Содержание:

Введение

Ubuntu — дистрибутив Linux, основанный на Debian GNU/Linux. Основным разработчиком и спонсором является компания Canonical. В настоящее время проект активно развивается и поддерживается свободным сообществом.

По утверждениям Canonical, Ubuntu используется примерно 20 миллионами пользователей по всему миру. Он является 1-м в списке самых популярных дистрибутивов Linux для веб-серверов. По количеству пользователей, посетивших сайт DistroWatch.com (на 2017 год), занимает 4-е место.

Обычно новые версии дистрибутива выходят каждые полгода и поддерживаются обновлениями безопасности в течение 9 месяцев (начиная с версии 13.04, до этого поддержка осуществлялась в течение полутора лет).

Версии LTS, выпускаемые раз в 2 года, поддерживаются в течение 5 лет — как серверные, так и десктопные варианты. (До версии 12.04 LTS срок поддержки для десктопных LTS-версий составлял 3 года.) На другие дистрибутивы LTS семейства Ubuntu действует полная поддержка в 3 года, а для основы системы (ядро, Xorg и прочие компоненты) — 5 лет.

Ubuntu поставляется с подборкой программного обеспечения для серверов и рабочих станций. Она устанавливается на настольные персональные компьютеры с помощью Live CD (версия Desktop)(возможно использование DVD и USB накопителей). Ранее присутствовала возможность использования текстового установщика (версия Alternate, предоставлялась до версии Ubuntu 12.04.2) и предоставлялись отдельные версии для CD и DVD дисков. В последней присутствовали несколько большие возможности — начиная от установки не только в графическом, но и в текстовом режимах, загрузки в режиме восстановления системы и заканчивая полной локализацией и большим количеством пакетов на диске. Есть версии для официально поддерживаемых архитектур, таких как i386 (до 20.04), AMD64, ARM. Кроме того, с 2013 года начата

разработка специальной версии Ubuntu для смартфонов на архитектуре ARM и x86.

Существует редакция Ubuntu Core, которая может работать на IoT-устройствах, и на роботах.

Изначально Ubuntu создавалась как временное ответвление от Debian с целью регулярно выпускать новую версию операционной системы каждые шесть месяцев. В отличие от других ответвлений Debian общего назначения, таких как Xandros, Linspire и Libranet, Canonical осталась близка к философии Debian и включает в Ubuntu в основном свободное программное обеспечение вместо того, чтобы частично положиться на несвободные добавления. Пакеты Ubuntu по большей части базируются на пакетах из нестабильной (unstable) группы пакетов Debian. В Ubuntu используется Advanced Packaging Tool от Debian для управления установленными пакетами. Тем не менее, пакеты для Ubuntu и Debian не обязательно совместимы друг с другом. Некоторые разработчики Ubuntu также занимаются ключевыми пакетами Debian, поэтому в случае внесения изменений в собираемые программы они вносятся в оба проекта. Однако в апреле 2005 основатель Debian Ян Мёрдок критиковал Ubuntu за несовместимость с пакетами Debian, говоря, что Ubuntu слишком далеко отклонился от Debian Sarge, чтобы остаться совместимым.

Ubuntu в настоящее время финансируется Марком Шаттлвортом и основанной им компанией Canonical. 8 июля 2005 Canonical объявила о создании Ubuntu Foundation и обеспечила начальное инвестирование в размере 10 миллионов долларов. Цель фонда состоит в том, чтобы гарантировать поддержку и развитие для всех будущих версий Ubuntu, но на 2009 год фонд остаётся незадействованным. Шаттлворт описывает его как чрезвычайный фонд на чёрный день.

В Ubuntu 17.10, была прекращена поддержка i386-процессоров, но версия для i386-процессоров могла быть установлена с Minimal CD или посредством обновления с предыдущих версий. В Ubuntu 17.10 также стала использоваться Wayland вместо Xorg, но в Ubuntu 18.04 LTS снова стала использоваться Xorg.

В Ubuntu 20.04 поддержка 32 битных процессоров прекращена окончательно.

Ubuntu ориентирована на удобство и простоту использования. Она включает широко распространённое использование утилиты sudo, которая позволяет пользователям выполнять администраторские задачи, не запуская потенциально опасную сессию суперпользователя.

Ubuntu, кроме того, имеет развитую интернационализацию, обеспечивающую максимальную доступность для представителей разных языковых групп. С версии 5.04 кодировкой по умолчанию является UTF-8.

Ubuntu для работы рекомендуется от 512 мегабайт RAM и, при установке на жёсткий диск, от пяти гигабайт свободного пространства, а предельно минимальные требования гораздо ниже.

Версия 6.06 и более поздние объединяют Live CD и установочный CD в один компакт-диск. Этот диск загружает рабочий стол со всеми возможностями, давая пользователям возможность узнать, поддерживаются ли их аппаратные средства, и экспериментировать с доступными приложениями, и уже затем устанавливать Ubuntu на жёсткий диск, используя графический инсталлятор Ubiquity (англ.) («вездесущность»). Однако можно перейти непосредственно к установке. Инсталляционный процесс сохраняет документы, созданные на «живом» рабочем столе. Альтернативная установка, использующая debian-installer, доступна для скачивания и нацелена на людей, разбирающихся в системе на более глубоком уровне, администраторов, устанавливающих много систем, и для сложного разбиения дисков, включая использование LVM или RAID, а также для установки с объёмом оперативной памяти менее 192 мегабайт. Также в дистрибутив входит программа создания загрузочного Live USB на базе USB Flash-диска, обладающего всеми возможностями Live CD и установочного CD. Это удобно для использования, например, на нетбуках. Однако на старых компьютерах не всегда есть опция загрузки с USB-флеш-накопителя.

При разработке компонентов Ubuntu активно используется язык программирования Python.

Основная часть

Ubuntu как один из наиболее популярных дистрибутивов обладает довольно развитыми защитными механизмами.

Основные защитные механизмы Ubuntu, будучи дистрибутивом на базе ядра Linux, заложены непосредственно в самом ядре Linux.

Самым основным набором защитных механизмов является Linux Security Modules (LSM), включающий в себя такие компоненты безопасности как: AppArmor, SELinux,

Smack и TOMOYO Linux.

LSM представляют собой реализацию в виде подгружаемых модулей ядра. В первую очередь, LSM применяются для поддержки контроля доступа. Сами по себе LSM не обеспечивают систему какой-то дополнительной безопасностью, а лишь являются неким интерфейсом для её поддержки.

Система LSM обеспечивает реализацию функций перехватчиков, которые хранятся в структуре политик безопасности, охватывающей основные операции, защиту которых необходимо обеспечить. Контроль доступа в систему осуществляется благодаря настроенным политикам.

Большинство операционных систем обладают средствами и методами управления доступом, которые в свою очередь определяют, может ли некий объект на уровне операционной системы (пользователь или программа) получить доступ к определенному ресурсу. Используются следующие методы управления доступом:

- Дискреционный контроль доступа (англ. Discretionary Access Control, DAC). Данный метод реализует ограничение доступа к объектам на основе групп, к которым они принадлежат. В Linux, в основе этого метода, лежит стандартная модель контроля доступа к файлам. Права доступа определены для следующих категорий: пользователь (владелец файла), группа (все пользователи, которые являются членами группы), другие (все пользователи, которые не являются ни владельцами файла, ни членами группы). Причем к каждой из групп предоставляются следующие права: права на запись, на чтение и на исполнение.
- Мандатное управление доступом (англ. Mandatory Access Control, MAC). Главным образом суть этого метода заключается в установлении определенных прав доступа субъекта к определенным объектам. В ОС реализовано следующее: программа обладает только теми возможностями, которые необходимы ей для выполнения своих задач, и не более того. Данный метод имеет преимущество в сравнении с предыдущим; если в программе будет обнаружена уязвимость, то возможности её доступа будут весьма ограничены.
- Контроль доступа на основе ролей (англ. Role-Based Access Control, RBAC). Права доступа реализованы в виде ролей, выдаваемых системой безопасности. Роли представляют собой определенные полномочия на выполнение конкретных действий группой субъектов над объектами в системе. Данная политика является улучшением дискреционного контроля доступа.

Модули LSM

AppArmor — программный инструмент упреждающей защиты, основанный на политиках безопасности (известных также как профили), которые определяют, к каким системным ресурсам и с какими привилегиями может получить доступ то или иное приложение. В AppArmor включён набор стандартных профилей, а также инструменты статического анализа и инструменты, основанные на обучении, позволяющие ускорить и упростить построение новых профилей.

Изначально программа была разработана компанией Immunix. После её приобретения компанией Novell инструмент был открыт под лицензией GNU GPL и включён в openSUSE. Позже адаптирован для Ubuntu.

В конце лета 2008 года Рассел Кокер, один из авторов SELinux, высказал мнение, что AppArmor бесперспективен, объяснив это тем, что даже в openSUSE появляется поддержка аналогичного и более популярного решения — SELinux. Однако вскоре разработку AppArmor продолжил сотрудник Canonical, а в июле 2010 года было объявлено о том, что AppArmor войдет в состав Linux-ядра версии 2.6.36. В мае 2013 года поддержка инструмента была внедрена в Debian 7 Wheezy.

SELinux — реализация системы принудительного контроля доступа, которая может работать параллельно с классической избирательной системой контроля доступа.

Оставаясь в рамках избирательной системы контроля доступа, операционная система имеет фундаментальное ограничение в плане разделения доступа процессов к ресурсам — доступ к ресурсам основывается на правах доступа пользователя. Это классические права rwx на трех уровнях — владелец, группа-владелец и остальные.

В SELinux права доступа определяются самой системой при помощи специально определенных политик. Политики работают на уровне системных вызовов и применяются самим ядром (но можно реализовать и на уровне приложения). SELinux действует после классической модели безопасности Linux. Иными словами, через SELinux нельзя разрешить то, что запрещено через права доступа пользователей или групп. Политики описываются при помощи специального гибкого языка описания правил доступа. В большинстве случаев правила SELinux «прозрачны» для приложений, и не требуется никакой их модификации. В состав некоторых дистрибутивов входят готовые политики, в которых права могут

определяться на основе совпадения типов процесса (субъекта) и файла (объекта) — это основной механизм SELinux. Две других формы контроля доступа — доступ на основе ролей и на основе многоуровневой системы безопасности. Например, «ДСП», «секретно», «совершенно секретно», «ОВ».

Самый простой для работы и с точки зрения поддержки тип политики — так называемая «целевая» политика, разработанная в рамках проекта Fedora. В рамках политики описано более 200 процессов, которые могут выполняться в операционной системе. Все, что не описано «целевой» политикой, выполняется в домене (с типом) unconfined_t. Процессы, работающие в этом домене, не защищаются SELinux. Таким образом, все сторонние пользовательские приложения будут без всяких проблем работать в системе с «целевой» политикой в рамках классических разрешений избирательной системы контроля доступа.

Кроме «целевой» политики, в состав некоторых дистрибутивов входит политика с многоуровневой моделью безопасности (с поддержкой модели Белла — Лападулы).

Третий вариант политики — «строгий». Тут действует принцип «что не разрешено, то запрещено» (принцип наименьших прав). Политика основывается на Reference Policy от компании Tresys.

SELinux был разработан Агентством национальной безопасности США, и затем его исходные коды были представлены для скачивания.

SELinux включён в состав ядра Linux (начиная с версии 2.6).

Также для функционирования SELinux требуются модифицированные версии некоторых утилит (ps, ls и других), которые обеспечивают поддержку новых функций ядра, и поддержка со стороны файловой системы.

В самом начале своего появления SELinux была реализована в виде патча. В данном случае было непросто настраивать политику безопасности. С появлением механизмов LSM, настройка и управление безопасностью значительно упростились (политика и механизмы усиления безопасности были разделены), SELinux была реализована в виде подгружаемых модулей ядра. Перед доступом к внутренним объектам операционной системы производится изменение кода ядра. Это реализуется при помощи специальных функций (перехватчиков системных вызовов), так называемых функций «хуков» (англ. hook functions). Функции-перехватчики хранятся в некоторой структуре данных, их целью является выполнение определенных действий по обеспечению безопасности, основанных на

заранее установленной политике. Сам модуль включает в себя шесть главных компонентов: сервер безопасности; кэш вектора доступа (англ. Access Vector Cache, AVC); таблицы сетевых интерфейсов; код сигнала сетевого уведомления; свою виртуальную файловую систему (selinuxfs) и реализацию функций-перехватчиков.

Возможности SELinux

- Разные политики в зависимости от поставленных задач
- Четкая реализация политик
- Поддержка приложений, запрашивающих политику и исполнение контроля доступа этих приложений (для примера, задача, запущенная в cron, с корректным контекстом)
- Независимые специфичные политики и интернациональные языковые политики SELinux
- Независимые специфичные форматы меток безопасности и их содержание
- Индивидуальные метки и рычаги управления для объектов ядра и сервисов (демонов)
- Кэширование доступных решений для эффективности
- Возможность изменения политик
- Различные меры для защищенности целостности системы и конфиденциальности данных
- Очень гибкие политики
- Управление процессом инициализации, наследование прав, запуск программ
- Управление файловыми системами, папками, файлами и открытыми дескрипторами
- Управление сокетами, сообщениями (ядра и системы) и сетевыми интерфейсами

Smack

— модуль безопасности ядра Linux, который защищает данные и взаимодействие процессов от злонамеренных манипуляций с помощью набора настраиваемых правил обязательного контроля доступа (MAC) с простотой в качестве основной цели разработки. Он был официально добавлен в Linux 2.6.25, является основным механизмом контроля доступа в мобильной операционной системе MeeGo. А также используется для изолирования веб-приложений HTML5 в архитектуре Tizen, в

коммерческой Wind River. Применяется для разработки встроенных устройств, в продуктах Philips Digital TV., и ОС Intel Ostro для устройств IoT.

С 2016 года Smack требуется во всех реализациях Linux автомобильного уровня (AGL), где он вместе с другими средствами Linux обеспечивает основу для инфраструктуры безопасности AGL.

Smack состоит из трех компонентов:

- Модуль ядра, реализованный как модуль безопасности Linux. Лучше всего он работает с файловыми системами, поддерживающими расширенные атрибуты.
- Сценарий запуска, который гарантирует, что файлы устройств имеют правильные атрибуты Smack, и загружает конфигурацию Smack.
- Набор исправлений для пакета GNU Core Utilities, чтобы он знал о расширенных атрибутах файлов Smack. Также был создан набор патчей, похожих на Busybox. SMACK не требует поддержки пространства пользователя.

Smack критиковали за то, что он был написан как новый модуль LSM вместо политики безопасности SELinux, которая может обеспечивать эквивалентную функциональность. Такие политики SELinux предлагались, но не были продемонстрированы. Автор Smack ответил, что это будет непрактично из-за сложного синтаксиса конфигурации SELinux и философской разницы между Smack и SELinux.

Tomoyo Linux

— модуль безопасности ядра Linux, который реализует обязательный контроль доступа (MAC).

Tomoyo Linux - это реализация MAC для Linux, которую можно использовать для повышения безопасности системы, а также использовать исключительно как инструмент системного анализа. Он был запущен в марте 2003 года и до марта 2012 года спонсировался NTT Data Corporation.

Tomoyo Linux фокусируется на поведении системы. Tomoyo Linux позволяет каждому процессу объявлять поведение и ресурсы, необходимые для достижения их цели. Когда защита включена, Tomoyo Linux ограничивает каждый процесс поведением и ресурсами, разрешенными администратором.

Обеспечивает защиту от неизвестных уязвимостей, так называемых Zero-Day Exploit.

Для нормальной работы приложения, необходимо лишь определить профили с отдельными политиками безопасности. Как системные администраторы, так и пользователи могут создавать собственные профили для приложений. Также TOMOYO может использовать адаптивную фильтрацию во время работы программы в обычном режиме.

Основные особенности Tomoyo Linux

- Системный анализ
- Повышенная безопасность за счет обязательного контроля доступа
- Автоматическая генерация политики
- Простой синтаксис
- Легкость использования

Так же имеется механизм безопасности ядра Linux — seccomp.

Seccomp обеспечивает возможность ограничивать набор доступных системных вызовов для приложений, а также с помощью механизма BPF (Berkeley Packet Filter) производить сложную фильтрацию вызовов и их аргументов. Впервые появился в ядре версии 2.6.12 в 2005 году.

Для работы с недоверенными или непроверенными, а поэтому потенциально опасными программами желательно использовать специально выделенные среды, из которых нельзя нанести вред работоспособности системы в целом. В таких средах (песочницах, контейнерах) для запускаемых программ лимитированы многие системные возможности, такие как доступ к сети, устройствам ввода-вывода, взаимодействие с операционной системой. Механизм seccomp определяет для процесса набор разрешённых системных вызовов и блокирует те, которые не были заранее объявлены. В настоящее время используется в ряде браузеров, Linux подобных ОС и некоторых системах виртуализации.

Так же для повышения уровня безопасности Linux делится на уровни (кольца) безопасности, всего их 4 (отсчёт начинается с 0). 0 уровнем является Ядро операционной системы, на этом уровне предоставляется полный доступ ко всем ресурсам системы. Последним же 3-им уровнем является уровень приложения.

Ещё одним механизмом безопасности Ubuntu является cgroups.

Контрольная группа (cgroups)

— группа процессов в Linux, для которой механизмами ядра наложена изоляция и установлены ограничения на некоторые вычислительные ресурсы (процессорные, сетевые, ресурсы памяти, ресурсы ввода-вывода). Механизм позволяет образовывать иерархические группы процессов с заданными ресурсными свойствами и обеспечивает программное управление ими.

Разработка была начата инженерами Google Полом Менэджем (Paul Menage) и Рохитом Сетом (Rohit Seth) в 2006 году и первоначально называлась «контейнеры процессов» (англ. process containers). В 2007 году проект был переименован в cgroups (от англ. control groups) по причине неоднозначности значения термина «контейнер» в ядре Linux.

Начиная с версии 2.6.24 ядра Linux технология включена в официальные версии ядра. С этого момента разработка значительно активизировалась, в механизм добавлено много дополнительных возможностей, механизм существенным образом используется в технологии инициализации systemd, а также является ключевым элементом в реализации системы виртуализации на уровне операционной системы LXC.

Одна из целей механизма — предоставить единый программный интерфейс к целому спектру средств управления процессами, начиная с контроля единичного процесса (таких как, например, утилита nice) вплоть до полной виртуализации на уровне системы (как у OpenVZ, Linux-VServer, LXC). Механизм предоставляет следующие возможности:

- ограничение ресурсов (англ. resource limiting): использование памяти, в том числе виртуальной;
- приоритизацию: разным группам можно выделить разное количество процессорного ресурса и пропускной способности подсистемы ввода-вывода;
- учёт: подсчёт затрат тех либо иных ресурсов группой;
- изоляцию: разделение пространств имён для групп таким образом, что одной группе недоступны процессы, сетевые соединения и файлы другой;
- управление: приостановку (freezing) групп, создание контрольных точек (checkpointing) и их перезагрузку.

LXC

— система виртуализации на уровне операционной системы для запуска нескольких изолированных экземпляров операционной системы Linux на одном узле. LXC не использует виртуальные машины, а создаёт виртуальное окружение с собственным пространством процессов и сетевым стеком. Все экземпляры LXC используют один экземпляр ядра операционной системы.

Данная система сходна с OpenVZ и Linux-VServer для Linux, а также FreeBSD jail и Solaris Containers. LXC основана на технологии cgroups, входящей в ядро Linux, начиная с версии 2.6.29.

Основные разработчики — Даниэль Лескано (Daniel Lezcano), Серж Айюн (Serge Hallyn) и Стефан Грабе (Stéphane Graber).

Среди примеров использования — применение в PaaS-хостинге Heroku для изоляции динамических контейнеров (dynos). В проекте Docker разработаны компоненты, обеспечивающие LXC высокоуровневыми сервисами управления и развёртывания.

netfilter — межсетевой экран (брандмауэр), встроен в ядро Linux с версии 2.4.

Заключение

Ubuntu включает в себя огромное количество встроенных механизмов безопасности, многие из которых встроены непосредственно в ядро Linux. Использование последних версий ядра Linux позволяет получить более новые механизмы безопасности и защиты.

Благодаря активной поддержке сообщества и наибольшей клиентской базе, Ubuntu может получать исправления безопасности быстрее многих дистрибутивов, на ровне со специализированными корпоративными дистрибутивами, как RedHat Enterprise Linux (RHEL) и др.

Новые сферы применения Ubuntu, такие как интернет вещей (IoT), пограничные вычисления (edge computing), робототехника возлагают на данный дистрибутив дополнительные требования, связанные с организацией безопасности.