MACINTOSH FORENSICS IN 90 MINUTES

Simson L. Garfinkel
simsong@acm.org

June 18, 2019

Loosely based on:
CFRS 764 - Mac Forensics
Spring 2019

GEORGE MASON UNIVERSITY

All images from Wikipedia unless otherwise noted

ISSA
Information Systems Security Association
National Capital Chapter

Online for tonight's talk

Introduction

- GMU CFRS 764 — Mac Forensics
- History of MacOS
- What makes the Mac different

Mac Forensic Opportunities

- Unix/Linux/Windows forensic techniques that work on the Mac
- Mac-specific collection opportunities

Mac Forensic Challenges

- Pervasive cryptography and Apple's "T2" chip
- APFS
- Mac logging

Mac Forensic Tools

- Open Source
- Proprietary

CFRS 764 — Mac Forensics

Overview

- “Presents students with the concepts, tools, and techniques used for forensic analysis of the Macintosh based computers. Classes will consist of lectures on the Macintosh operating system, reverse engineering, forensic practice and research, followed by exercises conducted in a lab environment.”
  —REVISED FOR 2019!

Spring 2019 • Wednesday 7:20 – 10:00pm
Spring 2020 • Thursday 4:30 – 7:10pm (tentative)

**CFRS 780 is the reusable course number*

There’s a lot to Mac Forensics!
(Overview of CFRS 764)

- Course Overview/Administrative Items; History; Encryption
- Live System Analysis: Stored Data, Log files and File Structures
- Live System Analysis: The Storage Layer, Disk Partitioning and Mac Filesystems
- Disk imaging and working with disk images.
- Live System Analysis: Processes, Network Connections, and other stuff
- Memory Analysis: Memory Capture and Volatility.
- Users Directory Artifacts Analysis
- Using dtrace
- System and Global Artifacts Analysis
- Isolation
- iOS, iTunes, and iCloud Contributions
- Recent Research in Mac Forensics
- Final presentations and Exam Prep

A bit about me*

Photo Credit: Declan Mcculough

Simson L. Garfinkel, Ph.D.
https://simson.net/
simsong@acm.org

Interests: Security, Privacy, Digital Forensics

1987	MIT (Chemistry, Political Science, STS)
1988	Columbia University (MS Journalism)
1995	Vineyard.NET (ISP)
1998	Sandstorm Enterprises (Digital Forensics Tools)
2002-2005	MIT CSAIL (PhD Computer Science)
2006-2014	Naval Postgraduate School (Associate Professor)
2015-2016	National Institute of Standards and Technology (NIST)
2017-	US Census Bureau

*affiliations are provided for identification purposes only

This lecture is not about iOS forensics

There's a lot of resources for iOS forensics.

iOS forensics is significantly different than Mac forensics

- iOS apps are more restricted than Mac apps
- Macs have more functionality
- Macs have more storage
- MacOS has more history

macOS is changing faster than iOS

- Apple is hardening macOS
  —*Forensics on the mac is getting harder*
  —*Old approaches no loner work*
- Apple is adding more identity information
  —*Creates more forensic opportunities*
  —*Primarily useful for identity intelligence, not malware analysis*

A brief introduction to the Mac

Information Systems Security Association
National Capital Chapter

History of MacOS
What makes MacOS different

Mac History 1984-2001
System 1 – MacOS 9

Macintosh 128K
1984

Macintosh II
1987

iMac
1998

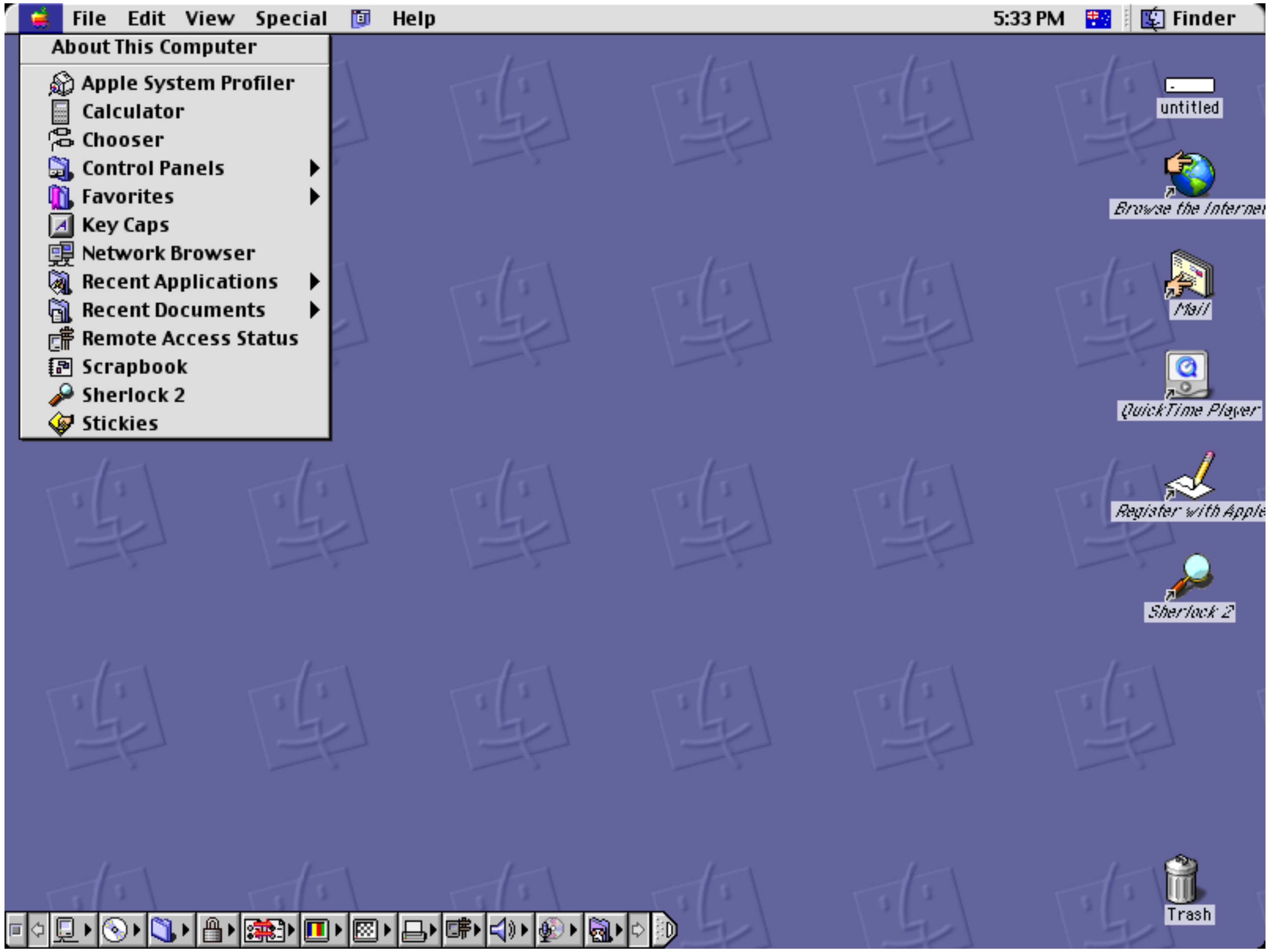
Key distinguishing features:

- Real mode operating system; no memory protection; cooperative multi-tasking.
- Smart peripherals favoring buses: Apple Desktop Bus, SCSI, USB
- Networked (AppleTalk: 1985-2009; TCP/IP: 1988-)
- Highly proprietary (Floppies; File System; etc.)

MacOS 9 —
The last classic Mac operating system (1999)

Introduced:

- Apple KeyChain
- Speech synthesis and recognition
- File encryption
- Ran on PowerPC



NeXT Computer

1985 — Founded by Steve Jobs

- Nine years after Apple

1987 — NeXT “Cube”

1988 — NeXTstation (bw & color)

1993 — NeXTSTEP ported to Intel

1995 — ported to SPARC and PA-RISC

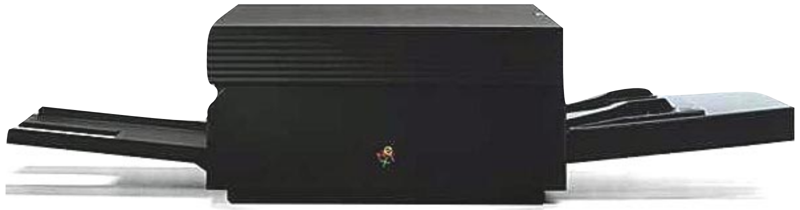
1996 — NeXT purchased by Apple
for next-generation Mac OS.

Operating system features:

- Mach microkernel from Carnegie Mellon University
- BSD Unix 4.3
- Display PostScript
- NeXTSTEP Object-Oriented Application Development Environment

Hardware features:

- Large bitmapped display; DSP sound; NeXT desktop bus; integrated laser printer



macOS X (ne MacOS X, OSX, Rhapsody)
2001-

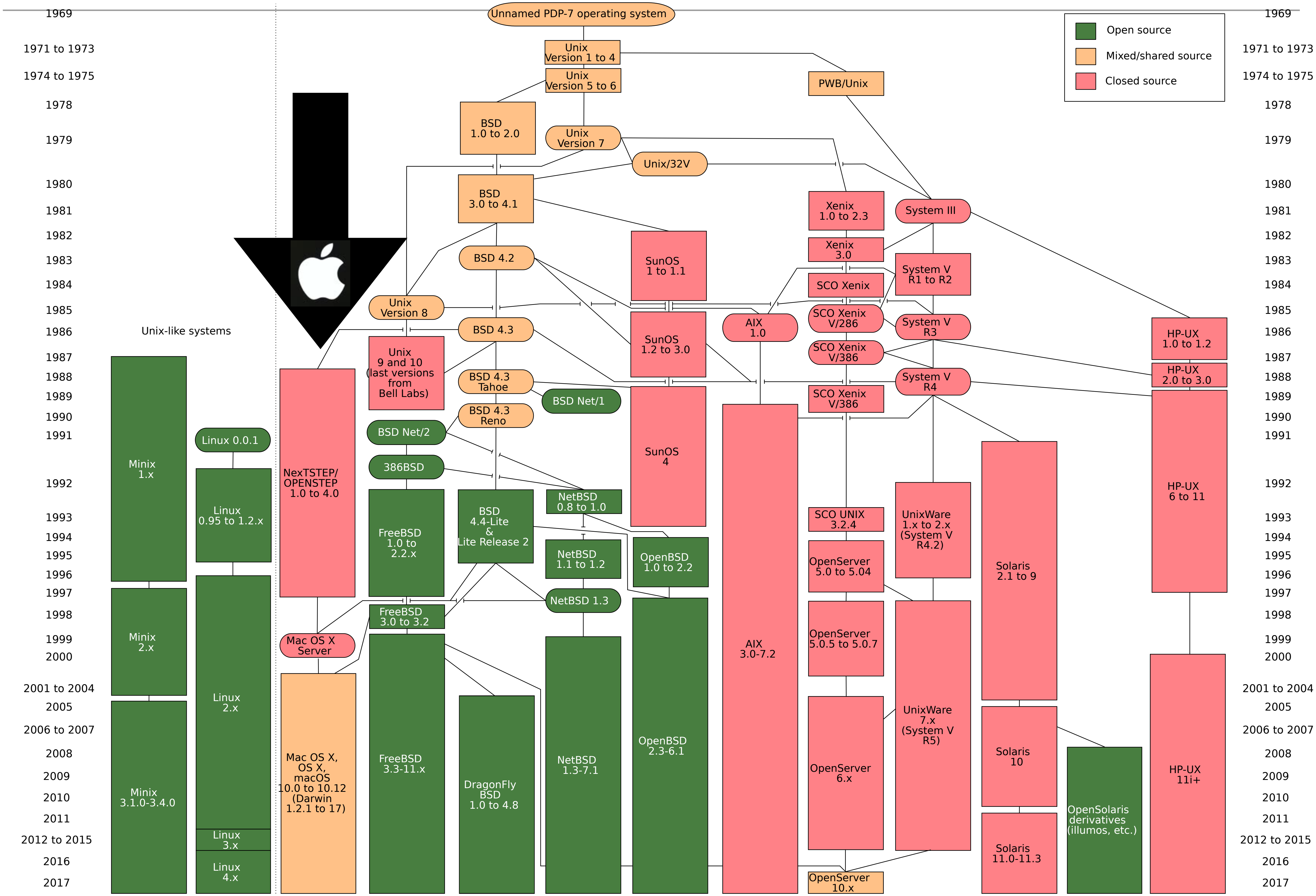
macOS X is the NeXTSTEP operating system, updated.

- Mach microkernel (memory management, processes, inter-process communication.)
- BSD kernel (monolithic kernel providing TCP/IP and many system services.)
  —*"XNU Kernel"*
- BSD utilities
- Quartz (ne DisplayPDF)
- OpenStep (ne NeXSTEP)
- APIs for legacy System 7/8/9 apps
- PowerPC and Intel (2005)

See also:

—*https://en.wikipedia.org/wiki/Star_Trek_project*

How all of this fits together



“Why Macs are Still Better Than PCs”
Advantages of Macs

Quickview

EMACS keybindings in all text fields

Migration Assistant

Consistent user interface

Startup options, including:

- Boot from any volume
- Target mode
- Recovery Mode

Hardware:

- Consistently high quality
- Excellent support policies (if you have AppleCare)

https://simson.net/page/Why_Macs_are_Still_Better_Than_PCs

"Why Macs are Still Better Than PCs"
Problems with Windows

NTFS is a lousy file system

- *—Poor performance*
- *—Locks open files and directories containing open files*
- *—Alternate data streams have no legitimate use.*

Legacy APIs make development difficult

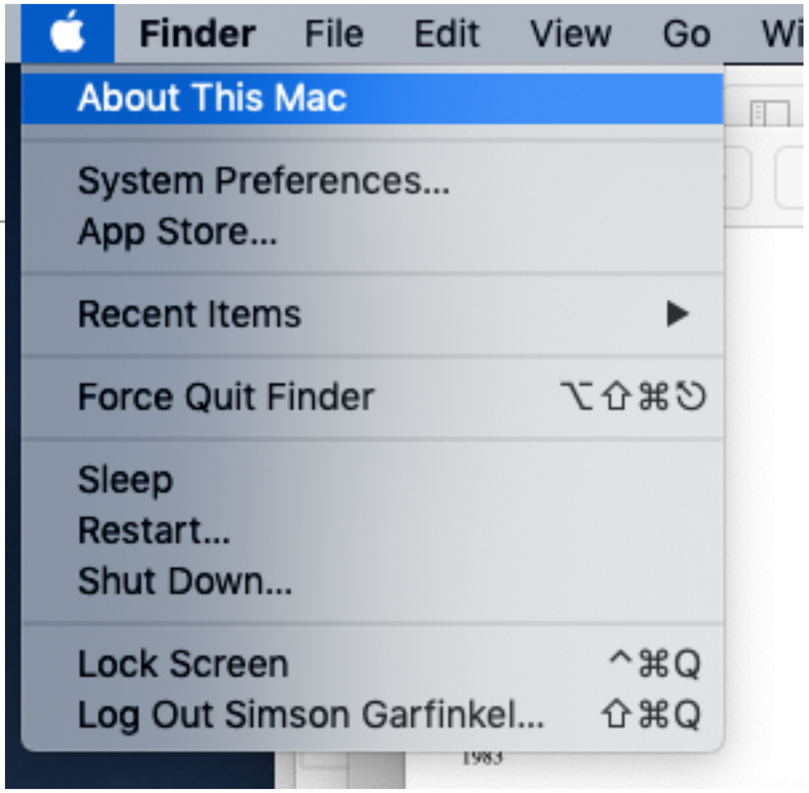
Process creation is really slow

Windows inter-process messaging is fundamentally flawed

… But it's a great platform for writing malware!

»access:denial //
t src=[erro
script src=[true]{?unk
function logged:#
function logged:#
function logged:#inp .[tru
{?unknown} m#4:80a?:/q.s statu
[true]local.config=(245.23.
name<l
[status?] code<[true]#status
malicious code logged

Today’s Mac Hardware Stack



CPU — Some Intel chip
Memory — Matched to the CPU
Storage:

- ATA/SATA/SCSI • USB • SD Card

Multi-purposes buses:

- USB • Thunderbolt • PCI • FireWire • Fibre Channel

Display

- Direct attached & bus-attrached

Network

- Ethernet (wired & wireless); WLAN
- Bluetooth

Other I/O devices (typically bus-attached)

- Audio
- Camera

Mac Boot Sequence

—*https://eclecticlight.co/2018/08/25/booting-the-mac-visual-summary/*

1. Mac Power On Self Test (POST)

- If no RAM is found, a single tone repeated every 5 seconds
- If RAM is found but fails POST, three tones followed by 5 second pause, repeating
- Other stuff (see article at https://eclecticlight.co/2018/08/10/booting-the-mac-loading-boot-efi-and-secure-boot/)

2. Run firmware for all hardware chips, including SMC, T2, NVRAM, audio, USB, storage, Wi-Fi, Ethernet, etc

3. Process any special keys that are down

4. If firmware password is set, get and validate password (if required)

5. Enumerate storage devices and boot device specified in NVRAM.

Boot the Mac: Special keys

Command (⌘)-R - macOS Recovery partition
Option (⌥) - Startup Manager (select startup disk or volume)
Shift (⇧) — Safe Mode
⌥⌘-R — macOS Recovery over Internet
⇧⌥⌘R — MacOS Recovery over Internet
⌥⌘PR — Reset NVRAM
C — Boot from CD/DVD
D — Apple Hardware Test or Apple Diagnostics
⌥D — Apple Hardware Test over Internet
N — Start from NetBoot server (not on T2-equipped computers)
⌘S — Single-user mode (macOS High Sierra or earlier)
T — Target mode. Make Mac external HD (Firewire or Thunderbolt)
⌘V — Verbose
X — Boot OSX (instead of Windows)
Eject (⏏) or F12 or mouse button or trackpad button — Eject Removable Media
Left⇧ — Prevent automatic login
Left⇧Control(^)⌥Power (⌽) — Reset SMC

Sources:

https://support.apple.com/en-us/HT201255 — Special keys on boot
https://support.apple.com/en-us/HT201236 — Special keys after startup
https://www.idownloadblog.com/2016/05/23/mac-startup-key-combinations/ — More combinations

Mac Forensic Opportunities

ISSA
Information Systems Security Association
National Capital Chapter

Existing forensic techniques that work on the Mac

Mac-specific collection opportunities

What are our options?

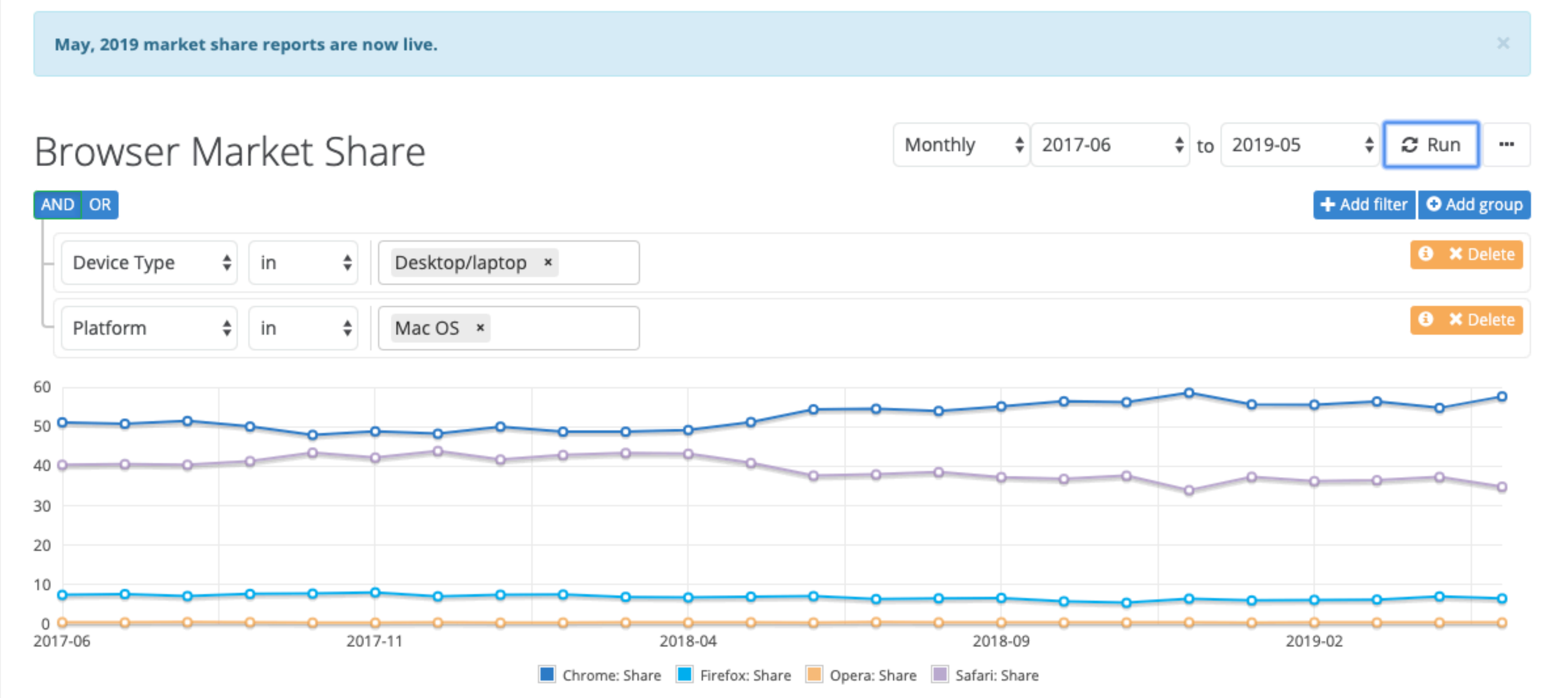
https://thisdata.com/blog/the-6-funniest-and-most-overused-images-of-hackers/

Existing forensic techniques that work on the Mac

ISSA
Information Systems Security Association
National Capital Chapter

Mac web browsers are similar to Windows web browsers

Safari — Default browser on MacOS X & iOS devices



May, 2019 market share reports are now live.

Browser Market Share

Monthly 2017-06 to 2019-05

Device Type in Desktop/laptop

Platform in Mac OS

Today: Chrome 60%, Safari 35%, Firefox 5%

Search at https://netmarketshare.com/browser-market-share.aspx

Safari: per-user databases of web activity. Complete and easy-to-parse (SQLite3)

Safari — $HOME/Library/Safari

AutoFillCorrection
Bookmarks.plist
CloudAutoFillCorrection
ClouTabs
Databases for remote web sites
History
PerSitePreferences
Recently Closed Tabs
User Notification Permissions

Mac-specific collection opportunities

dtrace — allows complex monitoring of most kernel APIs

- You must disable System Integrity Protection for most uses.
- Better for offline analysis than incident response.

fseventsd — list of file system "events" on each volume

- Metadata record of files created, deleted & modified
- Compact data structure, can go back months or years
- Similar to Windows and EXT4 journals, but much more complete
- Largely ignored by current forensic tools
  - —*BlackLight only parses when an option is selected*
  - —*Someone has written an Autopsy module; not obviously part of main release*

Keychain — A single encrypted database with:

- Passwords: websites, 802.11, encrypted volumes,
- Client-side certificates for end-to-end encryption
- Secure Notes

Persistance is similar to other Unix/Linux systems

System Boot:

EFI Boot ROM
EFI booter
XNU KernelCache
launchd (init in old Unix)

Launchd

/Library/LaunchAgents—Per-user agents installed by the admin
/Library/LaunchDaemons—System-wide daemons installed by the admin
/System/Library/LaunchAgents—Per-user agents provided by Apple
/System/Library/LaunchDaemons—System-wide daemons provided by Apple

—Agents — loaded upon user login
—daemons — loaded at system startup

```
Note: “plists” are used for more than launchd.
```

Tools for launching

launchctl — for controlling launchd

```
launchctl list

launchctl load -F plist
```

crontab — legacy cron control

```
crontab -l
crontab -e
crontab -u userid
```

persistence via kext

```
/System/Library/Extensions – OSX
/Library/Extensions – 3rd Party software
```

Other methods:

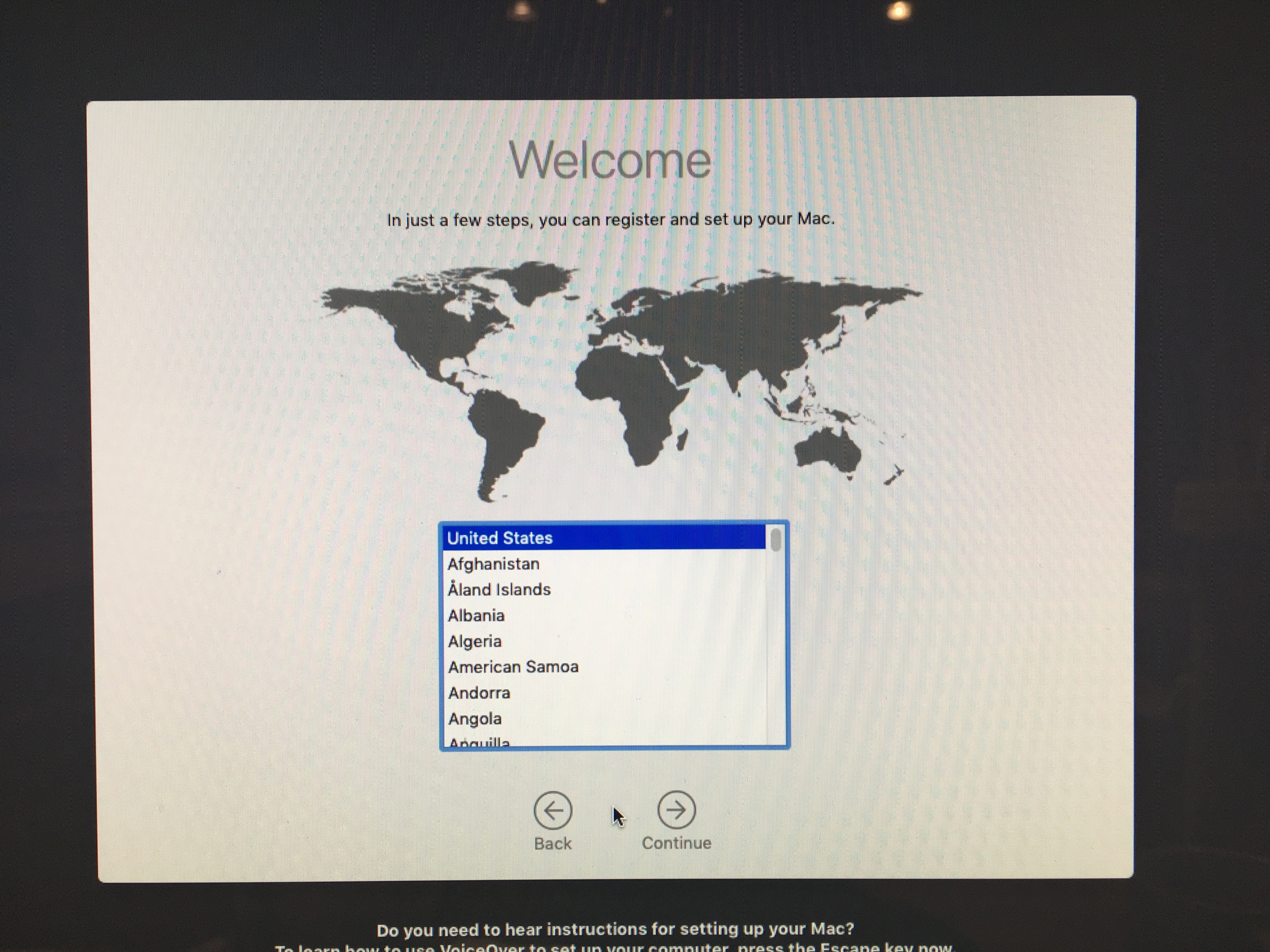
```
/Library/StartupItems/         /Library/PrefencePanes
/System/Library/StartupItems /System/Library/PreferencePanes
/etc/rc.common               ~/Library/PreferencePanes
```

Apple's push for integration creates forensic opportunities

Information Systems Security Association
National Capital Chapter

Recently I got a new mac mini!

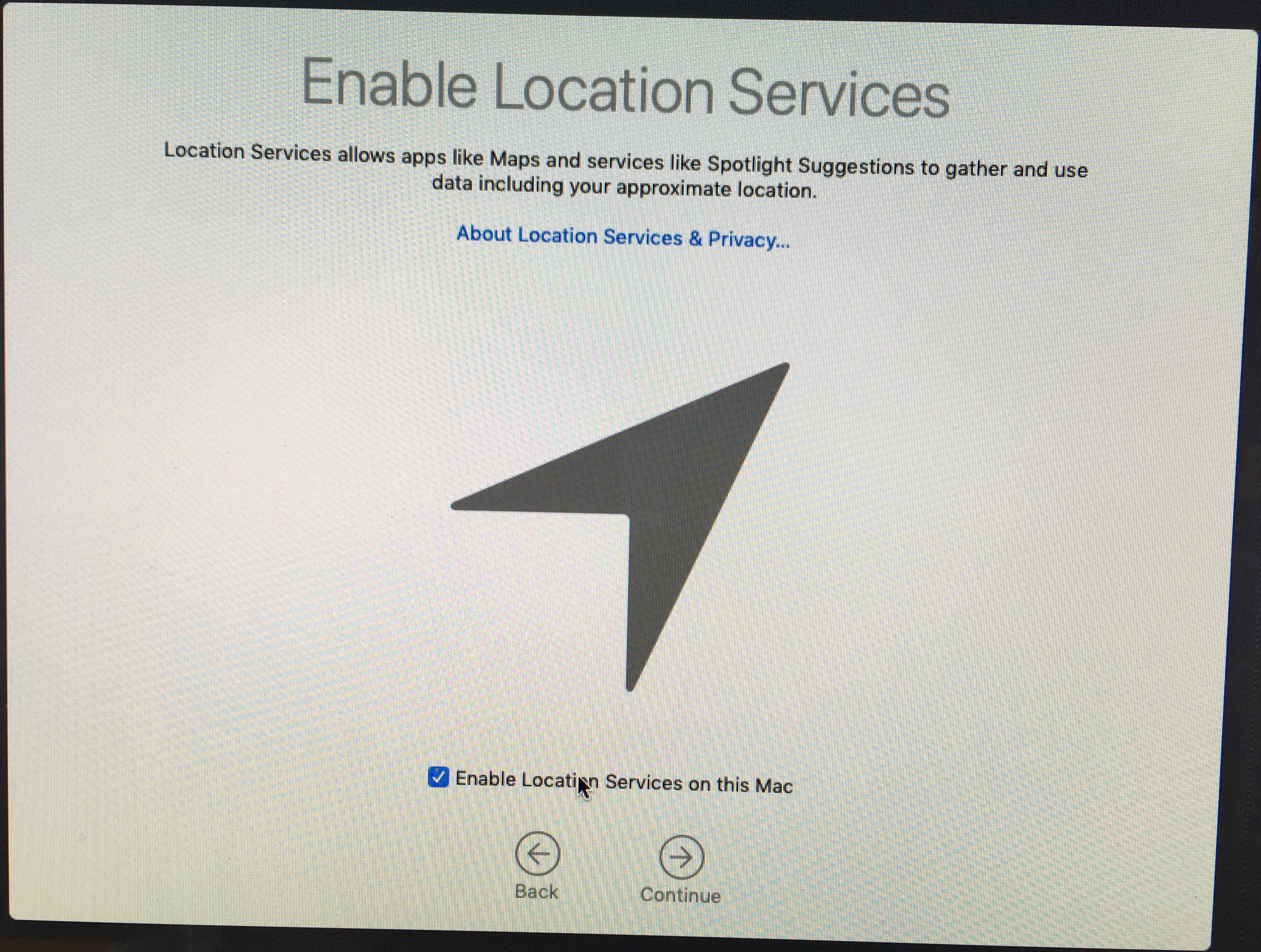
SIMSON GARFINKEL
Z0W2
1 Of 1
MAC MINI SG
C07XW0F6JYW0
W582853845
Order Date: DEC 17
mini



Welcome

In just a few steps, you can register and set up your Mac.

Do you need to hear instructions for setting up your Mac?
To learn how to use VoiceOver to set up your computer, press the Escape key now.

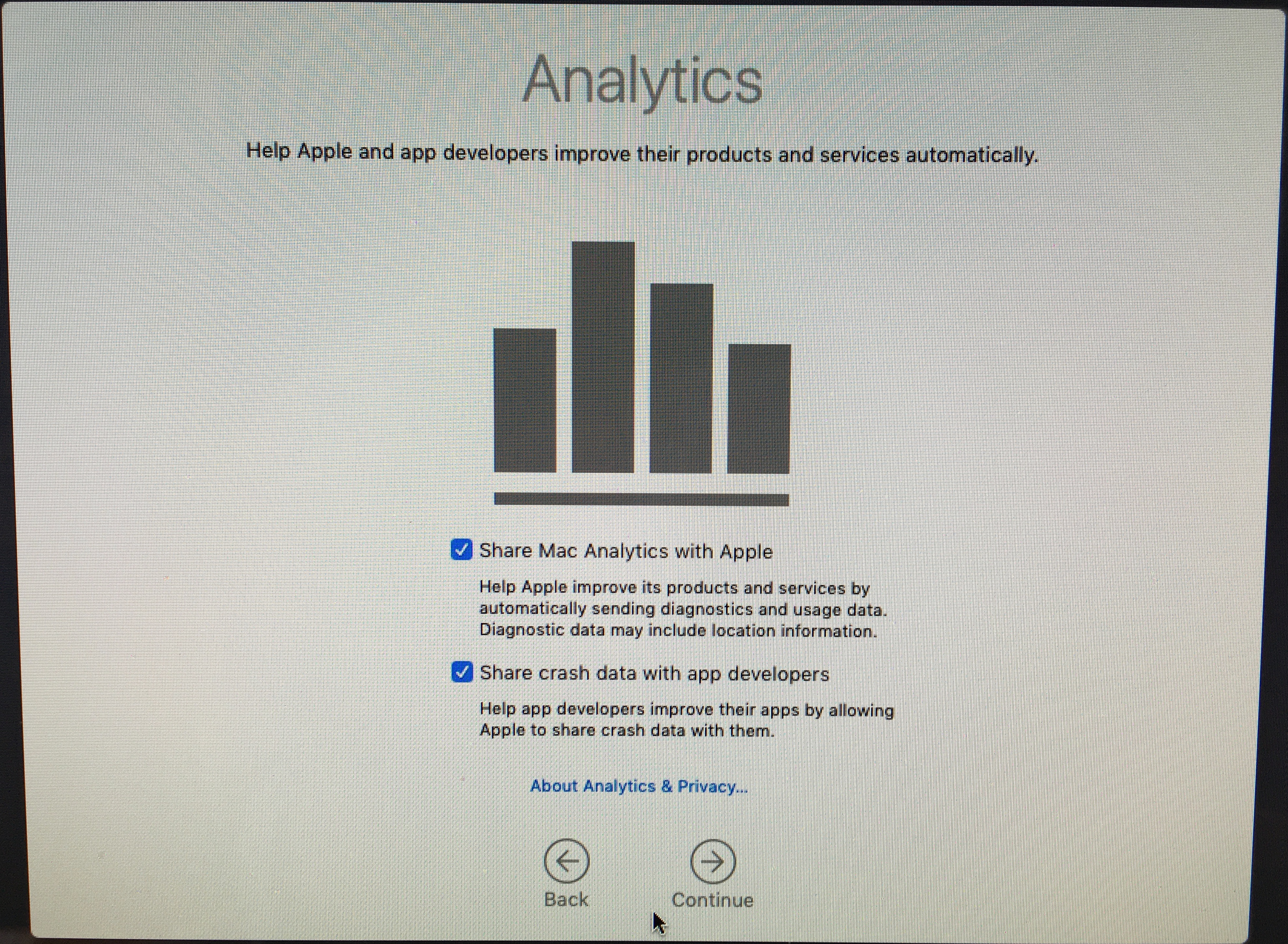


Enable Location Services

Location Services allows apps like Maps and services like Spotlight Suggestions to gather and use data including your approximate location.

About Location Services & Privacy...

☑ Enable Location Services on this Mac



Analytics

Help Apple and app developers improve their products and services automatically.

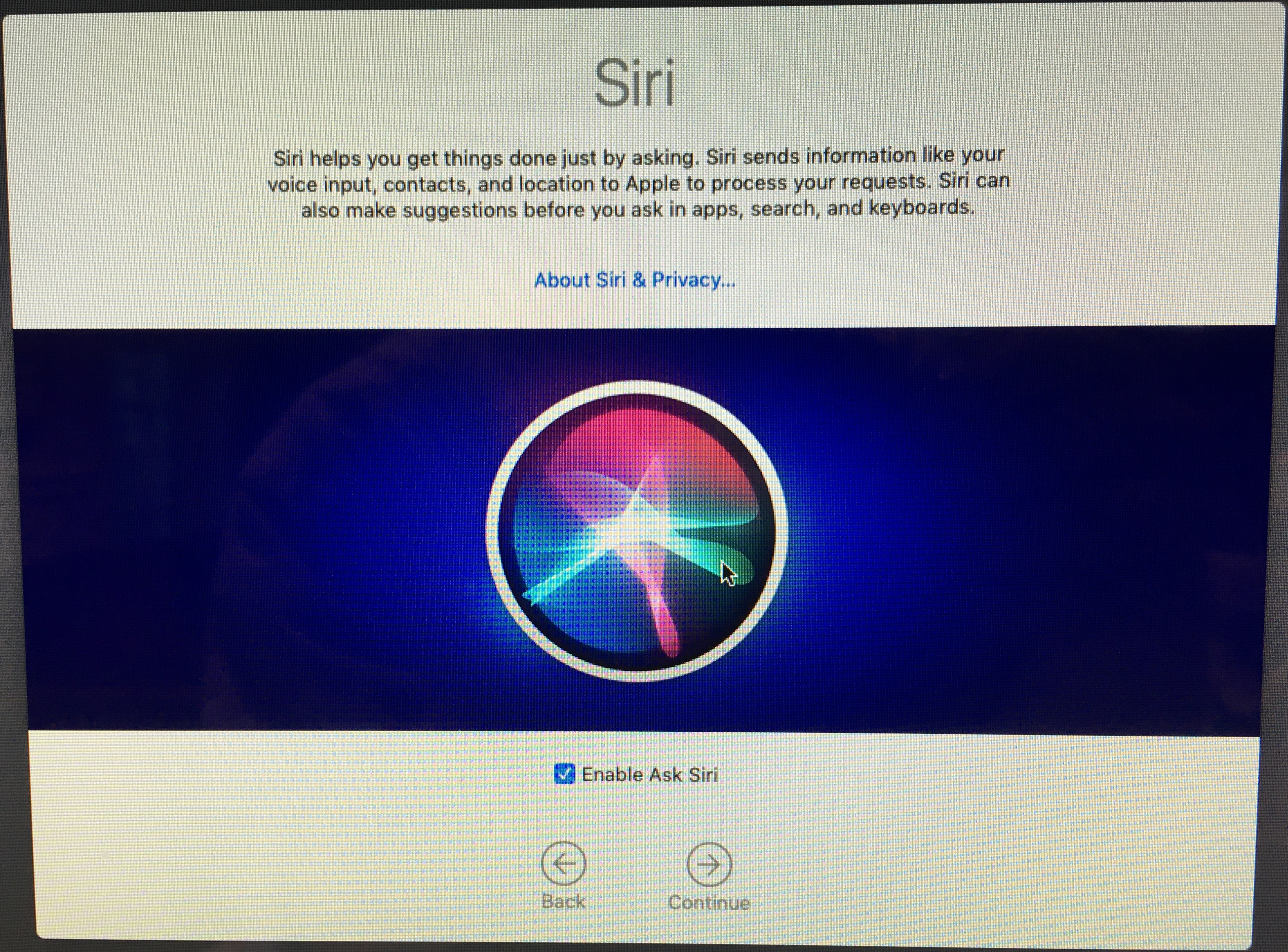
☑ Share Mac Analytics with Apple

Help Apple improve its products and services by automatically sending diagnostics and usage data. Diagnostic data may include location information.

☑ Share crash data with app developers

Help app developers improve their apps by allowing Apple to share crash data with them.

About Analytics & Privacy...

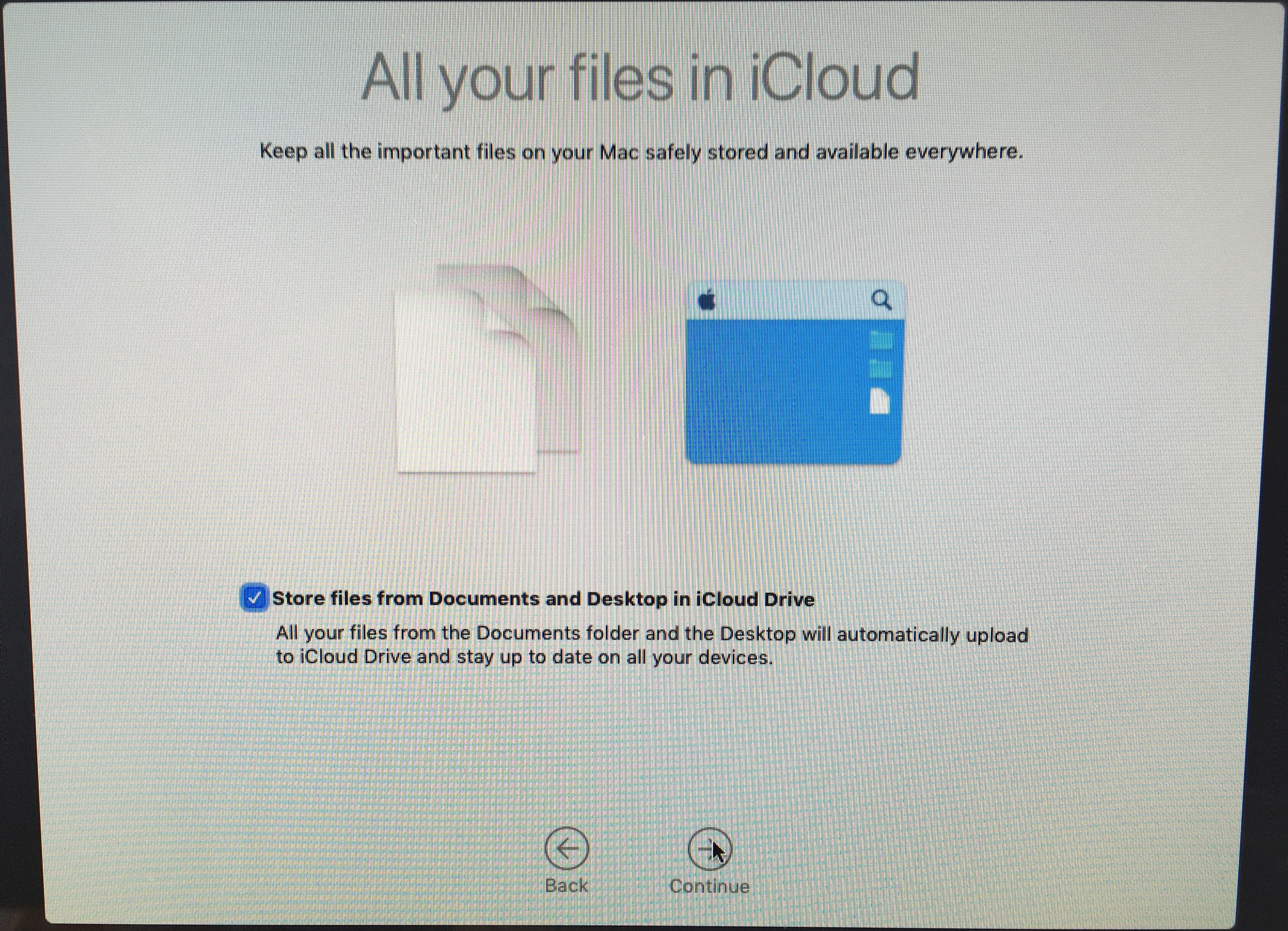


Siri

Siri helps you get things done just by asking. Siri sends information like your voice input, contacts, and location to Apple to process your requests. Siri can also make suggestions before you ask in apps, search, and keyboards.

About Siri & Privacy...

☑ Enable Ask Siri

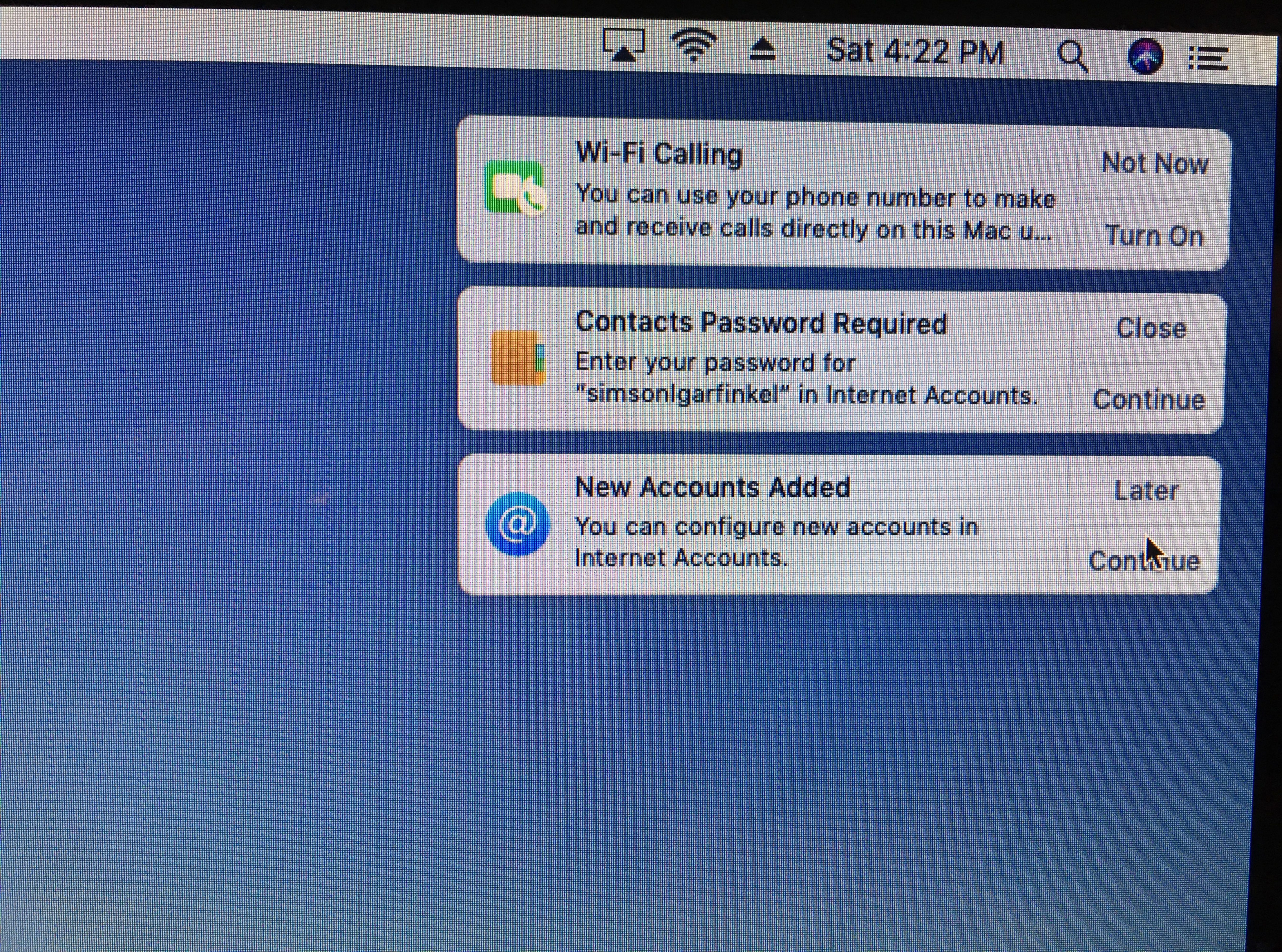


All your files in iCloud

Keep all the important files on your Mac safely stored and available everywhere.

☑ **Store files from Documents and Desktop in iCloud Drive**

All your files from the Documents folder and the Desktop will automatically upload to iCloud Drive and stay up to date on all your devices.



Wi-Fi Calling

You can use your phone number to make and receive calls directly on this Mac u...

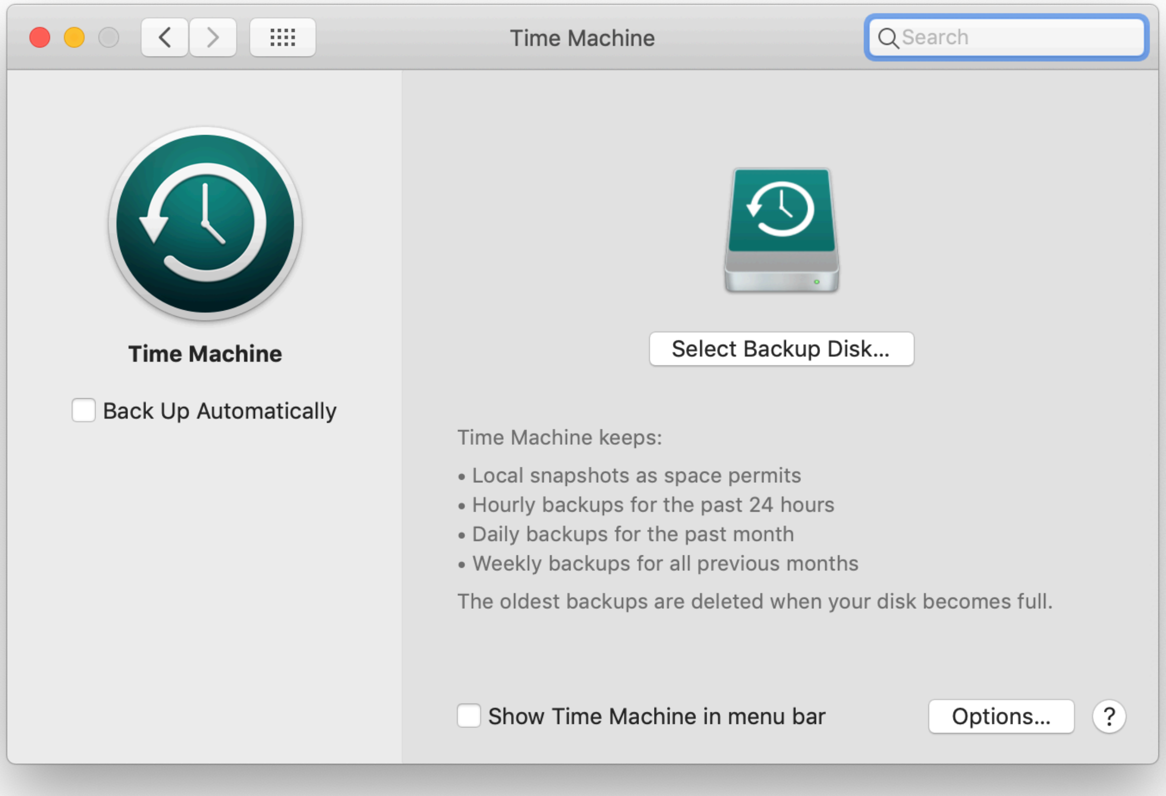
Contacts Password Required

Enter your password for "simsonlgarfinkel" in Internet Accounts.

New Accounts Added

You can configure new accounts in Internet Accounts.

Time Machine creates a forensic archive!



☐ Back Up Automatically

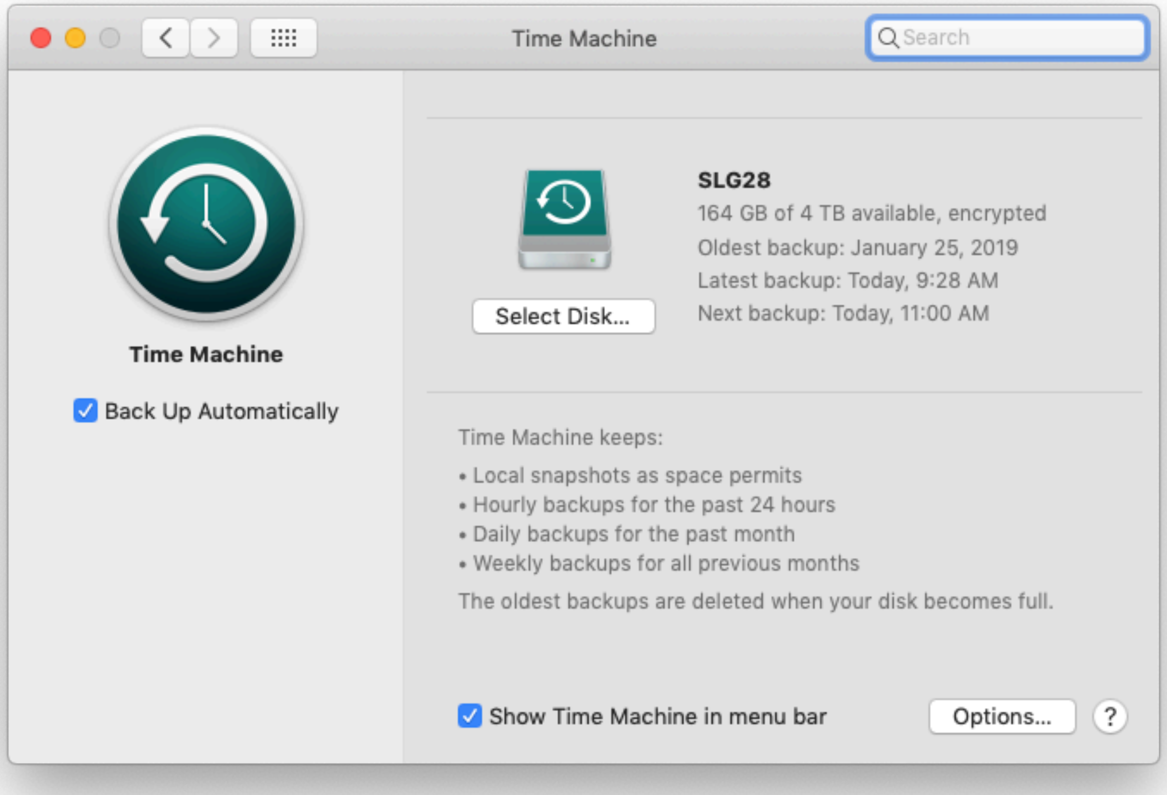
Time Machine keeps:

- Local snapshots as space permits
- Hourly backups for the past 24 hours
- Daily backups for the past month
- Weekly backups for all previous months

The oldest backups are deleted when your disk becomes full.

☐ Show Time Machine in menu bar

No Time Machine



☑ Back Up Automatically

SLG28
164 GB of 4 TB available, encrypted
Oldest backup: January 25, 2019
Latest backup: Today, 9:28 AM
Next backup: Today, 11:00 AM

Time Machine keeps:

- Local snapshots as space permits
- Hourly backups for the past 24 hours
- Daily backups for the past month
- Weekly backups for all previous months

The oldest backups are deleted when your disk becomes full.

☑ Show Time Machine in menu bar

Time Machine Activated

Check:

```
/Library/Preferences/com.apple.TimeMachine.plist
```

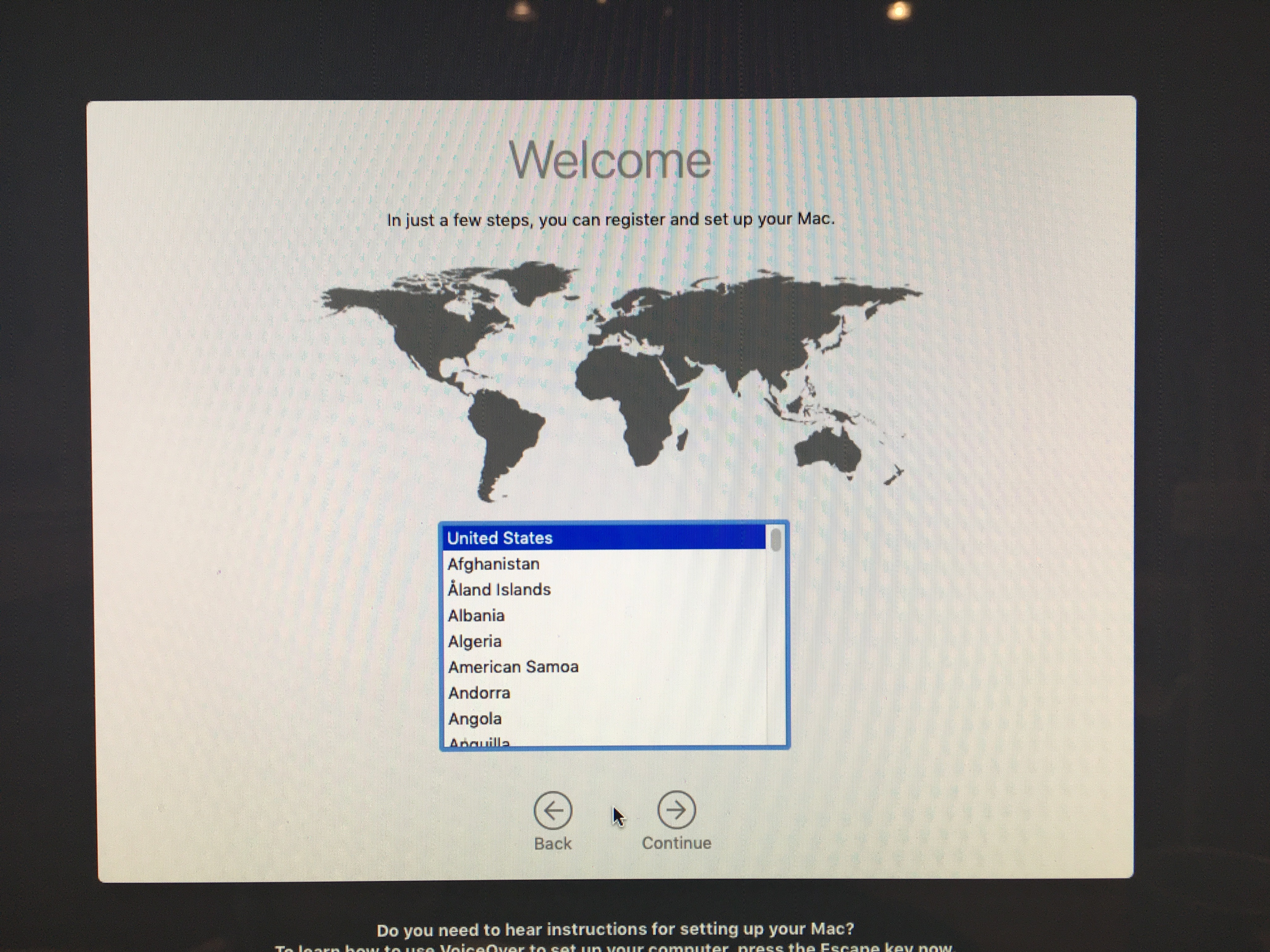
Backups don't need to be restored; they can be analyzed directly.

Mac Forensic Challenges

ISSA
Information Systems Security Association
National Capital Chapter

System Integrity Protection
Pervasive cryptography
Apple’s T2 chip
APFS
Logging

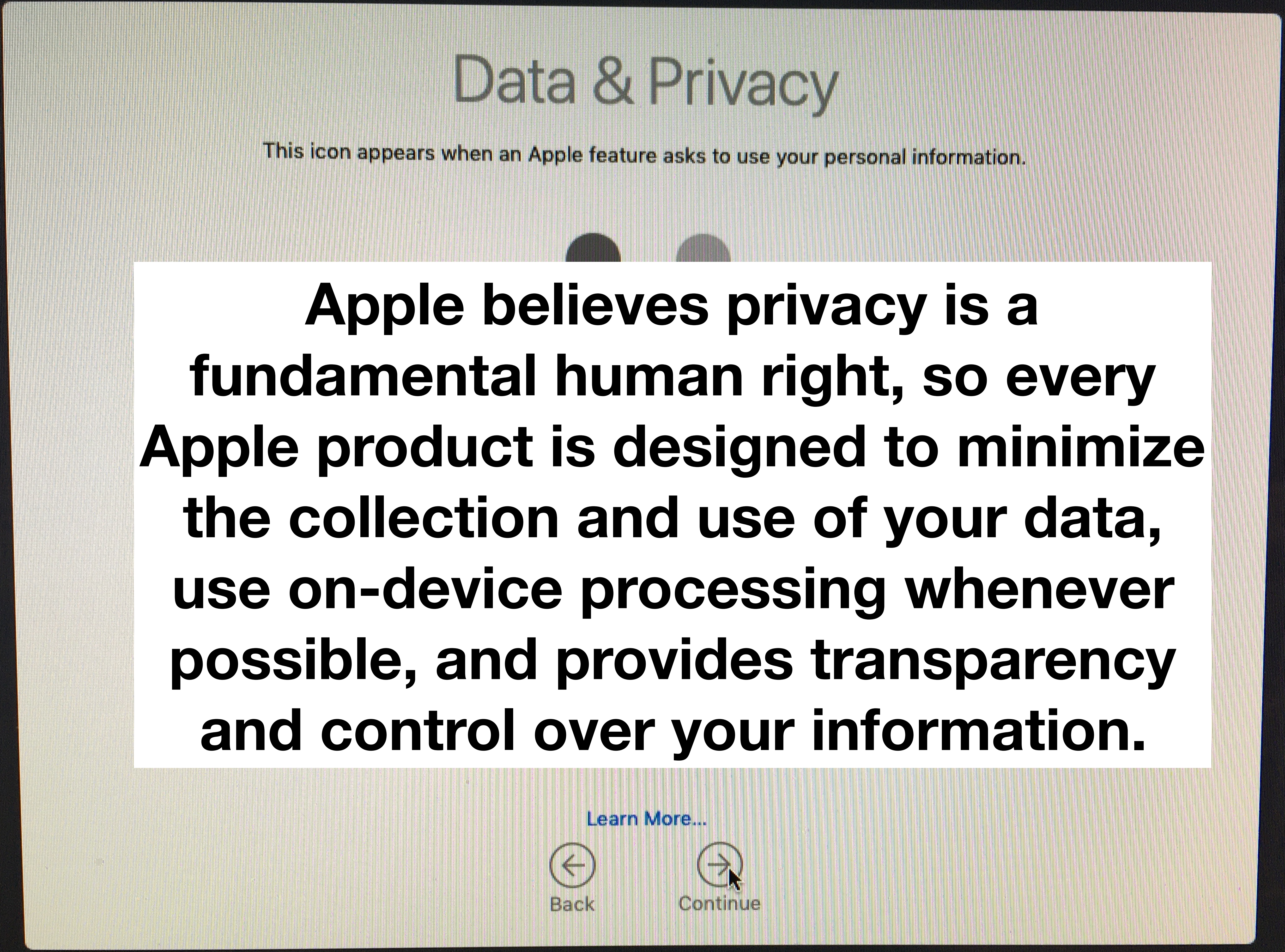
SIMSON GARFINKEL
Z0W2
1 Of 1
MAC MINI SG
C07XW0F6JYW0
W582853845
Order Date: DEC 17
mini



Welcome

In just a few steps, you can register and set up your Mac.

Do you need to hear instructions for setting up your Mac?
To learn how to use VoiceOver to set up your computer, press the Escape key now.

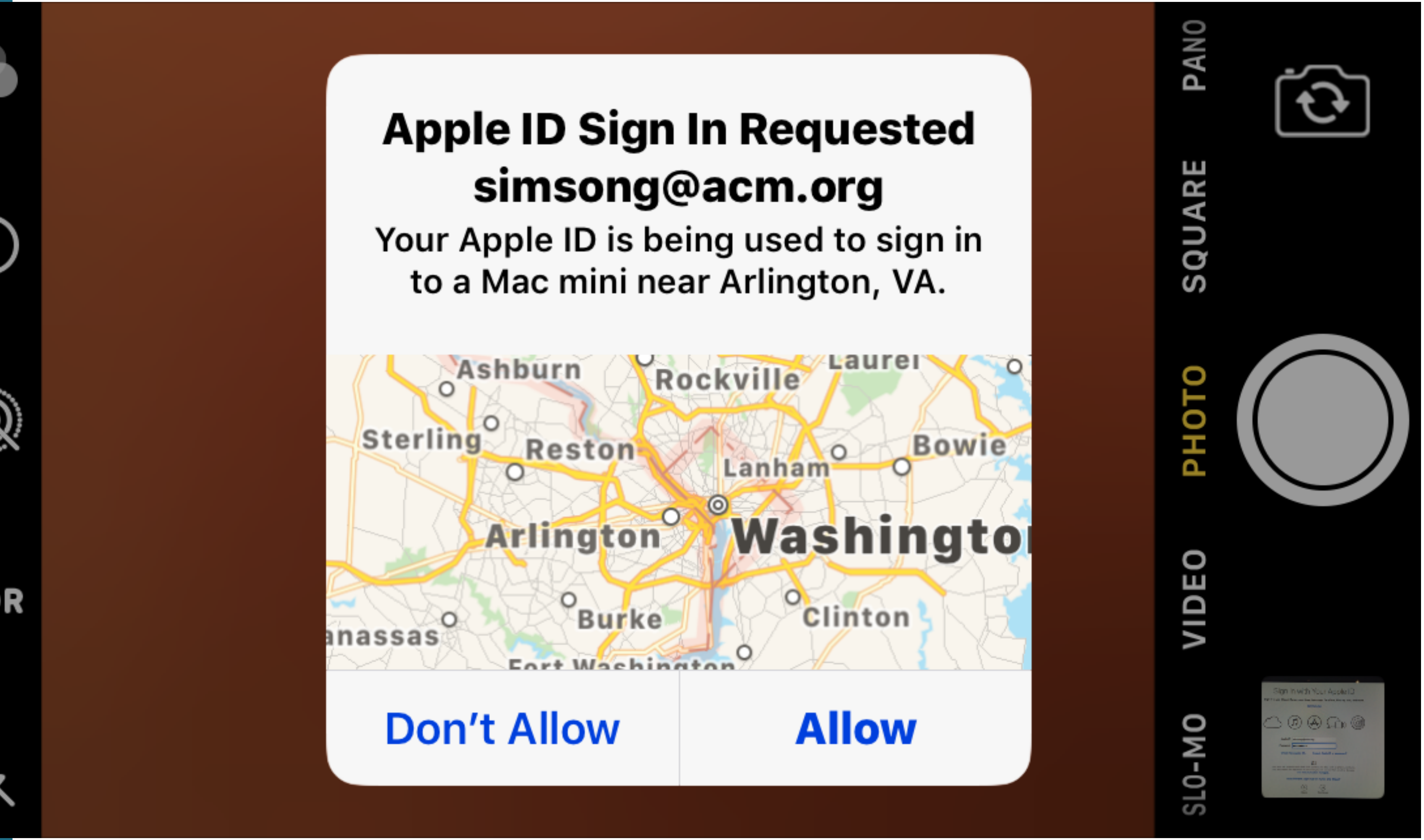


Data & Privacy

This icon appears when an Apple feature asks to use your personal information.

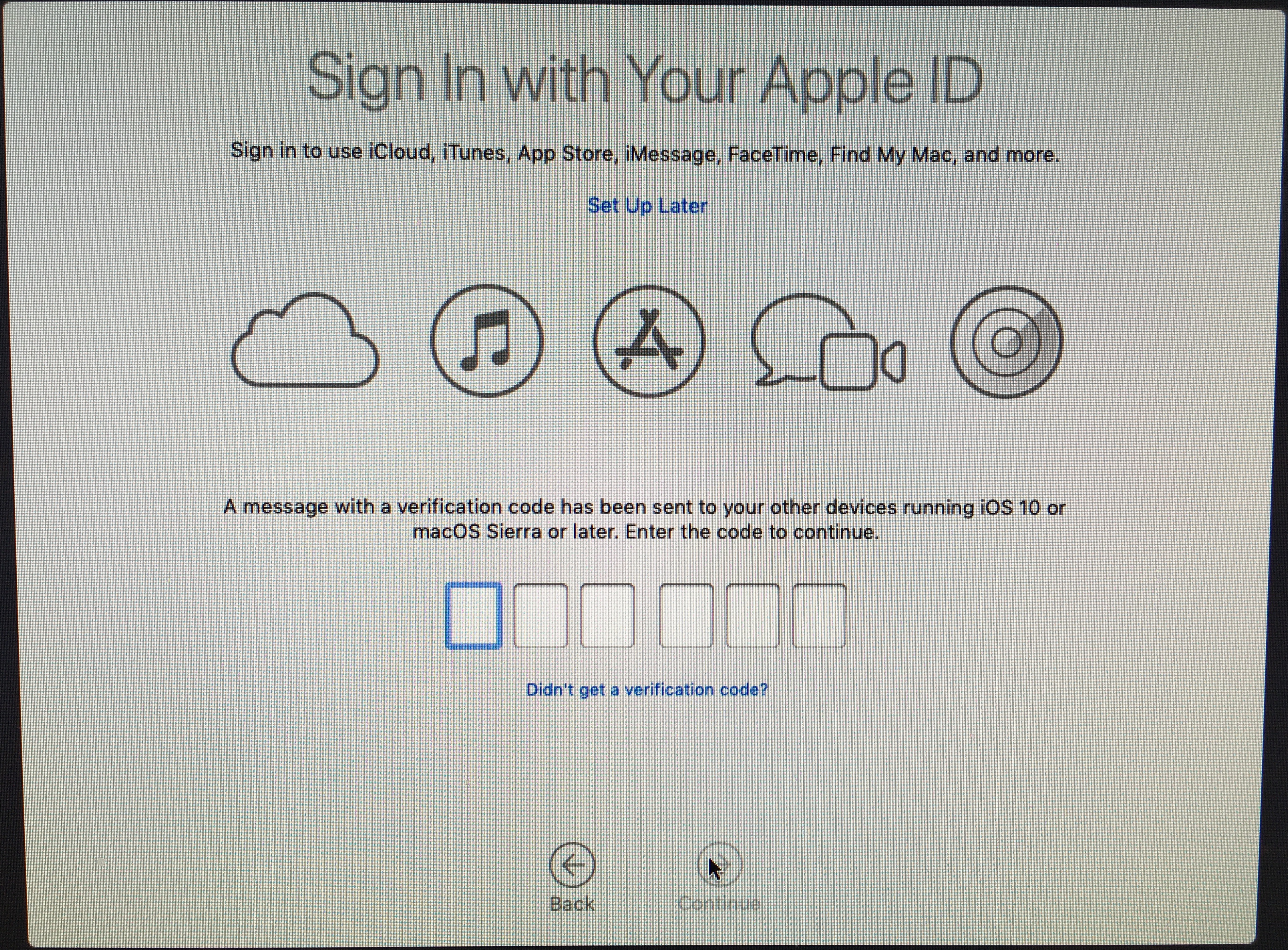
Apple believes privacy is a fundamental human right, so every Apple product is designed to minimize the collection and use of your data, use on-device processing whenever possible, and provides transparency and control over your information.

Integrated 2-factor



Apple ID Sign In Requested
simsong@acm.org

Your Apple ID is being used to sign in to a Mac mini near Arlington, VA.



Sign In with Your Apple ID

Sign in to use iCloud, iTunes, App Store, iMessage, FaceTime, Find My Mac, and more.

Set Up Later

A message with a verification code has been sent to your other devices running iOS 10 or macOS Sierra or later. Enter the code to continue.

□ □ □ □ □ □

Didn't get a verification code?

System Integrity Protection

DTrace is a powerful tool for monitoring a Mac

Developed by Sun Microsystems (now Oracle) for Solaris.

Operates by:

- **Compiling user-supplied code in dtrace language.**
- **Injecting code into the kernel**

History:

- **2004: Cantrill, Shapiro & Leventhal, "Dynamic Instrumentation of Production Systems," USENIX ATC**
  - *https://www.usenix.org/legacy/event/usenix04/tech/general/full_papers/cantrill/cantrill_html/*
- **2007: Apple ports DTrace to MacOS 10.5; adds Instruments API**
  - *Also adds P_LNOATTACH; prevents DTrace with System Integrity Protection*
- **2008: Ported to Linux**
- **2019: Microsoft releases for Windows 10 insider build 18342**

Using DTrace

DTrace requires root privileges.

Most scripts won’t work if System Integrity Protection is enabled.

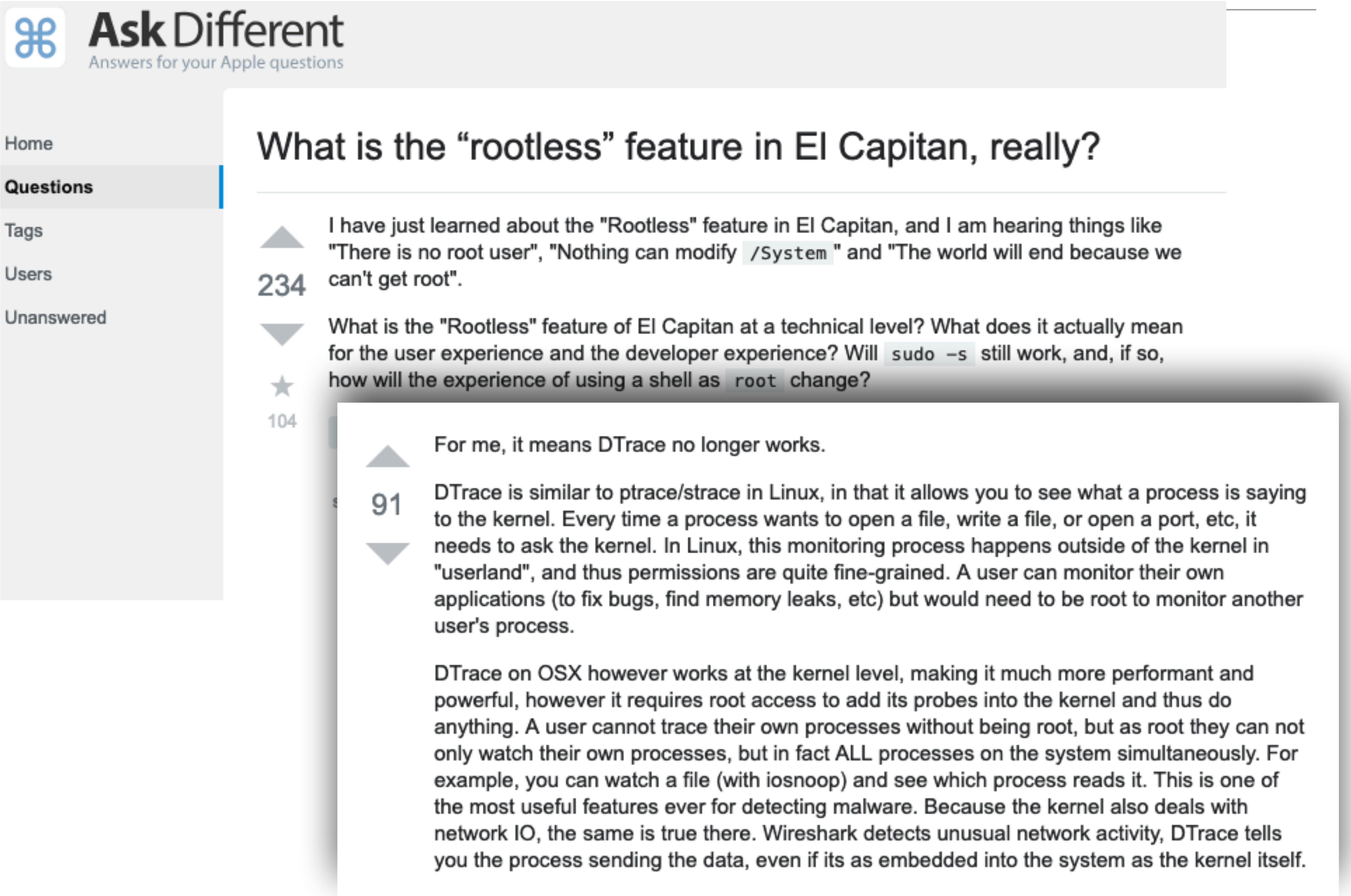
Enabling SIP requires reboot into single-user mode.

Here’s what happens if SIP is enabled:

```
$ sudo rwsnoop
dtrace: system integrity protection is on, some features will not be
available

  UID    PID CMD          D   BYTES FILE
dtrace: error on enabled probe ID 26 (ID 168: syscall::read:return):
invalid kernel access in action #1 at DIF offset 0
dtrace: error on enabled probe ID 28 (ID 468: syscall::pread:return):
invalid kernel access in action #1 at DIF offset 0
dtrace: error on enabled probe ID 28 (ID 468: syscall::pread:return):
invalid kernel access in action #1 at DIF offset 0
dtrace: error on enabled probe ID 28 (ID 468: syscall::pread:return):
invalid kernel access in action #1 at DIF offset 0
dtrace: error on enabled probe ID 29 (ID 469: syscall::pwrite:entry):
invalid kernel access in action #1 at DIF offset 0
dtrace: error on enabled probe ID 29 (ID 469: syscall::pwrite:entry):
invalid kernel access in action #1 at DIF offset 0
dtrace: error on enabled probe ID 29 (ID 469: syscall::pwrite:entry):
invalid kernel access in action #1 at DIF offset 0
```

SIP is sometimes called “rootless” because the root user no longer has full access.



What is the “rootless” feature in El Capitan, really?

I have just learned about the "Rootless" feature in El Capitan, and I am hearing things like "There is no root user", "Nothing can modify `/System`" and "The world will end because we can't get root".

What is the "Rootless" feature of El Capitan at a technical level? What does it actually mean for the user experience and the developer experience? Will `sudo -s` still work, and, if so, how will the experience of using a shell as `root` change?

For me, it means DTrace no longer works.

DTrace is similar to ptrace/strace in Linux, in that it allows you to see what a process is saying to the kernel. Every time a process wants to open a file, write a file, or open a port, etc, it needs to ask the kernel. In Linux, this monitoring process happens outside of the kernel in "userland", and thus permissions are quite fine-grained. A user can monitor their own applications (to fix bugs, find memory leaks, etc) but would need to be root to monitor another user's process.

DTrace on OSX however works at the kernel level, making it much more performant and powerful, however it requires root access to add its probes into the kernel and thus do anything. A user cannot trace their own processes without being root, but as root they can not only watch their own processes, but in fact ALL processes on the system simultaneously. For example, you can watch a file (with iosnoop) and see which process reads it. This is one of the most useful features ever for detecting malware. Because the kernel also deals with network IO, the same is true there. Wireshark detects unusual network activity, DTrace tells you the process sending the data, even if its as embedded into the system as the kernel itself.

T2 Chip
&
Pervasive Encryption

ISSA
Information Systems Security Association
National Capital Chapter

Security

Your data is safe

The Apple T2 Security Chip gives your Mac mini a higher-than-ever level of security. Your data is encrypted with keys tied specifically to your computer, and Secure Boot ensures that only legitimate macOS software loads at startup.

Learn more about keeping your data safe

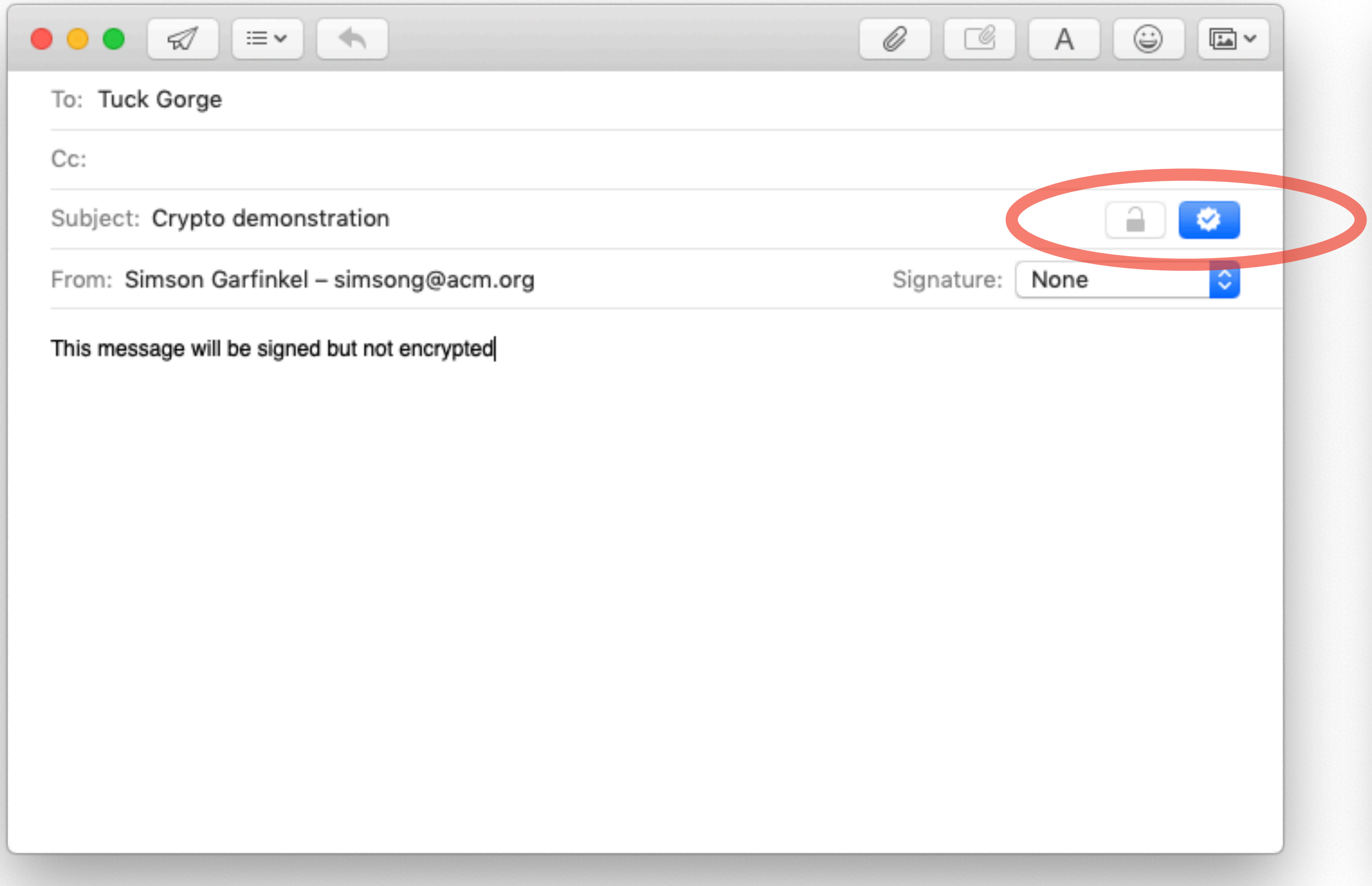
Apple Mail as full support for mail encryption

S/MIME — Add a public/private S/MIME key and Mail.app will:

- Offer to sign outgoing mail.
- Offer to encrypt if it has the public key for the recipient

Transparent support for PGP is available with plug-in

Encryption implementation is comprehensive



To: Tuck Gorge

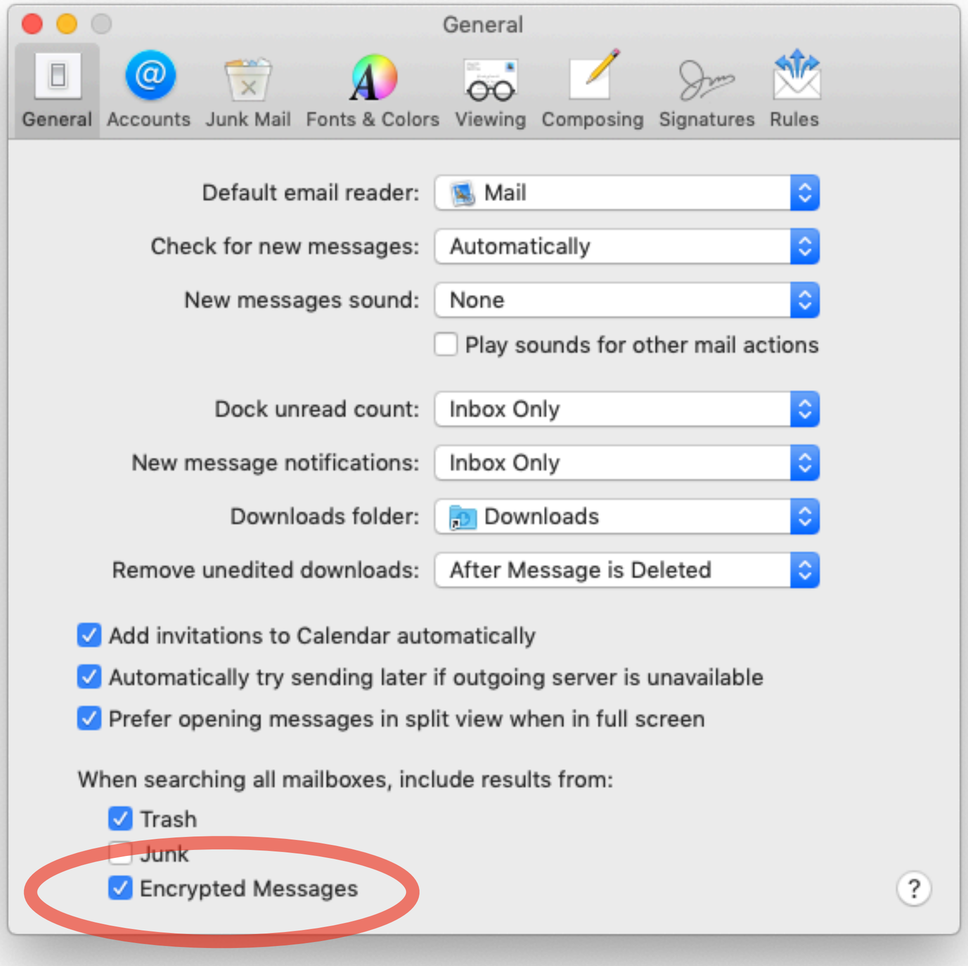
Cc:

Subject: Crypto demonstration

From: Simson Garfinkel – simsong@acm.org

Signature: None

This message will be signed but not encrypted



Default email reader: Mail

Check for new messages: Automatically

New messages sound: None

□ Play sounds for other mail actions

Dock unread count: Inbox Only

New message notifications: Inbox Only

Downloads folder: Downloads

Remove unedited downloads: After Message is Deleted

☑ Add invitations to Calendar automatically

☑ Automatically try sending later if outgoing server is unavailable

☑ Prefer opening messages in split view when in full screen

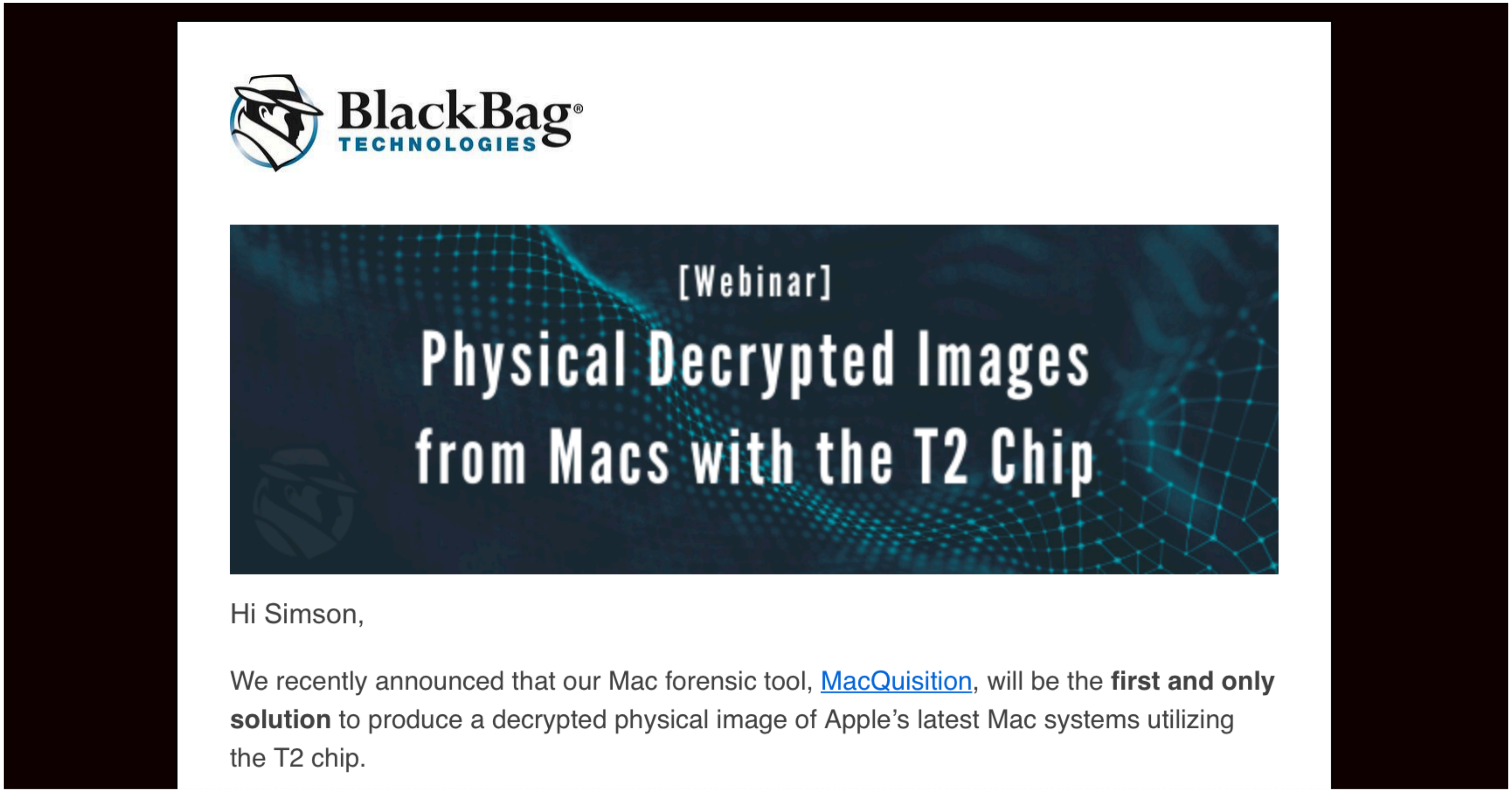
When searching all mailboxes, include results from:

☑ Trash

□ Junk

☑ Encrypted Messages

BlackBag had a great webinar on T2 and Physical Images



[Webinar]

Physical Decrypted Images from Macs with the T2 Chip

Hi Simson,

We recently announced that our Mac forensic tool, MacQuisition, will be the **first and only solution** to produce a decrypted physical image of Apple's latest Mac systems utilizing the T2 chip.

APFS

APFS — Apple File System

An advanced file system that supports:

- Files and volumes from 1–2^{63} bytes
- 64-bit file IDs
- 1 nanosecond time stamp granularity
- Cop-on-write
- Native encryption with per-file encryption keys
- Transparent support for SSD flash (erases after delete)

APFS — Where you find it

Internal drives:

- All new Apple devices
- All *OS devices running current operating systems were upgraded.

External drives:

- External drives were not automatically upgraded by macOS!

- If they were created before September 25, 2017:
  —*Probably HFS+*
  —*Possibly legacy FileVault*

- If they were created after September 25, 2017:
  —*Either HFS+ or APFS*
  —*May rely on the T2 chip*

“Fusion drives”

Apple proprietary hybrid drive

- SSD (24GB-128GB)
- HD (1TB - 3TB)

Managed by the OS, not by firmware

- Appears as two drives
- CoreStorage turns it into a single drive

Reliability issues

- Both hardware and software

Unified Logging System

Information Systems Security Association
National Capital Chapter

Logfiles — Mac OS X El Capitan v10.11 and earlier

Apple used traditional Unix logging.

/var/log — directory where logs were stored
syslogd — System logging utility
newsyslog — log rotation
/etc/newsyslog.d/ — log rotation configuration

“man newsyslog”

```
HISTORY
The newsyslog utility originated from NetBSD and first
appeared in FreeBSD 2.2.

AUTHORS
Theodore Ts'o, MIT Project Athena
Copyright 1987, Massachusetts Institute of Technology

BUGS
Does not yet automatically read the logs to find
security breaches.
```

Apple Unified Logging

2014 — Apple introduced Activity Tracing, Faults & Errors

Single logging mechanism for user & kernel mode

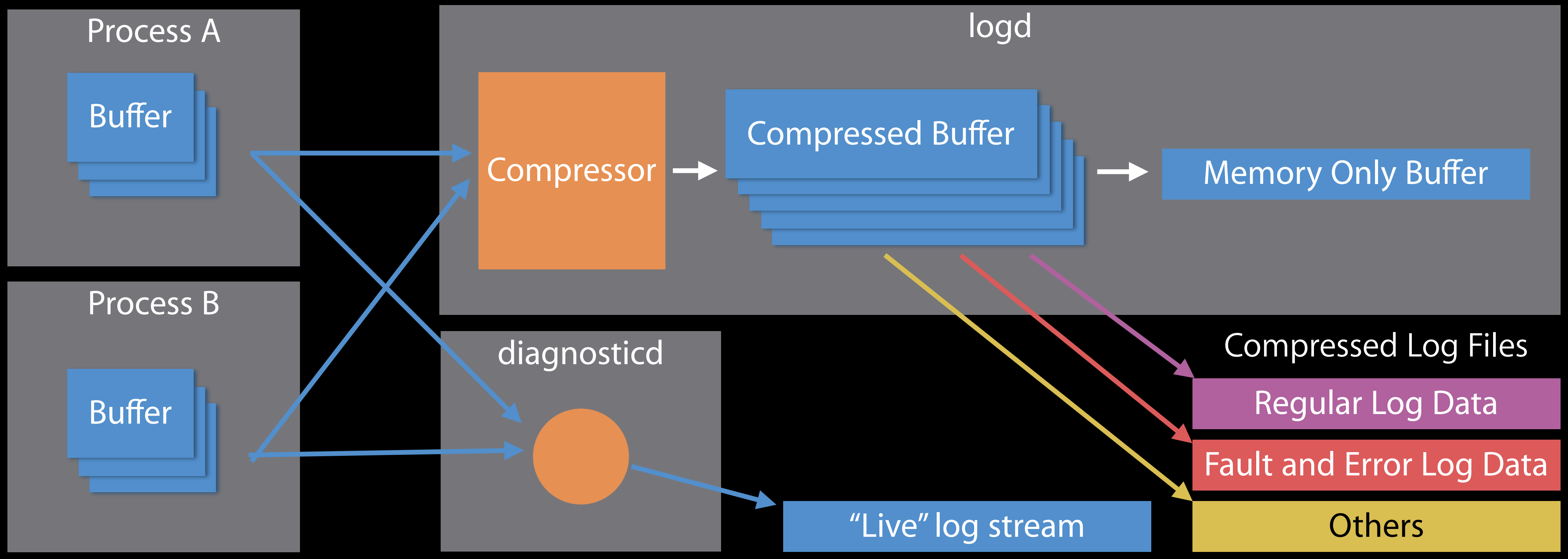
Efficient (better than a text file)

- *—Compresses data*
- *—Defers work and data collection*
- *—Manages log message lifecycle*
- *—Privacy designed into system — log messages not to contain PII*

Major improvements in logging:

Categorization and filtering designed into collection & display

Architecture

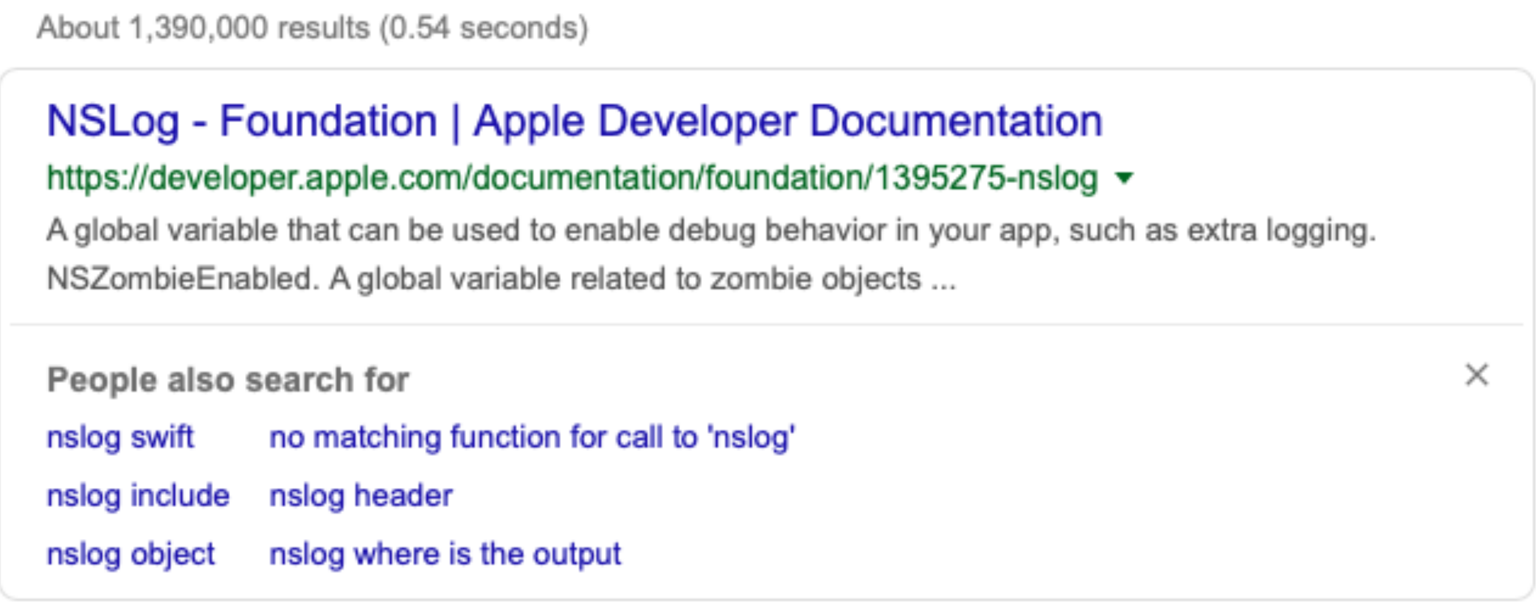


Profile can change routing and rules for given applications or subsystems

WWDC 2016 • Session 721 • iOS, macOS

Apple documentation

Google NSLog…



About 1,390,000 results (0.54 seconds)

NSLog - Foundation | Apple Developer Documentation

https://developer.apple.com/documentation/foundation/1395275-nslog ▾

A global variable that can be used to enable debug behavior in your app, such as extra logging. NSZombieEnabled. A global variable related to zombie objects ...

People also search for ×

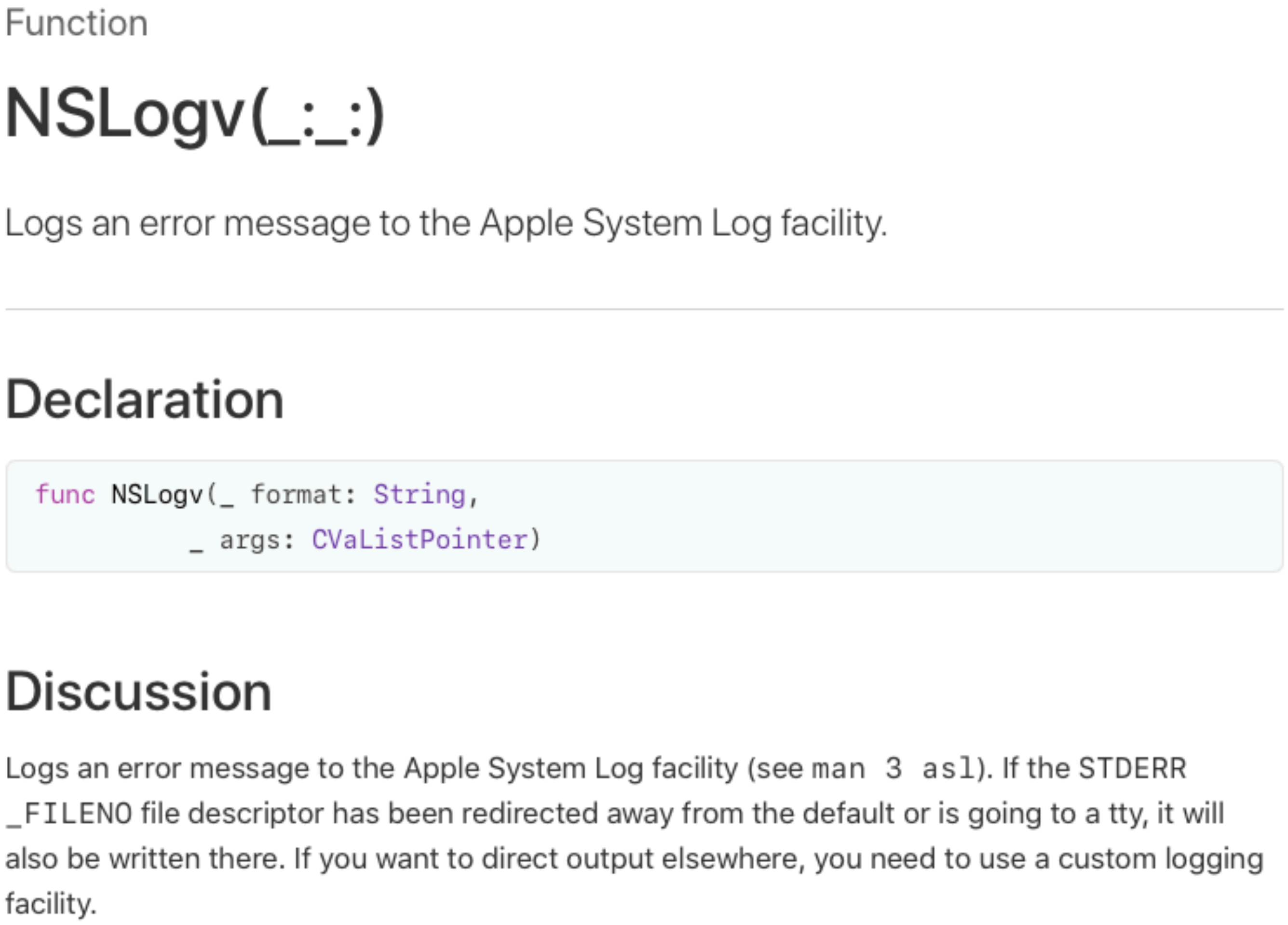
nslog swift	no matching function for call to 'nslog'
nslog include	nslog header
nslog object	nslog where is the output

→ NSLog()

- —*https://developer.apple.com/documentation/foundation/1395275-nslog*
- —*"Simply calls NSLogv passing it a variable number of arguments."*

→ NSLogv

- —*https://developer.apple.com/documentation/foundation/1395074-nslogv*



Function

NSLogv(_:_:)

Logs an error message to the Apple System Log facility.

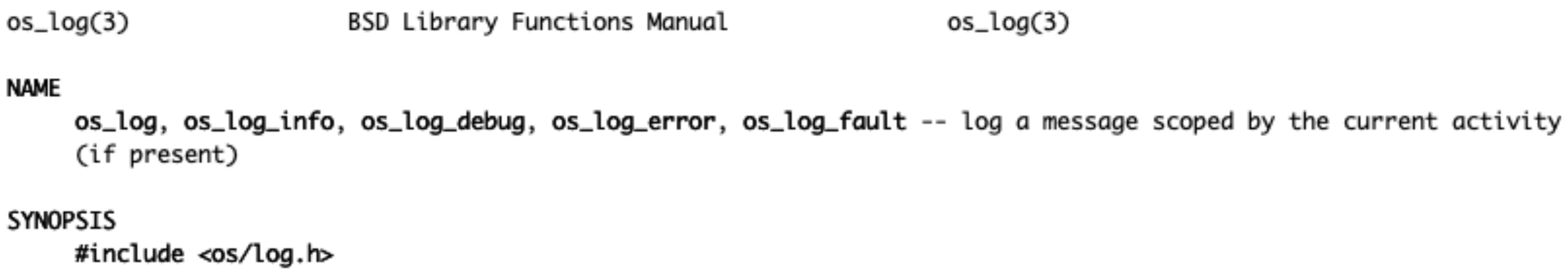
Declaration

```
func NSLogv(_ format: String,
          _ args: CVaListPointer)
```

Discussion

Logs an error message to the Apple System Log facility (see `man 3 asl`). If the `STDERR_FILENO` file descriptor has been redirected away from the default or is going to a tty, it will also be written there. If you want to direct output elsewhere, you need to use a custom logging facility.

“man 3 os_log”



```
os_log(3)               BSD Library Functions Manual               os_log(3)

NAME
     os_log, os_log_info, os_log_debug, os_log_error, os_log_fault -- log a message scoped by the current activity
     (if present)

SYNOPSIS
     #include <os/log.h>
```

DESCRIPTION

The unified logging system provides a single, efficient, high performance set of APIs for capturing log messages across all levels of the system. This unified system centralizes the storage of log data in memory and in a data store on disk. The system implements global settings that govern logging behavior and persistence, while at the same time providing fine-grained control during debugging via the log(1) command-line tool and through the use of custom logging configuration profiles. Log messages are viewed using the Console app in /Applications/Utilities/ and the log(1) command-line tool. Logging and activity tracing are integrated to make problem diagnosis easier. If activity tracing is used while logging, related messages are automatically correlated.

log(1) command line tool

“man 1 log”

```
log(1)      BSD General Commands Manual
log(1)
```

NAME

log -- Access system wide log messages created by os_log, os_trace and other logging systems.

SYNOPSIS

log [command [options]]

log help [command]

log collect [--output path] [--start date/time] [--size num [k|m]] [--last num [m|h|d]]

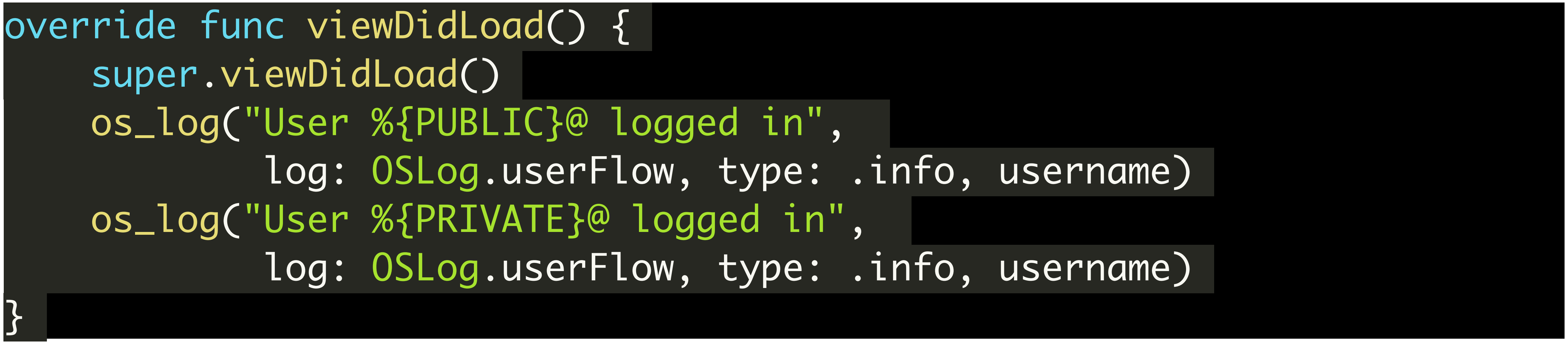
log config [--reset | --status] [--mode mode(s)] [--subsystem name [--category name]] [--process pid]

log erase [--all] [--ttl]

syslog -f reads the binary format

```
# ls -l
total 2208
-rw-------@ 1 root  171315 Apr  4 00:39 2019.04.03.G80.asl
-rw-------@ 1 root  210472 Apr  5 00:58 2019.04.04.G80.asl
-rw-------@ 1 root  392187 Apr  6 00:59 2019.04.05.G80.asl
…
# syslog -f 2019.04.03.G80.asl | head
NOTE:  Most system logs have moved to a new logging system.  See
log(1) for more information.
Apr  3 01:01:09 newdance com.apple.xpc.launchd[1]
(com.apple.imfoundation.IMRemoteURLConnectionAgent) <Warning>:
Unknown key for integer: _DirtyJetsamMemoryLimit
Apr  3 01:02:16 newdance syslogd[56] <Notice>: ASL Sender
Statistics
Apr  3 01:02:53 newdance com.apple.xpc.launchd[1]
(com.apple.imfoundation.IMRemoteURLConnectionAgent) <Warning>:
Unknown key for integer: _DirtyJetsamMemoryLimit
--- last message repeated 9 times ---
Apr  3 01:12:18 newdance syslogd[56] <Notice>: ASL Sender
Statistics
Apr  3 01:14:34 newdance com.apple.xpc.launchd[1]
(com.apple.imfoundation.IMRemoteURLConnectionAgent) <Warning>:
Unknown key for integer: _DirtyJetsamMemoryLimit
```

Example of impact of private vs. public



```
override func viewDidLoad() {
    super.viewDidLoad()
    os_log("User %{PUBLIC}@ logged in",
            log: OSLog.userFlow, type: .info, username)
    os_log("User %{PRIVATE}@ logged in",
            log: OSLog.userFlow, type: .info, username)
}
```

Viewed in XCode:

```
LogExample[7784:105423] [viewcycle] User Antoine logged in
LogExample[7784:105423] [viewcycle] User Antoine logged in
```

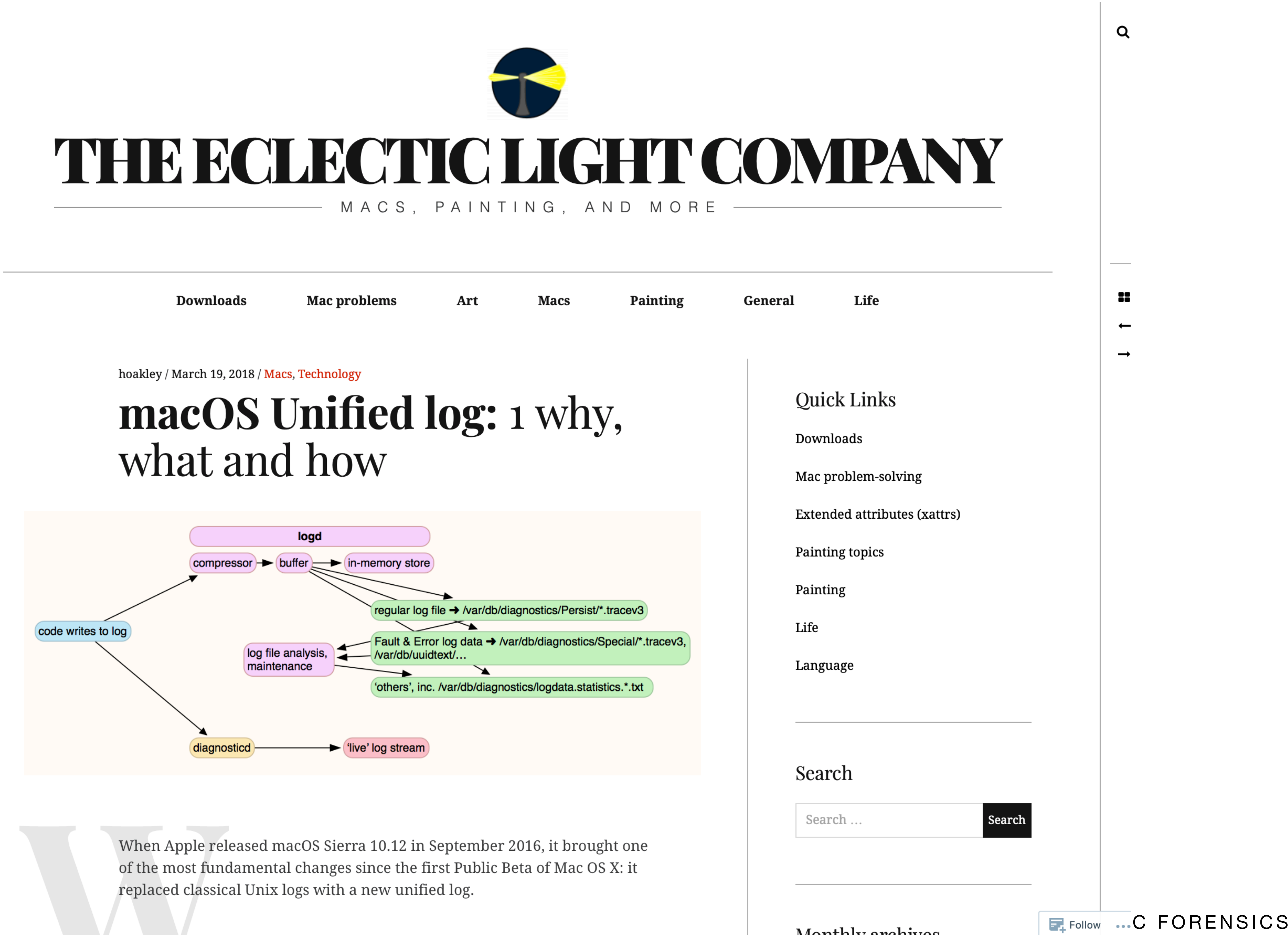
Viewed in Console.app:

```
debug 18:58:40 +0100   LogExample  User Antoine logged in
debug 18:58:40 +0100   LogExample  User <private> logged in
```

—*Source: https://www.avanderlee.com/debugging/oslog-unified-logging/*

Great write-up

https://eclecticlight.co/2018/03/19/macos-unified-log-1-why-what-and-how/



THE ECLECTIC LIGHT COMPANY

MACS, PAINTING, AND MORE

Downloads Mac problems Art Macs Painting General Life

hoakley / March 19, 2018 / Macs, Technology

macOS Unified log: 1 why, what and how

logd
compressor
buffer
in-memory store
regular log file ➜ /var/db/diagnostics/Persist/*.tracev3
code writes to log
Fault & Error log data ➜ /var/db/diagnostics/Special/*.tracev3, /var/db/uuidtext/…
log file analysis, maintenance
'others', inc. /var/db/diagnostics/logdata.statistics.*.txt
diagnosticd
'live' log stream

When Apple released macOS Sierra 10.12 in September 2016, it brought one of the most fundamental changes since the first Public Beta of Mac OS X: it replaced classical Unix logs with a new unified log.

Quick Links

Downloads

Mac problem-solving

Extended attributes (xattrs)

Painting topics

Painting

Life

Language

Search

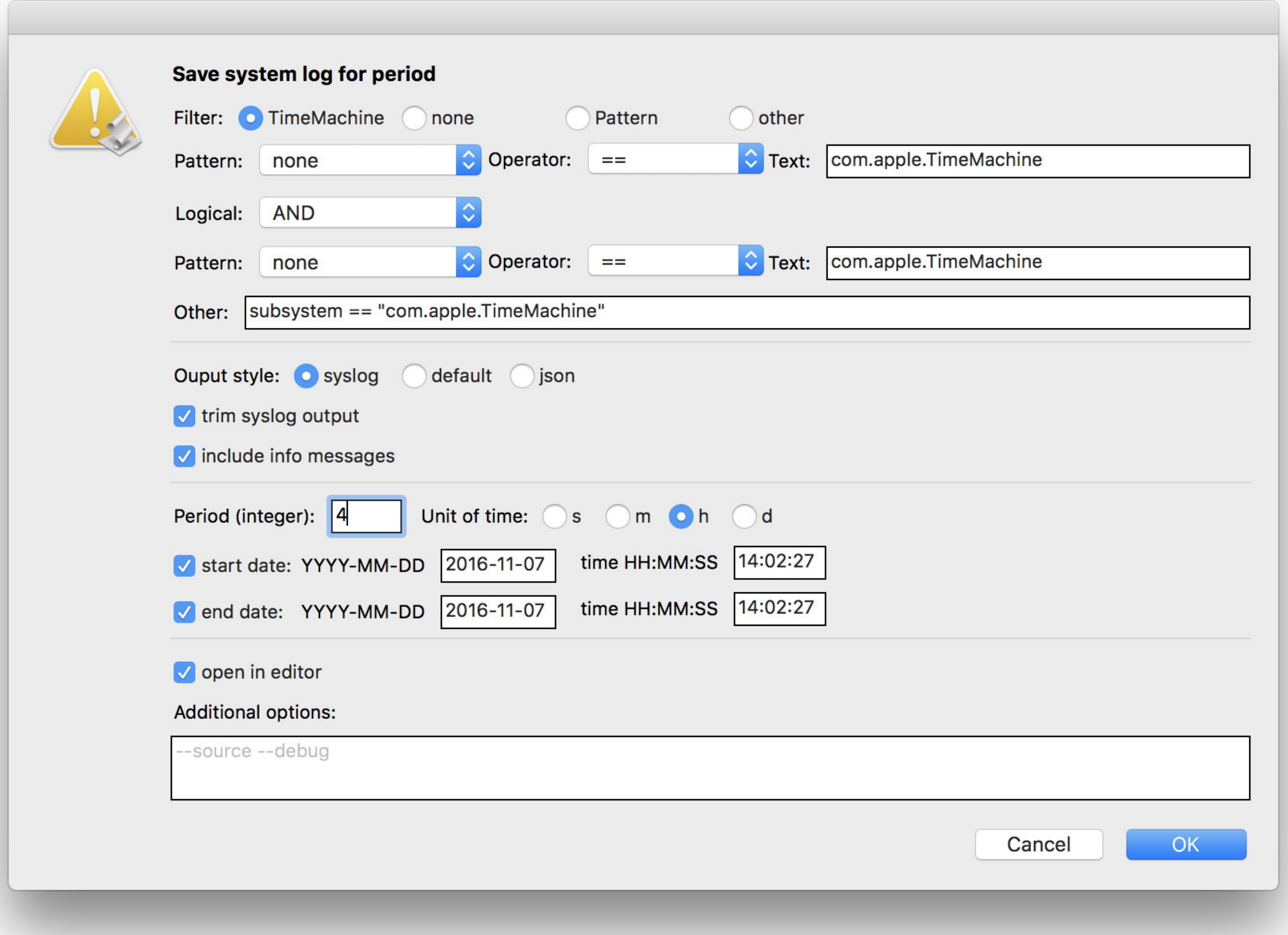
Search … Search

Monthly archives

Follow

Log tools from Electric Light Co
Log Logger

Regex search, export to CSV, flexible formatting



Save system log for period

Filter: ● TimeMachine ○ none ○ Pattern ○ other

Pattern: none Operator: == Text: com.apple.TimeMachine

Logical: AND

Pattern: none Operator: == Text: com.apple.TimeMachine

Other: subsystem == "com.apple.TimeMachine"

Ouput style: ● syslog ○ default ○ json

☑ trim syslog output

☑ include info messages

Period (integer): 4 Unit of time: ○ s ○ m ● h ○ d

☑ start date: YYYY-MM-DD 2016-11-07 time HH:MM:SS 14:02:27

☑ end date: YYYY-MM-DD 2016-11-07 time HH:MM:SS 14:02:27

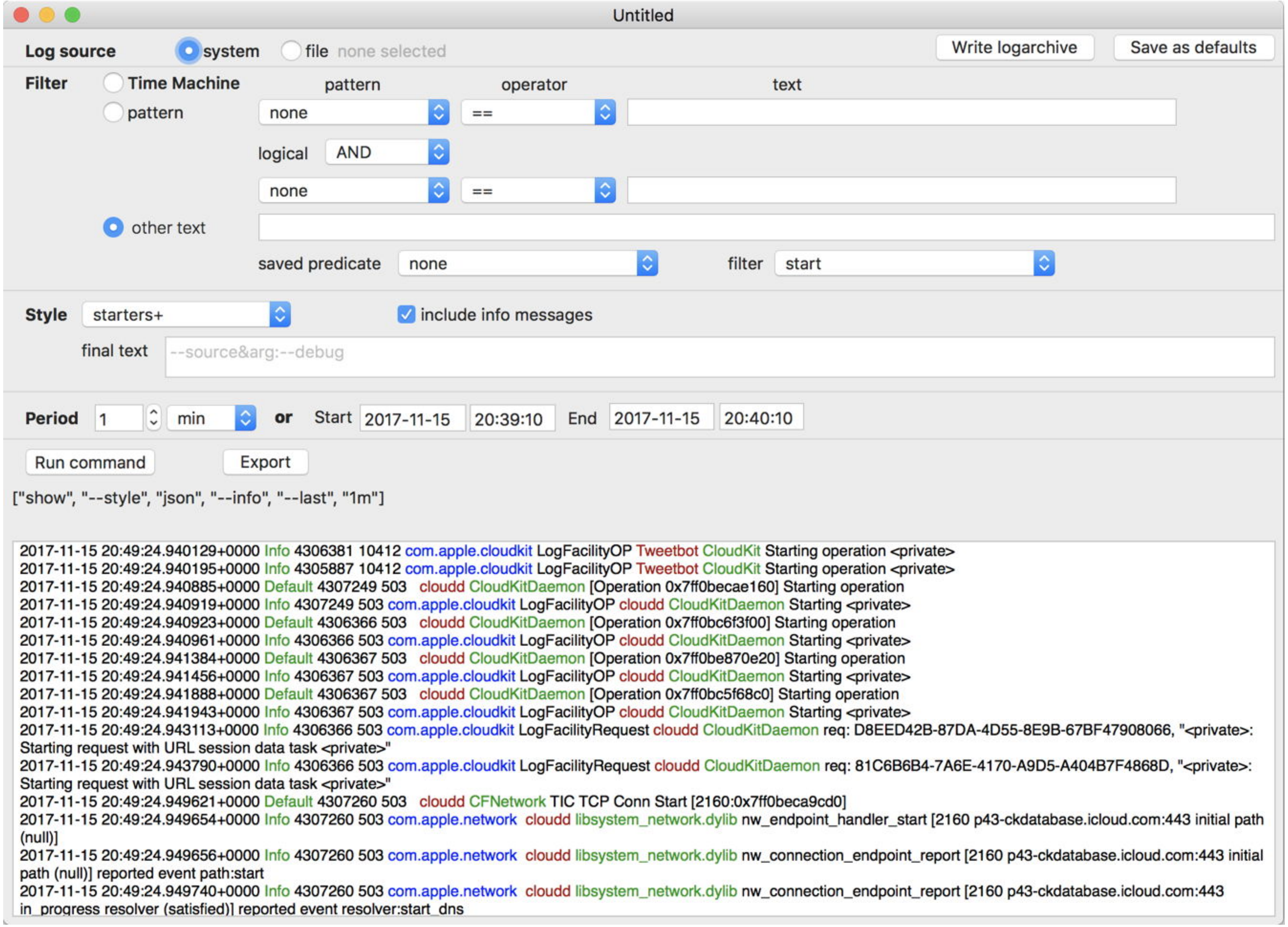
☑ open in editor

Additional options:

--source --debug

https://eclecticlight.co/downloads/

Consolidation



Log source ● system ○ file none selected

Filter ○ Time Machine

○ pattern none ==

logical AND

none ==

● other text

saved predicate none filter start

Style starters+ ☑ include info messages

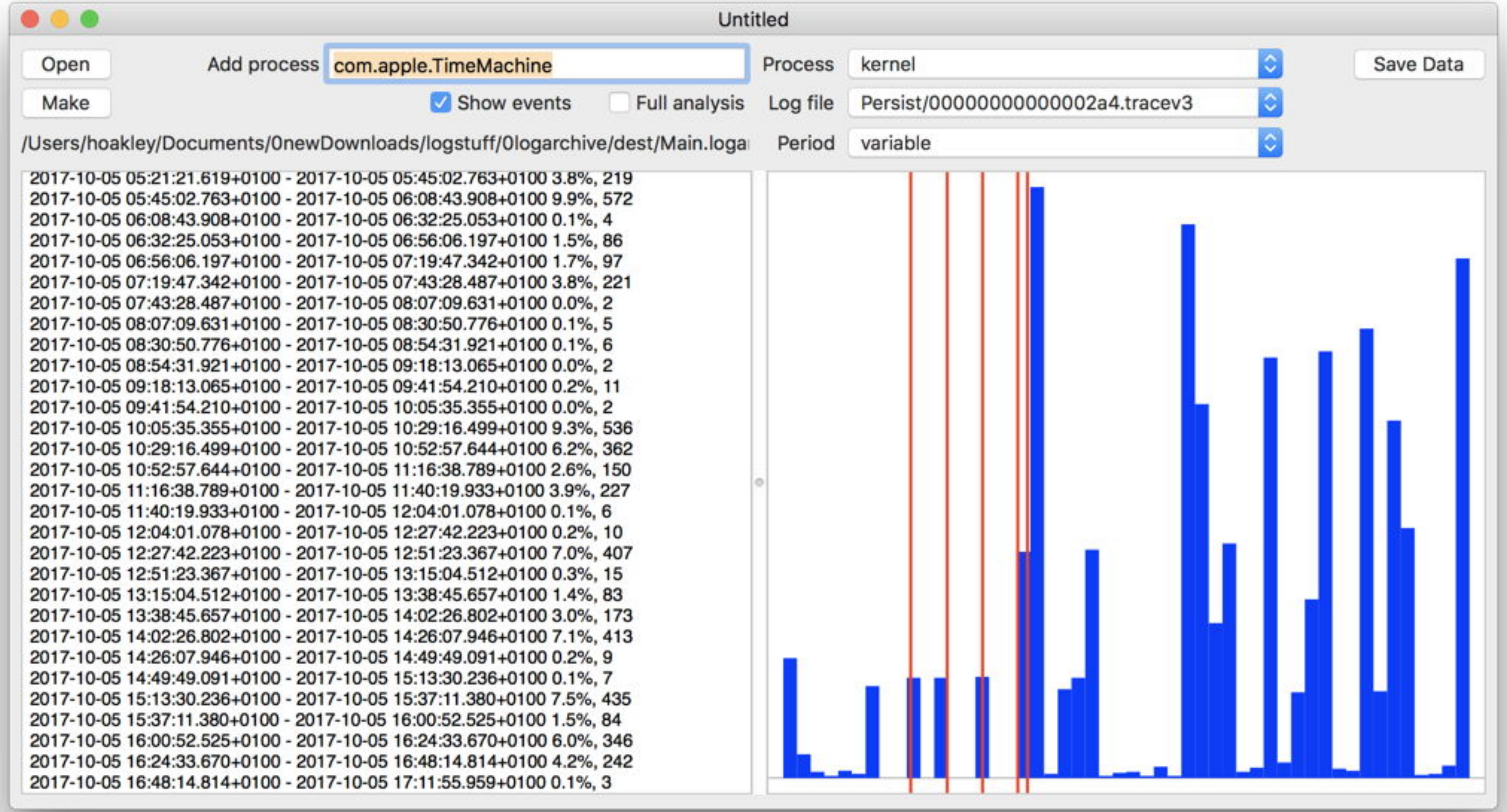
final text --source&arg:--debug

Period 1 min or Start 2017-11-15 20:39:10 End 2017-11-15 20:40:10

```
["show", "--style", "json", "--info", "--last", "1m"]
```

```
2017-11-15 20:49:24.940129+0000 Info 4306381 10412 com.apple.cloudkit LogFacilityOP Tweetbot CloudKit Starting operation <private>
2017-11-15 20:49:24.940195+0000 Info 4305887 10412 com.apple.cloudkit LogFacilityOP Tweetbot CloudKit Starting operation <private>
2017-11-15 20:49:24.940885+0000 Default 4307249 503   cloudd CloudKitDaemon [Operation 0x7ff0becae160] Starting operation
2017-11-15 20:49:24.940919+0000 Info 4307249 503 com.apple.cloudkit LogFacilityOP cloudd CloudKitDaemon Starting <private>
2017-11-15 20:49:24.940923+0000 Default 4306366 503   cloudd CloudKitDaemon [Operation 0x7ff0bc6f3f00] Starting operation
2017-11-15 20:49:24.940961+0000 Info 4306366 503 com.apple.cloudkit LogFacilityOP cloudd CloudKitDaemon Starting <private>
2017-11-15 20:49:24.941384+0000 Default 4306367 503   cloudd CloudKitDaemon [Operation 0x7ff0be870e20] Starting operation
2017-11-15 20:49:24.941456+0000 Info 4306367 503 com.apple.cloudkit LogFacilityOP cloudd CloudKitDaemon Starting <private>
2017-11-15 20:49:24.941888+0000 Default 4306367 503   cloudd CloudKitDaemon [Operation 0x7ff0bc5f68c0] Starting operation
2017-11-15 20:49:24.941943+0000 Info 4306367 503 com.apple.cloudkit LogFacilityOP cloudd CloudKitDaemon Starting <private>
2017-11-15 20:49:24.943113+0000 Info 4306366 503 com.apple.cloudkit LogFacilityRequest cloudd CloudKitDaemon req: D8EED42B-87DA-4D55-8E9B-67BF47908066, "<private>:
Starting request with URL session data task <private>"
2017-11-15 20:49:24.943790+0000 Info 4306366 503 com.apple.cloudkit LogFacilityRequest cloudd CloudKitDaemon req: 81C6B6B4-7A6E-4170-A9D5-A404B7F4868D, "<private>:
Starting request with URL session data task <private>"
2017-11-15 20:49:24.949621+0000 Default 4307260 503   cloudd CFNetwork TIC TCP Conn Start [2160:0x7ff0beca9cd0]
2017-11-15 20:49:24.949654+0000 Info 4307260 503 com.apple.network  cloudd libsystem_network.dylib nw_endpoint_handler_start [2160 p43-ckdatabase.icloud.com:443 initial path
(null)]
2017-11-15 20:49:24.949656+0000 Info 4307260 503 com.apple.network  cloudd libsystem_network.dylib nw_connection_endpoint_report [2160 p43-ckdatabase.icloud.com:443 initial
path (null)] reported event path:start
2017-11-15 20:49:24.949740+0000 Info 4307260 503 com.apple.network  cloudd libsystem_network.dylib nw_connection_endpoint_report [2160 p43-ckdatabase.icloud.com:443
in_progress resolver (satisfied)] reported event resolver:start_dns
```

Consolation, RunConsolation, Blowhole, Woodpile, DispatchView, T2M2, and RunT2M2



Add process: com.apple.TimeMachine

☑ Show events □ Full analysis

Process: kernel

Log file: Persist/00000000000002a4.tracev3

Period: variable

/Users/hoakley/Documents/0newDownloads/logstuff/0logarchive/dest/Main.loga

```
2017-10-05 05:21:21.619+0100 - 2017-10-05 05:45:02.763+0100 3.8%, 219
2017-10-05 05:45:02.763+0100 - 2017-10-05 06:08:43.908+0100 9.9%, 572
2017-10-05 06:08:43.908+0100 - 2017-10-05 06:32:25.053+0100 0.1%, 4
2017-10-05 06:32:25.053+0100 - 2017-10-05 06:56:06.197+0100 1.5%, 86
2017-10-05 06:56:06.197+0100 - 2017-10-05 07:19:47.342+0100 1.7%, 97
2017-10-05 07:19:47.342+0100 - 2017-10-05 07:43:28.487+0100 3.8%, 221
2017-10-05 07:43:28.487+0100 - 2017-10-05 08:07:09.631+0100 0.0%, 2
2017-10-05 08:07:09.631+0100 - 2017-10-05 08:30:50.776+0100 0.1%, 5
2017-10-05 08:30:50.776+0100 - 2017-10-05 08:54:31.921+0100 0.1%, 6
2017-10-05 08:54:31.921+0100 - 2017-10-05 09:18:13.065+0100 0.0%, 2
2017-10-05 09:18:13.065+0100 - 2017-10-05 09:41:54.210+0100 0.2%, 11
2017-10-05 09:41:54.210+0100 - 2017-10-05 10:05:35.355+0100 0.0%, 2
2017-10-05 10:05:35.355+0100 - 2017-10-05 10:29:16.499+0100 9.3%, 536
2017-10-05 10:29:16.499+0100 - 2017-10-05 10:52:57.644+0100 6.2%, 362
2017-10-05 10:52:57.644+0100 - 2017-10-05 11:16:38.789+0100 2.6%, 150
2017-10-05 11:16:38.789+0100 - 2017-10-05 11:40:19.933+0100 3.9%, 227
2017-10-05 11:40:19.933+0100 - 2017-10-05 12:04:01.078+0100 0.1%, 6
2017-10-05 12:04:01.078+0100 - 2017-10-05 12:27:42.223+0100 0.2%, 10
2017-10-05 12:27:42.223+0100 - 2017-10-05 12:51:23.367+0100 7.0%, 407
2017-10-05 12:51:23.367+0100 - 2017-10-05 13:15:04.512+0100 0.3%, 15
2017-10-05 13:15:04.512+0100 - 2017-10-05 13:38:45.657+0100 1.4%, 83
2017-10-05 13:38:45.657+0100 - 2017-10-05 14:02:26.802+0100 3.0%, 173
2017-10-05 14:02:26.802+0100 - 2017-10-05 14:26:07.946+0100 7.1%, 413
2017-10-05 14:26:07.946+0100 - 2017-10-05 14:49:49.091+0100 0.2%, 9
2017-10-05 14:49:49.091+0100 - 2017-10-05 15:13:30.236+0100 0.1%, 7
2017-10-05 15:13:30.236+0100 - 2017-10-05 15:37:11.380+0100 7.5%, 435
2017-10-05 15:37:11.380+0100 - 2017-10-05 16:00:52.525+0100 1.5%, 84
2017-10-05 16:00:52.525+0100 - 2017-10-05 16:24:33.670+0100 6.0%, 346
2017-10-05 16:24:33.670+0100 - 2017-10-05 16:48:14.814+0100 4.2%, 242
2017-10-05 16:48:14.814+0100 - 2017-10-05 17:11:55.959+0100 0.1%, 3
```

Mac Forensic Tools

ISSA
Information Systems Security Association
National Capital Chapter

Open Source
Proprietary

Built-in tools

sysdiagnose(1)

Used by Apple to generate system bug reports

“one-stop shopping for system diagnostics.”

—ps(1), zprint(1), and 60 (macOS) / 12 (iOS) other commands

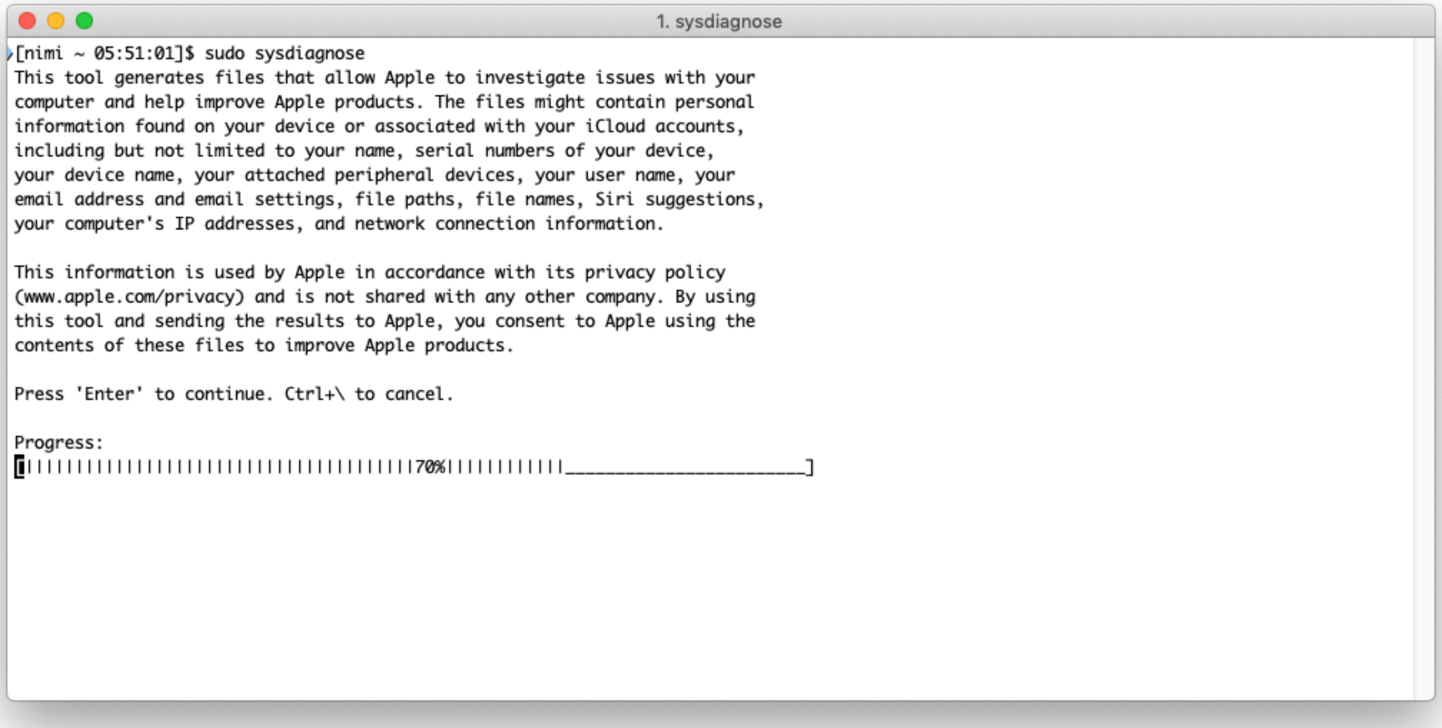
Also can be run from keychord:

MacOS: ⇧ ctrl ⌥ ⌘ .

iOS: Vol. Up + Vol. Down + Power

TvOS: Play/Pause + Vol. Down (5 seconds)

WatchOS: Digital Crown + Side Button (1 second)



```
[nimi ~ 05:51:01]$ sudo sysdiagnose
This tool generates files that allow Apple to investigate issues with your
computer and help improve Apple products. The files might contain personal
information found on your device or associated with your iCloud accounts,
including but not limited to your name, serial numbers of your device,
your device name, your attached peripheral devices, your user name, your
email address and email settings, file paths, file names, Siri suggestions,
your computer's IP addresses, and network connection information.

This information is used by Apple in accordance with its privacy policy
(www.apple.com/privacy) and is not shared with any other company. By using
this tool and sending the results to Apple, you consent to Apple using the
contents of these files to improve Apple products.

Press 'Enter' to continue. Ctrl+\ to cancel.

Progress:
[|||||||||||||||||||||||||||||||||||||||70%||||||||||||________________________]
```

Output available at '/var/tmp/sysdiagnose_2019.04.24_05-51-06-0400_Mac_OS_X_Macmini8-1_18E226.tar.gz'.

Output is big:

```
Output available at '/var/tmp/
sysdiagnose_2019.04.24_05-51-06-0400_Mac_OS_X_Macmini8-1
_18E226.tar.gz'.
[nimi ~ 05:54:03]$ ls -l /var/tmp/
sysdiagnose_2019.04.24_05-51-06-0400_Mac_OS_X_Macmini8-1
_18E226.tar.gz
-rw-rw-r--  1 root  wheel  267205499 Apr 24 05:54 /var/
tmp/
sysdiagnose_2019.04.24_05-51-06-0400_Mac_OS_X_Macmini8-1
_18E226.tar.gz
[nimi ~ 05:56:23]$
```

Output on my system: 1349 files!

```
    128 Apr 24 05:57 Accessibility/
1649716 Apr 24 05:51 BluetoothTraceFile.pklg
    262 Apr 24 05:52 DiskMountConditioner.json
    160 Apr 24 05:57 Preferences/
    256 Apr 24 05:57 SystemConfiguration/
    928 Apr 24 05:57 SystemProfiler/
   1056 Apr 24 05:57 WiFi/
 101667 Apr 24 05:53 acdiagnose-501.txt
    348 Apr 24 05:52 airport_info.txt
  16271 Apr 24 05:53 apfs_stats.txt
    412 Apr 24 05:52 applessdstats.txt
 110509 Apr 24 05:53 apsd-status.txt
   1237 Apr 24 05:52 bc_stats.txt
    289 Apr 24 05:53 bless_info.txt
    848 Apr 24 05:52 bootstamps.txt
    512 Apr 24 05:57 brctl/
 117379 Apr 24 05:53 ckksctl_status.txt
   2556 Mar  2 14:48 com.apple.windowserver.plist
    224 Apr 24 05:57 crashes_and_spins/
     45 Apr 24 05:53 csrutil-status.txt
      0 Apr 24 05:53 ctsctl-list-0.txt
```

spindump(8)

Collects detailed stats on all running programs.
Shows where programs are running.

```
Process:         1Password 7 [548]
UUID:            A3876C62-4FA9-3A7C-A1C3-64622AF89F27
Path:            /Applications/1Password 7.app/Contents/MacOS/1Password 7
Architecture:    x86_64
Parent:          launchd [1]
UID:             501
Footprint:       112.59 MB
Start time:      2019-04-24 06:07:45 -0400
End time:        2019-04-24 06:07:55 -0400
Num samples:     1001 (1-1001)
CPU Time:        0.003s (5.4M cycles, 1355.2K instructions, 4.01c/i)
Note:            2 idle work queue threads omitted

  Thread 0x19fd             DispatchQueue 1            1001 samples (1-1001)
priority 46 (base 46)     cpu time 0.002s (5.0M cycles, 1257.0K instructions, 3.95c/
i)
  1001  start + 1 (libdyld.dylib + 91093) [0x7fff5b7d83d5]
    1001  NSApplicationMain + 777 (AppKit + 13296) [0x7fff2c9ce3f0]
      1001  -[NSApplication run] + 699 (AppKit + 81584) [0x7fff2c9deeb0]
        1001  -[NSApplication(NSEvent)
_nextEventMatchingEventMask:untilDate:inMode:dequeue:] + 1361 (AppKit + 105875)
[0x7fff2c9e4d93]
          1001  _DPSNextEvent + 965 (AppKit + 110587) [0x7fff2c9e5ffb]
            1001  _BlockUntilNextEventMatchingListInModeWithFilter + 64 (HIToolbox +
42150) [0x7fff2e64b4a6]
              1001  ReceiveNextEventCommon + 603 (HIToolbox + 42773) [0x7fff2e64b715]
```

Open Source Tools

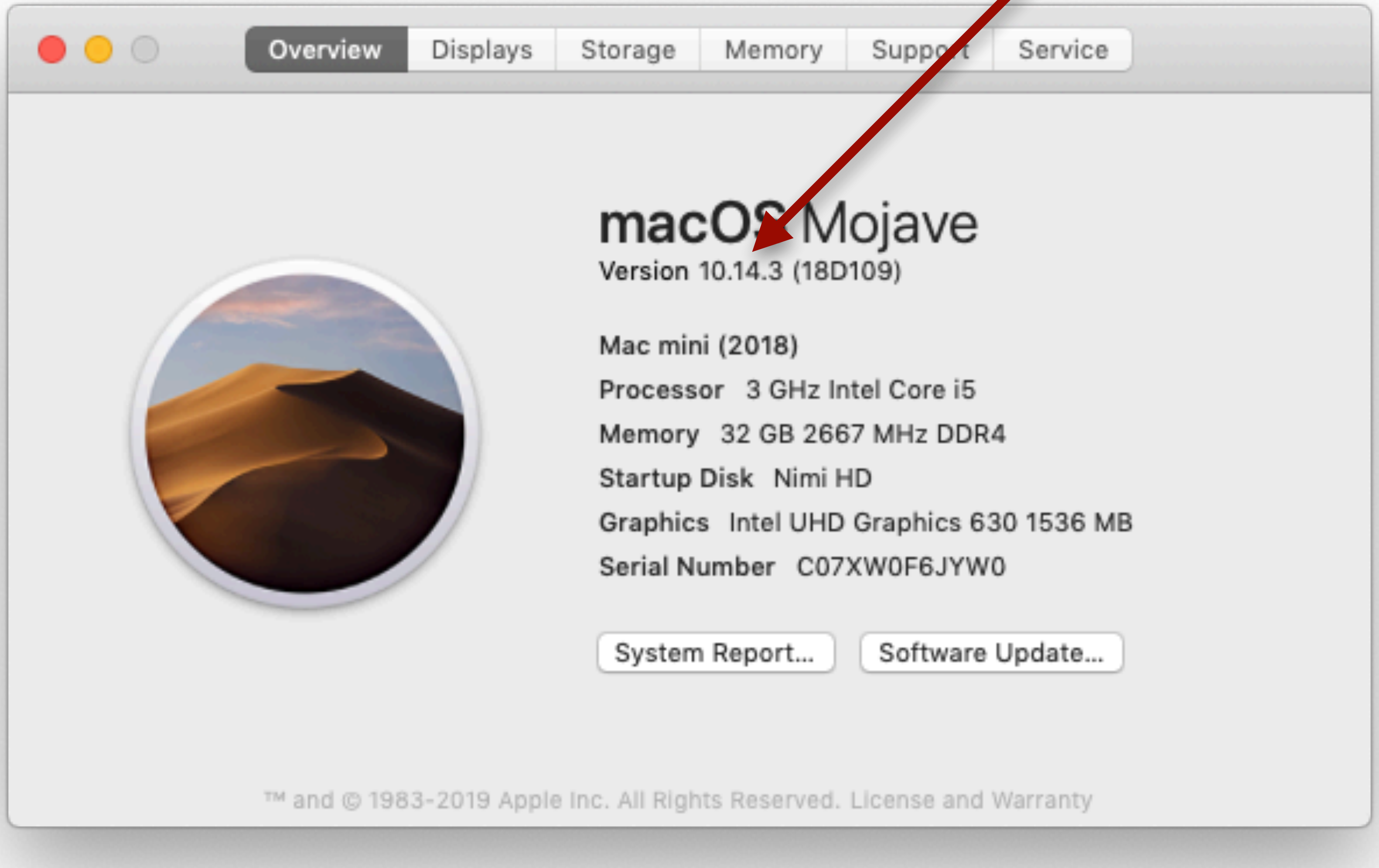
Volatility!

Original developed by Aaron Walters for his PhD thesis
Now maintained by The Volatility Foundation

Key things to note:

- **Volatility is a Python2.7 program.**
- **Volatility is also distributed as a "compiled" program.**
- **Volatility needs a "profile" for your kernel**
- **Creating a profile requires "debug" kernel.**
- **No debug kernel available for 10.14.3 yet.**

Click here to make kernel build appear



macOS Mojave
Version 10.14.3 (18D109)

Mac mini (2018)
Processor 3 GHz Intel Core i5
Memory 32 GB 2667 MHz DDR4
Startup Disk Nimi HD
Graphics Intel UHD Graphics 630 1536 MB
Serial Number C07XW0F6JYW0

™ and © 1983-2019 Apple Inc. All Rights Reserved. License and Warranty

Some open source developers have created tools for parsing mac-specific data structures.

Apple Pattern of Life Lazy Output'er (APOLLO)
https://github.com/mac4n6/APOLLO

MAC APT (Artifact Parsing Tool)
https://github.com/ydkhatri/mac_apt

OSX Auditor
https://github.com/jipegit/OSXAuditor

OSXRipper
https://github.com/bolodev/osxripper

iParser
http://az4n6.blogspot.co.uk/2012/08/automated-plist-parser.html
https://github.com/mdegrazia/iParser

Mac Plist Ripper
https://bitbucket.org/chrishargreaves/mac_plist_ripper

CCL Forensics BPlist parser
https://github.com/cclgroupltd/ccl-bplist

macmade/KeychainCracker.
https://github.com/macmade/KeychainCracker

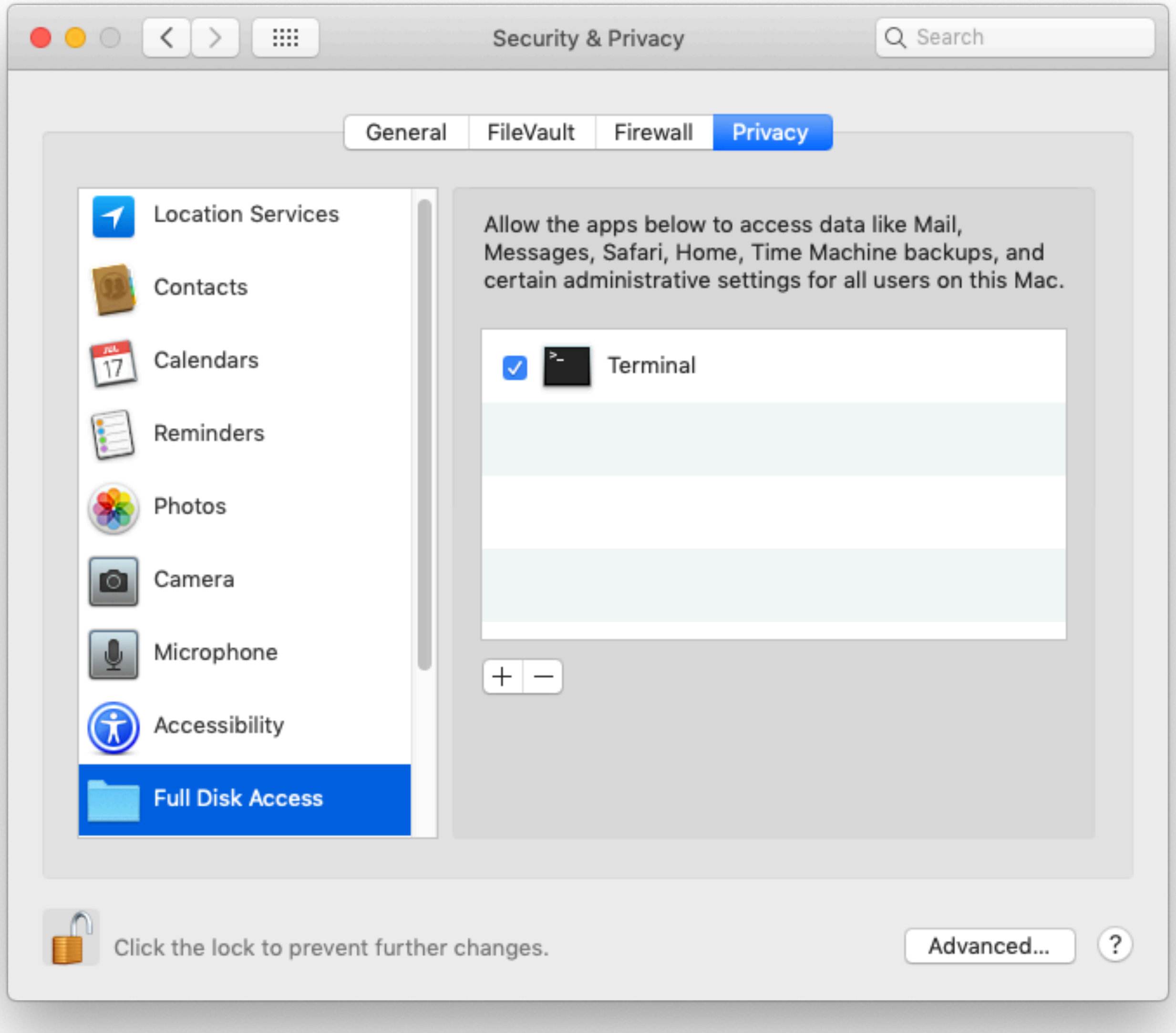
Most of these tools can be used on live systems or mounted disk images.

Under macOS 10.14,
parts of the file system are restricted from the user!

If you see this:

```
[nimi ~ 18:37:07]$ ls -l ~/Library/Mail/
ls: : Operation not permitted
[nimi ~ 18:37:13]$
```

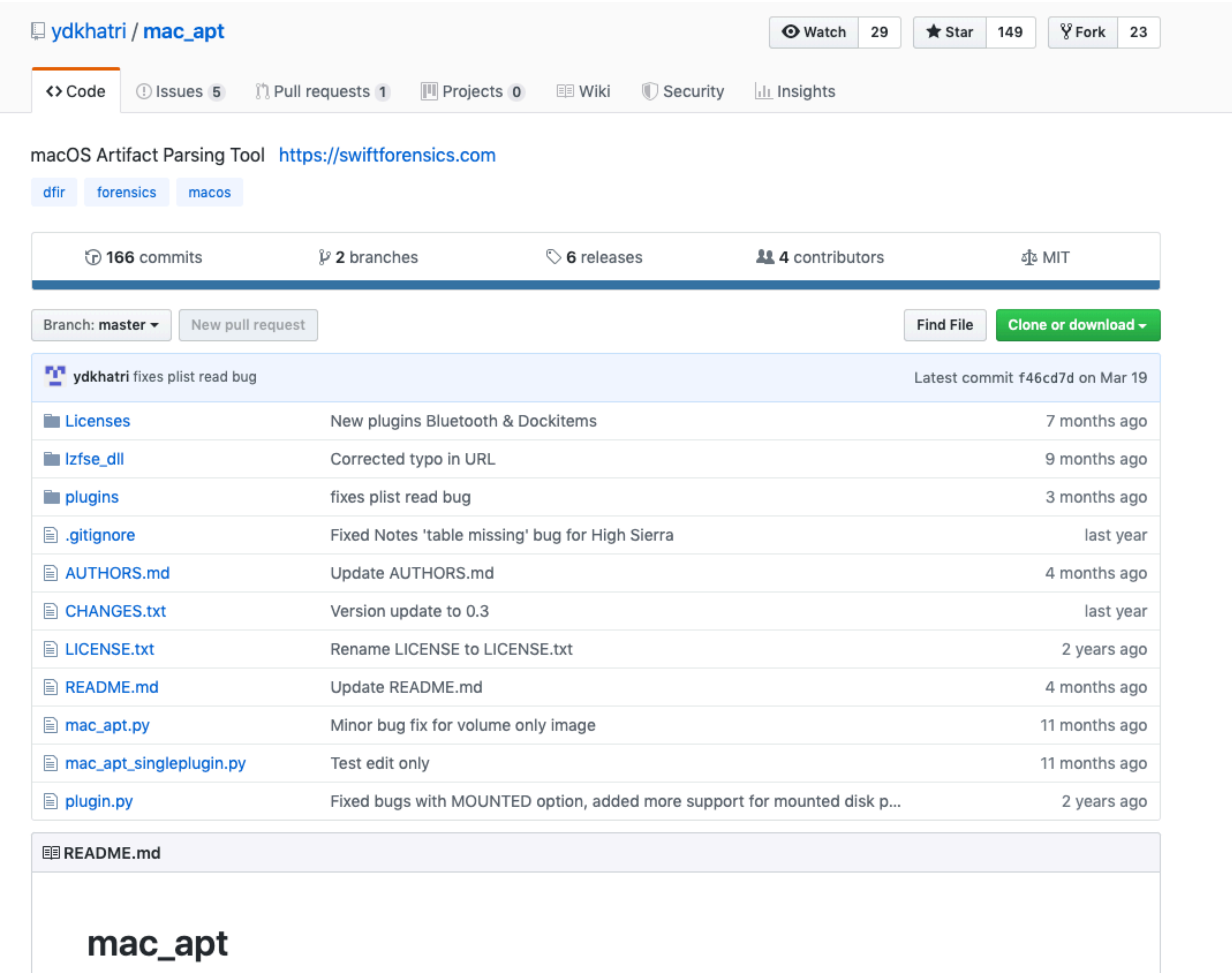
You need to do this:



Allow the apps below to access data like Mail, Messages, Safari, Home, Time Machine backups, and certain administrative settings for all users on this Mac.

☑ Terminal

mac_apt — comprehensive macOS artifact parser



macOS Artifact Parsing Tool https://swiftforensics.com

ydkhatri fixes plist read bug — Latest commit f46cd7d on Mar 19

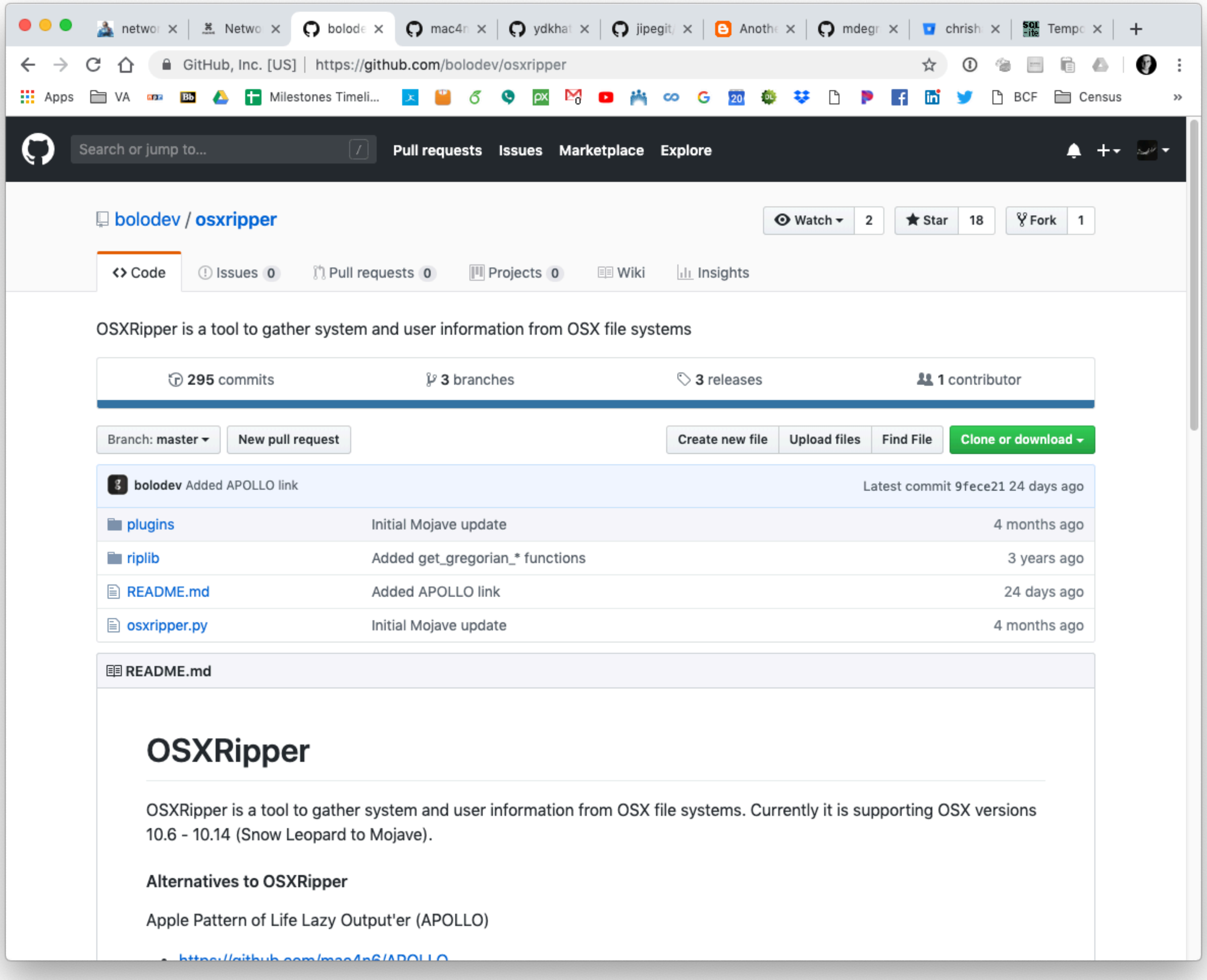
Licenses	New plugins Bluetooth & Dockitems	7 months ago
lzfse_dll	Corrected typo in URL	9 months ago
plugins	fixes plist read bug	3 months ago
.gitignore	Fixed Notes 'table missing' bug for High Sierra	last year
AUTHORS.md	Update AUTHORS.md	4 months ago
CHANGES.txt	Version update to 0.3	last year
LICENSE.txt	Rename LICENSE to LICENSE.txt	2 years ago
README.md	Update README.md	4 months ago
mac_apt.py	Minor bug fix for volume only image	11 months ago
mac_apt_singleplugin.py	Test edit only	11 months ago
plugin.py	Fixed bugs with MOUNTED option, added more support for mounted disk p...	2 years ago

README.md

mac_apt

OSXRipper
github.com/bolodev/osxripper

“OSXRipper is a tool to gather system and user information from OSX file systems. Currently it is supporting OSX versions 10.6 - 10.14 (Snow Leopard to Mojave).



OSXRipper is a tool to gather system and user information from OSX file systems

295 commits · 3 branches · 3 releases · 1 contributor

bolodev Added APOLLO link — Latest commit 9fece21 24 days ago

plugins	Initial Mojave update	4 months ago
riplib	Added get_gregorian_* functions	3 years ago
README.md	Added APOLLO link	24 days ago
osxripper.py	Initial Mojave update	4 months ago

README.md

OSXRipper

OSXRipper is a tool to gather system and user information from OSX file systems. Currently it is supporting OSX versions 10.6 - 10.14 (Snow Leopard to Mojave).

Alternatives to OSXRipper

Apple Pattern of Life Lazy Output'er (APOLLO)

- https://github.com/mac4n6/APOLLO

Commercial Tools

Physical Acquisition

Disk imaging:

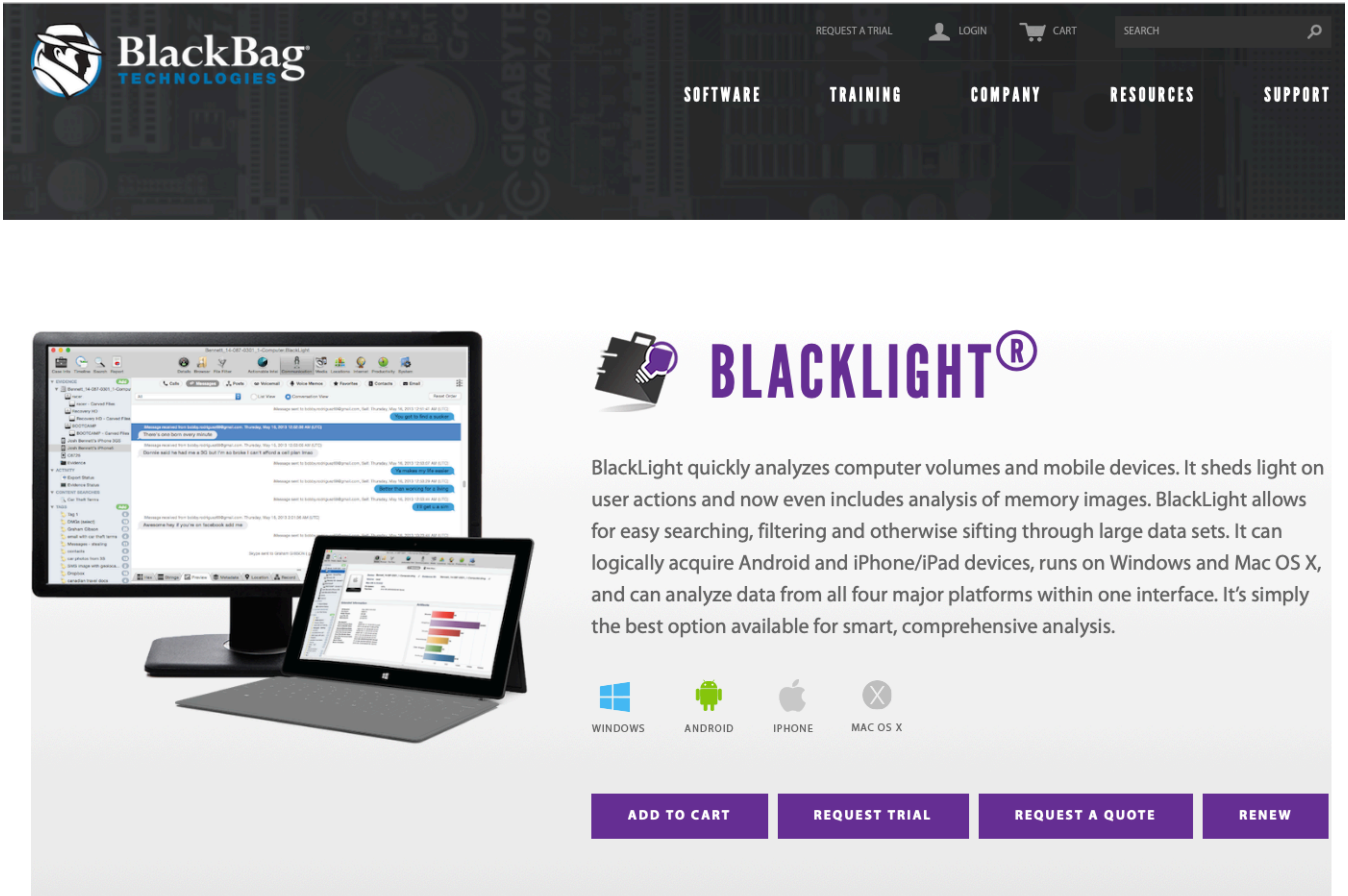
- Built-in Unix commands (e.g. dd)
- Open-source Unix images (e.g. guymager)
- Commercial tools (e.g. Macquisition)

- T2-encrypted drives:
  - —*All can image the plaintext if you have the password.*
  - —*There is (currently) no way to decrypt a T2-encrypted image if it was imaged without the password.*

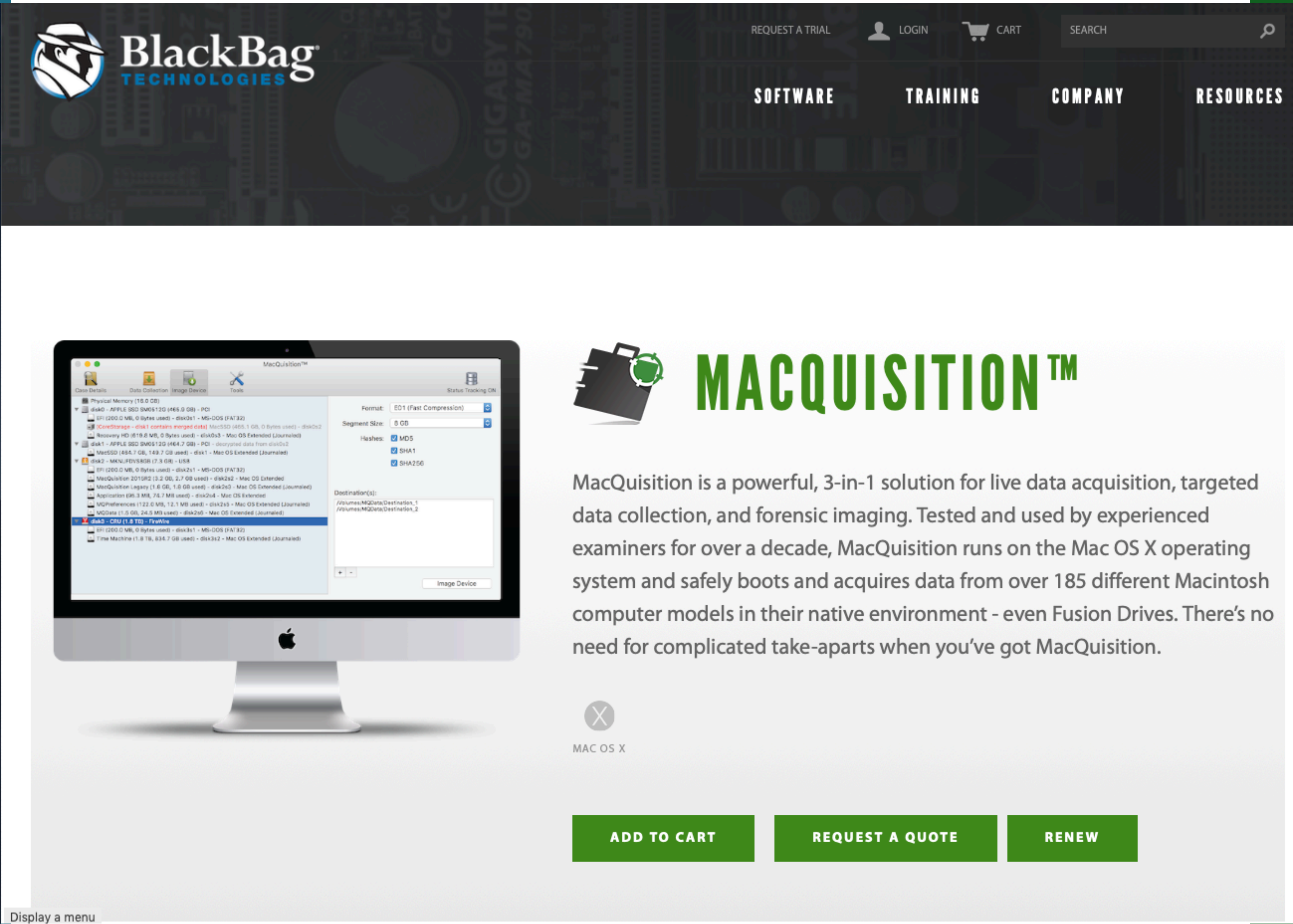
Memory Imaging:

- Easiest way is to run macOS in a VM and suspend!
- Failing that, use a commercial tool.

Blacklight is the leading forensics tool for MacOS. It runs on Mac and Windows and analyzes everything.



BlackLight quickly analyzes computer volumes and mobile devices. It sheds light on user actions and now even includes analysis of memory images. BlackLight allows for easy searching, filtering and otherwise sifting through large data sets. It can logically acquire Android and iPhone/iPad devices, runs on Windows and Mac OS X, and can analyze data from all four major platforms within one interface. It's simply the best option available for smart, comprehensive analysis.



```
Physical Memory (16.0 GB)
▼ disk0 - APPLE SSD SM0512G (465.9 GB) - PCI
    EFI (200.0 MB, 0 Bytes used) - disk0s1 - MS-DOS (FAT32)
    [CoreStorage - disk1 contains merged data] MacSSD (465.1 GB, 0 Bytes used) - disk0s2
    Recovery HD (619.8 MB, 0 Bytes used) - disk0s3 - Mac OS Extended (Journaled)
▼ disk1 - APPLE SSD SM0512G (464.7 GB) - PCI - decrypted data from disk0s2
    MacSSD (464.7 GB, 149.7 GB used) - disk1 - Mac OS Extended (Journaled)
▼ disk2 - MKNUFDVS8GB (7.3 GB) - USB
    EFI (200.0 MB, 0 Bytes used) - disk2s1 - MS-DOS (FAT32)
    MacQuisition 2015R2 (3.2 GB, 2.7 GB used) - disk2s2 - Mac OS Extended
    MacQuisition Legacy (1.6 GB, 1.6 GB used) - disk2s3 - Mac OS Extended (Journaled)
    Application (95.3 MB, 74.7 MB used) - disk2s4 - Mac OS Extended
    MQPreferences (122.0 MB, 12.1 MB used) - disk2s5 - Mac OS Extended (Journaled)
    MQData (1.5 GB, 24.5 MB used) - disk2s6 - Mac OS Extended (Journaled)
▼ disk3 - CRU (1.8 TB) - FireWire
    EFI (200.0 MB, 0 Bytes used) - disk3s1 - MS-DOS (FAT32)
    Time Machine (1.8 TB, 834.7 GB used) - disk3s2 - Mac OS Extended (Journaled)
```

Format: E01 (Fast Compression)

Segment Size: 8 GB

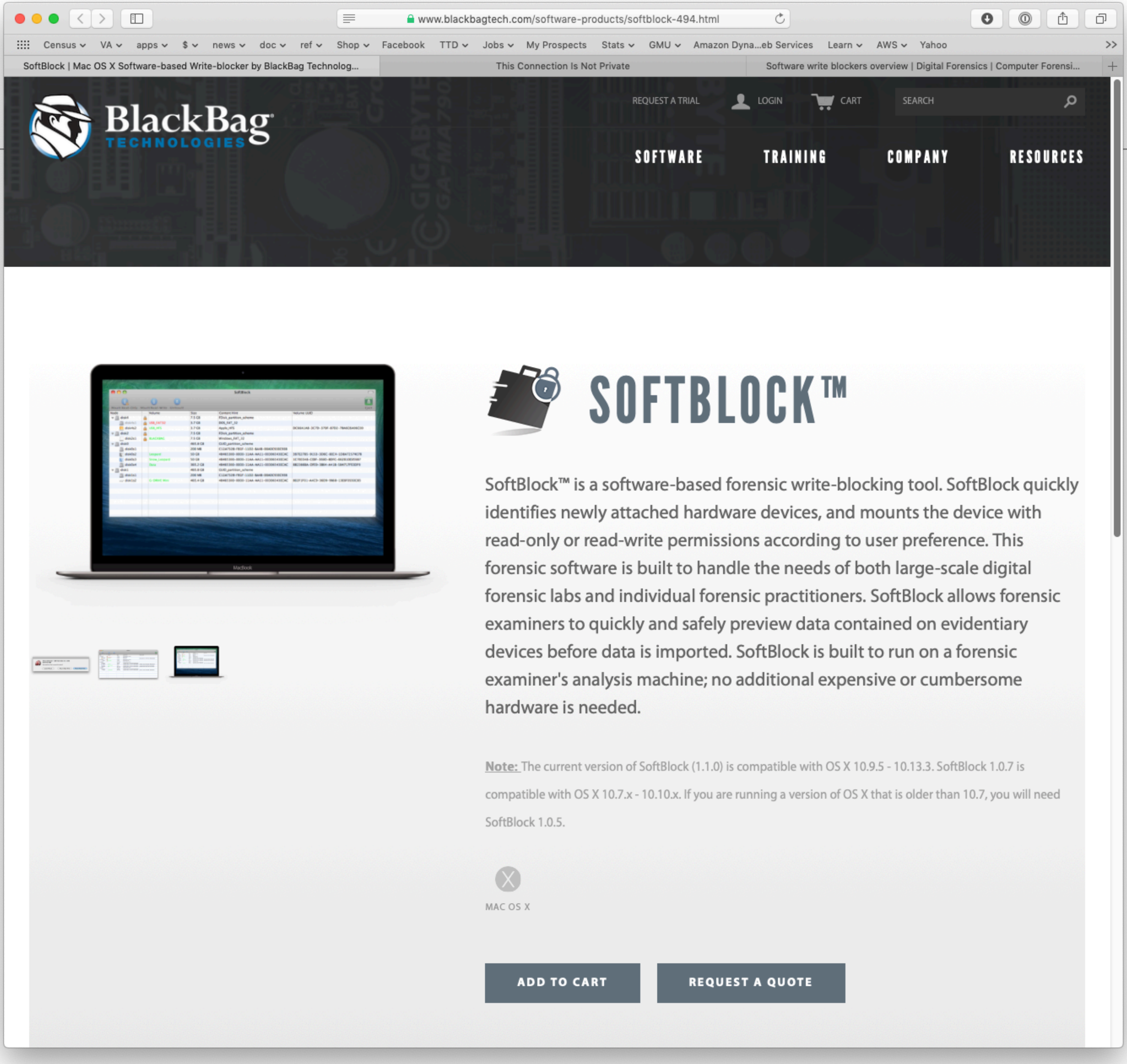
Hashes: ☑ MD5 ☑ SHA1 ☑ SHA256

Destination(s):

```
/Volumes/MQData/Destination_1
/Volumes/MQData/Destination_2
```

MACQUISITION™

MacQuisition is a powerful, 3-in-1 solution for live data acquisition, targeted data collection, and forensic imaging. Tested and used by experienced examiners for over a decade, MacQuisition runs on the Mac OS X operating system and safely boots and acquires data from over 185 different Macintosh computer models in their native environment - even Fusion Drives. There's no need for complicated take-aparts when you've got MacQuisition.


www.blackbagtech.com/software-products/softblock-494.html
Census
VA
apps
$
news
doc
ref
Shop
Facebook
TTD
Jobs
My Prospects
Stats
GMU
Amazon Dyna...eb Services
Learn
AWS
Yahoo
SoftBlock | Mac OS X Software-based Write-blocker by BlackBag Technolog...
This Connection Is Not Private
Software write blockers overview | Digital Forensics | Computer Forensi...
BlackBag TECHNOLOGIES
REQUEST A TRIAL
LOGIN
CART
SEARCH
SOFTWARE
TRAINING
COMPANY
RESOURCES

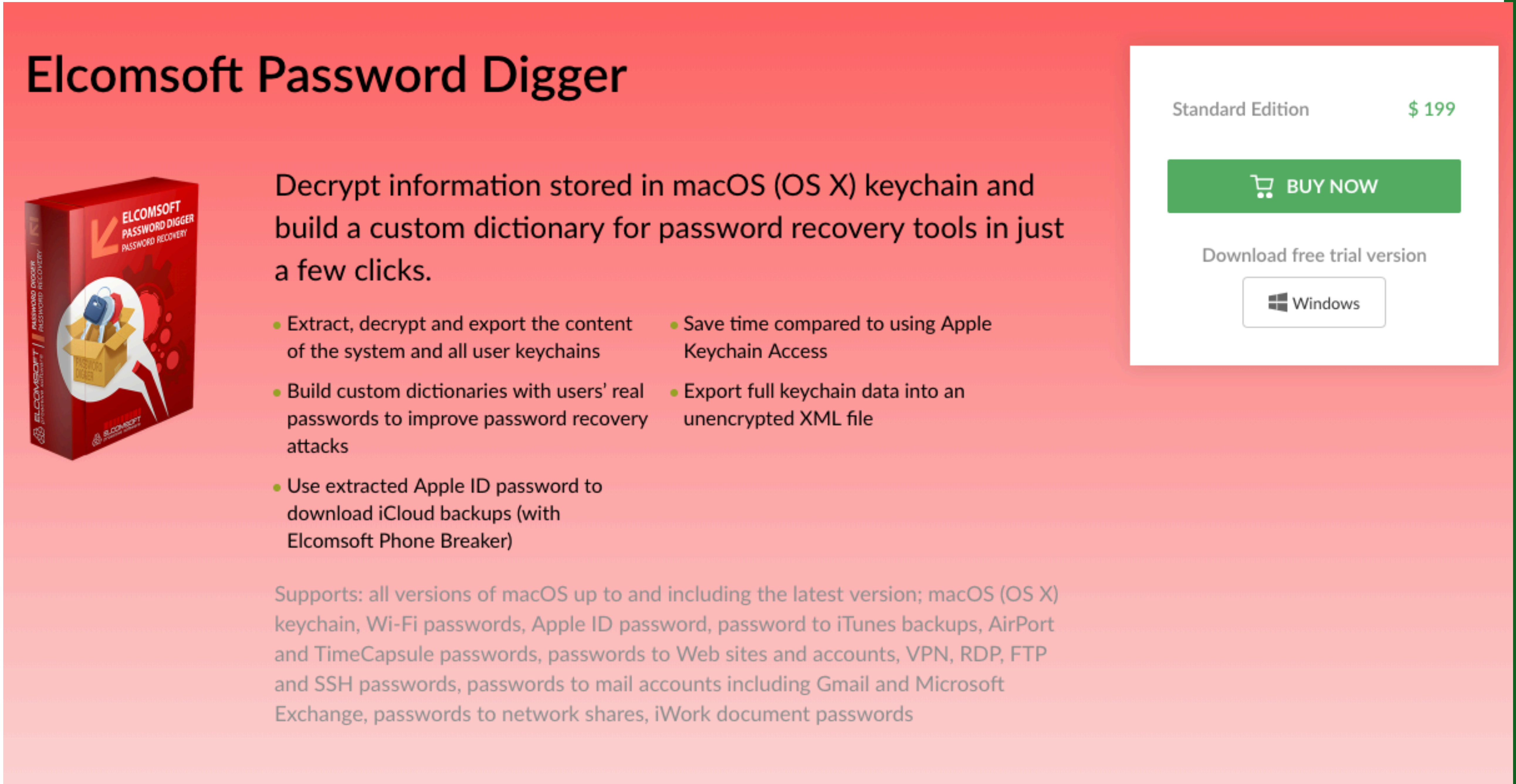
SOFTBLOCK™

SoftBlock™ is a software-based forensic write-blocking tool. SoftBlock quickly identifies newly attached hardware devices, and mounts the device with read-only or read-write permissions according to user preference. This forensic software is built to handle the needs of both large-scale digital forensic labs and individual forensic practitioners. SoftBlock allows forensic examiners to quickly and safely preview data contained on evidentiary devices before data is imported. SoftBlock is built to run on a forensic examiner's analysis machine; no additional expensive or cumbersome hardware is needed.

Note: The current version of SoftBlock (1.1.0) is compatible with OS X 10.9.5 - 10.13.3. SoftBlock 1.0.7 is compatible with OS X 10.7.x - 10.10.x. If you are running a version of OS X that is older than 10.7, you will need SoftBlock 1.0.5.

Elcomsoft Password Digger

$199. Runs on Windows; decrypts system and user keychains from MacOS computer



Elcomsoft Password Digger

Decrypt information stored in macOS (OS X) keychain and build a custom dictionary for password recovery tools in just a few clicks.

- Extract, decrypt and export the content of the system and all user keychains
- Build custom dictionaries with users' real passwords to improve password recovery attacks
- Use extracted Apple ID password to download iCloud backups (with Elcomsoft Phone Breaker)
- Save time compared to using Apple Keychain Access
- Export full keychain data into an unencrypted XML file

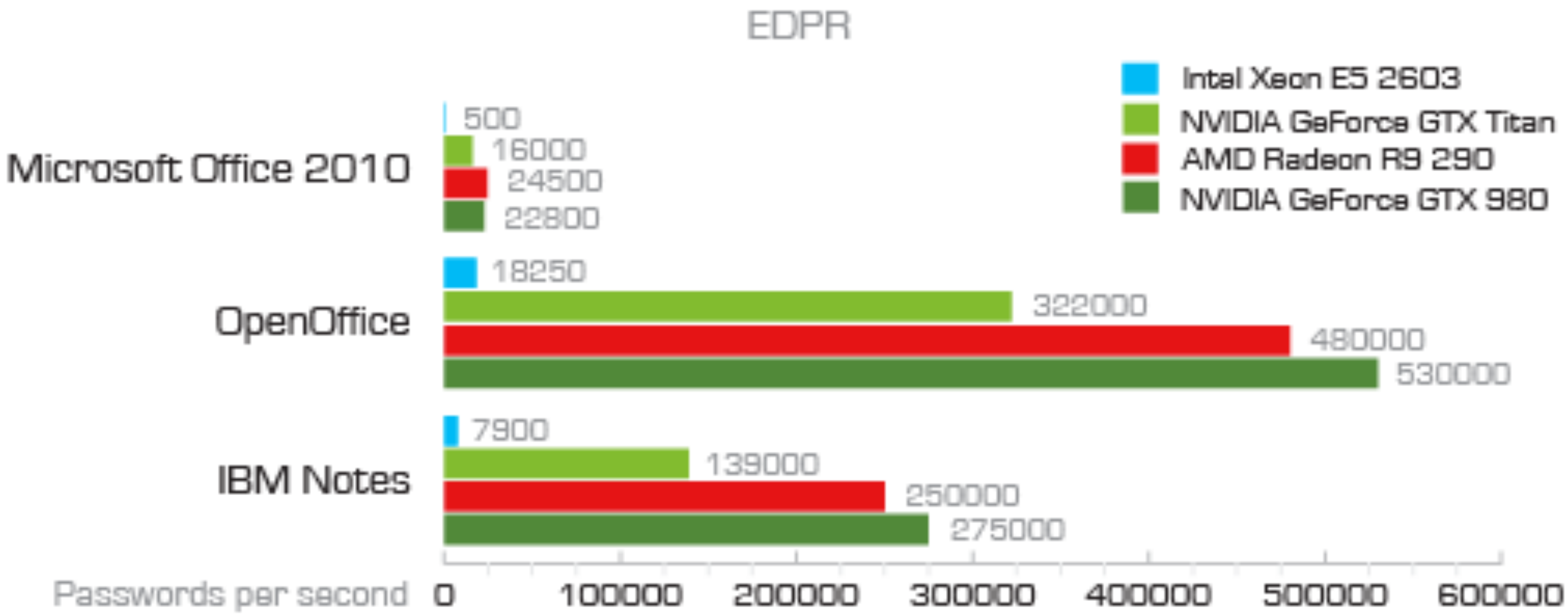
Supports: all versions of macOS up to and including the latest version; macOS (OS X) keychain, Wi-Fi passwords, Apple ID password, password to iTunes backups, AirPort and TimeCapsule passwords, passwords to Web sites and accounts, VPN, RDP, FTP and SSH passwords, passwords to mail accounts including Gmail and Microsoft Exchange, passwords to network shares, iWork document passwords

Standard Edition $ 199
BUY NOW
Download free trial version
Windows
ELCOMSOFT PASSWORD DIGGER PASSWORD RECOVERY

You must have a good dictionary for cracking modern encryption systems.

Cracks:

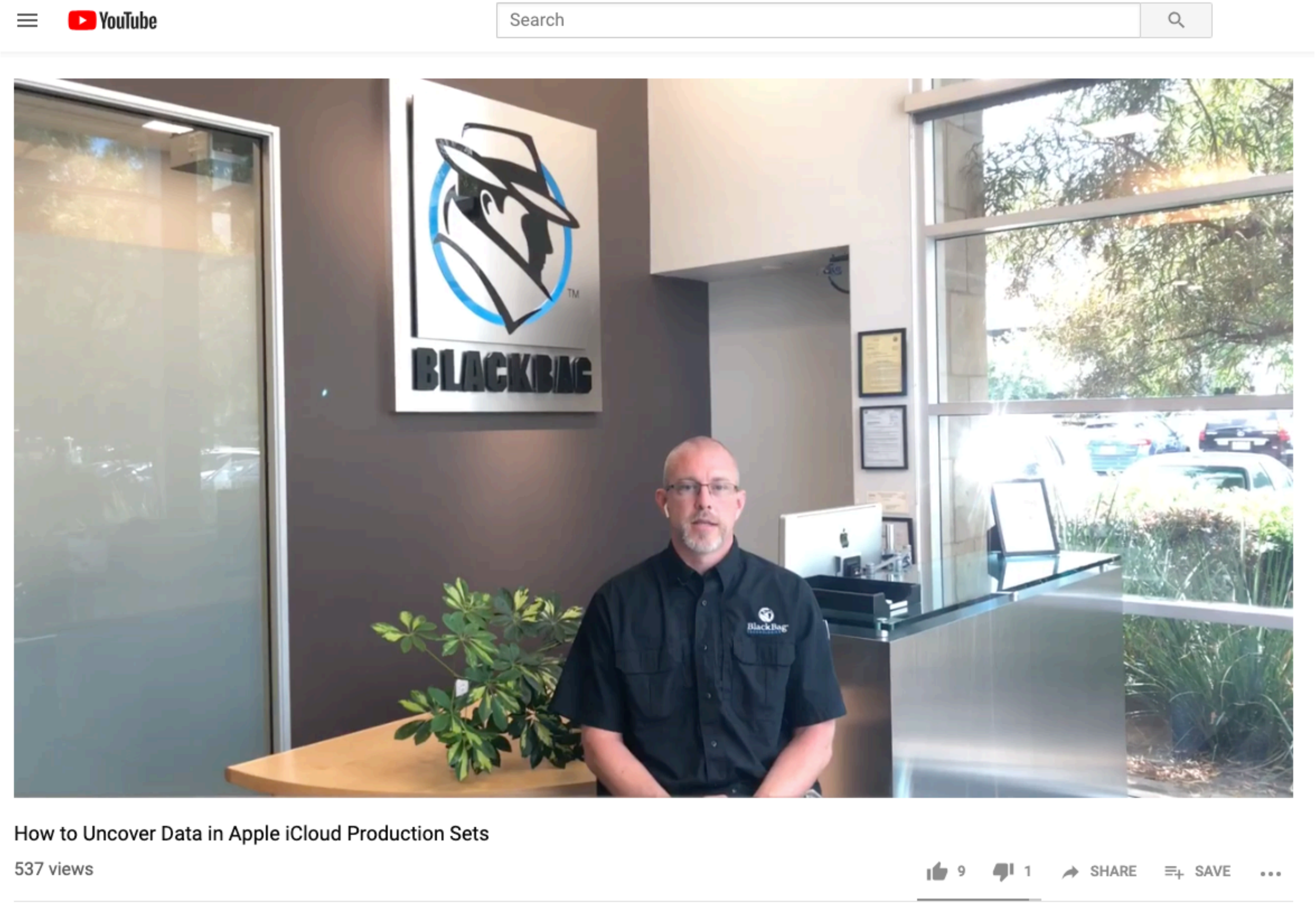
keychain; Wi-Fi passwords; Apple ID passwords; iTunes backups; AirPort and TImeCapsul passwords; passwords to Web sites; VPN; RDP; FTP and SSH; passwords to mail accounts. iWork document passwords



BlackBag: Apple iCloud Production Service
May 31, 2018

“Have you worked with Apple on a legal process? Make sure you reveal all the data using our new#iCloud processing service. Once processed, bring together iCloud production sets, mobile devices & desktop images in BlackLight. 🔦 Learn more: http://bit.ly/2NwG2en #DFIR #Mac4n6

REVEAL THE TRUTH
Uncover essential data in Apple iCloud production sets from Search Warrants



How to Uncover Data in Apple iCloud Production Sets

537 views

https://www.youtube.com/watch?v=eU24GV7x1KQ

Blog on iCloud production

Apple responds to lawful and legal requests

Apple has a document on their website for assisting you in producing requests.

Results may be encrypted and/or compressed

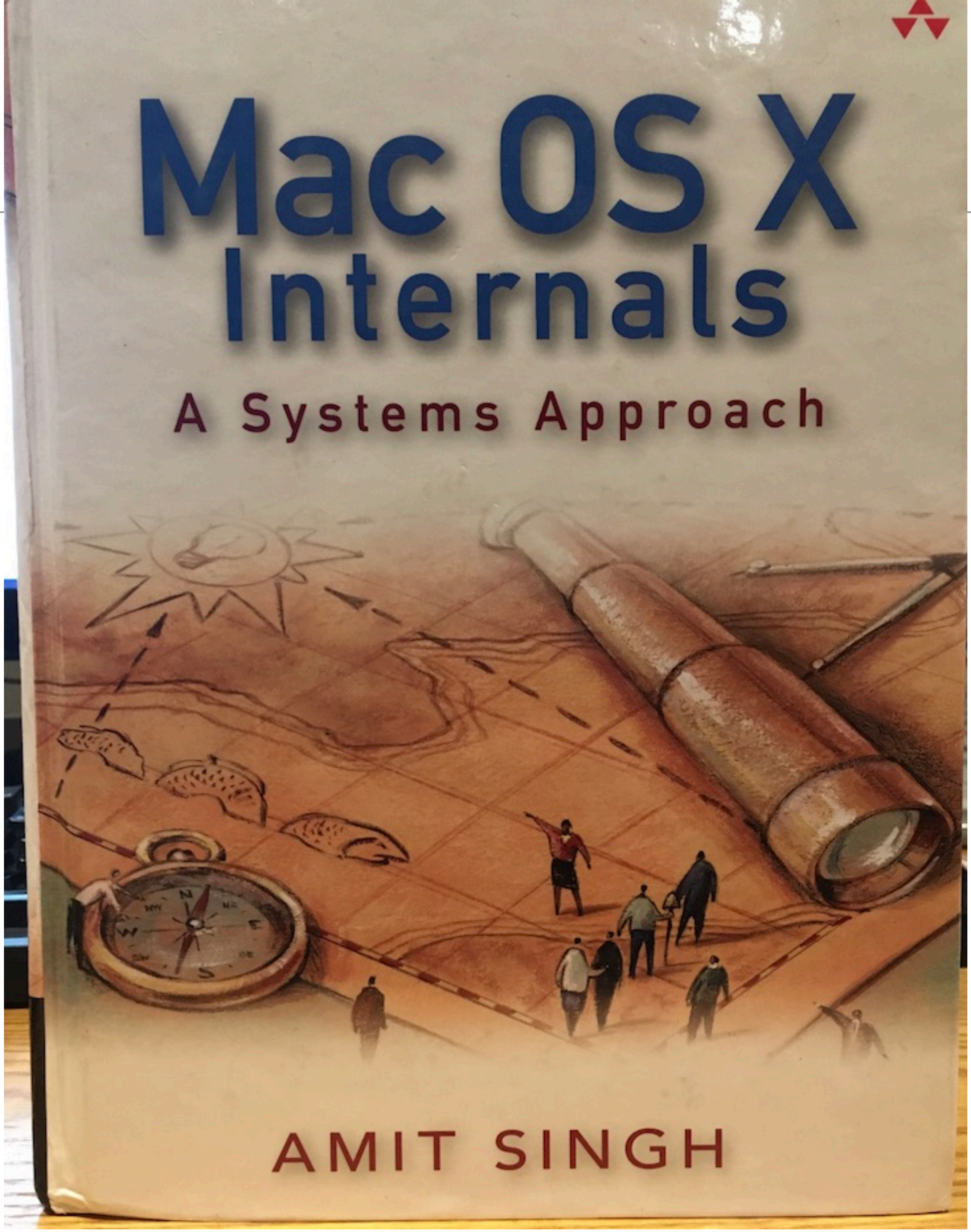
- Many organizations receiving iCloud productions don't know how to view it.
- 85% of the data can be missed.
- They can be imported directly into BlackLight.

BlackBag provides complementary support to customers.

This data has been used by BlackLight
in child exploitation cases.

- https://www.apple.com/legal/privacy/law-enforcement-guidelines-us.pdf

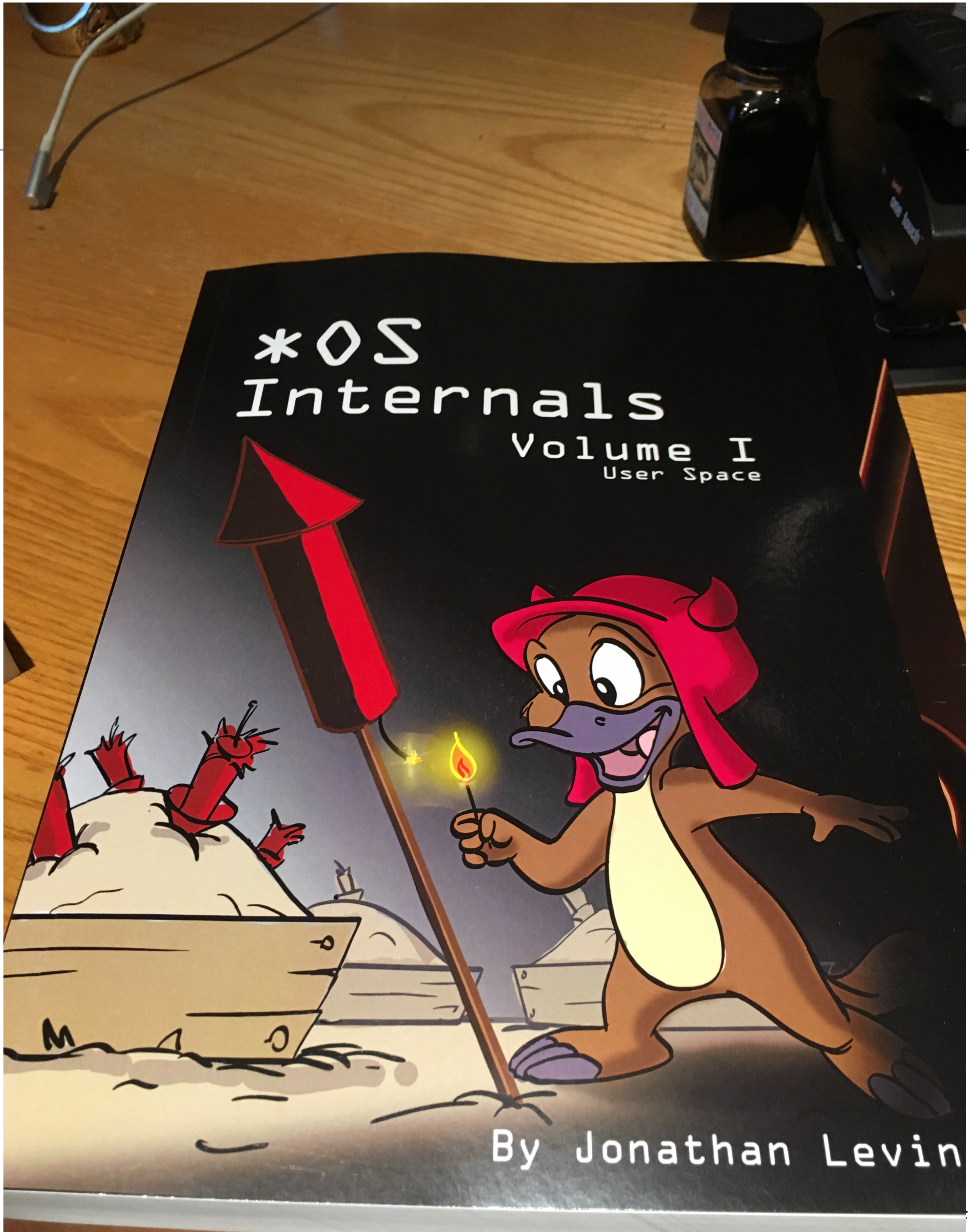
References


Mac OS X
Internals
A Systems Approach
AMIT SINGH

Mac OS X Internals — A Systems Approach
Amit Singh, 2007 • 1632 pages. http://osxbook.com

Key chapters:

- Open Firmware and Boot loader
  - —*http://osxbook.com/book/bonus/ancient/whatismacosx/arch_boot.html*
  - —*https://en.wikipedia.org/wiki/BootX_(Apple)*
  - —*https://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface*
- Kernel and User-Level Startup (180 pages)
- Processes (150 pages)
- Memory
- Interprocess Communication
- File Systems (HFS, ISO 9660, MS-DOS, NTS, UDF< UFS, AFP, FTP, NFS, SMB/CIFS, WebDAV, cddafs, deadfs, devfs, fdesc, specks and fifofs, synthfs, union, volfs)
- HFS+ File System (111 pages)


*OS
Internals
Volume I
User Space
By Jonathan Levin

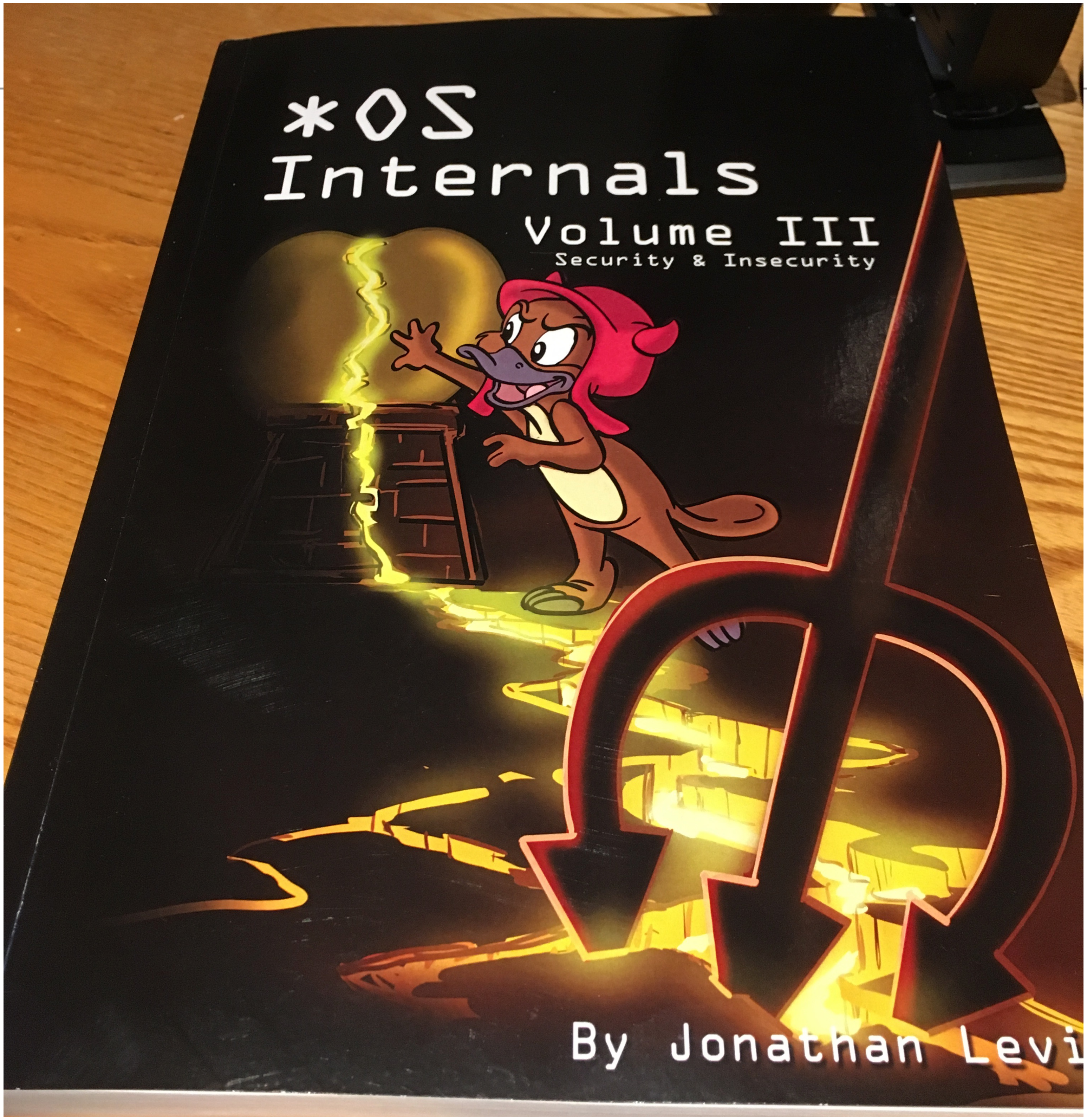
*OS Internals Volume I: User Space
Jonathan Levin • 515 pages • © 2017, 2018, 2019

Starts where Mac OS X internal Stops

- “Darwinism” — NeXTSTEP, MacOS, iOS, TvOS, WatchOS, eOS/BridgeOS, iDevice simulators
- Architecture of *OS
- *OS Filesystems
- UX and System Services — FSEvents, SpotLight, QuickLook, Duet, Printing, Siri, Voice Control, User Interface

Other chapters:

- Application Services
- Mach-O File Format (Fat Binaries)
- dyld internals
- Processes, Threads and the Grand Dispatcher
- Memory
- CFRun - RunLoopRun: — Objective-C and Swift
- Mach IPC
- LaunchD
- Process Tracing and Debugging
- Networking


*OS
Internals
Volume III
Security & Insecurity
By Jonathan Levi

*OS Internals Volume III: Security & Insecurity
Jonathan Levin • 516 pages • © 2016, 2018

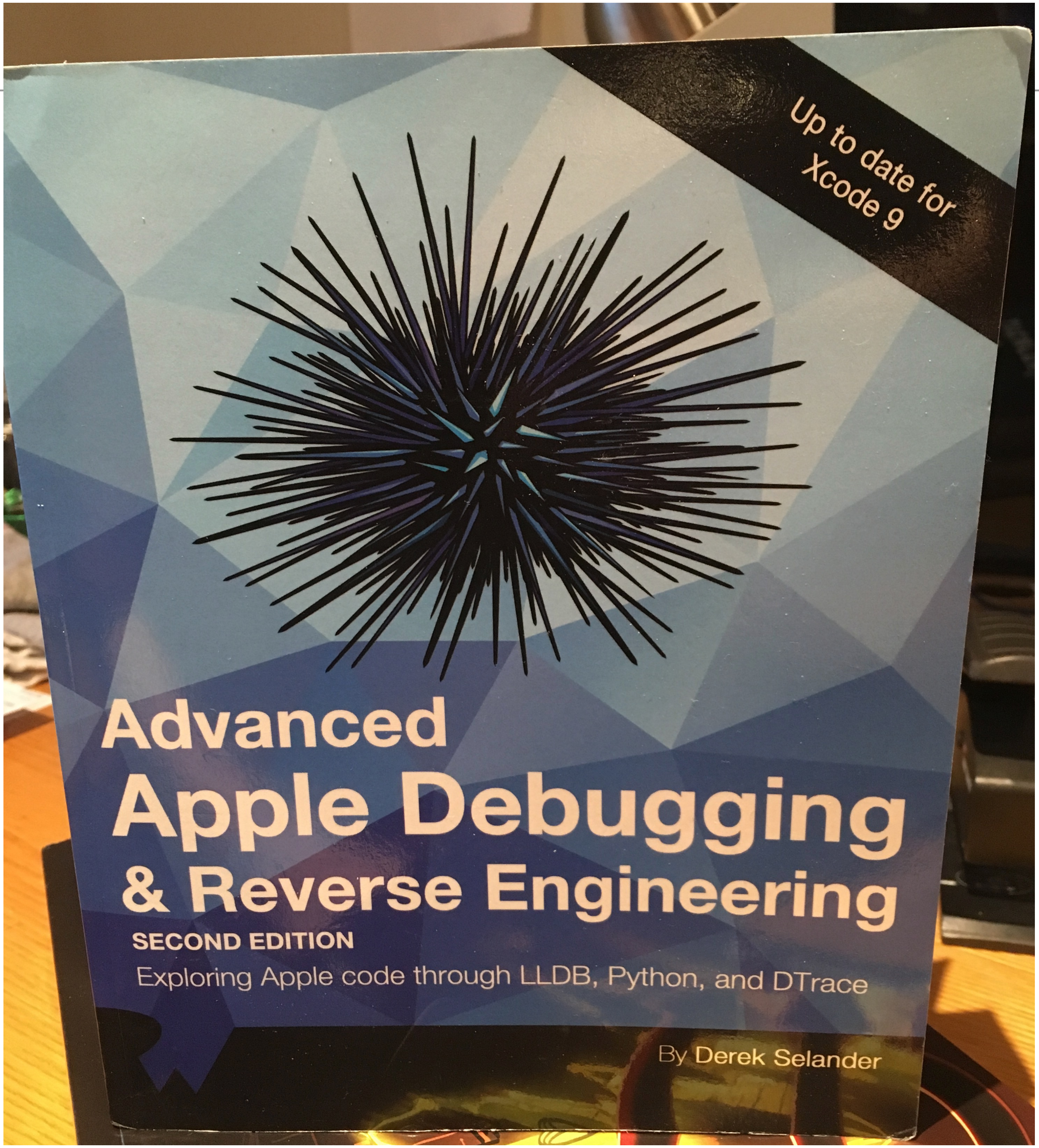
Part I: Defensive Techniques

- Authentication
- Auditing
- Authorization - KAuth
- Mandatory Access Control Framework
- Code Signing
- Software Restrictions
- AppleMobileFileIntegrity
- Sandboxing
- System Integrity Protection
- Privacy
- Data Protection

Part II: Vulnerabilities and Exploitation

- MacOS: Classic vulnerabilities
- iOS Jailbreaking
- evasi0n
- evasi0n 7
- Pangu Axe
- XuanYuan Sword
- TaiG
- Taig
- Pangu 9
- Pangu 9.3
- Pegasus
- Phoenix
- mach_portal
- Yalu
- async_wake

• MacOS Hardening Guide • Darwin 18 Changes


Up to date for Xcode 9
Advanced
Apple Debugging
& Reverse Engineering
SECOND EDITION
Exploring Apple code through LLDB, Python, and DTrace
By Derek Selander

Mor Resources

Mac 4N6 Resources

- http://bit.ly/mac4n6s

CFRS 768 Labs on Github

- https://github.com/simsong/cfrs764-spring2019