Инж. С.А. Козлов

ПРОБЛЕМЫ БЕЗОПАСНОСТИ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ LINUX-ПОДОБНЫХ ОПЕРАЦИОННЫХ СИСТЕМ В ВАЖНЫХ ДЛЯ БЕЗОПАСНОСТИ АЭС СИСТЕМАХ

В общем случае операционные системы (ОС) Linux нельзя рассматривать как готовую к употреблению именно безопасную систему. Существует множество рекомендаций на тему повышения безопасности работы в Linux, однако авторы этих рекомендаций не гарантируют, что выполнение именно их рекомендаций достаточно для достижения требуемого уровня безопасности системы [1]. В данной работе предпринята попытка систематизации доступных методов повышения уровня безопасности прикладного ПО для Linux-подобных ОС, функционирующих в составе систем управления на объектах атомной энергетики. Необходимый уровень безопасности всей системы в целом может быть достигнут на уровне прикладного ПО. Предполагается, что прикладное ПО создается один раз и предназначено для решения четко определенного круга задач, который не расширяется в процессе эксплуатации ПО.

Проблемы обеспечения безопасности

Основной проблемой во всех моделях безопасности для систем управления является несанкционированный доступ (НД). Попытки НД можно пресекать (не создавать возможности их выполнения), а также предотвращать их успешное выполнение. Для этого необходимо:

- четко определить набор допустимых операций в системе и всех субъектов / объектов, которые имеют право на их исполнение. ПО системы должно само отслеживать возможные попытки НД, сигнализировать об их осуществлении и временно блокировать операции, использующиеся для их осуществления. Например если при входе в систему несколько раз вводится неправильный пароль,

то работа пользователя, под именем которого осуществляется попытка входа, - блокируется;

- сократить набор средств работы с системой до минимально возможного уровня. Например для операций архивирования должна использоваться только одна утилита. Другими словами, выполнение определенных действий в системе должно осуществляться только единственным способом, а не несколькими;

- наделить всех участников взаимодействия с системой набором прав, позволяющих каждому участнику выполнять в системе только разрешенные ему операции;

- исключить возможность использования собственных прав для получения новых, принадлежащих другим пользователям.

Для выполнения перечисленных требований необходимо разработать соответствующую политику безопасности (ПБ), дополняющую ПБ ОС Linux. Объектами политики безопасности в данном случае являются компьютеры, сетевое оборудование, операционные системы. Субъектами ПБ – люди, ПО и оборудование, взаимодействующие с объектами.

При рассмотрении различных концепций построения ПБ часто оперируют понятием “вредитель”. Вредитель – это субъект, имеющий права доступа к объектам или желающий получить их, случайно или преднамеренно использующий эти права в собственных целях, в данном случае для получения привилегий доступа. Вредитель может и не наносить непосредственного вреда системе (вносить помехи, мешающие корректному функционированию системы, изменять функциональное поведение системы, заложенное в нее разработчиками), но его возможности осуществлять подобную деятельность содержат в себе потенциальную угрозу. Далее предполагается, что вредителем может оказаться любой субъект. Подчеркивается, что вредителем может оказаться субъект, не замышляющий никаких вредоносных действий. Например обычному пользователю могут быть случайно назначены права суперпользователя, которыми он может случайно воспользоваться.

Одной из важнейших проблем обеспечения безопасности является человеческий фактор. Вредителем, прежде всего, может оказаться сотрудник обслуживающего персонала. Он может использовать идентификатор и ключ администратора системы.

Таким образом используются атрибуты пользователя с более высокими привилегиями прав доступа.

Классической, для слабо защищенных систем, является ситуация, когда пользователь с низкими привилегиями использует свои права доступа для их расширения. Например вредитель после завершения сеанса работы с системой оставляет средства (программные, аппаратные), позволяющие впоследствии получить доступ к конфиденциальной информации (ID, пароли). Это наиболее опасный вид атак, так как он наиболее тяжело отслеживается.

Угроза несанкционированного доступа может исходить не только от субъектов, но и от объектов. При работе в неправильно настроенной ОС или с некорректно написанным прикладным ПО может возникнуть ситуация, при которой субъекту по умолчанию назначаются права, которыми в рамках заданной политики безопасности субъект не должен обладать.

На начальных этапах разработки, проблема разрешения аварийных ситуаций должна рассматриваться отдельно и независимо от разработки ПБ. Но далее эти задачи тесно переплетаются друг с другом и их необходимо решать в совокупности. Дело в том, что мощная система защиты от НД может стать существенным препятствием на пути течения процесса ликвидации или предупреждения аварийных/критических ситуаций, особенно если действия производятся в режиме ручного управления. Поэтому, с одной стороны, при аварийной ситуации ПБ должна переключаться в режим пониженной безопасности, чтобы не препятствовать оперативным действиям обслуживающего персонала. Но, с другой стороны, в аварийном состоянии система будет находится в наиболее уязвимом состоянии, поэтому злоумышленник может создать или сфальсифицировать такую ситуацию и воспользоваться моментом, что может привести к катастрофическим последствиям.

Возможности удаленного доступа и удаленного администрирования следует использовать только в случае крайней необходимости, так как они являются наиболее уязвимыми для атак из внешней сети.

Методы повышения безопасности

Чем меньше в системе пользователей, тем меньше риск

возникновения опасных ситуаций, связанных с неправильным назначением прав. Исходя из чего, в первую очередь необходимо стремиться к снижению количества субъектов, возможности которых позволяют производить деструктивные действия по отношению к объектам (сетевые администраторы; обслуживающий персонал, обладающий соответствующими правами доступа к системе).

Необходимо исключить возможность субъектов/объектов действовать от имени других субъектов/объектов. Взаимодействие объектов по информационным каналам связи должно производиться с гарантиями подлинности как принимаемой, так и передаваемой информации. Возможна нумерация пакетов с последующим их шифрованием непосредственно перед передачей в сетевой адаптер. Принимающая сторона расшифровывает пакет и анализирует значение счетчика. Если это значение не равно фактическому (подсчитанному принимающей стороной), то произошел сбой при передаче пакета, или в рассматриваемом сегменте сети появилось второе устройство с адресом отправителя, что означает попытку вторжения. Некоторые современные коммутаторы автоматически выявляют появление несанкционированных пакетов (т.е. пакетов с незнакомыми сетевыми адресами).

Наиболее удачным, но вместе с тем трудно реализуемым, решением является разработка собственного ПО, обладающего всеми необходимыми для работы с системой инструментами (в рамках этой статьи такое ПО именуется Единой Средой Оператора). Доступом к работе с системой посредством консоли должны обладать единичные, избранные пользователи, а еще лучше, если работа с консолью вообще не будет требоваться в процессе эксплуатации системы. Предлагаемое ПО также должно содержать в себе все необходимые средства по администрированию системы[1].

Рекомендуемые нами этапы проектирования подобного ПО:

- построение полного списка элементарных операций, которые могут выполняться в системе по запросам обслуживающего персонала (операторов);

[1] Имеется в виду тривиальное назначение паролей пользователям (см. ниже).

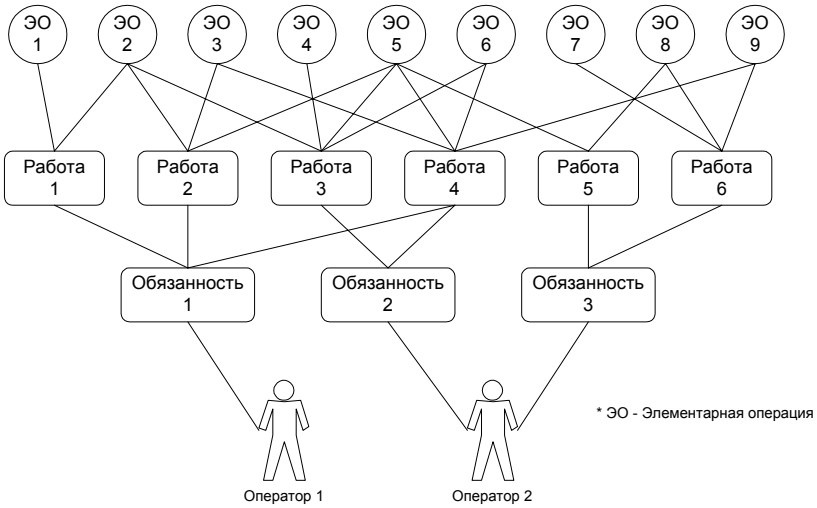
- определение полного перечня работ, которые могут выполняться при работе с системой. Под работой имеется в виду последовательное выполнение взаимосвязанных операций. Если какая-либо операция, входящая в состав необходимых для выполнения заданной работы, не может быть выполнена оператором, то данный оператор не может выполнять и работу;

- построение полного списка обязанностей обслуживающего персонала;

- группирование всех работ по отдельным обязанностям;

- назначение обязанностей отдельным операторам (физическим лицам).

Выполнение всех этапов приводит к получению иерархии, подобной приведенной на рисунке, в которой однозначно сопоставляются операции, работы, обязанности и непосредственные пользователи.



Распределение прав и обязанностей между пользователями

После построения подобной иерархии, на уровне прикладного ПО можно контролировать (разрешать / запрещать) выполнение отдельно взятым оператором всех операций и работ. Точнее, правильно разработанная система сама должна предоставлять оператору тот набор средств, на который у него есть права [2].

Предложенный подход замечателен тем, что такое дерево строится один раз на этапе проектирования ПО и больше никогда не модифицируется в процессе эксплуатации системы. А самое главное, что политика безопасности не требует постоянной поддержки, и системный администратор (СА) здесь отсутствует как класс. Необходимо наличие лишь человека, который изменяет пароли пользователей (назовем его администратором паролей (АП)), но он не может разрушить систему, как в случае использования стандартной ПБ Linux, в которой СА может делать буквально все (копировать, удалять, редактировать, форматировать).

Тем не менее, наиболее уязвимым местом в предлагаемом подходе является наличие АП. Ему никто не мешает назначить пароль любому пользователю и войти в систему от его лица. Для предотвращения подобных операций можно использовать системы идентификации пользователя по его биометрическим данным, например отпечаткам пальцев. Теперь каждому идентификатору пользователя должно соответствовать два подтверждения – пароль и отпечаток пальца. Вход в систему осуществляется в три этапа: ввод “логина”, ввод пароля, подтверждение и того, и другого – отпечаток пальца. Система должна хранить в памяти отпечатки пальцев всех людей, появление которых допускается на объекте. Основное преимущество этого подхода заключается в том, что ни один пользователь не сможет действовать от чужого лица.

Только использование прикладного ПО может дать гарантию безопасности системы. Если такое ПО создать невозможно, то каждому пользователю в системе должно быть разрешено использование только необходимых для его работы программных средств. Такой подход можно реализовать, но его недостатки уже перечислялись в начале этого раздела.

Ограничение и контроль доступа к объектам

Непосредственные попытки доступа к объектам должны предотвращаться на стадии обработки сетевых пакетов. В сегментах Ethernet должны передаваться только пакеты с заранее известными MAC адресами. Появление в сегменте не "родного" для него Ethernet-пакета означает подключение к сегменту дополнительного устройства, на что система безопасности должна выдать предупредительный сигнал. Для отслеживания “чужих”

пакетов могут использоваться специальные программы, осуществляющие контроль всего трафика в выделенном сегменте. Для предотвращения распространения “чужих” пакетов в другие сегменты можно использовать возможности коммутаторов, настраивая соответствующим образом их внутренние таблицы межпортовой переадресации и возможности маршрутизаторов, задавая им жесткие таблицы маршрутизации.

Предпосылки создания единой среды оператора

Проблемы построения простой в использовании и защищенной от НД многопользовательской информационной системы достаточно сложны. Связано это с тем, что на начальных этапах проектирования ПО весьма непросто определиться с несколькими вопросами:

- на кого рассчитывается данное ПО, т.е. каков круг конечных потребителей;
- насколько полно система должна поддерживать многопользовательский режим;
- откуда может исходить угроза НД.

ОС Linux создана таким образом, что большинство разработчиков и пользователей, опирающихся в своих проектах на использование этой ОС, могут решать любые свои задачи опираясь на стандартную ПБ [3]. Другими словами - система универсальна, однако существует множество пакетов, выполняющих широчайший круг задач и требующих для своего функционирования специфических условий. Сама ОС базируется на прошедшей испытание временем политике безопасности. Единственное, что смущает в использовании этой системы как ОС общего применения, это тот факт, что для работы с ОС Linux, посредством только входящих в дистрибутив стандартных средств, требуется высокий профессиональный уровень пользователей, что в большинстве случаев гарантировать невозможно.

Любая универсальность несет в себе опасность, связанную с неправильной конфигурацией системы, так как при большом разнообразии тяжело учесть все факторы. С другой стороны, чем система проще, и чем меньше используется программ, написанных сторонними производителями, тем меньше вероятность появления не учтенных на этапах разработки нештатных ситуаций и сбоев, возникающих из-за ошибок в ПО.

Политика безопасности Linux, требует постоянной поддержки со стороны системного администратора. Администратор обладает неограниченными правами в системе и, как и любой человек, может совершить ошибку. Естественно, что присутствие такого человека (точнее человека с такими возможностями) на объекте нежелательно, а, с другой стороны, он необходим, если использовать только ПБ ОС Linux. Проблема наличия администратора дополняется уже высказанным мнением о наличии на объекте пользователей-профессионалов.

Для решения этих задач могут быть использованы следующие рекомендации:

1. Количество сотрудников постоянно и сами сотрудники редко меняются. Поэтому не приходится создавать в системе новые учетные записи, изменять существующие и переназначать права доступа к каталогам, файлам и т. д., что позволяет исключить использование стандартных средств настройки политики безопасности ОС Linux и обойтись без высококвалифицированного администратора.
2. Обязанности всех сотрудников на объекте строго ограничены и не могут изменяться во времени, т.е. каждому сотруднику сопоставляется уникальный для него набор средств, которыми может пользоваться только он. Даже если сотрудник сменится, то понадобится тот же самый неизменный набор прав.
3. Система не должна поддерживать многопользовательский режим, т.е. в отдельно взятый момент времени с системой может работать только один человек. Никакой поддержки удаленной работы с системой нельзя допускать.
4. Для непосредственной работы с системой, каждый сотрудник обслуживающего персонала может пользоваться только ПО, созданным разработчиками системы, и сотрудникам не нужны стандартные средства Linux.

Функционально объединить все вышесказанное можно в Единой Среде Оператора (ЕСО). Кратко, суть ЕСО заключается в том, что выполнение всех возможных операций в системе максимально упрощается и автоматизируется посредством использования ЕСО. Никакой пользователь системы не может выйти за рамки ЕСО, т.е. применение командной строки или стандартных утилит администрирования не допускается. Напрямую все эти возможности пользователям недоступны.

Посредником между пользователем и всеми утилитами служит ЕСО, которая позволяет выполнять операции только в том объеме и только тем пользователям, которым это разрешается внутренней политикой безопасности ЕСО. Каждый пользователь системы работает только в специально предназначенной для него оболочке и пользуется только теми средствами, которые ему эта оболочка предоставляет. Любые другие средства данному пользователю недоступны. Оболочка состоит из окон, каждое из которых предоставляет доступ к различным инструментам, т.е. работая с конкретным окном, можно выполнять в системе только строго определенный набор действий. Работа всех пользователей в системе носит сеансовый характер. Пользователь подключается к системе (открывает сеанс), производит некоторые операции и отключается от системы (закрывает сеанс). Вход в систему другого пользователя возможен только в случае, если предыдущий пользователь закрыл сеанс, т.е. в каждый момент времени с системой может работать только один пользователь. Перед открытием сеанса доступно только одно окно, в которое оператор вводит свои данные, удостоверяющие его личность (происходит процесс аутентификации). По окончании ввода система (оболочка) переводится в соответствующий режим. Под режимом имеется в виду самонастройка на род работ, выполняемых подключающимся оператором.

Варианты самонастройки оболочки могут быть следующими:

- каждому пользователю соответствует четко заданный список прав, в котором перечисляются все окна, с которыми он может работать. Система не отображает окна, с которыми данному пользователю работать нельзя;
- для каждого пользователя заранее создается конкретная оболочка как отдельная программа, которая подгружается после открытия сеанса, или в единой программе отдельная ее часть отвечает за работу с конкретным пользователем.

Рекомендации по проектированию интерфейса пользователя:

1. Окон должно быть как можно меньше. Должна обеспечиваться возможность работы с окнами без использования манипулятора типа мышь, трекбол и т. п.

2. Элементы управления должны группироваться в окне по их функциональному назначению. Нужно максимально эффективно использовать свободное место на поверхности

окна для размещения на нем элементов управления.

3. Конструкция интерфейса должна способствовать предотвращению ошибочных действий оператора.

4. Все элементы интерфейса пользователя должны соответствовать одному стилю. Если используются квадратные кнопки, то они должны использоваться везде. Если кнопка “закрыть окно” располагается справа внизу окна, то она должна находиться на этом же месте и на всех окнах.

5. Избегать “выделение цветом”. В качестве элемента маркировки лучше использовать значки типа “восклицательный знак”, “вопросительный знак” и т. д.

6. Цветовая палитра должна быть максимально сокращена и иметь строгую стилистическую направленность.

7. Все ПО должно разрабатываться под строго определенное разрешение экрана. Если выбран режим 1024x768, то его необходимо использовать во всех программах.

8. Активация любого элемента управления должна всегда приводить к однозначным визуальным изменениям, ожидаемым оператором.

9. Состояние всех отображаемых элементов управления должно соответствовать текущему состоянию объектов управления.

10. Оператору всегда должно выдаваться подтверждение того, что система приняла его запрос, и о том, что запрос выполнен корректно. Если запрос выполняется достаточно быстро, желательно на время его выполнения запретить пользовательский ввод. Таким образом, оператор поймет, что система обрабатывает его действия.

11. Все сообщения об ошибках должны выводиться в краткой форме (1-2 строки текста), но должна существовать возможность получить исчерпывающую информацию об ошибке. Например можно в предупреждающем окне предусмотреть кнопку “подробнее”, после нажатия на которую краткое сообщение замещается подробным разъяснением.

12. Избегать запутанных справочных систем. Справочная система должна быть как можно более простой.

13. Пока не завершится работа с одним окном, все остальные окна должны быть недоступны. Можно на всех окнах разместить кнопку "Done" ("Close", "Ok"...), нажатие которой говорит о том, что оператор закончил работу с окном, и в зависимости от

контекста нужно вернуться к предыдущему окну или открыть следующее.

14. Избегать повсеместного использования элементов управления с реакцией "Apply" ("Применить"). Подтверждение ввода должно присутствовать на всех окнах или отсутствовать вообще.

15. Избавить оператора от возможности перетаскивать окна. При активации каждое окно должно появляться строго в определенном месте и иметь заранее определенный, постоянный размер.

16. Всегда должна быть доступна кнопка "Freeze", после нажатия на которую любой ввод блокируется пока не будет введен пароль пользователя, нажавшего на кнопку. Блокировка ввода должна блокироваться только на некоторое время, по истечении которого сеанс работы с системой автоматически завершается.

17. Все действия оператора должны фиксироваться в журнале, и все пользователи системы должны быть об этом открыто предупреждены. Ответственность за пагубные действия несет непосредственно тот человек, под именем которого был открыт сеанс.

Рекомендации по настройке графического интерфейса ОС Linux (X-Server)

В конечной версии ЕСО, которая будет устанавливаться непосредственно на объекте, в первую очередь должна быть отключена поддержка виртуальных консолей, переключение между которыми осуществляется нажатием комбинаций клавиш Alt+Fn (или Ctrl+Alt+Fn, если переключение производится в контексте X-приложения) [4]. Точнее, у пользователя должна отсутствовать возможность переключаться между консолями. На этапе разработки ПО поддержку виртуальных консолей отключать необязательно. Поддержка отключается посредством редактирования файла /etc/inittab (строки типа: c1:12345:respawn:/sbin/getty tty1 38400 linux) или построением ядра с сокращенным набором консолей. Проблемы могут возникнуть при запуске X-сервера, которому необходима как минимум одна свободная консоль для запуска. Решить проблемы можно запрещая любой пользовательский ввод во время загрузки X-сервера, а затем, после его инициализации и загрузки ЕСО, разрешая только ту

консоль, которую при инициализации занимает X-сервер. Полезными могут оказаться утилиты Xlock и vlock.

Все стандартные X-серверы, как правило, допускают останов или перезагрузку по нажатию на клавиши Ctrl+Alt+Backspace. Эту возможность также необходимо отключить путем соответствующей настройки X-сервера. Возможность в ходе работы переключаться между различными разрешениями экрана (нажатие комбинаций Ctrl+Alt+N) должна быть отключена, делается это посредством редактирования файла XF86Config. В секции Screen этого файла нужно оставить только одну настройку экрана.

Дополнительным эффективным решением является написание собственного драйвера клавиатуры, пропускающего только нажатия разрешенных комбинаций клавиш.

Заключение

Предложенные в работе меры по обеспечению политики безопасности прикладных программных модулей носят рекомендательный характер. Однако они могут быть использованы не только для обеспечения безопасности ОС Linux, но и любой Unix-подобной операционной системы с графическим или текстовым интерфейсом. Предложенные в работе рекомендации были использованы при разработке дистрибутива ОС Linux, получившего название LiNEM, и предназначенного для работы в составе программно-технического комплекса информационно-диагностической сети системы управления и защиты реакторов типа ВВЭР-1000 АЭС.

ЛИТЕРАТУРА

1. Larry Greenfield. Руководство пользователя Linux // www.linuxdoc.ru. 1994.

2. Операционная система Unix (руководство пользователя) // www.null.ru. 1998.

3. Lars Wirzenius. Руководство системного администратора ОС Linux // www.citforum.ru. 2001.

4. David Wheeler. Secure Programming for Linux and Unix // Linux.com. 2002.