Capabilities and Expectations Working Group

NTIA Multistakeholder Process

on IoT Upgradability and Patching

July 18, 2017 Meeting

Capabilities WG Overview

- Desired Outcomes:
  - A **shared understanding of the component steps in an update, including a baseline** for security purposes
- Draft Document Status
  - Basic Steps in an Illustrative Over-the-Air Update Process
  - Security features for each step, including basic steps and layers of enhancement
- Goals and audience
  - Voluntary, nonregulatory guidance
  - Update mechanisms should not introduce new security risks
  - Aimed at IoT manufacturers, solution implementers, system integrators, and those who deploy and maintain systems

WG2 Summary of Activity

- Initially pursued two tracks: device categories and the steps of an update
  - Goal: mapping between **update steps** and **necessary tech/capabilities** of that device and its supporting systems
  - Reviewed a wide range of device **categories, capabilities, and use cases**
  - Considered a **metric** for update security
  - *Decided for the moment that a better operative goal was to understand the **nature of an update** (software or firmware), and the **security features** of those updates.*
- Steps in an update that might apply to a very wide range of devices
- Security features to secure this update process
- Scoping:
  - **Connected, remotely addressable devices** (as opposed to non-connected devices)
  - The development of the update by manufacturer is out of scope.

Basic Steps

An Illustrative OTA Update Process

0. *Create – Update image created.*
   Important, but out of scope for guidelines.

1. **Sign – Ensure update integrity.**
   Update is signed.
2. **Protect – Prevent update exposure.**
   Update is encrypted/obfuscated.
3. **Send – Data is in motion.**
   Update is communicated to target.
4. **Receive – Update is received.**
   Target receives update deliverable.
5. **Check – Update is processed.**
   Target validates, decrypts, and processes update deliverable as needed.
6. **Announce – User made aware.**
   End user notified about/approves of update installation.
7. **Distribute – Image distributed.**
   Update parsed/distributed to hardware targets (e.g. CPU, FPGA).
8. **Process – Image is processed.**
   Target hardware receives, validates, and decrypts update.
9. **Stage – System in pre-update state.**
   System-specific pre-update activities.
10. **Apply – Update process triggered.**
    Actual image install process is run.
11. **Re-verify – Post-update verification.**
    Target validates integrity of install and communicates results (if needed).
12. **Activate – Updated code enabled.**
    New code begins execution if verified.
13. **Clean-up – Post-update activities.**
    System –specific verification, messaging, and clean-up. (Could be negative.)

Security Features Mapping

Basic

1. **Sign – Ensure update integrity.**
   128-bit hash.
2. **Protect – Prevent update exposure.**
   Ephemeral, unique AES-128 keys in device.
3. *Send – Data is in motion.*
   No special assumptions.
4. *Receive – Update is received.*
   No special assumptions.
5. **Check – Update is processed.**
   Hash validation, decryption w/ per-device keys.
6. **Announce – User made aware.**
   Optional end-user approval.
7. *Distribute – Image distributed.*
   No special assumptions.
8. **Process – Image is processed.**
   Target hardware validates, decrypts image.
9. **Stage – System in pre-update state.**
   Manufacturer defined.
10. **Apply – Update process triggered.**
    Actual image install process is run.
11. **Re-verify – Post-update verification.**
    Target hardware validates installed material.
12. *Activate – Updated code enabled.*
    No special assumptions.
13. *Clean-up – Post-update activities.*
    No special assumptions.

Security Features Mapping

Upgraded (+1)

1. **Sign – Ensure update integrity.**
   NIST key management for hashing.
2. **Protect – Prevent update exposure.**
   NIST key management for decryption.
3. **Send – Data is in motion.**
   Cryptographic endpoint verification (e.g. challenge/response) before update.
4. **Receive – Update is received.**
   Best practices (e.g. TLSv1.2, certificate pinning)
5. *Check – Update is processed.*
   See Basic.
6. *Announce – User made aware.*
   See Basic.
7. **Distribute – Image distributed.**
   Encrypted in motion; can target multiple layers.
8. **Process – Image is processed.**
   Target validation, decryption with AES-256.
9. *Stage – System in pre-update state.*
   See Basic.
10. **Apply – Update process triggered.**
    Optional synchronous updating, end-user coordination, and data persistence.
11. **Re-verify – Post-update verification.**
    Install validated w/hash (AES-256), checksum (CRC-16).
12. *Activate – Updated code enabled.*
    See Basic.
13. **Clean-up – Post-update activities.**
    Local success notification; external logging of successful updates (including ID, versioning).

Security Features Mapping
Enhanced (+2)

1. **Sign – Ensure update integrity.**
   Upgraded (+1), plus secure memory and PKI.
2. **Protect – Prevent update exposure.**
   Upgraded (+1), plus secure memory and PKI.
3. **Send – Data is in motion.**
   Upgraded (+1), plus PKI.
4. **Receive – Update is received.**
   Best practices (e.g. TLSv1.3, certificate pinning)
5. *Check – Update is processed.*
   See Basic.
6. *Announce – User made aware.*
   See Basic.
7. *Distribute – Image distributed.*
   See Upgraded (+1).
8. *Process – Image is processed.*
   See Upgraded (+1).
9. *Stage – System in pre-update state.*
   See Basic.
10. *Apply – Update process triggered.*
    See Upgraded (+1).
11. *Re-verify – Post-update verification.*
    See Upgraded (+1).
12. *Activate – Updated code enabled.*
    See Basic.
13. *Clean-up – Post-update activities.*
    See Upgraded (+1).

Step		Description	Basic	+1 ("Upgraded")	+2 ("Enhanced")	+3 ("Quantum")
0	*Create*	*Update creation is important, but not in scope for this guideline. There are still security considerations inherent in this step.*				
1	**Sign**	Update signed.	128-bit hash.	NIST key management for hashing.	Secure memory, PKI.	SHA3-256 or Lamport.
2	**Protect**	Encryption and/or obfuscation	Ephemeral, unique AES-128 keys in device.	NIST key management for decryption.	Secure memory, PKI.	LWE or RLWE key exchange
3	**Send**	Communicated to target device.	*No special assumption.*	Endpoint verification	PKI.	← *See Enhanced.*
4	**Receive**	Target device receives update.	*No special assumption.*	Best practices (e.g. TLSv1.2, cert. pinning)	TLSv1.3.	
5	**Check**	Target validates, decrypts, and processes as needed.	Validation, decryption w/per-device keys.	← *See Basic.*		
6	**Announce**	End-user notified about / approves update install.	Optional end-user approval			
7	**Distribute**	Image parsed, distributed to HW targets (e.g. CPU, FPGA).	*No special assumption*.	Encrypted in motion; can target multiple layers.	← *See Upgraded.*	
8	**Process**	Hardware target receives, validates, and decrypts image.	Target hardware validates, decrypts.	Target validation, decryption w/ AES-256.	Target and image validation w/ AES-256.	SHA3-256 or Lamport.
9	**Stage**	System-specific pre update tasks.	Manufacturer defined.	← *See Basic.*		
10	**Apply**	Image install process runs.	*No special assumption.*	Opt. update and end-user coordination, data persistence.	← *See Upgraded.*	
11	**Re-verify**	Install integrity check; optional communication of results.	Install results validated.	Validated with hash, checksum.	Minimum CRC-16 checksum and AES-256 hash.	SHA3-256 or Lamport hash and checksum.
12	**Activate**	New code executes if verified.	*No special assumption.*	← *See Basic.*		
13	**Clean-up**	System-specific: verification, messaging, and clean-up	*No special assumption.*	Local notification and external logging of update.	← *See Upgraded.*	