References

1. Leiting A. Einführung von ERP-Systemen // Unternehmensziel ERP-Einführung. IT muss Nutzen stiften. Springer Fachmedien Wiesbaden, 2012. VII. P. 83–195. DOI: 10.1007/978-3-8349-4462-7_4.

2. Ebel B. Produktionswirtschaft. Verlag: Kiehl Friedrich Verlag G, 2009. 204 p. ISBN 10: 347070449X / ISBN 13: 9783470704494.

3. Cf. URL: https://www.duden.de/suchen/dudenonline/global%20player (access date: 22.05.2020).

4. Cf. URL: https://www.finance-magazin.de/finanzabteilung/controlling/sap-oracle-microsoft-welchererp-anbieter-ist-der-beste-1393901/ (access date: 22.05.2020).

5. Cf. URL: https://www.computerwoche.de/a/kampf-der-erp-titanen, 3223108 (access date: 22.05.2020).

6. Global 100 Software Leaders by revenue // IDC (Data, ranking, oftware 100, customized sort), 2014. URL: https://www.pwc.com/gx/en/industries/technology/publications/global-100-software-leaders/explorethe-data.html (access date: 22.05.2020).

7. Brocke J. vom, Simons A., Niehaves B., Riemer K., Plattfaut R., Cleven A. Reconstructing the giant: On the im- portance of rigour in documenting the literature search process // Proc. of the 17th European Conference on Information Systems (ECIS 2009), Verona, Italy, 2009. Vol. 9 (2009). P. 2206–2217.

8. Eine Zukunft ohne Papier // GmbH, TeDo Verlag (MES, ERP, SCM, PPS, CRM, BPM, PLM, Stellenmarkt, Management, System), 2014. URL: https://www.it-production.com/allgemein/fertigungsprozesseeinezukunft-ohne-papier (access date: 22.05.2020).

9. Vogel-Heuser B., Bauernhansl T., Hompel, M. Handbuch Industrie 4.0 Bd.4: Allgemeine Grundlagen. Wiesbaden: Springer Berlin Heidelberg, 2017. P. 574., P. 587. DOI: 10.1007/978-3-662-53254-6.

УДК 681.322.067
doi:10.18720/SPBPU/2/id20-205

Буянов Борис Яковлевич[1],
канд. техн. наук, ст. науч. сотр., доцент;
Бабичева Анастасия Сергеевна[2],
инженер-программист;
Чербаев Павел Олегович[3],
инженер-программист

РЕАЛИЗАЦИЯ ЗАЩИЩЁННОСТИ КЛАССА 1Б В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ ПОД УПРАВЛЕНИЕМ UNIX — ПОДОБНЫХ ОПЕРАЦИОННЫХ СИСТЕМ

[1] Московский технический университет связи и информатики,
Москва, Россия,
b.buyanov@gmail.com
[2, 3] АО «Ордена Трудового Красного Знамени научно-исследовательский институт автоматической аппаратуры им. академика В. С. Семенихина»,
Москва, Россия,
[2] Anastasija.Babicheva@yandex.ru, [3] CherPaUl1997@yandex.ru

Аннотация. Рассмотрены возможности построения безопасной автоматизированной системы управления специального назначения под управлением операционной системы семейства UNIX**.** В соответствии, с действующими в Российской Федерации ГОСТами и руководящими документами по обеспечению требуемого уровня защищенности автоматизированных комплексов, проводится сравнительная характеристика имеющегося функционала UNIX–подобных операционных системах. Произведен анализ основных уязвимостей UNIX– подобных операционных систем и предложены методы их устранения для достижения требуемого класса защиты информации.

Ключевые слова: сетевой комплекс специального назначения, класс защиты 1Б, библиотека PAM, мандатный доступ, модель Белла–ЛаПадулы, уязвимость.

Boris Y. Buyanov[1],
Candidate of Technical Sciences, Associate Professor;
Anastasija S. Babicheva[2],
Software Engineer;
Pavel O. Cherbaev[2],
Software Engineer

CLASS 1B SECURITY IMPLEMENTATION IN AUTOMATED SPECIAL PURPOSE SYSTEMS UNDER CONTROL OF UNIX-LIKE OPERATING SYSTEMS

[1] Moscow Technical University of Communications and Informatics,
Moscow, Russia,
b.buyanov@gmail.com
[2, 3] JSC "Order of the Red Banner of Labor Scientific Research Institute of Automatic Equipment named after Academician V. S. Semenikhin",
Moscow, Russia,
[2] Anastasija.Babicheva@yandex.ru, [3] CherPaUl1997@yandex.ru

Abstract. The possibilities of building a secure automated control system for special purposes under the control of the operating system of the UNIX family are considered. In accordance with the GOSTs and the governing documents in force in the Russian Federation to ensure the required level of security for automated systems, a comparative description of the existing UNIX functionality is carried out – similar operating systems. The analysis of the main vulnerabilities of UNIX - similar operating systems is carried out and methods for their elimination are proposed to achieve the required class of information protection.

Keywords: special-purpose network complex, protection class 1B, PAM library, mandatory access, the Bell–LaPadula model, vulnerability.

Введение

Достижение высокого уровня защиты информации, является основным критерием современных комплексов специального назначения. Прежде всего, это касается подсистем хранения информации.

Для обеспечения требуемого уровня информационной безопасности, необходимо корректно управлять процессом доступа к информационным ресурсам и иметь возможность разграничивать права доступа к ним, тем самым позволив работать только определённому кругу лиц. Для упрощения реализации безопасной системы, в Российской Федерации (РФ) были выпущенные специальные стандарты и руководящие документы, которые содержат определённый минимальный функционал, необходимый любой из автоматизированных систем (АС). Поэтому следует выполнять определенный стандарт правил, который позволил бы обеспечить минимальный уровень защиты данных. В комплексах специального назначения, обойтись только ГОСТами по защите от несанкционированного доступа (НСД) будет недостаточно, чтобы гарантировано защитить информацию от доступа к ней злоумышленника.

Ниже произведен анализ уязвимостей *NIX операционных систем, на основании которого, предлагаются методы, обеспечивающие защиту информации в АС обеспечивающих классу защиты 1Б, подсистемы управления доступом.

1. Классификация уровней защищенности АС

Рассматриваемые требования к защите информации от несанкционированного доступа, приведенные в данном разделе выдержки из РД, действуют уже на почти тридцати лет для всех ныне действующих и проектируемых АС учреждений, организаций и предприятий, обрабатывающих конфиденциальную информацию.

После реализации определенного класса защиты согласно РД, АС присваивается определенный уровень защищённости. Основным критерием выступает определение принадлежности классификации АС.

Основными этапами классификации АС являются:

- разработка и анализ исходных данных;
- выявление основных признаков АС, необходимых для классификации;
- сравнение выявленных признаков АС с классифицируемыми;
- присвоение АС соответствующего класса защиты информации от НСД [1].

При этом уровень защищенности АС, принципиально зависит от количества пользователей, работающих с данной системой. Если работает только один пользователь, то это класс защиты третьего уровня. Если же рассматривать АС при равных правах большего количества пользователей, то данная система отвечает требованиям второго уровня защиты информации. Если же в системе имеется множественный доступ пользователей с разными правами доступа, исходя из того, что АС работает с конфиденциальной информацией разного уровня. В данном случае требуется создания некоторой мандатной модели (основанной на четкой ор-

ганизации отношений между объектом и субъектом, главная цель, которой установка меток конфиденциальности, отвечающих за разрешение или запрещение взаимодействие объектов и субъектов). При использовании данного механизма будем иметь дело с первым, самым высшим уровнем защиты информации.

В каждой группе присутствует внутренняя градация, выделяется заглавными русскими буквами, к примеру, в первой группе наивысшая 1А, самая низшая 1Д. Классификация строится в зависимости от выполнения АС указанных в РД функций.

В данном случае, рассматривается пример реализации класса защиты 1Б, который является вторым по уровню защищённости для группы лиц с разными правами доступа.

2. Требования по защите от несанкционированного доступа к информации в АС класса защиты 1Б

Защита информации от НСД является составной частью общей проблемы обеспечения безопасности информации [1].

В общем случае, комплекс программно-технических средств и организационных (процедурных) решений по защите информации от НСД, реализуется в рамках системы защиты информации от НСД (СЗИ НСД), условно состоящей из следующих четырех подсистем:

– управление доступом;

– регистрации и учета;

– криптографической;

– обеспечение целостности.

Минимальный перечень требований, согласно РД, к реализации подсистемы управления доступом в АС первой группы, представлен в таблице 1.

Для выполнения требований к подсистеме управления доступом класса 1Б, должно осуществляться:

– идентификация и проверка подлинности субъектов доступа при входе в систему по идентификатору (коду) и паролю временного действия длиной не менее восьми буквенно-цифровых символов;

– идентификация терминалов, ЭВМ, узлов сети ЭВМ, каналов связи, внешних устройств ЭВМ по физическим адресам (номерам);

– идентификация программ, томов, каталогов, файлов, записей, полей записей по именам;

– контроль доступа субъектов к защищаемым ресурсам в соответствии с матрицей доступа;

– управление потоками информации с помощью меток конфиденциальности. При этом уровень конфиденциальности накопителей должен быть не ниже уровня конфиденциальности, записываемой на него информации [1].

Таблица 1

Требования к подсистеме управления доступом в АС первой группы

Требования	Классы				
	1Д	1Г	1В	1Б	1А
1.1 Идентификация, проверка подлинности и контроль доступа субъектов:					
– в систему;	+	+	+	+	+
– к терминалам, электронно-вычислительным машинам (ЭВМ), узлам сети ЭВМ, каналам связи, внешним устройствам ЭВМ	-	+	+	+	+
– к программам;	-	+	+	+	+
– к томам, каталогам, файлам, записям, полям записей.	-	+	+	+	+
1.2 Управление потоками информации	-	-	+	+	+

Этот минимум функций, обеспечивающий уровень безопасности класса 1Б, который должен присутствовать в АС.

3. Выбор операционной системы для АС класса защиты 1Б

В последнее время, в России сложилась тенденция продвижения отечественного программного обеспечения (ПО) на базе операционных систем Linux. Причина, прежде всего, заключается в доступности и возможностях быстро обновить программное обеспечение, что невозможно сделать во многих других операционных системах (ОС) (например, ОС Windows). С точки зрения безопасности, Linux также более приспособлена, что основано на особенностях построения архитектуры данного программного обеспечения. Требования защиты Windows отличаются от отечественных, и состав возможностей может серьёзно различаться (например, система мандатных разграничения прав). В итоге, с точки зрения финансовой составляющей ОС Linux дешевле своих конкурентов.

Возрастание требований по защите информации АС, привело к появлению на рынке систем на базе ОС Linux, предназначенных для обработки информации ограниченного доступа и сертифицированных по требованиям защиты информации российских систем сертификации, таких как: ФСТЭК России, Минобороны России и ФСБ России [2].

Данная ОС семейства UNIX носит название Astra Linux. Сертифицирована по классу защиты 1А, большая часть возможностей специально заточены под обеспечение безопасности. Однако и тут есть свои недостатки. Ограниченность в ресурсах, неудобство работы, большая цена поддержки и подготовки персонала для администрирования в ОС Astra Linux. В небольших компаниях использования данного продукта недопустимо по финансовым соображениям.

Поэтому самым верным будет использовать ОС Linux, и уже в ней настроить требуемый функционал согласно РД по АС. Большинство функций описанные в требованиях обеспечения класса 1Б, присутствуют в Linux. Но они находятся в отключенном состоянии или недостаточном для обеспечения защиты режиме и потому нуждаются в более детальных настройках. Уже дополнительный функционал придется писать собственноручно, используя имеющиеся возможности в ОС Linux.

4. Требования к идентификации, проверке подлинности и контроле доступа субъектов в системах семейства UNIX

Под идентификацией понимается проверка того, что пользователь является зарегистрированным в системе пользователем и имеет доступ к ней или другому объекту (например, к программе, папке и т.д.). Процесс прохождения подлинности связан с аутентификацией данного пользователя, использования средств, которые дают возможность подтвердить идентификатор. Контроль доступа субъекта, дальнейшая работа систем безопасности с авторизированным пользователям, предоставлением доступных полномочий и контроль над соблюдением механизмов безопасности (требования к паролям, проверка завершении сессии, модификация учетных записей и другие необходимые функции).

Большинство систем семейства UNIX, используют примерно одни и те же модули для идентификации и аутентификации. На таблице 2, пример для 3 видов ОС [3].

Таблица 2

Сравнение механизмов идентификации и аутентификации в системах семейства UNIX

Платформа	Идентификация	Аутентификация
AIX	LAM	LAM/PAM
Solaris	NSS	PAM
Linux	NSS	PAM

Кроме того, в большинстве UNIX систем используется механизм идентификации NSS (Name Service Switch), с возможностями которого можно воспользоваться для реализации подсистемы «Идентификация, проверка подлинности и контроль доступа субъектов».

Основные файлы необходимые для идентификации (/etc/passwd, /etc/shadow, /etc/group). Именно эти файлы содержат некоторую информацию об авторизированных пользователях, которые позволяют затем его аутентифицировать.

Структура файла passwd имеет примерно следующий вид:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
...
```

Каждая запись в этом файле прописана разделяющим символом, двоеточием и выделяется 7 частей:

– имя пользователя. Это поле содержит имя пользователя. Для операционной системы не важно, какое имя имеет пользователь, система ориентируется на идентификатор;

– поле пароля. Это поле в ранних версиях Linux содержало зашифрованный пароль, а теперь, когда была введена технология теневых паролей, в этом поле просто ставится *x*. Практического применения это поле не имеет;

– идентификатор пользователя (UID). Уникальный идентификационный номер пользователя. Используется для установки прав пользователя;

– идентификатор группы, к которой принадлежит этот пользователь (GID) – групповой идентификатор;

– поле комментария;

– полный путь к домашнему каталогу пользователя;

– путь к командной оболочке [4].

Данный файл можно использовать как список идентификаторов системы, так и для доступа к ОС. На подобии построен и файл group, он указывает принадлежность идентификатора к определенной группе.

Второй связанный с файлом passwd, файл shadow, структура имеет примерно следующий вид:

```
root:$1$pOy8fNrf$uOh/dQlI03BMIdEAhWrE.0:12369:0:99999:7
:::
bin:*:12245:0:99999:7:::
daemon:*:12245:0:99999:7:::
sync:*:12245:0:99999:7:::
...
```

Запись также разделена двоеточиями:

– имя пользователя. Это поле просто дублируется из файла passwd;

– хэш пароля. В Linux пароли не хранится в открытом виде и шифруется по специальному алгоритму;

– остальные три информационных поля содержат различную служебную информацию.

Данный файл является продолжением passwd, в котором хранятся все данные о паролях авторизированных пользователей. Файл доступен, только суперпользователю (root) на правах чтение. Тем самым контроль над этим файлом можно отдать системному администратору, в следствии чего, он уже будет контролировать записи в данном файле для обычных пользователей.

Помимо данных файлов, требуется добавить файл, который будет хранить предыдущие пароли, например, opasswd и файл sysconfig, который будет содержать дополнительные настройки политики паролей.

Для организации работы всей системы авторизации, основой будет являться библиотека PAM. Данная библиотека существует во всех UNIX системах, в том числе и в новых версия Linux Ubuntu. PAM (Pluggable Authentication Modules) – подгружаемые модули аутентификации. PAM является набором динамически подключаемых модулей, с помощью которых привилегированный пользователь может выбирать, как приложение должно осуществлять процесс аутентификации. Эта технология имеет два основных преимущества. Первым преимуществом является модульность приложений, поддерживающих PAM. Это означает, что для приложения, поддерживающего PAM, появляется возможность изменить механизм аутентификации пользователей без перекомпиляции программы, достаточно изменить конфигурационный файл PAM. Второе преимущество использования PAM заключается в том, что администратор системы получает полную свободу в выборе схемы аутентификации для каждого отдельного приложения [4].

На рисунке 1 изображена структурная схема работы библиотеки PAM.

Конфигурационный файлом выступает, файл, лежащий в каталоге «/pam.d b», который имеет примерно следующий вид:

```
    #%PAM-1.0
    auth required /lib/security/pam_securetty.so
    auth required /lib/security/pam_stack.so ser-
vice=system-auth
    auth required /lib/security/pam_nologin.so
    account required /lib/security/pam_stack.so ser-
vice=system-auth
    password required /lib/security/pam_stack.so ser-
vice=system-auth
    session required /lib/security/pam_stack.so ser-
vice=system-auth
    session required /lib/security/pam_limits.so
    session optional /lib/security/pam_console.so
    ...
```

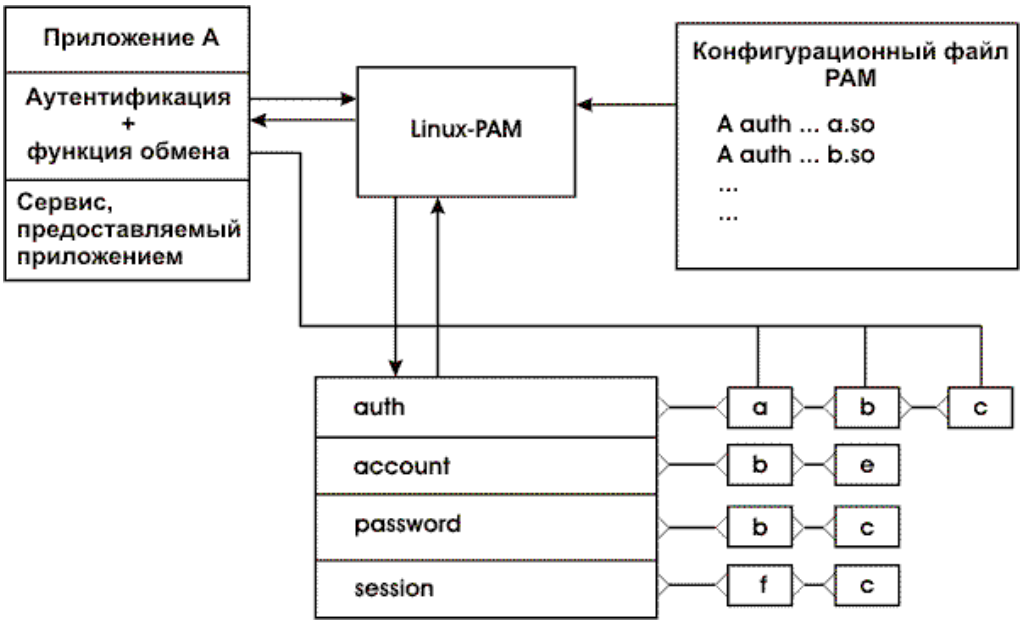

Рис. 1. Структурная схема взаимодействия приложения и библиотеки PAM

Правила записи в конфигурационный файл «/pam.d b»:

```
тип_модуля флаг_контроля путь_к_модулю параметры_модуля
```

Формат в виде искусственной таблицы в текстовом файле, количество столбцов 4, знак разделения между столбцами табуляция:

Каждый модуль (auth, account, password, session) и флаг (required, optional и другие) выполняет определённые действия согласно прописанным параметрам модуля, более подробно о каждом прописано в программных документациях Linux.

Все операции по аутентификации, шифрования пароля и его проверку, производит библиотека PAM. Библиотека Linux-PAM производит чтение параметров аутентификации приложения из конфигурационного файла и загружает необходимые модули в память. Затем загруженные модули попадают в одну из четырех управляющих групп и помещаются туда в порядке появления их в конфигурационном файле [4]. Именно конфигурационный файл и отвечает за расширенные политики пароля. Можно прописать и срок пароля, и запрет доступа при определенном количестве не успешных вводов пароля пользователя в зависимости от требуемой задачи.

Модули библиотеки находятся в каталоге «/lib/security», но могут располагаться и в другой директории, предварительно изменив ее в конфигурационном файле.

Для доступа к файлам, каталогам можно воспользоваться механизмами доступа ОС Linux. Для каждого файла можно прописать, доступен ли файл, данному пользователю указав права типа “rwx”, с помощью

стандартных команд в Linux. Существует метод подмены рабочего стола собственным (пустым) на стадии загрузки, тем самым также обеспечивая требуемую защиту, но лишая возможности организовать мандатную работу с файлами (второй пункт в РД).

5. Требование к управлению потоками информации

Данное требование, связано с организации мандатного доступа к файлам и организации передачи данных разного уровня секретности.

Мандатный доступ к файлам можно организовать с помощью встроенного в ОС модуля Smack, предварительно перекомпилирую ядро. Требуется найти в разделе Networking support -> Networking options -> CP/IP networking этот модуль.

Данный модуль основан на ключевой модели компьютерных систем с мандатным управлением доступом – модель Белла–ЛаПадулы [7].

Система обеспечения мандатного контроля доступа “Smack” состоит из трех основных компонент:

– компонент ядра, который реализован как модуль “Linux Security Modules”;

– скрипт загрузки и вспомогательные утилиты, предназначенные для загрузки базовых настроек, проверки корректности атрибутов Smack для отдельных файлов устройств;

– набор исправлений к пакету GNU Core Utilities, благодаря которому некоторые стандартные утилиты (типа ls) могут оперировать расширенными атрибутами файлов, используемых системой безопасности [5].

После компиляции и установки ядра будет получен архив с загрузочными скриптами, которые в соответствии с документацией, требуется правильно установить в системные папки.

Основной файл по созданию мандатных меток, расположен в директории «/etc/smack/accesses». Часть меток зарезервировано под стандартные, остальные можно присваивать для любых объектов или группы объектов. Строится правило по методике <объект> – <субъект> – <тип доступа> там же находится файл определяющий степень важности метки, к примеру, пользователь с самой высокой меткой будет иметь доступ ко всем файлам всех видов меток, за исключением запрещающих зарезервированных.

Управление потоками для информации, передающейся по сети, можно использовать механизм мандатных меток, прописанный в ГОСТ Р 58256–2018. С помощью программных средств при этом целостность поля Опции, который и отвечает за классификацию мандатных меток, не должна быть нарушена.

При передаче информации классификационные метки должны размещаться в каждом заголовке IP-пакета в поле Опции с типом Безопасность.

Классификационная метка при передаче информации представляет собой совокупность полей, входящих в поле Опции заголовка IP-пакета.

При отсутствии поля «Опции» в составе заголовка IP-пакета следует считать, что данный пакет имеет метку с нулевым значением и не имеет категорий конфиденциальности.

Значения полей, входящих в поле «Опции», совокупность которых определяет значение классификационной метки, устанавливаются в соответствии с таблицей 3 [6].

Таблица 3

Значения полей, входящих в поле «Опции»

Поля, входящие в поле Опции	TYPE (8 бит)	LENGTH (8 бит)	CLASSIFICATION LEVEL (8 бит)	PROTECTION AUTHORITY FLAGS (11+ байт)
Значение поля	10000010	XXXXXXXX	10101011	AAAAAAA1110] AAAAAAA0

Значение поля TYPE равно 130 в десятичном представлении, что определяет тип поля Опции – Безопасность (Security).

Поле LENGTH содержит значение длины поля Опции в октетах. Минимальная длина поля Опции – 3 октета (байта), включая поля TYPE и LENGTH. Поле PROTECTION AUTHORITY FLAGS может отсутствовать. Значение поля LENGTH не должно превышать 40 октетов.

Значение поля LENGTH меньше 3 октетов должно обрабатываться как ошибка.

В поле CLASSIFICATION LEVEL всегда указывается значение 10101011 (Unclassified).

Поле PROTECTION AUTHORITY FLAGS имеет переменную длину. Младший бит каждого октета (байта) используется для индикации наличия следующего октета. Если бит равен 1, есть следующий октет, если бит равен 0 – октет последний. Ситуации, когда в соответствии со значением поля LENGTH октет является последним, но его младший бит не равен 0, либо октет не является последним, а младший бит равен 0, должны обрабатываться как ошибки.

Классификационная метка должна представлять собой структуру, которая включает 8 бит для кодирования уровня (беззнаковое целое число. 256 возможных значений) и до 251 бита для кодирования категорий. При этом кодирование категорий рекомендуется осуществлять с использованием 64–разрядной битовой маски, заканчивающейся младшим битом.

Значение классификационной метки, равное нулю, соответствует информации, для которой не определены уровни конфиденциальности [6]. Более детально с примерами описано в ГОСТ Р 58256–2018.

С помощью данного поля можно определить сообщения разного степени секретности, каждый вид имеет свой специальный номер, прописанный в соответствии с ГОСТом. При этом программные средства строится на основе ключевой модели с мандатным управлением доступа – модели Белла–ЛаПадулы [7]. Принцип: невозможно возникновение информационных потоков от объектов с большим уровнем конфиденциальности к объектам с меньшим уровнем конфиденциальности. При передаче просматривается уровень секретности пакета и уровень секретности пользователя, сравнивается, и запрещает передавать пользователю пакеты уровень секретности, которых выше, чем получателя.

6. Выводы

Идентификация, проверка подлинности и контроль в систему, к терминалам, ЭВМ, узлам сети ЭВМ, каналам связи средствами стандартных файлов etc/passwd, /etc/shadow, /etc/group и другими файлами, если потребуется более надежная защита и средствами библиотеки PAM. Так же можно использовать средства NSS.

Идентификация, проверка подлинности и контроль внешним устройствам ЭВМ, к приложениям к томам, каталогам, файлам, записям, полям записей следует так же использовать вышеупомянутые средства, плюс использования стандартных методов управления доступом к объектам в Linux.

Управление потоками информации на локальном компьютере можно осуществить с помощью модуля “Smack”, тем самым определив мандатные метки для каждого пользователя. Есть и другие встроенные методы в Linux для реализации мандатных меток, например, “SELinux”. Так же можно реализовать данную процедуру ручным способом.

Управление потоками информации в сети, наиболее целесообразно использовать аппаратно-программные средства, добавляющие дополнительную информацию в пакеты, позволяющие классифицировать, к какому типу секретности относить отправленную информацию. При построении придерживаться модели Белла-ЛаПадулы, на основе которой построены большое количество сетевых систем специального назначения.

7. Анализ уязвимостей в системе безопасности АС

Любая задача злоумышленника – совершение несанкционированного доступа к системе, с целью получения личной выгоды. Эту выгоду получает путем получения конфиденциальной информации, и в дальнейшем использование ее в своих целях. Именно для совершения таких дей-

ствий и требуется найти наиболее уязвимые места в системе, чтобы каким-либо образом обойти систему защиту или изменить ее под себя.

Количество уязвимостей, в большей мере, зависит от качества работы системного администратора, правильной настройки ПО и его использования. Уязвимости могут проявиться на любом этапе реализации системы безопасности. Конкретного названия для данных уязвимостей нет. Они устраняются путем определения источников проблем. Существуют также уязвимости и в самой ОС.

Dirty COW – локальная уязвимость возникает во время копирования какого-либо объекта при процессе его же записи. Благодаря ей каждый непривилегированный пользователь может получить доступ к ОС. На некоторых старых ядрах Linux, вполне допустимо наличия еще не решеной данной уязвимости.

Уязвимость нулевого дня ядра – локальная уязвимость, которая предоставляла возможность повысить права текущего пользователя до прав суперпользователя “root”. Первопричиной является допущение ошибки с криптографическими данными ядра, которые хранятся в памяти. Данная уязвимость, прежде всего, актуальна для серии ОС Linux Ubuntu.

Указаны только пару уязвимостей локального характера, этим список не заканчивается, новые уязвимости, могут проявляться при воздействии пользователя с ОС. Поэтому, во избежание потери информации, в управлении механизмами безопасности, должен использоваться высококвалифицированный работник, умеющий работать, прежде всего, с самой ОС.

Заключение

В данной статье были рассмотрены возможные средства для реализации подсистемы управления контроля в АС специального назначения с обеспечением ее, класса защиты уровня 1Б. Подробно описаны методы по построению данного системного модуля. Так же были указаны некоторые элементы угрозы безопасности в ОС Linux.

Список литературы

1. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. Утверждено решением председателя Государственной технической комиссии при Президенте Российской Федерации от 30 марта 1992 г. [Электронный ресурс]. URL: https://fstec.ru/index?id=384:rukovodyashchij-dokument-reshenie-predsedatelya-gostekhkomissii-rossii-ot-30-marta-1992-g (дата обращения 11.05.2020).

2. Операционная система специального назначения Astra Linux Special Edition. [Электронный ресурс]. URL: http://www.cio-sibir.ru/files/Meet/2016/2016-10-07-Astra_Linux.pdf (дата обращения: 11.05.2020).

3. Галгали П., Гайтонде Р. Сравнение систем безопасности в AIX, Linux и Solaris // IBM developerWorks. 15.07.2007. [Электронный ресурс]. URL: https://www.ibm.com/developerworks/ru/library/au-compaixsolaris/index.html (дата обращения: 11.05.2020).

4. Исследование уровня безопасности операционной системы Linux. [Электронный ресурс]. URL: https://www.bestreferat.ru/referat-52957.html (дата обращения: 11.05.2020).

5. Ивашко Е. Система мандатного контроля доступа Smack // IBM developerWorks. 26.10.2010. [Электронный ресурс]. URL: https://www.ibm.com/developerworks/ru/library/l-apparmor-6/ (дата обращения: 11.05.2020).

6. ГОСТ Р 58256-2018. Управление потоками информации в информационной системе. Формат классификационных меток. Изд. офиц. М.: Стандартинформ, 2018. 8 с.

7. Девянин П. Модели безопасности компьютерных систем. Управление доступом и информационными потоками. 2-е изд., перераб. и доп. М.: Горячая линия–Телеком, 2013. 338 с.

УДК 004
doi:10.18720/SPBPU/2/id20-206

Сараджишвили Сергей Эрикович[1],
канд. техн. наук, доцент, доцент;
Морозов Юрий Алексеевич[2],
аспирант

ОСОБЕННОСТИ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ С ИСПОЛЬЗОВАНИЕМ LINKED OPEN DATA

[1, 2] Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Россия,
[1] SSaradg@yandex.ru, [2] stonefiz@gmail.com

Аннотация. В работе рассматриваются особенности обучения нейронных сетей с использованием открытых связанных данных. В рамках исследования проведен обзор публикаций, посвященных вопросам в этой области. В результате был описан подход обработки связанных данных для дальнейшего обучения и проведено тестовое обучение.

Ключевые слова: связанные открытые данные, машинное обучение, семантический веб, RDF.