# Unplugging PlugX

## Sinkholing the “PlugX worm” botnet

Félix Aimé

Charles Meslay

sekoia

# Unplugging PlugX

Sinkholing the "PlugX worm" botnet

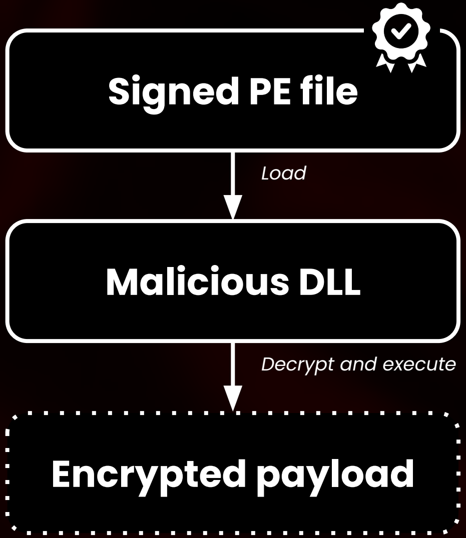
Félix Aimé

Charles Meslay

sekoia

# When our dear old friend PlugX…

- Typical RAT with lot of functionalities.
- Here for more than **15 years**.
- MSS-linked intrusion sets love it!
- Mostly launched via DLL side-loading.
- Many variants & **still in use in 2024**.



# is becoming a worm...

> Custom variant with a **wormable component**.
> Duplicates itself on connected **flash drives**
> Used to bypass **air gap** (replication & files exfiltration.)
> Few variants, four known C2s, linked to **Mustang Panda**.
> Like most of worms, it propagated worldwide.

**MUSTANG PANDA (2012-Today)**

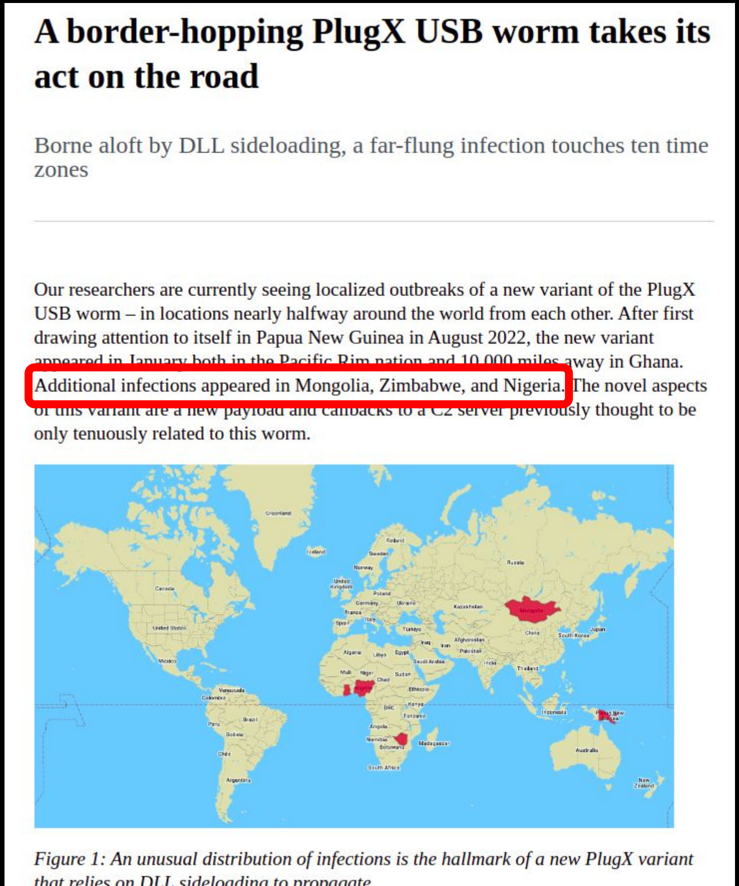
**Known malwares:**
Mostly PlugX variants and custom codes

**Known Infection vectors:**
Phishing, watering holes, USB worms

**Recent targeting**
Strategic topics in Asia, EU, Africa

**Infection vector kink**
Emails leading to malicious archives

# Prior publications...

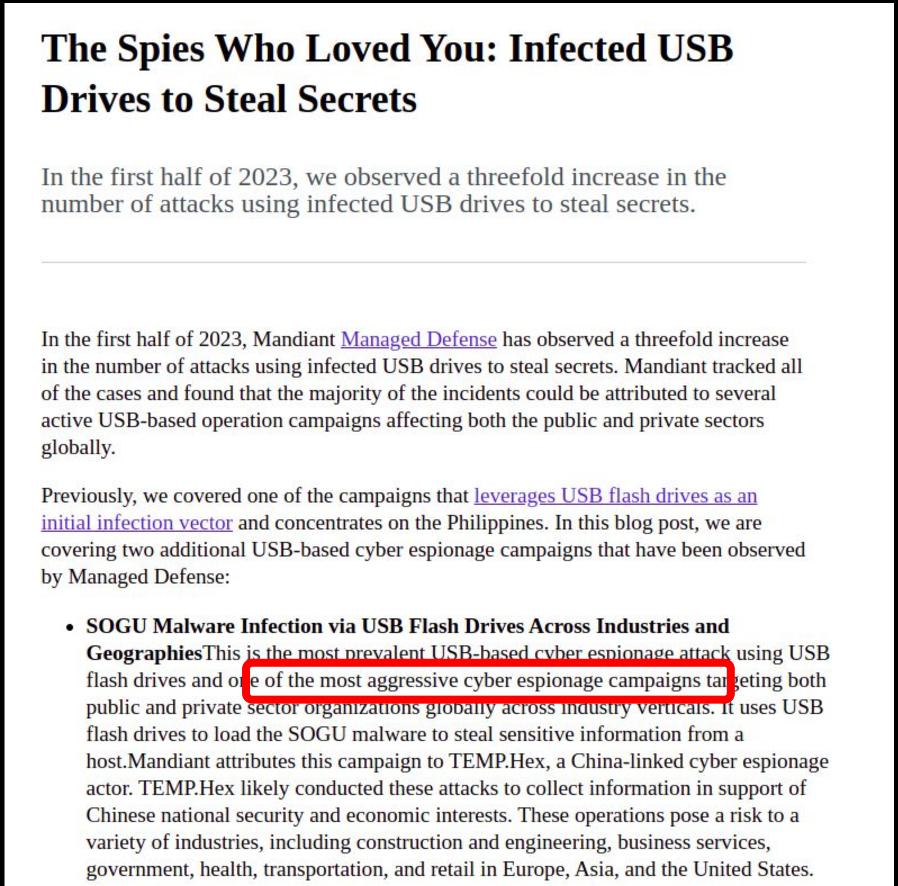


## A border-hopping PlugX USB worm takes its act on the road

Borne aloft by DLL sideloading, a far-flung infection touches ten time zones

Our researchers are currently seeing localized outbreaks of a new variant of the PlugX USB worm – in locations nearly halfway around the world from each other. After first drawing attention to itself in Papua New Guinea in August 2022, the new variant appeared in January both in the Pacific Rim nation and 10,000 miles away in Ghana. Additional infections appeared in Mongolia, Zimbabwe, and Nigeria. The novel aspects of this variant are a new payload and callbacks to a C2 server previously thought to be only tenuously related to this worm.

*Figure 1: An unusual distribution of infections is the hallmark of a new PlugX variant that relies on DLL sideloading to propagate*



## The Spies Who Loved You: Infected USB Drives to Steal Secrets

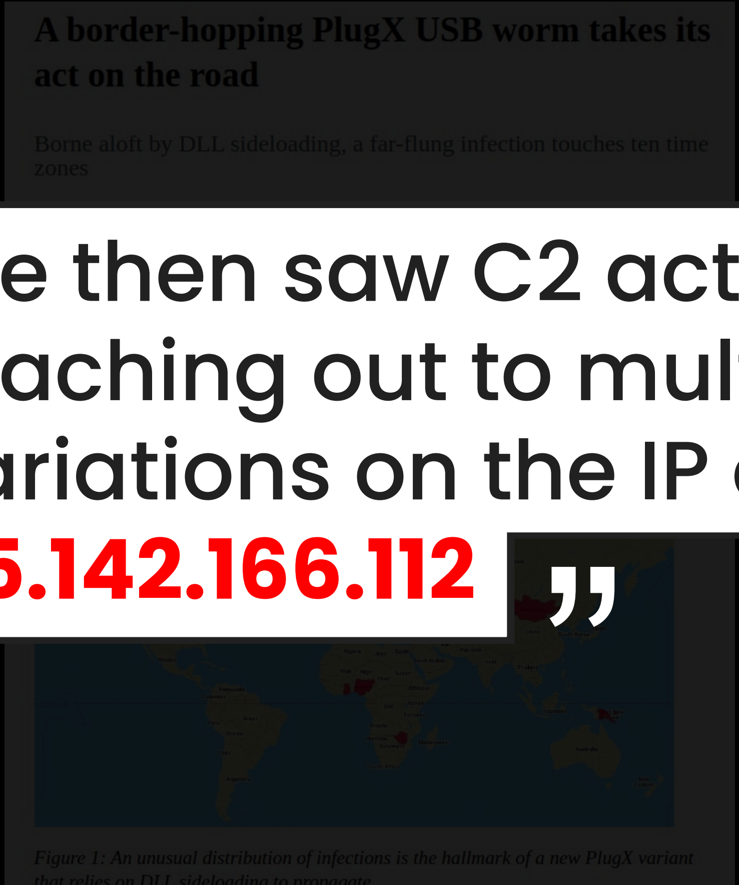
In the first half of 2023, we observed a threefold increase in the number of attacks using infected USB drives to steal secrets.

In the first half of 2023, Mandiant Managed Defense has observed a threefold increase in the number of attacks using infected USB drives to steal secrets. Mandiant tracked all of the cases and found that the majority of the incidents could be attributed to several active USB-based operation campaigns affecting both the public and private sectors globally.

Previously, we covered one of the campaigns that leverages USB flash drives as an initial infection vector and concentrates on the Philippines. In this blog post, we are covering two additional USB-based cyber espionage campaigns that have been observed by Managed Defense:

- **SOGU Malware Infection via USB Flash Drives Across Industries and Geographies**This is the most prevalent USB-based cyber espionage attack using USB flash drives and one of the most aggressive cyber espionage campaigns targeting both public and private sector organizations globally across industry verticals. It uses USB flash drives to load the SOGU malware to steal sensitive information from a host.Mandiant attributes this campaign to TEMP.Hex, a China-linked cyber espionage actor. TEMP.Hex likely conducted these attacks to collect information in support of Chinese national security and economic interests. These operations pose a risk to a variety of industries, including construction and engineering, business services, government, health, transportation, and retail in Europe, Asia, and the United States.

# Prior publications, hi 45.142.166.112!



## A border-hopping PlugX USB worm takes its act on the road

Borne aloft by DLL sideloading, a far-flung infection touches ten time zones

*Figure 1: An unusual distribution of infections is the hallmark of a new PlugX variant that relies on DLL sideloading to propagate*

## The Spies Who Loved You: Infected USB Drives to Steal Secrets

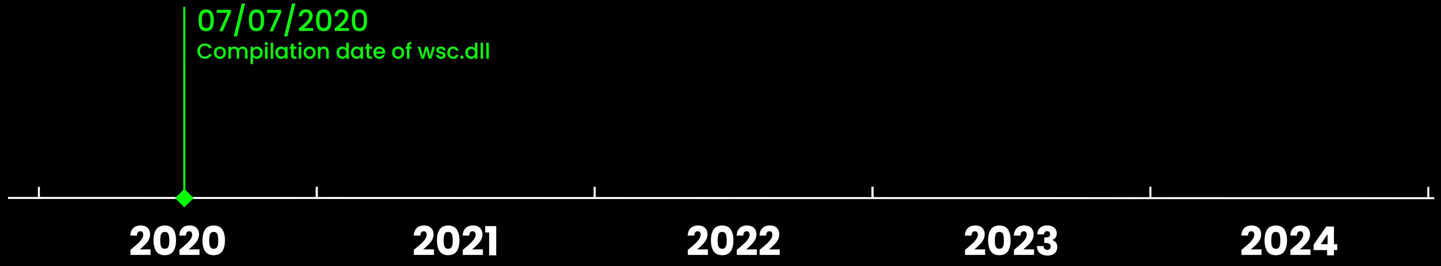
In the first half of 2023, we observed a threefold increase in the number of attacks using infected USB drives to steal secrets.

2023, Mandiant Managed Defense has observed a threefold increase ttacks using infected USB drives to steal secrets. Mandiant tracked all und that the majority of the incidents could be attributed to several igns affecting both the public and private sectors

campaigns that leverages USB flash drives as an rates on the Philippines. In this blog post, we are covering two additional USB-based cyber espionage campaigns that have been observed by Managed Defense:

- **SOGU Malware Infection via USB Flash Drives Across Industries and Geographies**This is the most prevalent USB-based cyber espionage attack using USB flash drives and one of the most aggressive cyber espionage campaigns targeting both public and private sector organizations globally across industry verticals. It uses USB flash drives to load the SOGU malware to steal sensitive information from a host.Mandiant attributes this campaign to TEMP.Hex, a China-linked cyber espionage actor. TEMP.Hex likely conducted these attacks to collect information in support of Chinese national security and economic interests. These operations pose a risk to a variety of industries, including construction and engineering, business services, government, health, transportation, and retail in Europe, Asia, and the United States.

> “We then saw C2 activity reaching out to multiple variations on the IP address **45.142.166.112**”



07/07/2020

Compilation date of wsc.dll

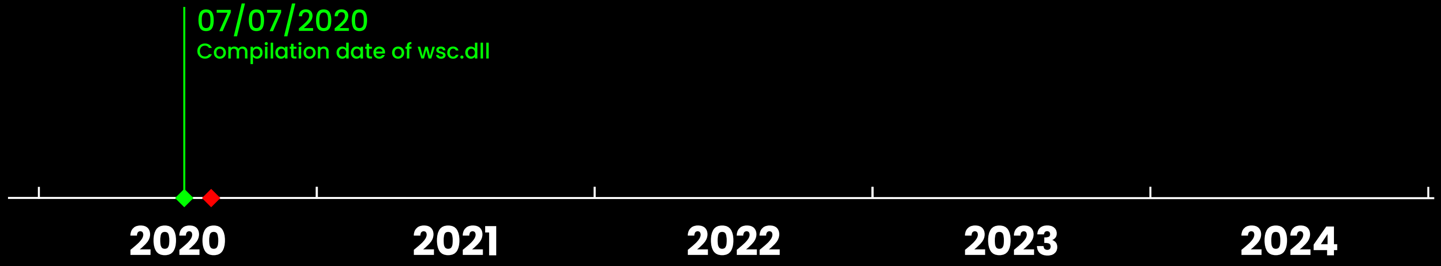
2020

2021

2022

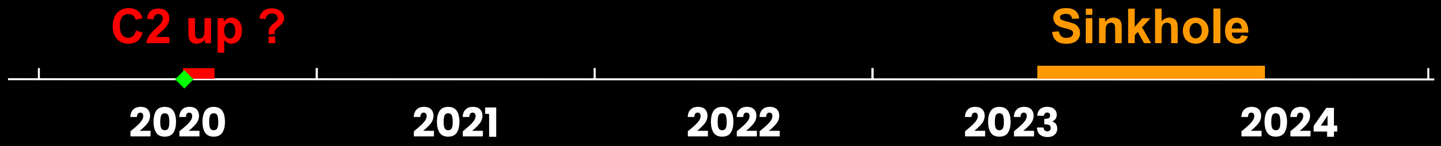
2023

2024



07/07/2020

Compilation date of wsc.dll



C2 up ?

Sinkhole

2020

2021

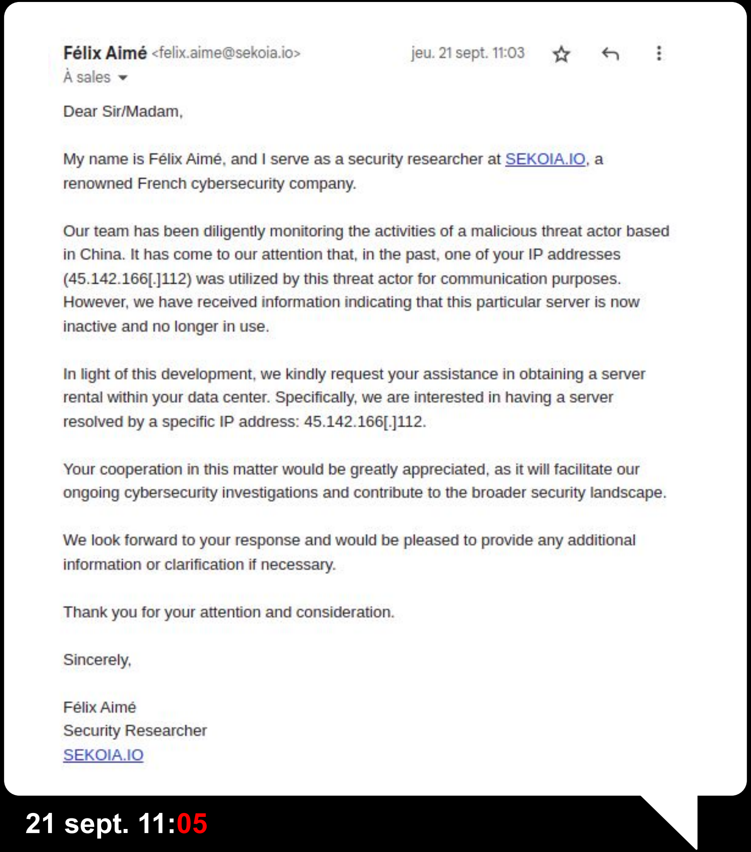
2022

2023

2024

# Sinkholing 45.142.166.112 for the price of a 🍺.

```
nmap -sP 45.142.166.112
Starting Nmap 7.80 ( https://nmap.org ) at 2023-09-21 10:32 CEST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.01 seconds
```



Félix Aimé <felix.aime@sekoia.io> jeu. 21 sept. 11:03

Dear Sir/Madam,

My name is Félix Aimé, and I serve as a security researcher at SEKOIA.IO, a renowned French cybersecurity company.

Our team has been diligently monitoring the activities of a malicious threat actor based in China. It has come to our attention that, in the past, one of your IP addresses (45.142.166[.]112) was utilized by this threat actor for communication purposes. However, we have received information indicating that this particular server is now inactive and no longer in use.

In light of this development, we kindly request your assistance in obtaining a server rental within your data center. Specifically, we are interested in having a server resolved by a specific IP address: 45.142.166[.]112.

Your cooperation in this matter would be greatly appreciated, as it will facilitate our ongoing cybersecurity investigations and contribute to the broader security landscape.

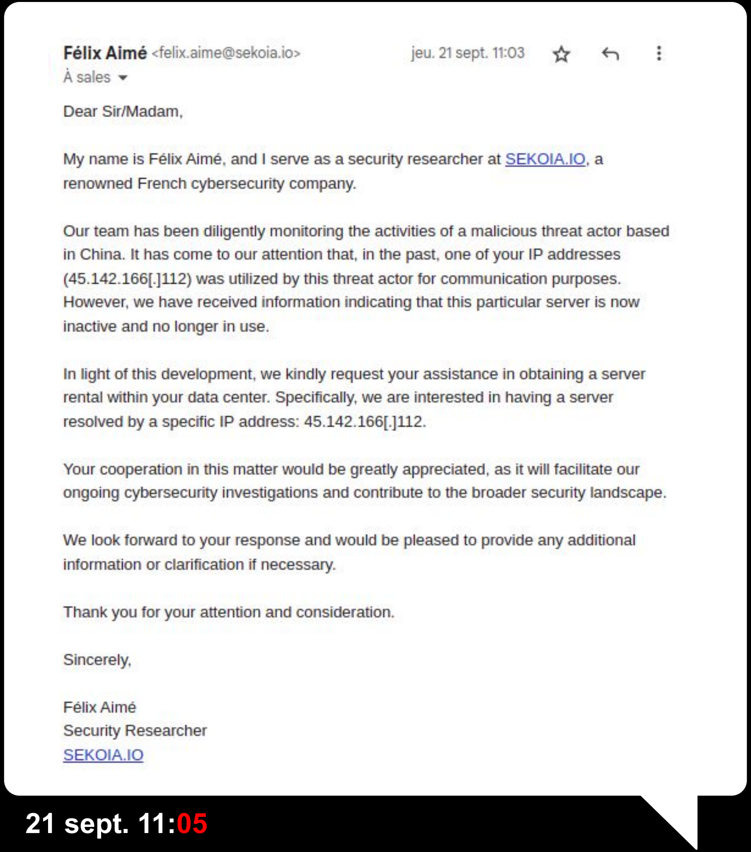
We look forward to your response and would be pleased to provide any additional information or clarification if necessary.

Thank you for your attention and consideration.

Sincerely,

Félix Aimé
Security Researcher
SEKOIA.IO

21 sept. 11:05



**Félix Aimé** <felix.aime@sekoia.io> jeu. 21 sept. 11:03

À sales

Dear Sir/Madam,

My name is Félix Aimé, and I serve as a security researcher at SEKOIA.IO, a renowned French cybersecurity company.

Our team has been diligently monitoring the activities of a malicious threat actor based in China. It has come to our attention that, in the past, one of your IP addresses (45.142.166[.]112) was utilized by this threat actor for communication purposes. However, we have received information indicating that this particular server is now inactive and no longer in use.

In light of this development, we kindly request your assistance in obtaining a server rental within your data center. Specifically, we are interested in having a server resolved by a specific IP address: 45.142.166[.]112.

Your cooperation in this matter would be greatly appreciated, as it will facilitate our ongoing cybersecurity investigations and contribute to the broader security landscape.

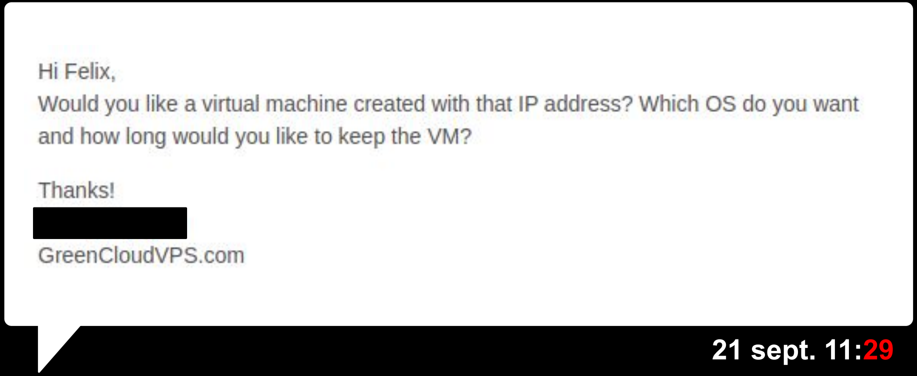
We look forward to your response and would be pleased to provide any additional information or clarification if necessary.

Thank you for your attention and consideration.

Sincerely,

Félix Aimé
Security Researcher
SEKOIA.IO

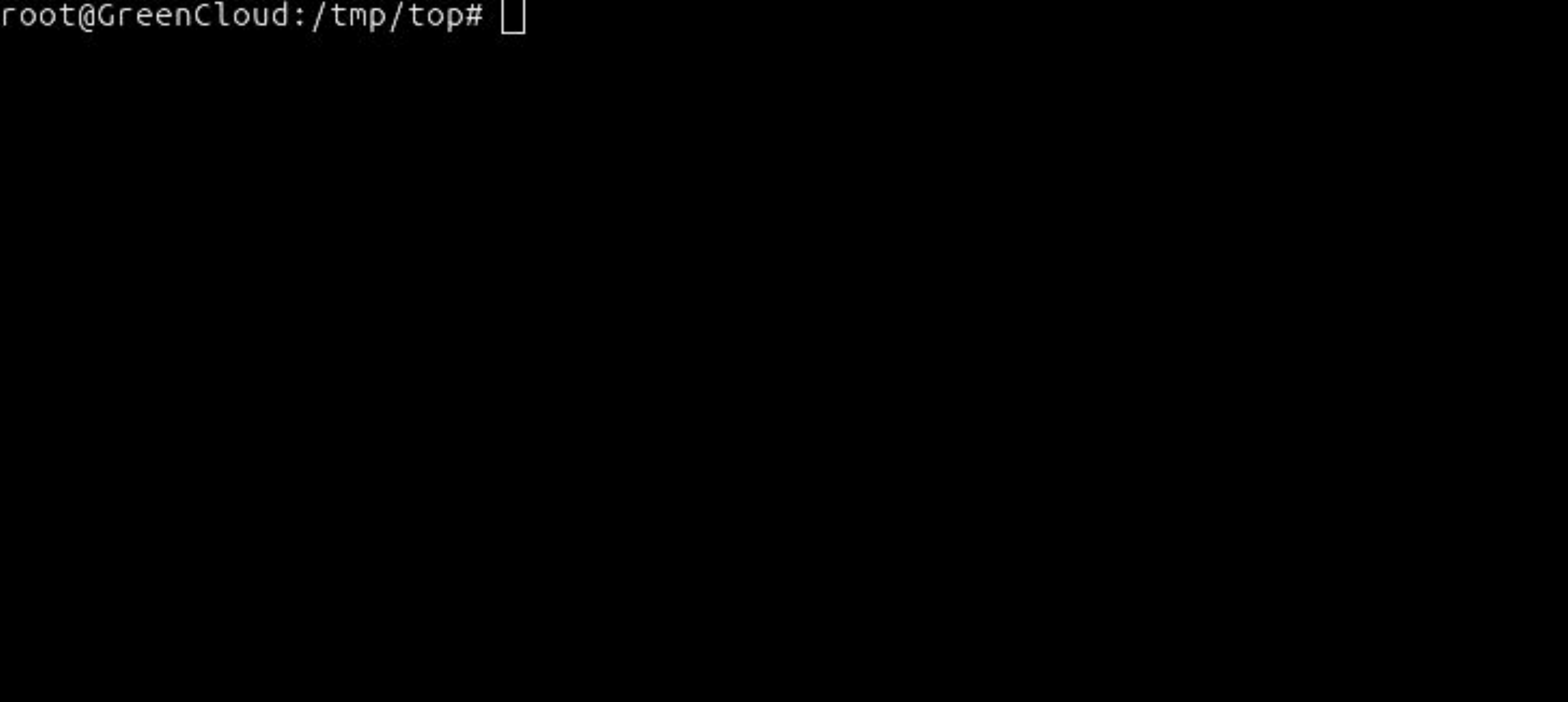
**21 sept. 11:05**



Hi Felix,
Would you like a virtual machine created with that IP address? Which OS do you want and how long would you like to keep the VM?

Thanks!
[redacted]
GreenCloudVPS.com

**21 sept. 11:29**



```
root@GreenCloud:/tmp/top# 
```

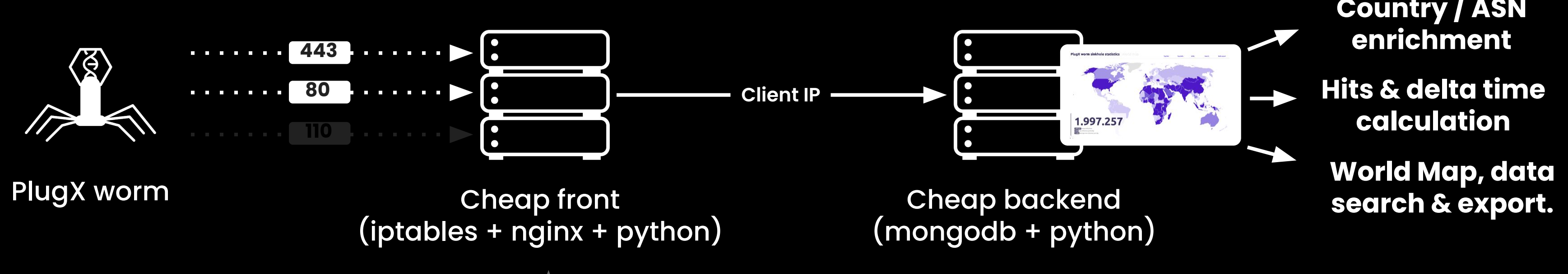
**~1000 requests per seconds**

# Received requests

## Typical PlugX requests received on the 443 (http, raw), 80 (http) and 110 (raw)

```
POST /[a-f0-9]{8}
Accept: */*
jsp-se: 0
jsp-st: 0
jsp-si: 61456
jsp-sn: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0;Win64;x64)AppleWebKit/537.36
Host: <ip>:443
Content-Length: 0
Connection: Keep-Alive
Cache-Control: no-cache
```

# A cheap sinkhole, which works.



PlugX worm

Cheap front
(iptables + nginx + python)

Cheap backend
(mongodb + python)

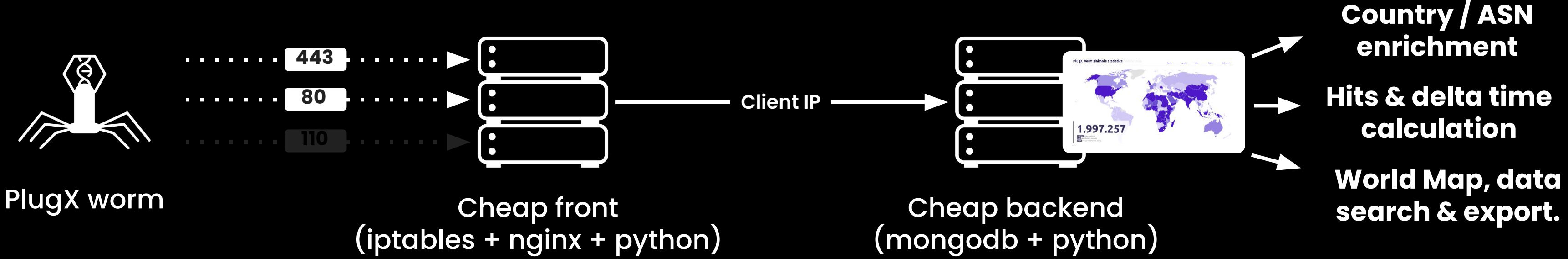
**Country / ASN enrichment**

**Hits & delta time calculation**

**World Map, data search & export.**

- Use of IPTables rate limiting
- Check of right PlugX headers
- Forward the client IP to the backend.

# A cheap sinkhole, which works.



443

80

110

PlugX worm

Cheap front
(iptables + nginx + python)

Client IP

PlugX worm sinkhole statistics

1.997.257

Cheap backend
(mongodb + python)

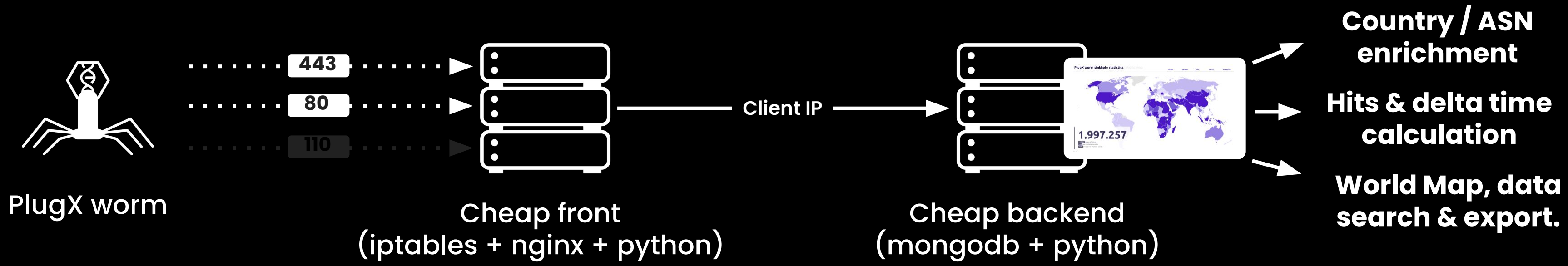
**Country / ASN enrichment**

**Hits & delta time calculation**

**World Map, data search & export.**

- Use of IPTables rate limiting
- Check of right PlugX headers
- Forward the client IP to the backend.

# A cheap sinkhole, which works.



PlugX worm

Cheap front
(iptables + nginx + python)

Client IP

Cheap backend
(mongodb + python)

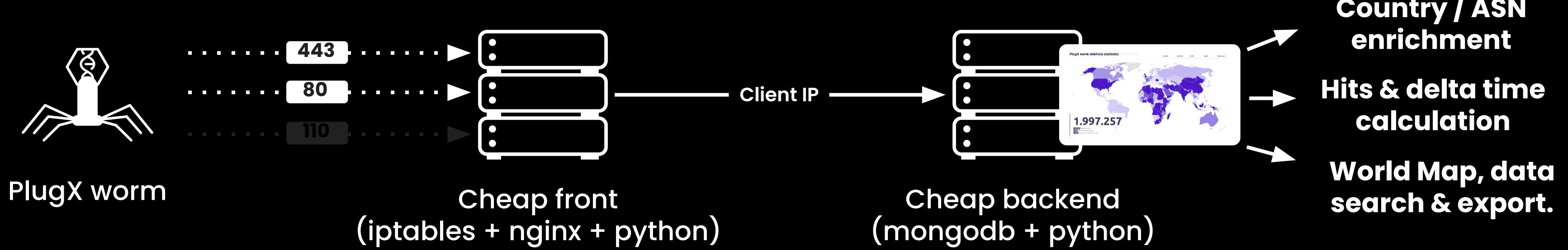
**Country / ASN enrichment**

**Hits & delta time calculation**

**World Map, data search & export.**

- Use of IPTables rate limiting
- Check of right PlugX headers
- Forward the client IP to the backend

# A cheap sinkhole, which works.



PlugX worm

443

80

110

Cheap front
(iptables + nginx + python)

Client IP

Cheap backend
(mongodb + python)

1.997.257

Country / ASN enrichment

Hits & delta time calculation

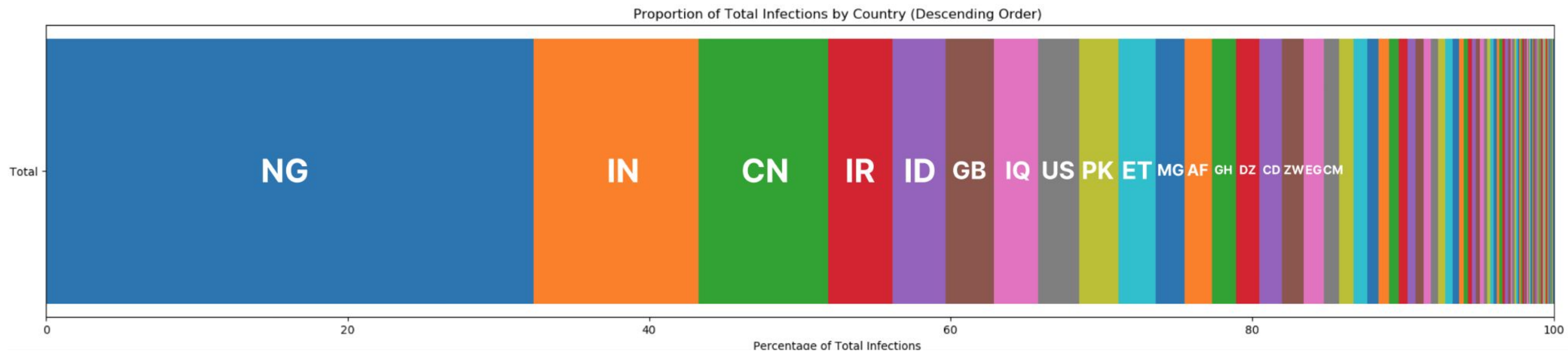
World Map, data search & export.

- Use of IPTables rate limiting
- Check of right PlugX headers.
- Forward the client IP to the backend.

# Some observations, after 6 months.

**+ 2.5M unique IPs** from +170 countries

**~ 90-100K  IPs seen per day** (so, essentially a small botnet)

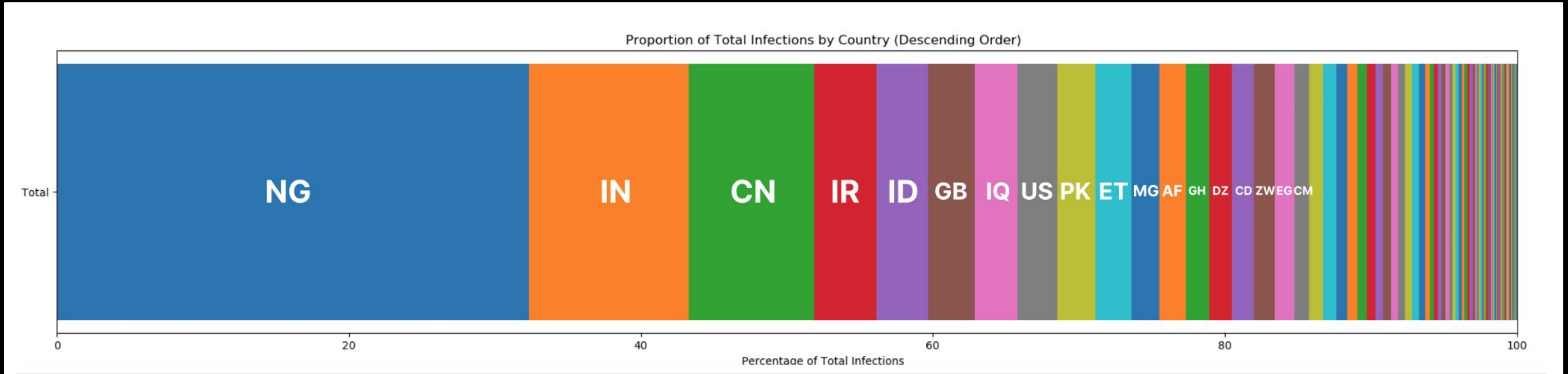


**Percentage per country, for ~100K IPs retrieved during one day - 152 countries**

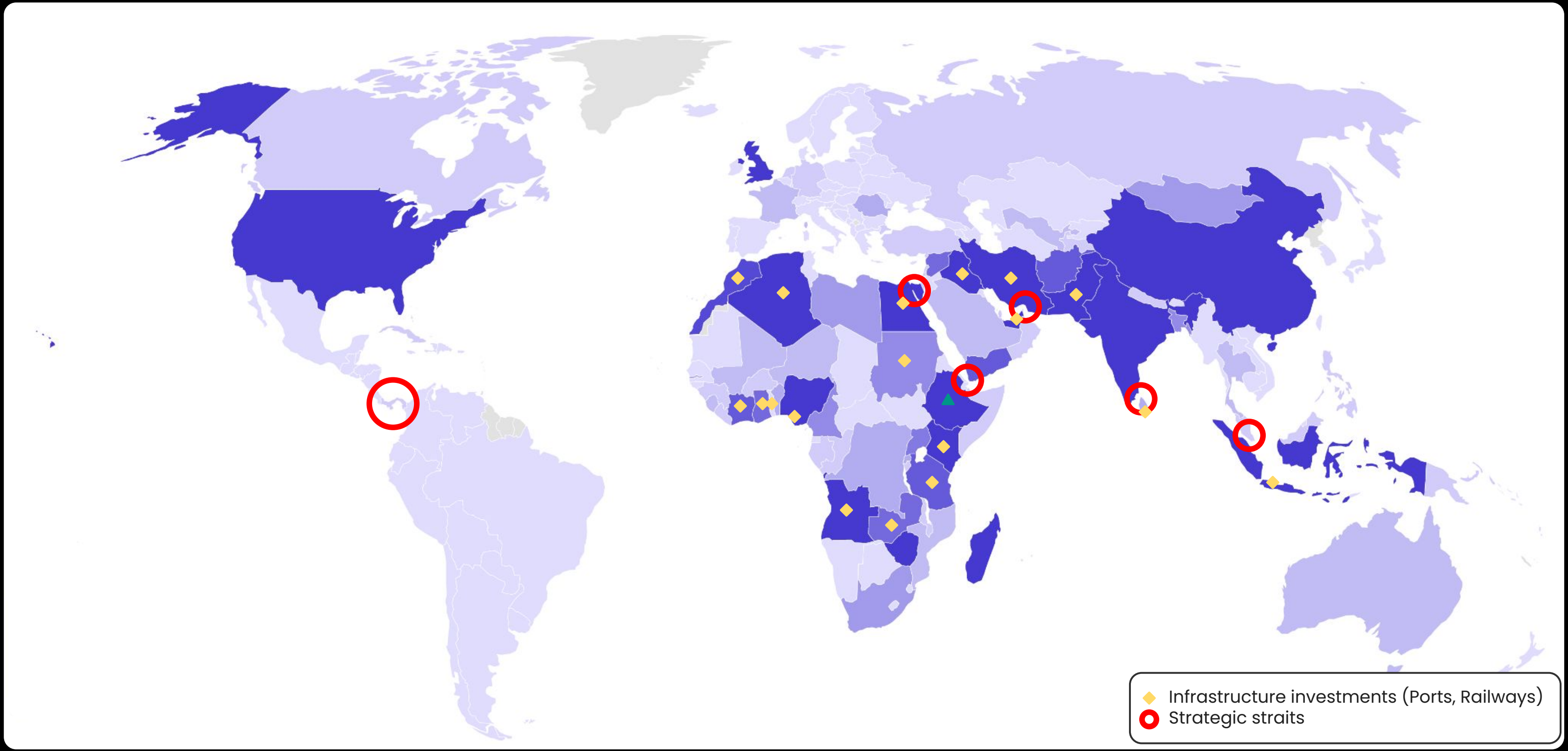
# Some observations, after 6 months.

**+ 2.5M unique IPs** from +170 countries

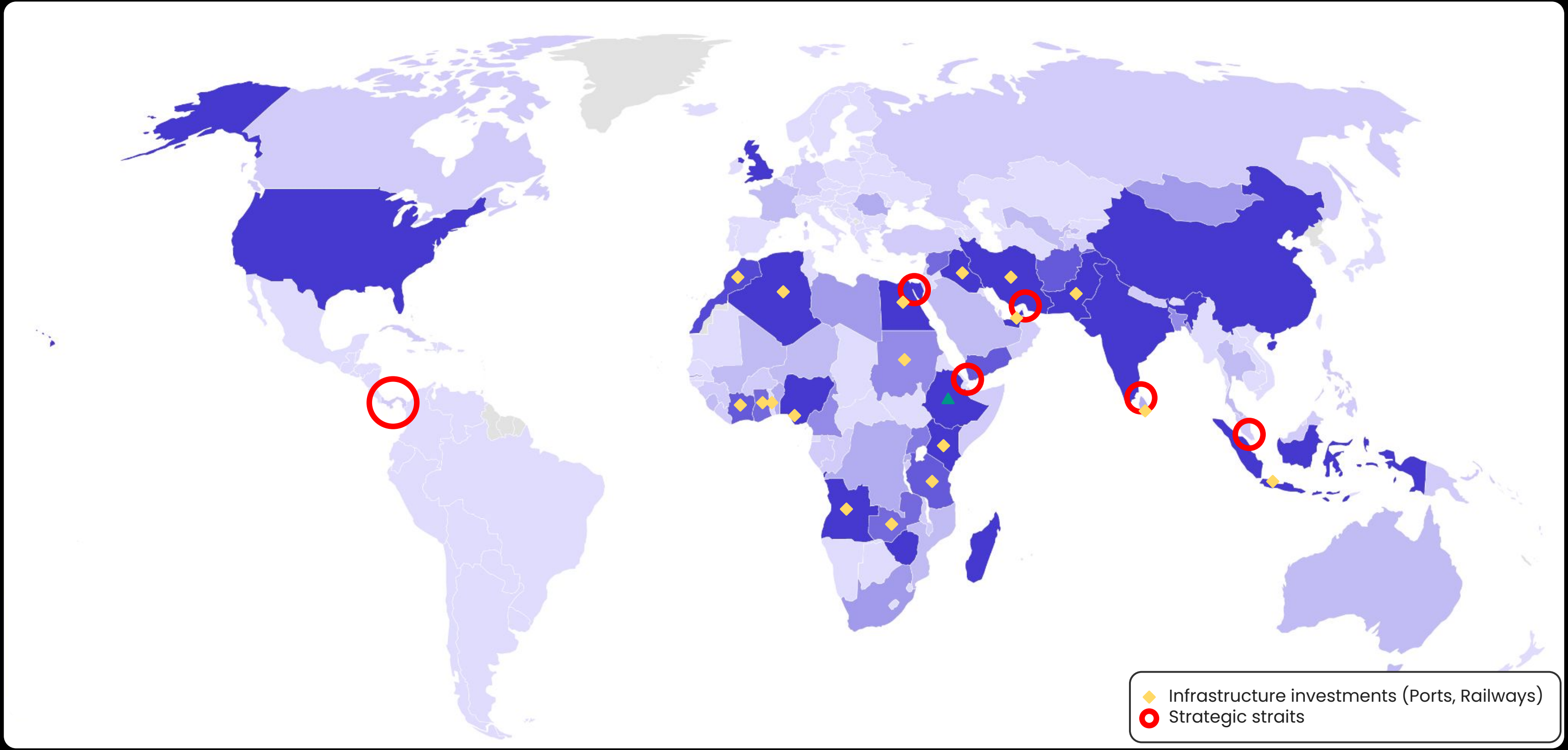
**~ 90-100K IPs seen per day** (so, essentially a small botnet)



**Percentage per country, for ~100K IPs retrieved during one day - 152 countries**



A possible implication in the **security of CN investments**?



A possible implication in the **security of CN investments**?

(hard to say as China invests **everywhere**... and the worm is **4 years old**.)

# Looking at remote disinfection opportunities.

# Why disinfect & initial idea

**A dead botnet might not be truly dead and can be repurposed**

> This example demonstrates a $7 IP takeover.

> IP takeover can occur at various levels.

**Propose sovereign disinfection to LEAs & National CERTs**

> Allow them to carry it out via an interface for specific ASNs.

> But, remember: it's still a worm.

## Ok, but how to achieve that?

# Why disinfect & initial idea

**A dead botnet might not be truly dead and can be repurposed**

> This example demonstrates a $7 IP takeover.
> IP takeover can occur at various levels.

**Propose sovereign disinfection to LEAs & National CERTs**

> Allow them to carry it out via an interface for specific ASNs.
> But, remember: it's still a worm.

## Ok, but how to achieve that?

# Why disinfect & initial idea

**A dead botnet might not be truly dead and can be repurposed**

> This example demonstrates a $7 IP takeover.

> IP takeover can occur at various levels.

**Propose sovereign disinfection to LEAs & National CERTs**

> Allow them to carry it out via an interface for specific ASNs.

> But, remember: it's still a worm.

## Ok, but how to achieve that?

# Why disinfect & initial idea

**A dead botnet might not be truly dead and can be repurposed**

> This example demonstrates a $7 IP takeover.

> IP takeover can occur at various levels.

**Propose sovereign disinfection to LEAs & National CERTs**

> Allow them to carry it out via an interface for specific ASNs.

> But, remember: it's still a worm.

**Ok, but how to achieve that?**

# Initial questions.

How does its com’s encryption work?

Can a workstation be disinfected remotely?

Can both the workstation and a flash drive be disinfected?

# Initial questions.

How does its com’s encryption work?

Can a workstation be disinfected remotely?

Can both the workstation and a flash drive be disinfected?

# Initial questions.

How does its com’s encryption work?

Can a workstation be disinfected remotely?

Can both the workstation and a flash drive be disinfected?

# Initial questions.

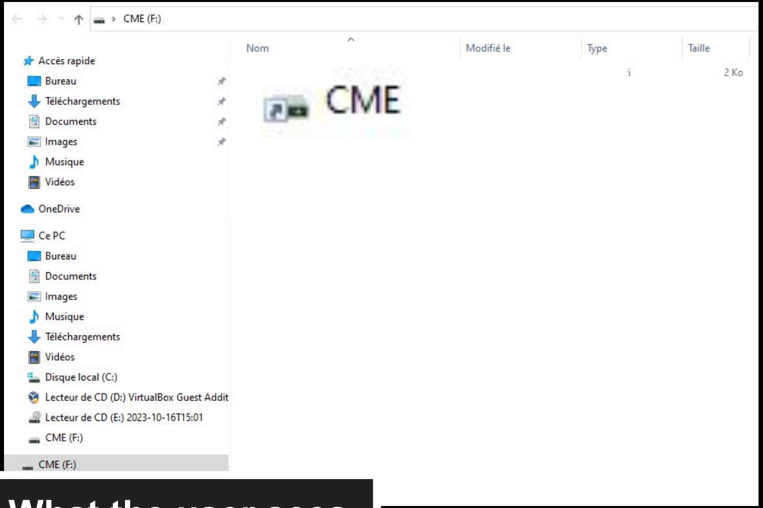
How does its com’s encryption work?

Can a workstation be disinfected remotely?

Can both the workstation and a flash drive be disinfected?

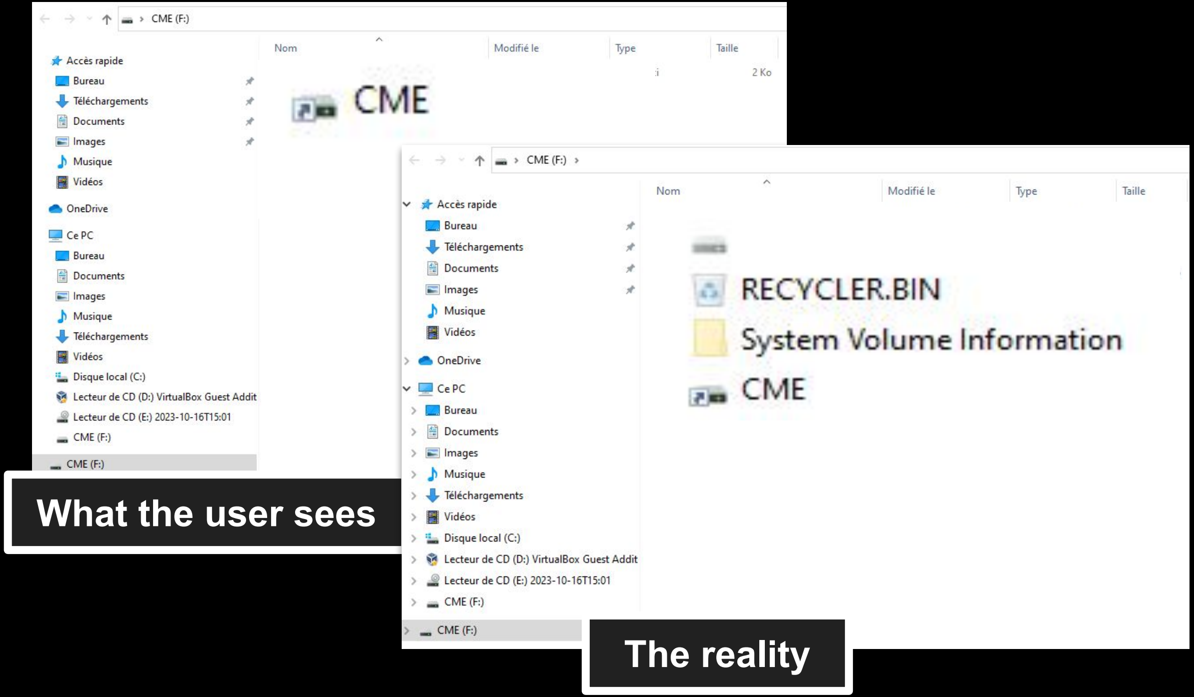
# 2 cents on PlugX worm

# 2 cents on PlugX worm



What the user sees

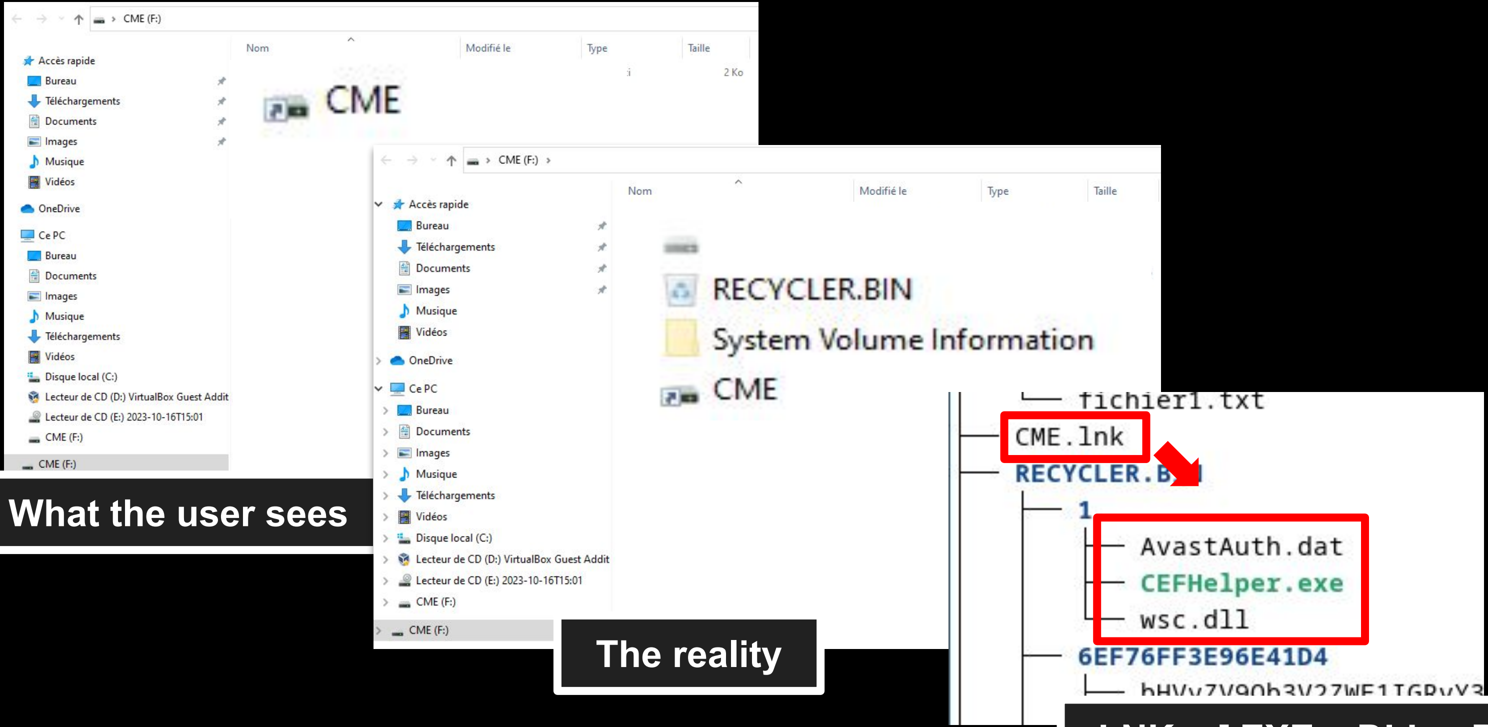
# 2 cents on PlugX worm



What the user sees

The reality

# 2 cents on PlugX worm



CME

What the user sees

RECYCLER.BIN

System Volume Information

CME

The reality

```
   └── fichier1.txt
├── CME.lnk
├── RECYCLER.BIN
│   ├── 1
│   │   ├── AvastAuth.dat
│   │   ├── CEFHelper.exe
│   │   └── wsc.dll
│   ├── 6EF76FF3E96E41D4
│   │   └── bHVyZV90b3V2ZWF1IGRvY3
```

LNK > [ EXE > DLL > BIN ]

# 2 cents on PlugX worm

**Once PlugX is executed from the Flash drive**

> Redirects the user to the hidden “data” directory
> Copies itself to the workstation
> Adds persistence
> Restarts itself from the workstation

# 2 cents on PlugX worm

**When executed from the workstation:**

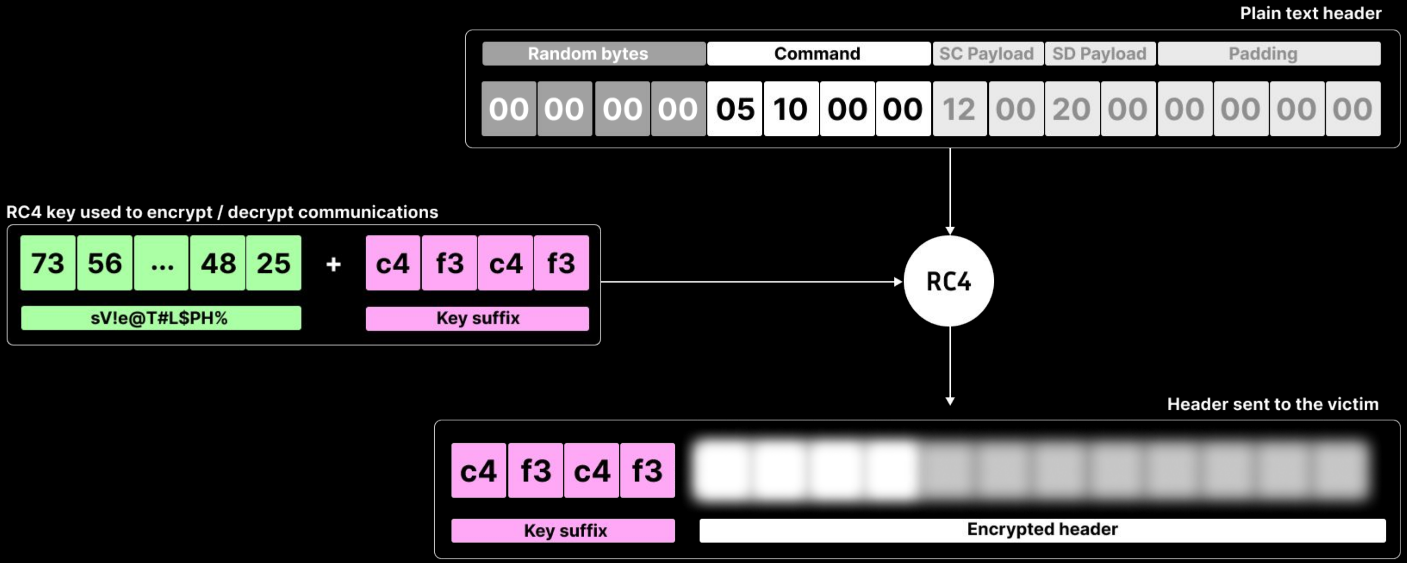
> Communicates with the C2 & awaits commands
> Automatically infects other flash drives

> Copies certain files to a hidden directory onto the Flash drive (air gap functionality)

# PlugX (weak) crypto coms.

## Use of RC4 for C2 communications:

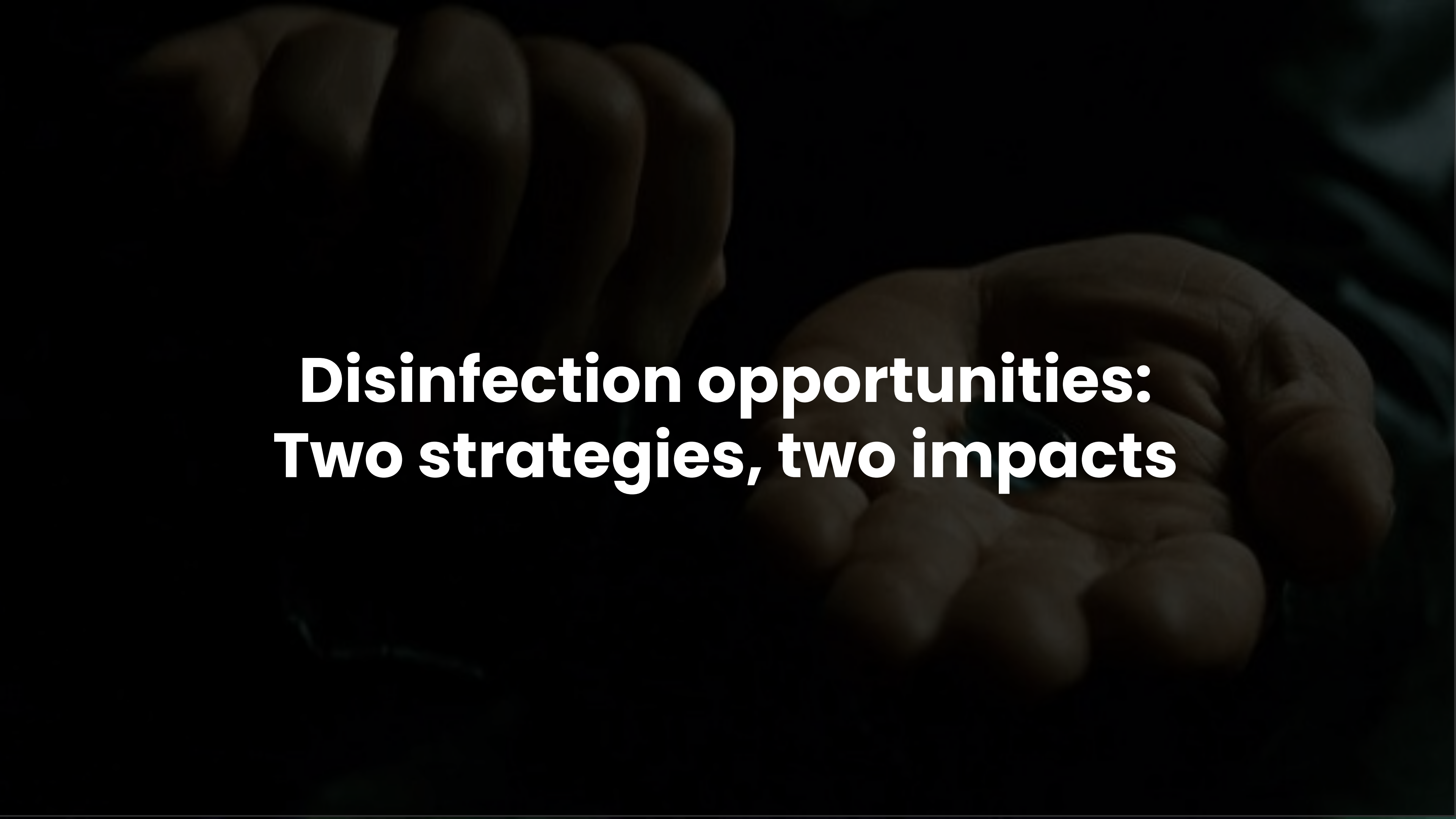
First part of the key hardcoded in the sample (**we know**)
Second part of the key provided by the C2 (**we control**)



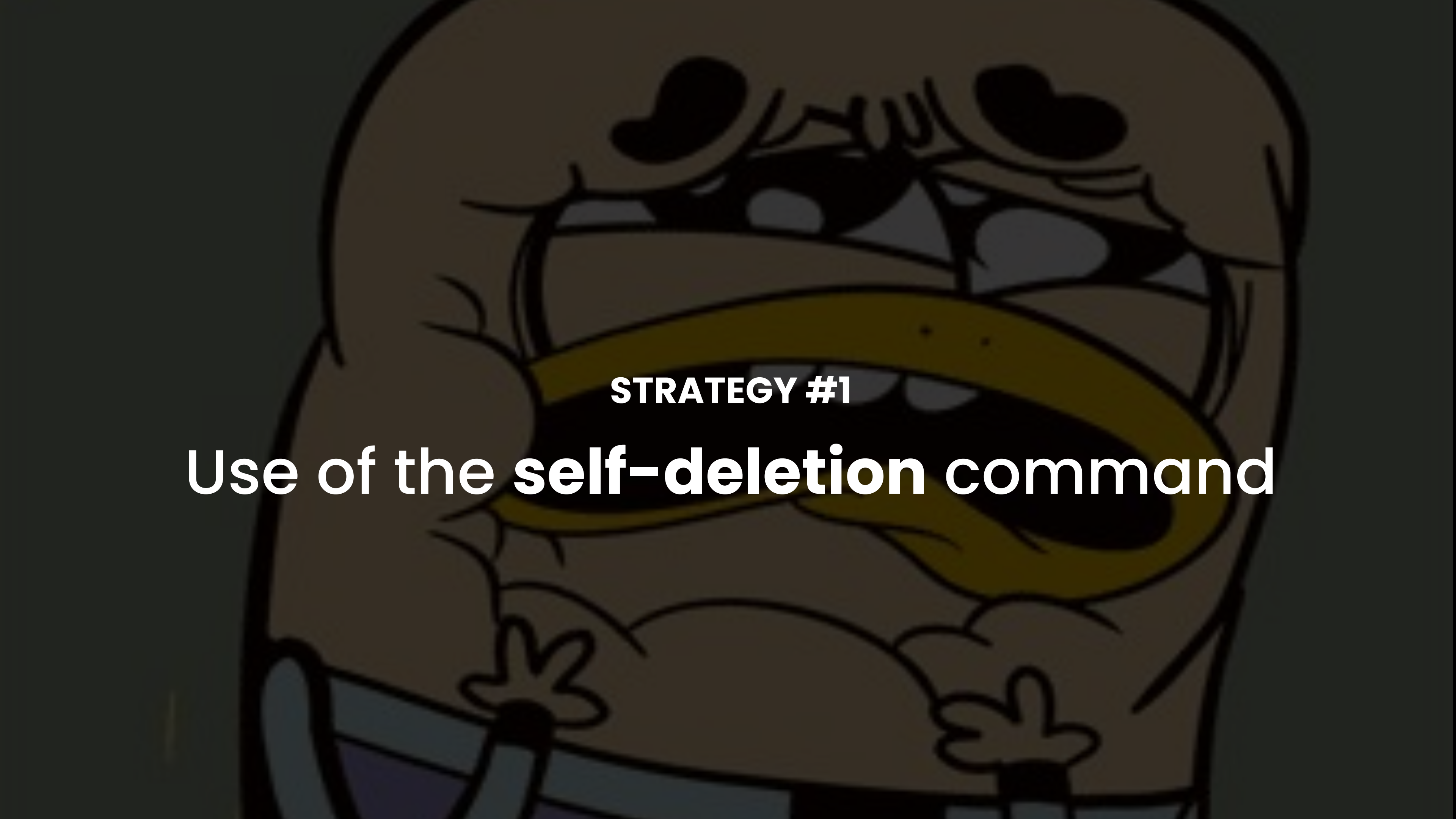
# PlugX (weak) crypto coms.

**Weak cryptography**

> No public key cryptography nor certificate pinning
> Easy to interact with workstations infected by PlugX



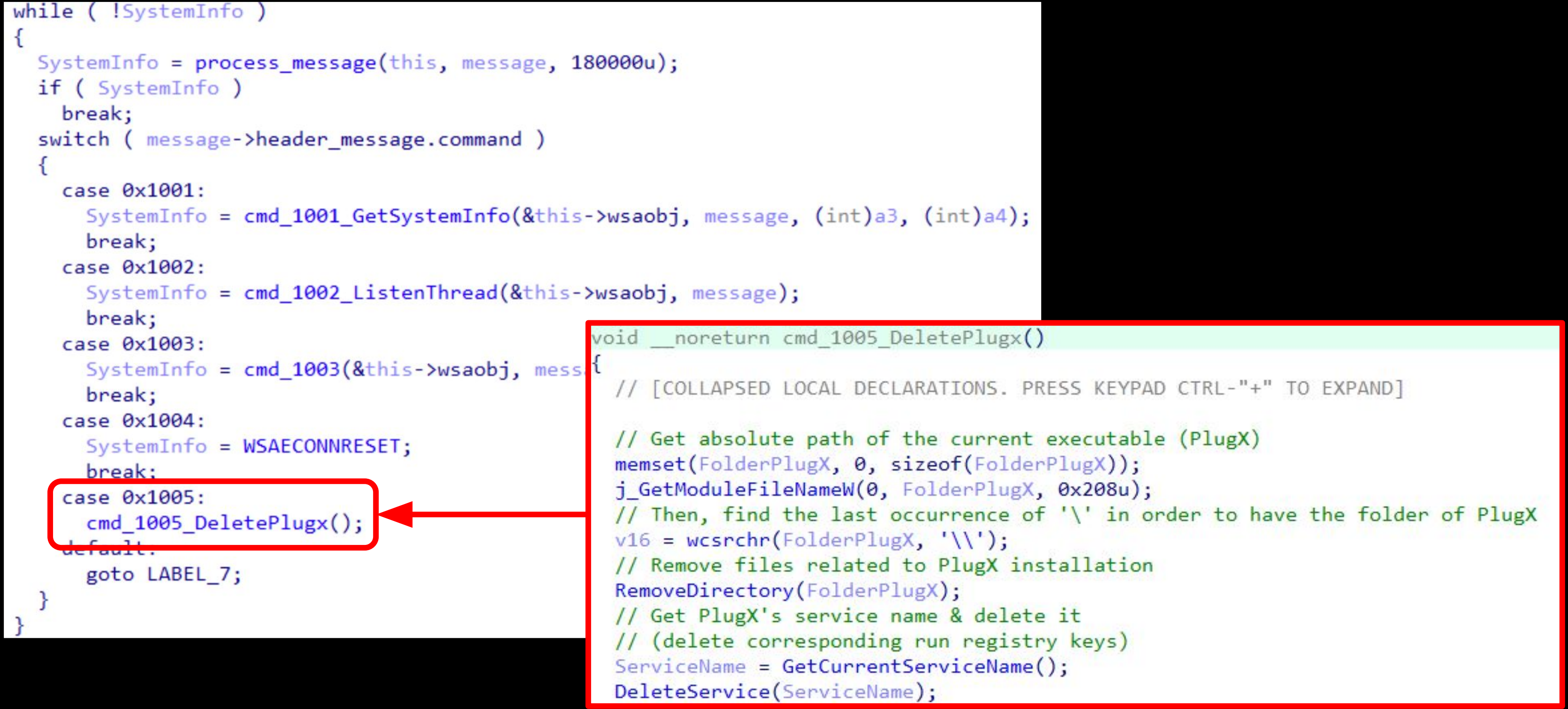
# Disinfection opportunities: Two strategies, two impacts



STRATEGY #1

# Use of the **self-deletion** command

# Deletion command (0x1005)

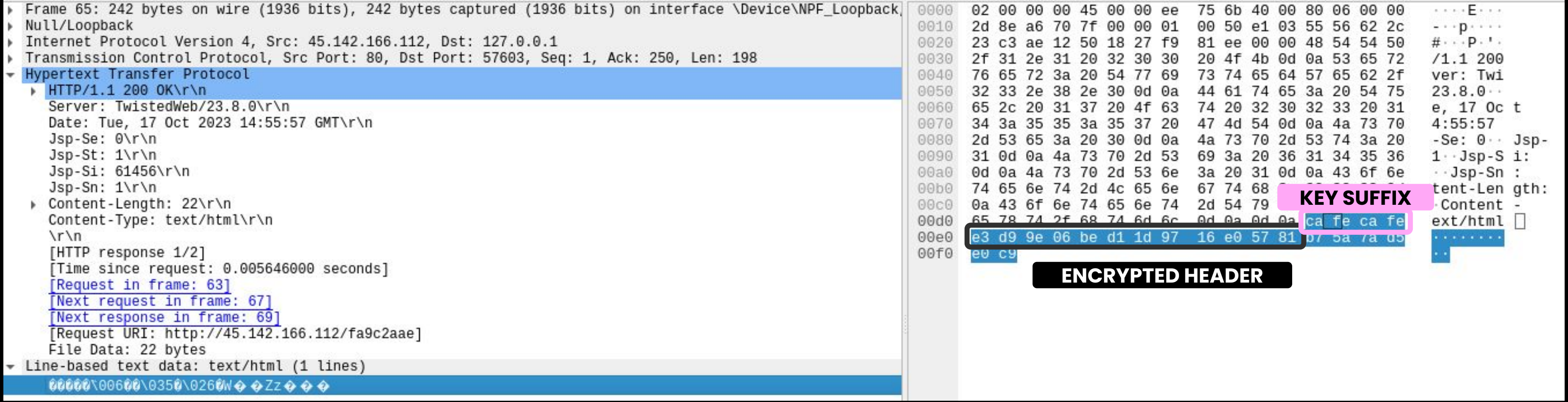


```
while ( !SystemInfo )
{
  SystemInfo = process_message(this, message, 180000u);
  if ( SystemInfo )
    break;
  switch ( message->header_message.command )
  {
    case 0x1001:
      SystemInfo = cmd_1001_GetSystemInfo(&this->wsaobj, message, (int)a3, (int)a4);
      break;
    case 0x1002:
      SystemInfo = cmd_1002_ListenThread(&this->wsaobj, message);
      break;
    case 0x1003:
      SystemInfo = cmd_1003(&this->wsaobj, mess
      break;
    case 0x1004:
      SystemInfo = WSAECONNRESET;
      break;
    case 0x1005:
      cmd_1005_DeletePlugx();
    default:
      goto LABEL_7;
  }
}
```

```
void __noreturn cmd_1005_DeletePlugx()
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  // Get absolute path of the current executable (PlugX)
  memset(FolderPlugX, 0, sizeof(FolderPlugX));
  j_GetModuleFileNameW(0, FolderPlugX, 0x208u);
  // Then, find the last occurrence of '\' in order to have the folder of PlugX
  v16 = wcsrchr(FolderPlugX, '\\');
  // Remove files related to PlugX installation
  RemoveDirectory(FolderPlugX);
  // Get PlugX's service name & delete it
  // (delete corresponding run registry keys)
  ServiceName = GetCurrentServiceName();
  DeleteService(ServiceName);
```

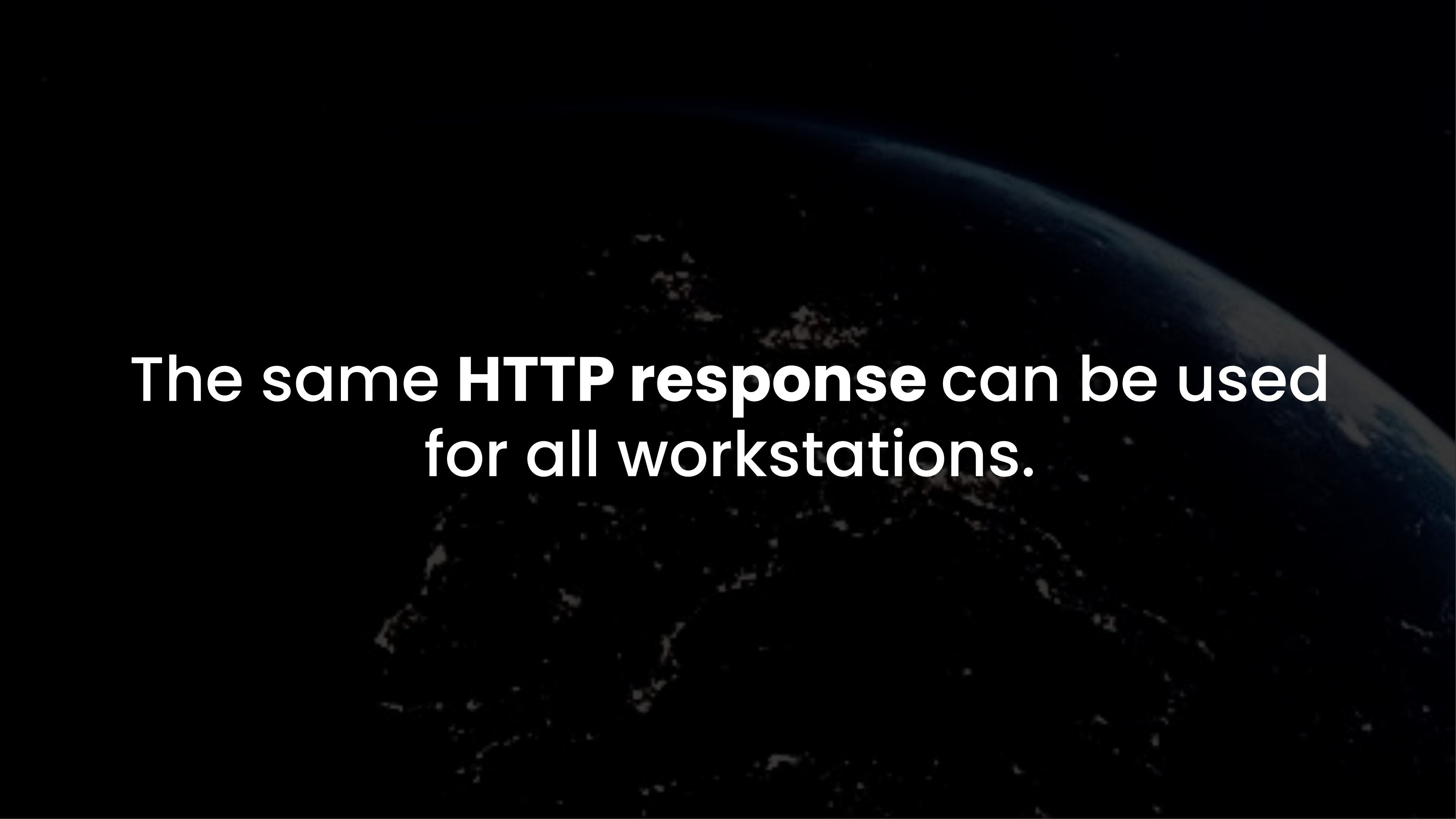
# One response example



```
Frame 65: 242 bytes on wire (1936 bits), 242 bytes captured (1936 bits) on interface \Device\NPF_Loopback,
Null/Loopback
Internet Protocol Version 4, Src: 45.142.166.112, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 80, Dst Port: 57603, Seq: 1, Ack: 250, Len: 198
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
  Server: TwistedWeb/23.8.0\r\n
  Date: Tue, 17 Oct 2023 14:55:57 GMT\r\n
  Jsp-Se: 0\r\n
  Jsp-St: 1\r\n
  Jsp-Si: 61456\r\n
  Jsp-Sn: 1\r\n
  Content-Length: 22\r\n
  Content-Type: text/html\r\n
  \r\n
  [HTTP response 1/2]
  [Time since request: 0.005646000 seconds]
  [Request in frame: 63]
  [Next request in frame: 67]
  [Next response in frame: 69]
  [Request URI: http://45.142.166.112/fa9c2aae]
  File Data: 22 bytes
Line-based text data: text/html (1 lines)
  �����\006��\035�\026�W� �Zz� � �
```

```
0000  02 00 00 00 45 00 00 ee  75 6b 40 00 80 06 00 00   ····E···
0010  2d 8e a6 70 7f 00 00 01  00 50 e1 03 55 56 62 2c   -··p····
0020  23 c3 ae 12 50 18 27 f9  81 ee 00 00 48 54 54 50   #···P·'·
0030  2f 31 2e 31 20 32 30 30  20 4f 4b 0d 0a 53 65 72   /1.1 200
0040  76 65 72 3a 20 54 77 69  73 74 65 64 57 65 62 2f   ver: Twi
0050  32 33 2e 38 2e 30 0d 0a  44 61 74 65 3a 20 54 75   23.8.0··
0060  65 2c 20 31 37 20 4f 63  74 20 32 30 32 33 20 31   e, 17 Oc t
0070  34 3a 35 35 3a 35 37 20  47 4d 54 0d 0a 4a 73 70   4:55:57
0080  2d 53 65 3a 20 30 0d 0a  4a 73 70 2d 53 74 3a 20   -Se: 0·· Jsp-
0090  31 0d 0a 4a 73 70 2d 53  69 3a 20 36 31 34 35 36   1··Jsp-S i:
00a0  0d 0a 4a 73 70 2d 53 6e  3a 20 31 0d 0a 43 6f 6e   ··Jsp-Sn :
00b0  74 65 6e 74 2d 4c 65 6e  67 74 68                  tent-Len gth:
00c0  0a 43 6f 6e 74 65 6e 74  2d 54 79                  ·Content -
00d0  65 78 74 2f 68 74 6d 6c  0d 0a 0d 0a ca fe ca fe   ext/html □
00e0  e3 d9 9e 06 be d1 1d 97  16 e0 57 81 b7 5a 7a d5   ········
00f0  e0 c9                                              ··
```

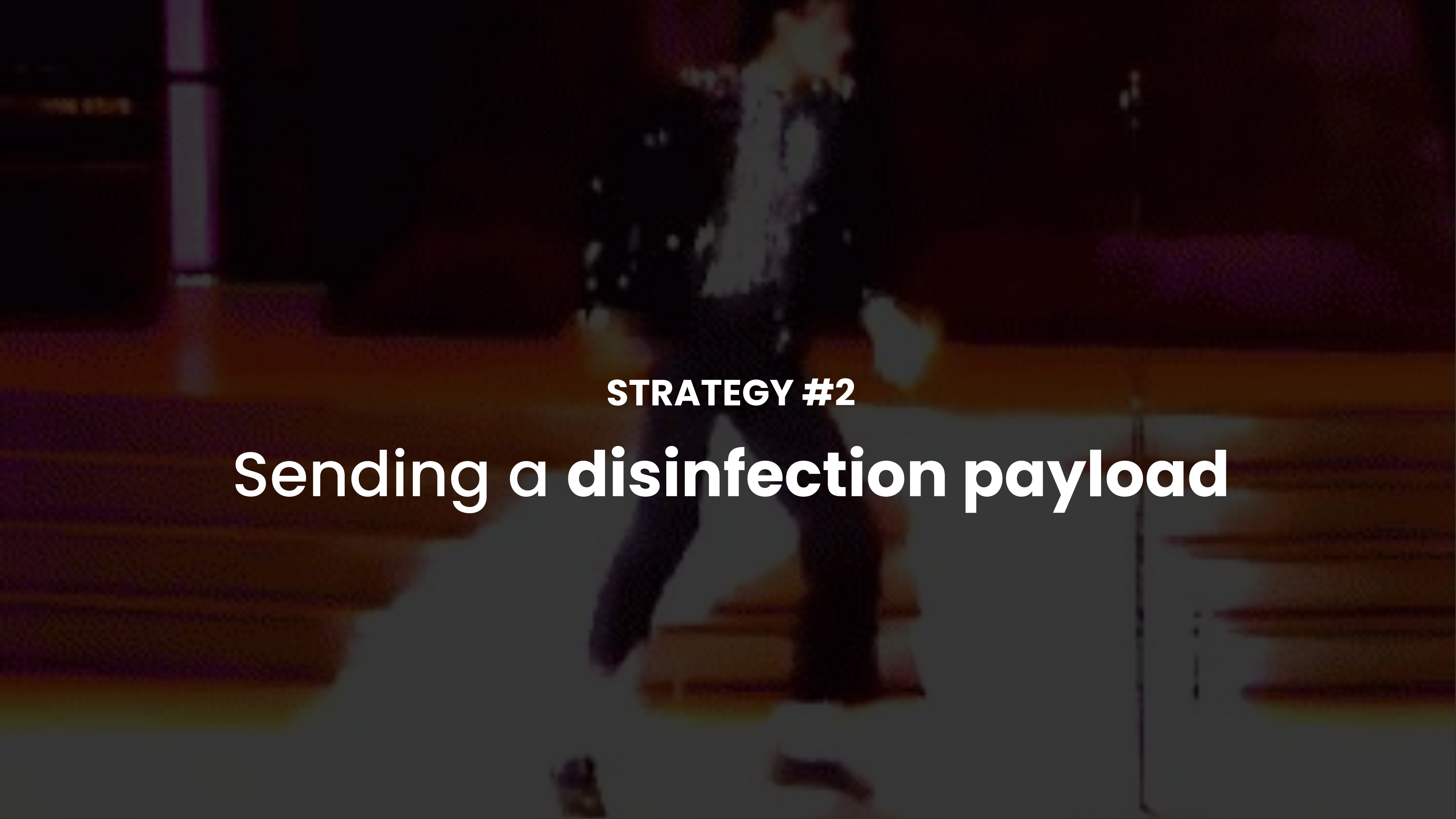
Just one **HTTP response** suffices in allowing workstation disinfection.



The same **HTTP response** can be used for all workstations.

...but the flash drive

**remains infected ;]**



STRATEGY #2

# Sending a **disinfection payload**

# Sending a disinfection payload

**The payload :**

- Disinfects workstation
- If a flash drive is plugged in & infected:
  - Removes PlugX binaries & staged data
  - Moves the “data” directory to the drive's root

**Tips:**

- Our payload shares many similarities with PlugX
- To reuse the PlugX code

# Sending a disinfection payload

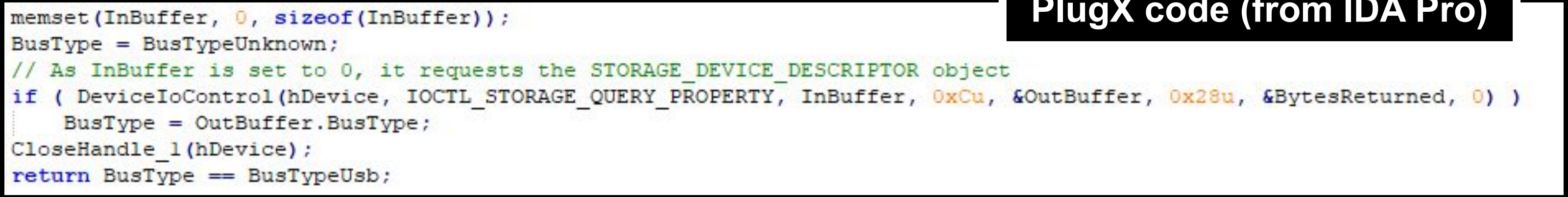
**The payload :**

- Disinfects workstation
- If a flash drive is plugged in & infected:
  - Removes PlugX binaries & staged data
  - Moves the “data” directory to the drive's root

**Tips:**

- Our payload shares many similarities with PlugX
- To reuse the PlugX code

# Find the differences

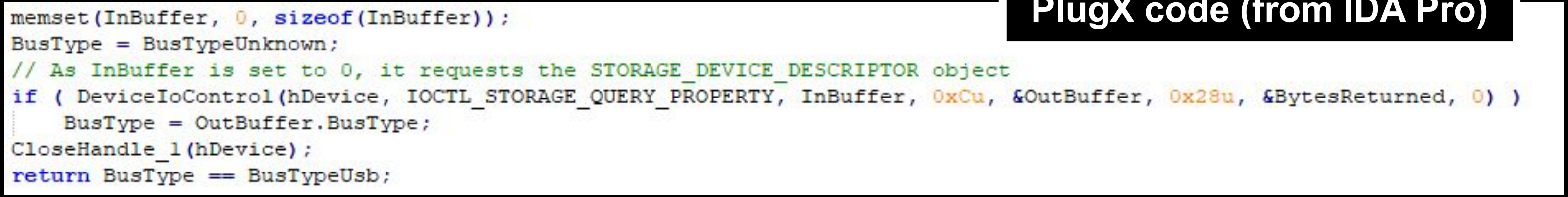


PlugX code (from IDA Pro)

```
memset(InBuffer, 0, sizeof(InBuffer));
BusType = BusTypeUnknown;
// As InBuffer is set to 0, it requests the STORAGE_DEVICE_DESCRIPTOR object
if ( DeviceIoControl(hDevice, IOCTL_STORAGE_QUERY_PROPERTY, InBuffer, 0xCu, &OutBuffer, 0x28u, &BytesReturned, 0) )
    BusType = OutBuffer.BusType;
CloseHandle_1(hDevice);
return BusType == BusTypeUsb;
```

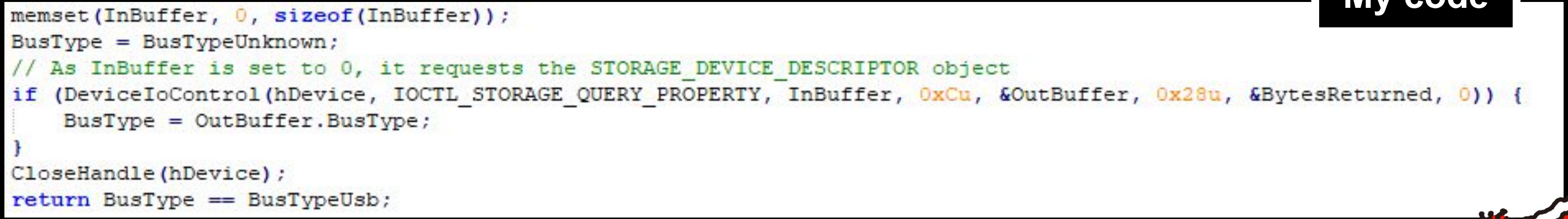
# Find the differences

PlugX code (from IDA Pro)



```
memset(InBuffer, 0, sizeof(InBuffer));
BusType = BusTypeUnknown;
// As InBuffer is set to 0, it requests the STORAGE_DEVICE_DESCRIPTOR object
if ( DeviceIoControl(hDevice, IOCTL_STORAGE_QUERY_PROPERTY, InBuffer, 0xCu, &OutBuffer, 0x28u, &BytesReturned, 0) )
    BusType = OutBuffer.BusType;
CloseHandle_1(hDevice);
return BusType == BusTypeUsb;
```

My code



```
memset(InBuffer, 0, sizeof(InBuffer));
BusType = BusTypeUnknown;
// As InBuffer is set to 0, it requests the STORAGE_DEVICE_DESCRIPTOR object
if (DeviceIoControl(hDevice, IOCTL_STORAGE_QUERY_PROPERTY, InBuffer, 0xCu, &OutBuffer, 0x28u, &BytesReturned, 0)) {
    BusType = OutBuffer.BusType;
}
CloseHandle(hDevice);
return BusType == BusTypeUsb;
```

# Sending a disinfection payload

**How to execute this payload?**

> 0x1002: Create a new listening thread
> 0x300e: Expand environment var (%TEMP%)
> 0x3007: CreateFile
> 0x10003008: WriteFile
> 0x10003009: CloseFile
> 0x300c: CreateProcess

**Six HTTP responses** suffices in allowing workstation and flash drive disinfection.

# But…

# Limitations, yeah, big limitations.

**No persistence**

- The infected flash drive has to be plugged in

**Very intrusive**

- Modifies the directory tree
- Removes staged data

# Lessons learned.

This case presented a **fun technical challenge**, yet its propagation vector renders it **nearly unstoppable**.

The $7 method we have used to obtain the IP address has been **successfully** applied in other cases.

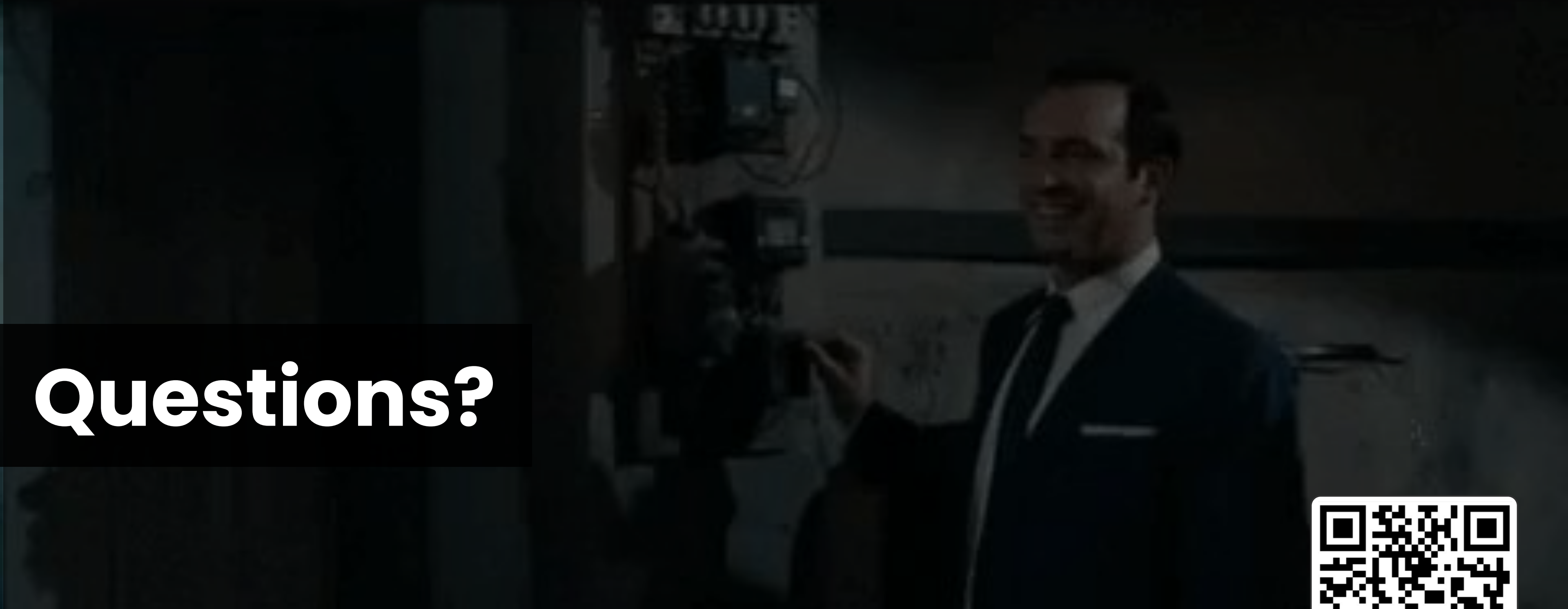
As this example illustrates, the potential for **botnet reuse must always be considered**, especially in the case of worms.

# Unanswered questions (yet).

Does this worm have **one or multiple patient zeros**? In how many countries?

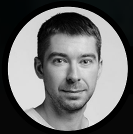
What was its real **intended purpose**?

What’s **the real status of the other three C2**? We saw that one of them has an **InetSim** :)



# Questions?

Félix Aimé



Charles Meslay

Blogpost