Memory Vulnerabilities in Memory-safe Languages

Veit Heller

Information Security Meetup Berlin, August 2020

July 22, 2020

Python

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2008	1			1											
2010	7	5		5	1										
2011	2	1									2				
2012	5	3			1		1				1				
2013	2	1													
2014	6	2	1	2						1	1				1
2015	1											1			
2016	5			1						1	1				
2017	3		1	2											
2018	8	5	2	2	1										
2019	9						1			1					
Total	49	17	4	13	3		2			3	5	1			1
% Of All		34.7	8.2	26.5	6.1	0.0	4.1	0.0	0.0	6.1	10.2	2.0	0.0	0.0	

Space Leaks

“Pinpointing space leaks is a skill that takes practice and perseverance. Better tools could significantly simplify the process.“
— Mitchell, Neil: Leaking Space. Eliminating memory hogs.

References

- These slides: https://github.com/hellerve/talks
- Go bug 20135: https://github.com/golang/go/issues/20135
- Breaking Erlang Maps: https://medium.com/@jlouis666/breaking-erlang-
- RustBelt: https://plv.mpi-sws.org/rustbelt
- Space leak: A Haskell Sore Spot: https://fremissant.net/leaky
- Auditing popular Rust crates: how a one-line unsafe has nearly ruined everything: https://medium.com/@shnatsel/auditing-popular-
- Xu, Hui et al.: Memory-Safety Challenge Considered Solved? An In-Depth Experience Report with All Rust CVEs

Thank you!

Questions?