Andrzej Białas
Barbara Flisiuk

Institute of Innovative Technologies EMAG

IT SECURITY DEVELOPMENT PROCESS IN THE EXPERIMENTAL SECLAB DEVELOPMENT ENVIRONMENT

Introduction

The paper concerns one of the three major processes of the ISO/IEC 15408 Common Criteria methodology: IT security development process dealing with the analysis of an IT product security and the workout of security functions which are implemented in the product at its successive development stages. Particularly, the paper describes how this process is organized in the experimental laboratory SecLab of the EMAG Institute, i.e. on the basis of specially prepared patterns and with the support of a dedicated tool. SecLab makes use of products which were developed in the course of the CCMODE project (Common Criteria compliant Modular Open IT security Development Environment) [CCMODE] completed by the EMAG Institute.

The CCMODE project resulted in the following:

- patterns for constructing the elements of a development environment, including patterns of the so called evaluation evidences,
- a method to implement patterns while constructing the environment,
- software which supports the environment implementation process and manages this environment during its exploitation – CCMODE Tools,
- know-how necessary to implement and exploit the environment.

The efficiency and quality of the IT security development process can be significantly improved by design patterns, common knowledge source and specialized software which supports the development process.

CCMODE products were developed for businesses which construct their own development environments for IT products. The SecLab laboratory demonstrates how to build such environments and how to exploit them properly and efficiently.

Rigorous processes of the Common Criteria (CC) standard [CC1-3, CCPortal] have been stipulated to ensure the reliability of IT products. Here reliability has been replaced by a more adequate term – assurance. Assurance means that when a certain threat occurs, the IT product will be able to perform its security objectives. In other words, the security measures will work and will properly protect the attacked asset. Assurance is measured by Evaluation Assurance Levels (EAL), from EAL1 (min.) to EAL7 (max.). The reliability of the applied security measures is a matter of the utmost importance for IT products which are to be used in high-risk operational environments (with real, serious threats and high value of the processed information or provided IT services).

The paper is an extension of [BiaFli14] which featured the standards that are the basis of the SecLab laboratory, its organization, tools worked out within the CCMODE project and used for the development of IT products, tools and methods to protect data related to projects carried out in SecLab, and sample projects completed there.

The paper demonstrates how the IT security development process, the basic CC-methodology process, is carried out on the basis of patterns. The process comprises the analysis of the IT product usability, its operational environment and risk factors. This way security requirements can be worked out and implemented in the form of IT product security functions.

The paper gives brief characteristics of three basic processes of Common Criteria (section Processes of Common Criteria methodology). Against this background the subprocesses of the IT security development process are described (section Implementation of IT security development process in SecLab). In the conclusions, further steps are mentioned leading to the implementation of security functions in a concrete IT product.

Before reading the paper the readers are recommended to have a look at the publications [Hig10, Her03, Bia11a, Bia12] or other sources of knowledge about the Common Criteria methodology [CCPortal, CCMODE].

Processes of Common Criteria methodology

The processes of the Common Criteria methodology were extensively presented in the form of formal models in the monograph [Bia08]. Therefore the description provided in this paper is a shortened illustration of the issue. Here it is important to note that due to the assumed future evaluation and certification, the IT product is called TOE (Target of evaluation) in the nomenclature of the CC standard.

The CC methodology comprises three basic processes:

- IT security development process; after different security analyses there is a document prepared, called Security Target (ST);
- TOE development process, including the preparation of TOE documentation;
- IT security evaluation process carried out in an independent, accredited laboratory in a country which implemented the CC standard and signed the Common Criteria Recognition Arrangement (CCRA) [CCPortal].

According to the Common Criteria methodology, the Security Target document, which is worked out in the course of the IT security development process, defines the TOE security functions (TSFs). TSFs are in compliance with Security functional requirements (SFRs) identified for the IT product.

These functions are later implemented in the IT product during the TOE development process, in compliance with the EAL declared for the TOE. During the TOE development process there is quite exhaustive documentation produced, which is in accordance with the Security assurance requirements (SARs) [Bia14]. The documentation is an extension of the Security Target and plays a role of evaluation evidences produced for the purposes of the third process – IT security evaluation.

In this paper the authors focus on the first of the three above mentioned processes and its subprocesses. The process execution was presented in the experimental development environment of SecLab with the use of design patterns and a supporting tool described in [BiaFli14].

Implementation of IT security development process in SecLab

The process of developing a Security Target for the IT product (TOE) includes a number of activities specified in the Common Criteria documents. The key activities can be grouped into seven basic subprocesses: featured in Figure 1 and presented in the successive subchapters.

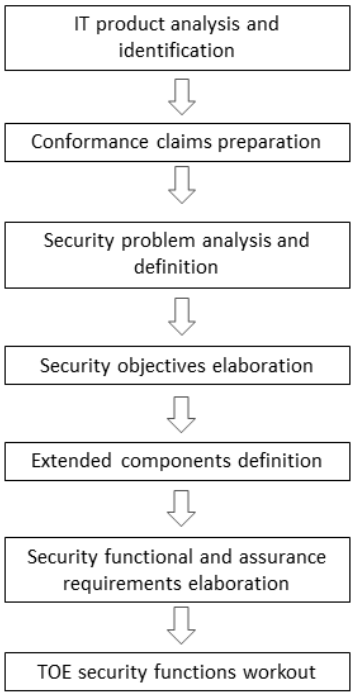

Fig. 1. IT security development process

Figure 1 presents the main succession of operations. In reality the process is carried out incrementally with possible returns to any subprocesses.

The Security Target document is a result of the IT security development process. The document structure and content are determined in [CC1-3]/part1 and in the components of the ASE (Security target evaluation) class [CC1-3]/part3.

Within the CCMODE project there was an STp pattern (Security target pattern) prepared for the IT security development process. The STp pattern has a form of an MS Word® template – Figure 2. It can be used with this editor or with the GenDoc application which is one of the main modules of CCMODE Tools. This toolset broadly supports the Common Criteria methodology, including the IT security development processes [Rog14].

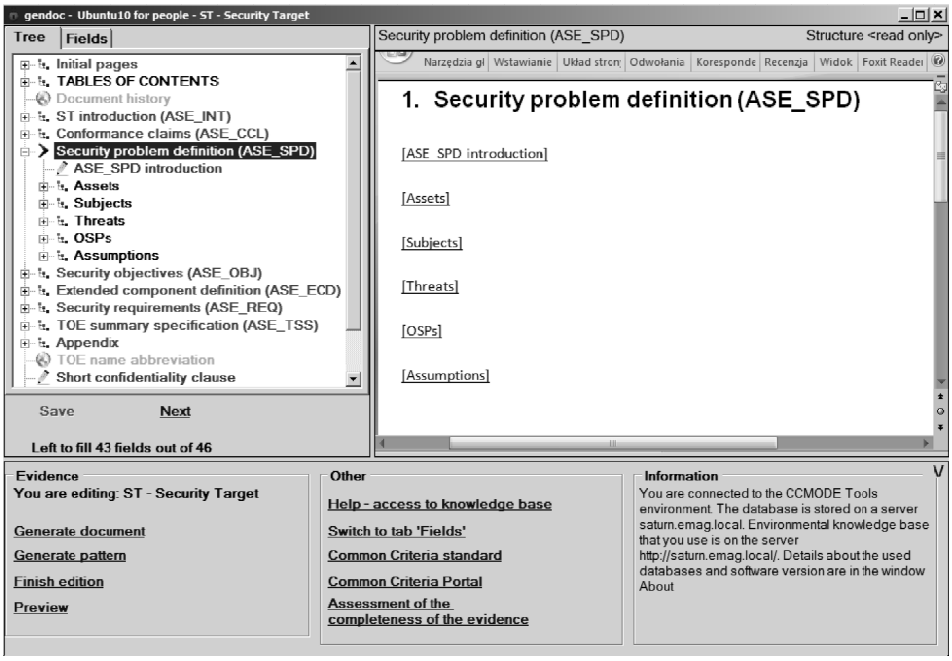

Fig. 2. Security Target pattern in CCMODE Tools GenDoc application

Source: EMAG's documentation, 2014.

On the left side of Figure 2 there is a hierarchical, formalized structure of the ST document. For the highlighted Security problem definition, its basic sections were presented in the right window. During the IT security development process all elements of the structure are filled with content about the developed IT product (TOE). In order to define the content, it is necessary to conduct a number of more or less complex analyses and rationales. Some of them require many knowledge sources and specialized supporting tools.

In the bottom part of the GenDoc application window (Figure 2), its quick-access functions are shown, including the access to external knowledge sources and the knowledge base of CCMODE Tools. While working on the ST and other documents, the developer gets some hints how to present a given issue in compliance with the CC standard and sometimes even ready-to-use phrases are prompted.

To carry out formalized tasks the developers have at their disposal a tool in the form of the Enterprise Architect® – EA plugin. It helps to model in the UML language and solve complex security issues in the realm of the Common Criteria methodology.

Figure 3 presents a part of the security model (see subsections: Security problem analysis and definition, Security objectives elaboration) of an intelligent sensor for mining.

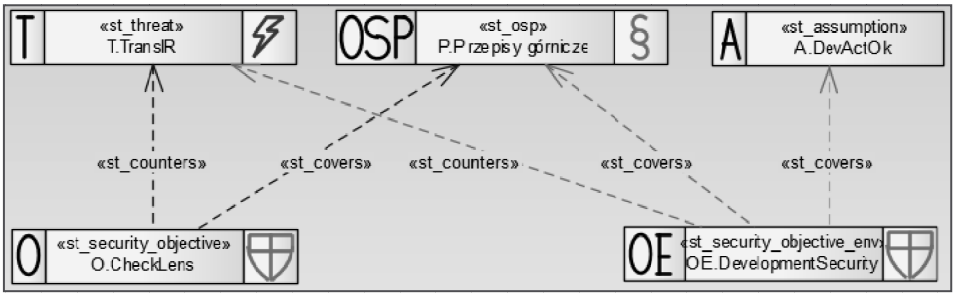

Fig. 3. IT security development process

Source: EMAG's documentation, 2014.

Please note the stereotyped UML classes representing threats (T), organizational security policies (OSP), assumptions (A), security objectives for the TOE (O), security objectives for the TOE environment (OE), and relationships between them. All security analyses are conducted with the use of the above mentioned EA plugin. The results of these analyses are transferred, with the help of the GenDoc application, to the Security Target fields. A part of the content is worked out by the developer, other is collected from the database of the tool.

In the SecLab laboratory the IT security development process is performed by developers with the use of CCMODE Toolset [BiaFli14, Bia12, Rog14]. The basic applied pattern is the Security target pattern (STp). This pattern is indispensable when the Security Target is prepared from scratch, on the basis of the user's requirements (a typical development path). The remaining patterns listed in section 4 of [BiaFli14] can be applied in special cases.

Below there is a simplified presentation of seven subprocesses of the IT security development process. The subprocesses are performed on the basis of the STp pattern.

IT product analysis and identification

The first subprocess comprises the analysis of the IT product and working out its informal description according to the CC requirements.

First the developer compiles different kinds of auxiliary management information for the whole project (IDs of the project and TOE, dates, versions, au-

thors, etc.) and then prepares a section of the Security Target called TOE overview. This section helps potential clients, who go through the list of evaluated products [CCPortal], to check whether the developed TOE will meet their requirements and whether it will be compatible with the hardware and software they use. The TOE overview includes:

- the use and major security features of the TOE – presents the possibilities of the TOE in terms of security and how to use them, in a language friendly to potential clients, e.g. for a firewall project: "MyFWL Firewall, version 1.9, enables to control the movement of packages between a public network and a protected private network on the level of IP address and port numbers. The firewall also serves as a proxy. It has embedded mechanisms for events registration and for management by the administrator";
- TOE type according to the general IT product taxonomy [CCPortal], e.g.: firewall, VPN-firewall, smart card, crypto-modem, intranet, web server, database, web server and database, LAN, LAN with web server and database, etc.;
- required non-TOE hardware/software/firmware – specifies hardware and software which should work in the TOE environment, e.g. if the TOE is an application these can be minimal requirements about the computer and the operating system on which this application works.

More detailed information for the evaluators and potential users concerning the structure and possibilities of the TOE are placed in the next section – TOE description, which is focused on two issues:

- physical scope of the TOE – specification of hardware elements, software, firmware and documentation which make up the TOE; it is important to point whether the given element is a part of the TOE or the TOE environment;
- logical scope of the TOE – features/functionalities of the TOE related to security and described as logical components; it is necessary to demonstrate whether or not the given feature/functionality belongs to the TOE.

The above results of the IT product analysis are placed in the "ST Introduction" section of the Security Target as identifiers and informal descriptors.

Conformance claims preparation

Conformance claims express the IT product compliance with the proper version of the CC standard and with security requirements contained in protection profiles (previously evaluated set of requirements the given Security Target must comply with). More importantly, however, conformance claims declare the EAL for the developed IT product. The developer should prepare proper declarations. The TOE de-

velopment process will be carried out according to the rigour determined in the Security assurance components (SARs) for the declared EAL [Bia14].

Security problem analysis and definition

Once the IT product is defined in terms of its usability, the developer performs main operations of the IT security development process. They begin from the analysis of the IT product security, which is to prepare a section of the Security Target called Security Problem Definition (SPD). SPD can have an informal or semiformal character. In the latter case, which is a more precise one, the so called generics are used as specification means. Generics are also applied in the successive subprocesses: to specify security objectives and TOE security functions. The features of generics are equal to semiformal SFR and SAR components. The issue of specification means, including generics and their models in UML and OWL, is exhaustively discussed in [Bia11a, Bia11b, Bia10a, Bia10b, Bia10c, Bia09].

The first operation of the SPD subprocess is the identification of assets. According to the Common Criteria methodology, the TOE contains two groups of assets:

- users' assets, such as: memory, electronic media, external devices, transmission lines and devices with their bands, computing power, etc.; these assets are used in production, processing and information transfer,
- assets which ensure the TOE security, related to its security functions (TSF), e.g. authentication data, cryptographic keys, secrets, assets attributes.

Elementary assets have two forms:

- active entities, i.e. the assets which initiate operations inside the TOE or on TOE information; they are called subjects (here: Sxx),
- passive entities, i.e. the assets which are the only source from which information is taken or serve as a place for storing information; they are called objects or "data and other assets" (here: DAx) and are the target of operations initiated by subjects.

Assets protected by the TOE can be placed inside or outside the TOE.

The following generic is a sample elementary description of a protected asset:

DAE.ProtNet. Hosts, workstations, their data and services on the private network protected by the firewall.

The asset is defined by means of the mnemonics of the ProNet generic (Protected network). The mnemonics is specified in the generic description (the text following the dot). The type of the generic – DAE prefix – shows that the protected assets are placed in the TOE environment.

For assets, authorized (SAU) subjects are identified, e.g.:

SAU.FullAccAdmin. TOE administrator, having full access rights.

or unauthorized (SNA), e.g.:

SNA.HighPotenIntrud. Intruder having high level skills, enough resources and deep motivation to perform a deliberate attack.

Another operation, the key one of SPD, is identification of threats and their description by means of threat generics (Txx):

TDA.IllegAcc. An attacker [Sparam <= SNA.HighPotenIntrud] on the hostile network may exploit flaws in service implementations (e.g. using a 'well known' port number for a protocol other than the one defined to use that port) to gain access to hosts or services [DAparam <= DAE.ProtNet].

The TDA prefix refers to all threats which are direct attacks. In the description of the generic there are two parameters: Sparam and DAparam. They represent, respectively, the generic which describes the subject: *SNA.HighPotenIntrud* (here the so called threat agent) and "data and other assets": *DAE.ProtNet* (here assets protected by the firewall).

The security problem can be expressed by means of threats specification for assets (as above) or by means of Organizational Security Policies (OSPs), which are expressed by generics too. While specifying threats or OSPs, certain assumptions are made (Axx) for the TOE environment, connections, users and their behaviours – also expressed by generics, e.g.:

AC.DualHomed. The firewall has separate network adapters for all network connections.

The DualHomed generic, concerning connectivity aspects (AC), describes the general structure of the firewall.

SPD is the result of the subprocess described here. It presents, in a concise and coherent way, the security problem for the developer to solve.

Security objectives elaboration

Another subprocess is security objectives elaboration whose target is to solve the security problem and present the solution in the form of a security objectives set. This can be done in an informal way, however, a more precise, semi-formal approach is recommended with the use of generics to specify security objectives. The security objectives define future security measures in an abstract manner and are formulated for the TOE – the O subset (TOE responsibility for solving the elementary security problem) or for the environment – the OE subset (environment responsibility). Many categories of objectives were defined in relation to the applied groups of security measures.

Here is an example of an objective which solves an elementary problem TDA.IllegAcc:

OACC.LmtIPAddr. The firewall enforces access control by limiting the valid range of addresses expected on each of the private and hostile networks (i.e. an external host cannot spoof an internal host).

The OACC prefix represents a category of objectives related to access control. It is formulated for the TOE, which means that its access control mechanism will be implemented in the TOE as one of its TSFs. This objective is supported by another one – for the TOE environment:

OSMN.SecConfManag. Security-relevant configuration management. Managing and updating system security policy data and enforcement functions, and other security-relevant configuration data, in accordance with organizational security policies.

Its performance will take place outside the TOE (it will not be specified to the form of a TOE security function). The OSMN prefix expresses a category of objectives which refer to security management.

The objectives have to solve fully the previously identified security problem (SPD). All SPD elements have to be covered by objectives (the problem solved) and neither of the objectives is redundant (the solution is effective). Each of such facts has to be justified.

The subprocess results in a coherent and justified security objectives specification with the objectives divided into those fulfilled by the TOE and those by the TOE environment.

Extended components definition

It is possible for the developers to define their own components provided that their form is in compliance with the one given by Common Criteria. This rare situation occurs when neither of the components described in the standard is able to express specific needs of the developer. For example, applications which generate cryptographic keys need a generator of random numbers. In the CC catalogues there are no proper SFR components for this generator to assess the quality (entropy) of the generated random numbers. If such a component needs to be used in the ST specification, it has to be defined first. This is the objective of the subprocess described here.

Security functional and assurance requirements elaboration

Security objectives represent elementary solutions to security problems. They can be expressed in an informal or semiformal (more precise) way. However, they will be always expressed in the natural language of developers, which can be interpreted quite freely. That is why it is necessary to translate this specification into a unified language defined in Common Criteria, i.e. to express security objectives by means of functional components – security functional requirements (SFRs).

For example, the TOE objective OACC.LmtIPAddr meets the requirements expressed by two functional components:

FDP_ACF.1 Security attribute based access control.

FDP_ACC.2 Complete access control.

The FDP class of functional components described in [CC1-3]/part2 signifies User data protection, while its family FDP_ACF describes access control functions and FDP_ACC – access policy rules.

The identified SFR components can have their dependent components which need to be analyzed too, either attached to the basic ones or rejected (with justification).

To put it simply, the subprocess of security functional and assurance requirements elaboration is equal to finding an SFR for each TOE security objective. The specification needs precise justification. The choice of SAR components results from an arbitrarily declared EAL.

TOE security functions workout

The last subprocess of IT security development is TOE security functions workout. The developer defines a set of TOE security functions to be implemented in the IT product according to security assurance requirements complying with the declared EAL. In the method applied here, SFR requirements are grouped around security objectives related to the TOE, conflicts and overlappings are solved, common parts of groups are identified and, on this basis, a set of TOE security functions is formulated.

For example, for the firewall project [Bia08, Bia09] the following six TSFs were identified:

TSF.LmtIPAddr. Function responsible for IP address control between hostile and protected networks, using: apparent source IP address or host name and destination IP address or host name.

TSF.LmtPortHost. Function responsible for port number control between hostile and protected networks, using apparent source port number and destination port number.
TSF.OnProxyAuth. Function responsible for authentication of the end user prior to establishing a through connection for specified services.
TSF.AdminAuth. Function responsible for the firewall administrator access control, ensuring that only authorized [Sparam <= SAU.FullAccAdmin] are able to access the firewall functionality. The detailed functionality: system login (identification, authentication), administrator accountability, logout.
TSF.Audit facilities. Function responsible for recording security related events and its management for audit purposes. These events may concern: IP addresses limitation, port number limitation, users' authentication on proxy for the selected network services, administrator's login, operations, and logout.
TSF.FirewallManagement. Function responsible for effective management of the TOE and its security functions. The firewall administrator [Sparam <= SAU.FullAccAdmin], and only the firewall administrator, can perform the following functions: display and modify the firewall access control parameters, initialize and modify user authentication data, display and modify user attributes, select events to be audited, identify the subset of auditable events deemed to indicate a possible or imminent security violation, associate separate authentication mechanisms with specific authentication events, verify the integrity of the firewall.

The TSF specification is the final phase of the security development process which results in the preparation of the Security Target.

Conclusions

The paper provides extended information about the Common Criteria methodology and its implementation in the SecLab laboratory described in [BiaFli14]. The paper is focused on one of the three main processes of the CC methodology, i.e. the IT security development process. The objective of this process is to analyze the security of the IT product and to work out security requirements for it. These security requirements are expressed in the form of TOE security functions. The result of the IT security development process is the Security Target document. TSF functions behave like black boxes whose contents will be worked out during the TOE development process. The result of this process will be a detailed project of the IT product. As this is quite an exhaustive issue, it will be presented in a separate work [Bia14].

SecLab is an experimental environment for the development of IT products in compliance with the Common Criteria methodology. SecLab was equipped with patterns of processes and documents, implemented in the CCMODE Tools specialized software. The objective of these operations is to improve the quality and efficiency of the development process and, this way, to increase chances for positive certification of the IT product.

References

[Bia11b] Białas A.: Common Criteria Related Security Design Patterns for Intelligent Sensors − Knowledge Engineering-Based Implementation. Sensors 2011, 11, 8085-8114, available at: http://www.mdpi.com/1424-8220/11/8/8085/.

[Bia10a] Białas A.: Common Criteria Related Security Design Patterns − Validation on the Intelligent Sensor Example Designed for Mine Environment. Sensors 2010, 10, 4456-4496, available at: http://www.mdpi.com/1424-8220/10/5/4456.

[Bia10b] Białas A.: Intelligent Sensors Security. Sensors 2010, 10, 822-859, available at: http://www.mdpi.com/1424-8220/10/1/822/.

[Bia10c] Białas A.: Patterns-based Development of IT Security Evaluation Evidences. The 11th International Common Criteria Conference, Antalya, 21-23 September 2010 (published in an electronic version), http://www.11iccc.org.tr/presentations.asp.

[Bia14] Białas A.: Common Criteria Compliant IT Product Development in the Experimental SecLab Laboratory. In: M. Pańkowska, J. Palonka, H. Sroka (eds.): Ambient Technologies and Creativity Support Systems. Uniwersytet Ekonomiczny, Katowice 2014.

[Bia12] Białas A. (ed): Komputerowe wspomaganie procesu rozwoju produktów informatycznych o podwyższonych wymaganiach bezpieczeństwa (Computer Support for the Development of IT Products of Enhanced Security). Wydawnictwo Instytutu Technik Innowacyjnych EMAG, financed by UE POIG 1.3.1, Katowice 2012.

[Bia08] Białas A.: Semiformal Common Criteria Compliant IT Security Development Framework. „Studia Informatica" 2008, Vol. 29, No 2B(77), Silesian University of Technology Press, Gliwice.

[Bia11a] Białas A. (ed.): Zastosowanie wzorców projektowych w konstruowaniu zabezpieczeń informatycznych zgodnych ze standardem Common Criteria (Design Patterns for the Development of IT Security in Compliance with Common Criteria). Wydawnictwo Instytutu Technik Innowacyjnych EMAG, financed by UE POIG 1.3.1, Katowice 2011.

[Bia09] Białas A.: Validation of the Specification Means Ontology on the Simple Firewall Case. In: H. Arabnia, K. Daimi (eds.). Proceedings of the 2009 International Conference on Security and Management (The World Congress in Applied Computing – SAM'09: July 13-16, Las Vegas, USA), Vol. I, 2009, Publisher: CSREA Press, pp. 278-284.

[BiaFli14] Białas A., Flisiuk B.: Specialized Development Environments for Security-enhanced IT Products and Systems. In: M. Pańkowska, J. Palonka, H. Sroka (eds.): Ambient Technologies and Creativity Support Systems. Uniwersytet Ekonomiczny, Katowice 2014.

[CCMODE] CCMODE (Common Criteria compliant, Modular, Open IT security Development Environment). http://www.commoncriteria.pl/.

[CC1-3] Common Criteria for IT Security Evaluation, part 1-3. v. 3.1.2009.

[CCPortal] Common Criteria Portal. http://www.commoncriteriaportal.org/, 2014.

[Her03] Hermann D.S.: Using the Common Criteria for IT Security Evaluation. CRC Press: Boca Raton, FL, USA, 2003.

[Hig10] Higaki W.H.: Successful Common Criteria Evaluation. A Practical Guide for Vendors. Copyright 2010 by Wesley Hisao Higaki, Lexington, KY 2010.

PROCES ROZWOJU BEZPIECZEŃSTWA IT W EKSPERYMENTALNYM ŚRODOWISKU ROZWOJU SECLAB

Streszczenie

Artykuł zawiera krótką charakterystykę trzech podstawowych procesów podejścia Common Criteria. Autorzy opisują, jak proces rozwoju bezpieczeństwa technologii informacji jest zorganizowany w eksperymentalnym laboratorium SecLab w Instytucie EMAG. Badany proces obejmuje analizę użyteczności produktu informatycznego, opis jego środowiska działania i czynniki ryzyka. Prowadząc analizę w ten sposób, autorzy opracowują specyfikację wymagań i dokonują wdrożenia funkcji bezpieczeństwa produktu informatycznego. W części obejmującej wnioski końcowe autorzy przedstawiają dalsze kroki wdrażania funkcji bezpieczeństwa w odniesieniu do konkretnych produktów informatycznych.