**КОНСПЕКТ 2**

***1. HTTP***

HTTP (Hypertext Transfer Protocol) — это протокол для передачи данных в интернете, который используется для обмена информацией между клиентами (например, веб-браузерами) и серверами.

**1. Клиент-серверная архитектура:**

Клиент отправляет запросы к серверу, который обрабатывает их и возвращает ответы.

**2. Основные методы HTTP включают:**

* GET: Запрашивает данные.
* POST: Отправляет данные на сервер.
* PUT: Обновляет ресурс.
* DELETE: Удаляет ресурс.

**3. Статусные коды: Используются для информирования клиента о результате запроса, например:**

* 200 OK: Успешный запрос.
* 404 Not Found: Ресурс не найден.
* 500 Internal Server Error: Ошибка на сервере.

**4. Безопасность:**

HTTP сам по себе не обеспечивает безопасность. Для защиты данных используется HTTPS (HTTP Secure), который добавляет уровень шифрования с помощью SSL/TLS.

**5. Cтруктура HTTP-запроса:**

Запрос состоит из:

* Метода (например, GET).
* URL (адрес ресурса).
* Версии протокола (например, HTTP/1.1).
* Заголовков (дополнительная информация).
* Тела запроса (опционально, например, при POST).

### **6. Структура HTTP-ответа:**

Ответ включает:

* Версию протокола.
* Статус-код (например, 200).
* Статус-фразу (например, OK).
* Заголовки (дополнительная информация).
* Тело ответа (например, HTML-код).

### **7. Заключение:**

HTTP — это основной протокол для обмена данными в интернете, позволяющий клиентам и серверам взаимодействовать друг с другом.

***2. SSL и TLS***

SSL (Secure Sockets Layer) и TLS (Transport Layer Security) — это протоколы для обеспечения безопасности передачи данных в интернете.

### **SSL:**

* **Разработка**: Создан в 1990-х годах компанией Netscape.
* **Версии**: Включает SSL 1.0, 2.0 и 3.0, которые больше не считаются безопасными.
* **Функции**: Обеспечивает шифрование, аутентификацию и целостность данных.

### **TLS:**

* **Разработка**: Преемник SSL, впервые представлен в 1999 году (TLS 1.0).
* **Улучшения**: Более безопасные алгоритмы шифрования и методы аутентификации.
* **Функции**: Также обеспечивает шифрование, аутентификацию и целостность данных, но более эффективно.

### **Основные отличия:**

* **Безопасность**: TLS более безопасен, чем SSL.
* **Алгоритмы**: TLS поддерживает современные алгоритмы шифрования.
* **Рукопожатие**: Процесс установления соединения в TLS более сложен и безопасен.

**Заключение**: TLS является стандартом для безопасной передачи данных в интернете, в то время как SSL устарел и не рекомендуется к использованию.

***3. Отличия HTTP и HTTPS***

HTTPS (порт 443) является более защищённым протоколом, в отличии от HTTP (порт 80). Он использует уровни шифрования SSL и TLS.

Но существует ряд причин, по которым HTTP до сих пор используется.

1. Протокол является производительней и проще в использовании из-за отсутствия необходимости шифрования и дешифрования данных.
2. Он может не поддерживаться на старых серверах.
3. Удобнее и экономнее в случаях, когда нужно работать с неважной информацией.
4. Используется для локальных тестов.

### **4. IP (Internet Protocol):**

#### IP — протокол, который используется для передачи данных по сети. Он отвечает за адресацию и маршрутизацию пакетов данных между устройствами в интернете.

**1. Версии**: Существуют две основные версии IP:

* **IPv4**: Использует 32-битные адреса, что позволяет создать около 4,3 миллиарда уникальных адресов. В настоящее время исчерпываются адреса IPv4.
* **IPv6**: Использует 128-битные адреса, что обеспечивает значительно большее количество уникальных адресов.

3. **Функции**:

* **Адресация**: Каждое устройство в сети получает уникальный IP-адрес, который используется для идентификации и связи.
* **Маршрутизация**: IP определяет, как пакеты данных перемещаются от источника к получателю через различные маршрутизаторы.

### **5. IPS (Intrusion Prevention System):**

IPS — система, предназначенная для обнаружения и предотвращения атак на компьютерные сети и системы.

**ФУНКЦИИ:**

* **Мониторинг трафика**: IPS анализирует сетевой трафик в реальном времени для выявления подозрительной активности.
* **Предотвращение атак**: При обнаружении угроз IPS может автоматически блокировать вредоносный трафик или предпринимать другие меры для защиты сети.
* **Логирование и отчетность**: IPS ведет журналы событий и предоставляет отчеты о безопасности, что помогает в анализе инцидентов.

3. **Типы**:

* **Сетевые IPS (NIPS)**: Устанавливаются на уровне сети и защищают всю сеть.
* **Хостовые IPS (HIPS)**: Устанавливаются на отдельных устройствах и защищают их от атак.

### **Заключение:**

IP — это протокол, который обеспечивает адресацию и маршрутизацию данных в сети, в то время как IPS — это система безопасности, предназначенная для обнаружения и предотвращения атак на сети и системы. Оба элемента играют важную роль в функционировании и защите современных компьютерных сетей.

**6. TCP\UDP\RPT\SCTP\FTP**

### **1. TCP (Transmission Control Protocol)**

#### **Назначение:**

#### Обеспечивает надежную, ориентированную на соединение передачу данных.

#### **Функции:**

* Установление соединения между клиентом и сервером (трехстороннее рукопожатие).
* Гарантия доставки данных (пересылка потерянных пакетов).
* Упорядочивание пакетов (гарантия правильного порядка).
* Контроль потока и управление перегрузками.

**Принцип работы:**

TCP разбивает данные на сегменты, добавляет заголовки с информацией о последовательности и отправляет их. Получатель отправляет подтверждения (ACK) о получении сегментов.

### **2. UDP (User Datagram Protocol)**

**Назначение**:

* Обеспечивает быструю, ненадежную передачу данных без установления соединения.

**Функции**:

* Нет гарантии доставки (пакеты могут быть потеряны).
* Нет упорядочивания пакетов.
* Меньше накладных расходов по сравнению с TCP.

**Как работает**:

* UDP отправляет данные в виде датаграмм без предварительного установления соединения. Каждая датаграмма содержит заголовок с адресом отправителя и получателя.

### **3. RPT (Reliable Multicast Transport Protocol)**

**Назначение**:

* Обеспечивает надежную передачу данных в многоточечном режиме (multicast).

**Функции**:

* Гарантия доставки данных нескольким получателям.
* Управление потерей пакетов и их повторной передачей.

**Как работает**:

* RPT использует механизмы, аналогичные TCP, для обеспечения надежности, но оптимизирован для многоточечной передачи, что позволяет эффективно рассылать данные нескольким получателям одновременно.

### **4. SCTP (Stream Control Transmission Protocol)**

**Назначение**:

* Обеспечивает надежную передачу данных с поддержкой многопоточности.

**Функции**:

* Поддержка нескольких потоков данных в одном соединении.
* Гарантия доставки и упорядочивания данных.
* Защита от атак "человек посередине".

**Как работает**:

* SCTP устанавливает соединение с помощью четырехстороннего рукопожатия и использует блоки данных, называемые "чемоданами" (chunks), для передачи информации. Каждый поток может передавать данные независимо, что позволяет избежать блокировок.

### **5. FTP (File Transfer Protocol)**

**Назначение**:

* Протокол для передачи файлов между клиентом и сервером.

**Функции**:

* Позволяет загружать и скачивать файлы.
* Поддерживает аутентификацию пользователей.
* Позволяет управлять файлами на сервере (например, удалять, переименовывать).

**Как работает**:

* FTP использует два канала: один для команд (управляющий канал) и другой для передачи данных. Обычно используется TCP для надежной передачи. Клиент устанавливает соединение с сервером, отправляет команды и получает файлы.

### **Заключение:**

Каждый из этих протоколов выполняет свою уникальную роль в сетевой коммуникации. TCP и UDP обеспечивают транспортировку данных, RPT и SCTP предлагают дополнительные функции для многоточечной передачи и потоков, а FTP предназначен для передачи файлов. Выбор протокола зависит от требований к надежности, скорости и типу передаваемых данных.