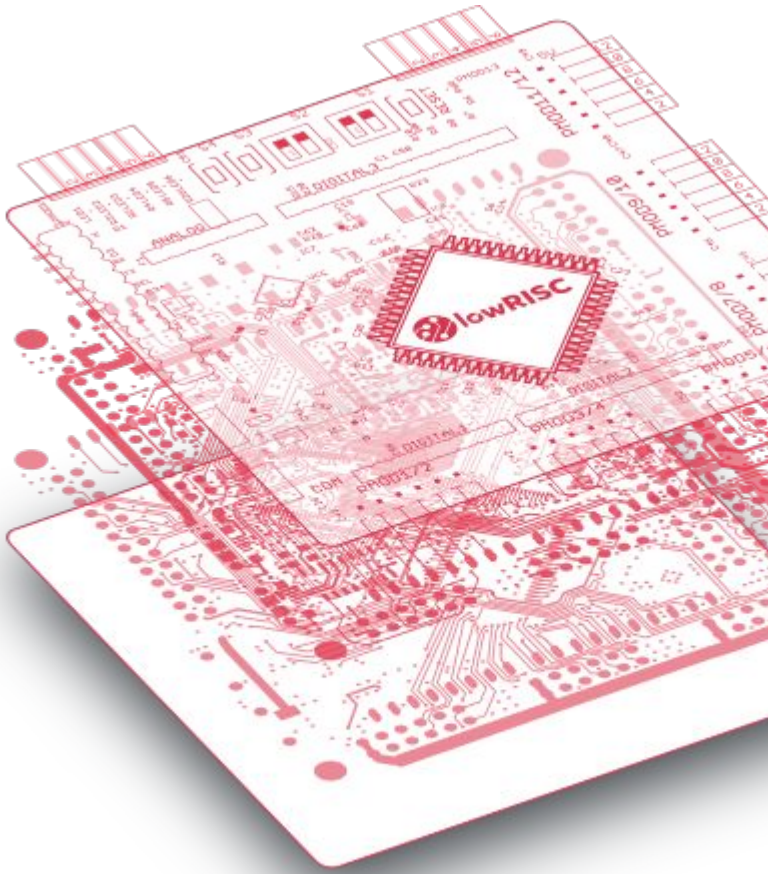
lowRISC

OpenTitan®'s Hardware Security Analysis Framework



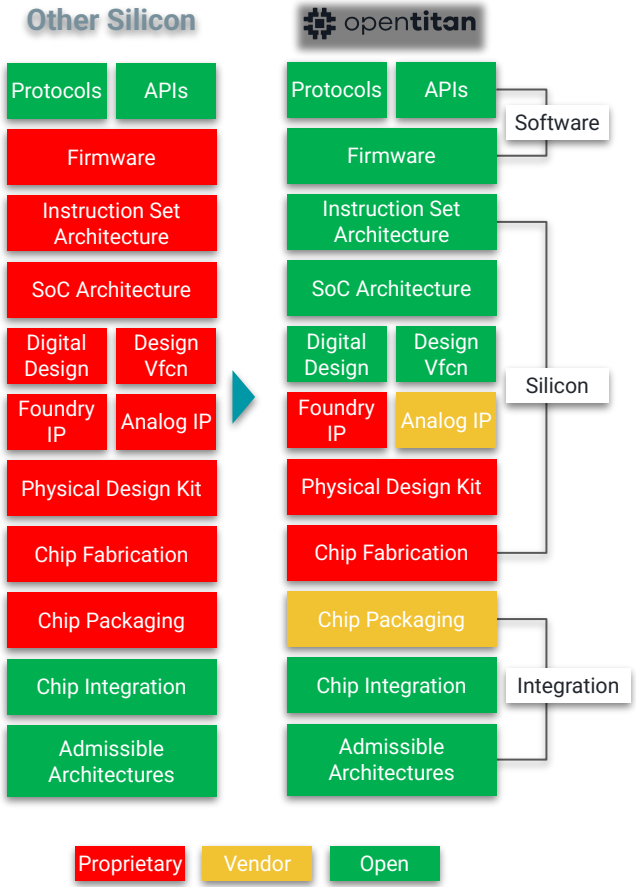
Pascal Nasahl

JAIF | October 2025

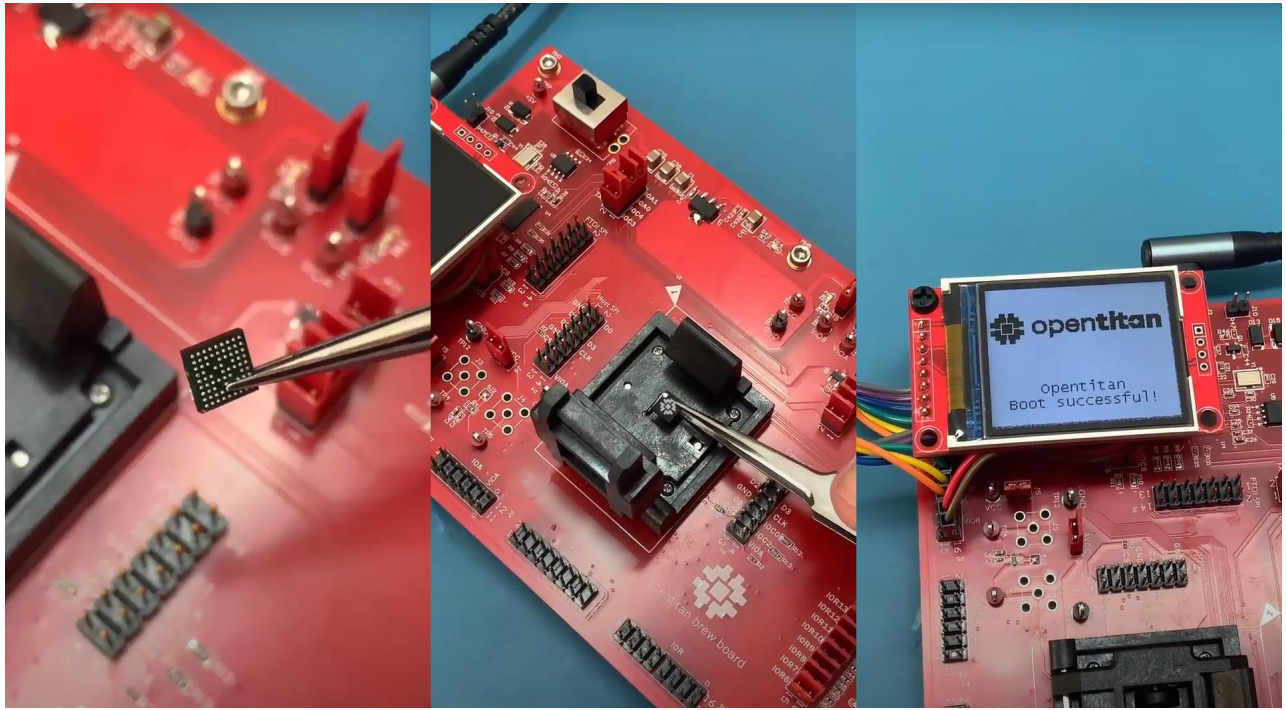
Introduction to OpenTitan®

The OpenTitan® partnership develops, verifies and maintains an ecosystem of high quality - **open source** - chip designs and security IP

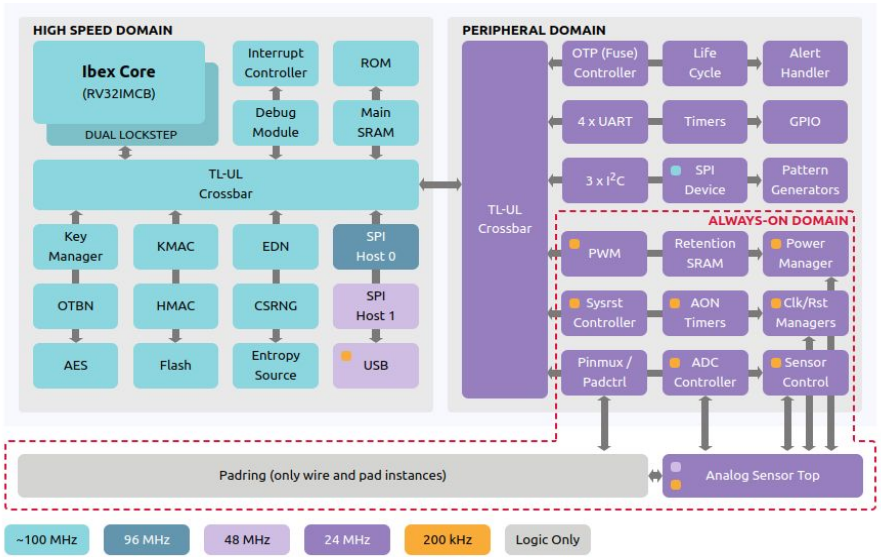
lowRISC · ETH zürich · Giesecke+Devrient · Google · nuvoTon · Western Digital · SEAGATE · Rivos · winbond · zeroRISC

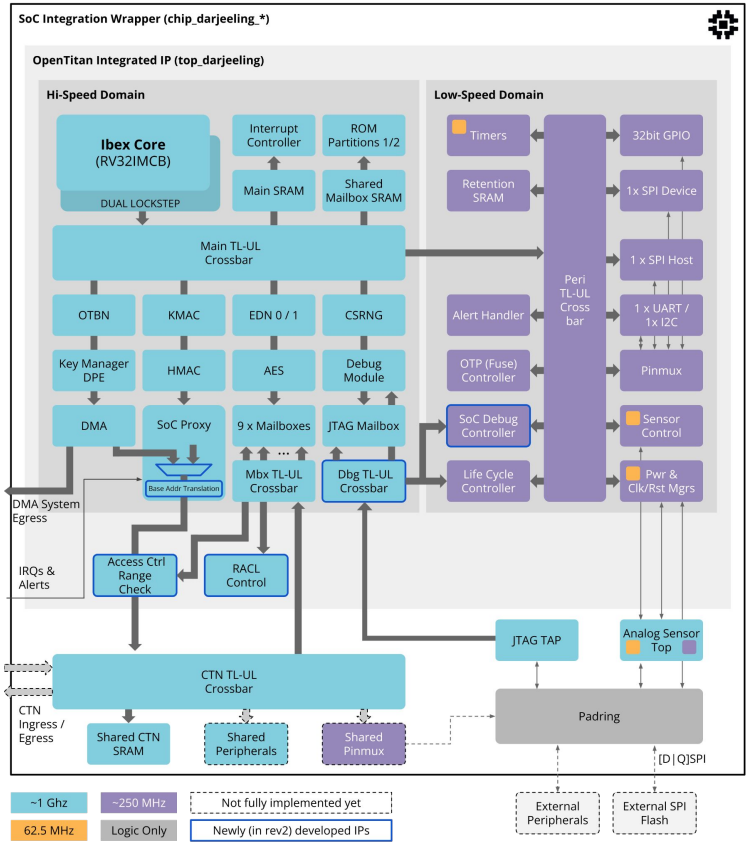


World's First Commercial-Grade Open Source RoT



OpenTitan® — Earl Grey & Darjeeling

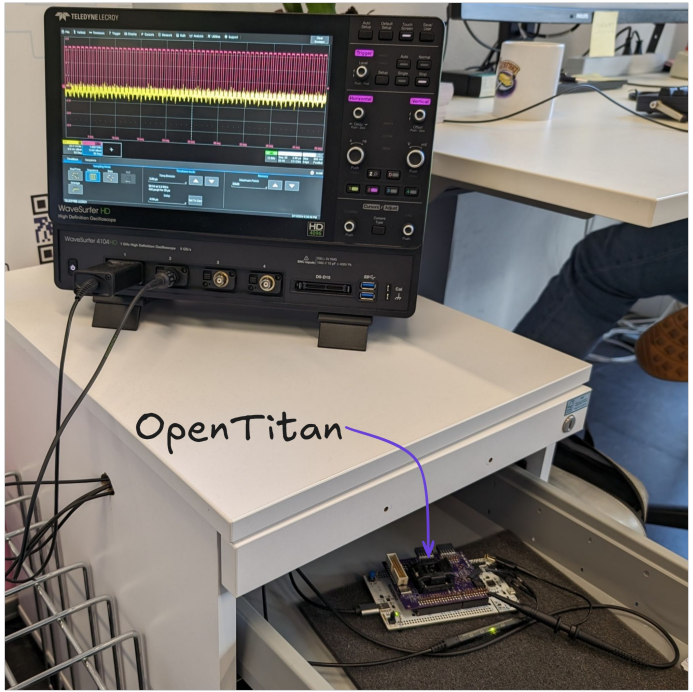




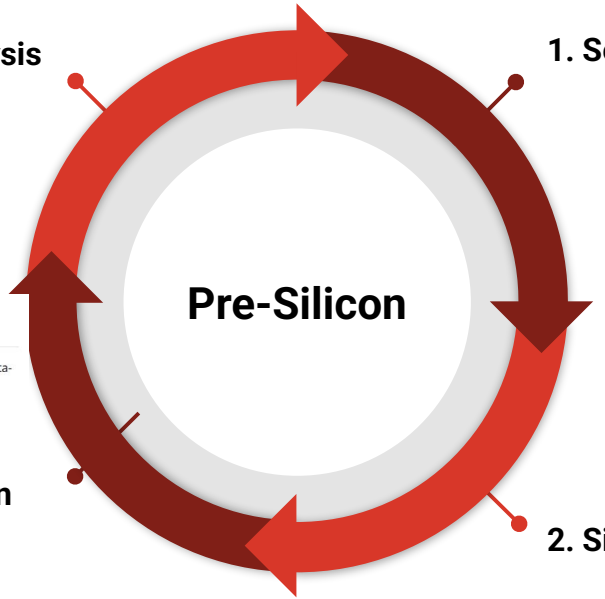
Secure IP Development Cycle

Threat Model

- Attacker with physical access to the chip
- Physical attacks are in scope
  - Fault Injection (FI)
  - Side-Channel Analysis (SCA)
- **Design IP with this threat model in mind**



Secure IP Development Cycle



1. Secure Hardware Development

opentitan

Secure Hardware Design Guidelines

Overview

Silicon designs for security devices require special guidelines to protect the designs against myriad attacks. For OpenTitan, the universe of potential attacks is described in our threat model. In order to have the most robust defensive posture, a general approach to secure hardware design should rely on the concepts of (1) defense in depth, (2) consideration of recovery methods post-breach, and (3) thinking with an attacker mindset.

2. Simulation

PROLEAD

A Probing-Based Hardware Leakage Detection Tool

Nicolai Müller[1] and Amir Moradi[2]

[1] Ruhr University Bochum, Horst Görtz Institute for IT Security, Bochum, Germany
firstname.lastname@rub.de
[2] University of Cologne, Institute for Computer Science, Cologne, Germany
firstname.lastname@uni-koeln.de

Fault-Resistant Partitioning of Secure CPUs for System Co-Verification against Faults

Simon Tollec[1], Vedad Hadžić[2], Pascal Nasahl[2,3], Mihail Asavoae[1], Roderick Bloem[2], Damien Couroussé[4], Karine Heydemann[5,6], Mathieu Jan[1] and Stefan Mangard[2]

[1] Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France, firstname.lastname@cea.fr
[2] Graz University of Technology, Graz, Austria, firstname.lastname@iaik.tugraz.at
[3] lowRISC C.I.C., Cambridge, United Kingdom, nasahlpa@lowrisc.org
[4] Univ. Grenoble Alpes, CEA, List, F-38000, Grenoble, France, firstname.lastname@cea.fr
[5] Thales DIS, Gémenos, France, firstname.lastname@thalesgroup.com
[6] Sorbonne Univ., CNRS, LIP6, F-75005, Paris, France

3. Formal Verification

SYNFI: Pre-Silicon Fault Analysis of an Open-Source Secure Element

Pascal Nasahl[†1,3], Miguel Osorio[1], Pirmin Vogel[2], Michael Schaffner[1], Timothy Trippel[1], Dominic Rizzo[1] and Stefan Mangard[3,4]

[1] Google, Mountain View, USA
[2] lowRISC CIC, Cambridge, United Kingdom
[3] Graz University of Technology, Graz, Austria
firstname.lastname@iaik.tugraz.at
[4] Lamarr Security Research, Graz, Austria

CocoAlma: Execution-aware Masking Verification

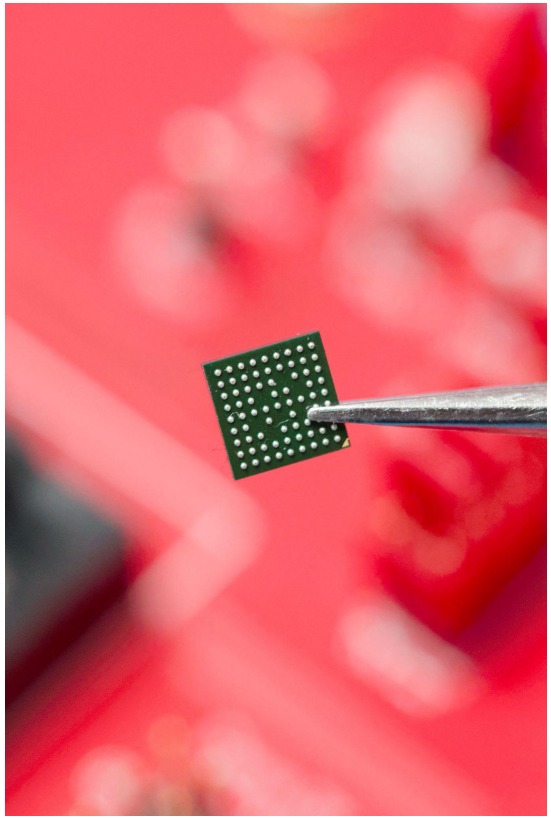
CocoAlma is an *execution-aware* tool for formal verification of masked implementations. It can verify any data-independent masked computation that can be implemented as a Verilog hardware circuit or as software running on a hardware platform, with properly labeled secret shares and randomness.

4. FPGA Analysis

OpenTitan's Security Testing Framework
Next slides!

Post-Silicon Analysis

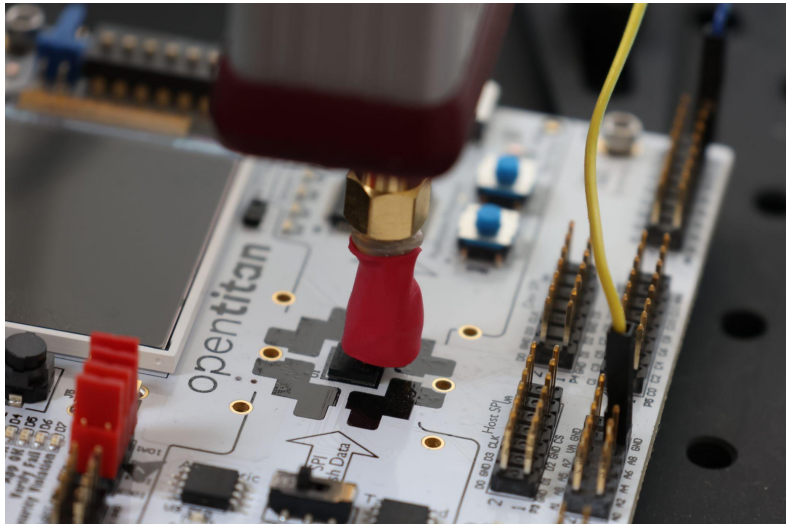
- Final step
- Covers real-world setting
  - With analog countermeasures, e.g., clock jitter
  - Whole chip instead of isolated IP
  - More noise
- Learnings influence software guidance & future chip generations
- OpenTitan’s Security Testing Framework



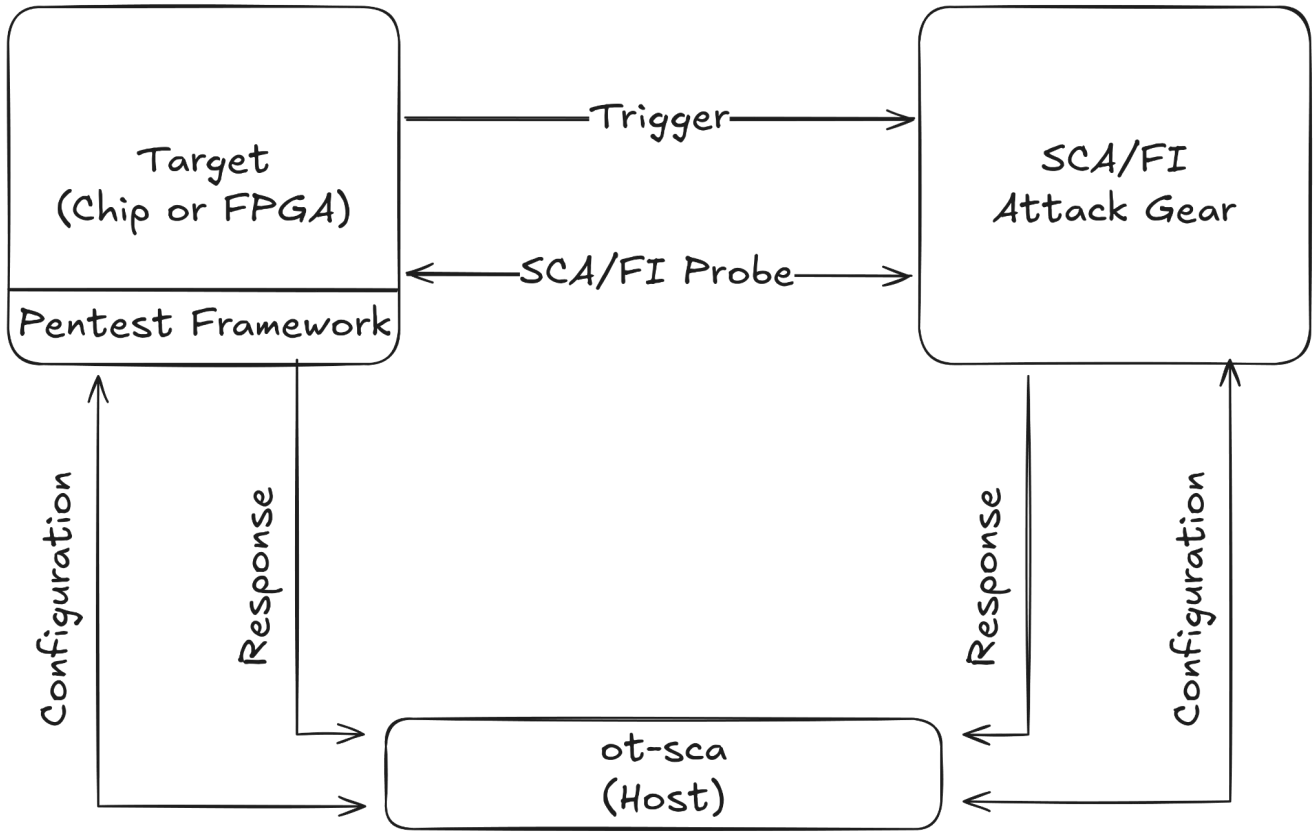
OpenTitan's Security Testing Framework

Security Testing Framework

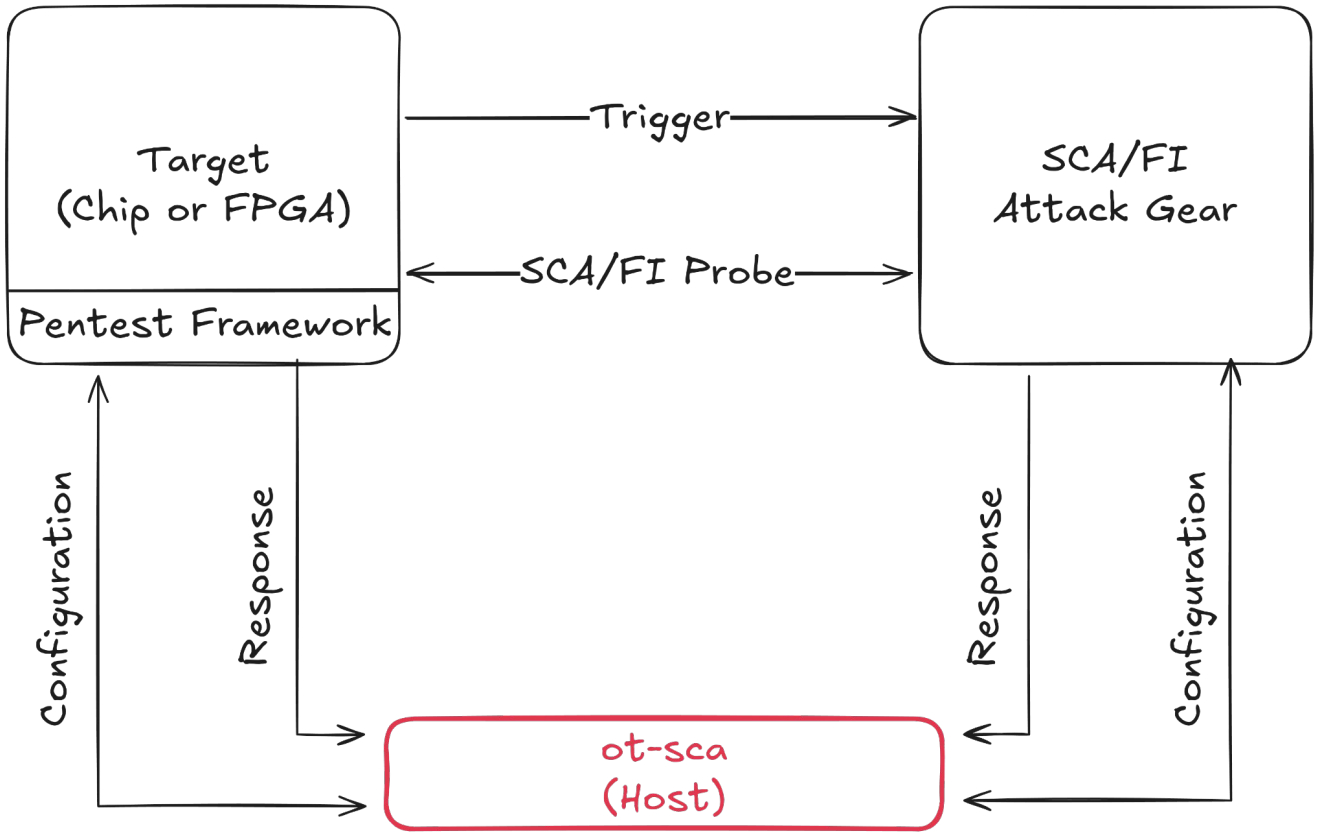
- Key component in secure IP development
  - Pre-silicon: FPGA
  - Post-silicon: Chip
- Collaborative platform for internal and external partners
  - OpenTitan developers
  - External labs
  - Certification body
- **Open-source**



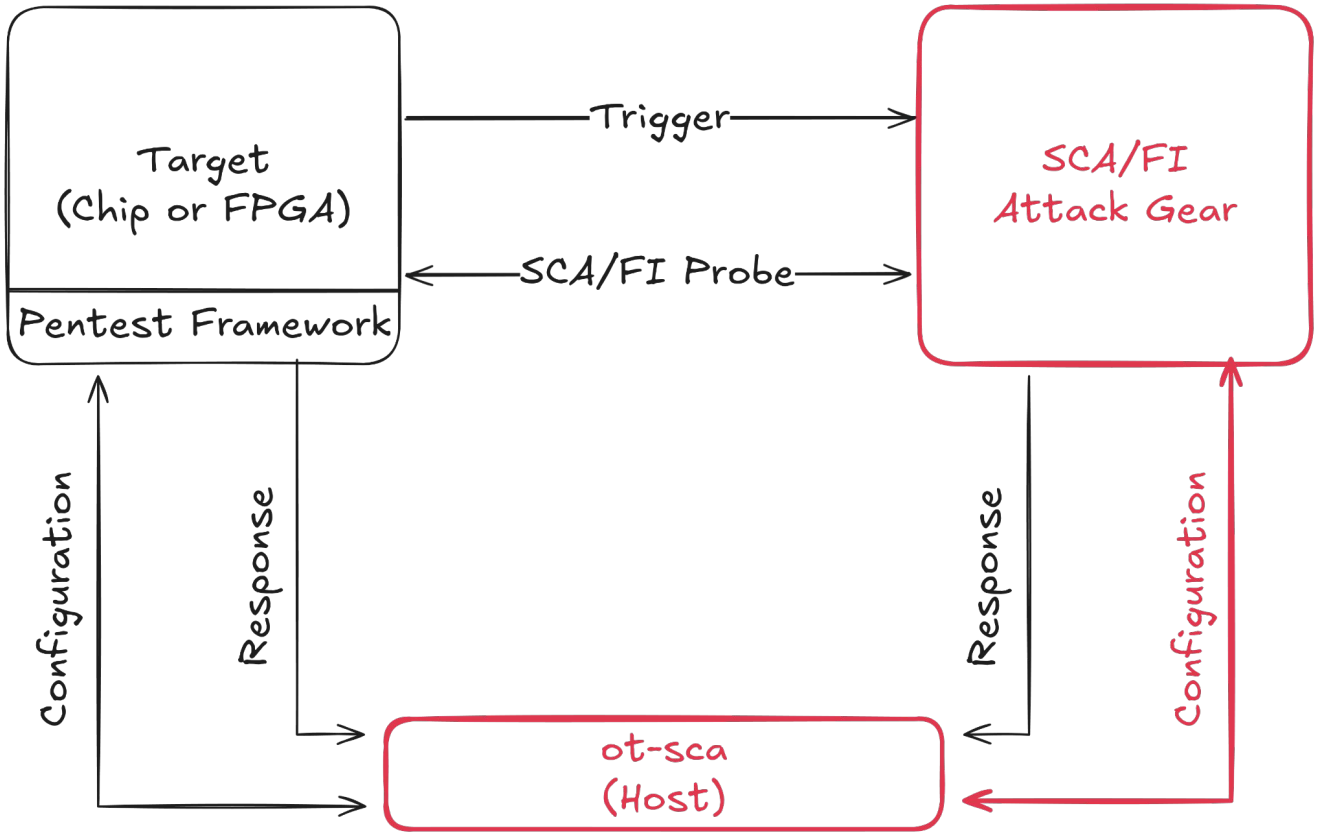
Security Testing Framework



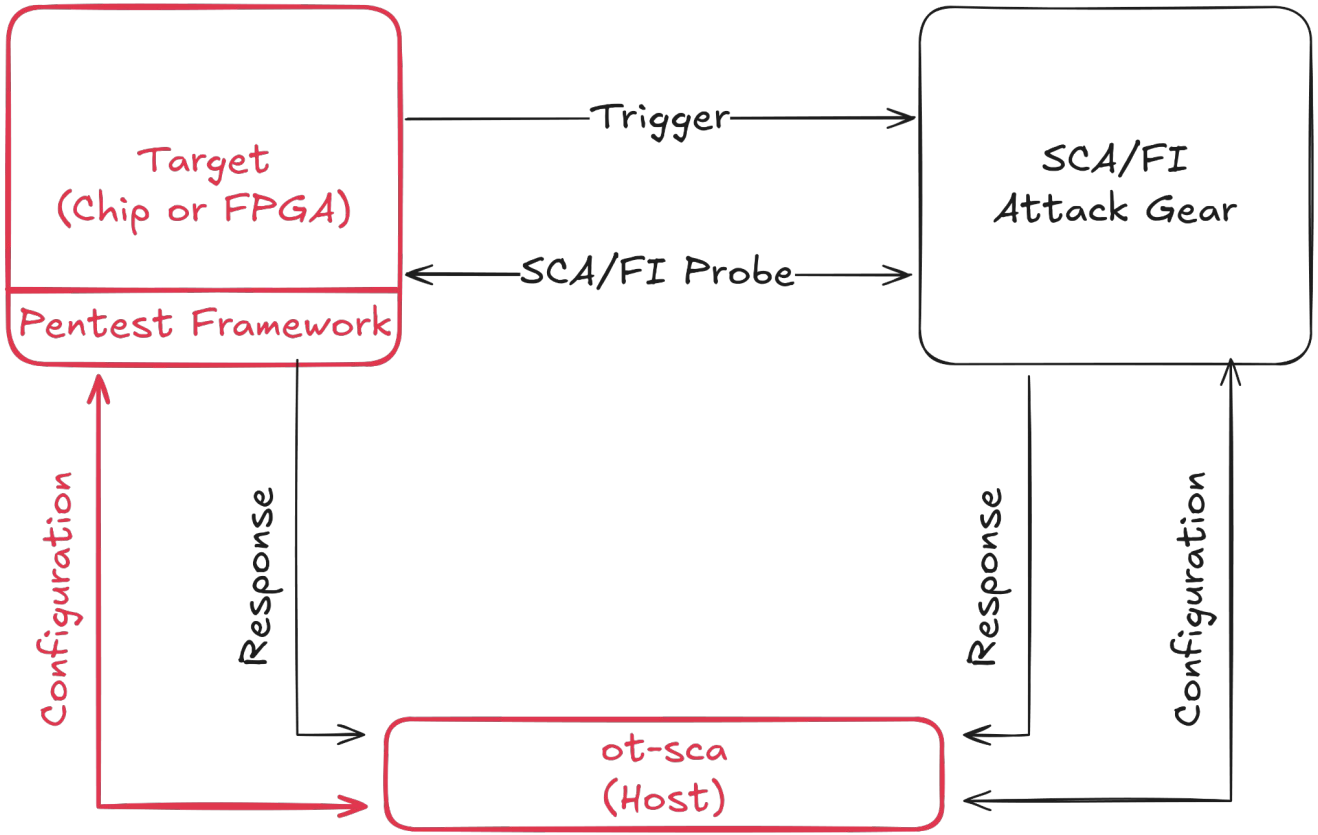
Security Testing Framework



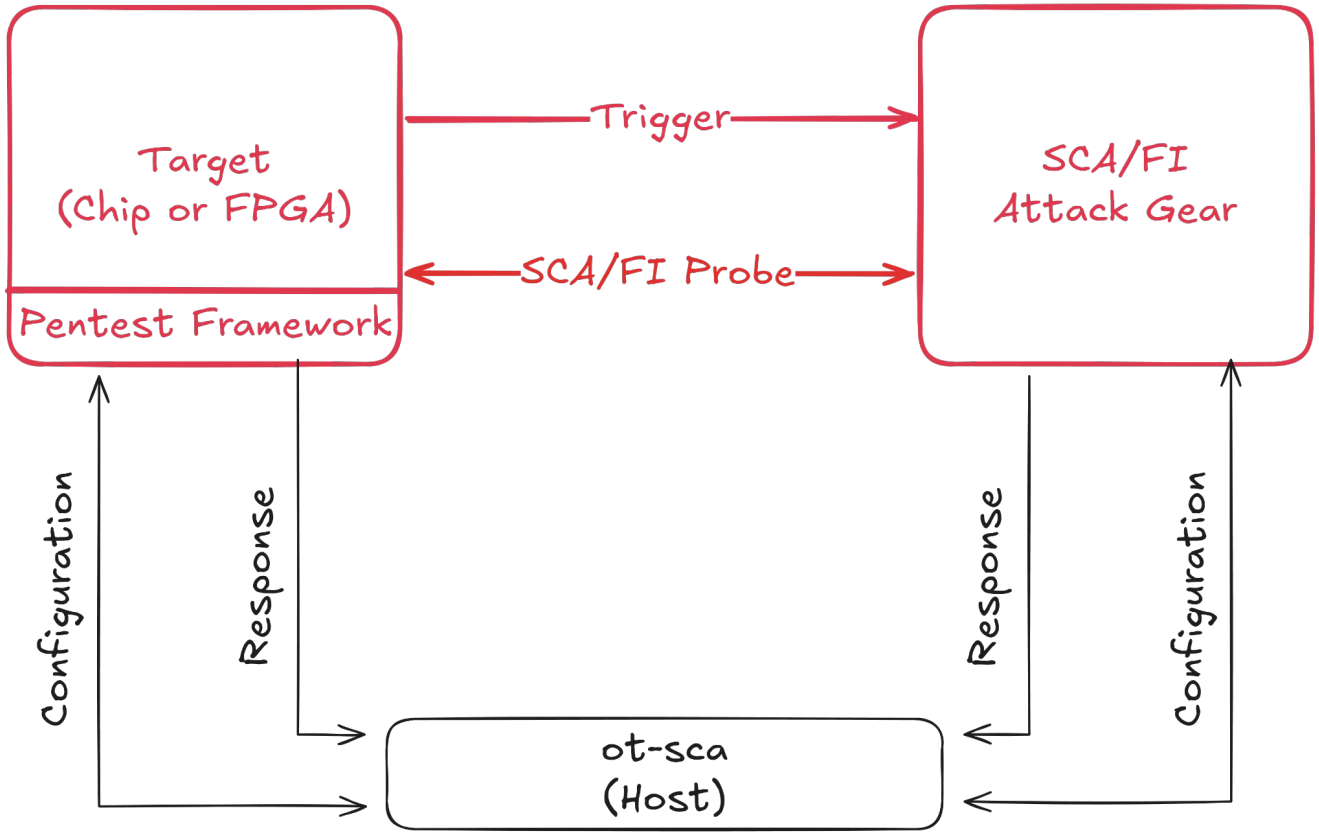
Security Testing Framework



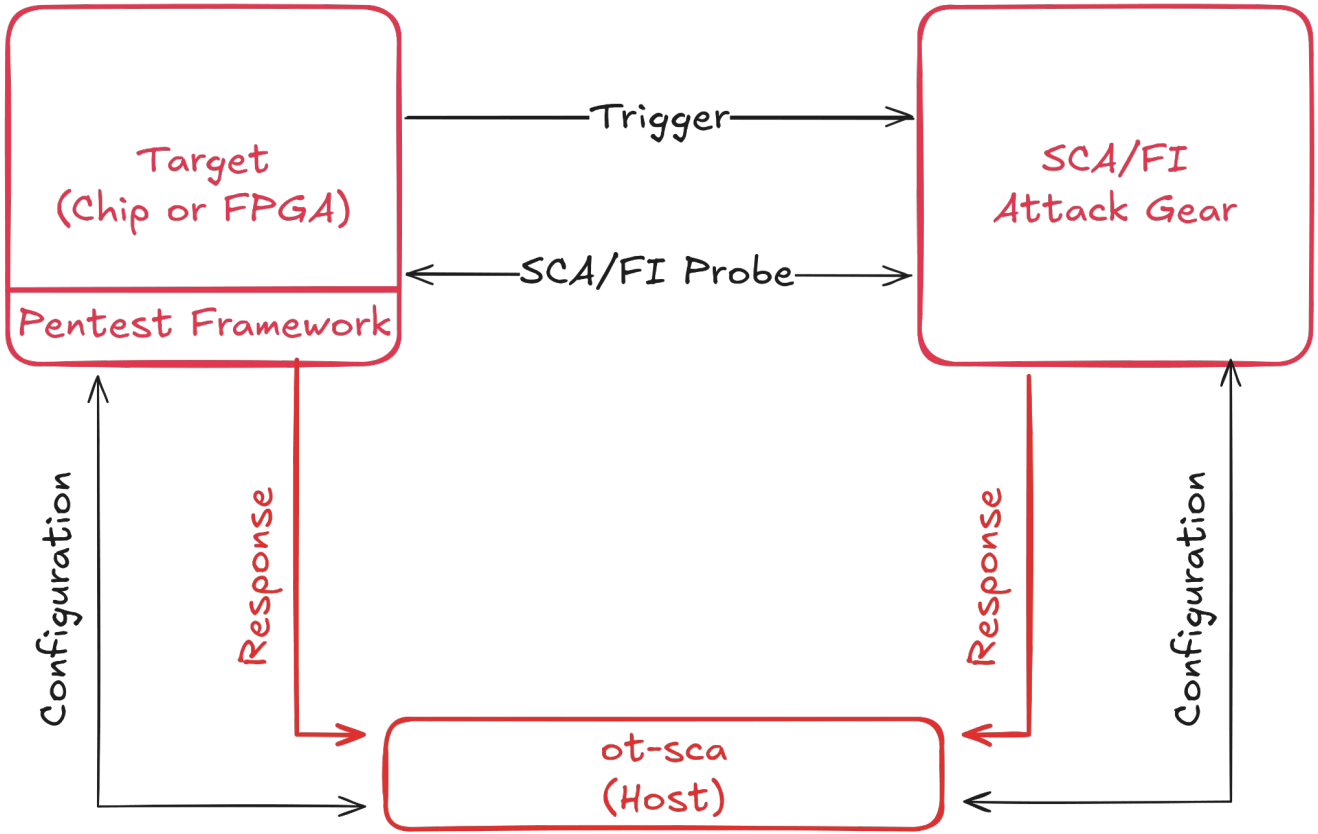
Security Testing Framework



Security Testing Framework

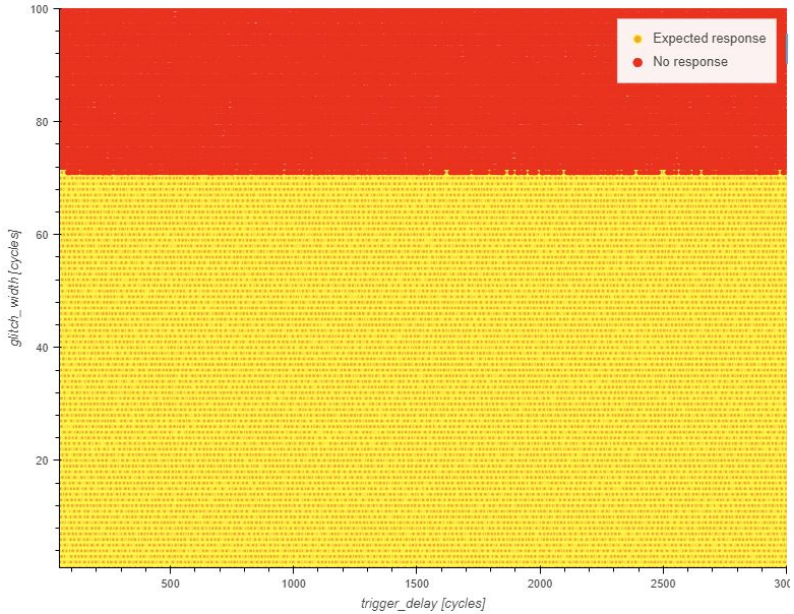


Security Testing Framework



OT-SCA Host Framework

- Coordinates SCA and FI evaluations
  - Configures equipment and target
  - Collects evaluation results
- Features:
  - Evaluation database
  - Fault parameter sweep
  - Batch mode for high SCA capture rates
  - Result visualization
  - Trace alignment
  - Analysis scripts (TVLA, …)
  - …
- Communication API is standalone to allow integration into its own framework



Pentest Device Framework

- Comprehensive SCA and FI evaluation framework
- >230 tests exercise the entirety of OpenTitan

Characterization Tests:

```
// FI code target.
PENTEST_ASM_TRIGGER_HIGH
asm volatile(NOP10 "beq x5, x6, correctbeq\n" NOP10
                   "j badbeq\n"
                   "correctbeq:\n"
                   "addi x5, x5, 0x11\n"
                   "addi x6, x6, 0x22\n"
                   "badbeq:\n");
PENTEST_ASM_TRIGGER_LOW
```

CryptoLib Tests:

```
// Trigger window.
pentest_set_trigger_high();
TRY(otcrypto_aes(&key, iv, mode, op, input, padding, output));
pentest_set_trigger_low();
```

Pentest Code Structure

```
status_t handle_ibex_fi_register_file(ujson_t *uj) {
  crypto_fi_ibex_register_file_t uj_input;
  TRY(ujson_deserialize_ibex_register_file_t(uj, &uj_input));

  pentest_init(uj_input);
  INIT_REGISTER_FILE

  PENTEST_ASM_TRIGGER_HIGH
  asm volatile(NOP1000);
  PENTEST_ASM_TRIGGER_LOW;

  reg_alerts = pentest_get_triggered_alerts();
  pentest_sensor_alerts_t sensor_alerts = pentest_get_sensor_alerts();

  ibex_rf_content_t rf = DUMP_REGISTER_FILE
  ibex_fi_faulty_data_t uj_output = check_rf_content(rf);

  RESP_OK(ujson_serialize_ibex_fi_faulty_data_t, uj, &uj_output);
  return OK_STATUS();
}
```

Receive test config from ot-sca

Test preparation

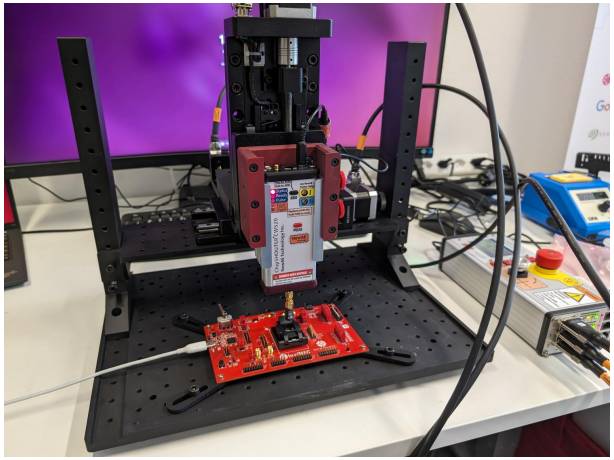
Trigger window

Test evaluation

Send evaluation result to ot-sca

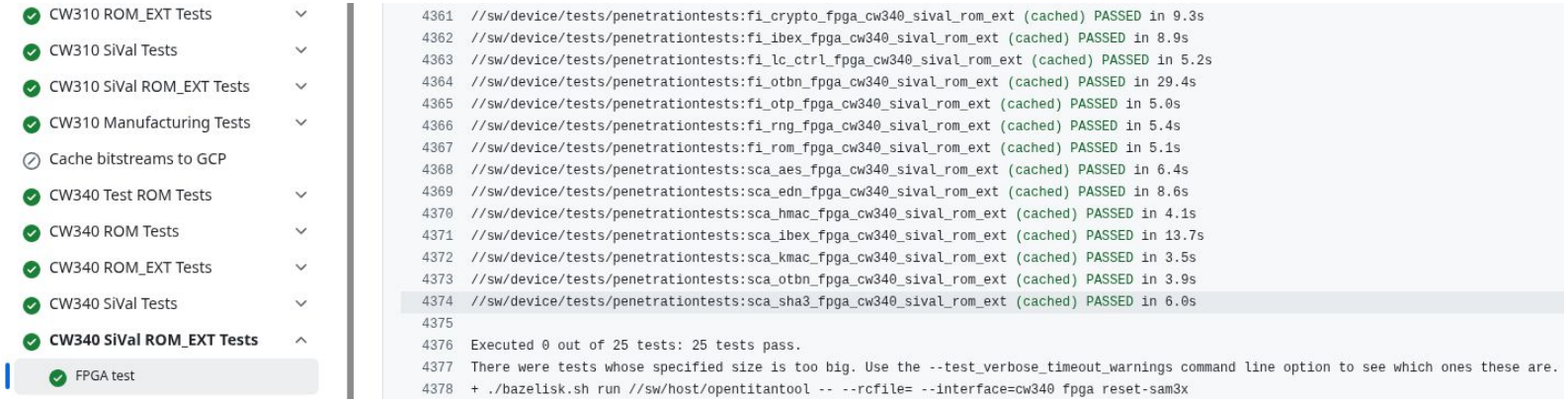
Supported Testing Equipment

- FI
  - Voltage glitching: CW Husky Crowbar
  - EMFI: ChipShouter + ChipShover XYZ table
- SCA
  - ChipWhisperer Husky scope
  - Scopes with VX11 support (tested with LeCroy oscilloscopes)
- **Easy to add new equipment by using driver classes**



Testing the Pentesting Framework

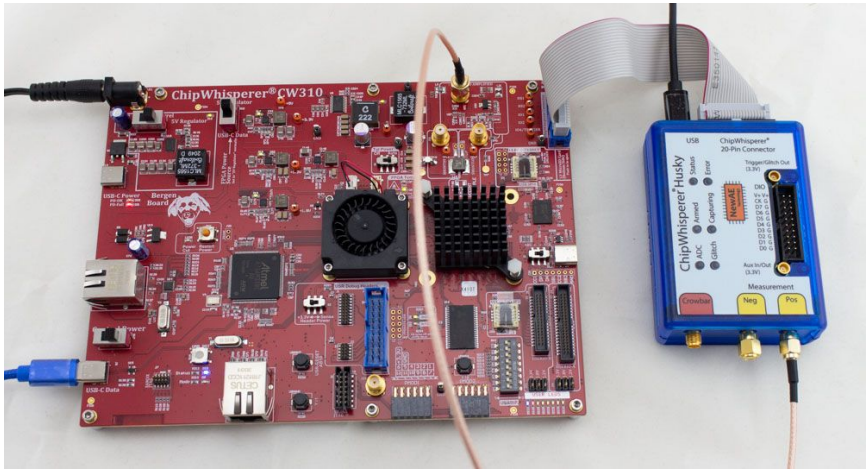
- **It needs to work**
  - Contributions by different organizations
  - Framework used by different internal and external OpenTitan partners
- Compare pentesting results to test vectors on silicon and FPGA
- Integrated into Continuous Integration (CI) pipeline of the OpenTitan repository



Getting Started

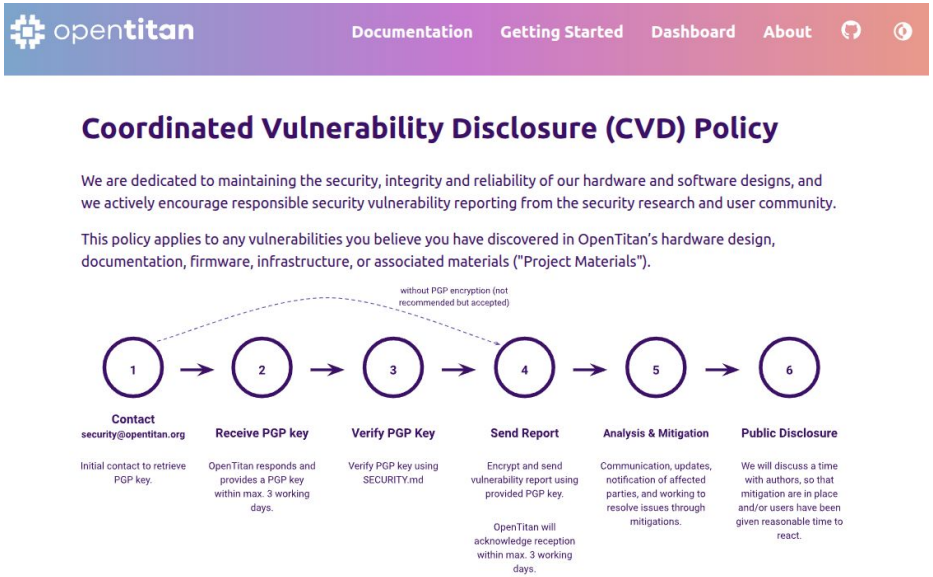
- Manual available
  - github.com/lowRISC/ot-sca
- Required Hardware
  - NewAE ChipWhisperer CW310 + Husky
  - NewAE testing equipment
  - Or own attack gear

```
$ git clone git@github.com:lowRISC/ot-sca.git
$ cd ot-sca
$ pip install -r python-requirements.txt
$ cd capture/
$ ./capture_aes.py -c configs/aes_sca_cw310.yaml -p aes
```



Call for Action

- Look into OpenTitan
- Start pentesting it
- Please follow the CVD process:
  - opentitan.org/cvd-policy
- More questions?
  info@lowrisc.org



Thank you!