

Impacto do Robustecimento de Sistemas no Desempenho de Serviços Internet

Sino 2006

Tiago Pedrosa
(pedrosa@ipb.pt)

Escola Superior de Tecnologia e de Gestão
Instituto Politécnico de Bragança

11 de Outubro de 2006

Estrutura

- 1 Introdução
- 2 Robustecimento
 - PaX
 - DAC e RBAC
 - Grsecurity
- 3 Testes
 - Cenário de Testes
 - Metodologia
 - Resultados
- 4 Conclusões
 - Trabalho Futuro

Estrutura

- 1 Introdução
- 2 Robustecimento
 - PaX
 - DAC e RBAC
 - Grsecurity
- 3 Testes
 - Cenário de Testes
 - Metodologia
 - Resultados
- 4 Conclusões
 - Trabalho Futuro

Introdução

O robustecimento de sistemas afecta o desempenho de serviços

- Quanto?
- Como?

Objectivos

- Sistemas de robustecimento – Prevenir ataques correntes e futuros
- Avaliar o mecanismo de aprendizagem automática
- Medir o impacto nos serviços de rede

Introdução

O robustecimento de sistemas afecta o desempenho de serviços

- Quanto?
- Como?

Objectivos

- Sistemas de robustecimento – Prevenir ataques correntes e futuros
- Avaliar o mecanismo de aprendizagem automática
- Medir o impacto nos serviços de rede

Estrutura

- 1 Introdução
- 2 Robustecimento
 - PaX
 - DAC e RBAC
 - Grsecurity
- 3 Testes
 - Cenário de Testes
 - Metodologia
 - Resultados
- 4 Conclusões
 - Trabalho Futuro

Robustecimento de sistemas

Tipicamente, o ciclo de protecção inclui:

- 1 Pesquisa/Descoberta de Falhas
- 2 Desenvolvimento de *Patch* de segurança
- 3 Aplicação do *Patch* de segurança no sistema

Relativamente a falhas não conhecidas/não previsíveis?

- Aprendizagem do comportamento "normal" do sistema
- Qualquer desvio, é considerado um ataque
- Mecanismos de *protecção de memória*
- É necessário aumentar a *granularidade de protecção*

Robustecimento de sistemas

Tipicamente, o ciclo de protecção inclui:

- 1 Pesquisa/Descoberta de Falhas
- 2 Desenvolvimento de *Patch* de segurança
- 3 Aplicação do *Patch* de segurança no sistema

Relativamente a falhas não conhecidas/não previsíveis?

- Aprendizagem do comportamento “normal” do sistema
- Qualquer desvio, é considerado um ataque
- Mecanismos de **protecção de memória**
- É necessário aumentar a **granularidade de protecção**

Robustecimento de sistemas

Tipicamente, o ciclo de protecção inclui:

- 1 Pesquisa/Descoberta de Falhas
- 2 Desenvolvimento de *Patch* de segurança
- 3 Aplicação do *Patch* de segurança no sistema

Relativamente a falhas não conhecidas/não previsíveis?

- Aprendizagem do comportamento “normal” do sistema
- Qualquer desvio, é considerado um ataque
- Mecanismos de **protecção de memória**
- É necessário aumentar a **granularidade de protecção**

Robustecimento de sistemas

Tipicamente, o ciclo de protecção inclui:

- 1 Pesquisa/Descoberta de Falhas
- 2 Desenvolvimento de *Patch* de segurança
- 3 Aplicação do *Patch* de segurança no sistema

Relativamente a falhas não conhecidas/não previsíveis?

- Aprendizagem do comportamento “normal” do sistema
- Qualquer desvio, é considerado um ataque
- Mecanismos de **protecção de memória**
- É necessário aumentar a **granularidade de protecção**

Protecção de memória – PaX

- 1 O PaX faz uso de páginas virtuais não executáveis
 - IA-32 não suporta o bit de executável
 - Não é possível distinguir páginas com código executável de páginas de apenas leitura
- 2 Fornece aleatoriedade para todo o espaço de endereçamento para binários ELF

Granularidade de Protecção – DAC e RBAC

Ambos requerem autenticação do utilizador

Discretionary Access Control

- Muito utilizado em sistemas Unix
- Baseia-se nas permissões de escrita, leitura e execução
- Estas permissões são afectadas ao dono do ficheiro, ao grupo do dono, e/ou a todos os utilizadores

Role-Based Access Control

- Constroi uma ACL de sistema na perspectiva do processo
- Efectua pesquisa linear sobre a ACL – tabelas de *hash*
- Pode ser configurado de forma automática (aprendizagem)
- Está a evoluir no sentido de permitir delegação, herança e conjuntos

Granularidade de Protecção – DAC e RBAC

Ambos requerem autenticação do utilizador

Discretionary Access Control

- Muito utilizado em sistemas Unix
- Baseia-se nas permissões de escrita, leitura e execução
- Estas permissões são afectadas ao dono do ficheiro, ao grupo do dono, e/ou a todos os utilizadores

Role-Based Access Control

- Constroi uma ACL de sistema na perspectiva do processo
- Efectua pesquisa linear sobre a ACL – tabelas de *hash*
- Pode ser configurado de forma automática (aprendizagem)
- Está a evoluir no sentido de permitir delegação, herança e conjuntos

Granularidade de Protecção – DAC e RBAC

Ambos requerem autenticação do utilizador

Discretionary Access Control

- Muito utilizado em sistemas Unix
- Baseia-se nas permissões de escrita, leitura e execução
- Estas permissões são afectadas ao dono do ficheiro, ao grupo do dono, e/ou a todos os utilizadores

Role-Based Access Control

- Constroi uma ACL de sistema na perspectiva do processo
- Efectua pesquisa linear sobre a ACL – tabelas de *hash*
- Pode ser configurado de forma automática (aprendizagem)
- Está a evoluir no sentido de permitir delegação, herança e conjuntos

Grsecurity

No âmbito deste trabalho usou-se o Grsecurity

- 1 Usa o PaX
- 2 Implementa RBAC
- 3 Configurável por auto-aprendizagem
- 4 Ferramenta de administração em *user-space*
- 5 Alteração das permissões da `/proc`

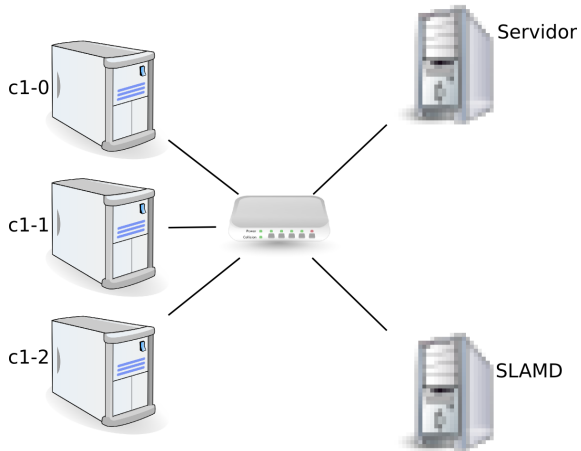
Grsecurity

- Auditoria, de entre muitos eventos auditados:
 - Exec, Chdir, Mount/Umount,
 - IPC, Sinais, Forks falhados,
 - Ptrace, Mudanças de hora/data, Execs dentro dos Chroots
- Prevenção
 - Utilização do PAX
 - Algumas syscalls foram robustecidas, entre as quais o Chroot, Ptrace, Mmap, Link/Unlink, Sysctl
- Isolamento
 - *Trusted Path Execution* – TPE
 - Remoção de variáveis de ambiente do tipo LD_*, que permite a execução aliatória de código, ex. LD_PRELOAD.
 - Testes de TPE à função Mmap
 - RBAC

Estrutura

- 1 Introdução
- 2 Robustecimento
 - PaX
 - DAC e RBAC
 - Grsecurity
- 3 Testes
 - Cenário de Testes
 - Metodologia
 - Resultados
- 4 Conclusões
 - Trabalho Futuro

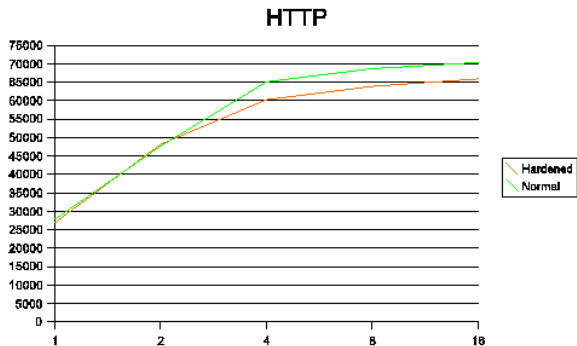
Cenário de Testes



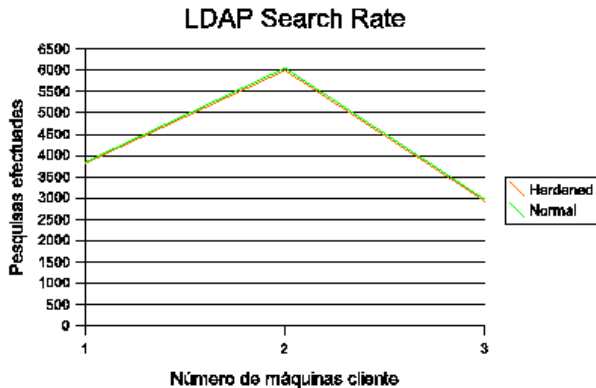
Metodologia

- 1 Instalação dos serviços a testar:
 - HTTP, SMTP, POP, LDAP
 - apache2, qmail com pop, openldap
- 2 Instalação e configuração do SLAMD (ferramenta de testes)
- 3 Fase de aprendizagem do RBAC
- 4 Gerar pedidos com base em (preferencialmente) threads
- 5 Introduzir novo nó cliente quando a carga de processador exceder os 80%
- 6 Aumentar o número de clientes enquanto o aumento de desempenho do servidor for de 5%

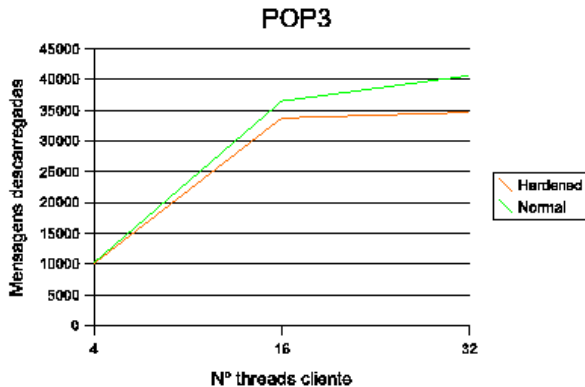
Resultados



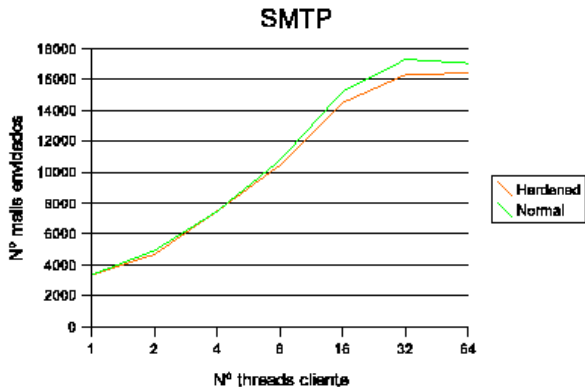
Resultados



Resultados



Resultados



Estrutura

- 1 Introdução
- 2 Robustecimento
 - PaX
 - DAC e RBAC
 - Grsecurity
- 3 Testes
 - Cenário de Testes
 - Metodologia
 - Resultados
- 4 Conclusões
 - Trabalho Futuro

Conclusões

O desempenho dos serviços HTTP, SMTP e LDAP é pouco afectado, o que não acontece com o serviço POP.

- 1 Serviços que utilizam memória partilhada intensivamente (HTTP, SMTP e LDAP) parecem sofrer menor quebra de desempenho;
- 2 O PaX parece ser o factor responsável pela degradação do desempenho devido à sobrecarga introduzida ao nível da gestão de memória.
- 3 Pode-se concluir que é conveniente optar por implementações de serviços que usem memória partilhada.
- 4 Quando tal não for possível, é necessário dar atenção especial à configuração do PaX.

Trabalho Futuro

- Pretendemos estender este estudo a outras arquitecturas e ambientes, tais como arquitecturas de 64 bits e máquinas virtuais.
- Adicionalmente, pensar em equacionar uma solução que permita uma gestão centralizada das regras do RBAC.