[21] Chin E, Felt AP, Greenwood K, Wagner D. Analyzing inter-application communication in Android. In: Proc. of the 9th Int'l Conf. on Mobile Systems, Applications, and Services. ACM Press, 2011. 239−252. https://dl.acm.org/citation.cfm?id=2000018

[22] Greengard S. Cybersecurity gets smart. Communications of the ACM, 2016,59(5):29−31. [doi: 10.1145/2898969]

[23] Zhou W, Zhou Y, Grace M, Jiang X, Zou X. Fast, scalable detection of piggybacked mobile applications. In: Proc. of the 3rd ACM Conf. on Data and Application Security and Privacy. ACM Press, 2013. 185−196. [doi: 10.1145/2435349.2435377]

[24] Chen K, Wang XQ, Chen Y, Wang P, Lee Y, Wang XF, Ma B, Wang AH, Zhang YJ, Zou W. Following devils footprints: Crossplatform analysis of potentially harmful libraries on Android and iOS. In: Proc. of the 37th IEEE Symp. on Security and Privacy, Ser. (S&P 2016). 2016. [doi: 10.1109/SP.2016.29]

[25] Backes M, Bugiel S, Derr E. Reliable third-party library detection in Android and its security applications. In: Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security. ACM Press, 2016. 356−367. [doi: 10.1145/2976749.2978333]

[26] Li MH, Wang W, Wang P, Wu DH, Liu J, Xue R, Huo W. LibD: Scalable and precise third-party library detection in Android markets. In: Proc. of the 39th Int'l Conf. on Software Engineering. ACM Press, 2017. [doi: 10.1109/ICSE.2017.38]

[27] Jing Q, Vasilakos AV, Wan J, Lu J, Qiu D. Security of the Internet of things: Perspectives and challenges. Wireless Networks, 2014, 20(8):2481−2501. [doi: 10.1007/s11276-014-0761-7]

[28] Cao Z, Hu J, Chen Z, Xu M, Zhou X. Feedback: Towards dynamic behavior and secure routing for wireless sensor networks. In: Proc. of the 20th Int'l Conf. on Advanced Information Networking and Applications—Vol.2. IEEE Computer Society, 2006. 160−164. [doi: 10.1109/AINA.2006.179]

[29] Jhaveri RH, Patel SJ, Jinwala DC. DoS attacks in mobile ad hoc networks: A survey. In: Proc. of the 2nd Int'l Conf. on Advanced Computing & Communication Technologies. IEEE, 2012. 535−541. [doi: 10.1109/ACCT.2012.48]

[30] Chen TM, Abu-Nimeh S. Lessons from Stuxnet. Computer, 2011,44(4):91−93. [doi: 10.1109/MC.2011.115]

[31] Douceur JR. The Sybil attack. In: Proc. of the Int'l Workshop on Peer-to-Peer Systems. Berlin, Heidelberg: Springer-Verlag, 2002. 251−260. [doi: 10.1007/3-540-45748-8_24]

[32] Hlavacs H, Treutner T, Gelas JP, Lefevre L, Orgerie AC. Energy consumption side-channel attack at virtual machines in a cloud. In: Proc. of the 9th Int'l Conf. on Dependable, Autonomic and Secure Computing (DASC). IEEE, 2011. 605−612. [doi: 10.1109/DASC.2011.110]

[33] Wu Z, Xu Z, Wang H. Whispers in the hyper-space: High-Speed covert channel attacks in the cloud. In: Proc. of the 21st USENIX Security Symp. 2012. 159−173. https://dl.acm.org/citation.cfm?id=2362802

[34] Liu F, Yarom Y, Ge Q, Heiser G, Lee RB. Last-Level cache side-channel attacks are practical. In: Proc. of the IEEE Symp. on Security and Privacy. 2015. 605−622. [doi: 10.1109/SP.2015.43]

[35] Rocha F, Correia M. Lucy in the sky without diamonds: Stealing confidential data in the cloud. In: Proc. of the 2011 IEEE/IFIP the 41st Int'l Conf. on Dependable Systems and Networks Workshops (DSN-W). IEEE, 2011. 129−134. [doi: 10.1109/DSNW.2011.5958798]

[36] Hong S, Xu L, Wang H, Gu G. Poisoning network visibility in software-defined networks: New attacks and countermeasures. In: Proc. of the 22th Annual Network and Distributed System Security Symp. (NDSS 2015). 2015. [doi: 10.14722/ndss.2015.23283]

[37] Dhawan M, Poddar R, Mahajan K, Mann V. SPHINX: Detecting security attacks in software-defined networks. In: Proc. of the 22th Annual Network and Distributed System Security Symp. (NDSS 2015). 2015. [doi: 10.14722/ndss.2015.23064]

[38] Slopek A, Vlajic N. Economic denial of sustainability (EDoS) attack in the cloud using Web-bugs. In: Proc. of the 17th Int'l Symp. on Research in Attacks, Intrusions, and Defenses (RAID 2014). Switzerland: Springer Int'l Publishing. 2014. 469−471. https://link.springer.com/book/10.1007/978-3-319-11379-1#page=482

[39] Zhao S, Lee PPC, Lui J, Guan X, Ma X, Tao J. Cloud-Based push-styled mobile botnets: A case study of exploiting the cloud to device messaging service. In: Proc. of the 28th Annual Computer Security Applications Conf. ACM Press, 2012. 119−128. [doi: 10.1145/2420950.2420968]

[40] Tankard C. Advanced persistent threats and how to monitor and deter them. Network Security, 2011,2011(8):16−19. [doi: 10.1016/S1353-4858(11)70086-1]

[41] Li F, Lai A, Ddl D. Evidence of advanced persistent threat: A case study of malware for political espionage. In: Proc. of the 6th Int'l Conf. on Malicious and Unwanted Software (MALWARE). IEEE, 2011. 102−109. [doi: 10.1109/MALWARE.2011.6112333]

[42] Juels A, Yen TF. Sherlock Holmes and the case of the advanced persistent threat. In: Proc. of the 5th USENIX Workshop on Large-Scale Exploits and Emergent Threats. 2012. https://dl.acm.org/citation.cfm?id=2228343

[43] Linn C, Debray S. Obfuscation of executable code to improve resistance to static disassembly. In: Proc. of the 10th ACM Conf. on Computer and Communications Security. ACM Press, 2003. 290−299. [doi: 10.1145/948109.948149]

[44] Moser A, Kruegel C, Kirda E. Limits of static analysis for malware detection. In: Proc. of the 23rd Annual Computer Security Applications Conf. IEEE, 2007. 421−430. [doi: 10.1109/ACSAC.2007.21]

[45] Ma XJ. Sandbox based intelligent malware analysis technology [Ph.D. Thesis]. Beijing: University of Chinese Academy of Sciences, 2017 (in Chinese with English abstract).

[46] Kolbitsch C, Kirda E, Kruegel C. The power of procrastination: Detection and mitigation of execution-stalling malicious code. In: Proc. of the 18th ACM Conf. on Computer and Communications Security. ACM Press, 2011. 285−296. [doi: 10.1145/2046707.2046740]

[47] Kharraz A, Robertson W, Balzarotti D, Bilge L, Kirda E. Cutting the Gordian knot: A look under the hood of ransomware attacks. In: Proc. of the Int'l Conf. on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer Int'l Publishing, 2015. [doi: 10.1007/978-3-319-20550-2_1]

[48] Google LLC. Google Play. All your entertainment, anywhere you go. 2012. http://googleblog.blogspot.co.uk/2012/03/introducing-google-play-all-your.html

[49] Kovacheva A. Efficient code obfuscation for Android. In: Proc. of the Int'l Conf. on Advances in Information Technology. Springer Int'l Publishing, 2013. 104−119. [doi: 10.1007/978-3-319-03783-7_10]

[50] Faruki P, Bharmal A, Laxmi V, Gaur MS, Conti M, Rajarajan M. Evaluation of Android anti-malware techniques against Dalvik bytecode obfuscation. In: Proc. of the IEEE 13th Int'l Conf. on Trust, Security and Privacy in Computing and Communications. IEEE, 2014. 414−421. [doi: 10.1109/TrustCom.2014.54]

[51] Li L, Bissyandé TF, Octeau D, Klein J. Droidra: Taming reflection to support whole-program analysis of Android apps. In: Proc. of the 25th Int'l Symp. on Software Testing and Analysis. ACM Press, 2016. 318−329. [doi: 10.1145/2931037.2931044]

[52] Rastogi V, Chen Y, Jiang X. Droidchameleon: Evaluating Android anti-malware against transformation attacks. In: Proc. of the 8th ACMSIGSAC Symp. on Information,Computer and Communications Security (ASIACCS 2013). 2013. 329−334. http://dl.acm.org/citation.cfm?id=2484355

[53] Egele M, Scholte T, Kirda E, Kruegel C. A survey on automated dynamic malware-analysis techniques and tools. ACM Computing Surveys (CSUR), 2012,44(2):6. [doi: 10.1145/2089125.2089126]

[54] Willems C, Holz T, Freiling F. Toward automated dynamic malware analysis using cwsandbox. IEEE Security and Privacy, 2007, 5(2):32−39. [doi: 10.1109/MSP.2007.45]

[55] Bayer U, Moser A, Kruegel C, Kirda E. Dynamic analysis of malicious code. Journal in Computer Virology, 2006,2(1):67−77. [doi: 10.1007/s11416-006-0012-2]

[56] Dinaburg A, Royal P, Sharif M, Lee W. Ether: Malware analysis via hardware virtualization extensions. In: Proc. of the 15th ACM Conf. on Computer and Communications Security. ACM Press, 2008. 51−62. [doi: 10.1145/1455770.1455779]

[57] Xu D, Ming J, Wu D. Cryptographic function detection in obfuscated binaries via bit-precise symbolic loop mapping. In: Proc. of the 38th IEEE Symp. on Security and Privacy. 2017. 22−24. [doi: 10.1109/SP.2017.56]

[58] Park Y, Reeves DS, Stamp M. Deriving common malware behavior through graph clustering. Computers & Security, 2013,39: 419−430. [doi: 10.1016/j.cose.2013.09.006]

[59] Liang Z, Yin H, Song D. HookFinder: Identifying and understanding malware hooking behaviors. In: Proc. of the Network & Distributed System Security Symp. 2008. 41. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.8123&rep=rep1&type=pdf

[60] Wu DJ, Mao CH, Wei TE, Lee HM, Wu KP. Droidmat: Android malware detection through manifest and API calls tracing. In: Proc. of the 7th Asia Joint Conf. on Information Security (Asia JCIS). IEEE, 2012. 62−69. [doi: 10.1109/AsiaJCIS.2012.18]

[61] Burguera I, Zurutuza U, Nadjm-Tehrani S. Crowdroid: Behavior-Based malware detection system for Android. In: Proc. of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices. ACM Press, 2011. 15−26. [doi: 10.1145/2046614.2046619]

[62] Chen ZF, Li QB, Zhang P, Ding WB. Data characteristics-based kernel malware detection. Ruan Jian Xue Bao/Journal of Software, 2016,27(12):3172−3191 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4927.htm [doi: 10.13328/j.cnki.jos.004927]

[63] Christodorescu M, Jha S, Seshia SA, Song D, Bryant R. Semantics-Aware malware detection. In: Proc. of the 2005 IEEE Symp. on Security and Privacy (S&P 2005). IEEE, 2005. 32−46. [doi: 10.1109/SP.2005.20]

[64] Kirda E, Kruegel C, Banks G, Vigna G, Kemmerer RA. Behavior-Based spyware detection. In: Proc. of the Usenix Security. 2006. 6. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.148.8514

[65] Fuchs AP, Chaudhuri A, Foster JS. ScAndroid: Automated security certification of Android. 2009. https://www.researchgate.net/publication/228847936_SCanDroid_Automated_security_certification_of_Android_applications

[66] Chan PPF, Hui LCK, Yiu SM. Droidchecker: Analyzing Android applications for capability leak. In: Proc. of the 5th ACM Conf. on Security and Privacy in Wireless and Mobile Networks. ACM Press, 2012. 125−136. [doi: 10.1145/2185448.2185466]

[67] Schultz MG, Eskin E, Zadok F, Zadok E, Stolfo SJ. Data mining methods for detection of new malicious executables. In: Proc. of the 2001 IEEE Symp. on Security and Privacy. IEEE, 2001. 38−49. [doi: 10.1109/SECPRI.2001.924286]

[68] Rieck K, Trinius P, Willems C, Holz T. Automatic analysis of malware behavior using machine learning. Journal of Computer Security, 2011,19(4):639−668. [doi: 10.3233/JCS-2010-0410]

[69] Amos B, Turner H, White J. Applying machine learning classifiers to dynamic Android malware detection at scale. In: Proc. of the 9th Int'l Wireless Communications and Mobile Computing Conf. (IWCMC). IEEE, 2013. 1666−1671. [doi: 10.1109/IWCMC.2013.6583806]

[70] Sadeghi A, Bagheri H, Garcia J. A taxonomy and qualitative comparison of program analysis techniques for security assessment of android software. IEEE Trans. on Software Engineering, 2016. [doi: 10.1109/TSE.2016.2615307]

[71] Cui J. The analysis and research of proxy and VPN communication software [MS. Thesis]. Beijing: Beijing University of Posts and Telecommunications, 2012 (in Chinese with English abstract).

[72] Wilhelm J, Chiueh T. A forced sampled execution approach to kernel rootkit identification. In: Proc. of the Workshop on Recent Advances in Intrusion Detection. Berlin, Heidelberg: Springer-Verlag, 2007. 219−235. [doi: 10.1007/978-3-540-74320-0_12]

[73] Moser A, Kruegel C, Kirda E. Exploring multiple execution paths for malware analysis. In: Proc. of the 2007 IEEE Symp. on Security and Privacy (SP 2007). IEEE, 2007. 231−245. [doi: 10.1109/SP.2007.17]

[74] Cadar C, Ganesh V, Pawlowski PM, Dill DL, Engler DR. EXE: Automatically generating inputs of death. ACM Trans. on Information and System Security, 2008,12(2):10. [doi: 10.1145/1180405.1180445]

[75] Comparetti PM, Salvaneschi G, Kirda E, Kolbitsch C, Kruegelx C, Zanero S. Identifying dormant functionality in malware programs. In: Proc. of the 2010 IEEE Symp. on Security and Privacy. IEEE, 2010. 61−76. [doi: 10.1109/SP.2010.12]

[76] Enck W, Gilbert P, Han S, Tendulkar V, Chun BG, Cox LP, Jung J, Mcdaniel P, Sheth AN. TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. ACM Trans. on Computer Systems, 2014,32(2):5. [doi: 10.1145/2494522]

[77] Christodorescu M, Jha S, Kruegel C. Mining specifications of malicious behavior. In: Proc. of the 1st India Software Engineering Conf. ACM Press, 2008. 5−14. [doi: 10.1145/1342211.1342215]

[78] Zhang HL, Zou W, Han XH. Drive-by-Download mechanisms and defenses. Ruan Jian Xue Bao/Journal of Software, 2013,24(4):843−858 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4376.htm [doi: 10.3724/SP.J.1001.2013.04376]

[79] Common vulnerabilities and exposures. https://cve.mitre.org

[80] Bass T, Gruber D. A glimpse into the future of id. Login: Special Issue Intrusion Detection, The USENIX Association Magazine, 1999, 40−45.

[81] Sun H, Li HP, Zeng QK. Statically detect and run-time check integer-based vulnerabilities with information flow. Ruan Jian Xue Bao/Journal of Software, 2013,24(12):2767−2781 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4385.htm [doi: 10.3724/SP.J.1001.2013.04385]

[82] Brumley D, Poosankam P, Song D, Zheng J. Automatic patch-based exploit generation is possible: Techniques and implications. In: Proc. of the 2008 IEEE Symp. on Security and Privacy (SP 2008). IEEE, 2008. 143−157. [doi: 10.1109/SP.2008.17]

[83] Cadar C, Dunbar D, Engler D. KLEE: Unassisted and automatic generation of high coverage tests for complex systems programs. In: Proc. of the OSDI 2008. 2008. http://zoo.cs.yale.edu/classes/cs422/2010/bib/engler08klee.pdf

[84] Godefroid P, Levin MY, Molnar D. Automated whitebox fuzz testing. In: Proc. of the 15th Annual Network and Distributed System Security Symp. (NDSS 2008). 2008. http://www.microsoft.com/en-us/research/wp-content/uploads/2007/05/tr-2007-58.pdf

[85] Engler D, Chen DY, Hallem S, Chou A, Chelf B. Bugsas deviant behavior: A general approach to inferring errors in systems code. In: Proc. of the ACM Symp. on Operating Systems Principles (SOSP). 2001. 57−72. http://web.stanford.edu/~engler/deviant-sosp-01.pdf

[86] Yamaguchi F, Wressnegger C, Gascon H, Rieck K. Chucky: Exposing missing checks in source code for vulnerability discovery. In: Proc. of the 2013 ACM SIGSAC Conf. on Computer & Communications Security. 2013. 499−510. [doi: 10.1145/2508859.2516665]

[87] Grieco G, Grinblat GL, Uzal L, Rawat S, Feist J, Mounier L. Toward large-scale vulnerability discovery using machine learning. In: Proc. of the Codaspy. 2016. 85−96. [doi: 10.1145/2857705.2857720]

[88] Yin H, Song D. Temu: Binary code analysis via whole-system layered annotative execution. Technical Report, UCB/EECS-2010-3, Berkeley: EECS Department, University of California, 2010.

[89] Henderson A, Prakash A, Yan LK, Hu X. Make it work, make it right, make it fast: Building a platform-neutral whole-system dynamic binary analysis platform. In: Proc. of the ISSTA 2014. 2014. http://www.cs.ucr.edu/~heng/pubs/issta14.pdf

[90] Kemerlis VP, Portokalidis G, Jee K, Keromytis A. libdft: Practical dynmaic data flow tracking for commodity systems. In: Proc. of the VEE 2012. 2012. [doi: 10.1145/2151024.2151042]

[91] Kang MG, Mccamant S, Poosankam P, Song D. DTA++: Dynamic taint analysis with targeted control-flow propagation. In: Proc. of the Network and Distributed System Security Symp. (NDSS 2011). 2011. https://people.eecs.berkeley.edu/~dawnsong/papers/2011%20dta++-ndss11.pdf

[92] Clause J, Li W, Orso A. Dytan: A generic dynamic taint analysis framework. In: Proc. of the Int'l Symp. on Software Testing and Analysis (ISSTA 2007). London, 2007. 196−206. https://www.cc.gatech.edu/fac/Alex.Orso/papers/clause.li.orso.ISSTA07.pdf

[93] Cui W, Peinado M, Cha SK, Fratantonio Y, Kemerlis VP. Retracer: Triaging crashes by reverse execution from partial memory dumps. In: Proc. of the 38th Int'l Conf. on Software Engineering. 2016. [doi: 10.1145/2884781.2884844]

[94] Xu J, Mu D, Chen P, Xing X, Wang P, Liu P. CREDAL: Towards locating a memory corruption vulnerability with your core dump. In: Proc. of the CCS 2016. 2016. [doi: 10.1145/2976749.2978340]

[95] Sidiroglou-Douskos S, Lahtinen E, Rittenhouse N, Piselli P, Long F, Kim D, Rinard M. Targeted automatic interger overflow discovery using goal-directed conditional branch enforcement. In: Proc. of the ASPLOS 2015. 2015. [doi: 10.1145/2775054.2694389]

[96] Newsome J, Song D. Dynamic taint analysis: Automatic detection, analysis, and signature generation of exploit attacks on commodity software. In: Proc. of the 12th Network and Distributed Systems Security Symp. 2005. http://users.ece.cmu.edu/~dawnsong/papers/taintcheck.pdf

[97] Wu R, Zhang H, Cheung SC, Kim S. Crashlocator: Locating crashing faults based on crash stacks. In: Proc. of the 2014 Int'l Symp. on Software Testing and Analysis. 2014. [doi: 10.1145/2610384.2610386]

[98] Brumley D, Poosankam P, Song D, Zheng J. Automatic patch-based exploit generation is possible: Techniques and implications. In: Proc. of the IEEE Symp. on Security and Privacy. IEEE, 2008. 143−157. [doi: 10.1109/SP.2008.17]

[99] Sha L, Fu J, Chen J, Peng G. PVDF: An automatic patch-based vulnerability description and fuzzing method. In: Proc. of the Communications Security Conf. IET, 2014. 1−8. [doi: 10.1049/cp.2014.0733]

[100] Zhang M, Yin H. Automatic generation of vulnerability-specific patches for preventing component hijacking attacks. In: Proc. of the Android application security. Springer Int'l Publishing, 2016. 45−61. [doi: 10.1007/978-3-319-47812-8_4]

[101] Cha SK, Avgerinos T, Rebert A, Brumley D. Unleashing mayhem on binary code. In: Proc. of the IEEE Symp. on Security and Privacy. IEEE, 2012. 380−394. [doi: 10.1109/SP.2012.31]

[102] Wang MH, Su P, Li Q, Ying L, Yang Y, Feng D. Automatic polymorphic exploit generation for software vulnerabilities. In: Proc. of the Int'l Conf. on Security and Privacy in Communication Systems. Springer Int'l Publishing, 2013. 216−233. [doi: 10.1007/978-3-319-04283-1_14]

[103] Hu H, Chua ZL, Adrian S, Saxena P, Liang Z. Automatic generation of data-oriented exploits. In: Proc. of the 24th USENIX Security Symp. (USENIX Security 2015). 2015. 177−192. https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-hu.pdf

[104] Hu H, Shinde S, Adrian S, Chua ZL, Saxena P, Liang Z. Data-Oriented programming: On the expresivenes of non-control data attacks. In: Proc. of the S&P 2016. 2016. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7546545

[105] Hornyack P, Han S, Jung J, Schechter S, Wetherall D. These are not the droids you're looking for: Retrofitting Android to protect data from imperious applications. In: Proc. of the 18th ACM Conf. on Computer and Communications Security. ACM Press, 2011. [doi: 10.1145/2046707.2046780]

[106] Zhou Y, Zhang X, Jiang X, Freeh VW. Taming information-stealing smartphone applications (on Android). In: Proc. of the Int'l Conf. on Trust and Trustworthy Computing. Berlin, Heidelberg: Springer-Verlag, 2011. 93−107. [doi: 10.1007/978-3-642-21599-5_7]

[107] Fawaz K, Shin KG. Location privacy protection for smartphone users. In: Yung M, Li N, eds. Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. Scottsdale: ACM Press, 2014. 239−250.

[108] Beresford AR, Rice A, Skehin N, Sohan R. Mockdroid: Trading privacy for application functionality on smartphones. In: Proc. of the 12th Workshop on Mobile Computing Systems and Applications. ACM Press, 2011. 49−54. [doi: 10.1145/2184489.2184500]

[109] Pearce P, Felt AP, Nunez G, Wagner D. Addroid: Privilege separation for applications and advertisers in Android. In: Proc. of the 7th ACM Symp. on Information, Computer and Communications Security. ACM Press, 2012. 71−72. [doi: 10.1145/2414456.2414498]

[110] Shekhar S, Dietz M, Wallach DS. ADsplit: Separating smartphone advertising from applications. In: Kohno T, ed. Proc. of the 21st USENIX Security Symp. Bellevue: USENIX Association, 2012. 553−567.

[111] Zhang X, Ahlawat A, Du W. AFrame: Isolating advertisements from mobile applications in Android. In: Proc. of the 29th Annual Computer Security Applications Conf. ACM Press, 2013. 9−18. [doi: 10.1145/2523649.2523652]

[112] Wu C, Zhou Y, Patel K, Liang Z, Jiang X. AirBag: Boosting smartphone resistance to malware infection. In: Bauer L, ed. Proc. of the 21st Annual Network and Distributed System Security Symp. (NDSS 2014). San Diego: Internet Society, 2014.

[113] Liu Y, Zhou T, Chen K, Chen H, Xia Y. Thwarting memory disclosure with efficient hypervisor-enforced intra-domain isolation. In: Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security. ACM Press, 2015. 1607−1619. [doi: 10.1145/2810103.2813690]

[114] Kurmus A, Zippel R. A tale of two kernels: Towards ending kernel hardening wars with split kernel. In: Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. ACM Press, 2014. 1366−1377. [doi: 10.1145/2660267.2660331]

[115] Zhou Z, Yu M, Gligor VD. Dancing with giants: Wimpy kernels for on-demand isolated I/O. In: Proc. of the 2014 IEEE Symp. on Security and Privacy. IEEE, 2014. 308−323. [doi: 10.1109/SP.2014.27]

[116] Nikolaev R, Back G. VirtuOS: An operating system with kernel virtualization. In: Proc. of the 24th ACM Symp. on Operating Systems Principles. ACM Press, 2013. 116−132. [doi: 10.1145/2517349.2522719]

[117] Azab AM, Ning P, Shah J, Chen Q. Hypervision across worlds: Real-Time kernel protection from the arm trustzone secure world. In: Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. ACM Press, 2014. 90−102. [doi: 10.1145/2660267.2660350]

[118] Li W, Li H, Chen H, Xia Y. Adattester: Secure online mobile advertisement attestation using trustzone. In: Proc. of the 13th Annual Int'l Conf. on Mobile Systems, Applications, and Services. ACM Press, 2015. 75−88. [doi: 10.1145/2742647.2742676]

[119] Zhou Y, Wang X, Chen Y, Wang Z. Armlock: Hardware-Based fault isolation for arm. In: Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. ACM Press, 2014. 558−569. [doi: 10.1145/2660267.2660344]

[120] Guan L, Lin J, Luo B, Jing J, Wang J. Protecting private keys against memory disclosure attacks using hardware transactional memory. In: Proc. of the 2015 IEEE Symp. on Security and Privacy. IEEE, 2015. 3−19. [doi: 10.1109/SP.2015.8]

[121] Xu R, Saïdi H, Anderson R. Aurasium: Practical policy enforcement for Android applications. In: Kohno T, ed. Proc. of the 21st USENIX Security Symp. Bellevue: USENIX Association, 2012. 539−552.

[122] Rastogi V, Qu Z, McClurg J, Cao Y, Chen Y. Uranine: Real-Time privacy leakage monitoring without system modification for Android. In: Proc. of the Int'l Conf. on Security and Privacy in Communication Systems. Springer Int'l Publishing, 2015. 256−276. [doi: 10.1007/978-3-319-28865-9_14]

[123] Backes M, Bugiel S, Hammer C, Schranz O, Styp-Rekowsky P. Boxify: Full-Fledged app sandboxing for stock Android. In: Jung J, ed. Proc. of the 24th USENIX Security Symp. Washington: USENIX Association, 2015. 691−706.

[124] Smalley S, Craig R. Security enhanced (SE) Android: Bringing flexible MAC to Android. In: Ning P, ed. Proc. of the 20th Annual Network and Distributed System Security Symp. (NDSS 2013). San Diego: Internet Society, 2013. 20−38.

[125] Bugiel S, Heuser S, Sadeghi AR. Flexible and fine-grained mandatory access control on Android for diverse security and privacy policies. In: King S, ed. Proc. of the 22nd USENIX Security Symp. Washington: USENIX Association, 2013. 131−146.

[126] Heuser S, Nadkarni A, Enck W, Sadeghi AR. ASM: A programmable interface for extending Android security. In: Fu K, ed. Proc. of the 23rd USENIX Security Symp. San Diego: USENIX Association, 2014. 1005−1019.

[127] Backes M, Bugiel S, Gerling S, Styp-Rekowsky P. Android security framework: Extensible multi-layered access control on Android. In: Proc. of the 30th Annual Computer Security Applications Conf. ACM Press, 2014. 46−55. [doi: 10.1145/2664243.2664265]

[128] Zhang Y, Yang M, Gu G, Chen H. Rethinking permission enforcement mechanism on mobile systems. IEEE Trans. on Information Forensics and Security, 2016,11(10):2227−2240. [doi: 10.1109/TIFS.2016.2581304]

[129] Wang R, Enck W, Reeves D, Zhang X. EASEAndroid: Automatic policy analysis and refinement for security enhanced Android via large-scale semi-supervised learning. In: Jung J, ed. Proc. of the 24th USENIX Security Symp. Washington: USENIX Association, 2015. 351−366.

[130] Cowan C, Pu C, Maier D, Hinton H, Walpole J. StackGuard: Automatic adaptive detection and prevention of buffer-overflow attacks. In: Rubin A, ed. Proc. of the Conf. on Usenix Security Symp. San Antonio: USENIX Association, 1998. 63−78.

[131] Trev N. Data Execution Prevention. Lect Publishing, 2011. http://nosmut.com/Executable_space_protection.html

[132] Miller FP, Vandome AF, Mcbrewster J. Address Space Layout Randomization. Alphascript Publishing, 2010. http://nosmut.com/Address_space_layout_randomization.html

[133] Prandini M, Ramilli M. Return-Oriented programming. IEEE Security & Privacy Magazine, 2012,10(6):84–87. [doi: 10.1109/MSP.2012.152]

[134] Abadi M, Budiu M, Erlingsson U, Ligatti J. Control-Flow integrity. In: Proc. of the 12th ACM Conf. on Computer and Communications Security. Alexandria: ACM Press, 2005. 340–353. [doi: 10.1145/1102120.1102165]

[135] Nie M, Su P, Li Q, Wang Z, Ying L, Hu J, Feng D. XEDE: Practical exploit early detection. In: Proc. of the 18th Int'l Symp. on Research in Attacks, Intrusions and Defenses (RAID 2015). Springer-Verlag, 2015. 198–221. [doi: 10.1007/978-3-319-26362-5_10]

[136] Monshizadeh M, Naldurg P, Venkatakrishnan VN. MACE: Detecting privilege escalation vulnerabilities in Web applications. In: Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. ACM Press, 2014. 690–701. [doi: 10.1145/2660267.2660337]

[137] Pellegrino G, Balzarotti D. Toward black-box detection of logic flaws in Web applications. In: Bauer L, ed. Proc. of the 21st Annual Network and Distributed System Security Symp. (NDSS 2014). San Diego: Internet Society, 2014.

[138] Weissbacher M, Robertson W, Kirda E, Kruegel C, Vigna G. Zigzag: Automatically hardening Web applications against client-side validation vulnerabilities. In: Jung J, ed. Proc. of the 24th USENIX Security Symp. San Antonio: USENIX Association, 2015. 737–752.

[139] Doupé A, Cui W, Jakubowski MH, Peinado M, Kruegel C, Vigna G. deDacota: Toward preventing server-side XSS via automatic code and data separation. In: Proc. of the 2013 ACM SIGSAC Conf. on Computer & Communications Security. ACM Press, 2013. 1205–1216. [doi: 10.1145/2508859.2516708]

[140] Dietz M, Shekhar S, Pisetsky Y, Shu A, Wallach DS. QUIRE: Lightweight provenance for smart phone operating systems. In: Wagner D, ed. Proc. of the USENIX Security Symp. San Francisco: USENIX Association, 2011. 31.

[141] Chan PPF, Hui LCK, Yiu SM. Droidchecker: Analyzing Android applications for capability leak. In: Proc. of the fifth ACM Conf. on Security and Privacy in Wireless and Mobile Networks. ACM Press, 2012. 125–136. [doi: 10.1145/2185448.2185466]

[142] Lu L, Li Z, Wu Z, Lee W, Jiang G. Chex: Statically vetting Android apps for component hijacking vulnerabilities. In: Proc. of the 2012 ACM Conf. on Computer and Communications Security. ACM Press, 2012. 229–240. [doi: 10.1145/2382196.2382223]

[143] Yang K, Zhuge J, Wang Y, Zhou L, Duan H. IntentFuzzer: Detecting capability leaks of Android applications. In: Proc. of the 9th ACM Symp. on Information, Computer and Communications Security. ACM Press, 2014. 531–536. [doi: 10.1145/2590296.2590316]

[144] Zhang M, Yin H. AppSealer: Automatic generation of vulnerability-specific patches for preventing component hijacking attacks in Android applications. In: Proc. of the 21st Annual Network and Distributed System Security Symp. (NDSS 2014). 2014. [doi: 10.14722/ndss.2014.23255]

[145] Arzt S, Rasthofer S, Fritz C, Bodden E, Bartel A, Klein J, Traon YL, Octeau D, McDaniel P. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps. ACM SIGPLAN Notices, 2014,49(6):259–269. [doi: 10.1145/2666356.2594299]

[146] Wei F, Roy S, Ou X. AMAndroid: A precise and general inter-component data flow analysis framework for security vetting of Android apps. In: Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. ACM Press, 2014. 1329–1341. [doi: 10.1145/2660267.2660357]

[147] Cao Y, Fratantonio Y, Bianchi A, Egelez M, Kruegely C, Vignay G, Chen Y. EdgeMiner: Automatically detecting implicit control flow transitions through the Android framework. In: Proc. of the 22nd Annual Network and Distributed System Security Symp. (NDSS 2015). 2015. [doi: 10.14722/ndss.2015.23140]

[148] Arzt S, Bodden E. StubDroid: Automatic inference of precise data-flow summaries for the Android framework. In: Proc. of the Int'l Conf. on Software Engineering. ACM Press, 2016. [doi: 10.1145/2884781.2884816]

[149] Zhang Y, Yang M, Xu B, Yang Z, Gu G, Ning P, Wang XS, Zhang B. Vetting undesirable behaviors in Android apps with permission use analysis. In: Gligor V, Yung M, eds. Proc. of the 2013 ACM SIGSAC Conf. on Computer & Communications Security. Berlin: ACM Press, 2013. 611–622.

[150] Nan Y, Yang M, Yang Z, Zhou S, Gu G, Wang XF. Uipicker: User-Input privacy identification in mobile applications. In: Jung J, ed. Proc. of the 24th USENIX Security Symp. Washington: USENIX Association, 2015. 993–1008.

[151] Huang J, Li Z, Xiao X, Wu Z, Lu K, Zhang X, Jiang G. Supor: Precise and scalable sensitive user input detection for Android apps. In: Jung J, ed. Proc. of the 24th USENIX Security Symp. Washington: USENIX Association, 2015. 977−992.

[152] Zhang N, Yuan K, Naveed M, Zhou X, Wang XF. Leave me alone: App-Level protection against runtime information gathering on Android. In: Proc. of the 2015 IEEE Symp. on Security and Privacy. IEEE, 2015. 915−930. [doi: 10.1109/SP.2015.61]

[153] Zhang Y, Reiter MK. Düppel: Retrofitting commodity operating systems to mitigate cache side channels in the cloud. In: Proc. of the 2013 ACM SIGSAC Conf. on Computer & Communications Security. ACM Press, 2013. 827−838. [doi: 10.1145/2508859.2516741]

[154] Varadarajan V, Ristenpart T, Swift M. Scheduler-Based defenses against cross-VM side-channels. In: Fu K, ed. Proc. of the 23rd USENIX Security Symp. San Diego: USENIX Association, 2014. 687−702.

[155] Moon SJ, Sekar V, Reiter MK. Nomad: Mitigating arbitrary cloud side channels via provider-assisted migration. In: Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security. ACM Press, 2015. 1595−1606. [doi: 10.1145/2810103.2813706]

[156] Pattuk E, Kantarcioglu M, Lin Z, Ulusoy H. Preventing cryptographic key leakage in cloud virtual machines. In: Fu K, ed. Proc. of the 23rd USENIX Security Symp. San Diego: USENIX Association, 2014. 703−718.

[157] Liang J, Jiang J, Duan H, Li K, Wan T, Wu J. When HTTPS meets CDN: A case of authentication in delegated service. In: Proc. of the 2014 IEEE Symp. on Security and Privacy. IEEE, 2014. 67−82. [doi: 10.1109/SP.2014.12]

[158] Chen J, Jiang J, Zheng X, Duan H, Liang J, Li K, Wan T, Paxson V. Forwarding-Loop attacks in content delivery networks. In: Proc. of the 23th Annual Network and Distributed System Security Symp. (NDSS 2016). 2016. [doi: 10.14722/ndss.2016.23442]

[159] Ray M, Dispensa S. Renegotiating TLS. 2009. https://kryptera.se/Renegotiating%20TLS.pdf

[160] Mavrogiannopoulos N, Vercauteren F, Velichkov V, Preneel B. A cross-protocol attack on the TLS protocol. In: Proc. of the 2012 ACM Conf. on Computer and Communications Security. ACM Press, 2012. 62−72. [doi: 10.1145/2382196.2382206]

[161] Georgiev M, Iyengar S, Jana S, Anubhai R, Boneh D, Shmatikov V. The most dangerous code in the world: Validating SSL certificates in non-browser software. In: Proc. of the 2012 ACM Conf. on Computer and Communications Security. ACM Press, 2012. 38−49. [doi: 10.1145/2382196.2382204]

[162] Fahl S, Harbach M, Muders T, Smith M, Baumgärtner L, Freisleben B. Why eve and mallory love Android: An analysis of Android SSL (in) security. In: Proc. of the 2012 ACM Conf. on Computer and Communications Security. ACM Press, 2012. 50−61. [doi: 10.1145/2382196.2382205]

[163] Brubaker C, Jana S, Ray B, Khurshid S, Shmatikov V. Using frankencerts for automated adversarial testing of certificate validation in SSL/TLS implementations. In: Proc. of the 2014 IEEE Symp. on Security and Privacy. IEEE, 2014. 114−129. [doi: 10.1109/SP.2014.15]

[164] He B, Rastogi V, Cao Y, Chen Y, Venkatakrishnan VN, Yang R, Zhang Z. Vetting SSL usage in applications with SSLint. In: Proc. of the 2015 IEEE Symp. on Security and Privacy. IEEE, 2015. 519−534. [doi: 10.1109/SP.2015.38]

[165] Pandita R, Xiao X, Yang W, Enck W, Xie T. Whyper: Towards automating risk assessment of mobile applications. In: King S, ed. Proc. of the 22nd USENIX Security Symp. Washington: USENIX Association, 2013. 527−542.

[166] Qu Z, Rastogi V, Zhang X, ChenY, ZhuT, Chen Z. Autocog: Measuring the description-to-permission fidelity in Android applications. In: Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. ACM Press, 2014. 1354−1365. [doi: 10.1145/2660267.2660287]

[167] Gorla A, Tavecchia I, Gross F, Zeller A. Checking app behavior against app descriptions. In: Proc. of the 36th Int'l Conf. on Software Engineering. ACM Press, 2014. 1025−1035. [doi: 10.1145/2568225.2568276]

[168] Huang J, Zhang X, Tan L, Wang P, Liang B. AsDroid: Detecting stealthy behaviors in Android applications by user interface and program behavior contradiction. In: Proc. of the 36th Int'l Conf. on Software Engineering. ACM Press, 2014. 1036−1046. [doi: 10.1145/2568225.2568301]

[169] Yang W, Xiao X, Andow B, Li S, Xie T, Enck W. Appcontext: Differentiating malicious and benign mobile app behaviors using context. In: Proc. of the 37th IEEE Int'l Conf. on Software Engineering. IEEE, 2015. 303−313. [doi: 10.1109/ICSE.2015.50]

[170] Yang Z, Yang M, Zhang Y, Gu G, Ning P, Wang XS. Appintent: Analyzing sensitive data transmission in Android for privacy leakage detection. In: Proc. of the 2013 ACM SIGSAC Conf. on Computer & Communications Security. ACM Press, 2013. 1043−1054. [doi: 10.1145/2508859.2516676]

[171] Zhang M, Duan Y, Feng Q, Yin H. Towards automatic generation of security-centric descriptions for Android apps. In: Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security. ACM Press, 2015. 518−529. [doi: 10.1145/2810103.2813669]

[172] Roesner F, Kohno T, Moshchuk A, Parno B, Wang HJ, Cowan C. User-Driven access control: Rethinking permission granting in modern operating systems. In: Proc. of the 2012 IEEE Symp. on Security and Privacy. IEEE, 2012. 224–238. [doi: 10.1109/SP.2012.24]

[173] Roesner F, Kohno T. Securing embedded user interfaces: Android and beyond. In: Kruegel C, Myers A, Halevi S, eds. Proc. of the 22nd USENIX Security Symp. Vienna: ACM Press, 2013. 97–112.

[174] Ringer T, Grossman D, Roesner F. AUDACIOUS: User-Driven access control with unmodified operating systems. In: Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security (CCS 2016). 2016. 204–216. [doi: 10.1145/2976749.2978344]

附中文参考文献:

[1] CNCERT 互联网安全威胁报告——2016,6. http://www.cac.gov.cn/2016-08/01/c_1119418586.htm

[2] 国家互联网应急中心.关于部分境内网站存在 Ramnit 恶意代码攻击的有关情况通报.2016. http://www.cert.org.cn/publish/main/10/2016/20160422145241769412671/20160422145241769412671_.html

[3] 国家互联网应急中心.植入恶意程序被控制联网智能设备安全隐患多.2016. http://www.cert.org.cn/publish/main/12/2016/20161201134333495740421/20161201134333495740421_.html

[20] 卿斯汉.Android 安全研究进展.软件学报,2016,27(1):45−71. http://www.jos.org.cn/1000-9825/4914.htm [doi: 10.13328/j.cnki.jos.004914]

[45] 马新建.基于沙箱的恶意代码智能分析技术研究[博士学位论文].北京:中国科学院大学,2017.

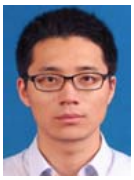
[62] 陈志锋,李清宝,张平,丁文博.基于数据特征的内核恶意软件检测.软件学报,2016,27(12):3172−3191. http://www.jos.org.cn/1000-9825/4927.htm [doi: 10.13328/j.cnki.jos.004927]

[71] 崔杰.代理类和 VPN 类通信工具的分析与研究[硕士学位论文].北京:北京邮电大学,2012.

[78] 张慧琳,邹维,韩心慧.网页木马机理与防御技术.软件学报,2013,24(4):843−858. http://www.jos.org.cn/1000-9825/4376.htm [doi: 10.3724/SP.J.1001.2013.04376]

[81] 孙浩,李会朋,曾庆凯.基于信息流的整数漏洞插装和验证.软件学报,2013,24(12):2767−2781. http://www.jos.org.cn/1000-9825/4385.htm [doi: 10.3724/SP.J.1001.2013.04385]

刘剑(1976－),男,云南石屏人,博士,副研究员,CCF 高级会员,主要研究领域为软件安全,Web 安全,移动安全,网络攻防.



张源(1987－),男,博士,讲师,主要研究领域为系统软件,系统安全.

苏璞睿(1976－),男,博士,研究员,博士生导师,主要研究领域为网络与系统安全.

朱雪阳(1971－),女,博士,副研究员,CCF 专业会员,主要研究领域为嵌入式系统设计,性能分析与优化,形式化方法.

杨珉(1979－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为网络空间安全,移动系统安全.

林惠民(1947－),男,博士,研究员,博士生导师,CCF 会士,主要研究领域为并发理论,进程代数,模型检测,形式化方法.

和亮(1985－),男,博士,副研究员,主要研究领域为网络与系统安全,软件漏洞分析.