



UL MCV 1376

Methodology for Marketing Claim Verification:
Security Capabilities Verified to level Bronze/
Silver/Gold/Platinum/Diamond

JUNE 12, 2019

1

UL MCV 1376

**Methodology for Marketing Claim Verification: Security Capabilities
Verified to level Bronze/Silver/Gold/Platinum/Diamond**

First Edition

June 12, 2019

COPYRIGHT © 2019 UNDERWRITERS LABORATORIES INC.

Table of Contents

REQUIREMENTS COVERAGE MATRIX.....	4
1 SOFTWARE UPDATES	5
1.1Software updates must be supported, using network or wireless interfaces where available	5
1.2Automatic software updates must be enabled by default.....	5
1.3Software updates must be cryptographically authenticated, and provide anti-roll back features	5
1.4Memory/storage locations used to store software updates after authentication must not be user accessible or externally writable	6
1.5Device implements a hardware based root of trust for updates and boot authentication	6
2 DATA & CRYPTOGRAPHY	7
2.1System defaults such as password, certificates, and/or cryptographic keys must be changed on initial setup	7
2.2A minimum password strength policy must be enforced for passwords that are used for access to sensitive services	7
2.3Sensitive data must be protected against exposure and unauthenticated modification	8
2.4Industry standard cryptographic algorithms must be used for security services	8
2.5Random number generation must ensure sufficient entropy.....	8
2.6Cryptographic keys must be managed to industry best practice	9
2.7Provide recovery mechanisms for credentials (where used).....	9
2.8Implement protection against brute force attacks	9
3 LOGICAL SECURITY.....	10
3.1Communication and debug ports must be protected against misuse.....	10
3.2Systems must be configured to secure defaults	10
3.3Customer access to disable unwanted features	10
3.4Memory and compiler protections must be implemented.....	11
3.5System software should be free of publicly disclosed vulnerabilities	11
3.6Unwanted / unnecessary features are removed	11
3.7System software must be tested to check for undisclosed vulnerabilities	11
3.8Logging and error messages must not expose sensitive data without authentication.....	12
3.9Systems must implement 'least privilege', or utilize hardware based features to protect sensitive code and data	12
4 SYSTEM MANAGEMENT	13
4.1Sensitive services must require authentication and ensure the confidentiality and integrity of data	13
4.2Permanent erasure of sensitive data must be supported.....	13
4.3Manual backup/override must be provided for safety related services.....	13
4.4No direct execution of scripts / commands	13
4.5System management services accessible over wireless and IP interfaces must implement session management to limit multiple sessions, and ensure on-going authentication.....	14
5 CUSTOMER IDENTIFIABLE DATA.....	15
5.1All customer identifiable data processed/collected/stored by the system must be documented	15

5.2	Collected/stored customer identifiable data can be erased on request.....	15
5.3	Customer identifiable data should be available for review and opt-in / opt-out options should be available	15
5.4	Notifications for privacy policy changes	15
5.5	Customer identifiable data must be anonymized	15
6	PROTOCOL SECURITY	16
6.1	Security sensitive communications must be robust against replay and MITM attacks	16
6.2	Device must support industry accepted wireless security defaults for any Wi-Fi connections.....	16
6.3	Connections to remote services must implement cryptographic authentication	16
6.4	Security protocols must implement secure defaults, and prevent downgrade attacks	16
6.5	Switched or wireless connections must allow for the use of an industry standard security protocol (such as TLS).....	17
7	PROCESS & DOCUMENTATION REQUIREMENTS	18
7.1	A documented process for the distribution of patches/updates must be maintained.....	18
7.2	A vulnerability management and disclosure program must be maintained	18
7.3	Cloud and App environments are regularly monitored for vulnerabilities	18
7.4	A 'Software Composition List' must be maintained	18
7.5	All protocols present in the device must be documented and justified	19
7.6	All services present in the device must be documented and justified	19
A.1	LOGICAL SECURITY POSTURE COSTING METHOD	21
A.2	ACCEPTABLE CIPHERSUITES	22
A.3	ACCEPTABLE CRYPTOGRAPHY	23
A.4	ACCEPTABLE PLAINTEXT PROTOCOLS	24

Requirements Coverage Matrix

Req. Category	Req. #	Req. Name	L1	L2	L3	L4	L5
1. Software Updates	1.1	Remote software updates supported	x	x	x	x	x
	1.2	Automatic software updates enabled	x	x	x	x	x
	1.3	Software update authentication	x	x	x	x	x
	1.4	Software update files inaccessible				x	x
	1.5	Hardware root of trust					x
2. Data & Cryptography	2.1	No default parameters	x	x	x	x	x
	2.2	Password policy for sensitive services		x	x	x	x
	2.3	Protect sensitive data		x	x	x	x
	2.4	Industry-standard cryptography			x	x	x
	2.5	RNG with sufficient entropy			x	x	x
	2.6	Industry best practice key management			x	x	x
	2.7	Credential recovery				x	x
	2.8	Brute force protection					x
3. Logical Security	3.1	Protect communication and debug ports			x	x	x
	3.2	Systems configured to secure defaults			x	x	x
	3.3	Unwanted functionality can be disabled			x	x	x
	3.4	Memory and compiler protection				x	x
	3.5	Software free from known vulnerabilities				x	x
	3.6	Unwanted / Unnecessary features removed				x	x
	3.7	Software tested for unknown vulnerabilities				x	x
	3.8	Logs or errors do not expose sensitive data				x	x
	3.9	Least privilege					x
4. System Management	4.1	Sensitive services require authentication	x	x	x	x	x
	4.2	Permanent erasure of sensitive data	x	x	x	x	x
	4.3	Manual back-up / override for safety critical operations		x	x	x	x
	4.4	No direct execution of commands / scripts				x	x
	4.5	Sensitive services implement session management				x	x
5. Customer Identifiable Data	5.1	Customer data collection is documented	x	x	x	x	x
	5.2	Customer data erased on request		x	x	x	x
	5.3	Opt in / out for data collection		x	x	x	x
	5.4	Notifications for privacy policy changes				x	x
	5.5	Customer data is anonymized					x
6. Protocol Security	6.1	Communications robust against replay and MITM attacks	x	x	x	x	x
	6.2	Industry standard Wi-Fi security	x	x	x	x	x
	6.3	Authentication for remote communications	x	x	x	x	x
	6.4	Secure defaults and downgrade prevention		x	x	x	x
	6.5	Secure switched / wireless connections		x	x	x	x
7. Process & Documentation	7.1	Documented patch / update process	x	x	x	x	x
	7.2	Vulnerability management program		x	x	x	x
	7.3a	No cloud / app			x	x	x
	7.3b	Regularly monitored cloud / app environment(s)			x	x	x
	7.4	Software Bill-of-Materials				x	x
	7.5	All protocols documented					x
	7.6	All services documented					x



Documents

Document Sharing Hub

1 Software Updates

1.1 Software updates must be supported, using network or wireless interfaces where available

Description / guidance:

No matter how well software is designed or tested, there will always be bugs and vulnerabilities that are missed. This is just a fact of software development and the sheer complexity of any body of code. So, the update of the software must be allowed in any device to ensure that it can be patched when any such bugs are found. It is an additional requirement that the software update must be able to be performed across a wireless or network interface, should the device provide such an interface. This increases the ease of use for the customer, removing disincentives to install updates. Security of the wireless and/or network interfaces is of course also important, and this is covered in later requirements.

This requirement ensures that such updates are possible, minimizing the risk that devices become permanently vulnerable through new attack methods that are discovered after the initial evaluation / shipping of the device. Checking for malicious changes to the software update is not covered under this requirement, and is instead addressed in a later requirement.

1.2 Automatic software updates must be enabled by default

Description / guidance:

Although software must be maintained to ensure on-going patching of security vulnerabilities, it is reasonable to expect that customers may not always know about the latest vulnerability in a device. Therefore it is important that automatic updates are implemented to update the software in a device, so that customers do not become a blocking factor in patching a high risk vulnerability. It is expected that device vendors will need to consider business / operational logic of the device when implementing such updates, so that a system does not reset during operation, but such business logic is considered outside of the scope of this assessment. It is important to note that although this requirement outlines the need for automatic updates to be supported, it does not mandate when and how such updates are provided, and device vendors may choose to deploy only high risk patches through such mechanisms. In all cases, vulnerabilities rated to CVSS 7 or higher must be patched within 1 month, those rated between CVSS 4 to 7 must be patched within 3 months, and vulnerabilities rated to less than 4 may be left unpatched.

1.3 Software updates must be cryptographically authenticated, and provide anti-roll back features

Description / guidance:

Although it is important to support software updates to ensure that devices can be patched and maintained in the field, such features can lead to additional vulnerabilities – where a 'bad actor' can install their own software into the device to prevent its normal operation.

To prevent this, it's important that any software update is cryptographically authenticated. Often this will be implemented by using a digital signature across the firmware image, which can be checked by the original firmware (or bootloader of the device) prior to installation. Using a digital signature based on a public key algorithm (such as RSA, or DSA) ensures that the devices themselves don't need the part of the key (the private or secret key) that is used to generate the authentication data.

Where a symmetric key system – such as a (H)MAC – is used, the secret key in each device must be unique per device. Otherwise once the firmware of one device is exposed (eg through a physical attack on that one device), a valid firmware signature for all other devices of this type can be created.

Therefore, public key cryptography is recommended to avoid the complexities of managing unique symmetric keys across device portfolios.

It is additionally required that the update implements 'anti-rollback' features – such as a 'monotonic' version number which is included in each release (that is a version number that only increases with each version), which is also checked to ensure that any bad actor can't just install a previous version of firmware; to 'reinstate' any otherwise patched vulnerabilities.

1.4 Memory/storage locations used to store software updates after authentication must not be user accessible or externally writable

Description / guidance:

Although software updates must be installed into the memory of the device so that they can be executed, having this code accessible after the authentication process has been performed allows for attacks where valid/unmodified code is maliciously altered after the authentication process, thereby invalidating the controls that authentication process is attempting to put in place to prevent the execution of malware. This requirement is simply to ensure that the software authentication and installation process is atomic – that it does not have discreet steps that can be separated and exploited to allow for invalid code to be installed.

1.5 Device implements a hardware based root of trust for updates and boot authentication

Description / guidance:

Although authenticating software updates is one important aspect of security, ensuring that any code is authenticated upon each boot of the device is also important. This ensures that even if changes are made to the executing code through some exploit, the changes cannot be made permanent and a reboot can be ensured to remove the malicious code.

2 Data & Cryptography

2.1 System defaults such as password, certificates, and/or cryptographic keys must be changed on initial setup

Description / guidance:

Ideally, system defaults should be avoided – but realistically that's not always possible. It may be necessary for something to be set to a default value to allow for the 'boot-strapping' of the system for the first time. However, the risk of using the default should be clearly outlined to the people operating that system for that first time, and this requirement outlines the need to force them to make a change from this default as part of the overall setup.

It is accepted that this can cause problems – forcing people to think up a new password during install, for example, can lead to a less than secure value being set (or a value that the user will forget the second they walk away). Where system defaults are automatically changed as part of the personalization / manufacturing process, these values must be set such that they are unique per device and statistically uncorrelated between devices (i.e. assigned randomly). Any such 'pre-changed' values must also be set in compliance with the password and cryptographic policies of the vendor.

Ultimately, defaults for passwords and other such items should only be implemented for values that are absolutely required to be present for normal operation, but which must be changed by the user before installation and operation. This also covers any 'test' values that may be in the firmware – such values must never be left in a production system.

Default and common values for certificate and public keys may be implemented to ease the remote management of systems. For example, it is common to have a globally fixed public key value across devices to authenticate software updates. However, such defaults must be clearly justified as to why they cannot be unique assigned per device. It is also required that such common public keys are kept unique to a specific type of device, so that any compromise of the root key does not render insecure the entire suite of devices from one manufacturer. This also ensures that code intended for one device type cannot be installed on another device type from the same manufacturer –as this can be used as a method for some attacks, or violation of business logic.

2.2 A minimum password strength policy must be enforced for passwords that are used for access to sensitive services

Description / guidance:

Passwords are often required and implemented to provide authentication of users. If not set to a value that is sufficiently secure, they can be easily guessed or brute-forced to bypass this authentication, allowing a bad-actor to gain access to the services the passwords are supposed to protect. Many attacks on devices are based on exploiting insecure, or default, password values.

Minimum levels for password security to sensitive services must be enforced, such that there is less than a 1 in 100,000 chance that any guess will be correct and that attempts to brute force the password domain in the device cannot be performed in less than 24 hours. These protections may include combinations of password strength and 'back off' timers on any password entry mechanisms to slow entry during high volume password entry attempts. System designers should consider the needs of customers to re-enter incorrect passwords cause through typographic errors, along with the need (or lack thereof) to support many hundreds or thousands of password entry attempts within a relatively short (e.g., 24 hour) period.