

Phân tích mã độc “KẾ HOẠCH, NHIỆM VỤ TRỌNG TÂM NĂM 2020.doc” đính kèm email phishing - VinCSS Blog

By Yến Hứa

Published: 2019-12-17 · Archived: 2026-04-06 00:21:09 UTC

Table of Contents

- - [Thông qua Steve Miller \(@stvemillertime\) của FireEye, chúng tôi có được thông tin và mẫu mã độc này. Do mẫu mã độc này có nội dung nhắm vào Việt Nam nên VinCSS quyết định sẽ phân tích để chia sẻ cho cộng đồng an ninh mạng Việt Nam.](#)
- [Thông tin cơ bản](#)
- [Phân tích sơ lược](#)
- [Phân tích hành vi](#)
- [Phân tích mã thực thi](#)
 - [Stage 1: Dump decoded PE payload](#)
 - [Stage2: Phân tích PE32 đã dump](#)
 - [Stage 3: Phân tích file mpsvc.dll:](#)
- [Indicators of compromise \(IOCs\)](#)

Thông qua Steve Miller (@stvemillertime) của FireEye, chúng tôi có được thông tin và mẫu mã độc này. Do mẫu mã độc này có nội dung nhắm vào Việt Nam nên VinCSS quyết định sẽ phân tích để chia sẻ cho cộng đồng an ninh mạng Việt Nam.

Thông tin cơ bản

- File name: **KẾ HOẠCH, NHIỆM VỤ TRỌNG TÂM NĂM 2020.doc**
- File Timestamps: **2019-12-09 18:50:00**
- Size: **783.77 KB (802578 bytes)**
- File type: **RTF (Rich Text Format)**
- Pages: **6**
- File Hash (SHA-256): **bcb226f7d614c905abc94aef9e096b03921cc8e2077c464224084670213e10b5**

Phân tích sơ lược

Sử dụng công cụ [rtfobj](#) trong bộ [oletools](#) để kiểm tra file. Thông tin có được như sau:

Dựa vào dấu hiệu *Equation* có thể khẳng định tài liệu này khai thác lỗ hổng của của [Microsoft Office Equation Editor](#) (CVE-2017-11882, CVE-2018-0802). Ngoài ra, như trong hình có thể thấy tài liệu này nhúng thêm một **OLE Stream 0** (*wd32PrvSE.wmf*) có kích thước **208 KB (212992 bytes)**. Tiếp tục sử dụng **rtfobj** để trích xuất object này:

So sánh nội dung của file vừa extract ở trên với nội dung của các file **8.t** và **e.m** mà chúng tôi đã phân tích trước đây thì có thể thấy kỹ thuật thực hiện tương tự nhau.

wd32PrvSE.wmf:

8.t:

e.m:

Phân tích hành vi

Tài liệu khi mở bằng trình đọc **Microsoft Word** sẽ thực hiện tạo các files tại thư mục **%Temp%** và tạo shortcut trong thư mục **Startup** của Windows (*%AppData%\Microsoft\Windows\Start Menu\Programs\Startup*) để khởi chạy mã độc khi người dùng khởi động lại máy:

Phân tích mã thực thi

Stage 1: Dump decoded PE payload

Cách thiết lập để thực hiện debug đã được đề cập ở nhiều bài viết. Khi mở tài liệu bằng ứng dụng **Microsoft Word**, tiến trình **EQNEDT32.exe** sẽ được khởi chạy, thông qua lỗi của ứng dụng này để tạo file **wd32PrvSE.wmf** trong thư mục **%Temp%**. Đặt bp tại hàm **CreateFileA/W** ta sẽ thấy tiến trình đọc file wmf:

Tiếp theo sẽ gọi hàm **ReadFile** để đọc nội dung của **wd32PrvSE.wmf** vào vùng nhớ đã được cấp phát:

Tương tự như các sample đã từng phân tích, sử dụng vòng lặp xor để thực hiện giải mã toàn bộ nội dung của file **wd32PrvSE.wmf** trong memory tại địa chỉ **0x1EE0000** (trên máy phân tích)

Thực hiện dump PE mới này và lưu lại để thực hiện phân tích tiếp. File dump được là một *PE32 exe*:

Stage2: Phân tích PE32 đã dump

Từ **WinMain** sẽ gọi tới hàm **DropFilesAndCreateLnk** (*sub_00401200*). Hàm này thực hiện cấu thành đường dẫn đầy đủ cho các files **mpsvc.dll**; **RuntimeBroke.exe**:

Sau đó ghi 2 file trên vào thư mục **%Temp%**:

RuntimeBroke.exe chính là **MsMpEng.exe** của *Windows Defender*:

Tiếp theo tạo file **StructuredQuery.tmp**:

Áp dụng kỹ thuật persistence thông qua startup folders bằng cách tạo file **RuntimeBroke.lnk** tại (*%AppData%\Microsoft\Windows\Start Menu\Programs\Startup*):

Code tại hàm **CreateLink** có nội dung như sau:

Ta có file shortcut là **RuntimeBroke** với target trỏ tới **RuntimeBroke.exe** ở thư mục **%Temp%** như sau:

Như vậy, khi người dùng khởi động lại máy thì tiến trình **RuntimeBroke.exe** sẽ khởi chạy. Thông qua kỹ thuật *DLL SideLoading*, **RuntimeBroke.exe** sẽ nạp file **mpsvc.dll** chứa mã độc để thực thi:

Stage 3: Phân tích file **mpsvc.dll**:

File này exports các hàm sau:

Kiểm tra thì thấy nó chỉ gọi tới **ServiceCrtMain**. Tại hàm, sẽ thực hiện mở file **StructuredQuery.tmp** thông qua hàm **IsStructuredQueryTmpNotExisted**:

Thông qua một vòng lặp liên tục để thực hiện việc truy xuất tới C2:

Code bên trong **sub_6A7F20A0 (offset 0x14A0)** sẽ thực hiện nhiệm vụ cấu thành các chuỗi sau trong memory:

Sau đó khởi tạo kết nối Internet với **User-Agent: HTTPS**, mở HTTP session tới **cloudflare-dns[.]com:443**, cấu thành target Object "**dns-query?name=pjfdknrvbz.mefound.com&type=A**" phục vụ cho hàm **HttpOpenRequest** nhằm khởi tạo một HTTP request với phương thức **GET**. Cuối cùng gửi request tới HTTP Server và gọi hàm **InternetReadFile** để đọc dữ liệu vào vùng buffer đã được cấp phát:

Căn cứ thông tin trên hình thì có thể thấy kẻ tấn công đang thực hiện kỹ thuật **DNS over HTTPS** (<https://developers.cloudflare.com/1.1.1.1/dns-over-https/>).

Do việc kết nối tới C2 không thành công nên quá trình phân tích tạm thời dừng lại tại đây. Chúng tôi sẽ tiếp tục phân tích chi tiết malware *mpsvc.dll* và cập nhật thêm khi có các thông tin cụ thể.

Indicators of compromise (IOCs)

Dropped files:Location: %Temp% folder

1. RuntimeBroke.exe—21.7 KB (22,224 bytes)

(Original filename: MsMpEng.exe)

SHA-256: 33bc14d231a4afaa18f06513766d5f69d8b88f1e697cd127d24fb4b72ad44c7a

2. mpsvc.dll—141 KB (144,384 bytes)

SHA-256: 87f0ba25135f7a42a7219b8a7aa1013755f03ad11b6a897a9066e3089b438432

3. StructuredQuery.tmp—0 bytes

Persistence:Startup folder: %AppData%\Microsoft\Windows\Start Menu\Programs\Startup

File: RuntimeBroke (shortcut); Target: %Temp%\RuntimeBroke.exe

Network:

GET https://cloudflare-dns[.]com/dns-query?name=pjfdknrvbz[.]mefound[.]com&type=A HTTP/1.1

Accept: application/dns-json

User-Agent: HTTPS

Host: cloudflare-dns.com

Name: pjfdknrvbz[.]mefound[.]com

Address: 185.244.150.84

Để tiện theo dõi, chúng tôi cung cấp bài phân tích dưới dạng PDF:

File Name: CSS-RD-ADV-191218-009_Phan_tich_ma_doc_Ke.hoach.Nhiem.Vu.Trong.Tam.2020.pdf

File hash (SHA-256): e09d4d203acfa90a6efe71dcd3fb54cf656d66180827e0ea7ac11a4b888d7aa7

[CSS-RD-ADV-191218-009_Phan_tich_ma_doc_Ke.hoach.Nhiem.Vu.Trong.Tam.2020.pdf – Google Drive](#)

Tran Trung Kien (aka m4n0w4r)

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

Source: https://blog.vincss.net/vi/re009-phan-tich-ma-doc-ke-hoach-nhiem-vu-trong-tam-nam-2020-doc-dinh-kem-email-phishing-2/