

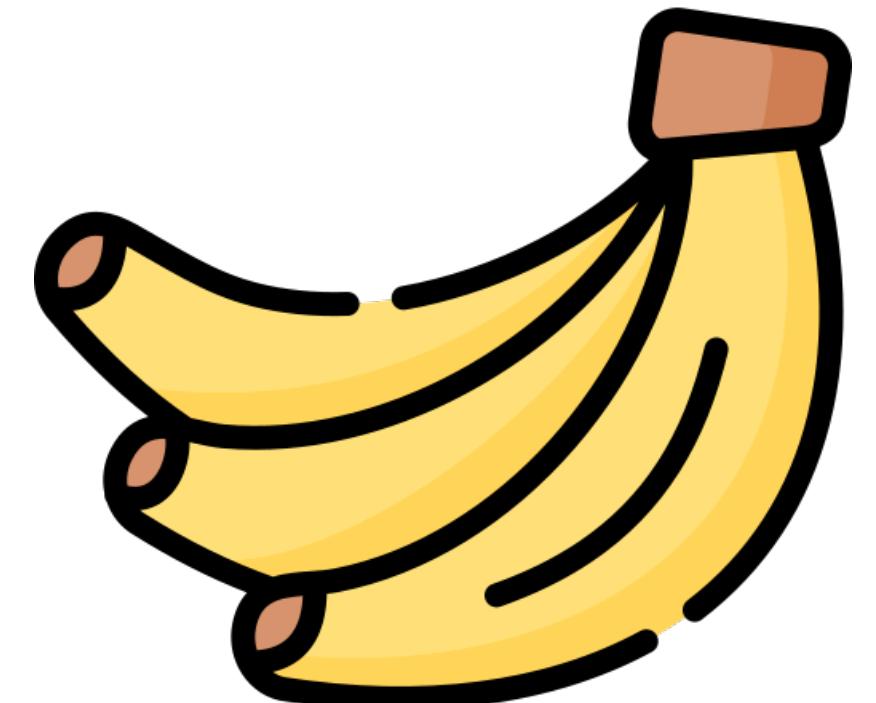
Launch and Environment Constraints ~~Overview~~ Deep Dive



OffSec™

Csaba Fitzl

Twitter: @theevilbit



whoami

🍌 lead content developer of
"EXP-312: Advanced macOS Control
Bypasses"
@ OffSec

🍌 ex red/blue teamer

🍌 macOS bug hunter

🍌 husband, father

🍌 hiking, trail running 🏔️ 🏔️



Europe's largest banana producer is:



agenda

1. old macOS vulnerabilities
2. Launch Constraints (Ventura)
3. old and current third party vulnerabilities
4. Launch and Environment constraints (Sonoma)

old macOS vulnerabilities

TCC bypass with imagent.app

TCC bypass with imagent.app

🍌 Found by Adam Chester (@_xpn_)

🍌 imagent.app with TCC and keychain related entitlements

🍌 loads plugins from:

🍌 imagent.app/Contents/PlugIns

🍌 code signing allows 3rd party plugins

🍌 copy app to /tmp/ and load your plugin

```
<key>com.apple.private.tcc.allow.overridable</key>
<array>
    <string>kTCCServiceAddressBook</string>
</array>

<key>keychain-access-groups</key>
<array>
    <string>iChat</string>
    <string>apple</string>
    <string>appleaccount</string>
    <string>InternetAccounts</string>
    <string>IMCore</string>
</array>
```

*TCC bypass using Directory
Utility.app, CVE-2020-27937*

TCC bypass using Directory Utility.app, CVE-2020-27937

🍌 found by Wojciech Regula (@_r3ggi)

🍌 Directory Utility with admin rights to change user properties, like HOME

🍌 allows plugins including non Apple

🍌 copy app to /tmp/ and load our plugin

🍌 change HOME -> new TCC.db -> our rules

```
<key>com.apple.private.tcc.allow</key>
<array>
    <string>kTCCServiceSystemPolicySysAdminFiles</string>
</array>
```

*TCC bypass using configd,
"powerdir"*

TCC bypass using configd, "powerdir"

🍌 Found by Jonathan Bar Or
(@yo_yo_yo_jbo)

🍌 configd has again user update rights (can change HOME)

🍌 -b allows loading an bundle (including non Apple)

🍌 normally launched by launchd but we could start it via command line as well

```
[Key] com.apple.private.tcc.allow
[Value]
    [Array]
        [String] kTCCServiceSystemPolicySysAdminFiles
```

Introducing Launch Constraints

Launch Constraints

🍌 introduced in macOS Ventura (13)

🍌 mitigates many logic vulnerabilities

🍌 defines 3 constraints:

🍌 Self Constraints

🍌 Parent Constraints

🍌 Responsible Constraints

LC in Action

```
csaby@max /tmp % cp -r /System/Applications/FindMy.app .

csaby@max /tmp % open FindMy.app
The application cannot be opened for an unexpected reason, error=Error Domain=RBSRequestErrorDomain
Code=5 "Launch failed." UserInfo={NSLocalizedFailureReason=Launch failed.,
NSUnderlyingError=0x6000000032d0 {Error Domain=NSPOSIXErrorDomain Code=162 "Unknown error: 162"
UserInfo={NSLocalizedDescription=Launchd job spawn failed}}}
```

```
csaby@max /tmp % log stream | grep AMFI
2023-09-19 14:18:21.273482+0200 0x2e3486  Default      0x0          0      0  kernel:
(AppleMobileFileIntegrity) AMFI: Launch Constraint Violation (enforcing), error info: c[1]p[1]m[1]e[2],
(Constraint not matched) launching proc[vc: 1 pid: 52468]: /private/tmp/FindMy.app/Contents/MacOS/
FindMy, launch type 0, failure proc [vc: 1 pid: 52468]: /private/tmp/FindMy.app/Contents/MacOS/FindMy
```

Launch Constraints Categories

LC Categories

🍌 category = defines a set of launch constraints

🍌 Ventura - 7 categories - documented by Linus Henze

🍌 Sonoma - 18 categories - documented by Csaba Fitzl

🍌 assigns each binary in the trust cache to a category

LC Category examples

Category 1:

```
Self Constraint: (on-authorized-authapfs-volume || on-system-volume) && launch-type == 1 &&  
validation-category == 1
```

Parent Constraint: is-init-proc

🍌 *on-authorized-authapfs-volume || on-system-volume* - System or Cryptex

🍌 *launch-type == 1* - system service

🍌 *validation-category == 1* - must present in the
trust cache

🍌 *is-init-proc* - launchd

```
/usr/libexec/routined  
/usr/libexec/nehelper  
/usr/libexec/remoted  
/usr/libexec/seld  
/usr/libexec/logd  
/usr/libexec/thermalmonitord
```

LC Category examples

Category 2:

Self Constraint: on-authorized-authapfs-volume || on-system-volume

🍌 *on-authorized-authapfs-volume || on-system-volume* - System or Cryptex

🍌 less restrictive

/usr/bin/brctl
/usr/bin/bputil
/usr/bin/bison
/usr/bin/bioutil
/usr/bin/binhex
/usr/bin/bc
/usr/bin/batch

trust cache

Trust Cache

🍌 A few places:

🍌 /System/Library/Security/OSLaunchPolicyData

🍌 /System/Volumes/Preboot/[uuid]/boot/[long hex]/usr/standalone/firmware/
FUD/BaseSystemTrustCache.img4

🍌 /System/Volumes/Preboot/[uuid]/boot/[long hex]/usr/standalone/firmware/
FUD/StaticTrustCache.img4

🍌 IMG4 and IM4P (P = Payload) files

Trust Cache

🍌IMG4 - extract IM4P

```
pyimg4 img4 extract -i BaseSystemTrustCache.img4 -p BaseSystemTrustCache.im4p  
pyimg4 img4 extract -i StaticTrustCache.img4 -p StaticTrustCache.im4p
```

🍌IM4P - extract data

🍌pyimg4 Python utility

```
csaby@max /tmp % pyimg4 im4p info -i /System/Library/Security/OSLaunchPolicyData  
Reading /System/Library/Security/OSLaunchPolicyData...  
Image4 payload info:  
    FourCC: ltrs  
    Description: 1  
    Data size: 329.14KB  
    Encrypted: False
```

```
csaby@max /tmp % pyimg4 im4p extract -i /System/Library/Security/OSLaunchPolicyData -o  
OSLaunchPolicyData.data  
Reading /System/Library/Security/OSLaunchPolicyData...  
Extracted Image4 payload data to: OSLaunchPolicyData.data
```

Trust Cache

👉 TC v2 can contain constraint category

👉 trustcache utility to analyze it

👉 category: 4th column

```
struct trust_cache_entry2 {  
    uint8_t cdhash[CS_CDHASH_LEN];  
    uint8_t hash_type;  
    uint8_t flags;  
    uint8_t constraintCategory;  
    uint8_t reserved0;  
} __attribute__((__packed__));
```

```
csaby@max /tmp % trustcache_macos_arm64 info OSLaunchPolicyData.data | head  
version = 2  
uuid = CCC03EBE-7949-460E-A335-14C6396FC927  
entry count = 13713  
000600f05b768de957b57afbd576ad031b7dc984 [none] [2] [2]  
0008df4d1e0d4b276a82f3e086bea3e93670cf94 [none] [2] [2]  
000c95ce2e4f99248a33e5c7f690452f89afc16b [none] [2] [0]  
0019e93d101896746f77a3b7047d2d8281352fc5 [none] [2] [3]  
0020f949545b505610f55f962520bfb4f1851f1d [none] [2] [2]  
002655b46255b6fb2f5b2a4987345cc0e46836f8 [none] [2] [2]  
002aa16545d098699c706c27a08e834243f5b984 [none] [2] [2]
```

reversing launch constraints

reversing LC

🍌 defined in AMFI
(AppleMobileFileIntegrity)

🍌 download KDK to get the KEXT

🍌 symbols with *kConstraintCategory** prefix

🍌 extract symbols from KEXT

```
kConstraintCategory1_Self  
kConstraintCategory2_Self  
kConstraintCategory3_Self  
kConstraintCategory4_Self  
kConstraintCategory5_Self  
kConstraintCategory6_Self  
kConstraintCategory7_Self  
kConstraintCategory8_Self  
kConstraintCategory9_Self  
kConstraintCategory10_Self  
kConstraintCategory12_Self  
kConstraintCategory13_Self  
kConstraintCategory14_Self  
kConstraintCategory15_Self  
kConstraintCategory16_Self  
kConstraintCategory17_Self  
kConstraintCategory18_Self  
kConstraintCategory20_Self  
kConstraintCategory1_Parent  
kConstraintCategory4_Parent  
kConstraintCategory5_Parent  
kConstraintCategory6_Parent  
kConstraintCategory8_Parent  
kConstraintCategory14_Parent  
kConstraintCategory15_Parent  
kConstraintCategory16_Parent  
kConstraintCategory18_Parent
```


reversing LC

🍌 ASN.1 DER encoded data (serialized)

🍌 many tools which can decode it, like python-asn1

🍌 DER encoding:

🍌 Type

🍌 Length

🍌 Value

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101  
  
[A] SEQUENCE  
[U] INTEGER: 1  
  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

7075020101B07030420C03246F72B03B
30220C1D6F6E2D617574686F72697A65
642D61757468617066732D766F6C756D
650101FF30150C106F6E2D7379737465
6D2D766F6C756D650101FF30100C0B6C
61756E63682D7479706502010130180C
1376616C69646174696F6E2D63617465
676F7279020101

[A] SEQUENCE
[U] INTEGER: 1

[C] SEQUENCE
[U] SEQUENCE
[U] UTF8STRING: \$or

[C] SEQUENCE
[U] SEQUENCE
[U] UTF8STRING: on-authorized-authapfs-volume

[U] BOOLEAN: True

[U] SEQUENCE
[U] UTF8STRING: on-system-volume

[U] BOOLEAN: True

[U] SEQUENCE
[U] UTF8STRING: launch-type

[U] INTEGER: 1

[U] SEQUENCE
[U] UTF8STRING: validation-category

[U] INTEGER: 1

DER decoding example

7075020101B07030420C03246F72B03B
30220C1D6F6E2D617574686F72697A65
642D61757468617066732D766F6C756D
650101FF30150C106F6E2D7379737465
6D2D766F6C756D650101FF30100C0B6C
61756E63682D7479706502010130180C
1376616C69646174696F6E2D63617465
676F7279020101

[A] SEQUENCE
[U] INTEGER: 1
[C] SEQUENCE
[U] SEQUENCE
[U] UTF8STRING: \$or
[C] SEQUENCE
[U] SEQUENCE
[U] UTF8STRING: on-authorized-authapfs-volume
[U] BOOLEAN: True
[U] SEQUENCE
[U] UTF8STRING: on-system-volume
[U] BOOLEAN: True
[U] SEQUENCE
[U] UTF8STRING: launch-type
[U] INTEGER: 1
[U] SEQUENCE
[U] UTF8STRING: validation-category
[U] INTEGER: 1

DER decoding example

7075020101B070**3042**0C03246F72B03B
30220C1D6F6E2D617574686F72697A65
642D61757468617066732D766F6C756D
650101FF30150C106F6E2D7379737465
6D2D766F6C756D650101FF30100C0B6C
61756E63682D7479706502010130180C
1376616C69646174696F6E2D63617465
676F7279020101

[A] SEQUENCE
[U] INTEGER: 1
[C] SEQUENCE
[U] SEQUENCE
[U] UTF8STRING: \$or
[C] SEQUENCE
[U] SEQUENCE
[U] UTF8STRING: on-authorized-authapfs-volume
[U] BOOLEAN: True
[U] SEQUENCE
[U] UTF8STRING: on-system-volume
[U] BOOLEAN: True
[U] SEQUENCE
[U] UTF8STRING: launch-type
[U] INTEGER: 1
[U] SEQUENCE
[U] UTF8STRING: validation-category
[U] INTEGER: 1

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

7075020101B07030420C03246F72**B03B**
30220C1D6F6E2D617574686F72697A65
642D61757468617066732D766F6C756D
650101FF30150C106F6E2D7379737465
6D2D766F6C756D650101FF30100C0B6C
61756E63682D7479706502010130180C
1376616C69646174696F6E2D63617465
676F7279020101

[A] SEQUENCE
[U] INTEGER: 1
[C] SEQUENCE
[U] SEQUENCE
[U] UTF8STRING: \$or
[C] SEQUENCE
[U] SEQUENCE
[U] UTF8STRING: on-authorized-authapfs-volume
[U] BOOLEAN: True
[U] SEQUENCE
[U] UTF8STRING: on-system-volume
[U] BOOLEAN: True
[U] SEQUENCE
[U] UTF8STRING: launch-type
[U] INTEGER: 1
[U] SEQUENCE
[U] UTF8STRING: validation-category
[U] INTEGER: 1

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101  
  
[A] SEQUENCE  
[U] INTEGER: 1  
  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
  
[U] SEQUENCE  
  
[U] UTF8STRING: on-system-volume  
  
[U] BOOLEAN: True  
  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101  
  
[A] SEQUENCE  
[U] INTEGER: 1  
  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101  
  
[A] SEQUENCE  
[U] INTEGER: 1  
  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

DER decoding example

```
7075020101B07030420C03246F72B03B  
30220C1D6F6E2D617574686F72697A65  
642D61757468617066732D766F6C756D  
650101FF30150C106F6E2D7379737465  
6D2D766F6C756D650101FF30100C0B6C  
61756E63682D7479706502010130180C  
1376616C69646174696F6E2D63617465  
676F7279020101
```

```
[A] SEQUENCE  
[U] INTEGER: 1  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: $or  
[C] SEQUENCE  
[U] SEQUENCE  
[U] UTF8STRING: on-authorized-authapfs-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: on-system-volume  
[U] BOOLEAN: True  
[U] SEQUENCE  
[U] UTF8STRING: launch-type  
[U] INTEGER: 1  
[U] SEQUENCE  
[U] UTF8STRING: validation-category  
[U] INTEGER: 1
```

```
(on-authorized-authapfs-volume || on-system-volume) && launch-type == 1 && validation-category == 1
```

attack mitigation

LC attack mitigation

↳ imagent.app & Directory Utility.app

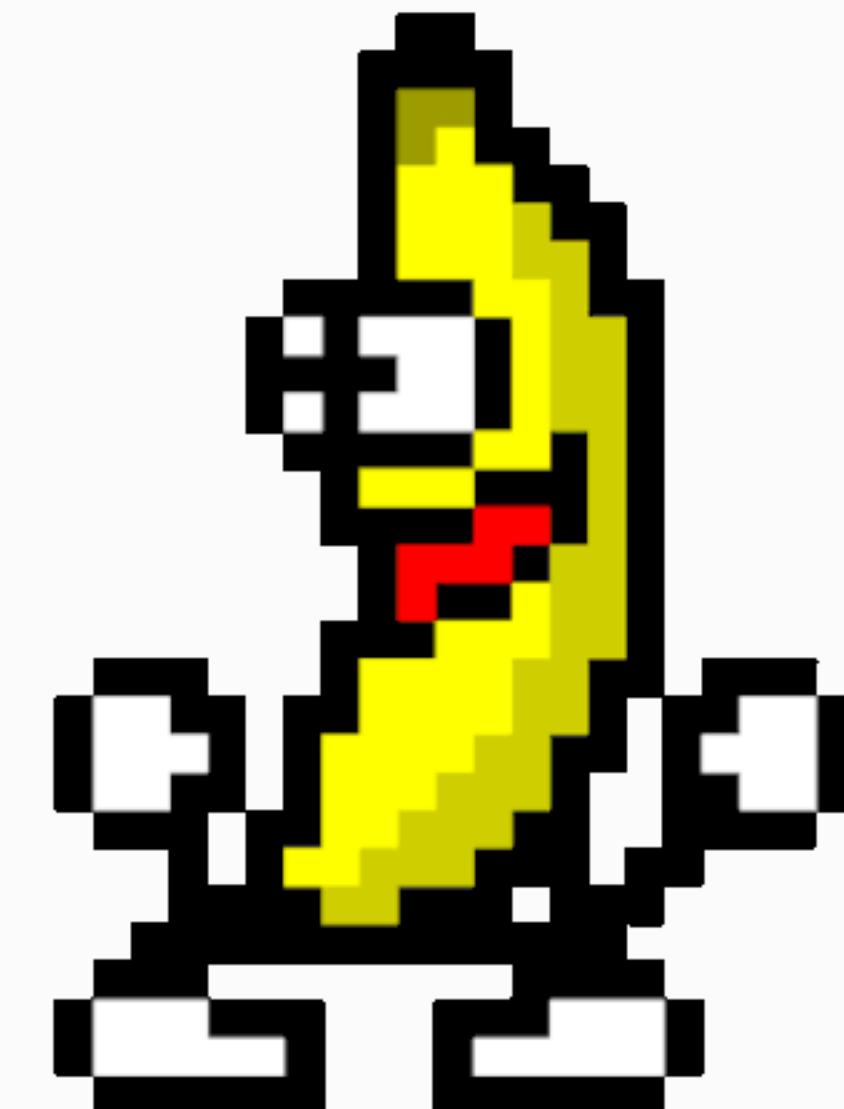
↳ (on-authorized-authapfs-volume || on-system-volume)

↳ wouldn't be able to start a copy

↳ configd

↳ Parent Constraint: is-init-proc + system service

↳ wouldn't be able to start from command line



typical third party attacks

XPC attacks

- 🍌 many blog post series from Wojciech Regula, Csaba Fitzl and others
- 🍌 XPC services which run as root are dangerous
- 🍌 XPC service must ensure that only the real client can connect
- 🍌 client validation is ~~hard~~ not straightforward
 - 🍌 must use audit token
 - 🍌 client must be signed by developer certificate issued by Apple
 - 🍌 client's version must be checked and / or
 - 🍌 client shouldn't possess dangerous entitlements (e.g.: com.apple.security.cs.disable-library-validation)

Electron framework attacks

- 🍌 Electron based apps are popular (Chrome, Signal, Slack, Discord, MS Teams, ...)
- 🍌 often have TCC permissions (Camera, Microphone, ...)
- 🍌 subject to injection attacks (local attack are not in their threat model)
- 🍌 ELECTRON_RUN_AS_NODE - allows node.js cli interaction
 - 🍌 can be disabled
- 🍌 arguments: --inspect and --remote-debugging-port - allow debugger attachment
 - 🍌 --remote-debugging-port can't be disabled

dylib injection

🍌 if "com.apple.security.cs.disable-library-validation" present

🍌 often used if third party plugins must be supported

🍌 opens up the attack surface for XPC and TCC bypasses

🍌 -> can inject code into the clients

Launch and Environment Constraints (for 3rd parties)

"In this talk I will talk about two mitigations which Apple introduced in order to protect against many types of logic vulnerabilities. Launch Constraints was introduced in macOS Ventura, and they can control who can launch a built-in system application and how. Environment Constraints were introduced in Sonoma, and it's basically the extension of Launch Constraints for third party apps. These two features are probably the most impactful when it comes to exploitation. I will review them in detail, how they are set up, what they do exactly, and what kind of vulnerability classes they mitigate. I will also go through a couple of past vulnerabilities, which could not have been exploited with these constraints present. Finally I will walk through how various third party apps should be set up in order to be secure."

-Csaba Fitzl

"In this talk I will talk about two mitigations which Apple introduced in order to protect against many types of logic vulnerabilities. Launch Constraints was introduced in macOS Ventura, and they can control who can launch a built-in system application and how. Environment Constraints were introduced in Sonoma, and it's basically the extension of Launch Constraints for third party apps. These two features are probably the most impactful when it comes to exploitation. I will review them in detail, how they are set up, what they do exactly, and what kind of vulnerability classes they mitigate. I will also go through a couple of past vulnerabilities, which could not have been exploited with these constraints present. Finally I will walk through how various third party apps should be set up in order to be secure."

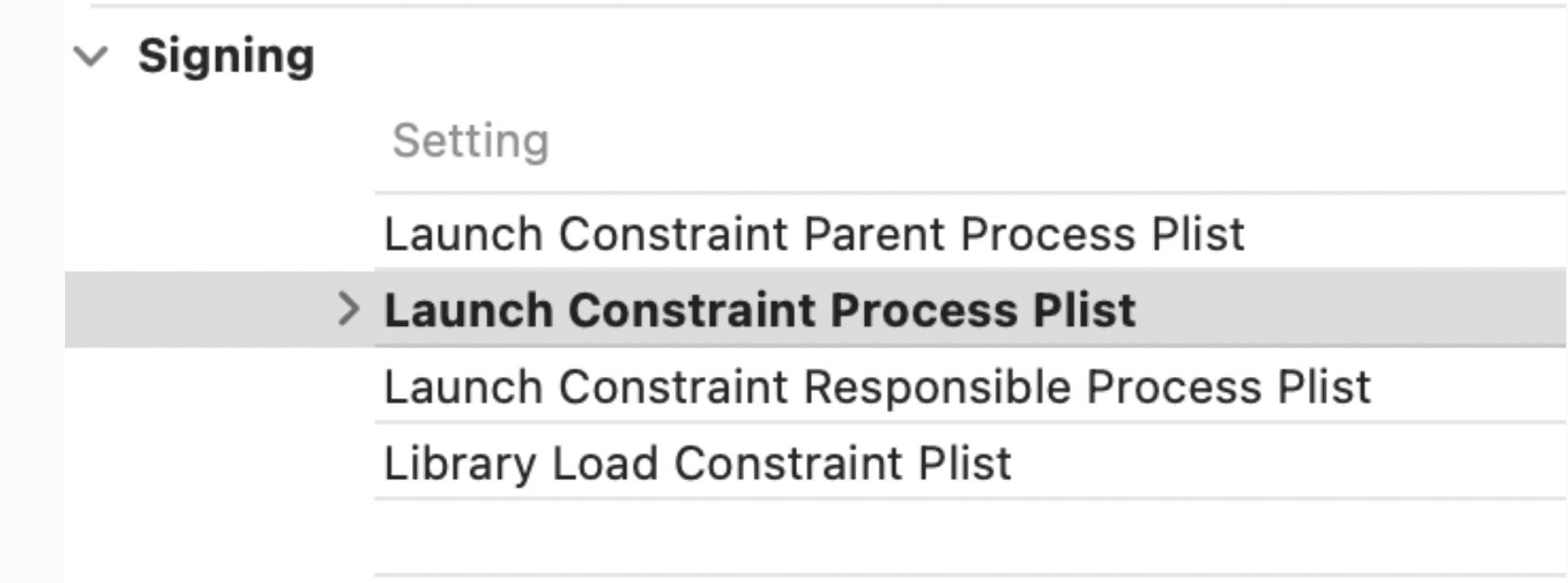
-Csaba Fitzl

"In this talk I will talk about two mitigations which Apple introduced in order to protect against many types of logic vulnerabilities. Launch Constraints was introduced in macOS Ventura, and they can control who can launch a built-in system application and how. Environment Constraints were introduced in Sonoma, and it's basically the extension of Launch Constraints for third party apps. These two features are probably the most impactful when it comes to exploitation. I will review them in detail, how they are set up, what they do exactly, and what kind of vulnerability classes they mitigate. I will also go through a couple of past vulnerabilities, which could not have been exploited with these constraints present. Finally I will walk through how various third party apps should be set up in order to be secure. rant about how Apple failed big time with LC for third parties"

-Csaba Fitzl

intro

- 🍌 now LC available for 3rd party apps
- 🍌 can define all 3 constraints (self, parent, responsible)
- 🍌 additionally +1 library load constraint
- 🍌 defined in code requirement in Xcode
- 🍌 well documented in Apple Developer Documentation



XPC "protection"

🍌 sounds like LC makes XPC secure

🍌 but!!!

🍌 but!!!

🍌 if the service is already running then
what? - LC is not in play

🍌 +bug: responsible process: XPC
service itself, not the client 🤦

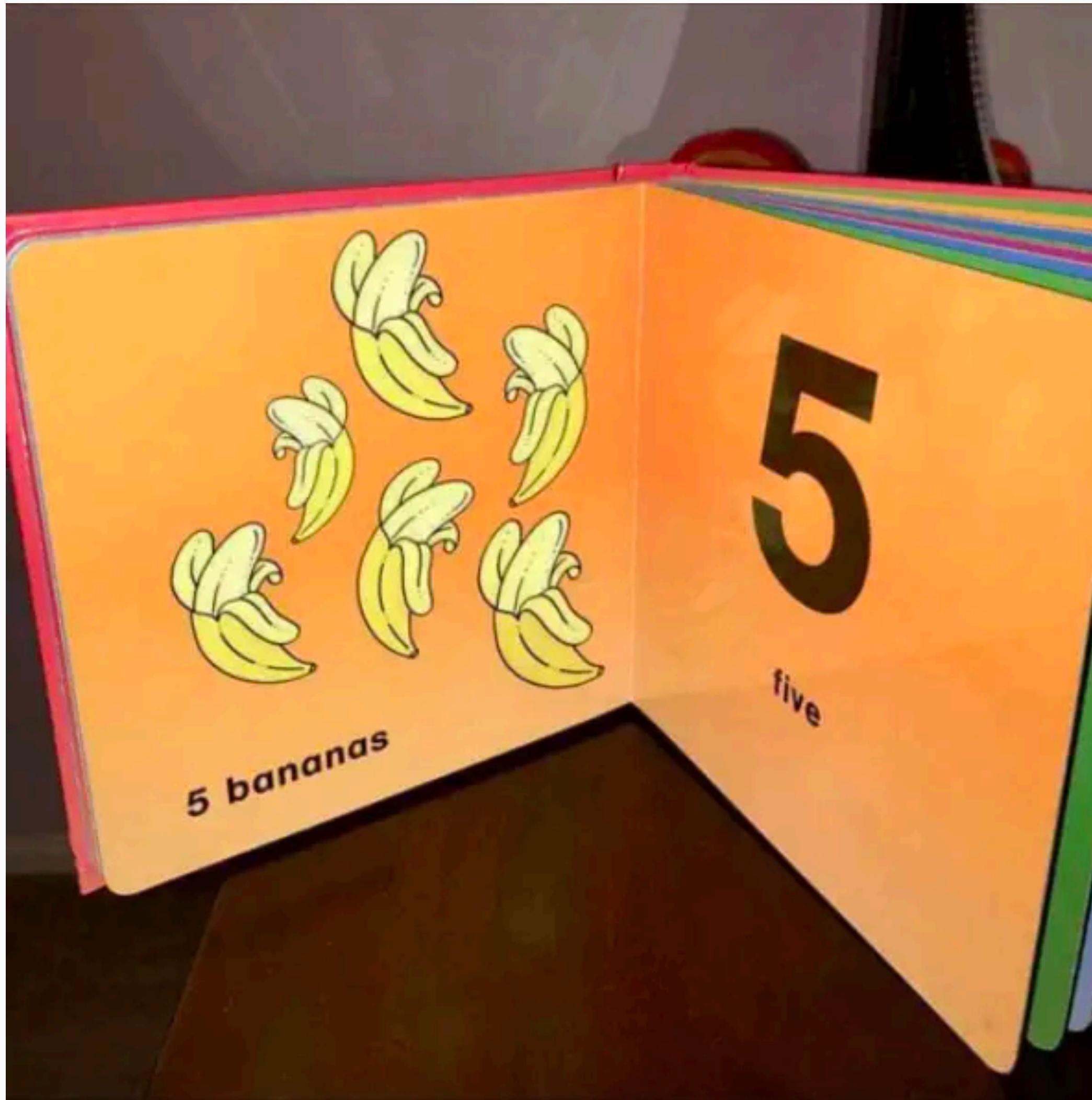
(FB13206884)

not exactly :-)

"Now let's walk through some process relationships and talk about how you can use launch constraints to secure them. First assume that MyDemo.app is your app. You can set a self constraint on my MyDemo.app to require that it launch as an application from Launch Services. When your app requests a connection to your XPC service, launchd spawns the XPC service and is the parent of that XPC service but your app is "responsible" for that XPC service. You could set a responsible process constraint on MyXPCDemo.xpc to indicate that only MyDemo.app should be responsible for it." *

* WWDC2023: Protect your Mac app with environment constraints

YOU HAD ONE JOB!



"Securing" Electron applications

🍌 posix_spawn is dangerous

🍌 launch type = 3 = launch as application

🍌 but!!!

🍌 but!!!

🍌 open command launches as application

🍌 can pass env vars and arguments

"Just like in real parent-child relationships, parent processes have a huge amount of influence over how a child behaves. On macOS, the power to posix_spawn another process gives the parent the ability to control nearly all input to the child. The parent process can also limit the child's access to system resources. This level of control can cause the child to load unexpected code, to run unexpected features, or to behave in ways that make the process more vulnerable to attack." *

* WWDC2023: Protect your Mac app with environment constraints

"Securing" Electron applications

or just use the API

```
NSWorkspaceOpenConfiguration *conf = [NSWorkspaceOpenConfiguration configuration];
conf.environment = @{
    @"ELECTRON_RUN_AS_NODE": @"1"
};
[[NSWorkspace sharedWorkspace] openURL:[NSURL fileURLWithPath:@"/Applications/Electron.app"]
    configuration:conf
    completionHandler:nil];
```



WeKnowMemes

my other issues

🍌 Apple's WWDC example: command line helper tool

🍌 anyone uses that anymore?

🍌 usual recommendation is XPC for the past (10?) years

🍌 doesn't feel real-world

YOU HAD ONE JOB!



library load constraints

🍌 if you used "com.apple.security.cs.disable-library-validation"

🍌 solves third party plugin support issues

🍌 problem:

🍌 assumes you are aware of all the plugins

conclusion

conclusion

- ✖ LC for Apple binaries
 - ✖ great improvement
 - ✖ mitigates many common attacks
- ✖ LC for 3rd parties
 - ✖ there is potential, but needs improvement
 - ✖ currently doesn't impact the most common attacks
 - ✖ library load constraints can be useful
 - ✖ start to use it



Csaba Fitzl
Twitter: @theevilbit

Resources

- [flaticon.com](#) - Freepik, [rsetiawan](#)