**A9. Tấn công sử dụng các thành phần với các lỗ hổng đã biết (Using Components with Known Vulnerabilities).**

Hầu như tất cả các phần của phần mềm hiện nay đều sử dụng các thành phần bên ngoài hoặc thư viện. Nó mang lại lợi thế đáng kể cho người lập trình về thời gian và công sức.

Các thành phần hợp thành như thư viện, framework, và các module phần mềm khác, hầu như luôn chạy với full đặc quyền. Nếu một thành phần lỗ hổng được khai thác, một cuộc tấn công như vậy có thể dễ dàng làm mất mát dữ liệu quan trọng.

Kẻ tấn công sử dụng các thành phần với các lỗ hổng được biết đến đó, có thể làm suy yếu hệ thống phòng thủ của ứng dụng và cho phép một loạt các cuộc tấn công có thể xẩy ra.

**Xác định vấn đề**

Không có phần mềm nào là hoàn hảo, và mặc dù các tổ chức phân phối các thành phần này làm việc chăm chỉ để làm cho sản phẩm tốt, họ nhận thức được hậu quả khi có những sai lầm xảy ra. Khi sử dụng các thư viện bên ngoài trong dự án của bạn, điều quan trọng là phải luôn nhớ rằng bạn đang mang lại chức năng tốt và các tính năng quan trọng, nhưng bạn cũng đang mở cửa cho các lỗi và lỗ hổng bảo mật tiềm năng.

Một ví dụ gần đây của điều này là lố hổng ‘remote-code execution with Expression Language injection' mà gần đây đã được giới thiệu thông qua hội nghị Spring Framework for Java. Lỗ này đã được fix ngay sau khi nó được phát hiện và một phiên bản vá lỗi của framework này đã được đưa ra. Tuy nhiên, người ta ước tính rằng 29.800.000 lượt tải chứa các lỗ hổng đã biết.

Ảnh hưởng kinh tế

kỹ thuật tấn công

Hacker xác định điểm yếu thông qua scan hoặc phân tích bằng tay.

Điểm yếu bảo mật

Phương pháp tấn công

Tác nhân đe dọa

Xem xét giá trị kinh doanh của những dữ liệu bị công khai trên những đường truyền dẫn về phương diện bảo mật, toàn viện và xác thực.

Hầu như mọi ứng dụng đều có những vấn đề bởi vì hầu hết các developer không chú trọng vào đảm bảo các thành phần, thư viện được cập nhật thường xuyên.

Những kẻ tấn công sử dụng các điểm yếu có thể khai thác được bao gồm injection, broken access control, XSS, etc…

Thư viện trong các framework có thể được xác định và khai thác bởi các tool tự động.

**Ví dụ kịch bản của một vụ tấn công**

Lỗ hổng thành phần (Component vulnerabilities) có thể gây ra hầu hết các loại rủi ro. Component gần như luôn luôn chạy với full quyền của các ứng dụng. Hai lỗ hổng thành phần dưới đây đã được tải về 22 triệu lần vào năm 2011.

* Apache CXF Authentication Bypass: Việc không cung cấp một identity token, kẻ tấn công có thể gọi bất kỳ dịch vụ web nào với đấy đủ sự cho phép (Apache CXF là một khung dịch vụ, không nên nhầm lẫn với các máy chủ Apache ứng dụng.)
* Spring Remote Code Execution: Việc lạm dụng Expression Language implementation trong Spring cho phép kẻ tấn công thực thi mã tùy ý, ảnh hwowgnr đến các máy chủ.

**Phát hiện và phòng chống**

* Hãy dành thời gian xem xét các tài liệu hiện có của các thành phần dự án của bạn cũng như các diễn đàn liên quan đến công cụ, ngôn ngữ … mà bạn đang sử dụng. Hầu hết các nhà cung cấp sẽ nhanh chóng thông báo cho người dùng khi các lỗ hổng được tìm thấy.
* Một trong những vấn đề lớn nhất của loại đe dọa này đến từ sự thiếu quan tâm của người sử dụng. Khi các lỗ hổng được phát hiện, bản cập nhật và sửa lỗi có xu hướng xuất hiện kịp thời, nhưng nhiều người dùng bỏ bê việc cập nhật của họ. Điêu này cho phép lỗ hổng đang tồn tại trong mã của họ, trong nhiều năm trong một số trường hợp.
* Một cách tiếp cận chủ động để tránh bị ảnh hưởng bởi lỗ hổng đã biết hoặc để ngăn chặn những lỗ hổng chưa được khám phá là bọc các tính năng bổ sung của các thành phần bên ngoài với code của bạn. Bằng cách này, bạn có thể kiểm soát những gì đi vào và ra khỏi các thành phần bên ngoài một cách hiệu quả.

**A10. Chuyển hướng và chuyển tiếp không an toàn (Unvalidated Redirects and Forwards)**

Việc chuyển hướng không an toàn người dùng đến một đường dẫn bên ngoài trang có thể tạo nguy cơ người dùng truy cập đến những trang chứa mã độc nhằm đánh cắp dữ liệu cá nhân.

Hãy xem xét các giá trị của việc duy trì lòng tin của người dùng.

Điều gì nếu họ bị nhiễm bởi phần mềm độc hại?

Điều gì nếu kẻ tấn công có thể truy cập các chức năng nội bộ?

kỹ thuật tấn công

bussiness impact

Ứng dụng thường xuyên chuyển hướng người dùng đến các trang khác. Đôi khi các trang đến được quy định trong một tham số thiếu chứng thực, cho phép kẻ tấn công để lựa chọn các trang đích.

Điểm yếu bảo mật

Phương pháp tấn công

Attacker sử dụng liên kết để chuyển hướng truy cập và lừa để nạn nhân click vào đó. Nạn nhân có nhiều khả năng nhấp vào nó, vì các liên kết là một trang web hợp lệ.

Tác nhân đe dọa

Bất kỳ ai có thể lừa người dùng bằng cách gửi một liên kết truy cập trên trang web của bạn.

Đổi hướng như vậy có thể cố gắng để cài đặt phần mềm độc hại hoặc lừa nạn nhân vào tiết lộ mật khẩu hoặc các thông tin nhạy cảm khác. chuyển tiếp không an toàn có thể cho phép vượt qua các phép kiểm soát truy cập.

**Ví dụ kịch bản của một vụ tấn công**

Kịch bản 1:

Ứng dụng này có một trang gọi là "redirect.jsp" mà có một tham số duy nhất có tên là"url". Kẻ tấn công tạo một URL độc hại để hướng người dùng đến một trang web độc hại mà thực hiện lừa đảo (phishing) và các phần mềm độc hại cài đặt. <http://www.example.com/redirect.jsp?url=evil.com>

Kịch bản 2:

Ứng dụng sử dụng chuyển tiếp (forward) để di chuyển yêu cầu (route request) giữa các bộ phận khác nhau của trang web. Để tạo điều kiện này, một số trang sử dụng một tham số để chỉ ra nơi người sử dụng phải được gửi nếu giao dịch thành công. Trong trường hợp này, kẻ tấn công tạo ra một URL sẽ vượt qua kiểm tra truy cập của ứng dụng và sau đó chuyển tiếp kẻ tấn công vào những chức năng rằng bình thường không thể truy cập được.

**Phát hiện**

Cách tốt nhất để tìm ra nếu một ứng dụng có bất kỳ một chuyển tiếp hay chuyển hướng không hợp lệ là:

1. Xem lại các mã nguồn có sử dụng các chuyển hướng hoặc chuyển tiếp (gọi là chuyển giao trong .NET.). Đối với mỗi lần sử dụng, xác định nếu các URL mục tiêu được chứa trong giá trị tham số nào không. Nếu có, các tham số này chỉ được chứa chỉ một điểm đến được cho phép.
2. Nhìn vào các thông số được cung cấp trước khi chuyển hướngđể xem chúng có xuất hiện như một URL đến hoặc một phần của URLnhư vậy. Nếu có, thay đổi URL đích và quan sát xem các trang web có chuyển hướng đến các đích mới.
3. Nếu mã chương trình không có sẵn, kiểm tra lại các thông số để xem chúng có phải là một phần của một trang chuyển hướng hoặc chuyển tiếp URL đích. Kiểm tra lại các thông số đó.

**Phòng tránh**

An toàn sử dụng các chuyển hướng và chuyển tiếp có thể được thực hiện trong một số cách sau đây :

1. Đơn giản chỉ cần tránh sử dụng chuyển hướng và chuyển tiếp.

2. Nếu sử dụng, tránh sử dụng tham số của người dùng cho việc xác định điểm đến. Điều này thường có thể thực hiện được.

3. Nếu việc sử dụng tham số cho điểm đến không thể tránh khỏi, đảm bảo giá trị của tham số là hợp lệ và đúng quyền của người dùng.