TÌM HIỂU TOP 10

NHỮNG RỦI RO BẢO MẬT NGUY HẠI NHẤT

TRÊN CÁC ỨNG DỤNG WEB NĂM 2013

(OWASP TOP 10 - 2013 )

Ngô Anh Long – Nhóm 1: Network security

Nội dung

[I. Giới thiệu về OWASP 3](#_Toc430722638)

[II. Những thay đổi từ bản cập nhật năm 2010 đến 2013 4](#_Toc430722639)

[III. Danh sách Top 10 rủi ro ứng dụng nhất năm 2013 6](#_Toc430722640)

[IV. Tìm hiểu về tiêm nhiễm mã độc(Injection) và sai lầm trong kiểm tra định danh (broken authenticate and session management): 7](#_Toc430722641)

[1. A1 - Tiêm nhiễm mã độc (Injection) 7](#_Toc430722642)

[Tôi có bị mắc lỗi nhúng mã hay không ? 8](#_Toc430722643)

[Làm cách nào để tôi có thể ngăn chặn lỗi nhũng mã 9](#_Toc430722644)

[Ví dụ về kịch bản tấn công 9](#_Toc430722645)

[2. A3 – Sai lầm trong kiểm tra định danh 9](#_Toc430722646)

[Tôi có bị mắc lỗi không ? 11](#_Toc430722647)

[Làm cách nào để ngăn chặn ? 12](#_Toc430722648)

[Vi dụ về kịch bản tấn công: 12](#_Toc430722649)

# Giới thiệu về OWASP

OWASP viết tắt của The Open Web Application Security Project (dự án mở về bảo mật ứng dụng Web), dự án là một sự cố gắng chung của cộng đồng giúp các tổ chức có thể phát triển, mua hoặc bảo trì các ứng dụng an toàn. Ở OWASP ta sẽ tìm thấy nhiều thứ “miễn phí” và “mở” (free and open) sau đây:

- Công cụ và các tiêu chuẩn về an toàn thông tin

- Sách về kiểm tra bảo mật, lập trình an toàn và các bài viết về bảo mật mã nguồn

- Thư viện và các tiêu chuẩn điều khiển bảo mật

- Các chi nhánh của hội ở khắp nơi trên thế giới

- Những nghiên cứu mới nhất

- Những buổi hội thảo toàn cầu

- Địa chỉ thư tín chung

- Và nhiều thứ khác, xem thêm tại [www.owasp.org](http://www.google.com/url?q=http%3A%2F%2Fwww.owasp.org&sa=D&sntz=1&usg=AFQjCNHzeeyXhbJKpAUHFK8OfOevFgc6uw)

Mục tiêu của dự án Top 10 là nâng cao nhận thức về ứng dụng bảo mật bằng cách xác định những rủi ro quan trọng nhất đối diện với các tổ chức. Dự án Top 10 được đề cập bởi nhiều tiêu chuẩn, sách, công cụ và các tổ chức, bao gồm MITRE, PCI DSS, DISA, FTC, …. Sự ra đời của OWASP Top 10 đánh dấu 12 năm của dự án này hoạt động nhằm nâng cao nhận thức về tầm quan trọng của những rủi ro trong bảo mật ứng dụng. OWASP Top 10 ra mắt lần đầu tiên vào năm 2003, vài bản cập nhật nhỏ được thực hiện vào năm 2004 và 2007, 2010 và trong bài giới thiệu này, OWASP đề cập tới Top 10 của 2013.

OWASP là một mô hình tổ chức mới. Không bị vấn đề thương mại hóa ảnh hưởng giúp cho OWASP đưa ra những thông tin chính xác, không thiên vị và có giá trị về an toàn thông tin. OWASP không liên kết với bất kì công ti kỹ thuật nào, dù OWASP hỗ trợ về các mặt kỹ thuật trong an toàn thông tin. Cũng giống như những dự án phần mềm mã nguồn mở, OWASP tạo ra rất nhiều sản phẩm bằng sự phối tác của cộng đồng.

Nền tảng OWASP là đơn vị phi lợi nhuận nhằm đảm bảo sự thành công lâu dài của dự án. Hầu hết mọi người liên quan đến OWASP là tình nguyện viên, bao gồm Hội đồng Quản Trị, Ủy ban toàn cầu, Lãnh đạo các chi nhánh, Lãnh đạo các dự án, và thành viên dự án. OWASP hỗ trợ sáng tạo trong nghiên cứu bảo mật với các khoản trợ cấp và cơ sỡ hạ tầng.

Tuy nhiên !

“ Đừng dừng lại ở 10. Có hàng trăm vấn đề có thể ảnh hưởng đến bảo mật tổng thể của một ứng dụng Web như đã thảo luận trong [OWASP Developer’s Guide](http://www.owasp.org/index.php/Guide). Đây là điều cần thiết nên đọc cho bất kỳ ai muốn phát triển ứng dụng Web ngày nay. Hướng dẫn về cách tìm kiếm hiệu quả những lỗ hổng trong ứng dụng Web được cung cấp trong [OWASP Testing Guide](http://www.owasp.org/index.php/Category:OWASP_Testing_Project) và [OWASP Code Review Guide](http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project), trong đó có cả những cập nhật đáng kể từ những phiên bản trước của OWASP Top 10 “

# Những thay đổi từ bản cập nhật năm 2010 đến 2013

Trong bối cảnh các nguy cơ an ninh tới ứng dụng ngày càng thay đổi nhanh chóng. Những mối nguy hiểm tiềm tàng luôn được làm cho thích nghi tốt hơn và nâng cao hơn bởi những kẻ tấn công, việc đưa ra các công nghệ mới với những điểm yếu mới phải song song với việc xây dựng cơ chế phòng thủ trong quá trình triển khai các hệ thống ngày càng phức tạp. Để giữ nhịp độ đó, OWASP luôn định kì cập nhật top 10. Trong bản công bố lần này, OWASP đã có các thay đổi sau:

1. Sai lầm trong kiểm tra định danh (Broken Authentication and Session Management) trở nên càng phổ biến hơn dựa trên những tập dữ liệu mà OWASP có được. Được di chuyển lên A2 thay vì A3 như bản cập nhật Top 10 - 2010.
2. Giả mạo yêu cầu {Cross Site Request Forgery (CSRF) được giảm mức độ phổ biến dựa trên những tập dữ liệu OWASP có được, giảm từ A5 – 2010 tới A8 – 2013. Trong 6 năm, các tổ chức và những người phát triển theo framework đã tập trung vào nó đủ để nhận định và giảm số vụ tấn công CSRF vào các ứng dụng trên thế giới. OWASP tin rằng đây là lí do mà CSRF giảm mức độ như vậy
3. OWASP mở rộng vấn đề “thiếu hạn chế truy cập bằng đường dẫn - Failure to Restrict URL Access” của Top 10 năm 2010 bao gồm thành:

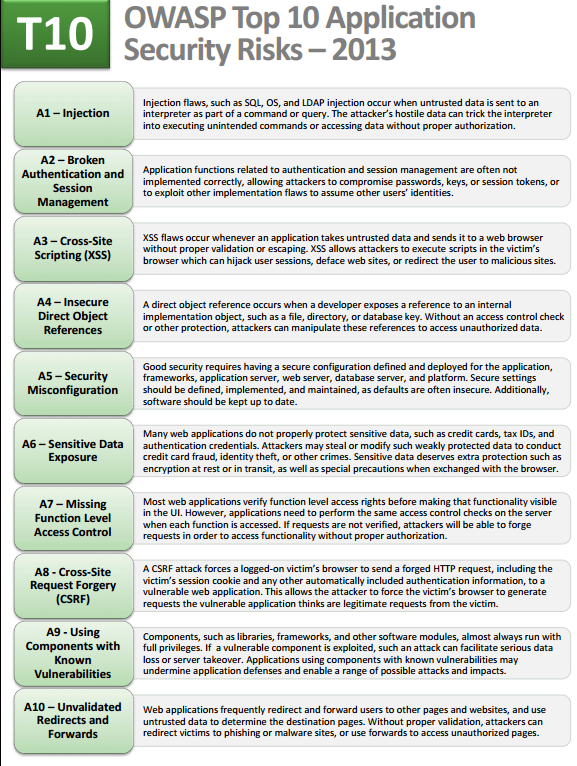
+ 2010 – A8 thiếu hạn chế truy cập bằng đường dẫn - Failure to Restrict URL Access trở thành 2013 – A7: Mất kiểm soát mức độ truy cập chức năng - Missing Function Level Access Control để bao phủ toàn bộ các vấn đề liên quan tới kiểm soát mức độ truy cập chức năng các ứng dụng, có rất nhiều các để chỉ ra chức năng nào có thể được truy cập, không chỉ bằng đường dẫn URL

1. OWASP gộp và mở rộng 2010-A7 & 2010-A9 để tạo nên:

2013-A6: Thông tin nhạy cảm dễ bị bóc trần ra ngoài - Sensitive Data Exposure. Chủ đề mới này bao phủ tất cả các cách bảo vệ thông tin nhạy cảm từ người dùng gửi và lưu trữ trong các ứng dụng, sau đó gửi trở lại cho trình duyệt.

1. OWASP thêm vào 2013-A9: Sử dụng các thành phần có thể bị tấn công dễ dàng - Using Known Vulnerable Components

# Danh sách Top 10 rủi ro ứng dụng nhất năm 2013



# Tìm hiểu về tiêm nhiễm mã độc(Injection) và sai lầm trong kiểm tra định danh (broken authenticate and session management):

## A1 - Tiêm nhiễm mã độc (Injection)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Đối tượng nguy hiểm {threat agent} | Khả năng tấn công: | Sự phổ biến | Khả năng phát hiện | Tác động đến kỹ thuật | Tác động đến kinh doanh |
| ? | Dễ |  |  | Nhiều | ? |
|  |  | Trung bình | Trung bình |  |  |
| Ai cũng có thể gửi các dữ liệu khác nhau đến hệ thống: người dùng bên ngoài, nội bộ, và quản trị viên | Kẽ tấn công có thể gửi những mã dạng văn bản tận dụng sơ hở trong cú pháp của trình biên dịch. Hầu như bất cứ đoạn mã nguồn nào cũng có thể mắc lỗi bao gồm cả những mã nguồn nội bộ | Lỗi nhúng mã có thể sảy ra khi một ứng dụng gửi những đoạn mã độc đến trình biên dịch. Lỗi này rất phổ biến đặc biệt ở những đoạn mã nguồn được sử dụng nhiều, như trong truy vấn SQL, LDAP, XPATH, các câu lệnh hệ điều hành, các tham số của chương trình v.v…  Lỗi nhúng mã thường dễ phát hiện khi rà soát mã nguồn, nhưng khó phát hiện hơn khi chạy thử chương trình. Những công cụ quét kiểm tra có thể giúp kẻ tấn công tìm ra chúng. |  | Lỗi này có thể khiến bạn mất hoặc hư hỏng hoặc không thể kiểm tra hay truy cập đến dữ liệu.  Nặng hơn nó có thể dẫn đến toàn bộ hệ thống bị chiếm quyền | Ta cần cân nhắc đến giá trị của những dữ liệu bị ảnh hưởng và các nền tảng đang chạy trình biên dịch. Tất cả dữ liệu có thể bị đánh cắp hoặc hủy hoại. Điều này có thể ảnh hưởng nghiêm trọng đến danh tiếng của bạn không ? |

### Tôi có bị mắc lỗi nhúng mã hay không ?

Cách tốt nhất để kiểm tra xem ứng dụng có mắc lỗi hay không là xác minh nếu tất cả các dữ liện đầu vào của trình biên dịch có thể phân biệt rõ ràng giữa mã độc và câu truy vấn hay câu lệnh. Ví dụ như trong truy vấn SQL, ta cần phải sử dụng tất cả những biến số trong những câu lệnh được chuẩn bị hay những thủ tục được lưu trữ, tránh sử dụng những câu truy vấn động.

Kiểm tra mã nguồn có thể là cách nhanh và chính xác để xác định nếu ứng dụng có an toàn hay không. Những công cụ phân tích mã nguồn có thể giúp nhân viên bảo mật hiểu được trình biên dịch và luồng dữ liệu đi trong ứng dụng. Những chuyên viên thâm nhập có thể xác thực những vấn đề tìm ra bằng cách tạo những chương trình thâm nhập thử vào hệ thống

Chương trình quét tự động có thể đưa ra những thông tin sâu sắc nếu chương trình của bạn thực sự mắc lỗi nhúng mã. Máy quét không thể luôn kiểm tra được trình biên dịch và khó biết chắc chắn nếu có một phương thức tấn công thành công. Những phần xử lý lỗi có thể giúp tìm ra lỗ hổng dể dàng hơn

### Làm cách nào để tôi có thể ngăn chặn lỗi nhũng mã

Để ngăn chặn cần phải phân biệt được mã độc và câu lệnh hay truy vấn

1. Tùy chọn thường được dùng là sử dụng một thư viên API an toàn, nó giúp bạn tránh việc sử dụng trình biên dịch một cách trực tiếp và cung cấp một giao diện phân tách tham số. Cần cẩn thận với APIs, như thủ tục lưu trữ dù đã phân tách tham số nhưng lỗi nhúng mã vẫn có thể sảy ra ở bên trong
2. Nếu không có những thư viện phân tách tham số, bạn có thể cẩn thận loại bỏ những kí tự đặc biệt bằng những công cụ của trình biên dịch đó. OWASP’s ESAPI có cung cấp thông tin về những công cụ đó.
3. Chúng tôi khuyến khích kiểm tra những dữ liệu đầu vào hợp lệ bằng các từ điển tiêu chuẩn {canonicalization} phù hợp. Đây vẫn chưa phải là cách phòng thủ an toàn nhất vì nhiều ứng dụng sử dụng những ký tự đặc biệt trong các thông tin đầu vào. OWASP’s ESAPI cũng cung cấp 1 danh sách rất nhiều những thư viện công cụ để kiểm tra dữ liệu đầu vào hợp lệ

### Ví dụ về kịch bản tấn công

Ứng dụng sử dụng mã độc khi xây dựng truy vấn SQLsau:

String query = "SELECT \* FROM accounts WHEREcustID='" + request.getParameter("id") +"'";

Kẻ tấn công có thể thay thế tham số ‘id’ trong trình duyệt để gửi đến: ‘ or ‘1’=’1. Việc này thay đổi ý nghĩa của câu truy vấn và trả ra giá trị của tất cả các tài khoản trong cơ sở dữ liệu thay vì chỉ của 1 nhân viên mà thôi

http://example.com/app/accountView?id=' or '1'='1

Trong trường hợp xấu nhất, kẻ tấn công có thể sử dụng điểm yếu này để thực thi những thủ tục lưu trữ trong cơ sở dữ liệu và giúp hắn chiếm quyền điều khiển cơ sở dữ liệu hoặc toàn bộ máy chủ

## A3 – Sai lầm trong kiểm tra định danh

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Đối tượng nguy hiểm {threat agent} | Khả năng tấn công: | Sự phổ biến | Khả năng phát hiện | Tác động đến kỹ thuật | Tác động đến kinh doanh |
| ? |  |  |  | Nặng | ? |
|  | Trung bình | Trung bình | Trung bình |  |  |
|  |  |  |  |  |  |
| Có thể là những kẻ tấn công nặc danh, hoặc người người dùng sở hữu tài khoản muốn ăn cắp thông tin từ tài khoản khác. Cũng nên cân nhắc những đối tượng nội bộ muốn che dấu hành động | Kẽ tấn công lợi dụng những sơ hở trong các đoạn mã kiểm tra định danh (ví dụ: xem tài khoản, password, ID phân làm việc) để giả mạo người dùng | Người phát triển thường xuyên tự xây dựng những mô hình chứng thực và quản lý các phiên làm việc, nhưng xây dựng đúng cách thì không dễ.  Kết quả là những cái tùy chọn đó thường xuyên có lỗi trong nhiều phần như đăng xuất, quản lý password, thời gian chờ, ghi nhớ người dùng, câu hỏi bí mật, cập nhật tài khoản v.v. Tìm những lỗi này có thể rất khó khăn bởi vì các cách triển khai có thể độc lập, không giống nhau. |  | Lỗi như thế có thể cho phép một vài hoặc toàn bộ tài khoản bị tấn công. Khi đã thành công, kẻ tấn công có thể làm nhiều thứ dưới quyền của nạn nhân. Những tài khoản đặc quyền thường là mục tiêu chính | Ta cần cân nhắc đến giá trị của những hệ thống và dữ liệu bị ảnh hưởng.  Cũng nên cân nhắc đến tác động kinh tế của việc lỗi này bị trình bày trước công luận |

### Tôi có bị mắc lỗi không ?

Thông tin tài khoản và phiên làm việc là những thứ cần được bảo vệ:

1. Những thông tin tài khoản có được bảo vệ khi lưu trữ dưới dạng băm hoặc mã hóa không ? Xem A7

2. Thông tin có thể bị đoán ra hoặc thay thế vì những điểm yếu trong các chương trình quản lý tài khoản hay không ? (tạo tài khoản, đồi password, lấy lại password, điểm yếu trong id phiên làm việc)

3. ID phiên làm việc có hiển thị trong URL không

4. ID phiên làm việc có thể bị giả mạo bằng các phương pháp định hình phiên làm việc hay không

5. Id phiên làm việc có thời gian chờ không và người dùng có thể đăng xuất không ?

6. Id phiên làm việc có thay đổi sau khi người dùng đăng nhập lại không

7. Password, Id phiên làm việc và những thông tin khác có được gửi đi sử dụng TLS connection không ? Xem thêm A9

Xem ASVS phần những yêu cầu V2 và V3 để biết thêm chi tiết

### Làm cách nào để ngăn chặn ?

Phương pháp được khuyến khích cho một tổ chức là cho phép người phát triển truy cập:

1. Một tập hợp những phương thức quản lý định danh mạnh. Nó cho phép:

a) Đáp ứng tất cả những yêu cầu của kiểm tra định danh được định nghĩa trong OWASP’s Application Security Verification Standard (ASVS) phần V2 (chứng thực) và V3 (quản lý phiên làm việc)

b) Có một giao diện đơn giản cho người phát triển. Hãy xem xét ESAPI Authentication and User APIs như một ví dụ điển hình cho việc sử dụng, phát triển

2. Cũng nên bỏ nhiều công sức để tránh bị ăn cắp phiên làm việc qua lỗi XSS , Xem thêm A2

### Vi dụ về kịch bản tấn công:

Kịch bản 1: Ứng dụng đặt vé máy bay cho phép viết lại URL, đặt id phiên làm việc trong URL

[**http://example.com/sale/saleitems;jsessionid=2P0OC2JDPXM0OQSNDLPSKHCJUN2JV?dest=Hawaii**](http://www.google.com/url?q=http%3A%2F%2Fexample.com%2Fsale%2Fsaleitems%3Bjsessionid%3D2P0OC2JDPXM0OQSNDLPSKHCJUN2JV%3Fdest%3DHawaii&sa=D&sntz=1&usg=AFQjCNFJF0guQBgdhdGlslFm99qbFpW4Og)

Một người dùng bình thường muốn cho người bạn của anh ta biết về việc bán vé. Anh ta email liên kết trên mà không biết anh ta đang gửi id phiên làm việc của mình. Người bạn của anh ta sử dụng liên kết đó có thể sử dụng phiên làm việc và cả thẻ tín dụng

Kịch bản 2: Thời gian chờ của ứng dụng không được chỉnh hợp lý. Người dung sử dụng một máy công cộng để truy cập vào trang web. Thay vì chọn “đăng xuất” anh ta chỉ đóng trình duyệt và đi. Một người khác sử dụng cùng trình duyệt đó vài giờ sau vẫn có thể sử dụng phiên làm việc đó

Kịch bản 3: Người nội bộ hoặc kẻ tấn công có quyền truy cập đến cơ sở dữ liệu lưu trữ password. Password không được mã hóa cho phép kẻ tấn công ăn cắp được những tài khoản đó