Possible variant of the confidential data leak detection model

I S Chernousov¹, A E Borshevnikov¹, D V Zakharchenko¹, Y V Dobrzhinskii¹

¹Far Eastern Federal University, Vladivostok, Russky Island, Ajax Bay, 10, 690922, Russia

E-mail chernousov.is@students.dvfu.ru

Abstract. The paper discusses a model for warning about leaks of electronic documents, describes a possible structure of a system for monitoring document leaks. Within the framework of the paper, the model is verified by evidence and formal methods.

Keywords: Data leaks; Leak detection; Corporate data protection; Electronic document tagging

1. Introduction

As the number of corporate data leaks is growing more and more, the question of the need for measures to combat data leaks arises. One of the options for increasing the efficiency of information protection systems is to warn owners about leaks that have already occurred, which allows them to take timely actions to minimize damage [1].

This mechanism cannot be considered as an independent element of the electronic data protection system, but only as an additional component to ensure awareness of data theft [2].

2. Sensitive data leak detection model

During the developing process of the mathematical model, the next main principles were taken [3]:

1. Control of the host;
2. Reference database;
3. Use of hash functions;
4. Control of the traffic.

Let us go directly to the description of the developed mathematical model. In the development process, a set-theoretic method will be used. Let’s define the subjects of interaction:

Consider the set $U$ is the set of all possible users (access subjects):

$$U = \{U_1, U_2, \ldots, U_s, E_1, \ldots, E_k, T\}, \ n = s + k + 1,$$

where, $n$ is the number of all users; $s$ is the number of all valid users; $k$ is the number of all intruders; $T$ is the trusted subject of a computer network, which is a computer network administrator and a trusted server.

Let $U_l, E_g = \{UID_{Ul(E_g)}\}$ be a user (attacker) identifier.

Continuing with the introduction of the concept of access objects:

$$\Theta = PC_l \times M_j,$$

where $\times$ - cartesian product of sets; $PC_l$ - i-th object of a computer network.

Consider j-th file:
\[ M_j = (m_j, mark_j, rand_j), \]  
where \( m_j \) is a message; \( mark_j \) is a mark itself; \( rand_j \) is a random number.

Based on the above entities, the following sets will be formed:
1. Set of all files \( M = \{ M_j; j = 1,f \} \);
2. Set of all devices \( PC = \{ PC_i; i = 1,k \} \).

Let \( |O| = r \).

Continuing with the introduction of \( i \)-th object of a computer network:
\[ PC_i = (IDT_i, IP_i, AM_i), \]
where, \( IDT_i \) is the identifier of \( i \)-th object of a computer network; \( IP_i \) – IP address of \( i \)-th object of a computer network; \( AM_i \) - matrix of access of subjects of access to objects of access.

Consider the trusted subject of a computer network \( T \):
\[ T = \{ (t_{mark}, UID, IDT, h(M_j), rand_j, IP_i) \}, \]
where, \( t_{mark} \) is the time the mark was created; \( h(M_j) \) - hash value from \( j \)-th file; \( rand_j \) is a random number[4].

For a better understanding of the interaction between users and the object, we introduce the access function \( \alpha \):
\[ \alpha: U \times O \rightarrow \{0,1\} \]
\[ \alpha(U_p, O_q) = \begin{cases} 0, & O_q \notin U_p \, p = 1, n, q = 1, r. \\ 1, & O_q \in U_p \end{cases} \]

We will assume that
\[ O_q \in U_p \leftrightarrow (PC_q \in U_p) \land (M_q \in U_p) \]

In the same connection, through the access function, it is possible to describe the violator of the access control rules who applied to documents to which he does not have access:
1. \( UID \neq 0 \) for \( \alpha(U_p, O_q) = 0 \) - internal violator;
2. \( UID = 0 \) for \( \alpha(U_p, O_q) = 0 \) - external violator

The access matrix can be described as follows:
\[ AM = \begin{pmatrix} AC_{1,1} & \cdots & AC_{1,n} \\ \vdots & \ddots & \vdots \\ AC_{r,1} & \cdots & AC_{r,n} \end{pmatrix}, \]

where \( AC_{p,q} = \alpha(U_p, O_q), p = 1, n, q = 1, r. \)

Unauthorized access to an object should be considered the case when \( (AC_{p,q} = 0) \land (M_q \in U_p) \).

For a mathematical description of the interaction of access objects \( O \), which in accordance with this scheme are user devices and files of critical importance to be protected, we introduce a membership function \( \beta \) such that:
\[ \beta: O \rightarrow \{0,1\} \]
\[ \beta(O_q) = \begin{cases} 0, & M_j \notin PC_i \\ 1, & M_j \in PC_i \end{cases} \]

The belong matrix can be described as follows:
\[ BM = \begin{pmatrix} B_{1,1} & \cdots & B_{1,f} \\ \vdots & \ddots & \vdots \\ B_{k,1} & \cdots & B_{k,f} \end{pmatrix}, \]

where \( B_{i,j} = \beta(O_q), q = 1, r, i = 1, k, j = 1, f. \)

By leakage we mean the situation:
(B_{i,j} = 0) \land (M_j \in PC_i) \quad (13)

Let's define the following operations:

1. Access to a file

\[ U_l \xrightarrow{\text{give}} PC_i; UID_l, ID(M_j) \quad (14) \]

The user contacts the device, to which he transmits his personal unique identifier and the identifier of the document to which he requests access [5].

1.1. \( PC_i; AC_{l,i} \xrightarrow{\text{give}} 1 \) checking for access rights;

1.2. \( M_j \xrightarrow{\text{give}} PC_i; M_j \);

1.3. \( PC_i \xrightarrow{\text{give}} U_l; M_j \).

2. Copy file

2.1. \( U_l \xrightarrow{\text{give}} PC_i; UID_l, ID(M_j) \);

2.2. \( PC_i; AC_{l,i} \xrightarrow{\text{give}} 1 \);

2.3. \( M_j \xrightarrow{\text{give}} PC_i; M_j \);

2.4. \( PC_i \xrightarrow{\text{give}} PC_k; M_j \);

2.5. \( PC_k \xrightarrow{\text{give}} M_j; M_j \).

3. File marking

3.1. \( T \xrightarrow{\text{give}} PC_i; mark_j, rand_j \);

3.2. \( PC_i; AC_{l,i} \xrightarrow{\text{give}} 1 \);

3.3. \( PC_i \xrightarrow{\text{give}} M_j; mark_j, rand_j \);

3.4. \( M_j; m_j, mark_j, rand_j \);

3.5. \( M_j \xrightarrow{\text{give}} PC_i; h(M_j) \);

3.6. \( PC_i \xrightarrow{\text{give}} T; h(M_j), IDT_i, IP_i \).

4. User access to the marked file

4.1. \( U_l \xrightarrow{\text{give}} PC_i; UID_l, ID(M_j) \);

4.2. \( PC_i; AC_{l,i} \xrightarrow{\text{give}} 1 \);

4.3. \( M_j \xrightarrow{\text{give}} PC_i; mark_j, rand_j, h(M_j) \);

4.4. \( PC_i \xrightarrow{\text{give}} T; mark_j, rand_j, h(M_j), IDT_i, IP_i \);

4.5. \( T \xrightarrow{\text{give}} PC_i; mark_j \);

4.6. \( PC_i \xrightarrow{\text{give}} M_j; mark_j \).

Let's proceed to the verification of the mathematical model of leak detection.

3. Evidence-based verification

Let us verify the constructed model with respect to the Bell-LaPadula model. The main provisions of the Bell-LaPadula (BLP) model are as follows[7].

The system model \( \Sigma(\theta_0, Q, F_T) \) is represented by the set:

1. Set of the access objects \( O \);

2. Set of the access subjects \( S \);

3. Sets of access rights \( R \) (in the classical model “read”, “write”);

4. Access matrix \( A[s,o] \);

5. Lattices \( \Lambda_L \) of security levels of \( L \) subjects and objects of the system;

6. Function \( F_T: S \cup O \rightarrow L \), that maps elements of the set \( S \) and \( O \) to the set \( L \);

7. The set of states of the system \( V \), which is determined by the set of ordered pairs \( (F_L, A) \);
8. Initial state \( \theta_0 \in V \);
9. A set of requests of \( Q \) subjects for access (performing an operation on objects, the execution of which transfers the system to a new state);
10. The transition function \( \mathcal{F}_T: (V \times Q) \rightarrow V^* \), which transfers the system from one state to another \( V^* \) when executing queries from \( Q \).

The states of the system are divided into dangerous and safe. To analyze and formulate conditions that ensure the safety of system states, the following definitions are introduced:

1. A state is called read-safe \( \leftrightarrow \) when, for each subject performing read access to an object in this state, the security level of that subject dominates over the security level of that object:
\[
\forall s \in S, \forall o \in O, read \in A[s, o] \rightarrow \mathcal{F}_L(s) \geq \mathcal{F}_L(o) \tag{15}
\]
2. A state is called write-safe \( \leftrightarrow \) when, for each subject accessing an object in this state, the security level of that object dominates the security level of that subject:
\[
\forall s \in S, \forall o \in O, write \in A[s, o] \rightarrow \mathcal{F}_L(s) \geq \mathcal{F}_L(o) \tag{16}
\]
3. A system state is safe when it is both read and write safe.

Let's correlate the elements of the developed and basic models:
1. \( O \sim O = PC_i \times M_j; \)
2. \( S \sim U; \)
3. \( A[s, o] \sim AM; \)
4. \( \Lambda_L \sim O_i = \alpha(U_1, O_1) \cdot U_1 + \alpha(U_2, O_1) \cdot U_2 + \cdots + \alpha(T, O_1) \cdot T; \)
5. \( \mathcal{F}_L : S \cup O \rightarrow L \sim \alpha(U_p, O_q); \)
6. \( V = \{ (\mathcal{F}_L, A) \sim AC_{p,q} \land (M_q \in U_p); \)
7. \( \theta_0 \in V \sim initial \ state \ of \ the \ system; \)
8. \( Q \sim operations; \)
9. \( \mathcal{F}_T: (V \times Q) \rightarrow V^* \sim B_{i,j} \land (M_j \in PC_i); \)
10. \( L = \{0,1\}. \)

Let us assume that read and write operations are indistinguishable within the framework of this model.

Consider two cases of leak \( (B_{i,j} \land (M_j \in PC_i) = 0). \)

Consider the first case of a leak:
1. \( E_i \rightarrow PC_i; UID_i, ID(M_j); \)
2. \( PC_i; AC_i \Rightarrow = 1; \)
3. \( M_j \rightarrow PC_i; M_j; \)
4. \( PC_i \rightarrow PC_k; M_j; \)
5. \( PC_k \rightarrow M_j; M_j; \)

The controversy arises in step 2, but the system cannot be brought to a safe state since no leak can be detected.

Let the file be marked, then consider the following situation:
1. \( E_i \rightarrow PC_i; UID_i, ID(M_j); \)
2. \( PC_i; AC_i \Rightarrow = 1; \)
3. \( M_j \rightarrow PC_i; mark_j, rand_j, h(M_j); \)
4. \( PC_i \rightarrow T: mark_j, rand_j, h(M_j), 1DT_i, 1P_i; \)
5. \( T \rightarrow PC_i; mark_j; \)
6. \( PC_i \rightarrow M_j; mark_j; \)
7. $M_j \xrightarrow{\text{give}} PC_i; M_j$

8. $PC_i \xrightarrow{\text{give}} U_i; M_j$

The controversy also arises at the step 2; however, unauthorized access cannot be detected either. You can get out of this situation by excluding illegitimate users from the system at the stage of preparing to launch the system.

Consider the second case of a leak:

1. $U_i(E_l) \xrightarrow{\text{give}} PC_i; UID_i, ID(M_j)$;
2. $PC_i: AC_{i,1} = 1; \notin$
3. $M_j \xrightarrow{\text{give}} PC_i; M_j$
4. $PC_i \xrightarrow{\text{give}} PC_k; M_j$
5. $PC_k \xrightarrow{\text{give}} M_j; M_j$
6. $U_i(E_l) \xrightarrow{\text{give}} PC_i; UID_i, ID(M_j)$;
7. $PC_k: AC_{i,k} = 1; \notin$
8. $M_j \xrightarrow{\text{give}} PC_k; M_j$
9. $PC_k \xrightarrow{\text{give}} U_i(E_l); M_j$

A contradiction can occur in steps 2 and 7, but the system cannot be brought to a safe state as no leak can be detected.

1. $U_i(E_l) \xrightarrow{\text{give}} PC_i; UID_i, ID(M_j)$;
2. $PC_i: AC_{i,1} = 1; \notin$
3. $M_j \xrightarrow{\text{give}} PC_i; M_j$
4. $PC_i \xrightarrow{\text{give}} PC_k; M_j$
5. $PC_k \xrightarrow{\text{give}} M_j; M_j$
6. $U_i(E_l) \xrightarrow{\text{give}} PC_i; UID_i, ID(M_j)$;
7. $PC_k: AC_{i,k} = 1; \notin$
8. $M_j \xrightarrow{\text{give}} PC_k; \text{mark}_j, \text{rand}_j, h(M_j)$;
9. $PC_k \xrightarrow{\text{give}} T; \text{mark}_j, \text{rand}_j, h(M_j), IDT_i, IP_i$
10. $T \xrightarrow{\text{give}} PC_k; \text{mark}_j$
11. $PC_i \xrightarrow{\text{give}} M_j; \text{mark}_j$
12. $M_j \xrightarrow{\text{give}} PC_k; M_j$
The contradiction arises at the steps 2 and 7, but the detection of the contradiction occurs at the 9 step. For each of the detected cases, the system is brought to a safe state in a countable number of steps by removing the user from the system. Let us bring the described procedures into a safe state due to the methods described above → $\theta_0$ - safe (it is this state that should be considered the initial one).

Due to the condition adopted within this model that the read and write operations are indistinguishable, it is necessary and enough to understand, that $F^i_L(s) = F^i_O(o)$ and $\text{read}, \text{write} \in A'[s, o]$, if $F^i_L(s) \neq F^i_O(o)$.

Let for the i-th user:

$$A_{C_{li}} = 0 \rightarrow \alpha(U_l, O_l) = 0 \rightarrow O_l \notin U_l \rightarrow F^i_L(s) = F^i_O(o) \quad (17)$$

$$A_{C_{li}} = 1 \rightarrow \alpha(U_l, O_l) = 1 \rightarrow O_l \notin U_l \rightarrow F^i_L(s) = F^i_O(o) \quad (18)$$

$$A_{C_{li}} = 0 \rightarrow \alpha(U_l, O_l) = 1 \rightarrow \ell \rightarrow \text{read}, \text{write} \notin A'[s, o] \quad (19)$$

$$A_{C_{li}} = 1 \rightarrow \alpha(U_l, O_l) = 0 \rightarrow \ell \rightarrow \text{read}, \text{write} \notin A'[s, o] \quad (20)$$

Then, by the main security theorem, the system is safe.

4. Conclusion
Here are the main results of the study:

1. A confidential data leak detection model was developed and verified by evidence-based method;
2. Among the main aspects characterizing the developed mathematical model, one can single out the ability to detect leaks outside the information system, which has a positive effect on the information security of the organization as a whole.

Thus, this work presented a new look at the issue of detecting leaks of confidential data.

References
[1] Papadimitriou P and Garcia-Molina H 2011 Data Leakage Detection (Knowledge and Data Engineering, IEEE Transactions) pp 51–63
[2] Riya N and Manisha G 2019 Data Leakage Detection in cloud using Watermarking Technique (ICCCI) pp 10–15
[3] Suryawanshi M and Patil S 2016 Avoiding the data leakage and providing privacy to data in networking (ICCUBEA) pp 1–4
[4] Salvia R, Ferrara R, Spoto F and Cortesi A 2018 SDLI: Static Detection of Leaks across Intents (IEEE TrustCom-18) pp 33–52
[5] Huang X, Lu Y, Li D and Ma M A Novel Mechanism for Fast Detection of Transformed Data Leakage (IEEE Access vol 6) pp 35926–35936
[6] Bidmeshki M, Antonopoulos A and Makris Y 2017 Information Flow Tracking in Analog/Mixed-Signal Designs through Proof-Carrying Hardware IP (Design, Automation and Test in Europe) pp 1703–1708
[7] Bell D Looking Back at the Bell-LaPadula Model (Proceedings of the 21st Annual Computer Security Applications Conference) pp 337–351
[8] Raja Koti B, Raj Kumar G and Naveen Kumar K 2019 Identification of Information Leakage and Guilt Agent by using MAC-IP Binding and Recursive Partitioning Algorithm to Modulate the Uncertainty in the Organization’s Network (International Journal of Innovative Technology and Exploring Engineering vol 9) pp 4091–4096
[9] Cinque M, Cotroneo D and Pecchia A 2018 Challenges and Directions in Security Information and Event Management (SIEM) (ISSREW vol 1) pp 95–99