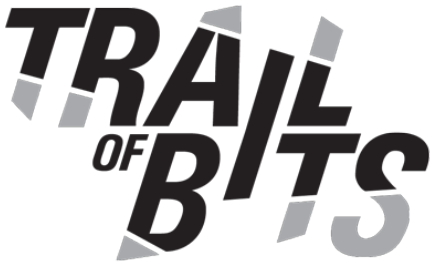

ArmorLock

Secure Enclave Vulnerability Report

January 25, 2021

Prepared For:
Brian Mastenbrook | *Western Digital*
brian.mastenbrook@wdc.com

Prepared By:
Artur Cygan | *Trail of Bits*
artur.cygan@trailofbits.com

Executive Summary

From January 5 to January 25, 2021, Trail of Bits assessed the security of Western Digital's ArmorLock product. Trail of Bits performed this assessment with three engineers, totaling six person-weeks. Western Digital provided us with firmware and client source code for ArmorLock version 1.4.0.

The assessment resulted in nineteen findings, ranging from high to informational severity. The only high-severity finding detailed a scenario in which ArmorLock did not use the Secure Enclave on Apple devices not supporting biometric authentication due to a faulty enclave detection mechanism. The keys used for setting up a secure channel to the drive, client identification, and drive unlocking were generated, stored, and used outside of the Secure Enclave, where they were subject to potential exposure.

As the issue weakens the security of ArmorLock users on the affected Apple devices, Western Digital immediately initiated remediation efforts for the underlying problem and began developing a migration strategy for the affected users. Western Digital also asked Trail of Bits to review the refactored key generation code and migration strategy. This analysis confirmed that the fixes were implemented correctly. This document describes the details of the issue and remediation for the finding.

The corresponding Western Digital tracking number for this vulnerability is WDC-21003.

Key exposure on some Apple devices with Secure Enclave

Severity: High
Type: Data Exposure
Target: `ArmorLock Apple clients`

Difficulty: Medium
Finding ID: TOB-WDCSF-011

Description
Secure enclaves are designed to protect the key material from leaking and are resistant to kernel and userspace exploits as the key material never leaves the enclave. ArmorLock uses the Apple Secure Enclave to generate and store multiple private keys used for setting up a secure channel to the drive, client identification, and drive unlocking. However, due to a faulty enclave detection mechanism, ArmorLock prior to version 1.4.1 does not use the Secure Enclave on all Apple devices having it.

The keychain API must be explicitly instructed by the calling code to use the Secure Enclave. As of January 2021, there is no API to detect if a Secure Enclave is present on an Apple device. The ArmorLock's code checks if a Secure Enclave is present on a macOS or iOS device based on biometric authentication feature availability.

However, this is not true for all devices. For example, the Mac Mini and iPad Air have a Secure Enclave but do not support biometric authentication. This causes the keys to be generated, stored, and used outside of the Secure Enclave. The affected keys are Transport Pairing Key (TPK), Default Unlock Key (DUK), and Unlock Pairing Key (UPK).

Exploit Scenario
An ArmorLock user configures a Mac Mini (2018 or later) as an authorized device. The affected keys are generated and stored in the software keychain instead of the Secure Enclave. An attacker having access to the user's keychain or the client's memory can steal the key material and unlock the user's drive.

Recommendation
Short term, modify the key generation code to assume that Secure Enclave is present. Then, on devices without a Secure Enclave, handle the eventual errors when adding keychain items.

Long term, add automated tests and expose an interface for quality assurance to check if the keys are stored inside the Secure Enclave to prevent this mistake from happening for all the current and future devices.

Remediation

Remediation of this finding required two different approaches. First, the refactored code provided by Western Digital corrected the weakness in the key generation when using the keychain API. Second, focused on protecting the users by securing the existing keys generated outside Secure Enclave on affected devices.

Western Digital adopted our recommendation and fixed the root cause of the issue. We were provided with a code diff and verified that the patch is implemented correctly. The keys are now always generated with the assumption that the Secure Enclave is available, and if not, the code falls back to using the software keychain. This guarantees that the keys are always generated and stored inside the Secure Enclave, given it is present on the device.

To secure the already generated key material, Western Digital prepared an automatic migration strategy involving key wrapping, as the Secure Enclave does not support importing any existing key material. This was also provided for analysis as a code diff. The TPK and DUK can't be automatically re-generated without breaking the application. The UPK will be eventually re-generated in the Secure Enclave; however, this requires plugging in the drive running firmware updated to a forthcoming version 1.5 or later. Until then, it will remain wrapped as the other two keys.

The keys subject to wrapping are detected by a test fetch of the key material, which would otherwise fail should the key reside in the Secure Enclave. The key material is then symmetrically encrypted and put into the relational database as a ciphertext. The symmetric key is stored in the Secure Enclave. This provides better security for the keys at rest; however, the key material will still reside in ArmorLock's memory during usage. We also verified that this mechanism is implemented properly during the code diff review.

A. Vulnerability Classifications

Vulnerability Classes	
Class	**Description**
Access Controls	Related to authorization of users and assessment of rights
Auditing and Logging	Related to auditing of actions or logging of problems
Authentication	Related to the identification of users
Configuration	Related to security configurations of servers, devices or software
Cryptography	Related to protecting the privacy or integrity of data
Data Exposure	Related to unintended exposure of sensitive information
Data Validation	Related to improper reliance on the structure or values of data
Denial of Service	Related to causing system failure
Error Reporting	Related to the reporting of error conditions in a secure fashion
Patching	Related to keeping software up to date
Session Management	Related to the identification of authenticated users
Timing	Related to race conditions, locking or order of operations
Undefined Behavior	Related to undefined behavior triggered by the program

Severity Categories	
Severity	**Description**
Informational	The issue does not pose an immediate risk, but is relevant to security best practices or Defense in Depth
Undetermined	The extent of the risk was not determined during this engagement
Low	The risk is relatively small or is not a risk the customer has indicated is important
Medium	Individual user's information is at risk, exploitation would be bad for client's reputation, moderate financial impact, possible legal

	implications for client
High	Large numbers of users, very bad for client's reputation, or serious legal or financial implications

Difficulty Levels	
Difficulty	**Description**
Undetermined	The difficulty of exploit was not determined during this engagement
Low	Commonly exploited, public tools exist or can be scripted that exploit this flaw
Medium	Attackers must write an exploit, or need an in-depth knowledge of a complex system
High	The attacker must have privileged insider access to the system, may need to know extremely complex technical details or must discover other weaknesses in order to exploit this issue