2018 Research Interest/Project Ideas

Mathias Payer

https://hexhive.github.io/publications/

==========

Protecting Applications in the Presence of Vulnerabilities

Applications written in C and C++ are prone to memory safety and type safety vulnerabilities that allow exploitation, code execution, or information leaks.

These vulnerabilities manifest themselves in, e.g., malware attacks, cryptlockers, or botnets. The HexHive group focuses on software security: protecting the integrity and availability of our systems in the presence of vulnerabilities. On one hand, we work on mitigations, stopping ongoing attacks against unknown vulnerabilities and, on the other hand, we develop sanitizers that enable developers to vet their applications and discover vulnerabilities.

For mitigations, we focus on stopping control-flow hijacking, an integral part of any exploit. We have worked on efficient control-flow integrity mechanisms leveraging binary rewriting and compiler-based transformations. More recently, we have introduced a new defense policy that enforces object type integrity.

Initially geared towards C++ applications, this property enforces integrity for any created object and protects virtual dispatch from adversarial state changes.

Interesting future directions are enforcement of type integrity and use-after-free safety.

For sanitizers, we focus on full memory safety and type safety. We have developed compiler-based mechanisms that allow developers to vet their code from any such vulnerabilities. These development tools are highly flexible and allow targeted discovery of bugs based on existing test cases or integrating with fuzzing to trigger any violations. Orthogonally, we focus on fuzzing techniques for binary-only IoT devices as well as binary-only program transformations to trigger vulnerabilities hidden deep in applications.

We release all prototype implementations as open-source. Our work has been integrated into mainstream compilers such as LLVM and we are working with industry partners (e.g., Intel, Google, or Mozilla) to integrate them into their products.