# Безопасность

## Общие принципы безопасности

Информационная безопасность — это непрерывный процесс защиты информационных систем от угроз трех видов:

- несанкционированный доступ (НСД) и использование
- нарушение целостности, конфиденциальности, подлинности и других характеристик данных
- нарушение доступности и/или полноценного функционирования информационной системы

Подсистема безопасности не является выделенным компонентом ОС и ее практически невозможно добавить в систему пост-фактум, т.е. она должна быть встроена с самого начала.

Принципы создания безопасной системы:

- принцип безопасных настроек по умолчанию
- принцип валидации данных, поступающих от других участников системы
- принцип полной медации — всегда проверка текущих прав
- принцип наименьших привилегий — выдавать права, достаточные для выполнения только требуемых операций и не более того
- принцип разделения привилегий — по возможности, требовать согласия нескольких участников для выполнения операций
- принцип экономии на механизме — механизмы защиты должны быть максимально простыми из возможных и реализовываться на самом низком из возможных уровне
- принцип минимального общего между разными участниками системы
- прицип открытого дизайна — архитектура, реализация и используемые алгоритмы в системе должны быть известны, а в секрете могут держаться только ограниченные по объему авторизационные данные (ключи, пароли и т.п.)
- принцип психологической приемлемости

Основные сервисы системы безопасности описываются аббревиатурой AAA:

- Аутентификация (Authentication) — установление "личности" стороны, с которой происходит взаимодействие

 

* pdf generated on 2015-01-11 from markdown sources with SBCL 1.1.14 on Linux 3.11.0-12-generic with 3bmd, cl-pdf & cl-typesetting

- Авторизация (Authorization) — проверка прав на выполнение каких-либо операций в системе
- Аудит (Accounting) — учет операций, связанных с системой безопасности (для возможности последующего расследования и установления причин проблемы)

Способы (факторы) аутентификации:

- по паролю
- по вопросу безопасности
- по одноразовому паролю
- по жетону (уникальным числом)
- с помощью сертификата ЭЦП

Аутентификация может быть как однофакторной, так и многофакторной.

# Механизмы работы системы безопасности

В системе безопасности рассматриваются 3 сущности: участники системы (субъекты, пользователи), ресурсы (например, файлы) и права доступа. Единицей управления является **домен защиты** — это пара объект и права доступа. Домен может соответствовать одному субъекту или их группе.

Матрица контроля доступа — это теоретическая модель, которая описывает матрицу, которая приводит в соответствие все ресурсы системы со всеми ее субъектами. В ячейках этой матрицы находятся права доступа конкретного пользователя/роли/группы к конкретному ресурсу. Такая матрица позволяет описать все права доступа в системе, однако ее практическое применение не эффективно.

На практике используются 2 следующих подхода:

- списки контроля доступа (Access Control Lists, ACL), которые предполагают хранение и учет прав на уровне ресурсов, т.е., фактически, по столбцам этой матрицы
- мандатные системы (Capability или C-list), которые предполагают хранение прав доступа у субъектов системы, т.е. по строкам матрицам

В системе, основанной на ACL, для каждого ресурса определен список субъектов с их правами. Например, в ФС Unix у каждой директории и файла определены 3 типа субъектов: пользователь-владелец, группа-владелец и все остальные,— а также 3 типа прав: чтение, запись и выполнение. Другим


* pdf generated on 2015-01-11 from markdown sources with SBCL 1.1.14 on Linux 3.11.0-12-generic with 3bmd, cl-pdf & cl-typesetting

примером ACL является список правил межсетевого экрана, ресурсами в которых являются хосты/подсети и/или возможность обращения к определенным портам/использования определенных протоколов. В такой системе вместо прав доступа устанавливаются действия экрана при обращении: разрешить, запретить, ограничить и т.д. При этом, учитывая потенциальную неограниченность различных субъектов-участников сети, в такой системе в основном используются обобщенные субъекты: все хосты, все внешние хосты, все хосты из определенной подсети и т.д.

В системах на основе мандатов мандат выдается отдельно для каждого субъекта на каждое право доступа. Мандат, как правило, реализуется как числовой жетон (token), который выдается субъекту системой безопасности. Этот жетон может быть:

- просто уникальным числом, которое записывается в базу данных для кортежа пользователь, домен безопасности. Проблема такого способа в потенциально неограниченном количестве записей в системе с большим количеством ресурсов и/или субъектов. В такой системе автентифицированному субъекту достаточно предоставить жетон для того, чтобы идентифицировать ресурс, к которому он хочет получить доступ, и получить доступ
- криптографической величиной, полученной применением односторонней функции к паре домен безопасности и секретный ключ, известный только ядру системы, `c = f(domain, key)`. В этом случае для проверки мандата субъект должен предоставить не только сам жетон, но и идентификатор ресурса и права доступа. Проверка будет производится повторным вычислением значения функции над теми же аргументами. Безопасность системы основана на невозможности получить то же значение жетона без знания секретного ключа. В этой системе затруднен отзыв отдельных мандатов, поскольку для отзыва мандата нужно либо изменить идентификатор ресурса, что повлияет на всех субъектов, имеющих к нему доступ, либо изменить ключ, который, как правило, уникален для пользователя, но отзыв ключа приведет к отзыву всех мандатов этого пользователя. Для решения этой проблемы используются т.н. непрямые объекты.

Хотя с точки зрения модели Матрицы прав доступа в обоих системах хранится одна и та же информация, эта модель описывает только статические характеристики системы и не описывает ее поведения в динамике.

При рассмотрении динамики работы системы безопасности на основе мандатов обладают следующими преимуществами перед списками контроля

 

* pdf generated on 2015-01-11 from markdown sources with SBCL 1.1.14 on Linux 3.11.0-12-generic with 3bmd, cl-pdf & cl-typesetting

доступа:

- отсутствие необходимости использования общего пространства имен ресурсов, известного всем субъектам системы
- возможность более гранулярного учета прав: в системе на основе списка контроля доступа администраторам системы необходимо исчерпывающее знание о всех субъектах системы — поскольку это психологически неприемлимо, субъекты обычно агрегируются более общими сущностями — **принципалами** безопасности

В то же время в мандатных системах более трудно решить следующие проблемы:

- ограничить передачу мандата от одного субъекта другому (такая возможность также может быть и полезным свойством системы)
- отзыв мандатов, особенно выборочный отзыв отдельных прав, а не всех мандатов для какого-то субъекта

Во многих реальных системах используется комбинация обоих подходов: например, в файловой системе Unix ACL используются для первичного контроля прав, а для проверки текущих прав используются мандат, который выдается после первичной авторизации, проверка которого намного эффективнее.

Системы на основе мандата часто используются для создания т.н. "песочниц", например для выполнения кода, полученных из недоверенных источников — поскольку в такой системе проще реализовать принцип "по-умолчанию без доступа".

# Реализация системы безопасности

Аппаратная платформа предоставляет такие базовые механизмы для поддержки системы безопасности:

- кольца процессора (CPU rings), которые используются в сегментной организации памяти
- привилегированные инструкции (в некоторых платформах)
- рандомизация адресного пространства программы, защита стека и т.п.

Используя эти примитивы ОС выстраивает систему безопасности. Основа этой системы в большинстве ОС:

 

* pdf generated on 2015-01-11 from markdown sources with SBCL 1.1.14 on Linux 3.11.0-12-generic with 3bmd, cl-pdf & cl-typesetting

- это выполнение всех критических операций в ядре ОС и предоставление ограниченного интерфейса к ним через механизм системных вызовов
- поділ користувачів на адміністраторів (в Unix-системах: особливий користувач `root`) і звичайних користувачів

# Литература

- Secure Systems Design Principles
- Defensive Programming
- CWE/SANS Top 25 Most Dangerous Software Errors
- Capability Myths Demolished
- Classic Buffer Overflow Explained
- Priviledge Escalation Bug in Linux
- How to Exploit an XSS

 

* pdf generated on 2015-01-11 from markdown sources with SBCL 1.1.14 on Linux 3.11.0-12-generic with 3bmd, cl-pdf & cl-typesetting