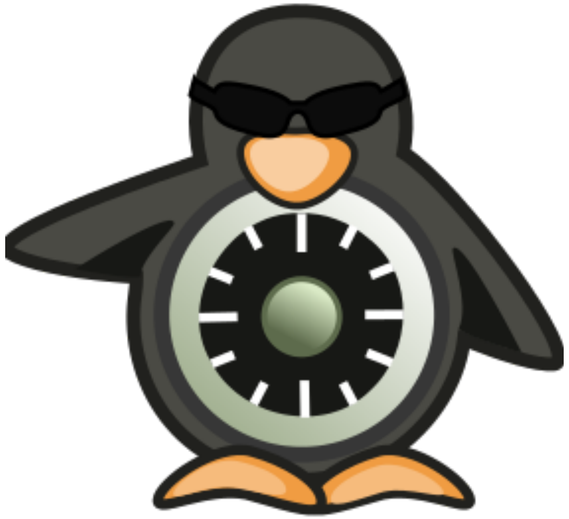


SELinux para Pessoas Normais

Introdução ao “Security-Enhanced Linux” sem causar AVCs

C0NFRAR1A

Carlos Rodrigues — Março de 2017

Porquê o SELinux?

- **Confinar serviços**

  …minimizar o impacto no sistema em caso de ataque
  …isolar instâncias entre si em cenários *multi-tennant*

- Confinar utilizadores

  …utilizadores com âmbito(s) limitado(s)
  …administradores para serviços específicos

- Controlar o acesso a informação sensível

  …com níveis de confidencialidade (ex. *public* ⇢ *top secret*)

Modelo de Acessos Tradicional

- O `root` tem controlo **total** sobre o sistema

  …processos a correr como `root` não têm quaisquer restrições
  …*capabilities* podem ser delegadas total ou parcialmente

- Os utilizadores **escolhem** as permissões dos seus ficheiros*

  …podem dar acesso a outros utilizadores ou grupos (ou a toda a gente)
  …só não podem delegar a escolha para terceiros (i.e. mudar o *owner*)

*DAC - **Discretionary** Access Control

Modelo de Acessos do SELinux

- Política de acessos *system-wide*

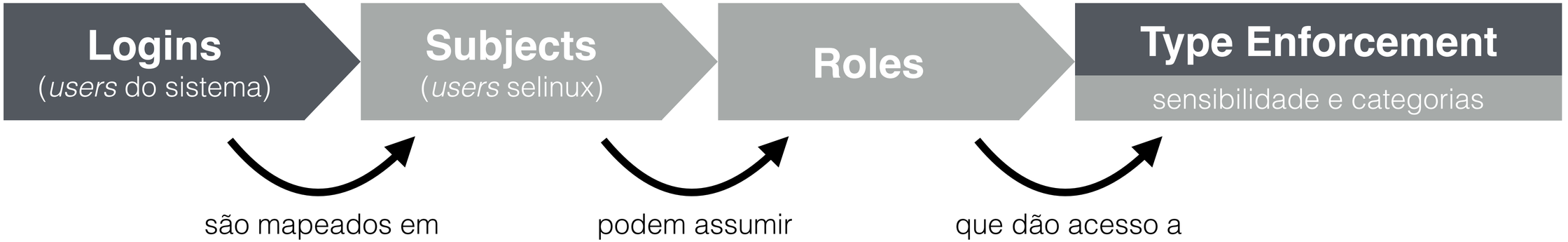
  …definida pelo administrador e **não-alterável** pelos utilizadores*
  …onde se define **explicitamente** o que é autorizado (*deny by default*)

- O controlo baseia-se em *type enforcement*

  …processos, ficheiros, *sockets*, etc. têm um **tipo** associado (contexto)
  …a política define **interacções** e **transições** autorizadas entre tipos

*MAC - **Mandatory** Access Control

Modelo de Acessos do SELinux



- Os *subjects* e os *roles* **não têm** permissões por si próprios*

  …são só caminhos para chegar a conjuntos de regras de *type enforcement*

- Os objectos podem ter (opcionalmente) níveis de sensibilidade** e categorias***

  …os níveis de sensibilidade são *read down* e *write up* (modelo Bell — La Padula)
  …as categorias seguem regras de dominância (intersecção de conjuntos)

MLS - **Multi-Level Access Control

*RBAC - **Role-Based** Access Control ***MCS - **Multi-Category** Access Control

Isto era a **teoria**, agora podem esquecer isto tudo…

Política de Referência

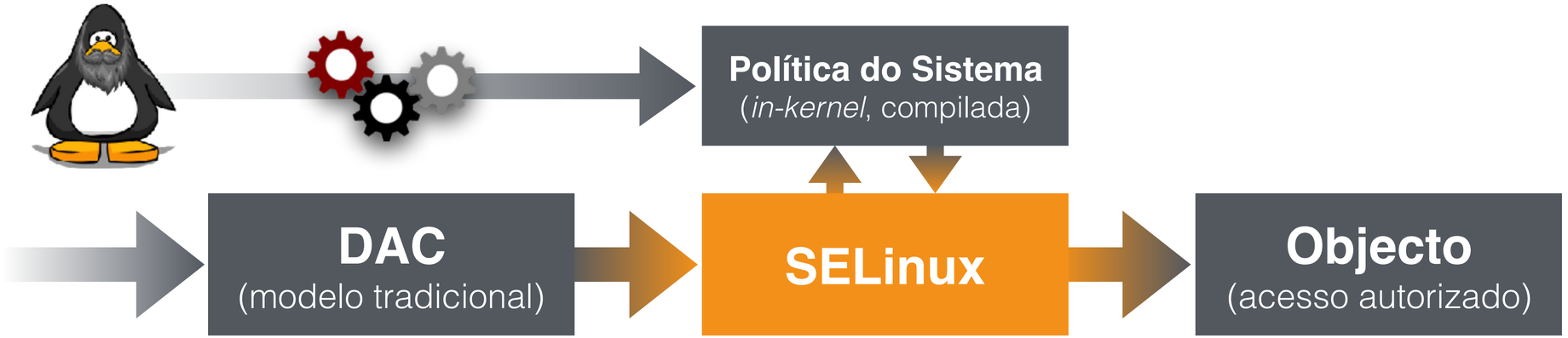
- Orientada a serviços específicos (*targeted policy*)

  …**todos** os serviços **sem uma política definida** correm *unconfined*
  …**todos** os utilizadores são *unconfined* **por omissão**

- Adicionalmente…

  …um serviço pode correr em modo *permissive* num sistema *enforcing*
  …os domínios *unconfined* podem ser desactivados (*strict policy*)

Controlo de Acessos



- O SELinux só é envolvido quando o DAC **já autorizou** o acesso

  …portanto só pode dar permissões que o utilizador **já teria** pelo modelo tradicional

- As decisões são guardadas na *Access Vector Cache* (**AVC**)

  …por isso é que as mensagens no `audit.log` são AVCs

Contexto de Segurança

- Na política *targeted,* o *subject* e o *role* são **pouco relevantes**

  …e podem ser largamente **ignorados**, mesmo ao escrever regras para serviços novos
  …a sensibilidade (sempre “s0”) e as categorias (ausentes) também podem ser **ignoradas**

- Em entidades passivas (ex: ficheiros), o *role* é **sempre** `object_r`

  …em que **`object_r`** é um *placeholder* para entidades onde os *roles* não fazem sentido

— DEMO —

Referências

- Por onde começar:

  red.ht/1VmGJ9j — SELinux User's and Administrator's Guide (RHEL 7)
  wiki.centos.org/HowTos/SELinux
  wiki.gentoo.org/wiki/SELinux

- Escrever novos *policy modules*:

  danwalsh.livejournal.com/35127.html
  oss.tresys.com/docs/refpolicy/api/system.html
  selinuxproject.org/page/NB_RefPolicy

Obrigado!

Perguntas?

Carlos Rodrigues
cer@brpx.com

C0NFRAR1A

Podem encontrar estes slides em speakerdeck.com/carlosefr