Classification - Project SAF

1. **Defense based on DOS Attack - 3 Papers**
   [1] Implements a comprehensive defense strategy across network, transport, and application layers, including Simplified Hop Count Filtering (SHCF), SYN Proxy Firewall, and traffic limits.
   [2] Deploys threshold limits, ACLs, IP Verify, and network load balancing for effective traffic management, with the threshold limit proving most effective.
   [3] Focuses on advanced detection algorithms and CAPTCHAs at the application layer to counter sophisticated attacks, with effectiveness reliant on distinguishing between legitimate users and attackers.

2. **Defense against Cross Site Scripting - 6 Papers**
   [4]Emphasizes educating users on checking URL authenticity, understanding link security implications, and implementing HTTPS, HTTP-only, Secure flags on cookies, and XSS protection on browsers to prevent attacks.
   [5] Utilizes static and dynamic mapping models to detect anomalies, sanitizes user inputs, and employs dedicated web service instances to isolate client activities, reducing the impact of attacks.
   [6] Relies on browser-built XSS filters and extensions like the XSS-Me add-on for Firefox to detect and neutralize malicious scripts before they execute, emphasizing client-side solutions for preventing XSS attacks.
   [7] Implements Moving Target Defense (MTD) technology, adding random attributes to web application elements to distinguish between legitimate and malicious JavaScript code, effectively mitigating XSS attacks with minimal impact on system performance.
   [8] The defense strategy outlined in the research focuses on several key measures to mitigate Cross-site Scripting (XSS) vulnerabilities in an international student website. These include input verification to validate data based on length, type, syntax, and business rules, output encoding to encode user-submitted data and prevent script execution, explicit specification of output encoding modes to avoid manipulation by attackers, and awareness of injection points within applications for targeted sanitization and encoding, collectively ensuring robust protection against XSS threats.
   [20] Utilizes the CXSSor tool, employing web crawler technology to proactively identify and mitigate XSS vulnerabilities in web applications by systematically detecting potential injection points and testing them with malicious payloads, focusing on early detection and prevention.

3. **Defense against Brute Force attack - 1 Paper**
   [9] implementation of a Honeypot system as a decoy server to protect real servers from brute force attacks, while also logging all interactions, including attempted attacks and gives insights into the attackers methods.

4. **Defense against CAPTCHA attacks - 2 Paper**

   [10]Implementation of Anti-segmentation Techniques like the use of adhesion, interlacing, and varying font styles to increase the difficulty of distinguishing characters by automated systems.

   [11] Visual Cryptography to create two shares of a CAPTCHA image, enhancing user authentication in online banking and enabling clients to distinguish between genuine and phishing sites

5. **Defense against SQL Injection Attacks - 5 Papers**

   [12] Defense strategies include filtering user input, utilizing Cloudflare protection, enforcing HTTPS, avoiding dynamic SQL, applying updates and patches, encrypting sensitive data, and monitoring SQL statements for anomalies.

   [13] Combines penetration testing with advanced tools. It employs Acunetix for vulnerability scanning, Burp Suite for intercepting and analyzing HTTP traffic, the CO2 extension for automatic SQL command generation, and SQL map for exploiting vulnerabilities. This multi tool approach aims to detect, assess, and improve security measures against SQL injection in web applications across varying security levels.

   [14] Presents a methodology and tool for injecting vulnerabilities and attacks into web applications, enabling automated testing of security mechanisms. This approach uses realistic vulnerabilities derived from a comprehensive field study, facilitating the evaluation of intrusion detection systems, web application firewalls, and vulnerability scanners. It provides an environment for assessing security tools' effectiveness, training security teams, and improving web application defenses.

   [15] Outlines defense strategies against web application attacks, including validating input structures, enforcing security policies, randomizing code, tracking untrusted data, and learning normal behaviors to spot anomalies, aiming for a multi-layered protection approach.

   [16] Addresses SQL Injection defense through multiple strategies including signature, knowledge, statistical, behavior-based, and hybrid detection methods. It specifically highlights Joza, a hybrid taint inference system for PHP applications, combining positive and negative inference to prevent SQL Injection with low performance impact. These methods aim to improve detection rates and reduce vulnerabilities in web applications.

6. **Defense against Web Crawlers - 1 Paper**

   [17] Mbot is a specially crafted tool used in research to emulate the behavior of non-compliant web crawlers. It's designed to intentionally disregard the guidelines set by robots.txt, a standard file used by websites to instruct web crawlers on which areas they are permitted or forbidden to access. While robots.txt serves as a voluntary set of rules for ethical crawler conduct, helping website administrators to safeguard their resources and sensitive data, Mbot is utilized to test and analyze the effectiveness of

search engine websites' defensive mechanisms against crawlers that do not follow these established norms.

7. **Web application defense using Firewall**

   [18]Web application defense strategy using a Web Application Firewall, combining ModSecurity for HTTP traffic filtering and a Reverse Proxy method for enhanced security against common attacks.

   [21] The hybrid web application firewall developed in the study combines signature-based detection for known attack patterns and anomaly detection, using statistical analysis of HTTP requests, to effectively prevent both known and emerging web-based attacks.

Reference Papers

[1] A Three-Layer Defense Mechanism Based on WEB Servers Against Distributed Denial of Service Attacks

[2] Analysis of UDP DDoS Flood Cyber Attack and Defense Mechanisms on Web Server with Linux Ubuntu 13

[3] DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications

[4] Vulnerability analysis of online banking sites to cross-site scripting and request forgery attacks in East Africa

[5]Detection of SQL Injection and XSS attacks in three-tier web applications

[6] Browser's Defenses Against Reflected Cross Site Scripting Attacks

[7] Research and Implementation of Cross Site Scripting Defense Method Based on Moving Target Defense Technology

[8] Research on Cross-site Scripting Vulnerability of XSS Based on International Student Website

[9] Website and Network Security Techniques against Brute Force Attacks using Honeypot

[10] A Case Study of Text-Based CAPTCHA Attacks

[11] Prevention of Phishing Website Attacks in Online Banking Systems Using Visual Cryptography

[12] Overview of SQL Injection Defense Mechanism

[13] Automatic Protection of Web Applications against SQL Injection an Approach Based on Acunetix Burp Suite and SQLMap.

[14] Evaluation of Web Security Mechanisms using Vulnerability Amp attack Injection.

[15] Recent Attack prevention techniques in web service applications.

[16] Defending against web application attacks approaches challenges and implications.

[17]Defense response of search engine websites to non cooperating crawlers.

[18] Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall

[19]Cyber security analysis of internet banking in emerging countries User and bank perspectives

[20] A Crawler-Based Vulnerability Detection Method for Cross Site Scripting

[21]Development of a hybrid web application firewall to prevent web based attacks