M2 CyberSecurity
Security Architecture: network, system, key management,
cybersecurity of industrial system.

Systems and Network Security

Florent Autréau - florent.autreau@imag.fr

2016 /2017

Network Security - Part 5

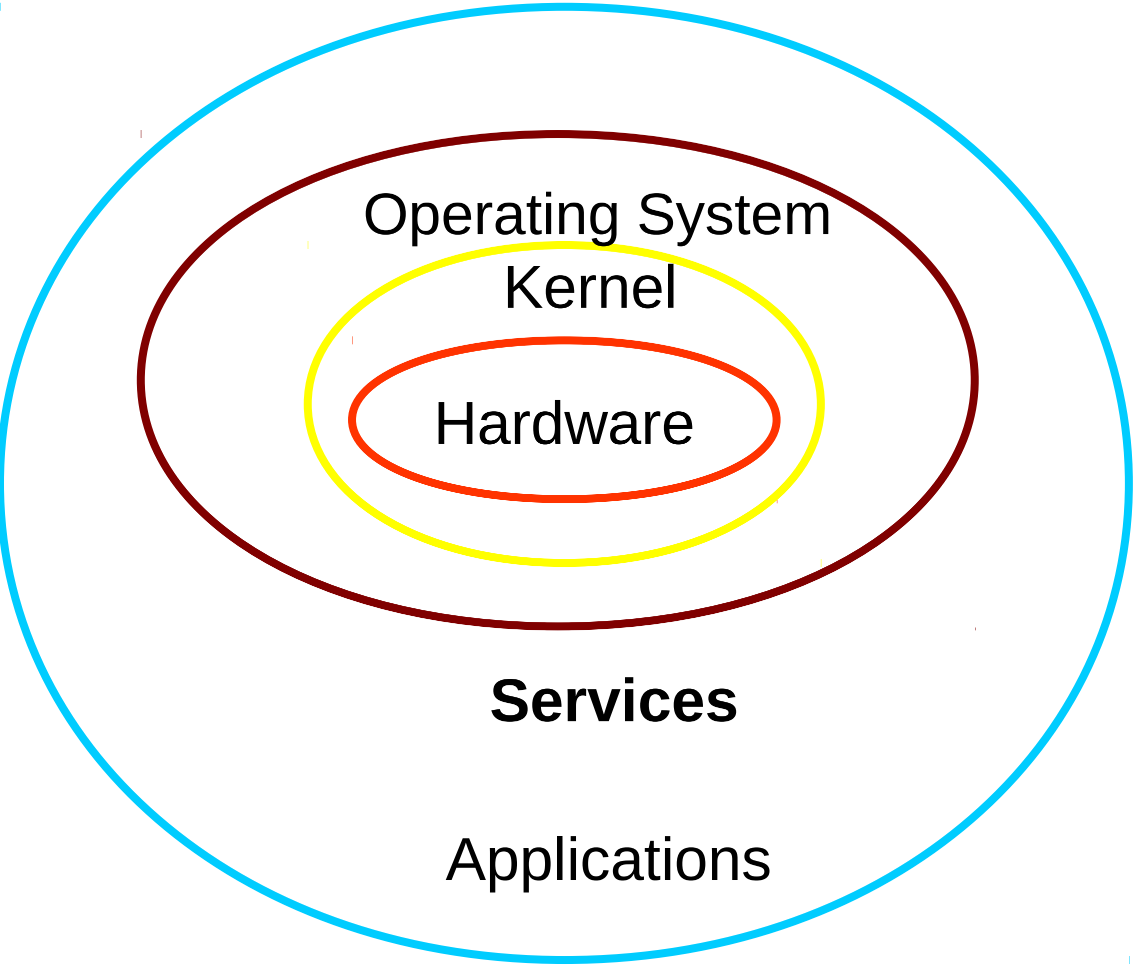
- Introduction
- Threat landscape
- Cryptography and Network Security
- Network Security – Vulnerabilities/Protection
  - IPv6 Overview
  - IPv6 Security considerations
- System Security - Hardening

Motivations

- Benefits of hardening
  - Security
  - Performance
  - Availability

Limit/control the exposure of your IS to failure, misbehavior, incident, intrusion, ...

Remember the Onion Model and its Layers of technology



OS Hardening

Installing kernel/software patches and configuring a system in order to prevent attackers from exploiting and attacking your system.

OS Hardening (cont.)

- Patches
  - Apply security patches to Linux kernel
  - Apply bug patches to software
- Security tools
  - Extra system logs and auditing
- System rules and policies
  - Restrict user privileges
  - Disabling unnecessary processes

SELinux

- Security-Enhanced Linux (SELinux)
- Linux feature that provides a variety of security policies for Linux kernel
- Included with CentOS / RHEL / Fedora Linux, Debian / Ubuntu, Suse, Slackware and many other distributions.

SELinux - Features

- Clean separation of policy from enforcement
- Well-defined policy interfaces
- Support for applications querying the policy and enforcing access control
- Independent of specific policies and policy languages
- Independent of specific security label formats and contents
- Individual labels and controls for kernel objects and services
- Caching of access decisions for efficiency
- Support for policy changes
- Separate measures for protecting system integrity (domain-type) and data confidentiality (multilevel security)
- Very flexible policy
- Controls over process initialization and inheritance and program execution
- Controls over file systems, directories, files, and open file descriptors
- Controls over sockets, messages, and network interfaces
- Controls over use of "capabilities"

SELinux – Pro/Cons

- Admin skill set (learning curve) - High
- Complex and powerful access control mechanism - Yes
- Detailed configuration required - Yes
- GUI tools to write / modify rules set - Yes
- CLI tools to write / modify rules set - Yes (see list of commands here)
- Ease of use - No (often described as horrible to use)
- Binary package - Available for most Linux distributions
- System performance impact: None
- Security Framework: Mandatory access controls using Flask
- Auditing and logging supported - Yes
- Typical user base - Enterprise users
- Documentation - Well documented

AppArmor

- AppArmor : Application Armor
- Yet Another security software for Linux which maintained and released by Novell under GPL
- Alternative to SELinux
- AppArmor works with file paths
- AppArmor is default in OpenSUSE and Suse Enterprise Linux

AppArmor - Features

- Full integration.
- Easy deployment.
- AppArmor includes a full suite of console and YaST-based tools to help you develop, deploy and maintain application security policies.
- Protects the operating system, custom and third-party applications from both external and internal threats by enforcing appropriate application behavior.
- Reporting and alerting. Built-in features allow you to schedule detailed event reports and configure alerts based on user-defined events.
- Sub-process confinement. AppArmor allows you to define security policies for individual Perl and PHP scripts for tighter Web-server security.

AppArmor – Pro/Cons

- Admin skill set (learning curve) - Medium
- Complex and powerful access control mechanism - Yes.
- Detailed configuration required - Yes.
- GUI tools to write / modify rules set - Yes (yast2 and wizards).
- CLI tools to write / modify rules set - Yes.
- Ease of use - Yes (often described as less complex and easier for the average user to learn than SELinux).
- Binary package - Available for Ubuntu / Suse / Opensuse and distros.
- System performance impact - None.
- Security Framework - Mandatory access controls.
- Auditing and logging supported - Yes.
- Typical user base - Enterprise users.
- Documentation - Documented (mostly available from Opensuse and Suse enterprise Linux).

GRsecurity

- Linux Kernel patch
- set of patches for the Linux kernel with an emphasis on enhancing security. It utilizes a multi-layered detection, prevention, and containment model
- GPL Licence.
- provides
  - Non-Executable Stack
  - Change root (chroot) hardening
  - /tmp race prevention
  - Extensive auditing
  - Additional randomness in the TCP/IP stack
  - /proc restrictions

GRSecurity - Features

- An intelligent and robust Role-Based Access Control (RBAC) system that can generate least privilege policies for your entire system with no configuration
- Change root (chroot) hardening
- /tmp race prevention
- Extensive auditing
- Prevention of arbitrary code execution, regardless of the technique used (stack smashing, heap corruption, etc)
- Prevention of arbitrary code execution in the kernel
- Randomization of the stack, library, and heap bases
- Kernel stack base randomization
- Protection against exploitable null-pointer dereference bugs in the kernel
- Reduction of the risk of sensitive information being leaked by arbitrary-read kernel bugs
- A restriction that allows a user to only view his/her processes
- Security alerts and audits that contain the IP address of the person causing the alert

GRSecurity – Pro/Cons

- Admin skill set (learning curve) - Low.
- Complex and powerful access control mechanism - No (it is simpler to administer than other two implementations. Also, policies are simpler to create, since there are no roles or complicated domain/file transitions).
- Detailed configuration required - No (works in learning mode).
- GUI tools to write / modify rules set - No.
- CLI tools to write / modify rules set - Yes (gradm tool).
- Ease of use - Yes.
- Binary package - Available for Ubuntu / RHEL / CentOS / Debian distros.
- System performance impact - None.
- Security Framework - Mandatory access controls (precisely, it is a RBAC implementation) using access control lists.
- Auditing and logging supported - Yes.
- Typical user base - Webserver and hosting companies.
- Documentation - unfortunately, is not well documented.

Feature	SELinux	AppArmor	grsecurity
Automated	No (audit2allow and system-config-selinux)	Yes (Yast wizard)	Yes (auto traning / gradm)
Powerful policy setup	Yes (very complex)	Yes	Yes
Default and recommended integration	CentOS / RedHat / Debian	Suse / OpenSuse	Any Linux distribution
Training and vendor support	Yes (Redhat)	Yes (Novell)	No (community forum and lists)
Recommend for	Advanced user	New / advanced user	New users
Feature	Pathname based system does not require labelling or relabelling filesystem	Attaches labels to all files, processes and objects	ACLs

Minimize the size of the Target

- Careful review and inspection of required software services
- Removal of uneeded software packages
- Define or use predefined qualified profile (ex : VM for web server, mail server, ...)
- Windows OS:
  - Use install/removal interface provided in Control Panel
- Solaris :
  - Review with **pkginfo** and remove with **pkgrm**
  - Use hardening toolkit such as JASS

Minimize the size of the Target (2)

- MacOS :
  - Use the '**software update**' tool
  - Or the **patch** utility (cli)
- Linux (RedHat/CentOS/Fedora):
  - Review with **yum list**
  - Remove with **yum remove**
- Linux (Debian/Ubuntu):
  - Review with **dpkg –list** and **dpkg --info**
  - Remove with **dpkg** or **apt-get remove**

Correct known problems

- Apply patches to remaining software
- Windows OS:
  - Use windowsUpdate (requiring legitimate copy)
  - For non microsoft software, use PSI -Personal Security Inspector - https://secunia.com/vulnerability_scanning/personal/
- Solaris :
  - Use patchfinder http://sunsolve.sun.com/patchfinder/
  - Or Review and download patches on sunsolve.com, apply them with pkgadd

Correct known problems (2)

- Linux (RedHat/CentOS/Fedora):
  - **yum update**
- Linux (Debian/Ubuntu):
  - **apt-get update**
  - **apt-get upgrade**

When not available on repositories/software depot, patching process might require build/recompile the given software (when opensource), then test and installation/deployment.

Protect Servers Physical Console Access

- Setup Time-out for Login Shells
- Setup Screen Locking (such as **vlock** on linux)
- GUI Screen Locking
- Disable Ctrl+Alt+Delete
  - Comment line in /etc/inittab
  - Or disable the reboot action in /etc/event.d/control-alt-delete file
- ...

Configure kernel parameters

- Network configuration:
  - Disable IP forwarding, drop source routed
  - Protect against SYN floods, Smurf attacks
  - Drop ICMP redirects, reduce ARP timeouts
  - Help stop remote network mapping efforts
- Other kernel parameters:
  - Enable stack protection
  - Prevent core dumps
  - Set limits on processes

Configure kernel parameters (2)

- Linux: modify /etc/sysctl.conf

```
# Turn on execshield
kernel.exec-shield=1
kernel.randomize_va_space=1
# Enable IP spoofing protection
net.ipv4.conf.all.rp_filter=1
# Disable IP source routing
net.ipv4.conf.all.accept_source_route=0
# Ignoring broadcasts request
net.ipv4.icmp_echo_ignore_broadcasts=1
net.ipv4.icmp_ignore_bogus_error_messages=1
# Make sure spoofed packets get logged
net.ipv4.conf.all.log_martians = 1
```

Physical Layer Controls

- *Locked perimeters and enclosures*
- *Electronic lock mechanisms for logging & detailed authorization*
- *Video & Audio Surveillance*
- *PIN & password secured locks*
- Biometric authentication systems
- Data Storage Cryptography
- *Electromagnetic Shielding*

Secure the boot process

- Update the BIOS (if any)
- Configure password protection in BIOS
- Prevent boot from untrusted sources
- Set Boot Loader Password (as in GRUB/LILO)
- Enable Authentication for Single-User Mode (in /etc/inittab)
- Disable Interactive Hotkey Startup

Link Layer Controls

- MAC Address Filtering- Identifying stations by address and cross-referencing physical port or logical access
- *Do not use VLANs to enforce secure designs. Layers of trust should be physically isolated from one another, with policy engines such as firewalls between.*
- *Wireless applications must be carefully evaluated for unauthorized access exposure. Built-in encryption, authentication, and MAC filtering may be applied to secure networks.*

How To

- Implement access restrictions using MAC address

```
/sbin/iptables -A INPUT -m mac --mac-source 00:21:EA:91:A0:08 -j DROP
```

Network Layer Controls

- Route policy controls - Use strict anti-spoofing and route filters at network edges
- Firewalls with strong filter & anti-spoof policy
- *ARP/Broadcast monitoring software*
- *Implementations that minimize the ability to abuse protocol features such as broadcast*

How To

- Firewall
  - Windows : use build-in firewall (and security center)
  - Unix : iptables/netfilter
- Encrypt data communications (ipsec, ssh, ssl)
- Avoid network traffic in clear
  - On Unix, avoid FTP, telnet, and R*services (Rlogin / Rsh) either by uninstalling or disabling.

Transport Layer Controls

- Strict firewall rules limiting access to specific transmission protocols and sub-protocol information such as TCP/UDP port number or ICMP type
- Stateful inspection at firewall layer, preventing out-of-state packets, “illegal” flags, and other phony packet profiles from entering the perimeter
- Stronger transmission and layer session identification mechanisms to prevent the attack and takeover of communications

How To

- Stateful Firewall
- Filter access to services with tcpwrappers

How To (2)

- Review running services and enable only the necessary ones
  - Windows : use Task Manager and Service in Control Panel
  - On Unix, disable all unnecessary services and daemons (services that runs in the background).
    - By using **chkconfig**
- Compartimentation :
  - One Network Service Per System or VM Instance
  - Use Compartimentation mechanism (chroot, jail, vm, ...)
- Find Listening Network Ports :
  - by using **netstat -tulpn**

Session Layer Controls

- Encrypted password exchange and storage
- Accounts have specific expirations for credentials and authorization
- Protect session identification information via random/cryptographic means
- Limit failed session attempts via timing mechanism, not lockout

How To

- Implement Password Policy (with GPO)
  - User Accounts and Strong Password Policy
  - Password Aging
    - **chage** -l userName
  - Restricting Use of Previous Passwords
  - Locking User Accounts After Login Failures
    - Using **faillog**
  - Verify No Accounts Have Empty Passwords
  - Disable Root Login
  - Make Sure No Non-Root Accounts Have UID Set To 0
  - Review /etc/sudoers

Presentation Layer Controls

- Careful specification and checking of received input incoming into applications or library functions
- Separation of user input and program control functions- input should be sanitized and sanity checked before being passed into functions that use the input to control operation
- Careful and continuous review of cryptography solutions to ensure current security versus know and emerging threats

Application Layer Controls

- Application level access controls to define and enforce access to application resources. Controls must be detailed and flexible, but also straightforward to prevent complexity issues from masking policy and implementation weakness
- Standards, testing, and review of application code and functionality-A baseline is used to measure application implementation and recommend improvements
- IDS systems to monitor application inquiries and activity
- Some host-based firewall systems can regulate traffic by application, preventing unauthorized or covert use of the network.

DNS Hardening Tips

- Enable bind chroot support.
- Apply port restrictions in firewall.
- Customize logging as desired.
- Authoritative DNS servers should not be used as resolving or caching DNS servers.
- Disable recursive queries on authoritative servers.
- Enable numerous security settings in /etc/named.conf to suit your environment.

- Consider using opendns servers

Mail Server Hardening Tips

- Chroot postfix (manual process)
- Ensure unauthorized parties can’t relay
- Establish port restrictions and access control with iptables.
- Configure smtp restrictions in postfix.
- Use ldap or access file to restrict inbound mail to valid users
- Anti-virus / Anti-Spam

Apache Hardening Tips

- Hide the Apache Version number, and other sensitive information.
- Ensure that files outside the web root are not served
- Turn off directory browsing, CGI execution, options, support for .htaccess.
- Run mod_security
- Tune and limit
  - Lower the Timeout value, Limit Request, Limiting the size of an XML Body, Limiting Concurrency
  - Restricting Access by IP, Adjusting KeepAlive settings
- Run Apache in a Chroot environment

How-To

- VM protection (hypervisor)
- Monitor activity and log management
  - Monitor Suspicious Log Messages With Logwatch / Logcheck
  - System Accounting with auditd
- ...

AuthN Considerations

- Enable two-factor authentication when available

  http://www.turnon2fa.com/
- Check your login history, the list of authorized devices and/or sources

Ressources

- CIS Hardening Guide
  - CIS Score Toolkit – cissecurity.org
- Win2k Hardening Guide
  - MSAT – Microsoft Security Assesement Tool
- OWASP – hardening guide for web application
  - Apache stk
  - IIS stk