INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2020, VOL. 66, NO. 3, PP. 499-505
Manuscript received February 23, 2020; revised July, 2020. DOI: 10.24425/ijet.2020.134005

Security Assessment Methodology for Isolated Systems of a Certain Class

Daniel Wiechecki

***Abstract*—This paper introduces security assessment methodology for isolated single-workstation multilayer systems processing sensitive or classified data according with a corresponding security model for such system. The document provides a high-level tool for systematizing certain-class-systems security models development. The models based on the introduced methodology cover data confidentiality and availability attributes protection on a sufficient level.**

***Keywords*—cybersecurity, security assessment methodology, security model, Graham-Denning model, Bell-La Padula model, Clark-Wilson model**

I. Introduction

THIS paper introduces security assessment methodology for isolated single-workstation multi-layer systems processing sensitive or classified data according with a corresponding security model for a representative computer system. The document provides a high-level tool for systematizing certain-class-systems security models development. The methodology is an adaptation of the general procedure for building classified IT systems described in legal documents (i.e. in Poland: [1],[18]) and standards (i.e. [13]).

The main security attribute protected in systems modelled using introduced algorithm is confidentiality of data processed in the system. Nonetheless, availability (on a sufficient level) and integrity (on a basic level) of the data is also regarded in the models. In the methodology, as a basis for considerations Graham-Denning [11], Bell-La Padula [4],[5] and Clark-Wilson [8] models were adopted. In the course of the article the alternative approaches are mentioned.

The main issues presented in the methodology are:

- Idea of abstract system entities – system layers, being the structures containing set of objects and linked entities collections. The layer-driven attitude towards the modelled systems definitions is shown in the methodology to prove the layers utility and facility.
- Description of security level determination recursive method based on probability of beating penetration path, being a set of activities that adversary must perform in order to penetrate the layer in an unauthorized manner.
- Concept of describing dynamics of the modelled systems via state graph definition – based on layers conception mixed with Clark-Wilson [8] model axioms utilization.

This work was presented at the International Scientific Conference Mathematical Cryptology & Cybersecurity (MC&C 2020), Warsaw, 16-17.01.2020.

The Author is a graduate and research project participant on Military University of Technology (e-mail: daniel.wiechecki@gmail.com).

The contents of the paper are strictly associated with virtual-machines–based security systems research conducted on Military University of Technology and constitutes author's master thesis research continuation.

II. Considered systems class

Systems class considered in the paper is defined by the following statements and conditions:

- Sensitive or classified data, protected in the system is in the form of files.
- The purpose of the system remains undefined.
- Types, formats nor content of the files are relevant – in order to increase the level of generality (hence – level of adequacy to the reality, regardless of the purpose of the system).
- System consists of a single workstation (in general it is not relevant as, in case of system consisting of more than one workstation, every workstation inherits the security scheme).
- User environments are set as virtual machines in a host operating system. The virtual machines are visible as file system items.
- At least two independent security layers are implemented in the system (defense-in-depth rule).
- Cryptographic protection mechanisms are used.
- Access to objects without proper access rights is prohibited in the system.
- Subjects allowed to access objects in "lower" system layers must be provided access to all the layers in between ("higher" layers).
- The system is isolated from external IT networks and devices.
- Data exchange is executed by recording data on removable media, only by authorized subjects.
- It is forbidden to update system hardware or software configuration – except for emergencies or software errors.

III. Security assessment methodology

Considerations included in this Section describe the global system definition. The definition contains static system issues, independent on the system's current state. Nevertheless, it is necessary to define each of the model elements described below to enable the possibility of system's dynamics considerations.

A. Security system and acceptable security level definition

Security system definition is a high-level perspective of security policies. Formally, it is represented as a tuple of defined system elements, and rules in the system as well as the conditions under which the system may be considered secure.

One of the mandatory conditions to be stated in the definition is the acceptable security level. It is a value or set of values representing the acceptable probability of beating penetration paths. Despite of the fact that the probability of beating "the deepest" layer is the most crucial value in the acceptable security level, it is recommended to define probability values for each of the defined penetration paths.

B. Classes, categories and sensitivity levels structure definition

Class is a legally required (data clause / security clearance) or subjective label representing security measures needed to be implemented when it comes to particular entity. It is required to define a linear order relation on the classes set. The defined labels structure is considered an organizational data availability limitations mechanism.

Category is a label informing about data domain or insight privileges. The set of categories is defined in order to restrict access to data according to the principle of least privilege. In multi-domain data processing systems it is recommended to implement a tree structure defining parent-child domains relationship.

Sensitivity level structure is a generic data structure binding classes and categories. The structure's elements should uniquely define privileges needed to access to the entity of a certain domain – category. It is required to define an order relation on the sensitivity level structure, similarly to [4]-[6].

C. Subjects, objects and layers definition

Objects are entities used, stored or processed in the system. It is recommended to define only objects essential for system to work according to its purpose to lower the model's complexity.

Objects are not granted organizational access rights in the model. The organizational rights are not identical with technical access rights (such as file system accessibility in operating system).

Layers are abstract entities grouping set of objects and vulnerabilities, threats and security measures associated with the objects. The purpose of layers in the model are to emphasize and include multi-layer system structure in mathematical considerations, providing higher adequacy to the reality. Similarly to objects, layers are not granted organizational access rights in the model.

Layers definition, separate from objects definition, leads to facilitation of the model. From the practical point of view, the defined layers may also be considered objects. However, due to inconsistencies in "layer" objects and "casual" objects definitions it is easier to create another structure – similar to equivalence class with a relation of sharing same threats being applicable the same security measures.

Subjects are entities identical to roles implemented in the system (they are not identical to the people working in the system). It is not forbidden to assign multiple roles to a single person, however, it is recommended to follow the principle of least privilege.

Subjects' organizational rights imply the need of technical access rights implementation on modelled objects and layers in the system. The technical access rights may be considered security measures in a model.

D. Objects to layers assignment

Each object must be assigned to a layer. The assignment is the projection of a modelled entities' dependencies in a modelled system. It is not possible to assign an object to multiple layers. Depending on the system's configuration, it is allowed to store copies of object in distinct layers, however, the case is not considered in the paper.

E. Sensitivity levels to subjects and objects assignment

It is recommended to follow assignments defined in [5]:

- Each object is assigned to a single sensitivity level, which determines subject's sensitivity level required to gain access to the object.
- Each subject is assigned two sensitivity levels:
  - Current sensitivity level, which determines permissions to objects in current subject's work session.
  - Authorize level, which determines maximum current sensitivity level subject can be assigned.
- Each layer is implicitly assigned a sensitivity level, equal to the greatest from sensitivity levels of the objects within the layer.

F. Access rights structure definition

It is recommended to define subjects' access rights to objects and layers exclusively. Based on the definition it is possible to restrict access to particular objects in a layer and allow users to access the object in a "lower" layer without granting privileges to objects in transitional layers. The access rights structure must be dependent on the purpose and configuration of the system.

There are many approaches towards access rights' structure definition. The most common are:

- Matrix structure – as introduced in i.e. [5],[6],[11].
- Access Control List structure – as introduced in i.e. [3].
- Role-Based Access structure – as introduced in i.e. [3],[14].
- Lattice structure – as introduced in i.e. [9],[19].

G. Integrity verification and transformation procedures definition

It is recommended to follow the certification and enforcement rules defined in [8]. Furthermore, it is mandatory to define integrity policy in the system, which, in particular, implies the way users actions are logged in the system. [20]

It is advised that the integrity verification and transformation procedures are assigned to the particular layers. The assignment prevents from organizational system deadlocks, as each one of the layers may require unique internal procedures, unable to execute in a different layer.

As the alternative approach, it is possible to use integrity policies described in Biba integrity model. [6] However, the integrity axioms introduced by Biba juxtaposed with confidentiality axioms from [5] may lead to contradiction in most of the modelled systems.

H. Vulnerabilities identification and threats structure definition

It is recommended to identify system's vulnerabilities based on a state graph. [20] This approach provides clear view on dynamic changes in the system. The defined vulnerabilities imply threats definitions (threats are considered practical ways of exploiting identified vulnerabilities). The state graph may be

considered a mathematical way of applying the approach introduced in i.e. [16]

It is required that defined threats structure contains probability values of a threat execution in a layer or in reference to a subject. Threat execution probability for a given layer is applicable for every object within the layer. For certain systems of the considered class it may be necessary to additionally define another threats structure - probability values of threats execution in reference to system objects, and bind it with the required structure.

An alternative approach towards vulnerabilities identification is based on the game theory. The approach has been introduced in, i.e. [2],[7],[15],[17]. The approach may be considered complementary for systems with more complicated data flow implemented.

I. Security measures for identified threats identification and security measures structure definition

It is recommended to create security measures structure definition identical with threats structure definition. This approach results in a convenient merge of the structures, which is mandatory to compute the model output.

It is required to check whether implemented security measures do not expose the system to new vulnerabilities. If any new vulnerability is identified, it is necessary to apply another security measures or reconsider existing ones.

J. Penetration paths definition

Penetration path is a set of activities that adversary must perform in order to penetrate the layer in an unauthorized manner. The approach adopted in the paper is that each layer has a single penetration path associated – the penetration path is a set of the most probable malicious activities, dynamically changing depending on implemented security measures. Therefore, it is recommended to associate penetration paths with a distinct layers and recursively bind penetration paths associated with following layers, so that

- defense-in-depth rule is modelled in a way adequate to the reality,
- it is possible to easily identify layers in need for security improvements.

K. Output computation

The output is computed using the defined penetration paths, given the conditions stated in security system definition. If the output values are greater than defined in acceptable security level it is mandatory to repeat security measures identification.

IV. System dynamics considerations

As mentioned in Sec. III, the described methodology focuses on system issues independent on system's current state. However, as mentioned in Sec. III.G and III.H, it is recommended to consider the system dynamics to define and identify integrity policy and vulnerabilities.

The below considerations show one of the possible approaches towards describing the modelled system's dynamics.

A. Active sub-model

Active sub-model is a section of a defined model associated with the possible states that a subject currently working can reach during its current work session. In a single session the subject must be assigned the current sensitivity level – not greater than its authorization level. The objects' set must be limited to the elements not greater than subject's current sensitivity level, which implies that the layers and layer-associated structures contents must be limited respectively.

It is required that neither of layers, integrity verification procedures, transformation procedures nor penetration paths structures have been changed in the sub-model definition, according to the global model. Moreover, it is prohibited to change any of the model's elements structure during being used in an active system session.

Given the above considerations, the sub-model consists of the following elements and structures:

- Subject currently working in the system with a current sensitivity level assigned.
- Subset of objects set limited to the ones the subject may have access to during the current system session.
- Substructure of access rights connected with the subject and the subset of active objects.
- Unchanged set of layers. The contents of layers may be altered, but it forbidden to remove any of the layers from the origin set.
- Substructure of threats and security measures connected with the subset of objects (if existing). The structure of threats and security measures connected with defined layers must remain unchanged.
- Unchanged structure of integrity verification and transformation procedures.
- Unchanged set of penetration paths.

B. State graph

State graph consists of nodes defined as a tuple of the following elements:

- Subject currently working in the system.
- Layer the object is currently working in.
- Subset of the active objects set adherent to the current layer.
- Substructure of the subject's current access rights tied with the layer and subset of the active objects.
- Substructure of threats bound with the current layer.
- Substructure of security measures bound with the current layer.
- Substructure of integrity verification procedures connected with the subset of the active objects and the current layer.
- Substructure of transformation procedures connected with the subset of the active objects and the current layer the subject has access rights to execute.

Current state change is possible only via executing the valid transformation procedure and after the state verification via integrity validation procedures. Due to the above, it is necessary that the state graph is a directed graph. The edges of the graph should be labeled with the subsets of procedures required to be executed in order to change the system state.

C. Vulnerabilities identification based on a state graph

Having defined the system's state graph it is convenient to identify vulnerabilities for each state. The identification is based on negating the states restrictions and checking whether defining the possibility of breaking the restrictions. Any of the

below scenarios in the system would be considered a vulnerability:

- Subject change during an active system session.
- More than a single subject during an active system session.
- The valid subject's current sensitivity level change.
- Active objects set modification without proper access rights.
- Layers set modification.
- Integrity verification procedures subset modification.
- Transformation procedures subset modification.

D. Integrity policy definition based on a state graph

As mentioned in Sec. IV.B, the system state change is dependent on predefined procedures. The procedures must maintain and validate the system's integrity. However, the above approach is not strong enough to secure data integrity. Suppose an adversary reaches a state in a state graph. The adversary exploits the state's integrity by bypassing or disabling procedures required to change the system state to any of the child states. If the adversary exploits data or layer integrity, there is a great possibility that all the data in "lower" system layers becomes corrupted or destroyed.

The mentioned situation implies that integrity protection in multi-layer systems must be strictly policed or inversely defined to implement sufficient security measures. Based on an inverse definition, a separate model is formed, therefore, in order to utilize the model corresponding to the described methodology it is necessary to thoroughly define the integrity policy.

It is strongly recommended to implement cryptography-based verification procedures in addition to access control mechanisms preventing from harmful and malicious data modification. Moreover, as the most complicated issue in systems of the considered class, it is necessary to implement a system event log. The below mechanisms may be considered:

- Local event log
  Saving logs within the closed environment and retrieving them periodically as the administrator's integrity maintenance procedure. The logs container should be placed in "the highest" layer to ensure logs availability. The proposition is inconsistent with confidentiality axioms [5].
- Remote logging environment
  The proposition requires implementing internal secure network, therefore remains inconsistent with the systems class assumptions.
- No logging in "lower" layers
  The proposition could be valid only if non-administrator users were not able to produce any materials in the system. Then the users environments should be destroyed without unsealing them at any time. This logging policy requires significantly greater availability protection mechanisms.

As shown above, the proposed solutions are vulnerable to many exploits – therefore insufficient to ensure data integrity protection singlehandedly.

V. Security model of the defined systems class representative system

The model defined in this Section is compatible with the methodology described in Sec. III. The model is based on [4],[5],[8],[11],[20].

A. Security system and acceptable security level definition

Security system *SS* is defined as a tuple of system elements definitions

$$SS = \{L, S, O, W, U, TP, IPV, ZG, ZB, SP\}, \quad (1)$$

where:

- L – sensitivity levels structure;
- S – subjects set;
- O – objects set;
- W – layers set;
- U – access rights structure;
- TP – transformation procedures set;
- IVP – integrity verification procedures set;
- ZG – threats structure;
- ZB – security measures structure;
- SP – penetration paths set.

Let P_{ACC} denote the sequence of acceptable security level values. The sequence elements are defined as following:

$$\begin{cases} P_{ACC_n} = P_{ACC_{n-1}} * \Delta p_n \\ P_{ACC_0} = 1 \end{cases}. \quad (2)$$

The values are the maximum acceptable probability of beating each of the penetration paths. $\Delta p_n \in \langle 0,1 \rangle$ is considered a relative maximum acceptable probability of beating SP_n, given SP_{n-1} as a reference point. Δp_n is a constant value, set based on system's security requirements.

The system is considered secure when

$$\underset{SP_n \in SP}{\forall} \left(P(SP_n) \le P_{ACC_n} \right). \quad (3)$$

Function $P : SP \to \langle 0,1 \rangle$ returns actual probabilities of beating the penetration paths. [20] The function is defined in section V.K.

B. Entity classes, entity categories and sensitivity levels structure definition

Let C denote classes set. A linear order relation „≥" is specified on the classes set.

Let K denote categories set. An inclusion relation „⊇" is specified on the set.

The sensitivity levels structure L is defined as following:

$$L = \{ \lambda = (c,k) : c \in C, k \in K \}. \quad (4)$$

A „domination" relation „≥" is specified on the structure

$$(\lambda_i \ge \lambda_j) \Leftrightarrow (c_i \ge c_j \wedge k_i \supseteq k_j),$$
$$\text{where } \lambda_i = (c_i, k_i) \wedge \lambda_j = (c_j, k_j). \quad (5)$$

The domination relation is a partial order relation. [4],[5]

C. Subjects, objects and layers definition

Let S denote subjects set, O - classes set and W layers set. The relation of "containing" layers within each other is denoted by „$\succ$". Layers set W is defined as follows:

$$W = \{W_1, W_2, W_3, W_4, W_5\}, \quad (6)$$

$$W_1 \succ W_2 \succ W_3 \succ W_4 \succ W_5, \text{ where:} \quad (7)$$

- W_1 is the physical layer, including workstation;
- W_2 is the host operating system layer;
- W_3 is the virtual machines layer;
- W_4 is the virtual machines operating systems layer;

• W_5 is the protected data layer.

In the presented case the workstation and the operating system are considered layers, not objects.

D. Objects to layers assignment

The relation of object assignment to a layer is denoted by „⊇". The subset of objects belonging to layer *W* is denoted O_w:

$$\bigvee_{w\in W}\bigvee_{o\in O}\left(o\subseteq w\Rightarrow o\in O_w\right). \tag{8}$$

Each object must be assigned to a layer

$$\bigcup_{w\in W} O_w \in O. \tag{9}$$

Each object belongs only to one layer

$$\bigvee_{w,v\in W, w\neq v}\left(O_w\cap O_v=\varnothing\right). \text{ [20]} \tag{10}$$

E. Sensitivity levels to subjects and objects assignment

For each subject two sensitivity levels are assigned – authorization level and current sensitivity level. Current sensitivity level cannot be dominate (in a relational sense) the authorization level. For each object there is one sensitivity level assigned. [5]

F. Access rights structure definition

Let U_O denote access rights to objects set.

$$U_O=\{r,w,a,e,g\}, \tag{11}$$

where:

• *r* – access right to read an object;
• *w* – access right to write to an object (with automatic *r* right);
• *a* – access right to append to an object (without *r* right);
• *e* – access right to execute an object;
• *g* – access right to grant access rights to the object. [4],[5],[11]

Let U_W denote access rights to layers set.

$$U_W=\{ew,co,ww,gw\}, \tag{12}$$

where:

• *ew* – access right to enter the layer;
• *co* – access right to create new object in a layer;
• *ww* – access right to execute assigned object access rights;
• *gw* – access right to grant access rights to the layer.

Access right structure *U* is defined by two matrices, constructed upon u_O and u_W functions:

$$\begin{cases} u_O: S\times O\rightarrow 2^{U_O} \\ u_W: S\times W\rightarrow 2^{U_W} \end{cases}. \tag{13}$$

G. Integrity verification and transformation procedures definition

Let *IVP* integrity verification procedures set and *TP* – transformation procedures set. The sets' elements are compatible with Clark-Wilson model rules. [8]

The *IVP* and *TP* sets are split into subsets regarding the layer the elements are affecting:

$$\begin{cases} \bigcup_{w\in W} IVP_w = IVP \\ \bigcup_{w\in W} TP_w = TP \end{cases}. \tag{14}$$

It is possible to distinguish three transformation procedures subsets:

• "Internal" transformation procedures enabling the system state change within the same layer – $TP_{I(w)}$.
• "External" transformation procedures, enabling the system state change by transition to the "lower" layer - $TP_{E(w)}$.
• "External" transformation procedures, enabling the system state change by transition to the "higher" layer - $TP_{E(w)}$.

$$TP_{E(w)}\cup TP_{I(w)}\cup TP_{E(w)}=TP_w. \tag{15}$$

H. State graph

Let St_{s,w,O_s} denote a state structure. St_{s,w,O_s} is defined as tuple of the following elements:

$$St_{s,w,O_s}=\left(s,\lambda_s,O_s,w,D_{s,w},TP_{s,w},IVP_w,ZG_w,ZB_w\right), \tag{16}$$

where:

• *s* is a subject currently working in the system;
• λ_s is the subject's current sensitivity level;
• O_s is a subset of active objects

$$O_s=\left\{o\in O:\lambda_o\leq\lambda_s\wedge u_O(s,o)\neq\varnothing\right\}. \tag{17}$$

• *w* is a layer the subject is currently working in.
• $D_{s,w}$ is the substructure of access rights connected with the subject and the subset of active objects

$$D_{s,w}=\left\{u\subseteq U:\bigcup_{o\in O_s}\left(u_O(s,o)\subseteq u\right)\wedge u_W(s,w)\subseteq u\right\}. \tag{18}$$

• $TP_{s,w}$ is the subset of TP_w associated with the subject – the set of transformation procedures the subject is allowed to execute in the layer determining the current state

$$TP_{s,w}\subseteq TP_w. \tag{19}$$

• IVP_w is the subset of *IVP* associated with the layer the subject is currently in.
• ZG_w is the substructure of *ZG* associated with the layer the subject is currently in (cf. V.I)

$$\bigcup_{w\in W} ZG_w = ZG. \tag{20}$$

• ZB_w is the substructure of *ZB* associated with the layer the subject is currently in (cf. V.J)

$$\bigcup_{w\in W} ZB_w = ZB. \tag{21}$$

Let GS_s denote the system's state graph related to a particular subject. GS_s is a directed graph with vertices set identical with system states set associated with the subject and

edges representing $TP_{s,w}$ defined subsets. Let V_s denote the set of vertices. Given the subject $s \in S$

$$\bigvee_{w \in W} \bigvee_{O_s \subseteq O} St_{s,w,O_s} \in V_s . \quad (22)$$

Let E_s denote the set of edges. Given the subject $s \in S$

$$E_S = \bigcup_{w \in W} TP_{s,w}. \quad (23)$$

Then GS_s is an ordered triple

$$GS_s = (V_s, E_s, \phi), \quad (24)$$

where

$$\phi : E_s \to \left\{ (x, y) \in V_s^{\,2} : x \neq y \right\}. \quad (25)$$

I. Vulnerabilities and threats

Given the state graph definition (Sec. V.H) it is possible to identify the system's vulnerabilities as an occurrence of at least one of the following:

- Access of a subject to any state defined in other subjects' state graph.

  Let $s, s' \in S$, $s \neq s'$. Let $S' = S \setminus \{s'\}$. Then a state $St' \in GS_{s'}$ such that

$$\exists St \in GS_s : St = St' \quad (26)$$

  is considered an invalid state and therefore a vulnerability.
- Unauthorized transition to a valid state.

  Let *V* and *E* denote vertices and edges (respectively) for each subject in the system

$$\begin{cases} \bigcup_{s \in S} V_s = V \\ \bigcup_{s \in S} E_s = E \end{cases}. \quad (27)$$

  Let θ denote a function of transition between the system states

$$\theta : V \times E \to V . \quad (28)$$

  Let *tp* denote a transformation procedure such that

$$\left(tp \in TP \wedge tp \notin TP_{s,w} \right). \quad (29)$$

  Let $V_{s,w}$ denote a set of vertices associated with subject *s* and layer *w*, as well as all the vertices associated with the subject *s* and layers adjacent to the layer *w*. If

$$\exists v, v' \in V_{s,w} : \theta(v, tp) = v' \quad (30)$$

  then the *tp* procedure is considered an invalid transition factor and therefore a vulnerability.
- Skip of the intermediate state when transitioning to a valid state.

  Let θ be the function defined in (28). Let $tp_1, tp_2 \in TP$ be valid transformation procedures. If there exists a transformation procedure *tp* such that

$$\exists v, v' \in V : \theta(\theta(v, tp_1), tp_2) = \theta(v, tp) = v', \quad (31)$$

  then the *tp* procedure is considered a vulnerability.
- A valid state corruption.

  Let θ be the function defined in (28). Let $v \in V$ be a vertex with a set of integrity verification procedures IVP_v. Let *X* denote the set defined as following:

$$X = \left\{ (v, ivp) : v \in V, ivp \in IVP_v \right\}. \quad (32)$$

  Let φ denote a function of system state validation

$$\varphi : X \to \{0,1\}. \quad (33)$$

  Suppose v is a valid vertex

$$\bigvee_{ivp \in IVP_v} \varphi(v, ivp) = 1. \quad (34)$$

  Let $w \in V$ be a predecessor of v and also a valid vertex. Let $TP_{w \to v}$ be a set of transformation procedures transitioning system state from w to v. If there exists a transformation procedure $tp \in TP_{w \to v}$ such that

$$\left(\theta(w, tp) = v \right) \Rightarrow \left(\exists_{ivp \in IVP_v} \varphi(v, ivp) = 0 \right), \quad (35)$$

  then the *tp* procedure is considered a vulnerability.
- Transition to a corrupted state.

  Let φ be the function defined in (33). Let θ be the function defined in (28). For any vertex $v \in V$ and any integrity verification procedure $ivp \in IVP$ assigned to the vertex, if there exists a transformation procedure *tp* such that

$$\varphi(\theta(v, tp), ivp) = 0 , \quad (36)$$

  then the *tp* procedure is considered a vulnerability.

Based on identified vulnerabilities the threats set, denoted *T*, is constructed. Let *ZG* denote the threats structure (the *ZG* structure can be any data structure possible to be mapped to using *T* set). In the example, the structure is a matrix – constructed using zg_W function:

$$zg_W : (S \cup W) \times T \to \langle 0,1 \rangle. \quad (37)$$

Values of zg_W function should be considered as a measure of "consequences" resulting from executing a threat to a subject or a layer (in the example, the measure is a probability of executing a threat in the system).

J. Security measures

Let *M* denote security measures set – identified based on T. Let *ZB* denote the security measures structure (the *ZB* structure can be any data structure possible to be mapped to using *M* and *T* sets). The structure consists of two matrices – constructed using zb_W and zb_T functions:

$$\begin{cases} zb_W : (S \cup W) \times M \to \langle 0,1 \rangle \\ zb_T : M \times T \to \{0,1\} \end{cases}. \quad (38)$$

Values of zb_W function should be considered as a measure of "limitation of threats execution probability" resulting from applying a security measure to a subject or a layer.

Values of zb_T function should be considered as a binary

indicator whether a threat's probability of being executed is limited by a security measure.

Having defined *ZG* and *ZB* structures, the merged structure, denoted ZG_{ZB} is constructed using zg_{zb} function:

$$zg_{zb}: (S \cup W) \times T \to \langle 0,1 \rangle \tag{39}$$

defined by

$$zg_{zb}(e,t) = \max\big(0, f(e,t)\big), \tag{40}$$

where

$$\begin{cases} f(e,t) = zg_w(e,t) - \sum_{m \in M} g(e,m,t) \\ g(e,m,t) = zb_W(e,m) * zb_T(m,t) \end{cases}. \tag{41}$$

K. Penetration paths definition

Let *SP* denote penetration paths set. The probability of beating penetration path $SP_n \in SP$ is expressed by the following recursive formula:

$$\begin{cases} P(SP_n) = P(SP_{n-1}) * P(W_n) \\ P(SP_0) = 1 \end{cases} \tag{42}$$

where $P(SP_i)$ is a probability of beating *i-th* penetration path and $P(W_j)$ is a probability of penetrating *j-th* layer, expressed by the following formula:

$$P(W_j) = \max_{t \in T} \Big(zg_{zb}\big(W_j, t\big) \Big). \tag{43}$$

CONCLUSION

The methodology described in the paper represents generic approach towards defining security models for multi-layer systems. Models compatible with the methodology provide high level of data confidentiality and availability protection but lack strong mechanisms preventing integrity loss in "lower" layers of the systems – due to reversed entities' integrity dependencies in security systems. As described in Sec. IV.D, it is strongly recommended to implement cryptographic integrity protection security measures and thoroughly identify system's vulnerabilities connected with data integrity in order to implement sufficient integrity protection policies.

The most important part in the methodology is the layer structure conception. Well-defined layers provide convenient recursive, defense-in-depth based security system definition. The recursion enables highly generic security system definitions, which may result in reducing models' complexity. It is recommended that the models' output is strictly connected with penetration paths definitions – directly bound to the defined layers structures.

The presented idea of state graph as a representation of the data flow and user actions seems to be a sufficient basis for vulnerabilities identification, although it may be found redundant in numerous examples of simply-configured systems.

To summarize - all of the modelled system elements are represented by defined entities and structures in the model. Due to the above fact, it is claimed that the described methodology compatible models' level of adequacy to the modelled systems is sufficient, which implies the models may be used as a tool for measuring systems' security level.

REFERENCES

[1] Act of August 5, 2010 for the protection of classified information, Journal of Laws for 2010 No. 182, item 1228.

[2] Alpcan T., Basar T., A game theoretic analysis of intrusion detection in access control systems, Proc. IEEE Conference on Decision and Control: p. 1568-1573, USA 2004.

[3] Barkley J., Comparing Simple Role Based Access Control Models and Access Control Lists, Proc. Second ACM workshop on Role-Based Access Control: p. 127-132, USA 1997.

[4] Bell D.E., Looking Back at the Bell-La Padula Model, Proc. 21st Annual Computer Security Applications Conference: p. 337-352, USA 2005.

[5] Bell D.E., La Padula L.J., Secure Computer System: Unified Exposition and Multics Interpretation, ESD-TR-75-306, Bedford 1974 MA: ESD/AFSC, Hanscom AFB.

[6] Biba K.J., Integrity Considerations for Secure Computer Systems, MITRE, USA 1975.

[7] Chukwudi A.E., Udoka E., Ikerionwu C., Game Theory Basics and Its Application in Cyber Security, Advances in Wireless Communications and Networks. Volume 3, Issue 4: p. 45-49, 2017.

[8] Clark D.D., Wilson D.R., A Comparison of Commercial and Military Computer Security Policies, Proc. IEEE Symposium on Research in Security and Privacy: p. 184-194, USA 1987.

[9] Denning D.E., A lattice model of secure information flow, Communications of the ACM Volume 19 Issue 5: p. 236-243, USA 1976.

[10] Ferraiolo D., Kuhn D.R., Role-Based Access Controls, Proc. 15th National Computer Security Conference: p. 554-563, USA 1992

[11] Graham R., Denning P., Protection - Principles and Practice, Proc. AFIPS Spring Joint Computer Conference: p. 417-429, USA 1972.

[12] Harrison M.A., Ruzzo W.L., Ullman J.D., Protection in Operating Systems, Communications of the ACM Volume 19 Issue 8: p. 461-471, USA 1976

[13] ISO/IEC 27002:2013, Information technology — Security techniques — Code of practice for information security controls.

[14] Kuhn D.R. Role Based Access Control on MLS Systems Without Kernel Changes, Proc. Third ACM Workshop on Role-Based Access Control: p. 25-32, USA 1998

[15] Laskowski W., Teoriogrowe modele bezpieczeństwa systemów teleinformatycznych, Proc. IV Konferencja Entuzjastów Informatyki: p. 81-90, Chełm 2005.

[16] Loscocco P.A., Smalley S.D., Muckelbauer P.A., Taylor R.C., Turner S.J., Farrell J.F., The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments. Proc. 21st National Information Systems Security Conference: p. 303–314, October 1998

[17] Lye K., Wing J., Game strategies in network security, Proc. 15th IEEE Computer Security Foundations Workshop, Copenhagen, Denmark 2002.

[18] Regulation of the Prime Minister dated July 20, 2011 on the basic requirements of IT security, Journal of Laws for 2011, No. 159, item 948.

[19] Sandhu R.S., Lattice-Based Access Control Models, IEEE Computer, Volume 26 Issue 11: p. 9-19, USA 1993.

[20] Wiechecki D., The security model of a virtual machines system with multi-layer cryptographic protection, Military University of Technology, Warsaw, 2019.