

Mathias Payer

EPFL, Spring 2019

Mathias Payer CS412 Software Security

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

Seature	6.86 US (Exper Date)	6.18 (Erg) (P)	7.84 Existy Facet	7.18 (Datey Distant)	8.84 KPS (Harshy Hense)	K10 Stopic bod	K.M (Jacksinge)	8.30 (Karric Koda)	18.04 625 Guild Speci	16.10 (Maxerick Reefue)	11.64 Norty Karahal	11.39 Kineic Cealed	12,04 125 (Peciliar Pangoted	12.18 Oversel	SLM Staring Respuil	13.30 Disucy Salamander)	14.04 (75 (Tooly Tele)	14.10 GRopic Unicard	13.M NNI Verveti	13.30 IWiy Recently	36.64 LTS Contal Noral	18.18 (190077 160	17.M Qery Zopei	17.50 ShtTul Aarthrank)	38.64 UTS Illiante Bistorett
An Open Parks Openant's hashing	pains mat																								
O'N costion																									
Automatic security updates	age in	spt.in	age in	aptin	aptin	aptin	spt.in	age in	eptie.	aptin	spt.in	age.in	aption	opt-in.	optin	apt.in	ept-in	opt-in.	optin	spt.in	anabled				
Kernel Divegationes																					26-04 LTS Rented				
Mesysten Capabilities						Ramont & Longrador			harted & userspace		Local &	terrat & sompoor		harted &	kanad & userspace	terrat & startspace	Company & Compan	Contract & Contracts Contract	Same 5 Using a Ce Shebut	Lancal L scampace Storbuit	Lamai L Longate Iostaut M. Serveri	Lanval E Compace Contact In second	Contract & Contracts Contract	Lannal A scampace Stational at second	Lannal & Longaco Jostaut
Carifigurative Recent	later and the second	-	lytakin.	liplation .	-	-	-	-	-	-	-	-	-	-	-	-			-					-	
Clear PMG and 24 GT GCCCAP	-	-	-	-	-	-	-	-	-	-	-	-	-	-		-	patrone	petrose	patrices	patrona	patriate	potrate	patricks.	patrical a	patrona
AppAmor			-	23	Larval	karnal 2.3	Lannal 2.3	1.11	15	25.1	2.6.1	2.1 0-betal	1.7.8	2.8.8	2.8.0	Extend 2.8.0 (with 1000 x2 backword	larnal 2.835-3408 Galifi same vil	2.8.96	karnal 2.8.1	kernel 2.30	2.33.80 (1.13 (1.13)	111.00 (2.13 (2.13)	2.11.0	2.11.0	2.11.0
Witness				-	universe.	unicerse	universe	urivena.	universe	universe	universe	seivene	universe.	universe	universe	aniumas	Sectores universe	universe	unicerse	universe	arive a	universe	universe	universe	aniverse
SNICK						kamel .	kand	konel	hone	kamd	kampi ut	kenel	konel	horse	kamd	kampi	konst	hond	kamd	kampi	konel	honel	kond	kamd main	kanel man
Encryption GM	alt installer	alt installer	alt installer	alt installer	alt installer	installer -	ingater	installer	insplan	inepsitor a Thirde	instation of	alt installer	instation	installer	installer a Blocks	instation	ingolier	intoler	installer a Thirda	instation in	instation	intaler	intohr	installer a Thirda	ingster .
etrypth						-91488	Kr.	or	di	of m	Hin.	-Shivete er	er-,	of	of m	, Elements	-Oriente or	of	of m	H-Course	or	di	of m	of the second se	if n.
Turpet Rectam				karnel & uperspace	kenel & compace	Ratio &	Local &	bertal &	harted &	karled &	Local &	James 6	Sector &	harted &	Rating &	Invest &	hered &	harted &	karled &	Local &	Louised &	barted &	haved &	Arrest &	Local &
Rodule				Rame & conspora (gen-tools)	Epite-books	userspace Apre-	Pam.	esergeca Ppm-	Conspect	dawn.			eper-	dan-	dans.		Company Company	dau-	dom-	Exercised Fpm.	Eper-	Conspect	daw.	stangaos Apro-	Ppm.
Stuck Protector		BX BAD	411.049.3	811 840 P	900.0493	11.142	DO DADA	00000	000.0487	21.245	BX PACE	OC DWIN	\$10.040.N	000.0487	21.544	an away	011040.0	000.0487	21.242	BX BAD	00000	000.0403	212 246.7	21.242	DOC DALLY
Heap Polacial Danker	şinx	yac.	geo.	pac.																					
Obfunction Starte ASLN	-	-	-	-	geoc.	<u>.</u>									- C										
	Name 1305 setup																								
Ener ALLA Inte ALLA					Inernal Errore paticity Inernal Senar ASLAU																				
VESO ASLA					to tel																				
Balt as PR																									
Bull with Faility																									
But we stop						ges pairs.	ges paters																		
Bult with BIND_NOW								parkage ba												parkapa bai					
Non-Executable Referry	642 sets	BE only	FAE only	MC enty	And soly	NE on y	NC only																		
(procilipadoragia protocción	-	-	-	-	Inernal & egecti																				
Research Inc.					-	-	-	-	-																
Hardina (Marilina)																									
off and arrived																									
					towned to equilit	kannel & renetti	Lawrant	Level	barrel																
	larval				Inernal (over pation)																				
					barnel (new patient)																				
	014	849	414	040 CAP 515 18000.05	619			ward	word	nord.	nati	No.1	ward	wet	not	nyedi	word	word.	rot	nati	ward	word	Text	not	nyseli
Keeding Read-only data	CM-345_MODULIS	04/315/#00805	CM/345/MOBULES	04/315/4000025	CAP, SIS, MODULIS	-	in the second	And a second	-			and a	and a second	-	and a	lane a	-	-	1000	in the second	and a second	-	-		land.
and and Mark preletar																									
Module POINT							-	-	-	-															
Kannel Address Originy Restriction											tame	kanal	bernel	Harnel	karnel	kannel	hand	hernel	eared.	kannel	ternet.	her net	warned	tared.	tarna!
Kernel Address Space Layout Randomication											-	-	-	-	-	-	-	Norted (206 and amittle only(kanod (306-and ambt4 angt4 angt4	kamel 1386 and amitté anije	kanel 1365 and amitté anity)	COR and another services	konsel (306 and ambliti (angl)	amété anipi	kangt 1385 and amittal anity)
Blackfot Rare Protocols																									
Syscal Sharing drong																									
													sparit	vyuš	ryedi	epedi	wywill	wywell	eyedi	special	spara	spark	eyedi typ2	eyesti	special
Block texas: UDI Secure Boot													andia	andid	amélik,	ametid,	eye.2 amild,	eyu.2 amild,	eye.20 amétik,	Kyb30 ametik	spect and da	Exep andia	eyo.2 amild,	ryb30 amdili,	Kyte22 amolia
Genetial													failback	formed formation	kensel Selback	ternal failbach	forback	barred forback	hered Sriback	Lawall Salitaci	failback	Invest Difference	hand Sribax	ternal Millack	ternal fallback

Figure 1:

- Configuration: safe defaults
- Subsystems: enable flexibility
- Mandatory Access Control (MAC): compartmentalization
- Filesystem encryption: secure data at rest
- Trusted Platform Module: tamper proofing
- Userspace Hardening: mitigations
- Kernel Hardening: mitigations
- See https://wiki.ubuntu.com/Security/Features

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



Figure 2:

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

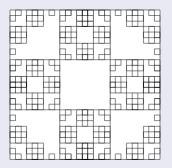


Figure 3:

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



Figure 4:

- Filesystem encryption: secure data at rest
 - Encrypted LVM;
 - eCryptfs



• Trusted Platform Module: tamper proof

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

- Userspace Hardening (2/2): mitigations
 - 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

- Kernel Hardening (1/2): *mitigations*
 - 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

- Kernel Hardening (2/2): mitigations
 - 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 | \dots } SE?



Figure 6:

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 on the slides follow a traditional SE approach, generalization to other SE approaches is left to the reader.

Secure SE lifecycle

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



Figure 7:

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 execution
 - 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



Figure 8: Source: https://www.wired.com/story/petya-plague-automatic-software-updates/

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