CS527 Software Security

Secure Software Lifecycle

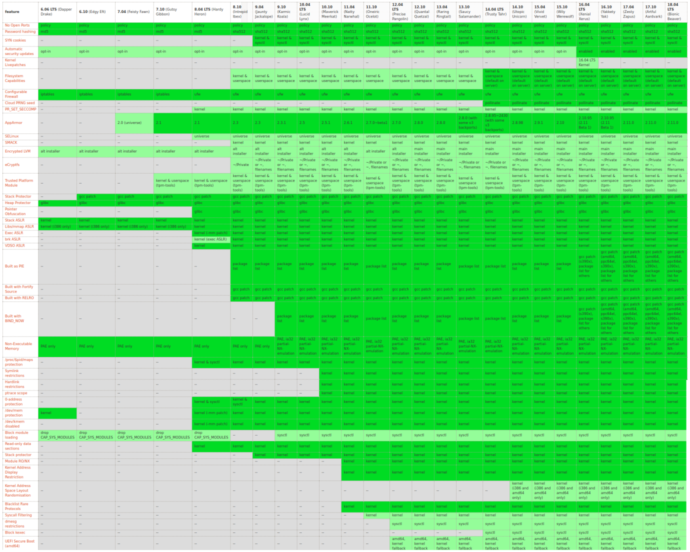
Mathias Payer

Purdue University, Spring 2018

Software liveliness

- Software is not a one-shot effort
- Software development, production, and maintenance are cost/labor intensive
- Software life-time can outlive hardware

Example: Ubuntu security evolution



feature	6.06 LTS (Dapper Drake)	6.10 (Edgy Eft)	7.04 (Feisty Fawn)	7.10 (Gutsy Gibbon)	8.04 LTS (Hardy Heron)	8.10 (Intrepid Ibex)	9.04 (Jaunty Jackalope)	9.10 (Karmic Koala)	10.04 LTS (Lucid Lynx)	10.10 (Maverick Meerkat)	11.04 (Natty Narwhal)	11.10 (Oneiric Ocelot)	12.04 LTS (Precise Pangolin)	12.10 (Quantal Quetzal)	13.04 (Raring Ringtail)	13.10 (Saucy Salamander)	14.04 LTS (Trusty Tahr)	14.10 (Utopic Unicorn)	15.04 (Vivid Vervet)	15.10 (Wily Werewolf)	16.04 LTS (Xenial Xerus)	16.10 (Yakkety Yak)	17.04 (Zesty Zapus)	17.10 (Artful Aardvark)	18.04 LTS (Bionic Beaver)
No Open Ports	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy	policy
Password hashing	md5	md5	md5	md5	md5	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512	sha512
SYN cookies	--	--	--	--	--	--	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl	kernel & sysctl
Automatic security updates	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	opt-in	enabled	enabled	enabled	enabled	enabled
Kernel Livepatches	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	16.04 LTS Kernel	--	--	--	--
Filesystem Capabilities	--	--	--	--	--	kernel & userspace	kernel & userspace	kernel & userspace	kernel & userspace	kernel & userspace	kernel & userspace	kernel & userspace	kernel & userspace	kernel & userspace	kernel & userspace	kernel & userspace	kernel & userspace (default on server)	kernel & userspace (default on server)	kernel & userspace (default on server)	kernel & userspace (default on server)	kernel & userspace (default on server)	kernel & userspace (default on server)	kernel & userspace (default on server)	kernel & userspace (default on server)	kernel & userspace (default on server)
Configurable Firewall	iptables	iptables	iptables	iptables	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw	ufw
Cloud PRNG seed	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	pollinate	pollinate	pollinate	pollinate	pollinate	pollinate	pollinate	pollinate	pollinate
PR_SET_SECCOMP	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
AppArmor	--	--	2.0 (universe)	2.1	2.1	2.3	2.3	2.3.1	2.5	2.5.1	2.6.1	2.7.0~beta1	2.7.0	2.8.0	2.8.0	2.8.0 (with some v3 backports)	2.8.95~2430 (with some v3 backports)	2.8.98	2.9.1	2.10	2.10.95 (2.11 Beta 1)	2.10.95 (2.11 Beta 1)	2.11.0	2.11.0	2.11.0
SELinux	--	--	--	--	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe	universe
SMACK	--	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Encrypted LVM	alt installer	alt installer	alt installer	alt installer	alt installer	alt installer	alt installer	alt installer	alt installer	alt installer	alt installer	alt installer	alt installer	main installer	main installer	main installer	main installer	main installer	main installer	main installer	main installer	main installer	main installer	main installer	main installer
eCryptfs	--	--	--	--	--	~/Private	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames	~/Private or ~, filenames
Trusted Platform Module	--	--	--	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)	kernel & userspace (tpm-tools)
Stack Protector	--	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch
Heap Protector	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc
Pointer Obfuscation	--	--	--	--	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc	glibc
Stack ASLR	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Libs/mmap ASLR	kernel (i386 only)	kernel (i386 only)	kernel (i386 only)	kernel (i386 only)	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Exec ASLR	--	--	--	--	kernel (-mm patch)	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
brk ASLR	--	--	--	--	kernel (exec ASLR)	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
VDSO ASLR	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Built as PIE	--	--	--	--	--	package list	package list	package list	package list	package list	package list	package list	package list	package list	package list	package list	package list	package list	package list	package list	gcc patch (s390x), package list for others	gcc patch (amd64, ppc64el, s390x), package list for others	gcc patch (amd64, ppc64el, s390x), package list for others	gcc patch (amd64, ppc64el, s390x), package list for others	gcc patch (amd64, ppc64el, s390x), package list for others
Built with Fortify Source	--	--	--	--	--	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch
Built with RELRO	--	--	--	--	--	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch	gcc patch
Built with BIND_NOW	--	--	--	--	--	--	--	package list	package list	package list	package list	package list	package list	package list	package list	package list	package list	package list	package list	package list	gcc patch (s390x), package list for others	gcc patch (amd64, ppc64el, s390x), package list for others	gcc patch (amd64, ppc64el, s390x), package list for others	gcc patch (amd64, ppc64el, s390x), package list for others	gcc patch (amd64, ppc64el, s390x), package list for others
Non-Executable Memory	PAE only	PAE only	PAE only	PAE only	PAE only	PAE only	PAE only	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation	PAE, ia32 partial-NX-emulation
/proc/$pid/maps protection	--	--	--	--	kernel & sysctl	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Symlink restrictions	--	--	--	--	--	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Hardlink restrictions	--	--	--	--	--	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
ptrace scope	--	--	--	--	--	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
0-address protection	--	--	--	--	kernel & sysctl	kernel & sysctl	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
/dev/mem protection	kernel	--	--	--	kernel (-mm patch)	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
/dev/kmem disabled	--	--	--	--	kernel (-mm patch)	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Block module loading	drop CAP_SYS_MODULES	drop CAP_SYS_MODULES	drop CAP_SYS_MODULES	drop CAP_SYS_MODULES	drop CAP_SYS_MODULES	--	--	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl
Read-only data sections	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Stack protector	--	--	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Module RO/NX	--	--	--	--	--	--	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Kernel Address Display Restriction	--	--	--	--	--	--	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Kernel Address Space Layout Randomisation	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	kernel (i386 and amd64 only)	kernel (i386 and amd64 only)	kernel (i386 and amd64 only)	kernel (i386 and amd64 only)	kernel (i386 and amd64 only)	kernel (i386 and amd64 only)	kernel (i386 and amd64 only)	kernel (i386 and amd64 only)
Blacklist Rare Protocols	--	--	--	--	--	--	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
Syscall Filtering	--	--	--	--	--	--	--	--	--	--	--	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel	kernel
dmesg restrictions	--	--	--	--	--	--	--	--	--	--	--	--	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl
Block kexec	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl	sysctl
UEFI Secure Boot (amd64)	--	--	--	--	--	--	--	--	--	--	--	--	amd64, kernel fallback	amd64, kernel fallback	amd64, kernel fallback	amd64, kernel fallback	amd64, kernel fallback	amd64, kernel fallback	amd64, kernel fallback	amd64, kernel fallback	amd64, kernel fallback	amd64, kernel fallback	amd64, kernel fallback	amd64, kernel fallback	amd64, kernel fallback

Figure 1:

Example: Ubuntu security evolution

- Configuration
- Subsystems
- Mandatory Access Control (MAC)
- Filesystem encryption
- Trusted Platform Module
- Userspace Hardening:
- Kernel Hardening
- See `https://wiki.ubuntu.com/Security/Features`

Example: Ubuntu security evolution

- Configuration:
  - No Open Ports;
  - Password hashing;
  - SYN cookies;
  - Automatic security updates;
  - Kernel Livepatches

Example: Ubuntu security evolution

- Subsystems:
  - Filesystem Capabilities;
  - Configurable Firewall;
  - Cloud PRNG seed;
  - PR_SET_SECCOMP

Example: Ubuntu security evolution

- Mandatory Access Control (MAC):
  - AppArmor;
  - SELinux;
  - SMACK

Example: Ubuntu security evolution

- Filesystem encryption:
  - Encrypted LVM;
  - eCryptfs

Example: Ubuntu security evolution

- Trusted Platform Module

Example: Ubuntu security evolution

- Userspace Hardening (1/2):
  - Stack Protector;
  - Heap Protector;
  - Pointer Obfuscation;
  - Address Space Layout Randomisation (ASLR):
    - Stack ASLR;
    - Libs/mmap ASLR;
    - Exec ASLR;
    - brk ASLR;
    - VDSO ASLR

Example: Ubuntu security evolution

- Userspace Hardening (2/2):
  - Built as PIE;
  - Built with Fortify Source;
  - Built with RELRO;
  - Built with BIND_NOW;
  - Non-Executable Memory;
  - /proc/$pid/maps protection;
  - Symlink restrictions;
  - Hardlink restrictions;
  - ptrace scope

Example: Ubuntu security evolution

- Kernel Hardening (1/2):
  - 0-address protection;
  - /dev/mem protection;
  - /dev/kmem disabled;
  - Block module loading;
  - Read-only data sections;
  - Stack protector;
  - Module RO/NX;
  - Kernel Address Display Restriction;
  - Kernel Address Space Layout Randomisation

Example: Ubuntu security evolution

- Kernel Hardening (2/2):
  - Blacklist Rare Protocols;
  - Syscall Filtering;
  - dmesg restrictions;
  - Block kexec;
  - UEFI Secure Boot (amd64)

Secure SE versus SE

What is the difference between software engineering and secure software engineering?
If we have secure SE, why not also $\{$reliable $|$ robust $|$ resilient $|$ reproducible $|$ trusted $|$ verifiable $|$ . . . $\}$ SE?

Secure SE versus SE

Software engineering (SE) is concerned with developing and maintaining software systems that behave reliably and efficiently, are affordable to develop and maintain, and satisfy all the requirements that customers have defined for them. It is important because of the impact of large, expensive software systems and the role of software in safety-critical applications. It integrates significant mathematics, computer science and practices whose origins are in engineering. See `http://computingcareers.acm.org/?page_id=12`

Why do we need secure SE?

- Prevent loss/corruption of data
- Prevent unauthorized access to data
- Prevent unauthorized computation
- Prevent escalation of privileges
- Prevent downtime of resources

Note, this is not a SE class. We will not focus on waterfall, incremental, extreme, spiral, agile, or continuous integration/continuous delivery. Examples follow the traditional SE approach, leaving it up to you to generalize to your favorite SE approach.

Secure SE lifecycle

- Requirements/Specification
- Design
- Implementation
- Testing
- Updates and patching

Requirements/Specification

- Regular SE requirement specification plus
- Security specification
- Asset identification
- Assess environment
- Use cases and abuse cases

Design

- Regular SE design plus
- Threat modeling
- Security design
- Execution environment and actors
- Design review
- Design documentation (prose)

Implementation

- Regular SE implementation plus
- Source code repository/version control
- Coding standards (assertions, documentation)
- Source code review process

Testing

- Regular SE testing plus
- Security test plans
- Automatic testing
  - Fuzzing
  - Symbolic exceution
  - Formal verification
- Red team testing
- Continuous integration testing (Jenkins, Travis, etc)

Updates and patching

- Dedicated security response team
- Continuous process: new features, security issues
- Regression testing
- Secure update deployment

Summary

- Software lives and evolves
- Security must be first class citizen
  - Secure Requirements/specification
  - Security-aware Design (Threats?)
  - Secure Implementation (Reviews?)
  - Testing (Read team, fuzzing, unit)
  - Updates and patching
- Example: Ubuntu security features