19234079

CPA SECURITY CHARACTERISTIC

SOFTWARE EXECUTION CONTROL

Version 1.0

Document History

Version	Date	Description
0.1	March 2012	Initial Draft
0.2	May 2012	Pre-release Draft
1.0	August 2012	Final Draft

This Security Characteristic is derived from the following files

File Name	Version
Software Execution Control	1.0
Policy Enforcement	1.0
Common Libraries	1.9

Soft copy location

DiscoverID 19234079

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.

CONTENTS

REFERENCES

[a] The Process for Performing Foundation Grade CPA Evaluations

[b] CESG Architectural Pattern - Mobile Remote End Point Devices, v1.0, May 2012, CESG

[c] Security Characteristic – IPsec VPN For Remote Working

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. Software Execution Control products are used to limit which software applications and services are able to run on an Operating System. They are used to control the attack surface of a platform by reducing the number of potential vulnerabilities exposed by a system, and also help to mitigate the impact of successful social engineering attacks.

3. These products are used in conjunction with an administrator-defined set of rules which define which software a non-privileged user is able to execute. These rules can be based on properties of the software (such as name, signature, etc) or be more generic (such as permitting execution of files based on their storage location on the system). The effectiveness of the security provided by these products is thus highly dependent on the rules that are designed by the system administrator.

4. This Security Characteristic does not address any threats against software that the Software Execution Control product is configured to permit executing. For example, if a document-reading application is permitted to execute, and a malicious document is opened with it, an attacker will still be able to gain access to the system. However, the attacker's ability to remain on the system may be constrained by the product's presence.

B. Typical Use Case(s)

5. Software Execution Control forms a key component of a well-configured endpoint by providing system administrators the ability to configure and constrain the software that is running on such systems. These products are therefore used to support endpoints deployed in accordance with CESG guidance such as CESG Architectural Pattern - Mobile Remote Endpoint Devices [b] Chapters 6 & 9.

6. Many compromises of endpoints occur because the user is able to execute software that is either itself malicious or which is not corporately supported (and protected) – and may thus be open to attack. Software Execution Control products provide administrators the ability to limit the applications and services which are allowed to run on a particular platform, thus providing improved platform manageability and a reduced attack surface.

C. Expected Operating Environment

7. Software Execution Control products (the ‘client’) can be implemented on many types of endpoint - ranging from a standard desktop computer within an office environment through to a mobile device.

8. The client will usually require a connection to an enterprise network in order for it to receive policy updates which will have been created by a system administrator. The server within the enterprise network which serves the policy is not covered by the scope of this Security Characteristic.

9. This Security Characteristic is applicable to products operating on all modern Operating Systems.

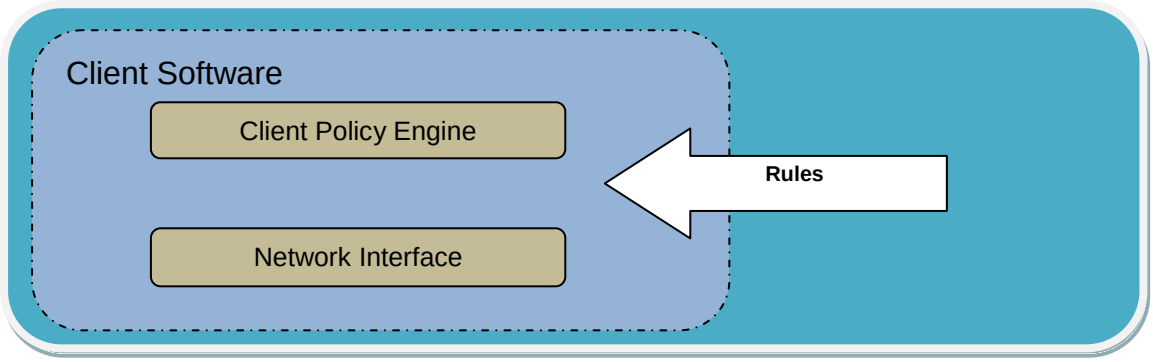
D. Interoperability

10. There are currently no defined standards for configuring Software Execution Control products, and so it not expected that the management application of one product would interoperate with the client software of another.

E. Additional Threat Information

11. This Security Characteristic does not provide mitigation against any physical threats to an endpoint such as theft or tampering; protection for these threats would need to be provided by alternative products or procedural mitigations.

F. High Level Functional Components



12. The functionality of the software can be broken down into two main components:

13. **Rules**
Rules created by an administrator are received by the client software. They are usually deployed over a network using a management server but could also be manually loaded onto individual machines by an administrator.

14. **Client Policy Engine**
This interprets the rules received from the management software and applies them to the endpoint to control the software which is allowed to execute.

15. **Network Interface**
If present, the network interface aspect allows the Client Software to communicate with the management software to receive administrator-defined rules for the product.

G. Future Enhancements

16. CESG welcomes feedback and suggestions on possible enhancements to this Security Characteristic.

II. SECURITY CHARACTERISTIC FORMAT

17. All CPA Security Characteristics contain a list of mitigations which are split into three categories: development, verification and deployment. 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.

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

- **DEV** – Development mitigations 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 items that the evaluator must test during the evaluation 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 which are published by CESG for the product.

19. Each mitigation includes:

- Informational text in italics, describing the threat to be mitigated.
- One or more specific mitigations, which describe what must be done.
- Optional additional explanatory text which expands upon the requirement.

20. In the mitigations listed below, the following terminology is 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.

21. 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 **should** 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 including 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.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.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.M321: Data Execution Prevention

This mitigation is required to counter exploitation of a software implementation error

At Foundation Grade the product **is required to** support Data Execution Prevention (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.

If the product is to be specifically deployed on an operating system that does not support ASLR, there is no requirement for ASLR compatibility.
Note: ASLR may be disabled for specific aspects of the product, provided there is justification of why this is required.

DEV.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 authenticated administrators are able to change the product's security enforcing settings.

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.

DEV.1 - Design >> Client Policy Engine

DEV.1.M741: Control of background execution

This mitigation is required to counter exploitation of background execution

At Foundation Grade the product **is required to** additionally control background execution.

Some user-accessible or network-facing software may be running as a service or daemon. It must be possible to control which of these applications are able to run.

DEV.1.M780: Ensure configuration updates are applied

This mitigation is required to counter prevention of configuration updates

At Foundation Grade the product **is required to** protect against configuration updates being blocked by a malicious user.

Where configuration updates are received from a server the update process must be implemented on the client without requiring user interaction (e.g. a user must not be able to disapprove a received configuration update).

DEV.1.M784: Protect enforcement code from modification

This mitigation is required to counter unauthorised modification of enforcement code

At Foundation Grade the product **is required to** implement enforcement code such it that cannot be modified or disabled by non-privileged users.

DEV.1.M786: Secure ruleset configuration delivery

This mitigation is required to counter spoofing of configuration updates

At Foundation Grade the product **is required to** ensure that configuration updates are authorised prior to application.

Authorisation may be established either locally by an administrator or remotely by a cryptographically protected mechanism.

DEV.1.M789: Log execution attempts

This mitigation is required to counter undetected execution control errors

At Foundation Grade the product **is required to** log all successful and unsuccessful attempts by all non-privileged users to execute code.

Where logs are created they must be passed on to a central location (which is available to the administrator), in a timely manner.

DEV.1.M790: Control of user-mode applications

This mitigation is required to counter lack of execution control coverage

At Foundation Grade the product **is required to** allow the administrator to define a ruleset to control which executable files, or groups of files, can be run by which non-privileged users.

A ruleset can be defined such that the administrator is able to control which executable files, or groups of files, can be executed.

At Foundation Grade the product **is required to** apply execution control to all user-mode executable types.

DEV.1.M791: Apply execution rules to all non-privileged users

This mitigation is required to counter exploitation of "default permit" behaviour

At Foundation Grade the product **is required to** apply the ruleset to all non-privileged users by default.

Where the administrator has not specified a ruleset for a specific non-privileged user, the product must ensure that a default ruleset is applied to that user.

The product must also ensure that the administrator can view which rulesets apply to which users.

DEV.1.M792: Deny user-mode code execution by default

This mitigation is required to counter exploitation of "default permit" behaviour

At Foundation Grade the product **should** only allow the execution of code specified in an allow-list.

To prevent simple spoofing of this rule the product must ensure that the specific file is unambiguously identified. This may be by its absolute path or by matching a digital signature.

B. Verification Mitigations

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.1 - Verify >> Network Interface

VER.1.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.2 - Verify >> Client Policy Engine

VER.2.M781: Verify enforcement

This mitigation is required to counter exploitation of ineffective enforcement

At Foundation Grade the evaluator **will** ensure that the configured ruleset is enforced and validate that all configuration options available to the administrator are applied as expected.

A test plan should be created and executed for each feature. Exhaustive testing of configuration combinations is not required.

VER.2.M785: Non-privileged user cannot disable enforcement

This mitigation is required to counter unauthorised modification of enforcement code

At Foundation Grade the evaluator **will** confirm users cannot bypass operating system controls.

The user must not be able to disable services, kill processes (other than those they have started), modify applications on the device or deactivate the product by rebooting the host device.

C. Deployment Mitigations

DEP.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.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.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 update procedure to be used by the administrator must be described within the product's security procedures.

DEP.1 - Deploy >> Client Policy Engine

DEP.1.M738: Third party application rules

This mitigation is required to counter gaining execution via permitted application

At Foundation Grade the deployment **is required to** not deploy uncontrollable applications.

All software installed on the client must be directly controllable by the product. Lockdown coverage must include execution control of code through any (non-locked-down) scripting language interpreter or runtime shell.

DEP.1.M787: Effective rule configuration

This mitigation is required to counter exploitation of an inadequate ruleset

At Foundation Grade the deployment **is required to** generate a ruleset tailored to the specific requirements of the host environment.

The ruleset needs to be carefully considered to ensure that users are not able to circumvent the intended controls. For example it is not appropriate to white-list a path of an executable in a location that is user-writable as this would allow an approved executable to be overwritten with a malicious one. The ruleset should reflect current CESG Good Practice Guides.

At Foundation Grade the deployment **is required to** ensure the ruleset is reviewed regularly.

A network constantly evolves over its lifetime. The ruleset needs to be reviewed at least quarterly to ensure that it remains appropriate for the network configuration.

IV. GLOSSARY

22. The following definitions are used in this document:

Term	Meaning
Administrators	Users that have privilege to alter policy or interact with privileged operating system functions such as loading new drivers or updating policy.
ASLR	Address Space Layout Randomisation
CPA	Commercial Product Assurance
DEP	Data Execution Prevention
Non-privileged user	Users that have specific role-based privileges such as password changes.
Security Characteristic	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.

THIS PAGE IS INTENTIONALLY LEFT BLANK