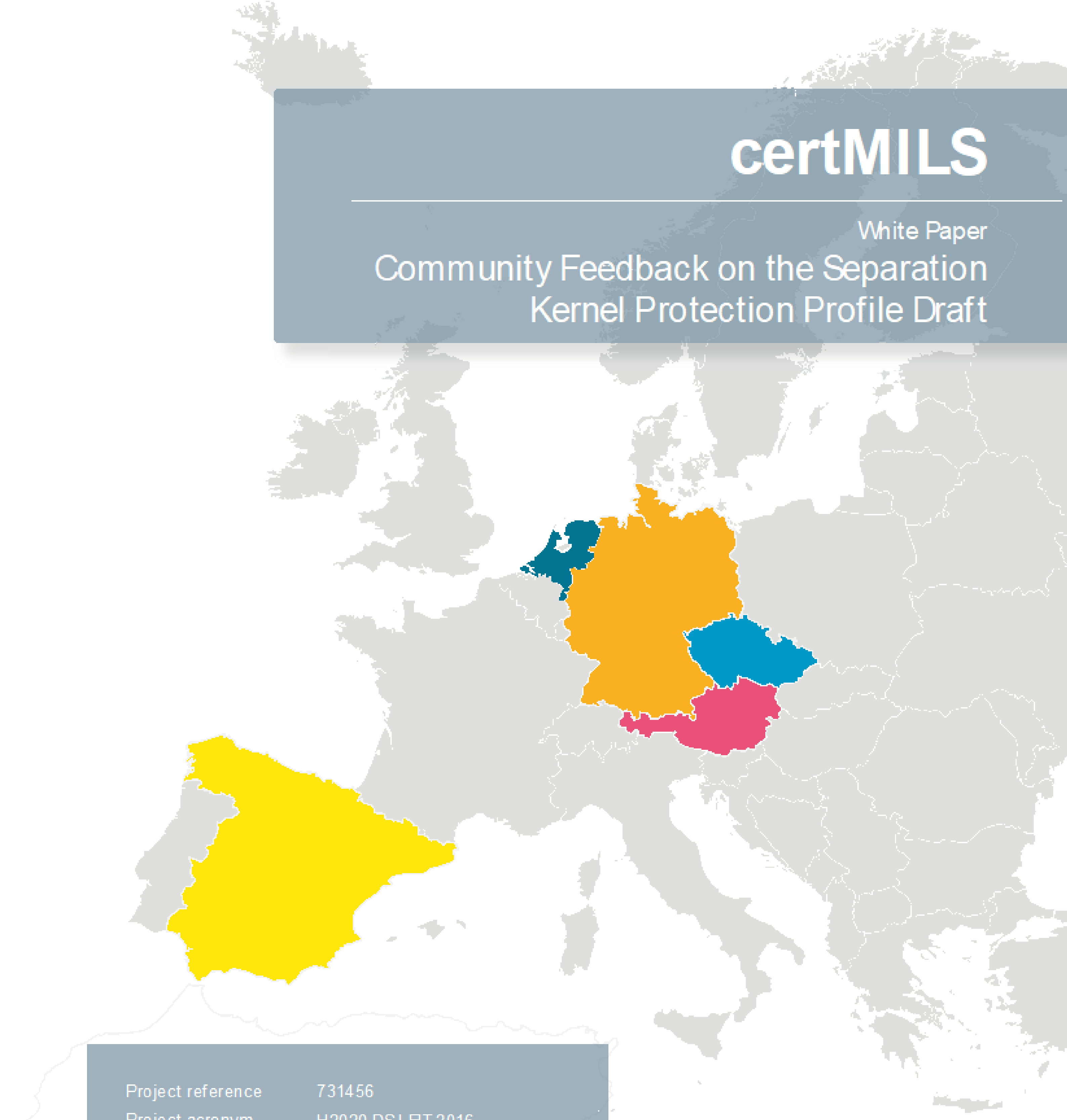
certMILS

White Paper

Community Feedback on the Separation Kernel Protection Profile Draft



Project reference	731456
Project acronym	H2020-DS-LEIT-2016
Project title	CERT-MILS Compositional security certification for medium- to high-assurance COTS-based systems in environments with emerging threats
Start date:	2017-01-01
Duration	48 months
Programme type	Horizon 2020
Project website	https://certmils.eu/

ezú elektrotechnický zkušební ústav

EPOCHE & ESPRI

SYSGO EMBEDDING INNOVATIONS

THALES

@SEC the information security provider

TECHNIKON

UniControls Transport and Industrial Control Systems

NXP

Schneider Electric

Universität Rostock Traditio et Innovatio

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 731456.

Editor

Thorsten Schulz (University of Rostock)

Contributors

Andreas Hohenegger (atsec information security GmbH)

Álvaro Ortega (Epoche & Espri)

Holger Blasum (Sysgo AG)

Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author’s view – the European Commission is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

Executive Summary

This white paper is reporting on interoperability aspects of the Common Criteria Base Separation Kernel Protection Profile (PP) draft [1]. This white paper captures the results of the collaboration on PP interoperability organised by University of Rostock in Task 9.2. It reports how the PP draft can be applied to the separation kernels of MILS platform providers and how well the PP draft addresses requirements of users such as system integrators. Previously, the WP 2 has created a PP with additional modules [2]. To make the proposed PP most accessible to all potential stakeholders in the MILS domain and the Separation Kernel application domain, WP 9 proposed to gather feedback from the community for integration into the PP draft.

The activities being discussed resemble mostly the Common Criteria User Forum presentations and its community involvement. Beyond that, certification bodies and a few known consortium contacts were directly contacted and invited to provide feedback. The questions asked, as well as the accumulated answers are presented.

The white paper closes with a discussion on the continued improvement of the PP for proposed acceptance and adoption.

Contents

Chapter 1 Introduction

The past has seen multiple efforts to establish a protection profile for separation kernel applications and related systems. These approaches were discussed in previous certMILS deliverables. To avoid producing “Yet-Another-Separation-Kernel-PP”, we have on the one hand studied the issues of previous approaches, and on the other hand, started to establish a broad support community as early as possible.

The targeted community consists of hardware providers, Separation Kernel developers (in this document we use “separation kernel developers” as synonym to “MILS platform provider” because the term “separation kernel” is established in the Common Criteria), potential system integrators, evaluation laboratories, and certification bodies. Part of task 9.3 builds and maintains the MILS community with a mailing list, website and annual workshops. However, this community is only partially involved with ISO 15408 / CC, but also with IEC 62443 related certification schemes.

For community involvement, Common Criteria has its own platform, the Common Criteria Users Forum. The CCUF community includes all of above groups in an open collaboration group with regular workshops, meetings and an online team platform.

Beyond establishing the Separation Kernel technical committee (SK-TC), which will be detailed in the following chapter, we also initiated joint work with the closely related Hypervisor TC. The CCUF workshops were also a chance to contact experts outside of these workgroups.

We published the certMILS base PP, and three CPU timing modules to the CCUF community. For non CCUF-community members, a copy was made available at the certMILS site, for feedback collection. The accompanying guidance documents were not pushed for the initial community feedback action, not to overload interested parties and decrease feedback participation. This approach was successful, as most feedback participants seemed to have studied the provided documents thoroughly and provided extensive, as well as positive feedback, that will be incorporated in both PP and guidance documents.

Chapter 2 CCUF communication activities

2.1 Initial activities

The Common Criteria Users Forum is the platform for communications between the CC community, the organizational committees, policy makers and evaluation schemes. A specific working group was started in spring 2018 representing the dominant MILS technology of a separation kernel with the Separation Kernel Technical Community (SK-TC). At the introduction in the CCUF meeting in Trondheim the Essential Security Requirements (ESR) were discussed. The first input widened the requirements for applicability of the targeted new SK-PP also for simpler systems, e.g. by describing alternatives to virtualization technologies and inclusion of a choice of simple CPU-time modules.

2.2 Targeted communication activities

The CCUF is an open platform and welcoming wide participation from users to CC experts. The platform will serve for further collaboration and community communication around the SK PP draft, after the questionnaire initiative has closed.

Further discussion will be needed, first internally in the project, then at the CCUF how the feedback will be included best in the PP draft. It is also up for discussion, how further efforts will be aligned with activities from the co-working Hypervisor TC. There exist different long-term directions, but also many similar aspects.

This section describes the security mechanisms available in the system M, which are provided by applications.

Chapter 3 Methodology for the community questionnaire

The previous chapter discussed the initial presentation of the ESR at the CCUF workshop in Trondheim. This was an interactive “round-table-style” session with free discussions on the slides and terms about the SK-ESR. While this approach gave good, early indications, where the initial document had weaknesses, but it came clear that further feedback requires a more structured process.

The plan was to generate a common set of questions addressing aspects that came up at the ESR session, issues that triggered discussions within the consortium, general applicability questions, etc. This questionnaire would be introduced on the CCUF workshop in Amsterdam. We planned guided interviews at the venue, call for participation through the different platforms, as well as directly inviting known industry contacts from consortium partners. We asked for comments on the questionnaire questions itself.

3.1 Questions

At first, general questions were collected within the project consortium. These addressed also the awareness of the MILS community and the certMILS project. The internal review deemed these out of scope, as the target is the CC Separation Kernel PP draft. The follow up revision was mainly written by the PP draft authors from WP2, having a better insight on critical issues needing feedback from the community.

The questions were improved towards

- proper CC language,
- longer explanation to guide non-CC-savvy participants,
- not expecting participants have read the CC,
- less presence of the project, as this is undesired practice in the CCUF realm.

The initial expectation to gather feedback based on the questionnaire within the workshop in Amsterdam was further lowered with this extended version.

The questionnaire was compiled for the Evasys evaluation system, as provided for research evaluation by the University of Rostock. It is secured by the University’s infrastructure. The system provides paper-based and electronic forms. Paper copies were brought to the CCUF workshop but none was completed.

The final questionnaire is appended to this document in its online-presentation.

3.2 Addressees

The PP draft documents were first published on the team collaboration site of the CCUF before the introductory workshop. The questionnaire was also published with a request for comments whether it is sufficient. The particular audience statistics of the TC mailing list is detailed in D9.3.

As a next step, the draft PP was introduced and discussed in a session at the CCUF Amsterdam workshop (October 25th 2018). The audience was populated by ~40 people coming from all groups from certification bodies, SW developers, evaluation laboratories, to research.

Our community feedback methodology was also introduced with a couple of slides, trying to engage listeners for feedback. However, audience members approached to complete the survey-form at the workshop excused themselves to first needing to read the PP drafts before commenting.

Following the workshop, in November, the questionnaire was also published for feedback in the MILS community mailing list, to (about 40) industrial contacts, three mailing lists on microkernels (seL4, Fiasco, XtratuM), the OpenGroup MILS API mailing list and European certification bodies. Feedback has been collected until December 8th 2018.

There was no technical restriction applied, refraining from multiple submissions. One participant sent two submissions with minor differences. For that repeated submission the data has been aggregated (i.e., all non-duplicate textual feedback is taken into account; numerical feedback is only counted once).

For confidentiality, there was a choice of three options:

- Upload to the CCUF Separation Kernel WG/TC with attribution for discussion [5x],
- Accumulate feedback to collected answers - attribution in acknowledgement, not linked [3x],
- Accumulate feedback to collected answers - without any attribution [3x].

3.3 Analysis

We have split roles of data collector and analyst. That is, the person, who collected the questions (data collector), was different from the person doing the analysis (analyst). I.e., in particular, the analysis was done without looking up any data, where people wanted to be anonymous respondents. In some cases, the analyst asked the data collector to aggregate / correlate data (e.g., Question 1.7 in Chapter 4).

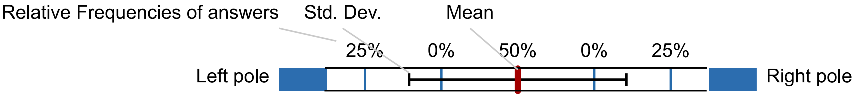
Some answers have been split up into multiple bullet points, if (1) they were covering different subjects and (2) this seemed advised for de-personalization. Some wordings have been smoothed / spellings have been corrected for de-personalization.

We are providing a preliminary analysis sections. The term “Preliminary analysis” has been chosen to indicate that we are open to further community feedback (e.g., from CCUF, beyond the scope of this certMILS white paper). The numbering of the answers *A1…An* was introduced ex post, to be able to refer to them more easily in the “Preliminary analysis” sections.

Chapter 4 Results report

The survey was completed by 12 participants. Most participants answered the majority of questions.

Legend to charts:



Legend to text answers:

> Text in *italics* is cited answer text. Answers have been grouped by subject (the questionnaire output had been sorted alphabetically). Some answers have been split up into multiple bullet points for better understandability and corrected for typos. See note in previous chapter.

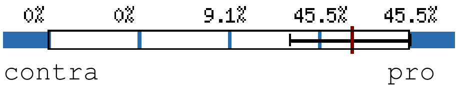
4.1 Section 1: General

1.1) Please indicate whether you are representing a separation kernel developer, integrator, CC evaluation facility, CC certification body, etc.

[11 answers] The most common self-identification was "*separation kernel developer*" (4 counts), semiconductor/HW platform/SoC provider (2 counts), followed by (one mention each) "*CC certifier*", "*certification body*", "*security researcher*", "*someone who has worked extensively on separation kernels, ISO 15408, etc.*", "*possible stakeholder (interested on separation kernel products, interested by CC certifications)*".

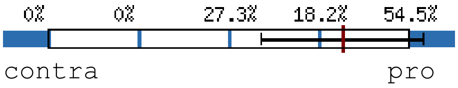
1.2) Do you think that there is a need for a separation kernel PP?

[11] Participants generally agreed with the need for a Separation Kernel PP.



1.3) Do you agree with the strategy to produce a modular separation kernel PP and its proposed scope?

[11] The modular approach of a PP was also embraced but with a slightly wider deviation.



Note: We posted a clarification on 23 November 2018 to the CCUF SK TC/WG online forum that the use of modules in this question "*does not necessarily prescribe that only CC v3.1.5 Part "modules" are used for every possible sort of modularity, we also have become aware of "selection-based SFRs" (such as e.g. in DSC [Dedicated Security Components] cPP) and "SFR packages" (such as in OSPP).*"

1.4) Do you think that the title "Separation Kernel Protection Profile" is appropriate? If not, which name(s) would you prefer?

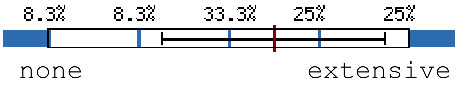
[7] 6 of them considered the name appropriate, one of these pointing that it should clearly distinguish from the sunset SKPP. One answer expressed that the respondent was not sure, and that "*the PP TOE description should generalize and provide good examples of instantiations, including microkernels and Type-1 hypervisors*".

Preliminary analysis

Topics mentioned were in relation to the sunset SKPP, and to Type-1/Type-2 hypervisors

1.5) Do you already have experience with the CC certification of a separation kernel?

[12] Previous experience with CC certification of a separation kernel was spread all across none to extensive.



1.6) For which countries do you need/plan a certificate and/or which countries' authorities do you plan to work with?

[5] France/ANSSI (3x), Germany/BSI (3x), USA (2x), Netherlands/NCSA (1x), EU (1x), NATO (1x), UK (1x), Australia (1x)

1.7 What Evaluation Assurance Level (EAL) would you need a certificate for? [none..multiple]

[8] *3; 3-4; 4; 4-6* (2x)*;* *5* and *7; 6-7; 7.*

Preliminary analysis

The common ground would be starting from EAL3, *if* an EAL level is used at all (compare Answer A5 to Question 3.10). Some participants indicated quite ambitious EAL levels.

1.8) Is there anything relevant that you would like the separation kernel WG to be aware of?

[6] some answers were split:

Overall feedback:

A1: We got the feedback from one participant that some of the questions (such as the usefulness etc.) were considered "*very leading*", or asked for information that is confidential and/or out of the scope of a PP review.
A2: *Not at the moment, I am still really new to this but I would like to stay in touch with the working group.*
A3: *You should get feedback from leading separation kernel companies such as Green Hills.*

Mapping to CPU privilege levels/relation to hypervisors:

A4: *The separation kernel PP should explicitly address both micro-kernels and hypervisors (e.g. Type-1 micro-hypervisors). The difference being that a hypervisor runs in a special hardware provided privilege level (e.g. hypervisor mode), higher than the normal 'kernel mode', thus allowing it to run a broad range of software in a partition: from bare-metal C code, to a RTOS, to full third-party OS – e.g. Linux+ Android. I'm not sure that a Type-2 hypervisor would qualify, especially if the core services of the hypervisor require redirection to a Host OS such as Linux. We believe however that Type-1 hypervisors can be designed to meet the definitions of the separation kernel, as described in Section 1.3 TOE Overview.*
A5: *In many CPU architectures, there are additional privilege levels higher than the level of the separation kernel, running software independent of the separation kernel. The separation kernel must trust this software and is normally unable to vet, inspect or control this software. It can however limit / control and monitor communications between less privileged software running under the separation kernel, and this higher privilege software.*

Assets (compare Question 3.1):

A6: *Assets: Devices / IO Memory. We believe that handling of device/IO memory is distinct from AS.MEM since the devices may contain logic/software that can be controlled/influences by the partition with access to the device, and the partition may be able to circumvent separation kernel protections AS.MEM in particular through its control of the device. In particular, devices which are themselves bus-mastering should be addresses, since these bus-mastering devices can be programmed to independently access system memory, and this may bypass separation kernel memory protection. Certain hardware level protections can be implemented in hardware and should be used by the separation kernel. If not available, access to such devices should only be given to trusted partitions. Acceptable hardware level protections include System MMU and Bus Firewalls. These, with correct implementation, allow a separate software entity (e.g. the separation kernel) to limit the memory accessible by the bus-mastering device, independent of and non-bypassible by the partition that has access to the device, thus the separation kernel can limit the memory accessible to the device – e.g. it could give the device access only to the same memory ranges that its associated assigned partition has access to.*
A7: *We believe that a third (or more) asset classes for such hardware devices / IO memory be defined and addressed in the PPs.*

Side channels (compare Question 3.5):

A8: *Side channels: Many side-channels – such as the various Spectre/meltdown CPU issues and shared cache based channels, are present in modern CPU architectures. The separation kernel needs to balance the target security level with implementation complexity and system performance. A higher security posture should require careful considerations and mitigation for side-channels. For example, on multicore or multi-threaded CPUs, running software from different security domains concurrently on different cores, can open the system to information leaks, especially if the Level 1 caches, TLBs etc are shared between CPUs.*

A9: *You should discuss explicitly whether you are considering software-based side-channel based threats (e.g. Spectre).*

A10 *Finally, advanced hardware may support access from multiple partitions, and support isolation of these partitions interactions with the device in hardware. In this case, side channels in such hardware should also be considered.*

Assurance (compare Question 3.10):

A11 *Focus assurance "levels" on levels of AVA_VAN to avoid unnecessary red tape. For example, check out DTSec* [3].

A12 *The fields of application for CC- certified separation kernels seem to be high security products especially for the public government or similar. Therefore it seems necessary to have a high EAL.*

Scheme:

A13 One respondent suggested to *"create a government-funded non-profit that has its own scheme and can work internationally"* instead of focussing on government-assisted schemes.

Preliminary analysis

With regard to that, some questions were considered very leading (A1) – we tried to be "leading" to encourage answers at all. We generally did not try to push these into any direction. Many questions were based on questions previously discussed within certMILS. With regard to comment A2: the respondent has been invited to the group. With regard to comment A3: the CCUF technical community is open to all vendors. A4 asks for the relation of separation kernels and Type I/II hypervisors. A5 raises the topic that there also might be CPU privileges higher than the CPU privilege level the separation kernel is running in. A6 to A12 are discussed within their own focus, i.e. Questions 3.1, 3.5, 3.10. Concerning A13, this seems quite ambitious.

4.2 Section 2: Terms and definitions

2.1) Dou you find the description of "separation kernel" quoted in the beginning of the questionnaire appropriate? [y / suggest]

[Context: the description in the beginning of the questionnaire was: "A Separation Kernel (separation kernel) is a special kind of operating system that allows to effectively separate different partitions from each other. Applications themselves are hosted in those partitions. They can also be entire operating systems. The separation kernel is installed and runs on a hardware platform (e.g. embedded systems, desktop class hardware)."]

[8] ***A1:*** *Yes,* without further comment (5x)

A2: *Good basis to start with*

A3: It was pointed out that another term that needs to be defined are partitions. It was asked what guarantees the separation kernel provides.

A4: *It could be mentioned what a partition contains / what kind of resources are separated in general (hardware/software/physical?).*

A5: *Would include also drivers as an example of a partition.*

A6: *Shouldn't the description explicitly mention time and space?*

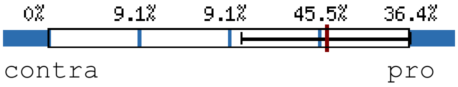
A7: *The current definition misses to highlight the cyber security aspect of the separation (so make the difference with other technologies using virtualization). Proposal: A Separation Kernel* *(separation kernel) is a special kind of operating system that allows to effectively and securely separate different partitions from each other and ensures no unauthorised communication or interference between them occurs. Applications themselves are hosted in those partitions. They can also be entire operating systems. The separation kernel is installed and runs on a hardware platform (e.g., embedded systems, desktop class hardware).*

Preliminary analysis

A3-A6 are discussed in context of Question 2.3. A7 indicates that the respondent expected that the topic of how partitions communicate is included in the definition of separation kernels.

2.2) The PP draft uses the term System Security Policy (SSP) for the set of configuration choices of a separation kernel. Do you agree to use that term?

[11] While mostly accepted, the term System Security Policy also had a vote of slight disagreement.



2.3) The PP draft uses the term "partition" for the domains separated by the separation kernel and defines it as follows:

A partition is a logical unit maintained by the separation kernel and configured by the SSP. For each partition, the separation kernel provides resources. Resources of a partition comprise physical memory and allocated CPU time for each CPU. A partition can host active subjects.

Do you agree with the term partition and its definition? [y / n+reason / suggest.]

[10]***A1:****"Yes", without further comment (2x)*

A2: *Partition looks OK. Active subject: how is this definition different from just saying "guest OS applications"*

A3: *The term "partition" could be misleading since it is also used for the (virtual) segmentation of hard drives. Maybe another term without a "historical" definition would be better, but I don't have an idea.*

A4: *Memory-mapped resources other than memory should not be excluded from the definition: device registers, access to HSM, co-processors or I/O interfaces.*

A5: *What is a subject? Also, can the partition access to other hardware resources (interrupt? DMA?)*

A6: *Would also include CPU(s), CPU registers and HW peripherals as resources of a partition.*

A7: *No / what about interrupts and device access? Also access to software services / capabilities can be restricted resources provided: physical memory, io device memory, CPU time, inter-partition communication conduits.*

A8: *I agree with the term and partially agree with the definition. The resources of a partition currently mentioned are only memory and CPU. I think these resources shall also contain means the partition will use to communicate either with the separation kernel, with other partitions or with outside world. As mentioned in section 1.3.2* [of the PP]*, the generic security features implemented by the separation kernel are separation in space, separation in time and control of communication between partitions – this later aspect is not really addressed by the document.*

Preliminary analysis

A2: see Question 2.5. A3 seems hard to realize, as the term partition is widely used in the separation kernel community. A4-A8 enumerate resources of a partition beyond memory and CPU.

2.4) The PP draft distinguishes partitions only as trusted or untrusted (and whether they are inside or outside of the TOE). Other criteria (e.g., fine-grained privileges) are consciously not attributed.

Would you state any objections or additional distinctions? [n / ...]

[9] ***A1:*** *"No", without further comment (2x)*

A2: *Trusted/untrusted is more aligned with access control. Also, time critical, non-time critical? Stressing the importance of enforcement of allocated CPU time...*

A3: *I agree with the current distinction. However, some more explanations and precisions shall be provided.*

- **A3.1:** *What does "untrusted" mean here?*
- **A3.2:** *How is a trusted partition outside of the TOE is as such by the TOE? How we make the difference between a trusted partition part of the TOE and those outside of the TOE?*
- **A3.3:** *Moreover, I have a problem with the fact that it is said (page 5 Section 1.3.2.6 use case "system with trusted partitions" and page 6 Section 1.4.1) that trusted partitions (even those outside of the TOE if my understanding is correct) have the potential to violate the SSP. If the SSP can be violated than the main scope of using such a separation kernel is lost. A SSP can be "wrong" from the security point of view but in this case the separation kernel cannot be held responsible for it. All the separation kernel shall guarantee is that the SSP cannot be violated. Maybe I misunderstood so reformulation/clarification is needed on this point.*

A4: *If they are within the TOE or not might be also motivated by other reasons that only security. For example, it can be for performance reasons too.*

A5: *It is important to allow ST definitions that are compatible with other standards (e.g. the ISO 26262), which may have additional partition attributes.*

A6: *Maybe the allowed connections/ communication channels between trusted "partitions" need to differentiated as an extra criteria since they need to be controlled by the TOE but are neither trusted nor untrusted in the way, partitions are defined as such.*

A7: *This is fine as long as it does not rule out refinement of privileges later on. In a more finegrained privilege / capability model, certain capabilities could be deemed security-critical and any partition having such a privilege would need to be classified as trusted.*

A8: *You can always create a PP later that refines this?*

A9: *Possibly a third type of partition: 'protected' meaning it is untrusted, however the software run within it is protected by the separation kernel. This protection may include – write protection – e.g. the separation kernel may explicitly limit the partition's ability to self modify its code, the separation kernel may perform integrity checking or monitor 9escry9ur of the partition, or the separation kernel may provide secure boot (e.g. cryptographic signature checking of the initial state of the partition).*

Preliminary analysis

A7 and A8 discuss the concept of refinement, which is generally possible if the PP defines common ground: the CC allow to refine a PP in an ST [4], Part 1, Section 9.3 if it levies "the same or more, restrictions on the TOE and the same or less restrictions on the operational environment of the TOE". A3.2 asks for the difference between trusted and untrusted partitions, and A4 mentions that the motivation for trusted partitions could also be performance. A3.1 suggests an explicit definition of an untrusted partition. Response A3.3 reflects on whether an SSP can be violated. Therefore, should the SSP be defined that it cannot be violated or should the term be changed? E.g., "partition-level configuration" or similar.

2.5) The PP draft defines the term "active subject" (which is used to define an attacker as follows):

> An active subject may consist of any executable machine instructions that are loaded during start-up or runtime of the separation kernel (i.e. become active) in the context of an untrusted partition. Executable machine instructions can come from, for example:
>
> (1) applications (e.g. binaries or libraries in formats such as ELF, COFF etc.) that the system integrator has initially installed,
>
> (2) on-the-fly downloads (including, e.g. "malware")
>
> (3) on-the-fly compilation of source code.

Do you agree with the term “active subject” and its definition? If not, why / what else to use / how to adapt the definition? [y / suggestion]

[10]***A1:***“*Yes*”, without further comment (2x)

On the scope:

A2: *Machine instructions can also be generated without on-the-fly compilation of source code, i.e., through self-modifying code (e.g., malware decrypting/unpacking itself). Should active subjects also cover compromised programs in other partitions, e.g., a successful ROP attack against a trusted partition? What about firmware compromise: are devices part of a partition (related: what about DMA protection)?*
A3: *Perhaps it would be useful to add (4) gadgets residing in trusted code.*
A4: *Saying “may consist” is too open-ended for an evaluation: is dead code […] in or not? A script file? Think about these things as “what is the decision rule for the evaluator/certifier?”*
A5: *The active subject would rather be the process, which executes the considered executable machine instructions.*
A6: *Yes & No. We need a definition of “active subject” describing the entities inside the partitions, which are executed to some job and, at this point, the proposed definition is ok. BUT, using this definition to define the attacker bothers me as the definition includes (among others) system integrator while this is the one defining the SSP! I think there is a conflict here. So I strongly disagree to use this definition for “attacker”. My comment is also related with the previous one [cross-lookup: Answer 3.1 to Question 2.4] : what does “untrusted partition” mean?*
A7: *downloads/compilations .... are also some sort of executables 10escr it? Why I we say – pre-configured applications / on-the-fly applications?*

On the term used:

A8: *Why “active”? There is a passive subject too? If yes, how do you define it?*
A9: *active agent? Active tenant?*
Related: Answer A2 to Question 2.3: “*Active subject: how is this definition different from just saying “guest OS applications*””

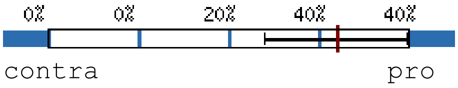
Preliminary analysis

A4 observes that “may consist” should be replaced by stronger phrasing. Other answers ask about where ROP, dead code, and self-modifying code would fit in. “Active subject” / “active entity” might need more explanation, e.g., the drafts avoid “guest applications” because a guest application could have multiple threads/tasks, each of which is an active subject in the context of evaluating an SFR.

Concerning A6, it may accidentally happen that the integrator merges a malicious application into an untrusted partition. However, in this case, the human being who is the attacker is the author of the malicious application and not the integrator. After further discussion, we could distinguish agents and human beings in the documentation of the PP itself.

2.6) Would you agree with the term “integrator” for the role that configures the TOE – assuming that it is properly defined?

[10] The term “integrator” for the role configuring the TOE was generally agreed with.



2.7) Are there any other terms or definition that you would question? [n / ...]

[4] ***A1:*** *I suggest constraining the TOE scope clearer. The picture suggests that the hardware is in, but the text says it is not. From the most active CC domain, the smartcard domain, we are not used to having hardware out of scope, so it is important to make explicit that the hardware is out of scope. However that immediately means that you have to be very exact on what is needed from the hardware to be suitable. If you don’t, ATE and AVA of the evaluation will be very, very expensive as the developer and evaluator have to show that all possible fringe implementations fulfilling the hardware requirements still lead to the TOE fulfilling the SFRs.*

Also note that the TSF (hence: TOE) has to implement all SFRs. There is a grey area where the environment (hardware here) 'just provides common functionality', but that is tricky. You are soon in the discussion whether an OS that only sets the MMU but does not actually enforce any access control, arguably misses an SFR-enforcing module (the MMU) and hence fails the evaluation. This is a hard (composition) problem.

A2: *If 'configures the TOE' includes signing the code (for secure boot), it should be clear that the integrator has an on-going role, more like the TOE landlord. Any changes to the TOE configuration require the integrator to use his keys to sign modified code. Parties that use the TOE but can't change its configuration are tenants.*

A3: *The separation kernel system may support decentralized control, or cascaded permissions – allowing a tiered set of integrators. E.g. there may be a primary integrator, responsible for the overall SSP, and then second level integrators responsible for a sub-set of the partitions.*

A4: *The terms untrusted and attacker should be clarified. Currently it is considered that the attacker can be inside a trusted partition. What about the attacker outside of the system trying to attack via the hardware interfaces?*

Preliminary analysis

About the TSF scope, the respondent interpreted the PP [1] Figure 1 different than its description, which could, e.g., be addressed by marking up "TSF scope" in the Figure 1 of the PP. A2 observes that the integrator not only establishes the initial integration, but sustains ongoing maintenance, such as signing code for updates. A3 raises the topic of tiered integrator chains, which so far have not explicitly been discussed in the PP.

4.3 Section 3 Generic PP properties

3.1) Are there more assets that you think any separation kernel has to protect and therefore should be included? Conversely, should we refrain from including Memory and CPU Time in the PP? [...]

[9] The topic of devices was raised 3 times.

A1: *It was suggested that IO Memory / Devices should be treated independently as well as interrupts access and DMA.*

A2: *It was suggested that memory should perhaps be more clearly stated to include both general storage, and registers.*

A3: *It was suggested that critical configuration registers could be memory mapped, and may be access controlled in the same way as general storage, but they aren't the same.*

Interrupts (raised 3 times):

A4: *separate interrupts and devices may need to be assigned different partitions. The implementation of the communications channel may be shared memory + interrupts.*

Memory:

A5: *It was suggested to distinguish different kinds of memory: exclusive to partition or shared with the separation kernel, the shared one being used for communication.*

A6: *Is Memory here DRAM, the physical memory address space, or some more general idea of "memory" as any kind of state-based storage. If it is the second case, some (legacy) architectures or devices may not use memory-mapped I/O and are thus not covered. Are CPU registers considered as part of memory in the sense of the third option? Certain CPU functionalities / extensions may also be partitioned, e.g., access to FPU, SIMD, AVX, SGX, performance counters, debug, crypto, and other accelerators.*

A7: *DMA*

The topic of inter-partition communication channels was raised 3 times:

A8: It was suggested to consider inter-partition communication channels, or the rights to control and create/delete these, as assets.

A9: It was pointed out that *it was unclear how the assets named in the protection profile were linked to communication channels between partitions, between a partition and the separation kernel, and between a partition and outside world can be expressed: Either put an additional note in the document how assets linked to communication and memory should be inserted, or another asset linked with partition communication should be defined. The implementation of the communications channel may be shared memory + interrupts, or a physical interface (PCIe, Ethernet). The implementation of the communications channel may be shared memory + interrupts, or a physical interface (PCIe, Ethernet).*

A10 It was suggested to add *CPU registers (user data), separation kernel and partition code and also possibly partition configuration.*

A11 It was suggested to explicitly say *what are the physical and logical resources accessible for each world (trusted, untrusted).*

Lastly, it was pointed out that

A12 *the security objectives OT.CONFIDENTIALITY and OT.INTEGRITY mention "For each asset", but there is currently only one asset concerned (AS.MEM)*.

Preliminary analysis

Devices, interrupts, different types of memory and inter-partition communication channels have been mentioned several times independently. If, for minimalism these are kept out of the base PP, guidance could to be quite clear how these are covered / why these are not covered.

A12: the formulation was intentional to ease extensibility.

3.2) Would you like to list important assets that some separation kernels do protect, i.e., as candidates for additional PP modules?

[3]
- *I/O memory management unit*
- *Hardware Security Modules and similar devices*
- *Controls for dynamic changes to the SSP:*
  - *Inter partition communication protections*
  - *CPU Time controls – e.g. control of changes to priority / timeslices*
  - *Memory controls – e.g. control of dynamic memory ownership / access changes*
- *Communication with high-privileged software.*
  - *Some separation kernels allow partitions to communicate with higher level software, e.g. ARM TEEs via TrustZone.*
  - *The separation kernel may limit the communications: limit access to at fine-grained API level, pointer checking in messages, secure identificationof the partition to the high privilege software.*
- *Power management / energy management*
  - *Manage a partition's power-management requests (e.g. vote aggregation) – Limit energy usage in different scenarios*
- *Hardware keys / Secure storage*
  - *access to per-partition secure storage (e.g. key store) or hardware keys (or providing a partition with derived keys from a hardwareroot key)*
- *Time of day*
  - *access or no-access to wall-clock time*
  - *secure distribution of time*
- *DMA, PCI, IOMMU,*
- *HW resources, such as peripherals. The separation kernel would control access to these peripherals. But is seems covered by the intended I/O MMU PP-module. Critical configuration registers? Interrupts, devices, CPU features, software capabilities (e.g., inter-partition communication channels)*

Preliminary analysis

The question and its answers do not target the base PP, but are relevant as feedback for structuring future modules.

3.3) Do you see any meaningful distinction between primary and secondary assets or objects versus resources for separation kernels?

[6]

- “*no*”
- One respondent asked about the differences between primary and secondary assets.
- another answer cautioned that the distinction only serves “*to structure your security problem definition -> objectives -> SFRs analysis for the reader. In evaluation anything that breaks anything of the SFRs is considered a fail, and the assets are only used to clarify edge cases there. Primary/secondary asset distinction is not relevant in an evaluation: the SFR seems broken, any asset is compromised -> fail*”.
- Another answer suggested that “*primary assets should relate to fundamental properties of the hardware, and mediating access to them secondary assets are software provided constructs – ie. Assets provided by the separation kernel by function of a software implementation*.”
- Similarly, there was a response (labelled “*perhaps*”): “*I think secondary assets are those assets which are important to system management, but don’t directly control (and can’t undermine) memory & CPU time. System credentials used in authentication.*”
- A third answer pointed to “*primary: see 3.1, secondary: see 3.2*”.

Preliminary analysis

Overall, there seems to be no pressing need for distinction of primary and secondary assets. If it is to be introduced, then it could be introduced along the line physical assets / logical assets. In this case, it might be easier to distinguish “logical” and “physical assets”.

3.4) The present PP draft largely sees the attacker as an entity that already exists in a partition:

> Threat agents are active subjects within an untrusted partition.
>
> For the computation of the attack potential, this may have the implication that factors measuring the effort needed to get inside the partitions come out small.

Do you have relevant applications for which these factors may matter and be relevant to achieve a certification according to a given EAL that would not be possible otherwise? [n / 13escry. Application]

[8] Several answers were critical of the concept of assuming that the attacker was not within the TOE:

- it was emphasized that “*an untrusted partition must be assumed to be actively malicious*”
- another respondent stated that it seemed reasonable to assume “*that the attacker is an active subject in the untrusted partition “for free”, so no points for say “access to the TOE”,* meaning that the *“separation kernel has to protect really well”.*
- “*No. We shall assume untrusted code really is untrusted, whatever the means to introduce malicious code within an untrusted partition.*”
- Similarly, as part of another response it was stated “*For untrusted partitions it is sound to assume they are compromised already as they are not trusted anyway.”*
- *“No”*

Three answers pointed to external threat agents:

- “*If there are devices that are neither trusted & in the TOE nor belong to an untrusted partition, then the definition may miss device-based threat agents. Similarly a trusted partition outside of the TOE might become compromised through SW vulnerabilities in the partition (if communication with the untrusted world is possible), and then be not considered as a threat agent as well.*”
- A similar answer pointed out that “*Threat agents may also be external, and attack the system through communication links – e.g. to cause information leak out of the partition. This mostly affects software within the separation kernel, not the separation kernel itself – since we don’t believe that the separation kernel should directly interface with external systems. The*

partitions or SPP themselves may need to use their own PPs / certification that addresses this. E. g. existing VPN PPs."

- A third answer stated "*This depends on how devices are partitioned and whether there are trusted partitions outside of the TOE.*"
- Similarly it was pointed out that "*Threat agent can also be from outside. For example, an attack coming from the network with the driver running in a trusted partition (for performance reasons) can be the threat agent.*"

Other answers:

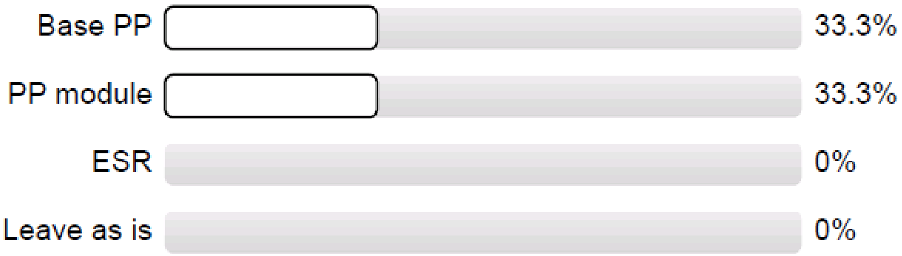
- *If one updates the untrusted application (or part of it) without any particular check for its integrity/authenticity or if the mechanisms put in place for these checks are not strong enough, the factors measuring the effort to get inside the partition should not be small.*
- *I don't understand the question. I think the PP needs to assume threat agents are sharing the multicore hardware with trusted/ legitimate partitions. Since you're using the term threat agent, my earlier comment [cross-lookup: Question 2.5 answer A5] about calling the active subject 'active agent' makes sense. Agents (or trusted agents) run in trusted partitions, threat agents run in untrusted partitions.*

Preliminary analysis

The concept of assuming that the attacker is not yet within a partition was observed several times. For instance, it was discussed, whether an attack path external threat agent => trusted partition would be in scope.

3.5) A possible limitation of separation kernels (aside from faulty hardware) is the exploitation of collaborative side-channels (covert channels). Assuming that an attacker has access to two or more partitions that should be separated according to the security policy, the attacker could exploit side-channels, such as timing channels (that always exist in the real world). Currently, this problem is not mentioned in the PP, but quite likely comes up during a certification according to a derived ST at higher EALs.

Where should the case of covert channels be discussed?



[12] One comment:

> *"Your question starts out discussing collaborative side channels, but the 'where should' question doesn't mention collaborative. Non-collaborative side channels should be discussed in the base PP. Collaborative side channels should perhaps be discussed in the PP module, or elsewhere."*

Preliminary analysis

See Question 3.9.

3.6) At present, the PP draft does not describe any Organizational Security Policies.

Do you think that there are OSPs, which always apply and should thus be included? [n / list]

[7] Most answers were negative (one exception).

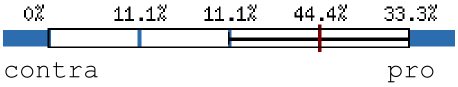
- "*no*" (4x)
- don't know about OSPs / what are OSPs (2x)
- "*least privilege*"

Preliminary analysis

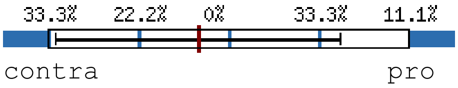
An OSP only has been mentioned once (least privilege). Least privilege can be meaningfully combined with separation kernels. However, it is not sure, whether this should be treated as a “must”. (E.g., it is difficult to evaluate, whether a policy applied to a separation kernel is really “least privilege”.)

3.7) Should the PP, intended to describe a minimal separation kernel, include FDP_RIP?

[9] When answered, whether the issue of residual information protection (FDR_RIP) should also be covered in a PP for a minimal separation kernel, participants were mostly in favour.



3.8) Should FDP_RIP alternatively be provided by one or several PP module(s)?



[9] Whether residual information flow should be split off to a PP module seems more controversial, with the tendency against a separate module.

3.9) If you would like to see FDP_RIP included, can you describe which residual information flows via which resources it should address?

[6] This question elicited both high-level and low-level answers:

Some answers at the level of resources/communication channels explicitly maintained by the separation kernel:

A1: *Reallocation of resources to another partition*
A2: *all sensitive information (classification of information needs to be done before starting the TOE) which flows between partitions or to external resources.*
A3: *any separation kernel must ensure that partition-specific machine state cannot leak to other partitions – basic part of the separation kernel scheduler*
A4: *additional partition resources dealing with communication*.

Answers touching the micro-architectural level:

A5: *accessible registers and freed memory pages, for side channel protection a lot more (micro)architectural components need to be considered, e.g., caches, branch prediction, performance counters, TLBs...*
A6: The most detailed answer distinguished “Transitioning between partitions, reset of a partition and system crash reset”:

- *Transitioning between partitions.*
  - *sanitizing shared resources (memory)*
  - *sanitizing CPU resources which may be used as covert channel indirectly – e.g. speculative access to system register contents.*
- *Reset of a partition – e.g. after incorrect behaviour or malware detected*
  - *sanitation of partition resources*
  - *note: this may be expensive (in time) for the separation kernel, and deferred to a trusted partition to implement.*
- *System Crash reset*
  - *sanitization of resources / assets on system crash – e.g. prior to reboot.*
  - *informing higher level privilege software (e.g. trustzone) of sensitive memory locations that should be sanitized in case of system reset (e.g. watchdog reset, higher level system detection of security violations, etc).*

Preliminary analysis

Concerning residual information flow, respondents discussed whether to distinguish collaborative and non-collaborative side channels. It was raised that there is residual information protection of resources maintained at the separation kernel level and at the microarchitecture-level information flow.

3.10) The PP draft suggests making a standard choice of assurance packages according to an EAL. Do you think that there are assurance activities that should be mandatory for a minimal separation kernel beyond those of an EAL?

[6] ***A1:*** "No" (3x), one with the comment "*EAL ones are good but augmented, ALC_FLR.3 should be mandatory. A high level of AVA_VAN should be nice to have*."

A2: *"unsure"*

A3: *There is a trend to make ALC_FLR.2 mandatory (forcing developers to have a bug fix procedure they actually use). ASE_TSS.2 can help under EAL4 to give users and evaluators more insight in how the TOE is constructed.*

A4: *This is a question for the industry and requirement setters. I personally would consider ADV_IMP.2 and AVA_VAN.3 the minimum for which separation kernel evaluations start to have actual added value.*

A5: *Avoid use of EALs and create a custom assurance package with the minimum necessary product-specific evaluation, such as AVA_VAN and nothing more. Organizational / process based assurance can and will be evaluated naturally by labs but lower the efficiency/cost bar by not prescribing this. This is because many companies that have the best processes in place deliver really bad code – you need to test the code, not the company. Consider using an extended package with a base PP (like DTSec) if you want to handle multiple assurance levels.*

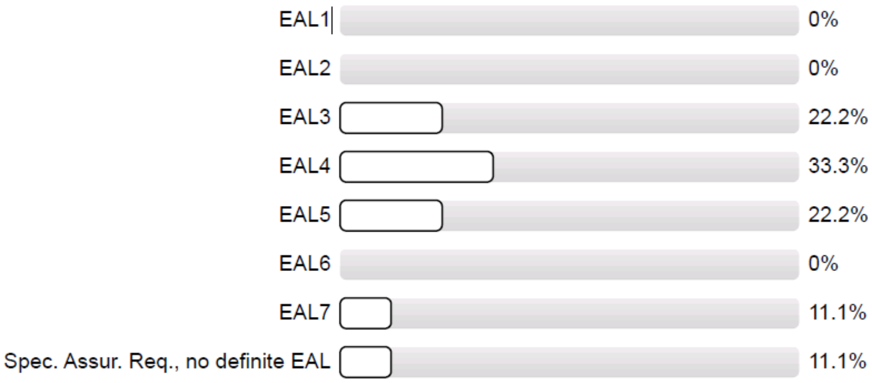
Preliminary analysis

One respondent raised that no EAL level should be used, like in DTSec for medical devices. Looking up the DTSec Protection Profile and its Extended Packages [3], we found a custom SAR packaging consisting of ASE (ST work units), AVA_VAN.x, with *x* depending on the chosen extension package, and, because, DTSec is in the medical domain, (not applicable here) conformance to life-cycle requirements coming from the medical IEC 62304. The respondent observed that such a streamlined approach (i.e., ASE + AVA + IEC 62304 equivalents) might help evaluations to focus on building robust systems.

A high level of AVA_VAN was mentioned two times. ALC_FLR was mentioned twice.

3.11) Otherwise, what should be the minimal EAL for the PP

[9]



Preliminary analysis

See discussion at Questions 1.7 and 3.10.

3.12) The PP draft and the CPU-time modules intentionally use only SFRs instantiated from the catalogue proposed in CC Part 2. Do you think that the choice made sufficiently represents the security functionality of a minimal separation kernel? [y / add. SFR]

[5]
- *"unsure (time is missing to analyse further)"*
- *"yes" (3x)*

Preliminary analysis

While it is tempting to assume that we have found a perfect SFR set, it is more likely, also taking into account the low response rate for this particular question, that respondents did not have time for detailed SFR analysis. This is acceptable for a first community reach-out.

3.13) Besides the mandatory CPU-time modules, the following PP modules have been considered (though not published):

I/O MMU Module, Cryptographic Services Module, Information Flow Control Module, Management Module, Network Interface Partitioning Module, HAL Module, CPU-time Modules, Secure Boot Module, Security Audit Module, Storage Module, Secure update Module.

Do you miss anything that would likely not be covered by one of those modules? [n / suggest]

[6] ***A1:*** *No* (4x)
A2: *Side Channel Module, SW Attestation Module*
A3: *One answer pointing to the following modules*

- *Residual information protection module(s)*
- *Memory management module – e.g. ability to dynamically modify partition's memory and I/O device assets*
- *VPN Module – part of network interface partitioning?*
- *OTA Module – over the air updates of system, partitions*
- *Secure Partition Loading – run-time loading of new/reloading existing partitions securely.*

One answer additionally pointed out that "*it would already be great if we could have some of these PP-modules, such as I/O MMU, information flow control (almost mandatory for all cases, it should be a priority for next PP-module development), secure boot, secure update.*"

Preliminary analysis

New module proposals are SW attestation in A2, side channels in A2 and residual information protection in A3. Something like memory management from A3 could be part of a generic management module. VPN could be part of network interface partitioning, as remarked by respondent in A3. An OTA module (A3) could be part of secure boot. Secure partition loading could be part of secure boot.

4.4 Section 4 If you had time to read the PP

4.1) Do you find the structure of the PP clear enough? [y / n+explain]

[8]
- *Yes (5x)* (without further comment)
- *I like its simplicity, keep it that way!*
- *Yes, a bit too verbose.*
- *Some things could be more detailed but for a first draft of a PP it's ok with the chosen abstraction layer.*

Preliminary analysis

The structure does not seem to be difficult to understand.

4.2) Are there SFRs, which you would formulate differently or like to see more explanations? [n / suggest.]

[2]
- *More explanations on possible options on how possible assignment options, as application notes.*
- *No*

Preliminary analysis

This was not a popular question, perhaps as it was very technical. Overall, this did not elicit new SFRs, which is OK. It was suggested to give more explanations on how assignment options can be realized.

4.3) Would you require additional information in order to understand the PP? [n / suggest]

[3] ***A1:*** *No (2x)*

A2: *Yes, my previous comments are showing it. To summarize, the comments addressed:*

- ***A2.1:*** *enhance separation kernel definition to highlight cyber security aspects and differentiate from other virtualisation products*
- ***A2.2:*** *active subject to define the attacker – disagree with current proposal [cross-reference: answer A6 to Question 2.5]*
- ***A2.3:*** *clarifications about the trusted partitions not part of TOE versus trusted partitions part of TOE*
- ***A2.4:*** *define what untrusted means*
- ***A2.5:*** *partitions communications not addressed by the PP while it is said this feature is part of the generic security features of the separation kernel*

Preliminary analysis

A2.1, A2.3-A2.5 are discussed in the context of Question 2.4. A2.2 is discussed in the context of Question 2.5.

4.4) Do you think that the PP could be applied to the separation kernels you are interested in? [y / n+expl.]

[5] ***A1:*** *Yes* (3x)

A2: *yes if suggested clarifications are addressed*

A3: *Maybe. Some hypervisors have similar robustness and memory protection requirements but are unable to provide time guarantees. If the time guarantees could be made optional in the PP (ST must declare whether it is supported), it may broaden the reach. Most kernels that make time guarantees are unable to prove those guarantees anyway. Even so-called real-time kernels. Unless you can cover the entire system with a static worst case analysis with all critical sections, good luck...*

Preliminary analysis

A3 suggests making the asset ‘CPU time’ optional.

4.5) Is there anything that, in your opinion, would hinder the application of the drafted PP for the certification of a product (TOE) according to an ST derived from it? [n / y+expl.]

[5]

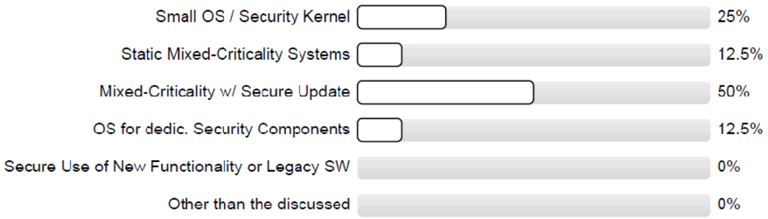
- *No* (4x)
- *TOE scope not including hardware is a big hurdle for many certification bodies, so I would suggest making that very clear and then getting the PP certified under one of the SOGIS issuing schemes (ANSSI, BSI, NSCIB) to close that discussion off. Composition is a big hairy problem you have not addressed.*

Preliminary analysis

It has been observed that the PP scope does not include hardware, which needs to be carefully communicated. While the PP intentionally excludes hardware, a feasible approach may tackle hardware during the CC evaluation.

4.6) Is your use case covered by the ones discussed in the PP?

[8]



Preliminary analysis

Use cases mentioned in Section 1.3 of the PP draft seem to be adequately realistic.

4.7) If any, what are your objections against the TOE scope described in the PP draft? [n / y+expl.]

[5] ***A1:*** *Hardware not being in scope*
A2: *The trusted partition makes me uncomfortable as part of the TOE. Goes against the concept of an separation kernel IMHO.*
A3: *none* (3x)

Preliminary analysis

For A1, it would be possible improve the description of the role of hardware during evaluation. (I.e., hardware is not out of scope of evaluation.)

4.8) If you have read one or more of the CPU-time modules (! Excellent !):

Is there any method of "time-slicing" that you see not covered by the available modules, or anything else that you would like to mention regarding one or more of those modules? [n / suggest]

[8] ***A1:*** *In combination with some RTOS used for functional safety functions, there is a need to distinguish pre-emptive from deadline based execution models.*
A2: *An attacker could target the deadline of a safety function.*
A3: *Very good PP-module indeed.*
A4: *Without doing a certification of this, I am not certain that the current SFR is clear enough to make a pass/fail decision on. I imagine that these kind of timeliness requirements are actually pretty exact, i.e. there is a way to say "1ms over the allotted time = fail".*
A5: *In the research literature there are additional modes of time sharing – e.g. deadline based scheduling. Energy aware scheduling (including big-little) while we primarily make of the methods on time-slicing already describes, leaving the option open to additional methods makes sense.*
A6: *No* (2x)
A7: *Not yet read it*

Preliminary analysis

Additional methods such as deadline based scheduling have been proposed (see A5, A1). It was proposed to include a description in the guidance how to evaluate them, i.e., there is a way to say "1ms over the allotted time = fail", concerning A4.

Chapter 5 Summary of topics

Participants of the survey had the choice to indicate a level of attribution / confidentiality. Most have chosen public, i.e., to link their affiliation with their answers. Other choices were to have their participation mentioned in a summary without direct linkage, or to stay anonymous.

Beyond the publication of this white paper, it is also planned to publish and discuss personalized answers given in the survey as is on the CCUF discussion forum. This only applies to the dataset of participants, who opted for public use. This is planned for beginning of 2019 on the web platform, as well as in form of a presentation on an upcoming CCUF (e.g. via the online forum).

Some of the summarized inputs touch aspects that did not receive enough coverage in the current draft, need further concretization or have been overlooked.

5.1 Observations identified by preliminary analysis

The following is a list of relevant topics identified in our preliminary analysis (see Chapter 4). These will be made available to the CCUF to continue the discussion. In a tentative classification, Section 5.1.1 summarizes observations covering definitions, while Section 5.1.2 presents observations regarding the scope of the PP.

5.1.1 Definitions/clarification

One respondent suggested to differentiate the certMILS work from a previous sunset effort, *SKPP* [5] (Q1.4). The aspect 'hardware', and how it is presented in the TOE/TSF description, was raised in in Q1.8 A3, Q2.7 and Q4.5. There seems to be interest in an explanation regarding SFR *possible assignment options* (Q4.2). For the definition of a *partition and its assets,* respondents suggested some other resources, e.g., interrupts, devices, and various forms of memory (Q2.3 A2, A3.1). The *treatment of communication between partitions* was mentioned in Q2.1 A7, Q2.3 A8, Q2.4 A6, Q3.1, Q4.2, Q4.3. Issues in the context of *trusted partitions* were, the topic how to distinguish trusted partitions that are part of the TOE, from trusted partitions that are not part of the TOE, as well as the aspect of performance (Q2.4). Conversely, one respondent also missed the definition of an *untrusted partition* (Q2.4 A3.1). For *active subjects*, some additional scenarios (ROP, self-modifying code) have been identified. Moreover, alternative terms have been suggested in Q2.5 A2. It was observed that the current definition of the *SSP* allows it to be violated (Q2.4 A3.3). The term integrator was generally agreed with, and it was pointed out that the integrator extends its role beyond initial setup throughout maintenance (e.g., signing updates, see Q2.7 A2).

5.1.2 Scope

It was suggested to explain CPU levels and the relation of separation kernels to Type 1/Type 2 hypervisors (Q1.4, Q1.8 A4+A5). Additional scheduling options such as deadline-based scheduling were mentioned, it was pointed out, how to evaluate these, and it was questioned, whether time guarantees shall be mandatory (Q4.4, Q4.8). In the context of EAL selection, one respondent suggested to analyse the DTSec SAR approach (i.e., no predefined EAL, ASE + AVA + domain specific assurance). Regarding the topic 'residual information flow', a respondent suggested to distinguish collaborative and non-collaborative side channels. A respondent differentiated between information flows generated by explicit resource management and reallocation of the separation kernel, and microarchitectural-level information flow (Q3.9). In the discussion of threat agents, a new attack path, external threat agent → trusted partition, has been raised (Q3.4).

5.2 How to proceed

It was suggested to ensure that more separation kernel vendors are involved (Q1.8 A3). As outlined above, we plan to submit the results and analysis of this document to the CCUF SK-TC/WG for discussion.

Chapter 6 Summary

We have made the certMILS PP available for review by separation kernel developers, hardware vendors, certification bodies, evaluation labs and security professionals/users to provide a draft for a future community PP. We asked how the PP draft can be applied to the separation kernels of MILS platform providers (separation kernel vendors) and how well the PP draft addressed requirements of users such as system integrators and report on the results. To the best of our knowledge, this is the first poll on a protection profile draft in questionnaire form. Given that the field is highly specialized, extensive and meaningful feedback has been obtained by the means of the questionnaire. The PP was generally appreciated, to be clearly structured and understandable (Section 4.4, Questions 4.1 and 4.3) and, apart from cautions regarding the scope, its interoperability (Section 4.4, Questions 4.4, 4.5 and 4.7). Furthermore, we have obtained feedback regarding terms and fundamental definitions, which we assume, will be helpful to evolve the PP. Other feedback relates to detailed aspects such as CPU privileges, timing schedules, residual information flow and external threat agents. Specific discussion points are listed in the previous chapter. We plan to share this feedback for further discussion at the CCUF Separation Kernel TC to enable improvements to the base PP, modules, as well as supporting guidance.

6.1 Acknowledgements

We would like to thank all participants for their contributions to the feedback survey on the Base Separation Kernel PP draft. Beyond those that need to stay without attribution for confidentiality reasons, we like to thank

Axel Söding-Freiherr von Blomberg, OpenSynergy, DE

Carl van Schaik, Cog Systems Pty Ltd, AU

Christoph Baumann, KTH Royal Institute of Technology, SE

Cristina Simache, Continental, FR

Geoffrey Waters, NXP, US

Guillaume Dufay, Prove&Run, FR

Julien Delange, Reblochon Development Company, US

Chapter 7 List of abbreviations

Abbreviation	Translation
ALC	CC SAR assurance class for life-cycle support
ASE	CC SAR assurance class for security target evaluation
ATE	CC SAR assurance class for tests
AVA	CC SAR assurance class for vulnerability assessment
AVA_VAN	CC SAR assurance family for vulnerability assessment
CC	Common Criteria [4]
CCUF	Common Criteria Users Forum
cPP	Collaborative Protection Profile
CPU	Central Processing Unit
DSC	*DSC [Dedicated Security Components]*
I/O MMU	Input/Output Memory Management Unit
MMU	Memory Management Unit
OS	Operating System
OSPP	Operating System Protection Profile
PP	Protection Profile
ROP	Return-oriented programming
RTOS	Real-time operating system
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SK	Separation Kernel
ST	Security Target
TC	Technical Community
TOE	Target of evaluation
TSF	TOE security functionality
WG	Working Group

Chapter 8 Bibliography

[1] certMILS, “Whitepaper on Base Separation Kernel Protection Profile Draft,” 2018.

[2] certMILS, “Whitepaper on List of extensions of base PP,” 2018.

[3] Diabetes Technology Society Standard for Wireless Device Security (DTSec), “Protection Profile for Connected Diabetes Devices (CDD),” 2017. [Online]. Available: https://www.diabetestechnology.org/dtsec.shtml?ver=4.

[4] Common Criteria Sponsoring Organizations, “Common Criteria for Information Technology Security Evaluation. Version 3.1, revision 5,” April 2017. [Online]. Available: http://www.commoncriteriaportal.org/cc/.

[5] Information Assurance Directorate, “U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness. Version 1.03,” June 2007. [Online]. Available: https://web.archive.org/web/20110108022547/http://www.niap-ccevs.org/pp/pp_skpp_hr_v1.03.pdf.

Chapter 9 Annex 1

Questionnaire as used for the online survey.

contents by: Separation Kernel Working Group
questionnaire contact: Thorsten Schulz
Feedback on the Separation Kernel Base PP draft

□ Activate contrast mode

1 2 3 4 5

1 General

The separation kernel working group seeks to establish a separation kernel Protection Profile (PP). A draft of a possible PP and the mandatory CPU-Time modules have been uploaded, which is based on the ESR; some previous work in the certMILS research project; and also tries to take into account some first feedback received via CCUF. The intent of the PP is not yet to fix anything, i.e. everything is open to discussion, which this questionnaire aims to support. The PP has been drafted as a modular PP, according to CC v3.1 R5 and attempts to cover a broad spectrum of separation kernels. That is, the current base PP draft intends to cover only a minimal set of features, such that Security Target (ST) authors do not need to justify deviations from the PP. There is a set of three PP modules, the inclusion of one of which is considered mandatory (the CPU-time Modules). Extended functionality should be covered by additional PP modules.

Among others, the PP draft describes a separation kernel as follows:

A Separation Kernel (SK) is a special kind of operating system that allows to effectively separate different partitions from each other. Applications themselves are hosted in those partitions. They can also be entire operating systems. The SK is installed and runs on a hardware platform (e.g. embedded systems, desktop class hardware).

Editorial notes:

There are no mandatory questions, you can leave any question unanswered.
Though, for you feedback to be submitted to the collection system, **you must go to the last page (5)** and use the **submit button**.

Some questions ask for (dis)agreement or alternative proposals. For (dis)agreement please insert 'y' / 'n' to indicate an answer, or provide suggestion/explanation as an alternative.

The questionnaire is hosted on servers on the premised of the University in Rostock, Germany. The questionnaire's raw data will be handled internally. The feedback will be accumulated and published in a public report in the CCUF Separation Kernel Working Group as well as a project report by 2019. Please indicate your choice of feedback attribution on the last page of the questionnaire.

If you have issues with the questionnaire, please contact [thorsten.schulz@uni-rostock.de].

1.1 Please indicate whether you are representing a separation kernel developer, integrator, CC evaluation facility, CC certification body, etc.

1.2 Do you think that there is a need for a separation kernel PP?

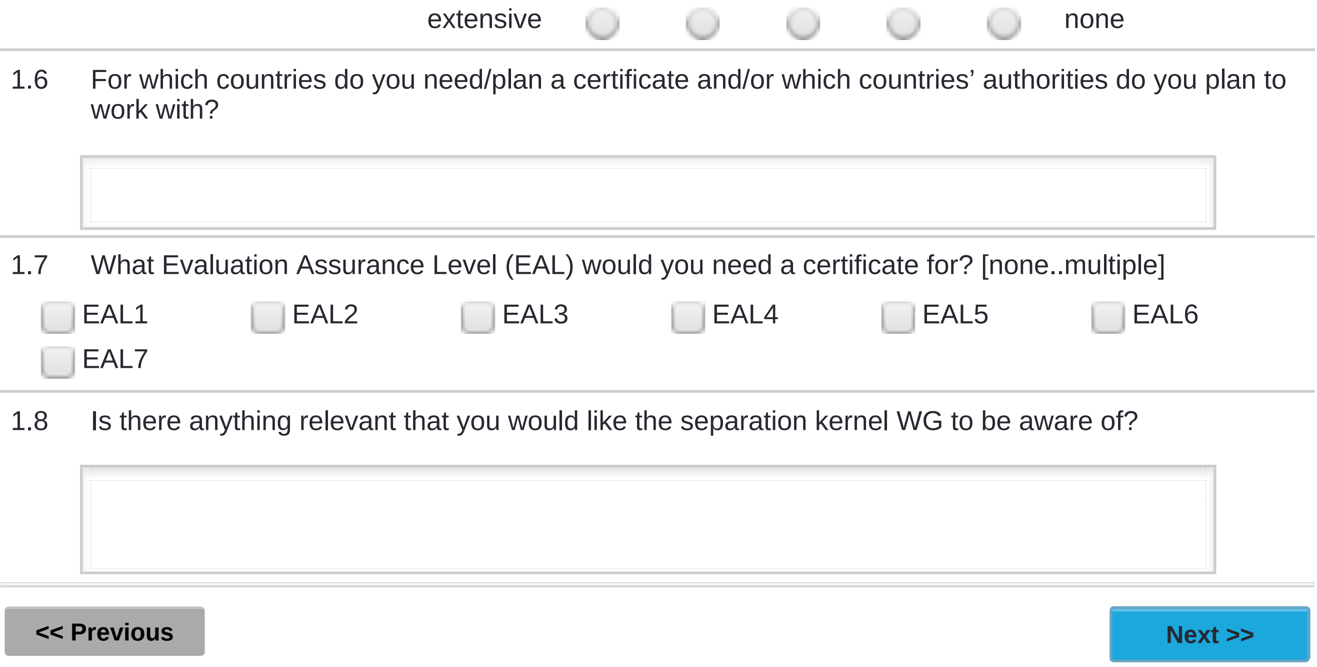
pro ○ ○ ○ ○ ○ con

1.3 Do you agree with the strategy to produce a modular separation kernel PP and its proposed scope?

pro ○ ○ ○ ○ ○ con

1.4 Do you think that the title “Separation Kernel Protection Profile” is appropriate? If not, which name(s) would you prefer?

1.5 Do you already have experience with the CC certification of a separation kernel?



1.6 For which countries do you need/plan a certificate and/or which countries’ authorities do you plan to work with?

1.7 What Evaluation Assurance Level (EAL) would you need a certificate for? [none..multiple]

□ EAL1 □ EAL2 □ EAL3 □ EAL4 □ EAL5 □ EAL6
□ EAL7

1.8 Is there anything relevant that you would like the separation kernel WG to be aware of?

contents by: Separation Kernel Working Group
questionnaire contact: Thorsten Schulz
Feedback on the Separation Kernel Base PP draft

□ Activate contrast mode

1 | **2** | 3 | 4 | 5

2 Terms and Definitions

As a document specific to security evaluations according to the CC, the PP and its modules use a certain, abstract language that may require time to get accustomed with. Terms used should, however, be in line with the CC, be vendor neutral – as far as possible, and be defined unambiguously.

2.1 Dou you find the description of “separation kernel” quoted in the beginning of the questionnaire appropriate? [y / suggest]

2.2 The PP draft uses the term System Security Policy (SSP) for the set of configuration choices of a separation kernel. Do you agree to use that term?

pro ○ ○ ○ ○ ○ con

The PP draft uses the term “*partition*” for the domains separated by the separation kernel and defines it as follows:

A partition is a logical unit maintained by the separation kernel and configured by the SSP. For each partition, the separation kernel provides resources. Resources of a partition comprise physical memory and allocated CPU time for each CPU. A partition can host active subjects.

2.3 Do you agree with the term partition and its definition? [y / n+reason / suggest.]

2.4 The PP draft distinguishes partitions only as *trusted* or *untrusted* (and whether they are inside or outside of the TOE). Other criteria (e.g., fine-grained privileges) are consciously not attributed. Would you state any objections or additional distinctions? [n / ...]

The PP draft defines the term “*active subject*” (which is used to define an attacker as follows):

An active subject may consist of any executable machine instructions that are loaded during start-up or runtime of the separation kernel (i.e. become active) in the context of an untrusted partition. Executable machine instructions can come from, for example:
(1) applications (e.g. binaries or libraries in formats such as ELF, COFF etc.) that the system integrator has initially installed,
(2) on-the-fly downloads (including, e.g. “malware”)
(3) on-the-fly compilation of source code.

2.5 Do you agree with the term “*active subject*” and its definition? If not, why / what else to use / how to adapt the definition? [y / suggestion]

2.6 Would you agree with the term “*integrator*” for the role that configures the TOE - assuming that it is properly defined?

pro ○ ○ ○ ○ ○ con

2.7 Are there any other terms or definition that you would question? [n / ...]

<< Previous

Next >>

contents by: Separation Kernel Working Group
questionnaire contact: Thorsten Schulz
Feedback on the Separation Kernel Base PP draft

□ Activate contrast mode

3 Generic PP Properties

The PP draft consciously limits itself to the assets *Memory* and *CPU Time*. This is motivated by a view that other assets can either be represented as one of these two seen from a different viewpoint and would be different for different separation kernels.

3.1 Are there more assets that you think *any* separation kernel has to protect and therefore should be included? Conversely, should we refrain from including *Memory* and *CPU Time* in the PP? [...]

3.2 Would you like to list important assets that *some* separation kernels do protect, i.e., as candidates for additional PP modules?

3.3 Do you see any meaningful distinction between primary and secondary assets or objects versus resources for separation kernels? (please name examples)

3.4 The present PP draft largely sees the attacker as an entity that already exists in a partition:

Threat agents are active subjects within an untrusted partition.

For the computation of the attack potential, this may have the implication that factors measuring the effort needed to get inside the partitions come out small.
Do you have relevant applications for which these factors may matter and be relevant to achieve a certification according to a given EAL that would not be possible otherwise? [n / descr. application]

A possible limitation of separation kernels (aside from faulty hardware) is the exploitation of collaborative side-channels (covert channels). Assuming that an attacker has access to two or more partitions that should be separated according to the security policy, the attacker could exploit side-channels, such as timing-channels (that always exist in the real world). Currently, this problem is not mentioned in the PP, but quite likely comes up during a certification according to a derived ST at higher EALs.

3.5 Where should the case of covert channels be discussed?

□ Base PP □ PP module □ ESR □ Leave as is

3.6 At present, the PP draft does not describe any Organizational Security Policies.

Do you think that there are OSPs, which always apply and should thus be included? [n / list]

At present, the base PP draft does not include Residual Information Protection SFRs (FDP_RIP), which could be used to model the cleaning of residual information (e.g., at the transition between two partitions). In terms of generic properties, FDP_RIP could be described (->CCUF) as sanitization of shared resources.

3.7 Should the PP, intended to describe a minimal separation kernel, include FDP_RIP?

pro ○ ○ ○ ○ ○ con

3.8 Should FDP_RIP alternatively be provided by one or several PP module(s)?

pro ○ ○ ○ ○ ○ con

3.9 If you would like to see FDP_RIP included, can you describe which residual information flows via which resources it should address?

3.10 The PP draft suggests making a standard choice of assurance packages according to an EAL. Do you think that there are assurance activities that should be mandatory for a minimal separation kernel beyond those of an EAL?

3.11 Otherwise, what should be the minimum EAL for the PP? [single choice]

○ EAL1 ○ EAL2
○ EAL3 ○ EAL4
○ EAL5 ○ EAL6
○ EAL7 ○ Spec. Assur. Req., no definite EAL

3.12 The PP draft and the CPU-time modules intentionally use only SFRs instantiated from the catalogue proposed in CC Part 2. Do you think that the choice made sufficiently represents the security functionality of a *minimal* separation kernel? [y / add. SFR]

3.13 Besides the mandatory CPU-time modules, the following PP modules have been considered (though not published):

I/O MMU Module, Cryptographic Services Module, Information Flow Control Module, Management Module, Network Interface Partitioning Module, HAL Module, CPU-time Modules, Secure Boot Module, Security Audit Module, Storage Module, Secure update Module.

Do you miss anything that would likely not be covered by one of those modules? [n / suggest]

contents by: Separation Kernel Working Group
questionnaire contact: Thorsten Schulz
Feedback on the Separation Kernel Base PP draft

□ Activate contrast mode

1 | 2 | 3 | **4** | 5

4 If you had time to read the PP

4.1 Do you find the structure of the PP clear enough? [y / n+explain]

4.2 Are there SFRs, which you would formulate differently or like to see more explanations? [n / suggest.]

4.3 Would you require additional information in order to understand the PP? [n / suggest]

4.4 Do you think that the PP could be applied to the separation kernels you are interested in? [y / n+expl.]

4.5 Is there anything that, in your opinion, would hinder the application of the drafted PP for the certification of a product (TOE) according to an ST derived from it? [n / y+expl.]

4.6 Is your use case covered by the ones discussed in the PP?

○ Small OS / Security Kernel
○ Static Mixed-Criticality Systems
○ Mixed-Criticality w/ Secure Update
○ OS for dedic. Security Components
○ Secure Use of New Functionality or Legacy SW
○ Other than the discussed

4.7 If any, what are your objections against the TOE scope described in the PP draft? [n / y+expl.]

4.8 If you have read one or more of the CPU-time modules (*! excellent !*):

Is there any method of “time-slicing” that you see not covered by the available modules, or anything else that you would like to mention regarding one or more of those modules? [n / suggest]

<< Previous

Next >>

contents by: Separation Kernel Working Group
questionnaire contact: Thorsten Schulz
Feedback on the Separation Kernel Base PP draft

□ Activate contrast mode

1 | 2 | 3 | 4 | **5**

5 Attribution

5.1 Please choose how we may attribute and publish your feedback. (It will on any choice be included in the accumulated report)

○ Upload to the CCUF Separation Kernel WG/TC with attribution for discussion
○ Accumulate feedback to collected answers - attribution in acknowledgement, not linked
○ Accumulate feedback to collected answers - without any attribution

5.2 You can add any contact information here, e.g., name, e-mail, organization. Please indicate, which we may use for above attribution.

This questionnaire is research work supported by the certMILS EU-project. Some Participants of the CCUF Working Group are part of the certMILS project.
The project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 731456.

<< Previous

Submit