**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**BÁO CÁO**

**MÔN: AN TOÀN VÀ BẢO MẬT HỆ THỐNG THÔNG TIN**

**ĐỀ TÀI: NGHIÊN CỨU VÀ TRIỂN KHAI HỆ THỐNG THU THẬP VÀ PHÂN TÍCH TẤN CÔNG VÀO ỨNG DỤNG WEB OWASP HONEYPOT PROJECT**

Sinh viên thực hiện:

* Lê Tuấn Khôi 0850080026
* Tô Sỹ Đạt 0850080065
* Võ Nguyễn Phi Hùng 0850080074
* Giảng viên hướng dẫn Th.S Phạm Trọng Huynh
* Lớp: 08\_CNPM

**TP. Hồ Chí Minh, Tháng 02 năm 2023**

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM**

**KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**



**BÁO CÁO**

**MÔN: AN TOÀN VÀ BẢO MẬT HỆ THỐNG THÔNG TIN**

**ĐỀ TÀI: NGHIÊN CỨU VÀ TRIỂN KHAI HỆ THỐNG THU THẬP VÀ PHÂN TÍCH TẤN CÔNG VÀO ỨNG DỤNG WEB OWASP HONEYPOT PROJECT**

Sinh viên thực hiện:

* Lê Tuấn Khôi 0850080026
* Tô Sỹ Đạt 0850080065
* Võ Nguyễn Phi Hùng 0850080074
* Giảng viên hướng dẫn Th.S Phạm Trọng Huynh
* Lớp: 08\_CNPM

**TP. Hồ Chí Minh, Tháng 02 năm 2023**

**NHẬN XÉT**



**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc132630868)

[**1.1.** **Tổng quan về đề tài:** 1](#_Toc132630869)

[**1.1.1.** **Lý do chọn đề tài** 1](#_Toc132630870)

[**1.1.2.** **Phạm vi nghiên cứu** 1](#_Toc132630871)

[**1.1.3.** **Tính thực tiễn** 2](#_Toc132630872)

[**1.1.4.** **Mục tiêu của đề tài** 3](#_Toc132630873)

[CHƯƠNG 2: TỔNG QUAN VỀ HONEYPOT VÀ OWASP HONEPOT PROJECT 4](#_Toc132630874)

[**2.1. Khái niệm Honeypot** 4](#_Toc132630875)

[**2.1.1. Phân loại Honeypot:** 8](#_Toc132630876)

[**2.1.2. Honeypot sản xuất và honeypot nghiên cứu** 10](#_Toc132630877)

[**2.1.2.1. Honeypot sản xuất** 10](#_Toc132630878)

[**2.1.2.2. Honeypot nghiên cứu** 10](#_Toc132630879)

[**2.1.3. Honeypot theo độ phức tạp** 11](#_Toc132630880)

[**2.1.4. Honeypot hoạt động như thế nào trong an ninh mạng** 12](#_Toc132630881)

[**2.1.5. Các loại Honeypot phổ biến** 12](#_Toc132630882)

[**2.1.6. Lợi ích và rủi ro khi sử dụng Honeypot an ninh mạng** 15](#_Toc132630883)

[**2.1.6.1. Lợi ích:** 15](#_Toc132630884)

[**2.1.6.2. Rủi ro:** 15](#_Toc132630885)

[**2.2. Khái niệm về OWASP Honeypot Project** 15](#_Toc132630886)

[**2.2.1. OWASP là gì:** 15](#_Toc132630887)

[**2.2.3. OWASP Honeypot Project là gì.** 16](#_Toc132630888)

[**2.2.3.1. Một số dự án của OWASP Honeypot Project** 17](#_Toc132630889)

[**2.2.4. Mục đích của OWASP Honeypot Project** 19](#_Toc132630890)

[**2.2.5. Các công cụ của OWASP Honeypot Project** 20](#_Toc132630891)

[**2.3. Các kỹ thuật tấn công ứng dụng web** 21](#_Toc132630892)

[**2.4. Phương pháp phòng tránh tấn công ứng dụng web** 29](#_Toc132630893)

[CHƯƠNG 3: CÀI ĐẶT THỰC NGHIỆM 31](#_Toc132630894)

[**3.1. Triển khai OWASP Honeypot Project** 31](#_Toc132630895)

[**3.2. Kết quả thu được** 36](#_Toc132630896)

[CHƯƠNG 4: KẾT LUẬN 38](#_Toc132630897)

[**4.1. Tóm lại** 38](#_Toc132630898)

[**4.2. Những việc thực hiện được** 38](#_Toc132630899)

[**4.3. Những việc chưa thực hiện được** 38](#_Toc132630900)

**DANH MỤC HÌNH**

[Hình 2. 1. Mô hình minh họa Honeypot 5](#_Toc132630903)

[Hình 2. 2. Mô hình hệ thống Honeypot 6](#_Toc132630904)

[Hình 2. 3. Các loại hình Honeypot 9](#_Toc132630905)

[Hình 2. 4. Low-Interaction Honeypot 13](#_Toc132630906)

[Hình 2. 5. High-Interaction Honeypot 14](#_Toc132630907)

[Hình 2. 6. Hybrid Honeypot 14](#_Toc132630908)

[Hình 2. 7. Logo OWASP 16](#_Toc132630909)

[Hình 2. 8. OWASP Honeypot Project 17](#_Toc132630910)

[Hình 2. 9. OWASP WebGoat 17](#_Toc132630911)

[Hình 2. 10. OWASP Honeyd 18](#_Toc132630912)

[Hình 2. 11. OWASP thug 18](#_Toc132630913)

[Hình 2. 12. OWASP Amun 19](#_Toc132630914)

[Hình 2. 13. OWASP Shockpot 19](#_Toc132630915)

[Hình 2. 14. Injection attack 21](#_Toc132630916)

[Hình 2. 15. Cross-Site Scripting (XSS) 22](#_Toc132630917)

[Hình 2. 16. Cross-Site Request Forgery (CSRF) 23](#_Toc132630918)

[Hình 2. 17. Broken Authentication and Session Management 24](#_Toc132630919)

[Hình 2. 18. Security Misconfiguration 25](#_Toc132630920)

[Hình 2. 19. Insecure Cryptographic Storage 26](#_Toc132630921)

[Hình 2. 20. Insufficient Transport Layer Protection 26](#_Toc132630922)

[Hình 2. 21. Unvalidated Input 27](#_Toc132630923)

[Hình 2. 22. File inclusion vulnerabilities 28](#_Toc132630924)

[Hình 2. 23. DDoS attacks 29](#_Toc132630925)

[Hình 3. 1. Hình ảnh cài xong elasticsearch 34](#_Toc132630926)

[Hình 3. 2. Docker 35](#_Toc132630927)

[Hình 3. 3. Giao diện của owasp honeypot project 36](#_Toc132630928)

[Hình 3. 4. Ip dest honeypot và ip dest network 37](#_Toc132630929)

[Hình 3. 5. Port dest honeypot và port dest net work 37](#_Toc132630930)

[Hình 3. 6. Log Explorer 37](#_Toc132630931)

**LỜI NÓI ĐẦU**

Tất cả các tổ chức và doanh nghiệp đều đang đối mặt với nguy cơ bị tấn công mạng. Những kẻ tấn công luôn tìm cách tìm kiếm điểm yếu trong hệ thống mạng để tấn công và chiếm quyền kiểm soát. Vì vậy, việc bảo vệ hệ thống mạng là một vấn đề quan trọng và cần được ưu tiên hàng đầu.

Trong quá trình tìm kiếm các giải pháp bảo mật hiệu quả, nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project là một trong những giải pháp được nhiều chuyên gia bảo mật đánh giá cao. OWASP Honetpot Project là một dự án mã nguồn mở cung cấp các giải pháp phát hiện và chống lại các cuộc tấn công mạng. Dự án này cho phép người dùng triển khai các máy chủ giả lập, được gọi là “Honeypot”, để thu thập thông tin về các cuộc tấn công và các chiến thuật tấn công.

Việc triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project giúp người dùng hiểu rõ hơn về các thức tấn công của các Hacker, đồng thời giúp họ nhanh chóng phát hiện ra các hành vi tấn công và các mối đe dọa mới.

Tuy nhiên việc triển khai hệ thống này không phải là điều dễ dàng và đòi hỏi nhiều kiến thức về bảo mật mạng và phát hiện tấn công. Để thực hiện thành công việc này, người dùng cần phải tìm hiểu và áp dụng các công cụ và phần mềm hỗ trợ để nâng cao hiệu quả của quá trình thu thập và phân tích tấn công.

Vì vậy, nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project là một trong những giải pháp bảo mật hiệu quả và được ưa chuộng hiện nay. Tuy nhiên, để đạt được hiệu quả cao nhất, người dùng cần phải đầu tư thời gian và nỗ lực để nghiên cứu và áp dụng các công cụ và phần mềm hỗ trợ, đồng thời cần phải hiểu rõ các chiến thuật và hành vi tấn công của Hacker. Điều này sẽ giúp họ xây dựng được các hiện pháp phòng thủ hiệu quả để bảo vệ hệ thống mạng của mình trước các cuộc tấn công

Ngoài ra, việc triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project cũng đòi hỏi sự chú ý đến việc bảo vệ dữ liệu và thông tin của người dùng. Các thông tin thu thập được từ các Honeypot có thể chứa những dữ liệu nhạy cảm về hệ thống và các thành phần của nó, vì vậy việc bỏa mật thông tin là rất quan trọng để tránh rò rỉ dữ liệu.

Trong lời nói đầu này, chúng em sẽ trình bày về quá trình nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project. Chúng em sẽ giải thích chi tiết về cách triển khai các Honeypot và cách thu thập dữ liệu từ chúng. Chúng em cũng sẽ thảo luận về các kỹ thuật phân tích tấn công và các công cụ hộ trợ để giúp người dùng hiểu rõ hơn về cách thức tấn công của Hacker.

Ngoài ra, chúng em sẽ đề cập đến các vấn đề liên quan đến bảo mật thông tin trong quá trình triển khai hệ thống này và đưa ra các giải pháp để giảm thiểu nguy cơ rò rỉ dữ liệu. Cuối cùng, chúng em sẽ trình bày về những kết quả đạt được từ những quá trình nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project, đồng thời đưa ra những khuyến nghị để cải thiện hiệu quả của quá trình này.

Chúng em hy vọng rằng thông qua lời nói đầu này, người đọc sẽ hiểu rõ hơn về tầm quan trọng của việc bảo vệ hệ thống mạng và sẽ được hỗ trợ để triển khai một hệ thống thu thập và phân tích tấn công hiệu quả sử dụng ứng dụng OWASP Honeypot Project.

**LỜI CẢM ƠN**

Lời đầu tiên trong quá trình học tập, nghiên cứu và hoàn thành đồ án này, chúng em đã nhận được những sự hỗ trợ và giải đáp các câu hỏi, thắc mắc, một vấn đề còn mơ hồ một cách tận tình từ các giảng viên bộ môn trong chuyên ngành. Chúng em đã tiếp thu và ghi nhận những đóng góp, kiến thức mới mẻ và hiệu quả để phục vụ cho quá trình nghiên cứu, thực hiện và hoàn thành đồ án một cách tối đa nhất. Chúng em xin chân thành cảm ơn sự nhiệt tình của giảng viên hướng dẫn và các giảng viên bộ môn.

Đặt biệt hơn nữa, nhóm chúng em xin gửi lời cảm ơn sâu sắc và chân thành đến thầy về sự đóng góp quan trọng của thầy trong quá trình học tập và nghiên cứu của nhóm chúng em về bảo mật hệ thống thông tin.

Thầy là một giảng viên rất tận tâm và am hiểu về lĩnh vực an toàn và bảo mật hệ thống thông tin. Những bài giảng của thầy luôn đầy đủ, chi tiết và cập nhật những kiến thức mới nhất trong ngành, giúp cho nhóm chúng em và các bạn sinh viên có thể nắm vững được kiến thức cơ bản và nâng cao kiến thức chuyên môn.

Không chỉ giảng dạy, thầy còn luôn hỗ trợ và tạo điều kiện cho sinh viên trong việc thực hành và nghiên cứu. Thầy luôn sẵn sàng giải đáp mọi thắc mắc và hỗ trợ sinh viên trong quá trình thực hiện các đề tài, đồ án và dự án nghiên cứu.

Với sự hướng dẫn tận tình của thầy, nhóm chúng em đã có cơ hội được nghiên cứu và thực hành về các công nghệ bảo mật mới nhất, đặc biệt là việc triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project.

Những kiến thức và kinh nghiệm này sẽ rất hữu ích cho nhóm chúng em trong công việc và sự nghiệp sau này.

Một lần nữa, nhóm chúng em xin chân thành cảm ơn thầy ThS.Phạm Trọng Huynh vì những điều tuyệt vời thầy đã mạng đến cho chúng em và mong rằng sẽ có thêm nhiều cơ hội được học hỏi và làm việc cùng thầy.

Trân trọng.

Nhóm 17

# **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

* 1. **Tổng quan về đề tài:**
     1. **Lý do chọn đề tài**

Với sự bùng nổ của cuộc cách mạng công nghiệp 4.0, các mối đe dọa an ninh mạng ngày càng trở nên phức tạp và tinh vi hơn. Trong khi đó, OWASP Honeypt Project là một dự án mã nguồn mở được thiết kế để giúp người dùng thu thập thông tin tấn công mạng và phân tích các chiến thuật tấn công mới nhất. Điều này giúp tăng cường khả năng phát hiện và ngăn chặn các cuộc tấn công trên hệ thống mạng của tổ chức.

Hơn nữa, việc nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project còn mang lại nhiều lợi ích khác cho nghiên cứu và giảng dạy an toàn thông tin. Nó giúp người dùng hiểu rõ hơn về các mối đe dọa an ninh mạng cũng như cung cấp các phương pháp và công cuuj để phát hiện và ngăn chặn các cuộc tấn công mạng. Ngoài ra, nghiên cứu này cũng giúp cải thiện khả năng đào tạo và huấn luyện các chuyên gia an toàn thông tin.

Chính vì lý do trên, nhóm chúng em quyết định lựa chọn để tài về nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project như một phần của đồ án nghiên cứu của nhóm.

* + 1. **Phạm vi nghiên cứu**

Phạm vi nghiên cứu chủ yếu xoay quanh các yếu tố sau:

* Tìm hiểu về OWASP Honeypot Project: Nghiên cứu các tài liệu liên quan đến OWASP Honeypot Project, bao gồm cách thức hoạt động của nó, các tính năng và ứng dụng của nó.
* Phát triển hệ thống thu thập dữ liệu tấn công: Thiết lập hệ thống thu thập dữ liệu tấn công vào ứng dụng OWASP Honeypot Project, bao gồm cách cài đặt và cấu hình.
* Phân tích dữ liệu thu thập được: Sử dụng các công cụ phân tích để xác định các đặc điểm của các cuộc tấn công, bao gồm các hành vi, kỹ thuật và mục tiêu của chúng.
* Đánh giá và cải thiện tính hiệu của OWASP Honeypot Project: Đánh giá tính hiệu quả của ứng dụng OWASP Honeypot Prooject trong việc phát hiện và phòng ngừa các cuộc tấn công, đồng thời đề xuất các cải tiến để nâng cao tính hiệu quả của nó.

Từ những nhiệm vụ trên, phạm vi nghiên cứu sẽ xoay quanh việc phân tích tấn công vào ứng dụng OWASP Honeypot Project và đề xuất các cải tiến để nâng cao tính hiệu quả của nó.

* + 1. **Tính thực tiễn**

Hiện nay, việc bảo vệ an toàn thông tin đã trở thành một trong những vấn đề cấp bách và quan trọng đối với các tổ chức, doanh nghiệp và cá nhân. Với sự phát triển không ngừng của công nghệ thông tin,các mối đe dọa mạng ngày càng tinh vi và đa dạng hơn. Chính vì vậy, việc nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng là một trong những giải pháp hiệu quả để bảo vệ hệ thống thông tin khỏi các cuộc tấn công mạng.

Honeypot là một công cụ quan trọng trong việc thu thập dữ liệu về các cuộc tấn công mạng. OWASP Honeypot Project là một dự án mã nguồn mở được xây dựng để cung cấp cho người dùng các Honeupot chất lượng cao để giám sát và phát hiện các cuộc tấn công mạng. Việc nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công mạng. Việc nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project không chỉ giúp cho các chuyên gia an ninh mạng có thể tiếp cận với các dữ liệu tấn công quan trọng mà còn giúp các nhà phát triển mạng hiểu rõ hơn về các mối đe dọa mạng, từ đó phát triển các giải pháp bảo mật hiệu quả hơn.

Do đó, chủ đề nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project có tính thực tiễn cao và cần được quan tâm trong việc bảo vệ an toàn thông tin mạng.

* + 1. **Mục tiêu của đề tài**

Các mục tiêu bao gồm:

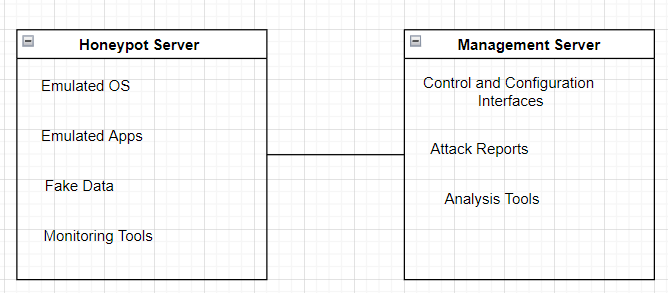
* Nghiên cứu về hệ thống Honeypot: Tìm hiểu về hệ thống Honeypot, các thức hoạt động và các loại Honeypot khác nhau có thể triển khai và sử dụng hiệu quả.
* Triển khai hệ thống Honeypot: Xây dựng một hệ thống Honeypot thông qua việc triển khai và cấu hình một số Honeypot phổ biến của OWASP Honeypot Project như Cowrie, Dionaea hoặc Glastopf.
* Thu thập dữ liệu tấn công: Sử dụng hệ thống Honeypot triển khai để thu thập dữ liệu tấn công từ các hacker và tổng hợp chúng để phân tích.
* Phân tích dữ liệu tấn công: Áp dụng các phương pháp và công cụ phân tích dữ liệu để tìm hiểu và đánh giá cách thức tấn công của Hacker, từ đó tìm ra những lỗ hổng bảo mật của hệ thống thông tin.
* Đề xuất các biện pháp bảo mật: Dựa trên những kết quả phân tích, đề xuất và triển khai các biện pháp bảo mật phù hợp để tăng cường độ an toàn cho hệ thống thông tin.

# **CHƯƠNG 2: TỔNG QUAN VỀ HONEYPOT VÀ OWASP HONEPOT PROJECT**

## **2.1. Khái niệm Honeypot**

Honeypot là một công nghệ mới với tiềm năng khổng lồ cho cộng đồng bảo mật. Định nghĩa đầu tiên được đưa ra đầu tiền bởi một vài biểu tượng về bảo mật máy tính, cụ thể là Cliff Stoll trong cuốn sách “The Cuckoo’s Egg” và trong bài báo của Bill Cheswick. Từ đó, Honeypot tiếp tục được phát triển với những công cụ bảo mật mạnh mẽ mà chũng ta biết cho đến nay. Thuật ngữ “Honeypot” được nhắc đến lần đầu tiên vào ngày 4 tháng 8 năm 1999 trong bài báo “To Buil a Honeypot” của tác giả Lance Spitzner – một trong những người đứng ra thành lập dự án Honeynet ( Honeynet Project ), giới thiệu về ý tưởng xây dựng hệ thống Honeynet nhằm mục đích nghiên cứu các kỹ thuật tấn công của Hacker; từ đó, có biện pháp ngăn chặn tấn công kịp thời. Và tháng 6 năm 2000, dự án Honeynet được thành lập bởi 30 chuyên gia an ninh mạng ở các Công ty bảo mật như: Foundstone, Security Focus, Source Fre, …., tình nguyện tham gia nghiên cứu phi lợi nhuận. 2 Dự án Honeynet được triển khai ở 8 quốc gia ( Mỹ, Ấn Độ, Hy Lạp,…) với 12 trạm Honeynet, bao gồm 24 hệ thống Unix và 19 hệ thống Linux, cùng với một số hệ thống khác như : Suse 6.3, Suse 7.1,Window,… Bước đầu tiên để hiểu được Honeypot thì trước hết phải hiểu Honeypot là cái gì?Nó không giống như firewall, hay hệ thống IDS, Honeypot không giải quyết cụ thể một vấn đề nào đó. Thay vào đó, nó là một công cụ rất linh hoạt trong đó có nhiều hình dạng và kích cỡ. Nó có thể làm tất cả mọi thứ từ phát hiện các cuộc tấn công mã hóa trong các mạng IPv6. Sự linh hoạt này cung cấp một sức mạnh thực sự cho Honeypot. Nó cũng là sự hỗn hợp làm cho kẻ tấn công khó xác định và hiểu. Honeypot là một hệ thống tài nguyên thông tin được xây dựng với mục đích giả dạng đánh lừa những kẻ sử dụng và xâm nhập không hợp pháp, thu hút sự chú ý của chúng, ngăn không cho chúng tiếp xúc với hệ thống thật. Honeypot có thể được xem như “Mắt ong”; và tất nhiên là Honeypot cũng có phải có “Mật ngọt” – tức là có chứa các Hệ thống tài nguyên thông tin có giá trị, nhạy cảm, có tính bí mật như : thông tin về chứng khoán, thông tin tài khoản ở các ngân hàng, thông tin bí mật an ninh quốc gia…., để làm “mồi” dụ Hacker chú ý đến tấn công. Hệ thống tài nguyên thông tin có nghĩa là Honeypot có thể giả dạng bất cứ loại máy chủ tài nguyên nào như là Mail Server, Domain Name Server, Web Server…, được cài đặt chạy trên bất cứ Hệ điều hành nào như: Linux ( Red hat, Fedora…), Unix( Solaris), Window ( Window NT, Window 2000, Window XP, Window 2003, Vista,…..), ….Honeypot sẽ trực tiếp tương tác với tin tặc và tìm cách khai thác thông tin về tin tặc như hình thức tấn công, công cụ tấn công hay cách thức tiến hành tấn thay vì bị tấn công.

**Mô hình minh họa Honeypot:**

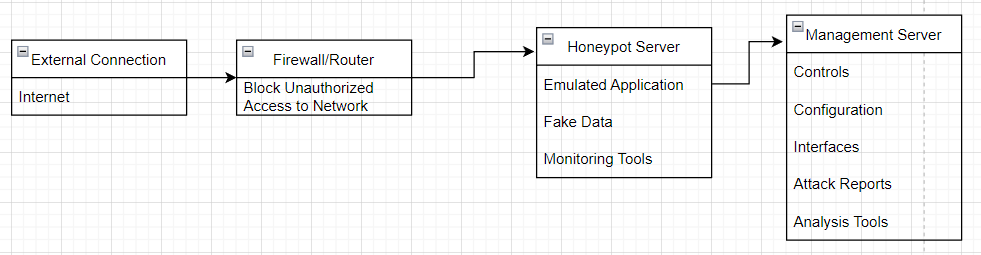
****

Hình 2. 1. Mô hình minh họa Honeypot

Trong mô hình này, Honeypot Server là một máy chủ hoặc một thiết bị được cấu hình để giả vờ là một hệ thống thực sự. Nó chứa một số phần mềm giả lập để tạo ra các ứng dụng, dữ liệu và các yếu tố mạng để tạo ra một môi trường nhắm mục tiêu cho kẻ tấn công. Nó cũng chứa các công cụ giám sát để phát hiện các cuộc tấn công.

Management Server là trung tâm điều khiển và cấu hình cho Honeypot Server. Nó cung cấp các giao diện điều khiển và cấu hình cho người quản trị, cũng như các công cụ phân tích để phân tích các cuộc tấn công và báo cáo lại. Các cuộc tấn công được phát hiện bởi Honeypot Server được báo cáo về Management Server để phân tích và đưa ra các biện pháp phòng ngừa.

**Mô hình hệ thống Honeypot**

****

Hình 2. 2. Mô hình hệ thống Honeypot

Mô hình này bao gồm các thành phần sau:

* External Connection: đại diện cho kết nối đến Internet.
* Firewall/Router: đảm bảo chỉ các yêu cầu được cho phép được chuyển tiếp đến mạng.
* Honeypot Server: đại diện cho honeypot, nơi các cuộc tấn công được phát hiện và ghi nhận.
* Management Server: đại diện cho trung tâm quản lý và giám sát honeypot, cung cấp các giao diện điều khiển và cấu hình cho người quản trị, cũng như các công cụ phân tích để phân tích các cuộc tấn công và báo cáo lại.

Các yêu cầu đến từ Internet sẽ được chuyển đến Firewall/Router để xác định xem liệu chúng có được phép truy cập vào mạng hay không. Nếu được phép, các yêu cầu này sẽ được chuyển đến Honeypot Server. Honeypot Server sẽ giả vờ là một hệ thống thực sự và thu thập thông tin về các cuộc tấn công và các kẻ tấn công, và báo cáo về chúng cho Management Server. Management Server sẽ giám sát honeypot và thu thập thông tin từ nó, cũng như quản lý và điều khiển honeypot để tăng tính hiệu quả của hệ thống.

**Ưu điểm của Honeypot:** Honeypot là một khái niệm rất đơn giản, trong đso cung cấp một số đặc điểm mạnh mẽ.

* Dữ liệu nhỏ được đặt giá trị cao: Honeypot thu thập một lượng nhỏ thông tin. Thay vì đăng nhập một GB dữ liệu một ngày, họ chỉ phải đăng nhập một MB dữ liệu một ngày. Thay vì tạo ra 10.000 cảnh báo mỗi ngày, nó có thể chỉ tạo 10 thông báo mỗi ngày. Hãy nhớ rằng, Honeypot chỉ nắm bắt các hành động xấu, bất kỳ sự tương tác với Honeypot như không xác thực hay các hành động độc 3 hại. Như vậy, Honeypot đã giảm thiểu được “tiếng ồn”, có nghĩ là với bộ thu thập dữ liệu nhỏ, nhưng thông tin có giá trị cao, nhưng đó chỉ là những hành động xấu. Điều này có nghĩa là sẽ dễ dàng hơn nhiều để phân tích các dữ liệu mà Honeypot thu thập và lấy được giá trị từ nó.
* Công cụ và chiến thuật mới: Honeypots được thiết kế để nắm bắt tất cả những gì được tương tác vào nó, bao gồm các công cụ, chiến thuật không bao giờ thấy trước.
* Nguồn lực tối thiểu: Honeypots yêu cầu nguồn lực tối thiểu, nó chỉ nắm bắt các hoạt động xấu. Điều này có nghĩa là một máy tính 128MB bộ nhớ RAM có thể dễ dàng xử lý một mạng lớp B toàn bộ ngồi một mạng OC-12.
* Mã hóa hay IPv6: Không giống như hầu hết các công nghệ bảo mật( như hệ thống IDS) các Honeypots làm việc tốt trong môi trường mã hóa hay IPv6. Nó không phân biệt những điều gì tương tác với nó. Nó chỉ nắm bắt các hành động xấu.
* Thông tin: Honeypots có thể thu thập một vài thông tin chi tiết.
* Honeypots là công nghệ đơng giản, ít có nhưng sai lầm hoặc cấu hình sai

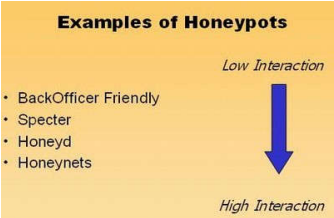
**Nhược điểm của Honeypot:** Giống như nhiều công nghệ, các Honeypots cũng có những yếu điểm. Đó là do chúng không thể thay thế các công nghệ hiện đại, nhưng làm việc với các công nghệ hiện có.

* Hạn chế View: Honeypots chỉ có thể theo dõi và nắm bắt hoạt động trực tiếp tương tác với họ. Honeypots sẽ không nắm bắt các cuộc tấn công chống lại các hệ thống khác, trừ khi kẻ tấn công hoặc đe dọa tương tác với các honeypots.
* Rủi ro: Tất cả các công nghệ bảo mật đều có nguy cơ. Tường lửa có nguy cơ bị xâm nhập, mã hóa có nguy cơ bị phá vỡ, các cảm biến IDS có nguy cơ không phát hiện các cuộc tấn công. Honeypots cũng không phải là trường hợp khác, honeypots có nguy cơ được thực hiện trên của kẻ xấu và được sử dụng để gây tổn hại cho các hệ thống khác. Có rất nhiều nguy cơ khác nhau dẫn đến sự khác nhau của Honeypots.
* Tiêu tốn tài nguyên: Honeypot cần phải sử dụng một lượng tài nguyên nhất định để giả mạo hệ thống thật, điều này có thể ảnh hưởng đến hiệu suất của hệ thống.
* Không thể ngăn chặn tấn công: Honeypot không thể ngăn chặn các cuộc tấn công như các công nghệ bảo mật truyền thống, nnos chỉ có thể giúp phát hiện và thu thập thông tin về các cuộc tấn công.
* Rủi ro về bảo mật: nếu không được cấu hình và quản lý đúng cách, Honeypot có thể tạo ra rủi ro bảo mật cho hệ thống, ví dụ như bị tấn công và chiếm quyền kiểm soát.

### **2.1.1. Phân loại Honeypot:**

Honeypot được chia làm hai loại chính: Tương tác thấp và tương tác cao

* Tương tác thấp: Honeypot chỉ cài đặt chương trình (chẳng hạn như: Honeyd, BackOfficer Friendly, Specter,) mô phỏng giả các dịch vụ, ứng dụng, và hệ điều hành. Loại này có mức độ rủi ro thấp, dễ triển khai và bảo dưỡng nhưng lại bị giới hạn về dịch vụ.
* Tương tác cao: Honeypot được cài đặt, chạy các dịch vụ, ứng dụng và hệ điều hành thực ( Chẳng hạn như Honeynet ). Loại này có mức độ thông tin thu thập được cao nhưng mức độ rủi ro cao và tốn thời gian để vận hành và bảo dưỡng.



Hình 2. 3. Các loại hình Honeypot

Một số ví dụ về các loại Honeypot

* BackOfficer Friendly (BOF): là một loại hình Honeypot rất dễ vận hành và cấu hình và có thể hoạt động trên bất kì phiên bản nào của Windows và Unix nhưng nhược điểm của nó là chỉ tương tác được với một số dịch vụ đơn giản như FTP, Telnet, SMTP…
* Specter: đây cũng là loại hình Honeypot tương tác thấp nhưng có khả năng tương tác tốt hơn so BackOfficer, loại Honeypot này có thể giả lập trên 14 cổng ( Port ); và có thể cảnh báo, quản lý từ xa. Tuy nhiên, cũng giống như BackOfficer thì Specter có nhược điểm là bị giới hạn số dịch vụ và không linh hoạt.
* Honeyd:
  + Loại Honeypot này có thể lắng nghe trên tất cả các cổng TCP và UDP, những dịch vụ mô phỏng được thiết kế với mục đích ngăn chặn và ghi lại những cuộc tấn công, tương tác với kẻ tấn công trong vai trò là một hệ thống nạn nhân.
  + Hiện nay, Honeyd có nhiều phiên bản và có thể mô phỏng được khoảng 473 hệ điều hành.
  + Honeyd là loại hình Honeypot tương tác thấp có nhiều ưu điểm tuy nhiên Honeyd có nhược điểm là không thể cung cấp một hệ điều hành thật để tương tác với tin tặc và không có cơ chế cảnh báo khi phát hiện hệ thống bị xâm nhập hoặc gặp phải nguy hiểm.

### **2.1.2. Honeypot sản xuất và honeypot nghiên cứu**

### **2.1.2.1. Honeypot sản xuất**

Honeypot sản xuấy (production honeypot) là một loại honeypot được triển khai trong mội trường sản xuất thực tế nhằm mục đích thu thập thông tin và phát hiện tấn công một các chính xác và hiệu quả. Khác với các loại Honeypot khác như research Honeypot hay low\_interantion honeypot, honeypot sản xuất được đặt trong môi trường sản xuất thực tế và hoạt động giống như các hệ thống thực tế, với các chức năng và tính năng giống nhau. Sự khác biệt chủ yếu nằm ở việc Honeypot sản xuất được đặt trong môi trường được sử dụng bởi người dùng thật sự, với các ứng dụng và dữ liệu thật sự, trong khi các Honeypot khác thường được triển khai trong môi trường ảo hoặc giả lập với các thông tin giả tạo.

Honeypot sản xuất có thể được triển khai để giảm sát các kết nối mạng, phát hiện tấn công và chống lại các hành vi xâm phạm và phá hoại. Nó có thể được sử dụng để bảo vệ các hệ thống quan trọng, dữ liệu và ứng dụng trên các môi trường sản xuất, đồng thời cung cấp các thông tin cần thiết cho những người quản trị hệ thống và nhà nghiên cứu bảo mật trong quá trình phát hiện và phân tích các mối đe dọa mạng.

### **2.1.2.2. Honeypot nghiên cứu**

Honeypot nghiên cứu là các honeypot được sử dụng để nghiên cứu và phát triển các phương pháp phiingf chống tấn công mạng và bảo mật thông tin.

Các Honeypot này được cài đặt và cấu hình để thu thập dữ liệu tấn công và phân tích chúng để biểu rõ hơn về cách thức tấn công, phương pháp tấn công, các lỗ hổng bảo mật được tấn công và các mối đe dọa an ninh mạng khác. Dữ liệu thu thập từ Honeypot nghiên cứu được sử dụng để nghiên cứu và phát triền các giải pháp bảo mật, phân tích động lực tấn công mạng và đào tao nhân viên bảo mật.

### **2.1.3. Honeypot theo độ phức tạp**

Honeypot là một phương pháp giám sát bảo mật được sử dụng để thu thập thông tin về các cuộc tấn công mạng bằng cách tạo ra một môi trường giả lập hay một điểm yếu giả để lừa hacker tấn công. Theo đó, honeypot không phải là một thuật toán hay một giải thuật mà là một phương pháp hoặc một kỹ thuật được áp dụng trong lĩnh vực bảo mật mạng.

Tuy nhiên, việc triển khai và quản lý một honeypot có thể gặp phải những vấn đề liên quan đến độ phức tạp, bao gồm:

* **Cài đặt và cấu hình honeypot:** đòi hỏi kiến thức sâu về hệ điều hành, mạng máy tính và bảo mật mạng để có thể thiết lập honeypot một cách hiệu quả và an toàn.
* Theo dõi và phân tích dữ liệu: honeypot thu thập thông tin về các cuộc tấn công mạng, do đó việc phân tích dữ liệu thu được là rất quan trọng để có thể phát hiện các mối đe dọa và tấn công mới.
* **Bảo mật và giám sát honeypot:** honeypot cũng có thể trở thành điểm yếu của hệ thống nếu không được bảo vệ và giám sát cẩn thận. Việc quản lý honeypot đòi hỏi kỹ năng và kinh nghiệm để có thể đảm bảo an toàn cho hệ thống mạng.
* **Honeypot tương tác thấp:** Honeypot tương tác thấp sử dụng tương đối ít tài nguyên và thu thập thông tin cơ bản về kẻ tấn công. Loại này dễ bị phát hiện,dễ lợi dụng và chỉ cung cấp thông tin tình báo hạn chế về kẻ thù vì chúng khá đơn giản, không có khả năng thu hút sự chú ý của kẻ tấn công trong thời gian dài.
* **Honeypot tương tác cao:** Honeypot có tính tương tác cao được thiết kế để thu hút tội phạm mạng trong thời gian dài thông qua mạng lưới các mục tiêu khám phá. Giúp nhóm an ninh mạng hiểu sâu hơn về cách thức hoạt động của những kẻ thù này. Hầu hết các honeypot nghiên cứu đều được coi là các honeypot tương tác cao.
* **Công nghệ lừa đảo:** Kỹ thuật bảo mật này áp dụng tự động hóa thông minh ( trí tuệ nhân tạo (AI), máy học (ML) và các công nghệ tiên tiến dựa trên dữ liệu khác) để tự động hóa việc thu thập và phân tích dữ liệu . Với công nghệ này tổ chức xử lý thông tin nhanh hơn và mở rộng nỗ lực trên một môi trường đánh lừa phức tạp hơn.

Tóm lại, honeypot không phải là một thuật toán hay giải thuật, nhưng triển khai và quản lý honeypot có thể đòi hỏi kiến thức và kỹ năng chuyên môn để đảm bảo hiệu quả và an toàn cho hệ thống mạng.

### **2.1.4. Honeypot hoạt động như thế nào trong an ninh mạng**

Honeypot là một kỹ thuật được sử dụng trong lĩnh vực bảo mật mạng để giả lập một hệ thống mục tiêu dùng để thu thập thông tin về các cuộc tấn công và những người tấn công. Nó là một máy chủ, chương trình hoặc mạng mà được cài đặt để nhận và ghi lại các hoạt động tấn công.

Khi một kẻ tấn công tìm kiếm các hệ thống để tấn công, họ có thể phát hiện và nhắm mục tiêu một honeypot. Khi kẻ tấn công xâm nhập vào honeypot, họ sẽ nghĩ rằng họ đã tìm thấy một hệ thống dễ bị tấn công. Trong thực tế, honeypot được thiết lập để theo dõi và ghi lại các hoạt động của kẻ tấn công.

Các honeypot có thể được cài đặt để giả vờ như là một máy chủ web, máy chủ email hoặc bất kỳ loại hệ thống nào khác mà kẻ tấn công có thể muốn tấn công. Honeypot cũng có thể được cài đặt để giả vờ là các hệ thống khác nhau với các cấu hình khác nhau để thu thập thông tin về các loại tấn công khác nhau.

Bẫy honeypot có thể được sản xuất để trông giống như một cổng thanh toán, là mục tiêu phổ biến của tin tặc. Honeypot hoặc honeynet cũng thu hút các tác nhân quan tâm đến việc thu thập tài sản trí tuệ (IP), bí mật thương mại hoặc thông tin nhạy cảm có giá trị khác.

Khi đã ở trong mạng, có thể theo dõi chuyển động của tội phạm mạng để hiểu rõ hơn về phương pháp và động cơ của chúng. Điều này sẽ giúp tổ chức điều chỉnh các giao thức bảo mật hiện có để ngăn chặn các cuộc tấn công tương tự vào các mục tiêu hợp pháp trong tương lai.

Để làm cho các honeypots hấp dẫn hơn, chúng thường chứa các lỗ hổng bảo mật có chủ ý nhưng không nhất thiết phải rõ ràng. Một mạng không được bảo mật đầy đủ sẽ khó có thể lừa được kẻ thù tinh vi mà kẻ xấu còn có thể cung cấp thông tin sai lệch hoặc thao túng môi trường để làm giảm hiệu quả của công cụ.

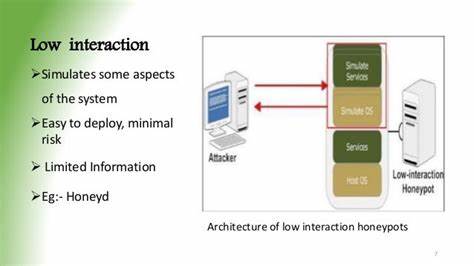
Mục đích của honeypot là để giúp các chuyên gia bảo mật hiểu rõ hơn về các chiến thuật và phương pháp tấn công của các kẻ xâm nhập, từ đó đưa ra các biện pháp bảo vệ và phòng chống tấn công cho các hệ thống mạng thực sự.

### **2.1.5. Các loại Honeypot phổ biến**

Có ba loại honeypot chính là low-interaction honeypot, high-interaction honeypot và hybrid honeypot. Dưới đây là khái niệm từng loại honeypot:

**Low-Interaction Honeypot:**

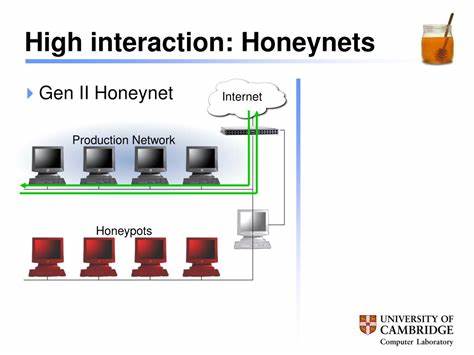
Low-interaction honeypot là loại honeypot nhằm mô phỏng một số dịch vụ mạng, chẳng hạn như SMTP, HTTP, FTP, Telnet, và DNS. Tuy nhiên, các dịch vụ này chỉ được triển khai đủ để mô phỏng mà không cung cấp các chức năng đầy đủ của dịch vụ. Như vậy, loại honeypot này ít tiêu tốn tài nguyên hơn so với các loại honeypot khác và đồng thời cũng ít có khả năng bị tấn công.



Hình 2. 4. Low-Interaction Honeypot

**High-Interaction Honeypot:**

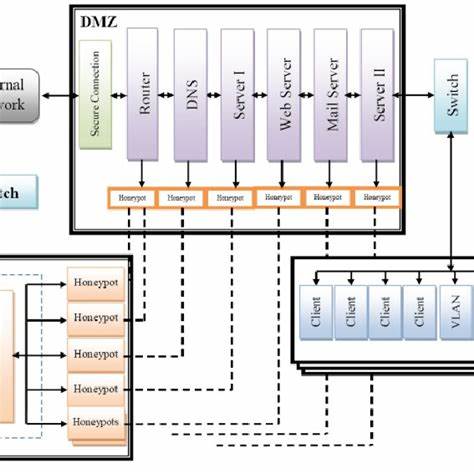
High-interaction honeypot cung cấp môi trường mô phỏng hệ thống thực tế, với các dịch vụ và ứng dụng đầy đủ và đầy đủ tính năng. Điều này làm cho honeypot này trông giống với hệ thống thực sự, cho phép kẻ tấn công tương tác với honeypot, làm cho honeypot trở nên nguy hiểm và đáng tin cậy trong việc thu thập thông tin tấn công. Tuy nhiên, honeypot này cần phải sử dụng nhiều tài nguyên và có nguy cơ bị tấn công thật sự.



Hình 2. 5. High-Interaction Honeypot

**Hybrid Honeypot:**

Hybrid honeypot là một loại honeypot kết hợp giữa low-interaction và high-interaction honeypot. Loại honeypot này cung cấp các dịch vụ và ứng dụng đầy đủ như high-interaction honeypot, nhưng các dịch vụ này được triển khai với các giả định, giúp tiết kiệm tài nguyên. Ngoài ra, loại honeypot này còn cho phép chuyển đổi giữa các cấp độ tương tác để đáp ứng các yêu cầu khác nhau của honeypot.



Hình 2. 6. Hybrid Honeypot

Tóm lại, mỗi loại honeypot đều có ưu điểm và nhược điểm riêng, người quản trị cần phải tùy chọn loại honeypot phù hợp với mục đích sử dụng và tài nguyên có sẵn của họ.

### **2.1.6. Lợi ích và rủi ro khi sử dụng Honeypot an ninh mạng**

### **2.1.6.1. Lợi ích:**

Thu thập thông tin: Honeypot giúp thu thập thông tin về các cuộc tấn công và các kẻ tấn công, từ đó giúp các chuyên gia bảo mật tìm ra những mối đe dọa mới và phát triển các biện pháp bảo vệ phù hợp.

Giảm thiểu thiệt hại: Honeypot giúp giảm thiểu thiệt hại do các cuộc tấn công. Khi kẻ tấn công chọn honeypot làm mục tiêu, họ sẽ không thể tấn công vào hệ thống mạng thực sự của bạn.

Điều tra tấn công: Honeypot giúp phát hiện các cuộc tấn công và cung cấp thông tin cần thiết để điều tra các cuộc tấn công này.

Kiểm tra khả năng bảo mật: Honeypot giúp kiểm tra khả năng bảo mật của các hệ thống mạng của bạn. Khi bạn giả lập một honeypot và đặt nó trong cùng một mạng với các hệ thống mạng thực sự của bạn, bạn có thể theo dõi các cuộc tấn công và đánh giá khả năng bảo mật của các hệ thống này.

### **2.1.6.2. Rủi ro:**

Có thể gây khó khăn trong việc phát hiện các cuộc tấn công thực sự: Nếu honeypot được thiết lập không đúng cách, nó có thể gây khó khăn trong việc phát hiện các cuộc tấn công thực sự đang diễn ra trên các hệ thống mạng thực sự.

Có thể trở thành mục tiêu của kẻ tấn công: Nếu honeypot không được bảo vệ tốt, nó có thể trở thành mục tiêu của các kẻ tấn công, từ đó gây ra các rủi ro an ninh mới.

Tốn kém chi phí: Việc thiết lập và duy trì honeypot có thể tốn kém chi phí, bao gồm cả chi phí cài đặt phần mềm, phần cứng và thời gian vàng để quản lý và duy trì honeypot.

## **2.2. Khái niệm về OWASP Honeypot Project**

### **2.2.1. OWASP là gì:**

OWASP là viết tắt của OPEN WEB APPLICATION SECURITY PROJECT, một tổ chức phi lợi nhuận chuyên về an ninh ứng dụng web. OWASP phát triển các công cụ và tài liệu giúp đảm bảo an toàn cho các ứng dụng web.



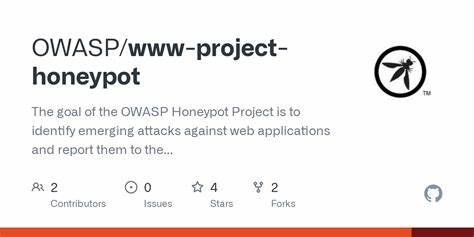
Hình 2. 7. Logo OWASP

### **2.2.3. OWASP Honeypot Project là gì.**

OWASP Honeypot Project là một dự án thuộc OWASP (Open Web Application Security Project) nhằm phát triển và giới thiệu các honeypot cho cộng đồng. Các honeypot này được tạo ra để giúp giám sát, phát hiện và phân tích các cuộc tấn công mạng đối với các ứng dụng web. Dự án cũng cung cấp các công cụ để phân tích, giám sát và phát hiện các cuộc tấn công mạng.

Mục tiêu của dự án là tạo ra các honeypot chất lượng cao để thu thập thông tin về các mối đe dọa mới và các kỹ thuật tấn công mới nhất. Thông tin này sau đó có thể được chia sẻ với cộng đồng để giúp cải thiện khả năng phát hiện và ngăn chặn các cuộc tấn công.

Dự án OWASP Honeypot Project hiện có ba honeypot chính là OWASP WebGoat, OWASP Honeyd và OWASP Thug, được sử dụng để tạo ra các môi trường mô phỏng mạng máy tính và các lỗ hổng bảo mật giả để thu hút sự chú ý của kẻ tấn công và giám sát các hoạt động của chúng. Ngoài ra, dự án cũng cung cấp các công cụ để phân tích các bản ghi honeypot, đồng thời phát triển các mô hình để giám sát các cuộc tấn công mạng và thu thập thông tin về các mối đe dọa mới.



Hình 2. 8. OWASP Honeypot Project

### **2.2.3.1. Một số dự án của OWASP Honeypot Project**

* **OWASP WebGoat:**

Là một honeypot cho các cuộc tấn công mạng vào ứng dụng web. Nó cung cấp các bài tập để giúp người dùng học cách phát hiện và ngăn chặn các cuộc tấn công mạng vào ứng dụng web.



Hình 2. 9. OWASP WebGoat

* **OWASP Honeyd:**

Là một honeypot có khả năng giả lập các máy tính và các hệ thống mạng. Nó được sử dụng để tạo ra các honeypot mạng và thu thập thông tin về các cuộc tấn công.



Hình 2. 10. OWASP Honeyd

* **OWASP Thug:**

Là một honeypot cho các cuộc tấn công mạng vào trình duyệt web. Nó được sử dụng để phát hiện các cuộc tấn công và giúp cải thiện khả năng phát hiện của các công cụ bảo mật.



Hình 2. 11. OWASP thug

* **OWASP Amun:**

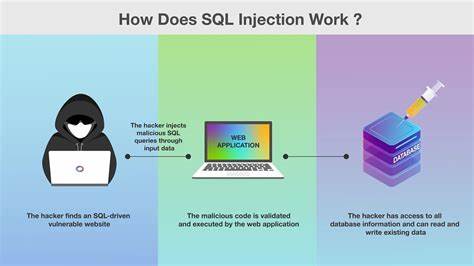
Là một honeypot cho các cuộc tấn công mạng vào các ứng dụng web. Nó cung cấp các tính năng như giám sát và phát hiện các cuộc tấn công, ghi lại các hoạt động của kẻ tấn công và tạo ra các báo cáo chi tiết về các hoạt động này.



Hình 2. 12. OWASP Amun

* **OWASP Shockpot:**

Là một honeypot cho các cuộc tấn công mạng vào các hệ thống điều khiển công nghiệp. Nó được sử dụng để tạo ra các honeypot cho các hệ thống SCADA và giúp người dùng phát hiện và phòng ngừa các cuộc tấn công vào hệ thống này.



Hình 2. 13. OWASP Shockpot

### **2.2.4. Mục đích của OWASP Honeypot Project**

Mục đích chính của dự án OWASP Honeypot Project là tạo ra các công cụ honeypot chất lượng cao để giúp người dùng tăng cường bảo mật mạng của họ. Các công cụ honeypot này giúp thu thập thông tin về các cuộc tấn công mạng, giúp người dùng phát hiện và phòng ngừa các cuộc tấn công mạng trước khi chúng gây ra thiệt hại nghiêm trọng.

Ngoài ra, dự án cũng tập trung vào việc phát triển các kỹ thuật và công cụ phân tích log để giúp người dùng hiểu rõ hơn về các hoạt động của kẻ tấn công và tăng cường khả năng phát hiện các mối đe dọa trong tương lai.

Một mục đích khác của dự án là giáo dục và tăng cường nhận thức về bảo mật mạng. Dự án cung cấp các tài liệu, hướng dẫn và khóa học để giúp người dùng hiểu rõ hơn về các cuộc tấn công mạng và cách bảo vệ mạng của mình.

### **2.2.5. Các công cụ của OWASP Honeypot Project**

Dự án OWASP Honeypot Project bao gồm nhiều công cụ honeypot được phát triển để giúp người dùng tăng cường bảo mật mạng của họ. Dưới đây là một số công cụ nổi bật của dự án:

* Honeyd: Là một dự án honeypot nổi tiếng của OWASP Honeypot Project. Honeyd cho phép người dùng tạo ra nhiều máy chủ ảo với các hệ điều hành khác nhau để mô phỏng các hệ thống thực tế. Honeyd có thể thu thập thông tin về các cuộc tấn công mạng và phân tích log để tìm ra các mối đe dọa.
* Thug: Là một honeypot hướng đến các cuộc tấn công từ mã độc JavaScript. Thug có khả năng mô phỏng trình duyệt web để thu thập thông tin về các cuộc tấn công và phân tích các tập tin JavaScript độc hại.
* Glastopf: Là một honeypot web được thiết kế để giả lập các lỗ hổng web phổ biến để thu thập thông tin về các cuộc tấn công. Glastopf sử dụng các kỹ thuật giả lập để tạo ra các trang web giả mạo và ghi lại các cuộc tấn công vào chúng.
* Shockpot: Là một honeypot chuyên về các cuộc tấn công mạng từ phía khách hàng. Shockpot giả lập các trang web và dịch vụ khách hàng để thu thập thông tin về các cuộc tấn công từ các máy tính khách.
* Wordpot: Là một honeypot dành cho các cuộc tấn công mạng liên quan đến WordPress. Wordpot giả lập các trang web WordPress giả mạo để thu thập thông tin về các cuộc tấn công và phân tích các plugin và theme của WordPress để tìm ra các lỗ hổng bảo mật.

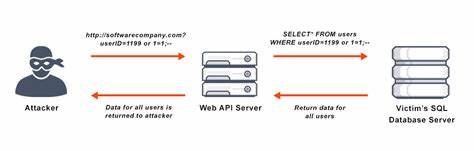
Ngoài ra, OWASP Honeypot Project còn có nhiều công cụ khác như Omnivore, Honeytrap, Conpot, Cowrie, Snare, Capture-HPC, ... để giúp người dùng tăng cường bảo mật mạng của họ.

## **2.3. Các kỹ thuật tấn công ứng dụng web**

**Injection**

**Injection attack** xảy ra khi dữ liệu không đáng tin cậy được gửi đến trình thông dịch mã (code interpreter) thông qua việc điền các form (biểu mẫu) hoặc một số dữ liệu khác gửi đến ứng dụng web. Ví dụ, kẻ tấn công có thể nhập SQL database code vào một biểu mẫu yêu cầu username ở dạng plaintext. Nếu việc điền các biểu mẫu đó không được bảo mật đúng cách, điều này sẽ dẫn đến việc SQL code đó được thực thi. Đây được gọi là một cuộc tấn công **SQL injection**.

Các cuộc tấn công **injection** có thể được ngăn chặn bằng cách xác thực và / hoặc “khử trùng” dữ liệu do người dùng gửi. (*Xác thực* nghĩa là từ chối các dữ liệu đáng ngờ, trong khi “khử trùng” nghĩa là làm sạch các phần dữ liệu có vẻ đáng ngờ.) Ngoài ra, quản trị viên cơ sở dữ liệu có thể thiết lập các biện pháp kiểm soát để giảm thiểu lượng thông tin bị lộ ra qua một cuộc tấn công injection.



Hình 2. 14. Injection attack

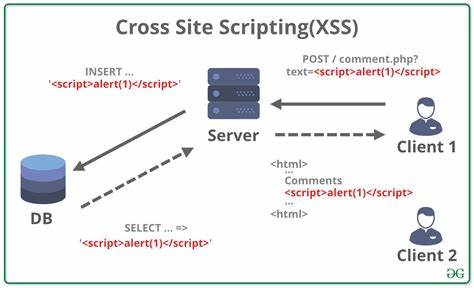
**Cross-Site Scripting (XSS)**

Cross-Site Scripting (XSS) là một kỹ thuật tấn công thường được sử dụng để khai thác các lỗ hổng bảo mật trong các ứng dụng web. Khi ứng dụng web không kiểm tra và xử lý đầu vào đúng cách, kẻ tấn công có thể chèn mã độc vào trang web và lừa người dùng bình thường thực hiện các hành động không an toàn như thực hiện các lệnh độc hại hoặc truy cập trái phép vào thông tin cá nhân của người dùng.

Các loại tấn công XSS phổ biến bao gồm:

* Reflected XSS: tấn công được thực hiện thông qua việc chèn mã độc vào các đầu vào không được kiểm tra của ứng dụng web, ví dụ như các trường tìm kiếm hoặc các tham số trên URL. Khi người dùng truy cập vào các trang web chứa mã độc này, các mã độc này sẽ được thực thi trên trình duyệt của người dùng.
* Stored XSS: tấn công được thực hiện thông qua việc chèn mã độc vào các trang web hoặc cơ sở dữ liệu của ứng dụng web. Khi người dùng truy cập vào các trang web hoặc thông tin chứa mã độc này, các mã độc này sẽ được thực thi trên trình duyệt của người dùng.
* DOM-based XSS: tấn công được thực hiện thông qua các lỗ hổng bảo mật trên DOM (Document Object Model) của trang web. Khi người dùng truy cập vào các trang web bị tấn công, các mã độc này sẽ được thực thi trên trình duyệt của người dùng.

Các hậu quả của tấn công XSS có thể rất nguy hiểm, bao gồm đánh cắp thông tin cá nhân của người dùng, phá hoại hoặc thay đổi nội dung trang web và tiềm tàng cho các cuộc tấn công khác.



Hình 2. 15. Cross-Site Scripting (XSS)

**Cross-Site Request Forgery (CSRF)**

Cross-Site Request Forgery (CSRF) là một kỹ thuật tấn công ứng dụng web mà kẻ tấn công tạo ra một yêu cầu giả mạo từ phía người dùng đã xác thực. Khi người dùng đang đăng nhập vào một trang web và trong cùng một trình duyệt mà không đăng xuất khỏi tài khoản đó, kẻ tấn công có thể sử dụng lỗ hổng này để tạo ra một yêu cầu từ trang web đã được xác thực.

Ví dụ, nếu trang web cho phép người dùng thay đổi mật khẩu bằng cách gửi một yêu cầu POST đến /changepassword, kẻ tấn công có thể tạo ra một trang web giả mạo và chứa một yêu cầu POST tới /changepassword. Khi người dùng bấm vào liên kết trên trang web giả mạo, yêu cầu POST sẽ được gửi đến trang web đã được xác thực và mật khẩu của người dùng sẽ được thay đổi bởi kẻ tấn công.

Để ngăn chặn CSRF, các nhà phát triển web có thể sử dụng các kỹ thuật như sử dụng token CSRF để xác thực yêu cầu. Khi người dùng gửi yêu cầu, token này sẽ được đính kèm và kiểm tra xem có hợp lệ hay không. Nếu không hợp lệ, yêu cầu sẽ bị từ chối.



Hình 2. 16. Cross-Site Request Forgery (CSRF)

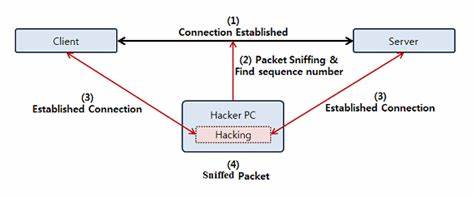
**Broken Authentication and Session Management**

Broken Authentication and Session Management là một lỗ hổng bảo mật trong các ứng dụng web, khiến cho kẻ tấn công có thể truy cập trái phép vào các tài khoản người dùng hoặc chiếm quyền điều khiển phiên đăng nhập của họ.

Các lỗ hổng này thường xảy ra do việc cài đặt kém hoặc không chính xác các phương thức xác thực và quản lý phiên đăng nhập, ví dụ như:

* Sử dụng mật khẩu yếu hoặc lưu trữ mật khẩu theo cách không an toàn.
* Không kích hoạt tính năng bảo mật 2 lớp, chẳng hạn như mã OTP (One-Time Password).
* Không giới hạn số lần đăng nhập sai hoặc thời gian phiên đăng nhập.
* Sử dụng mã định danh phiên quá dễ dàng để đoán, ví dụ như sử dụng ID người dùng làm mã phiên hoặc sử dụng thời gian để tạo ra mã phiên.
* Không xóa phiên đăng nhập khi người dùng đăng xuất hoặc đóng trình duyệt.

Những lỗ hổng này có thể được sử dụng để chiếm quyền điều khiển tài khoản người dùng, thực hiện các hoạt động trái phép như thay đổi thông tin cá nhân, thực hiện giao dịch không hợp lệ hoặc truy cập vào các tài nguyên không được phép. Để ngăn chặn Broken Authentication and Session Management, các nhà phát triển cần cài đặt các phương thức xác thực và quản lý phiên đăng nhập an toàn, đồng thời đảm bảo các phương thức này được áp dụng đầy đủ trên toàn bộ ứng dụng web.



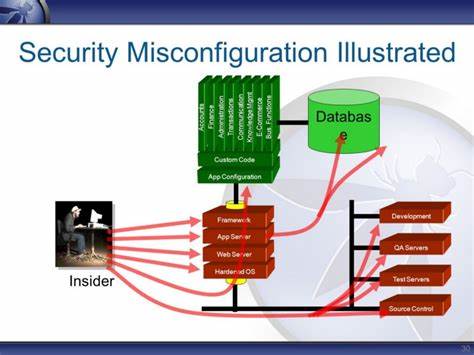
Hình 2. 17. Broken Authentication and Session Management

**Security Misconfiguration**

Security Misconfiguration (lỗi cấu hình bảo mật) là một lỗi thường gặp trong ứng dụng web khi các cài đặt bảo mật không được thiết lập đúng cách hoặc để lại các thiết lập mặc định, tạo nên các lỗ hổng bảo mật mà tấn công viên có thể tận dụng.

Các lỗi cấu hình bảo mật có thể là lỗi thiết lập quyền truy cập, thiết lập không đúng các mức độ bảo mật, sử dụng các cấu hình mặc định, để lộ thông tin nhạy cảm, cấu hình sai các bảo mật session, bảo mật transport, bảo mật CORS, bảo mật các thư viện, framework sử dụng và các lỗi cấu hình khác.

Những lỗ hổng này có thể cho phép kẻ tấn công xâm nhập vào hệ thống, thực hiện các hành động không đúng với quyền truy cập của họ, truy cập, thay đổi, xóa hoặc lấy cắp dữ liệu trên hệ thống, tấn công từ chối dịch vụ (DoS), thực hiện các cuộc tấn công khác vào hệ thống.



Hình 2. 18. Security Misconfiguration

**Insecure Cryptographic Storage**

Insecure Cryptographic Storage (Lưu trữ mã hóa không an toàn) là một trong những lỗ hổng bảo mật phổ biến trong ứng dụng web. Lỗ hổng này xảy ra khi các dữ liệu nhạy cảm (như mật khẩu, thông tin tài khoản, thông tin thẻ tín dụng) được lưu trữ trên máy chủ một cách không an toàn hoặc mã hóa sai cách.

Khi dữ liệu được lưu trữ không được mã hóa hoặc được mã hóa không đúng cách, kẻ tấn công có thể dễ dàng đánh cắp các thông tin nhạy cảm của người dùng và sử dụng chúng cho các mục đích xấu. Ví dụ, nếu mật khẩu của người dùng được lưu trữ dưới dạng văn bản thuần túy thay vì được mã hóa, thì kẻ tấn công có thể dễ dàng truy cập vào tài khoản của người dùng và thực hiện các hoạt động không đúng.

Để tránh Insecure Cryptographic Storage, các nhà phát triển cần sử dụng các thuật toán mã hóa mạnh mẽ và đảm bảo việc mã hóa và giải mã được thực hiện đúng cách. Ngoài ra, các dữ liệu nhạy cảm nên được lưu trữ trên một phương tiện lưu trữ an toàn và không được truyền đi dưới dạng văn bản thuần túy.

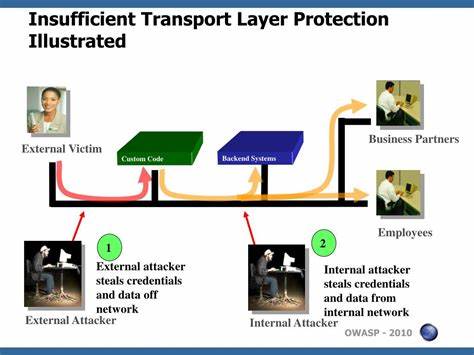


Hình 2. 19. Insecure Cryptographic Storage

**Insufficient Transport Layer Protection**

Insufficient Transport Layer Protection là một lỗ hổng bảo mật của ứng dụng web, thường xảy ra khi các thông tin được truyền đi giữa máy khách và máy chủ bằng một kênh không được mã hóa hoặc được mã hóa bằng các phương thức yếu. Khi thông tin được truyền đi bằng kênh không được mã hóa, thì thông tin đó có thể bị đánh cắp bởi kẻ tấn công, bằng cách sử dụng các kỹ thuật như sniffer hoặc man-in-the-middle. Khi thông tin được mã hóa bằng các phương thức yếu, thì nó có thể bị dễ dàng giải mã bởi kẻ tấn công sử dụng các công cụ mã hóa và giải mã.

Một số ví dụ về Insufficient Transport Layer Protection là sử dụng HTTP thay vì HTTPS để truyền thông tin nhạy cảm như tên người dùng và mật khẩu, sử dụng các thuật toán mã hóa yếu để mã hóa thông tin, sử dụng các chứng chỉ SSL/TLS không hợp lệ hoặc đã hết hạn, và sử dụng các kênh truyền khác như FTP hoặc Telnet mà không được mã hóa.



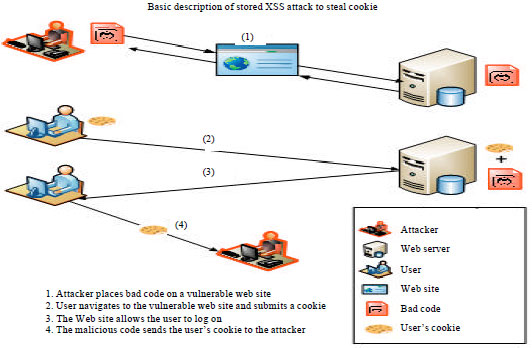
Hình 2. 20. Insufficient Transport Layer Protection

**Unvalidated Input**

Unvalidated Input là một lỗ hổng bảo mật trong ứng dụng web, xảy ra khi dữ liệu được gửi đến ứng dụng mà không được kiểm tra và xác nhận tính hợp lệ trước khi được sử dụng. Điều này có thể dẫn đến việc ứng dụng xử lý dữ liệu không đúng cách hoặc bị lỗi, hoặc có thể bị tấn công bởi các kẻ tấn công để thực hiện các cuộc tấn công khác nhau như Injection Attacks, Cross-Site Scripting (XSS), hoặc Command Injection.

Ví dụ về Unvalidated Input là khi người dùng đăng nhập vào ứng dụng mà không được kiểm tra tính hợp lệ của tên đăng nhập và mật khẩu, cho phép người dùng đăng nhập bằng các giá trị không hợp lệ và thực hiện các cuộc tấn công. Một ví dụ khác là khi người dùng nhập vào một URL mà không được kiểm tra và xác nhận tính hợp lệ trước khi được sử dụng, cho phép người dùng truy cập vào các trang web độc hại hoặc tải về các tệp tin độc hại.

Để giảm thiểu lỗ hổng Unvalidated Input, các nhà phát triển cần phải xác thực và kiểm tra tính hợp lệ của dữ liệu đầu vào trước khi sử dụng nó. Các phương pháp để xác thực và kiểm tra tính hợp lệ của dữ liệu đầu vào bao gồm sử dụng các biểu thức chính quy (regular expressions), giới hạn các giá trị nhập vào, kiểm tra độ dài của chuỗi, và sử dụng các công cụ kiểm tra đầu vào.



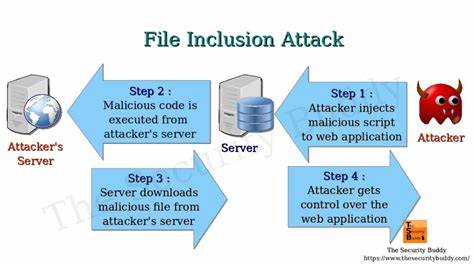
Hình 2. 21. Unvalidated Input

**File inclusion vulnerabilities**

File inclusion vulnerabilities là một loại lỗ hổng phát hiện trong các ứng dụng web, cho phép kẻ tấn công bao gồm một tệp trên máy chủ web mà không nên được bao gồm. Điều này có thể dẫn đến kẻ tấn công truy cập trái phép vào thông tin nhạy cảm hoặc thực thi mã tùy ý trên máy chủ web.

Có hai loại chính của lỗ hổng bao gồm local file inclusion (LFI) và remote file inclusion (RFI). LFI xảy ra khi kẻ tấn công có thể bao gồm một tệp cục bộ trên máy chủ web, trong khi RFI xảy ra khi kẻ tấn công có thể bao gồm một tệp từ một máy chủ web khác.

File inclusion vulnerabilities thường do việc không kiểm tra đầu vào đúng cách gây ra, chẳng hạn như không kiểm tra đầu vào được cung cấp bởi người dùng hoặc không làm sạch đầu vào đúng cách trước khi sử dụng nó để xây dựng đường dẫn tệp. Những lỗ hổng này có thể được giảm thiểu thông qua kiểm tra đầu vào đúng cách và sử dụng các thực hành lập trình an toàn.



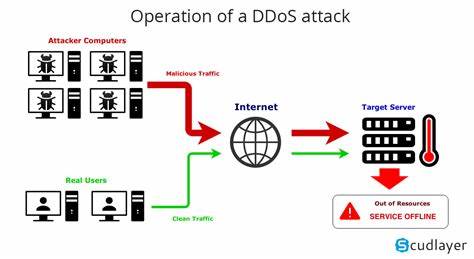
Hình 2. 22. File inclusion vulnerabilities

**DDoS (Distributed Denial of Service) attacks**

DDoS (Distributed Denial of Service) attacks là một dạng tấn công mạng nhằm làm ngập bằng thông tin yêu cầu (requests) đến một hệ thống hay một trang web từ nhiều nguồn khác nhau, đến mức làm hệ thống không thể xử lý được. Kết quả là, hệ thống sẽ trở nên chậm hoặc tạm ngừng hoạt động, không còn phản hồi các yêu cầu từ người dùng hợp lệ.

DDoS attacks thường được thực hiện bởi một nhóm lớn các máy tính, được kiểm soát từ xa bởi kẻ tấn công. Những máy tính này được kiểm soát bởi phần mềm độc hại hoặc bởi việc xâm nhập vào các hệ thống khác. Khi bị kiểm soát, các máy tính này sẽ được sử dụng để gửi lượng lớn các yêu cầu đến hệ thống mục tiêu, tạo ra một lượng lớn thông tin yêu cầu nhân tạo, chiếm dụng băng thông và tài nguyên hệ thống, từ đó làm cho hệ thống trở nên quá tải và không thể xử lý được các yêu cầu từ người dùng hợp lệ.

DDoS attacks là một trong những mối đe dọa lớn đối với các tổ chức trực tuyến và website. Các hệ thống bị tấn công có thể bị tạm ngừng hoạt động trong một khoảng thời gian dài, gây thiệt hại đến danh tiếng của công ty, mất doanh thu và ảnh hưởng đến trải nghiệm người dùng.



Hình 2. 23. DDoS attacks

## **2.4. Phương pháp phòng tránh tấn công ứng dụng web**

Để phòng tránh tấn công ứng dụng web, ta có thể thực hiện các biện pháp sau:

Cập nhật hệ thống: Đảm bảo hệ thống đang chạy phiên bản phần mềm mới nhất và các lỗ hổng bảo mật đã được vá.

Áp dụng các chuẩn bảo mật: Áp dụng các chuẩn bảo mật như OWASP Top 10 để giảm thiểu các lỗ hổng bảo mật.

Kiểm tra đầu vào: Kiểm tra và xác thực tất cả đầu vào từ người dùng trước khi sử dụng nó trong ứng dụng web.

Kiểm tra và xử lý lỗi: Kiểm tra và xử lý lỗi để tránh các lỗ hổng bảo mật có thể được khai thác.

Sử dụng mã hóa: Sử dụng mã hóa để bảo vệ dữ liệu quan trọng, chẳng hạn như mật khẩu hoặc thông tin tài khoản.

Quản lý phiên: Sử dụng quản lý phiên để đảm bảo rằng phiên của người dùng được duy trì và xác thực.

Kiểm tra và giám sát hệ thống: Kiểm tra và giám sát hệ thống để phát hiện và phản ứng nhanh chóng trước các hành vi bất thường.

Sử dụng giải pháp bảo mật mạng: Sử dụng giải pháp bảo mật mạng như firewall, IPS và WAF để ngăn chặn các cuộc tấn công từ xa và tấn công chéo trang.

Thực hiện kiểm tra bảo mật thường xuyên: Thực hiện kiểm tra bảo mật thường xuyên để phát hiện và giải quyết các lỗ hổng bảo mật còn tồn đọng trên hệ thống.

Tất cả các biện pháp trên đều cần phải được thực hiện đầy đủ và liên tục để đảm bảo bảo mật cho ứng dụng web.

# **CHƯƠNG 3: CÀI ĐẶT THỰC NGHIỆM**

## **3.1. Triển khai OWASP Honeypot Project**

Để triển khai OWASP Honeypot Project trên máy ảo Linux Ubuntu, bạn có thể thực hiện các bước sau:

* Cài đặt các phần mềm cần thiết:
* Hệ điều hành Linux Ubuntu.
* Các phần mềm hỗ trợ cho việc triển khai honeypot như Kippo, Cowrie, Honeyd, Dionaea, Glastopf, Snort, Suricata.
* Cấu hình mạng để honeypot có thể được truy cập từ bên ngoài.
* Cài đặt honeypot:
* Tùy thuộc vào loại honeypot mà bạn chọn, bạn có thể cài đặt Kippo, Cowrie, Honeyd, Dionaea, Glastopf, Snort hoặc Suricata bằng cách sử dụng các lệnh cài đặt phần mềm trên Ubuntu như apt-get.
* Cấu hình honeypot:
* Sau khi cài đặt honeypot, bạn cần cấu hình nó để phù hợp với môi trường mạng của bạn. Các tùy chọn cấu hình có thể khác nhau tùy thuộc vào honeypot mà bạn sử dụng. Ví dụ: Cấu hình file cấu hình cho Kippo hoặc Cowrie, tạo các tập tin giả mạo cho Glastopf.
* Khởi động honeypot và giám sát hoạt động của nó:
* Sau khi cấu hình honeypot, bạn có thể khởi động nó và giám sát hoạt động của nó để phát hiện các tấn công và các hoạt động bất thường trong mạng.

Lưu ý: Ngoài các bước trên, để triển khai thành công honeypot, bạn nên nghiên cứu kỹ về các honeypot và cấu hình phù hợp để tăng hiệu quả giám sát và phát hiện các tấn công. Bạn cũng cần có kiến thức về bảo mật mạng để hiểu rõ hơn về các mối đe dọa và biện pháp phòng chống.

**Bước 1: Cài đặt Elastichsearch**

Elaticsearch là một nền tảng để tìm kiếm và phân tích dữ liệu phân tán trong thời gian thực. Nó là một lựa chọn phổ biến do khả năng sử dụng, các tính năng mạnh mẽ và khả năng mở rộng của nó.

Bài viết này sẽ hướng dẫn bạn cài đặt Elaticsearch, định cấu hình nó cho trường hợp sử dụng của bạn, bảo mật cài đặt của bạn và bắt đầu làm việc với máy chủ Elaticsearch của bạn.

điều kiện tiên quyết

Trước khi làm theo hướng dẫn này, bạn sẽ cần:

• Một máy chủ Ubuntu 20.04 với 4GB RAM và 2 CPU được thiết lập với người dùng sudo chưa root. Bạn có thể đạt được điều này bằng cách làm theo Thiết lập máy chủ ban đầu với Ubuntu 20.04

• Đã cài đặt OpenJDK 11

Đối với hướng dẫn này, chúng tôi sẽ làm việc với lượng CPU và RAM tối thiểu cần thiết để chạy Elaticsearch. Lưu ý rằng dung lượng CPU, RAM và dung lượng lưu trữ mà máy chủ Elaticsearch của bạn sẽ yêu cầu tùy thuộc vào khối lượng nhật ký mà bạn mong đợi.

* **Cài đặt và định cấu hình Elaticsearch**

Các thành phần Elaticsearch không có sẵn trong kho gói mặc định của Ubuntu. Tuy nhiên, chúng có thể được cài đặt bằng APT sau khi thêm danh sách nguồn gói của Elastic.

Tất cả các gói được ký bằng khóa ký Elaticsearch để bảo vệ hệ thống của bạn khỏi giả mạo gói. Các gói đã được xác thực bằng khóa sẽ được trình quản lý gói của bạn coi là đáng tin cậy. Trong bước này, bạn sẽ nhập khóa GPG công khai của Elaticsearch và thêm danh sách nguồn gói Đàn hồi để cài đặt Elaticsearch.

Để bắt đầu, hãy sử dụng cURL, công cụ dòng lệnh để truyền dữ liệu bằng URL, để nhập khóa GPG công khai của Elaticsearch vào APT. Lưu ý rằng chúng tôi đang sử dụng các đối số -fsSL để tắt tất cả tiến trình và các lỗi có thể xảy ra (ngoại trừ lỗi máy chủ) và để cho phép cURL thực hiện yêu cầu trên một vị trí mới nếu được chuyển hướng. Đưa đầu ra của lệnh cURL vào chương trình khóa apt, chương trình này sẽ thêm khóa GPG công khai vào APT..

curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add –

Tiếp theo, thêm danh sách nguồn đàn hồi vào thư mục sources.list.d, nơi APT sẽ tìm kiếm các nguồn mới:

echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list

Tiếp theo, hãy cập nhật danh sách gói của bạn để APT sẽ đọc nguồn Đàn hồi mới:

sudo apt-get update

Sau đó cài đặt Elaticsearch bằng lệnh này:

sudo apt-get install elasticsearch

Elaticsearch hiện đã được cài đặt và sẵn sàng để được định cấu hình.

* **Định cấu hình Elaticsearch**

Để định cấu hình Elaticsearch, chúng tôi sẽ chỉnh sửa tệp cấu hình chính của nó elaticsearch.yml nơi lưu trữ hầu hết các tùy chọn cấu hình của nó. Tệp này nằm trong thư mục /etc/elasticsearch.

Sử dụng trình soạn thảo văn bản ưa thích của bạn để chỉnh sửa tệp cấu hình của Elaticsearch. Ở đây, chúng tôi sẽ sử dụng nano:

sudo nano /etc/elasticsearch/elasticsearch.yml

**Lưu ý: Tệp cấu hình của Elaticsearch ở định dạng YAML, có nghĩa là chúng ta cần duy trì định dạng thụt lề. Đảm bảo rằng bạn không thêm bất kỳ khoảng trống nào khi chỉnh sửa tệp này.**

Tệp elaticsearch.yml cung cấp các tùy chọn cấu hình cho cụm, nút, đường dẫn, bộ nhớ, mạng, khám phá và cổng của bạn. Hầu hết các tùy chọn này được cấu hình sẵn trong tệp nhưng bạn có thể thay đổi chúng theo nhu cầu của mình. Với mục đích trình diễn cấu hình một máy chủ, chúng tôi sẽ chỉ điều chỉnh cài đặt cho máy chủ lưu trữ mạng.

Elaticsearch lắng nghe lưu lượng truy cập từ mọi nơi trên cổng 9200. Bạn sẽ muốn hạn chế quyền truy cập bên ngoài vào phiên bản Elaticsearch của mình để ngăn người ngoài đọc dữ liệu của bạn hoặc tắt cụm Elaticsearch của bạn thông qua [REST API] (https://en.wikipedia.org /wiki/Representational\_state\_transfer). Để hạn chế quyền truy cập và do đó tăng tính bảo mật, hãy tìm dòng chỉ định network.host, bỏ ghi chú và thay thế giá trị của nó bằng localhost để nó có dạng như sau:

/etc/elasticsearch/elasticsearch.yml

. . .

# ---------------------------------- Network -----------------------------------

#

# Set the bind address to a specific IP (IPv4 or IPv6):

#

network.host: localhost

Chúng tôi đã chỉ định máy chủ cục bộ để Elaticsearch lắng nghe trên tất cả các giao diện và IP bị ràng buộc. Nếu bạn muốn nó chỉ nghe trên một giao diện cụ thể, bạn có thể chỉ định IP của nó thay cho localhost. Lưu và đóng elaticsearch.yml. Nếu bạn đang sử dụng nano, bạn có thể làm như vậy bằng cách nhấn CTRL+X, tiếp theo là Y rồi ENTER .

Đây là những cài đặt tối thiểu bạn có thể bắt đầu để sử dụng Elaticsearch. Bây giờ bạn có thể bắt đầu Elaticsearch lần đầu tiên.

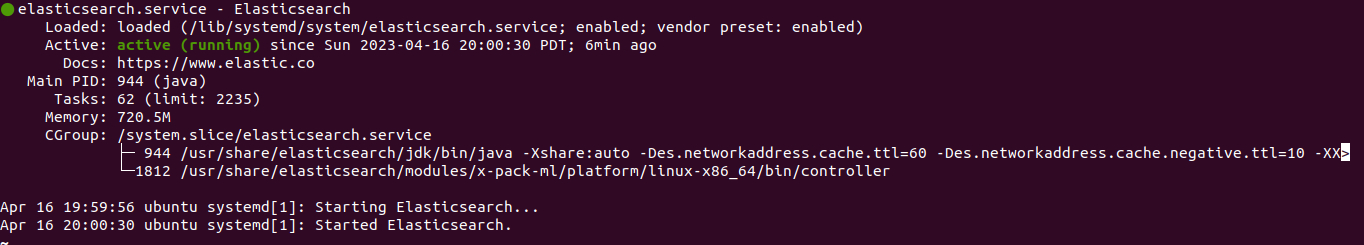
Bắt đầu dịch vụ Elaticsearch với systemctl. Hãy cho Elaticsearch một vài phút để khởi động. Nếu không, bạn có thể gặp lỗi về việc không thể kết nối.

sudo systemctl start elasticsearch

Tiếp theo, hãy chạy lệnh sau để kích hoạt Elaticsearch khởi động mỗi khi máy chủ của bạn khởi động:

sudo systemctl enable elasticsearch

Với Elaticsearch được bật khi khởi động, hãy chuyển sang bước tiếp theo để thảo luận về bảo mật.



Hình 3. 1. Hình ảnh cài xong elasticsearch

**Bước 2: Docker**

Cập nhật chỉ mục gói apt và cài đặt các gói để cho phép apt sử dụng kho lưu trữ qua HTTPS:

$sudo apt-get update

$ sudo apt-get install \

ca-certificates \

curl \

gnupg \

lsb-release

1. Thêm khóa GPG chính thức của Docker::

$sudo mkdir -m 0755 -p /etc/apt/keyrings

$curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

1. Sử dụng lệnh sau để thiết lập kho lưu trữ:

$echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

1. Cập nhật chỉ mục gói apt:

$ sudo apt-get update

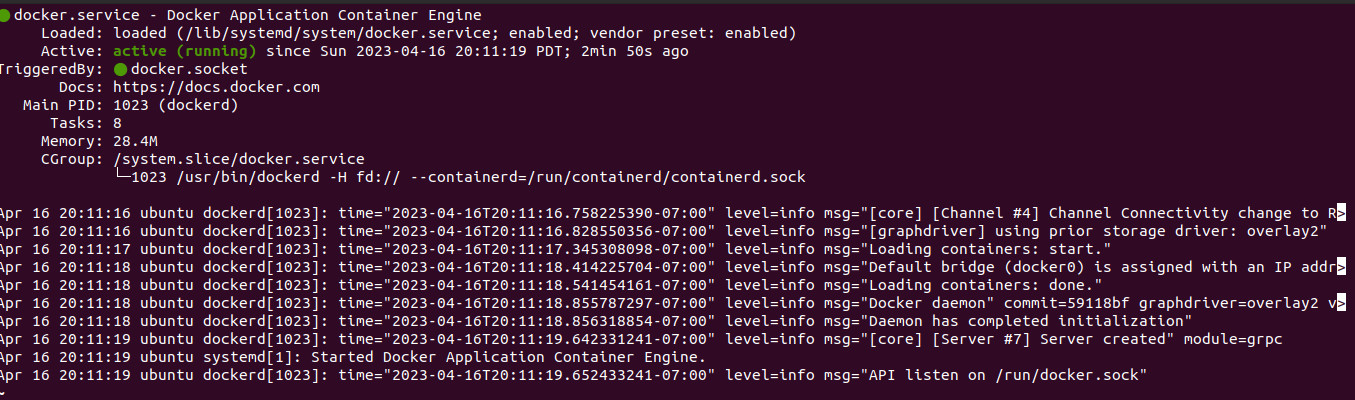
1. Cài đặt Docker Engine, containerd và Docker Compose.

$sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

1. Test

$ sudo service docker start

$ sudo service docker status



Hình 3. 2. Docker

**Bước 3:** Tải bộ cài OWASP Honeypot Project

<https://github.com/OWASP/Python-Honeypot>

* Sử dụng filezilla upload phần mềm này vào ubuntu server (/setup)

**Bước 4:** Cài đặt Pip3

$sudo apt install python3-pip

**Bước 5:** Cài thư viện

$Cd /setup/owasp-honey..

$sudo Pip3 install -r requirements.txt

$sudo apt-get install libpcap-dev libnet-dev tshark

**Bước 6:** Test Module

$sudo python3 ohp.py -m all --test

**Bước 7:** Kết nối

$sudo python3 ohp.py --start-api-server

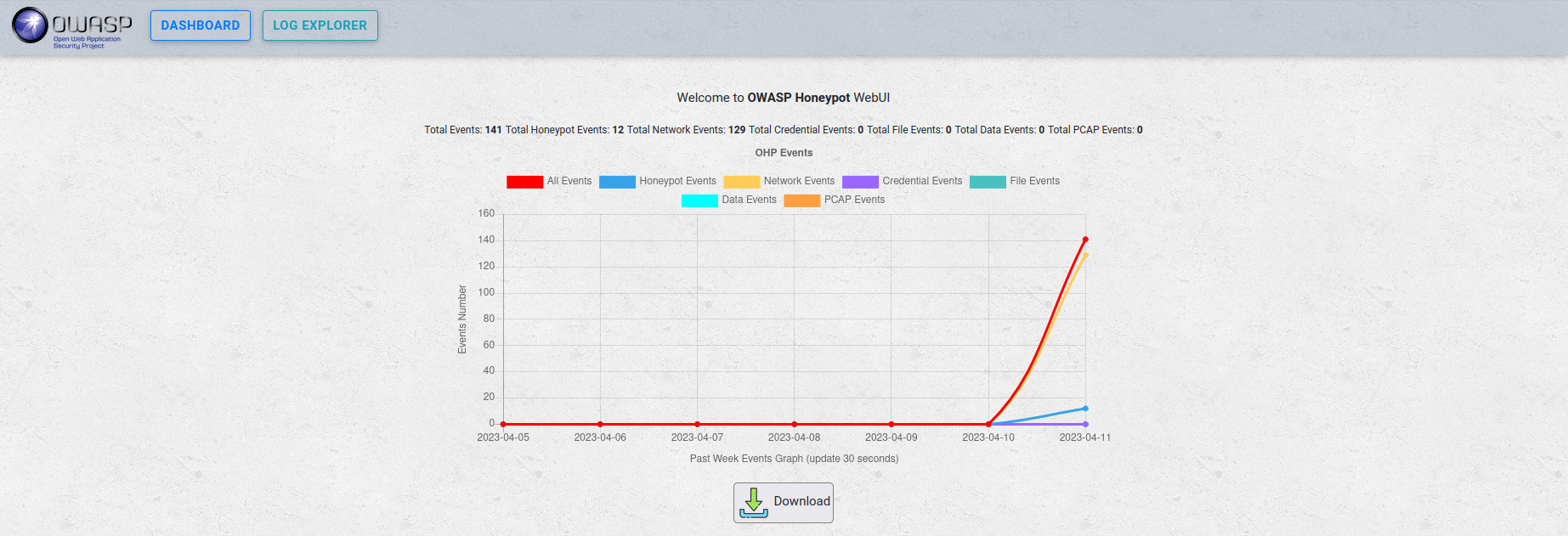
**Bước 8:** chạy chương trình

$sudo python ohp.py

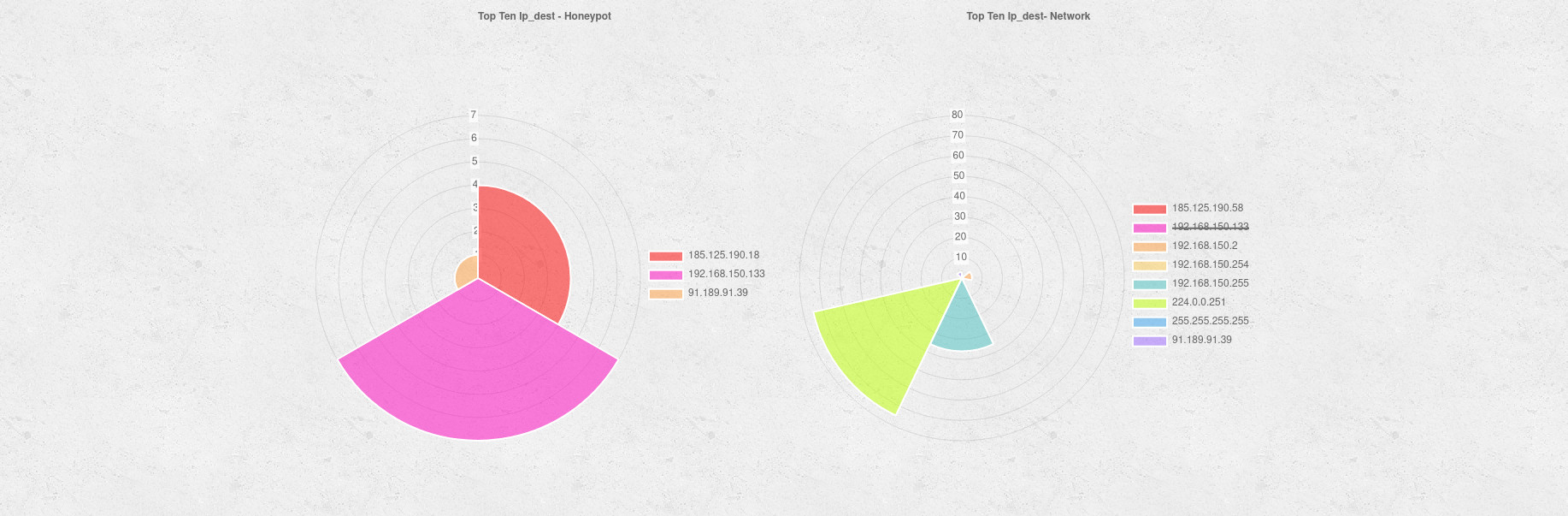
**Bước 9:** Kiểm tra WebUI

<http://ip:5000>

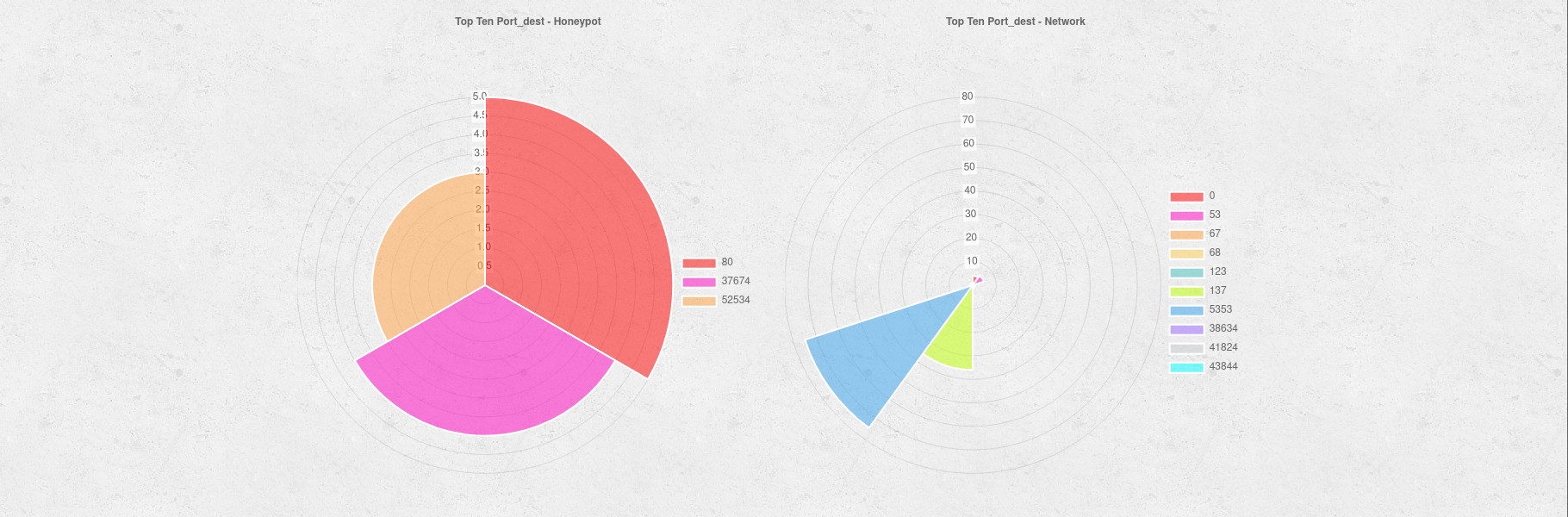
## **3.2. Kết quả thu được**



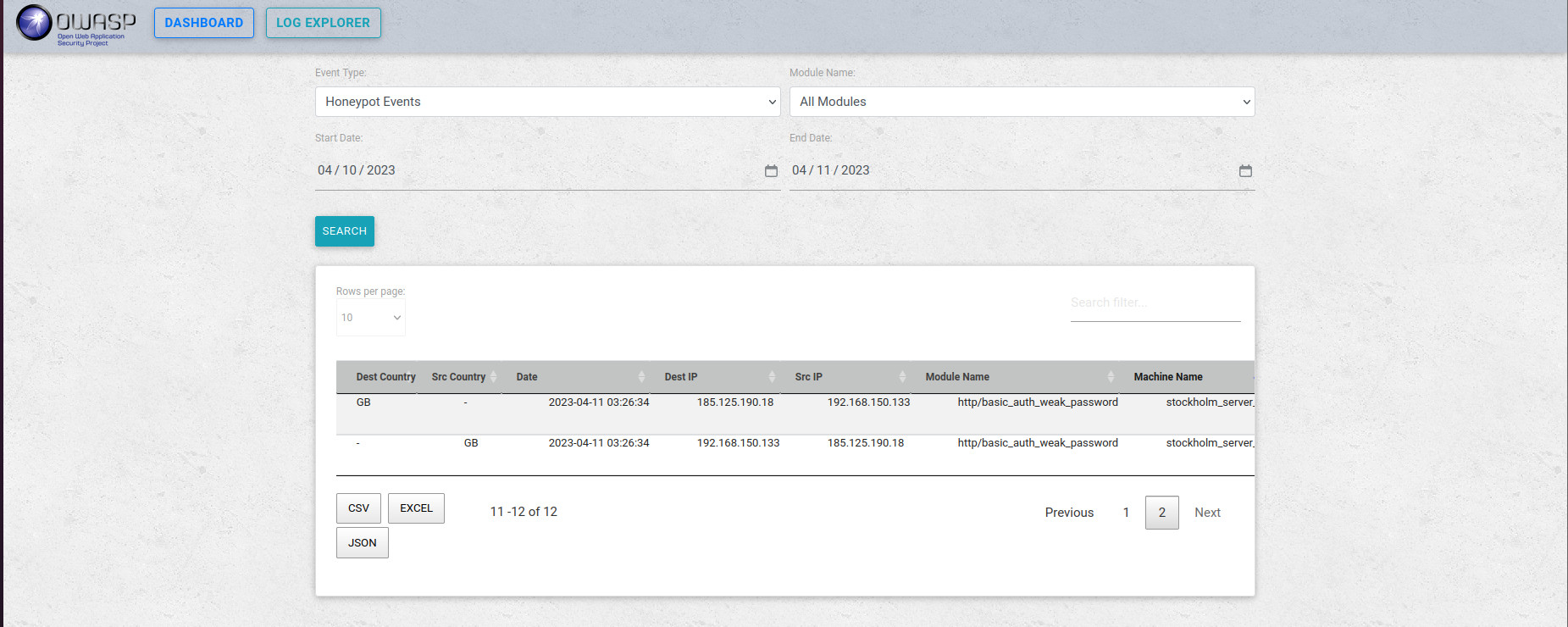
Hình 3. 3. Giao diện của owasp honeypot project



Hình 3. 4. Ip dest honeypot và ip dest network



Hình 3. 5. Port dest honeypot và port dest net work



Hình 3. 6. Log Explorer

# **CHƯƠNG 4: KẾT LUẬN**

## **4.1. Tóm lại**

Kết quả đạt được của bài luận " Nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project" bao gồm:

* Tìm hiểu và nắm được kiến thức cơ bản về an toàn và bảo mật hệ thống thông tin, đặc biệt là về các kỹ thuật tấn công ứng dụng web và các phương pháp phòng chống tấn công.
* Nghiên cứu và tìm hiểu về OWASP Honeypot Project, một dự án mã nguồn mở cung cấp các công cụ honeypot để thu thập thông tin về các cuộc tấn công ứng dụng web.
* Triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project trên môi trường máy ảo Linux Ubuntu, bao gồm cài đặt và cấu hình các thành phần của hệ thống như Apache, MySQL, PHP, và các công cụ honeypot như Honeyd, Thug, Glastopf, và Conpot.
* Thực hiện các thử nghiệm và phân tích kết quả thu thập được từ các cuộc tấn công vào ứng dụng OWASP Honeypot Project, bao gồm các kiểu tấn công Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), và DDoS.
* Đánh giá hiệu quả của hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project, đồng thời đề xuất các phương pháp cải tiến để nâng cao hiệu quả của hệ thống trong việc phát hiện và phòng chống các cuộc tấn công ứng dụng web.

Tổng kết, bài luận đã đạt được các kết quả quan trọng trong việc nghiên cứu và triển khai hệ thống thu thập và phân tích tấn công vào ứng dụng OWASP Honeypot Project, qua đó giúp cải thiện khả năng phòng chống và phát hiện các cuộc tấn công ứng dụng web trên các hệ thống thông tin hiện đại.

## **4.2. Những việc thực hiện được**

\_ Đã hoàn thành và chạy được ở localhost cũng như trên website

\_ Thực hiện trên cả Owasp Zap để kiểm thử và tìm lỗi ở trên website

## **4.3. Những việc chưa thực hiện được**

**TÀI LIỆU THAM KHẢO**

<https://github.com/OWASP/Honeypot-Project/issues>

<https://securityonline.info/owasp-honeypot/>

<https://readthedocs.org/projects/owasp-honeypot/>

<https://pypi.org/project/honeypots/>

<https://github.com/OWASP/Python-Honeypot>