

Systematic Analysis of Kernel Security Performance and Energy Costs

Fabian Rauscher, Benedict Herzog, Timo Hönig, and Daniel Gruss

29.08.2025

> isec.tugraz.at







- Security is important



- Security is important
- Mitigating dangerous vulnerabilities is important



- Security is important
- Mitigating dangerous vulnerabilities is important
- Performance is important



- Security is important
- Mitigating dangerous vulnerabilities is important
- Performance is important
- When mitigating vulnerabilities performance is optimized



- Security is important
- Mitigating dangerous vulnerabilities is important
- Performance is important
- When mitigating vulnerabilities performance is optimized
- What about energy?







- Performance = Energy ...



- Performance = Energy ...
- ... kind of



- Performance = Energy ...
- ... kind of
- CPUs are a lot more complex than 30 years ago



- Performance = Energy ...
- ... kind of
- CPUs are a lot more complex than 30 years ago
 - out-of-order execution



- Performance = Energy ...
- ... kind of
- CPUs are a lot more complex than 30 years ago
 - out-of-order execution
 - branch prediction



- Performance = Energy ...
- ... kind of
- CPUs are a lot more complex than 30 years ago
 - out-of-order execution
 - branch prediction
 - speculation



- Performance = Energy ...
- ... kind of
- CPUs are a lot more complex than 30 years ago
 - out-of-order execution
 - branch prediction
 - speculation
 - ...







- Rebooting the system after the benchmarks for every CVE is slow



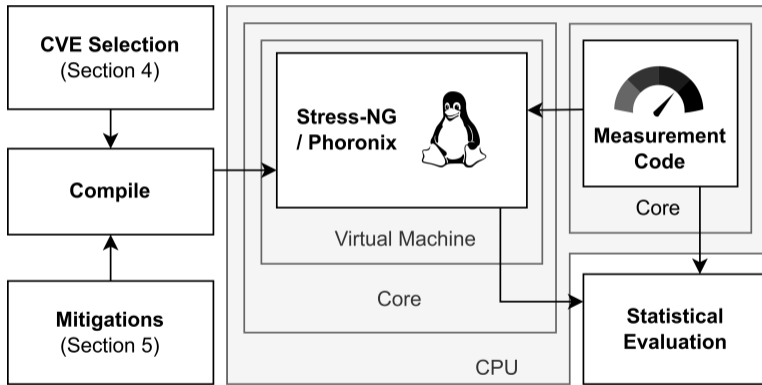
- Rebooting the system after the benchmarks for every CVE is slow
- Cloud scenario



- Rebooting the system after the benchmarks for every CVE is slow
- Cloud scenario
 - Tested Kernel runs in a VM



- Rebooting the system after the benchmarks for every CVE is slow
- Cloud scenario
 - Tested Kernel runs in a VM
 - Measurement code runs on the host









- Microbenchmarks using Stress-NG



- Microbenchmarks using Stress-NG
- Macrobenchmarks using Phoronix







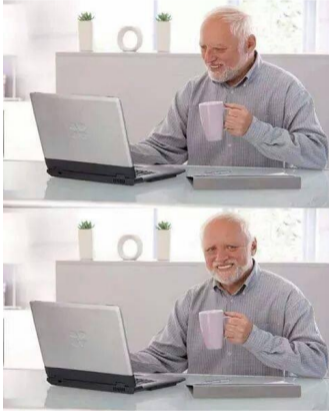
- We test all CVE patches since Linux 4.0

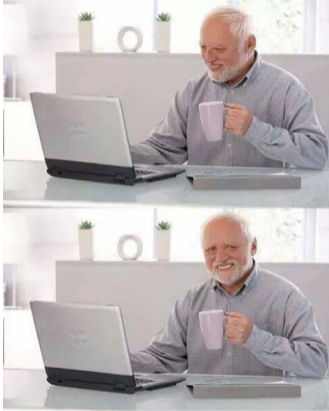


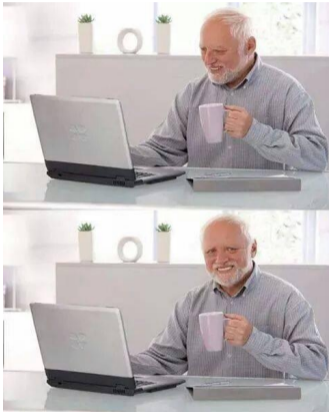
- We test all CVE patches since Linux 4.0
→ 1616 CVEs



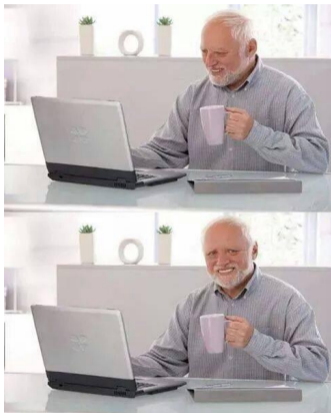
- We test all CVE patches since Linux 4.0
 - 1616 CVEs
 - 3232 Kernels



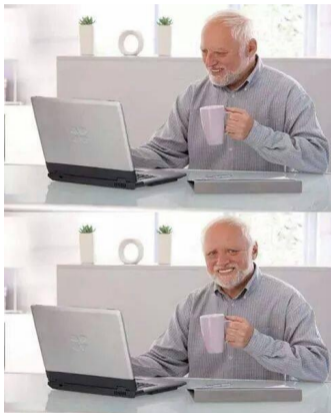




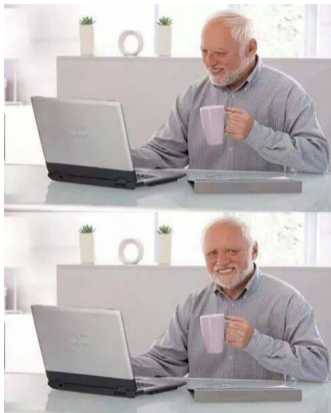
- Running all benchmarks on one kernel: ≈ 6 min



- Running all benchmarks on one kernel: ≈ 6 min
- Running all benchmarks on all kernels once: ≈ 14 days



- Running all benchmarks on one kernel: ≈ 6 min
- Running all benchmarks on all kernels once: ≈ 14 days
- 100 samples ≈ 3.7 years



- Running all benchmarks on one kernel: ≈ 6 min
 - Running all benchmarks on all kernels once: ≈ 14 days
 - 100 samples ≈ 3.7 years
- We need to prefilter







- 1 Build all Kernels with debug information



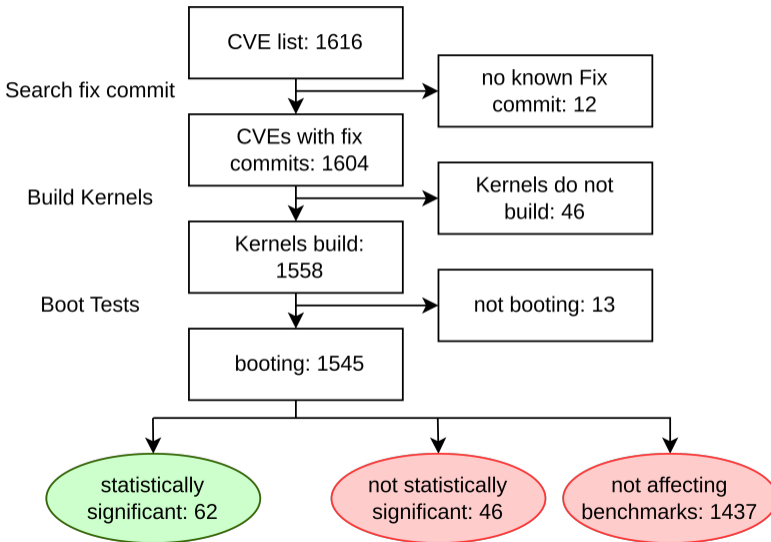
- 1 Build all Kernels with debug information
- 2 Set a breakpoint on every line that changed in the patch

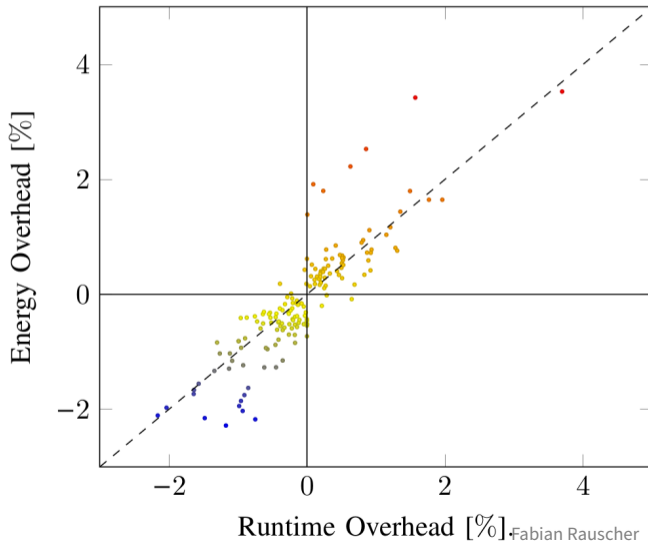


- 1 Build all Kernels with debug information
- 2 Set a breakpoint on every line that changed in the patch
- 3 Run all benchmarks



- 1 Build all Kernels with debug information
- 2 Set a breakpoint on every line that changed in the patch
- 3 Run all benchmarks
- 4 Throw out CVEs and specific benchmarks for some CVEs where no breakpoint is hit











- CVE-2017-1000112



- CVE-2017-1000112
 - Adds new checks in the UDP implementation



- CVE-2017-1000112
 - Adds new checks in the UDP implementation
 - UDP benchmark: Runtime +1.5%, Energy consumption +1.4%



- CVE-2017-1000112
 - Adds new checks in the UDP implementation
 - UDP benchmark: Runtime +1.5%, Energy consumption +1.4%
- CVE-2020-29534



- CVE-2017-1000112
 - Adds new checks in the UDP implementation
 - UDP benchmark: Runtime +1.5%, Energy consumption +1.4%
- CVE-2020-29534
 - Pipe reference count issue



- CVE-2017-1000112
 - Adds new checks in the UDP implementation
 - UDP benchmark: Runtime +1.5%, Energy consumption +1.4%
- CVE-2020-29534
 - Pipe reference count issue
 - Pipe benchmark: Runtime -0.6%, Energy consumption -0.9%



- CVE-2017-1000112
 - Adds new checks in the UDP implementation
 - UDP benchmark: Runtime +1.5%, Energy consumption +1.4%
- CVE-2020-29534
 - Pipe reference count issue
 - Pipe benchmark: Runtime -0.6%, Energy consumption -0.9%
 - Reason: less branch misses (-2.1%)

Mitigations







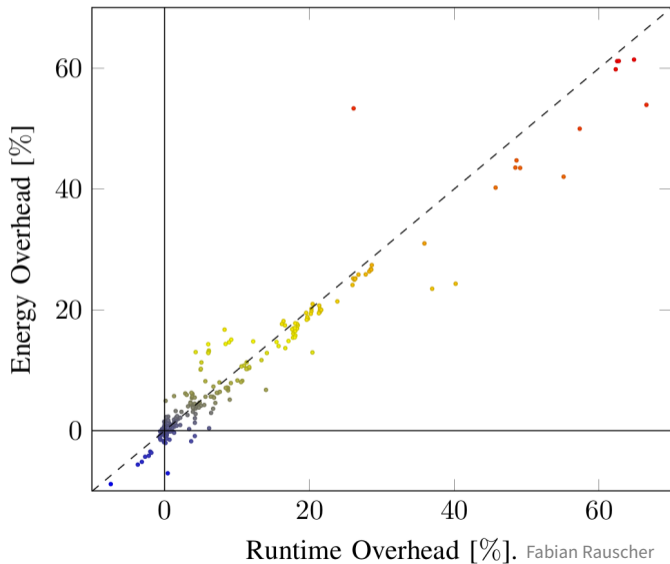
- 2018: first software mitigations against CPU bugs with Meltdown and Spectre

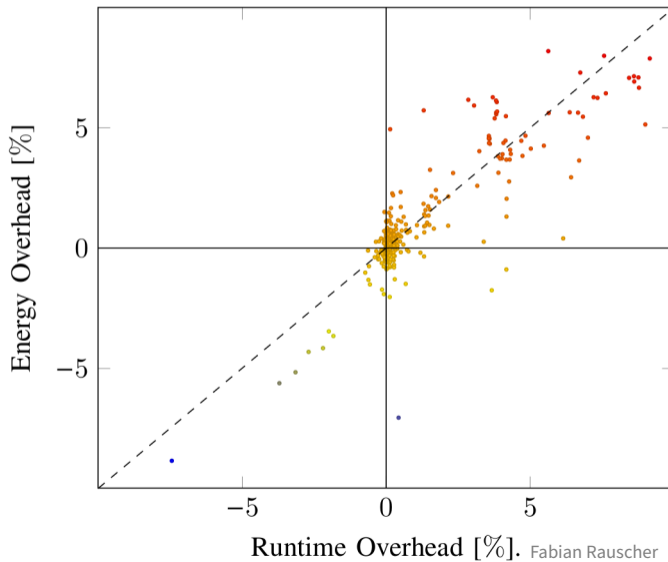


- 2018: first software mitigations against CPU bugs with Meltdown and Spectre
- Can be turned on or off through the Kernel command line options



- 2018: first software mitigations against CPU bugs with Meltdown and Spectre
- Can be turned on or off through the Kernel command line options
- By now we have 20 mitigations for x86 alone





Mitigation Results (Runtime)

Benchmark

icache	4.3	4.3	*	3.9	4	4.3	71	4.7	*	*	17	*	*	16	16	6.4	238	16	*	*
fork	1.8	1.3	0.7	1.3	1.5	1.4	35	11	2.2	*	6.8	*	*	5	4.8	5.6	66	4.7	*	*
pthread	-7.4	*	*	*	*	*	36	*	-41	*	*	5.5	*	23	27	-9.9	379	25	*	*
context	3.6	3.6	4.3	3.6	3.6	3.6	322	3.6	0.7	*	87	*	*	96	96	5.6	972	96	*	*
pipe	21	21	13	21	21	21	81	55	1.5	*	19	0.5	*	19	20	10	261	19	*	*
io	8.8	8.8	7	10	8.6	10	40	8.6	13	*	6.7	*	*	7.3	6.4	7.6	75	9	*	*
sock	17	17	10	17	17	18	177	16	*	*	45	*	*	48	48	7.2	536	49	*	*
udp	26	26	14	26	26	26	109	313	1.4	*	15	*	*	18	18	15	347	17	*	*
futex	11	11	9.1	11	11	11	112	174	64	*	20	*	*	9.2	8.9	6.7	214	8.4	*	*
aio	17	18	8.4	17	17	18	354	18	1.1	*	84	*	*	94	94	19	1,147	94	*	*
switch	28	28	17	28	28	28	125	404	1.4	*	21	*	*	20	20	10	387	20	*	*
sctp	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	78	*	*	*
signal	4.1	4	3.2	4.2	4	3.9	233	4.1	1.3	*	57	*	*	62	62	7.6	734	62	*	*
cpu	*	0.1	*	0.1	*	*	1.7	0.1	6.7	*	*	*	*	0.1	*	*	3.2	*	-0.2	*
net.-loop.	6.1	6	2.8	6.1	6	6	26	55	0.3	*	8.3	*	*	5.1	5	3.1	89	5	*	*
mutex	-0.1	-0.1	-0.1	-0.1	-0.1	0.1	0.5	-0.1	-0.1	*	0.2	*	*	0.2	-0.1	0.2	1.5	0.2	*	*
osb files	3.8	3.8	2.3	3.8	3.8	3.9	4.2	3.8	0.4	*	1.5	*	*	1.6	1.7	3.7	12	1.5	*	*
osb processes	0.1	0.1	0	0.1	0.1	0.1	1.3	0.2	0.1	*	0.3	*	*	0.3	0.3	0.2	4.2	0.3	*	*
osb threads	0.1	0.1	0.1	0.1	0.1	0.1	1.3	0.2	0.1	*	0.3	*	*	0.3	0.3	0.2	4.2	0.3	*	*
osb programs	0.1	0.1	0.1	0.2	0.1	0.1	1.4	0.2	0.1	*	0.3	*	*	0.3	0.3	0.2	4.2	0.3	*	*
osb allocations	0.2	0.2	0.1	0.2	0.2	0.2	1.3	0.3	0.1	*	0.2	*	*	0.2	0.2	0.3	4.3	0.2	*	*
apache	0	0	*	0	0	0	0.1	0	*	*	0	*	*	0	0	0	0.4	0	*	*
pmbench	0	0	0	0	0	0	0.2	0	0	*	0	*	*	0	0.1	0	0.7	0	*	*

spectre_v2=retpoline
 spectre_v2=retpoline,generic
 spectre_v2=retpoline,lfence
 spectre_v2=eibrs
 spectre_v2=eibrs,retpoline
 spectre_v2=eibrs,lfence
 spectre_v2=ibrs
 spectre_v2=on
 spec_store_bypass_disable=on
 spectre_v1
 pti=on
 l1tf=flush
 l1tf=full
 mds=full
 tsx_async_abort=full
 retbleed=full
 retbleed=unret
 mmio_stale_data=ibpb
 l1d_flush=full
 kvm.nx_huge_pages=force

Mitigation

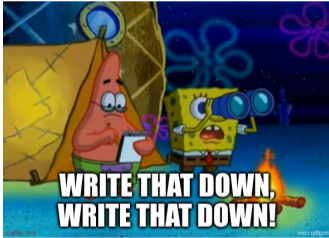
Mitigation Results (Energy)

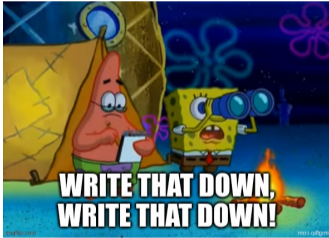
Benchmark

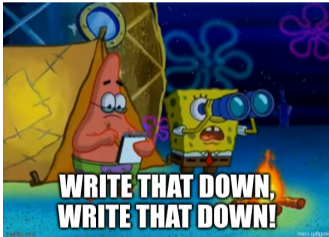
icache	3.7	4.1	*	3.1	3.7	3.9	64	4.5	*	*	15	*	*	18	17	5.6	211	17	*	*		
fork	1.9	1.4	*	0.7	1	1.1	30	10	2.1	*	5.5	*	*	4.1	4.7	5.6	53	3.8	*	*		
pthread	-8.8	*	*	*	*	*	23	*	-46	*	*	4.3	*	21	25	-10	313	24	*	*		
context	4.6	4.3	2.8	4.7	4.5	4.4	289	4.6	0.8	*	79	*	*	117	118	8.2	841	118	*	*		
pipe	19	19	11	19	20	19	69	42	1.7	*	18	0.7	0.5	18	19	10	209	18	*	*		
io	6.7	7.1	4.6	8.2	6.9	8.3	24	7.1	6.8	*	3.6	*	*	6.2	2.9	6.4	41	5.1	*	*		
sock	14	15	8	16	16	15	151	13	1.3	*	40	*	*	44	43	6.3	425	43	*	*		
udp	25	25	12	25	24	25	98	269	1.6	*	13	*	*	17	17	14	282	16	*	*		
futex	10	10	7.9	10	11	10	87	149	61	*	12	*	*	15	14	7.3	169	14	*	*		
aio	16	17	7.1	16	16	16	308	16	1	*	79	*	*	92	91	19	959	92	*	*		
switch	27	26	16	26	26	26	108	332	1.3	*	20	*	*	19	20	10	316	20	*	*		
sctp	*	*	*	*	*	*	0.4	*	*	*	*	*	*	*	*	*	73	*	*	*	*	
signal	4.4	3.9	2.6	3.7	3.7	3.8	192	4.5	1.8	*	49	*	*	61	61	8	600	59	*	*		
cpu	*	*	*	*	*	*	2.1	0.2	5.6	*	0.4	*	*	*	*	*	4	0.2	*	*	*	
net.-loop.	13	14	6.2	13	12	12	53	109	0.9	*	16	*	*	11	10	5.9	166	10	*	*		
mutex	*	*	*	*	*	*	*	*	1.5	*	*	0.3	-0.1	*	0.2	*	1.3	*	*	*	*	
osb files	6.1	5.4	3.1	5.6	5.6	5.7	5.5	6.1	*	*	1.6	*	*	2.2	2.4	6.3	14	3.2	*	*		
osb processes	0.8	0.3	*	0.3	0.4	0.3	0.9	*	*	*	*	*	*	*	0.2	0.7	2	*	*	*		
osb threads	*	*	-0.8	*	*	*	-0.4	*	1.1	*	-0.5	*	*	*	*	1.3	-0.9	*	*	*	*	
osb programs	0.4	*	*	0.2	*	*	0.9	*	0	*	*	*	*	0.7	0	0.2	1.3	0.1	*	*	*	
osb allocations	0.2	*	*	0.4	0.3	0.2	5.7	0.6	4.9	*	1.7	*	*	2.3	1	0.5	13	2.2	*	*	*	
apache	0.5	0.4	0.7	0.4	0.6	0.5	-2	-0.6	*	*	*	*	*	0.8	1.1	0.4	-7.1	1.1	*	*	*	
pmbench	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0.6	*	*	*	*	*

spectre_v2=retpoline generic
 spectre_v2=retpoline,lfence
 spectre_v2=eibrs,retpoline
 spectre_v2=eibrs,lfence
 spectre_v2=eibrs
 spectre_v2=on
 spec_store_bypass_disable=on
 spectre_v1
 pti=on
 l1tf=flush
 l1tf=full
 mds=full
 tsx_async_abort=full
 retbleed=unret
 retbleed=ibpb
 mmio_stale_data=full
 l1d_flush=on
 kvm.nx_huge_pages=force

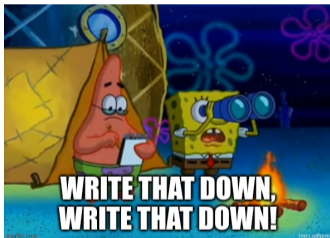
Mitigation



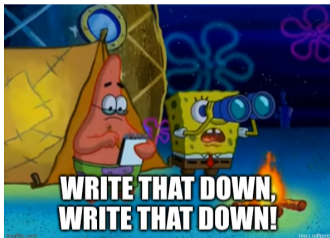




- for large overheads performance and energy consumption correlate (relatively) well



- for large overheads performance and energy consumption correlate (relatively) well
- for small overheads they can differ by a lot



- for large overheads performance and energy consumption correlate (relatively) well
- for small overheads they can differ by a lot
- use energy consumption as an extra metric





We



We

- ... made the first large scale analysis of energy and performance of Linux security patches and mitigation



We

- ... made the first large scale analysis of energy and performance of Linux security patches and mitigation
- ... created a method for efficient benchmarking of patches



We

- ... made the first large scale analysis of energy and performance of Linux security patches and mitigation
- ... created a method for efficient benchmarking of patches
- ... analyzed 1616 CVEs, covering an 8 year time frame



We

- ... made the first large scale analysis of energy and performance of Linux security patches and mitigation
- ... created a method for efficient benchmarking of patches
- ... analyzed 1616 CVEs, covering an 8 year time frame
- ... analyzed all available Linux mitigations



We

- ... made the first large scale analysis of energy and performance of Linux security patches and mitigation
- ... created a method for efficient benchmarking of patches
- ... analyzed 1616 CVEs, covering an 8 year time frame
- ... analyzed all available Linux mitigations
- ... collect performance counter data of interesting cases and discuss them in detail

This research was made possible by generous funding from:



Funded by
the European Union



European Research Council
Established by the European Commission



Der Wissenschaftsfonds.

Deutsche
Forschungsgemeinschaft



Federal Ministry
of Education
and Research



Red Hat



This research is supported in part by the European Research Council (ERC project FSsec 101076409), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 539710462 (“DOSS”), 502228341 (“Memento”) and 465958100 (“NEON”), the Bundesministerium für Bildung und Forschung (BMBF, Federal Ministry of Education and Research) in Germany (project SUSTAINET-inNOvAte 16KIS2262), and the Austrian Science Fund (FWF project NeRAM 10.55776/16054). Additional funding was provided by generous gifts from Red Hat, and Intel. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding parties.



Systematic Analysis of Kernel Security Performance and Energy Costs

Fabian Rauscher, Benedict Herzog, Timo Hönig, and Daniel Gruss

29.08.2025

> isec.tugraz.at