Economia com OpenBSD + PF + CARP

Humberto Sartini
http://web.onda.com.br/humberto

Palestrante

Humberto Sartini

- Analista de Segurança do Provedor OndaRPC
- Participante dos projetos:
  - Rau-Tu Linux (http://www.rau-tu.unicamp.br/linux)
  - HoneypotBR (http://www.honeypot.com.br/)
  - RootCheck (http://www.ossec.net/rootcheck/)
- Participante do:
  - IV e V Fórum Internacional de SL
  - Conferência Internacional de SL (Curitiba - 2003)
  - Conisli (São Paulo/SP – 2004)
  - Latinoware (Curitiba - 2005)

Tópicos

- História do OpenBSD
- Evolução do OpenBSD
- Características do OpenBSD
- Filtro de Pacotes (PF)
- Common Address Redundancy Protocol (CARP)
- Quality Of Services (QoS)
- Estudo de Caso

História do OpenBSD

- Por volta de 1970 várias licenças do Unix (AT&T) foram doadas para algumas universidades norte americanas

- O único problema era a restrição que a licença impunha

- Criadores do Unix, programadores e a universidade formaram um grupo para adicionar extensões e novidades ao Unix

História do OpenBSD

- Houve a adição do TCP/IP diretamente no kernel do sistema operacional, controle de processos, memória virtual, sistemas de arquivos novos e outros recursos

- Tecnicamente BSD é o nome dado aos avanços e melhorias do sistema operacional UNIX realizado pelo grupo

História do OpenBSD

- Surgiu o grupo NetBSD para guardar, organizar, ajudar a manter o sistema, além de liberar novas releases

- O grupo FreeBSD formou-se poucos meses depois do NetBSD e tem como objetivo o suporte a arquitetura i386

História do OpenBSD

- Em meados dos anos 90, o OpenBSD se formou a partir de um “fork” do NetBSD por divergências nas políticas de segurança e modelo de desenvolvimento

- Theo de Raadt, atual líder, dividiu o NetBSD em OpenBSD no dia 18 de outubro de 1995, as 08:37, quando fez o primeira ramo de desenvolvimento no CVS da árvore do NetBSD

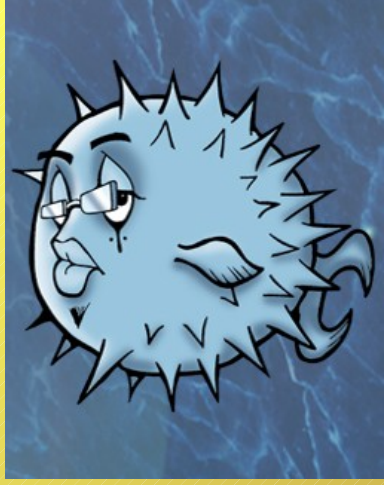
História do OpenBSD

- A primeira release foi a 2.0, em 1996

- No caso do OpenBSD releases são as liberações feitas a cada período de desenvolvimento, ou seja, de seis em seis meses

- A versão atual 3.8 foi liberada em 01/11/2905 e a próxima está prevista para 01/05/2006.

Evolução do OpenBSD



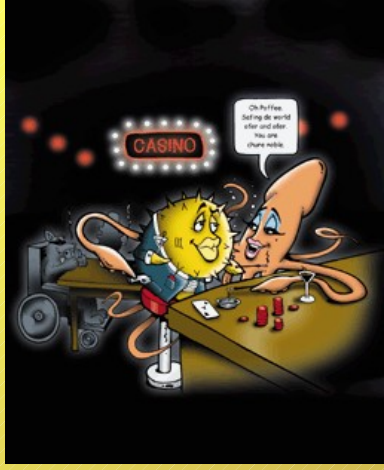
- OpenBSD 3.0- E-Reailed (OpenBSD Mix)

  - OpenSSH Protocolo 1 e 2
  - Mudanças na documentação
  - Ports mais completo
  - Novo Filtro de Pacotes PF
  - Mais de 1000 pacotes pré compilados

Evolução do OpenBSD

- OpenBSD 3.1 - Systemagic
  - Melhoras do PF
  - AuthPF
  - Suporte a RAID
  - Bridge Wavelan
  - Mais de 1000 pacotes pré compilados

Evolução do OpenBSD



- OpenBSD 3.2 - GoldFlipper
  - Apache Chroot por padrão
  - Encriptação de Hardware Simétrico e Assimétrico
  - Systrace
  - Ferramentas contra possíveis Buffer Overflow
  - Mais de 1800 pacotes pré compilados

Evolução do OpenBSD

- OpenBSD 3.3 – Puff the Barbarian
  - ProPolice
  - W^X (W xor X)
  - Servidor X e Console X com mais segurança e privilégio separado
  - PF com novas melhorias
  - Mais de 2000 pacotes pré-compilados

Evolução do OpenBSD

- OpenBSD 3.4 – The Legend of Puffy Hood
  - Syslog com privilégios separados
  - strcpy, strcat, sprintf, vsprintf
  - Melhorias no ProPolices
  - Suporte a novos hardwares
  - Suporte Ready Only NTFS
  - Mais de 2400 pacote pré compilados

Evolução do OpenBSD

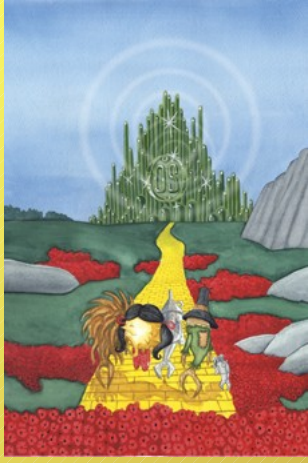
- OpenBSD 3.5 – Carp License and Redundancy must be free

  - Novas Plataformas Amd64, Cats e mvme88k
  - Otimização no PF
  - Carp e PFSync
  - BGPD
  - Mais de 2500 pacotes pré compilados

Evolução do OpenBSD

- OpenBSD 3.6 – Pond-erosa Puff (live)
  - Suporte SMP
  - OpenNTPD
  - Melhorias NFS
  - OpenSSH 3.9
  - Suporte a novos hardware
  - Mais de 2005 pacotes pré compilados

Evolução do OpenBSD



- OpenBSD 3.7 – Wizard of OS
  - Novas plataformas Zaurus e SGI
  - Suporte a diversos USB e Wireless
  - OSPFD
  - OpenSSH 4.1
  - Mais de 3000 pacotes pré compilados

Evolução do OpenBSD

- OpenBSD 3.8 – Hackers of the Lost RAID
  - bioctl – Gerência de RAID
  - ipsecctl – Gerência IPSEC
  - ifstated – Monitorar Interface
  - sasyncd – Sincronismo IPSec
  - Novos hardwares
  - Mais de 3200 pacote pré compilados

Características do OpenBSD

- Focado na portabilidade, padronização (POSIX, ANSI, X/Open, etc), exatidão, segurança proativa e criptografia integrada

- Sua inspiração é ser o número um da indústria de segurança

- Desenvolvido por voluntários e os fundos são provenientes de CD's, camisetas e doações

Características do OpenBSD

- Até Junho de 2002, o site do OpenBSD mantinha o slogan "No remote hole in the default install, in nearly 6 years." Depois que uma falha foi descoberta no OpenSSH, foi alterado para "Only one remote hole in the default install, in more than 8 years".

Características do OpenBSD

- Somente 04 (quatro) alertas de segurança na versão 3.7 (CVS, Sudo, Zlib e Zlib)

- Impossibilidade de Buffer Overflow devido a novas tecnologias
  - strlcpy() e strlcat()
  - Separação e Revogação de Privilégios
  - Cadeia Chroot

Características do OpenBSD

- Todos os sistemas operacionais modernos utilizam o código BSD em alguma parte de seu desenvolvimento

- A Internet esta apoiada na pilha TCP/IP que o time BSD ajudou a desenvolver sendo um advento revolucionário por ter incorporado a conexão diretamente no kernel possibilitando um aumento da performance e que os sistemas operacionais entrassem na era do “networking”

Características do OpenBSD

- O código está apoiado na licença BSD, realmente livre, que permite qualquer um fazer o que bem entender com o código, incluindo ganhar dinheiro licenciando o código e usá-lo em outro trabalho

Filtro de Pacotes - PF

- Lançado como parte integrante do OpenBSD 3.0 em dezembro de 2001

- Necessário após mudança de licença do projeto IPFilter, que acompanhava a instalação base do OpenBSD

- Durante algumas semanas o OpenBSD ficou sem firewall na instalação padrão

Filtro de Pacotes - PF

- NORMALIZAÇÃO DE PACOTES

Ajuda a evitar este tipo de ataque normalizando o tráfego passado em ambas as direções. A identificação de sistemas operacionais pode ser frustrada utilizando esta técnica provendo uma maior segurança em uma rede. Este tipo de característica ajuda a fazer o pf único, no meio de dispositivos de filtragem comerciais

Filtro de Pacotes - PF

- IMPRESSÕES DIGITAIS BASEADAS NO SO

  Baseado no p0f, reconhece assinaturas de sistemas operacionais. Útil para barrar acessos de worms, vírus e afins

- BALANÇO DE CARGA

  Possibilidade de que links possam ser balanceados e roteando conexões entre eles, podendo haver um aumento de performance.

Filtro de Pacotes - PF

- MODULATE STATE

  Mesmas características que o KEEP STATE, exceto que trabalha somente com TCP e o número de seqüência inicial é fortemente aleatório

- KEEP STATE

  Funciona com TCP, UDP e ICMP, mantendo a tabela de conexão e não processando as regras de um pacote caso faça parte da tabela de conexão

Common Address Redundancy Protocol(CARP)

- Introduzido no OpenBSD 3.5 em Outubro de 2003

- Protocolo que permite múltiplos hosts na mesma rede local compartilharem um mesmo endereço IP

- É feito para prover grande segurança e é um protocolo independente

- Permite load balance, alta disponibilidade e substitui VRRP

Common Address Redundancy Protocol(CARP)

- Intervalos de advertisement configuráveis (quem mais adverte vira master)

- Usado em conjunto com o PF e PFSYNC provê alta disponibilidade

- Como é um load balance baseado em Layer 2, uma das máquinas pode receber mais carga que a outra.

Common Address Redundancy Protocol(CARP)

- Exemplos:

- **HOST01**

➔ /etc/hostname.carp0:
inet 10.0.0.1 255.255.255.0 10.0.0.255 vhid 1 pass minhasenha

- **HOST02**

➔ /etc/hostname.carp0:
inet 10.0.0.1 255.255.255.0 10.0.0.255 vhid 1 advskew 100 pass minhasenha

Quality of Services (QoS)

- O ALTQ (Alternate Queueing for BSD UNIX) suporta vários algortimos e iremos focar nos seguintes:
  - Filas Baseadas em Classe (CBQ)
  - Fila de Prioridade(PRIQ)
  - Hierarchical Fair Service Curve (HFSC)

Quality of Services (QoS)

- CBQ (Filas Baseadas em Classe ou Class Based Queuing)
  - Divide a largura de banda em diversas filas ou classes
  - Endereço de origem ou destino, portas, protocolos, etc
  - Vários níveis de prioridade

Quality of Services (QoS)

- CBQ (Filas Baseadas em Classe ou Class Based Queuing)
  - Forma hierárquica de organização

Fila Raiz (10Mbps)
- Fila ssh (2Mbps, priority 1)
- Fila http (5Mbps, priority 4)
- Fila mail (2Mbps, priority 5)
- Fila ftp (1Mbps, priority 7)

Quality of Services (QoS)

- PRIQ (Fila de Prioridade ou Priority Queuing)
  - Uma fila com alta prioridade é "*sempre*" processada na frente de uma fila com prioridade menor
  - A fila raiz é definida onde é configurado o total de banda disponível e, então, subfilas são definidas sob a raiz

Quality of Services (QoS)

- PRIQ (Fila de Prioridade ou Priority Queuing)

  Fila Raiz (10Mbps)
  - Fila ssh (priority 1)
  - Fila http (priority 4)
  - Fila mail (priority 5)
  - Fila ftp (priority 7)

Quality of Services (QoS)

- HFSC (Hierarchical Fair Service Curve)
  - Mesmas funcionalidades do CBQ
  - Compatilhamento de Banda
  - Algoritmo mais otimizado

Quality of Services (QoS)

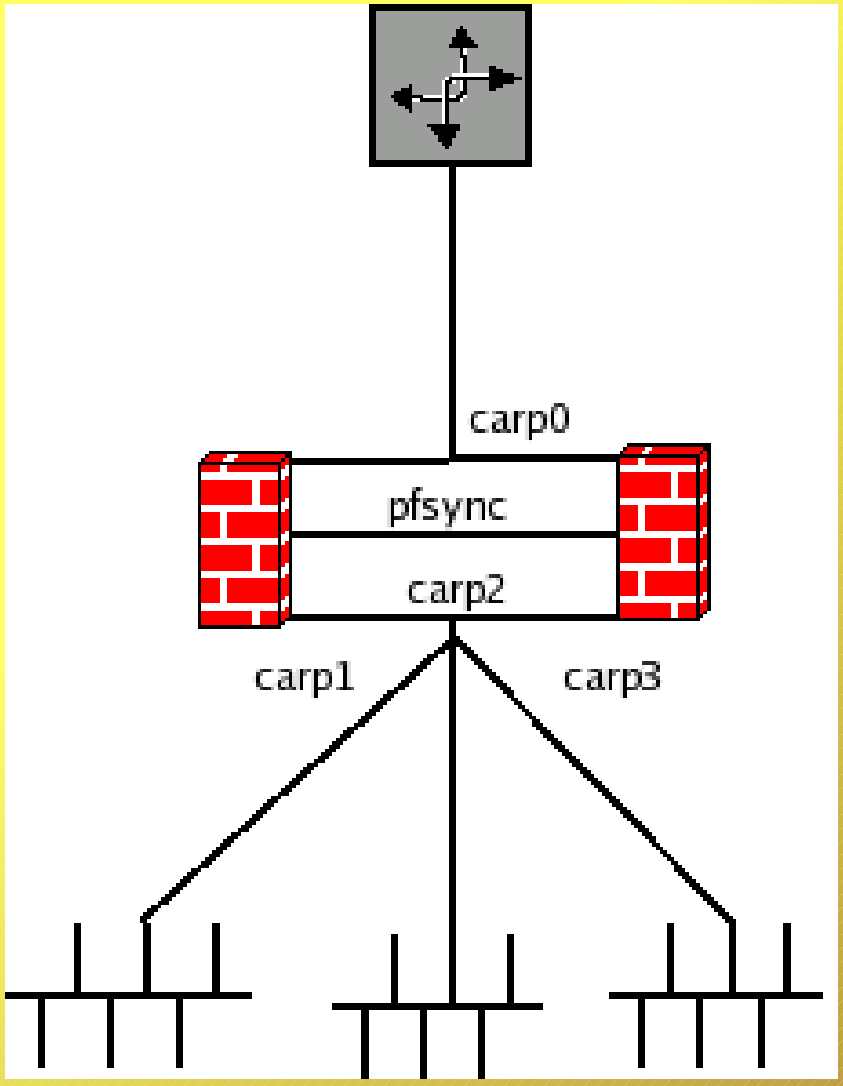
- HFSC (Hierarchical Fair Service Curve)

```
altq on { em0 } hfsc bandwidth 70Mb  qlimit 75
  queue { deflt, http, ssh, mail }
queue   deflt bandwidth 5% qlimit 75 priority 4
  hfsc(linkshare 2% default realtime 3% upperlimit
  5% red)
queue   http bandwidth 46% qlimit 75 priority 5
  hfsc(linkshare 5% realtime 39% upperlimit 46%
  red)
queue   ssh bandwidth 3% qlimit 75 priority 7
  hfsc(realtime 2% upperlimit 3% red)
queue   mail bandwidth 36% qlimit 75 priority 5
  hfsc(linkshare 5% realtime 29% upperlimit 36%
  red)
```

Estudo de Caso

- Substituição de Firewall comercial
- Necessidade de migração de hardware
- Não teria Alta disponibilidade
- Software Proprietário
- Custo elevado !!

Estudo de Caso



Estudo de Caso

- Limites de Conexões

  - set limit { states 80000, frags 30000, src-nodes 45000 }
  - set timeout { interval 10, frag 30 }
  - set timeout { tcp.first 60, tcp.opening 30, tcp.established 3600 }
  - set timeout { udp.first 20, udp.single 10, udp.multiple 15 }
  - set timeout { icmp.first 11, icmp.error 6 }
  - set timeout { other.first 40, other.single 20, other.multiple 30 }

Estudo de Caso

- Utilização de HFSC para QoS
- Regras Anti Spoof
- VPN – ISAKMPD – IPSEC
- Flags TCP

```
pass in quick on em0 proto tcp from any to 20.20.20.20
port { 80 443 } flags S/SA keep state queue http
```

- Regras Stateful Tracking Options

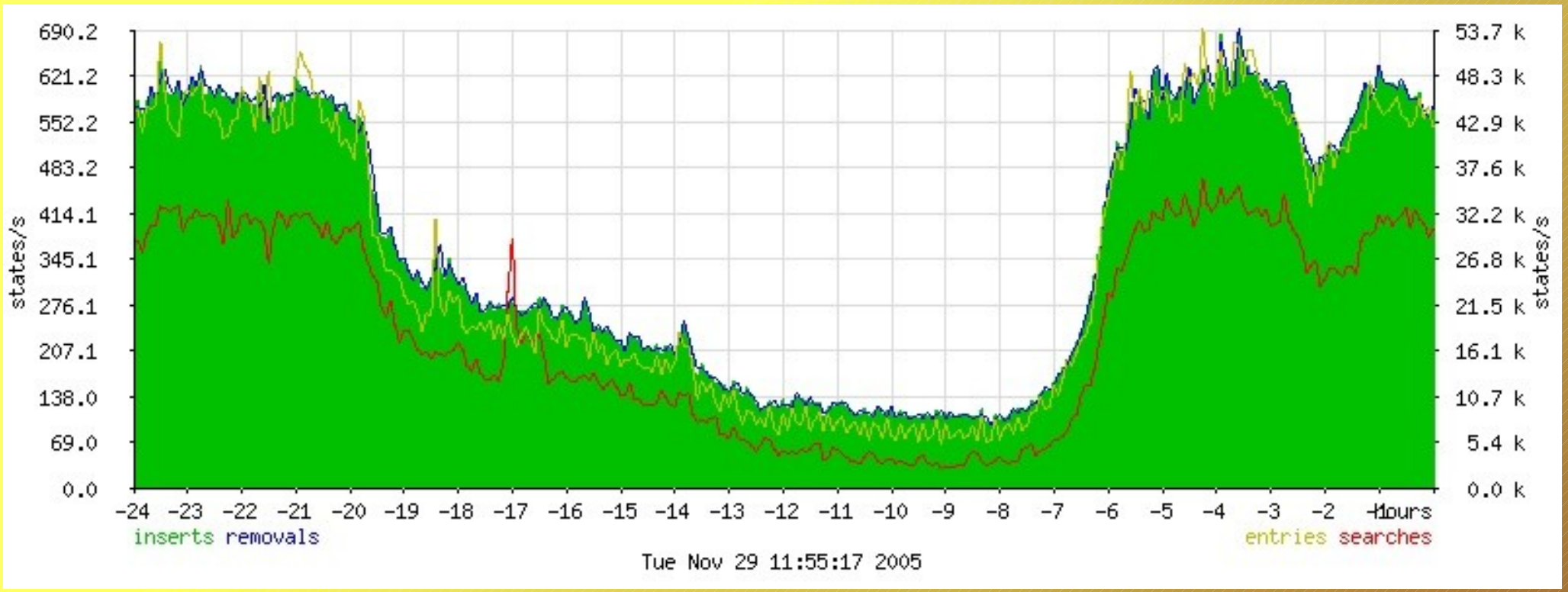
```
    pass in on $ext_if proto tcp to $web_server \
        port www flags S/SA keep state \
        (max 200, source-track rule, max-src-nodes 100,
max-src-states 3)
```

Estudo de Caso

- Ferramentas de Gerência
  - pfctl
    - Parte integrante do sistema, funciona em modo texto
  - mrtg
    - Gráfico de número de conexões

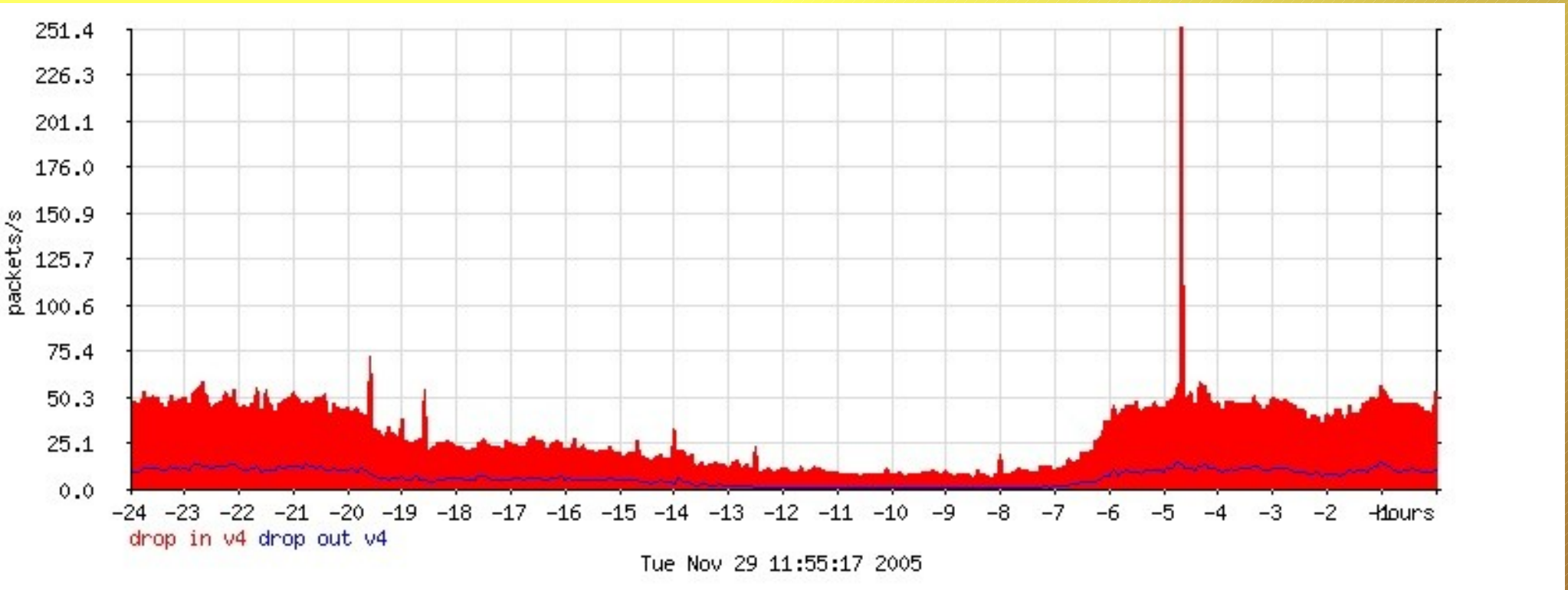
Estudo de Caso

- Ferramentas de Gerência – PFSysInfo



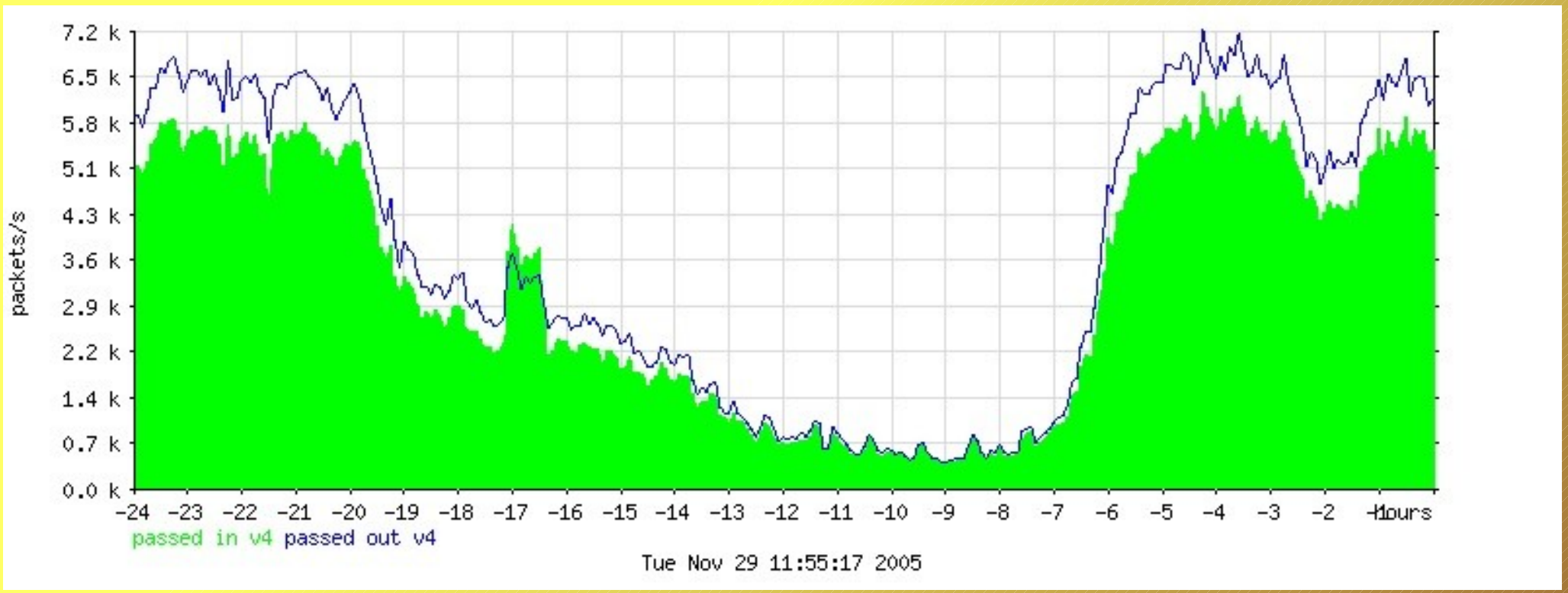
Estudo de Caso

- Ferramentas de Gerência – PFSysInfo



Estudo de Caso

- Ferramentas de Gerência – PFSysInfo



Estudo de Caso

- Ferramentas de Gerência – PFSysInfo

Contato

Através do site ou e-mail

http://web.onda.com.br/humberto

humberto@onda.com.br

Créditos

Sites consultados:

- OpenBSD

http://www.openbsd.org

- PFSysInfo

http://team.gcu-squad.org/~aflab/

- Site Pessoal João Henrique F. Freitas

http://paginas.terra.com.br/informatica/joaohf/openbsd/openbsd.html

Créditos

• Figura Slide 01:
http://www.openbsd.org/art/ramblo.jpg

• Figura Slide 09:
http://www.openbsd.org/images/Rock.jpg

• Figura Slide 10:
http://www.openbsd.org/images/Systemagic.jpg

• Figura Slide 11:
http://www.openbsd.org/images/MrPond.gif

• Figura Slide 12:
http://www.openbsd.org/images/Barbarian.gif

• Figura Slide 13:
http://www.openbsd.org/images/Hood.gif

• Figura Slide 14:
http://www.openbsd.org/images/Carp.gif

Créditos

- Figura Slide 15:
http://www.openbsd.org/images/Ponderosa.jpg

- Figura Slide 16:
http://www.openbsd.org/images/Wizard.jpg

- Figura Slide 17:
http://www.openbsd.org/images/Jones.jpg

- Outras Figuras:
  Arquivo Pessoal