FDO Ownership Voucher Double Extension Vulnerability Guidance

Working Draft, March 06, 2023

This version:
https://fidoalliance.org/specs/fidoiot/appnote1-ov-guidance-v1.0-fd-20230306.html

Issue Tracking:
GitHub

Editor:
Geoffrey Cooper (Intel)

Version:
1.0

Abstract

Implementation guidance to protect against rogue supply chain attacks against Ownership Vouchers.

Table of Contents

1. Introduction§

The situation in this issue is that an ownership voucher is extended to two prospective owners, A & B. A is the legitimate target, and B is a rogue owner. This extension is illegal according to the FDO protocol. However, it might still happen if a supply chain entity is careless, compromised, or malicious. In this case, both A and B have a valid ownership voucher, and either can onboard the device. Figure 1 outlines this scenario.

But which of A or B actually onboards the device? The result depends on how A and B use the TO0 protocol. A and B compete for Rendezvous server entries as follows. A uses the TO0 protocol to send its ownership voucher to the RV server and negotiates a timeout, say 10 hours. An hour later, B uses the TO0 protocol to send its ownership voucher to the same RV server. Since the GUID is the same, the RV server pairs B's request against A's entry and renegotiates the timeout, perhaps for another 10 hours. Since the RV server indexes its entries by GUID, during the first hour, A's rendezvous information is returned. If the device performs the TO1 protocol during this time, onboarding will proceed to the TO2 protocol with A's configured address (RVTO2Addr). However, after the first hour, and during the subsequent 9 hours, B's information is returned. A device arriving during this interval will apply B's Rendezvous address, onboarding to the server of B's choice. If we think of the device as a random arrival, it is 9 times more likely to choose B over A.

If A and B choose different RV servers, the device could end up at either A or B, depending on which server the device tries first.

If we assume a malicious attack, where B knows of A, B can mount a denial of service (DoS) attack against A to prevent it from renewing the RV server entry, allowing it a freer hand.

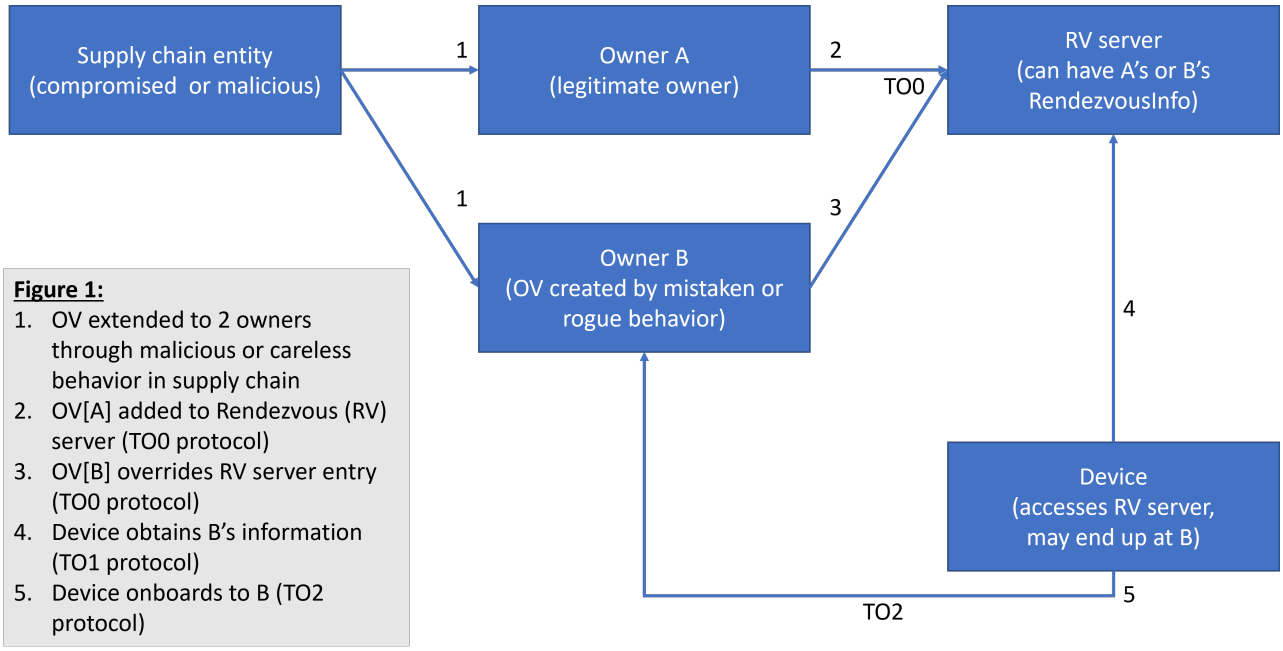

Figure 1 *Ownership Voucher Double Extension Vulnerability*

2. Classic Countermeasures§

The most obvious countermeasure is for A to verify that the device was actually onboarded as planned. For this to happen, A must know whether the device has been delivered and installed at the target site. Then, if B actually onboards the device, A can notice that its owner never received a TO2 call from the device and take steps to remove the physical device and try again.

This means that B might have access to a device within A for a short time. More serious is that this presumes A has current information that the device has been delivered and onboarded. If A controls both the server and the premises of the device, this can be true. For example, if A is managing devices in a company building, and 10 devices are scheduled to be onboarded on a given day, then A can check at day's end whether only 9 devices were onboarded. From the shipping manifests, it can then determine which one failed to onboard.

If the device is shipped to the target location by a 3rd party, not under A's control, this countermeasure becomes problematic. Since A has no way to know when the device will be delivered, it cannot determine if it has failed to onboard or is simply still in transit. Even if the transit delay is known in general, a rogue device might still be installed in the target environment for a significant period of time.

Another countermeasure is for A to use network security to ensure that devices on its premises can only contact its own trusted servers (which must include the RV server). This cannot prevent B from polluting the Rendezvous server, but can prevent A from using the TO2 protocol to access B. Again, this is only possible if A controls both the device install location and the premises where the device is installed.

Both these countermeasures fail in the case where the premises are owned independently of the server, and there is no way for the owner to control the network security of the premises.

3. Modification of Rendezvous Server Behavior§

To ameliorate this situation, we can add verification to the Rendezvous Server to make it harder for B to compromise A's position. We attempt to add behavior to the RV server that is:

- Within the framework of the FDO specification (i.e., in accord with the TO0 and TO1 protocol specifications)
- Uses existing information available to the Rendezvous server

The pertinent discussion appears in the FDO 1.1 protocol specification, section 5.3.3. The RV server is given the ability to verify to the trust of an ownership voucher (Note: this section has several typographical errors where ownership voucher is called 'ownership proxy'). The section gives several mechanisms by which a RV server may measure the trust of an ownership voucher, and includes the statement:

"A given Rendezvous Server MAY choose to reject Ownership Vouchers that are not trusted."

We take this to mean that additional trust mechanisms may also apply for given RV servers.

For this situation, we mandate that the RV server take the following additional steps:

1. The RV server shall cache the owner public key from the ownership voucher for each stored Rendezvous entry. This public key is OVEntryPayload.OVEPubKey in the last entry of OVEntries.
2. The RV server shall reject TO0 requests where the owner public key does not match the key of the RV server's existing entry.
3. The RV server shall maintain memory of the public key, or its hash, used for each GUID after the RV entry is timed out for purposes of TO1, and reject other owner public keys as in [2].
    1. This public key entry shall be persisted for a timeout sufficient to make a DoS attack difficult to sustain.
    2. We propose a timeout of one week. The timeout shall be extended whenever the TO0 protocol completes successfully
    3. The RV server may persist the complete entry from the last successful TO0 or may maintain only a smaller information consisting of \
    4. An attempt to run the TO0 protocol with a different public key than persisted shall return in error from the TO0.OwnerSign message. The recommended error code is INVALID_OWNERSHIP_VOUCHER.

A prospective owner A which has no control of the device' target premises, may ensure that all potential RV servers are successfully registered with its ownership voucher *before* shipping the device to the onboarding premises. Then owner A can maintain the RV server entry/entries until the device onboards. Now a malicious owner B may be detected as follows:

- If owner B has a valid ownership voucher with the same GUID but a different owner key, the RV server reject the request because its owner key does not match the saved entry.
- If owner B attempts to use A's public key, it will fail to verify the ownership voucher in the TO0.OwnerSign message, because it does not have A's private key. The Rendezvous Server MUST verify the signature on TO0.to1d, as per protocol specification, section 5.3.3.
- If owner B has owner A's public and private keys, it has compromised A beyond our ability to detect. However, with even modest protection of A's private key, this is only possible if B has violated A's premises.

4. Application§

This modification of behavior is recommended for all Rendezvous servers where the ownership of the onboarding premises does not match the ownership of the onboarding device. It is safe to be applied in other situations, can constitute default behavior for a Rendezvous Server.

It is likely that a rendezvous server in a closed network benefits little from this approach.