BÁO CÁO BÀI TẬP

Môn học: Cơ chế hoạt động của mã độc

Tên chủ đề: Advanced Virus Techniques

GVHD: Phan Thế Duy

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT230.O21.ATCL

STT	Họ và tên	MSSV	Email
1	Hồ Ngọc Thiện	21522620	21522620@gm.uit.edu.vn
2	Chu Nguyễn Hoàng Phương	21522483	21522483@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá
1	Tìm hiểu nguyên lý phát hiện sandbox (thí dụ như Cuckoo Sandbox,…). Trình bày ngắn gọn các kỹ thuật nhận diện sandbox để trốn tránh cho mã độc.	100%
2	Hiện thực lại mã độc chống phân tích động khả năng nhận biết môi trường (environmental sensivity) + Nhận biết đang chạy trong môi trường máy ảo + Nhận biết, phát hiện đang bị gỡ lỗi (debugging)	60%
3	Tìm hiểu kỹ thuật Environmental Keying, tìm hiểu cách thức hoạt động của một trong các loại mã độc sau: Equation	100%

[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

BÁO CÁO CHI TIẾT

Viết chương trình lây nhiễm virus vào tập tin thực thi (tập tin thực thi trên Windows – PE file 32 bits) có tính năng đơn giản (mục đích demo giáo dục) như yêu cầu bên dưới.

Về chức năng, mục đích của payload (sử dụng lại phần virus cơ bản của bài tập 01):

- Hiển thị thông điệp ra màn hình thông qua cửa sổ "pop-up" với tiêu đề cửa sổ là "Infection by NT230" và cấu trúc thông điệp là "MSSV01_MSSV02_MSSV03" (thông tin MSSV của các thành viên trong nhóm). Lưu ý: không có dấu "".

- Hoàn trả chức năng gốc ban đầu của chương trình bị lây nhiễm (không phá hủy chức năng của chương trình vật chủ).

Về khả năng trốn tránh việc phát hiện:

a) Tìm hiểu nguyên lý phát hiện sandbox (thí dụ như Cuckoo Sandbox,…). Trình bày ngắn gọn các kỹ thuật nhận diện sandbox để trốn tránh cho mã độc.

Nguyên lý phát hiện sandbox của mã độc là một phần trong lĩnh vực phân tích động (dynamic analysis) của bảo mật thông tin. Mục tiêu của nguyên lý này là xác định xem mã độc đang chạy trong một môi trường sandbox hay không. Một môi trường sandbox là một môi trường cô lập được tạo ra để chạy mã độc mà không gây nguy hại đến hệ thống chủ.

Một số kỹ thuật nhận diện sandbox thường được sử dụng để trốn tránh cho mã độc:

- Kiểm tra môi trường ảo: Mã độc có thể kiểm tra các chỉ số hệ thống hoặc registry để xác định xem môi trường hiện tại có phải là môi trường ảo hay không. Điều này có thể được thực hiện thông qua việc kiểm tra các giá trị đặc biệt, các tệp tin, các tiến trình hoặc thông tin hệ điều hành.
- Giao tiếp với máy chủ kiểm soát: Mã độc có thể thử gửi thông tin về môi trường hiện tại tới một máy chủ kiểm soát bên ngoài. Nếu không có kết nối mạng hoặc không có phản hồi từ máy chủ, mã độc có thể kết luận rằng nó đang chạy trong một môi trường sandbox.
- Kiểm tra tính chất của môi trường: Mã độc có thể kiểm tra các thông số như độ trễ mạng, quyền truy cập hệ điều hành, quyền truy cập vào tệp tin, thông tin phần cứng và phần mềm để phát hiện xem môi trường hiện tại có phải là môi trường sandbox hay không.

- Kiểm tra các tệp tin và tiến trình: Mã độc có thể kiểm tra danh sách các tệp tin và tiến trình hiện đang chạy để phát hiện xem có sự hiện diện của các công cụ phân tích động hoặc các tiến trình đáng ngờ khác.
- Kiểm tra hệ thống: Mã độc có thể xác định một số tính năng của hệ thống thực không có sẵn trong hộp cát hoặc môi trường ảo như **Số lượng lõi CPU và các thành phần phần cứng, Chữ ký hệ thống kỹ thuật số, Các chương trình đã cài đặt....**
- Kiểm tra dựa trên hoạt động của người dùng: Người dùng tương tác với máy tính theo nhiều cách khác nhau, nhưng không có tương tác giống con người trong môi trường sandbox. Do đó, kẻ tấn công có thể dạy phần mềm độc hại chờ một hành động cụ thể của người dùng như **Di chuyển và click chuột, Cuộn một tài liệu, Lưu tập tin vào các thư mục cụ thể** và chỉ thể hiện hành vi nguy hiểm sau đó.

b) Hiện thực lại mã độc chống phân tích động khả năng nhận biết môi trường (environmental sensivity) (trang bị thêm cho payload ban đầu của các bài tập trước).

Các nhóm sinh viên CHỌN 2 KỸ THUẬT TRONG SỐ 3 KỸ THUẬT: anti-debugging, anti-VM và anti-sandbox để hiện thực tính năng:

+ Nhận biết đang chạy trong môi trường máy ảo

+ Nhận biết đang chạy trong môi trường sandbox,

+ Nhận biết, phát hiện đang bị gỡ lỗi (debugging),

Một khi nhận biết đang bị đặt trong môi trường phân tích, nó sẽ không thực hiện hành vi, không thể hiện bản chất của mình (vd: payload không thực thi đoạn mã mục tiêu cho trước, dừng chương trình…).

+ Nhận biết, phát hiện đang bị gỡ lỗi (debugging)

Cách 1: Để phát hiện sự hiện diện của trình gỡ lỗi, phần mềm độc hại có thể đọc một số giá trị hoặc có thể sử dụng API hiện có để phát hiện xem phần mềm độc hại có đang được gỡ lỗi hay không. Một trong những thủ thuật phát hiện trình gỡ lỗi đơn giản bao gồm việc sử dụng hàm winAPI có tên KERNEL32.IsDebuggerPresent.

```
#include <windows.h>
#include <iostream>
using namespace std;
int main(int argc, char* argv[])
{
```

```
if (IsDebuggerPresent()) {
      MessageBox(NULL, L"Debugger detected !!!", L"InfectionbyNT230", MB_OK)
   }
   else {
      MessageBox(NULL, L"21522620-21522483", L"InfectionbyNT230", MB_OK);
   }

   getchar();
   return 0;
}
```

Ta chỉnh sửa lại payload (thêm hàm IsDebuggerPresent()) và chèn vào y như Ex01

Cách 2: Phát hiện trình gỡ lỗi bằng PEB

Khi quy trình được tạo bằng API CreateProcess và nếu cờ tạo được đặt là DEBUG_ONLY_THIS_PROCESS thì một trường đặc biệt sẽ được đặt trong cấu trúc dữ liệu PEB trong bộ nhớ

```
#include <windows.h>
#define WIN32_LEAN_AND_MEAN

   int detectDebugger()
   {
      __asm
      {
         ASSUME FS : NOTHING
         MOV EAX, DWORD PTR FS : [18]
         MOV EAX, DWORD PTR DS : [EAX + 30]
         MOVZX EAX, BYTE PTR DS : [EAX + 2]
         RET
      }
   }
int main(int argc, char* argv[])
{
   if (detectDebugger()) {
```

```
        MessageBox(NULL, L"Debugger detected !!!", L"InfectionbyNT230", MB_OK);
    }
    else {
        MessageBox(NULL, L"21522483-21522483", L"InfectionbyNT230", MB_OK);
    }

}
```

+ Nhận biết đang chạy trong môi trường máy ảo

Các mẫu phần mềm độc hại thường được các nhà phân tích phân tích trong một môi trường biệt lập như Máy ảo . Để ngăn cản việc phân tích các mẫu bên trong phần mềm độc hại trên máy ảo, phần mềm độc hại bao gồm tính năng bảo vệ chống VM hoặc chúng chỉ đơn giản thoát ra khi phần mềm độc hại chạy trong môi trường biệt lập.

Các kỹ thuật sau đây có thể được sử dụng để phát hiện xem mẫu có đang chạy bên trong máy ảo hay không.

1. **Dựa trên thời gian.**

- Phát hiện dựa trên thời gian sử dụng Time Stamp Counter (TSC), đây là một thanh ghi 64-bit được tìm thấy trên tất cả các bộ xử lý x86 từ Pentium trở lên. Nó đếm số chu kỳ từ khi thiết bị được khởi động.
- Khi mã được giả lập, sẽ có sự khác biệt về thời gian đếm giữa máy ảo và máy thật. Vì vậy, phát hiện các sự khác biệt này có thể giúp xác định xem một mẫu phần mềm độc hại có đang chạy trong một môi trường ảo hay không.

```
#include <iostream>

using namespace std;

int main()
{
    unsigned int time1 = 0;
    unsigned int time2 = 0;
    __asm
    {
        RDTSC
```

```
        MOV time1, EAX
        RDTSC
        MOV time2, EAX
    }
    if ((time2 - time1) > 100)
    {
        //cout << "VM Detected" << endl;
        return 0;
    }
    MessageBox(NULL, L"21522620-21522483", L"InfectedbyNT230", MB_OK);
    return 0;
}
```

2. Dựa trên hiện vật (artifacts)

- Malware sử dụng sự hiện diện của cấu hình máy ảo dựa trên các file, network hoặc device artifacts. Malware thường kiểm tra sự hiện diện của những artifacts này để phát hiện sự hiện diện của debugger hoặc môi trường ảo.
- Trong trường hợp tốt nhất sẽ là artifacts, Vmware tạo ra các khóa registry cho Bộ điều khiển đĩa ảo, có thể được tìm thấy trong registry bằng cách sử dụng khóa sau đây. HKLMSYSTEMCurrentControlSetServicesDiskEnum như "SCSIDisk&Ven_VMware_&Prod_VMware_Virtual_S&Rev_1.04&XXX&XXX"

```
int main()
{
    TCHAR lszValue[100];
    HKEY hKey;

    if (RegOpenKeyEx(HKEY_LOCAL_MACHINE, _T("SYSTEM\\CurrentControlSet\\Services\\DiskEnum"), 0, KEY_READ, &hKey) == ERROR_SUCCESS)
    {
        DWORD valueSize = sizeof(lszValue);

        if (RegQueryValueEx(hKey, _T("0"), NULL, NULL, (LPBYTE)lszValue, &valueSize))
        {
            if (_tcsstr(lszValue, _T("VMware")))
```

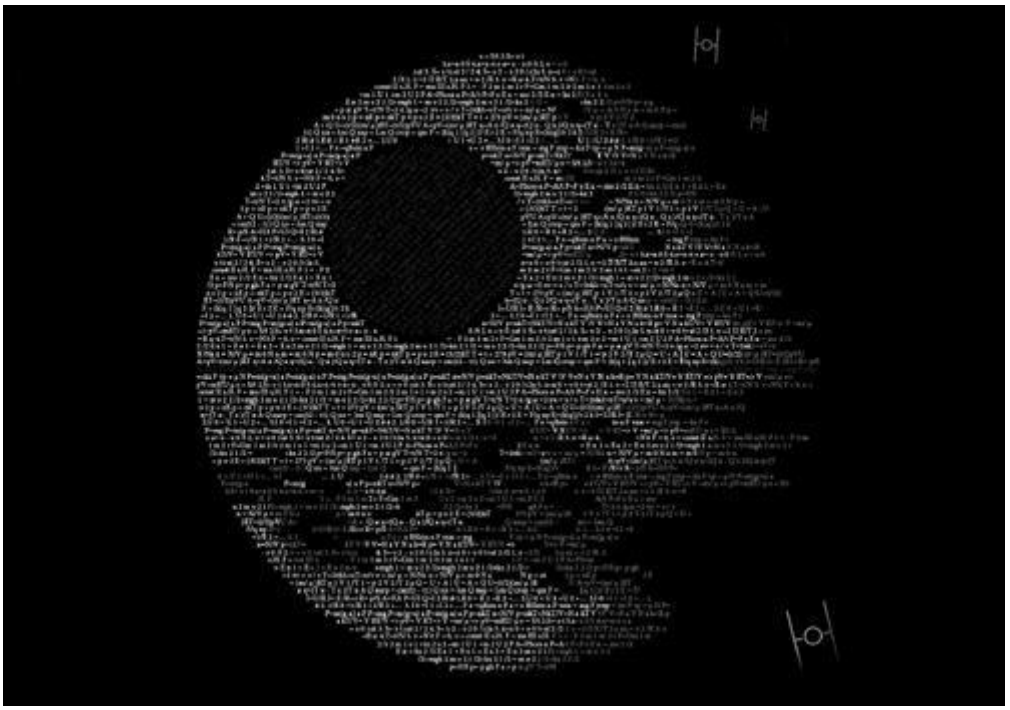
```
            {
                std::cout << "VMware Detected" << std::endl;
            }
        }
        RegCloseKey(hKey);
    }
    MessageBox(NULL, L"21522620-21522483", L"InfectedbyNT230", MB_OK);
    return 0;
}
```

c) Nâng cao: Tìm hiểu kỹ thuật Environmental Keying – một trong những cơ chế trốn tránh phát hiện của mã độc:

+ Nhóm sinh viên tìm hiểu cách thức hoạt động của kỹ thuật này thông qua phân tích các trường hợp báo cáo điển hình được thống kê trên MITRE. Ví dụ: Tìm hiểu cách thức hoạt động của một trong các loại mã độc sau: Equation, InvisiMole, Ninja, PowerPunch, APT41,…

+ Trình bày báo cáo yêu cầu này dưới dạng sơ đồ lây nhiễm (kèm giải thích ngắn gọn qui trình), nhấn mạnh vào vai trò của kỹ thuật Environmental Keying.

Tìm hiểu về Equation



1. Equation group là gì ?

Equation group là một tác nhân đe dọa rất tinh vi đã được tham gia vào nhiều hoạt động CNE (khai thác mạng máy tính) trở lại năm 2001 và có lẽ sớm nhất là năm 1996.Equation group sử dụng nhiều nền tảng phần mềm độc hại, một số trong đó vượt qua mối đe dọa "Regin" nổi tiếng trong sự phức tạp và tinh vi. Equation group có lẽ là một trong những nhóm các nhóm tấn công mạng tinh vi trên thế giới; và họ là những người tiên tiến nhất trong việc đe dọa.

2. Các phương pháp Equation Group đã sử dụng

Nhóm Equation được biết đến với việc sử dụng các thuật toán mã hóa và chiến lược che giấu tinh vi trong các hoạt động của họ. Nhóm này thường sử dụng một triển khai cụ thể của thuật toán mã hóa **RC5** trong phần mềm độc hại của họ. Một số mô-đun gần đây cũng sử dụng **RC6, RC4 và AES**, cùng với các hàm băm và hàm mật mã khác.

Một kỹ thuật đặc biệt thu hút sự chú ý khiến chúng ta nhớ đến phần mềm độc hại phức tạp khác, Gauss. Trình tải GrayFish sử dụng SHA-256 một nghìn lần trên ID đối tượng NTFS duy nhất của thư mục Windows của nạn nhân để giải mã giai đoạn tiếp theo từ sổ đăng ký. Điều này liên kết duy nhất sự lây nhiễm với máy cụ thể và có nghĩa là tải trọng không thể được giải mã nếu không biết ID đối tượng NTFS.

Kỹ thuật này là một ví dụ về Environmental Keying, một kỹ thuật sử dụng các thông tin môi trường cụ thể để tạo khóa mã hóa/giải mã. Trong trường hợp này, ID đối tượng NTFS của thư mục Windows là thông tin môi trường được sử dụng để tạo khóa. Điều này đảm bảo rằng phần mềm độc hại chỉ có thể chạy trên máy cụ thể bị nhiễm và không thể chạy trên các máy khác.

Environmental Keying là một kỹ thuật tinh vi có thể khiến việc phát hiện và phân tích phần mềm độc hại trở nên khó khăn hơn. Bằng cách sử dụng các thông tin môi trường cụ thể để tạo khóa mã hóa, kẻ tấn công có thể nhắm mục tiêu vào các môi trường cụ thể và tránh bị phát hiện bởi các hệ thống phòng thủ chung.

3. Các nền tảng phần mềm độc hại của nhóm Equation

Nhóm Equation đã sử dụng một số nền tảng phần mềm độc hại khác nhau trong các hoạt động của họ. Những nền tảng này được thiết kế để cung cấp cho nhóm khả năng truy cập và kiểm soát các hệ thống bị nhiễm một cách linh hoạt và bí mật. Dưới đây là một số nền tảng phần mềm độc hại đáng chú ý nhất được sử dụng bởi nhóm Equation:

EQUATIONDRUG: Đây là một nền tảng tấn công rất phức tạp được nhóm sử dụng trên các nạn nhân của họ. Nó hỗ trợ hệ thống plugin mô-đun, có thể được tải lên và dỡ xuống một cách linh hoạt bởi kẻ tấn công.

DOUBLEFANTASY: Đây là một Trojan kiểu trình xác thực, được thiết kế để xác nhận mục tiêu là mục tiêu dự định. Nếu mục tiêu được xác nhận, chúng sẽ được nâng cấp lên một nền tảng tinh vi hơn như EQUATIONDRUG hoặc GRAYFISH.

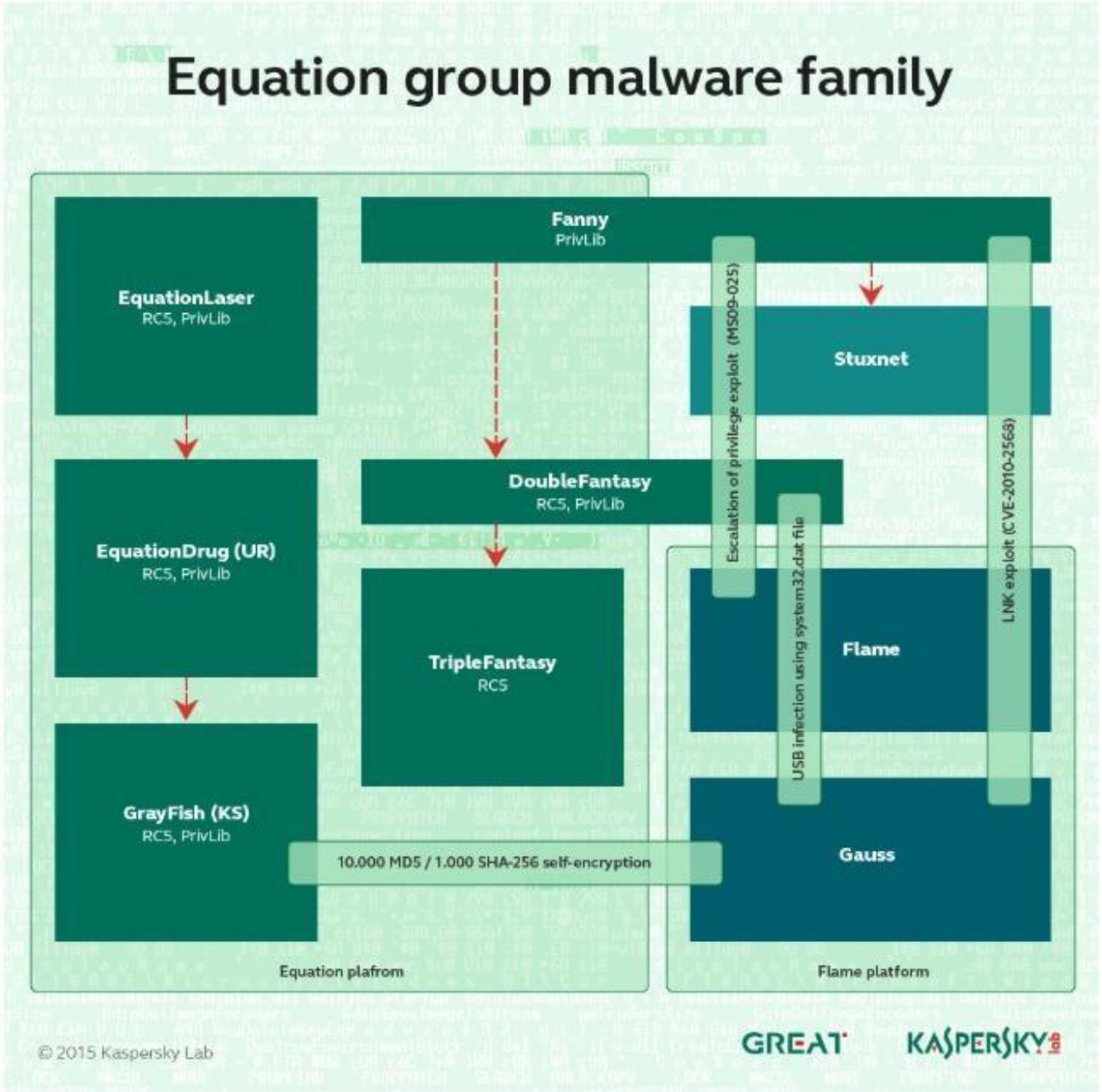
EQUESTRE: Tương tự như EQUATIONDRUG.

TRIPLEFANTASY: Backdoor đầy đủ tính năng đôi khi được sử dụng cùng với GRAYFISH. Có vẻ như là bản nâng cấp của DOUBLEFANTASY và có thể là plugin kiểu trình xác thực gần đây hơn.

GRAYFISH: Nền tảng tấn công tinh vi nhất từ nhóm Equation. Nó hoàn toàn nằm trong sổ đăng ký, dựa vào bootkit để thực thi khi khởi động hệ điều hành.

FANNY: Một con sâu máy tính được tạo vào năm 2008 và được sử dụng để thu thập thông tin về các mục tiêu ở Trung Đông và Châu Á. Một số nạn nhân dường như đã được nâng cấp trước tiên lên DoubleFantasy, sau đó là hệ thống EQUATIONDRUG. Fanny sử dụng các khai thác cho hai lỗ hổng zero-day sau này được phát hiện với Stuxnet.

EQUATIONLASER: Một implant ban đầu từ nhóm EQUATION, được sử dụng vào khoảng năm 2001-2004. Tương thích với Windows 95/98 và được tạo ra vào khoảng thời gian giữa DOUBLEFANTASY và EQUATIONDRUG



4. DOUBLEFANTASY

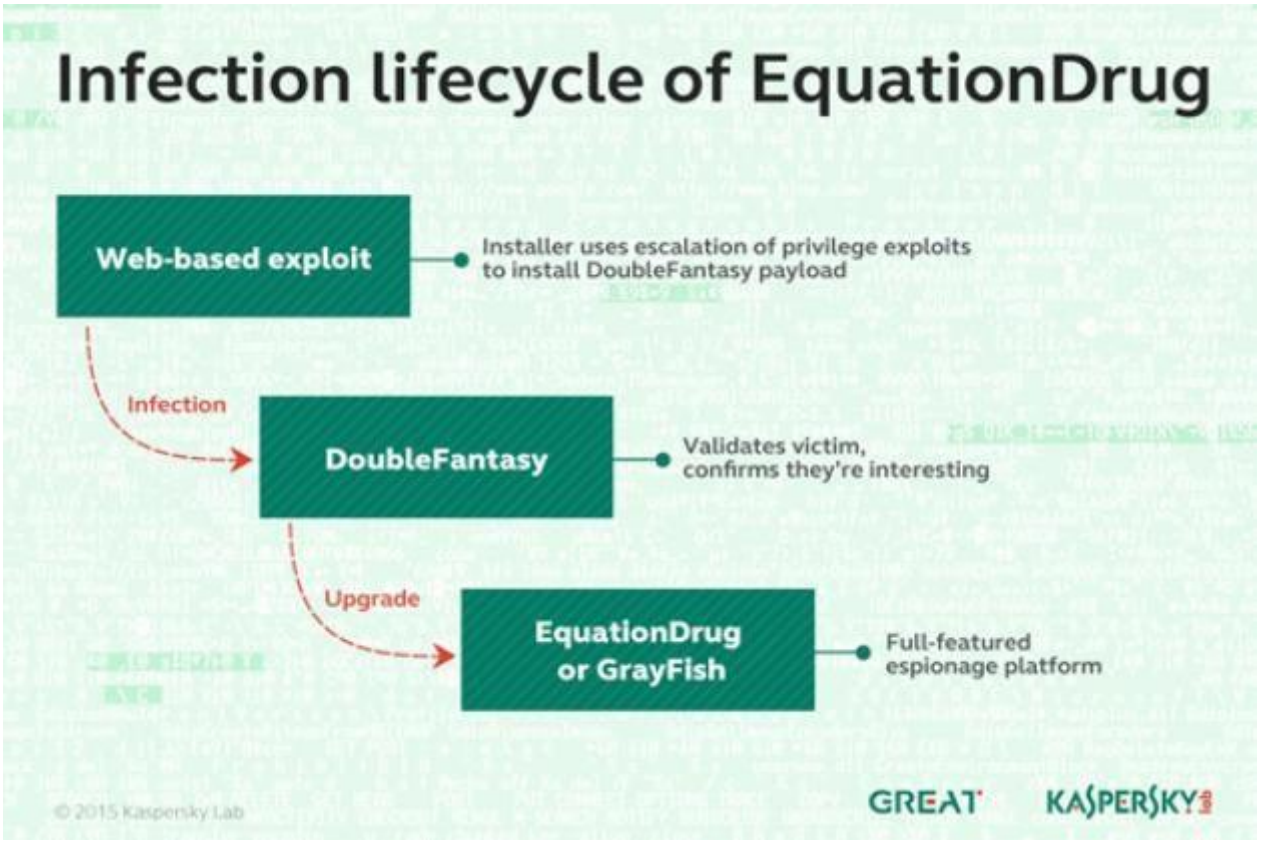
Phần mềm DoubleFantasy do nhóm Equation sử dụng có hai mục đích:

- Xác nhận xem nạn nhân có đáng quan tâm hay không: Nếu nạn nhân được cho là đáng quan tâm, họ sẽ được nâng cấp lên nền tảng EquationDrug hoặc GrayFish.
- Duy trì một cửa hậu vào máy tính của mục tiêu tiềm năng: Điều này cho phép nhóm Equation duy trì quyền truy cập vào máy tính của nạn nhân ngay cả khi họ không được cho là đáng quan tâm ngay lập tức.

Phần mềm DoubleFantasy lưu trữ các thông tin sau trong khối cấu hình của nó:

- Số phiên bản nội bộ: Điều này giúp xác định phiên bản cụ thể của phần mềm đang được sử dụng.
- Máy chủ hợp pháp được sử dụng để xác thực kết nối internet: Chúng được sử dụng để đảm bảo rằng phần mềm chỉ liên lạc với các máy chủ hợp pháp.
- Máy chủ chỉ huy và điều khiển (C&C): Đây là các máy chủ mà phần mềm liên lạc để nhận hướng dẫn và gửi dữ liệu.

5. EQUATIONDRUG



Tổng quan:

- EQUATIONDRUG là một trong những nền tảng gián điệp phức tạp nhất của nhóm Equation.
- Nền tảng này được phát triển từ năm 2003 đến năm 2013 và sau đó được thay thế bởi GrayFish.
- EQUATIONDRUG được tạo ra như một bản nâng cấp từ nền tảng EQUATIONLASER.

Quá trình lây nhiễm:

- Nạn nhân không bị nhiễm EQUATIONDRUG ngay lập tức.
- Đầu tiên, kẻ tấn công sẽ lây nhiễm cho họ bằng DOUBLEFANTASY, một plugin xác thực.
- Nếu nạn nhân được xác nhận là mục tiêu tiềm năng, trình cài đặt EQUATIONDRUG sẽ được phân phối.

Chức năng:

- Theo mặc định, một bộ các mô-đun cốt lõi được cài đặt vào máy tính của mục tiêu cùng với EQUATIONDRUG, cho phép kẻ tấn công kiểm soát hoàn toàn hệ điều hành.
- Trong trường hợp các tính năng cơ bản của phần mềm độc hại không đủ, EquationDrug hỗ trợ thêm các plugin mới để mở rộng chức năng của nó.
- Đã tìm thấy 35 plugin khác nhau cho EquationDrug và 18 trình điều khiển.

Hạn chế:

- Các mô-đun cốt lõi của EquationDrug, được thiết kế để móc sâu vào hệ điều hành, không chứa chữ ký kỹ thuật số đáng tin cậy và không thể chạy trực tiếp trên các hệ điều hành hiện đại.
- Mã kiểm tra xem phiên bản hệ điều hành có phải trước Windows XP/2003 hay không.
- Một số plugin ban đầu được thiết kế để sử dụng trên Windows 95/98/ME.
- Nếu mục tiêu sử dụng hệ điều hành hiện đại như Windows 7, kẻ tấn công sẽ sử dụng các nền tảng TripleFantasy hoặc GrayFish.

Chức năng tự hủy:

- EquationDrug có bộ đếm thời gian tích hợp, có thể được thiết kế để tự hủy nếu không nhận được lệnh từ C&C trong một khoảng thời gian (vài tháng).

Lưu trữ dữ liệu:

- Thông tin bị đánh cắp từ PC và được chuẩn bị để truyền đến C&C được lưu trữ dưới dạng mã hóa trong một số tệp phông chữ giả (*.FON) bên trong thư mục Windows\Fonts trên máy tính của nạn nhân.

6. GRAYFISH

Tổng quan:

- GRAYFISH là nền tảng phần mềm độc hại tinh vi nhất của nhóm Equation.
- Nó được thiết kế để cung cấp cơ chế duy trì hiệu quả (gần như "vô hình"), lưu trữ ẩn và thực thi lệnh độc hại bên trong hệ điều hành Windows.
- GRAYFISH được phát triển từ năm 2008 đến năm 2013 và tương thích với tất cả các phiên bản hiện đại của hệ điều hành Microsoft, bao gồm Windows NT 4.0, Windows 2000, Windows XP, Windows Vista, Windows 7 và 8 - cả phiên bản 32 bit và 64 bit.

Bootkit tinh vi:

- GRAYFISH bao gồm một bootkit cực kỳ tinh vi, phức tạp hơn bất kỳ bootkit nào chúng tôi từng thấy trước đây. Điều này cho thấy những nhà phát triển có trình độ cao nhất đứng sau việc tạo ra nó.
- Khi máy tính khởi động, GRAYFISH chiếm quyền cơ chế tải hệ điều hành bằng cách tiêm mã của nó vào bản ghi khởi động. Điều này cho phép nó kiểm soát việc khởi chạy Windows ở từng giai đoạn. Trên thực tế, sau khi bị nhiễm, máy tính không tự chạy nữa: chính GRAYFISH chạy nó từng bước, thực hiện các thay đổi cần thiết trên đường đi.

Cơ chế giải mã nhiều giai đoạn:

- Sau khi GRAYFISH khởi động Windows, nó khởi chạy cơ chế giải mã nhiều giai đoạn (bốn đến năm giai đoạn) để thực thi mã trong môi trường Windows.
- Sau đó, nó chạy các mô-đun chức năng được lưu trữ bên trong sổ đăng ký Windows.
- Mỗi giai đoạn giải mã và thực thi giai đoạn tiếp theo, và toàn bộ nền tảng sẽ chỉ khởi động sau khi thực thi thành công tất cả các cấp độ. Nếu xảy ra lỗi trong quá trình khởi chạy, toàn bộ nền tảng GRAYFISH sẽ tự hủy.

Hệ thống tệp ảo được mã hóa:

- Để lưu trữ thông tin bị đánh cắp cũng như thông tin phụ trợ của riêng nó, GRAYFISH triển khai Hệ thống tệp ảo (VFS) được mã hóa của riêng nó bên trong sổ đăng ký Windows.

Khai thác trình điều khiển hợp pháp:

- Để vượt qua cơ chế bảo mật hệ điều hành hiện đại chặn việc thực thi mã không tin cậy ở chế độ hạt nhân, GRAYFISH khai thác một số trình điều khiển hợp pháp, bao gồm một trình điều khiển từ chương trình CloneCD. Trình điều khiển này (ElbyCDIO.sys) chứa một lỗ hổng mà GRAYFISH khai thác để đạt được việc thực

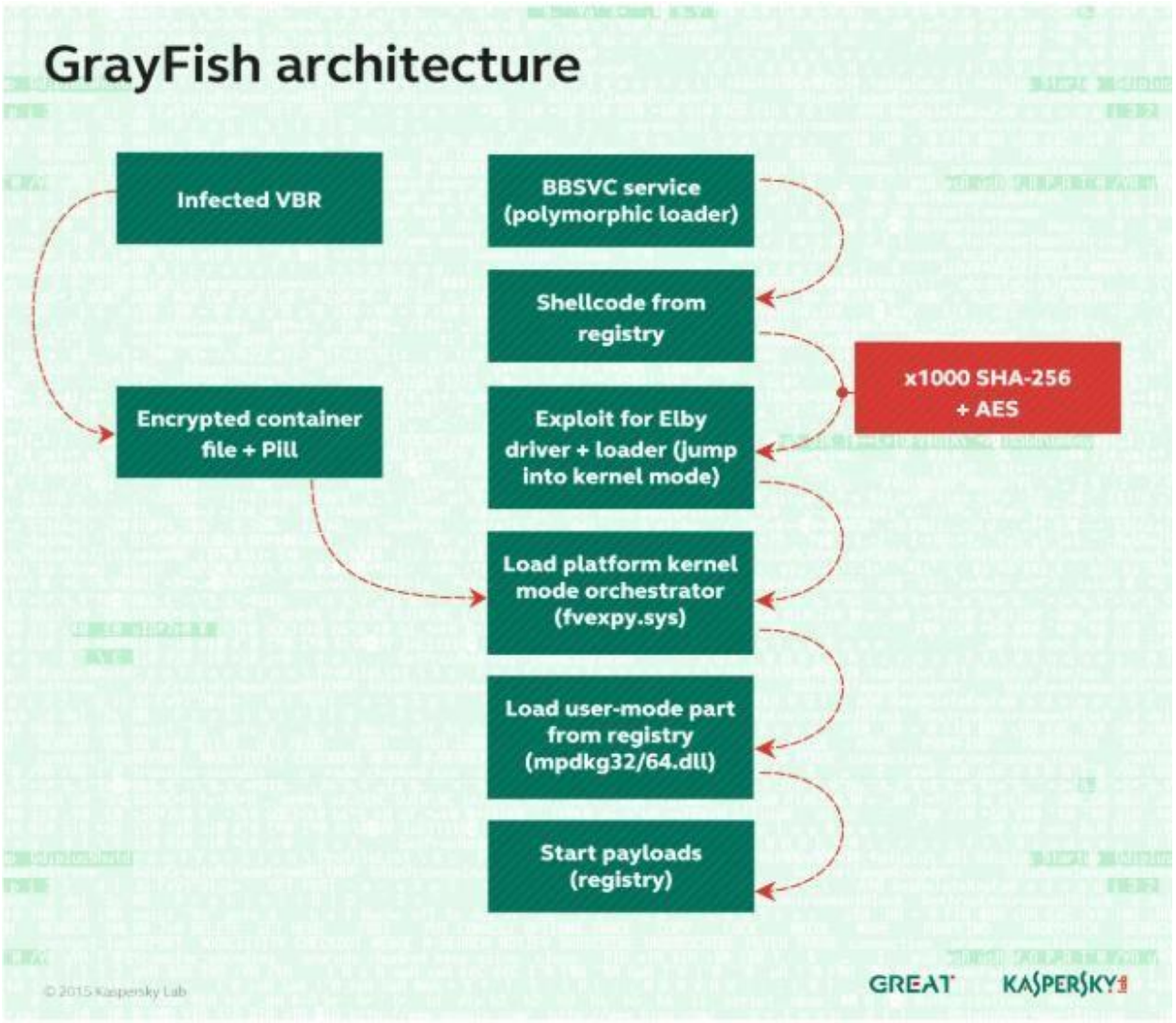
thi mã ở cấp hạt nhân. Mặc dù lỗ hổng này đã được phát hiện vào năm 2009, chữ ký kỹ thuật số vẫn chưa bị thu hồi.

Vô hình đối với phần mềm chống vi-rút:

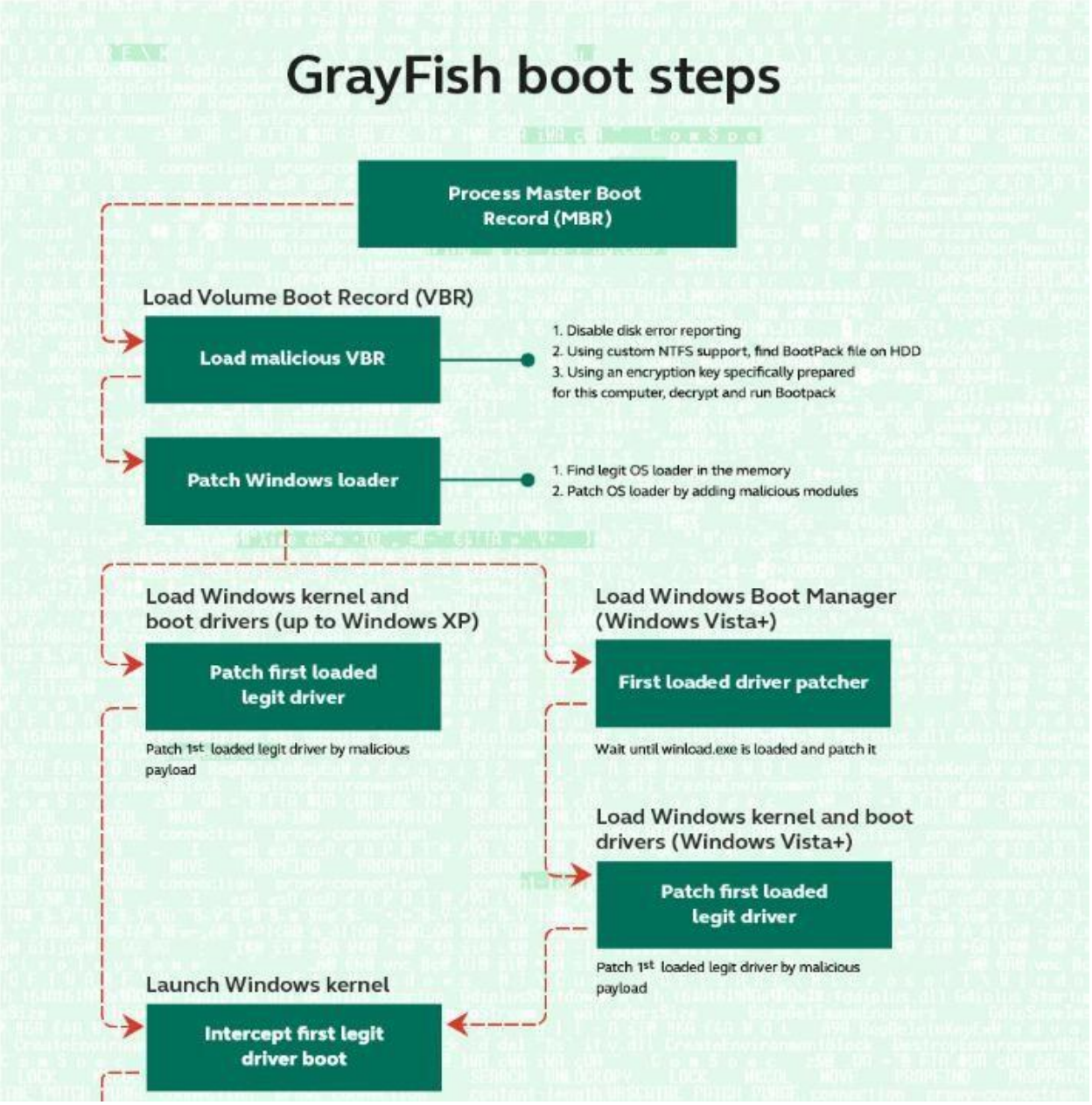
- Việc triển khai GRAYFISH dường như được thiết kế để khiến nó vô hình với các sản phẩm chống vi-rút. Khi được sử dụng cùng với bootkit, tất cả các mô-đun cũng như dữ liệu bị đánh cắp được lưu trữ dưới dạng mã hóa trong sổ đăng ký và được giải mã và thực thi động. Hoàn toàn không có mô-đun thực thi độc hại nào trên hệ thống tệp của hệ thống bị nhiễm.

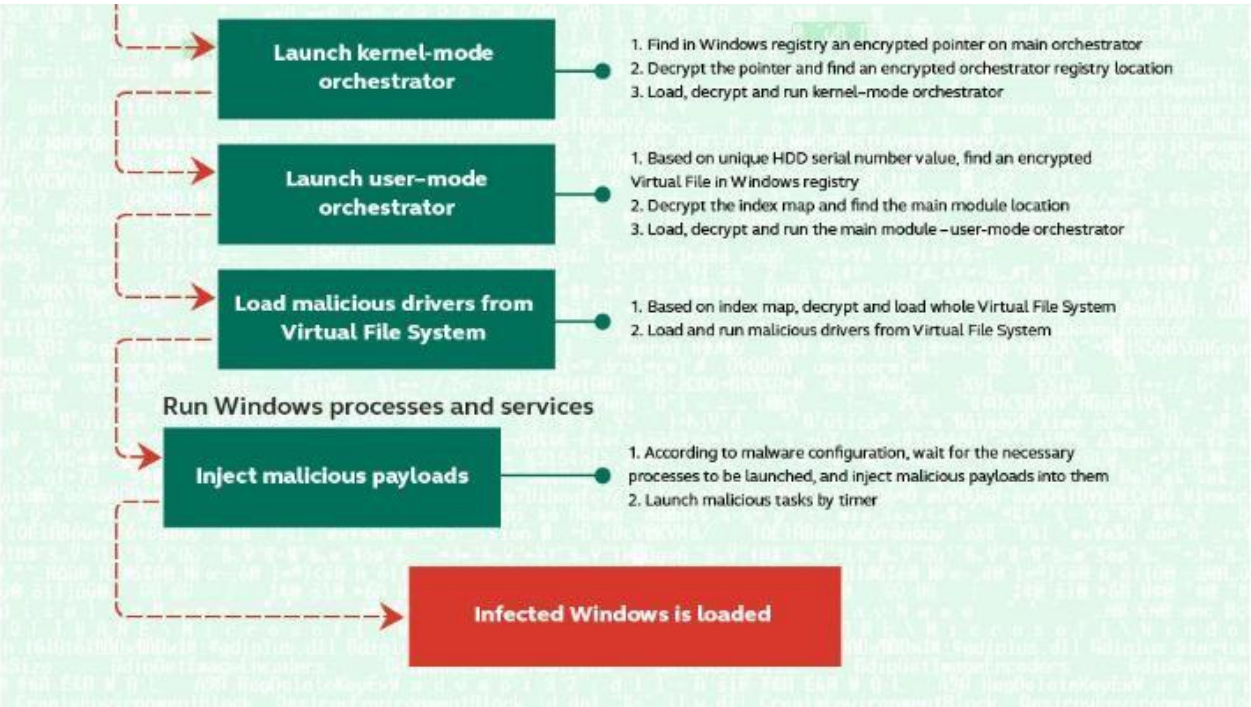
Hành vi đặc biệt:

- Bộ tải GRAYFISH giai đoạn đầu tiên tính toán băm SHA-256 của NTFS của thư mục hệ thống (%Windows% hoặc %System%) Object_ID một nghìn lần. Kết quả được sử dụng làm khóa giải mã AES cho giai đoạn tiếp theo. Điều này có phần tương tự như Gauss, tính toán băm MD5 trên tên thư mục mục tiêu của nó 10.000 lần và sử dụng kết quả làm khóa giải mã.



Cấu trúc GrayFish


GrayFish boot steps
Process Master Boot Record (MBR)
Load Volume Boot Record (VBR)
Load malicious VBR
1. Disable disk error reporting
2. Using custom NTFS support, find BootPack file on HDD
3. Using an encryption key specifically prepared for this computer, decrypt and run Bootpack
Patch Windows loader
1. Find legit OS loader in the memory
2. Patch OS loader by adding malicious modules
Load Windows kernel and boot drivers (up to Windows XP)
Patch first loaded legit driver
Patch 1st loaded legit driver by malicious payload
Load Windows Boot Manager (Windows Vista+)
First loaded driver patcher
Wait until winload.exe is loaded and patch it
Load Windows kernel and boot drivers (Windows Vista+)
Patch first loaded legit driver
Patch 1st loaded legit driver by malicious payload
Launch Windows kernel
Intercept first legit driver boot



Cơ chế bootloader của GrayFish

7. Fanny

Tổng quan:

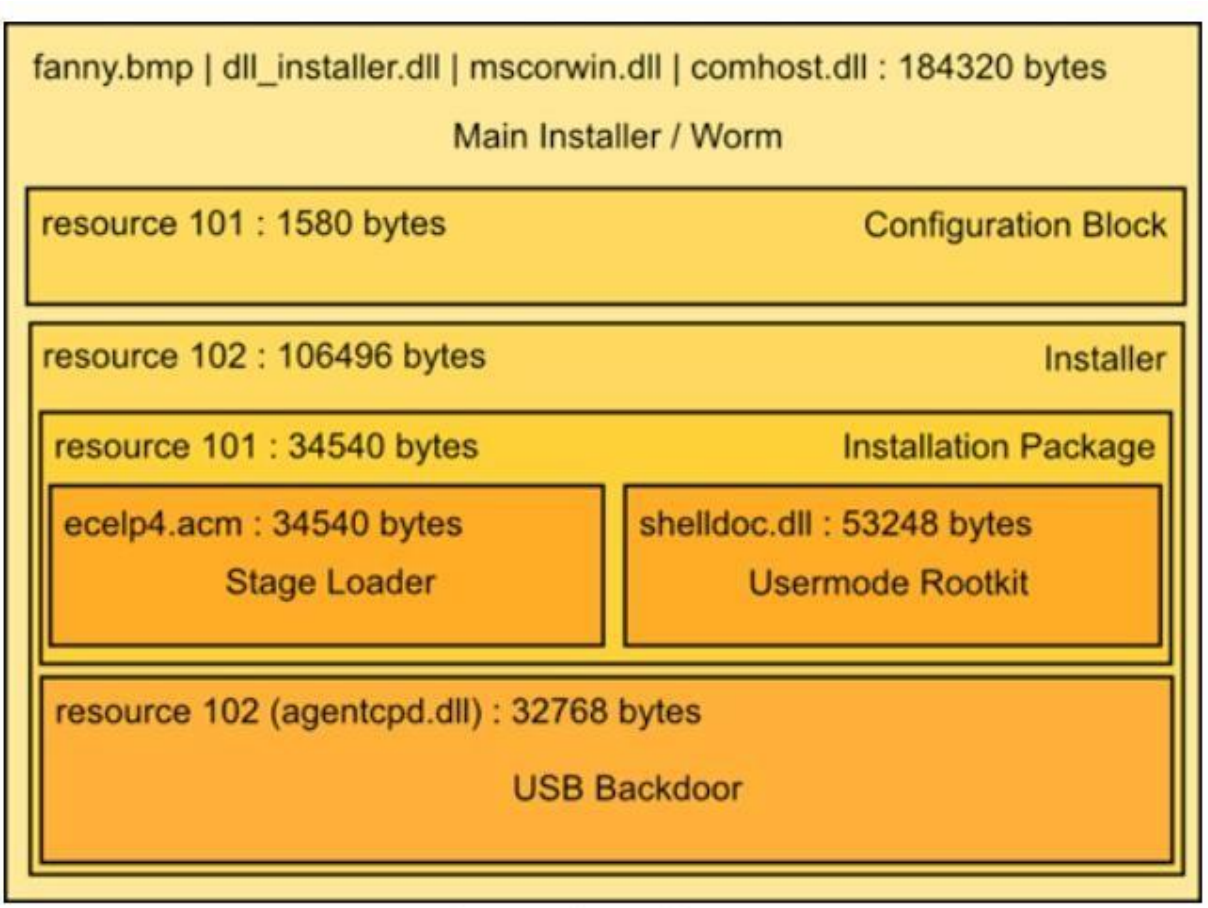
- Fanny là một con sâu máy tính được nhóm Equation tạo ra vào năm 2008 và được phân phối khắp Trung Đông và Châu Á.
- Fanny sử dụng hai lỗ hổng zero-day, sau này được phát hiện trong quá trình khám phá Stuxnet.
- Để lây lan, nó sử dụng lỗ hổng Stuxnet LNK và USB.
- Fanny sử dụng lỗ hổng được vá bởi bản tin MS09-025 của Microsoft, lỗ hổng này từ năm 2009 cũng được sử dụng trong một trong những phiên bản đầu tiên của Stuxnet.

Mối liên hệ với Stuxnet:

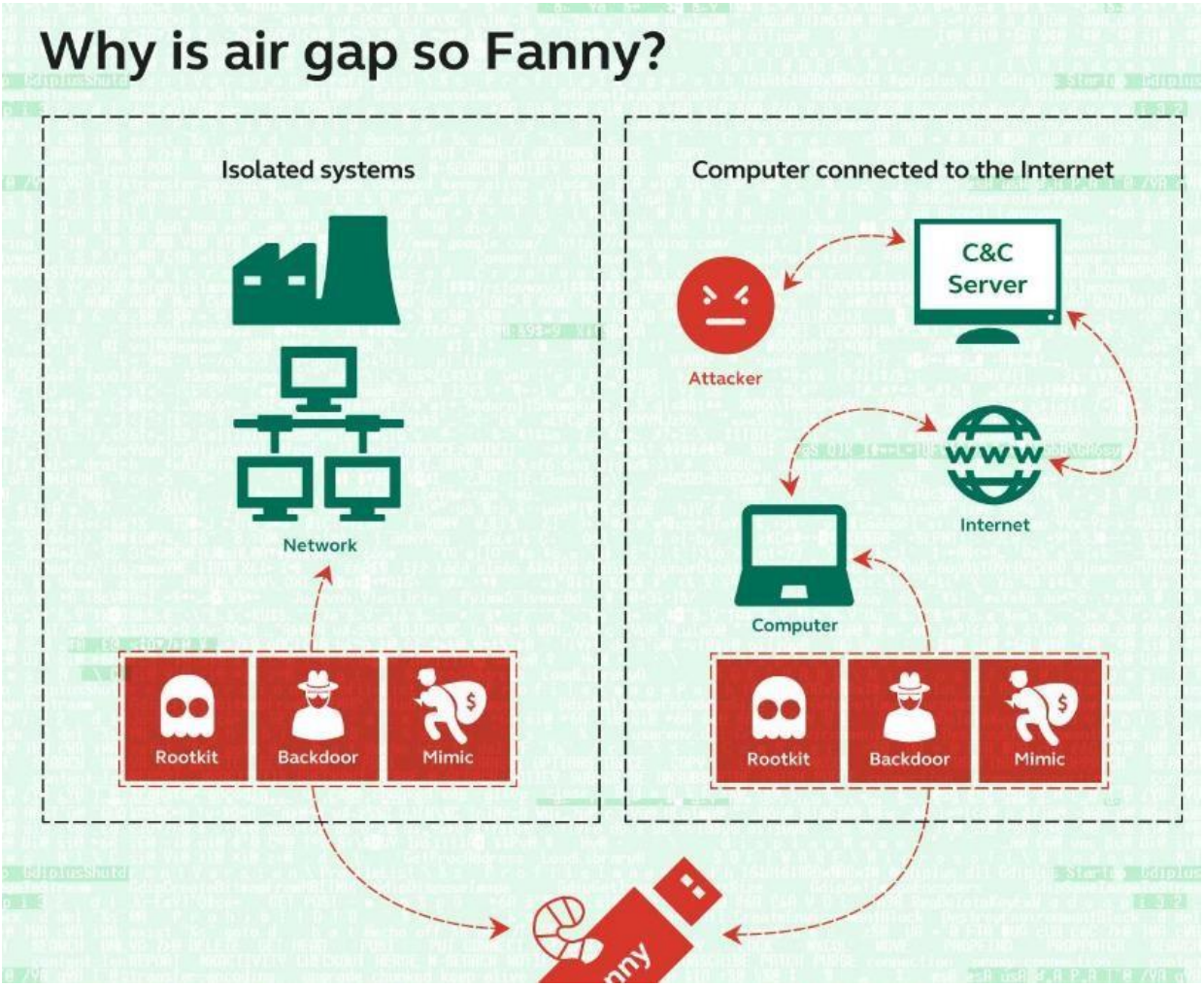
- Điều quan trọng cần lưu ý là hai lỗ hổng này đã được sử dụng trong Fanny trước khi chúng được tích hợp vào Stuxnet, điều này cho thấy nhóm EQUATION đã truy cập vào các zero-day này trước nhóm Stuxnet.
- Trên thực tế, việc sử dụng tương tự cả hai lỗ hổng cùng nhau trong các con sâu máy tính khác nhau, vào cùng một thời điểm, cho thấy nhóm EQUATION và các nhà phát triển Stuxnet có thể là cùng một nhóm hoặc hợp tác chặt chẽ với nhau.

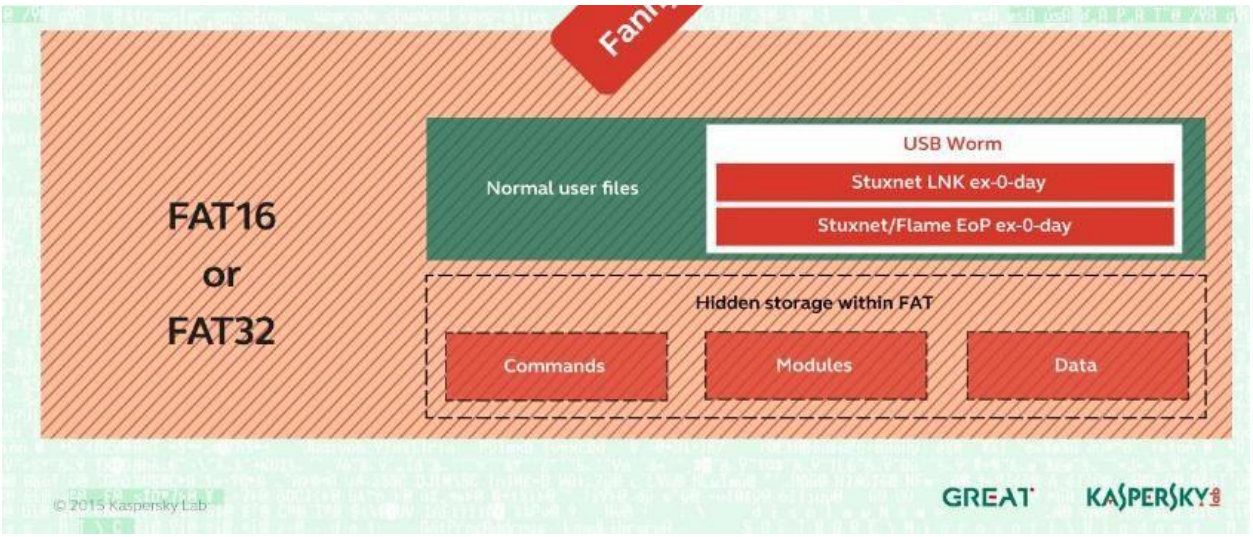
Mục đích chính:

- Mục đích chính của Fanny dường như là lập bản đồ các mạng bị air gap.
- Đối với điều này, nó sử dụng cơ chế chỉ huy và kiểm soát dựa trên USB độc đáo.
- Khi một ổ USB bị nhiễm, Fanny tạo một khu vực lưu trữ ẩn trên ổ đĩa.
- Nếu nó lây nhiễm vào máy tính không có kết nối internet, nó sẽ thu thập thông tin hệ thống cơ bản và lưu nó vào khu vực ẩn của ổ đĩa.
- Sau đó, khi một ổ đĩa chứa thông tin ẩn được cắm vào máy tính có kết nối internet bị Fanny lây nhiễm, dữ liệu sẽ được thu thập từ khu vực ẩn và gửi đến C&C.
- Nếu kẻ tấn công muốn chạy lệnh trên các mạng bị air gap, họ có thể lưu các lệnh này vào khu vực ẩn của ổ USB.
- Khi ổ đĩa được cắm vào máy tính bị air gap, Fanny sẽ nhận dạng các lệnh và thực thi chúng.
- Điều này cho phép nhóm Equation chạy lệnh bên trong các mạng bị air gap thông qua việc sử dụng ổ USB bị nhiễm và cũng lập bản đồ cơ sở hạ tầng của các mạng đó.



Fanny malware diagram


Why is air gap so Fanny?
Isolated systems
Network
Rootkit
Backdoor
Mimic
Computer connected to the Internet
C&C Server
Attacker
www
Internet
Computer
Rootkit
Backdoor
Mimic



Module Fanny

MD5	0a209ac0de4ac033f31d6ba9191a8f7a
Size	184320
Type	Win32 DLL
Internal name	dll_installer.dll
Compiled	2008.07.28 08:11:35 (GMT)

8. Các lỗ hổng được sử dụng bởi nhóm Equation

- Lỗ hổng Windows Kernel EoP được sử dụng trong Stuxnet 2009 (atempsvc.ocx), đã được vá bởi MS09-025. (CVE chưa biết).
- Lỗ hổng TTF đã được vá bởi MS12-034 (có thể là CVE-2012-0159).
- Lỗ hổng TTF đã được vá bởi MS13-081 (có thể là CVE-2013-3894).
- Lỗ hổng LNK như được sử dụng bởi Stuxnet. (CVE-2010-2568).
- CVE-2013-3918 (Internet Explorer).
- CVE-2012-1723 (Java).
- CVE-2012-4681 (Java).

Tổng quan:

Ít nhất bốn trong số các lỗ hổng này đã được nhóm EQUATION sử dụng như zero-day.

Ngoài ra, chúng ta quan sát thấy việc sử dụng các lỗ hổng chưa biết, có thể là zero-day, chống lại Firefox 17, như được sử dụng trong Trình duyệt TOR.

Một trường hợp thú vị là việc sử dụng CVE-2013-3918, ban đầu được sử dụng bởi nhóm APT đứng sau cuộc tấn công Aurora năm 2009. Nhóm EQUATION đã chiếm được khai thác của họ và sử dụng lại nó để nhắm mục tiêu vào người dùng chính phủ ở Afghanistan.

9. Các kỹ thuật lây nhiễm được sử dụng bởi nhóm Equation

Mã tự sao chép (sâu) - Fanny: Fanny là một con sâu máy tính được thiết kế để tự động lây lan qua mạng.

Phương tiện vật lý, CD-ROM: Nhóm Equation đã sử dụng CD-ROM bị nhiễm để lây nhiễm cho các nạn nhân. Điều này liên quan đến kỹ thuật "chặn", nơi những kẻ tấn công chặn hàng hóa được vận chuyển và thay thế chúng bằng các phiên bản bị Trojan hóa.

USB + khai thác: Nhóm Equation cũng đã sử dụng ổ USB bị nhiễm kết hợp với các khai thác để lây nhiễm cho các nạn nhân. Điều này cho phép họ khai thác các lỗ hổng trong hệ điều hành của nạn nhân để cài đặt phần mềm độc hại của họ.

Khai thác dựa trên web: Nhóm Equation cũng đã sử dụng các khai thác dựa trên web để lây nhiễm cho các nạn nhân. Điều này liên quan đến việc nhắm mục tiêu các lỗ hổng trong các trang web hoặc trình duyệt web để cài đặt phần mềm độc hại của họ trên máy tính của nạn nhân.

Tấn công CD-ROM:

Các cuộc tấn công sử dụng phương tiện vật lý (CD-ROM) đặc biệt thú vị vì chúng cho thấy việc sử dụng kỹ thuật "chặn", nơi những kẻ tấn công chặn hàng hóa được vận chuyển và thay thế chúng bằng các phiên bản bị Trojan hóa.

Một sự cố như vậy liên quan đến việc nhắm mục tiêu vào những người tham gia một hội nghị khoa học ở Houston. Khi trở về nhà, một số người tham gia đã nhận được qua thư một bản sao của các bài báo hội nghị, cùng với một bản trình chiếu bao gồm nhiều tài liệu hội nghị.

CD-ROM bị xâm nhập đã sử dụng "autorun.inf" để thực thi trình cài đặt bắt đầu bằng cách cố gắng leo thang đặc quyền bằng hai khai thác nhóm EQUATION đã biết.

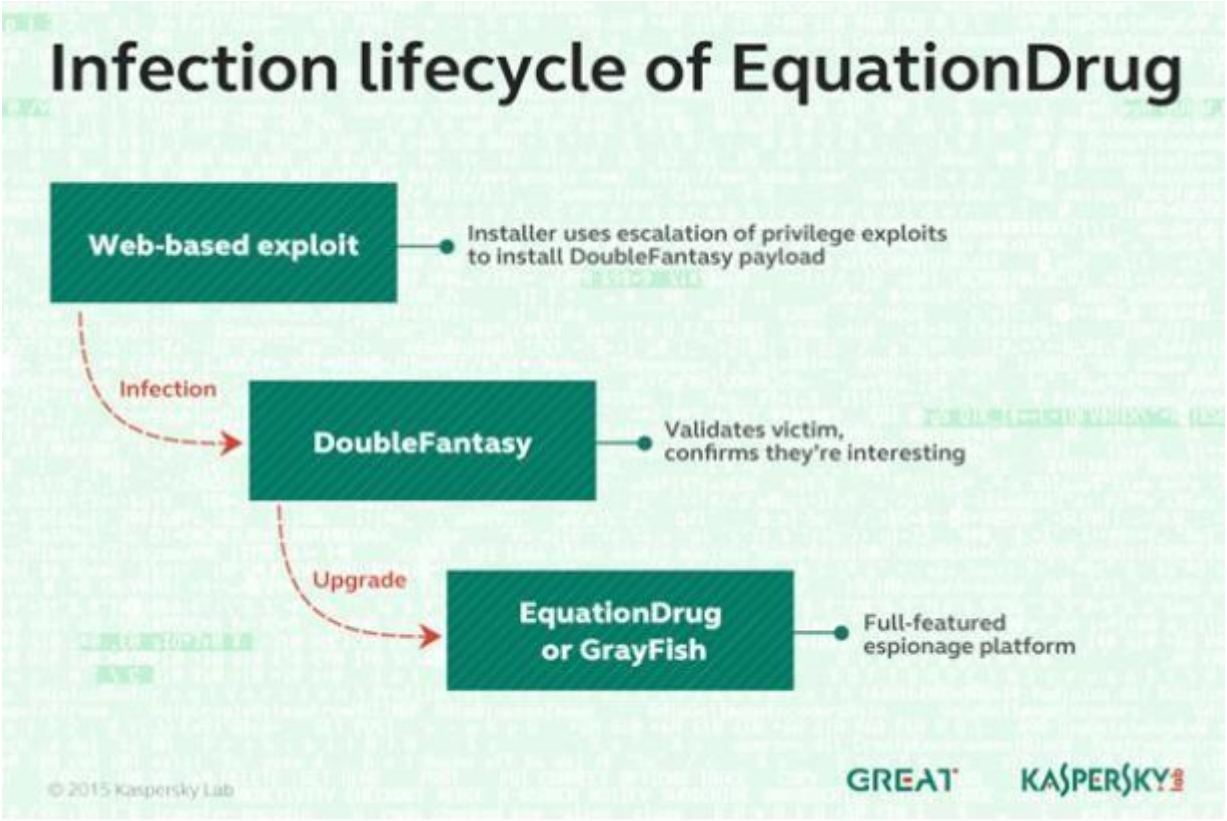
Tiếp theo, nó đã cố gắng chạy implant DOUBLEFANTASY của nhóm và cài đặt nó vào máy của nạn nhân.

Cách thức chính xác mà các CD này bị chặn là không rõ. Chúng tôi không tin rằng ban tổ chức hội nghị đã cố tình làm điều này. Đồng thời, phần mềm độc hại

DOUBLEFANTASY cực kỳ hiếm, cùng với trình cài đặt của nó với hai khai thác zero-day, không vô tình xuất hiện trên CD.

Ví dụ khác:

Một ví dụ khác là CD cài đặt Oracle bị Trojan hóa có chứa trình tải Trojan EQUATIONLASER cùng với trình cài đặt Oracle.



10. Khả năng lây nhiễm phần mềm độc hại của nhóm Equation vào phần mềm ổ cứng

Mặc dù việc triển khai hệ thống phần mềm độc hại của họ cực kỳ phức tạp, thậm chí vượt qua cả Regin về độ tinh vi, nhưng có một khía cạnh trong công nghệ tấn công của nhóm EQUATION vượt xa bất kỳ điều gì chúng ta từng thấy trước đây.

Đây là khả năng lây nhiễm phần mềm độc hại vào phần mềm ổ cứng.

Mô-đun lập trình lại phần mềm ổ cứng EQUATIONDRUG có phiên bản 3.0.1 trong khi mô-đun lập trình lại GRAYFISH có phiên bản 4.2.0.

Plugin lập trình lại phần mềm ổ cứng của nhóm EQUATION có ID nội bộ 80AA, là một số duy nhất trong bảng ID plugin của nhóm.

Điều này cho phép các plugin khác xác định và sử dụng nó khi cần.

Cả hai phiên bản 32 bit và 64 bit của plugin đều được tìm thấy.

Plugin hỗ trợ hai chức năng chính: lập trình lại phần mềm ổ cứng bằng tải trọng tùy chỉnh từ nhóm EQUATION và cung cấp API vào một tập hợp các khu vực ẩn (hoặc lưu trữ dữ liệu) của ổ cứng.

Điều này đạt được một số điều quan trọng:

- Sự kiên trì cực độ tồn tại sau khi định dạng đĩa và cài đặt lại hệ điều hành.
- Một kho lưu trữ ẩn, liên tục được ẩn bên trong ổ cứng.

Phiên bản plugin 3 có khả năng lập trình lại sáu "loại" ổ đĩa:

- “Maxtor”, “Maxtor STM”
- “ST”, “Maxtor STM”, <Seagate Technology>
- “WDC WD”, <Western Digital Technologies, Inc>
- “SAMSUNG”, <SAMSUNG ELECTRONICS CO. LTD>
- “WDC WD”, <Western Digital Technologies, Inc> các kiểm tra cụ thể của nhà cung cấp bổ sung được sử dụng (sinh ra hai lớp con)
- <Seagate Technology>

Phiên bản plugin 4 phức tạp hơn và có thể lập trình lại 12 "loại" ổ đĩa.

- “WDC WD”, <Western Digital Technologies Inc> các kiểm tra cụ thể của nhà cung cấp bổ sung được sử dụng
- “ST”, “Maxtor STM”, “SEAGATE ST”, <Seagate Technology>
- “SAMSUNG”, <SAMSUNG ELECTRONICS CO., LTD.>
- “WDC WD”, <Western Digital Technologies, Inc.> các kiểm tra cụ thể của nhà cung cấp bổ sung được sử dụng
- <HGST a Western Digital Company>, “IC”, “IBM”, “Hitachi”, “HTS”, “HTE”, “HDS”, “HDT”, “ExcelStor”
- “Max”, “Maxtor STM”
- <MICRON TECHNOLOGY, INC.>, “C300”, “M4”
- <HGST a Western Digital Company>, <TOSHIBA CORPORATION>
- “OCZ”, “OWC”, “Corsair”, “Mushkin” các kiểm tra cụ thể của nhà cung cấp bổ sung được sử dụng
- <Samsung Electronics Co., Ltd., Storage System Division>, <Seagate Technology>, <SAMSUNG ELECTRONICS CO., LTD.>

- <TOSHIBA CORPORATION COMPUTER DIVISION>, “TOSHIBA M”
- <Seagate Technology>, “ST”

11. Những rò rỉ nhóm Equation:

Một số từ khóa bị lãng quên trong các mô-đun mà chúng ta phân tích bao gồm:

- SKYHOOKCHOW
- prkMtx - mutex duy nhất được sử dụng bởi thư viện khai thác của nhóm Equation (“PrivLib”)
- “SF” - như trong “SFInstall”, “SFConfig”
- “UR”, “URInstall” - “Thực hiện cài đặt sau UR…”
- “implant” - từ “Hết thời gian chờ đợi sự kiện“canInstallNow”từ EXE cụ thể của implant!”
- STEALTHFIGHTER - (VTT/82055898/STEALTHFIGHTER/2008-10-16/14:59:06.229-04:00)
- DRINKPARSLEY - (Hướng dẫn sử dụng/DRINKPARSLEY/2008-09-30/10:06:46.468-04:00)
- STRAITACID - (VTT/82053737/STRAITACID/2008-09-03/10:44:56.361-04:00)
- LUTEUSOBSTOS - (VTT/82051410/LUTEUSOBSTOS/2008-07-30/17:27:23.715-04:00)
- STRAITSHOOTER STRAITSHOOTER30.exe
- DESERTWINTER - c:\desert3\objfre_w2K_x86\i386\DesertWinterDriver.pdb
- GROK - standalonegrok_2.1.1.1
- “RMGREE5” - c:\users\rmgree5...

Các chuỗi VTT ở trên dường như chứa dấu thời gian và bộ đếm lây nhiễm. Điều này có thể cho thấy nhóm EQUATION tấn công khoảng 2000 người dùng mỗi tháng, mặc dù có thể một số nạn nhân “không thú vị” có thể được loại bỏ.

12. Phân tích cách sử dụng thuật toán mã hóa RC5/6 của nhóm Equation:

Nhóm Equation sử dụng thuật toán mã hóa RC5 và RC6 khá rộng rãi trong các công cụ của họ.

Họ cũng sử dụng XOR đơn giản, bảng thay thế, RC4 và AES.

RC5 và RC6 là hai thuật toán mã hóa được thiết kế bởi Ronald Rivest vào năm 1994 và 1998.

Chúng rất giống nhau, với RC6 bổ sung thêm một phép nhân trong mật mã để làm cho nó trở nên mạnh mẽ hơn.

Cả hai mật mã đều sử dụng cùng một cơ chế thiết lập khóa và cùng các hằng số ma thuật có tên P và Q.

Việc triển khai RC5/6 từ phần mềm độc hại của nhóm Equation đặc biệt thú vị và đáng được chú ý đặc biệt vì tính đặc thù của nó.

Dưới đây là cách nó trông trong mã giả:

```
*(_DWORD *)buf = 0xB7E15163;
i = 1;
do
{
*(_DWORD *)(buf + 4 * i) = *(_DWORD *)(buf + 4 * i - 4) - 0x61C88647;
++i;
}
while ( i < 44 );
```

Người ta có thể nhận thấy ngay các hằng số 0xB7E15163 và 0x61C88647.

Dưới đây là cách mã thiết lập khóa RC6 thông thường trông như thế nào:

```
void RC5_SETUP(unsigned char *K) /* secret input key K[0...b-1] */
{ WORD i, j, k, u=w/8, A, B, L[c];
 /* Initialize L, then S, then mix key into S */

 for (i=b-1,L[c-1]=0; i!=-1; i--) L[i/u] = (L[i/u]<<8)+K[i];

 for (S[0]=0xB7E15163,i=1; i<t; i++) S[i] = S[i-1]+0x9E3779B9;

[...]
}
```

Điều thú vị là, việc sử dụng hằng số Q trong mã tham chiếu có phần khác biệt.

Bên trong phần mềm độc hại của nhóm Equation, thư viện mã hóa sử dụng một phép trừ với hằng số 0x61C88647.

Trong hầu hết các mã RC5/6 có sẵn công khai, hằng số này thường được lưu trữ dưới dạng 0x9E3779B9, về cơ bản là -0x61C88647.

Vì phép cộng nhanh hơn phép trừ trên một số phần cứng nhất định, nên việc lưu trữ hằng số dưới dạng âm và cộng nó thay vì trừ nó là hợp lý.

Tìm kiếm "0x61C88647 0xB7E15163" trên Google chỉ cho kết quả khoảng hai trang, điều này cho thấy sự kết hợp các hằng số này tương đối hiếm.

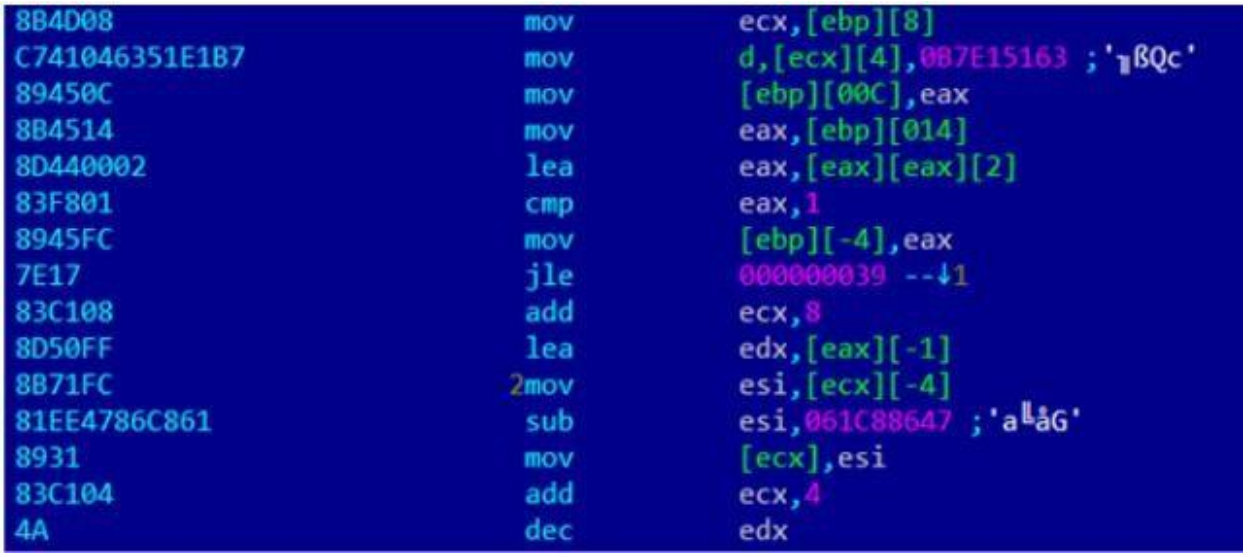
Hầu hết các kết quả đều có trên các diễn đàn Trung Quốc.

Tìm kiếm hằng số nghịch đảo 2 "0x9E3779B9 0xB7E15163" cho kết quả khổng lồ 2500.

```
5.1 Definition of initialization constants

   Two constants, Pw and Qw, are defined for any word size W by the
   expressions:

        Pw = Odd((e-2)*2**W)

        Qw = Odd((phi-1)*2**W)

   where e is the base of the natural logarithm (2.71828 ...), and phi
   is the golden ratio (1.61803 ...), and 2**W is 2 raised to the power
   of W, and Odd(x) is equal to x if x is odd, or equal to x plus one if
   x is even.  For W equal to 16, 32, and 64, the Pw and Qw constants
   are the following hexadecimal values:

#define P16  0xb7e1
#define Q16  0x9e37
#define P32  0xb7e15163
#define Q32  0x9e3779b9
#define P64  0xb7e151628aed2a6b
#define Q64  0x9e3779b97f4a7c15
#if W == 16
#define Pw   P16 /* Select 16 bit word size */
#define Qw   Q16
#endif
#if W == 32
#define Pw   P32 /* Select 32 bit word size */
#define Qw   Q32
#endif
#if W == 64
#define Pw   P64 /* Select 64 bit word size */
#define Qw   Q64
#endif
```

Điều thú vị là, Regin cũng triển khai các hằng số tương tự trong mã RC5 của nó.

Dưới đây là cách mã thiết lập khóa RC5 trông như thế nào trong Regin



Tổng cộng, chúng ta đã xác định được 20 phiên bản biên dịch khác nhau của mã RC5/6 trong phần mềm độc hại của nhóm Equation.

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File .DOCX và .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: [Mã lớp]-ExeX_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).

  *Ví dụ: [*NT101.K11.ANTT*]-Exe01_Group03.*
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài nộp.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá:

- Hoàn thành tốt yêu cầu được giao.
- Có nội dung mở rộng, ứng dụng.

Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.

HẾT