УДК 004.451.56 Вестник УрФО № 2(40) / 2021, с. 22–30

Окороков В. А., Лащук Д. Е.

DOI: 10.14529/secur210203

ПРИМЕНЕНИЕ СХЕМЫ МНОГОУРОВНЕВОГО ДОСТУПА ДЛЯ ОРГАНИЗАЦИИ ЗАЩИТЫ КОНФИДЕНЦИАЛЬНЫХ ДАННЫХ

Предотвращение утечек конфиденциальной информации, хранящейся и обрабатываемой вычислительными системами, является одной из важнейших проблем, с которыми приходится сталкиваться при обработке данных с ограниченным доступом, в частности персональных данных. К настоящему моменту существует множество примеров, когда конфиденциальные данные похищаются злоумышленниками и используются ими в преступных целях или оказываются доступными широкому кругу пользователей Internet. Данное обстоятельство показывает необходимость разработки надежных механизмов, обеспечивающих разграничение доступа к данным, что позволяет минимизировать ущерб, возникающий из-за компрометации конфиденциальной информации.

Базовые механизмы, обеспечивающие разграничения доступа, обычно встроены в ядро операционной системы (ОС) и требуют значительных усилий для их реализации и настройки в рамках конкретного экземпляра ОС. В статье рассматривается один из основных методов разграничения доступа, который используются в защищенных операционных системах, а именно метод многоуровневого доступа. Реализация метода требует достаточно больших усилий разработчиков и зачастую связана с необходимостью модификации существующего прикладного программного обеспечения. Отмечается актуальность данной тематики для изучения курса «Безопасность операционных систем» студентами, обучающимися на направлениях, связанных с обеспечением информационной безопасности и защиты информации в вычислительных системах.

В рамках статьи рассмотрены различные подходы к реализации многоуровневого доступа и отмечены основные проблемы его реализации [1]. Рассмотрено также влияние ограничений для субъектов, имеющих различные уровни доступа, на основе использования дополнительных критериев безопасности для каждого субъекта и модификации методов трансформации разрешённых потоков информации в системе.

Ключевые слова: *операционные системы, доступ к информации, многоуровневый доступ, методы мандатного доступа, информационная безопасность.*

APPLICATION OF A MULTI-LEVEL ACCESS SCHEME FOR PROTECTING CONFIDENTIAL DATA

Preventing leaks of confidential information stored and processed by computer systems is one of the most important problems that one has to face when processing data with limited access, in particular personal data. By now, there are many examples when confidential data is stolen by intruders and used by them for criminal purposes or becomes available to a wide range of Internet users. This circumstance shows the need to develop reliable mechanisms that ensure the delimitation of access to data, which makes it possible to minimize the damage arising from the compromise of confidential information.

The basic mechanisms that provide access control are usually built into the operating system (OS) kernel and require significant effort to implement and configure within a specific OS instance. The article discusses one of the main methods of access control, which are used in protected operating systems, namely the multilevel access method. The implementation of the method requires a fairly large effort of developers and is often associated with the need to modify the existing application software. The relevance of this topic for the study of the course "Security of operating systems" by students enrolled in areas related to ensuring information security and information protection in computing systems is noted.

Within the framework of the article various approaches to the implementation of multilevel access are considered and the basic problems of its implementation are noted [1]. The influence of restrictions for subjects with different levels of access is also considered, based on the use of additional security criteria for each subject and modification of methods for transforming allowed information flows in the system.

Keywords: *operating systems, access to information, multi-level access, mandatory access methods, information security.*

Введение

В настоящее время проблема разграничения доступа в защищенных операционных системах является одной из самых острых. В связи с тем, что вычислительные системы обычно рассчитаны на работу в многопользовательском режиме, необходимо корректно обеспечивать поддержку разграничения доступа пользователей к ресурсам, связанным с хранением и обработкой данных. Важно исключить возможность несанкционированного доступа к конфиденциальной информации со стороны пользователей, которые не имеют таких полномочий [1].

В системах защиты информации должны быть реализованы два простых, но очень важных правила:

- компактность;
- простота.

Система защиты не должна заметно снижать производительность вычислительных систем, а также затруднять деятельность пользователей. При реализации этих критериев следует обеспечить контроль доступа и защиту данных [1].

Важность проблемы разграничения доступа в защищенных операционных системах (ОС) требует повышенного внимания к изучению существующих методов в рамках курса «Безопасность операционных систем». В данной статье рассматриваются методические основы и базовый теоретический материал, обеспечивающий изучение методов мандатного разграничения доступа в рамках упомянутого курса.

Многоуровневый доступ

Механизм дискреционного управления доступом, который изначально использовался в ОС семейства Linux, позволяет пользователям, владельцам файлов, явным образом указать возможность доступа субъектов, т.е. процессов, запущенных различными пользователями системы, к объектам (сущностям), которые могут быть файлами, папками и другими компонентами вычислительной системы. Однако данный механизм не предназначен для регулирования передачей информации между субъектами и объектами, содержащими данные различных уровней конфиденциальности. Например, если данные, хранящиеся в объекте с высоким уровнем конфиденциальности прочитаны процессом, имеющим высокий уровень доступа, они могут быть записаны в любой другой, доступный этому процессу на запись объект, который может иметь низкий уровень конфиденциальности, а механизм дискреционного управления доступом не может этому воспрепятствовать. Таким образом, в рамках использования механизма дискреционного управления доступом возможность изменения степени конфиденциальности информации при ее передаче от одного объекта другому может являться причиной утечки конфиденциальных данных.

Для регулирования потоков конфиденциальной информации используется многоуровневая (мандатная) модель доступа, которая предполагает, что все субъекты системы разбиваются на конечное число групп, в соответствии с уровнем допуска к информации, а объекты (сущности) аналогичным образом группируются по признаку конфиденциальности. Число групп субъектов и объектов совпадают. Предполагается также, что группы могут быть линейно упорядочены по убыванию уровня допуска субъекта и степени секретности объекта.

Детализация доступа внутри одного уровня обычно реализуется на базе использования системы категорий сущностей, для которых доступ может выполняться. Например, в рамках информационной системы может храниться секретная информация о характеристиках танков и ракет. При этом один субъект должен быть допущен только к информации о танках, а другой к информации о ракетах.

В защищенных ОС поддержка схемы многоуровневого управления доступом выполняется путем связывания с субъектами и объектами вспомогательных структур данных, называемых метками безопасности [2].

• Для субъектов доступа в метке содержится информация относительно уровней допуска субъекта, а также перечень категорий, которые в рамках указанного интервала уровней допуска, разрешены для просмотра субъектом.

• Для объектов доступа в метке указывается уровень конфиденциальности (секретности) информации и категория содержащихся в объекте данных.

Системы многоуровневой защиты основаны на использовании двух базовых моделей: Белла-ЛаПадулы и Биба, которые регулируют потоки информации внутри системы [1].

Модель Белла-ЛаПадулы

Наибольшей эффективностью с точки зрения сохранения конфиденциальности информации обладает модель Белла-ЛаПадулы [2].

Модель Белла-ЛаПадулы использует следующие принципы организации потоков данных в системе:

• Процесс пользователя (субъект доступа) может записать данные в некоторый объект, только в том случае, если его уровень допуска будет меньше или равен уровню конфиденциальности объекта.

• Процесс пользователя (субъект доступа) может читать данные, находящиеся в объекте, только в том случае, когда его уровень допуска больше или равен уровня конфиденциальности (секретности) объекта.

Таким образом, предотвращается передача секретных документов на более низкие уровни допуска.

В рамках модели создаются благоприятные условия для утери и нарушения целостности данных, поскольку данные пользователей высших уровней секретности могут несанкционированно модифицироваться или уничтожаться пользователями низших уровней секретности [2].

Как показано на рисунке 1, в рамках модели Белла-Ла Падулы передача информации осуществляется от объектов с низким уровнем секретности к объектам с высоким уровнем секретности.

Например, субъект X с уровнем доступа «Совершенно секретно» может читать данные, хранящиеся в объекте с уровнем секретности «Секретно» или «Совершенно секретно», а записывать только в объекты с уровнем

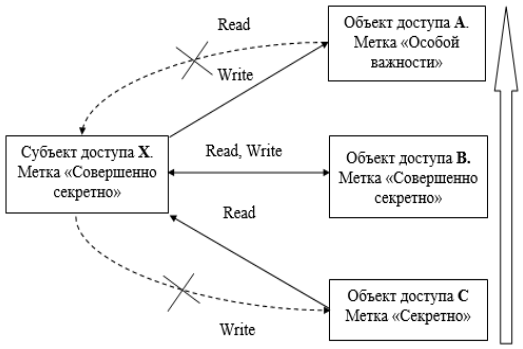

Рис. 1. Потоки данных в модели Белла-ЛаПадулы

секретности«Совершенно секретно» или «Особой важности».

Модель Биба

Модель Биба многоуровневого управления доступом характеризуется следующими базовыми свойствами [2]:

• Процесс пользователя (субъект доступа) может прочитать данные из некоторого объекта, только в том случае, если его уровень допуска будет меньше или равен уровню конфиденциальности объекта.

• Процесс пользователя (субъект доступа) может записывать данные, в некоторый объект, только в том случае, когда его уровень допуска больше или равен уровня конфиденциальности (секретности) объекта.

Таким образом, предотвращается модификация или уничтожение документов более высоких уровней конфиденциальности.

Следует отметить, что рассмотренные модели многоуровневой защиты несовместимы, т.к. позволяют обеспечить либо конфиденциальность, либо целостность данных.

Недостатки рассмотренных моделей можно частично устранить, если использовать вместо операции записи операцию добавления add. При добавлении данных потерь информации не происходит, что позволяет обеспечить целостность данных [4].

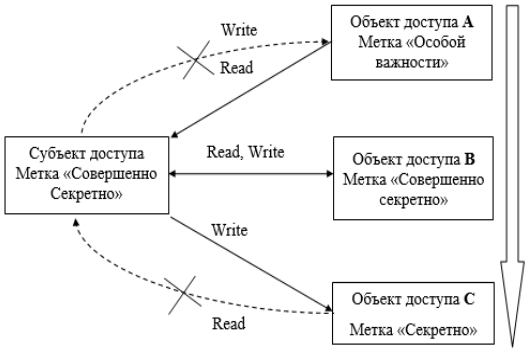

Рис. 2. Потоки данных в модели Биба

Как видно на рисунке 2, в рамках модели Биба передача информации осуществляется от объектов с высоким уровнем секретности к объектам с низким уровнем секретности. Такое направление потока данных в значительной степени гарантирует сохранение целостности информации, но при этом возможна утечка данных с высоким уровнем конфиденциальности (секретности).

Проблемы реализации мандатного управления доступом в защищенных ОС

В защищенных ОС механизм мандатного управления доступом является базовым механизмом разграничения доступа, однако при реализации и настройке этого механизма следует иметь в виду, что он не гарантирует полной защиты данных.

Существуют стандартные приемы взлома защиты, например, злоумышленник может найти в системе объект, в который возможна запись данных, и который не входит в область действия мандатного управления доступом. Такой объект может оказаться, например, в каталоге временных файлов /tmp.

Злоумышленник может также найти возможность для реализации передачи данных в оперативной памяти, например используя механизм разделяемой памяти, причем этот обмен не контролируется или некорректно контролируется механизмом защиты ОС. Часто такого рода обмен данными реализуются в программах, обеспечивающих графический интерфейс современных ОС. Возможен неконтролируемый обмен данными при использовании отладчиков, входящих в интегрированные среды программирования, а также при использовании процессами встроенных в ОС примитивов, обеспечивающих взаимодействие процессов.

На практике обычно не удается достигнуть абсолютной надежности системы мандатного управления доступом. Однако, можно либо значительно сократить возможности создания злоумышленником средств обмена данными, либо, либо сделать пропускную способность таких средств недостаточной для организации хищения конфиденциальных данных [3].

Одной из существенных проблем реализации мандатного доступа является возможность существования двунаправленных информационных потоков. Особенно часто с проблемой их реализации приходится сталкиваться при организации запросов к удаленному серверу, для которого обычно за-

прос на чтение данных и ответ пересылаются по одному каналу связи. В этом случае уровень конфиденциальности данных при передаче по сети может искажаться из-за различных уровней доступа процессов клиента и сервера. Например, если процесс с высоким уровнем доступа посылает запрос к данным, находящимся на сервере с низким уровнем доступа, то такой запрос не может быть обработан сервером из-за недостаточного уровня доступа. Если сервер имеет высокий уровень доступа, то все данные, которые он возвращает процессу будут также иметь высокий уровень конфиденциальности.

Довольно существенные проблемы возникают из-за возможной несовместимости мандатной модели с прикладным программным обеспечением. Например, для работы текстового редактора с документами, имеющими высокий уровень конфиденциальности, он должен иметь высокий уровень доступа. В тоже время при сохранении редактором документа с произвольным уровнем конфиденциальности, этому документу будет присваиваться уровень конфиденциальности, соответствующий уровню доступа редактора. Коррекция возможных ошибок в определении уровня конфиденциальности требует доработки прикладного программного обеспечения, что может привести к появлению запрещенных потоков данных.

К аналогичным проблемам может привести обращение субъектов с различными уровнями доступа к системным объектам или организации асинхронного доступа при реализации операций ввода-вывода.

Мандатное управление доступом в защищенных операционных системах

Основной задачей, решаемой путем реализации мандатного управления доступом в защищённых ОС, является уменьшение возможностей по созданию и эксплуатации средств, использующихся для хищения конфиденциальной информации, а также существенное снижение доступности их реализации злоумышленником. При этом даже при наличии уязвимостей в системе защиты снижается риск утечки конфиденциальных данных в результате ошибочных действий пользователя.

Злоумышленник, пытающийся преодолеть систему защиты, должен затратить значительные усилия на изучение системы и поиск уязвимостей. В результате вредоносное программное обеспечение, целью разработки которого является реализации хищения конфиденциальных данных, в условиях мандатного управления доступом оказывается практически бесполезным.

Сложные модели управления доступом в ОС семейства Linux иногда реализуются на базе использования пакета Security Enhanced Linux (SELinux). В современных защищенных ОС данный пакет обычно не используется, поскольку требует большой объем работ по проверке его корректности. Кроме того средства, которые используются для описания политики безопасности в рамках данного пакета, отличаются громоздкостью и их использование крайне неудобно. И главное, в пакет SELinux не включены утилиты, позволяющие обеспечивать контроль целостности информации в рамках многоуровневой модели доступа.

Одним из примеров успешной реализации мандатного доступа в защищенных ОС является операционная система специального назначения Astra Linux [5]. В рамках этой ОС реализация мандатного управления доступом базируется на использовании модуля подсистемы безопасности PARSEC, который совместно с подключаемыми модулями аутентификации PAM обеспечивает управление политикой безопасности ОС. Вместе с мандатным управлением доступом подсистема безопасности реализует мандатный контроль целостности и базовые функции аудита.

Для определения в некотором экземпляре ОС наименований мандатных уровней используется файл /etc/parsec/mac_levels, например:

Секретно:0

Совершенно секретно:1

Особой важности:2

Для определения именования неиерархических категорий используется файл /etc/parsec/mac_categories, например:

Танки:0

Ракеты:1.

Указанные файлы позволяют определить базовые атрибуты мандатной системы управления доступом.

При реализации иерархических категорий используется битовая маска, указывающая к каким категориям относится информация, хранящаяся в данном объекте. Количество бит в маске соответствует числу категорий, например для двух категорий из примера приведенного выше возможны следующие значения битовой маски:

00 - информация, хранящаяся в объекте, не имеет отношения ни к танкам, ни к ракетам;

01 - информация, хранящаяся в объекте, имеет отношения к танкам, но не к ракетам;

10 - информация, хранящаяся в объекте, имеет отношения к ракетам, но не к танкам;

11 - информация, хранящаяся в объекте, имеет отношение и к танкам, и к ракетам.

Значения параметров, которые были назначены субъектам доступа, и которые отображаются в соответствующих учетных записях, перечисляются в каталоге /etc/parsec/macdb. Структура каталога организована таким образом, что каждому пользователю, у которой указан отличный от нуля мандатный уровень или перечень неиерархических категорий, содержащий, по меньшей мере, одну категорию, создается отдельный текстовый файл, имя которого совпадает с идентификатором пользователя UID (User IDentifier). Файл содержит строку вида [5]:

<uname>:<umin_level>:<umin_categories>:<umax_level>:<umax_categories>

где:

uname - имя учётной записи пользователя;

umin_level - минимальный уровень допуска, доступный процессам, запущеным от имени учётной записи пользователя;

umin_categories – определяет минимальное множество неиерархических категорий, установленных для заданной учётной записи пользователя;

umax_level - максимальный уровень допуска, доступный процессам, запущеным от имени учётной записи пользователя;

umax_categories – определяет максимальное множество неиерархических категорий, доступных для некоторого пользователя.

Файл /etc/parsec/mac [5] содержит атрибуты суперпользователя root. Формат файла аналогичен формату файла обычного пользователя.

Метки, задающие конфиденциальность объектов, по умолчанию задаются следующим образом [5]:

1. Корневому каталогу присвоен наивысший уровень конфиденциальности, который может быть назначен в конкретной ОС (по умолчанию 3).

2. Предполагается возможность сохранения объектов, для которых заданы все возможные неиерархические категории.

3. Задаются атрибуты CCNR (определяет, что каталог может содержать сведения об объектах с различными метками безопасности, но не большими, чем его собственная метка) и CCNRI (определяет, что контейнер может содержать сведения об объектах с различными уровнями целостности, но не большими, чем его собственный уровень целостности).

Данный набор настроек фактически подразумевает, что в файловая система может сохранять любые данные с произвольными мандатными метками, при этом уровень конфиденциальности объектов не может превосходить некоторого максимально возможного значения, которое назначается корневому каталогу. Процесс с произвольным уровнем доступа, включая минимальный, может читать данные, находящиеся в каталоге, находящемся на вершине иерархии.

Стандартным директориям, например /bin, /etc, /lib, /usr и т.д., которые обеспечивают функционирование системы, назначается низший уровень конфиденциальности и пустая маска неиерархических категорий. В таком случае оказывается невозможным создание в перечисленных каталогах объектов с высоким уровнем конфиденциальности. Данное обстоятельство позволяет устранить значительную часть проблем, возникающих из-за возможной несовместимости системного и прикладного ПО с мандатным управлением доступом.

В параметрах каталога /dev, предназначенного для хранения системных файлов устройств, содержащих описания и параметры устройств, входящих в состав вычислительной системы, обычно назначается наивысший уровень конфиденциальности и устанавливается атрибут CCNR. Это позволяет создавать средствами ОС устройства, которые могут быть использованы для вывода конфиденциальных данных на внешние элементы вычислительной системы.

Основы администрирования мандатного управления доступом и контроль целостности в защищенных ОС

Возможности администрирования подсистемы, обеспечивающей мандатное управление доступом ОС Astra Linux, реализуются с помощью вспомогательной программы «Управление политикой безопасности», имеющей графический интерфейс. Данная программа может быть запущена с помощью пункта «Настройки» главного пользовательского

меню. В тоже время существует альтернативный способ управления уровнями доступа пользователей, который реализуется утилитой pdp-ulbls. При использовании этой утилиты возможно изменение уровней доступа субъектов и множества доступных им категорий. Наиболее часто использующийся набор параметров данной утилиты перечислен ниже:

pdp-ulbls -m<smin>:<smax> -c<cat-smin:cat-smax> <username>,

где:

smin - минимальный уровень доступа, задаваемый для пользователя;

smax - максимальный уровень доступа, задаваемый для пользователя;

cat-smin – определяет минимальный набор неиерархических категорий, задаваемый для учётной записи пользователя;

cat-smax - определяет максимальный набор неиерархических категорий, задаваемый для учётной записи пользователя.

<username> - имя пользователя.

Для решения задач контроля целостности в защищенных ОС реализованы следующие основные утилиты [5]:

• средство подсчета контрольных сумм файлов;

• средство контроля соответствия дистрибутиву;

• средства регламентного контроля целостности.

Утилита командной строки gostsum, входящая в набор утилит ОС, используется ля подсчета контрольных сумм файлов и оптических носителей.

Синтаксис утилиты командной строки х5ъ:

gostsum [КЛЮЧИ]... [ФАЙЛ]

Основные параметры утилиты [5]:

--gost-2012 - устанавливает, что будет использован алгоритм ГОСТ Р 34.11-2012 [6] с длиной хэш-кода 256 бит (по умолчанию);

--gost-2012-512 - устанавливает, что будет использован алгоритм ГОСТ Р 34.11-2012 с длиной хэш-кода 512 бит

-b - устанавливает размер блоков, которыми будет считываться файл

-o - задает имя файла для вывода контрольной суммы (по умолчанию - стандартный поток вывода);

-d - задает имя файла устройства чтения оптических дисков (файла с образом оптического диска) для подсчета контрольной суммы;

П р и м е р

Подсчет контрольной суммы оптического носителя

gostsum -d /dev/cdrom

Средство контроля соответствия дистрибутиву [5] предоставляет возможность контроля соответствия объектов файловой системы дистрибутиву ОС. Для обеспечения контроля целостности объектов в состав дистрибутива входит файл gostsums.txt со списком контрольных сумм по ГОСТ Р 34.11-2012 [6] с длиной хэш-кода 256 бит для всех файлов, входящих в пакеты программ дистрибутива.

Средства регламентного контроля целостности

Организация регламентного контроля целостности ОС, прикладного ПО и СЗИ обеспечивается набором программных средств на основе «Another File Integrity Checker» [5]. В указанном наборе реализована возможность для проведения периодического (с использованием системного планировщика заданий cron) вычисления контрольных сумм файлов и соответствующих им атрибутов расширенной подсистемы безопасности PARSEC (мандатных атрибутов и атрибутов расширенной подсистемы протоколирования) с последующим сравнением вычисленных значений с эталонными. В указанном наборе программных средств реализовано использование библиотеки libgost, обеспечивающей подсчет контрольных сумм в соответствии с ГОСТ Р 34.11-94 [6].

Эталонные значения контрольных сумм и атрибутов фалов хранятся в базе данных. База контрольных сумм и атрибутов может быть создана при помощи команды:

afick -i

Для вычисления контрольных сумм могут использоваться алгоритмы: MD5-Digest, SHA1 и ГОСТ Р 34.11-2012 с длиной хэш-кода 256 бит.

Заключение

В статье рассмотрен базовый теоретический материал, касающийся организации защиты информации в операционных системах семейства Linux. Отмечено, что использование традиционной для UNIX-подобных систем методики разграничения доступа к данным, основанной на дискреционной модели, не обеспечивает управления потоками информации в системе, что может стать причиной непреднамеренной утечки конфиденциальных сведений.

Показано, что наилучшим из существующих решений, реализованных в современных защищенных операционных системах, является применение методики мандатного доступа, базирующегося на реализации схемы многоуровневой системы защиты. Рассмотрены основные возможности защищенной ОС Astra Linux по организации поддержки многоуровневого доступа, а также сделан обзор существующих вспомогательных программ и структур данных, обеспечивающих реализацию и настройку многоуровневой модели доступа в рамках защищенной ОС.

Собранный в статье материал позволяет обеспечить изучение темы «Организация многоуровневой защиты в защищенных ОС» в рамках курса «Защита операционных систем», а также может быть использован для настройки многоуровневого доступа в экземплярах защищенных ОС.

Литература

1. Хартиков, Д. М. Исследование и разработка методов реализации мандатных средств управления доступом в сетевых ОС семейства UNIX: дис. … канд. тех. наук / Д. М. Хартиков – М.: МЦСТ, 2004. – 194 с.

2. Безбогов, А. А. Методы и средства защиты компьютерной информации: учеб. пособие / А. А. Безбогов, А. В. Яковлев, В. Н. Шамкин. – Тамбов: Изд-во ТГТУ, 2006. – 120 с.

3. Нестеров, С. А. Информационная безопасность и защита информации: учебное пособие. /С. А. Нестеров. СПб.: Изд-во политехн. ун-та, 2009. – 126 с.

4. Барабанов, А. В. Семь безопасных информационных технологий: мандатный метод управления доступом / А. В. Барабанов, А. В. Дорофеев, А. С. Марков, В. Л. Цирлов / под ред. <А. С. Маркова>. – М.: Изд-во ДМК ПРЕСС, 2017. – 224 с.

5. Операционная система специального назначения «Astra Linux special edition». Руководство по КСЗ. в 2 ч. Ч. 1: 2012. – 100 с.

6. ГОСТ Р 34.11-2012 Информационная технология. Криптографическая защита информации. Функция хэширования .- М.: Стандартинформ, 2013. -24 с.

References

1. Hartikov, D. M. Issledovanie i razrabotka metodov realizacii man-datnyh sredstv upravlenija dostupom v setevyh OC semejstva UNIX: dis. … kand. teh. nauk / D. M. Hartikov – M.: MCST, 2004. – 194 s.

2. Bezbogov, A. A. Metody i sredstva zashhity komp'juternoj informacii: ucheb. posobie / A. A. Bezbogov, A. V. Jakovlev, V. N. Shamkin. – Tambov: Izd-vo TGTU, 2006. – 120 s.

3. Nesterov, S. A. Informacionnaja bezopasnost' i zashhita informacii: uchebnoe posobie. /S. A. Nesterov. SPb.: Izd-vo politehn. un-ta, 2009. – 126 s.

4. Barabanov, A. V. Sem' bezopasnyh informacionnyh tehnologij: man-datnyj metod upravlenija dostupom / A. V. Barabanov, A. V. Dorofeev, A. S. Markov, V. L. Cirlov / pod red. <A. S. Markova>. – M.: Izd-vo DMK PRESS, 2017. – 224 s.

5. Operacionnaja sistema special'nogo naznachenija «Astra Linux special edition». Rukovodstvo po KSZ. v 2 ch. Ch. 1: 2012. – 100 s.

6. GOST R 34.11-2012 Informacionnaja tehnologija. Kriptograficheskaja zashhita informacii. Funkcija hjeshirovanija. – M.: Standartinform, 2013. – 24 s.

ОКОРОКОВ Валерий Анатольевич, кандидат физико-математических наук, доцент кафедры защиты информации ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)». Россия, 454080, г. Челябинск, проспект В.И. Ленина, д. 76. E-mail: okorokovva@susu.ru

ЛАЩУК Дмитрий Евгеньевич, студент кафедры защиты информации ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)». Россия, 454080, г. Челябинск, проспект В.И. Ленина, д. 76. E-mail: darthus7@mail.ru

OKOROKOV Valeriy, candidate of physical and mathematical sciences, Associate Professor of Department of Information Security Department of «South Ural State University». Russia, 454080, Ural Federal District, Chelyabinsk Region, Chelyabinsk, prosp. V. I. Lenin, 76. E-mail: okorokovva@susu.ru

LASHCHUK Dmitry, student of the Information Security Department of «South Ural State University». Russia, 454080, Ural Federal District, Chelyabinsk Region, Chelyabinsk, prosp. V. I. Lenin, 76. E-mail: darthus7@mail.ru