

# Aula2

Administração de redes Linux  
Laboratório Daniel Bauman Henry  
LabMAP



MAP - IME - USP

# Agenda

1. Uso do SSH
2. Criptografe tudo!
3. Compilando Programas
  - Configure
  - Makefile
4. Gerenciamento de pacotes
5. Slackbuilds - Como criar um pacote
6. Slackbuilds.org

# SSH

- Rede TCP/IP - Pacotes
- História: telnet & sniffers
  - tcpdump
  - wireshark (ethereal)
- ssh - Secure Shell
  - ssh user@host
- scp - Secure Copy
  - scp file user@host:directory/
- Tunelando conexões
  - ssh user@host -ND 8888 sleep 99999



# Criptografe tudo! Caso contrário...

- Lendo headers http:
  - `tcpdump -nnvvXSs 0 -l -A -i eth0 dst host -w sniff`
  - Exemplos:
    - Cookies (roubo de sessão)
    - Autenticações simples (base64)
    - Formulários de login/senha (roubo de senha)



# Compilando programas - Autotools

- Resumo:

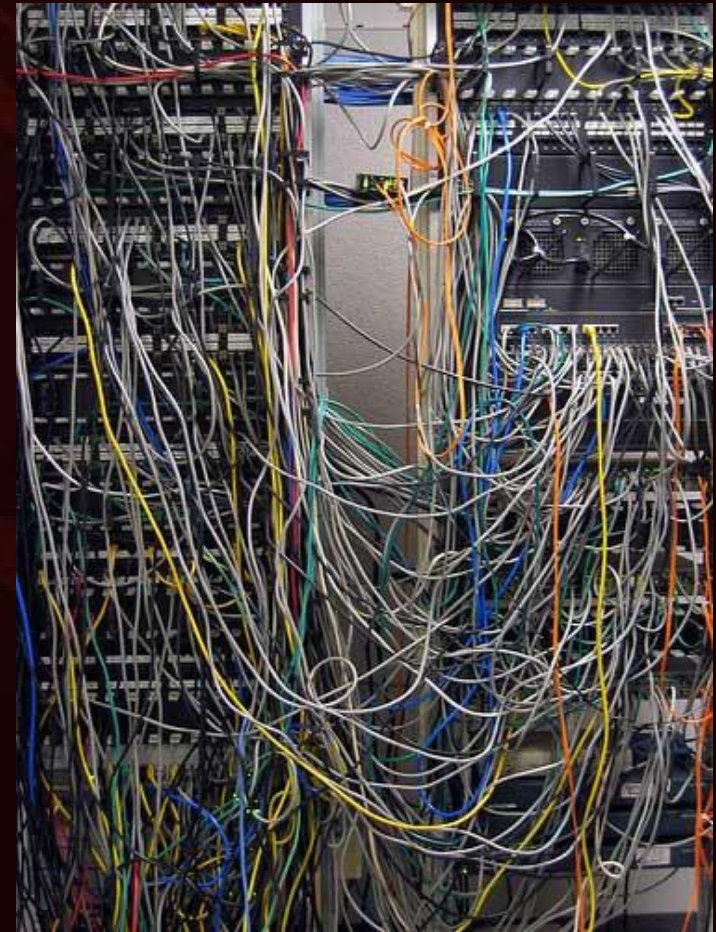
- configure: descobre informações da máquina e cria o makefile adequadamente, além de verificar as dependências do programa
- make: compila tudo usando o makefile gerado pelo configure
- make install: copia os binarios tudo mais para o sistema, geralmente é necessário ser root





# Compilando programas - Autotools

- Problemas:
  - você não tem controle sobre o que e onde foi instalado
  - fica difícil replicar a instalação para outras máquinas
  - para desinstalar ou atualizar depende da (má) qualidade do Makefile



# Compilando programas - Pacotes

- O sistema de pacotes mantem um catálogo de tudo que foi instalado
- Exemplos: dpkg, apt, rpm - são anabolizados!
- O Slackware usa o pkgtool - KISS: é apenas um tar!
- Não checa dependencias
- pkgtools:
  - installpkg
  - removepkg
  - upgradepkg
  - makepkg
  - explodepkg
  - pkgtool



# Onde encontrar pacotes?

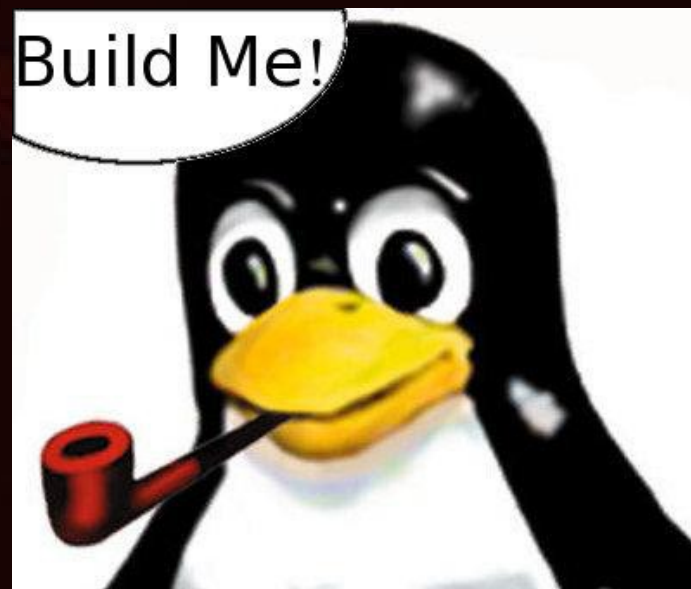
- Fazer seus próprios pacotes
  - `make install DESTDIR=/tmp/meupacote`
  - `makepkg`
  - Problemas:
    - retrabalho na atualização
    - padronização de diretórios
- Baixá-los prontos
  - [linuxpackages.net](http://linuxpackages.net)
  - [www.slackware.eu](http://www.slackware.eu)
  - Problemas:
    - Atende as necessidades?
    - Você confia nos binários de outra pessoa?





# SlackBuilds

- Como gerar os pacotes de uma maneira organizada?
  - Atualizar
  - Distribuir
  - Documentar compilação
- Mantendo a simplicidade: um SlackBuild é apenas um script em bash que compila o programa e cria um pacote para ser instalado com pkgtools
- Também são usados pelo Volkerding para compilar os pacotes oficiais da distribuição (ficam no diretório source)



# SlackBuilds

- Resumidamente, um SlackBuild:
  - Verifica a arquitetura da máquina, para setar os parâmetros corretos do compilador
  - Roda o configure
  - Roda o make
  - Roda o make install em um diretório separado
  - Gera o pacote com o makepkg
- Veremos um exemplo mais adiante

# SlackBuilds.org

- Um grande repositório de slackbuilds, mantidos pela comunidade, para suprir a necessidade de programas extras, que não são mantidos na distribuição oficial
- Apesar de mantidos pela comunidade, existe uma equipe que avalia e testa todos os slackbuilds, portanto podemos confiar mais.
- As submissões devem seguir um padrão criterioso definido pela equipe do projeto: <http://slackbuilds.org/guidelines/>

**SLACKBUILDS**  
dot org

# SlackBuilds.org - exemplo

<http://slackbuilds.org/repository/13.37/graphics/dia/>

<http://slackbuilds.org/repository/13.37/system/unrar/>

# Exceções

- Nem sempre é possível criar um SlackBuild ou pacote:
  - licença
  - procedimento de instalação
  - Ex: compilador intel, IRPF
- Ou ainda o programa é muito simples e criar o pacote seria um overkill chato
  - Ex: Vuze
- Saída: instalar no /opt



whatever!



# Tarefas

- Habilitar o telnet na sua máquina
- Compilar e instalar o md5deep sem pacote
- Descobrir o que foi instalado
- Remover o md5deep
- Repetir o processo usando um Slackbuild
- Instalar o unrar
- Instalar o Vuze no /opt
- Habilitar o telnet na sua máquina
- Instalar o wireshark
- Sniffar uma conexão telnet
- Sniffar uma conexão ssh

