CASSIDIAN CYBERSECURITY

Disass : a new malware analysis framework

Ivan Fontarensky

Who am I ?

- Ivan Fontarensky
  @ifontarensky
- Working in CSIRT Team for Cassidian CyberSecurity
  - Incident Response
  - Malware Analysis
- Have been “playing around” with malware analysis
- Working on other project : Yara Community

Reason why we build Disass ?

- Cassidian CyberSecurity is involved in the "ACDC" European project.
- We found piles of malware during Incident Response
  - Need to quickly extract valuable information from the malware
- Malware streams received from partners
  - Analysis must be automated
- Malware are evolving fast
  - Building basic analysis scripts is not enough

What is Disass ?

- Framework to ease reverse automation
- Written in python 2.7
- Licensed under GPL v3
- Disass is based on :
  - Distorm3 (linear disassembly engine) by Gil Dabah
  - pefile by Ero Carrera

Disass is not ….

- Disass is not another disassembler
- Disass is not a debugger
- Disass is not an emulator

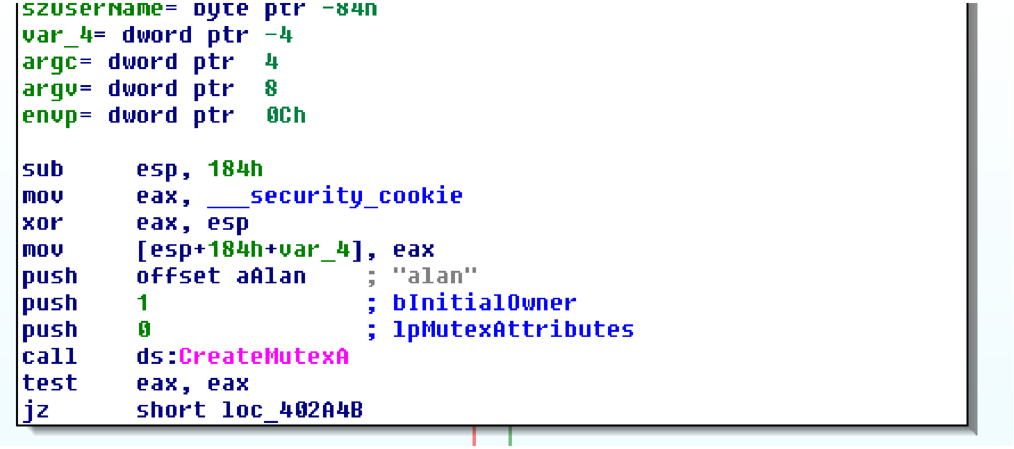
Main functionalities

- Automated disassembly engine
  - Human readable automation scripts
  - Interactive shell to help writing automation script
- Evaluate a possible register value
- Follow both branches in case of conditional jump
- Jumping in the middle of opcodes is allowed by Distorm3

=> Resilience to malware evolution

Sample Script

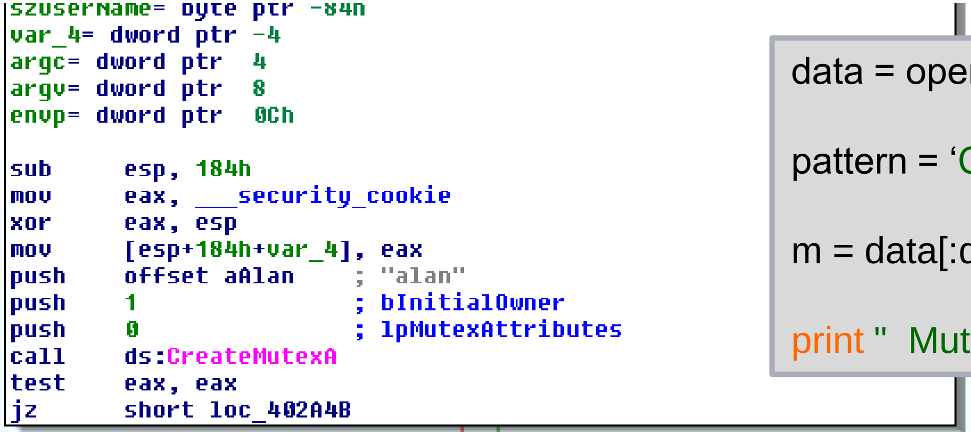
We want to get the Mutex value for this malware :



```
szUserName= byte ptr -84h
var_4= dword ptr -4
argc= dword ptr  4
argv= dword ptr  8
envp= dword ptr  0Ch

sub     esp, 184h
mov     eax, ___security_cookie
xor     eax, esp
mov     [esp+184h+var_4], eax
push    offset aAlan      ; "alan"
push    1                 ; bInitialOwner
push    0                 ; lpMutexAttributes
call    ds:CreateMutexA
test    eax, eax
jz      short loc_402A4B
```

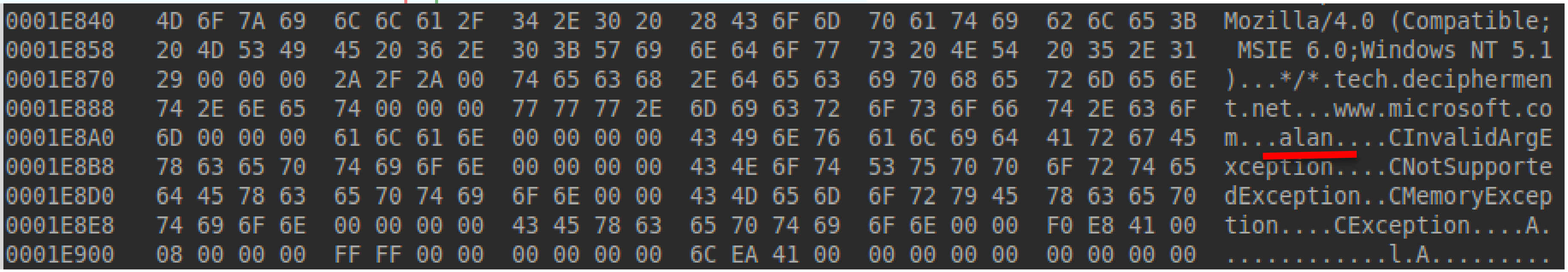
Sample Script

We want to get the Mutex value for this malware : Writing basic python script



```
szUserName= byte ptr -84h
var_4= dword ptr -4
argc= dword ptr  4
argv= dword ptr  8
envp= dword ptr  0Ch

sub     esp, 184h
mov     eax, ___security_cookie
xor     eax, esp
mov     [esp+184h+var_4], eax
push    offset aAlan     ; "alan"
push    1                ; bInitialOwner
push    0                ; lpMutexAttributes
call    ds:CreateMutexA
test    eax, eax
jz      short loc_402A4B
```

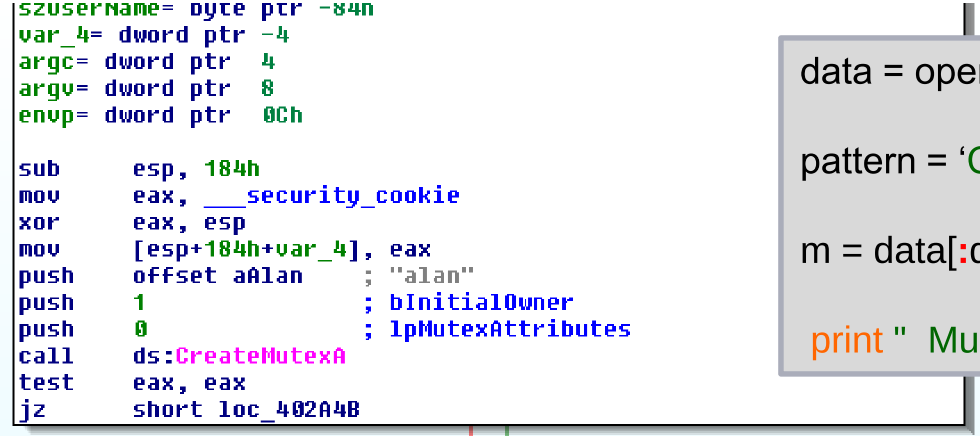
```
data = open(‘/tmp/malware.exe’,’rb’).read()

pattern = ‘CInvalidArgException’

m = data[:data.find(pattern)].rsplit('\x00\x00')

print "  Mutex\t:", m.rsplit('\x00\x00')[-3]
```



```
0001E840   4D 6F 7A 69  6C 6C 61 2F  34 2E 30 20  28 43 6F 6D  70 61 74 69  62 6C 65 3B  Mozilla/4.0 (Compatible;
0001E858   20 4D 53 49  45 20 36 2E  30 3B 57 69  6E 64 6F 77  73 20 4E 54  20 35 2E 31   MSIE 6.0;Windows NT 5.1
0001E870   29 00 00 00  2A 2F 2A 00  74 65 63 68  2E 64 65 63  69 70 68 65  72 6D 65 6E  )...*/*.tech.deciphermen
0001E888   74 2E 6E 65  74 00 00 00  77 77 77 2E  6D 69 63 72  6F 73 6F 66  74 2E 63 6F  t.net...www.microsoft.co
0001E8A0   6D 00 00 00  61 6C 61 6E  00 00 00 00  43 49 6E 76  61 6C 69 64  41 72 67 45  m...alan....CInvalidArgE
0001E8B8   78 63 65 70  74 69 6F 6E  00 00 00 00  43 4E 6F 74  53 75 70 70  6F 72 74 65  xception....CNotSupporte
0001E8D0   64 45 78 63  65 70 74 69  6F 6E 00 00  43 4D 65 6D  6F 72 79 45  78 63 65 70  dException..CMemoryExcep
0001E8E8   74 69 6F 6E  00 00 00 00  43 45 78 63  65 70 74 69  6F 6E 00 00  F0 E8 41 00  tion....CException....A.
0001E900   08 00 00 00  FF FF 00 00  00 00 00 00  6C EA 41 00  00 00 00 00  00 00 00 00  ............l.A.........
```

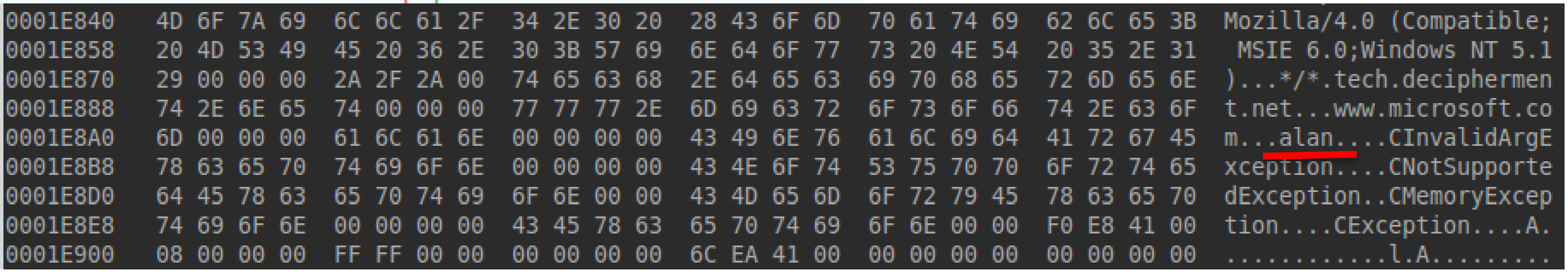
Sample Script

Too much fix value



```
szUserName= byte ptr -84h
var_4= dword ptr -4
argc= dword ptr  4
argv= dword ptr  8
envp= dword ptr  0Ch

sub     esp, 184h
mov     eax, ___security_cookie
xor     eax, esp
mov     [esp+184h+var_4], eax
push    offset aAlan     ; "alan"
push    1                ; bInitialOwner
push    0                ; lpMutexAttributes
call    ds:CreateMutexA
test    eax, eax
jz      short loc_402A4B
```

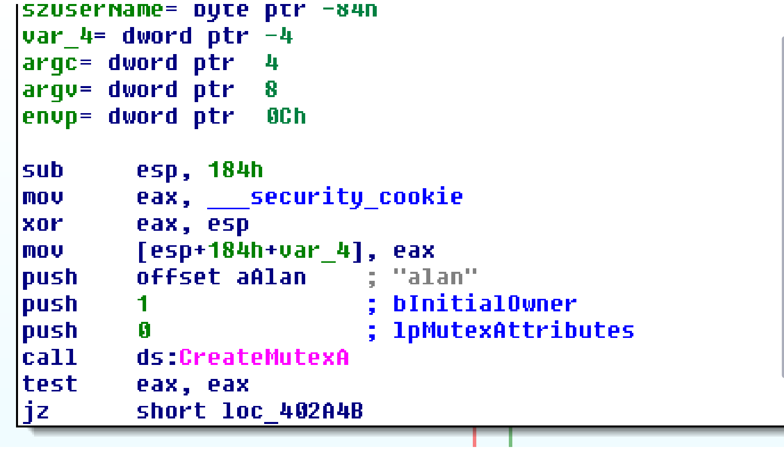
```
data = open('/tmp/malware.exe','rb').read()

pattern = 'CInvalidArgException'

m = data[:data.find(pattern)].rsplit('\x00\x00')

 print "  Mutex\t:", m[-3]
```



```
0001E840   4D 6F 7A 69  6C 6C 61 2F  34 2E 30 20  28 43 6F 6D  70 61 74 69  62 6C 65 3B  Mozilla/4.0 (Compatible;
0001E858   20 4D 53 49  45 20 36 2E  30 3B 57 69  6E 64 6F 77  73 20 4E 54  20 35 2E 31   MSIE 6.0;Windows NT 5.1
0001E870   29 00 00 00  2A 2F 2A 00  74 65 63 68  2E 64 65 63  69 70 68 65  72 6D 65 6E  )...*/*.tech.deciphermen
0001E888   74 2E 6E 65  74 00 00 00  77 77 77 2E  6D 69 63 72  6F 73 6F 66  74 2E 63 6F  t.net...www.microsoft.co
0001E8A0   6D 00 00 00  61 6C 61 6E  00 00 00 00  43 49 6E 76  61 6C 69 64  41 72 67 45  m...alan....CInvalidArgE
0001E8B8   78 63 65 70  74 69 6F 6E  00 00 00 00  43 4E 6F 74  53 75 70 70  6F 72 74 65  xception....CNotSupporte
0001E8D0   64 45 78 63  65 70 74 69  6F 6E 00 00  43 4D 65 6D  6F 72 79 45  78 63 65 70  dException..CMemoryExcep
0001E8E8   74 69 6F 6E  00 00 00 00  43 45 78 63  65 70 74 69  6F 6E 00 00  F0 E8 41 00  tion....CException....A.
0001E900   08 00 00 00  FF FF 00 00  00 00 00 00  6C EA 41 00  00 00 00 00  00 00 00 00  ............l.A.........
```

Sample Script

We want to get the Mutex value for this malware :



```
szUserName= byte ptr -84h
var_4= dword ptr -4
argc= dword ptr  4
argv= dword ptr  8
envp= dword ptr  0Ch

sub     esp, 184h
mov     eax, ___security_cookie
xor     eax, esp
mov     [esp+184h+var_4], eax
push    offset aAlan    ; "alan"
push    1               ; bInitialOwner
push    0               ; lpMutexAttributes
call    ds:CreateMutexA
test    eax, eax
jz      short loc_402A4B
```

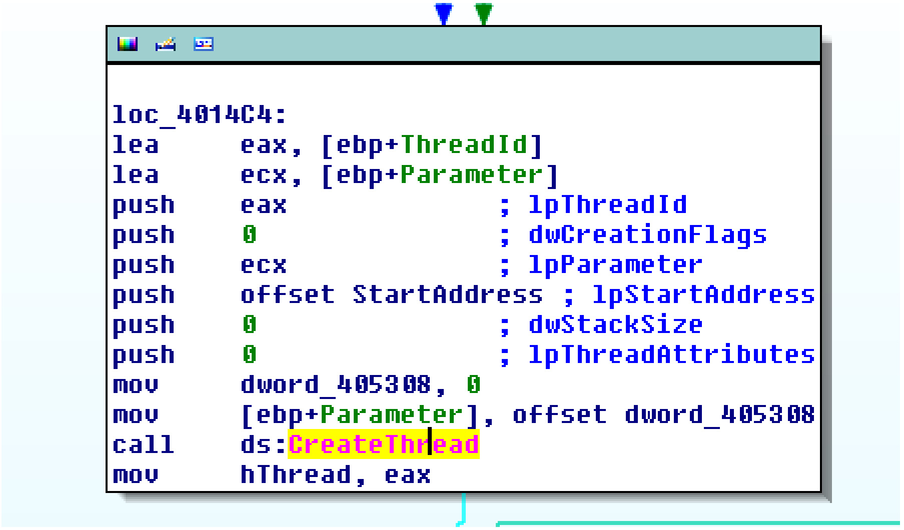
```
data = open('/tmp/malware.exe','rb').read()

pattern = 'CInvalidArgException'

m = data[:data.find(pattern)].rsplit('\x00\x00')

 print "  Mutex\t:", m[-3]
```

```
disass = Disass32(path='/tmp/malware.exe', verbose=False)

if disass.go_to_next_call('CreateMutexA'):

        address_mutex = disass.get_arguments(3, convention=STDCALL)

        print "  Mutex\t:", disass.get_string(address_mutex)
```

CASSIDIAN CYBERSECURITY

Sample Script

We want to jump in Thread

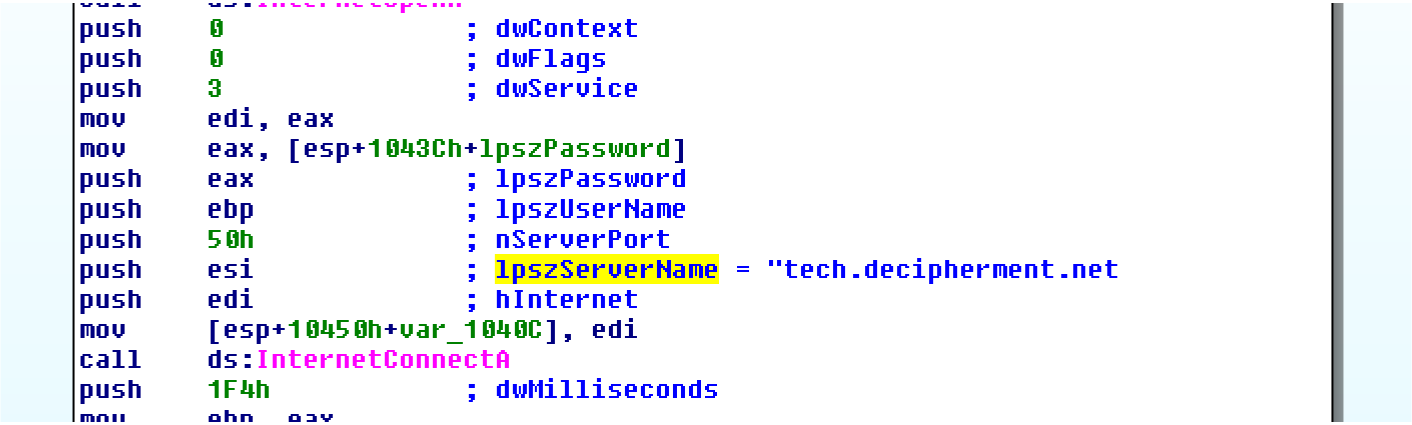


```
loc_4014C4:
lea     eax, [ebp+ThreadId]
lea     ecx, [ebp+Parameter]
push    eax                 ; lpThreadId
push    0                   ; dwCreationFlags
push    ecx                 ; lpParameter
push    offset StartAddress ; lpStartAddress
push    0                   ; dwStackSize
push    0                   ; lpThreadAttributes
mov     dword_405308, 0
mov     [ebp+Parameter], offset dword_405308
call    ds:CreateThread
mov     hThread, eax
```

```
disass = Disass32(path=‘/tmp/malware.exe’, verbose=False)

if disass.go_to_next_call('CreateThread'):

        startAddress = disass.get_arguments(3)

        disass.set_virtual_position(startAddress)
```

Sample Script

We want to get the C&C :



```
call    ds:InternetOpenA
push    0               ; dwContext
push    0               ; dwFlags
push    3               ; dwService
mov     edi, eax
mov     eax, [esp+1043Ch+lpszPassword]
push    eax             ; lpszPassword
push    ebp             ; lpszUserName
push    50h             ; nServerPort
push    esi             ; lpszServerName = "tech.decipherment.net
push    edi             ; hInternet
mov     [esp+10450h+var_1040C], edi
call    ds:InternetConnectA
push    1F4h            ; dwMilliseconds
mov     ebp, eax
```

```
if disass.go_to_next_call(‘InternetConnectA'):

        print "  CC1\t:", disass.get_string(disass.get_arguments(2))
```

CASSIDIAN CYBERSECURITY

Demo

Disass scripts vs. malware evolution

- **Set a different C&C**
- **Bugs fix in malware**
- **New features in malware**
- **Packing**
- **Encryption**

Disass is available

Disass is available here (Alpha release):

Disass support PE32 on x86

http://bitbucket.cassidiancybersecurity.com/disass

Questions ?

Ivan.Fontarensky@cassidian.com

@ifontarensky

Disass : End

Distorm support Intel x86 8bit 16bit 32bit and 64bit

Elfesteem to manage ELF format