

# Capítulo 4

---

## **Footprint e Fingerprint**

# Footprint

---

Busca detalhada de informações sobre o alvo para uma intrusão.

# Footprint

---

- ❑ É a organização de idéias como um todo, tentando criar o melhor e mais completo perfil do alvo a ser atacado.
  - ❑ O intuito é criar um perfil de uma máquina-alvo, para descobrir falhas que possam ser exploradas a partir de configurações e senhas padrões.
-

# Footprint

---

- ❑ A partir do resultado do Footprint é que é traçado a estratégia de ataque.
  - ❑ Um Footprint dura, enquanto for necessário.
  - ❑ Pode ser colocado em prática de muitas formas, e é limitado apenas pela imaginação do atacante.
-

# Objetivos comuns de Footprint

---

- ❑ Levantamento de Informações de Domínios:
    - Nomes de domínios.
    - Responsáveis pelos domínios
    - Servidores de domínios.
  - ❑ Identificação do SO de máquina-alvo (Fingerprint).
  - ❑ Descobrir subredes.
  - ❑ Serviços TCP e UDP disponíveis.
  - ❑ Topologia da rede.
-

# Objetivos comuns de Footprint

---

- ❑ Contas de Email, FTP e outros serviços.
  - ❑ Nomes de usuários e de grupos.
  - ❑ *Banners* que identificam versões de serviços.
  - ❑ Identificação de roteador e Tabelas de roteamento.
  - ❑ Servidores ocultos por NAT (Network Address Translator).
  - ❑ Endereços de e-mails.
-

# Objetivos comuns de Footprint

---

- ❑ Informações de serviços SNMP mal configurados.
  - ❑ Intervalos (Ranges) de IP de domínios.
  - ❑ Estrutura de segurança quanto a existência de:
    - Firewalls
    - Sistemas IDS
    - Honeypots
-

# Footprint

---

- Engenharia Social.
  - Levantamento de Informações do Alvo:  
Whois ou comando host (Linux/Unix).
  - Leitura de Banners para identificar servidores.
  - Fingerprint do SO
  - Enumeração dos Serviços e Versões
  - Enumeração das Informações dos Serviços.
  - Enumeração das Vulnerabilidades.
-

# Engenharia Social

---

- É uma forma pessoal, ilícita, utilizada por crackers, para adquirir disfarçadamente, quaisquer informações fundamentais para a manutenção da segurança de um sistema.
-

# Levantamento de Informações de Domínio

---

- ❑ Consulta na Base Whois (Internic).

```
whois <domínio>
```

```
whois <ip/domínio>@registro.br
```

```
fwhois <domínio>
```

```
xwhois <domínio> (ferramenta Linux)
```

- ❑ Procura na FAPESP (base do país).

```
http://registro.fapesp.br/
```

O domínio procurado está num provedor ou numa estação da própria empresa ???

---

# Levantamento de Informações de Domínio

---

- ❑ Consulta na base DNS pelos comandos `host` ou `dig` ou `nslookup` (utilitário que pesquisa DNS), no Linux.
  - ❑ Cada domínio possui uma base de dados DNS dos sub-domínios ali cadastrados.
-

# Comando host

---

- ❑ Consultando toda a base DNS:

```
>host -l -v -t any <empresa>.com.br
```

- ❑ Descobrendo qual é o servidor de email:

```
>host -t mx <empresa>.com.br
```

- ❑ Descobrendo os IPs de servidores DNS:

```
>host -t ns <empresa>.com.br
```

- ❑ Verificando os CNAME (quais o servidores FTP, Web e outros):

```
>host -t CNAME <empresa>.com.br
```

---

# Comando dig

---

- ❑ Buscando informações sobre o servidor DNS:

```
>dig -t ns <empresa>.com.br
```

- ❑ Buscando informações do registro MX:

```
>dig -t mx <empresa>.com.br
```

- ❑ **Buscando informações sobre o registro SOA:**

```
>dig -t soa <empresa>.com.br
```

---

# Comando nslookup

---

❑ Varredura nas informações de um domínio (consultando CNAME)

❑ CNAME = nomes canônicos

```
>nslookup
```

```
Set type=cname
```

```
www.<empresa>.com.br
```

---

# Levantamento de Informações de Domínio

---

- ❑ Levantamento de URL, através de consulta DNS, com a ferramenta IPZoner:

```
> ./IPZoner -s <ip_de> -t <ip_para>
```

- ❑ Exemplo:

```
> ./IPZoner -s 195.131.27.1 -t  
195.131.27.254
```

---

# Levantamento de Informações de Domínio

---

- ❑ Levantamento de rotas de pacotes numa/entre redes (quais servidores e roteadores existem, a topologia da rede e identificar a estrutura de segurança), através do utilitário **tracert** (Linux, Unix) ou **tracert** (Windows).
-

# Rota de pacotes

---

❑ Exemplo: `tracert vitima.com.br`  
router -> router -> máquina ->  
... > servidor

❑ Exemplo: Traceroute analisando uma porta.

```
tracert -p25 192.168.0.2
```

testa se há resposta na porta 25 (SMTP).

---

# Footprint

---

Leitura de Banners

# Leitura de Banners

---

## □ Identificando o servidor SMTP

- Com Netcat na porta 25.

```
> nc <ip> 25
```

- Com a ferramenta SMTPScan que utiliza um banco de dados de perfil de servidores SMTP.

```
> ./smtpscan inf.ufsc.br
```

---

# Leitura de Banners - DNS

---

□ Identificando a versão BIND em um servidor DNS:

- Com a ferramenta `dnsver.pl`

```
> ./dnsver.pl -t 50 -v <ip>
```

- Com a ferramenta `mig-named`

```
> ./mig-named -h <ip> -t 15 -d
```

---

# Leitura de Banners - DNS

---

- ❑ Identificando versão BIND de DNS, porta 53, com a ferramenta grabbb :

```
> ./grabbb -m -a 200. ... .  
          -b 200. ... .254 53  
200. ... .103:53  
200. ... .199:53  
200. ... .3:53  
> ./mig-named -h 200. ... .103 -t 50 -d  
[200. ... .103]:[53] 9.2.1  
> ...  
> ./mig-named -h 200. ... .3 -t 50 -d  
[200. ... .3]:[53] 9.2.1
```

- ❑ BIND (Berkeley Internet Name Domain) é uma implementação do Domain Name System (DNS)
-

# Identificando SSH, Web

---

- Identificando servidores SSH, porta 22:

```
> ./grabbb -m -a 200. ... .2 -b 200. ... .254 22
```

```
> ./scanssh 200. ... .0/24 | grep -v refused |  
  grep -v timeout | grep -v unreachable
```

- Identificando servidores Web:

```
> ./grabbb -m -a 200. ... .104 -b 200. ... .254 80
```

```
200. ... .195:80:
```

```
200. ... .106:80:
```

```
> httpdtype 200. ... .195
```

```
...
```

```
> httpdtype 200. ... .106
```

```
...
```

---

# Contra medidas – Leitura de Banners

---

- ❑ Utilizar a **obscuridade** por meio de eliminação de banners, restrição a consultas DNS e configurações que dificultem o levantamento das informações de banners.
  - ❑ Obscuridade é complemento de segurança;
  - ❑ Para agregar valor à segurança;
  - ❑ Ver [www.linuxsecurity.com.br](http://www.linuxsecurity.com.br)
  - ❑ Fazer atualizações de *patches*.
-

# Footprint

---

Conceituando Portas

Protocolos TCP, ICMP, UDP, IP

Base para Scanners de Porta

# Portas

---

- ❑ Sistema Operacional: kernel, serviços do sistema, serviços de comunicação (rede) e aplicações dos usuários, que podem se utilizar de serviços.
  - ❑ A forma de identificação de um ponto de acesso de serviço de rede (SAP, OSI) é a porta de protocolo TCP/IP.
  - ❑ Sockets TCP/IP = (IP, portas)
-

# Portas

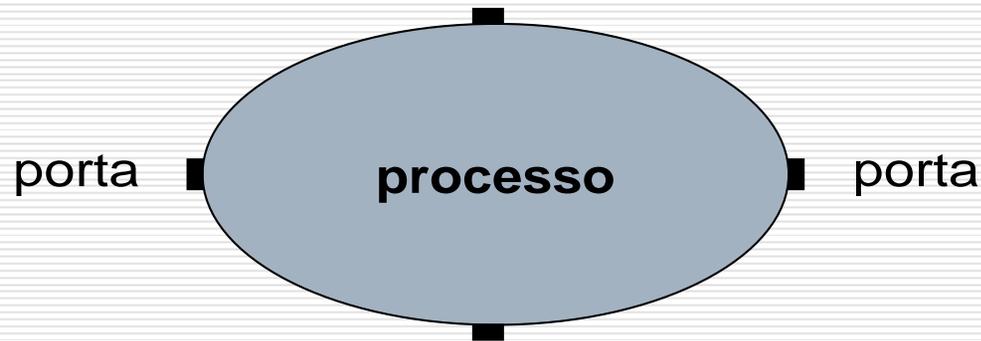
---

- ❑ A porta é a unidade que permite identificar o tráfego de dados destinado a diversas aplicações.
  - ❑ A identificação única de um processo acessando os serviços de rede TCP/IP é o *socket* TCP/IP, formado pelo par IP da máquina e a porta(s) usada(s) para acessar um serviço(s) de rede utilizado(s) por uma aplicação.
-

# Portas simultâneas

---

- Cada processo pode utilizar mais de uma porta simultaneamente (entrada, saída), mas, em um dado instante, uma porta só pode ser usada por uma aplicação.



# Portas

---

- ❑ Uma aplicação que deseje utilizar os serviços de rede deverá requisitar uma ou mais portas para realizar a comunicação.
  - ❑ A mesma porta usada por uma aplicação pode ser usada por outra, desde que a primeira tenha liberado aquela de utilização.
-

# Portas

---

- A forma de utilização de portas mostra uma distinção entre a parte cliente e a parte servidora de uma aplicação TCP/IP.
-

# Portas

---

- Uma aplicação-servidora deve utilizar um número de porta bem conhecido, de modo que um cliente qualquer, querendo utilizar os serviços do servidor, tenha que saber apenas o endereço IP da máquina onde o serviço está sendo executado.
-

# Portas

---

- ❑ A aplicação cliente pode utilizar um número de porta qualquer.
  - ❑ Os números de porta de 1 a 1023 são números bem conhecidos para serviços de rede, atribuídos pela IANA (Internet Assigned Numbers Authority).
-

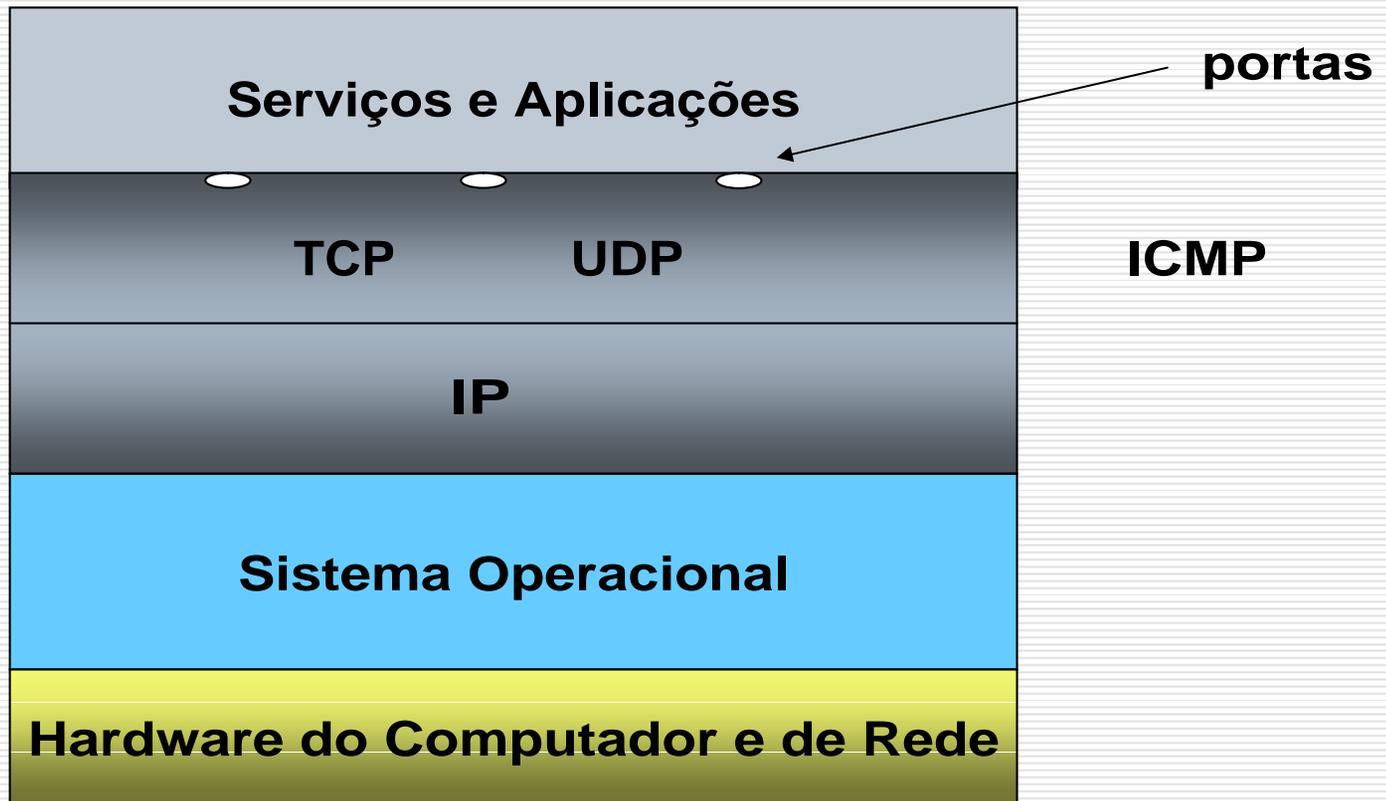
# Portas

---

- ❑ Os números de 1024 a 65535 podem ser atribuídos para outros serviços, e são geralmente usados pelos programas-cliente de um protocolo.
  - ❑ As portas servem para identificar o tipo de aplicação que gerou as mensagens de dados, e para qual tipo de aplicação as mensagens de dados devem ser entregues.
-

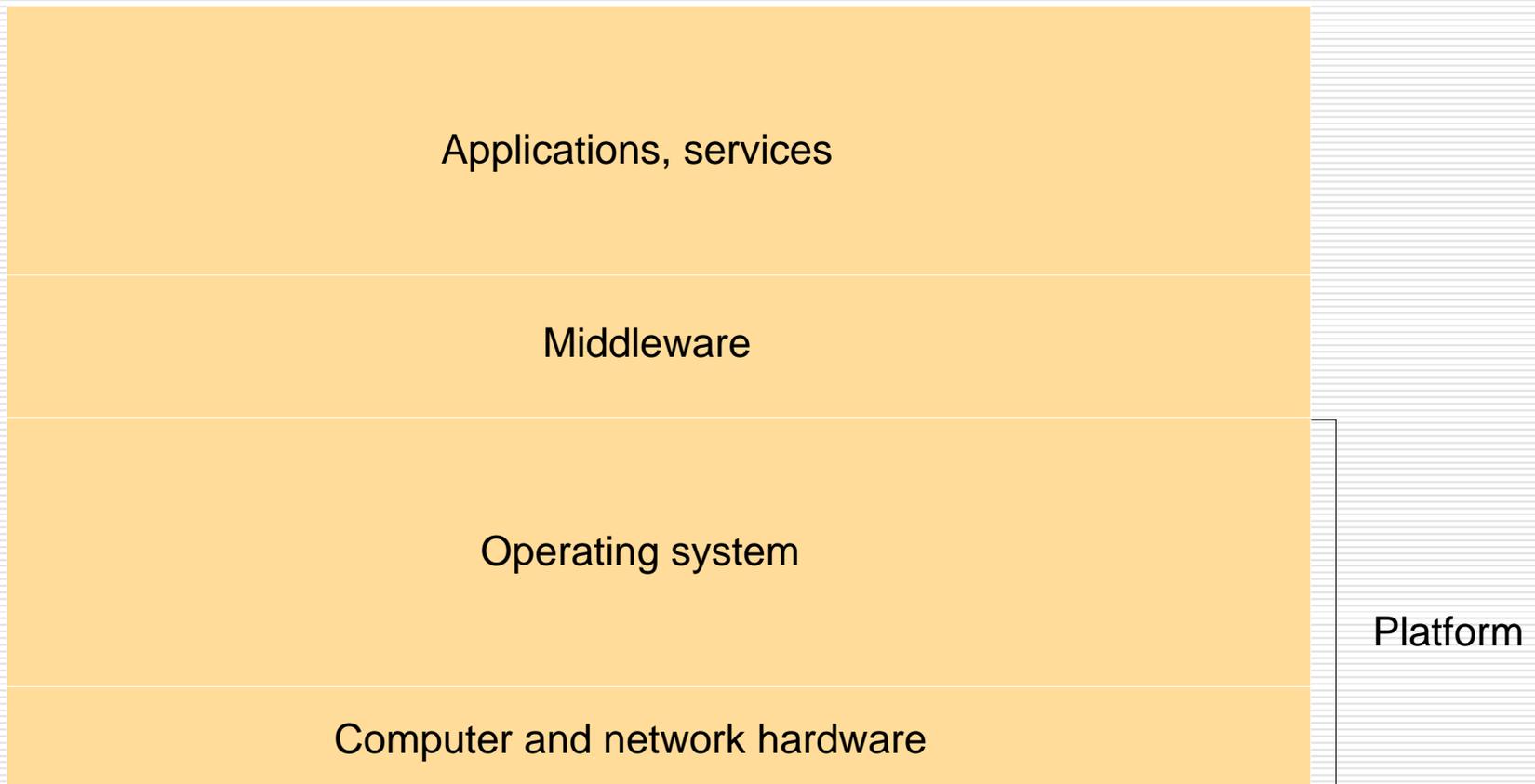
# Portas TCP

---



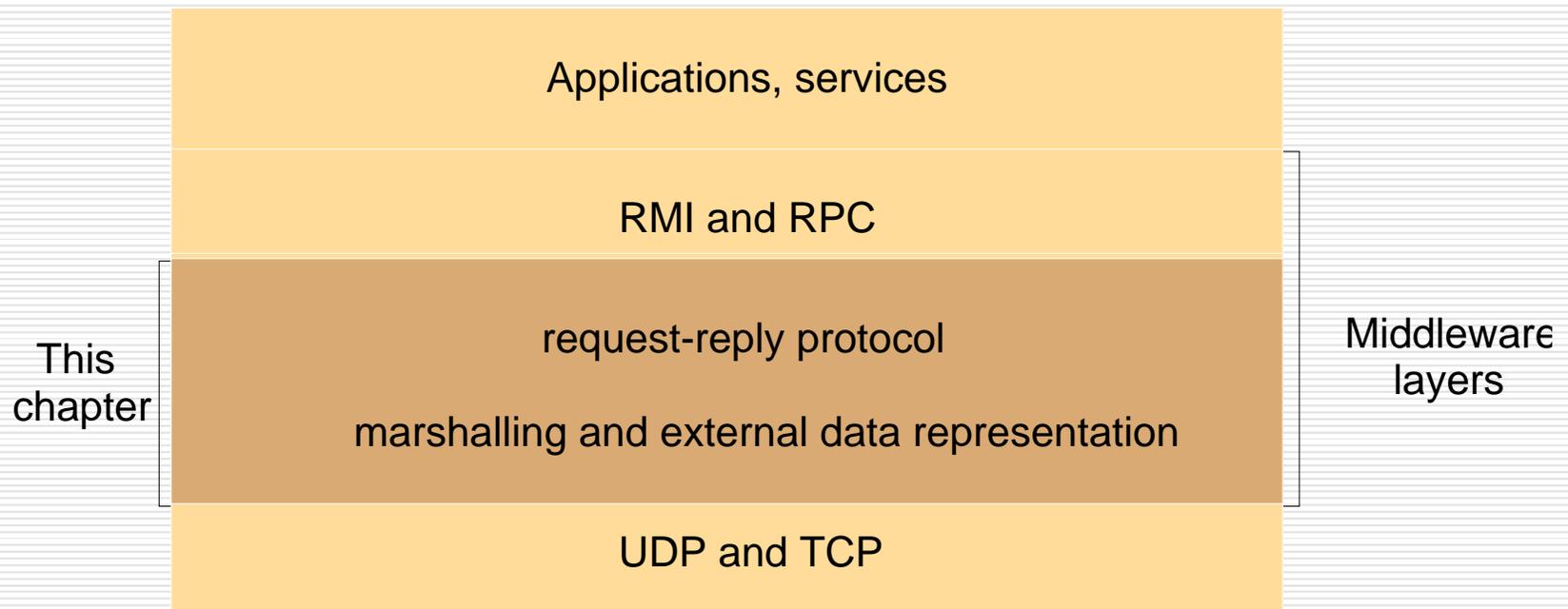
# A camada de aplicações e serviços

---

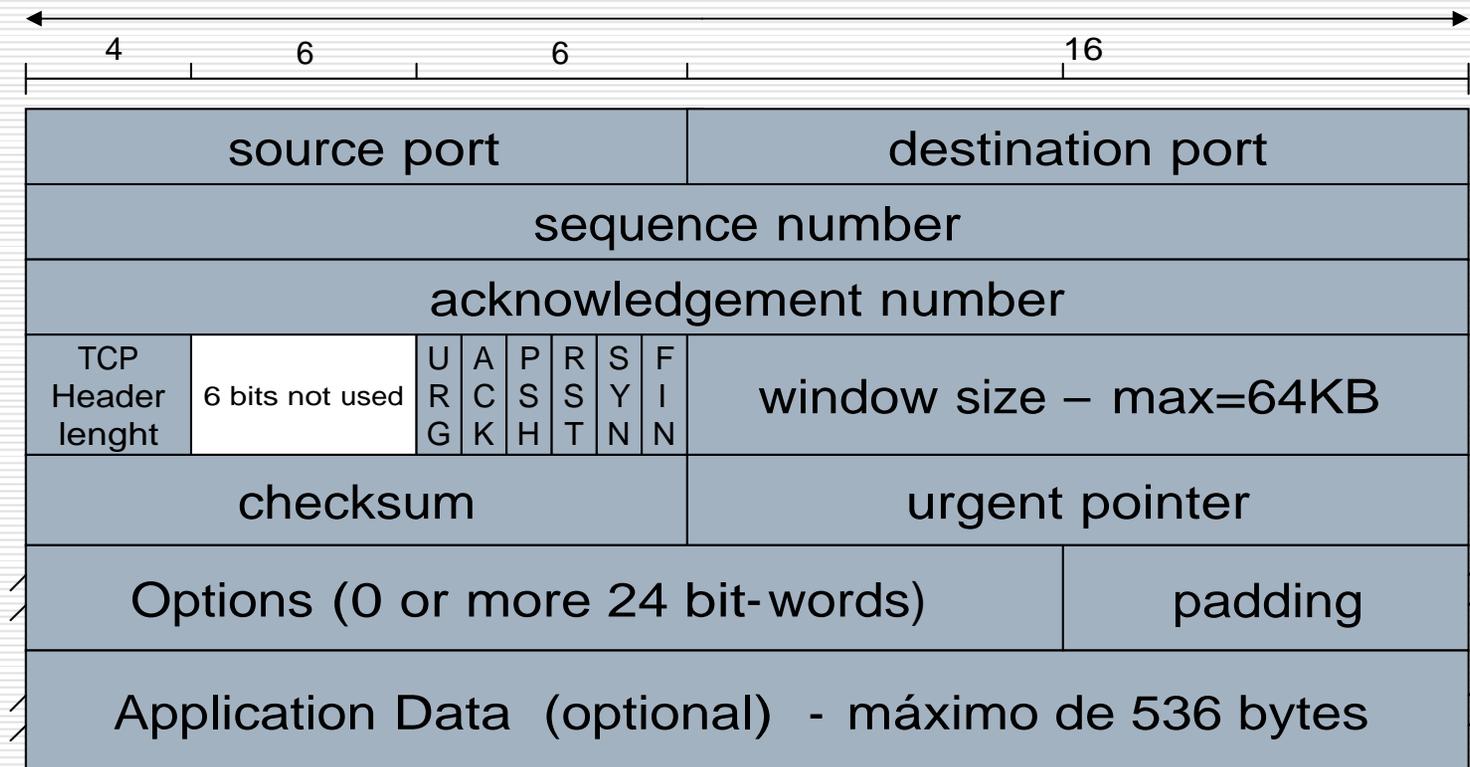


# TCP e UDP como camadas de software

---



# Protocolo TCP – Segmento TCP



# TCP – Bits de Controle

---

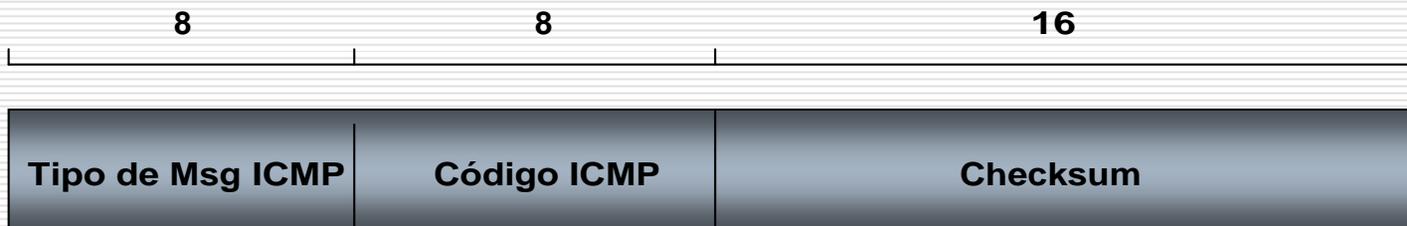
Bit	Significado
URG	O campo indicador Urgente é válido.
ACK	O campo número de confirmação é válido.
PSH	Força a entrega de dados.
RST	Reiniciar a conexão.
SYN	Sincronismo, determina o número de sequência inicial.
FIN	O transmissor chegou ao fim de seus dados.

---

# Protocolo ICMP

---

- ❑ Encapsulado no protocolo IP, mas não é um protocolo de alto nível (TCP, UDP).



---

<b>Valor</b>	<b>Alguns Tipos de mensagem ICMP</b>
<b>0</b>	<b>Resposta à mensagem de Echo</b>
<b>3</b>	<b>Aviso de destino inalcançável</b>
4	Redução da Velocidade de Transmissão
5	Solicitação de Redirecionamento
<b>8</b>	<b>Mensagem de Echo</b>
11	Tempo de Vida Excedido (Time To Live)
12	Problema nos parâmetros
...	...

---

# Bits de Varredura

---

- ❑ Varreduras usando TCP usam os bits de controle:

SYN, ACK, RST, FIN, URG, PSH

- ❑ Varreduras usando ICMP usam pacotes IP contendo ICMP tipo 3.
-

# Protocolo UDP

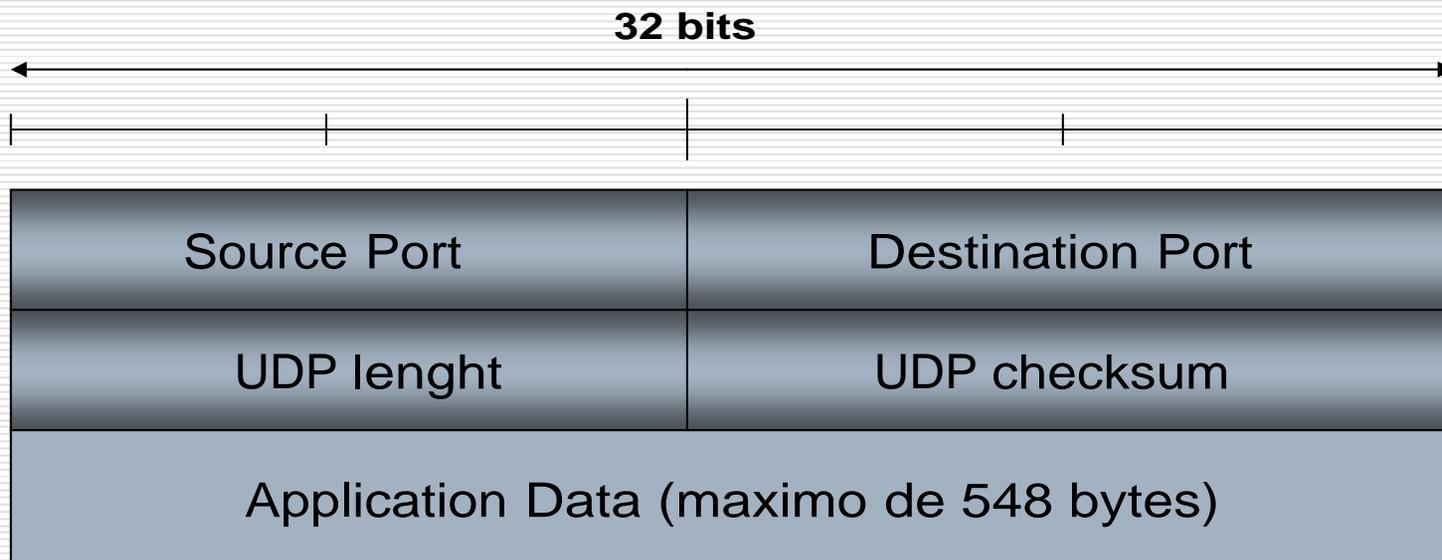
---

- ❑ *Suite* de protocolos Internet.
  - ❑ User Datagram Protocol (*RFC 768*).
  - ❑ Um protocolo de transporte sem conexão.
  - ❑ Provê um modo de aplicações enviarem datagramas UDP encapsulados em pacotes IP.
  - ❑ Muitas aplicações que têm um *request* e um *response* usam UDP (Echo, Whois, DNS, ... ).
-

# O segmento UDP

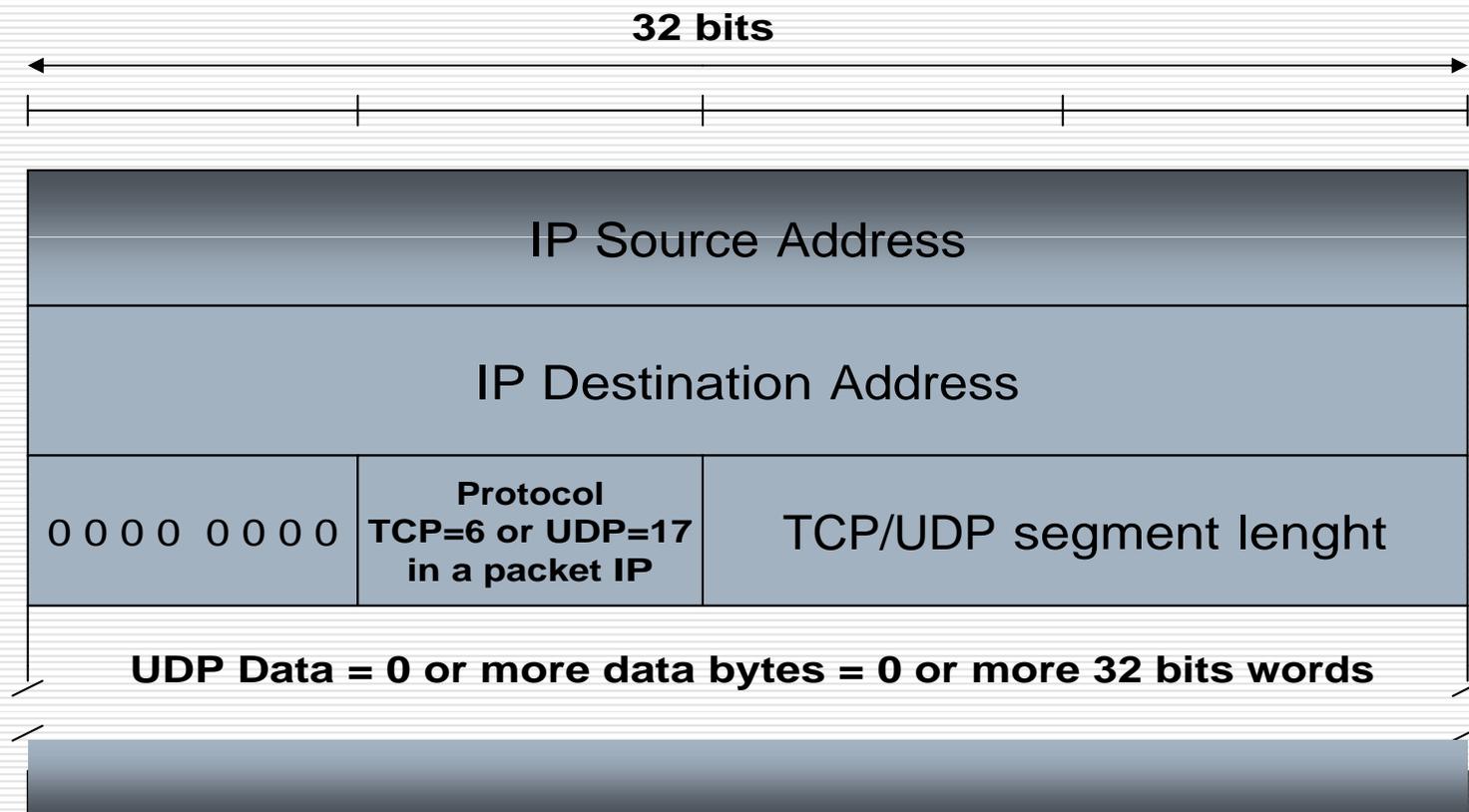
---

- Um segmento UDP consiste de um cabeçalho de 8 bytes seguido por dados da aplicação.



# O Pseudo Cabeçalho TCP/UDP

---



# Estrutura de um pacote IPv4

---

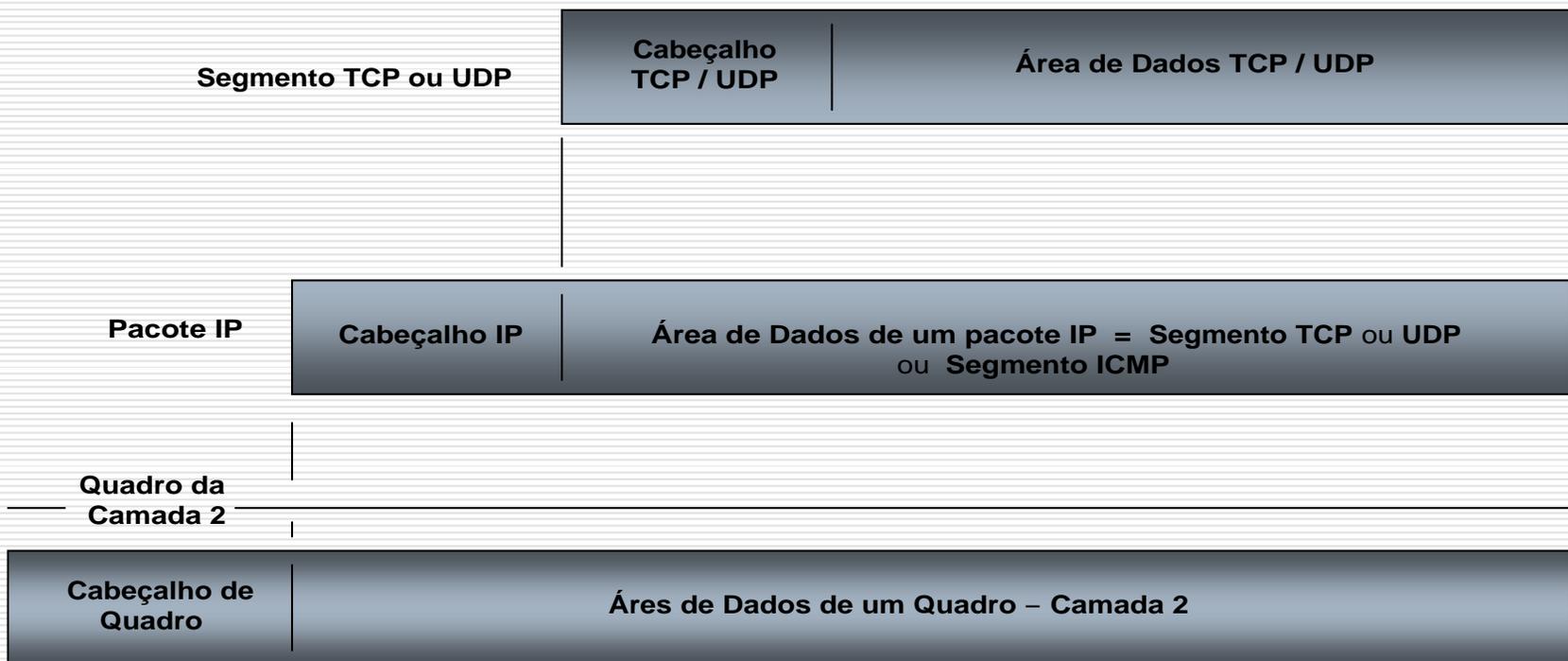
Versão (4 bits)
Tamanho do Cabeçalho (4bits)
Tipo de Serviço (1 byte)
Tamanho Total (4 bytes)
Identificação (4 bytes)
Flags (3 bits)
Deslocamento do Fragmento (13 bits)
Tempo de Vida (1 byte)
Protocolo TCP / UDP / ICMP (1 byte)
Checksum do Cabeçalho (4 bytes)
Endereço IP de Origem (4 bytes)
Endereço IP de Destino (4 bytes)
Opções + Padding (4 bytes – opcional)
Dados TCP / UDP / ICMP (até 65.511 ou 65.515 bytes)

← Segmentos: TCP ou UDP ou ICMP

---

# Encapsulamento de Segmentos

---



# Pseudo Cabeçalho

---

- ❑ Existe, apenas, para efeito de cálculo do checksum.
  - ❑ **Não é transmitido.**
  - ❑ O checksum do TCP é calculado da mesma forma que no UDP.
  - ❑ O checksum é calculado somando-se o cabeçalho, o pseudo-cabeçalho e o campo de dados.
-

# Footprint

---

## **Enumeração dos Serviços e Versões Scanners de Porta**

Visto como **ameaça** ou  
**auto-monitoramento**

# Scanners de Portas

---

- ❑ Pesquisam faixas de endereços IP.
  - ❑ Descobrem portas abertas (que têm serviços rodando).
  - ❑ Informações sobre o Sistema Operacional de uma máquina alvo (Fingerprint).
-

# Scanner Nmap

---

- ❑ Nmap (<http://www.nmap.org>)
  - ❑ Código Aberto.
  - ❑ Licença GNU GPL.
  - ❑ Auditoria de Sistemas.
  - ❑ Pode ser usado para **Footprint** e **Fingerprint**.
-

# Mostrando o Nmap

---

```
# /usr/local/nmap -O ganassi
```

```
Starting nmap V. 2.53 (www.insecure.org/nmap/)
```

```
Interesting ports on ganassi (10.8.10.231):
```

```
(The 1515 ports scanned but not shown below are in state: closed)
```

Port	State	Service
7/tcp	open	echo
9/tcp	open	discard
13/tcp	open	daytime
19/tcp	open	chargen
21/tcp	open	ftp
23/tcp	open	telnet
25/tcp	open	smtp
37/tcp	open	time
79/tcp	open	finger
111/tcp	open	sunrpc
512/tcp	open	exec
513/tcp	open	login
514/tcp	open	shell
515/tcp	open	printer

---

# Técnicas de **Fingerprint**

---

Técnica de levantamento de informações para **identificar o sistema operacional** da máquina-alvo.

# Fingerprint

---

- ❑ Informação fundamental para um invasor buscar uma possibilidade de intrusão.
  - ❑ Técnicas Clássicas.
  - ❑ Técnicas mais elaboradas.
  - ❑ Crackers e Script Kiddies utilizam ferramentas: **Queso**, **Nmap**.
  - ❑ **Queso** foi projetada para fingerprint.
  - ❑ **Nmap** pode fazer fingerprint na pilha TCP do host-alvo (usando UDP, TCP, ICMP).
-

# Lembrando o conceito de intrusão

---

- ❑ **Análise da Vulnerabilidade** (descobrir o melhor caminho para chegar até a invasão).
  - ❑ **Preparação das Ferramentas** (constrói ou escolhe as ferramentas para a invasão).
  - ❑ **Ameaça ou Tentativa** (quando o invasor pula o muro).
  - ❑ **Ataque** (concretiza o arrombamento).
  - ❑ **Intrusão ou Invasão** (quando obtém sucesso).
-

# Idéia básica para uma intrusão

---

- ❑ Ao **determinar qual SO** está rodando, o invasor pode **organizar suas ferramentas** de acordo com a plataforma-alvo.
  - ❑ O invasor pode ter como objetivo, “**rootear**” a máquina-alvo, e deve sempre saber as **diferenças dos formatos binários** de cada sistema.
-

# Idéia básica para uma intrusão

---

- ❑ O invasor tem em mente que, ao saber o SO de um host-alvo, ele pode **visar um serviço** do respectivo sistema,
  - ❑ descobrir uma vulnerabilidade desse serviço, e tendo em mãos um **exploit funcional para explorar esse serviço**,
-

# Idéia básica para uma intrusão

---

- ❑ Ele terá uma oportunidade que lhe permitirá “**rootear**” (assumir o perfil de administrador com senha de root).
-

# Investidas Errôneas

---

- ❑ Um investida errônea sobre o serviço pode tirá-lo do ar e/ou chamar a atenção do administrador.
  - ❑ Casos freqüentes de queda de serviços, por razões desconhecidas: verificação dos arquivos de *log* do servidor, Firewall e IDS.
-

# Formas de Fingerprint

---

- ❑ Técnicas Clássicas
  - ❑ Fingerprint com **Cheops**
  - ❑ Fingerprint com **Nmap** ou **Nmap** e **Nift**
  - ❑ UDP Echo
  - ❑ TCP Syn
  - ❑ TCP Echo
  - ❑ TCP Ack
  - ❑ ICMP Echo
  - ❑ Usar ferramentas como **snmpwalk** ou **LANguard** sobre servidores habilitados com SNMP e configurados de forma padrão.
-

# Fingerprint com **Cheops**

---

- ❑ **Cheops** é um programa mapeador de redes pequenas, que tem vários recursos, entre eles, a capacidade de fazer *fingerprint*.
  - ❑ Não identifica todos os sistemas remotos ...
-

# Fingerprint com **Nmap**

---

## □ *Fingerprint* através da Pilha TCP/IP

Extrair informações de uma máquina através das características implementadas em sua pilha TCP/IP.

---

# Fingerprint com Nmap

---

- ❑ `nmap-os-fingerprints` (nome do arquivo dos perfis de SOs)
  
  - ❑ Para usar o recurso de Fingerprint, utilizar a opção `"-O"`:  

```
nmap -O <ip>
```
  - ❑ Fingerprint em uma única porta:  

```
nmap -O -p80 <ip>
```
  - ❑ Fingerprint com modo de varredura máxima:  

```
nmap -O -p21 -osscan_guess <ip>
```
  
  - ❑ 

```
nmap -n -p80 -PO -O --osscan_guess <ip>
```
  - ❑ 

```
nmap -n -P6001 -PO -O -osscan_guess localhost
```
-

# Fingerprint com **Nift**

---

- ❑ **Nift** é uma **ferramenta front-end** para **Nmap** e outras ferramentas.
  - ❑ Apresenta uma **interface gráfica**.
  - ❑ Tem recursos para varreduras de serviços, fingerprint e varredura ICMP.
  - ❑ O objetivo de **Nift** é identificar o alvo e enumerar serviços.
  - ❑ Download de Nift em: ....
-

# Fingerprint com Nmap

---

- ❑ Descobrir quais os respectivos SOs.

```
nmap -sS -p80 -O -v <host>
```

```
nmap -sS -p80 -O -ossan_guess -v <host>
```

- ❑ Fazendo um teste numa corporação de nome empresa. O parâmetro <empresa>.log é um arquivo de log.

```
nmap -sS -F -o <empresa>.log -v -O www.<empresa>.com/24
```

Este comando faz SYN scan nas portas conhecidas em (/etc/services), "loga" o resultado no arquivo <arquivo>.log e em seguida faz um scan do SO e um scan na classe "C". Veja o resultado: Site e o SO.

---

# Fingerprint com Nmap

---

- ❑ Quando é anunciado um “bug” de segurança, esses invasores podem ir a um *site de exploits* em busca de uma ferramenta para explorar tal “bug”.
  - ❑ “modus operandi do script kiddie”
-

# Footprint

---

## **Técnicas de Varreduras**

Enumeração dos Tipos de Serviços e Versões

Varredura de Portas → Serviços

Serviços → Varredura de Vulnerabilidades

# Enumeração

---

- Extração de informações do ambiente-alvo, como os serviços de rede TCP e UDP, que requerem portas.
-

# Enumeração dos Tipos de Serviços Disponíveis e Versões

---

- ❑ Varreduras de Portas Clássicas
  - ❑ Varreduras TCP, UDP, ICMP.
  
  - ❑ Port Scanners
    - NetStat (Windows)
    - Netcat
    - [Nmap](#)
    - Amap (ideal para leitura de *banners*)
    - Blaster
    - Hping2
  
  - ❑ Intrusão ou para Auto-Monitoramento
-

# Footprint

---

Enumeração de Informações dos  
Serviços

# Varreduras a partir de Serviços

---

- ❑ SMTP Scan (levanta dados a partir do serviço SMTP).
  - ❑ SMB Scan (compartilhamento Windows, em UNIX, provido pelo Samba).
  - ❑ RPC Scan (levanta dados a partir do serviço de RPC)
  - ❑ **Intrusões ou Auto-Monitoramento**
-

# Vulnerabilidades

---

- São as **falhas de segurança** em um sistemas de software ou de hardware que podem ser exploradas para permitir a efetivação de uma **intrusão**.
-

# Footprint

---

Descoberta de vulnerabilidades

# Um scanner de vulnerabilidades

---

- ❑ Nessus (<http://www.nessus.org>)
  - ❑ Scanner de segurança que identifica vulnerabilidades, e tenta testar as encontradas.
  - ❑ Administração Remota.
-

# Varredura de Vulnerabilidades

---

- ❑ Enumeração das falhas e configurações padrões dos serviços.
  - ❑ Serve para concretizar **ataques**:  
são usados *Exploits* (ferramentas para a exploração de vulnerabilidades) para os respectivos serviços levantados.
  - ❑ Ou para realizar **Auto-Monitoramento**
-

# Mostrando o Nessus

---

```
# nessus -T text localhost 1241 noorder targetfile outfile
```

---

# Mostrando o Nessus

---

```
Nessus Scan Report
```

```
-----
```

## SUMMARY

- Number of hosts which were alive during the test : 1
- Number of security holes found : 2
- Number of security warnings found : 15
- Number of security notes found : 1

## TESTED HOSTS

```
192.168.0.90 (Security holes found)
```

---

# Mostrando o Nessus

---

## DETAILS

```
+ 192.168.0.90 :
. List of open ports :
  o unknown (161/udp) (Security hole found)
  o unknown (32779/udp) (Security warnings found)
  o unknown (32775/tcp) (Security warnings found)
  o unknown (32776/udp) (Security warnings found)
  o unknown (32778/udp) (Security warnings found)
  o unknown (32774/udp) (Security hole found)
  o unknown (32777/udp) (Security warnings found)
  o unknown (32780/udp) (Security warnings found)
  o unknown (32775/udp) (Security warnings found)
  o lockd (4045/udp) (Security warnings found)
  o unknown (32781/udp) (Security hole found)

. Vulnerability found on port unknown (32774/udp) :

  The sadmin RPC service is running.
  There is a bug in Solaris versions of
  this service that allow an intruder to
  execute arbitrary commands on your system.

  Solution : disable this service
  Risk factor : High
```

# SUSSEN - Interface para Nessus

---

- ❑ Um cliente não oficial para o Nessus, denominado **SUSSEN**:
  - ❑ Integração com MySQL Server V4.0, como backend.
  - ❑ Suporte a múltiplos servidores Nessus.
  - ❑ Suporte a geração de múltiplos relatórios.
  - ❑ Baseado em GNOME/Gtk+ 2.2 APIs.
-

# SUSSEN - Interface para Nessus

---

- Integração com ajuda de manual on-line.
  - Política de gerenciamento de plugins e scanners de porta.
  - Suporte a internacionalização e localização.
  - Suporte à XML.
  - [http:// .....](http://.....)
-

# Referências para Scanners

---

- ❑ Noordergraaf, Alex. Enterprise Server Products. How Hackers Do It: Trick, Tools and Techniques. Sun BluePrints™ OnLine – May, 2002:  
<http://www.sun.com/blueprints>
  - ❑ CERT: <http://www.cert.org>.
  - ❑ Nessus: <http://www.nessus.org>
  - ❑ Nmap: <http://www.nmap.org>
  - ❑ Serafim, Vinícius da Silveira. Atacantes: Suas principais técnicas e ferramentas. Gseg - UFRGS.  
<http://www.inf.ufrgs.br/~gseg/>
  - ❑ CVE: <http://cve.mitre.org>.
-

# Ameaças

---

- ❑ **Footprint e Fingerprint** são consideradas aqui como **ameaças**.
  - ❑ Footprint e Fingerprint: etapas da **anatomia de ataques**.
-