Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 108–122.

[5] Weidong Cui, Marcus Peinado, Sang Kil Cha, Yanick Fratantonio, and Vasileios P Kemerlis. 2016. RETracer: triaging crashes by reverse execution from partial memory dumps. In *Proceedings of the 38th International Conference on Software Engineering*. ACM, 820–831.

[6] Yingnong Dang, Rongxin Wu, Hongyu Zhang, Dongmei Zhang, and Peter Nobel. 2012. ReBucket: a method for clustering duplicate crash reports based on call stack similarity. In *Software Engineering (ICSE), 2012 34th International Conference on*. IEEE, 1084–1093.

[7] CVE Details. 2019. Vulnerability Statistics Of Chrome. https://www.cvedetails.com/product/15031/Google-Chrome.html?vendor_id=1224.

[8] Jonathan Foote. 2018. GDB 'exploitable' plugin. https://github.com/jfoote/exploitable.

[9] Michael Gegick, Pete Rotella, and Tao Xie. 2010. Identifying security bug reports via text mining: An industrial case study. In *Mining software repositories (MSR), 2010 7th IEEE working conference on*. IEEE, 11–20.

[10] Google. 2016. Chromium Issue 388665. https://bugs.chromium.org/p/chromium/issues/detail?id=388665.

[11] Google. 2016. Chromium Issue 595834. https://bugs.chromium.org/p/chromium/issues/detail?id=595834.

[12] Google. 2018. Chromium Issue 386988. https://bugs.chromium.org/p/chromium/issues/detail?id=386988.

[13] Google. 2018. Chromium Issue 610600. https://bugs.chromium.org/p/chromium/issues/detail?id=610600.

[14] Google. 2018. Chromium Issue 718858. https://bugs.chromium.org/p/chromium/issues/detail?id=718858.

[15] Google. 2018. Chromium Issue 729991. https://bugs.chromium.org/p/chromium/issues/detail?id=729991.

[16] Google. 2018. Chromium Issue 733549. https://bugs.chromium.org/p/chromium/issues/detail?id=733549.

[17] Google. 2018. Reporting Security Bugs. https://dev.chromium.org/Home/chromium-security/reporting-security-bugs.

[18] Google. 2019. Chrome Reward Program Rules. https://www.google.com/about/appsecurity/chrome-rewards/.

[19] Google. 2019. Chromium Issues. https://bugs.chromium.org/p/chromium/.

[20] Google. 2019. ClusterFuzz. https://github.com/google/clusterfuzz.

[21] Google. 2019. ClusterFuzz Crash type. https://google.github.io/clusterfuzz/reference/glossary.

[22] Google. 2019. Google V8. https://chromium.googlesource.com/v8/v8/.

[23] Gustavo Grieco, Guillermo Luis Grinblat, Lucas Uzal, Sanjay Rawat, Josselin Feist, and Laurent Mounier. 2016. Toward large-scale vulnerability discovery using machine learning. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*. ACM, 85–96.

[24] Part Guide. 2016. Intel 64 and IA-32 architectures software developer's manual. *Volume 3 (3A, 3B, 3C & 3D): System Programming Guide* (2016).

[25] Mark Andrew. Hall. 1999. Correlation-Based Feature Selection for Machine Learning. (1999).

[26] Choongwoo Han. 2019. Case Study of JavaScript Engine Vulnerabilities. https://github.com/tunz/js-vuln-db.

[27] Christian Holler, Kim Herzig, and Andreas Zeller. 2012. Fuzzing with Code Fragments. In *USENIX Security Symposium*. 445–458.

[28] Nicholas Jalbert and Westley Weimer. 2008. Automated duplicate detection for bug tracking systems. In *Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008. IEEE International Conference on*. IEEE, 52–61.

[29] Gaeul Jeong, Sunghun Kim, and Thomas Zimmermann. 2009. Improving bug triage with bug tossing graphs. In *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*. ACM, 111–120.

[30] Jeffrey D. Ullman Jure Leskovec, Anand Rajaraman. 2014. Mining of Massive Datasets. http://infolab.stanford.edu/~ullman/mmds/book.pdf.

[31] Jaweria Kanwal and Onaiza Maqbool. 2012. Bug prioritization to facilitate bug report triage. *Journal of Computer Science and Technology* 27, 2 (2012), 397–412.

[32] Dongsun Kim, Xinming Wang, Sunghun Kim, Andreas Zeller, Shing-Chi Cheung, and Sooyong Park. 2011. Which crashes should i fix first?: Predicting top crashes at an early stage to prioritize debugging efforts. *IEEE Transactions on Software Engineering* 37, 3 (2011), 430–447.

[33] Kaspersky Lab. 2018. Attacks leveraging exploits for Microsoft Office grew fourfold in early 2018. https://www.kaspersky.com/about/press-releases/2018_microsoft-office-exploits.

[34] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* 18, 17 (2017), 1–5. http://jmlr.org/papers/v18/16-365.html

[35] Microsoft. 2014. Definition of a Security Vulnerability. https://docs.microsoft.com/en-us/previous-versions/tn-archive/cc751383(v=technet.10).

[36] Microsoft. 2019. EdgeHTML issue tracker. https://developer.microsoft.com/en-us/microsoft-edge/platform/issues/.

[37] Microsoft. 2019. Microsoft ChakraCore. https://github.com/Microsoft/ChakraCore.

[38] Microsoft. 2019. Microsoft Edge on Windows Insider Preview Bounty Program. https://www.microsoft.com/en-us/msrc/bounty-edge.

[39] MITRE. 2018. Common Weakness Enumeration. https://cwe.mitre.org/.

[40] MITRE. 2019. CWE-762. https://cwe.mitre.org/data/definitions/762.html.

[41] Mozilla. [n. d.]. Mozilla Bug Bounty Program. https://www.mozilla.org/en-US/security/bug-bounty/.

[42] Mozilla. [n. d.]. Mozilla Client Bug Bounty Program. https://www.mozilla.org/en-US/security/client-bug-bounty/.

[43] Mozilla. 2017. Mozilla Bug 1344415. https://bugzilla.mozilla.org/show_bug.cgi?id=1344415.

[44] Mozilla. 2018. Mozilla Bug 1493900. https://bugzilla.mozilla.org/show_bug.cgi?id=1493900.

[45] Mozilla. 2018. Mozilla Bug 1493903. https://bugzilla.mozilla.org/show_bug.cgi?id=1493903.

[46] Mozilla. 2019. Bugzilla. https://bugzilla.mozilla.org/.

[47] Mozilla. 2019. Exploitable crashes. https://developer.mozilla.org/en-US/docs/Mozilla/Security/Exploitable_crashes.

[48] Mozilla. 2019. Mozilla SpiderMonkey. https://github.com/mozilla/gecko-dev.

[49] Mozilla. 2019. Security Advisories for Firefox. https://www.mozilla.org/en-US/security/known-vulnerabilities/firefox/.

[50] G Murphy and D Cubranic. 2004. Automatic bug triage using text categorization. In *Proceedings of the Sixteenth International Conference on Software Engineering & Knowledge Engineering*. Citeseer.

[51] Stephan Neuhaus, Thomas Zimmermann, Christian Holler, and Andreas Zeller. 2007. Predicting vulnerable software components. In *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 529–540.

[52] Anh Tuan Nguyen, Tung Thanh Nguyen, Tien N Nguyen, David Lo, and Chengnian Sun. 2012. Duplicate bug report detection with a combination of information retrieval and topic modeling. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*. ACM, 70–79.

[53] Jesse Ruderman. 2007. Introducing jsfunfuzz. http://www.squarefree.com/2007/08/02/introducing-jsfunfuzz/.

[54] Adrian Schroter, Adrian Schröter, Nicolas Bettenburg, and Rahul Premraj. 2010. Do stack traces help developers fix bugs?. In *2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010)*. IEEE, 118–121.

[55] Offensive Security. 2019. Exploit Database. https://www.exploit-db.com/.

[56] Konstantin Serebryany, Derek Bruening, Alexander Potapenko, and Dmitriy Vyukov. 2012. AddressSanitizer: A Fast Address Sanity Checker. In *USENIX Annual Technical Conference*. 309–318.

[57] Internet World Stats. 2019. INTERNET USAGE STATISTICS. https://www.internetworldstats.com/stats.htm.

[58] ChakraCore team. 2019. ChakraCore Roadmap. https://github.com/Microsoft/ChakraCore/wiki/Roadmap.

[59] Yuan Tian, Chengnian Sun, and David Lo. 2012. Improved duplicate bug report identification. In *Software Maintenance and Reengineering (CSMR), 2012 16th European Conference on*. IEEE, 385–390.

[60] Shubham Tripathi, Gustavo Grieco, and Sanjay Rawat. 2017. Exniffer: Learning to Prioritize Crashes by Assessing the Exploitability from Memory Dump. In *Asia-Pacific Software Engineering Conference (APSEC), 2017 24th*. IEEE, 239–248.

[61] Spandan Veggalam, Sanjay Rawat, Istvan Haller, and Herbert Bos. 2016. Ifuzzer: An evolutionary interpreter fuzzer using genetic programming. In *European Symposium on Research in Computer Security*. Springer, 581–601.

[62] W3Techs. 2019. Usage of JavaScript for websites. https://w3techs.com/technologies/details/cp-javascript/all/all.

[63] Webkit. 2019. Webkit JavaScriptCore. https://git.webkit.org/.

[64] Rongxin Wu, Hongyu Zhang, Shing-Chi Cheung, and Sunghun Kim. 2014. CrashLocator: locating crashing faults based on crash stacks. In *Proceedings of the 2014 International Symposium on Software Testing and Analysis*. ACM, 204–214.

[65] Jun Xu, Dongliang Mu, Ping Chen, Xinyu Xing, Pei Wang, and Peng Liu. 2016. Credal: Towards locating a memory corruption vulnerability with your core dump. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 529–540.

[66] Guanhua Yan, Junchen Lu, Zhan Shu, and Yunus Kucuk. 2017. Exploitmeter: Combining fuzzing with machine learning for automated evaluation of software exploitability. In *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE, 164–175.

[67] ZERODIUM. 2019. ZERODIUM Exploit Acquisition Program. https://zerodium.com/program.html.

[68] Boyou Zhou, Anmol Gupta, Rasoul Jahanshahi, Manuel Egele, and Ajay Joshi. 2018. Hardware Performance Counters Can Detect Malware: Myth or Fact?. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. ACM, 457–468.