FFI-rapport 2015/00456

# Information sharing across security domains

Nils Agne Nordbotten, Federico Mancini, Bodil Hvesser Farsund, Raymond Haakseth, Anne Marie Hegland, and Frode Lillevold

FFI Forsvarets forskningsinstitutt

FFI-rapport 2015/00456

# Information sharing across security domains

Nils Agne Nordbotten, Federico Mancini, Bodil Hvesser Farsund, Raymond Haakseth,
Anne Marie Hegland, and Frode Lillevold

Norwegian Defence Research Establishment (FFI)

10 August 2015

FFI-rapport 2015/00456

1294

P: ISBN 978-82-464-2558-0

E: ISBN 978-82-464-2559-7

## Keywords

Sikkerhetsdomener

Datautveksling

Informasjonsmerking

Guard

## Approved by

| | |
|---|---|
| Raymond Haakseth | Project Manager |
| Anders Eggen | Director |

## Summary

Security is recognized as one of the main technological challenges in realizing the potential of Network Based Defence. In particular, cross-domain information sharing is subject to significant security concerns, especially when there is a larger security span between the domains.

This report considers alternative approaches to enable sharing of information between security domains, with a focus on providing the necessary assurance in the information flow control. This includes transfer mechanisms such as one-way diodes, manual review, security filters, and guards, as well as access solutions providing users access to multiple security domains.

Guards using security labels as part of the basis for their release decisions are found to provide the most general solution for two-way automatic information transfer between domains. The correctness of the security labels is then critical for secure information flow control. While cryptographic mechanisms can be used to ensure the authenticity of the labels, a more fundamental problem is how to ensure that the security labels are correct in the first place. The report considers different approaches to provide sufficient confidence in the correctness of security labels, using content and origin as the two bases for establishing the security properties (e.g., confidentiality classification) of a data object. Several mitigations are also identified to further enhance security in cross-domain transfer scenarios.

The specific information exchange requirements of a given scenario, as well as the operational environment, provide the basis for choosing among the different cross-domain solutions presented. Domain isolation should still be the default unless information exchange requirements justify a cross-domain interconnection. Where a one-way information flow from Low to High is sufficient, a one-way diode is preferable. Where two-way transfer of information objects between security domains is required, a label based guard should generally be used although the use of a simpler security filter may be justified in lower risk scenarios. Provided that security labels can be trusted to be correct, a label based guard can provide significantly better protection against data exfiltration attempted by a human insider or targeted malware being present on the High side, than a simpler security filter. However, in many scenarios, it can be a significant challenge to assure the trustworthiness of security labels. Access solutions, providing access to multiple security domains from a single machine, have the advantage of limiting actual information transfer between security domains and do not require labelling of information. However, a centralized approach seems to be required in the case of many users.

The report also contains two appendixes:Appendix A provides a survey of existing solutions and literature with regard to guards, security filters, and security labelling. While many solutions are available providing a wide range of functionality, most solutions are found to be based on operating systems providing limited assurance (evaluated at EAL4), indicating that they are not certifiable at higher assurance levels. Appendix B provides an overview of concepts, technologies, and products that can potentially be used in order to build cross-domain solutions, including a survey of high assurance operating systems, hardware security mechanisms available on commodity platforms, and Attribute Based Access Control.

## Sammendrag

Informasjonssikkerhet er ansett som en av de viktigste teknologiske utfordringene for å realisere Nettverksbasert Forsvar. Spesielt byr utveksling av informasjon mellom sikkerhetsdomener på store sikkerhetsutfordringer, særlig hvis sikkerhetsspennet mellom domenene er stort.

Denne rapporten vurderer ulike metoder for deling av informasjon mellom sikkerhetsdomener, med fokus på å oppnå tilstrekkelig tillit til kontrollen av informasjonsflyten. Dette inkluderer enveis dioder, manuelle review-prosesser, sikkerhetsfiltre og guard-løsninger. I tillegg diskuteres løsninger for å gi aksess til informasjon i flere sikkerhetsdomener, uten nødvendigvis å koble dem sammen.

Guard-løsninger betraktes som den mest generiske tilnærmingen. Disse løsningene gir nødvendig fleksibilitet til å dekke de fleste brukstilfeller og baserer sine avgjørelser på blant annet sikkerhetsmerker knyttet til informasjonen. Korrektheten til sikkerhetsmerker er da kritisk for kontrollen av informasjonsflyten mellom sikkerhetsdomener. Det å forsikre seg om at korrekt sikkerhetsmerke blir påført informasjonen i første omgang er et vanskelig problem. Etter påføring kan imidlertid kryptografiske metoder brukes for å sikre autentisiteten av både sikkerhetsmerker og informasjon. I rapporten drøftes ulike tilnærminger for å sikre korrektheten av sikkerhetsgraderingen og andre sikkerhetsattributter som skal inngå i et sikkerhetsmerke. Flere risikoreduserende tiltak som ytterligere kan øke sikkerheten når informasjon flyttes mellom sikkerhetsdomener, blir også identifisert.

Hvilken løsning som bør brukes, avhenger av det gitte scenarioet, det operasjonelle miljøet og kravene til informasjonsutveksling. Hvis det ikke er krav om å utveksle informasjon mellom domener, bør isolasjon av domener fortsatt være utgangspunktet. Dersom det kun kreves enveis informasjonsflyt fra lavt til høyt domene, bør en diodeløsning brukes. Der toveis informasjonsutveksling er nødvendig bør en guard-løsning brukes. Sikkerhetsfiltre kan også brukes for toveis informasjonsutveksling i scenarioer med lav risiko, men en guard gir betydelig bedre beskyttelse mot målrettet skadevare hvis sikkerhetsmerker er korrekte og pålitelige. I mange scenarioer kan det imidlertid være en betydelig utfordring å etablere nødvendig tillit til sikkerhetsmerker. Aksessløsninger som gir brukeren tilgang til informasjon fra flere sikkerhetsdomener, har den fordelen at informasjonen ikke flyttes mellom domenene og derfor ikke trenger å merkes. Det å ha mange terminaler knyttet opp mot flere domener samtidig er imidlertid også en sikkerhetsrisiko, da de kan kompromitteres og brukes til å omgå sikkerhetsmekanismene som skiller domenene. En slik løsning kan være aktuell for noen få brukere, men generelt vil en sentralisert løsning være nødvendig.

Denne rapporten inneholder også to appendiks:Appendiks A gir en oversikt over litteratur og eksisterende guard- og sikkerhetsfilterløsninger samt løsninger for merking av informasjon. Det er mange slike løsninger tilgjengelig, med mye ulik funksjonalitet. De fleste av disse løsningene er basert på bruk av operativsystem med begrenset tillitsnivå (evaluert til EAL 4), noe som indikerer at sertifisering til høyere tillitsnivå ikke er realistisk. Appendiks B gir en oversikt over konsepter, teknologier og produkter som potensielt kan brukes til å bygge løsninger for informasjonsutveksling mellom sikkerhetsdomener. Dette inkluderer oversikt over operativsystem med høyt tillitsnivå, maskinvarebaserte sikkerhetsmekanismer på standard hyllevare samt attributtbasert tilgangskontroll.

# Contents

# 1 Introduction

Network Based Defence or Network Enabled Capability aims to increase the operational capability of war fighters through effective information sharing and timely access to the right information. One of the biggest obstacles to achieve this goal is the lack of coherent and sufficiently secure solutions that allow information sharing across different classification, operational, and coalition boundaries, while still protecting the confidentiality, availability, and integrity of sensitive information and critical services.

Traditionally, the separation of security domains has to a large extent been relied upon to provide this security protection. Partly as an implication of this, single domains are usually not designed to fulfil the stringent security requirements needed to be connected with other domains. Sometimes, it may even be unclear what the exact security requirements would be. In fact, there are many cases where it is probably best to leave the security domains completely separated or only provide for a limited one-way interconnection using a diode. In other situations, the operational value of more effective information sharing may outweigh the disadvantage of the increased security risk. For instance, more effective information sharing may improve mission effectiveness and personnel safety to such an extent that the resulting additional risk would need to be accepted. This is not to say that the additional risk should not be mitigated when possible.

As we will see later in this report, there exist various security solutions and products that may be used to facilitate cross-domain information sharing. These solutions may be well suited to solve specific cases, enabling clear operational value to be gained in the given case. From a more general perspective, however, there are several problems. One is directly related to the solutions and products themselves, in that their applicability is often limited by the assurance of the provided security functionality. That is, there is often insufficient evidence that a solution fulfils the security objectives in order for it to be trusted in a given scenario. In that case, the security solution fails to reduce the security risk to an acceptable level.

Assuming that sufficient assurance is provided, other problems easily occur once solutions are to scale. That is, a solution that is suitable for solving a specific information sharing requirement, between two given domains, may still not be suitable for more general use on a larger scale. In such a situation, issues such as covert channels, cascade effects, intra-domain trust, management, auditing, and information control becomes much more challenging. In particular, security solutions for cross-domain information sharing will often need to be integrated with commodity systems. Achieving this on a larger scale, while preserving the assurance of the overall security solution, is by no means an easy task or something that is solved by a single product. Approaching this problem may require a very conscious strategy, deliberately structuring the infrastructure, the flow and processing of information, and the placement of security mechanisms.

Chapter 2 of this report considers the different high-level approaches to cross-domain information sharing. Chapter 3 provides a more detailed discussion of how to obtain confidence in the information flow control provided by one of the approaches from Chapter 2, namely guards utilizing security labels as basis for release decisions. Finally, in Chapter 4, we provide some conclusions and recommendations. Some possible mitigation strategies to reduce the risks arising when connecting different security domains are also discussed. An overview of the content and organization of these chapters is given in Figure 1.1.

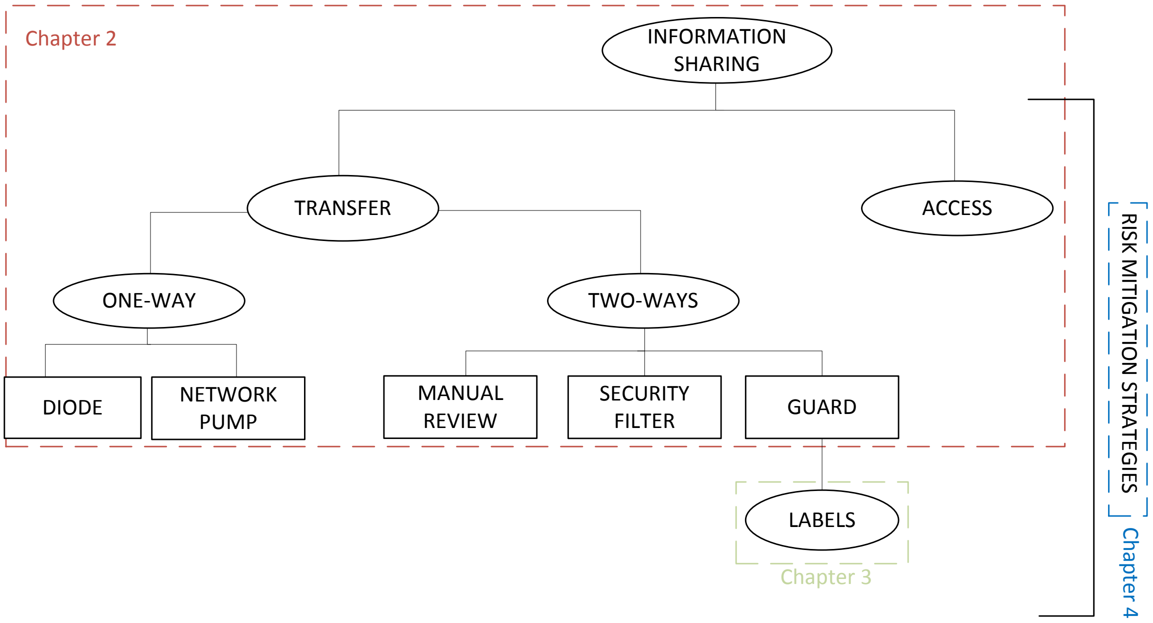

*Figure 1.1 Overview of the solutions considered in this report.*

In addition, a more detailed survey of literature and existing products with regard to guards, security filters, and labelling is provided in Appendix A. Furthermore, Appendix B provides an overview of high assurance operating systems, hardware technologies, and concepts that can be used as building blocks to enable cross-domain information sharing.

## 1.1 Problem definition

When considering the problem of sharing information across security domains, there are basically two possible approaches. One is to physically transfer information from one security domain to another. The other is to allow users to access the information directly in different security domains, without necessarily moving the information between domains. In the first case we speak about *transfer* solutions, while the latter is referred to as *access* solutions.

It is for the sake of the discussions in this document assumed that a domain is required to protect itself and its information, according to requirements for self-protection. We will therefore refer to the domain to be protected as the *High domain*, while the *Low domain* is generally not trusted from the perspective of the High domain.

In the case of transfer solutions the problem is to prevent information from leaking from the high domain to the low domain, while releasing information that is allowed by policy to be shared with the lower domain. Additionally, malicious code and other unwanted data must be prevented from entering the high domain. Typically, the devices implementing this solution are administered by the same authorities controlling the high domain that is to be protected. Consequently, there may be two transfer solutions for a single interconnection, one in each direction, each protecting its respective “high” domain.

Access solutions address situations where it is desirable to be able to have access to information from different domains at the same time, preferably on the same machine or screen (e.g., in order to minimize the equipment needed or to improve usability). In this case the challenge is to maintain physical or logical separation so that no information can leak between the different domains.

These two approaches may be seen as complementary to each other, and might be combined to achieve a better user experience and increased assurance. Depending on the specific scenario, they potentially have different implications with regard to security, user experience, flexibility, scalability, and requirements for system integration and infrastructure support.

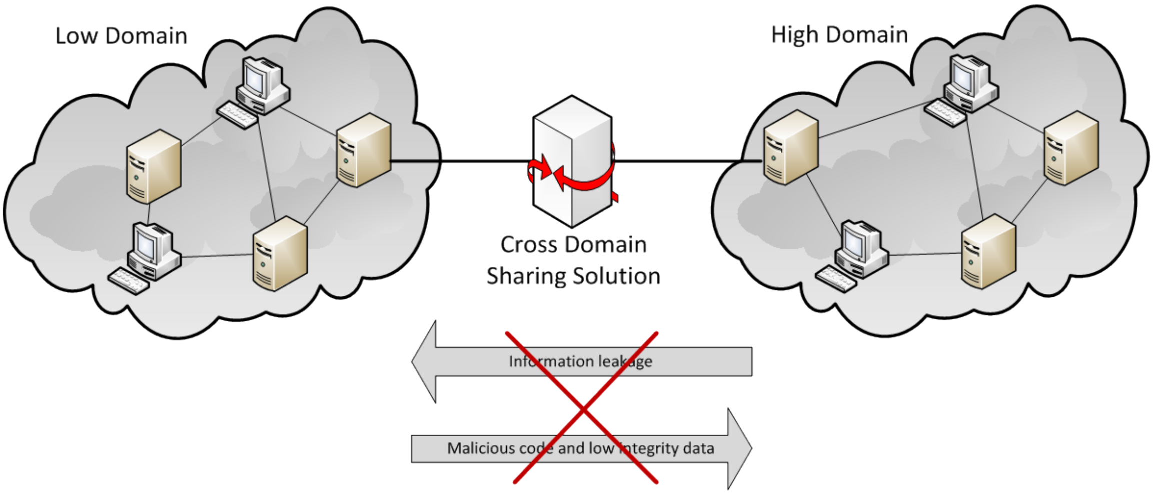
Low Domain
High Domain
Cross Domain Sharing Solution
Information leakage
Malicious code and low integrity data

*Figure 1.2 Generic challenges to be solved by a cross domain information sharing solution*

## 1.2 Threats

The most significant threat when allowing information to flow between different security domains is that unintended data flow can cause some sort of harm. This can either be information leakage from the high to the low domain, resulting in loss of confidentiality, or data flow from the low to the high domain that causes loss of integrity or availability.

The approaches presented in the next chapter all try to address these threats. However, there is no approach that can satisfy all operational needs for cross-domain information exchange without also introducing significant risks, and each approach might be more suitable for some scenarios rather than others. Such potential risks, which we will refer to as *model-specific* threats, will be discussed for each model where appropriate.

In general, the confidentiality, integrity, and availability of data can also be exposed to threats that arise as a consequence of the concrete implementation and deployment of such solutions, rather than problems in the models themselves.

These may include:

- Coding errors and bugs
- Wrong physical set-up
- Configuration errors
- Human errors
- Abuse of privileges
- Other possible vulnerabilities that can be exploited to bypass or deactivate the security mechanisms in place

Although some arguments can be made about how likely each model might be to incur in this type of more generic threats once it is implemented, a real vulnerability analysis can only be done on a concrete implementation. A typical example is covert channels. Covert channels are a path of information flow that is not meant as a communication channel and are almost impossible to completely eliminate in many types of systems. Examples are to modulate information into the timings of events observable to a receiver or by varying the size of the messages being sent between the domains.

# 2 Approaches to cross-domain information sharing

Isolated security domains are advantageous both in terms of confining sensitive information and in providing integrity and availability protection. True isolation is however in most cases not practically possible, and various means for cross-domain information sharing are therefore utilized. At the most restricted, movable media is used to move data between the domains.

We will in this chapter provide an overview of different approaches to provide more automated cross-domain information sharing. Chapter 2.1 takes a look at diodes and network pumps, which are typically used to enable unidirectional information flow from low to high. For two-way information exchange, we first discuss the use of manual review and release in Chapter 2.2. The use of guards and security filters are then discussed in Chapter 2.3. All these approaches are examples of transfer solutions, where data objects are moved or copied from one security domain to another. The alternative approach of using an access solution, providing the user access to multiple security domains simultaneously instead of transferring the information between the domains, is discussed in Chapter 2.4. In Chapter 2.5 we discuss the role of MLS and the Bell-La Padula model in this context, before a summarizing discussion is provided in Chapter 2.6.

## 2.1 Diodes and Network Pumps

The concept of a one-way diode, also known as a data or information diode, is to allow data to flow only one way from a transmitter to a receiver. A diode is typically implemented as a hardware device, e.g., a physical connection having only a sending module at one side and a receiving module at the other side thereby making it physically impossible to send any information in the reverse direction.

A diode is typically used as a mechanism to enable automatic, or semi-automatic, information transfer from a lower classified domain to a higher classified domain, as shown in Figure 2.1. As a diode only enables information to flow one way, it is ensured that no information can leak through the diode from the high domain to the low domain. If confidentiality is not the main concern, a diode may alternatively be used to protect the sending domain from being affected (e.g., in terms of integrity or availability) by the receiving domain.

Diode systems are used with supporting software that handles the actual move and control of data across the diode, and this may also include functionality for traffic monitoring and detection of malicious code.

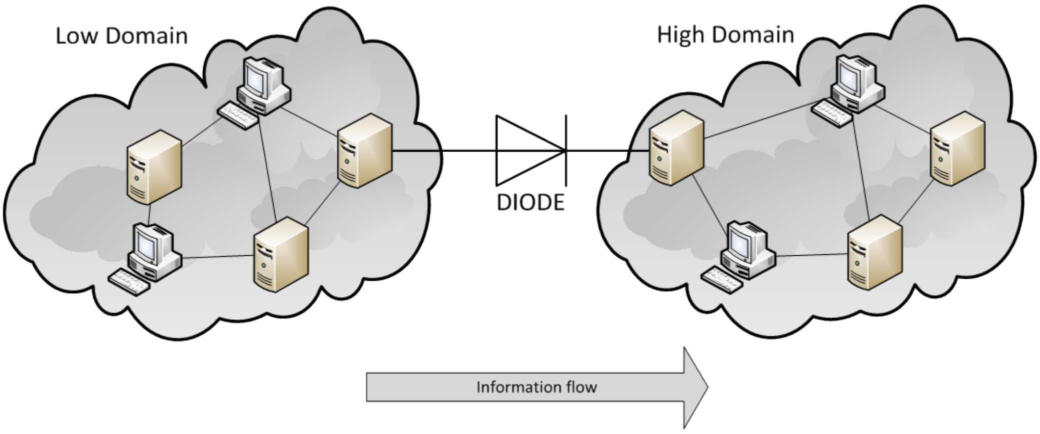

*Figure 2.1 Enabling one-way information flow from low to high using a diode*

A concept closely related to diodes is that of network pumps [1]. A network pump differs from a diode by allowing acknowledgements to be sent in the reverse direction, thereby enabling end-to-end reliability. In order to limit the covert channels that would result from having acknowledgements flow from high to low, the network pump buffers the packets from the sender (low) while sending the packets to the receiver (high). The acknowledgements from the receiver are received by the network pump, which again sends the acknowledgements to the sender (low) with randomized timing.[1] Still, a system with a network pump offers limited flexibility in that it only allows receipt acknowledgements to be sent from high to low.

[1] Although this significantly reduces the bandwidth of the covert channel, there is for instance still a covert channel in the acceptance or rejection of a connection by a receiver that can be used to communicate one bit per connection [19]. The capacity of this covert channel can be restricted by enforcing a limit on the connection frequency.

While diodes are available from several vendors and are relatively commonly used to enable one-way information flow between two security domains, networks pumps are less common.  There is however a hardware version of the network pump which has been used in the U.S. Navy and other U.S. government facilities [2].

## 2.2 Manual review and release

While diodes and network pumps can provide an efficient solution to move data from a lower classified to a higher classified domain, they do not provide any means for sending information in the opposite direction.

The most immediate solution to control the flow of data from a high to a low domain is to put one or more operators in charge of reviewing the data before it is allowed to leave the high domain. The operators are trusted to enforce the security policies by inspecting all data, removing or obscuring the sensitive parts, and releasing only the portions that are not in violation of the policy, if any. In the case of paper documents the practice is well established, but with electronic data there are different challenges to consider.

From a practical point of view, although a person may be able to evaluate the content of a document better than a machine, human verification is error prone and scales badly. A DVD or memory stick can easily contain the equivalent of an entire encyclopaedia and electronic data can come in many different forms (e.g., hidden meta-data or binary files) that are not always amenable to human verification [3]. Specialized tools can alleviate some of these problems, but not completely solve them.

In its most rudimentary form the user may have the responsibility to perform the “review and release” when moving a file from one workstation to another over an air-gap. For strengthened security a dedicated operator terminal (or similar) would be used. From a technical perspective, mechanisms should be in place to limit the chance of data leakage from the operator terminal and to verify that the data submitted for review and release is authorized to be processed. Such mechanisms may differ based on how data enter and leave the operator terminal, but it is in any case critical that the terminal’s integrity is preserved.

An isolated and dedicated terminal where the data is moved to and from the terminal using movable storage devices may be easier to protect, given that the removable devices are handled properly. A terminal connected only to the high domain may result in more efficient data processing as the documents to be declassified can be sent directly over the network, but it can potentially also give access to sensitive material stored in the domain. If the terminal is connected also to the low domain, then a specialized device is needed to guarantee that only data that has been reviewed and authorized for release can be transferred to the low domain. Such high-assurance devices are discussed in the next chapter. Manual review can also be one of the criteria used to authorize data release as proposed by the NATO Information Clearing House concept and its related Release Authorities [4], as well as for the purpose of security labelling.

## 2.3 Guards, security filters, and IEGs

Another approach to cross-domain information exchange is the use of a guard. A guard is a type of gateway or proxy accepting or rejecting a data object in order to enforce information flow control between the security domains, according to some specified security policy. A guard is typically used to provide two-way information flow, as illustrated in Figure 2.2, but may also be used to provide one-way information flow given such a policy.

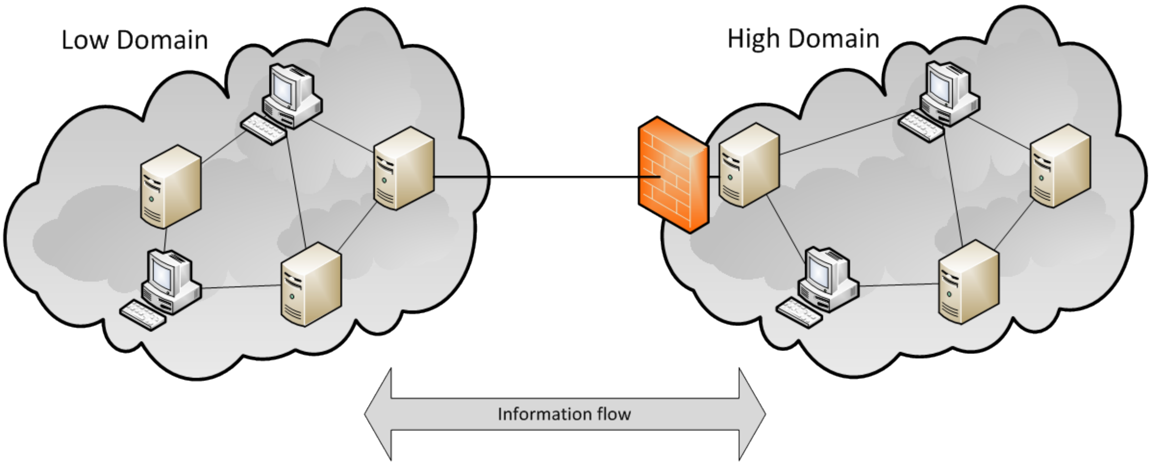

*Figure 2.2 Enabling two-way information flow using a guard or security filter.*

Various guards differ in their basis for determining whether an object is releasable according to the applicable security policy or not. One alternative is to only allow packets or data objects with a more or less strictly defined format, possibly with specified permissible values for each field. Although this is sometimes referred to as a guard, we will more precisely refer to this as a security filter. Security filters may operate on IP/UDP/TCP packets, or operate similar to an application level firewall (e.g., filtering XML messages based on XML schemas).

Another type of guard may base release decisions on specific properties (e.g., the confidentiality classification) of the data, where the properties are specified within a security label associated with the data. In this report, the term guard is used to refer to this type of label-based guard. Depending on the trustworthiness of the security labels, label based guards may provide stronger security than security filters and may also include the functionality of a security filter.

Guards and security filters may also make use of content checkers such as dirty word lists, checking for words that are not allowed, or more advanced content analysis. It should be noted that in the case of security filters and content checking mechanisms, any trusted high side entity with knowledge of the acceptable content/formats may be able to deceive the security control. Thus, such solutions may provide limited protection against malicious insiders or targeted malware within the high domain. Still, security filters may be effective in preventing data loss due to mistakes and common malware.

This problem can to some extent be mitigated by guards or security filters authenticating senders and controlling their authorization for transferring data. In this case, entities may potentially have different authorizations with regard to what type of data is allowed to be transferred and also to which recipients. Such functionality could alternatively be provided by some other external component in conjunction with the guard or security filter.

A guard or security filter may in principle operate as a stand-alone component. For larger interconnections, however, the guard or security filter may typically be part of some sort of information exchange gateway, as specified by NATO and national equivalents. An Information Exchange Gateway (IEG) is a NATO concept intended to provide security architectures and standardized interfaces in order to enhance interoperability between domains, where at least one is a NATO domain. While an IEG not necessarily contains a guard or security filter, guards are expected to be utilized to provide the required assurance in the information flow control for the more demanding IEG-scenarios. An IEG may also contain additional security functionality, such as border protection, virus scanning, intrusion detection, and application proxies. While the IEG may typically be implemented as a protected network segment similar to a demilitarized zone (DMZ), it can also be envisioned having all the IEG functionality implemented within a single device.

A survey of existing guard, security filter, and labelling products is provided in Appendix A, together with an overview of related work in the literature. The survey shows that there are quite a few guard and security filter products available, providing support for many data types and applications. Most of the products are however based on operating systems providing limited assurance (i.e., evaluated at Common Criteria EAL4), indicating that it would be difficult to evaluate these solutions at higher assurance levels. Most existing labelling solutions also provide limited assurance being applications or plug-ins running on top of commodity operating systems. Higher assurance labelling may be provided by guards functioning as labelling gateways. An example of this is the recent prototype guard (designed to pass EAL5 evaluation) for use in service-oriented architectures [5], developed as part of a CD&E activity in collaboration between FFI and Thales Norway.

## 2.4 Access solutions

An alternative or complementing approach to actually transferring information between security domains is to enable users to access information and services in multiple security domains. This has traditionally been achieved through the swivel chair approach, where the user has multiple workstations each connected to a different security domain. To reduce equipment, these workstations are often connected to the same keyboard, mouse, and monitor through a keyboard, video, and mouse switch. However, due to for instance space, weight, and power constraints, the swivel chair approach is not a usable solution in all cases. Even for usage where the swivel chair approach is applicable, it may be advantageous to be able to access multiple security domains from a more integrated user system, e.g., in order to provide a better overview to the user or to save space/weight. No matter the motivation for realizing such a solution, it is critical that information does not leak between the security domains. Thus, strong separation is necessary.

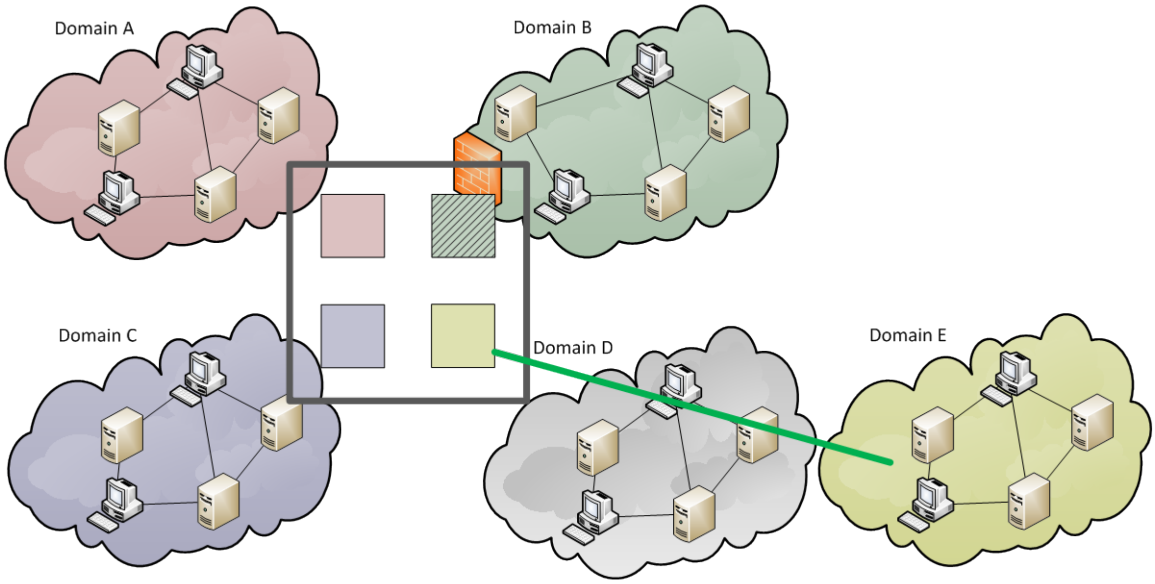

*Figure 2.3 Illustration of (user) system with access to multiple security domains*

An illustration of a system with access to multiple security domains is shown in Figure 2.3. The system illustrated by the large square box is connected directly to domains A, B, and C, as a traditional user system within these domains. In addition the system also has a logical connection with domain E through a secure channel. As illustrated by the connection to domain B, the system's access to a security domain may be limited by some type of security gateway to increase the security of the multi-domain solution. An example of such a limited connection could be a case where the access solution is only allowed to connect to a remote desktop server within the connected domain or where the system is only allowed to receive messages of a given format. Although the access solution provides access to domains A, B, C, and E, a pure access solution does not provide any information flow between the domains.

Access solutions can be realized in various ways. One option is to integrate multiple computers as a single device, providing sufficient electromagnetic shielding between each sub-computer. Such solutions have been previously available and certified according to the Common Criteria [6], and there is also a related U.S. patent [7]. With the increasing miniaturization of computer hardware, it should be feasible to realize such systems even in smaller form factors.

An alternative to providing physical separation through separate hardware is to provide logical separation. Using a MILS[2] separation kernel, strong separation can be provided between multiple partitions, each for instance containing avirtual machines running a commodity operating system. As explored in a previous FFI-report [8], a separation kernel can be used as a basis for implementing a workstation or portable device capable of handling multiple security domains. Currently, there are several commercial/government solutions providing logical separation of security domains for end-user systems [9] [10] [11] [12].

[2] MILS is originally an acronym for Multiple Independent Levels of Security, but is now often (better) used as a proper noun.

Using such products, it is typically possible to also allow for some information exchange between security levels, according to configuration. Thus, while an access solution in its pure form does not facilitate any information transfer between the security domains, there may also be mechanisms for providing some limited information exchange between security domains. Even without allowing for information transfer between the domains, information may in principle be aggregated within a high partition of the access solution, thereby enabling a coherent presentation of the information to the user as well as automatic processing on the full information set.

While logical separation in principle also can be provided by more traditional desktop virtualization products, the security assurance provided by such products is generally insufficient for use in scenarios with a high security risk as considered in this report.

A third option is to enable access to different security domains through remote desktop solutions. For lower risk interconnections this may be provided using standard solutions in combination with the appropriate security measures.

To reduce security risk, authorized users of an access solution should have clearance to access all the connected security domains of that system. Thus, access solutions typically do not support multi-level security in the sense of enforcing appropriate access to users with different security clearances.

## 2.5 Multilevel Security (MLS)

The term multilevel security can have slightly different meanings according to the context in which it is used. Traditionally it has been used to refer to a system operating in multilevel security mode, i.e., where not all users have the required clearance to access all the information. However, the term is also used to refer to a system capable of handling information of different classifications according to policy, such as guards. When using the term multilevel security in this document, we refer to the former usage unless stated otherwise.

The most well-known theoretical model for systems handling information classified at different levels is the Bell-La Padula model, and systems based on variations of it, are often referred to as MLS systems.  The model uses the notion of objects and subjects that are assigned static security levels. Objects are typically information or resources while subjects can be users or processes. A system is guaranteed to remain in a secure state if two rules are fulfilled: No subject can read an object of higher security level than itself, and no subject can write to an object of lower security level than itself.[3]

While the Bell-La Padula model is well understood, it presents challenges in practice. The no-write-down rule is very restrictive and often leads to an over-classification of information, since

[3] Bell-La Padula also has a notion of discretional access control through an access control matrix which can be used to implement need-to-know based access.

information can flow only toward higher classifications. Often this is circumvented by some notion of "secure process" that can downgrade information, hereby destroying the simplicity of the security arguments.

MLS systems are often said to be inherently difficult to realize. Starting in the 1970s, significant effort has been spent on developing MLS operating systems. Several of these efforts have been successful in the sense that certified high assurance systems with strong security mechanisms have been realized, e.g., GEMSOS and STOP as discussed in Appendix B. Still, there is no commodity general purpose operating system supporting MLS, and such a system cannot be expected to become available in the foreseeable future.  Thus, while existing MLS operating systems may provide a basis for implementing security solutions such as guards, MLS operating systems as such are currently not a viable strategy for information sharing.

## 2.6 Discussion

There are strong security arguments for keeping security domains isolated, thus, security domains should not be interconnected by default. When security domains are to be interconnected, however, the presented approaches may support different information exchange requirements and result in different security implications.

If the required information exchange is mainly from Low to High, the traditional one-way diode solution provides a preferable solution by guaranteeing that there is no information flow from High to Low. Of course, appropriate security controls like virus checking and access control must still be applied to the incoming information flows. Combined with suitable application proxies this approach can accommodate a quite wide range of application requirements as long as there is no need for information flow from High to Low. Such applications could include importing weather forecast information or  updated antivirus signatures as well as more integrated solutions where for instance information from Low sensors are utilized by a High system.

When information is to flow from High to Low, manual intervention is often employed to overcome the downgrading problem. This is typically performed by either transferring files using movable media or utilizing a more special purpose downgrade solution, where manual review may be part of the process. However, manual review does not scale with regard to message frequency and size, and there are limitations as to what types of data can be effectively reviewed by a human. As a result, human review is generally not well suited for machine-to-machine communication.

Guards introduce the possibility for automatic two-way information exchange between security domains, and can also be utilized to improve the security of existing two-way interconnections. In particular, label-based guards have the potential to provide stronger security than what can be achieved using security-filters alone. This additional security comes at the cost that security labelling has to be integrated and performed in such a manner that the information within the labels can be trusted by the guard, which is the topic of the next chapter. Thus, due to the added complexity of label-based guards, security filters may be preferable in some scenarios. This could for instance be the case for lower risk interconnections at the tactical level, where the additional bandwidth overhead of security labels may be undesirable and where labelling may be difficult to integrate effectively. Nevertheless, it must be taken into consideration that a security filter can be easily bypassed by malware executing on the High side, provided the acceptable format/content is known and the malware  is able to transmit through the security filter. Thus, for general use, it is advisable that information flow control is enforced by guards which base their decisions on security labels in addition to the use of security filter functionalities.

Access solutions provide an alternative to transfer solutions in some situations, by providing the user with access to services and information within different domains, without releasing data objects between the domains. This limits the flow of information between security domains, which is advantageous from a security perspective. There are some concerns if there are many multi-domain capable user devices connected directly to multiple domains. A high number of such devices would result in the same number of potential interconnection points between security domains, as the compromise of any one of these devices could potentially break the separation between the domains. Thus, it potentially imposes significant operational audit requirements and demands for high assurance solutions relative to the security span.

A potential mitigation for the previous scalability issue is to centralize the interconnection points, e.g., having a centralized remote desktop server or centralized network interconnections providing access to the different domains. The user devices would still be required to provide separation between desktops from different security domains, but the server and/or interconnection point would provide an audit and enforcement point where the consequences of a compromised user machine could be more easily mitigated.

It may also be noted that while access solutions may provide flexibility for human users, this approach is generally less applicable for use in machine-to-machine communication, as the application in that case would need to be capable of handling information from multiple security domains. Thus, transfer solutions are generally more suitable for machine-to-machine communication.

# 3 Confidence in the correctness of security labels

Based on the discussion in Chapter 2.6, we consider a guard as the most generally applicable solution when it comes to two-way transfer of data objects between security domains. However, although the use of security labels can potentially provide stronger security and greater flexibility than other solutions, it presents also additional challenges. Namely that the information carried by these labels must be trusted to be correct, since they are part of the basis for the guard’s release decisions.

As shown in Figure 3.1, threats to the correctness and authenticity of labels arise long before they are processed by a guard.  Thus, even assuming that the guard can be trusted to operate correctly and that covert channels have had their capacity limited to an acceptable level, there are still other problems specific to the labelling process that need to be addressed. The purpose of this chapter is therefore to analyse various aspects of this process in order to identify where confidence in the correctness and authenticity of labels is established.

We first define more precisely what a label is in this context and the different phases of a typical labelling process. Security labels are often associated only with the confidentiality level, i.e., classification, of an object.  Here we consider a more general concept where a label is a separate object consisting of a collection of metadata which is then somehow associated to the data object it refers to [13]. This metadata is possibly in the form of attribute and value pairs and is used to determine how a data object should be handled to preserve its security [14].

More formally, the process of creating a label and binding it to a data object can be divided quite naturally into three phases [15] as illustrated in Figure 3.1:

1. Selection of metadata to be put in the label
2. Creation of the security label
3. Binding of the label to the data object

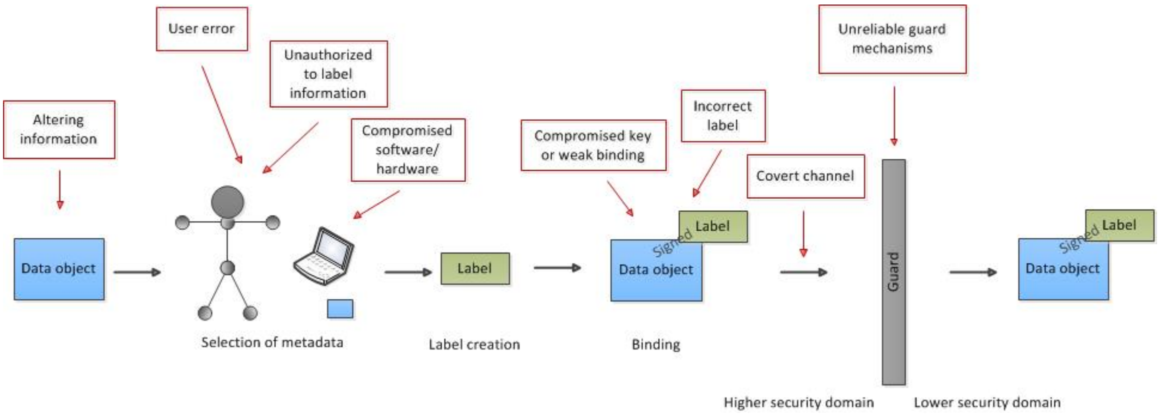

*Figure 3.1 Illustration of threats related to the use of a guard. It should be noted that in many scenarios there may not be a human user involved.*

The environment where these steps take place and the entities executing them, all play a role in defining the level of confidence we can have in the labels that are generated. Higher platform assurance can give some guarantee that the process is executed as intended. For instance, a special purpose labelling device may be able to provide higher confidence in the correctness of security labels than labelling performed within a user application running on a commodity operating system. This because, for instance, malware may not easily change the labels or abuse the keys used for the binding.

At the same time, if an authorized entity erroneously labels classified information as unclassified, protecting the authenticity of this security label is insufficient to establish confidence in the label correctness. In this case, labelling performed within an unclassified environment may provide higher confidence that the information labelled is indeed unclassified than labelling of unclassified data within a classified environment.

Each phase will now be discussed in detail in a dedicated section.

## 3.1 Selection of metadata

Selecting the right metadata is important because if the data object is labelled with a too permissive security label, this would constitute a security violation as it could enable the data object to be released to domains not authorized to receive the data. Likewise, a too restrictive security label is also undesirable as this may compromise the availability of the information to authorized recipients. It may also be noted that errors in the syntax/formatting of the security label would generally not constitute a risk to confidentiality as such a label is to be rejected by the guard (or other security control).

The selection process is going to take place after the creation of the data object. So the first problem to consider is how to make sure that the integrity of the data object is preserved until the label is created and attached to it, so that the label still describes the object correctly. However, we can simply assume that the metadata selection will start only once the data object is on a platform evaluated at a sufficiently high assurance level, and that decisions will be based on that version of the object, no matter how it may have been modified before arriving there. The platform can be a commodity computer operated by the same user that created or received the object or a dedicated high-assurance platform running a labelling service. Such a service could also either choose the metadata autonomously or evaluate a labelling request already containing the chosen metadata.  Each can be a reasonable solution in different scenarios.

Concerning the approaches that can be used for choosing the correct values to be put in the label, we identify two main ones: one based on the content of the object and one based the source(s) (e.g., environment) from which it originates.

### 3.1.1 Content based classification

This approach is based on determining the attributes (e.g., confidentiality classification) of a data object by analysing its content. In the case of a data object, it could be the creator (i.e., a human author or application) making this decision based on knowledge of the data object itself and the applicable labelling policies. Alternatively, another person may review the data object (e.g., document) in order to decide its classification or to verify that the claimed classification is indeed correct.

As discussed in Chapter 2.2, the use of human review is not practical in many cases, e.g., due to the nature or amount of data. Thus, in many cases the content analysis would need to be performed automatically. The exact use of a content checker for this purpose depends to a large extent on the context. That is, the content checker would need to be carefully tailored to the context in which it is used in order to precisely be able to classify data objects. For instance, in the case of a dirty word checker the list of dirty words would need to be tailored to topics considered sensitive within that community of interest, and/or the specific application that generated the data object, or the list would likely be either too strict or too loose. Likewise, a content checker based on checking the format and values of messages should also be adapted to the specifics of the data objects used in a given environment to provide the highest confidence.

In the case of a human review it must be expected that the quality of the review may vary greatly depending both on the specific reviewer and her familiarity with the topic area of the material being reviewed, as well as the type and amount of material to be reviewed. Likewise, the actual assurance provided by automatic content checkers will vary greatly depending on the scenario. A generic content checker may have limited ability to detect malicious data exfiltration/downgrading attempts in general scenarios.

### 3.1.2 Origin based classification

Origin based classification is based on knowledge about the origin(s) of the data within the data object. The origin(s) can in principle be any type of source (e.g., security domain, another data object, computer, or virtual machine) which could imply that specific attributes should be set in the security label. For instance, a data object may be classified as restricted because it originates from a restricted domain. Likewise, a subset of a restricted data object (e.g., document) may be labelled as restricted without risk of leaking data. It may be noted that origin based classification provides best accuracy when dealing with single level origins (as in single level mode of operation). Labelling all data objects originating from a system high Secret domain as Secret would likely result in too high classification of many data objects.

## 3.2 Creation of the label

Once the correct metadata has been selected, it is possible to create the label object. This phase of the labelling is relatively simple as long as it takes place within the same trustworthy environment as the selection, where also the data object integrity can be guaranteed. If selection and creation take place on different platforms, then it is important to establish a secure connection to transfer the metadata and possibly the object without compromising their integrity. A further verification of the metadata proposed for the label may also be performed by the agent in charge of creating the label if the confidence in the platform or entity that chose the metadata is deemed insufficient.

Besides that, the actual format of the label is an important aspect of the labelling question. Although internally in one domain any arbitrary format may be used, a standard for how a label should be constructed is a clear advantage to facilitate interoperability and cooperation in a cross-domain environment. Appendix A.2.1 reviews some existing standards.

## 3.3 Binding

The binding itself is usually achieved either by storing label and data together inside the secure boundary of the system or through some binding whose strength can vary from that of a digital signature to a simple reference or checksum [16]. However, simply storing data object and label together in a protected location, as many databases do, is not a viable solution for guard-based solutions. The binding is lost when label and data object are moved outside the secure boundary, and a guard has no means to verify that the binding between the received objects has been preserved. Thus, we assume here that an explicit binding is used and that the "binding data" (e.g. the signature) is stored as one of the metadata in the label.

As in the case of the label generation we still need to assume that both the integrity of the data object and the label has been preserved throughout the whole process. So either the binding takes places on the same trusted platform of were the label was generated, or both data object and label must be securely be transferred to a separate trusted device specifically designed to offer a binding service. A NATO standard for a service that binds metadata to objects has been defined in [17], although it has been kept very generic in order to be able to accommodate different operational requirements and standards. For instance, a strong cryptographic binding like a signature may not be required.

More concrete problems related to the binding process are the correct implementation of the cryptographic services employed, in conjunction with a secure key management. Typically a separate key management (e.g., PKI) is used for each security domain, re-signing (or re-labelling) each data object being received from other domains. This has the advantage of avoiding the burden of cross-domain key management and avoids placing excessive trust in foreign domains. It would be problematic to trust security labels signed by another domain to make release decisions, as a signing key from e.g., a lower classified domain could then be used to leak information from the higher domain. Re-signing/re-labelling avoids this problem, at the cost that if end-to-end message authentication is required this will need to be provided separately. Nevertheless, secure key management is essential to the security of label based information release.

## 3.4 Discussion

Summarizing, we can say that the confidence in the correctness of the labels depends on the correct choice of metadata, i.e., a correct classification, on a sufficiently trustworthy platform and the authenticity of the label and data object. While trustworthy binding and label generation may be achieved using available technology, the problem remains of a correct classification of data objects. On one hand automatic classification is preferable as it can be performed efficiently and in high-assurance devices, but it can guarantee correct labelling only in some specific situations since it does not allow for generic and reliable content analysis. On the other hand, manual classification is still needed for content-based classification of human-generated data, especially in the case of declassification, but it is inefficient and it cannot be performed in a distributed manner as long as commodity PCs are not replaced with more trustworthy devices.

In order to achieve an effective and secure information flow in a cross-domain scenario, reliable automatic classification and labelling is needed as machine-to-machine communication will stand for a large fraction the data traffic, while human-generated data could be manually classified by individual users when needed given the right equipment.

In the meanwhile, high confidence in security labels may be achieved by limiting the scope of the labels and placing a dedicated high-assurance labelling device in front of an environmentand let it classify data objects based on their origin. Figure 3.2 shows the conceptual process when a data object is labelled and later released through a guard or other release mechanisms, including the potential incoming information flows. Although environment A and B may typically represent different networks and security domains, these environments could for instance also represent separate computers or be logically separated by a separation kernel running on a single machine.

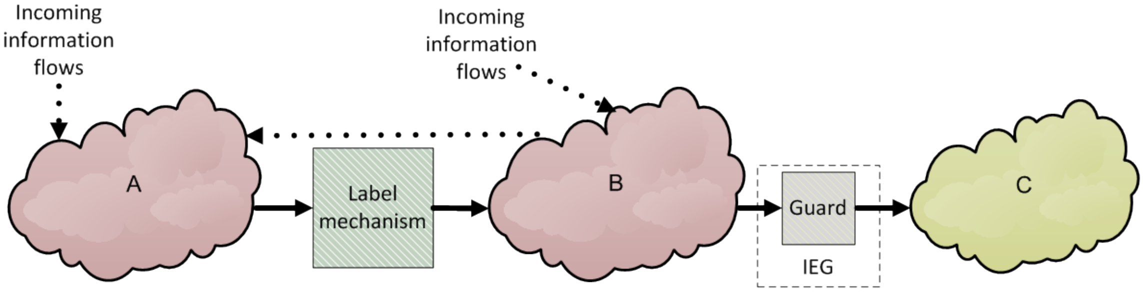

*Figure 3.2 Information flows in data labelling and release*

In this case, a data object that is created in environment A will have a classification no higher than that of environment A (and none of the incoming information flows to A will be classified higher than A). Thus, if the confidence in the correctness of the classification is to be based on potential information flow, all data originating from A is to be labelled with the classification of environment A. If environment B has a higher classification than A, there will be no information flow from B to A (unless mediated by a guard or similar mechanism which will likely introduce covert channels), and the security label will maintain confidence in the classification of the data object as it travels through B. When using origin as basis for determining the classification of a data object, labelling should be performed before any unnecessary information flows may affect the classification of the data object. Thus, the potential information flows and needs for labelling may need to be considered when designing the system and network architectures.

Origin based labelling is available in some guards, so this approach could be realized in practice, but provides limited flexibility at least when applied at the security domain granularity. For instance, it would not enable information flow from a domain with high classification to domains with lower classification. Hence, it would be desirable to be to be able to perform also content based classification in such stand-alone devices that can be built with higher assurance. A framework for policy-based labelling also supporting content based classification has recently been proposed in [18]. While such a framework may be able to perform automatic classification of data objects in some scenarios, another and perhaps more likely application would be to increase the confidence in the labels proposed by for instance the author of a data object. Also, security filters could provide some simple form for content checking that may be employed in specific situations.

# 4 Conclusions

The conclusion we can draw from the discussions in Chapter 2 and 3 is that different cross-domain solutions may fit the specific information exchange requirements of different scenarios and operational environments. However, given the intrinsic risk for information leakage of these solutions, isolation should still be the default.

In general we can say that where a one-way information flow (from Low to High) is sufficient, a one-way diode should still be preferred, potentially in combination with a security filter for increased security. For two-way information flow, a label-based guard is preferable for general use, although the use of a simpler security filter may be justified in lower risk scenarios. Provided that security labels can be trusted to be correct, a guard can provide significantly better protection against data exfiltration from the inside, such as targeted malware being present on the High side or a human insider, than a simpler security filter alone.

Whereas sufficient confidence in the correctness of security labels may not be achievable in all scenarios, we have identified content and origin based classification, together with authenticity, as bases for confidence in security labels. Application systems, system architecture, and labelling solutions should be designed to enable these bases of confidence to be utilized.

Access solutions providing a user access to multiple security domains from a single machine has the advantage of limiting actual information transfer between security domains and don’t require labelling of information. However, a centralized approach seems required in the case of many users, as user devices connected directly to multiple security domains could potentially shortcut the security domains if compromised. For a more select group however direct connection to multiple domains could be an alternative.

In the way forward, apart from realizing high assurance solutions such as guards to fulfil the applicable information exchange requirements, one should strive to improve the confidence in the correctness of security labels as this would have a significant impact on overall security. To further reduce the security risk of cross-domain information sharing, more customised mitigation strategies should also be investigated. Some are mentioned in the next section.

## 4.1 Risk mitigation strategies

Information exchange between security domains will weaken the domain isolation, and thus mitigation approaches that may reduce the security risk of cross-domain information exchange, should be considered. Some of these measures may for instance be taken within the cross-domain solution itself, whereas some would need to be implemented outside the solution to be effective. Some possible approaches are explored here.

### 4.1.1 Content checkers

There may be several types of content checkers such as antivirus, dirty word, and format checkers, serving different security objectives. For instance, content checking may be performed

within the guard before releasing data in order to provide an additional verification that the security label of the data object is indeed correct, i.e., that the data object does not contain non-releasable information.  At the same time, a content checker could also be used in conjunction with a diode to prevent malicious data to enter the high domain.
However, simple content checkers may not be able to prevent more targeted attacks.

### 4.1.2 Monitoring and detection

Detection may be a typical function within a cross-domain solution, especially when two-ways information flow is allowed. Monitoring of abnormal traffic patterns or atypical protocol usage can detect intrusion attempts towards the high side network and provide indication of data exfiltration. Additional information, such as the history of the data objects, may give a better basis for the detection of data leakage.

### 4.1.3 Restrictions on senders and/or label originators

Restrictions on what entities in the high domain can send to other domains can be used to reduce the number of trusted entities that would be able to leak out information, either directly or by signalling over a covert channel.  In the case of guards, this can be extended also to the entities that are able to label data, and possibly misuse the labelling. A separation of duty may also be enforced between those who are allowed to label data and those who can initiate transfers to other domains. This may to some extent limit the potential damage that can be caused by a single entity.

### 4.1.4 Improved intra-domain access control

Today, confidentiality, integrity, and availability are to a large extent ensured by the controlled boundary of each security domain, rather than stringent (access) control within each domain. A significant disadvantage of such an approach is that one becomes very vulnerable in the case of a breached security domain boundary or in the presence of a malicious insider, either human or software. Since the risk of such a breach can considerably increase if the domain boundaries are weakened by a cross-domain solution, need-to-know based access control within the high domain can be efficient to reduce the consequences in the case where a trusted entity is compromised

# 5 Bibliography

[1] M. H. Kang, I. S. Moskowitz and D. C. Lee, "A network pump," *IEEE Transactions on Software Engineering,* vol. 22, no. 5, pp. 329-338, 1996.

[2] M. H. Kang, "The Pump: A Decade of Covert Fun," in *Proc. Annual Computer Security Applications Conference*, 2005.

[3] D. Knott, "A Case for Forensics Tools in Cross-Domain Data Transfers," 2002. [Online]. Available: https://www.sans.org/reading-room/whitepapers/forensics/case-forensics-tools-cross-domain-data-transfers-1126. [Accessed 1 September 2014].

[4] A. Domingo and H. Wietgrefe, "On the federation of information in coalition operations: building single information domains out of multiple security domains," in *Proc. IEEE Military Communications Conference*, 2013.

[5] R. Haakseth, N. A. Nordbotten, Ø. Jonsson and B. Kristiansen, "A High Assurance Cross-Domain Guard for Use in Service-Oriented Architectures," in *Proc. International Conference on Military Communications and Information Systems (to appear)*, 2015.

[6] T. Gjertsen and N. A. Nordbotten, "Military operational systems in field - multiple levels of security," FFI-rapport 2009/01137, 2009.

[7] P. Shiakallis, E. Harvey, J. McGinn and G. Purser, "Multi-domain secure computer system". U.S. Patent 8,646,108, 4 February 2014.

[8] N. A. Nordbotten and T. Gjertsen, "Towards a certifiable MILS based workstation," FFI-rapport 2012/00049, 2012.

[9] Bertin Technologies, "PolyXene: the high-security software platform," [Online]. Available: http://www.bertin-technologies.com/polyxene-high-security-operating-system.aspx. [Accessed 10 October 2014].

[10] Secunet, "SINA Workstation (SINA Virtual Workstation)," [Online]. Available: http://www.secunet.com/en/topics-solutions/high-security/sina/sina-workstation/. [Accessed 10 October 2014].

[11] General Dynamics, "TVE Trusted Multilevel Computing Solution," [Online]. Available: http://www.gdc4s.com/tve.html. [Accessed 10 October 2014].

[12] General Dynamics, "TACLANE-MultiBook," [Online]. Available: http://www.gdc4s.com/taclane-multibook.html. [Accessed 10 October 2014].

[13] A. Magar, "CR 2005-166 - Investigation of Technologies and Techniques for Labelling Information Objects to Support Access Management," DRDC, Ottawa, Canada, 2005.

[14] R. Housley, *RFC-1457: Security Label Framework for the Internet*, 1993.

[15] S. Oudkerk and G. Lunt, "An Incremental Approach to Trusted Labelling In Support Of Cross-Domain Information Sharing," NC3A, The Hague, Netherlands, 2011.

[16] ITU-T, "X.841: Information technology – Security techniques – Security information objects for access control," 2000.

[17] G. Lunt, S. Oudkerk and A. Ross, "NATO Metadata Binding Service," NCIA, The Hague, Netherlands, 2010.

[18] K. Kongsgård, N. A. Nordbotten and S. Fauskanger, "Policy-Based Labelling: A Flexible Framework for Trusted Data Labelling," in *Proc. International Conference on Military Communications and Information Systems (to appear)*, 2015.

[19] A. Aldini and M. Bernardo, "An Integrated View of Security Analysis and Performance Evaluation: Trading QoS with Covert Channel Bandwidth," in *Proc. SAFECOMP*, *LNCS 3219*, 2004.

# Appendix A Existing guard and labelling solutions

## A.1 Guards

In the following we first provide a survey of guard related publications, and then provide a brief survey of available guard products.

### A.1.1 Literature review

The basic concept of using guards for enabling cross-domain information exchange appears to data back to the late 1970s. The ACCAT[4] Guard [1] supported mail transfers and database queries, but was to a large extent based on manual review as operators were responsible for reviewing and sanitizing each downgrade performed by the guard. More specifically, the ACCAT guard was operated by guard operators who were responsible for sanitizing database query responses (from High to Low) and a security watch officer whose responsibility was to review each downgrade performed by the guard.

A somewhat more recent guard, the Secure Network Server Mail Guard (SMG) [2] enables automatic filtering of e-mail (SMTP) based on configurable filters. The filter configuration could include for instance sender or recipient address, classification label (i.e., a line of text indicating the security classification of a message), type of attachments, whether attachments have been reviewed, the existence of a valid digital signature, and whether confidentiality protection is provided through message encryption. An assured pipeline was provided based on the type enforcement provided by the underlying platform [3], placing type restrictions on collections of programs and data items. The formal assurance of the SMG was concerned with the correctness of this pipeline (i.e., the correct ordered data flow through guard components and that guard components are constrained from interfering), rather than the correctness of the information filtering.

In [4] it is shown how an assured guard pipeline can be implemented using the integrity categories of the mandatory access control provided by the GEMSOS operating system. More specifically, it is shown how assured guard pipelines can be created based on the use of the integrity categories of the mandatory access control, where GEMSOS provides process isolation and support for multi-level subjects. The GEMSOS operating system is used for the GemSeal guard [5].

NCIA has previously developed a medium assurance XML guard [6] for NATO, and has also proposed a Protection Profile [7] for evaluation of high assurance guards according to Common Criteria.

A prototype of a high assurance guard for use in service-oriented architectures [8][9] has been developed as part of a CD&E activity at FFI, in collaboration with Thales Norway. This guard aims to provide sufficient flexibility for use in service oriented-architectures and has been

[4] Advanced Command and Control Architectural Testbed

developed to be certifiable at EAL 5 according to Common Criteria. It supports the use of XML confidentiality labels and requires a separation kernel as the underlying platform.

### A.1.2 Existing guard and security filter products

We will now provide a brief survey of publicly known guard and security filter products. It should be noted that limited information is publicly available about most of these products, thus, the exact functionality and assurance provided by these products are not clear. Nevertheless, the following survey provides an overview of available products as well as an indication of their capabilities.

In general, a wide variety of formats and protocols are supported. While the main focus is on discrete data objects, there are also several products with support for streaming media. For instance, Trusted Manager (TMAN) [54] supports MPEG 2 and 4 video embedded with Key-Length-Values (KLV) transported over SSL.  TMAN performs verification of data stream content and structure and can filter the stream based on sensitivity labels as well as geographic coordinates contained within the KLVs. KLV metadata is also supported by SimShield from Raytheon and the High Speed Guard (HSG) [24][26]. A standard for KLV encoding of security metadata in MPEG2 is found in [46], but it is unclear whether these products make use of this standard.

With regard to assurance there is little evidence available to provide a comparison of the available products. It may be observed however that a majority (10) of the products discussed below is based on some variation of SELinux, while four are based on Solaris with Trusted Extensions. Neither of these are high assurance operating systems (both having been evaluated according CC EAL4), indicating that these guard and security filter solutions are not likely to be certifiable at higher assurance levels. There is also one solution, i.e., the BAE XTS Guard, that is based on STOP 7. As discussed in Appendix B.1.1, STOP 7 is also only evaluated at EAL 4 although it is targeted as a high assurance operating systems and the previous version of STOP was evaluated at EAL5. The products standing out in terms of assurance may therefore appear to be the Turnstile and SecureOne from Rockwell Collins, which both are based on high assurance separation kernels (AAMP7 and VxWorks MILS respectively). Remarkably, it may seem that no guard has so far been certified above EAL4 according to Common Criteria, although Thales TSF101 security filter has been evaluated at EAL5.

**Clearswift Deepsecure 2.1 / LogicaCMG Clearswift Bastion II**

The Bastion II supports SMTP/MIME, X.400 P1&P7, X.525 and SNMPv2. It is implemented using Trusted Solaris and has been certified according to Common Criteria EAL 4 [10][11].

Clearswift Deepsecure 2.1 is based on the Clearswift Bastion II and has also been evaluated at EAL4 [12][13]. It supports SMTP and X.400 messages (including STANAG4406/ACP 123 Military Messages). The evaluation also included X.841 label support libraries for Solaris and Windows.

### Tresys XD Sidecar

The XD Sidecar [14] is a security filter that can be used in conjunction with a guard to provide content inspection (e.g., for malicious content, dirty words, and hidden content) and cleansing of content. It supports Microsoft Office documents, pdf, compressed files, various image formats, and XML. It is said to be based on the filtering technology of NSA's Assured File Transfer (AFT) cross-domain solution.

### BAE XTS Guard

The XTS Guard [16] supports e-mail (SMTP), XML, office files, images, and chat. It is based on BAE's STOP 7 operating system, which has been evaluated at EAL 4+, and is available both as an enterprise version and as a ruggedized version in various form factors. It supports various content filters to be plugged-in (including the BAE NephronMaxx [17] which provides virus detection, file format verification, file name blacklisting, and encrypted data detection). The XTS Guard is said to provide the capabilities of the older XTS DataSync Guard (DSG) [15] and the XTS DII.

### Boeing eXMeritus HardwareWall (HWW)

The eXMeritus HardwareWall [18] supports bi-directional data transfer, combining physical one-way transfer, mandatory access control, data labelling, content review, and multiple proxies. It is accredited at Protection Levels 3, 4, and 5 in the U.S. (unknown in what configurations), being applicable for interconnecting systems from Top Secret to Unclassified including Internet. It is said to deploy on SELinux and supports multiple protocols and data formats (e.g., FTP, SCP, HTTP(s), XMPP, SNMP, DTD, XLS, NTF, PDF, TFD, WAV, XML, DOC, TIF, RTF, TXT, NITF, PPT, HDF4, HDF5, TAR, GZ, ZIP, BZ2, and JAR). It also supports XML signatures, XML Schema verification, and multiple content checks in addition to tools for human review.

### SafeNet Multi-Domain eXchange System (MDeX)

MDeX [19][20] consists of the MDeX Transfer System (MTS), Security Domain Intermediary (SDI), and Remote Management Station (RMS). MTS provides the core security functionality and is based on Solaris 10 with Trusted Extensions, SafeNet information flow engine, Java, and Oracle XACML. It provides flexibility with regard to supported data types through pluggable protocol interfaces and content filters (e.g., supporting JMS, Telemetry, file sharing, structured text, and XML schema data type filtering). It is accredited for specific data types at Protection Level 4 in the U.S. (for up to Unclassified – Secret and Secret – Top Secret).

### Owl Computing Technologies Enterprise Cross Domain Solution – File Transfer (ECDS FT01)

ECDS [21] is primarily intended for transferring files from Low to High, although it can also be used in a High to Low configuration. Files are controlled with ClamAV and ASCII filters before being transferred. ECDS makes use of Owl 2500 DualDiode Communication Cards (NIAP CC EAL4 certified) to provide one-way transfer and a hardened Linux operating system.

### Lockheed Martin Radiant Mercury (RM)

RM [22][23] supports “hundreds of data formats” (including USMTF, OTH-Gold, NITF, TADIL, TDIMF, DIS PDU, and XML) and has functionality to perform sanitization of formatted data. It was originally developed in 1992 and apparently it has more recently been ported from Solaris 10 with Trusted Extensions to SELinux.[5] It has been accredited at Protection Level 4 in the U.S.

### Raytheon High Speed Guard (HSG)

HSG [24][25] is targeted towards applications with high bandwidth requirements (9 Gbps). It supports Web services over HTTP (including XML/SOAP parsing), file transfer, MPEG2/4 real-time video streaming, and VoIP. HSG also supports generic TCP and UDP connections. It runs on Red Hat Enterprise Linux (with SELinux), was accredited up to Protection Level 4 in 2002, and has more than 190 installations. There is also a web-based Human Review Manager that can be used in conjunction with the HSG.

### Raytheon SimShield

SimShield [26] is intended for use in conjunction with distributed simulation operations, live (e.g., multinational) training, and testing. SimShield supports DIS, HLA, TENA, RTP and MPEG2-TS protocols, as well as KLV metadata, and can also sanitize and label data. It is said to be engineered to satisfy SABI and TSABI requirements.

### Raytheon Trusted Gateway System (TGS)

TGS [27] provides for a two-person manual review process using a web-based interface, in addition to automatic content checks such as virus scanning, file type verification, dirty word search (supporting specification of “clean words” that contain a dirty word that should be ignored), and deep content inspection (including removal of hidden and embedded data and metadata). Data that pass the automatic content checks can be released by explicitly authorized users, to approved destinations, without invoking the human review process. Authorization controls (username, password, X.509 digital certificates, clearance level, and group management) are configured and managed within the TGS or through an Active Directory or LDAP server on the high-side network. TGS runs on Red Hat Enterprise Linux (with SELinux components), and is said to be engineered for TSABI and SABI requirements.

[5] http://www.dtic.mil/descriptivesum/Y2013/Navy/stamped/0304231N_5_PB_2013.pdf

### Raytheon WebShield

WebShield [28] is an HTTP(s) guard (acting as a Web proxy) providing browsing and search of Low network from High network, also supporting chat (XMPP over HTTP). It provides virus scanning, dirty word search, file type verification, and active content blocking. It runs on Red Hat Enterprise Linux and is said to be engineered to satisfy TSABI and SABI requirements.

### Lockheed Martin Trusted Manager (TMAN) II

TMAN II [29][30] is aimed towards sharing across intelligence networks, supporting multiple file types (including graphics, imagery, mission data, mark-up languages, audio, video, and office formats in addition to streaming media). Release of information is generally performed by human review, but TMAN also supports automatic downgrade e.g., for highly formatted alphanumeric data or when data sensitivity can be ascertained through a digital signature. TMAN runs on Solaris 10 with Trusted Extensions, is certified at Protection Level 4, and has multiple SABI and TSABI accreditations.

### Rockwell Collins Turnstile Version D and MicroTurnstile

Turnstile [31] is based on the AAMP7 multiprocessor (i.e., a hardware based separation kernel) and can be accredited to Protection Level 5 (Top Secret – Unclassified). According to a presentation from 2007 [32] the Turnstile conforms to the Common Information Sharing Standard for Information Security Marking and allows comprising matching rules on tag values using AND, OR, NOT, EQUAL, and CONTAINS operators. According to the same presentation the Turnstile basic guard engine has also been ported to the Rockwell Collin’s SecureOne guard. SecureOne [33][34] makes use of WindRivers VxWorks MILS as the underlying platform.
The MicroTurnstile [35] is a USB-variant of the Turnstile that only weighs 65 grams and is intended as a soldier-wearable device. It makes use of a configurable rule set and supports Variable Message Format (VMF) messages and VoIP.

### Thales XOMail/Guard

The XOMail/Guard [36] allows automatic two-way flow of mail based on security labels. It has been certified according to TCSEC B1+ using Unix System V/MLS as the underlying platform. It is indicated in [37] that a new XOmail GUARD will be released in 2016.

### Thales Trusted Security Filter (TSF)

The Trusted Security Filter (TSF-101) [38][39] has been certified according to Common Criteria EAL5. It enables IP datagrams complying with the filter criteria to be released while non-complying messages are logged.

### Infodas RSGate

The RS Gate by Infodas [40] is a security gateway offering both automatic content checkers for XML documents based on configurable sets of rules and the possibility of manual review through a secure viewer where the operator can inspect the document in trasnsit.  Reviewed documents are then signed for release and sent forward as e-mail attachment. It is not clear whether the hwole solution is certified, but it uses the  GeNUGate firewall which is certified at EAL 4+, to monitor the traffic from low to high.

**Nexor**

The Nexor Sentinel 3E Filtering System [41] is the filtering engine of a high assurance mail guard based on SELinux. It offers four types of filters supporting the security policies: dirty word searching, structured and unstructured security labels and allowed attachment types. abel  It is EAL 4+ certified.

**Advatech Pacific TACDS and CenturionCDS**

The Tactical Cross Domain Solution (TACDS) and CenturionCDS [42] build on a common core called CyberGuard security framework, which is FIPS 140-2 Level 4 certified. They support configurable and pluggable filters for different types of message formats, including video, XML, binary and others. The CenturionCDS in particular also offer hardware enforced domain separation and secure boot and trusted platform verification upon power up.

## A.2 Data labelling

Here we present available products and standards related to security labels.

### A.2.1 Standards for the definition of labels

The actual format of the label is dependent on the language and syntax used to implement it, but XML labels seem to be the preferred choice to label discrete information objects (e.g., data files, reports, documents, photographs, database elements) as reported in [43], which presents an overview of standards and studies adopting security labels for access control up to 2005. As a further confirmation of this trend, the only other more recent standards of which we are aware are also XML-based. One  was proposed in 2010 as a joint effort of the NATO working group IST-068/RTG-031, to be used in conjunction with a guard mechanism [44], and the other is the Data Encoding Specifications for Information Security Marking Metadata (DES-ISM) latest updated in 2013 [45].

A separate standard for security metadata used to label video streams has been defined in [46]. This standard does not use XML as chosen format, but KLV (Key-Length-Value), a data encoding standard often used to embed information in video feeds. How this metadata is to be created, embedded, and used is however not part of the specification as “Originators and application users are responsible for the proper handling and ultimately for the use and disposition of classified information”.

Some of the guard and security filters presented in the previous chapter of this appendix also claim to use security or “sensitivity” labels, but the specifics of these labels are unclear. Sometimes the label consists of a single line of text reporting the classification level.

### A.2.2 Commercial products using labels

When it comes to existing products using security labels, we find some guard and security filters claiming to filter information based on security labels, but they usually assume that the label is already bound to the data objects [36,42]. Some offer also labelling functionalities [18,26], but most likely using a policy based on origin. MLS operating systems (see Appendix B.1) use security labels internally for access control and separation, not to export the data to another domain. While these OSs and their underlying hardware platforms might be trusted, the labels are not flexible enough to be used with specific file formats and applications.

There do exist some products that provide labelling functionalities for labelling e-mails in Microsoft Outlook or documents in Microsoft Office like those described in [47,48] and mentioned in [18], but they build on Windows security mechanisms, and therefore cannot be more trusted than the operating system itself. Many databases also offer security labels [49,50], but they can protect the data and the label only as long as they are inside the DB since no explicit binding seems to be available.

Concluding, no products specifically designed to generate high-assurance labels seem to be available when the user also is part of the selection process as described in Chapter 3. The reason is that ultimately, although the labelling program is correctly implemented, we cannot trust the operating system on which the program is running, or even the application used to create the data that needs to be labelled, so the metadata selection phase is not trustworthy enough to give high confidence in the labels.

# Bibliography

[1] J.P.L. Woodward, “Applications for multilevel secure operating systems,” *Proc. of the National Computer Conference*, 1979.

[2] R. Smith, “Constructing a High Assurance Mail Guard,” *Proc. National Computer Security Conference*, 1994

[3] O. Sami Saydjari, “LOCK: An Historical Perspective,” *Proc. Computer Security Applications Conference*, 2002.

[4] M.R. Heckman, R.R. Schell, and E.E. Reed, “A High-assurance, Virtual Guard Architecture,*”Proc. Military Communications Conference*, pp. 1233-1241, 2012

[5] M.R. Heckman, R.R. Schell, and E.E. Reed, “Composing a high-assurance infrastructure out of TCB components,” *Layered Assurance Workshop*, 2011.

[6] K. Wrona, S. Oudkerk, and G. Hallingstad, “Designing medium assurance XML-labelling guards for NATO,” *Proc. Military Communications Conference*, 2010.

[7] K. Wrona and N. Menz, “Protection Profile for the NATO high assurance ABAC guard (HAAG), version 1.3,” *NCIA Technical Report TR-2012-SPW0084-18-13-4*, 2013.

[8] R. Haakseth, N.A. Nordbotten, B. Kristiansen, and Ø. Jonsson, “CD&E EP1328 Guard for cross-domain information exchange,” FFI-report 2014/01182, 2014.

[9] R. Haakseth, N.A. Nordbotten, Ø. Jonsson, and B. Kristiansen, “A high assurance cross-domain guard for use in service-oriented architectures,” to appear in *Proc. International Conference on Military Communications and Information Systems*, 2015.

[10] UK IT Security Evaluation and Certification Scheme, “CS Bastion II”, Common Criteria Certification Report No. P184, 2003, http://www.commoncriteriaportal.org/files/epfiles/CRP184.pdf

[11] Andrea Gilbert, “Clearswift CS Bastion II Security Target (EAL4),” 2003, http://www.commoncriteriaportal.org/files/epfiles/cs_bastion.pdf

[12] CESG, “Clearswift DeepSecure Release 2.1,” Common Criteria Certification Report No. P228, 2006, http://www.commoncriteriaportal.org/files/epfiles/CRP228.pdf

[13] Clearswift, “Deepsecure (CSDS) Release 2.1 Security Target,” 2006, http://www.commoncriteriaportal.org/files/epfiles/DeepSecure2.1.pdf

[14] Tresys Technology, “XD Sidecar: Assured Information Solution for Complex File Type Filtering,” http://www.tresys.com/products/datasheets/XD-Sidecar-Datasheet.pdf

[15] Cross Domain Wiki, “Data Sync Guard,” http://www.crossdomain.org/index.php?title=DataSync_Guard_(DSG)

[16] BAE, “XTS Guard,” http://www.baesystems.com/cs/groups/public/documents/document/mdaw/mtg4/~edisp/baes_165741.pdf

[17] BAE, “NephronMaxx,” http://www.baesystems.com/cs/groups/public/documents/document/mdaw/mtg4/~edisp/baes_165742.pdf

[18] Boeing, “eXmeritus HardwareWall secure data transfer system," http://www.boeing.com/advertising/c4isr/isr/232341_HardwareWall_Insert.pdf

[19] SafeNet, “SafeNet Multi-Domain eXchange (MDeX) System – Product Brief,” http://www.safenet-inc.com/uploadedFiles/resources/product-brief-new/data-protection/MDeX%20General%20Description.pdf

[20] Shawn Campbell, “Multi-Domain eXchange (MDeX) System – cross-domain (aka assured information sharing for tactical coalition, disaster recovery, and service-based assured information sharing,” http://www.carahsoft.com/pdf/MDeXSystemLuncheonBrief.pdf

[21] Owl Computing Technologies, Enterprise Cross Domain Solution (ECDS-FT01), http://www.owlcti.com/pdfs/datasheets/government/ECDS.pdf

[22] SPAWAR, “Radiant Mercury,” http://www.fas.org/irp/program/disseminate/radiant_mercury.pdf

[23] Lockheed Martin, “Lockheed Martin’s Radiant Mercury System Certified Using National Institute of Standards of Standards & Technology Guidelines,” 2010, http://www.lockheedmartin.com/us/news/press-releases/2010/december/LockheedMartinsRadiantMer.html

[24] Raytheon, “High Speed Guard,” 2014, http://www.raytheon.com/capabilities/rtnwcm/groups/gallery/documents/digitalasset/rtn_216064.pdf

[25] DefenseNews, “On Guards,” 2010, http://www.defensenews.com/article/20100801/C4ISR01/8010304/On-guards

[26] Raytheon, “SimShield,” 2014, http://www.raytheon.com/capabilities/rtnwcm/groups/gallery/documents/digitalasset/rtn_216073.pdf

[27] Raytheon, “Trusted Gateway System – Secure Multi-Directional Data Transfer,” http://www.trustedcs.com/resources/brochures/RTCS_TrustedGatewaySystem_datasheet.pdf

[28] Raytheon, “WebShield,” http://www.raytheon.com/capabilities/rtnwcm/groups/gallery/documents/digitalasset/rtn_216416.pdf

[29] Lockheed Martin, “Trusted Manager (TMAN) – Secure, Trusted Information Sharing,” http://www.lockheedmartin.com/content/dam/lockheed/data/isgs/documents/TMAN%20Brochure.pdf

[30] Lockheed Martin, “Lockheed Martin Cross Domain Solution GSA Purchasing,” 2012, http://www.lockheedmartin.com/content/dam/lockheed/data/isgs/documents/isgs-cds-pricing.pdf%20(2).pdf’

[31] Rockwell Collins, “Turnstile – high assurance, cross domain solution,” 2012, http://www.rockwellcollins.com/~/media/Files/Unsecure/Products/Product%20Brochures/Information%20Assurance/Cross%20Domain/Turnstile%20data%20sheet.aspx

[32] M. Bortz et al., “A High-Assurance Cross Domain Platform,” *NSA High Confidence Systems and Software (HCSS)*, 2007, http://cps-vo.org/node/2339

[33] Rockwell Collins, “SecureOne Cross Domain Technologies,” 2011, http://www.rockwellcollins.com/~/media/Files/Unsecure/Products/Product%20Brochures/Information%20Assurance/Cross%20Domain/SecureOne%20data%20sheet.aspx

[34] Military Aerospace, “Rockwell Collins SecureOne runs on Wind River VxWorks MILS platform, achieving cross-domain, multilevel security on a single aircraft display for reduced SWaP”, 2011, http://www.militaryaerospace.com/articles/2011/11/rockwell-collins-secureone.html

[35] Rockwell Collins, “MicroTurnstile Cross Domain Solution,” http://www.rockwellcollins.com/~/media/Files/Unsecure/Products/Product%20Brochures/Information%20Assurance/Cross%20Domain/MicroTurnstile%20data%20sheet.aspx

[36] Thales Norway, “XOmail/Guard – The MLS certified X.400 Mail Guard Variant,” 2006, http://www.xomail.com/dl/xomail-guard-brochure.pdf

[37] Thales Norway, “XOmail – Secure information exchange,” https://www.thalesgroup.com/sites/default/files/asset/document/thales-xomail.pdf

[38] Thales, “TSF-101,” https://www.thalesgroup.com/en/node/562046

[39] Sertit, “Thales Trusted Security Filter TSF101,” http://sertit.no/product/15

[40] GeNUA / Infodas, “RS Gate: Security gateway for Information Exchange at Red/Black Interfaces,” http://www.infodas.de/download/flyer_rs-gate_englisch.pdf

[41] Denise Cater, “Nexor Sentinel 3E Filtering System Certification Report,”2012, http://www.tuv-nederland.nl/download.php?c_report=36

[42] Advatech Pacific, “Tactical Cross Domain Technologies,” http://www.tacticalcds.com/

[42] Lockheed Martin, “Trusted Manager Streaming Data,” http://www.lockheedmartin.com/content/dam/lockheed/data/isgs/documents/TMAN%20Streaming%20Data%2008%2027%2013.pdf

[43] A. Magar, “Investigation of Technologies and Techniques for Labelling Information Objects to Support Access Management”, DRDC Ottawa CR 2005-166, November 2005.

[44] S. Oudkerk, I. Bryant, A. Eggen and R. Haakseth, "A Proposal for an XML Confidentiality Label Syntax and Binding of Metadata to Data Objects," in *NATO RTO Symposium on Information Assurance and Cyber Defence*, 2010.

[45] U.S. Intelligence Community, “XML Data Encoding Specification for Information Security Markings”, Version 11, 2013. http://www.dni.gov/index.php/about/organization/chief-information-officer/information-security-marking-metadata

[46] Motion Imagery Standards Board (MISB), “Security Metadata Universal and Local Sets for Digital Motion Imagery,” MISB STANDARD 0102.10, 2013. http://www.gwg.nga.mil/misb/docs/standards/ST0102.10.pdf

[47] Boldoni James, " Information Classification Delivering Security and Business Value”, http://info.boldonjames.com/l/35632/2014-03-31/6hpr/35632/11998/Information_Classification_WP.pdf, 2010.

[48] Titus, “Optimizing Information Sharing - How Military Organizations Can Share Information Securely and Effectively By Using a Classification Management Solution”, http://www.titus.com/resources/marketo/WEB_CLS_WP_Optimizing_Info_Sharing_Military.pdf?alild=15079987, 2011.

[49] Oracle, “Oracle Label Security with Oracle Database 12c”, http://www.oracle.com/technetwork/database/options/label-security/label-security-wp-12c-1896140.pdf?ssSourceSiteId=ocomen, 2013.

[50] IBM, “Label-Based Access Control”, http://www-01.ibm.com/support/knowledgecenter/SSGU8G_12.1.0/com.ibm.sec.doc/ids_lb_002.htm

# Appendix B Enabling concepts and technologies

We will here consider some enabling technologies and concepts that may be useful building blocks in creating solutions for cross-domain information exchange. First, we consider available high assurance operating systems that may potentially serve as platform for high assurance solutions (e.g., guards, labelling mechanisms, and access solutions). Then we consider hardware mechanisms (e.g., Trusted Platform Module and TrustZone) available on many commodity systems that may be used to provide integrity attestation, hardware based isolation, and protection of cryptographic keys. Although these are not high assurance solutions, they may potentially have an important role in implementing security functionality with higher trust on end-user machines.

An overview of Attribute Based Access Control (ABAC) and the related NATO/NCIA proposal of Content Based Protection and Release (CPR) is then given. ABAC could potentially for instance serve an important role in strengthening the need-to-know access control within domains, thereby reducing the potential damage caused by insiders or malware with regard to information leakage.

## B.1 High assurance operating systems

The security and trustworthiness of a system not only depends on the proper security functionality being in place, but also on the assurance that this functionality is correctly provided under all relevant circumstances and that it fulfills the security requirements of the system. More specifically, the Common Criteria [1] defines assurance as the grounds for confidence that a target of evaluation meets its functional security requirements.

Put in another way, assurance can be viewed as a means for reducing security risk to an acceptable level. If a security breach only has a low impact, lower assurance may be appropriate. On the other hand, if a security breach may have high impact, high assurance that such a security breach will not happen is likely required. As such, a high assurance system is one which is applicable to high-risk situations.

Although the term high assurance is somewhat inconsistently used, it has traditionally been used to refer to products evaluated according to Common Criteria Evaluation Assurance Level (EAL) 6 or 7 or equivalent, and often also EAL 5. Products evaluated at EAL 6 would here be applicable for use in high-risk situations, while products evaluated at EAL 7 would be applicable in extremely high-risk situations. Because several nations have moved away from the concept of EALs, and the new Common Criteria Recognition Agreement (CCRA) is not applicable to high assurance evaluations, there has been limited progress in the evaluation of high assurance operating systems the last years. As the number of evaluated high assurance operating systems is limited, this report considers systems that have been evaluated to at least EAL 5 or equivalent or that are believed to be capable of being evaluated to such a level.

Cross-domain information exchange is one application that may include a high security risk and thus require high assurance security solutions. For instance, loss of information classified as

NATO Secret is by definition considered to cause serious damage to NATO, and the interconnection of such security domains with unclassified and potentially uncontrolled security domains as such pose a high security risk.

In order to implement a security service with a given level of assurance, it should be based on an operating platform with at least an equivalent level of assurance. This chapter therefore provides a brief survey of available high assurance operating systems.

It should be noted that the term operating system is used in a broad sense in this survey. Some of the discussed solutions provide a very limited functionality, and as such do not provide the functionality one would typically expect from an operating system. Indeed, no commodity high assurance general purpose operating system seems likely to become available in the near future.

It should also be mentioned that there are several operating systems with a strong emphasize on security that is not included in this survey. For instance, SELinux and Solaris Trusted Extensions (Trusted Solaris) provide substantial security functionality, but are not high assurance operating systems as considered in this survey. Specifically, Solaris 10 with Trusted Extensions and Red Hat Enterprise Linux are evaluated to EAL4 (both conformant with the Controlled Access Protection Profile, the Role Based Access Control Protection Profile, and the Labeled Security Protection Profile). According to the NSA [1], SELinux was simply intended as an example of how mandatory access control can be added into a system such as Linux, not an attempt to implement a secure system.

### B.1.1 STOP/XTS-400

XTS-400 from BAE Systems is an MLS type of operating system and is a successor to SCOMP (Secure Communications Processor), XTS-200, and XTS-300. XTS-400 was first successfully evaluated at EAL5 (augmented with ALC_FLR.3 and ATE_IND.3) in 2005, and an updated version was again evaluated at the same level in 2008 (both times with support for the Labeled Security Protection Profile and the Controlled Access Protection Profile).

XTS-400 is a combination of the Secure Trusted Operating Program (STOP) version 6.4.U4 and specific hardware. The hardware included in the evaluation is based on the Intel IA-32 architecture,[6] and also includes several peripheral devices such as hard-disks, floppy drives, tape drives, video controllers, DVD-drives, keyboard/mouse, monitors, Ethernet cards, and printers.

[6] CPUs: Intel Pentium III and Intel Xeon (P4) “Prestonia”. Motherboards: Intel L440GX and Intel SW7501.

The security target for XTS-400 [4] states that operating systems evaluated against that security target will:

Associate sensitivity labels with all objects and all its users will have an associated clearance level identifying the maximum security level of data that they may access.

Allow simultaneous use of the system by multiple users, all with different clearances and needs-to-know.

Allow simultaneous network connectivity to networks of differing sensitivities/classifications (including IPv6 networks).

Provide mandatory integrity protection of files.
Provide an untrusted operating environment that includes common Linux commands and tools.
Provide an Application Programming Interface/Application Binary Interface which is suitable for running most Linux applications in their binary format (no recompilation required).

To achieve this, the STOP kernel provides support for multitasking where each process is isolated in a virtual process environment. The security kernel running in ring 0 performs both mandatory access control as well as integrity control when a process is to access an object. The security kernel also provides I/O services and an inter-process communication message mechanism. Discretionary access control to the file system is enforced by the trusted system services running in ring 1, while the operating system services running in ring 2 provide a Linux interface to the applications running in ring 3. Although this provides a layered security model, there are also trusted applications running in ring 3.

A more recent offering, STOP 7 [5], is to provide broader deployment options and is used for the XTS Guard 5. Although STOP 7 is targeted as a high assurance operating system, it has so far only been evaluated to EAL 4+ [6].

### B.1.2 MILS Separation Kernels

A MILS separation kernel provides the means to have several strongly separated partitions, each hosting a guest operating system or a native application running on an interface provided by (or on top of) the separation kernel. The latter is beneficial when implementing high assurance functionality where it is not desirable to have a complete operating system within the partition. It is also possible to configure one- or two-way data flows between partitions (e.g., using FIFO message channels or shared memory). Likewise, a partition may be assigned ownership of a physical device (e.g., USB, disk, network interface, keyboard, mouse, and so on), but the separation kernel itself does not provide functionality to share such a device between multiple partitions in a secure manner.

Separation kernels implemented in software are available from several vendors:

- Integrity-178B from Green Hills Software
- VxWorks MILS from WindRiver
- PikeOS from Sysgo
- LynxSecure from LynuxWorks

Additionally, there is the Advanced Architecture MicroProsessor 7 Government Version (AAMP7G) from Rockwell Collins [7], basically providing a separation kernel in hardware. The AAMPG7 is said to have been proven mathematically using formal methods as specified for EAL7 of the Common Criteria.

Of the software implementations, Integrity-178B has been certified to EAL6+ [8-10], on specific PowerPC platforms, according to the U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness (SKPP) [11]. The general availability of the evaluated hardware is however unclear.

As of 2011 the SKPP has been "sunset" [12], meaning no additional evaluations (including maintenance evaluations) will be performed according to this profile. Furthermore, the Information Assurance Directorate is said to from now on be focusing on specific government systems (e.g., making use of separation kernels) instead of performing general evaluations (e.g., of separation kernels).

There is ongoing work towards performing an evaluation of PikeOS in Europe, and the EURO-MILS project[7] is planned to publish a proposal for a new Protection Profile.

It may also be noted that there is ongoing work on an open-source separation kernel, the Muen Separation Kernel.[8] Furthermore, Dornerworks has performed substantial work on the ARLX separation kernel based on the Xen hypervisor [2].

### B.1.3 PolyXene

PolyXene [13] (v2) is an end-user solution for handling multiple security levels using a MILS type of architecture. The previous version of PolyXene (v1) has been successfully evaluated according to Common Criteria EAL 5 [14], but the security target is not publicly available.

Technically, the system is separated into a trusted zone and one or more standard zones, where a zone is similar to a separation kernel partition. The trusted software is executed within the trusted zone, while the standard zones each may be used to run a guest operating system (e.g., Windows or Linux). This virtualization capability within the standard-zones appears to be provided using the Xen virtual machine.[9] Inter-zone communication (e.g., one-way FIFO channels) can be

[7] http://www.euromils.eu/

[8] http://muen.codelabs.ch/

[9] By having a Xen virtual machine within each standard zone one gets multiple (i.e., poly) Xen instances, which could perhaps be the origin for the name PolyXene.

enabled through the use of virtual devices. Virtual devices may also be used for providing standard zones access to a device owned by the trusted zone (thereby enabling device sharing).

PolyXene is mainly targeted at end-user workstations and laptops, supporting features such as token based authentication. Still, the PolyXene zones may also be used to provide separation and (e.g., one-way) information flow control between the various functions of a guard. However, the illustrated design for this [13] makes use of Linux/Windows within each zone. Without an application programming interface on top of the hypervisor, a stripped down Linux may be the most practical solution for this usage. Whether this is a suitable solution depends on the assurance requirements for the guard filter. Implementing application specific functionality within the trusted zone (if at all possible) would imply modifications to the existing trusted code base, which is generally not desirable.

### B.1.4 GEMSOS security kernel

The Gemini Multiprocessing Secure Operating System (GEMSOS) security kernel [15] has been previously evaluated according to TCSEC Class A1 (i.e., the Orange Book) as part of the Gemini Trusted Network Processor [16]. The rights to GEMSOS are owned by Aesec, after their acquisition of Gemini Computers in 2003.

GEMSOS is an MLS type operating system that runs on the Intel x86/IA-32 architecture. It implements a reference monitor to enforce mandatory access control. Access control is enforced based on security labels specifying both confidentiality and integrity (using a hierarchical lattice combined with non-hierarchical categories for each) [17].

In [17] it is shown how the mandatory access control provided by GEMSOS can be used for implementing a guard solution, where GEMSOS provides process isolation and support for multi-level subjects. It also illustrates how assured guard pipelines can be created based on the use of the integrity categories of the mandatory access control.

### B.1.5 seL4

The seL4 microkernel [18] has been subject to full formal verification, and has now (as of 2014) been open-sourced (including proofs) by General Dynamics and NICTA.[10] A microkernel may be seen as a minimalistic kernel (operating in privileged mode), providing a minimal set of features. The seL4 microkernel has for instance been proved to provide information flow non-interference and integrity and authority confinement. The seL4 may form a secure basis as part of a more functionally rich operating system, can support virtualization, and can be configured as a separation kernel providing strong isolation. seL4 supports both ARM and x86 processors, although the port to x86 has not been formally verified.

[10] http://sel4.systems

### B.1.6 Summary

Several high assurance operating systems with varying functionality have been discussed. While there are other high assurance operating systems that could have been included, the ones included were selected due to being considered as candidate systems for implementing high assurance components (in particular guards) for use in cross-domain solutions today.

Depending on the specific scenario, several (potentially all) of the discussed systems may be suitable for implementing cross-domain components. Several of the systems (e.g., STOP, separation kernels, and GEMSOS) have also been previously used or demonstrated for this purpose. An advantage of separation kernels is that there are several vendors, while their strong separation and controlled information flow is well suited to guard design. Likely differences in the provided APIs, additional functionality, and supported hardware still implies that porting from one separation kernel to another may not be without cost. It may also be noted that the seL4 microkernel appears very promising, and its further development is therefore interesting.

Independent of what operating system is selected for a given project, one should also consider what development tools and support can be provided, as well as what documentation is available to support the evaluation of the final system.

## B.2 OS independent mechanisms for data and integrity protection

Even though an operating system is certified for high-assurance, it is still critical to be able to establish trust in the device running such an operative system. This means being able to verify that a device is actually running the expected software, it has an approved configuration and its integrity is preserved over time.

For high certification this is achieved by certifying the OS together with the hardware it will run on as a whole. The integrity of this “box” is then protected by employing tamper-proof or tamper-responsive technologies that prevent physical tampering or show clear signs of it. This helps assessing the physical integrity of the system for someone who can directly interact with it, but other systems that communicate remotely with it may also need some proof of its identity and integrity. This is why the manufacturer would also place a private key in the device and release a corresponding public certificate to allow a third party to verify its genuineness.

This approach is very common, but it is little flexible as any upgrade would require a new certification of both software and hardware and it is often difficult to integrate and manage such devices in an existing infrastructure. The advantage is that as long as the public certificate is validated correctly, we have some assurance that the device is the genuine one and that the security properties for which it was certified are preserved.
However, new mechanisms are emerging in commercial systems that could help create more flexible high-assurance systems where the hardware provides certified secure functionalities like strong memory separation and integrity independently from the OS.

These technologies are not yet mature enough to be used to the required assurance levels, but some building blocks already exist.

### B.2.1 Integrity attestation

Given a certified hardware platform that can run different types of critical software, the challenge is to bootstrap trust in the device itself, and then monitoring or protecting its integrity while it is operating. This because we do not have a static sealed system anymore, but something that can potentially run different software with different configurations, and a flexible mechanism to measure the integrity of the system is necessary.

A way to achieve an integrity verification of the system, that is becoming increasingly widespread, is to have a trusted component on the system measuring all components that are executed since the system is booted. Usually this is achieved by having each component measuring the next one in the boot chain, hence creating a chain of measurements where each measurement is trusted if the component who took it is itself trusted. This is often called a *trusted boot*.

The condition for the trusted boot process to work is that a *trustworthy* and immutable piece of code or hardware initiates the measuring process, so that a trust chain of measurements can be built. This first component is known as *the root of trust for measurement*. In addition, the integrity measurements must be stored for later verification in a shielded location that protects them from any modification; hence a *root of trust for storage* is needed. Finally, a private-public key pair is usually employed as a *root of trust for reporting* to sign the measurements so that a third party can verify where they originated.

This approach has been promoted strongly by the Trusted Computing Group, which developed the specifications for a commodity chip with the described properties called Trusted Platform Module (TPM) [19].

The problem is that the TPM has not been certified for assurance levels higher than EAL 4+ [20]. There exists a protection profile that models TC support for high assurance Security kernel [21] which has been evaluated for EAL 5, but no products seem to have been certified according to this PP yet.

Another issue is that the actual core of the root of trust, namely the code that starts the measuring process, does not reside in the TPM itself, but in the BIOS, so that it is more exposed to potential tampering and not as standardized as the TPM itself. Research has showed that this code is very buggy and not too hard to subvert [22]. This in spite of a NIST publication with guidelines on how to build a secure BIOS [23].

The TPM is essentially a crypto-processor specialized for measuring and reporting the integrity of a platform, but unlike other crypto co-processors it does not offer any particular tamper resistance feature. In fact, physical attacks are out of its specifications.

Other programmable crypto co-processor like the IBM PCIe family (CEX3C/4765), which are validated to FIPS PUB 140-2 security--overall Level 4, might be more suitable for high-assurance systems, but should specifically be programmed to perform the same operations a TPM was designed to do from the start.

### B.2.2 Hardware-based isolation

The limitation of the Trusted Boot process is that the integrity measurements are reliable only until the operating system takes control of the platform. At that point there are often too many processes to monitor, and checking the binary file of each application only helps as long as there are no security holes that can be exploited at run time. Besides, a poorly implemented BIOS can jeopardize the security of the whole process.

Intel and AMD developed hardware-based technology to mitigate the problem: Intel v-Pro [24] and AMD SVM [25]. The concept is identical in both cases. Rather than statically measuring the pre-boot environment, a special processor instruction can be invoked at any time to create a sanitized environment, with protected memory and no direct DMA access, where security critical software can be launched and run in isolation from a potentially compromised system (including the BIOS).

This does not replace the TPM or equivalent solutions, but complements it by recording all the actions that took place on the platform form the moment the special processor instruction was invoked to the code of the application that was launched. By reporting these measurements, a third party can verify that a given application is indeed running in a secure environment.

Since the most common application of this technology is to securely launch a “secure” hypervisor or micro-kernel, which then takes care to enforce some software-based isolation mechanism, Intel and AMD also developed various technologies to offer hardware-based virtualization and memory isolation like Intel Vt-x and AMD-v. Such technology can be used to harden high-assurance kernels as well, that rely on compartmentalization to protect system integrity.

One problem that has been brought up though, is that ultimately the BIOS can still compromise the security of this solution through the SMM [25], and that the software provided by Intel to set up the secure environment (the so called SINIT ACM) can be buggy and allow an attacker to bypass the hardware security mechanisms [26]. Such systems need therefore to be certified for high assurance themselves before they can support a high-assurance OS.

ARM developed a set of hardware extensions, called TrustZone [28], that provide a secure execution environment in parallel to the normal one. These are quite flexible and can be used to implement either simple secure APIs, more advanced stand-alone applications, or a separate secure OS running alongside the normal rich OS. The technology is mainly aimed at mobile devices like mobile phones and tablets.

## B.3 ABAC and CPR

The eXtensible Access Control Markup Language (XACML) [29] defines a language for specifying access control policies. In XACML, access control policies and decisions can be based on attributes of the subject (i.e., the user, process, or other entity requesting access), the resource, the environment, and/or the action to be performed. This is illustrated in Figure B.1. Such a model has become known as Attribute Based Access Control (ABAC), and provides a high degree of flexibility when defining access control policies. It may also be noted that XACML 3.0, the most recent version of XACML, takes an even more general approach to attributes and also allows new attribute categories to be defined.

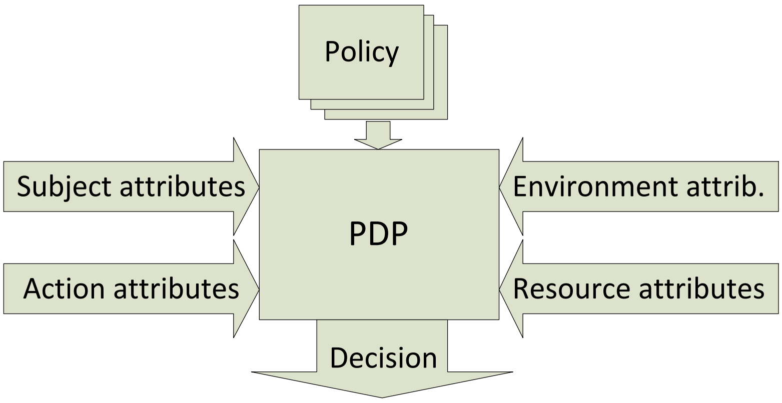

*Figure B.1 Conceptual ABAC model with Policy Decision Point (PDP) making policy decisions on the basis of a set of policies and applicable attributes.*

The attributes of a subject may for instance include its roles, security clearance, and a unique identifier. Environment attributes may for instance include such things as the time of day, the current threat level, or the network classification. A given rule within a policy considers specific attributes to decide its outcome, and may potentially make use of arithmetical, comparative, set, and Boolean operators.

Because attributes can be used to specify both roles and identities, ABAC can be seen to encompass both identity and role-based access control. In principle, it can also be used to enforce mandatory access control such as imposed by the Bell-La Padula model. In this respect, an ABAC model can be used to model and implement a wide variety of other access control models.

ABAC has gained increasing popularity, and has become the recommended access control model for the U.S. Federal Government to promote information sharing between diverse and disparate organizations. [30].

Also in NATO, the use of ABAC is seen as a possible enabler for secure and efficient information sharing among coalition partners and external organizations. The proposed solution is referred to as Content-based Protection and Release (CPR) [31]. In CPR, access control decisions are taken by considering the attributes of the user (i.e., subject), the resource, and the terminal of the user.

The user terminal could be considered as part of the environment or have its attributes associated with the subject, but is considered a distinct attribute category in CPR.

In the CPR concept, security markings (e.g., confidentiality classification) are ultimately no longer to be used except for perhaps on paper documents. Instead the releaseability and protection requirements of a data object are to be decided based on its content properties (i.e., resource attributes), in context with the subject and terminal attributes, as specified by policies. This separates the association of attributes with resources from the process of determining their protection requirements and release conditions, which is seen as an advantage.

With regard to policy, CPR distinguishes between a release policy and a protection policy. The first considers user, resource, and contextual (i.e., environmental) attributes, while the latter considers resource, terminal, and contextual attributes. A motivation for this is to provide separation between the management of the two policies.

## Bibliography

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, 2012.

[2] D. Greve and S.H. VanderLeest, “Data Flow Analysis of a Xen-based Separation Kernel,” *Proc. Layered Assurance Workshop*, 2013. http://www.acsac.org/2013/workshops/law/2013-law-proceedings.pdf

[3] NSA, Security-Enhanced Linux, 2009. http://www.nsa.gov/research/selinux/

[4] BAE Systems Information Technology, "Security Target, Version 1.22 for XTS-400, Version 6.4.U4," http://www.niap-ccevs.org/cc-scheme/st/st_vid10293-st.pdf.

[5] BAE Systems, "STOP OS, Secure application platform," http://www.baesystems.com/cs/groups/public/documents/document/mdaw/mtu3/~edisp/baes_134239.pdf

[6] Communications Security Establishment Canada, “Certification report, EAL 4+ Evaluation of BAE Systems STOP OS v7.3.1,” 2012, www.cse-cst.gc.ca/.../bae-systems-v731-cert-eng.pdf

[7] Rockwell Collins, Advanced Architecture MicroProcessor 7 Government Version (AAMP7G), http://www.rockwellcollins.com/~/media/Files/Unsecure/Products/Product%20Brochures/Information%20Assurance/Crypto/AAMP7G%20data%20sheet.aspx

[8] NIAP, "Validated Product - Green Hills Software INTEGRITY-178B Separation Kernel, comprising: INTEGRITY-178B Real Time Operating System (RTOS), version 750CXe," 2008. http://www.niap-ccevs.org/st/vid10119/

[9] NIAP, "Validated Product - Green Hills Software INTEGRITY-178B Separation Kernel, comprising: INTEGRITY-178B Real Time Operating System (RTOS), version IN-ICR750-0402-GH01_Rel (Version 4.2) running on Compact PCI card, version CPN 944-2021-021 w/PowerPC, v750CXE," 2011. http://www.niap-ccevs.org/st/vid10362/.

[10] NIAP, "Assurance Continuity - Green Hills Software INTEGRITY-178B Separation Kernel, comprising: INTEGRITY-178B Real Time Operating System (RTOS), version IN-ISP448-0100-SK_LMFWPCD2_Rel running on JSF PCD System Processor CCA, version 437140-007 w/PowerPC, v7748," 2009. http://www.niap-ccevs.org/st/vid10119/maint200/

[11] National Information Assurance Partnership, "Archived U.S. Government Approved Protection Profile - U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness, Version 1.03," http://www.niap-ccevs.org/pp/PP_SKPP_HR_V1.03/

[12] NIAP, "SKPP Sunset Q&A," http://www.niap-ccevs.org/announcements/SKPP Sunset Q&A.pdf

[13] Bertin Technologies, “PolyXene – A high end solution for Multiple Independent Levels of Security, PolyXene 2.0 Technology White Paper”

[14] Bertin Technologies, PolyXene [Available:] http://www.polyxene.fr/

[15] Aesec, GEMSOS Security Kernel RTOS – High Assurance MLS Security & Performance.

[16] National Computer Security Center, Final Evaluation Report, Gemini Computers, Incorporated – Gemini Trusted Network Processor, Version 1.01, 1995. www.aesec.com/eval/NCSC-FER-94-008.pdf

[17] M.R. Heckman, R.R. Schell and E.E. Reed, “A high-assurance, virtual guard architecture,” MILCOM 2012.

[18] G. Klein et al., “Comprehensive formal verification of an OS microkernel,” ACM Transactions on Computer Systems, Vol. 32(1), 2014.

[19] TGC, "TPM Main Specifications - Part 1 Design Principles Version 1.2 Rev.116," Trusted Computing Group, 2011.

[20] Infineon, “Infineon’s TPM Security Chips Are First to Receive Global TCG and Common Criteria Certification and UK Government Approval; Showing World Trust in Infineon Security Expertise for PC and Data Network Protection”, Press Release, December 8, 2009.

[21] H. Löhr, A.-R. Sadeghi, C. Stüble, M. Weber and M. Winandy, "Modeling Trusted Computing Support in a Protection Profile for High Assurance Security Kernels," in *Proceedings of TRUST 2009 - The 2nd International Conference of Trusted Computing*, Oxford, UK, 2009.

[22] J. Butterworth, C. Kallenberg, X. Kovah and A. Herzog, "BIOS chronomancy: fixing the core root of trust for measurement," Proceedings of CCS'13 - ACM Conference on Computer and Communications Security, Berlin, Germany, 2013.

[23] D. Cooper, W. Polk, A. Regenscheid and M. Souppaya, "Nist Special Publication 800- 147: BIOS Protection Guidelines," NIST, 2011.

[24] D. Grawrock, Dynamics of a Trusted Platform: A Building Block Approach, Intel Press, 2009.

[25] AMD, Secure Virtual Machine Architecture Reference Manual, AMD, 2005.

[26] R. Wojtczuk and J. Rutkowska, "Attacking Intel® Trusted Execution Technology," in *Black Hat DC*, 2009.

[27] R. Wojtczuk and J. Rutkowska, "Attacking Intel TXT via SINIT code execution hijacking," The Invisible Things Lab, 2011.

[28] ARM, ARM Security Technology: Building a Secure System using TrustZone® Technology, ARM Limited, 2009.

[29] OASIS eXtensible Access Control Markup Language (XACML) TC, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[30] V. C. Hu et al., “Guide to Attribute Based Access Control (ABAC) Definition and Considerations,” NIST Special Publication 800-162, 2014.

[31] A. Armando, M. Grasso, S. Oudkerk, S. Ranise, and K. Wrona, “Content-based Information Protection and Release in NATO operations,” 18th ACM symposium on access control models and technologies (SACMAT), 2013.