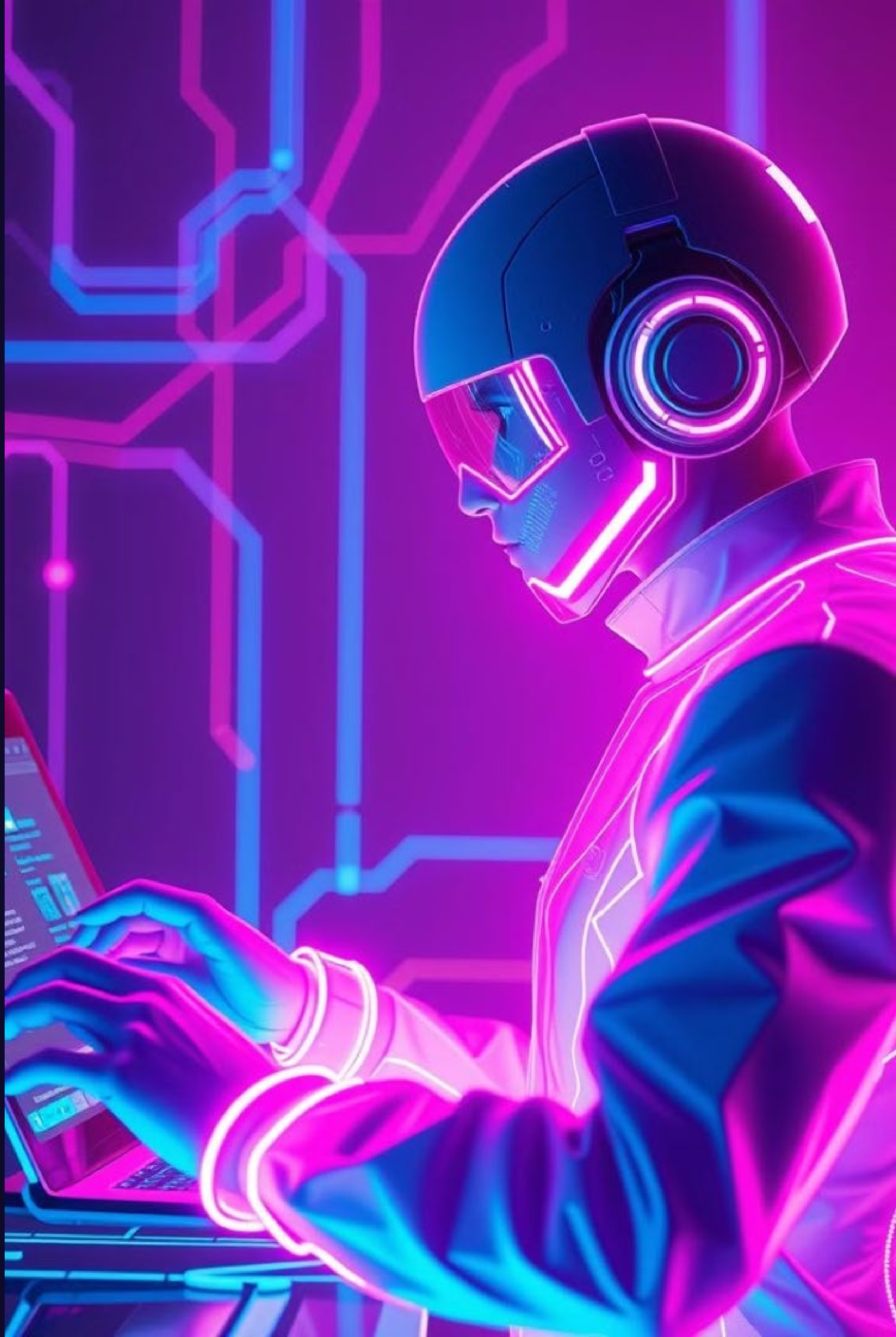
Attempt to change the email address without a valid CSRF token.

3 Step 3

Create a phishing website to trick users into visiting a malicious link, triggering the vulnerable endpoint and capturing their credentials.

4 Remediation

Implement strong CSRF protection mechanisms to validate requests and ensure user consent before executing actions.



Stored XSS Vulnerability: Overview & Impact

Vulnerability Type	Stored XSS
Location	User-Agent header on /basket endpoint
Impact	Potential for stealing session cookies, hijacking user accounts, and executing malicious scripts within the user's browser.

Reproducing & Mitigating Stored XSS

Step 1

Initiate a purchase process and intercept the request using a tool like Burp Suite.

Step 2

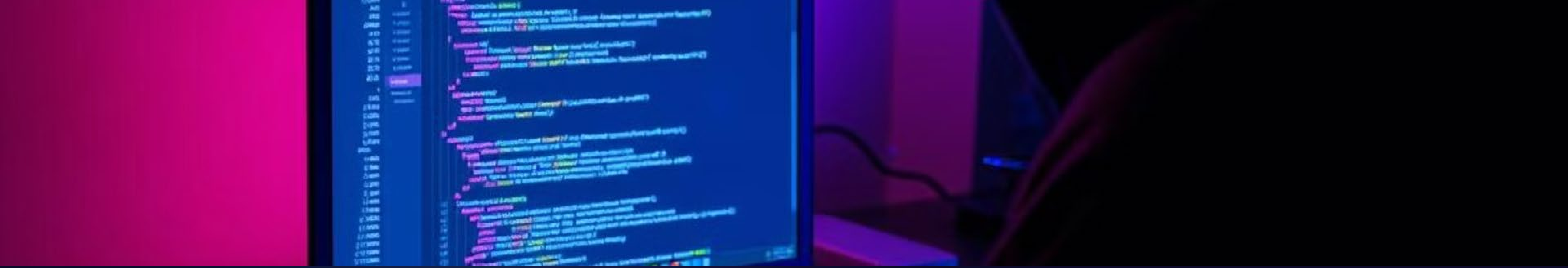
Inject malicious JavaScript code into the User-Agent header of the intercepted request.

Step 3

Observe the injected script executing in the user's browser.

Recommendations

Implement robust input validation, sanitize user inputs, and use Content Security Policy (CSP) to mitigate XSS vulnerabilities.



Reflected Cross-Site Scripting (XSS)

1 **Web Application Vulnerability**

This type of attack allows attackers to inject malicious scripts into a website, potentially stealing sensitive information, hijacking user sessions, or redirecting users to malicious websites.

2 **Risk Rating**

Reflected XSS is rated as a medium to high risk due to its potential to compromise user data and system security.

3 **Vulnerability Description**

Reflected XSS occurs when an attacker injects malicious scripts into a website's input fields, which are then reflected back to the user's browser.

4 **Impact**

Successful XSS attacks can lead to various malicious outcomes, such as account takeover, data theft, and unauthorized actions within the user's context.

Reflected XSS: Mitigation Strategies

1

Input Validation

Sanitize user input by removing or escaping any characters that could potentially be used to execute malicious scripts.

2

Content Security Policy (CSP)

Implement a CSP to define the trusted sources of content and restrict the execution of scripts from untrusted sources, limiting the impact of XSS attacks.

3

Output Encoding

Encode all user-supplied data before it is displayed on the website, preventing malicious scripts from being executed.

Insecure Direct Object References (IDOR)

Type	Risk Rating	Description
Web Application Vulnerability	Medium Risk	IDOR allows attackers to access and manipulate data by modifying IDs in requests, bypassing intended access controls.

IDOR: Exploitation & Mitigation

1 **Steps to Reproduce**

Attackers exploit IDOR vulnerabilities by modifying identifiers in web requests to gain unauthorized access to data.

2 **Impact**

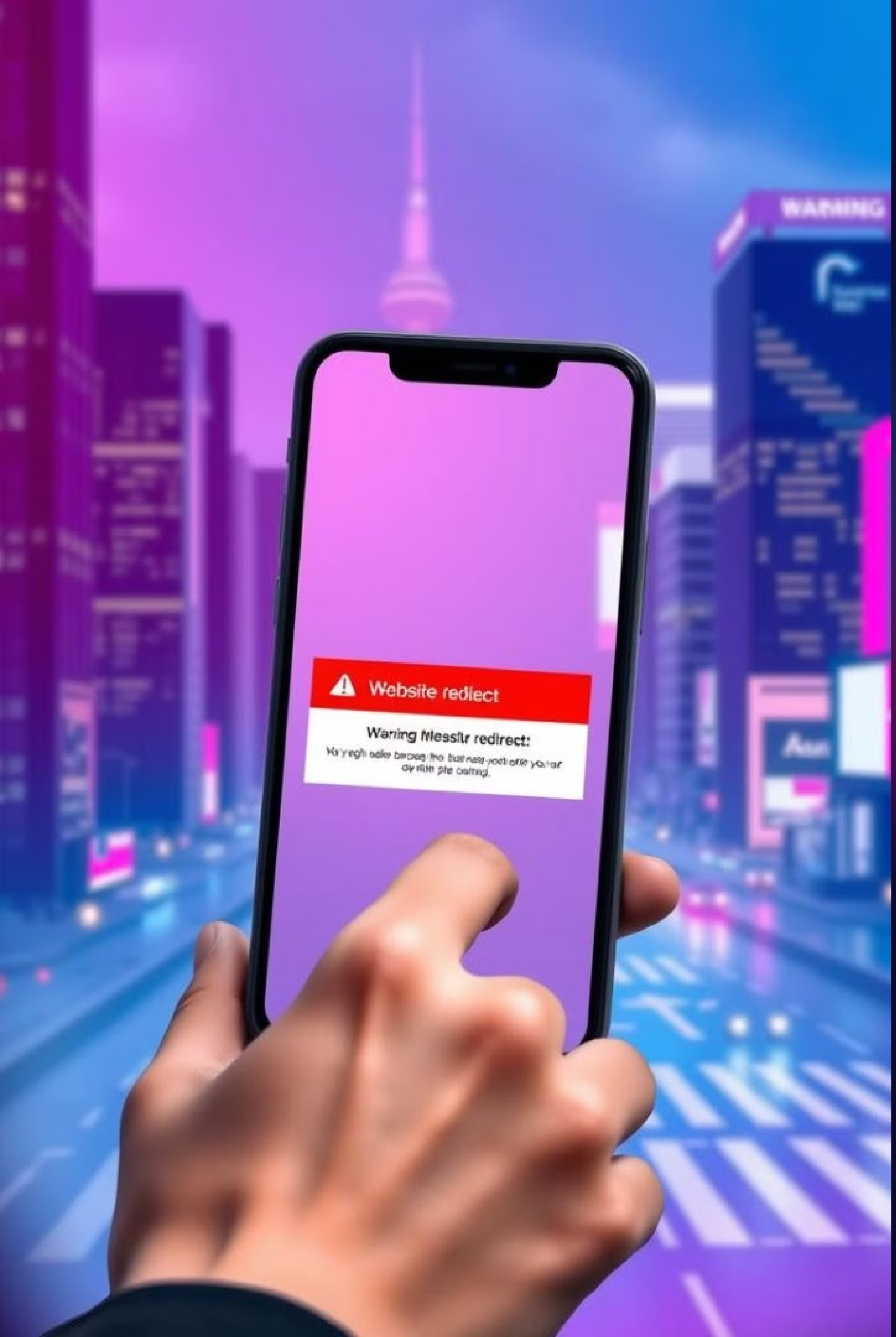
IDOR vulnerabilities lead to unauthorized access to sensitive user data, potentially compromising privacy and security.

3 **Recommendations**

Implement robust access control mechanisms to restrict access to data based on user permissions, validating all user inputs to prevent manipulation of identifiers.

Open Redirect Vulnerabilities

Type	Web Application
Risk Rating	Low
Description	Open redirects allow attackers to manipulate redirection URLs, directing users to malicious websites.



Open Redirect: Mitigation Strategies

Avoid Unnecessary Redirects

Minimize the use of redirects to reduce the potential for exploitation.

Validate Redirect URLs

Ensure that redirect URLs are only pointing to trusted and authorized destinations.

Implement Input Sanitization

Validate user inputs to prevent attackers from manipulating redirect URLs.

XXE (XML External Entity)

XXE is a high-risk vulnerability allowing attackers to read sensitive server files via XML.

1 Exploiting XXE

Attackers exploit this vulnerability by injecting payloads to access sensitive files. This attack involves sending crafted XML requests with malicious external entity references.

2 Server File Access

These malicious references can lead to the retrieval of sensitive files like configuration files, system logs, or even passwords stored on the server.

3 Data Exfiltration

Attackers can then exfiltrate this stolen data using various channels like network connections or external services, compromising the security of the system.

XXE Impact & Recommendations

XXE attacks pose a significant threat, potentially exposing sensitive information and facilitating further attacks on internal systems.

Impact

XXE exploits can lead to the disclosure of sensitive information, jeopardizing confidentiality, integrity, and availability of critical data and systems.

- Data Exfiltration
- System Compromise
- Unauthorized Access

Recommendations

Several steps can mitigate the risk of XXE attacks.

- Disable DTD processing
- Validate XML data
- Use a secure XML parser

Remote Code Execution (RCE) Vulnerability

1

Discovery

Identified vulnerability in /account/order/6 endpoint allowing arbitrary command execution.

2

Exploitation

Successfully executed commands by manipulating request parameters.

3

Impact

Achieved full control over the server, posing a critical security risk.

RCE Exploitation Technique

Payload Crafting

Developed a sophisticated PHP reverse shell payload for maximum impact.

Delivery Method

Utilized URL encoding to bypass security filters and deliver the payload.

Reverse Shell

Established a Netcat listener to catch and maintain persistent access.

Admin Panel Vulnerability

Default Credentials

Admin panel accessible using common default login: admin:admin.

Misconfiguration

Failure to change default credentials exposes critical administrative functions.

Remediation

Implement strong, unique passwords and multi-factor authentication for all admin accounts.

Sensitive Data Disclosure

1 Vulnerability Location

Misconfigured subdomain nahamstore-2020-dev.nahamstore.thm exposes customer data.

2 Exposed Information

Social Security Numbers (SSNs) and other personally identifiable information (PII) accessible.

3 Potential Consequences

High risk of identity theft and severe legal ramifications for the company.

4 Urgent Action Required

Immediate data protection measures and security audit of all subdomains necessary.

SQL Injection (Standard)

Standard SQL injection attacks manipulate SQL queries through unsanitized user input, allowing attackers to access or modify sensitive data.

1

Vulnerable Parameter

Attackers identify vulnerable parameters in web applications, usually input fields accepting user-provided data.

2

SQL Injection Payload

They then craft malicious SQL payloads containing commands designed to bypass security measures and manipulate the database.

3

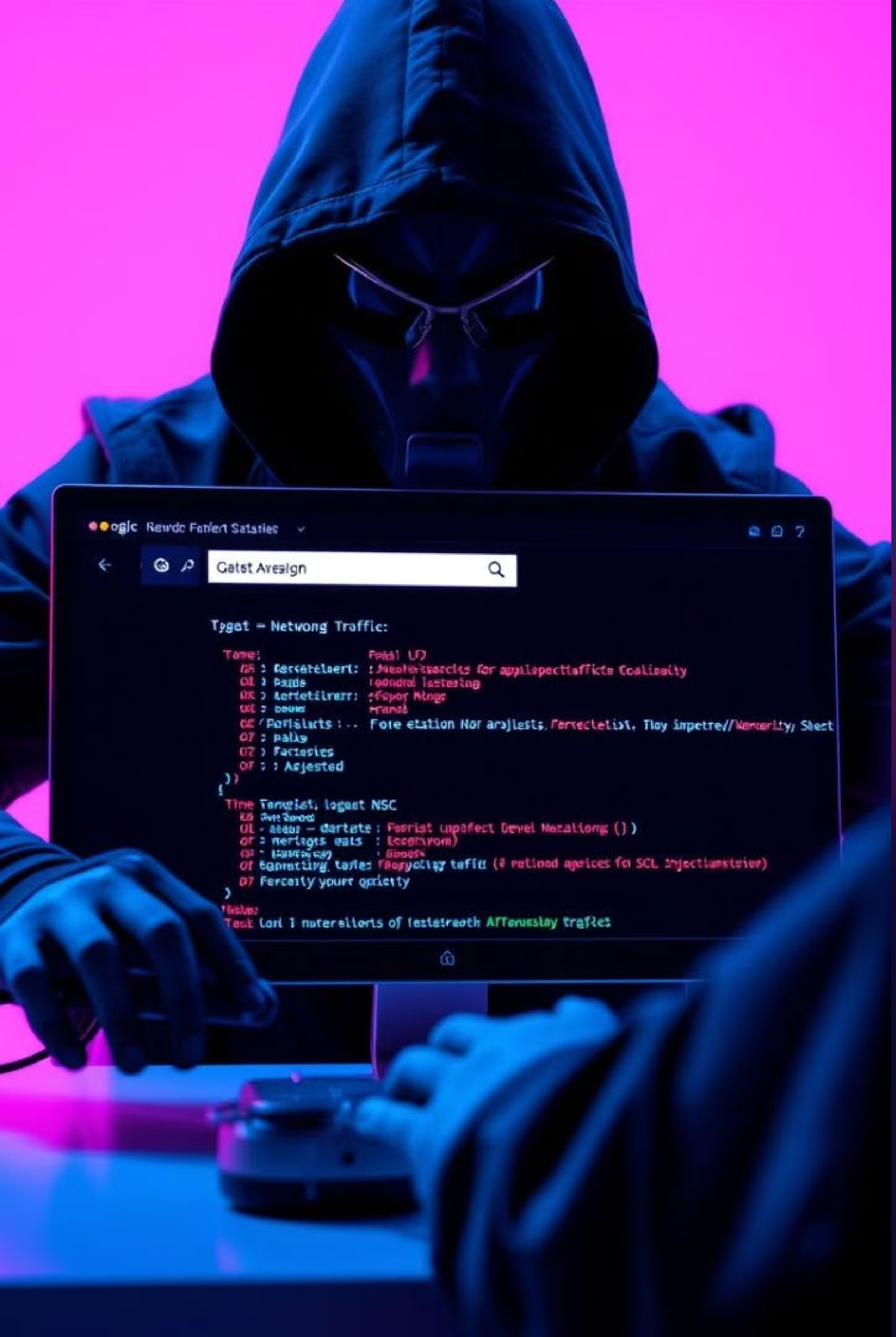
Data Extraction

The injected payloads can extract sensitive data such as usernames, passwords, or financial records from the database, compromising user accounts and sensitive information.

SQL Injection (Standard) Impact & Recommendations

Standard SQL injection attacks can have severe consequences, allowing unauthorized access to database information, potentially leading to data loss and system corruption.

Impact	Recommendations
Data Breaches	Prepared Statements
System Compromise	Input Sanitization
Denial of Service	Regular Security Audits



SQL Injection (Blind)

Blind SQL injection attacks use inference-based techniques to gather data without receiving direct feedback from the database.

1

Interception

Attackers use tools like Burp Suite to intercept network requests and responses between the web application and the database.

2

Blind SQL Injection

They then craft queries that elicit different responses based on the truth or falsity of conditions, allowing them to infer data without direct feedback from the database.

3

Data Extraction

Using techniques like time-based attacks, boolean-based attacks, or error-based attacks, attackers can extract sensitive information one bit at a time, piecing together the desired data.

SQL Injection (Blind) Impact & Recommendations

Blind SQL injection attacks are particularly dangerous as they can extract data without triggering typical database error messages, making them difficult to detect.

Impact

Blind SQL injection can lead to the exfiltration of sensitive data without leaving clear traces in logs or error messages, increasing the difficulty of detection and response.

Recommendations

Implementing robust security measures can significantly reduce the risk of blind SQL injection attacks.

Mitigation

Rate limiting, error handling, and input validation are critical for preventing these attacks.

Thank You

We appreciate your time and look forward to discussing how we can further enhance your cybersecurity posture.