DOI 10.37882/2223–2966.2021.07.14

ВЕРИФИКАЦИЯ МОДЕЛИ СИСТЕМЫ МОНИТОРИНГА УТЕЧЕК КОНФИДЕНЦИАЛЬНЫХ ДОКУМЕНТОВ

VERIFICATION OF THE MODEL OF A SYSTEM FOR MONITORING LEAKS OF CONFIDENTIAL DOCUMENTS

D. Zakharchenko
A. Borshevnikov

Summary. The article discusses a new formal security model designed to warn about leaks of electronic documents. This model allows you to track the movement of documents on the Internet and detect unauthorized access to them. This document also provides verification of the developed model using Bell-LaPadula and take-grant models.

Keywords: information security, security policy, data protection, data leaks, leak detection, formal methods, take-grant model, Bell-LaPadula model.

Захарченко Даниил Владимирович
Аспирант, Дальневосточный федеральный университет (г. Владивосток)
daniilZakharchenko@gmail.com
Боршевников Алексей Евгеньевич
Старший преподаватель, Дальневосточный федеральный университет (г. Владивосток)
borshevnikov.ae@dvfu.ru

Аннотация. В статье рассматривается новая формальная модель безопасности, предназначенная для предупреждения о произошедших утечках электронных документов. Данная модель позволяет отслеживать перемещения документов в сети интернет и обнаруживать несанкционированный доступ к ним. Также в данном документе приводится верификация разработанной модели с применением моделей Белла-ЛаПадулы и take-grant.

Ключевые слова: информационная безопасность, политика безопасности, защита корпоративных данных, утечки данных, обнаружение утечек, формальные методы, модель take-grant, модель Белла-ЛаПадулы.

Одной из значимых проблем в обеспечении защищенности данных являются утечки конфиденциальной информации. В связи с тем, что при обеспечении защищенности систем большая часть усилий направлена именно на защиту от внешних угроз, стало сильно заметно влияние инцидентов утечек данных, происходящих по вине сотрудников. При этом число таких инцидентов растет ежегодно, за прошлое десятилетие число утечек выросло в 2,5 раза, при этом в последние 3 года было зафиксирована стабилизация ситуации с ростом утечек. Так, в 2010 году было известно о 794 случаях утечек информации в мире. В 2017 году их число составило 2131 по всему миру [1]. В 2018 году число известных утечек по миру составило 2263 [2]. В 2019 году было обнаружено уже 2509 подобных инцидентов [3].

Согласно отчету компании infowatch за первые 9 месяцев 2020 году [4], в мире, за этот промежуток времени, произошло 1773 случая утечки информации, что на 7,4% меньше, чем за соответствующий промежуток времени 2019 года. В России в 2020 году за это время произошло 302 случая, что на 5,6% больше соответствующего периода 2019 года. При этом на внутреннего нарушителя приходится 52,6% от всех утечек по всему миру. Для России этот показатель равен 79,1%.

В среднем, на внутренние утечки приходится около половины от их общего количества. При этом важно отметить, что большая часть утечек происходит по вине сотрудника. Так, в России за 2017 год, 69,3% от общего числа инцидентов приходились на долю рядового сотрудника, 6,7% — руководителей, и 21,3% происходили по вине внешних злоумышленников. В мире ситуация отличается в сторону большего числа внешних виновников: 50,3% — рядовые сотрудники, 2,2% — руководители, 41,7% — внешний злоумышленник.[5]

Для обнаружения и предотвращения утечек конфиденциальной информации из информационной системы применяются различные DLP системы [6]. Примерами таких систем являются [7, 8, 9]. Система, разрабатываемая Баранкова И.И., Михайлова У.В. и Лукьянов Г.И. [7] предназначена для отслеживания подозрительных файлов и блокирования устройства пользователя при их обнаружении.

Так же в качестве примера подобных систем можно привести DLP-систему, описанную в работе В.Е. Морозова, А.В. Дрозда [9]. Суть данной системы заключается в перехвате исходящего сетевого трафика с дальнейшим выделением контролируемой информации. Контроль осуществляется как через сервер, мониторящий

трафик, так и с применением ПО, установленного на конечные устройства пользователей.

В качестве примера коммерческой DLP системы можно рассмотреть комплексную dlp-систему компании SearchInform, предназначенную как для контроля конечных устройств пользователей, так и контроля трафика, проходящего через узлы компании [8].

Для дополнительных способов контроля утечек могут применяться средства, которые позволяют отследить сам факт свершения утечки, для скорейшего реагирования на произошедший инцидент. В качестве примера подобных систем можно привести систему, разработанную Slim Trabelsi [10]. Данная система предназначена для отслеживания факта появления данных путем мониторинга наиболее популярных сетевых ресурсов, на которых выкладывается подобная информация. Схожую задачу так же решает коллектив Timo Malderle1, Matthias W¨ubbeling, Sven Knauer, Michael Meier [11]. Их целью является разработка системы, которая бы отслеживала публикацию утечек и самостоятельно уведомляла бы потерпевших о инцидентах.

Другим способом защиты является использование специализированных систем документооборота, таких как описанная в работе Н.Д. Быстрицкого, Н.В. Макарова-Землянского, В.С. Назаров [12].

В данной статье рассматривается модель контроля фактов утечек данных, базирующаяся на внедрении специальных меток в электронный документ, которые позволяют отслеживать событие открытия документа и информируют доверенное лицо об этом событии.

Для начала определим множество пользователей (субъектов доступа) U, которые могут получить доступ к документам. В это множество входят легитимные пользователи, которым разрешен доступ к документам, доверенный пользователь системы, которым выступает администратор компьютерной сети и доверенный сервер, а также нелегитимные пользователи, которые могут получить доступ к документам без разрешения доступа к ним:

$$U = \{U_1, U_2, \dots, U_s, E_1, \dots, E_k, T\}, \quad (1)$$

$$n = s + k + 1, \quad (2)$$

где n — количество всех пользователей в компьютерной сети; s — количество легитимных пользователей; k — количество нелегитимных пользователей (злоумышленники); T — доверенный субъект компьютерной сети.

Для определения каждого из этих пользователей можно использовать идентификатор пользователя (или злоумышленника):

$$U_i, E_g = \{UID_{U_i(E_g)}\} \quad (3)$$

В самих защищаемых документах выделим 3 основных компонента — текст (сообщение), хранящийся в документе; метка документа, позволяющая идентифицировать сам документ, а также сотрудника, осуществившего маркировку; случайное число, являющееся идентификатором самой метки. Обозначим j-тый документ в модели:

$$M_j = (m_j, mark_j, rand_j), \quad (4)$$

где m_j — сообщение j-го файла; $rand_j$ — случайное число; $mark_j$ — метка j-го файла.

Из свойств устройств, на которых будет осуществляться доступ к документам, наибольший интерес представляют идентификатор устройства, адрес устройства в сети и список разрешенных для открытия на этом устройстве документов. Таким образом опишем устройства, с которых осуществляется доступ к документам, а также устройства, на которых производится маркировка:

$$PC_i = (IDT_i, IP_i, AM_i), \quad (5)$$

где IDT_i — идентификатор i-го объекта компьютерной сети; IP_i — IP-адрес i-го объекта компьютерной сети; AM_i — матрица доступа субъектов доступа к объектам доступа.

Соответственно будут образованы множество всех файлов M, множество всех устройств PC:

$$M = \{M_j : j = \overline{1, f}\}, \quad (6)$$

$$PC = \{PC_i : i = \overline{1, k}\}, \quad (7)$$

В качестве объектов доступа O в модели определяется на каких объектах компьютерной сети могут быть открыты защищаемые документы:

$$O = PC_i \times M_j, \quad (8)$$

$$|O| = r, \quad (9)$$

где × — декартово произведение множеств; PC_i — i-й объект компьютерной сети; M_j — j-й файл, r мощность множества объектов.

Для определения маркированного документа необходимо хранить информацию о нем в системе. Для этого

применяется доверенный субъект компьютерной сети. Для идентификации, когда была проведена маркировка, необходимо хранить время маркировки, для точной идентификации устройства, на котором была проведена маркировка и пользователя, ее осуществившего — идентификатор пользователя, идентификатор и сетевой адрес устройства, для представления изначального содержимого документа — его хэш. Для связи метки и документа хранится идентификатор самой метки:

$$T = \{(t_{mark}, UID_l, IDT_i, h(M_j), rand_j, IP_i)\}, \quad (10)$$

где t_{mark} — время создания временной метки; $h(M_j)$ — значение хеш-функции от j-го файла.

Введем функцию доступа, описывающую пользователем доступа к объекту системы. В случае, если пользователь имеет право на доступ к объекту, функция возвращает 1, в ином случае — 0:

$$б: U \times O \to \{0,1\}, \quad (11)$$

$$б(U_p, O_q) = \begin{cases} 0, O_q \notin U_p \\ 1, O_q \in U_p \end{cases}, \quad (12)$$

где $p = \overline{1,n}$, а $q = \overline{1,r}$.

В свою очередь, пользователь имеет доступ к объекту тогда и только тогда, когда и документ, и устройство одновременно принадлежат пользователю:

$$O_q \in U_p \leftrightarrow (PC_q \in U_p) \wedge (M_q \in U_p). \quad (13)$$

При применении функции доступа ко всем пользователям ко всем объектам строится матрица доступа, которую можно описать следующим образом:

$$AM = \begin{pmatrix} AC_{1,1} & \cdots & AC_{1,n} \\ \vdots & \ddots & \vdots \\ AC_{r,1} & \cdots & AC_{r,n} \end{pmatrix}, \quad (14)$$

где $AC_{p,q} = б(U_p, O_q), p = \overline{1,n}, q = \overline{1,r}$.

Следовательно, можно определить несанкционированный доступ к объекту как случай, когда пользователь получает доступ к документу, в то время как матрица доступа для них возвращает 0:

$$(AC_{p,q} = 0) \wedge (M_q \in U_p). \quad (15)$$

Теперь определим, может ли документ быть на устройстве системы. Для этого введём функцию принадлежности некоторого сообщения некоторому устройству:

$$в: O \to \{0,1\}, в(O_q) = \begin{cases} 0, M_j \notin PC_i \\ 1, M_j \in PC_i \end{cases}. \quad (16)$$

Применяя данную функцию ко всем документам и устройствам, построим матрицу принадлежности файла устройству:

$$BM = \begin{pmatrix} B_{1,1} & \cdots & B_{1,f} \\ \vdots & \ddots & \vdots \\ B_{k,1} & \cdots & B_{k,f} \end{pmatrix}, \quad (17)$$

где $B_{i,j} = в(O_q), q = \overline{1,r}, i = \overline{1,k}, j = \overline{1,f}$.

Теперь можно определить утечку данных. Под ней будем понимать ситуацию, когда сообщение находится на устройстве при том, что соответствующий элемент матрицы принадлежности равен 0:

$$(B_{i,j} = 0) \wedge (M_j \in PC_i). \quad (18)$$

Пусть $ID(M_j)$ — идентификатор сообщения, а под *give* подразумевается передача данных от одного элемента модели другому. Определим 4 операции работы с файлами:

1. Получение пользователем доступа к файлу $ID(M_j)$:

$$U_l \xrightarrow{give} PC_i: UID_l, ID(M_j), \quad (19)$$

$$PC_i: AC_{l,i} == 1, \quad (20)$$

$$M_j \xrightarrow{give} PC_i: M_j, \quad (21)$$

$$PC_i \xrightarrow{give} U_l: M_j. \quad (22)$$

2. Копирование файла $ID(M_j)$:

$$U_l \xrightarrow{give} PC_i: UID_l, ID(M_j) \quad (23)$$

$$PC_i: AC_{l,i} == 1 \quad (24)$$

$$M_j \xrightarrow{give} PC_i: M_j, \quad (25)$$

$$PC_i \xrightarrow{give} PC_k: M_j, \quad (26)$$

$$PC_k \xrightarrow{give} M_j: M_j, \quad (27)$$

3. Маркирование файла $ID(M_j)$:

$$T \xrightarrow{give} PC_i: mark_j, rand_j, \quad (28)$$

$$PC_i: AC_{l,i} == 1, \quad (29)$$

$$PC_i \xrightarrow{give} M_j: mark_j, rand_j, \quad (30)$$

$$M_j \xrightarrow{give} PC_i: h(M_j), \quad (31)$$

$$M_j \xrightarrow{give} PC_i: h(M_j), \quad (32)$$

$$PC_i \xrightarrow{give} T: h(M_j), IDT_i, IP_i. \quad (33)$$

4. Получение пользователем доступа к маркированному файлу $ID(M_j)$:

$$U_l \xrightarrow{give} PC_i: UID_l, ID(M_j), \quad (34)$$

$$PC_i: AC_{l,i} == 1, \quad (35)$$

$$M_j \xrightarrow{give} PC_i: mark_j, rand_j, h(M_j), \quad (36)$$

$$PC_i \xrightarrow{give} T: mark_j, rand_j, h(M_j), IDT_i, IP_i \quad (37)$$

Далее рассмотрим верификацию предложенной модели с помощью двух моделей: модели Белла-ЛаПадулы и модели take-grant.

Проведем верификацию построенной модели относительно модели Белла-ЛаПадулы (D. Bell, L. LaPadula) [13]. Основным положением модели Белла-ЛаПадулы является основная теорема безопасности — состояние системы безопасно тогда и только тогда, когда оно безопасно и по чтению, и по записи.

Для верификации, предлагаемой в рамках данной работы модели, введем следующие обозначения [13]:

O — множество объектов доступа; S — множество субъектов доступа; R — множество прав доступа; $A[s, o]$ — матрица доступа; Λ_L — решетка L уровней безопасности между субъектами и объектами системы; $\mathcal{F}_L$ — отображение объединения множества S и O на множество L; V — множество состояний системы; ϑ_0 — начальное состояние; Q — запросы субъектов на доступ; $\mathcal{F}_T$ — отображение, переводящее систему из одного состояния в другое при выполнении запросов.

Соотнесем элементы разрабатываемой и базовой моделей:

$$O \sim O = PC_i \times M_j \quad (40)$$

$$S \sim U \quad (41)$$

$$A[s, o] \sim AM \quad (42)$$

$$\Lambda_L \sim O_i = б(U_1, O_i) \cdot U_1 + \\ + б(U_2, O_i) \cdot U_2 + \cdots + б(T, O_i) \cdot T \quad (43)$$

$$\mathcal{F}_L : S \cup O \to L \sim б(U_p, O_q) \quad (44)$$

$$V = \{(\mathcal{F}_L, A)\} \sim AC_{p,q} \wedge (M_q \in U_p) \quad (45)$$

$\vartheta_0 \in V \sim$ начальное состояние системы;

Q — операции;

$$\mathcal{F}_T: (V \times Q) \to V^* \sim B_{i,j} \wedge (M_j \in PC_i) \quad (46)$$

$$L = \{0,1\}. \quad (47)$$

Будем считать, что операции read и write неотличимы в рамках данной модели.

Рассмотрим 2 случая утечки

$$(B_{i,j} \wedge (M_j \in PC_i) = 0) \quad (48)$$

Рассмотрим первый случай утечки:

1. $E_l \xrightarrow{give} PC_i: UID_l, ID(M_j)$;
2. $PC_i: AC_{l,i} == 1$; ↯
3. $M_j \xrightarrow{give} PC_i: M_j$;
4. $PC_i \xrightarrow{give} PC_k: M_j$;
5. $PC_k \xrightarrow{give} M_j: M_j$.

Противоречие возникает на 2-м шаге, однако систему невозможно привести к безопасному состоянию, так как нельзя обнаружить утечку.

Пусть файл промаркирован, тогда рассмотрим следующую ситуацию:

1. $E_l \xrightarrow{give} PC_i: UID_l, ID(M_j)$;
2. $PC_i: AC_{l,i} == 1$; ↯
3. $M_j \xrightarrow{give} PC_i: mark_j, rand_j, h(M_j)$;
4. $PC_i \xrightarrow{give} T: mark_j, rand_j, h(M_j), IDT_i, IP_i$;
5. $T \xrightarrow{give} PC_i: mark_j$;
6. $PC_i \xrightarrow{give} M_j: mark_j$;
7. $M_j \xrightarrow{give} PC_i: M_j$;
8. $PC_i \xrightarrow{give} U_l: M_j$.

Противоречие также возникает на 2-м шаге, однако также нельзя обнаружить несанкционированный доступ. Можно выйти из этой ситуации путем исключения

нелегитимных пользователей из системы на этапе подготовки к запуску системы.

Рассмотрим второй случай утечки конфиденциальных данных.

1. $U_l(E_l) \xrightarrow{give} PC_i: UID_l, ID(M_j)$;
2. $PC_i: AC_{l,i} == 1;$ ↯
3. $M_j \xrightarrow{give} PC_i: M_j$;
4. $PC_i \xrightarrow{give} PC_k: M_j$;
5. $PC_k \xrightarrow{give} M_j: M_j$;
6. $U_l(E_l) \xrightarrow{give} PC_i: UID_l, ID(M_j)$;
7. $PC_k: AC_{l,k} == 1;$ ↯
8. $M_j \xrightarrow{give} PC_k: M_j$;
9. $PC_k \xrightarrow{give} U_l(E_l): M_j$.

Противоречие может возникнуть на 2-м и 7-м шагах, однако систему невозможно привести к безопасному состоянию, так как нельзя выявить утечку.

1. $U_l(E_l) \xrightarrow{give} PC_i: UID_l, ID(M_j)$;
2. $PC_i: AC_{l,i} == 1;$ ↯
3. $M_j \xrightarrow{give} PC_i: M_j$;
4. $PC_i \xrightarrow{give} PC_k: M_j$;
5. $PC_k \xrightarrow{give} M_j: M_j$;
6. $U_l(E_l) \xrightarrow{give} PC_i: UID_l, ID(M_j)$;
7. $PC_k: AC_{l,k} == 1;$
8. $M_j \xrightarrow{give} PC_k: mark_j, rand_j, h(M_j)$;
9. $PC_k \xrightarrow{give} T: mark_j, rand_j, h(M_j), IDT_i, IP_i$;
10. $T \xrightarrow{give} PC_k: mark_j$;
11. $PC_i \xrightarrow{give} M_j: mark_j$;
12. $M_j \xrightarrow{give} PC_k: M_j$;
13. $PC_k \xrightarrow{give} U_l(E_l): M_j$.

Противоречие возникает на 2-м и 7-м шагах, однако обнаружение противоречия происходит на 9-м шаге. Для каждого из обнаруженных случаев система приводится к безопасному состоянию за счетное количество шагов с помощью удаления пользователя из системы. Приведем описанные процедуры в безопасное состояние за счет методов, описанных выше → ϑ_0 — безопасное (именно это состояние стоит считать начальным).

За счет принятого в рамках данной модели условия, что операции read и write неотличимы, то необходимо и достаточно понять, что $\mathcal{F}_L^*(s) = \mathcal{F}_L^*(o)$ и $read, write \in A^*[s,o]$, если $\mathcal{F}_L^*(s) \neq \mathcal{F}_L^*(o)$

Пусть для i-го пользователя:

$$AC_{i,l} = 0 \rightarrow б(U_i, O_l) = \\ = 0 \rightarrow O_l \notin U_i \rightarrow \mathcal{F}_L(s) = \mathcal{F}_L(o)$$

$$AC_{i,l} = 1 \rightarrow б(U_i, O_l) = \\ = 1 \rightarrow O_l \in U_i \rightarrow \mathcal{F}_L(s) = \mathcal{F}_L(o)$$

$$AC_{i,l} = 0 \rightarrow б(U_i, O_l) = \\ = 1 \rightarrow ↯ \rightarrow read, write \notin A^*[s,o]$$

$$AC_{i,l} = 1 \rightarrow б(U_i, O_l) = \\ = 0 \rightarrow ↯ \rightarrow read, write \notin A^*[s,o]$$

Тогда по основной теореме безопасности система безопасна.

Далее проведем верификацию построенной модели относительно модели take-grant [14, 15].

Расширенная модель состоит из следующих 6 команд, порождающих неявные информационные потоки: получение доступа для записи, получение доступа для чтения, получение доступа для чтения из другого субъекта, получение доступа для чтения из другого субъекта при доступе по записи, получение доступа для внесения изменений.

Основными элементами расширенной модели take-grant являются: $\Gamma(O, S, E)$ — компьютерная система, описываемое графом; O — множество объектов доступа; S — множество субъектов доступа S; E — множество ребер графа Γ; α — набор прав доступа.

Соотнесем элементы модели take-grant с элементами предлагаемой в данной работе модели:

- $S \sim U$
- $O \sim O \sim PC_i * M_j$
- $б \sim б$
- $E \sim AM$

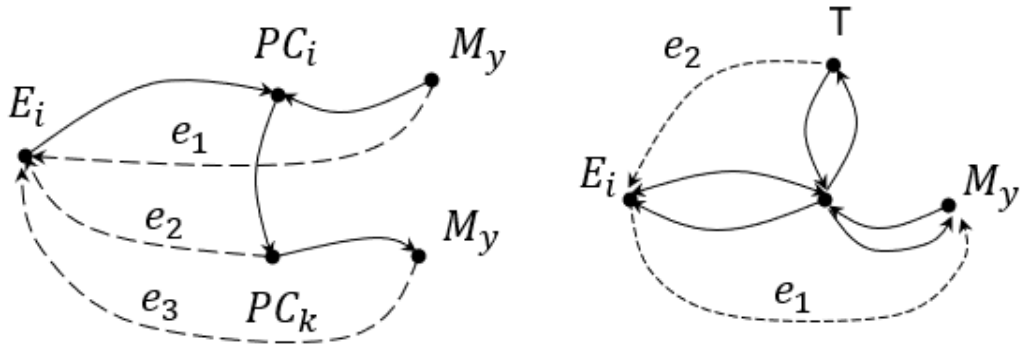

Рис. 1. Случаи утечки

Будем считать, что операции read и write не отличны в рамках модели.

Два случая утечки (рассматриваются случаи, когда у нас происходит маркирование документов) (Рис. 1).

В рамках 1 случая возникает несанкционированные каналы e_1, e_2, e_3, которые не могут быть устранены

В рамках 2 случая возникают несанкционированные каналы e_1, e_2, которые устраняются на шаге 9.

В рамках работы над моделью был реализован программный комплекс, осуществляющий мониторинг утечек конфиденциальных документов. Комплекс состоит из 4 компонент: программа маркировки документов, web-сервис мониторинга утечек, приложение администрирования и приложение уведомлений.

Программа маркировки документов. Предназначена для маркировки конфиденциальных документов. Осуществляет маркировку документов doc/docx, xls/xlsx, ppt/pptx. Пересылает информацию о результатах маркировки web-сервису.

Web-сервис мониторинга утечек. Предназначен для получения уведомлений от маркированных документов, обработки информации от приложения маркировки, предоставления информации о произошедших событиях. При получении информации от открытых документов или маркировщика заносит данную информацию в базы.

Приложение администрирования. Предназначен для взаимодействия администраторов с комплексом. Позволяет настраивать систему, получать подробную информацию о событиях открытия документов.

Приложение уведомлений. Предназначен для мгновенного отображения экстренных инцидентов, когда документ открывается на устройстве, которого нет в списке разрешенных.

ЛИТЕРАТУРА

1. Шугаев В.А. Классификация инсайдерских угроз информации/ Шугаев В.А., Алексеенко С.П.— Вестник Воронежского института МВД России.— 2020.— № 2.— С. 143–153
2. Калач А.В. Современное состояние утечек информации и объемов скомпрометированных данных в мире / Калач А.В., Пеев Д.Н.— Актуальные проблемы деятельности подразделений УИС — 2019.— С. 415–417
3. Исследование утечек информации ограниченного доступа в 2019 году. [Электронный ресурс].— Режим доступа: https://www.infowatch.ru/analytics/reports/27720, свободный (дата обращения: 18.12.2020).
4. Утечки информации ограниченного доступа: отчет за 9 месяцев 2020 г. [Электронный ресурс].— Режим доступа: https://www.infowatch.ru/analytics/reports/30708, свободный (дата обращения: 03.01.2021).
5. Калач А.В. Современное состояние динамики утечек и обеспечения безопасности информации в российской федерации / Калач А.В., Пеев Д.Н, Зыбин Д.Г.— Вестник Воронежского института МВД России.— 2018.— № 3.— С. 74–80
6. Боридько И.С. / Боридько И.С., Забелинский А.А., Коваленко Ю.И.— Безопасность информационных технологий.— 2012.— Т. 19.— № 3.— С. 20–24.
7. Баранкова И.И. Разработка системы wordsearch для защиты конфиденциальной информации от утечек / Баранкова И.И., Михайловна У.В., Лукьянов Г.И.— Вестник УРФО. Безопасность в информационной сфере — 2018.— № 1.— С. 14–18
8. Филяк П.Ю. Применение DLP-системы searchinform для обеспечения безопасности в сети internet / Филяк П.Ю., Старченко В.И., Цапегородцев А.В., Рашидов С.Х.У — Информация и безопасность — 2018.— Т. 21 — № 4.— С. 472–477

9. Морозов В.Е. Использование программного комплекса «Контур информационной безопасности SEARCHINFORM» для предотвращения утечек данных из корпоративных сетей/ В.Е. Морозов, А.В. Дроздов.— Преступность в сфере информационных и телекоммуникационных технологий: проблемы предупреждения, раскрытия и расследования преступлений.— 2015.— № 1.— С. 128–135
10. Trabelsi S. Monitoring Leaked Confidential Data. In2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS) 2019 Jun 24 (pp. 1–5). IEEE.
11. Malderle T, Wübbeling M, Knauer S, Meier M. Warning of Affected Users About an Identity Leak. InInternational Conference on Soft Computing and Pattern Recognition 2018 Dec 13 (pp. 278–287). Springer, Cham.Быстрицкий Н.Д. Особенности функционирования криптологического программного комплекса «Креветка» / Н.Д. Быстрицкий, Н.В. Макаров-Землянский, В.С. Назаров.— Экономика: вчера, сегодня, завтра.— 2019.— № 1–1.— С. 51–60
12. Bell, D.E. Looking Back at the Bell-LaPadula Model. Proceedings of the 21st Annual Computer Security Applications Conference: pp. 337–351. DOI:10.1109/CSAC.2005.37
13. Bishop, M. Computer Security: Art and Science.— Boston: Addison Wesley.— 2003.
14. Frank J., Bishop M. Extending The Take-Grant Protection System // Department of Computer Science.— University of California at Davis, 1984

Дальневосточный федеральный университет