

Malware Analysis Report

Remote Access Trojan-Reverse Shell

Aug 2022 | Jan Duinkerken | v1.1

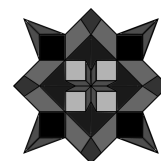
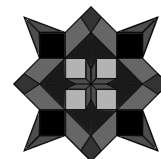


Table of Contents

Table of Contents	2
Executive Summary	3
High-Level Technical Summary	4
Basic Static Analysis	5
Basic Dynamic Analysis.....	6
Advanced Static Analysis.....	7
Advanced Dynamic Analysis	8
Indicators of Compromise	9
Network Indicators.....	9
Host-based Indicators.....	10
Rules & Signatures	11
Appendices	12
A. Yara Rules	12
B. DNS Record Queries.....	12
C. Decompiled Code Snippets	13

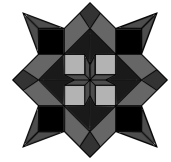


Executive Summary

SHA256 hash	481eae82ac4cd1a9cfadc026a628b18d7b4c54f50385d28c505fbc3e999b8b0
----------------	---

This Trojan's main functionality is creating a Reverse shell listening on open on port 443 with the purpose of letting the attacker execute code remotely and have complete access to our machines

YARA signature rules are attached in Appendix A. Malware sample and hashes have been submitted to VirusTotal for further examination.



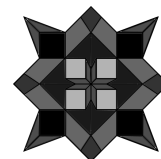
High-Level Technical Summary

This is a really basic trojan; it consists of just one file that launches an A record DNS query to aaaaaaaaaaaaaaaaaaaaaa.kadusus.local and if it gets a response it then serves a shell using port 443

RAT.ReverseShell.exe

A Record DNS:
aaaaa...kadusus.local

Serve Reverse
Shell on port 443



Basic Static Analysis

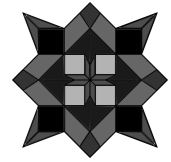
After running floss against the RAT sample we only got one interesting result:

```
C:\Users\Jan\Desktop  
λ floss RAT.Unknown2.exe | grep cmd  
@cmd.exe /c
```

Fig 1: FLOSS Command Example Output

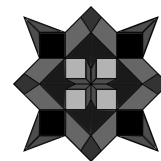
This tells us the Trojan is calling the cmd, but we still don't know what commands it will run or any more concrete information on how it deploys the shell and when it does

We also analyzed the binary using PEView and PESTudio and didn't find anything incriminating, but it is interesting that this is a 64 bit executable.



Basic Dynamic Analysis

At first glance the Trojan doesn't seem to do anything, it doesn't spawn a shell or create any files. It is only if the domain `aaaa...kadusus.local` is reachable that it opens a TCP connection on port 443 and serves the reverse shell.



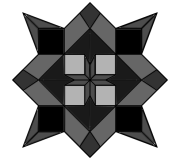
Advanced Static Analysis

Running the binary through a decompiler confirms our suspicions that this sample is written in Nim, due to the fact that the main function is divided in 4 stages: main, WinMain, NimMain and NimMainInner.

```
dbg.main
dbg.mainCRTStartup
sym.NimMain
sym.NimMainInner
sym.NimMainModule
sym.PreMain
sym.PreMainInner
sym.WinMain
```

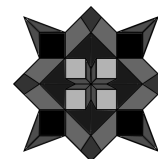
Fig 2: Main Methods as Shown by Cutter

Some interesting parts of the assembly code can be found in [Appendix C](#).



Advanced Dynamic Analysis

By running the executable through a debugger (x64dbg), we were not able to find any more important information. So, we can conclude that the only functionality of this trojan is the one we discovered during the Basic Dynamic Analysis section of this report



Indicators of Compromise

The full list of IOCs can be found in the Appendices.

Network Indicators

4	2.0322011/9	10.0.0.3	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1.1
5	4.086557307	10.0.0.3	239.255.255.250	SSDP	179 M-SEARCH * HTTP/1.1
6	7.081421519	10.0.0.3	239.255.255.250	SSDP	179 M-SEARCH * HTTP/1.1
7	10.098188836	10.0.0.3	239.255.255.250	SSDP	179 M-SEARCH * HTTP/1.1
8	14.897858639	10.0.0.3	10.0.0.4	DNS	94 Standard query 0x21bf A aaaaaaaaaaaaaaaaaa.kadusus.local
9	14.923186025	10.0.0.3	10.0.0.4	DNS	94 Standard query 0x21bf A aaaaaaaaaaaaaaaaaa.kadusus.local
10	15.843976619	10.0.0.3	10.0.0.3	DNS	110 Standard query response 0x21bf A aaaaaaaaaaaaaaaaaa.kadusus.local A 10.0.0.4
11	15.847311591	10.0.0.3	10.0.0.4	TCP	66 14412 -> 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
12	15.847150773	10.0.0.4	10.0.0.3	TCP	66 443 -> 14412 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
13	15.847469862	10.0.0.3	10.0.0.4	TCP	60 14412 -> 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
14	15.855539562	10.0.0.4	10.0.0.3	DNS	110 Standard query response 0x21bf A aaaaaaaaaaaaaaaaaa.kadusus.local A 10.0.0.4
15	20.867618922	PcsCompu_51:30:a7	10.0.0.4	ARP	42 Who has 10.0.0.3? Tell 10.0.0.4
16	20.867968890	PcsCompu_24:b2:e7	PcsCompu_51:30:a7	ARP	60 10.0.0.3 is at 08:00:27:24:b2:e7
17	42.692583198	10.0.0.3	239.255.255.250	SSDP	179 M-SEARCH * HTTP/1.1
18	45.699557397	10.0.0.3	239.255.255.250	SSDP	179 M-SEARCH * HTTP/1.1
19	48.719701193	10.0.0.3	239.255.255.250	SSDP	179 M-SEARCH * HTTP/1.1

Frame 8: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_24:b2:e7 (08:00:27:24:b2:e7), Dst: PcsCompu_51:30:a7 (08:00:27:51:30:a7)
Internet Protocol Version 4, Src: 10.0.0.3, Dst: 10.0.0.4
User Datagram Protocol, Src Port: 57466, Dst Port: 53
Domain Name System (query)
Transaction ID: 0x21bf
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
Name: aaaaaaaaaaaaaaaaaa.kadusus.local type A, class IN
Name: aaaaaaaaaaaaaaaaaa.kadusus.local
[Name Length: 34]
[Label Count: 3]
Type: A (Host Address) (1)
Class: IN (0x0001)

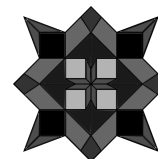
Fig 3: WireShark Packet Capture of DNS Query

Time ...	Process Name	PID	Operation	Path	Result	Detail
7:16:5...	RAT.Unknown...	3908	TCP Reconnect	aaaaaaaaaaaaaaaaaaaa.kadusus.local:14417 -> aaaaaaaaaaaaaaaaaaaaa.kadusus.local/https	SUCCESS	Length: 0, seqnum:...
7:16:5...	RAT.Unknown...	3908	TCP Reconnect	aaaaaaaaaaaaaaaaaaaa.kadusus.local:14417 -> aaaaaaaaaaaaaaaaaaaaa.kadusus.local/https	SUCCESS	Length: 0, seqnum:...
7:16:5...	RAT.Unknown...	3908	TCP Reconnect	aaaaaaaaaaaaaaaaaaaa.kadusus.local:14417 -> aaaaaaaaaaaaaaaaaaaaa.kadusus.local/https	SUCCESS	Length: 0, seqnum:...
7:16:5...	RAT.Unknown...	3908	TCP Reconnect	aaaaaaaaaaaaaaaaaaaa.kadusus.local:14417 -> aaaaaaaaaaaaaaaaaaaaa.kadusus.local/https	SUCCESS	Length: 0, seqnum:...
7:16:5...	RAT.Unknown...	3908	TCP Disconnect	aaaaaaaaaaaaaaaaaaaa.kadusus.local:14417 -> aaaaaaaaaaaaaaaaaaaaa.kadusus.local/https	SUCCESS	Length: 0, seqnum:...

Fig 4: Trojan Serving a Reverse Shell on Port 443.

```
C:\Users\Jan
λ ncat -nvlp 443
Ncat: Version 5.59BETA1 ( http://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 127.0.0.1:14425.
whoami
desktop-b01ntda\jan
```

Fig 5: Reverse Shell

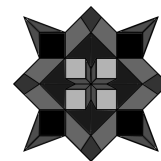


Host-based Indicators

[-] RAT.Unknown2.exe (2960)	C:\Users\Jan\De...		DESKTOP-B01N...	"C:\Users\Jan\D...	8/20/2022 7:20:1...	n/a
[-] cmd.exe (3920)	Windows Comma...	C:\Windows\SYS...	Microsoft Corporat...	DESKTOP-B01N...	cmd.exe /c id	8/20/2022 7:23:3...
[-] Conhost.exe (380)	Console Window ...	C:\Windows\Syst...	Microsoft Corporat...	DESKTOP-B01N...	\\??C:\Windows\...	8/20/2022 7:23:3...
[-] id.exe (4136)	ShimGen generat...	C:\ProgramData\...	Chocolatey Softw...	DESKTOP-B01N...	id	8/20/2022 7:23:3...
[-] id.exe (5108)		C:\ProgramData\...		DESKTOP-B01N...	"C:\ProgramData\...	8/20/2022 7:23:3...
[-] Idle (0)	Idle					8/18/2022 5:07:0...
[-] System (4)	System			NT AUTHORITY\...		8/18/2022 5:07:1...
[-] MemCompression (1188)	MemCompression			NT AUTHORITY\...		8/18/2022 8:08:5...
[-] Registry (72)	Registry			NT AUTHORITY\...		8/18/2022 5:07:0...
[-] smss.exe (328)	Windows Session ...	C:\Windows\Syst...	Microsoft Corporat...	NT AUTHORITY\...	\SystemRoot\Syst...	8/18/2022 5:07:1...
[-] csrss.exe (424)	Client Server Runt...	C:\Windows\syst...	Microsoft Corporat...	NT AUTHORITY\...	%SystemRoot%\s...	8/18/2022 5:08:1...
	Client Server Runt...	C:\Windows\syst...	Microsoft Corporat...	NT AUTHORITY\...	%SystemRoot%\s...	8/18/2022 5:08:1...

Description: Windows Command Processor
Company: Microsoft Corporation
Path: C:\Windows\SYSTEM32\cmd.exe
Command: cmd.exe /c id

Fig 6: RAT Process Spawning a Child CMD Process to Run the Remote Commands

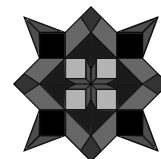


Rules & Signatures

A full set of YARA rules is included in Appendix A.

```
C:\Users\Jan\Desktop
λ yara32 rule.yara RAT.Unknown2.exe.malz -s -w -p 32
RAT_ReverseShell RAT.Unknown2.exe.malz
0xfa00:$string1: kadusus
0xf910:$string2: cmd
0x136f4:$string2: cmd
0x20c85:$string2: cmd
0x4ff72:$string2: cmd
0x4ffd1:$string2: cmd
0x4fff3:$string2: cmd
0x50018:$string2: cmd
0x50036:$string2: cmd
0x55aec:$string2: cmd
0x56806:$string2: cmd
0x635c6:$string2: cmd
0x6562a:$string2: cmd
0x670ac:$string2: cmd
0x6bd9e:$string2: cmd
0x6bdbb:$string2: cmd
0x6c48f:$string2: cmd
0x6df48:$string2: cmd
0x6df75:$string2: cmd
0x0:$PE_magic_byte: MZ
```

Fig 7: YARA Rules Firing



Appendices

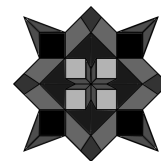
A. Yara Rules

Full Yara repository located at: <http://github.com/HuskyHacks/PMAT-lab>

```
rule RAT_ReverseShell {  
  
  meta:  
    last_updated = "2022-08-23"  
    author = "Jan Duinkerken"  
    description = "Rule for identifying the kadusus trojan"  
  
  strings:  
    $string1 = "kadusus" ascii  
    $string2 = "cmd"  
    $PE_magic_byte = "MZ"  
  
  condition:  
    $PE_magic_byte at 0 and  
    $string1 and  
    $string2  
}
```

B. DNS Record Queries

Domain
aaaaaaaaaaaaaaaaaaaaa.kadusus.local

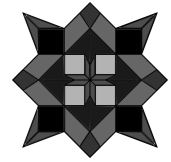


C. Decompiled Code Snippets

```
[0x0040ce36]
mov     rcx, r9
lea     rdx, [0x00411940]
call    appendString.part.0      ; sym.appendString.part.0_4
lea     rcx, [0x0042b7d0]
mov     rdx, r9
call    asgnRef                  ; sym.asgnRef_3
mov     r9d, 1
mov     r8d, 6
mov     edx, 1
mov     word [0x0042b810], 0x1bb
mov     ecx, 2
call    newSocket___Jq0qsT9cdAR4d7YGWwa2QIA ; sym.newSocket___Jq0qsT9cdAR4d7YGWwa2QIA
lea     rcx, [0x0042b7e0]
mov     rdx, rax
call    asgnRef                  ; sym.asgnRef_3
mov     rax, qword [0x00412310]
lea     rcx, [env]               ; jmpbuf env
mov     byte [var_20h], 0
mov     rdx, qword [rax]
mov     qword [var_130h], rdx
lea     rdx, [var_130h]
mov     qword [rax], rdx
mov     rdx, rbp
call    _setjmp                  ; sym._setjmp ; int setjmp(jmpbuf env)
cdqe
mov     qword [var_128h], rax
test    rax, rax
jne     0x40cec8
```

```
[0x0040cec3]
call    command___M5FdZVu9bbC9c8rPhMsT9axZA ; sym.command___M5FdZVu9bbC9c8rPhMsT9axZA
```

Fig 8: Create New Socket and Execute Command



```
[0x0040ccb2]
mov     rcx, qword [rbx]
mov     edx, 1
mov     qword [0x0042b7f0], rsi
inc     rsi
call    resizeString          ; sym.resizeString
mov     rcx, rbx
mov     rdx, rax
call    asgnRef               ; sym.asgnRef_3
mov     rcx, qword [rbx]
mov     rdx, rdi
call    appendString.part.0   ; sym.appendString.part.0_4
cmp     rsi, 0x14             ; 20
jne     0x40ccb2
```

Fig 9: Creating the DNS Record Query Address at Runtime