Атаки, проблемы и методы защиты

Завоевание доверия

Условия

- A находится рядом с C;
- C, D - атакующие узлы в разных сегментах;

Описание

Узел может нарабатывать себе хороший рейтинг, имитируя легитимное поведение с помощью обмена пакетами с неизвестным узлу A узлом D. Став приоритетным методом связи за счет рейтинга для A, сможет контролировать возможность передачи его пакетов.

Решения

- Дожидаться на A подтверждения получения отправленных пакетов (включая подтверждение корректности);
- Считать рейтинг C исключительно на основе успешно переданных им пакетов от узла A.

Узлы "эгоисты"

Условия

- A хочет передать пакет B;
- Узел C находится на маршруте между ними;

Описание

В целях экономии трафика/снижения нагрузки/пр. или целенаправленного вреда узел C может принимать пакет и не передавать его далее по сети, не сообщая об этом. Таких узлов может быть множество.

Решения

- Понижать рейтинг за "неперадачу";
- Передавать сообщения от сетьузлов с низким рейтингом с пониженным приоритетом.

Атаки Сивиллы

Узлами рандеву

Условия

- RN - узел атакующего, ложно заявляющий о себе как об узле "рандеву";
- CertRN - сертификат узла RN.

Описание

Атакующий может создать множество узлов рандеву и некорректно (либо совсем не-) распределять подключающихся пользователей.

Решения

- Подписывать хорошие узлы CertRN, сертификатами других узлов рандеву, в особенности, "корневым" сертификатом разработчика, образуя сеть доверия;
- При подключении пользователь смотрит на уровень доверия CertRN и принимает решение о продолжении работы.

Узлами пользователей

Условия

- (D, CertD) - новый пользователь атакующих с новыми сертификатами;
- Узел "рандеву" RN с сертификатом CertRN контролирует сегмент сети;

Описание

Атакующие могут создать большое число узлов и заполонить сеть, входя через RN, централизовано "ломая" впоследствии процесс обмена данными с помощью своего доминирующего количества.

Решения

- При легитимном поведении узла A, RN может подписывать CertA своим CertRN, помечая доверенными;
- Определять атаку за счет счетчика зарегистрированных узлов Count, доверенных TrustedCount и отношения TrustedCount / Count;
- При определении проводящейся атаки выгнать CertD, не являющийся доверенным, и не пускать новые не доверенные узлы.

Спам на узел

Условия

- A выложила в сеть свой идентификатор для связи;
- Атакующий желает добиться отказа ее устройства/ПО;

Описание

Так как узлы не имеют ограничений на отправление пакетов другим участника сети, а регистрация в ней ничего не стоит, спамер может создать множество новых узлов и слать с них пакеты по идентификатору A.

Решения

- Сменить ключевую пару (плохо, предыдущие подписи теряют смысл);
- Использовать временные ID, чтобы не менять ключевую пару (хорошо, описание ниже).

Протоколы

Временный ID

Используется для организации связи с отдельным лицом/группой лиц, не позволяя выходить на контакт просто по открытому ключу.

Выработка

Алиса хочет дать Бобу свой идентификатор.

1. A: генерирует или получает X - случайный идентификатор сессии;
2. A -> B: Temp_ID = hash(PubA, X), X.

Проверка

1. Получить PubA' ;
2. hash(PubA', X) == Temp_ID ? OK : FAILURE.

Остальное использование полностью аналогично обычным идентификаторам. Для смены просто генерируется новый X, возможно, с помощью уже установленного защищенного контакта через сеть.

Обмен ключами

Публичным

Алиса и Боб обменялись идентификаторами ID_A и ID_B соответственно за пределами сети.

1. A -> B: PubA;
2. B: hash(PubA, X) == ID_A ? OK : FAILURE;

   B -> A: PubB;
3. A: hash(PubB, Y) == ID_B ? OK : FAILURE.

Сеансовым

Диффи-Хеллман

Плюс - не требуется передача данных по каналу.

Минус - для каждой пары ключей способен создать лишь один общий сеансовый. Решается использованием его в качестве источника энтропии в детерминированном ГСЧП, но порождает возможность атаки последнего и сложности по поддержанию общего состояния.

Использование открытого ключа

Алиса хочет связаться с Бобом, имея его публичный ключ PubB.

1. A: K - случайный сеансовый ключ;

   A -> B: Ck = encrypt_asym(K, PubB), Sk = sign(K, PrivA);
2. B: K' = decrypt_asym(Ck, PrivB), check_sign(K', Sk, PubA) ? OK : FAILURE.

Вместо использования подписи можно использовать расширение протокола:

3. X - случайное число;

   B -> A: Cx = encrypt_sym(X, K);

4. A -> B: X' = decrypt_sym(Cx, K);

5. B: X' == X ? OK : FAILURE.

Плюс - возможность генерировать по необходимости любое число сеансовых ключей.

Минусы - необходимость использовать канал данных для совершения как минимум одной передачи данных, требуются отдельные функции асимметричного шифрования.

Криптоалгоритмы

Эллиптические кривые

Стандарты

- ECDSA - американский стандарт подписи, широко распространен;
- ГОСТ 34.10-2012 - отечественный стандарт подписи.

Оба стандарта позволяют как подпись, так и шифрование. Выбор эллиптической кривой отходит либо пользователю, либо берется готовая надежная из известных публикаций и стандартов, но должен соответствовать требованиям стандарта (все есть в вики).

Генерация ключей

Для генерации собирается энтропия с надежного источника (внутренний генератор процессора, внешние генераторы типа random.org) и действий пользователя (движения мышью и пр.). Полученные случайные данные используются для генерации.

В качестве защиты от перебора можно использовать класс функций KDF (PBKDF2, scrypt, etc).

Симметричное блочное

Предпочтительно использовать надежные (выбор неочевиден) шифры с длиной ключа 256 бит и более. Генерация ключа симметричного шифрования точно так же требует источник энтропии и порождающую функцию KDF.

При шифровании сообщений из нескольких блоков нужно использовать режим CBC. Если шифруется один неполный блок нужно добавить в свободное пространство случайных данных для усложнения перебора.

Подобную технику можно использовать для всех блоков в принципе, выделив в их структуре фиксированные поля на шум.