_> RED TEAM SPACE 2023 SPONSORS

KRAV MAGA HACKING

METABASE Q

Securetia

Vulseek

SOLIDARITY LABS

BLACKMANTISECURITY

Open - Sec

xelere
Making IT better

PUCARA

>>2023_EDITION<<

macOS Red Teaming 101

The information in this presentation will self-destruct in 5…4…3…2…1…

NAME:

SPEAKER NAME: Ricardo LOgan

DATE:

TALK DAT:03/11/2023

EKOPARTY
HACK THE PLANET

RED TEAM SPACE
EKOPARTY 2023

ABOUT ME

Ricardo L0gan

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

Outline

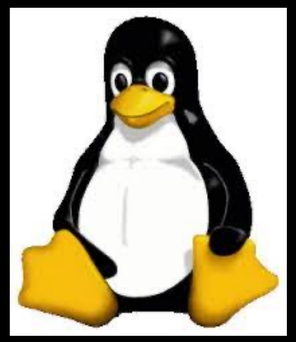
0x00 Motivation of Research
0x01 macOS Security (Default / Corp)
0x02 Hacking macOS target
0x03 Native macOS Tools
0x04 Reference
0x05 Conclusion

0x00 Motivation of Research

- Lockbit 2023
- CriptoMiner 2023
- Silver Sparrow 2021
- OSX/Linker 2019
- LoundMiner 2019
- Mac Auto Fixer 2018
- Crossrider (OSX/Shalyer) 2018
- OSX/MaMi 2018
- Others….

0x00 Motivation of Research

Is one of them safer?



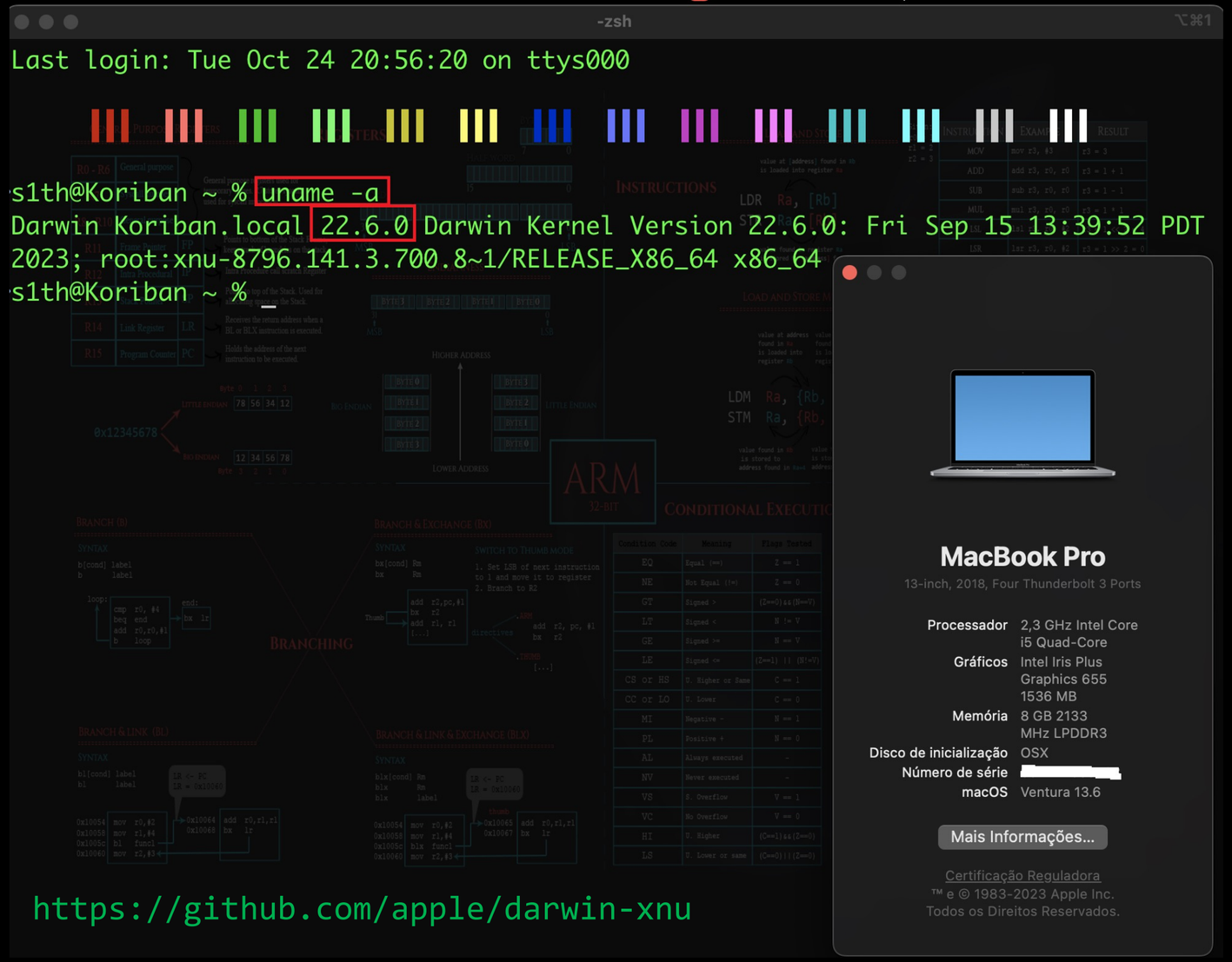
Windows

macOS®

?

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) chips.



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	ARM/Intel
Monterey	2021	(Supported) + ARM/Intel
Ventura	2022	(Supported) + ARM/Intel
Sonoma	2023	(Supported) + ARM/Intel

SIP (System Integrity Protection)

FileVault

Xprotect

Secure Boot

Gatekeeper

TCC

SSV

0x02 Hacking macOS Target

```
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> _
```

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.

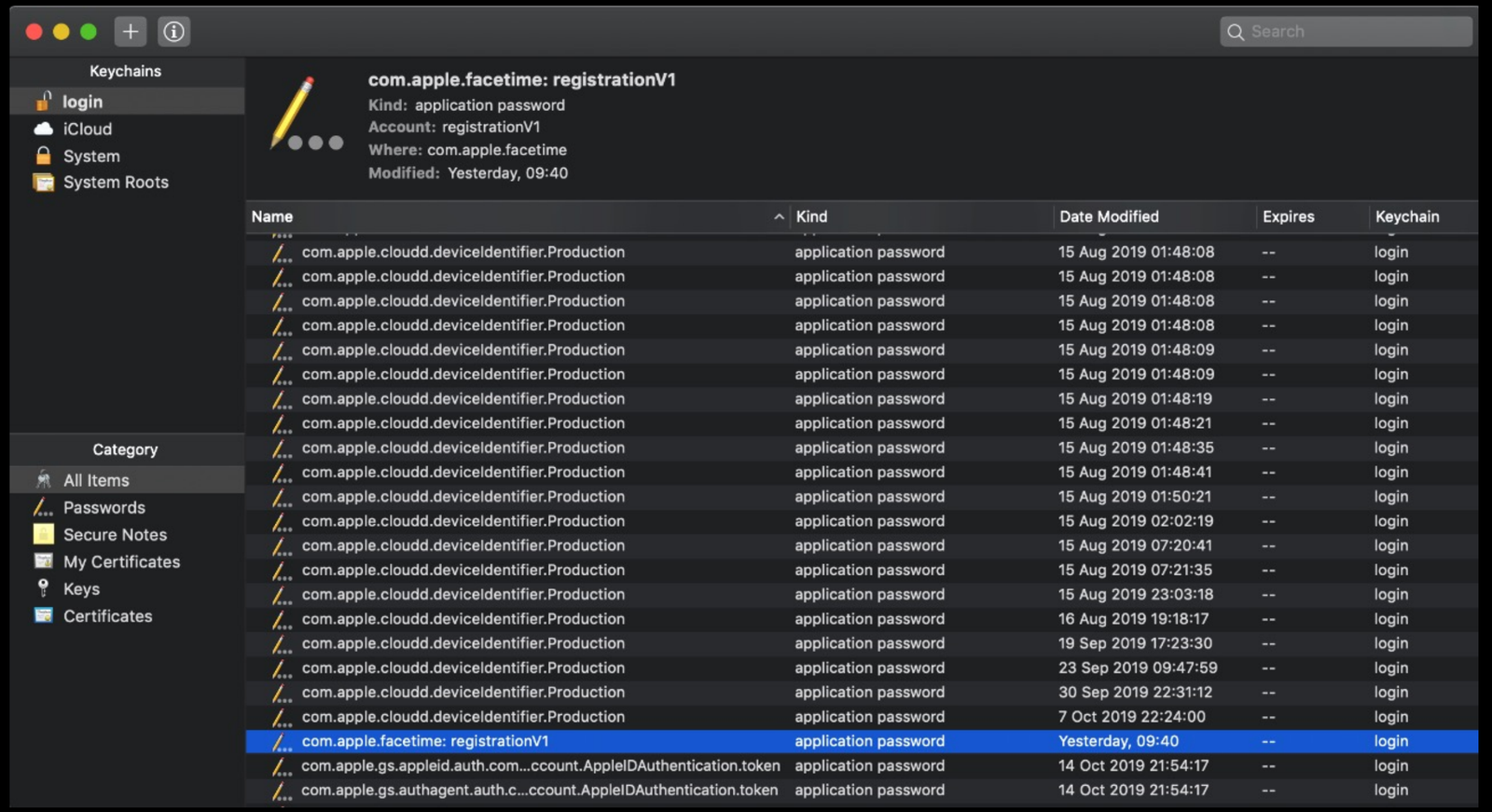
However, the presence of KEXT files can also pose a security challenge. As part of the kernel, they have deep system access, and if not properly managed, they can be exploited by attackers to compromise system security.

```
-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
```

0x02 Hacking macOS Target

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.



0x02 Hacking macOS Target

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]
└──────> $:> _
```


```
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]
```

```
l0gan@Zion-Inf3ct3d:[~/Library/Keychains][16:18:14]
└──────> $:> file login.keychain-db
login.keychain-db: Mac OS X Keychain File
```

0x02 Hacking macOS Target

```
### Lanchctl (System)

/Library/LaunchDaemons/
/Library/LaunchDaemons/

/Library/LaunchAgents/
/Library/LaunchAgents/

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

Lanchctl (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.

0x02 Hacking macOS Target

```
bash
l0gan@Zion-Inf3ct3d:[~/Library/LaunchAgents][23:01:10]
└──────> $:> pwd
/Users/l0gan/Library/LaunchAgents
l0gan@Zion-Inf3ct3d:[~/Library/LaunchAgents][23:01:12]
└──────> $:> ls -l
total 24
-rw-r--r--  1 l0gan  staff  685 Aug 21 18:31 com.dropbox.DropboxMacUpdate.agent.plist
-rw-r--r--@ 1 l0gan  staff  809 Oct  2 22:53 com.google.keystone.agent.plist
-rw-r--r--@ 1 l0gan  staff  915 Oct  2 22:53 com.google.keystone.xpcservice.plist
l0gan@Zion-Inf3ct3d:[~/Library/LaunchAgents][23:01:16]
└──────> $:> _
```

```
bash
l0gan@Zion-Inf3ct3d:[/Library/LaunchAgents][23:01:47]
└──────> $:> pwd
/Library/LaunchAgents
l0gan@Zion-Inf3ct3d:[/Library/LaunchAgents][23:01:49]
└──────> $:> ls -l
total 24
-rw-r--r--  1 root  wheel  674 Oct  7 22:25 com.bjango.istatmenus.agent.plist
-rw-r--r--  1 root  wheel  682 Oct  7 22:25 com.bjango.istatmenus.status.plist
-r-xr-xr-x  1 root  wheel  582 Jun 25 14:24 com.kaspersky.kav.gui.plist
l0gan@Zion-Inf3ct3d:[/Library/LaunchAgents][23:01:51]
└──────> $:> _
```

```
bash
l0gan@Zion-Inf3ct3d:[/Library/LaunchDaemons][23:02:23]
└──────> $:> pwd
/Library/LaunchDaemons
l0gan@Zion-Inf3ct3d:[/Library/LaunchDaemons][23:02:24]
└──────> $:> ls -l
total 48
-rw-r--r--  1 root  wheel   632 Sep 19 14:55 com.apple.installer.osmessagetracing.plist
-rw-r--r--  1 root  wheel   584 Oct  7 22:25 com.bjango.istatmenus.daemon.plist
-rw-r--r--  1 root  wheel   557 Oct  7 22:25 com.bjango.istatmenus.fans.plist
-rw-r--r--  1 root  wheel   608 Aug 15 21:58 com.bjango.istatmenus.installerhelper.plist
-r-xr-xr-x  1 root  wheel  1080 Jun 25 14:24 com.kaspersky.kav.plist
-rw-r--r--  1 root  wheel   382 Aug 15 07:31 org.wireshark.ChmodBPF.plist
l0gan@Zion-Inf3ct3d:[/Library/LaunchDaemons][23:02:26]
└──────> $:> _
```

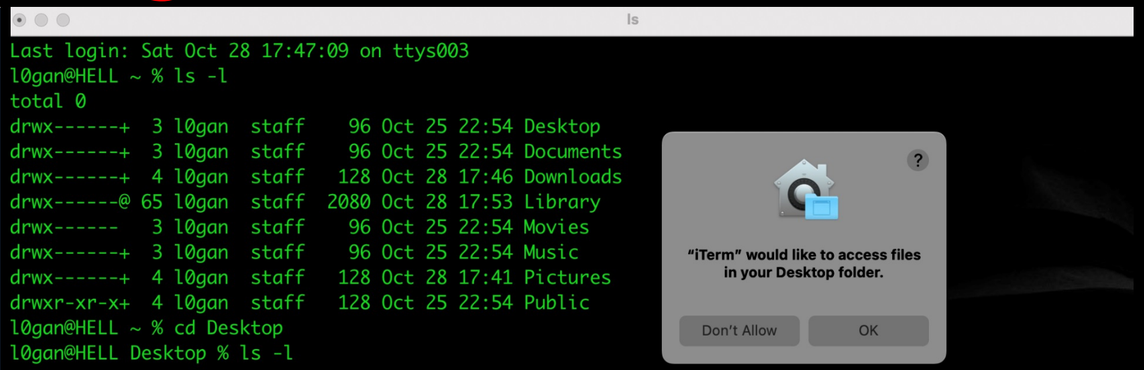
```
$ launchctl load file.plist
```

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

It contains properties and configuration settings for various programs. PLIST files are formatted in XML and based on Apple's Core Foundation DTD.

0x02 Hacking macOS Target

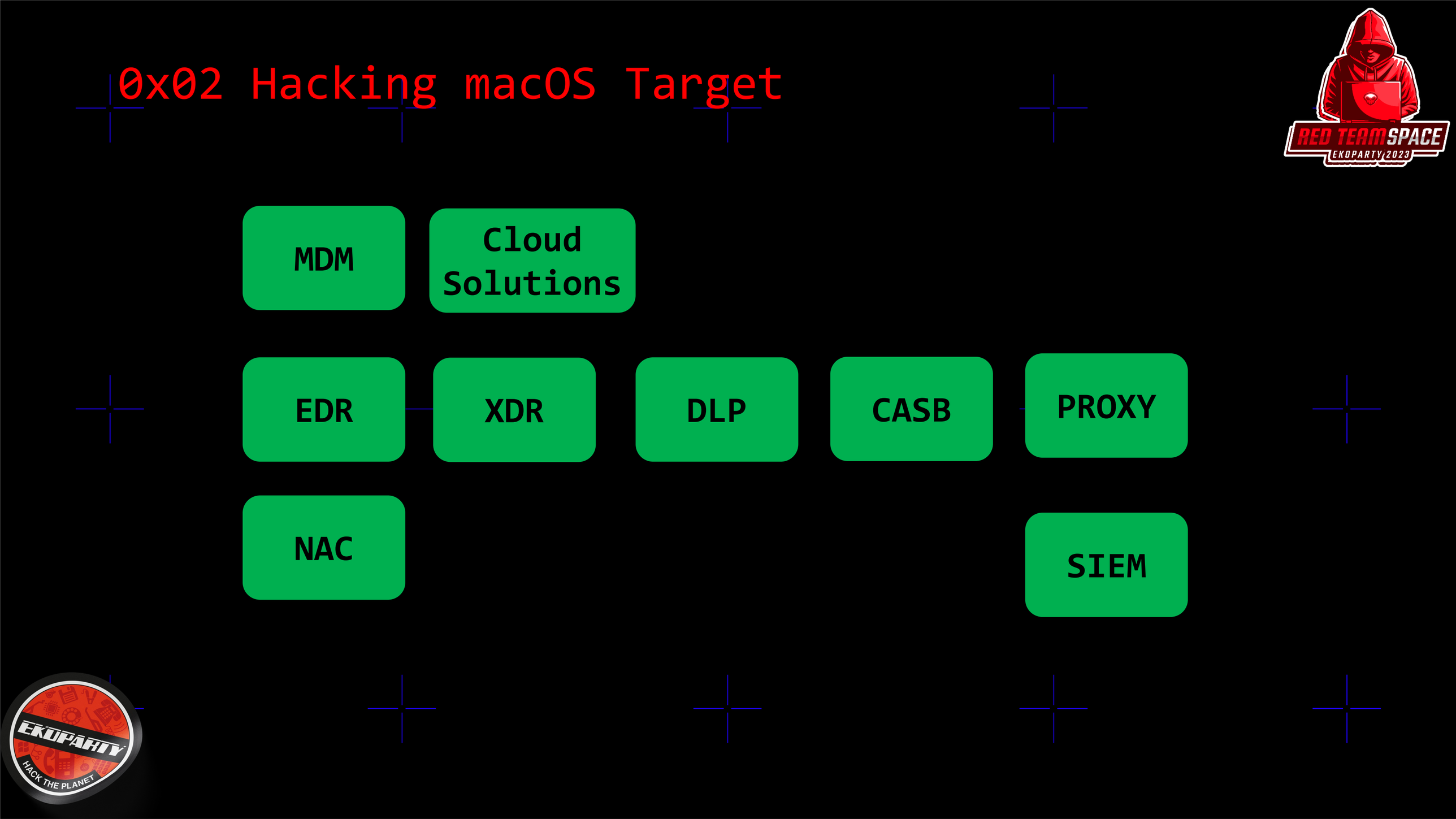
```
### TCC (Privacy Protections)

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

### TCC (Not Protected)

/tmp
~/.ssh
```



TCC(Transparency, Consent and Control)

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.

0x02 Hacking macOS Target

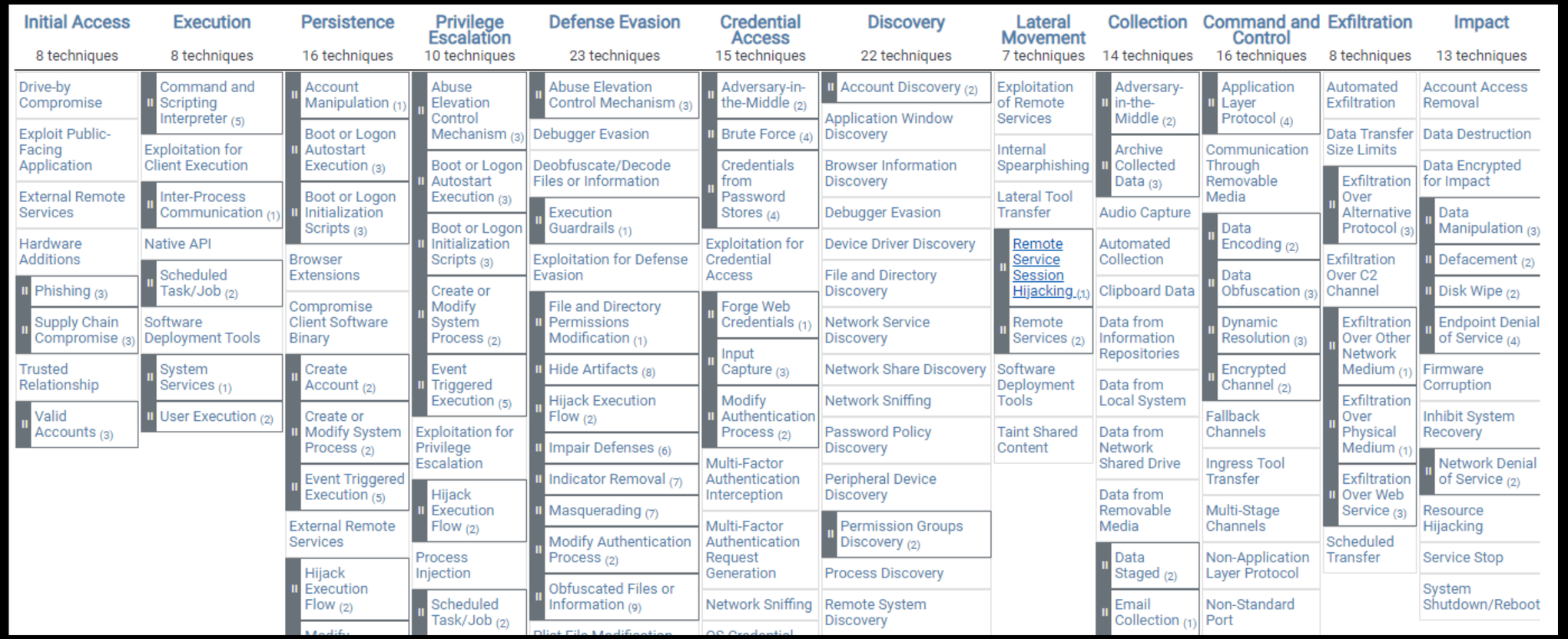


And now Where do I go?

* Use know threat skills and automation to find vulnerabilities in enterprise *

0x02 Hacking macOS Target

MITRE | ATT&CK®



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
8 techniques	8 techniques	16 techniques	10 techniques	23 techniques	15 techniques	22 techniques	7 techniques	14 techniques	16 techniques	8 techniques	13 techniques
Drive-by Compromise	Command and Scripting Interpreter (5)	Account Manipulation (1)	Abuse Elevation Control Mechanism (3)	Abuse Elevation Control Mechanism (3)	Adversary-in-the-Middle (2)	Account Discovery (2)	Exploitation of Remote Services	Adversary-in-the-Middle (2)	Application Layer Protocol (4)	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	Exploitation for Client Execution	Boot or Logon Autostart Execution (3)	Boot or Logon Autostart Execution (3)	Debugger Evasion	Brute Force (4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (3)	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
External Remote Services	Inter-Process Communication (1)	Boot or Logon Initialization Scripts (3)	Boot or Logon Initialization Scripts (3)	Deobfuscate/Decode Files or Information	Credentials from Password Stores (4)	Browser Information Discovery	Lateral Tool Transfer	Audio Capture	Data Encoding (2)	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
Hardware Additions	Native API	Browser Extensions	Create or Modify System Process (2)	Execution Guardrails (1)	Exploitation for Credential Access	Debugger Evasion	Remote Service Session Hijacking (1)	Automated Collection	Data Obfuscation (3)	Exfiltration Over C2 Channel	Data Manipulation (3)
Phishing (3)	Scheduled Task/Job (2)	Compromise Client Software Binary	Event Triggered Execution (5)	Exploitation for Defense Evasion	Forge Web Credentials (1)	Device Driver Discovery	Remote Services (2)	Clipboard Data	Dynamic Resolution (3)	Exfiltration Over Other Network Medium (1)	Defacement (2)
Supply Chain Compromise (3)	Software Deployment Tools	Create Account (2)	Exploitation for Privilege Escalation	File and Directory Permissions Modification (1)	Input Capture (3)	File and Directory Discovery	Software Deployment Tools	Data from Information Repositories	Encrypted Channel (2)	Exfiltration Over Physical Medium (1)	Disk Wipe (2)
Trusted Relationship	System Services (1)	Create or Modify System Process (2)	Hijack Execution Flow (2)	Hide Artifacts (8)	Modify Authentication Process (2)	Network Service Discovery	Taint Shared Content	Data from Local System	Fallback Channels	Exfiltration Over Web Service (3)	Endpoint Denial of Service (4)
Valid Accounts (3)	User Execution (2)	Event Triggered Execution (5)	Process Injection	Hijack Execution Flow (2)	Multi-Factor Authentication Interception	Network Share Discovery		Data from Network Shared Drive	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
		External Remote Services	Scheduled Task/Job (2)	Impair Defenses (6)	Multi-Factor Authentication Request Generation	Network Sniffing		Data from Removable Media	Multi-Stage Channels		Inhibit System Recovery
		Hijack Execution Flow (2)		Indicator Removal (7)	Network Sniffing	Password Policy Discovery		Data Staged (2)	Non-Application Layer Protocol		Network Denial of Service (2)
				Masquerading (7)		Peripheral Device Discovery		Email Collection (1)	Non-Standard Port		Resource Hijacking
				Modify Authentication Process (2)		Permission Groups Discovery (2)					Service Stop
				Obfuscated Files or Information (9)		Process Discovery					System Shutdown/Reboot
						Remote System Discovery					

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

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 ~ % system_profiler SPFirewallDataType
Firewall:

    Firewall Settings:

      Mode: Limit incoming connections to specific services and applications
      Applications:
          com.agilebits.onepassword7: Allow all connections
          com.apple.controlcenter: Allow all connections
          com.facebook.sonar.helper.Renderer: Allow all connections
          com.google.Chrome: Block all connections
          com.hexrays.ida64: Allow all connections
          com.rileytestut.AltServer: Allow all con
          com.spotify.client: Allow all connection
          org.python.python: Allow all connections
      Firewall Logging: Yes
      Stealth Mode: No

s1th@Koriban ~ %
```

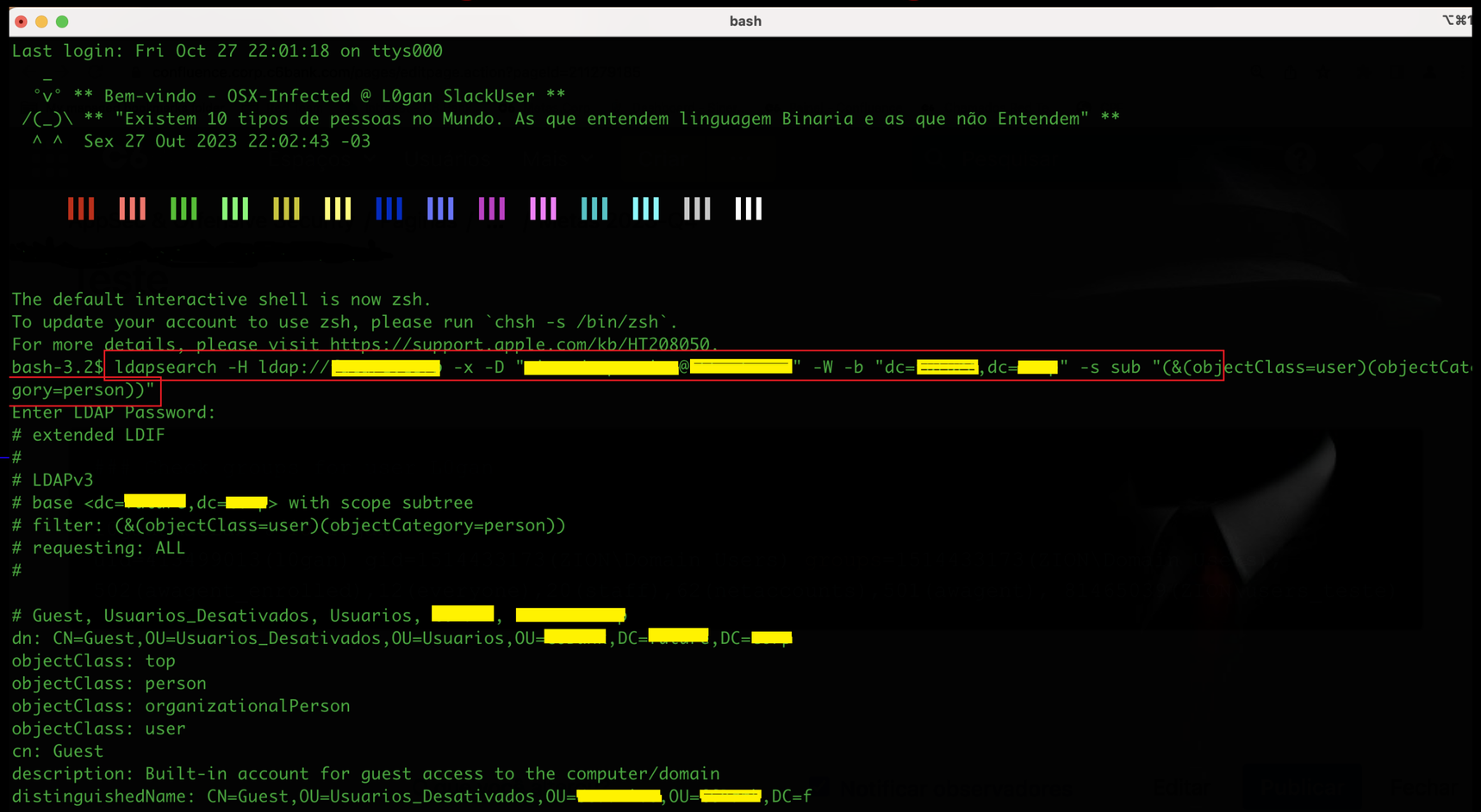
```
chacal@Hell ~
chacal@Hell ~ netstat -an | grep "LISTEN"
tcp4       0      0  *.631                  *.*                    LISTEN
tcp6       0      0  *.631                  *.*                    LISTEN
tcp4       0      0  *.3031                 *.*                    LISTEN
tcp6       0      0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

```
bash-3.2$ dsconfigad -show
Active Directory Forest          = [redacted]
Active Directory Domain          = [redacted]
Computer Account                 = [redacted]

Advanced Options - User Experience
  Create mobile account at login = Enabled
     Require confirmation        = Disabled
  Force home to startup disk     = Disabled
     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       = 0
  Restrict Dynamic DNS updates   = not set
  Namespace mode                 = domain
bash-3.2$
```

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

0x02 Hacking macOS Target




```
### 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
```

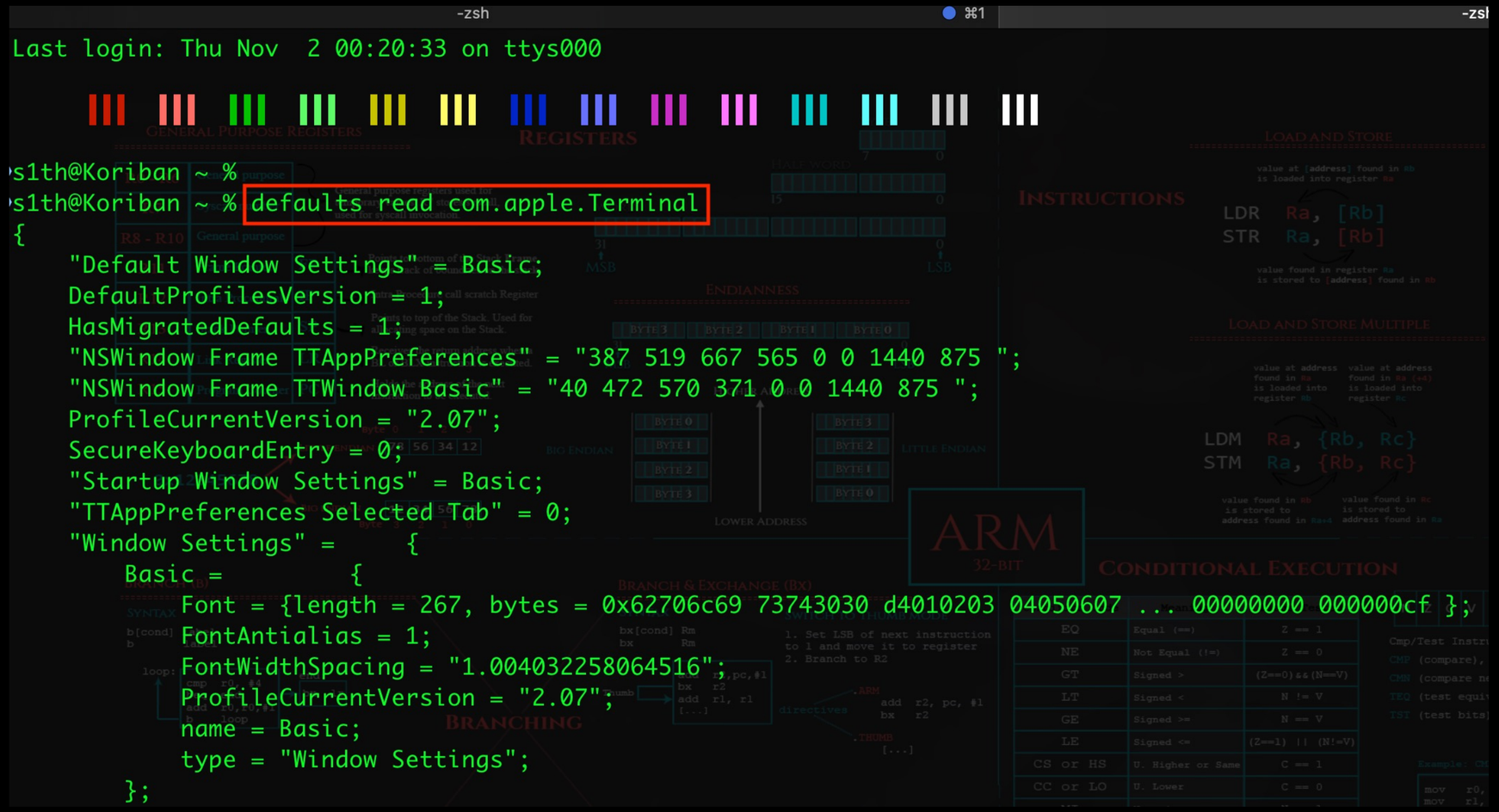
0x02 Hacking macOS Target

```
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 wich files and folders are intentionally excluded from antivirus scanning.
*Note: In this example, we used Microsoft ATP, but the same check can be performed wither other vendors.

0x02 Hacking macOS Target



0x02 Hacking macOS Target

Dump Creds of all non-service accounts

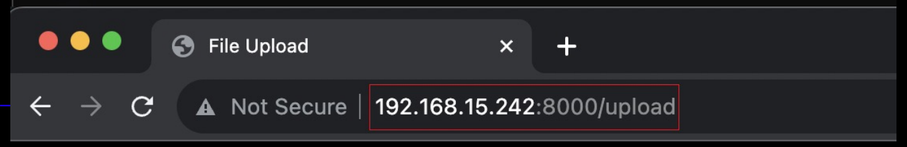
0x02 Hacking macOS Target

mdfind command


```
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.XprotectFramework.BehaviorService.sb
/System/Library/PrivateFrameworks/XprotectFramework.framework/Versions/A/XPCServices/XProtectUpdateService.xpc
/System/Library/PrivateFrameworks/XprotectFramework.framework/Versions/A/XPCServices/XProtectUpdateService.xpc/Contents/MacOS/X
tectUpdateService
/System/Library/PrivateFrameworks/XprotectFramework.framework/Versions/A/XPCServices/XprotectService.xpc
/System/Library/PrivateFrameworks/XprotectFramework.framework/Versions/A/XPCServices/XprotectService.xpc/Contents/MacOS/Xprotec
rvice
/System/Library/PrivateF
/System/Library/PrivateF
/System/Library/PrivateF
rotectBehaviorService
/Library/Apple/System/Li
/Library/Apple/System/Li
/Library/Apple/System/Li
/Users/s1th/Desktop/help
/Library/Apple/System/Li
/Library/Apple/System/Li
/Library/Apple/System/Li
/Library/Apple/System/Li
/Library/Apple/System/Li
/Library/Apple/System/Li
/Library/Apple/System/Li
/Library/Apple/System/Li
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/CommandLineTools/SDKs/MacOSX14.0.sdk/usr/include/stdlib.h
/Library/Frameworks/Python.framework/Versions/3.10/Resources/English.lproj/Documentation/tutorial/stdlib.html
/usr/local/Cellar/gcc/12.2.0/include/c++/12/stdlib.h
/usr/local/Cellar/gcc/12.2.0/include/c++/12/tr1/stdlib.h
/usr/local/Cellar/gcc/11.3.0_1/include/c++/11/stdlib.h
/usr/local/Cellar/gcc/11.3.0_1/include/c++/11/tr1/stdlib.h
/Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdlib.h
s1th@Koriban ~ %
s1th@Koriban ~ % _
```

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
        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

+ https://tmpfiles.org

+ https://pastebin.com

+ https://transfer.sh

+ https://anonfile.net

+ https://gofile.io

+ https://send.cm


```
### 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
```

+ DNS Exfiltration ???
+ Others Techniques???

0x03 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)
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

........
```

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

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

_> ¡GRACIAS POR PARTICIPAR!

KRAV MAGA HACKING

METABASE Q

Securetia

Vulseek

SOLIDARITY LABS

BLACKMANTISECURITY

Open - Sec

xelere
Making IT better

PUCARA

Thanks a Lot

Any Questions ?

ricardologanbr@gmail.com

https://github.com/loganbr/Presentations

Twitter: @l0ganbr