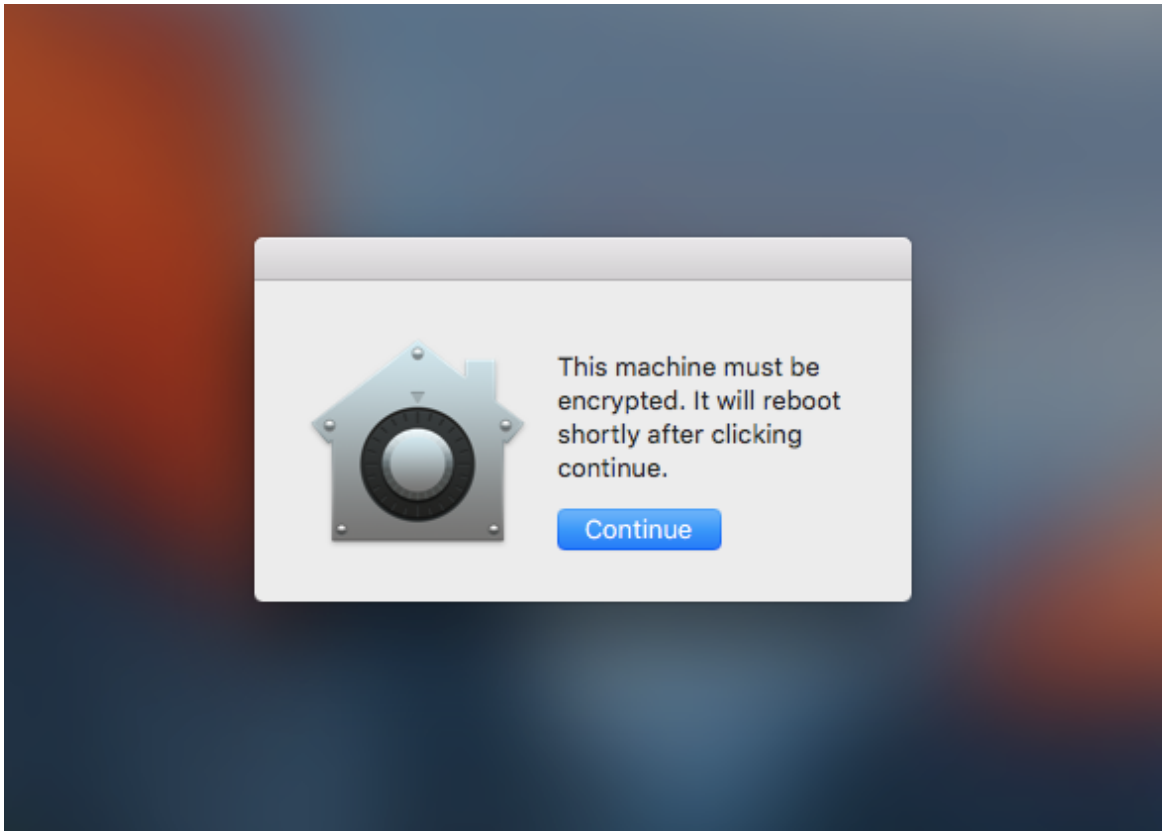
Crypt FileVault 2 Escrow Service for Macs, Whitepaper



Prepared by Allister Banks. Last Updated June, 2016

Background

This whitepaper describes how the Crypt server and client components combine to provide self-hosted FileVault 2 recovery key escrow for Macs. It was created by Graham Gilbert, and is in active use in many institutions worldwide. Crypt incorporates an enforcement GUI agent and a Django-based web console for escrowing the recovery key, among other secrets. The web component logs access and incorporates an approval workflow, so techs are accountable for retrieving the information, and enables the tracking of other secrets like firmware or local account passwords.

Executive Summary

When Apple re-built its FileVault concept in OS 10.7 to enable whole-volume encryption, it allowed the generated recovery key to be stored on their servers, based on the users Apple ID. As this did not sit well with large enterprises, Google released an open source project for triggering encryption and implementing escrow to a App Engine-hosted web application, with access control and notification functionality. Graham Gilbert similarly created Crypt-Server, so that a Django version of the web app could be self-hosted, and it was released along with a much more user-friendly GUI Crypt client application in 2012. That client can be branded, and its successor Crypt2 enables user-initiated encryption and escrow. It can block login until the machine is encrypted, and therefore keeps it in that state even if users running as admin turn FileVault off. The web app allows delegation and logged access control so managers can approve access by techs.

Features and Operation Overview

Client

Formerly, a login hook would check if you wanted to skip initiating encryption for a local user login, so that it's only the end user from a corporate directory service, or a user from a specific UID range that would be able to initiate encryption and unlock the disk. As of Crypt2 the client is handled in an authorization plugin, which removes multiple prompts for a users credentials before initiating the encryption process, and as before it removes the need for the logging in user to be an admin on the computer. Tom Burgin did a lot of the work to convert the client into Swift, making it even more Mac-native, and debug info and flexible preferences were added at the same time.

The client uses curl with Apple's Open Transport SSL validation functionality, which therefore will ensure it's talking to the proper host when escrowing the key. It outputs status to syslog to assist in troubleshooting and track actions that it needs to perform.

Server

Crypt-Server can be bound to LDAP (when using the provided Docker image) and logins can be restricted to an admin group. Within that group, a manager can be allowed to approve access requests, but otherwise techs would only have access to see what secrets are stored. Initiating the request for approval gets logged, and a timeout window can be applied so techs would not be able to access the information after that time.

This is provided in part by using the Django web framework's built-in admin functionality, and the management console also allows mass actions to be performed in a GUI (e.g. in the case of purging old data after a re-imaging project where that historically logged data is decided to no longer be of use).

Differentiation and Compliance

The project accepts code contributions from large organizations like Dropbox, the Montefiore Health System, and Airbnb to implement things like strict certificate checking on the server and client improvements. Many popular Mac management systems that offer escrow have no way of delegating or tracking access to the recovery key, nor do they accept arbitrary secrets or have a client-side mechanism to allow users to self-initiate encryption and escrow. They also require their admin user to be added to the list for even the most basic of functionality, which removes the appearance that the end user is the only one the computer is assigned to, which in turn diminishes their sense of responsibility. Even though the database for Crypt stays in an encrypted format on-disk, it allows bulk migration in or out, unlike other products, whose labor-intensive tools can result in vendor lock-in.

Some environments store other secrets, like firmware or local admin account passwords in Crypt Server. This allows the same access control workflow, and can be interfaced with the same curl-based, TLS-secured communication. Niceties include email notifications on access and settings to enable a mode that allows all techs with access to retrieve all keys be default, but not the ones for their own machines.

Compliance

PCI-DSS recommends many common-sense practices that the design and deployment of Crypt-Server supports, including SSL for both escrow and admin access(as specified in section 2.3[1]) shipping the key off the system once encryption is initiated(as per 3.2.3) and that access to the keys are logged (3.4.1).

In general, the use of Apple's FileVault is backed up by its FIPS 140-2 conformance[2]. Of the NIST guidelines, SC-12[3] in particular calls out the critical aspect of escrowing the cryptographic keys that protect full disk encryption, which of course includes FIPS 140-2 conformance.

As per HIPAA Administrative Simplification 164.312(a)(2)[4], in addition to the client enforcing the encrypted state, controlling access to the FileVault keys becomes vital to emergency procedures for obtaining protected health information.

Conclusion

By automating and providing a workflow around encryption, maintaining security becomes manageable at scale. Crypt helps avoid lockout during recovery of equipment from employees that are no longer with the company, & delivers a complete solution that ensures data safety is not compromised.

---

[1] https://www.pcisecuritystandards.org/document_library?category=pcidss&document=pci_dss

[2] http://training.apple.com/pdf/WP_FileVault2.pdf

[3] http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf

[4] http://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/administrative/securityrule/securityrulepdf.pdf