U2F設備安全測試

嘶吼RoarTalk
網絡安全觀察者

已關注

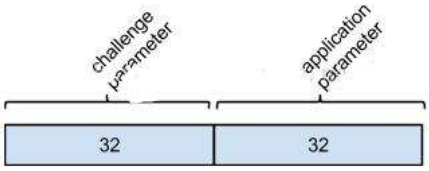
4 人贊同了該文章

在上一篇文章中，我對U2F設備的各種安全密鑰進行了一個很粗淺的安全介紹，這其中就包括了：基本功能，物理特徵等，如果你感興趣可以點此了解詳情。在今天的這篇文章中，我會從一個更低級別的特徵來測試U2F設備的安全性。

安全密鑰用的是FIDO U2F規範，U2F (Universal 2nd Factor) 是Yubico, Yahoo 和Google 聯合開發的基於物理設備的雙因素認證協議，目前已經完成標準化，從屬於FIDO (Fast Identity Online) 聯盟名下。另外，FIDO U2F規範借鑒了智能卡ISO7816-4規範中的許多標準。FIDO U2F盡可能對每種可能的傳輸（USB，NFC或藍牙）都進行規範，已確定它們在傳輸時是如何封裝U2F消息的，例如，USB接口會如何封裝U2F消息。目前FIDO正在對目前的標准進行修訂，以應對未來可能的情況，不過目前所有的安全密鑰標準都實現了基本的安全功能。簡而言之，U2F包含三大部分：註冊，驗證和檢查。

註冊時U2F設備會創建一個新的密鑰對，更準確地說，註冊時服務器會產生一個32字節的Challenge數據和一個32字節的應用id。

註冊請求消息有兩部分，如下圖所示：



challenge parameter[32 bytes]， challenge parameter是對客戶端數據進行SHA-256運算得到的哈希值，另外Challenge是一個隨機數。Client Data是一個由FIDO客戶端準備的JSON數據結構。

application parameter[32 bytes]，application parameter是請求註冊的應用id的SHA-256結果。而對於瀏覽器，應用id是登錄頁面的起始地址中的URL的哈希值。

U2F設備的密鑰對

U2F設備產生的密鑰對是服務器關聯的，一對密鑰對應一個服務器，而不是一個U2F設備對應一個服務器。在註冊的時候，服務器給U2F設備傳入服務器相關信息，U2F設備產生一對密鑰對，將此密鑰對和服務器相關信息相關聯，給此密鑰對分配一個key handle,將其和公鑰傳給服務器，服務器將註冊的賬戶信息，公鑰，key handle全部關聯在一起並保存。

鑰對，如果密鑰對存在，檢驗密鑰對應的服務器信息是否和傳入的服務器信息匹配，如果不匹配，說明服務器是偽造或者不正確的。如果正確，U2F設備等待用戶按鍵確認，用戶按鍵後，U2F設備對Challenge數據做簽名，應用id返回給服務器，服務器驗證應用id，如果簽名正確，說明此公鑰對應的唯一私鑰是正確的，表明用戶擁有合法的U2F設備，如果簽名不正確，說明此用戶正在偽造身份登錄。

U2F設備安全特性

然而一個比較有趣的地方是，U2F設備不帶無窮大的安全存儲空間。在實際研究過程中，我發現key handle實際上是一個加密的私鑰，即用令牌卸載存儲的私有密鑰。然而，從理論上講，key handle可能只是一個整數，用於對令牌內的存儲密鑰進行索引。所以為了保證安全性，私鑰的保護很重要。U2F協議允許一個廉價的設備，同時保證此設備不會洩露私鑰。Key handle可以不是U2F設備上一個私鑰的索引，相反，Key Handle可以用來存儲私鑰和服務器相關信息，這些信息可以被加密保存到一個Key Handle中（例如使用aes加密私鑰和服務器數據）。

鑑於此，在新修訂的FIDO U2F規範中，有許多關鍵的安全規範還是應該保留下來，比如：

1.Key Handle应加密，以防止在那里找到ECDSA私钥。
2.一个安全密钥的Key Handle不能与另一个安全密钥一起使用，即使是相同的类型。
3.如果一个安全密钥被要求生成数百个密钥对，那么它们都应该是不同的。
4.所有的签名都应该有唯一的随机数，否则就可能出现安全漏洞，让人提取你的私钥。
5.应该对应用id进行检查。

但除此之外，還有其他一些安全性需要進行測試：

1.Key Handle是否是U2F令牌产生的，如果U2F令牌发现key handle不是自己创建的，直接进行错误返回，
2.签名是否被正确编码了？U2F设备的签名采用的是ASN.1编码。
3.USB协议传输的是64字节数据包，最后的填充字节应全部为零，而不是随机的内存字节。

給了這麼多理論的安全建議，但最終還是要理論結合實踐，下面就讓我們看看目前市場上的U2F設備是如何進行密鑰管理的。

Yubico

Yubico的產品目前還處於比較領先的位置，產品還暫時沒有發現什麼漏洞，必定U2F的標準就是Yubico牽頭制定的。

Vasco公司的SecureClick

Vasco公司的SecureClick也是一款很不錯的產品，具體介紹點此github.com/hillbrad/U2F ...。這是一個藍牙低功耗（BLE）令牌，這意味著它可以與Android和iOS設備配合使用。對於非移動設備，它包括一個USB-A BLE加密狗。SecureClick使用Chrome擴展來配置和配對跨平台的加密狗。加密狗看起來像一個普通的USB設備，但區別就是它能從令牌中檢測到到一個BLE信號，此時它會“斷開”，並進行重新連接，以執行U2F操作。一旦需要用戶註冊或認證的操作完成，則令牌就會自動斷電，並且加密狗再次斷開並重新恢復到原始USB設備的狀態。

如果你正在使用Linux，那配置的udev（設備管理器）就可以正常顯示令牌供應商的ID和產品ID的訪問權限，不過由於供應商ID和產品ID是不同的，所以令牌無法正常工作， Chrome擴展程序也很混亂。還有一個問題就是BLE令牌設備的電池經常會因電量耗盡而停止工作。

飛天誠信的ePass

ePass系列是由飛天誠信公司推出USB Key，主要是用作基於公鑰體系的數字證書和私鑰的安全載體，大小如同家門鑰匙，形狀和表面上的U盤相像，可以穿在鑰匙環上隨身攜帶。目前飛天誠信已

不過ePass並不具備這樣的安全密鑰，因為從9個前導零位開始的數字在開始時都有一個無效的零字節。可以推測，從17個零位開始的數字在開始時都有兩個無效的零字節，但是我沒有足夠的時間來尋找具體的案例。不過我可以推測這種安全密鑰生成的256個簽名中，只有一個屬於無效編碼。

此外，Key Handle的最後八個字節似乎是多餘的，也就是說，你可以將它們更改為你喜歡的任何值，至於是否能起到安全密鑰就不一定了。其實這倒不是什麼問題，真正的問題是，它們是否會被使用？另外，USB數據包中的填充數據不為零，不過這只是顯然了傳輸緩衝區的以前內容，並不會洩露什麼敏感數據。

Thetis

不知是什麼原因，我無法在此設備上測試一些安全特性，比如Key Handle的可變性，應用id是否經過了檢查等。對Key Handle的可變性檢查的響應是無效的，根據FIDO U2F規範，檢查響應的返回狀態為0x9000（“NO_ERROR”），但其實返回狀態應該為0x6985或0x6a80。之後，它開始拒絕所有的含有0x6a80的Key Handle（包括有效的）的響應。

該設備具有與飛天誠信的ePass相同的非最小簽名編碼問題，此外，如果你按鍵的速度太快，這個安全密鑰就會陷入混亂並拒絕一些status 0x6ffe請求。另外，USB填充字節也不為零。

U2F Zero

1KiB ping消息崩潰此設備（即它停止響應USB消息，需要拔下並重新插入）。測試一個損壞的鑰匙柄也會崩潰，因此我無法運行許多測試。

一個1KiB的ping消息即會讓U2F Zero崩潰，並停止響應任何USB消息，如果要重新運行就要拔下來重新插入。另外測試一個崩潰的Key Handle也會崩潰，所以最後我只能放棄對其的安全性測試。

KEY-ID/HyperFIDO

Key-ID和HyperFIDO設備，具有相同的固件，它們具有與ePass相同的非最小編碼問題，除此之外，還存在ASN.1缺陷。在ASN.1 DER中，如果一個數字的最高有效位被設置，那該數字就是負數。如果不是負數，則需要一個零填充字節。當測試最高有效位時安全密鑰會檢查第一個字節是否是> 0x80，但按照規範應該檢查第一個字節是否是>= 0x80。這樣檢查的結果就是有時會產生含有負數的簽名，使得檢查無效。

另外，USB填充字節不是零，而且還包括了不屬於請求或響應的一部分數據。雖然這些數據不太可能有什麼實質的影響，但它們是從哪裡來的呢？這個確實是個有待研究的問題。

還有就是封裝的密鑰也有一些問題。首先，字節16到31是設備和應用id的函數，因此，在不同的帳戶使用時，給定站點可以被動地識別到相同的標記。字節48至79未經身份驗證，並且在更改時，除了簽名錯誤之外，所有內容仍然有效，這表明這些字節就是加密的私鑰或生成它的加密種子。最後，雖然字節32到47不能被任意操作，但是可以用來自不同的key handle的替換相同的字節，這就會導致簽名錯誤。

> 本文翻譯自：imperialviolet.org/2017 ...，如若轉載，請註明原文地址：4hou.com/tools/7935.htm ... 更多內容請關注“嘶吼專業版”——Pro4hou

發佈於2017-10-16 10:17

信息安全