LỖ HỔNG CROSS SITE SCRIPTING (XSS), TẤN CÔNG VÀ CÁC BIỆN PHÁP KHẮC PHỤC

KS. Nguyễn Ngọc Quân

Tổ NCPT An toàn thông tin

Tóm tắt*: XSS (Cross site scripting) là một lỗ hổng ứng dụng web trong đó một người dùng cuối có thể tấn công bằng cách chèn vào các website động (ASP, PHP, CGI, JSP ...) những thẻ HTML hay những đoạn mã script nguy hiểm có thể gây nguy hại cho những người sử dụng khác. Lỗ hổng XSS đã tồn tại từ lâu nhưng kịch bản hiện nay vẫn có thể thực hiện với những kiểu tấn công mới trong tương lai. Bài viết này trình bày một nghiên cứu chuyên sâu trong sự nguy hiểm của lỗ hổng XSS và cách khai thác lỗ hổng, nó cũng giới thiệu các biện pháp khắc phục các cuộc tấn công XSS.*

1. GIỚI THIỆU

Với sự ra đời của công nghệ phát triển web động, cùng với việc sử dụng ngày càng nhiều các ứng dụng web thì cũng gây ra nhiều lỗ hổng hơn cho Web. Cross Site Scripting (gọi tắt là CSS hay thường là XSS) là một trong những cuộc tấn công tiêm mã phổ biến nhất. XSS là một lỗ hổng dựa trên việc tiêm mã - (Injection) được tìm thấy trong các ứng dụng web trong đó các mã độc hại được tiêm như các biến đầu vào vào payload. Khi người dùng hợp pháp truy cập vào một ứng dụng web bị lây nhiễm , các mã độc hại được lặp lại cho trình duyệt của người dùng. Mã tiêm có khả năng đọc , thay đổi và truyền tải dữ liệu được phân loại truy cập bằng trình duyệt như cookies, session tokens.

XSS (Cross- site Scripting (XSS) - OWASP) là một lỗ hổng đó là tồn tại từ lâu . Một cái nhìn chi tiết hơn về XSS có tham khảo (Shanmugam & Ponnavaikko , 2008). XSS là một lỗ hổng trong top 10 lỗ hổng hàng năm của OWASP. Trong bài báo này tập trung chính khai thác XSS, đó là các cuộc tấn công có thể được thực hiện sau khi lỗ hổng XSS được tìm thấy hoặc kết hợp với các công cụ khai thác. Trong bài báo này , đầu tiên bài báo trình bày các tính năng cơ bản của XSS, một số cách phát hiện XSS phổ biến và các công cụ khai thác lỗ hổng XSS trong Phần 2. Trong phần 3, bài viết mô tả các loại XSS: Non-Persistent or Reflected Vulnerability; Stored or Persistent vulnerability; DOM based or Local XSS. Phần 4 cung cấp mô tả về một số lỗ hổng bảo mật mới và thú vị được tìm thấy trên các trang web gần đây và làm thế nào để có thể khai thác . Trong Phần 5 , bài báo đã liệt kê một vài biện pháp khắc phục có thể được thực hiện trên phía máy chủ cũng như trên các client để bảo vệ một trang web hay ứng dụng từ các lỗ hổng XSS và cuối cùng là kết luận..

2. NỘI DUNG NGHIÊN CỨU

Ở đây, bài báo trình bày một phân tích ngắn gọn về các framework phổ biến khác nhau mà tồn tại cho việc phát hiện ra các lỗ hổng XSS trong các ứng dụng web, và cách khai thác chúng. Chúng làm việc bằng cách injecting các payload và chạy các script trên lỗ hổng web.

2.1. Xenotix

Xenotix (Abraham , 2012) về cơ bản là một công cụ kiểm tra thâm nhập được sử dụng để khai thác bài XSS. Nó có một danh sách payload được xây dựng, có hơn 450 payload XSS, mà chúng có thể vượt qua các bộ lọc XSS cơ bản được sử dụng bởi các nhà phát triển web. Nó có thể sử dụng các payload một cách manual hay chế độ tự động. Đồng thời Nó có thể hoạt động như một key logger để lưu lại tổ hợp phím được thực hiện bởi người dùng khi người truy cập vào trang bị nhiễm.Kẻ tấn công cũng có thể tải về một tập tin thực thi mã độc trên hệ thống của người dùng mà họ không nhận thức được việc đó. Khi người dùng truy cập các trang đã bị nhiễm, java applet client.jar

sẽ truy cập vào cửa sổ lệnh của hệ thống của họ. Attacker sử dụng lệnh echo để viết các script có tên winconfig.vbs trong thư mục (% temp%) và sau đó cmd.exe sẽ thực thi winconfig.vbs để tải về tập tin thực thi độc hại theo quy định của kẻ tấn công trong URL vào thư mục temp và đổi tên nó thành update.exe, cuối cùng nó sẽ thực hiện update.exe. Một lỗ hổng khác được cung cấp bởi Xenotix là cài đặt một reverse shell (Hammer , 2006) tại hệ thống của người sử dụng để truy cập vào máy tính của họ.

Mặc dù là một công cụ đơn giản, nhưng đây là một công cụ đáng được quan tâm. Tính năng keylog không được duy trì nhiều vì nó chỉ có thể capture được bên trong trang bị nhiễm. Nếu tải về ổ đĩa thì chỉ chạy được 16 bit hỗ trợ các file exe.

2.2. XSSF

XSSF được mô tả rõ trong (Tomes , 2011) (htt1) và (xssf - Cross-Site Scripting Framework - project Google Hosting) nhằm mục đích để đưa ra những mối nguy hiểm tiềm tàng liên quan đến các lỗ hổng XSS.Công việc cơ bản của nó bao gồm việc tạo ra một kênh thông tin liên lạc (được gọi là một tunnel XSSF) với trình duyệt mục tiêu (trong đó có một lỗ hổng XSS) để thực hiện các cuộc tấn công khác nhau. Kẻ tấn công có thể thực hiện các cuộc tấn công khác nhau, mỗi cuộc tấn công tồn tại trên một module riêng biệt. Một số lượng lớn các mô-đun như: file stealer, iphone Skype call, network scanning và nhiều lỗ hổng tồn tại khác có thể được thực hiện để khai thác các lỗ hổng ứng dụng web này. XSSF cơ bản hoạt động bằng cách tạo ra một đường hầm liệt kê tất cả các id của nạn nhân khi nạn nhân đến trên một trang web có lỗ hổng XSS. Những kẻ tấn công sau đó kiểm tra trình duyệt của người dùng, tìm kiếm cách khai thác phù hợp, thực hiện nó và gửi một phiên cho người dùng. Sau đó nó có thể truy cập vào hệ thống của người dùng. Các cuộc tấn công XSS có thể được thực hiện bao gồm việc tạo ra một đường hầm XSSF có thể cung cấp truy cập của các máy chủ cục bộ của máy tính từ xa cho kẻ tấn công và cho phép hắn có được chức năng của nó. Đồng thời sử dụng XSSF đã được tích hợp với giao diện điều khiển Metasploit (Offensive Security Ltd , 2012) người ta có thể chạy bất kỳ trình duyệt dựa trên việc khai thác một trang web có lỗ hổng XSS để có được session Meterpreter của nó để gán quyền truy cập hệ thống. Một tính năng khác của công cụ này là XSSF tấn công tự động trong đó khai thác khác nhau có thể được thêm vào trong một hàng đợi , mỗi id công việc riêng của mình và có thể được thực hiện tự động một khi nạn nhân thăm liên kết có lỗ hổng được cung cấp bởi những kẻ tấn công.

Một mặt XSSF cung cấp nhiều tính năng tuyệt vời như một công cụ thành công cho các tấn công Post XSS, mặt khác nó lại không cung cấp một số lượng lớn các phương tiện để phát hiện các lỗ hổng XSS. Đồng thời làm việc với XSSF framework cùng với các hiểu biết của Metasploit.

2.3. BeEF

BeEF là viết tắt của Framework trình duyệt khai thác. Nó là một công cụ kiểm tra thâm nhập mạnh mẽ cho trình duyệt web. Nó sử dụng vector phía khách hàng khác nhau để đánh giá các góc độ an ninh thực tế của môi trường mục tiêu. Framework này bao gồm các mô-đun lệnh khác nhau, sử dụng đơn giản và mạnh mẽ các API góp phần hiệu quả vào việc đánh giá. Nó cho phép phát triển nhanh chóng và dễ dàng sử dụng các mô-đun.

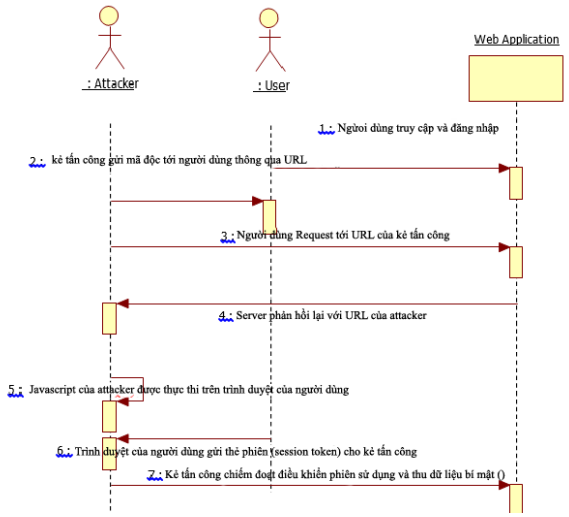
BEeF kết hợp một hoặc nhiều các trình duyệt web đưa ra các mô-đun lệnh, đạo diễn các cuộc tấn công chống lại hệ thống từ bên trong của trình duyệt. Các trình duyệt khác nhau có khả năng nằm trong bối cảnh an ninh (context security) khác nhau, và mỗi bối cảnh có thể có một tập hợp các hướng tấn công đặc thù. Framework cho phép kiểm tra xâm nhập để chọn các module cụ thể (trong thời gian thực) nhắm mục tiêu mỗi trình duyệt, và trong mỗi bối cảnh (context).

BEEF framework là một công cụ mạnh mẽ có thể sử dụng các lỗ hổng XSS để khởi động các cuộc tấn công khác nhau như một vài tên được kể tới sau đây: browser fingerprinting (thu thập thông tin về trình duyệt), persistence , network fingerprinting, DNS enumeration, Port scanning, và IRC NAT.

3. CÁC LOẠI TẤN CÔNG XSS

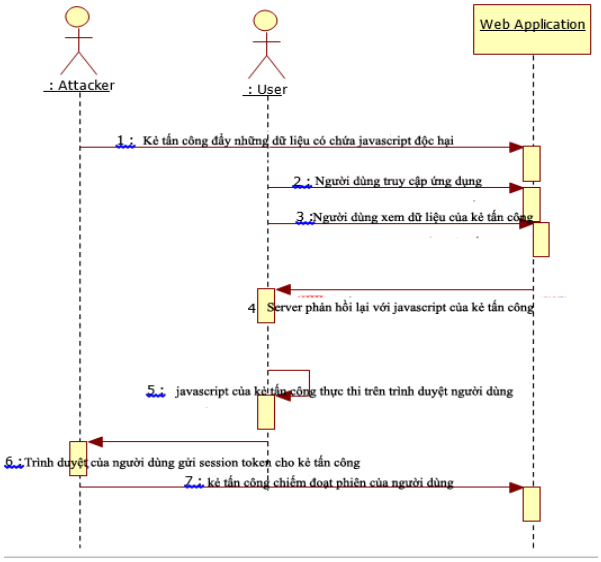
Hiện nay có 3 loại tấn công cross site scripting phổ biến: Non-Persistent or Reflected Vulnerability (Tấn công Reflected hoặc cross site scripting không liên tục); Stored or Persistent vulnerability; DOM based or Local XSS

Những lỗ hổng tồn tại trên những website khác nhau hoặc các ứng dụng web có thể được phân loại thành 3 loại. Chúng được giải thích và mô tả chi tiết như sau:



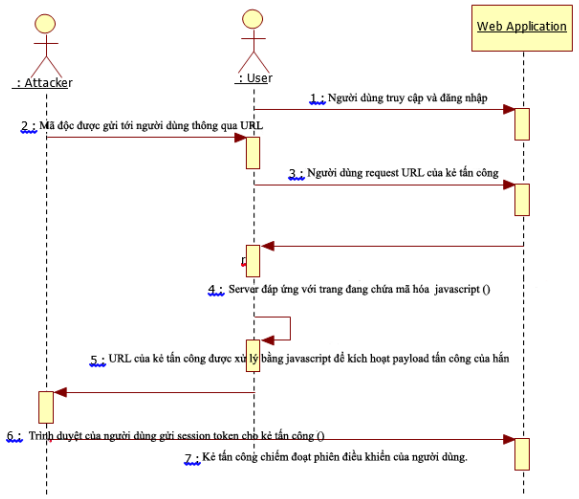
Hình 1. Tấn công Reflected hoặc cross site scripting không liên tục

Các cuộc tấn công không liên tục (Hình 1) được thực hiện khi dữ liệu được cung cấp bởi một khách hàng web được sử dụng ngay lập tức bằng server-side script để tạo ra một trang kết quả cho người dùng. Nếu dữ liệu người dùng cung cấp không còn giá trị và được bao gồm trong các trang kết quả mà không cần mã hóa HTML, việc này cho phép mã phía máy khách được tiêm vào trang năng động. Mã tiêm có thể được phản hồi trên máy chủ web, như trong kết quả tìm kiếm, hoặc như một thông báo lỗi, hoặc bất kỳ thông điệp trả lời như vậy mà bao gồm một phần của đầu vào gửi đến máy chủ như một phần của yêu cầu. Các cuộc tấn công Reflect có thể được gửi đến người dùng thông qua một con đường khác, như trong một e-mail thông báo, hoặc có thể trên một số máy chủ web khác. Khi một người dùng bị lừa click vào một liên kết độc hại hoặc submit một form đặc biệt, mã tiêm đi đến máy chủ web có lỗ hổng, reflect cuộc tấn công ngược trở lại trình duyệt của nạn nhân. Các trình duyệt sau đó thực thi mã vì nó đến từ một máy chủ trusted



Hình 2. Stored or Persistent vulnerability

Stored or Persistent vulnerability (hình 2) cho phép những tấn công mạnh nhất, trong đó các mã độc hại được gửi đến một trang web, nơi nó được lưu trữ trong thời gian nhất định (trong một cơ sở dữ liệu, hệ thống tập tin, hoặc bất kỳ đâu) và sau đó hiển thị cho người sử dụng trong một trang web trang web mà không được mã hóa bằng cách sử dụng các thực thể HTML. Một ví dụ về một tình hình như vậy là với bảng tin trực tuyến, nơi mà người dùng được phép đăng bài định dạng HTML cho người dùng khác để đọc.



Hình 3. DOM based or Local XSS

Dựa trên DOM (Document Object Model) (hình 3) hoặc Local XSS, kẻ tấn công

nhúng dữ liệu tấn công trong các side client, từ bên trong một vài trang trên máy chủ web. Ví dụ, nếu một phần của JavaScript truy cập một URL yêu cầu các tham số và viết một vài HTML trên trang riêng của mình, việc sử dụng thông tin này mà không được mã hóa bằng cách sử dụng các thực thể HTML, thì có thể sẽ xuất hiện lỗ hổng XSS, khi mà văn bản dữ liệu này sẽ được tái giải thích bởi các trình duyệt như HTML mà có thể bao gồm thêm các script phía máy trạm.

4. CÁC TẤN CÔNG KHAI THÁC XSS

4.1. Dữ liệu trên Android có nhiều lỗ hổng

Các lỗ hổng được giải thích ở đây (Cannon 2013) tồn tại trong framework Android 2.2. Nó có thể được khai thác để truy cập các tập tin được lưu trữ trong SDcard của các thiết bị chạy Android. Các Trình duyệt trên Android không nhắc nhở người dùng khi tải về một tập tin, ví dụ như một tập tin như "payload.html" được tự động tải về / sdcard / download / payload.html. Một JavaScript có thể được sử dụng để mở file " payload " một cách tự động mà là nguyên nhân trình duyệt để hiển thị các file local và cho phép các cách thức để có thể truy cập vào SDcard và các tập tin được lưu trữ bên trong đó. Sau đó, Nó có thể gửi nội dung của các tập tin truy cập trở lại trang web có lỗ hổng.Việc khai thác đơn giản là sử dụng JavaScript và chuyển hướng, nó có thể được sử dụng trên nhiều thiết bị cầm tay và các phiên bản khác nhau của Android. Nhưng nó cũng có một vài hạn chế như tên và đường dẫn của tập tin được truy cập đã được biết đến trước đó. Vì nó không phải là một lỗ hổng root nên nó không thể truy cập tất cả các tập tin, mà chỉ có những gì được lưu trữ trên SDcard.

4.2. Skype's improper URI scheme and embeddable Webkit browser on IOS

Lỗ hổng này như được giải thích trong (Kumar, 2011) và (Purviance, 2011) và (iPhones Make Automatic Skype Calls | Security Generation, 2010) tồn tại trong framework của iOS. Nó có thể bị khai thác bởi một kẻ tấn công để truy cập vào cơ sở dữ liệu SQLLite Address Book của người dùng và cũng để đặt cuộc gọi trực tiếp sử dụng Skype. Ứng dụng Skype được phát triển cho iOS sử dụng một tập tin HTML được lưu trữ local để hiển thị tin nhắn chat từ người dùng Skype khác, nhưng nó thất bại trong việc mã hóa "Full Name" của người dùng đến (incoming users), cho phép kẻ tấn công để thực thi mã JavaScript độc hại khi nạn nhân xem tin nhắn.

Vấn đề ở đây là thực hiện khai thác bằng cách sử dụng trình duyệt nhúng Webkit. Ngoài ra các nhà phát triển Skype đã thiết lập các chương trình URI cho trình duyệt nhúng "file :/ /" cho phép kẻ tấn công truy cập hệ thống tập tin và đọc bất kỳ tập tin có thể được đọc bởi các ứng dụng iOS sandbox.

Trong tương lai, Cần hạn chế các ứng dụng của bên thứ ba để thực hiện các hành động được xác định bởi URL cũng như URI cho phép các trang web nhúng một iframe mà buộc Skype mở ra(nếu nó được cài đặt) và gọi một số cụ thể. JavaScript <iframe src="skype://1900expensivepremiumnumber ?call"> </ iframe>.

4.3. HTML5 API for cross domain calls

Lỗ hổng này chỉ có thể được khai thác trên các hệ thống Windows. HTML5 có hai API để thực hiện cuộc gọi liên miền - Cross Origin Requests và WebSockets. Bằng cách sử dụng chúng, JavaScript có thể tạo ra các kết nối tới bất kỳ IP nào và với bất kỳ cổng (ngoài cổng bị chặn), làm cho chúng một đối tượng lý tưởng cho tấn công port scanning. Các API có thể bị khai thác để xác định xem nếu các cổng đang được kết nối là mở hay đóng hay lọc. Nó như vậy bằng sự giúp đỡ của hai thuộc tính: 'ready state' cho biết tình trạng của các kết nối tại một thời điểm nhất định và "'time duration' mà mỗi "readyState" là giá trị cuối.

Do đó bằng cách quan sát sự khác biệt trong hành vi chúng ta có thể xác định bản chất của các cổng. Là một cấp độ ứng dụng quét thành công của nó cũng phụ thuộc vào bản chất của các ứng dụng đang chạy trên các cổng mục tiêu. Khi một yêu cầu được gửi đến số loại ứng dụng mà chúng đọc yêu cầu và giữ im lặng giữ cho socket open, có thể có nhiều đầu vào hoặc đầu vào trong một định dạng cụ thể. Nếu mục tiêu đang chạy một ứng dụng như vậy thì tình trạng của nó

không thể được xác định. Vì ngay cả khi cổng đóng có thể vẫn được xác định chúng ta có thể mở rộng kỹ thuật này để thực hiện các chức năng quét mạng cũng như phát hiện IP nội bộ.

4.4. HTML5 implementation of AJAX history

HTML5 có một tính năng cho phép người dùng truy cập các trang web khác nhau và liên kết trong một trang web mà không thay đổi URL. Nó được thực hiện với sự giúp đỡ của chức năng window.history.pushState (). Nó được tạo ra cho các trang web AJAX để sửa đổi dễ dàng trong thanh địa chỉ cửa sổ và lịch sử thao tác. Đó là một tính năng tuyệt vời và thuận tiện cho các nhà phát triển - ví dụ, các ứng dụng AJAX có thể dễ dàng hỗ trợ trở lại và nút bấm phía trước mà không cần đến URI định danh đoạn (#). Nhưng nó cũng có thể được khai thác cho một trang web có lỗ hổng XSS vì nó cho phép kẻ tấn công để chuyển hướng người dùng đến bất kỳ liên kết mà không thay đổi URL trong thanh địa chỉ.

4.5. Access to the WScript ActiveX control in Internet Explorer

Các thiết lập bảo mật trong Internet Explorer phép truy cập vào điều khiển ActiveX WScript thông qua ngôn ngữ script như JavaScript và VBScript. Các mẫu ứng dụng cho thấy làm thế nào để sử dụng đối tượng ActiveX "WScript.shell" để tương tác với máy của khách hàng. Với việc kiểm soát ai có thể thực hiện các lệnh tương tự như một dấu nhắc trình báo mà không thông báo cho người sử dụng. Sử dụng Shell người ta cũng có thể tạo, xóa và sửa đổi các tập tin văn bản thông qua WScript.FileSystemObject. IE7 đã đưa vào một điều khiển bảo mật mới được gọi là "nguồn dữ liệu truy cập trên toàn miền", mà bây giờ bằng cách mặc định được thiết lập để nhắc nhở người dùng nếu họ muốn cho phép kịch bản của bạn để nói chuyện với “domains” khác (nó xem xét hệ thống tập tin như là một miền riêng biệt) nhưng người ta có thể viết một kịch bản tập tin trực tiếp vào đĩa và sau đó thực hiện nó, nhận được xung quanh các điều khoản IE7.

4.6. File API in HTML5

Lỗ hổng này hiện đang được thực thi trong Webkit (mới nhất của Google Chrome) và có thể bị khai thác để chuyển đổi trình duyệt chrome Google vào một file server. File API trong HTML5 cho phép các JavaScript truy cập các file, một khi nó được lựa chọn bởi người sử dụng (tức là trước khi tải lên nó). Ngoài việc cung cấp kinh nghiệm để các file upload tốt hơn, nó cũng có thể được sử dụng một cách độc hại như là để ăn cắp các file của bạn trong tấn công XSS. Với phong cách thông minh bạn có thể ẩn *inputtype=file* điều khiển để người dùng không hề biết rằng anh ta sẽ tải lên các tập tin. Trong trường hợp này các tập tin được lựa chọn bởi người sử dụng trong 'Open File' hộp thoại là người duy nhất có thể được truy cập. Tuy nhiên *inputtype=directory file* là một tính năng tuyệt vời cho phép người dùng tải lên nội dung của một thư mục được lựa chọn, như vậy cho phép truy cập toàn bộ thư mục cho kẻ tấn công.

4.7. XSS MAP

Google trong khi thu thập dữ liệu cho các Xem Google Street cũng đã thu thập dữ liệu của các mạng không dây trong vùng lân cận và địa chỉ MAC của các router và sau đó phối hợp ánh xạ chúng vào GPS. Ở đây, như xây dựng trong (Higgins, 2010), một XSS khai thác có thể được sử dụng để lập bản đồ vị trí của người dùng. Việc khai thác XSS có thể lấy địa chỉ MAC của router của mục tiêu và sau đó phối hợp sử dụng Google Maps để xác định GPS. Một trang độc hại bạn đang truy cập có thể thực hiện một XSS khai thác và phục hồi của bạn tọa độ GPS từ Google Maps. Các bộ định tuyến và trình duyệt web tự chúng không chứa bất kỳ dữ liệu vị trí địa lý / GPS và không của nó Geo vị trí dựa trên IP. Nó hoạt động thông qua Router XSS mà có được địa chỉ MAC của router thông qua AJAX . Địa chỉ MAC sau đó được gửi đến kẻ tấn công sẽ chuyển nó đến địa điểm Dựa dịch vụ của Google mà có thể bản đồ vị trí (GPS gần đúng tọa độ) của một người sử dụng dựa trên địa chỉ MAC của mình.

4.8. NAT PINNING - IRC Over HTTP

Trong cuộc tấn công XSS, một trang web buộc router của người dùng hoặc tường

lửa, không biết rằng tới chúng, forward đến cổng bất kỳ số cổng trở lại máy của người dùng. Khi nạn nhân nhấp chuột vào một URL XSS có lỗ hổng có một hình thức ẩn kết nối với http://attacker.com:6667 (port IRC), người dùng submit form mà không biết. Một kết nối HTTP được tạo ra bởi kẻ tấn công tới máy chủ IRC (kết nối giả) chỉ đơn giản là lắng nghe. Router của nạn nhân nhìn thấy một " kết nối IRC " (mặc dù khách hàng của mình đang nói trong HTTP) và một nỗ lực tại một ' DCC Chat" . Direct Client- to-Client (DCC) là một tiểu giao thức IRC liên quan cho phép trao đổi các tập tin và thực hiện các cuộc trò chuyện không chuyển tiếp bằng cách cho phép các Peers kết nối với nhau bằng cách sử dụng một máy chủ IRC cho tín hiệu bắt tay.Chat DCC yêu cầu mở một cổng local trên máy trạm mà được kết nối ngược từ. Khi mà router là ngăn chặn tất cả các kết nối từ bên trong, nó quyết định để chuyển tiếp lưu lượng đến cổng Chat DCC ngược về máy của nạn nhân để cho phép NAT traversal cho những kẻ tấn để kết nối trở lại và trò chuyện với anh ta. Tuy nhiên, kẻ tấn công có chỉ định cổng . Ví dụ, cổng 21 (FTP) , các cổng router chuyển tiếp 21 trở lại hệ thống nội bộ của nạn nhân. Kẻ tấn công có một con đường rõ ràng để kết nối với các nạn nhân trên cổng 21 và khởi động một cuộc tấn công.

4.9. Browser Exploits

Bất kỳ ai có thể khai thác các stack ứng dụng trình duyệt và thực hiện một mã shell hoặc mở một phiên Meterpreter bằng cách sử dụng lỗi bộ nhớ liên quan đến lỗ hổng XSS. Những lỗ hổng khác cũng có thể trả về phiên Meterpreter mà không tấn công các ứng dụng stack một cách trực tiếp. Ví dụ như java applet của ký tự có thể được sử dụng để download các mã độc và thực hiện một tập tin exe.

5. BIỆN PHÁP KHẮC PHỤC XSS

Trong các ứng dụng web thế giới ngày nay đang được phổ biến rộng rãi để cung cấp các dịch vụ trực tuyến khác nhau. Nhưng đồng thời lỗ hổng ứng dụng đang được phát hiện và công bố với tốc độ đáng báo động. Trên thế giới, bảo mật web có thể dễ dàng bị xâm nhập, bảo mật sẽ trở thành bắt buộc để bảo vệ mình khỏi các cuộc tấn côn. Các biện pháp khác nhau có thể được áp dụng để tránh bị thành nạn nhân của XSS. Các cơ chế ngăn ngừa (XSS (Cross Site Scripting) Cheat Sheet - OWASP, 2013) có thể được thực hiện một trong hai ở phía máy chủ hoặc phía khách hàng.

5.1. Server Side protection

Để bảo vệ khỏi các lỗ hổng XSS, các biện pháp sau đây có thể được thực hiện bởi nhà phát triển tại phía máy chủ. Các khái niệm cơ bản sử dụng ở đây là, không tin tưởng vào đầu vào cung cấp (bao gồm cả các tập tin cookie) của người dùng. Người sử dụng cần được xác nhận và xác nhận trước khi cho phép truy cập vào nó. Bảo vệ có thể được thực hiện bằng cách hạn chế các miền và đường dẫn để chấp nhận cookie, thiết lập chúng như HttpOnly, sử dụng SSL và không bao giờ lưu trữ dữ liệu bí mật trong các cookie. Có thể vô hiệu hóa việc sử dụng các Script một cách an toàn từ các trang web khách hàng.

Các Header nội dung Chính sách An ninh cũng có thể được sử dụng để bảo mật chống lại việc khai thác lỗ hổng XSS. Ngoài ra, mã hóa một cách thích hợp các ký tự điều khiển HTML, JavaScript, CSS, và URL nên được thực hiện để làm cho chúng vô hại trước khi chúng được hiển thị trong trình duyệt. Sử dụng các bộ lọc có để làm sạch đầu vào người dùng: filter_sanitize_encoded (để mã hóa URL), htmlentities (lọc HTML), ilter_sanitize_magic_quotes (áp dụng addslashes ()). Các bộ lọc này giữ một chiếc đồng hồ đầu vào người sử dụng và kiểm tra javascript hoặc HTTP POST trong các đầu vào và sau đó ngăn chặn các script được thực thi. Ngoài những biện pháp có một số thư viện bảo mật có sẵn để mã hóa người dùng nhập vào như Project OWASP Encoding có sẵn tại Google Code, các lọc HTML hoặc Htmlawed cho PHP Anti-XSS Class. Các ứng dụng thuần AntiSamy API cho Net hoặc. XSS-HTML -Bộ lọc cho Java.

5.2. Endpoint Protection

Người dùng có thể thực hiện các bước để ngăn chặn trở thành nạn nhân của cross-site scripting bằng cách cài đặt add-ons trình duyệt khác nhau. Những add ons giữ một chiếc đồng hồ trên các trường đầu vào khác nhau (form, URL, vv), nếu một JavaScript

hoặc HTTP POST là gặp phải, nó sau đó sử dụng các bộ lọc XSS để ngăn chặn những script thực hiện. Ví dụ về các tiện ích bao gồm NoScript cho FireFox; NotScripts cho Chrome và Opera trong khi Internet Explorer 8 có chúng như là một tính năng đã được xây dựng từ trước.

6. KẾT LUẬN

Hiện tại, ứng dụng web đã trở thành một phần không thể thiếu của cuộc sống của chúng ta. Nhưng các trang web này thường tồn tại nhiều lỗ hổng và dễ bị tấn công. Bài viết này đã khám phá một trong những lỗ hổng tồn tại một cách phổ biến và chỉ ra cách khai thác nó . XSS là một cuộc tấn công tiêm mã tiền chi phối có thể hình thành các cơ sở khai thác rất mạnh mẽ. Nó thường có thể được kết hợp với các lỗ hổng khác để thực hiện các cuộc tấn công quan trọng hơn nữa. Trong bài báo này, đã thảo luận một vài cuộc tấn công phổ biến. Chúng tôi liệt kê một vài công cụ để phát hiện XSS và khai thác lỗ hổng XSS, cùng với các tính năng chính của chúng. Hơn nữa chúng tôi đã đề cập tới một vài lỗ hổng XSS mới nhất cũng như các cuộc tấn công XSS đồng thời giải thích các khái niệm đằng sau chúng. Trong kết luận đã liệt kê một vài cơ chế bảo vệ có thể được thực hiện hoặc trên server hoặc client để bảo vệ mình khỏi các cuộc tấn công XSS.

7. TÀI LIỆU THAM KHẢO:

1. *http://santoshdudhade.blogspot.in/2012/07/x ssf-v22-cross-site-scripting-framework.html*
2. *Abraham, A. (2012). Detecting and Exploiting XSS with Xenotix XSS Exploit Framework.*
3. *Cannon, T. (2013, november 23). Android Data Stealing Vulnerability | thomascannon.net*
4. *Cross-site Scripting (XSS)- OWASP . (n.d.). Retrieved February 2013, from www.owasp.org.*
5. *Kumar, M. (2011, September 20). iPhone Skype XSS Vulnerability Lets Hackers Steal Phonebook*

Thông tin tác giả:

Nguyễn Ngọc Quân

Sinh năm: 1985

Lý lịch khoa học:

- Tốt nghiệp đại học kỹ thuật điện Quốc gia Saint Peterburgs, 2009, chuyên ngành khoa học máy tính.
- Hiện đang công tác tại Tổ NCPT An toàn thông tin thuộc Viện công nghệ Thông tin và Truyền thông – CDIT, Học viện Công nghệ Bưu chính Viễn thông.

Lĩnh vực nghiên cứu hiện nay: an ninh hạ tầng mạng, an ninh ứng dụng và bảo mật điện toán đám mây.

Email: quannn@ptit.edu.vn