**4 слайд:**

Разработка Astra Linux была начата в 2008 году АО «НПО РусБИТех». Стало недостаточно простых реализаций политик безопасности, которые содержат много логических и практических уязвимостей. Существовала версия SELinux (Security-Enhanced Linux), но это зарубежная разработка, которая могла содержать потенциальные закладки от спецслужб других стран, поэтому было решено отказаться от применения данного ядра.

Собрались лучшие умы и решили реализовать всё то, что пишется в книжках по мбкс с упором на отечественные разработки, так в Astra Linux появились модели мандатного разграничения доступа, запатентованная мандатная сущностно-ролевая ДП-модель управления доступам и информационными потоками (МРОСЛ ДП-модель), которая лишена недостатков модели Белла-Лападулы (деклассификация, нарушение логики доступа к данным при обработке потока инофрмации в распределенной среде) и дополнительные способы разграничения доступа.

(МРОСЛ ДП-модель — мандатная сущностно-ролевая ДП-модель.

ДП-модель — модель логического управления доступом «Д» и информационными потоками «П»)

Указанная математическая модель реализована в программном коде специалистами АО «НПО „РусБИТех“» и Академии ФСБ России и верифицирована Институтом системного программирования Российской академии наук. В результате дедуктивной верификации[46] модель была полностью формализована и верифицирована[47].

В настоящее время используемая в Astra Linux Special Edition модель разграничения доступа является единственной практически реализованной моделью, не основанной на SELinux, в российских реализациях операционных систем на базе Linux.

**5 слайд:**

На скриншоте представлены различия между системами. Самая слабая, версия Орел или же Astra Linux Common Edition.

Самая сильная – Смоленск или же Astra Linux Special Edition.

Орел

Соответствует действующим условиям лицензирования ОС Astra Linux Common Edition и рекомендуется для обработки общедоступной информации в составе значимых объектов КИИ, не требующих применения СЗИ, прошедших оценку соответствия требованиям безопасности информации в форме обязательной сертификации, а также для применения в информационных системах, обрабатывающих информацию ограниченного доступа, совместно с сертифицированными средствами технической и криптографической защиты информации.

Воронеж

Рекомендуется для обработки конфиденциальной информации в ГИС, в информационных системах персональных данных, а также в составе значимых объектов КИИ. Дополнительно используется в других информационных (автоматизированных) системах для обработки информации ограниченного доступа без содержания сведений, составляющих гостайну.

Смоленск

Рекомендуется для обработки информации любой категории доступа в ГИС, в информационных системах персональных данных, в составе значимых объектов КИИ, иных информационных (автоматизированных) системах, обрабатывающих информацию ограниченного доступа, в т.ч. содержащую сведения, составляющие гостайну до степени «совершенно секретно» или «особой важности» (в зависимости от класса защиты, установленного в сертификате соответствия для используемого конечным пользователем варианта исполнения Astra Linux) включительно.

**6 слайд:**

● ОС Astra Linux Special Edition обеспещивает защиту информации, содержащую сведения, составляющие государственную тайну с грифом не выше «совершенно секретно»

● Система защиты информации (СЗИ) Astra Linux основана на мандатной сущностно-ролевой модели управления доступом и информационными потоками в ОС (МРОСЛ ДПмодель)

● СЗИ реализована в виде модуля безопасности ядра ОС Linux (Linux Security Module, LSM)

**8 слайд:**

Модель оперирует со следующими основными понятиями:

● Сущности

● Сессии

● Учетные записи пользователей

● Роли

● Права доступа

● Доступы

Модель описывает ограничения на доступы к сущностям и ролям, опираясь на три механизма защиты информации:

● Мандатный контроль целостности (MIC)

○ Высоко-целостные сущности должны быть защищены от модификации низко-целостными сессиями.

● Мандатное управление доступом (MAC)

○ Доступы к сущности от сессий, которые не обладают сопоставимой меткой конфиденциальности, запрещены.

● Управление доступом на основе ролей (RBAC)

Модель определяет 44 операции в форме программных контрактов:

● 34 из них описывают:

○ создание или удаление сущностей, сессий, ролей, учетных записей пользователей

○ изменение меток целостности и конфиденциальности

○ получение и удаление доступов и прав доступа

● 10 операций служат для более точного анализа безопасности модели в терминах информационных потоков по памяти и по времени

**9 слайд**:

Посмотрим на дистрибутив в живую и рассмотрим процесс установки.

Как видно на скриншоте, установка возможна двумя способами.

Установка с помощью графического окна, что далее будет продемонстрировано

Установка с помощью графического окружения, которое нарисовано в ncurses. Для того, чтобы понять как это, вспомните, когда мы писали танчики на С на парах Андрея Васильевича. Данный вариант установки подходит для более опытных пользователей, т.к. установка может быть неинтуинтивна понятна пользователю.

Третий вариант, Режим Восстановления, загружает систему в Live режиме и позволяет починить возникшие проблемы, которые требуют отмонтированный диск.

**10 слайд:**

На деле, обычное пользовательское соглашение, но забавно видеть в ОС что-то непосредственно относящееся к законодательству РФ.

**11 слайд**:

Далее происходит обычная настройка дистрибутива, ничем не отличная от настройки Debian 10-11 версий. Поэтому некоторые этапы мы пропустим.

**12 слайд:**

Одной из интересной особенностью Astra Linux является то, что при установке на выбор пользователю предлагается несколько вариантов ядер. На мой взгляд необычное решение, т.к. при установке системы тебе часто хочется получить последние обновления и самую свежую ОС, соответственно с самым свежим ядром. Но справедливости ради, Astra Linux по умолчанию предлагает поставить именное такое ядро linux-5.10-generic.

Расскажем различия между выбор ядрами:

1. Нумерация, 4.15, 5.10, 5.4 – говорит о нумерации ядер в процессе их выпуска. Чем выше номер, тем новее и свежее ядро.
2. Отличия версии hardened от generic, заключается в том, что hardened ядра ориентированны на безопасность ядра с определенным набором патчей, защищающих от эксплойтов ядра и пространства пользователя. Содержит больше защитных особенностей, чем linux.

Стоит также учитывать, что все эти ядра модифицированны и написаны специально под Астру, если у вас есть время и желание, то это ядро можно пореверсить и разобраться в том, как оно устроено. Отмечу, что данную процедуру можно проводить только в образовательных целях, т.к. это проприетарное ядро и незаконная обратная разработка уголовно наказуема.

**13 слайд**:

Astra Linux отличается от других дистрибутивов тем, что предлагает при установке сразу выбрать защитные механизмы для системы. Рассмотрим каждый из них подробнее:

* Включить блокировку консоли.
  + Опция отключает запуск терминала Fly, который предустановлен по умолчанию. Опция может быть полезна при установке на машины, которые будут использоваться простыми пользователями для исключения возможных вредоносных действий. Например, злоумышленник может попросить жертву открыть консоль и ввести туда несколько команд, смысл которых обычный пользователь не понимает, что может привести к утечке данных организации.
* Включить блокировку интерпретаторов
  + Опция отключает различные интерпретаторы, например perl, python и другие.
* Включить межсетевой экран ufw
  + Встроенный фаерволл в Linux, который позволяет управлять сетью.
* Включить системные ограничения ulimits
  + Опция позволяет более тонко настроить систему, например ограничить количество одновремнных открытых файловых дескрипторов в ОС.
* Отключить возможность трассировки ptrace
  + Опция отключает возможность дебага программ в данной ОС. Почему так? В Linux механизм ptrace аналог win API CreateRemoteDebugger, который позволяет подключиться к процессу и управлять его виртуальной памятью, ходом исполнения и т.д.
* Запретить установку бита исполнения
  + Опция позволяет запретить помечать файлы, как исполняемые. Что это значит? Поскольку в ОС Linux отсутствует как таковое понятие расширение файла, то пользователь сам выбирает через какую программу ему открыть тот или иной файл. Например, ELF можно открыть в шестнадцатеричном редакторе, открыть в текстовом редакторе или даже в качестве картинки. И эта опция как раз и говорит о том, что пусть пользователь делает все что хочет, но исполнять данный ELF он не будет, поскольку непосредственно для исполнения файла необходимы эти права.
* Использовать sudo с паролем
  + При запуске команды sudo для исполнения под root пользователем теперь будет спрашивать пароль пользователя.

Остальные опции интуитивно понятны и не требуют дополнительного разъяснения.

**15 слайд:**

Как видно на скришоте, загрузочный экран отличается, что уже говорит о другой версии системы, но логика осталась прежняя.

Лицензионное соглашение идентично тому, что выводится в Astra Linux Common Edition.

Поэтому сразу перейдем к следующему интересному этапу.

**16 слайд:**

В отличие от Astra Linux Common Edition данная версия предлагает несколько уровней защищенности ОС. По умолчанию выбирается максимальный Смоленск.

Также как было отмечено в начале презентации, проект Astra Linux предлагает 3 уровня защищенности, которые в самой младшей версии выбрать нельзя.

**17 слайд**:

Как можно видеть на скриншоте, появляются прославленные Мандатный контроль целостности и Мандатное управление доступом, которое, кстати выбрано по умолчанию.

Также появляются новые дополнительные опции, такие как замкнутная программная среда – система будет полностью изолирована от окружающего мира, а также очистка освобождаемой внешней памяти – это принудительная очистка ОЗУ, для предотвращения утечки данных. Например, мы запустили программу КриптоПРО и разработчики просто забыли очистить и забить нулями чанк на куче, в которой хранился введеный пользователем пароль. Это потенциальная опасность и утечка данных. Поэтому эта опция была придумана именно для этого, чтобы убрать из памяти чувствительные данные.

**18 слайд**:

Так выглядит окно входа в систему на Astra Linux Special Edition. По моему личному мнению, оно выглядит гораздо приятнее, нежели на Common Edition.

**19 слайд:**

При входе в систему появляется окно, которое предлагает выбрать контроль целостности под которым мы будем работать дальше.

**20 слайд:**

В качестве интереса сравнение дистрибутивов лоб в лоб. Слева Common Edition, справа Special Edition. Как можно заметить, красный цвет говорит нам о том, что у нас выбран высокий уровень целостности.

**21 слайд:**

Также стоит отметить то, что Astra Linux Special Edition не подходит для личного использования от слова совсем, поскольку обновления системы и пакетов происходит только с поставкой новой версии ОС. Поэтому если в ядре или каком-то пакете была обнаружена CVE устранить ее будет достаточно проблематично, в отличие от Astra Linux Common Edition, где можно выполнить обновление пакетов с помощью публичного репозитория.

В Astra Linux SE можно видеть сообщение о том, что команда исполняется под привелигерованным пользователем.

**22 слайд**:

Как показано на скриншоте слева, в Astra Linux CE установлено более свежее ядро, в отличие от Astra Linux SE, что также порождает определенные проблемы. Т.к. на данный момент ядро 5.4 является относительно старым и содержит множество ядерных уязвимостей вида LPE, а также RCE, что критично для данной системы.

**26 слайд:**

**Краткая история обнаружения уязвимости Dirty Pipe:**

Исследователь, Макс Келлерманн, нашедший данную уязвимость, обнаружил ее в процессе обработки обращения в службу поддержки по поводу поврежденных файлов, а именно: клиент пожаловался, что загруженные им журналы доступа не могут быть распакованы. Он обнаружил, что файл журнала действительно был поврежден, однако он может быть распакован, при ручном исправлении CRC (Cyclic redundancy check/ Циклический избыточный код). После данного инцидента он отложил его изучение на месяцы, что в дальнейшем позволило ему провести анализ подобных инцидентов и кода, который и позволил ему сделать вывод о том, что ошибка кроется в ядре Linux.

**27 слайд:**

**Технические детали уязвимости Dirty Pipe**

С помощью этой уязвимости непривилегированный локальный пользователь может модифицировать содержимое кэша страниц памяти, связанных с файлами, доступными только для чтения, и таким образом повышать свои привилегии в системе. Уязвимость возникает из-за использования частично неинициализированной памяти в структуре буфера *конвейера* во время его построения. Отсутствие нулевой инициализации нового члена структуры приводит к использованию устаревшего значения *флагов*. Злоумышленники могут воспользоваться этим для получения доступа на запись к страницам памяти в кэше, даже если эти страницы были изначально помечены атрибутом «*Только для чтения»*.

**28 слайд:**

Как видно на скришоте, благодаря уязвимости в ядре эскалация привелегий была выполнена успешно.

Давайте теперь попробуем другой подход.

**31 слайд:**

Что такое **мандатный контроль целостности**? Поясним на примере. Ключевым компонентом ОС является ядро. Соответственно, мы обязаны обеспечить для него максимально защищенную среду выполнения в самой операционной системе, чтобы уменьшить количество возможных способов атаки на ядро.  
  
Для этого в операционной системе реализуется мандатный контроль целостности, за счет чего ОС сегментируется по различным подсистемам — так, чтобы взлом одной подсистемы не повлиял на работоспособность других. Если произойдет взлом непривилегированного пользователя ОС (уровень целостности 0) или сетевой подсистемы (уровень целостности 1), системы виртуализации (уровень целостности 2), графического интерфейса (уровень целостности 8) или другого компонента, это не повлечет за собой дискредитацию всего КСЗ (уровень целостности 63).  
  
При этом надо отметить, что указанные уровни не являются иерархическими, то есть не располагаются друг над другом и полностью изолированы друг от друга с точки зрения возможности прав записи. Принадлежность объекта к тому или иному уровню целостности монитор обращений определяет по битовой маске.

Чтобы уровни целостности не воспринимались как иерархические — то есть, например, «уровень 8 имеет больше прав, чем уровень 2», что неверно — каждый из уровней получает свое наименование. Так, например, восьмой уровень целостности называется «Графический сервер», максимально возможный уровень целостности администратора в системе — «Высокий», а нулевой уровень целостности (пользовательский) — «Низкий».

Таким образом операционная система получает набор правил, как изолировать друг от друга системные процессы, и теперь понимает, какие именно процессы, даже запущенные пользователем с высокими привилегиями, не имеют права на запись в другие процессы или файлы.

**32 слайд:**

Если в результате эксплуатации уязвимости (в том числе, нулевого дня) злоумышленник получит контроль над каким-либо процессом в системе и повысит свои полномочия до привилегированного пользователя (например, root), его метка целостности останется прежней, и, соответственно, он не получит возможности влиять на системные процессы, менять настройки или скрыть свое присутствие в системе.

Таким образом, значимой мишенью для злоумышленника становится уже не вся операционная система, а только hardened ядро и максимально компактный монитор обращений, что уже существенно сокращает поверхность атаки.

Помимо мандатного, есть еще динамический и регламентный контроль целостности. Их применяют для исключения запуска и использования недоверенного или стороннего ПО, а также периодических проверок целостности системы.

**33 слайд:**

*Динамический контроль* вычисляет и проверяет электронную цифровую подпись исполняемых файлов в момент их запуска. Если ЭЦП нет или она неправильная, в запуске программ будет отказано. В какой-то степени это реализация концепции белых списков, но за счет использования иерархии ключей, выданных разработчикам программного обеспечения.

**35 слайд:**

Как видно, после включения динамического контроля целостности система просто убивает неподписанный бинарный файл.

**36 слайд:**

*Регламентный контроль* проверяет целостность и неизменность ключевых для системы файлов, сравнивая их контрольные суммы с эталонными значениями. Это могут быть как конфигурационные файлы, так и любые другие.   
Таким образом в ОС применяется эшелонированная защита от уязвимостей в приложениях и их подмены, чем минимизируется вред от угроз безопасности, в том числе и тех, которые используют уязвимости «нулевого» дня.

**37 слайд**:

Уязвимость PwnKit представляет из себя эксплойт вида LPE(Local Privelege Escalation), который позволяет повысить привелегии в системе и подняться до рута.

**38 слайд:**

**Описание уязвимости: PwnKit** **(CVE-2021-4034)**

Исследовательская группа Qualys обнаружила уязвимость, приводящую к повреждению памяти, в pkexec polkit, корневой программе SUID, которая устанавливается по умолчанию во всех основных дистрибутивах Linux. Эта легко эксплуатируемая уязвимость позволяет любому непривилегированному пользователю получить полные привилегии root на уязвимом хосте, используя эту уязвимость в конфигурации по умолчанию.

**39 слайд:**

**Технические детали уязвимости PwnKit:**

Уязвимость возникает из-за ошибки в коде pkexec, связанной с неправильной обработкой числа аргументов, когда при отсутствии аргументов (т. е. список аргументов argv, который передается системному вызову execve(), пуст) возможно передать и выполнить «небезопасную» переменную в окружение pkexec. Обычно такие «небезопасные» переменные удаляются из окружения SUID-программ с помощью ld.so до выполнения main()-функции, однако данная уязвимость позволяет вновь ввести подобную переменную. При этом, т. к. у pkexec установлен SUID-бит от root, код выполняется от Учетной Записи root.

**40 слайд:**

Как видно на скришоте, благодаря уязвимости в ядре эскалация привелегий была выполнена успешно.

Давайте теперь попробуем другой подход, такой же как мы применяли в Astra Linux Common Edition.

41 слайд: Живая демонстрация: если создать юзера и задать для него уровень целостности и корректные настройки, то это не поможет, поскольку работает мандатный контроль целостности.