**Угрозы целостности**

Целостность можно подразделить на **статическую (confidentiality)** (неизменность информационных объектов) и **динамическую** (корректное выполнение транзакций). **Транзакция** — группа операций, которые либо выполняются полностью, с соблюдением целостности данных, либо не выполняются вообще.

**Угрозы статической целостности**

С целью нарушения статической целостности злоумышленник может:

* ввести неверные данные;
* изменить данные.

Под данными здесь подразумевается не только предметная, но и служебная информация. Поэтому внедрение вредоносного ПО — это пример нарушения целостности программ.

К угрозам статической целостности относятся нарушения аутентичности и апеллируемости. При этом если обеспечить подлинность или неотказуемость невозможно, то соответствующие данные не могут рассматриваться в качестве доказательства.

**Ситуация: невозможность обеспечения аппелируемости**

Показательный случай нарушения целостности (но не подлинности — письмо изначально не было аутентичным) имел место в 1996 году. Служащая Oracle (личный секретарь вице-президента) предъявила судебный иск, обвиняя президента корпорации в незаконном увольнении после того, как она отвергла его ухаживания. В доказательство своей правоты женщина привела электронное письмо, якобы отправленное её начальником президенту. Содержание письма сейчас не важно, важно время отправки. Дело в том, что вице-президент предъявил, в свою очередь, файл с регистрационной информацией компании сотовой связи, из которого явствовало, что в указанное время он разговаривал по мобильному телефону, находясь вдалеке от своего рабочего места. Таким образом, в суде состоялось противостояние «файл против файла». Очевидно, один из них был фальсифицирован или изменён, то есть была нарушена его целостность. Суд решил, что подделали электронное письмо (секретарша знала пароль вице-президента, поскольку ей было поручено его менять), и иск был отвергнут (не удалось обеспечить неотказуемость по отношению к вице-президенту).

Рассмотрим далее некоторые примеры атак, нарушающих статическую целостность.

**Фарминг (pharming, подлог)** — перенаправление пользователя с легитимного веб-сайта на фиктивный. Этот приём может использовать уязвимость серверов доменных имён (DNS). DNS-серверы переводят символьные имена (www.mail.ru) в IP-адреса (212.51.24.76), в процессе фарминговой атаки эта функция изменяется и пользователю возвращается ip-адрес подложного сайта (199.64.10.91). При попадании на него пользователя, конфиденциальные данные обычно выуживаются путём фишинга. То же самое происходит при изменении на компьютере жертвы файла hosts. Таким образом, нарушается статическая целостность служебной информации, обеспечивающей корректную адресацию в Интернете, что создаёт условия для атак на раскрытие конфиденциальной информации.

**SQL-инъекция** — это способ взлома программ, работающих с базами данных. Такая атака может случиться из-за некорректной обработки входящих данных, используемых в SQL-запросах. SQL-инъекция часто даёт возможность атакующему выполнить произвольный запрос к базе данных и поэтому может рассматриваться как угроза не только целостности, но и конфиденциальности.

**Пример: SQL**

Предположим, что код, формирующий SQL-запрос (на Delphi), выглядит так:

statement := 'SELECT \* FROM users WHERE name = "' + userName + '";';

Если значение userName берётся из формы без каких-либо преобразований, то становится возможной SQL-инъекция. Так, если ввести значение "a"; DROP TABLE users; SELECT \* FROM data WHERE name LIKE "%   
то будет сгенерирована такая SQL-команда:

SELECT \* FROM users WHERE name = "a"; DROP TABLE users; SELECT \* FROM data WHERE name LIKE "%";

Правильный, безопасный код должен обеспечивать экранирование спецсимволов, т. е. в данном случае проводить замену кавычки на ", апострофа на ', обратной косой черты на \. Это можно делать таким кодом: statement := 'SELECT \* FROM users WHERE name = ' + QuoteParam(userName) + ';';

где QuoteParam() — некая функция, обеспечивающая экранирования. Perl, PHP, Java, Delphi и другие языки, ориентированные на базы данных, имеют встроенные средства, автоматически выполняющие эту операцию (для Delphi это свойство TQuery.Params).

**Угрозы динамической целостности**

Угрозами динамической целостности являются нарушение атомарности (неделимости) транзакций, переупорядочение, кража, дублирование данных или внесение дополнительных сообщений (сетевых пакетов и т. п.). Если эти действия происходят в процессе передачи данных по сети, то они называются активным прослушиванием.

**Пример: man in the middle (атака «человек посередине»)**

Атакой “man in the middle” называется способ компрометации канала связи, при котором злоумышленник, подключившись к каналу между контрагентами, осуществляет активное вмешательство в протокол передачи, удаляя, искажая информацию или навязывая ложную. При этом ни один из контрагентов не может догадаться о присутствии злоумышленника.

Такая атака может осуществляться и при использовании криптографических средств защиты. Допустим, Алиса хочет написать Бобу. А Мэлори хочет перехватить её сообщение или отправить Бобу ложное сообщение. Чтобы что-то написать, Алиса должна попросить у Боба его открытый ключ. Если Боб отсылает Алисе свой открытый ключ, но Мэлори может его перехватить, это может привести к атаке “man in the middle”. Она реализуется следующим образом. Мэлори отсылает Алисе сфальсифицированное сообщение, якобы от Боба, но включает в него свой открытый ключ. Алиса, полагая, что это открытый ключ Боба, шифрует своё сообщение открытым ключом Мэлори и отправляет его Бобу. Мэлори снова перехватывает сообщение, расшифровывает его, оставляет себе копию и зашифровывает его (после внесения изменений, если они нужны) при помощи открытого ключа, который Боб отправлял Алисе. Когда Боб получит это сообщение, он будет думать, что оно пришло от Алисы. Таким образом, надёжная коммуникация возможна только в том случае, когда обе стороны могут убедиться в том, что используют открытые ключи друг друга, а не злоумышленника.

#### Атомарность транзакций

В этом смысле под транзакцией понимается неделимая с точки зрения воздействия на БД последовательность операторов манипулирования данными (чтения, удаления, вставки, модификации), такая, что либо результаты всех операторов, входящих в транзакцию, отображаются в состоянии базы данных, либо воздействие всех этих операторов полностью отсутствует.

Лозунгом транзакции является «Все или ничего»: при завершении транзакции оператором COMMIT(высокоуровневый аналог операции END TRANSACTION в интерфейсе RSS, см. лекцию 12) результаты гарантированно фиксируются во внешней памяти (смысл термина *commit* состоит в запросе «фиксации» результатов транзакции); при завершении транзакции оператором ROLLBACK (высокоуровневый аналог операции RESTORE в интерфейсе RSS, см. лекцию 12) результаты гарантированно отсутствуют во внешней памяти (смысл термина *rollback* состоит в запросе ликвидации результатов транзакции).

**Непреднамеренные ошибки** возникают независимо от желания лиц, сообщающих или регистрируют данные.

Непреднамеренные ошибки регистрации могут иметь случайный или систематический характер

**Случайные непреднамеренные ошибки** регистрации - это ошибки, возникающие вследствие различных случайных причин: описка, оговорка и т др. Они приводят к отклонениям данных наблюдения от фактических размеров признаки с одинаковой вероятностью ю как в сторону увеличения, так и в сторону уменьшения данных При достаточно большом количестве единиц наблюдений случайные ошибки могут взаимно погашаться и не производить существенного влияния на результаты видеонаблюдения.

**Систематические непреднамеренные ошибки** регистрации возникают из определенных неслучайных причин и приводят к отклонениям данных наблюдения от фактических размеров признаки в сторону увеличения или уменьшения Причиной таких ошибок может быть не исправление измерительных приборов, нечеткая формулировка вопросов, несовершенство статистического инструментария, склонность людей к округлению цифр.