HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

THÁI THỊ MỸ HẠNH

NGHIÊN CỨU CÁC KỸ THUẬT KIỂM THỬ BẢO MẬT ỨNG DỤNG WEB

LUẬN VĂN THẠC SỸ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

THÁI THỊ MỸ HẠNH

NGHIÊN CỨU CÁC KỸ THUẬT KIỂM THỬ BẢO MẬT ỨNG DỤNG WEB

CHUYÊN NGÀNH : HỆ THỐNG THÔNG TIN

MÃ SỐ : 8.48.01.04

TÓM TẮT LUẬN VĂN THẠC SĨ

HÀ NỘI – 2020

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: PGS.TS. Lê Hữu Lập

Phản biện 1: ……

Phản biện 2: ……..

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ..... giờ ..... ngày ..... tháng .... năm .....

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông

MỞ ĐẦU

Ngày nay với cơ sở hạ tầng về mạng Internet phát triển rất mạnh mẽ, cùng với đó là sự ra đời của hàng hoạt các ứng dụng Web để đáp ứng nhu cầu của người sử trong mọi lĩnh vực của cuộc sống. Các ứng dụng web cho các dịch vụ khác nhau đã nhận được sự tin tưởng của khách hàng qua một thời gian dài. Hàng triệu triệu dữ liệu được tải và chia sẻ giữa các nền tảng khi mọi người cho rằng các giao dịch được giám sát an toàn. Tuy nhiên, khi các cuộc tấn công trên mạng tiếp tục gây ra sự hoang mang, nguy cơ bảo mật ứng dụng và dữ liệu của chúng ta trong lĩnh vực kỹ thuật số ngày càng tăng. Kiểm tra tính bảo mật của các ứng dụng web là rất quan trọng. Nó là một kỹ thuật kiểm thử phần mềm nhằm mục đích xác minh rằng chức năng của phần mềm có khả năng chống lại các cuộc tấn công và dữ liệu được phần mềm xử lý được bảo vệ. Để thiết lập các yêu cầu chung mà phần mềm phải đáp ứng, có các tiêu chuẩn bảo mật phần mềm. Luận văn này nhằm mục đích mô tả và áp dụng một quy trình cần thiết để xác minh tính bảo mật của một ứng dụng web.

Đề tài "**Nghiên cứu các kỹ thuật kiểm thử bảo mật ứng dụng web**" nhằm nghiên cứu và thử nghiệm các kỹ thuật nhằm phát hiện để ngăn chặn kịp thời các nguy cơ về an ninh, bảo mật ứng dụng web. Luận văn giúp hiểu các mối đe dọa đến từ đâu và cách để bảo vệ ứng dụng web của mình trước các cuộc tấn công phổ biến nhất.

Bố cục luận văn bao gồm phần mở đầu, phần kết luận và các chương nội dung được tổ chức như sau:

- **Chương 1**: Tổng quan về bảo mật website và các ứng dụng trên web. Chương này đưa ra các khái niệm tổng quan về bảo mật website, các ứng dụng trên web và tình trạng bảo mật website hiện nay.
- **Chương 2**: Các kỹ thuật kiểm thử bảo mật ứng dụng web. Chương hai tập trung vào việc đưa ra các khái niệm tổng quan về kiểm thử bảo mật website; các phương pháp, quy trình, kỹ thuật kiểm thử bảo mật ứng dụng web.
- **Chương 3**: Các công cụ kiểm thử và thực nghiệm kiểm thử bảo mật ứng dụng web.. Nội dung chương 3 trình bày về công cụ kiểm thử Zed Attack Proxy. Sau khi phân tích công cụ đã trình bày, thực hiện kiểm thử bảo mật cho website https://duticrm.info và https://hack-yourself-first.com/

Mặc dù có nhiều cố gắng nhưng do thời gian hạn chế nên luận văn không tránh khỏi những khiếm khuyết. Kính mong các thầy cô và đồng nghiệp thông cảm và cho các ý kiến góp ý.

Xin trân trọng cảm ơn!

Chương 1. TỔNG QUAN VỀ BẢO MẬT VÀ CÁC LỖ HỔNG BẢO MẬT TRÊN WEB

Trong chương này, luận văn sẽ giới thiệu một cách tổng quan về bài toán bảo mật website, sự cần thiết bảo mật website, các lỗ hổng bảo mật và tình trạng tấn công các trang web hiện nay.

1.1 Tổng quan về bảo mật [1]

1.1.1 Bảo mật website

Bảo mật là bảo vệ an toàn thông tin trước những "tay" chuyên rình mò thông tin của người khác. Tùy thuộc vào các yêu cầu của mỗi hệ thống, mỗi hệ thống có những mục đích về bảo mật khác nhau, nhưng chúng đều có điểm chung là: Đảm bảo sự an toàn dữ liệu cho hệ thống và bảo vệ các tài nguyên trên mạng trước sự tấn công nhằm phá vỡ hệ thống hoặc sử dụng trái phép các tài nguyên của một số người có chủ ý xấu.

Quá trình phân tích bảo mật nên được chạy song song với phát triển ứng dụng Web. Nhóm lập trình viên và nhà phát triển chịu trách nhiệm phát triển mã cũng chịu trách nhiệm thực hiện các chiến lược khác nhau, phân tích sau rủi ro, giảm thiểu và giám sát.

1.1.2 Bảo mật ứng dụng web

Bảo mật ứng dụng web là một nhánh của bảo mật thông tin liên quan cụ thể đến bảo mật của các trang web, ứng dụng web và dịch vụ web. Ở cấp độ cao, bảo mật ứng dụng web dựa trên các nguyên tắc bảo mật ứng dụng nhưng áp dụng chúng cụ thể cho các hệ thống Internet và web.

Bảo mật ứng dụng web là quá trình bảo mật dữ liệu bí mật được lưu trữ trực tuyến khỏi sự truy cập và sửa đổi trái phép. Điều này được thực hiện bằng cách thực thi các biện pháp chính sách nghiêm ngặt. Các mối đe dọa bảo mật có thể ảnh hưởng đến dữ liệu được lưu trữ với mục đích xấu là cố gắng truy cập vào thông tin nhạy cảm.

1.2 Các lỗ hổng bảo mật [3], [4], [6]

Lỗ hổng bảo mật là tập hợp những điều kiện mà cho phép một kẻ xấu tấn công làm vi phạm những chính sách bảo mật một cách tương minh hoặc ngầm. Đó 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. “90% lỗ hổng bảo mật bắt nguồn từ ứng dụng web, 90% nhà quản trị chưa có cái nhìn tổng quan về bảo mật Web App”[3]. Đây là lý do dẫn tới số lượng các cuộc tấn công trên mạng ngày càng nhiều.

1.2.1 A1: Injection

Injection là một kĩ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng trong việc kiểm tra dữ liệu nhập trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu để "tiêm vào" (inject) và thi hành các câu lệnh bất hợp pháp (không được người phát triển ứng dụng lường trước). Các lỗ hổng đặc trưng: SQL injection; OS injection; LDAP injection; NoSQL injection…

1.2.2 A2: Lỗi xác thực (Broken Authentication)

Đây là nhóm các vấn đề có thể xảy ra trong quá trình xác thực, quản lý phiên đăng nhập.

- Những điểm yếu gây ra lỗi xác thực như:

  - Ứng dụng cho phép một công cụ tự động gửi nhiều yêu cầu để đăng nhập, sau đó những kẻ tấn công sẽ dùng công cụ để quét tên người dùng và mật khẩu để tìm ra cặp Tên người dùng/mật khẩu có trong ứng dụng.
  - Ứng dụng cho phép sử dụng những mật khẩu yếu hoặc vô ý chưa xóa những cặp Tên người dùng/mật khẩu mặc định.
  - Tính năng quên mật khẩu nhưng thiếu an toàn với những câu hỏi dạng kiến thức.

1.2.3 A3: Rò rỉ dữ liệu nhạy cảm (Sensitive Data Exposure)

Những ứng dụng web không quản lý, bảo vệ thông tin nhạy cảm một cách đúng đắn và để rò rỉ những thông tin như tài chính, sức khoẻ của khách hàng, thông tin cá nhân, thông tin tài khoản,...Hoặc đôi khi việc chia sẻ những tài liệu cá nhân cho bên thứ 3 sẽ thuộc loại quy phạm quy định pháp luật tuỳ vào quy định của từng quốc gia. Mức độ ảnh hưởng: phụ thuộc vào thông tin bị lộ ra ngoài.

1.2.4 A4: Tấn công thực thể bên ngoài XML (XML External Entities- XEE)

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.

Nhiều vấn đề XXE công khai đã được phát hiện, bao gồm cả tấn công các thiết bị nhúng. XXE xảy ra ở rất nhiều nơi bất ngờ, bao gồm cả các phụ thuộc lồng nhau sâu sắc. Cách dễ nhất là tải lên tệp XML độc hại, nếu được chấp nhận.

1.2.5 A5: Kiểm soát truy cập bị hỏng (Broken Access Control)

Lỗi liên quan đến việc kẻ tấn công có thể sửa chữa để chiếm quyền của người khác. Lỗi này tập trung vào những chức năng liên quan đến việc quản lý quyền hạn (như Quản trị viên hay người dùng)

Việc hạn chế những thứ mà người dùng đã đăng nhập mới được xem hoặc được làm trong ứng dụng nếu hoàn thành không chính xác sẽ bị những kẻ tấn công lợi dụng mà không cần tới việc đăng nhập.

1.2.6 A6: Lỗi cấu hình (Security Misconfiguration)

- Khi cấu hình chưa đủ hoặc chưa đúng sẽ dẫn đến hậu quả liên quan đến bảo mật. Kẻ tấn công sẽ khai thác những lỗ hổng đó trong phần cấu hình để tấn công vào hệ thống:
  - Không cập nhật những lỗ hổng liên quan đến bảo mật trên server dẫn đến tồn tại những lỗ hổng (Ví dụ một ứng dụng được phát hành từ tháng 1. Trong khoảng thời gian từ tháng 1 đến tháng 9, nhà phát hành có thể đưa ra nhiều bản vá những lỗ hổng bảo mật nhưng chưa cập nhật trên server nên sẽ bị kẻ tấn công khai thác).
  - Thư mục, tệp được cấu hình chưa đúng, chưa hợp lý. Ví dụ như một thư mục cá nhân nhưng cấu hình theo chế độ công khai.
  - Bật những dịch vụ không cần thiết, những dịch vụ này không sử dụng đến nên sẽ quản lý không tốt. Kẻ tấn công có thể tấn công vào những dịch vụ đó thay vì trực tiếp vào những dịch vụ đang chạy.

1.2.7 A7: Lỗ hổng Cross Site Scripting- XSS

XSS là một thuật ngữ được sử dụng để mô tả một lớp các cuộc tấn công cho phép kẻ tấn công chèn các tập lệnh phía máy khách thông qua trang web vào trình duyệt của những người dùng khác. Kẻ tấn công chèn các đoạn mã JavaScript vào ứng dụng web. Khi đầu vào này không được lọc, chúng sẽ được thực thi mã độc trên trình duyệt của người dùng.

1.2.8 A8: Chuyển đổi cấu trúc dữ liệu không an toàn(Insecure Deserialization)

Quá trình khôi phục lại dữ liệu, thông tin như ban đầu không an toàn, tạo lỗ hổng cho những kẻ tấn công khai thác.

1.2.9 A9: Sử dụng các thành phần có lỗ hổng (Using Components with Know Vulnerabilities)

Ứng dụng thường kết hợp với các thư viện khác, hoặc những ứng dụng mã nguồn mở thì cài thêm plug-in,...Việc dùng những thành phần, thư viện, frameworks chứa sẵn những lỗ hổng sẽ

làm cho ứng dụng của bạn dễ bị khai thác hơn. Việc tận dụng những ứng dụng đã có và cộng với một khối lượng code-base của nó khá lớn dễ dẫn đến bạn không hiểu và mất kiểm soát hay tệ hơn là có cả nguy cơ bảo mật bên trong những thư việc này.

1.2.10 A10: Không ghi nhật ký và giám sát đầy đủ (Insufficient Logging & Monitoring)

Trang web dễ bị lỗi nếu không có tính năng ghi nhật ký sự kiện lại. Những kẻ tấn công dựa vào việc thiếu giám sát và phản ứng kịp thời để đạt được mục đích mà không bị phát hiệ

1.3 Tình trạng tấn công các trang web hiện nay

Trong quý 3 năm 2019 vừa qua, hệ thống CyStack Attack Map đã ghi nhận 127.367 website bị tấn công và chiếm quyền điều khiển. Như vậy, số website này đã giảm 27% so với con số 175.451 website bị tấn công trong quý 2. Cụ thể trong quý 3, số lượng website tháng 7, 8 và 9 lần lượt là 38.385, 44.848 và 44.134, giảm hơn so với mức trung bình 42.483 website/tháng của quý trước.

Tên miền .com phổ biến vẫn là đối tượng được nhắm đến nhiều nhất bởi các hacker, sau đó là tên miền .net với 5,99%. Ngoài ra còn có tên miền đặc trưng của các quốc gia như: .in (Ấn Độ), .ua (Australia), .id (Indonesia), .br(Brazil), .ru (Nga), .vn (Việt Nam), … (Hình 1.4).

Trước những diễn biến khó lường của các cuộc tấn công mạng cùng tốc độ phát triển không ngừng nghỉ của Internet mỗi doanh nghiệp tổ chức cần phải có những kế hoạch cụ thể để triển khai hệ thống bảo mật cũng như hệ thống ứng cứu sự cố của mình nhằm giảm thiểu rủi ro thiệt hại do mất an toàn thông tin gây nên.

1.4 Kết chương

Có thể nói vấn đề bảo mật website và các ứng dụng là hết sức hệ trọng và cần thiết đối với mỗi đơn vị doanh nghiệp và cá nhân khi thiết lập website của mình. Có rất nhiều lỗ hổng bảo mật website để nảy sinh ra các nguy cơ tấn công của tội phạm ngày càng gia tăng.

Trong chương 2, luận văn sẽ đi sâu trình bày về các kỹ thuật kiểm tra thử nghiệm về độ bảo mật của các website và ứng dụng trên nó.

Chương 2. CÁC KỸ THUẬT KIỂM THỬ BẢO MẬT ỨNG DỤNG WEB

Trong chương này luận văn tập trung trình bày về các kỹ thuật kiểm thử bảo mật ứng dụng web.

2.1 Kiểm thử bảo mật ứng dụng web [2], [7]

Trong những năm gần đây, các ứng dụng web có xu hướng trở thành phổ biến. Một số lượng lớn các giao dịch điện tử bao gồm thương mại điện tử, ngân hàng điện tử, học tập điện tử, các hoạt động của chính phủ ... có thể được tiến hành trực tuyến bất cứ lúc nào và bất cứ nơi đâu. Do vậy việc đảm bảo an toàn thông tin trên môi trường internet là một vấn đề vô cùng quan trọng.

2.1.1 Ứng dụng web

Ứng dụng là một loại chương trình có khả năng làm cho máy tính thực hiện trực tiếp một công việc nào đó người dùng muốn thực hiện.

Ứng dụng web là chương trình máy tính sử dụng trình duyệt web và công nghệ web để thực hiện các tác vụ qua Internet. Thông qua ứng dụng web, người dùng có thể thực hiện một số công việc: tính toán, chia sẻ hình ảnh, mua sắm …

2.1.2 Kiểm thử bảo mật ứng dụng web

Kiểm thử bảo mật là một quá trình để xác định xem hệ thống có bảo vệ dữ liệu và duy trì chức năng như dự định hay không, kiểm tra xem dữ liệu bí mật có được giữ bí mật hay không và người dùng chỉ có thể thực hiện những nhiệm vụ mà họ được phép thực hiện (Ví dụ: người dùng sẽ không thể thay đổi chức năng của ứng dụng web theo cách không có chủ ý, ...).

Khi kiểm tra một chức năng tức là đang cố chứng minh rằng một tính năng hoạt động cho người dùng cuối - thực hiện những gì như mong đợi và không cản trở việc hoàn thành nhiệm vụ. Người kiểm thử có thể sẽ ưu tiên tập trung vào các tính năng được sử dụng thường xuyên hơn, được sử dụng bởi nhiều người dùng hơn, được coi là quan trọng nhất,... Là người kiểm thử bảo mật, người dùng cuối bây giờ là kẻ tấn công đang cố gắng phá vỡ ứng dụng. Mục tiêu thử nghiệm là chứng minh rằng một kịch bản tấn công cụ thể không thành công với bất kỳ kịch bản tấn công nào.

Kiểm thử bảo mật ứng dụng web là một quy trình tổng quan bao gồm vô số các quy trình cho phép kiểm tra bảo mật của ứng dụng Web, là quá trình kiểm tra, phân tích và báo cáo về mức độ bảo mật của ứng dụng Web.

2.2 Phân loại kiểm thử bảo mật

2.2.1 Kiểm thử yêu cầu và thiết kế

Bất kỳ hệ thống nào cũng được xây dựng từ một tập hợp các yêu cầu. Đôi khi những yêu cầu này được viết một cách rõ ràng, nhưng thường chúng là những phát biểu mập mờ không được định nghĩa rõ ràng. Kiểm thử bảo mật ở giai đoạn yêu cầu thiết kế cho ứng dụng web chính là việc xem xét các yêu cầu thiết có đã mô tả rõ các tiêu chí cụ thể về yêu cầu bảo mật cho ứng dụng web hay chưa.

2.2.2 Kiểm thử mã nguồn

Phương pháp kiểm tra độ bảo mật của ứng dụng thông qua mã nguồn của ứng dụng. Phương pháp kiểm thử này chủ yếu dùng để xác định sự an toàn của thuật tốn được dùng trong ứng dụng, xác độ nguy cơ rò rỉ thông tin, nguy cơ bị tấn công chiếm quyền kiểm sốt thông qua mã nguồn.

2.2.3 Kiểm thử các thiết lập của trình duyệt

Các thiết lập của trình duyệt có thể được cài đặt trong các trình duyệt như Mozilla FireFox và Microsoft Internet Explorer cho phép giới hạn truy cập đến các nội dung Internet có thể gây hại. Người sử dụng sẽ thường có các chỉnh sửa các thiết lập này. Hơn nữa, có một sự thay đổi lớn phía người sử dụng về khả năng làm chủ các thiết lập này. Những người sử dụng Web ngày càng được đào tạo nhiều hơn cách sử dụng các thiết lập để bảo vệ chính họ. Vì vậy, chúng ta cần phải kiểm thử nhiều sự kết hợp của các thiết lập.

2.2.4 Kiểm thử tường lửa

Kiểm thử tường lửa nhằm nhận biết các hiệu ứng về chức năng được tạo ra bởi sự chuyển dữ liệu qua các mạng khác nhau. Cần nhắc lại rằng nhóm kiểm thử phần mềm không chịu trách nhiệm kiểm thử sự hiệu quả của các tường lửa và sự cấu hình chúng.

2.3 Quy trình kiểm thử bảo mật [3]

Một trong những quy tắc bảo mật quan trọng nhất là biết rõ về môi trường của bạn. Giai đoạn này sẽ xác định phạm vi (sẽ kiểm tra hệ thống nào; kế hoạch và mục tiêu cần đạt được với kiểm tra thâm nhập) và các tài nguyên và công cụ (máy quét lỗ hổng hoặc công cụ kiểm tra thâm nhập) để sử dụng để thực hiện kiểm tra.

2.3.1 Giai đoạn khám phá

Đây là giai đoạn thu thập các thông tin về các hệ thống nằm trong phạm vi thử nghiệm bảo mật. Thu thập càng nhiều thông tin về website mục tiêu càng tốt về hệ thống, thông tin về tài khoản gồm tên và mật khẩu (nếu tìm được).

2.3.2 Đánh giá lỗ hổng

Sau khi bản thiết kế hoàn thiện, phần kỹ thuật bắt đầu, nơi các thành phần được xác định để phát triển. Nó có thể là ngôn ngữ mã hóa, nền tảng, công nghệ stack,… Mỗi thành phần đi kèm với tập hợp các điểm yếu và điểm mạnh của nó, do đó điều quan trọng là xác định các lỗ hổng trước giai đoạn code. Điều này giúp xác định các lựa chọn an toàn hơn và giảm đáng kể chi phí để sửa chúng.

Kiểm thử bảo mật có thể được chia thành các kỹ thuật như kiểm thử bảo mật thủ công, kiểm thử bảo mật tự động và có thể kết hợp cả kiểm thử bảo mật thủ công & tự động. Bằng cách sử dụng các công cụ kiểm thử bảo mật tự động, không thể tìm thấy tất cả các lỗ hổng. Một số lỗ hổng có thể được xác định bằng cách sử dụng quét thủ công. Vì vậy, những người kiểm thử bảo mật có kinh nghiệm sử dụng kinh nghiệm và kỹ năng của họ để tấn công một hệ thống bằng cách sử dụng các phương pháp kiểm thử bảo mật thủ công.

a. Kiểm thử thủ công

Các kỹ sư kiểm thử thủ công thực hiện các phương pháp sau:

- Thu thập dữ liệu
- Đánh giá tính dễ bị tổn thương
- Triển khai thực tế
- Chuẩn bị báo cáo

Quá trình tìm kiếm lỗi bảo mật trong mã nguồn của ứng dụng bằng phương pháp thủ công thì phải đòi hỏi người kiểm thử phải có một phương pháp kiểm thử và ra soát hợp lý. Bởi vì khối lượng tập tin cũng như nội dung trong các ứng dụng web là rất lớn, nếu như không có một phương pháp rà soát và đánh giá hợp lý thì sẽ tiêu tốn rất nhiều thời gian để phát hiện lỗi.

a. Kiểm thử tự động

Phương pháp kiểm thử tự động là quá trình các công cụ sẽ thực hiện tự động quét thư mục, tập tin của ứng dụng web và tự động xác định các điểm mà cần kiểm tra dữ liệu. Trên cơ sở đã xác

định các điểm kiểm tra công cụ sẽ thực hiện đệ trình các tập dữ liệu được định nghĩa sẵn và chờ sự phản hồi từ phía ứng dụng web để kiểm tra xem liệu ứng dụng đó có bị các lỗi bảo mật hay không.

Kiểm thử bảo mật tự động nhanh hơn, hiệu quả, dễ dàng và đáng tin cậy để kiểm tra lỗ hổng và rủi ro của website một cách tự động. Công nghệ này không yêu cầu bất kỳ kỹ sư chuyên gia nào, nó có thể được vận hành bởi người có ít kiến thức nhất về lĩnh vực này. Các công cụ quét bảo mật tự động rất tốt trong việc tìm kiếm các lỗ hổng phổ biến một cách nhanh chóng và có hệ thống.

2.3.3 Giai đoạn khai thác

Trong giai đoạn này, cố gắng khai thác các lỗ hổng được xác định trong giai đoạn trước (tức là giai đoạn khám phá) để có quyền truy cập vào hệ thống đích.

Với việc biết được chính xác port nào đang mở, dịch vụ nào đang chạy, điểm yếu hay lỗ hổng nào mà các dịch vụ này đang dính, chúng ta có thể tiến hành khai thác, đây chính là bước gần giống với một cuộc tấn công thực sự nhất.

2.3.4 Giai đoạn báo cáo

Đây là giai đoạn cuối cùng của quá trình kiểm thử. Mẫu báo cáo sẽ thường bao gồm các mục cơ bản sau:

- Tên lỗ hổng
- Mô tả nguy cơ của lỗ hổng
- Minh chứng về lỗ hổng
- Khuyến cáo khắc phục (có thể có hoặc không)

2.4 Các kỹ thuật kiểm thử bảo mật [5], [6], [8], [9]

Kiểm thử bảo mật có thể được chia thành hai loại: kiểm tra hộp trắng, hộp đen. Các phương pháp được sử dụng thường phụ thuộc vào tình hình và nhu cầu thử nghiệm.

2.4.1 Kiểm thử hộp đen [6], [12]

2.4.1.1. *Khái niệm*

Trong kỹ thuật Kiểm thử bảo mật hộp đen, người kiểm thử đánh giá hệ thống mục tiêu mà không có bất kỳ kiến thức nào về chi tiết hệ thống. Họ sẽ không xem xét bất kỳ đoạn code nào và không thu thập thông tin về hệ thống mục tiêu từ chủ sở hữu của hệ thống. Các kỹ thuật kiểm thử bảo mật hộp đen liên quan đến việc cố gắng phân tích và tìm lỗ hổng trong việc chạy phần mềm

bằng cách thao tác đầu vào mà không có bất kỳ kiến thức nào về mã nguồn. Vì vậy, người kiểm thử sẽ phát một cuộc tấn công toàn diện chống lại hệ thống để tìm ra điểm yếu hoặc lỗ hổng trong hệ thống.

2.4.1.2. *Kiểm tra bảo mật ứng dụng động DAST [11], [12]*

Kiểm tra bảo mật ứng dụng động (Dynamic Application Security Testing -DAST): Cách tiếp cận DAST liên quan đến việc tìm kiếm các lỗ hổng trong ứng dụng web mà kẻ tấn công có thể cố gắng khai thác.

Các công cụ DAST được coi như các công cụ kiểm thử mũ đen hoặc hộp đen, nơi mà những người kiểm thử không có hiểu biết về hệ thống. Họ dò tìm những lỗ hổng an ninh của một ứng dụng trong trạng thái chạy của nó. DAST phát hiện các lỗ hổng bằng cách thực sự thực hiện các cuộc tấn công.

- Một số công cụ DAST mã nguồn mở:

a. Zed Attack Proxy [7], [12]

Zed Attack Proxy (ZAP) là một công cụ nguồn mở được cung cấp bởi OWASP để thực hiện kiểm tra bảo mật. Nó giúp tìm ra các lỗ hổng bảo mật trong các ứng dụng. 8. Đây là một trong những công cụ bảo mật miễn phí phổ biến nhất thế giới và được các tình nguyện viên duy trì tích cực. Đây là một công cụ kiểm tra thâm nhập tích hợp dễ dàng để tìm một số lỗ hổng bảo mật trong một ứng dụng web trong khi đang phát triển và thử nghiệm một ứng dụng.

ZAP sẽ tiến hành thu thập dữ liệu ứng dụng web bằng trình thu thập dữ liệu của nó và quét thụ động từng trang mà nó tìm thấy. Sau đó, ZAP sẽ sử dụng trình quét để tấn công tất cả các trang, chức năng và tham số được phát hiện.

b. Nikto [12], [16]

Nitko là một trình quét máy chủ web nguồn mở thực hiện quét các máy chủ web để tìm các tệp/ chương trình nguy hiểm tiềm tàng, các phiên bản lỗi thời và các sự cố cụ thể của phiên bản khác. Nó cũng quét các cấu hình máy chủ như tùy chọn máy chủ HTTP và sẽ cố gắng xác định các máy chủ và phần mềm web đã cài đặt, giúp kiểm tra nhanh các vấn đề mà web server đối tượng đang gặp phải như: các cấu hình phía dịch vụ máy chủ hoặc phần mềm sai sót, chương trình hay file mặc định được tìm thấy, các chương trình hay file không an toàn được tìm thấy, những lỗ hổng cơ bản ở ứng dụng web.

Nikto cho phép người sử dụng tùy biến viết các thành phần và nhúng kết với Nikto để thực thi. Hơn nữa, Nikto cũng hỗ trợ nhiều định dạng của những chương trình quét lỗi bảo mật khác

như: Nmap, Nessus. Một điều bất tiện là Nikto không có giao diện đồ họa nên đòi hỏi người sử dụng phải có kiến thức lập trình và sử dụng các lệnh command.

c. Acunetix Web Vulnerability Scanner (WVS) [12], [13]

Acunetix WVS (Web Vulnerability Scanner) là chương trình tự động kiểm tra các ứng dụng Web để tìm kiếm các lỗ hổng bảo mật như SQL Injection, hay Cross-Site Scripting,… và tìm kiếm những chính sách đối với mật khẩu đăng nhập cũng như các phương thức xác thực vào Web Site. Acunetix WVS có thể tự động kiểm tra các lổ hỗng thông dụng và các mối nhạy cảm khác của những website có thể truy cập bằng trình duyệt, hay những ứng dụng được xây dụng trên các kỹ thuật tiên tiến như AJAX..

2.4.2 Kiểm thử hộp trắng [5], [12]

2.4.2.1. Khái niệm

Trong kỹ thuật Kiểm thử bảo mật hộp trắng, người kiểm thử truy cập hệ thống đích với các chi tiết đầy đủ về hệ thống. Vì đã có chi tiết đầy đủ về hệ thống nên kiểm thử hộp trắng có thể được thực hiện nhanh hơn nhiều so với kiểm nghiệm hộp đen. Các kỹ thuật kiểm tra bảo mật hộp trắng nhằm phân tích mã nguồn và kiến trúc ứng dụng từ quan điểm đảm bảo an ninh.

2.4.2.2. Kiểm tra bảo mật ứng dụng tĩnh SAST [11]

Kiểm tra bảo mật ứng dụng tĩnh (Static Application Security Testing- SAST): được thiết kế để phân tích mã nguồn và/ hoặc các phiên bản mã được biên dịch để giúp tìm ra các lỗi bảo mật.

Đối lập với DAST, những công cụ SAST có thể được xem như quá trình kiểm thử mũ trắng hay kiểm thử hộp trắng, nơi mà người kiểm thử biết thông tin về hệ thống và phần mềm được kiểm thử, bao gồm cả kiến trúc hệ thống, mã nguồn, … Những công cụ SAST kiểm tra mã nguồn để dò tìm và báo cáo những điểm yếu có thể dẫn đến những lỗ hổng an ninh.

- Một số công cụ SAST:

a. RIPS [14]

RIPS - RIPS Open Source là một bộ phân tích mã nguồn tĩnh cho các lỗ hổng trong các ứng dụng web PHP. RIPS là một công cụ được viết bằng PHP để tìm kiếm lỗ hổng trong ứng dụng PHP sử dụng việc phân tích code tĩnh. Bằng việc chia nhỏ và phân tích tất cả các file code, RIPS có khả năng để chuyển PHP source code vào một mô hình chương trình để phát hiện các hàm tiềm ẩn khả năng bị hỏng mà có thể bị bị phá hỏng bởi đầu vào của người dùng trong quá trình thiết kế. Ngoài việc cho phép tìm kiếm lỗi tự động, RIPS còn cung cấp khản năng phân tích code bằng tay.

b. AppScan Source [15]

IBM AppScan Source được phát hành bởi IBM – tập đoàn về máy tính có tuổi đời lớn nhất thế giới. AppScan Source có 2 phiên bản: Standard (dành cho doanh nghiệp vừa và nhỏ) & Enterprise (dành cho các tập đoàn lớn).

- Các chức năng chính:
  - Cung cấp kiến thức về bảo mật ứng dụng web
  - Scan ứng dụng web & mobile app để tìm lỗ hổng
  - Đề xuất phương án khắc phục
  - Xuất báo cáo riêng theo đặc trưng từng ngành

2.5 Đánh giá các kỹ thuật kiểm thử bảo mật

Các công cụ áp dụng kỹ thuật hộp trắng giúp cho người kiểm thử có thể phân tích được các lỗi tiềm ẩn, rủi ro của code từ đó giúp các lập trình viên dễ dàng tìm và sửa lỗi. Để có một website hoàn chỉnh đến tay người dùng thì ngoài việc coding còn phải tích hợp với các hệ thống cơ sở dữ liệu, cơ sở hạ tầng, máy chủ cũng như các thiết lập an toàn khác cho website. Do đó cần thực hiện thêm các kỹ thuật kiểm thử hộp đen bằng tay hoặc thông qua công cụ. Nhưng nếu áp dụng toàn bộ kỹ thuật kiểm thử hộp đen tức là đẩy cả giá trị hợp lệ, không hợp lệ vào thực hiện kiểm thử bảo mật sẽ mất thời gian (vì thực tế các giá trị hợp lệ đã được kiểm tra theo chức năng). Do đó sẽ ưu tiên lựa chọn công cụ phát triển dựa theo kỹ thuật kiểm thử hộp đen nhưng các giá trị đầu vào là không hợp lệ (phương pháp Fuzzing) để tìm các lỗi liên quan đến bảo mật của website.

Trên cơ sở phân tích về một số kỹ thuật và một số thuật toán được sử dụng trong các công cụ phân tích, dò quét lỗ hổng cho website, bằng việc so sánh ưu, nhược điểm của các công cụ này, tác giả xin đề xuất chọn sử dụng công cụ Zed Attack Proxy (ZAP) để thử nghiệm dò quét và đánh giá mức độ an toàn bảo mật của các website trong thực tế.

2.6 Kết chương

Kiểm thử bảo mật được phân chia theo nhiều lĩnh vực: thiết kế, mã nguồn, các thiết lập của trình duyệt, tường lửa. Quy trình kiểm thử bảo mật gồm bốn giai đoạn: Khám phá, Đánh giá lỗ hổng, Khai thác và Báo cáo. Có các kỹ thuật kiểm thử khác nhau, tuy nhiên kiểm thử hộp đen sẽ dò xét được lỗ hổng bảo mật đầy đủ.

Trong chương 3, luận văn sẽ trình bày sâu về công cụ kiểm thử hộp đen là Zed Attack Proxy đồng thời thực nghiệm kiểm thử trên trang web cụ thể.

Chương 3. CÁC CÔNG CỤ KIỂM THỬ VÀ THỰC NGHIỆM KIỂM THỬ BẢO MẬT ỨNG DỤNG WEB

Trong chương này, luận văn sẽ tập trung vào việc trình bày công cụ kiểm thử Zed Attack Proxy (ZAP), thực hiện kiểm thử và đánh giá kết quả thử nghiệm, bộ dữ liệu thử nghiệm, kết quả thử nghiệm cũng như đưa ra kết luận đánh giá về công cụ thử nghiệm.

3.1 Giới thiệu công cụ kiểm thử Zed Attack Proxy [7]

3.1.1 Tổng quan

Zed Attack Proxy thường được gọi là ZAP là một công cụ kiểm tra bảo mật nguồn mở cho một ứng dụng web được phát triển bởi OWASP, cung cấp nhiều tùy chọn để thực hiện kiểm tra thâm nhập bảo mật tự động hoặc thủ công cho các ứng dụng web.

Điểm nổi bật của công cụ ZAP:

- ZAP thu thập thông tin và phân tích các trang web bao gồm cả nội dung AJAX
- Có thể phân loại các lỗ hổng dựa trên mức độ nghiêm trọng
- Có khả năng tự xác minh các lỗ hổng
- Tạo ra các báo cáo và nhật ký chi tiết về các lỗ hổng
- Dễ sử dụng, dễ dàng để cài đặt
- Hỗ trợ kiểm thử bằng hình thức thủ công và tự động

3.1.2 Mô hình hoạt động

ZAP sẽ tiến hành thu thập dữ liệu ứng dụng web bằng trình thu thập dữ liệu của nó và quét thụ động từng trang mà nó tìm thấy. Sau đó, ZAP sẽ sử dụng trình quét để tấn công tất cả các trang, chức năng và tham số được phát hiện.

Các bước thực hiện gồm:

1. Khởi chạy công cụ quét web

2. Nhập Url của ứng dụng web sẽ được kiểm tra

3. Nhấp vào nút quét và đợi quá trình quét hoàn tất

4. Nếu quá trình quét được thực hiện thành công, một báo cáo sẽ được hiển thị cùng với kết quả.

3.2 Thực nghiệm kiểm thử bảo mật dựa Web dựa trên ZAP

3.2.1 Giai đoạn lập kế hoạch và khám phá

a. Chuẩn bị các thông tin về ứng dụng web cần kiểm thử và công cụ kiểm thử

- Thực hiện kiểm thử trên website: https://duticrm.info/ và https://hack-yourself-first.com/.
- Sử dụng công cụ kiểm thử Zed Attack Proxy để thực hiện quét các lỗ hổng bảo mật.

b. Thực nghiệm

Để chạy Quick Start, thực hiện các bước:

1. Khởi động ZAP và bấm vào tab **Quick Start** của Cửa sổ Workspace.

2. Nhấp vào nút **Automated Scan.**

3. Trong hộp văn bản URL để tấn công, hãy nhập URL đầy đủ của ứng dụng web bạn muốn tấn công. Ở ví dụ 1 nhập: https://duticrm.info/

Ngoài ra Trang web: https://duticrm.info/ có sử dụng ajax (có các tính năng như xử lý, tính toán, đăng ký, submit mà không load lại trang) nên sẽ lựa chọn thêm vào: **Use ajax spider.** Mục đích sử dụng giống như spider là để tìm kiếm link bằng cách giả lập thao tác trên màn hình. Từ đó sẽ tìm được link chuyển tiếp mà n mà spider thông thường không thể làm được.

4. Nhấp vào nút **Attack**

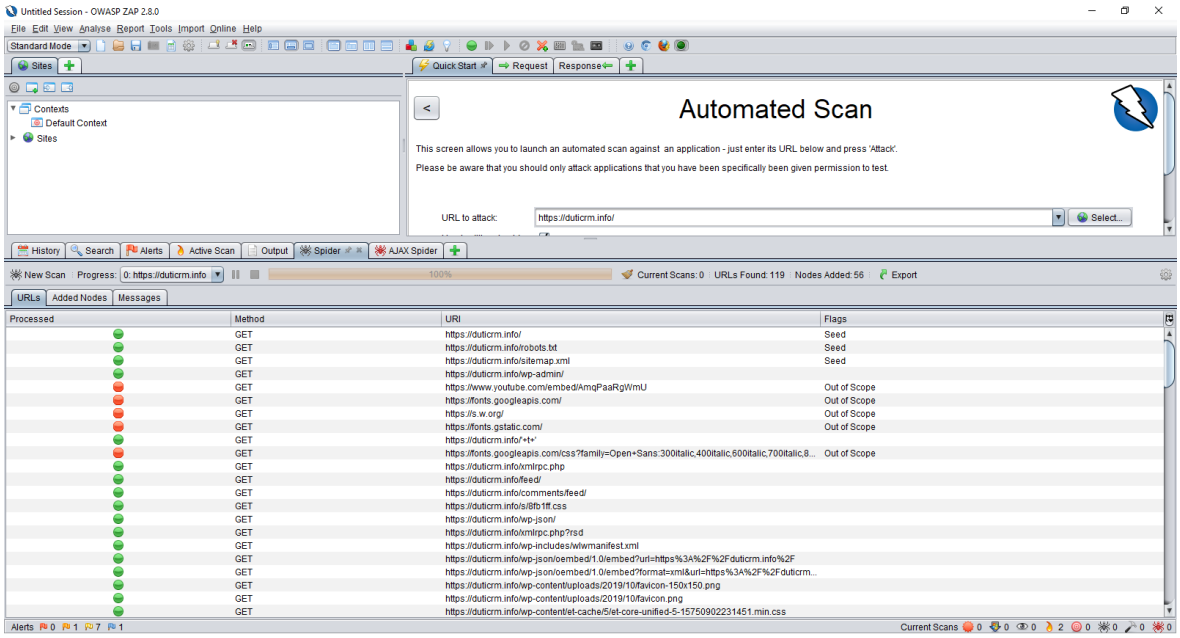
ZAP sẽ tiến hành thu thập dữ liệu ứng dụng web bằng trình thu thập dữ liệu của nó và quét chủ động từng trang mà nó tìm thấy. Sau đó, ZAP sẽ sử dụng trình quét để tấn công tất cả các trang, chức năng và tham số được phát hiện.

Quy trình tấn công phổ biến của OWASP ZAP được chia thành hai bước cơ bản: thu thập dữ liệu của ứng dụng web và cố gắng xâm nhập tấn công trang web. OWASP ZAP cung cấp hai mô-đun Spider và Attack để thực hiện các bước này:

- **Thu thập dữ liệu ứng dụng web bằng cách sử dụng OWASP ZAP Spider**

Mô-đun Spider của OWASP ZAP được sử dụng để quét nội dung của ứng dụng web bằng cách theo các liên kết và kiểm tra các mẫu URL phổ biến để trả về danh sách các URL có sẵn mà ứng dụng có. Bước này là bắt buộc vì mô-đun tấn công sẽ được khởi chạy sau này cần phải có danh sách URL đã biết để khởi chạy các cuộc tấn công. Bước thu thập thông tin được khởi chạy với các

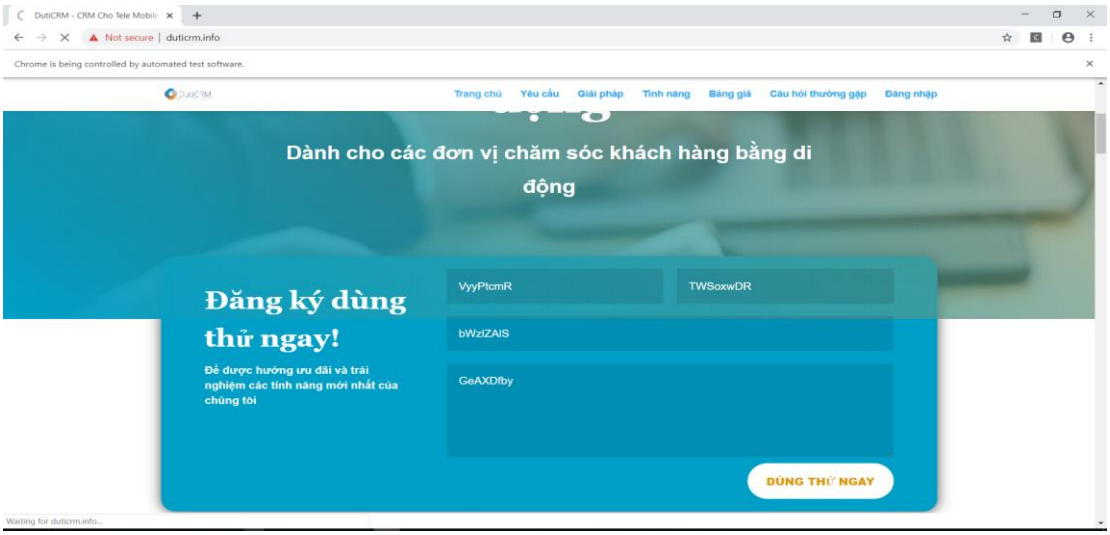
tùy chọn mặc định bằng cách sử dụng tùy chọn Attack->Spider. Sau khi khởi chạy Spider, OWASP ZAP sẽ biên dịch danh sách các URL đã được tìm thấy trong ứng dụng.



Hình 3.1. Giao diện mô-đun Spider

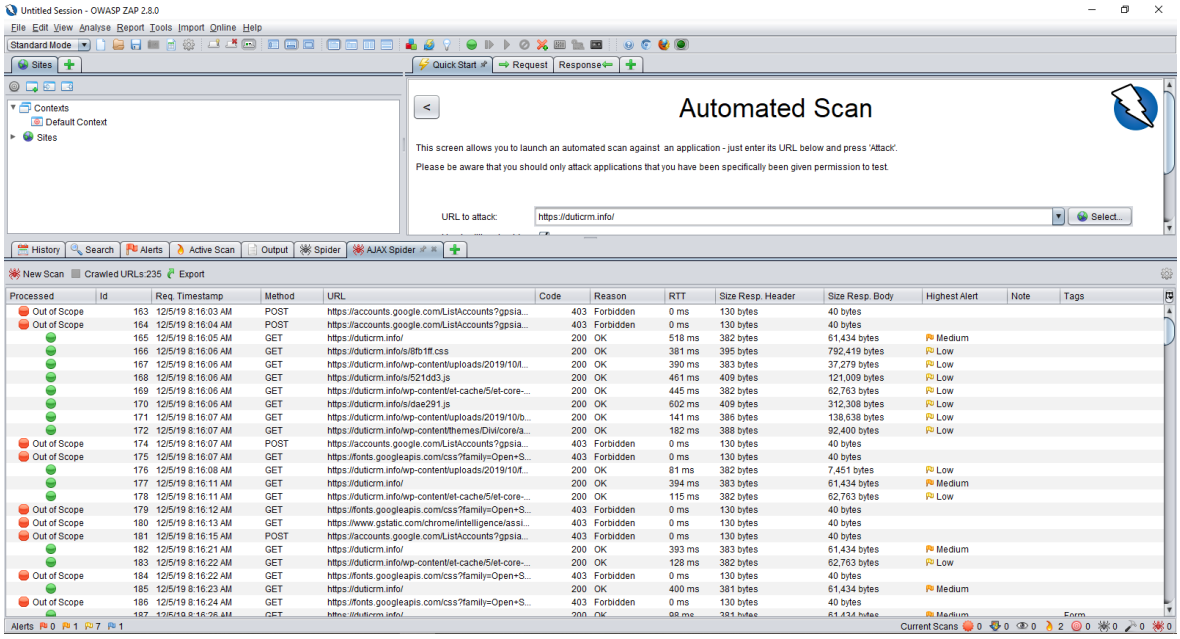
- **Thu thập dữ liệu trên ứng dụng web ở các tính năng liên quan đến AJAX bằng AJAX Spider**

  Lúc này phần mềm sẽ tự động giả lập thao tác trên màn hình như sau:



Hình 3.2: Thao tác tự động tại form Đăng ký dùng thử

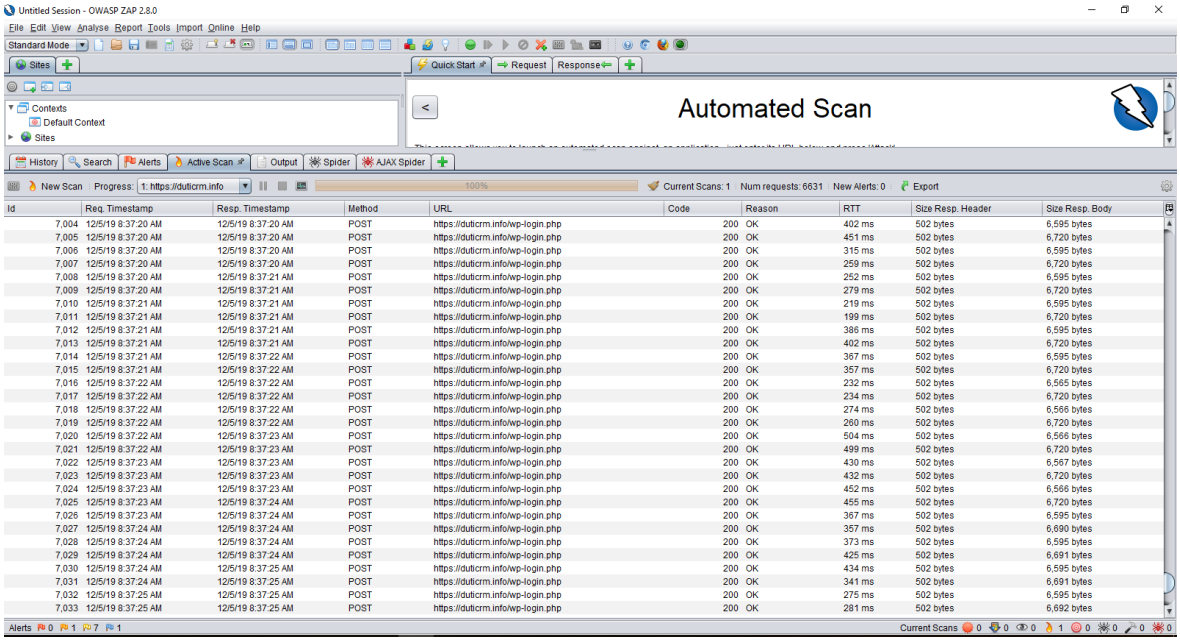
Sau đó danh sách các URL liên quan được tìm thấy trong ứng dụng được liệt kê ở mô đun AJAX Spider:



Hình 3.3: Mô đun AJAX Spider

- **Tấn công ứng dụng web bằng mô-đun Active Scan**

Mô-đun Active Scan của OWASP ZAP cho phép thực hiện các cuộc tấn công hoạt động thực sự để tìm và kiểm tra sự tồn tại của các lỗ hổng trong ứng dụng web cần kiểm thử:



Hình 3.4: Giao diện mô-đun Active Scan quét thành công các lỗ hổng bảo mật

Tiến trình của quá trình quét hoạt động có thể được theo dõi từ cửa sổ Progress Information.

Các lỗ hổng bảo mật sẽ được thống kê chi tiết ở tab Alerts, những Request có thể bị lỗi sẽ được liệt kê dưới chi tiết. Sau đó Đọc các thông tin lỗi, thực hiện tiến hành tái tạo lỗi. Có 4 màu thể hiện mức độ nguy hiểm từ cao đến thấp là Đỏ (Cao) ➞ Cam (Trung bình) ➞ Vàng (Thấp)➞ Xanh (Rất thấp):

Khi click vào mỗi lỗ hổng, nhìn vào cột bên cạnh ta có thể thấy được thông tin về lỗ hổng cũng như giải pháp sửa chữa lỗ hổng được đưa ra.

3.2.2 Đánh giá lỗ hổng

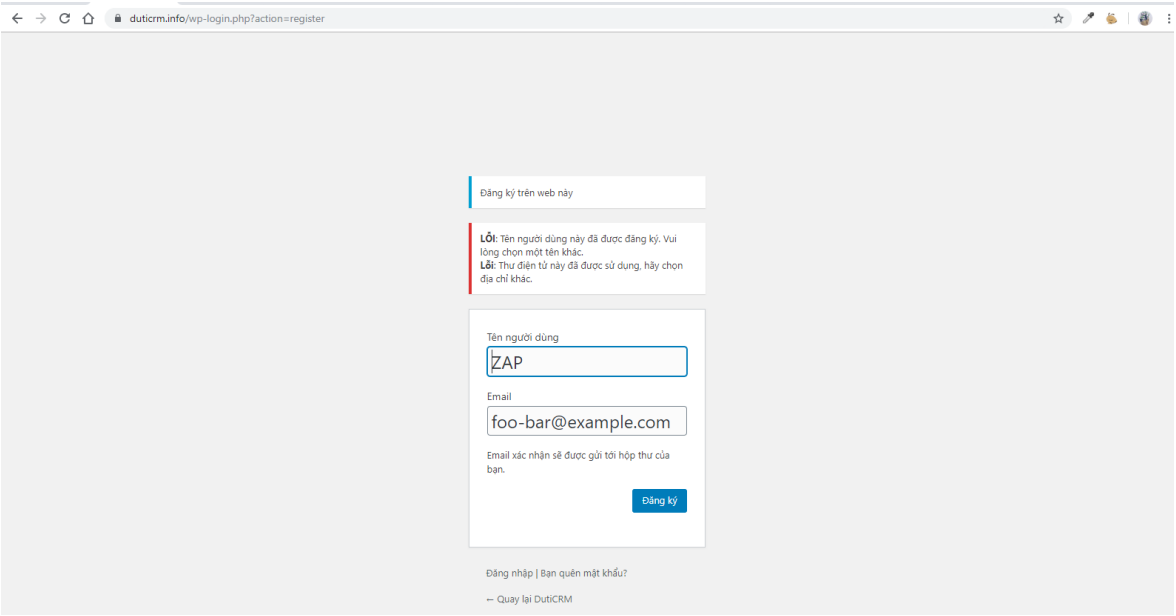
Ví dụ như trên tab Alert ta thấy có lỗi: Absence of Anti-CSRF Tokens, tiến hành phân tích, đánh giá lỗi này ta có các thông tin như sau:

- CSRF (Cross Site Request Forgery) là kĩ thuật tấn công bằng cách sử dụng quyền chứng thực của người sử dụng đối với 1 website khác. CSRF là kiểu tấn công hacker lợi dụng phiên làm việc của mình trên trình duyệt đó để thực hiện một hành động nào đó mà mình không biết. Hacker tạo một trang web giả mạo giống trang web mình vừa truy cập, chứa những thông tin giống form thật. Nhưng khi người dùng thực hiện các giao dịch ở đây sẽ bị mất thông tin.
- Một trong những kỹ thuật để phòng chống CSRF là sử dụng CSRF Token. Tạo ra một token tương ứng với mỗi form, token này sẽ là duy nhất đối với mỗi form và thường thì hàm tạo ra token này sẽ nhận đối số là"SESSION" hoặc được lưu thông tin trong SESSION. Khi nhận lệnh HTTP POST về, hệ thống sẽ thực hiện so khớp giá trị token này để quyết định có thực hiện hay không.

3.2.3 Giai đoạn khai thác

Giai đoạn này ta sẽ sử dụng những thông tin thu thập được ở trên cùng với kinh nghiệm của mình để tiến hành dò tìm và xác thực các lỗ hổng tồn tại trên hệ thống website. Tiếp lỗ hổng bên trên: Absence of Anti-CSRF Tokens, tiến hành tái tạo lại thủ công lỗ hổng trên ta được:

Tại trang đăng ký người dùng: nhập Tên người dùng = ZAP; Email = foo-bar@example.com, hiển thị thông báo lỗi như sau:



Hình 3.5: Giao diện Đăng ký người dung

Về kỹ thuật Anti-CSRF token: khi một người dùng gửi một form hoặc tạo các request được chứng thực yêu cầu một cookie, một mã token anti-CSRF nên được bao gồm trong request. Ứng dụng web sẽ xác nhận các token đã tồn tại và chính xác trước khi xử lý yêu cầu. Nếu token không có hoặc không đúng thì request sẽ bị từ chối. Nhìn từ phần Request thể hiện ở tool ZAP, khi gửi thông tin yêu cầu đăng nhập, không tìm thấy dòng nào chứa Anti-CSRF token

3.2.4 Giai đoạn báo cáo

Theo như thống kê các cảnh báo lỗ hổng từ Mô-đun Alert, ta có bảng Tổng hợp báo cáo như sau: Có 179 lỗ hổng được tìm thấy với 3 mức độ nguy hại: Medium (Trung bình), Low (Thấp), Informational (Rất thấp):

Bảng 3.1: Tổng hợp lỗ hổng bảo mật web: duticrm.info

No	Type	Number issues	Severity
1	X-Frame-Options Header Not Set	4	Medium
2	Absence of Anti-CSRF Tokens	23	Low
3	Cookie No HttpOnly Flag	25	Low
4	Cookie Without Secure Flag	16	Low
5	Incomplete or No Cache-control and Pragma HTTP	38	Low

	Header Set		
6	Cross-Domain JavaScript Source File Inclusion	1	Low
7	Web Browser XSS Protection Not Enabled	16	Low
8	X-Content-Type-Options Header Missing	55	Low
9	Charset Mismatch	1	Informational

Tương tự, thực hành trên trang: https://hack-yourself-first.com/. Thực hiện các bước như Ví dụ 1, ta có kết quả như sau:

Bảng 3.2: Tổng hợp lỗ hổng bảo mật web: https://hack-yourself-first.com/

No	Type	Number issues	Severity
1	Cross Site Scripting (Reflected)	1	High
2	Path Traversal	3	High
3	SQL Injection - Microsoft SQL Server	1	High
4	X-Frame-Options Header Not Set	49	Medium
5	Absence of Anti-CSRF Tokens	75	Low
6	Cookie No HttpOnly Flag	1	Low
7	Cookie Without SameSite Attribute	1	Low
8	Cookie Without Secure Flag	2	Low
9	Incomplete or No Cache-control and Pragma HTTP Header Set	52	Low
10	Secure Pages Include Mixed Content	2	Low
11	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	105	Low
12	Web Browser XSS Protection Not Enabled	52	Low
13	X-AspNet-Version Response Header Scanner	105	Low

14	X-Content-Type-Options Header Missing	102	Low
15	Information Disclosure - Suspicious Comments	3	Informational
16	Timestamp Disclosure - Unix	52	Informational

3.3 Đánh giá

- ZAP cung cấp cho các chuyên gia an ninh và kỹ sư kiểm thử phần mềm một loạt các tính năng tuyệt vời trong một gói đơn giản, liền mạch và rất mạnh mẽ:
  - Giao diện thân thiện dễ sử dụng. Phần mềm mã nguồn mở nên cập nhật thường xuyên bởi đội ngũ cộng tác viên trên thế giới.
  - Phù hợp với những người kiểm thử nghiệp dư đến chuyên nghiệp
  - Không cần có kiến thức sâu mà lập trình, hiểu biết chi tiết về hệ thống
  - Không yêu cầu truy cập vào mã nguồn nên có thể thực hiện nhanh chóng, thường xuyên
  - Có thể thực hiện quét tự động hoặc thủ công, tùy theo nhu cầu
  - Trong báo cáo có phân loại các lỗ hổng dựa trên các mức độ nghiêm trọng
  - Sử dụng được với bất kỳ ngôn ngữ lập trình nào và với cả các framework sẵn có hay đã tùy chỉnh.
- Ngoài một số điểm mạnh nêu trên, ZAP cũng có một số hạn chế như:
  - Nhiều loại lỗ hổng bảo mật rất khó tìm thấy tự động, chẳng hạn như các vấn đề xác thực, các vấn đề kiểm soát truy cập, sử dụng mật mã không an toàn,...
  - Quét bằng ZAP, dữ liệu có thể bị ghi đè hoặc tải trọng độc hại được đưa vào trang web thực hiện kiểm thử. Các trang web nên được quét trong một môi trường giống như thực tế nhưng không phải là thực tế (Ví dụ môi trường Pre-production) để đảm bảo kết quả chính xác trong khi bảo vệ dữ liệu ở môi trường thực tế.
  - Chỉ phù hợp với doanh nghiệp quy mô vừa và nhỏ

KẾT LUẬN

1. Những đóng góp của luận văn

Tấn công mạng là khá thường xuyên trong những thời điểm hiện tại. Việc tiến hành kiểm thử bảo mật web trên cơ sở quét lỗ hổng thường xuyên và kiểm tra thâm nhập để phát hiện các lỗ hổng là điều quan trọng và cấp thiết của mỗi cơ quan, doanh nghiệp và cá nhân đối với website của mình. Tần suất tiến hành kiểm tra xâm nhập phải phụ thuộc vào chính sách bảo mật của tổ chức. Tuy nhiên, tiến hành kiểm tra xâm nhập thường xuyên có thể xác định điểm yếu của hệ thống và giúp nó tránh các vi phạm an ninh. Một trong các cách là sử dụng các công cụ kiểm thử. Các công cụ kiểm thử cung cấp giao diện trực quan và đơn giản để xác định các lỗ hổng bảo mật cụ thể.

Luận văn đã thực hiện nghiên cứu về bảo mật website, kiểm thử bảo mật website đặc biệt đi sâu nghiên cứu các phương pháp kiểm thử cho website và sử dụng công cụ kiểm thử bảo mật Zed Attack Proxy (ZAP) để kiểm thử web: https://duticrm.info và https://hack-yourself-first.com/.

Cụ thể kết quả đạt được như sau:

- Luận văn đã trình bày về các vấn đề bảo mật website, kiểm thử bảo mật website. Đồng thời cũng trình bày các phương pháp kiểm thử website và các công cụ hiện có tương ứng. Thực hiện so sánh phân tích các ưu nhược điểm và khả năng áp dụng thực tế của của phương pháp và công cụ để lựa chọn công cụ áp dụng vào công việc kiểm thử bảo mật website thực tế.
- Tiếp theo luận văn trình bày việc sử dụng công cụ Zed Attack Proxy (ZAP) cho việc kiểm thử bảo mật website https://duticrm.info và https://hack-yourself-first.com/. Toàn bộ kết quả đạt được trong phần áp dụng thực nghiệm này là quá trình cố gắng của bản thân trong việc vận dụng lý thuyết, tìm tòi, nghiên cứu, học hỏi từ đồng nghiệp, bạn bè và một vài dự án trong thực tế.
- Kết quả nghiên cứu rất có ích cho việc kiểm thử bảo mật của website https://duticrm.info và https://hack-yourself-first.com/. Từ đó giúp tìm ra các lỗi, lỗ hổng bảo mật của website để kịp thời đưa ra các phương án khắc phục sớm đảm bảo website an toàn và không bị tấn công bởi tin tặc.

2. Hướng phát triển

Luận văn sử dụng nghiên cứu các phương pháp, công cụ kiểm thử ứng dụng cho kiểm thử bảo mật website.

Vì vậy hướng nghiên cứu tiếp theo là có thể xây dựng được một hệ thống/công cụ áp dụng các phương pháp kiểm thử khác nhau để đưa vào triển khai ở đơn vị mà không phải tốn thêm chi phí mua bản quyền của công cụ.