Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>June 2011</td>
<td>Initial Draft</td>
</tr>
<tr>
<td>0.91</td>
<td>Sept 2011</td>
<td>Second Draft</td>
</tr>
<tr>
<td>1.0</td>
<td>March 2012</td>
<td>Third Draft</td>
</tr>
<tr>
<td>1.3</td>
<td>May 2012</td>
<td>Release Version</td>
</tr>
</tbody>
</table>

This Security Characteristic is derived from the following files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Application Firewall - v1.2.cxl</td>
<td>1.2</td>
</tr>
<tr>
<td>Common Libraries - v1.6.cxl</td>
<td>1.6</td>
</tr>
<tr>
<td>Firewall –v0.1.cxl</td>
<td>0.1</td>
</tr>
<tr>
<td>Generic Network Device – v0.3.cxl</td>
<td>0.3</td>
</tr>
<tr>
<td>Hardware Libraries v1.3.cxl</td>
<td>1.3</td>
</tr>
<tr>
<td>Passphrase Libraries – v2.cxl</td>
<td>2.0</td>
</tr>
</tbody>
</table>

Soft copy location

DiscoverID 18397081

This document is authorised by:

Deputy Technical Director (Assurance)
CESG

This document is issued by CESG

For queries about this document please contact:

CPA Administration Team
CESG
Hubble Road
Cheltenham
Gloucestershire
GL51 0EX
United Kingdom

Tel: +44 (0)1242 221 491
Email: cpa@cesg.gsi.gov.uk

The CPA Authority may review, amend, update, replace or issue new Scheme Documents as may be required from time to time.
UNCLASSIFIED

CPA Security Characteristics for Web Application Firewalls

CONTENTS

REFERENCES ........................................................................................................................................ iv

I. OVERVIEW ...................................................................................................................................... 1
   A. Product Aims ................................................................................................................................ 1
   B. Typical Use Case(s) ..................................................................................................................... 1
   C. Expected Operating Environment ............................................................................................ 2
   D. Compatibility ............................................................................................................................... 2
   E. High Level Functional Components .......................................................................................... 2
   F. Future Enhancements ................................................................................................................ 3

II. SECURITY CHARACTERISTIC FORMAT ............................................................................... 4

III. REQUIREMENTS ......................................................................................................................... 5
   A. Design Mitigations ....................................................................................................................... 5
   B. Verification Mitigations .............................................................................................................. 14
   C. Deployment Mitigations ............................................................................................................. 17

IV. GLOSSARY .................................................................................................................................... 22
REFERENCES

[a] The Process for Performing Foundation Grade CPA Evaluations, v1.3, August 2011, CESG

I. OVERVIEW

1. This document is a CPA Security Characteristic – it describes requirements for a particular type of assured product for evaluation and certification under CESG’s Commercial Product Assurance (CPA) scheme.

A. Product Aims

2. A Web Application Firewall (WAF) protects a Web Server and Web Applications running on it from certain types of attack.

3. The WAF described by this Security Characteristic is capable of preventing a range of malicious communications from reaching a protected web server and the applications running on it. The WAF is also capable of preventing responses containing undesired data being communicated from the protected Web Server and reaching a client. It is acting as a specialised Intrusion Prevention System (IPS), rather than an Intrusion Detection System (IDS). IDS type products do exist which are sometimes described as Web Application Firewalls, but these are not capable of preventing malicious requests from reaching the protected application.

B. Typical Use Case(s)

4. A WAF is expected to sit between a less trusted network and a web server. Its role is to protect both the server, and web applications running on it, from attack. This is achieved by actively blocking content deemed malicious, as defined within the WAF’s ruleset.

5. A WAF takes the form of an appliance which allows an administrator to implement a set of defences tailored to the requirements of their particular web application(s). The ruleset needs to be developed for each individual web application, although some predefined templates or rules may be present to assist with this.

6. A WAF seeks to protect a web server, the web applications running on it, and database servers running on, or connected to the web server from certain types of attack. The WAF must operate effectively without the application or server it is protecting being aware of its presence. The application should not need to be developed with the WAF in mind.

7. This Security Characteristic specifically excludes from scope products protecting clients (e.g. web browsers) from malicious web servers.

8. IDS Type devices, which cannot prevent malicious communications from reaching the web server, are also specifically excluded from this characteristic.
C. Expected Operating Environment

9. WAF’s are only expected to be used within large enterprise networks as part of a defence in depth design.

D. Compatibility

10. The WAF should be compatible with any HTTP or HTTPS web server. Various products will also understand one or more content types such as HTML, XML and JSON.

11. A WAF should have the ability to decrypt and apply its ruleset to the contents of TLS/SSL VPNs and optionally, re-encrypt before forwarding it to the web application server. This can be achieved by using the correct certificates and acting as a “Man-in-the-middle” between client and server.

E. High Level Functional Components

12. The diagram below is used to illustrate the various high-level functional components which the mitigations listed in this Security Characteristic refer to. These are used to structure the Security Characteristic, and to give context to each mitigation.

13. Rules Configuration: The rules which are set or configured by the administrator to protect a specific web application from attack.

14. Device Configuration: The settings which allow the device to successfully interact on the network, such as network interface settings.

15. HTTP Parser: The element which allow the WAF to inspect HTTP content and apply rules.

16. Logging: The recording and reviewing of security related events.
17. **Network Interface:** The network interfaces which allow the device to connect to the network containing the web server to protect, the less trusted network, and a management network.

F. **Future Enhancements**

18. CESG welcomes feedback and suggestions on possible enhancements to this Security Characteristic
II. SECURITY CHARACTERISTIC FORMAT

19. All CPA Security Characteristics contain a list of mitigations which are split into three requirement categories: development, verification and deployment requirements. Within each of these sets the mitigations can be grouped based on areas of the product (as illustrated in the High Level Functional Component Diagram above), such as bulk encryption or authentication, or they may be overarching requirements which apply to the whole product. Reference [a] describes how evaluation teams should interpret Security Characteristics.

20. The three types of mitigations are denominated as follows:

- **DEV** – These are mitigations that are included by the developer during the design or implementation of the product. These are validated via a review of the product’s design or implementation during a CPA evaluation.

- **VER** – Verification mitigations are specific mitigations that the evaluator must test during the assessment of the product.

- **DEP** – Deployment mitigations are points that must be considered by users or administrators during the deployment of the product. These mitigations are incorporated into the security procedures for the product.

21. Each mitigation includes informational text in italics, describing the threat that it is expected to mitigate. It also lists at least one specific mitigation, which describes what must actually be done to achieve that requirement. In some cases there is additional explanatory text which expands upon these requirements.

22. In the requirements listed below, the following terminology can be used:

- ‘Must’, ‘Mandatory’ and ‘Required’ are used to express a mitigation that is essential. All mitigations and detailed mitigations are mandatory unless there is an explicit caveat, such as ‘if supported by the product’.

- ‘Should’ and ‘Strongly Recommended’ are used whenever a requirement is highly desirable, but is not essential. These are likely to become mandatory in future iterations of the Security Characteristic.

- ‘Could’ and ‘Recommended’ are used to express a non-mandatory requirement that may enhance security or functionality.

23. For example:

**DEV.M1:** [A mitigation]

This mitigation is required to counter [a threat].

At Foundation the product must [do something].

This can be achieved by [explanatory comment].
III. REQUIREMENTS

A. Design Mitigations

DEV.M41: Crash reporting
  This mitigation is required to counter exploitation of a software implementation error.
  At Foundation Grade the product is required to ensure crashes are logged.
  Where it is possible that sensitive data may end up in the crash data, this must be handled as red data and must only be available to an administrator. Crash data from both the product and the underlying operating system must be considered.

DEV.M42: Heap hardening
  This mitigation is required to counter exploitation of a software implementation error.
  At Foundation Grade the product is required to use the memory management provided by the operating system. Products should not implement their own heap.

DEV.M43: Stack protection
  This mitigation is required to counter exploitation of a software implementation error.
  At Foundation Grade the product is required to be compiled with support for stack protection in all libraries, where the tool chain supports it.
  If more recent versions of the tool chain support it for the target platform then they should be used in preference to a legacy tool chain.

DEV.M46: User least privilege
  This mitigation is required to counter taking advantage of existing user privilege.
  At Foundation Grade the product is required to operate correctly from a standard account without elevated privileges.

DEV.M159: Update product
  This mitigation is required to counter exploitation of a software logic error.
  This mitigation is required to counter exploitation of a software implementation error.
  At Foundation Grade the product should support the use of software updates.

DEV.M321: Data Execution Protection
  This mitigation is required to counter exploitation of a software implementation error.
  At Foundation Grade the product is required to support Data Execution Protection (DEP) when enabled on its hosting platform and must not opt out of DEP.
  If the product is to be specifically deployed on a platform that does not support either Software DEP or Hardware-enforced DEP, there is no requirement for DEP compatibility.

DEV.M340: Address Space Layout Randomisation
  This mitigation is required to counter exploitation of a software implementation error.
  At Foundation Grade the product is required to be compiled with full support for ASLR, including all libraries used.
  ASLR may be disabled for specific aspects of the product, provided there is justification of why this is required.
DEV.M355: Secure software delivery
This mitigation is required to counter installation of malware on host
This mitigation is required to counter installing compromised software using the update process
At Foundation Grade the product should be distributed via a cryptographically protected mechanism, such that the authenticity of software can be ensured.
Initial code for the product, and any subsequent updates, must be distributed in such a way that tampering is cryptographically detectable. The recipient of the software must be able to ensure the identity of the originator (ie vendor).

DEV.M605: Store manufacturer's public key securely
This mitigation is required to counter modification of the manufacturer’s public key on device
At Foundation Grade the product is required to ensure there are no methods to gain unauthorised access to keys on the device.

DEV.M621: Ensure product fails securely
This mitigation is required to counter high valid traffic volumes
This mitigation is required to counter use of malformed/unusual traffic
At Foundation Grade the product is required to handle any failures in a secure manner.
The firewall must not allow traffic to traverse the firewall if the device has failed.

DEV.M622: Log unusual traffic
This mitigation is required to counter high valid traffic volumes
This mitigation is required to counter use of malformed/unusual traffic
At Foundation Grade the product is required to record unusual traffic to a log.
Malformed traffic is traffic which the firewall is not capable of dealing with correctly.

High valid traffic volumes are volumes of traffic, at which the firewall will be unable to operate correctly.

DEV.M624: Raise alerts
This mitigation is required to counter use of malformed/unusual traffic
At Foundation Grade the product is required to raise alerts on unusual events.
Unusual events could be high traffic volumes, audit logs reaching maximum capacity amongst others.

Alerts could also be raised when an event exceeds an administrator defined threshold.

DEV.M636: Remove or disable all software debug interfaces
This mitigation is required to counter use of debug mode or ruleset bypass function
At Foundation Grade the product is required to have no useable software debugging interfaces available in the final manufactured product.

DEV.M637: Retain data on power loss
This mitigation is required to counter exploitation of incorrect operations of device
At Foundation Grade the product is required to ensure that operationally important data is not lost from power loss.
This includes important data such as:
- Firewall configuration
- Logging / auditing data
- Firewall rulesets.
DEV.M659: Drop erroneous or excess requests

This mitigation is required to counter high invalid traffic volumes

At Foundation Grade the product is required to discard requests if the functionality of the device is at risk.

If the WAF cannot process any more requests, the packets must be discarded. For example, dropping excess incoming requests until the WAF is able to handle any further requests.

Malformed requests must always be dropped and not be processed further.

DEV.1 - Design >> Rules Configuration

DEV.1.M609: Configuration export

This mitigation is required to counter loss of configuration data due to modification

At Foundation Grade the product should provide the ability to export and import the product configuration.

DEV.1.M611: Configuration only by administrators

This mitigation is required to counter modification of rules without valid administrator credentials

At Foundation Grade the product is required to ensure that only an authenticated administrator can change device configuration settings.

DEV.2 - Design >> Logging

DEV.2.M612: Sanitise logged data

This mitigation is required to counter supplying a malicious script through logged data

At Foundation Grade the product is required to ensure logged data is appropriately sanitised prior to display.

The method and content of sanitisation will change depending on the content in the logs and where the logs are displayed. For example, output to a HTML viewer for the logs will need to be encoded whereas logging output to a text file may not need to be sanitised.

DEV.2.M627: Protect access to logs

This mitigation is required to counter modification of logging generation

This mitigation is required to counter sanitisation of illegitimate access from logs

At Foundation Grade the product is required to provide ability to automatically push logs to external device.

At Foundation Grade the product is required to ensure that only an authenticated administrator can manage logs.

At Foundation Grade the product is required to not overwrite logs without alerting the administrator.

At Foundation Grade the product is required to ensure that all logs are time stamped.

Timestamps must be accurate and the deployment must take measures to ensure this.

Such measures could be NTP synchronisation or a manual process.
DEV.3 - Design >> HTTP Parser

DEV.3.M41: Crash reporting
This mitigation is required to counter exploitation of a software implementation error.
At Foundation Grade the product is required to ensure crashes are logged. Where it is possible that sensitive data may end up in the crash data, this must be handled as red data and must only be available to an administrator. Crash data from both the product and the underlying operating system must be considered.

DEV.3.M42: Heap hardening
This mitigation is required to counter exploitation of a software implementation error.
At Foundation Grade the product is required to use the memory management provided by the operating system. Products should not implement their own heap.

DEV.3.M43: Stack protection
This mitigation is required to counter exploitation of a software implementation error.
At Foundation Grade the product is required to be compiled with support for stack protection in all libraries, where the tool chain supports it.
If more recent versions of the tool chain support it for the target platform then they should be used in preference to a legacy tool chain.

DEV.3.M159: Update product
This mitigation is required to counter exploitation of a software logic error.
This mitigation is required to counter exploitation of a software implementation error.
At Foundation Grade the product should support the use of software updates.

DEV.3.M321: Data Execution Protection
This mitigation is required to counter exploitation of a software implementation error.
At Foundation Grade the product is required to support Data Execution Protection (DEP) when enabled on its hosting platform and must not opt out of DEP.
If the product is to be specifically deployed on a platform that does not support either Software DEP or Hardware-enforced DEP, there is no requirement for DEP compatibility.

DEV.3.M340: Address Space Layout Randomisation
This mitigation is required to counter exploitation of a software implementation error.
At Foundation Grade the product is required to be compiled with full support for ASLR, including all libraries used.
ASLR may be disabled for specific aspects of the product, provided there is justification of why this is required.

DEV.3.M639: Drop traffic that does not conform to ruleset
This mitigation is required to counter exploitation of incorrect operations of device.
At Foundation Grade the product is required to only allow traffic which adheres to rules to traverse the device.
DEV.3.M641: Ensure device denies traffic on start up  
*This mitigation is required to counter exploitation of incorrect operations of device.* 
At Foundation Grade the product is required to only allow packets to traverse the device when the device is fully operational. The device may take time to load rules and configuration data at start up, therefore the firewall shouldn't process traffic until these are loaded. This also applies to re-starting/rebooting of device.

DEV.3.M642: Identify unexpected served content  
*This mitigation is required to counter modification of files on the web server(s).* 
At Foundation Grade the product is required to allow the administrator to define rules that detect unexpected served content. Available rule-types should include string matching by regular expression and DOM fingerprinting. Matching these rules is required to identify and drop any unexpected served content such as defacements.

DEV.3.M644: Requests can be filtered based on the contents of HTTP parameters  
*This mitigation is required to counter a Cross Site Scripting attack.* 
This mitigation is required to counter supplying malicious parameters. At Foundation Grade the product is required to allow rules to be set which enable request parameters to be validated using a whitelist. Rules should be able to be set per URL. Rules should assist in the prevention of common attacks, for example, ensuring that requests with unexpected additional HTTP parameters or those listed in the OWASP Top 10: http://www.owasp.org. At Foundation Grade the product is required to allow rules to be set to whitelist request parameters based on intended web application. Blacklists should also be supported when whitelisting is not possible within an organisational deployment. At Foundation Grade the product is required to allow the Administrator to choose whether to drop or rewrite requests and responses which don't conform to the ruleset.

DEV.3.M645: HTTP input sanitisation  
*This mitigation is required to counter supplying malicious parameters.* 
This mitigation is required to counter a Cross Site Scripting attack. At Foundation Grade the product is required to sanitise HTTP content. The product must provide an Administrator with the capability to configure the device to sanitise data must be sanitised such that any malicious injected non-conformant content is removed before being stored by the underlying web application. The type of sanitisation will depend on the intended use of the data, such as database storage or for display on a web page as HTML. The product is required to allow an administrator to choose whether to sanitise or drop (see DEV.3.M644) as appropriate.

DEV.3.M646: Inbound and outbound traffic malicious content checking  
*This mitigation is required to counter uploading of malicious content to Web Application Server.* 
This mitigation is required to counter downloading of malicious content to client. At Foundation Grade the product is required to identify the mimetype of inbound content and support a whitelists of allowed mimetypes.

This information is exempt under the Freedom of Information Act 2000 (FOIA) and may be exempt under other UK information legislation. Refer any FOIA queries to GCHQ on 01242 221491 x30306 or infoleg@gchq.gsi.gov.uk.
DEV.3.M647: Requests can be filtered based on HTTP method

This mitigation is required to counter use of an unexpected HTTP method
At Foundation Grade the product is required to filter based on HTTP method and must support a whitelist of allowed HTTP methods per URL.
At Foundation Grade the product is required to allow the Administrator to choose whether to drop or rewrite requests and responses which don't conform to the ruleset.

DEV.3.M648: Prevent data exfiltration

This mitigation is required to counter use of an unexpected HTTP method
This mitigation is required to counter supplying malicious parameters
This mitigation is required to counter accessing a prohibited URL
This mitigation is required to counter supplying a malformed HTTP request
At Foundation Grade the product is required to allow the Administrator to define rules to detect unexpected served content.
Available rule-types should include string matching by regular expression, response size checking, and DOM fingerprinting. Matching these rules is required to identify and drop any unexpected served content

Web Application responses must be checked for any unauthorised/sensitive data being sent from the web application, as this could indicate an attack is taking place to extract sensitive data.

Heuristics could be used, and include checking response sizes, key words and regular expression matching, for example National Insurance numbers or credit card numbers.

DEV.3.M649: WAF authentication

This mitigation is required to counter supplying malformed headers
This mitigation is required to counter supplying malicious parameters
This mitigation is required to counter use of an unexpected HTTP method
This mitigation is required to counter accessing a prohibited URL
This mitigation is required to counter supplying a malformed HTTP request
At Foundation Grade the product should provide the ability to ensure that only requests authenticated using SAML can reach the web server.

DEV.3.M651: HTTP protocol can be checked for correctness

This mitigation is required to counter supplying malformed headers
This mitigation is required to counter supplying a malformed HTTP request
At Foundation Grade the product should ensure that HTTP requests and responses that reach the server are fully compliant with RFC2616.
At Foundation Grade the product is required to ensure the HTTP protocol is fully parsed and reconstructed.
At Foundation Grade the product is required to allow the Administrator to choose whether to drop or rewrite requests and responses which don't conform to the ruleset.
DEV.3.M652: Enforce additional cookie security
This mitigation is required to counter stealing cookies
At Foundation Grade the product should monitor and control the list of valid cookies per web application, by default.
This mitigation can be achieved by maintaining a list of currently valid cookies, in cases where those cookies always originate on the server. For example when the server first sets a cookie it is added to the list, it is removed from the list when it expires or when it is unset or replaced by the server. Invalid cookies in request headers must not reach the application.

Cookies which are expected to be generated and set on the client, for example using JavaScript, can be excluded from these checks by the admin.
At Foundation Grade the product is required to allow an administrator to set the HTTP-only flag on each cookie.
At Foundation Grade the product is required to allow an administrator to set the secure flag on each cookie.

DEV.3.M653: Requests can be filtered based on the HTTP header referrer field
This mitigation is required to counter a Cross Site Request Forgery attack and other forced browsing attacks
At Foundation Grade the product is required to allow filtering rules to be set based on a whitelist of valid referrer URLs for each request URL.

DEV.3.M656: Requests can be filtered based on URL
This mitigation is required to counter accessing a prohibited URL
At Foundation Grade the product is required to have a configurable URL whitelist available per web application.

DEV.3.M658: Filtering of HTTP response headers
This mitigation is required to counter fingerprinting of HTTP headers
At Foundation Grade the product is required to allow specific HTTP response headers to be edited or removed.
HTTP response headers must be inspected for any unauthorised/sensitive data relating to the configuration of the web application or server.

DEV.4 - Design >> Hardware

DEV.4.M44: (Hardware ONLY) Data validation on untrusted input
This mitigation is required to counter exploitation of product through user-provided input
At Foundation Grade the product is required to validate interface input commands before attempting to process them.
For example, random inputs must not cause insecure behaviour.

DEV.4.M109: (Hardware ONLY) Protection of sensitive data lines
This mitigation is required to counter installation of hardware-level malware
At Foundation Grade the product is required to ensure physical access to internal data lines carrying sensitive data requires breaching of the tamper protection.
In this context, sensitive data is defined as key material, user data and configuration data.

DEV.4.M161: (Hardware ONLY) Anti-Tamper mechanisms
This mitigation is required to counter probing of management circuits and reading raw storage
At Foundation Grade the product is required to block access to sensitive hardware and test ports.
The anti-tamper mechanism should provide sufficient protection against an attacker using a standard set of tools such as screwdrivers and voltage meters.
DEV.4.M167: (Hardware ONLY) Control use of internal reconfiguration ports
This mitigation is required to counter exfiltration/modification of the product configuration
At Foundation Grade the product is required to control the use of internal reconfiguration ports of the product.
For example, the reconfiguration ports should be within the anti-tamper boundary of the product or should be removed from the final manufactured product.
NOTE: This only applies to FPGA implementations.

DEV.4.M169: (Hardware ONLY) Disable test ports
This mitigation is required to counter exploitation of debug/testing features
At Foundation Grade the product is required to have all hardware testing ports, such as JTAG connectors, disabled.

DEV.4.M313: (Hardware ONLY) Use security and intellectual property protection
This mitigation is required to counter exfiltration/modification of the product configuration
At Foundation Grade the product should use any available security and intellectual property protection features provided by the hardware.
Any security features should be configured to be on by default.
NOTE: This only applies to FPGA implementations.

DEV.4.M314: (Hardware ONLY) Prevent access to security critical components
This mitigation is required to counter reconfiguring security functionality
At Foundation Grade the product is required to house critical security components within the tamper boundary.

DEV.4.M315: (Hardware ONLY) Prevent configuration data readback
This mitigation is required to counter exploitation of debug/testing features
At Foundation Grade the product is required to prevent configurable hardware from allowing configuration data readback.
NOTE: This only applies to FPGA implementations.

DEV.4.1 - Design >> Hardware >> Network Interface
DEV.4.1.M109: (Hardware ONLY) Protection of sensitive data lines
This mitigation is required to counter installation of hardware-level malware
At Foundation Grade the product is required to ensure physical access to internal data lines carrying sensitive data requires breaching of the tamper protection.
In this context, sensitive data is defined as key material, user data and configuration data.

DEV.5 - Design >> Device Configuration
DEV.5.M267: Provide an automated configuration tool to enforce required settings
This mitigation is required to counter exploitation of an accidental misconfiguration
At Foundation Grade the product is required to be provided with a configuration tool, or other method, for an administrator to initially set it up into a suitable configuration.
If the product requires more than 12 options to be changed or set by an administrator to comply with these Security Characteristics, the developer must supply a tool or policy template which helps the administrator to achieve this in fewer steps.
DEV.5.M353: Ensure product security configuration can only be altered by an authenticated system administrator

This mitigation is required to counter unauthorised alteration of product's configuration

At Foundation Grade the product is required to ensure that only administrators are able to change the product's security enforcing settings. The only security enforcing setting a user should be able to change is their passphrase.

DEV.6 - Design >> Authentication

DEV.6.M612: Sanitise logged data

This mitigation is required to counter supplying a malicious script through logged data

At Foundation Grade the product is required to ensure logged data is appropriately sanitised prior to display. The method and content of sanitisation will change depending on the content in the logs and where the logs are displayed. For example, output to a HTML viewer for the logs will need to be encoded whereas logging output to a text file may not need to be sanitised.

DEV.6.M615: Inform administrator of account activity

This mitigation is required to counter exploitation of poor management of passphrases by the administrator

This mitigation is required to counter dictionary and exhaustion attacks

At Foundation Grade the product should display recent authentication history. It is recommended that on login the user be notified of the date and time of the last successful login and any failed login attempts since the last successful login.

If recent authentication history is displayed, it is strongly recommended that users are told what to do, preferably on the screen, if the history is not what is expected.

DEV.6.M616: Anti Hammer

This mitigation is required to counter dictionary and exhaustion attacks

At Foundation Grade the product is required to have a mechanism for limiting the rate of login attempts.
B. Verification Mitigations

VER.M80: Protocol robustness testing
This mitigation is required to counter discovery of a vulnerability in the implementation of the protocol
At Foundation Grade the evaluator will perform testing using commercial fuzzing tools.
Fuzz testing is described in more detail in the Process for Performing Foundation Grade Evaluations.

VER.M347: Verify update mechanism
This mitigation is required to counter installing compromised software using the update process
At Foundation Grade the evaluator will validate the developer's assertions regarding the suitability and security of their update process.
The update process must provide a mechanism by which updates can be authenticated before they are applied.
The process and any configuration required must be documented within the Security Procedures.

VER.M638: Ensure data is not lost on power loss
This mitigation is required to counter exploitation of incorrect operations of device
At Foundation Grade the evaluator will ensure that operationally important data is not lost from power loss.
This includes important data such as:
Firewall configuration
Logging / auditing data
Firewall rulesets.

VER.1.- Verify >> Hardware >> Network Interface

VER.1.M80: (Hardware ONLY) Protocol robustness testing
This mitigation is required to counter discovery of a vulnerability in the implementation of the protocol
At Foundation Grade the evaluator will perform testing using commercial fuzzing tools.
Fuzz testing is described in more detail in the Process for Performing Foundation Grade Evaluations.

VER.2 - Verify >> Rules Configuration

VER.2.M630: Product allows Deny-All default
This mitigation is required to counter exploitation of omission/error in rule configuration
At Foundation Grade the evaluator will ensure that the product provides the ability to enforce "Default Deny" on rulesets.
### VER.3 - Verify >> HTTP Parser

<table>
<thead>
<tr>
<th>Mitigation ID</th>
<th>Description</th>
</tr>
</thead>
</table>
| VER.3.M640 | **Verify correct operation of ruleset**  
This mitigation is required to counter exploitation of incorrect operations of device  
At Foundation Grade the evaluator will perform testing to ensure traffic is processed correctly for a given ruleset. Traffic which does not conform should be dropped.  
Verification can be achieved by creating a sample ruleset and then testing against that ruleset. The device will conform to this mitigation if the expected outcomes are observed during testing. |
| VER.3.M641 | **Ensure device denies traffic on start up**  
This mitigation is required to counter exploitation of incorrect operations of device  
At Foundation Grade the evaluator will ensure that no traffic traverses the device until the device is fully operational.  
The device may take time to load rules and configuration data at start up, therefore the firewall shouldn't process traffic until these are loaded.  
This also applies to re-starting/rebooting of device. |
| VER.3.M643 | **Requests are filtered based on the contents of HTTP parameters**  
This mitigation is required to counter a Cross Site Scripting attack  
At Foundation Grade the evaluator will ensure that HTTP parameters not conforming with the ruleset are dropped. |
| VER.3.M645 | **HTTP input sanitisation**  
This mitigation is required to counter supplying malicious parameters  
At Foundation Grade the evaluator will ensure that all HTTP traffic is sanitised.  
The evaluator must ensure that all HTTP input to the device is sanitised such that any malicious injected content is sanitised before being stored by the underlying web application. |
| VER.3.M648 | **Prevent data exfiltration**  
This mitigation is required to counter accessing a prohibited URL  
This mitigation is required to counter supplying a malformed HTTP request  
This mitigation is required to counter supplying malformed headers  
This mitigation is required to counter supplying malicious parameters  
This mitigation is required to counter use of an unexpected HTTP method  
At Foundation Grade the evaluator will ensure that the Web Application Firewall drops any unexpected served content. |
| VER.3.M650 | **Requests are filtered based on HTTP method**  
This mitigation is required to counter use of an unexpected HTTP method  
At Foundation Grade the evaluator will ensure that HTTP methods not conforming to the ruleset are dropped. |
| VER.3.M654 | **Requests are filtered based on the HTTP header referrer field**  
This mitigation is required to counter a Cross Site Request Forgery attack and other forced browsing attacks  
At Foundation Grade the evaluator will ensure that requests with invalid HTTP header referrer fields are dropped. |
| VER.3.M655 | **Requests are filtered based on URL**  
This mitigation is required to counter accessing a prohibited URL  
At Foundation Grade the evaluator will perform testing to ensure that requests are filtered based on URLs.  
Requests not conforming to the ruleset are dropped, and those conforming are passed on for further processed. |
VER.3.M657: Filtering HTTP response headers

This mitigation is required to counter fingerprinting of HTTP headers. At Foundation Grade the evaluator will ensure that HTTP response headers not conforming to the ruleset are dropped. HTTP response headers must be inspected for any unauthorised/sensitive data relating to the configuration of the web application or server.
C. Deployment Mitigations

DEP.M39: Audit log review
This mitigation is required to counter exploitation of a software logic error
This mitigation is required to counter exploitation of a software implementation error
At Foundation Grade the deployment is required to regularly review audit logs for unexpected entries.

DEP.M46: User least privilege
This mitigation is required to counter taking advantage of existing user privilege
At Foundation Grade the deployment is required to ensure all user accounts have the fewest privileges required to enable business functionality.

DEP.M131: Operating system verifies signatures
This mitigation is required to counter installation of a malicious privileged local service
At Foundation Grade the deployment is required to enable signature verification for applications, services and drivers in the host operating system, where supported and where the product makes use of it.

DEP.M159: Update product
This mitigation is required to counter exploitation of a software logic error
This mitigation is required to counter exploitation of a software implementation error
At Foundation Grade the deployment is required to update to the latest version where possible.

DEP.M340: Address Space Layout Randomisation
This mitigation is required to counter exploitation of a software implementation error
At Foundation Grade the deployment is required to enable ASLR in the host Operating System where available.

DEP.M348: Administrator authorised updates
This mitigation is required to counter installing compromised software using the update process
At Foundation Grade the deployment is required to confirm the source of updates before they are applied to the system.
The administrator is required to have authorised the updates before use. If an automatic process is used, the administrator must also configure the product to authenticate updates.
The administrator is required to use the update process described within the Security Procedures.

DEP.M606: Control access to device management
This mitigation is required to counter attacking management protocol
At Foundation Grade the deployment is required to restrict which network interfaces can be used for device management.
If a local console port or dedicated management interface is available, it must be possible to configure the other network interfaces to not have management services accessible on them.
Similarly, it must also be possible to restrict which network interfaces have management services enabled on them.
DEP.M620: Physical security controls

This mitigation is required to counter physically destroying the device
At Foundation Grade the deployment is required to store the device in an appropriately secured area.
This applies to both operational and non-operational storage.

DEP.M623: Take action on receiving alerts

This mitigation is required to counter high valid traffic volumes
This mitigation is required to counter use of malformed/unusual traffic
At Foundation Grade the deployment is required to assess impact of alerts and follow organisational procedures for incident resolution.

DEP.1 - Deploy >> Rules Configuration

DEP.1.M608: Configuration export procedure

This mitigation is required to counter loss of configuration data due to modification
At Foundation Grade the deployment should export the product configuration whenever changes are made.

DEP.1.M610: Configuration auditing

This mitigation is required to counter loss of configuration data due to modification
This mitigation is required to counter modification of rules without valid administrator credentials
At Foundation Grade the deployment is required to audit the product configuration.
Auditing will highlight any changes made to the configuration. These changes should then be analysed to ensure they were made by a valid administrator and that they do not introduce any vulnerabilities.

DEP.1.M628: Administrators Guidance on ruleset management

This mitigation is required to counter exploitation of omission/error in rule configuration
At Foundation Grade the deployment is required to ensure administrators are educated on how to configure the ruleset.

DEP.1.M629: Adopt Deny-All default rule set

This mitigation is required to counter exploitation of omission/error in rule configuration
At Foundation Grade the deployment is required to automatically enforce "Default Deny" rule at the end of every ruleset.

DEP.2 - Deploy >> Logging

DEP.2.M625: Log all relevant actions

This mitigation is required to counter modification of logging generation
At Foundation Grade the deployment is required to automatically export logs to management/red side device.
At Foundation Grade the deployment is required to configure the product to log capture all actions deemed of interest.
Ensure that log data is detailed enough to allow forensic investigation during any incident management.
Sensitive data such as passwords and keys must not be written to the logs.
DEP.2.M626: Monitor logs for unexpected entries

This mitigation is required to counter modification of logging generation
This mitigation is required to counter sanitisation of illegitimate access from logs
At Foundation Grade the deployment is required to assess impact of entries and follow organisational procedures for incident resolution.

DEP.3 - Deploy >> HTTP Parser

DEP.3.M39: Audit log review
This mitigation is required to counter exploitation of a software logic error
This mitigation is required to counter exploitation of a software implementation error
At Foundation Grade the deployment is required to regularly review audit logs for unexpected entries.

DEP.3.M159: Update product
This mitigation is required to counter exploitation of a software logic error
This mitigation is required to counter exploitation of a software implementation error
At Foundation Grade the deployment is required to update to the latest version where possible.

DEP.3.M340: Address Space Layout Randomisation
This mitigation is required to counter exploitation of a software implementation error
At Foundation Grade the deployment is required to enable ASLR in the host Operating System where available.

DEP.4 - Deploy >> Device Configuration

DEP.4.M38: Use automated configuration tool
This mitigation is required to counter exploitation of an accidental misconfiguration
At Foundation Grade the deployment is required to be configured using automated tools if provided.

DEP.5 - Deploy >> Hardware

DEP.5.M26: (Hardware ONLY) Physical tamper evidence
This mitigation is required to counter physical compromise of device
This mitigation is required to counter installation of hardware-level malware
At Foundation Grade the deployment is required to educate users to regularly check that tamper labels are intact.
At Foundation Grade the deployment is required to provide administrators with advice on the tamper threat.
Advice should include looking for possible damage to tamper evident seals.

In the event of tampering, the event should be reported as soon as possible and the product must be removed from use immediately. Any product that shows evidence of tampering must not be returned to service.
At Foundation Grade the deployment is required to place tamper evident seals over access points on product.
Use tamper evidence (e.g. stickers) to make entry to system internals detectable by physical inspection. Tamper stickers should be uniquely identifiable to prevent an attacker successfully replacing it with a new, undamaged sticker.
UNCLASSIFIED

DEP.5.M620: (Hardware ONLY) Physical security controls

This mitigation is required to counter compromising physical security surrounding device

At Foundation Grade the deployment is required to store the device in an appropriately secured area.

This applies to both operational and non-operational storage.

DEP.5.1 - Deploy >> Hardware >> Network Interface

DEP.5.1.M26: (Hardware ONLY) Physical tamper evidence

This mitigation is required to counter physical compromise of device
This mitigation is required to counter installation of hardware-level malware

At Foundation Grade the deployment is required to educate users to regularly check that tamper labels are intact.

At Foundation Grade the deployment is required to provide administrators with advice on the tamper threat.

Advice should include looking for possible damage to tamper evident seals.

In the event of tampering, the event should be reported as soon as possible and the product must be removed from use immediately. Any product that shows evidence of tampering must not be returned to service.

At Foundation Grade the deployment is required to place tamper evident seals over access points on product.

Use tamper evidence (e.g. stickers) to make entry to system internals detectable by physical inspection. Tamper stickers should be uniquely identifiable to prevent an attacker successfully replacing it with a new, undamaged sticker.

DEP.6 - Deploy >> Authentication

DEP.6.M282: Initial passphrase is changed on first use

This mitigation is required to counter use of system default passphrases

At Foundation Grade the deployment is required to ensure passphrase is changed on first logon.

The system must force users to use an initial passphrase once only, i.e. forces the passphrase to change on first logon.

It is strongly recommended that initial passphrases have a limited lifetime between generation and first use that is as short as is practicable.
DEP.6.M613: Provide guidance on passphrase management

This mitigation is required to counter a social engineering attack on the administrator.
This mitigation is required to counter exploitation of poor management of passphrases by the administrator.
This mitigation is required to counter dictionary and exhaustion attacks.
At Foundation Grade the deployment is required to provide training to administrators on passphrase management.

Administrators should be provided with guidance regarding the secure handling of passphrases which allow access to sensitive systems.
Administrators must be taught never to disclose passphrases, even to their superiors.
Administrators must also be made aware of the risks of using protectively marked devices in public or untrusted areas. Passphrases should not be entered in areas where others could see them being entered.
An administrator must not use passphrases in more than one system.
At Foundation Grade the deployment is required to ensure any hardcopies of passphrases are stored securely.

At Foundation Grade the deployment should educate administrators about social engineering methods used by attackers.

DEP.6.M614: Suitable passphrase length and complexity

This mitigation is required to counter exploitation of poor management of passphrases by the administrator.
This mitigation is required to counter dictionary and exhaustion attacks.
At Foundation Grade the deployment is required to ensure passwords are at least 8 characters long.
User generated passphrases are acceptable, but machine generated passphrases should be used where possible.
## IV. GLOSSARY

24. The following definitions are used in this document:

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL</td>
<td>Access Control List</td>
</tr>
<tr>
<td>ASLR</td>
<td>Address Space Layout Randomisation</td>
</tr>
<tr>
<td>CPA</td>
<td>Commercial Product Assurance</td>
</tr>
<tr>
<td>Crash</td>
<td>Unexpected event which causes the device to not function as intended</td>
</tr>
<tr>
<td>DOM</td>
<td>Document Object Model</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Array</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Transport Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transport Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hypertext Transport Protocol Secure</td>
</tr>
<tr>
<td>IDS</td>
<td>Intrusion Detection System</td>
</tr>
<tr>
<td>IPS</td>
<td>Intrusion Prevention System</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>JTAG</td>
<td>Joint Test Action Group</td>
</tr>
<tr>
<td>MIME</td>
<td>Multipurpose Internet Mail Extensions</td>
</tr>
<tr>
<td>OWASP</td>
<td>Open Web Application Security Project</td>
</tr>
<tr>
<td>Ruleset</td>
<td>The set of rules that determine how a packet is handled by the firewall</td>
</tr>
<tr>
<td>Security Characteristic</td>
<td>A standard which describes necessary mitigations which must be present in a completed product, its evaluation or usage, particular to a type of security product</td>
</tr>
<tr>
<td>SHA</td>
<td>Secure Hash Algorithm</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>WAF</td>
<td>Web Application Firewall</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>