Poster: Automatic Detection of Confused-Deputy Attacks on ARM TrustZone Environments

Darius Suciu[1], Stephen McLaughlin[2], Hayawardh Vijayakumar[2], Lee Harrison[2], Michael Grace[2], Amir Rahmati[1,2]
[1]*Stony Brook University*,[2]*Samsung Research America*,
{dsuciu,amir}@cs.stonybrook.edu {h.vijayakuma,s.mclaughlin,lee.harrison,m1.grace,amir.rahmati}@samsung.com

Smartphones are increasingly used for both normal and security-critical applications. On one hand, they are used for communication, gaming, and video streaming. On the other hand, they perform security-sensitive operations such as banking transactions, two-factor authentication, and confidential data storage. The complexity of modern applications and the operating systems (OSes) needed to support these operations makes devices prone to software bugs. These bugs introduce vulnerabilities that attackers exploit to obtain access to sensitive user data such as banking passwords and encryption keys.

To isolate security-critical applications from normal applications, ARM-based smartphones use ARM TrustZone [1]. TrustZone allows a single system-on-chip to run in two different worlds - a *Secure World* and a *Non-secure* or *Normal World* - by isolating CPU registers, memory, and peripherals in hardware. Code running in the Secure World can access both secure and non-secure memory and peripherals, whereas code running in the Normal World can access only non-secure versions. Each world has its own separate software stack of applications and OSes. To enable communication between these two worlds, the Secure World OS provides a set of Secure Monitor Calls (SMCs) to the Normal World.

The Secure World only runs applications approved by device manufacturers. Thus, to obtain Secure World access, attackers have to either trick device manufacturers into installing their malware or find vulnerable Secure World components that are exposed though SMCs to Normal World. Recent works [2]–[4] have shown that SMCs can be exploited to obtain control over Secure World applications and even the Secure World OS, compromising the entire device. Even though these known vulnerabilities have been patched, new exploitable bugs may be found as the Secure World code expands.

To mitigate Secure World attacks, Secure World applications are split into two groups, Trusted applications (TAs) and Trusted drivers (TDs). The Normal World can only communicate though SMCs with TA, which are constrained to their own address space. TDs, on the other hand, can access and change memory pages used by TAs, TDs, Normal World applications or the Normal World OS, but are not directly accessible from the Normal World. TAs have to rely on TDs to make any changes in their address space (e.g. allocate or map memory pages). For example, using Inter-Process Communication (IPC), TAs can request TDs to map additional memory, specific physical pages or even copy data from/to TA-provided physical memory locations. This separation prevents attacker controlled TAs or TA confused-deputy attacks (e.g., Boomerang [5]) from compromising the Normal World kernel. The division of Secure World applications enables device manufacturers only allow internally verified executables to run as TDs and restrict third party executables to run as TAs, outside the Normal World Trusted Computing Base(TCB).

In this work, we describe a new class of confused-deputy attacks, which enable attackers to use compromised TAs to trick TDs into leaking data or injecting malicious code into attacker controlled locations. This type of attack not only re-enables attackers to compromise the Normal World kernel using a compromised TA, but also allows them to compromise other TAs running in the Secure World.

TDs are responsible for preventing TAs from maliciously using their exposed IPCs (e.g., changing Normal World kernel code, leaking TA encryption keys, etc.). Unfortunately, TDs do not know the Normal World layout, or the location of TA's confidential data. Consequently, only lax constraints are placed on the IPCs exposed by TDs (e.g., can't copy data into/from Secure World kernel memory or security-sensitive e-fuse locations). These constraints are insufficient for TDs to distinguish between legitimate and malicious IPC requests. For example, certain TAs rely on TD IPC requests to share memory pages. However, compromised TAs can use the same IPCs to request sharing of memory pages containing confidential data. On TDs examined, we have found IPCs that can be used by attackers to read or modify any memory pages belonging to the Normal World or other TAs.

To identify TDs vulnerable to the new confused-deputy attacks, we have designed a tool capable of automatically identifying vulnerable IPCs in TD binaries though symbolic analysis [6]. This tool solves the following binary analysis challenges: (I) Identifying TD binaries capable of performing memory operations, (II) Isolating the TDs that rely on IPC provided input for the memory operations, and (III) Identifying the constraints imposed on the IPCs input. This tool enables TD developers to easily identify dangerous IPCs and ensure the constrains imposed prevent confused-deputy attacks. To automatically detect vulnerable TDs in a system, future challenges represent (1) automatically identifying the minimal constraints required to prevent malicious use of the IPCs, and (2) optimizing our tool for efficient TD symbolic analysis.

The novel confused-deputy attack presented allows compromised TAs to trick TDs into leaking Secure World confidential data, or compromise memory pages belonging to other TAs, TDs, even Normal World OS and applications. Initial results indicate that our semantic analysis based tool can identify exploitable IPCs exposed by TDs. Further work is aimed to enable our tool to automatically identify vulnerable TDs.

References

[1] ARM. Bulding a secure system using trustzone technology. *ARM Technical White Paper*, 2009.

[2] CVE-2014-7920. Online at https://nvd.nist.gov/vuln/detail/CVE-2014-7920.

[3] CVE-2014-7921. Online at https://nvd.nist.gov/vuln/detail/CVE-2014-7921.

[4] CVE-2016-2431. Online at https://nvd.nist.gov/vuln/detail/CVE-2016-2431.

[5] A. Machiry, E. Gustafson, C. Spensky, C. Salls, N. Stephens, R. Wang, A. Bianchi, Y. R. Choe, C. Kruegel, and G. Vigna. Boomerang: Exploiting the semantic gap in trusted execution environments. In *Proceedings of the 2017 Network and Distributed System Security Symposium (NDSS)*, 2017.

[6] Y. Shoshitaishvili, R. Wang, C. Salls, N. Stephens, M. Polino, A. Dutcher, J. Grosen, S. Feng, C. Hauser, C. Kruegel, and G. Vigna. SoK: (State of) The Art of War: Offensive Techniques in Binary Analysis. In *IEEE Symposium on Security and Privacy*, 2016.