**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
Khoa Mạng máy tính và Truyền thông**

**🙢 🕮 🙠**

**Bài Tập Nhóm**

**Môn:**

**AN TOÀN MẠNG MÁY TÍNH**

**Giáo viên hướng dẫn: Tô Nguyễn Nhật Quang**

**Nhóm sinh viên thực hiện :**

***Trần Hữu Tiến 13521072***

***Nguyễn Đặng Thắng***

***Ngô Tuấn Cường***

**Thành phố Hồ Chí Minh - tháng 5/2015**

# Lời cảm ơn

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô, gia đình và bạn bè.

Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý Thầy Cô ở Khoa Khoa Mạng Máy Tính và Truyền Thông – Trường Đại Học Công Nghệ Thông Tin đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Và đặc biệt, trong học kỳ này, Khoa đã tổ chức cho chúng em được tiếp cận với môn học mà theo em là rất hữu ích đối với sinh viên ngành Mạng Máy Tính cũng như tất cả các sinh viên thuộc các chuyên ngành Khoa Mạng Máy Tính và Truyền Thông khác. Đó là môn học “An Toàn Mạng Máy Tính”.

Em xin chân thành cảm ơn đã tận tâm Tô Nguyễn Nhật Quang đã hướng dẫn chúng em qua từng buổi học trên lớp cũng như những buổi nói chuyện, thảo luận về lĩnh vực sáng tạo trong nghiên cứu khoa học. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì em nghĩ bài thu hoạchnày của em rất khó có thể hoàn thiện được. Một lần nữa, em xin chân thành cảm ơn thầy.

Bài thu hoạch được thực hiện trong khoảng thời gian gần 8 tuần. Bước đầu đi vào thực tế, tìm hiểu về lĩnh vực sáng tạo trong nghiên cứu khoa học, kiến thức của em còn hạn chế và còn nhiều bỡ ngỡ. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô và các bạn học cùng lớp để kiến thức của em trong lĩnh vực này được hoàn thiện hơn.

MỤC LỤC

[Lời cảm ơn 1](#_Toc436958848)

[1 Hợp Đồng Nhóm 5](#_Toc436958849)

[1.1 Thành viên 5](#_Toc436958850)

[1.2 Mục tiêu đề ra 5](#_Toc436958851)

[1.3 Quy tắc 5](#_Toc436958852)

[1.4 Bảng phân công công việc 6](#_Toc436958853)

[1.5 Đánh giá 7](#_Toc436958854)

[1.6 Cam kết 7](#_Toc436958855)

[Dựa trên hợp đồng nhóm đã đưa ra, chúng tôi cam kết sẽ hoàn thành đúng yêu cầu trên. 7](#_Toc436958856)

[Hồ Chí Minh, Ngày…Tháng…Năm… 7](#_Toc436958857)

[Thành viên 1 Thành viên 2 Thành viên 3 Thành viên 4 7](#_Toc436958858)

[( kí tên ) ( kí tên ) ( kí tên ) ( kí tên ) 7](#_Toc436958859)

[2 Chương 1 8](#_Toc436958860)

[2.1 Firewall 8](#_Toc436958861)

[**2.1.1** **Khái niệm** 8](#_Toc436958862)

[**2.1.2** **Lịch sử** 8](#_Toc436958863)

[**2.1.3** **Lý do sử dụng Firewall** 9](#_Toc436958864)

[**2.1.4** **Các loại Fireware** 9](#_Toc436958865)

[**2.1.5** **Nhiệm vụ.** 10](#_Toc436958866)

[**2.1.6** **Chức năng chính của Firewall** 10](#_Toc436958867)

[**2.1.7** **Firewall bảo vệ và chống lại cái gì?** 10](#_Toc436958868)

[**2.1.8** **Các thành phần cơ bản của Firewall và cơ chế hoạt động** 11](#_Toc436958869)

[**2.1.9** **Thuận lợi và hạn chế của Firewall** 13](#_Toc436958870)

[**2.1.10** **Kết luận** 14](#_Toc436958871)

[2.2 Các thuật toán mã hóa 15](#_Toc436958872)

[**2.2.1** **Phương thức mã hóa RSA** 15](#_Toc436958873)

[Là phương thức mã hóa công khai. Do Ron Rivest, Adi Shamir và Len Adleman tại học viên MIT đề xuất năm 1977. Và hiện nay đang được sử dụng rộng rãi. 15](#_Toc436958874)

[**2.2.2** **Phương thức mã hóa DES** 17](#_Toc436958875)

[**2.2.3** **Thuật toán mã hóa AES** 19](#_Toc436958876)

[2.3 VNP (virtual private network) 21](#_Toc436958877)

[**2.3.1** **Đặt vấn đề** 21](#_Toc436958878)

[**2.3.2** **VPN giới thiệu** 21](#_Toc436958879)

[**2.3.3** **Tính bảo mật của VNP** 23](#_Toc436958880)

[**2.3.4** **Các kĩ thuật sử dụng trong VNP** 24](#_Toc436958881)

[**2.3.5** **Kết luận** 25](#_Toc436958882)

[3 Một số phương pháp tấn công và cách phòng tránh 25](#_Toc436958883)

[3.1 Sniff (Nghe lén) 25](#_Toc436958884)

[**3.1.1** **Khái niệm** 25](#_Toc436958885)

[**3.1.2** **Sơ lược về cách tấn công** 26](#_Toc436958886)

[**3.1.3** **Các loại sniff và cơ chế hoạt động** 26](#_Toc436958887)

[**3.1.4** **Cách phát hiện và phòng chống** 27](#_Toc436958888)

[**3.1.5** **Mức độ ảnh hưởng đến hệ thống** 29](#_Toc436958889)

[3.2 Phishing 29](#_Toc436958890)

[**3.2.1** **Khái niệm** 29](#_Toc436958891)

[**3.2.2** **Thời gian xuất hiện** 29](#_Toc436958892)

[**3.2.3** **Sơ lược cách tấn công** 30](#_Toc436958893)

[**3.2.4** **Cách phòng chống** 30](#_Toc436958894)

[**3.2.5** **Mức độ ảnh hưởng** 31](#_Toc436958895)

[3.3 SQL Injection 32](#_Toc436958896)

[**3.3.1** **Khái niệm** 32](#_Toc436958897)

[**3.3.2** **Thời gian xuất hiện.** 32](#_Toc436958898)

[**3.3.3** **Cách thức tấn công** 32](#_Toc436958899)

[**3.3.4** **Kỹ thuật an ninh dùng để phòng chống SQL Injection** 36](#_Toc436958900)

[**3.3.5** **Mức độ ảnh hưởng** 37](#_Toc436958901)

[3.4 Tấn công từ chối dịch vị (DDoS) 38](#_Toc436958902)

[**3.4.1** **Khái niệm** 38](#_Toc436958903)

[**3.4.2** **Thời gian xuất hiện** 38](#_Toc436958904)

[**3.4.3** **Các cách tấn công** 38](#_Toc436958905)

[**3.4.4** **Cách thức phòng chống:** 41](#_Toc436958906)

[**3.4.5** **Mức độ ảnh hưởng:** 41](#_Toc436958907)

**LỜI NÓI ĐẦU**

Sự phát triển của mạng máy tính trong thập niên vừa qua rất nhanh chóng, từ những nhóm người dùng chung một mạng để chia sẽ tập tin, trao đổi thông tin một cách dễ dàng hơn cho đến những hệ thống mạng của một doanh nghiệp hay một tập đoàn lớn nhằm tăng hiệu quả về thời gian và kinh tế, tất cả đều nhờ đến mạng máy tính. Chính vì sự phổ biến và quan trọng đó đã có không ít kẻ xấu nhằm lợi dụng lỗ hổng bảo mật của các hệ thống mạng máy tính đó để nhằm xâm nhập hệ thống với quyền quản trị để lấy đi những thông tin quan trọng nhằm mục đích xấu. Đã có nhiều cách tấn công vào mạng, hệ thống khác nhau, được ra đời ở nhiều nơi trên thế giới, những cuộc tấn công như vậy thường gây thiệt hại về mặt kinh tế rất lớn, và tất nhiên những chuyên gia về mạng hệ thống củng đã tìm được những cách khắc phục để hạn chế tối đa tổn thất đó. Trên thực tế đó, hôm nay, nhóm chúng tôi xin trình bày những bài tập của giáo viên bộ môn đưa ra và dần dần áp dụng vào thực tế. Trong đề tài mà chúng tôi thự hiện vẫn còn một số thiếu sót mong mọi người bỏ qua cho. Xin cảm ơn!

# Hợp Đồng Nhóm

## Thành viên

|  |  |
| --- | --- |
| **Họ và Tên** | **MSSV** |
| Trần Hữu Tiến | 13521072 |
| Nguyễn Đặng Thắng |  |
| Ngô Tuấn Cường |  |

## Mục tiêu đề ra

* Hoàn thành tốt những bài tập được thầy giao.
* Giúp đỡ nhau trong công việc.
* Rèn luyện khả năng làm việc nhóm.
* Hoàn thành học phần một cách xuất sắc nhất.
* Chia sẻ những kinh nghiệm quý giá trong quá trình tìm hiểu của mình cho các bạn trong nhóm.

## Quy tắc

* Làm việc trên cơ sở đồng thuận về ý kiến quan điểm chung.
* Tôn trọng ý kiến cá nhân của các thành viên nếu nó vẫn đảm bảo yêu cầu của công việc.
* Hoàn thành công việc đúng thời hạn.
* Giúp đỡ lẫn nhau khi có khó khăn.
* Ai không hoàn thành thì sẽ phải chịu trách nhiệm về công việc của mình.

## Bảng phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã số sinh viên** | **Thành viên** | **Vai trò** | **Công việc** |
| 13521072 | Trần Hữu Tiến | Member | * Giới thiệu một số phương pháp tấn công và cách phòng tránh. * Thiết kế đồ án. * Đăng kí và nộp đồ án. |
|  | Nguyễn Đặng Thắng | Member |  |
|  | Ngô Tuấn Cường | Member |  |

## Đánh giá

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MSSV** | **Họ Tên** | **Tỷ lệ công việc được giao (%)** | **Tỷ lệ công việc hoàn thành (%)** | **Nhận xét**  **Trọng số điểm** |
| 13521072 | Trần Hữu Tiến | 33% | 95% | Tích cực trong công việc, hoành thành đúng thời hạn. 4.5 điểm |
|  | Nguyễn Đặng Thắng | 33% |  |  |
|  | Ngô Tuấn Cường | 33% |  |  |

Bài 1: TỔng quan về an ninh mạng

## Câu 1

### Mô tả cấu trúc của một gói tin TCP và giải thích các chức năng của TCP header.

**Một gói tin TCP gồm 2 phần:**

* Header.
* Dữ liệu.

Chức năng của TCP header:

* **Cấu trúc:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **+** | **Bít 0 – 3** | **4 – 9** | **10 - 15** | **16 - 31** |
| **0** | Source Port | | | Destination Port |
| **32** | Sequence Number | | | |
| **64** | Acknowledgement Number | | | |
| **96** | Data Offset | Reserved | Flags | Window |
| **128** | Checksum | | | Urgent Pointer |
| **160** | Options (optional) | | | |
| **160/192+** | Data | | | |

* **Giải thích:**
  + **Source port:** Số hiệu của cổng tại máy tính gửi.
  + **Destination port:** Số hiệu của cổng tại máy tính nhận.
  + **Sequence number:** Trường này có 2 nhiệm vụ. Nếu cờ SYN bật thì nó là số thứ tự gói ban đầu và byte đầu tiên được gửi có số thứ tự này cộng thêm 1. Nếu không có cờ SYN thì đây là số thứ tự của byte đầu tiên.
  + **Acknowledgement number:** Nếu cờ ACK bật thì giá trị của trường chính là số thứ tự gói tin tiếp theo mà bên nhận cần.
  + **Data offset:** Trường có độ dài 4 bít quy định độ dài của phần header (tính theo đơn vị từ 32 bít). Phần header có độ dài tối thiểu là 5 từ (160 bit) và tối đa là 15 từ (480 bít).
  + **Reserved:** Dành cho tương lai và có giá trị là 0.
  + **Flags (hay Control bits):** Bao gồm 6 cờ:
    - **URG:** Cờ cho trường Urgent pointer.
    - **ACK:** Cờ cho trường Acknowledgement.
    - **PSH:** Hàm Push.
    - **RST:** Thiết lập lại đường truyền.
    - **SYN:** Đồng bộ lại số thứ tự.
    - **FIN:** Không gửi thêm số liệu.
* **Window:** Số byte có thể nhận bắt đầu từ giá trị của trường báo nhận (ACK).
* **Checksum:** 16 bít kiểm tra cho cả phần header và dữ liệu. Phương pháp sử dụng được mô tả trong [RFC 793](http://tools.ietf.org/html/793).
* **Urgent pointer:** Nếu cờ URG bật thì giá trị trường này chính là số từ 16 bít mà số thứ tự gói tin (*sequence number*) cần dịch trái.

# Một số phương pháp tấn công và cách phòng tránh

## Sniff (Nghe lén)

### **Khái niệm**

Sniffer là một chương trình nghe trộm gói tin (*packet sniffer*) (còn gọi là chương trình phân tích mạng, chương trình phân tích giao thức hay chương trình nghe trộm Ethernet) là một phần mềm có khả năng chặn và ghi lại giao thông dữ liệu qua một mạng viễn thông số hoặc một phần của một mạng dựa trên những đặc điểm của giao thức TCP/IP.

Sniffer ban đầu được sử dụng để các nhà quản trị mạng theo dõi và bảo trì hệ thống mạng của mình. Tuy nhiên sau này các hacker lại sử dụng nó như một công cụ với mục đích đánh hơi, nghe lén các thông tin trang mạng.

### **Sơ lược về cách tấn công**

Khi các dòng dữ liệu di chuyển qua lại một mạng, chương trình nghe trộm bắt lấy từng gói tin rồi giải mã và phân tích nội dung của nó theo RFC hoặc các đặc tả thích hợp khác. Tùy theo cấu trúc mạng người ta có thể nghe trộm tất cả hoặc chỉ một phần của giao thông dữ liệu từ một máy trong mạng. Đối với các mục đích theo dõi mạng (*network monitoring*), người ta có thể theo dõi tất cả các gói tin trong một mạng LAN bằng cách sử dụng một thiết bị chuyển mạch với một cái gọi là cổng theo dõi (nó lặp lại tất cả các gói tin đi qua tất cả các cổng của thiết bị chuyển mạch).

Những điều kiện để Sniff xảy ra:

* Sniff có thể hoạt động trong mạng LAN, mạng WAN, mạng WLAN.
* Điều kiện cần chỉ là dùng cùng Subnet Mark với Sniffer.
* Ngoài ra ta còn cần một công cụ để bắt và phân tích gói tin như: Cain&Abel, Ettercap, HTTP sniffer.

### **Các loại sniff và cơ chế hoạt động**

* Active:
* Môi trường: chủ yếu hoạt động trong môi trường có các thiết bị chuyển mạch gói. Phổ biến hiện nay là các dạng mạch sử dụng switch.
* Cơ chế hoạt động: Chủ yếu hiện nay thường dùng cơ chế ARP và RARP

(2 cơ chế chuyển đổi từ IP sang MAC và từ MAC sang IP) bằng cách phát đi các gói tin đầu độc, mà cụ thể ở đây là phát đi các gói thông báo cho máy gởi gói tin là “tôi là người nhận” mặc dù không phải là “người nhận”.

* Đặc điểm: do phải gởi gói tin đi nên có thể chiếm băng thông mạng. Nếu sniff quá nhiều máy trong mạng thì lượng gói gởi đi sẽ rất lớn (do liên tục gởi đi các gói tin giả mạo) có thể dẫn đến nghẽn mạng hay gây quá tải trên chính NIC của máy đang dùng sniff (thắt nút cổ chai).
* Hacker sử dụng các loại tấn công:
* MAC flooding:. Hacker sẽ gởi những gói ARP reply giả tạo với số lượng khổng lồ nhằm làm tràn bộ nhớ switch từ đó switch sẽ chạy chế độ forwarding mà không chuyển mạch gói.
* Giả MAC: Các sniffer sẽ thay đổi MAC của mình thành MAC của một máy hợp lệ và qua được chức năng lọc MAC của thiết bị.
* Đầu độc DHCP: Hacker giả mạo địa chỉ IP của mình là địa chỉ của gateway, từ đó mọi host trong mạng đều gửi gói tin đến gateway nhưng thực chất là đến máy của hacker.
* Passive:
* Môi trường: chủ yếu hoạt động trong môi trường không có các thiết bị chuyển mạch gói. Phổ biến hiện nay là các dạng mạng sử dụng hub, hay các mạng không dây.
* Cơ chế hoạt động: do không có các thiết bị chuyển mạch gói nên các host phải bị broadcast các gói tin đi trong mạng từ đó có thể bắt gói tin lại xem (dù host nhận gói tin không phải là nơi đến của gói tin đó).
* Đặc điểm: do các máy tự broadcast các gói nên hình thức sniff này rất khó phát hiện.

### **Cách phát hiện và phòng chống**

#### **Đối với active sniff**

1. Cách phát hiện

* Dựa vào quá trình đầu độc arp của sniffer để phát hiện.
* Vì phải đầu độc ARP nên sniffer sẽ liên tục gởi các gói tin đầu độc tới các victim. Do đó, ta có thể dùng một số công cụ bắt gói trong mạng để có thể phát hiện.
* Một cách khác ta có thể kiểm tra bảng ARP của host. Nếu ta thấy trong bảng ARP này có hai MAC giống nhau thì lúc này có khả năng mạng đang bị sniffer.
* Dựa trên băng thông:
* Do quá trình gởi gói tin đầu độc của sniffer nên quá trình này có thể chiếm băng thông, từ đây ta có thể dùng một số công cụ kiểm tra băng thông để phát hiện.
* Tuy nhiên cách này không hiệu quả và chính xác cũng không cao.
* Một số công cụ phát hiện sniff hay phát hiện đầu độc ARP:
* Xarp
* Arpwatch
* Symantec EndPoint.

1. Cách phòng chống

* Công cụ kiểm tra băng thông: Như đã nêu trên các sniffer có thể gây nghẽn mạng do đó có thể dùng các công cụ kiểm tra băng thông. Tuy nhiên, cách làm này không hiệu quả.
* Công cụ bắt gói tin: Các sniffer phải đầu độc ARP nên sẽ gởi arp đi liên tục, nếu dùng các công cụ này ta có thể thấy được ai đang sniff trong mạng. Cách này tương đối hiệu quả hơn, nhưng có một vài công cụ sniff có thể giả IP và MAC để đánh lừa.
* Thiết bị: Đối với thiết bị ta có thể dùng các loại có chức năng lọc MAC để phòng chống. Riêng với switch có thể dùng thêm chức năng VLAN trunking, có thể kết hợp thêm chức năng port security (tương đối hiệu quả do dùng VLAN và kết hợp thêm các chức năng bảo mật).
* Cách khác: Ngoài ra ta có thể cấu hình SSL, tuy hiệu quả, nhưng chưa cao vẫn có khả năng bị lấy thông tin.

#### **Đối với passive sniff**

1. Cách phát hiện

* Khó có khả năng phát hiện, vì bất kỳ host nào trong mạng cũng có thể bắt được gói tin.
* Tuy nhiên, dạng mạng để loại sniff này hoạt động chủ yếu dạng mạng thường dùng trong gia đình rất ít sử dụng cho doanh nghiệp.
* Tuy nhiên, hiện nay các doanh nghiệp thường dùng mạng không dây cho các máy tính xách tay thì có thể sử dụng thêm các tính năng lọc MAC của thiết bị, hay có thể xác thực bằng tài khoản, mật khẩu hay khóa truy cập.

1. Cách phòng chống

Thay thế các hub bằng các switch, lúc này các gói tin sẽ không còn broadcast đi nữa , nhưng lúc này ta lại đứng trước nguy cơ bị sniff dạng active.

### **Mức độ ảnh hưởng đến hệ thống**

Sniff là hình thức nghe lén thông tin trên mạng nhằm khai thác hiệu quả hơn tài nguyên mạng, theo dõi thông tin bất hợp pháp. Tuy nhiên, sau này các hacker dùng sniff để lấy các thông tin nhạy cảm như password, thông tin cá nhân... Do đó, sniff cũng là một cách hack.

Sniff thường tác động đến các gói tin, ít tác động mạnh đến phần hệ thống nên sniff rất khó phát hiện. Do đó, tuy sniff hoạt động đơn giản nhưng rất hiệu quả.

Do gần như không trực tiếp tác động lên hệ thống mạng nên các hình thức sniff sau khi hoạt động thường ít để lại dấu vết hay hậu quả nghiêm trọng.

## Phishing

### **Khái niệm**

Tấn công giả mạo (thuật ngữ tiếng Anh: phishing, biến thể từ fishing, nghĩa là câu cá, có thể ảnh hưởng từ chữ phreaking, nghĩa là sử dụng điện thoại người khác không trả phí, ám chỉ việc "nhử" người dùng tiết lộ thông tin mật) là một hành vi giả mạo ác ý nhằm lấy được các thông tin nhạy cảm như tên người dùng, mật khẩu và các chi tiết thẻ tín dụng bằng cách giả dạng thành một chủ thể tin cậy trong một giao dịch điện tử .

Cách thực hiện chủ yếu là đánh lừa người dùng truy cập vào một web giả mô phỏng lại giao diện trang web đăng nhập (login page) của các website có thật. Sau đó, kẻ lừa đảo sẽ dẫn dụ nạn nhân (victim) điền các thông tin vào trang “dỏm” đó rồi truyền tải đến anh ta (thay vì đến server hợp pháp) để thực hiện hành vi đánh cắp thông tin bất hợp pháp mà người sử dụng không hay biết. Ngay cả khi có sử dụng chứng thực máy chủ, có khi vẫn phải cần vài kĩ năng phức tạp mới xác định được website là giả mạo.

### **Thời gian xuất hiện**

Một kĩ thuật tấn công giả mạo đã được mô tả chi tiết trong một bài báo khoa học và thuyết minh của “Nhóm Người dùng HP Toàn cầu” vào năm 1987. Tuy nhiên, thuật ngữ "phishing" được đề cập lần đầu tiên trên nhóm tin “Usenet” vào ngày 2 tháng 1 năm 1996, dù thuật ngữ này có thể đã xuất hiện từ trước đó trong bản in của tạp chí dành cho hacker “2600” .

### **Sơ lược cách tấn công**

Trước đây, hacker thường sử dụng trojan (gián điệp) đến máy nạn nhân nhằm ăn cắp thông tin như password. Sau này cách lừa đảo lấy thông tin như Phishing được sử dụng nhiều hơn. Phương pháp phổ biến nhất của phishing là sử dụng một số form được thiết kế để tạo một liên kết tới một địa chỉ email có vẻ như thuộc về một tổ chức đang tin cậy hoặc một tổ chức giả mạo. Đường dẫn URLs sai chính tả hoặc việc sử dụng tên miền phụ là thủ thuật phổ biến được sử dụng hiện nay bởi những hacker.

Phishing gồm 2 bước chính:

* Tìm cách lừa người dùng đăng nhập vào trang web đăng nhập giả. Cách làm chính là thông qua gửi một email có chưa URL đến trang web giả.
* Tạo một trang web giả giống với trang web thật.
* Không chỉ có vậy, hacker còn kết hợp nhiều xảo thuật khác như tạo những email (giả) cả địa chỉ lẫn nội dung sao cho có sức thu hút, mã hóa đường link (URL) trên thanh address bar, tạo IP server giả…

### **Cách phòng chống**

Phòng chống Phishing là không khó. Phishing chủ yếu đánh vào sự tin tưởng của người dùng khi thực hiện nhưng giao dịch online. Vì thế cách tốt nhất là người dùng phải cẩn thận khi nhận được yêu cầu cung cấp thông tin nhạy cảm.

Nếu nhấn vào liên kết ở email đó nó đưa bạn đến một trang đăng nhập (giả). Dù bên ngoài nó giống hệt trang thật, ngay cả địa chỉ hay thanh trạng thái nhìn cũng có vẻ thật. Nhưng nếu bạn xem kỹ liên kết trên thanh address bar thì bạn sẽ thấy ở phía sau chữ @ mới là địa chỉ thật. Bạn nên quan sát kĩ trước khi điền thông tin. Tốt hơn hết là xem mã nguồn (view source) của form thì rõ là form thông tin không phải truyền đến citibank mà là đến một nơi khác.

Với cách tiếp cận theo kiểu “biết cách tấn công để phòng thủ” trên, chúng ta sẽ thấy rõ hơn bản chất của một cuộc tấn công phishing – tấn công đơn giản, nhưng hiệu quả thì rất cao. Một khi bạn hiểu được cách thức tấn công thì chắc rằng bạn cũng sẽ có cách đối phó thích hợp.

* Tóm lại:
* Cẩn thận với những email lạ, đặc biệt là những email yêu cầu cung cấp thông tin dù vẫn biết là phải tránh nhưng không ít trường hợp đều chủ quan.
* Xem kỹ nội dung có chính xác, có giống với những biểu mẫu thường gặp không. Nếu sai chính tả như trên thì bạn nên xem xét lại.
* Nếu có yêu cầu xác nhận thì xem kỹ liên kết, nếu có ký tự là như @ hay %01 thì có khả năng giả mạo.
* Với các trang xác nhận thông tin quan trọng, họ luôn dùng giao thức http secure (có ‘s’ sau http) nên địa chỉ có dạng http**s**://.... chứ không phải là http:// thường. Ngân hàng kêu ta xác nhận lại dùng http:// “thường” thì chắc là ngân hàng… giả.
* Nên thường xuyên cập nhật các miếng vá lỗ hổng bảo mật cho trình duyệt (web browser). Cài thêm chương trình phòng chống virus, diệt worm, trojan.

### **Mức độ ảnh hưởng**

Phishing là hình thức gian lận, lừa người dùng cung cấp thông tin cá nhân đặc biệt là thông tin về tài khoản ngân hàng, giao dịch trực tuyến… Sau đó sử dụng những thông tin đó chiếm đoạt tài chính của nạn nhân.

Phishing tấn công đơn giản nhưng mang lại hiệu quả cao và còn có thể mở rộng tấn công cho nên tình trạng lừa đảo này ngày càng có dấu hiệu tăng lên về số lượng và quy mô .

Hằng năm trên thế giới xảy ra hàng loạt vụ lừa đảo, chiếm dụng tài khoản ngân hàng của người dùng, thiệt hại lên đến hàng triệu USD.

Hiện nay không chỉ có ngân hàng mà nó còn mở rộng sang nhiều lĩnh vực khác. Năm 2011 nhiều tên tuổi đình đám của thế giới như: Google, Sony, IMF cũng đã bị hacker tấn công chiếm đoạt thông tin.

## SQL Injection

### **Khái niệm**

SQL injection là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject (tiêm vào) và thi hành các câu lệnh SQL bất hợp pháp. SQL injection có thể cho phép những kẻ tấn công thực hiện các thao tác, delete, insert, update,… trên cơ sở dữ liệu của ứng dụng, thậm chí là server mà ứng dụng đó đang chạy, lỗi này thường xảy ra trên các ứng dụng web có dữ liệu được quản lý bằng các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, Oracle, DB2, Sysbase...

### **Thời gian xuất hiện.**

Ban đầu SQL Injection được công bố lần đầu tiên như một lưu ý trong một bài viết về phương pháp khai thác lỗ hổng dịch vụ Web của Microsoft. Bài viết này được xuất hiện đầu tiên tại số thứ 54 của tạp chí Phrack (giáng sinh năm 1998), lúc đó là một tạp chí thường kỳ về chủ đề Hacking. Số báo thứ 54 đó được biết tới như là điểm khởi đầu của loại hình SQL Injection.

### **Cách thức tấn công**

Có bốn dạng tấn công thường gặp bao gồm: vượt qua kiểm tra lúc đăng nhập, sử dụng câu lệnh SELECT, sử dụng câu lệnh INSERT, sử dụng các stored-procedures.

#### **Dạng tấn công vượt qua kiểm tra lúc 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. 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.

Ví dụ, trong trường hợp sử dụng ASP, người ta có thể dùng 2 trang : 1 trang HTML để hiển thị Form nhập liệu và 1 trang ASP để xử lý thông tin nhập vào từ phía người dùng như sau:

- Trang nhập liệu: *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>*

Trang xử lý nhập liệu*: 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 %>*

Chỗ sơ hở trong đoạn mã xử lý nhập liệu trên 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 tin tặc có thể điều khiển câu truy vấn sẽ được thực hiện. Ví dụ, nếu người dùng nhập chuỗi trong ngoặc 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.

#### **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 để dò tìm các điểm yếu khởi đầu cho việc tấn công. Ví dụ, trong các trang tìm kiếm. Các 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. Giả sử đoạn mã nhập vào là:

*' DROP TABLE T\_AUTHORS --*

Câu truy vấn sẽ thực hiện việc xóa bảng.

#### **Dạng tấn công sử dung 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 SQLi, 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*

#### **Dạng tấn công sử dụng store-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 ‘cmdd.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.

### **Kỹ thuật an ninh dùng để phòng chống SQL Injection**

Biết được về mức độ ảnh hưởng của SQL Injection là rất lớn nên các nhà phát triển web và các chuyên gia bảo mật không ngừng tìm ra những cách để chống lại lỗ hổng bảo mật này, cho đến hiện nay đã có rất nhiều cách thức củng như thủ thuật để chống lại, có thể kể đến một số cách sau đây:

#### **Kiểm soát chặt chẽ dữ liệu nhập vào**

*<%*

*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 hãy kiểm tra dữ liệu có đúng kiểu hay không bằng hàm Is Numeric().

Ngoài ra có thể xây dựng hàm loại bỏ một số kí tự và từ khóa nguy hiểm như: ;--, select, insert, xp\_, … ra khỏi chuỗi dữ liệu nhập từ phía người dùng để hạn chế các tấn công dạng này:

*<%*

*Function KillChars(sInput)*

*dim badChars*

*dim newChars*

*badChars = array("select", "drop", ";", "--", "insert", "delete", "xp\_")*

*newChars = strInput*

*for i = 0 to uBound(badChars)*

*newChars = replace(newChars, badChars(i), "")*

*next*

*KillChars = newChars*

*End Function*

*%>*

#### **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

#### **Sử dụng thủ thuật.**

Ngoài các cách trên, một số web master còn sử dụng một số thủ thuật để chống lại tấn công SQL Injection đơn giản và tạm thời như: Dùng hàm để loại bỏ các câu lệnh và ký tự không cần thiết để người tấn công không thể dung các câu lệnh để tấn công.

### **Mức độ ảnh hưởng**

Cho đến nay, có thể nói rằng SQL Injection như là một trong những lỗ hổng có mức tàn phá ghê gớm nhất tới hoạt động kinh doanh của các doanh nghiệp, tổ chức, làm thất thoát các thông tin mang tính nhạy cảm được lưu trong cơ sở dữ liệu của ứng dụng như là Tên người dùng, mật khẩu, địa chỉ, địa chỉ thư điện tử, số điện thoại, thông tin thẻ tín dụng…

## Tấn công từ chối dịch vị (DDoS)

### **Khái niệm**

Tấn công từ chối dịch vụ là tên gọi chung của cách tấn công làm cho một hệ thống nào đó bị quá tải không thể cung cấp dịch vụ, làm gián đoạn hoạt động của hệ thống hoặc hệ thống phải ngưng hoạt động**.**

### **Thời gian xuất hiện**

Vào năm 1998 Chương trình Trinoo Distributed Denial of Service (DDoS) được viết bởi Phifli đươc coi như mở đầu cho hình thức tấn công từ chối dịch vụ.

### **Các cách tấn công**

#### **SYN Attack**

Được xem là một trong những kiểu tấn công DoS kinh điển nhất. Lợi dụng sơ hở của thủ tục TCP khi “bắt tay ba bước”, mỗi khi client muốn thực hiện kết nối với server thì nó thực hiện việc bắt tay ba bước thông qua các gói tin (packet):

* Bước 1: client sẽ gửi gói tin (packet chứa SYN=1) đến máy chủ để yêu cầu kết nối.
* Bước 2: khi nhận được gói tin này, server gửi lại gói tin SYN/ACK để thông báo cho client biết là nó đã nhận được yêu cầu kết nối và chuẩn bị tài nguyên cho việc yêu cầu này. Server sẽ dành một phần tài nguyên để nhận và truyền dữ liệu. Ngoài ra, các thông tin khác của client như địa chỉ IP và cổng (port) cũng được ghi nhận.
* Bước 3: cuối cùng client hoàn tất việc bắt tay ba bước bằng cách hồi âm lại gói tin chứa ACK cho server và tiến hành kết nối.

Do TCP là thủ tục tin cậy trong việc giao nhận nên trong lần bắt tay thứ hai, server gửi gói tin SYN/ACK trả lời lại client mà không nhận lại được hồi âm của client để thực hiện kết nối thì nó vẫn bảo lưu nguồn tài nguyên chuẩn bị kết nối đó và lặp lại việc gửi gói tin SYN/ACK cho client đến khi nhận được hồi đáp của client.

Điểm mấu chốt ở đây là làm cho client không hồi đáp cho Server, và có càng nhiều, càng nhiều client như thế trong khi server vẫn lặp lại việc gửi packet đó và giành tài nguyên để chờ trong lúc tài nguyên của hệ thống là có giới hạn.

Thường, để giả địa chỉ IP, các hacker hay dùng Raw Sockets (không phải gói tin TCP hay UDP) để giả mạo hay ghi đè giả lên IP gốc của gói tin. Khi một gói tin SYN với IP giả mạo được gửi đến server, nó cũng như bao gói tin khác, vẫn hợp lệ đối với server và server sẽ cấp vùng tài nguyên cho đường truyền này, đồng thời ghi nhận toàn bộ thông tin và gửi gói SYN/ACK ngược lại cho client.

Vì địa chỉ IP của client là giả mạo nên sẽ không có client nào nhận được SYN/ACK packet này để hồi đáp cho máy chủ. Sau một thời gian không nhận được gói tin ACK từ client, server nghĩ rằng gói tin bị thất lạc nên lại tiếp tục gửi tiếp SYN/ACK, cứ như thế, các kết nối tiếp tục mở.

#### Flood attack

Một kiểu tấn công DoS nữa cũng rất hay được dùng vì tính đơn giản của nó và vì có rất nhiều công cụ sẵn có hỗ trợ đắc lực cho kẻ tấn công là Flood Attack, chủ yếu thông qua các website.

Về nguyên tắc, các website đặt trên máy chủ khi chạy sẽ tiêu tốn một lượng tài nguyên nhất định của máy chủ. Dựa vào đặc điểm đó, những kẻ tấn công dùng các phần mềm như smurf chẳng hạn để liên tục yêu cầu máy chủ phục vụ trang web đó để chiếm dụng tài nguyên.

#### **Smurf attack**

Là thủ phạm sinh ra cực nhiều giao tiếp ICMP (ping) tới địa chỉ Broadcast của nhiều mạng với địa chỉ nguồn là mục tiêu cần tấn công. Chúng ta cần lưu ý là: Khi ping tới một địa chỉ là quá trình hai chiều – Khi máy A ping tới máy B máy B reply lại hoàn tất quá trình. Khi tôi ping tới địa chỉ Broadcast của mạng nào đó thì toàn bộ các máy tính trong mạng đó sẽ Reply lại tôi. Nhưng giờ tôi thay đổi địa chỉ nguồn, thay địa chỉ nguồn là máy C và tôi ping tới địa chỉ Broadcast của một mạng nào đó, thì toàn bộ các máy tính trong mạng đó sẽ reply lại vào máy C chứ không phải tôi và đó là tấn công Smurf. Kết quả đích tấn công sẽ phải chịu nhận một đợt Reply gói ICMP cực lớn và làm cho mạng bị dớt hoặc bị chậm lại không có khả năng đáp ứng các dịch vụ khác.

#### **Tấn công dịch vụ phân tán DDoS**

Để thực hiện thì kẻ tấn công tìm cách chiếm dụng và điều khiển nhiều máy tính/mạng máy tính trung gian (đóng vai trò zombie) từ nhiều nơi để đồng loạt gửi ào ạt các gói tin với số lượng rất lớn nhằm chiếm dụng tài nguyên và làm tràn ngập đường truyền của một mục tiêu xác định nào đó. Làm cho sever hoàn toàn tê liệt. Theo cách này thì dù băng thông có bao nhiêu đi chăng nữa thì cũng không thể chịu đựng được số lượng hàng triệu các gói tin đó nên hệ thống không thể hoạt động được nữa và như thế dẫn đến việc các yêu cầu hợp lệ khác không thể nào được đáp ứng, server sẽ bị “đá văng” khỏi internet.

#### **Tấn công từ chối dịch vụ phản xạ nhiều vùng DRDOS.**

Xuất hiện vào đầu năm 2002, là kiểu tấn công mới nhất, mạnh nhất trong họ DoS. Nếu được thực hiện bởi kẻ tấn công có tay nghề thì nó có thể hạ gục bất cứ hệ thống nào trên thế giới trong phút chốc. Mục tiêu chính của DRDoS là chiếm đoạt toàn bộ băng thông của máy chủ, tức là làm tắc nghẽn hoàn toàn đường kết nối từ máy chủ vào xương sống của Internet và tiêu hao tài nguyên máy chủ. Trong suốt quá trình máy chủ bị tấn công bằng DRDoS, không một máy khách nào có thể kết nối được vào máy chủ đó. Tất cả các dịch vụ chạy trên nền TCP/IP như DNS, HTTP, FTP, POP3,... đều bị vô hiệu hóa. Về cơ bản, DRDoS là sự phối hợp giữa hai kiểu DoS và DDoS. Nó có kiểu tấn công SYN với một máy tính đơn, vừa có sự kết hợp giữa nhiều máy tính để chiếm dụng băng thông như kiểu DDoS. Kẻ tấn công thực hiện bằng cách giả mạo địa chỉ của server mục tiêu rồi gửi yêu cầu SYN đến các server lớn như Yahoo, Micorosoft, chẳng hạn để các server này gửi các gói tin SYN/ACK đến server mục tiêu. Các server lớn, đường truyền mạnh đó đã vô tình đóng vai trò zoombies cho kẻ tấn công như trong DDoS.

Quá trình gửi cứ lặp lại liên tục với nhiều địa chỉ IP giả từ kẻ tấn công, với nhiều server lớn tham gia nên server mục tiêu nhanh chóng bị quá tải, bandwidth bị chiếm dụng bởi server lớn. Tính “nghệ thuật” là ở chỗ chỉ cần với một máy tính với modem 56kbps, một hacker lành nghề có thể đánh bại bất cứ máy chủ nào trong giây lát mà không cần chiếm đoạt bất cứ máy nào để làm phương tiện thực hiện tấn công.

### **Cách thức phòng chống:**

* + Tăng cường khả năng xử lý của hệ thống:
* Tối ưu hóa các thuật toán xử lý, mã nguồn của máy chủ web.
* Nâng cấp hệ thống máy chủ.
* Nâng cấp đường truyền và các thiết bị liên quan.
* Cài đặt đầy đủ các bản vá cho hệ điều hành và các phần mềm khác để phòng ngừa khả năng bị lỗi tràn bộ đệm, cướp quyền điều khiển,v.v…
  + Sử dụng các tường lửa cho phép lọc nội dung thông tin (tầng ứng dụng) để ngăn chặn các kết nối nhằm tấn công hệ thống.
  + Phân tích luồng tin (traffic) để phát hiện các dấu hiệu tấn công và cài đặt các tường lửa cho phép lọc nội dung thông tin (tầng ứng dụng) ngăn chặn theo các dấu hiệu đã phát hiện.
  + Ngoài ra ta củng có thể cụng một số công cụ để chống lại như:
* Sử dụng hệ thống thiết bị, phần mềm hoặc dịch vụ giám sát an toàn mạng (đặc biệt về lưu lượng) để phát hiện sớm các tấn công từ chối dịch vụ.
* Sử dụng thiết bị bảo vệ mạng có dịch vụ chống tấn công DDoS chuyên nghiệp kèm theo, ví dụ như: Arbor, Checkpoint, Imperva, Perimeter,…

### **Mức độ ảnh hưởng:**

Một cuộc tấn công mạng theo phương thức tấn công từ chối dịch vụ (DDoS) phổ biến hiện nay có thể làm tê liệt những hệ thống mạng doanh nghiệp, hay một ngân hàng gây thiệt hại hàng tỉ đồng và không chỉ thiệt hại về tài sản, khi một hệ thống bị tê liệt thì sẽ hệ lụy đến nhiều hoạt động hay giao dịch của hệ thống đó.

Từ khi ra đời kỹ thuật tấn công từ chối dịch vụ cho đến nay đã có rất nhiều các cuộc tấn công dựa trên phương thức này, và củng xảy xa rất nhiều cuộc tấn công đình đám, khiến các ông lớn như Google hay Facebook củng dính phải.

Nguồn tài liệu tham khảo:

1. [http://vi.wikipedia.org](http://vi.wikipedia.org/wiki/T%C6%B0%E1%BB%9Dng_l%E1%BB%ADa)
2. [http://www.quantrimang.com.vn](http://www.quantrimang.com.vn/firewall-cac-khai-niem-tinh-chat-nguyen-ly-va-su-dung-firewall-23)
3. [https://nguyentanvy.wordpress.com](https://nguyentanvy.wordpress.com/2011/11/28/cc-thnh-ph%E1%BA%A7n-c%E1%BB%A7a-firewall-v-c%C6%A1-ch%E1%BA%BF-ho%E1%BA%A1t-d%E1%BB%99ng/)
4. <http://congdd.wordpress.com>
5. <http://exploitx.org>