[RE012-1] Phân tích mã độc lợi dụng dịch Covid-19 để phát tán giả mạo “Chỉ thị của thủ tướng Nguyễn Xuân Phúc” – Phần 1

blog.vincss.net/vi/re012-1-phan-tich-ma-doc-loi-dung-dich-covid-19-de-phat-tan-gia-mao-chi-thi-cua-thu-tuong-nguyen-xuan-phuc-phan-1-2/

17/10/2020

*Lợi dụng tình hình diễn biến của dịch COVID-19 hiện tại đang rất phức tạp, nhiều nhóm tin tặc đã và đang âm thầm thực hiện các chiến dịch APT nhắm vào các cá nhân và tổ chức nhằm trục lợi. Tại Việt Nam cũng không ngoại lệ. Mới đây chúng tôi ghi nhận mẫu mã độc (nghi ngờ từ nhóm **Mustang Panda**) giả mạo chị thị của thủ tướng Nguyễn Xuân Phúc về phòng tránh dịch COVID-19. Trong bài viết này chúng tôi sẽ phân tích phương thức mà kẻ tấn công sử dụng để lây nhiễm vào máy người dùng.*

1. Thông tin về sample

File name: *Chi Thi cua thu tuong nguyen xuan phuc.rar*

File Hash (SHA-256): bbbeb1a937274825b0434414fa2d9ec629ba846b1e3e33a59c613b54d375e4d2

File Size: 172 KB

File type: RAR

File Timestamps: 2020:03:03 14:46:12

Archived File Name: Chi Thi cua thu tuong nguyen xuan phucChi Thi cua thu tuong nguyen xuan phuc.lnk

Hình 1: Nội dung của tài liệu xuất hiện khi mã độc thực thi

2. Phân tích mã độc

2.1. Phân tích hành vi của mã độc

Theo thông tin ở trên, mã độc gửi kèm email phishing là một file nén. Trong file nén này chứa một file **Chi Thi cua thu tuong nguyen xuan phuc.lnk** có kích thước **712 KB**:

Hình 2: Nội dung trong file nén

File .lnk đơn giản là một shorcut được Windows sử dụng làm tham chiếu đến file gốc. Các file này thường sử dụng cùng một biểu tượng với file gốc, nhưng thêm một mũi tên cuộn tròn nhỏ để cho biết nó trỏ đến một vị trí khác. Khi người dùng vô tình mở file .lnk trong file nén trên, hành vi của mã độc sẽ diễn ra theo trình tự:

¨ Khởi chạy **cmd.exe**, mục đích để gọi **mshta.exe** với tham số truyền vào là file .lnk đã được giải nén tạm ở thư mục **%Temp%**:

Hình 4: Thực thi mshta.exe với tham số truyền vào là file lnk

¨ **mshta.exe** có nhiệm vụ phân tích file, tìm kiếm và thực thi script được nhúng trong file. Từ đây, thực hiện các hành động sau:

- Tạo các file **3.exe, http_dll.dll, http_dll.dat, Chi Thi cua thu tuong nguyen xuan phuc.doc** trong thư mục **%LocalAppData%Temp**.
- Khởi chạy **3.exe**, tiến trình này sẽ tạo các file **unsecapp.exe, http_dll.dll, http_dll.dat** trong thư mục **%AllUsersProfile%Microsoft Malware Protectionydy**.
- Thiết lập run key **Microsoft Malware Protectionydy** trong Registry (**HKLMSOFTWAREMicrosoftWindowsCurrentVersionRun & HKCUSOFTWAREMicrosoftWindowsCurrentVersionRunMicrosoft Malware Protectionydy**) trỏ tới file **unsecapp.exe** đã tạo ở trên.
- Gọi **WINWORD.EXE** để mở tài liệu **%Temp%Chi Thi cua thu tuong nguyen xuan phuc.doc** với nội dung như ở Hình 1 nhằm đánh lừa người dùng.

Hình 5: Luồng thực thi của các tiến trình

¨ Tiến trình **unsecapp.exe** sau khi thực thi sẽ kết nối tới C2 là **vietnam[.]zing[.]photos**:

Hình 6: Tiến trình unsecapp.exe kết nối tới C2

Hai file **3.exe** và **unsecapp.exe** thực chất là cùng là một file và có Certificate nhằm qua mặt các phần mềm Antivirus:

Hình 7: 3.exe và unsecapp.exe trùng hash

Hình 8: Thông tin Certificate mà độc sử dụng

2.2. Phân tích chi tiết file lnk và VBScript

Như mô tả ở phần trên, khi người dùng mở file **Chi Thi cua thu tuong nguyen xuan phuc.lnk** trong **Chi Thi cua thu tuong nguyen xuan phuc.rar**, **mshta.exe** sẽ được gọi để thực thi script. Như vậy, nội dung của script này phải được nhúng sẵn trong file .lnk. Sử dụng **010 Editor** để mở file .lnk và tìm kiếm chuỗi **<script**, kết quả có được thông tin về đoạn VBScript được nhúng trong file:

Hình 9: Nội dung của script nhúng trong file

Trích xuất toàn bộ nội dung của script. Nội dung của script như sau:

¨ Khai báo các biến **CAwyFTsgCQ, yilJSYTMMh, TPDgWjZcyJ** và gán lần lượt nội dung của **3.exe, http_dll.dll, http_dll.dat** cho từng biến:

Hình 10: Tạo biến chứa nội dung của file

¨ Tạo lập đường dẫn cho các files và gọi hàm **vSWGUThohAGJ** để tạo files:

Hình 11: Tạo các file 3.exe, http_dll.dll, http_dll.dat

¨ Thực thi file **3.exe** đã tạo, tạo tài liệu **Chi Thi cua thu tuong nguyen xuan phuc.doc** và mở tài liệu này để đánh lừa người dùng.

Hình 12: Thực thi 3.exe và mở tài liệu Chi Thi cua thu tuong nguyen xuan phuc.doc

2.3. Phân tích chi tiết các payload

2.3.1. Phân tích unsecapp.exe

Như đã đề cập ở trên, mã độc sau khi thực thi thành công sẽ thiết lập run key **Microsoft Malware Protectionydy** trong Registry (**HKLMSOFTWAREMicrosoftWindowsCurrentVersionRun**) trỏ tới file **unsecapp.exe**. File này bản chất là **EHttpSrv.exe (*ESET HTTP Server Service*)** thuộc sản phẩm **Eset Smart Security** của hãng **ESET**. Kẻ tấn công đã lợi dụng file này để thực hiện kĩ thuật **DLL side-loading** nhằm tải và thực thi mã độc nằm trong thư viện **http_dll.dll**.

Hình 13: unsecapp.exe gọi hàm LoadLibraryW để nạp http_dll.dll

2.3.2. Phân tích http_dll.dll

http_dll.dll sau khi được nạp sẽ thực thi code tại **DllMain**, tại đây mã độc gọi hàm thực hiện công việc sau:

¨ Lấy địa chỉ thuộc **unsecapp.exe** tính từ **base address + 0x157A**.

¨ Gọi hàm **VirtualProtect** để thay đổi **16 bytes** từ địa chí tính toán ở trên thành **PAGE_EXECUTE_READWRITE**.

¨ Patch code tại địa chỉ đó thông qua kĩ thuật **push – ret** để nhảy tới hàm thực hiện nhiệm vụ giải mã mà thực thi Shellcode.

Hình 14: Sử dụng kĩ thuật push-ret để nhảy tới hàm tại địa chỉ 0x10001230

Tại hàm **DecryptShellCodeAndExcecute (0x10001230)**, mã độc tiếp tục thực hiện:

¨ Cấu thành đường dẫn tới **http_dll.dat**, file này chứa payload đã bị mã hóa:

Hình 15: Cấu thành đường dẫn tới http_dll.dat

¨ Gọi hàm **FileReadAll (0x10001030)**, đọc toàn bộ nội dung của **http_dll.dat** vào vùng nhớ đã cấp phát:

Hình 16: Hàm FileReadAll chịu trách nhiệm đọc nội dung http_dll.dat

Hình 17: Code của hàm FileReadAll

¨ Trích xuất key giải mã (**10 bytes đầu** của **http_dll.dat**), cấp phát vùng nhớ và copy toàn bộ dữ liệu của **http_dll.dat** vào vùng nhớ đã được cấp phát. Gọi hàm **XorDecrypt (0x100014B0)** để giải mã payload mới trên bộ nhớ:

Hình 18: Thực hiện giải mã payload mới trên bộ nhớ

¨ Cuối cùng gọi hàm **VirtualProtect** để thay đổi vùng nhớ của payload mới thành **PAGE_EXECUTE_READWRITE** và gọi thẳng tới payload này để thực thi. Payload cuối cùng này sẽ làm nhiệm vụ giải mã cấu hình có thông tin về thư mục "**Microsoft Malware Protectionydy**" dùng để lưu các payload, thông tin về C2 như đã đề cập ở trên và thực hiện nhiệm vụ kết nối tới C2.

Hình 19: Thực thi payload mới đã giải mã trên bộ nhớ

Bằng thông tin phân tích được ở trên, có thể giải mã và thu được payload mới mà không cần debug:

Hình 20: http_dll.dat trước và sau khi giải mã

Payload có được là một dll (**HT.dll**):

Hình 21: Payload mới là một dll

Bài viết xin được tạm dừng tại đây, trong phần tiếp theo chúng tôi sẽ phân tích chi tiết về cách thức hoạt động của payload cuối cùng (HT.dll).

Indicators of compromise (IOCs)

Dropped file:

%LocalAppData%Temp

1. 3.exe [SHA256: c3159d4f85ceb84c4a0f7ea9208928e729a30ddda4fead7ec6257c7dd1984763]
2. http_dll.dll [SHA256: 79375c0c05243354f8ba2735bcd086dc8b53af709d87da02f9206685095bb035]
3. http_dll.dat [SHA256: b62d35d8edae874a994fff12ec085a0bf879c66b3c97fd13fe0a335b497342e5]
4. Chi Thi cua thu tuong nguyen xuan phuc.doc [SHA256: e3556d6ba5e705b85599b70422928165c8d4130074029a8dcd04a33f4d1aa858]

%AllUsersProfile%Microsoft Malware Protectionydy

1. unsecapp.exe [SHA256: c3159d4f85ceb84c4a0f7ea9208928e729a30ddda4fead7ec6257c7dd1984763]
2. http_dll.dll [SHA256: 79375c0c05243354f8ba2735bcd086dc8b53af709d87da02f9206685095bb035]
3. http_dll.dat [SHA256: b62d35d8edae874a994fff12ec085a0bf879c66b3c97fd13fe0a335b497342e5]

Persistence Registry:

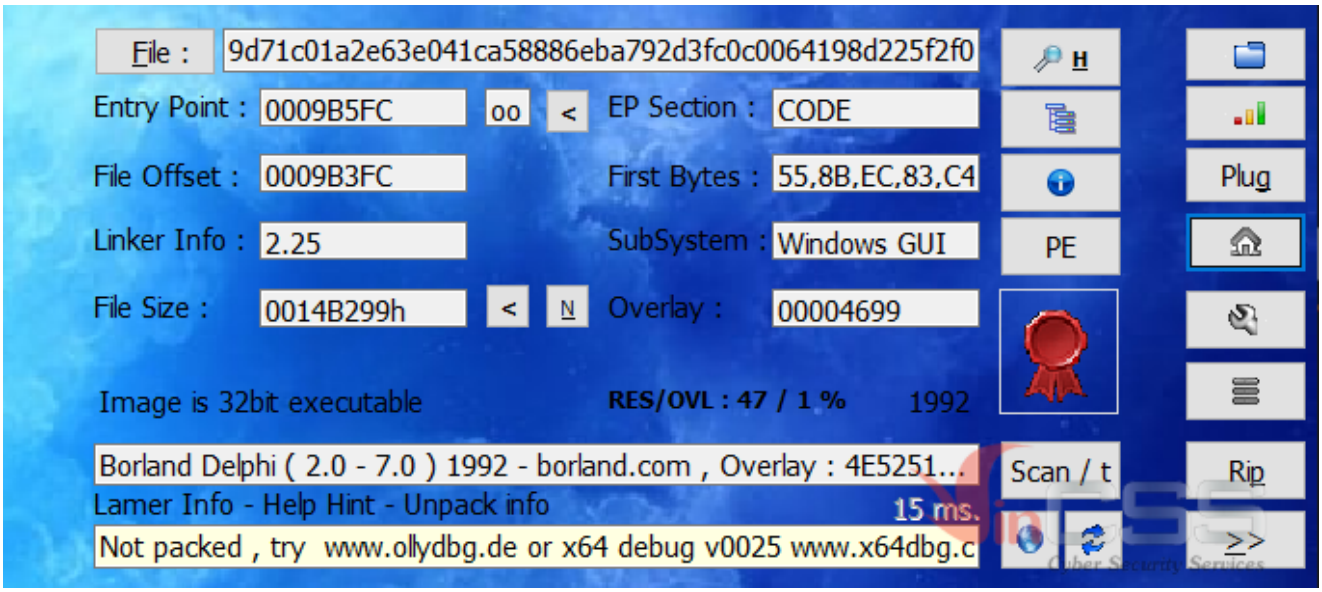
HKLMSOFTWAREMicrosoftWindowsCurrentVersionRunMicrosoft Malware Protectionydy = C:ProgramDataMicrosoft Malware Protectionydyunsecapp.exe”
HKCUSOFTWAREMicrosoftWindowsCurrentVersionRunMicrosoft Malware Protectionydy = C:ProgramDataMicrosoft Malware Protectionydyunsecapp.exe”

C2:

Domain: vietnam[.]zing[.]photos
IP: 104.160.44.85

R&D Center – VinCSS (a member of Vingroup)

↗ Trở lại
Bài viết liên quan



17/12/2023

[RE016] Malware Analysis: ModiLoader

1. Giới thiệu Gần đây, tôi có tìm hiểu một dòng loader có tên là ModiLoader. Loader này được phát tán thông qua các dịch vụ Malspam để lừa người dùng thực thi mã độc. Tương tự như các dòng loader khác, ModiLoader cũng thông qua nhiều bước (stage) để tải về payload cuối cùng có nhiệm vụ đánh […]

12/12/2023

[RE027] Nhóm APT Mustang Panda có thể vẫn đang tiếp tục hoạt động tấn công vào các tổ chức tại Việt Nam

Tại VinCSS, chúng tôi liên tục chủ động theo dõi tình hình an ninh mạng, săn tìm các mẫu mã độc và đánh giá mức độ nguy hiểm của chúng, đặc biệt là các mẫu mã độc nhắm tới Việt Nam. Gần đây, trong quá trình thực hiện hunting trên nền tảng của VirusTotal, thực hiện tìm kiếm các mẫu byte đặc trưng liên quan tới nhóm Mustang Panda (PlugX), chúng tôi đã phát hiện một loạt mẫu mã độc mà chúng tôi nghi ngờ là của nhóm này được tải lên từ Việt Nam.

24/04/2022

[RE026] A Deep Dive into Zloader – the Silent Night

Zloader, một banking trojan còn biết đến với những tên gọi khác như Terdot hay Zbot. Dòng trojan này được phát hiện lần đầu tiên vào năm 2016, và theo thời gian số lượng phát tán của nó liên tục gia tăng. Code của Zloader được cho là xây dựng dựa trên mã nguồn bị rò rỉ của mã độc ZeuS nổi tiếng. Vào năm 2011, khi mã nguồn của ZeuS được công khai thì từ đó tới nay nó được sử dụng trong nhiều mẫu mã độc khác nhau.

11/10/2021

[RE024] Tìm hiểu về IDA Microcode

Giới thiệu Tổng quan khi biên dịch một chương trình, compiler sẽ thực hiện như sau: Các bước cơ bản của một chương trình compiler Khi decompile một chương trình sang mã giả C, hexrays sẽ làm điều ngược lại: Các bước cơ bản của một chương trình decompiler Một trong những bước quan […]

27/09/2021

[RE025] TrickBot … many tricks

Được phát hiện lần đầu vào năm 2016, tới thời điểm hiện tại TrickBot (còn được biết đến với những tên gọi khác như TrickLoader hay Trickster) đã trở thành một trong những mã độc nguy hiểm và phổ biến nhất hiện nay. Những kẻ đứng đằng sau TrickBot liên tục phát triển để thêm các tính năng và thủ thuật mới. Mã độc này được phát triển dưới dạng mô-đun, theo đó payload chính sẽ chịu trách nhiệm tải các plugin khác có khả năng thực hiện các tác vụ cụ thể, bao gồm đánh cắp tài khoản và thông tin nhạy cảm, cung cấp khả năng truy cập từ xa, lây lan qua mạng cục bộ, và tải xuống phần mềm độc hại khác.