Grid systems installation and configuration

Louis Poncet SA3/GD

www.eu-egee.org

Summary

- Introduction
- Security begins at the installation
- Management of an up-to-date system
- Disabling unwanted services
- Configuring system level firewall
- World-writable files and directories
- Conclusion

Introduction

- **Why security per centre is fundamental :**
  - A malicious user getting grid access can destroy everything
  - Any centre can be an access to all grid resources
- **We need heterogeneous monitoring of the activities; it enables us to find serious middleware bugs and security problems**
- **We need to work together to improve grid security, such a large deployment increases the security risks**
- **One centre insecure = grid insecure**
- **Resources have to be allocated for the security activities at all sites part of EGEE/LCG**

Security begins at the installation

- **Kickstart installation; default network install for SLX**
  - 1 kickstart profile per type of nodes (software and hardware)
  - Minimal set of packages depending of the node type
    - Minimal for core services hosts (apt will install what is really require)
    - For the WN some extra packages categories (ex: development)
    - Partitioning on the HD(s)
  - Network settings configuration of network devices
  - A “keep certificate and server ssh keys” process in the kickstart can be implemented using a RAM disk
  - Removing all unused services and tools
  - Setting of SSH and extra repositories for internal tools
  - Force a real and full update at the first reboot
  - Periodic reinstallation is a good idea
- **Localization (example : CERN case)**
  - Installation of tools for security, monitoring, CERN tools and fabric management

Extract of our kickstarts

```
<...>
#Root password
rootpw --iscrypted *NP*
authconfig --enableshadow --enablemd5
<...>
#Firewall configuration
firewall --enabled --ssh
<...>
#Package install information
%packages --resolvedeps
@ base
#@ Administration Tools
@ System Tools
#@ development-tools
#@ text-internet
#emacs
#XFree86-xauth
ntp
<...>
%pre
mkdir -p /tmp/preserve
mkdir -p /tmp/fs
for I in 1 2 3
do
   L=`e2label /dev/sda$I`
   if [ ""$L = "/" ]
   then
      mount /dev/sda$I /tmp/fs
   fi
done

if [ -f /tmp/fs/etc/grid-security/hostcert.pem ]; then
    cp /tmp/fs/etc/grid-security/host*.pem /tmp/preserve
fi


if [ -f /tmp/fs/etc/ssh/ssh_host_key ]; then
    cp /tmp/fs/etc/ssh/ssh_host_* /tmp/preserve
fi

cd /
umount /tmp/fs
<...>
```

```
<...>
# Removing unwanted packages from Base install
chroot /mnt/sysimage /usr/bin/yum -y remove acpid apmd
aspell aspell-en bind-libs bind-utils bluez-bluefw bluez-
hcidump bluez-libs bluez-utils cr
yptsetup cups cups-libs curl cyrus-sasl-plain desktop-
file-utils fontconfig freetype ftp htmlview indexhtml
ipsec-tools ipw2100-firmware ipw2200-
firmware irda-utils isdn4k-utils jpackage-utils jwhois
lftp lha libgcrypt libgpg-error libidn libjpeg libpcap
libpng libtiff libwvstreams libxslt
 lrzsz minicom mtr mt-st NetworkManager nfs-utils nfs-
utils-lib nss_ldap numactl parted pcmcia-cs pdksh perl-
AppConfig-caf pinfo portmap ppp redh
at-lsb redhat-menus rp-pppoe rsync stunnel system-config-
network-tui system-config-securitylevel-tui tcpdump
unix2dos wireless-tools words wvdial
 xinetd xmlsec1 xmlsec1-openssl xorg-x11-libs xorg-x11-
Mesa-libGL ypbind yp-tools perl-CAF
<...>
#Security repository
echo "[GD Security]" >> /mnt/sysimage/etc/yum.repos.d/gd-
security.repo
echo "name=GD Security" >> /mnt/sysimage/etc/yum.repos.d/
gd-security.repo
echo "baseurl=http://grid-deployment.web.cern.ch/grid-
deployment/gis/apt/security/sl3/en/i386" >> /mnt/
sysimage/etc/yum.repos.d/gd-security.repo
echo "enabled=1" >> /mnt/sysimage/etc/yum.repos.d/gd-
security.repo
echo "" >> /mnt/sysimage/etc/yum.repos.d/gd-security.repo
<...>
# SSH
echo "PermitRootLogin without-password" >> /mnt/sysimage/
etc/ssh/sshd_config
echo "Protocol 2" >> /mnt/sysimage/etc/ssh/sshd_config
```

Keeping systems up-to-date

- **Usage of a advance package manager**
- **The principle is to maintain a repository accessible by all hosts of the computing centre**
  - Depending of the bandwidth it is possible to use a Web proxy
    - All downloads are made one time
  - The repositories commonly contain few categories
    - An OS, updates and externals that contain security fixes for installed applications
    - Signatures should checked for the packages (ex : yum & apt configuration)
    - Running periodically a package verifier can be really useful
- **The problem is the kernel & kernel modules updates which need a reboot**

Tools for service and package

- **Installation / Maintenance**
  - APT
  - YUM
  - Squid (web proxy)
  - Fabric management tools
  - chkconfig
  - rpmverify
- **Project providing external tools repositories :**
  - jpackage - http://www.jpackage.org/
  - Dag - http://dag.wieers.com
  - Fresh Rpms - http://freshrpms.net
- **For monitoring the status of your settings :**
  - Pakiti - http://pakiti.sourceforge.net/

Example : upgrade of middleware

```
[root@XXX root]# netstat -tap | grep portmap
tcp        0      0 *:sunrpc                    *:*                         LISTEN      2104/portmap
[root@lxb6121 root]# chkconfig --list | grep portmap
portmap         0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@XXX root]# rpmverify -qv chkconfig
[root@lxb6121 root]#  chkconfig portmap  off
[root@lxb6121 root]#  chkconfig --list | grep portmap
portmap         0:off   1:off   2:off   3:off   4:off   5:off   6:off

[root@lxb6125 yum.repos.d]# ls -l
total 152
-rw-r--r--  1 root root  622 May 22 18:41 atrpms.repo
-rw-r--r--  1 root root  413 May 22 18:41 cern-extra.repo
-rw-r--r--  1 root root  436 May 22 18:41 cern-extra-srpms.repo
-rw-r--r--  1 root root  642 May 22 18:41 cern-only.repo
-rw-r--r--  1 root root  664 May 22 18:41 cern-only-srpms.repo
-rw-r--r--  1 root root  379 May 22 18:41 cern.repo
-rw-r--r--  1 root root  401 May 22 18:41 cern-srpms.repo
-rw-r--r--  1 root root  511 May 22 18:41 cern-test.repo
-rw-r--r--  1 root root  536 May 22 18:41 cern-test-srpms.repo
-rw-r--r--  1 root root  485 May 22 18:41 cern-update.repo
-rw-r--r--  1 root root  507 May 22 18:41 cern-update-srpms.repo
-rw-r--r--  1 root root  363 Sep 13 10:40 dag.repo
-rw-r--r--  1 root root  148 Sep 13 10:16 gd-lemon.repo
-rw-r--r--  1 root root  215 Sep 13 10:16 gd-security.repo
-rw-r--r--  1 root root  312 Sep 25 14:14 glite.repo
-rw-r--r--  1 root root 1039 Sep 13 10:41 jpackage.repo
-rw-r--r--  1 root root  191 Sep 13 10:43 lemon.repo
[root@lxb6125 yum.repos.d]# cat cern.repo
#
[main]
[slc4-base]
name=Scientific Linux CERN 4 (SLC4) base system packages
baseurl=http://linuxsoft.cern.ch/cern/slc4X/$basearch/yum/os/
gpgkey=http://linuxsoft.cern.ch/cern/slc4X/$basearch/docs/RPM-GPG-KEY-cern
       http://linuxsoft.cern.ch/cern/slc4X/$basearch/docs/RPM-GPG-KEY-jpolok
gpgcheck=1
enabled=1
protect=1
```

Package signature verification

YUM update (exemple)

```
[root@lxb6125 yum.repos.d]# yum update
<.....>
=============================================================================
 Package                  Arch       Version           Repository        Size
=============================================================================
Installing:
 kernel                   x86_64     2.6.9-55.0.6.EL.cern  slc4-update        12 M
 kernel-module-openafs-2.6.9-55.0.6.EL.cern  x86_64     1.4.4-2.cern     slc4-update       3.4 M
 kernel-module-openafs-2.6.9-55.0.6.EL.cern  x86_64     1.4.4-4.cern     slc4-update       3.4 M
 kernel-module-openafs-2.6.9-55.0.6.EL.cern  x86_64     1.4.1-26.cern    slc4-update       3.4 M
 kernel-module-openafs-2.6.9-55.0.6.EL.cernsmp  x86_64     1.4.4-4.cern     slc4-update       3.4 M
 kernel-module-openafs-2.6.9-55.0.6.EL.cernsmp  x86_64     1.4.1-26.cern    slc4-update       3.4 M
 kernel-module-openafs-2.6.9-55.0.6.EL.cernsmp  x86_64     1.4.4-2.cern     slc4-update       3.4 M
 kernel-smp               x86_64     2.6.9-55.0.6.EL.cern  slc4-update        12 M
Updating:
 GFAL-client              x86_64     1.10.1-1          glite-wn-64       2.4 M
 cert-glite-WN            noarch     3.1.0-3           glite-wn-64       4.0 k
 cyrus-sasl               i386       2.1.19-14         slc4-update       1.2 M
 cyrus-sasl               x86_64     2.1.19-14         slc4-update       1.2 M
 cyrus-sasl-md5           x86_64     2.1.19-14         slc4-update        64 k
 kernel-module-openafs-2.6.9-55.0.2.EL.cern  x86_64     1.4.4-4.cern     slc4-update       3.4 M
 kernel-module-openafs-2.6.9-55.0.2.EL.cernsmp  x86_64     1.4.4-4.cern     slc4-update       3.4 M
 kernel-module-openafs-2.6.9-55.EL.cern  x86_64     1.4.4-4.cern     slc4-update       3.4 M
 lcg_util                 x86_64     1.6.1-2           glite-wn-64       192 k
 openafs                  x86_64     1.4.4-4.cern      slc4-update       5.7 M
 openafs-client           x86_64     1.4.4-4.cern      slc4-update       1.0 M
 openafs-compat           x86_64     1.4.4-4.cern      slc4-update       7.5 k
 openafs-kpasswd          x86_64     1.4.4-4.cern      slc4-update       127 k
 openafs-krb5             x86_64     1.4.4-4.cern      slc4-update       130 k
 openafs-server           x86_64     1.4.4-4.cern      slc4-update       2.4 M
Removing:
 kernel                   x86_64     2.6.9-55.EL.cern   installed           37 M
 kernel-smp               x86_64     2.6.9-55.EL.cern   installed           35 M

Transaction Summary
=============================================================================
Install      8 Package(s)
Update      15 Package(s)
Remove       2 Package(s)
Total download size: 69 M
Is this ok [y/N]:
```

I can't upgrade the kernel and afs now. But I have to upgrade all others

YUM (update)

```
[root@lxb6125 yum.repos.d]# yum update --disablerepo=slc4-update
<...>
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Package lcg_util.x86_64 0:1.6.1-2 set to be updated
---> Package GFAL-client.x86_64 0:1.10.1-1 set to be updated
---> Package cert-glite-WN.noarch 0:3.1.0-3 set to be updated
--> Running transaction check

Dependencies Resolved

=============================================================================
 Package                Arch       Version          Repository        Size
=============================================================================
Updating:
 GFAL-client            x86_64     1.10.1-1         glite-wn-64       2.4 M
 cert-glite-WN          noarch     3.1.0-3          glite-wn-64       4.0 k
 lcg_util               x86_64     1.6.1-2          glite-wn-64       192 k

Transaction Summary
=============================================================================
Install      0 Package(s)
Update       3 Package(s)
Remove       0 Package(s)
Total download size: 2.6 M
Is this ok [y/N]: y
Downloading Packages:
(1/3): lcg_util-1.6.1-2.x 100% |=========================| 192 kB    00:00
(2/3): GFAL-client-1.10.1 100% |=========================| 2.4 MB    00:00
(3/3): cert-glite-WN-3.1. 100% |=========================| 4.0 kB    00:00
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating  : GFAL-client                  ######################### [1/6]
  Updating  : lcg_util                     ######################### [2/6]
  Updating  : cert-glite-WN                ######################### [3/6]
  Cleanup   : lcg_util                     ######################### [4/6]
  Cleanup   : GFAL-client                  ######################### [5/6]

Updated: GFAL-client.x86_64 0:1.10.1-1 cert-glite-WN.noarch 0:3.1.0-3 lcg_util.x86_64 0:1.6.1-2
Complete!
```

Disabling all unneeded services

- **Install all vs Install nothing**
  - **Install all :**
    - The base installation contains already a pretty big list of tools and expose network services that you don't need
      - *You can remove the package*
      - *Or prevents the service the start*
  - **Install nothing** :
  - A basic installation that contain a really minimum set of packages.
    - If you install an application :
      - *You need it and you want to use it*
      - *So you configure and start it*
- **Let's take a typical case :**
  - MySQL packaging: during the installation a little message arrive during the installation of the package explaining you how to set a password, then install and START the network service, ("skip-networking" option can be use to avoid MySQL to listen on the network).

Removing unused/unneeded tools

- **Everything that does not need to be started for the node usage is stopped and the package removed if possible**
  - We remove all unneeded network services including clients
  - All hacking tools sniffer, scanner and unneeded setuid binaries (man find) / kernel modules
  - Some packages cannot be removed due to dependency issues

- **To verify that it is properly done network scan has to be done**
  - Is there network services listening that should not (ex: chkconfig)
  - Is there a vulnerability on a known services (ex: Nessus)

System level firewall

- **Why a local firewall when my site has a network one ?**
  - Prevent attacks from the LAN
- **One firewall profile per type of services**
- **Block what you want as you want**
  - You can block port access for all host or just a set of host
  - That decrease the load of a service (misconfiguration)

```
[root@XXXX root]# iptables -L | grep policy
Chain INPUT (policy DROP)
Chain FORWARD (policy DROP)
Chain OUTPUT (policy ACCEPT)
```

World-writable files / directories

- **During the configuration the creation of files / directories can be in a writable mode**
  - On AFS for example if a directories is writable by everybody the sub (files or directories) will be too
  - A file in mode 777 in the path of users can be really dangerous because anybody can create a malicious executable (ex: find $PATH -perm)
- **During jobs execution some files or folders can be create in 777 mode**
- **The root user of a machine can per mistake create a files in 777 mode**

Debugging

- **Debugging pretty often begins with :**
  - Deactivation of the firewall
  - Changing files permissions
  - Running as root (all permissions granted)
  - And continuing to do really dirty stuff till it work

- **Here the problem start :**
  - We know how it work without any security stuff but it is mandatory to be as secure as possible

Conclusion

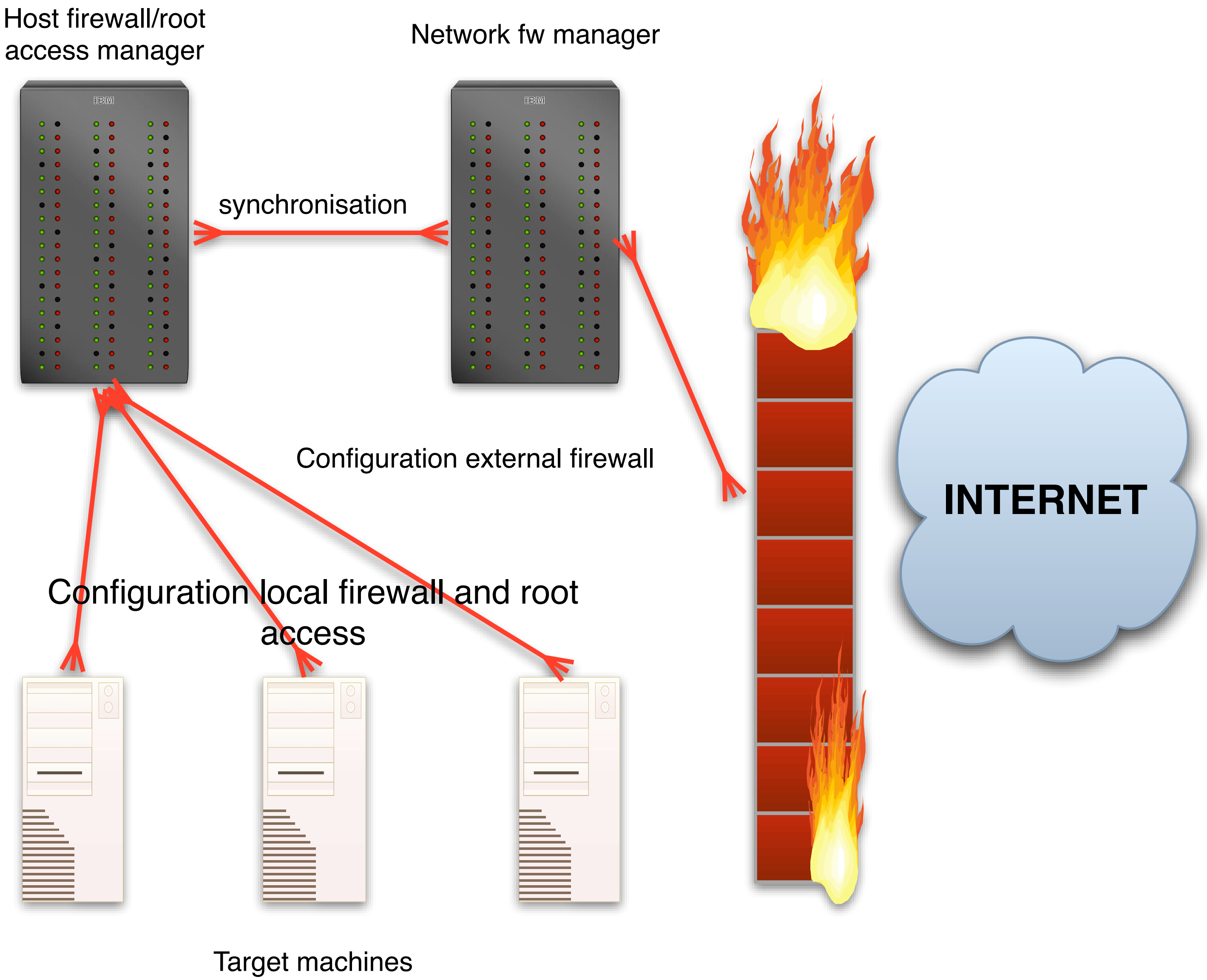
- Every site is unique you have to set security procedures at yours
- All the points that were in the presentation are really simple and must be applied
- We have to work together to improve the quality of our site security
- All comments are welcome
- Reading O’reilly book about security and system administration is cheaper than repairing a hacker attack

Questions

CERN GD example

- **In GD we have our repository with our packages**
  - Monitoring
  - Check security updates status
  - Set user root access
  - Set firewall depending of the type of node
- **Let's take the case of a GD node**
  - Lemon monitor the activities of CPU, Memory management and disk access
  - Every hour an update check for security update
  - Every hour there is also a check of the list of root to set
  - Download and application of the firewall
  - Everything is manage trough a centrally manage system
  - For the certification machine there is also a monitoring of the patch in certification
- **The system stays up-to-date and check that every hour**

Our architecture



Host firewall/root access manager
Network fw manager
synchronisation
Configuration external firewall
INTERNET
Configuration local firewall and root access
Target machines