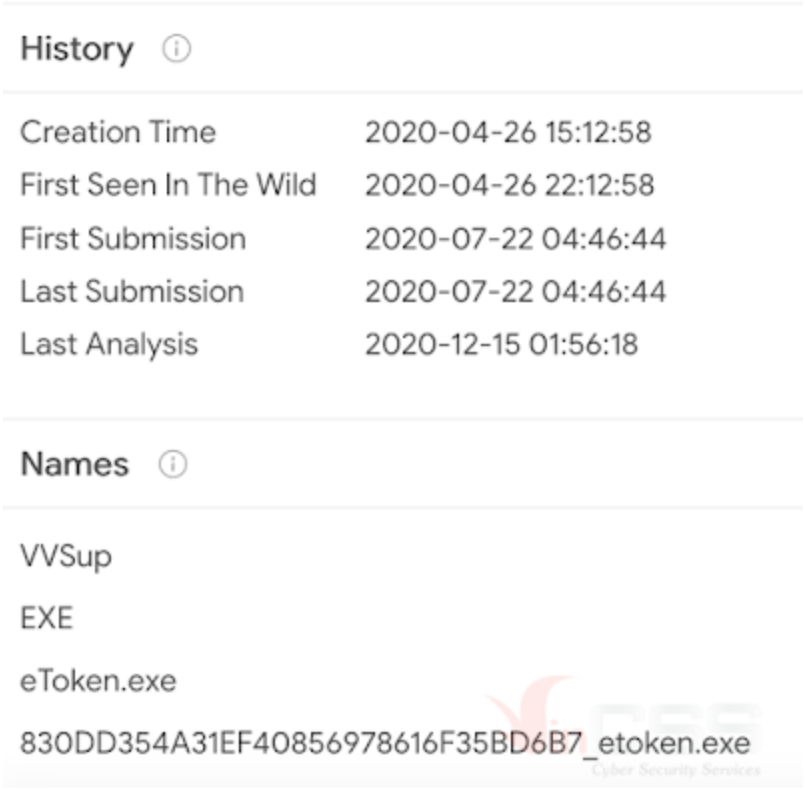
[RE017-1] Phân tích kỹ thuật dòng mã độc mới được sử dụng để tấn công chuỗi cung ứng nhắm vào Ban Cơ yếu Chính phủ Việt Nam của nhóm tin tặc Panda Trung Quốc (Phần 1)

blog.vincss.net/2020/12/phan-tich-ky-thuat-dong-ma-doc-moi-co-nhieu-dau-hieu-lien-quan-toi-nhom-tin-tac-Panda.html

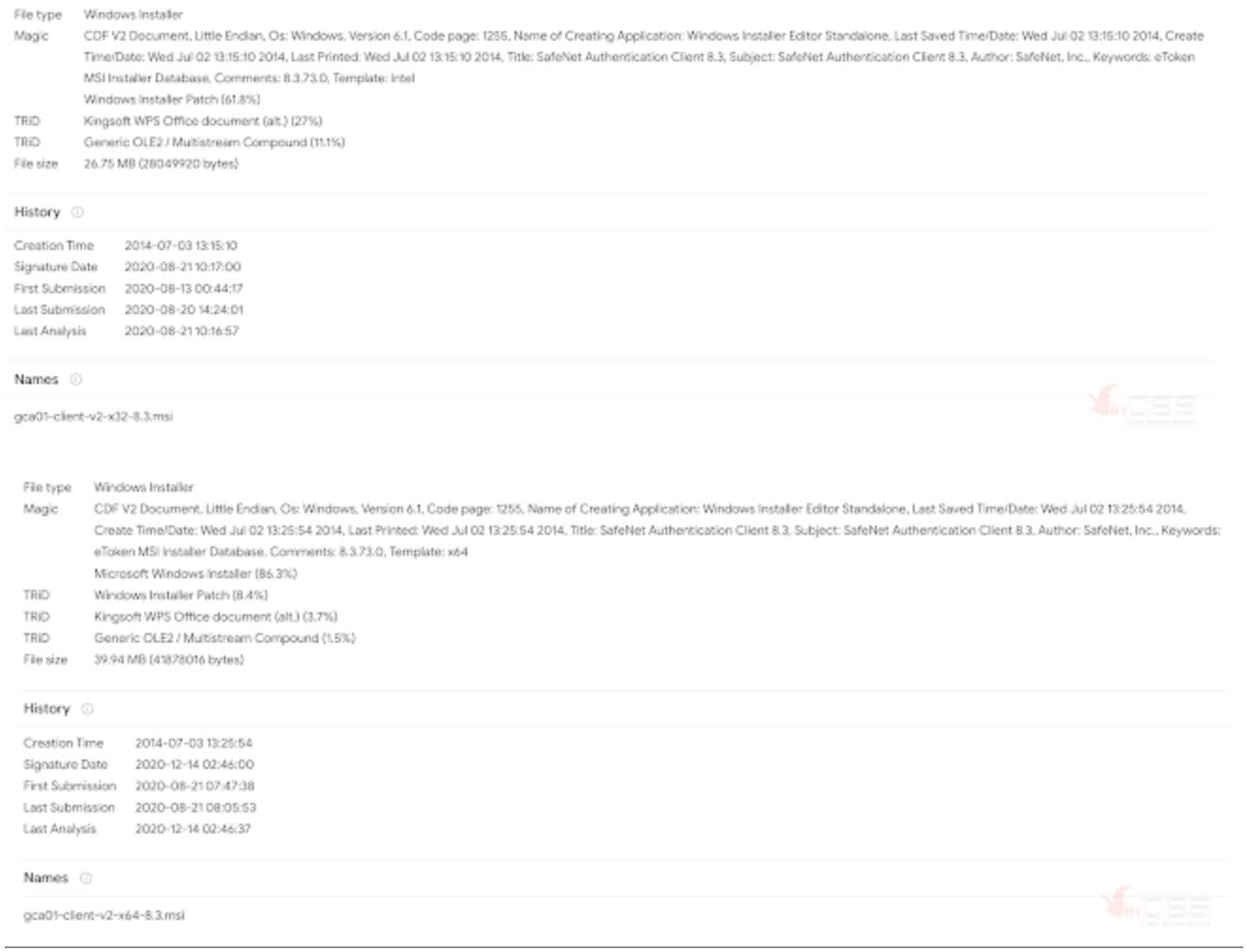
I. Mở đầu

Trong quá trình thực hiện các hoạt động theo dõi và phân tích các mẫu mã độc, chúng tôi phát hiện một bài blog thú vị của NTT tại đây. Lần theo các hash trong bài viết này, chúng tôi thấy một mã hash cho thông tin chú ý trên VirusTotal:



Hình 1. Thông tin từ mã hash trên blog của NTT

Nhân sự kiện một nhóm tin tặc được cho là của Nga tấn công khai thác chuỗi cung ứng phần mềm để nhắm vào hàng loạt các cơ quan lớn của nước Mỹ, cùng với việc phát hiện thấy từ khoá **eToken.exe** là một trong số các phần mềm được sử dụng khá phổ biến ở các cơ quan, tổ chức và doanh nghiệp tại Việt Nam, chúng tôi đã sử dụng **eToken.exe** và **SafeNet** làm keywords tìm kiếm trên các nền tảng VirusTotal và Google. Kết quả, chúng tôi đã phát hiện ra thông tin về hai files (1, 2) cài đặt rất đáng chú ý đã được tải lên VirusTotal từ tháng **8/2020**:



Hình 2. Thông tin tìm kiếm được trên VirusTotal

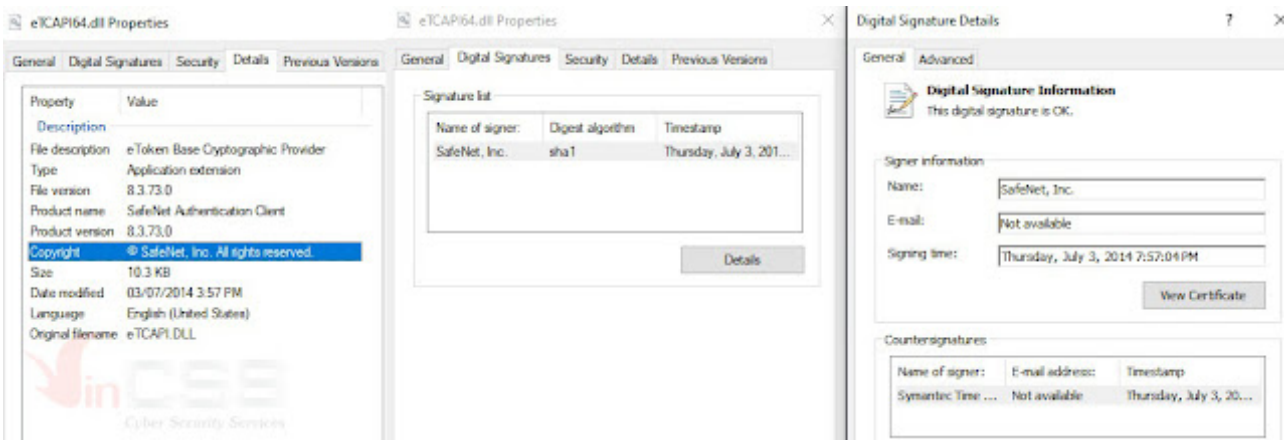
Khi thấy tên các file cài đặt là **gca01-client-v2-x32-8.3.msi** và **gca01-client-v2-x64-8.3.msi** khá quen thuộc, chúng tôi đã thử tải về hai file này từ trang chủ cung cấp. Kết quả, các mã hash này trùng nhau. Tuy nhiên, ở thời điểm hiện tại thì các file trên trang chủ đã được gỡ bỏ và thay thế bằng phiên bản sạch, chính gốc. Bước đầu, chúng tôi nhận định đây có thể là một chiến dịch tấn công có chủ đích APT nhằm vào chuỗi cung ứng phần mềm để từ đó làm bàn đạp tấn công vào các cơ quan, tổ chức và doanh nghiệp quan trọng tại Việt Nam.

Ngày 17/12, ESET đã công bố một phát hiện một chiến dịch tấn công APT được họ gọi là "Chiến dịch SignSight" nhằm vào Ban Cơ yếu Chính phủ Việt Nam. Được biết, ESET đã thông báo cho VNCERT và Ban Cơ yếu Chính phủ Việt Nam và đã nhận được thông tin Ban Cơ yếu đã phát hiện bị tấn công trước đó và đã thông báo cho các cơ quan liên quan đã tải về các phần mềm bị tin tặc can thiệp.

Ở thời điểm phân tích, chúng tôi đã có được hai file cài đặt đã bị tin tặc can thiệp, thay đổi. Toàn bộ chuỗi bài viết này sẽ tập trung vào phân tích các dấu hiệu và kỹ thuật đã được tin tặc áp dụng vào các mẫu mã độc trong hai file cài đặt này.

II. Phân tích file cài đặt

Ứng dụng này tên là "**SafeNet Authentication Clients**" của công ty **SafeNet .Inc**. Các file PE (*Portable Executable*) đa số đều được sign bằng certificate của SafeNet.



Hình 3. Các PE files được sign bằng certificate của SafeNet

Sử dụng công cụ **UniExtract**, chúng tôi trích xuất ra toàn bộ file từ một bộ cài đặt, bộ **x64**. Tổng số file là **218 files**, **68 t**hư mục con, tổng dung lượng là **75.1 MB** (*78,778,368 bytes*). Để tìm ra được file bị tin tặc chèn vào trong số lượng files trên, chúng tôi chỉ tập trung vào phân tích và xác định các files PE không được sign.

Chạy công cụ **sigcheck** thuộc bộ **SysInternals Suite** của Micorsoft, với các thông số kiểm tra signed, hash, quét hết các PE file, kết quả scan hash trên VirusTotal, output là csv file. Sau đó sắp xếp theo unsigned file, kết quả từ VirusTotal, chúng tôi phát hiện ra file đã bị tin tặc chèn vào, được đặt tên là **eToken.exe**.

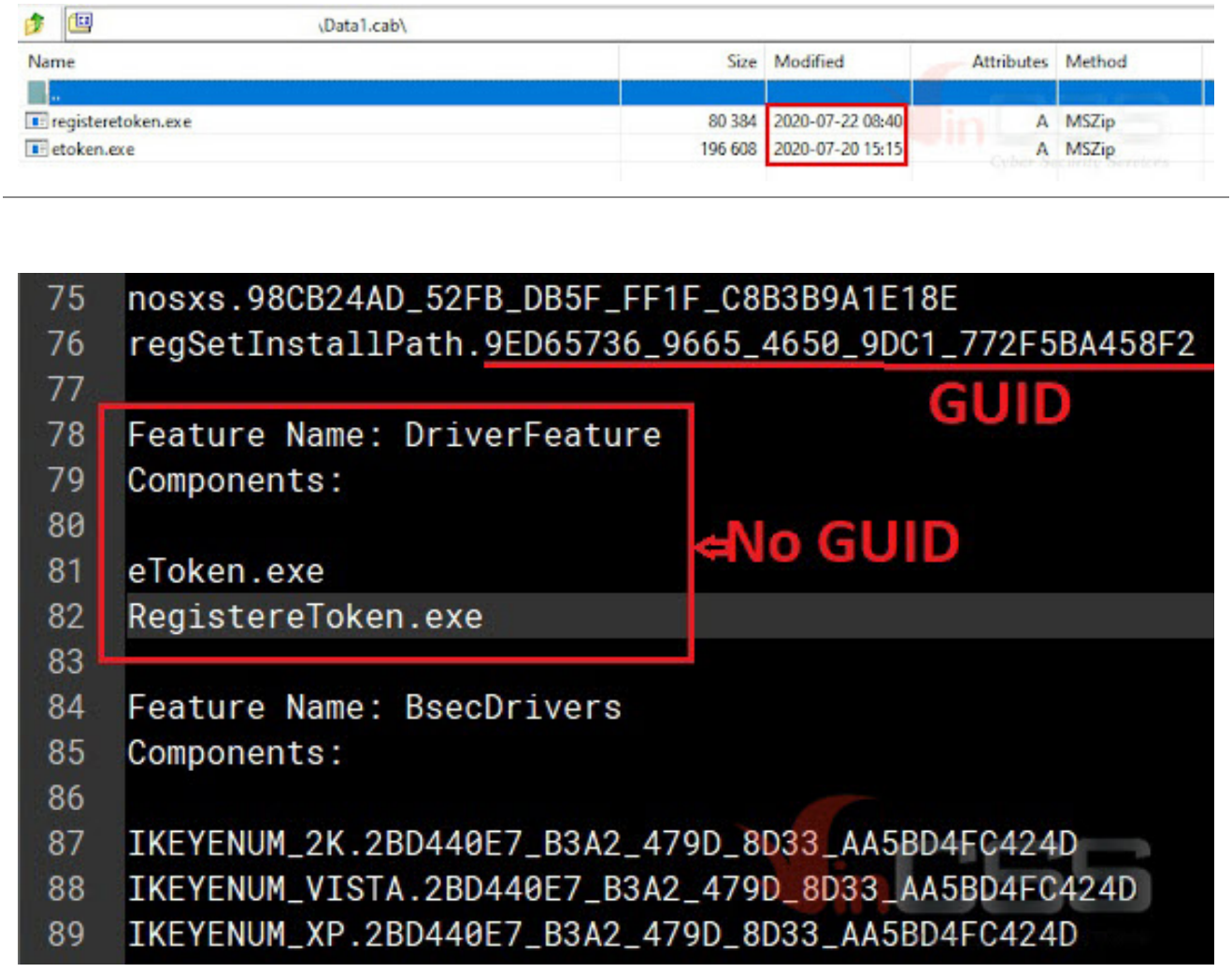
Hình 4. Phát hiện ra file đã bị tin tặc chèn vào

Mã hash của file **eToken.exe** trên trùng với một mã hash trong report của NTTSecurity. Các dấu hiệu lạ khác là PE 32bit nhưng lại nằm trong thư mục x64, các version info như *"Company, Description, Product…"* không hợp lệ với một ứng dụng lớn của một công ty như vậy. Link kết quả scan của file eToken trên VirusTotal.

Vì bộ ứng dụng trên được build với **Visual C++** của Visual Studio 2005, đã cũ, và dùng thư viện Qt4, nên một số file dll của bộ cài đặt này cũng không được signed. Chúng tôi đã kiểm tra từng file và xác định là các file sạch, chỉ còn ba file là nghi ngờ: **RegistereToken.exe**, **eTOKCSP.dll** và **eTOKCSP64.dll**.

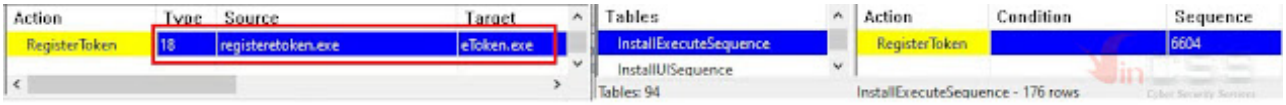
Vậy **eToken.exe** file chính là một malware mà tin tặc đã thêm vào bộ cài đặt của bộ phần mềm nói trên. Để tìm hiểu cách mà eToken.exe được thực thi, chúng tôi tiến hành phân tích file cài đặt, msi file (*Microsoft Windows Installer file*): **gca01-client-v2-x64-8.3.msi**

Tiến hành trích xuất file msi ra dạng raw trước installing, chúng tôi thu được hai file **.cab** (*Microsoft Cabinet file*): **Data1.cab** và **Cabs.w1.cab**. Đây là điểm dị thường vì thông thường một file msi chỉ có một file **.cab** chính. Kiểm tra file **Data1.cab** và MSI log text file, **eToken.exe** và **RegistereToken.exe** đều nằm trong file **Data1.cab**. Và cả hai file **.exe** đều không có **GUID ID** đi kèm:



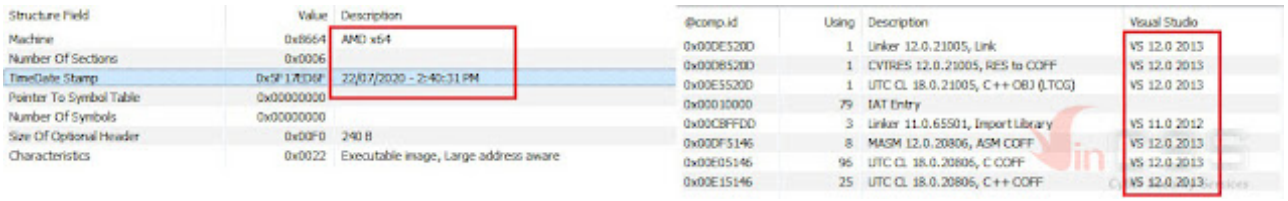
Hình 5. Các file exe không có GUID ID đi kèm

Kiểm tra feature: DriverFeature, và hai file **eToken.exe** và **RegistereToken.exe** của file msi với công cụ **Orca** của Microsoft (*công cụ chuyên dùng để phân tích, sửa đổi msi file*). Thông qua tìm kiếm, ta thấy tin tặc đã chèn thêm một custom action: **RegisterToken** (*không có "e"*) vào file msi và thêm vào **CustomAction** đó vào cuối **InstallExecuteSequence**. **RegistereToken.exe** sẽ được gọi với parameter là **eToken.exe**:



Hình 6. Tin tặc chèn thêm một custom action

Phân tích file **RegistereToken.exe**, chúng tôi thấy file này được build vào thời gian "**Wednesday, 22.07.2020 07:40:31 UTC**", tức **22/07/2020, 2h40m31s PM GMT +7, PE64**, dùng **VC++ 2013**:



Hình 7. Thông tin của file RegistereToken.exe

Mã giả C của **RegistereToken.exe** chỉ làm nhiệm vụ gọi API **WinExec** để thực thi tham số truyền vào:

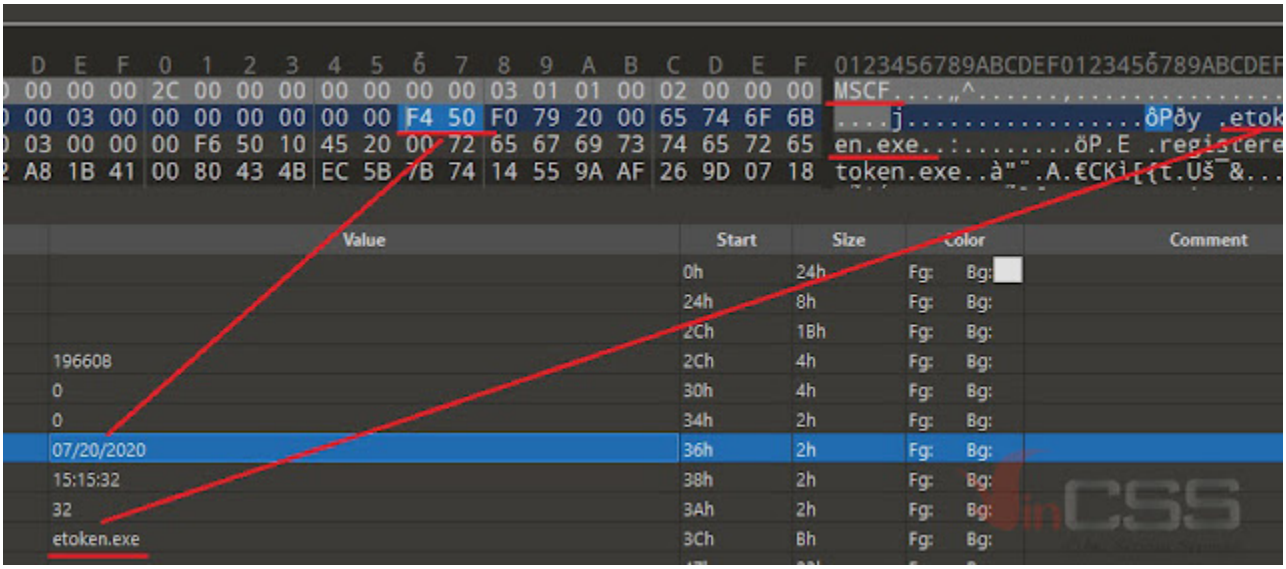
```
10   szExePath[0] = 0;
11   memset(&szExePath[1], 0, MAX_PATH);
12   GetModuleFileNameA(0i64, szExePath, MAX_PATH);
13   strrchr(szExePath, '\\')[1] = 0;
14   pos = (char *)&mask + 31;
15   while ( *++pos != 0 )
16   {
17       ;
18   }
19
20   // pos = point to NULL char szExePath
21   pExeInput = argv[1];
22   i = 0i64;
23   do
24   {
25       aChar = pExeInput[i++];
26       pos[i - 1] = aChar;
27   }
28   while ( aChar );
29   WinExec(szExePath, 0);
30   return 0;
31
```

Hình 8. Nhiệm vụ của RegistereToken.exe

Kết luận, qua các timestamp trong file **Data1.cab** và **RegistereToken.exe**, chúng tôi xác định như sau

- Tin tặc tạo và sửa đổi file install **.msi** và tạo file **Data1.cab** tại timestamp: **20/07/2020 – 15:15** giờ UTC, add file **eToken.exe** vào lúc này.
- Build file **RegistereToken.exe** tại timestamp: **22/07/2020 – 07:40** giờ UTC.
- Add file **RegistereToken.exe** vào **Data1.cab** tại timestamp**: 22/07/2020 – 08:40** giờ UTC.

Ghi chú: Theo Cab file format, hai trường **Date** và **Time** của một file trong file cab là **DOS Datetime format**, mỗi trường là một Word 2 bytes, ghi lại thời gian file được thêm vào theo DOS time. Các chương trình xử lý file Cab sẽ convert và hiển thị theo UTC time. Tức các UTC time trên chính là time hiện tại trên máy tin tặc. Xem thêm tại đây.



Hình 9. Thông tin MS DOS Datetime

III. Phân tích file eToken.exe

1. Phân tích PE Structure file

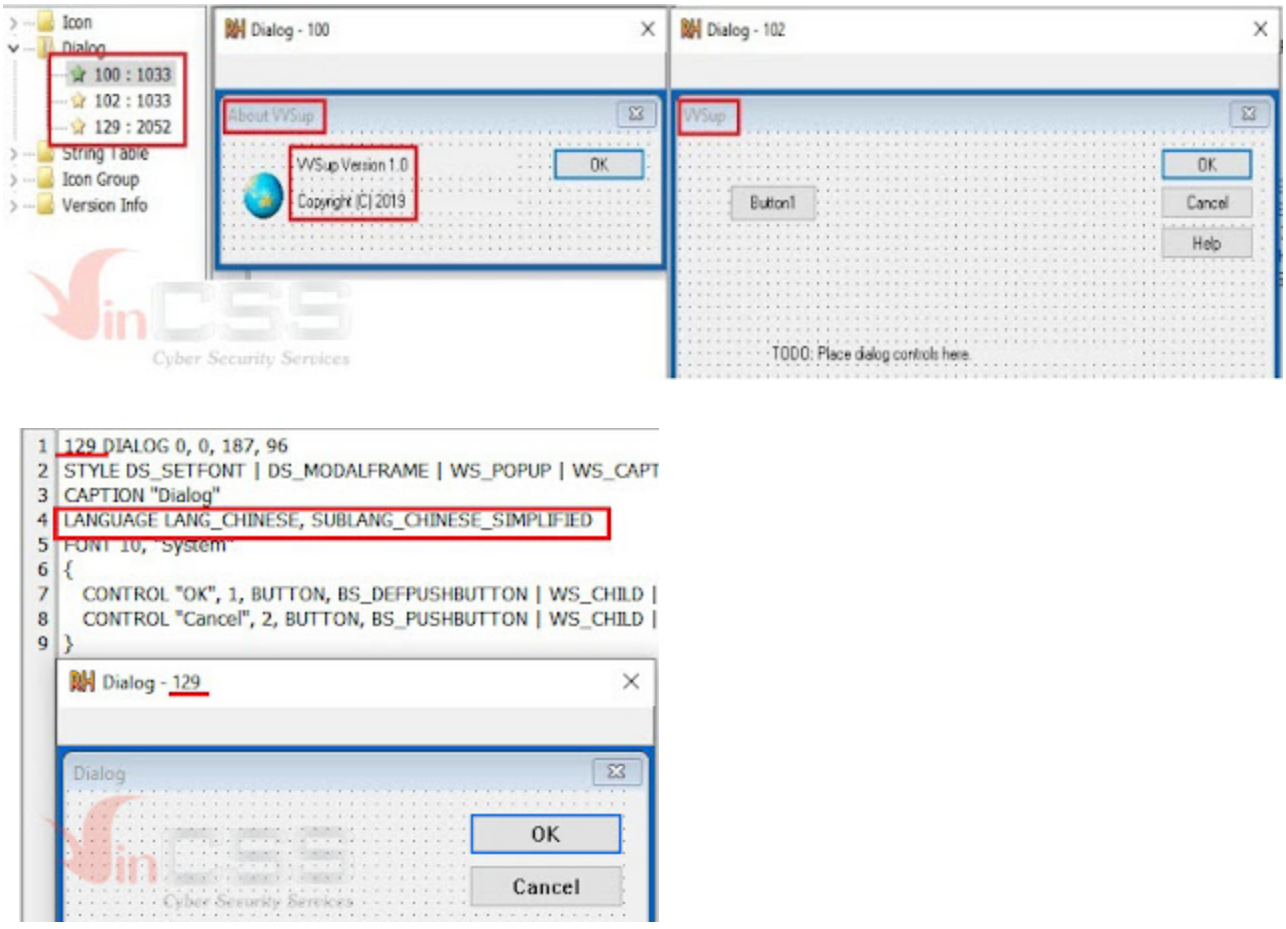
File **eToken.exe**:

- Size = 192 KB (196,608 bytes)
- MD5 = 830DD354A31EF40856978616F35BD6B7
- SHA256 = 97A5FE1D2174E9D34CEE8C1D6751BF01F99D8F40B1AE0BCE205B8F2F0483225C

Các thông tin về compiler, RichID và build timestamp:

- Build bằng **VC++ 6** của Microsoft Visual Studio, Service Pack 6.
- Build lúc: 26/04/2020 – 15:12:58 UTC.
- Checksum đúng, file chưa bị sửa đổi PE Header.
- Link với thư viện **MFC42.dll**, thư viện *Microsoft Foundation Class v4.2* của Microsoft, là thư viện hỗ trợ lập trình GUI trên Windows, luôn đi kèm trong các bộ Visual Studio.
- Link với một thư viện khá đặc biệt: **dbghelp.dll**. Dùng hàm API **MakeSureDirectoryPathExist**. Xem thêm tại đây.

Kiểm tra resource section của file, chúng tôi xác định được đây là một Dialog application, được tạo bằng *MFC Wizard* của Visual Studio 6. Tên project là **VVSup**, tức file **.exe** khi build ra sẽ là **VVSup.exe**. Dialog chính có resource **IDD = 102**, dialog About có **IDD = 100**. Và điểm đặc biệt là còn có một dialog khác có **IDD = 129**, được tin tặc add vào bằng MFC Wizard, và quên (*hay cố ý*) vẫn để resource language là **Chinese**. Dialog này cũng không có control nào được thêm vào, ngoài hai button **OK** và **Cancel** mặc định. Chúng ta sẽ lưu ý dialog này.



Hình 10. Thông tin resource của file

2. Static analysis code của eToken với IDA

eToken.exe (**VVSup.exe**) được build với mode link dynamic DLL với **MFC42.dll**, nên file **.exe** sẽ nhỏ và các hàm của MFC42 libirary sẽ được dễ dàng xác định qua name import của DLL. Quy tắc name mangling của Microsoft VC++ compiler phản ánh tên class, tên hàm, tên thông số, kiểu gọi… của các hàm. IDA giúp chúng ta xác định các hàm import by ordinal của **MFC42.dll** bằng file **mfc42.ids** và **mfc42.idt** đi kèm của IDA.

Tuy nhiên **VVSup** được build với option **RTTI** (*Runtime Type Information*) bị tắt, nên sẽ không có thông tin về **RTTI** và **Virtual Method Table** của đầy đủ các class (lớp) trong file. Chúng ta chỉ có **RTTI** của class **type_info**, lớp **root** của **RTTI**.

(*Còn tiếp...*)

Trương Quốc Ngân (aka HTC)

Chuyên gia Phân tích mã độc - VinCSS (a member of Vingroup)