ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

KĨ NĂNG CHUYÊN NGHIỆP CHO KĨ SƯ

NHẬN BIẾT VÀ PHÒNG CHỐNG CÁC LOẠI HÌNH TẤN CÔNG SỬ DỤNG MÃ ĐỘC

Giảng viên hướng dẫn: **Thạc sĩ: thầy Nguyễn Cao Trí**

Sinh viên thực hiện: **Đoàn Trần Cao Trí - 2010733**
Nguyễn Hoàng Đắc Phú - 2010514
Du Thành Đạt - 2010206
Bùi Khánh Vĩnh - 2010091
Đặng Đăng Khánh - 2011383
Ngô Duy Khoa - 2010338

Tp. Hồ Chí Minh, Tháng 1/2022

Contents

1 Giới thiệu

Chắc hẳn, thuật ngữ tấn công mạng không còn xa lạ với tất cả mọi người, đặc biệt là genZ, những người trẻ sôi động với thời lượng hoạt động trên các trang mạng xã hội cực lớn. Tuy nhiên, để có thể thoải mái và tự tin trải nghiệm trên không gian mạng xã hội và các website, các bạn cần phải biết giữ cho mình an toàn cũng như là phải hiểu thế nào là an toàn thông tin cá nhân. Cụ thể, theo thống kê trên toàn thế giới, tổng số vụ tấn công mạng năm 2019 đến 2020 có xu hướng ngày càng cao, theo nhiều mức độ, tệ nạn tấn công và đánh cắp thông tin diễn ra nhiều nhất tại châu Mỹ và kế sau đó là châu Á, trong đó có Việt Nam.

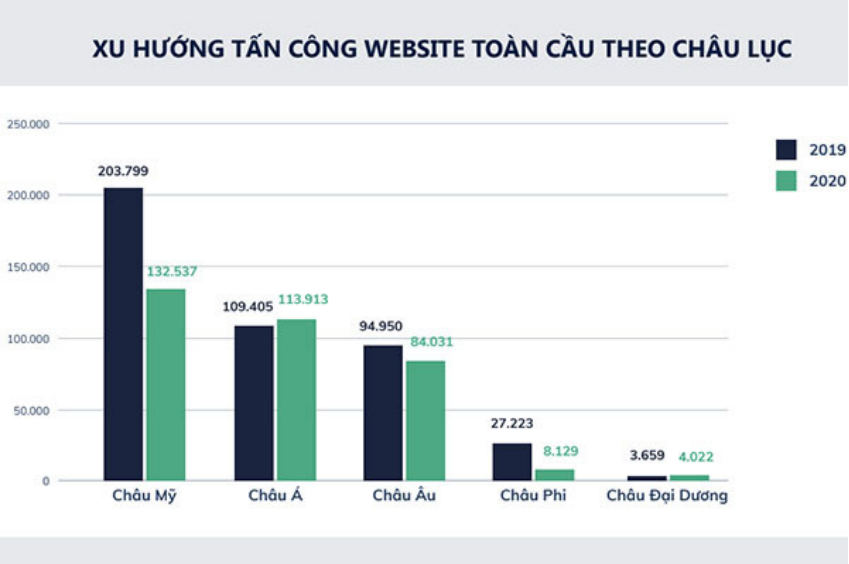


Figure 1: *Số vụ tấn công website trên toàn châu lục*

Riêng tại Việt Nam, số vụ tấn công từ cùng kỳ 2019 đến giữa 2020 giữ mức trung bình 1000 vụ tấn công và đứng vào top 20 quốc gia bị tấn công mạng trên bảng xếp hạng thế giới.

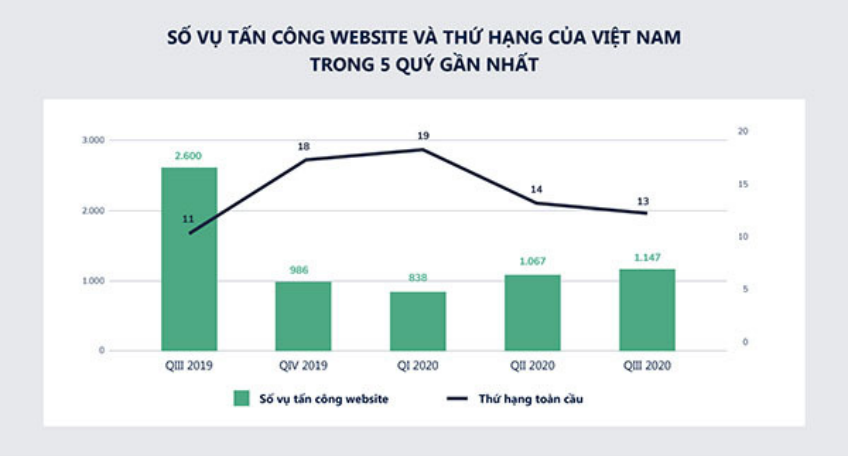


Figure 2: *Số vụ tấn công website tại Việt Nam*

Chính vì lẽ đó, đề tài nghiên cứu này xin giới thiệu đến quý độc giả một sản phẩm với chủ đề nhận biết, phòng chống các loại hình tấn công sử dụng mã độc.

2 Các phương thức tấn công

2.1 Virus

2.1.1 Giới thiệu

Virus máy tính là những đoạn mã chương trình được một cá nhân, tổ chức nào đó thiết kế để xâm nhập vào máy tính của bạn với mục đích ăn cắp thông tin, xóa dữ liệu, gửi email nặc danh, dọa dẫm đòi tống tiền, lấy trộm thông tin quốc gia, trộm tiền ngân hàng,...
Virus máy tính có khả năng tự sao chép chính nó từ vật chủ lây nhiễm này sang vật chủ lây nhiễm khác. Vật mang virus có thể là tệp chương trình, văn bản, bộ nhớ,...Sau khi máy bị nhiễm virus, các phần mềm độc hại có thể khiến máy tính hoạt động chậm hơn, làm hỏng hoặc mất các file dữ liệu, lỗi hệ thống.
Vòng đời virus gồm 4 giai đoạn:

- Dormant: Trong giai đoạn này virut không làm gì cho đến khi được kích hoạt bởi một ai đó hay một sự kiện nào đó.
- Propagation: Trong giai đoạn này virus thực hiện việc copy chính nó tới các chương trình, vị trí khác trong ổ đĩa.
- Triggering: Trong giai đoạn này virus được kích hoạt để thực thi chức năng của nó.
- Execution: Chức năng của virus được thực thi. Chức năng có thể là vô hại như gửi một thông điệp nào đó tới màn hình, hoặc một chức năng có hại như phá hủy các chương trình, các file hệ thống.

2.1.2 Các loại virus thường gặp

- Memory - resident virus: Cư trú rong bộ nhớ chính như là một phần của chương trình hệ thống. Theo đó virus sẽ gây ảnh hưởng mỗi khi chương trình được thực thi..
- Program file virus: Gây ảnh hưởng đến các file chương trình như exe/com/sys.
- Polymorphic virus (virut đa hình): Loại virut này tự thay đổi hình thức của nó, gây khó khăn cho các chương trình anti-virus. Virus “Tequilla” là loại virus đa hình đầu tiên xuất hiện năm 1991.
- Boot Sector virus: Là loại virut đầu tiên trên thế giới được phổ biến rộng rãi và được viết vào năm 1986. Boot virut lợi dụng tiến trình boot của máy tính để thực hiện việc kích hoạt mình. Khi máy tính được khởi động, nó luôn tìm đến master boot record được lưu trữ tại địa chỉ head 0, track 0, sector 1 để đọc thông tin. Boot Sector virus lây lan sang đĩa cứng khi khởi động hệ thống từ đĩa mềm bị nhiễm.
- Stealth virus: Đây là loại virus có khả năng tự che dấu không để cho hệ điều hành và phần mềm chống virus biết. Nó nằm trong bộ nhớ để ngăn chặn sử dụng hệ điều hành và che dấu những thay đổi về kích thước các tập tin. Những virus này chỉ bị phát hiện khi chúng còn ở trong bộ nhớ. Có nhiều boot sector virus có khả năng Stealth. Ví dụ virus "The Brain” được tạo ra tại Pakistan bởi Basit và Amjad. Chương trình này nằm trong phần khởi động (boot sector) của một đĩa mềm 360Kb và nó sẽ lây nhiễm tất cả các ổ đĩa mềm. Đây là loại "stealth virus" đầu tiên.
- Macro virus: Là tập lệnh được thực thi bởi một ứng dụng nào đó. Macro virus phổ biến trong các ứng dụng Microsoft Office khi tận dụng khả năng kiểm soát việc tạo và mở file để thực thi và lây nhiễm. Ví dụ: virus Baza, Laroux và một số virus Staog xuất hiện năm 1996 tấn công các file trong hệ điều hành Windows 95, chương trình bảng tính Excel và cả Linux. Virus Melisa cũng là một trong những Macro virus nổi tiếng.
- Email virus: Là những virus được phát tán qua thư điện tử. Ví dụ virus Melissa được đính kèm trong thư điện tử. Nếu người dùng mở file đính kèm Macro được kích hoạt sau đó email virus này tự động gửi chính nó tới tất cả những hòm thư có trong danh sách thư của người đó.

2.1.3 Cách tấn công

Máy tính hoạt động nhờ các lệnh ở dạng mã máy thuộc dãy số nhị phân để thực hiện một tác vụ nào đó do con người điều khiển. Mã máy được lập trình dẫn tới những công việc được người dùng điều khiển lập đi lập lại nhiều lần và trở thành routine, sau đó sẽ thực thi routine đó. Routine được tạo thành bởi hai cấu trúc là điểm vào (entry) – nơi bắt đầu và điểm ra (exit) – trả lại điều khiển khi đã hoàn thành công việc.
Virus cũng sẽ được viết dưới dạng một routine nhưng sẽ bị sửa tham số địa chỉ, thay vì địa chỉ của máy tính người dùng thì sẽ bị đưa đến vị trí của người tạo ra virus. Do virus máy tính hoạt động dưới dạng mã lệnh nên ít ai có thể phát hiện ra sớm và kịp thời.
Trong một số trường hợp virus được chèn vào phần đầu của chương trình. Code của chương trình gốc có thể được tách rời trong một khối khi bị lây nhiễm. Tuy nhiên, có nhiều trường hợp phức tạp hơn.

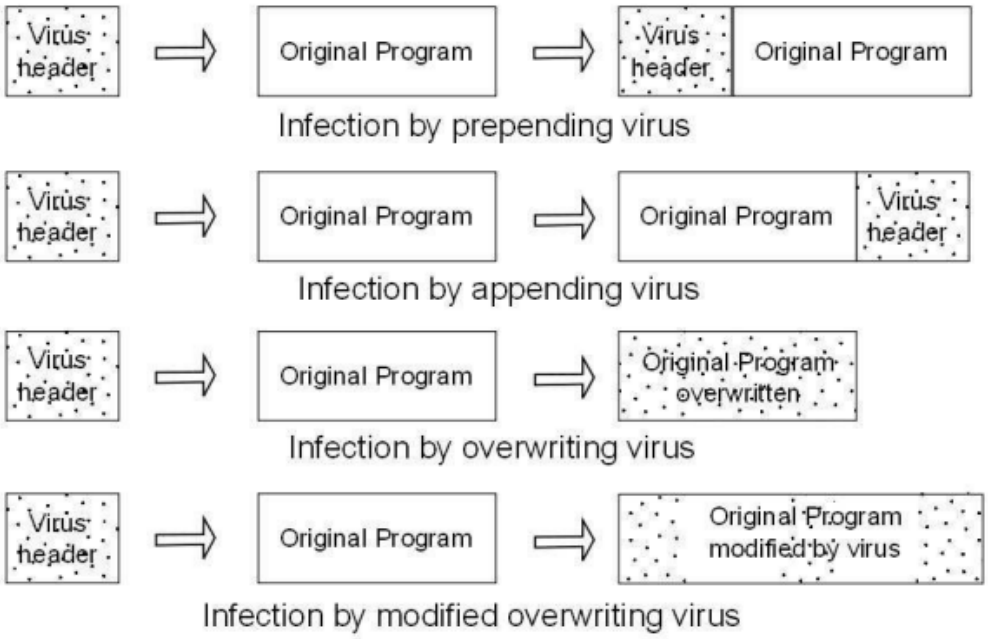


Figure 3: *Vị trí của virus so với chương trình gốc*

Các vi rút ghi đè sẽ ghi đè một phần của chương trình chủ (host program) và sửa đổi header của chương trình chủ để bắt đầu thực thi virus. Trong trường hợp này, kích thước của tệp bị nhiễm có thể không thay đổi. Mô hình này nguy hiểm vì chương trình ban đầu bị hư hỏng vĩnh viễn. Các virus ghi đè đã sửa đổi sẽ tự ghi vào đầu, cuối và các vị trí khác trong chương trình gốc. Có nhiều loại virus ghi đè được sửa đổi phức tạp có thể xáo trộn và ghi đè lên máy chủ gốc theo nhiều cách khác nhau và khiến chúng trở nên vô dụng sau khi bị lây nhiễm.

2.1.4 Cách lây lan

- Virus lây nhiễm theo cách cổ điển: Qua các thiết bị lưu trữ di động: USB, đĩa mềm, đĩa CD, USB hay các thiết bị giải trí kỹ thuật số khác.
- Virus lây nhiễm qua Email: Qua một liên kết trong email hoặc qua các file đính kèm theo email (attached mail).
- Virus lây nhiễm qua mạng Internet: Qua các file tài liệu, phần mềm, trang web chứa virus, qua các lỗi bảo mật hệ điều hành, ứng dụng sẵn có trên hệ điều hành hoặc phần mềm của hãng thứ ba.

2.1.5 Sự kiện tiêu biểu

- Google Trung Quốc (2009): Vào nửa cuối năm 2009, hãng Google tại Trung Quốc đã dính hàng loạt vụ tấn công mạng mang tên Chiến dịch Aurora (Operation Aurora). Không riêng gì Google, khoảng 30 tập đoàn lớn nữa cũng bị loại mã độc này ảnh hưởng. Phía Google cho biết các mã độc này đã không đạt được mục tiêu khi chỉ lấy được 2 tài khoản truy cập của Google. Bởi mã độc này lây lan chủ yếu qua trình duyệt web Internet Explorer nên hàng loạt các nước như Đức, Pháp, Autralia đã khuyến cáo người dùng nên chuyển sang các trình duyệt khác.
- Play Station Network (2011): Trên trang mạng xã hội Twitter vào tối khuya ngày 7/12, nhóm hacker tự nhận tên gọi LizardSquad đã đăng tải thông tin úp mở rằng chính họ là thủ phạm của vụ tấn công nhằm vào Sony PlayStation Network. Vào thời điểm đó, khi truy cập vào PlayStation Store, khách hàng nhận được thông báo "Page Not Found! It's not you. It's the Internet's fault." kèm theo đó là nhiều phản hồi từ giới game thủ cho biết rất khó khăn để chơi các game trực tuyến. Theo Sony, hãng đã tốn tổng cộng 140 triệu Bảng Anh nhằm khắc phục sự cố này.
- Heartbleed (2012-2014): Những cuộc tấn công này cho phép hacker truy cập vào các đoạn hội thoại của người sử dụng mà họ không hề hay biết và qua đó để lại lối mở cho lần truy cập sau đó. Đoạn mã lỗi này tồn tại suốt 2 năm trước khi bị phát hiện và bị Google Security tìm ra cách tiêu diệt vào năm 2014.
- Yahoo (2012-2014): Trong khoảng thời gian này, tất cả dữ liệu như mật khẩu, thông tin cá nhân, số liệu tài khoản... của khách hàng lưu giữ trong tài khoản Yahoo đã bị đánh cắp. Hiện Yahoo vẫn chưa thể giải quyết triệt để cuộc tấn công này và gây ảnh hưởng đến nhiều công ty liên kết khác như My Space (thiệt hại 359 triệu USD), LinkedIn (164 triệu USD) và Adobe (152 triệu USD).
- Sony Picture Entertainment (2014): Ngày 24/11/2014, Sony Pictures Entertainment - một công ty con của tập đoàn Sony Nhật Bản - đã hứng chịu một đợt tấn công bảo mật khiến toàn bộ hệ thống máy tính nhân viên tại hãng phim này phải ngưng hoạt động.

2.1.6 Cách phần mềm diệt virus hoạt động

Mặc dù chương trình diệt virus nhằm mục đích ngăn chặn bất kỳ cuộc tấn công nào của virus, nhưng có khả năng một số tệp đã bị lây nhiễm trước khi chương trình được cài đặt hoặc trong khoảng thời gian chương trình không được cập nhật. Trong trường hợp đó, chương trình diệt virus phải khử trùng các tệp bị nhiễm. Đây là một trong những chức năng khó và quan trọng nhất của bất kỳ chương trình diệt virus nào. Phần mềm diệt virus phải áp dụng nhiều phương pháp khác nhau để loại code virus khỏi tệp bị nhiễm và khôi phục tệp ở dạng ban đầu.

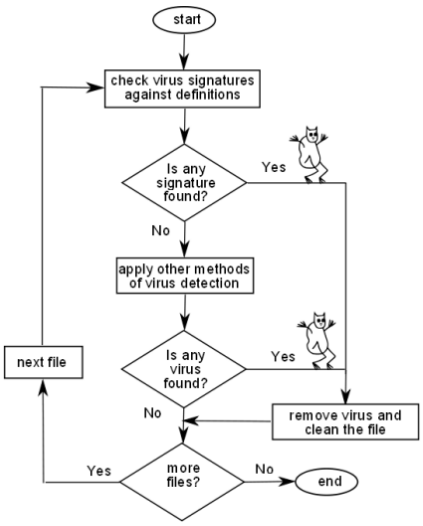


Figure 4: *Trình tự xử lý của phần mềm diệt virus*

Trước tiên, phần mềm diệt virus cố gắng phát hiện sự hiện diện của virus bằng các phương pháp khác nhau. Phát hiện signature, là phương pháp phổ biến nhất, có thể được áp dụng đầu tiên. Nếu không tìm thấy signature thì chương trình chống virus sẽ áp dụng các phương pháp khác như quét theo phương pháp heuristic. Nếu không có nghi ngờ nào được nêu ra trên một tệp, thì tệp đó được coi là chưa bị nhiễm. Trình tự quét và dọn virus được mô tả ở hình trên. Phần mềm chống vi-rút phải quét lần lượt tất cả các tệp và thực hiện một loạt các hành động đối với các tệp bị nhiễm.

Việc đầu tiên của bất kỳ phần mềm diệt virus nào là sửa chữa các tệp hoặc các phần bị hỏng của đĩa.

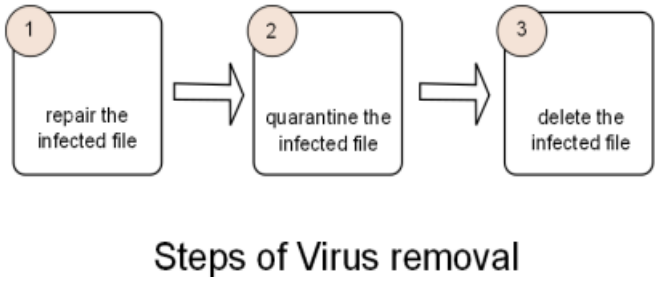


Figure 5: *Các giải pháp theo thứ tự ưu tiên*

Tuy nhiên, nếu phần mềm diệt virus không biết phương pháp sửa chữa sự lây nhiễm thì nó sẽ cô lập tệp bị nhiễm bệnh để cách ly và sửa chữa trong tương lai. Nếu một loại virus được phát hiện là quá nguy hiểm hoặc tệp bị hư hỏng nghiêm trọng thì phần mềm diệt virus có thể xóa tệp bị nhiễm. Do đó, các quá trình thường được sắp xếp theo thứ tự, chẳng hạn như sửa chữa tệp (ưu tiên nhất), cách ly tệp (nếu không thể sửa chữa) và xóa tệp (ít ưu tiên nhất).

2.1.7 Cách phòng chống

- Cài đặt các phần mềm chống virus
- Bật tường lửa
- Giữ cho máy tính luôn được cập nhật
- Đóng băng hệ thống
- Định kỳ sao lưu dữ liệu

2.1.8 Tham khảo

+ https://www.thegioididong.com/game-app/virus-may-tinh-la-gi-cach-phong-chong-cac-loai-virus-phuong-1331152

+ https://ben.com.vn/tin-tuc/virus-may-tinh-la-gi/

+ https://viettelidc.com.vn/tin-tuc/5-vu-tan-cong-mang-lon-nhat-the-gioi-truoc-virus-wannacry

+ https://congnghe.vn/muc/bao-mat/tin/den-luot-sony-playstation-network-bi-tan-cong-1708896

+ https://arxiv.org/ftp/arxiv/papers/1306/1306.4666.pdf

2.2 Trojan Horse

Figure 6: *Nguồn: Bizflycloud*

2.2.1 Giới thiệu

Trojan Horse là kỹ thuật che giấu các đoạn mã độc dưới vỏ bọc bên trong các phần mềm thông thường để bí mật xâm nhập nhằm đánh cắp thông tin cá nhân, mật khẩu, thậm chí chiếm quyền điều khiển máy tính.

Bản chất của Trojan là không tự lây lan mà sử dụng phần mềm khác để phát tán.

2.2.2 Phương thức hoạt động chung

Trojan horse giả mạo và ngụy trang dưới dạng tệp "an toàn" để qua mặt các phần mềm bảo vệ, phần mềm diệt virus tuyến đầu. Tới thời điểm nhất định hoặc người dùng vô tình kích hoạt, chúng sẽ bắt đầu hoạt động đánh cắp thông tin cá nhân, chiếm quyền điều khiển máy tính...

2.2.3 Hậu quả thường gặp

- Xoá hoặc thay đổi dữ liệu.
- Làm hỏng hoặc sai lệch chức năng của các phần mềm.
- Lây nhiễm các phần mềm ác tính khác như là virus.
- Thu thập thông tin gián điệp.
- Ăn cắp thông tin như là mật khẩu và số thẻ tín dụng.
- Điều khiển các hành vi phạm tội.

2.2.4 Nguồn gốc

- Đường dẫn hoặc tệp tin đính kèm trong các thư điện tử, diễn đàn trực tuyến, mạng xã hội,...
- Phần mềm tải về từ nơi không đáng tin cậy.

2.2.5 Phương thức tấn công

Trojan Horse có các cách tấn công từ đơn giản đến phức tạp:

Từ một số đường dẫn (đặc biệt là đường dẫn rút gọn **bit.ly**) mà bạn cho là tin cậy được nhận qua mail, khi truy cập vào, có thể một số tệp độc hại đã tràn vào máy bạn. Hoặc từ một source code của một ứng dụng nào đó bạn muốn tải về, nếu đó không phải trang chủ thực sự của ứng dụng thì chỉ cần 1 tệp gài vào là đủ khiến máy tính bạn nhiễm loại virus này. Theo khoa học bảo mật, Trojan có nhiều loại ví dụ như: BackDoor, Rootkit và Spyware, Rootkit, Exploit,... Sau đây là một số loại phổ biến

1. BackDoor Được xem như là loại Trojan đơn giản nhất nhưng lại gây nguy cơ độc hại nhất. BackDoor thường được dùng chủ yếu để cài Botnet (Thuật ngữ ghép bởi Robot và Network).
2. Rootkit Để trợ giúp các phần mềm độc hại Trojan khác ẩn giấu khỏi các phần mềm diệt virus đồng thời kéo dài thời hạn tồn tại của chúng. Rootkits được thiết kế để hỗ trợ điều đó.

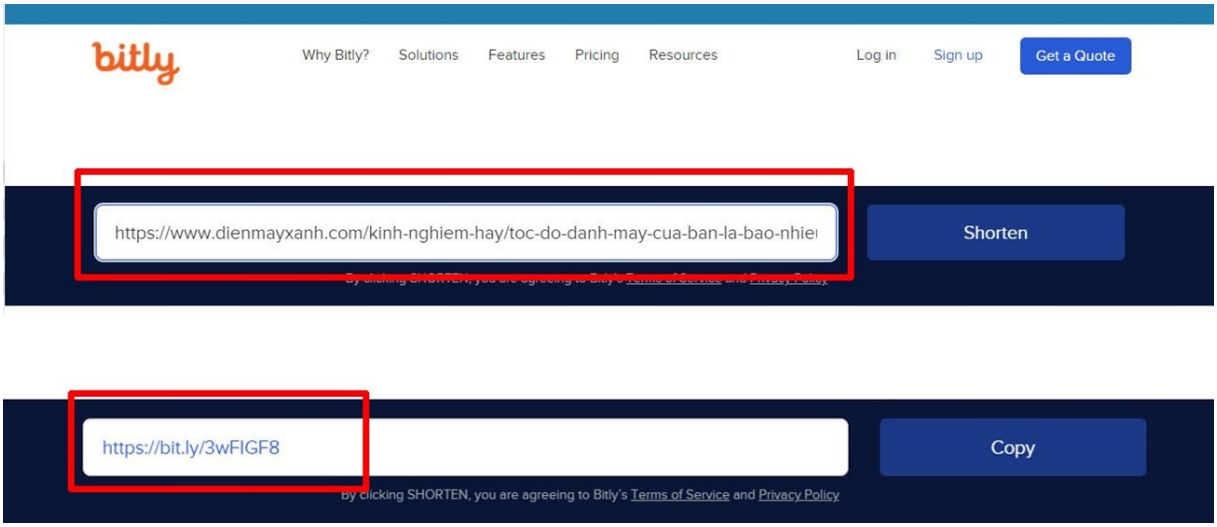
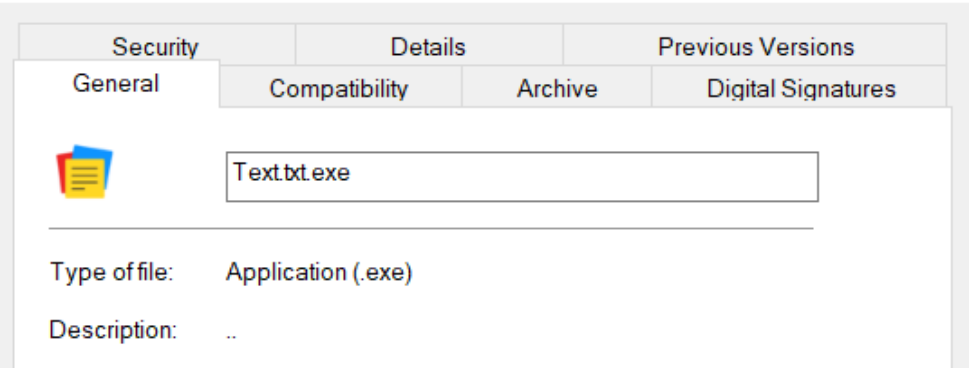


Figure 7: *Nguồn: Điện máy xanh*

Một số trường hợp tấn công đơn giản

- Ví dụ đơn giản của một Trojan horse là một chương trình mang tên "text.txt" được đăng trên một trang Web với lời giới thiệu là đoạn văn mẫu tiểu luận Triết học Marx LêNin; nhưng, khi chạy, chương trình này lại xoá tất cả tệp trong máy tính.
- Ví dụ khác (**cân nhắc trước khi thử**) Trojan horse có ở www.freewebs.comem_ce_dodoctor.exe. Chương trình này sẽ tự động tắt máy khi chạy và sẽ tự chép phiên bản vào thư mục "StartUp" và như vậy máy sẽ tự động tắt ngay lập tức mỗi lần máy được khởi động. Tin vui là con Trojan horse này sẽ tự hủy sau một giờ hoạt động hoặc có thể được xóa bỏ bằng cách khởi động vào chế độ chờ lệnh (command prompt) và từ đó xóa tệp này bằng lệnh xóa. Chương trình này chỉ chạy được trên Windows XP.

Ngoài ra, trong nhiều ứng dụng của Windows đã có cấu hình mặc định không cho phép hiển thị các đuôi này. Do đó, nếu một Trojan horse có tên chẳng hạn là "Readme.txt.exe" thì tệp này sẽ hiển thị một cách mặc định thành "Readme.txt" và nó sẽ đánh lừa người dùng rằng đây chỉ là một loại hồ sơ văn bản không thể gây hại.



Các biểu tượng cũng có thể được gán với các loại tệp khác nhau và có thể được đính kèm vào thư điện tử. Khi người dùng mở các biểu tượng này thì các Trojan horse ẩn giấu sẽ tiến hành những tác hại bất ngờ. Hiện nay, các Trojan horse không chỉ xoá các tệp, bí mật điều chỉnh cấu hình của máy tính bị nhiễm mà còn dùng máy này như là một cơ sở để tấn công các máy khác trong mạng.

2.2.6 Cách phòng chống

Cách hữu hiệu nhất là đừng bao giờ mở các đính kèm được gửi đến một cách bất ngờ. Khi các đính kèm không được mở ra thì Trojan horse cũng không thể hoạt động. Cẩn thận với ngay cả các thư điện tử gửi từ các địa chỉ quen biết. Trong trường hợp biết chắc là có đính kèm từ nơi gửi quen biết thì vẫn cần

phải thử lại bằng các chương trình chống virus trước khi mở nó. Các tệp tải về từ các dịch vụ chia sẻ tệp như là Kazaa hay Gnutella rất đáng nghi ngờ, bao gồm các phần mềm hack/crack của một ứng dụng bản quyền hay game,... vì các dịch vụ này thường bị dùng như là chỗ để lan truyền Trojan horse.

2.2.7 Tài liệu tham khảo

- https://www.kaspersky.com

2.3 SQL-INJECTION(SQLi)

Figure 8: *SQLi, nguồn từ: YMtech IT Consulting*

2.3.1 Giới thiệu

+ **SQLi hay SQL injection** là một kỹ thuật cho phép hacker lợi dụng những lỗ hổng bảo mật liên quan đến hệ cơ sở dữ liệu thông qua hình thức đầu vào của trang web hay hiển thị các exception để thực hiện các câu lệnh SQL bất hợp pháp nhằm phục vụ cho mục đích của mình.

+ **SQL rất phổ biến** vì đa phần các ứng dụng, web hiện nay đều có cơ sở dữ liệu của riêng nó, và hầu hết đều được lập trình bằng sql cùng một số framework, ngôn ngữ phổ biến như sqlite3, php, asp.net,...

2.3.2 Các phần dễ bị tấn công

+ Phần login khi hiển trị trang đăng nhập.

+ Phần request http.

+ Cookie của browser.

+ Các Database viết bằng SQL thuần.

2.3.3 Cách tấn công

2.3.3.a Incorrectly constructed SQL statements

+Không kiểm tra kí tự thoát:

Việc thiếu đoạn mã kiểm tra truy vấn giúp cho end_user có thể thực hiện một số truy vấn không mong muốn thông qua các câu lệnh WHERE, SELECT với các truy vấn mang giá trị là hằng đúng. Chẳng hạn:

```
string statement = ‘‘SELECT * FROM users WHERE name = ‘‘‘ + username + ‘‘‘;’’;
```

Khi đó với việc ta chỉ cần gán trá trị cho username như:

```
‘ or 1 = 1-’ hoặc a’ or ‘t’ = ‘t
```

Khiến câu truy vấn có thể generate ra như sau:

```
SELECT * FROM users WHERE name = ‘a’ or ‘t’ = ‘t’;
SELECT * FROM users WHERE name = ‘’ or 1 = 1’’;
```

Làm hiển thị toàn bộ thông tin của table Users.

Bên cạnh đó kết hợp việc thực hiện nhiều truy vấn SQL cùng lúc thông qua biến username cũng có thể xảy ra chẳng hạn kết hợp câu lệnh xóa table dưới đây:

```
a' or 1 = 1; DROP TABLE users; SELECT * FROM data WHERE = 'a;
```

Khiến câu truy vấn generate như sau:

```
SELECT * FROM users WHERE name = 'a' or 1 = 1;DROP TABLE users;
SELECT * FROM data WHERE = 'a';
```

Như vậy sẽ làm mất hết toàn bộ dữ liệu users

+Không kiểm tra kiểu dữ liệu: Lỗi này xảy ra khi định nghĩa dữ liệu đầu vào không rõ ràng hoặc thiếu việc lọc dữ liệu đầu vào. Chẳng hạn như:

```
statement:= "SELECT * FROM data WHERE id = " + a_variable + ";"
```

Nhìn vào câu lệnh ta có thể ngầm hiểu biến id mang giá trị số nguyên (INT) nhưng việc không nêu ra kĩ kiểu dữ liệu id, ta có thể chèn một chuỗi vào đó mà không cần kí tự thoát. ví dụ:

```
id = ''2; DROP TABLE users'';
```

Khi đó thao tác xóa dữ liệu hoàn thành qua câu lệnh được generate như sau:

```
SELECT * FROM data WHERE id=1;DROP TABLE users;
```

2.3.3.b Blind SQL injection

Là một kiểu tấn công SQLi truy vấn cơ sở dữ liệu sử dụng việc đoán biết lỗi. Cách tấn công này thường được dùng cho web, app qua phương thức get, set trong https nơi cấu hình chỉ hiển thị lỗi chung chứ không hiển thị lỗi SQL. Phương pháp này còn được sử dụng để dự đoán liệu web có bị lỗi SQL injection hay không.

+Blind SQL injection dựa vào nội dung phản hồi:

```
http://jobsvietnam.net/leademployers.php?id=1 and 1 = 2:
```

Nếu câu lệnh không làm hiển thị nội dung ta thực hiện bước tiếp theo.

```
http://jobsvietnam.net/leademployers.php?id=1 and 1 = 1:
```

Nếu bước này đúng, tứ web hiển thị thì trang web đã bị lỗi sql injection

+Blind SQL injection dựa vào độ trễ của thời gian phản hồi:

```
http://www.shop-online.com/product_detail.php?id=1 and if(1=1, sleep(10), false)
```

Nếu web đợi 10 s tức là web này bị lỗi sql injection, ta có thể tấn công và khai thác để xác định thời gian chính xác mà trang cần tải khi giá trị nhập vào là đúng.

+Điều kiện lỗi

```
SELECT 1/0 from users where username='Ralph';
```

Nếu câu lệnh báo lỗi do phép 1/0 sai tức là trong cơ sở dữ liệu có username tên 'Raplh'.

2.3.3.c Chèn SQL Injection dựa trên cookie

Một cách tiếp cận khác với SQL Injection là sửa đổi cookie thành các truy vấn cơ sở dữ liệu chứa mã độc. Các phần mềm độc hại có thể được triển khai trên thiết bị người dùng thông qua thay đổi của cookie, nhằm mục đích đưa SQL Injection vào các dữ liệu Back-end.

2.3.3.d Dạng tấn công sử dụng stored-procedures

Nếu ta thay đoạn mã tiêm vào dạng: '; EXEC xp_cmdshell 'cmdd.exe dir C: '. thì hệ thống sẽ thực hiện lệnh liệt kê các mục có trong ổ đĩa C. Việc tấn công này còn phụ thuộc vào cmd, thông qua quyền root.

2.3.3.e Tấn công sử dụng câu lệnh SELECT, INSERT, UNION

2.3.4 Tác hại

+ Thu thập thông tin của user: kẻ tấn công sử dụng tài khoản của user này để mạo danh và sử dụng các đặc quyền của họ.
+ Truy cập cơ sở dữ liệu bất hợp pháp.
+ Thay đổi dữ liệu: xóa hoặc thêm dữ liệu quan trọng.
+ Truy cập mạng: kẻ tấn công truy cập vào máy chủ với các đặc quyền của hệ điều hành

2.3.5 Tác hại

+ Thu thập thông tin của user: kẻ tấn công sử dụng tài khoản của user này để mạo danh và sử dụng các đặc quyền của họ.
+ Truy cập cơ sở dữ liệu bất hợp pháp.
+ Thay đổi dữ liệu: xóa hoặc thêm dữ liệu quan trọng.
+ Truy cập mạng: kẻ tấn công truy cập vào máy chủ với các đặc quyền của hệ điều hành.

2.3.6 Cách phòng chống

2.3.6.a Sử dụng Framework mã hóa thông tin thành parameter:

Mã hóa các thông tin thành tham số thay vì cộng chuỗi => như thế thuận tiện cho việc kiểm tra với các hàm trong framework cũng như hạn chế các lỗi đã xuất hiện ở trên

+ASP.NET Razor Example

```
txtUserId = getRequestString("UserId");
txtSQL = "SELECT * FROM Users WHERE UserId = @0";
db.Execute(txtSQL,txtUserId);
#Note that parameters are represented in the SQL statement by a @ marker
```

+PHP

```
$stmt = $dbh->prepare("INSERT INTO Customers (CustomerName,Address,City)
VALUES (:nam, :add, :cit)");
$stmt->bindParam(':nam', $txtNam);
$stmt->bindParam(':add', $txtAdd);
$stmt->bindParam(':cit', $txtCit);
$stmt->execute();
```

2.3.6.b Mã hóa dữ liệu:

+Hash một chiều:

Thay vì lưu mật khẩu xuống database, ta sẽ lưu mật khẩu đã qua hashing, như thế chỉ có người dùng biết password ngay cả admin, DBA cũng không biết được mật khẩu.

```
struct User{
private:
        string name;
        string password;
        //...
};

struct DataBase; //static

void register(string name, string password){
        string hashPw = HashHelper.hash(password);
        DataBase.saveUser(User(name,hashPw));
}

bool login(string name, string password){
        string hashPwDB = DataBase.getHashPwdFromLogin(name);
        string hashPwUser = HashHelper.hash(password);
        return hashPwDB == hashPwUser;
}
```

Hạn chế: Do hai mật khẩu giống nhau sẽ có cùng cách hash, vì vậy dựa vào từ điển tạo ra các mật khẩu có thể có khi đó dùng brute force để tìm mật khẩu đã được hash.

+Hash password with random salt: (salt là chuỗi kí tự random)

Phương pháp này ra đời nhằm khắc phục hạn chế của Hash một chiều khi chuỗi kí tự cộng thêm vào được random bất kì, hầu như brute force để mò là bất khả thi (vì ta không biết chính xác độ dài, mã, có kí tự đặc biệt hay không).

```
struct User{
private:
        string name;
        string password;
        string salt;
        //...
};

struct DataBase; //static

void register(string name, string password){
        string salt = Salt.getRandom();
        string hashPw = HashHelper.hash(salt + password);
DataBase.saveUser(User(name,hashPw,salt));
}

void login(string name, string password){
        string hashPwDB = DataBase.getHashPwdFromLogin(name)
        string hashPwUser = HashHelper.hash(password + salt);
        return hashPwDB = hashPwUser;
}
```

+Sử dụng các cách encrypt:

- Cổ điển: Cesar (dịch vòng), thay thế (hoán vị hoặc ánh xạ kí tự này đến kí tự khác), Affine (lợi dụng tính chất nếu gcd(a,n) = 1 thì phương trình ax + b = d (mod n) có nghiệm duy nhất)
- Hiện đại: RSA, DES,...

2.3.6.c Không để lộ thông tin hệ thống (message, exception):

Tác hại: Lộ thông tin framework, plugin, api security giúp dễ tìm được cách truy xuất.

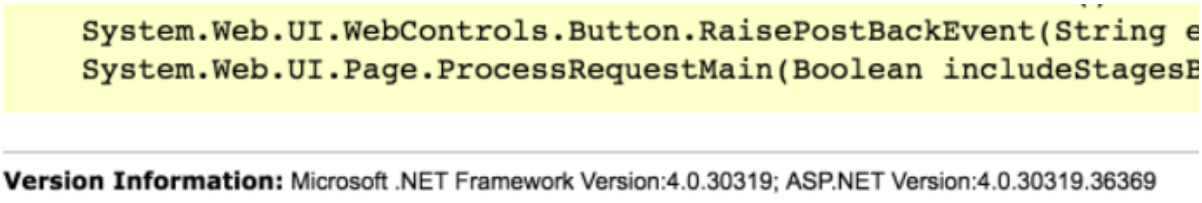


Figure 9: *Hiển thị thông tin .Net*

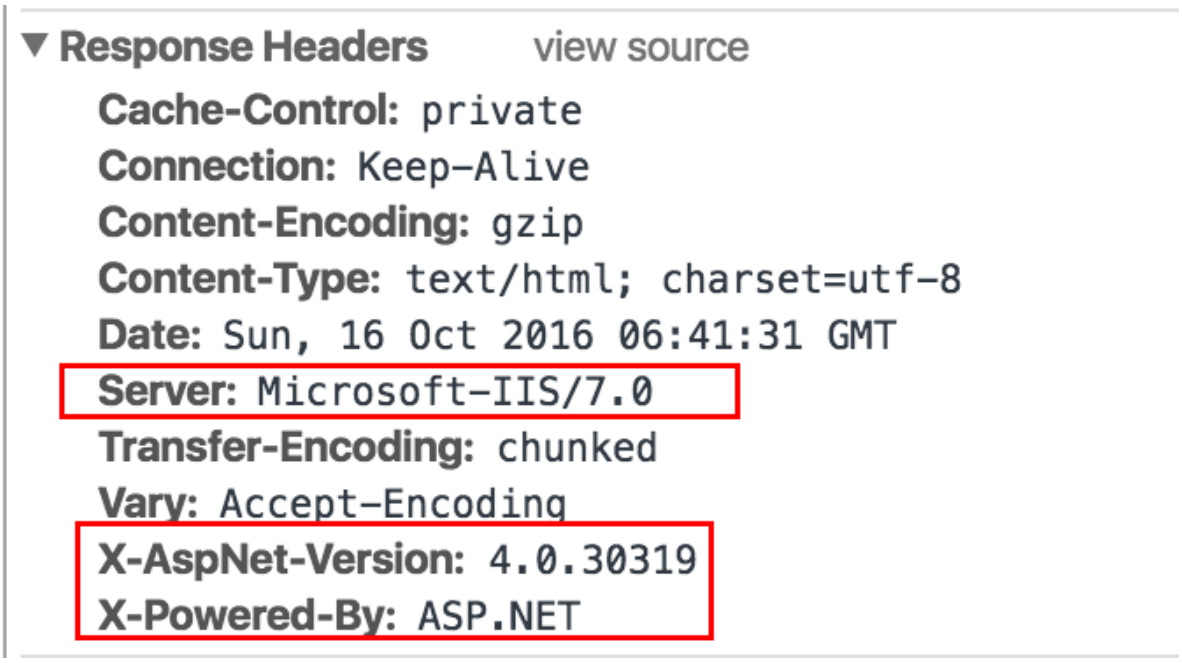


Figure 10: *thông tin để trong header trả về từ server*

Cách phòng chống:

+ Viết code khóc đọc (uglify code) hoặc loại bỏ các thẻ http dư thừa để che giấu các thư viện được xài.

+ Chỉ hiển thị error cho người dùng biết chứ không hiển thị exception.

+ Cập nhật framework, theo dõi thường xuyên lỗ hổng của thư viện.

2.3.6.d Regex - Detect các mã độc hại

```
(?<!prepare)\(('|")SELECT.+FROM.+('|").*\..*
```

Tìm tất cả truy vấn SELECT trong Plugin mà không sử dụng prepare. Thư viện prepare() để bảo vệ truy vấn khỏi các cuộc tấn công sql injection.

Bên cạnh đó ta cũng có thể xài regex để viết lại tên miền cho đúng cũng như kiểm tra input có phù hợp với yêu cầu mình mong muốn.

2.3.6.e Backup Dữ liệu:

Việc sao lưu dữ liệu thường xuyên giúp chúng ta khi có trục trặc gì xảy ra thì vẫn có thể tìm lại được dữ liệu như ban đầu. Chúng ta không thể phòng chống hết tất cả rủi ro do mã độc mang lại nên việc như thế này là rất cần thiết.

2.3.7 Sự kiện tiêu biểu

+ 29/6/2007, một tên tội phạm máy tính đã phá hoại trang web của Microsoft Vương quốc Anh.
+ Vào ngày 21 tháng 2 năm 2014, Diễn đàn Quản trị Internet của Liên hợp quốc đã có 3.215 thông tin chi tiết về tài khoản bị rò rỉ.
+ BKAV bị tấn công vào ngày 15/8/2021.

2.3.8 Tham khảo:

+ https://toidicodedao.com/2016/11/15/lo-hong-sql-injection-than-thanh/
+ https://www.w3schools.com/sql/sql_injection.asp
+ http://expressmagazine.net/development/1512/tan-cong-kieu-sql-injection-va-cac-phong-chong-trong-aspnet
+ http://freetuts.net/ky-thuat-tan-cong-sql-injection-va-cach-phong-chong-trong-php-107.html
+ https://vi.wikipedia.org/wiki/SQL_injection
+ https://en.wikipedia.org/wiki/SQL_injection
+ https://viblo.asia/p/write-up-rootme-web-khai-thac-sql-injection-Qbq5Qa145D8
+ https://viblo.asia/p/blind-sql-injection-la-gi-blind-injection-khac-voi-cac-loai-sql-injection-khac-nhu-the-nao-3Q75wX0DKWb

2.4 Ransomware

2.4.1 Giới thiệu

Ransomware là một loại mã độc có mục đích tống tiền bằng cách xâm nhập vào máy tính và thao túng dữ liệu người dùng. Chúng mã hóa dữ liệu của người dùng, thay đổi nội dung, tên file và đuôi file khiến cho file không sử dụng được hoặc khóa quyền truy cập thiết bị của người dùng. Người dùng cần phải trả tiền để lấy lại quyền truy cập dữ liệu qua chuyển khoản hoặc bitcoin.
Trong thời gian 2019-2020, số lượng các vụ tấn công ransomware tăng 62%, chỉ tính riêng khu vực Bắc Mỹ đã tăng 158%, dựa theo báo cáo bảo mật của SonicWall năm 2021. FBI cũng đã nhận hơn 2500 báo cáo về ransomware trong năm 2020, tăng hơn 20% so với năm 2019, theo báo cáo thường niên của Trung tâm Báo cáo Tội phạm mạng IC3. Số tiền thu được từ ransomware được báo cáo trong năm 2020 là khoảng 29,1 triệu $, tăng gấp hơn 2 lần so với số tiền 8,9 triệu $ năm trước đó.

2.4.2 Cách tấn công

Máy tính thường bị nhiễm ransomware theo một số đường phổ biến sau:

- Sử dụng các phần mềm crack, không rõ nguồn gốc.
- Click vào các file đính kèm có chứa ransomware trong mail hoặc tin nhắn.
- Click vào những quảng cáo không rõ nguồn gốc chứa mã độc.
- Truy cập các trang web có độ bảo mật kém, chứa nội dung đồi trụy.

2.4.3 Các dạng ransomware

1. Ransomware mã hóa (encrypting ransomware)
   Encrypting ransomware là loại mã độc tống tiền phổ biến nhất. Chúng mã hóa dữ liệu dữ liệu của người dùng. Sau khi xâm nhập vào máy tính của người dùng, chúng sẽ âm thầm kết nối với server của kẻ tấn công, tạo ra hai chìa khóa – một khóa công khai để mã hóa các file của bạn, một khóa riêng do server của hacker nắm giữ, dùng để giải mã. Các file này sẽ bị đổi đuôi thành những định dạng nhất định và báo lỗi khi người dùng cố gắng mở. Sau khi mã hóa file, crypto ransomware sẽ hiển thị một thông báo trên máy tính của bạn, thông báo về việc bạn đã bị tấn công và phải trả tiền chuộc cho chúng. Trong một vài trường hợp, kẻ tấn công còn tạo thêm áp lực bằng cách đòi hỏi nạn nhân phải trả tiền trong thời hạn nhất định. Sau thời hạn đó, khóa giải mã file sẽ bị phá hủy hoặc mức tiền chuộc sẽ tăng lên.

2. Ransomware không mã hóa (non-encrypting ransomware)
   Non-encrypting ransomware (hay còn gọi là Locker) là loại phần mềm không mã hóa file của nạn nhân. Thay vì mã hóa dữ liệu của người dùng, nó khóa và chặn người dùng khỏi thiết bị. Nạn nhân sẽ không thể thực hiện được bất kỳ thao tác nào trên máy tính (ngoại trừ việc bật – tắt màn hình). Trên màn hình cũng sẽ xuất hiện hướng dẫn chi tiết về cách thanh toán tiền chuộc để người dùng có thể truy cập lại và sử dụng thiết bị của mình.

3. Leakware (Doxware)
   Một số loại ransomware đe dọa công khai dữ liệu của nạn nhân lên mạng nếu không chịu trả tiền chuộc. Nhiều người có thói quen lưu trữ các file nhạy cảm hoặc ảnh cá nhân ở máy tính nên sẽ không tránh khỏi việc hoảng loạn, cố gắng trả tiền chuộc cho hacker. Loại ransonware này thường được gọi là leakware hoặc doxware.

4. Mobile ransomware
   Thông thường, mobile ransomware xuất hiện dưới dạng phần mềm chặn người dùng khỏi việc truy cập dữ liệu (loại non-encrypting) thay vì mã hóa dữ liệu bởi dữ liệu trên mobile có thể dễ dàng khôi phục thông qua đồng bộ hóa trực tuyến (online sync).

   Mobile ransomware thường nhắm vào nền tảng Android, vì hệ điều hành này cấp quyền "Cài đặt ứng dụng" cho bên thứ ba. Khi người dùng cài đặt file .apk chứa mobile ransomware, sẽ có 2 kịch bản có thể xảy ra:

- Chúng sẽ hiển thị pop-up (tin thông báo) chặn không cho người dùng truy cập vào tất cả các ứng dụng khác.
- Sử dụng hình thức "bắt buộc nhấp chuột" (clickjacking) để khiến người dùng vô tình cấp quyền quản trị thiết bị. Khi đó, mobile ransomware sẽ truy cập sâu hơn vào hệ thống và thực hiện các hình thức vi phạm khác.

Đối với hệ điều hành iOS, kẻ tấn công cần áp dụng những chiến thuật phức tạp hơn, chẳng hạn như khai thác tài khoản iCloud và sử dụng tính năng "Find my iPhone" để khóa quyền truy cập vào thiết bị.

2.4.4 Các cuộc tấn công tiêu biểu

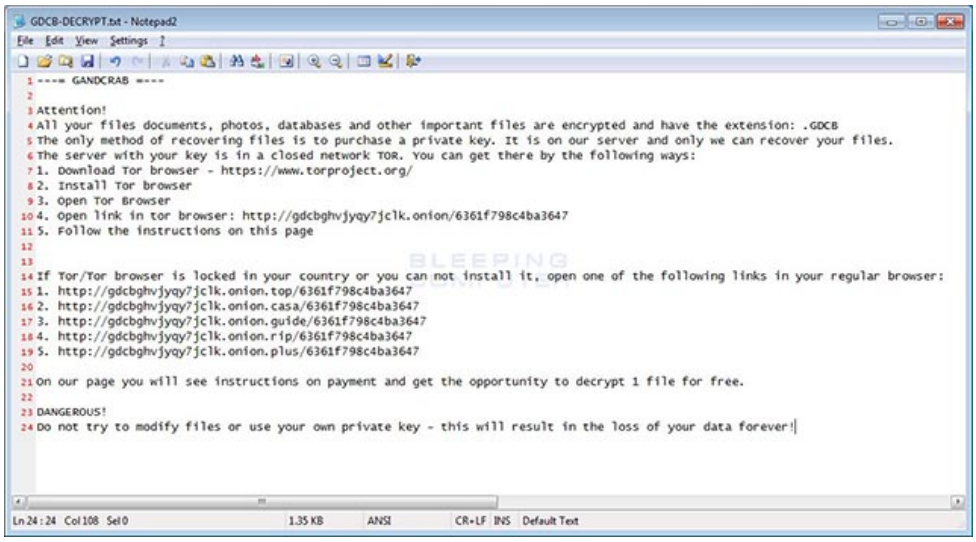
1. WannaCry
   WannaCry là một dạng mã độc tống tiền, thực hiện mã hóa các dữ liệu quan trọng để người dùng không sử dụng được (crypto ransomware) hoặc chặn quyền truy cập của người dùng trên máy tính (locker ransomware). WannaCry nhắm vào những máy tính sử dụng hệ điều hành Windows, yêu cầu thanh toán bằng Bitcoin để trả lại dữ liệu.

Thông báo nhiễm mã độc WannaCry

Vụ tấn công WannaCry xảy ra vào năm 2017 phá hủy hệ thống máy tính của khoảng 150 quốc gia. Khoảng 230.000 máy tính bị dính mã độc này, gây ảnh hưởng đến các tổ chức, doanh nghiệp lớn như FedEx, Hệ thống Dịch vụ Y tế Anh (NHS), bộ nội vụ Nga,... với số tiền thiệt hại khoảng 4 tỉ$. Các tội phạm mạng đã chiếm quyền kiểm soát máy tính thông qua một lỗ hổng trên hệ điều hành Windows bị cáo buộc do Cơ quan An ninh Quốc gia Mỹ (NSA) phát triển. Lỗ hổng này có tên là EternalBlue, được công khai bởi một nhóm hacker tên là Shadow Breakers trước vụ tấn công WannaCry. Microsoft đã tung bản vá lỗi trước vụ tấn công gần 2 tháng nhưng nhiều cá nhân, tổ chức đã không nâng cấp phần mềm nên đã gây ra vụ tấn công diện rộng.
Ban đầu, người ta đặt ra giả thiết rằng WannaCry đã lan truyền qua một chiến dịch phishing (gửi liên tục những email có chứa đường link hay liên kết dụ người dùng tải về mã độc). Mặc dù vậy, EternalBlue là lỗ hổng cho phép WannaCry sinh sản và phát tán với DoublePulsar như "cửa hậu" được tải trên máy tính để thực hiện mã độc WannaCry.

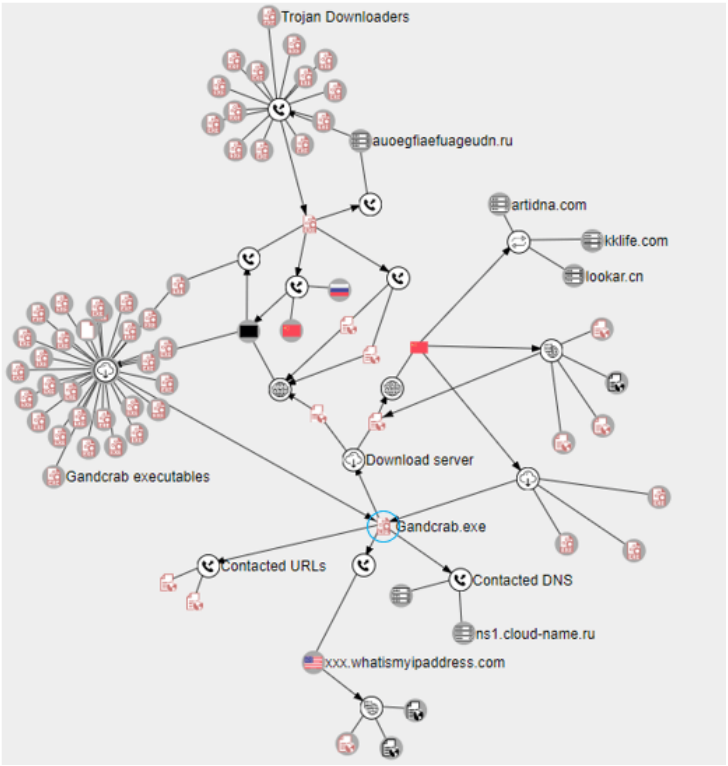
2. GandCrab

Mã độc tống tiền GandCrab lần đầu tiên được phát hành vào ngày 28/1/2018, được phát tán dưới dạng email spam, các bộ công cụ khai thác lỗ hổng và các chiến dịch xây dựng phần mềm độc hại khác nhau. Mã độc GandCrab được phân phối thông qua bộ công cụ khai thác lỗ hổng RIG, khi bị nhiễm toàn bộ file trên máy người dùng sẽ bị mã hóa và phần mở rộng sẽ được thêm đuôi .gand hoặc .crab, đồng thời sinh ra một file CRAB-DECRYPT.txt nhằm yêu cầu và hướng dẫn người dùng trả tiền chuộc từ 400-1000 $ bằng tiền điện tử DASH và sử dụng tên miền cấp cao (TLD). Điều đáng tiếc là tên miền TLD này lại không bị ICANN (Tập đoàn Internet cấp số và tên miền) xử phạt và do đó vô tình nó đã cung cấp thêm một mức độ "bảo mật" cho những kẻ tấn công.

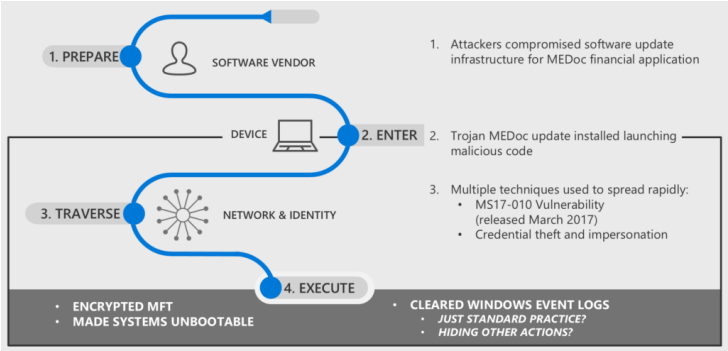


Thông báo tiền chuộc của GandCrab

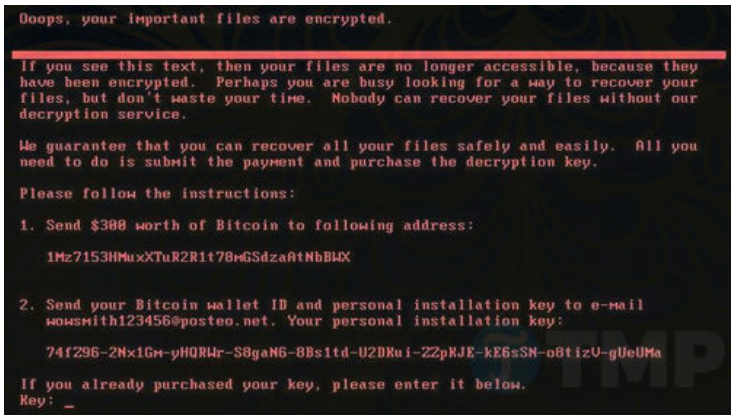
Theo điều tra, địa chỉ URL được kết nối đến một máy chủ ở Trung Quốc và có dấu hiệu bị xâm nhập trái phép. Mạng lưới của GandCrab khá phức tạp và có liên quan đến mã độc Trojan.



3. Petya và NotPetya
   Mã độc Petya được khám phá lần đầu tiên vào tháng 3/2016, khi các nhà nghiên cứu bảo mật của CheckPoint để ý rằng dù các vụ tấn công tống tiền trở nên ít đi nhưng chúng có những điểm khác biệt đáng lưu ý trong cách hoạt động. Đến tháng 5/2016, bản nâng cấp của Petya xuất hiện payload thứ cấp (dùng để xóa, mã hóa dữ liệu người dùng nếu người dùng không thể chiếm được quyền kiểm soát hệ thống). Petya nhắm vào các thiết bị chạy hệ điều hành Windows.
   Petya sử dụng lỗ hổng CVE-2017-0144 trong giao thức Server Message Block, giao thức mà WannaCry đã khai thác để lây lan tới những thiết bị chưa vá lỗ hổng này. Sau khi khai thác được lỗ hổng này,nó không mã hóa các tập tin dữ liệu của người dùng mà thay đổi Master Boot Record (MBR) và mã hóa Master File Table (MFT) khiến thiết bị không thể khởi động cũng như mã hóa các file khác. Khi người dùng không thể kết nối với hệ thống được nữa, nó sẽ gửi một tin nhắn hướng dẫn các hồi phục lại hệ thống, yêu cầu trả tiền chuộc bằng Bitcoin. Điều này khiến cho hệ điều hành không thể định vị được vị trí để giải mã được file. Bên cạnh đó Petya còn sử dụng kĩ thuật ăn cắp thông tin xác thực để lây lan tới các thiết bị khác khó bị tổn thương hơn. Biến thể mới của Petya sử dụng phương pháp lây lan tương tự như WannaCry nên nói không quá rằng Petya là bản nâng cấp WannaCry.



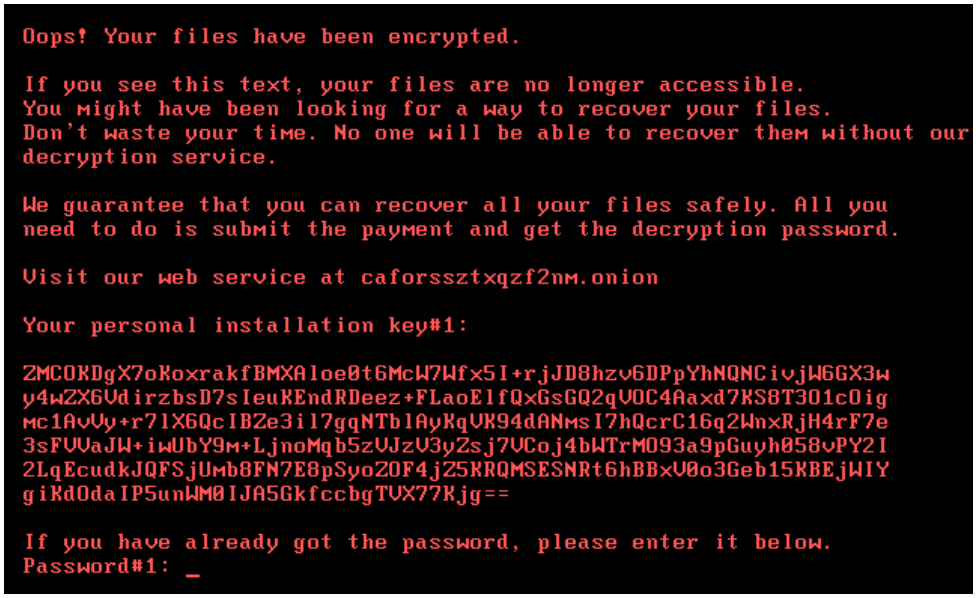
Cách hoạt động của Petya



Thông báo nhiễm mã độc Petya

Khi mã độc Petya đã hoành hành được một thời gian thì một biến thể khác của nó đã được ra mắt thông qua một vụ tấn công mạng với hơn 80 quốc gia, mục tiêu chủ yếu nhắm vào Ukraine vào tháng 6/2017 với tên gọi NotPetya, sử dụng lại chính lỗ hổng EternalBlue đã được WannaCry khai thác trước đây. Bên cạnh đó, Petya và NotPetya có cách mã hóa và reboot khác nhau. Tuy NotPetya là một mã độc tống tiền, nhưng nó đã được sửa đổi để không thể thu hồi các thay đổi mà nó đã gây ra, do đó độ nguy hiểm cao hơn nhiều so với Petya. Các chuyên gia bảo mật đến từ Google cáo buộc chính phủ Nga, đặc biệt là nhóm Sandworm đứng sau vụ tấn công này.

4. Bad Rabbit
   Bad Rabbit bắt đầu xuất hiện và lây lan từ các nước Đông Âu, chủ yếu ở Nga và Ukraine từ ngày 24/10/2017. Bad Rabbit có những điểm tương đồng với WannaCry và Petya.



Thông báo nhiễm mã độc BadRabbit

   Bad Rabbit được ngụy trang dưới hình thức chương trình cài đặt Adobe Flash khi người dùng lướt qua các trang web độc hại với đường link sau khi được điều hướng là `hxxp://1dnscontrol.com/flash_install.php`. Mã độc Bad Rabbit được nhúng vào mã nguồn JavaScript của các trang web trên.
   Khi đã tải về các bản Adobe Flash giả, mã độc sẽ được lưu trong thiết bị với đường dẫn C:/Windows/infpub.dat và chạy mã độc bằng cách sử dụng rundll. Mã độc sử dụng Mimikatz để tập hợp các chứng chỉ của thiết bị cũng như chứa các chứng chỉ bị mã hóa và sau đó sử dụng chúng để lây lan tới các thiết bị khác qua hệ thống mạng. File infpub.dat sẽ mã hóa các file trong thiết bị như ransomware bình thường, tải về file dispci.exe mã hóa các ổ đĩa ngăn chặn quá trình bootloader nhằm khởi động lại thiết bị.

Ngoài các vụ tấn công tiêu biểu nói trên, còn có một vài vụ tấn công ransomware khá nổi tiếng trên thế giới như Reveton (2012), CryptoLocker (2013), CryptoWall (2014), TorrentLocker (2014), Fusob (2015), SamSam (2016). Số tiền thiệt hại mà những phần mềm này gây ra lên tới hàng triệu USD trên toàn cầu.

2.4.5 Cách xử lí khi bị dính ransomware

1. Không trả tiền chuộc cho tội phạm dưới bất kì hình thức nào
   Mục đích quan trọng nhất của tội phạm khi cài ransomware là tiền. Càng hoảng loạn mà trả tiền chuộc, tội phạm càng có cơ hội lộng hành. Bên cạnh đó, việc trả tiền chuộc không đảm bảo việc dữ liệu sẽ được trả lại. Do đó, cần bình tĩnh, không trả tiền chuộc cho tội phạm trong bất cứ hoàn cảnh nào.

2. Khi bị nhiễm ransomware, điều đầu tiên cần làm là ngắt kết nối mạng để tránh lây lan mã độc tới các thiết bị sử dụng mạng chung. Trong trường hợp máy tính không bị khóa, có thể làm một số thao tác như bật tường lửa, chạy phần mềm diệt virus hoặc tự gỡ mã độc thủ công. Mọi thứ sẽ trở nên phức tạp hơn nhiều nếu máy tính bị khóa. Nếu bạn không có kiến thức chuyên sâu về máy tính, hãy nhờ những người hoặc tổ chức có chuyên môn để giải quyết vấn đề.

3. Hiện nay, có một số phần mềm có khả năng khôi phục lại dữ liệu với một số loại ransomware nổi tiếng như No More Ransom phối hợp giữa trung tâm tội phạm mạng châu Âu của Europol, cảnh sát đơn vị tội phạm công nghệ cao quốc gia của Hà Lan, McAfee và Kaspersky, Free Ransom Decryption của Kaspersky, công cụ của Avast. Tuy nhiên, ransomware càng ngày càng tinh vi và

các loại ransomware mới ra mắt không có phần mềm có khả năng khôi phục, nguy cơ bị mất dữ liệu hoàn toàn là rất lớn. Do đó chúng ta cần cẩn thận, cảnh giác, trang bị kĩ kiến thức, kĩ năng khi đối mặt với Ransomware.

2.4.6 Cách phòng chống

- Cập nhật hệ điều hành và ứng dụng thường xuyên.
- Không click vào, không tải file từ các đường link, trang web lạ, không mở các đường liên kết từ email lạ.
- Sử dụng, cập nhật các phần mềm bảo vệ internet, bật VPN khi sử dụng internet ở các địa điểm công cộng.
- Lưu trữ dự phòng các file trên máy tính trên USB, đĩa cứng, Drive,... để bảo vệ các file khi máy tính bị tấn công

2.4.7 Tham khảo

https://cystack.net/vi/blog/ransomware-la-gi
https://www.pbs.org/newshour/nation/why-ransomware-attacks-are-on-the-rise-and-what-can-be-done-to-stop-them
https://en.wikipedia.org/wiki/Ransomware
https://www.kaspersky.com/resource-center/threats/ransomware-wannacry
https://www.acronis.com/en-us/articles/gandcrab/
https://www.mcafee.com/enterprise/en-us/security-awareness/ransomware/petya.html
https://securelist.com/bad-rabbit-ransomware/82851/
https://www.proofpoint.com/us/threat-reference/bad-rabbit

2.5 CoinMiner

2.5.1 Giới thiệu

CoinMiner là một phần mềm độc hại lạm dụng nguồn tài nguyên máy tính nhằm thực hiện các chương trình đào tiền ảo. Chúng chiếm tài nguyên (CPU, GPU, RAM, băng thông, pin, ...) mà không để chủ nhân của máy tính biết. Với sự phát triển của các đồng tiền điện tử, các tội phạm mạng nhìn ra cơ hội để xâm nhập vào hệ thống của một tổ chức và bí mật đào tiền ảo.

2.5.2 Các dạng mã độc

- File thực thi: Là những dạng mã độc đặc trưng hoặc các file chứa chương trình không mong muốn tiềm ẩn (PUA) có đuôi .exe ở trên máy tính và được thiết kế để đào tiền ảo.
- Mã độc dạng trình duyệt: Các công cụ đào tiền ảo viết bằng JavaScript này (hay những ngôn ngữ tương đương) thực hiện công việc trên một trình duyệt web, hao tốn tài nguyên khi trình duyệt mở ở trên website. Một số công cụ được sử dụng cố ý bởi chủ website ở nơi chạy quảng cảo (ví dụ CoinHive), trong khi một số khác đã được đưa vào các trang web chính thống mà chủ nhân website không hề biết.
- Mã độc không dấu vết nâng cấp: Mã độc thực hiện đào tiền ảo bằng cách làm các chương trình thực hiện sai mục đích bình thường (ví dụ MSH.Bluwimps), nhằm chèn thêm các lệnh đào tiền ảo.

2.5.3 Cách tấn công

Rất nhiều thiết bị nhiễm virus theo các đường sau:

- Email kèm tập tin đính kèm có chứa mã độc.
- Web hosting sử dụng các công cụ exploit để khai thác các lỗ hổng trong trình duyệt web và các phần mềm khác
- Các trang web chiếm quyền sử dụng máy tính bằng cách chạy các đoạn mã khi người dùng truy cập vào trang web

Đào coin là quá trình thực hiện các thuật toán phức tạp nhằm duy trì sổ cái blockchain. Quá trình này tạo ra coin nhưng cần đòi hỏi nhiều tài nguyên của máy tính.
CoinMiner vốn dĩ không phải là mã độc. Một số tổ chức và cá nhân đầu tư điện năng cũng như phần cứng để đào coin chính đáng. Tuy vậy, những người khác sử dụng tài nguyên và điện năng một cách luân phiên từ các hệ thống của công ty cũng như cá nhân, điều mà chủ nhân của những hệ thống này không hề mong muốn. Các tội phạm mạng nhìn thấy cơ hội để kiếm tiền bằng cách cung cấp, tải và chạy các mã độc đào tiền ảo để hao tốn tài nguyên máy tính của người khác.

2.5.4 Cách nhận biết

- Sử dụng CPU và GPU ở mức cao.
- Thiết bị nóng hơn bình thường.
- Khởi động lại không được hoặc bất thường.
- Thời gian phản hồi yêu cầu chậm.
- Hoạt động mạng bất thường.

s

2.5.5 Cách phòng chống

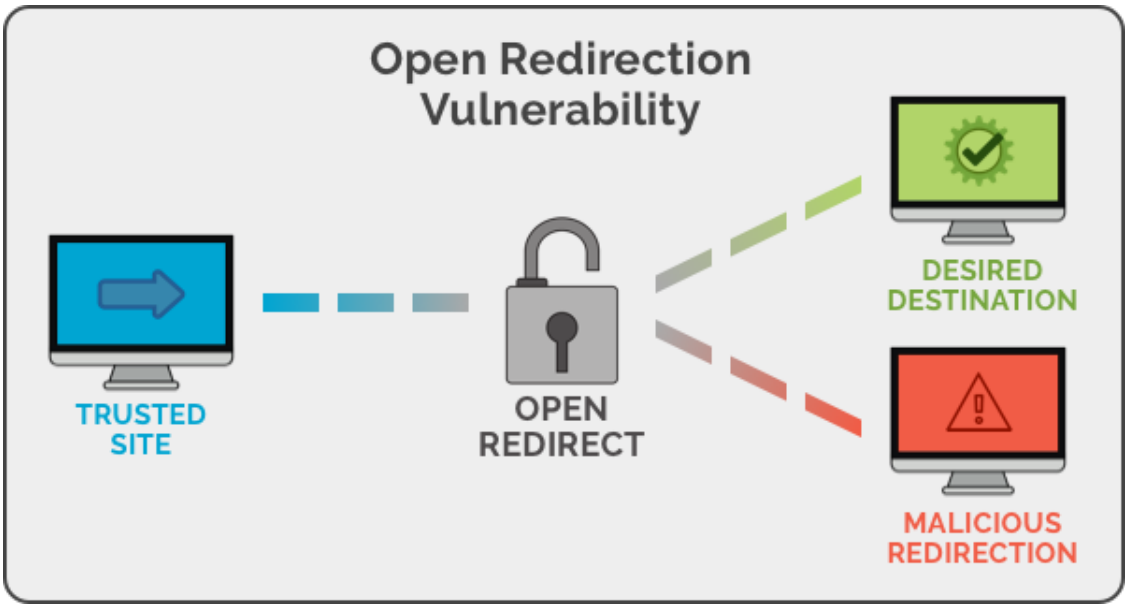
Kích hoạt ứng dụng nhận biết chương trình không mong muốn tiềm ẩn (PUA): Một số công cụ coin miner không được xem là mã độc nhưng lại được xem như PUA. Nhiều chương trình PUA gây ảnh hưởng xấu đến hiệu suất máy tính cũng như năng suất lao động.
Bên cạnh đó, do coin miner càng ngày càng trở nên phổ biến trong các cuộc tấn công. Chúng ta nên áp dụng các mẹo để phòng chống mã độc sau cho coin miner:

- Cẩn thận khi click vào các đường link hay tập tin đính kèm, coi chừng các trang web lạ
- Cập nhật các ứng dụng thường xuyên
- Lưu giữ file trên máy tính ở một nới khác
- Sử dụng mật khẩu mạnh để tránh bị mất mật khẩu

2.5.6 Tham khảo

https://support.norton.com/sp/en/us/home/current/solutions/v125881893
https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/coinminer-malware

2.6 Open Redirects



2.6.1 Giới thiệu

Open Redirects (*Chuyển hướng mở*) là phương thức kẻ tấn công sẽ chuyển người dùng sang một trang web khác có nhiều lỗ hổng về bảo mật với trang web ban đầu. Để rồi từ đó có thể thực hiện nhiều cách khác nhau để đánh cắp thông tin của người dùng. Theo xếp hạng của The Open Web Application Security Project (OWASP) thì nó nằm trong top 10 nguy cơ bảo mật web của năm 2013.

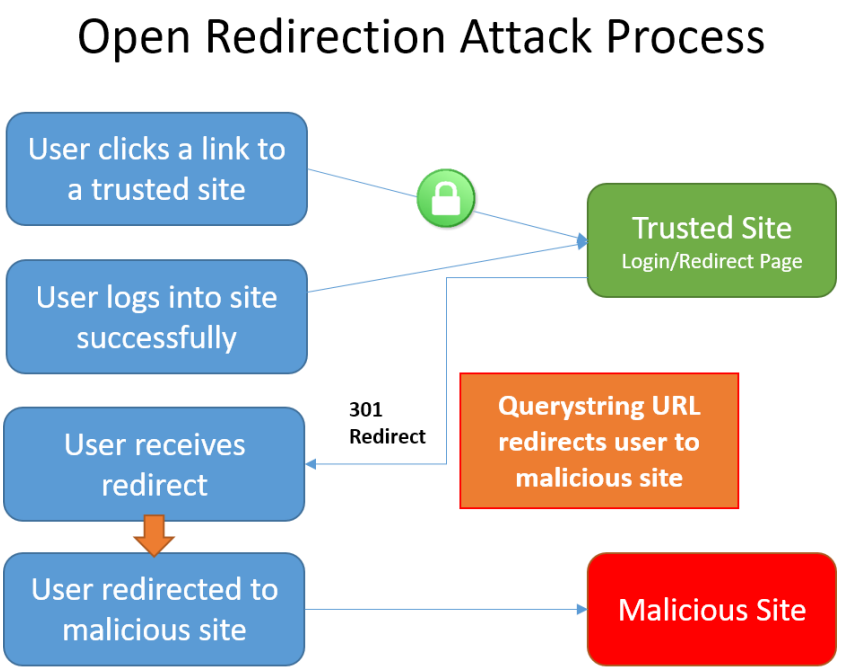
2.6.2 Cách tấn công

Redirects (*Chuyển hướng*) xảy ra khi trang web hoặc một ứng dụng web thay đổi URL được truy cập trong máy người dùng. Có một số cách để thực hiện việc này từ back-end. Thông thường, chuyển hướng được thực hiện bằng cách gửi các tiêu đề HTTP cụ thể đến máy khách nhưng bạn cũng có thể tạo chuyển hướng, ví dụ: bằng cách sử dụng mã JavaScript.

Open redirects vulnerability (lỗ hổng chuyển hướng) tồn tại khi máy người dùng được cung cấp điểm đến của chuyển hướng và nó không được lọc hoặc xác thực. Dưới đây là một số ví dụ về chuyển hướng an toàn và chuyển hướng không an toàn:

- Nếu trang web hợp pháp chuyển hướng khách hàng đến một URL cố định, đó là một chuyển hướng an toàn.
- Nếu trang web hợp pháp xây dựng URL chuyển hướng một cách an toàn dựa trên các thông số do người dùng cung cấp thì đó là chuyển hướng an toàn.
- Nếu trang web hợp pháp xây dựng URL chuyển hướng dựa trên các tham số do người dùng cung cấp nhưng không đủ xác thực hoặc lọc đầu vào thì đó là chuyển hướng không an toàn (kẻ tấn công có thể thao túng đầu vào).
- Nếu trang web hợp pháp cho phép người dùng chỉ định URL chuyển hướng đích thì đó là chuyển hướng không an toàn.

Mặc dù có vẻ như là một hành động vô hại khi cho phép người dùng quyết định nơi họ muốn được chuyển hướng nhưng điều đó dẫn tới việc các kẻ tấn công có thể thao túng và chuyển hướng người dùng từ trang web này sang trang web khác. Kỹ thuật này có thể có tác động nghiêm trọng đến bảo mật ứng dụng, đặc biệt là khi kết hợp với các lỗ hổng và thủ thuật khác nhưng cộng đồng an ninh mạng không chú trọng đủ đến vì coi đây chỉ là một lỗ hổng đơn giản thường được kết nối với các mưu đồ lừa đảo và kỹ thuật xã hội.



Ví dụ về lỗ hổng chuyển hướng như sau bạn đang truy cập trang web "example.com" và kẻ tấn công có thể tạo ra URL như sau https://example.com/redirect.php?url=http://attacker.com. Từ đó kẻ tấn công có thể gửi URL này cho các đối tượng sử dụng trang web "example.com và chuyển hướng họ sang trang "attacker.com". Vì nhìn thấy URL là "https://example.com" nên một số người dùng tin tưởng mà không hay biết mình đang vào trang web bảo mật kém.

2.6.3 Hậu quả

Nếu chúng ta bị tấn công bởi open redirection thì nó có thể dễ dàng dẫn tới các kiểu tấn công khác như

- **Phishing:** Điều dễ thấy và rõ ràng nhất để sử dụng open redirect là khiến nạn nhân rời khỏi trang web ban đầu đến một trang web giống hệt nhằm mục đích đánh cắp thông tin đăng nhập của người dùng, sau đó quay lại trang web dễ bị ban đầu mà nạn nhân không nhận thấy.
- **Cross-site Scripting (XSS):** Nếu chuyển hướng cho phép sử dụng các giao thức dữ liệu: hoặc javascript: và máy khách hỗ trợ các giao thức như vậy trong chuyển hướng, thì kẻ tấn công có thể thực hiện một cuộc tấn công XSS.
- **Server-Side Request Forgery (SSRF):** Chuyển hướng mở có thể được sử dụng để tránh các bộ lọc SSRF.
- **Content-Security-Policy bypassing:** Nếu bạn sử dụng CSP để bảo vệ chống lại XSS và một trong các miền thuộc danh sách trắng có chuyển hướng mở, thì lỗ hổng này có thể được sử dụng để bỏ qua CSP.
- **CRLF Injection:** Nếu tham số chuyển hướng cho phép ngắt dòng, kẻ tấn công có thể cố gắng thực hiện tách tiêu đề phản hồi.

2.6.4 Cách phòng chống

Không cho phép chuyển hướng ngoại tuyến Bạn có thể ngăn chuyển hướng đến các miền khác bằng cách kiểm tra URL được chuyển đến chức năng chuyển hướng. Đảm bảo rằng tất cả các URL chuyển hướng đều là đường dẫn tương đối - tức là chúng bắt đầu bằng một ký tự /. (Lưu ý rằng các URL bắt đầu bằng // sẽ được trình duyệt hiểu là một URL tuyệt đối, bất khả tri về giao thức - vì vậy chúng cũng nên bị từ chối.)

Nếu bạn cần thực hiện chuyển hướng bên ngoài, hãy xem xét đưa vào danh sách trắng các trang web riêng lẻ mà bạn cho phép chuyển hướng đến.

Kiểm tra liên kết giới thiệu khi thực hiện chuyển hướng Các trang trên trang web của bạn chỉ nên kích hoạt chuyển hướng đến các URL được chuyển trong tham số truy vấn. Bất kỳ trang web nào khác kích hoạt chuyển hướng cần được xử lý hết sức nghi ngờ. Là lớp bảo vệ thứ hai, hãy kiểm tra xem Người giới thiệu trong yêu cầu HTTP có khớp với miền của bạn bất cứ khi nào bạn thực hiện chuyển hướng hay không.

2.7 File Vulnerability

Figure 11: *Tấn công bằng file*

2.7.1 Định nghĩa:

Trước khi tìm hiểu các lỗ hổng bảo mật về file, ta sẽ xem web shell là gì trước.

Webshell:

+ Là một dạng mã độc có nhiều chức năng hỗ trợ hacker chiếm quyền điều khiển server, hay quản trị web.

+ Web shell thường được viết bằng ngôn ngữ cùng vơi ngôn ngữ của web đó. Phổ biến hiện nay nhất là shell web PHP, khi đa phần các trang web được viết trên template wordpress có sẵn. Phần này chủ yếu đa phần tìm hiểu về các lỗ hổng đến file php.

+ Đặc biệt chúng khó có thể bị phát hiện bởi phần mềm diệt virus hay tường lửa bởi vì chúng không phải một file mà chỉ là một shell script. Chỉ cần tải được các tệp tin này lên máy chủ thì tức máy chủ đã bị điều khiển dù ta không cần biết mật khẩu hay tài khoản, cũng như không phải là root, admin.

File Vulnerability:

Là một lỗ hổng phổ biến khi ta dùng các file chứa web shell để tải lên trình duyệt, thông qua đó chiếm quyền điều khiên thực hiện mục đích của hacker. Có hai loại File Vulnerability phổ biến đó là:

- **File Upload Vulnerability:** Các ứng dụng web hiện nay, hầu như đều có chức năng upload ảnh, upload post, nhưng nếu việc validate được thực hiện ở front-end và phía back-end kiểm tra không chặt, qua đó chúng ta có thể biến mã độc thành "ảnh", thành "post" để có thể upload lên chiếm quyền điều khiển web.

- **File Inclusion Vulnerability:**

  + Cho phép Hacker có thể tấn công từ xa mà không cần nhìn thấy cấu trúc tệp của server hoặc tấn công tập trung vào một mục tiêu trên trang web.

  + Lỗi này thường xảy ra khi trong code php có yêu cầu chứa include, require, include_once, require_once cho phép gọi một file khác. Trong các trang web, để nhận biết lỗi này, ta có thể để ý vào url của giao thức https thường có đuôi chứa php?page=, php?file=, php?index=,... ta có thêm dấu ' vào cuối để kiểm tra thử:

+ Nếu hiển thị dòng này, tức web của bạn đã bị lỗi này:

```
Warning: Warning: include() [function.include]: Failed opening ''' for
inclusion (include_path='.;C:\php5\pear') in C:\wamp\www\FI.php on line 40
```

2.7.2 Cách tấn công:

File Upload Vulnerability: Một số web có chức năng upload file thường có filter để ngăn chặn qua các lỗ hổng này, để có thể tấn công chúng ta phải bypass qua các filter đó.

- **Client side filters** Đây là một kiểu xác thực được thực hiện ở máy khách, kiểm tra bởi phía front-end, nhằm yêu cầu người dùng xác thực đúng định dạng, chẳng hạn như khi input email phải có @, ảnh phải có đuôi png, jpg, url phải có tên miền cũng như giao thức,.. filer được hỗ trợ bởi các ngôn ngữ html5, java script. Để bypass qua Filter này, ta có thể sử dụng những cách như sau:
  + Tắt java script của trình duyệt thông qua Development Tool Kit (DTK)
  + Giả mạo file gửi lên trang web (chẳng hạn ta muốn gửi file php nhưng web yêu cầu upload ảnh, vậy ta chỉ cần addon thêm đuôi jpg vào là được)
  + Giả mạo request gửi lên trang web.
- **The extension Black listing:** Blacklist như tên gọi của nó, là một danh sách các từ khóa, đuôi bị cấm, extension này được sử dụng để lọc ra các file vi phạm không đúng định dạng, không đúng yêu cầu. Nhưng việc thầu hết toàn bộ các từ khóa bị cấm hầu như không thế, ta có một số cách để bypasss qua filter này, thông qua ví dụ về PHP shell Web.
  + **Đổi đuôi extension:** nếu .php bị regex phát hiện, ta có thể đổi thành .php1, .php2, .ph3, ngay cả đuôi như .p1, .cgi, shell vẫn chạy bình thường
  + **Đổi kiểu chữ, mã chữ:** thử xem bộ lọc có phân biệt chữ hoa hay chữ thường không, ta có thể đổi thành .pHP, .PHP, và trên nhiều loại mã, unicode, utf8 kết hợp cùng cách 1.
  + **Chồng extension:** ta cũng có thể để đuôi đi kèm với nhau, .jpg.php, .php.jpg,...
- **The extension White listing:** Trái ngược với Blacklist, whitelist là tập hợp các đuôi, các từ khóa hợp lệ, filter chỉ chấp nhận duy nhất những đuôi đó chẳng hạn như: .png, .jpg, .gif, .jpeg,...
  + **Null Byte Injection:** các ký tự nullbyte thường là 0x00 trong hex, %00, 00%, điểm đặc biệt các kí tự cùng kí tự nullbyte này sẽ biến mất trong quá trình tải file, ví dụ nếu ta không kiểm tra kĩ file: shell.php%00.jpg thì khi tải lên phần từ nullbyte trở về sau sẽ biến mất để lại shell.php tải thành công.
  + **Chồng extension như blacklist**
  + **Invalid Extension Bypass:** ngoài ra có một số lỗi từ web sever, nếu chúng ta sử dụng đuôi có extension là .test, thì hệ điều hành sẽ không nhận ra. Như vậy hệ điều hành sẽ bỏ qua kiểm tra file này mà thực thi luôn.
- **The content length and malicious script checks:** Có một số trang khá ít thường kiểm tra độ dài file upload, như thế, chúng ta sẽ sử dụng lệnh shell ngắn để bypass như:

```
<?system($_GET[0]);
```

File Inclusion Vulnerability:

RFI (Remote File Inclusion): xảy ra khi ứng dụng web tải xuống và thực thi một tệp từ xa. Các tệp từ xa này thường được lấy ở dạng HTTP hoặc FTP URI dưới dạng tham số do người dùng cung cấp cho ứng dụng web. việc sử dụng hàm include rất nhiều và cũng là thiết đặt mặc định của server như là set allow_url_include = On, allow_url_open = On. Lỗ hổng này sẽ khiến kẻ tấn công có thể thực thi các lệnh từ xa trên máy chủ web , xoá các phần của web và lấy dữ liệu thông tin của trang web. Ví dụ với url dưới đây:

```
http://localhost/DVWA/vulnerabilities/fi/index.php?page=
```

Ta sẽ thử trang web này, có tải lên một trang web khác hay không, ví dụ là: http://www.google.com

```
http://localhost/DVWA/vulnerabilities/fi/index.php?page=http://www.google.com
```

Nếu thành công như hình dưới đây tức ta có thể tải lên các đoạn script gây hại.

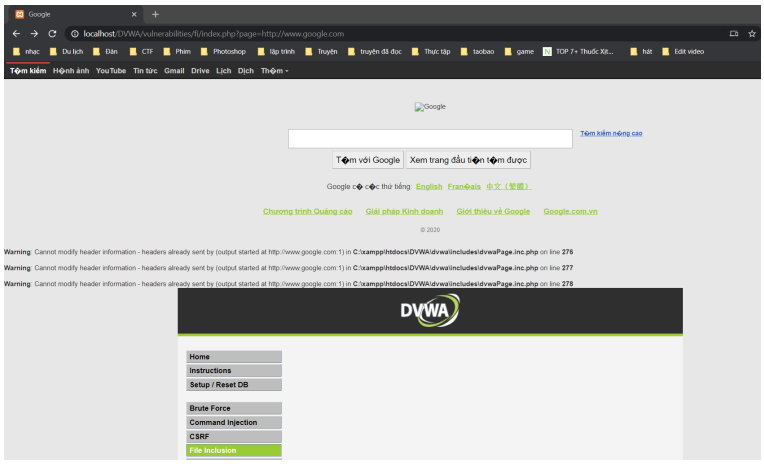


Figure 12: *Dấu hiệu bị lỗ hổng file - tải code một trang web khác lên được.*

Ta có thể tải đoạn script gây hại trong file script.html như sau:

```
http://localhost/DVWA/vulnerabilities/fi/index.php?page=http://localhost:80/script.html
```

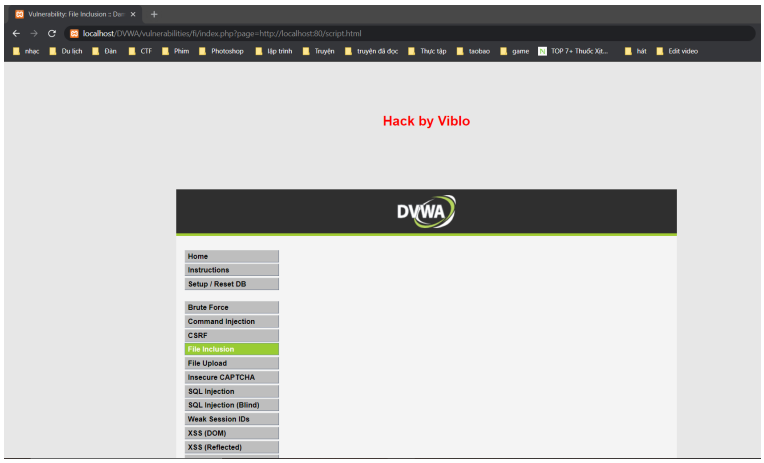


Figure 13: *Web bị hack bởi script.html*

Tuy nhiên một số trang web trở về sau, đều có bộ lọc thông qua hàm str_replace() để thay đổi các chuỗi kí tự "http://" và "https://" hay "../" thành "" dựa vào đầu nhập được thêm vô. Ta không thể viết bình thường như trên, vậy để tránh việc thay đổi đó, ta có thể thay đổi kí tự mới thành:

```
htthttp://p://www.google.com
```

Như thế khi bộ lọc thực hiện xong, sẽ trả lại giá trị ban đầu cho ta.

LFI (Local File Inclusion): tương tự như RFI, nhưng khi allow_url_include = Off, chúng ta không thể khai thác qua url thông tin từ xa, lúc này khai thác sẽ dựa trên việc đưa vào tệp cục bộ. Khai thác tệp cục bộ cho phép chúng ta đọc các tệp cảm biến trên máy chủ, ví dụ như là / etc / passwd, / etc / group, httpd.conf, .htaccess, .htpasswd hoặc bất kì file có cấu hình quan trọng nào. Phương thức này thường kết hợp với bypass null injection,

+ Với /etc/passwd: ta có thể tấn công như thế này: tại sao lại có nhiều ../ như vậy, để đoán biết được đường dẫn thư mục ở đâu, này chúng ta phải mò từ từ. Như vậy đọc được rồi, chúng ta sẽ biết mật khẩu để từ đó có thể chiếm quyền quản trị root.

```
/vulnerable.php?language=../../../../../etc/passwd\%00,
```

+ Tương tự như /etc/passwd với httpd.conf: Thực hiện đọc file này để có được thông tin về error_log, access_log, ServerName, DocumentRoot, ...

+ Với proc/self/environ%00, . Kẻ tấn công có thể sửa đổi tiêu đề HTTP (chẳng hạn như Tác nhân người dùng) trong cuộc tấn công này thành mã PHP để khai thác việc thực thi mã từ xa lên các máy local khác.

```
/vulnerable.php?language=../../../../../proc/self/environ
```

2.7.3 Tác hại:

- Hacker thông qua các webshell được tải lên web, có thể nắm quyền làm chủ hệ thống từ đó thực hiện các hành vi phạm pháp như tống tiền, chỉnh sửa, ăn cắp thông tin,...
- Nhiều máy tính cá nhân sử dụng chung một dịch vụ web, app đó đều có thể bị ảnh hưởng bởi mã độc.
- Là nguyên nhân dẫn đến các cuộc tấn công khác:
  + Code execution on the web server
  + Cross Site Scripting Attacks (XSS)
  + Denial of service (DOS)
  + Data Manipulation Attacks

2.7.4 Cách phòng chống:

+ Set giá trị allow_url_fopen và allow_url_include thành off để giới hạn việc có thể gọi các tệp tin từ xa.

+ Không cho trình phân tích thư mục như "/".

+ Xác thực đầu vào chặt hơn.

+ Giới hạn loại tệp tải lên.

+ Giới hạn quyền của thư mục. Tức tệp tải lên chỉ có những quyền hạn nhất định và tự động loại bỏ nếu cần.

+ Thường xuyên cập nhật thư viện, framework lên phiên bản mới nhất.

+ Tránh việc sử dụng các biến global variable, (biến này dễ bị tấn công).

2.7.5 Sự kiện tiêu biểu:

Cách tấn công này thường đi kèm với các cách tấn công khác, nên sự kiện tiêu biểu đó cũn gắn liền với những cách tấn công đó.

2.7.6 Tham khảo:

+ https://viblo.asia/p/file-inclusion-vulnerability-exploit-4P856NMa5Y3
+ https://viblo.asia/p/khai-thac-cac-lo-hong-file-upload-phan-1-aWj53L6pK6m
+ https://whitehat.vn/threads/web-pentest-bai-7-tim-kiem-cac-lo-hong-lien-quan-den-file-upload.15214/
+ https://www.junookyo.com/2011/12/tan-cong-file-inclusion-file-inclusion.html
+ https://securitydaily.net/tan-cong-file-inclusion/
+ https://en.wikipedia.org/wiki/File_inclusion_vulnerability

2.8 Backdoor

2.8.1 Giới thiệu

Backdoor là một loại malware vượt qua các quy trình xác thực thông thường để truy cập trái phép hệ thống. Nói đơn giản, Backdoor là một đoạn mã cho phép ra vào hệ thống mà không bị phát hiện. Khi xâm nhập hệ thống, kẻ tấn công có thể đánh cắp thông tin; cài đặt thêm malware; đưa ra các lệnh hệ thống; chiếm quyền điều khiển thiết bị; phát động các cuộc tấn công như: Distributed denial of service (DDoS), Watering hole attack, Advanced persistent threat (APT)...

Figure 14: *Backdoor*

Trong các cuộc tấn công, Backdoor là phần mềm độc hại thường trú và đợi lệnh điều khiển từ các cổng dịch vụ TCP hoặc UDP. Backdoor khi chạy trên máy bị nhiễm, nó sẽ thường trực trong bộ nhớ và mở một cổng cho phép kẻ tấn công truy nhập vào máy nạn nhân thông qua cổng mà nó đã mở và kẻ tấn công có toàn quyền điều khiển máy bị nhiễm. Backdoor nguy hiểm ở chỗ nó hoàn toàn chạy ẩn trong máy. Nhiều Backdoor được hẹn trước giờ để kết nối ra ngoài (đến 1 giờ nhất định mới mở 1 port để hacker đột nhập vào) nên rất khó phát hiện ngay cả scan port.
Có nhiều loại Backdoor khác nhau có thể được tạo, và không phải tất cả chúng đều có mục đích xấu:

- Built-in (hoặc Proprietary) Backdoor được tạo bởi các nhà phát triển hoặc nhà cung cấp dịch vụ với mục đích chính là sửa chữa phần mềm. Tuy nhiên, chúng cũng có thể bị kẻ xấu khai thác để lấy quyền truy cập.
- Backdoor malware được tạo bởi những kẻ xấu.

Bất kỳ malware nào cung cấp cho hacker quyền truy cập vào thiết bị của bạn đều có thể được coi là Backdoor - điều này bao gồm Rootkit, Trojan, Spyware, Cryptojackers, Keylogger, Worm và thậm chí là Ransomware.
Nhiệm vụ chính của backdoor là lấy thông tin người dùng đang sử dụng phần mềm. Sau đó thực hiện hành động nào đó, ví dụ gửi các thông tin này lưu trữ lên server, hay còn được biết đến là đánh cắp thông tin người sử dụng phần mềm. Như vậy có thể thấy thực chất backdoor chính là việc trao đổi dữ liệu giữa người dùng phần mềm và server.

2.8.2 Cách tấn công

Để tội phạm mạng cài đặt thành công Backdoor trên thiết bị của bạn, trước tiên chúng cần có quyền truy cập vào thiết bị của bạn, thông qua truy cập vật lý, tấn công bằng malware hoặc bằng cách khai thác lỗ hổng hệ thống - đây là một số lỗ hổng phổ biến hơn mà hacker nhắm mục tiêu :

- Cổng mở (Open ports)
- Mật khẩu yếu.
- Phần mềm lỗi thời.
- Tường lửa yếu.

"Khai thác" là các cuộc tấn công có chủ đích lợi dụng các lỗ hổng phần mềm (thường là trong phần mềm chạy trên web như trình duyệt, Adobe Flash, Java, v.v.) để cung cấp cho hacker quyền truy cập vào hệ thống của bạn. Hacker có thể tạo ra các trang web và quảng cáo độc hại quét máy tính của bạn để tìm lỗ hổng phần mềm và sử dụng các hành vi khai thác để thực hiện những việc như lấy cắp dữ liệu, làm hỏng mạng hoặc cài đặt Backdoor trên thiết bị của bạn
. Vì vậy, khi một tệp malware lây nhiễm vào thiết bị của bạn, hoặc thiết bị của bạn bị đánh cắp hoặc hư hỏng hoặc bạn trở thành mục tiêu của một cuộc tấn công khai thác, hacker có thể cài đặt một Backdoor trên hệ thống của bạn. Dưới đây là một vài ví dụ về các loại Backdoor khác nhau thường được sử dụng:

- Trojan là các tệp malware giả vờ là các tệp an toàn để giành quyền truy cập vào thiết bị của bạn. Sau khi bạn nhấp vào "allow insert-program-here to make changes on your device?" trên PC của bạn, Trojan sau đó có thể tự cài đặt trên thiết bị của bạn. Trojan Backdoor có thể cho phép người dùng truy cập các tệp và chương trình của bạn hoặc cài đặt các tệp malware nghiêm trọng hơn trên thiết bị của bạn.

- Rootkit là các mối đe dọa malware phức tạp có khả năng che giấu các hoạt động của chúng khỏi hệ điều hành để hệ điều hành cấp đặc quyền bảo mật (quyền truy cập root) cho Rootkit. Rootkit có thể cho phép hacker truy cập từ xa vào thiết bị của bạn, thay đổi tệp của bạn, quan sát hoạt động của bạn và phá hoại hệ thống của bạn. Rootkit có thể ở dạng phần mềm hoặc thậm chí là chip máy tính bị thay đổi.

- Backdoor phần cứng là các chip máy tính đã được sửa đổi hoặc phần cứng khác cung cấp cho người ngoài quyền truy cập vào một thiết bị. Điều này có thể bao gồm điện thoại, thiết bị IoT như bộ điều nhiệt và hệ thống an ninh gia đình, bộ định tuyến và máy tính. Các Backdoor phần cứng có thể giao tiếp dữ liệu người dùng, cung cấp quyền truy cập từ xa hoặc được sử dụng để giám sát. Các Backdoor phần cứng có thể được vận chuyển cùng với các sản phẩm (bởi một nhà sản xuất giả mạo hoặc vì một số mục đích lành mạnh), nhưng chúng cũng có thể được cài đặt khi thiết bị bị đánh cắp.

- Backdoor mật mã về cơ bản là một "siêu chìa khóa" có thể mở khóa mọi phần dữ liệu được mã hóa sử dụng một giao thức mã hóa cụ thể. Các tiêu chuẩn mã hóa như AES sử dụng mã hóa đầu cuối để chỉ các bên đã trao đổi khóa mật mã được tạo ngẫu nhiên mới có thể giải mã thông tin được chia sẻ. Backdoor là một cách để phá vỡ cuộc trò chuyện an toàn này, thao túng bài toán phức tạp của một giao thức mật mã cụ thể để cung cấp cho người dùng bên ngoài quyền truy cập vào tất cả dữ liệu được mã hóa đang được chia sẻ giữa các bên.

2.8.3 Sự kiện tiêu biểu

DoublePulsar Cryptojacker: Vào năm 2017, các nhà nghiên cứu bảo mật đã phát hiện ra rằng Backdoor malware DoublePulsar (ban đầu được phát triển bởi NSA, Cơ quan An ninh Quốc gia Hoa Kỳ) đang được sử dụng để theo dõi PC Windows, cài đặt một cryptojacker(một malware đào tiền điện tử) trên máy tính có đủ bộ nhớ và sức mạnh CPU. Cryptojacker đã đánh cắp sức mạnh xử lý từ các máy tính bị nhiễm virus để khai thác Bitcoin, bí mật kết hợp hàng nghìn PC vào một mạng botnet khai thác tiền điện tử khổng lồ.

Microsoft Azure: Tháng 11/2021, Microsoft bảo vệ các máy chủ Azure ở châu Á thành công trước đợt tấn công DDoS kỷ lục 3,47 Tbps. Đợt tấn công này kéo dài tầm 15 phút và bắt nguồn từ khoảng 10.000 nguồn trên khắp thế giới, phối hợp nhiều phương pháp tấn công khác nhau bao gồm SSDP, CLDAP, DNS, NTP.

2.8.4 Cách giải quyết

Nên thực hiện một số hành động sau ngay lập tức nếu nghi ngờ một cuộc tấn công Backdoor:

- Tuyệt đối không sử dụng các phần mềm không đáng tin cậy. Ngoài ra, không truy cập vào các website nguy hiểm, không cài các ActiveX và JavaScript trên các website đó, bởi chúng có thể sẽ đính kèm Trojans.

- Xem lại nhật ký ứng dụng website scanner để xác định bất kỳ tệp nào liên tục bị xóa.

- Yêu cầu nhà cung cấp mạng hoặc nhóm chuyên về IT xem xét nhật ký truy cập website để tìm bất kỳ điều gì bất thường.
- Kiểm tra Content Management System (CMS) và gỡ cài đặt bất plugin nào không sử dụng.
- Cập nhật tất cả các plugin trên website.
- Backup phiên bản trước đó. Nếu không thể tìm thấy Backdoor của cuộc tấn công, giải pháp cuối cùng là phục hồi về phiên bản trước đó của website.

2.8.5 Tham khảo

+ https://www.safetydetectives.com/blog/what-is-a-Backdoor-and-how-to-protect-against-it/

+ https://viblo.asia/p/what-is-a-Backdoor-attack-bWrZnmpvKxw

+ https://thinkview.vn/microsoft-thanh-cong-vo-hieu-hoa-dot-tan-cong-ddos-lon-nhat-lich-su-3824.html?fbclid=IwAR3S05z9h1trE5_PR2fjcWGUa_dv9rbFLFtS9iEZ-5f5Ig445tJo0ysOWzM

+ https://ictnews.vietnamnet.vn/bao-mat/backdoor-la-gi-va-cach-phong-ve-hieu-qua-tren-khong-gian-mang-273549.html

.

2.9 Spyware

2.9.1 Giới thiệu

+ **Spyware** được coi là một loại phần mềm độc hại được tạo ra để theo dõi các hoạt động và thông tin cá nhân của người dùng một cách trái phép.

+ Spyware chạy ngầm trong hệ thống và lưu lại những hoạt động của người dùng, đánh cắp thông tin cá nhân hoặc tệ hơn là những thông tin nhạy cảm, sau đó, gửi đến các công ty, nhà quảng cáo nhằm thu lại lợi nhuận hoặc một mục đích xấu xa nào đó.

+ Một số loại Spyware còn có thể lấy cắp những mật khẩu được lưu trong máy tính.

2.9.2 Lịch sử

+ **Ngày 16-10-1995:** Thuật ngữ "Spyware" lần đầu tiên xuất hiện trên một bài báo với ý nghĩa chế nhạo mô hình kinh doanh của Microsoft.

+ **Năm 1999:** Định nghĩa Spyware - phần mềm gián điệp đã xuất hiện trên báo chí, thu hút mạnh mẽ truyền thông đại chúng.

+ **Tháng 06-2000:** Từ "Spyware" được sử dụng bởi chủ sở hữu ZoneLabs - nhà sản xuất phần mềm diệt virus ZoneAlarm - ứng dụng phát hiện và chống lại Spyware đầu tiên.

+ **Năm 2006:** Phần mềm gián điệp Spyware ngày càng phổ biến và trở thành mối đe dọa lớn cho toàn cầu, đặc biệt là các máy tính chạy Windows và Internet Explorer. Nguyên nhân chính là do sự tích hợp giữa Windows và Internet Explorer cho phép lấy dữ liệu nơi quan trọng nhất của hệ điều hành.

2.9.3 Cách thức xâm nhập

+ **Lỗ hổng bảo mật (Security vulnerabilities):** Spyware xâm nhập thông qua các lỗ hổng bảo mật khi tải xuống, các website độc hại, các cửa sổ bật lên (pop-up),...

+ **Các phần mềm mang "tiện ích" (Misleading marketing):** Ngày nay, các phần mềm "dọn rác", "tăng tốc" thiết bị trở thành vỏ bọc hoàn hảo cho Spyware tự do xâm nhập. Kể cả khi bị xóa bỏ khỏi thiết bị, Spyware vẫn có thể hoạt động trong hệ thống.

+ **Thông qua Trojans:** Tương tự như cách thức phía trên, Trojans chính là những phần mềm độc hại trong vỏ bọc rất bình thường. Ngày nay, hầu hết Trojans không trực tiếp gây hại cho máy tính mà gián tiếp bằng cách phân tán các crypto jackers, ransomwares, viruses,...

+ **Spyware trên điện thoại di động (Mobile device spyware):** Khi các thiết bị di động trở nên phổ biến cũng chính là lúc các Spyware dành riêng cho loại thiết bị này xuất hiện. Spyware trên điện thoại di động thường là các ứng dụng chứa mã độc, các ứng dụng được đặt tên giả, đường liên kết tải xuống giả mạo,...

2.9.4 Các loại Spyware

+ **Password stealers:** là loại Spyware ăn cắp mật khẩu từ máy tính bị nhiễm. Các loại mật khẩu bao gồm các mật khẩu được lưu trong trình duyệt, các mật khẩu lưu trong hệ thống,...

+ **Banking Trojans:** loại Spyware này tập trung khai thác lỗ hổng các hệ thống, website, giao dịch từ các tổ chức tài chính, ngân hàng, công ty giao dịch điện tử,.... Phần lớn, Banking Trojans thay đổi hoặc thêm trái phép các nội dung giao dịch (transaction content).

+ **Infostealers:** chủ yếu khai thác thông tin cá nhân từ tên người dùng, mật khẩu, địa chỉ email đến thông tin hệ thống như file log,... Ngoài ra, giống như những Trojan khác, Infostealers cũng có thể khai thác lỗ hổng bảo mật của trình duyệt, từ đó truy cập những thông tin quan trọng khác của người dùng.

+ **Keyloggers:** theo dõi những thao tác trên bàn phím của người dùng.

2.9.5 Mục tiêu của Spyware

+ Những hacker phát tán Spyware với ý đồ tạo ra một mạng lưới thu thập càng nhiều dữ liệu cá nhân càng tốt, vì vậy tất cả người dùng đều có thể trở thành nạn nhân của Spyware.

+ Từ các thông tin lấy được, các hacker có thể giao dịch với những tổ chức quảng cáo, tống tiền, lừa đảo, hoặc trực tiếp rút tiền (Banking Trojans).

2.9.6 Cách phòng chống Spyware

+ Kiểm tra kỹ các liên kết trước khi truy cập.

+ Chỉ tải xuống file từ các nguồn uy tín.

+ Cảnh giác với các email từ nguồn lạ.

+ Sử dụng phần mềm bảo mật, chống Spyware chất lượng.

2.10 Adware

2.10.1 Giới thiệu

+ **Adware** là những phần mềm mà quảng cáo được cho phép chạy song song khi chương trình đang chạy.

+ Những đoạn quảng cáo này được tác giả chèn vào bằng những đoạn mã, chúng có thể xuất hiện ở dạng cửa sổ bật lên (pop-up) hoặc các thanh ngay trên màn hình đang chạy của phần mềm.

+ Adware tạo ra nguồn lợi nhuận rất lớn cho nhà phát triển bằng cách tạo tự động những quảng cáo như đã nói ở trên.

2.10.2 Lịch sử

+ **Năm 1995:** các chuyên gia cho rằng Adware thuộc vào loại phần mềm gián điệp vì họ cho rằng chúng đã lấy cắp thông tin người dùng trái phép. Một thời gian sau, tính hợp pháp của phần mềm này tăng lên, từ phần mềm gián điệp trở thành đơn giản là một phần mềm không mong muốn.

+ Với lợi nhuận và tính hợp pháp ngày càng tăng, Adware ngày càng nhiều. **Đến năm 2005 - 2008**, người ta mới lại dõi theo và chú ý đến độ nguy hiểm của phần mềm này.

2.10.3 Nguyên nhân bị nhiễm

+ **Website độc hại:** người dùng không may truy cập vào những trang web đã bị khai thác lỗ hổng, sau đó Adware lợi dụng những điều kiện trên, tải xuống và cài được vào máy tính.

+ **Vô tình tự cài đặt:** những phần mềm miễn phí, crack bản quyền,... có thể chứa các phần mềm độc hại như Adware. Đôi khi do vô tình hay cố ý, bạn đã tự tay cài đặt chúng vào chính thiết bị của mình.

2.10.4 Adware trên các thiết bị di động

+ Các Adware trên thiết bị di động thường được đóng gói với các phần mềm, hầu hết chúng đều miễn phí và hoạt động dựa trên hành vi của người dùng.

+ Mặc dù không độc hại, các phần mềm quảng cáo gây rất nhiều phiền toái với nhiều cửa sổ bật lên, những thông báo liên tục. các yêu cầu cài đặt nhiều phần mềm lạ khác.

2.10.5 Tại sao Adware lại phổ biến?

+ Các phần mềm miễn phí đôi khi lại yêu cầu các bạn "trả phí" bằng chính thông tin của các bạn.

+ Thông thường, nhà phát triển sẽ được trả một lượng doanh thu cho một số lượt nhấp vào quảng cáo nhất định.

2.10.6 Cách thức hoạt động

+ Đóng gói các ứng dụng cùng phần mềm quảng cáo thường được ghi trong phần Điều khoản và Điều kiện trước khi cài đặt. Tuy nhiên, người dùng thường không bao giờ đọc một chuỗi khổng lồ thông tin như thế.

+ Những thông tin người dùng mà Adware thu được chủ yếu được dùng để xây dựng hồ sơ hành vi ảo, từ đó phát quảng những quảng cáo liên quan, điều khiển những hành vi của người dùng.

+ Đồng thời, thông tin người dùng cũng được phân phối đi nhiều tổ chức, đó chính là lý do người dùng cuối bị spam bởi các cá nhân hay tổ chức về những quảng cáo không hề liên quan.

3 Tài liệu tham khảo

- securitybox.vn/ma-doc-la-gi-7-loai-ma-doc-pho-bien
- securitybox.vn/20-cong-cu-kiem-tra-bao-mat-website-hieu-qua-nhat