**AppArmor –песочница для приложений**

На заре развития электронной вычислительной техники о безопасности мало кто задумывался. Однопользовательский режим и не слишком сложные по современным меркам операционные системы позволяли владельцу компьютера очень хорошо понимать все процессы, происходящие в системе.

Затем компьютеры стали более распространенными и возникли проблемы, связанные с безопасностью. Тогда на смену концепции «разрешено все» пришло новое правило: «разрешено все, что не запрещено». Какой из двух подходов хуже? Разрешая все, операционная система была беззащитна перед пользователем. Однако второй подход привел к новым проблемам и появлению стремительно развивающихся бизнес-направлений: подпольного бизнеса создателей вирусов и официального – разработчиков антивирусов, межсетевых экранов и других программ обеспечения безопасности. Современные ботнеты, состоящие из миллионов компьютеров «зомби», прикрыты иллюзией безопасности: «разрешено все, что не запрещено»!

В серии статей мы рассмотрим систему AppArmor, реализующую другой – менее популярный, но более мощный – принцип: «запрещено все, что не разрешено явно». Быть может, его повсеместная реализация позволит снизить накал гонки «эксплойтов» и обновлений безопасности? В первом материале серии мы дадим только основные сведения об этой замечательной разработке. В последующих – остановимся на работе с AppArmor подробнее, а также рассмотрим другие средства обеспечения безопасности, такие как SELinux

**Уязвимое приложение – уязвимая система?**

С какими правами доступа работают программы, запущенные на вашем компьютере? Почтовый клиент и браузер имеют доступ к переписке в jabber-клиенте, jabber-клиенту не запрещается обращаться к вашей почте и файлам профиля браузера... А так ли необходимы программам столь большие возможности? Для комфортной работы вполне достаточно, чтобы jabber-клиент имел доступ только к каталогу с историей сообщений и своими настройками. Почтовому клиенту для работы хватит собственного каталога с настройками, сообщениями, плюс дополнительного каталога, из которого он будет брать и в который будет сохранять файлы, прикрепляемые к письмам. Интернет-браузеру достаточно собственного профиля и каталога обмена файлами.

Свободный доступ ко всем файлам пользователя – это очень плохая практика с точки зрения безопасности. Однажды в библиотеке обработки графических файлов найдут критическую уязвимость, кто-то создаст и опубликует пример ее реализации, и вот уже ваш недруг на досуге копается в вашем домашнем каталоге, получив к нему доступ через графический редактор! Или ваш торрент-клиент, взломанный злобным хакером, рассылает по всему миру спам и управляет огромным ботнетом (хотя для выполнения своих функций ему не требуется доступа к утилите mail). Кому-то это может показаться невероятным и граничащим с паранойей, но такой сценарий вполне возможен. Все это следствие того, что запущенная от вашего имени программа обладает точно такими же правами, как и вы сами. Как же от этого защититься?

**Бронирование приложений или Безопасная «песочница»**

Одним из наиболее популярных низкоуровневых приложений, реализующих защиту системы на основе принципа «запрещено все, что не разрешено явно», является разработка компании Novell – система AppArmor. Ее основное назначение – защита системы от скомпрометированных (т.е. взломанных) программ. Основана такая защита на предоставлении программе минимально необходимых для работы возможностей и привилегий. Это значит, что ваш торрент-клиент не будет иметь прав на запуск программы mail, а jabber-клиент – на доступ к каталогу электронной почты. Другими словами, AppArmor помещает приложение в своеобразную «песочницу», ограниченную точно заданными возможностями, и в случае взлома весь ущерб, нанесенный хакером через скомпрометированную программу, ограничится лишь содержимым этой самой «песочницы».

Для обеспечения защиты системному администратору необходимо лишь определить каждой программе свой профиль безопасности, ограничивающий доступ к каталогам, другим программам и POSIX-возможностям. В свою очередь, AppArmor берет на себя задачу контроля приложения и оповещения при попытке программы воспользоваться неразрешенными ресурсами. Таким нехитрым образом AppArmor на основе традиционного дискреционного контроля доступа Unix предоставляет новую модель – мандатный контроль доступа того, что запущенная от вашего имени программа обладает точно такими же правами, как и вы сами. Как же от этого защититься?

Для обеспечения защиты системному администратору необходимо лишь определить каждой программе свой профиль безопасности, ограничивающий доступ к каталогам, другим программам и POSIX-возможностям. В свою очередь, AppArmor берет на себя задачу контроля приложения и оповещения при попытке программы воспользоваться неразрешенными ресурсами. Таким нехитрым образом AppArmor на основе традиционного дискреционного контроля доступа Unix предоставляет новую модель – мандатный контроль доступа.

**Проверяем статус AppArmor**

Дискреционный и мандатный контроль доступа

ДКД реализован в большинстве операционных систем. Заключается он в следующем: каждому объекту (файлу, каталогу и т.д.) приписывается владелец, который может произвольно изменять права доступа на принадлежащий ему объект.

Сила ДКД в простоте – это ключевая причина, по которой он широко известен и реализован в наиболее распространенных ОС.

Основные недостатки ДКД следующие:

* ограничение глобальной политики: ДКД позволяет пользователю определять доступ к своим данным вне зависимости от глобальных политик. Если ДКД является глобальной политикой, то существуют сложности с гарантированием согласованности правил.
* отсутствие контроля информационного потока: информация может быть скопирована из одного объекта в другой, тогда режим контроля доступа к копии не зависит от режима контроля доступа к оригинальному объекту.
* уязвимость перед скомпрометированным ПО: политики ДКД могут быть легко изменены владельцем, поэтому хакер, получивший контроль над программой, запущенной владельцем объекта, может изменить политику ДКД без ведома самого владельца.

Работа AppArmor основана на профилях – каждому приложению, работающему под контролем AppArmor, назначается профиль, который указывает, какие права и возможности доступны приложению. Для упрощения развертывания и настройки в пакет включается набор стандартных профилей для наиболее популярных (и требующих защиты)приложений. Также готовые профили можно найти для многих распространенных (в основном, серверных) программ – ищите ссылку на сайт в разделе «Ссылки». Таким образом, все, что необходимо системному администратору, решившему воспользоваться AppArmor, – это правильный выбор приложений, нуждающихся в ограничении привилегий, и создание/редактирование профилей безопасности.

Далее мы подробно рассмотрим работу с системой AppArmor, настройку и установку стандартных профилей и проверим работоспособность.

* Для пользователей OpenSUSE в Yast2 имеется графический интерфейс к AppArmor, однако он не очень удобен. К тому же в других дистрибутивах такого интерфейса нет. Поэтому далее мы будем ориентироваться на работу в командной строке.
* Все указанные ниже команды проверялись в openSUSE 11.1 и Kubuntu 8.04.Пользователям других дистрибутивов следует проверить наличие портированной версии AppArmor и, при необходимости, воспользоваться соответствующими инструкциями поустановке.

Чтобы уменьшить дистрибутивные отличия, выполним простенькую команду:

*root: ln -s /etc/init.d/apparmor /sbin/rcapparmor*

*root: rcapparmorUsage: /sbin/rcapparmor {start|stop|restart|try-restart|reload|force-reload|status|kill}*

Как вы уже догадались, эта команда управляет демоном AppArmor.

Прежде всего проверим, запущен ли AppArmor на компьютере.

Kubuntu выдал нам следующий результат:

root: rcapparmor statusapparmor module is loaded.2 profiles are loaded.2 profiles are in enforce mode. /usr/sbin/cupsd /usr/lib/cups/backend/cups-pdf0 profiles are in complain mode.1 processes have profiles defined.1 processes are in enforce mode : /usr/sbin/cupsd (4091)0 processes are in complain mode.0 processes are unconfined but have a profile defined.

Вывод той же самой команды в openSUSE:

*root: rcapparmor statusapparmor module is loaded.11 profiles are loaded.10 profiles are in enforce mode. /usr/sbin/ntpd /usr/sbin/identd /sbin/klogd /sbin/syslogd /sbin/syslog-ng /usr/sbin/traceroute /usr/sbin/nscd /bin/ping /usr/sbin/mdnsd /usr/sbin/avahi-daemon1 profiles are in complain mode. /bin/arch5 processes have profiles defined.0 processes are in enforce mode :0 processes are in complain mode.5 processes are unconfined but have a profile defined. /sbin/klogd (1934) /usr/sbin/nscd (3742) /sbin/syslog-ng (1937) /usr/sbin/avahi-daemon (3450) /usr/sbin/ntpd (3773)*

**Мандатный контроль доступа**

МКД определяет глобальную политику безопасности, субъектами которой являются в том числе пользователи. Владелец не может установить права доступа к объекту более слабые, чем определено системной политикой. Системные политики указывают, кому разрешен доступ. Обычные пользователи не могут изменить политику.

Основной недостаток МКД – сложность создания и сопровождения системной политики.

Как видите, по умолчанию OpenSUSE контролирует больше программ, чем Kubuntu. Разберем подробнее, о чем рапортует нам команда rcapparmor status в OpenSUSE (как представившая больше информации). Итак, модуль apparmor загружен и готов к использованию, всего установлено 11 профилей. Существуют два возможных режима контроля AppArmor применительно к каждому приложению: enforce и complain. В режиме enforce система AppArmor будет ограничивать работу приложения согласно профилю и сообщать о попытках нарушения установленных правил, а в режиме complain – только сообщать о нарушениях правил, позволяя приложению выполнять любые запрашиваемые действия. Нетрудно догадаться, что второй режим используется, как правило, для отладки профилей (чтобы отсутствие какого-то конкретного разрешения не нарушило нормальной работы программы), а первый режим собственно и защищает систему от нештатной работы приложения. В нашем примере в OpenSUSE в режиме enforce работает 10 приложений, а в режиме complain – только одно. О переключении между этими режимами будет рассказано ниже.

Важной особенностью AppArmor является то, что сопоставление профиля приложению осуществляется согласно абсолютному пути к запускаемому файлу. Если вы скопируете, например, утилиту ping из каталога /bin/ в каталог /usr/bin или переименуете ее в my\_ping,то никакие ограничения AppArmor на нее действовать уже не будут – не будет профиля для «новой» программы. При некоторых (очень редких!) условиях эта особенность может быть использована хакером, зато она же дает широкие возможности (о которых будет рассказано в следующей статье).

Как unconfined помечены процессы, для которых профили существуют, но ограничения AppArmor не распространяются (в нашем примере пять таких программ). Это бывает, когда соответствующие процессы были запущены до загрузки AppArmor. Бороться с этим просто: перезапустите нужное приложение и оно перейдет в список enforce (или complain):

*root: kill -9 3773; /usr/sbin/ntpdroot: apparmor\_status...5 processes have profiles defined.1 processes are in enforce mode : /usr/sbin/ntpd (6707)0 processes are in complain mode.4 processes are unconfined but have a profile defined. /sbin/klogd (1934) /usr/sbin/nscd (3742) /sbin/syslog-ng (1937) /usr/sbin/avahi-daemon (3450)*

Число загруженных профилей можно увеличить – за это отвечает пакет apparmor-profiles. ВOpenSUSE он уже установлен, а в Kubuntu придется сделать это вручную:

*user: sudo apt-get install apparmor-profiles*

Теперь проверим что изменилось:

*user: sudo apparmor\_status15 profiles are loaded.*

3 profiles are in enforce mode. */usr/lib/cups/backend/cups-pdf /usr/sbin/cupsd /usr/sbin/avahi-daemon12 profiles are in complain mode. /usr/sbin/ntpd /usr/sbin/identd /usr/sbin/nmbd /usr/sbin/dnsmasq /sbin/klogd /sbin/syslogd /usr/sbin/smbd /sbin/syslog-ng /usr/sbin/traceroute /usr/sbin/nscd /usr/sbin/mdnsd /bin/ping5 processes have profiles defined.1 processes are in enforce mode : /usr/sbin/cupsd (4091)0 processes are in complain mode.4 processes are unconfined but have a profile defined. /usr/sbin/avahi-daemon (4047) /usr/sbin/avahi-daemon (4046) /sbin/syslogd (3954) /sbin/klogd (4001)*

**Добавление профилей. Документация**

Для получения более подробной технической информации читайте man apparmor.Чтобы узнать все детали синтаксиса файлов профилей, читайте man apparmor.d.Техническая документация на AppArmor в OpenSUSE содержится в файле /usr/share/doc/packages/apparmor-docs/techdoc.pdf, устанавливаемом вместе спакетом apparmor-docs.

Загляните в каталог /usr/share/doc/apparmor-profiles/extras в Kubuntu или /etc/apparmor/profiles/extras в OpenSUSE. Здесь вы найдете большое число дополнительных профилейкак серверных процессов, так и «десктопных» (включая всеми любимую команду man).В этом же каталоге можно найти, в том числе, и профиль для Firefox и Opera – под контролем AppArmor ваш браузер станет еще более безопасным! В имени каждого из профилей закодировано контролируемое приложение: файл usr.bin.skype предназначен для программы skype, которая находится в каталоге /usr/bin/skype (первый слеш отбрасывается, остальные заменяются точкой).

Добавим эту программу к уже запущенным под контролем AppArmor. Для этого скопируем файл профиля в нужный каталог и перезапустим демон AppArmor:

в Kubuntu:

*root: cp /usr/share/doc/apparmor-profiles/extras/usr.bin.skype /etc/apparmor.d*

в OpenSUSE:

*root: cp /etc/apparmor/profiles/extras/usr.bin.skype /etc/apparmor.d/*

Теперь перезапустим демон и проверим, что приложение перешло под контроль AppArmor:

*root: rcapparmor reloadReloading AppArmor profiles : done.root: rcapparmor status...13 profiles are in complain mode. /usr/sbin/identd... /usr/bin/skype...*

**Переключение режимов контроля**

Статус AppArmor в OpenSUSE и Kubuntu показал различное отношение к режимам контроля в этих дистрибутивах: в OpenSUSE в режиме enforce находится 10 приложений, в то время как в Kubuntu (после установки дополнительных профилей) — всего 3, а основная часть – в режиме complain.

Для переключения режимов контроля в состав AppArmor входят специальные утилиты –enforce и complain Достаточно лишь указать команду и набрать имя нужного приложения. Вот, например, переключение режима контроля для утилиты ping:

*root: enforce ping*

Установка вынужденного режима для /etc/apparmor.d/bin.ping.

*root: complain ping*

Установка щадящего режима для /etc/apparmor.d/bin.ping.

**Как это работает**

Проведем небольшой эксперимент для того чтобы узнать, как это все работает. Для примера возьмем все ту же утилиту ping – в статусе мы видим, что эта утилита имеет профиль:

root: rcapparmor statusapparmor module is loaded.11 profiles are loaded.10 profiles are in enforce mode.... /bin/ping...

Даже не заглядывая в этот профиль, легко догадаться, что минимально необходимые привилегии ping не включают в себя доступ к каталогам пользователя. Вот и проверим, как ведет себя AppArmor в случае нестандартного поведения утилиты.

Сначала небольшая подготовка:

*user: touch ~/test*

Теперь мы заменим утилиту ping на что-нибудь другое, например, на cp (далее будем действовать от имени пользователя root):

*root: mv /bin/ping /bin/my\_pingroot: cp /bin/cp /bin/ping*

Раньше уже говорилось, что AppArmor сопоставляет каждой программе ее профиль согласно абсолютному пути к приложению, а значит, новой команде, несмотря на то что она полностью идентична утилите cp, сопоставляется профиль, предназначенный для ping.

Замечание 1. Мы видели, что в Ubuntu утилита ping контролируется AppArmor в режиме complain, а в OpenSUSE – в режиме enforce. Чтобы в дальнейшем не отвлекаться на это небольшое различие, переведите контроль в режим enforce:

*root: enforce bin.ping*

Установка вынужденного режима для /etc/apparmor.d/bin.ping.

Теперь попробуем «нестандартное» поведение (не забудьте заменить «user» в командах на правильное имя пользователя):

*root: ping /home/user/test /home/user/test1ping: cannot open `test' for reading*: Отказано в доступе

Ага, значит, AppArmor запретил нашей утилите скопировать файл. Что было записано влогах?

type=APPARMOR\_DENIED msg=audit(1238136923.088:35): operation="inode\_permission" requested\_mask="r::" denied\_mask="r::" fsuid=0 name="/proc/5911/mounts" pid=5911 profile="/bin/ping"type=APPARMOR\_DENIED msg=audit(1238136923.088:36): operation="inode\_permission" requested\_mask="r::" denied\_mask="r::" fsuid=0 name="/proc/filesystems" pid=5911 profile="/bin/ping"type=APPARMOR\_DENIED msg=audit(1238136923.088:37): operation="inode\_permission" requested\_mask="::r" denied\_mask="::r" fsuid=0 name="/home/lynx/test" pid=5911 profile="/bin/ping"type=APPARMOR\_DENIED msg=audit(1238136923.236:38): operation="inode\_permission" requested\_mask="r::" denied\_mask="r::" fsuid=0 name="/usr/share/locale-bundle/ru/LC\_MESSAGES/coreutils.mo" pid=5911 profile="/bin/ping"

Замечание 2. В OpenSUSE отчеты AppArmor записываются в файл /var/log/audit/audit.log, а в Kubuntu сообщения идут напрямую в /var/log/messages. Выше показано сообщение из OpenSUSE, формат вывода в Kubuntu отличается незначительно.

Не слишком понятные сообщения, но из них видно, что

type=APPARMOR\_DENIED: сообщение от AppArmor, который кому-то что - то запретил;

requested\_mask="r::", denied\_mask="r::": тип действия, которое программа пыталасьвыполнить и которое ему запретили;

fsuid=0: идентификатор пользователя, нулевой идентификатор принадлежит root;

pid=5911: идентификатор процесса, именно под таким запустилась наша «исправленная»утилита ping;

profile="/bin/ping": собственно профиль программы.

Теперь переведем нашу утилиту ping в режим контроля complain:

root: complain bin.ping

Установка щадящего режима для /etc/apparmor.d/bin.ping.

*root: ping /home/user/test /home/user/test1root: ls /home/user | grep testtesttest1*

Все прошло замечательно! Файл был успешно скопирован. А что появилось нового в логах?

type=APPARMOR\_ALLOWED msg=audit(1238137012.864:40): operation="inode\_permission" requested\_mask="r::" denied\_mask="r::" fsuid=0 name="/proc/5948/mounts" pid=5948 profile="/bin/ping"type=APPARMOR\_ALLOWED msg=audit(1238137012.868:41): operation="inode\_permission" requested\_mask="r::" denied\_mask="r::" fsuid=0 name="/proc/filesystems" pid=5948 profile="/bin/ping"type=APPARMOR\_ALLOWED msg=audit(1238137012.868:42): operation="inode\_permission" requested\_mask="::r" denied\_mask="::r" fsuid=0 name="/home/lynx/test" pid=5948 profile="/bin/ping"type=APPARMOR\_ALLOWED msg=audit(1238137012.868:43): operation="inode\_permission" requested\_mask="w::" denied\_mask="w::" fsuid=0 name="/home/lynx/test1" pid=5948 profile="/bin/ping"type=APPARMOR\_ALLOWED msg=audit(1238137012.868:44): operation="setattr" requested\_mask="w::" denied\_mask="w::" fsuid=0 attribute="size,mtime,ctime," name="/home/lynx/test1" pid=5948 profile="/bin/ping"

Основное отличие от предыдущих записей аудита – это type=APPARMOR\_ALLOWED: AppArmor разрешил выполнение действия, хотя и оставил сообщение в логах.

Не забудьте удалить результаты ваших экспериментов:

root: rm test test1

и вернуть на место утилиту ping:

root: mv /bin/my\_ping /bin/ping

Теперь, когда мы убедились в работоспособности AppArmor и узнали где искать отчеты о нестандартном поведении, можно применять на практике принцип «запрещено все, что не разрешено явно».

**Что защищать?**

Авторство на AppArmor

AppArmor был создан компанией Immunix, которую позже поглотила Novell. В 2006 году компания Novell открыла исходный код AppArmor под лицензией GPL.

Чтобы узнать какие программы следует ограничивать профилями AppArmor, загляните в новостную ленту сообщений об уязвимостях. Какие программы там упоминаются? Есть ли они у вас? Браузер и почтовый клиент, сервисы, к которым можно обратиться из Интернет(ssh, ftp, ntp и т.д.), программы, в которых вы открываете файлы, скачанные из Интернет: текстовый редактор, графический редактор/просмотрщик... Ваш выбор будет зависеть от серьезности отношения к информационной безопасности и критичности Linux-системы. Вы можете передать под контроль AppArmor только часто используемые программы, с помощью которых открываете файлы, полученные из неизвестных источников (читай: полученные из Internet), либо защитить все, что попадет в поле зрения. В любом случае, используя систему защиты на основе принципа «запрещено все, что не разрешено явно», вы изрядно затрудните жизнь потенциальным злоумышленникам.

Подведем итоги.

В этой статье было дано краткое введение в работу системы AppArmor, позволяющей без особых усилий повысить защищенность Linux-системы. Мы рассмотрели только использование стандартных профилей, которые, тем не менее, существуют в большом количестве для самых разных программ. В следующей статье мы остановимся подробнее на создании новых профилей, рассмотрим синтаксис правил и вспомогательные утилиты.

Не все потенциальные и реальные Linux-пользователи любят работать в командной строке