МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Московский Государственный Университет им. М.В.Ломоносова

ДИСЦИПЛИНА

«Введение в проблемы информационной безопасности»

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

по проведению лабораторной работы № 2

«Атака ARP-spoofing»

Москва, 2016

Оглавление

ЦЕЛЬ РАБОТЫ

Получить знания о принципах работы протокола ARP и способах осуществления атак «man-in-the-middle», научиться пользоваться основными функциональными возможностями программ Wireshark и Ettercap, произвести с их помощью перехват данных, передаваемых между отдельными узлами сети.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1. Протокол ARP

1.1. Назначение

Отдельные компьютеры и сетевое оборудование, соединенное с помощью линий связи осуществляют передачу информации, представленной в виде двоичного кода, при этом каждый сетевой интерфейс имеет уникальный адрес (MAC-адрес), тогда как маршрутизация пакетов осуществляется на сетевом уровне на основе IP-адресов.

Между IP-адресом устройства и физическим адресом его сетевого интерфейса не всегда можно провести однозначное соответствие (более того, в некоторых случаях число устройств постоянно меняется, как например в городских WiFi-сетях). Для того чтобы получить такое соответствие, используется протокол ARP (Address Resolution Protocol).

Помимо выполнения указанной выше функции протокол ARP также может использоваться для проверки, нет ли в сети хостов с совпадающими IP-адресами, а также для оповещения всех узлов о появлении нового (инициатором выступает устройство, подключившееся к сети).

1.2. Принцип работы

Предположим, компьютеры A, B, C были соединены в локальную сеть с помощью коммутатора S (рисунок 1).

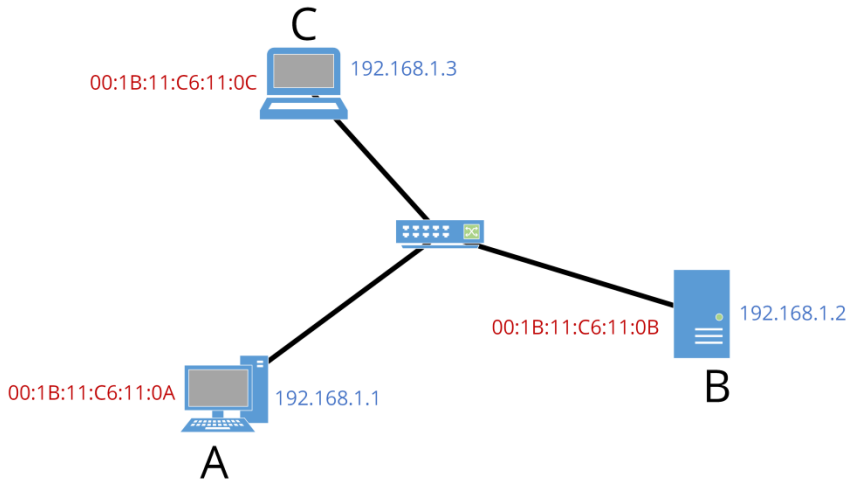

Рисунок 1 - Топология сети

Физические адреса их интерфейсов установлены производителем, а сетевые адреса – администратором сети. Коммутатор (switch) выбирает порт, по которому пойдет сообщение, на основании MAC-адреса получателя.

Если узлу A: IP 192.168.1.1 необходимо отправить данные на узел B: 192.168.1.2, то, для успешной коммутации, ему необходим физический адрес интерфейса-получателя. Для его нахождения узел A посылает ARP-запрос (рисунок 2), структура которого изображена в таблице 1.

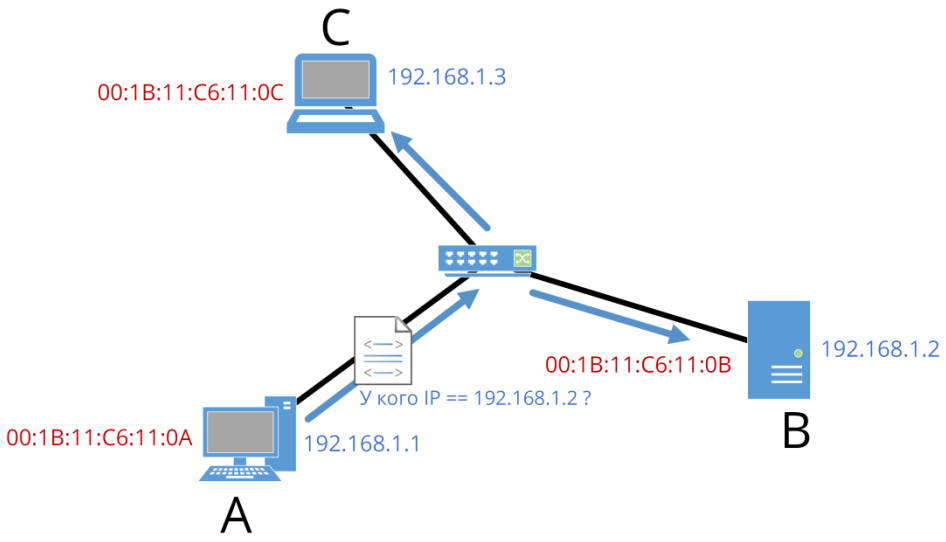

Рисунок 2 - ARP-запрос

Таблица 1 - Структура ARP-запроса

№ п/п.	Название	Длина поля, бит	Значение	Комментарий
1	Тип канального протокола (HTYPE)	16	0x0001	Означает Ethernet
2	Тип сетевого протокола (PTYPE)	16	0x0800	Означает IPv4
3	Длина физического адреса (HLEN)	8	6	В MAC-адресе 6 байт
4	Длина логического адреса (PLEN)	8	4	В IPv4 адресе 4 байта
5	Код операции (operation)	16	0x0001	Означает запрос
6	Физический адрес отправителя (SHA)	HLEN	00:1B:11:C6:11:0A	Свой адрес

№ п/п.	Название	Длина поля, бит	Значение	Комментарий
7	Логический адрес отправителя (SPA)	PLEN	192.168.1.1	
8	Физический адрес получателя (THA)	HLEN	00:00:00:00:00:00	Неизвестно. Необходимо получить
9	Логический адрес получателя (TPA)	PLEN	192.168.1.2	Чей MAC необходимо получить

Узел B, получив запрос и сверив адрес получателя со своим (192.168.1.2), производит следующие действия:

- Вносит в свою ARP-таблицу запись о соответствии MAC и IP адресов узла-отправителя;
- Отправляет ARP-ответ на MAC-адрес узла B, в котором сообщает свой MAC (рисунок 3).

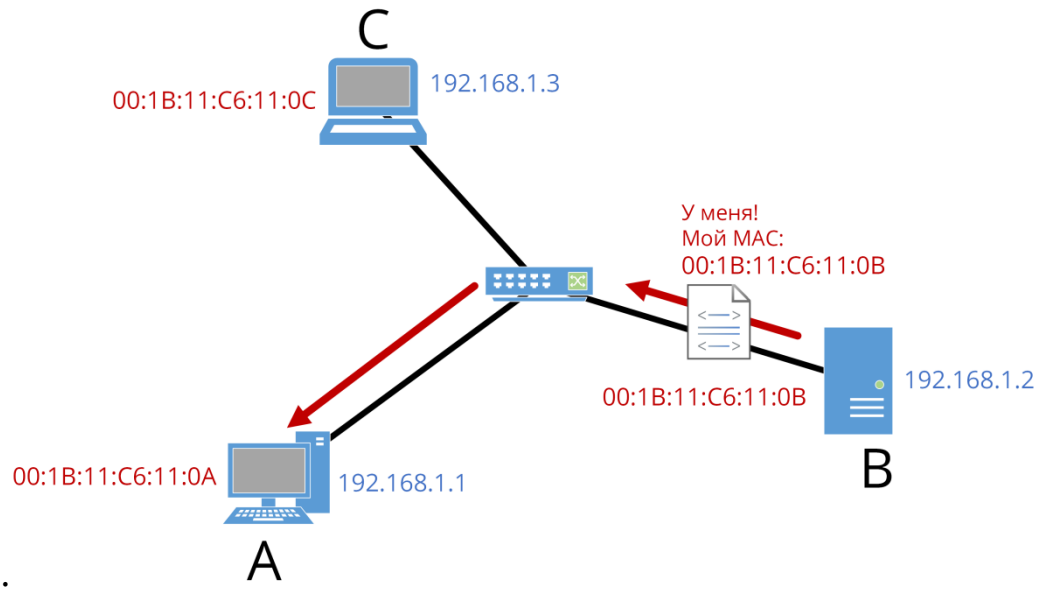

Рисунок 3 - ARP-ответ

Таблица 2 - Структура APR-ответа

№ п/п.	Название	Длина поля, бит	Значение	Комментарий
1	HTYPE	16	0x0001	Означает Ethernet
2	PTYPE	16	0x0800	Означает IPv4
3	HLEN	8	6	В MAC-адресе 6 байт
4	PLEN	8	4	В IPv4 адресе 4 байта
5	operation	16	0x0002	Означает ответ
6	SHA	HLEN	00:1B:11:C6:11:0B	Свой адрес
7	SPA	PLEN	192.168.1.2	
8	THA	HLEN	00:1B:11:C6:11:0A	Адрес получателя, известен из запроса
9	TPA	PLEN	192.168.1.1	

Так как ARP-запрос широковещательный, его получат все узлы широковещательного домена, а значит, занесут адреса отправителя в ARP-кэш.

1.3. ARP-таблица

Пары IP-MAC содержатся в ARP-таблицах на каждом устройстве. Посмотреть таблицу можно с помощью команды **arp** (работа с ARP-таблицей) с аргументом **-a** (показать записи). Результат выполнения приведен на рисунке 4.

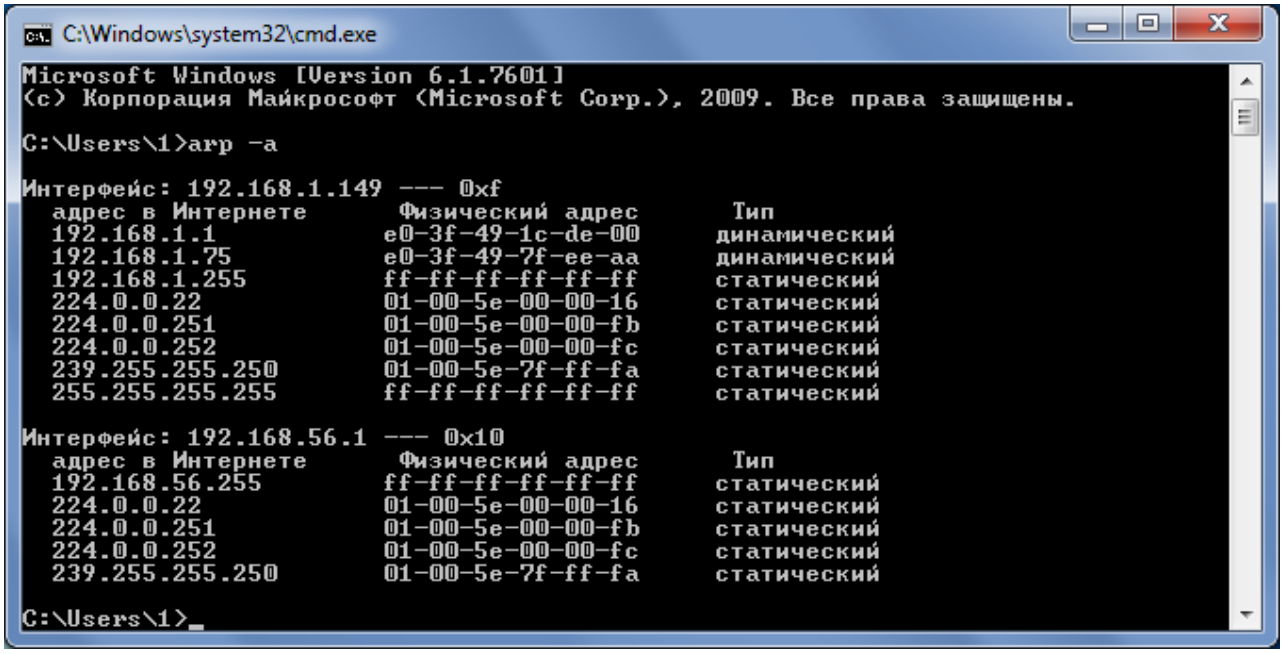

Рисунок 4 - ARP-таблица

Записи в таблице могут быть как статическими (не меняются) и динамическими (получены посредством обмена ARP-сообщениями). В зависимости от устройства динамические записи могут существовать от нескольких секунд до 4 часов.

Статические записи можно добавлять с помощью аргумента -s.

В Linux в некоторых дистрибутивах есть утилита arp, а в некоторых (например Debian) почему-то такой утилиты нет, и вместо нее можно использовать arp-scan. Возможно, arp лежит в net-tools... Утилита arp запускается в Debian только в окружении root.

1.4. Безопасность протокола ARP

Протокол ARP не предусматривает каких-либо средств, обеспечивающих его безопасность. ARP-сообщения не требуют аутентификации и не генерируют подтверждений, а значит могут быть отправлены с любого узла. Более того, получив ответ, хост заносит соответствующую запись в свою ARP-таблицу вне зависимости от того, посылал он запрос или нет. Этим может воспользоваться злоумышленник для совершения атаки ARP-spoofing.

1.5. Атака ARP-спуфинг

Данная атака заключается в отправке злоумышленником ложных ARP-пакетов, с помощью которых удается перенаправить трафик на компьютер злоумышленника.

Рассмотрим такую атаку на примере вышеупомянутой сети. Пусть злоумышленник получил доступ к узлу C и заинтересован в перехвате данных между узлами A и B. Уязвимости протокола APR позволят ему это сделать.

Состояние ARP-таблиц в узлах сети до атаки представлено на рисунке 5.

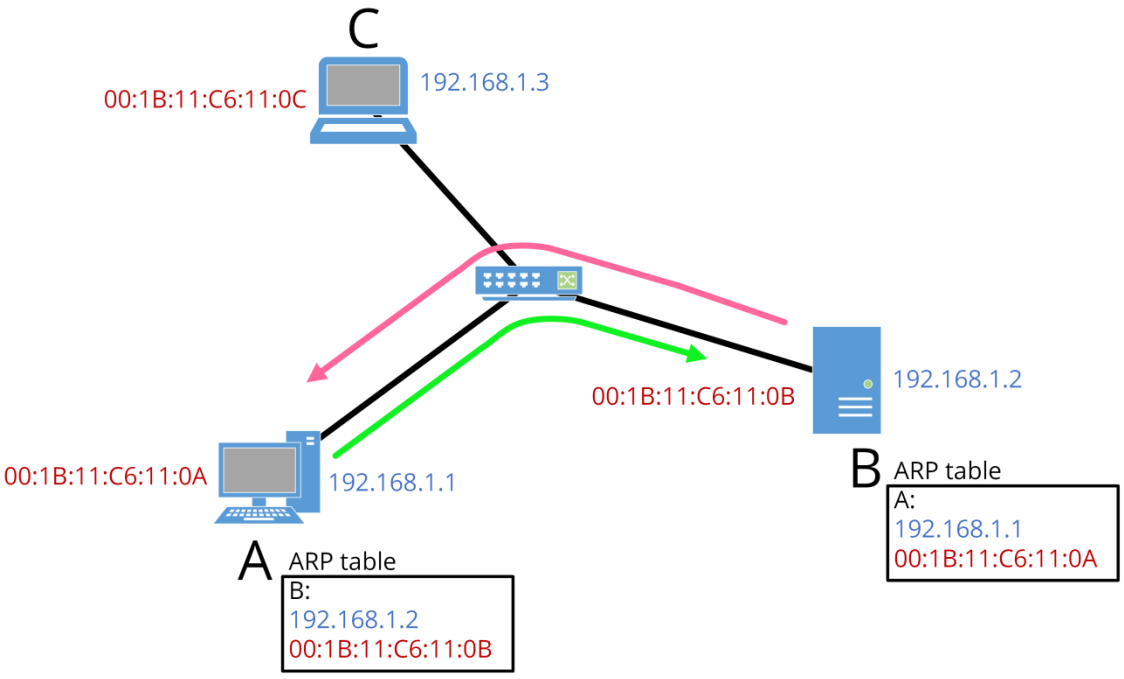

Рисунок 5 - Состояние сети перед атакой

Для перенаправления маршрута A→B Злоумышленник отправляет на узел A ARP-пакеты, в которых сетевой адрес узла B соответствует физическому адресу C (рисунок 6).

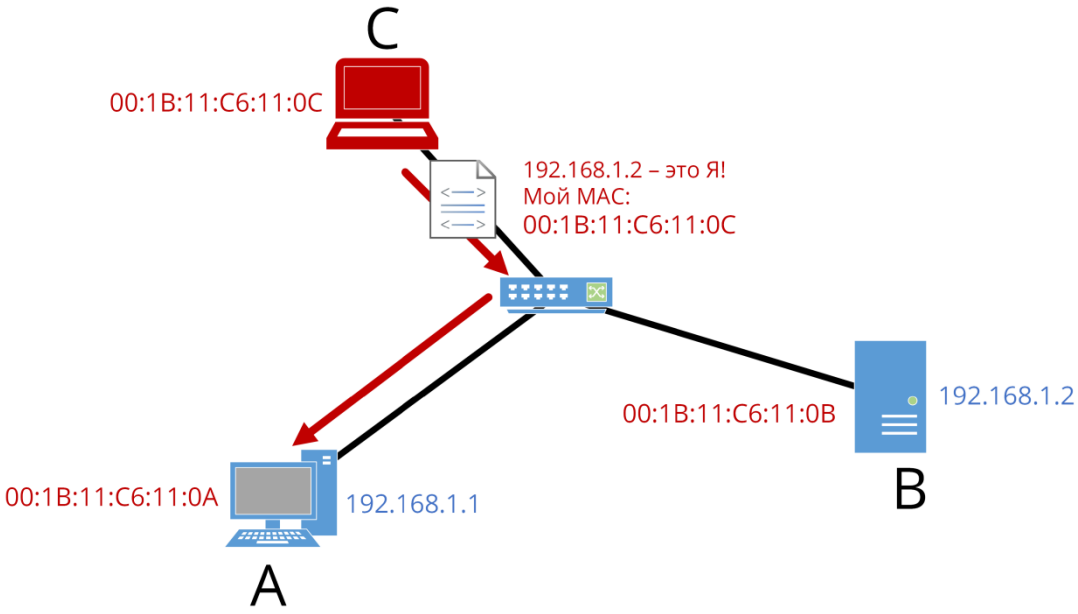

Рисунок 6 - Атака на узел A

Для того чтобы осуществлять перехват трафика и в обратном направлении, злоумышленник выполняет аналогичную операцию в отношении узла B (рисунок 7).

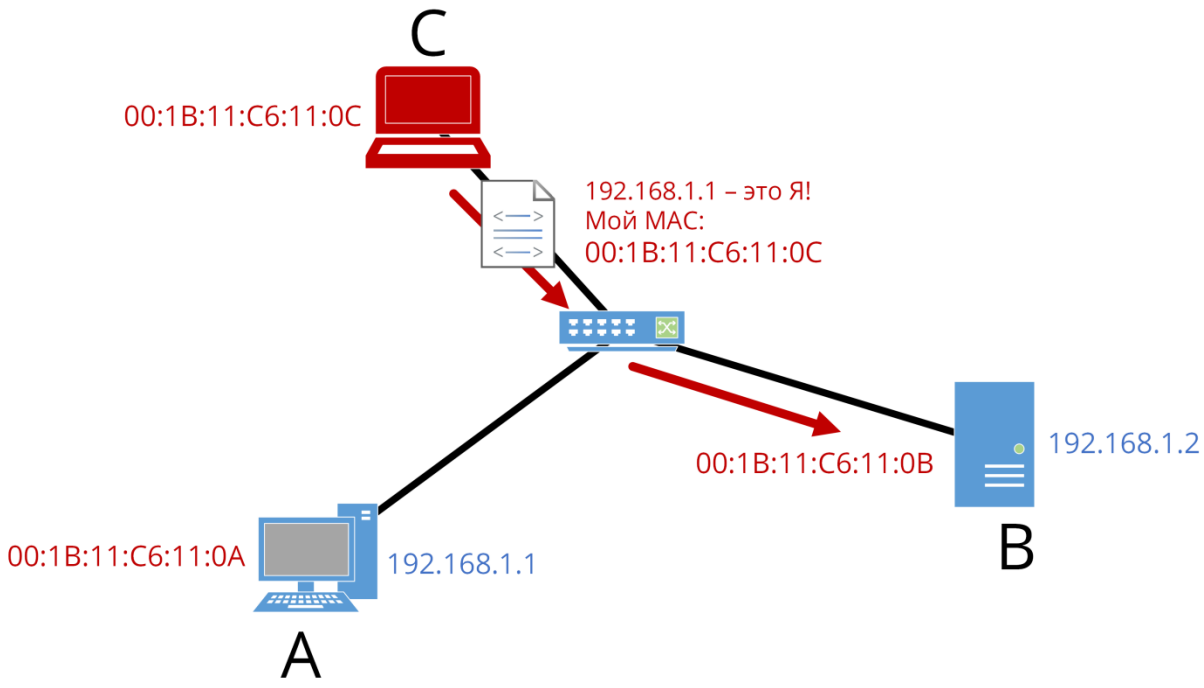

Рисунок 7 - Атака на узел B

В результате, из-за неверных записей в ARP-таблицах хостов, весь трафик проходит через злоумышленника (рисунок 8) и может быть им проанализирован.

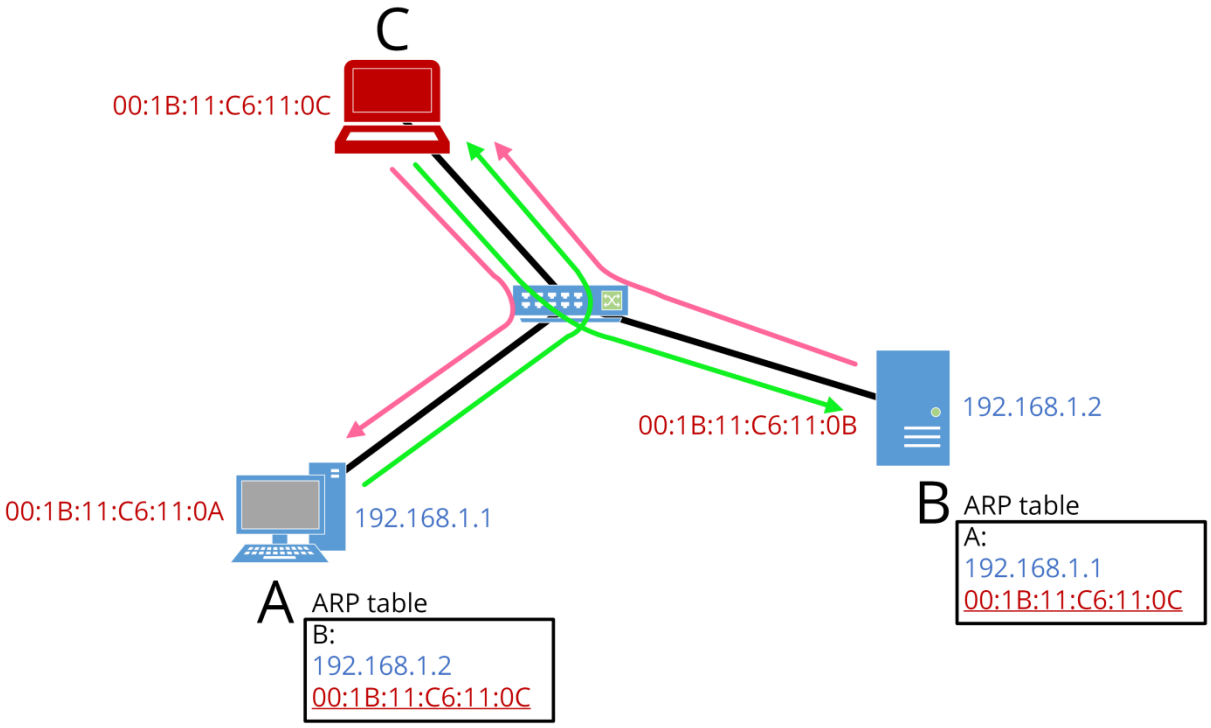

Рисунок 8 - Направление транспортных потоков во время атаки

Так как динамические записи носят временный характер, злоумышленнику приходится постоянно отправлять в сеть ложные пакеты. Когда злоумышленник прекратит генерацию сообщений, сеть вернется в прежнее состояние после первого же ARP-запроса.

2. Анализатор сетевого трафика Wireshark

2.1. Общие сведения

Wireshark представляет из себя мощный анализатор сетевого трафика, поддерживающий подавляющее большинство протоколов и обладающий разнообразными средствами фильтрации и анализа трафика. Помимо этого он имеет удобный пользовательский интерфейс и является кроссплатформенным продуктом, работающим в том числе в Linux и Windows. Стартовое меню программы показано на рисунке 9.

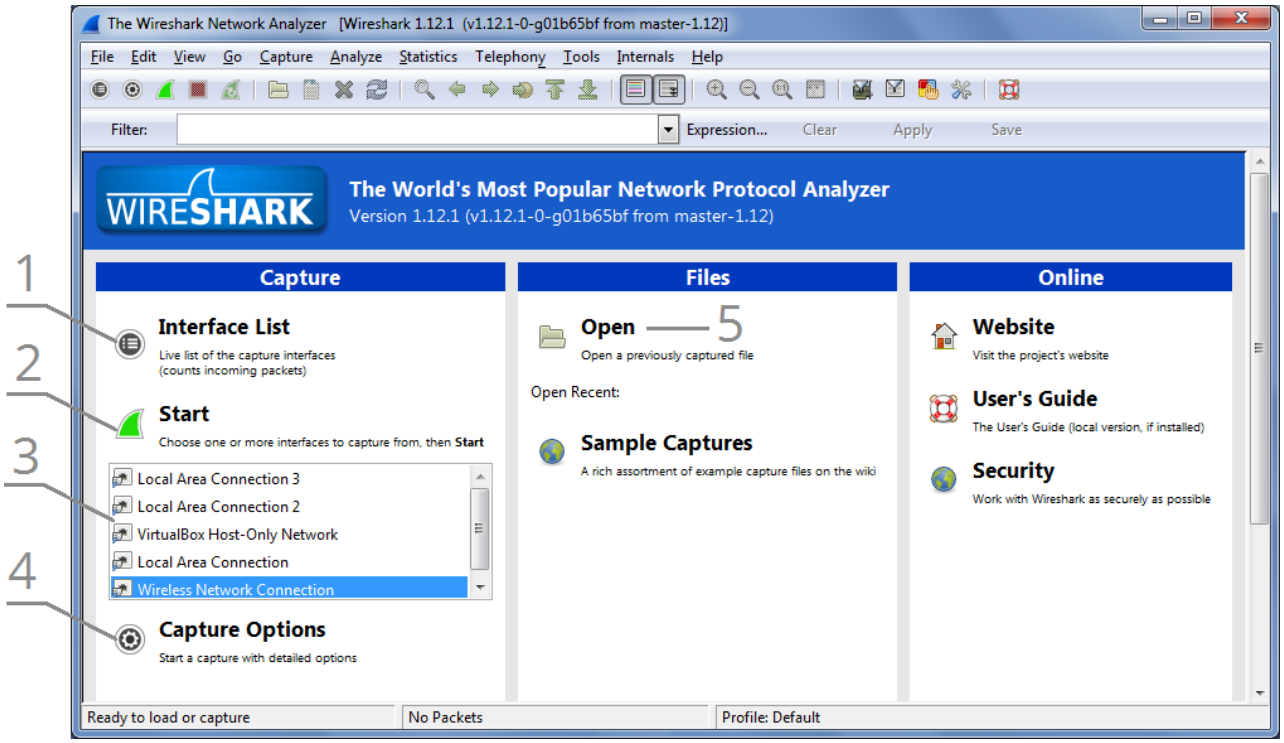

Рисунок 9 - Стартовое меню программы Wireshark

Цифрами обозначены следующие элементы:

1. Вызов меню выбора сетевых интерфейсов;
2. Запуск захвата трафика;
3. Краткий список интерфейсов;

4. Вызов меню опций захвата;
5. Выбор дампа захваченных ранее пакетов для анализа.

2.2. Захват трафика

Для осуществления захвата трафика с помощью Wireshark необходимо выбрать сетевой интерфейс из краткого списка и нажать кнопку запуска. В случае если интерфейсов много, можно вызвать меню выбора сетевых интерфейсов (рисунок 10), где можно получить о них более подробную информацию, а также увидеть количество трафика, проходящего через каждый.

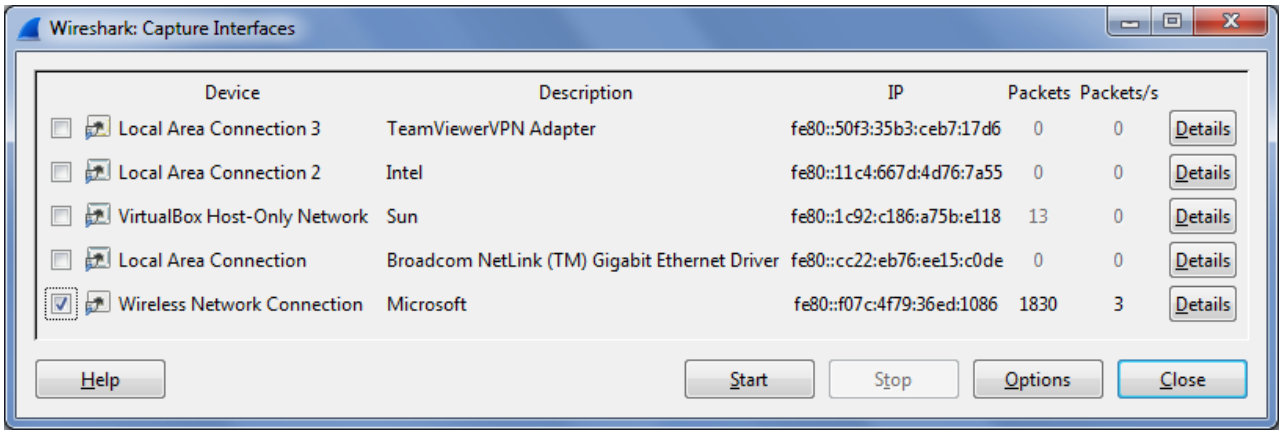

Рисунок 10 - Меню выбора сетевых интерфейсов для захвата. Числа справа отражают интенсивность трафика через каждый интерфейс

Вид окна программы после запуска захвата трафика представлен на рисунке 11.

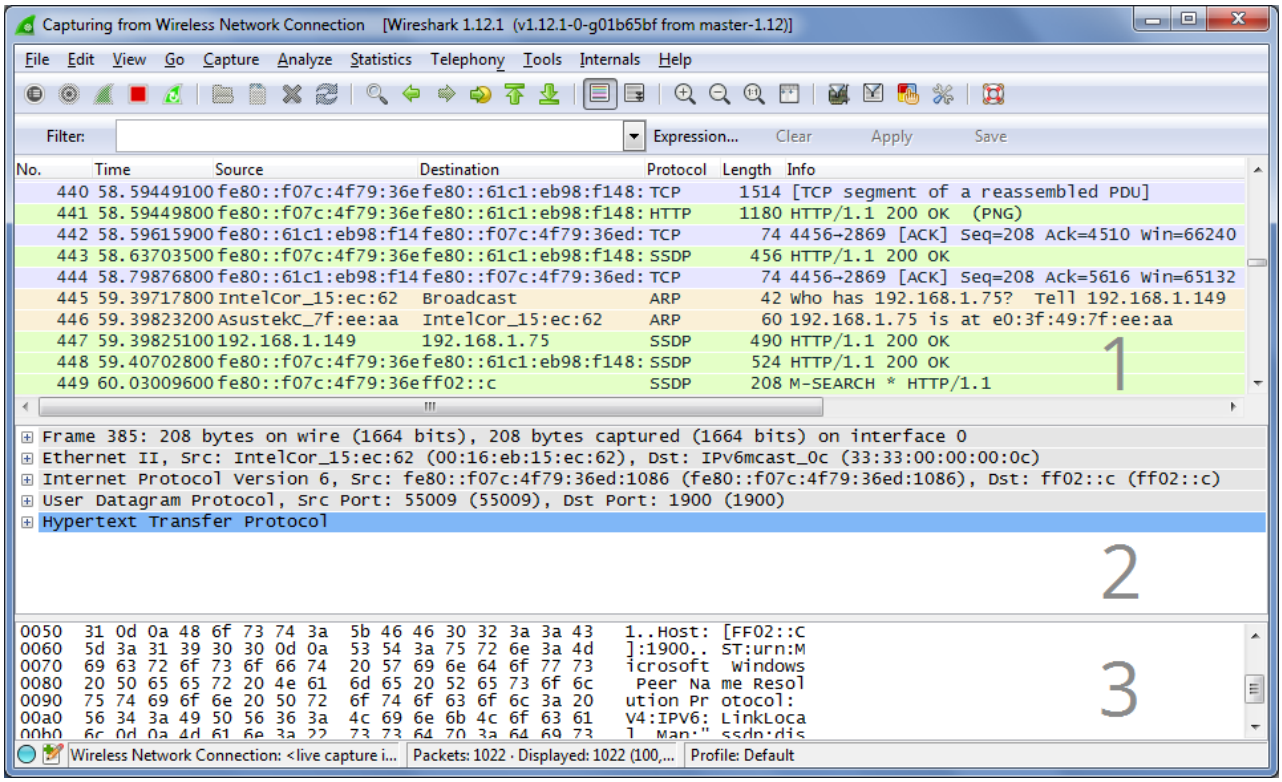

Рисунок 11 - Wireshark в режиме захвата трафика

Цифрами обозначены следующие элементы:

1. Окно захваченных пакетов. Пакеты разного назначения отображаются разным цветом.
2. Окно содержимого выбранного пакета, разложенное по стеку протоколов. Присутствует возможность открыть заголовок каждого протокола и посмотреть значения полей. При выборе любого поля, в 3-м разделе выделяются соответствующие ему двоичные данные.
3. Представление пакета в двоичном виде. Разделен на 4 столбца: первый – номер первого байта строки в 16-ричном виде, второй и третий – представление пакета в 16-ричном виде, четвертый – представление пакета в кодировке ASCII.

2.3. Фильтрация трафика

В программе Wireshark существуют 2 вида фильтров:

- Фильтры захвата (Capture filters);
- Фильтры отображения (Display filters).

Фильтры захвата настраиваются перед началом захвата. Для их настройки необходимо вызвать опции захвата (Capture options) (рисунок 12).

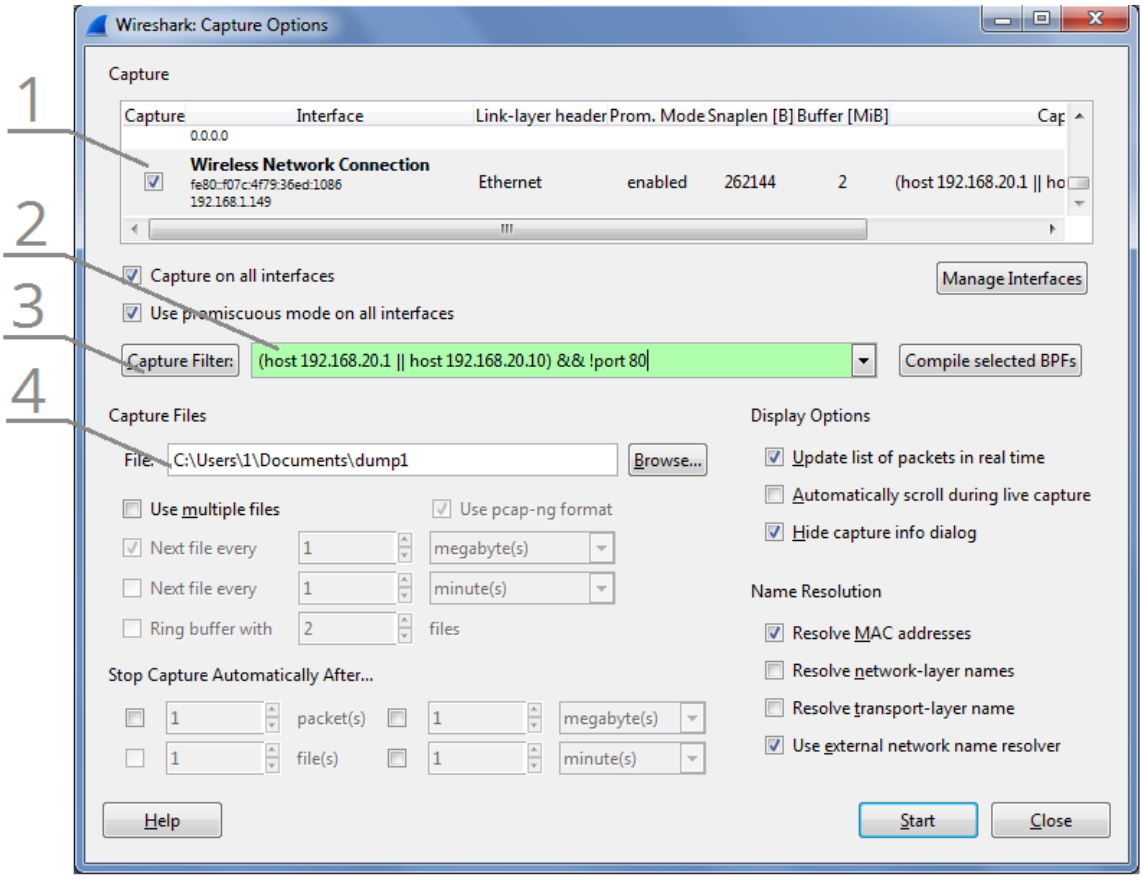

Рисунок 12 - Меню настройки захвата

Цифрами обозначены следующие элементы:

1. Список сетевых интерфейсов;
2. Выражение для фильтра захвата;
3. Сформировать фильтр на основе имеющихся шаблонов;
4. Путь к файлу для сохранения дампа трафика.

Выражение для фильтра захвата может содержать определенные параметры и их значения, например host 192.168.20.1 означает, что будет производиться только захват трафика с хостом 192.168.20.1. Список возможных параметров можно узнать, нажав кнопку Capture Filter (3) и выбрав интересующее условие.

Отдельные условия можно объединять с помощью логических функций || - или, && - и, ! - не, а также скобок.

При необходимости можно сохранять трафик в файл, для этого необходимо выбрать путь к нему в поле (4).

Фильтры отображения обладают более гибкими настройками, их можно менять в процессе работы захвата трафика. Элементы управления, отвечающие за фильтры отображения приведены на рисунке 13.

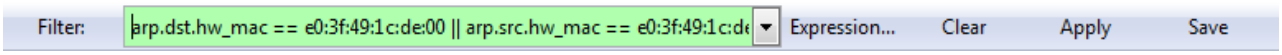

Рисунок 13 - Элементы управления фильтрами отображения

Так же, как и с фильтрами захвата, параметры фильтрации задаются выражениями и позволяют использовать логические функции. Но, в отличие от предыдущих, с помощью фильтров отображения можно ставить условия, используя поля заголовков пакетов любого уровня. Для просмотра шаблонных фильтров используется кнопка Filter.

Для фильтрации по определенному полю удобно использовать меню создания фильтра (рисунок 14), вызываемое кнопкой Expression.

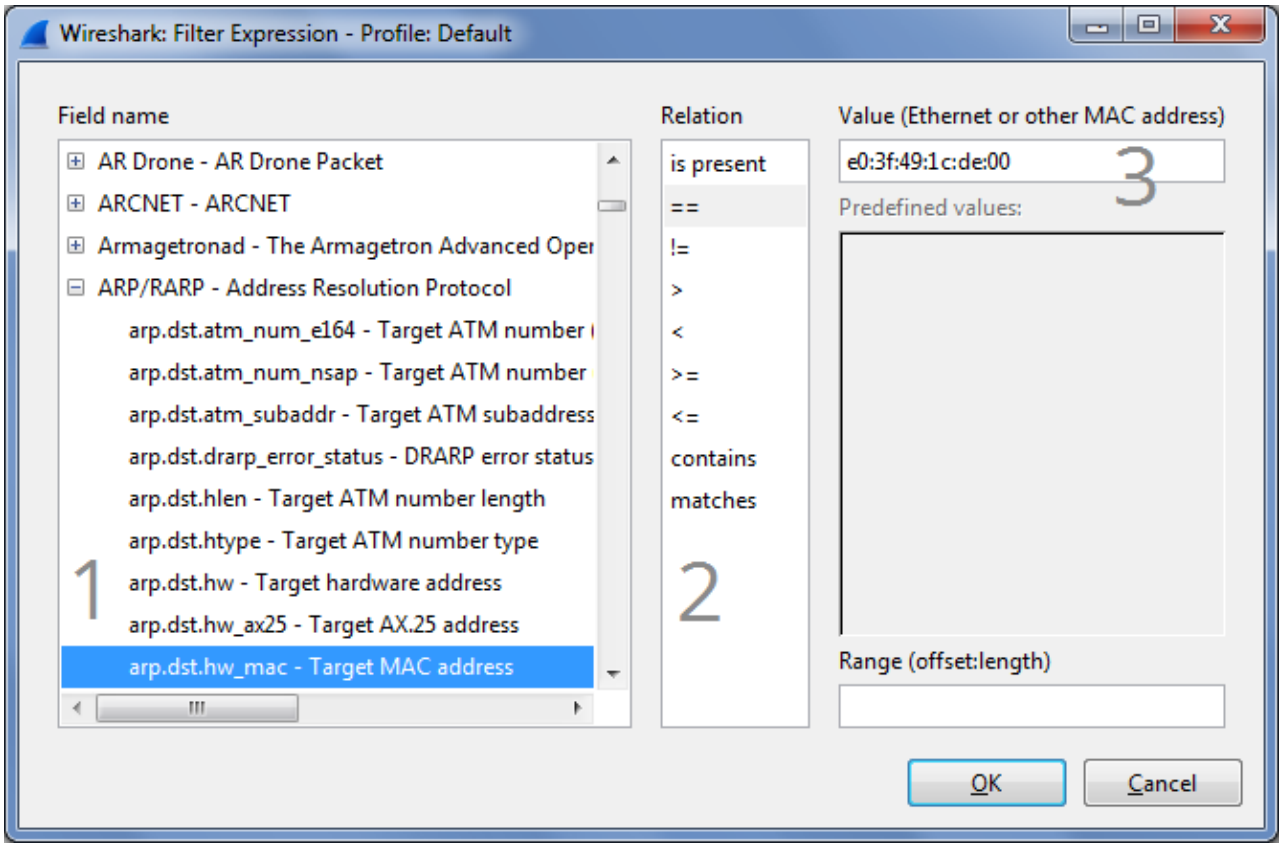

Рисунок 14 - Меню формирования выражения фильтрации

Цифрами отмечены следующие элементы:

1. Список протоколов и полей их заголовков. Для быстрого поиска можно набрать название интересующего протокола на клавиатуре.
2. Логическое выражение. Помимо обычных математических отношений можно использовать «contains» - содержится в качестве подстроки (удобно использовать при поиске в http-заголовке) и «matches» - соответствие регулярному выражению.
3. Значение, с которым выполняется сравнение.

При нажатии на кнопку OK сформированное выражение появится на месте курсора (из-за этого могут быть ошибки).

Для применения фильтра служит кнопка Apply. С помощью Save можно сохранить получившийся фильтр в качестве шаблонного.

2.4. Просмотр TCP-сессии

Данные, передаваемые в текущей TCP-сессии можно просматривать в обычном режиме декапсуляции пакетов, а можно воспользоваться специальным инструментом просмотра TCP-сессий. Для его вызова необходимо щелкнуть правой кнопкой мыши на интересующем пакете и выбрать «Follow TCP Stream» (рисунок 15). Окно просмотра TCP-сессии представлено на рисунке 16.

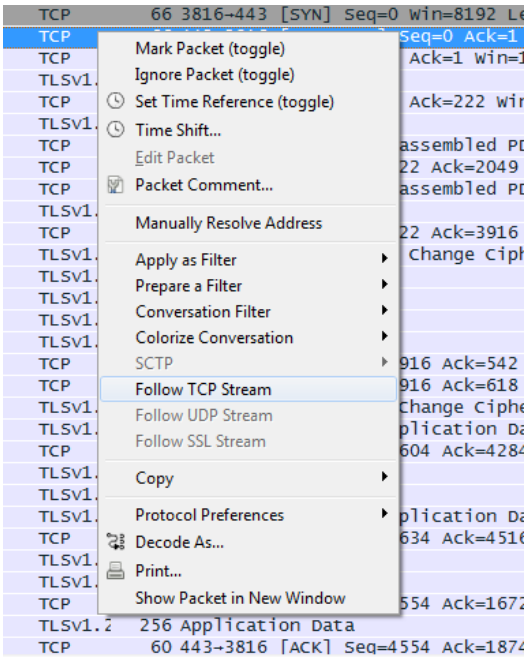

Рисунок **15** - Вызов окна просмотра TCP-сессии

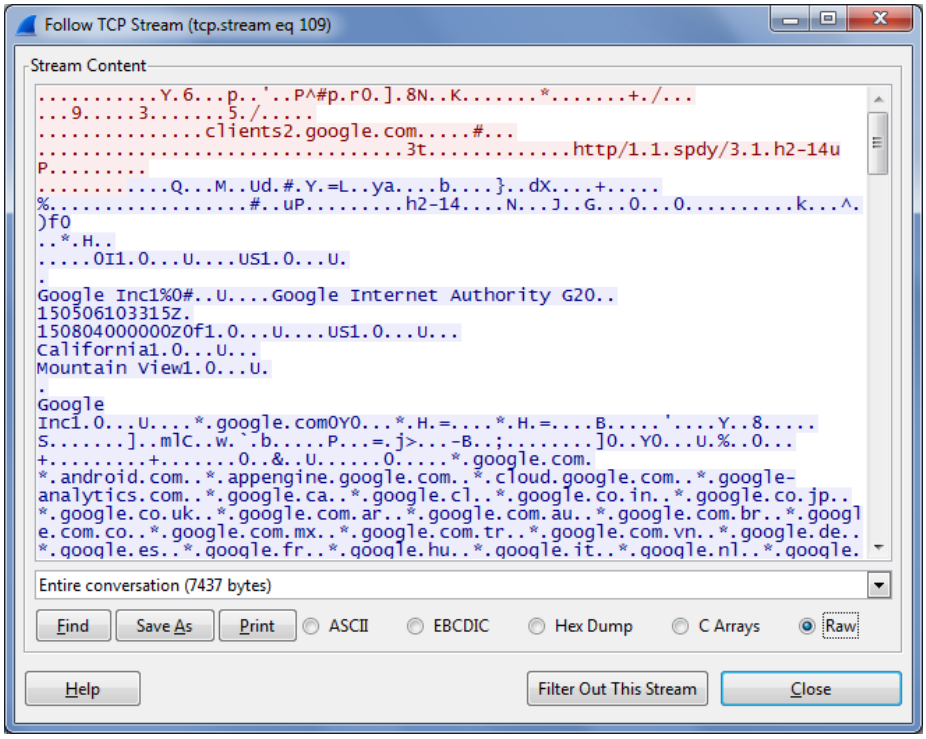

Рисунок 16 - Окно просмотра TCP-сессии

2.5. Просмотр содержимого HTTP

Wireshark умеет выделять из перехваченных HTTP-пакетов отдельные объекты. Для их просмотра можно воспользоваться соответствующей опцией:

File → Export Objects → HTTP

Окно просмотра HTTP-объектов представлен на рисунке 17.

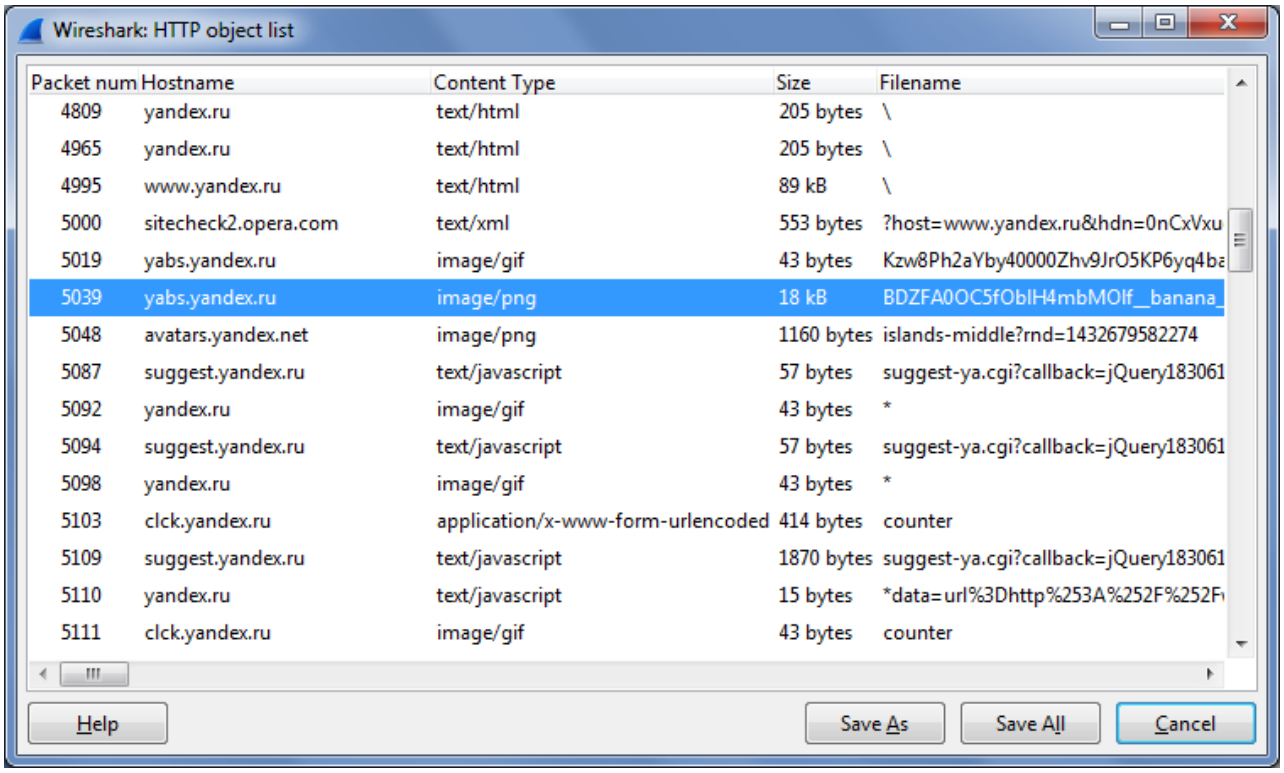

Рисунок **17** - Окно просмотра HTTP-объектов

В этом режиме можно выбрать интересующий объект и сохранить на диске для дальнейшего просмотра.

3. Средство проведения MITM-атак Ettercap

3.1. Общие сведения

Программа Ettercap служит для проведения атак вида «Man in the middle».

Поддерживает как графический интерфейс (запуск с параметром -G), так и консольный (запуск с параметром -T). Графический интерфейс программы представлен на рисунке 18.

Рисунок 18 - Графический интерфейс программы Ettercap

Первое, что предлагает выбрать программа – это вид прослушивания в меню Sniff (рисунок 19):

- Unified sniffing – обычное прослушивание через выбранный интерфейс.
- Bridged sniffing – прослушивание в качестве сетевого моста, затрудняет обнаружение.

Заметим, что в рамках данной лабораторной работы захват трафика осуществляется с помощью Wireshark, а Ettercap используется исключительно для генерации пакетов.

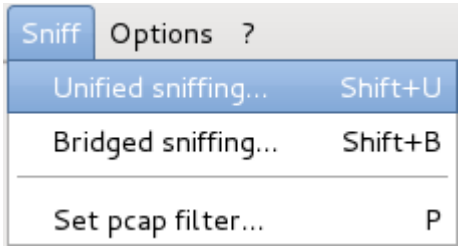

Рисунок 19 - Выбор вида прослушивания

3.2. Выбор целей для атаки

После выбора метода прослушивания в программе появляются дополнительные пункты меню. Так, появляется возможность просканировать сеть командой Hosts → Scan for hosts (рисунок 20) и вывести список узлов командой Hosts → Hosts list.

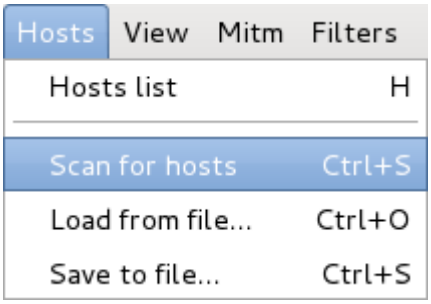

Рисунок 20 - Комманда по поиску узлов сети

Из найденных хостов необходимо выбрать как минимум 2 цели для атаки, для этого можно использовать команды Add to Target 1 и Add to Target 2. Посмотреть текущий список целей можно командой Targets → Current Targets (рисунок 21), после чего появляется вкладка Targets, содержащая выбранные цели.

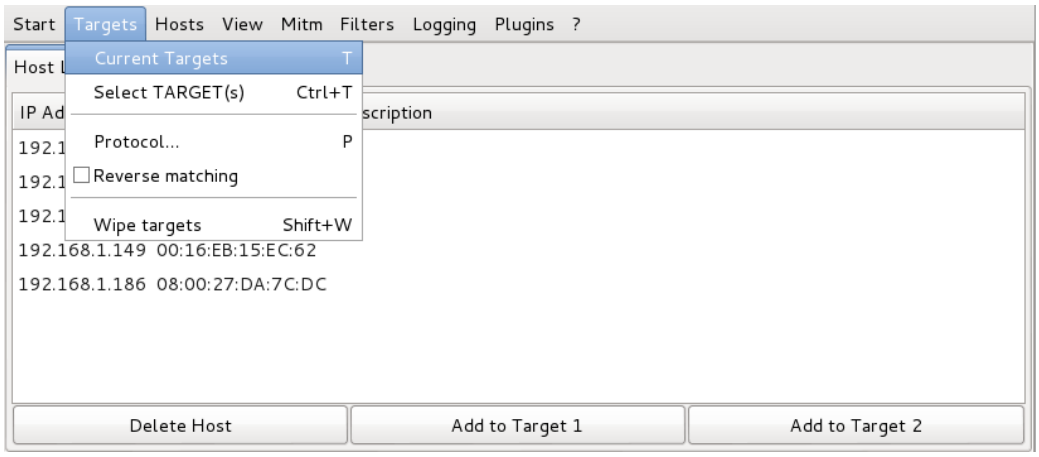

Рисунок 21 - Выбор целей для атаки

3.3. Запуск атаки

Вид атаки можно выбрать в пункте меню Mitm (рисунок 22). При выборе ARP-poisoning возникает окно со следующими опциями:

- Sniff remote connections – используется для перехвата трафика, проходящего через выбранный узел
- Only sniff one-way – перехватывает только трафик, передаваемый с Target 1 на Target 2.

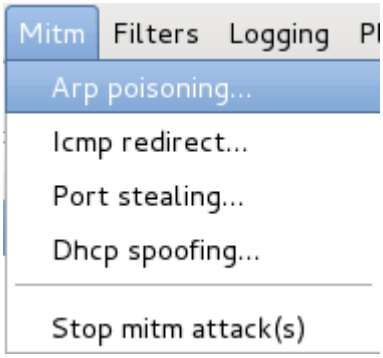

Рисунок **22** - Выбор вида атаки

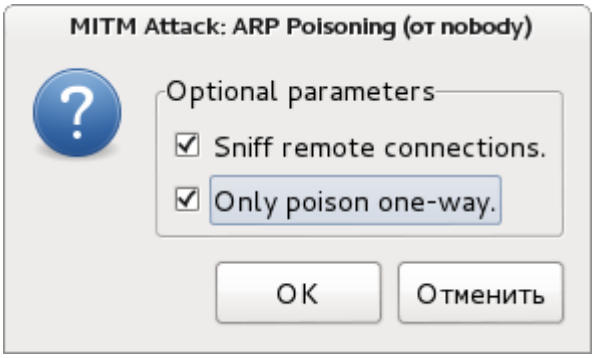

Рисунок **23** – Параметры

После нажатия на кнопку ОК, начинается реализация атаки. Для её остановки необходимо выбрать Mitm → Stop mitm attacks.

ПРАКТИЧЕСКАЯ ЧАСТЬ

3.4. Используемые средства

Лабораторная работа выполняется на ПК с установленной программой Oracle VirtualBox. В качестве гостевой ОС используется Kali Linux, включающий в себя все необходимые средства для осуществления данной атаки.

Также для обеспечения перехвата необходимо иметь возможность перевода сетевого интерфейса в режим «monitor» или «promiscuous». В ОС Windows для этого необходимо установить библиотеку WinPcap, в Linux – LibPcap (в Kali Linux уже присутствует).

В процессе выполнения предусматривается работа со следующими программными продуктами:

- Wireshark – анализатор сетевого трафика;
- Ettercap – Программа для проведения MITM атак.

В качестве источника трафика для перехвата используется программа-бот, установленная на несколько компьютеров, являющихся целями для атаки. Программа имитирует работу пользователя, выполняя интернет запросы каждые несколько секунд.

3.5. Подготовка компьютерного класса к проведению лабораторной работы

Перед проведением лабораторной работы в компьютерном классе необходимо удостовериться в том, что были выполнены следующие действия:

1. Выделено необходимое количество компьютеров по количеству учащихся, выполняющих работу;
2. Выделено достаточное количество компьютеров-жертв из расчета не более 5 учащихся на одну «жертву»;

3. Участвующие в лабораторной работе компьютеры соединены в коммутируемую сеть;
4. На каждый компьютер участника установлена библиотека WinPcap (в случае ОС Windows) или LibPcap (в случае Linux) для успешного перевода сетевого адаптера в режим мониторинга;
5. На компьютеры учащихся установлено программное обеспечение для запуска виртуальных машин с возможностью подключения к сетевому интерфейсу в режиме сетевого моста (Virtual Box либо VMWare);
6. У учащихся есть доступ к виртуальной машине, имеющей в своем составе средства анализа трафика (Wireshark) и средства для осуществления MITM-атаки (Ettercap), в параметрах виртаульной машины установлено соединение с сетью вида «сетевой мост»;
7. На компьютерах-жертвах установлен интерпретатор Python версии 2.x.x (необходимо для запуска бота) и интернет-браузер (для имитации действий пользователя, желательно Google Chrome).

Непосредственно перед проведением работы необходимо выполнить следующие действия:

1. На компьютерах-жертвах запускается интернет-браузер;
2. В случае наличия прокси-сервера, в браузере осуществляются необходимые настройки;
3. В конфигурационном файле (config.txt) для программы-бота устанавливаются следующие настройки: в поле period после знака равенства записывается период отправки запроса в секундах, в поле browser_path после знака равенства записывается путь к интернет-браузеру, в поле URL после знака равенства записывается посылаемый ботом интернет-запрос;
4. После сохранения изменений в конфигурационном файле, запускается исполняющий файл bot.exe (вариант: internetbot.py в случае ОС Windows это делается через cmd.exe, путем перехода в нужную

папку с помощью команды cd и запуска программы на исполнение с помощью команды python.exe internetbot.py).

После вышеописанных действий компьютер-жертва начинает отправлять запросы по указанному URL. Для остановки необходимо закрыть консольное окно с программой либо прекратить ее выполнение нажатием Ctrl+C в консоли.

3.6. Задача для учащихся

Используя предоставленные программные средства реализовать атаку ARP-spoofing на указанный компьютер (по вариантам).

Перехватить данные, передаваемые жертвой.

Извлечь из полученных данных запрашиваемые жертвой сведения.

Вопросы теоретического характера изложены в главе 1 теоретической части, интерфейс и функционал программных средств описан в главах 2 и 3 теоретической части.

СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист.
2. Задание.
3. Экранные снимки, демонстрирующие выполнение атаки.
4. Перехваченные текстовые запросы
5. Ответы на контрольные вопросы.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какими функциональными возможностями обладает Ettercap?
2. Как осуществляется фильтрация пакетов?
3. В чем заключается атака ARP-spoofing?
4. Как можно защититься от ARP-атак?
5. Как можно обнаружить ARP-атаку?

6*. Как осуществляется доставка трафика получателю в процессе атаки (при подмененных адресах)?

7*. Можно ли обнаружить факт прослушивания трафика сетевым устройством?