Защита информации в сети

Введение

доц. Нестеренко В.А.

Содержание курса

- Вспомогательный материал (сети, адресация, протоколы, …)
- Сбор данных потока пакетов в сети (tcpdump, wireshark).
- Обработка и первичный анализ собранной информации (работа с dump файлами pcap).
- Анализ собранной информации. Методы Data Mining. Выявление аномалий в потоке пакетов сети.
- Модель обнаружения нарушений.
- Программные средства обеспечения безопасности в сети (nmap, portsentry, iptables, snort).

Выявление нарушений в системе

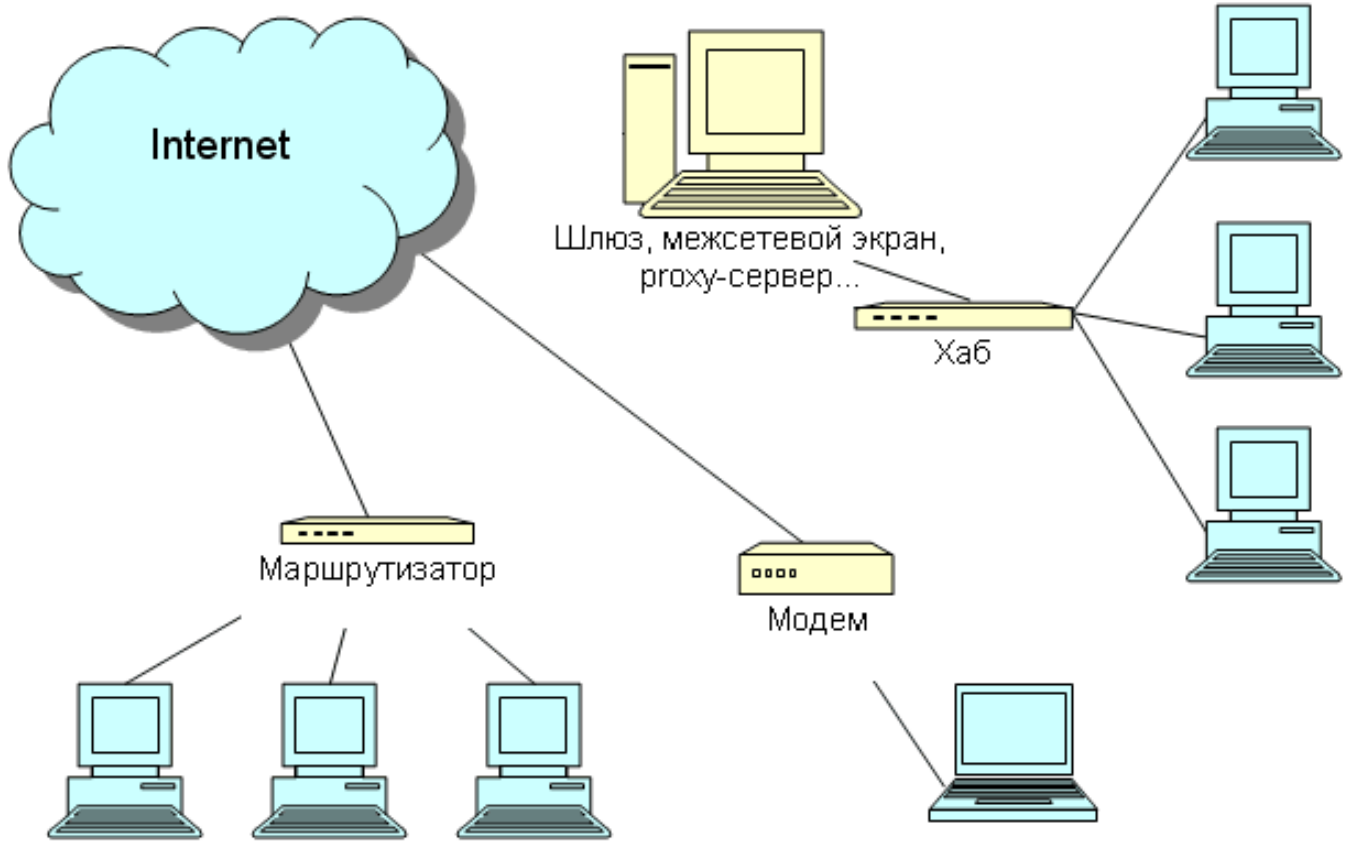
Решение задачи обнаружения нарушений в системе можно разбить на следующий этапы:

1. *Мониторинг и сбор информации. Выбор набора характеристик событий.*
2. *Предварительная обработка собранной информации, подготовка для анализа. (масштабирование данных, выделение значимых характеристик событий, снижение размерности характеристик, …)*
3. *Выбор метода обнаружения нарушений и построение модели состояния системы. Обработка данных в рамках используемой модели. Оценка качества работы системы обнаружения вторжений.*
4. *Анализ полученных результатов и реакция по результатам анализа.*

Сеть, протоколы (краткие сведения)

Аналогия:

компьютеры фиксированной и открытой архитектуры ↔ сети



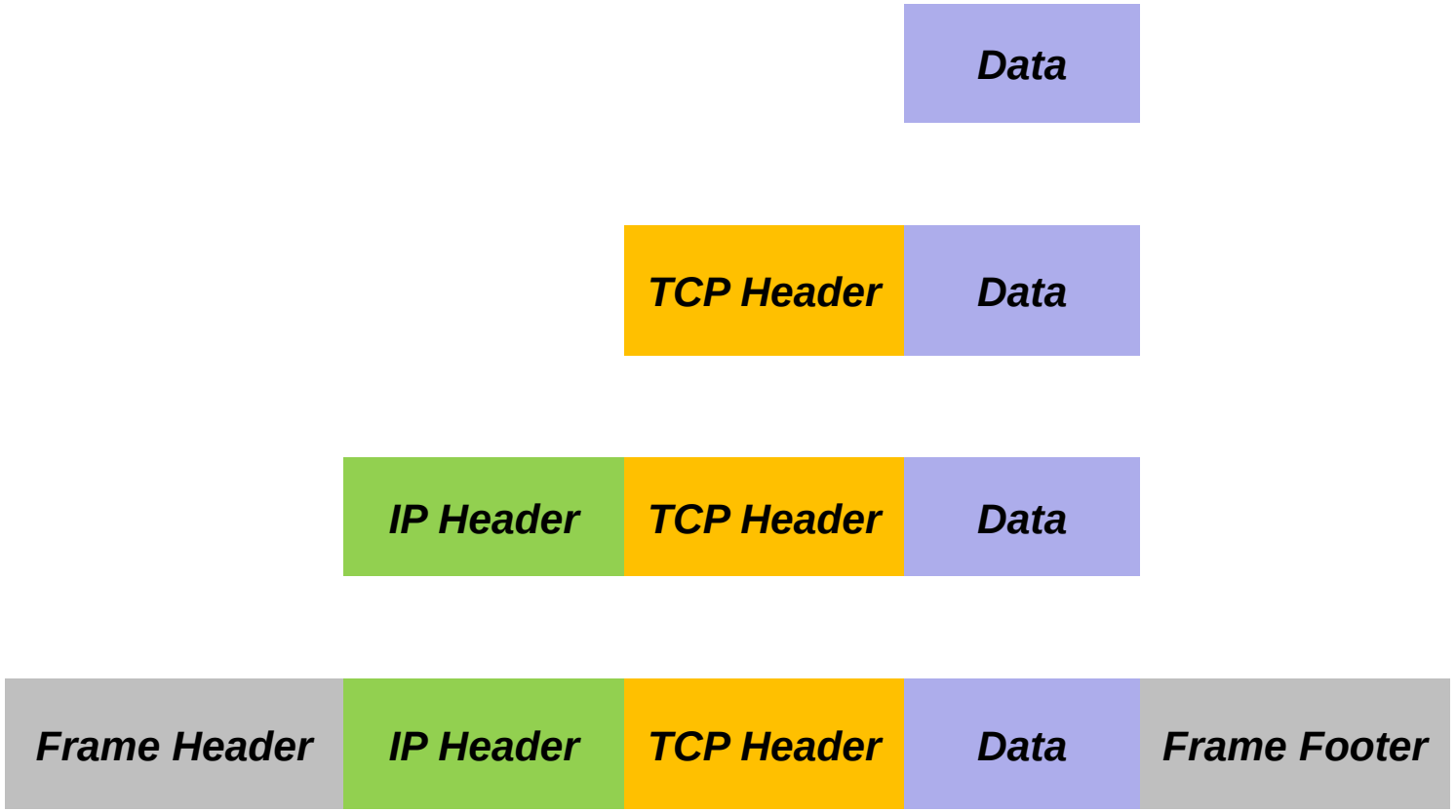
Сеть, протоколы (краткие сведения)

Протокол: Набор спецификаций описывающих формат для передачи и приёма информации в сети.

- В разных участках сети могут применяться разные протоколы.
- Для согласования протоколов разных участков введён *стек протоколов*.
- Стек протоколов не меняет содержимого пакетов, а использует *инкапсуляцию пакетов*: включение одного сетевого пакета в состав другого.

Инкапсуляция пакетов

Инкапсуля́ция: это метод создания сетевых протоколов, при протоколы передачи пакетов одного уровня сети включаются или инкапсулируются в протоколы пакетов более высокого уровня.



Application Layer
(приложение)

Transport Layer
(сокеты)

Network Layer
(маршрутизатор)

Link Layer
(физическая сеть)

Заголовки пакетов – основной источник информации для анализа состояния и выявления нарушений в сети.

Application Layer, Прикладной уровень

Протокол TFTP (Trivial File Transfer Protocol)

```
typedef struct _TFTP {
    WORD code;          // тип пакета
    union {
        struct {                    // code 1/2(RRQ/WRQ-запрос на чтение/запись)
            string  name;           // имя файла (/0 – признак конца строки)
            string  type;           // формат имени (/0 – признак конца строки)
        };
        struct {                    // code 3(DATA-данные)
            WORD nblock;            // номер блока
            BYTE data[512];         // данные
        };
        struct {                    // code 4(ACK-подтверждение)
            WORD nblock;            // номер блока
        };
        struct {                    // code 5(ERROR-подтверждение)
            WORD ecode;             // error code
            string  emessage;       // error message
        };
    };
} TFTP, *PTFTP;
```

Transport Layer, Транспортный уровень

Протокол UDP

```
typedef struct _UDPHDR {
    WORD    source;     // номер порта источника
    WORD    dest;       // номер порта приёмника
    WORD    len;        // размер сообщения (заголовок+данные)
WORD    check;      // контрольная сумма
} UDPHDR, *PUDPHDR;
```

Протокол TCP

```
typedef struct _TCPHDR {
    WORD    source;     // номер порта источника
    WORD    dest;       // номер порта приёмника
    DWORD   seq;        // квитанция
    DWORD   ack_seq;    //  подтверждение квитанции
    BYTE    res;        // размер заголовка – 4 бита + резерв
    BYTE    flags;      // флаги
    WORD    window;     // размер окна
    WORD    check;      // контрольная сумма
    WORD    urg_ptr;    // указатель срочность
} TCPHDR, *PTCPHDR;
```

Network Layer, Сетевой уровень

Протокол IP

```
typedef struct _IPHDR {
    BYTE    version:4; // номер версии, IPv4 или IPv6
            ihl:4;     // размер заголовка в четвёрках бит
    BYTE    tos;       // тип обслуживания
    WORD    tot_len;   // размер пакета (заголовок + данные)
    WORD    id;        // идентификатор фрагментированных пакетов
    WORD    frag_off;  // флаги и смещение фрагментации
    BYTE    ttl;       // время жизни
    BYTE    protocol;  // протокол верхнего уровня (6-tcp, 17-udp)
    WORD    check;     // контрольная сумма
    DWORD   saddr;     // IP адрес источника
    DWORD   daddr;     // IP адрес приёмника
} IPHDR, *PIPHDR;
```

Link Layer, Уровень связи

Большое разнообразие протоколов. Конкретный протокол зависит от физической реализации сети.

Протокол Ethernet DIX или Ethernet II

```
// преамбула
typedef struct _ETHIIHDR {
    BYTE    src[6];     // физический (MAC) адрес источника
    BYTE    dsr[6];     // физический (MAC) адрес приёмника
    WORD    type;       // протокол верхнего уровня (0x0800 – Ipv4, 0x86DD – IPv6, …)
} ETHIIHDR, *P ETHIIHDR;
```

Протоколы 802.3, 802.11, PPP, PPPoE, …

Практические примеры с использованием Wareshark.