Cyberwar

Conceitos de Defesa em Profundidade e Sistemas de *Firewall*

Vinícius Vieira

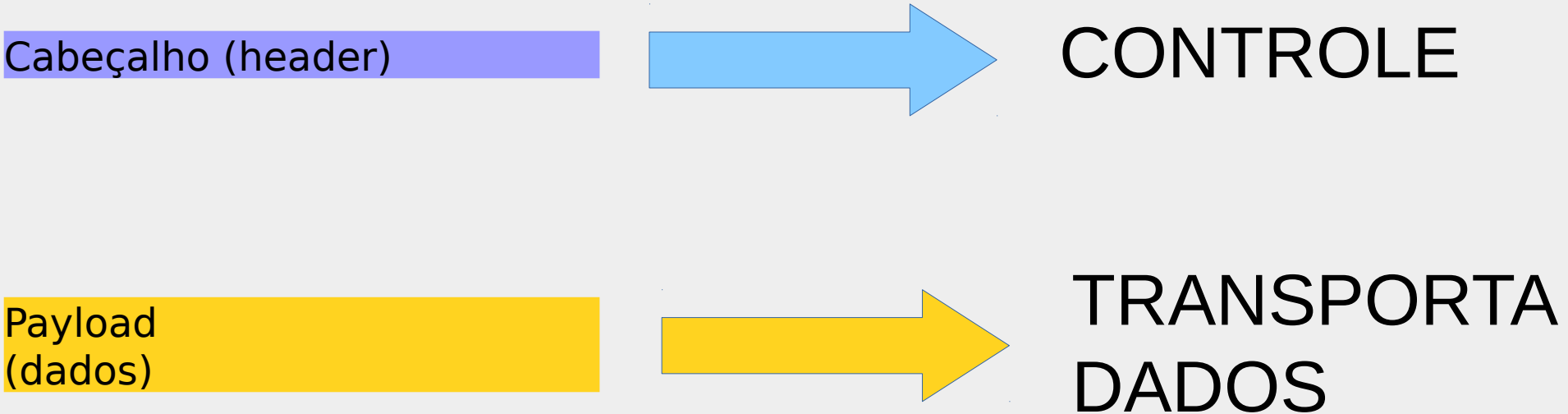
Roteiro

Introdução:
- Revisão do Modelo OSI;

Desenvolvimento:
- Conceitos de Defesa em Profundidade;
- Sistemas de *Firewall;*
- NETFILTER/IPTABLES;

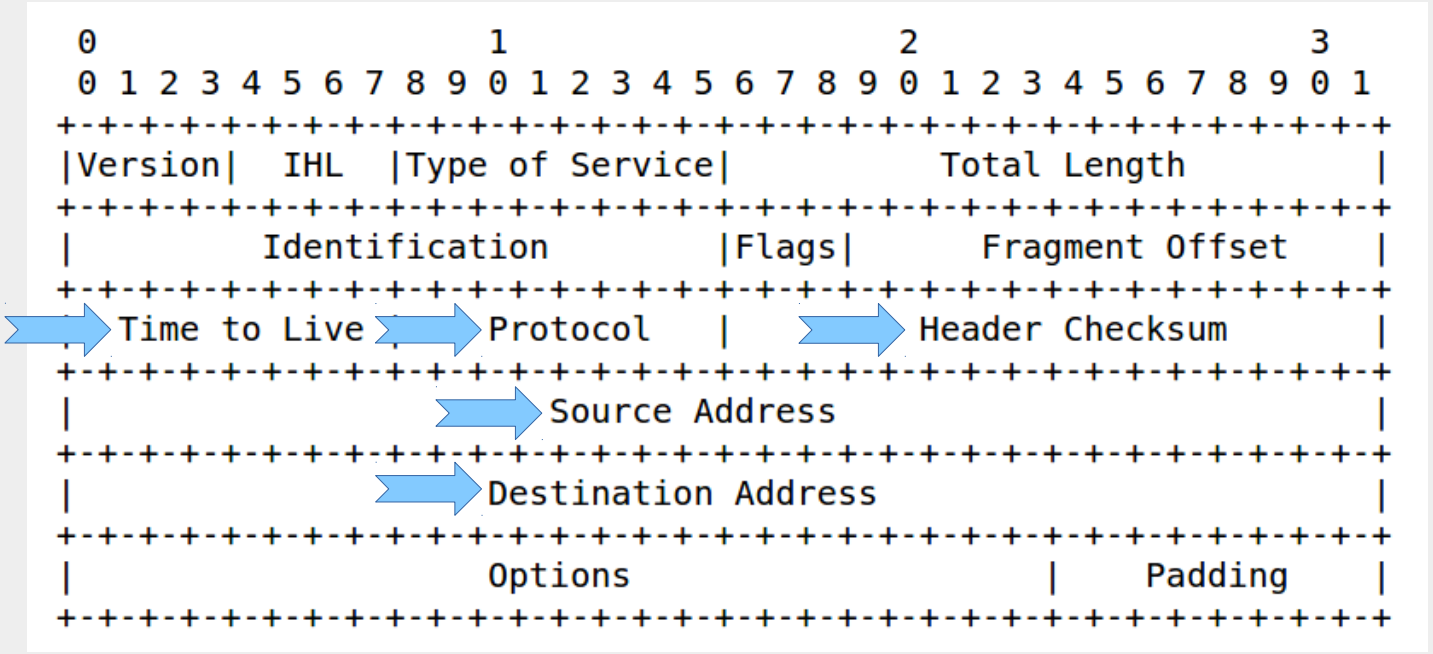
Conclusão:
- Trabalho pedido;
- Orientações, dúvidas.

Protocolos de Rede - Estrutura



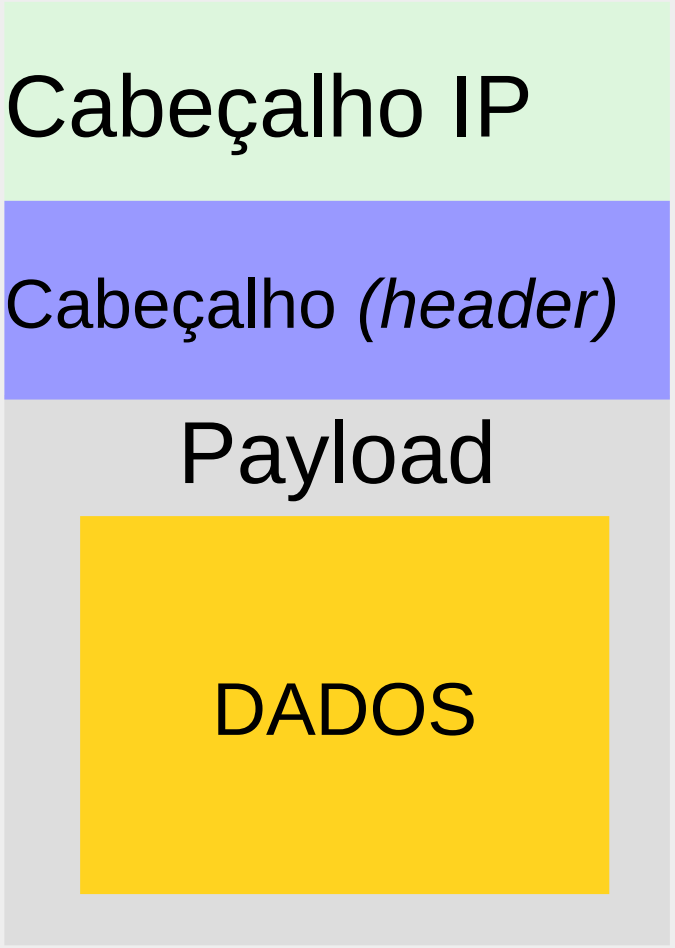
Ex: um pacote IP deve ter origem e destino: **Cabeçalho;**
Conteúdo transportado: **Payload.**

O Protocolo IP

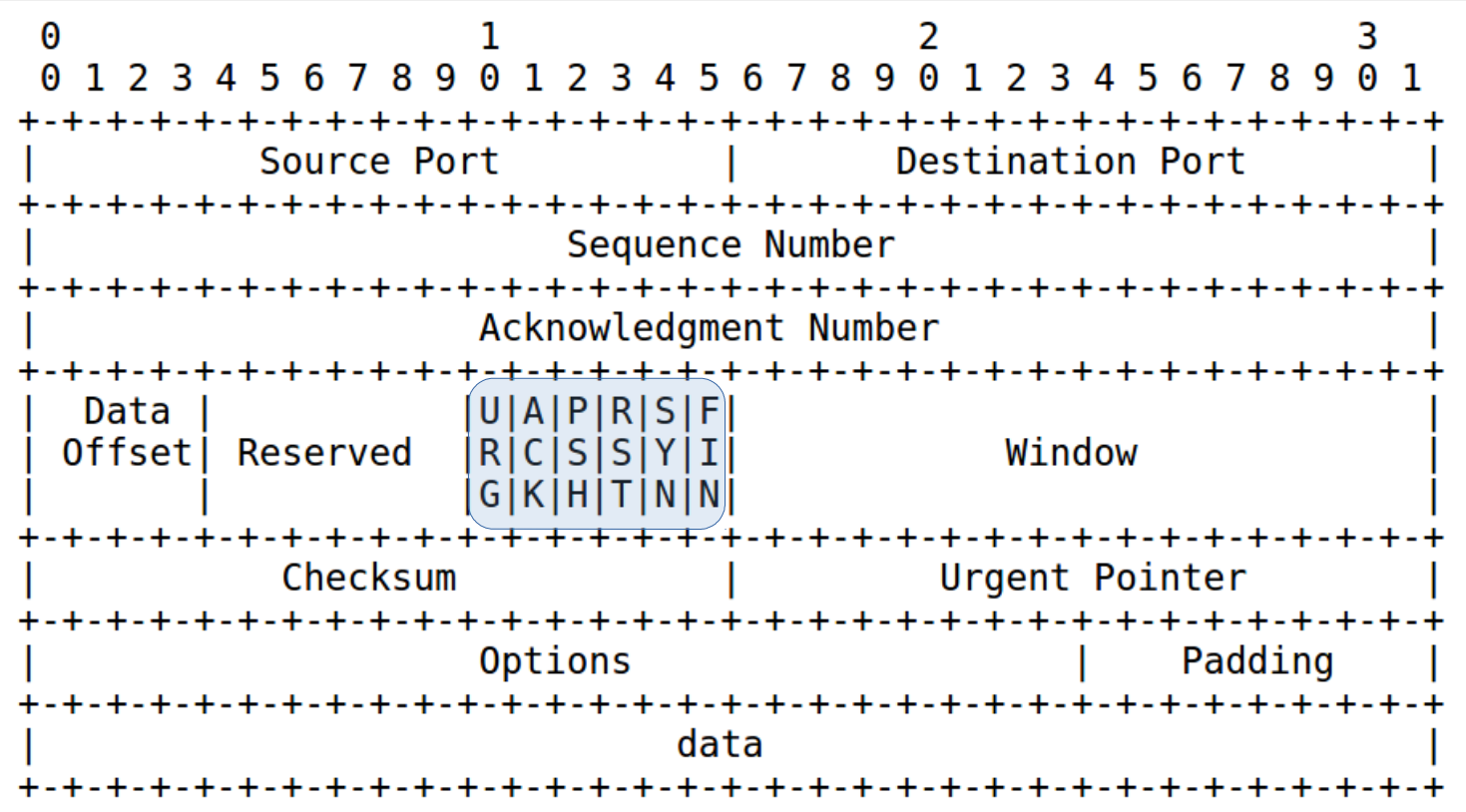


Função do Protocolo IP: Transportar outro protocolo.
Características: Endereçamento, encapsulamento

Encapsulamento



O Protocolo TCP



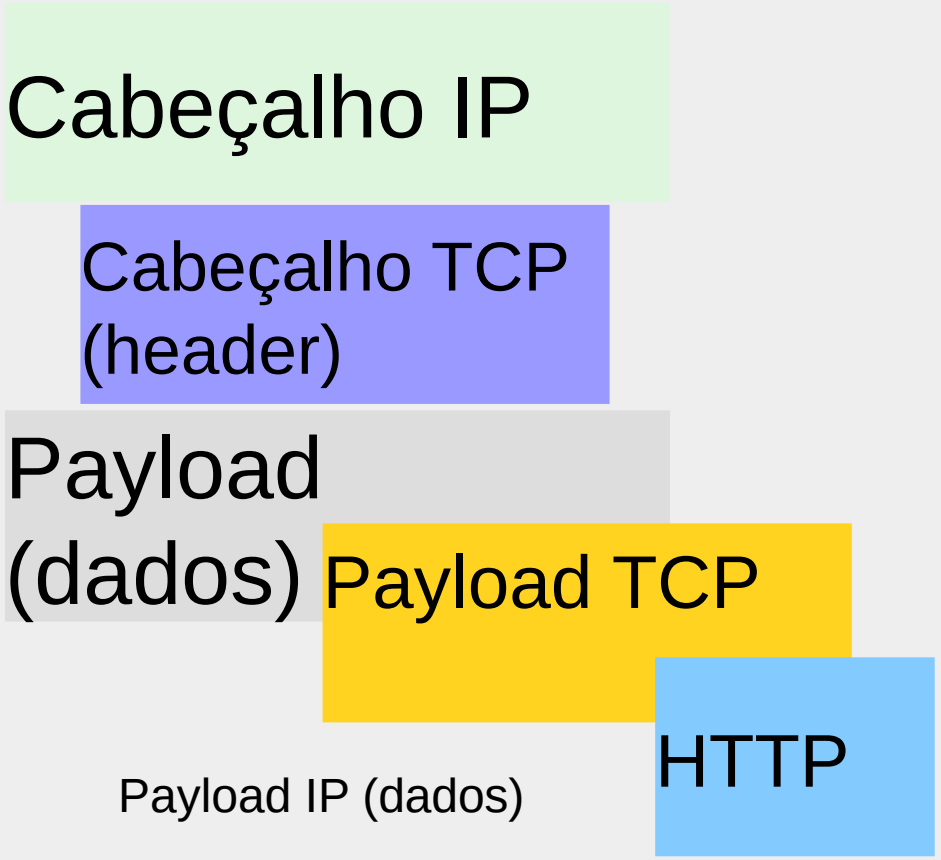
TCP: Full Duplex (Three Way Handshake);

Flags: enviadas a portas

Exemplo: Conexão WEB

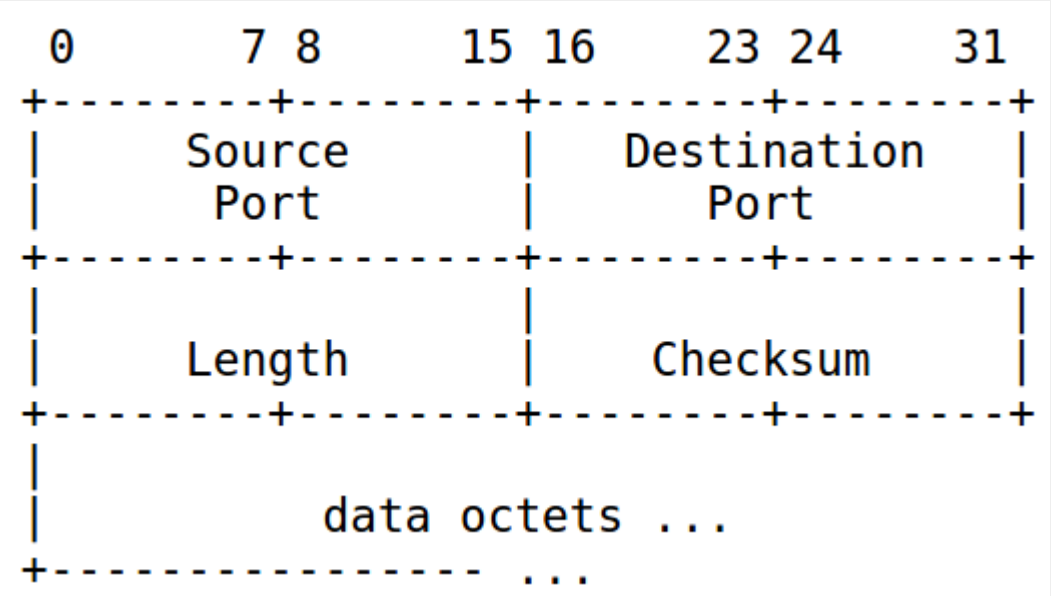
Cliente: SYN
Servidor: [Aloca recursos] ACK SYN
Cliente: ACK
Cliente: PUSH (Payload: HTTP: “get /”)
Servidor: ACK PUSH (Payload: “index.html”)
Cliente: ACK
Servidor: PUSH (Payload: “index.html”)
Cliente: ACK
(...)
Servidor:FIN
Cliente: ACK FIN
Servidor: ACK [Desaloca recursos]

Encapsulamento



Outros protocolos IP...

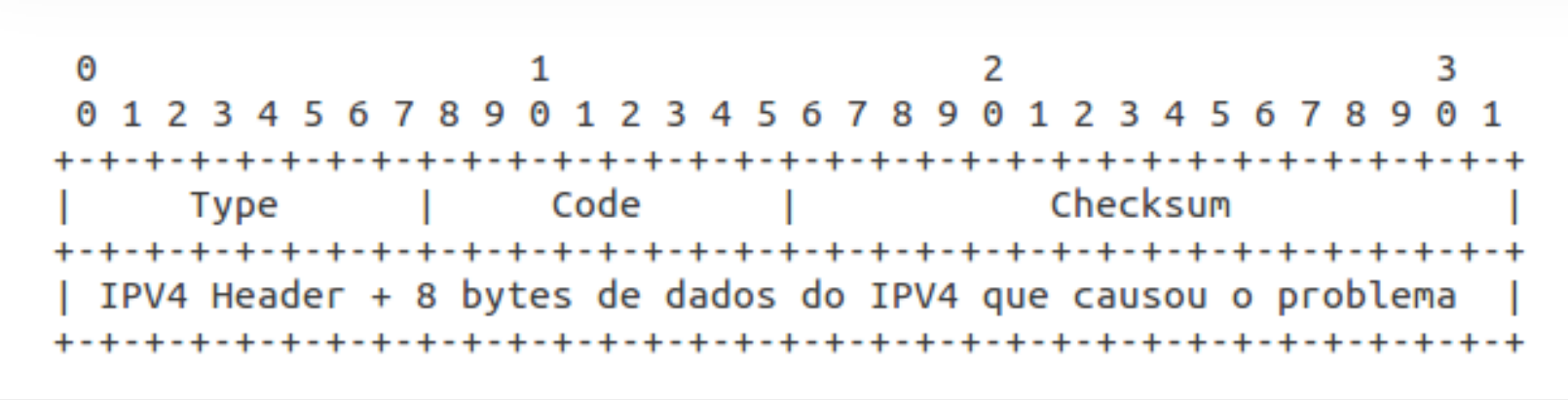
UDP:



Velocidade, sem garantia de entrega

Outros protocolos IP...

ICMP:



Função: Mensagens de controle da Rede IP

Tipos mais comuns: 3 (*Destination Unreachable*), 8 (*Echo Request*);
Códigos: 1 (*Destination host unreachable*), 3 (*Destination port unreachable*), etc.

O Modelo OSI (Open Systems Interconnection)

Modelo de 7 camadas:

Data Flow Layers (4 camadas inferiores): controlam basicamente as funções de rede e procuram oferecer serviços de transferência de dados com qualidade aceitável para as aplicações que utilizam a rede;

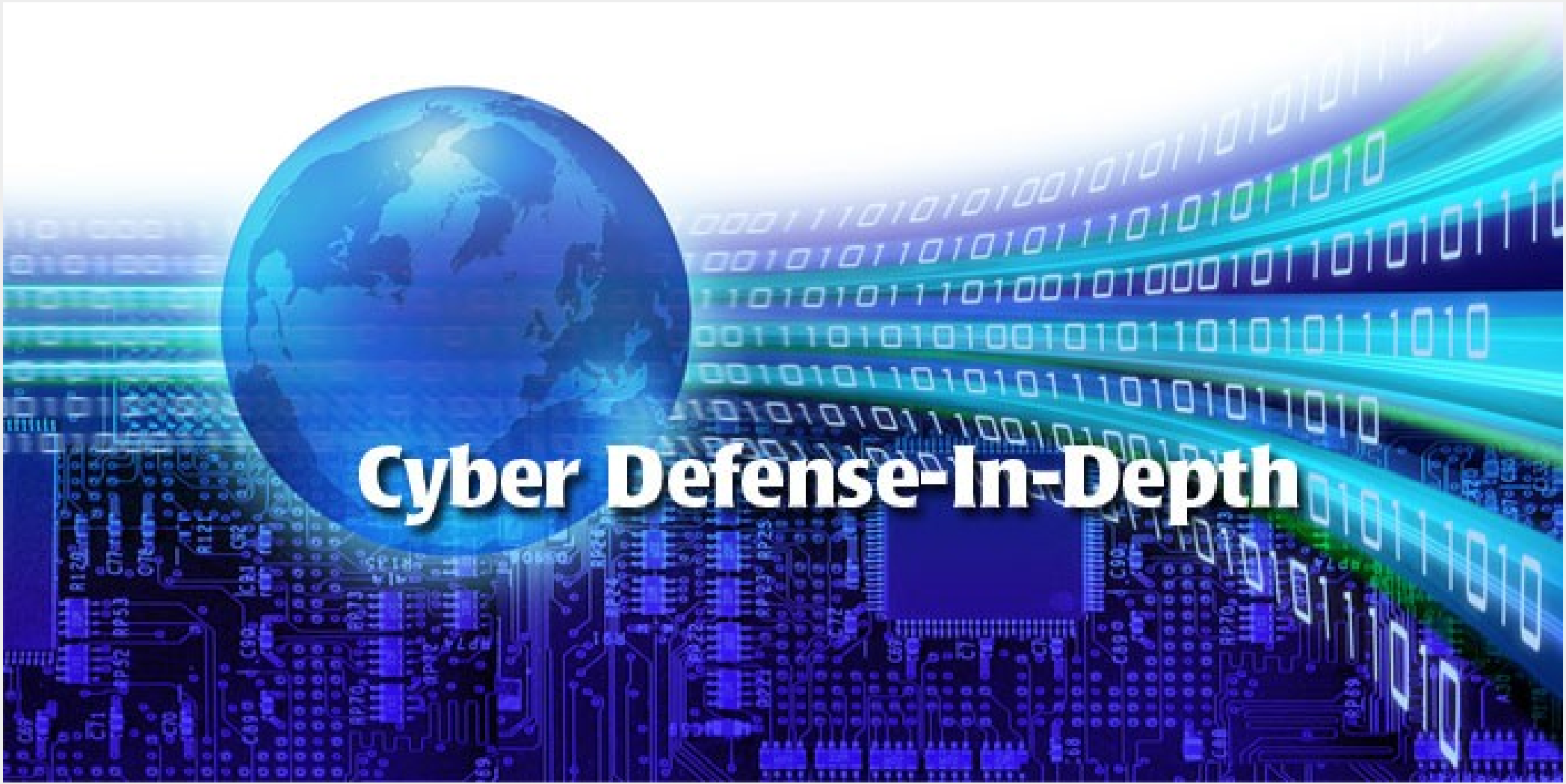
Application Layers (3 camadas superiores): tratam somente das funções específicas das aplicações, sem preocupação com os detalhes de redes. Assim, as aplicações podem se concentrar nas funções dos sistemas que os usuários estão utilizando.

Modelo OSI

Fonte: eriberto.pro.br

Resumindo OSI:

- Open Systems Interconnection, projetado pela ISO (International Organization for Standardization);
- Criado para permitir compatibilidade entre diferentes eqp;
- Depende do entendimento de protocolos;
- É uma cadeia de encapsulamentos;
- Uma camada depende da outra;
- Provê conectividade entre equipamentos distintos;
- Um tráfego de rede pode não atingir as camadas superiores;
- Seu entendimento não deve ser meramente acadêmico.


Cyber Defense-In-Depth

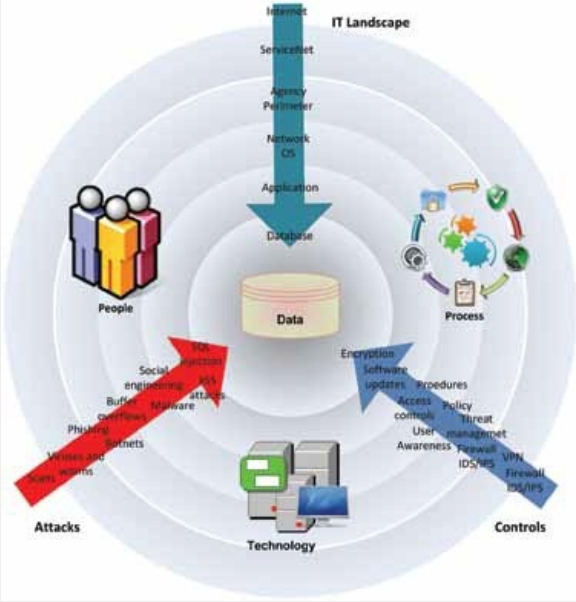
Defesa em Profundidade

- Conceito antigo aplicado a ideias recentes;

- Estratégia de proteger os canais de comunicação em várias etapas (redundância);

- Se um ataque provoca falha em um mecanismo de segurança, outros podem fornecer proteção;

Defesa em Profundidade

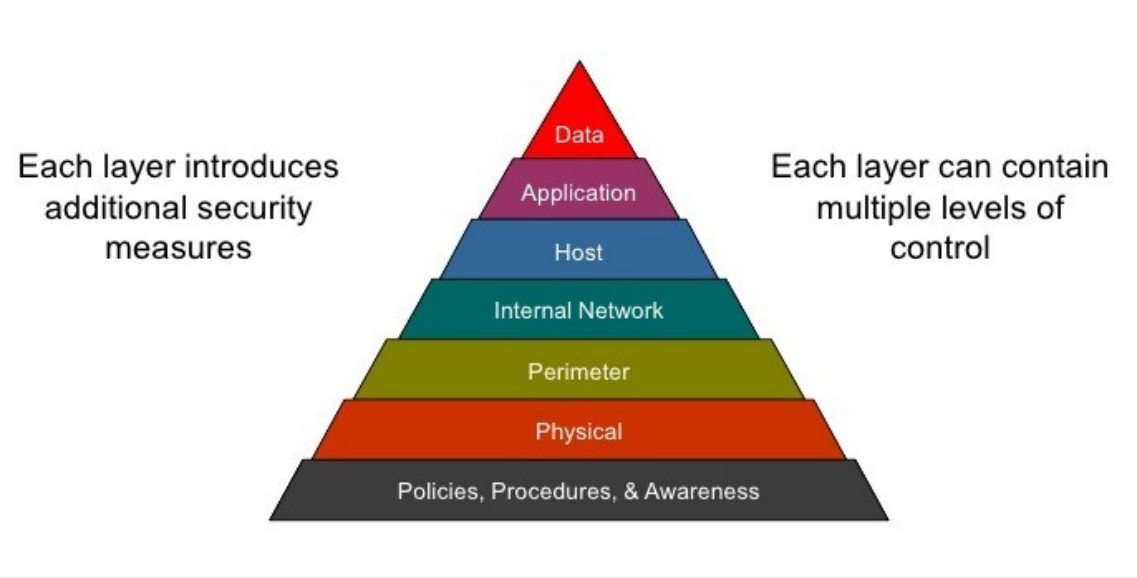


- Processo e prática constantes que se concentram na proteção, detecção e reação a ameaças.

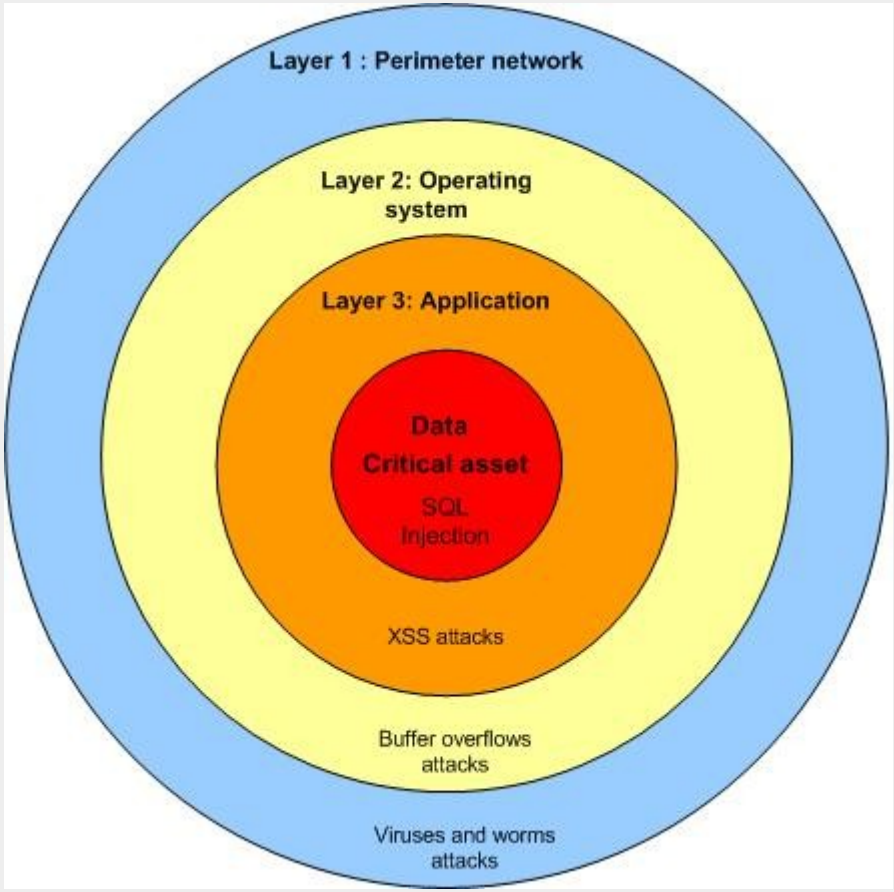
- Não se deve depender apenas de um mecanismo de segurança, não importando o quão forte ele pareça ser, porque a falha deste mecanismo único pode vir a comprometer toda a segurança do sistema.

Defesa em Profundidade

Exemplo: Confiar única e totalmente em um equipamento de proteção de borda ("Firewall") para fornecer segurança para uma aplicação de uso interno: um atacante determinado pode contornar as regras, utilizando um ataque físico ou obtendo informações mediante um ataque de engenharia social.



Defesa em Profundidade



Defesa em Profundidade

- As pessoas fazem parte do processo!!!

- O risco total para o sistema precisa ser mensurado.

Ex: Uma aplicação com autenticação usuário/senha - exigir um aumento do tamanho da senha de 8 para 15 caracteres, incluindo caracteres especiais, números, maiúsculas, minúsculas, etc.

Pode resultar em usuários anotarem suas senhas, diminuindo assim a segurança global do sistema;

Uma alternativa: Autenticação de duplo fator (smart card, tokens).

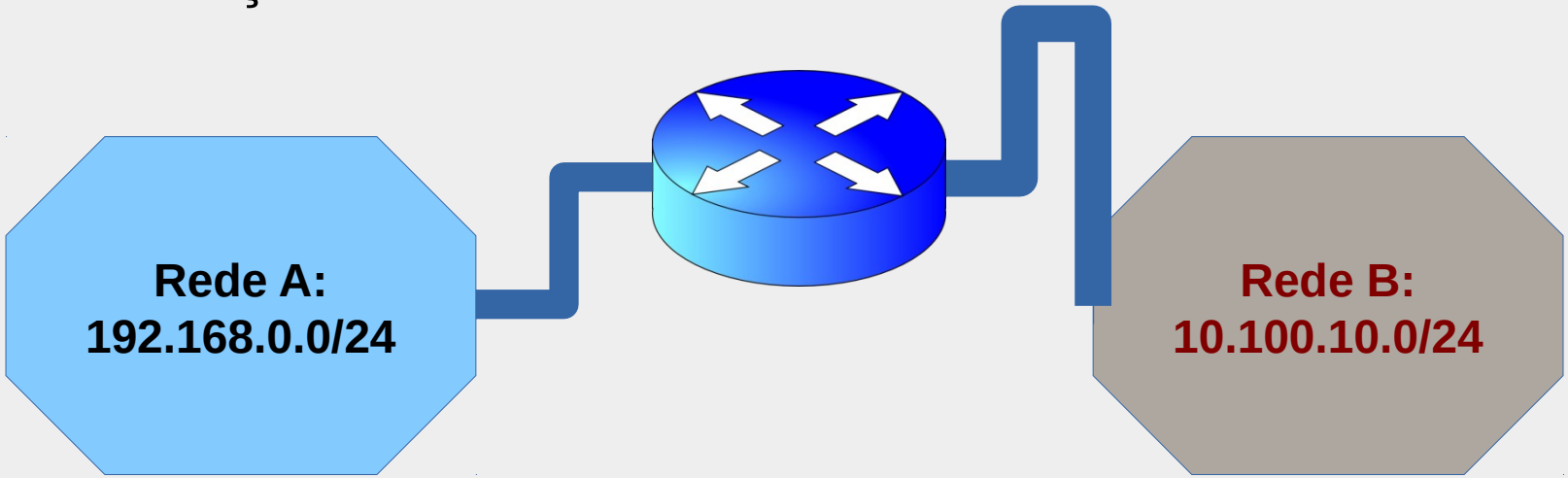
Como implementar?

- Utilização de ferramentas de segurança fundamentais, como anti-vírus, anti-malware, anti-spam, firewalls, filtragem de conteúdo, controles de acesso, criptografia, detecção e prevenção de intrusão, entre outras.

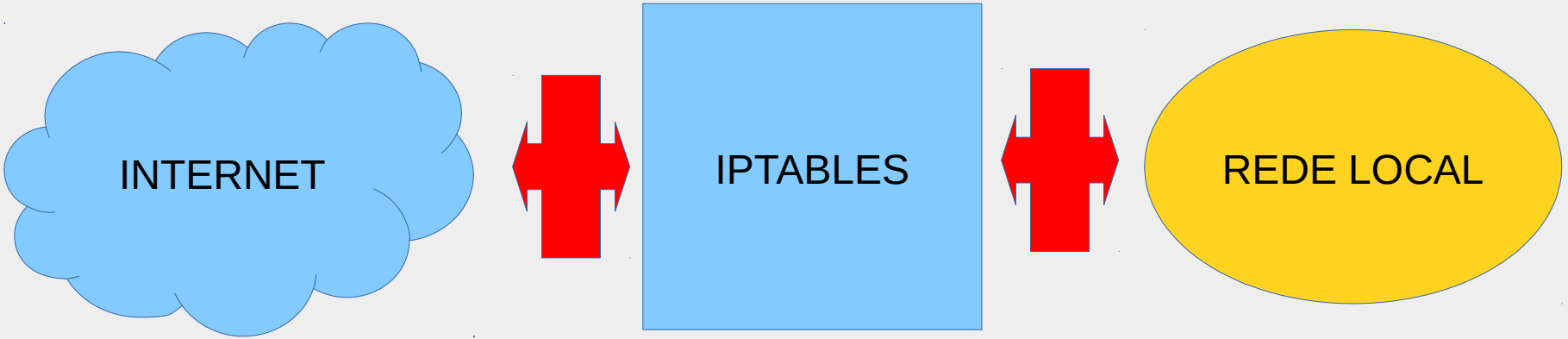
- Analogia: Time de futebol – atacantes fecham a saída de bola, defesa compactada (diminui espaço do adversário), zagueiros em constante atenção, goleiro fazendo seu papel.

Antes de Sistemas de *Firewall*: **Roteamento**

- O roteamento interliga segmentos de redes diferentes, utilizando a camada de rede do modelo OSI;
- Não há mudança no endereçamento IP, salvo em casos de NAT;
- Há mudanças na camada 2.



Firewall



- Firewall é um CONCEITO, SISTEMA, não é um equipamento ou solução isolada;
- Elementos: filtros de pacotes, proxies, IDS, IPS, HIDS, Antivírus, etc;
- Defesa em Profundidade;
- Todo esforço físico e lógico voltado para a segurança da rede (*by Eriberto*).

Sistemas de Firewall (Cont.)

- Um filtro de pacotes isoladamente não pode ser considerado um Firewall;
- O IPTABLES atua como filtro de pacotes, trabalhando nas **camadas 2, 3 e 4 do modelo OSI**. (E a camada 7??)
- Pode manipular **estados de conexões** (*NEW*, *ESTABLISHED*, *INVALID* e *RELATED*);
- Em conjunto com outros elementos, tem papel importante na proteção da Rede. Mas, sozinho, não faz frente a todas as ameaças.

Proxies

- Elemento intermediário entre cliente e servidor. Função: evitar que o cliente tenha contato direto com o servidor.

- Em um modelo de segurança baseado em Sistemas de Firewall, nenhum cliente deverá ter acesso direto a um servidor, mas apenas via Proxy.

Proxies (Cont.)

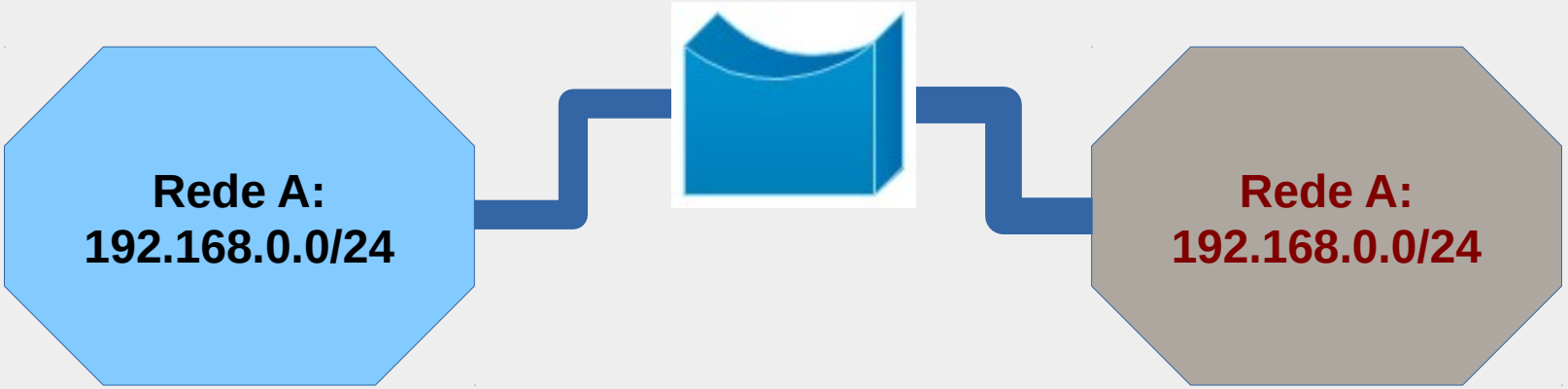
- A melhoria na performance está ligada à existência ou não de cache, e não do proxy propriamente dito;

- São elementos simples e devem ser diferentes dos Servidores a que vão servir.

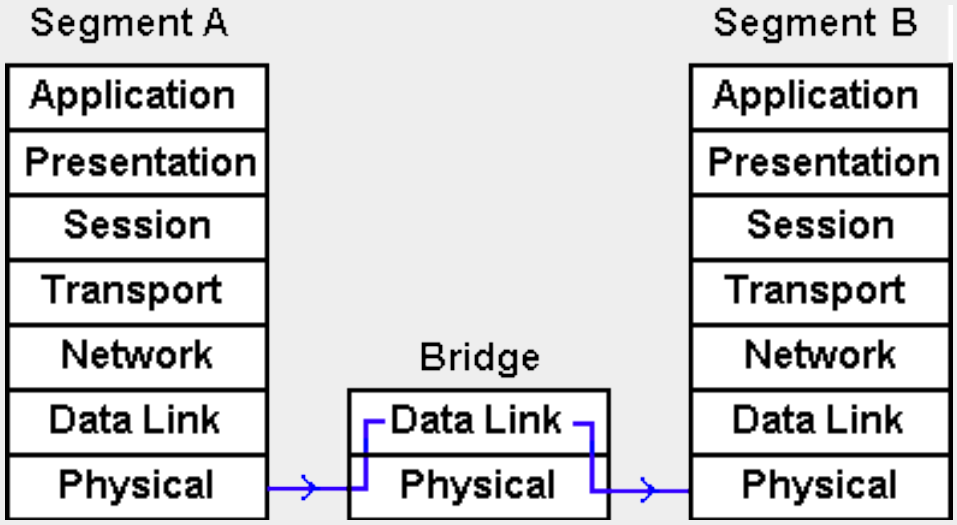
- São meros “retransmissores”: Não há necessidade de *features*, recursos, etc. O trabalho pesado fica com o servidor. => Segurança.

Bridges

- Não há mudanças no Payload! O que temos lá dentro? IP, TCP/UDP, HTTP, SMTP, etc etc…
- Para Sistemas de Firewall, ela atua de modo invisível e permite a visualização de todo o tráfego entre dois pontos, inclusive **Payloads**;
- Se trafegar em claro, o conteúdo é visível;

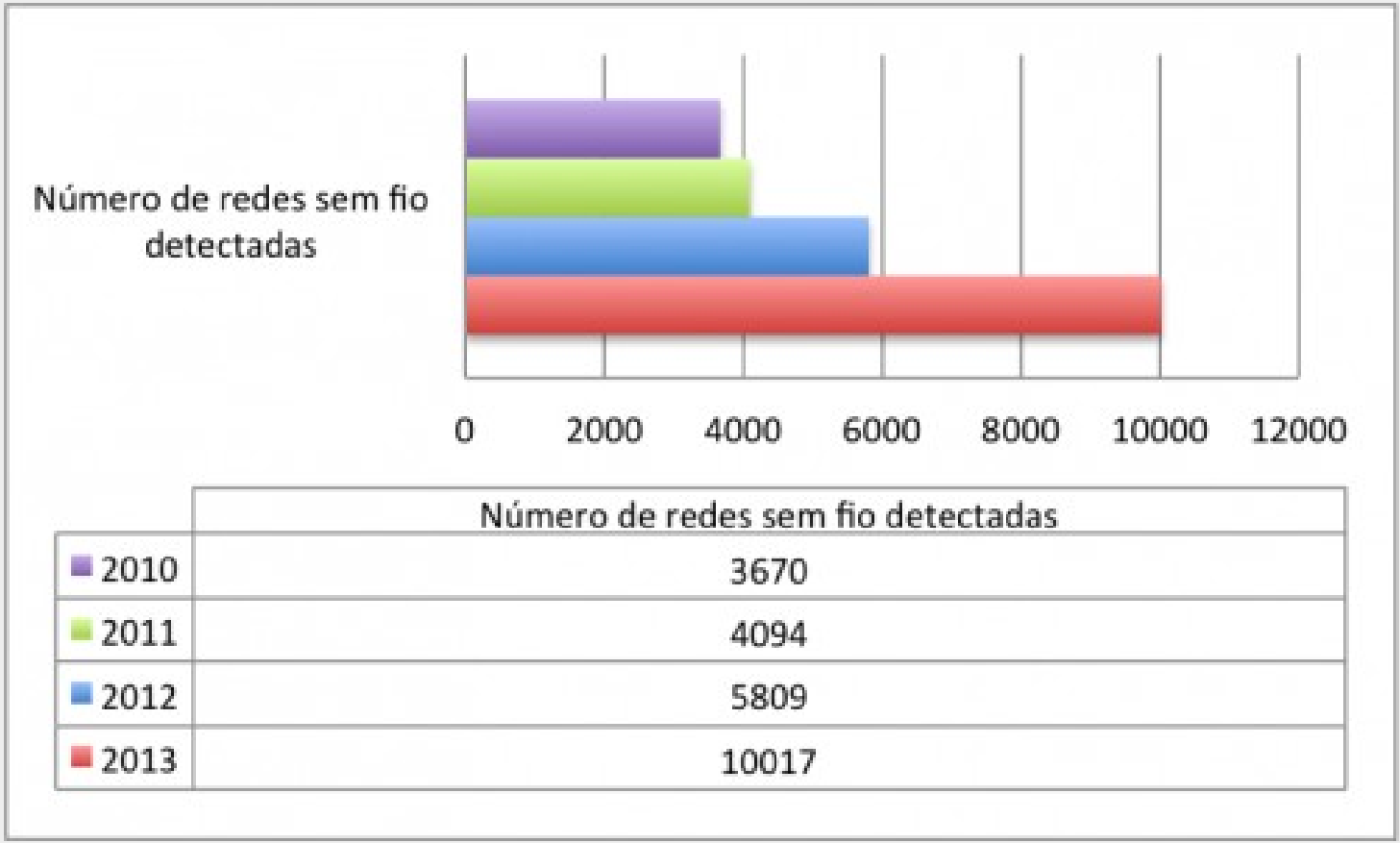


Bridges



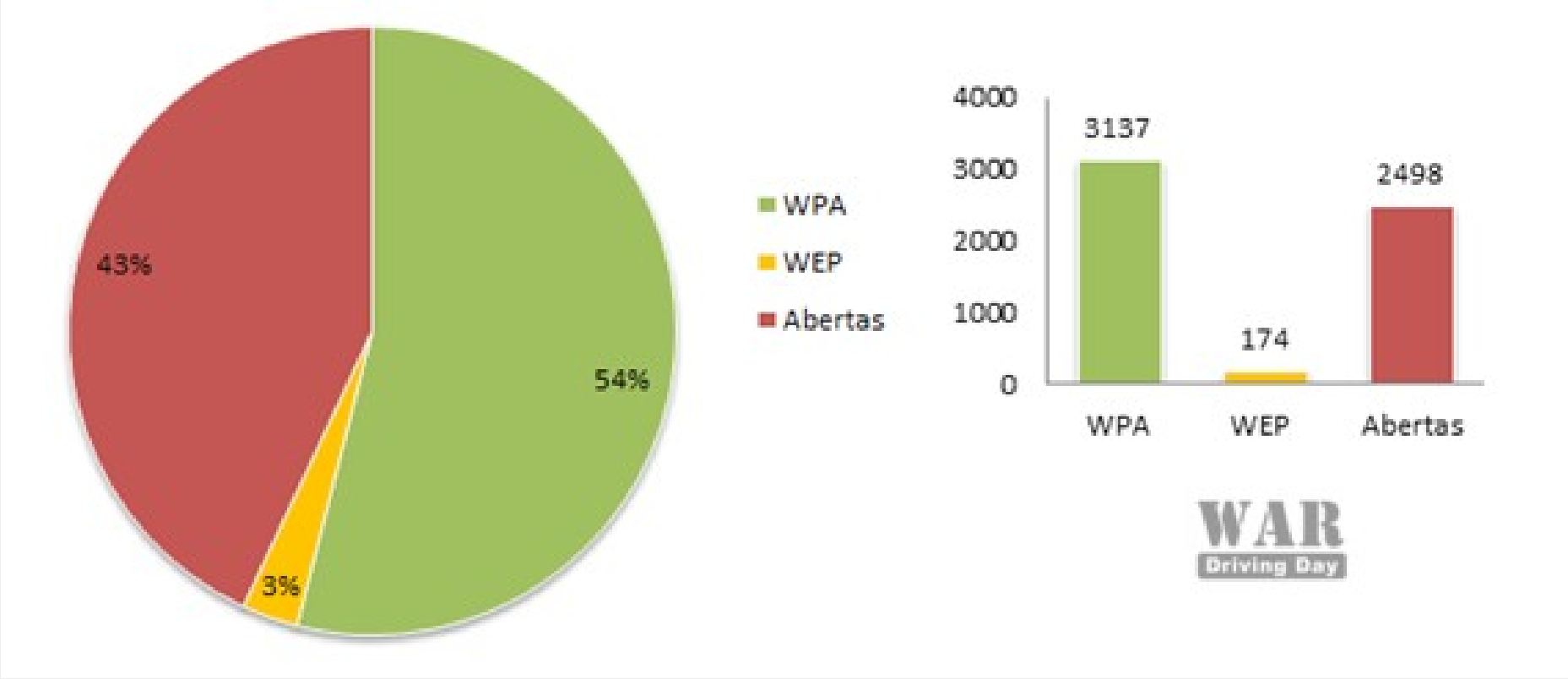
- Interligação entre porções de uma mesma rede;
- Pode ser um dispositivo com duas interfaces ligando duas redes;
- Atua na camada 2: Não há modificações em nível de frame ethernet;
- Pode ser vista como uma “emenda”: endereços físicos de origem e destino não se alteram, como no caso de roteamento – Transparência;
- Uma das funções: conversão de tecnologias de camada 2 – Ex: Modens ADSL (ATM – Ethernet ou DOCSIS – Ethernet). Mudanças apenas em cabeçalhos.

Case: Redes Wirelles no RJ



	Número de redes sem fio detectadas
2010	3670
2011	4094
2012	5809
2013	10017

Case: Redes Wirelles no RJ



Sistemas de Detecção de Intrusões (IDS – Intrusion Detection Systems)

- Objetivo principal: geração de registros detalhados sobre todas as operações da rede (logs).

- Captura uma cópia dos pacotes em tráfego na rede, analisando os cabeçalhos (endereços e portas de origem e destino etc), **payloads suspeitos**, aplicações, etc.;

- Existência de falsos positivos (alerta de uma atividade normal): situação normal e esperada neste sistema - Necessidade de análise manual.

- Dois métodos:

Host Intrusion Detection Systems (HIDS): Instalados em clientes;

Network Intrusion Detection Systems (NIDS): Instalados em pontos chaves da Rede (roteadores, gateways, etc).

Sistemas de Prevenção de Intrusões (IPS - Intrusion Prevention Systems)

- Objetivo principal: tomada de ações imediatas frente a ações consideradas ilegais, com geração de registros detalhados sobre todas as operações da rede (logs);

- Endereços e portas de origem e destino, **payloads suspeitos**, usuários, etc.;

- Existência de falsos positivos: Não podem acontecer (bloqueio de conexões legítimas);

- **Situação ideal: Filtro de Pacotes – Filtro de Estados – Proxy - IPS – IDS – Firewall (Teoria da Cebola).** Na dúvida, o IPS pode deixar passar uma conexão. Neste caso, o IDS vai registrar este evento. Após a análise manual, o IPS é reconfigurado com uma regra contra o evento malicioso.

Resumindo...

Camada	Elementos
Camada de Aplicação	Proxies, IDS, IPS, AV
Camada de Transporte	Filtros de estados e pacotes, Proxies, IDS, IPS, AV usam algumas informações para atuar na camada de Aplicação
Camada de Rede	Filtros de pacotes; Proxies, IDS, IPS, AV usam algumas informações para atuar na camada de Aplicação
Camada de Enlace	Filtros de pacotes com endreços físicos de origem; Proxies, IDS, IPS e AV usam algumas informações para atuar na camada de Aplicação

NETFILTER (IPTABLES)

- O Netfilter (Iptables) é um filtro em nível de pacotes, e baseia-se na análise dos cabeçalhos dos pacotes (endereço/porta de origem/destino, protocolos, estados de conexão);
- Funciona através da comparação de regras para saber se um pacote tem ou não permissão para trafegar pelas interfaces.
- Não funciona de forma automática. São necessários conhecimentos básicos de rede TCP/IP, roteamento e portas para criar as regras que executarão os filtros. A eficiência depende do controle adequado das regras que serão criadas.

IPTABLES (Cont.)

- As regras (*rules*) de fitragem, geralmente, são compostas assim:

#iptables -t [tabela] -A [chain] [opções] -j [alvo] [opções]

Ex:

#iptables -t filter -A INPUT -i eth0 -p icmp -j DROP

#iptables -t filter -A OUTPUT -o eth1 -p tcp -j LOG

IPTABLES (Cont.)

#iptables -t [tabela] -A [chain] [opções] -j [alvo] [opções]

-Comandos básicos:

L: Lista todas as regras existentes (*LIST*);

D: Exclui uma regra específica (*DELETE*);

F: Zera todas as regras (*FLUSH*);

Exemplos:

#iptables -L -t filter

#iptables -L -t nat

#iptables -L -t mangle

IPTABLES (Cont.)

Policy

- Política da chain. Por padrão, a política de cada chain é configurada como ACCEPT. A única opção além desta é a DROP.
- Policy ACCEPT: Se nenhuma regra der match, o tráfego é permitido;
- Policy DROP: Se nenhuma regra der match, o tráfego é bloqueado;

IPTABLES (Cont.)

#iptables -t [tabela] -A [chain] [opções] -j [alvo] [opções]

Ex:

#iptables -L -t filter

#iptables -P INPUT DROP

#iptables -L -t filter

#iptables -P INPUT ACCEPT

#iptables -L -t filter

IPTABLES (Cont.)

- Tabelas:

FILTER: Tabela padrão. Contém 3 *chains* padrões:

INPUT - Consultada para dados que chegam a máquina;

OUTPUT - Consultada para dados que saem da máquina;

FORWARD - Consultada para dados que são redirecionados para outra interface de rede ou outra máquina.

IPTABLES (Cont.)

NAT - Usada para dados que gera outra conexão. Possui 3 chains padrões:

PREROUTING - Consultada quando os pacotes precisam ser modificados logo que chegam (ex: redirecionamento de portas);

OUTPUT - Consultada quando os pacotes gerados localmente precisam ser modificados antes de serem roteados. Somente é consultado para conexões que se originam de IPs de interfaces locais.

POSTROUTING - Consultada quando os pacotes precisam ser modificados após o tratamento de roteamento.

IPTABLES (Cont.)

MANGLE: Utilizada para alterações especiais de pacotes (ex: TOS). Possui 5 chains padrões:

PREROUTING - Consultada quando os pacotes precisam ser modificados antes de ser enviados para o chain PREROUTING da tabela nat.

INPUT - Consultado quando os pacotes precisam ser modificados antes de serem enviados para o chain INPUT da tabela filter.

FORWARD - Consultado quando os pacotes precisam ser modificados antes de serem enviados para o chain FORWARD da tabela filter.

OUTPUT - Consultado quando os pacotes precisam ser modificados antes de serem enviados para o chain OUTPUT da tabela nat.

POSTROUTING - Consultado quando os pacotes precisam ser modificados antes de serem enviados para o chain POSTROUTING da tabela nat.

IPTABLES (Cont.)

#iptables -t [tabela] -A [chain] [opções] -j [alvo] [opções]

-Alvos:

DROP: bloqueia a passagem do pacote;

ACCEPT: aceita a passagem do pacote;

LOG: registra o pacote no sistema de log;

REJECT: rejeita o pacote com uma mensagem ICMP de retorno.

IPTABLES (Cont.)

#iptables -t [tabela] -A [chain] [opções] -j [alvo] [opções]

- Opções:

-p: Protocolo - Define qual o protocolo TCP/IP deverá ser tratado (TCP, UDP e ICMP);

Ex: **#iptables -t filter -A INPUT -p tcp -j ACCEPT**

-s: Origem - Define qual o endereço de origem em que a regra vai atuar. Deve ter dois argumentos: endereço/máscara;

Ex: **#iptables -t filter -A INPUT -s 192.168.0.10 -j DROP**

-d: Destino - Define qual o endereço de origem em que a regra vai atuar.

Ex: **#iptables -t filter -A OUTPUT -d 192.168.0.10 -j DROP**

IPTABLES (Cont.)

#iptables -t [tabela] -A [chain] [opções] -j [alvo] [opções]

- Opções:

-i: Interface de entrada: Define o nome da interface de rede onde trafegarão os pacotes de entrada;

Ex: **#iptables -t filter -A INPUT -i eth0 -j DROP**

-o: Interface de saída: Define o nome da interface de rede onde trafegarão os pacotes de saída;

Ex: **#iptables -t filter -A OUTPUT -o eth1 -j DROP**

IPTABLES (Cont.)

#iptables -t [tabela] -A [chain] [opções] -j [alvo] [opções]

--source-port ou --sport - Especifica uma porta ou faixa de portas de origem;

Ex: **#iptables -t filter -A INPUT -p tcp --sport 8080 -j DROP**

--destination-port ou --dport - Especifica uma porta ou faixa de portas de destino;

Ex: **#iptables -t filter -A OUTPUT -p udp --dport 53 -j DROP**

!: Exceção - executar uma ação exceto com o dado que vem a seguir.

Sistemas de Firewall (Cont.)

Tratamento dos estados de conexão: Necessária a ativação do módulo “state”

NEW - Novas conexões;

ESTABLISHED - Conexões já estabelecidas;

RELATED - Pacotes relacionados indiretamente a uma conexão. Ex: mensagens de erro ICMP.

INVALID - Pacotes que não podem ser identificados por algum motivo. Ex: Respostas a conexões desconhecidas.

IPTABLES (Cont.)

#iptables -t [tabela] -A [chain] [opções] -j [alvo] [opções]

Ex:

#iptables -A INPUT -m state --state INVALID -j DROP

#iptables -A OUTPUT -m state --state NEW -j ACCEPT

#iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT

IPTABLES (Cont.)

Utilização de Scripts:

Shell script comum, com os comandos que devem ser executados, na ordem necessária. Ex:

```
#!/bin/bash
iptables -F
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -p tcp -i eth0 -s 192.168.10.100 -j DROP
iptables -A INPUT -p tcp -i eth0 -s 192.168.10.0/24 -j ACCEPT
```

IPTABLES (Cont.)

#iptables [-t tabela] [opção] [chain] [dados] -j [ação ou alvo]

-Exemplos:

iptables -A INPUT -s 192.168.200.10 -i eth0 -j REJECT

Rejeita todos os pacotes que vem do endereço 192.168.200.10.

iptables -A INPUT -s 192.168.20.20 -i eth0 -j DROP

?????

IPTABLES (Cont.)

#iptables [-t tabela] [opção] [chain] [dados] -j [ação ou alvo]

-Exemplos:

iptables -A INPUT -s 192.168.20.20 -i eth0 -j LOG

iptables -A INPUT -s 192.168.20.20 -i eth0 -j ACCEPT

iptables -A INPUT -s 192.168.20.20 -i eth0 -j DROP

???

iptables -t filter -A INPUT ! -s 192.168.20.30 -j DROP

IPTABLES (Cont.)

#iptables [-t tabela] [opção] [chain] [dados] -j [ação ou alvo]

-Exemplos:

iptables -A INPUT -s 192.168.20.20 -i eth0 -j LOG

iptables -A INPUT -s 192.168.20.20 -i eth0 -j ACCEPT

iptables -A INPUT -s 192.168.20.20 -i eth0 -j DROP

???

iptables -t filter -A INPUT ! -s 192.168.20.30 -j DROP

Rejeitar todos os pacotes EXCETO os que vem do endereço 192.168.20.30.

IPTABLES (Cont.)

#iptables [-t tabela] [opção] [chain] [dados] -j [ação ou alvo]

-Exemplos:

iptables -A INPUT -p icmp --icmp-type 8 -j REEJCT --reject-with icmp-host-unreachable

- **icmp-net-unreachable**
- **icmp-host-unreachable**
- **icmp-port-unreachable**
- **icmp-proto-unreachable**
- **icmp-net-prohibited**
- **icmp-host-prohibited or**
- **icmp-admin-prohibited**

IPTABLES (Cont.)

Ferramenta relacionada:

Firewall Builder

http://www.fwbuilder.org/

Objetivo: criar um ambiente gráfico para administração das regras do Iptables. Exporta as configurações para, além do próprio IPTABLES, appliances que realizam filtros de pacotes;

Features: Identificação de regras inválidas, geração automática de configurações, utilização de SSH para implementação das regras no filtro, etc

Qual é a diferença entre os alvos **REJECT** e **DROP**?

Que impactos essas diferentes técnicas de bloqueio de pacotes tem para o cliente?

Que importância há na configuração do módulo de estados de conexões?