CTF на Физтехе

Занятие 8

Веб: HTTP

Стек TCP/IP

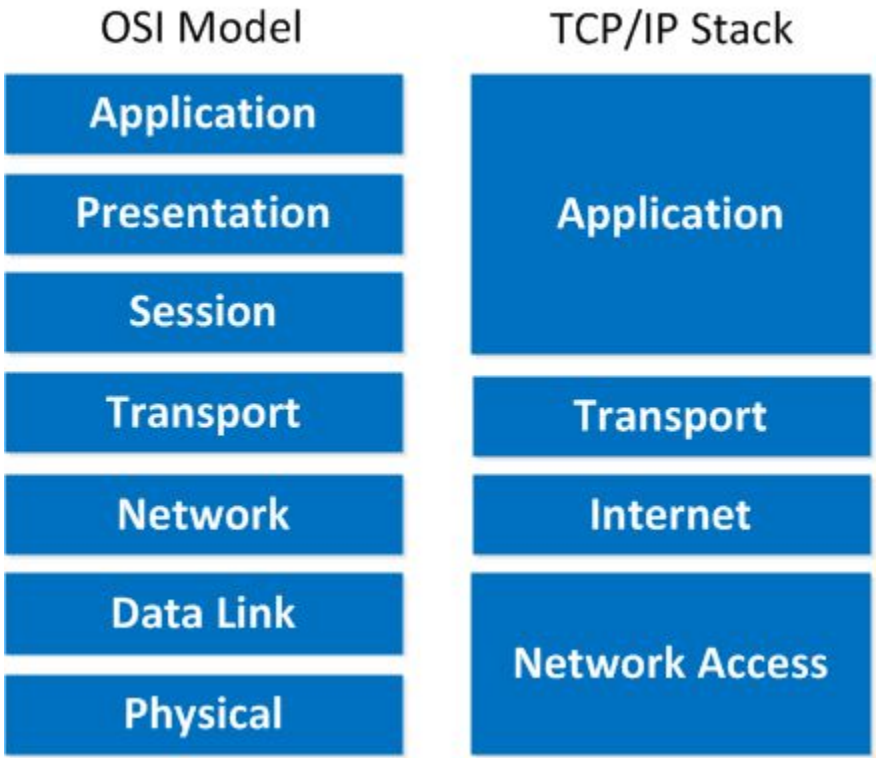
Модель OSI

Слой (Layer)	Название	Протоколы
7	Прикладной уровень Application layer	HTTP, ...
6	Уровень представления Presentation layer	
5	Сеансовый уровень Session layer	
4	Транспортный уровень Transport layer	TCP, UDP, ...
3	Сетевой уровень Network layer	IP, ICMP, ...
2	Канальный уровень Data link layer	Ethernet, ...
1	Физический уровень Physical layer	IEEE 802.3, IEEE 802.11, ...

Software

Hardware

Стек TCP/IP


OSI Model
TCP/IP Stack
Application
Presentation
Session
Transport
Network
Data Link
Physical
Application
Transport
Internet
Network Access

Протокол IP

- Протокол 3 уровня (сетевой уровень)
- Объединяет сегменты сети в глобальную сеть (Интернет)
- IP-адрес: уникальный сетевой адрес узла в компьютерной сети
- Версия IPv4
  - Длина IP-адреса 4 байта
  - Пример: 192.168.0.3
- Версия IPv6
  - Длина IP-адреса 16 байт
  - 2001:0db8:85a3:0000:0000:8a2e:0370:7334

Протокол TCP

- Transmission Control Protocol (протокол управления передачей)
- Протокол 4 уровня (транспортный уровень)
- Работает поверх IP
- Устанавливает соединение
- Передает поток данных
- Обеспечивает гарантированную доставку
- Используется: Web, SSH, FTP, SMTP, IMAP/POP, ...

Протокол UDP

- User Datagram Protocol (протокол пользовательских датаграмм)
- Протокол 4 уровня (транспортный уровень)
- Работает поверх IP
- Не обеспечивает гарантированную доставку
- “Быстрее”, чем TCP
- Используется: DNS, Media streaming, Games, Tunneling/VPN, ...

Порт

- Проблема: IP-адрес один, а программ много
- Как отправить данные определенной программе?

- Для установки соединения или отправки данных протоколы TCP и UDP используют понятие порта
- Порт - натуральное число от 1 до 65535
- При открытии соединения отправитель указывает порт, на который он хочет отправить данные, а получатель - на котором хочет их получать

Часто используемые порты

Port	Protocol	Service/Transport
20/21	FTP	TCP
22	SSH	TCP
23	Telnet	TCP
25	SMTP	TCP
53	DNS	TCP/UDP
69	TFTP	UDP
80	HTTP	TCP
110	POP3	TCP
135	RPC	TCP
161/162	SNMP	UDP
1433/1434	MSSQL	TCP

Полезные утилиты

- netcat - утилита для передачи данных по TCP или UDP
  - nc -l 9999
  - nc 127.0.0.1 9999
- nmap - утилита для сканирования портов
  - nmap -F andreyknvl.com
- netstat - утилита для отображения различной сетевой информации
  - sudo netstat -ntlp
- ping - утилита для отправки ICMP ECHO_REQUEST запроса
  - ping google.com

HTTP

HTTP

- Протокол прикладного уровня
- Реализован поверх TCP
- По умолчанию используется 80 порт
- Текстовый протокол
- Технология клиент-сервер => структура: запрос - ответ
- Используется для общения между браузером и веб сервером
- Версии: HTTP/1.1, HTTP/2.0

URL

<схема>://<логин>:<пароль>@<хост>:<порт>/<URL-путь>?<параметры>#<якорь>

Sidenote: URI vs URL vs URN

URL encoding

- Определенные символы в URL должны кодироваться специальным образом, некоторые из них:

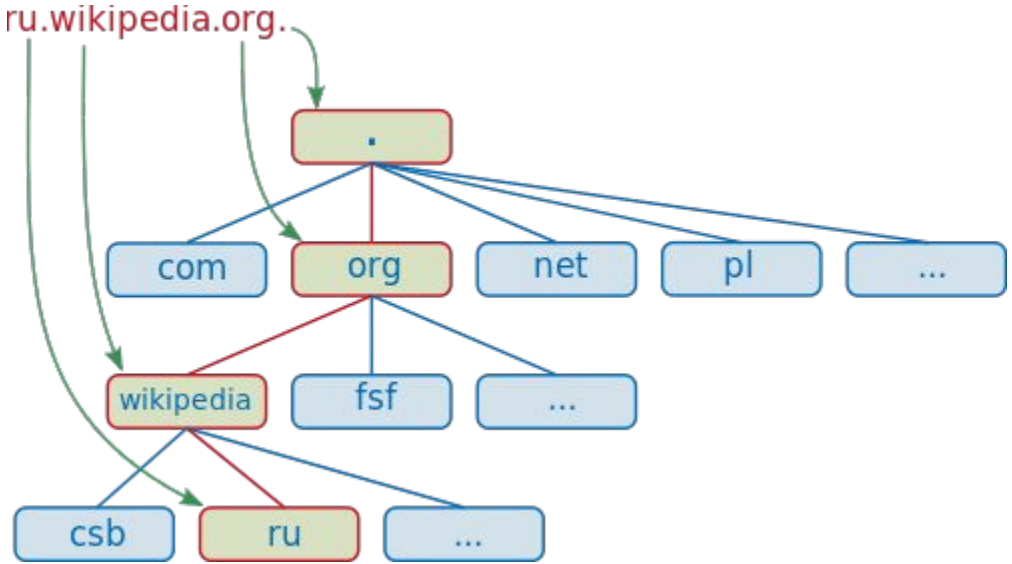
!	#	$	&	'	(	)	*	+	,	/	:	;	=	?	@	[	]
%21	%23	%24	%26	%27	%28	%29	%2A	%2B	%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D

- “http://www.example.com/new pricing.htm”
- http://www.example.com/new%20pricing.htm

- http://www.url-encode-decode.com/

DNS

- Domain Name System (система доменных имен)
- Обычно использует UDP
- Используется для получения IP-адреса по имени хоста



HTTP: структура запроса

Стартовая строка	GET /feed HTTP/1.1
Заголовки	Host: vk.com Content-Type: text/plain; charset=utf-8
Пустая строка	
Тело сообщения	<Data>

HTTP: структура ответа

Стартовая строка	HTTP/1.1 200 OK
Заголовки	Content-Type: text/html; charset=windows-1251 Content-Length: 1000
Пустая строка	
Тело сообщения	<!DOCTYPE html> ...

Demo

HTTP: методы

- GET - используется для запроса содержимого
- HEAD - аналогичен GET, но в ответе отсутствует тело
- POST - используется для передачи данных
- OPTIONS
- PUT, PATCH, DELETE
- TRACE
- CONNECT

HTTP: параметры GET

- Вместе с запросом GET можно передавать параметры
- GET http://google.com/search?q=vk HTTP/1.1

- Параметры отделяются ‘?’ и разделяются ‘&’
- http://domain.org/index.php?field1=value1&field2=value2&field3=value3

- Все параметры закодированы с помощью URL encoding
- http://domain.org/index.php?var=This+is+a+simple+%26+short+test.

HTTP: параметры POST

- Вместе с запросом POST тоже можно передавать параметры
- Параметры передаются в теле запроса
- Параметры разделяются ‘&’ и кодируются URL encoding

```
POST /add_name HTTP/1.1
Host: domain.org
...

fname=John&lname=Smith
```

HTTP: коды состояния

- 1xx Informational (Информационный)
  - 100 Continue
- 2xx Success (Успех)
  - 200 OK
- 3xx Redirection (Перенаправление)
  - 302 Found
- 4xx Client Error (Ошибка клиента)
  - 404 Not Found
- 5xx Server Error (Ошибка сервера)
  - 502 Bad Gateway

HTTP: заголовки

- <Имя>: <Значение>
- Host: vk.com
- Referer: http://google.com/search?q=vk
- User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20120101 Firefox/33.0
- Content-Type: text/plain; charset=utf-8
- Accept-Encoding: gzip, deflate
- Connection: keep-alive
- ...

Demo

HTTP: авторизация

- HTTP поддерживает авторизацию
- http://natas0.natas.labs.overthewire.org

- Basic access authentication
  - WWW-Authenticate: Basic realm="nmrs_m7VKmomQ2YM3:"
  - Authorization: Basic QWxhZGRpbjpPcGVuU2VzYW1l
  - http://natas0:natas0@natas0.natas.labs.overthewire.org
- Digest access authentication

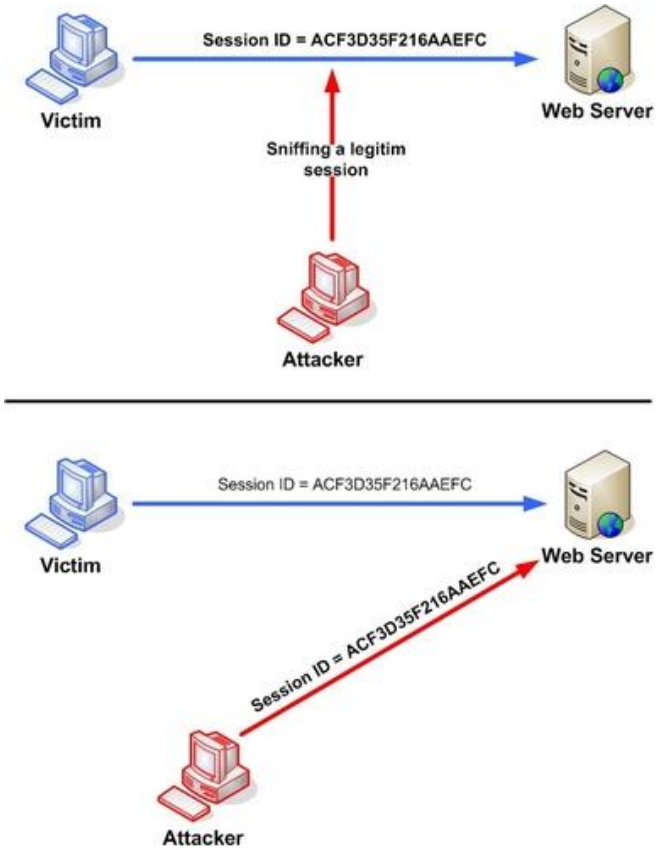
Demo

HTTP: Cookie

- Небольшой фрагмент данных
- Отправляется веб-сервером в ответ на запрос
  - Set-Cookie: name=value
- Сохраняется у пользователя и передается с каждым запросом
  - Cookie: name=value
- Используется для:
  - аутентификации пользователя
  - хранения персональных настроек пользователя
  - ...

Demo

Session hijacking attack



HTTP vs HTTPS

- Протокол HTTP
  - Трафик передается в открытом виде
  - Возможен Man-In-The-Middle

- Протокол HTTPS
  - HTTPS = HTTP Secure = HTTP + TLS (SSL)
  - Обеспечивает шифрование трафика
  - Обеспечивает аутентификацию сервера

HTML

- HTML == HyperText Markup Language (язык гипертекстовой разметки)
- Веб-сервер отдает браузеру описание страницы на языке HTML
- Браузер парсит HTML и отрисовывает страницу графически
- Web frontend == HTML + CSS + JavaScript

Demo

HTML: формы

```
<form action="/add_name">

  First name: <input type="text" name="fname"><br>

  Last name: <input type="text" name="lname"><br>

  <input type="submit" value="Submit">

</form>
```

```
POST /add_name HTTP/1.1

Host: domain.org

...


fname=John&lname=Smith
```

Полезные консольные утилиты

- curl - сделать запрос к URL
  - Copy as cURL в development tools
- wget - скачать файл по URL
- lynx - консольный браузер
- tcpdump - слушать трафик

Python requests

```
import requests

r = requests.get(“http://example.com/ex.php?p1=c1&p2=c2”)

print r.status_code

print r.text


import requests

payload = {“p1” : “c1”, “p2” : “c2”}

r = requests.post(“http://example.com/ex.php”, data=payload)

print r.status_code

print r.text
```

Инструменты

- Development tools (Chrome, Firefox)
- Firebug (Chrome, Firefox)
- EditThisCookie (Chrome)
- Tamper Data (Firefox)

- Burp Suite

- Wireshark

Вопросы?