system hardening - an introduction



general



- Hardening a system against:
 - (Security) Threats
- Threat: Everything that can possibly violate the security goals:
 - Confidentiality
 - Integrity
 - Availability



Why doing it and what's the impact?

Decrease the attack surface

Decrease available information after

successful exploitation through an

attacker

Increase the likelihood of detection of an event

Learn more about your system

Installed software

Configuration

Inner workings

- Depends on the use case
- General impacts
 - Nice:
 - Performance
 - Stability
 - Manageability
 - System Know How
 - Bad:
 - Complexity
 - Usability



Concepts related to hardening





Reduction of attack surface

Death Star Defensive Systems



- Less attack surface -> less security risk
- But beware, only one risk is needed to make your death star explode
- Therefore think in layers and minimize the exposure on each layer
- Similar to
 - Minimum Exposure
 - \circ Keeping it simple



Defense in depth

- Define measures for each layer
- If one line of defense gets breached, another layer can halt the attacker





Principle of least privilege



What is a privilege?

The ability to do something like accessing or modifying a resource

Restrict the privileges of parties within a system, so the party can only access the information needed.

Any subject (tool / user / service / ...) should only run with the minimal privileges needed to complete a task

Other concepts

l

- Simplicity
- Open Design
- Compartmentalization
- Secure, Fail-Safe Defaults
- No Single Point of Failure
- Log security-relevant system events
- Usable security mechanisms



What measures to choose?





What measures to choose ?

Risk management !

- 1. First get to know your system
- 2. Identify your threats
- 3. Rate them, order, prioritize
- 4. Act based on risks (e.g. do nothing, find requirements based on threats)

- Scale the depth of the process to meet your security goals.
- A more formal process would be the SQUARE process





Sources of measures



BSI Grundschutz Katalog

- Provides a baseline of protections
- Helps to identify components, threats and measures

Common Vulnerabilities and Exposures (CVE)

- Database of public information of exploits, vulnerability and exposures
- Helps to identify threats and measures

More sources

- Center of internet security (very detailed harding for applications)
- Cisofy (harding for linux and unix systems)
- Security Technical Implementation Guides (STIGs)
- NIST SP 800-123: Guide to General Server Security
- \circ NSA Guides
- Google + RTFM (the manual sometimes is indeed useful)

hardening steps

- 1. Nice to have: isolate the system
- 2. Take a known good system state as start point
- 3. Create an list of components installed
- 4. Find measures
- 5. Apply measures
- 6. Create a baseline defined on the hardened state
- 7. Return the system into production
- 8. Use the system as template



system documentation

- Step 1: Do it
 - Level 0: baseline + changes
 - Level 1: dependencies, configurations, maintenance access, responsibilities
- Can be easily combined with checklists !
- Some tools:
 - Markdown based tools
 - Notebook
 - Confluence
 - MS stuff:
 - OneNote
 - Word
 - zim (OneNote Alternative)
 - echo "\$EDITOR"





Short Example

Approach: Layer by Layer

Physical Access



- Threat: Attacker gains physical access to the machine and does evil stuff.
- Measures:
 - Set a boot password (hdd, boot and master password)
 - Lock the PC if booted
 - Use the full disk encryption (LUKS vs HDD built in encryption)
- Threat: Attacker modifies the boot code.
- Measures:
 - Use Secure Boot with our own keys
 - Sign the boot code and verify the integrity of the boot code
- Threat: Attacker modifies the hardware
- Measures:
 - Use the built in intrusion sensor
 - Or lots of duct tape

Network Layer



- Threat: Attacker intercepts our network traffic
- Measure:
 - Encryption
 - TLS, Secure NTP, DNS,
- Threat: Attacker tracks us per MAC Address or SSID we search
- Measure
 - Random MAC for searching SSIDs
 - Random MAC for connection to a network
- Threat: Attacker attacks the protocol stack
- Measure:
 - Use a hardened kernel
 - Harden the kernel settings for networking
 - Use applications to detect the attacker (Intrusion detection systems (IDS))

Network Layer - In detail

- Bind services to local host
- Use a firewall
- Harden the kernel and applications for network usage
- Disable IPv6 if not needed
- Hardening of TLS (e.g. RFC 7525)
- VPN, Encrypted Tunnels (e.g. Proxies)
- Encryption of DNS is a rising issue





Host layer



- Asset: a arch linux install
- Threat: Attacker attacks on of our applications in user space
- Measure:
 - Isolate the application (systemd (per cgroups), chroot, container, docker)
- Threat: Malware infects our system
- Measure:
 - Install antivirus software
 - Check for changes in the system (e.g. aide, tripwire)
- Threat: Sensitive key material gets stolen
- Measures:
 - Move keys to a secure storage (e.g. a hardware dongle)
 - Use a decent key management to minimize impact of compromisation

Encryption



- lug
- Keep your data confidential
- Per disk:
 - home, root, var,
 - don't forget swap
- Per folder:
 - per fuse-fs in the sync folder (e.g encfs)
 - encrypt the home drives of the user
- Per application
- A word of warning on LUKS disk encryption:
 - Do not forget to backup the HEADERS or it will byte you if the drives failed in the first block

Application Layer - e.g. Browser

uff UG

- Asset: a browser
- Threat: Attacker exploits our system via the browser
- Measure:
 - Isolate the browser
 - \circ Harden the browser
- Threat: Attacker tracks us across different
- Measure:
 - Enforce HTTPs usage
 - Use a hardened browser
 - Use plugins against tracking

Isolation of applications under linux

- fully virtualized environment
 - qemu + kvm, virtualbox, vmware,
- namespaces (container)
 - docker, lxc, nspawn, ...
 - firejail !!!
- In a jail / chroot

•••

- Limited per cgroups
 - Can be done per systemd unit files (e.g. limit ressources, block access to files, ...)
- As a low privileged user
 - never a user who is root or is intended to become root
- Don't even think about it



systemd-nspawn

- Container without docker or lxc
- Nice:
 - Comes pre installed with most distros
 - Only depends on systemd (and the kernel)
 - Can be easily integrated (per systemd services)
 - Can use predefined images
 - Many more features
- Bad:

•••

- o ...???
- Examples: Wordpress, skype, steam,





Data Layer



- Asset: top secret documents
- Threat: Attacker reads documents
- Measure:
 - Use a sophisticated access control mechanism
 - Encryption
 - Hide them (e.g. per steganographie)
- Threat: Attacker deletes documents
- Measure:
 - Make backups and keep logs

Hardening tips for applications



Specific Applications - Web Server



- Permissions
 - Run the web server as low privileged user
- Use TLS
 - For encryption of the traffic
 - For authentication of the user
 - Use secure and modern TLS settings
 - Maybe use techniques like DANE or HSTS
- Limit Information send by the server (e.g. Header, error messages)
- Disable error traces
- Use web application firewall (WAF; e.g. modsecurity)
- Isolate the web server if possible
- Use a reverse proxy for better auditing and more

Firewalls

Typical Firewalls

- iptables
- nftables
- arptables
- ebtables

"Classic" ACLs:

- TCP Wrapper
- SELinux

I'M NOT SAYING IT'S THE FIREWALL

BUT IT'S THE FIREWALL

Specific Applications - SSH



- Authentication
 - Two-factor Authentication with SSH (per pam)
 - Disable root login
 - Disable passwords (use keys)
 - Limit authentication tries
 - Hardware token for more secure key management
- Disable old protocols version (version 1)
- Use DNS hostname checking
- Watch for changes of the signature of the server
- User and groups configuration;
 - Least privilege for users (e.g. user which is only allowed to read certain files)
 - Whitelist users
- Change the port
- Use modern crypto

Specific Applications - sudo; git

l

sudo

git

- Enable auditing
- Use the fine-grained access control mechanisms
- Get mails on sudo access

- use a proper identity
- sign your commits
- secure access to the repo
- follow the git best practices

Specific Applications - Key Storage



- A difficult topic, but here's a simple example
- Store the asymmetric keys in a hardware dongle and keep a backup offline
 - Only subkeys are on the token, master key is offline
 - Key can not be extracted from the token
 - Cryptographic operations are done on the token
- What can be done with them:
 - Create a set of gpg keys for multiple usages:
 - For SSH (e.g. for authentication)
 - For GPG (e.g. for sign and encryption; mails, git, passwords)
 - For TLS (e.g. your own CA or client cert)
- Some tokens
 - Yubikey 4
 - GPG Card





Tools for automated auditing and hardening



Nessus Scanner



- Commercial Tool with a community version
- Used to assess vulnerabilities in systems

Sec	urityCenter								
		Dasnobard							Add C
event	Vulnerabiliti g								Add Component
vent Vul	nerabilities - Top 100 Vulnerabilities			ile O O	Event Vulnerabilities	Exploitable and	Malware		ili O
gin ID	Name		Family	Total		Total	Total %	Exploitable	Malware
0001	Users Discovered		Generic	72	Vulnerabilities	89	0%	32	0
0000	Host Discovered		Generic	40	Check	0	07	0	
1023	Network Statistics		Generic	29	Overflow				
035	Web Access Statistics		Generic	27	Dis	3/	0%		U U
019	Login Statistics		Generic	26	Dos	46	0%	0	0
009	Detected-Change Statistics		Generic	25	Statistics	17972	100%	0	0
029	Stats Daemon Event Statistics		Generic	23	injection	3	0%	0	0
011	DNS Statistics		Generic	23	Detection	3147	17%	0	0
005	Connection Statistics		Generic	23	Disclosure	22	0%	0	0
030	System Statistics		Generic	21	Execution	10	03		0
006	Continuous Statistics		Generic	21	Butase				
039	Indicator Statistics		Generic	18	oyposa	2	0%	0	0
038	Other Statistics		Generic	18	Last lijedatori 6. ekontes age				
036	Web Error Statistics		Generic	18	Event Vulnerabilities - 1	on 10 Systems			4.0
					radur anuterenutries -	op to ayatems			III. V
ent Vuli	nerabilities - Event Summary Statistics		B00341	ılı 0 🔍	35	Low	Medium	I High	Critical
ent Vuli	nerabilities - Event Summary Statistics		■ 800541 ■ 804833 ■ 804835 ■ 804838 ■ 804888 ■ 804888 ■ 804888	ılı O Ø	Si de la construcción de la cons		Verban	a man	санса 9 ⁰ 9 ⁰
ent Vuli ent Vuli ent Vuli	nerabilities - Event Summary Statistics		■ 60041 ■ 60483 ■ 60483 ■ 60453 ■ 60453 ■ 60454 ■ 604545	sL ○ ♥	to the set of the set	Low	Verban Status	a man	р ^о р [*] е
ent Vuli ent Vuli	rerabilities - Event Summary Statistics		■ 6004 ■ 6443 ■ 6443 ■ 64637 ■ 66637 ■ 66638 ■ 66646	sk ○ ♥	to the set of the set	i urr	rability Trending	a max	р стил р р р р ф ф ф ф ф ф
ent Vuli ent Vuli	rerabilities - Event Summary Statistics			sh ○ ♥	a de la construcción de la const	t car	a bedue		р ^о р ^о а. О
ent Vuli dere 12 har	rerabilities - Event Summary Statistics		€ 6004 6043 6043 6043 6043 6043 6043 6043	ak ○ ♥	terrorent Vulnerabilities - 2 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1	t cre	rability Trending	a man	1 cmud
dend 22 here	erabilites - Browser Event Valnerabilities			al. O O	terrer Valnerabilities - 2	S Day Event Vulne	and and a second s		р ⁰ р ⁰ ф ф
detel 2 here	erabilities - Event Summary Statistics			at. 0 0	Server Vulnerabilities - 2 Sever Vulnerabilitie	5 Day Event Vulne	rability Trending		р ^ф д ^ф d 0
ent Vul	erabilities - Browser Event Vulnerabilities			at. 0 0 at.	Le laborat 3 hone au Contraction and a laboration and a	5 Day Event Vulne	a sedual		6 ⁰ 0 ⁰
ent Vul	erabilities - Event Summary Statistics			al. O O al. O O	Provent Vulnerabilities - 2 Provent Vulnerab	is Day Event Vulne	rability Trending		р ⁰ д ⁰ d. 0
ent Vul	erabilities - Event Summary Statistics			al. O O	Landra Homes	5 Day Event Vulne	a brites		р соция р ф ф ф ф л h ©
entre Vult	erabilities - Event Summary Statistics			1. 0 0 1. 0 0	The second secon	S Day Event Vulne	ability Trending		

lynis



Audit tool from cisofy

- Can audit linux system igodol
 - Kernel Settings, Distro Settings, Applications Settings and more

]

- Can be used for penetration tests igodol
- Can audit dockerfiles

Lynis security scan details:

Tests performed : 198 Plugins enabled : 1

Components:

- Firewall [V] [X]
- Malware scanner

cis-cat

l

- From the center for internet security
- Can run with their benchmarks
- Is free and has a pro version for large scale usage

e <u>O</u> ptio	ns Help he CENTER for NTERNET SECURITY	Configuration	Assessm	ent To
atform:	Ubuntu 64-bit version 3.11.0-15-gene	ric JRE: Sun Mi	icrosystem	s Inc. 1.
Benchma	ark Execution Status			
Number	Title		Time	Result
93/154	Set User/Group Owner and Permission on /etc/cri	on weekly	<1 second	Fail
94/154	Set User/Group Owner and Permission on /etc/cro	on.monthly	<1 second	Fail
95/154	Set User/Group Owner and Permission on /etc/cri	on.d	<1 second	Fail
96/154	Restrict at/cron to Authorized Users		<1 second	Fail
97/154	Set Password Creation Requirement Parameters	Using pam cracklib	<1 second	Fail
98/154	Set Lockout for Failed Password Attempts		<1 second	Fail
99/154	Limit Password Reuse		<1 second	Fail
100/1	Set SSH Protocol to 2		<1 second	Fail
101/1	Set LogLevel to INFO		<1 second	Fail
102/1	Set Permissions on /etc/ssh/sshd_config		<1 second	Fail
103/1	Disable SSH X11 Forwarding		<1 second	Fail
104/1	Set SSH MaxAuthTries to 4 or Less	Þ	<1 second	Fail
105/1	Set SSH IgnoreRhosts to Yes		<1 second	Fail
106/1	Set SSH HostbasedAuthentication to No		<1 second	Fail
107/1	Disable SSH Root Login		<1 second	Fail
108/1	Set SSH PermitEmptyPasswords to No		<1 second	Fail
109/1	Do Not Allow Users to Set Environment Options		<1 second	Fail
110/1	Use Only Approved Cipher in Counter Mode		<1 second	Fail



- Based on the Security Content Automation Protocol (SCAP)
- Currently a hot standard for auditing / hardening
- Has a complete pipeline built for hardening systems
- SCAP can be used with multiple tools (most of the time)



do it yourself (diy)



- If you got a specific set of requirements
 - most task are easy to automate (e.g. ansible, bash, python, ...)
 - many good examples are out there
- Take a look at <u>http://dev-sec.io</u>
 - Open source and automated (Big Thx to Simon for adding this)



the unspoken



- Nexpose (Rapid7)
- Secutor Compliance Automation Toolkit (S-CAT) (ThreatGuard)
- SCAP Compliance Checker (SPAWAR)
- BigFix Compliance (IBM)
- System Center Configuration Manager (Microsoft)
- Security Center (tenable networks)



the unspoken - extended edition



- Secutor Prime 5 (ThreatGuard)
- Qualys (Qualys)
- SAINT Security Suite 8 (Saint)
- BMC Server Automation (bmc)
- IBM Endpoint Manager (IBM)
- Policy Auditor 6.2 (intel)
- Tripwire Enterprise 8 (Tripwire)



More useful links



- Sources of measures in the beginning
- Web
 - Archlinux Wiki Security
 - Other distro wikis (e.g. red hat security guide)
- Books
 - Linux Hardening in Hostile Networks. Server Security from TLS to Tor
 - Network Hardening: An Automated Approach to Improving Network Security
- Other
 - Google and RTFM





thats it ! thx