(K)ERIS: A Novel Approach for API Security Testing, Applied to the System Call Interface of the Linux kernel

Dimitris E. Simos, Bernhard Garn

Combinatorial Designs meet Software Testing and Information Security

Motivation

- We cannot test everything.
- Exhaustive search of problem space increases time needed exponentially.
- Automated detection of security vulnerabilities.

Combinatorial Security Testing (CST)

- Parameters and values provide abstract models of attacks.
- Generated test sets provide 100% coverage of $t$-way parameter value combinations.
- Automated test set generation, execution and evaluation via dedicated test oracle.

Technical Challenges

- Generation of minimal $t$-way test sets is a hard combinatorial optimization problem.
- Modelling of parameters, values and constraints is domain-specific.
- Deploy CST to all application layers of information security.

Combinatorial API Testing

- **Focus:** Test APIs function calls of software / libraries.
- **Modeling:** Combinatorial models:
  - IPM via equivalence- and category partitioning
  - IPM via novel flattening methodology
- **ERIS:** Highly configurable testing framework encompassing CT, execution environment, logging and database infrastructure.

Large-Scale Kernel Testing

Case Study

- Total of 3082 systems-under-test:
  - 23 different system calls
  - 134 kernel versions
- Kernel versions tested in the range of v4.0 up to v4.6:
  - The final releases
  - All release candidates
  - A selection of stable releases
- 102h execution time.

Evaluation via Differential Testing

- Compare number of accepted vs rejected system calls between versions.
- Mostly stable behaviour between versions.
- Largest deviations in the `settimeofday` system call.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Accepted</th>
<th>Rejected</th>
<th>Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>pid</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>arg_address</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>page_0xff</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
</tbody>
</table>

Evaluation via Kernel Address Sanitizer (KASAN)

- Test oracle uses internal dynamic memory error detector of Linux.
- Fine-tuned combinatorial model of a network configuration setup.
- Demonstrated reproducibility of vulnerability in the `sendto` system call.

Automated Test Execution Framework

- **Ease of use:** Only high-level parameters needed, everything else handled by the system.
- **Test-runs:** Each invocation runs in a dedicated virtual machine.
- **Logging:** Extensive information is captured:
  - Adjustable to user demands / needs
- **Database:** Allows sophisticated post-processing queries.

ERIS: Combinatorial Kernel Testing

- **Focus:** Reliability and quality assurance of kernel software.
- **Motivation:** Kernel is the central authority to ensure security.
- **SUTs:** System calls of every git-commit of any (variant of) Linux.
- **Evaluation:** Various kernel crashes for RCs and distribution kernels.

Vision

- **Goal:** Extend approach.
- **Modeling:** Optimization and automation of testing.
- **Automated $t$-way testing and translation layers.
- **Testing of security patches to ensure attack-free environments.

Vision

- **Continuous integration tests of kernel versions.**
- **Web monitoring platform.**