



macOS Red Teaming on Corporate Scenarios

SPEAKER:

Ricardo Logan

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 macOS TCC Bypass

0x03 Conclusion

0x04 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

Is one of them safer?



?

0x01 macOS Security (Default / Corp)

Version	Release Date
Cheetah	2001
Puma	2001
Jaguar	2002
Panther	2003
Tiger	2005
Leopard	2007
Snow Leopard	2009
Lion	2011
Mountain Lion	2012
Mavericks	2013
Yosemite	2014
El Capitan	2015
Sierra	2016
High Sierra	2017
Mojave	2018
Catalina	2019
Big Sur	2020
Monterey	2021
Ventura	2022
Sonoma	2023
Sequoia	2024

The first version of macOS I started using

ARM/Intel
(Supported) + ARM/Intel
(Supported) + ARM/Intel
(Supported) + ARM/Intel
Is coming...

SIP

FileVault

Xprotect

Secure Boot

Gatekeeper

TCC

SSV



0x01 macOS Security (Default / Corp)

MDM

VA

EDR

XDR

DLP

CASB

PROXY

**Firewall
Host**

SIEM

And now?

Where should I go?



0x02 macOS TCC Bypass

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)
```

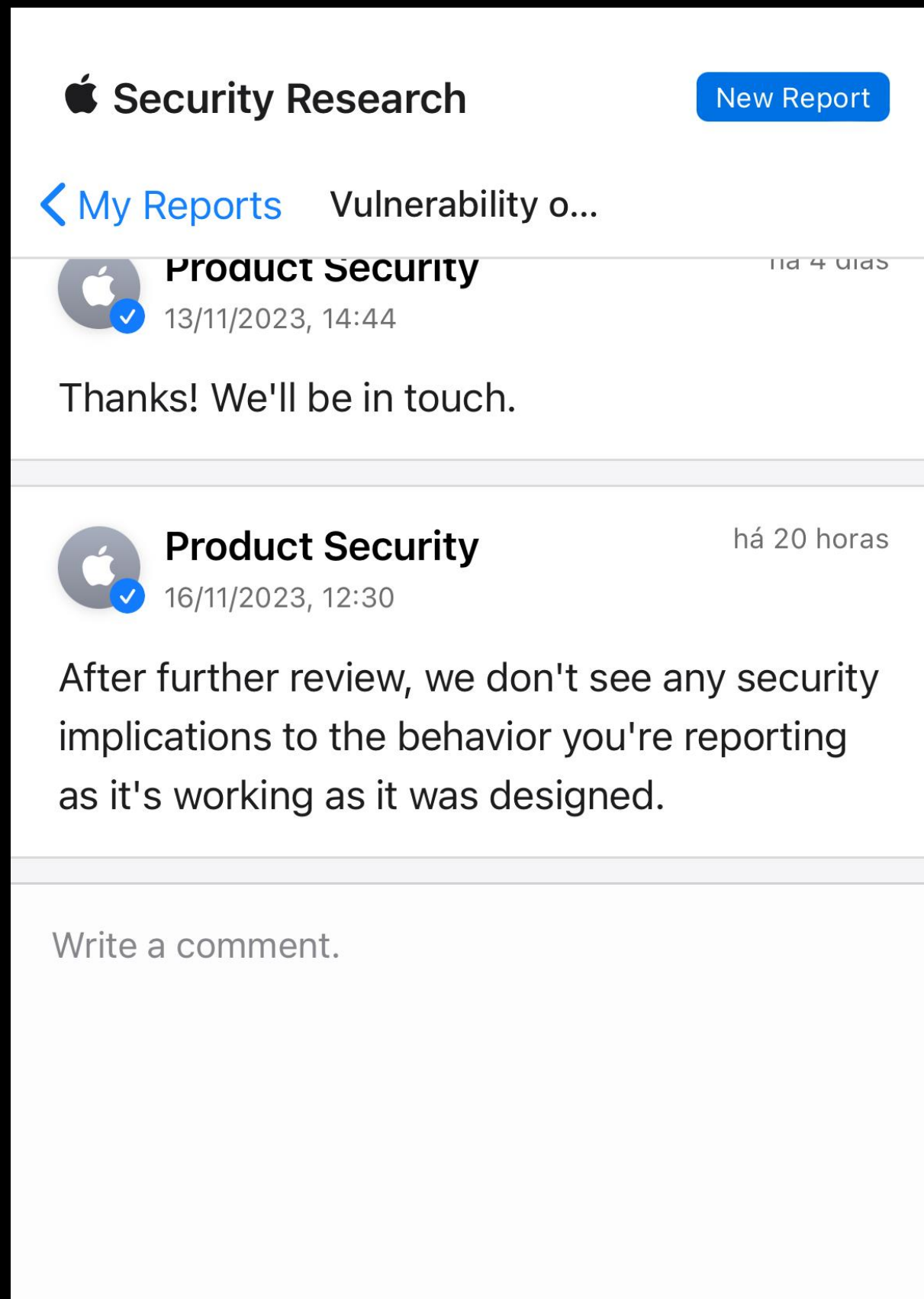
```
/tmp
```

```
~/.ssh
```

```
is
Last login: Sat Oct 28 17:47:09 on ttys003
l0gan@HELL ~ % ls -l
total 0
drwx-----+  3 l0gan  staff   96 Oct 25 22:54 Desktop
drwx-----+  3 l0gan  staff   96 Oct 25 22:54 Documents
drwx-----+  4 l0gan  staff  128 Oct 28 17:46 Downloads
drwx-----@ 65 l0gan  staff 2080 Oct 28 17:53 Library
drwx-----  3 l0gan  staff   96 Oct 25 22:54 Movies
drwx-----+  3 l0gan  staff   96 Oct 25 22:54 Music
drwx-----+  4 l0gan  staff  128 Oct 28 17:41 Pictures
drwxr-xr-x+  4 l0gan  staff  128 Oct 25 22:54 Public
l0gan@HELL ~ % cd Desktop
l0gan@HELL Desktop % ls -l
```



0x02 macOS TCC Bypass



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.

0x02 macOS TCC Bypass

DB Browser for SQLite - /home/I0gan/TCC.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

Database Structure Browse Data Edit Pragmas Execute SQL

Create Table Create Index Modify Table Delete Table Print

Name	Type	Schema
Tables (6)		
access	CREATE TABLE access (service TEXT NOT NULL, client TEXT NOT NULL, cl	
service	TEXT	"service" TEXT NOT NULL
client	TEXT	"client" TEXT NOT NULL
client_type	INT...	"client_type" INTEGER NOT NULL
auth_value	INT...	"auth_value" INTEGER NOT NULL
auth_reason	INT...	"auth_reason" INTEGER NOT NULL
auth_version	INT...	"auth_version" INTEGER NOT NULL
csreq	BLOB	"csreq" BLOB
policy_id	INT...	"policy_id" INTEGER
indirect_ob...	INT...	"indirect_object_identifier_type" INTEGER
indirect_ob...	TEXT	"indirect_object_identifier" TEXT NOT NULL DEFAULT 'UNUSED'
indirect_ob...	BLOB	"indirect_object_code_identity" BLOB
flags	INT...	"flags" INTEGER
last_modified	INT...	"last_modified" INTEGER NOT NULL DEFAULT (CAST(strftime('%s', 'now'))
pid	INT...	"pid" INTEGER
pid_version	INT...	"pid_version" INTEGER
boot_uuid	TEXT	"boot_uuid" TEXT NOT NULL DEFAULT 'UNUSED'
last_remin...	INT...	"last_reminded" INTEGER NOT NULL DEFAULT 0
access_overrides	CREATE TABLE access_overrides (service TEXT NOT NULL PRIMARY KEY)	
service	TEXT	"service" TEXT NOT NULL
active_policy	CREATE TABLE active_policy (client TEXT NOT NULL, client_type INTEGER	
client	TEXT	"client" TEXT NOT NULL
client_type	INT...	"client_type" INTEGER NOT NULL
policy_id	INT...	"policy_id" INTEGER NOT NULL
admin	CREATE TABLE admin (key TEXT PRIMARY KEY NOT NULL, value INTEGER	
key	TEXT	"key" TEXT NOT NULL
value	INT...	"value" INTEGER NOT NULL
expired	CREATE TABLE expired (service TEXT NOT NULL, client TEXT NOT NULL, cl	
service	TEXT	"service" TEXT NOT NULL
client	TEXT	"client" TEXT NOT NULL
client_type	INT...	"client_type" INTEGER NOT NULL
csreq	BLOB	"csreq" BLOB
last_modified	INT...	"last_modified" INTEGER NOT NULL
expired_at	INT...	"expired_at" INTEGER NOT NULL DEFAULT (CAST(strftime('%s', 'now')) AS
policies	CREATE TABLE policies (id INTEGER NOT NULL PRIMARY KEY, bundle_id T	
id	INT...	"id" INTEGER NOT NULL
bundle_id	TEXT	"bundle_id" TEXT NOT NULL
uuid	TEXT	"uuid" TEXT NOT NULL
display	TEXT	"display" TEXT NOT NULL
Indices (1)		
active_policy_id	CREATE INDEX active_policy_id ON active_policy(policy_id)	
Views (0)		
Triggers (0)		

Mode: Text

Type of data currently in cell

Size of data currently in table

Remote

Identity

DBHub

Name

SQL Log

DB Browser for SQLite - /home/I0gan/TCC.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

Create Table Create Index Print

Name	Type	Schema
Tables (6)		
access	CREATE TABLE access (service TEXT NOT NULL, client TEXT NOT	
access_overrides	CREATE TABLE access_overrides (service TEXT NOT NULL PRIMA	
active_policy	CREATE TABLE active_policy (client TEXT NOT NULL, client_type I	
admin	CREATE TABLE admin (key TEXT PRIMARY KEY NOT NULL, value II	
expired	CREATE TABLE expired (service TEXT NOT NULL, client TEXT NOT	
policies	CREATE TABLE policies (id INTEGER NOT NULL PRIMARY KEY, bu	
active_policy_id	CREATE INDEX active_policy_id ON active_policy(policy_id)	
Views (0)		
Triggers (0)		

Mode: Text

Type of data currently in cell

Size of data currently in table

Remote

Apply

Database Structure Browse Data Edit Pragmas Execute SQL

Table: access

	service	client	client_type	auth_value	ai
Filter	Filter	Filter	Filter	Filter	Fi
1	ktCCServiceLiverpool	com.apple.accessibility.heard	0	2	
2	ktCCServiceLiverpool	com.apple.imagent	0	2	
3	ktCCServiceLiverpool	com.apple.bird	0	2	
4	ktCCServiceLiverpool	com.apple.CloudDocs.iCloudDriveFileProvi...	0	2	
5	ktCCServiceLiverpool	com.apple.securityd	0	2	
6	ktCCServiceLiverpool	com.apple.Safari	0	2	
7	ktCCServiceLiverpool	com.apple.upload-request-...	0	2	
8	ktCCServiceLiverpool	com.apple.security.cuttlefish	0	2	
9	ktCCServiceLiverpool	com.apple.Passbook	0	2	
10	ktCCServiceLiverpool	com.apple.shortcuts	0	2	
11	ktCCServiceLiverpool	com.apple.findmy.findmylocateagent	0	2	
12	ktCCServiceLiverpool	com.apple.biomesyncd	0	2	
13	ktCCServiceLiverpool	com.apple.amsengagementd	0	2	
14	ktCCServiceUbiquity	com.apple.weather.widget	0	2	
15	ktCCServiceLiverpool	com.apple.stocks	0	2	
16	ktCCServiceLiverpool	com.apple.stocks.widget	0	2	
17	ktCCServiceUbiquity	com.apple.weather	0	2	
18	ktCCServiceLiverpool	com.apple.StatusKitAgent	0	2	
19	ktCCServiceLiverpool	com.apple.voicebankingd	0	2	
20	ktCCServiceLiverpool	com.apple.Maps	0	2	

0x02 macOS TCC Bypass

```
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 %
```

DB Browser for SQLite - /Users/s1th/Library/Application Support/com.apple.TCC/TCC.db

lados

Escrever modificações

Reverter modificações

Abrir projeto

Salvar projeto

regar dados

Editar pragmas

Executar SQL

Filtrar em...

Modo: Texto

client

Passbook

passd

You do not have permissions to save to the given location.

Please try selecting a different location.

OK

na célula: Texto / Numérico

Save

Identidade

Selecione uma identidade para se cone

DBHub.io

Local

Ba

```
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 %  
  
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 %
```

0x02 macOS TCC Bypass

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.

0x03 Conclusion

- Are your SOC and Blue Team monitoring and protecting the company from attacks?
- Are the controls really effective 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"

0x04 Reference

Hacking is a way of life

macOS Red Teaming on Corporate Scenarios (Defcon 2024)

<https://github.com/loganbr/Presentations/blob/main/2024%20-%20Defcon%20Red%20Team%20Village.pdf>

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