Practical Malware Analysis & Triage
Malware Analysis Report

Malware.Cryptlib64.dll

Mar 2024 | Teo Heng Shi | v1.0

Table of Contents

Executive Summary

SHA256 hash	732f235784cd2a40c82847b4700fb73175221c6ae6c5f7200a3f43f209989387

The malware.cryptlib64.dll utilized embed function to execute its malicious activities.

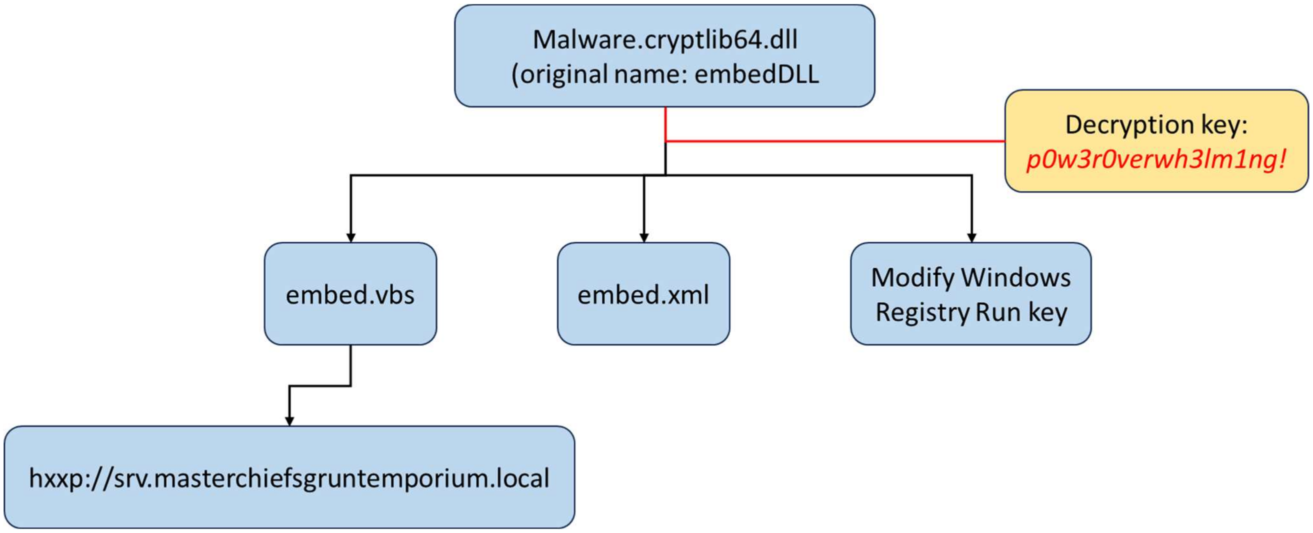
Upon detonated, two (2) files – *embed.xml* and *embed.vbs* have been created and one (1) Windows Registry key modification have been observed. It is noted that the malware persists by saving itself in the registry key, triggering the execution of *embed.vbs* upon user login. Symptoms of infection include beaconing to the URL(s) listed in Appendix B.

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

High-Level Technical Summary

The malware.cryptlib64.dll generates the files – embed.xml and embed.vbs and saved it into C:\Users\Public\ and C:\Users\Public\Documents respectively.

The VBScript code will invoke MSBuild.exe to process C:\Users\Public\embed.xml. The content of embed.xml contains a sizable block of base64-encoded data. Upon triggered, it is observed that embed.vbs will initiate traffic to hxxp://srv.masterchiefsgruntemporium.local



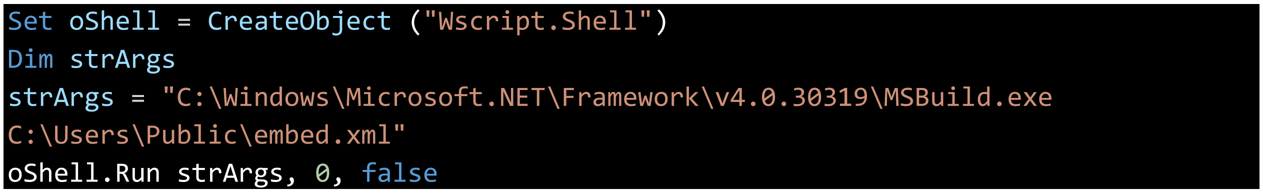
Malware Composition

DemoWare consists of the following components:

File Name	SHA256 Hash
embed.vbs	66fd543f31545082cf8fcc45a6ab1094bc118c45634f2be450f84f4e5745b291
embed.xml	f1548cd02784606c8abac865abf5ed6220d34eea88c7a5715e0183d7f050f4ab

embed.vbs

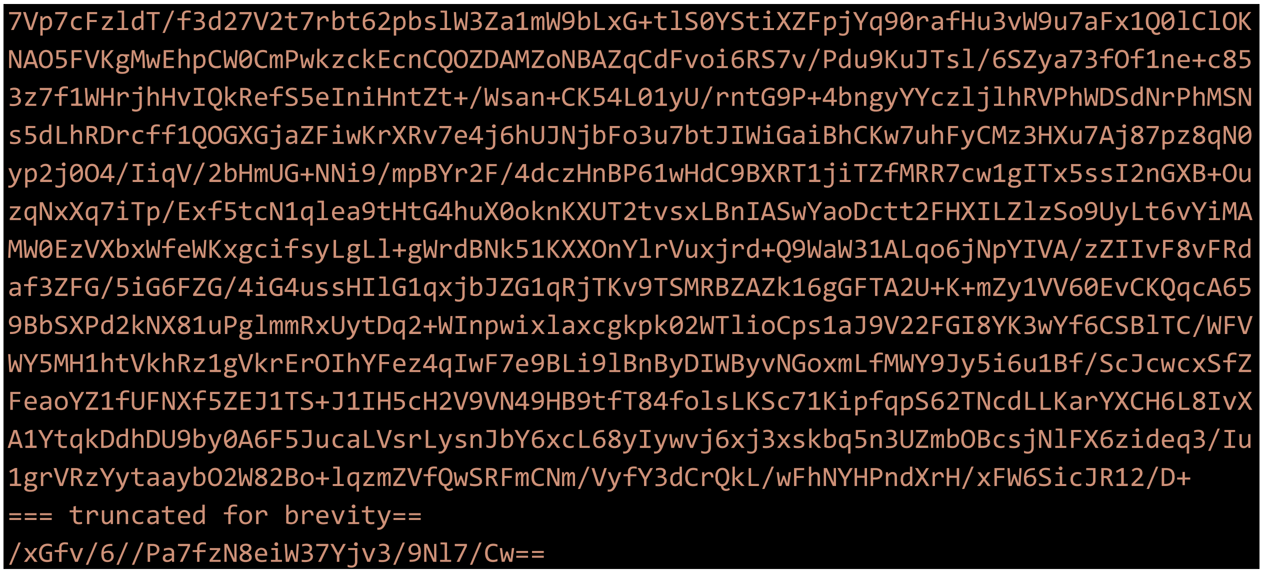
The script will trigger MSBuild.exe under C:\Windows\Microsoft.Net\Framework\v4.0.30319 and to process embed.xml under C:\Users\Public directory



```
Set oShell = CreateObject ("Wscript.Shell")
Dim strArgs
strArgs = "C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe
C:\Users\Public\embed.xml"
oShell.Run strArgs, 0, false
```

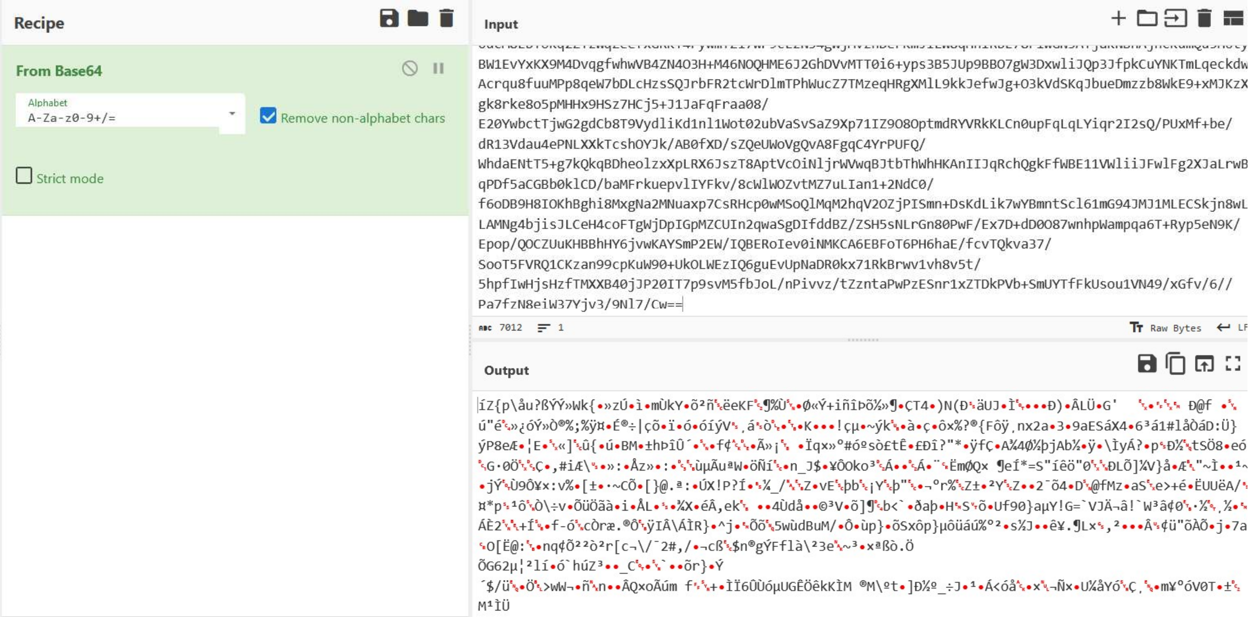
embed.xml:

A Base64 encoded payload was observed to be under the embed.xml file; which could not be translated through Cyberchef



```
7Vp7cFzldT/f3d27V2t7rbt62pbslW3Za1mW9bLxG+tlS0YStiXZFpjYq90rafHu3vW9u7aFx1Q0lClOK
NAO5FVKgMwEhpCW0CmPwkzckEcnCQOZDAMZoNBAZqCdFvoi6RS7v/Pdu9KuJTsl/6SZya73fOf1ne+c85
3z7f1WHrjhHvIQkRefS5eIniHntZt+/Wsan+CK54L01yU/rntG9P+4bngyYYczljlhRVPhWDSdNrPhMSN
s5dLhRDrcff1QOGXGjaZFiwKrXRv7e4j6hUJNjbFo3u7btJIWiGaiBhCKw7uhFyCMz3HXu7Aj87pz8qN0
yp2j0O4/IiqV/2bHmUG+NNi9/mpBYr2F/4dczHnBP61wHdC9BXRT1jiTZfMRR7cw1gITx5ssI2nGXB+Ou
zqNxXq7iTp/Exf5tcN1qlea9tHtG4huX0oknKXUT2tvsxLBnIASwYaoDctt2FHXILZlzSo9UyLt6vYiMA
MW0EzVXbxWfeWKxgcifsyLgLl+gWrdBNk51KXXOnYlrVuxjrd+Q9WaW31ALqo6jNpYIVA/zZIIvF8vFRd
af3ZFG/5iG6FZG/4iG4ussHIlG1qxjbJZG1qRjTKv9TSMRBZAZk16gGFTA2U+K+mZy1VV60EvCKQqcA65
9BbSXPd2kNX81uPglmmRxUytDq2+WInpwixlaxcgkpk02WTlioCps1aJ9V22FGI8YK3wYf6CSBlTC/WFV
WY5MH1htVkhRz1gVkrErOIhYFez4qIwF7e9BLi9lBnByDIWByvNGoxmLfMWY9Jy5i6u1Bf/ScJcwcxSfZ
FeaoYZ1fUFNXf5ZEJ1TS+J1IH5cH2V9VN49HB9tfT84folsLKSc71KipfqpS62TNcdLLKarYXCH6L8IvX
A1YtqkDdhDU9by0A6F5JucaLVsrLysnJbY6xcL68yIywvj6xj3xskbq5n3UZmbOBcsjNlFX6zideq3/Iu
1grVRzYytaaybO2W82Bo+lqzmZVfQwSRFmCNm/VyfY3dCrQkL/wFhNYHPndXrH/xFW6SicJR12/D+
=== truncated for brevity==
/xGfv/6//Pa7fzN8eiW37Yjv3/9Nl7/Cw==
```

Fig 1: Base64 encoded payload.



Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

☐ Strict mode

Input

```
BW1EvYxKX9M4DvqgfwhwVB4ZN4O3H+M46NOQHME6J2GhDVvMTT0i6+yps3B5JUp9BBO7gW3DxwliJQp3JfpkCuYNKTmLqeckdw
Acrqu8fuuMPp8qeW7bDLcHzsSQJrbFR2tcWrDlmTPhWucZ7TMzeqHRgXMlL9kkJefwJg+O3kVdSKqJbueDmzzb8WkE9+xMJKzX
gk8rke8o5pMHHx9HSz7HCj5+J1JaFqFraa08/
E20YwbctTjwG2gdCb8T9VydliKd1nl1Wot02ubVaSvSaZ9Xp71IZ9O8OptmdRYVRkKLCn0upFqLqLYiqr2I2sQ/PUxMf+be/
dR13Vdau4ePNLXXkTcshOYJk/AB0fXD/sZQeUWoVgQvA8FgqC4YrPUFQ/
WhdaENtT5+g7kQkqBDheolzxXpLRX6JszT8AptVcOiNljrWVwqBJtbThWhHKAnIIJqRchQgkFfWBE11VWliiJFwlFg2XJaLrwB
qPDf5aCGBb0klCD/baMFrkuepvlIYFkv/8cWlWOZvtMZ7uLIan1+2NdC0/
f6oDB9H8IOKhBghi8MxgNa2MNuaxp7CsRHcp0wMSoQlMqM2hqV2OZjPISmn+DsKdLik7wYBmntScl61mG94JMJ1MLECSkjn8wL
LAMNg4bjisJLCeH4coFTgWjDpIGpMZCUIn2qwaSgDIfddBZ/ZSH5sNLrGn80PwF/Ex7D+dD0O87wnhpWampqa6T+Ryp5eN9K/
Epop/QOCZUuKHBBhHY6jvwKAYSmP2EW/IQBERoIev0iNMKCA6EBFoT6PH6haE/fcvTQkva37/
SooT5FVRQ1CKzan99cpKuW90+UkOLWEzIQ6guEvUpNaDR0kx71RkBrwv1vh8v5t/
5hpfIwHjsHzfTMXXB40jJP20IT7p9svM5fbJoL/nPivvz/tZzntaPwPzESnr1xZTDkPVb+SmUYTfFkUsou1VN49/xGfv/6//
Pa7fzN8eiW37Yjv3/9Nl7/Cw==
```

ABC 7012 1 Raw Bytes LF

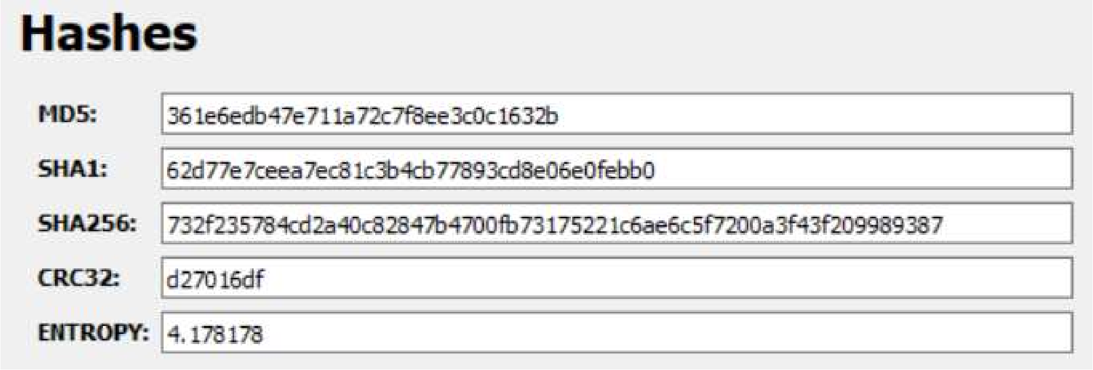
Output

```
íZ{p\åu?ßÝÝ»Wk{•»zÚ•ì•mÙkY•õ²ñ␡ëeKF␡¶%Ù␡•Ø«Ý+iñîÞõ½»¶•ÇT4•)N(Ð␡äUJ•Ì␡•••Ð)•ÂLÜ•G'   ␡•␡␡ Ð@f •␡
ú"é␡»¿óÝ»Ò®%;%ÿ¤•É®÷|çõ•ï•ó•óíýV␡,á␡ò␡•␡•K•••!çµ•~ýk␡•à•ç•ôx%?®{Fôÿ,nx2a•3•9aESáX4•6³á1#låÒáD:Ü}
ýP8eÆ•¦E•␡«]␡û{•ú•BM•±hÞîÛ´•␡•f¢␡␡•Ã»¡␡ •Ïqx»°#óºsò£tÊ•£ÐÎ?"*•ÿfÇ•A¾4Ø¼þjAb¼•ÿ•\ÌyÁ?•p␡Ð¼␡tSÖ8•eó
␡G·0Ö␡␡Ç•,#iÆ\␡•»:•Åz»•:•␡␡ùµÃuªW•öÑí␡•n_J$•¥ÔOko³␡Á••␡Á•¨␡ËmØQx ¶eÍ*=S"íêö"0␡␡ÐLÕ]¼V}å•Æ␡"~Ì••¹~
•jÝ␡Ù9Ô¥x:v%•[±•·~CÕ•[}@.ª:•ÚX!P?Í•␡¼_/␡␡Z•vE␡þb␡¡Y␡þ"␡•¬°r%␡Z±•²Y␡Z••2¯õ4•D␡@fMz•aS␡e>+é•ËUUëA/␡
¤*p␡¹ô␡Ò\÷v•ÕüÖãà•i•ÅL•␡•¾X•éÂ,ek␡ ••4Ùdå••©³V•õ]¶␡b<`•ðaþ•H␡S␡õ•Uf90}aµY!G=`VJÄ¬ã!`W³â¢0␡·½␡,½•␡
ÁÈ2␡␡+Í␡•f-ó␡còræ.®Ô␡ÿIÂ\ÂÌR}•^j•␡Õõ␡5wùdBuM/•Ô•ùp}•õSxôp}µôüáú%°²•s½J••ê¥.¶Lx␡,²•••Â␡¢ü"õÀÕ•j•7a
␡O[Ë@:␡•nq¢Õ²²ò²r[c¬\/¯2#,/•¬cß␡$n®gÝFflà\²3e␡~³•xªßò.ö
ÕG62µ¦²lí•ô`húZ³••_C␡•␡`••õr}•Ý
´$/ü␡•Ö␡>wW¬•ñ␡n••ÃQxoÃúm f␡␡+•ÌÏ6ÛÙóµUGÊÖêkKÌM ®M\ºt•]Ð½º_÷J•¹•Á<óå␡•x␡¬Ñx•U¼åYó␡Ç,␡•m¥°óV0T•±␡
M¹ÌÜ
```

Basic Static Analysis

{Screenshots and description about basic static artifacts and methods}

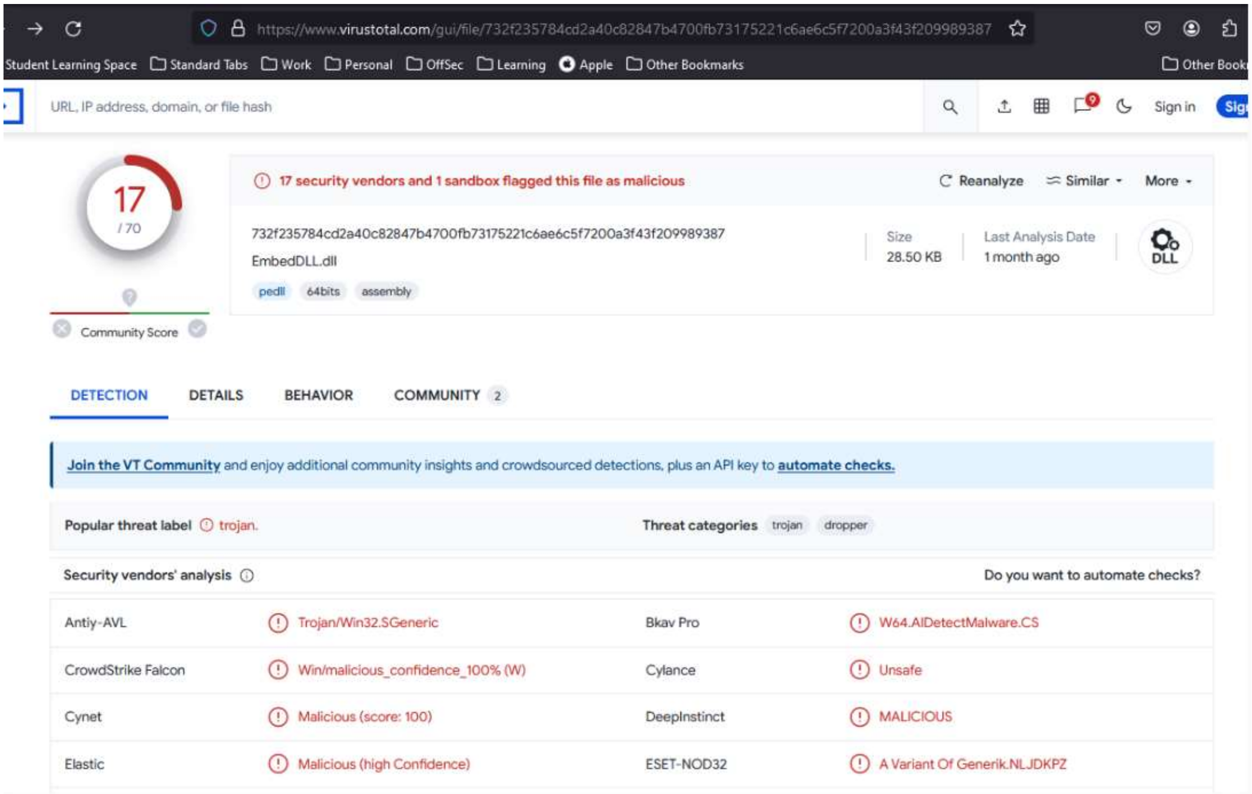
Using Cutter, the SHA256 and MD5sum information can be extracted from the dashboard



Hashes

MD5:	361e6edb47e711a72c7f8ee3c0c1632b
SHA1:	62d77e7ceea7ec81c3b4cb77893cd8e06e0febb0
SHA256:	732f235784cd2a40c82847b4700fb73175221c6ae6c5f7200a3f43f209989387
CRC32:	d27016df
ENTROPY:	4.178178

With the hash value, we run a check with VirusTotal, where we observed there are 17 security vendors who flagged this as malicious.



17 / 70

17 security vendors and 1 sandbox flagged this file as malicious

732f235784cd2a40c82847b4700fb73175221c6ae6c5f7200a3f43f209989387
EmbedDLL.dll
pedll 64bits assembly

Size 28.50 KB
Last Analysis Date 1 month ago
DLL

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

Popular threat label trojan.

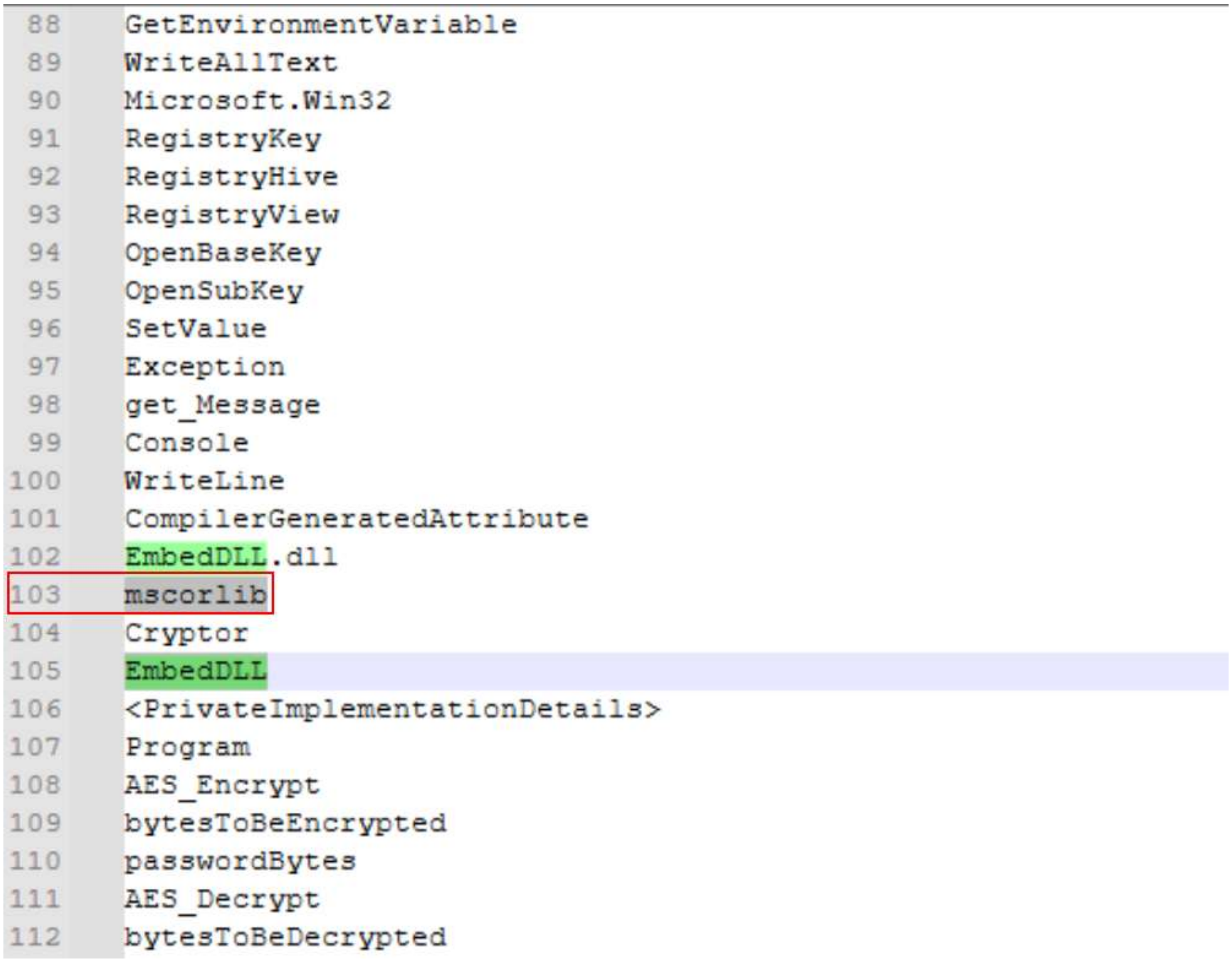
Threat categories trojan dropper

Security vendors' analysis

Do you want to automate checks?

Antiy-AVL	Trojan/Win32.SGeneric	Bkav Pro	W64.AIDetectMalware.CS
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cylance	Unsafe
Cynet	Malicious (score: 100)	DeepInstinct	MALICIOUS
Elastic	Malicious (high Confidence)	ESET-NOD32	A Variant Of Generik.NLJDKPZ

From the floss output, we noted the keywords mscorlib, which is an abbreviation for Multi-language Standard Common Object Runtime Library. This means that the dll is a .Net application, or specifically, a C# (C-Sharp) application.

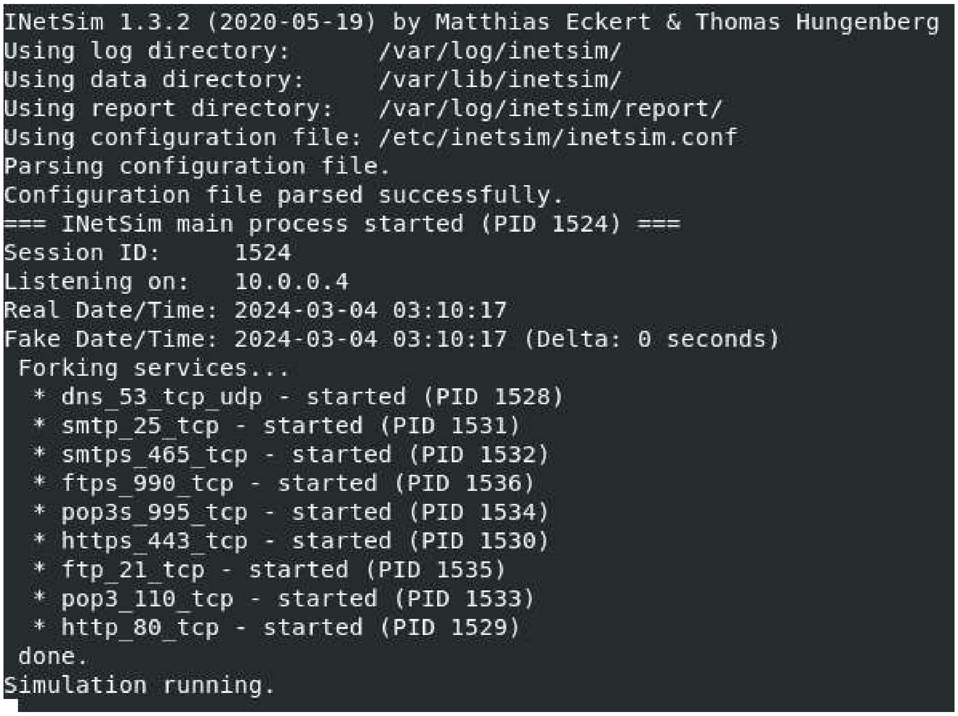


```
GetEnvironmentVariable
WriteAllText
Microsoft.Win32
RegistryKey
RegistryHive
RegistryView
OpenBaseKey
OpenSubKey
SetValue
Exception
get_Message
Console
WriteLine
CompilerGeneratedAttribute
EmbedDLL.dll
mscorlib
Cryptor
EmbedDLL
<PrivateImplementationDetails>
Program
AES_Encrypt
bytesToBeEncrypted
passwordBytes
AES_Decrypt
bytesToBeDecrypted
```

Basic Dynamic Analysis

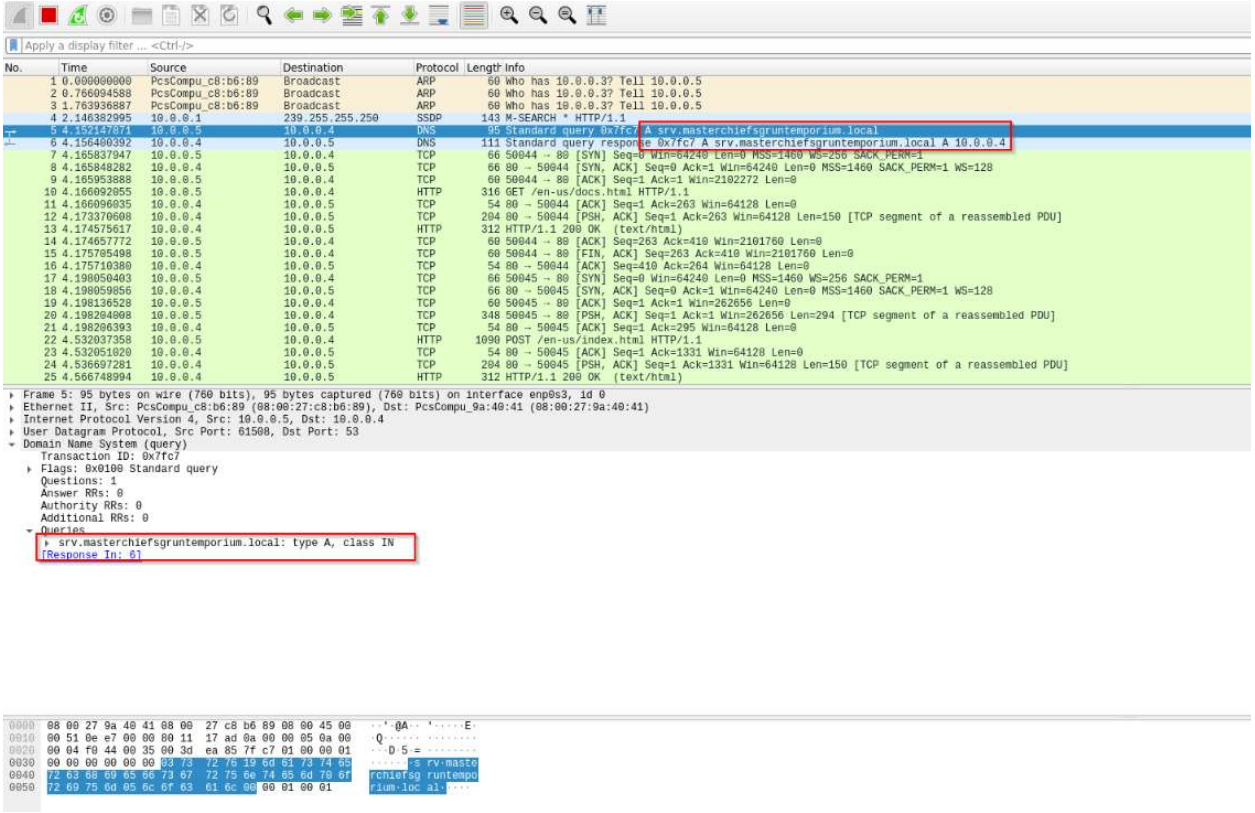
{Screenshots and description about basic dynamic artifacts and mUsingethods}

Prior to detonating the malicious dll, setup inetsim to act as a proxy server to serve out the required information and also to monitor the type of information that is triggered from the source malware.



```
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 1524) ===
Session ID:     1524
Listening on:   10.0.0.4
Real Date/Time: 2024-03-04 03:10:17
Fake Date/Time: 2024-03-04 03:10:17 (Delta: 0 seconds)
 Forking services...
  * dns_53_tcp_udp - started (PID 1528)
  * smtp_25_tcp - started (PID 1531)
  * smtps_465_tcp - started (PID 1532)
  * ftps_990_tcp - started (PID 1536)
  * pop3s_995_tcp - started (PID 1534)
  * https_443_tcp - started (PID 1530)
  * ftp_21_tcp - started (PID 1535)
  * pop3_110_tcp - started (PID 1533)
  * http_80_tcp - started (PID 1529)
 done.
Simulation running.
```

Upon activation, we noted through wireshark that traffic was requested from the source (or affected host) to *srv.masterchiefsgruntemporium.local*

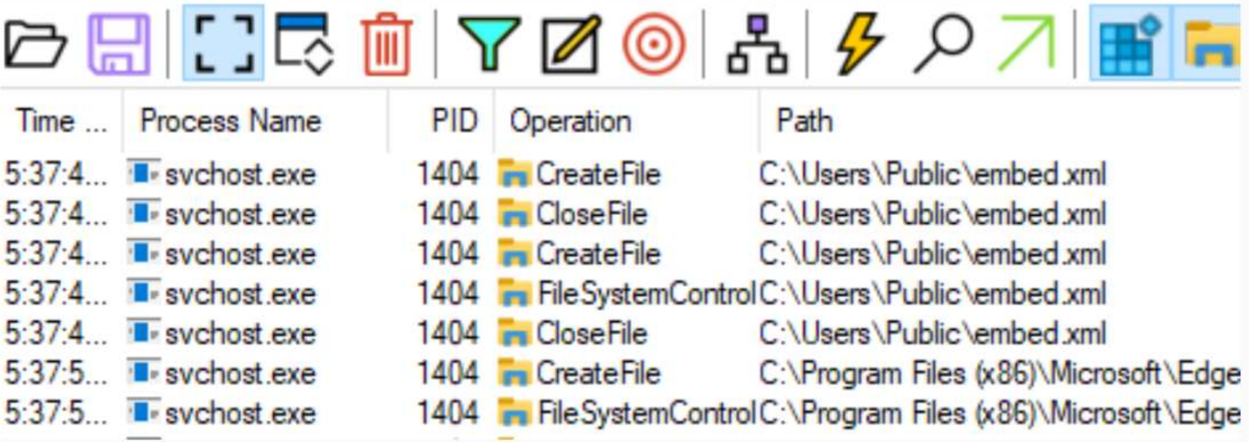


```
No.  Time        Source              Destination      Protocol Length Info
   1 0.000000000 PcsCompu_c8:b6:89   Broadcast        ARP        60 Who has 10.0.0.3? Tell 10.0.0.5
   2 0.766094588 PcsCompu_c8:b6:89   Broadcast        ARP        60 Who has 10.0.0.3? Tell 10.0.0.5
   3 1.763936887 PcsCompu_c8:b6:89   Broadcast        ARP        60 Who has 10.0.0.3? Tell 10.0.0.5
   4 2.146382995 10.0.0.1            239.255.255.250  SSDP      143 M-SEARCH * HTTP/1.1
   5 4.152147871 10.0.0.5            10.0.0.4         DNS        95 Standard query 0x7fc7 A srv.masterchiefsgruntemporium.local
   6 4.156400392 10.0.0.4            10.0.0.5         DNS       111 Standard query response 0x7fc7 A srv.masterchiefsgruntemporium.local A 10.0.0.4
   7 4.165837947 10.0.0.5            10.0.0.4         TCP        66 50044 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   8 4.165848282 10.0.0.4            10.0.0.5         TCP        66 80 → 50044 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
   9 4.165953888 10.0.0.5            10.0.0.4         TCP        60 50044 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
  10 4.166092055 10.0.0.5            10.0.0.4         HTTP      316 GET /en-us/docs.html HTTP/1.1
  11 4.166096035 10.0.0.4            10.0.0.5         TCP        54 80 → 50044 [ACK] Seq=1 Ack=263 Win=64128 Len=0
  12 4.173370608 10.0.0.4            10.0.0.5         TCP       204 80 → 50044 [PSH, ACK] Seq=1 Ack=263 Win=64128 Len=150 [TCP segment of a reassembled PDU]
  13 4.174575617 10.0.0.4            10.0.0.5         HTTP      312 HTTP/1.1 200 OK  (text/html)
  14 4.174657772 10.0.0.5            10.0.0.4         TCP        60 50044 → 80 [ACK] Seq=263 Ack=410 Win=2101760 Len=0
  15 4.175705498 10.0.0.5            10.0.0.4         TCP        60 50044 → 80 [FIN, ACK] Seq=263 Ack=410 Win=2101760 Len=0
  16 4.175710380 10.0.0.4            10.0.0.5         TCP        54 80 → 50044 [ACK] Seq=410 Ack=264 Win=64128 Len=0
  17 4.198050403 10.0.0.5            10.0.0.4         TCP        66 50045 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  18 4.198059856 10.0.0.4            10.0.0.5         TCP        66 80 → 50045 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
  19 4.198136528 10.0.0.5            10.0.0.4         TCP        60 50045 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
  20 4.198204008 10.0.0.5            10.0.0.4         TCP       348 50045 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262656 Len=294 [TCP segment of a reassembled PDU]
  21 4.198206393 10.0.0.4            10.0.0.5         TCP        54 80 → 50045 [ACK] Seq=1 Ack=295 Win=64128 Len=0
  22 4.532037358 10.0.0.5            10.0.0.4         HTTP     1090 POST /en-us/index.html HTTP/1.1
  23 4.532051020 10.0.0.4            10.0.0.5         TCP        54 80 → 50045 [ACK] Seq=1 Ack=1331 Win=64128 Len=0
  24 4.536697281 10.0.0.4            10.0.0.5         TCP       204 80 → 50045 [PSH, ACK] Seq=1 Ack=1331 Win=64128 Len=150 [TCP segment of a reassembled PDU]
  25 4.566748994 10.0.0.4            10.0.0.5         HTTP      312 HTTP/1.1 200 OK  (text/html)
```

```
▸ Frame 5: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: PcsCompu_c8:b6:89 (08:00:27:c8:b6:89), Dst: PcsCompu_9a:40:41 (08:00:27:9a:40:41)
▸ Internet Protocol Version 4, Src: 10.0.0.5, Dst: 10.0.0.4
▸ User Datagram Protocol, Src Port: 61508, Dst Port: 53
▾ Domain Name System (query)
    Transaction ID: 0x7fc7
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▸ srv.masterchiefsgruntemporium.local: type A, class IN
    [Response In: 6]
```

```
0000  08 00 27 9a 40 41 08 00  27 c8 b6 89 08 00 45 00   ··'·@A·· '·····E·
0010  00 51 0e e7 00 00 80 11  17 ad 0a 00 00 05 0a 00   ·Q······ ········
0020  00 04 f0 44 00 35 00 3d  ea 85 7f c7 01 00 00 01   ···D·5·= ········
0030  00 00 00 00 00 00 03 73  72 76 19 6d 61 73 74 65   ·······s rv·maste
0040  72 63 68 69 65 66 73 67  72 75 6e 74 65 6d 70 6f   rchiefsg runtempo
0050  72 69 75 6d 05 6c 6f 63  61 6c 00 00 01 00 01      rium·loc al·····
```

Through Procmon, we also noted that upon detonation, files were created using embed.xml



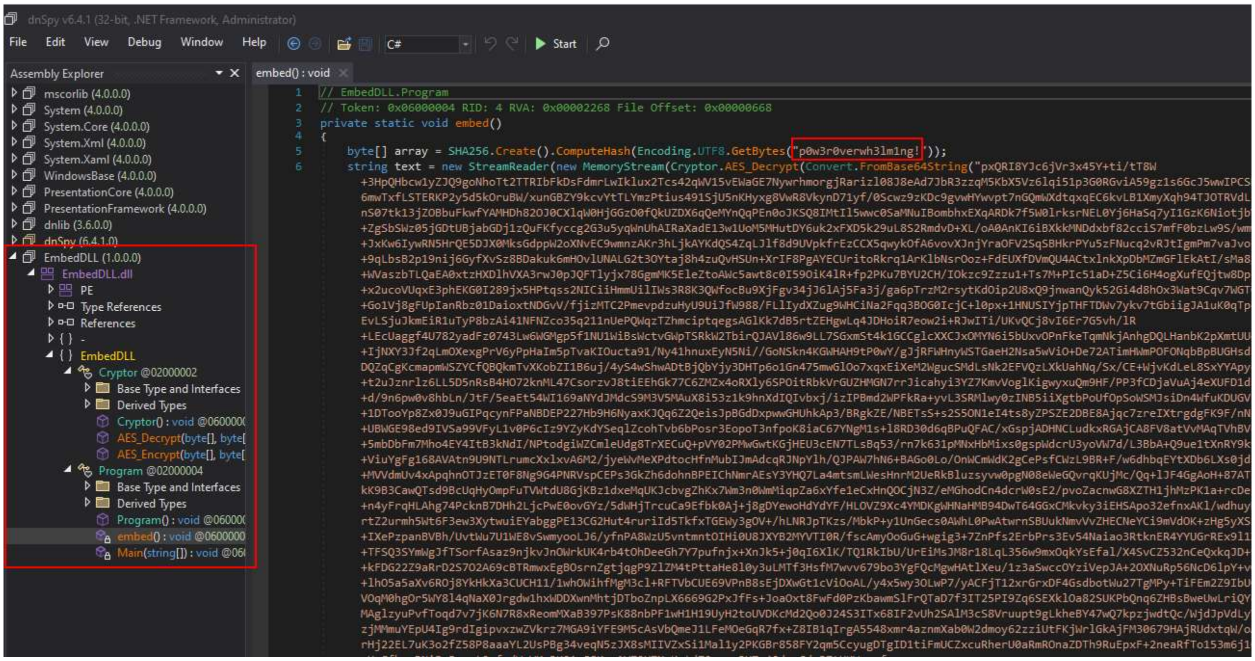
Time ...	Process Name	PID	Operation	Path
5:37:4...	svchost.exe	1404	CreateFile	C:\Users\Public\embed.xml
5:37:4...	svchost.exe	1404	CloseFile	C:\Users\Public\embed.xml
5:37:4...	svchost.exe	1404	CreateFile	C:\Users\Public\embed.xml
5:37:4...	svchost.exe	1404	FileSystemControl	C:\Users\Public\embed.xml
5:37:4...	svchost.exe	1404	CloseFile	C:\Users\Public\embed.xml
5:37:5...	svchost.exe	1404	CreateFile	C:\Program Files (x86)\Microsoft\Edge
5:37:5...	svchost.exe	1404	FileSystemControl	C:\Program Files (x86)\Microsoft\Edge

Advanced Static Analysis

{Screenshots and description about findings during advanced static analysis}

To further analyse the binary and confirm on the source code, dnSpy was used to inspect the code. From the screenshot below, we noted a few things:

1. The binary is named as EmbedDll;
2. The main method is ***embed()***. This is useful when we want to trigger the dll binary;
3. The payload (base64 encoded) is encrypted using the keyword: p0w3r0verwh3lm1ng!;
4. There are 2 components within EmbedDLL, namely Cryptor and Program.



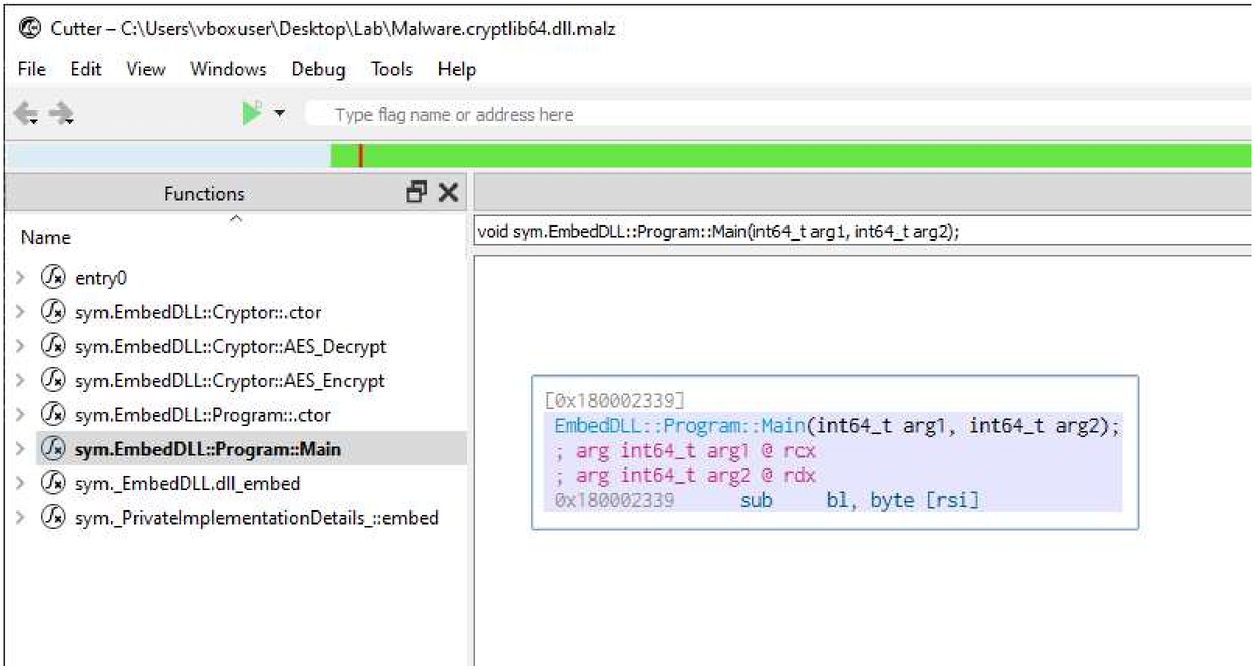
```
// EmbedDLL.Program
// Token: 0x06000004 RID: 4 RVA: 0x00002268 File Offset: 0x00000668
private static void embed()
{
    byte[] array = SHA256.Create().ComputeHash(Encoding.UTF8.GetBytes("p0w3r0verwh3lm1ng!"));
    string text = new StreamReader(new MemoryStream(Cryptor.AES_Decrypt(Convert.FromBase64String("pxQRI8YJc6jVr3x45Y+ti/tT8W
      +3HpQHbcw1yZJQ9goNhoTt2TTRIbFkDsFdmrLwIklux2Tcs42qWV15vEWaGE7NywrhmorgjRarizl08J8eAd7JbR3zzqM5KbX5Vz6lqi51p3G0RGviA59gz1s6GcJ5wwIPCS
      6mwTxfLSTERKP2y5d5kOruBW/xunGBZY9kcvYtTLYmzPtius491SjU5nKHyxg8VwR8VkynD71yf/0Scwz9zKDc9gvwHYwvpt7nGQmWXdtqxqEC6kvLB1XmyXqh94TJOTRVdL
      nS07tk13jZOBbuFkwfYAMHDh82OJ0CXlqW0HjGGzO0fQkUZDX6qQeMYnQqPEn0oJKSQ8IMtIl5wwc0SaMNuIBombhxEXqARDk7f5W0lrksrNEL0Yj6HaSq7yI1GzK6Niotjb
      +ZgSbSWz05jGDtUBjabGDj1zQuFKfyccg2G3u5yqWnUhAIRaXadE13w1UoM5MHutDY6uk2xFXD5k29uL8S2RmdvD+XL/oA0AnKI6iBXkkMNDdxbf82cciS7mfF0bzLw9S/wm
      +JxKw6IywRN5HrQE5DJX0MksGdppW2oXNvEC9wmnzAKr3hLjkAYKdQS4ZqLJlf8d9UVpkfrEzCCX5qwykOfA6vovXJnjYraOFV2SqSBHkrPYu5zFNucq2vRJtIgmPm7vaJvo
      +9qLbsB2p19nij6GyfXvSz8BDakuk6mHOvlUNALG2t3OYtaj8h4zuQvHSUn+XrIF8PgAYECUritoRkrq1ArKlbNsrOoz+FdEUXfDVmQU4ACtxlnkXpDbMZmGFlEkAtI/sMa8
      +WVaszbTLQaEA0xtzHXDlhVXA3rwJ0pJQFTlyjx78GgmMK5EleZtoAWc5awt8c0I59OiK4lR+fp2PKu7BYU2CH/IOkzc9Zzzu1+Ts7M+PIc51aD+Z5Ci6H4ogXufEQjtw8Dp
      +x2ucoVUqxE3phEKG0I289jx5HPtqss2NICiiHmmUilIWs3R8K3QWfocBu9XjFgv34jJ6lAj5Fa3j/ga6pTrzM2rsytKdOip2U8xQ9jnwanQyk52Gi4d8hOx3Wat9Cqv7WGT
      +Go1Vj8gFUpIanRbz01DaioxtNDGvV/fjizMTC2PmevpdzuHyU9UiJfW988/FLlIydXZug9WHCiNa2Fqq3BOG0IcjC+l0px+1HNUSIYjpTHFTDWv7ykv7tGbiigJA1uK0qTp
      EvLSjuJkmEiR1uTyP8bzAi41NFNZco35q211nUePQWqzTZhmciptqegsAGlKk7dB5rtZEHgwLq4JDHoiR7eow2i+RJwITi/UKvQCj8vI6Er7G5vh/lR
      +LEcUaggf4U782yadFz0743Lw6WGMgp5f1NU1WiBsWctvGWpTSRkW2TbirQJAVl86w9LL7SGxmSt4k1GCCglcXXCJxOMYN6i5bUxvOPnFkeTqmNkjAnhgDQLHanbK2pXmtUU
      +IjNXY3Jf2qLmOXexgPrV6yPpHaIm5pTvaKIOucta91/Ny41hnuxEyN5Ni//GoNSkn4KGWHAH9tP0wY/gJjRFWHnyWSTGaeH2Nsa5wViO+De72ATimHWmPOFONqbBpBUGHsd
      DQZqCgKcmapmWSZYCfQBQkmTvXKobZI1B6uj/4yS4wShwADtBjQbYjy3DHTp6o1Gn475mwGlOo7xqxEiXeM2WgucSMdLsNk2EFVQzLXkUahNq/Sx/CE+WjvKdLeL8SxYYApy
      +t2uJznrlz6LL5D5nRsB4HO72knML47CsorzvJ8tiEEhGk77C6ZMZx4oRXly6SPOitRbkVrGUZHMGN7rrJicahyi3YZ7KmvVoglKigwyxuQm9HF/PP3fCDjaVuAj4eXUFD1d
      +d/9n6pw0v8hbLn/JtF/5eaEt54WI169aNYdJMdcS9M3V5MAuX8i53z1k9hnXdIQIvbxj/izIPBmd2WPFkRa+yvL3SRMlwy0zINB5iiXgtbPoUfOpSoWSMJsiDn4WfuKDUGV
      +1DTooYp8Zx0J9uGIPqcynFPaNBDEP227Hb9H6NyaxKJQq6Z2QeisJpBGdDxpwwGHUhkAp3/BRgkZE/NBETsS+s2S5ON1eI4ts8yZPSZE2DBE8Ajqc7zreIXtrgdgFK9F/nN
      +UBWGE98ed9IVSa99VFyL1v0P6cIz9YZyKdYSeqlZcohTvb6bPosr3EopoT3nfpoK8iaC67YNgM1s+l8RD30d6qBPuQFAC/xGspjADHNCLudkxRGAjCA8FV8atVvMAqTVhBV
      +5mbDbFm7Mho4EY4ItB3kNdI/NPtodgiWZCmleUdg8TrXECuQ+pVY02PMwGwtKGjHEU3cEN7TLsBq53/rn7k631pMNxHbMixs0gspWdcrU3yoVW7d/L3BbA+Q9ue1tXnRY9k
      +ViuYgFg168AVAtn9U9NTLrumcXxlxvA6M2/jyeWvMeXPdtocHfnMubIJmAdcqRJNpYlh/QJPAW7hN6+BAGo0Lo/OnWCmWdK2gCePsfCWzL9BR+F/w6dhbqEYtXDb6LXs0jd
      +MVVdmUv4xApqhnOTJzET0F8Ng9G4PNRVspCEPs3GkZh6dohnBPEIChNmrAEsY3YHQ7La4mtsmLWesHnrM2UeRkBluzsyvw0pgN08eWeGQvrqKUjMc/Qq+lJF4GgAoH+87AT
      kK9B3CawQTsd9BcUqHyOmpFuTVWtdU8GjKBz1dxeMqUKJcbvgZhKx7Wm3n0WmMiqpZa6xYfe1eCxHnQOCjN3Z/eMGhodCn4dcrW0sE2/pvoZacnwG8XZTH1jhMzPK1a+rcDe
      +n4yFrqHLAhg74PcknB7DHh2LjcPwE0ovGYz/5dWHjTrcuCa9Efbk0Aj+j8gDYewoHdYdYF/HLOVZ9Xc4YMDKgWHNaHMB94DwT64GGxCMkvky3iEHSApo32efnxAKl/wdhuy
      rtZ2urmh5Wt6F3ew3XytwuiEYabggPE13CG2Hut4ruriId5TkfxTGEWy3gOV+/hLNRJpTKzs/MbkP+y1UnGecs0AWhL0PwAtwrnSBUukNmvVvZHECNeYCi9mVdOK+zHg5yXS
      +IXePzpanBVBh/UvtWu7U1WE8vSwmyooLJ6/yfnPA8WzU5vntmntOIHi0U8JXYB2MYVTI0R/fscAmyOoGuG+wgig3+7ZnPfs2ErbPrs3Ev54Naiao3RtknER4YYUGrREx9l1
      +TFSQ3SYmWgJfTSorfAsaz9njkvJnOWrkUK4rb4tOhDeeGh7Y7pufnjx+XnJk5+j0qI6XlK/TQ1RkIbU/UrEiMsJM8r18LqL356w9mxOqkYsEfal/X4SvCZ532nCeQxkqJD+
      +kFDG22Z9aRrD2S7O2A69cBTRmwxEgBOsrnZgtjqgP9ZlZM4tPttaHe8l0y3uLMTf3HsfM7wvv679bo3YgFQcMgwHAtlXeu/1z3aSwccOYziVepJA+2OXNuRp56NcD6lpY+v
      +lhO5a5aXv6ROj8YkHkXa3CUCH11/1whOWihfMgM3cl+RFTVbCUE69VPnB8sEjDXwGt1cViOoAL/y4x5wy3OLwP7/yACFjT12xrGrxDF4GsdbotWu27TgMPy+TiFEm2Z9IbU
      VOqM0hgOr5WY8l4qNaX0Jrgdw1hxWDDXwnMhtjDTboZnpLX6669G2PxJfFs+JoaOxt8FwFd0PzKbawmSlFrQTaD7f3IT25PI9Zq6SEXklOa82SUKPbQnq6ZHBsBweUwLriQY
      MAglzyuPvfToqd7v7jK6N7R8xReomMXaB397PsK88nbPF1wH1H19UyH2toUVDKcMd2Qo0J24S3ITx68IF2vUh2SAlM3cS8Vruupt9gLkheBY47wQ7kpzjwdtQc/WjdJpVdLy
      zjMMmuYEpU4Ig9rdIgipvxzwZVkrz7MGA9iYFE9M5cAsVbQmeJ1LFeMOeGqR7fx+Z8IB1qIrgA5548xmr4aznmXab0W2dmoy62zziUtFKjWrlGkAjFM30679HAjRUdxtqW/o
      rHj22EL7uK3o2fZ58P8aaaYL2UsPBg34veqN5zJX8sMIIVZxSi1Mal1y2PKGBr858FY2qm5CcyugDTgID1tiFmUCZxcuRherU0aRmROnaZDTh9RuEpxF+2neaRfTo153m6j1
```

Advanced Dynamic Analysis

{Screenshots and description about advanced dynamic artifacts and methods}

Based on the information retrieved from Advanced Static Analysis, we reviewed the disassembly flow chart under Cutter.

Under Main function

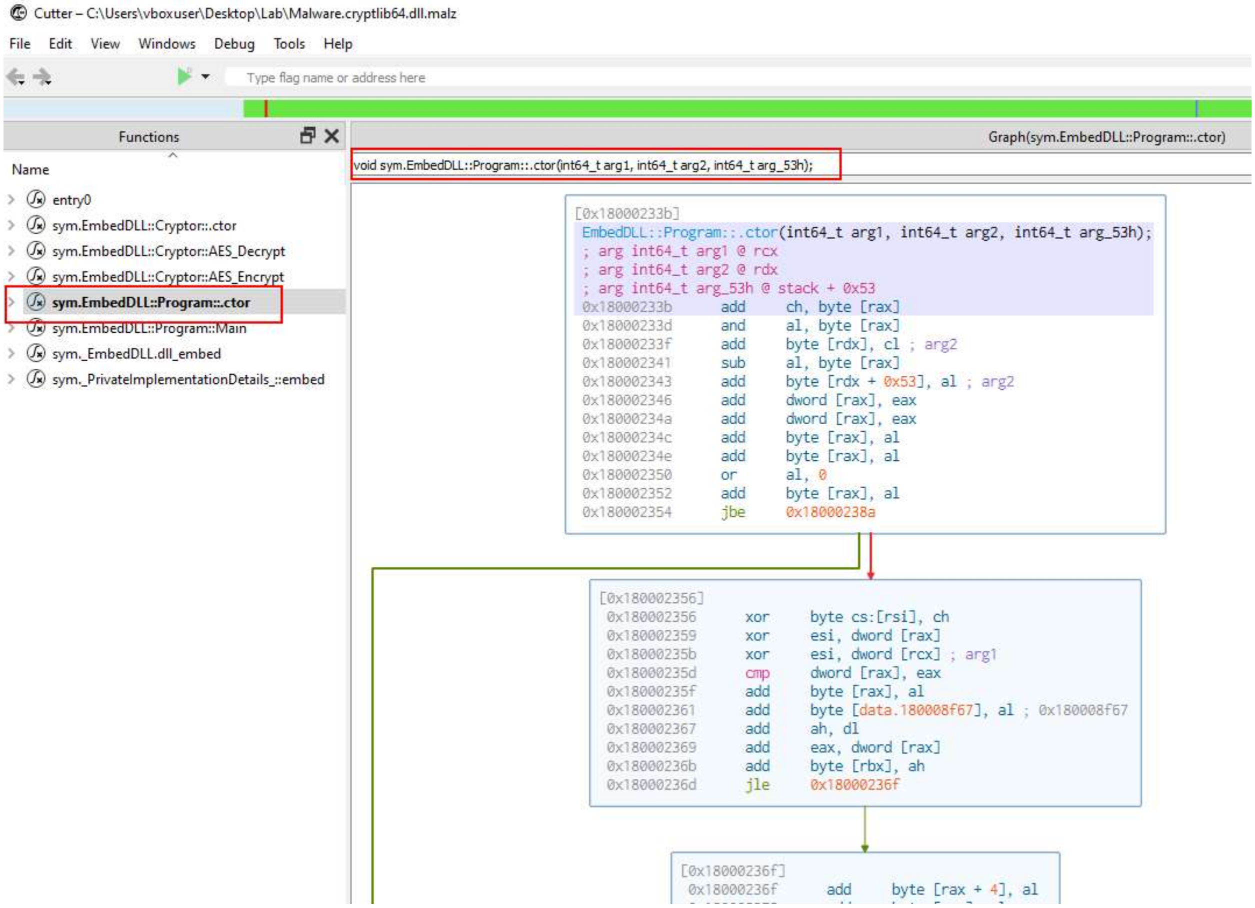

Cutter – C:\Users\vboxuser\Desktop\Lab\Malware.cryptlib64.dll.malz
File
Edit
View
Windows
Debug
Tools
Help
Type flag name or address here
Functions
Name

- entry0
- sym.EmbedDLL::Cryptor::.ctor
- sym.EmbedDLL::Cryptor::AES_Decrypt
- sym.EmbedDLL::Cryptor::AES_Encrypt
- sym.EmbedDLL::Program::.ctor
- **sym.EmbedDLL::Program::Main**
- sym._EmbedDLL.dll_embed
- sym._PrivateImplementationDetails_::embed

```
void sym.EmbedDLL::Program::Main(int64_t arg1, int64_t arg2);
```

```
[0x180002339]
EmbedDLL::Program::Main(int64_t arg1, int64_t arg2);
; arg int64_t arg1 @ rcx
; arg int64_t arg2 @ rdx
0x180002339      sub     bl, byte [rsi]
```

Which basically does not tell us much about the execution flow for embedDLL program. If we explore further into the ctor, we will notice more information

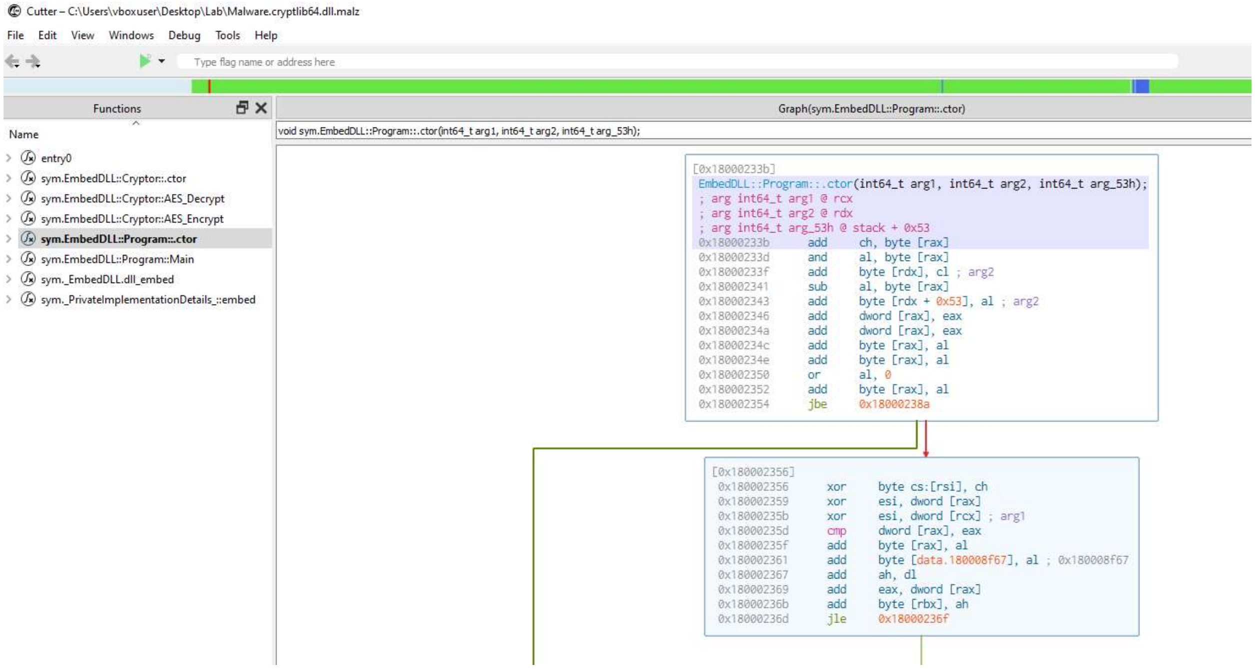

Cutter – C:\Users\vboxuser\Desktop\Lab\Malware.cryptlib64.dll.malz
File Edit View Windows Debug Tools Help
Type flag name or address here
Functions
Name
entry0
sym.EmbedDLL::Cryptor::.ctor
sym.EmbedDLL::Cryptor::AES_Decrypt
sym.EmbedDLL::Cryptor::AES_Encrypt
sym.EmbedDLL::Program::.ctor
sym.EmbedDLL::Program::Main
sym._EmbedDLL.dll_embed
sym._PrivateImplementationDetails_::embed
Graph(sym.EmbedDLL::Program::.ctor)

```
void sym.EmbedDLL::Program::.ctor(int64_t arg1, int64_t arg2, int64_t arg_53h);
```

```
[0x18000233b]
EmbedDLL::Program::.ctor(int64_t arg1, int64_t arg2, int64_t arg_53h);
; arg int64_t arg1 @ rcx
; arg int64_t arg2 @ rdx
; arg int64_t arg_53h @ stack + 0x53
0x18000233b      add     ch, byte [rax]
0x18000233d      and     al, byte [rax]
0x18000233f      add     byte [rdx], cl ; arg2
0x180002341      sub     al, byte [rax]
0x180002343      add     byte [rdx + 0x53], al ; arg2
0x180002346      add     dword [rax], eax
0x18000234a      add     dword [rax], eax
0x18000234c      add     byte [rax], al
0x18000234e      add     byte [rax], al
0x180002350      or      al, 0
0x180002352      add     byte [rax], al
0x180002354      jbe     0x18000238a
```

```
[0x180002356]
0x180002356      xor     byte cs:[rsi], ch
0x180002359      xor     esi, dword [rax]
0x18000235b      xor     esi, dword [rcx] ; arg1
0x18000235d      cmp     dword [rax], eax
0x18000235f      add     byte [rax], al
0x180002361      add     byte [data.180008f67], al ; 0x180008f67
0x180002367      add     ah, dl
0x180002369      add     eax, dword [rax]
0x18000236b      add     byte [rbx], ah
0x18000236d      jle     0x18000236f
```

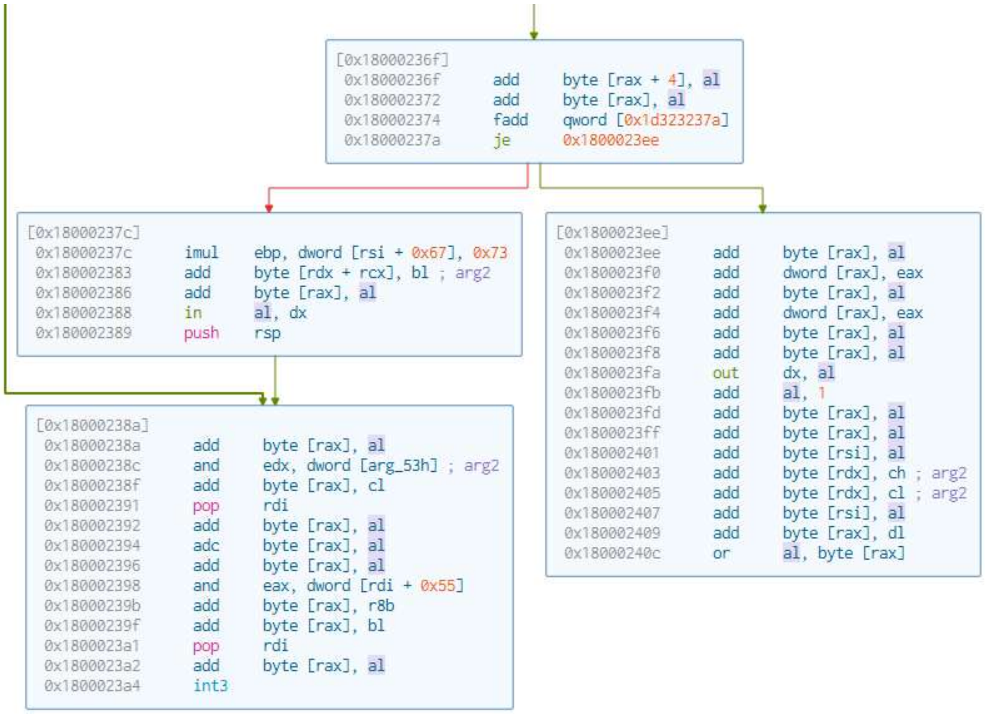
```
[0x18000236f]
0x18000236f      add     byte [rax + 4], al
```


Cutter – C:\Users\vboxuser\Desktop\Lab\Malware.cryptlib64.dll.malz
File Edit View Windows Debug Tools Help
Type flag name or address here
Functions
Name
entry0
sym.EmbedDLL::Cryptor::.ctor
sym.EmbedDLL::Cryptor::AES_Decrypt
sym.EmbedDLL::Cryptor::AES_Encrypt
sym.EmbedDLL::Program::.ctor
sym.EmbedDLL::Program::Main
sym._EmbedDLL.dll_embed
sym._PrivateImplementationDetails_::embed
Graph(sym.EmbedDLL::Program::.ctor)

```
void sym.EmbedDLL::Program::.ctor(int64_t arg1, int64_t arg2, int64_t arg_53h);
```

```
[0x18000233b]
EmbedDLL::Program::.ctor(int64_t arg1, int64_t arg2, int64_t arg_53h);
; arg int64_t arg1 @ rcx
; arg int64_t arg2 @ rdx
; arg int64_t arg_53h @ stack + 0x53
0x18000233b      add     ch, byte [rax]
0x18000233d      and     al, byte [rax]
0x18000233f      add     byte [rdx], cl ; arg2
0x180002341      sub     al, byte [rax]
0x180002343      add     byte [rdx + 0x53], al ; arg2
0x180002346      add     dword [rax], eax
0x18000234a      add     dword [rax], eax
0x18000234c      add     byte [rax], al
0x18000234e      add     byte [rax], al
0x180002350      or      al, 0
0x180002352      add     byte [rax], al
0x180002354      jbe     0x18000238a
```

```
[0x180002356]
 0x180002356      xor     byte cs:[rsi], ch
 0x180002359      xor     esi, dword [rax]
 0x18000235b      xor     esi, dword [rcx] ; arg1
 0x18000235d      cmp     dword [rax], eax
 0x18000235f      add     byte [rax], al
 0x180002361      add     byte [data.180008f67], al ; 0x180008f67
 0x180002367      add     ah, dl
 0x180002369      add     eax, dword [rax]
 0x18000236b      add     byte [rbx], ah
 0x18000236d      jle     0x18000236f
```



```
[0x18000236f]
 0x18000236f      add     byte [rax + 4], al
 0x180002372      add     byte [rax], al
 0x180002374      fadd    qword [0x1d323237a]
 0x18000237a      je      0x1800023ee
```

```
[0x18000237c]
 0x18000237c      imul    ebp, dword [rsi + 0x67], 0x73
 0x180002383      add     byte [rdx + rcx], bl ; arg2
 0x180002386      add     byte [rax], al
 0x180002388      in      al, dx
 0x180002389      push    rsp
```

```
[0x1800023ee]
 0x1800023ee      add     byte [rax], al
 0x1800023f0      add     dword [rax], eax
 0x1800023f2      add     byte [rax], al
 0x1800023f4      add     dword [rax], eax
 0x1800023f6      add     byte [rax], al
 0x1800023f8      add     byte [rax], al
 0x1800023fa      out     dx, al
 0x1800023fb      add     al, 1
 0x1800023fd      add     byte [rax], al
 0x1800023ff      add     byte [rax], al
 0x180002401      add     byte [rsi], al
 0x180002403      add     byte [rdx], ch ; arg2
 0x180002405      add     byte [rdx], cl ; arg2
 0x180002407      add     byte [rsi], al
 0x180002409      add     byte [rax], dl
 0x18000240c      or      al, byte [rax]
```

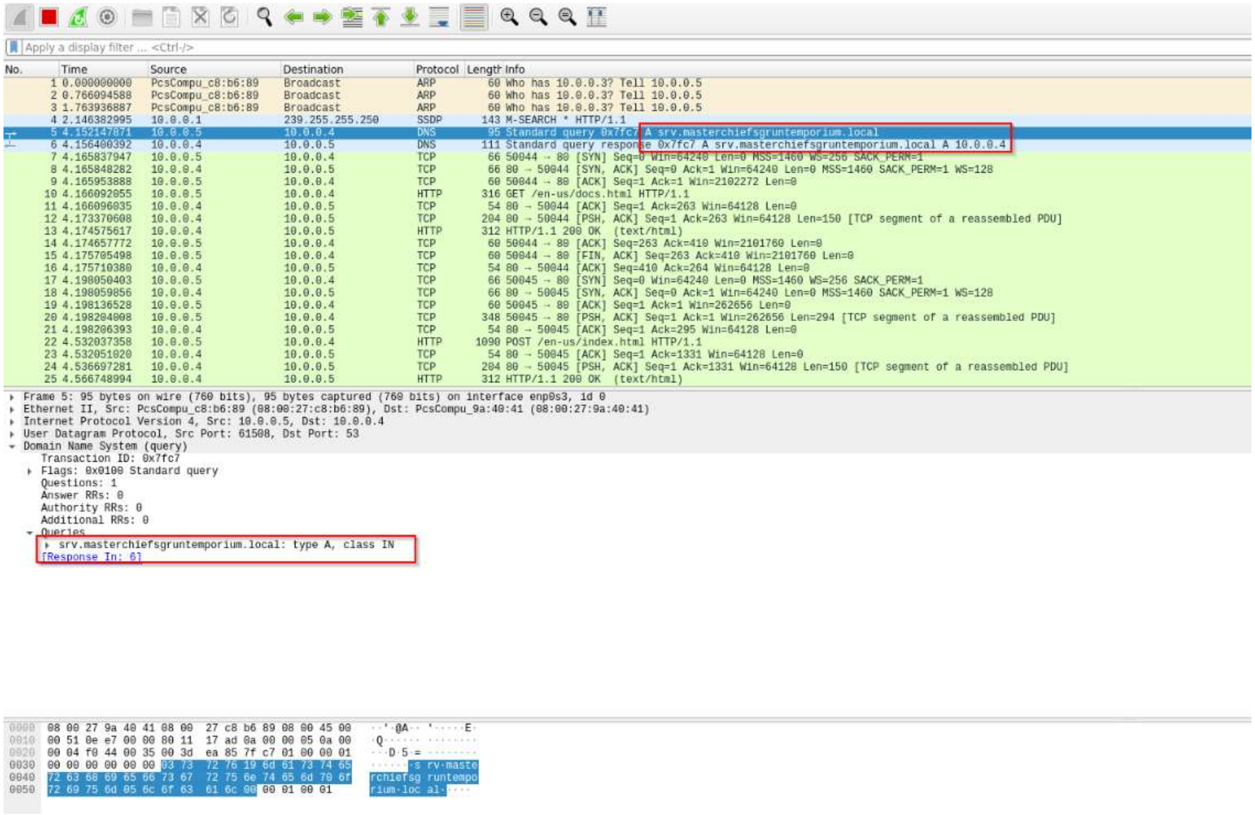
```
[0x18000238a]
 0x18000238a      add     byte [rax], al
 0x18000238c      and     edx, dword [arg_53h] ; arg2
 0x18000238f      add     byte [rax], cl
 0x180002391      pop     rdi
 0x180002392      add     byte [rax], al
 0x180002394      adc     byte [rax], al
 0x180002396      add     byte [rax], al
 0x180002398      and     eax, dword [rdi + 0x55]
 0x18000239b      add     byte [rax], r8b
 0x18000239f      add     byte [rax], bl
 0x1800023a1      pop     rdi
 0x1800023a2      add     byte [rax], al
 0x1800023a4      int3
```

Indicators of Compromise

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

Network Indicators

{Description of network indicators}



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

```
No.  Time         Source             Destination      Protocol Length Info
   1 0.000000000  PcsCompu_c8:b6:89  Broadcast        ARP        60 Who has 10.0.0.3? Tell 10.0.0.5
   2 0.766094588  PcsCompu_c8:b6:89  Broadcast        ARP        60 Who has 10.0.0.3? Tell 10.0.0.5
   3 1.763936887  PcsCompu_c8:b6:89  Broadcast        ARP        60 Who has 10.0.0.3? Tell 10.0.0.5
   4 2.146382995  10.0.0.1           239.255.255.250  SSDP      143 M-SEARCH * HTTP/1.1
   5 4.152147871  10.0.0.5           10.0.0.4         DNS        95 Standard query 0x7fc7 A srv.masterchiefsgruntemporium.local
   6 4.156400392  10.0.0.4           10.0.0.5         DNS       111 Standard query response 0x7fc7 A srv.masterchiefsgruntemporium.local A 10.0.0.4
   7 4.165837947  10.0.0.5           10.0.0.4         TCP        66 50044 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   8 4.165848282  10.0.0.4           10.0.0.5         TCP        66 80 → 50044 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
   9 4.165953888  10.0.0.5           10.0.0.4         TCP        60 50044 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
  10 4.166092055  10.0.0.5           10.0.0.4         HTTP      316 GET /en-us/docs.html HTTP/1.1
  11 4.166096035  10.0.0.4           10.0.0.5         TCP        54 80 → 50044 [ACK] Seq=1 Ack=263 Win=64128 Len=0
  12 4.173370608  10.0.0.4           10.0.0.5         TCP       204 80 → 50044 [PSH, ACK] Seq=1 Ack=263 Win=64128 Len=150 [TCP segment of a reassembled PDU]
  13 4.174575617  10.0.0.4           10.0.0.5         HTTP      312 HTTP/1.1 200 OK  (text/html)
  14 4.174657772  10.0.0.5           10.0.0.4         TCP        60 50044 → 80 [ACK] Seq=263 Ack=410 Win=2101760 Len=0
  15 4.175705498  10.0.0.5           10.0.0.4         TCP        60 50044 → 80 [FIN, ACK] Seq=263 Ack=410 Win=2101760 Len=0
  16 4.175710380  10.0.0.4           10.0.0.5         TCP        54 80 → 50044 [ACK] Seq=410 Ack=264 Win=64128 Len=0
  17 4.198050403  10.0.0.5           10.0.0.4         TCP        66 50045 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  18 4.198059856  10.0.0.4           10.0.0.5         TCP        66 80 → 50045 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
  19 4.198136528  10.0.0.5           10.0.0.4         TCP        60 50045 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
  20 4.198204008  10.0.0.5           10.0.0.4         TCP       348 50045 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262656 Len=294 [TCP segment of a reassembled PDU]
  21 4.198206393  10.0.0.4           10.0.0.5         TCP        54 80 → 50045 [ACK] Seq=1 Ack=295 Win=64128 Len=0
  22 4.532037358  10.0.0.5           10.0.0.4         HTTP     1090 POST /en-us/index.html HTTP/1.1
  23 4.532051020  10.0.0.4           10.0.0.5         TCP        54 80 → 50045 [ACK] Seq=1 Ack=1331 Win=64128 Len=0
  24 4.536697281  10.0.0.4           10.0.0.5         TCP       204 80 → 50045 [PSH, ACK] Seq=1 Ack=1331 Win=64128 Len=150 [TCP segment of a reassembled PDU]
  25 4.566748994  10.0.0.4           10.0.0.5         HTTP      312 HTTP/1.1 200 OK  (text/html)
```

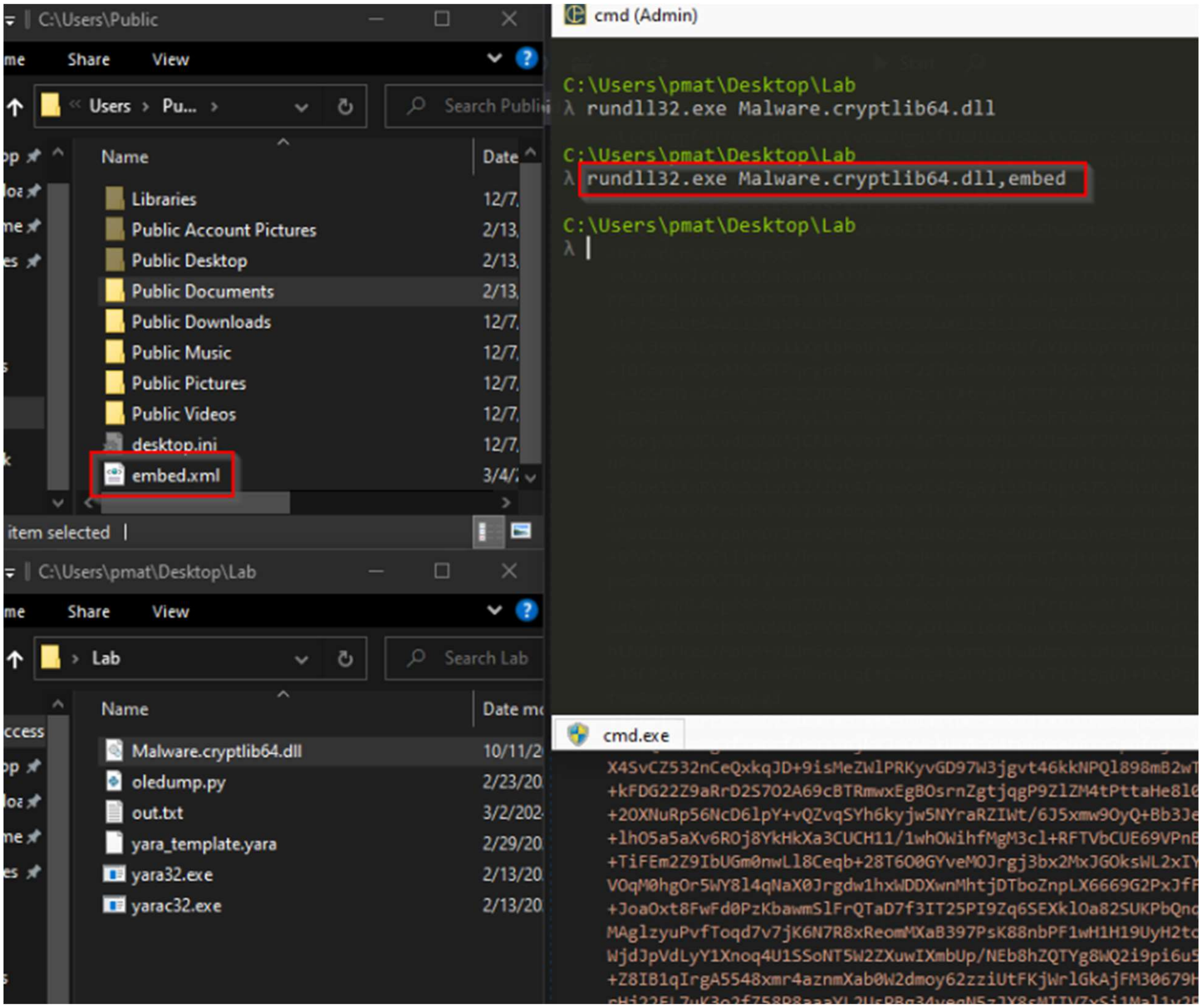
```
▸ Frame 5: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: PcsCompu_c8:b6:89 (08:00:27:c8:b6:89), Dst: PcsCompu_9a:40:41 (08:00:27:9a:40:41)
▸ Internet Protocol Version 4, Src: 10.0.0.5, Dst: 10.0.0.4
▸ User Datagram Protocol, Src Port: 61508, Dst Port: 53
▾ Domain Name System (query)
    Transaction ID: 0x7fc7
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▸ srv.masterchiefsgruntemporium.local: type A, class IN
    [Response In: 6]
```

```
0000  08 00 27 9a 40 41 08 00  27 c8 b6 89 08 00 45 00   ··'·@A·· '·····E·
0010  00 51 0e e7 00 00 80 11  17 ad 0a 00 00 05 0a 00   ·Q······ ········
0020  00 04 f0 44 00 35 00 3d  ea 85 7f c7 01 00 00 01   ···D·5·= ········
0030  00 00 00 00 00 00 03 73  72 76 19 6d 61 73 74 65   ·······s rv·maste
0040  72 63 68 69 65 66 73 67  72 75 6e 74 65 6d 70 6f   rchiefsg runtempo
0050  72 69 75 6d 05 6c 6f 63  61 6c 00 00 01 00 01      rium·loc al·····
```

Fig 3: WireShark Packet Capture of initial beacon check-in

Host-based Indicators

{Description of host-based indicators}

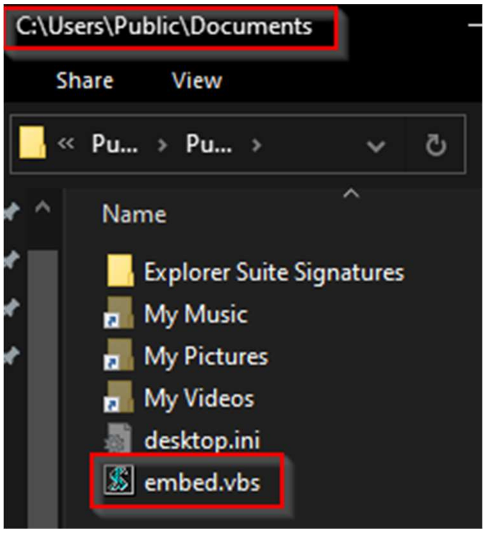


Name	Date
Libraries	12/7,
Public Account Pictures	2/13,
Public Desktop	2/13,
Public Documents	2/13,
Public Downloads	12/7,
Public Music	12/7,
Public Pictures	12/7,
Public Videos	12/7,
desktop.ini	12/7,
embed.xml	3/4/:

Name	Date mo
Malware.cryptlib64.dll	10/11/2
oledump.py	2/23/20
out.txt	3/2/202
yara_template.yara	2/29/20
yara32.exe	2/13/20
yarac32.exe	2/13/20

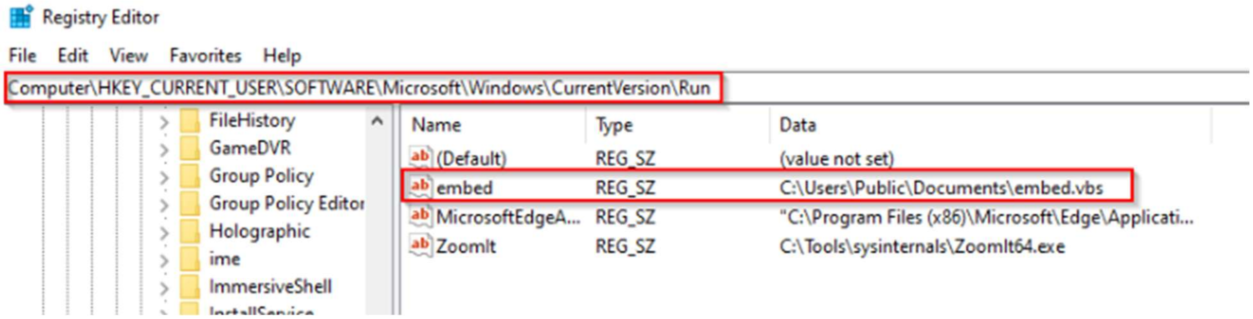
```
C:\Users\pmat\Desktop\Lab
λ rundll32.exe Malware.cryptlib64.dll

C:\Users\pmat\Desktop\Lab
λ rundll32.exe Malware.cryptlib64.dll,embed

C:\Users\pmat\Desktop\Lab
λ
```

```
X4SvCZ532nCeQxkqJD+9isMeZWlPRKyvGD97W3jgvt46kkNPQl898mB2wT
+kFDG22Z9aRrD2S7O2A69cBTRmwxEgBOsrnZgtjqgP9ZlZM4tPttaHe8l0
+2OXNuRp56NcD6lpY+vQZvqSYh6kyjw5NYraRZIWt/6J5xmw9OyQ+Bb3Je
+lhO5a5aXv6ROj8YkHkXa3CUCH11/1whOWihfMgM3cl+RFTVbCUE69VPnE
+TiFEm2Z9IbUGm0nwLl8Ceqb+28T6O0GYveMOJrgj3bx2MxJGOksWL2xIY
VOqM0hgOr5WY8l4qNaX0Jrgdw1hxWDDXwnMhtjDTboZnpLX6669G2PxJfF
+JoaOxt8FwFd0PzKbawmSlFrQTaD7f3IT25PI9Zq6SEXklOa82SUKPbQnc
MAglzyuPvfToqd7v7jK6N7R8xReomMXaB397PsK88nbPF1wH1H19UyH2tc
WjdJpVdLyY1Xnoq4U1SSoNT5W2ZXuwIXmbUp/NEb8hZQTYg8WQ2i9pi6uS
+Z8IB1qIrgA5548xmr4aznmXab0W2dmoy62zziUtFKjWrlGkAjFM30679H
```



C:\Users\Public\Documents

```
Explorer Suite Signatures
My Music
My Pictures
My Videos
desktop.ini
embed.vbs
```



Computer\HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Name	Type	Data
(Default)	REG_SZ	(value not set)
embed	REG_SZ	C:\Users\Public\Documents\embed.vbs
MicrosoftEdgeA...	REG_SZ	"C:\Program Files (x86)\Microsoft\Edge\Applicati...
ZoomIt	REG_SZ	C:\Tools\sysinternals\ZoomIt64.exe

Rules & Signatures

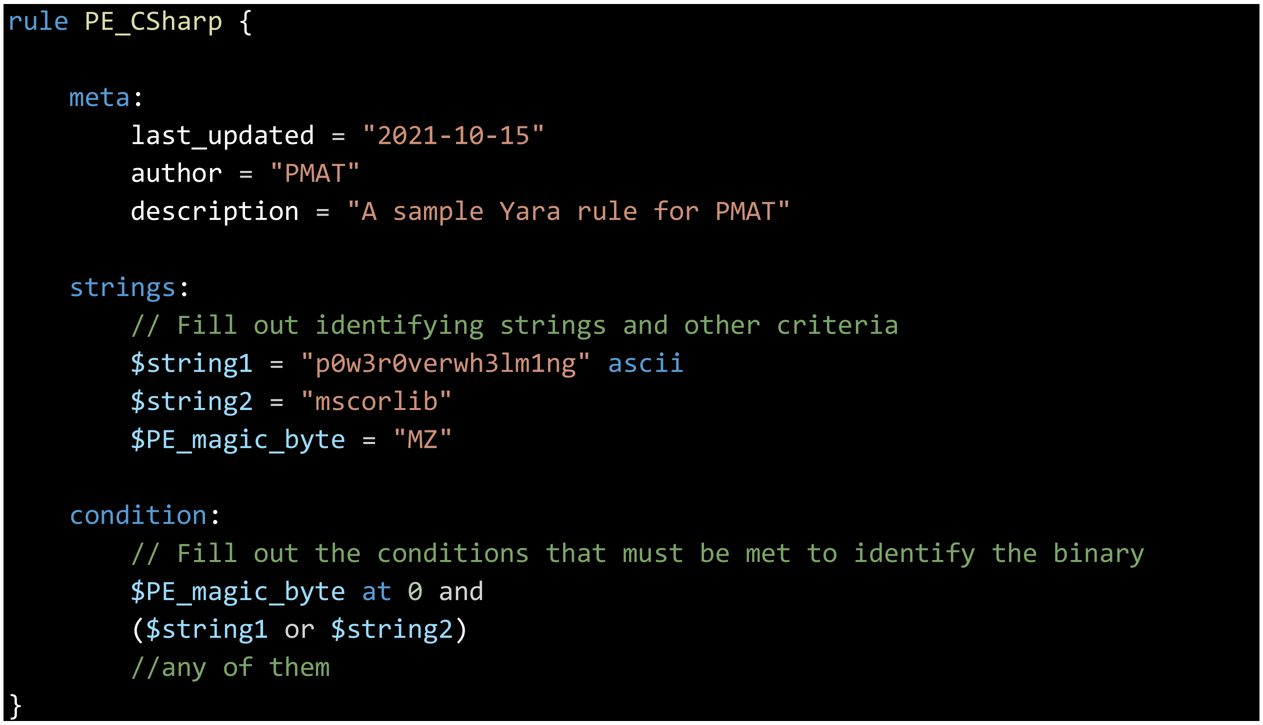
A full set of YARA rules is included in Appendix A.

{Information on specific signatures, i.e. strings, URLs, etc}

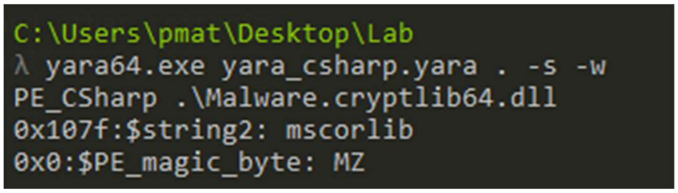
Appendices

A. Yara Rules

Full Yara repository located at: http://github.com/te0001hi/pmat_lab



```
rule PE_CSharp {

    meta:
        last_updated = "2021-10-15"
        author = "PMAT"
        description = "A sample Yara rule for PMAT"

    strings:
        // Fill out identifying strings and other criteria
        $string1 = "p0w3r0verwh3lm1ng" ascii
        $string2 = "mscorlib"
        $PE_magic_byte = "MZ"

    condition:
        // Fill out the conditions that must be met to identify the binary
        $PE_magic_byte at 0 and
        ($string1 or $string2)
        //any of them
}
```



```
C:\Users\pmat\Desktop\Lab
λ yara64.exe yara_csharp.yara . -s -w
PE_CSharp .\Malware.cryptlib64.dll
0x107f:$string2: mscorlib
0x0:$PE_magic_byte: MZ
```

B. Callback URLs

Domain	Port
hxxp:// srv.masterchiefsgruntemporium.local	80