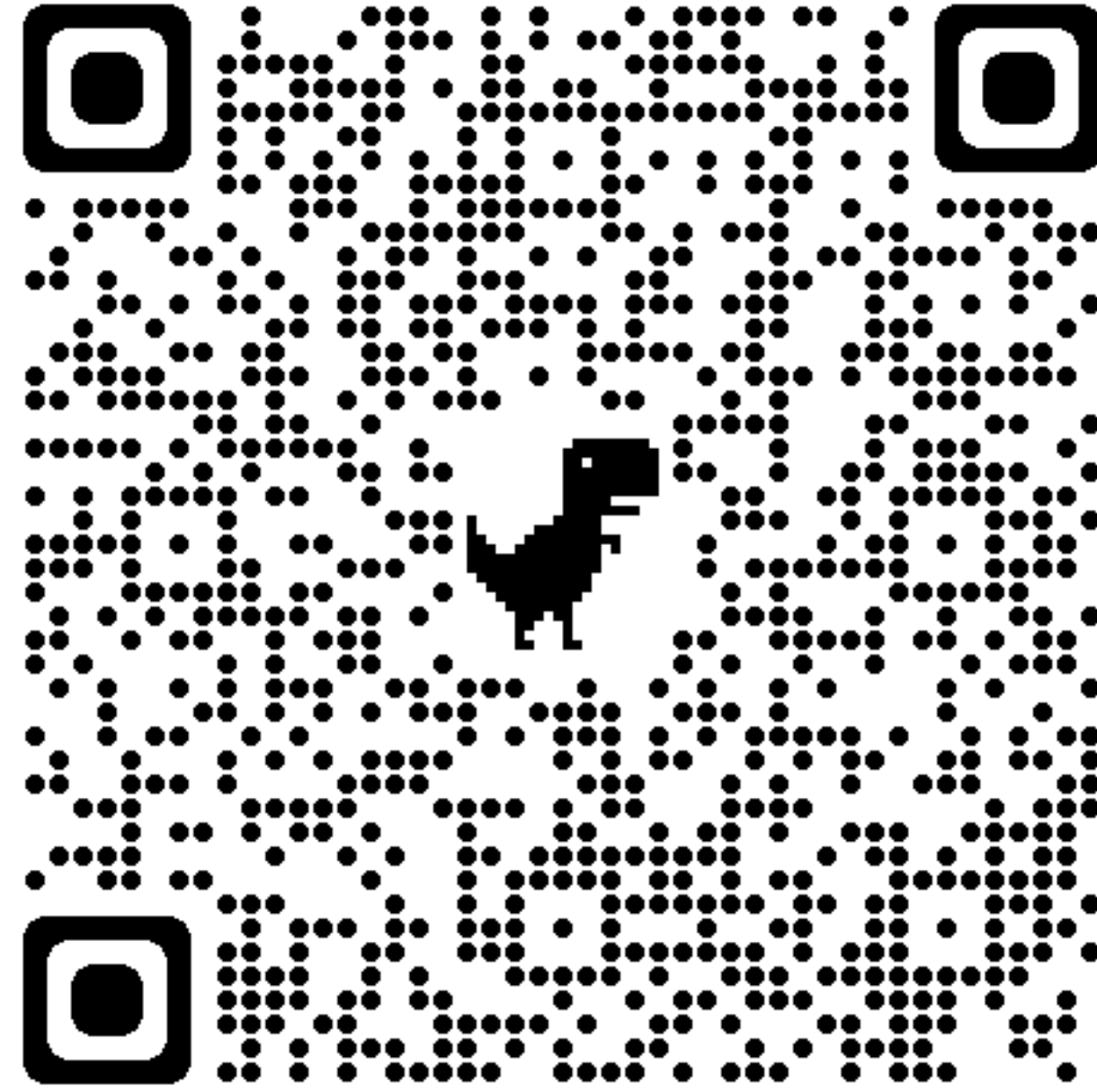


Practical Automation for Apple Devices

**Or How I Stopped Worrying and
Learned to Love our Task Server**



bit.ly/PracticalAutomation

Agenda

- Introduction
- What is automation?
- Why is it worthwhile? **bit.ly/PracticalAutomation**
- Four steps to automating a project
- Examples
- A tour of our task server
- A crash course in Jamf scripting and LaunchDaemons

bit.ly/PracticalAutomation

Introduction

Who am I anyway?

```
%whoami  
rebecca.latimer
```


The Tech Lady

- District Technology Technician
- Hibbing Public School District
- 3000 iPads, 300 Macs
- Not even remotely an expert
- Automation enthusiast



My First Task When I Got Hired

1. Find all users in Jamf that didn't have an iPad or computer assigned
2. Delete them
3. One by one by one by one by one
4. BY ONE BY ONE BY ONE BY ONE
5. (There were 3000 of them)
6. Go crazy
7. Decide to find a better way

What is Automation?

**Automation just means doing things
automatically**



Automation Can Mean Different Things

- Any time you move from a manual process to an automatic one, it is automation!
- Some examples:
 - Using an MDM
 - Using a service to sync rosters
 - Scheduling app updates
 - Getting automatic reports emailed to you
 - Scripting repetitive tasks

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE
EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

		HOW OFTEN YOU DO THE TASK					
		50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
HOW MUCH TIME YOU SHAVE OFF	1 SECOND	DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
	5 SECONDS	DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
	30 SECONDS	4 WEEKS	DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
	1 MINUTE	8 WEEKS	DAYS	DAY	4 HOURS	1 HOUR	5 MINUTES
	5 MINUTES	9 MONTHS	4 WEEKS	DAYS	21 HOURS	5 HOURS	25 MINUTES
	30 MINUTES		6 MONTHS	5 WEEKS	DAYS	DAY	2 HOURS
	1 HOUR		10 MONTHS	2 MONTHS	DAYS	DAYS	5 HOURS
	6 HOURS				2 MONTHS	2 WEEKS	DAY
	DAY					8 WEEKS	DAYS

“Any daily work task that takes 5 minutes will cost over 20 hours a year, or over half of a work week.

Even if it takes 20 hours to automate that daily 5 minute task, the automation will break even in a year.”

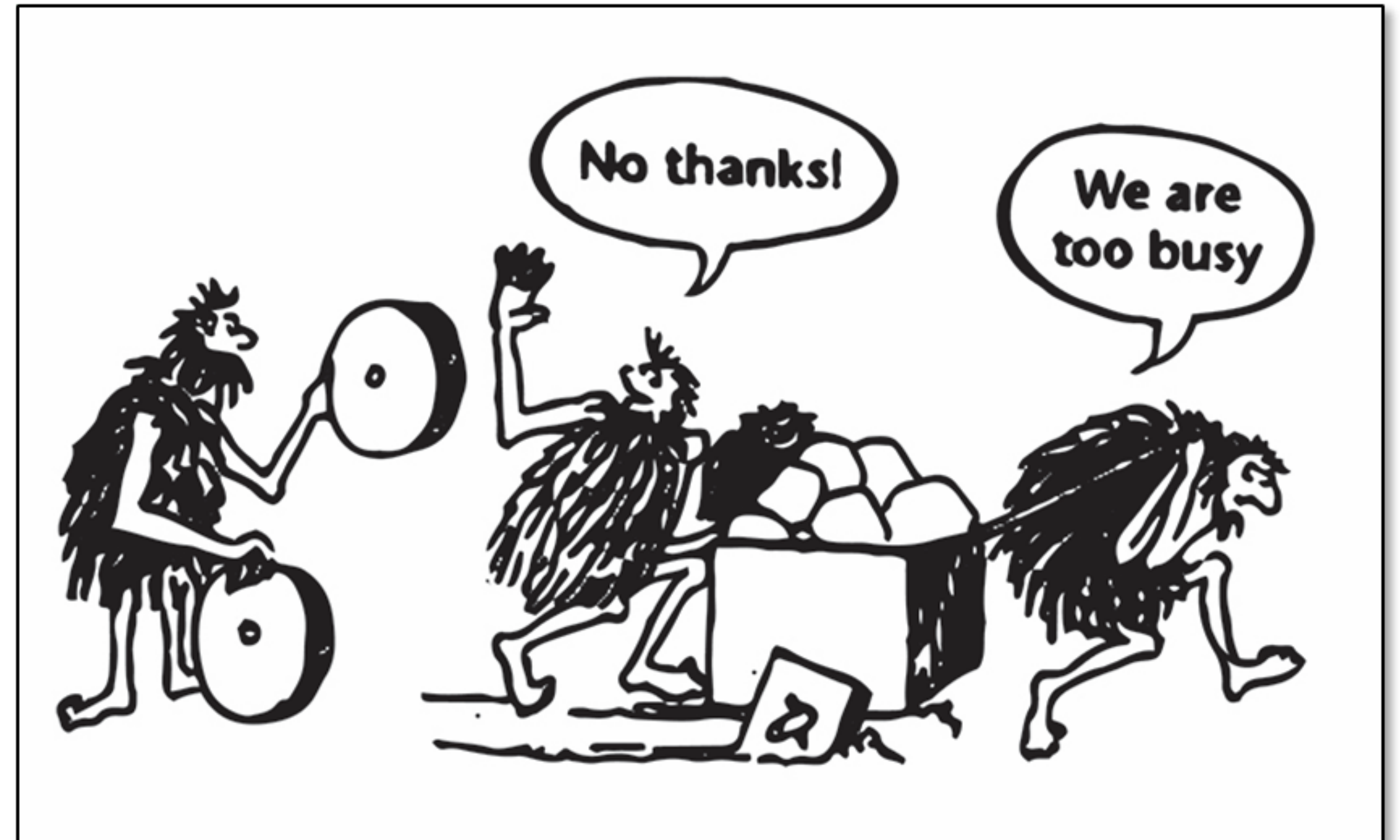
Anthony J. Stieber, Breaking Into Information Security

The Compound Effect

- Automation builds on itself
- Time saved while one automation is running can be used to build another
- And another and another
- Skills you learn can be applied to other projects as well
- The Jamf Pro API is a great example

Finding the Time

- We are all too busy in EDU
- Setting aside even a half hour a day helps
- Go for the low-hanging fruit first



Some Numbers (Part One)

- Example: Deleting classes from Jamf every semester
- Pre-automation: 2 hour process, twice a year
- Post-automation: 5 minute process
- Scripting time: 4 hours
- After year one, we break even on time
- After year two, those hours can be spent on other projects
- Like...

Some Numbers (Part Two)

- Example: Renaming iPads
- Pre-automation: 30 minute process, once a week
- Post-automation: 0 minute process, happens daily automatically
- Scripting time: 1 hour
- This is a great example of the compound effect
- Since I had spent so long with the Jamf API, I could confidently knock out this script quickly and efficiently
- Even better than just scripting the process, I also wrote a LaunchDaemon to run it for me!

How Are You Spending Your Time?

- In order to save time, you need to know how you are spending it
- Try tracking your time for a day or two
- The results might be eye-opening
- Some apps are great for this
- You can also go low-tech with a stopwatch



Ready to Automate?

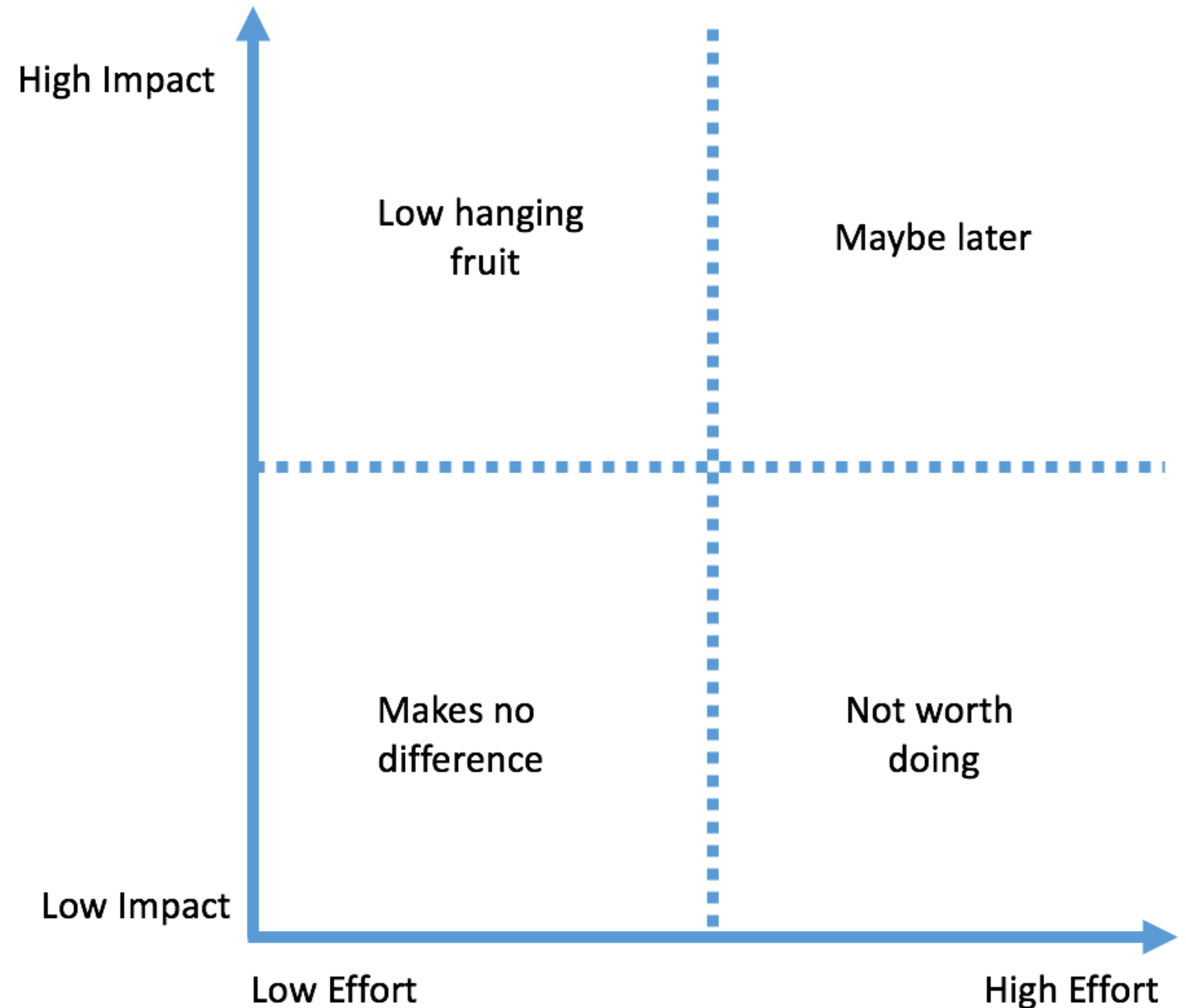


Four Steps to Automation

1. Find your lowest-hanging fruit
2. Don't reinvent the wheel
3. Implement the solution
4. Document what you have done

1 - Lowest Hanging Fruit

This is the thing that is the most important to get done but would be the easiest to automate



2- Don't Reinvent the Wheel

- If you are having this problem, it's likely that someone else has too
- Slack, Jamf-nation, and just plain Googling are all good methods
- Their solution may not be exactly right for you but it could put you on the right path



MacAdmins
macadmins.slack.com

 **jamf** | NATION

Google

🔍 rename ipads jamf api

3 - Implement

- Tweaks may be required
- Keep in mind that solutions are not one-size-fits-all
- Don't expect the world - many solutions are open source
- Make sure you understand what you are implementing



4 - Document

- Documentation takes time, but it will save you time later
- This is especially important if you have the same kind of project later
- You can also use this to help others who might be in the same position



Enjoy your free time

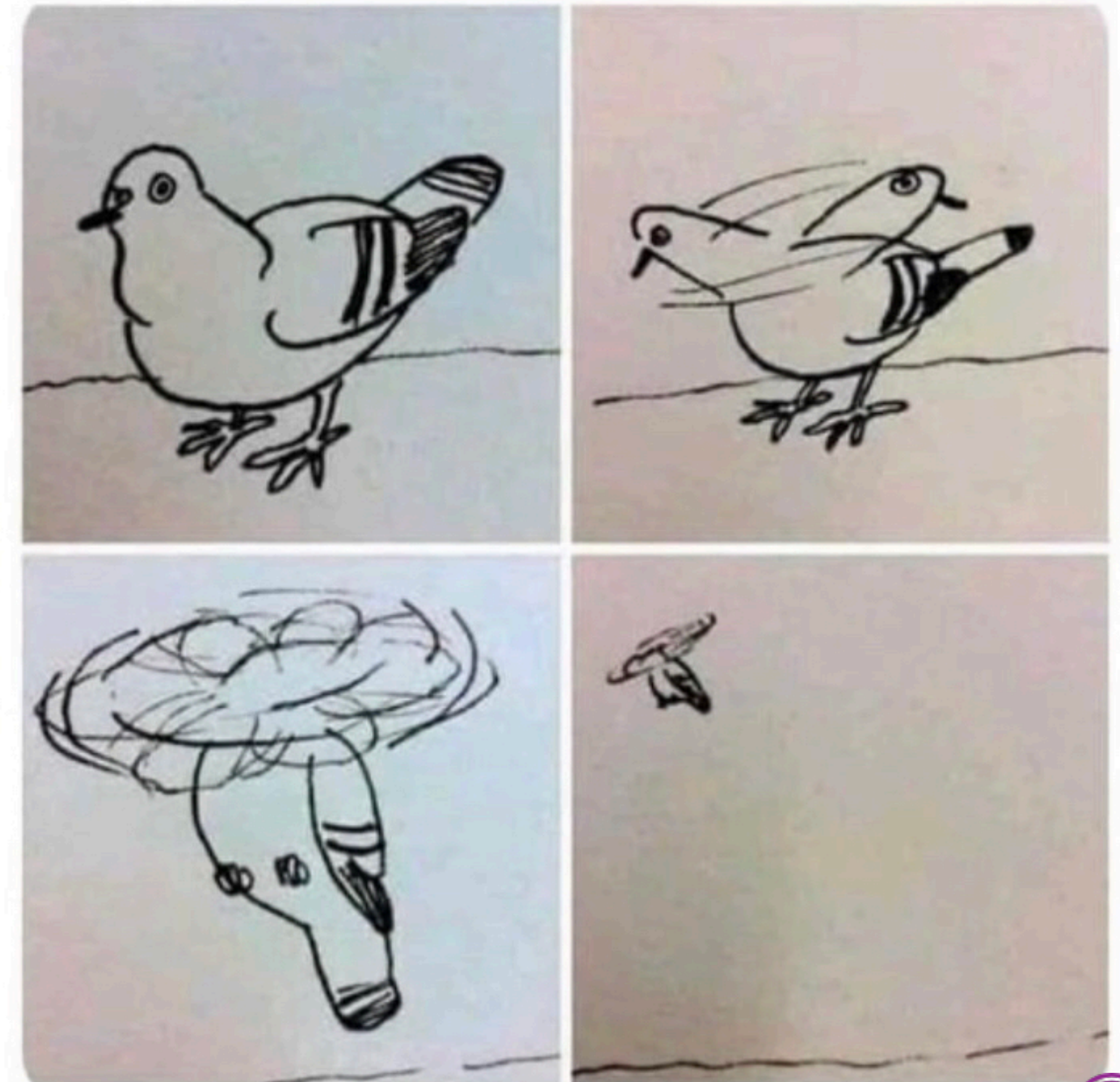
- Get a snack!
- Consider sharing with others what you have done
- Spend your newfound time looking for the next lowest-hanging fruit
- (Or just go eat some fruit)



When your program
is a complete mess,
but it does its job

Examples

(Things I have done, for better or
worse)

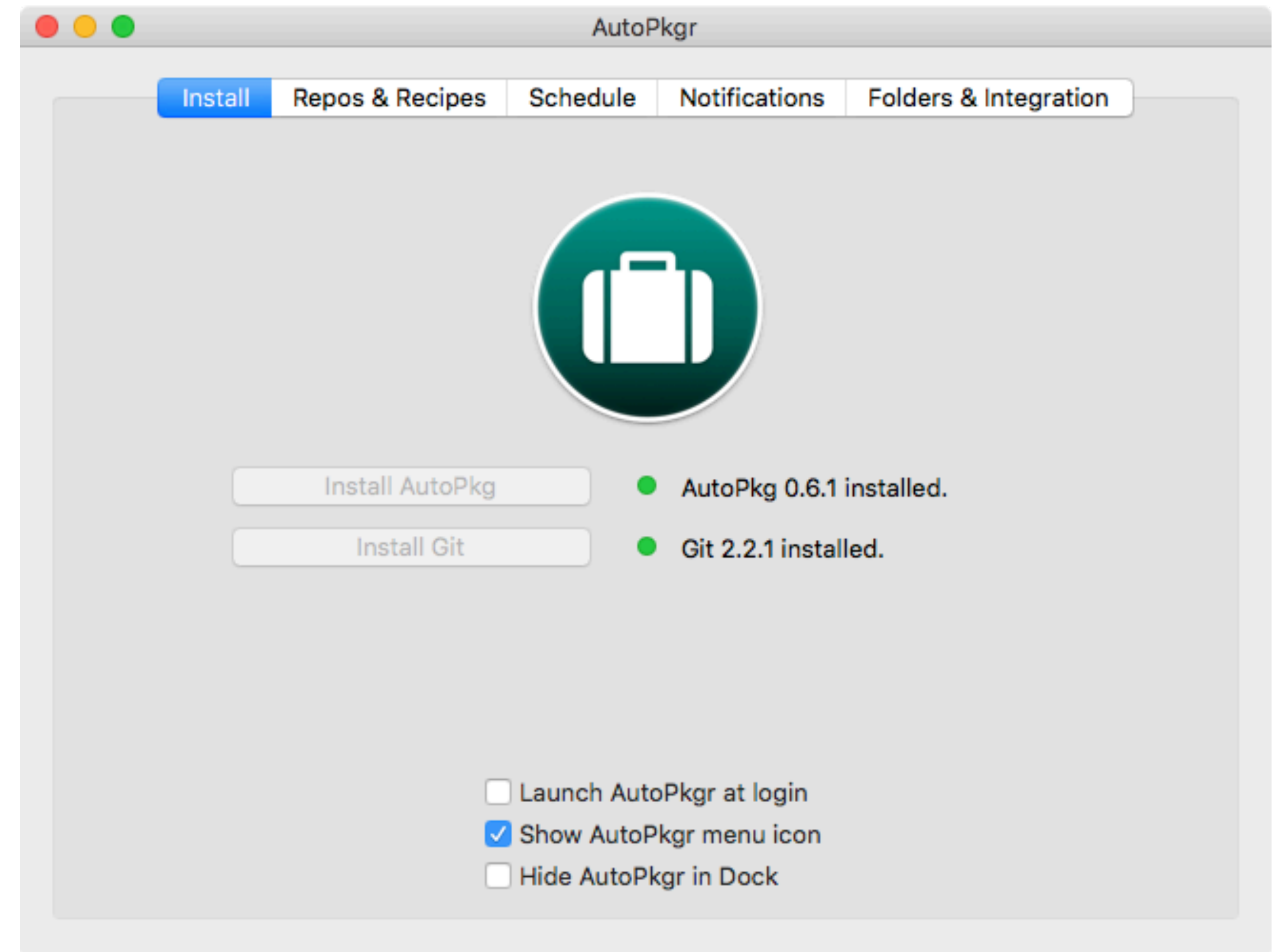


Example 1: Updating Applications

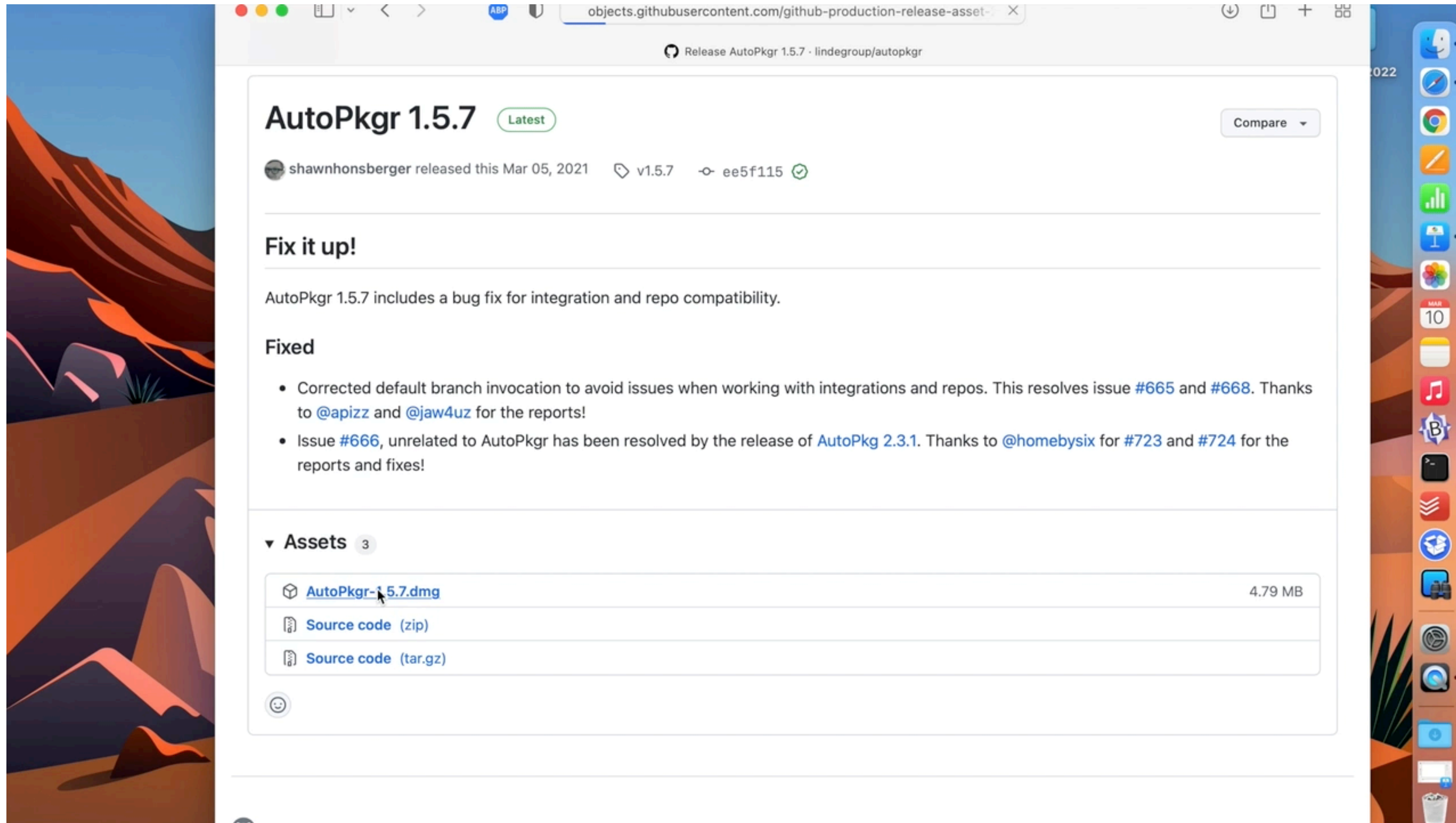
- Probably the lowest-hanging fruit out there! High impact, very low effort
- This is a solved problem: AutoPKG is a fantastic tool
- Implementing this in your own environment may require a task server to be set up
- You can use AutoPKG with JSSimporter if you use Jamf to make your life even easier

AutoPKG (and AutoPKGr)

Your new secret weapons



Installing AutoPKGr



Example 2: Getting Updated Reports

- Very low-hanging fruit - effort is very low (built in) and impact is high
- This exists but there isn't great documentation for it
- Just check a few boxes
- Reports can go to you daily or weekly or whatever you like
- If it ISN'T built in, you may be able to use an API to get the information that you want

Mobile Devices : Advanced Mobile Device Search

← 9 Active Students Not on iOS 15.2

Search

Criteria

Display

Reports

File Format File format to export the report

Comma-Separated Values (.csv) ▼

Inventory Item Choose the inventory item on which to base your results

Devices ▼

Download Report

Email Reporting

Configure Email Reports

☒ Schedule automatic email reports


Frequency Frequency at which the reports are calculated and sent

☒ Daily

☐ Days per week

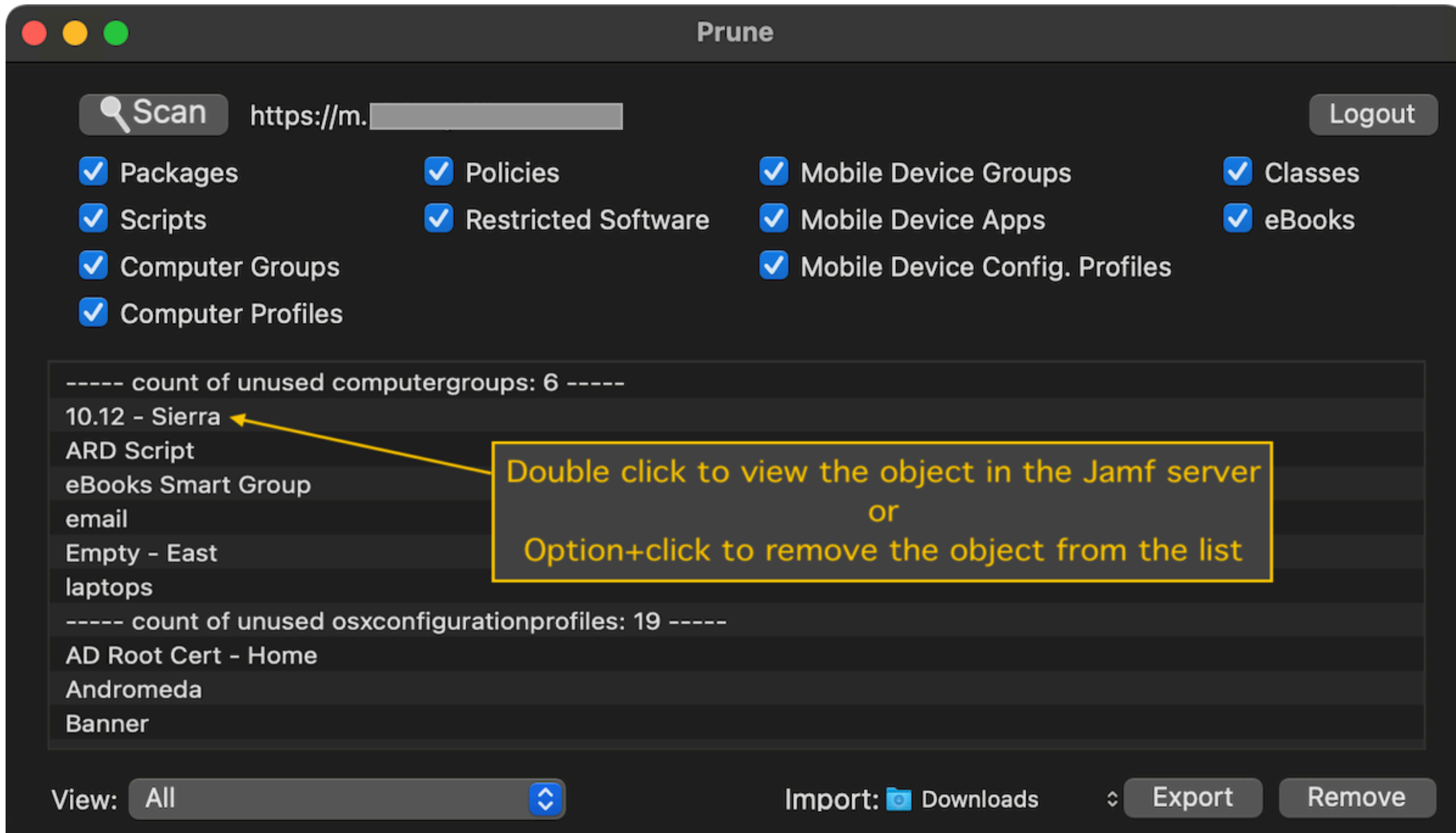
☐ 1st ▼ day of every month

☐ 1st ▼ Monday ▼ of every month

At 9 ▼ 00 ▼ a.m. ▼  Time is displayed in Coordinated Universal Time (UTC)

Example 3: Cleaning up Policies, Profiles, etc.

- The impact is a bit lower on this one (except when it isn't) but the effort is extremely low
- Solved problem! Use Prune
- <https://github.com/BIG-RAT/Prune>



Example 4: Deleting ASM Classes

- This qualifies as low-hanging fruit because of the time impact - deleting classes one by one is awful
- There were some solutions out there but I ended up mostly writing my own - you can benefit from my struggle!
- Implementing this is as easy as running the script locally on your own computer
- Once you understand how the API works, you can modify it to meet your own needs

Deleting Classes

- <https://github.com/GingerScripting/jamf-pro-api/blob/main/deleteClasses.sh>
- Custom script that I wrote
- Searches for criteria that you specify (such as S1 for Semester 1 classes)
- Creates a CSV on your desktop you can review before deleting
- Use or modify to your heart's content!

Example 5: Renaming iPads

- Pretty low-hanging fruit - keeps our inventory up to date and makes it easier to find lost iPads
- Again, some good options out there but ultimately had to write my own script
- Implementing this one takes a little work if you want to go as far as I did
- This was a good exercise in how LaunchDaemons work

Renaming iPads

- <https://github.com/GingerScripting/iPadNightlyRename>
- This repo contains both a script and a LaunchDaemon
- Run the renameDevices.sh script to rename iPads on demand
- Install the LaunchDaemon if you want to automate the process even further

A Tour of Our Task Server

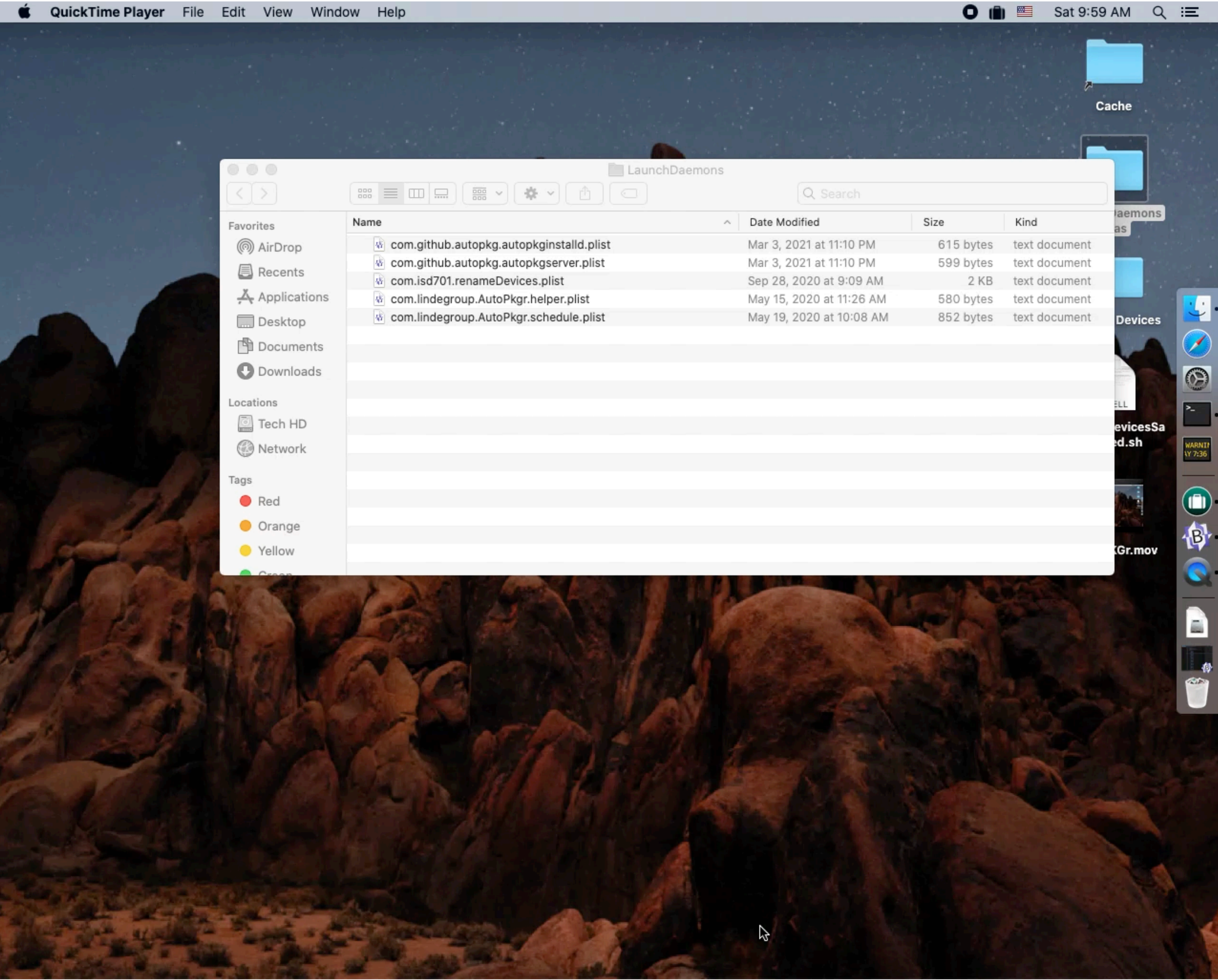
Hopefully not a three-hour tour



What Is A Task Server?

- Any computer can work for this - virtual or real
- If you have moved a lot of services to the cloud, you can use your old devices for this
- Doesn't have to be anything special - just a computer that has an internet connection that is always on
- You COULD run this on your own machine but it is not recommended





A Crash Course in the Jamf Pro API

- Find documentation at [your Jamf Pro instance]/api
- The jamf-api channel on MacAdmins Slack is a great resource
- Best way to start is to find something you like and tweak it
- Authentication is tricky - basic is okay IF you are only running a script on a device that YOU control
- Lets look at an example of one of my scripts

Logging In

```
#!/bin/bash

#sends the Enable Lost Mode command to all devices in an advanced search

#Add your credentials and Jamf Pro URL here if you don't want to be prompted for them. This is also necessary if
you are running the script as root (jamf or LaunchDaemon)
jssUser=
jssPassword=
jssURL=

#You can also uncomment this line if you want the script to read which jamf server the computer it is running on
connects to.
#jssURL=$(/usr/bin/defaults read ~/Library/Preferences/com.jamfsoftware.jss.plist url)

if [ -z $jssURL ]; then
    echo "Please enter the JSS URL:"
    read -r jssURL
fi

if [ -z $jssUser ]; then
    echo "Please enter your JSS username:"
    read -r jssUser
fi

if [ -z $jssPassword ]; then
    echo "Please enter JSS password for account: $jssUser:"
    read -r -s jssPassword
fi

echo "Logging in to $jssURL as $jssUser"
```


Xpath and Device Numbers

```
xpath() {  
    # the xpath tool changes in Big Sur  
    if [[ $(sw_vers -buildVersion) > "20A" ]]; then  
        /usr/bin/xpath -e "$@"  
    else  
        /usr/bin/xpath "$@"  
    fi  
}  
  
echo "Please enter the ID number for the advanced search that you would like to send the Lost Mode  
command to:"  
read -r searchID  
  
deviceNumbers=$(curl -X GET -H "Accept: application/xml" -s -u "${jssUser}":"${jssPassword}" \\\n${jssURL}/JSSResource/advancedmobiledevicesearches/id/"$searchID" | xpath "//mobile_device//id" \\\n2> /dev/null | awk -F'</?id>' '{for(i=2;i<=NF;i++) print $i}'))
```

Creating the Function

```
setLostMode() {  
  
goodXML="<mobile_device_command>  
  <general>  
    <command>EnableLostMode</command>  
    <lost_mode_message>[ENTER MESSAGE HERE]</lost_mode_message>  
    <lost_mode_phone>[ENTER PHONE NUMBER HERE]</lost_mode_phone>  
    <lost_mode_footnote>[ENTER FOOTNOTE HERE]</lost_mode_footnote>  
    <always_enforce_lost_mode>true</always_enforce_lost_mode>  
    <lost_mode_with_sound>true</lost_mode_with_sound>  
  </general>  
  <mobile_devices>  
    <mobile_device>  
      <id>$device</id>  
    </mobile_device>  
  </mobile_devices>  
</mobile_device_command>"  
  
curl -X POST \  
  ${jssURL}/JSSResource/mobiledevicecommands/command/EnableLostMode \  
  --user "$jssUser":"$jssPassword" \  
  --header "Content-Type: text/xml" \  
  --request POST \  
  --data "$goodXML"  
}
```

Looping Through

```
for device in "${deviceNumbers[@]}; do
    setLostMode
    echo "Lost mode set on device number $device"
done
```


LaunchDaemons

- User-created LaunchDaemons usually live in /Library/LaunchDaemons
- Format should be familiar - just a plist
- LaunchDaemons can be used to schedule tasks
- Creating a custom LaunchDaemon is easy
- <https://zerolaunched.herokuapp.com>

LaunchDaemon Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Label</key>
    <string>com.isd701.scheduler</string>
    <key>Program</key>
    <array>
    <string>/usr/local/policyReport.sh</string>
    </array>
    <key>StartCalendarInterval</key>
    <dict>
    <key>Hour</key>
    <integer>2</integer>
    <key>Weekday</key>
    <integer>0</integer>
    </dict>
</dict>
</plist>
```

In Summary

- Find what you spend your time on
- Automate the easiest and most impactful projects (lowest-hanging fruit) first
- Don't reinvent the wheel - see if others have done the work first
- Document what you have done
- Spend your free time learning new things

More Resources

- Companion post - Same as bitly link on bottom of pages - <https://gingerscripting.com/practical-automation-for-apple-devices/>
- Mac Admins Slack - <https://www.macadmins.org> (I am rebelati there)
- <https://github.com/gingerscripting> - My personal scripts
- <https://gingerscripting.com> - My blog
- <https://twitter.com/gingerscripting> - Twitter

Thank You!

