Towards Auditing of Control-Flow Integrity

Luke Atherton - Supervised by: Konstantinos Markantonakis

Information Security Group, Smart Card and IoT Security Center
Royal Holloway, University of London

The Smart Card and Internet of Things Security Centre

Objectives

- Investigate existing method providing control-flow integrity;
- Propose a solution for enabling the audit of control-flow integrity.

Introduction

- Control-flow integrity is a useful measure of secure software execution;
- When using control-flow integrity as a policy is states that the execution flow of an application must follow the control-flow graph generated from the application;
- The problem of enforcing control-flow integrity can be approached from a three different directions: prevention, detection and attestation;
- In this paper, we intend to add a fourth method of enforcing control-flow integrity - audit. We will propose a solution which enables the tracking and storing of control-flow data in audit-friendly reports.

Control-Flow Graphs

Control-flow graphs (CFG) are a method used to formally describe the legitimate paths an application can take during execution. A simple of measure of control-flow integrity is to verify whether instructions are processed in an order which abides by the application's CFG.
CFGs consist of vertices - basic blocks, and edges - transition. Basic blocks represent a sequence of instructions which will always run from beginning to end. Transitions are made, for example, when a *jump* occurs.
Transitions can take several forms:

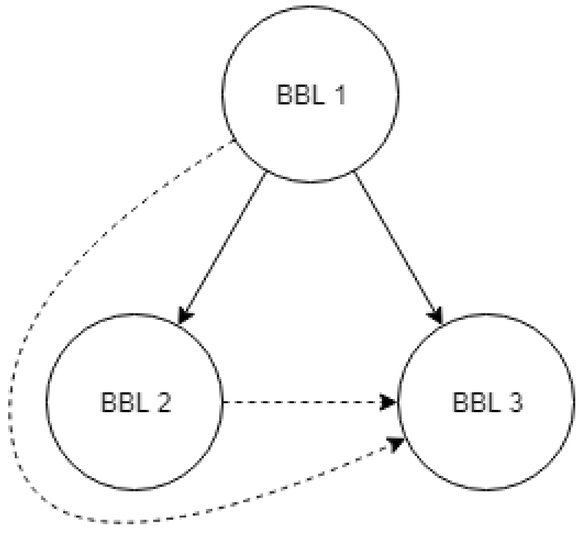
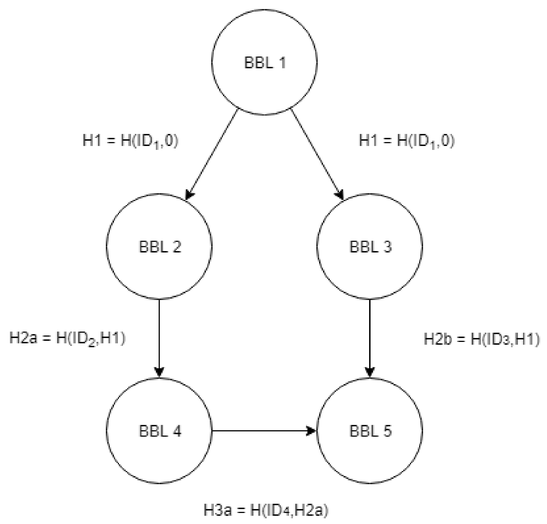


Figure 1:Illegal control-flow

Control-Flow Integrity

Control-flow integrity can be proved using several methods:
Prevention Theoretically CFI could not compromised. E.g. Read xor Write or deterministically encrypted instructions (ref).
Detection CFI compromise is detected during executions. E.g. stack canaries or shadow stacks (ref).
Attestation Variation of solutions have been examined, where control-flow is tracked at real time and used in an attestation protocol.



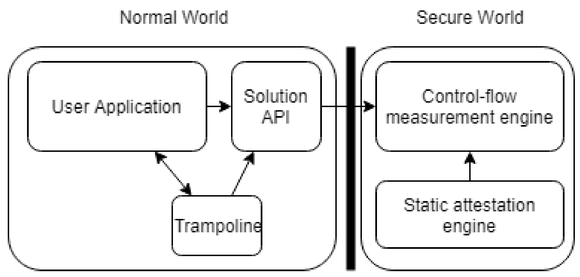


Figure 3:ARM TrustZone Implementation

Control-flow Monitoring

We propose to insert x

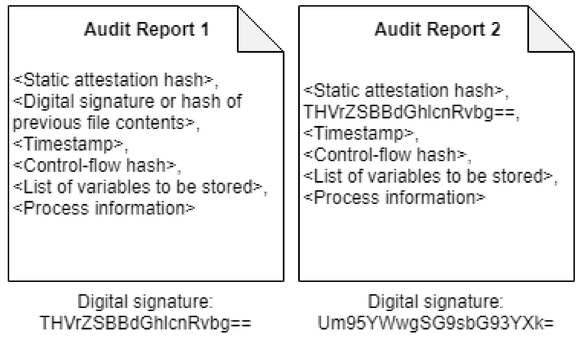


Figure 4:Audit files

Audit Files

As well as containing the control-flow hash, the audit files will contain:

- Initial attestation report;
- Digital signature of previous report;
- Operating environment information such as important variables and currently running processes.

Benefits

The proposed solutions enables the a new method of handling control-flow within embedded systems:

- Historic evidence of control-flow;
- Binding of variables to control-flow snapshot;

Conclusion

Further work also

References

Contact Information

- Web: https://scc.rhul.ac.uk/
- Email: luke.atherton.2018@live.rhul.ac.uk
- Twitter: @LucialHz