
archlinux
A simple, lightweight gnu/linux distribution.

Projeto de melhorias de segurança para Arch Linux usando grsecurity e hardening.

Quem sou e o que faço?

Leandro I S Carvalho

Pós Graduado pelo CESMAC.
Fundador do projeto Arch Linux Brasil.

Usuário de Linux desde 2005 e desde 2006 usuário do Arch Linux.

Analista de Suporte e Segurança na Infox (Aracaju/SE).
Atual coordenador do projeto Arch Linux Brasil.

archlinux
A simple, lightweight gnu/linux distribution.

O que veremos ???

* Diferenças entre as versões do pacman no quesito segurança;

* Compilação dos pacotes no Arch em ambiente chroot;

* grsecurity;

* gradm;

* hardening (pam e filesystem).

PACMAN

Versões 2.x e 3.x:

* Apenas MD5

Apartir da versão 4.x:

Uso do gpgme:

* Assinatura de pacotes e base de dados (infomações sobre os pacotes);
* Verificação de assinaturas dos pacotes e dos sources;
* Utilização de MD5.

Inconvenientes?

* Necessidade de mudança na forma de submeter os pacotes (devtools);
* Usuários tem que verificar/confiar nas chaves geradas (--edit-key);
* Utilização de chave mestre, para assinas as chaves dos desenvolvedores e a base de dados.

Compilação em chroot

Preparação do ambiente:

```
$ pwd
/home/leandro/Pacotes
$ mkdir chrootpkg
$ sudo mkarchroot chrootpkg/root base base-devel sudo
```

Compilando um pacote:

```
$ pwd
/home/leandro/Pacotes/pkgbuilds/bash
$ ls
bash.install dot.bash_logout dot.bashrc PKGBUILD system.bash_logout
      system.bashrc
$ sudo makechrootpkg -c -r /home/leandro/Pacotes/chrootpkg -- -L
$ ls
bash-4.2.010-1-x86_64-build.log  bash-4.2.010-1-x86_64-package.log
dot.bash_logout dot.bashrc  system.bash_logout  bash-4.2.010-1-x86_64-
check.log bash.install dot.bash_profile PKGBUILD system.bashrc
```

Compilação em chroot

Observações:

* /etc/makepkg.conf do sistema indica onde será armazenado os pacotes;

* As flags de compilação devem ser referenciadas no makepkg.conf do chroot;

* Utilizando usuário normal via sudo, um novo chroot é criado;

* Necessário passar o parâmetro “-- -L”, para “logar” todo o procedimento de compilação;

* O ambiente é criado na máquina local do desenvolvedor, para depois enviar para o servidor ftp (mirror principal).

grsecurity

O que é e suas características?

* Conjunto de patches aplicados diretamente nos sources do kernel;

* Visa melhorar a segurança do sistema a nível de kernel;

* Suporte da versão 2.6.32.46 até 3.0.8 (mais atual);

* Conceito de segurança pró-ativa;

* Proteção contra 0-day (????);

* Confinamento e restrições do sistema através do RBAC e chroot;

* Uso do PaX (PAge eXecute);

* Necessário uso de flags de hardening para compilação de toda a distribuição.

grsecurity

Níveis de segurança?

* Baixo (Low);

* Médio (Medium);

* Alto (High);

* Custom (Você que customiza).

grsecurity

chroot

* Visa previnir:

* Ataques que escalam privilégios;
* Variações de vulnerabilidades;

* Modificações implementadas no chroot:

* Memória compartilha não é anexada;
* Sem kill;
* Sem ptrace (independente de arquitetura);
* Sem remount ou mount;
* Sem visualização de processo (mesmo se o /proc estiver montado) for a do ambiente chroot;
* Sem (f)chmod +s;
* Sem mknod;
* Sem escrita de sysctl

grsecurity

RBAC (Role Based Access Control)

* Sem acesso de root para pápeis especiais;

* Pápeis especiais sem necessidade de autenticação;

* Interpretação de herança pelo Kernel;

* Resolução em real-time de expressões regulares;

* Capacidade de negar ptraces a processos específicos;

* device especial (/dev/grsec) no kernel para autenticação e registro de logs (apredizagem);

* Pathnames completos para processos pais e processos filhos

grsecurity

RBAC (Role Based Access Control)

* Status das funções do RBAC para gradm

* Suporte a read, write, append, execute, view, e read-only ptrace para permissões do object;

* Suporte a hide, protect e substituição de subject flags;

* Suporte a PaX flags;

* Proteção para memória compartilhada;

* Proteção para /proc/pid filedescriptor/memory;

* Sem dependência de filesystem ou arquitetura;

* Sem alocação de memória em runtime.

grsecurity

PaX

* Ferramentas: chpax e paxctl;
* flags nas informações na memória;
* acesso dos programas a memória marcada como non-writable;
* Prevenção para execução de memória paginada a partir de sobreescrita e com inserção de código de máquina;
* Previne exploração de muitos vulnerabilidades de segurança como:
    * buffer overflow;
    * stack overflow;
    * heap overflow;
    * stack smashing.

* PIE (Position Independent Executables) execução independente da posição da memória:
    * Flags de compilação -fPIE
    * Difere do código tradicional na medida em que os acessos às funções são feitos, através de uma tabela de acesso indireto.

grsecurity

PaX

* SSP (Stack Smashing Protector):
  * Utilizado pelo PaX para detecção de buffer overflow, GCC insere código de inicialização nas funções que criam buffer na memória;
  * *Bit Canary.*

* ASLR (Address Space Layout Randomization)

Mecanismo de segurança que introduz aleatoriedade no processo de alocação dos segmentos de um processo em memória. Esse processo é realizado toda vez que um aplicativo é executado e carregado em memória pelo sistema operacional.

gradm

* Ferramanta que permite administrar e manter as políticas no sistema;

* É um pacote extra que faz parte do conjunto provido pelo grsecurity.

* Assim como o grsecurity, sua instalação é manual.

* As regras por padrão não são ativas.

* O sysadmin tem que determinar quando o sistema deve usar o RBAC.

* Permite:
    * Habilitar ou desabilitar o sistema RBAC;
    * Recarregar as regras do RBAC;
    * Mudar suas regras;
    * Informar uma senha para o modo administrador.)

gradm

Exemplos de comandos:

gradm -E (habilita o sistema)

gradm -D (desabilita o sistema)

gradm -a admin (habilita o papel de administrador)

gradm -u admin (desabilita o papel de administrador)

gradm –F –L /etc/grsec/learning.log -O /etc/grsec/policy
(iniciando uma aprendizagem completa do sistema)

Hardening

Melhorando as senhas no sistema.

PAM

* Módulos: cracklib e unix
* /etc/pam.d/passwd

> difok = Diferente da senha anterior

> retry = Quantas tentativas

> minlen = Define o mínimo de caracteres

> dcredit = Dígitos

> ocredit = Caracteres

> use_authtok = Força a informação da senha anterior

> nullok = Permite acesso se a senha estiver em branco

Hardening

Melhorando as senhas no sistema.

PAM

* Módulos: tally
* /etc/pam.d/login

> deny	=	Nega o acesso depois de N tentativas
> unlock_time	=	Permite o acesso depois de N segundos
> onerr	=	Se algo estranho acontecer, ele retorna a saída definida
> file	=	Define o arquivo de log, geralmente /var/log/faillog

Hardening

Filesystem

* /etc/fstab

> noexec
> nosuid
> nodev

Referências:

grsecurity.net
pax.grsecurity.net
pax.grsecurity.net/docs/aslr.txt
gentoo.org/proj/en/hardened/index.xml
linuxfromscratch.org/hlfs/index.html
paginas.fe.up.pt/~ee07061/styx/index.php
Manual do GCC
Manual do PAM
ibm.com/developerworks/br/library/l-pam/
debian.org/doc/manuals/securing-debian-howto/
puschitz.com/SecuringLinux.shtml
csrc.nist.gov/groups/SNS/rbac/

Boas práticas para segurança da informação:

* BS 7799
* ISO 27000

Obrigado!

Contatos??

IRC:

/server irc.freenode.net
/j #archlinux-br
skate_forever

Twitter:

@skate_forever

E-mail:

leandro @ archlinux-br.org
carvalho.inacio @ gmail.com

Site:

archlinux.org
archlinux-br.org
blog.leandroinacio.eti.br