

OLLYDBG

MATTIA PASTORELLI

COMANDA:

Fate riferimento al malware: Malware_U3_W3_L3, presente all'interno della cartella Esercizio_Pratico_U3_W3_L3 sul desktop della macchina virtuale dedicata all'analisi dei malware. Rispondete ai seguenti quesiti utilizzando OllyDBG:

- All'indirizzo 0040106E il Malware effettua una chiamata di funzione alla funzione «CreateProcess». Qual è il valore del parametro «CommandLine» che viene passato sullo stack?
 - (1) • Inserite un breakpoint software all'indirizzo 004015A3. Qual è il valore del registro EDX? (
 - (2) Eseguite a questo punto uno «step-into». Indicate qual è ora il valore del registro EDX
 - (3) motivando la risposta
 - (4). Che istruzione è stata eseguita?
- Inserite un secondo breakpoint all'indirizzo di memoria 004015AF. Qual è il valore del registro ECX?
 - (6) Eseguite un step-into. Qual è ora il valore di ECX?
 - (7) Spiegate quale istruzione è stata eseguita
 - (8). • BONUS: spiegare a grandi linee il funzionamento del malware



ALL'INDIRIZZO 0040106E IL MALWARE EFFETTUÀ UNA CHIAMATA DI FUNZIONE ALLA FUNZIONE «CREATEPROCESS». QUAL È IL VALORE DEL PARAMETRO «COMMANDLINE» CHE VIENE PASSATO SULLO STACK?

Per prima cosa importiamo il malware all'interno del programma, attraverso lo strumento “GO TO”, possiamo dirigerci direttamente all'indirizzo richiesto.

Una volta trovato l'indirizzo, effettuiamo un comando di breakdown e avviamo il malware.

Il risultato sarà riportato nella sezione a destra del Register

Il valore del parametro “CommandLine” è “CMD”

00401063	6A 00	PUSH 0		
00401065	6A 00	PUSH 0		
00401067	68 30504000	PUSH Malware_.00405030		
0040106C	6A 00	PUSH 0		
0040106E	FF15 04404000	CALL DWORD PTR DS:[&KERNEL32.CreateProcessA]	ASCII "cmd"	kernel32.CreateProcessA
00401074	8945 EC	MOV DWORD PTR SS:[EBP-14],EAX		



```
pProcessInfo
pStartupInfo
CurrentDir = NULL
pEnvironment = NULL
CreationFlags = 0
InheritHandles = TRUE
pThreadSecurity = NULL
pProcessSecurity = NULL
CommandLine = "cmd"
ModuleFileName = NULL
CreateProcessA
Timeout = INFINITE
```

INSERITE UN BREAKPOINT SOFTWARE ALL'INDIRIZZO 004015A3. QUAL È IL VALORE DEL REGISTRO EDX?

The screenshot shows a debugger interface with two main panes. The left pane displays assembly code for a function named 'SE handler installation'. The right pane shows the 'Registers (FPU)' window. The assembly code includes instructions like PUSH EBP, MOV EBP,ESP, and various pushes and moves to the stack. The registers window shows the following values:

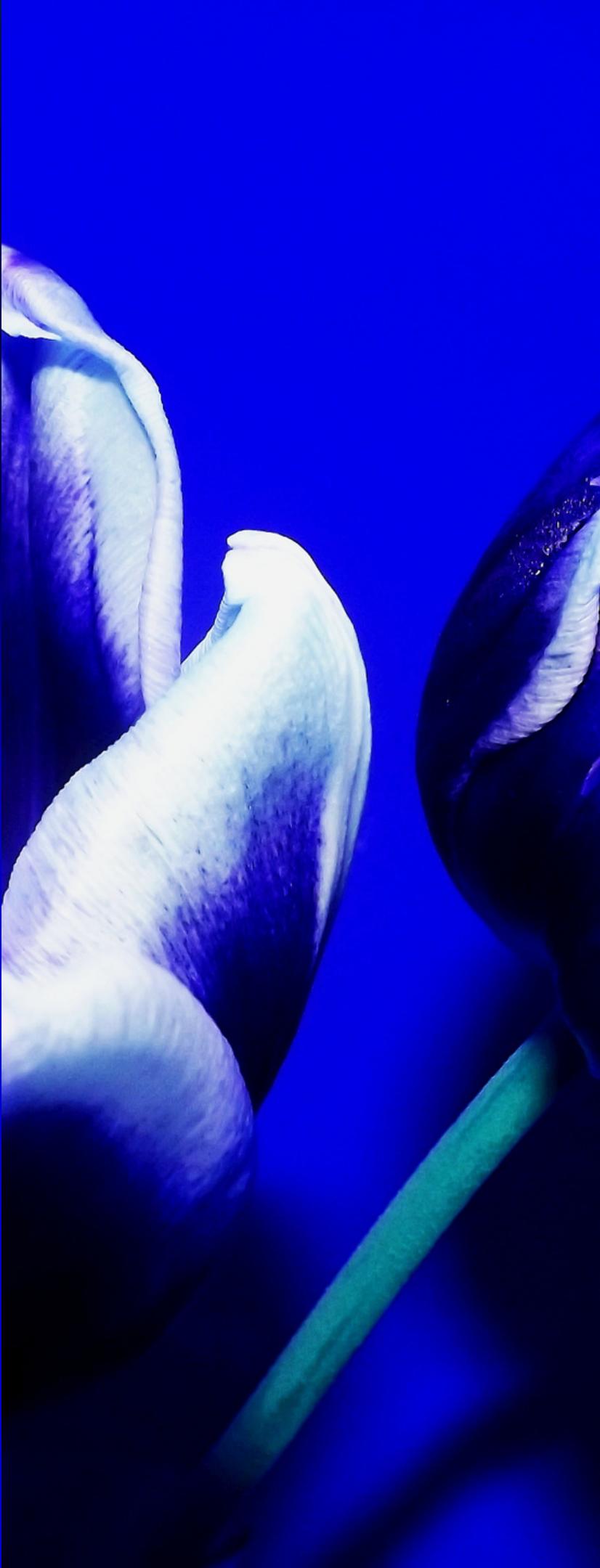
Register	Value
EAX	1DB10106
ECX	7EFDE000
EDX	00001DB1
EBX	7EFDE000
ESP	0018FF5C
EBP	0018FF88
ESI	00000000
EDI	00000000

The EIP register is set to 004015A3, which is highlighted in red. The CPU status register (EFL) shows a value of 00000206. Below the registers, there is a summary of the processor state.

Stesso procedimento del precedente, questa volta viene richiesto il valore di EDX:

Il valore di EDX è 00001DB1

ESEGUITE A QUESTO PUNTO UNO «STEP-INTO». INDICATE
QUAL È ORA IL VALORE DEL REGISTRO EDX.
MOTIVANDO LA RISPOSTA, CHE ISTRUZIONE È STATA
ESEGUITA?



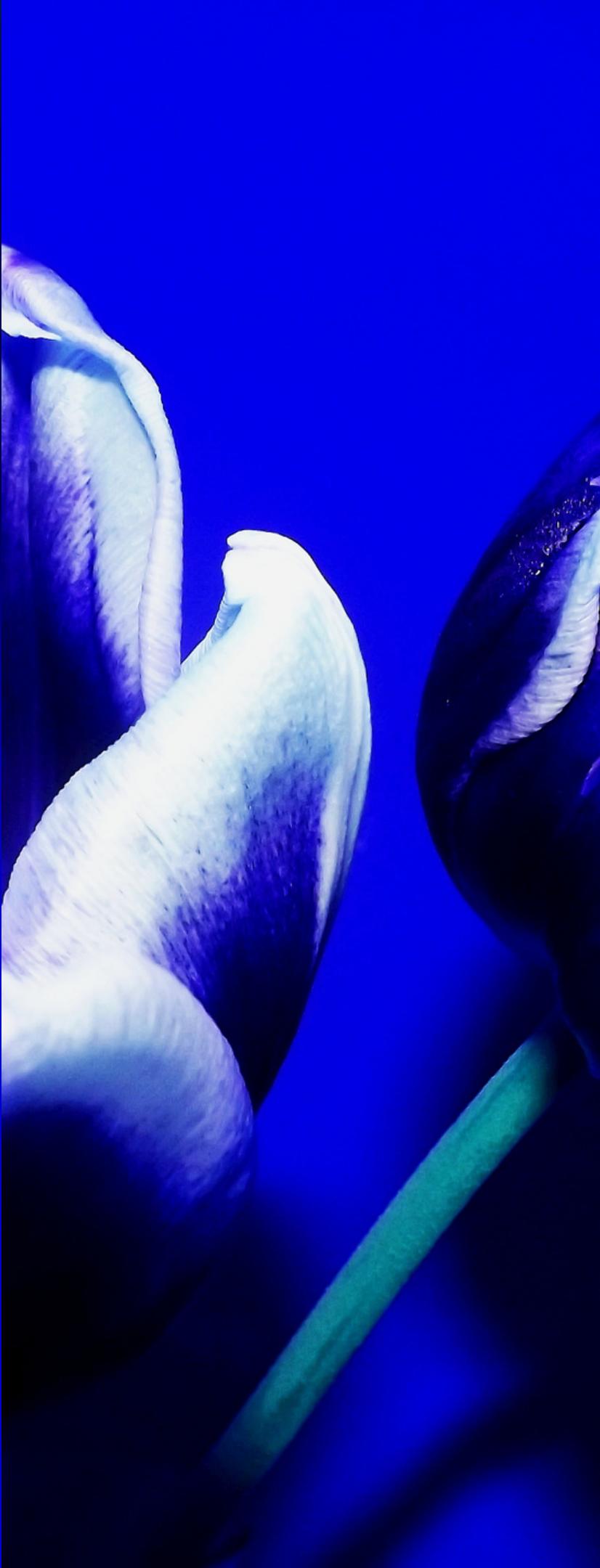
Screenshot of a debugger showing assembly code and registers. The assembly window shows the following code:

```
00401577 $ 55 PUSH EBP
00401578 . 8BEC MOV EBP,ESP
00401579 . 6A FF PUSH -1
0040157C . 68 C0404000 PUSH Malware_.004040C0
00401581 . 68 3C204000 PUSH Malware_.0040203C
00401586 . 64:A1 00000000 MOV EAX,DWORD PTR FS:[0]
0040158C . 50 PUSH EAX
0040158D . 64:8925 000000 MOV DWORD PTR FS:[0],ESP
00401594 . 83EC 10 SUB ESP,10
00401597 . 53 PUSH EBX
00401598 . 56 PUSH ESI
00401599 . 57 PUSH EDI
0040159A . 8965 E8 MOV DWORD PTR SS:[EBP-18],ESP
0040159D . FF15 30404000 CALL DWORD PTR DS:[<&KERNEL32.GetVersion kernel32.GetVersion
004015A3 . 33D2 XOR EDX,EDX
004015A5 . 8A04 MOV DL,AH
004015A7 . 8915 D4524000 MOV DWORD PTR DS:[4052D4],EDX
004015AD . 8BC8 MOV ECX,EAX
004015AF . 81E1 FF000000 AND ECX,0FF
004015B5 . 8900 D0524000 MOV DWORD PTR DS:[4052D0],ECX
004015BB . C1E1 08 SHL ECX,8
```

The Registers window shows the following state:

Register	Value
ERX	10B10106
ECX	7EFDE000
EDX	00001DB1
EBX	7EFDE000
ESP	0018FF5C
EBP	0018FF88
ESI	00000000
EDI	00000000
EIP	004015A3 Malware_.004015A3
C 0	ES 002B 32bit 0(FFFFFFFF)
P 1	CS 0023 32bit 0(FFFFFFFF)
A 0	SS 002B 32bit 0(FFFFFFFF)
S 0	DS 002B 32bit 0(FFFFFFFF)
T 0	FS 0053 32bit 7EFDD000(FFF)
D 0	GS 002B 32bit 0(FFFFFFFF)
O 0	LastErr ERROR_SUCCESS (00000000)
EFL	00000206 (NO,NB,NE,A,NS,PE,GE,G)

Malware avviato su breakpoint impostato a : 004015A3 Valore di EDX = 00001DB1 prima dello step-into



Screenshot of a debugger showing assembly code and registers. The assembly window shows the same code as the previous screenshot, but the Registers window shows a different state after the step-into:

Register	Value
ERX	10B10106
ECX	7EFDE000
EDX	00000000
EBX	7EFDE000
ESP	0018FF5C
EBP	0018FF88
ESI	00000000
EDI	00000000
EIP	004015A5 Malware_.004015A5
C 0	ES 002B 32bit 0(FFFFFFFF)
P 1	CS 0023 32bit 0(FFFFFFFF)
A 0	SS 002B 32bit 0(FFFFFFFF)
S 1	DS 002B 32bit 0(FFFFFFFF)
T 0	FS 0053 32bit 7EFDD000(FFF)
D 0	GS 002B 32bit 0(FFFFFFFF)

Dopo aver eseguito lo step-into, il risultato di EDX è 0, perchè l'istruzione XOR EDX,EDX equivale ad inizializzare a zero una variabile.

INSERITE UN SECONDO BREAKPOINT ALL'INDIRIZZO DI MEMORIA 004015AF. QUAL È IL VALORE DEL REGISTRO ECX?

(6) ESEGUITE UN STEP-INTO. QUAL È ORA IL VALORE DI ECX?

(7) SPIEGATE QUALE ISTRUZIONE È STATA ESEGUITA

The screenshot shows the assembly and registers windows of a debugger. The assembly window displays the following code snippet:

```
00401577: 55          PUSH EBP
00401578: 8BEC        MOV EBP,ESP
0040157A: 6A FF        PUSH -1
0040157C: 68 C0404000  PUSH Malware_.004040C0
00401581: 68 3C204000  PUSH Malware_.0040203C
00401586: 64:A1 00000000 MOV EAX, DWORD PTR FS:[0]
0040158C: 50          PUSH EAX
0040158D: 64:8925 000000 MOV DWORD PTR FS:[0],ESP
00401594: 83EC 10      SUB ESP,10
00401597: 53          PUSH EBX
00401598: 56          PUSH ESI
00401599: 57          PUSH EDI
0040159A: 8965 E8      MOV DWORD PTR SS:[EBP-18],ESP
0040159D: FF15 30404000 CALL DWORD PTR DS:[<&KERNEL32.GetVersion kernel32.GetVersion
004015A3: 33D2          XOR EDX,EDX
004015A5: 8A04          MOV DL,AH
004015A7: 8915 D4524000 MOV DWORD PTR DS:[4052D4],EDX
004015AD: 8BC8          MOV ECX,EAX
004015AF: 81E1 FF000000 AND ECX,0FF
004015B5: 890D D0524000 MOV DWORD PTR DS:[4052D0],ECX
004015BB: C1E1 08        SHL ECX,8
004015BE: 03CA          ADD ECX,EDX
004015C0: 890D CC524000 MOV DWORD PTR DS:[4052CC],ECX
```

The registers window shows the following values:

Register	Value
EAX	1DB10106
ECX	7EFDE000
EDX	00000000
EBX	7EFDE000
ESP	0018FF5C
EIP	004015A3
ESI	00000000
EDI	00000000
C 0	ES 002B 32bit 0(FFFFFFFF)
P 1	CS 0023 32bit 0(FFFFFFFF)
A 0	SS 002B 32bit 0(FFFFFFFF)
Z 0	DS 002B 32bit 0(FFFFFFFF)
S 0	FS 0053 32bit 7EFDD000(FFF)
T 0	GS 002B 32bit 0(FFFFFFFF)
D 0	LastErr ERROR_SUCCESS (00000000)
EFL	00000206 (NO,NB,NE,A,NS,PE,GE,G)
ST0	empty 0.0
ST1	empty 0.0

Il valore di ECX è 7EFDE000 prima del step-into

E' stata eseguita la funzione And Ecx, OFF

The screenshot shows the assembly and registers windows of a debugger. The assembly window displays the following code snippet:

```
00401581: 68 3C204000  PUSH Malware_.0040203C
00401586: 64:A1 00000000 MOV EAX, DWORD PTR FS:[0]
0040158C: 50          PUSH EAX
0040158D: 64:8925 000000 MOV DWORD PTR FS:[0],ESP
00401594: 83EC 10      SUB ESP,10
00401597: 53          PUSH EBX
00401598: 56          PUSH ESI
00401599: 57          PUSH EDI
0040159A: 8965 E8      MOV DWORD PTR SS:[EBP-18],ESP
0040159D: FF15 30404000 CALL DWORD PTR DS:[<&KERNEL32.GetVersion kernel32.GetVersion
004015A3: 33D2          XOR EDX,EDX
004015A5: 8A04          MOV DL,AH
004015A7: 8915 D4524000 MOV DWORD PTR DS:[4052D4],EDX
004015AD: 8BC8          MOV ECX,EAX
004015AF: 81E1 FF000000 AND ECX,0FF
004015B5: 890D D0524000 MOV DWORD PTR DS:[4052D0],ECX
004015BB: C1E1 08        SHL ECX,8
004015BE: 03CA          ADD ECX,EDX
004015C0: 890D CC524000 MOV DWORD PTR DS:[4052CC],ECX
004015C6: C1E8 10        SHR EAX,10
004015C9: A3 C8524000  MOV DWORD PTR DS:[4052C8],EAX
004015CE: 6A 00          PUSH 0
004015D0: E8 33090000  CALL Malware_.00401F08
004015D5: 59          POP ECX
004015D6: 85C0          TEST EAX,EAX
004015D8: 75 08          JNZ SHORT Malware_.004015E2
004015E0: C9              RET
```

The registers window shows the following values:

Register	Value
EAX	1DB10106
ECX	00000006
EDX	00000001
EBX	7EFDE000
ESP	0018FF5C
EIP	004015B5
ESI	00000000
EDI	00000000
C 0	ES 002B 32bit 0(FFFFFFFF)
P 1	CS 0023 32bit 0(FFFFFFFF)
A 0	SS 002B 32bit 0(FFFFFFFF)
Z 0	DS 002B 32bit 0(FFFFFFFF)
S 0	FS 0053 32bit 7EFDD000(FFF)
T 0	GS 002B 32bit 0(FFFFFFFF)
D 0	LastErr ERROR_SUCCESS (00000000)
EFL	00000206 (NO,NB,NE,A,NS,PE,GE,G)
ST0	empty 0.0
ST1	empty 0.0
ST2	empty 0.0
ST3	empty 0.0
ST4	empty 0.0

Il valore di ECX è 00000006 dopo lo step-into

BONUS:

Il seguente malware è un trojan, utilizzato per creare una backdoor sul computer della vittima.

Analizzando l'assembly troviamo elementi che ci fanno comprendere che il malware cerca di ottenere l'indirizzo IP della vittima attraverso il comando “GetHostName” e “Recv”.

Attraverso il comando “BackdoorServer” il malware crea un server in ascolto su una determinata porta.

Popular threat label	trojan.idicaf/r06cc0df321	Threat categories	trojan	Family labels	idicaf r06cc0df321
Security vendors' analysis		Do you want to automate checks?			
AhnLab-V3	Backdoor/Win32.Agent.R9408	Alibaba		Backdoor:Win32/Idicaf.9f3a5556	
ALYac	Backdoor.XIW	Antiy-AVL		Trojan[Backdoor]/Win32.Agent	
Arcabit	Backdoor.XIW	Avast		Win32:Agent-OLH [Trj]	
AVG	Win32:Agent-OLH [Trj]	Avira (no cloud)		BDS/Agen.twe.134160	
BitDefender	Backdoor.XIW	ClamAV		Win.Trojan.Idicaf-9937585-0	
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cylance		Unsafe	