**TRƯỜNG ĐẠI HỌC AN GIANG  
KHOA CÔNG NGHỆ THÔNG TIN**

****

**TIỂU LUẬN**

**AN TOÀN HỆ THỐNG VÀ AN NINH MẠNG**

**CÁC KỸ THUẬT TẤN CÔNG WEB SERVER**

**HỌ TÊN SINH VIÊN**

DTH225650 - Huỳnh Quốc Huy

**HỌ TÊN GIẢNG VIÊN**

Lê Hoàng Anh

**AN GIANG, THÁNG 8 NĂM 2025**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN**

**AN TOÀN HỆ THỐNG VÀ AN NINH MẠNG**

**CÁC KỸ THUẬT TẤN CÔNG WEB SERVER**

**HỌ TÊN TÁC GIẢ**

DTH225650 - Huỳnh Quốc Huy

**HỌ TÊN GIẢNG VIÊN HƯỚNG DẪN**

Lê Hoàng Anh

**AN GIANG, THÁNG 8 NĂM 2025**

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

**Giảng viên hướng dẫn**

( Ký và ghi rõ họ tên)

Nội dung nhận xét:

* **Đồng ý** hay **Không đồng ý**  cho sinh viên báo cáo TTCK; Nếu không đồng ý cần ghi rõ lý do.
* Kết quả được so với yêu cầu;
* Ý kiến khác (nếu có)

**MỤC LỤC**

[**1.** **TỔNG QUAN VỀ AN TOÀN HỆ THỐNG** 2](#_Toc207641130)

[**1.1.** **Giới thiệu** 2](#_Toc207641131)

[**1.2.** **Các khái niệm cơ bản, các giai đoạn của một cuộc tấn công, phân loại hacker** 2](#_Toc207641132)

[**1.2.1.** **Các khái niệm cơ bản và liên quan đến web server** 2](#_Toc207641133)

[**1.2.2.** **Các giai đoạn của một cuộc tấn công** 3](#_Toc207641134)

[**1.2.3.** **Phân loại hacker:** 4](#_Toc207641135)

[**1.3.** **Vai trò của an toàn hệ thống** 4](#_Toc207641136)

[**2.** **CÁC KỸ THUẬT TẤN CÔNG WEB SERVER** 5](#_Toc207641137)

[**2.1.** **SQL Injection:** 6](#_Toc207641138)

[**2.1.1.** **Giới thiệu về SQL Injection** 6](#_Toc207641139)

[**2.1.2.** **Các loại Lỗi thường gặp dẫn đến SQL Injection** 6](#_Toc207641140)

[**2.1.3.** **Các dangj tấn công SQL Injection thông dụng.** 8](#_Toc207641141)

[**2.2.** **Cross-Site Scripting (XSS)** 9](#_Toc207641142)

[**2.3.** **Directory Traversal (Path Traversal)** 9](#_Toc207641143)

[**2.4.** **File Upload Vulnerability** 9](#_Toc207641144)

[**2.5.** **Denial of Service (DoS/ DDoS)** 9](#_Toc207641145)

[**2.6.** **Tấn công Brute-Force và Dictionary Attack** 9](#_Toc207641146)

[**3.** **THỰC HÀNH DEMO** 9](#_Toc207641147)

[**3.1.** **Mô tả** 9](#_Toc207641148)

[**3.2.** **Các công cụ sử dụng** 9](#_Toc207641149)

[**3.3.** **Hướng dẫn từng bước** 9](#_Toc207641150)

[**4.** **BIỆN PHÁP PHÒNG CHỐNG VÀ BẢO VỆ** 9](#_Toc207641151)

[**4.1.** **Phòng Chống Sql Injection Và Xss** 9](#_Toc207641152)

[**4.1.1.** **Prepared Statement** 9](#_Toc207641153)

[**4.1.2.** **Input Sanitation** 9](#_Toc207641154)

[**4.2.** **Bảo mật file upload** 9](#_Toc207641155)

[**4.3.** **Các biện pháp khác** 9](#_Toc207641156)

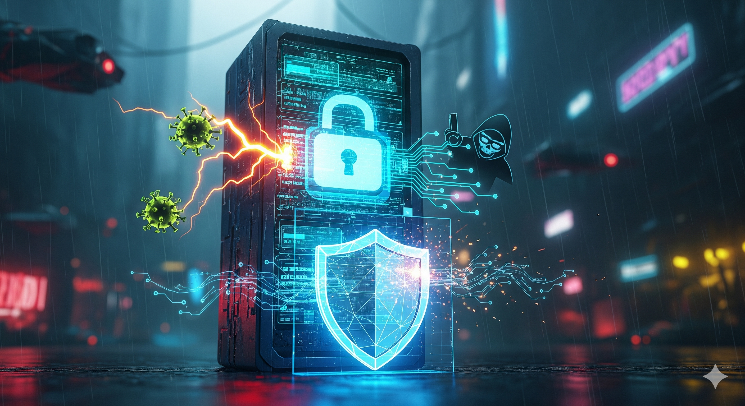
[**4.3.1.** **Firewall** 9](#_Toc207641157)

[**4.3.2.** **IDS/IPS** 9](#_Toc207641158)

[**5.** **REFERENCES** 9](#_Toc207641159)

1. **TỔNG QUAN VỀ AN TOÀN HỆ THỐNG**
   1. **Giới thiệu**

Trong kỷ nguyên số hóa của cuộc cách mạng công nghiệp 4.0, các hệ thống website, máy chủ,… đã trở thành nền tảng cốt lõi của mọi tổ chức và doanh nghiệp, hoạt động như một cầu nối quan trọng để cung cấp dịch vụ, chia sẻ thông tin và thực hiện các giao dịch trực tuyến.Với việc hoạt động không ngừng nghỉ "24/7", các máy chủ web (web server) trở thành mục tiêu hàng đầu của những kẻ tấn công mạng. Một cuộc tấn công thành công không chỉ gây ra những thiệt hại nặng nề về tài chính mà còn ảnh hưởng nghiêm trọng đến uy tín, làm mất lòng tin của khách hàng và đối tác ảnh hưởng đến thông tin cá nhân người dùng về (tài khoản mật khẩu, thông tin và bí mật kinh doanh của doanh nghiệp,…) và nhiều hậu quả nặng nề khác. Mục tiêu của các cuộc tấn công này có thể là đánh cắp thông tin nhạy cảm (như dữ liệu thẻ tín dụng, tài khoản ngân hàng), công bố thông tin mật, hoặc thay đổi giao diện trang web (defacing).



Hình 1.1

Tuy nhiên, bên cạnh những hacker mũ đen (Black Hat) với mục đích phá hoại, tồn tại một nhóm chuyên gia được gọi là hacker mũ trắng (White Hat), hay Certified Ethical Hacker (CEH). Đây là những cá nhân có kỹ năng tấn công tương tự nhưng sử dụng chúng cho mục đích thiện chí và hợp pháp. Họ thực hiện các cuộc tấn công thử nghiệm (penetration test) để tìm kiếm và đánh giá các lỗ hổng bảo mật, từ đó đưa ra các giải pháp khắc phục, biện pháp bảo vệ và phòng chống các tội phạm công nghệ cao.



Hình 1.2

Đồ án này được xây dựng với mục tiêu cung cấp một cái nhìn toàn diện về các mối đe dọa này. Chúng tôi không chỉ đi sâu vào việc phân tích các kỹ thuật tấn công phổ biến mà còn tập trung vào khía cạnh thực hành, nhằm giúp người học có thể tự xây dựng và phòng ngừa và bảo vệ hệ thống web server. Cụ thể hơn, đồ án sẽ phân tích chuyên sâu và làm rõ chi tiết các kỹ thuật tấn công web server phổ biến nhất hiện nay, bao gồm SQL Injection, Cross-Site Scripting (XXS), Insecure Direct Object References, Sensitive data exposure, và các lỗ hổng khác như Directory Traversal, File Upload Vulnerability… Dự trên cơ sở lý thuyết cũng như các cách thức tấn công web server này chúng tôi thiết lập một môi trường ảo mô phỏng các cuộc tấn công một cách an toàn và hợp pháp có tính tham khảo và nghiên cứu và chia sẻ. Từ đó giúp người học nắm vững các bước tấn công và cách thức khai thác hệ thống.

Qua các cơ sở đó, từ lý thuyết đến cách thức hoạt động, xâm nhập tấn công một hệ thống máy chủ web, đồ án đưa ra các biện pháp cũng như các chiến lược và phòng thủ hiệu quả, bao gồm kỹ thuật lập trình an toàn, cấu hình hệ thống đúng cách và sử dụng các công cụ bảo mật chuyên dụng như Firewalls và IDS/IPS

* 1. **Các khái niệm cơ bản, các giai đoạn của một cuộc tấn công, phân loại hacker**

Để hiểu về các kỹ thuật tấn công, trước tiên cần nắm vững các thuật ngữ nền tảng trong lĩnh vực an toàn thông tin và các giai đoạn của một cuộc tấn công điển hình.

* + 1. **Các khái niệm cơ bản và liên quan đến web server**
       1. **Các khái niệm về an toàn hệ thông và an ninh mạng**
  + **Threat**: là các mối đe dọa đối với sự an toàn của thông tin bao gồm hacker, virus, sự cố máy tính như hư hỏng phần cứng, lỗi phần mềm cho đến những nguyên nhân do thiên tai, hỏa hoạn cũng là các threat. [1]
  + **Vulnerability**: là những điểm yếu về bảo mật của hệ thống như thiếu các bản vá lỗi bảo mật, sử dụng chính sách mật khẩu yếu … đều là các điểm nhạy cảm có khả năng bị các threat khai thác gây mất an toàn thông tin. [1]
  + **Exploit:** là quá trình khai thác các điểm yếu bảo mật để đánh cắp thông tin, tiến trình này có thể được thực hiện bởi những tác nhân bên trong hay bên ngoài hệ thống[1].
  + **Remote exploit:** là quá trình khai thác các lổ hỗng bảo mật từ xa ở trên máy tính khác hay từ internet.[1]
  + **Local exploit**: là quá trình khai thác những điểm yếu bảo mật ngay trên hệ thống để tiến hành leo thang nâng quyền hạn của một tài khoản, hay bẻ khóa mật khẩu của ứng dụng.[1]
  + **Target of evaluation:** là những mục tiêu có khả năng chứa các lổ hỗng bảo mật có thể bị tấn công. Các mục tiêu nay có thể là một máy chủ, máy trạm,những ứng dụng hay các trang web.[1]
  + **Attack:** là thuật ngữ chỉ tiến trình tấn công vào mục tiêu.[1]

**Phân loại theo vị trí:** Tấn công có thể xuất phát từ bên trong (inside) bởi các nhân viên hoặc những người có liên quan đến tổ chức, hoặc từ bên ngoài (outside) bởi các hacker ngoại vi. Cả hai đều mang tính rủi ro cao.

* + - 1. **Một số khái niệm liên quan đến webserver**
  + **Webserver:** là một máy tính có kết nối internet được cài các chương trình đặc biệt nhằm nhiệm vụ xử lý các yêu cầu từ phía người dùng và phản hồi các yêu cầu đó. Máy chủ web có thể xử lý dữ liệu và cung cấp thông tin đến máy khách thông qua các máy tính cá nhân trên môi trường Internet qua giao thức HTTP, giao thức được thiết kế để gửi các tệp tin đến trình duyệt Web, và các giao thức khác.[2]

**Web Server:** là máy chủ web, một phần mềm hoặc máy tính được thiết kế để lưu trữ, xử lý và phục vụ các tệp tin và trang web cho các yêu cầu từ các máy tính khác qua mạng. Nó hoạt động như một cầu nối giữa máy chủ và máy khách, cho phép truy cập vào các trang web và các tệp tin thông qua giao thức truyền tải siêu văn bản (HTTP) hoặc các giao thức khác.[3]

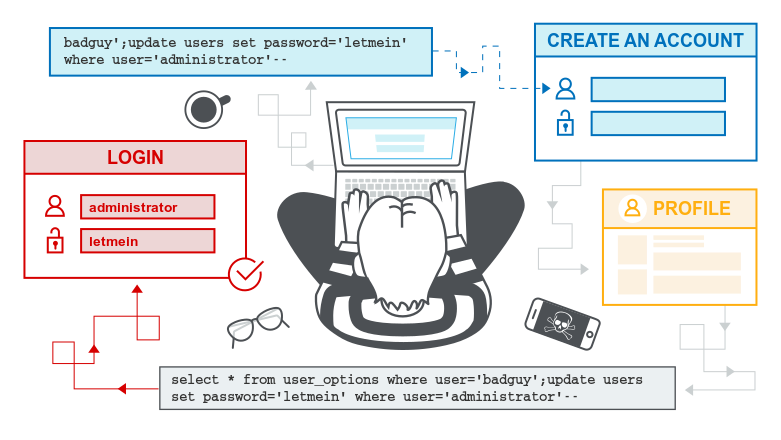
* + **Request:** có thể hiểu là thông tin gửi từ client (máy khách) lên server (máy chủ). Khi sử dụng trình duyệt truy cập một địa chỉ nào đó. Ví dụ khi truy cập địa chỉ google.com thì ngay lập tức trình duyệt sẽ dựa vào tên miền để gửi yêu cầu truy cập đến địa chỉ IP mà tên miền này đang trỏ tới, lúc này phía máy chủ sẽ phân tích yêu cầu và sẽ gửi luồng xử lý tới vị trí lưu trữ của mã nguồn PHP (hoặc mã nguồn bất kì) và nhiệm vụ của các mã nguồn là tiếp nhận yêu cầu, phân tích yêu cầu đó và trả kết quả lại cho máy khách là các trang web mà máy chủ Google tìm kiếm được.[2]
  + **Response :** là những dữ liệu mà máy chủ web trả về cho máy khách. Có thể là một chuỗi HTML hoặc các dữ liệu đa phương tiện như ảnh, video… Trình duyệt web của máy khách sẽ có nhiệm vụ xử lý và hiển thị các nội dung này thành giao diện trực quan cho người sử dụng.[2]
  + **Database:** là nơi lưu trữ các nội dung của một website, trên cơ sở dữ liệu có thể lưu thông tin về các tài khoản, mật khẩu, email … của người dùng. Hoặc thậm chí là các thông tin cực kỳ quan trọng như thông tin về thẻ thanh toán và hồ sơ y tế. Vậy nên đây là một mục tiêu phổ biến của Hacker.[2]
  + **Lỗ hổng bảo mật:** là những điểm yếu nằm trong thiết kế và cấu hình của hệ thống, lỗi của lập trình viên hoặc sơ suất trong quá trình vận hành. Cách thức hoạt động cơ bản của việc khai thác lỗ hổng bảo mật Web cơ bản: Hacker sử dụng các công cụ dò quét để phát hiện một loạt các website có cấu hình bảo mật kém hoặc website trên các nền tảng khác nhau như “WordPress” hay “Joomla” có các lỗ hổng đã được công bố nhưng chưa được xử lý. Từ đó Hacker sẽ lợi dụng để tấn công, cài đặt các mã độc và phá hoại các website nhằm trục lợi.[2]
    - 1. **Thuật ngữ liên quan đến Giao thức và Phiên làm việc**

Các cuộc tấn công không chỉ nhắm vào chính máy chủ mà còn vào cách giao tiếp giữa máy chủ và người dùng.

* + **Session Hijacking (Chiếm đoạt phiên làm việc): Sau** khi người dùng đăng nhập, web server sẽ tạo ra một phiên làm việc (session) và gán một mã định danh duy nhất (session ID). Kẻ tấn công sẽ đánh cắp mã này và giả mạo người dùng để truy cập vào tài khoản mà không cần mật khẩu.
  + **Man-in-the-Middle (MITM):** Kẻ tấn công chèn vào giữa hai thiết bị đang giao tiếp (ví dụ: trình duyệt và web server) để nghe lén, đọc, và thậm chí sửa đổi dữ liệu.
  + **DNS Spoofing/Cache Poisoning:** Kẻ tấn công làm sai lệch thông tin trong bộ nhớ đệm DNS của máy tính hoặc server, điều hướng người dùng truy cập một website giả mạo do chúng kiểm soát thay vì website thật.
    - 1. **Thuật ngữ liên quan đến mã độc và công cụ tấn công**
  + **Web Shell:** Một chương trình độc hại được tải lên web server thông qua lỗ hổng tải tệp (File Upload Vulnerability). Web Shell cho phép kẻ tấn công thực thi các lệnh từ xa trên server thông qua giao diện web, biến server thành một công cụ để tấn công các hệ thống khác.
  + **Botnet:** Mạng lưới các máy tính bị nhiễm phần mềm độc hại (bot), được kiểm soát từ xa bởi kẻ tấn công. Botnet thường được sử dụng để thực hiện các cuộc tấn công DDoS với quy mô lớn, làm quá tải máy chủ bằng cách gửi hàng triệu yêu cầu cùng lúc.
  + **Zero-Day Exploit:** Một lỗ hổng bảo mật chưa được công khai hoặc chưa được nhà sản xuất phần mềm vá lỗi. Đây là một dạng lỗ hổng cực kỳ nguy hiểm vì các biện pháp phòng chống thông thường không thể phát hiện và ngăn chặn được.
    - 1. **Phân loại lỗ hổng và cách tấn công Web Server**

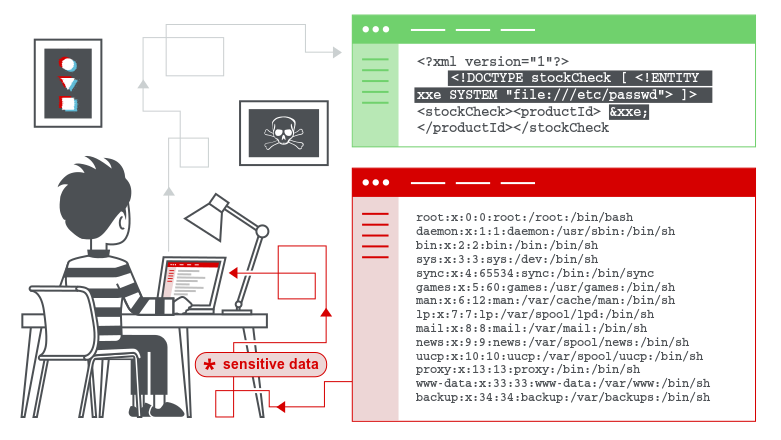
Những cuộc tấn công web server thường không chỉ dựa vào một lỗi duy nhất mà có thể là sự kết hợp của nhiều lỗ hổng khác nhau. Hiểu rõ các loại lỗ hổng và phương pháp tấn công sẽ giúp bạn có cái nhìn toàn diện hơn.

* + **SQL injection:** Cho phép kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào và các thông báo lỗi do hệ quản trị cơ sở dữ liệu trả về để inject (tiêm vào) và thực thi các câu lệnh SQL bất hợp pháp. SQL Injection cho phép xóa, chèn, cập nhật, v.v. trên cơ sở dữ liệu của website.[2]



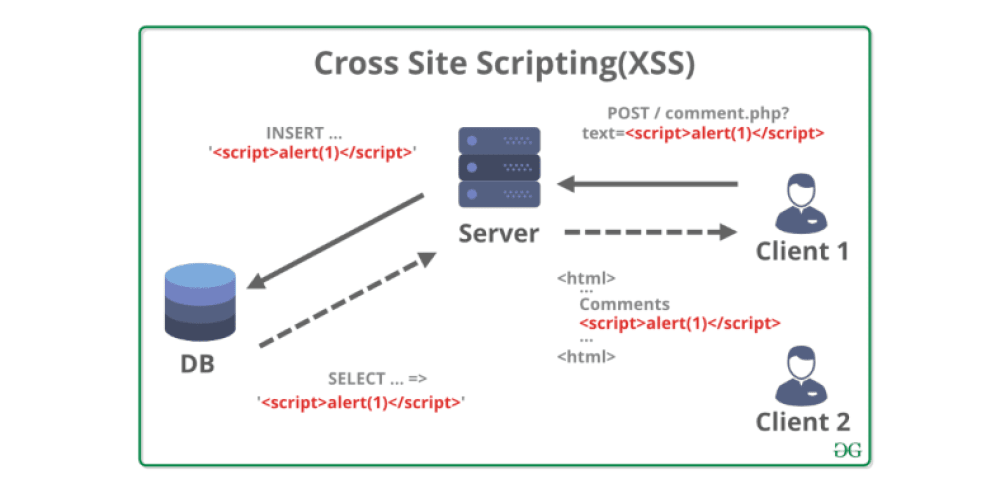
Hình 1.3

* + **XML External Entity Processing:** XML là một ngôn ngữ đánh dấu mở rộng, được ứng dụng rất rộng rãi. Nó sử dụng để trao đổi dữ liệu giữa các ứng dụng. Hiện nay có rất nhiều loại tài liệu sử dụng định dạng XML như rtf, pdf, tệp hình ảnh (svg) hay các file cấu hình.



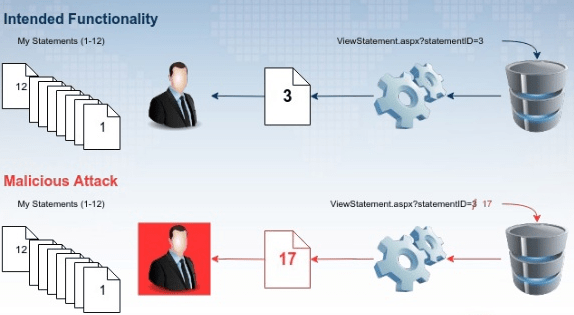
Hình 1.4

* + - Kỹ thuật tấn công này dựa vào việc cho phép khai báo External Entity (đối tượng mở rộng) trong phần DTD của dữ liệu XML, attacker có thể khai báo một entity để đọc nội dung của file bất kỳ trong hệ thống nếu trình phân tích XML (parser) được cấu hình không tốt.
    - Để tránh bị khai thác lỗ hổng này thì cần cấu hình trình phân tích XML (parser), không cho phép sử dụng khai báo External Entity (đối tượng mở rộng) trong phần định kiểu tài liệu DTD.[2]
  + **Cross Site Scripting (XSS):** Cho phép Hacker chèn những đoạn script độc hại (thường là Javascript hoặc HTML) vào website và thực thi trong trình duyệt của người dùng.



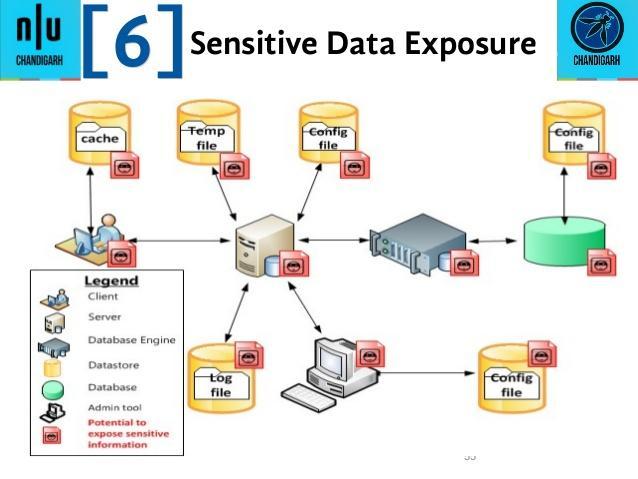
Hình 1.5

* + - Kẻ tấn công có thể dùng XSS để gửi những đoạn script độc hại tới một người dùng bất kỳ để lấy cookie, keylogging hoặc tiến hành lừa đảo.
    - Ngoài ra trong một số trường hợp đặc biệt, lỗ hổng XSS còn có thể xảy ra ở phía máy chủ web. Điều này thường gây ra hậu quả nghiêm trọng. Kẻ tấn công có thể đọc được các file nhạy cảm trên máy chủ.[2]
  + **Insecure Direct Object References:** Đây là trường hợp điển hình của việc cho rằng những gì người dùng nhập vào là tin cậy từ đó dẫn đến lỗ hổng bảo mật. Lỗ hổng này xảy ra khi chương trình cho phép người dùng truy cập các tài nguyên (dữ liệu, các tệp, cơ sở dữ liệu) mà không thực hiện quá trình kiểm soát quyền hạn (hoặc quá trình này không hoàn chỉnh) , dẫn đến kẻ tấn công có thể truy cập một cách bất hợp pháp vào các dữ liệu nhạy cảm, quan trọng trên máy chủ.
    - Một đoạn mã có module "download.php" và cho phép người dùng tải tệp xuống sử dụng tham số là tên file để tải xuống tệp từ máy chủ. Ví dụ “download.php?file=something.txt”. Do sai sót của nhà phát triển, việc kiểm tra quyền hạn đã bị bỏ qua. Kẻ tấn công có thể sử dụng lỗ hổng này để tải về bất kì tệp nào trên hệ thống mà ứng dụng có quyền truy cập. Chẳng hạn như mã nguồn ứng dụng, tệp nhạy cảm hoặc các dữ liệu khác trên máy chủ.



Hình 1.6

* + - Một ví dụ phổ biến khác là chức năng đặt lại mật khẩu dựa vào đầu vào của người dùng để xác định mật khẩu đặt lại. Sau khi nhấp vào URL hợp lệ, kẻ tấn công có thể sửa đổi trường tên người dùng trong URL để “đóng giả” người quản trị.
  + **Sensitive data exposure:** Lỗ hổng này thuộc về khía cạnh quản lý và mã hóa tài nguyên. Dữ liệu nhạy cảm phải được mã hóa mọi lúc, bao gồm cả khi gửi đi và khi lưu trữ – không được phép có ngoại lệ.
    - Thông tin thẻ tín dụng và mật khẩu người dùng không bao giờ được gửi đi hoặc được lưu trữ không mã hóa. Ngoài ra, các tiêu chuẩn an ninh web đề nghị sử dụng AES (256 bit trở lên) và RSA (2048 bit trở lên).



Hình 5

* + - Việc để lộ các dữ liệu nhạy cảm có thể làm ảnh hưởng nghiêm trọng đến hệ thống, rò rỉ các thông tin có giá trị cao khiến kẻ tấn công có thể lợi dụng các thông tin này để phát động các cuộc tấn công khác nguy hiểm hơn rất nhiều.
  + **Lỗi cấu hình (Misconfiguration):** Đây không phải là lỗi trong mã nguồn hay hệ điều hành, mà là sai sót trong quá trình thiết lập và quản trị. Kẻ tấn công khai thác những điểm yếu này để chiếm quyền kiểm soát.
    - **Directory Listing (Liệt kê thư mục):** Khi web server được cấu hình cho phép hiển thị nội dung của một thư mục, kẻ tấn công có thể dễ dàng xem danh sách tệp, bao gồm cả những tệp nhạy cảm như tệp sao lưu hay tệp cấu hình chứa thông tin đăng nhập.
    - **Default Credentials (Thông tin đăng nhập mặc định):** Nhiều dịch vụ web vẫn giữ lại tên người dùng và mật khẩu mặc định sau khi cài đặt. Kẻ tấn công có thể sử dụng những thông tin này để truy cập trái phép.
    - **Unnecessary Services (Các dịch vụ không cần thiết):** Việc bật các dịch vụ không cần thiết như FTP, WebDAV, hay các cổng quản trị từ xa có thể tạo ra các "cửa sau" để kẻ tấn công xâm nhập.
  + **Tấn công khai thác lỗ hổng Web Server (Web Server Exploitation )**
    - **HTTP Verb Tampering:** Kẻ tấn công thay đổi các động từ HTTP (như GET, POST) sang các động từ khác (như HEAD, TRACE) để vượt qua các lớp bảo vệ và truy cập các tài nguyên nhạy cảm.
    - **URL Manipulation:** Kẻ tấn công thay đổi các tham số trong URL để truy cập các tệp hoặc thư mục bị cấm. Đây là một biến thể của Directory Traversal. Ví dụ, thay đổi id=123 thành id=124 để xem thông tin của một người dùng khác.
    - **Buffer Overflow:** Một lỗi lập trình xảy ra khi một chương trình cố gắng ghi dữ liệu vượt quá giới hạn của bộ nhớ đệm (buffer). Kẻ tấn công có thể lợi dụng lỗi này để chèn và thực thi mã độc trên máy chủ. Mặc dù lỗi này phổ biến hơn ở các ứng dụng cấp thấp, nó vẫn có thể xảy ra trong các ứng dụng web và web server.
    1. **Các giai đoạn của một cuộc tấn công**

Một cuộc tấn công mạng thường diễn ra qua năm giai đoạn tuần tự:

* + **Bước 1: Reconnaissance (Thu thập thông tin)**: Giai đoạn đầu tiên, kẻ tấn công thu thập thông tin về mục tiêu.
    - **Passive Reconnaissance**: Thu thập thông tin một cách bị động mà không tương tác trực tiếp với mục tiêu, ví dụ như tìm kiếm trên Google, cơ sở dữ liệu Whois, hoặc nghe lén.
    - **Active Reconnaissance**: Tương tác trực tiếp với mục tiêu để thu thập thông tin phản hồi, ví dụ như quét mạng để xác định các máy chủ đang hoạt động.
  + **Bước 2: Scanning (Dò quét)**: Kẻ tấn công sử dụng các công cụ như Nmap, Nessus để quét lỗ hổng, quét địa chỉ IP và các cổng mở.
  + **Bước 3: Gaining Access (Thâm nhập)**: Sau khi phát hiện lỗ hổng, hacker sẽ khai thác để xâm nhập vào hệ thống. Họ có thể đánh cắp mật khẩu, phá hủy dữ liệu hoặc nâng quyền để truy cập các thông tin bí mật.
  + **Bước 3: Maintaining Access (Duy trì quyền truy cập)**: Sau khi đã thâm nhập thành công, hacker sẽ cài đặt các chương trình gián điệp như trojan hay backdoor để duy trì quyền kiểm soát, nghe lén thông tin hoặc quay lại vào những lần sau.
  + **Bước 4: Covering Track (Xóa dấu vết)**: Giai đoạn cuối cùng, kẻ tấn công xóa các bằng chứng để tránh bị phát hiện, chẳng hạn như xóa các tệp nhật ký (log files) hoặc các chương trình đã cài đặt.
    1. **Phân loại hacker:**
  1. **Black Hat (Mũ đen)**: Nhóm hacker có mục đích xấu, sử dụng kỹ năng để xâm nhập bất hợp pháp, phá hoại hoặc đánh cắp dữ liệu.
  2. **White Hat (Mũ trắng)**: Certified Ethical Hacker, có nghĩa là hacker thiện chí được chứng nhận. Các chuyên gia bảo mật sử dụng kỹ năng của mình cho mục đích phòng chống và bảo vệ, chỉ thực hiện tấn công khi có sự cho phép.
  3. **Gray Hat (Mũ xám)**: Hoạt động ở ranh giới giữa hai nhóm trên, có thể tìm kiếm lỗ hổng mà không được phép nhưng sau đó lại thông báo cho chủ sở hữu hệ thống.
  4. **Script Kiddie**: Hầu hết những kẻ tấn công là nhóm này. Họ sử dụng các công cụ và đoạn mã có sẵn để khai thác lỗ hổng mà không có kiến thức sâu về kỹ thuật.
  5. **Hacktivist**: Nhóm tấn công vì mục tiêu xã hội hoặc chính trị, ví dụ như thay đổi nội dung trang web để gửi thông điệp.
  6. **Vai trò của an toàn hệ thống**

An toàn thông tin được xây dựng trên bốn yếu tố cốt lõi: Tính bí mật (Confidentiality), Tính toàn vẹn (Integrity), Tính sẵn sàng (Availability), và Tính xác thực (Authenticity). Vai trò của an toàn hệ thống là bảo vệ những yếu tố này khỏi sự phá hoại của kẻ tấn công.

* **Tính bí mật:** Bảo vệ thông tin khỏi sự truy cập trái phép. Kẻ tấn công làm mất tính bí mật khi nghe lén dữ liệu trên đường truyền để trộm mật khẩu, thông tin cá nhân hoặc các tài liệu mật.

Mục tiêu của việc triển khai các giải pháp bảo mật, áp dụng chính sách an ninh là để bảo vệ bốn đặc tính cơ bản trên. Ngược lại, điều mà các kẻ tấn công nhắm tới là phá vỡ các yêu tố cơ bản này của thông tin. Khi một hacker tấn công theo hình thức nghe lén dữ liệu trên đường truyển để trộm mật khẩu của hộp thư điện tử hay tài khoản ftp thì lúc này họ đang phá vỡ tính riêng tư, bí mật của dữ liệu. Đặc tính này được gọi là Confidentiality.[1]

* **Tính toàn vẹn:** Đảm bảo dữ liệu không bị sửa đổi hay thay đổi trái phép. Một cuộc tấn công thành công có thể thay đổi nội dung cơ sở dữ liệu hoặc trang web, phá hoại tính toàn vẹn của thông tin.

Trong trường hợp dữ liệu được truyền sau khi bị chặn bắt sẽ được thay đổi và gởi về cho các máy tính client hay server như giả mạo chứng chỉ điện tử (fake certificate) thì hacker đang tác động vào tính toàn vẹn hay Integrity của thông tin. Hoặc hình thức tấn công Bitflipping còn được gọi là integrity attack, chuyên chặn các dữ liệu truyền hay thông tin trên máy tính và thay đổi nội dung làm cho quản trị viên không thể xác nhận được dữ liệu đúng như mục tiêu của người gởi. Một ví dụ của tấn công Bit-flipping là đánh vào cơ chế mã hóa cipher (mã hóa theo dòng) làm thay đổi ciphertext mặc dù hacker không hề hay biết plaintext là gì (plaintext là dữ liệu gốc, dữ liệu này sau khi được áp dụng một thuật toán mã hóa sẽ cho ra ciphertext. Với dạng tấn công này hacker có thể thay đổi nội dung thông điệp từ “Tôi nợ anh 10.000.000 VND “ thành “Tôi nợ anh 1.000.000 VND” . [1]

* **Tính sẵn sàng:** Đảm bảo các dịch vụ và hệ thống luôn hoạt động bình thường, sẵn sàng phục vụ người dùng. Các cuộc tấn công từ chối dịch vụ (DoS/DDoS) là ví dụ điển hình nhằm làm quá tải hệ thống, khiến người dùng hợp lệ không thể truy cập dịch vụ.

Vậy khi nào các hacker sẽ làm mất tính sẳn sàng hay Availability của dữ liệu và dịch vụ ? Ví dụ một trang web cung cấp thông tin hay dịch vụ cho người dùng bị tấn công từ chối dịch vụ bằng cách gởi một số lượng rất lớn những yêu cầu kết nối, làm cho hệ thống không thể đáp ứng các yêu cầu truy cập khác thì tính sẳn sàng hay khả dụng của trang

web đã bị vô hiệu .[1]

* **Tính xác thực:** Đảm bảo danh tính của người dùng hoặc hệ thống là chính xác và không bị giả mạo. Các cuộc tấn công như chiếm phiên làm việc (session hijacking) hay giả mạo địa chỉ MAC (MAC address spoofing) trực tiếp nhắm vào cơ chế xác thực.

Còn trong trường hợp hacker đánh cướp session (trình bày trong Module 11 Session Hijacking) để chiếm phiên làm việc đã qua xác thực của người dùng thi họ đang tấn công và cơ chế xác thực của hệ thống, đây chính là cơ chế Authenticity. Một tình huống thông dụng khác tấn công vào cơ chế xác thực là giả mạo địa chỉ MAC của thiết bị mạng còn được gọi là MAC address spoofing, dạng tấn công này sẽ vượt qua cơ chế kiểm soát của các router hay hệ thống kiểm soát truy cập trên mạng không dây dựa trên địa chỉ vật lý thông qua phương pháp giả mạo địa chỉ MAC của một máy tính hợp lệ. [1]

Việc bảo vệ web server và các hệ thống liên quan không chỉ là nhiệm vụ kỹ thuật mà còn là chiến lược kinh doanh quan trọng để duy trì hoạt động, bảo vệ tài sản số và xây dựng niềm tin vững chắc với người dùng.

1. **CÁC KỸ THUẬT TẤN CÔNG WEB SERVER**
   1. **SQL Injection:**
      1. **Giới thiệu về SQL Injection**

Đa số các ứng dụng web ngày nay đều sử dụng Ngôn ngữ Truy vấn Cấu trúc (SQL) để quản lý và truy xuất dữ liệu từ các hệ quản trị cơ sở dữ liệu như Oracle, MS SQL hay MySQL. Chính vì vậy, các lỗ hổng liên quan đến SQL thường được xếp vào nhóm nguy hiểm nhất, và một trong những dạng tấn công phổ biến nhất là SQL Injection.

SQL Injection là một kỹ thuật tấn công cho phép kẻ tấn công lợi dụng những lỗ hổng trong quá trình kiểm tra và lọc dữ liệu đầu vào của các ứng dụng web. Bằng cách "tiêm" (inject) các câu lệnh SQL bất hợp pháp thông qua các form nhập liệu, kẻ tấn công có thể thực thi các truy vấn không mong muốn trên cơ sở dữ liệu, thậm chí trên cả máy chủ đang chạy ứng dụng đó.

Tấn công SQL Injection có thể gây ra những hậu quả nghiêm trọng, từ việc đánh cắp thông tin nhạy cảm của người dùng (như tài khoản, mật khẩu, thông tin thẻ tín dụng) cho đến việc xóa, thay đổi hoặc chèn dữ liệu. Điều này đã được chứng minh qua các vụ tấn công lớn trong lịch sử. Ví dụ, một đợt tấn công vào tháng 12 năm 2010 đã lấy đi hàng trăm ngàn thông tin khách hàng. Nổi tiếng nhất là vụ việc hacker Albert Gonzalez đã đánh cắp 130 triệu thông tin thẻ tín dụng thông qua lỗ hổng này. Tầm quan trọng của SQL Injection đã được khẳng định khi nó đứng đầu danh sách các lỗ hổng bị tấn công nhiều nhất vào năm 2010, cho thấy tính phổ biến và hiệu quả của kỹ thuật này.

* + 1. **Các loại Lỗi thường gặp dẫn đến SQL Injection**

Lỗi SQL Injection thường phát sinh từ sự thiếu sót trong việc xử lý dữ liệu đầu vào của lập trình viên. Có ba dạng lỗi chính:

1. **Không kiểm tra ký tự thoát truy vấn (Escaping Characters)**

Đây là dạng lỗi cơ bản nhất, xảy ra khi mã nguồn không kiểm tra chặt chẽ các ký tự đặc biệt như dấu nháy đơn ( ' ) trong các câu truy vấn. Khi đó, kẻ tấn công có thể chèn các chuỗi ký tự độc hại để biến đổi câu truy vấn gốc.

* + **Ví dụ minh họa:**

Một đoạn mã ASP đơn giản dùng để xác thực đăng nhập:

**statement = "SELECT \* FROM users WHERE name = '" + userName + "';"**

Câu lệnh này được thiết kế để tìm một người dùng có tên khớp với biến userName do người dùng nhập vào.

Nếu kẻ tấn công nhập giá trị **a' or 'true'='true** vào trường userName, câu truy vấn sẽ trở thành:

**SELECT \* FROM users WHERE name =** '**a' OR 'true'='true';**

Vì **'true'='true’** luôn đúng, câu truy vấn sẽ trả về tất cả các bản ghi, cho phép kẻ tấn công vượt qua bước xác thực mà không cần biết mật khẩu.

Trong trường hợp sử dụng API cho phép thực hiện nhiều truy vấn cùng lúc (như một số phiên bản của MySQL), kẻ tấn công có thể thực thi thêm các lệnh khác.

* + Ví dụ, nhập giá trị: **a';DROP TABLE users; SELECT \* FROM data WHERE 't' = 't**

Câu truy vấn sẽ biến đổi thành:

**SELECT \* FROM users WHERE name = 'a';DROP TABLE users; SELECT \* FROM DATA WHERE 't' = 't';**

Hậu quả là bảng users sẽ bị xóa, gây ra thiệt hại nghiêm trọng.

**b) Xử lý không đúng kiểu dữ liệu (Incorrect Data Type Handling)**

Lỗi này xảy ra khi lập trình viên mong đợi một kiểu dữ liệu cụ thể (ví dụ: số nguyên) nhưng lại không kiểm tra tính hợp lệ của dữ liệu đầu vào. Điều này cho phép kẻ tấn công chèn chuỗi ký tự vào trường số, từ đó thực thi các lệnh SQL độc hại.

* + **Ví dụ minh họa:**

Một câu lệnh SQL tìm kiếm dữ liệu dựa trên ID:

**statement := "SELECT \* FROM data WHERE id = " + a\_variable + ";"**

Biến **a\_variable** được mong đợi là một số. Tuy nhiên, nếu kẻ tấn công nhập **1;DROP TABLE users,** câu truy vấn sẽ trở thành:

SELECT \* FROM DATA WHERE **id=1;DROP TABLE users**;

Câu lệnh này sẽ tìm bản ghi có id bằng 1 và sau đó xóa toàn bộ bảng users.

**c) Blind SQL Injection (Tấn công SQL mù)**

Đây là một dạng tấn công tinh vi hơn, trong đó hậu quả của cuộc tấn công không được hiển thị trực tiếp cho kẻ tấn công. Kẻ tấn công phải dựa vào các dấu hiệu gián tiếp như thời gian phản hồi của máy chủ hoặc sự thay đổi nhỏ trên giao diện người dùng để suy luận và trích xuất dữ liệu từng bit một. Dạng tấn công này đòi hỏi nhiều công sức nhưng vẫn có thể được tự động hóa bằng các công cụ chuyên dụng.

* + 1. **Các dạng tấn công SQL Injection thông dụng.**

Có bốn dạng tấn công chính thường được tin tặc sử dụng:

**Dạng 1: Vượt qua kiểm tra lúc đăng nhập (Login Bypass)**

Kẻ tấn công lợi dụng lỗi SQL để vượt qua trang xác thực. Thay vì nhập tên đăng nhập và mật khẩu hợp lệ, chúng sử dụng các chuỗi như ' OR 1=1-- để biến đổi câu truy vấn, khiến điều kiện xác thực luôn đúng và cho phép truy cập trái phép.

**Dạng 2: Tấn công bằng câu lệnh SELECT**

Đây là dạng tấn công phức tạp hơn, yêu cầu kẻ tấn công phải hiểu rõ các thông báo lỗi từ hệ thống. Kẻ tấn công sử dụng từ khóa **UNION** **SELECT** để kết hợp một truy vấn hợp pháp với một truy vấn độc hại.

**Ví dụ:** Nhập vào trường tìm kiếm chuỗi: **' UNION SELECT ALL SELECT OtherField FROM OtherTable WHERE ' '=' (\*).** Câu truy vấn sẽ tìm kiếm thêm dữ liệu từ bảng OtherTable thay vì bảng ban đầu.

Một biến thể khác là dùng lệnh **DROP** **TABLE** để xóa bảng dữ liệu, như **ví dụ:** ' DROP TABLE T\_AUTHORS --.

**Dạng 3: Tấn công bằng câu lệnh INSERT**

Dạng này thường xảy ra trên các form đăng ký hoặc cập nhật thông tin. Nếu hệ thống không kiểm tra tính hợp lệ của dữ liệu, kẻ tấn công có thể chèn các câu lệnh SQL vào các trường nhập liệu.

**Ví dụ:** Một câu lệnh **INSERT** có dạng: **INSERT** **INTO** TableName **VALUES**('Value One', 'Value Two', 'Value Three'). Nếu kẻ tấn công nhập ' + (**SELECT** **TOP** **1** **FieldName** **FROM** **TableName**) + ' vào một trong các trường, câu truy vấn sẽ thực thi thêm một lệnh SELECT để lấy thông tin từ bảng TableName.

**Dạng 4: Tấn công bằng Stored Procedures**

Đây là dạng tấn công nguy hiểm nhất, đặc biệt nếu ứng dụng web được thực thi với quyền quản trị hệ thống (sa). Kẻ tấn công có thể sử dụng các thủ tục lưu trữ (stored procedures) có sẵn trong cơ sở dữ liệu để thực thi các lệnh trên hệ điều hành của máy chủ.

**Ví dụ:** Chuỗi tiêm vào ' ; EXEC xp\_cmdshell 'cmd.exe dir C: ' có thể khiến máy chủ thực thi lệnh liệt kê thư mục trên ổ đĩa C:. Mức độ phá hoại tùy thuộc vào câu lệnh được chèn.

* + 1. **Quy trình tấn công SQL Injection**

Để thực hiện một cuộc tấn công SQL Injection, hacker thường tuân theo một quy trình gồm ba bước chính:

Sử dụng trình duyệt và các công cụ: Tìm kiếm các trang web có form đăng nhập, form tìm kiếm hoặc bất kỳ nơi nào chấp nhận dữ liệu đầu vào của người dùng. Kẻ tấn công có thể tìm kiếm trong mã nguồn của trang để xác định các lệnh POST hoặc GET.

Kiểm tra lỗ hổng: Kẻ tấn công dùng các chuỗi truy vấn kiểm tra đơn giản như dấu nháy đơn ( ' ) hoặc '' or 1=1-- để xem máy chủ có phản hồi bất thường hay không. Phản hồi như use 'a'='a' có thể là dấu hiệu cho thấy máy chủ dễ bị tấn công.

Khai thác và lấy dữ liệu: Khi phát hiện lỗ hổng, hacker sẽ sử dụng các lệnh SQL như SELECT để lấy dữ liệu hoặc INSERT để chèn thông tin độc hại.

Các bước này thường được tự động hóa bằng các công cụ tấn công chuyên dụng như **Havji, SQL Ninja, Web Acunetix Scanner,** và **W3AF**.

* + 1. **Các công cụ tấn công SQL Injection**

Có nhiều công cụ hỗ trợ tấn công SQL Injection, giúp tự động hóa quy trình tìm kiếm và khai thác lỗ hổng:

* + **Havji:** Một chương trình miễn phí, tự động hóa việc khai thác lỗ hổng SQL Injection.
  + **SQL Ninja:** Công cụ chuyên dùng cho tấn công SQL Injection, đặc biệt trên hệ quản trị SQL Server.
  + **Acunetix Web Security Scanner:** Một chương trình thương mại có khả năng quét và phát hiện các lỗi bảo mật, bao gồm SQL Injection, XSS, Directory Traversal, v.v.
  + **W3AF (Web Application Attack and Audit Framework):** Một framework mạnh mẽ giúp kiểm thử và tấn công ứng dụng web.
    1. **Biện pháp phòng chông tấn công SQL Injection**

Để bảo vệ web server khỏi các cuộc tấn công SQL Injection, cần áp dụng các biện pháp sau:

* + **Kiểm tra và lọc dữ liệu đầu vào (Input Validation & Sanitization):** Đây là biện pháp cơ bản và quan trọng nhất. Lập trình viên phải kiểm tra chặt chẽ các biến hoặc dữ liệu đầu vào, đặc biệt là các form nhập liệu trên trang web. Các ký tự đặc biệt mà hacker thường dùng cần được lọc hoặc từ chối.
  + **Áp dụng Prepared Statements (Câu lệnh chuẩn bị):** Đây là phương pháp hiệu quả nhất để ngăn chặn SQL Injection. Thay vì xây dựng câu lệnh SQL bằng cách ghép chuỗi, bạn sử dụng các tham số (placeholder). Điều này giúp tách biệt hoàn toàn mã SQL với dữ liệu đầu vào, ngăn chặn mã độc thực thi.
  + **Gán quyền thích hợp cho người dùng:** Tài khoản kết nối đến cơ sở dữ liệu từ ứng dụng web chỉ nên có các quyền tối thiểu cần thiết. Tránh sử dụng tài khoản có quyền quản trị tối cao (sa).
  + **Hạn chế thông báo lỗi chi tiết:** Không nên hiển thị các thông báo lỗi cơ sở dữ liệu một cách trực tiếp cho người dùng. Điều này có thể tiết lộ cấu trúc của cơ sở dữ liệu và cung cấp thông tin hữu ích cho kẻ tấn công.
  + **Cập nhật và vá lỗi máy chủ:** Thường xuyên cập nhật các bản vá lỗi bảo mật cho máy chủ cơ sở dữ liệu và hệ điều hành.
  + **Sử dụng hệ thống tường lửa (Firewall) và IDS/IPS:** Đặt máy chủ cơ sở dữ liệu sau hệ thống tường lửa để tránh tương tác trực tiếp. Các hệ thống IDS (Intrusion Detection System) và IPS (Intrusion Prevention System) như SNORT có thể giúp phát hiện và ngăn chặn các truy vấn khả nghi.
  + **Cài đặt các module/plugin hỗ trợ:** Đối với các mã nguồn mở như Joomla, có thể cài đặt thêm các plugin chống SQL Injection và các lỗ hổng khác để tăng cường bảo mật.
  1. **Cross-Site Scripting (XSS)**
     1. **Giới thiệu về Cross-Site Scripting (XSS)**

**Cross-Site Scripting (XSS)** là một trong những lỗ hổng bảo mật ứng dụng web phổ biến nhất, cho phép kẻ tấn công chèn các đoạn mã độc hại (thường là JavaScript, nhưng cũng có thể là HTML hoặc các ngôn ngữ kịch bản khác) vào các trang web hợp pháp. Khi người dùng truy cập vào trang web bị lỗi, đoạn mã này sẽ được thực thi trên trình duyệt của họ. Mục tiêu của cuộc tấn công XSS không phải là máy chủ web mà là người dùng cuối, nhằm đánh cắp thông tin nhạy cảm của họ hoặc thực hiện các hành động độc hại.

XSS là một trong những mối đe dọa hàng đầu đối với an toàn ứng dụng web, cùng với SQL Injection và Authentication Hijacking. Nó thường được tận dụng thông qua các khung tiếp nhận dữ liệu của trang web, chẳng hạn như ô tìm kiếm, form bình luận, hoặc các trường nhập liệu khác mà dữ liệu không được kiểm tra và lọc cẩn thận trước khi hiển thị lại cho người dùng. [1]

* + 1. **Cách thức hoạt động và Các dạng tấn công**

XSS hoạt động dựa trên nguyên lý đơn giản: ứng dụng web tin tưởng vào dữ liệu đầu vào của người dùng và hiển thị lại nó mà không có biện pháp phòng ngừa. Kẻ tấn công sẽ chèn mã độc vào dữ liệu đó, và khi trình duyệt của người dùng tải trang, nó sẽ thực thi mã như thể đó là một phần của trang web.

Có ba dạng tấn công XSS chính:

* + **Reflected XSS (XSS phản chiếu):** Đây là dạng tấn công phổ biến nhất. Kẻ tấn công gửi một URL có chứa mã độc hại đến người dùng. Khi người dùng nhấp vào URL này, mã độc sẽ được gửi đến máy chủ, máy chủ sẽ phản chiếu lại mã đó và trình duyệt của người dùng sẽ thực thi nó. Ví dụ, một kẻ tấn công có thể tạo một URL có chứa kịch bản JavaScript để đánh cắp cookie của nạn nhân.
  + **Stored XSS (XSS được lưu trữ):** Dạng này nguy hiểm hơn vì mã độc được lưu trữ vĩnh viễn trên máy chủ. Kẻ tấn công chèn mã độc vào một trường nhập liệu nào đó (ví dụ: bình luận trên một blog, bài viết trên một diễn đàn) và mã này được lưu vào cơ sở dữ liệu. Mỗi khi bất kỳ người dùng nào truy cập vào trang đó, mã độc sẽ được thực thi. Điều này cho phép một cuộc tấn công tác động đến nhiều người dùng mà không cần tương tác trực tiếp với từng nạn nhân.
  + **DOM-based XSS (XSS dựa trên DOM):** Dạng tấn công này xảy ra hoàn toàn ở phía client (trình duyệt của người dùng) mà không cần dữ liệu được gửi đến máy chủ. Kẻ tấn công lợi dụng các lỗ hổng trong mã JavaScript của trang web để thay đổi Cây mô hình đối tượng tài liệu (DOM) của trình duyệt, từ đó chèn và thực thi mã độc.
    1. **Tác hại của XSS**

Khi một cuộc tấn công XSS thành công, kẻ tấn công có thể thực hiện nhiều hành vi độc hại, bao gồm:

* + **Đánh cắp Cookie:** Kẻ tấn công có thể lấy cắp cookie của nạn nhân, trong đó thường chứa thông tin phiên làm việc. Điều này cho phép kẻ tấn công chiếm quyền điều khiển phiên (session hijacking), truy cập vào tài khoản của người dùng mà không cần mật khẩu.
  + **Chuyển hướng người dùng:** Chuyển hướng người dùng đến các trang web giả mạo (phishing) để lừa họ nhập thông tin cá nhân.
  + **Thực thi các hành động độc hại:** Buộc người dùng thực hiện các hành động mà họ không hề biết, chẳng hạn như thay đổi mật khẩu, xóa bài viết, hoặc gửi tiền.
  + **Đính kèm mã độc:** Chèn các mã độc hại vào trang để cài đặt phần mềm độc hại (malware) lên máy tính của người dùng.
    1. **Biện pháp phòng chống và bảo vệ**

Để phòng chống XSS, các nhà phát triển và quản trị viên cần áp dụng các biện pháp bảo mật chặt chẽ, chủ yếu tập trung vào việc xử lý dữ liệu đầu vào và đầu ra.

**Xác thực và lọc dữ liệu đầu vào (Input Validation & Sanitization):** Đây là nguyên tắc cốt lõi để phòng chống XSS. Tất cả dữ liệu đầu vào từ người dùng, bao gồm cả từ các trường nhập liệu, URL và cookie, cần phải được kiểm tra tính hợp lệ và được lọc để loại bỏ các ký tự đặc biệt có thể được sử dụng để chèn mã.

**Mã hóa đầu ra (Output Encoding):** Khi hiển thị dữ liệu từ người dùng lên trang web, cần mã hóa các ký tự đặc biệt như <, >, ", ' để trình duyệt không hiểu đó là một phần của mã HTML/JavaScript mà chỉ coi là văn bản thuần túy. Ví dụ, thay thế < bằng &lt; và > bằng &gt;.

**Sử dụng Content Security Policy (CSP):** Chính sách bảo mật nội dung (CSP) là một lớp bảo mật bổ sung, giúp ngăn chặn việc thực thi các mã độc. CSP cho phép máy chủ web chỉ định những nguồn nội dung (script, CSS, hình ảnh, v.v.) nào được phép tải và thực thi trên trang.

**Sử dụng các công cụ quét lỗ hổng:** Các công cụ quét lỗ hổng ứng dụng web chuyên dụng như Web Acunetix Scanner hoặc W3AF có thể giúp tự động hóa quá trình kiểm tra và phát hiện các lỗ hổng XSS trên website, từ đó giúp quản trị viên có thể khắc phục kịp thời.

**Áp dụng nguyên tắc "least privilege":** Cấu hình các ứng dụng web và máy chủ với quyền hạn tối thiểu cần thiết để hoạt động. Điều này hạn chế khả năng gây hại của một cuộc tấn công XSS nếu nó xảy ra.

* 1. **Directory Traversal (Path Traversal)**
     1. **Tổng quan về tấn công Directory Traversal**

Directory Traversal, hay còn gọi là Path Traversal, là một lỗ hổng bảo mật nghiêm trọng cho phép kẻ tấn công truy cập vào các tệp tin và thư mục nằm ngoài thư mục gốc của ứng dụng web. Lỗ hổng này xảy ra khi ứng dụng web không kiểm tra hoặc lọc đầy đủ dữ liệu đầu vào của người dùng, cho phép kẻ tấn công sử dụng các chuỗi ký tự đặc biệt như ../ (để di chuyển lên một cấp thư mục) để truy vấn hệ thống tập tin của máy chủ.

Đây là một trong những điểm yếu thông dụng của ứng dụng web. Hacker lợi dụng lỗi này để truy vấn hệ thống tập tin và thư mục hệ thống thông qua trình duyệt web hoặc các công cụ khác, với mục đích cuối cùng là đọc các tệp nhạy cảm (như tệp cấu hình, mật khẩu, hoặc mã nguồn) hoặc thậm chí là thực thi lệnh trái phép trên máy chủ. [1]

* + 1. **Cách thức hoạt động và ví dụ thực tiễn (IIS Unicode Exlpoit)**

Một trong những ví dụ điển hình và kinh điển nhất của tấn công Directory Traversal chính là **IIS Unicode Exploit**. Mặc dù đây là một lỗi bảo mật cũ, nó minh họa rõ ràng nguyên lý hoạt động của lỗ hổng này.

Lỗi này tồn tại trên các hệ thống máy chủ web chạy IIS 5 chưa được vá lỗi và liên quan đến cách IIS diễn dịch các ký tự Unicode. Về cơ bản, **Unicode** sẽ chuyển đổi các ký tự của bất kỳ ngôn ngữ nào sang **định dạng chung**. Kẻ tấn công sẽ tận dụng sự diễn dịch sai này để chuyển đổi các ký tự **../** thành **mã Unicode tương ứng**, nhằm đánh lừa máy chủ web.

Cụ thể, chuỗi ký tự **../** được chuyển thành **mã** **Unicode %255C** hoặc các dạng mã hóa khác. Kẻ tấn công sẽ chèn chuỗi này vào URL để vượt qua các cơ chế kiểm soát của máy chủ web và truy cập vào các thư mục hệ thống.

Ví dụ:

Giả sử một trang web có đường dẫn hợp lệ là -**http://www.netpro.etc/scripts/welcome.html**. Nếu máy chủ bị lỗi, kẻ tấn công có thể thử một yêu cầu sau trên trình duyệt:

**http://www.netpro.etc/scripts/..%255C..%255C/winnt/system32/cmd.exe?/c+dir+c:\**

Phân tích yêu cầu này, ta thấy:

* + **..%255C..%255C:** Đây là chuỗi ký tự **../** đã được mã hóa hai lần. Nó cho phép kẻ tấn công di chuyển lên hai cấp thư mục từ thư mục **/scripts/**.
  + /**winnt/system32/cmd.exe**: Sau khi thoát khỏi thư mục gốc của web, kẻ tấn công có thể truy cập vào các tệp tin hệ thống như cmd.exe (công cụ dòng lệnh của Windows).
  + **?/c+dir+c:\:** Đây là các đối số được truyền cho cmd.exe, yêu cầu nó thực thi lệnh dir c:\ để liệt kê toàn bộ nội dung của ổ đĩa C: trên máy chủ.

Nếu cuộc tấn công này thành công, trình duyệt web của kẻ tấn công sẽ hiển thị toàn bộ nội dung của ổ đĩa C:, cho phép họ thu thập thông tin nhạy cảm.

* + 1. **Hậu quả của tấn công Directory Traversal**

Một cuộc tấn công Directory Traversal thành công có thể dẫn đến các hậu quả nghiêm trọng:

* + Đọc và trích xuất dữ liệu nhạy cảm: Kẻ tấn công có thể truy cập các tệp cấu hình, tệp nhật ký, hoặc các tệp chứa thông tin quan trọng.
  + Chiếm quyền điều khiển: Trong các trường hợp nghiêm trọng, lỗ hổng này có thể kết hợp với các lỗ hổng khác để cho phép kẻ tấn công thực thi mã từ xa, chiếm quyền điều khiển hoàn toàn máy chủ.
  + Tiến hành các cuộc tấn công tiếp theo: Thông tin thu thập được có thể được sử dụng để leo thang tấn công sang các thành phần khác của hệ thống.
    1. **Biện pháp phòng chống và bảo vệ**

Để bảo vệ máy chủ web khỏi tấn công Directory Traversal, các nhà phát triển và quản trị viên cần thực hiện các biện pháp sau:

* + **Mã hóa đầu ra và lọc đầu vào:** Mặc dù tấn công Directory Traversal chủ yếu liên quan đến cách xử lý đầu vào, việc mã hóa đầu ra cũng giúp ngăn chặn các cuộc tấn công phức tạp. Quan trọng nhất là cần kiểm tra và lọc chặt chẽ các ký tự đặc biệt như ../ hay các mã hóa Unicode tương ứng để đảm bảo chúng không được sử dụng để truy cập các thư mục ngoài thư mục gốc.
  + **Cài đặt bản vá lỗi và cập nhật đầy đủ:** Đảm bảo hệ điều hành và phần mềm máy chủ web luôn được cập nhật các bản vá lỗi mới nhất. Điều này giúp khắc phục các lỗ hổng đã được biết như IIS Unicode Exploit.
  + **Cấu hình ngăn chặn duyệt thư mục:** Thiết lập máy chủ web để ngăn không cho người dùng duyệt thư mục trên trang web.
  + **Thiết lập quyền truy cập chặt chẽ (Least Privilege):** Áp dụng nguyên tắc least privilege (chỉ cấp quyền vừa đủ) cho các tài khoản người dùng và dịch vụ. Điều này đảm bảo rằng ngay cả khi kẻ tấn công có thể vượt qua cơ chế bảo mật, họ cũng không có đủ quyền để truy cập vào các thư mục hệ thống quan trọng.
  + **Sử dụng Firewall và IDS/IPS:** Các hệ thống Firewall và IDS/IPS (Intrusion Detection/Prevention System) có thể được cấu hình để phát hiện và chặn các chuỗi tấn công đặc trưng như ../ hoặc các biến thể đã được mã hóa của nó.

Bằng cách áp dụng các biện pháp trên, chúng ta có thể tăng cường độ vững chắc của máy chủ web và giảm thiểu rủi ro từ các cuộc tấn công Directory Traversal.

* 1. **File Upload Vulnerability**
     1. **Khái niệm File Upload Vulnerability và lỗ hổng**

Lỗ hổng File Upload là một trong những điểm yếu nguy hiểm nhất của ứng dụng web, cho phép kẻ tấn công tải lên các tệp tin độc hại lên máy chủ. Mối nguy này xuất phát từ việc cấu hình lỏng lẻo của máy chủ web hoặc ứng dụng web. Từ đó hacker có thể “lợi dụng các tài nguyên chia sẽ trên máy chủ web được cấu hình không hợp lý, hay việc gán quyền bị sai như cho phép người dùng bất kì được phép upload và thực thi các chương trình”.[1]

Bản chất của lỗ hổng này nằm ở việc thiếu kiểm soát nghiêm ngặt đối với các tệp tin được người dùng tải lên. Một ứng dụng web cho phép người dùng tải lên các tệp tin với mục đích hợp lệ như ảnh đại diện, tài liệu, hoặc video. Tuy nhiên, nếu các cơ chế kiểm tra không được thực hiện đầy đủ, kẻ tấn công có thể "tặng quà" cho hệ thống bằng cách tải lên một tệp tin chứa mã độc, ngụy trang dưới một định dạng tệp an toàn. Các loại tệp tin độc hại phổ biến nhất được sử dụng trong tấn công này là **webshell**, một dạng **backdoor** (cổng sau) được viết bằng các ngôn ngữ kịch bản web như PHP, ASP, hoặc Python. Webshell cho phép kẻ tấn công thực thi các lệnh hệ thống từ xa thông qua trình duyệt web, biến máy chủ trở thành một "máy tính ma" (zombie) trong mạng botnet của họ.

* + 1. **Các kỹ thuật tấn công**

Quá trình tấn công thường diễn ra qua các bước sau:

* + **Phát hiện lỗ hổng:** Kẻ tấn công tìm kiếm các trang web có chức năng tải tệp tin lên (ví dụ: tải ảnh đại diện, tải tài liệu, v.v.). Các công cụ quét lỗ hổng ứng dụng web như **Acunetix Web Security Scanner** có thể giúp xác định các điểm yếu này.
  + **Tạo tệp tin độc hại (Webshell):** Hacker sẽ tạo một tệp tin chứa mã độc. Loại mã độc được sử dụng phổ biến nhất để chiếm quyền điều khiển là **Trojan** và **Backdoor** [1]. Kẻ tấn công có thể tạo một tệp tin PHP đơn giản có nội dung như **<?php system($\_GET['cmd']); ?>,** sau đó lưu với tên shell.php. Tệp tin này về cơ bản là một backdoor, cho phép hacker thực thi lệnh hệ thống từ xa thông qua tham số cmd trong URL.
  + **Tải lên tệp tin độc hại:** Kẻ tấn công sử dụng chức năng tải lên của ứng dụng web để đưa tệp shell.php lên máy chủ. Do lỗ hổng, máy chủ sẽ chấp nhận và lưu trữ tệp tin này trong một thư mục có thể truy cập công khai (public folder).
  + **Kích hoạt Webshell:** Sau khi tệp tin đã được tải lên, hacker sẽ truy cập vào đường dẫn của tệp tin đó (ví dụ: **http://website.com/uploads/shell.php**). Tại đây, họ có thể truyền các lệnh hệ thống thông qua URL, ví dụ: **http://website.com/uploads/shell.php?cmd=ls** (để liệt kê danh sách tệp tin trong thư mục trên máy chủ) hoặc **http://website.com/uploads/shell.php?cmd=whoami** (để xem tài khoản người dùng hiện tại).
  + **Chiếm quyền điều khiển và leo thang tấn công:** Một khi đã có quyền thực thi lệnh, kẻ tấn công có thể thực hiện nhiều hành vi nguy hiểm hơn, bao gồm:
  + **Đọc và ghi tệp tin:** Truy cập vào các tệp tin nhạy cảm như tệp cấu hình cơ sở dữ liệu để lấy tên người dùng và mật khẩu.
  + **Tải lên các mã độc khác:** Tải lên các công cụ khai thác phức tạp hơn, hoặc thậm chí là tải về các tập tin SAM (Security Account Manager) hay tập tin /etc/passwd trên các máy chủ web dùng hệ điều hành Linux như đã được đề cập trong giáo trình.
  + **Tấn công leo thang đặc quyền (Privilege Escalation):** Tìm cách chiếm quyền quản trị (administrator) trên máy chủ để có thể điều khiển toàn bộ hệ thống.
  + **Deface website:** Thay đổi nội dung trang web để khoe chiến tích.
    1. **Biện pháp phòng chống và bảo vệ**

Để ngăn chặn lỗ hổng file upload, các nhà phát triển và quản trị viên cần thực hiện một quy trình kiện toàn bảo mật (hardening) cho ứng dụng web và máy chủ, bao gồm các biện pháp sau:

Xác thực và kiểm tra loại tệp tin nghiêm ngặt:

* + **Kiểm tra mở rộng tệp tin (File Extension):** Chỉ cho phép các đuôi tệp tin đã được định nghĩa trước (ví dụ: .jpg, .png). Hạn chế sử dụng các định dạng có thể thực thi mã như .php, .asp, .exe.
  + Kiểm tra kiểu MIME (MIME Type): Kiểm tra Content-Type của tệp tin trong yêu cầu HTTP. Ví dụ, chỉ chấp nhận image/jpeg cho ảnh. Tuy nhiên, phương pháp này dễ bị qua mặt nên không nên chỉ dựa vào nó.
  + **Kiểm tra nội dung tệp tin (File Content):** Đây là biện pháp hiệu quả nhất. Sử dụng các hàm để đọc nội dung tệp tin và đảm bảo rằng nó thực sự là một tệp hình ảnh hợp lệ (bằng cách kiểm tra các magic bytes ở đầu tệp) hoặc một tệp tin an toàn.
  + Đặt tệp tin tải lên trong một thư mục an toàn:
    - **Tách riêng thư mục tải lên:** Lưu tệp tin tải lên vào một thư mục nằm ngoài thư mục gốc của web server (webroot).
    - **Loại bỏ quyền thực thi:** Cấu hình máy chủ web để các tệp tin trong thư mục tải lên không có quyền thực thi. Điều này đảm bảo rằng ngay cả khi hacker tải lên được webshell, họ cũng không thể kích hoạt nó.
    - **Thay đổi tên tệp tin tải lên:** Đổi tên tệp tin tải lên thành một chuỗi ngẫu nhiên hoặc tên duy nhất sau khi tải lên thành công, thay vì giữ nguyên tên gốc của người dùng.
    - **Giảm thiểu đặc quyền (Least Privilege):** Gán quyền truy cập thấp nhất có thể cho thư mục tải lên. Chỉ cấp quyền ghi (write) cho người dùng/dịch vụ cần thiết, và loại bỏ quyền thực thi (execute).

Bằng cách áp dụng các biện pháp trên, chúng ta có thể bảo vệ hệ thống web khỏi một trong những mối đe dọa trực tiếp và nguy hiểm nhất từ việc tải lên các tệp tin độc hại.

* 1. **Denial of Service (DoS/ DDoS)**
     1. **Khái niệm về Tấn công Từ chối Dịch vụ (DoS)**

Khi các phương thức tấn công truyền thống như khai thác lỗ hổng cấu hình hay lỗ hổng lập trình trở nên khó khăn, các hacker thường chuyển sang một hình thức tấn công khác, đó là Tấn công Từ chối Dịch vụ (Denial of Service - DoS). Bản chất của DoS là làm cho một hệ thống máy chủ, một trang web hoặc một dịch vụ mạng bị tê liệt, không thể đáp ứng các yêu cầu hợp lệ từ người dùng. Đây được coi là giải pháp cuối cùng khi không tìm thấy cách nào để đột nhập vào hệ thống mục tiêu.

Tấn công DoS đánh vào bản chất tự nhiên của quá trình truyền thông giữa client và server. Khi một server nhận quá nhiều yêu cầu đồng thời, nó sẽ bị quá tải về tài nguyên (CPU, bộ nhớ, băng thông mạng) và buộc phải từ chối các yêu cầu truy cập khác. Hậu quả của một cuộc tấn công DoS không chỉ là thiệt hại về kinh tế do dịch vụ bị gián đoạn, mà còn làm tổn hại nghiêm trọng đến uy tín của tổ chức.

Mục đích chính của các cuộc tấn công DoS là:

* + Làm tràn ngập hệ thống mạng bằng một lượng lớn dữ liệu không cần thiết, khiến các giao dịch thông thường không thể thực hiện được.
  + Ngắt kết nối giữa các máy tính, ngăn chặn người dùng truy cập dịch vụ.
  + Chặn một máy chủ nào đó không cho truy cập dịch vụ.
  + Ngăn các phản hồi đối với một hệ thống hoặc người dùng cụ thể.

Các hacker thường sử dụng các công cụ DoS để gửi một lượng lớn gói tin rác, các yêu cầu kết nối không hoàn chỉnh hoặc các gói tin có kích thước quá lớn, làm tiêu hao tài nguyên của máy chủ và dẫn đến tình trạng từ chối dịch vụ.

* + 1. **Các dạng Tấn Công DoS phổ biến**

Trong môi trường tấn công DoS, có nhiều kỹ thuật khác nhau được các hacker sử dụng để làm quá tải hệ thống mục tiêu. Dưới đây là một số dạng tấn công điển hình:

* + **Ping of Death:** Kẻ tấn công gửi các gói tin IP có kích thước lớn hơn mức tối đa cho phép (thường là 65.535 byte) đến mục tiêu. Máy tính nạn nhân sẽ tốn nhiều thời gian và tài nguyên để xử lý các gói tin này, dẫn đến quá tải và không thể đáp ứng các yêu cầu hợp lệ khác.
  + **LAND** **Attack:** Kỹ thuật này sử dụng các gói tin có địa chỉ IP nguồn và IP đích giống hệt nhau, và thậm chí cả cổng nguồn và cổng đích cũng trùng lặp. Việc xử lý các gói tin này có thể khiến hệ thống mục tiêu rơi vào trạng thái lặp vô tận (loop), dẫn đến tê liệt và từ chối dịch vụ.
  + **WinNuke:** Tấn công này nhắm vào cổng 139 (NetBIOS) trên các máy tính Windows để gửi các gói tin "rác" (out-of-band - OOB) đến mục tiêu. Tấn công WinNuke làm tràn ngập bộ nhớ đệm của giao thức IP, gây ra sự cố cho hệ thống.
  + **CPU Hog:** Đây là một công cụ tấn công DoS làm quá tải tài nguyên CPU của máy tính bị tấn công, khiến hệ thống không thể xử lý các tác vụ thông thường.
  + **Bubonic:** Công cụ này hoạt động bằng cách gửi các gói tin TCP với các thiết lập ngẫu nhiên, làm cho mục tiêu bị quá tải và có thể bị sụp đổ.
  + **RPC Locator:** Dịch vụ này trên hệ thống Windows có thể bị tấn công gây tràn bộ đệm (buffer overflow), dẫn đến tình trạng từ chối dịch vụ trên các máy chủ.
  + **SMURF Attack**: Đây là một hình thức tấn công lỗi thời nhưng vẫn là một ví dụ điển hình về việc lạm dụng giao thức mạng. Kẻ tấn công gửi một lượng lớn yêu cầu ICMP ECHO Request (Ping) đến một mạng con (subnet) theo dạng broadcast, sử dụng địa chỉ nguồn giả mạo là địa chỉ của nạn nhân. Điều này khiến tất cả các máy tính trong mạng con đó đồng thời phản hồi lại cho nạn nhân, làm băng thông mạng bị chiếm dụng hoàn toàn.
  + **“SYN” Flooding:** Tấn công này khai thác lỗ hổng trong quy trình bắt tay ba bước (**Three-way handshake)** của giao thức TCP. Khi một máy client gửi yêu cầu đồng bộ hóa (**SYN**) đến server, server sẽ phản hồi bằng một gói tin **SYN/ACK** và chờ gói tin **ACK** cuối cùng từ client để hoàn tất kết nối. Kẻ tấn công sẽ gửi một số lượng cực lớn các gói tin **SYN** đến server nhưng không bao giờ hoàn tất quá trình bắt tay. Điều này khiến server phải duy trì một lượng lớn các kết nối "một nửa" (half-open connection), làm tiêu hao tài nguyên và không thể tiếp nhận thêm các yêu cầu hợp lệ khác.
    1. **Tấn công từ chối dịch vụ phân tán (DDoS)**

Tấn công Từ chối Dịch vụ Phân tán (**Distributed Denial of Service - DDoS**) là một hình thức tấn công DoS nhưng ở quy mô lớn hơn, được thực hiện từ nhiều máy tính khác nhau. Để thực hiện DDoS, hacker sử dụng một hệ thống mạng máy tính "ma" gọi là botnet. Mỗi máy tính trong mạng này được gọi là một bot hay zombie, đã bị lây nhiễm trojan và có thể được điều khiển từ xa.

Một cuộc tấn công DDoS điển hình bao gồm ba thành phần chính:

* + **Master (Handler):** Chương trình hoặc server điều khiển, được hacker sử dụng để ra lệnh cho toàn bộ mạng botnet.
  + **Slave (Zombie/Bot):** Các máy tính bị lây nhiễm và được điều khiển từ xa.
  + **Victim:** Mục tiêu bị tấn công.
  + Hacker thường phát động các cuộc tấn công DDoS bằng cách ra lệnh cho tất cả các bot trong mạng botnet đồng loạt gửi các yêu cầu đến mục tiêu. Với số lượng khổng lồ các yêu cầu từ hàng nghìn, thậm chí hàng triệu bot, máy chủ nạn nhân sẽ bị quá tải ngay lập tức, dẫn đến sập hệ thống và từ chối tất cả các dịch vụ.
    1. **Các công cụ tấn công điển hình**

Các công cụ tấn công đóng vai trò quan trọng trong việc tự động hóa và thực hiện các cuộc tấn công DoS/DDoS.

* + **Low Orbit Ion Cannon (LOIC):** Đây là một công cụ DDoS phổ biến, được sử dụng rộng rãi bởi các nhóm hacker như Anonymous. LOIC cho phép người dùng thực hiện tấn công DoS bằng cách tạo ra một lượng lớn các yêu cầu TCP, UDP hoặc HTTP đến một máy chủ mục tiêu.
  + **Trinoo:** Một công cụ DDoS hoạt động trên nền tảng Linux, có khả năng gửi một lượng lớn gói tin UDP với địa chỉ IP nguồn giả mạo. Vì UDP là một giao thức thiếu tin cậy và không có cơ chế kiểm tra tính hợp lệ của IP nguồn, Trinoo có thể dễ dàng làm quá tải hệ thống của nạn nhân.
  + **Tribal Flood Network (TFN):** Công cụ này có khả năng làm suy yếu tài nguyên và băng thông của hệ thống mục tiêu bằng cách gửi một số lượng lớn các gói tin UDP và ICMP. TFN sau đó được nâng cấp lên phiên bản TFN2K để khó bị phát hiện hơn.
  + **SubSeven:** Một trong những trojan đầu tiên được sử dụng để tạo ra các mạng botnet tấn công DDoS.
  + **OWASP HTTP Post Tool:** Công cụ này được sử dụng để tấn công và kiểm tra bảo mật cho máy chủ web. Nó có thể được sử dụng để tấn công từ chối dịch vụ các máy chủ web bị lỗi, đặc biệt là các máy chủ Apache.
    1. **Biện pháp phòng chống và giảm thiểu tác hại**

Để phòng chống các cuộc tấn công từ chối dịch vụ, cần áp dụng một chiến lược bảo mật toàn diện và đa lớp. Dưới đây là các biện pháp quan trọng:

* + **Quản lý bản vá (Patch Management):** Thường xuyên cập nhật các bản vá lỗi mới nhất cho hệ điều hành, máy chủ web và các ứng dụng liên quan. Việc vá các lỗ hổng bảo mật sẽ ngăn chặn hacker lợi dụng các điểm yếu của hệ thống để tấn công, ví dụ như lỗi RPC Locator service đã được đề cập.
  + **Sử dụng Firewall và IDS/IPS:**
    - **Firewall:** Cấu hình tường lửa để chặn tất cả các tín hiệu ICMP từ bên ngoài và chỉ mở các cổng dịch vụ cần thiết (như cổng 80 cho HTTP, 443 cho HTTPS).
    - **Intrusion Detection System (IDS):** Triển khai các hệ thống phát hiện xâm nhập để nhận biết các luồng truyền thông nguy hiểm, các cuộc tấn công hoặc virus đang lây lan trên mạng.
  + **Intrusion Prevention System (IPS):** Ngoài chức năng phát hiện, IPS **còn có khả năng tự động ngăn chặn các cuộc tấn công.**
  + **Network-ingress filtering:** Áp dụng cơ chế lọc để loại bỏ các luồng dữ liệu có địa chỉ nguồn giả mạo hoặc không rõ ràng, giúp ngăn chặn một số dạng tấn công như SMURF.
  + **Giới hạn lưu lượng (Rate-limiting):** Cấu hình các thiết bị mạng như router hoặc firewall để giới hạn băng thông trên các giao thức khác nhau. Kỹ thuật này, còn được gọi là traffic shaping, giúp hạn chế số lượng yêu cầu đến server trong một khoảng thời gian nhất định, giảm thiểu tác động của các cuộc tấn công SYN Flooding hoặc các cuộc tấn công dựa trên việc gửi quá nhiều gói tin.
  + Giám sát và kiểm tra hệ thống:
    - Host-auditing: Sử dụng các công cụ chuyên dụng để quét các tệp tin trên máy chủ nhằm phát hiện các công cụ tấn công DDoS hoặc các chương trình botnet nguy hiểm.
    - Network-auditing: Chạy các chương trình quét mạng để phát hiện các thành phần của mạng botnet (agent) và loại bỏ chúng.
  + Kiện toàn bảo mật (Hardening):
    - Tắt các dịch vụ không cần thiết trên máy chủ web.
    - Thực hiện kiểm tra bảo mật đối với các khu vực tiếp nhận dữ liệu đầu vào để ngăn chặn các cuộc tấn công chèn mã độc.
    - Sử dụng các hệ thống bảo vệ DDoS chuyên dụng.
  + Kiểm tra tính toàn vẹn của tập tin: Các công cụ như Tripwire trên Linux hay tính năng Windows File Protection (WFP) trên Windows giúp kiểm tra tính toàn vẹn của các tập tin hệ thống, từ đó phát hiện những thay đổi bất hợp pháp do trojan hoặc các mã độc khác gây ra.
  1. **Tấn công Brute-Force và Dictionary Attack**
     1. **Giới thiệu**

Trong lĩnh vực an toàn hệ thống, việc bẻ khóa mật khẩu (Password Cracking) là một trong những mục tiêu đầu tiên và quan trọng nhất của tin tặc. Lý do là vì mật khẩu là thông tin then chốt để truy cập vào hệ thống. Tỉ lệ thành công của các cuộc tấn công mật khẩu thường cao vì người dùng có xu hướng đặt những mật khẩu dễ nhớ, liên quan đến thông tin cá nhân như ngày sinh, tên người thân, hoặc thậm chí là sử dụng chung một mật khẩu cho nhiều dịch vụ khác nhau. Khi một mật khẩu bị lộ, các hệ thống khác cũng có thể bị ảnh hưởng.

Có nhiều kỹ thuật bẻ khóa mật khẩu, từ thủ công đến tự động. Tấn công thủ công đòi hỏi tin tặc phải có kiến thức về nạn nhân để đoán mật khẩu. Tuy nhiên, các phương pháp tự động dựa trên cơ chế dò từ điển (Dictionary) và tấn công vét cạn (Brute-force) thường hiệu quả hơn.

* + 1. **Các dạng tấn công mật khẩu**

Tấn công mật khẩu có thể phân loại các dạng tấn công mật khẩu thành bốn nhóm chính:

* + **Passive Online Attack:** Tin tặc nghe lén (sniffing) các gói tin trên mạng để bắt giữ mật khẩu trong quá trình xác thực. Mật khẩu bị bắt sau đó sẽ được so sánh với các từ trong một từ điển hoặc danh sách từ (word list). Mặc dù các mật khẩu thường được băm (hashed) hoặc mã hóa (encrypted) khi truyền trên mạng, vẫn có những công cụ đặc biệt giúp tin tặc phá vỡ các thuật toán mã hóa này.
  + **Active Online Attack:** Tin tặc cố gắng đoán mật khẩu của người quản trị một cách tự động. Cuộc tấn công này dựa trên giả định rằng quản trị viên sử dụng mật khẩu đơn giản, dễ đoán và chỉ hữu dụng với những mật khẩu yếu. Các chương trình tự động có thể nhanh chóng tạo ra các tổ hợp từ, từ điển hoặc danh sách từ để cố gắng đăng nhập. Hầu hết các hệ thống đều có cơ chế khóa tài khoản sau một số lần đăng nhập sai để ngăn chặn loại tấn công này.
  + **Offline Attack:** Đây là hình thức tấn công được thực hiện tại một vị trí khác, không phải trực tiếp trên máy tính chứa mật khẩu. Tin tặc cần phải truy cập vật lý vào máy tính, sao chép các tệp tin chứa mật khẩu (ví dụ: tệp SAM trên Windows hoặc /etc/passwd trên Linux) và sau đó sử dụng các công cụ chuyên dụng để bẻ khóa tại máy tính của mình. Offline Attack bao gồm ba loại hình chính:
    - **Dictionary Attack:** Đây là cách tấn công đơn giản và nhanh nhất. Nó sử dụng một tập tin từ điển chứa các từ có thể là mật khẩu. Hàm băm của các từ trong từ điển sẽ được so sánh với hàm băm của mật khẩu người dùng đã được lưu trữ. Kỹ thuật này chỉ hiệu quả nếu mật khẩu nằm trong từ điển và không chứa các ký tự đặc biệt hoặc số.
    - **Hybrid Attack:** Đây là cấp độ tấn công tiếp theo nếu Dictionary Attack thất bại. Cuộc tấn công này bắt đầu với một tập tin từ điển và sau đó thay thế hoặc bổ sung các con số, ký tự đặc biệt vào các từ trong từ điển. Ví dụ, nó có thể thử Adm1n1strator thay vì Administrator.
    - **Brute-Force Attack:** Là cuộc tấn công sử dụng thuật toán vét cạn, thử mọi tổ hợp có thể có của các chữ cái (hoa và thường), số và ký hiệu. Đây là phương pháp chậm nhất nhưng hiệu quả nhất, vì nó sẽ thành công nếu có đủ thời gian và sức mạnh xử lý.
  + **Non-Electronic Attack:** Đây là các cuộc tấn công không sử dụng kiến thức kỹ thuật, mà dựa vào yếu tố con người (Social engineering) để lừa nạn nhân tiết lộ thông tin.
    - **Ví dụ:**
      * Giả mạo Facebook: Lấy cắp thông tin đăng nhập để chiếm đoạt tài khoản, lừa đảo bạn bè hoặc phát tán mã độc.
        + Cách thức lừa đảo: Kẻ gian sẽ tạo ra một trang web có tên miền gần giống Facebook (ví dụ: facebook-login-reward.com, meta-security-login.net). Họ thường gửi đường link này qua tin nhắn Messenger, email hoặc quảng cáo trên các hội nhóm với nội dung hấp dẫn như: "Nhận 1000 lượt thích và 5000 lượt theo dõi miễn phí!": Trang web yêu cầu bạn đăng nhập để "kích hoạt" tính năng, "Cảnh báo: Tài khoản của bạn có hoạt động bất thường. Vui lòng xác minh để tránh bị khóa.": Đây là chiêu đánh vào tâm lý lo sợ, khiến người dùng vội vàng nhập thông tin. "Bạn đã trúng thưởng iPhone 15! Vui lòng đăng nhập để nhận quà.": Lợi dụng lòng tham để dẫn dụ người dùng.
      * Giả mạo Riot Games: Chiếm đoạt tài khoản game (League of Legends, Valorant) để bán, lấy các vật phẩm quý hiếm hoặc lừa đảo game thủ khác.
        + Cách thức lừa đảo: Các trang web lừa đảo thường có tên miền tương tự như riot-games-reward.com, leagueoflegends-giveaway.net. Đường link được phát tán qua kênh chat trong game, Discord hoặc các diễn đàn game với những lời mời chào hấp dẫn:"Sự kiện nhận skin hiếm/tướng miễn phí nhân dịp kỷ niệm.": Người dùng sẽ thấy một trang web yêu cầu đăng nhập bằng tài khoản Riot để nhận phần quà". Nhận gói trang phục tối thượng (Ultimate Skin) cho nhân vật yêu thích.": Trang web lừa đảo sẽ hiển thị hình ảnh các skin đắt tiền, kích thích sự tò mò và lòng tham của người dùng.
    1. **Công cụ tấn công Brute-Force và Dictionary**

Để thực hiện các cuộc tấn công Brute-Force và Dictionary, tin tặc thường sử dụng các công cụ chuyên dụng, một số công cụ được đề cập trong giáo trình bao gồm:

* + **L0phtCrack (LC5/LC6):** Một ứng dụng mạnh mẽ có khả năng bẻ khóa mật khẩu bằng cách sử dụng từ điển hoặc tấn công vét cạn. Tốc độ của nó phụ thuộc vào hiệu năng của máy tính.
  + **John The Ripper:** Công cụ dòng lệnh phổ biến, có thể bẻ khóa mật khẩu trên cả hệ điều hành Windows và Unix.
  + **OphCrack:** Một Live CD mạnh mẽ và miễn phí, chuyên dùng để bẻ khóa mật khẩu hệ thống Windows.
  + **SMB Relay và SMB Relay 2:** Các công cụ này thực hiện tấn công man-in-the-middle để nghe lén các thông tin xác thực (đã được băm) trong các giao dịch SMB.
  + **Samdump và Pwdump2:** Công cụ dùng để trích xuất thông tin tài khoản và mật khẩu đã được mã hóa từ tệp tin SAM của hệ thống Windows.
    1. **Giải pháp phòng chống và bảo vệ**

Để bảo vệ các web server và hệ thống khỏi các cuộc tấn công bẻ khóa mật khẩu, các quản trị viên cần áp dụng các biện pháp phòng ngừa chặt chẽ:

* + **Chính sách mật khẩu mạnh:** Đây là biện pháp cơ bản và quan trọng nhất. Mật khẩu cần có độ dài tối thiểu 8 ký tự, bao gồm chữ hoa, chữ thường, số và ký tự đặc biệt. Mật khẩu càng dài và phức tạp, thời gian để bẻ khóa bằng Brute-Force càng lâu.
  + **Không sử dụng mật khẩu mặc định hoặc dễ đoán:** Tránh sử dụng các mật khẩu phổ biến như password, 123456, admin, hoặc những thông tin cá nhân như ngày sinh, tên người yêu.
  + **Sử** **dụng cơ chế khóa tài khoản:** Cấu hình hệ thống để tự động khóa tài khoản sau một số lần đăng nhập sai nhất định (ví dụ: 5 lần) trong một khoảng thời gian.
  + **Áp dụng Syskey:** Trên các hệ thống Windows, Syskey là một tính năng bảo mật nâng cao giúp mã hóa thông tin tài khoản và mật khẩu, làm cho các công cụ bẻ khóa thông thường trở nên vô dụng.
  + **Thường xuyên thay đổi mật khẩu:** Theo khuyến nghị của các chuẩn an toàn thông tin như ISO 27001, mật khẩu nên được thay đổi định kỳ (ví dụ: 24 hoặc 48 ngày).
  + **Giám sát nhật ký (Event Viewer Log):** Quản trị viên cần thường xuyên kiểm tra nhật ký hệ thống để phát hiện các dấu hiệu bất thường, chẳng hạn như các nỗ lực đăng nhập thất bại liên tục từ một địa chỉ IP duy nhất, có thể là dấu hiệu của một cuộc tấn công.
  + **Sử dụng xác thực đa yếu tố (MFA):** Đây là một biện pháp hiệu quả giúp bảo vệ tài khoản ngay cả khi mật khẩu bị lộ, bằng cách yêu cầu thêm một yếu tố xác thực thứ hai (ví dụ: mã OTP gửi về điện thoại).

1. **THỰC HÀNH DEMO**
   1. **Mô tả**
   2. **Các công cụ sử dụng**
   3. **Hướng dẫn từng bước**
2. **BIỆN PHÁP PHÒNG CHỐNG VÀ BẢO VỆ**
   1. **Phòng Chống Sql Injection Và Xss**
      1. **Prepared Statement**
      2. **Input Sanitation**
   2. **Bảo mật file upload**
   3. **Các biện pháp khác**
      1. **Firewall**
      2. **IDS/IPS**
3. **REFERENCES**

[1] “Bộ ebook PDF Đào tạo hacker mũ trắng - CEH v9 (Tiếng Việt).” Accessed: Aug. 30, 2025. [Online]. Available: https://www.thekalitools.com/2017/04/bo-ebook-pdf-dao-tao-hacker-mu-trang-ceh.html

[2] “Tổng quan một số kỹ thuật khai thác lỗ hổng bảo mật Web (P1).” Accessed: Sept. 02, 2025. [Online]. Available: https://viblo.asia/p/tong-quan-mot-so-ky-thuat-khai-thac-lo-hong-bao-mat-web-p1-gGJ59MOP5X2

[3] “Web Server là gì? Tìm hiểu về Web Server.” Accessed: Sept. 02, 2025. [Online]. Available: https://viblo.asia/p/web-server-la-gi-tim-hieu-ve-web-server-WR5JRv0rJGv