



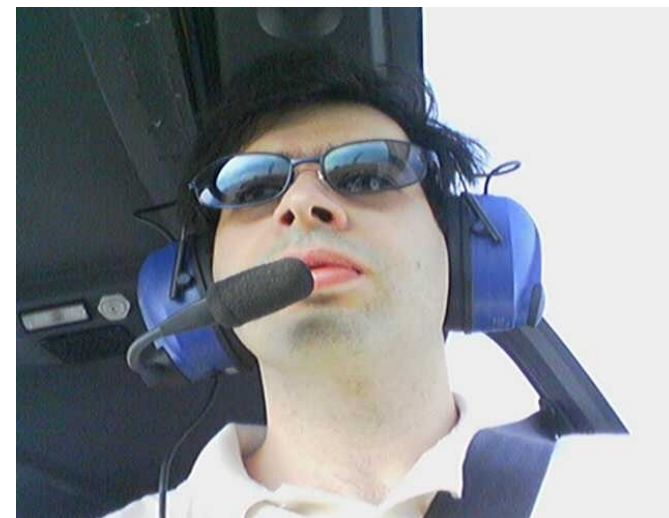
Protecting confidential files using SE-Linux

Giuseppe “Gippa” Paternò
Visiting Researcher
Trinity College Dublin



Who am I

- Visiting Researcher at Trinity College Dublin (Ireland)
- Solution Architect and EMEA Security Expert in Red Hat
- Previously Security Solution Architect in Sun and also in IBM
- Red Hat Certified Security Specialist (RHCSS), Red Hat Certified Architect (RHCA) and Cisco Certified Network Professional (CCNP)
- Part of the world-wide security community (especially SEMEA)
- Published books and whitepapers
- Forensic analysis for local govts
- More on:
 - <http://www.scss.tcd.ie/Giuseppe.Paterno/>
 - <http://www.gpaterno.com/>
 - <http://www.linkedin.com/in/gpaterno>





Disclaimer

I do not speak on behalf of my employer, nor I am authorized to represent it publicly.

All and any opinion and results expressed in this presentation are solely mine and do not represent my employer point-of-view.

All the tests and any project contribution are done as a TCD researcher out of business hours.



The challenge

- The challenge was to protect highly confidential PDF files
 - A J2EE web-based application with smartcard authentication that must fulfill given PDF files to the allowed users.
 - System administrators should manage the machines but they can't access in any way the PDF files and any attempt must be logged.
- The customer: a government agency



Requirements (1/2)

- Unique identification of the users via LDAP
- Any audit log must be sent to a central logging system
- System administrators must not become root, but execute some given programs via “sudo”
- System administrators should have different privilege levels, from operator to full sysadmin powers
- System administrators cannot do a “su -” to access root user, although they know the password, unless authorized to execute “su”.

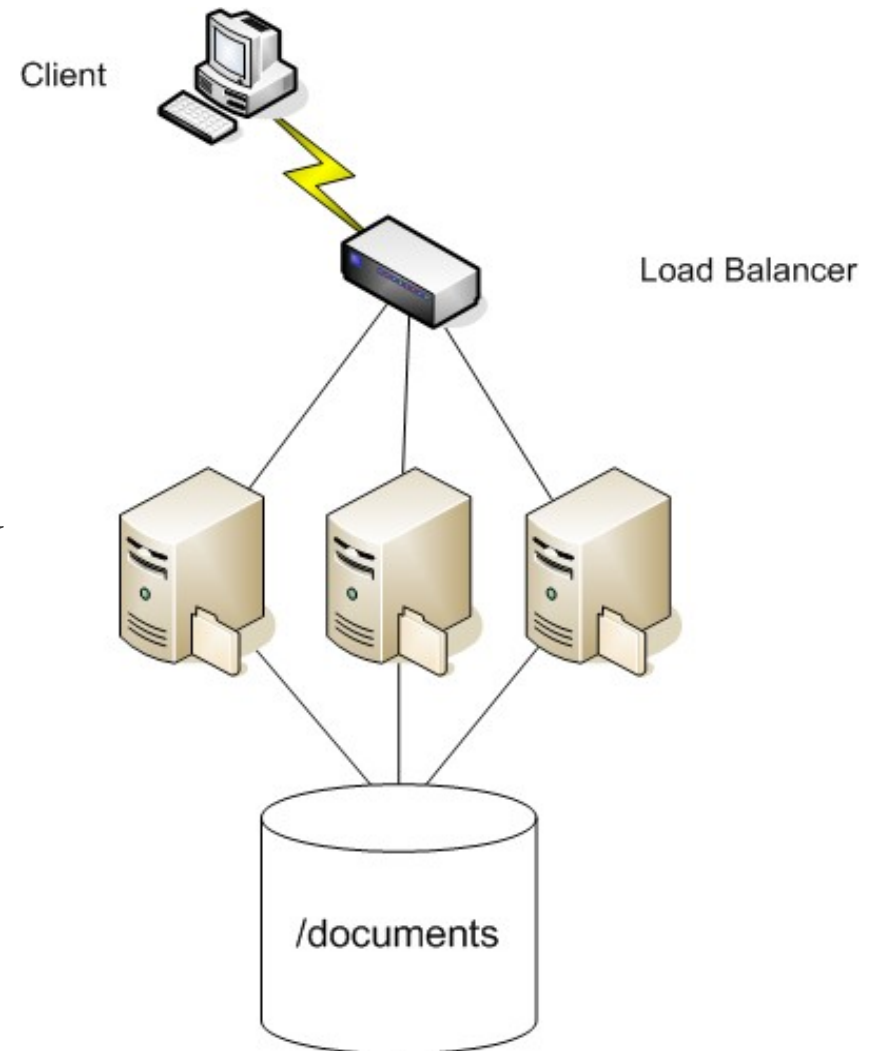


Requirements (2/2)

- No user, with the exclusion of *root* and the application server user *appserv*, are allowed to access “/documents/” directory and related documents/subdirectories
- The *root* user have the right to access the protected directory, but any read attempt must be under audit.
- The application server user must not be under audit for performance issues.
- Both the application server and some batches will run through the user *appserv* that has right to access documents.

The architecture

- 3 nodes cluster with balancing
- Shared filesystem across nodes through GFS
 - PDF files are located in the clustered fs.
- Jboss Application Server
 - Apache frontend to allow smart-card authentication
- Linux as the OS
 - Red Hat Enterprise Linux
 - Red Hat Cluster Suite and GFS
 - SE-Linux for mandatory access





Access levels

Privilege level	Description
operator	Can connect to the machine and access the logs
appmanager	Operator privileges + restart services
admin	Operator privileges + can do “su –”. No access to PDF files
Application user (<i>appserv</i>)	The application server will run under this user and has right to access the PDF files
root	Unix administrator, only console access allowed



Introduction to SE-Linux

- SE-Linux is a security module to implement mandatory access control (MAC)
- Developed by National Security Agency (USA) and upstream in the vanilla kernel
- As default, *anything not explicitly permitted is denied.*
- Rules are called “policies”
- Basically two kind of pre-defined policies
 - “targeted”: only daemons are confined (i.e. The ones under init.d), leaving anything else unconfined.
 - “stricted”: anything is confined, even users.



SE-Linux: access control

- SELinux has three types of access control:
 - Type Enforcement (TE): Type Enforcement is the primary control system in a MAC and used in the policies (subject, object, action)
 - Role-Based Access Control (RBAC): based on the SELinux users (does not mean that are the same of system users), but unused in the “target” policy, it is meant to define users' roles.
 - Multi Level Security (MLS) and Multi-Category Security (MCS): almost unused, it is needed to label files with a given category



The implementation

- The big issue was to find a “formula” that was able to mix security with manageability:
 - The systems are managed to operators with basic skills.
 - My objective was not to change their habits
- I decided to use:
 - SE-Linux in targeted mode
 - Multi-Category Security, assigning a special category to PDF files
 - An ad-hoc SE-Linux modules
 - Appropriate configuration of system tools



Category management

- A category was created to “label” the PDFs
- Implemented in */etc/selinux/targeted/setrans.conf*
 - *s0:c3=TopSecret*
- Any file must have this label in order to be protected, either via *chcat* or *restorecon*
- Enable user(s) to access the PDF files
 - *chcat -l +TopSecret appserv*
- SE-Linux information are stored in the “extended attributes” (xattr) of the filesystem
 - GFS is a cluster filesystem that support xattrs



Category management

- The permissions: default users can't access any category, root can access all the categories
- *TopSecret* authorization was granted to the *appserv* user

```
# semanage login -l
```

Login Name	SELinux User	MLS/MCS Range
__default__	user_u	s0
appserv	user_u	-TopSecret
root	root	SystemLow-
SystemHigh		



SE-Linux module

- Based on two files:
 - *docsecret.te*
 - Contains policies and type definitions
 - The *docsecret_t* type protect access from other confined processes that need explicit grant
 - *docsecret.fc*
 - Contains contexts to be applied to files
 - Allows to automatically label all the files under */documents* as *TopSecret*
- Files are compiled and loaded in memory as an SE-Linux module (*docsecret.pp*)



Admin access: restrictions

- Sudo was configured to allow the group *appmanager* to execute start/stop of the application server
- The “su” command is restricted to the *admin* group
- SSH access limited to:
 - operator, appmanager, admin
 - Remote root login is forbidden
 - It listen only the admin network with a controlled access
- The *root* can log only on the console
- The root password is owned by the service manager, who is formally responsible for any information loss



Admin access: protection

What happens if we execute any command through sudo or after we do a “su -”:

```
# id
```

```
uid=0 (root) gid=0 (root)  
groups=0 (root), 1 (bin), 2 (daemon), 3 (sys), 4 (adm)  
, 6 (disk), 10 (wheel)  
context=user_u:system_r:unconfined_t
```

Note: even if the user became root, SE-Linux labels are enforced, therefore the precious documents are protected.

```
# cat /documents/mydoc.pdf
```

```
cat: /documents/mydoc.pdf: Permission denied
```



Audit

- Any attempt to access the document directory must be logged
 - No logs for the *appserv* user for performance reason
 - Sent to an external syslog server, so that any attempt to delete logs are useless
- SE-Linux logs through the audit process in `/var/log/audit/audit.log`
- Configured the audit subsystem in: `/etc/audit/audit.rules`
 - `-a exit,always -S open -S truncate -F dir=/documents -F uid!=300`
 - Configured also the syslog plugin to sent to remote log server.



ACL

- Also ACLs have been placed in the directory
 - The ACL belongs to the DAC “world”
 - DAC works together with MAC, is not ignored
 - It allows further protection if someone from console disables SELinux (setenforce 0) for any reason
- The ACLs
 - `# chmod 0750 appserv:appserv /documents/`
 - `# setfacl -m appserv:rwx /documents/`
 - `# setfacl -m root:rwx /documents/`
 - `# getfacl --access /documents/ | setfacl -d -M- /documents/`



Demo now!



Conclusions

- SE-Linux is for sure very interesting, though is very hard to configure and manage. You have to find the right balance:
 - Maybe not useful for a printer server
 - Very useful in a “border” web server for internet/intranet/extranet
 - Not always certified to be used with commercial applications (eg: Oracle, check your vendor)



Conclusions

- You have to think in a “Defense-in-depth” philosophy:
 - Try to use several protection/security layers
 - Security should be close to the data we want to protect:
 - In this “case study” is important that PDF are crypted to ensure that any bug won't cause an information loss
- You can download the paper “protecting confidential files with SE-Linux” from my web sites:
- <http://www.scss.tcd.ie/Giuseppe.Paterno/>
- <http://www.gpaterno.com/>



Questions?



Thank you!!

Giuseppe “Gippa” Paternò
Visiting Researcher
Trinity College Dublin

paternog@cs.tcd.ie

<http://www.scss.tcd.ie/Giuseppe.Paterno/>