

Firmware Password Manager 2.5

Todd McDaniel
Marriott Library, ITS



J. Willard Marriott Library

THE UNIVERSITY OF UTAH

THE UNIVERSITY OF UTAH

A quick note about SCL Jamf Tools...

- LDAP logins implemented
- Cargo ship sped up significantly (~3x with 2 cores)

```
tmp_policies = pool.map(fetch_parse_policy, policy_id_list)
```

- Github updated soon!

A quicker note about SCL Jamf EA collection

- 8 Extension Attributes

Bluetooth device battery level

Disk free space reporter

Estimated date of manufacture

Estimated Age

SMARTmon check

Time Machine Status

Attached Displays

External Encrypted Disks

- https://github.com/univ-of-utah-marriott-library-apple/scl_jamf_extension_attribute_collection

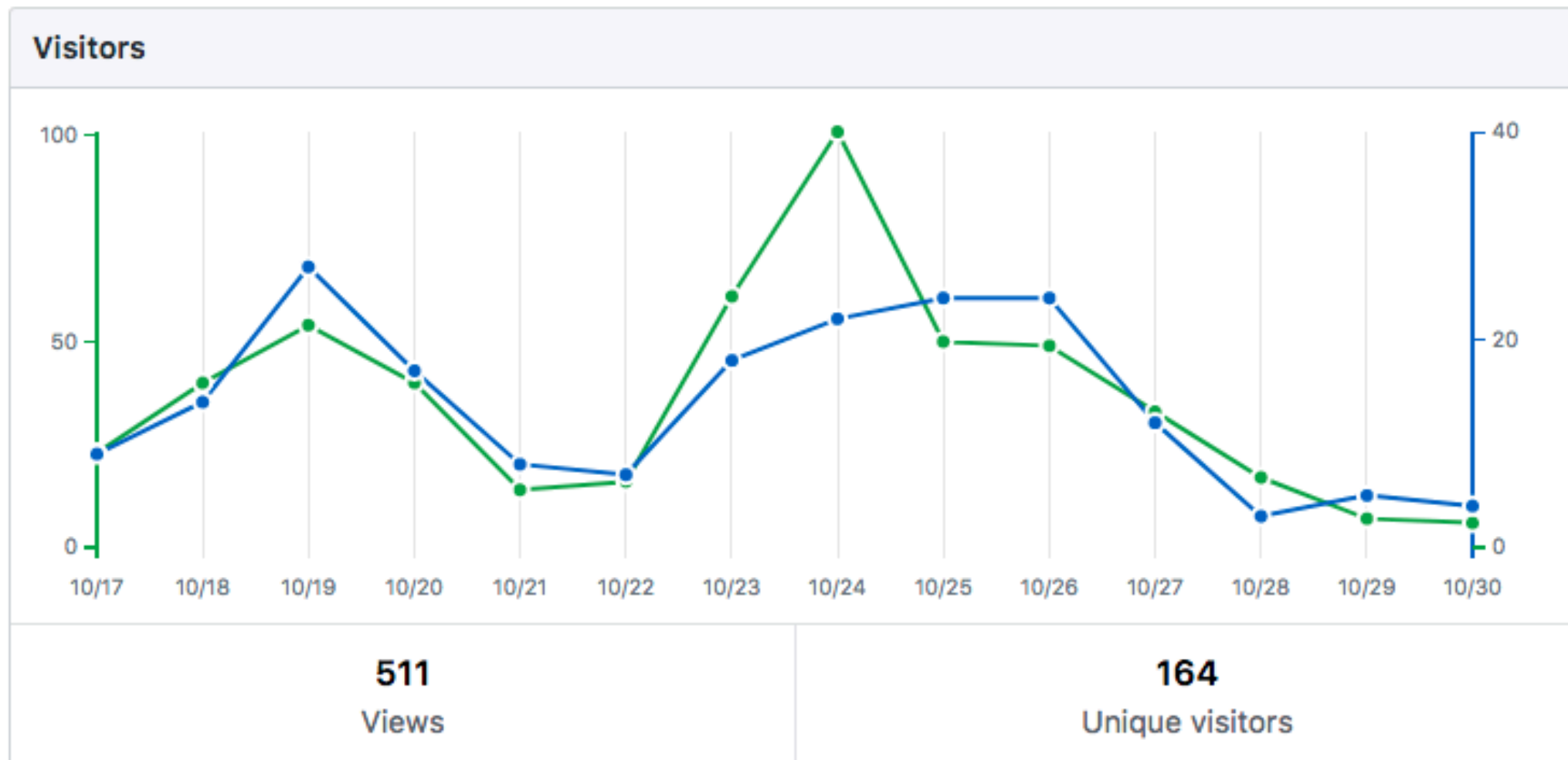
Intro to FWPM

```
#####  
# firmware_password_manager.sh  
#  
# This script uses Apple's setregproptool to automate changing the firmware  
# password.  
#  
#  
# 1.0.0    2014.08.20  Initial version. tjm  
# 1.1.0    2015.06.03  Logic updates. tjm  
#                               Yosemite  
#                               Error handling  
#  
#####
```

Version 1.0

```
# firmware_password_manager.py #####
#
# A Python script to help Macintosh administrators manage the firmware passwords
# of their computers.
#
#
# 2.0.0 2015.11.05 Initial python rewrite. tjm
#
# 2.1.0 2016.03.07 "Now with spinning rims"
# bug fixes, obfuscation features,
# additional tools and examples. tjm
#
# 2.1.1 2016.03.16 slack identifier customization,
# logic clarifications. tjm
#
# 2.1.2 2016.03.16 cleaned up argparse. tjm
#
# 2.1.3 2016.04.04 remove obsolete flag logic. tjm
#
# 2.1.4 2017.10.23 using rm -P for secure delete,
# added additional alerting, additional pylint cleanup. tjm
#
# 2.5.0 2017.11.xx removed flags, uses configuration file.
# added testing fuctionality. tjm
#
```

Version 2

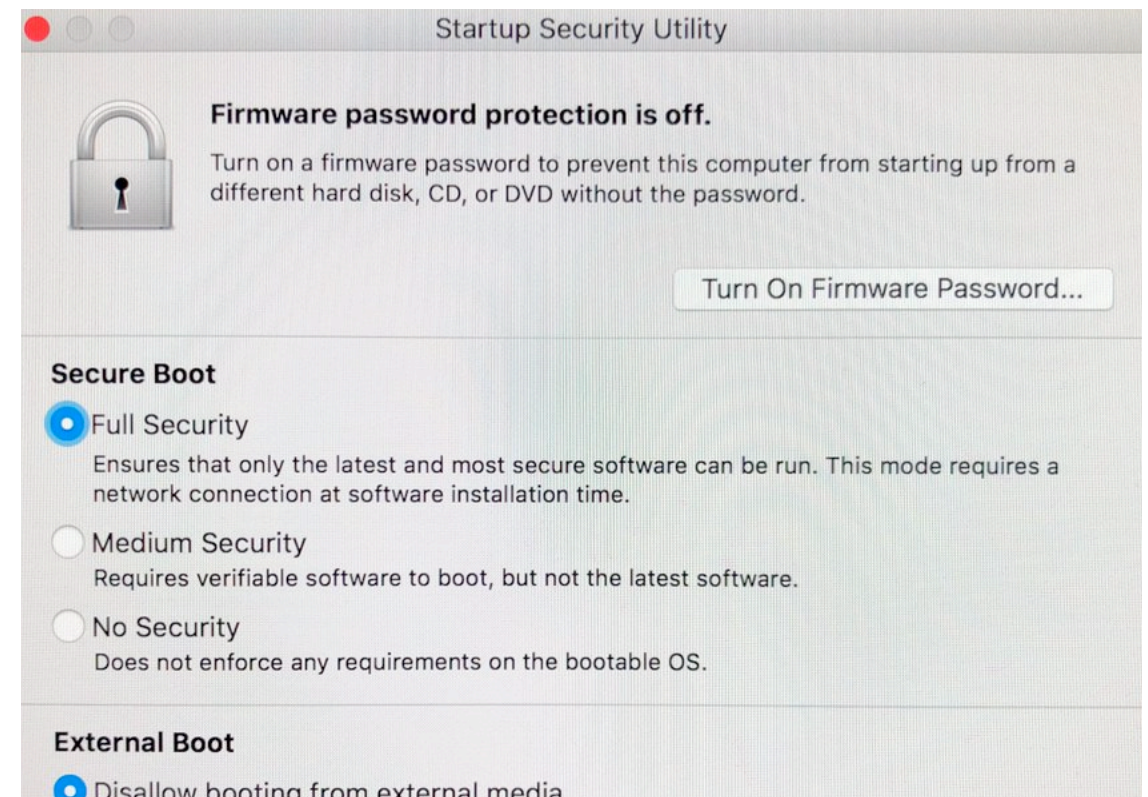


Thank you for your interest!

Why use firmware passwords?

- Prevents non-privileged users from booting from a foreign device
- Strong user passwords ***AND*** Full disk encryption
- Renders device effectively useless

Apple's solution
(Hasn't changed since
Mac OS 9, or earlier...)



Other Management Systems
are the same as Apple's
(JAMF, etc)



Why use FWPM?

- Management solution, not “set-it & forget-it”
- Staff turnover
- Fat fingering (prevention and recovery)
- Open source

Before you start

- Keyfile
- Plaintext or obfuscated
(doesn't look like anything to me)
- Hash
- nvram

How FWPM works

- Compare hashes
- Injest keyfile
- Find current password
- Change password
- Update hash
- Report

What's New?

FWPM 2.1.4 (late October)

- srm removed in 10.12
- rm -P (3x overwrite)

FWPM 2.5 “Classic”

- Ease of use
- Minimize flags
- Eliminate the need to edit the script
- Move options to configuration file
- Add ability to use setregproptool

Current interface

```
usage: firmware_password_manager.py [-h] [-r] -k KEYFILE [-t] [-v]
                                     [-m MANAGEMENT] [-#] [-n] [-s] [-o] [-b]
                                     [-i IDENTIFIER]
```

Manages the firmware password on Apple Computers.

optional arguments:

-h, --help	show this help message and exit
-r, --remove	Remove firmware password
-k KEYFILE, --keyfile KEYFILE	Set path to keyfile
-t, --testmode	Test mode. Verbose logging, will not delete keyfile.
-v, --version	show program's version number and exit
-m MANAGEMENT, --management MANAGEMENT	Set nvram management string
-#, --hash	Set nvram string to hash of keyfile
-n, --nostring	Do not set nvram management string
-s, --slack	Send important messages to Slack.
-o, --obfuscated	Accepts a plist containing the obfuscated keyfile.
-b, --reboot	Reboots the computer after the script completes successfully.
-i IDENTIFIER, --identifier IDENTIFIER	Set slack identifier. [IP, hostname, MAC, computername, serial]

New interface

usage: firmware_password_manager.py [-h] -c CONFIGFILE [-t]
[-v]

```
/_ _/ /_ _/ University of Utah  
 _/   _/   Marriott Library  
 _/   _/   Mac Group  
 _/   _/   https://apple.lib.utah.edu/  
_/_/      https://github.com/univ-of-utah-marriott-library-apple
```

Manages the firmware password on Apple Macintosh computers.

optional arguments:

-h, --help	show this help message and exit
-t, --testmode	Test mode. Verbose logging, will not delete keyfile.
-v, --version	show program's version number and exit

Required management settings:

Choosing one of these options is required to run FWPM. They tell FWPM how you want to manage the firmware password.

-c CONFIGFILE, --configfile CONFIGFILE
Read configuration file

configuration file

```
[flags]  
fwpm_action:  
management_string_type:  
custom_string:
```

```
[keyfile]  
path:  
use_obfuscated:  
remote_mode:  
server_path:  
username:  
password:
```

```
[logging]  
use_logging:  
log_path:
```

```
[slack]  
use_slack:  
slack_identifier:
```

```
slack_info_url:  
slack_info_channel:  
slack_info_bot_name:
```

```
...
```

FWPM for JAMF Pro

- Keyfile included in script
- JAMF script security

Firmware Password Manager JSS

General	Script	Options	Limitations
---------	--------	---------	-------------

Script Contents

```
156 request.get_method = lambda: 'PUT'
157 url = opener.open(request)
158
159
160 def main():
161     #
162     # Open log file
163     logger = loggers.file_logger(name='FWPW_Manager for JSS')
164     logger.info("Running Firmware Password Manager for JSS")
165
166
167     #
168     # These two variables replace the separate keyfile.
169     # current_keylist contains all of the previously used password(s)
170     # new_key contains the new pass
171     current_keylist = ["OldestPassw0rd", "currentPassw0rd",]
172     new_key = "newPassw0rd"
173
174
175     local_args = []
176     if len(sys.argv) >= 4:
177         local_args = sys.argv[4:]
178         local_args = [x for x in local_args if x]
179         logger.info("Local args: %r" % local_args)
180
```

Escrowing FWPM for Jamf

- Random firmware passwords for each machine.
- Sounded like such a cool idea. :)
- Easy to add to Skeleton Key

FWPM Installer Builder

- Make it as easy as possible to deploy FWPM



FWPM Tool Builder v.02



Firmware Password Manager **Installer Builder**



Build FWPM installer

Build keyfile installer

Current JSS Hash

Quit




Firmware Password Manager Installer Builder



Build custom FWPM Installer:

Use Slack: ☒ Yes ☐ No

Client Identifier: IP 

Info URL:

Info Channel name:


Info Bot Name:

Error URL:

Error Channel name:

Error Bot Name:

Packaging Options:

Signing Certificate: 

Optional: ☒ configuration file

☒ pexpect

☒ management_tools

Output:

Package Output Path: 

Build Installer

Return to home

Quit



FWPM Tool Builder v.02



Firmware Password Manager Installer Builder



Build custom FWPM Keyfile Installer:

Keyfile Source: ...

Firmware Password: ☐ Enable ☒ Disable

Packaging Options:

Signing Certificate: ▾

Include configuration: ☒ Enable ☐ Disable

Postinstall Actions:

Reboot after run: ☐ Enable ☒ Disable

Output:

Package Output Path: ...

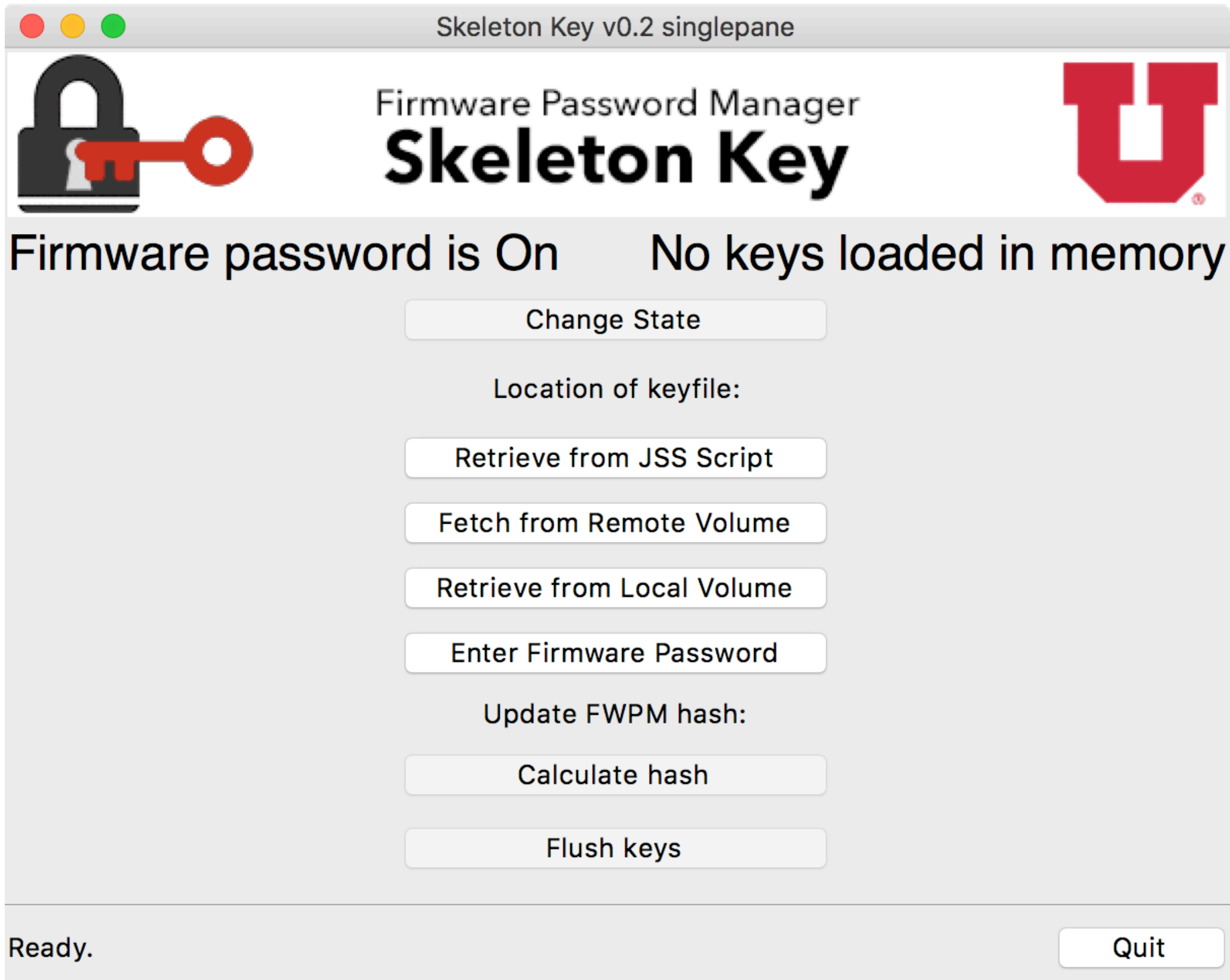
Build Keyfile

Return to home

Quit

Skeleton Key

- FWPM GUI Utility
- Help Desk, Technicians
- Combined with escrowed password, nifty solution



Skeleton Key v0.2 singlepane



Firmware Password Manager
Skeleton Key



Firmware password is On No keys loaded in memory

Change State

Location of keyfile:

Retrieve from JSS Script

Fetch from Remote Volume

Retrieve from Local Volume

Enter Firmware Password

Update FWPM hash:

Calculate hash

Flush keys

Ready.

Quit

When?

Future directions

- Continue making fwpm as secure as possible
- Build firmware password profiles?

Questions?

Code and Documentation:

https://github.com/univ-of-utah-marriott-library-apple/firmware_password_manager

Firmware Password Manager for OS X

<https://apple.lib.utah.edu/firmware-password-manager-for-os-x/>

MacAdmins Slack:

@todd.mcdaniel_uutah