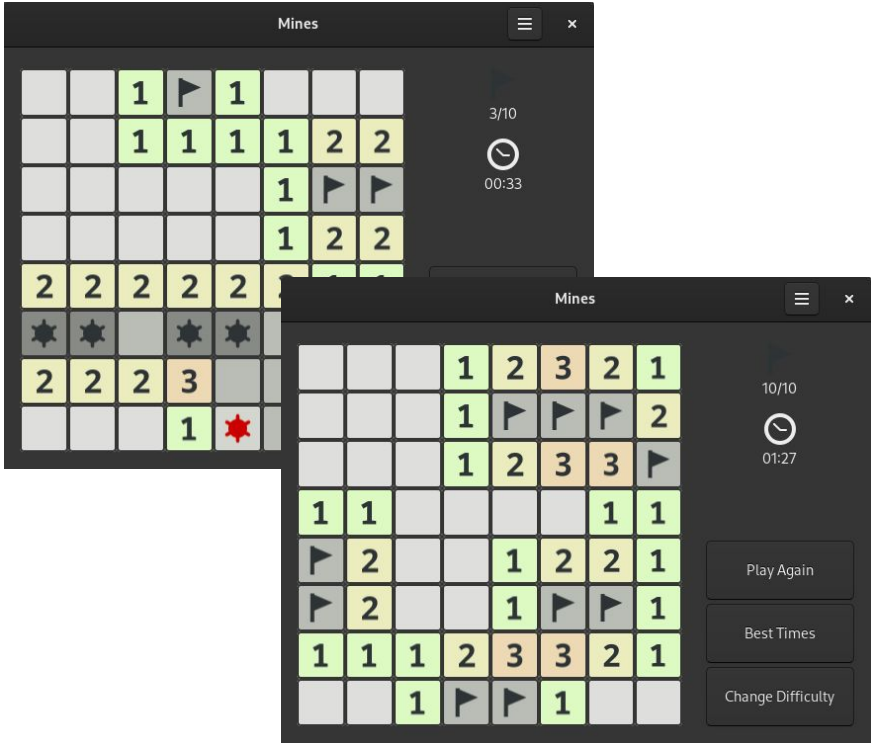
What the Fuzz, Security?

Mathias Payer
infosec.exchange/@gannimo



EPFL

hexhive

Cyberangriffe sind Alltag

Cyberangriff auf OneLog

Aus Tagesschau vom 25.10.2024.

News > Schweiz >

Anmeldung nicht mehr möglich

Hackerangriff verursacht Login-Probleme bei Schweizer Medien

Schneider Electric erneut gehackt

Von Christian Wingeier, 5. November 2024 um 16:53

SECURITY SCHNEI

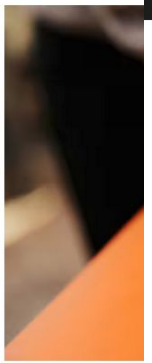


Foto: Bianca Ackerma

**Die Hackergrup
Daten von Schn
Baguettes gefo**

Der französische
Opfer eines Cybe
von der Ransomw
Managed-File-Tra
dann die Ransom

IT INSIDE IT

INSERIEREN JOBPORTAL EVENTS REPORTS

TECHNOLOGIE PARTNER

DEEPIMPACT

GOLD SPONSOREN

AUSZEICHNUNGEN

Award2023

WINNER 2024

Bacs: Zahl der gemeldeten Cybervorfälle hat sich verdoppelt

Von Keystone-sda / paz, 7.

SECURITY CYBERAN

Bacs-Direktor Florian S

**Rasant gestieger
der Installation v
waren hingegen**

Im ersten Halbjahr
34'789 Cybervorf

Stromanbieter Tibber gehackt, 50.000 deutsche Kunden betroffen

Tibber bestätigt, dass Hacker eingedrungen sind und Kundendaten an sich gebracht haben. Im Darknet werden diese nun verkauft.

203

Katholische Institutionen in St. Gallen von Cyberangriff getroffen

Von Keystone-sda / kjo, 28. Oktober 2024 um 12:33

SECURITY CYBERANGRIFF VERWALTUNG KANTON ST. GALLEN

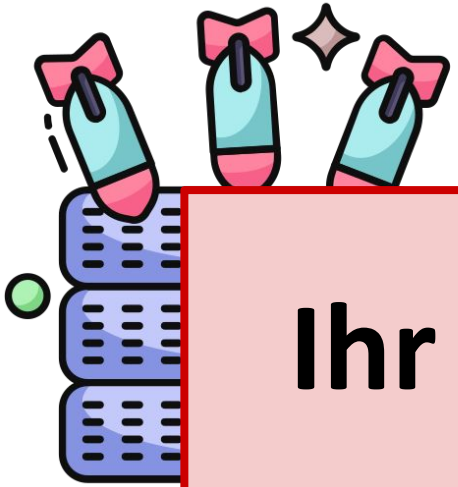
k St. Gallen

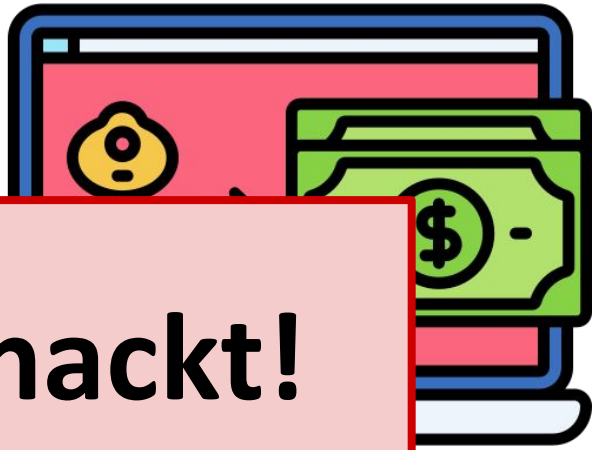
**öfliche Ordinariat wie auch die
Gallen sind nach einer Cyberattacke**

chenende sind kirchliche Institutionen im
rreichbar. Betroffen sind etwa die
ntons, die Stiftsbibliothek, das Bischöfliche
da und die Pensionskasse der Diözese St.

Wieso gibt es Angriffe?

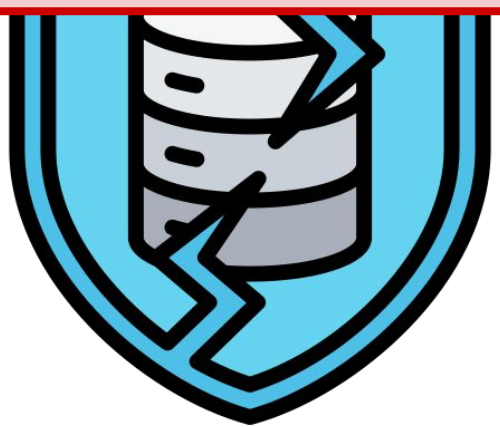
Ransomware





Ihr werdet alle gehackt!

Denial of Service

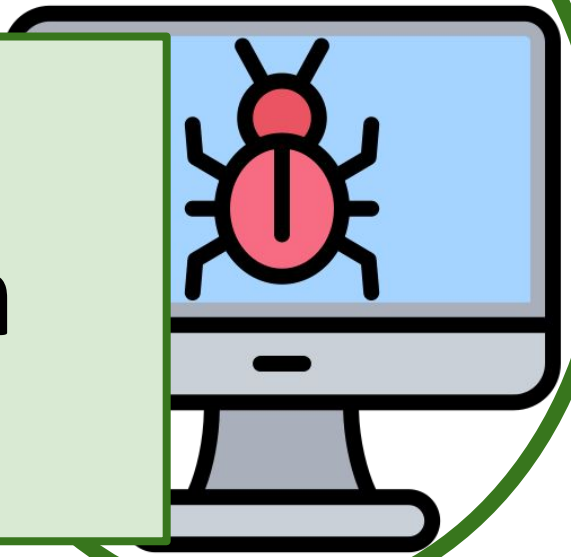


Spionage

Angriffsvektoren bei KMUs

Insiderangriff

Software Bugs



Software kann mit technischen Lösungen abgesichert werden

Schwache Passwörter

Fakt 1: Jede Software hat Bugs

Home / Innovation / Security

Microsoft: 70 percent of all security bugs are memory safety issues

Percentage of memory safety issues has been hovering at 70 percent for the past 12 years.

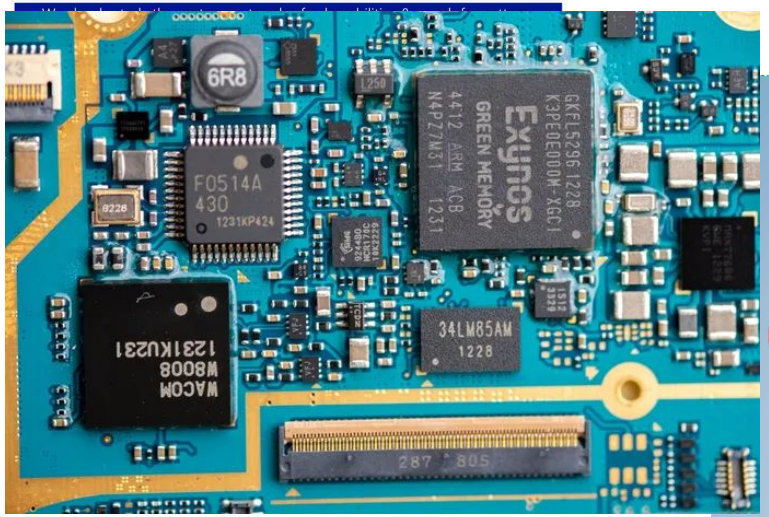
Written by **Catalin Cimpanu**, Contributor on Feb. 11, 2019

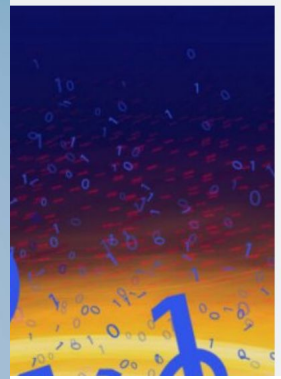
syzbot Linux

Open [1164] | Subsystems | Fixed [4420] | Invalid [9944] | Kernel Health | Bug Lifetimes | Fuzzing | Crashes

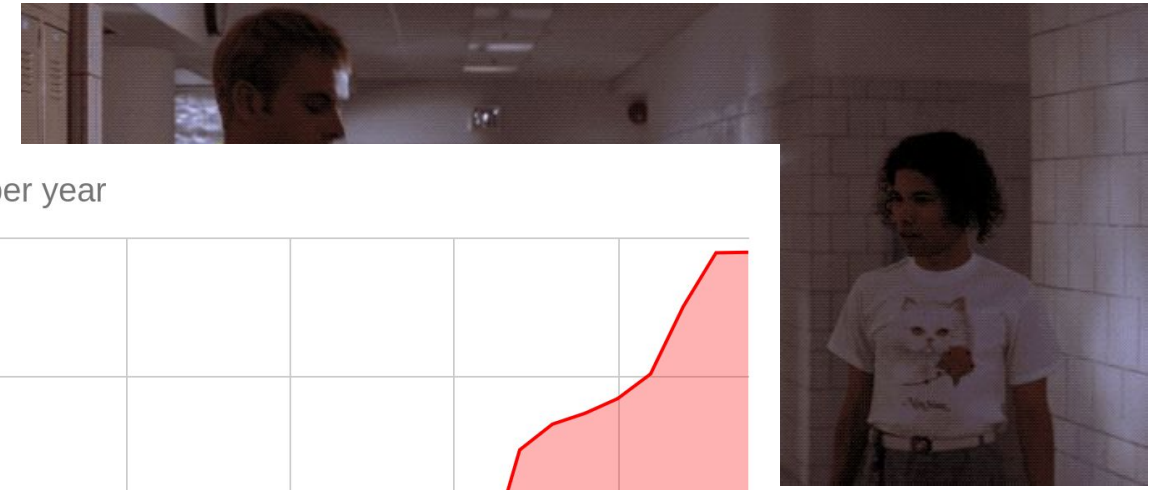
Instances [tested repos]:

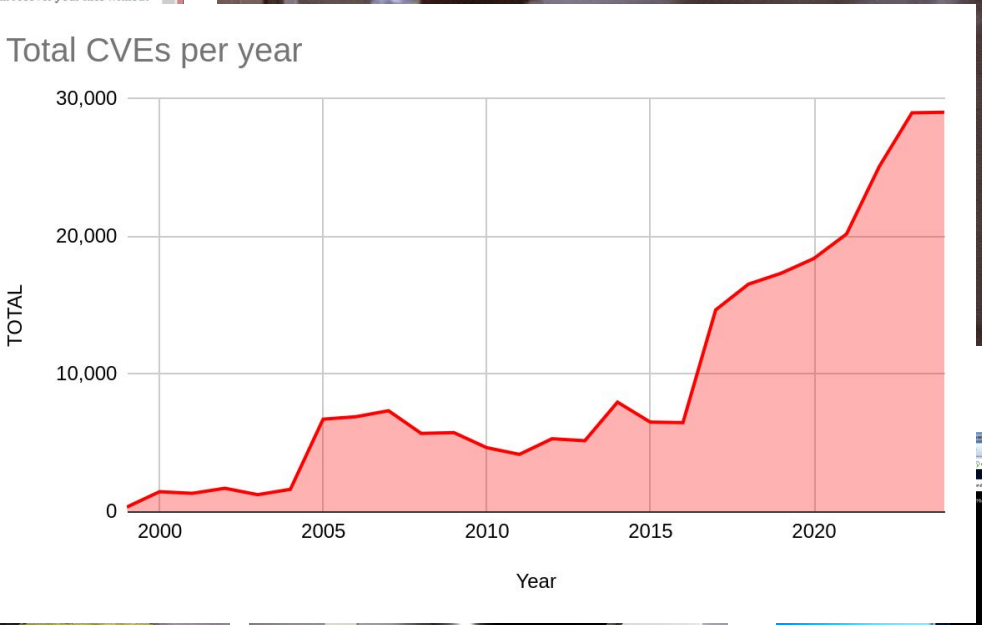
Name	Last active	Uptime	Corpus	Coverage	Crashes	Execs	Kernel build				syzkaller build		
							Commit	Config	Freshness	Status	Commit	Freshness	Status
ci-qemu-upstream	now	1h56m	41462	732887	474	1994555	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-qemu-upstream-386	now	1h30m	42407	690284	274	1794725	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-qemu2-arm32	now	2h01m	106687	124893	3	963936	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-qemu2-arm64	now	2h04m	85850	99332	1	667485	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-qemu2-arm64-compat	now	2h12m	86053	97941	2	559463	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-qemu2-arm64-mte	now	2h24m	103714	120925	8	1085595	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-qemu2-riscv64	now	4d08h	16409	309333	284	121436	950b879b7f02	.config	52d	failing	f325deb0	4d09h	
ci-upstream-bpf-kasan-gce	now	4d08h	24447	398884	5	1654017	a6f6a95f2580	.config	5d12h		f325deb0	4d09h	
ci-upstream-bpf-next-kasan-gce	now	47m	17741	378676	83	2134894	92b2e810f0d3	.config	4h57m		f325deb0	4d09h	
ci-upstream-gce-arm64	now	4d08h	90240	659623	420	11956112	59caa87f9dfb	.config	4d23h		f325deb0	4d09h	
ci-upstream-gce-leak	now	1h18m	51134	1092207	21	1534040	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-upstream-kasan-gce	now	1h48m	40061	642084	38	3993352	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-upstream-kasan-gce-386	now	1h35m	34586	565005	22	1731109	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-upstream-kasan-gce-root	now	1h23m	55681	992930	59	3603938	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-upstream-kasan-gce-selinux-root	now	2h00m	52242	1020377	37	3134948	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-upstream-kasan-gce-smack-root	now	1h05m	75933	819911	51	4273148	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci-upstream-kmsan-gce	now	4d08h	65142	420992	110	2532857	90ea0df61c98	.config	13d		f325deb0	4d09h	
ci-upstream-kmsan-gce-386	now	4d08h	62085	457804	129	731475	90ea0df61c98	.config	13d		f325deb0	4d09h	
ci-upstream-linux-next-kasan-gce-root	now	2d15h	73330	1121999	36	3850191	4b0f4525dc4f	.config	2d16h		f325deb0	4d09h	
ci-upstream-net-kasan-gce	now	7h38m	33528	426604	45	5225325	ceb29474bbbc	.config	8h21m		f325deb0	4d09h	
ci-upstream-net-this-kasan-gce	now	7h15m	32020	417595	37	3244102	e669ce46740a	.config	8h07m		f325deb0	4d09h	
ci2-upstream-fs	now	1h10m	20516	191506	261	6336096	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci2-upstream-kcsan-gce	now	1h59m	58089	397348	97	5180957	6ab608fe852b	.config	2h44m		f325deb0	4d09h	
ci2-upstream-usb	now	3d21h	1980	51037	1099	935583	d629c0e221cd	.config	4d12h		f325deb0	4d09h	

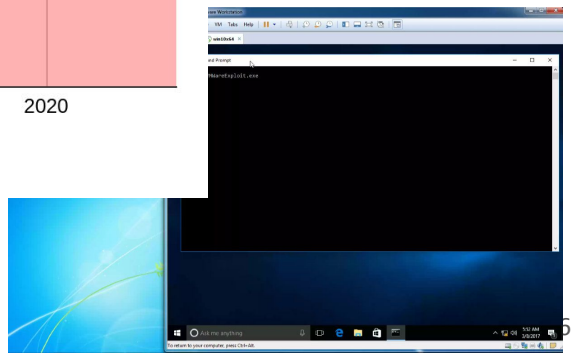




Fakt 2: Viele Bugs können ausgenutzt werden







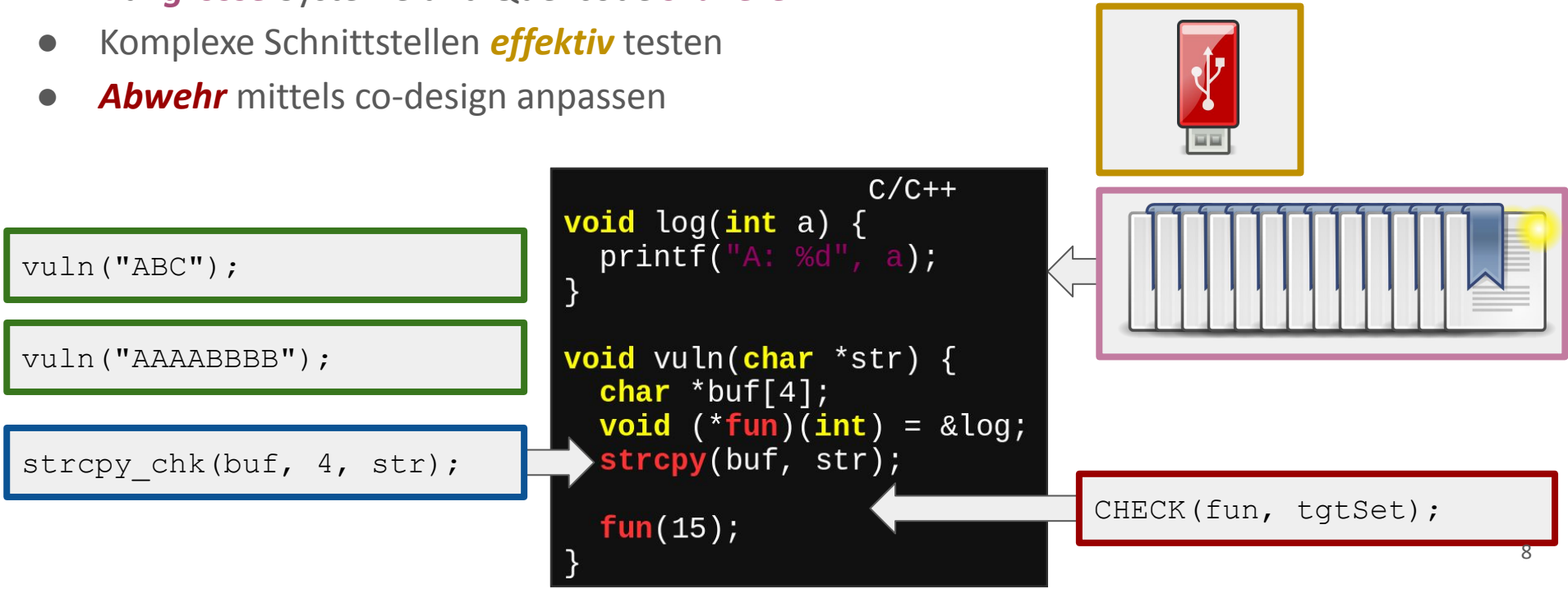
Fakt 3: Software ist unglaublich komplex

Google Chrome:	76 MLoC
Gnome:	9 MLoC
Xorg/Wayland:	1 MLoC
glibc:	2 MLoC
Linux kernel:	17 MLoC

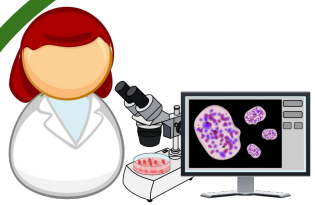
Margaret Hamilton mit dem Quellcode des Apollo Guidance Computers (NASA, ‘69)

Cyber-Security: Was sind die Schlüsselfragen?

- Schwachstellen ***effizient*** erkennen
- Tests ***automatisch*** generieren
- Auf ***grosse*** Systeme und Quellcode ***skalieren***
- Komplexe Schnittstellen ***effektiv*** testen
- ***Abwehr*** mittels co-design anpassen



hexhive



Software Testen

- Ziel: Fehler finden
- Developer-orientiert

Mitigation/Abwehr

- Ziel: Exploitation verhindern
- Schützt das Endgerät

Kompartimente

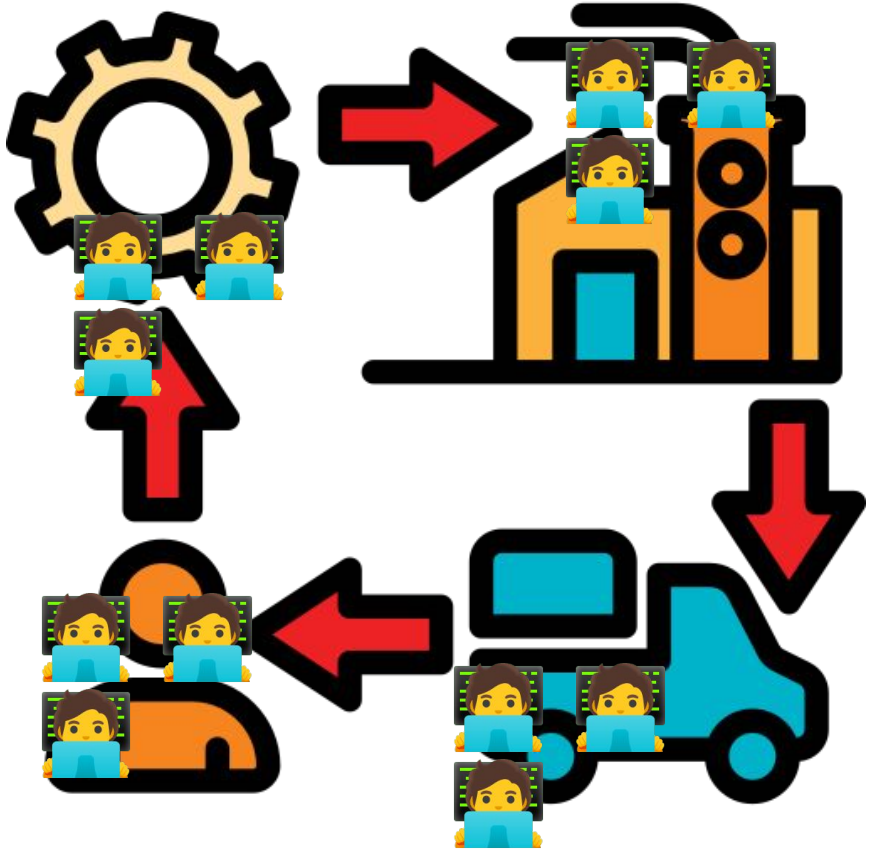
- Ziel: Segmentierung
- Mehrschichtige Sicherheit

Wo ist Software überall im Einsatz?

Software ist allgegenwärtig

Software Supply Chain (und Angriffe)

Quellcode/
Bibliotheken



Entwickler und
Systeme

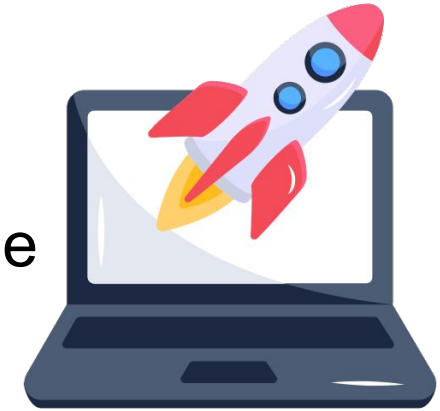
Installierte
Systeme

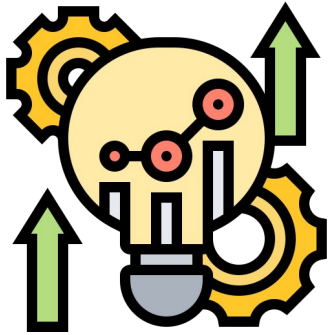
Markt (“MS Apps”)

Sicherheit verschmilzt mit DevOps

Automatisiertes Testen

Ausliefern und Pflege





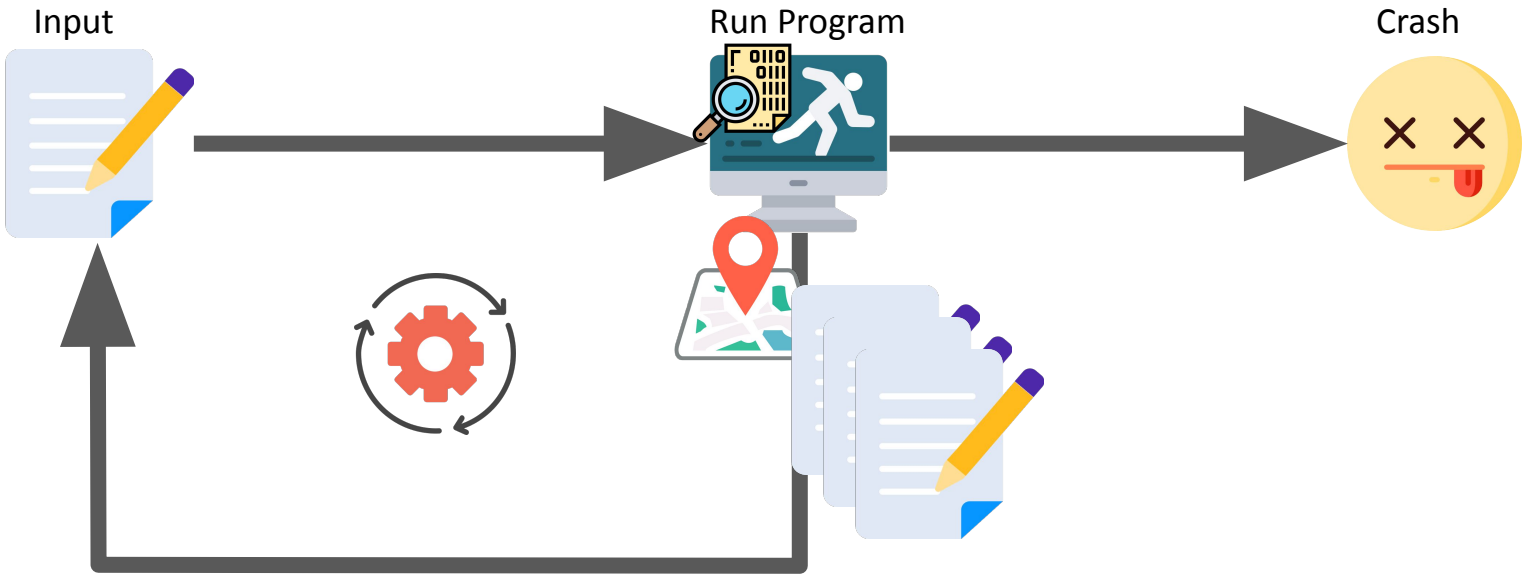
Kontinuierliche Integration

Softwareentwicklung

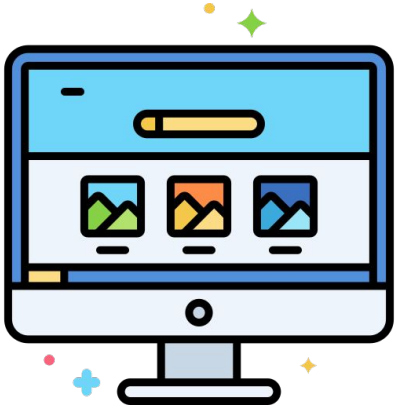
Fuzzing in a Nutshell

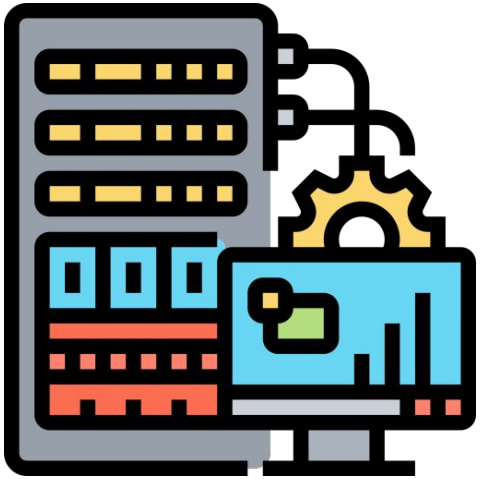
```
$ ./testme --help
Usage: testme <int32_arg>

$ ./testme AAAA
Please enter an integer!

$ cat fuzzer.sh
while :
do
  len=$(($RANDOM % 255))
  input="$(dd if=/dev/urandom bs=$len count=1)"
  ./testme $input || echo $input >> crash_seeds
done
```

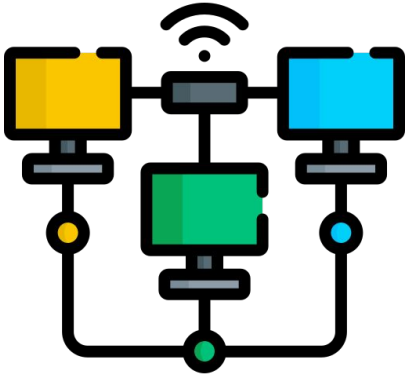
Fuzzing: Automatisches (Fuzz) Testing

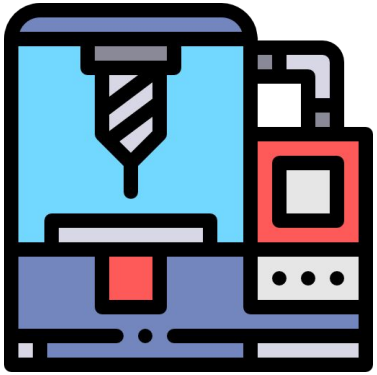


Fuzzing aus unserer Forschung









Effektive Cyber-Sicherheit in Unternehmen

Sicherheit ist ***planbar***!

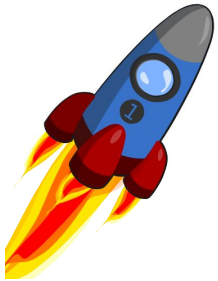
- Dokumentierte Angriffsvektoren und Strategien

Software muss ***getestet*** und ***dokumentiert*** sein

- SBOM, Supply Chain und Fuzzing als Buzzwords
  die in jedem Unternehmen ein Thema sein sollten

Cyber-Ressourcen sind beschränkt

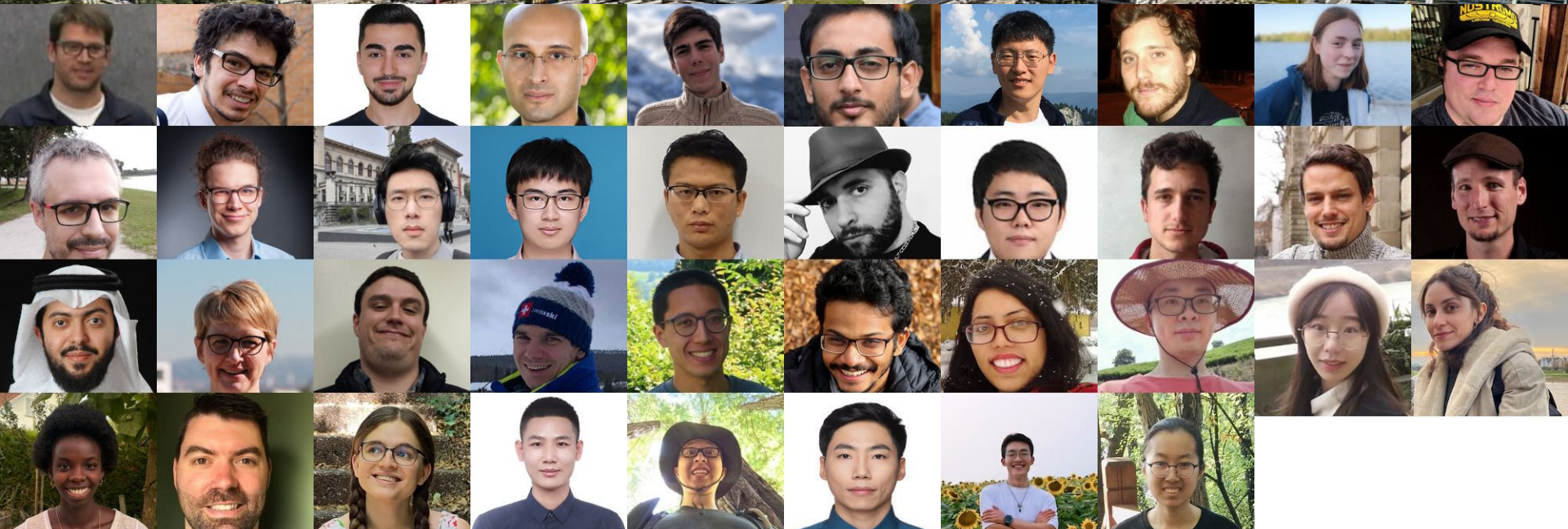
- Analysten brauchen Zeit für Strategien



hexhive

EPFL

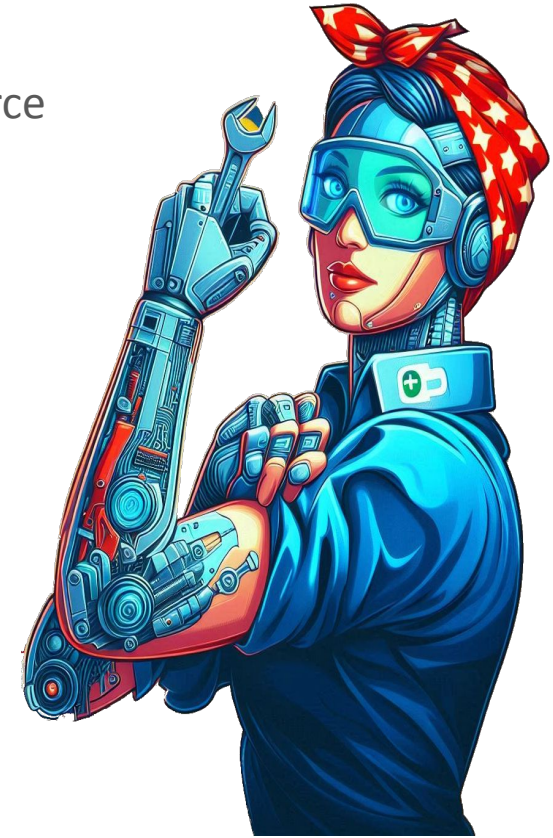
Join us on this research journey!



Supply Chain und Software Sicherheit: We can do it!

Software Bugs sind allgegenwärtig und eine nachwachsende Ressource

- Firmen brauchen Strategien für die Cyber-Sicherheit
- SecDevOps integriert Sicherheit in den Entwicklungsprozess
- Testing erhöht Softwarequalität und Sicherheit
- Kompartmentalisierung verringert die restliche Angriffsfläche



Es braucht Forschung *und* die Industrie zusammen

- Academia entwickelt neue Prozesse: Testing/Kompartimente
- Industrie liefert Daten und Praxis: Prozesse und Anwendung

EPFL

Mathias Payer (infosec.exchange/@gannimo)

hexhive