Record-Replay Architecture as a General Security Framework

综述报告

1. 关于 RnR-Safe

本论文是由 UIUC 的 i-acoma 小组发表于 2018HPCA，由美国国家科学基金委(NSF)部分赞助。

为了应对不断发展的安全威胁，论文提出了一种新的框架，记录和确定性重放（RnR），用于补充硬件安全性功能，称为 RnR-Safe。

2. 发展脉络

提到记录重放（RnR），一般大家会先想到声音的记录和重现；由于重放的确定性，记录和重放是一种流行的架构技术，在调试和可疑程序分析领域中得到广泛研究。

有几篇论文研究了 RnR 在安全领域中的使用。ReVirt[1](2002)展示了在虚拟机层面使用 RnR 对 Linux 内核中的时间竞争事件进行事后离线分析的示例；IntroVirt [2](2005)探索了使用 VM 级 RnR 来确定一旦发现零日攻击系统是否先前已被利用；Speck [3](2008)探索了使用 OS 级别检测和程序级 RnR 的组合来从程序的关键路径中删除安全检查；ParanoidAndroid [4](2010)和 Secloud [5](2013)探索了在云中维护移动设备复制品的可能性，并在云中执行程序级 RnR。

Aftersight [6](2008)首次提出并执行了异构重放，在 VMware 中记录虚拟机执行过程，并在 QEMU 中重放以进行工作负载的深度分析；然后，V2E[7](2012)沿用了 Aftersight 的思路，使用 KVM+TEMU 透明地实现了 RnR。

和本篇论文提出的 RnR-Safe 最接近的工作就是 Aftersight 和 V2E。Aftersight 为 VM 级 RnR 提供了在线分析的一般方向，V2E 则是透明地实现了该思路；遗憾的是本论文没有关于 V2E 的调研，但是在最后评估阶段所使用测试平台 Insight[8](2014)框架和也是采用的 KVM+QEMU 方式，功能上应该接近。

Aftersight 建议使用 VM 级 RnR 对系统执行过程进行在线动态分析，它假设完整的重放分析一直在运行，并能够赶上（或只是稍慢于）记录; 否则，它会失去精确度并可能引入误报；对于 ROP 检测所需的重度分析，这不是一个合理的假设；而 V2E 没有揭示其性能评估（估计性能不理想）。

RnR-Safe 框架提供了在线 RnR 安全分析的关键实践因素—隔离的检查点和报警重放者，即基于需求的触发警报重放，而不是不断运行分析。实现了即时在线重放，这是 Aftersight 和 V2E 难以保障实现的。

RnR-Safe 虽然是一种新的想法，但是总体看来还是在 Aftersight 思路下的扩展思维。尽管 RnR 由于其确定性重放的特性得到广泛研究，但是目前还没有一个得到广泛应用和认可的 RnR 平台（缺少类似 Hadoop、Docker 这种被广泛接受的平台），多数想法目前可行性并不高，RnR-Safe 也是这种问题。

3. 主要研究机构

RnR 并不算是热门的研究领域，落地成功的项目很少，多数作为开源项目用于研究目的。目前安全方面顶会应属 NDSS，但是 RnR 主要作为解决相关问题的一种框架/结构出现，发表在 HPCA 和 SOSP 等会议也常见。

调试方向来看，Mozilla 研究小组费时五年左右于 2017 完成了 RR[9]项目，利用最新的硬件功能实现了低开销的记录重放，可能是以后的一种思路；安全领域，Aftersight 的思路应该还是主流，V2E 作者 Lok Kwong Yan 所在团队一直致力于恶意软件分析，RnR 也只是他们提出的一种解决方案。

2014 年以后关于 RnR 的论文很少，有的大多也是关于移动端的构思。自 2014 年以来，能够逃避检测的恶意软件数量激增了 2000％Kruegel[10](2015), Lastline 的首席科学家 Christopher Kruegel 认为沙箱不能被视为恶意软件检测中的银弹，这种基于虚拟机的 RnR 前景不明；目前相关研究热度可能降低，转而求助人工智能研究相关的解决手段。我们小组目前的华为项目最后就要求采用人工智能手段分析相关数据等，希望可以登上 AI 的大船，借其东风。

4. 最新进展

目前最新的想法就是前面提到的的就是 Mozilla 的 RR 项目，借助最新的硬件功能降低 RnR 的性能开销，增强其可部署性。借助硬件发展的潮流，增强 RnR 的实用性，将其推广。

5. 其他相关工作

5.1 控制流完整性

实现控制流完整性（CFI）[11]是防止代码重用攻击的合理技术。它需要防止控制流图（CFG）或阴影堆栈不允许的分支目的地。轻松的方法通过放宽有效分支目标的定义来避免影子堆栈或 CFG。有效的分支目标取决于目标指令的类型或位置。例如，英特尔 CET 重新使用多字节 NOP 指令来标记直接分支的有效目的地。其他方法通过分支目的地相对于功能边界的接近来定义有效性。一般来说，这种方法无法完全消除小工具，允许 ROP 有效载荷构建。此外，CFI 的影子堆栈完整性和寿命是 CFI 的另一个关注点。

5.2 地址空间布局随机化

ASLR 通过随机化堆栈，堆和程序指令的位置来加强系统抵御 ROP 攻击。因此，攻击者必须首先通过地址泄露攻击发现代码和堆栈的位置。这一额外要求增加了安装 ROP 攻击的难度。此外，为了进一步加强 ASLR，有人提出了加强系统防止地址泄露攻击的建议。总之，ASLR 是一种实用，有效且广泛部署的强化技术，确实使 ROP 攻击更难以安装。但是，直到地址泄露攻击面被消除，ASLR 无法完全消除 ROP 攻击。

5.3 内存安全

保证内存安全可以完全消除代码重用攻击。通过有效的内存引用进行内存访问是安全的，并且当它没有违反原始分配的范围时。确保第一个属性需要跟踪每个指针的生命周期，检测已释放的指针的任何取消引用。确保第二个属性需要检查每个指针取消引用，以确保它在结构的边界内。最近，英特尔已开始通过内存保护扩展（MPX）[12]在其处理器中加入硬件加速边界检查。虽然 MPX 是在商品系统上提供存储安全性的最重要的一步，但它并不能防止前面提到的第一个属性。如果没有此属性，则使用后的攻击面仍然存在。

6 参考文献

[1] G. W. DUNLAP, S. T. KING, S. CINAR, M. A. BASRAI, and P. M. CHEN, “ReVirt: Enabling Intrusion Analysis Through Virtual-machine Logging and Replay,” SIGOPS Oper. Syst. Rev., vol. 36, Dec. 2002.

[2] A. JOSHI, S. T. KING, G. W. DUNLAP, AND P. M. CHEN, “Detecting Past and Present Intrusions Through Vulnerability-specific Predicates,” in Symposium on Operating Systems Principles, SOSP ’05, 2005.

[3] E. B. NIGHTINGALE, D. PEEK, P. M. CHEN, AND J. FLINN, “Parallelizing Security Checks on Commodity Hardware,” in International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS XIII, 2008.

[4] G. PORTOKALIDIS, P. HOMBURG, K. ANAGNOSTAKIS, AND H. BOS, “Paranoid Android: Versatile Protection for Smartphones,” in Computer Security Applications Conference, ACSAC ’10, 2010.

[5] S. ZONOUZ, A. HOUMANSADR, R. BERTHIER, N. BORISOV, AND W. SANDERS, “Secloud: A Cloud-based Comprehensive and Lightweight Security Solution for Smartphones,” Comput. Secur., vol. 37, Sept. 2013.

[6] J. CHOW, T. GARFINKEL, AND P. M. CHEN, “Decoupling Dynamic Program Analysis from Execution in Virtual Environments,” USENIX ATC, June 2008.

[7] LOK-KWONG YAN, MANJUKUMAR JAYACHANDRA, MU ZHANG, AND HENG YIN. “V2E: combining hardware virtualization and software emulation for transparent and extensible malware analysis.” ACM Sigplan Notices, 47(7):227-238, 2012.

[8] R. SENTHILKUMARAN AND P. KULKARNI, “InSight: A Framework for Application Diagnosis using Virtual Machine Record and Replay,” Tech. Rep. TR-CSE-2014-57, Department of Computer Science and Engineering, Indian Institute of Technology Bombay, January 2014.

[9] ROBERT O’CALLAHAN, CHRIS JONES, NATHAN FROYD, KYLE HUEY, ALBERT NOL. “Engineering Record And Replay For Deployability Extended Technical Report”, USENIX, January 2017.

[10] LAKHANI. “Malware Sandbox and Breach Detection Evasion Techniques.” Dr. Chaos. Retrieved 28 October 2015.

[11] M. ABADI, M. BUDIU, U. ERLINGSSON, AND J. LIGATTI, “Control-flow Integrity,” in Conference on Computer and Communications Security, CCS ’05, 2005.

[12] Intel Corporation, Intel 64 and IA-32 Architectures Software Developer’s Manual, Volume 1. 2017.