MỤC LỤC

DANH MỤC HÌNH

CHƯƠNG I: TỔNG QUAN VỀ HỆ THỐNG HONEYNET

Chương này sẽ trình bày kiến thức tổng quan, cơ bản về Honeynet bao gồm: nguồn gốc, quá trình phát triển của Honeynet, các khái niệm về Honeypot, Honeynet, phân loại Honeypot và chức năng, vai trò, ý nghĩa của Honeynet trong nhiệm vụ đảm bảo an ninh mạng, cùng với một số mô hình triển khai trên thế giới.

1.1.HONEYPOT

1.1.1.Khái niệm

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,…

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 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 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

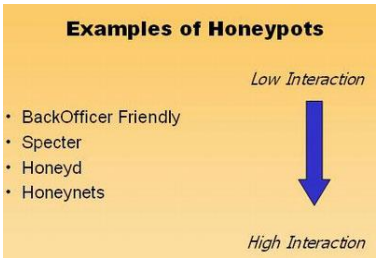
đặt chạy trên bất cứ hệ điều hành nào như: Linux (Red hat, Fedora…), Unix(Solaris), Window (Window NT, Window 2000, Window XP, Window 2003, Vista,…..), Honeypot sẽ trực tiếp tương tác với tin tặc và tìm cách khai thác thông tin về tin tặc như hình thức tấn công, công cụ tấn công hay cách thức tiến hành thay vì bị tấn công.

1.1.2.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.

Đến đây thì cũng đã thấy rõ được ưu và khuyết điểm giữa hai mô hình. Mô hình tương tác cao làm cho mục đích triển khai hệ thống Honeynet ngày càng hoàn thiện đúng với nhiệm vụ của nó là thu thập tất cả phương pháp, kỹ thuật tấn công, công cụ mà hacker sử dụng. Do đó trong đề tài này nhóm đã chọn mô hình tương tác cao vào việc triển khai thực tế, cụ thể là Honeynet thế hệ thứ ba.



Hình 1.1.2 - Các loại hình Honeypot

Một số ví dụ về các loại honeypot :

a. BackOfficer Friendly (BOF):

Là một loại hình Honeypot rất dễ vận hành và cấu hình, 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…

b. Specter:

Đây cũng là loại hình Honeypot tương tác thấp nhưng có khả năng tương tác tốt hơn so với BackOfficer, loại Honeypot này có thể giả lập trên 14 cổng 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.

c. 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.

1.2.HONEYNET

1.2.1.Khái niệm

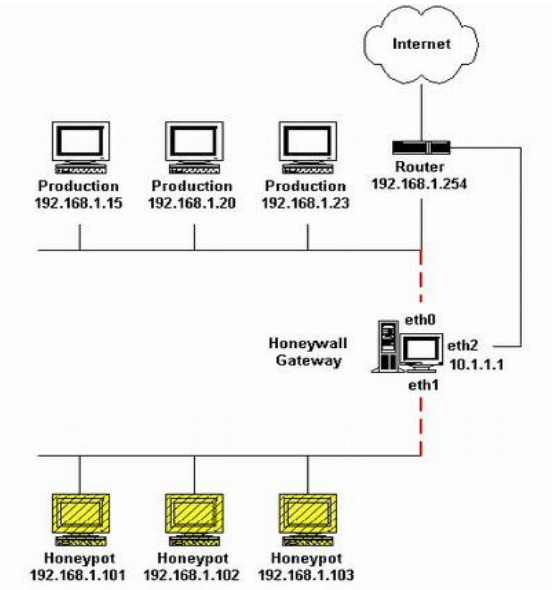
Một trong các công cụ chính mà nhóm dự án Honeynet sử dụng để thu thập thông tin là Honeynet. Honeynet khác với các hệ thống Firewall, hệ thống phát hiện và ngăn chặn xâm nhập: các hệ thống tuy đều có khả năng bảo vệ hệ thống mạng và tài nguyên mạng nhưng các hệ thống này đều là thực hiện nhiệm vụ “phòng 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 , 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.

Quan trọng nhất khi xây dựng một hệ thống Honeynet chính là Honeywall. 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.



Hình 1.2.1 - Mô hình kiến trúc Honeynet

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 eth2 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, kỹ thuật 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 , 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 .

1.2.2.Các chức năng của Honeynet

a. Điều khiển dữ liệu

Khi hacker sử dụng các mã độc (như : virus, trojan, spyware, worm,…) để thâm nhập vào hệ thống Honeynet, thì hai công cụ IDS Snort và Firewall Iptable ở trên Honeywall sẽ thực hiện kiểm soát các hoạt động của các loại mã độc này, cũng như các hành vi mà hacker thực hiện trên hệ thống, đồng thời đưa ra các cảnh báo cho người quản lý hệ thống biết để kịp thời xử lý.

Các luồng dữ liệu khi đi vào không bị hạn chế, nhưng khi đi ra ngoài thì sẽ bị hạn chế . Chính vì vậy mà hacker sẽ rất khó khăn, thậm trí nếu hệ thống Honeynet được cấu hình tốt thì hacker sẽ không thể thu thập được đầy đủ thông tin về hệ thống, điều này cũng có nghĩa là hacker sẽ không thể xâm nhập thành công vào hệ thống mạng.

b. Thu nhận dữ liệu

Khi dữ liệu đi vào thì Honeynet sẽ xem xét và ghi lại tất cả các hoạt động có tính phá hoại và sau đó sẽ phân tích các động cơ hoạt động của tin tặc và chính công cụ Snort trên Honeywall thực hiện chức năng này. Dựa trên các luật (rule) định nghĩa dấu hiệu tấn công mà Snort sẽ cho rằng một hoạt động có được coi là hoạt động có tính phá hoại hay không, nếu phải nó sẽ thực hiện ghi lại log và đưa ra các cảnh báo. Nhờ vậy, mà toàn bộ quá trình tấn công của hacker đều sẽ được ghi lại một cách chi tiết.

c. 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.

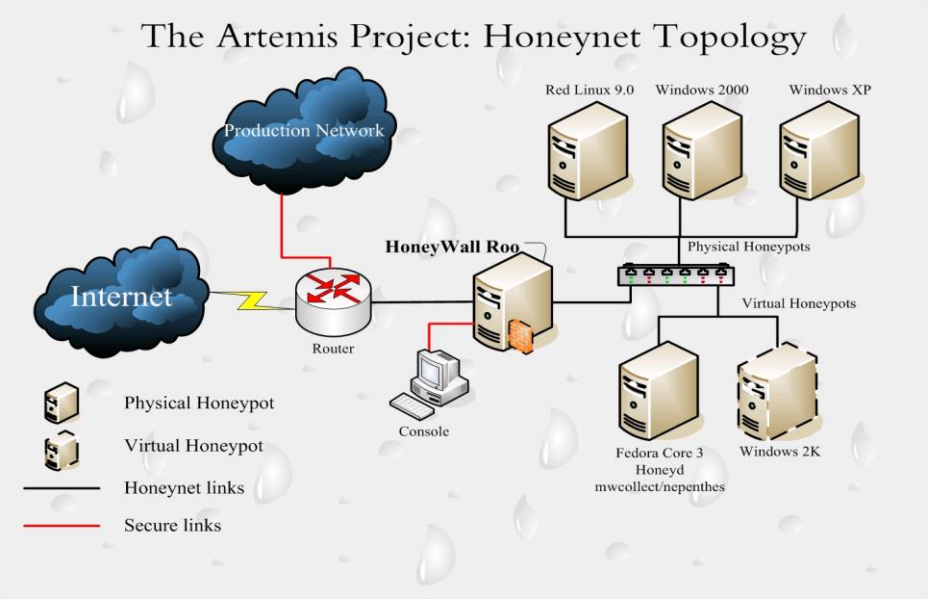
d. Thu thập dữ liệu

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.

1.2.3.Một số mô hình triển khai Honeynet trên thế giới

Tiếp theo là một số mô hình triển khai hệ thống Honeynet trên thế giới nhằm nghiên cứu, thu thập thông tin kỹ thuật tấn công của hacker trên mạng.

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



Hình 1.2.3a - Sơ đồ triển khai dự án Artemis đại học Bắc Kinh, Trung Quốc

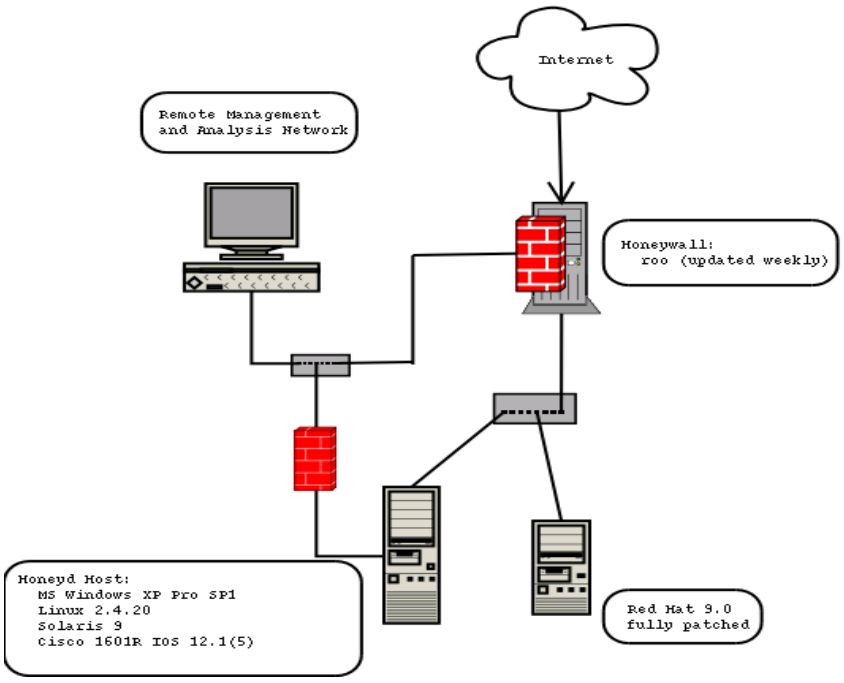
Hình 1.2.3a là sơ đồ triển khai Honeynet tại đại học Bắc Kinh, Trung Quốc trong một dự án có tên là Artemis. Hiện tại, dự án đang triển khai trên nền Honeynet thế hệ thứ III, mô hình triển khai gồm ba honeypot với các hệ điều hành khác nhau: Red Hat Linux9.0, Windows XP, Windows 2000 và các honeypot 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 thứ 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 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.

b. Mô hình triển khai Honeynet trong dự án Honeynet tại Hy Lạp

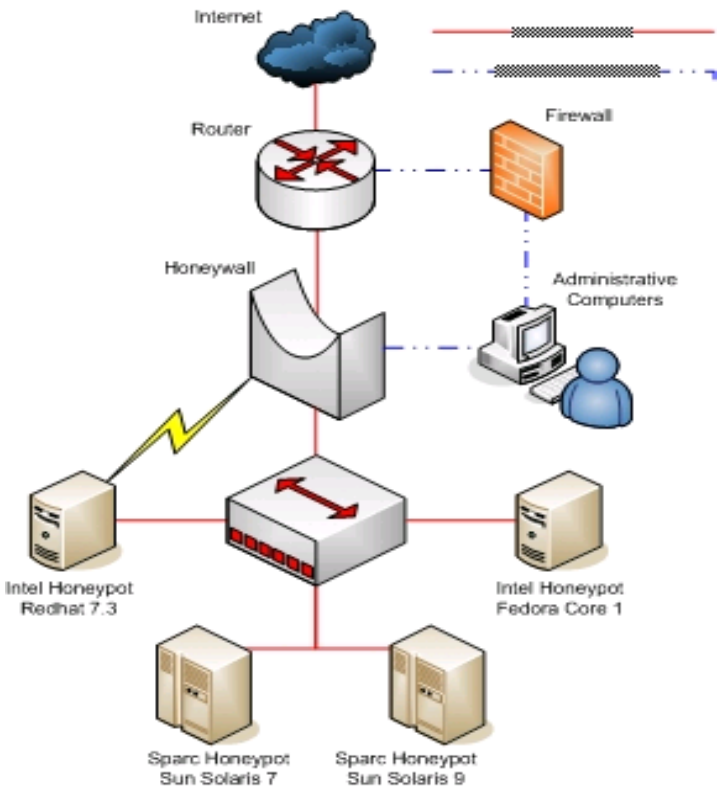


Hình 1.2.3b - Sơ đồ triển khai Honeynet của Greek Honeynet Project

Hình 1.2.3b 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 Cisco1601R IOS 12.1.

Trong mô hình này, Honeywall cũng có ba card mạng và sơ đồ triển khai 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 Console ngay cả khi hacker kiểm soát được các Honeypot ảo này.

c. Mô hình triển khai Honeynet trong dự án Honeynet tại Anh



Hình 1.2.3c - Sơ đồ triển khai Honeynet của UK Honeynet Project

Cuối cùng là 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.

1.3.VAI TRÒ VÀ Ý NGHĨA CỦA HONEYNET

Qua các phần trên, ta có thể tóm tắt lại vai trò và ý nghĩa của Honeynet như sau:

- Honeynet giúp tìm hiểu, 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.
- 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.

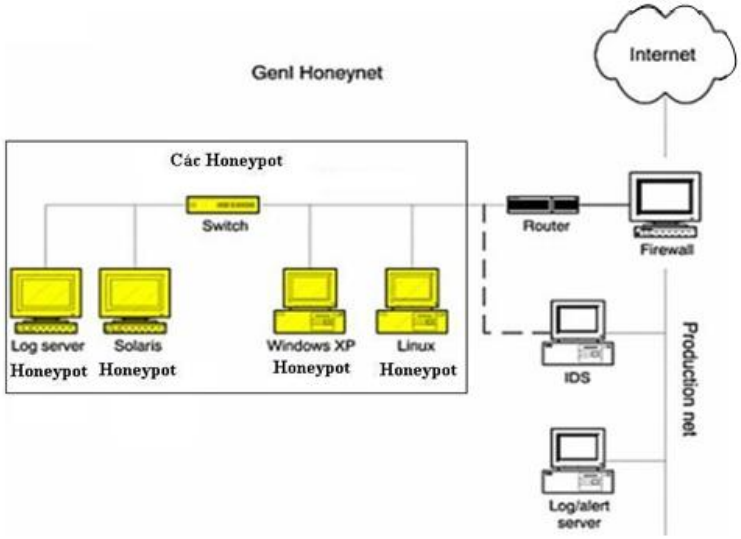
Kết luận: qua chương này, chúng ta đã có những hiểu biết, kiến thức cơ bản về Honeynet cùng với vai trò và mục đích của việc xây dựng - triển khai hệ thống này và chúng ta cũng đã biết một số mô hình Honeynet đã được triển khai trên thế giới.

CHƯƠNG II: MÔ HÌNH KIẾN TRÚC HONEYNET

2.1.MÔ HÌNH KIẾN TRÚC VẬT LÝ

2.1.1.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. Honeynet được bố trí trên một mạng riêng biệt với vùng mạng sản xuất để giảm nguy cơ mất an toàn cho hệ thống.



Hình 2.1.1 - Mô hình kiến trúc vật lý Honeynet thế hệ I

Ở mô hình Honeynet thế hệ I này thì hệ thống tường lửa (Firewall) và hệ thống phát hiện xâm nhập (Intrusion 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 kiểm 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.

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 nhiệm vụ kiểm 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.

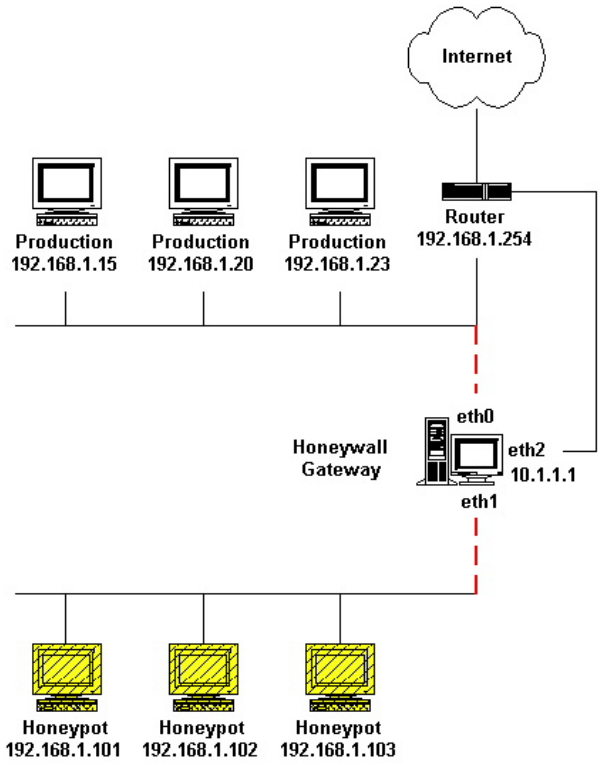
2.1.2.Mô hình kiến trúc Honeynet thế hệ II, III

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.

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).

Honeywall là sự kết hợp 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.

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.



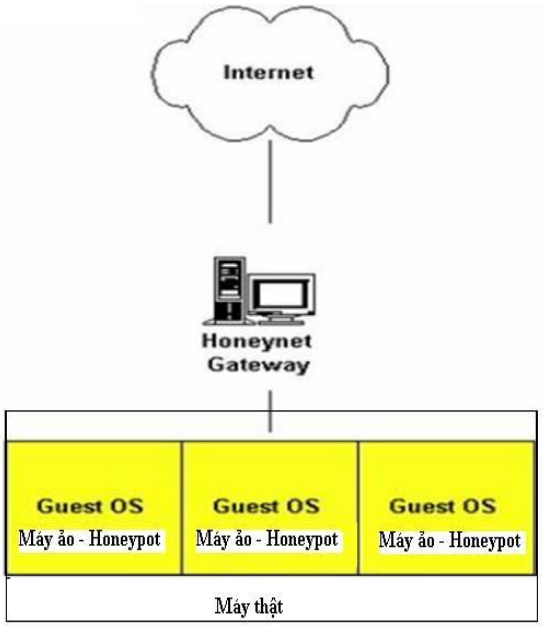
Hình 2.1.2 - Mô hình kiến trúc Honeynet thế hệ II, III

2.1.3.Hệ thống Honeynet ảo

Việc triển khai xây dựng hệ thống Honeynet yêu cầu 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 cơ bản vẫn giống như Honeynet II và III, vẫn sử dụng một Honeywall Gateway nhưng chỉ khác ở chỗ Honeynet 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 việc quản lý.

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



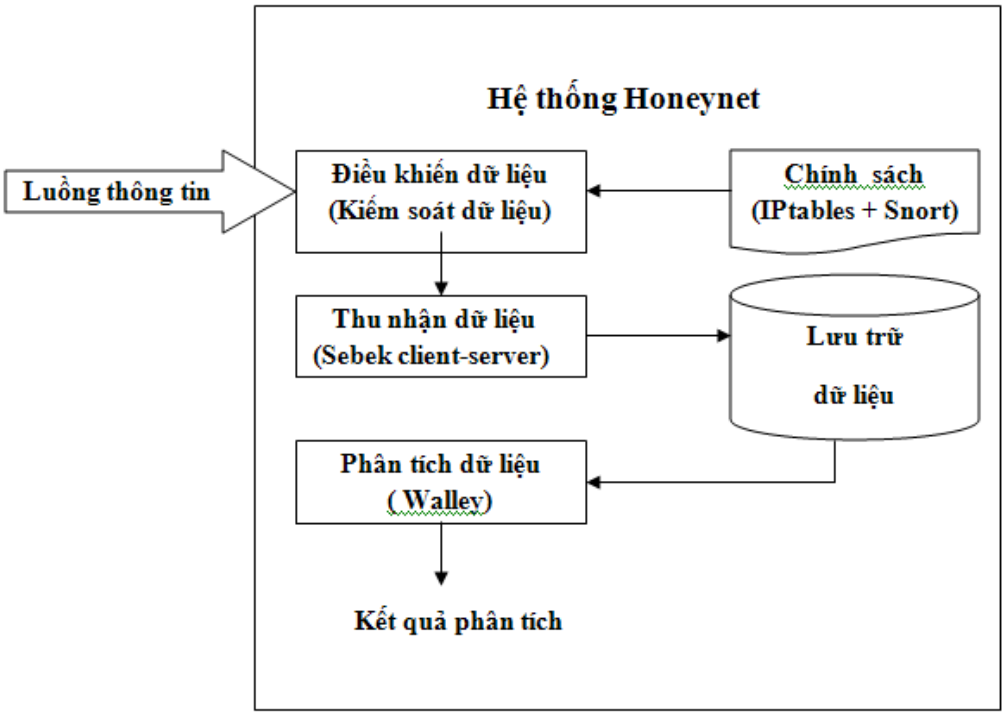
Hình 2.1.3 - Mô hình Honeynet ảo

Sơ đồ trên gồm hai máy tính vật lý: máy 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 đề tài về mô hình kiến trúc logic của Honeynet dưới đây sẽ cho chúng ta hiểu rõ về phương thức hoạt động, làm việc của hệ thống Honeynet.

2.2.MÔ HÌNH KIẾN TRÚC LOGGIC

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 logic chung như sau:



Hình 2.2 - Mô hình kiến trúc logic của Honeynet

Trong một hệ thống Honeynet bao gồm ba module chính :

Module điều khiển dữ liệu (kiểm soát dữ liệu): nhiệm vụ của module này là kiểm soát dữ liệu ra vào 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. Walleye, Hflow trong Honeywall sẽ thực hiện nhiệm vụ này.

Để giúp hiểu rõ hơn về hoạt động của hệ thống Honeynet, đề tài sẽ tiếp tục phân tích kỹ hơn về ba module này.

2.2.1.Module điều khiển dữ liệu (kiểm soát dữ liệu)

2.2.1.1.Vai trò - nhiệm vụ

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

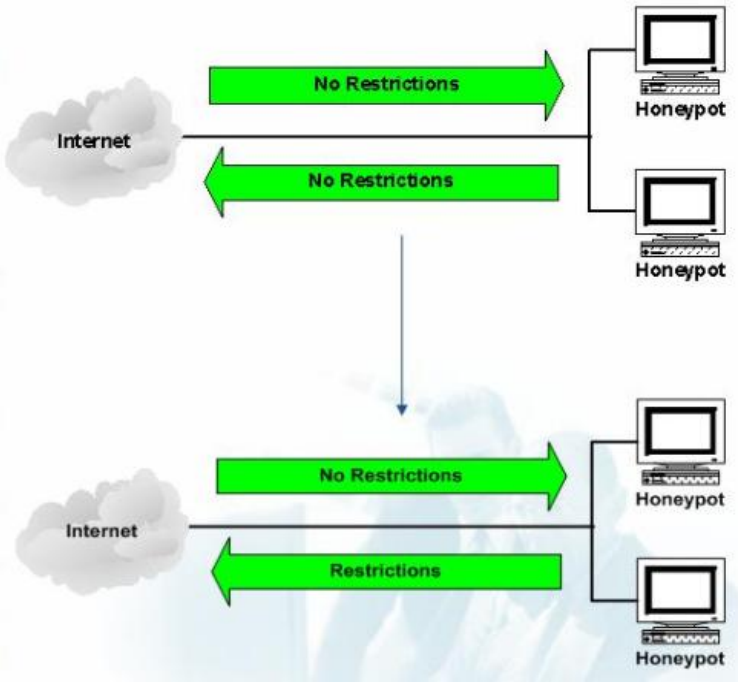
- Kẻ tấn công có thể chiếm được quyền kiểm soát Honeynet và thực hiện các hành vi phá hoại hệ thống.
- Honeynet bị kẻ tấn công lợi dụng biến thành công cụ để tấn công vào các hệ thống mạng bên ngoài khác …

Từ đó đặt ra yêu cầu là 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 kiềm 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. Kẻ tấn công phải bị kiểm soát để nó không thể thực hiện điều đó. Yêu cầu đặt ra là module 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.



Hình 2.2.1.1 - Mô hình kiểm soát dữ liệu trước và sau khi có module

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

2.2.1.2.Cơ chế

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

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.

Tóm lại, số lượng kết nối cho phép đi 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.

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 Intrusion Prevention System), cụ thể ở đây là Snort.

Mục đích của NIPS là để phát hiện và ngăn chặn những tấn công đã biết được đinh nghĩa trong tập luật (Rule) của NIPS. NIPS thực hiện công việc này bằng phương pháp kiểm 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 (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 loại bỏ gói tin hoặc thay thế, sửa chữa gói tin.

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 System), cụ thể ở đây là hệ thống Snort.

2.2.1.3.Kiểm soát dữ liệu trong Honeynet II

Honeynet được phát triển qua ba thế hệ là I, II và III. Về mặt bản chất thì cả ba thế hệ này đều có cách thức kiểm soát dữ liệu gần giống nhau. Tuy nhiên thế hệ II,III có những điểm cải tiến nâng cao hơn sao với thế hệ 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.

a. Tường lửa Iptables

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 gói tin (package) trong Iptables

Iptables sẽ kiểm tra tất cả các gói tin đi qua nó, quá trình kiểm tra này được thực hiện một cách tuần tự.

Iptables cơ bản gồm ba bảng Filter, Mangle, Nat và các chain trong mỗi bảng. Với chúng người quản trị có thể tạo ra các rules cho phép các gói tin vào ra hệ thống (được bảo vệ bằng iptables) tuỳ theo ý muốn của mình. Chức năng cụ thể của chúng như sau:

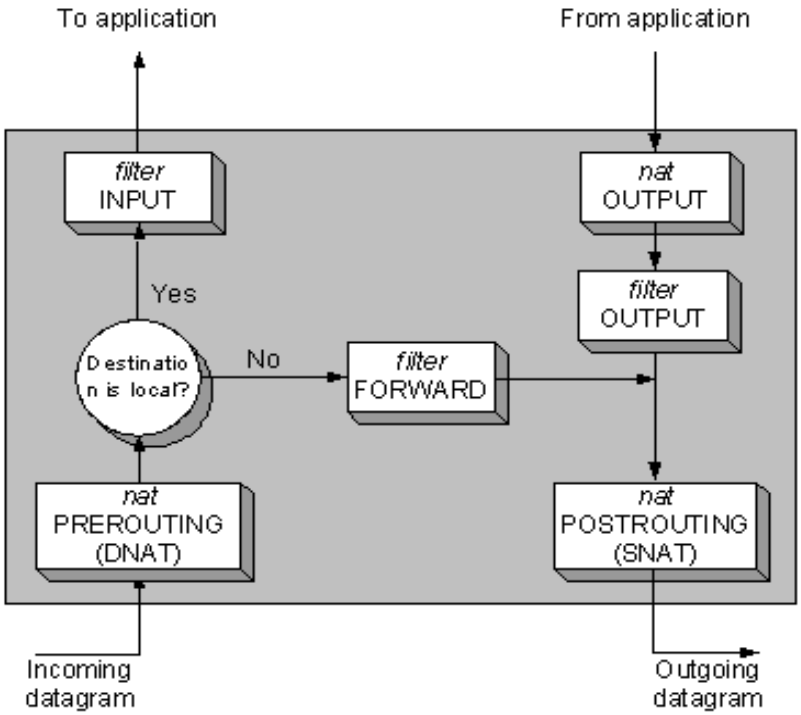
Mangle: dùng để chỉnh sửa QOS(quality of service) bit trong phần TCP Header của gói tin

Filter: đúng như tên gọi, nó dùng để lọc các gói tin gồm các build-in chain

- Forward chain: lọc những gói tin đi qua hệ thống (đi vào một hệ thống khác).
- Input chain: lọc những gói tin đi vào hệ thống.
- Output chain: lọc những gói tin đi ra từ hệ thống.

Nat: sửa địa chỉ gói tin gồm các build-in chain

- Pre-routing: sửa địa chỉ đích của gói tin trước khi nó được routing bởi bảng routing của hệ thống (destination NAT hay DNAT).
- Post-routing: ngược lại với Pre-routing, nó sửa địa chỉ nguồn của gói tin sau khi gói tin đã được routing bởi hệ thống (source NAT hay SNAT).

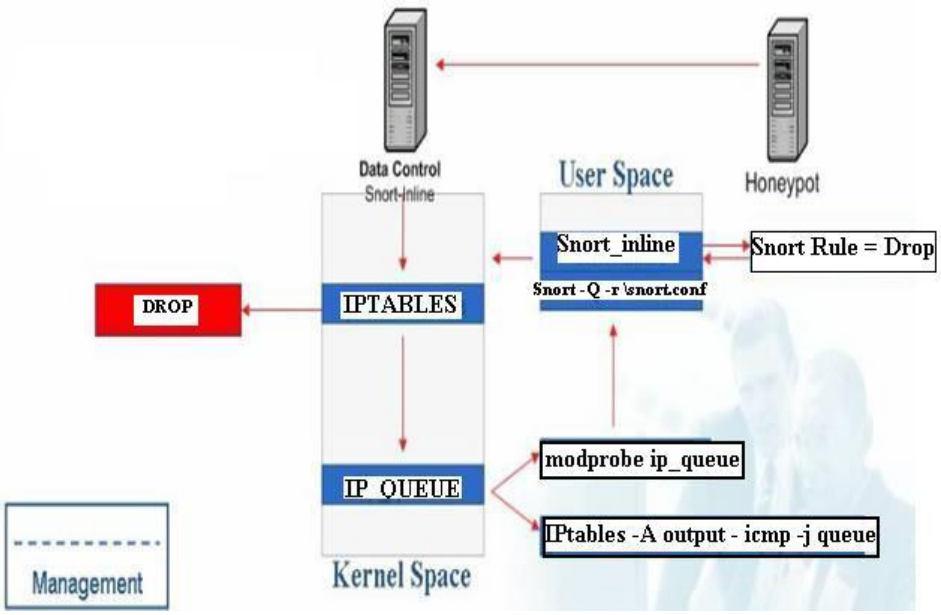


Hình 2.2.1.3a – Quá trình lọc và xử lý gói tin trong Iptables

b. IDS Snort

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. Kết hợp cùng với Iptables, Snort thực hiện chặn các cuộc tấn công 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.

Hiện tại, trong hệ thống Honeynet, Snort đã được nâng cấp lên một mức cao hơn là Snort_inline. Snort_inline cải tiến hơn so với Snort ở chỗ: nó chỉ kiểm tra nội dung các gói tin sau khi đã đi qua Iptables (Snort lại lắng nghe tất cả các gói tin trên interface được chỉ định). Vì nó lắng nghe tất cả các gói tin đến nên việc xử lý khá chậm. Tường lửa Iptables cho các gói tin đi qua vào hàng đợi (Queue) và Snort_inline mở từng gói tin để kiểm tra.



Hình 2.2.1.3b – Cơ chế làm việc của Snort

- Cơ chế loại bỏ gói tin: 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 là nó 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 kiểm 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 tới Iptables chặn gói tin này lại không cho ra bên ngoài. Biện pháp này thực hiện đơn giản nhưng kém linh hoạt dễ làm cho hacker nghi ngờ.
- Cơ chế thay thế gói tin: cơ chế này cơ bản cũng giống như cơ chế 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 đổi 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 tấn công 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 trong hệ thống 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.

2.2.2.Module thu nhận dữ liệu

2.2.2.1.Vai trò - nhiệm vụ

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.

Module thu nhận dữ liệu thực hiện giám sát và ghi lại các hành vi của kẻ tấn công bên trong Honeynet. 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 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 luồng 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 module thu nhận dữ liệu:

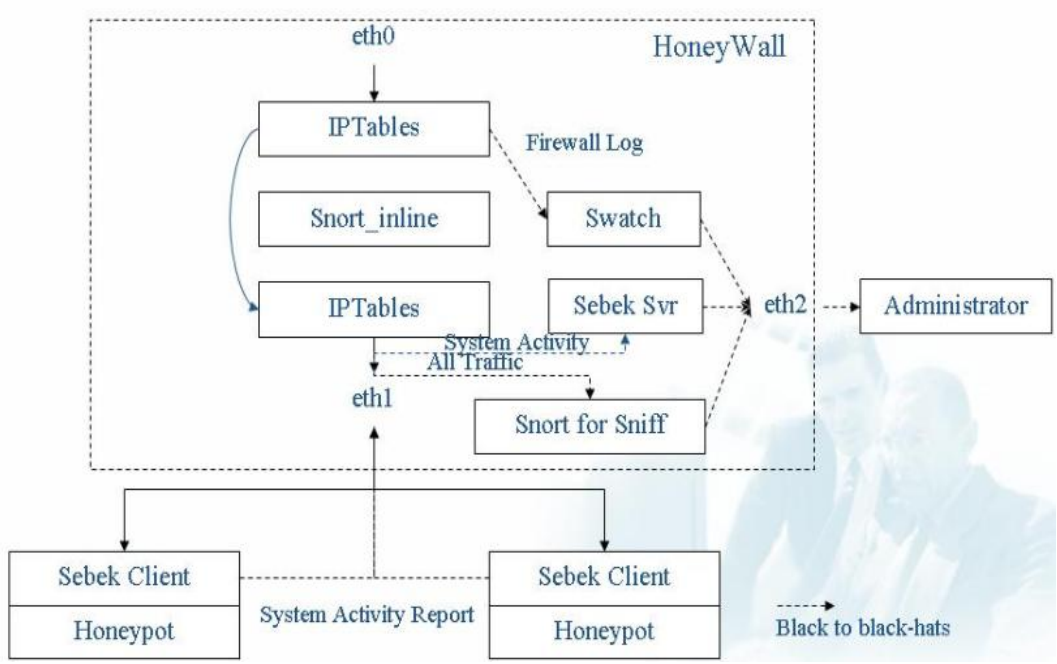
- Thu nhận càng nhiều dữ liệu càng tốt
- Đảm bảo tính chính xác, sẵn sàng
- Che dấu đối với hacker

2.2.2.2.Cơ chế

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 của các Honeypot trong hệ thống (nhờ vào Sebek client - server).



Hình 2.2.2.2a - Sơ đồ thu nhận dữ liệu

a. Thu nhận từ tường lửa (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 là 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 thời gian của gói tin).

b. Thu nhận dữ liệu từ luồng mạng (nhờ công cụ Snort)

Trong Honeynet tầng thu nhận dữ liệu này được thực hiện bởi Snort được cấu hình ở chế độ thu nhận tất cả các gói tin trong mạng. 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 trên đường truyền. Snort thực hiện lắng nghe trên toàn mạng để bắt giữ và kiểm tra nội dung của tất cả các gói tin qua nó.

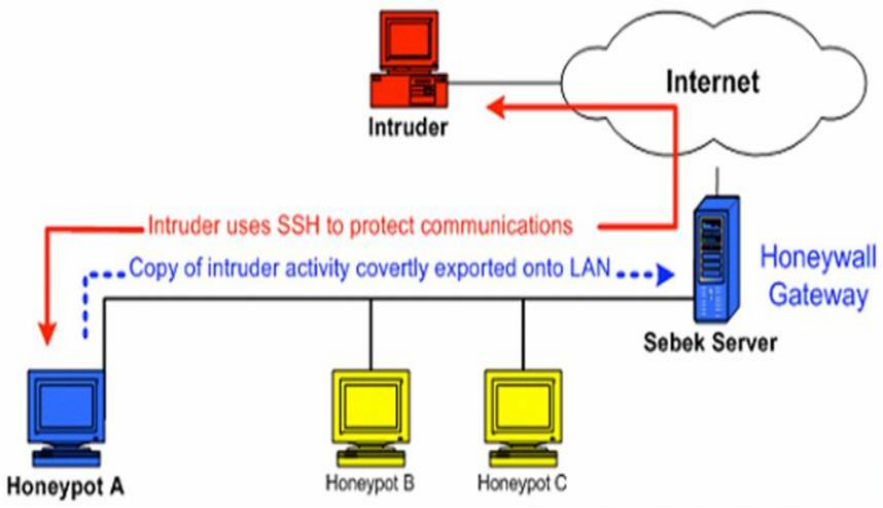
c. Thu nhận dữ liệu hoạt động trên các Honeypot

Nhiệm vụ của module 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 ba 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 dụng 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 gửi các thông tin thu thập được về Sebek server.



Hình 2.2.2.2b - Mô hình hoạt động của Sebek

Tóm lại: module thu nhận dữ liệu thực hiện nhiệm vụ thu nhận dữ liệu nhằm khám phá ra các kỹ thuật 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.

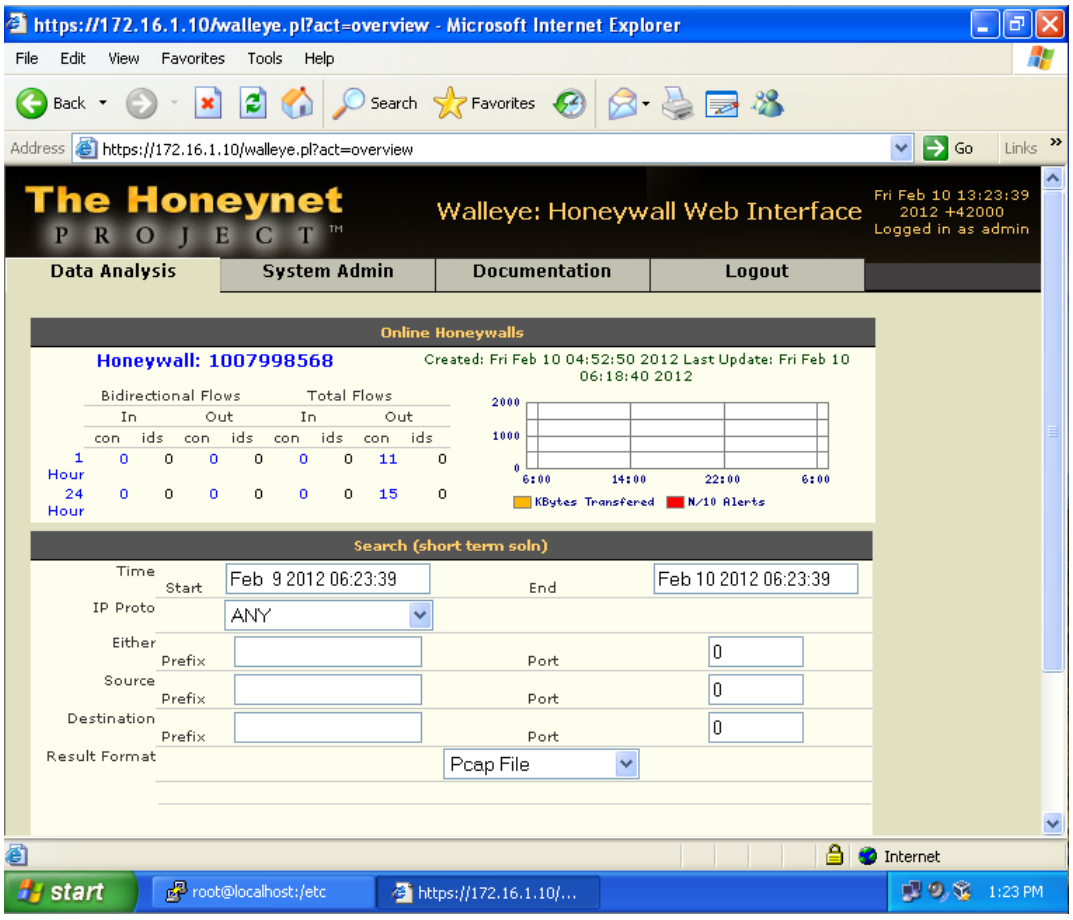
2.2.3.Module phân tích dữ liệu

2.2.3.1.Vai trò - nhiệm vụ

Vai trò của module phân tích dữ liệu trong Honeynet nhằm hỗ trợ người phân tích 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 hiện ra vấn đề trọng tâm cần phân tích (như dữ liệu liên quan đến quá trình tấn công hay hoạt động bất hợp pháp của hacker).

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 tìm 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.

Honeynet 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.



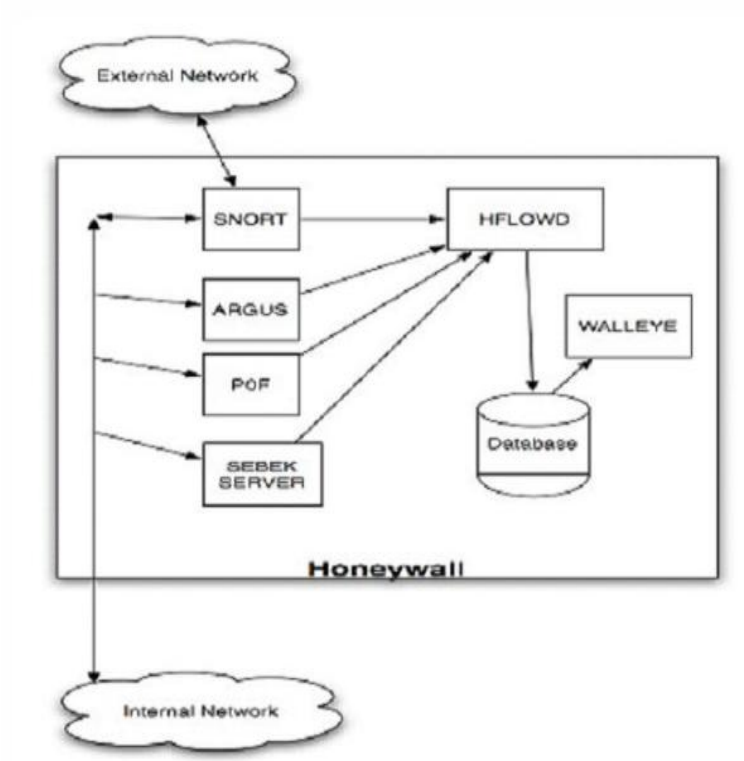
Hình 2.2.3.1 - Giao diện web Walleye

2.2.3.2. Cơ chế

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.



Hình 2.2.3.2 – Sơ đồ kiến trúc Honeywall

Hflow có nhiệm vụ kết hợp dữ liệu từ module thu nhận dữ liệu gửi về, chuẩn hóa dữ liệu và lưu vào cơ sở dữ liệu.

Walleye có nhiệm vụ lấy dữ liệu thu thập được, đã được Hflow chuẩn hóa trong cơ sở 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 của hệ thống, chi tiết các hoạt động trong mạng.

Tóm lại: module này có nhiệm vụ hỗ trợ người phân tích 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.

Kết luận: ở chương này của đề tài 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. Để thực hiện tốt mục đích của đề tài là nghiên cứu Honeynet nhằm mục đích phòng chống và phát hiện tấn công thì bên cạnh việc nghiên

cứu Honeynet, chúng ta cũng cần phải có kiến thức, hiểu biết nhất định về những hướng tấn công phổ biến hiện nay mà hacker sử dụng.

CHƯƠNG III: MỘT SỐ KỸ THUẬT TẤN CÔNG PHỔ BIẾN HIỆN NAY

3.1.TẤN CÔNG XSS (CROSS SITE SCRIPTING)

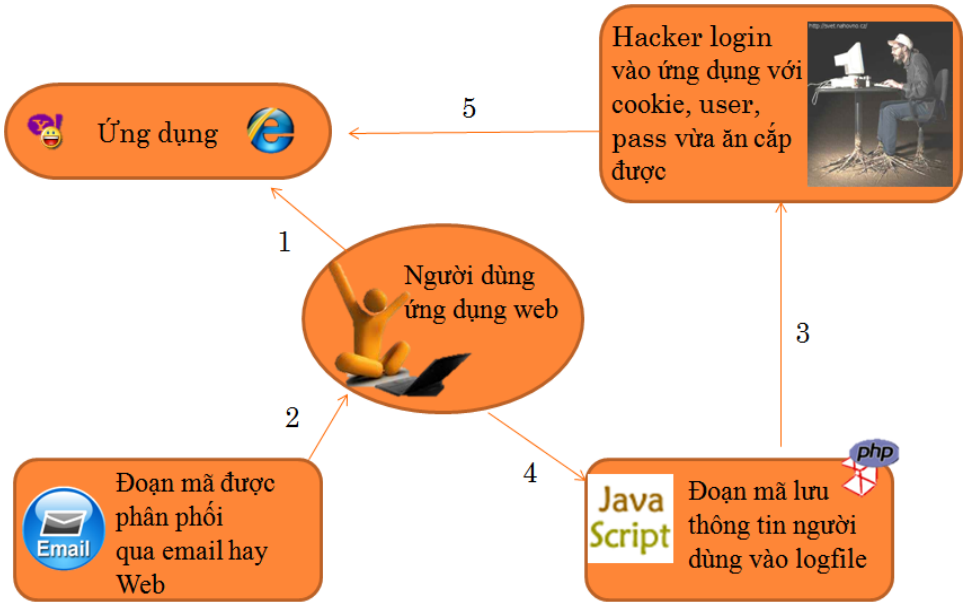
3.1.1.Giới thiệu

Cross-Site Scripting hay còn được gọi tắt là XSS (thay vì gọi tắt là CSS để tránh nhầm lẫn với CSS-Cascading Style Sheet của HTML) là một kĩ thuật tấn công bằng cách chèn vào các website động (ASP, PHP, CGI, JSP ...) những thẻ HTML hay những đoạn mã script nguy hiểm có khả năng đánh cắp hay thiết lập được những thông tin quan trọng như cookies, mật khẩu, usename.... Trong đó, những đoạn mã nguy hiểm được chèn vào hầu hết được viết bằng các Client-Site Script như JavaScript, JScript, DHTML và cũng có thể là cả các thẻ HTML.

3.1.2.Phương pháp tấn công XSS truyền thống

Khác với các lỗi khác là gây hại trực tiếp lên hệ thống chứa web site, còn XSS lại không gây hại đến hệ thống của sever mà đối tượng tấn công chủ yếu của XSS lại là người dùng.

Ứng dụng Web thường lưu trữ thông tin quan trọng ở cookie. Cookie là mẫu thông tin mà ứng dụng lưu trên đĩa cứng của người sử dụng. Nhưng chỉ ứng dụng thiết lập ra cookie thì mới có thể đọc nó. Do đó chỉ khi người dùng đang trong phiên làm việc của ứng dụng thì hacker mới có cơ hội đánh cắp cookie. Công việc đầu tiên của hacker là tìm trang đích để dụ người dùng đăng nhập sau khi đã tìm ra lỗ hổng trên ứng dụng đó.



Hình 3.1.2 – Các bước khai thác XSS theo truyền thống

Tóm tắt các bước thực hiện:

- Bước 1: Hacker biết được người dùng đang sử dụng một ứng dụng Web có lỗ hỏng XSS.
- Bước 2: Người dùng nhận được một liên kết thông qua email hay trên chính trang Web (như trên banner dễ dàng thêm 1 liên kết do chính hacker tạo ra…).
- Bước 3: Chuyển nội dung thông tin (cookie, tên, mật khẩu…) về máy chủ của hacker.
- Bước 4: Hacker tạo một chương trình cgi hoặc một trang web để ghi nhận những thông tin đã đánh cắp vào một tập tin
- Bước 5: Sau khi nhận được thông tin cần thiết, hacker có thể sử dụng để thâm nhập vào tài khoản của người dùng.

3.1.3.Biện pháp phòng chống

Với những dữ liệu, thông tin đăng nhập của người dùng, người thiết kế ứng dụng Web cần phải thực hiện vài bước cơ bản sau:

- Tạo ra danh sách những thẻ HTML được phép sử dụng.
- Xóa bỏ thẻ <script>

- Lọc ra bất kì một đoạn mã JavaScript/Java/VBScript/ActiveX/Flash Related nào.
- Lọc dấu nháy đơn hay kép.
- Lọc kí tự Null (vì khả năng thêm một đoạn mã bất kì sau kí tự Null khiến cho ứng dụng dù đã lọc bỏ thẻ <script> vẫn không nhận ra do ứng dụng nghĩ rằng chuỗi đã kết thúc từ kí tự Null này).
- Xóa những kí tự " > ", " < "
- Vẫn cho phép nhập những kí tự đặc biệt nhưng sẽ được mã hóa theo chuẩn riêng.
- Đối với người dùng, cần cấu hình lại trình duyệt để nhắc nhở người dùng có cho thực thi ngôn ngữ kịch bản trên máy của họ hay không? Tùy vào mức độ tin cậy mà người dùng sẽ quyết định.

Nhận xét: kĩ thuật XSS khá phổ biến và dễ dàng áp dụng, tuy nhiên mức độ thiệt hại chỉ dừng lại ở mức độ tấn công trên máy nạn nhân thông qua những liên kết hay form lừa đảo mà hacker đưa đến cho nạn nhân. Vì thế, ngoài việc ứng dụng kiểm tra tính đúng đắn của dữ liệu trước khi sử dụng thì việc cần nhất là người dùng nên cảnh giác trước một trang Web lạ.

Có thể nói, nhờ vào sự cảnh giác của người dùng thì 90% đã đạt được sự bảo mật trong kĩ thuật này.

3.2.TẤN CÔNG DoS (DENIAL OF SERVICE)

3.2.1.Khái niệm

Tấn công kiểu DoS là kiểu tấn công làm cho các dịch vụ mạng bị tê liệt, không còn khả năng đáp ứng được yêu cầu nữa. Loại tấn công này ảnh hưởng đến nhiều hệ thống, rất dễ thực hiện và lại rất khó bảo vệ hệ thống khỏi kiểu tấn công DoS. Thông thường, kiểu tấn công Dos dựa trên những giao thức.Ví dụ với giao thức ICMP, hacker có thể sử dụng bomb e-mail để gửi hàng ngàn thông điệp email với mục đích tiêu thụ băng thông để làm hao hụt tài nguyên hệ thống trên mail server hoặc có thể dùng phần mềm gửi hàng loạt yêu cầu đến máy chủ khiến cho máy chủ không thể đáp ứng những yêu

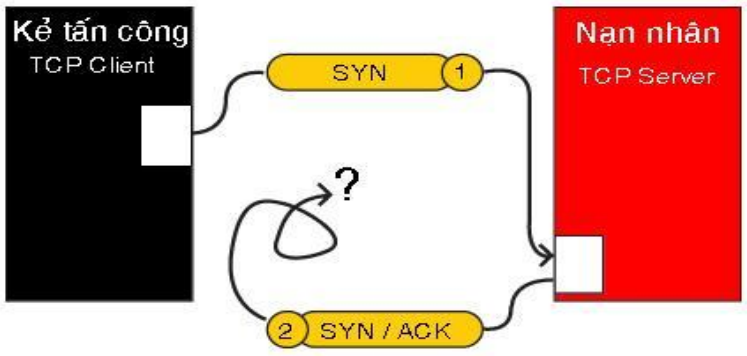
cầu chính đáng khác.

3.2.2.Những nguy cơ bị tấn công bằng DoS

- Tấn công trên Swap Space: hầu hết các hệ thống đều có vài trăm MB không gian chuyển đổi (swap space) để phục vụ cho những yêu cầu từ máy khách. Swap space thường dùng cho các tiến trình con có thời gian ngắn nên Dos được thực hiện dựa trên phương thức làm tràn đầy swap space.
- Tấn công trên Bandwidth: phần băng thông dành cho mỗi hệ thống là giới hạn, vì thế nếu hacker cùng lúc gửi nhiều yêu cầu đến hệ thống thì phần băng thông không đủ đáp ứng cho một khối lượng dữ liệu lớn đó và dẫn đến hệ thống bị phá vỡ.
- Tấn công vào Ram: tấn công Dos chiếm một khoảng lớn của Ram cũng có thể gây ra các vấn đề phá hủy hệ thống. Kiểu tấn công BufferOverflow là một ví dụ cho cách phá hủy này.
- Tấn công vào Disks: một kiểu tấn công cổ điển là làm đầy đĩa cứng. Đĩa cứng có thể bị tràn và không thể sử dụng được nữa.

3.2.3.Một số dạng tấn công thường gặp

3.2.3.1.Lợi dụng TCP thực hiện phương pháp SYN flood truyền thống



Hình 3.2.3.1 - Tấn công DoS truyền thống

Bất cứ một gói tin SYN nào, máy chủ cũng phải để một phần tài nguyên của hệ thống như bộ nhớ đệm để nhận và truyền dữ liệu cho đường truyền đó. Tuy nhiên, tài nguyên của hệ thống là có hạn và hacker sẽ tìm mọi cách để hệ thống vượt qua giới

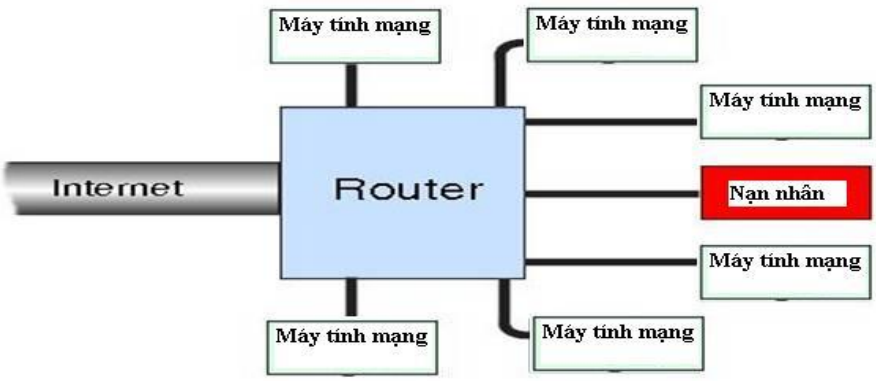
hạn đó. Nếu máy chủ sau khi gửi trả một gói tin SYN/ACK để thông báo chấp nhận kết nối cho máy yêu cầu nhưng nếu địa chỉ IP của máy yêu cầu này là giả mạo thì gói tin không thể đến được đích nên máy chủ vẫn phải dành tài nguyên cho yêu cầu đó. Sau một thời gian không nhận được phản hồi từ máy khách, máy chủ lại tiếp tục gửi một gói tin SYN/ACK để xác nhận lần nữa và cứ như vậy, kết nối vẫn tiếp tục mở.

Nếu như hacker gửi nhiều gói tin SYN đến máy chủ đến khi máy chủ không thể tiếp nhận thêm một kết nối nào nữa thì lúc này hệ thống đã bị phá vỡ.

Tóm lại: chỉ với một đường truyền băng thông nhỏ, hacker đã có thể phá vỡ một hệ thống. Thêm vào đó, địa chỉ IP của hacker có thể được sửa đổi nên việc xác định thủ phạm là một vấn đề hết sức khó khăn.

3.2.3.2.Tấn công vào băng thông

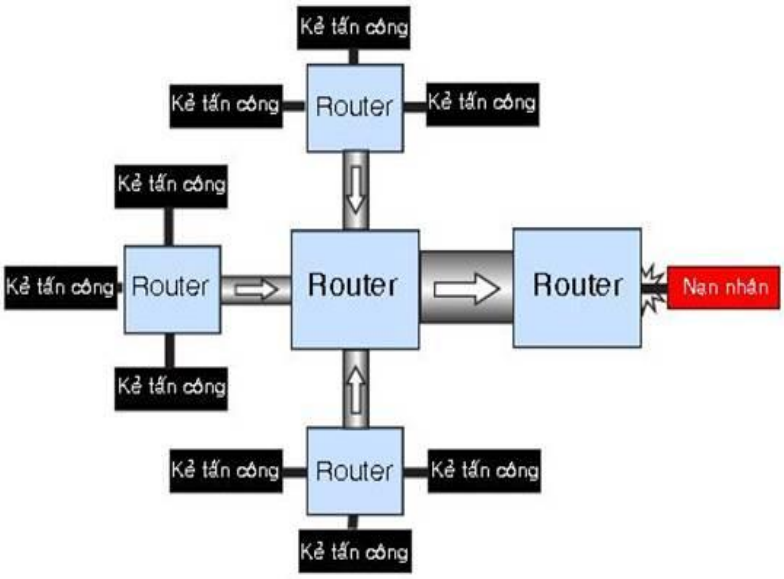
a. DoS



Hình 3.2.3.2a – Kiểu tấn công DoS vào băng thông

Kiểu tấn công này được sử dụng khi đường truyền mạng của hacker là quá thấp so với đường truyền của máy đích. Không giống như kiểu tấn công DoS truyền thống, kiểu tấn công vào băng thông lớn hơn sẽ lợi dụng những gói tin từ những hệ thống khác nhau cùng một lúc tiến đến hệ thống đích khiến cho đường truyền của hệ thống đích không còn khả năng đáp ứng, máy chủ không còn khả năng nhận một gói tin nào nữa.

b. DDoS

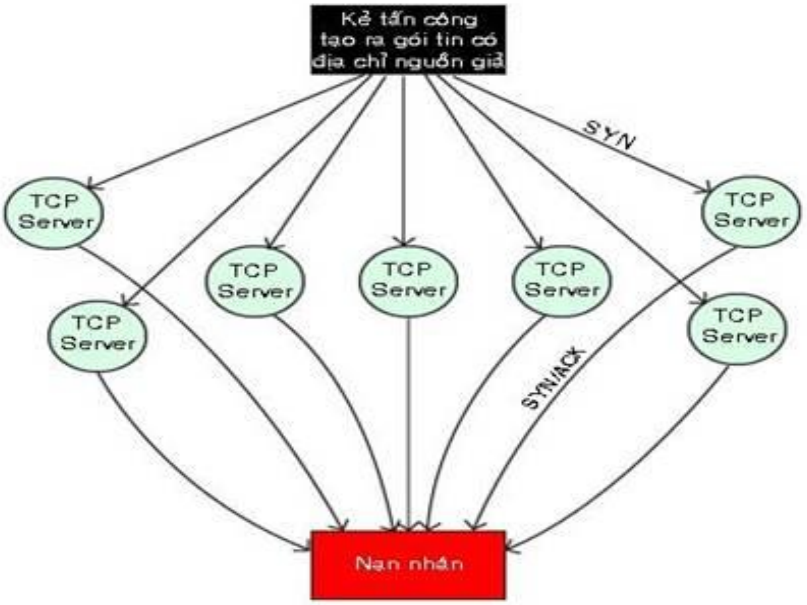


Hình 3.2.3.2b – Tấn công DDoS

Theo hình trên, tất cả các gói tin đi vào một mạng máy tính qua một "Big Pipe" (ống dẫn lớn), sau đó được router chia ra những "Small Pipe" (ống dẫn nhỏ) cho nhiều máy tính con tùy theo địa chỉ IP của gói tin. Nhưng nếu toàn bộ "Big Pipe" bị làm ngập bằng những gói tin chỉ hướng đến một máy nhất định trong mạng máy tính con này, router đành phải chấp nhận loại bỏ phần lớn các packet để chỉ còn lại số lượng vừa đủ đi qua "Small Pipe" của máy tính đó. Kiểu tấn công này sẽ loại máy đích ra khỏi Internet. Đây là phương pháp tấn công kiểu từ chối dịch vụ nhưng không gọi là DoS mà gọi là DDoS (kiểu từ chối dịch vụ phân tán), nghĩa là cùng một lúc nhiều máy sẽ được phát động để gửi gói tin đến máy đích (mặc dù đường truyền của mỗi máy không cao nhưng nhiều đường truyền lại hợp thành một ống dẫn " Big Pipe"), làm cho máy đích không còn khả năng tiếp nhận gói tin và bị loại khỏi mạng Internet

c. DRDoS

DRDoS (Distributed Reflection Denial of Service) - Thế hệ tiếp theo của DdoS



Hình 3.2.3.2c – Tấn công kiểu DRDoS

Bằng cách giả địa chỉ IP của máy đích, hacker sẽ cùng lúc gửi nhiều gói tin đến các hệ thống máy trên mạng, các hệ thống này khi nhận gói tin SYN giả này, chấp nhận kết nối và gửi trả một gói tin SYN/ACK để thông báo. Vì địa chỉ IP của gói tin SYN bị hacker sửa đổi thành địa chỉ IP máy đích nên những gói tin SYN/ACK sẽ được gửi về cho máy đích. Cùng một lúc nhận được nhiều g ói tin, đường truyền của máy đích không đủ khả năng đáp ứng, hệ thống máy đích từ chối nhận bất kì gói tin nào và lúc này hệ thống máy đích đã bị sụp đổ.

3.2.3.Biện pháp phòng chống

Kiểu tấn công từ chối dịch vụ là kiểu tấn công gây nhiều khó khăn trong vấn đề bảo vệ cũng như điều tra tìm ra thủ phạm nhất, bởi vì hầu hết hacker đã thay đổi địa chỉ IP của máy nên rất khó xác định ai là thủ phạm.

Cách phòng chống khả năng khuyếch đại đường truyền:

- Huỷ khả năng broadcast tại router biên

- Tăng kích thước hàng đợi kết nối → kết quả: có thể phòng tránh khả năng tràn hàng đợi qua nhiều kết nối nhưng cách này sử dụng nhiều tài nguyên
- Giảm thời gian thiết lập kết nối
- Dùng những phần mềm phát hiện và phá hủy kiểu tấn công DoS

Nhận xét: kiểu tấn công từ chối dịch vụ tuy chỉ khiến cho hệ thống bị phá vỡ trong vài phút nhưng hậu quả thì khá to lớn (ảnh hưởng trên phạm vi tiền và uy tín). Đây là kĩ thuật thường được hacker sử dụng trong trường hợp không thể chiếm quyền quản trị trên hệ thống hoặc muốn phá hủy uy tín của cơ quan đó. Thêm vào đó việc giả mạo địa chỉ khiến cho hacker càng dễ dàng thực hiện viêc tấn công mà không sợ bị phát hiện.

3.3.TẤN CÔNG SQL INJECTION

3.3.1.Khái niệm

SQL Injection là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng trong việc kiểm tra dữ liệu nhập trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu để "tiêm vào" và thi hành các câu lệnh SQL bất hợp pháp (không được người phát triển ứng dụng lường trước). Hậu quả của nó rất tai hại cho phép những kẻ tấn công chiếm quyền kiểm soát web. Lỗi này thường xảy ra trên các ứng dụng web có dữ liệu được quản lí bằng các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, Oracle, DB2, Sysbase.

Về bản chất, đây là lỗi tồn tại trong các hệ quản trị cơ sở dữ liệu (SQL server, MySQL, Oracle ...). Tuy nhiên, chúng ta có thể khắc phục lỗi này bằng cách kiểm tra dữ liệu được nhập vào trên ứng dụng Web.

3.3.2.Các dạng tấn công bằng SQL Injection

Có bốn dạng thông thường bao gồm: vượt qua kiểm tra lúc đăng nhập (authorization bypass), sử dụng câu lện SELECT, sử dụng câu lệnh INSERT, sử dụng các stored-procedures.

3.3.2.1.Dang tấn công vượt qua kiểm tra đăng nhập

Với dạng tấn công này, tin tặc có thể dễ dàng vượt qua các trang đăng nhập nhờ vào lỗi khi dùng các câu lệnh SQL thao tác trên cơ sở dữ liệu của ứng dụng web.

Xét một ví dụ điển hình, thông thường để cho phép người dùng truy cập vào các trang web được bảo mật, hệ thống thường xây dựng trang đăng nhập để yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu. Sau khi người dùng nhập thông tin vào, hệ thống sẽ kiểm tra tên đăng nhập và mật khẩu có hợp lệ hay không để quyết định cho phép hay từ chối thực hiện tiếp.

Trong trường hợp này, người ta có thể dùng hai trang, một trang HTML để hiển thị form nhập liệu và một trang ASP dùng để xử lí thông tin nhập từ phía người dùng.

Ví dụ:

login.htm

```
<form action="ExecLogin.asp" method="post">
Username: <input type="text" name="fUSRNAME"><br>
Password: <input type="password" name="fPASSWORD"><br>
<input type="submit">
</form>
```

execlogin.asp

```
<%
Dim vUsrName, vPassword, objRS, strSQL
vUsrName = Request.Form("fUSRNAME")
vPassword = Request.Form("fPASSWORD")
strSQL = "SELECT * FROM T_USERS " & _
"WHERE USR_NAME=' " & vUsrName & _
" ' and USR_PASSWORD=' " & vPassword & " ' "
Set objRS = Server.CreateObject("ADODB.Recordset")
objRS.Open strSQL, "DSN=..."
If (objRS.EOF) Then
Response.Write "Invalid login."
Else
Response.Write "You are logged in as " & objRS("USR_NAME")
End If
Set objRS = Nothing
%>
```

Thoạt nhìn, đoạn mã trong trang **execlogin.asp** dường như không chứa bất cứ một lỗ hổng về an toàn nào. Người dùng không thể đăng nhập mà không có tên đăng nhập và mật khẩu hợp lệ. Tuy nhiên, đoạn mã này thực sự không an toàn và là tiền đề cho

một lỗi SQL Injection. Đặc biệt, chỗ sơ hở nằm ở chỗ dữ liệu nhập vào từ người dùng được dùng để xây dựng trực tiếp câu lệnh SQL. Chính điều này cho phép những kẻ tấn công có thể điều khiển câu truy vấn sẽ được thực hiện. Ví dụ, nếu người dùng nhập chuỗi sau vào trong cả 2 ô nhập liệu username/password của trang login.htm là:
' OR ' ' = ' '. Lúc này, câu truy vấn sẽ được gọi thực hiện là:
*SELECT * FROM T_USERS WHERE USR_NAME =" OR "=" and USR_PASSWORD= " OR "="*

Câu truy vấn này là hợp lệ và sẽ trả về tất cả các bản ghi của T_USERS và đoạn mã tiếp theo xử lí người dùng đăng nhập bất hợp pháp này như là người dùng đăng nhập hợp lệ.

3.3.2.2.Dạng tấn công sử dụng câu lệnh SELECT

Dạng tấn công này phức tạp hơn. Để thực hiện được kiểu tấn công này, kẻ tấn công phải có khả năng hiểu và lợi dụng các sơ hở trong các thông báo lỗi từ hệ thống.

Xét một ví dụ rất thường gặp trong các website về tin tức. Thông thường, sẽ có một trang nhận ID của tin cần hiển thị rồi sau đó truy vấn nội dung của tin có ID này. Ví dụ: http://www.myhost.com/shownews.asp?ID=123. Mã nguồn cho chức năng này thường được viết khá đơn giản theo dạng

```
<%
Dim vNewsID, objRS, strSQL
vNewsID = Request("ID")
strSQL = "SELECT * FROM T_NEWS WHERE NEWS_ID =" & vNewsID
Set objRS = Server.CreateObject("ADODB.Recordset")
objRS.Open strSQL, "DSN=..."
Set objRS = Nothing
%>
```

Trong các tình huống thông thường, đoạn mã này hiển thị nội dung của tin có ID trùng với ID đã chỉ định và hầu như không thấy có lỗi. Tuy nhiên, giống như ví dụ đăng nhập ở trước, đoạn mã này để lộ sơ hở cho một lỗi SQL Injection khác. Kẻ tấn công có thể thay thế một ID hợp lệ bằng cách gán ID cho một giá trị khác và từ đó,

khởi đầu cho một cuộc tấn công bất hợp pháp, ví dụ như: 0 OR 1=1 (nghĩa là, http://www.myhost.com/shownews.asp?ID=0 or 1=1).

Câu truy vấn SQL lúc này sẽ trả về tất cả các article từ bảng dữ liệu vì nó sẽ thực hiện câu lệnh:

*SELECT * FROM T_NEWS WHERE NEWS_ID=0 or 1=1*

Một trường hợp khác, ví dụ như trang tìm kiếm. Trang này cho phép người dùng nhập vào các thông tin tìm kiếm như Họ, Tên, … Đoạn mã thường gặp là:

```
<%
Dim vAuthorName, objRS, strSQL
vAuthorName = Request("fAUTHOR_NAME")
strSQL = "SELECT * FROM T_AUTHORS WHERE AUTHOR_NAME =' " & _
vAuthorName & " ' "
Set objRS = Server.CreateObject("ADODB.Recordset")
objRS.Open strSQL, "DSN=..."
…
Set objRS = Nothing
%>
```

Tương tự như trên, tin tặc có thể lợi dụng sơ hở trong câu truy vấn SQL để nhập vào trường tên tác giả bằng chuỗi giá trị:

' UNION SELECT ALL SELECT OtherField FROM OtherTable WHERE ' '=' (*)

Lúc này, ngoài câu truy vấn đầu không thành công, chương trình sẽ thực hiện thêm lệnh tiếp theo sau từ khóa UNION nữa.

Tất nhiên các ví dụ nói trên, dường như không có gì nguy hiểm, nhưng hãy thử tưởng tượng kẻ tấn công có thể xóa toàn bộ cơ sở dữ liệu bằng cách chèn vào các lệnh nguy hiểm như lệnh DROP TABLE. Ví dụ như: **' DROP TABLE T_AUTHORS --**

Chắc các bạn sẽ thắc mắc là làm sao biết được ứng dụng web bị lỗi dạng này được. Rất đơn giản, hãy nhập vào chuỗi (*) như trên, nếu hệ thống báo lỗi về cú pháp dạng: Invalid object name “OtherTable”; ta có thể biết chắc là hệ thống đã thực hiện câu

SELECT sau từ khóa UNION, vì như vậy mới có thể trả về lỗi mà ta đã cố tình tạo ra trong câu lệnh SELECT.

Cũng sẽ có thắc mắc là làm thế nào có thể biết được tên của các bảng dữ liệu mà thực hiện các thao tác phá hoại khi ứng dụng web bị lỗi SQL injection. Cũng rất đơn giản, bởi vì trong SQL Server, có hai đối tượng là sysobjects và syscolumns cho phép liệt kê tất cả các tên bảng và cột có trong hệ thống. Ta chỉ cần chỉnh lại câu lệnh SELECT, ví dụ như:

' UNION SELECT name FROM sysobjects WHERE xtype = 'U' là có thể liệt kê được tên tất cả các bảng dữ liệu.

3.3.2.3.Dang tấn công sử dụng câu lệnh INSERT

Thông thường các ứng dụng web cho phép người dùng đăng kí một tài khoản để tham gia. Chức năng không thể thiếu là sau khi đăng kí thành công, người dùng có thể xem và hiệu chỉnh thông tin của mình. SQL injection có thể được dùng khi hệ thống không kiểm tra tính hợp lệ của thông tin nhập vào.

Ví dụ, một câu lệnh INSERT có thể có cú pháp dạng: *INSERT INTO TableName VALUES('Value One', 'Value Two', 'Value Three').* Nếu đoạn mã xây dựng câu lệnh SQL có dạng :

```
<%
strSQL = "INSERT INTO TableName VALUES(' " & strValueOne & " ', ' " _
& strValueTwo & " ', ' " & strValueThree & " ') "
Set objRS = Server.CreateObject("ADODB.Recordset")
objRS.Open strSQL, "DSN=..."
...
Set objRS = Nothing
%>
```

Thì chắc chắn sẽ bị lỗi SQL injection, bởi vì nếu ta nhập vào trường thứ nhất ví dụ như: *' + (SELECT TOP 1 FieldName FROM TableName) + '*. Lúc này câu truy vấn sẽ là: *INSERT INTO TableName VALUES(' ' + (SELECT TOP 1 FieldName FROM TableName) + ' ', 'abc', 'def').* Khi đó, lúc thực hiện lệnh xem thông tin, xem như bạn

đã yêu cầu thực hiện thêm một lệnh nữa đó là: *SELECT TOP 1 FieldName FROM TableName*

3.3.2.4.Dạng tấn công sử dụng stored-procedures

Việc tấn công bằng stored-procedures sẽ gây tác hại rất lớn nếu ứng dụng được thực thi với quyền quản trị hệ thống 'sa'. Ví dụ, nếu ta thay đoạn mã tiêm vào dạng: *' ; EXEC xp_cmdshell 'cmd.exe dir C: '***.** Lúc này hệ thống sẽ thực hiện lệnh liệt kê thư mục trên ổ đĩa C:\ cài đặt server. Việc phá hoại kiểu nào tuỳ thuộc vào câu lệnh đằng sau cmd.exe.

3.3.3.Cách phòng tránh

Như vậy, có thể thấy lỗi SQL injection khai thác những bất cẩn của các lập trình viên phát triển ứng dụng web khi xử lí các dữ liệu nhập vào để xây dựng câu lệnh SQL. Tác hại từ lỗi SQL injection tùy thuộc vào môi trường và cách cấu hình hệ thống. Nếu ứng dụng sử dụng quyền dbo (quyền của người sở hữu cơ sở dữ liệu - owner) khi thao tác dữ liệu, nó có thể xóa toàn bộ các bảng dữ liệu, tạo các bảng dữ liệu mới, … Nếu ứng dụng sử dụng quyền sa (quyền quản trị hệ thống), nó có thể điều khiển toàn bộ hệ quản trị cơ sở dữ liệu và với quyền hạn rộng lớn như vậy nó có thể tạo ra các tài khoản người dùng bất hợp pháp để điều khiển hệ thống của bạn. Để phòng tránh, ta có thể thực hiện ở hai mức:

3.3.3.1.Kiểm soát chặt chẽ dữ liệu đầu vào

Để phòng tránh các nguy cơ có thể xảy ra, hãy bảo vệ các câu lệnh SQL bằng cách kiểm soát chặt chẽ tất cả các dữ liệu nhập nhận được từ đối tượng Request (Request, Request.QueryString, Request.Form, Request.Cookies, and Request.ServerVariables). Ví dụ, có thể giới hạn chiều dài của chuỗi nhập liệu hoặc xây dựng hàm EscapeQuotes để thay thế các dấu nháy đơn bằng 2 dấu nháy đơn như:

```
<%
Function EscapeQuotes(sInput)
sInput = replace(sInput, " ' ", " ' ' ")
EscapeQuotes = sInput
End Function
```

%>

Trong trường hợp dữ liệu nhập vào là số, lỗi xuất phát từ việc thay thế một giá trị được tiên đoán là dữ liệu số bằng chuỗi chứa câu lệnh SQL bất hợp pháp. Để tránh điều này, đơn giản là kiểm tra dữ liệu có đúng kiểu hay không bằng hàm IsNumeric().

3.3.3.2.Thiết lập cấu hình an toàn cho hệ quản trị cơ sở dữ liệu

Cần có cơ chế kiểm soát chặt chẽ và giới hạn quyền xử lí dữ liệu đến tài khoản người dùng mà ứng dụng web đang sử dụng. Các ứng dụng thông thường nên tránh dùng đến các quyền như dbo hay sa. Quyền càng bị hạn chế, thiệt hại càng ít.

Ngoài ra để tránh các nguy cơ từ SQL Injection attack, nên chú ý loại bỏ bất kì thông tin kĩ thuật nào chứa trong thông điệp chuyển xuống cho người dùng khi ứng dụng có lỗi. Các thông báo lỗi thông thường tiết lộ các chi tiết kĩ thuật có thể cho phép kẻ tấn công biết được điểm yếu của hệ thống.

Kết luận: do thời gian thực hiện đề tài có hạn, vì vậy chương này chỉ trình bày một số kỹ thuật tấn công được xem là hay gặp và có mức độ nguy hiểm cao. Trong các kỹ thuật tấn công trình bày ở trên thì tấn công SQL-Injection là thường gặp hơn cả. Do đó ở phần kiểm tra hệ thống Honeynet thông qua việc khai thác lỗ hổng được dựa trên kịch bản tấn công SQL-Injection.

CHƯƠNG IV: TRIỂN KHAI - CÀI ĐẶT - KIỂM TRA HỆ THỐNG

4.1.BẢO MẬT VÀ TỐI ƯU HÓA HỆ THỐNG LINUX

Trước khi bước vào việc triển khai hệ thống Honeynet, công việc cũng không kém phần quan trọng là việc bảo mật và tối ưu hóa hệ thống Linux nhằm tránh tình trạng bị hacker lợi dụng lỗ hổng của hệ thống để khai thác thông tin, đánh cắp dữ liệu. Sau đây sẽ là một số bước cần thiết nhằm giữ an toàn cho hệ thống máy chủ Linux.

4.1.1.Mã hóa dữ liệu được truyền đi

Bất kỳ ai nằm trên cùng một mạng đều có thể sử dụng một trình bắt gói tin (packet sniffer) để tóm lấy các thông tin (như thông số cấu hình mạng, username, password, câu lệnh, file, …) chưa được mã hóa được gửi bởi các chương trình như FTP, Telnet, Rlogin/Rsh. Giải pháp chung cho việc này là chuyển sang sử dụng các chương trình như OpenSSH , SFTP, hoặc FTPS.

4.1.2.Giảm tối thiểu các gói phần mềm được cài đặt

Hãy tránh cài đặt các gói phần mềm không cần thiết để tránh nguy cơ lỗ hổng ẩn chứa trong các phần mềm đó bị khai thác. Sử dụng các trình quản lý gói như *yum, rpm, apt-get, dpkg*… để xem tất cả các gói đã cài trên hệ thống. Sau đó, xóa bỏ các gói không cần thiết.

Đối với Redhat-based Distro

 # yum list installed

 # yum list packageName

 # yum remove packageName

Hoặc với Debian-based Distro

 # dpkg –list

 # dpkg –info packageName

 # apt-get remove packageName

4.1.3.Mỗi dịch vụ mạng chạy trên một hệ thống thực (hoặc máy ảo) riêng biệt

Nên chạy các dịch vụ mạng khác nhau trên các server tách biệt nhau. Điều này giúp giảm thiểu rủi ro các dịch vụ sẽ bị "chết chùm" khi chúng nằm trên cùng một server. Ví dụ, nếu một hacker có thể khai thác thành công một lỗ hổng của phần mềm như Apache, hắn sẽ có toàn quyền truy cập vào server và gây ảnh hưởng cho các dịch vụ khác như MySQL, e-mail nằm trên cùng server với Apache.

4.1.4.Cập nhật đầy đủ, thường xuyên cho Linux kernel và các phần mềm khác

Áp dụng các bản vá bảo mật là một công việc quan trọng trong kế hoạch bảo trì Linux sever. Linux cung cấp tất cả các công cụ cần thiết để đảm bảo hệ thống của bạn luôn được cập nhật, đồng thời giúp nâng cấp dễ dàng giữa các phiên bản. Bạn nên kiểm duyệt và áp dụng tất cả các bản cập nhật ngay khi có thể. Sử dụng *yum, apt-get*... để thực hiện việc cập nhật này.

 # yum update

hoặc

 # apt-get update && apt-get upgrade

4.1.5.Quản lý tài khoản người dùng và chính sách mật khẩu mạnh

Sử dụng lệnh *useradd/usermod* để tạo và quản lý các tài khoản người dùng. Hãy đảm bảo rằng việc áp dụng chính sách mật khẩu mạnh. Ví dụ, một mật khẩu mạnh bao gồm ít nhất 8 ký tự và kết hợp cả chữ cái, chữ số, ký tự đặc biệt, chữ hoa, chữ thường… Nhưng hầu hết mọi người đều chọn một mật khẩu dễ ghi nhớ. Điều này tạo cơ hội để hacker sử dụng các công cụ như John the ripper để dò tìm các mật khẩu yếu trên server của bạn. Cấu hình file pam_cracklib.so để thực thi chính sách mật khẩu mà bạn mong muốn.

4.1.5.1.Thời gian tồn tại của mật khẩu (password aging)

Lệnh chage cho phép thay đổi số ngày giữa các lần thay đổi mật khẩu và ngày thay đổi mật khẩu lần cuối. Dựa vào hai thông tin này, hệ thống sẽ xác định xem khi nào người dùng cần thay đổi mật khẩu của họ. File **/etc/login.defs** cũng bao gồm cấu hình password aging.

Để vô hiệu hóa password aging, tức là mật khẩu sẽ không bao giờ hết hạn gõ lệnh sau:

 # chage -M 99999 userName

Để xem thông tin về thời gian mãn hạn của mật khẩu, gõ:

 # chage -l userName

4.1.5.2.Hạn chế sử dụng lại các mật khẩu trước đó

Trong Linux, có thể ngăn chặn tất cả các người dùng sử dụng lại các mật khẩu cũ, đã từng sử dụng trước đó. Tham số remember của module pam_unix cho biết số lượng các mật khẩu trước đó sẽ không được sử dụng lại.

4.1.5.3.Khóa tài khoản sau một số lần đăng nhập thất bại

Trong Linux, có thể sử dụng lệnh *faillog* để hiển thị các bản ghi hoặc để thiết lập giới hạn số lần đăng nhập thất bại. Để mở khóa cho tài khoản sau khi tài khoản này bị khóa sau một số lần đăng nhập thất bại, gõ:

faillog -r -u userName //–reset = -r

Lưu ý: có thể sử dụng lệnh passwd để khóa hoặc mở khóa tài khoản

 # passwd -l userName //khóa tài khoản

 # passwd -u userName //mở khóa tài khoản

4.1.5.4.Xác định các tài khoản sử dụng mật khẩu rỗng

Gõ lệnh sau:

 # awk -F: '($2 == "") {print}' /etc/shadow

Khóa tài khoản có mật khẩu rỗng

 # passwd -l accountName

4.1.5.5.Đảm bảo rằng không có người dùng thông thường nào có UID = 0

Chỉ có tài khoản root có UID =0 với quyền hạn cao nhất để truy cập vào hệ thống. Gõ lệnh sau để hiển thị tất cả các tài khoản với UID = 0

awk -F: '($3 == "0") {print}' /etc/passwd

Sẽ thấy ít nhất một dòng tương ứng với tài khoản root như sau:

root:x:0:0:root:/root:/bin/bash

Trường thứ 3 cho biết giá trị UID. Nếu có thêm các tài khoản khác có UID =0, hãy đổi lại UID cho những tài khoản này sử dụng lệnh sau:

 # usermod –u new_UID userName

4.1.6.Đừng bao giờ đăng nhập với tài khoản root

Hạn chế tối đa đăng nhập với tài khoản root. Thay vào đó, nên sử dụng sudo để thực thi các lệnh với quyền root khi cần thiết mà không cần phải cung cấp mật khẩu của root. Ngoài ra, sudo cũng cung cấp các tính năng auditing và tracking giúp bạn biết được người dùng đã chạy những lệnh sudo nào.

4.1.7.Bảo mật vật lý cho máy chủ

Cấu hình lại BIOS và vô hiệu hóa việc khởi động từ các thiết bị ngoại vi như DVD/CD/USB. Nên đặt mật khẩu bảo vệ cho GRUB. Những máy chủ quan trọng cần được khóa cẩn thận trong các IDC (Internet Data Center) và tất cả mọi người phải trải qua các bước kiểm tra an ninh trước khi truy cập vào server của bạn.

4.1.8.Tắt hết các dịch vụ không cần thiết

Phải cần loại bỏ tất cả các dịch vụ (service hoặc daemon) không cần thiết khỏi quá trình khởi động của hệ thống. Điều này giúp gia tăng tốc độ hoạt động cũng như sự ổn định của hệ thống và quan trọng hơn là giúp server giảm thiểu được nguy cơ bị tấn công bề mặt (surface attack) vào các dịch vụ ẩn chứa lỗ hổng. Sử dụng lệnh chkconfig để liệt kê tất cả các dịch vụ được khởi chạy cùng hệ thống ở runlevel 3

 # chkconfig –list | grep ’3:on’

Để tắt một dịch vụ nào đó, gõ:

 # service serviceName stop

Hoặc

 # chkconfig serviceName off

4.1.9.Gỡ bỏ X Windows

Thành phần đồ họa X Windows thật không cần thiết chạy trên Linux server. Không có lý do nào lại chạy X Windows trên Mail server hoặc Apache web server . Chúng ta có thể loại bỏ hoàn toàn X Windows để nâng cao hiệu suất hoạt động và bảo mật cho server theo các bước sau:

- Chỉnh sửa file **/etc/inittab** để thiết lập cho Linux khởi chạy ở runlevel 3.
- Xóa X Windows system bằng lệnh yum

 # yum groupremove "X Window System"

4.1.10.Bảo vệ Linux kernel với /etc/sysctl.conf

File **/etc/sysctl.conf** được sử dụng để cấu hình các tham số cho kernel tại thời điểm kernel được thực thi. Linux đọc và áp dụng các thiết lập trong file này tại quá trình khởi động. Dưới đây là nội dung mẫu của file **/etc/sysctl.conf**

 # Turn on execshield

kernel.exec-shield=1

kernel.randomize_va_space=1

 # Enable IP spoofing protection

net.ipv4.conf.all.rp_filter=1

 # Disable IP source routing

net.ipv4.conf.all.accept_source_route=0

 # Ignoring broadcasts request

net.ipv4.icmp_echo_ignore_broadcasts=1

net.ipv4.icmp_ignore_bogus_error_messages=1

 # Make sure spoofed packets get logged

net.ipv4.conf.all.log_martians = 1

4.1.11.Chia tách các phân vùng ổ cứng

Tách biệt các file của hệ điều hành khỏi các chương trình, dữ liệu cá nhân của người dùng. Hay nói cách khác, đặt các loại dữ liệu này trên các phân vùng khác nhau, sẽ giúp hệ thống chạy ổn định và an toàn hơn. Đảm bảo rằng các tập tin hệ thống dưới đây được gắn (mount) trên các phân vùng riêng rẽ:

/usr

/home

/var và /var/tmp

/tmp

Tạo thêm các phân vùng tách biệt cho Apache và FTP server. Chỉnh sửa file **/etc/fstab** và chắc rằng đã thêm vào các tùy chọn cấu hình sau:

- noexec: Không gán quyền thực thi (execute) cho các file nhị phân (binary) trên phân vùng này. Điều này sẽ ngăn cản việc thực thi file binary nhưng cho phép chạy các script.
- nodev: Không cho phép (hoặc diễn dịch) các thiết bị kiểu character hoặc thiết bị kiểu block trên phân vùng này.
- nosuid: Các SUID/SGID sẽ mất hiệu lực trên phân vùng này.

Ví dụ, một dòng mẫu trong file /etc/fstab dưới đây sẽ giới hạn việc truy cập của người dùng trên phân vùng /dev/sda5 (thư mục gốc trên FPT server):

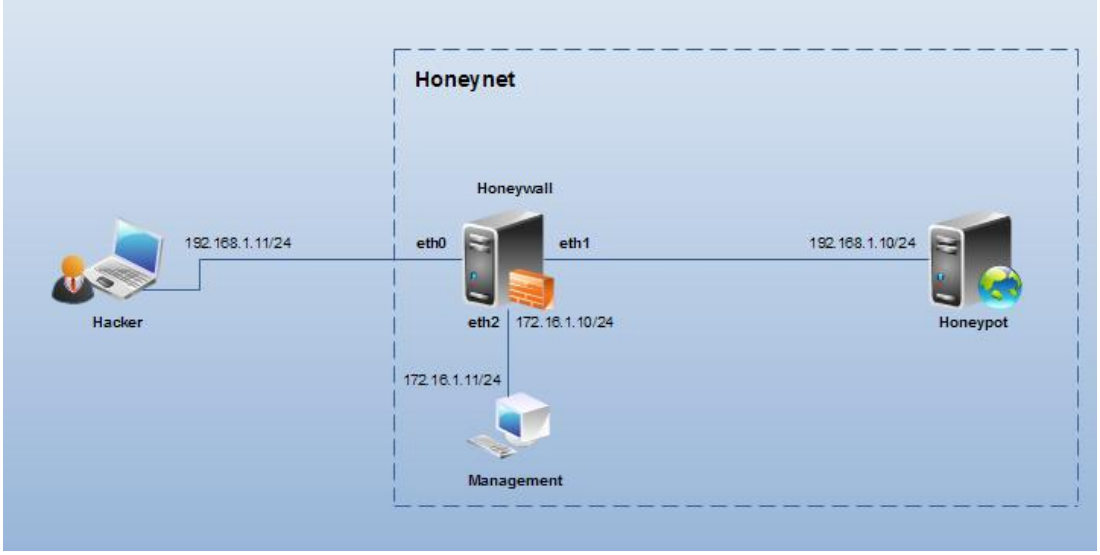
/dev/sda5 /ftpdata ext3 defaults,nosuid,nodev,noexec 1 2

4.1.12.Tạm thời tắt IPv6

Internet Protocol version 6 (IPv6) là một giao thức mới hoạt động ở tầng Internet của bộ giao thức mạng TCP/IP. IPv6 đang dần thay thế cho IPv4 do những lợi ích mà nó mang lại. Hiện tại, vẫn chưa có công cụ tốt nào có thể kiểm tra các vấn đề bảo mật trên hệ thống hỗ trợ IPv6. Hầu hết các Distro Linux đã bắt đầu hỗ trợ giao thức IPv6. Các cracker có thể gửi các lưu lượng độc hại thông qua IPv6 vì hầu hết các quản trị viên chưa giám sát các lưu lượng IPv6 này. Vì thế, trừ khi hệ thống mạng yêu cầu sử dụng IPv6, tốt nhất bạn nên tạm thời vô hiệu hóa IPv6 hoặc cấu hình trên Firewall để cản lọc các gói tin IPv6.

4.2.MÔ HÌNH TRIỂN KHAI

Do điều kiện khách quan thiếu thiết bị, không có địa chỉ IP tĩnh. Nhóm đề tài đã triển khai hệ thống Honeynet trên hệ thống máy ảo.



Hình 4.2 - Mô hình triển khai trong đề tài

Bảng cấu hình các thành phần trong Honeynet

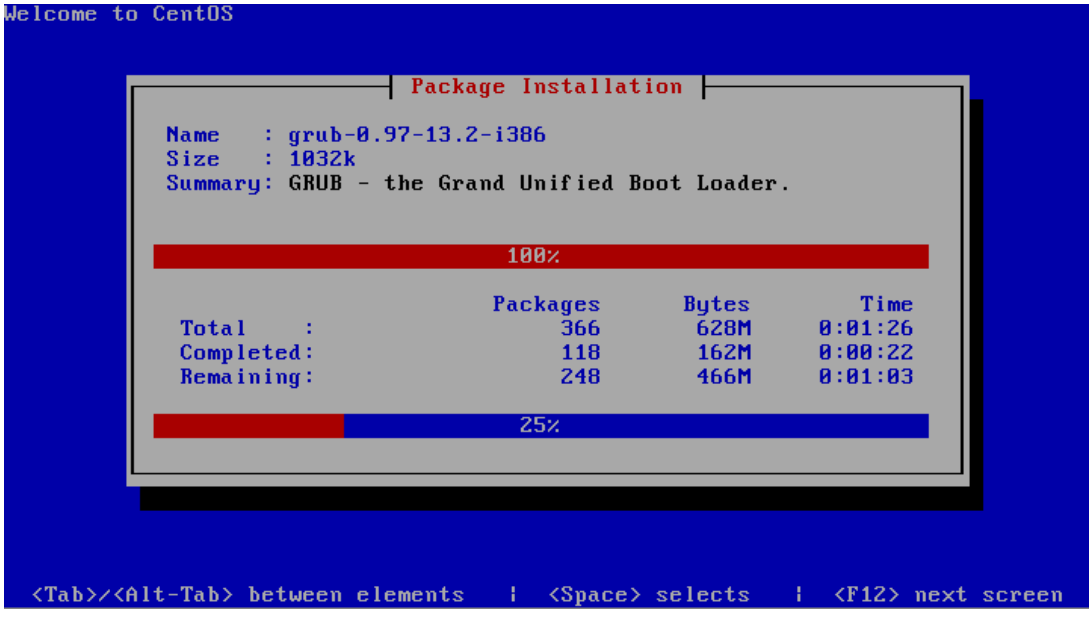
Loại	Hệ điều hành
Server (Honeynet gateway){3 card mạng}	Honeywall Roo 1.4
Web Server (Honeypot)	Windows server 2003
Client (Management)	Windows XP SP3
Client (Hacker)	Windows XP SP3

4.3.CÀI ĐẶT VÀ CẤU HÌNH HỆ THỐNG HONEYNET

4.3.1.Cài đặt và cấu hình Honeywall

Bước cài đặt Honeywall khá đơn giản, sau khi bỏ đĩa Honeywall CDROM vào và boot từ CD, sẽ xuất hiện màn hình cài đặt và cảnh báo dữ liệu trên ổ cứng sẽ bị xóa hết. Nếu đồng ý chỉ cần gõ phím Enter và sau đó quá trình cài đặt sẽ được diễn ra tự động.

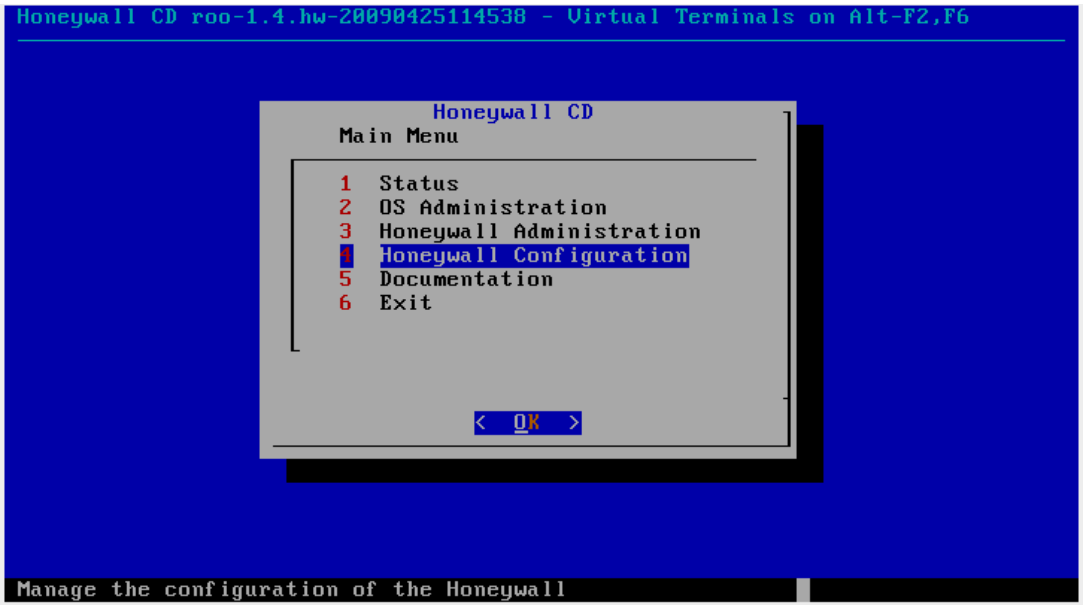
Hình 4.3.1a - Màn hình cảnh báo và cài đặt Honeywall



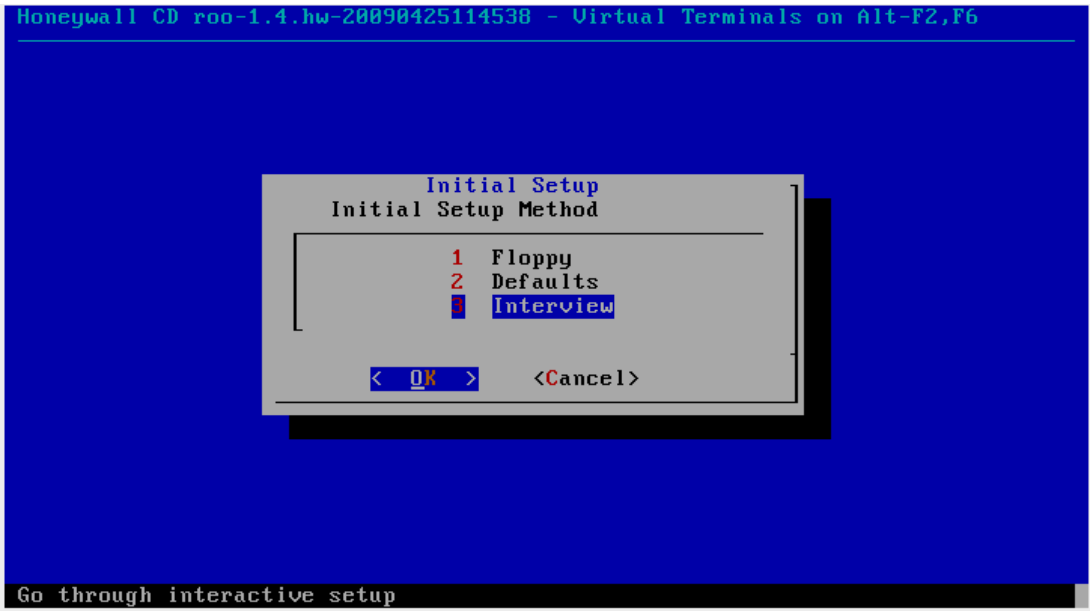
Hình 4.3.1b - Quá trình cài đặt Honeywall

Sau khi cài đặt xong hệ thống sẽ tự khởi động lại và xuất hiện như hình 4.3.1c. Mặc định hệ thống sẽ có hai tài khoản roo và root, để đảm bảo an toàn phải đăng nhập vào roo trước sau đó mới dùng lệnh *su* - để chuyển vào root và cấu hình. Cấu hình Honeywall có hai cách: bằng giao diện đồ họa và dòng lệnh, về bản chất thì nó giống nhau. Giao diện dòng lệnh sẽ cấu hình tại tập tin **/etc/Honeywall.conf.** Ở đây nhóm sẽ cấu hình bằng giao diên đồ họa.

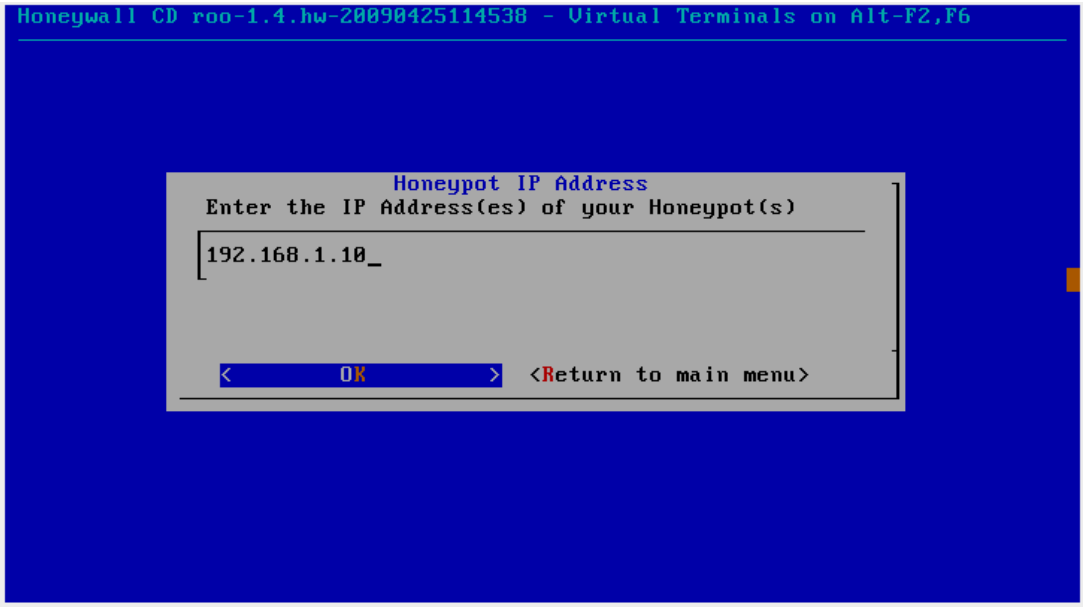
Hình 4.3.1c - Màn hình đăng nhập hệ thống



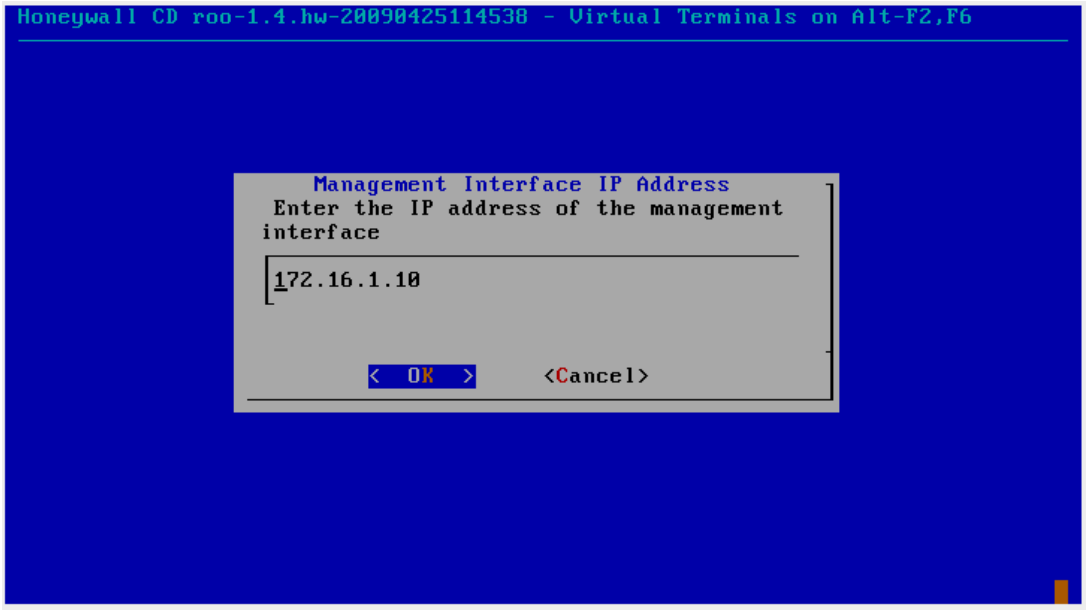
Hình 4.3.1d - Màn hình cấu hình hệ thống



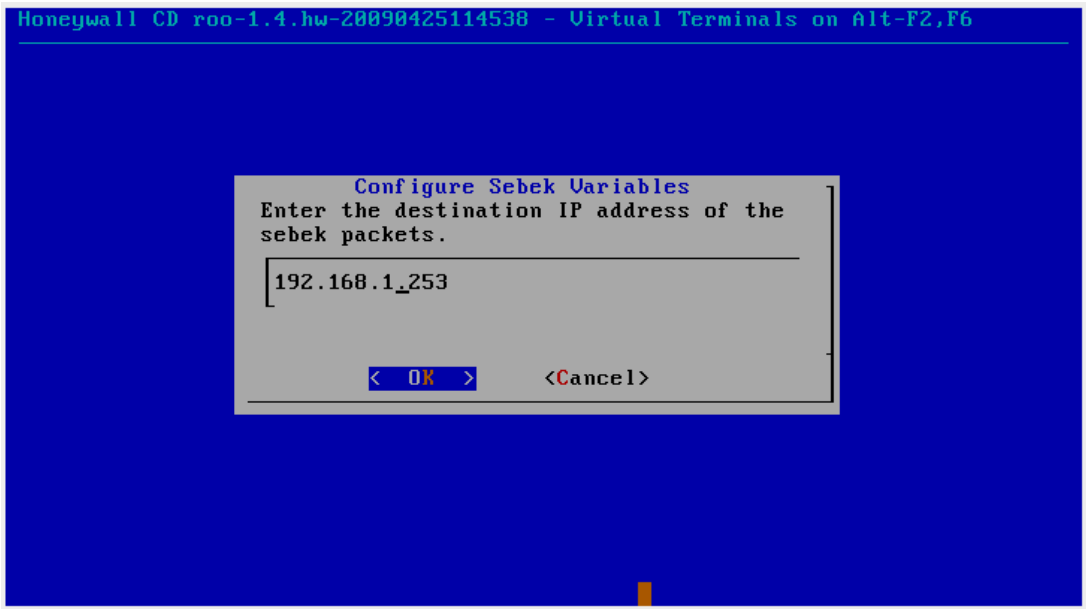
Hình 4.3.1e - Lựa chọn phương thức cấu hình



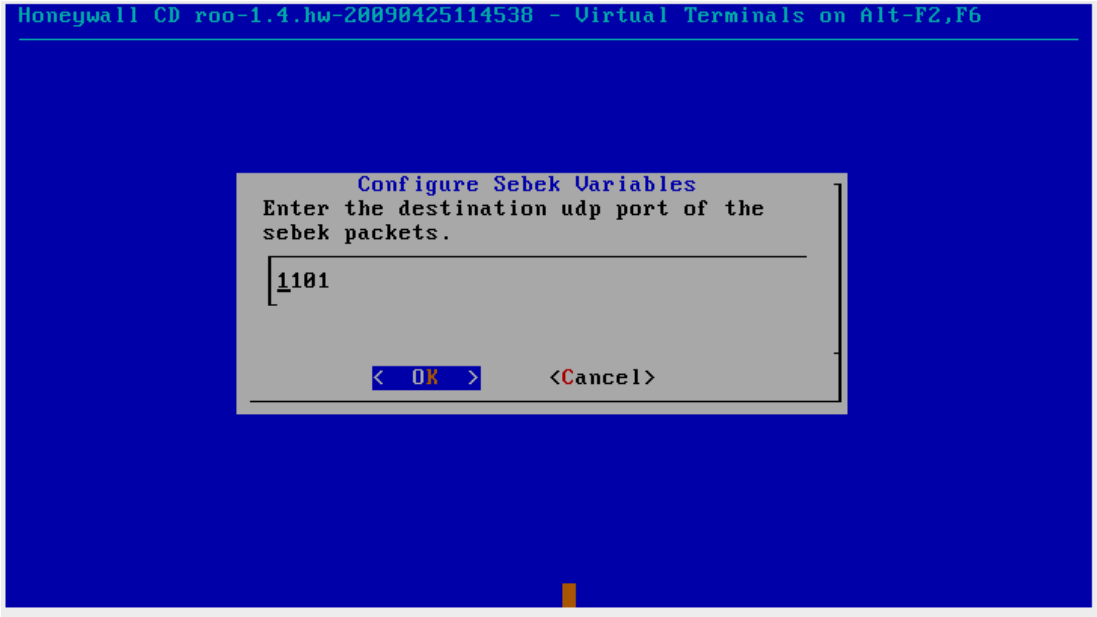
Hình 4.3.1f - Địa chỉ IP của các Honeypot



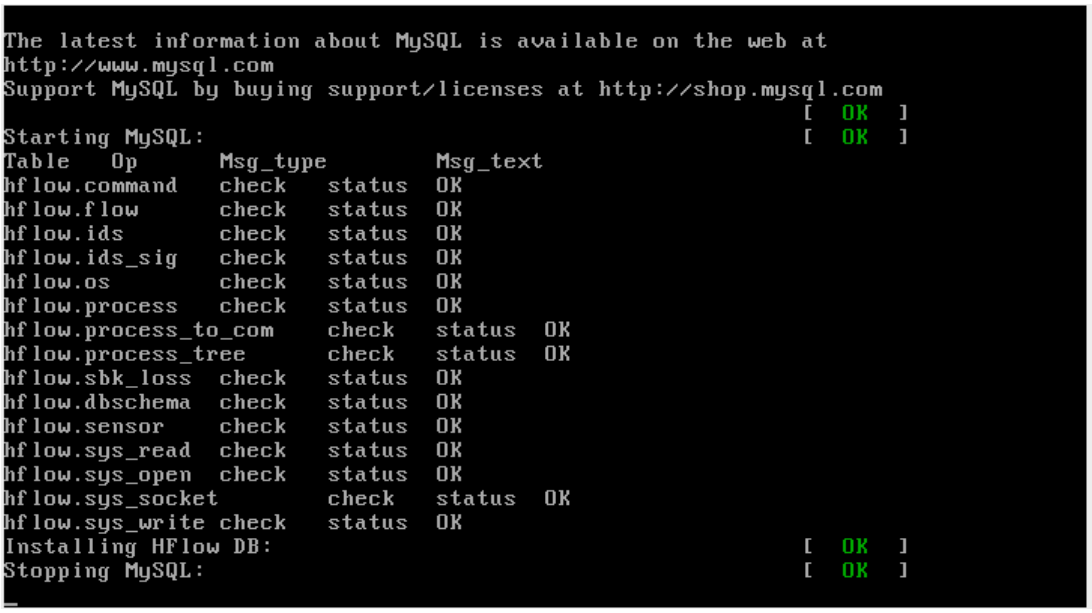
Hình 4.3.1g - Địa chỉ IP card quản lý



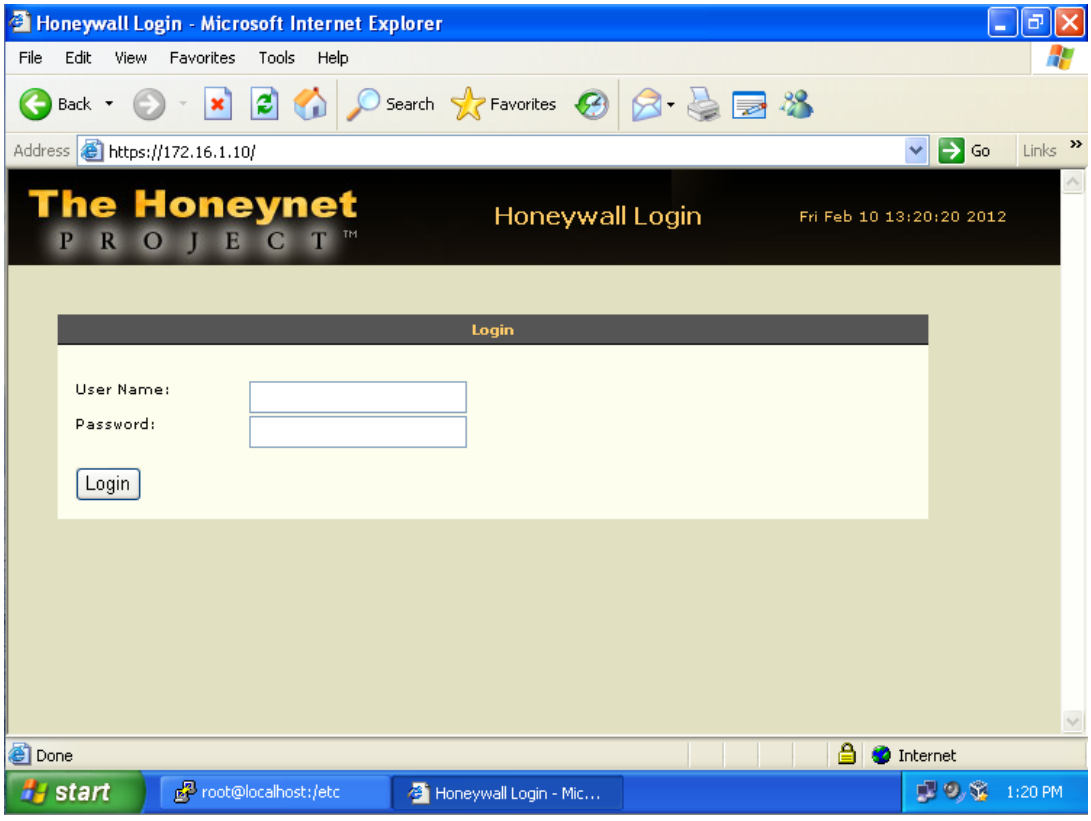
Hình 4.3.1h - Địa chỉ IP đích của gói tin Sebek



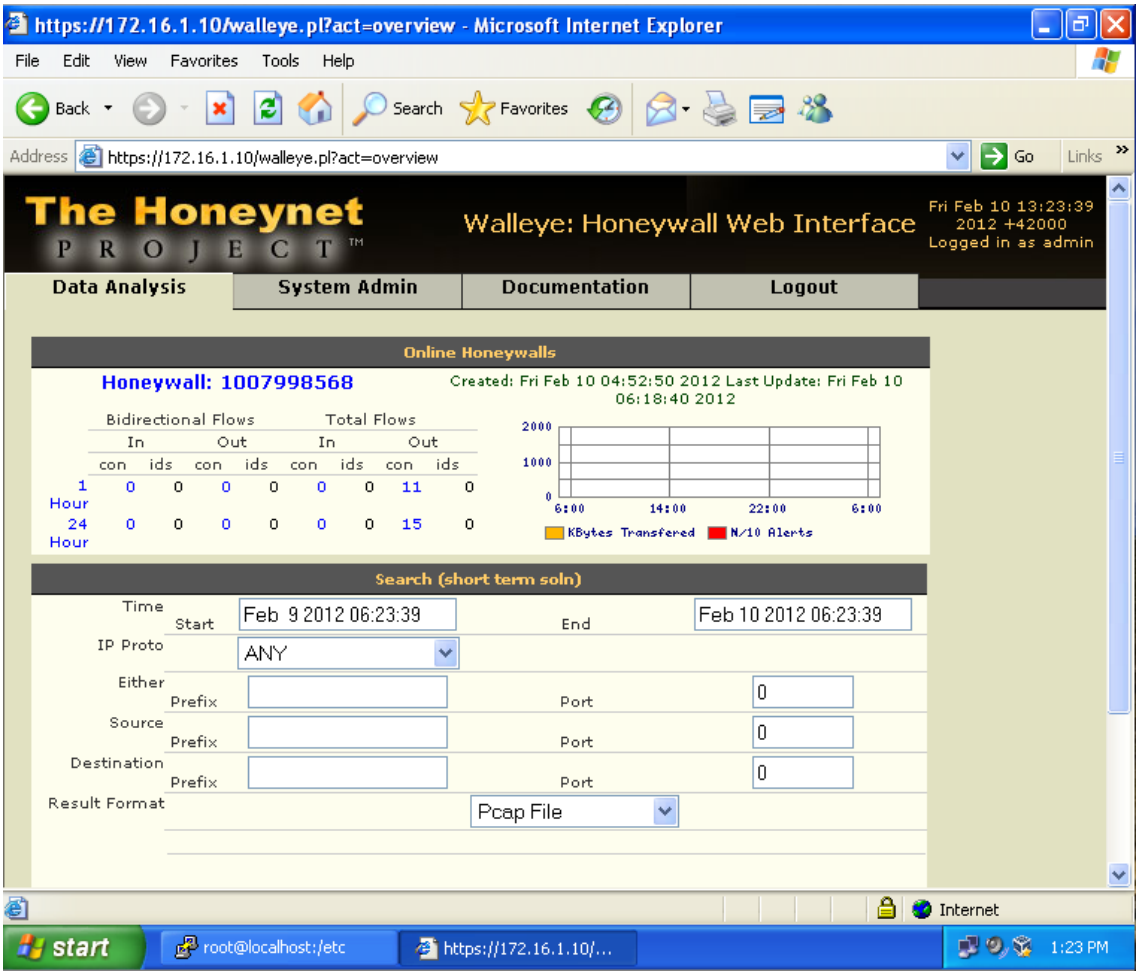
Hình 4.3.1i - Cổng đích của gói tin Sebek



Hình 4.3.1j - Hệ thống khởi động lại sau khi cấu hình



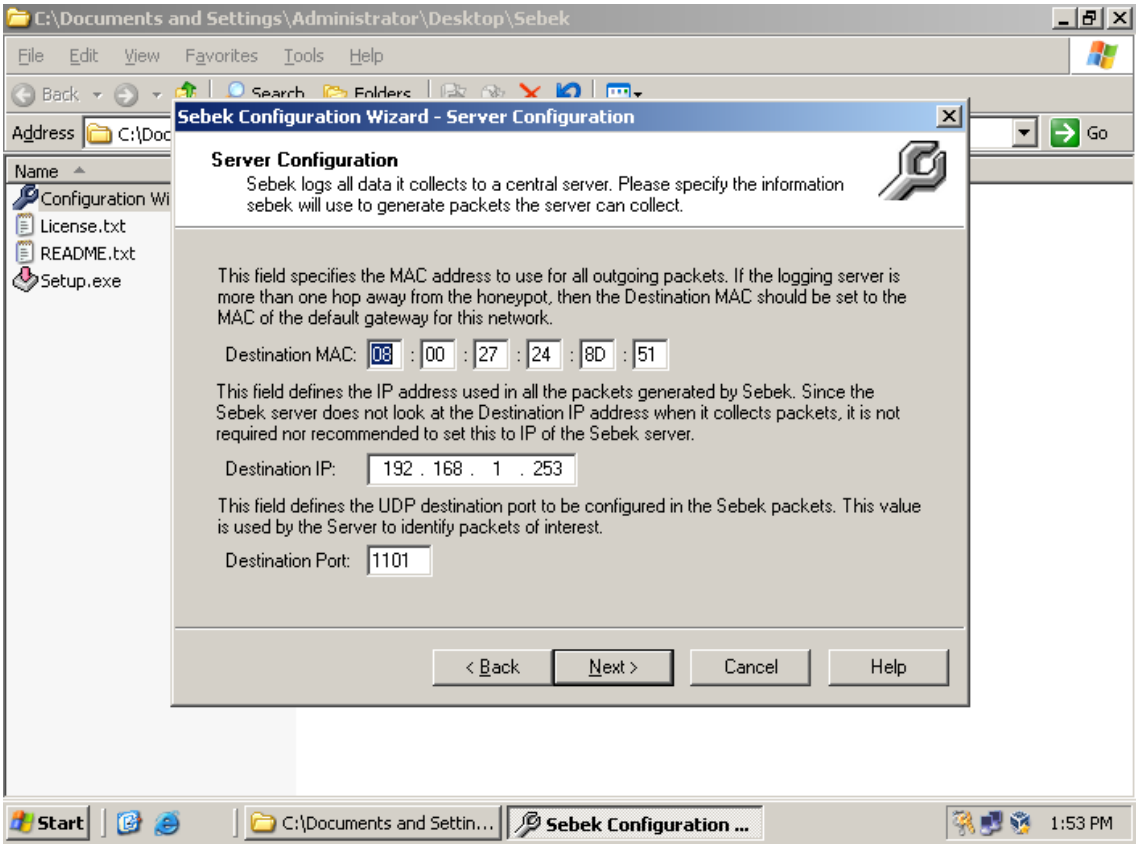
Hình 4.3.1k - Màn hình đăng nhập walleye từ máy quản lý



Hình 4.3.1l - Giao diện phân tích dữ liệu

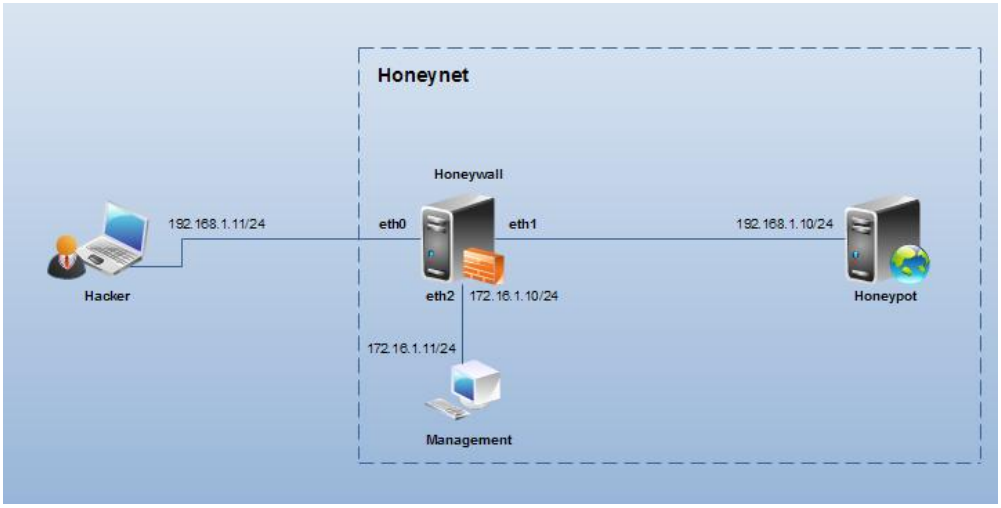
4.3.2.Cài đặt và cấu hình Sebek (client)

Việc cài đặt Sebek client nhằm mục đích gửi dữ liệu về các hành động của hacker trên Honeypot về Sebek server để người quản trị phân tích. Việc cài đặt trên khá đơn giản, chỉ cần tải tập tin cài đặt **Sebek-Win32-latest.zip** từ trang chủ và giải nén ra. Chạy file Setup.exe và sau đó chạy tập tin **Configuration Winzard.exe** để cấu hình cổng, địa chỉ IP đích của gói tin Sebek, địa chỉ MAC của card mạng mà Honeywall kết nối với Honeypot.



Hình 4.3.2 - Màn hình cấu hình Sebek client

4.4.KIỂM TRA HỆ THỐNG



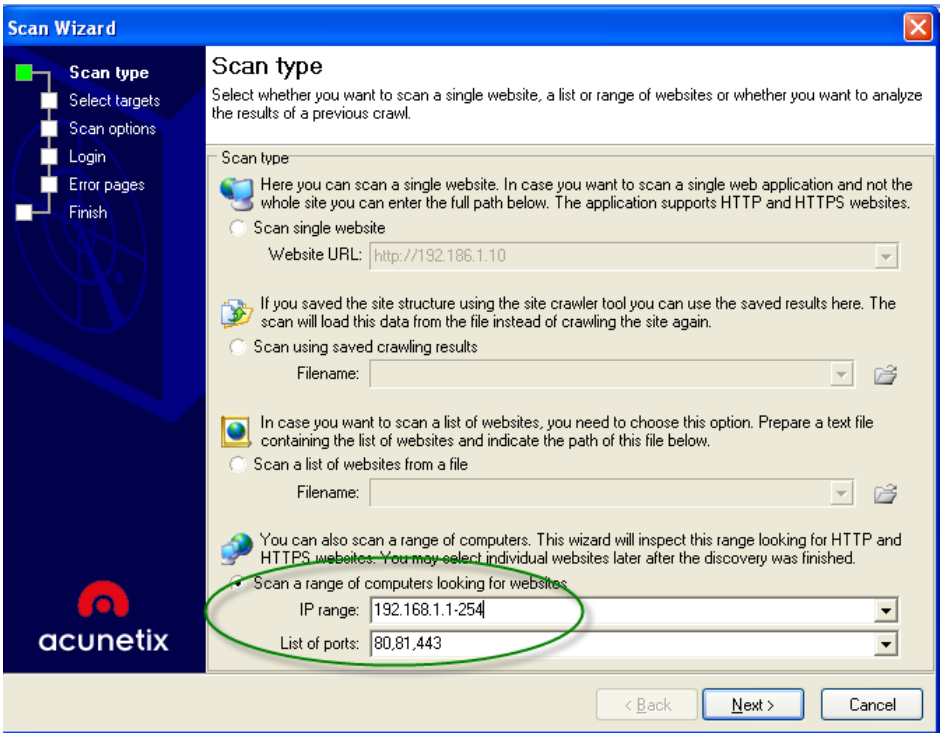
Hình 4.4 - Mô hình tấn công, kiểm tra hệ thống

Hacker sau khi thực hiện dò quét trên mạng để tìm kiếm và phát hiện ra Web Server (địa chỉ IP 192.168.1.10) của hệ thống và thu thập tất cả các thông tin trên máy Web Server này.

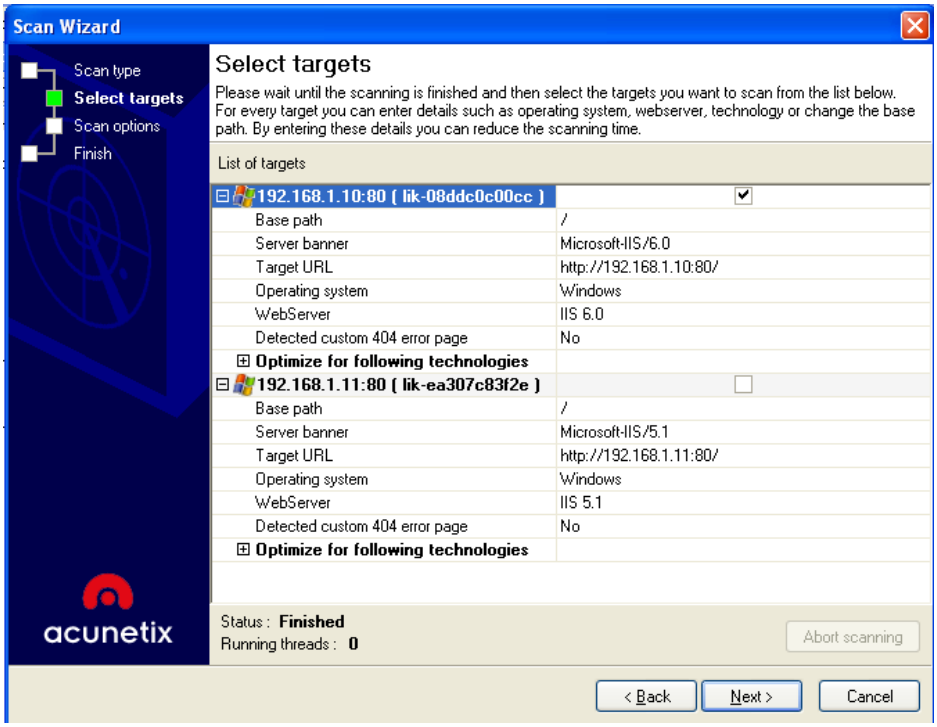
Toàn bộ thông tin quá trình tấn công của hacker vào hệ thống Honeynet sẽ được ghi lại đầy đủ, chi tiết. Những thông tin hệ thống Honeynet thu thập sẽ được người phân tích thực hiện phân tích, đánh giá dưới sự trợ giúp của công cụ quản lý và hỗ trợ phân tích Walleye trong Honeynet nhằm biết được quá trình tấn công của hacker.

4.4.1.Quá trình hacker thực hiện tấn công

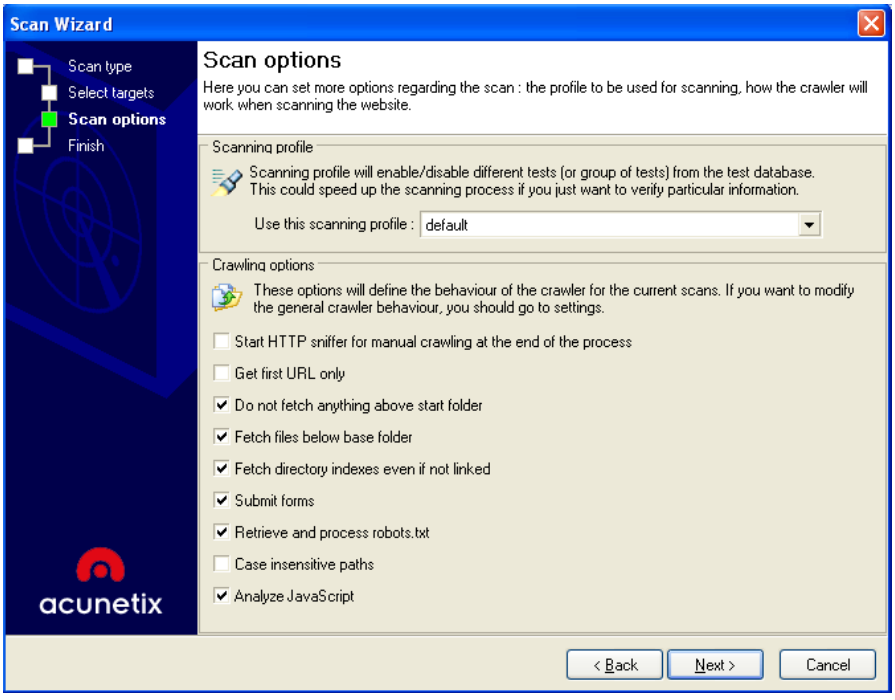
Đầu tiên hacker sử dụng chương trình Acunectix Web Vulnerability Scanner 4 để tiến hành quét lỗi các địa chỉ ip từ 1-254 và cổng 80,81,443.



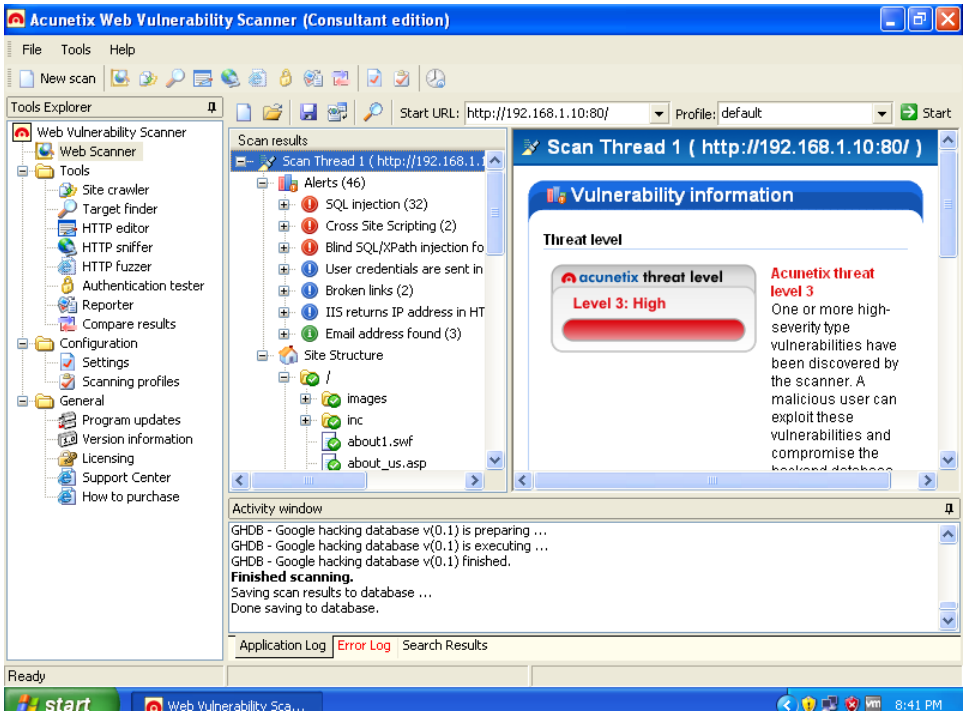
Hình 4.4.1a - Quét dải IP



Hình 4.4.1b - Chọn mục tiêu tấn công



Hình 4.4.1c - Tùy chỉnh cách thức quét



Hình 4.4.1d - Kết quả sau khi quét

Sau khi thực hiện dò quét hệ thống, hacker biết được Web server tồn tại lỗi SQL Injection, tiếp theo hacker thực hiện quá trình tấn công khai thác lỗi SQL Injection.

Quá trình tấn công SQL Injection:

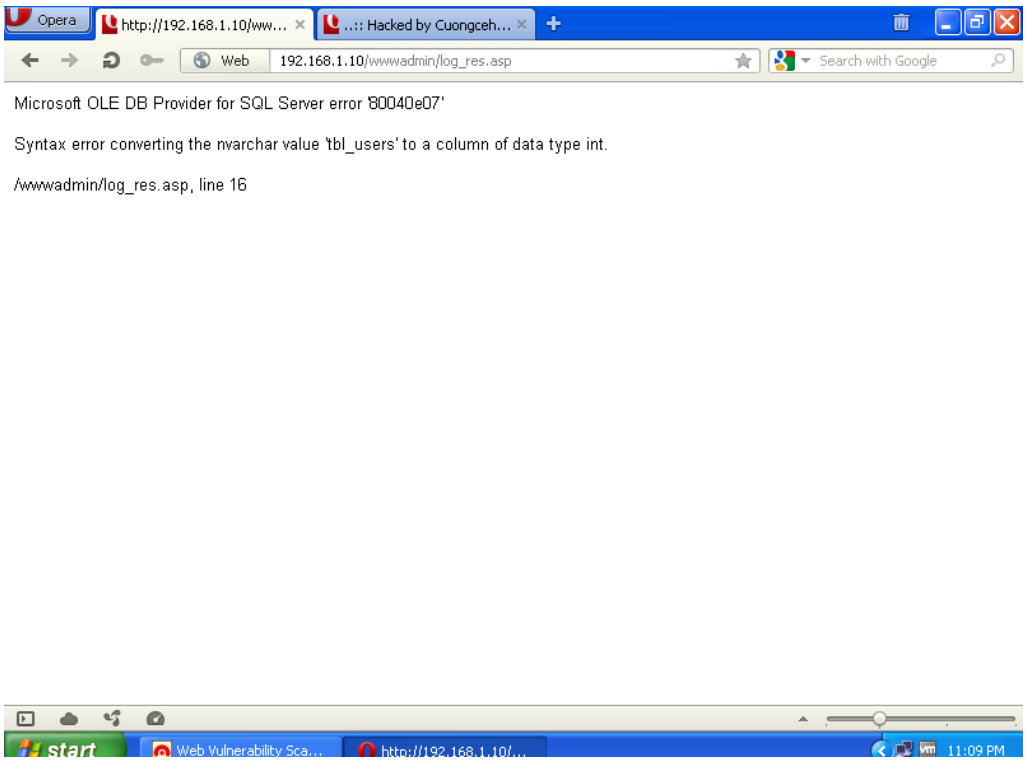
Sau khi quét được lỗ hổng SQL Injection trên máy mục tiêu, hacker tiến hành tiêm mã sql vào. Dựa trên các đối tượng sysobjects và syscolumns và bảng hệ thống INFORMATION_SCHEMA.TABLES để xác định tên bảng chứa các tài khoản quản trị và các trường của bảng này qua form đăng nhập thành viên của website tồn tại lỗi SQL Injection. Đầu tiên hacker phải dò được tên bảng chứa các tài khoản bằng cách gõ vào khung username:

' union select name,1,1 from sysobjects where name > 'a'-

Lưu ý: số cột trong hai câu SELECT phải khớp với nhau. Nghĩa là số lượng cột trong câu lệnh SELECT ban đầu và câu lệnh UNION SELECT phía sau bằng nhau và cùng kiểu. Vì ở đây, ta không biết số lượng các cột của bảng ở câu lệnh SELECT ban đầu nên ta buộc phải tìm bằng cách lần lượt tăng thêm các số '1' vào câu lệnh UNION SELECT.

Sau quá trình dò thì câu lệnh cuối cùng là:

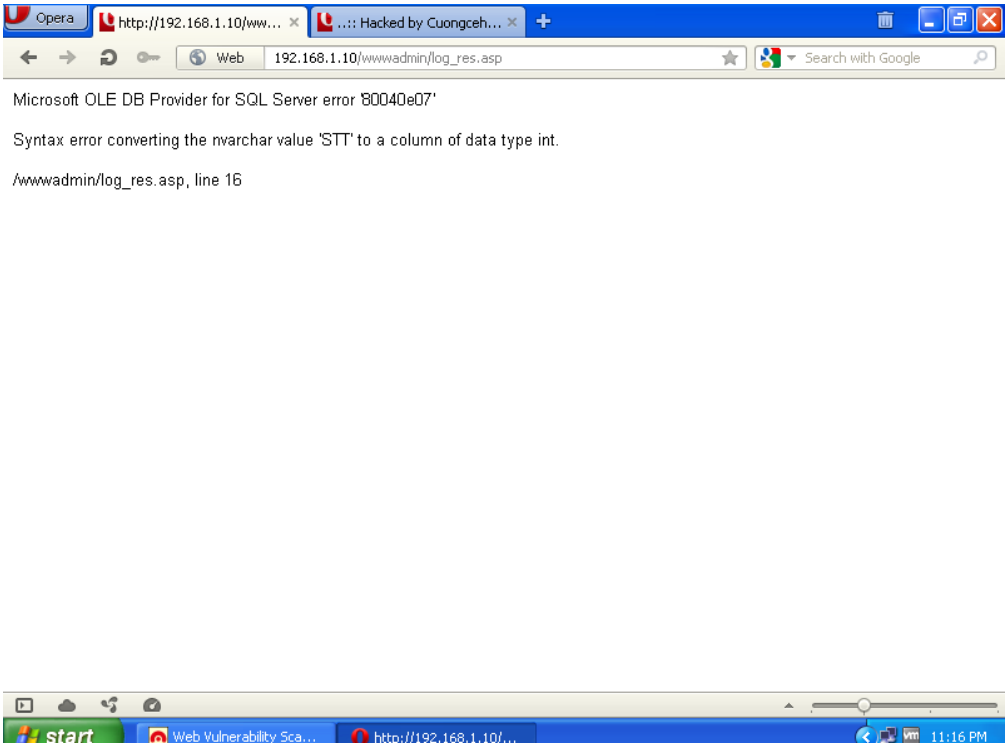
‘ union select name,1,1,1,1 from sysobjects where name > ‘tbl_u’-



Hình 4.4.1e - Màn hình lộ bảng tbl_users

Sau khi lộ ra bảng tbl_users, hacker tiếp tục khai thác các trường trong bảng đó bằng câu lệnh bên dưới và khai thác được cột STT

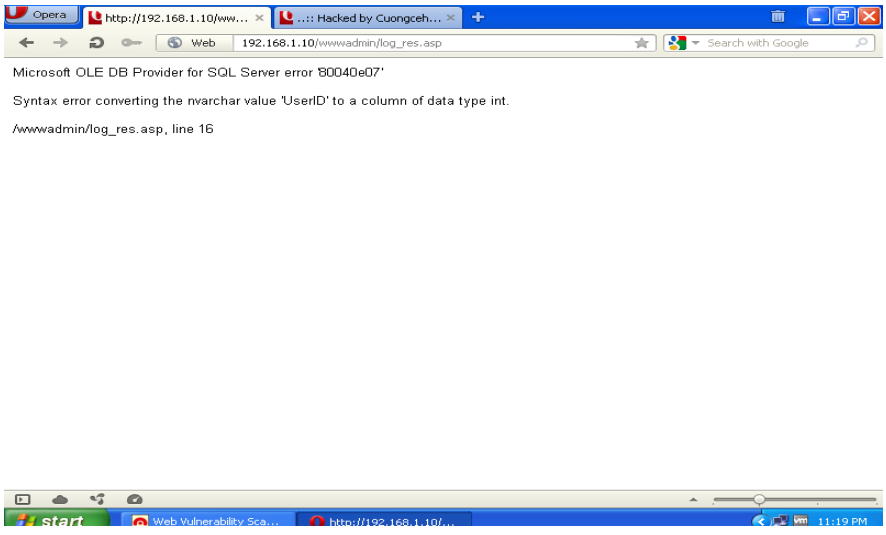
' union select column_name,1,1,1,1 from information_schema.columns where table_name='tbl_users'-



Hình 4.4.1f - Màn hình khai thác được cột STT

Để khai thác tiếp các cột khác trong bảng, hacker dùng câu lệnh:

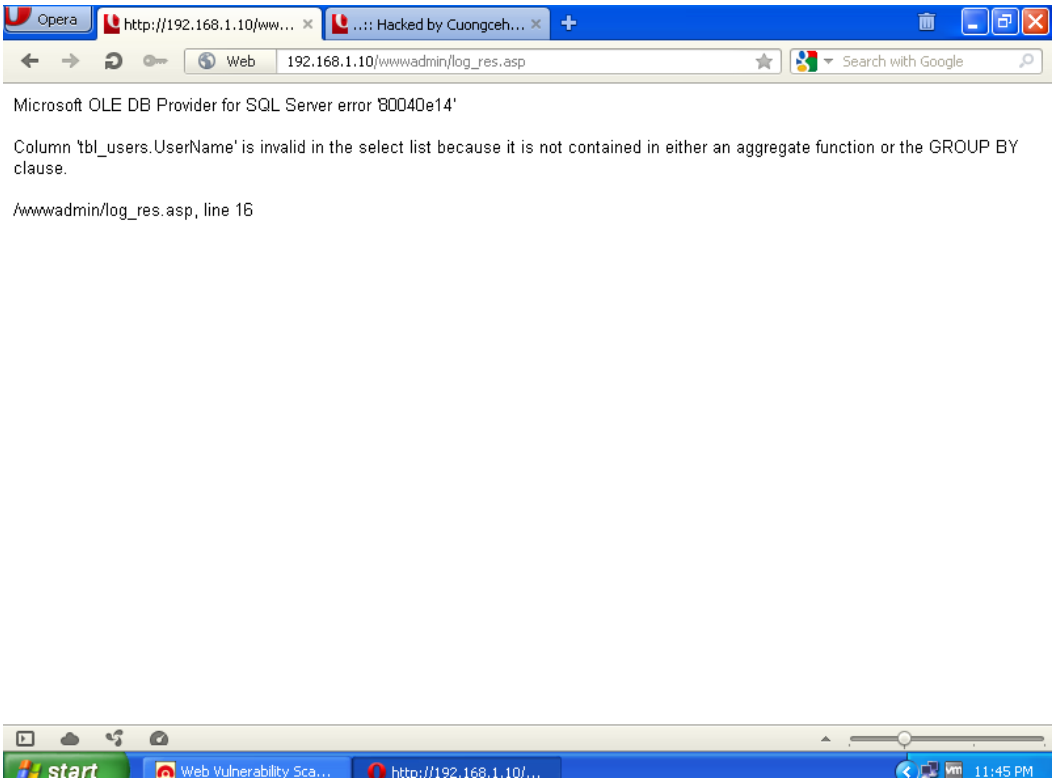
' union select column_name,1,1,1,1 from information_schema.columns where table_name='tbl_users' and column_name > 'STT'-



Hình 4.4.1g - Màn hình khai thác cột UserID

Cứ thế hacker sẽ thu được các cột: STT, UserID, UserName, Password, Passlen bằng câu lệnh:

*‘ union select * from tbl_users where UserID =’’ group by STT,UserID-*



Hình 4.4.1h - Màn hình khai thác cột UserName

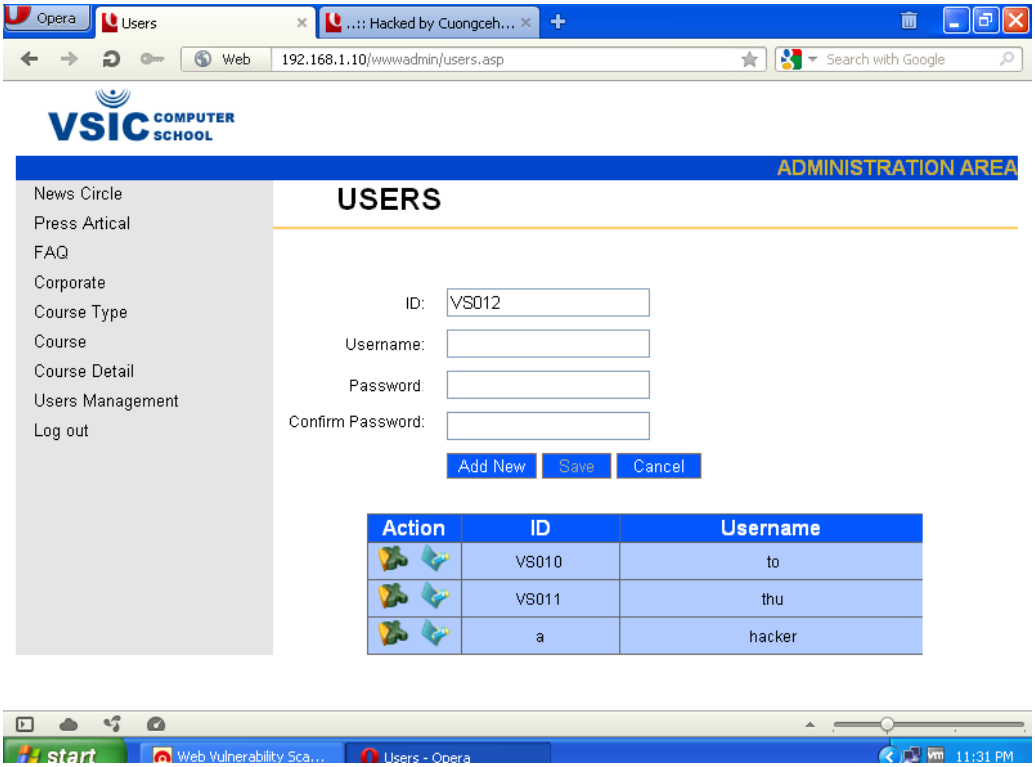
Tiếp đến hacker sẽ phải kiểm tra kiểu dữ liệu của từng cột bằng câu lệnh:

‘ union select sum(STT) from tbl_users-

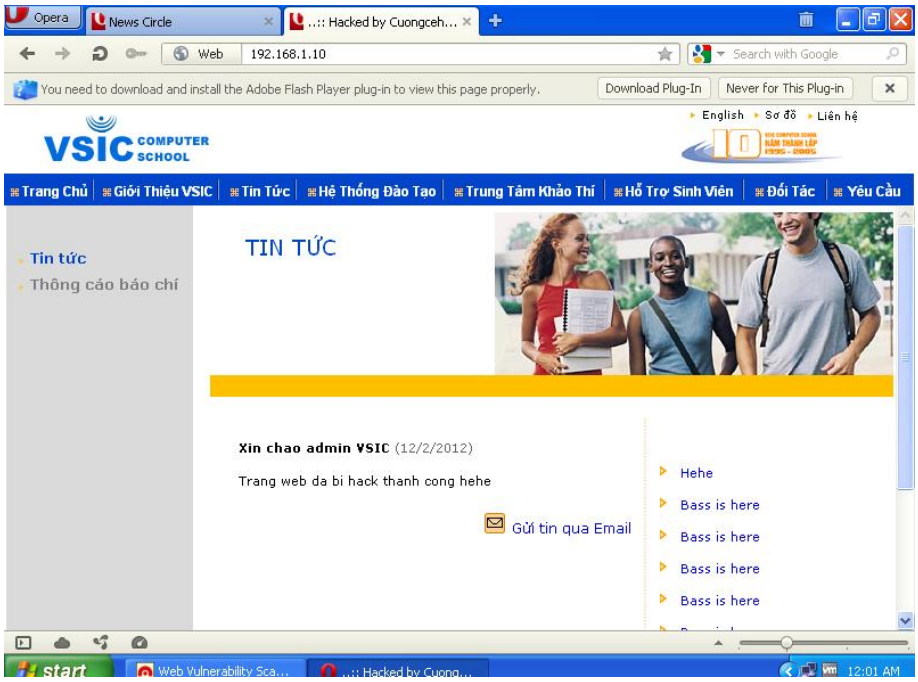
Nếu nhận được thông báo: “All queries in an SQL statement containing a UNION operator must have an equal number of expressions in their target lists”. Thì có thể hiểu là kiểu int. Còn nếu nhận được thông báo: “The sum or average aggregate operation cannot take a nvarchar data type as an argument” thì có thể hiểu là kiểu chuỗi.

Từ đó hacker sẽ tiêm câu lệnh thêm user vào cơ sở dữ liệu dựa trên những yếu tố đã khai thác là:

‘; insert into tbl_users values(9,’a’,’hacker’,’123’,3)-



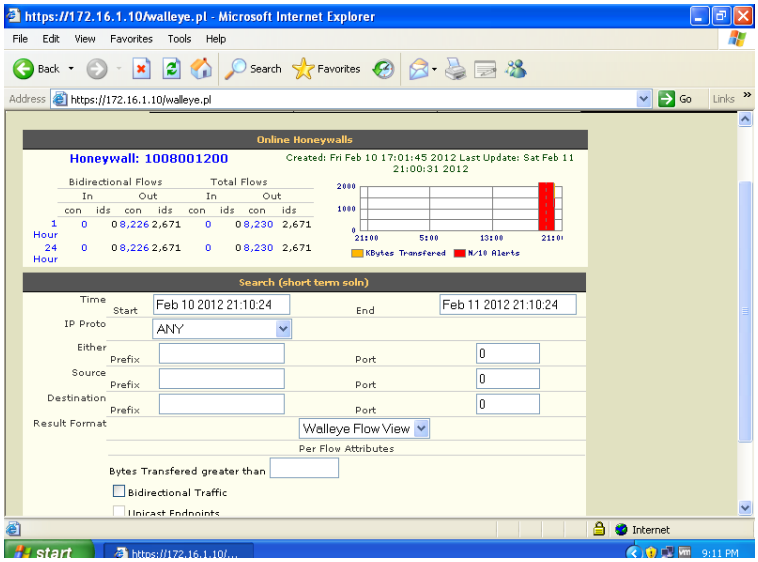
Hình 4.4.1i - Màn hình đăng nhập thành công của hacker vào trang quản trị



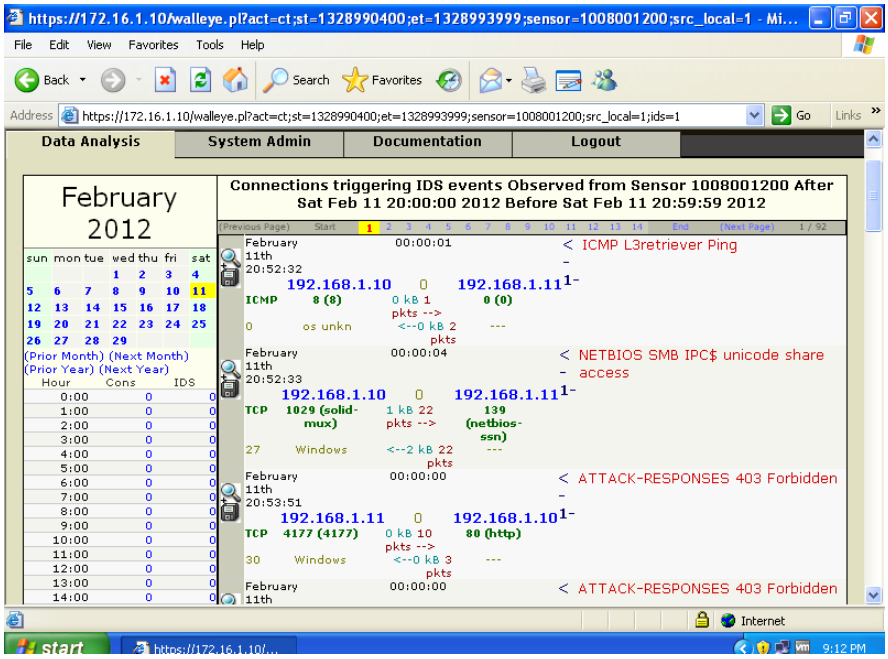
Hình 4.4.1j - Màn hình hacker thay đổi nội dung trên Web

4.4.2.Kết quả thu thập của hệ thống

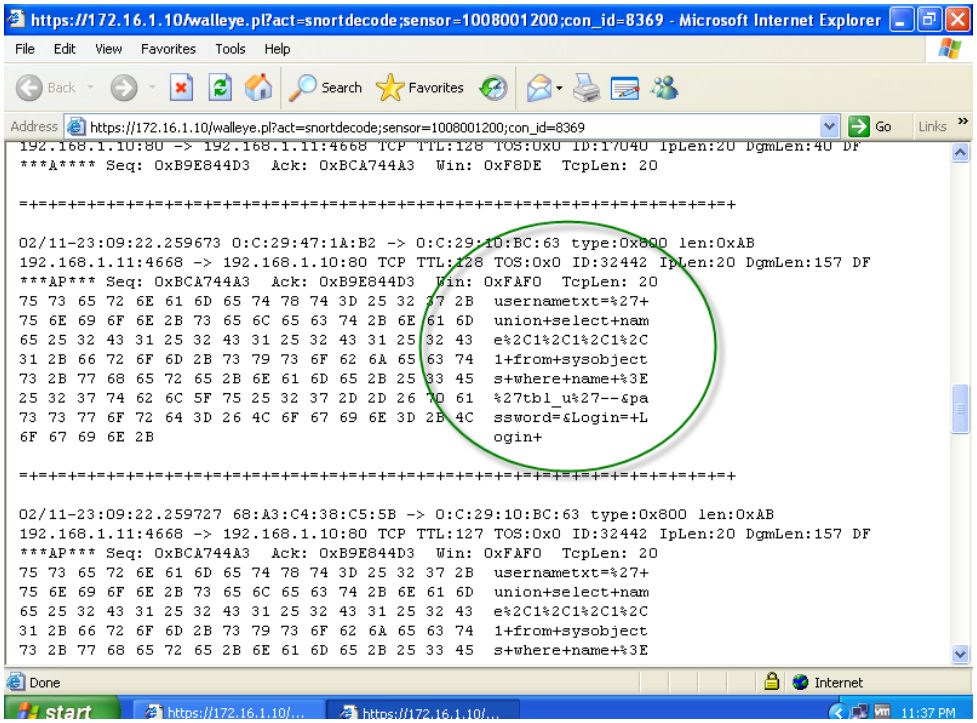
Từ máy quản lý ta truy cập vào Honeywall thông qua giao diện Web, sau đó tiến hành chọn lọc để xem và phân tích gói tin mà hệ thống Honeynet thu thập được.



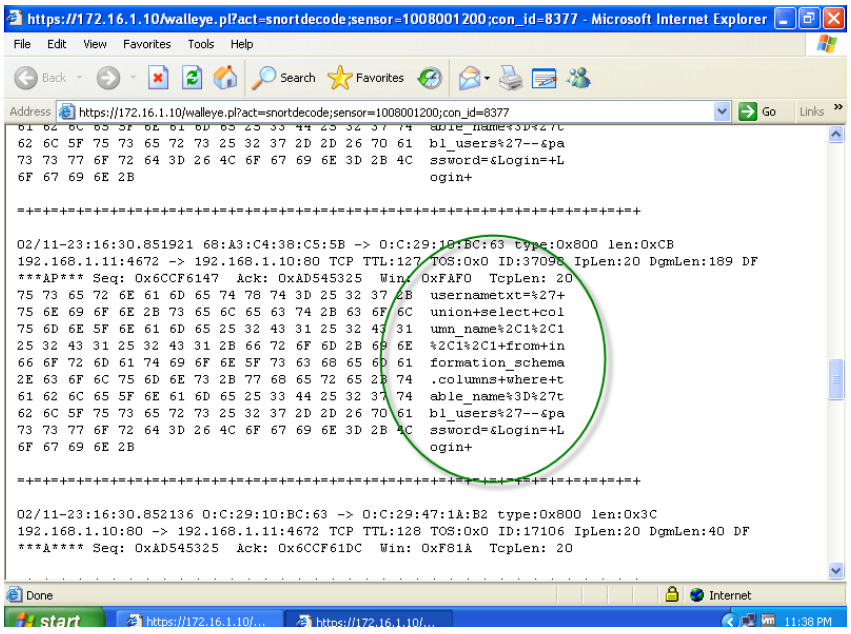
Hình 4.4.2a - Giao diện phân tích dữ liệu trên Walleye



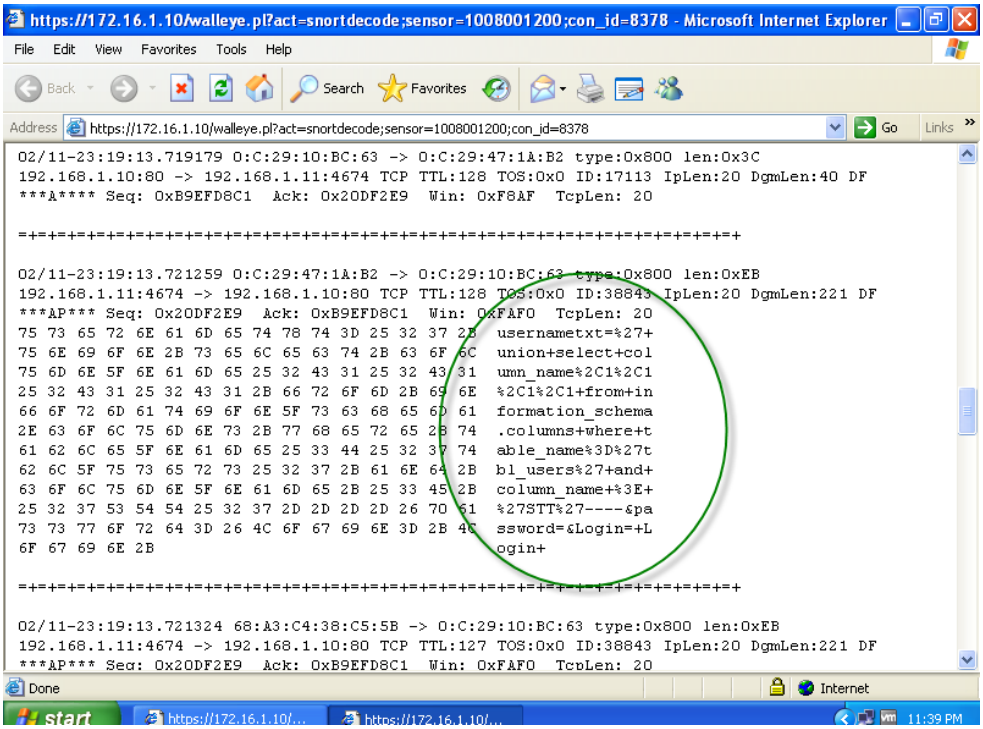
Hình 4.4.2b - Các gói tin thu nhận được



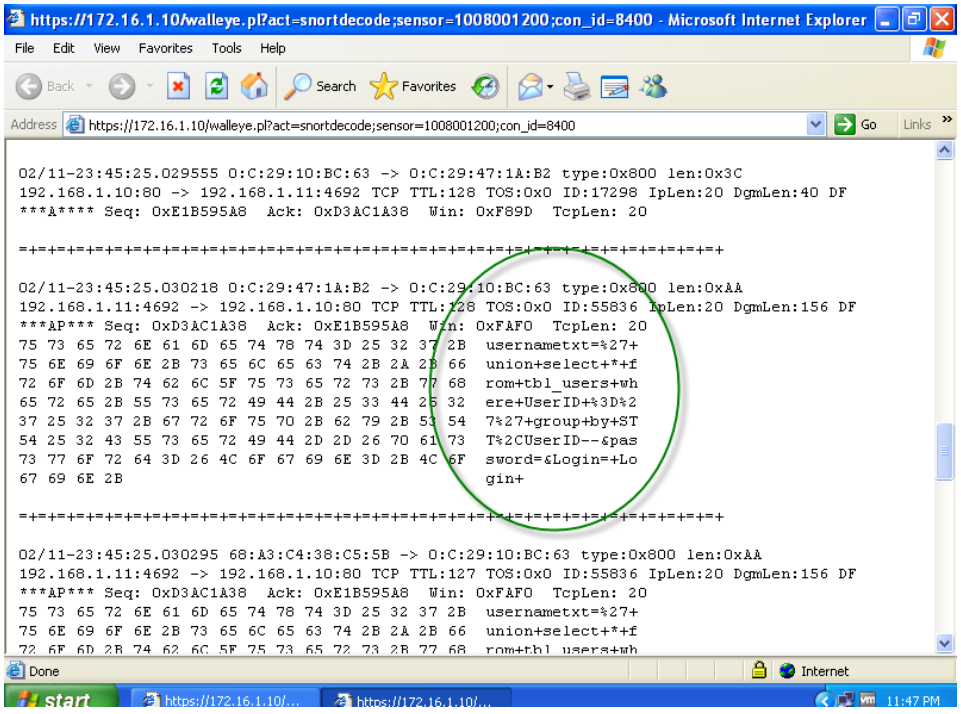
Hình 4.4.2c - Nội dung đoạn mã dò bảng trong cơ sở dữ liệu



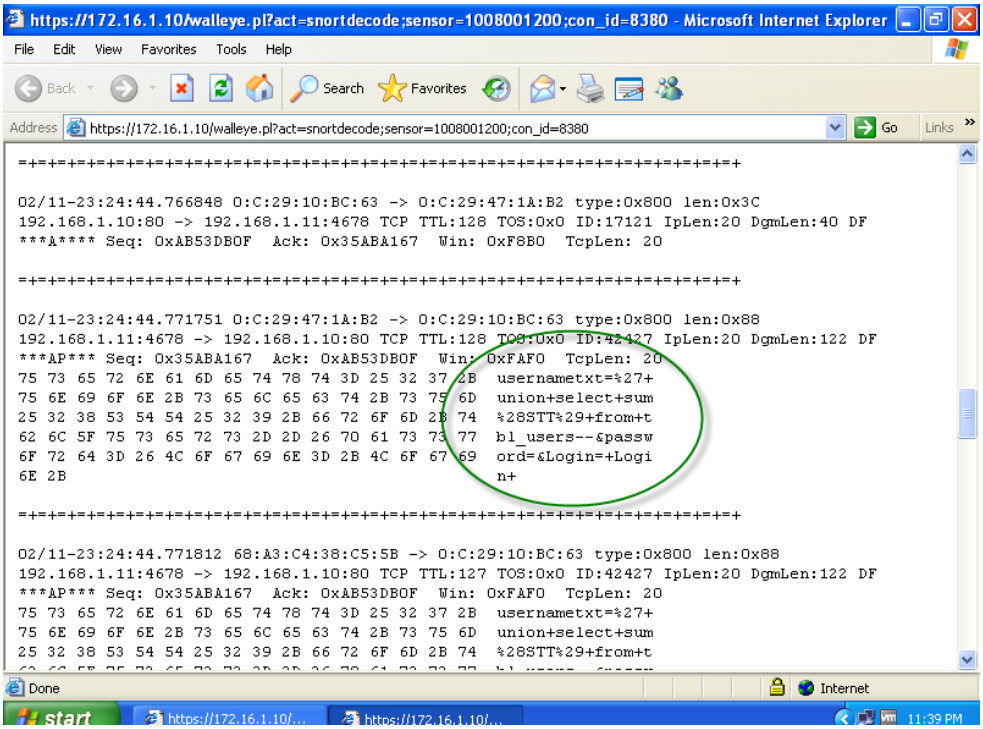
Hình 4.4.2d - Nội dung đoạn mã hacker đang dò cột đầu tiên trong bảng tbl_users



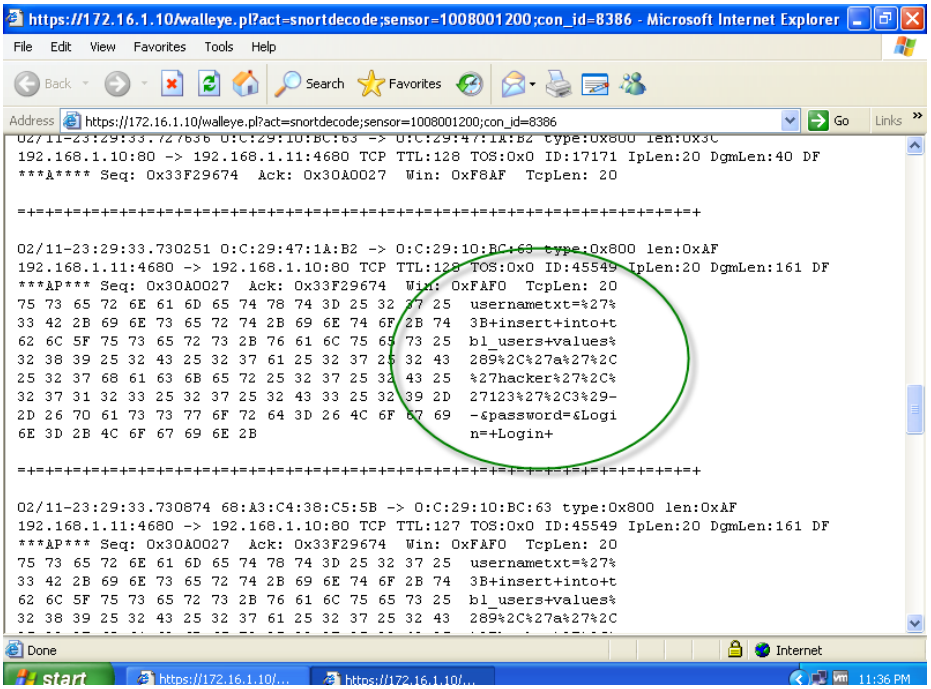
Hình 4.4.2e - Hacker đang dò tiếp cột thứ 2 trong bảng tbl_users



Hình 4.4.2f - Hacker dò các cột còn lại trong bảng tbl_users



Hình 4.4.2g - Hacker kiểm tra kiểu dữ liệu của cột STT



Hình 4.4.2h - Hacker đang thêm tài khoản vào cơ sở dữ liệu

Kết luận: trong chương cuối của đề tài này đã trình bày chi tiết quá trình triển khai, cài đặt, cấu hình và kiểm tra hệ thống bằng việc tấn công dựa vào kịch bản có sẵn. Nhờ vào hệ thống Honeynet chúng ta đã thấy được toàn bộ quá trình tấn công của hacker từ việc khai thác lỗi SQL Injection để chiếm quyền nhằm sửa đổi nội dung, làm mất tính toàn vẹn của trang web.

TÀI LIỆU THAM KHẢO

TRANG WEB

[1] http://www.bkacad.com

[2] http://www.kmasecurity.net

[3] http://www.quantrimang.com.vn

[4] http://www.kmasecurity.blogspot.com

[5] http://manthang.wordpress.com

[6] http://www.honeypots.net

[7] http://stankiewicz.free.fr/Wikka/wikka.php?wakka=HowtoHoneypot

[8] http://staff.washington.edu/dittrich/pnw-honeynet/honeywall-hw-guide/

[9] http://www.honeynet.pk/honeywall/index.htm

[10] http://old.honeynet.org/tools/index.html

[11] http://www.honeynet.org.es/papers/

[12] http://www.linuxvoodoo.com/resources/security/vmware

[13] http://www.scribd.com/doc/53535828/18/Virtual-honeynet

[14] http://www.scis.ulster.ac.uk/~kevin/top_hony.html

[15] http://honeynet.org.es/papers/vhwall/