REFERENCES

[1] Saswat Anand, Patrice Godefroid, and Nikolai Tillmann. 2008. Demand-Driven Compositional Symbolic Execution. In *2008 14th Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. ETAPS, 367–381.

[2] Armis. 2021. Home - Armis. https://www.armis.com/. (2021).

[3] Peter Boonstoppel, Cristian Cadar, and Dawson Engler. 2008. RWset: Attacking Path Explosion in Constraint-Based Test Generation. In *2008 14th Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. ETAPS, 351–366.

[4] Suhabe Bugrara and Dawson Engler. 2013. Redundant State Detection for Dynamic Symbolic Execution. In *Proceedings of the 2013 USENIX conference on Annual Technical Conference (USENIX ATC '13)*. 199–212.

[5] Marcel Böhme, Van-Thuan Pham, Manh-Dung Nguyen, and Abhik Roychoudhury. 2017. Directed Greybox Fuzzing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2329–2344.

[6] Cristian Cadar, Daniel Dunbar, Dawson R Engler, et al. 2008. Klee: unassisted and automatic generation of high-coverage tests for complex systems programs. In *OSDI*, Vol. 8. 209–224.

[7] Daming D Chen, Maverick Woo, David Brumley, and Manuel Egele. 2016. Towards Automated Dynamic Analysis for Linux-based Embedded Firmware. In *NDSS*, Vol. 1. 1–1.

[8] Hongxu Chen, Yinxing Xue, Yuekang Li, Bihuan Chen, Xiaofei Xie, and Xiuheng Wu. 2018. Hawkeye: Towards a desired directed grey-box fuzzer. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2095–2108.

[9] Libo Chen, Yanhao Wang, Quanpu Cai, Yunfan Zhan, Hong Hu, Jiaqi Linghu, Qinsheng Hou, Chao Zhang, Haixin Duan, and Zhi Xue. 2021. Sharing More and Checking Less: Leveraging Common Input Keywords to Detect Bugs in Embedded Systems. In *30th USENIX Security Symposium (USENIX Security 21)*.

[10] Vitaly Chipounov, Volodymyr Kuznetsov, and George Candea. 2011. S2E: A platform for in-vivo multi-path analysis of software systems. *Acm Sigplan Notices* 46, 3 (2011), 265–278.

[11] Abraham A Clements, Naif Saleh Almakhdhub, Khaled S Saab, Prashast Srivastava, Jinkyu Koo, Saurabh Bagchi, and Mathias Payer. 2017. Protecting bare-metal embedded systems with privilege overlays. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 289–303.

[12] Abraham A Clements, Logan Carpenter, William A Moeglein, and Christopher Wright. 2021. Is Your Firmware Real or Re-Hosted?. In *Workshop on Binary Analysis Research (BAR)*, Vol. 2021. 21.

[13] Abraham A Clements, Eric Gustafson, Tobias Scharnowski, Paul Grosen, David Fritz, Christopher Kruegel, Giovanni Vigna, Saurabh Bagchi, and Mathias Payer. 2020. HALucinator: Firmware re-hosting through abstraction layer emulation. In *29th USENIX Security Symposium (USENIX Security 20)*. 1201–1218.

[14] Crispan Cowan, Calton Pu, Dave Maier, Jonathan Walpole, Peat Bakke, Steve Beattie, Aaron Grier, Perry Wagle, Qian Zhang, and Heather Hinton. 1998. Stackguard: Automatic adaptive detection and prevention of buffer-overflow attacks. In *USENIX Security Symposium*.

[15] eCos. 2021. eCos Home Page. https://ecos.sourceware.org/. (2021).

[16] FreeRTOS™. 2021. Real-time operating system for microcontrollers. https://www.freertos.org/. (2021).

[17] Ghidra. 2021. Ghidra. https://ghidra-sre.org/. (2021).

[18] Barak Hadad and Dor Zusman. 2020. From an URGENT/11 Vulnerability to a Full Take-Down of a Factory, Using a Single Packet. In *Black Hat Asia*.

[19] Kyriakos Ispoglou, Daniel Austin, Vishwath Mohan, and Mathias Payer. 2020. Fuzzgen: Automatic fuzzer generation. In *29th USENIX Security Symposium (USENIX Security 20)*. 2271–2287.

[20] Chung Hwan Kim, Taegyu Kim, Hongjun Choi, Zhongshu Gu, Byoungyoung Lee, Xiangyu Zhang, and Dongyan Xu. 2018. Securing Real-Time Microcontroller Systems through Customized Memory View Switching. In *NDSS*.

[21] Mingeun Kim, Dongkwan Kim, Eunsoo Kim, Suryeon Kim, Yeongjin Jang, and Yongdae Kim. 2020. FirmAE: Towards Large-Scale Emulation of IoT Firmware for Dynamic Analysis. In *Annual Computer Security Applications Conference*. 733–745.

[22] Su Yong Kim, Sangho Lee, Insu Yun, Wen Xu, Byoungyoung Lee, Youngtae Yun, and Taesoo Kim. 2017. CAB-Fuzz: Practical Concolic Testing Techniques for COTS Operating Systems. In *Proceedings of the 2017 USENIX conference on Annual Technical Conference (USENIX ATC 17)*. 689–701.

[23] Microsoft. 2006. Data Execution Prevention (DEP). (2006). http://support.microsoft.com/kb/875352/EN-US/.

[24] NASA. 2021. Command & Data-handling Systems. https://mars.nasa.gov/mro/mission/spacecraft/parts/command/. (2021).

[25] Nathan Voss. 2017. afl-unicorn: Fuzzing Arbitrary Binary Code. https://hackernoon.com/afl-unicorn-fuzzing-arbitrary-binary-code-563ca28936bf. (2017).

[26] Hui Peng, Yan Shoshitaishvili, and Mathias Payer. 2018. T-Fuzz: fuzzing by program transformation. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 697–710.

[27] Dawei Qi, HOANG D. T. NGUYEN, and Abhik Roychoudhury. 2013. Path exploration based on symbolic output. *ACM Transactions on Software Engineering and Methodology 22*, 32 (2013), 1–41.

[28] David A. Ramos and Dawson Engler. 2015. Under-Constrained Symbolic Execution: Correctness Checking for Real Code. In *24th USENIX Security Symposium (USENIX Security 15)*. 49–64.

[29] Siegfried Rasthofer, Steven Arzt, Marc Miltenberger, and Eric Bodden. 2016. Harvesting Runtime Values in Android Applications That Feature Anti-Analysis Techniques. In *NDSS*, Vol. 16. 21–24.

[30] Sanjay Rawat, Vivek Jain, Ashish Kumar, Lucian Cojocar, Cristiano Giuffrida, and Herbert Bos. 2017. VUzzer: Application-aware evolutionary fuzzing. In *Proceedings of the 24th Network and Distributed System Security Symposium*. The Internet Society.

[31] Nilo Redini, Aravind Machiry, Ruoyu Wang, Chad Spensky, Andrea Continella, Yan Shoshitaishvili, Christopher Kruegel, and Giovanni Vigna. 2020. Karonte: Detecting insecure multi-binary interactions in embedded firmware. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1544–1561.

[32] Majid Salehi, Danny Hughes, and Bruno Crispo. 2020. μSBS: Static Binary Sanitization of Bare-metal Embedded Devices for Fault Observability. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*. 381–395.

[33] Ben Seri, Gregory Vishnepolsky, and Dor Zusman. 2019. Critical vulnerabilities to remotely compromise VxWorks, the most popular RTOS. *White paper, ARMIS, URGENT/11* (2019).

[34] Yan Shoshitaishvili, Ruoyu Wang, Christopher Salls, Nick Stephens, Mario Polino, Andrew Dutcher, John Grosen, Siji Feng, Christophe Hauser, Christopher Kruegel, et al. 2016. Sok:(state of) the art of war: Offensive techniques in binary analysis. In *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 138–157.

[35] Nick Stephens, John Grosen, Christopher Salls, Andrew Dutcher, Ruoyu Wang, Jacopo Corbetta, Yan Shoshitaishvili, Christopher Kruegel, and Giovanni Vigna. 2016. Driller: Augmenting Fuzzing Through Selective Symbolic Execution. In *NDSS*, Vol. 16. 1–16.

[36] Fish Wang and Yan Shoshitaishvili. 2017. Angr-the next generation of binary analysis. In *2017 IEEE Cybersecurity Development (SecDev)*. IEEE, 8–9.

[37] Haohuang Wen, Zhiqiang Lin, and Yinqian Zhang. 2020. FirmXRay: Detecting Bluetooth Link Layer Vulnerabilities From Bare-Metal Firmware. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 167–180.

[38] WindRiver. 2021. VxWorks: The Leading RTOS for the Intelligent Edge. https://www.windriver.com/products/vxworks. (2021).

[39] Michelle Y Wong and David Lie. 2016. Intellidroid: a targeted input generator for the dynamic analysis of android malware. In *NDSS*, Vol. 16. 21–24.

[40] Qiuping Yi, Zijiang Yang, Shengjian Guo, Chao Wang, Jian Liu, and Chen Zhao. 2018. Eliminating Path Redundancy via Postconditioned Symbolic Execution. *IEEE Transactions on Software Engineering* 44, 1 (2018), 25–43.

[41] Zeping Yu, Rui Cao, Qiyi Tang, Sen Nie, Junzhou Huang, and Shi Wu. 2020. Order Matters: Semantic-aware neural networks for binary code similarity detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1145–1152.

[42] Zimu Yuan, Muyue Feng, Feng Li, Gu Ban, Yang Xiao, Shiyang Wang, Qian Tang, He Su, Chendong Yu, Jiahuan Xu, et al. 2019. B2SFinder: Detecting Open-Source Software Reuse in COTS Software. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 1038–1049.

[43] Yu Zhang, Wei Huo, Kunpeng Jian, Ji Shi, Haoliang Lu, Longquan Liu, Chen Wang, Dandan Sun, Chao Zhang, and Baoxu Liu. 2019. SrFuzzer: An automatic fuzzing framework for physical soho router devices to discover multi-type vulnerabilities. In *Proceedings of the 35th Annual Computer Security Applications Conference*. 544–556.

[44] Yaowen Zheng, Ali Davanian, Heng Yin, Chengyu Song, Hongsong Zhu, and Limin Sun. 2019. FIRM-AFL: high-throughput greybox fuzzing of iot firmware via augmented process emulation. In *28th USENIX Security Symposium (USENIX Security 19)*. 1099–1114.

[45] Wenzhe Zhu, Zhou Yu, Jiashui Wang, and Ruikai Liu. 2019. Dive into VxWorks Based IoT Device: Debug the Undebugable Device. In *Black Hat Asia*.