БЕЗОПАСНОЕ ВЫПОЛНЕНИЕ СТОРОННЕГО ПРОГРАММНОГО КОДА

Головлев А.А.

Головлев Андрей Алексеевич – студент,
электроэнергетический факультет,
Вологодский государственный университет, г. Вологда

Аннотация: *оценка эффективности системы защиты информации (СЗИ) от угроз кибернетического характера требует применения той или иной строгой формальной постановки понятия «кибернетическая безопасность», которая, в свою очередь, должна быть органично вплетена в немалый существующий понятийный базис теории и практики обеспечения безопасности информации в автоматизированных системах (АС), максимально приближенным к практическим особенностям реализации угроз, так сказать, кибернетической природы, присущих конкретным АС. Основой существования подобных систем является выполнение программных кодовых последовательностей. Безопасность на уровне программного обеспечения должна быть реализована по умолчанию. Проблемным остается реализация структуры гарантированности стороннего кода самим пользователем.*

Ключевые слова: *программный код, безопасное выполнение, система безопасности, кибернетическая природа, пользователь системы.*

Согласно [4] безопасность информации – состояние информации, в котором обеспечивается сохранение определенных политикой безопасности свойств информации. Среди прочих, одним из основных принципов построения комплексов средств защиты СЗИ является реализация, в том или ином виде, концепции диспетчера доступа, сущность которой заключается в обеспечении контроля всех запросов субъектов (единиц вычислительной работы) по использованию объектов (файлов, каталогов, объектов памяти, устройств и др.) с одновременным принятием решения о допустимости или запрете такого обращения согласно правил разграничения доступа (ПРД) как составной части политики безопасности АС. Разграничение доступа в современных АС базируется на применении субъектно-объектной модели. Она предоставляет развитый формальный аппарат для доказывания существенных свойств базовых, идеализированных политик безопасности, как дискреционной (от англ. – Discretionary Access Control, DAC), так и мандатной (от англ. – Mandatory Access Control, MAC) и, как следствие, позволяет применять аксиому: безопасность системы может быть вполне обеспечена путем управления доступом субъектов к объектам на протяжении всего жизненного цикла АС [5]. При этом практическая реализация того или иного варианта политики безопасности обусловила необходимость внедрения органического сочетания прав субъекта доступа, как единицы вычислительной работы порожденного в контексте сеанса работы пользователя с определенными полномочиями (ролью) по использованию объектов АС. Ролевая политика безопасности (от англ. – Role Base Access Control, RBAC) стала в некотором смысле компромиссом между дискреционной и мандатной модели, а также основным подходом к организации разграничения доступа в современных операционных системах (ОС) общего назначения благодаря своей практической наглядности и понятности. Так, успешная авторизация средствами ОС пользователя как физического лица, приводит к открытию соответствующего сеанса работы и порождение в его контексте первичного вычислительного процесса (лидер сеанса) с полномочиями доступа к объектам согласно роли пользователя в АС. Запуск на выполнение в рамках сеанса работы пользователя других программ приводит к формированию в том или ином виде иерархического дерева родственных отношений между создаваемыми вычислительными процессами с подражанием полномочий лидера сеанса, который выступает корнем деревья. Исходя из изложенного, все вычислительные процессы или субъекты, инициированные пользователем, имеют одинаковые, совпадая с лидером сеанса, полномочия по доступу к объектам АС в пределах предусмотренного для их полномочий информационного домена. Считается, что у непривилегированного, обычного пользователя не имеет возможности расширить (изменить) свои полномочия, тем самым создать необходимые условия для выхода за пределы своего информационного домена и нарушения ПРД.

Современные ОС общего назначения, реализуя концепцию принудительной многозадачности на основе разделения процессорного времени, предусматривают одновременную, так сказать, жизнь значительного конечного множества вычислительных процессов. Некоторые из них выполняются в контексте сеансов работы обычных, физических пользователей, но в основном большая часть – это так называемые служебные вычислительные процессы (демоны) запущенные в ходе инициализации АС от имени так сказать физически несуществующих пользователей в качестве удобной абстракции первичной передачи полномочий (псевдопользователь). Другими словами, в АС построенной с применением ОС общего назначения практически всегда, даже когда осуществляется единичный сеанс работы физического пользователя, есть определенное количество вычислительных процессов инициированных

от имени того или иного псевдопользователя с присущим ему информационным доменом, согласно выполняемых служебных функций, как правило, с применением повышенных полномочий. Соответственно принуждение служебного вычислительного процесса к выполнению непредвиденной работы в ходе сеанса работы физического пользователя с гораздо большей вероятностью может привести к определенному нарушению безопасности информации АС, не нарушая ПРД.

Принуждение любого вычислительного процесса к выполнению непредвиденной его разработчиками работы базируется на применении методов, способов, приемов эксплуатации потенциально-опасных дефектов проектирования и программной реализации. Существует немало примеров непредусмотренного разработчиками внешнего влияния на программы во время выполнения, последствиями которого, как правило, являются: аварийное завершение работы, навязывание выполнения стороннего кода, расширение полномочий атакующего, непредвиденная модификация данных и утечка информации. Наиболее репрезентативными классификациями известных потенциально опасных дефектов программ, условий, способа их эксплуатации, а также возможные последствия в форме структурированных, рассчитанных на специалистов шаблонов считаются проекты Common Vulnerability Enumeration, Common Weakness Enumeration, Common Attack Pattern Enumeration and classification.

Известно, что главная цель политики безопасности АС заключается в разработке и внедрении такой организации обработки информации, при которой будет минимизирована величина потенциальных убытков от возможных угроз, опираясь на функционирование диспетчера доступа относительно строгого соблюдения ПРД в ходе эксплуатации АС. При этом существующие положения нормативных документов технической защиты информации в компьютерных системах от несанкционированного доступа лишь касаются или рассматривают поверхностно принцип корректной работы программного обеспечения АС, считая его как само собой разумеющееся, нормальное предположение.

Последнему противоречит мировой опыт расследования инцидентов информационной безопасности в компьютеризированных системах: значительное количество нарушений политики безопасности информации является результатом целенаправленной злонамеренной деятельности непривилегированных авторизованных пользователей на некотором этапе эксплуатации АС с применением штатных, специальных программных средств непредусмотренного разработчиками влияния на алгоритм работы служебных вычислительных процессов на основе эксплуатации потенциально-опасных дефектов их проектирования и программной реализации. Заставляя, таким образом, служебный вычислительный процесс до выполнения непредусмотренной разработчиками работы от обычного авторизованного пользователь-нарушитель может по своей инициативе инициировать внесение изменений в его информационный домен, нарушая тем самым прописанную в политике безопасности технологию обработке информации на этапе эксплуатации АС, но не вступая при этом в конфликт с диспетчером доступа. Данное явление будем рассматривать как бесконфликтный несанкционированный доступ непривилегированного, авторизованного пользователя-нарушителя и основную причину нарушения политики безопасности информации для физически отделенных от сетевой среды передачи АС построенных с применением современных ОС общего назначения.

Таким образом, можно констатировать, что политика безопасности информации не может быть полной, если она базируется только на строгой реализации ПРД диспетчером доступа и без должного внимания к корректной работе системного программного обеспечения даже в выделенных АС. Благодаря наличию потенциально-опасных дефектов проектирования и реализации служебных программ, утилит (примерно до 5-ти на 1000 строк исходного текста) всегда существует достаточно большая вероятность бесконфликтного несанкционированного доступа с нарушением политики безопасности, но не ПРД, и, как следствие, уменьшение эффективности функционирования АС в целом путем непредвиденного воздействия на нормальную работу ее кибернетической составляющей (вычислительная среда, процессы). Фактически мы видим требование практики относительно необходимости интенсивного исследования ряда актуальных взаимосвязанных научных задач исходя из реалий информационной борьбы, мирового опыта расследования инцидентов информационной безопасности, что, в свою очередь, позволил выявить определенные кризисные явления теории защиты информации.

Прежде всего, необходимо разработать:

- содержательную и формальную постановку понятие «кибернетическая безопасность»;
- модель оценки значение показателя кибернетической безопасности для выбранной АС.

Исходя из изложенного, под кибернетической безопасностью АС будем понимать состояние АС, в котором исключена (минимизирована) возможность выполнения непредсказуемой политикой безопасности информации вычислительной работы. При построении регулярного выражения показателя кибернетической безопасности будем применять несколько существенных предположений:

1. Функционирование любой АС представляет собой последовательность переходов одного состояния в другой в случайные моменты времени и может рассматриваться как случайный процесс.

2. Семантика состояний АС и условия переходов между состояниями определяются исходя из их значимости для описания участия пользователей, как физических лиц, принятой в организации технологии обработки информации и администрирования (проведение регламентных работ), а также с учетом практической реализации принципов ролевой политики безопасности в современных ОС общего назначения.

3. Совокупность сеансов работы пользователей АС, запуск на выполнение в их пределах вычислительных процессов, а также их свойства, рассматриваются как потоки заявок на обслуживание до исполняющего устройства в терминах теории массового обслуживания. Функция исполняющего устройства реализуется ОС как операционная среда пользователя.

4. Любой от обычного авторизованный пользователь АС в любой момент времени может начать целенаправленную злонамеренную деятельность нарушителя, выполняя соответствующую последовательность действий штатными программными средствами или программную реализацию специального алгоритма расширения полномочий, что само по себе является нарушением политики безопасности АС, учитывая возможные последствия. Успех попытки расширения полномочий зависит от предоставленного для этого интервала времени. По результатам неуспешных попыток нарушитель совершенствует свои действия, программные средства, приближаясь к желаемому результату. Успешной может являться последняя попытка. В данной постановке только логин от обычного пользователя АС может быть внезапно начать следовать целям нарушителя.

5. Будем считать, что после завершения сеанса работы непривилегированного пользователя завершаются все вычислительные процессы, которые запущены на исполнение в его контексте, а АС переходит в состояние, в котором наличие или отсутствие кибернетической безопасности является неопределенной.

6. Администратор – привилегированный пользователь, который среди прочих задач при проведении регламента в пределах своего сеанса обеспечивает выполнение всей полноты мер восстановления, при необходимости, кибернетической безопасности АС. Администратор имеет все полномочия. В данной постановке администратор не может быть нарушителем.

7. Переход между состояниями АС является результатом наступления соответствующих событий в терминах теории цепей Маркова: начало, завершение сеанса работы пользователя или администратора.

Список литературы

1. *Алексеев М.О.* Об обнаружении алгебраических манипуляций с помощью операции умножения // Информационно-управляющие системы? 2014. № 3 (70). С. 103-108.
2. *Карандеев Д.Ю., Голубничий А.А*. Реализация метода ветвей и границ в статистической среде R // Интернет-журнал Науковедение? 2015. Т. 7. № 6 (31). С. 109.
3. *Кульман Н.Ю., Кульман Т.Н., Кочеткова А.В., Шлейхер В.С*. Отображение данных в окнах сторонних приложений // В сборнике: Ситуационные центры и информационно-аналитические системы класса 4i для задач мониторинга и безопасности (SCVRT2015-16). Труды Международной научной конференции: в 2-х томах, 2016. С. 164-167.
4. *Фадеев С.Г.* К вопросу о внесении изменений в сторонний исходный код // Вестник научных конференций, 2015. № 4-4 (4). С. 149-150.
5. *Яремчук С.* Безопасен ли открытый код? // Системный администратор, 2012. № 6 (115). С. 8-11.