

NSPX30

A sophisticated AitM-enabled implant
evolving since 2005

Facundo Munoz

Malware Researcher



Digital Security
Progress. Protected.



Facundo Munoz

- Malware Researcher at ESET since 2021
- Hunting and analysing APT malware

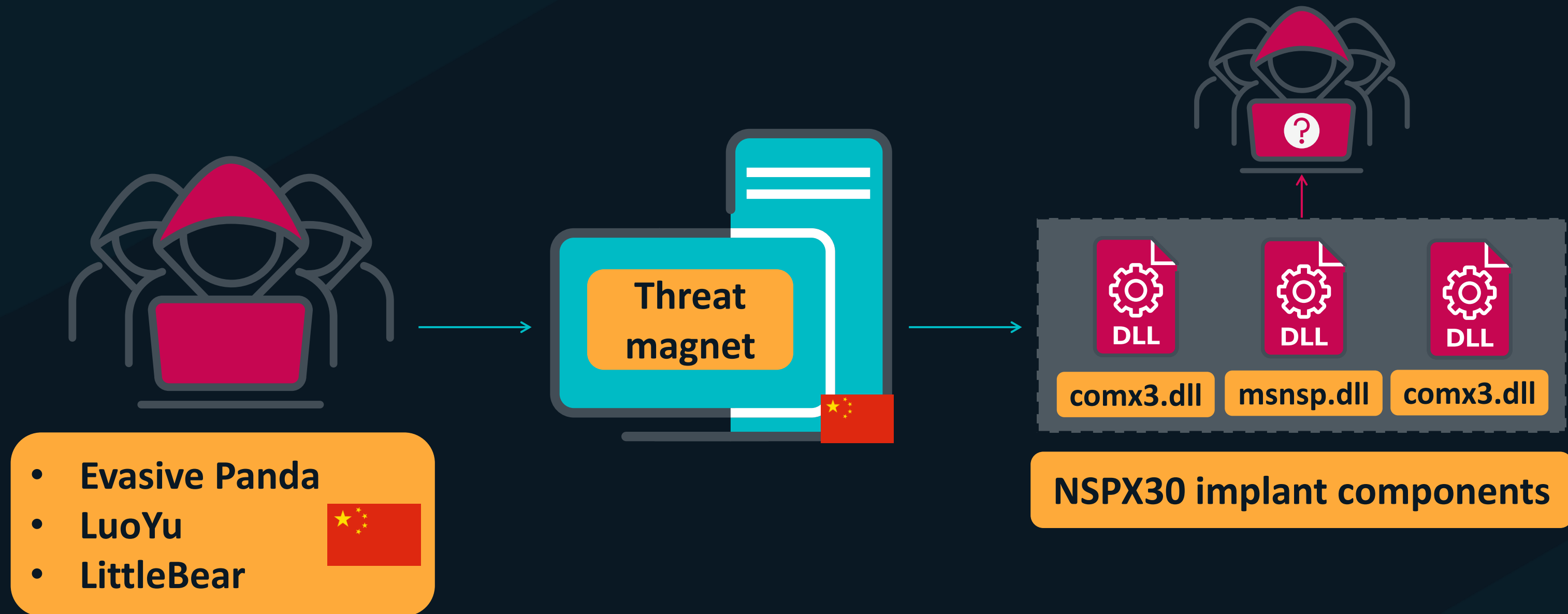


facundo.munoz@eset.com

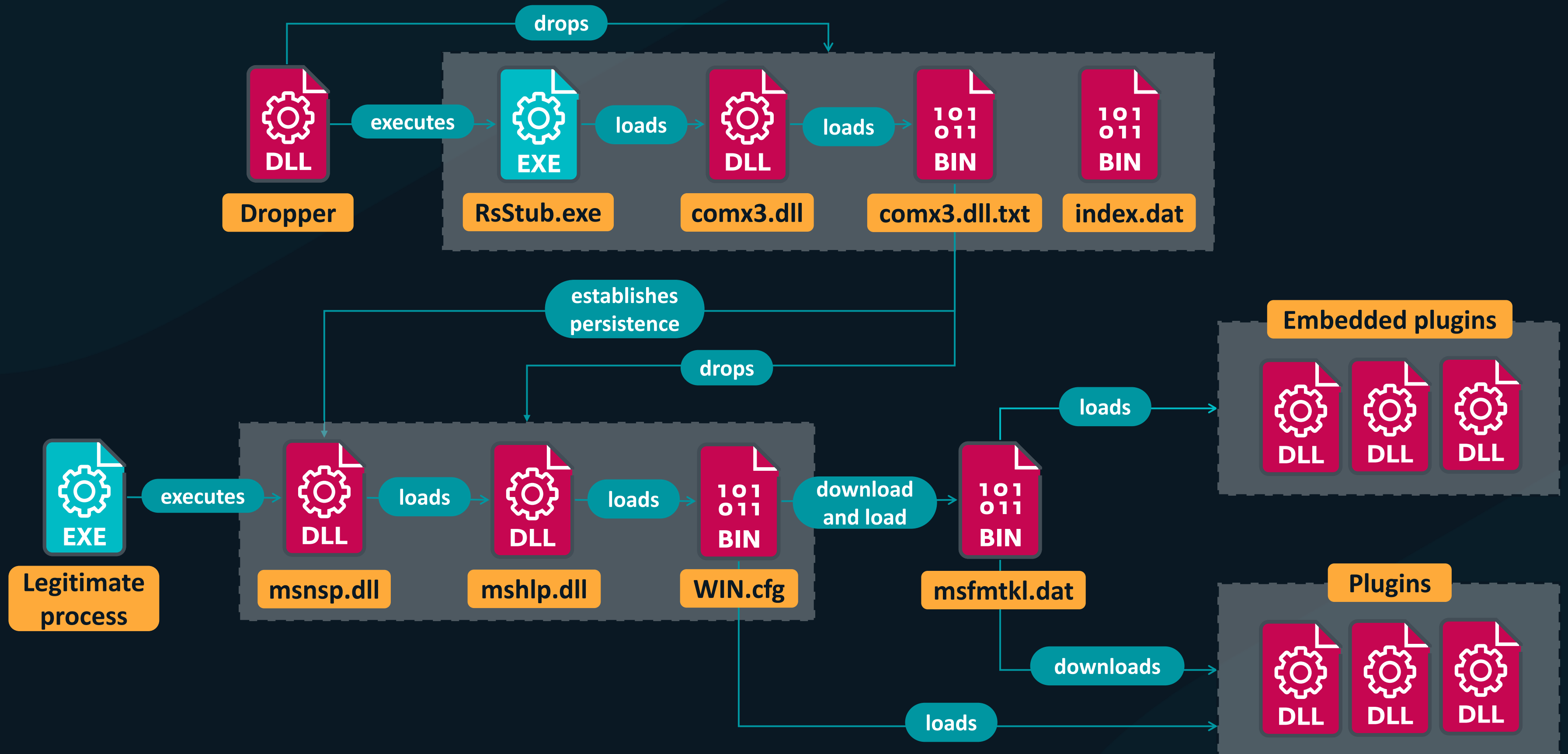
Agenda

- ✔ Discovery of the NSPX30 implant
- ✔ Our research, evolution of NSPX30 and attribution to Blackwood APT
- ✔ Case study of an initial access via AitM
- ✔ The NSPX30 implant and its features
- ✔ The China-aligned AitM club
- ✔ Conclusion

In the beginning there were... *many detections*



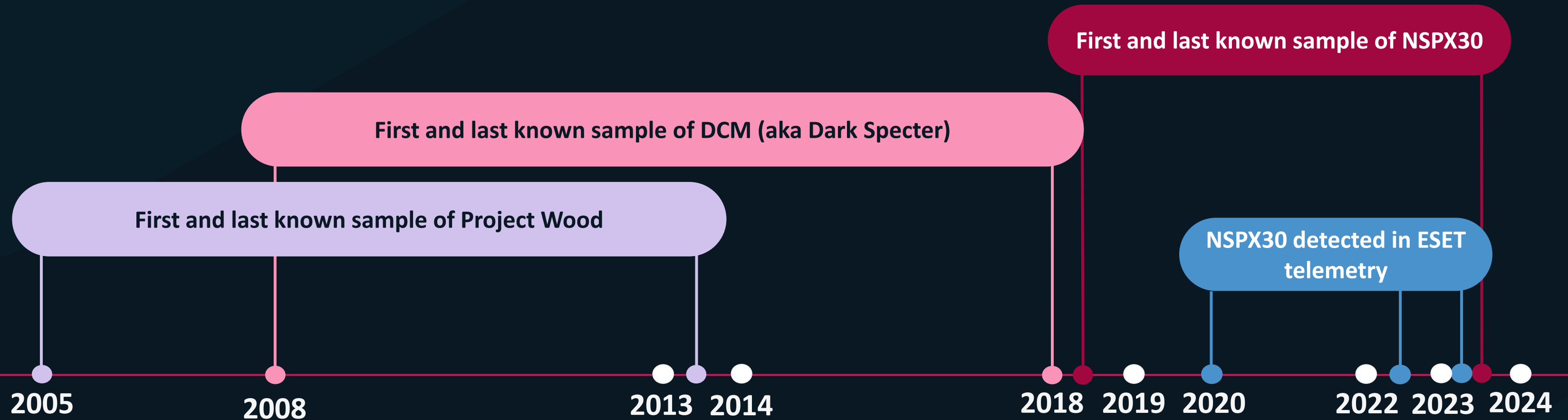
The NSPX30 implant





Malware paleontology

Timeline of NSPX30 and its ancestors



Compilation timestamps, to trust or not to trust



**Project Wood backdoor
MainFuncOften.dll**

**Compiled on
2005-01-09 08:21:22**



**Project Wood dropper
Unknown file name ☹️**

**Compiled on
2005-01-09 08:21:39**

Compilation timestamps, to trust or not to trust



Project Wood backdoor
MainFuncOften.dll

Compiled on
2005-01-09 08:21:22



Project Wood dropper
Unknown file name ☹️

Compiled on
2005-01-09 08:21:39

17 seconds difference!

UPX version

Dropper



000003D0	00	00	00	00	00	00	00	00	00	00	00	00	31	2E	32	34	00	1.24.
000003E0	55	50	58	21	0C	09	05	09	C6	BF	B8	96	B6	DB	30	81	UPX!.....ç,-ŶÛ0..	
000003F0	E7	39	04	00	0D	48	01	00	00	20	04	00	26	0A	00	29	ç9...H.....&..)	
00000400	7F	FF	FF	FF	51	53	55	56	8B	35	1C	C0	40	00	8D	44	.ÿÿÿQSUV<5.À@..D	
00000410	24	0C	57	50	68	06	00	02	00	BD	B4	E0	1F	6A	00	55	\$.WPh.....½´à.j.U	
00000420	68	ED	B7	ED	7F	15	00	80	FF	D6	8B	3D	20	3E	68	AC	hí·í...€ÿÖ<=.>h-	
00000430	28	FF	74	24	14	FF	D7	0A	10	91	B7	B5	ED	8B	1D	24	(ÿt\$.ÿ×..´·μí<.\$	

upx124w.zip	2003-04-22	126.0 kB
upx124d.zip	2003-04-22	185.9 kB
upx124a.zip	2003-04-22	333.2 kB
upx-1.24-src.tar.gz	2003-04-22	223.0 kB
upx-1.24-linux.tar.gz	2003-04-22	156.4 kB

UPX 1.24 was released in 2003

Rich header metadata

Offset	Name	Value	Unmasked Value	Meaning	ProductId	BuildId	Count	VS version
80	DanS ID	241ad14	536e6144	DanS				
84	Checksum...	512fcc50	0	0				
88	Checksum...	512fcc50	0	0				
8C	Checksum...	512fcc50	0	0				
90	Comp ID	512fcc53512fcc50	300000000	0.0.3	Unknown	0	3	
98	Comp ID	512fcc575123d02b	7000c1c7b	7291.12.7	AliasObj60	7291	7	Visual Studio 97 05.00
A0	Comp ID	512fcc4b5121d0d3	1b000e1c83	7299.14.27	Masm613	7299	27	Visual Studio 97 05.00
A8	Comp ID	512fcc41513cd332	1100131f62	8034.19.17	Linker512	8034	17	
B0	Comp ID	512fcc86512ecc50	d600010000	0.1.214	Import0	0	214	Visual Studio
B8	Comp ID	512fcc4f5124ea66	1f000b2636	9782.11.31	Utc12_CPP	9782	31	Visual Studio 6.0 06.00
C0	Comp ID	512fcc295125ea66	79000a2636	9782.10.121	Utc12_C	9782	121	Visual Studio 6.0 06.00
C8	Comp ID	512fcc51512becaf	1000420ff	8447.4.1	Linker600	8447	1	
D0	Rich ID	68636952		Rich				
D4	Checksum	512fcc50		512fcc50				

Visual Studio 6.0 was released in 1998

Assessment: high confidence that is unlikely that attackers modified all these indicators.

Project Wood aka PeerWoodCOften

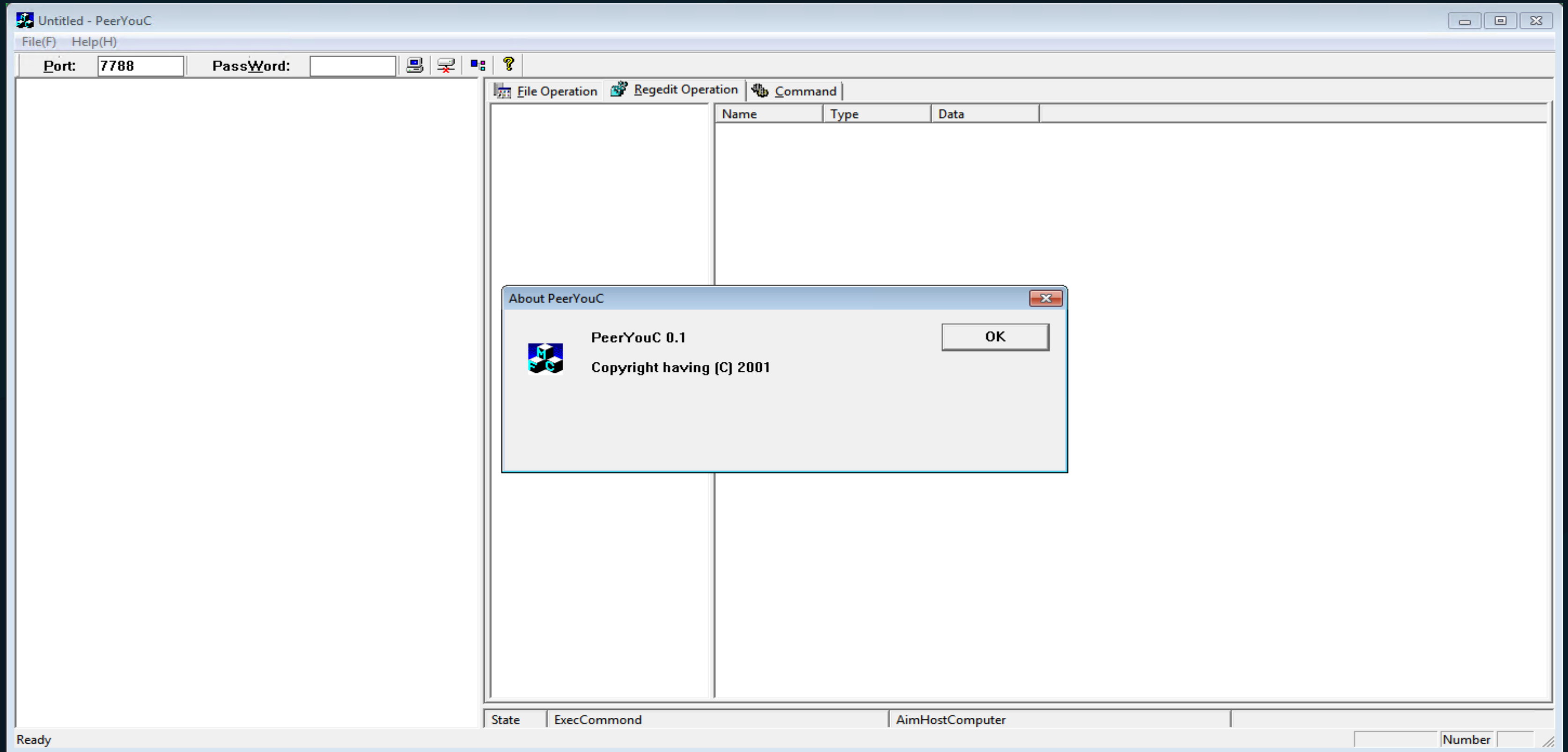
Dropper



00000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0E	00
00000010	50	00	65	00	65	00	72	00	57	00	6F	00	6F	00	64	00
00000020	43	00	4F	00	66	00	74	00	65	00	6E	00	00	00	00	00
00000030	0C	00	48	00	65	00	6C	00	6C	00	6F	00	20	00	57	00
00000040	6F	00	72	00	6C	00	64	00	21	00	00	00	00	00	0E	00
00000050	50	00	45	00	45	00	52	00	57	00	4F	00	4F	00	44	00
00000060	43	00	4F	00	46	00	54	00	45	00	4E	00	00	00	00	00

.....
P.e.e.r.W.o.o.d.
C.O.f.t.e.n.....
..H.e.l.l.o...W.
o.r.l.d.!.....
P.E.E.R.W.O.O.D.
C.O.F.T.E.N.....

PeerYou RAT



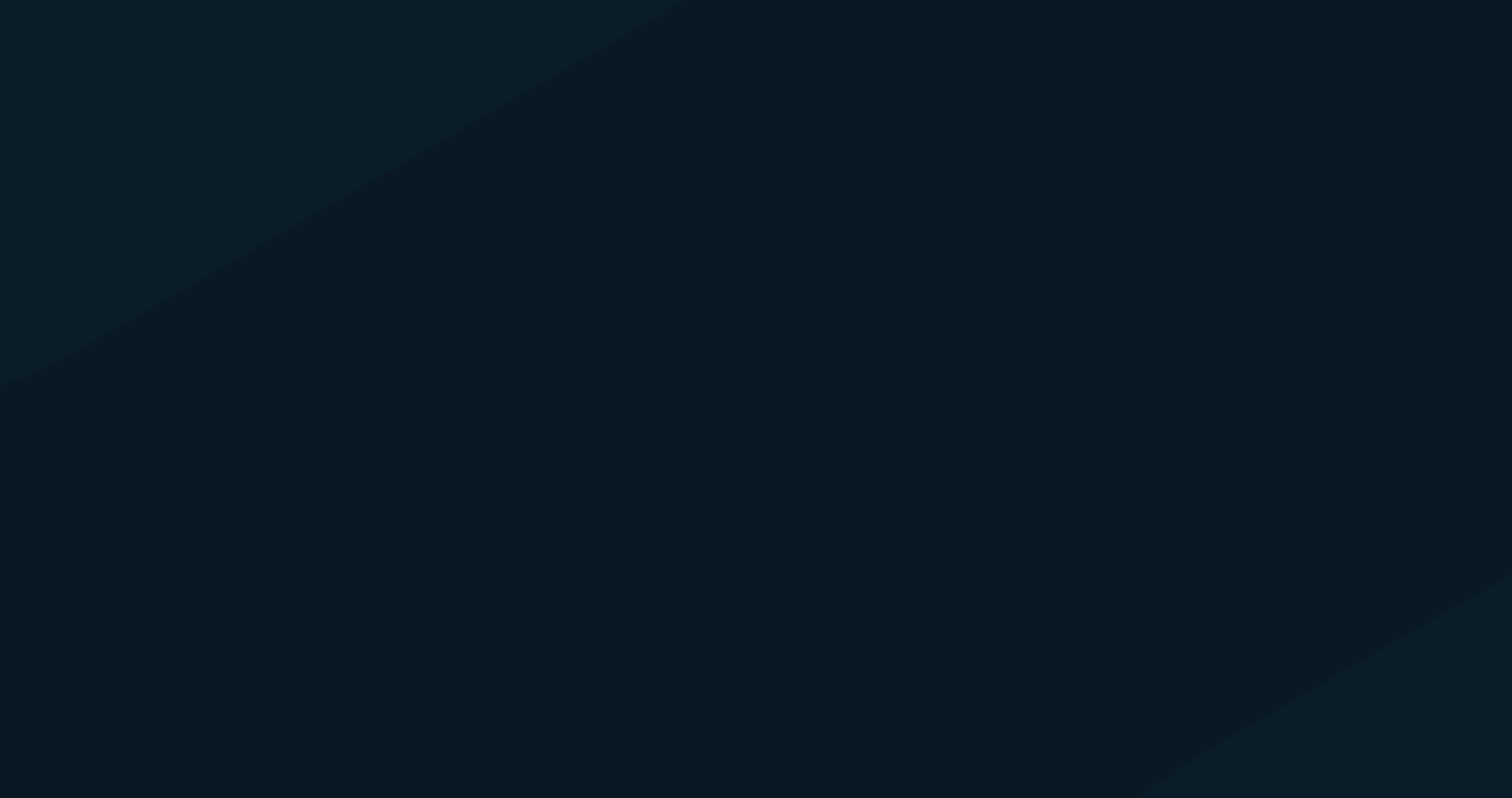
Open sourced in 2001

Possibly of Chinese origin

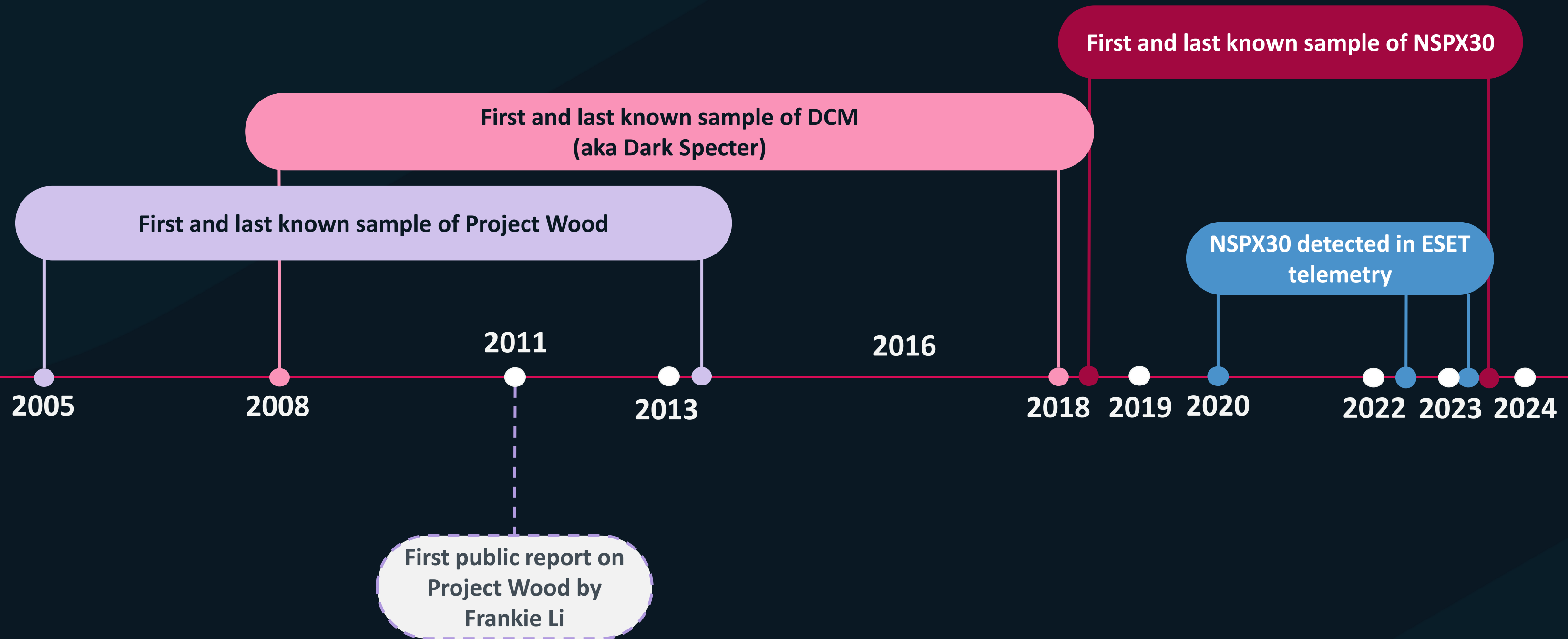
Still around in Chinese websites

Many variants

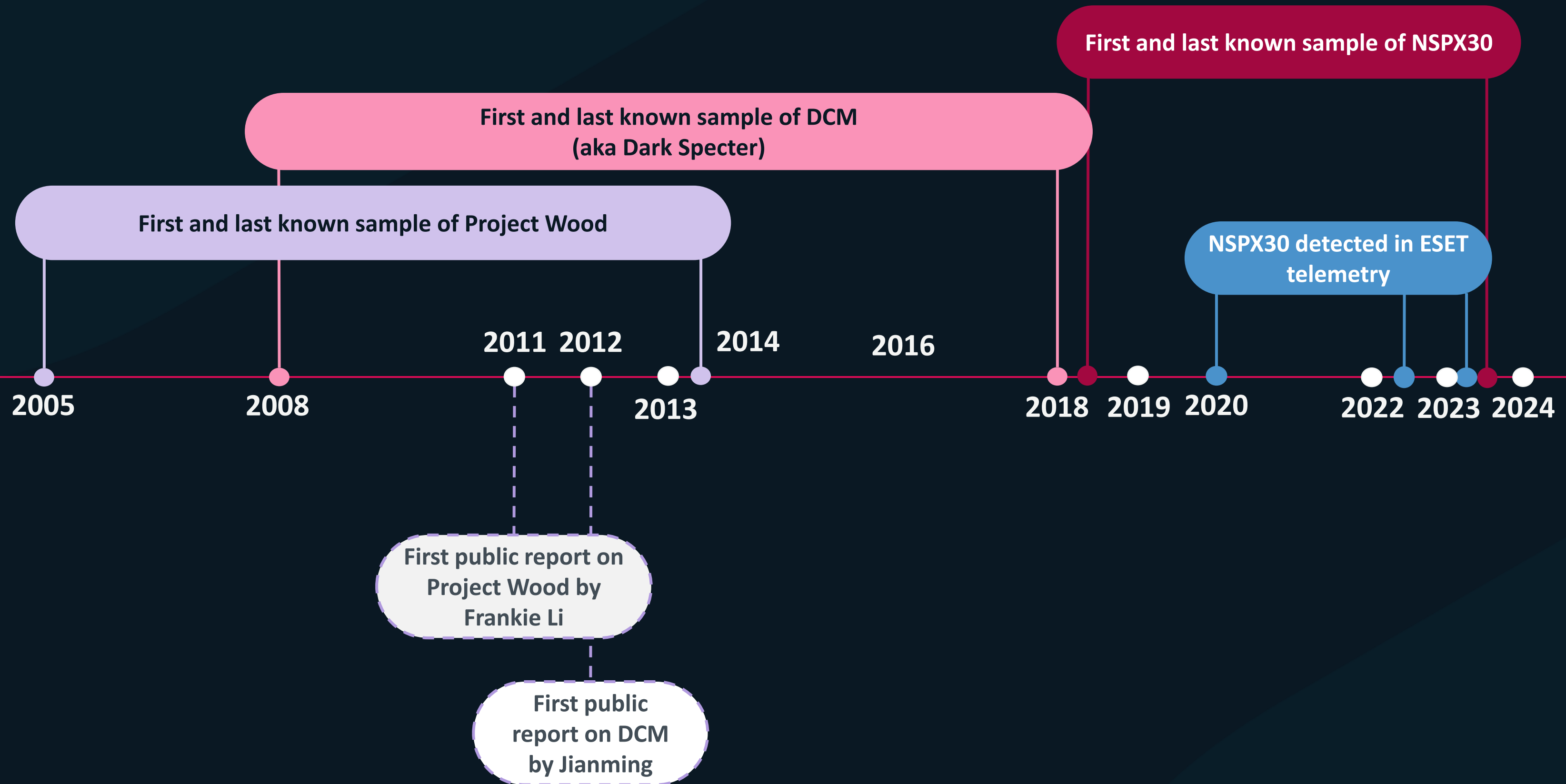
Timeline of publications about NSPX30's ancestors



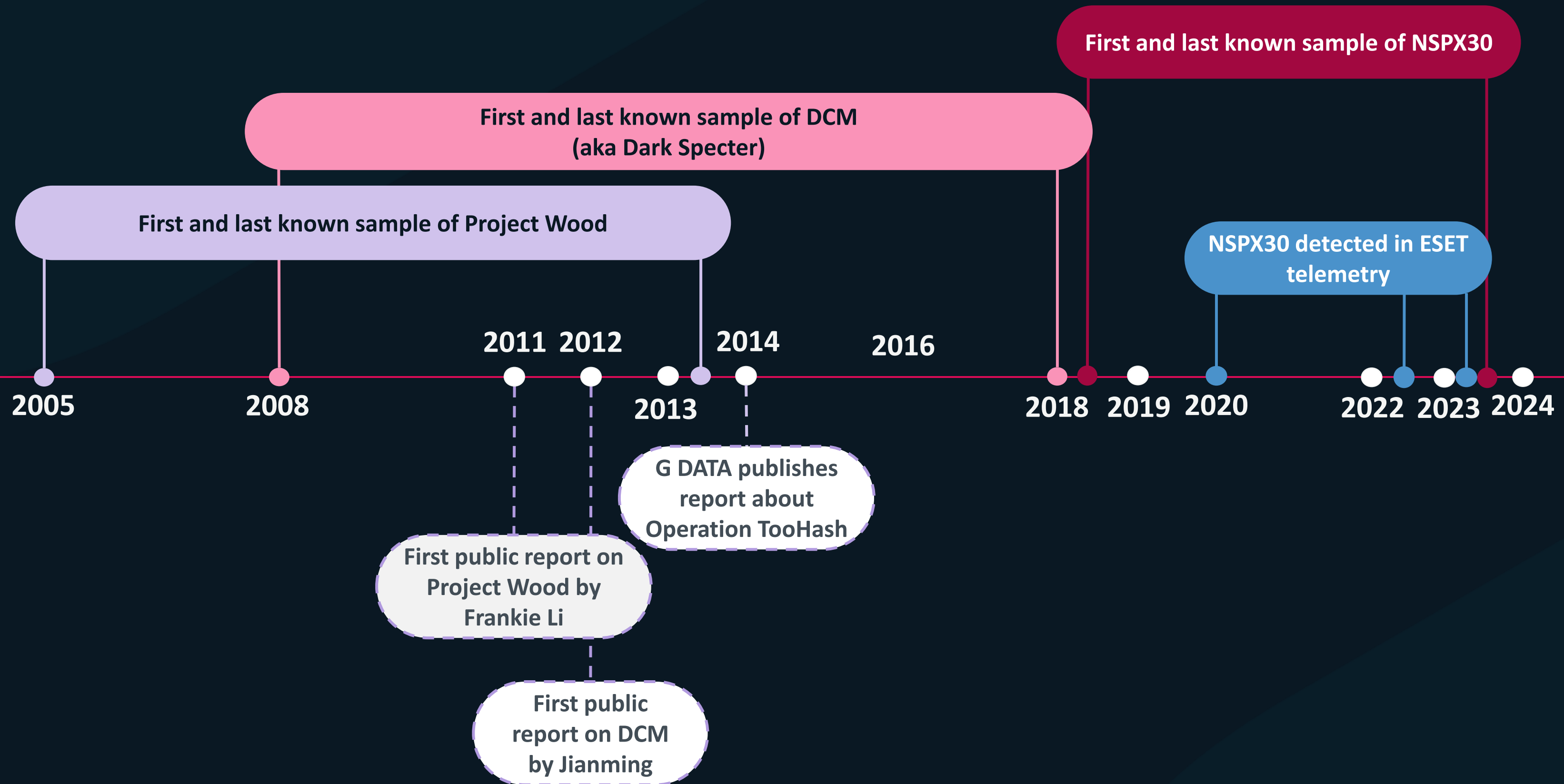
Timeline of publications about NSPX30's ancestors



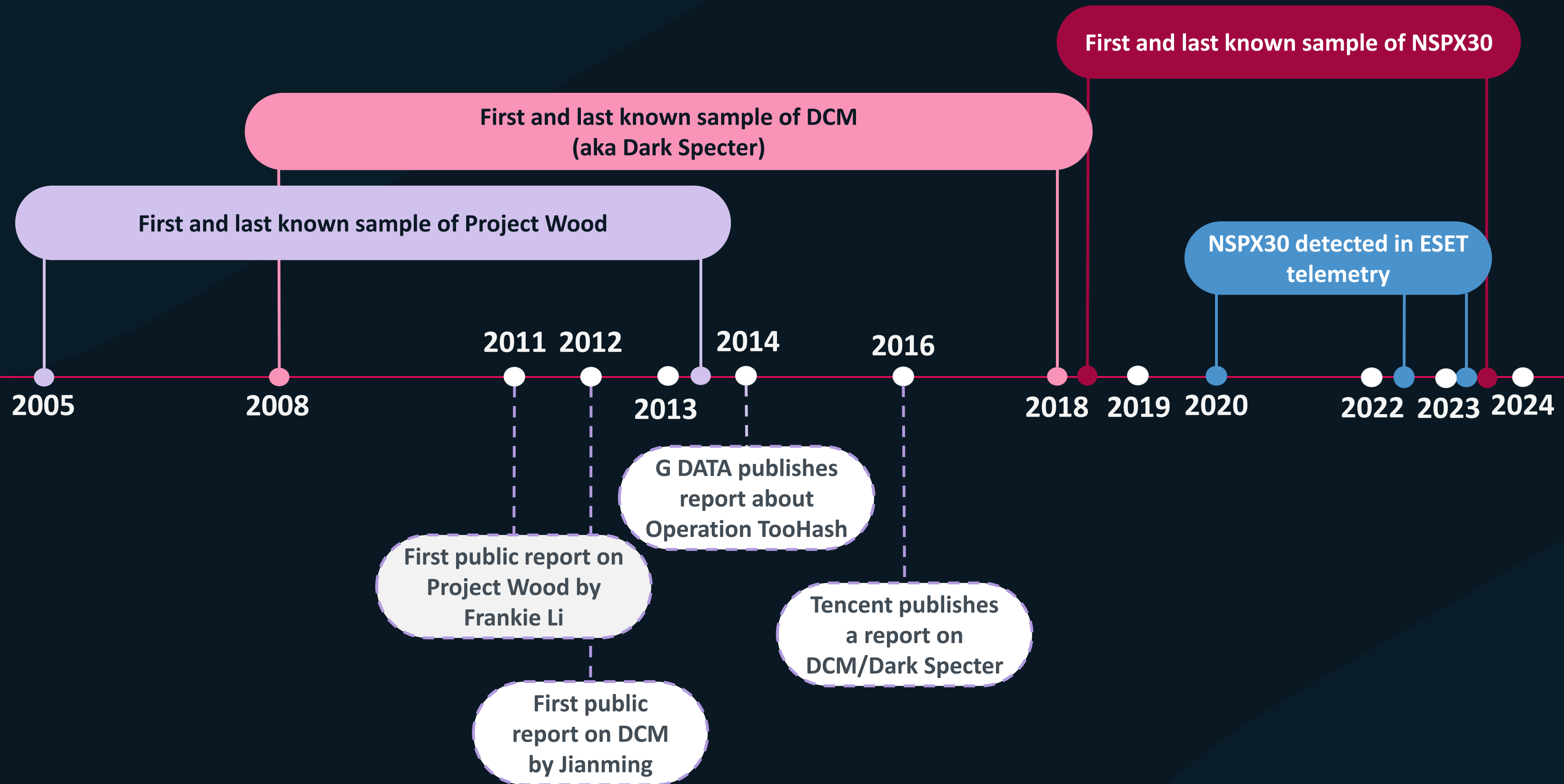
Timeline of publications about NSPX30's ancestors



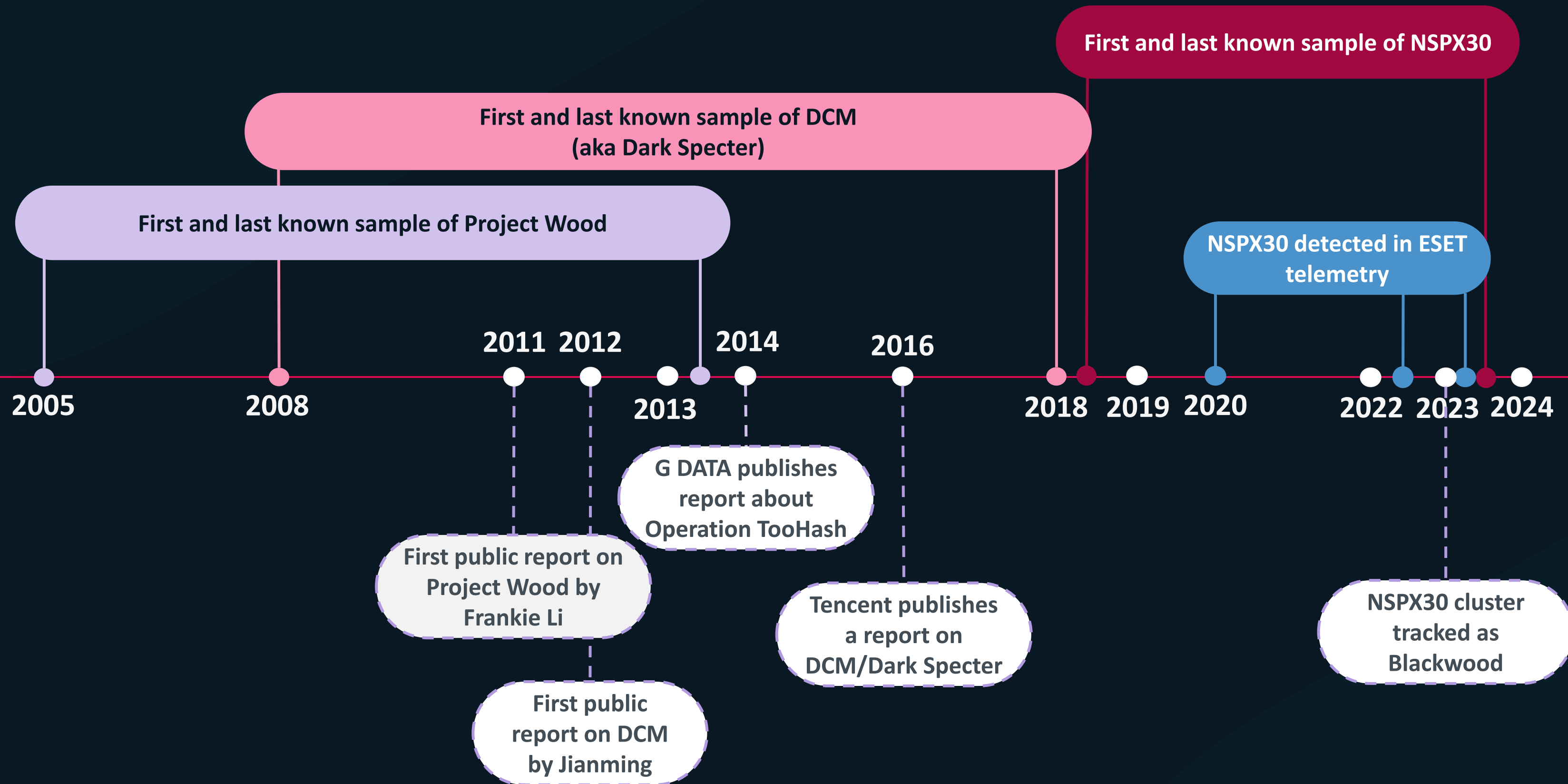
Timeline of publications about NSPX30's ancestors



Timeline of publications about NSPX30's ancestors



Timeline of publications about NSPX30's ancestors



Blackwood profile



Targeted regions

Japan



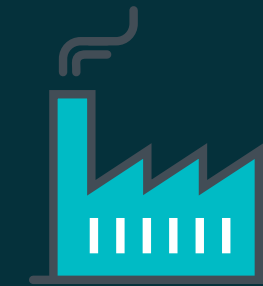
China



United Kingdom



Sectors

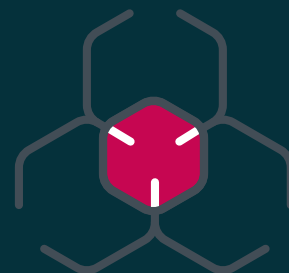


Manufacturing



Individuals

Toolkit



NSPX30

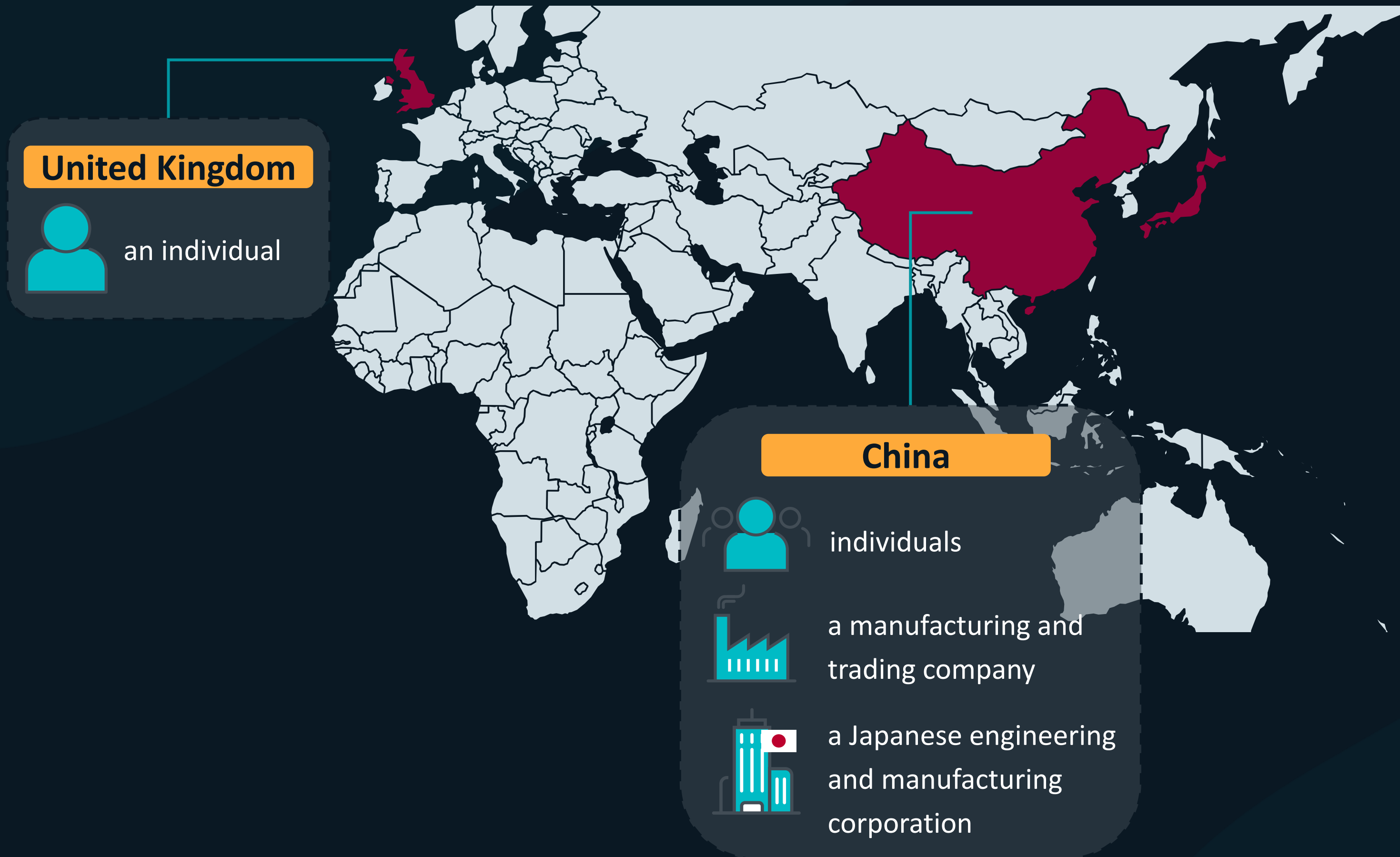


AitM

Geographical distribution of NSPX30 victims



Geographical distribution of NSPX30 victims

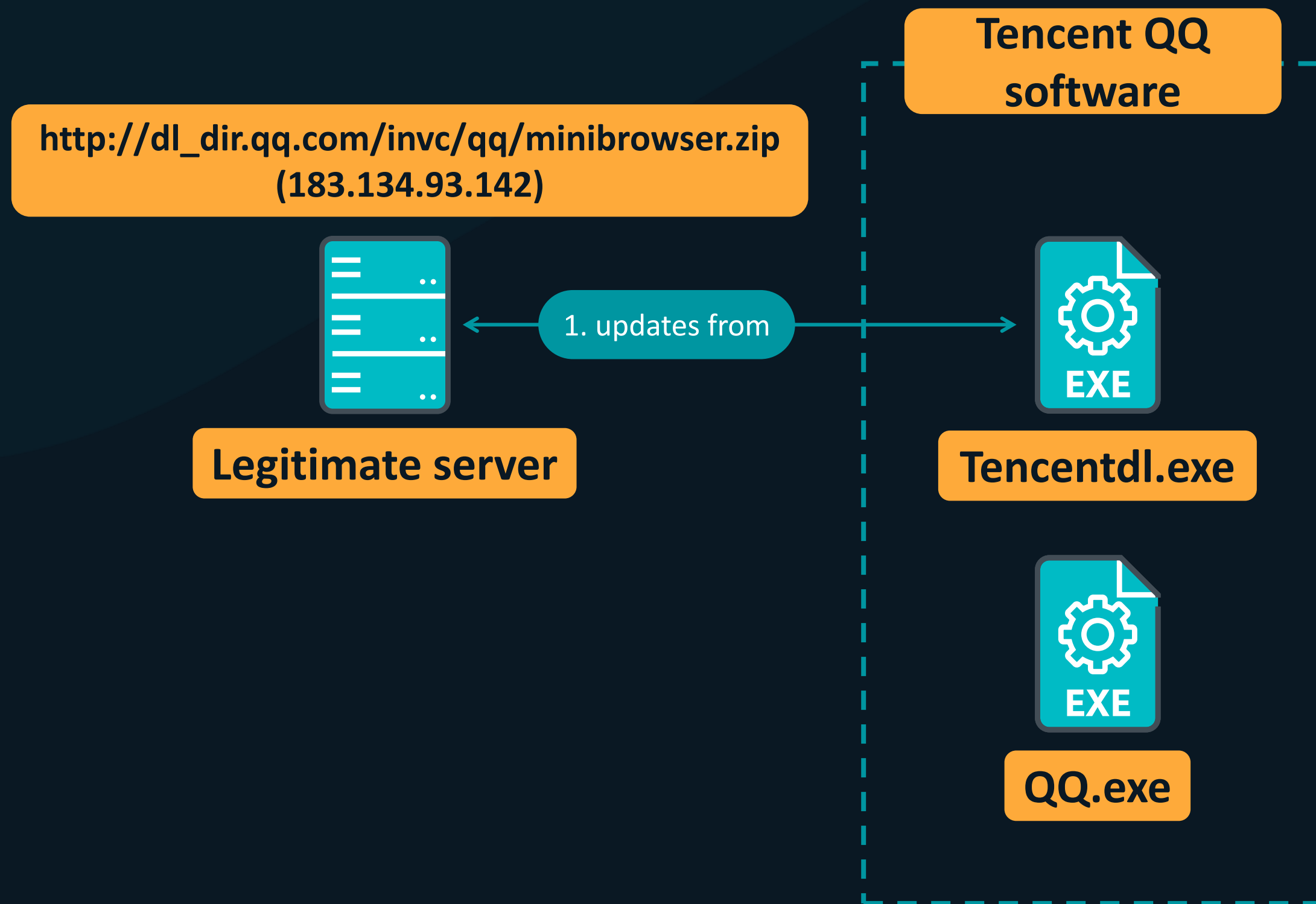


Geographical distribution of NSPX30 victims

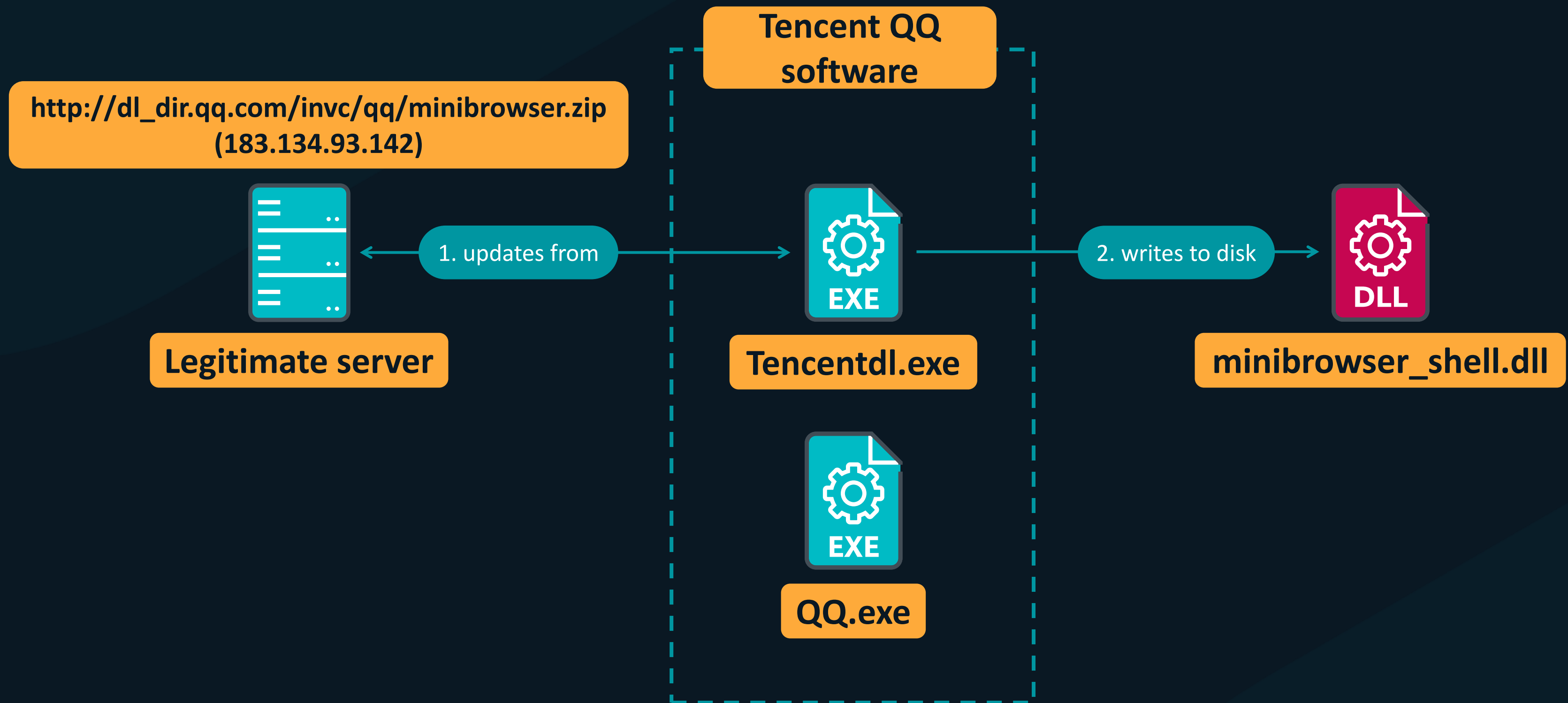


How Blackwood uses AitM?

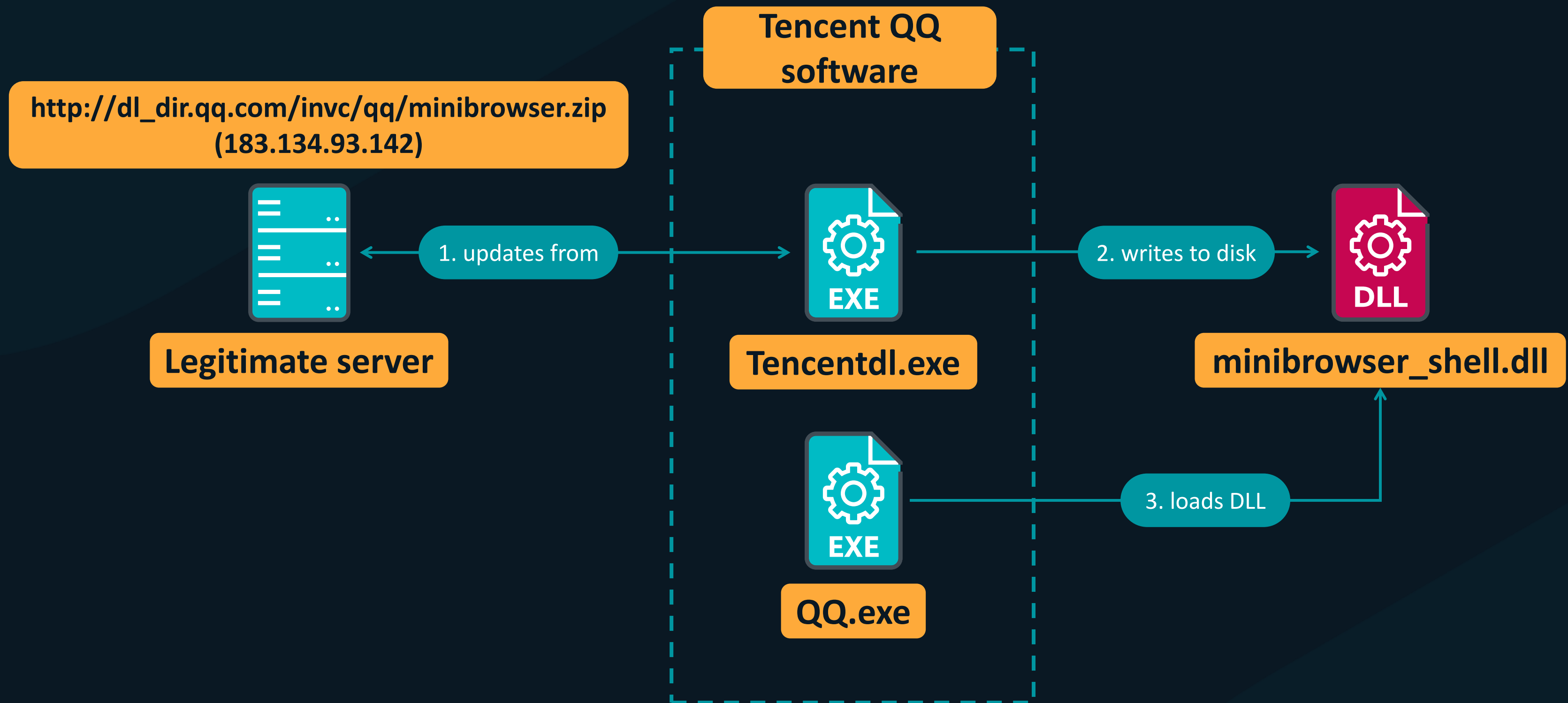
What we observed on victim machines



What we observed on victim machines



What we observed on victim machines



Successfully hijacked software updates by Blackwood and others!



Tencent QQ



Sogou Pinyin



WPS Office

Successfully hij...



Windows

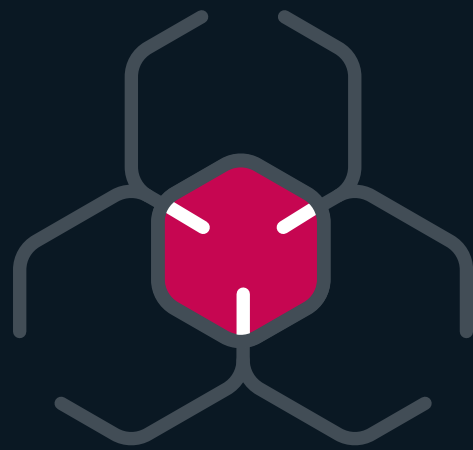




MISSION FAILURE



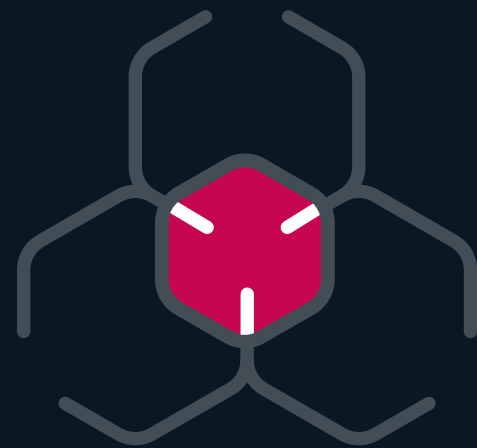
What we learned from the attacks



Initial access

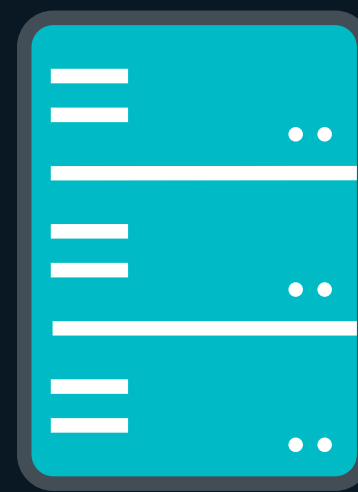
NSPX30 dropper downloaded through **HTTP** by legitimate software.

What we learned from the attacks



Initial access

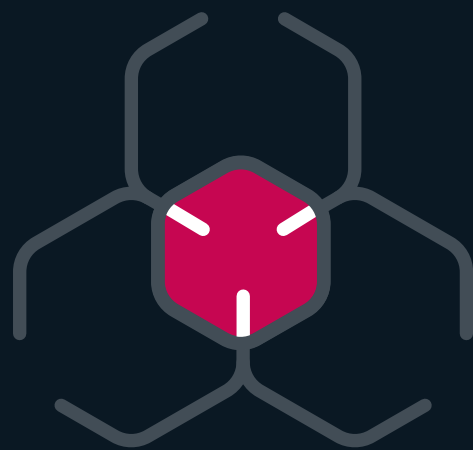
NSPX30 dropper downloaded through **HTTP** by legitimate software.



Legitimate servers

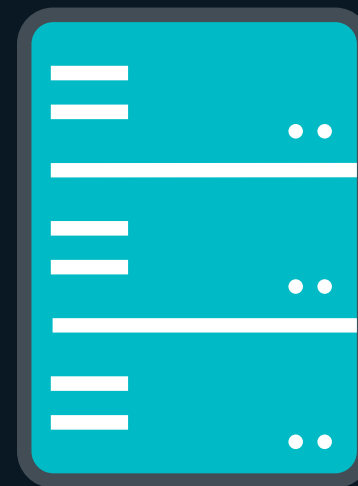
IP address associated to domains were from legitimate infrastructure.

What we learned from the attacks



Initial access

NSPX30 dropper downloaded through **HTTP** by legitimate software.



Legitimate servers

IP address associated to domains were from legitimate infrastructure.

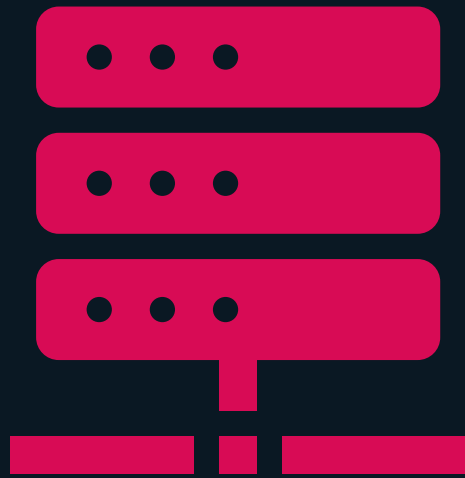


Flexible dropper

NSPX30 dropper can be DLL/EXE, if required: in ZIP archive.

The NSPX30 and its design

NSPX30 main features

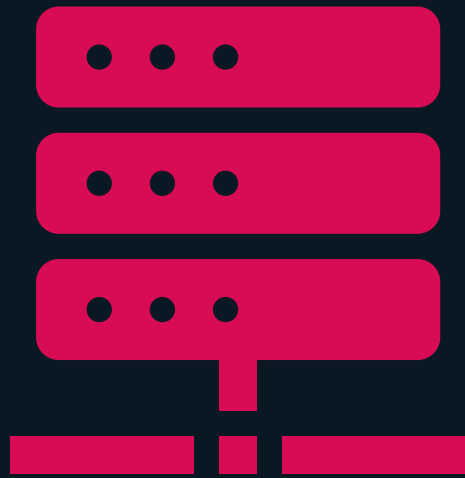


Reliance on AitM

NSPX30 dropper component is delivered via hijacked updates.

Communicates with its infrastructure through AitM.

NSPX30 main features



Reliance on AitM

NSPX30 dropper component is delivered via hijacked updates.

Communicates with its infrastructure through AitM.

msnsp.dll

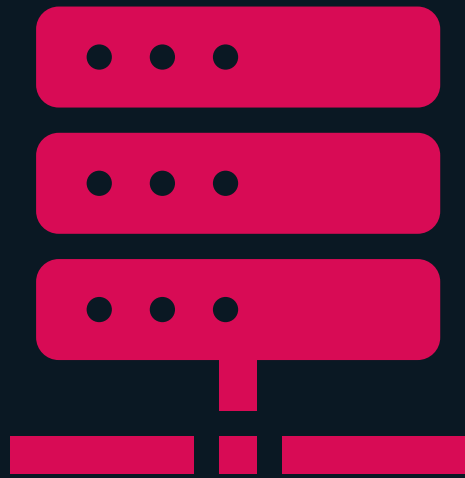


Persistence

Loader persisted as a Winsock Namespace Package (NSP) DLL.

Malicious DLL is automatically loaded into processes that use Winsock.

NSPX30 main features



Reliance on AitM

NSPX30 dropper component is delivered via hijacked updates.

Communicates with its infrastructure through AitM.

msnsp.dll



Persistence

Loader persisted as a Winsock Namespace Package (NSP) DLL.

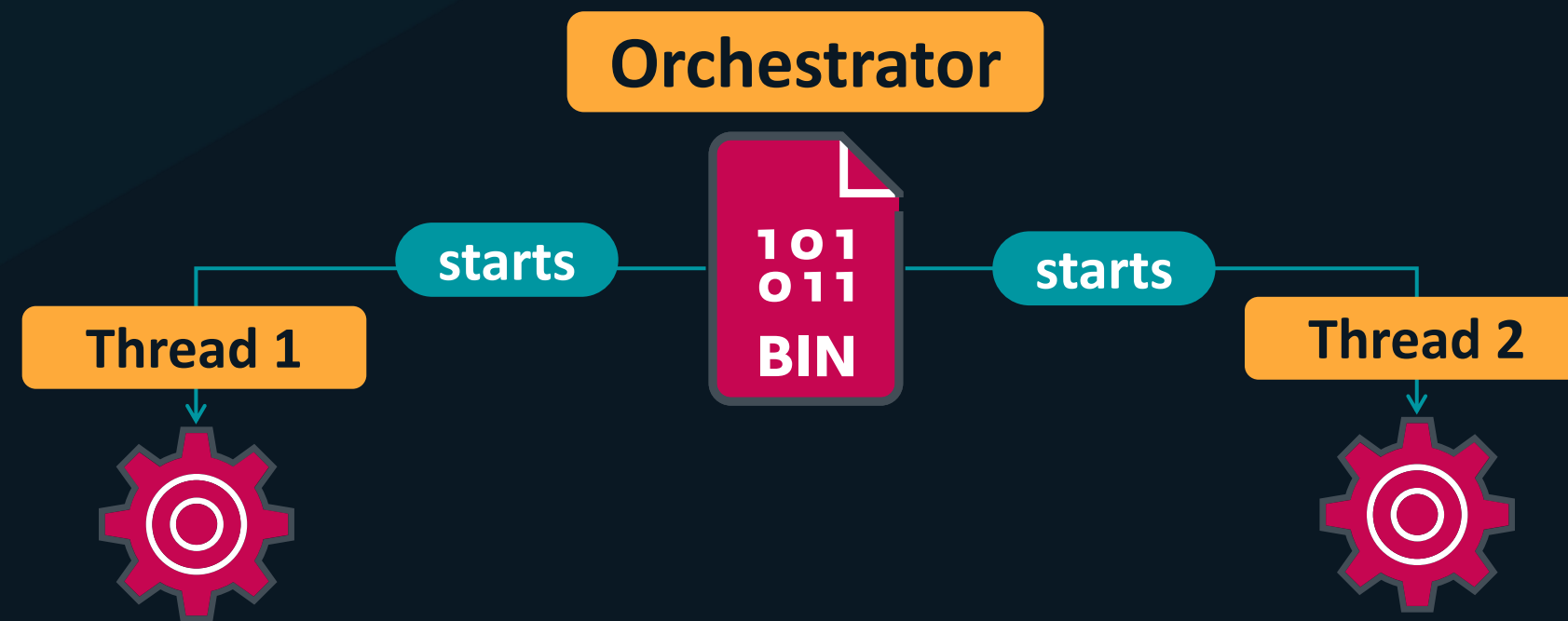
Malicious DLL is automatically loaded into processes that use Winsock.



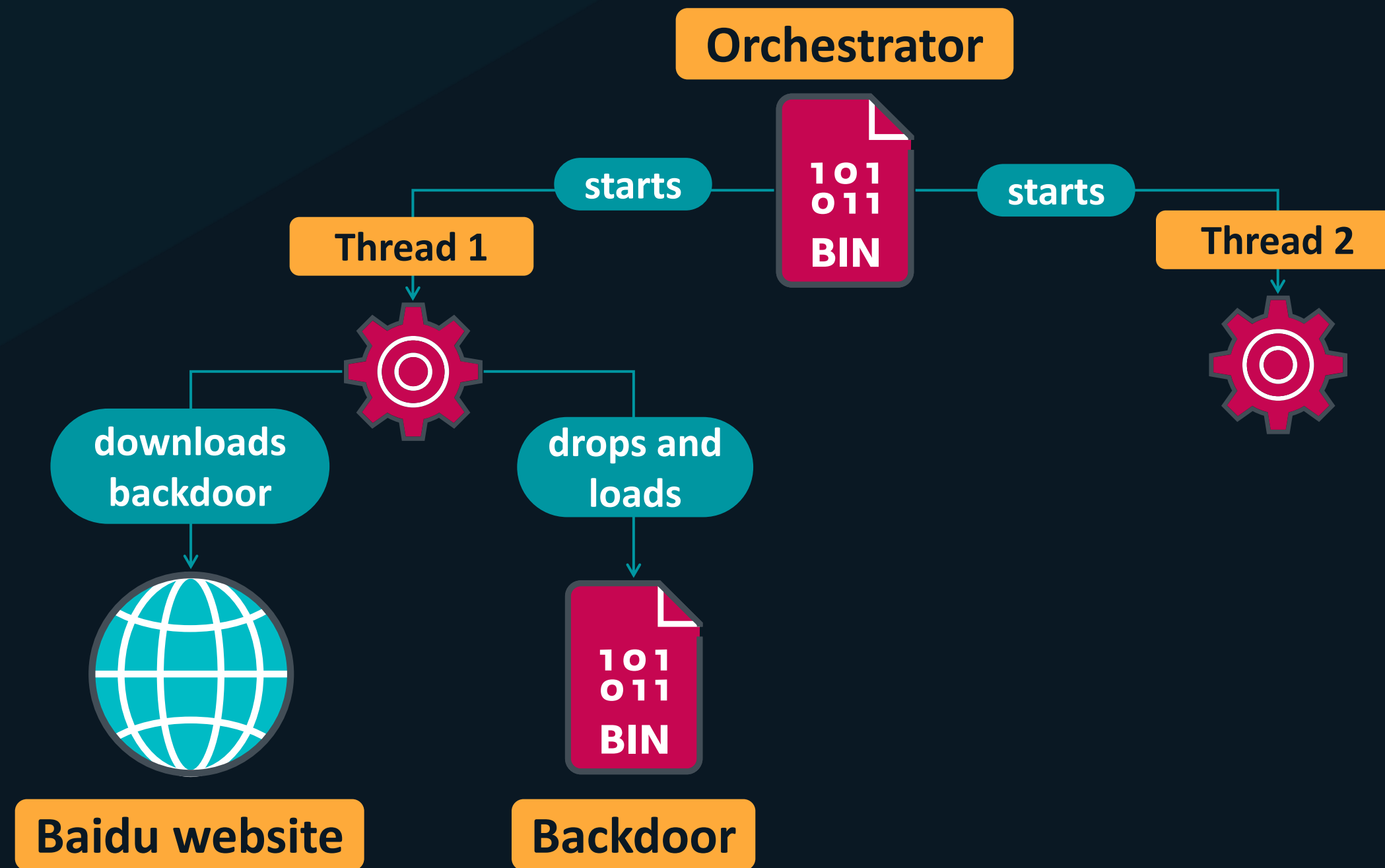
Highly modular

Composed of many components: loaders, orchestrator, backdoor, and three groups of plugins.

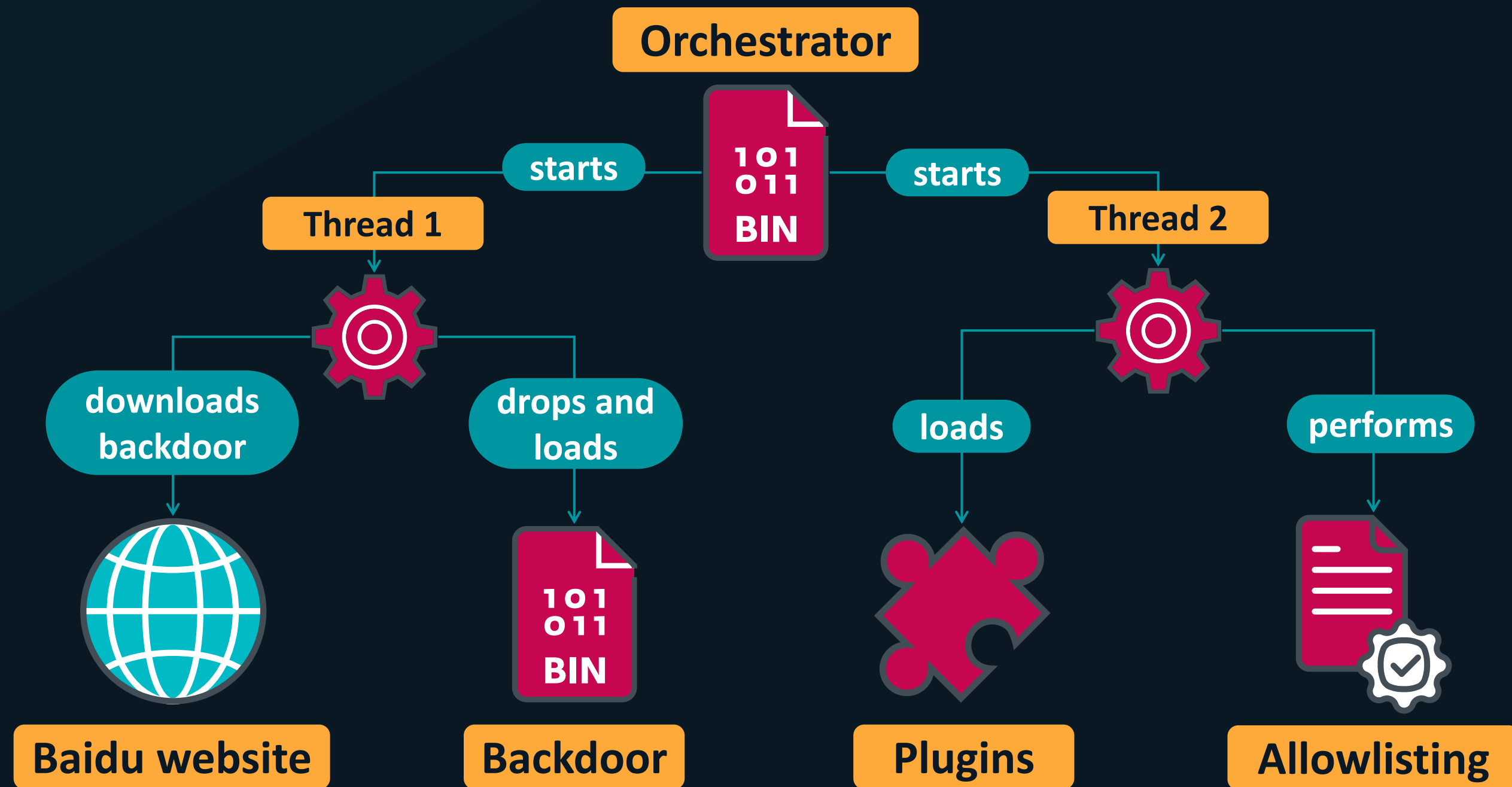
Orchestrator main functionality



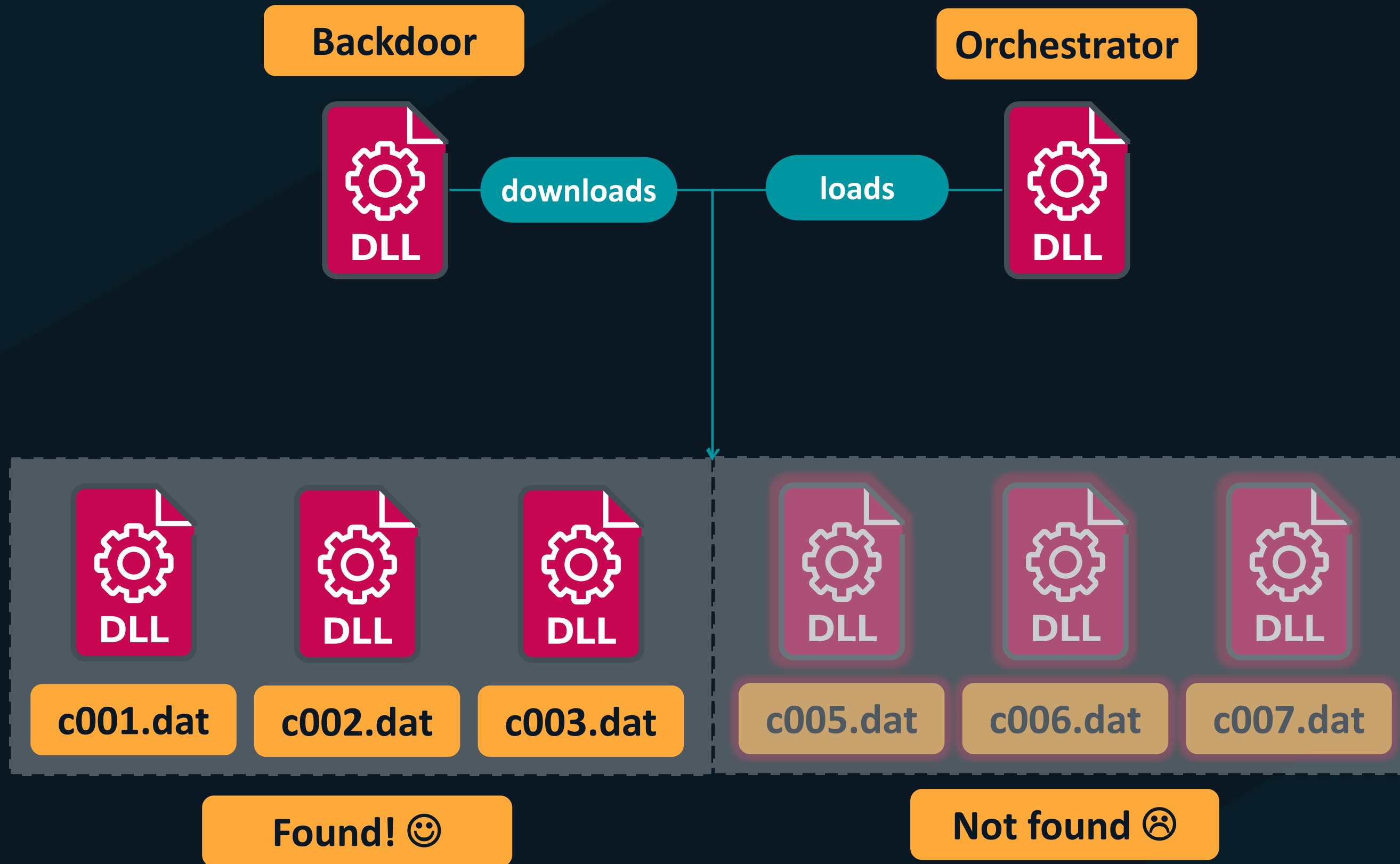
Orchestrator main functionality



Orchestrator main functionality



Orchestrator plugins



The orchestrator does not load a plugin named c004.dat!

Orchestrator plugins

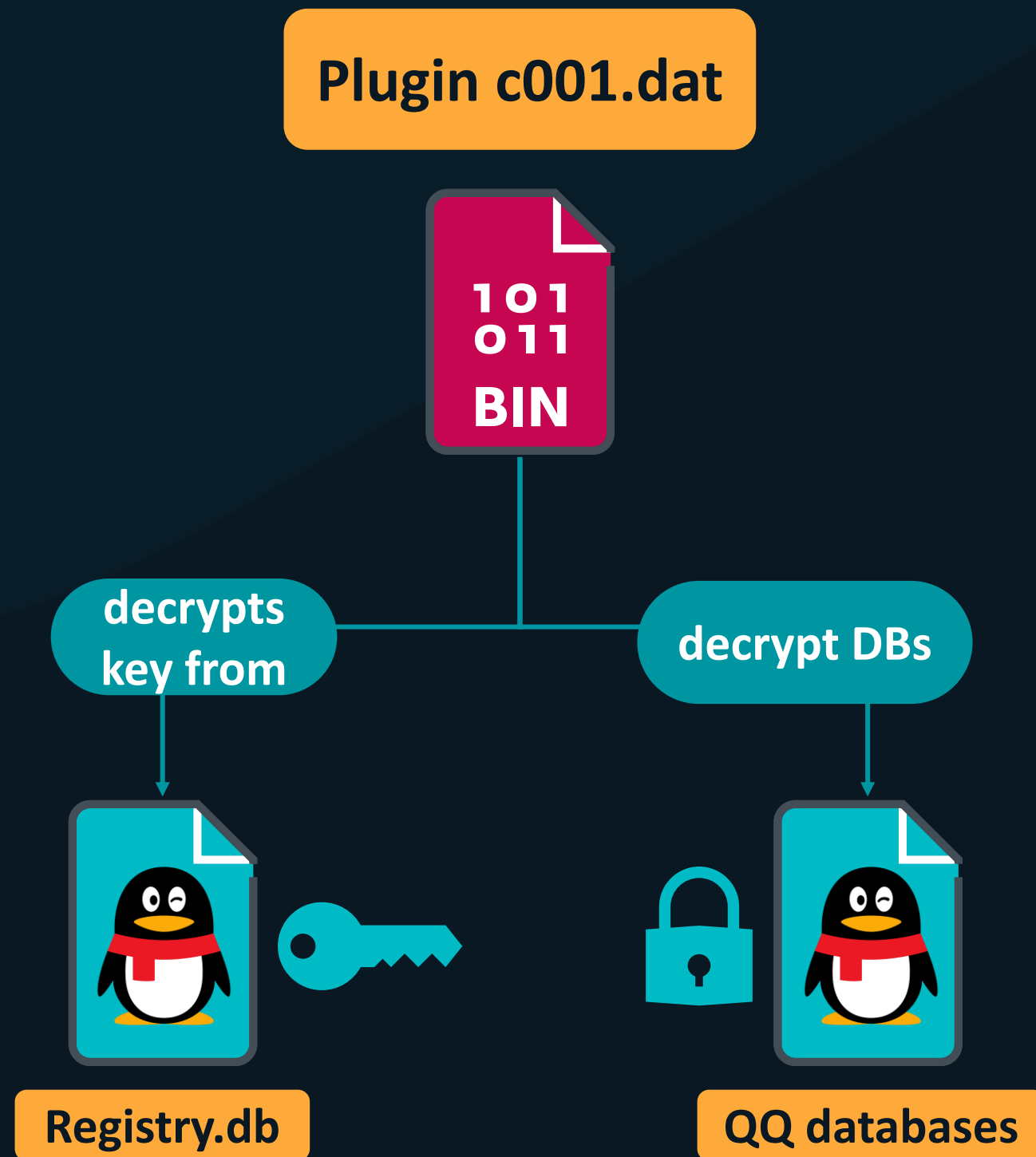
Plugin c001.dat



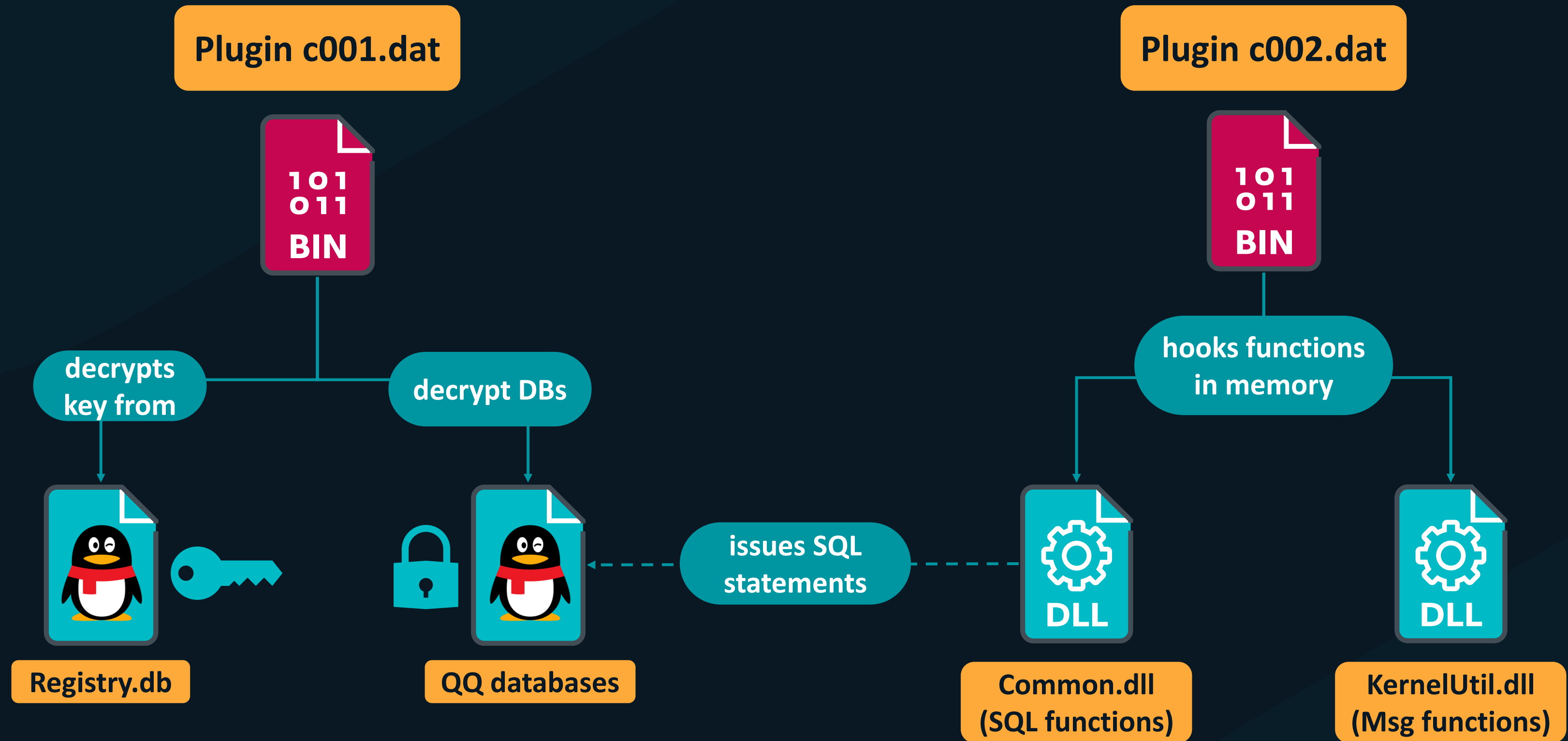
Plugin c002.dat



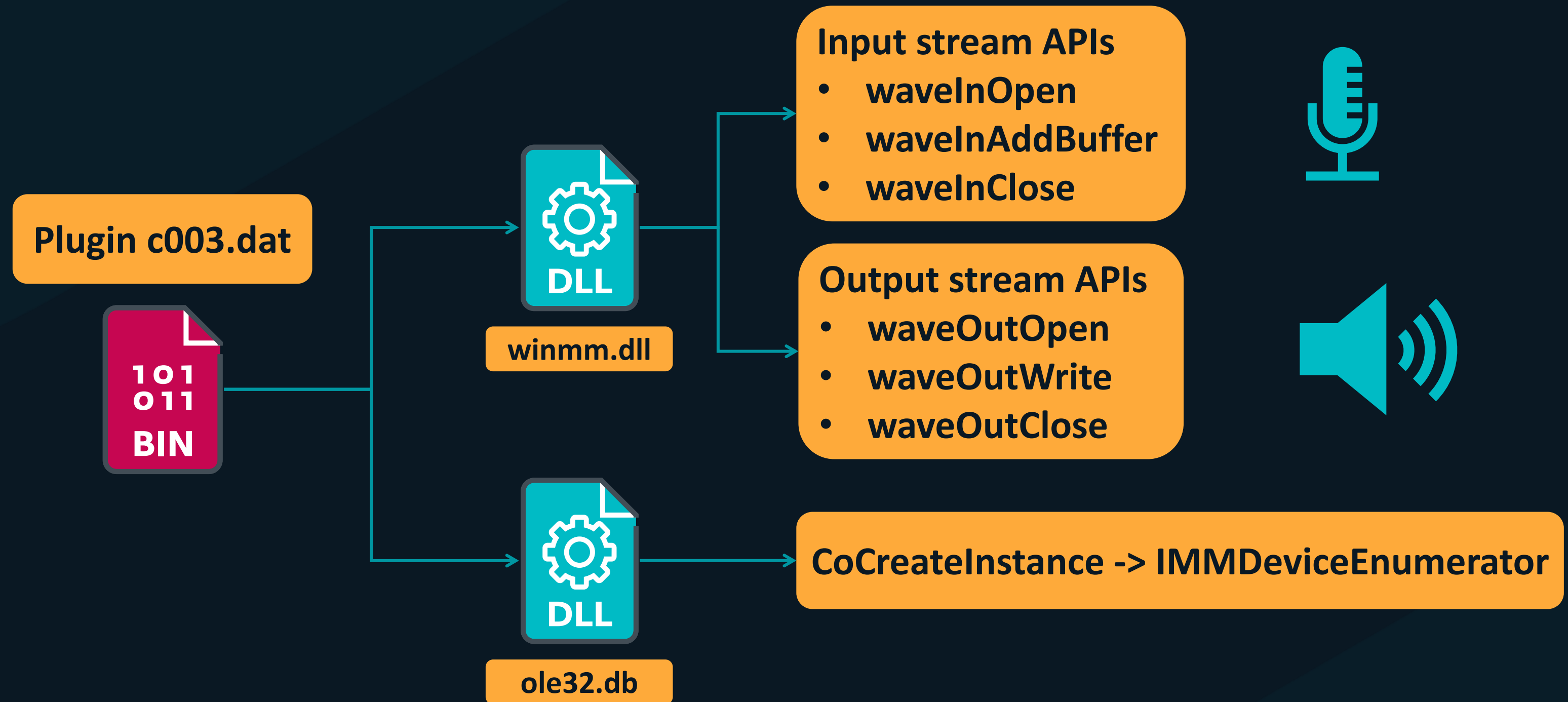
Orchestrator plugins



Orchestrator plugins



Orchestrator plugins



"AITM is more than our favourite technique, it's a way of life!" NSPX30's developers, maybe.

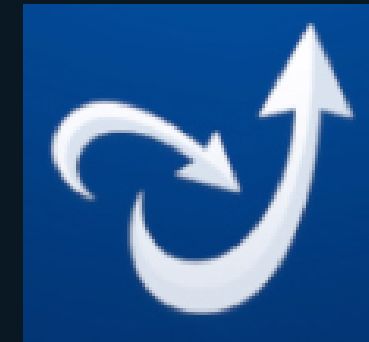
Allowlisting in security software



**Tencent PC
Manager**



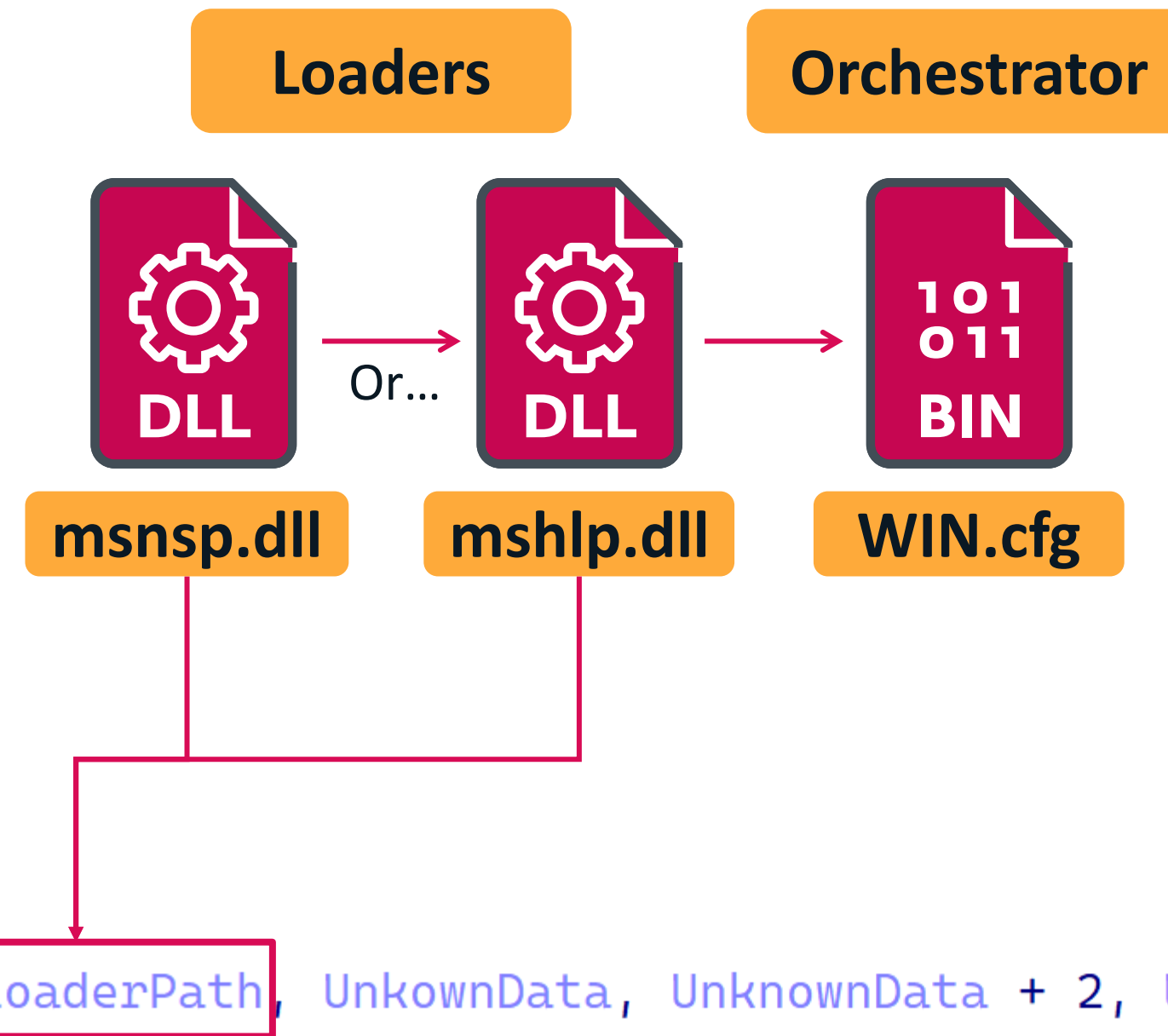
**360 Safeguard
And Antivirus**



Kingsoft Antivirus

Allowlisting in Tencent PC Manager

```
if ( (CreateTavInstance)(2, &TavInstanceVtbl) >= 0 && ((*TavInstanceVtbl + 8))(TavInstanceVtbl) )
{
    HIWORD(UnknownData[0]) = 0;
    memset(&UnkownData[2], 0, 0x1FCu);
    v11 = 0;
    wmemcpy(Source, L"TSPDLdr.dll", 11);
    memset(&Source[11], 0, 0x1F2);
    UnknownData[2] = 0x52A06DFB;
    UnknownData[3] = 0x65877684;
    UnknownData[1] = 0x52A8624B;
    *&UnkownData[4] = 0x52A06DFB;
    *&UnkownData[8] = 0x65877684;
    UnknownData[4] = 0xFF1A4EF6;
    *UnkownData = 0x52A8624B;
    *&UnkownData[12] = 0xFF1A4EF6;
    wcsncpy(&UnkownData[16], Source, 0xEFu);
    ((*TavInstanceVtbl + 12))(TavInstanceVtbl, pszLoaderPath, UnkownData, UnkownData + 2, UnkownData + 2);
}
```

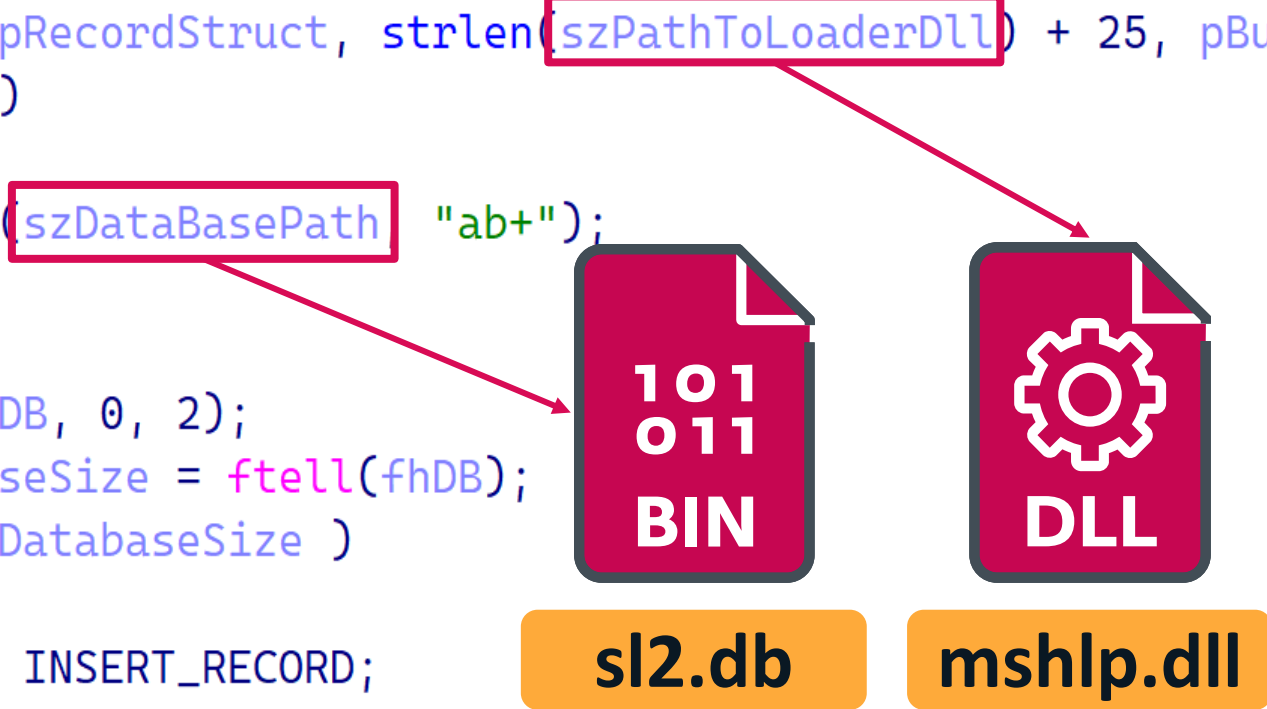


Used by NSPX30

Allowlisting in 360 software

Target: 360 Antivirus

```
NSPX_EncodeData(pRecordStruct, strlen(szPathToLoaderDll) + 25, pBuffer);
if ( pBuffer[0] )
{
    fhDB = fopen(szDataBasePath "ab+");
    if ( fhDB )
    {
        fseek(fhDB, 0, 2);
        dwDatabaseSize = ftell(fhDB);
        if ( !dwDatabaseSize )
        {
            goto INSERT_RECORD;
        }
        pDatabaseBuffer = malloc(dwDatabaseSize + 1);
        pDatabaseBuffer[dwDatabaseSize] = 0;
        fseek(fhDB, 0, 0);
        fread(pDatabaseBuffer, 1u, dwDatabaseSize, fhDB);
        v7 = strstr(pDatabaseBuffer, pBuffer[0]) != 0;
        free(pDatabaseBuffer);
        if ( !v7 )
        {
INSERT_RECORD:
            fseek(fhDB, 0, 2);
            fprintf(fhDB, "%s\r\n", pBuffer[0]);
        }
        fclose(fhDB);
    }
}
```



Used by: NSPX30 and Gelsemium.

Target: 360 Safeguard

```
strcpy(ProcName, "XDOpen");
XDOpen = GetProcAddress(result, ProcName);
strcpy(v14, "XDAddRecordsEx");
XDAddRecordsEx = GetProcAddress(v5, v14);
strcpy(v23, "XDClose");
XDClose = GetProcAddress(v5, v23);
strcpy(&sznewspy_killer[9], "l");
strcpy(&sznewspy_killer[10], "l");
strcpy(sz360safe, "3");
strcpy(&sz360safe[1], "6");
strcpy(&sz360safe[2], "0");
strcpy(&sz360safe[3], "S");
strcpy(&sz360safe[4], "a");
strcpy(&sz360safe[5], "f");
strcpy(v25, "e");
v25[1] = 0;
strcpy(sznewspy_killer, "n");
strcpy(&sznewspy_killer[1], "e");
strcpy(&sznewspy_killer[2], "w");
strcpy(&sznewspy_killer[3], "s");
strcpy(&sznewspy_killer[4], "p");
strcpy(&sznewspy_killer[5], "y");
strcpy(&sznewspy_killer[6], "_");
strcpy(&sznewspy_killer[7], "k");
strcpy(&sznewspy_killer[8], "i");
strcpy(&sznewspy_killer[11], "e");
strcpy(v8, "r");
v8[1] = 0;
Handle = (XDOpen)(sz360safe, sznewspy_killer, szSpeedmem2FilePath);
(XDAddRecordsEx)(Handle, StructureWithDllPath, v31);
(XDClose)(Handle);
```

Used by: NSPX30, Gelsemium, and McRAT.

Allowlisting in Kingsoft Antivirus

```
wcscpy(LibFileName, L"security\\kxescan\\khhistory.dll");
LibraryW = LoadLibraryW(LibFileName);
if ( LibraryW )
{
    strcpy(ProcName, "KSDllGetClassObject");
    KSDllGetClassObject = GetProcAddress(LibraryW, ProcName);
    if ( KSDllGetClassObject )
    {
        v7 = 0;
        (KSDllGetClassObject)(&unk_10009050, &unk_10009060, &v7);
        if ( v7 )
        {
            v4 = *(*v7 + 12);
            if ( v4 )
            {
                v4(v7, 0);
                v5 = *(*v7 + 16);
                if ( v5 )
                {
                    v8[0] = 1;
                    v8[2] = pszLoaderPath;
                    v8[1] = 12;
                    v5(v7, v8);
                    return 1;
                }
            }
        }
    }
}
```



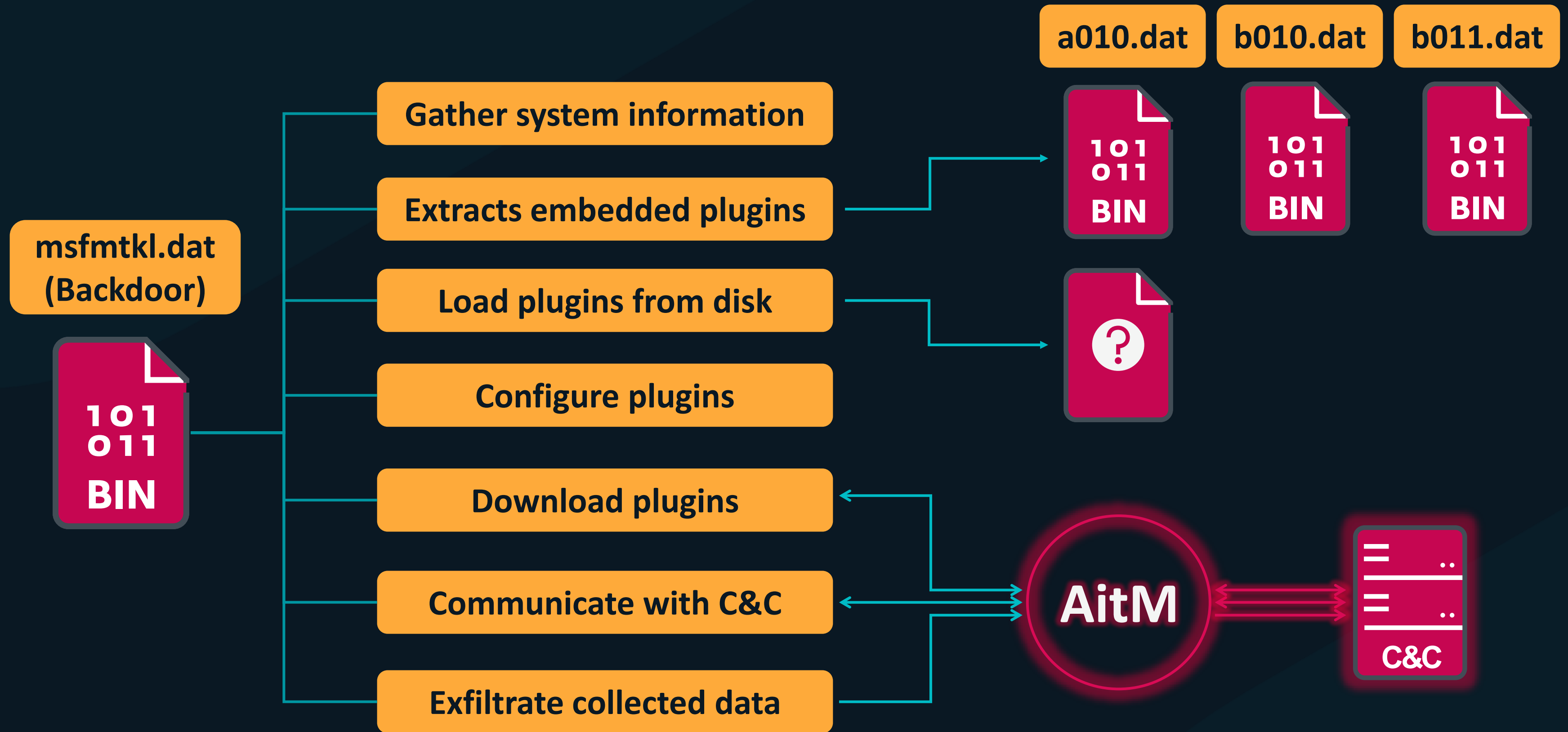
msnsp.dll

Used by: NSPX30

Bonus mention: allowlisting in Windows Defender

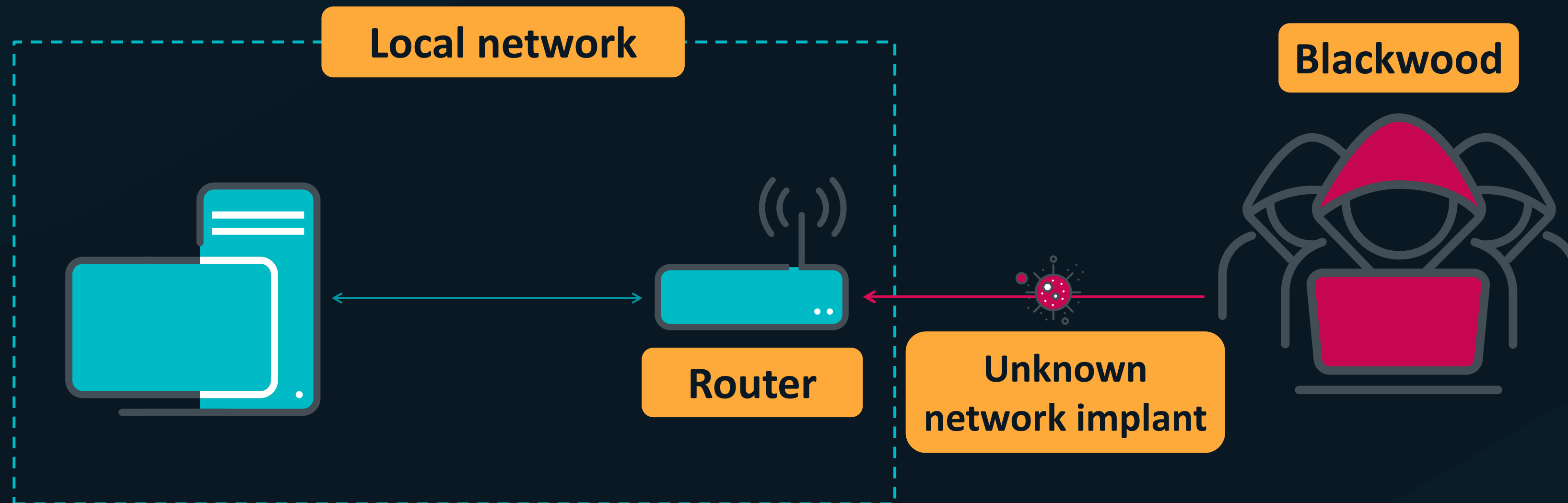
```
strcpy(
    szCommandDisableSubmit,
    "cmd /c powershell -inputformat none -outputformat none -NonInteractive -Command \"Set-MpPreference -SubmitSamplesConsent 0\"");
strcpy(
    szCommandExclusion,
    "cmd /c powershell -inputformat none -outputformat none -NonInteractive -Command \"Add-MpPreference -ExclusionPath \"%s\\\"\"");
memset(szmshlpPath, 0, sizeof(szmshlpPath));
v35 = 0;
v36 = 0;
Installer_CreateWindowsDirectory(szmshlpPath);
strcpy(v19, "mshlp.dll");
strcat(szmshlpPath, v19);
Installer_CreateProcessAndWait(szCommandDisableSubmit);
Installer_CreateProcessAndWait(szCommandExclusion, szmshlpPath);
```


Overview of the backdoor's functionality

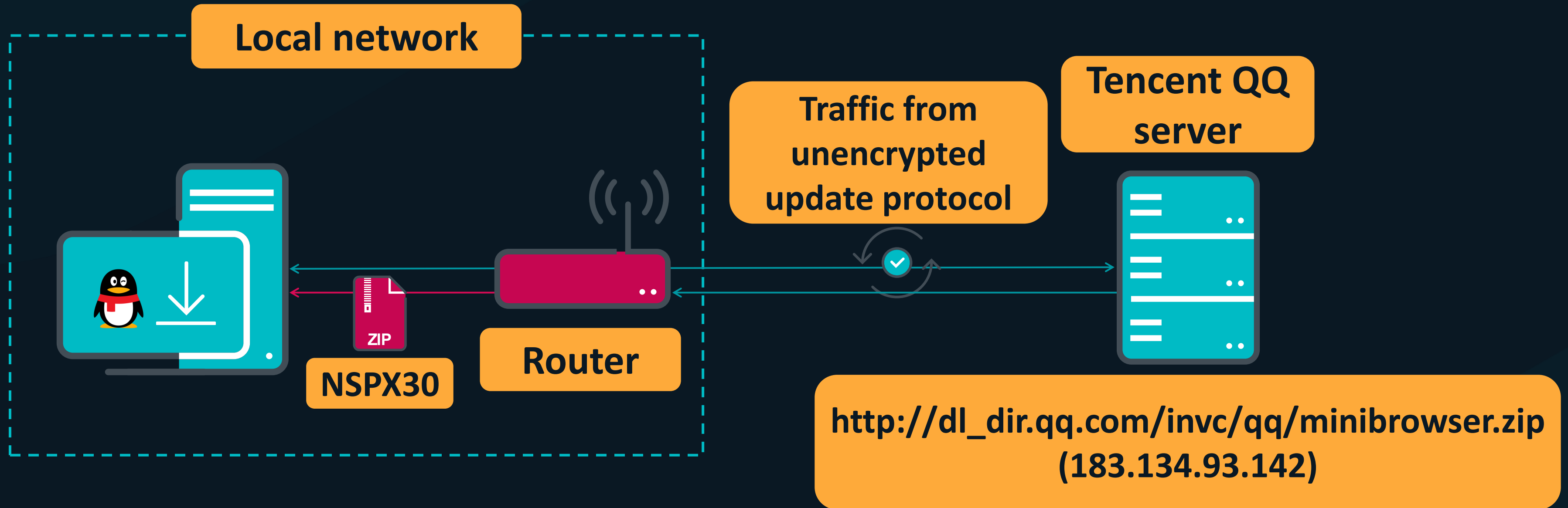


Anonymizing the attacker's infrastructure via AitM

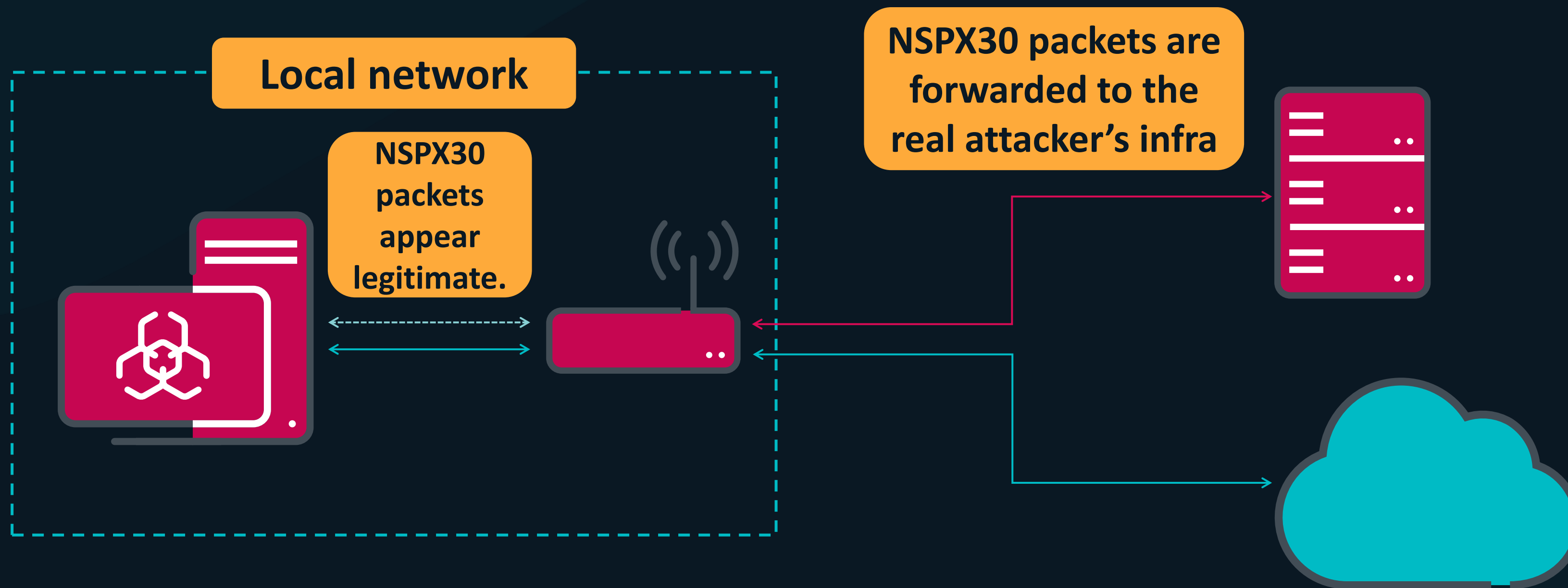
How we believe Blackwood operates



How we believe Blackwood operates



How we believe Blackwood operates



Downloading components

```
Hypertext Transfer Protocol
GET /id=87f4de4c6d827f49b60e867824c02827&ad=32&os=6.2&t=0 HTTP/1.1\r\n
> [Expert Info (Chat/Sequence): GET /id=87f4de4c6d827f49b60e867824c02827&ad=32&os=6.2&t=0 HTTP/1.1\r\n]
Request Method: GET
Request URI: /id=87f4de4c6d827f49b60e867824c02827&ad=32&os=6.2&t=0
Request Version: HTTP/1.1
User-Agent: Mozilla/4.0 (compatible;MSIE 5.0; Windows 98)\r\n
Host: www.baidu.com\r\n
```

Fingerprint



<http://www.baidu.com>

[/id=<ID>&ad=32&os=<OS_MAJOR_VERSION>.<OS_MINOR_VERSION>&t=<ATTEMPTS>](#)

Mozilla/4.0 (compatible;MSIE 5.0; Windows 98)

UDP interception

```
> Internet Protocol Version 4, Src: 10.1.40.149, Dst: 180.76.76.11
> User Datagram Protocol, Src Port: 63137, Dst Port: 53
▼ Domain Name System (query)
  Transaction ID: 0xfead
  > Flags: 0x0100 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  ▼ Queries
    ▼ microsoft.com: type A, class IN
      Name: microsoft.com
      [Name Length: 13]
      [Label Count: 2]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
  ▼ Extraneous Data (1166 bytes)
    Extraneous Data Bytes: aa9d9d9d9d9d9da5aa7ba97978a97eab79a5afaa7ba5
    [Extraneous Data Length: 1166]
```

0000	00 10 db ff 50 00 00 50 56 98 76 83 08 00 45 00	P..P V.v...E.
0010	04 c9 39 dc 00 00 80 11 00 00 0a 01 28 95 b4 4c	..9.....(..L
0020	4c 0b f6 a1 00 35 04 b5 37 b4 fe ad 01 00 00 01	L....5.. 7.....
0030	00 00 00 00 00 00 09 6d 69 63 72 6f 73 6f 66 74	m icrosoft
0040	03 63 6f 6d 00 00 01 00 01 aa 9d 9d 9d 9d 9d 9d	.com.....
0050	a5 aa 7b a9 79 78 a9 7e ab 79 a5 af aa 7b a9 a4	..{.yx.~ .y...{..
0060	7f ab ad 78 a5 ab aa a5 af a9 7e ad af a5 af aa	...x.... .~.....
0070	f8 26 25 02 00 e8 03 00 00 01 5a 04 44 45 53 4b	.&%..... .Z.DESK
0080	54 4f 50 2d 48 53 38 54 34 51 53 00 00 00 00 00	TOP-HS8T 4QS.....

Fingerprint



DNS server at 180.76.76.11/24

Port: 53, 4499, 8000

Transaction ID always 0xFEAD

Domain: microsoft.com

Appended data to exfiltrate

UDP interception

Beginning at 180.76.76.11

Port: 53 or 4499 or 8000

Destination	Protocol	Length	Info
180.76.76.11	DNS	1239	Standard query 0xfead A microsoft.com
180.76.76.12	DNS	125	Standard query 0xfead A microsoft.com
180.76.76.13	DNS	1239	Standard query 0xfead A microsoft.com
180.76.76.14	DNS	1239	Standard query 0xfead A microsoft.com
180.76.76.15	DNS	1239	Standard query 0xfead A microsoft.com
180.76.76.16	DNS	1239	Standard query 0xfead A microsoft.com

Baidu DNS is at 180.76.76.76



Public DNS

- [How to use public DNS on PC](#)

Introduce how to set up 180.76.76.76 on Windows, MAC, Linux and other platforms

- [How to use public DNS on mobile devices](#)

Introduce how to set up 180.76.76.76 on Adnroid and IOS mobile phones

- [How to use public DNS on the router side](#)

Introduce how to set 180.76.76.76 on the router

- [What are the characteristics of Baidu's public DNS service?](#)

Safe, no hijacking, more accurate

- [What is the IP of Baidu's public DNS?](#)

ipv4 180.76.76.76 ipv6 2400:da00::6666

What about ISP infrastructure compromise?

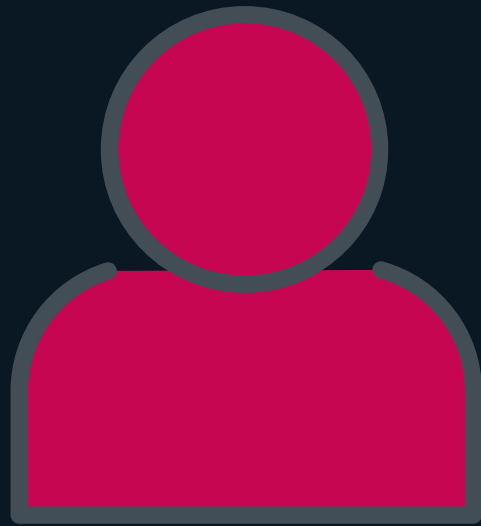
2022/1/27 Japan Security Analyst Conference 2022

LuoYu: Continuous Espionage Activities Targeting Japan with the new version of WinDealer in 2021

Leon Chang, Yusuke Niwa, Suguru Ishimaru

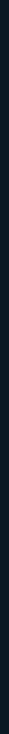
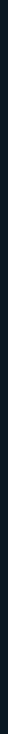


What about ISP compromise?



Global reach

Not all targets are located in China.

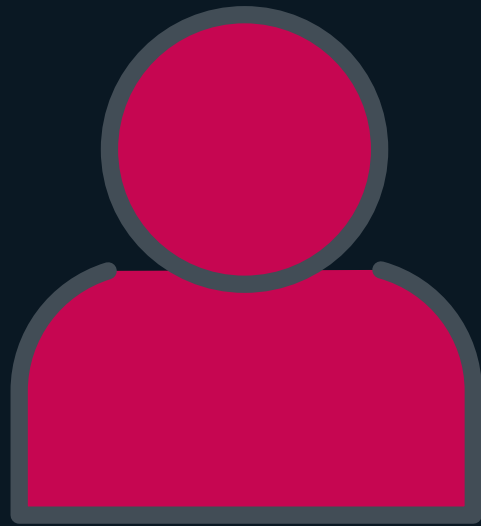


China

Conclusion

Summary

What about ISP compromise?



Global reach

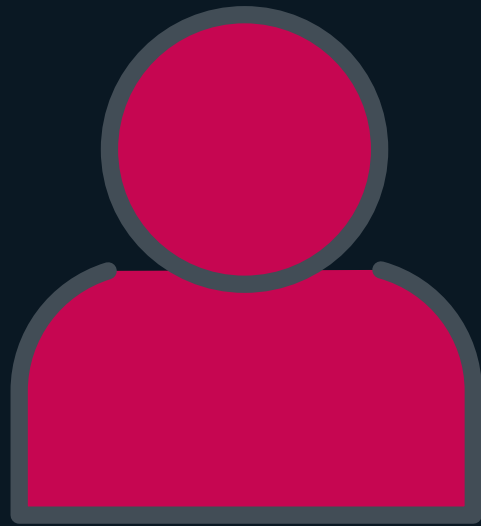
Not all targets are located in China.



Not always China

Some servers from Baidu network 186.76.76.0/24 are *anycast*: they could be geolocated around the world.

What about ISP compromise?



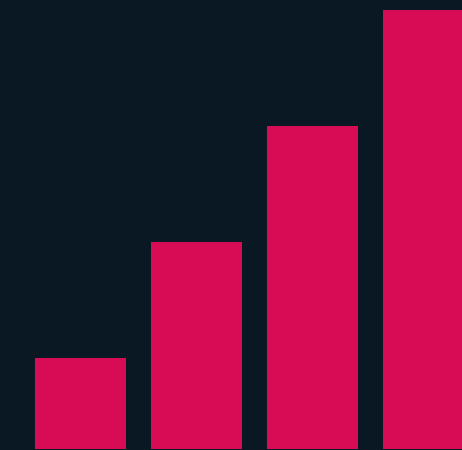
Global reach

Not all targets are located in China.



Not always China!

Some servers from Baidu network 186.76.76.0/24 are *anycast*: they could be geolocated around the world.



Reliability

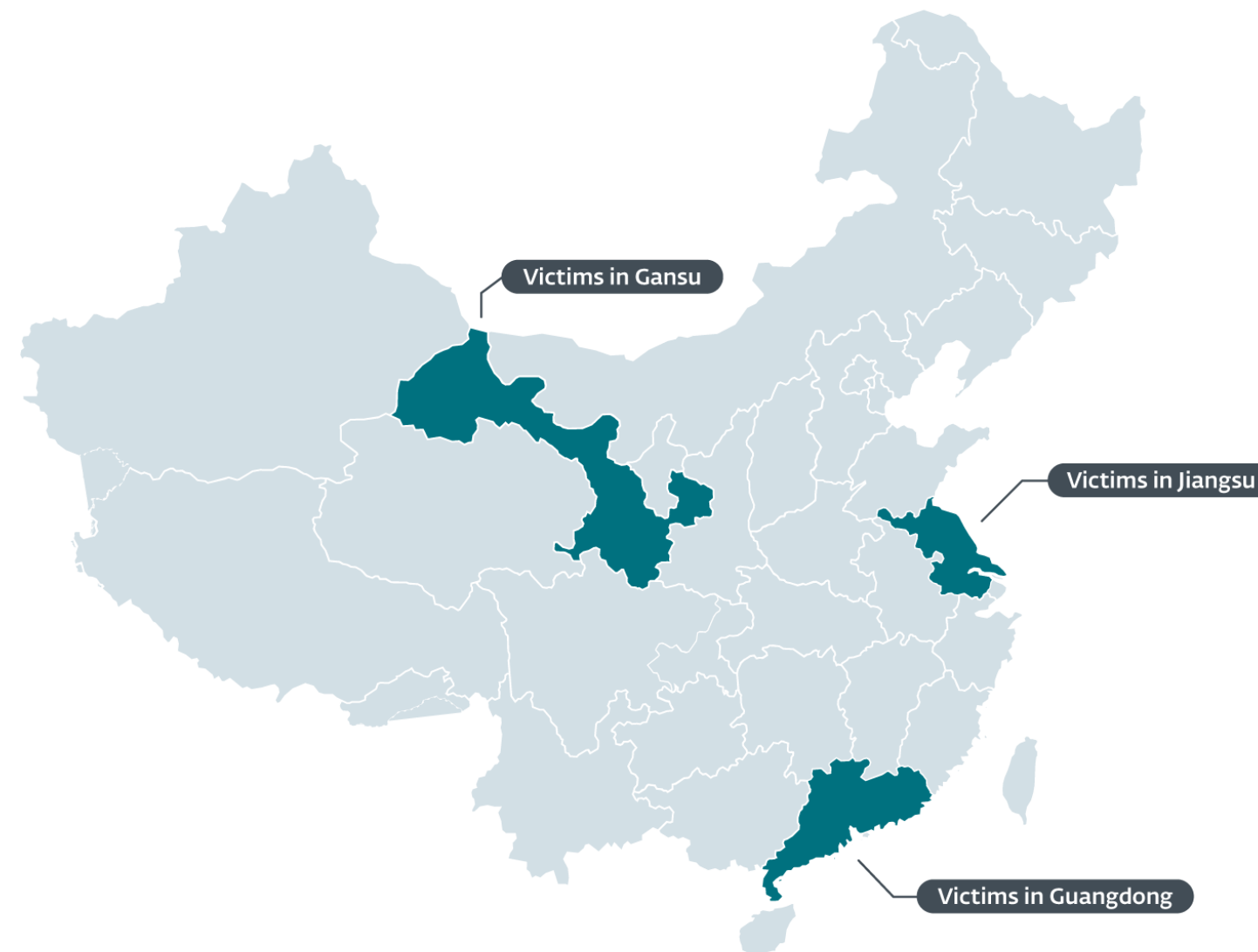
AitM mechanism appears to be reliable. Exfiltration requires A LOT of packets.

Assessment: ISP compromise for AitM is unlikely.

The China-aligned AitM club

Evasive Panda APT group delivers malware via updates for popular Chinese software

ESET Research uncovers a campaign by the APT group known as Evasive Panda targeting an international NGO in China with malware delivered through updates of popular Chinese software



China-aligned APTs with AitM capability tracked by ESET



Evasive Panda



LittleBear



LuoYu



Blackwood

AitM via compromised network device, or ISP? We don't know.

AitM working outside of China networks? Yes.

The update hijacking mechanism seems suspiciously similar for all four clusters

TheWizards APT



Targeted regions

Philippines



Hong Kong



UAE



China



Sectors

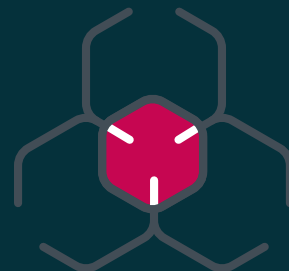


Gambling

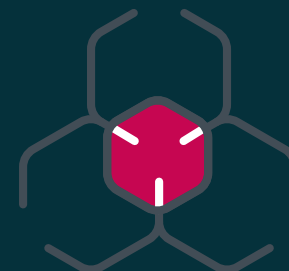


Individuals

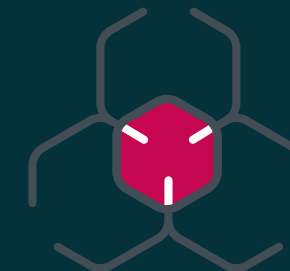
Toolkit



WizardNet



DarkNights



Spellbinder

TheWizards APT

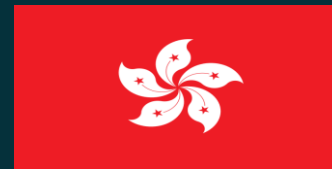


Targeted regions

Philippines



Hong Kong



UAE



China



Sectors

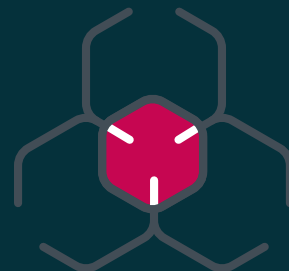


Gambling

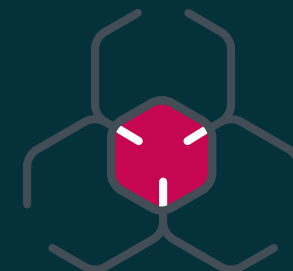


Individuals

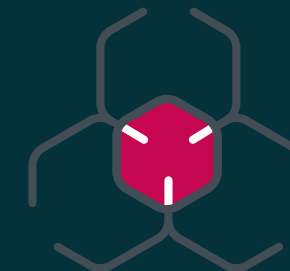
Toolkit



WizardNet



DarkNights



Spellbinder

Spellbinder: lateral movement tool that performs AitM via IPv6 *Stateless Address Autoconfiguration* (SLAAC) spoofing attack

Neighbor Discovery Protocol

[Article](#) [Talk](#)

From Wikipedia, the free encyclopedia

The **Neighbor Discovery Protocol** (**NDP**), or simply **Neighbor Discovery** (**ND**), is a protocol of the [Internet protocol suite](#) used with [Internet Protocol Version 6](#) (IPv6).^[1] It operates at the [network layer](#) of the Internet model,^{[2][3]} and is responsible for gathering various information required for network communication, including the configuration of local connections and the [domain name servers](#) and gateways.^[4]

Functions [\[edit \]](#)

NDP defines five [ICMPv6](#) packet types for the purpose of router solicitation, router advertisement, neighbor solicitation, neighbor advertisement, and network redirects.^[4]

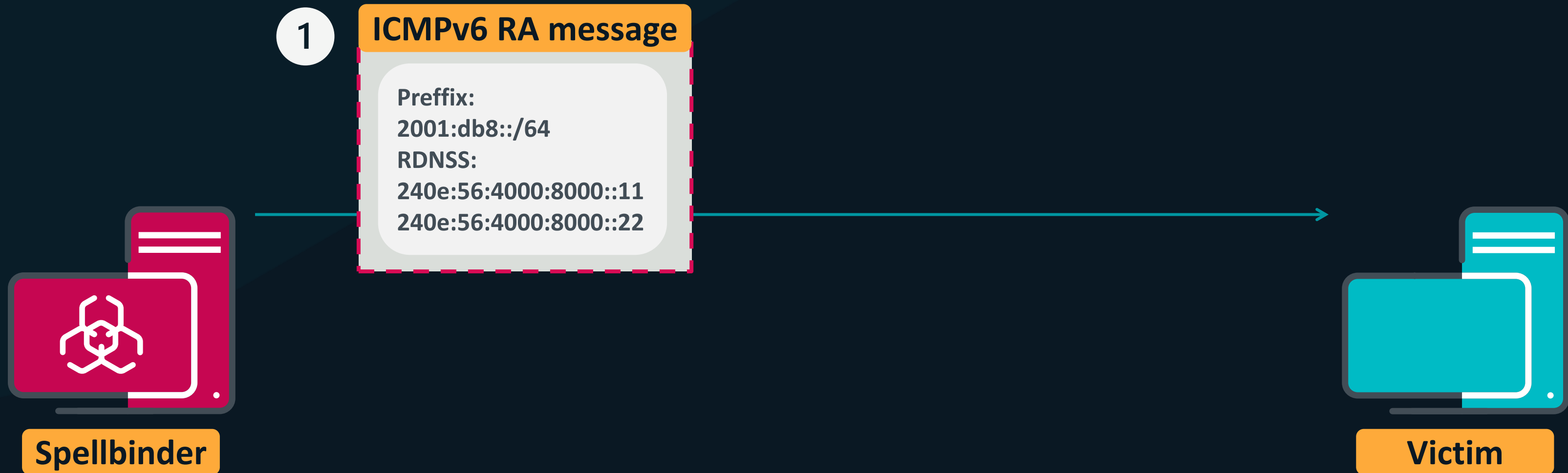
Router Solicitation (Type 133)

Hosts inquire with Router Solicitation messages to locate routers on an attached link.^[5] Routers which forward packets not addressed to them generate Router Advertisements immediately upon receipt of this message rather than at their next scheduled time.

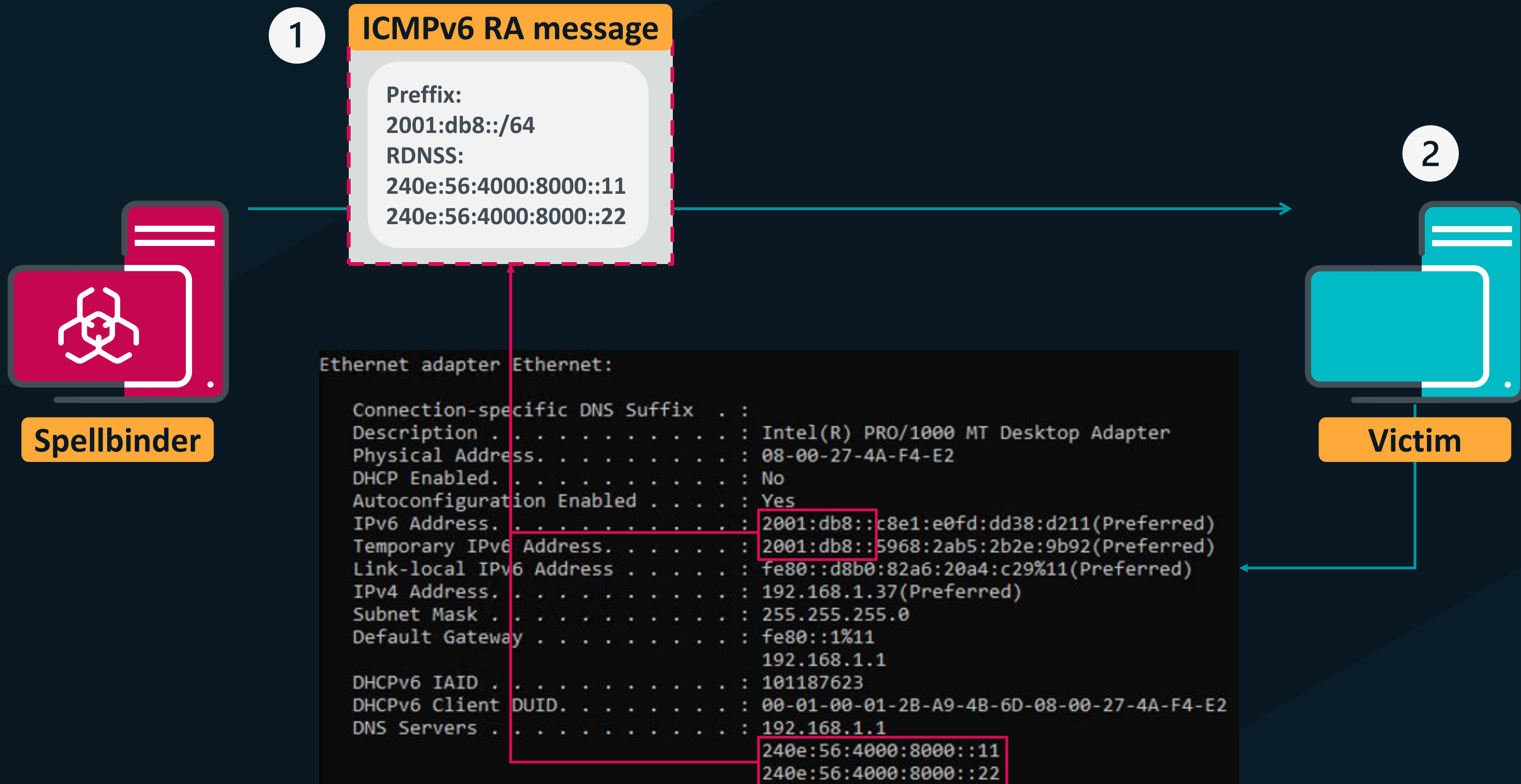
Router Advertisement (Type 134)

Routers advertise their presence together with various link and Internet parameters either periodically, or in response to a Router Solicitation message.

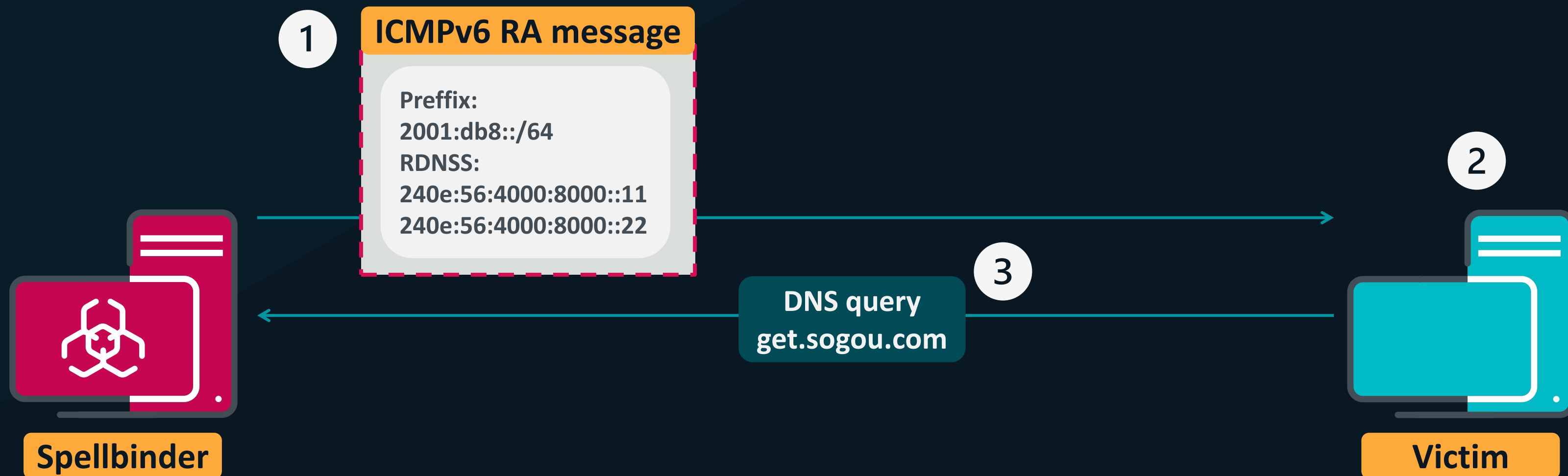
TheWizards approach to AitM



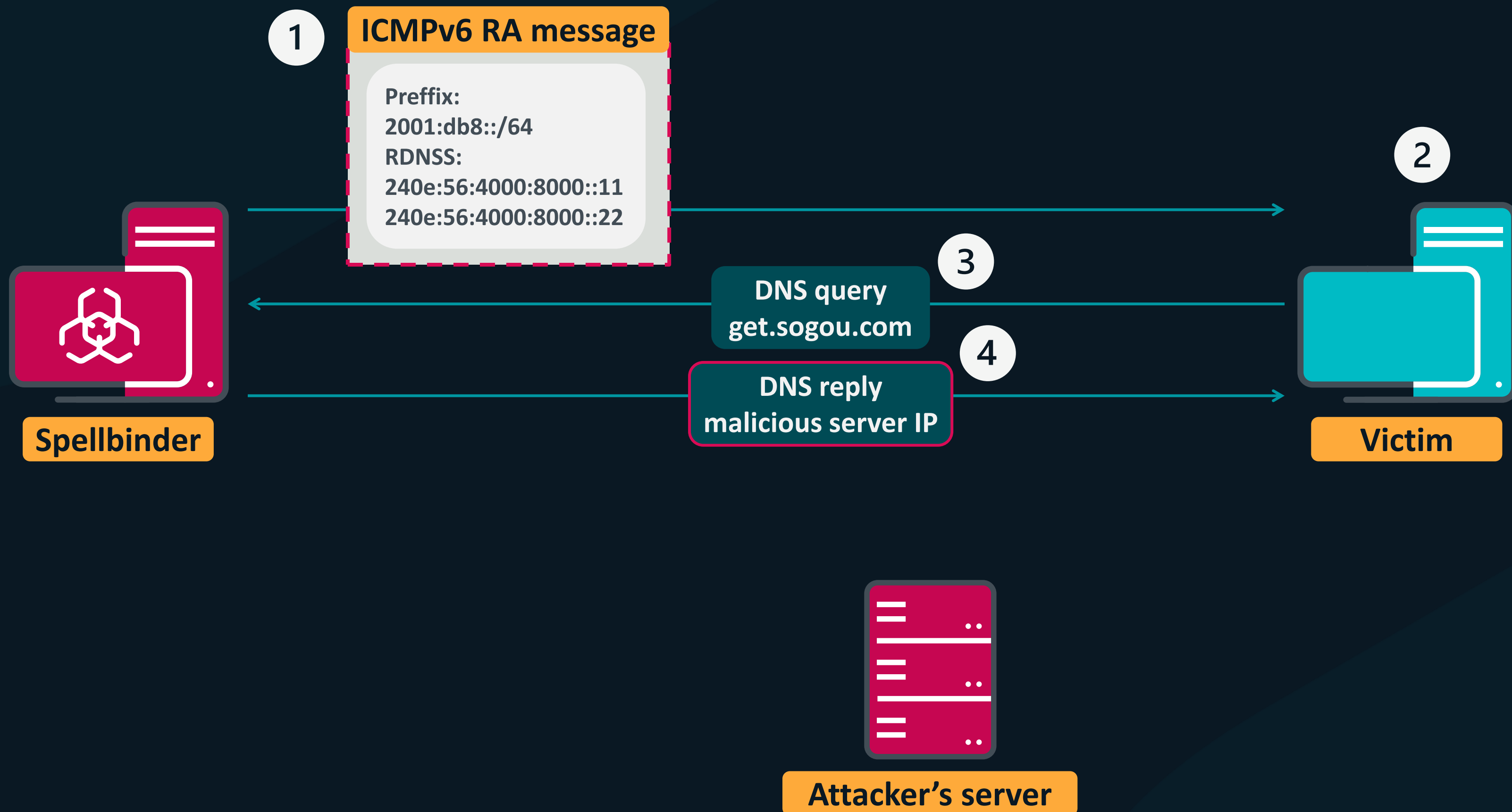
TheWizards approach to AitM



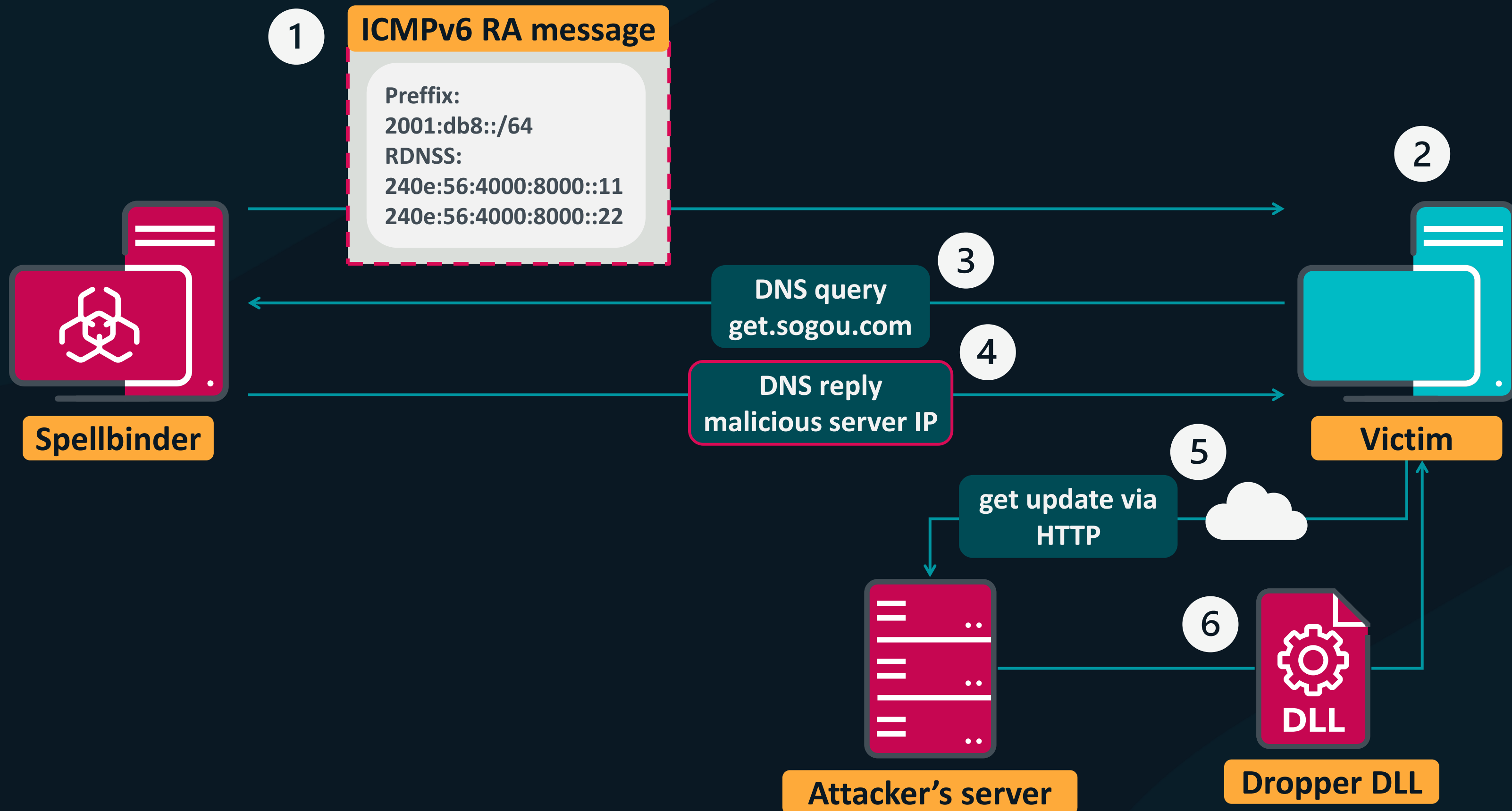
TheWizards approach to AitM



TheWizards approach to AitM



TheWizards approach to AitM



Spellbinder's IPv6 SLAAC attack

- ✓ **Attack vector discussed by the IETF as early as 2008**
- ✓ **IPv6 is enabled by default on modern Windows OS**
- ✓ **Very effective:**
 - Dozens of machines compromised in a short time
 - No noticeable effect for the victims

Conclusion

What is known and what is not known

- NSPX30 and DCM for Win32
- Project Wood for Win32

- Project Wood is alive and well:
- Linux version recently found!



What is known and what is not known

- NSPX30 and DCM for Win32
- Project Wood for Win32

- Project Wood is alive and well:
- Linux version recently found!

Is NSPX30 developed by a digital
quartermaster?



What is known and what is not known

- NSPX30 and DCM for Win32
- Project Wood for Win32

- Project Wood is alive and well:
- Linux version recently found!

Is NSPX30 developed by a digital
quartermaster?

How is Blackwood able to accurately
find its victims in cyberspace?



Reports by CitizenLab and McAfee

WUP! There It Is **Privacy and Security Issues in QQ Browser**

By Jeffrey Knockel, Adam Senft, and Ron Deibert

March 28, 2016

We Chat, They Watch **How International Users Unwittingly Build up WeChat's** **Chinese Censorship Apparatus**

By Jeffrey Knockel, Christopher Parsons, Lotus Ruan, Ruohan Xiong, Jedidiah Crandall, and Ron Deibert

Apps Sending Plain HTTP Put Personal Data at Risk



McAfee Labs

| JAN 14, 2015

| 5 MIN READ

What is known and what is not known

- NSPX30 and DCM for Win32
- Project Wood for Win32

- Project Wood is alive and well:
- Linux version recently found!

Is NSPX30 developed by a digital
quartermaster?

How is Blackwood able to accurately
find its victims in cyberspace?

The elusive AitM network implant



どうも ありがとうございます

Q&A



Digital Security
Progress. Protected.