



**Информационная безопасность и
компьютерные сети**
Практическая работа №3
«Прокси-сервера»

*Гетьман Александр Игоревич
Маркин Юрий Витальевич
Обыденков Дмитрий Олегович*

Практические задания курса

	P1.1	P1.2	P1.3	P2	P3.1	P3.2	Бонус
Максимальный балл	1	1	2	2	1	2	1
Дата выдачи	19.09.2022	19.09.2022	19.09.2022	17.10.2022	14.11.2022	14.11.2021	-
Дата окончания приема	03.10.2022	03.10.2022	03.10.2022	14.11.2022	28.11.2022	05.12.2022	-
Сложность	☼	☼☼	☼☼☼	☼☼☼	☼☼	☼☼☼☼	-

$$\text{Бонус} = \text{if } P_{1.1} + P_{1.2} + P_{1.3} + P_2 + P_{3.1} + P_{3.2} = 9 \text{ then } 1 \text{ else } 0$$

Соглашение о наименовании

- Меня зовут:
 - *Иванов Петр Сергеевич*
- Мой вуз:
 - *МГУ*
 - *МФТИ*
 - *ВШЭ*
- Моя группа:
 - *999*
 - *М99-999х*
 - *МСП99*
- Подстановки:
 - <фамилияио> = *ivanovps*
 - <группа> = *999|m99_999х|msp99*
 - <вуз> = *msu|mipt|hse*
- Тема письма с решением:
 - *msu-999-p1_{1, 2, 3}*
 - *mipt-m99_999х-p1_{1, 2, 3}*
 - *hse-msp99-p1_{1, 2, 3}*
- Название архива:
 - *ivanovps-999-p1_{1, 2, 3}.zip*
 - *ivanovps-m99_999х-p1_{1, 2, 3}.zip*
 - *ivanovps-msp99-p1_{1, 2, 3}.zip*
- Посылки с иным форматом будут **проигнорированы**
 - Дефис является разделителем, используйте исключительно латинские символы, цифры и нижнее подчеркивание

Правила отправки решений

Not before	Not after	Максимальное количество посылок* в день**
	27.11.2022 / 04.12.2022 23:59	1
28.11.2022 / 05.12.2022 00:00	28.11.2022 / 05.12.2022 23:59	2
29.11.2022 / 06.12.2022 00:00		0

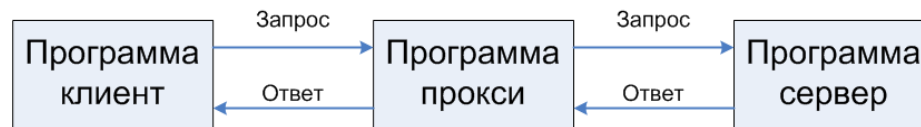
1. Посылки с некорректным форматом не будут проверяться (с уведомлением);
2. Если на посылку нет реакции более 2 рабочих дней, то письмо могло попасть в спам — свяжитесь с преподавателем иным способом (@dmt_obd);
3. Не объединяйте отправку P3.1/P3.2 в один тред;
4. В последний день перед дедлайном вы можете послать 2 посылки, а не одну;
5. (*) Для P1.1/P1.2/P1.3 отдельные счетчики посылок;
6. (**) Счетчик посылок обнуляется в 00:00.

Прокси-сервера

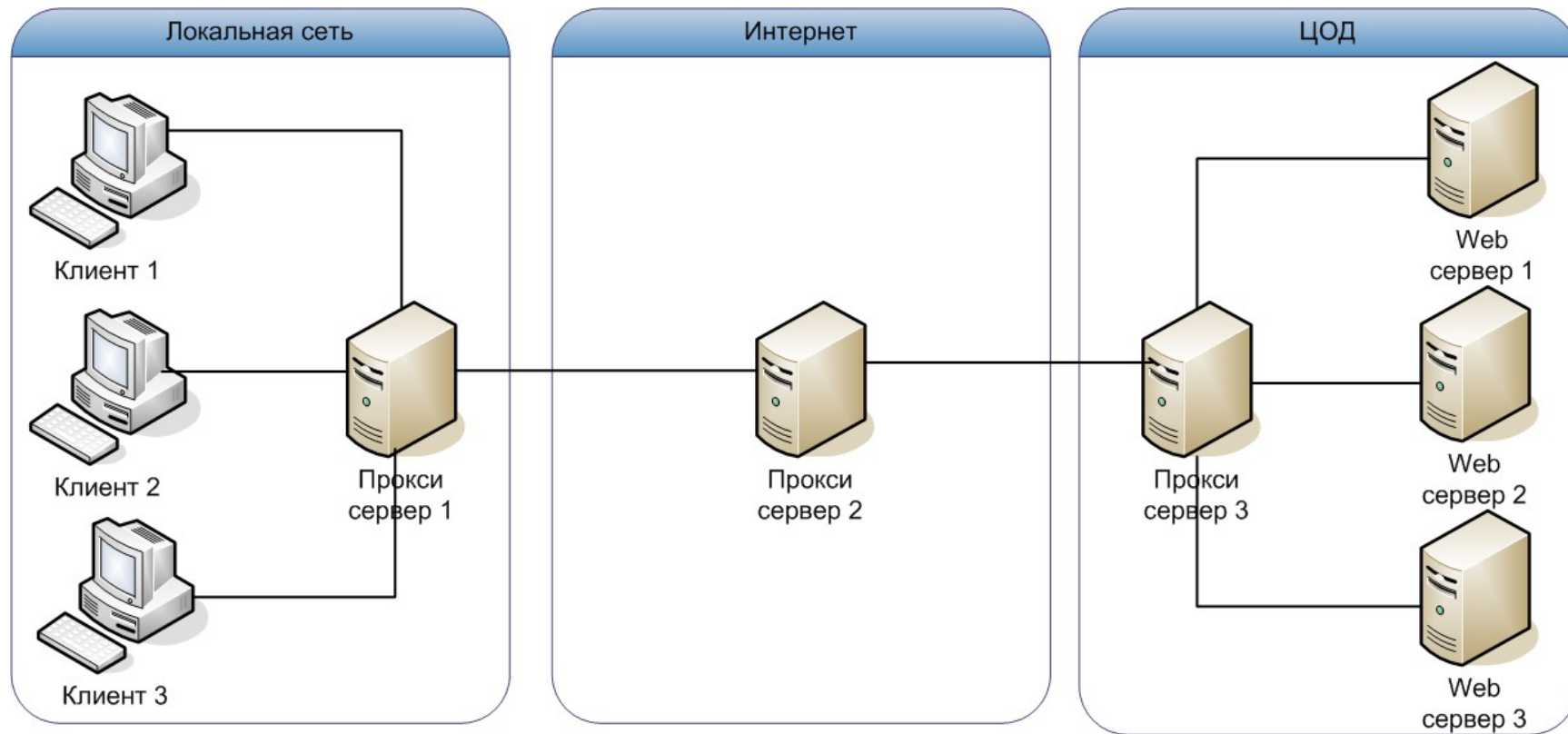
- **Прокси-сервер** –это программа-посредник, позволяющая выполнять запросы к сетевым ресурсам таким образом, что запрос сначала получает прокси, и только затем передаётся целевому ресурсу
- При прохождении запроса через прокси-сервер идентификационные данные клиента *заменяются* на данные прокси-сервера
- Прокси-сервер может *обрабатывать* и *модифицировать* проходящие через него пакеты

Алгоритм обработки запросов:

1. Получение запроса от клиента
2. Извлечение адреса целевого сетевого ресурса
3. Установка соединения и запрос к целевому сетевому ресурсу
4. Получение ответа от целевого сервера и пересылка ответа клиенту



Точки размещения



Функции

- *Локальная сеть*

- Кеширование, сжатие данных
- Применение политик безопасности:
 - Блокировка опасных запросов и ответов
 - Ограничение доступа к ресурсам

- *Интернет*

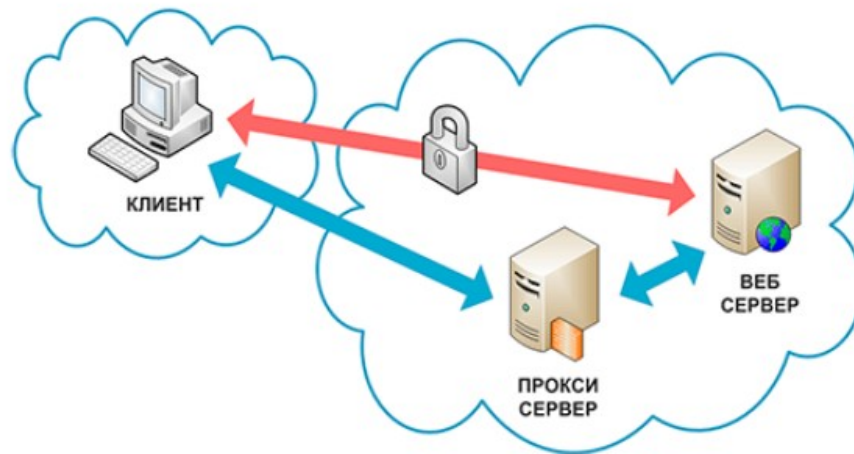
- Соккрытие факта взаимодействия между клиентом и сервером:
 - Соккрытие пользователя от целевого сервера
 - Соккрытие цели доступа от интернет провайдера

- *ЦОД*

- Защита от атак
- Балансировка нагрузки

Наиболее частые причины:

- Обход региональных блокировок
- Автоматизированный сбор информации из открытых источников (*краулинг*)
- Анонимизация



Режимы использования

Прозрачность:

- *Стандартный*
 - Пользователь явно прописывает использование прокси-сервера сетевыми приложениям
- *Прозрачный (transparent)*
 - Пользователь может не знать о наличии прокси-сервера
 - Не требуется дополнительная настройка сетевых программ

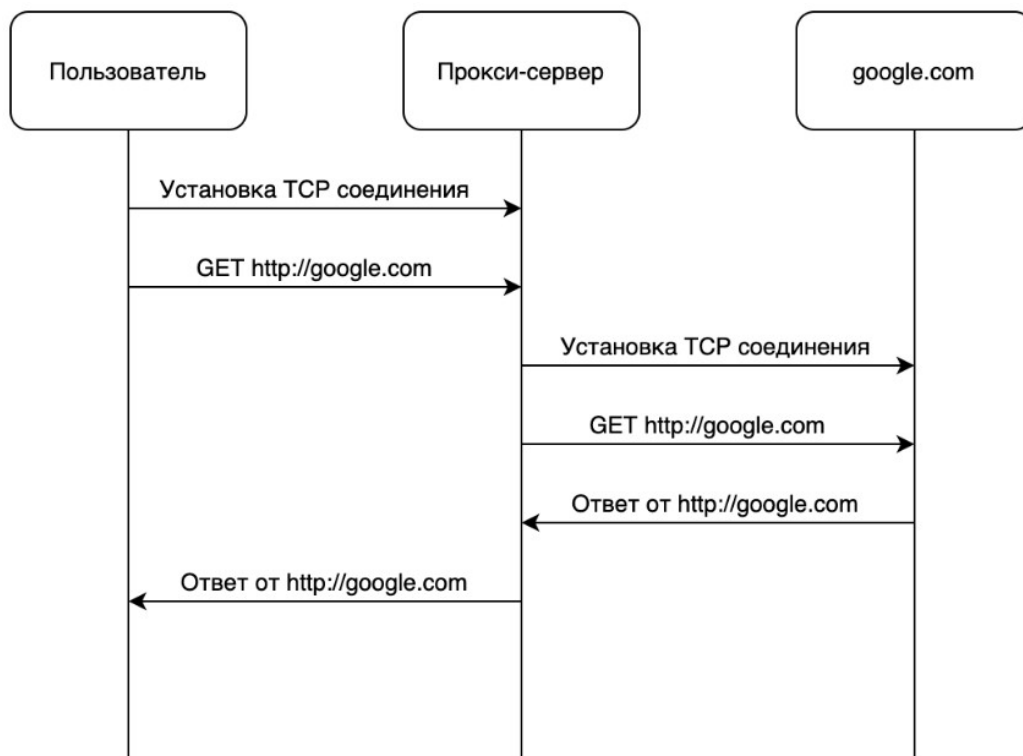
Направление:

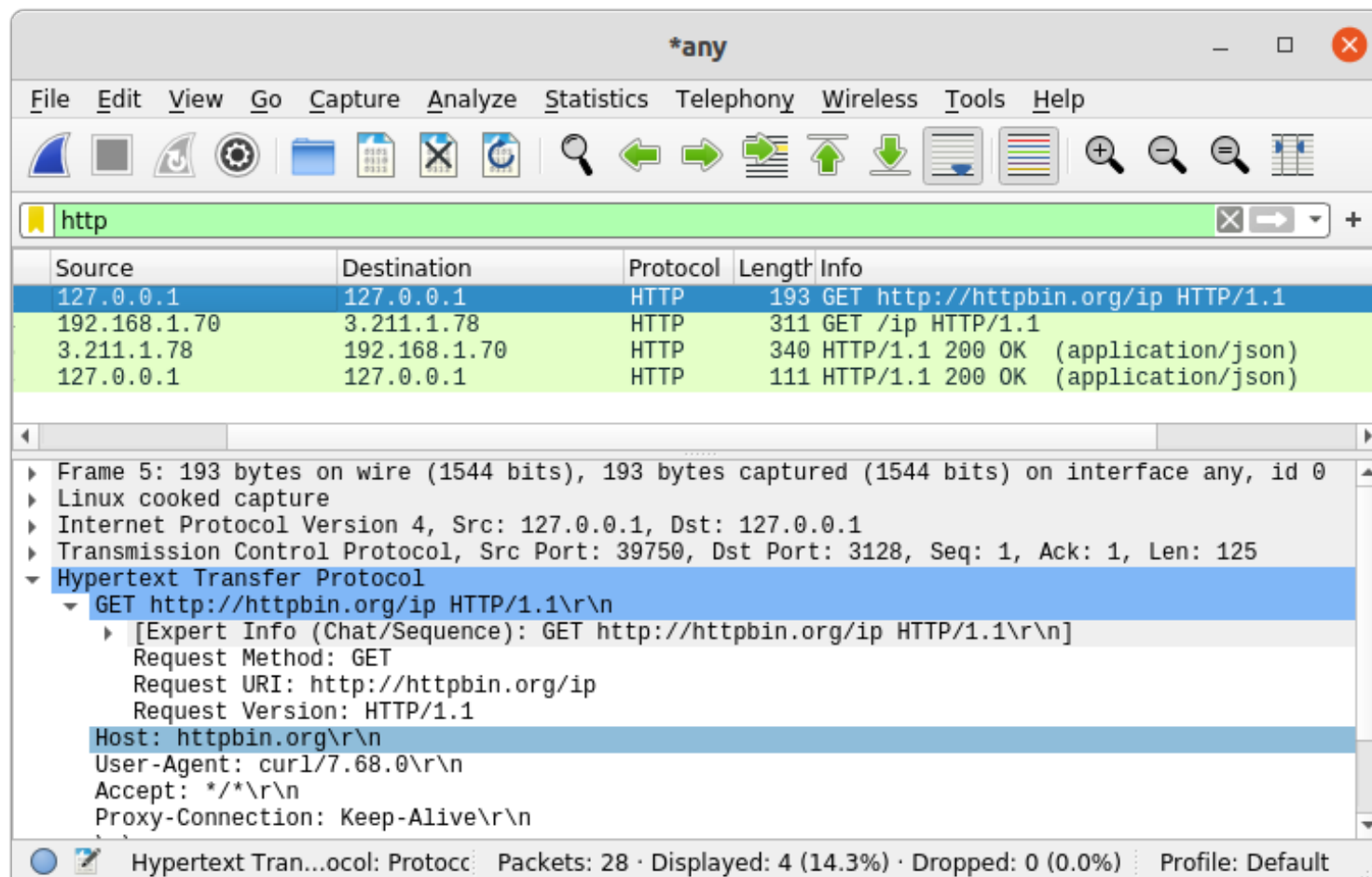
- *Прямой*
 - Прокси-сервер заменяет идентификационные данные клиента
- *Обратный (reverse)*
 - Прокси-сервер заменяет идентификационные данные сервера
 - Обычно обратные прокси-сервера взаимодействуют с набором сокрытых от клиента серверов

Протоколы проксирования

	<i>Протокол проксирования</i>	<i>Поддержка протоколов</i>	<i>Безопасность</i>	<i>Порт</i>
HTTP	Прикладной (текстовый)	HTTP	-	80, 8080, 3128
HTTPS	Прикладной (текстовый)	TCP	TLS	80, 8080, 3128
SOCKS	Сессионный (бинарный)	TCP, UDP (v5)	-	1080, 4145

HTTP





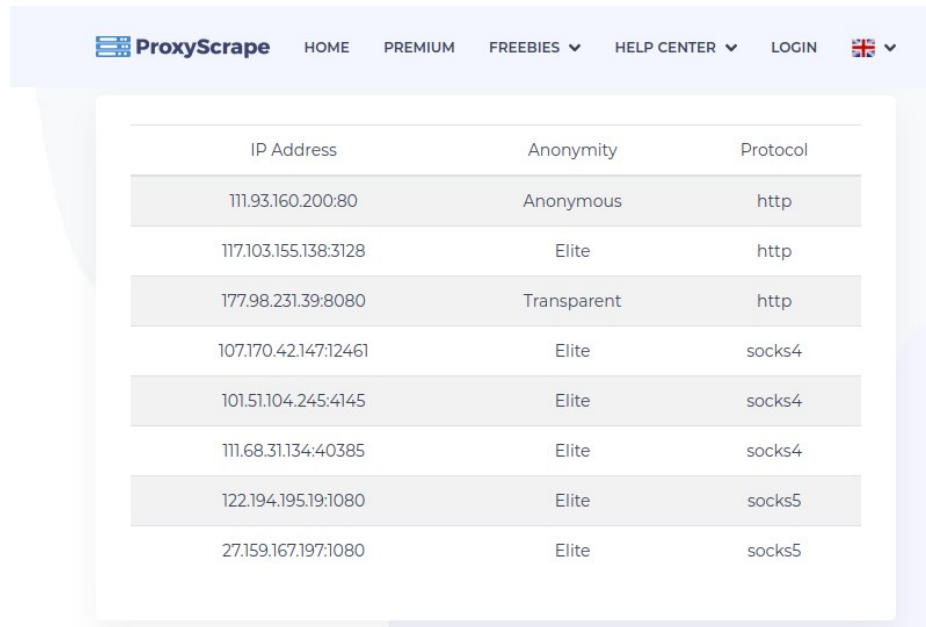
Заголовки HTTP-прокси

- Proxy-Authenticate – метод аутентификации
 - Basic – аутентификация с помощью логина/пароля
 - Bearer – OAuth 2.0 аутентификация с помощью токена доступа
 - Digest – аутентификация по схеме Вызов-Ответ
 - NOBA – аутентификация с использованием цифровой подписи
 - Mutual – аутентификация с использованием пароля
- Forwarded – реальный IP-адрес клиента
- X-Forwarded-For – аналогичен Forwarded (!)
- X-Forwarded-Host – реальный адрес ресурса, от которого пришел ответ (!)
- X-Forwarded-Proto – протокол, используемый для подключения к прокси (!)
- Via – информация о прокси-серверах, через которые прошел запрос

Анонимность прокси-сервера

Категории:

- Transparent – не скрывает для целевого ресурса реальный IP-адрес пользователя
- Anonymous – скрывает реальный IP-адрес пользователя, не скрывает то, что используется прокси-сервер
- Elite – не раскрывает целевому ресурсу факт использования прокси-сервера



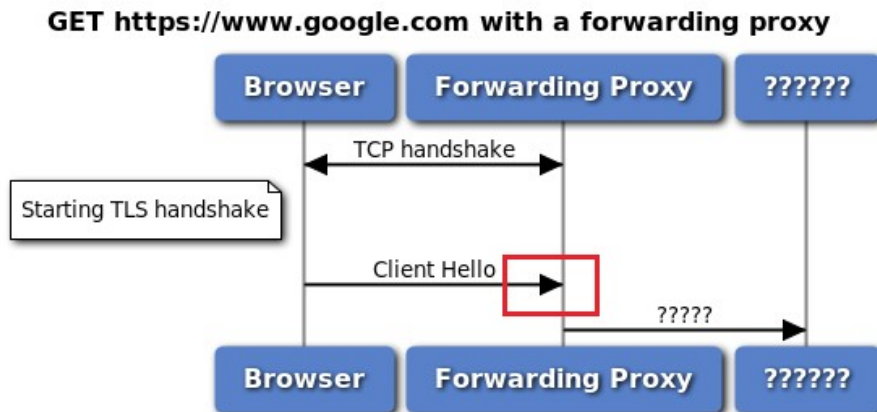
The screenshot shows the ProxyScape website interface. At the top, there is a navigation bar with the ProxyScape logo and links for HOME, PREMIUM, FREEBIES, HELP CENTER, LOGIN, and a language selector (UK flag). Below the navigation bar is a table listing various proxy servers. The table has three columns: IP Address, Anonymity, and Protocol. The data rows are as follows:

IP Address	Anonymity	Protocol
111.93.160.200:80	Anonymous	http
117.103.155.138:3128	Elite	http
177.98.231.39:8080	Transparent	http
107.170.42.147:12461	Elite	socks4
101.51.104.245:4145	Elite	socks4
111.68.31.134:40385	Elite	socks4
122.194.195.19:1080	Elite	socks5
27.159.167.197:1080	Elite	socks5

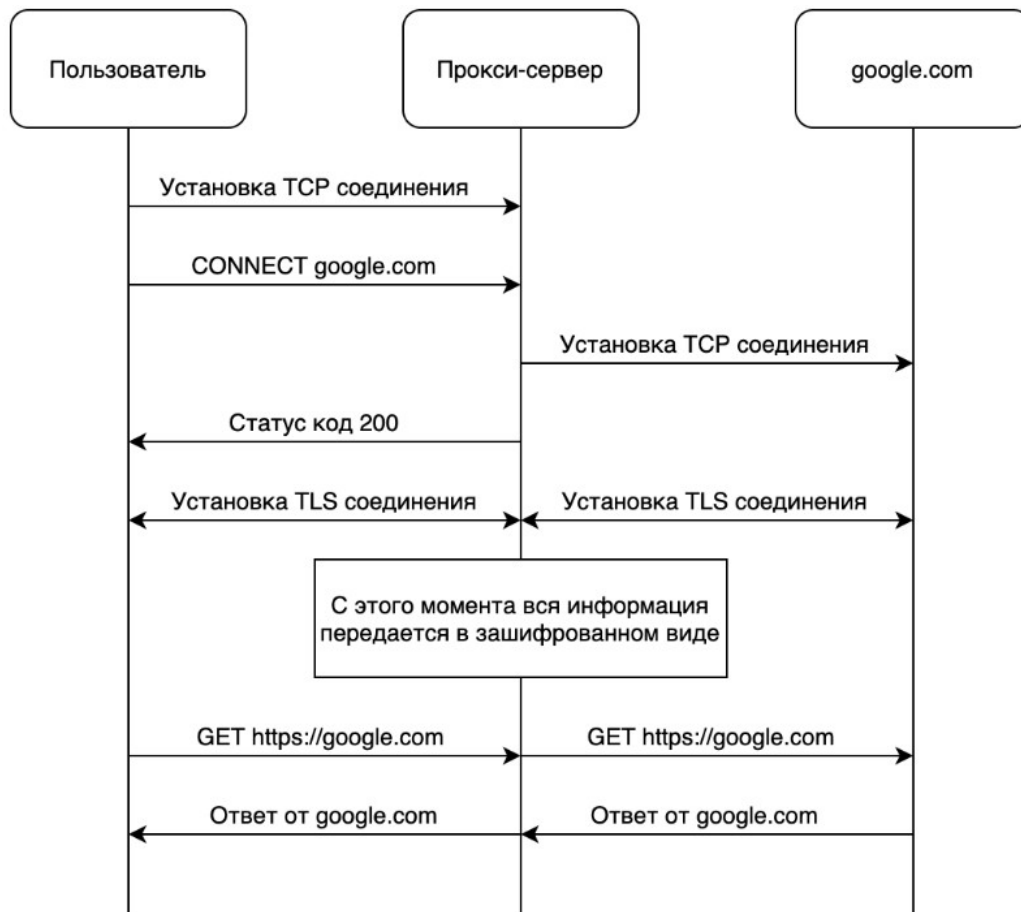
Почему HTTP-прокси не подходят для HTTPS?

Алгоритм инициализации HTTPS:

1. Установление TCP-соединения
 1. SYN
 2. SYN/ACK
 3. ACK
2. Установление TLS-соединения
 1. Отправка бинарного ClientHello пакета
 2. ???



HTTPS



*any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0 or tcp.stream eq 1

Source	Destination	Protocol	Length	Info
127.0.0.1	127.0.0.1	TCP	76	39830 → 3128 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=...
127.0.0.1	127.0.0.1	TCP	76	3128 → 39830 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495...
127.0.0.1	127.0.0.1	TCP	68	39830 → 3128 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=20118023...
127.0.0.1	127.0.0.1	HTTP	182	CONNECT httpbin.org:443 HTTP/1.1
127.0.0.1	127.0.0.1	TCP	68	3128 → 39830 [ACK] Seq=1 Ack=115 Win=65408 Len=0 TSval=201180...
192.168.1.70	18.208.255.250	TCP	76	39142 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 ...
18.208.255.250	192.168.1.70	TCP	76	443 → 39142 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1460 S...
192.168.1.70	18.208.255.250	TCP	68	39142 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=162359325...
127.0.0.1	127.0.0.1	HTTP	107	HTTP/1.1 200 Connection established
127.0.0.1	127.0.0.1	TCP	68	39830 → 3128 [ACK] Seq=115 Ack=40 Win=65536 Len=0 TSval=20118...
127.0.0.1	127.0.0.1	TLSv1.2	585	Client Hello
127.0.0.1	127.0.0.1	TCP	68	3128 → 39830 [ACK] Seq=40 Ack=632 Win=65024 Len=0 TSval=20118...
192.168.1.70	18.208.255.250	TLSv1.2	585	Client Hello
18.208.255.250	192.168.1.70	TCP	68	443 → 39142 [ACK] Seq=1 Ack=518 Win=28160 Len=0 TSval=2630294...
18.208.255.250	192.168.1.70	TLSv1.2	1516	Server Hello
192.168.1.70	18.208.255.250	TCP	68	39142 → 443 [ACK] Seq=518 Ack=1440 Win=62104 Len=0 TSval=1623...

Frame 5: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits) on interface any, id 0

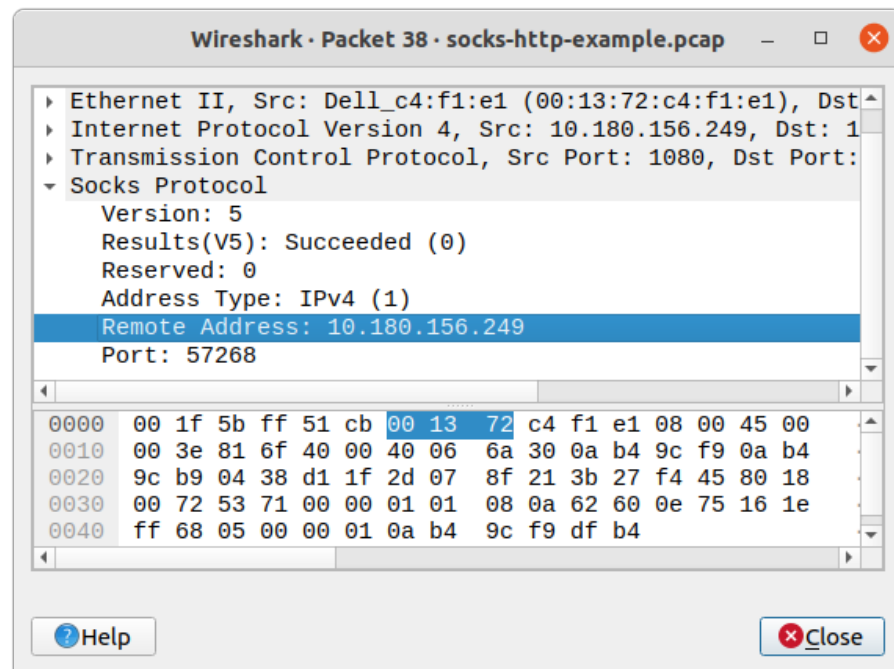
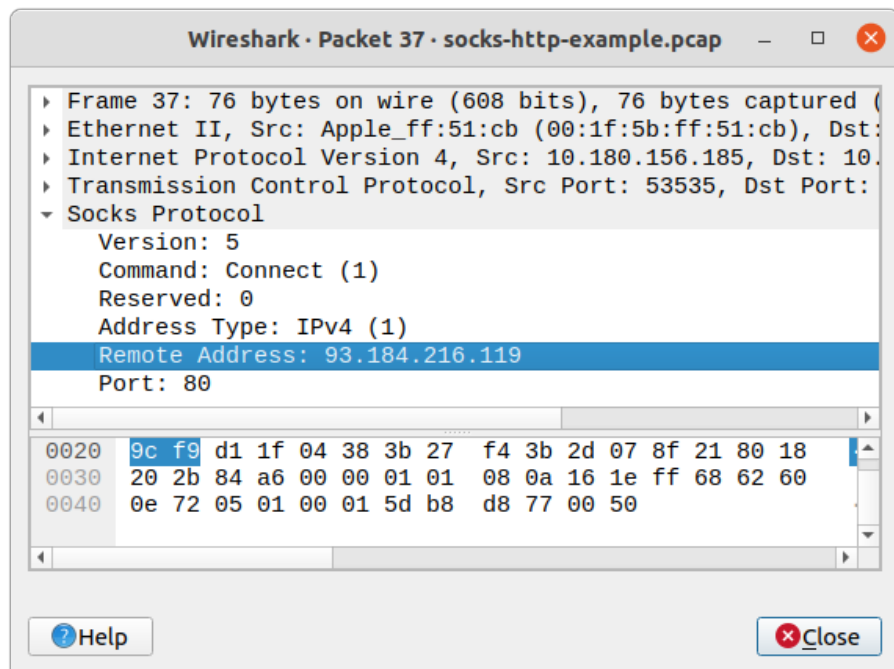
- Linux cooked capture
- Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- Transmission Control Protocol, Src Port: 39830, Dst Port: 3128, Seq: 1, Ack: 1, Len: 114
- Hypertext Transfer Protocol
 - CONNECT httpbin.org:443 HTTP/1.1\r\n
 - [Expert Info (Chat/Sequence): CONNECT httpbin.org:443 HTTP/1.1\r\n]
 - Request Method: CONNECT
 - Request URI: httpbin.org:443
 - Request Version: HTTP/1.1
 - Host: httpbin.org:443\r\n
 - User-Agent: curl/7.68.0\r\n
 - Proxy-Connection: Keep-Alive\r\n
 - \r\n
 - [Full request URI: httpbin.org:443]
 - [HTTP request 1/1]

HTTP Request HTTP-Version (http.request.version), 8 bytes Packets: 80 · Displayed: 67 (83.8%) · Dropped: 0 (0.0%) Profile: Default

SOCKS



Инициализация SOCKS соединения



Прокси-клиент

- Конфигурирование прикладного ПО
 - Параметры прокси-сервера явно указываются в настройках
- Проксификаторы
 - Проксирование осуществляется неявно для сетевого приложения
 - Приложение может не поддерживать функцию проксирования

```
$ curl ident.me
```

```
109.252.32.17
```

```
$ curl --proxy  
socks://204.101.61.82:4145  
ident.me
```

```
204.101.61.82
```

```
$ SOCKS_SERVER=204.101.61.82:4145  
socksify curl ident.me
```

```
204.101.61.82
```

Как работает socksify?

```
$ cat `which socksify`  
...  
LIBRARY="${SOCKS_LIBRARY:-${FULLPATH}libdsocksd.so.0}"  
...  
if test x"$PRELOAD_NAME_ONLY" = x; then  
    LD_PRELOAD="${LIBRARY}${SOCKSIFY_PRELOAD_LIBS:+${PRELOAD_SEPERATOR}}${  
        {SOCKSIFY_PRELOAD_LIBS}${PRELOAD_POSTFIX:+${PRELOAD_SEPERATOR}}${  
        {PRELOAD_POSTFIX}"  
...  
export LD_PRELOAD  
...  
exec "$@"
```

Поддержка прокси в сетевых программах, реализуется при помощи подмены динамических библиотек сетевых функций.

Squid

Многофункциональный прокси-сервер:

- Поддержка HTTP, FTP, Gopher, HTTPS
- Кэширование
- Редирект
- Контроль доступа
 - IP, домен, путь, сертификат, страна и прочее
- Ограничение полосы пропускания
- Режим обратного прокси
 - Балансировка, кэширование
- Прозрачный прокси

Поддержка Windows, Linux, BSD, Solaris

Первая версия появилась в 1996



Практическое задание №3.1

Настроить прокси-сервер Squid и записать сетевые трассы:

- Модификация HTTP-заголовков:
 - <фамилияио>-<группа>-ua.pcapng
- Ограничение доступа к ресурсу:
 - <фамилияио>-<группа>-acl.pcapng

Пример для Иванова И.И. из 123 группы:
ivanovii-123-ua.pcapng

Дефис является разделителем, используйте исключительно латинские символы и цифры.

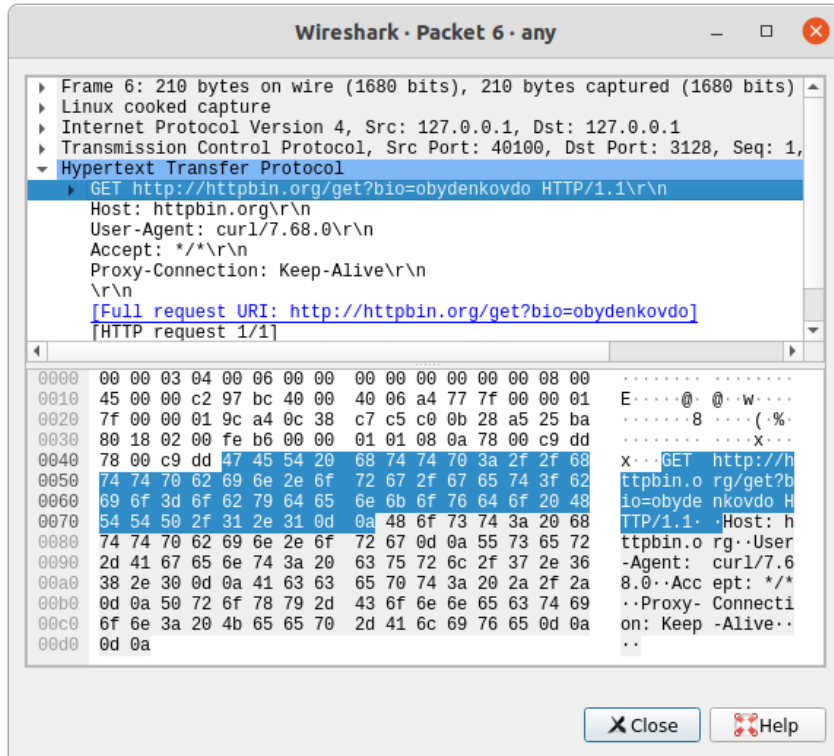
Что?	Два файла в архиве с названием <фамилияио>-<группа>-p3_1.zip
Куда?	insecon@ispras.ru (тема: <вуз>-<группа>-p3_1)
Когда?	Крайний срок 28.11.2022

№3.1 — Ограничение доступа

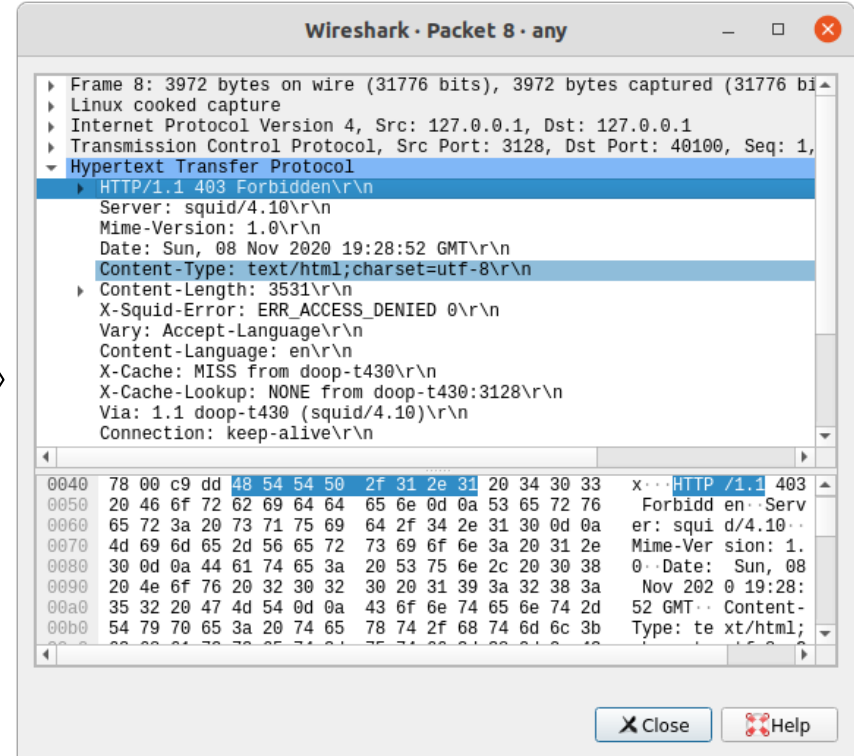
Описание теста с ограничением доступа к ресурсу:

- GET-запрос через локальный Squid-прокси к `ident.me` по HTTP
 - Заголовок `User-Agent`: `<фамилияио>`
 - Прокси-сервер должен запретить доступ к ресурсу и вернуть ошибку
- GET-запрос через локальный Squid-прокси к `httpbin.org/get?bio=<фамилияио>` по HTTP
 - Прокси-сервер должен разрешить доступ к ресурсу
- Сетевая трасса может быть записана утилитами Wireshark, tshark, tcpdump
 - Материалы по Wireshark [1], [2]
- В сетевой трассе должны присутствовать:
 - TCP пакеты между браузером, прокси-сервером и сетевым ресурсом
 - Все прочие пакеты должны быть отфильтрованы
- Метки времени должны быть корректны и отражать реальный порядок пакетов
- TCP-соединения должны присутствовать целиком — с TCP handshake

Ограничение доступа



Squid



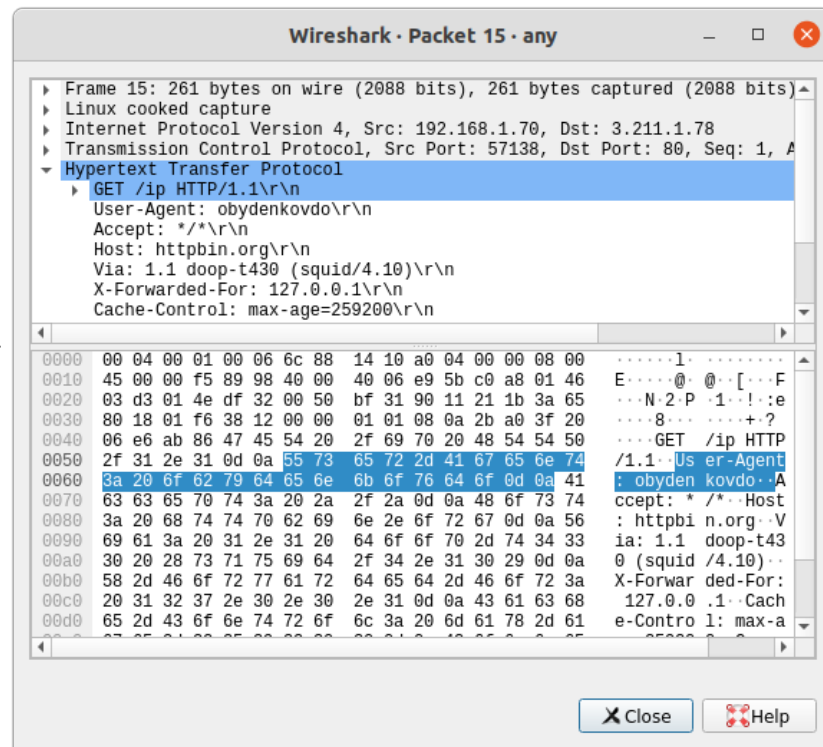
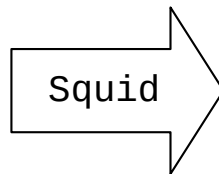
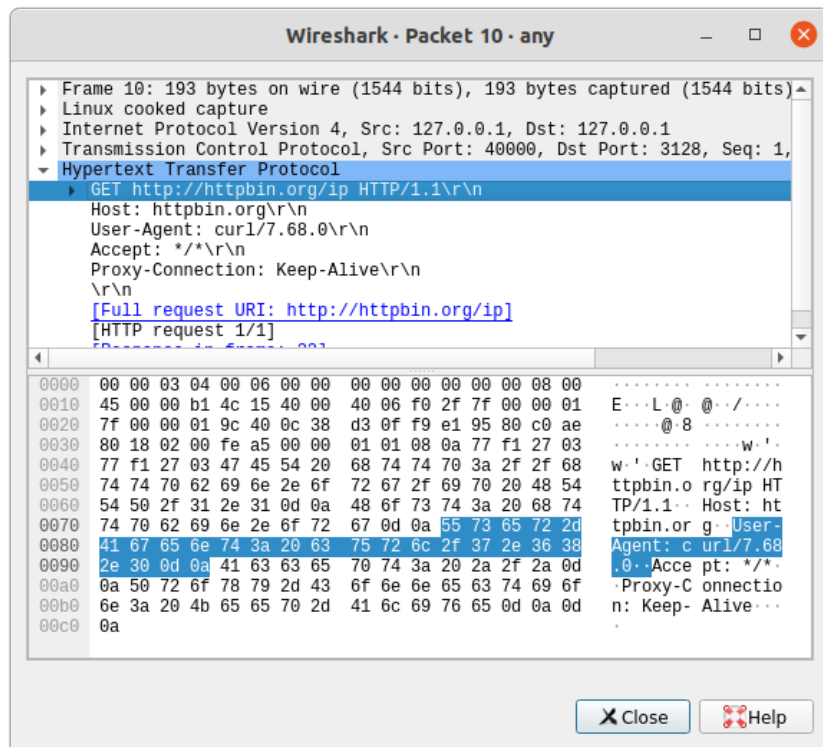
№3.1 — Модификация заголовка

Описание теста с модификацией HTTP-заголовков:

- GET-запрос к ресурсу `httpbin.org/ip` должен пройти через HTTP локальный прокси-сервер Squid
- Прокси сервер Squid должен заменить заголовок `User-Agent` на `<фамилияио>`

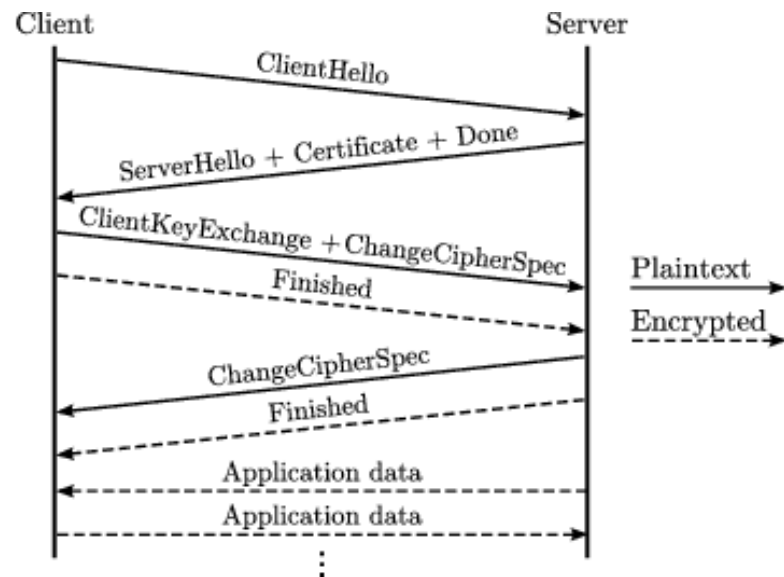
- Сетевая трасса может быть записана утилитами Wireshark, tshark, tcpdump
 - Материалы по Wireshark [\[1\]](#), [\[2\]](#)
- В сетевой трассе должны присутствовать:
 - TCP пакеты между браузером, прокси-сервером и сетевым ресурсом
 - Все прочие пакеты должны быть отфильтрованы
- Метки времени должны быть корректны и отражать реальный порядок пакетов
- TCP-соединения должны присутствовать целиком — с TCP handshake

Модификация заголовка



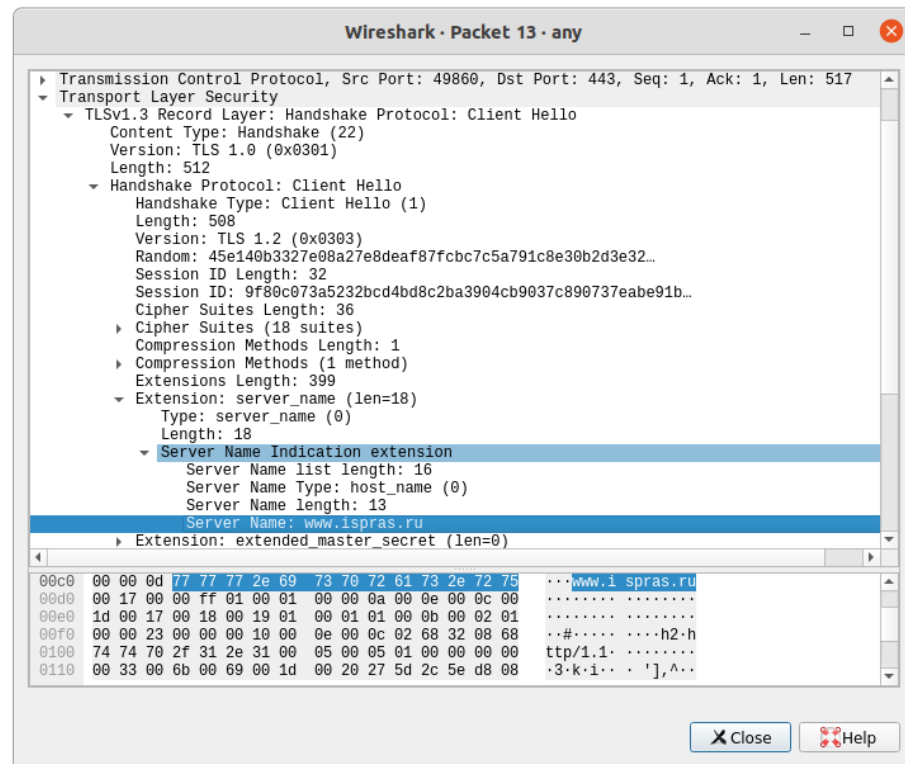
Блокировка HTTPS-трафика

- HTTPS-трафик зашифрован, нельзя просматривать заголовки пакетов
- Блокировка доступа для HTTPS:
 - IP-адреса — для обхода блокировки ресурсу достаточно сменить IP адрес
 - DNS-запросы — требуется перехват всех запросов или контролировать все DNS-сервера
 - Расширение SNI — далее



Server Name Indication

- **SNI** — *опциональное* расширение протокола TLS, позволяющее клиентам сообщать имя хоста, с которыми он желает соединиться
 - Позволяет серверу обслуживать несколько сетевых ресурсов с различными сертификатами по одному адресу <IP>:<Port>
 - Требуется поддержка со стороны браузера клиента, иначе передаётся дефолтный сертификат
- Расширение передавалось в открытом виде до TLS 1.3
 - В TLS 1.3 расширения могут быть зашифрованы ключом, получаемым через DNS (Encrypted SNI — ESNI)
 - С октября 2020 года в России некоторые провайдеры блокируют ESNI и TLS 1.3 трафик



Практическое задание №3.2

Настроить прокси-сервер Squid и записать сетевые трассы и SSLKEYLOG:

- Ограничение доступа к ресурсу:

- <фамилияио>-<группа>-acl.pcapng
- <фамилияио>-<группа>-acl.log
- <фамилияио>-<группа>-acl.conf

- Перехват HTTPS-трафика:

- <фамилияио>-<группа>-bump.pcapng
- <фамилияио>-<группа>-bump.log
- <фамилияио>-<группа>-bump.conf

- Сертификат для Squid:

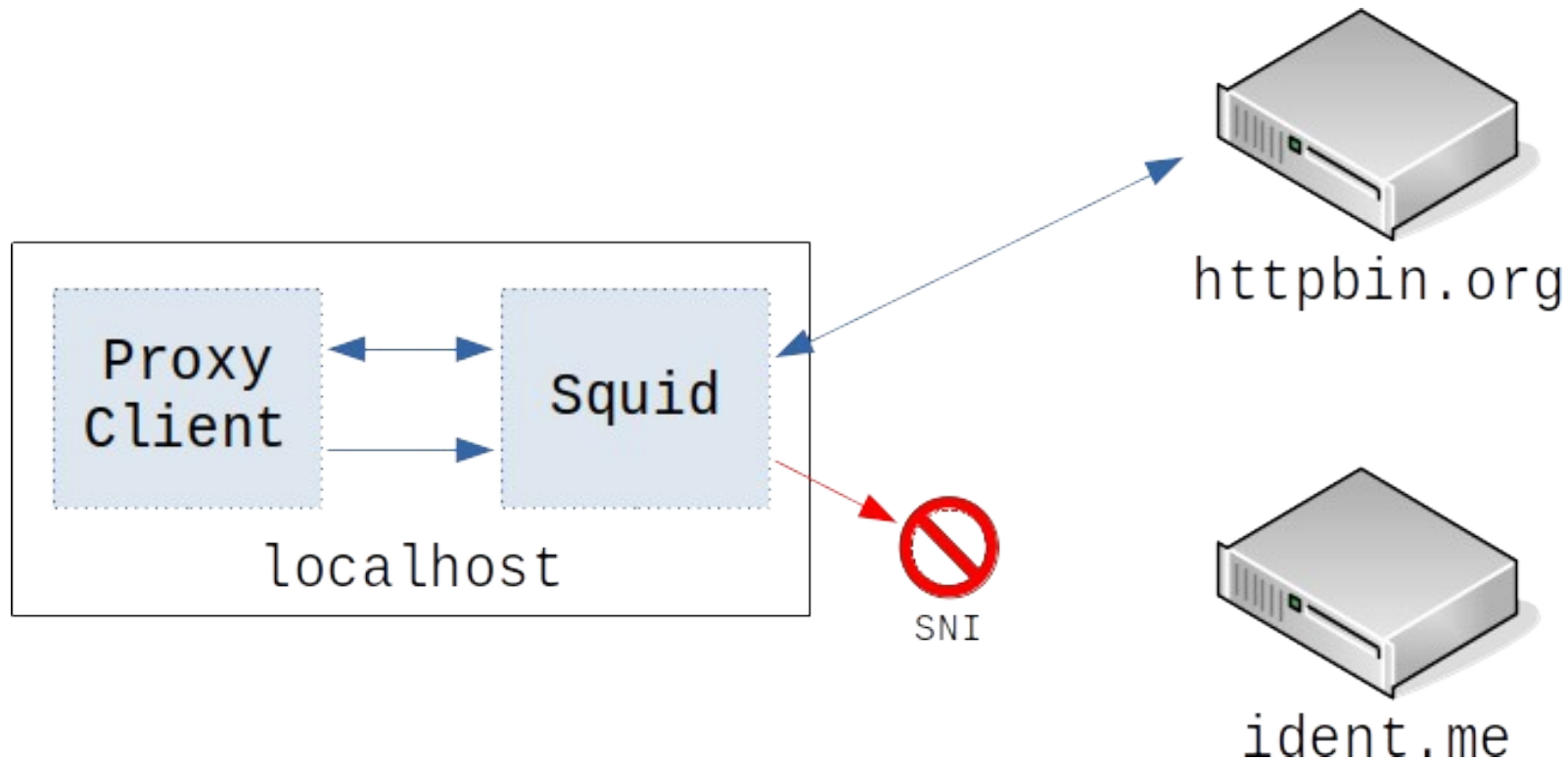
- <фамилияио>-<группа>-bump.crt
- <фамилияио>-<группа>-bump.key
- <фамилияио>-<группа>-ca.crt

Пример для Иванова И.И. из 123 группы:
ivanovii-123-ua.pcapng

Дефис является разделителем, используйте исключительно латинские символы и цифры.

Что?	Девять файлов в архиве с названием <фамилияио>-<группа>-p3_2.zip
Куда?	insecon@ispras.ru (тема: <вуз>-<группа>-p3_2)
Когда?	Крайний срок 05.12.2022

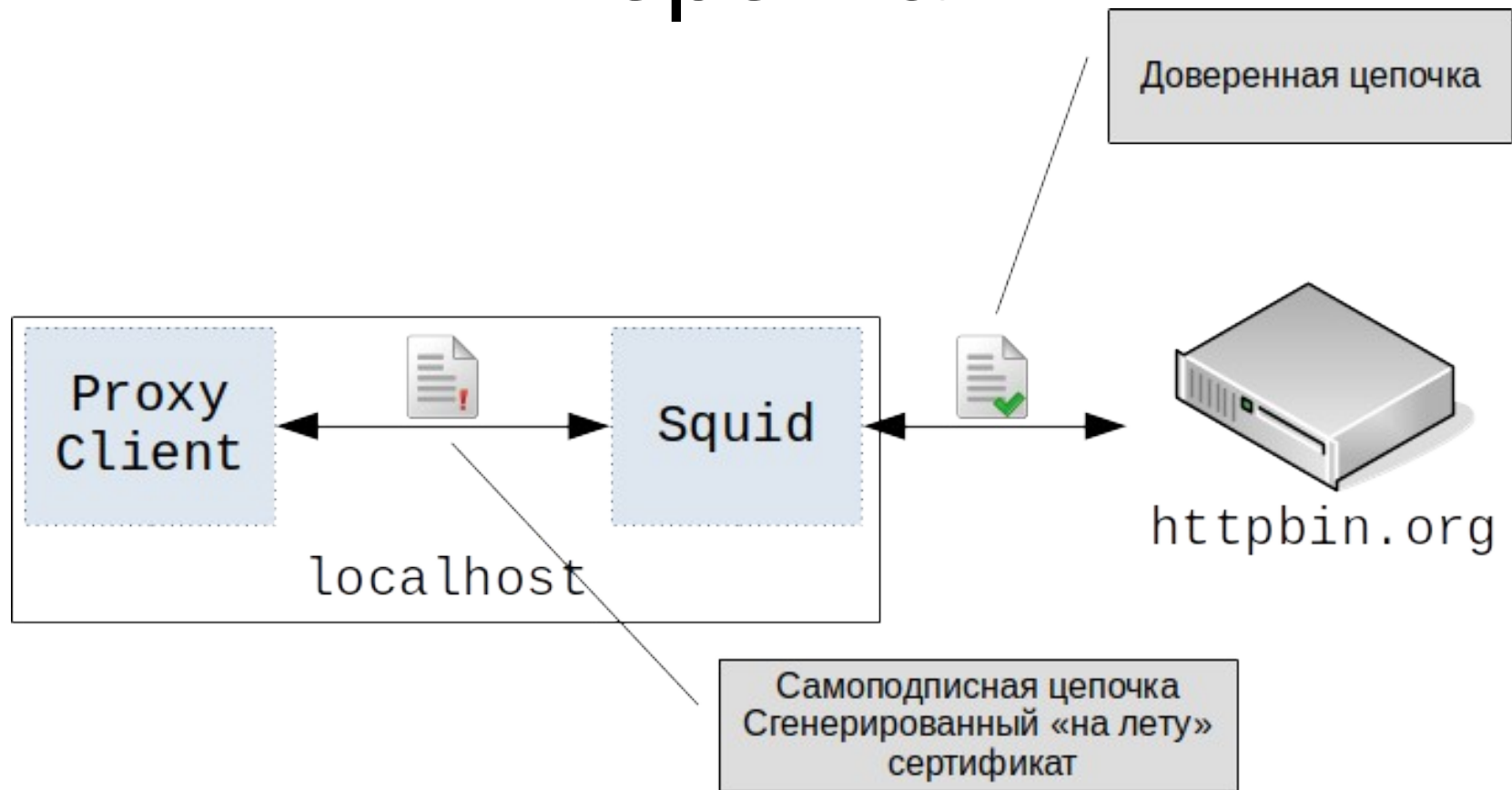
Ограничение доступа



Ограничение доступа

1. Клиент отправляет HTTP CONNECT прокси-серверу
2. Прокси-сервер устанавливает TCP-соединение с ресурсом
(или после этапа 3)
3. Прокси-сервер получает ClientHello с SNI от клиента
4. Прокси-сервер применяет политики ACL
 - Клиент устанавливает TLS соединение с целевым ресурсом
(разрешено)
 - Клиент получает от прокси-сервера TCP-FIN (запрещено)

Перехват



Перехват

1. Клиент отправляет HTTP CONNECT прокси-серверу
2. Прокси-сервер устанавливает TCP-соединение с ресурсом
(или после этапа 3)
3. Прокси-сервер получает ClientHello с SNI от клиента
4. Прокси-сервер устанавливает TLS-соединение с целевым ресурсом
5. Прокси-сервер генерирует сертификат для данного целевого ресурса, подписанный нашим сертификатом
6. Прокси-сервер устанавливает TLS-соединение с клиентом, используя сгенерированный сертификат

№3.2 - Подробности

- Описание теста с ограничением доступа:
 - Запрос через локальный Squid-прокси к `ident.me` по HTTPS
 - Squid закрывает соединение после получения ClientHello с SNI
 - Запрос через локальный Squid-прокси к `httpbin.org/get?bio=<фамилияио>` по HTTPS
 - Соединение успешно установлено
- Описание теста с перехватом HTTPS:
 - Запрос через локальный Squid-прокси к `httpbin.org/get?bio=<фамилияио>` по HTTPS
 - Соединение успешно установлено, ответ от ресурса получен
 - Squid осуществляет подмену сертификата сервера на сгенерированный
- Сетевая трасса может быть записана утилитами Wireshark, tshark, tcpdump
 - Материалы по Wireshark [1], [2]
- В сетевой трассе должны присутствовать:
 - TCP пакеты между браузером, прокси-сервером и сетевым ресурсом
 - Все прочие пакеты должны быть отфильтрованы
- Метки времени должны быть корректны и отражать реальный порядок пакетов
- TCP-соединения должны присутствовать целиком — с TCP и TLS handshake

№3.2 - Сертификат

- Ключевая пара:
 - RSA 4096 бит;
- Сертификат
 - Подписан корневым сертификатом (рекомендуется использовать из задания P1.1);
 - Срок действия 1 год;
 - C=RU, ST=Moscow, L=Moscow, O=<фамилия-ио>, OU=<фамилия-ио> P3_2, CN=<фамилия-ио> Squid CA, email=<адрес вашей почты>;
 - X.509 v3 расширения:
 - *Basic Constrains*:
 - Critical
 - PathLen=0
 - CA=True
 - *Key Usage*:
 - Critical
 - Digital Signature
 - Certificate Sign
 - CRL sign.

Как отфильтровать трассу?

- Отбираем нужные TCP-потоки по номерам:
 - Transmission Control Protocol
 - [Stream Index]
- Фильтруем:
 - `tcp.sream == X` or `..` or `tcp.stream == Y`
- Сохраняем:
 - File – Export Specified Packets [All packets + Displayed]

