АБСОЛЮТНО НЕВИДИМОЕ
сканирование портов с поддельным IP-адресом

Алексей Волков, © 2003

Основные вопросы:

- Техника TCP ip - хэндшейкинга
- Техника традиционных методов сканирования
- ID-последовательности и их предсказание
- Сканирование с подменой IP в теории и на практике
- Исходный код и примеры
- Демонстрация
- Ответы на вопросы

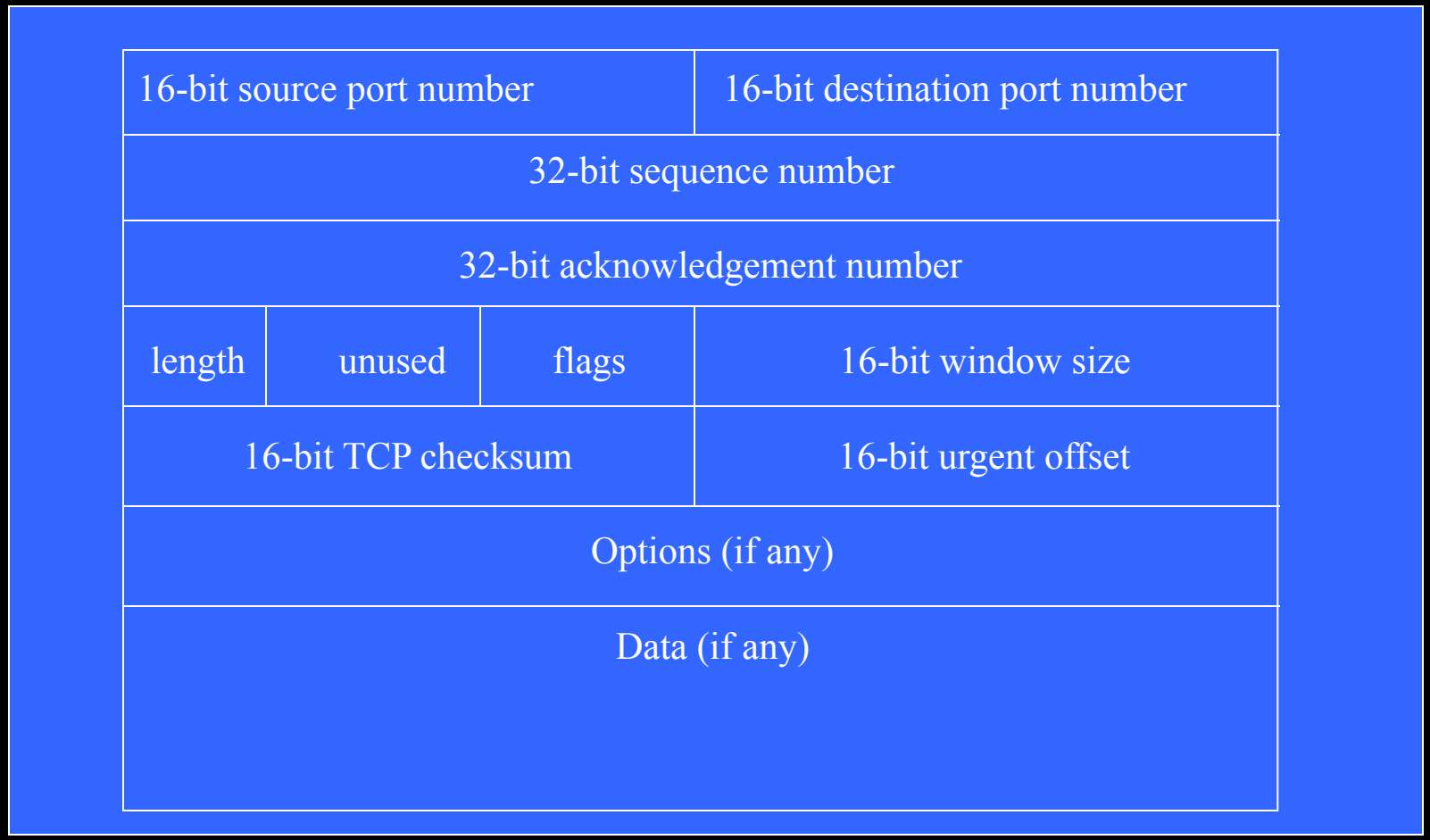
Техника TCP - хэндшейкинга

- Определения
- Заголовок Tcp
- Традиционная схема 3-х стороннего хэндшейкинга

Определения

- Открытое соединение между двумя компьютерами в протоколе TCP/IP называется сокетом и определяется:
- IP-адресом источника
- Номером порта источника
- IP-адресом приемника
- Номером порта приемника
- Начальным значением SEQ источника
- Начальным значением SEQ приемника
- И номером ID который увеличивается с каждым переданным пакетом

Заголовок TCP пакета



TCP/IP-хэндшейкинг

Src ip,Dst ip

Src prt, Dst Prt

Syn = in seq#

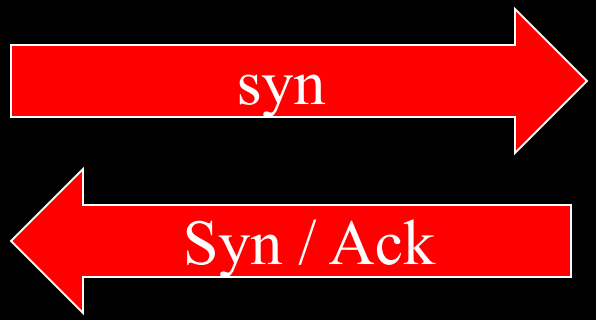
Ack = NULL

Flags = S

Src ID = src ID + 1

TCP/IP - хэндшейкинг

Src ip,Dst ip
Src prt, Dst Prt
Syn = src seq#
Ack = NULL
Flags = S



Src ip,Dst ip
Src prt, Dst Prt
Syn = Dst seq#
Ack = src seq# +1
Flags =
Dst ID = Dst ID + 1

TCP/IP-хэндшейкинг

Src ip,Dst ip

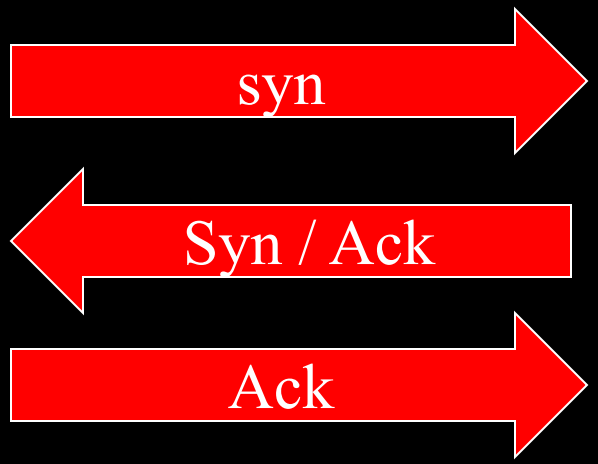
Src prt, Dst Prt

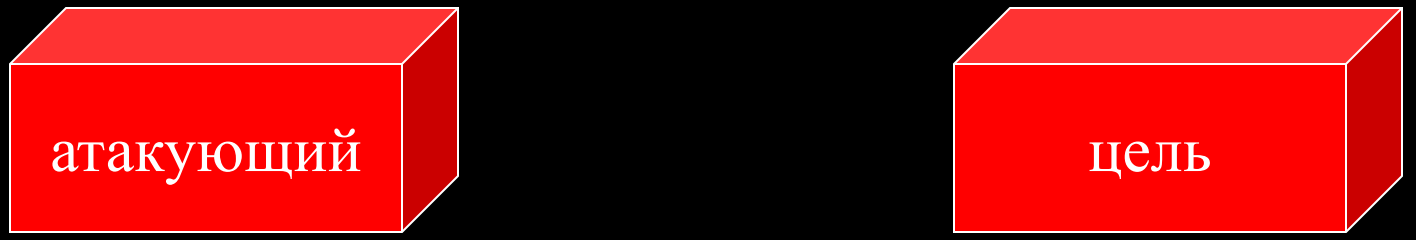
Syn = src seq#

Ack = dst seq# +1

Flags = A

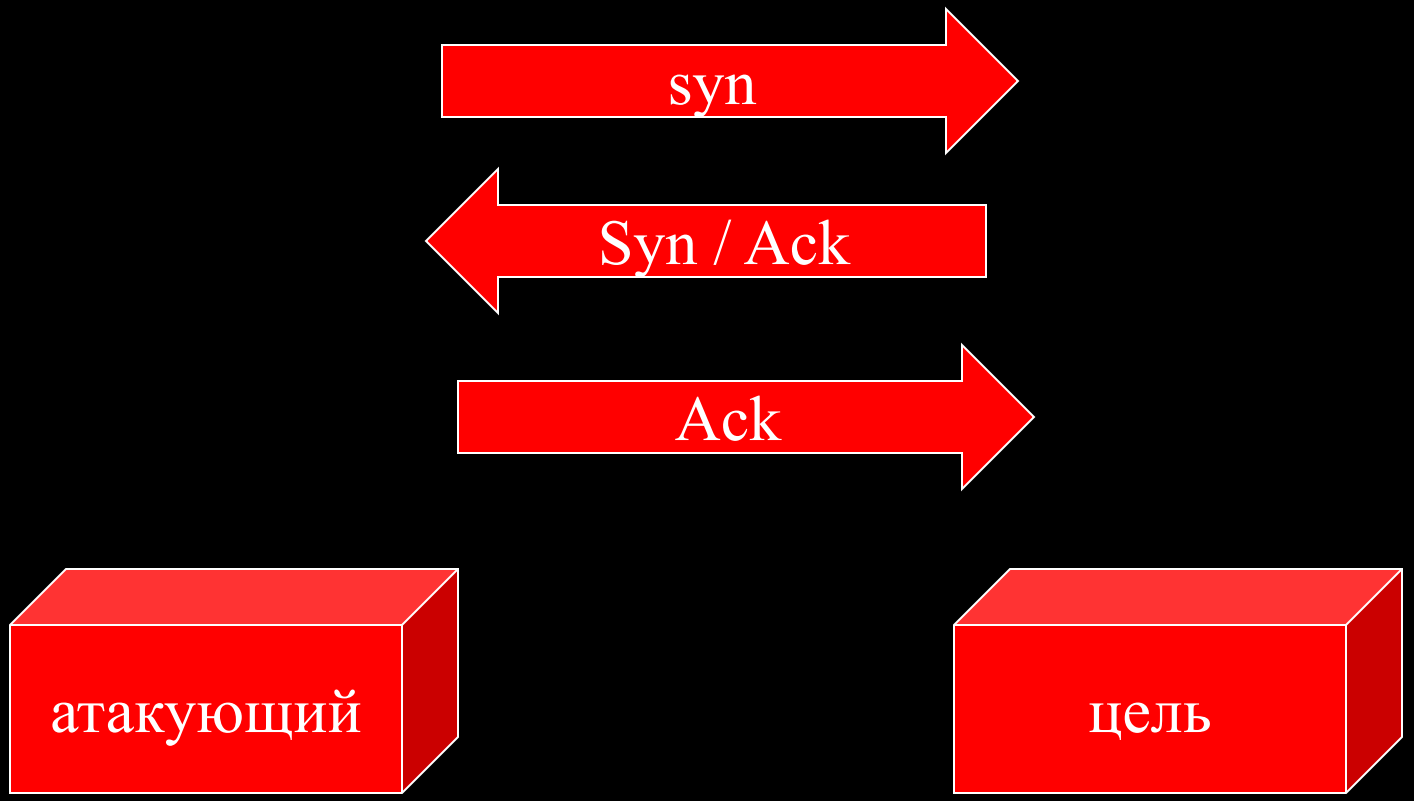
Src ID = src ID + 1



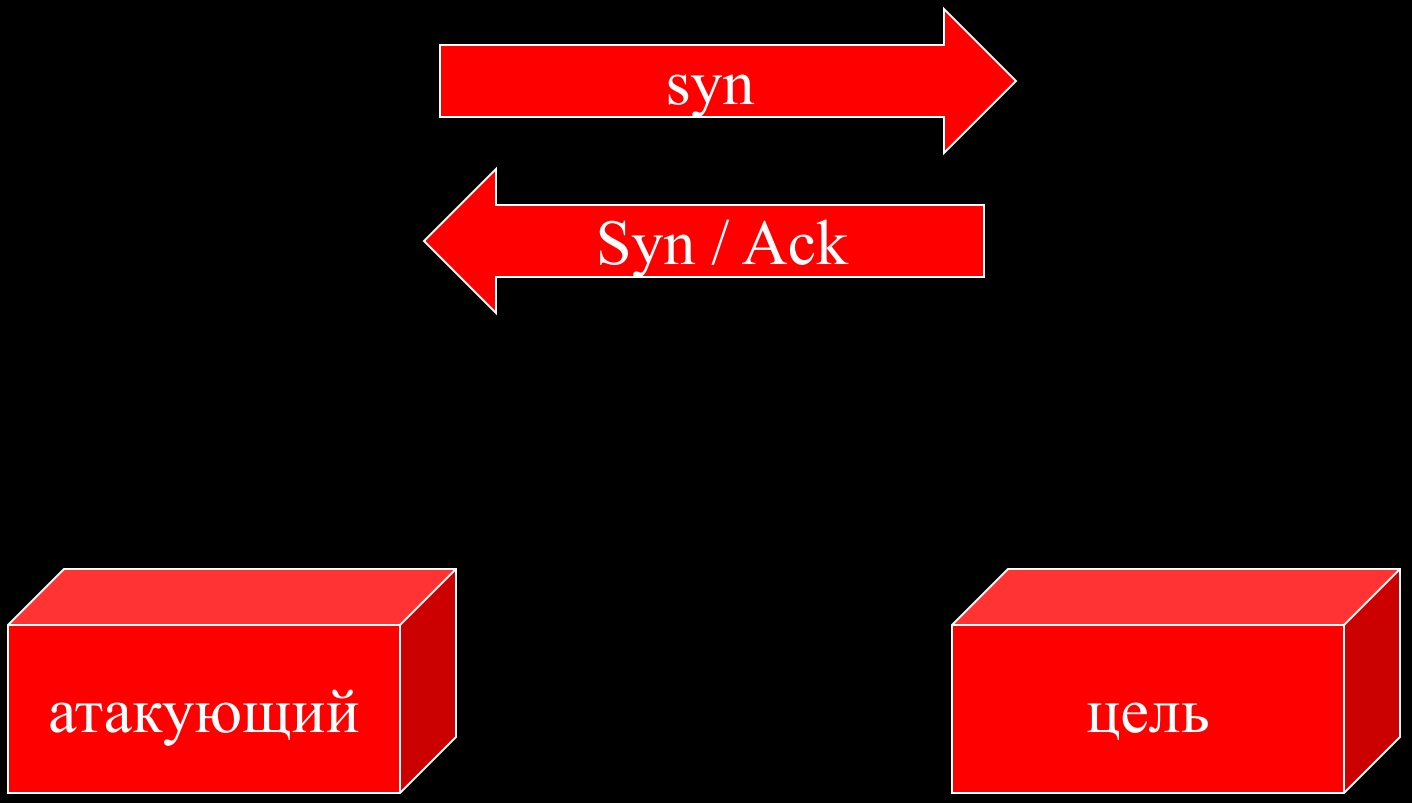


Установление сокета

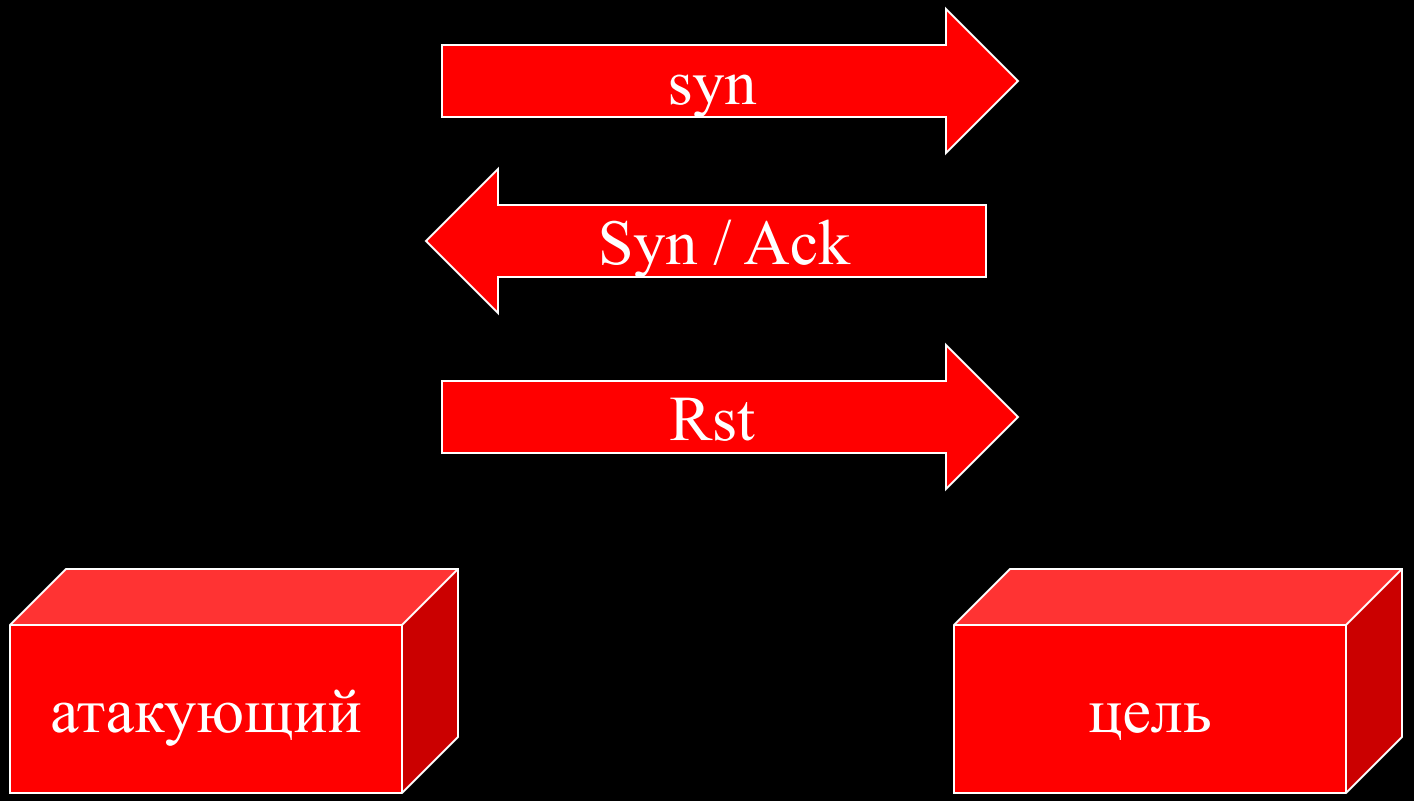
Традиционное сканирование



Невидимое SYN-сканирование



Невидимое SYN-сканирование 2



Что такое IPID?

Специальным образом сгенерированная последовательность, число, характеризующее номер пакета. Используется для упрощения процедуры «сборки» пакетов на стороне приемника после фрагментации.

Увеличивается с каждым посланным пакетом.

Существуют различные схемы увеличения этого номера.

Что такое ID – флаг ?

- Идентифицирует каждую TCP-сессию.
- В некоторых случаях используется для «сборки» пакетов
- Счетчик ID увеличивается с каждым посланным пакетом
- Его содержат все пакеты, включая RST

Предсказание значения ID-флага

- Большинство UNIX-систем дают случайное или псевдо-случайное приращение этому счетчику.
- До сегодняшнего дня не было случая, чтобы ID-флаг представлял угрозу с точки зрения безопасности.
- Windows 95 увеличивает id# на 1
- Windows 2000 увеличивает id# на 254
- Вот почему в этих ОС применяется обратный порядок бит в id#.

«Поддельное» сканирование в теории

- Постоянно опрашивая ложный хост на предмет увеличения его id можно увидеть, отправил ли сканируемый хост syn/ack или reset.
- Анализируя это можно определить, какой порт открыт, а какой нет
- На стороне сканируемого хоста эта операция абсолютно невидима.

«Поддельное» сканирование в теории

- Поскольку известно, что Windows увеличивает значение id# при отправке пакета, опрашивая хост, можно определить сколько пакетов он передал между нашими тестами
- Это производится путем мониторинга увеличения ID#

«Поддельное» сканирование в теории

- Если порт открыт, хост или сервер ответит пакетом syn/ack
- Если порт закрыт, в ответ придет rst

«Поддельное» сканирование в теории

Если хост принял syn ack от неизвестного источника, он отправляет в ответ rst

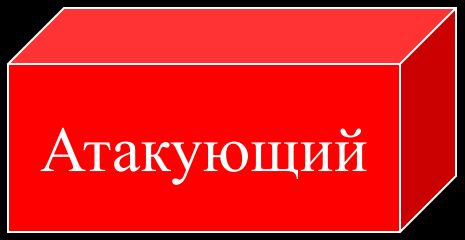
Если хост принял rst пакет от неизвестного источника, он НЕ ОТВЕЧАЕТ на пакет.

«Поддельное» сканирование на практике

Или как все это работает

Представляем участников

172.0.0.1

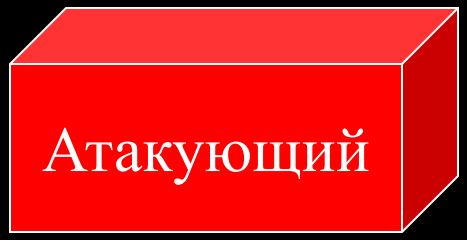


10.0.0.1

192.0.0.1

Зачем трое?

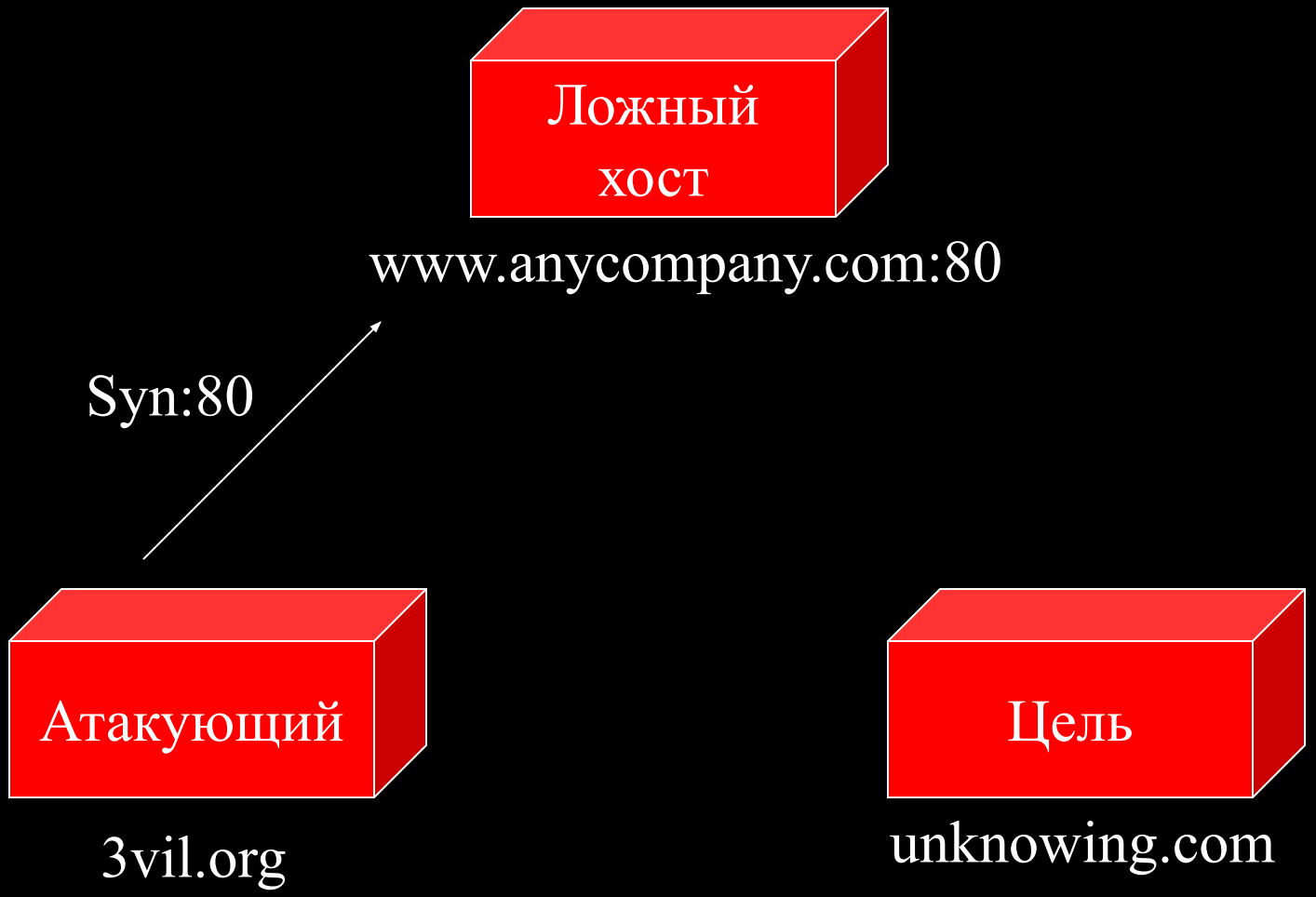
www.anycompany.com:80



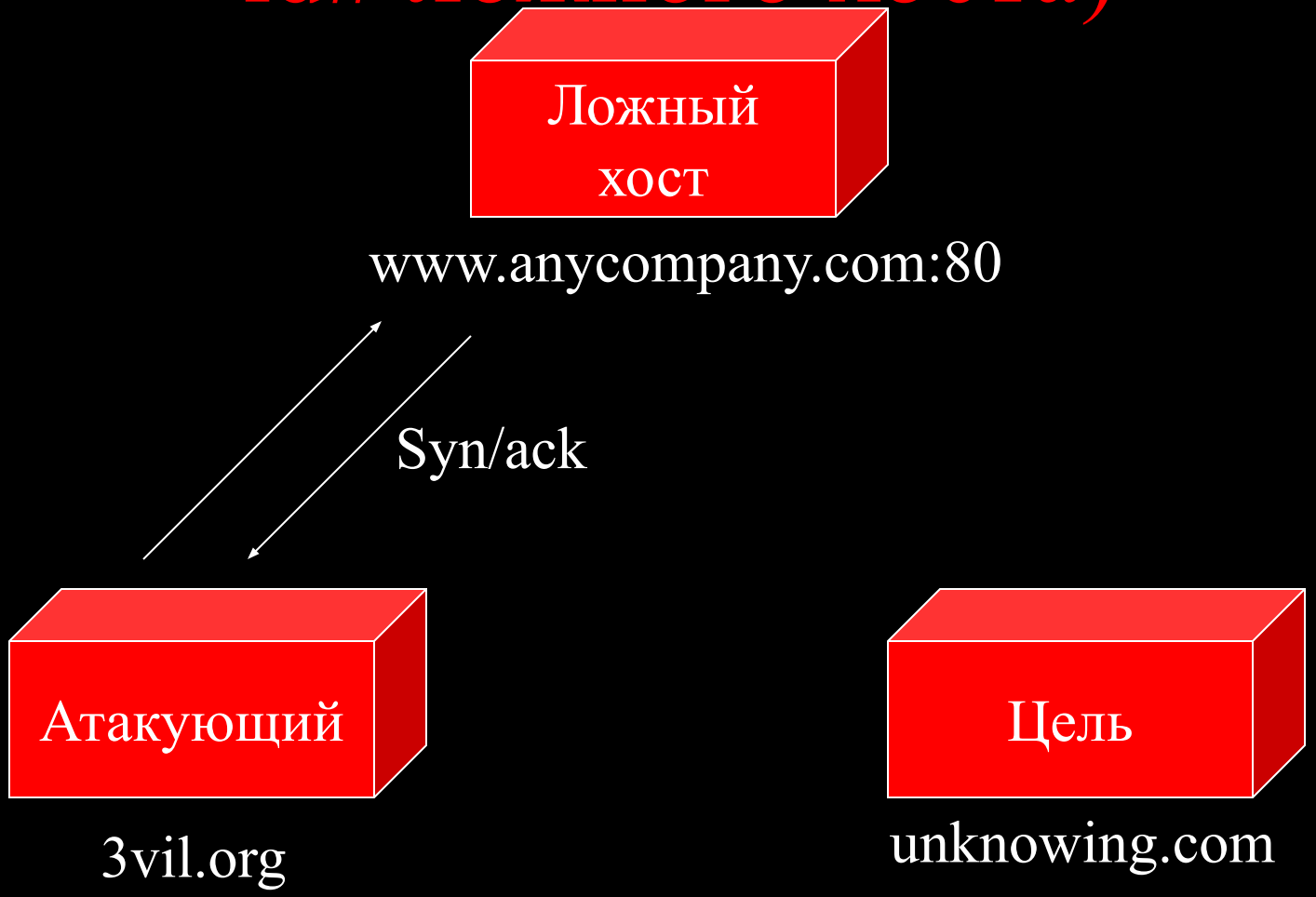
3vil.org

unknowing.com

Первый шаг (синхронизация с id# ложного хоста)



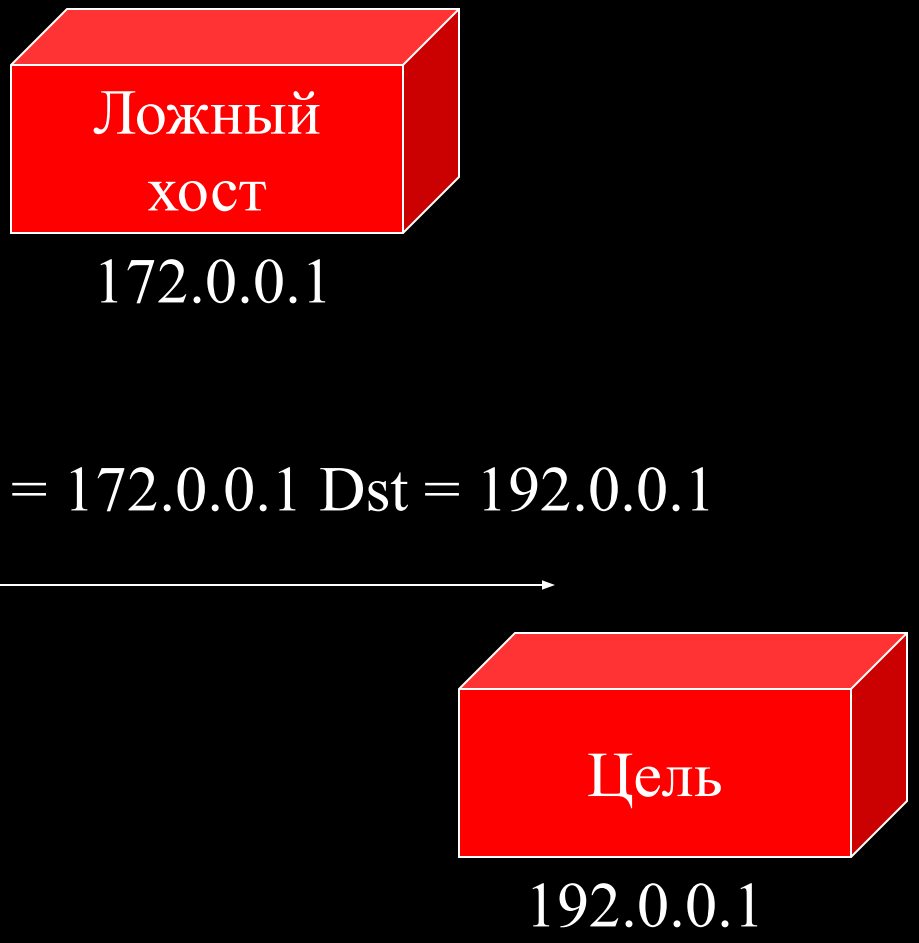
Первый шаг (синхронизация с id# ложного хоста)



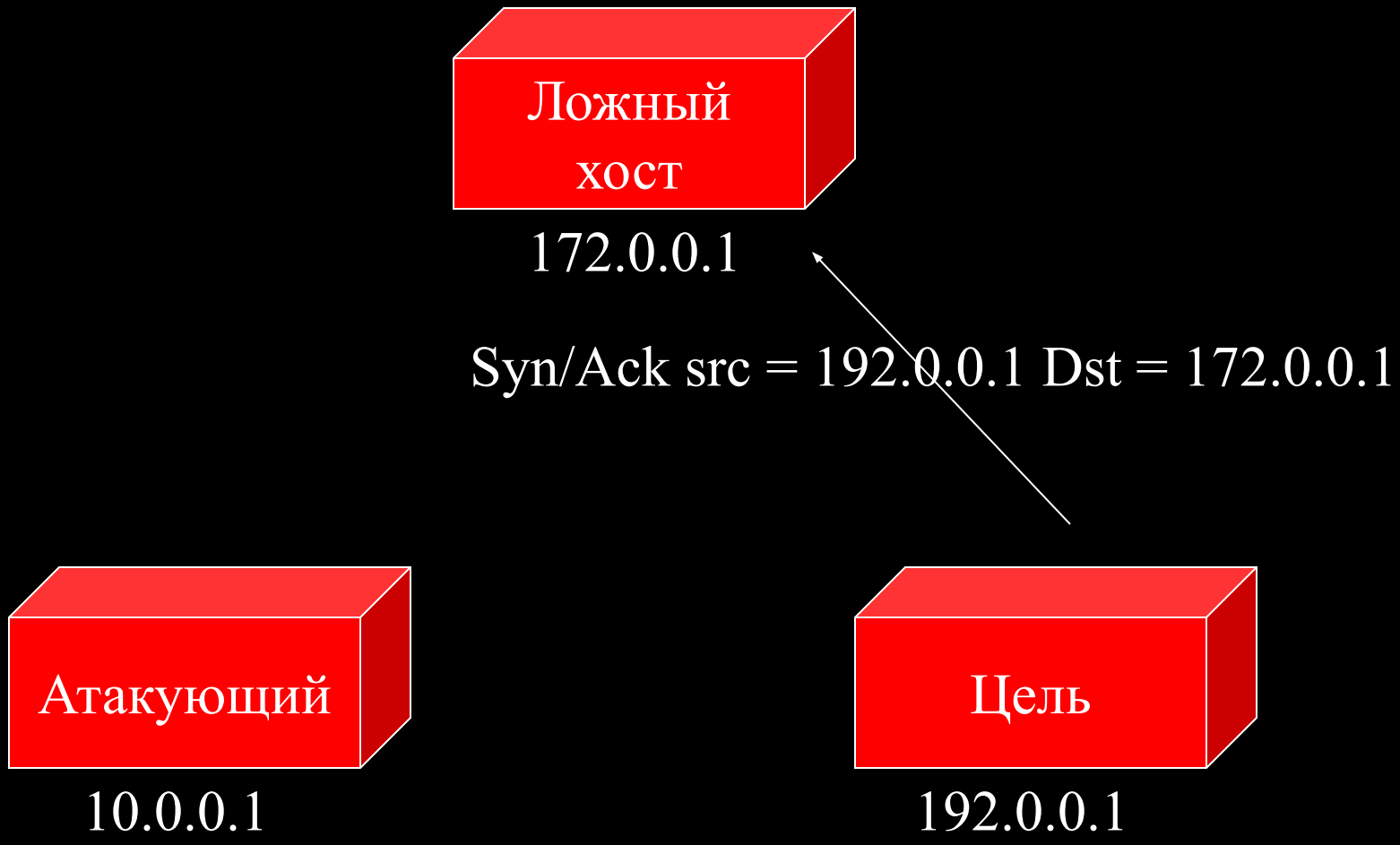
Зачем это надо?

- Теперь атакующему известно начальное значение ID# ложного хоста и характер его изменения

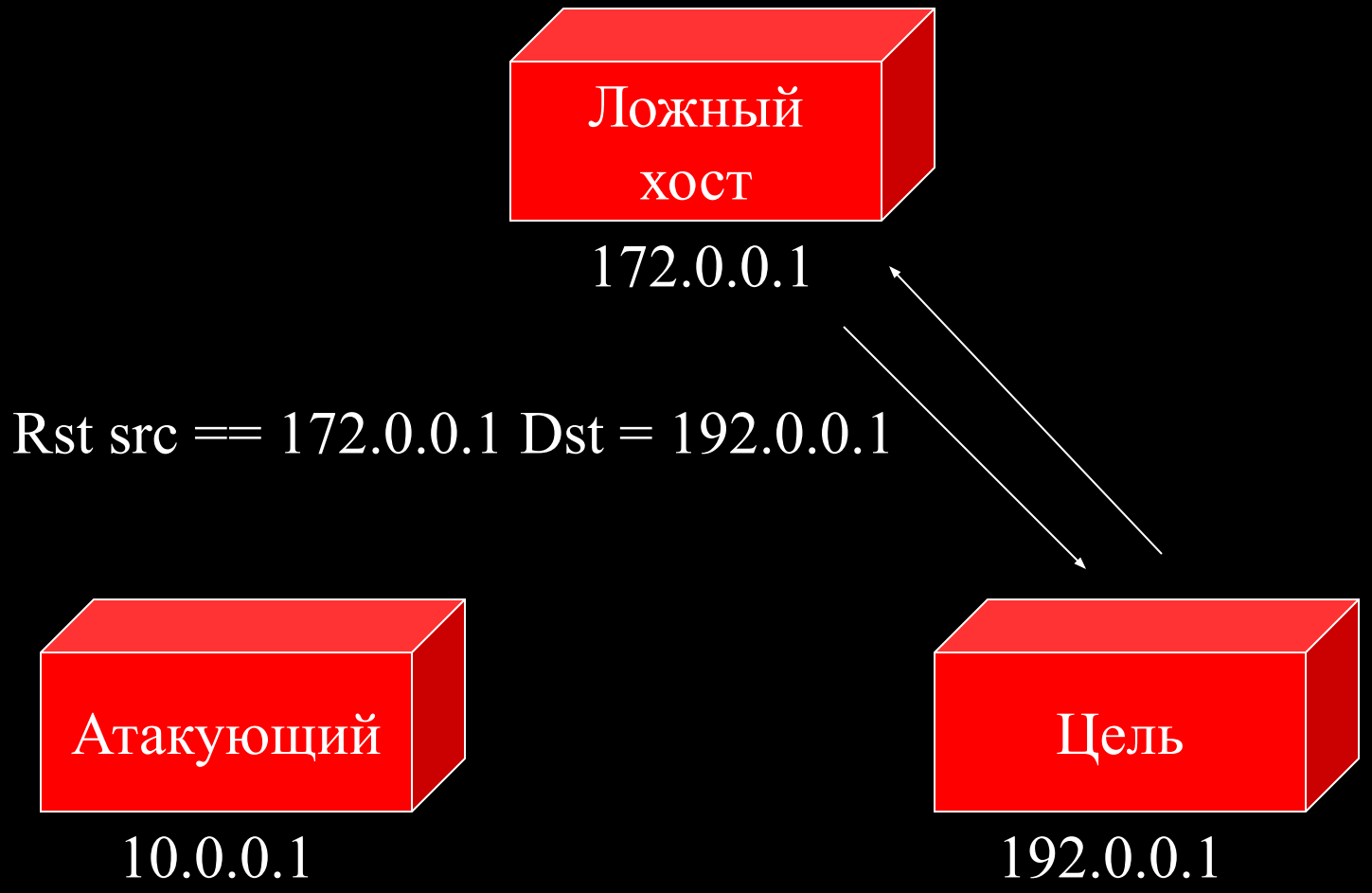
Шаг 2 (подделываем источник)



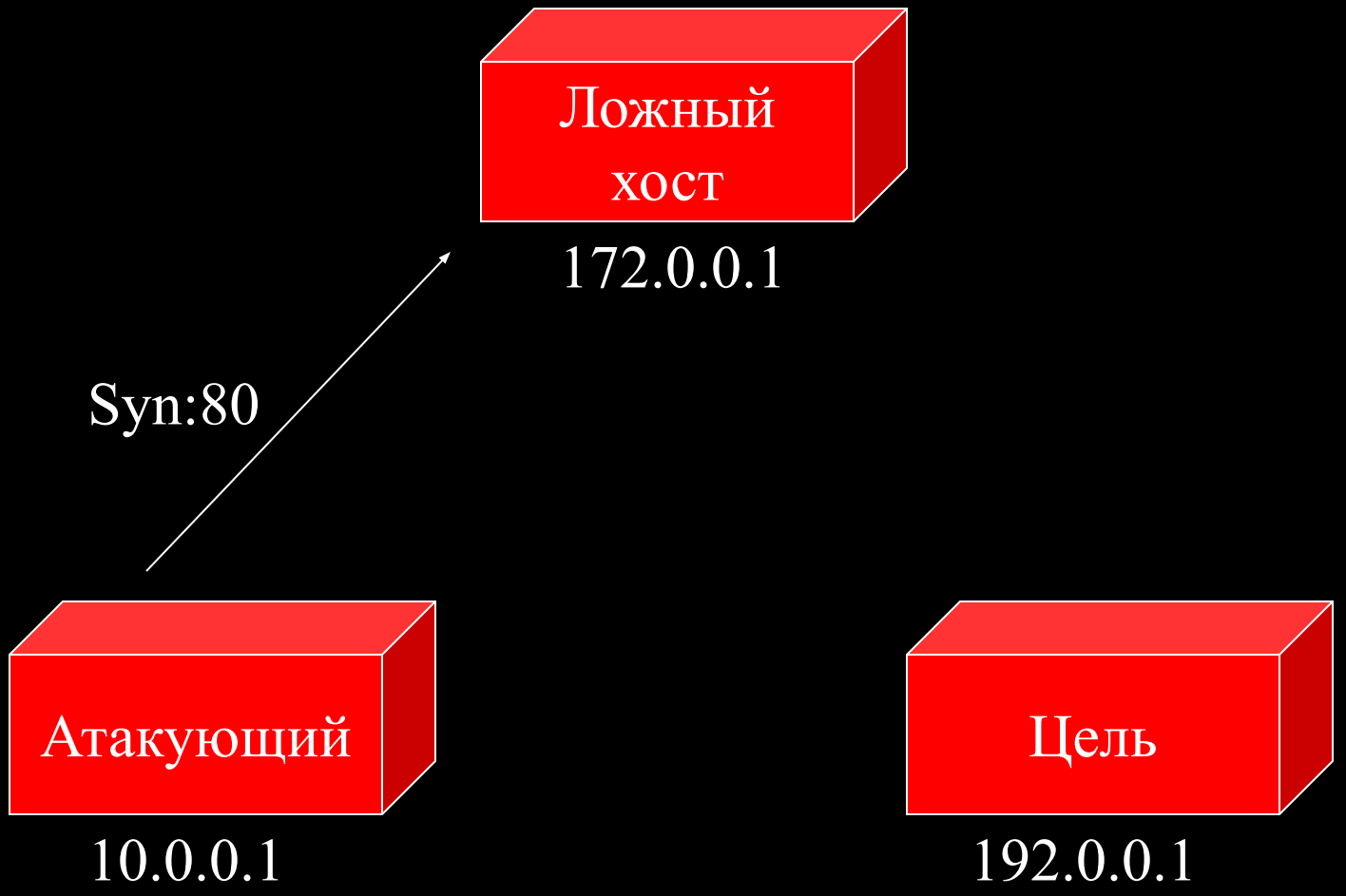
Шаг 3 (сброс ответов)



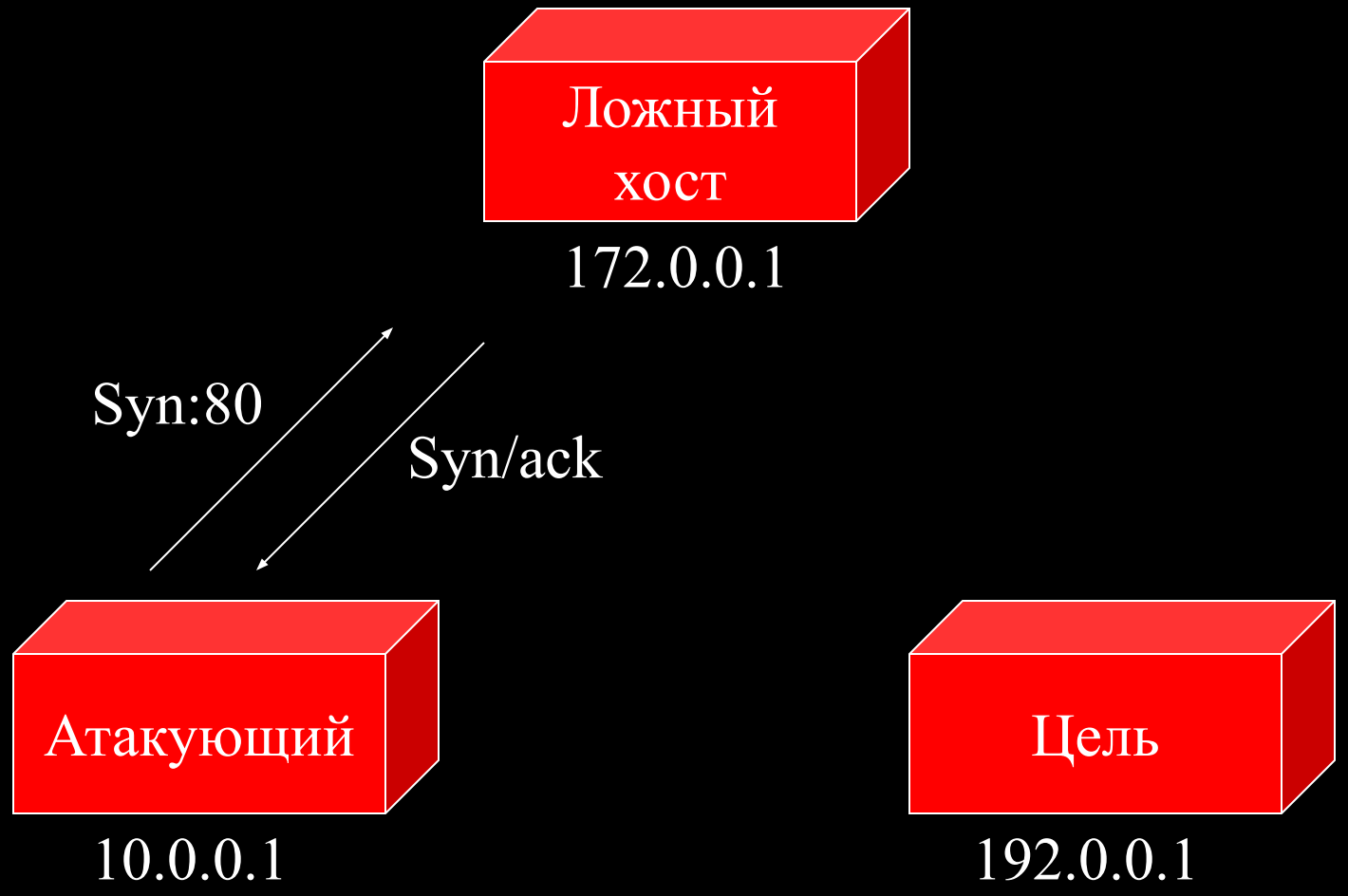
Шаг 3 (сброс ответов)



Шаг 4 (тест ID ложного хоста)



Шаг 4 (тест ID ложного хоста)

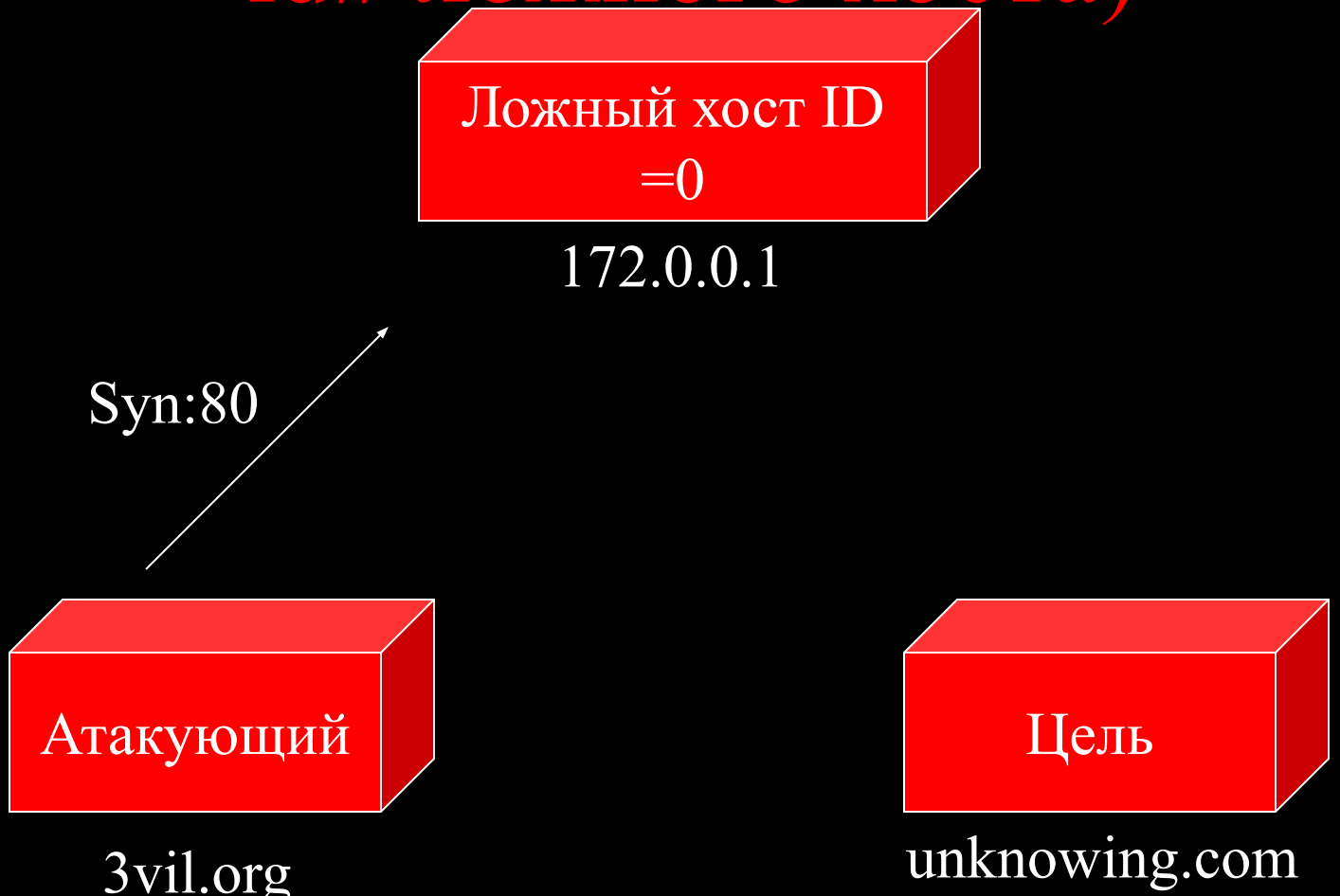


Если порт открыт:

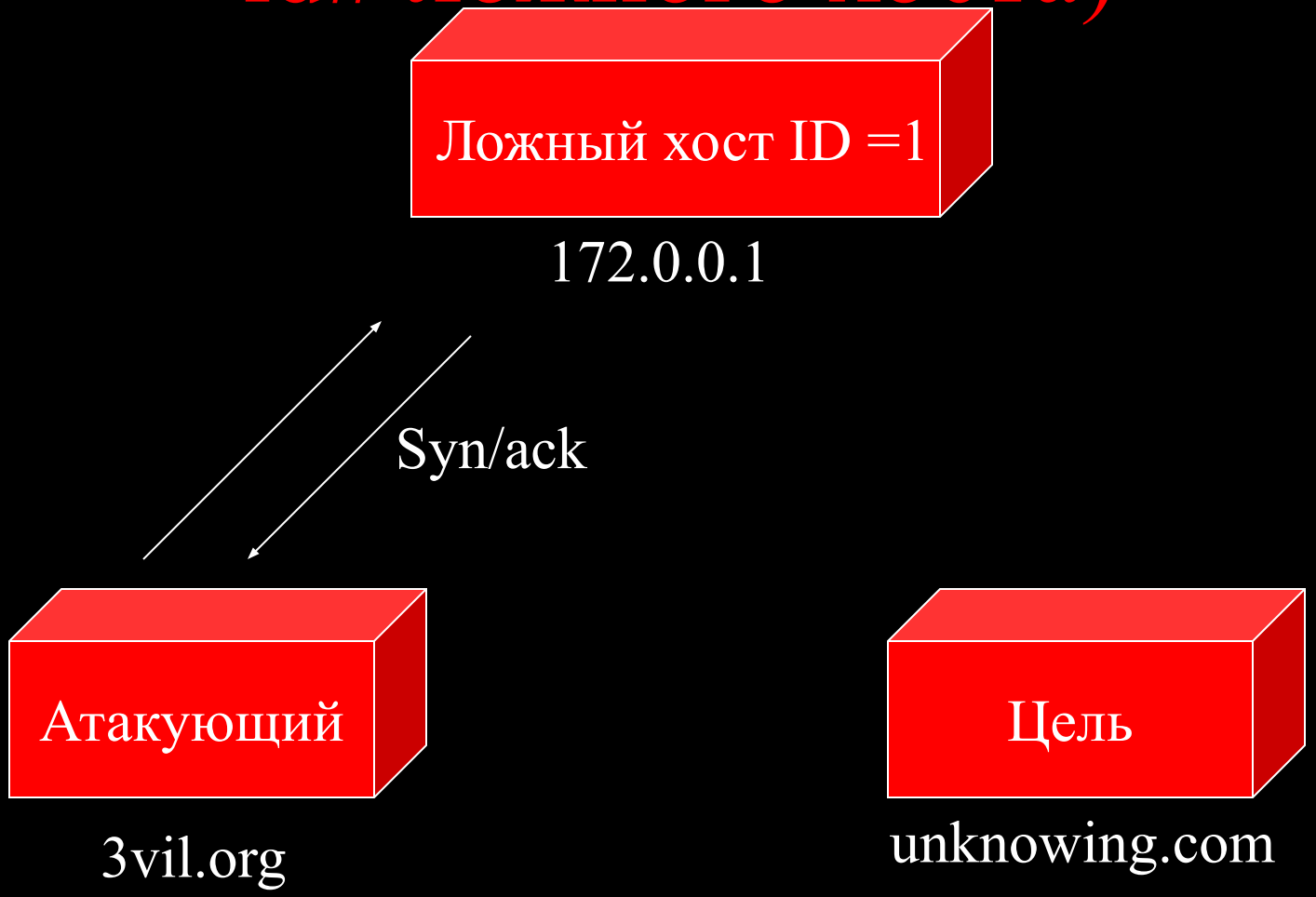
Значение ID будет увеличено.

Смотрим пример:

Первый шаг (синхронизация с id# ложного хоста)



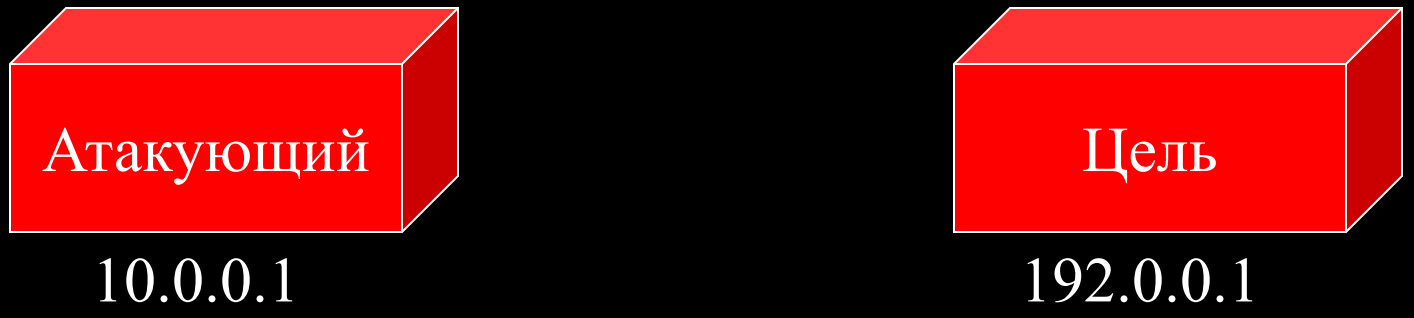
Первый шаг (синхронизация с id# ложного хоста)



Шаг 2 (подделываем источник)

172.0.0.1

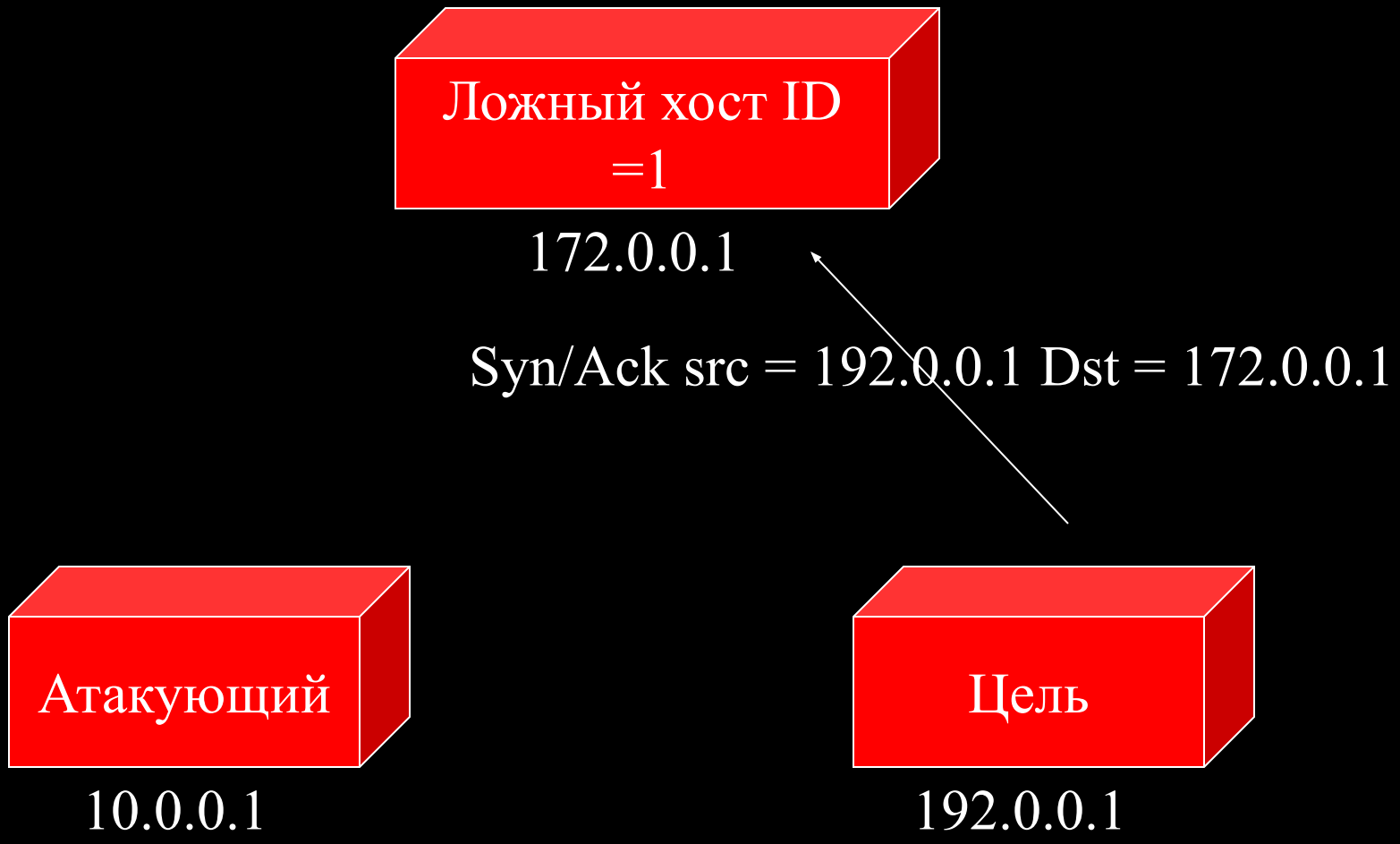
Syn src = 172.0.0.1 Dst = 192.0.0.1



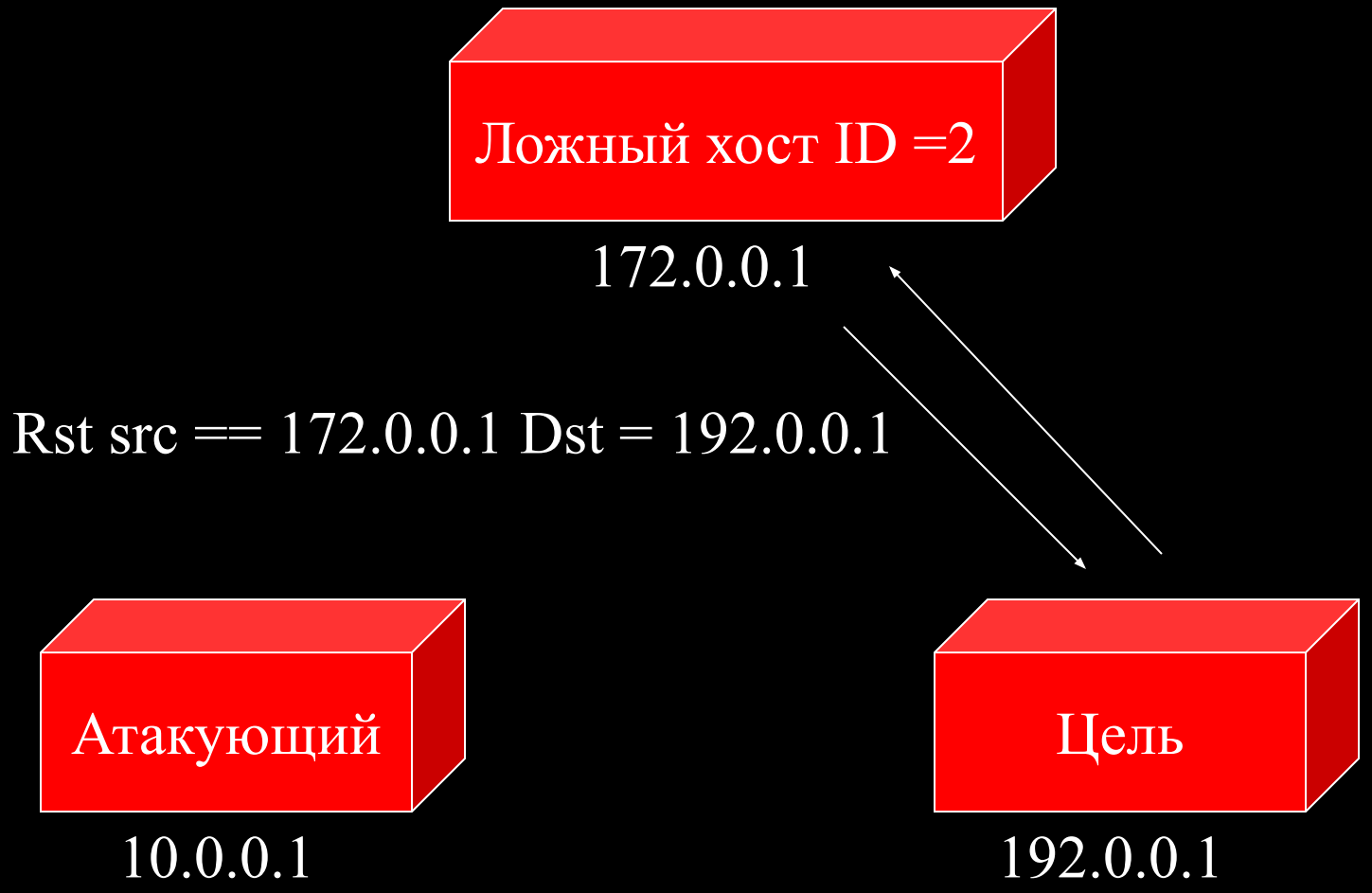
10.0.0.1

192.0.0.1

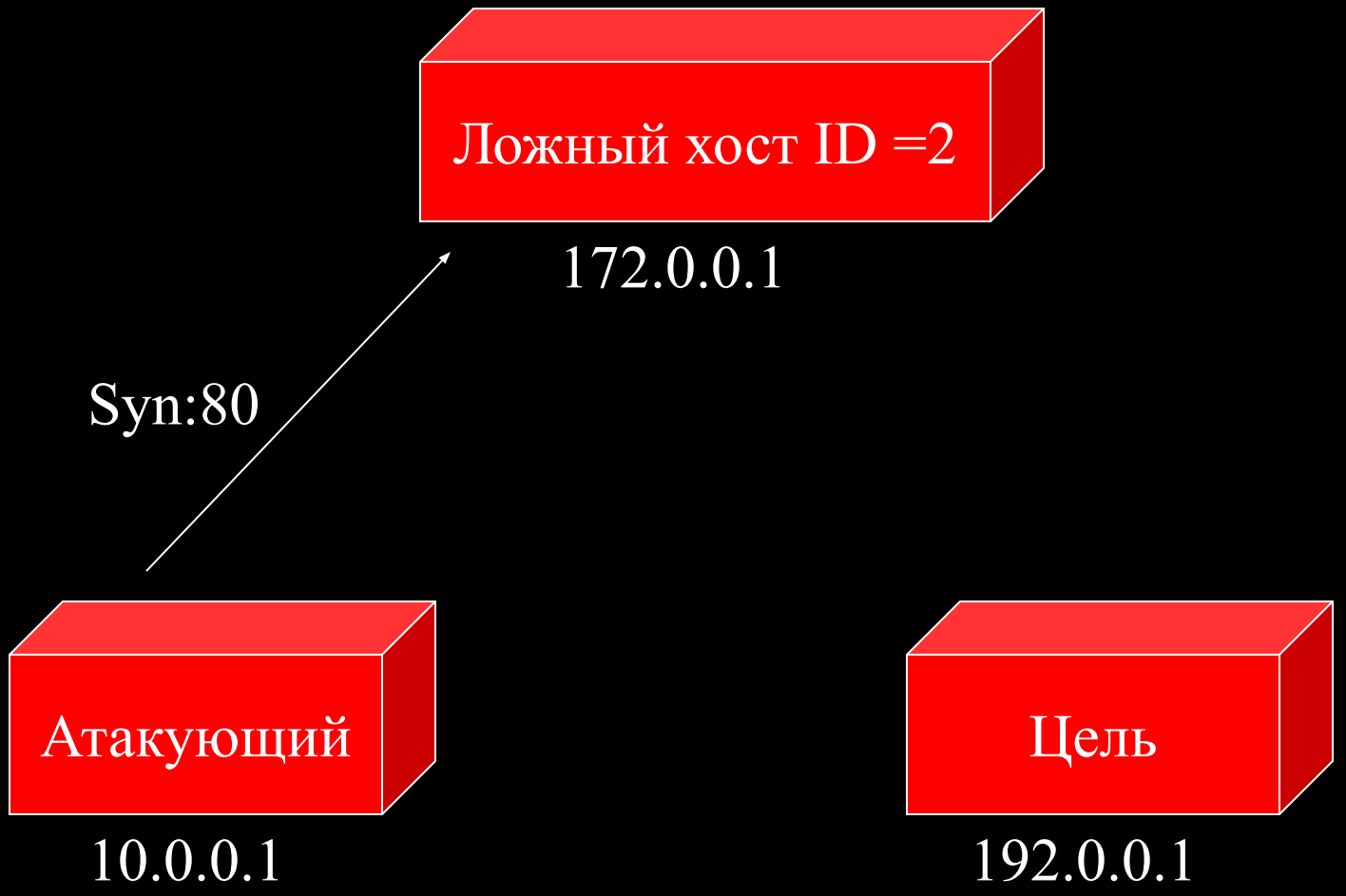
Шаг 3 (сброс ответов)



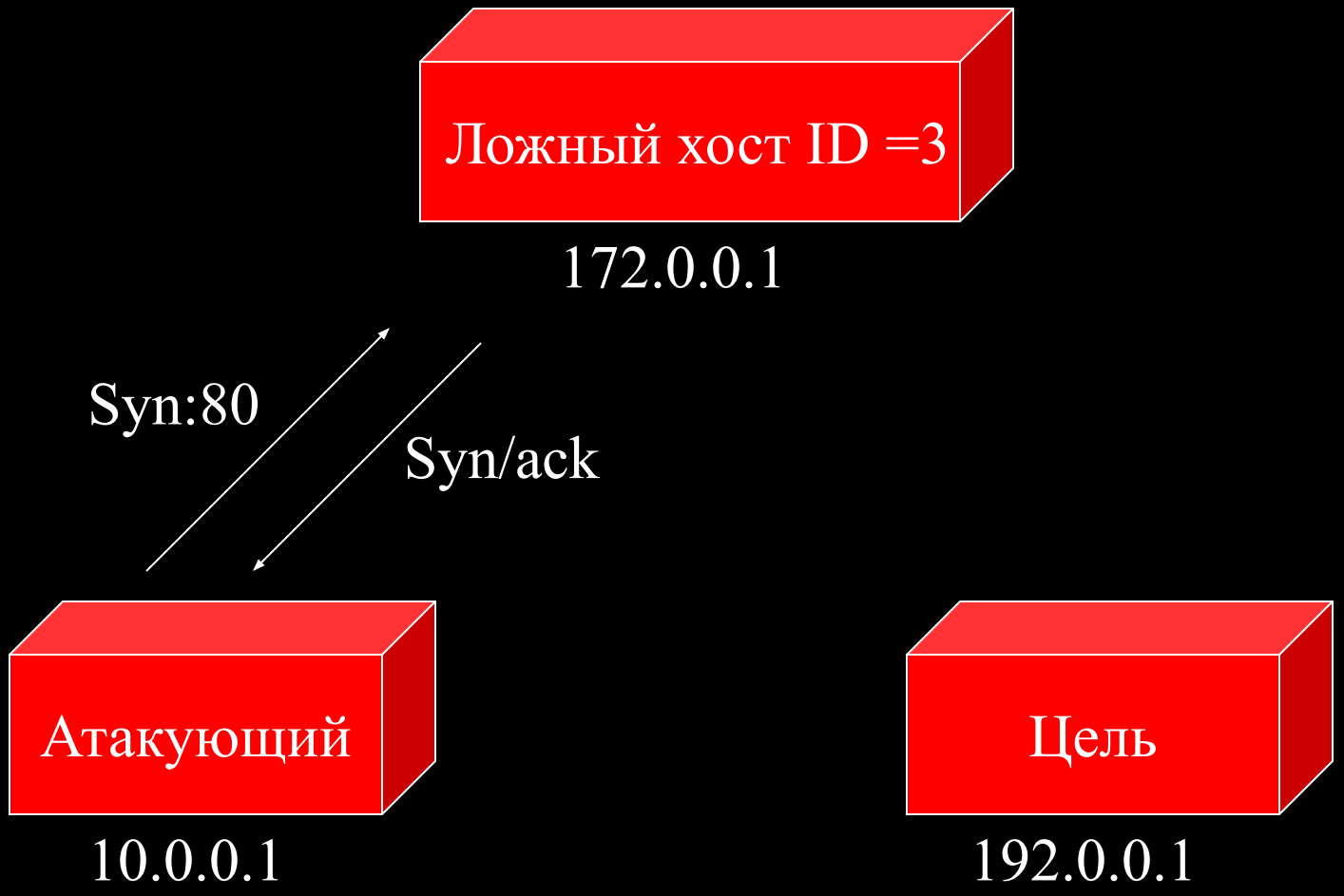
Шаг 3 (сброс ответов)



Шаг 4 (тест ID ложного хоста)



Шаг 4 (тест ID ложного хоста)

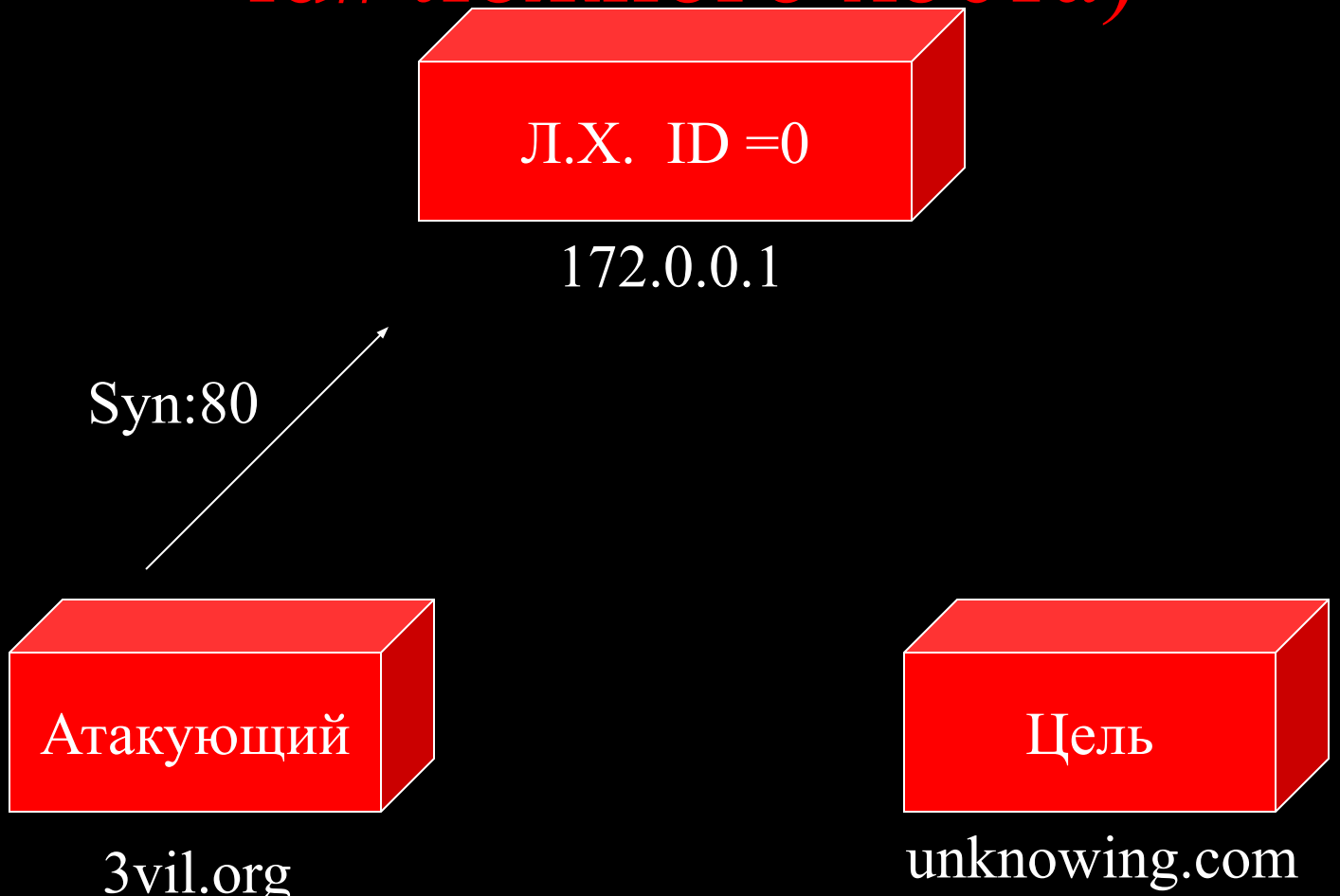


Если порт закрыт:

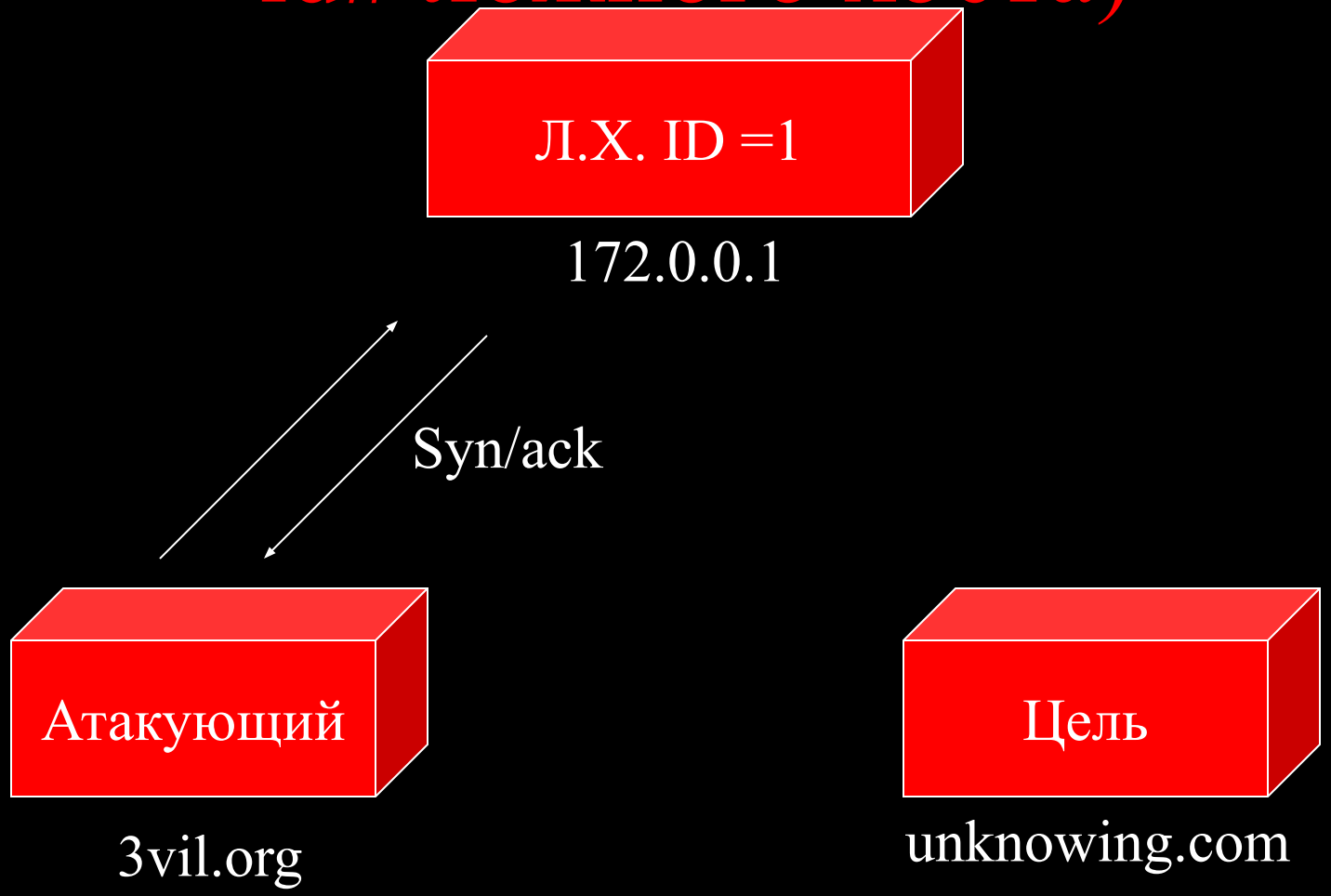
Значение ID не увеличивается.

Смотрим пример:

Первый шаг (синхронизация с id# ложного хоста)



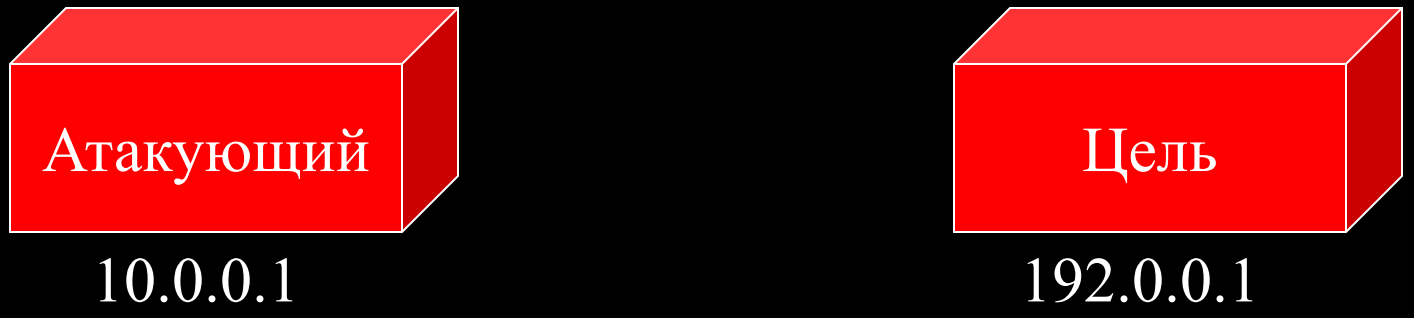
Первый шаг (синхронизация с id# ложного хоста)



Шаг 2 (подделываем источник)

172.0.0.1

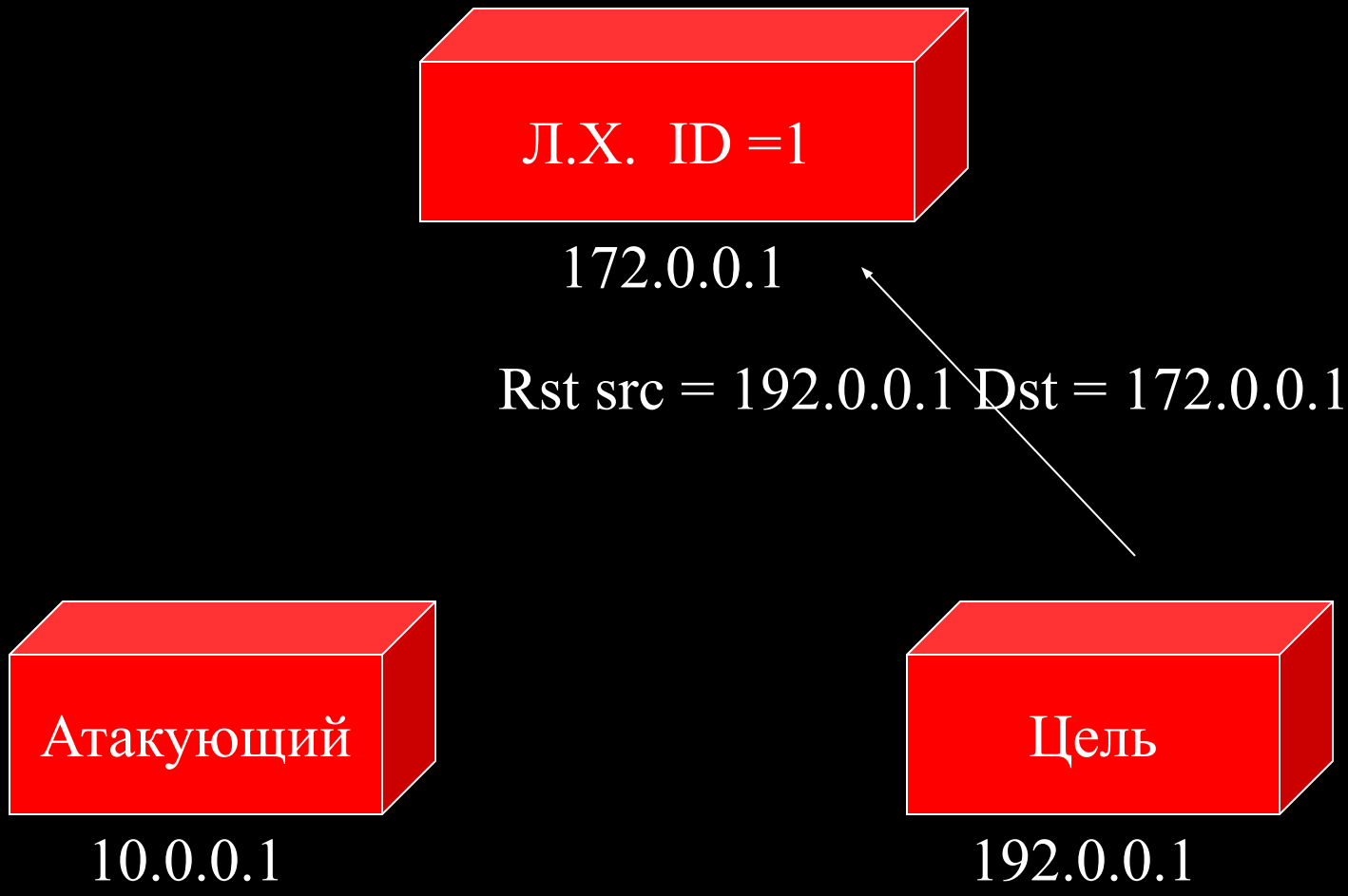
Syn src = 172.0.0.1 Dst = 192.0.0.1



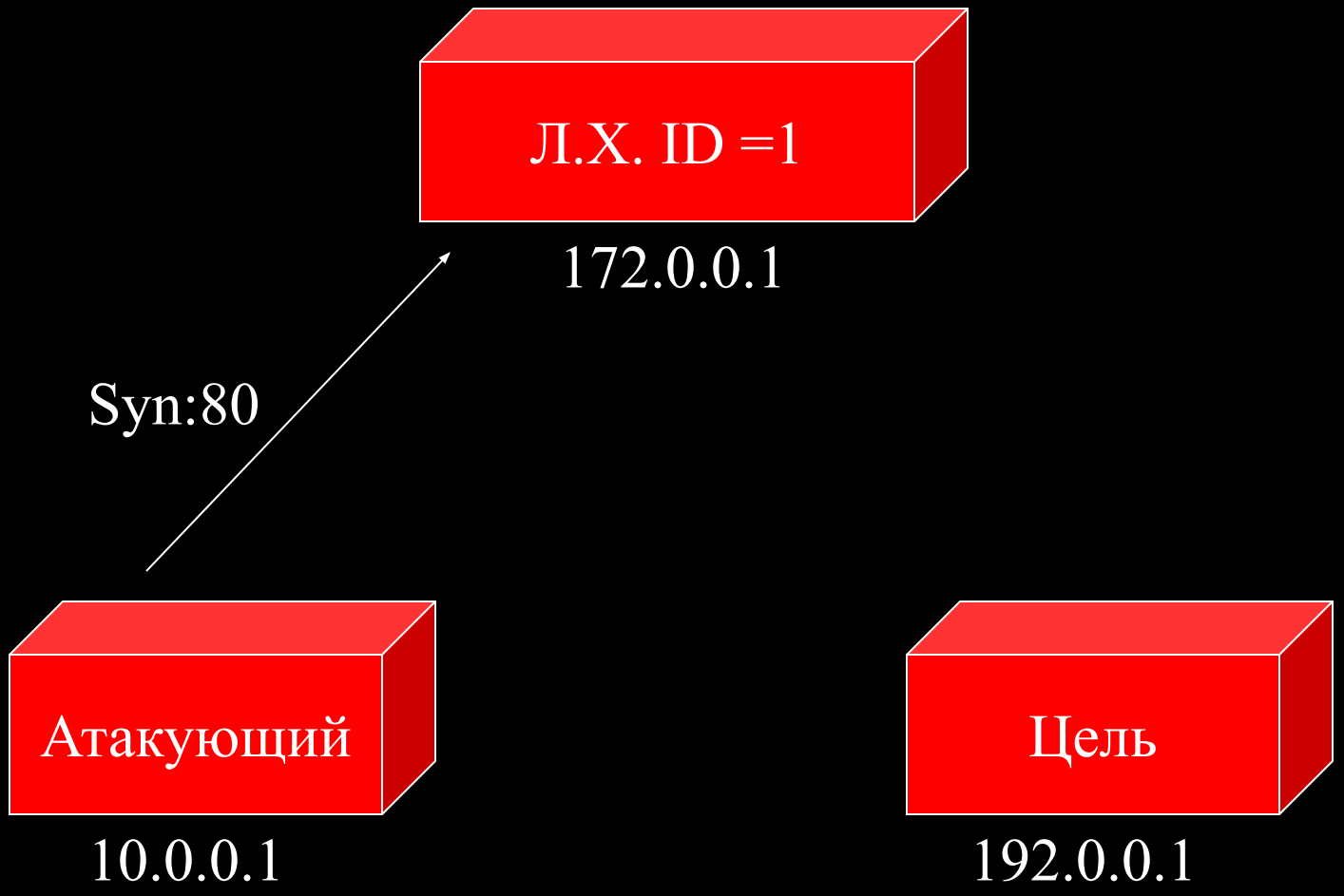
10.0.0.1

192.0.0.1

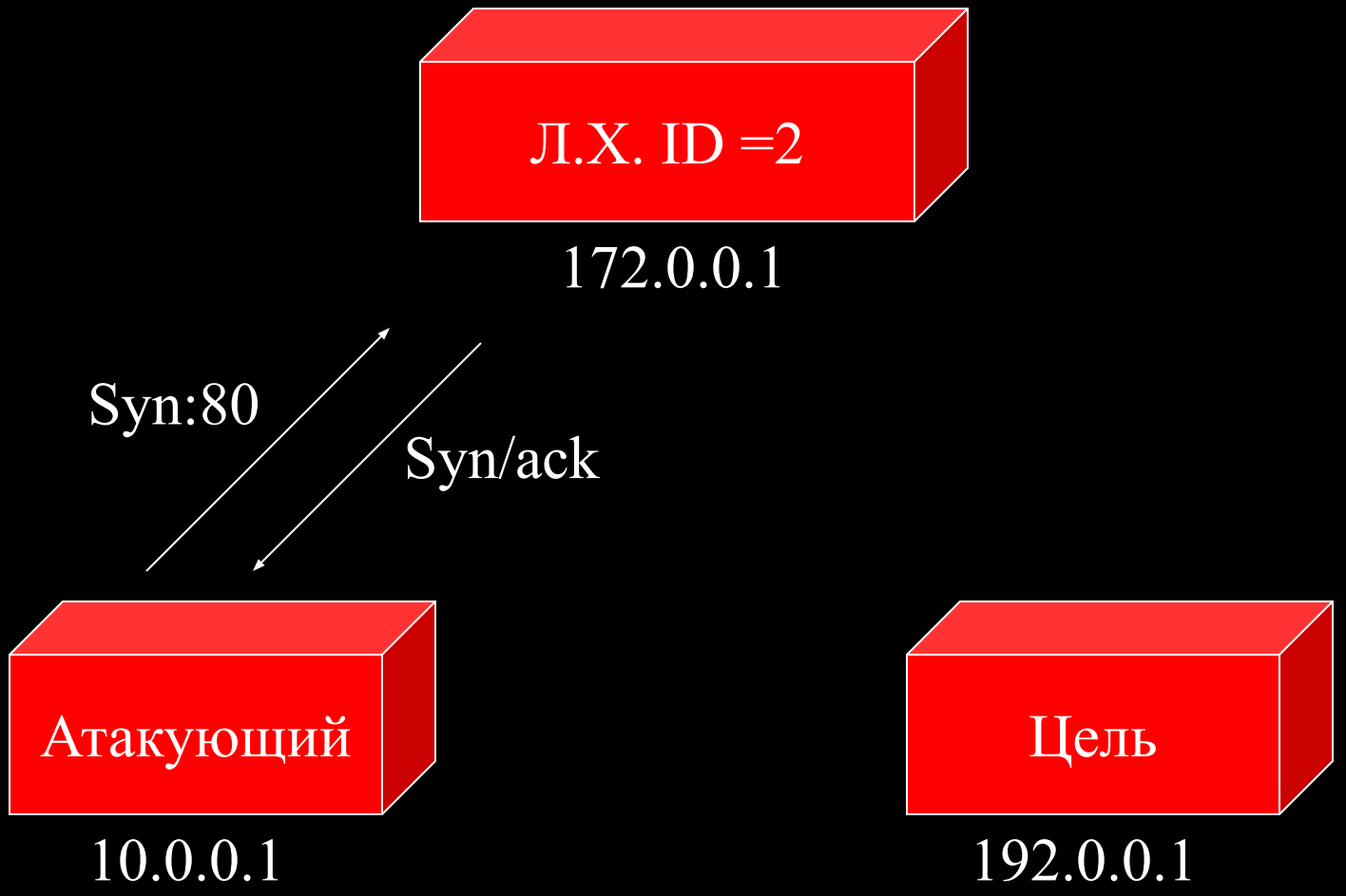
Шаг 3 (сброс ответов)



Шаг 4 (тест ID ложного хоста)



Шаг 4 (тест ID ложного хоста)



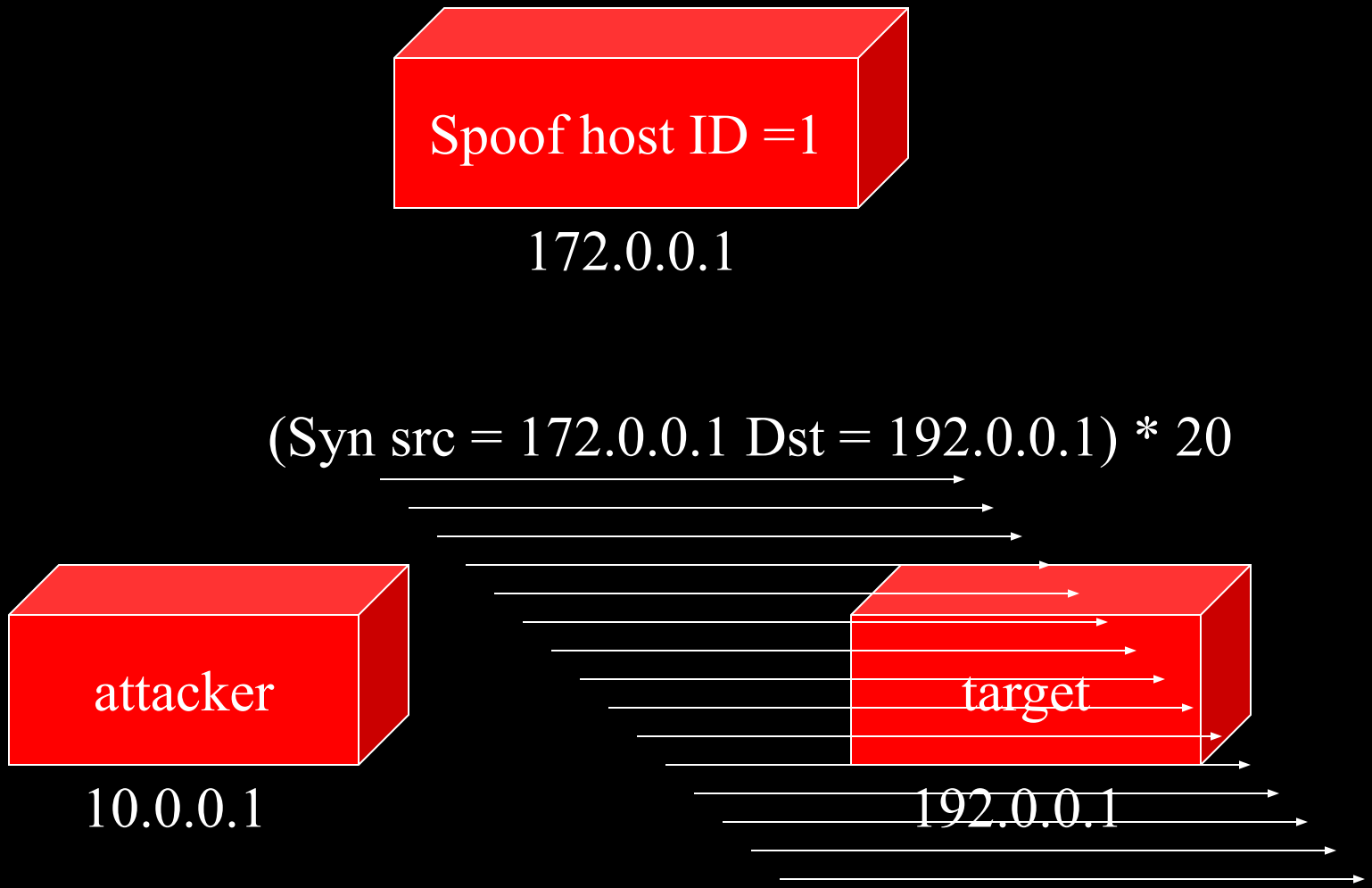
Итоги:

- Постоянно опрашивая ложный хост на предмет увеличения его id можно увидеть, отправил ли сканируемый хост syn/ack или reset.
- Анализируя это можно определить, какой порт сканируемого хоста открыт, а какой нет
- На стороне сканируемого хоста эта операция абсолютно невидима.

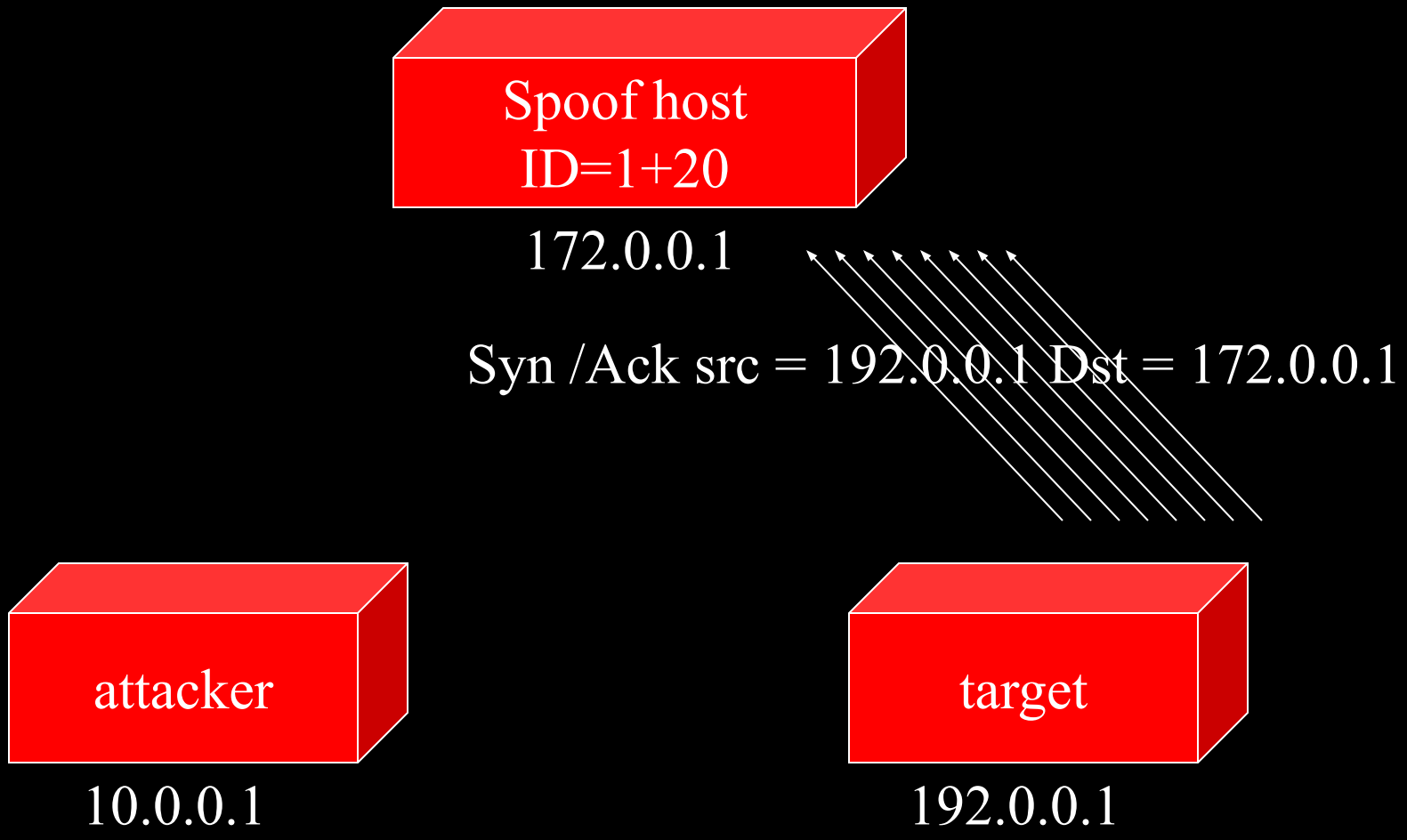
Недостатки этой техники

- Если ложный хост активен и имеет несколько подключений, значение id# будет увеличиваться для каждого отправляемого пакета.
- Это приведет к ложным «открытым» портам.
- Этого можно избежать, отправляя несколько тестов на один порт.
- Затем вычислить увеличение
- Порт будет открыт, если увеличение будет > (#число_отпр_пакетов*255)/2

Шаг 2 (подделываем источник)



Шаг 3 (сброс ответов)



Вот и все!!!

Теперь наш хост
успешно просканирован.

© Thomas Olofsson, C.T.O, Defcom.

Эта техника реализована в сканере RuNmap v. 3.27

alex@cherepovets-city.ru

http://www.cherepovets-city.ru/insecure