Pupy RAT hiding under WerFault's cover

labs.k7computing.com/index.php/pupy-rat-hiding-under-werfaults-cover/

By Saikumaravel January 4, 2023

We at K7 Labs recently identified an interesting technique used by threat actors to execute a Remote Admin Tool. We all know that **WerFault.exe** is used for the Windows Error Reporting. This blog describes how threat actors use the legitimate WerFault.exe to execute Pupy RAT on the victims' machine.

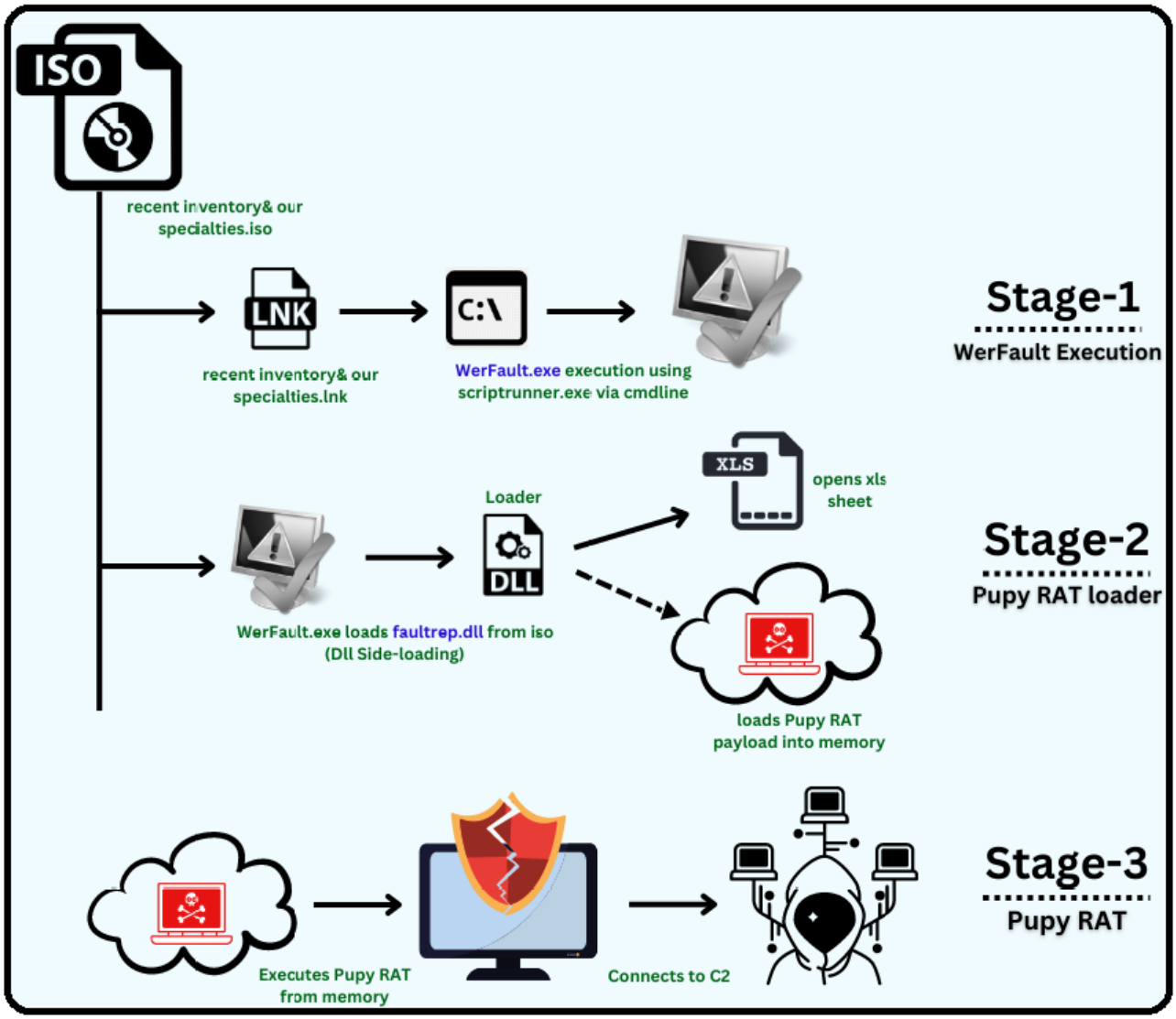


Figure 1: Execution flow

Analysis of Binary

Stage 1 – WerFault Execution

Recently we came across an ISO image, **recent inventory & our specialties.iso** from a twitter feed. The ISOcontains four files, a legitimate WerFault.exe,a malicious DLL named faultrep.dll, a shortcut file named recent inventory & our specialties.lnkand a XLS file named File.xls. The shortcut file has the same name as the ISO image. When the victim opens that shortcut file, it uses scriptrunner.exe LOLBin via cmd to execute WerFault.exe from the ISO.

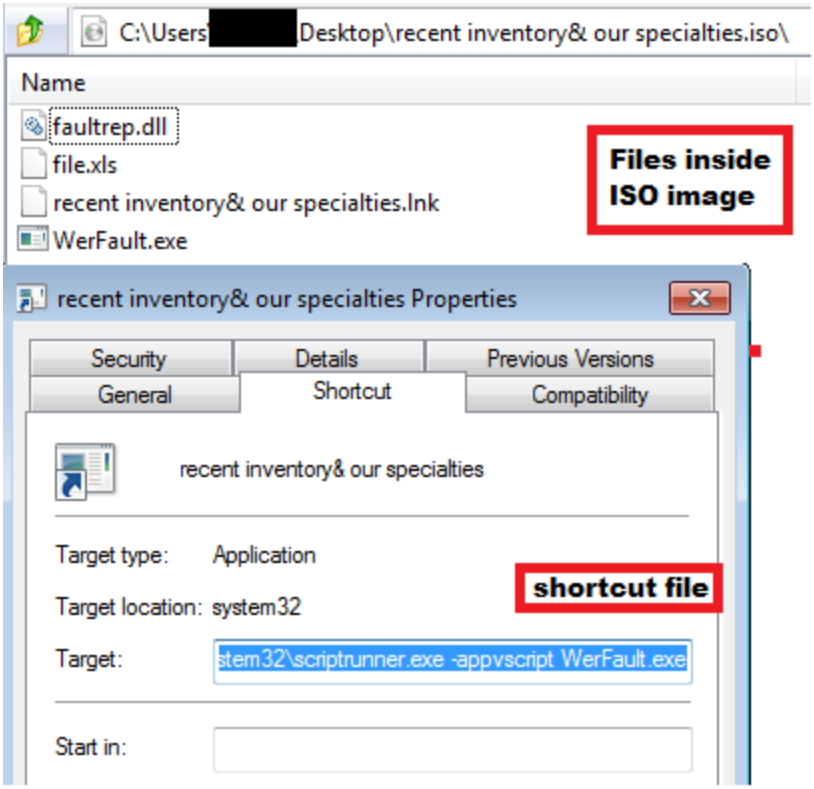


Figure 2: ISO & shortcut file

Stage 2 – Pupy RAT loader

Originally, Faultrep.dll is the name of DLL used by WerFault.exe is, which is present in the default windows folder. When WerFault.exe starts executing, it uses DLL Side-Loading technique to load the Faultrep.dll from the ISO and it has a dummy export function **WerpInitiateCrashReporting** similar to the original DLL.This malicious Faultrep.dll is compiled in C.

The DLL has a custom API resolving function with two arguments, DLL hash and Function hash.

```
.text:0000000180001B30 push    rdi
.text:0000000180001B31 mov     edx, 5A153F58h
.text:0000000180001B36 mov     ecx, 7040EE75h
.text:0000000180001B3B push    rsi
.text:0000000180001B3C push    rbx
.text:0000000180001B3D sub     rsp, 30h
.text:0000000180001B41 mov     [rsp+48h+var_24], 0B848h
.text:0000000180001B4A mov     [rsp+48h+var_1C], 0E0FF0000h
.text:0000000180001B52 call    mw_resolve_api
.text:0000000180001B57 mov     edx, 844FF18Dh
.text:0000000180001B5C mov     ecx, 7040EE75h
.text:0000000180001B61 mov     rbx, rax
```

DLL and Function Hash

Figure 3: API Resolving

We noticed that this loader uses the same API resolving function as Guloader. The DLLs resolved were kernel32 and advapi32.

After resolving the APIs, it starts to serve its purpose. Using the resolved function **CreateThread,** it creates two threads. The first thread opens a lure excel sheet named **file.xls** from the ISO.

```
.text:0000000180001AE8 mov     [rsp+48h+var_16+0Dh], 0
.text:0000000180001AED lea     rdx, [rsp+48h+var_16]
.text:0000000180001AF2 xor     r9d, r9d
.text:0000000180001AF5 mov     rcx, 'slx.elif'
.text:0000000180001AFF mov     qword ptr [rsp+48h+var_16+5], rcx
.text:0000000180001B04 lea     r8, [rsp+48h+var_16+5]
.text:0000000180001B09 xor     ecx, ecx
.text:0000000180001B0B mov     dword ptr [rsp+48h+var_16], 'nepo'
.text:0000000180001B13 mov     [rsp+48h+var_16+4], 0
.text:0000000180001B18 mov     [rsp+48h+var_20], 1
.text:0000000180001B20 mov     [rsp+48h+var_28], 0
.text:0000000180001B29 call    rax
```

Thread 1: Function to open file.xls

file.xls [Compatibility Mode] - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW

A1 | Recent inventory & Our Specialties

Recent inventory & Our Specialties

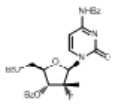
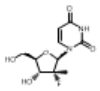
产品编号	产品结构	CAS号	中/英文名称	可提供包装（可接受定制）g/kg	纯度%（可接受定
R000001		817204-32-3	(2R,3R,4R,5R)-5-(4-benzamido-2-oxopyrimidin-1(2H)-yl)-2-((benzoyloxy)methyl)-4-fluoro-4-methyltetrahydrofuran-3-yl benzoate	100 g; 500 g; 1 kg; 5 kg; 10 kg;10kg+	>98%
R000002		863329-66-2	2'-deoxy-2'-fluoro-2'-C -Methyluridine; 1-((2R,3R,4R,5R)-3-fluoro-4-hydroxy-5-(hydroxymethyl)-3-methyltetrahydrofuran-2-yl)pyrimidine-2,4(1H,3H)-dione	100 g; 500 g; 1 kg; 5 kg; 10 kg;10kg+	>99%

Figure 4: First thread opening Excel sheet

While manually resolving the function, we found that one of the functions it resolved was **SystemFunction032** from the advapi32.dll. This function is undocumented in MSDN and on further searching we found the documentation on WineAPI. With that documentation, we understood that the function is used for RC4 encryption and accepts two arguments: keyand *data.* On further analysis we found the RC4 decryption function which contains the data and hard coded string as key.

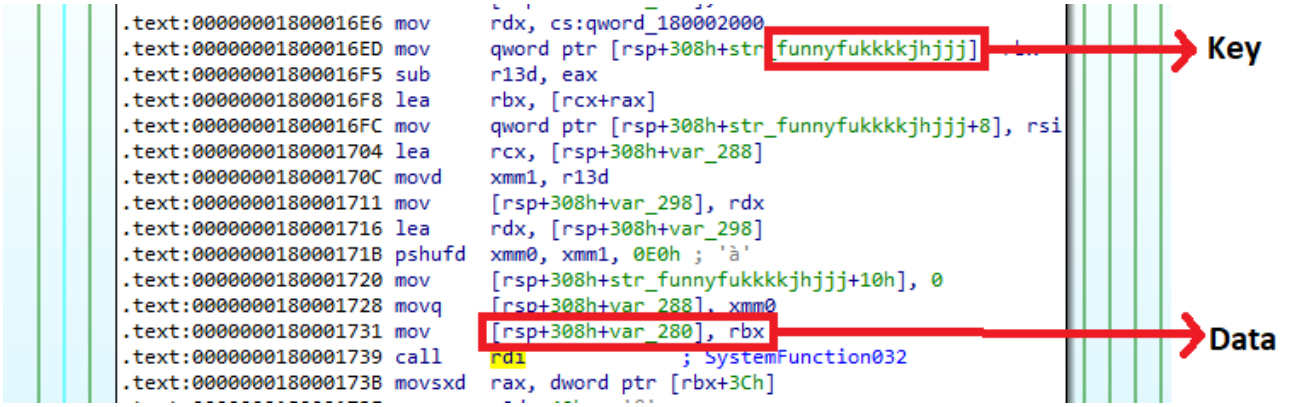


Figure 5: Second thread doing RC4 decryption

The data is pointed to the address of the overlay. So we dumped the encrypted overlay data and using the key we further decrypted it. After decrypting the data, we confirmed that the data is a PE file with the magic bytes.

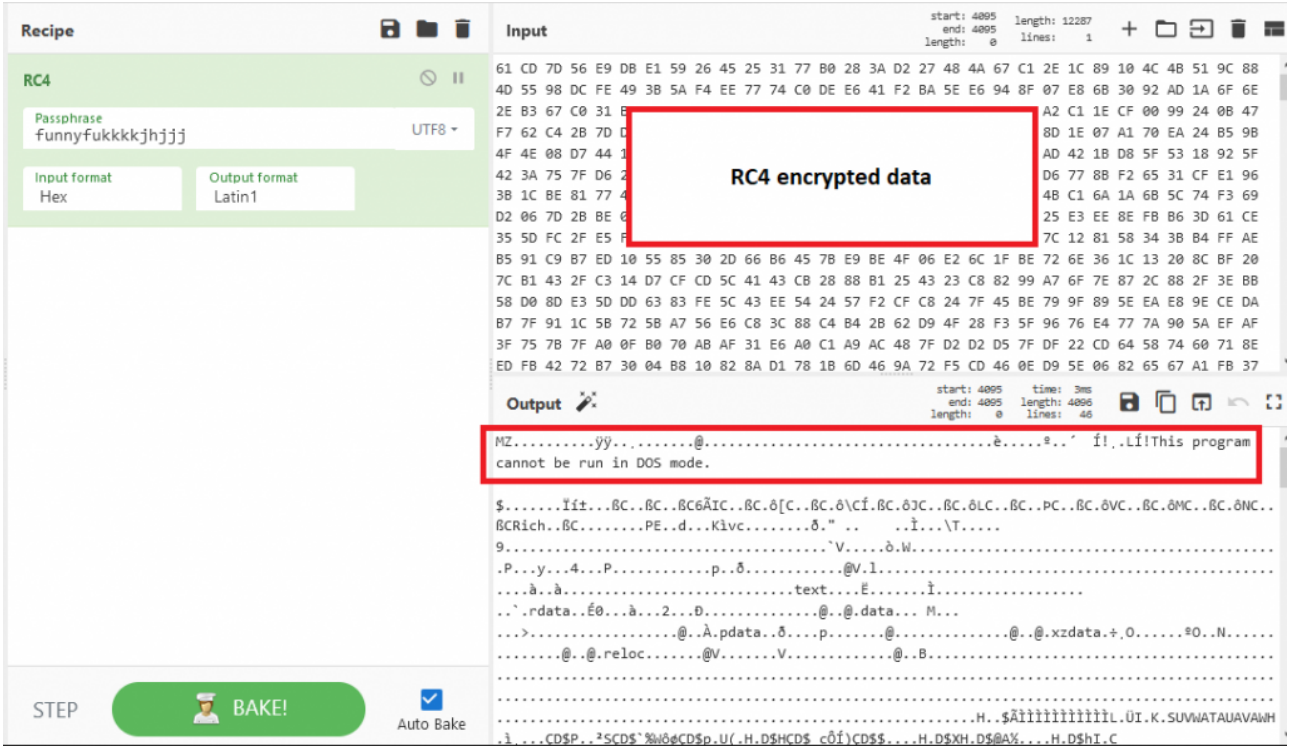


Figure 6: RC4 Decryption

We dumped the decrypted output data to a PE file. It was compiled with C & Python and found that it is a **Pupy RAT.** This RAT is loaded into the memory and executed while WerFault.exe was executing in the front.

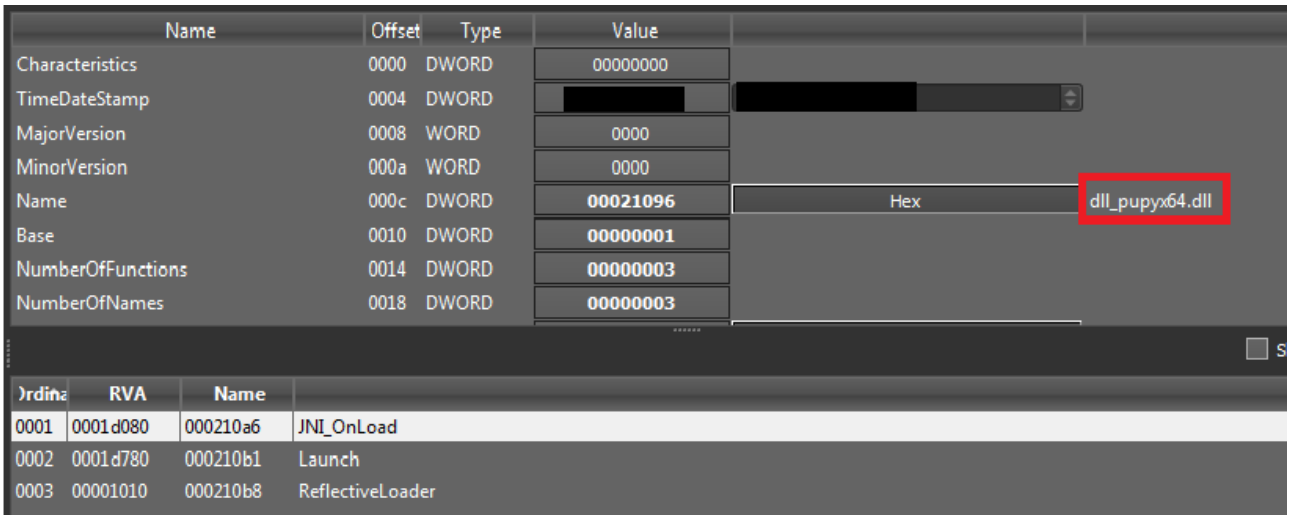


Figure 7: Decrypted PE file

Stage 3 – Pupy RAT

Pupy RAT is an open-source cross platform Remote Admin Tool available in Github
According to the sources, since 2013 it has possibly been used by APT33 and APT35 from Iran for cyber espionage operations like the one that was discovered in 2020 and targeted a major European energy organisation.

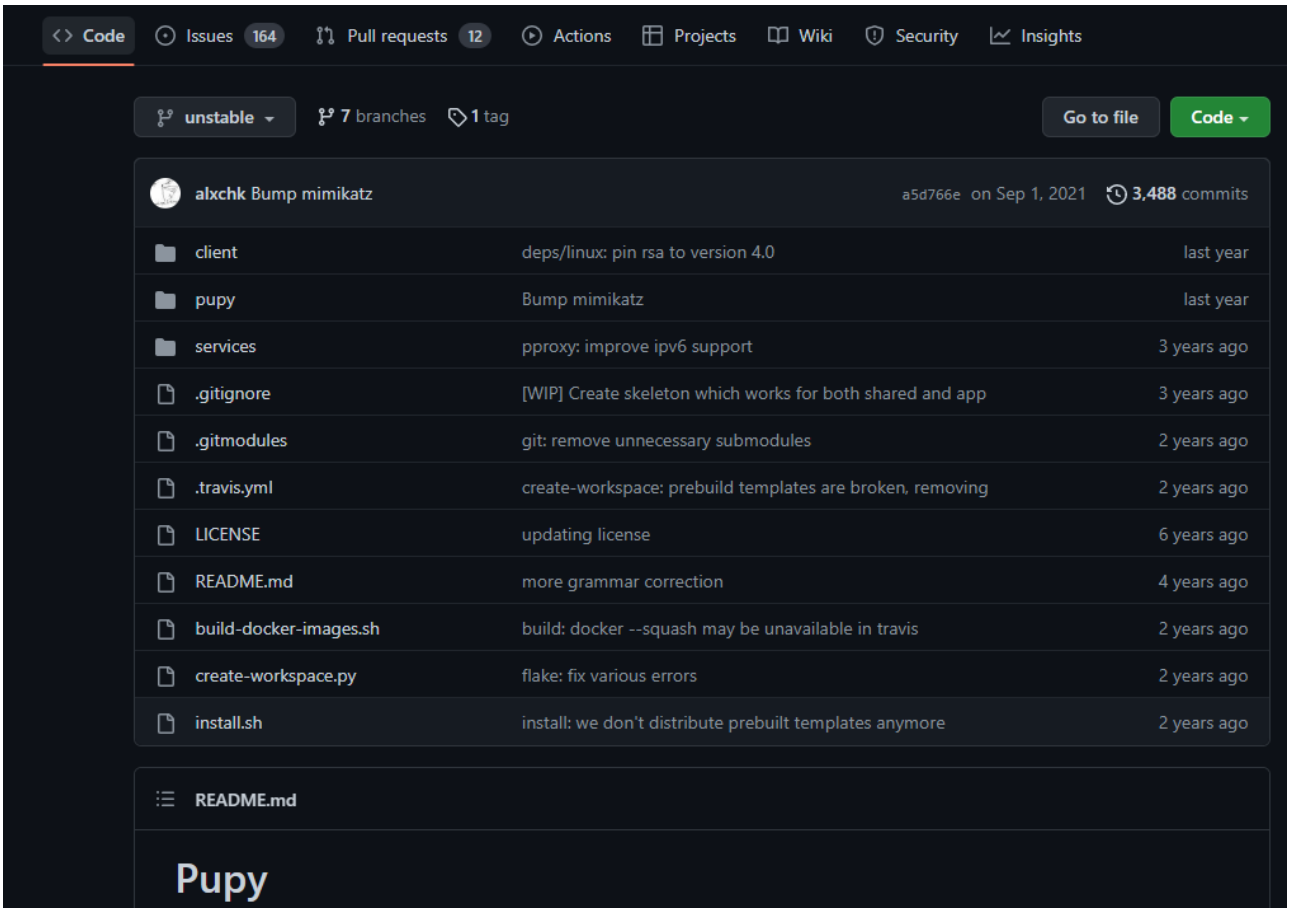


Figure 8: Pupy RAT Github

It was executed from memory and based on the analysis of ReflectiveLoader function, is capable of executing any PE file in-memory, remotely. It tries to make a C2 connection in the background when the victim believes WerFault is running. Since the C2 was down at the time of analysis, RAT was unable to establish a connection for carrying out any further malicious activity. With the XLS sheet in Chinese, we believe that the victim is from China.

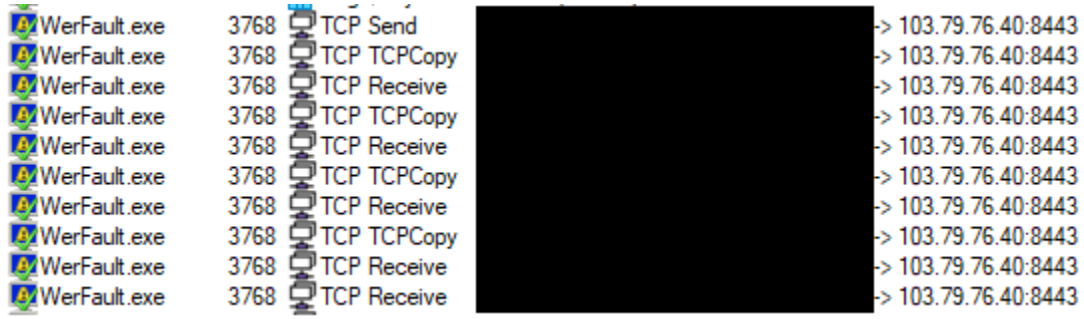


Figure 9: Pupy RAT C2 connection

We at K7 Labs provide detection against latest threats and also for this newer variant of Loader. Users are advised to use a reliable security product such as “K7 Total Security” and keep it up-to-date so as to safeguard their devices.

IoCs

Filename	Hash	K7 Detection Name
Stage 1 – WerFault Execution recent inventory & our specialties.iso	D069812AA63B631897498621DE353519	Trojan (0059ce2b1)
Stage 2 – Pupy RAT loader faultrep.dll	42A5798608F196CE7376CE196F4452FE	Trojan (0059ce2b1)
Stage 3 – Pupy RAT Decrypted PupyRAT	F365A8BDFD9B39C4F8B9D99613818207	Trojan (0001140e1)

C2

hxxp[://103[.79[.76[.40/

References

https://twitter.com/SBousseaden/status/1603425101528956935