|  |  |
| --- | --- |
| logomobile.png | **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**HỆ THỐNG HONEYPOT**

Nghành: **CÔNG NGHỆ THÔNG TIN**

Lớp: **18DTHB3**

**Giảng Viên Hướng Dẫn : Ths.Tống Thanh Văn**

Sinh Viên Thực Hiện :  **Lê Tuấn Đạt-1811063584**

**Lê Tiến Anh-1811061252**

**Lê Hải Đăng-1811061887**

TP.Hồ Chí Minh ,2021

|  |  |
| --- | --- |
| logomobile.png | **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**HỆ THỐNG HONEYPOT**

Nghành: **CÔNG NGHỆ THÔNG TIN**

Lớp: **18DTHB3**

**Giảng Viên Hướng Dẫn : Ths.Tống Thanh Văn**

Sinh Viên Thực Hiện :  **Lê Tuấn Đạt-1811063584**

**Lê Tiến Anh-1811061252**

**Lê Hải Đăng-1811061887**

TP.Hồ Chí Minh ,2021

Nhận Xét Của Giảng Viên

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

................................................................................................................................................

Mục Lục

Contents

[LỜI NÓI ĐẦU 7](#_Toc70446460)

[**Chương I: TỔNG QUAN VỀ HỆ THỐNG HONEYPOT** 9](#_Toc70446461)

[1.1 Tổng quan về đề tài 9](#_Toc70446462)

[1.2 Mục tiêu của đề tài 9](#_Toc70446463)

[1.3 Cấu trúc đề tài 10](#_Toc70446464)

[**Chương II: CƠ SỞ LÝ THUYẾT** 10](#_Toc70446465)

[2.1 Khái niệm Honeypot: 10](#_Toc70446466)

[2.2Phân loại Honeypot: 15](#_Toc70446467)

[Honeyd: 16](#_Toc70446468)

[2.3. HONEYNET 16](#_Toc70446469)

[2.3.1. Khái niệm 16](#_Toc70446470)

[**2.3.2. Chức năng** 19](#_Toc70446472)

[**- Điều khiển được luồng dữ liệu** 19](#_Toc70446473)

[2.3.3 Một số mô hình triển khai Honeynet trên thế giới 20](#_Toc70446474)

[20](#_Toc70446475)

[2.3.5 Vai trò và ý nghĩa của Honeynet 23](#_Toc70446476)

[.2.3.6.. Khả năng an toàn và các rủi ro 24](#_Toc70446477)

[**2.3.7 Đánh giá so sánh mức độ an toàn** 24](#_Toc70446479)

[**2.**4 **MÔ HÌNH KIẾN TRÚC** 24](#_Toc70446481)

[Mô hình kiến trúc vật lý 25](#_Toc70446482)

[Mô hình kiến trúc Honeynet thế hệ I 25](#_Toc70446483)

[Mô hình kiến trúc Honeynet II, III 28](#_Toc70446484)

[1.3,Hệ thống Honeynet ảo 29](#_Toc70446485)

[Mô hình kiến trúc loggic của Honeynet 30](#_Toc70446486)

[31](#_Toc70446487)

[Module điều khiển dữ liệu (hay kiểm soát dữ liệu) 32](#_Toc70446488)

[Vai trò - nhiệm vụ của Module điều khiển 32](#_Toc70446489)

[Cơ chế kiểm soát dữ liệu 34](#_Toc70446490)

[Giới hạn số lượng kết nối ra bên ngoài 34](#_Toc70446491)

[Lọc gói tin độc hại (Packet Scrubbed) 35](#_Toc70446492)

[Kiểm soát dữ liệu trong Honeynet II 37](#_Toc70446493)

[Tường lửa IPTABLES 37](#_Toc70446494)

[IDS Snort 40](#_Toc70446495)

[Module thu nhận dữ liệu 45](#_Toc70446496)

[Vai trò - nhiệm vụ của Module thu nhận dữ liệu 45](#_Toc70446497)

[Cơ chế thu nhận dữ liệu 46](#_Toc70446498)

[Thu nhận dữ liệu từ Firewall 47](#_Toc70446499)

[Thu nhận dữ liệu từ luồng dữ liệu mạng 48](#_Toc70446500)

[Thu nhận dữ liệu từ hoạt động trên các Honeypot 48](#_Toc70446501)

[Modul phân tích dữ liệu 53](#_Toc70446502)

[1.1.2. Vai trò 53](#_Toc70446503)

[1.1.3. Cơ chế phân tích dữ liệu 53](#_Toc70446504)

[2.5.2 HONEYPOT CHO MALWARE TRÊN CÁC THIẾT BỊ  USB 57](#_Toc70446505)

[**CHƯƠNG III: KẾT QUẢ THỰC NGHIỆM** 59](#_Toc70446506)

[3.1 Công cụ sử dụng 59](#_Toc70446507)

[3.1.1 Công cụ KFsensor 59](#_Toc70446508)

[3.2.2 Công cụ FileZilla 61](#_Toc70446509)

[3.2 Các bước thực hiện 62](#_Toc70446511)

[3.3 Kết luật 68](#_Toc70446512)

[CHƯƠNG IV:KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 69](#_Toc70446513)

[4.1 Kết luận 69](#_Toc70446514)

[4.2 Hướng phát triển 69](#_Toc70446515)

[4.3Tài Liệu Tham Khảo 69](#_Toc70446516)

Mục Lục Hình Ảnh

# LỜI NÓI ĐẦU

Ngày nay, công nghệ thông tin đang phát triển với tốc độ “vũ bão”, bên cạnh những mặt tích cực và lợi ích to lớn mà Xã hội thông tin mang lại cho nhân loại thì lại tồn tại các mặt tiêu cực như : các nguy cơ tấn công mạng nhằm phá hoại hệ thống mạng, nguy cơ bị đánh cắp các thông tin “nhạy cảm “ của cá nhân, các tổ chức, doanh nghiệp, các cơ quan Nhà nước ….. Để ngăn chặn lại những nguy cơ này, đòi hỏi các Cơ quan, tổ chức, doanh nghiệp, phải tổ chức xây dựng các Hệ thống an ninh mạng nhằm đảm bảo an toàn cho Hệ thống mạng của Cơ quan mình.

Và trong vô số các biện pháp ngăn chặn đó, thì "Honeypot" (tạm gọi là Mắt ong) và "Honeynet" (tạm gọi là Tổ ong) được coi là một trong những cạm bẫy hết sức hiệu quả, được thiết kế với mục đích này. Đối với các tin tặc thì Hệ thống này quả là những “ Cạm bẫy đáng sợ ”; vì vậy, giới Hacker thường xuyên thông báo – cập nhật các hệ thống Honeynet mới được triển khai trên thế giới ở các diễn đàn Hacker, nhằm tránh “sa bẫy” những hệ thống Honeynet này.

Khác với các hệ thống An ninh mạng khác như: Hệ thống phát hiện xâm nhập và chống xâm nhập ( IDS - IPS ), Hệ thống Firewall,…, được thiết kế làm việc thụ động trong việc phát hiện - ngăn chặn sự tấn công của tin tặc ( Hacker ) vào hệ thống mạng; thì Honeynet lại được thiết kế nhằm chủ động lôi kéo Hacker tấn công vào hệ thống giả được bố trí bên cạnh hệ thống thật nhằm mục đích:

* Thu thập các kỹ thuật – phương pháp tấn công, các công cụ mà Hacker sử dụng, đặc biệt là các kỹ thuật tấn công mạng mới , các mẫu virus- mã độc mới.

Giúp chúng ta sớm phát hiện ra các lỗ hổng bảo mật tồn tại trên các sản phẩm công nghệ thông tin đã triển khai - cài đặt trên Hệ thống thật. Từ đó, sớm có biện pháp ứng phó - khắc phục kịp thời. Đồng thời, cũng kiểm tra độ an toàn của hệ thống mạng, các dịch vụ mạng ( như : Web, DNS, Mail,…), và độ an toàn - tin cậy - chất lượng của các sản phẩm thương mại công nghệ thông tin khác ( đặc biệt là các Hệ điều hành như : Unix, Linux, Window,…) Thu thập các thông tin, dấu vết của Hacker ( như : địa chỉ IP của máy Hacker sử dụng tấn công, vị trí địa lý của Hacker, thời gian Hacker tấn công,…). Từ đó, giúp chuyên gia an ninh mạng truy tìm thủ phạm.

Tuy nhiên, do điều kiện thời gian có hạn nên trong “Bài cáo này” chỉ trình bày nội dung “Nghiên cứu Hệ thống Honeypots nhằm nghiên cứu một số kỹ thuật tấn công dịch vụ Web”, nhờ đó giúp chúng ta sớm phát hiện và kịp thời khắc phục các lỗi hổng bảo mật tồn tại trên dịch vụ Web. Em hi vọng thông qua nội dung trình bày nghiên cứu của em dưới đây sẽ giúp chúng ta hiểu được Hệ thống Honeypot cùng với vai trò - tác dụng to lớn của Hệ thống này trong nhiệm vụ đảm bảo An ninh mạng hiện nay.

**Chương I: TỔNG QUAN VỀ HỆ THỐNG HONEYPOT**

Chương này sẽ trình bày kiến thức tổng quan, cơ bản về Honeypot,mục tiêu và cấu trúc của đề tài.

* 1. Tổng quan về đề tài

Tổng quan Trong những năm gần đây, các cuộc xâm nhập mạng gia tăng đáng kể, do sự phổ dụng của các công cụ tấn công được tự động hoặc được lập kịch bản. Điều này đã thúc đẩy sự quan tâm đến các hệ thống Honeypots, hệ thống này có thể được dùng để “bẫy” và giải mã các phương pháp tấn công. Các chuyên gia bảo mật cho biết: các kẻ tấn công hiện đều đang rất ngán ngẩm khi phải tấn công vào một hệ thống Linux dạng trung bình. Chi phí cho một cuộc đột nhập thành công vào một hệ thống sử dụng Linux cao hơn nhiều so với chi phí bỏ ra để đột nhập vào hệ thống sử dụng Windows. Dự án mang tên Honeypots được tạo ra với mục đích giả lập các hệ thống mạng Linux bình thường để câu nhử các cuộc tấn công nhằm nghiên cứu độ an toàn của các hệ thống máy chủ Linux. Các kết quả nghiên cứu do Honeypots đưa ra cho biết: khoảng thời gian tồn tại an toàn của các hệ thống máy chủ chạy Linux đã gia tăng đột ngột trong 2 năm gần đây. Honeypots đã chỉ ra rằng: Trong giai đoạn hiện nay, một hệ thống máy chủ Linux chưa được cài đầy đủ các bản sửa lỗi vẫn có thể “chịu đựng” an toàn trung bình là 3 tháng trước các cuộc tấn công, khi so sánh với giai đoạn 2001-2002 chỉ là 72 giờ. Một số hệ thống máy chủ của dự án đã an toàn trong suốt 9 tháng trời trước mọi cuộc tấn công.

* 1. Mục tiêu của đề tài

Giúp chúng ta thấy rõ quá trình tấn công của hacker diễn ra cụ thể như thế nào: các tấn công cụ thể diễn ra từng bước ra sao. Sau khi chiếm được quyền điều khiển honeypot, hacker làm gì v.v..

Thu được kỹ thuật tấn công của hacker: Kỹ thuật tấn công SQL-injection, thực thi web shell từ xa (Remote Procedure Call – RPC ), phát tán virus.

Ngoài ra còn có thể giúp chúng ta biết được công cụ giúp hacker phát hiện website bị lỗi SQL Injection mà không phải phát hiện bằng tay.

Với mục đích nghiên cứu Honeypot để thu thập các kỹ thuật tấn công dịch vụ web. Từ đó, giúp chúng ta sớm phát hiện các lỗ hổng bảo mật, điểm yếu của Web để chúng ta sớm có các biện pháp khắc phục, sử lý kịp thời, đảm bảo an toàn cho Website lẫn người sử dụng web

* 1. Cấu trúc đề tài

Chương 1 Tổng quan về hệ thống Honeypot

Chương 2 Cơ sở hệ thống Honeypot

Chương 3 Kết quả thực nghiệm

Chương 4 Kết luận và hướng phát triển

# **Chương II: CƠ SỞ LÝ THUYẾT**

## 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.

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.

* **Ưu điểm của Honeypot**: Honeypot là một khái niệm rất đơn giản, trong đó 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 độ

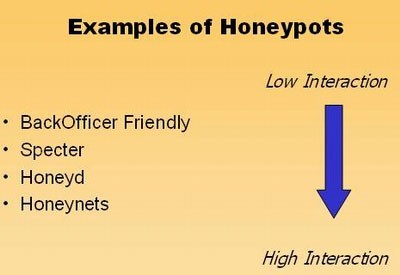
hại. Như vậy, Honeypot đã giảm thiểu được “tiếng ồn”, có nghĩ là với bộ thu thập dữ liệu nhỏ, nhưng thông tin có giá trị cao, nhưng đó chỉ là những hành động xấu. Điều này có nghĩa là sẽ dễ dàng hơn nhiều để phân tích các dữ liệu mà Honeypot thu thập và lấy được giá trị từ nó.

* + 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 tại, 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.

## 2.2Phân 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 1.1*-* Các loại hình Honeypot

Một số ví dụ về các loại honeypot :

1. **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…

1. **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.

## 2.3. HONEYNET

### 2.3.1. Khái niệm

Một trong các công cụ chính mà Nhóm dự án Honeynet sử dụng để thu thập thông tin là Honeynet. Honeynet khác với các hệ thống Firewall, hệ thống phát hiện và ngăn chặn xâm nhập, hệ thống mã hóa ở chỗ : các hệ thống tuy đều có khả năng bảo vệ hệ thống mạng và tài nguyên mạng nhưng các hệ thống này đều là thực hiện nhiệm vụ “Phòng thủ”, mang tính thụ động; ngược lại, Honeynet lại là hệ thống chủ động lôi kéo, thu hút sự chú ý và tấn công của Hacker nhằm thu thập các thông tin của Hacker như: Kỹ thuật tấn công của Hacker, công cụ Hacker sử dụng, các loại mã độc mới được xuất hiện,....

Honeynet (tạm gọi là “Tổ ong”) là một hình thức của honeypot tương tác cao. Khác với các honeypot khác, Honeynet là một hệ thống thật, hoàn toàn giống một mạng làm việc bình thường ; và Honeynet cung cấp các hệ thống, ứng dụng, các dịch vụ thật như : Web, Mail, File server,...

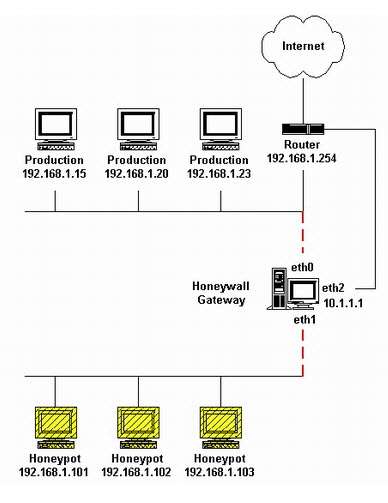
Hệ thống Honeynet có thể triển khai xây dưng ở nhiều cơ quan, tổ chức với nhiều mục đích khác nhau như: Các cơ quan nhà nước, doanh nghiệp có thể sử dụng Honeynet nhằm kiểm tra độ an toàn của hệ thống mạng của mình và ngăn chặn kẻ tấn công tấn công vào hệ thống thật; các cơ quan, tổ chức, doanh nghiệp hoạt động trong lĩnh vực an ninh mạng có thể sử dụng Honeynet nhằm thu thập các loại mã độc hại mới như: virus, worm, spyware, trojan,… , để kịp thời viết chương trình cập nhật diệt mã độc cho sản phẩm Anti-virus của công ty mình…..

Nhiệm vụ quan trọng nhất khi Triển khai xây dựng – cài đặt một hệ thống Honeynet chính là Honeywall. Honeywall là gateway ở giữa honeypot và mạng bên ngoài. Nó hoạt động ở tầng 2 như là Bridged.

Các luồng dữ liệu khi vào và ra từ honeypot đều phải đi qua Honeywall. Để kiểm soát các luồng dữ liệu này, cũng như thu thập các dấu hiệu tấn công, và ngăn chặn tấn công của các Hacker thì Honeywall sử dụng hai công cụ chính là:

* Một là IDS Snort (hay còn gọi là IDS sensor) gồm có các luật ( Rule ) định nghĩa các dấu hiệu tấn công, và thực hiện hiện bắt các gói tin ( Packet ).
* Hai là Firewall Iptables gồm có các luật (Rule) định nghĩa sự cho phép (Allow ) hoặc không cho phép ( Deny ) các truy cập từ bên ngoài vào hoặc bên trong hệ thống ra, và kiểm soát các luồng dữ liệu qua Honeywall.

Dưới đây là một ví dụ về Honeynet:



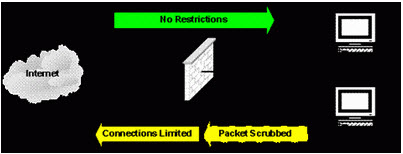
Hình 2.3.1 Honeywal

Với mô hình này Honeywall gồm có 3 card mạng là : eth0, eth1, eth2 . Card mạng eth0 thì kết nối với Production Network, card eth1 thì kết nối với các Honeypot, còn card thứ 3 kết nối với Router. Khi Hacker từ bên ngoài Internet tấn công vào hệ thống thì các Honeypot sẽ đóng vai trò là hệ thống thật tương tác với Hacker, và thực hiện thu thập các thông tin của Hacker như : địa chỉ IP của máy Hacker sử dụng, Kỹ thuật Hacker tấn công, các công cụ mà Hacker sử dụng …. Các thông tin này đều sẽ bị ghi lại trên Honeywall, và được các chuyên gia an ninh mạng sử dụng để phân tích kỹ thuật tấn công của Hacker ; qua đó, đánh giá được mức độ an toàn của hệ thống, và có biện pháp kịp thời khắc phục các điểm yếu tồn tại trong hệ thống .

### **2.3.2. Chức năng**

### **- Điều khiển được luồng dữ liệu**

* Khi các mã hiểm độc thâm nhập vào honeynet, sẽ bị kiểm soát các hoạt động.
* Các luồng dữ liệu khi đi vào không bị hạn chế, nhưng khi đi ra ngoài thì sẽ bị hạn chế.



Hình 2.3.2 Minh họa luồng dữ liệu

-Thu nhận dữ liệu: Là quá trình thu nhận những sự kiện xảy ra trong hệ thống, nó có thể là do attacker hay các đoạn mã độc hại gây ra. Mục đích của thu thập dữ liệu đó là ghi nhận toàn bộ những sự kiện diễn ra trong hệ thống. Làm cơ sở cho quá trình phân tích dữ liệu sau này.

-Phân tích dữ liệu: Mục đích chính của honeynet chính là thu thập thông tin. Khi đã có thông tin thì người dùng cần phải có khả năng để phân tích các thông tin này. Để thực hiện tốt công việc này, đòi hỏi người phân tích phải có một kiến thức rất tốt về an ninh mạng, phải am hiểu về các kỹ thuật tấn công mạng. Vì vậy, thông thường người thực hiện phân tích thường là các chuyên gia an ninh mạng.

Thu thập dữ liệu: Trong tường hợp hệ thống triển khai nhiều Honeynet thì phải thu thập dữ liệu từ các honeynet về một nguồn tập trung. Thường thì chỉ có các các tổ chức, trung tâm an ninh mạng lớn có quy mô toàn cầu thì họ mới triển khai nhiều honeynet, đặc biệt là các Công ty cung cấp các sản phẩm diệt virus như: Trend Micro, Symantec….. Còn đa số các tổ chức chỉ có một honeynet.

### 2.3.3 Một số mô hình triển khai Honeynet trên thế giới

## 

1. Mô hình triển khai Honeynet của Đại học Bắc Kinh-Trung Quốc

Hình 2.3.3 - Sơ đồ triển khai dự án Artemis đại học Bắc Kinh, Trung Quốc

## 

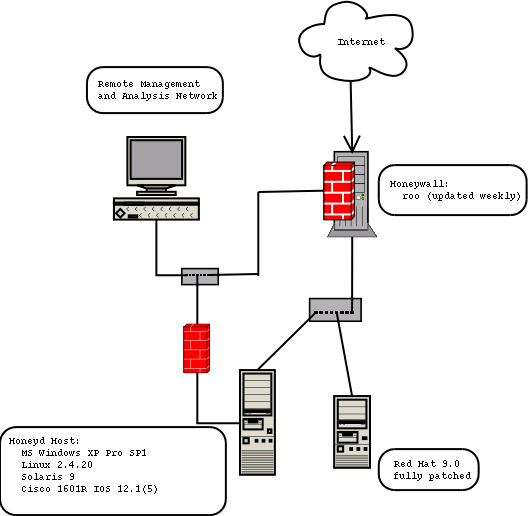
Hình 2.3.3 là sơ đồ triển khai Honeynet tại đại học Bắc Kinh, Trung Quốc trong một dự án có tên là Artemis. Hiện tại, dự án đang triển khai trên nền Honeynet thế hệ thứ III, mô hình triển khai gồm ba honeypot với các hệ điều hành khác nhau: Red Hat Linux9

Windows XP, Windows 2000 và các honeypot ảo được giả lập chương trình honeyd. Và ở mô hình này, Honeywall gồm có 3 card mạng:

* + Card thứ 1 được kết nối với 1 Router bên ngoài
  + Card thứ 2 được kết nối với các Honeypot bên trong
  + Card thư 3 thì được kết nối an toàn với Máy Console

Khi Hacker tấn công vào thì ba Honeypot và Honeypot ảo sẽ tương tác với Hacker, và tiến hành thu thập các thông tin của Hacker như: địa chỉ IP của máy Hacker sử dụng, các tool mà Hacker dùng, cách thức Hacker thâm nhập vào hệ thống……

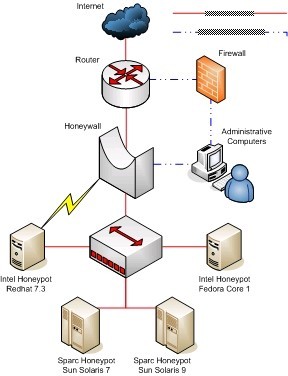
Toàn bộ quá trình tấn công của Hacker sẽ được Honeywall ghi lại và đưa ra các cảnh báo ( Alert ) cho người dùng biết.

Hình 2.3.4 - Sơ đồ triển khai Honeynet của Greek Honeynet Project

Hình 2.3.4 là sơ đồ triển khai Honeynet trong dự án Honeynet tại Hy Lạp, hệ thống Honeynet sử dụng Honeywall phiên bản roo-1.0.hw-189, một honeypot với hệ điều hành Red Hat 9.0 (DNS Server) và bốn honeypot ảo giả lập bằng honeyd các hệ điều hành: MS Windows XP Pro SP1, Linux 2.4.20, Solaris 9 và Cisco 1601R IOS 12.1(5).

Trong mô hình này, Honeywall cũng có ba card mạng, và sơ đồ triển cũng gần giống với mô hình triển khai của Đại học Bắc Kinh nhưng chỉ khác ở chỗ giữa máy Console (Remote Management and Analysis Network ) và bốn máy Honeypot ảo có thêm một Firewall. Firewall này sẽ đảm bảo bảo vệ an toàn cho máy Consle ngay cả khi Hacker kiểm soát được các Honeypot ảo này.

-Mô hình triển khai Honeynet trong dự án Honeynet tại Anh



Hình 2.4.5 - Sơ đồ triển khai Honeynet của UK Honeynet Project

Cuối cùng, hình 2.3.5 mô tả sơ đồ triển khai Honeynet của dự án Honeynet tại Anh. Trong mô hình này, họ đã triển khai bốn Honeypot với các hệ điều hành: Red hat 7.3, Fedora Core 1, Sun Solaris 7, Sun Solaris 9. Mô hình này cũng gần giống với hai mô hình trên; chỉ khác nhau ở chỗ Máy Console ngoài kết nối tới Honeywall thì còn kết nối với Router và được bảo vệ bằng một Firewall đứng giữa.

## 2.3.5 Vai trò và ý nghĩa của Honeynet

Qua các phần trên, ta có thể tóm tắt lại các vai trò và ý nghĩa của Honeynet như sau: Honeynet giúp khám phá, thu thập các phương pháp - kỹ thuật tấn công của

Hacker, các công cụ Hacker sử dụng, đặc biệt là các kỹ thuật tấn công mới , các mẫu virus- mã độc mới….Nhờ đó có những phân tích, định hướng mục tiêu tấn công, thời điểm tấn công, kỹ thuật tấn công,… của Hacker. Từ đó, kịp thời đưa ra các dự báo, cảnh báo sớm để mọi người phòng tránh.

Ví dụ gần đây nhất là vụ cảnh báo của các chuyên gia an ninh mạng thế giới về đợt tấn công của Hacker bằng mã độc sâu (worm) Conficker vào ngày 1/4/2009. Tuy nhiên, do được cảnh báo từ trước và sự nỗ lực của các chuyên gia an ninh mạng quốc tế mà đợt tấn công này đã không diễn ra như mong đợi của Hacker.

Như vậy, Honeynet hoạt động như một hệ thống cảnh báo sớm.

Honeynet là môi trường thử nghiệm có kiểm soát an toàn giúp sớm phát hiện ra các lỗ hổng bảo mật tồn tại trên các sản phẩm công nghệ thông tin đã triển khai - cài đặt trên Hệ thống thật (Đặc biệt là các lỗ hổng Zero – day). Từ đó, sớm có biện pháp ứng phó - khắc phục kịp thời. Đồng thời, honeynet cũng giúp kiểm tra độ an toàn của hệ thống mạng, các dịch vụ mạng ( như : Web, DNS, Mail,…), và kiểm tra độ an toàn - tin cậy - chất lượng của các sản phẩm thương mại công nghệ thông tin khác (đặc biệt là các Hệ điều hành như: Unix, Linux, Window,…).

Thu thập các thông tin, dấu vết của Hacker ( như : địa chỉ IP của máy Hacker sử dụng tấn công, vị trí địa lý của Hacker, thời gian Hacker tấn công,…). Từ đó, giúp chuyên gia an ninh mạng truy tìm thủ phạm.

## .2.3.6.. Khả năng an toàn và các rủi ro

Honeynet có thể là một công cụ rất mạng, chúng ta có thể thu thập thông tin rộng rãi trên nhiều mối đe dọa. Để thu được những thông tin đó, bạn phải cho phép attacker và những chương trình mã độc hại có quyền sử dụng hệ thống, thực thi những hành động của nó, chính điều này sẽ làm cho hệ thống có thể gặp nhiều rủi ro. Các rủi ro này là khác nhau đối với mỗi tổ chức khác nhau, mỗi tổ chức phải xác định được rủi ro quan trọng mà mình có thể bị. Có thể có các loại rủi ro sau:

- Gây thiệt hại cho hệ thống khác

- Rủi ro khi hệ thống bị phát hiện

- Rủi ro khi hệ thống bị vô hiệu hóa

- Các rủi ro khác

### Để giảm thiểu các rủi ro chúng ta phải thực hiện việc giám sát và duy trì hệ thống theo thời gian thực, không được sử dụng các công cụ tự động. Khi triển khai hệ thống chúng ta phải thay đổi những cấu hình mặc định của hệ thống, do các công nghệ honeypot bao gồm cả honeywall đều là mã nguồn mở, mọi người đều có thể tiếp xúc với mã nguồn của nó, trong đó có cả những hacker.

### **2.3.7 Đánh giá so sánh mức độ an toàn**

Trong vô số các biện pháp ngăn chặn tin tặc đột nhập vào hệ thống thì "Honeypot" (tạm gọi là mắt ong) và "Honeynet" (tạm gọi là tổ ong) được coi là một trong nhữngcamj bẫy được kế với mục đích này.

Với hệ thống honeynet, ta có thể xây dựng được một hệ thống mạng với chi phí rẻ, nhưng hiệu quả trong việc bảo vệ mạng máy tính, bảo vệ an toàn thông tin trong mạng

**2.3.8 Kết luận**: Qua chương này, chúng ta đã có những hiểu biết, kiến thức cơ bản về Honeynet cùng với vai trò và mục đích của xây dựng – triển khai Hệ thống này, và chúng ta cũng đã biết một số mô hình Honeynet đã được triển khai trên thế giới . Ở chương sau, chúng ta sẽ tìm hiểu kỹ hơn về mô hình kiến trúc và nguyên lý hoạt động của Hệ thống này.

**2.**4 **MÔ HÌNH KIẾN TRÚC**

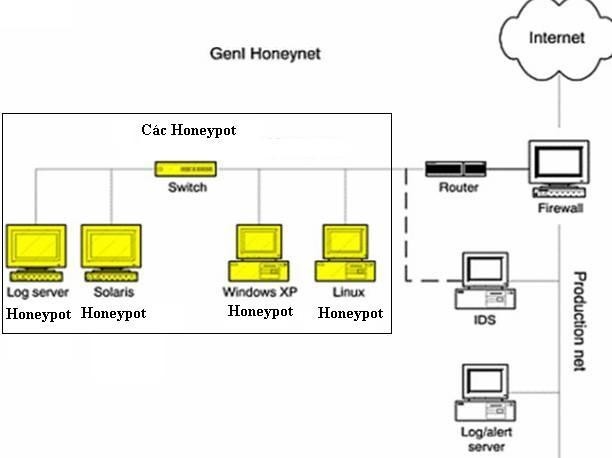
### Mô hình kiến trúc vật lý

### Mô hình kiến trúc Honeynet thế hệ I

Mô hình Honeynet thế hệ I gồm một mạng riêng biệt được tạo ra đặt đằng sau một thiết bị điều khiển truy nhập mạng, thường là tường lửa (Firewall); và bất kỳ luồng dữ liệu vào ra Honeynet đều phải đi qua tường lửa. Honeyney được bố trí trên một mạng riêng biệt với vùng mạng sản xuất để giảm nguy cơ mất an toàn cho hệ thống.

Ở mô hình Honeynet thế hệ I này thì hệ thống tường lửa (Firewall) và Hệ thống phát hiện xâm nhập ( Instruction Detection System – IDS) là hai hệ thống độc lập nhau. Đây chính là sự khác biệt giữa Honeynet I với Honeynet II và Honeynet III. Ở mô hình Honeynet II và III thì hai hệ thống Firewall và IDS được kết hợp thành một hệ thống Gateway duy nhất là Honeywall.

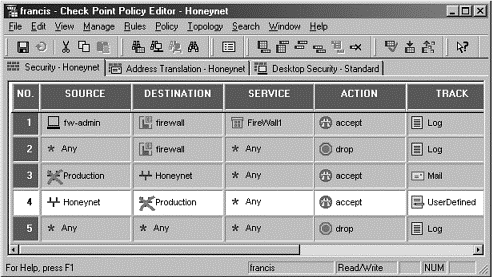
Trong hệ thống Honeynet, Firewall giữ vai trò kiếm soát các luồng dữ liệu ra vào hệ thống, nhằm chỉ cho Hacker tấn công vào Honeynet và ngăn chặn không cho Hacker tấn công vào vùng mạng sản xuất hay không cho Hacker biến Honeynet làm công cụ để tấn công các Hệ thống mạng bên ngoài. Firewall thực hiện được nhiệm vụ này là dựa vào các luật (Rule) định nghĩa sự cho phép (Allow) hoặc không cho phép (Deny ) các truy cập từ bên ngoài vào hoặc bên trong hệ thống ra. Dưới đây là hình minh họa một số luật của Firewall (Check Ponit) đối với Honeynet:



Hình 2.1- Mô hình kiến trúc vật lý Honetnet thế hệ I

Ở mô hình Honeynet thế hệ I này thì hệ thống tường lửa (Firewall) và Hệ thống phát hiện xâm nhập ( Instruction Detection System – IDS) là hai hệ thống độc lập nhau. Đây chính là sự khác biệt giữa Honeynet I với Honeynet II và Honeynet III. Ở mô hình Honeynet II và III thì hai hệ thống Firewall và IDS được kết hợp thành một hệ thống Gateway duy nhất là Honeywall.

Trong hệ thống Honeynet, Firewall giữ vai trò kiếm soát các luồng dữ liệu ra vào hệ thống, nhằm chỉ cho Hacker tấn công vào Honeynet và ngăn chặn không cho Hacker tấn công vào vùng mạng sản xuất hay không cho Hacker biến Honeynet làm công cụ để tấn công các Hệ thống mạng bên ngoài. Firewall thực hiện được nhiệm vụ này là dựa vào các luật (Rule) định nghĩa sự cho phép (Allow) hoặc không cho phép (Deny ) các truy cập từ bên ngoài vào hoặc bên trong hệ thống ra. Dưới đây là hình minh họa một số luật của Firewall (Check Ponit) đối với Honeynet:



Hình 2.2 – Một số luật Firewall đối với Honeynet

Bên cạnh Firewall, Honeynet còn bố trí hệ thống phát hiện xâm nhập IDS-Snort. Snort có nhiệm vụ kịp thời phát hiện và ngăn chặn các kỹ thuật tấn công đã được biết, đã được định nghĩa trong tập luật (Rule) của Snort (Các luật của Snort định nghĩa các dấu hiệu, các mẫu tấn công mạng). Snort thực hiện thanh tra nội dung các gói tin, và so sánh nội dung các gói tin này với tập luật. Khi Snort phát hiện thấy các gói tin có nội dung gây nguy hiểm cho hệ thống mạng thì Snort sẽ chặn các gói tin này lại để ngăn chặn tấn công của Hacker vào hệ thống và đưa ra cảnh báo cho người quản trị biết.

Dưới đây là một ví dụ về cảnh báo của Snort khi phát hiện thấy sự tấn công của sâu Red Code lan truyền trên mạng qua dịch vụ web:

*[\*\*] [1:1256:2] WEB-IIS CodeRed v2 root.exe access [\*\*] [Classification: Web Application Attack] [Priority: 1]*

*12/21-22:07:24.686743 216.80.148.118:2094 -> 10.1.1.106:80*

*TCP TTL:111 TOS:0x0 ID:17545 IpLen:20 DgmLen:112 DF*

*\*\*\*AP\*\*\* Seq: 0xE34143C1 Ack: 0x68B5B8F Win: 0x2238 TcpLen: 20 [\*\*] [1:1002:2] WEB-IIS cmd.exe access [\*\*]*

*[Classification: Web Application Attack] [Priority: 1]*

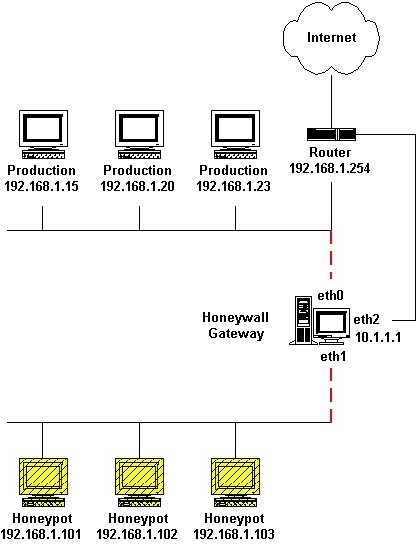
*12/21-22:08:50.889673 216.80.148.118:1864 -> 10.1.1.106:80*

*TCP TTL:111 TOS:0x0 ID:24785 IpLen:20 DgmLen:120 DF*

*\*\*\*AP\*\*\* Seq: 0xEEE40D32 Ack: 0x8169FC4 Win: 0x2238 TcpLen: 20*

### Mô hình kiến trúc Honeynet II, III

1. Honeynet thế hệ II được phát triển vào năm 2002 và Honeynet thế hệ III được đưa ra vào cuối năm 2004. Về cơ bản, Honeynet II và Honeynet III có cùng một kiến trúc. Điểm khác biệt chính là Honeynet III cải tiến việc triển khai và quản lý.
2. Một thay đổi cơ bản trong kiến trúc của Honeynet II và Honeynet III so với Honeynet I là sử dụng một thiết bị đơn lẻ điều khiển việc kiểm soát dữ liệu và thu nhận dữ liệu được gọi là Honeywall (Honeynet Sensor).
3. Honeywall là sự kết chức năng của hai hệ thống tường lửa Firewall và hệ thống phát hiện xâm nhập IDS của mô hình kiến trúc Honeynet I. Nhờ vậy chúng ta dễ dàng triển khai và quản lý hơn.
4. Sự thay đổi trong Honeywall chủ yếu ở module kiểm soát dữ liệu. Honeywall làm việc ở tầng hai (trong mô hình OSI) như là một thiết bị Bridge. Nhờ sự thay đổi này mà Honeynet II, Honeynet III đã khiến cho kẻ tấn công khó phát hiện ra là chúng đang tương tác với Hệ thống “bẫy” Honeynet vì hai đầu card mạng của eth0 (kết nối với mạng bên ngoài Honeynet – phía hacker) và eth1 (kết nối với Honeynet) đều không có địa chỉ mạng IP. Vì vậy, Honeynet hoàn toàn “trong suốt” với Hacker.



Hình 2.3 - Mô hình kiến trúc Honeyney thế hệ II, III

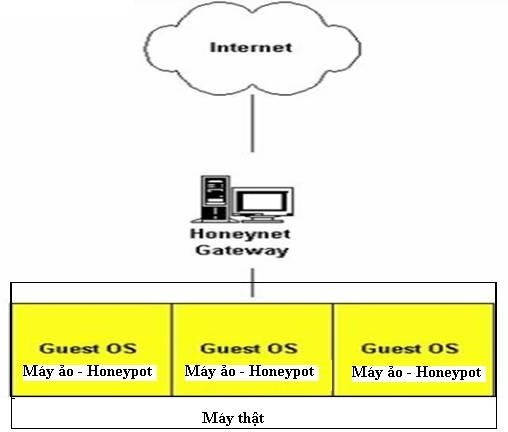
### 1.3,Hệ thống Honeynet ảo

Việc triển khai- xây dựng hệ thống Honeynet yêu cầu từ một lượng lớn thiết bị phần cứng tùy theo quy mô của hệ thống Honeynet mà chúng ta cần triển khai. Nhằm giảm chi phí đầu tư một lượng lớn thiết bị phần cứng trên, người ta đưa ra một mô hình kiến trúc Honeynet mới. Đó là Mô hình kiến trúc hệ thống Honeynet ảo.

Về mặt bản chất, mô hình này vẫn cơ bản giống như Honeynet II và III, vẫn sử dụng một Honeywall Gateway nhưng chỉ khác ở chỗ Honeyney ảo là một mô hình kiến trúc vật lý mới của Honeynet nhằm triển khai hầu như toàn bộ hệ thống Honeynet trên một hệ thống máy đơn ( Máy thật). Mục đích để làm giảm chi phí xây dựng hệ thống Honeynet và dễ dàng cho quản lý.

Hai lựa chọn để triển khai hệ thống Honeynet ảo là sử dụng công cụ phần mền VMWare và User Mode Linux cho phép tạo ra nhiều máy tính ảo trên một hệ thống máy tính thật. Trong đó, VMWare là sản phẩm thương mại, giải pháp được hỗ trợ thiết kế để chạy trên đa môi trường hệ điều hành cùng một lúc. VMWare chỉ chạy trên kiến trúc Intel bởi vậy chỉ các hệ điều hành trên kiến trúc Intel mới làm việc VMWare. Còn User Mode Linux (còn gọi là UML) là giải pháp mã nguồn mở với tính năng tương tự. Tuy nhiên, UML hiện tại đang bị giới hạn cho hệ điều hành Linux.

Bên cạnh những ưu điểm, hệ thống Honeynet ảo cùng một số hạn chế là bị giới hạn hệ điều hành và kiến trúc được hỗ trợ bởi phần mềm.



Hình 2.4 - Mô hình kiến trúc Honeynet ảo

Sơ đồ trên gồm hay máy tính vật lý: Máy tính thứ nhất là Honeynet gateway (cài Honeywall) hoạt động cũng như ở mô hình Honeynet II, III là kiểm soát dữ liệu, thu nhận dữ liệu cho Honeynet. Và trên máy thứ hai thì cài đặt nhiều hệ điều hành máy ảo, mỗi hệ điều hành máy ảo là một honeypot.

Tóm lại: trong các mô hình kiến trúc Honeynet trên thì ngày nay mô hình Honeynet ảo là phổ biến hơn cả. Tuy nhiên, hoạt động của Honeynet trong mô hình này vẫn giống như hoạt động của Honeynet II,III, và cơ bản giống như Honeynet I. Phần trình bày của Đồ án về Mô hình kiến trúc loggic của Honeynet dưới đây sẽ cho chúng ta hiểu rõ về phương thức hoạt động, làm việc của Hệ thống Honeynet

## Mô hình kiến trúc loggic của Honeynet

Dù Honeynet được triển khai – xây dựng theo mô hình nào, ở thế hệ Honeynet nào đi nữa thì Honeynet vẫn có mô hình kiến trúc loggic chung như sau:

## 

**Hệ thống Honeynet**

**Luồng thông tin**

**Chính sách**

**(IPtables + Snort)**

**Lưu trữ dữ liệu**

**Kết quả phân tích**

**Phân tích dữ liệu ( Walley)**

**Thu nhận dữ liệu (Sebek client-server)**

**Điều khiển dữ liệu (Kiểm soát dữ liệu)**

Hình 2.5 - Mô hình kiến trúc logic của Honeynet

Trong một hệ thống Honeynet bao gồm ba module chính:

Module điều khiển dữ liệu ( hay kiểm soát dữ liệu): nhiệm vụ của Module này là kiểm soát dữ liệu vào – ra Hệ thống Honeynet, kiểm soát hoạt động của kẻ tấn công, ngăn chặn kẻ tấn công sử dụng hệ thống mạng Honeynet để tấn công hay gây tổn hại cho các hệ thống bên ngoài khác. Để thực hiện được nhiệm vụ này, Honeynet đã sử dụng hai công cụ chính là Firewall Iptables và IDS-Snort.

Module thu nhận dữ liệu : nhiệm vụ của Module này là thu thập thông tin, giám sát và ghi lại các hành vi của kẻ tấn công bên trong Hệ thống Honeynet. Để thực hiện được nhiệm vụ này, Honeynet đã sử dụng công cụ Sebek client- server.

Module phân tích dữ liệu : nhiệm vụ của Module này là hỗ trợ phân tích dữ liệu thu nhận được nhằm đưa ra: kỹ thuật, công cụ và mục đích tấn công của hacker. Từ đó, giúp

đưa ra các biện pháp phòng chống kịp thời. Và các công cụ Walley, Hflow trong Honeynet sẽ thực hiện được nhiệm vụ này.

Căn cứ vào mô hình kiến trúc logic của Honeynet, ta có thể tóm tắt qúa trình hoạt động của Hệ thống Honeynet như sau:

Đầu tiên, luồng dữ liệu đi vào sẽ được kiểm soát bởi chính sách luật của Firewall Iptables (Firewall Iptables gồm có các luật (Rule) định nghĩa sự cho phép (Allow) hoặc không cho phép (Deny) các truy cập từ bên ngoài đi vào hoặc bên trong hệ thống đi ra, và

kiểm soát các luồng dữ liệu qua Honeywall) và chính sách luật của IDS-snort (hay còn gọi là IDS sensor: gồm có các luật (Rule ) định nghĩa các dấu hiệu tấn công).

Tiếp theo, Module thu thập dữ liệu sẽ sử dụng công cụ Sebek client – server để tiến hành thu thập thông tin. Thông tin thu thập được sẽ được lưu vào trong Cơ sở dữ liệu (Data Store).

Cuối cùng, nhờ sự hỗ trợ của các công cụ Walley, Hflow, Module phân tích sẽ tiến hành thực hiện phân tích nội dung các thông tin thu thập được ở trong Cơ sở dữ liệu. Từ đưa ra kết quả phân tích cho thấy Honeynet có phải đang bị tấn công hay không? Nếu bị tấn công thì kiểu kỹ thuật tấn công (chẳng hạn như: Dos-Ddos, XSS, SQL-injection,….) của kẻ tấn công là gì ? Công cụ Hacker sử dụng là gì?....

Để giúp hiểu kỹ hơn về hoạt động của Hệ thống Honeynet, Đồ án sẽ tiếp tục phân tích kỹ hơn về ba Module này.

## Module điều khiển dữ liệu (hay kiểm soát dữ liệu)

## Vai trò - nhiệm vụ của Module điều khiển

Khi Honeynet không có sự kiểm soát dữ liệu thì Hệ thống sẽ phải đối mặt với những nguy cơ lớn như :

* + - * Kẻ tấn công có thể chiếm được quyền kiểm soát Honeynet và thực hiện các hành vi phá hoại hệ thống.
      * Honeynet bị kẻ tấn công lợi dụng biến thành công cụ để tấn công vào các hệ thống mạng bên ngoài khác…...

Từ đó, đặt ra yêu cầu cần phải có cơ chế kiểm soát dữ liệu, cụ thể là:

* Thứ nhất là cho phép kẻ tấn công tấn công vào bên trong hệ thống Honeynet nhưng phải kiểm soát được các hành vi của kẻ tấn công.
* Thứ hai là ngăn chặn, loại bỏ các tấn công của kẻ tấn công ra bên ngoài.

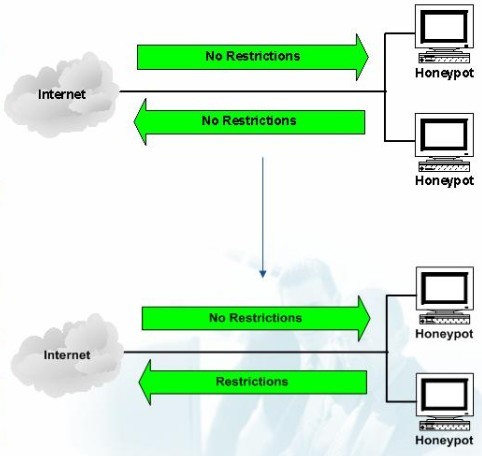
Nhiệm vụ của module điều khiển dữ liệu là ngăn chặn kẻ tấn công sử dụng hệ thống mạng Honeynet để tấn công hay gây tổn hại cho các hệ thống bên ngoài khác.

Khi một honeypot bên trong Honeynet bị hacker kiểm soát, chúng ta phải kiểm chế hoạt động và đảm bảo honeypot không bị sử dụng để gây tổn hại cho các hệ thống khác.

Kiểm soát dữ liệu làm giảm nhẹ nguy cơ đe dọa, nó kiểm soát hoạt động của kẻ tấn công bằng việc giới hạn các luồng thông tin vào/ra trong hệ thống mạng

Nguy cơ đe dọa ở đây, đó là một khi kẻ tấn công gây tổn hại tới hệ thống bên trong Honeynet, chúng có thể sử dụng chính hệ thống Honeynet này để tấn công các hệ thống khác bên ngoài hệ thống Honeynet. Ví dụ một hệ thống nào đó trên Internet. Kẻ tấn công phải bị kiểm soát để nó không thể thực hiện điều đó. Yêu cầu đặt ra là Modul điều khiển dữ liệu phải hoạt động tốt sao cho kẻ tấn công chỉ thực hiện các tấn công vào hệ thống Honeynet mà không gây tổn hại tới các hệ thống khác ở bên ngoài.

- Dưới đây là mô hình kiểm soát dữ liệu:



Hình 2.6 - Mô hình kiểm soát dữ liệu

Với mô hình kiểm soát dữ liệu này thì thông tin đi vào Honeynet không bị hạn chế nhưng thông tin ra đi ra thì lại bị hạn chế, bị kiểm soát chặt chẽ.

### Cơ chế kiểm soát dữ liệu

Việc kiểm soát dữ liệu được thực hiện ngay tại Gateway (Honeywall), và dựa trên hai cơ chế là:

***\**** Một là giới hạn số lượng kết nối ra bên ngoài

\* Hai là lọc gói tin độc hại - Packet Scrubbed.

Để hiểu được cơ chế kiểm soát dữ liệu, chúng ta sẽ đi vào tìm hiểu từng cơ chế này.

## a, Giới hạn số lượng kết nối ra bên ngoài

Cơ chế này cho phép bất kỳ kết nối nào đi vào nhưng lại giới hạn kiểm soát số lượng kết nối ra bên ngoài và khi đạt tới giới hạn thì tất cả các kết nối ra bên ngoài về sau sẽ bị chặn lại. Cơ chế này được thực hiện thông qua sử dụng Firewall IPtables, Firewall phải tính số lượng kết nối ra bên ngoài và khi đạt tới giới hạn nào đó hệ thống sẽ chặn các kết nối vượt quá. Nhờ vậy, mà giảm thiểu nguy cơ kẻ tấn công sử dụng hệ thống Honeynet làm công cụ để thực hiện tấn công vào các hệ thống bên ngoài khác (Bởi vì để thực hiện các hoạt động này đòi hỏi cần sử dụng nhiều kết nối từ trong hệ thống Honeynet ra bên ngoài.).

Việc giới hạn được thiết lập bởi người quản trị, không có một quy tắc giới hạn cụ thể nào cố định cho Module điều khiển dữ liệu, người thiết kế hệ thống căn cứ vào yêu cầu và mục đích của hệ thống để đưa ra các giới hạn phù hợp với tình hình thực tế. Nếu tăng số lượng kết nối ra bên ngoài sẽ cho phép hoạt động tấn công của hacker diễn ra nhiều hơn từ đó chúng ta thu được nhiều thông tin có giá trị hơn song đồng thời cũng gây nhiều nguy hiểm hơn. Còn nếu cho phép ít hoặc không cho kết nối ra bên ngoài, thì sẽ ít nguy cơ hơn song cách này gây ra sự nghi ngờ cho kẻ tấn công và có thể phát hiện ra chúng đang tương tác với hệ thống Honeynet. Sau đó, chúng có thể thực hiện các hành vi phá hoại như: xóa dữ liệu hay đưa vào các thông tin sai lệch nhằm phá vỡ các tính toàn vẹn, tính sẵn sàng của thông tin.

Tóm lại, số lượng kết nối cho phép ra bên ngoài tùy thuộc vào cái mà chúng ta cố gắng tìm hiểu và số lượng nguy cơ mà chúng ta chấp nhận đối mặt.Thông thường, hệ

thống Honeynet cho phép từ 10 đến 20 kết nối ra bên ngoài trong 1 ngày. Số lượng này đối với hacker là khá mềm dẻo giống như mở 1 kết nối ra bên ngoài để tải công cụ hay thực hiện IRC để truyền thông. Song lại đủ số lượng để chặn hầu hết các tấn công ra bên ngoài như tấn công từ chối dịch vụ hay dò quét hệ thống.

## b) Lọc gói tin độc hại (Packet Scrubbed)

Cơ chế này có nhiệm vụ phát hiện ra những luồng dữ liệu gây nguy hiểm cho hệ thống. Cơ chế lọc gói tin độc hại thường được thực hiện bởi hệ thống ngăn chặn xâm nhập mức mạng NIPS (Network Intrustion Prevention Systems), cụ thể ở đây là hệ thống IDS-Snort.

Mục đích của NIPS là để phát hiện và ngăn chặn những tấn công đã biết được đinh nghĩa trong tập các luật (Rule) của NIPS. NIPS thực hiện công việc này bằng phương pháp thanh tra mỗi gói tin khi nó đi qua gateway, nó thực hiện so sánh nội dung gói tin với cơ sở dữ liệu mẫu tấn công có sẵn (Các Rule) nhằm phát hiện ra dấu hiệu tấn công.

Khi phát hiện ra luồng dữ liệu tấn công, hệ thống sẽ thực hiện các biện pháp ngăn chặn tấn công thích hợp. Trên thực tế, NIPS thực hiện ngăn chặn bằng việc thực hiện hai biện pháp sau :

* Thứ 1 là loại bỏ gói tin : thực hiện hủy bỏ gói tin chứa nội dung độc hại không cho đi ra bên ngoài (chặn cuộc tấn công). Biện pháp này thực hiện đơn giản song kém linh hoạt dễ gây nghi ngờ cho hacker.
* Thứ 2 là thay thế, sửa chữa gói tin : thay vì loại bỏ gói tin thì NIPS sẽ thực hiện thay thế nội dung bên trong gói tin khiến nó vô hại đối với hệ thống bên ngoài (vô hiệu hóa cuộc tấn công). NIPS sẽ thay đổi một vài byte bên trong đoạn mã khai thác, làm mất hiệu lực chức năng của nó và cho phép nó tiếp tục đi ra ngoài. Hacker sẽ thấy cuộc tấn công được phát động như ý muốn. Biện pháp này cho phép chúng ta giành được quyền kiểm soát hành vi của kẻ tin tặc tốt hơn đồng thời nó cũng hết sức linh hoạt khiến hacker khó phát hiện hơn.

Tóm lại, Cơ chế lọc gói tin độc hại được thực hiện thông qua hệ thống ngăn chặn xâm nhập mức mạng NIPS (Network Intrustion Prevention Systems), cụ thể ở đây là hệ thống IDS-Snort.

### Kiểm soát dữ liệu trong Honeynet II

Honeynet được phát triển qua ba thế hệ là thế hệ I, II và III. Về mặt bản chất thì cả ba thế hệ Honeynet này đều có cách thức kiểm soát dữ liệu gần giống nhau. Tuy nhiên, Honeynet II,III có những điểm cải tiến nâng cao hơn sao với Honeynet I. Vì vậy, chúng ta sẽ phân tích về Kiểm soát dữ liệu trong Honeynet II để minh họa cho Module kiểm soát dữ liệu của Honeynet. Từ đó, giúp chúng ta thấy được những cải tiến nâng cao của Honeynet II so với Honeynet I.

## Tường lửa IPTABLES

Giới thiệu:

Chương trình tường lửa IPtables là do Netfilter Organiztion viết ra để nhằm tăng tính năng bảo mật trên hệ thống Linux. Iptables cung cấp các tính năng sau:

* Tích hợp tốt với nhân (kernel) của Linux.
* Có khả năng phân tích gói tin (package) hiệu quả.
* Lọc gói tin dựa vào địa chỉ MAC và một số cờ hiệu trong TCP Header
* Cung cấp chi tiết các tùy chọn để ghi nhận sự kiện hệ thống .
* Cung cấp kỹ thuật NAT
* Có khả năng ngăn chặn một số cơ chế tấn công theo kiểu DoS Cơ chế xử lý package trong iptables :

IPtables sẽ kiểm tra tất cả các package khi nó đi qua iptables host, quá trình kiểm tra này được thực hiện một cách tuần tự entry đầu tiên đến entry cuối cùng.

Có ba loại bảng trong iptables:

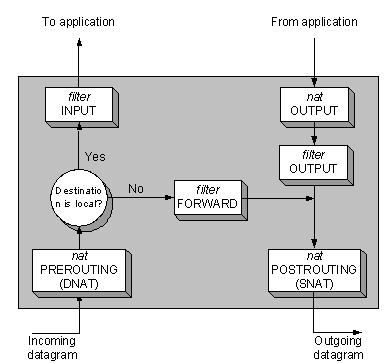
* + Mangle table: chịu trách nhiệm biến đổi quality of service bits trong TCP header.

Thông thường loại table này được ứng dụng trong SOHO (Small Office/Home Office).

* + Filter queue: chịu trách nhiệm thiết lập bộ lọc packet (packet filtering), có ba loại built-in chains được mô tả để thực hiện các chính sách về firewall (firewall policy rules) :
    - Forward chain : Cho phép packet nguồn chuyển qua firewall.
    - Input chain : Cho phép những gói tin đi vào từ firewall.
    - Output chain : Cho phép những gói tin đi ra từ firewall.
  + NAT queue: thực thi chức năng NAT (Network Address Translation), cung cấp hai loại built-in chains sau đây:
* Pre-routing chain : NAT từ ngoài vào trong nội bộ. Quá trình NAT sẽ thực hiện trước khi khi thực thi cơ chế routing. Điều này thuận lợi cho việc đổi địa chỉ đích để địa chỉ tương thích với bảng định tuyến của firewall, khi cấu hình ta có thể dùng khóa DNAT để mô tả kỹ thuật này.
* Post-routing chain: NAT từ trong ra ngoài. Quá trình NAT sẽ thực hiện sau khi thực hiện cơ chế định tuyến. Quá trình này nhằm thay đổi địa chỉ nguồn của gói tin. Kỹ thuật này được gọi là NAT one-to-one hoặc many-to-one, được gọi là Source NAT hay SNAT.

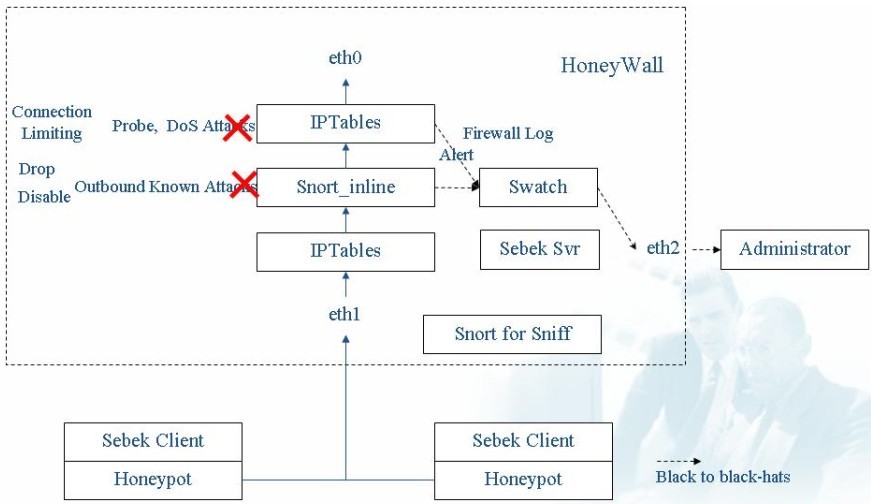
Để có cái nhìn tổng quát đối với việc lọc và xử lý gói trong iptables, chúng ta xem hình sau:

Hình 2.7 – Quá trình lọc và xử lý gói tin của IPtables



Trong Hệ thống Honeynet thì IPtables có vai trò hết sức quan trọng trong việc kiểm soát dữ liệu, thực hiện giới hạn số lượng kết nối ra bên ngoài, chỉ cho Hacker tấn công vào Honeynet nhưng lại ngăn chặn Hacker biến Honeynet thành công cụ, bàn đạp để tấn công vào vùng mạng sản xuất và các hệ thống mạng bên ngoài.

Hình 2.8 dưới đây mô tả quá trình kiểm soát dữ liệu của Honeynet:



Quá trình kiểm soát dữ liệu của IPTABLES: Sau khi Hacker tấn công được vào Honeynet, chiếm được quyền kiểm soát Honeypot thì Hacker sẽ cố gắng tiếp tục sử dụng Honeypot làm công cụ để thực tấn công các Hệ thống mạng bên ngoài như : tấn công từ trối dịc vụ, tấn công dò quét hệ thống,…. Để thực hiện được các tấn công này thì sẽ phải cần phải mở rất nhiều kết nối từ Honeypot ra bên ngoài. Tuy nhiên, do có tường lửa IPTABLES thực hiện nhiệm vụ giới hạn số lượng kết nối ra bên ngoài nên Hacker không thể thực hiện thành công các tấn công này.

## IDS Snort

Giới thiệu:

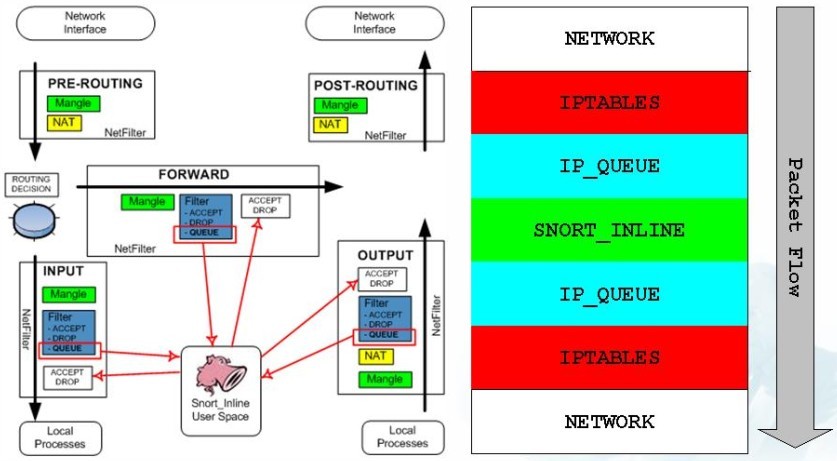
Snort là một trong các sản phẩm an ninh mạng phát hiện xâm nhập (Instruction Detection System – IDS) được sử dụng phổ biến nhất hiện nay. Snort chứa một tập luật định nghĩa dấu hiệu các kỹ thuật tấn công đã được biết. Snort thực hiện thanh tra nội dung các gói tin và so sánh nội dung các gới tin này cơ sở dữ liệu mẫu các tấn công (các Rule). Khi phát hiện thấy có dấu hiệu bị tấn công thì Snort có thể phản ứng bằng nhiều cách khác nhau phụ thuộc vào cấu hình mà chúng ta thiết lập, chẳng hạn như nó có thể gởi thông điệp cảnh báo đến nhà quản trị hay loại bỏ gói tin khi phát hiện có sự bất thường trong các gói tin đó.

Tuy nhiên snort cũng có điểm yếu. Đó là tương tự như các bộ quét virus (virus scanner), snort chỉ có thể chống lại các cuộc tấn công một cách hiệu quả nếu như nó biết được dấu hiệu (signature) của các cuộc tấn công đó. Dựa vào điểm này, các hacker "cao thủ" có thể điều chỉnh các cuộc tấn công để thay đổi signature của cuộc tấn công đó. Từ đó các cuộc tấn công này có thể "qua mặt" được sự giám sát của snort. Như vậy có thể thấy rằng, để snort hoạt động một cách hiệu quả thì một trong những yếu tố quan trọng cần phải chú ý là các luật viết cho snort. Khi snort hoạt động, nó sẽ đọc các tập luật, giám sát luồng dữ liệu chạy qua hệ thống và sẽ phản ứng nếu có bất kì luồng dữ liệu nào phù hợp với tập luật của nó. Cụ thể hơn, tập luật có thể được tạo ra để giám sát các nỗ lực quyét cổng (scanning), tìm dấu vết (footprinting), hoặc nhiều phương pháp khác mà các hacker dùng để tìm cách chiếm quyền hệ thống. Tập luật này có thể được tạo ra bởi người dùng hoặc người dùng có thể truy cập đến trang chủ của snort là: [http://www.snort.org](http://www.snort.org/) để lấy về.

**Vai trò, hoạt động của Snort trong Honeynet:**

Trong việc kiểm soát dữ liệu của Honeynet, Snort đóng vai trò hết sức quan trọng, thực hiện nhiệm vụ lọc gói tin độc hại. Nhìn vào hình 2.3 – sơ đồ kiểm soát dữ liệu của Honeynet ở trên ta thấy: Sau khi Hacker tấn công được vào Honeynet, chiếm được quyền kiểm soát Honeypot thì Hacker sẽ cố gắng tiếp tục sử dụng Honeypot làm công cụ để thực tấn công các Hệ thống mạng bên ngoài như : tấn công từ trối dịc vụ, tấn công dò quét hệ thống,….Kết hợp cùng với IPTABLES, Snort đã thực hiện chặn đứng các cố gắng tấn công này của Hacker bằng cách lọc các gói tin độc hại do Hacker tạo ra. Khi phát hiện ra gói tin độc hại thì Snort sẽ thay đổi nội dung gói tin thành vô hại hoặc chặn các gói tin này lại.

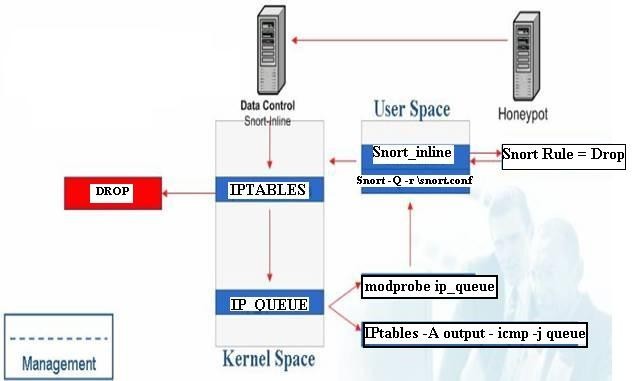
Trong hệ thống Honeynet, người ta đã nâng cấp chức năng của Snort lên một mức cao hơn là Snort\_inline. Snort\_inline cải tiến hơn so với Snort ở chỗ: nó chỉ thanh tra nội dung các gói tin sau khi đã đi qua Iptables (Trong khi đó, Snort lại lắng nghe các gói tin trên interface được chỉ định và dựa vào các luật đã được thiết lập sẵn cho phép xác định có sự xâm nhập hay phá hoại...Vì nó lắng nghe tất cả các packet đến nên việc xử lý khá chậm). Điều này làm cho giảm bớt số lượng các gói tin cần xử lý và tăng tốc độ xử lý. Tường lửa IPtables cho các gói tin đi qua vào hàng đợi (QUEUE) và Snort\_inline mở từng gói tin để “kiểm tra” và “tẩy rửa” các gói tin.Và Hình 2.4 dưới đây sẽ cho chúng ta thấy rõ quá trình hoạt động này của Snort\_inline:



Hình 2.9 - Quá trình hoạt động này của Snort\_inline

Cơ chế loại bỏ gói tin: Hình 2.9 dưới đây cho thấy: trên đường đi của các gói tin từ Honeypot ra bên ngoài thì phải đi qua được sự kiểm tra của tường lửa IPTABLES; cụ thể ở đây, IPTABLES sử dụng các luật của mình để kiểm tra tính hợp lệ của gói tin. Sau đó, IPTABLES đưa các gói tin hợp lệ vào hàng đợi để tiếp tục được Snort\_inline kiểm tra một lần nữa. Snort sẽ thanh tra nội dung các gói tin này và so sánh với các mẫu tấn công đã được lưu trong cơ sở dữ liệu. Khi phát hiện thấy có dấu hiệu tấn công thì Snort sẽ gửi yêu cầu IPtables chặn gói tin này lại không cho ra bên ngoài.

Biện pháp này thực hiện đơn giản song kém linh hoạt dễ gây nghi ngờ cho hacker:



Hình 2.10 - Cơ chế làm việc của Snort\_inline

Cơ chế thay thế gói tin: cơ chế này cơ bản cũng giống như cơ loại bỏ gói tin, chỉ khác ở chỗ: khi phát hiện ra tấn công, thay vì gửi yêu cầu tới IPtables chặn và loại bỏ gói tin thì Snort\_inline sẽ thay thế, sửa chữa nội dung bên trong gói tin khiến nó vô hại đối với hệ thống bên ngoài. Snort\_inline sẽ thay đổi một vài byte bên trong đoạn mã khai thác, làm mất hiệu lực chức năng của nó và cho phép nó tiếp tục đi ra ngoài. Hacker sẽ thấy cuộc tấn công được phát động như ý muốn. Biện pháp này cho phép chúng ta giành được quyền kiểm soát hành vi của kẻ tin tặc tốt hơn đồng thời nó cũng hết sức linh hoạt khiến hacker khó phát hiện hơn.

Tóm lại: Module điều khiển dữ liệu có vai trò hết sức quan trọng của Honeynet, thực hiện kiểm soát dữ liệu đi ra bên ngoài hệ thống, kiểm soát hoạt động của kẻ tấn công, giúp ngăn chặn kẻ tấn công sử dụng hệ thống mạng Honeynet để tấn công hay gây tổn hại cho các hệ thống bên ngoài khác.

### Module thu nhận dữ liệu

### Vai trò - nhiệm vụ của Module thu nhận dữ liệu

Thu nhận dữ liệu nhằm khám phá ra kỹ thuật xâm nhập, tấn công, công cụ và mục đích của hacker. Đồng thời phát hiện ra các lỗ hổng hệ thống. Đóng vai trò vô cùng quan trọng trong Honeynet, không có module thu nhận dữ liệu thì Honeynet sẽ không thể thực hiện mục đích triển khai – xây dựng của mình, không có giá trị.

Module thu nhận dữ liệu thực hiện giám sát và ghi lại các hành vi của kẻ tấn công bên trong Honeyney. Những hành vi đó được tổ chức thành những dữ liệu cơ sở và là cốt lõi của việc nghiên cứu và phân tích. Để có nhiều dữ liệu thu nhận và để thu thập đầy đủ thông tin, chi tiết của các hành vi của kẻ tấn công thì cần phải có nhiều cơ chế thu nhận dữ liệu khác nhau.

Module này sử dụng nhiều cơ chế khác nhau để thu nhận nhiều loại dữ liệu khác nhau. Việc thu nhận dữ liệu có thể được thực hiện bằng nhiều phương thức như:

* Thu nhận dữ liệu từ tường lửa
* Thu nhận dữ liệu từ luồn dữ liệu mạng
* Thu nhận dữ liệu từ hoạt động của honeypot trong hệ thống

Để đảm bảo Honeynet hoạt động tốt thì yêu cầu đối với modul thu nhận dữ liệu

* + - * Thu nhận càng nhiều dữ liệu càng tốt
      * Đảm bảo tính chính xác, sẵn sàng
      * Che dấu đối với hacker

### Cơ chế thu nhận dữ liệu

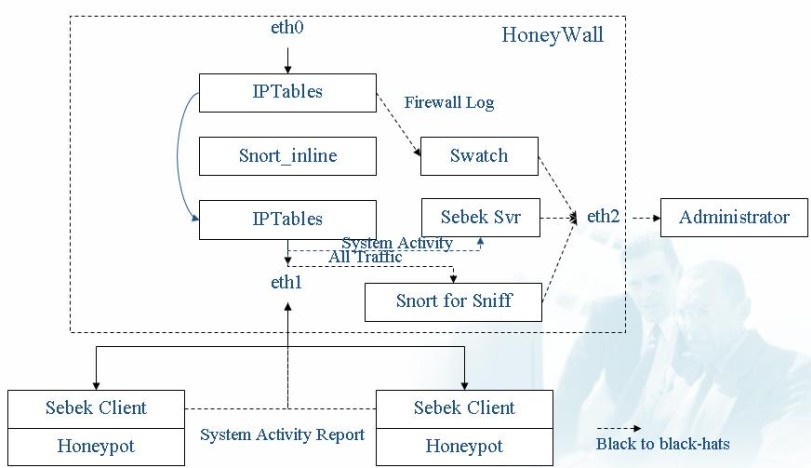
Nhằm đáp ứng các yêu cầu của việc thu nhận dữ liệu, Module thu nhận dữ liệu thực hiện thu nhận dữ liệu trong Honeynet dựa trên ba tầng :

+ Thu nhận từ tường lửa (sử dụng nhật ký của tường lửa – Firewall Log).

+ Thu nhận từ luồng mạng (nhờ công cụ Snort).

+ Thu nhận từ hoạt động Honeypot trong hệ thống (nhờ vào Sebek client –server).

Chúng ta sẽ thấy rõ ba cơ chế này qua sơ đồ thu nhận dữ liệu dưới đây:



Hình 2.11 - Sơ đồ thu nhận dữ liệu

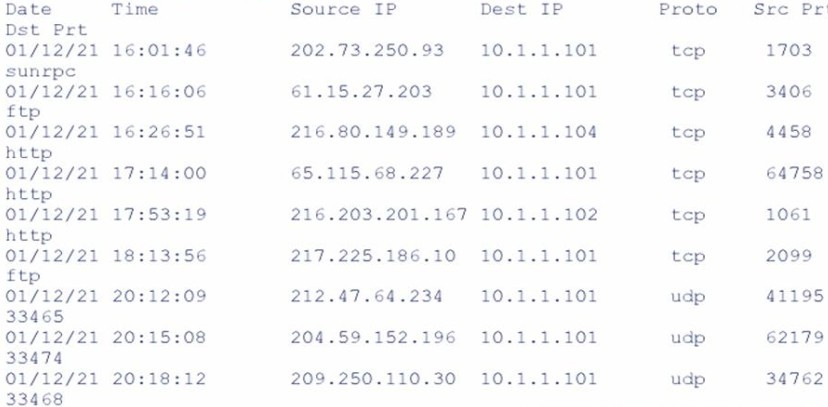
### Thu nhận dữ liệu từ Firewall

Firewall cho phép thu nhận dữ liệu rất tốt bởi vì tất cả luồng dữ liệu đều phải qua nó.

Các thông tin mà Firewall ghi lại bao gồm :

* Địa chỉ IP nguồn của gói tin (có thể địa chỉ IP của máy tính Hacker).
* Địa chỉ IP đích của gói tin (thường là địa chỉ của các Honeypot).
* Giao thức truyền thông được sử dụng (thường là các giao thức truyền thông của các dich vụ mạng mà Honeypot được xây dựng để Hacker tấn công).
* Cổng nguồn của gói tin
* Cổng đích của gói tin ( thường là số cổng của các giao thức mạng mà Honeynet mở để cho phép Hacker tấn công).
* Thời điểm diễn ra cuộc tấn công (dựa trên tem thời gian của gói tin).

Dưới đây là một phần thông tin của file nhật ký (log) trên firewall, ghi lại đầy đủ các thông tin trên :



Hình 2.12 - Nhật ký sử dụng thu nhận dữ liệu trên Honeynet

### Thu nhận dữ liệu từ luồng dữ liệu mạng

Thu nhận dữ liệu từ luồng dữ liệu mạng thực hiện thu nhận mọi gói tin với đầy đủ nội dung payload của gói tin đi vào hay đi ra hệ thống Honeynet. Trong Honeynet tầng thu nhận dữ liệu này được thực hiện bởi Snort (tích hợp vào trong Honeywall) được cấu hình ở chế độ thu nhận tất cả các gói tin trong mạng (thực hiện lắng nghe trên Interface của mạng).

Snort thực hiện việc thu nhận gói tin trong mạng thông qua công cụ hỗ trợ Libpcap (trên Linux) hoặc Winpcap (trên Window) để bắt các gói tin**.** Vai trò quan trọng nhất của Snort là thu nhận tất cả luồng dữ liệu mạng vào-ra hệ thống Honeynet. Snort được sử dụng để bắt và ghi nhận mọi gói tin và payload của gói tin trên đường truyền.

Như sơ đồ thu nhận dữ liệu ở hình 2.7, ta thấy : Snort thực hiện lắng nghe trên toàn mạng để bắt giữ và thanh tra nội dung của tất cả các gói tin qua nó.

### Thu nhận dữ liệu từ hoạt động trên các Honeypot

Nhiệm vụ của modul thu nhận dữ liệu là ghi lại toàn bộ các hoạt động của hacker tương tác với hệ thống Honeynet. Chúng ta có thể chia các hoạt động tương tác thu thập thông tin thành 4 mức sau:

* + Hoạt động mức mạng
  + Hoạt động mức hệ thống
  + Hoạt động mức ứng dụng
  + Hoạt động mức người dùng

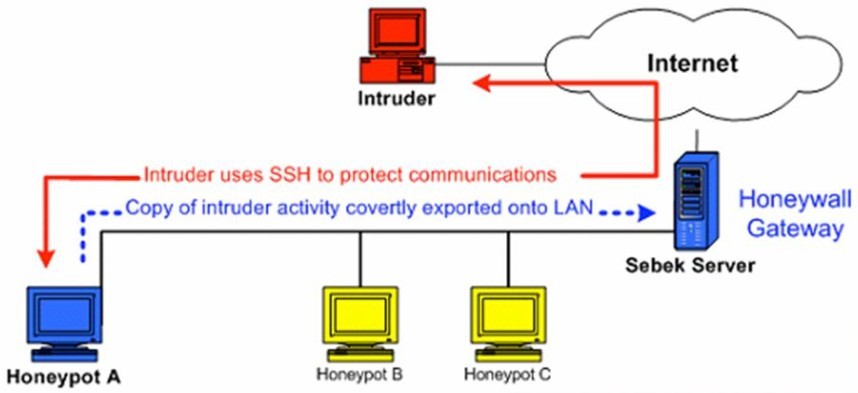
Hai tầng thu nhận dữ liệu được trình bày ở trên (thu nhận dữ liệu từ tường lửa và thu nhận dữ liệu từ luồng mạng) thực hiện thu nhận được các hoạt động mức mạng. Tầng thu nhận dữ liệu thứ 3 này sẽ thu nhận: các hoạt động mức hệ thống, hoạt động mức ứng dụng và hoạt động mức người dùng. Đây chính là tầng thu nhận dữ liệu chủ yếu trong Honeynet.

Để thu nhận được dữ liệu từ các Honeypot, Honeynet đã sử công cụ Sebek client – server thực hiện công việc này. Trong đó, Sebek server đã tích hợp trong Honeywall, còn Sebek client là một chương trình hoạt động như một rookit, được cài đặt trên Honeypot, có khả năng ẩn các tiến trình, file, và cả dữ liệu trong registry (với Windows), ghi lại các thông số về kết nối mạng; thực hiện giám sát tất cả các hoạt động, các kết nối mạng của Honeypot, và báo cáo gửi các thông tin thu thập được về Sebek server.

**Tóm lại:** Sebek là công cụ ghi lại các hoạt động của hacker trên các Honeypot, ghi lại các “keystroke” của hacker, hoạt động theo mô hình client-server.

**Mô hình hoạt động của Sebek:**

Dưới đây là mô hình hoạt động của Sebek:



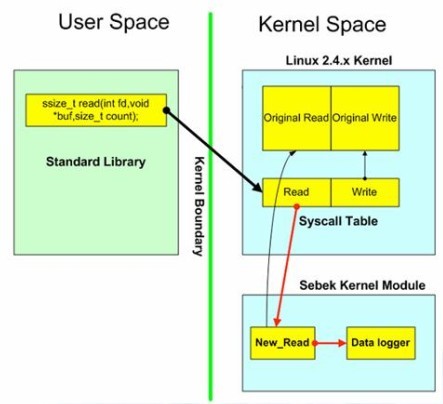
Hình 2.13 - Mô hình hoạt động của Sebek

Sebek hoạt động theo mô hình client–server. Theo mô hình này, thành phần Sebek client được cài đặt trên các Honeypot, còn thành phần Sebek server thì được tích hợp sẵn trên Honeywall. Khi kẻ tấn công thực hành vi xâm nhập vào các Honeypot thì Sebek client sẽ thu thập toàn bộ các thông tin về hoạt động của Hacker và gửi về Sebek server ở Honeywall. Và tại đây, các thông tin thu thập được sẽ được đem ra để phân tích.

Tiếp theo, chúng ta sẽ tiếp tục tìm hiểu chi tiết về vai trò, hoạt động của từng thành phần Sebek client và Sebek server.

**Sebek client :**

* Sebek client sử dụng kỹ thuật rootkit, tức là Sebek client hoạt động như một rootkit, có khả năng ẩn các tiến trình, file, và cả dữ liệu trong registry (với Windows), ghi lại các thông số về kết nối mạng; thực hiện giám sát tất cả các hoạt động, các kết nối mạng của các Honeypot.
* Sebek client được cài đặt trên các Honeypot và nằm hoàn toàn trong nhân của hệ điều hành. Gồm có cả phiên bản cho window (Sebek-Win32-3.0.4.zip) và cho cả Linux (sebek-linux-3.0.3).
* Sebek client thực hiện các nhiệm vụ sau:
  + Thu nhận dữ liệu:
    - Bắt toàn bộ dữ liệu hoạt động thông qua hàm read()
    - Thay thế hàm read() trong System Call Table bằng hàm read() mới
    - Hàm read() mới gọi hàm read cũ đồng thời sao dữ liệu vào bộ đệm gói tin
    - Thêm header vào và gửi tời Sebek Server

Chúng ta sẽ thấy rõ quá trình thu nhận dữ liệu này của Sebek client ở hình dưới đây: 

Hình 2.14 - Sebek client thu nhận dữ liệu

* + Truyền gói tin tới Sebek Server:

Sau khi thu thập được dữ liệu thì Sebek client sẽ gửi các dữ liệu này Sebek server ở trên Honeywall.

Để gửi dữ liệu về Sebek server thì Sebek client đóng gói các dữ liệu thành các gói tin sebek. Các gói tin sebek này đều có đặc điểm là chúng đều có chung một địa chỉ IP mạng đích, địa chỉ IP này là tham số bắt tay liên lạc giữa Sebek client và sebek server; nói cách khác : địa chỉ IP đích trên tất cả các gói tin sebek chính là tham số giúp Honeywall nhận ra các gói tin sebek để bắt giữ các gói tin sebek này. Địa chỉ IP này được khai báo khi cài đặt Honeywall và cả khi cài đặt Sebek client trên Honeypot. Theo mặc định, honeywall đặt IP cho địa chỉ là : 10.0.0.253. Tuy nhiên, ta có thể đổi bằng một địa chỉ IP khác. Để thấy rõ điều này, chúng ta có thể xem phần cấu hình – cài đặt Honeywall và cài đặt Sebek client ở chương IV.

**Sebek Server :**

* Sebek server được cài đặt và cấu hình trên Honeywall, thực hiện nhiệm vụ nhận dữ liêu thu thập được được gửi từ Sebek client về.
* Sebek server sử dụng các công cụ:
  + Công cụ sbk\_extract để thực hiện trích rút dữ liệu sebek từ luồng dữ liệu mạng.
  + Công cụ sbk\_ks\_log.pl trích rút “keystroke” đưa ra đầu ra chuẩn.
  + Công cụ sbk\_upload.pl tải đưa liệu sebek vào cơ sở dưc liệu.

Tóm lại : Module thu nhận dữ liệu thực hiện nhiệm vụ thu nhận dữ liệu nhằm khám phá ra các kỹ thuật xâm nhập, tấn công, công cụ và mục đích của hacker. Đồng thời phát hiện ra các lỗ hổng hệ thống. Đóng vai trò vô cùng quan trọng trong Honeynet, không có module thu nhận dữ liệu thì Honeynet sẽ không thể thực hiện mục đích triển khai – xây dựng của mình, không có giá trị.

## Modul phân tích dữ liệu

## Vai trò

Vai trò của module phân tích dữ liệu trong Honeynet nhằm hỗ trợ người phân thích thực hiện việc sàng lọc, thu gọn dữ liệu nhằm loại bỏ những dữ liệu dư thừa, để dễ dàng tìm ra mối tương quan giữa các dữ liệu nhằm phát hiển ra vấn đề trọng tâm cần phần tích (như dữ liệu liên quan đến quá trình tấn công, xâm nhập hay hoạt động bất hợp pháp của hacker). Hỗ trợ phân tích dữ liệu thu nhận được nhằm đưa ra: kỹ thuật, công cụ và mục đích tấn công của hacker. Từ đó giúp người quản trị đưa ra các biện pháp phòng chống kịp thời.

Đối với hệ thống Honeynet, nếu kẻ tấn công sử dụng Kỹ thuật tấn công mới hay công cụ tấn công mới thì Honeynet lưu giữ lại toàn bộ các dữ liệu về quá trình thực hiện tấn công này của Hacker. Do đó, người quản trị có thể sử dụng các dữ liệu này để phân tích và đưa ra cơ chế, mục đích, công cụ, phương pháp của cuộc tấn công thậm chí có thể xây dựng các mẫu tấn công mới để cập nhật cho hệ thống IDS giúp cho IDS phát hiện ra tấn công mới này nếu tiếp tục gặp lại ở lần sau.

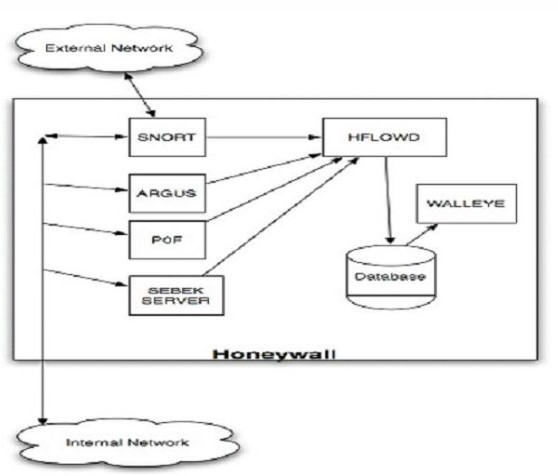
Honeyney cung cấp cho chúng ta một số công cụ như Hflow, Walleye để hỗ trợ người quản trị dễ dàng phân tích, tìm ra cơ chế, mục đích, công cụ và phương pháp tấn công của hacker.

## Cơ chế phân tích dữ liệu

Honeynet hỗ trợ hai công cụ sau để thực hiện quá trình phân tích dữ liệu :

* Một là Hflow: có khả năng tự động kết hợp dữ liệu
* Hai là Walleye: có khả năng báo cáo, thống kê thông qua giao diện web thân thiện với người dùng.

Cả hai công cụ này đều được tích hợp sẵn trên Honeywall. Dưới đây là Sơ đồ kiến trúc Honeywall :



Hình 2.15 - Sơ đồ kiến trúc Honeywall

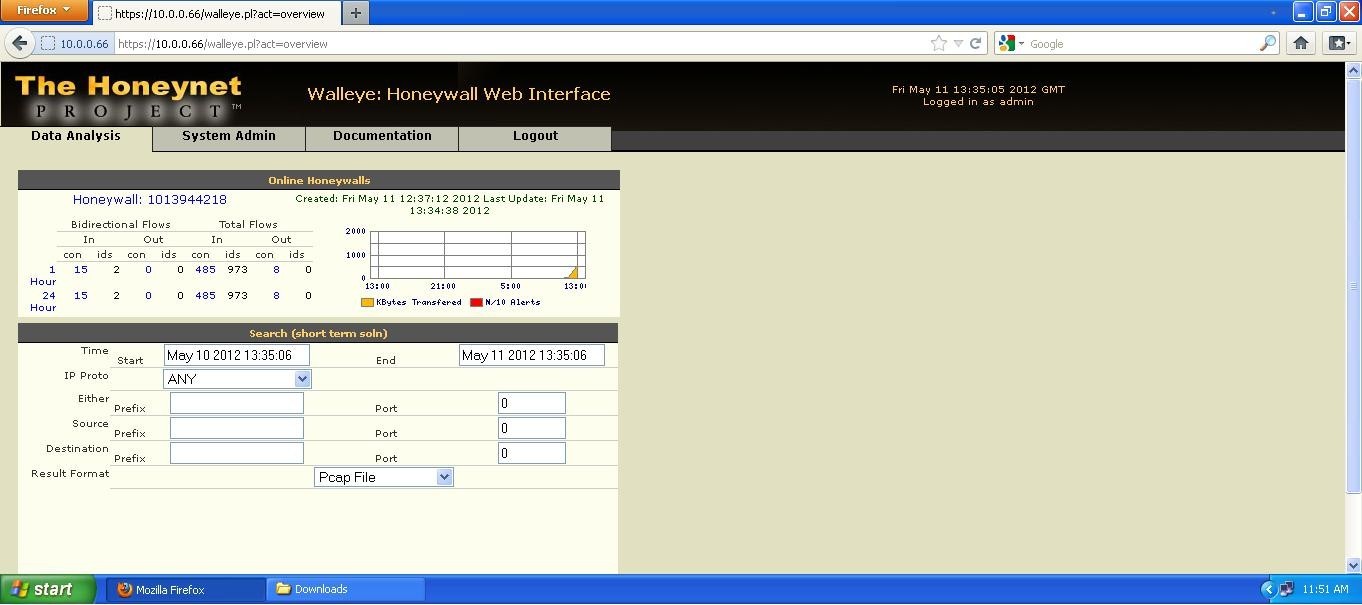
Theo sơ đồ kiến trúc này thì quá trình Hflow và Walleye thực hiện phân tích dữ liệu như sau :

**Hflow:**

Hflow có nhiệm vụ kết hợp dữ liệu từ module thu nhận dữ liệu gửi về, chuẩn hóa dữ liệu sau đó lưu vào cơ sở dữ liệu (ở đây là My SQL).

Hflow tự động xác định :

* Hệ điều hành khởi tạo kết nối mạng.
* Sự kiện IDS liên quan đến kết nối mạng.
* Sự kiện IDS liên quan đến tiến trình và người dùng trên Honetpot.
* Danh sách các tệp liên quan đến cuộc tấn công.
* **Walleye :**
* Walleye có nhiệm vụ lấy dữ liệu thu thập được đã được Hflow chuẩn hóa trong Cơ sơ dữ liệu để cung cấp cho người phân tích thông qua giao diện web.Nhờ vậy, mà người phân tích có thể nắm được khung cảnh chung các hoạt động hệ thống, nắm được chi tiết các hoạt động trong mạng.
* Dưới đây là giao diện của Walleye:



Hình 2.16 - Giao diện của Walleye

**Tóm lại :** Module này có nhiệm vụ hỗ trợ người phân thích thực hiện việc sàng lọc, thu gọn dữ liệu nhằm loại bỏ những dữ liệu dư thừa, hỗ trợ phân tích dữ liệu thu nhận được nhằm đưa ra: kỹ thuật, công cụ và mục đích tấn công của hacker. Từ đó giúp người quản trị đưa ra các biện pháp phòng chống kịp thời.

**Kết luận:** Ở chương này, Chuyên đề đã trình bày về mô hình kiến trúc và nguyên lý hoạt động của Hệ thống Honeynet. Qua đó, giúp chúng ta hiểu sâu hơn về quá trình làm việc, hoạt động của Honeynet.

2.5 các công cụ liên quan

2.5.1 Malwave

Malware là từ viết tắt của malicious software tức là các phần mềm độc hại, chúng thực hiện các hành vi bất hợp pháp trên các thiết bị của người khác vì các lợi ích về kinh tế cũng như chính trị. Đối tượng mà chúng hướng tới không giới hạn ở người dùng máy tính thông thường, thậm chí các công ty, chính phủ và ngay cả internet cũng bị ảnh hưởng nghiêm trọng. Một khi bị nhiễm bởi malware, một máy tính có thể bị chi phối theo các ràng buộc mà người tạo ra nó xây dựng lên, có thể làm lộ các thông tin cá nhân hoặc bị điều khiển để tham gia tấn công DDoS.

Phần mềm anti-virus của McAfee đã thống kê được hơn 20 triệu mẫu malware mới trong năm 2011. Việc phòng chống malware đỏi hỏi sự hiểu biết sâu rộng về cơ chế lây nhiễm của malware cũng như là kỹ thuật chiếm quyền kiểm soát máy tính (rootkit).

Để thu thập các mẫu malware để phân tích thì honeypot là một giải pháp. Honeypot thường giả dạng là các dịch vụ mạng hoặc hệ điều hành có nhiều lỗ hổng. Cách này được chứng minh là thành công trong việc thu hút malware để thu thập số lượng các mạng bị lây nhiễm, cách thức mà malware ẩn mình hay đơn giản là lỗ hổng mà malware khai thác. Mục đích là để các nhà nghiên cứu và các công ty phát triển phần mềm anti-virus phân tích các dạng malware mới để từ đó đưa ra các biện pháp đối phó bằng việc tạo ra các dấu hiệu nhận biết cho các malware chưa từng được biết tới. Có rất nhiều định nghĩa về honeypot nhưng tựu chung là chúng giải quyết vấn đề là làm thế nào để lừa các malware lây nhiễm vào máy của mình.

Một số malware được biết tới nhiều trong thời gian gần đây như Conficker, Stuxnet và Flame không chỉ sử dụng mạng là phương tiện truyền dẫn mà còn có thể nhân bản chính nó vào các thiết bị lưu trữ di động. Theo đó trong bản báo cáo về bảo mật của Microsoft  (Microsoft`s Security Intelligence Report) thì hơn 1/4 số malware được phát hiện bởi Malicious Software Removal Tool trong nửa đầu năm 2011 có khả năng khai thác vào tính năng autorun của Window từ thiết bị USB

## 2.5.2 HONEYPOT CHO MALWARE TRÊN CÁC THIẾT BỊ  USB

Mục tiêu của bất kỳ honeypot là thu thập thông tin của các malware bao gồm thông tin về người viết malware hoặc cách thức lây nhiễm của chúng. Trong trường hợp của bài báo chỉ xét đến trường hợp thu thập mẫu malware ở máy tính bị lây nhiễm đầu tiên.

Có nhiều điểm khác nhau trong vòng đời của malware mà tại đó một honeypot có thể thu thập bản sao của file thực thi malware. Ở đây, “vòng đời” có thể hiểu là quá trình malware lây nhiễm sang máy tính, ẩn mình, thực thi các tác vụ như là gửi spam hoặc thu thập thông tin cá nhân và cuối cùng là lây nhiễm sang các máy tính khác.

Đặc biệt Stuxnet đã chỉ ra rằng cách lây lan qua các thiết bị lưu trữ di động là đặc điểm quan trọng của những malware gần đây. Các thiết bị như USB vừa rẻ lại dễ dàng sử dụng, chúng được sử dụng rộng rãi trong việc trao đổi dữ liệu và đây là môi trường lý tưởng của malware trong việc lây nhiễm vào hệ thống. Stuxnet có thể lây nhiễm vào máy tính thông qua các file được copy trước đó vào USB khi chúng được cắm vào máy tính đã bị lây nhiễm. Hình 1 mô tả quá trình lây nhiễm đó

Yêu cầu đặt ra là xây dựng một honeypot để khai thác các hành vi chung nhất trong quá trình lây nhiễm lên USB bằng cách xây dựng một thiết bị lưu trữ USB ảo mà có thể kết nối tới được các máy tính bị lây nhiễm. Tuy nhiên, cách tiếp cận này có hai nhược điểm:

1. Máy tính mục tiêu bị lây nhiễm trước khi có thể chụp (capture) lại được malware.

2. Các malware thông minh đã lây nhiễm lên máy tính có khả năng phát hiện và phá hủy các thiết bị giả dạng như vậy.

Chúng ta không mong chờ chụp lại được malware trước khi chúng đã phá hủy hệ thống, do đó mục đích của việc xây dựng này là biết được cách thức hoạt động cũng như cách lây nhiễm của chúng. Có thể nói đây là tuyến phòng thủ cuối cùng sau khi các giải pháp bảo vệ trước đó gặp thất bại.

Honeypot được xây dựng ở đây kết hợp các lý thuyết về ổ đĩa flash USB (USB flash driver) ảo hứa hẹn sẽ đem lại một số lợi ích:

1. Cung cấp định nghĩa mới về Host-based IDS ở mức thấp.

2. Nếu malware đã lây nhiễm lên các thiết bị ảo được xây dựng ở đây thì vô hình  malware đã cung cấp cho chúng ta tất cả các file thực thi cũng như hành vi của malware.

Honeypot có thể hoạt động sai, hai kịch bản có thể đặt ra: hoặc là ứng dụng hợp lệ ghi dữ liệu lên các thiết bị USB kết nối tới chúng hoặc người dùng vô tình copy file lên các thiết bị ảo này, vì vậy các thiết bị USB ảo này phải được để ẩn đối với các user của hệ điều hành để ngăn ngừa tình huống ghi dữ liệu như vậy. Quá trình gắn (mount) các thiết bị này càng ngắn thì càng dễ dàng ẩn toàn bộ tiến trình đối với các user. Vì vậy việc thiết kế các honeypot này phải tính toán, thiết kế sao cho thời gian gắn là nhỏ nhất.

Đối với malware, thì các file được ghi vào USB với mục đích là lây nhiễm sang các máy tính khác, vì vậy cần kiểm tra các file thực thi trong toàn bộ quá trình. Thông thường xác suất chúng cung cấp tất cả các file là lớn, và nếu không như vậy thì chí it chúng ta cũng biết được nơi để tìm các dữ liệu còn lại. Về mặt lý thuyết, malware chỉ cần ghi một đoạn mang đặc tính của malware (loader) vào USB sau đó tải về các mailware ảo ở địa chỉ nào đó trên mạng. Nhưng cách tiếp cận này đỏi hỏi máy tính mục tiêu phải có kết nối mạng, các loại malware như vậy chưa được tìm thấy trong thực tế

2.6 Ứng Dụng của Honeypot vào thực tế

Hiện nay trên thế giới đã có rất nhiều tổ chức , cơ quan , đặc biệt là các Công ty – tố chức An ninh mạng đã tiến hành triển khai Hệ thống Honeynet như : Symantec, Trend Micro ,Snort… ; và ngay ở Việt Nam ta thì Trung tâm an ninh mang Bkis cũng đã triển khai Hệ thống này. Hiện tại hoạt động rất tốt, có tác dụng hữu hiệu trong việc giúp các chuyên gia an ninh mạng nghiên cứu và sớm phát hiện ra các lỗ hổng bảo mật tồn tại trên các sản phẩm công nghệ thông tin ; Các kỹ thuật tấn công mạng mới , các mẫu virus- mã độc mới ; giúp truy tìm dấu vết - tung tích các tin tặc ; kiểm tra độ an toàn của hệ thống mạng và qua đó góp phần bảo Hệ thống mạng ngăn chặn sự xâm nhập trái phép của các tin tặc.

# **CHƯƠNG III: KẾT QUẢ THỰC NGHIỆM**

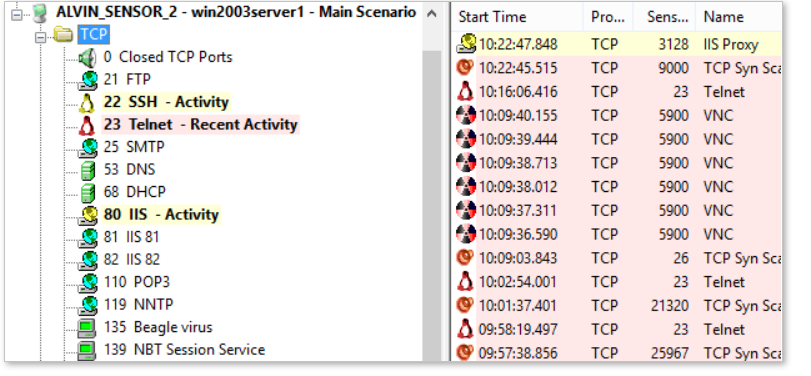
## 3.1 Công cụ sử dụng

### 3.1.1 Công cụ KFsensor

KFSensor hoạt động như một honeypot, được thiết kế để thu hút và phát hiện tin tặc và sâu bằng cách mô phỏng các dịch vụ hệ thống dễ bị tấn công và trojan.

KFSensor được cấu hình sẵn để giám sát tất cả các cổng TCP và UDP, cùng với ICMP. Nó cũng được cấu hình để mô phỏng các dịch vụ thông thường.

Nó bắt đầu giám sát ngay sau khi cài đặt và có thể dễ dàng tùy chỉnh để thêm các dịch vụ khách hàng bổ sung sau này.



Bằng cách phản hồi với một mô phỏng của một dịch vụ thực, KFSensor có thể tiết lộ bản chất của một cuộc tấn công trong khi vẫn duy trì toàn quyền kiểm soát và tránh nguy cơ bị xâm phạm.

Cũng như các cuộc tấn công dịch vụ riêng lẻ, KFSensor phát hiện và phản hồi các cuộc tấn công quét cổng và từ chối dịch vụ DOS và ngăn chính nó bị quá tải.

Bằng cách phản hồi với mô phỏng của một dịch vụ thực, KFSensor có thể tiết lộ bản chất của một cuộc tấn công, đồng thời duy trì toàn quyền kiểm soát sự cố và tránh nguy cơ bị tổn hại.

Cũng như các cuộc tấn công dịch vụ riêng lẻ, KFSensor cũng phát hiện và phản hồi các cuộc tấn công quét cổng và từ chối dịch vụ (DOS); và ngăn chính nó bị quá tải

KFSensor có thể gửi cảnh báo thời gian thực qua email hoặc thông qua tích hợp với hệ thống SEIM.

Bảng điều khiển quản trị KFSensor cho phép các sự kiện được lọc và kiểm tra chi tiết, cho phép phân tích toàn diện bất kỳ cuộc tấn công nào.

KFSensor cũng cung cấp một kết xuất gói đầy đủ để phân tích bổ sung, sử dụng các công cụ như Wireshark.

Mô-đun Báo cáo KFSensor cung cấp một loạt các báo cáo và đồ thị có thể được sử dụng để phân tích nhiều khía cạnh khác nhau của các cuộc tấn công mà một tổ chức phải đối mặt.

Các báo cáo đặc biệt hữu ích trong việc nêu bật các kiểu tấn công chỉ có thể nhận dạng được theo thời gian.

Tất cả các báo cáo có thể được lọc theo khoảng thời gian, loại tấn công và vị trí của khách truy cập, cho phép nghiên cứu và phân tích chi tiết về một mối đe dọa cụ thể.

### 3.2.2 Công cụ FileZilla

**FileZilla Client** (còn được gọi tắt là **FileZilla**) là trình khách FTP [tự do](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_t%E1%BB%B1_do), [mã nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F), [đa nền tảng](https://vi.wikipedia.org/wiki/%C4%90a_n%E1%BB%81n_t%E1%BA%A3ng). Sẵn dùng cho [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux), và [Mac OS X](https://vi.wikipedia.org/wiki/Mac_OS_X). Hỗ trợ [FTP](https://vi.wikipedia.org/wiki/FTP), [SFTP](https://vi.wikipedia.org/w/index.php?title=SFTP&action=edit&redlink=1), và [FTPS](https://vi.wikipedia.org/w/index.php?title=SSH_file_transfer_protocol&action=edit&redlink=1). Tính đến [5 tháng 3](https://vi.wikipedia.org/wiki/5_th%C3%A1ng_3) [2009](https://vi.wikipedia.org/wiki/2009), FileZilla đứng thứ 5 trong số những phần mềm được tải về nhiều nhất từ [SourceForge.net](https://vi.wikipedia.org/w/index.php?title=SourceForge.net&action=edit&redlink=1).[[1]](https://vi.wikipedia.org/wiki/FileZilla_Client#cite_note-1)

Cùng với FileZilla Client là [FileZilla Server](https://vi.wikipedia.org/wiki/FileZilla_Server) (hỗ trợ FTP và FTP trên SSL/TLS), là [trình chủ FTP](https://vi.wikipedia.org/w/index.php?title=Tr%C3%ACnh_ch%E1%BB%A7_FTP&action=edit&redlink=1) hỗ trợ bởi cùng một dự án.

Mã nguồn của FileZilla được lưu trử trên [SourceForge.net](https://vi.wikipedia.org/w/index.php?title=SourceForge.net&action=edit&redlink=1). Vào [tháng 11](https://vi.wikipedia.org/wiki/Th%C3%A1ng_m%C6%B0%E1%BB%9Di_m%E1%BB%99t) [2003](https://vi.wikipedia.org/wiki/2003) dự án được đề cử là *dự án của tháng*.

## Tính năng và hạn chế

Những tính năng chính của Filezilla là:

* **Site manager**: Cho phép người sử dụng có thể tạo một danh sách các trang FTP cùng với các dữ liệu kết nối, như [cổng](https://vi.wikipedia.org/w/index.php?title=Port&action=edit&redlink=1) sử dụng, [giao thức](https://vi.wikipedia.org/wiki/Giao_th%E1%BB%A9c_truy%E1%BB%81n_th%C3%B4ng), tài khoản đăng nhập, [mật khẩu](https://vi.wikipedia.org/wiki/M%E1%BA%ADt_kh%E1%BA%A9u_truy_nh%E1%BA%ADp) đăng nhập,...
* **Message log**: Phía trên của cửa sổ. Hiển thị thông tin của phiên kết nối, các lệnh của trình khách và hồi đáp của trình chủ.
* **File and folder view**: Hiển thị phía dưới **message log**, cung cấp một giao diện đồ họa cho FTP. Người dùng có thể quản lý([tải lên](https://vi.wikipedia.org/wiki/T%E1%BA%A3i_l%C3%AAn_v%C3%A0_t%E1%BA%A3i_xu%E1%BB%91ng), [tải xuống](https://vi.wikipedia.org/wiki/T%E1%BA%A3i_l%C3%AAn_v%C3%A0_t%E1%BA%A3i_xu%E1%BB%91ng)) bằng thao tác kéo thả trực quan.
* **Transfer queue**: Phía dưới cùng của cửa sổ, cho biết trạng thái của các quá trình, trạng thái của các tệp trong hàng đợi.

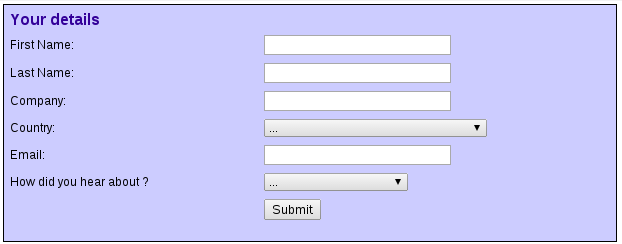
Từ phiên bản 2.2.23, FileZilla sử dụng [Unicode](https://vi.wikipedia.org/wiki/Unicode). Kết quả là nó không còn chạy trên Windows 9x/ME nữa.

Phiên bản hiện tại của FileZilla cho [Mac OS X](https://vi.wikipedia.org/wiki/Mac_OS_X) không còn hỗ trợ Mac OS X v10.4 ‘Tiger’ và các bản trước đó. Running FileZilla with versions older than Mac OS X v10.5 ‘Leopard’ requires compilation

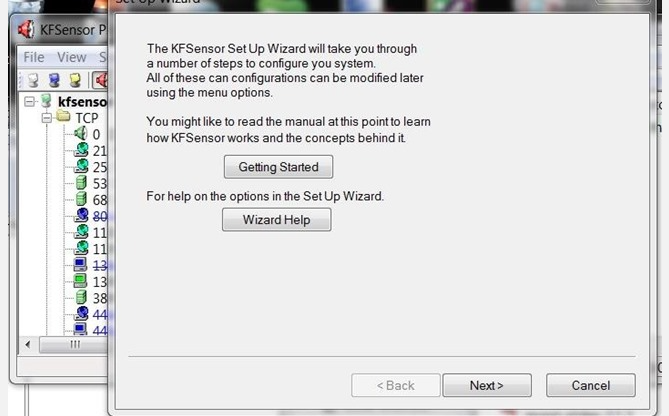
## 3.2 Các bước thực hiện

Bước 1 Cài đặt KFSensor

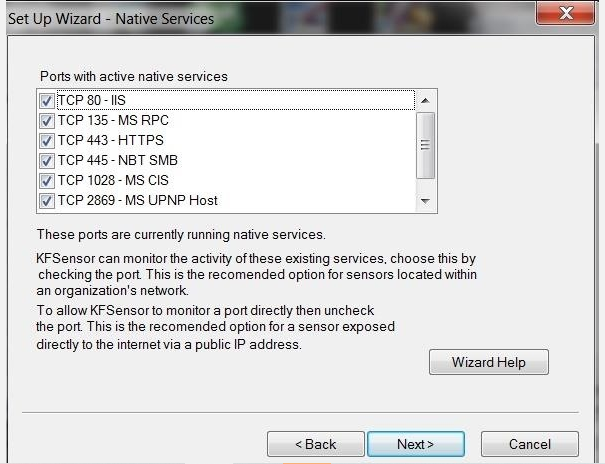
Hiện nay, có khá nhiều các phần mềm cho phép tạo honeypot miễn phí cũng như trả phí như honeynet, honeyd, Tiny honeypot, netbait, mantrap... Trong bài hướng dẫn này, mình sẽ sử dụng phiên bản thương mại của phần mềm **KFSensor**cho Windows.



Chúng ta tiến hành cài đặt như các phần mềm trên Windows khác. Khi hoàn thành, click chuột phải vào icon KFSensor và chọn "run as administrator", chúng ta sẽ nhìn thấy như hình dưới đây

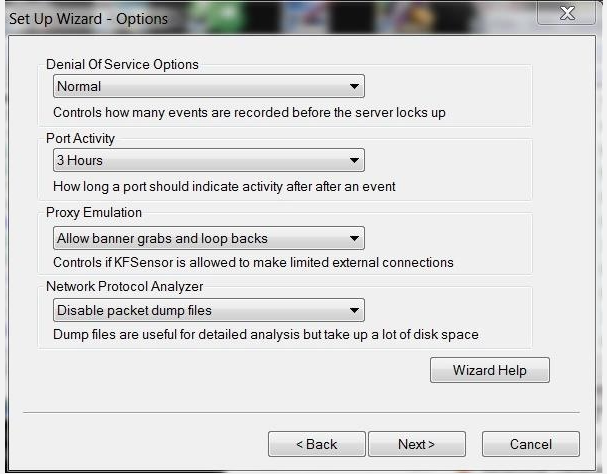


Sau một vài tùy chọn ở chế độ mặc định, chúng ta sẽ tới mục (như hình dưới) cho phép chúng ta chọn dịch vụ chính.

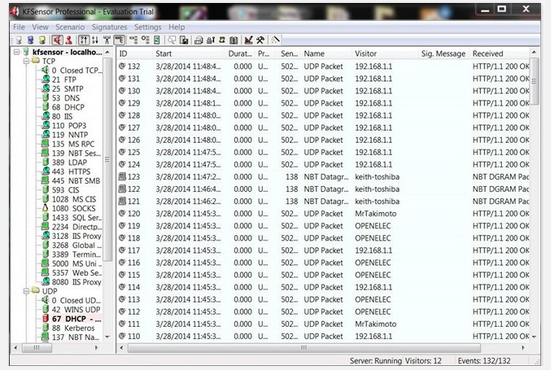


**Bước 2:**Lựa chọn các tùy chọn

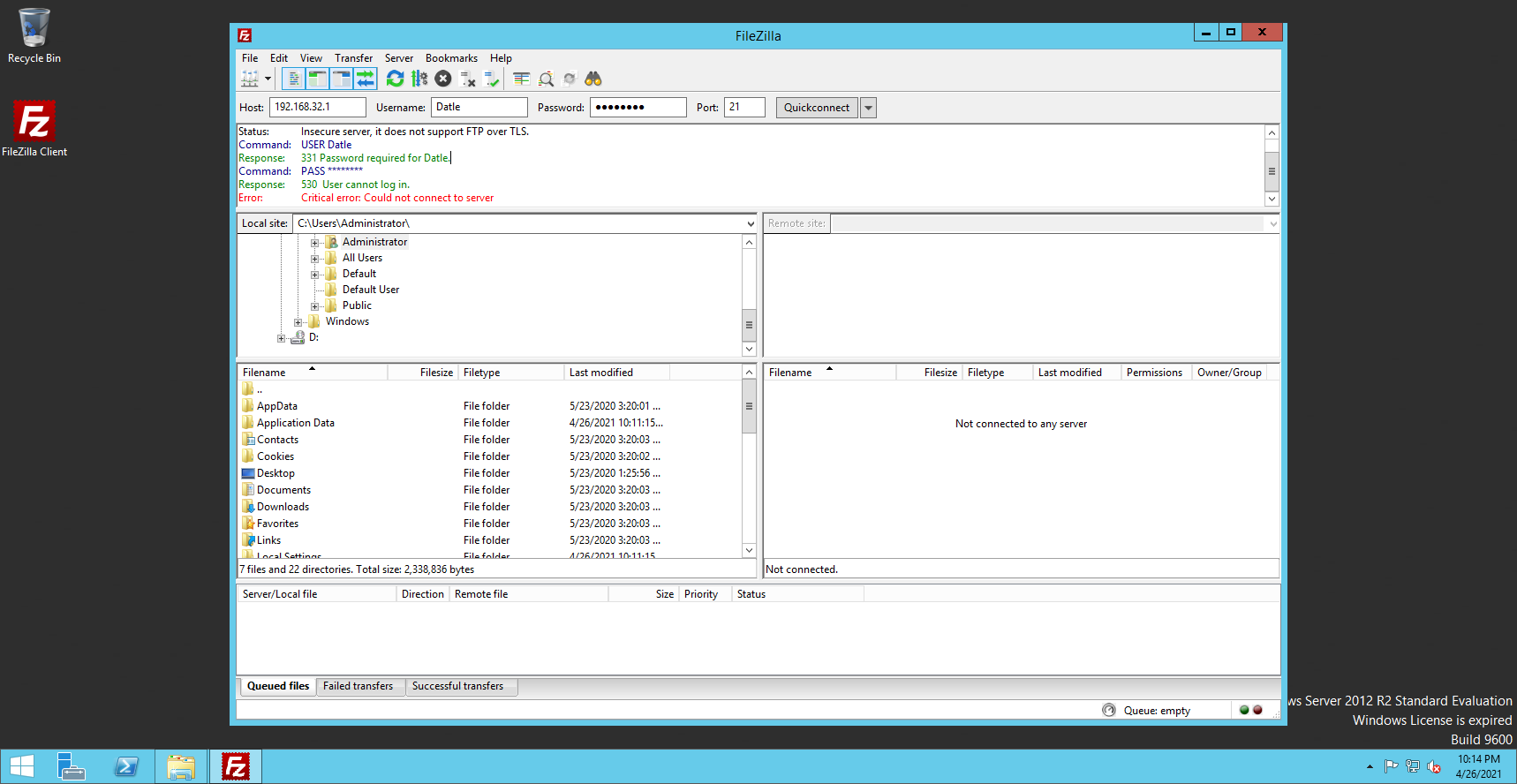
Cuối cùng, chúng ta có một vài tùy chọn để hoàn tất. Ở đây, nó cho phép chúng ta chụp các gói tin do đó chúng ta có thể phân tích tấn công với các công cụ phân tích gói tin như wireshark, tcpdump... Chúng ta cần lưu ý tùy chọn này, bởi nó quá trình thu thập gói tin sẽ chiếm rất nhiều không gian ổ đĩa cứng.



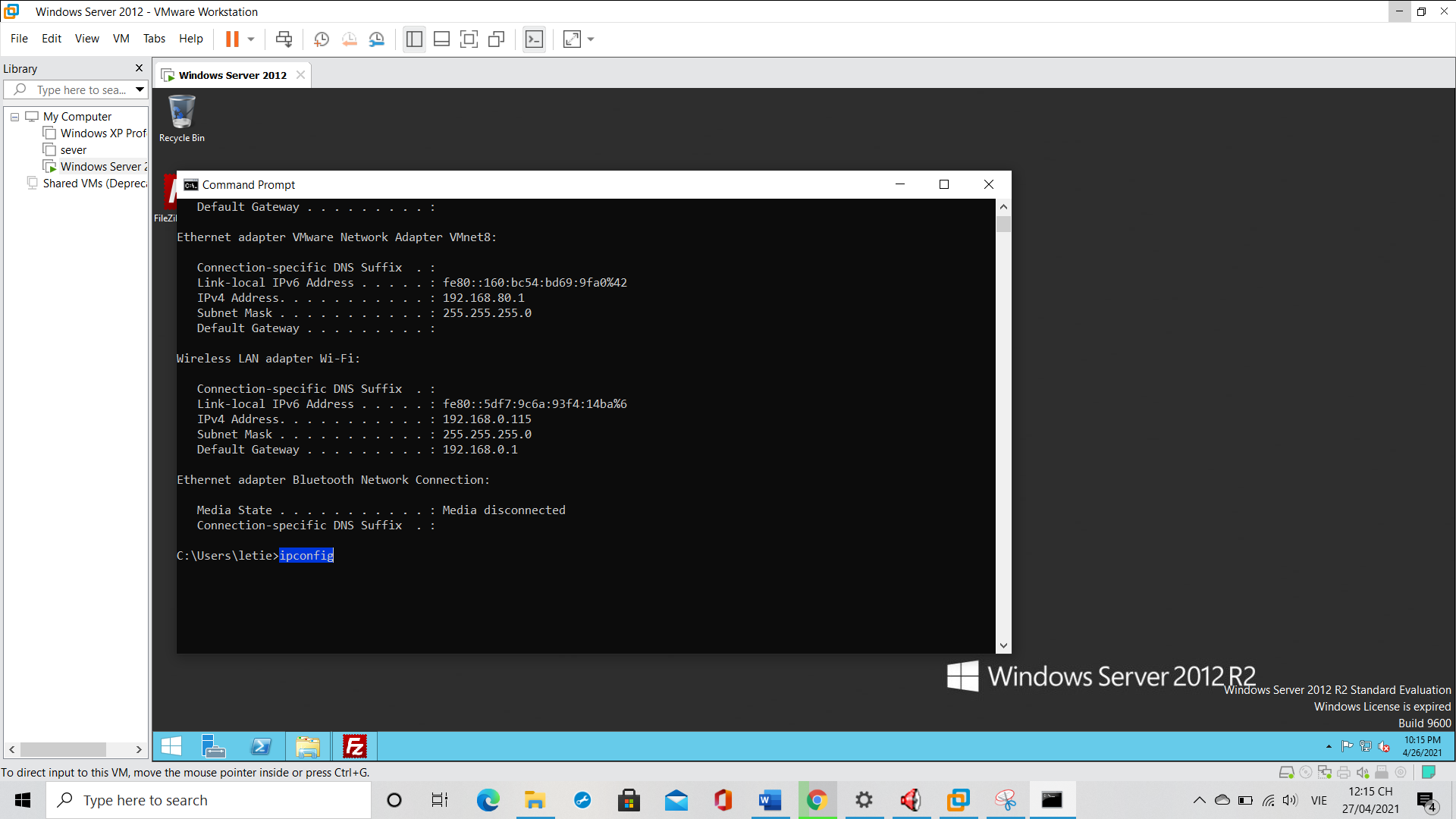
**Bước 3:** Cài đặt honeypot và theo dõi  
Khi chúng ta hoàn tất việc thiết lập các tùy chọn, click Finish.



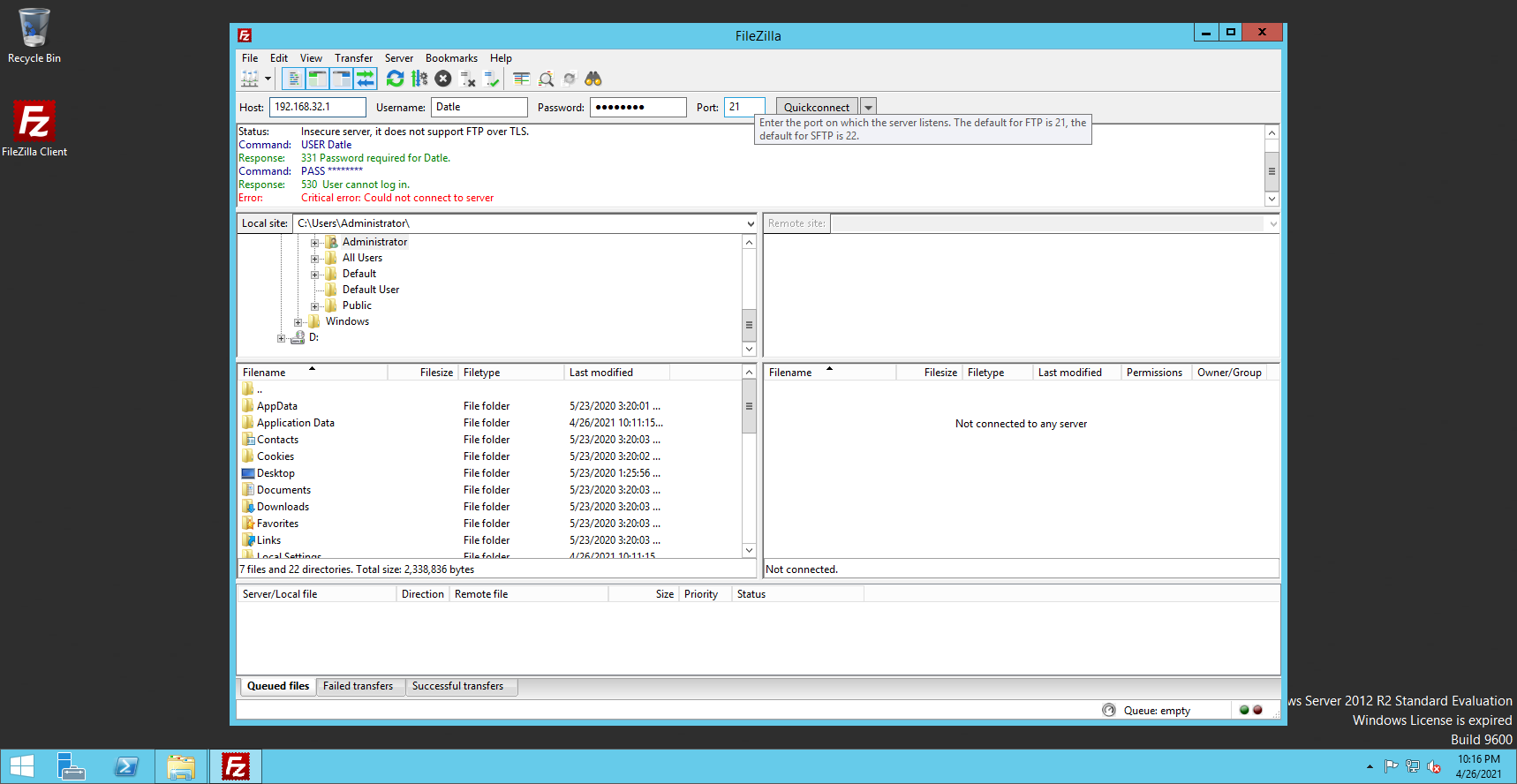
Bước 4 cài công cụ filezilla



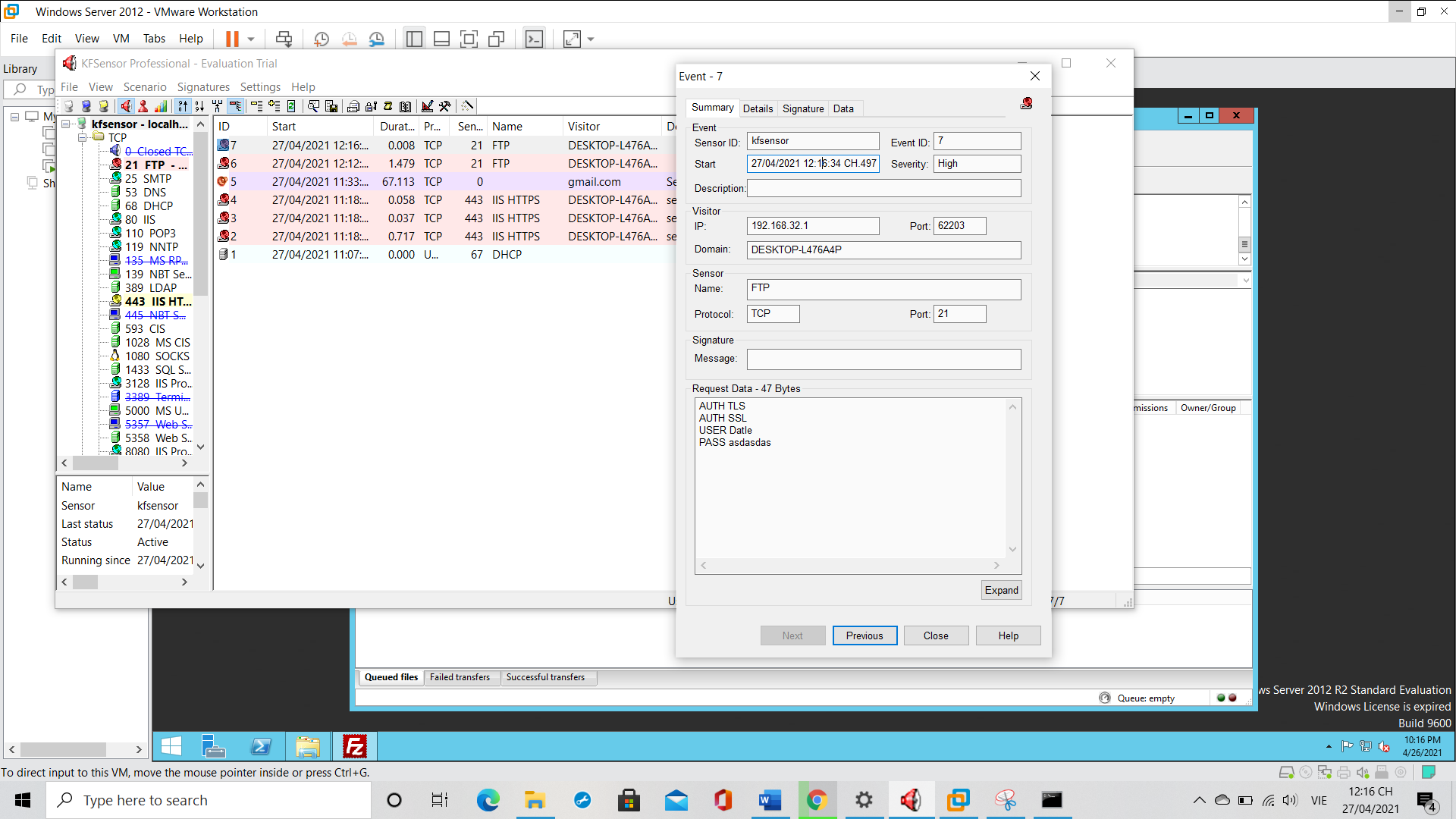
Bước 5 Thực hiện chạy filezilla trên máy ảo Malwave đồng thời vào cmd trên máy ảo để lấy ip của máy bằng lệnh ipconfig



Bước 6 sau khi lấy được ip ta thử scan vào máy thật của chúng ta



Bước 7 Sau khi scan ta thấy bên công cụ Kfsennor đã bắt được thời gian truy cập từ máy ảo qua máy chính và đồng thời thông báo là không thể kết nói



## 3.3 Kết luật

Thông qua các bước thực hiện ở trên chúng ta thấy được Một hệ thống chứa nhiều các dịch vụ vất thường và nhiều cổng mở. Hầu hết, các hệ thống để ngăn ngừa bị thâm nhập, họ sẽ thường đóng các dịch vụ không cần thiết. Do đó, khi scan một hệ thống, với một vài lệnh scan đơn giản mà chúng ta có thể biết được nhiều dịch vụ trên hệ thống thì chúng ta nên đề phòng. Khi tấn công hệ thống, bạn thấy rằng có quá nhiều các thiết lập mặc định hoặc có rất ít các hoạt động trên hệ thống này hoặc có quá ít các phần mềm trên hệ thống hoặc quá ít không gian lưu trữ, thì đây cũng có thể là một honeypot Nếu chúng ta tìm thấy một thư mục chứa các thông tin quan trọng như số thẻ tín dụng, security code... dưới dạng clear-text trên server, chúng ta cần kiểm tra kỹ lưỡng bởi rất ít hệ thống để các thông tin này dưới dạng clear-text.

# CHƯƠNG IV:KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 4.1 Kết luận

Trong quá trình nghiên cứu thực hiện Đồ án tố “Nghiên cứu Hệ thống Honeypot”, dưới sự hướng dẫn tận tình của thầy ........., em đã nghiên cứu và nắm vững được hoạt động, mục đích của Hệ thống Honeypot. Có khả năng triển khai áp dụng và phát triển Honeypot vào thực tế. Do điều kiện về thời gian và thiết bị triển khai còn thiếu nên việc nghiên cứu triển khai xây dựng Hệ thống Honeypot và ứng dụng thực tế của Honeypot của em chỉ giới hạn ở phạm vi thí nghiệm. Mặc dù đã rất cố gắng nhưng Đồ án này không tránh khỏi có những sai xót. Em rất mong được sự góp ý, giúp đỡ nhiệt tình của các thầy cô và các bạn để đề tài này của em được hoàn thiện.Em xin chân thành cảm ơn.

4.2 Hướng phát triển

Tìm hiểu thêm về các cách phòng thủ bằng Honeypot, Tìm hiểu các cách để tránh xa Honeypot, tìm hiểm thêm về bảo mật thiết bị OIT bằng hệ thống honeypot

## 4.3Tài Liệu Tham Khảo

Dưới đây là một số link tham khảo : [*http://www.tracking-hackers.com/papers/honeypots.html*](http://www.tracking-hackers.com/papers/honeypots.html)[*www.honeynet.org*](http://www.honeynet.org/)

[*http://www.icst.pku.edu.cn/honeynetweb/honeyneten/HoneynetTopology.htm*](http://www.icst.pku.edu.cn/honeynetweb/honeyneten/HoneynetTopology.htm)[*http://www.honeynet.org.gr/reports/apr2005-sept2005.html*](http://www.honeynet.org.gr/reports/apr2005-sept2005.html)[*http://www.honeynet.org/papers/phishing/details/index.html*](http://www.honeynet.org/papers/phishing/details/index.html)[*http://www.exploit-db.com/webapps/*](http://www.exploit-db.com/webapps/)

[*http://vietcloud.net/thong-tin/91/0/top-10-ky-thuat-tan-cong-tren-web.aspx*](http://vietcloud.net/thong-tin/91/0/top-10-ky-thuat-tan-cong-tren-web.aspx)