KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

BÁO CÁO ĐỒ ÁN MÔN HỌC

ĐỀ TÀI: Develop Breach Attack Simulation Web Application Firewall

Nhóm: 07

21522620 - Hồ Ngọc Thiện
21522492 - Ngô Minh Quân
21522312 - Phùng Đức Lương
21522483 - Chu Nguyễn Hoàng Phương

GVHD: Ths.Nguyễn Duy

University of Information Technology, VNU-HCM, Vietnam

Nội dung báo cáo

- Phần I: Ngữ cảnh mục tiêu đề tài
- Phần II: Phương pháp
- Phần III: Giới thiệu các công cụ Scanning
- Phần IV: Kết quả

Nội dung báo cáo

- **Phần I: Ngữ cảnh mục tiêu đề tài**
- **Phần II: Phương pháp**
- **Phần III: Giới thiệu các công cụ Scanning**
- **Phần IV: Kết quả**

Ngữ cảnh, mục tiêu đề tài

- **Breach and Attack Simulation (BAS)** là một phương pháp kiểm thử an ninh mạng mô phỏng các cuộc tấn công thực tế nhằm đánh giá hiệu quả của các biện pháp bảo mật. Quy trình tự động và liên tục này cho phép các tổ chức kiểm thử hệ thống phòng thủ thường xuyên, đảm bảo rằng các biện pháp bảo mật luôn được cập nhật và có khả năng phát hiện cũng như giảm thiểu các mối đe dọa tiềm tàng.

- Với **Web Application Firewall (WAF) Simulation**, chúng ta kiểm tra xem cấu hình, triển khai và các tính năng WAF có thể chặn các payload trước khi chúng đến gần các ứng dụng web hay không.

- Chúng ta mô phỏng hành động rằng hacker, cố gắng bypass WAF và tiếp cận ứng dụng web, sau đó hắn cố gắng thực hiện các hành động độc hại như khai thác thông tin nhạy cảm, gây thiệt hại và chuyển hướng người dùng đến các trang web bị nhiễm bằng các cuộc tấn công ứng dụng XSS, SQLi, Command Injection….

Nội dung báo cáo

- Phần I: Ngữ cảnh mục tiêu đề tài
- Phần II: Phương pháp
- Phần III: Giới thiệu các công cụ Scanning
- Phần IV: Kết quả

Phương pháp

- Quét lỗ hổng: Dùng công cụ như Acunetix, ZAP, Qualys, Nessus… để quét ứng dụng qua WAF, phát hiện lỗ hổng chưa chặn được (SQLi, XSS...).
- Tạo báo cáo: Mô tả chi tiết lỗ hổng còn tồn tại nhằm cải thiện rule trên WAF để ngăn chặn.

Phương pháp

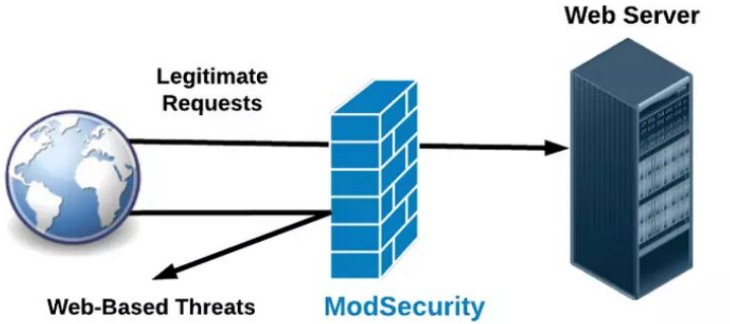
1. DVWA

- DVWA (Damn Vulnerable Web Application) là một ứng dụng web được thiết kế đặc biệt cho mục đích học tập và kiểm tra bảo mật. DVWA được dùng để thực hành các kỹ thuật tấn công phổ biến trên ứng dụng web như SQL Injection, Cross-Site Scripting (XSS), và Cross-Site Request Forgery (CSRF)....
- Cách cài đặt: https://github.com/digininja/DVWA

Phương pháp

2. ModSecurity (WAF)

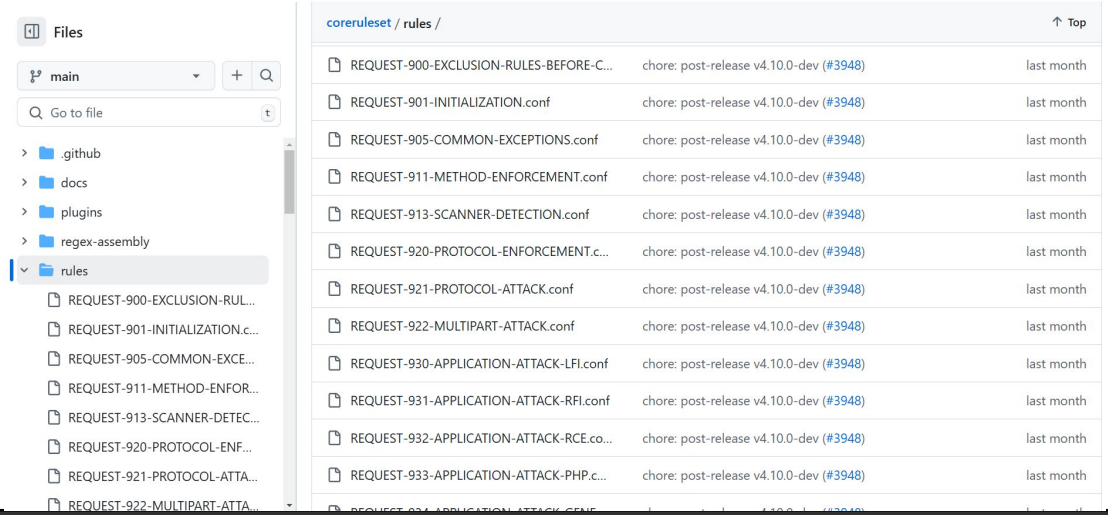
- ModSecurity là WAF cung cấp thêm một lớp bảo mật cho các ứng dụng web. Đây là một mô-đun nguồn mở cho Apache, IIS và Nginx HTTP Server giúp bảo vệ các trang web có lỗ hổng web (SQLi, XSS, Path Traversal, ...).
- Một số chức năng của Modsecurity: Prevent attacks to web application (Apache support), Logging and monitoring, SSL/TLS support, Customizable rules...
- Cách cài đặt: sudo apt install libapache2-mod-security2



Phương pháp

3. ModSecurity - CRS

- ModSecurity Core Rule Set (CRS) là một tập hợp các quy tắc bảo mật mã nguồn mở được thiết kế để chạy với ModSecurity nhằm bảo vệ các ứng dụng web khỏi nhiều cuộc tấn công, bao gồm OWASP Top 10, với tỉ lệ false positive thấp.
- Link: https://github.com/coreruleset/coreruleset.git



Phương pháp

STT	File	Chặn lỗ hổng / Mục đích
1	REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example	Định nghĩa các quy tắc loại trừ chạy trước CRS
2	REQUEST-901-INITIALIZATION.conf	Khởi tạo các quy tắc và thiết lập môi trường
3	REQUEST-905-COMMON-EXCEPTIONS.conf	Xử lý các ngoại lệ phổ biến
4	REQUEST-911-METHOD-ENFORCEMENT.conf	Giới hạn và kiểm tra các phương thức HTTP
5	REQUEST-913-SCANNER-DETECTION.conf	Phát hiện và chặn các công cụ quét (scanners)
6	REQUEST-920-PROTOCOL-ENFORCEMENT.conf	Chặn các vi phạm liên quan đến giao thức HTTP
7	REQUEST-921-PROTOCOL-ATTACK.conf	Phát hiện và chặn các tấn công giao thức HTTP
8	REQUEST-922-MULTIPART-ATTACK.conf	Xử lý các tấn công liên quan đến dữ liệu multipart
9	REQUEST-930-APPLICATION-ATTACK-LFI.conf	Chặn Local File Inclusion (LFI)
10	REQUEST-931-APPLICATION-ATTACK-RFI.conf	Chặn Remote File Inclusion (RFI)

11	REQUEST-932-APPLICATION-ATTACK-RCE.conf	Chặn Remote Code Execution (RCE)
12	REQUEST-933-APPLICATION-ATTACK-PHP.conf	Chặn PHP Injection
13	REQUEST-934-APPLICATION-ATTACK-GENERIC.conf	Chặn các tấn công ứng dụng không đặc thù
14	REQUEST-941-APPLICATION-ATTACK-XSS.conf	Chặn Cross-Site Scripting (XSS)
15	REQUEST-942-APPLICATION-ATTACK-SQLI.conf	Chặn SQL Injection
16	REQUEST-943-APPLICATION-ATTACK-SESSION-FIXATION.conf	Chặn Session Fixation
17	REQUEST-944-APPLICATION-ATTACK-JAVA.conf	Chặn tấn công liên quan đến ứng dụng Java
18	REQUEST-949-BLOCKING-EVALUATION.conf	Đánh giá mức độ rủi ro và chặn các yêu cầu
19	RESPONSE-950-DATA-LEAKAGES.conf	Phát hiện và ngăn chặn rò rỉ dữ liệu nhạy cảm
20	RESPONSE-951-DATA-LEAKAGES-SQL.conf	Ngăn chặn rò rỉ dữ liệu liên quan đến SQL

21	RESPONSE-952-DATA-LEAKAGES-JAVA.conf	Ngăn chặn rò rỉ dữ liệu liên quan đến Java
22	RESPONSE-953-DATA-LEAKAGES-PHP.conf	Ngăn chặn rò rỉ dữ liệu liên quan đến PHP
23	RESPONSE-954-DATA-LEAKAGES-IIS.conf	Ngăn chặn rò rỉ dữ liệu liên quan đến IIS
24	RESPONSE-955-WEB-SHELLS.conf	Phát hiện và chặn các web shell
25	RESPONSE-959-BLOCKING-EVALUATION.conf	Đánh giá mức độ rủi ro từ phản hồi và chặn
26	RESPONSE-980-CORRELATION.conf	Phân tích tương quan các phản hồi để phát hiện hành vi bất thường
27	RESPONSE-999-EXCLUSION-RULES-AFTER-CRS.conf.example	Định nghĩa các quy tắc loại trừ chạy sau CRS
28	iis-errors.data	Dữ liệu liên quan đến lỗi IIS, phục vụ phát hiện tấn công
29	java-classes.data	Danh sách các lớp Java để phát hiện rò rỉ dữ liệu
30	java-code-leakages.data	Dữ liệu phát hiện rò rỉ mã nguồn Java

31	java-errors.data	Danh sách lỗi Java để phát hiện rò rỉ dữ liệu
32	lfi-os-files.data	Danh sách file hệ điều hành để phát hiện Local File Inclusion (LFI)
33	php-config-directives.data	Danh sách các chỉ thị cấu hình PHP để phát hiện tấn công
34	php-errors-pl2.data	Danh sách lỗi PHP cho bảo vệ cấp độ 2 (PL-2)
35	php-errors.data	Danh sách lỗi PHP để phát hiện rò rỉ dữ liệu
36	php-function-names-933150.data	Danh sách hàm PHP để phát hiện PHP Injection
37	php-function-names-933151.data	Danh sách hàm PHP để phát hiện PHP Injection
38	php-variables.data	Danh sách các biến PHP để phát hiện tấn công
39	restricted-files.data	Danh sách các file bị hạn chế để phát hiện truy cập trái phép
40	restricted-upload.data	Danh sách các file bị hạn chế khi upload

Nội dung báo cáo

- Phần I: Ngữ cảnh mục tiêu đề tài
- Phần II: Phương pháp
- Phần III: Giới thiệu các công cụ Scanning
- Phần IV: Kết quả

Giới thiệu các công cụ Scanning

1. Acunetix

- Acunetix là một công cụ quét tập trung vào việc tìm kiếm các lỗ hổng bảo mật trên các trang web, ứng dụng web, và các API.

Chức năng chính:

+ Phát hiện các lỗ hổng phổ biến như SQL Injection, Cross-Site Scripting (XSS), và các cấu hình không an toàn.
+ Quét cả các ứng dụng web hiện đại như những ứng dụng dùng JavaScript, AJAX, và HTML5.
+ Cung cấp chức năng quét tự động và có khả năng kiểm tra nhiều ứng dụng cùng lúc.

Giới thiệu các công cụ Scanning

2. ZAP

- OWASP ZAP là một công cụ mã nguồn mở được phát triển bởi dự án OWASP, nhằm quét và kiểm tra lỗ hổng bảo mật cho các ứng dụng web.

Chức năng chính:

+ Quét các lỗ hổng bảo mật trong ứng dụng web như SQL Injection, XSS, và các lỗi cấu hình bảo mật.
+ Cung cấp tính năng kiểm thử thâm nhập (pen testing) bằng cách mô phỏng các cuộc tấn công trên web.
+ Hỗ trợ tính năng proxy để giám sát lưu lượng HTTP/HTTPS và phân tích các yêu cầu, phản hồi.

Giới thiệu các công cụ Scanning

3. Qualys

- Qualys Community Edition là một phiên bản miễn phí của Enterprise TruRisk Platform được thiết kế dành cho cá nhân, các nhóm nhỏ, hoặc doanh nghiệp muốn trải nghiệm và sử dụng các tính năng cơ bản của nền tảng này để phát hiện lỗ hổng bảo mật và đảm bảo tuân thủ các tiêu chuẩn bảo mật.

Chức năng chính:

+ Khám phá và kiểm kê tất cả IT assets
+ Quản lý vulnerabilities
+ Quét Web Applications

Giới thiệu các công cụ Scanning

4. Nessus

- Nessus là một công cụ quét lỗ hổng bảo mật được thiết kế chủ yếu để quét các máy chủ, hệ điều hành, và thiết bị mạng để tìm các lỗ hổng bảo mật.

Chức năng chính:

+ Phát hiện lỗ hổng trong hệ thống và ứng dụng mạng.

+Quét các cấu hình sai và thiết lập an ninh không an toàn.

+Tích hợp với các công cụ quản lý lỗ hổng khác để báo cáo và giám sát.

+ Có khả năng phát hiện các lỗ hổng phổ biến và phức tạp như Heartbleed, Shellshock, và SQL Injection.

Nội dung báo cáo

- Phần I: Ngữ cảnh mục tiêu đề tài
- Phần II: Phương pháp
- Phần III: Giới thiệu các công cụ Scanning
- Phần IV: Kết quả

Kết quả

Số lượng vulnerability được phát hiện bởi các tool scanner trong trường hợp có WAF và không WAF

Tool	Chưa có WAF	Có WAF
Acunetix	39	29
ZAP Proxy	34	31
Qualys	96	79
Nessus	58	62

Cám ơn thầy và các bạn đã lắng nghe!