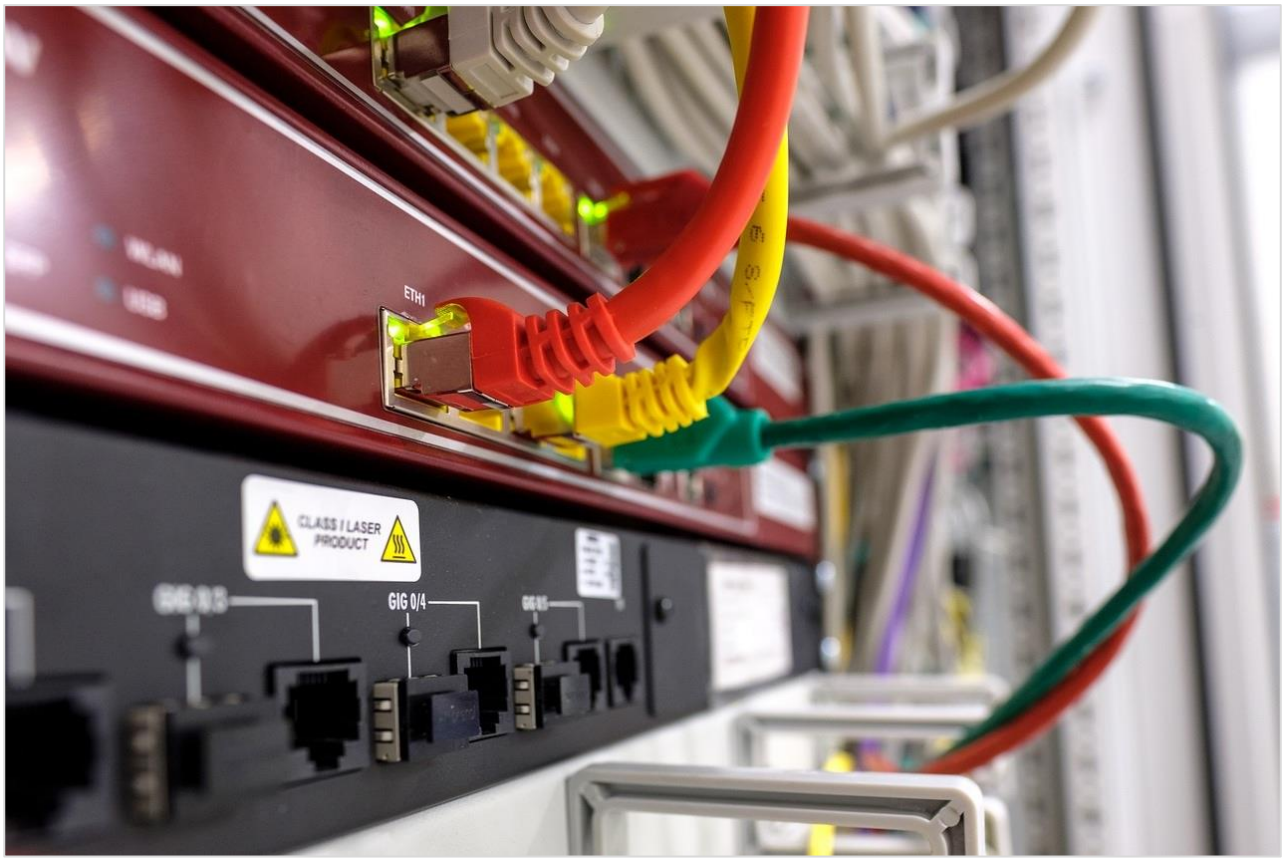
National Cyber Security Centre



Vendor security assessment

Assessing the security of network equipment.

Version 1.0

Published March 2022

Contents

Introduction

The security of network equipment is critical to the security of any network. When selecting equipment that will support a critical service or critical infrastructure, customers should make an assessment of the security of that equipment and consider that assessment as part of their procurement and risk management processes.

This guidance provides advice on how to assess the security of network equipment. It provides guidance to support public telecommunications operators (the providers of Public Electronic Communications Networks and Services), in meeting their duties under the Telecommunications (Security) Act 2021, and, when they are finalised following the Government's consultation, the Electronic Communications (Security Measures) Regulations 2022. For example, under draft Regulation 3.(3)(e), the network provider would be required:

(e) to take appropriate measures in the procurement, configuration, management and testing of equipment to ensure the security of the equipment and functions carried out on the equipment

This guidance is referenced in the draft Telecommunications Security Code of Practice, in particular draft measures 5.01 and 10.1. Whilst this guidance is not expected to form part of that code (when it is finalised) and will not be necessary or sufficient to meet new supply chain legal requirements, it is important advice that providers can use to help their compliance.

While written to support telecommunications operators, the advice within this guidance may also be useful to other providers of critical services or critical infrastructure who rely on network equipment to deliver their services. The NCSC acknowledge that the degree of assessment of the security of network equipment advised in this document is most appropriate where the network equipment is supporting a critical service. In addition, to perform the assessment described in this document effectively, customers may require appropriate contractual rights to perform the recommended audits and tests.

This guidance should be used when making selection decisions for network equipment. However, as noted below, security is an ongoing activity. As with other areas of performance, customers should continue to assess and retain evidence of the vendor's track record in security during the equipment's lifetime, as this will support future security assessments.

This guidance does not take account of, and cannot mitigate, the threats that may arise because of additional risks specific to a particular vendor in the supply chain. These risks include the degree to which it might be susceptible to being influenced or required to act contrary to the interests of the customer or their national security. In such circumstances, additional controls specific to the vendor in question may be required.

Summary of approach to assessment

This document provides guidance on how to assess a vendor's security processes and their supplied network equipment. The purpose of the approach is to objectively assess the cyber risk due to use of the vendor's equipment. This is performed by gathering objective, repeatable evidence on the security of the vendor's processes and network equipment.

Assessing the cyber risk due to a vendor requires:

- evidence from the vendor themselves
- testing to validate the vendor's claims
- third party evidence

For each criterion in this document, there are a range of product-specific spot checks that may be performed and evidence may be obtained directly from lab-tests on the product itself. These three components together will help build an understanding of how well a vendor is building a new product.

However, such an approach will always be fallible. While evidence will be customer-driven, it can only provide examples of vendor behaviour. To be effective, both the approach *and* security standards needs to be sustained over many years, with evidence of good and bad practice recorded to support future security assessments and procurement decisions.

When assessing vendor security practices, the NCSC recommends operators to not rely exclusively upon vendor documentation to assess vendor security. Security assessments should be based on the vendor's implemented security behaviour. This includes product-line specific spot checks, and objective evidence extracted from the product.

External audits and international schemes

One of the biggest challenges when assessing the security of network equipment is the industry practice of producing regional or operator-specific versions of products. Where vendors follow this practice, international customers cannot share the burden of gaining evidence or assurance about product quality or security, whether through working with each other or through international testing schemes.

It may be possible to rely on independent, external sources to provide some of the required evidence, provided:

- it is applicable to the customer's product (specifically the same hardware and code base)
- all evidence can be revalidated by the customer, and some evidence has been randomly selected to be revalidated

Generally, vendor audits or evaluations that rely on vendor documentation are unlikely to provide useful evidence unless it is possible to verify that the audit relates to the security of the network equipment. For the same reason, audits or evaluations where the evidence behind the audit is not widely available and testable should also not be considered. For example, as currently defined, the private, paper-based assessments performed under GSMA's NESAS[1] scheme are unlikely to provide useful evidence in support of the customer's assessment of product security.

[1] https://www.gsma.com/security/network-equipment-security-assurance-scheme/

Support from the security research community

Given the range, scale and complexity of network equipment participation from the global security research community (including both commercial labs and academia) is essential to support customers in understanding security risk. For this reason, vendors should be encouraged to be transparent and open about their security practices, and should be encouraged to support responsible, independent security researchers in performing their own testing and analysis.

To support the development of increasingly secure and open telecommunications equipment, DCMS has stated that it intends to establish a UK National Telecommunications Lab (UKTL). This will be a secure research facility that will bring together telecommunications operators, existing and new suppliers, academia, and the government to create representative networks in which to research and test new ways of increasing security and interoperability.

The approach to assessment

Assessing a vendor's approach to security requires a four-tiered approach:

Assess

Assessing a Security Declaration provided by the vendor. This should state the vendor's approach to security, and the security promises that the vendor makes to its customers. In the interests of developing the security ecosystem, the NCSC recommends that the vendor openly publishes their Security Declaration. This provides confidence to customers that the vendor's approach is consistent for all customers and product lines, and allows the wider security community to participate in the security discussion.

Check

Performing Spot Checks on the vendor's implemented security processes for specific, independently chosen product releases. As all details should be readily available to the vendor within their own systems, providing advance notice of the choice should not be necessary.

Analyse

Performing Lab Tests against equipment. The tests should either be against all equipment or the equipment should be randomly selected from the equipment provided by the vendor. Lab tests should be automated wherever possible so they can be easily repeated at low cost. Lab tests performed independent of the customer should be against the same product version track, hardware, software, firmware, and configuration as used by the customer.

Sustain

Holding vendors to the standard in the Security Declaration throughout the entire period of the customer's relationship with the vendor. Customers should analyse root causes of issues and record the vendor's security performance to ensure future assessments are made with a rigorous evidence base.

Recommendations for applying this four-tiered approach are provided below.

Assessing vendor security performance

When assessing vendor security practises one essential source of data is the vendor's security performance.

Customers should consider both the vendor's security culture and behaviour as evidenced by:

- maturity of vendor risk assessment and security assessment processes
- vendor transparency, openness, and collaboration with the security research community
- vendor assessment, management, and support to customers in relation to any security vulnerabilities and incidents
- vendor compliance with security obligations and requirements
- vendor approach to product and component support

Security incidents in themselves are not evidence of poor security practice. All major companies are likely to be impacted by security incidents and depending on their cause and how they are handled, security incidents may provide an example of good practice. The customer should consider whether the incident could have been reasonably avoided, or its impact could have been reasonably reduced.

Similarly, product security vulnerabilities or issues are not in themselves evidence of poor security practice as such issues will occur in all products. However, where issues are simplistic, or due to poor product management or maintenance, this may be evidence of poor practice.

Vendor security assessment criteria

The following table can be used to assist in assessing the security processes of vendors and their network equipment. The table describes the information that customers should expect within the Security Declaration, Spot Checks that should be considered to collect evidence, and the Lab Testing that customers or third parties should consider making against equipment. For Spot Checks and Lab Testing, it is assumed that the customer will be given sufficient access to vendor processes and equipment to perform an effective evaluation prior to making decisions based upon this evaluation.

When third parties are used, the customer should satisfy themselves that the third party was sufficiently independent, had sufficient technical competence, and gained sufficient information about the vendor's day-to-day practices to provide them with the confidence required reliable evidence.

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.A: Product lifecycle management					
V.A: Overall aim	The vendor's products are properly supported throughout the lifetime of the product.	To provide confidence that a product will be maturely managed by the vendor, receiving updates and security critical fixes for the supported lifetime of the product.	As part of the Security Declaration, the vendor describes how products are supported.	-	-
V.A.1: Product lifecycle process	The vendor clearly identifies the lifecycle for each product. Vendors should have an End of Life Policy which details how long products will be supported after End of Sale.	To provide confidence that products will be supported until a given date. Also, that the vendor's support dates apply globally, meaning that the vendor is likely to continue to invest in product maintenance throughout this period.	The vendor describes their product's lifecycle within the Security Declaration. For each release within a product line, the vendor publishes End of Sale dates on their website as soon as they become applicable. The End of Life Policy should detail how long, and in what way, products will be supported after the End of Sale date has been announced. The location of this information is referenced in the Security Declaration.	Check product release history. Explore how the vendor is keeping components up-to-date.	-
V.A.2: Software maintenance	Each product is maintained through its published life cycle. This maintenance, as a minimum, covers security fixes for the product.	To provide confidence that products can be patched against security issues discovered in the product throughout its supported lifetime.	The vendor clearly describes how they will support products during their lifetime, including what support they will provide under each support class.	View records showing the history of security fixes applied to the product, including a roadmap for resolution of any outstanding vulnerabilities.	Pick a sample of known vulnerabilities for a customer-selected product and check how and when they were patched in accordance with the vendor's policies. (see V.A.7).

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
					Test the product to verify that the equipment is no longer vulnerable to the vulnerability or variants of the vulnerability.
V.A.3: Software version control	Each product has a version-controlled code repository which logs every code modification. This audit log will detail: -what code has been modified, added, or removed -why the change was made -who made the change -when the change was made -which version of the code has been built into the released product.	To provide confidence to that the vendor can track exactly what code is being deployed within products. It is essential for effective investigation of supply chain attacks.	The vendor describes how they maintain the integrity of their code base.	The vendor demonstrates how changes are made based on normal processes, and how changes via other means would be rejected. Explore a change and verify that processes were followed.	
V.A.4: Software releases	Each product goes through a rigorous software release cycle including internal testing before a version is released for general availability. Software will not be released if it does not comply with the Secure Engineering requirements detailed below. Each product should have regular external testing carried out on it by an independent third party.	This requirement exists to provide confidence that vendors test their software releases and validate that their internal secure engineering processes have been followed. The tests should also ensure that previously resolved security vulnerabilities are not reintroduced.	The vendor describes their software release cycle, including the gates, and the testing performed.	View the build and test process. Review the testing performed against a customer-chosen product line and version. Check that testing tools are well configured and view the test results. Verify that tests are included to check for previously resolved vulnerabilities and issues. The vendor demonstrates that issues were correctly fixed as a result of any failed tests.	Check accuracy of a set of the vendor's test results by repeating the tests in the customer's or third party's lab.
V.A.5: Development processes and feature development	There is one primary release train of the product. Forking of new versions is minimised. Where necessary, customer-specific functionality is provided as optional modules.	This requirement exists to provide confidence that the vendor is shipping them a generally available version of the product, so they know the product can be supported throughout its lifetime using the general support routes.	The Security Declaration describes the vendor's development process, including how and when new product versions are released, and how the number of versions is kept to a manageable level.		

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
	Any new features are brought into the main product line during the standard development roadmap.	It is highly unlikely that the vendor will be able to properly support a proliferation of feature-specific product versions.			
V.A.6: International release and forking	The vendor maintains a single, global version line for each product. There are a minimal number of other versions (ideally none).	This requirement exists to provide the confidence that the product is globally supported and that any issues discovered can easily be mitigated. It is highly unlikely that the vendor will be able to properly support a proliferation of customer-specific product versions.	The vendor publishes details of all released versions of their products, including binary hashes. It is expected that this information will be on the vendor's website. The vendor references its public list of product versions within its Security Declaration.	The vendor describes the full release train of the product, including why each version was created.	Based on the vendor's published information, or otherwise, test that product versions supplied by the vendor are the 'global' versions and have matching binary hashes.
V.A.7: Use of tools, software and libraries	Third party tools (e.g. code compilers) software components and software libraries that are used within and in the development of the product are inventoried. Any of the above that are material to the security of the vendor's software are maintained throughout its lifetime.	Out-of-support tools, software components, software or libraries are unlikely to use modern security features. If exposed, they can cause known vulnerabilities to be embedded in the product. Vulnerabilities in critical security protections of the product must be patched, to minimize the impact of exploits.	The Security Declaration describes how third party software components are maintained, explicitly stating when, if ever, out-of-support components will be included in any product versions, stating justifications.	For a customer-selected product, the vendor provides a list of third party components that are material to the security of the product, (e.g. those components exposed via interfaces). Verify that these components are still actively maintained, and there is a support plan for the lifetime of the product.	Scan product interfaces to inventory known third party tools and determine if they are being maintained. Examine the product to verify that the vendor's component list appears accurate.
V.A.8: Software documentation	The vendor provides up-to-date and technically accurate documentation alongside new releases of the product. This documentation, as a minimum, shall detail how to securely configure, manage, and update the product.	This provides the customer with the information they require to help them securely deploy and manage the product throughout its lifetime in their networks, and independently assess the security of that configuration. This helps to reduce the customer's ongoing dependence on the vendor.	The Security Declaration makes commitments about the release of product documentation to customers.		Using documentation, set up, operate, configure, and update the product without support from the vendor.
V.B: Product security management					

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.B: Overall aim	Products will be developed in a 'secure by default' manner.	These requirements exist to provide confidence that a product they deploy has been developed using standard security mitigations and secure coding techniques.			
V.B.1: Security culture	The vendor has a security culture which ensures that security principles are followed.	This provides confidence that developers within the company are known to follow the security principles and development requirements.	The Security Declaration describes the senior ownership of the security culture within the vendor, and the mechanisms that exist to allow staff to raise security concerns.	-	-
V.B.2: Secure Development Lifecycle	The vendor has a Secure Development Lifecycle[2] to embed security into product development. All development teams follow, and can evidence that they follow, the Secure Development Lifecycle processes.	This provides confidence that security is embedded in the development process and that there is a consistent security culture within the company.	The Security Declaration describes how the vendor develops secure products, including how the vendor verifies that its secure coding standards are followed.	The vendor demonstrates how they gain confidence that the Secure Development Lifecycle has been followed by developers. The vendor describes how they ensure their code is of high quality. Verify examples of security controls built into the product development processes.	Search for signs that the vendor's security controls built into their Secure Development Lifecycle are working (e.g. that subcomponents are resistant to malformed inputs).
V.B.3: Internal component management	Any shared internal components or libraries are kept up to date and only the latest stable, supported version is used. These components and libraries are not to be modified for specific builds and are supported for the lifetime of the product.	This provides confidence that any internal shared components being used within a product will be maintained throughout the lifetime of the main product.	The Security Declaration makes clear commitments around the maintenance of internal components.	For a customer-selected product, the vendor can list the product's software and hardware components. Verify that only recently released versions of shared internal components and libraries are used. Explore whether the product line has forked any shared libraries.	In a lab, verify that the released product contains only one version of each internal software component or library, and that all internal components have been recently built.
V.B.4: External component management	Only supported external components are used within a product. The vendor monitors the external component's changelog so that only the latest supported, stable version is used within the product.	This provides confidence that any third party component a vendor chooses to use will be currently supported, and that any security issue discovered with the component will be patched.	The Security Declaration makes clear commitments on the use of supported external components.	For a customer-selected product release, verify that it is only using supported versions of external components and libraries. Explore how these components will be updated when they reach end-of-life.	In a lab, verify that the released product is only using fully supported versions of all external components. Search for evidence of internally-forked external components or libraries.

[2] The 'Secure Development Lifecycle' is the process through which the vendor integrates security considerations throughout the product development lifecycle.

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
	Additionally, the vendor monitors the external component's security advisories and pull in any security fixes and integrate them into their product with a security update.	Extended support contracts are likely to increase security risk and should be avoided.		Explore whether the product line has forked any externally-developed code, and if so, explore how it is maintained.	
V.B.5: Unsafe Functions	There are no unsafe functions used within the vendor's released code. Unsafe functions are those commonly associated with security vulnerabilities or those considered unsafe by industry best practise.	These functions are frequently the cause of product vulnerabilities .	The Security Declaration clearly states whether unsafe functions are used within the vendor's code base.	Request code metrics on use of unsafe functions	
V.B.6: Redundant and duplicate code	The vendor's source tree is maintained to a level that there is limited redundant or duplicate code.	Redundant code makes a product more difficult to understand and maintain. Increases the likelihood that security critical changes won't be applied to access the product.	The Security Declaration makes clear statements about how the vendor produces code to reduce complexity and increase maintainability.	Request code metrics on how much duplicated code exists within the source tree	
V.B.7: File structure	The vendor's source tree is maintained to a level where code complexity is minimised, and functions perform single, clear actions.	Code clarity reduces the risk of error or vulnerability and makes issues easier to spot.	The Security Declaration makes clear statements about how the vendor produces code to reduce complexity and increase maintainability.		
V.B.8: Debug functionality	There is no engineering debug functionality present within the vendor's released products that could weaken or bypass the product's security mechanisms.	Engineering debug functionality may be used by an attacker to exploit a product.	The Security Declaration makes clear statements confirming that no engineering debug functionality is present within a released version of a product.	Ask the vendor to demonstrate that inclusion of debug functionality within a release build results in a build failure.	
V.B.9: Comments	The source tree has suitable and understandable comments through it, explaining what the code is for and why it performs its actions.	Commenting helps ensure product can be easily supported in the future and speeds up vulnerability fixes.	The Security Declaration makes clear statements about how the vendor produces code to reduce complexity and increase maintainability.		
V.C: Protected development and build environments					

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.C: Overall aim	The NCSC expects the product is developed within a secure environment.	A secure environment helps to maintain the integrity of the product and reduces the risk of supply chain attack.	The Security Declaration describes how the vendor maintains the integrity of its products through securing the development and build environments.		
V.C.1: Segregation of development environment	Development environment is segregated from corporate network and protected from the internet.	This protects the development environment from compromise via straightforward attacks.		Ask to see details of penetration-tests or red team[3] exercises, where the objective was to modify the vendor's codebase or development environment.	
V.C.2: Segregation of build environment	Build environment is segregated from corporate network and protected from the internet. Very few people can make changes.	This protects the build environment from compromise via straight-forward attacks.		Ask to see details of penetration-tests or red team exercise , where the objective was to modify the vendor's build environment.	
V.C.3: Build environments and automation	Build environments are simple, and the build process is automated.	Simple build environments and an automated build process makes the product build easier to understand, less likely to have errors and reduces the risk of compromise.	The Security Declaration describes how the vendor build process can be understood and maintained.	For a customer-selected product release, the vendor explains the build environment and its dependencies, and demonstrates the automated process via which a build is performed.	
V.C.4: Role-based access	Only individuals with a need have access to the internal code base, and access is controlled and limited based on role.	Minimising access reduces the impact of a malicious insider.	The Security Declaration describes how the vendor enforces role-based access controls to its development and build environments.	The vendor demonstrates that access to the development and build environment is limited.	
V.C.5: Code review	All code is independently reviewed prior to acceptance. Feedback processes exist.	Code review is essential to maintaining coding standards, and to reduce the risk due to a malicious insider.	The Security Declaration describes how and when the vendor caries out internal code review and audit.	For any change made to the code, the vendor can demonstrate how that change was reviewed or audited.	-

[3] A 'red-team' exercise is one where responsible penetration testers are seeking to gain access to an asset within the vendor's network, such as their development environment.

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.C.6: Repeatable builds	All builds of released software can be replicated at a future date.	Replicated builds allow the vendor to demonstrate what components were included in a past build. Tracking of each build, what components are built into it and which versions of the components were used is critical to verifying the integrity of a build.	The Security Declaration makes clear statements about how the vendor maintains their build environment and code base to enable repeated builds with a minimal number of differences – with an explanation for each difference.	The vendor reproduces a previous build and confirms that it is functionally identical to a version that was released. The vendor demonstrates that they have retained copies of any external dependencies necessary for the build.	A released build and a reproduced build are compared to verify functional equivalence.

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.D: Exploit mitigations					
V.D: Overall aim	The vendor implements standard security mitigations used in a modern product.	Each of these mitigations has a demonstrable positive impact on the security of a product by helping to mitigate well known vulnerability classes. Modern platforms, operating systems, development languages, libraries and development tools regularly offer security enhancing technologies to both minimise the occurrence of security defects, and to minimise their impact should they occur.	The Security Declaration describes the vendor's policy with respect to the use of defensive security techniques.		
V.D.1: Heap protections	The vendor makes use of modern heap protection mitigations to help prevent heap-based memory corruption attacks against the product.	Widely used to make it more difficult for an attacker to exploit any security issues.	The Security Declaration states whether the vendor's products use heap protections throughout their product.		Verify that heap mitigations are enabled by (automatically) inspecting the product for this mitigation.
V.D.2: Stack protections	The vendor only ships executable code that has been compiled using modern stack mitigations.	Widely used to make it more difficult for an attacker to exploit any security issues.	The Security Declaration states whether the vendor's products use stack protections throughout their product.		Verify that stack mitigations are enabled by (automatically) inspecting the product for this mitigation.
V.D.3: Data execution prevention	The vendor supports hardware-enforced data execution prevention (for example DEP or NX).	Widely used to make it more difficult for an attacker to exploit any security issues.	The Security Declaration states whether the vendor's products use hardware-enforced data execution prevention throughout their product.		Verify that data execution prevention mitigations are enabled by (automatically) inspecting the product for this mitigation.
V.D.4: Address space layout randomisation	The vendor only ships executable code that has been compiled using modern ASLR techniques.	Widely used to make it more difficult for an attacker to exploit any security issues.	The Security Declaration states whether the vendor's products use ASLR throughout their product.		Verify that address space layer randomisation mitigations are enabled by (automatically) inspecting the product for this mitigation.

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.D.5: Memory mapping protections	The vendor's product will have no memory pages mapped by default as both 'Writable' and 'Executable'. This excludes areas of the code required to do Just-In-Time code compilation.	Widely used to make it more difficult for an attacker to exploit any security issues.	The Security Declaration states whether the vendor's products have any read-write memory pages. If any Just-In-Time code compilation is required, this should be described in the security declaration.		Verify that there are no executables that map memory pages as both writable and executable by (automatically) inspecting the product.
V.D.6: Least Privilege code	The vendor follows a 'least privilege' methodology when developing and executing code within their products. The vendor ensures that their product only runs at or requests the minimum privilege level required for it to fulfil its advertised purpose. If higher privilege levels are ever required, then the product implements segregations to elevate privilege for the specific task.	Products that run at higher privilege levels than required can provide a route for attackers to exploit a host system.	The Security Declaration states the vendor's 'least privilege' methodology.		Verify that executable code running on the vendor's platform runs with the least level of privilege required. Verify that any privileged executables drop privilege after completing their privileged task.
V.D.7: Security improvement and secure execution environments	The vendor has plans to continue to improve its product's security. As an example, this may include detailing how and when they plan to implement secure execution environments[4].	Product security needs to continue to evolve to keep pace with the threat environment.		Explore the vendor's future security roadmap, discussing how the vendor's product security will increase over time.	
V.E: Secure updates and software signing					
V.E: Overall aim	The source of the code that runs on the device is known, and the mechanisms to change the code on the device are secure.	Reduces the risk of supply chain attack between code production by the vendor, and delivery to the device.			

[4] Secure execution environments are a significant upcoming security technology that increases product security by enabling execution of sensitive workloads on untrusted hardware.

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.E.1: Software and firmware signing	Vendor's software and firmware is digitally signed.	Signing of software and firmware provides strong evidence that the developer produced the code.	The Security Declaration describes whether software and firmware are digitally signed, and any processes for allowing customers to deploy their own code.		Test that shipped executable code (binaries, scripts, etc) are digitally signed using the vendor's public code signing certificate by automatically inspecting each file.
V.E.2: Signature verification	Software signatures are verified before binaries are executed.	Allows the device to check the source of the code.	The Security Declaration describes how signatures are checked prior to code execution. States whether that check is hardware backed.		Test that a modification of a signed binary results in the device refusing to run the binary.
V.E.3: Secure update	Updates are delivered via a secure channel that is mutually authenticated between the device and the update server.	Using a secure channel reduces the risk of an attacker exploiting the update mechanism.	The Security Declaration describes the security properties of the update mechanism.		Perform the update process, verifying that updates are delivered over a secure channel.
V.E.4 Downgrade protection	Built-in detection capabilities alert whenever software is downgraded during an install process.	Publicly known vulnerabilities in an older version of the product are common causes of exploit and compromise.	The Security Declaration describes how downgrade attacks are prevented by the vendor.		Test that a signed update which is of an older version to that currently installed produces a log message or other alert likely be seen by the system administrator.
V.F: Hardware roots of trust and secure boot					
V.F: Overall aim	The vendors use a secure hardware root of trust within their products. These are commonly referred to as one of the following: TEE (Trusted Execution Environment), TPM (Trusted Platform Module), or DSC (Dedicated Security Component).	A hardware root of trust enables the vendor to use modern security mitigations such secure boot and code signing.	The Security Declaration describes the vendor's approach to the provision of hardware-backed security.		
V.F.1: Hardware root-of-trust	The equipment contains a hardware root-of-trust for identity and storage.	A hardware root-of-trust is necessary to provide hardware-backed functionality that cannot be remotely modified by an attacker.	The Security Declaration states the presence and properties of any hardware root of trust with the products.	-	Test that private keys associated with identity or device secrets are not stored in the filesystem in clear text.

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.F.2: Secure Boot	Each product will support a secure boot process, initiated by the hardware root-of-trust (V.F.1) to bring the equipment into a known good state on restart.	Secure boot makes it harder for any compromise of the device to persist after a power cycle. Should devices be compromised, secure boot is required to restore trust in the equipment. Otherwise, the equipment may need to be scrapped.	The Security Declaration describes the vendor's support of a secure boot, and how the vendor's products can be returned to a known good state in the event of compromise.	-	Verify that the product can be returned to a known good state. Test that the device fails to boot should one or more of the signed binaries or scripts used during the boot process be modified.
V.F.3: Securing JTAG	Each compute element on a product will have debug interfaces (such as JTAG and UART) access disabled.	With physical access, debug interfaces like JTAG can be used to circumvent the integrity of a product or steal device secrets.			Test that JTAG equipment cannot establish communication with any of the system's JTAG TAP controllers.
V.G: Security testing					
V.G: Overall aim	The vendor rigorously tests the security of their products prior to release.	Through security testing and resolution, the number of vulnerabilities in the product is reduced, as is the risk of exploitation.	The Security Declaration describes the vendor's approach to security testing across its product range.		
V.G.1: Automated testing	Once developed, extensive security tests are automatically run against all versions of applicable products.	This ensures that testing is at a scale comparable to that employed by an attacker.	The Security Declaration describes the automated tests run against every product version.	For a customer-chosen product release, ask to see the test results from automated testing.	The customer, or third party, applies their own automated tests where possible.
V.G.2: Testing rigour	Developers cannot modify the build environment to hide or disregard build issues, or issues detected by automated tests. Failing builds are automatically rejected. Therefore, code used in released products do not create any compiler errors or security related warnings during build.	Developers may seek to bypass checks if permitted, leading to more vulnerable products.	The Security Declaration states whether tests can be bypassed.	For a customer-chosen product release, ask to see build results. Verify that the results do not highlight issues that should not be accepted in a released build.	
V.G.3: Security Testing	Security functionality is tested to demonstrate correct operation.	If security functionality is mis-implemented, the device will likely be vulnerable.	The Security Declaration states whether security testing is performed to verify correct operation.	For a customer-chosen product release, ask to see the results from security testing. Verify that issues were resolved, including root-causes.	Repeat tests of security functionality.

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.G.4: Negative testing[5]	Extensive negative testing is performed against every product release, including a wide range of potential failure cases, inappropriate message sequencing and malformed messages.	By testing with extensive negative test cases, the vendor is more likely to catch easy-to-detect issues.	The Security Declaration states whether negative testing is performed and describes the scale of this testing.	For a customer-chosen product release, ask to see the test results from negative testing. Verify that issues were resolved, including root-causes.	Perform negative tests against the product, ideally using a distinct toolset to the vendor.
V.G.5: Fuzzing[6]	Fuzzing is performed against the product, especially focusing on interfaces which cross security boundaries. The approach is sophisticated enough to ensure that a high proportion of code is tested.	A specific form of negative testing, the vendor tests their products against randomly-generated, malformed data, to catch easy-to-detect issues.	The Security Declaration states whether fuzz testing is performed and indicates the scope of this testing.	For customer-chosen product release, ask to see the test results from fuzzing, alongside data on code coverage. Verify that issues were resolved, including root-causes.	Perform fuzzing of the product, ideally using a distinct toolset to the vendor.
V.G.6: External testing	External security research teams perform testing against a selection of major product releases. Some of this testing is un-scoped.	By subjecting the device to an external third party, vulnerabilities are more likely to be detected and remediated.	The Security Declaration contains explicit details about how the vendor partners with external labs and academics to ensure the security of their products is independently tested.	Ask to see the results from external tests. Verify that issues were resolved, including root-causes.	
V.G.7: Dynamic application security testing (DAST)[7]	The vendor has a DAST solution integrated into the vendor's test process.	Applying DAST during testing can identify different types of vulnerabilities to that of fuzzing and negative testing.	The Security Declaration states how the vendor performs dynamic application security testing.	Ask to see the results from the DAST suite. Verify that issues were resolved, including root-causes.	Perform dynamic application security testing on the product, ideally using a distinct toolset to the vendor.
V.H: Secure management and configuration					
V.H: Overall aim	Any product can be easily set up to run securely.	Insecurely configured products are more likely to be exploited.	The Security Declaration describes the vendor's approach to helping operators securely configure products. This includes whether products are released in a 'secure' configuration.		

[5] 'Negative Testing' is the testing of failure conditions to check they are handled gracefully by the equipment.

[6] 'Fuzzing' is a testing technique that involves providing invalid, unexpected, or random data to check that these inputs are handled gracefully by the equipment.

[7] Dynamic Application Security Testing (DAST) a procedure that actively investigates running applications with penetration tests to detect possible security vulnerabilities.

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.H.1: Product hardening	The product can be easily hardened into a secure configuration. Documentation exists to help customers perform this hardening process. Alerts are created should the device be taken out of the hardened state.	Insecurely configured products are more likely to be exploited.	The Security Declaration states whether products can be easily hardened into a secure configuration.	Verify that guidance is provided on secure configuration for provided products.	Test that the hardening guide can be easily deployed as-is to the product without impacting necessary functions. Test that alerts are created should the device be taken out of the hardened state.
V.H.2: Protocol Standardisation	The product can be configured to only use standardised protocols.	Proprietary protocols do not allow for thorough, independent security testing, or correct behaviour to be understood by the customer.			Analyse traffic from the equipment to ensure that there are no proprietary protocols in use.
V.H.3: Management plane security	By default, the product is configured to only use up-to-date, secure protocols on the management plane.	Without secure protocols and user-based access it is not possible to securely manage equipment and associate administrative changes with a specific administrator.	The Security Declaration confirms whether the product only uses secure management protocols by-default.		Test that no weak or deprecated security protocols are enabled on the management plane.
V.H.4: Management access	Access to the management plane is user-based and supports asymmetric-key-based (e.g. X.509 certificates or SSH keys).	This allows customers to limit administrative privilege and investigate potentially malicious changes. The use of asymmetric key based authentication allows for more secure authentication and helps mitigate the risk of password sharing.			Test that the management plane gives administrators user-based access and supports asymmetric-key-based authentication.
V.H.5: No unencrypted protocols	Secure protocols are used whenever possible (e.g. SSH and HTTPS). If an unencrypted protocol is enabled, and a secure alternative exists, the product warns the administrator, and provides the option to create a security alert.	To prevent the use of insecure protocols, which increases the risk of exploitation.			Test that there are no unencrypted protocols and services are enabled by default on the product. Test that enabling an unencrypted protocol on the product results in appropriate warnings and alerts.

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.H.6: No undocumented administrative mechanisms	The product does not have any undocumented administrator accounts. Examples include, but are not limited to, hard coded passwords, access key pairs (SSH keys) or other administrative access tokens.	Undocumented administrative accounts may be exploited without customer awareness.	The Security Declaration explicitly states whether there are any undocumented administrative accounts on the product.		Search for evidence of undocumented administrator accounts in released products.
V.H.7: No undocumented administrative features	The product does not have any undocumented administration features.	Undocumented administrative features may be exploited without customer awareness.	The Security Declaration explicitly states whether there are any undocumented administrative features on the product.		Search for evidence of undocumented administrator features in released products.
V.H.8: No default credentials	No default passwords are left on the device after the initial setup. For clarity, this also means there are no administrative accounts coded into the vendor's software.	Failure to disable any non-unique or hardcoded accounts renders the equipment highly vulnerable to exploitation.	The Security Declaration explicitly states how default credentials are removed from all devices, and whether hard-coded administrative accounts exist.		Test that there are no default credentials on the device after initial setup. Scan products for potential hardcoded password strings.
V.H.9: Good Practice Guidance	The vendor is explicit about the threats to the equipment that they have sought to mitigate, and those they have not. The vendor provides detailed configuration and notes on how the equipment can be protected in networks.	By helping understand the security decisions taken by the vendor, and set up the equipment securely, security mistakes are less likely to be made.	The Security Declaration describes the vendors approach to security analysis, and how they support customers in minimising risk.	For a customer-chosen product, explore the vendor's product security analysis, and consider whether the vendor has understood the risk environment and established appropriate mitigations.	
V.J: Vulnerability and Issue Management					
V.J: Overall aim	Effective processes exist to manage security issues and vulnerabilities. These issues are quickly and effectively resolved.	Products are most vulnerable from when an issue is discovered until it is patched. Effective issue management reduces this risk.	The Security Declaration describes the vendors approach to resolving issues.		
V.J.1: Issue tracking and remediation	The vendor has a process for issuing remediation. This ensures the vulnerability is resolved in all impacted products. Vulnerabilities are patched within appropriate timeframes.	If issues are not resolved across all versions of all product lines, the same issue may continue to be exploitable in some product version.	The Security Declaration provides the vendor's timescales on the resolution of security issues and describes how the vendor traces vulnerabilities across all products.	Assuming a software component is vulnerable, ask to see all products that contain that component.	Test whether a previously reported and resolved vulnerability may still be exploited across a range of products.

Topic	Security expectation	Why it matters	Evaluation: security declaration	Evaluation: customer or 3rd party spot checks	Evaluation: customer or 3rd party lab test
V.J.2: Issue comprehension.	For issues, the vendor identifies the root cause analysis of the issue and is able to detail the origin of the vulnerability.	Proper vulnerability management requires the vendor to understand its own product and quickly assess impact of a vulnerability.		For a customer-chosen vulnerability, the vendor can provide details of the vulnerability, the root cause of the vulnerability, and how and when the vulnerability was correctly resolved.	
V.J.3: Vulnerability reporting	The vendor provides a publicly advertised route for disclosure of security issues that links into their vulnerability management process.	This allows external people and organisations to responsibly disclose security issues to the vendor.	The Security Declaration describes how vulnerabilities may be reported to the vendor.	Explore how the vendor resolved a previously reported issue.	
V.J.4: Issue transparency	The vendor is transparent about their patching of security issues.	In the sector, most security issues are patched without customers becoming aware of their existence. This makes it difficult for customers to judge risk.	The Security Declaration provides metrics on security issues, both reported and resolved. A list of all patched security issues in the product is available.		
V.J.5 Product Security Incident Response Team (PSIRT)[8]	The vendor has set up the PSIRT structures within its organisation.	Product security is not restricted to R&D. PSIRT brings together R&D, QM, TAC, OPS to be responsible for secure product operation by customers.	The Security Declaration describes how to contact vendor's PSIRT team.	Ask the vendor for Product Security Incident Response plan of selected release.	When vulnerabilities are found during lab testing, report these to the PSIRT team and verify that the vendor's response is effective.

[8] Product Security Incident Response Team (PSIRT) is the common name for the vendor's team that handles the receipt, investigation and public reporting of security vulnerability information relating to the vendor's products.