The DFIR Practitioner's Guide to the Research & Development Process

Presented by:
Joe T. Sylve, Ph.D.
Director of Research & Development
#whoami

- Former Practitioner
  - Managing Partner, 504ENSICS Labs
- Researcher
  - Director of R&D, BlackBag Technologies
- Educator
  - Adjunct Professor, University of New Orleans
- Organizer
  - NOLASec
  - BSidesNOLA
  - DFRWS
- CV
  - [https://goo.gl/4etPVx](https://goo.gl/4etPVx)
Motivation

• There’s an curious lack of overlap among the attendees of research-focused conferences like DFRWS and practitioner-focused conferences like SANS DFIR Summit

• This is problematic
  • In order for research to be useful, it needs to be informed by the needs of practitioners
  • If practitioners and tool vendors aren’t aware of the latest research, then that research may never be applied in any significant way
  • We all represent a small (but growing) community that can only benefit from the sharing of ideas the comes when we’re not all sitting in our own sub-communities

• There are common misconceptions about research
  • My job is not any harder than your job
  • My academic qualifications aren’t what make me a good researcher
  • Our sets of skills (and even tools!) largely overlap
Motivation

• There’s an curious lack of overlap among the attendance of research-focused conferences like DFRWS and practitioner-focused conferences like SANS DFIR Summit.
• This is problematic.
  • In order for research to be useful, it needs to be informed by the needs of practitioners.
  • If practitioners aren’t aware of the latest research, then that research may never be applied in a significant way.
• We all represent a small (but growing) community that can only benefit from the sharing of ideas.
• There are common misconceptions about research.
  • My job is not any harder than your job.
  • My academic qualifications aren’t what make me a good researcher.
  • Our sets of skills (and even tools!) largely overlap.
Overview

• Discuss the Research and Development process
• Walk through a real-world example
• Drop some unreleased info about APFS snapshots in the process

My goal is that by the end of this talk, you will be interested in becoming more involved in the research community
The Research & Development Process

- Problem Identification
- Data Collection and Preliminary Analysis
- Analysis
- Identification of Future Research
- Publishing and Validation
The Research & Development Process

- Problem Identification
- Data Collection and Preliminary Analysis
- Analysis
- Identification of Future Research
- Publishing and Validation
Problem Identification
Identifying a Problem

• The first step of any good scientific process is to identify a problem that needs solving
• Often, the best problem candidates are ones that you’re personally invested in
  • A question you or a client need answering
  • Something that piques your interest or curiosity
  • (A little bit of obsession goes a long way)
• The best applied research is often limited in scope
  • Set attainable goals (and be flexible as you learn new things)
  • Even a small amount of new knowledge can be useful to you or others
  • Don’t try to “solve forensics”
APFS Snapshots

• Apple’s APFS file system contains a capability called “snapshots”, which allow preserving point-in-time states of the file system that can be mounted in order to recover deleted or overwritten files.

• During our previous work with APFS, we observed that when Time Machine is enabled, macOS automatically takes snapshots of system volumes
  • These snapshots appear to stick around for about 24 hours
  • Some snapshots are “dataless” which means only the blocks of data the contain the file system metadata are guaranteed to be preserved (not the file extents)
  • Dataless snapshots seem to stick around longer than the 24 hours and usually are the oldest of the snapshots available, but they’re not always present

• We’ve often been asked about these “dataless” snapshots, but could not find any public mention of them in either Apple’s documentation or public literature
What Apple Says...

“Time Machine saves one snapshot of your startup disk approximately every hour, and keeps it for 24 hours. It keeps an additional snapshot of your last successful Time Machine backup until space is needed. And in High Sierra or later, another snapshot is saved before installing any macOS update.”

Our Open Research Question

What causes snapshots to become “dataless” and what can we learn about their lifetime?
Data Collection and Preliminary Analysis
The Hunt is On!

- Since we’re interested in learning behavior related to APFS, the logical place to start would be to analyze the APFS KEXT
- macOS KEXT’s are nice to analyze, because the majority of symbols are exported publicly, so we get function names and sometimes parameter information to work with
- IDA Pro only shows one function that contains the word “dataless” 😊
Dead Ends Will Happen...

- Analyzing this function doesn’t help us answer our question, but it does give us some information we didn’t know.

- Files in “dataless” snapshots that don’t have file content preserved can be identified by a compression type value of 0x80000001 in their com.apple.decmpfs xattr.
Strings Can Often be Very Useful...

- Functions can be inlined by compilers, which means their function names won’t be exposed by themselves.
- Apple tends to print function names in debug logs when something goes wrong.
- We can sometimes use this information to find the inlined function content.
- `handle_snap_make_database` sounds interested in looks like a possible function name.
- The string is only referenced once in an ioctl handling function.
Don’t Get Lost in the Weeds...

• ioctl functions allow communication between user and kernel space

• They can by used to ask the kernel to perform specific tasks our or behalf by providing a numerical identifier and some parameters

• This ioctl handler is VERY large and complicated, but if we start at our cross-referenced string and work our way up through the control flow we come across this check

• This looks like a great candidate for an ioctl identifier!
Expanding our Search

• The only time our handle_make_snap_dataless functionality is called by the KEXT is in the ioctl handler, so the KEXT itself doesn’t seem to control when snapshots become dataless
• We need to find userland components that trigger this functionality
• If this were malware analysis what would you do next?
YARA to the Rescue

- YARA is a great tool to find strings or binary patterns in executables
- Our ioctl identifier should obviously be embedded in any executable code that triggers our functionality from userland
- Let’s write a signature and do a search!
Identifying a Framework

• Searching the System directory only resulted in one hit that made sense
  • /System/Library/PrivateFrameworks/TimeMachine.framework/Versions/A/TimeMachine
• Analyzing that framework located a single function that calls the ioctl

```
rule ioctl_search
{
    strings:
    $le = {14 4A 08 C0}

    condition:
    $le

}
```
Identifying the Binary

- Now we need to find a system binary that imports the TimeMachine framework and calls the TMDisk makeDatalessAPFSSnapshot function.
- Another YARA search gives us only a single viable hit:
  - `/System/Library/CoreServices/backupd.bundle/Contents/Resources/backupd-helper`
- The backupd-helper is a daemon that is launched at startup and is responsible for handling TimeMachine related tasks.
- We now have our target that we need to analyze!
Types of Snapshots

- There are three classes of snapshots
  - Stable
  - Reference
  - Dataless
- Snapshots can be assigned to multiple classes and are assigned a value from 1-5 depending on their classifications

* Value of 5 is used if name doesn’t start with "com.apple.TimeMachine."
Purge Actions

• There are four possible actions for a snapshot during a purge
  • Skip Purging
  • Make Dataless
  • Unmount & Delete
  • Force Unmount & Delete

• The given action is determined based on the type of the snapshot and an “urgency” level of 1-4
Purge Actions

- **Urgency Level 1**
  - Force Unmount & Delete: Type 1

- **Urgency Level 2**
  - Unmount & Delete: Type 5
  - Force Unmount & Delete: Type 1

- **Urgency Level 3**
  - Make Dataless: Type 3
  - Unmount & Delete: Type 5
  - Force Unmount & Delete: Types 1 & 4

- **Urgency Level 4**
  - Force Unmount & Delete: All Types

Stable

Reference

Dataless

4

3

2

1*
Purge Exceptions

- There are a few checks done prior to the urgency-based checks that can result in certain snapshots always being skipped.
- Snapshots with certain prefixes are exempt from purging:
  - com.apple.CDM
  - com.apple.test.
  - com.faronics.deepfreeze.
- In addition snapshots of type 5 seem to never actually be purged:
  - This check may have been added as an afterthought or could be a bug in the code.
Identification of Future Research
Unanswered Questions

• Sometimes research raises more questions than it answers
  • This is good job security for a researcher!

• It’s good to note these unanswered questions so that they may be used to spawn future research efforts by you or others

• In this case there are a number of unanswered questions
  • What gives a snapshot a “reference” or “stable” classification?
  • Is this possibly related to the last successful TM backup?
  • What controls the “urgency” parameter?
  • Does our observed behavior match our static analysis?
  • Is this behavior different on other versions of macOS?
Unanswered Questions

• Sometimes research raises more questions than it answers
  • This is good job security for a researcher!

• It’s good to note these unanswered questions so that they may be used to spawn future research efforts by you or others

• In this case there are a number of unanswered questions
  • What gives a snapshot a “reference” or “stable” classification?
  • Is this possibly related to the last successful TM backup?
  • What controls the “urgency” parameter?
  • Does our observed behavior match our static analysis?
  • Is this behavior different on other versions of macOS?

That just raises further questions!
Publishing and Validation
Sharing is Caring

• Knowledge is useless unless it's shared
• This allows others to validate your research and build upon it
  • You also get tons of #DFIR street cred
• There are any number of ways to share your results with the world
  • Each has different barriers to access & reach

• Some examples
  • Twitter #DFIR
  • Blog Posts
  • Forensic Lunch
  • DFIR Rapid Review
  • Conference Presentations
  • Academic Publications
Questions?

Follow us on social for more webinars, blogs, product releases, tips and tricks and giveaways!