macOS Red Teaming on Corporate Scenarios

SPEAKER:

Ricardo L0gan

Agenda

About Me:

Security Specialist with extensive experience in enterprise networks and enthusiastic on malware research, pentest and reverse engineering. I have been focused in the last years in research for vulnerability and malware for macOS environment.

I am part of the staff for some security conferences organizations such as H2HC (Hackers to Hackers Conference), BsidesSP and SlackShow/Slackzine Community.

Brazil

Agenda

0x00 Motivation of Research
0x01 macOS Security (Default / Corp)
0x02 Hacking macOS target
0x03 POC Extra (Bypass TCC)
0x04 Native macOS Tools
0x05 Conclusion
0x06 Reference

0x00 Motivation of Research

Empire Transfer 2024

RustDoor 2024

MetaStealer 2024

Lockbit 2023

CriptoMiner 2023

Silver Sparrow 2021

OSX/Linker 2019

Mac Auto Fixer 2018

LoundMiner 2019

Crossrider (OSX/Shalyer) 2018

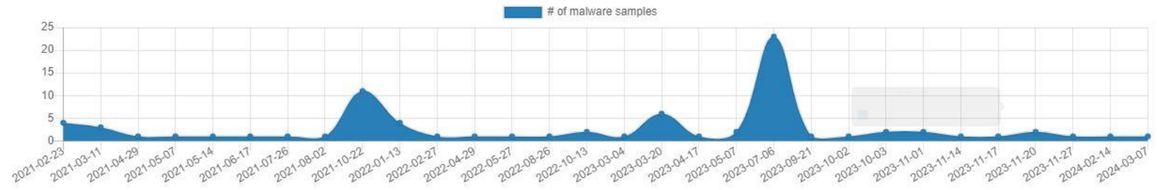
OSX/MaMi 2018

Others….

0x00 Motivation of Research

Database Entry

Tag:	macOS Alert
Firstseen:	2020-07-01 06:42:26 UTC
Lastseen:	2024-03-07 17:11:54 UTC
Sightings:	90



Fonte: https://bazaar.abuse.ch/browse/tag/macos

Malware Samples

The table below shows all malware samples that are associated with this particulare tag (**max 400**).

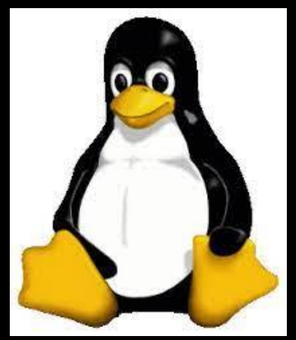
Show 50 entries Search:

Firstseen (UTC)	SHA256 hash	Tags	Signature
2024-03-07 17:11:55	c802c94d0836039aa986e...	atomic stealer AtomicStealer macho macos stealer	AtomicStealer
2024-02-14 19:27:12	08ff8a6500d623b062dce...	dmg macOS ViaCrackSite	n/a
2023-11-27 14:21:56	0db57feffa1f92816f9477f...	AMOS AmosStealer macho macOS OSX	AmosStealer
2023-11-20 13:43:25	a7bb346838db73301fe1...	mac Mach-O macho macOS Meterpreter OSX	n/a
2023-11-20 13:34:13	a6fd2f09eb81bf3afc83c4...	mac Mach-O macho macOS Meterpreter OSX	n/a
2023-11-17 07:12:52	a40d65307e67af3f18246...	AMOS AmosStealer ClearFake dmg macOS	AmosStealer
2023-11-14 18:26:31	60792845a1086b5c2ad7...	Adware dmg macOS	n/a
2023-11-01 17:28:39	3ea2ead8f3cec030906dc...	macho macOS	n/a
2023-11-01 17:24:07	2360a69e5fd7217e97712...	HLoader macho macOS	n/a
2023-10-03 11:26:19	b86e245d71f7a1056a7c5...	AMOS AmosStealer AtomicSteal macOS	n/a
2023-10-03 11:26:02	05ee833f167c4afa9e746...	AMOS AmosStealer AtomicSteal macOS	n/a
2023-10-02 10:21:33	135fc266af08a28fc7b2d3...	Mach-O macho macOS Meterpreter	Meterpreter
2023-09-21 19:50:52	6b0bde56810f7c0295d5...	AMOS Atomic macOS stealer xz	n/a
2023-07-06 15:19:57	016a1a4fe3e9d57ab0b2a...	macho macOS RealstStealer	n/a
2023-07-06 15:19:07	4b93ec3fd49c0111e8a11...	macOS pkg RealstStealer	n/a
2023-07-06 15:17:30	2c321b1416fb7226bffd1...	macho macOS RealstStealer	n/a

Fonte: https://bazaar.abuse.ch/browse/tag/macos

0x00 Motivation of Research

Is one of them safer?



Windows

macOS®

?

0x01 macOS Security (Default / Corp)

Version	Release Date	
Cheetah	2001	
Puma	2001	
Jaguar	2002	
Panther	2003	
Tiger	2005	
Leopard	2007	
Snow Leopard	2009	The first version of macOS I started using
Lion	2011	
Mountain Lion	2012	
Mavericks	2013	
Yosemite	2014	
El Capitan	2015	
Sierra	2016	
High Sierra	2017	
Mojave	2018	
Catalina	2019	
Big Sur	2020	ARM/Intel
Monterey	2021	(Supported) + ARM/Intel
Ventura	2022	(Supported) + ARM/Intel
Sonoma	2023	(Supported) + ARM/Intel
Sequoia	2024	Is coming...

SIP

FileVault

Xprotect

Secure Boot

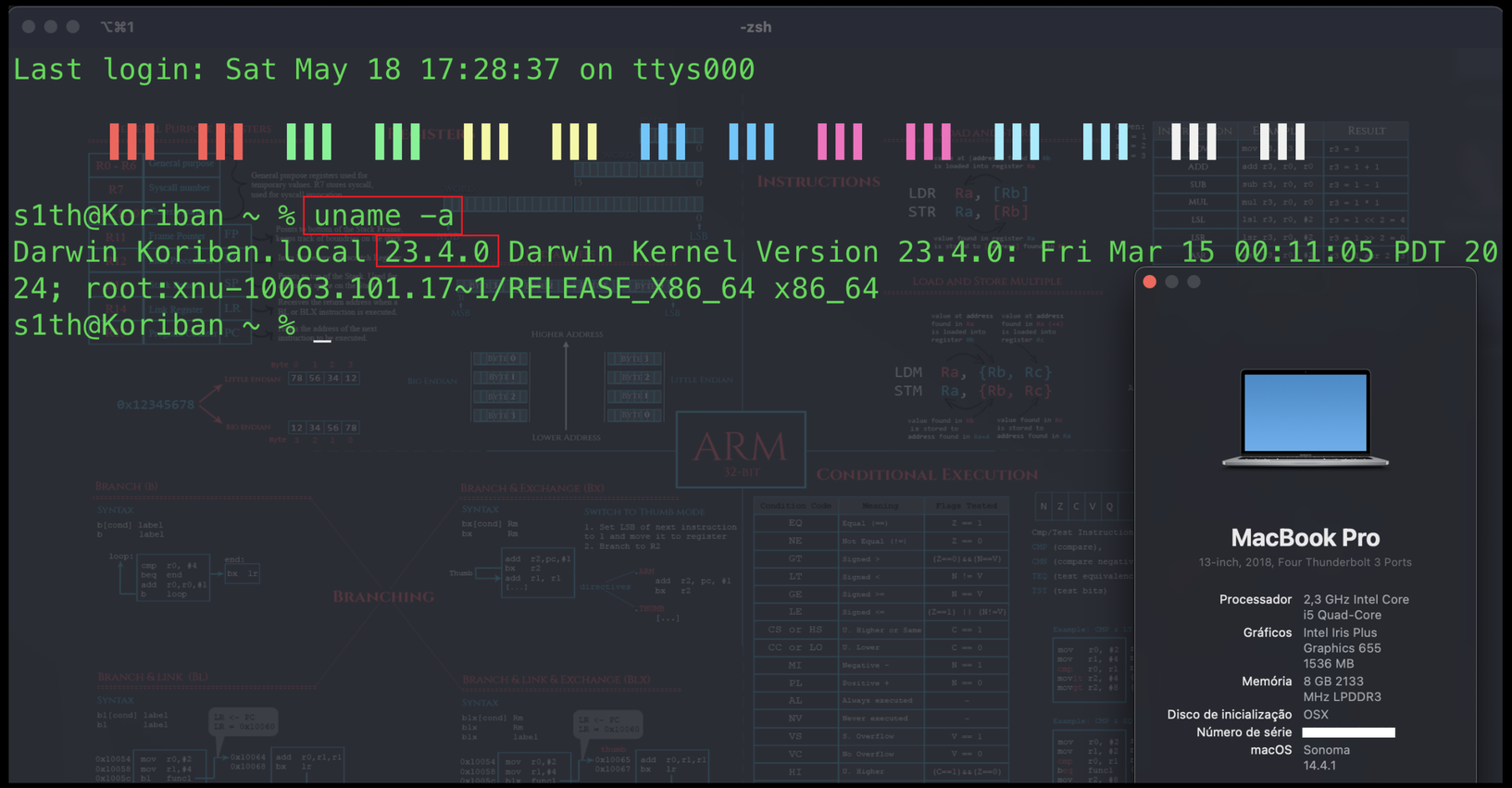
Gatekeeper

TCC

SSV

0x01 macOS Security (Default / Corp)

- A modern Operating system (macOS fanBoy LOL) Unix based;
- Kernel XNU is based on micro-kernel of NeXTSTEP (Mach) and kernel of BSD (FreeBSD);
- Lots of userland applications;
- macOS has grown significantly in market share;
- macOS supports both intel and ARM64(Silicon M1 M2 M3) chips.



Link XNU: https://github.com/apple/darwin-xnu

0x01 macOS Security (Default / Corp)

KEXT (Kernel Extension) files in macOS are components of the operating system's kernel that allow for the extension and enhancement of kernel functionality. These files are essential for the operation of macOS and play a critical role in the interaction between hardware and system software.

```
$ sudo kextload payload.kext
```

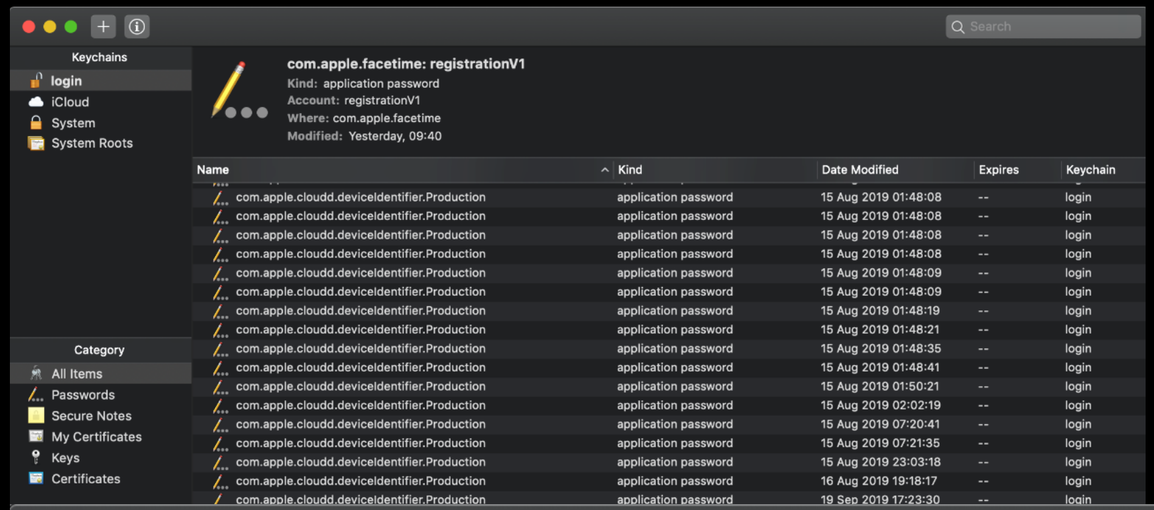
```
-zsh
s1th@Koriban Extensions % pwd
/System/Library/Extensions
s1th@Koriban Extensions %
s1th@Koriban Extensions % ls -l
total 0
drwxr-xr-x@ 3 root  wheel   96 16 Set 04:48 AFKACIPCKext.kext
drwxr-xr-x@ 3 root  wheel   96 16 Set 04:48 AFTK_Kext.kext
drwxr-xr-x@ 3 root  wheel   96 16 Set 04:48 AGXFirmwareKextG13GRTBuddy.kext
drwxr-xr-x@ 3 root  wheel   96 16 Set 04:48 AGXFirmwareKextG13XRTBuddy.kext
drwxr-xr-x@ 3 root  wheel   96 16 Set 04:48 AGXFirmwareKextG14GRTBuddy.kext
drwxr-xr-x@ 3 root  wheel   96 16 Set 04:48 AGXFirmwareKextG14PRTBuddy.kext
drwxr-xr-x@ 3 root  wheel   96 16 Set 04:48 AGXFirmwareKextG14XRTBuddy.kext
drwxr-xr-x@ 3 root  wheel   96 16 Set 04:48 AGXFirmwareKextRTBuddy64.kext
```

```
l0gan@Zion-Inf3ct3d:[~][20:58:07]
└──────> $:> sudo sqlite3 /var/db/SystemPolicyConfiguration/KextPolicy
SQLite version 3.24.0 2018-06-04 14:10:15
Enter ".help" for usage hints.
sqlite> SELECT * from kext_policy;
EG7KH642X6|com.vmware.kext.vmci|1|VMware, Inc.|1
EG7KH642X6|com.vmware.kext.vmnet|1|VMware, Inc.|1
EG7KH642X6|com.vmware.kext.vmx86|1|VMware, Inc.|1
EG7KH642X6|com.vmware.kext.vmioplug.18.1.2|1|VMware, Inc.|1
2Y8XE5CQ94|com.kaspersky.kext.klif|1|Kaspersky Lab UK Limited|1
2Y8XE5CQ94|com.kaspersky.nke|1|Kaspersky Lab UK Limited|1
sqlite> _
```

0x01 macOS Security (Default / Corp)

The Keychain in macOS is a password and security key management system that provides secure storage and protection for sensitive information, such as:

- ❑ Passwords;
- ❑ Encryption Keys;
- ❑ Certificates;
- ❑ Authentication Information.




```
l0gan@Zion-Inf3ct3d:[~][16:15:17]
└──────> $:> security list-keychains
    "/Users/l0gan/Library/Keychains/login.keychain-db"
    "/Library/Keychains/System.keychain"
l0gan@Zion-Inf3ct3d:[~][16:15:26]
```

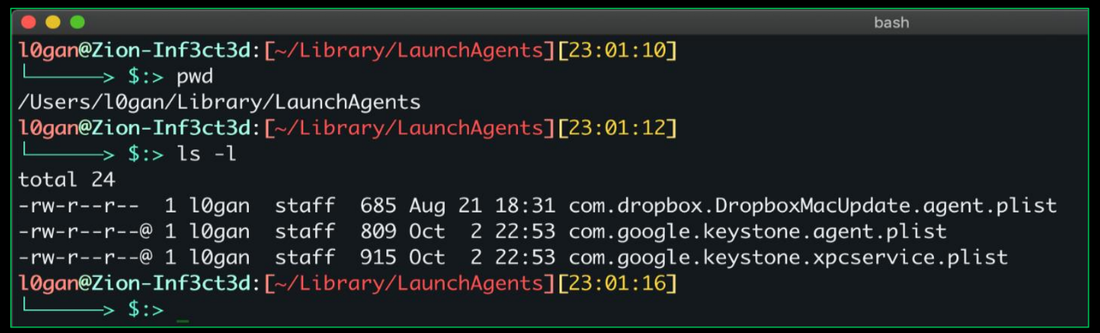
```
bash
sh ⌘1
l0gan@Zion-Inf3ct3d:[~/Library/Keychains][22:04:29]
└──────> $:> pwd
/Users/l0gan/Library/Keychains
l0gan@Zion-Inf3ct3d:[~/Library/Keychains][22:04:31]
└──────> $:> ls -l
total 976
drwx------  10 l0gan  staff     320 Oct 17 16:10 8DEA70A8-20E3-5FB9-BBC4-132BC9525660
-rw-r--r--@  1 l0gan  staff  474568 Oct 19 19:49 login.keychain-db
-rw-------   1 l0gan  staff   23804 Oct  8 09:52 metadata.keychain-db
l0gan@Zion-Inf3ct3d:[~/Library/Keychains][22:04:34]
└──────> $:>
```

0x01 macOS Security (Default / Corp)

Launchctl (LaunchDaemons / LaunchAgents)

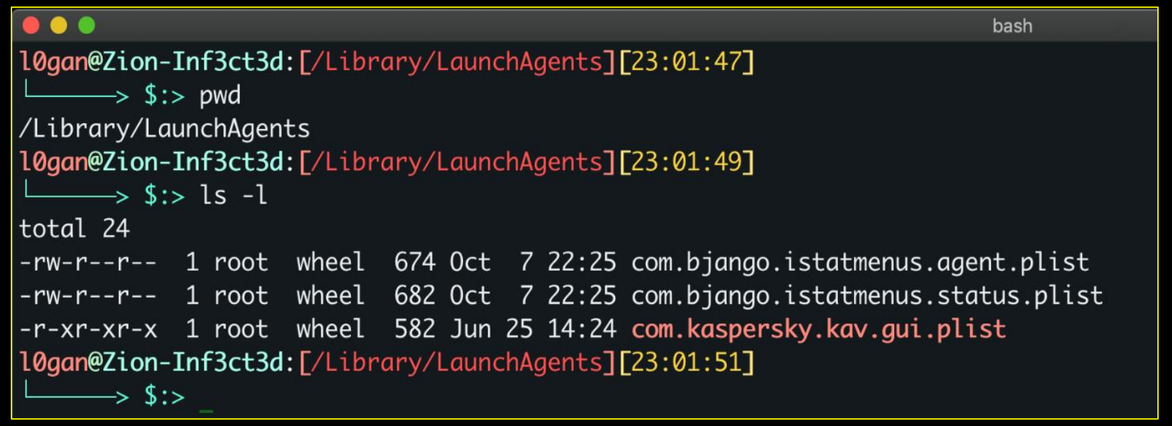
Directories LaunchDaemons and LaunchAgents are used to manage and execute automated tasks and services in the system. These directories are essential for maintaining the persistence of tasks and programs. This can be of interest to an attacker.

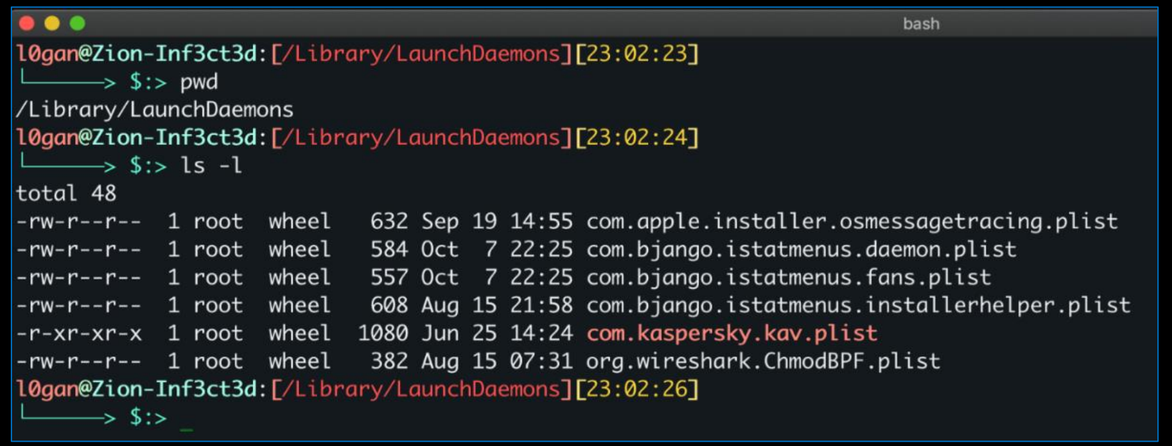
```
### Launchctl (System)
/Library/LaunchDaemons/
/Library/LaunchAgents/


### Launchctl (User)
~/Library/LaunchAgents/
```



$ launchctl load file.plist

PLIST file is a settings file, also known as a "properties file," used by macOS applications.

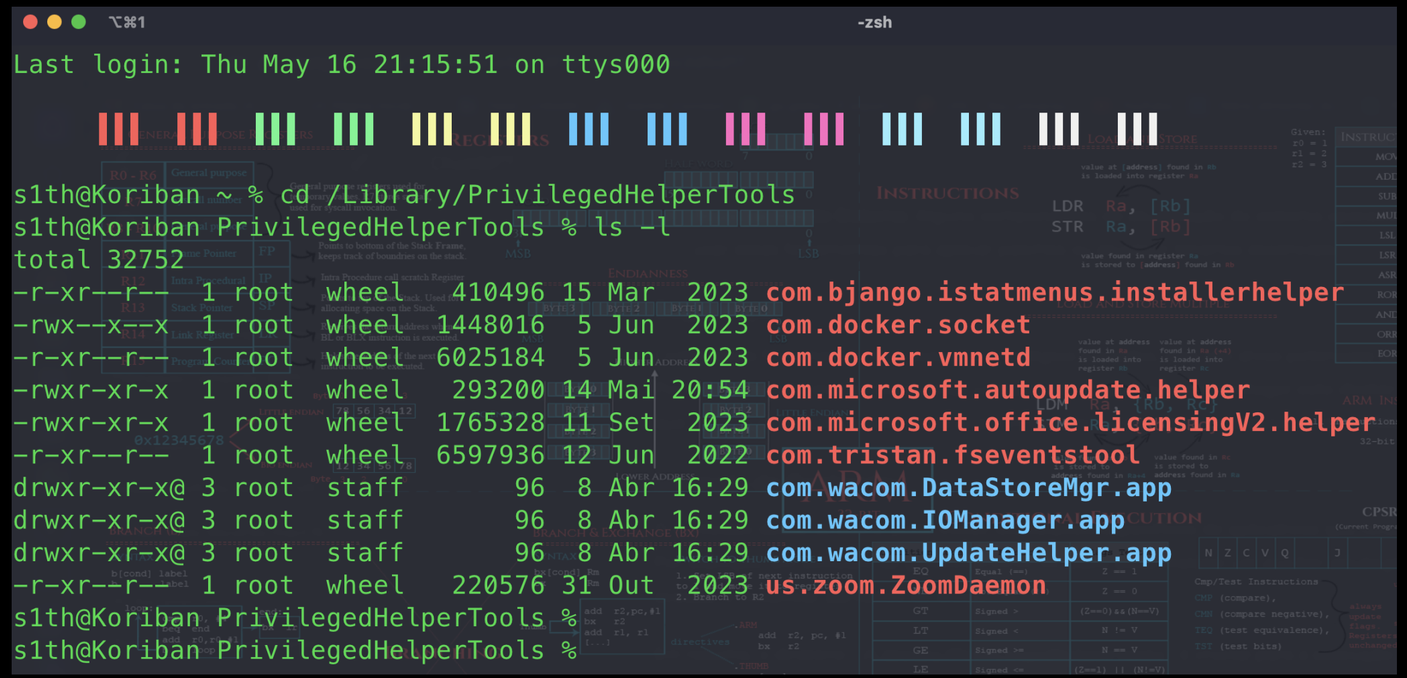




0x01 macOS Security (Default / Corp)

$ PrivilegeHelperTools (FOLDER)

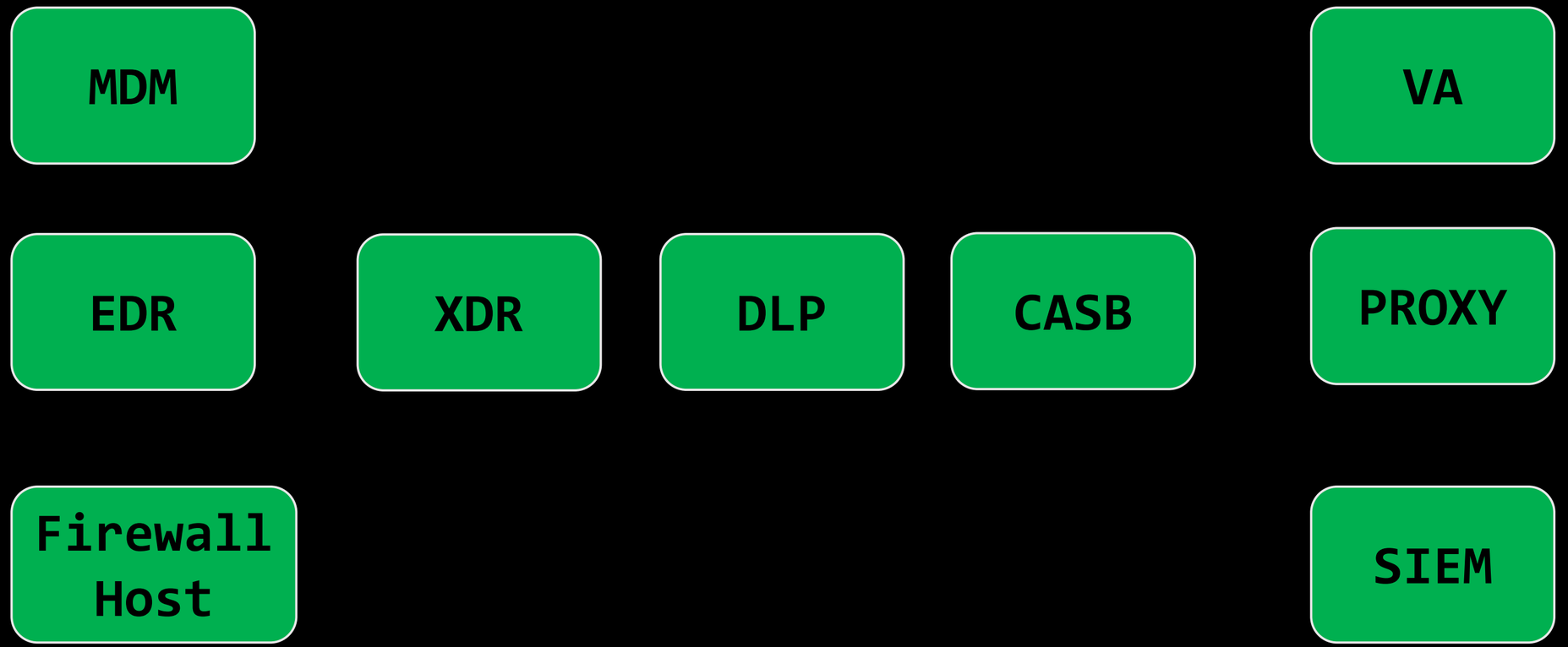
The /Library/PrivilegedHelperTools directory is crucial for executing tasks that require elevated privileges on macOS, ensuring that certain operations can be performed safely and efficiently with the necessary permissions.



Main Functions:

- Execute Tasks with Elevated Privileges
- Support for System and Application Services
- Software Installers, Security Software, and System Updates

0x01 macOS Security (Default / Corp)

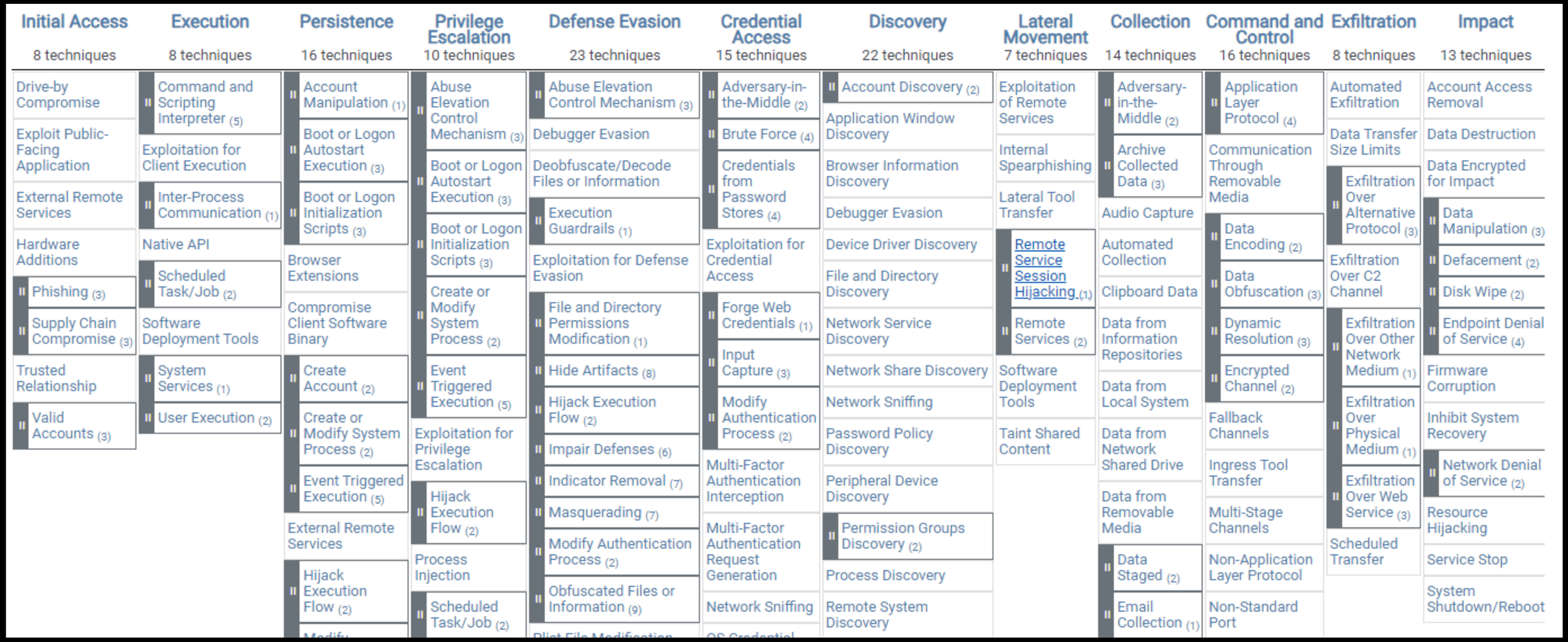


And now?

Where should I go?

0x02 Hacking macOS Target

MITRE | ATT&CK®

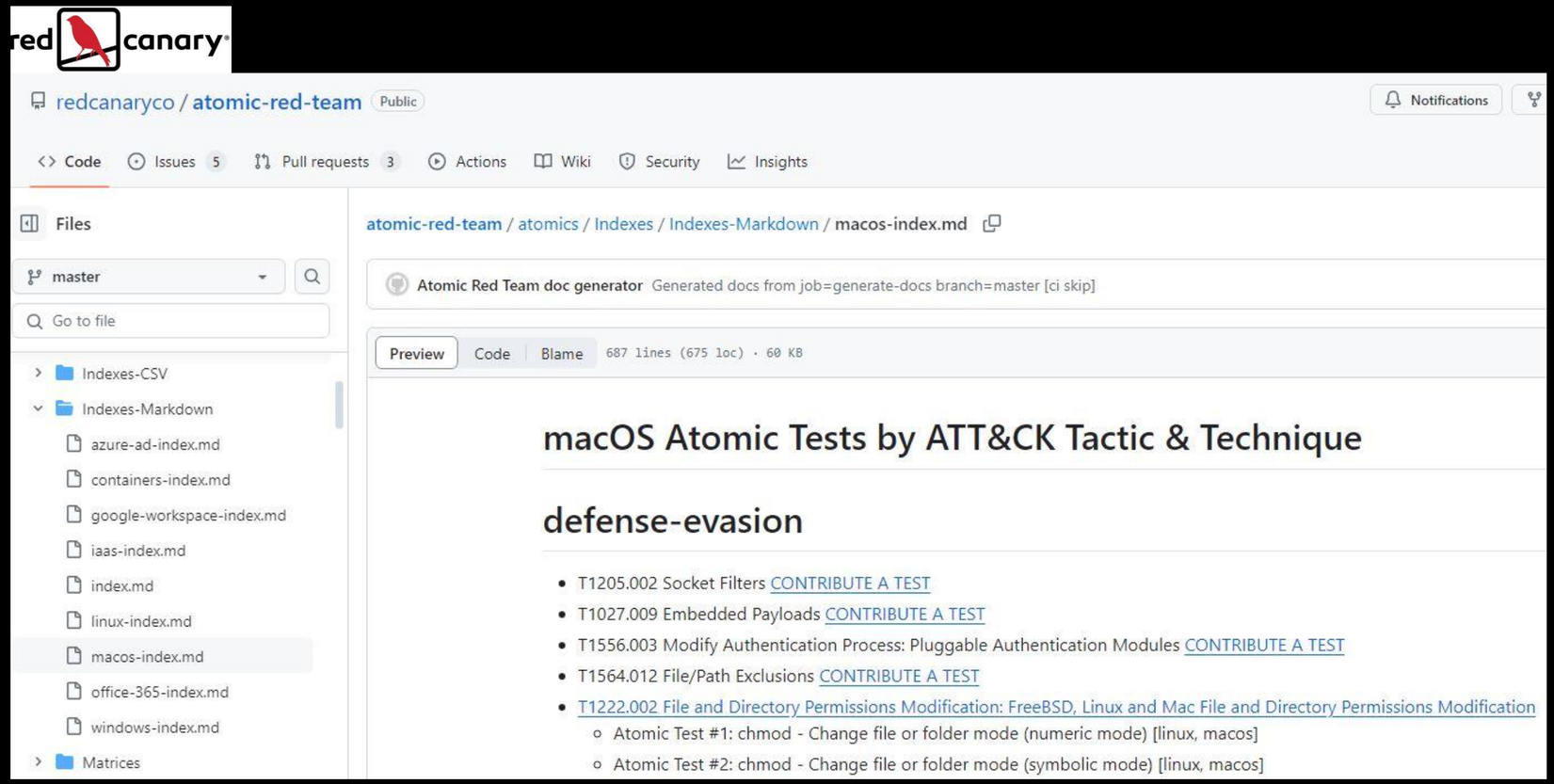


Fonte: https://attack.mitre.org/matrices/enterprise/macos/

0x02 Hacking macOS Target

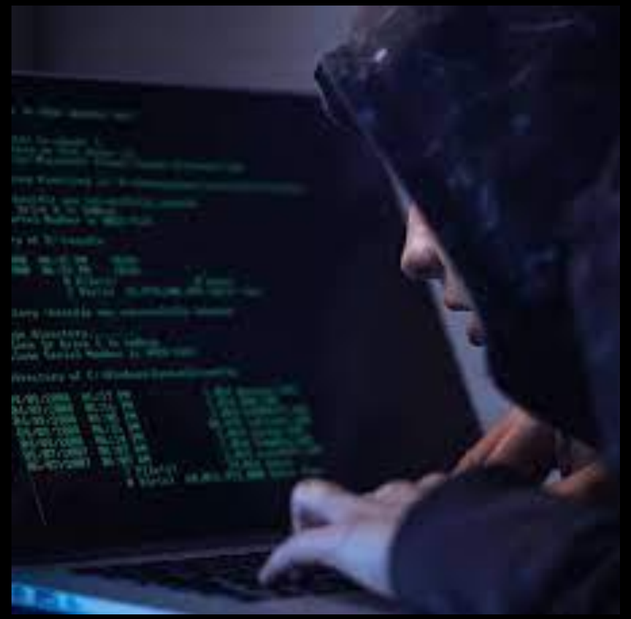
TA = Tactic Analysis

1. Initial Access (TA0001)
2. Execution (TA0002)
3. Persistence (TA0003)
4. Privilege Escalation (TA0004)
5. Defense Evasion (TA0005)
6. Credential Access (TA0006)
7. Discovery (TA0007)
8. Lateral Movement (TA0008)
9. Collection (TA0009)
10. Exfiltration (TA0010)
11. Command and Control (TA0011)



Fonte: https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/Indexes/Indexes-Markdown/macos-index.md

0x02 Hacking macOS Target



Thinking in a scenario where the company has an employee who is interested in stealing confidential information and/or compromising the environment.

DROP MALICIOUS PAYLOAD

0x02 Hacking macOS Target

```
s1th@Koriban ~ %
s1th@Koriban ~ % system_profiler SPFirewallDataType
Firewall:

    Firewall Settings:

      Mode: Block all incoming connections
      Firewall Logging: Yes
      Stealth Mode: No

s1th@Koriban ~ %
```

```
chacal@Hell ~
chacal@Hell ~ netstat -an | grep "LISTEN"
tcp4       0      0  *.631                  *.*                    LISTEN
tcp6       0      0  *.631                  *.*                    LISTEN
tcp4       0      0  *.3031                 *.*                    LISTEN
tcp6       0      0  *.3031                 *.*                    LISTEN
tcp4       0      0  *.22                   *.*                    LISTEN
tcp6       0      0  *.22                   *.*                    LISTEN
tcp4       0      0  *.88                   *.*                    LISTEN
tcp6       0      0  *.88                   *.*                    LISTEN
tcp4       0      0  *.445                  *.*                    LISTEN
tcp6       0      0  *.445                  *.*                    LISTEN
tcp6       0      0  *.65275                *.*                    LISTEN
tcp4       0      0  *.65275                *.*                    LISTEN
chacal@Hell ~
chacal@Hell ~
```

0x02 Hacking macOS Target

```
macadmin in ~
➜  system_profiler SPApplicationsDataType
Applications:

    Hopper Disassembler v4:

      Version: 4.3.11
      Obtained from: Identified Developer
      Last Modified: 28/12/17, 14:59
      Kind: Intel
      Signed by: Developer ID Application: Cryptic Apps (2AMA2753NF), Developer ID Certificat
ion Authority, Apple Root CA
      Location: /Applications/Hopper Disassembler v4.app

    iMovie:

      Version: 10.4
      Obtained from: App Store
      Last Modified: 16/12/23, 00:16
      Kind: Universal
      Signed by: Apple Mac OS Application Signing, Apple Worldwide Developer Relations Certif
ication Authority, Apple Root CA
```

0x02 Hacking macOS Target

```
macadmin in ~
➜  dsconfigad -show
Active Directory Forest          = zion.orion
Active Directory Domain          = zion.orion
Computer Account                 = hell$

Advanced Options - User Experience
  Create mobile account at login = Disabled
     Require confirmation        = Disabled
  Force home to startup disk     = Enabled
     Mount home as sharepoint    = Enabled
  Use Windows UNC path for home  = Enabled
     Network protocol to be used = smb
  Default user Shell             = /bin/bash

Advanced Options - Mappings
  Mapping UID to attribute       = not set
  Mapping user GID to attribute  = not set
  Mapping group GID to attribute = not set
  Generate Kerberos authority    = Enabled

Advanced Options - Administrative
  Preferred Domain controller    = not set
  Allowed admin groups           = not set
  Authentication from any domain = Enabled
  Packet signing                 = allow
  Packet encryption              = allow
  Password change interval       = 14
  Restrict Dynamic DNS updates   = not set
  Namespace mode                 = domain
```

```
macadmin in ~
➜  dscl . list /Users UniqueID | awk '{print $1}' | sort -n | tail -6
chacal
daemon
l0gan
nobody
root
s1th
macadmin in ~
```

0x02 Hacking macOS Target

```
macadmin in ~/Desktop
➜  ldapsearch -H ldap://zion.orion -x -D "macadmin@zion.orion" -W -b "dc=zion,dc=orion" -s sub "(&(objectClass=us
er)(objectCategory=person))"
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=zion,dc=orion> with scope subtree
# filter: (&(objectClass=user)(objectCategory=person))
# requesting: ALL
#

# Administrator, Users, zion.orion
dn: CN=Administrator,CN=Users,DC=zion,DC=orion
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Administrator
description: Built-in account for administering the computer/domain
distinguishedName: CN=Administrator,CN=Users,DC=zion,DC=orion
instanceType: 4
whenCreated: 20231026201813.0Z
whenChanged: 20240219234125.0Z
uSNCreated: 8196
memberOf: CN=Group Policy Creator Owners,CN=Users,DC=zion,DC=orion
memberOf: CN=Domain Admins,CN=Users,DC=zion,DC=orion
memberOf: CN=Enterprise Admins,CN=Users,DC=zion,DC=orion
memberOf: CN=Schema Admins,CN=Users,DC=zion,DC=orion
memberOf: CN=Remote Desktop Users,CN=Builtin,DC=zion,DC=orion
```

```
macadmin in ~/Desktop
➜  ldapsearch -H ldap://zion.orion -x -D "macadmin@zion.orion" -W -b "dc=zion,dc=orion" -s sub "(&(objectClass=us
er))" sAMAccountName | grep "sAMAccountName" | awk '{print $2}'
Enter LDAP Password:
requesting:
Administrator
Guest
DC01$
krbtgt
admin
DESKTOP-DD3R17L$
macadmin
HELL$
dico
macadmin2
teste02
teste03
teste04
teste05
teste01
macadmin in ~/Desktop took 3.2s
```

Password Spray

```
### Check groups for user L0gan

l0gan@macLab % id l0gan
uid=413499013(l0gan) gid=1514433173(ZION\Domain Users) groups=1514433173(ZION\Domain Users),
502(awagent_enrolled),12(everyone),20(staff),62(netaccounts),501(awagent), 81465039(ZION\users_teste)
```

0x02 Hacking macOS Target

Disable Protection Solution (like EDR / XDR / DLP / SIEM / Proxy / Etc…)

```
### Turn off Security Solution (System)

sudo launchctl stop /Library/LaunchDaemons/com.X-Protection.plist
sudo launchctl unload /Library/LaunchDaemons/com.X-Protection.plist

sudo launchctl stop /Library/LaunchAgents/com.X-Protection.plist
sudo launchctl unload /Library/LaunchAgents/com.X-Protection.plist

### Turn off Security Solution (User)
launchctl stop ~/Library/LaunchAgents/com.X-Protection.plist
launchctl unload ~/Library/LaunchAgents/com.X-Protection.plist
```

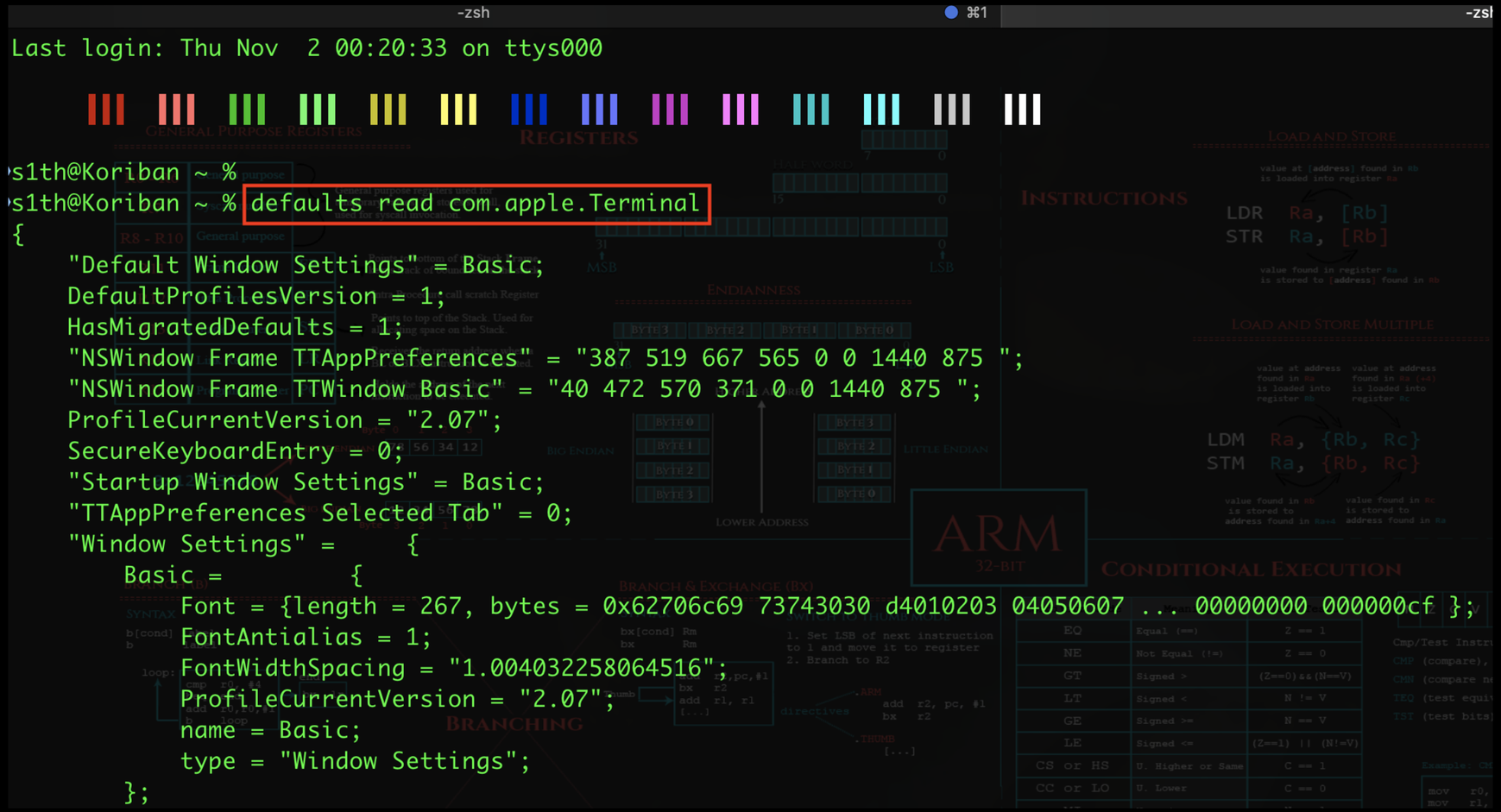
0x02 Hacking macOS Target

```
bash-3.2$
bash-3.2$ mdatp exclusion list
==========================================
No exclusions
==========================================
bash-3.2$
bash-3.2$
```

```
bash-3.2$
bash-3.2$ mdatp exclusion list
==========================================
Excluded folder
Path: "/tmp/exclusion/"
==========================================
bash-3.2$
bash-3.2$
```

Listing the exclusion folders of an antivirus can be dangerous, as it provides an attacker with information about which files and folders are intentionally excluded from antivirus scanning.
*Note: In this example, we used Microsoft ATP, but the same check can be performed with other vendors.

0x02 Hacking macOS Target



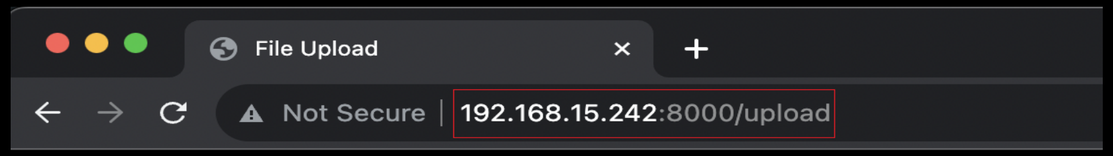
0x02 Hacking macOS Target

```
s1th@Koriban ~ % mdfind xprotect
2023-10-23 23:37:37.597 mdfind[42875:1773091] [UserQueryParser] Loading keywords and predicates for locale "pt_BR"
2023-10-23 23:37:37.600 mdfind[42875:1773091] [UserQueryParser] Loading keywords and predicates for locale "pt"
/System/Library/Sandbox/Profiles/com.apple.XprotectFramework.AnalysisService.sb
/System/Library/FeatureFlags/Domain/XProtect.plist
/System/Library/PrivateFrameworks/XprotectFramework.framework/Versions/A/XPCServices/XProtectBehaviorService.xpc/Contents/Resou
s/com.apple.Xprotec
/System/Library/Priv
/System/Library/Priv
tectUpdateService
/System/Library/Priv
/System/Library/Priv
rvice
/System/Library/Priv
/System/Library/Priv
/System/Library/Priv
rotectBehaviorServi
/Library/Apple/Syst
/Library/Apple/Syst
/Library/Apple/Syst
/Users/s1th/Desktop
/Library/Apple/Syst
/Library/Apple/Syst
/Library/Apple/Syst
/Library/Apple/Syst
/Library/Apple/Syst
/Library/Apple/Syst
/Library/Apple/Syst
/Library/Apple/Syst
s1th@Koriban ~ %
s1th@Koriban ~ % mdfind -name stdlib.h
2023-10-23 23:37:02.270 mdfind[42842:1772471] [UserQueryParser] Loading keywords and predicates for locale "pt_BR"
2023-10-23 23:37:02.271 mdfind[42842:1772471] [UserQueryParser] Loading keywords and predicates for locale "pt"
/Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdlib.h
/Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdlib.h
/Library/Developer/CommandLineTools/SDKs/MacOSX13.3.sdk/usr/include/stdlib.h
/Library/Developer/CommandLineTools/SDKs/MacOSX13.3.sdk/usr/include/c++/v1/stdlib.h
/Library/Developer/CommandLineTools/SDKs/MacOSX13.3.sdk/usr/include/xlocale/_stdlib.h
/Library/Developer/CommandLineTools/SDKs/MacOSX14.0.sdk/usr/include/c++/v1/stdlib.h
/Library/Developer/CommandLineTools/SDKs/MacOSX14.0.sdk/usr/include/xlocale/_stdlib.h
/Library/Developer/Comm
/Library/Frameworks/Pyt
/usr/local/Cellar/gcc/1
/usr/local/Cellar/gcc/1
/usr/local/Cellar/gcc/1
/usr/local/Cellar/gcc/1
/Library/Developer/Comm
s1th@Koriban ~ %
s1th@Koriban ~ % _
```

```
macadmin in ~
➜  mdfind password
2024-02-19 21:58:51.193 mdfind[7407:715014] [UserQueryParser] Loading keywords and predicates for locale "en_US"
2024-02-19 21:58:51.193 mdfind[7407:715014] [UserQueryParser] Loading keywords and predicates for locale "en"
/Users/macadmin/test-pass.txt
/Library/WebServer/share/httpd/manual/mod/quickreference.html.ko.euc-kr
/Library/WebServer/share/httpd/manual/mod/directives.html.ko.euc-kr
/Library/WebServer/share/httpd/manual/mod/mod_auth_basic.html.ko.euc-kr
/Library/WebServer/share/httpd/manual/howto/auth.html.ko.euc-kr
/Library/WebServer/share/httpd/manual/howto/htaccess.html.ko.euc-kr
/Library/WebServer/share/httpd/manual/programs/ab.html.ko.euc-kr
/System/Library/LaunchDaemons/com.apple.PasswordService.plist
/System/Library/Frameworks/PDFKit.framework/Versions/A/Resources/Base.lproj/NewPasswordView.nib
/System/Library/Frameworks/PDFKit.framework/Versions/A/Resources/Base.lproj/PDFPasswordAlert.nib
```

0x02 Hacking macOS Target

```
s1th@Koriban Desktop %
s1th@Koriban Desktop % ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether [redacted]
        inet 192.168.15.242 netmask 0xffffff00 broadcast 192.168.15.255
        media: autoselect
        status: active
s1th@Koriban Desktop %
s1th@Koriban Desktop % python -m uploadserver
File upload available at /upload
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
```



File Upload

Choose Files database

Token (only needed if server was started with token option):

Submit

Uploading database:

Success: Files accepted

+ Onedrive / Dropbox / GoogleDrive / Etc…

+ HTTP / HTTPS
- https://tmpfiles.org
- https://pastebin.com
- https://transfer.sh
- https://anonfile.net
- https://gofile.io
- https://send.cm

+ C2
+ E-mail / FTP / FTPS
+ Mass Storage / Bluetooth
+ DNS Exfiltration
+ Others Techniques

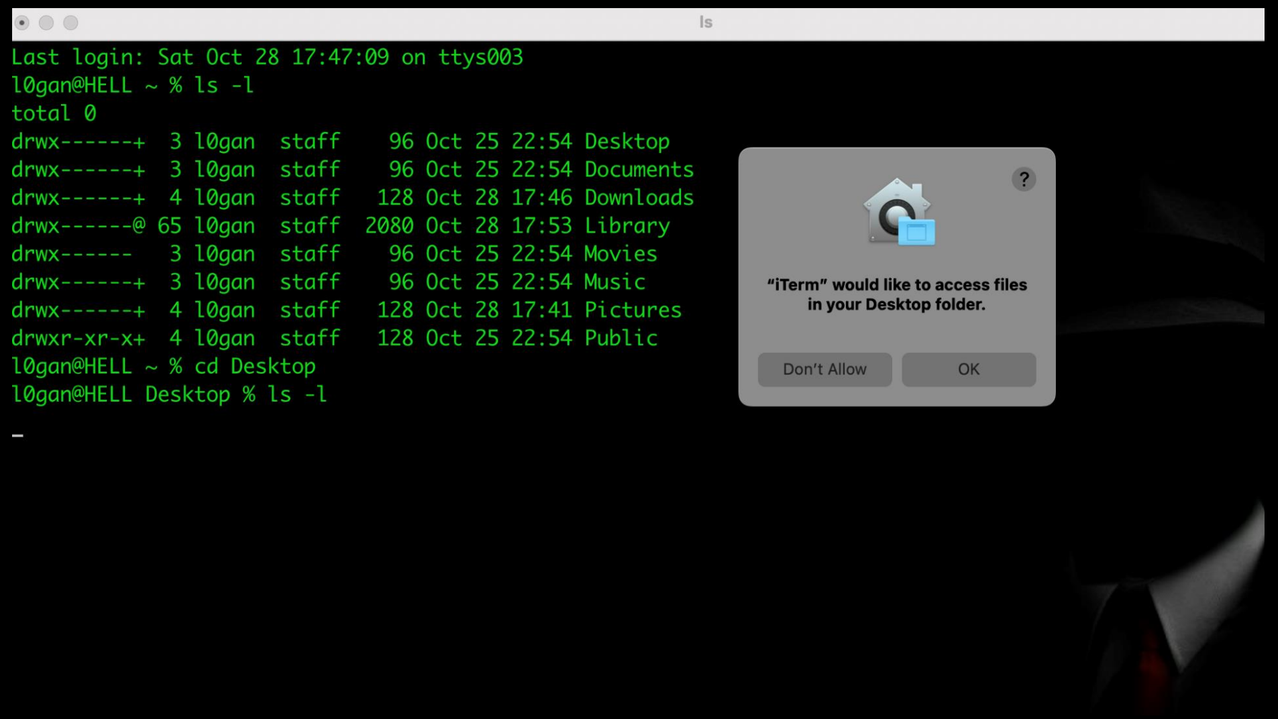
```
### Listar pastas compartilhadas de uma maquina em dominio
$ smbutil view //dominio.alvo:l0gan.user@dc-server-share


### Montar um compartilhamento do Server AD (Share)
$ mount_smbfs //l0gan.user@dc-server-share/SYSVOL ~/Desktop/Files
```

0x03 POC Extra (Bypass TCC)

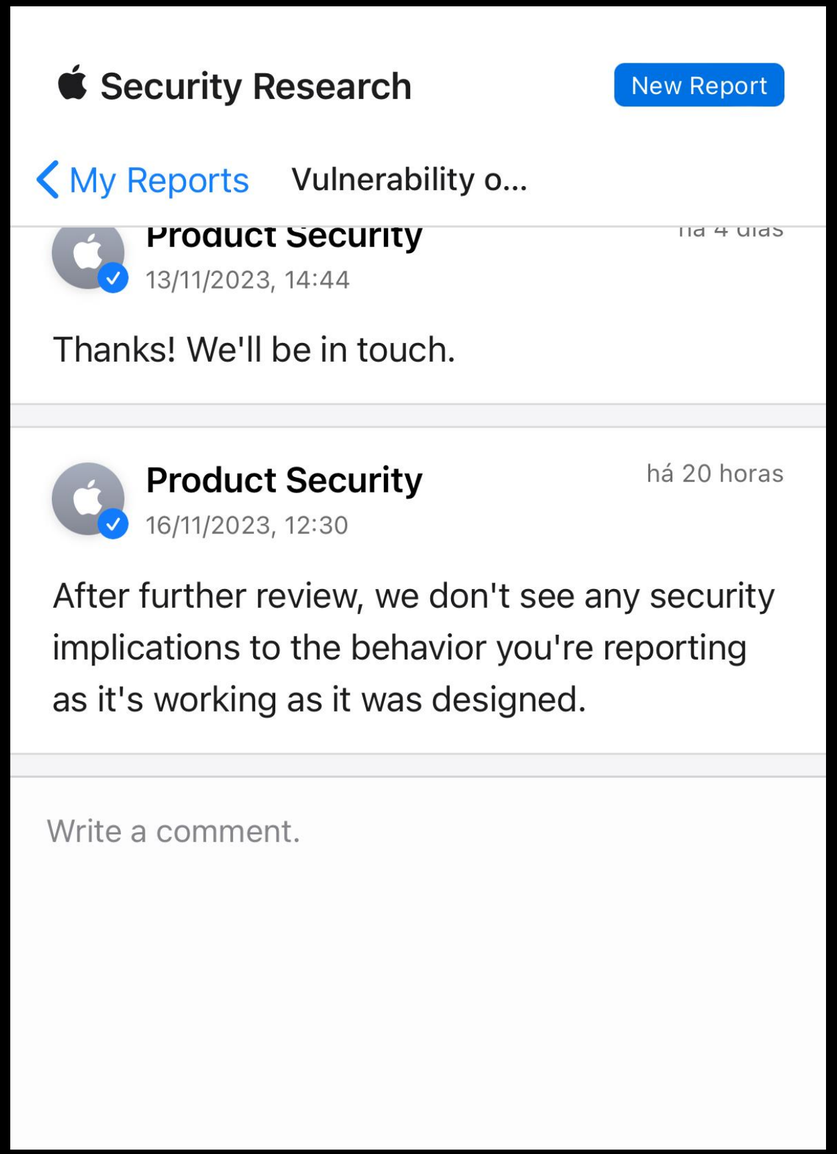
TCC(Transparency, Consent and Control) – Included in macOS since version 10.11 El Capitan

A bypass in macOS TCC is dangerous because it can compromise privacy, security, and system integrity by allowing apps or process to access sensitive resources without consent.

```
### TCC (Privacy Protections)

~/Desktop
~/Documents
~/Downloads
iCloud Drive
etc...

### TCC (Not Protected)

/tmp
~/.ssh
```



0x03 POC Extra (Bypass TCC)



Vulnerability found on TCC (Transparency Consent and Control)

- Access and modify of files protected by the system (TCC+ SSV+SIP).
- Bypass TCC component in macOS does not validate the use of the "open ." the command must block the access to the folder from the terminal to the finder.

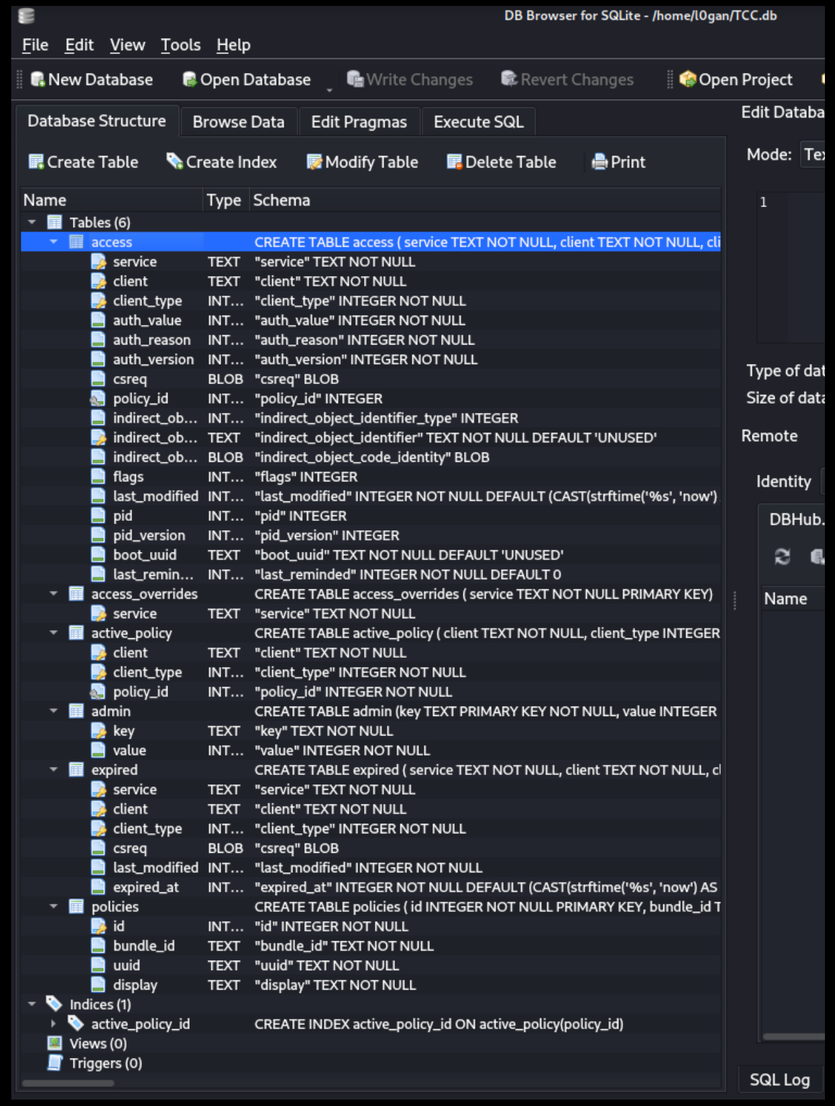
Risk:

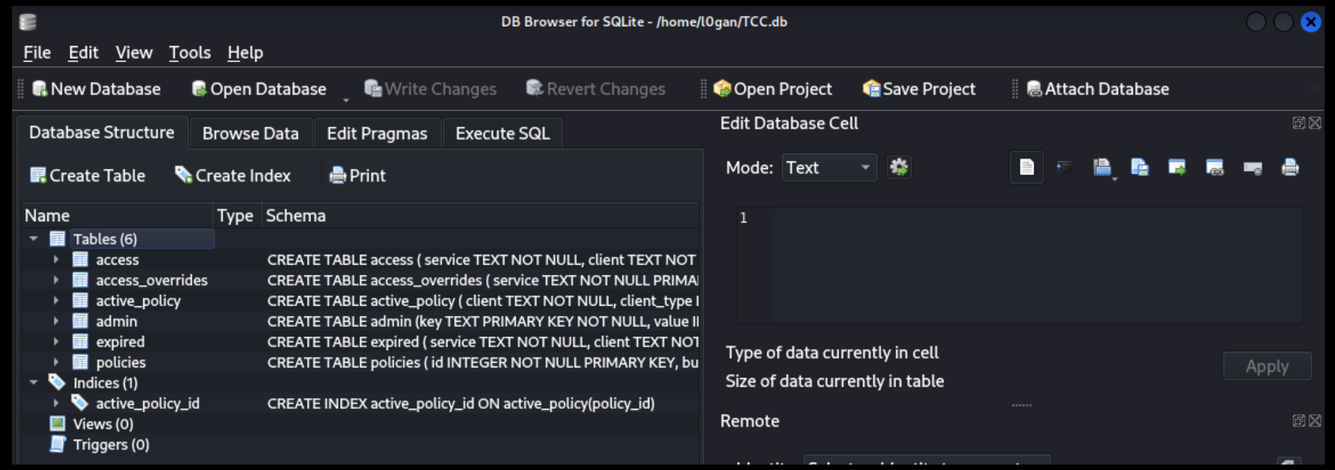
Drop file with new TCC.db with a malicious entry to disable some security protections that could be explored by another binary (like malware).

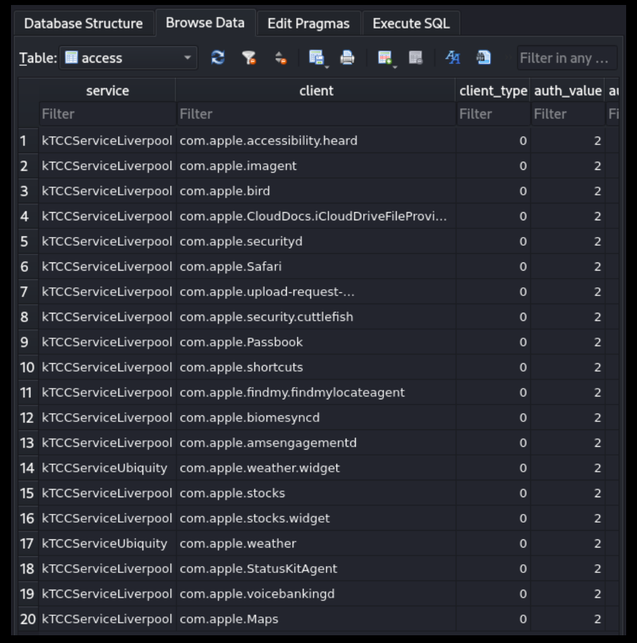
Resolution:

In my opinion, TCC.db should have a flag created in the operating system based on the hardware and the operating system to ensure that it should not be possible to rewrite TCC.db by an installation generated by another machine.

0x03 POC Extra (Bypass TCC)

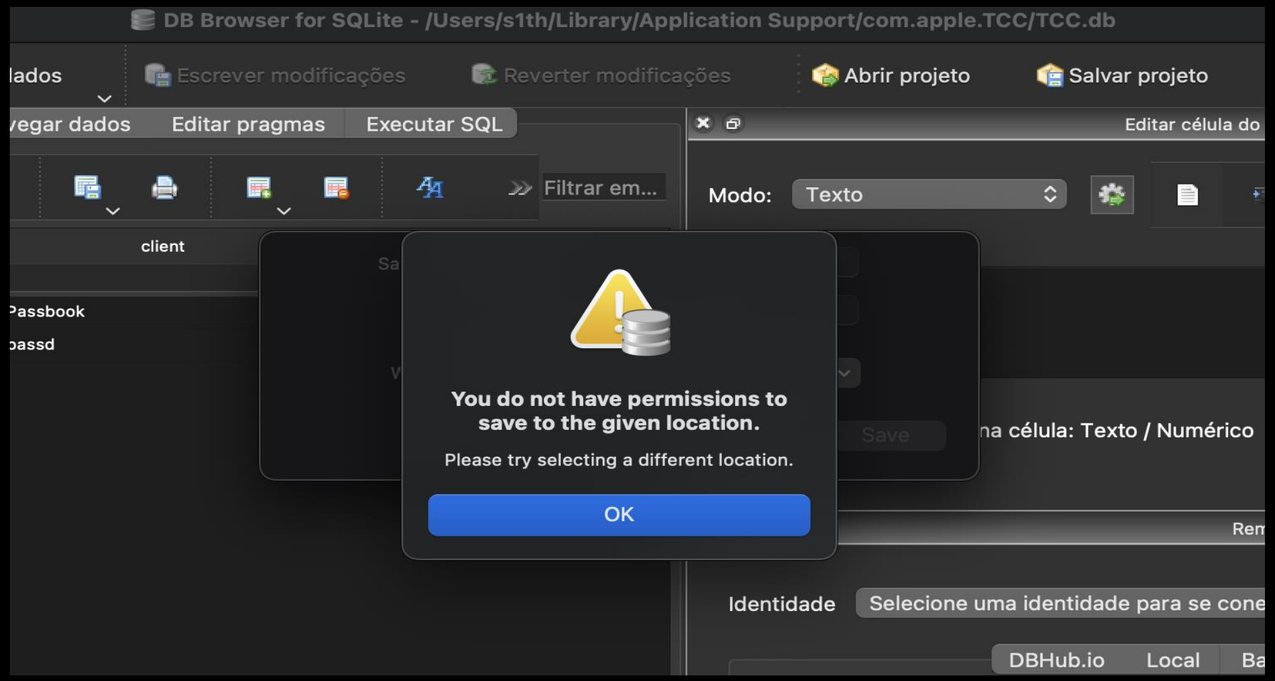






0x03 POC Extra (Bypass TCC)

```
s1th@Koriban ~ %
s1th@Koriban ~ % cd Library/Application\ Support/com.apple.TCC
s1th@Koriban com.apple.TCC % ls -l
total 0
ls: .: Operation not permitted
s1th@Koriban com.apple.TCC %
s1th@Koriban com.apple.TCC % csrutil status
System Integrity Protection status: enabled.
s1th@Koriban com.apple.TCC %
```




```
Last login: Thu May  9 15:09:53 on console

s1th@Koriban ~ % cd Library/Application\ Support/com.apple.TCC
s1th@Koriban com.apple.TCC %
s1th@Koriban com.apple.TCC % ls -l
total 160
drwxr-xr-x  6 s1th  staff    192 24 Abr 02:52 AdhocSignatureCache
-rw-r--r--@ 1 s1th  staff  81920  4 Mai 22:35 TCC.db
s1th@Koriban com.apple.TCC % csrutil status
System Integrity Protection status: disabled.
s1th@Koriban com.apple.TCC %
s1th@Koriban com.apple.TCC % _
```

```
-zsh
Last login: Thu May  9 20:54:54 on ttys000

s1th@Koriban ~ % cd /Library/Application\ Support/com.apple.TCC
s1th@Koriban com.apple.TCC %
s1th@Koriban com.apple.TCC % ls -l
total 168
drwxr-xr-x  25 root  wheel    800 24 Abr 02:52 AdhocSignatureCache
-rw-r--r--   1 root  wheel  20480  9 Mai 15:09 REG.db
-rw-r--r--   1 root  wheel  65536  2 Mai 22:25 TCC.db
s1th@Koriban com.apple.TCC % _
```

0x03 POC Extra (Bypass TCC)

POC ->

Bypass

Replace the TCC.db file located in a protected folder:~/Library/Application Support/com.apple.TCC with a new modified TCC.db.

Automator is an application developed by Apple Inc. for macOS, which can be used to automate repetitive tasks through point-and-click or drag and drop. Automator enables the repetition of tasks across a wide variety of programs, including Finder, Safari, Calendar, Contacts and others.

0x04 Native macOS Tools

```
csrutil    -> Configure system security policies (SIP)
codesign   -> create and manipulate code signatures
dsconfigad-> retrieves/changes configuration for Active Directory
dscl       -> Directory Service command line utility
dtrace     -> generic front-end to the DTrace facility
fs_usage   -> report system calls and page faults related to filesystem activity in real-time
Launchctl -> Manage and Inspect daemons, agents and XPC Services (PLIST)
Lipo       -> create or operate on universal files
mdfind     -> finds files matching a given query
nettop     -> Display updated information about the network
networksetup -> configuration tool for network settings in System Preferences.
nm         -> display name list (symbol table)
osalang    -> information about installed OSA languages
otool      -> object file displaying tool like a objdump and ldd
plutil     ->  property list utility (Convert plist files)
security   -> Command line interface to keychains and Security framework
spctl      -> SecAssessment system policy security (Gatekeeper)
sysctl     -> get or set kernel state

........
```

0x05 Conclusion

- Are your SOC and Blue Team monitoring and protecting the company from attacks?
- Are the controls really effectives and well implemented?
- Are your systems updated and with last security patches?
- Do you make security tests (Pentest) recurrent in your macOS endpoints?
- Do you have a well oriented team or update service with the last published vulnerabilities?

"A motivated attacker achieves his/her goals regardless of time"

0x06 Reference

Hacking is a way of life

My First Publication of this research (Nullbyte 2019)
https://github.com/loganbr/Presentations/blob/main/2019%20-%20Nullbyte%20-%20macOS%20Hacking%20Tricks.pdf

Research about malwares (mach-o files) for macOS(H2HC 2016)
https://github.com/loganbr/Presentations/blob/main/2016%20-%20H2HC%20-%20R3v3rs1ng%20on%20Mach-O%20File%20%20Version%200.pdf

Cedric Owens - Gone Apple Pickin: Red Teaming MacOS Environments in 2021
https://www.youtube.com/watch?v=IiMladUbL6E&t=93s

Objective-See / Objective ByTheSea
https://objective-see.org
https://objectivebythesea.org

Research Group Focus on Apple Devices #Attackium

https://discord.gg/tSpGtcUHVJ

Thanks a Lot

Any Questions ?

ricardologanbr@gmail.com

Twitter: @l0ganbr