Integrated Development Environment for Secure Digital Hardware Design

Omar Assad, Hamza Haydar, Ben Krett, Joey Ah-kiow
Academic Advisor: Dr. Benjamin Tan
Industry Sponsor: Intel Corporation

Problem

Cybersecurity has emerged as a major concern with the increased applications and interconnectivity of computer systems. Notorious attacks, such as the one on Colonial pipeline[1], have demonstrated the disastrous consequences of inadequate security.

THE WALL STREET JOURNAL.

Subscribe | Sig

English Edition ▾ | Print Edition | Video | Audio | Latest Headlines | More ▾

Home World U.S. Politics Economy Business Tech Markets Opinion Books & Arts Real Estate Life & Work Style Sports Search

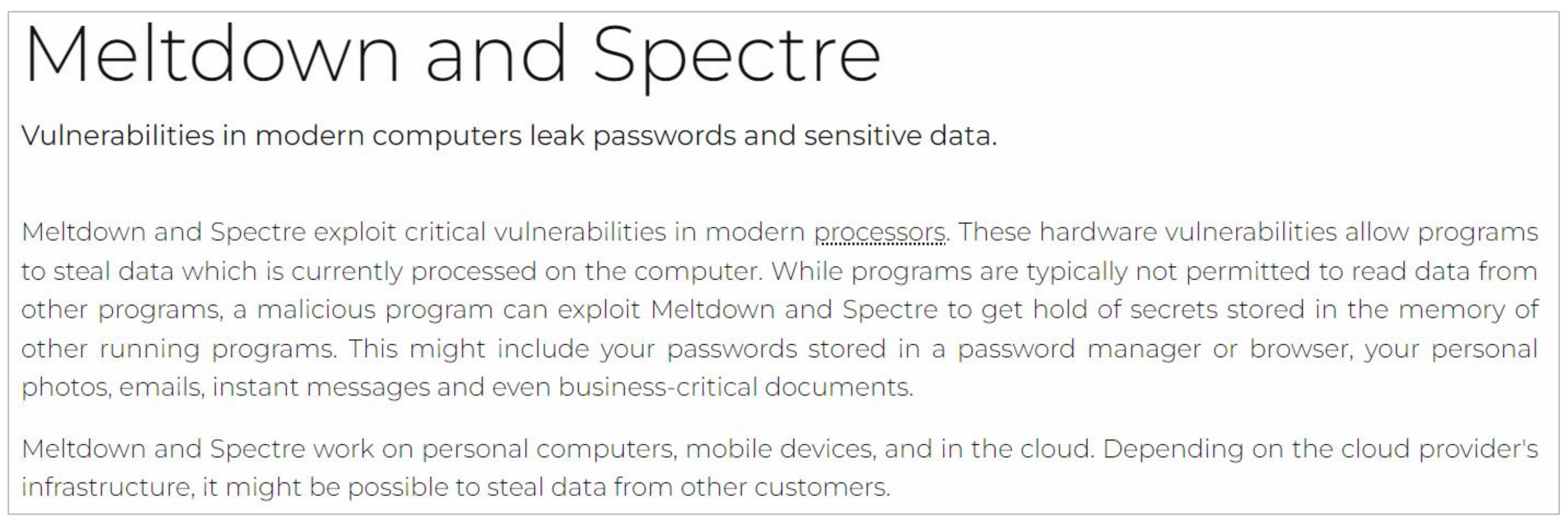
PRO CYBER NEWS

The Log4j Vulnerability: Millions of Attempts Made Per Hour to Exploit Software Flaw

Hundreds of millions of devices are at risk, U.S. officials say; hackers could use the bug to steal data, install malware or take control

Figure 1: The Wall Street Journal Headline[2]

One way these attacks can be orchestrated is through exploiting vulnerabilities -- flaws in the design of the system itself; there were more than 25,000 CVEs (publicly disclosed vulnerabilities) in 2022. Cybersecurity efforts have largely focused on detecting and remediating software vulnerabilities. However, recent research has shown that security vulnerabilities in hardware, such as processors, can also be exploited.


Meltdown and Spectre

Vulnerabilities in modern computers leak passwords and sensitive data.

Meltdown and Spectre exploit critical vulnerabilities in modern processors. These hardware vulnerabilities allow programs to steal data which is currently processed on the computer. While programs are typically not permitted to read data from other programs, a malicious program can exploit Meltdown and Spectre to get hold of secrets stored in the memory of other running programs. This might include your passwords stored in a password manager or browser, your personal photos, emails, instant messages and even business-critical documents.

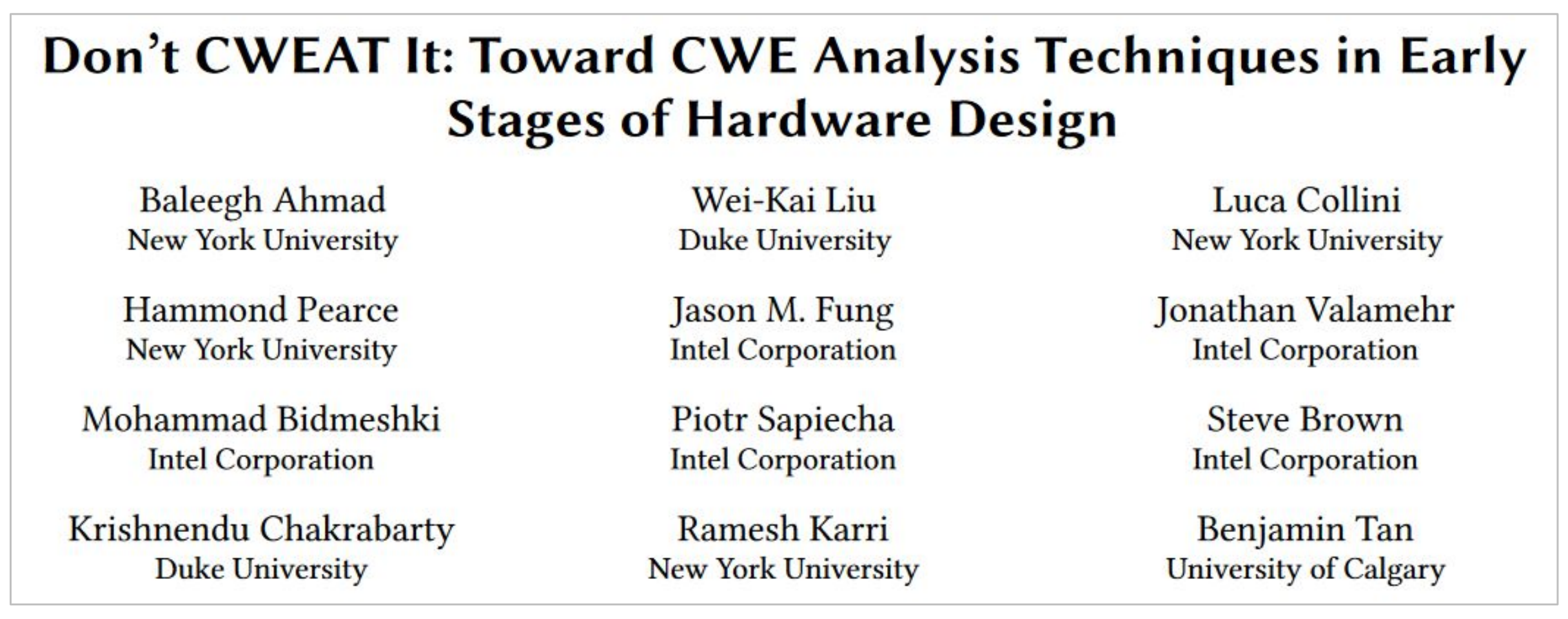
Meltdown and Spectre work on personal computers, mobile devices, and in the cloud. Depending on the cloud provider's infrastructure, it might be possible to steal data from other customers.

Figure 2: Meltdown and Spectre Hardware Vulnerability Description[3]

The tools and techniques available to detect these issues in hardware require expertise in hardware design and cybersecurity, a rare combination; How can we assist hardware designers in catching these issues early?

Motivation

A recent research paper has showcased that "Static Analysis" can be effective in detecting these "security bugs" in hardware designs. These scanners are attractive options because they can potentially be applied at earlier stages of design, require no security expertise to use, and are fully automated. Our project's motivation is to create an development environment which can integrate these scanners to enable their use, and maximize their value and effectiveness.


Don't CWEAT It: Toward CWE Analysis Techniques in Early Stages of Hardware Design

Baleegh Ahmad, New York University
Wei-Kai Liu, Duke University
Luca Collini, New York University
Hammond Pearce, New York University
Jason M. Fung, Intel Corporation
Jonathan Valamehr, Intel Corporation
Mohammad Bidmeshki, Intel Corporation
Piotr Sapiecha, Intel Corporation
Steve Brown, Intel Corporation
Krishnendu Chakrabarty, Duke University
Ramesh Karri, New York University
Benjamin Tan, University of Calgary

Figure 3: Research Paper Title and Authors[4]

Our three major objectives are:

1. ability to integrate and use scanners
2. ability for users to provide "context" to maximize scanning effectiveness
3. ability to display detected issues intuitively

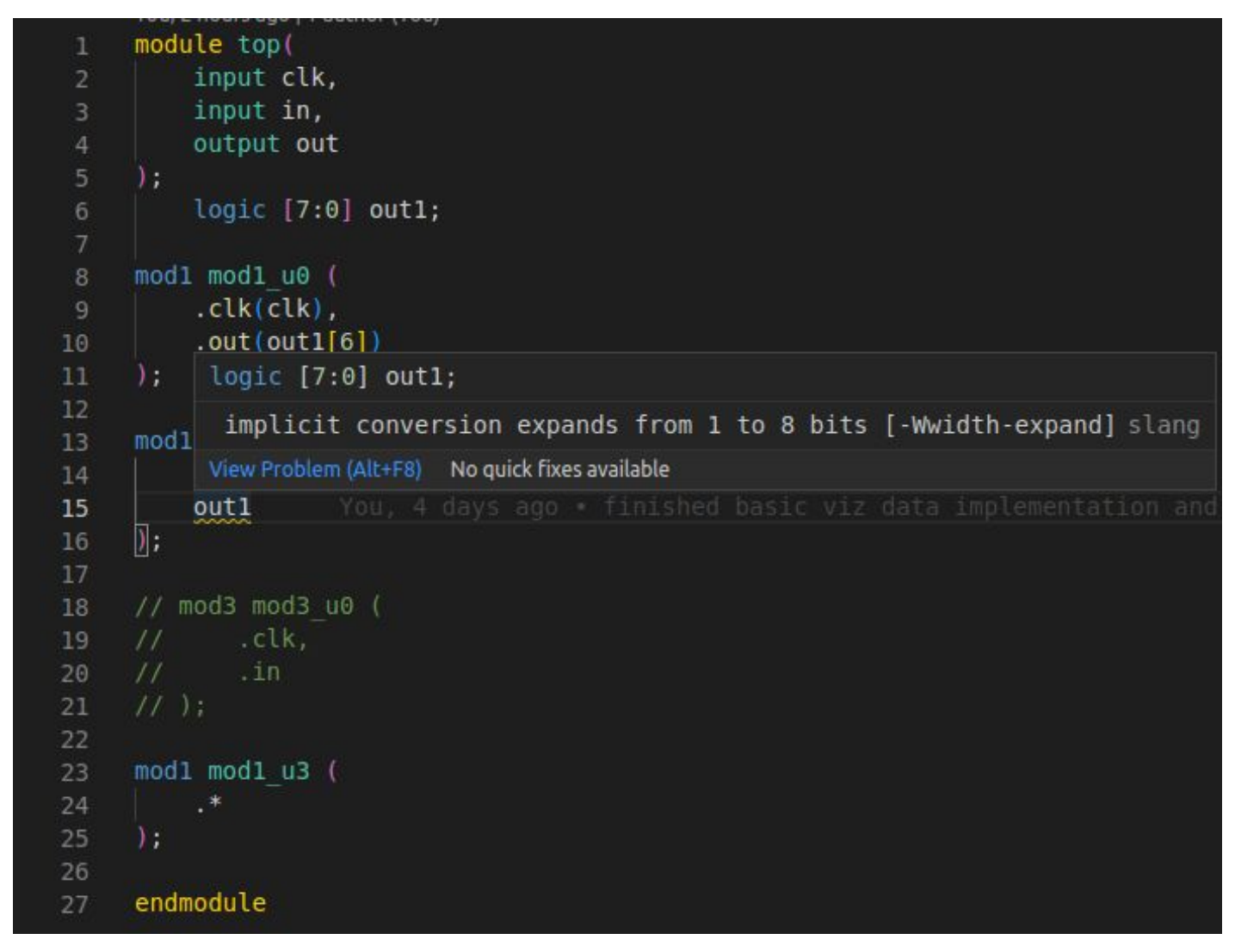
Solution & Initial Results

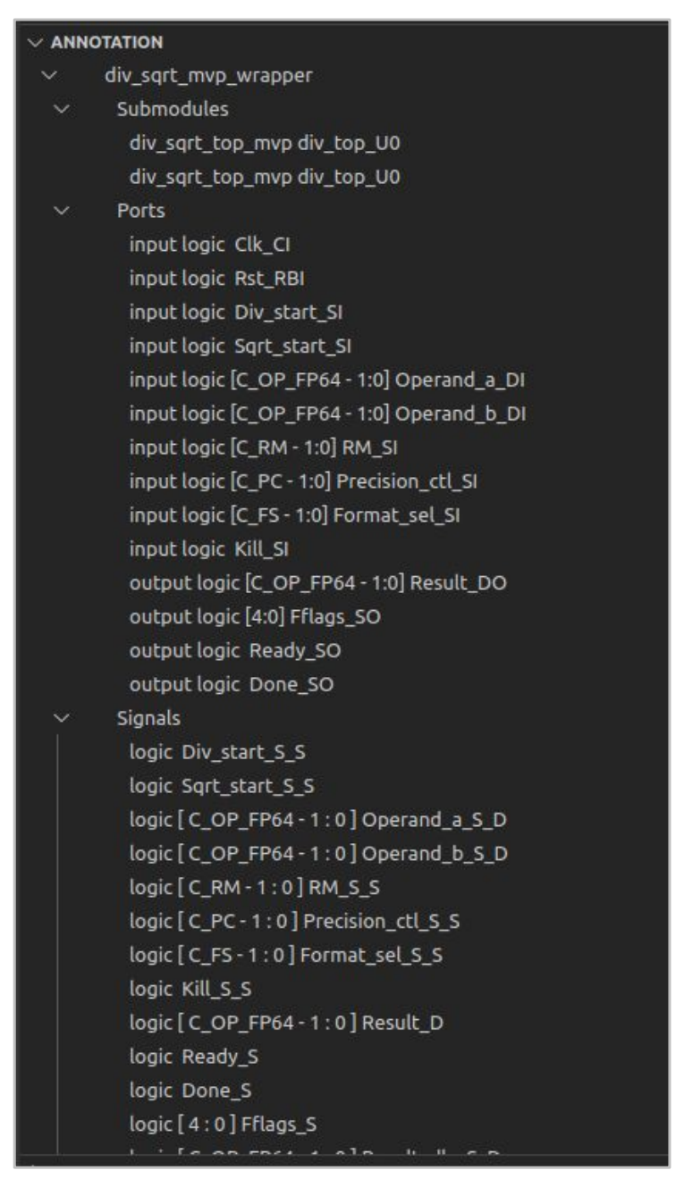
Our solution consists of a Visual Studio (VS) Code Extension. VS Code was chosen for its popularity (74.47% of respondents of the 2022 Stack Overflow survey[5] (71,010 responses) used VS Code as their primary code editor). It also offered the most powerful feature set and documentation for extension development.

Our motivation translated into three main features:

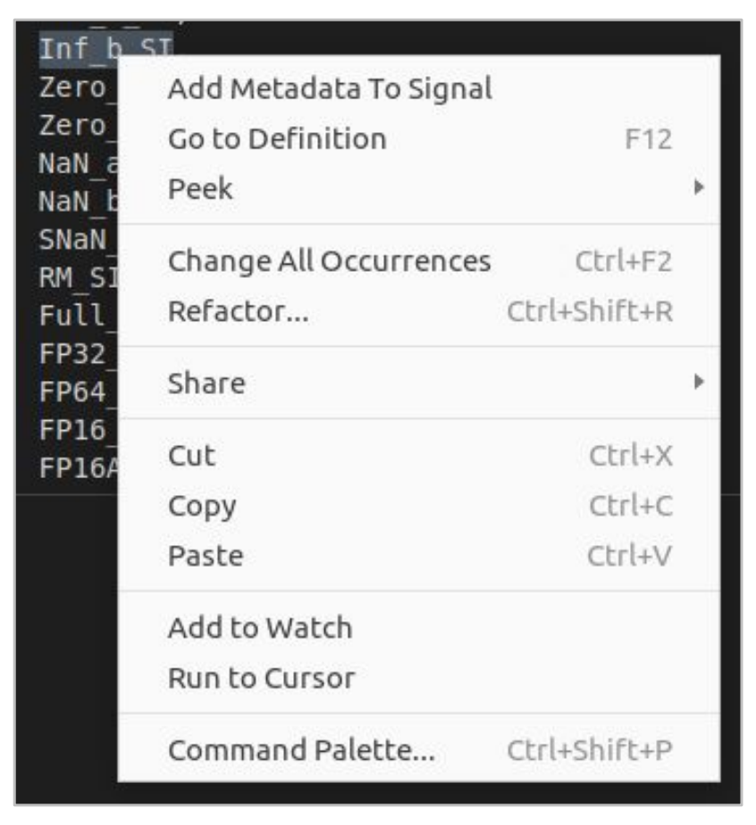
1. Process RTL Project:
   a. Project configuration
   b. SystemVerilog language support
   c. Hierarchical module tree view
2. Design annotation:
   a. UI for manual user annotations
   b. back-end to store data
3. Security feedback:
   a. easy addition/removal of scanners
   b. standardized and configurable interface for scanners
   c. design visualizations

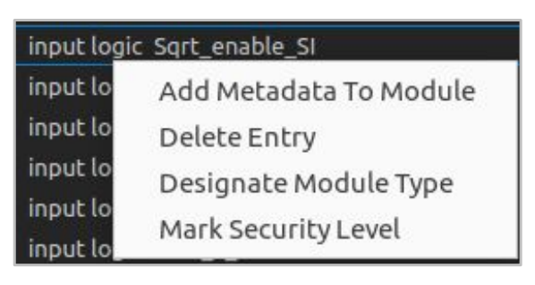
1. Process RTL Project:

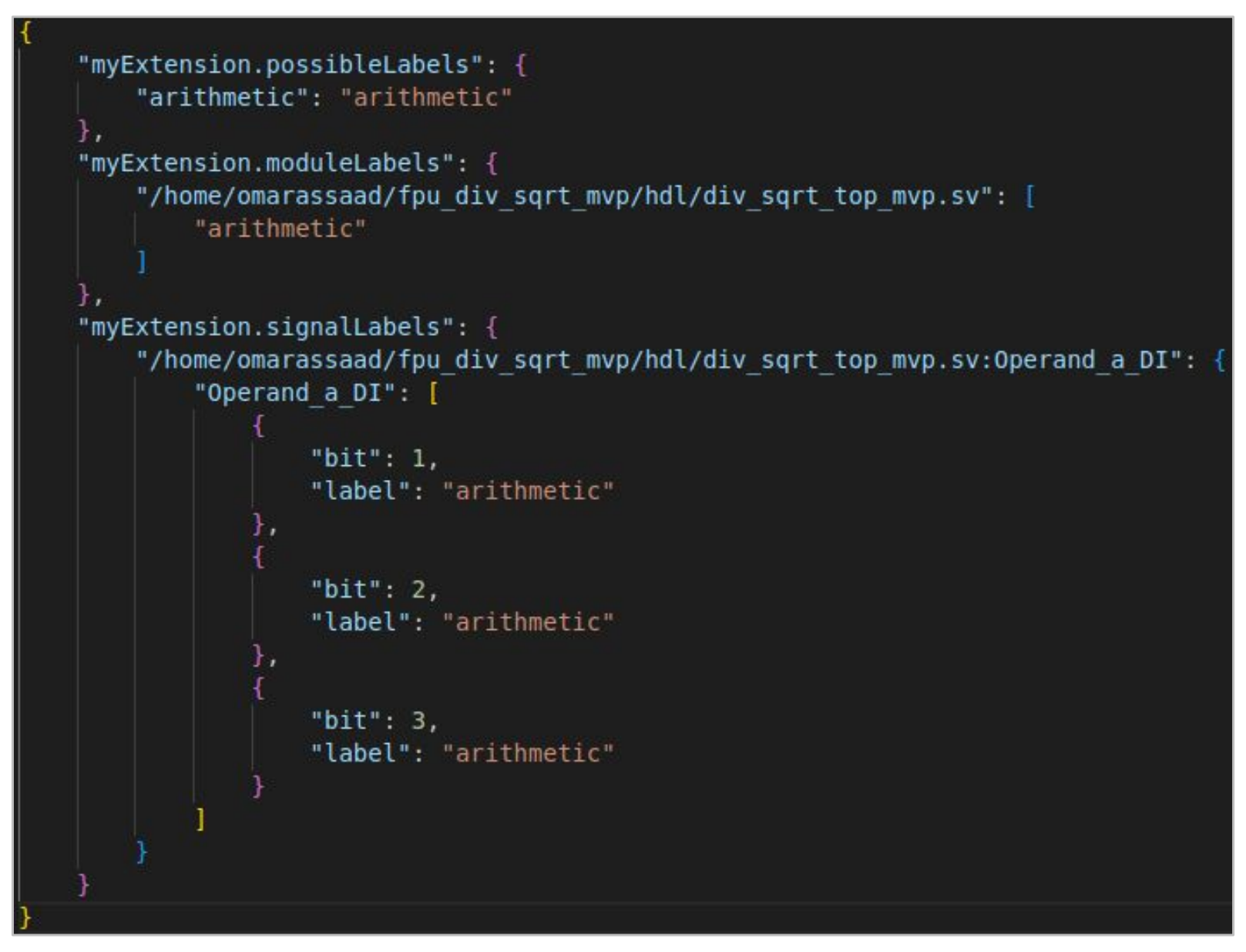

```
module top(
    input clk,
    input in,
    output out
);
    logic [7:0] out1;

mod1 mod1_u0 (
    .clk(clk),
    .out(out1[6])
);  logic [7:0] out1;
     implicit conversion expands from 1 to 8 bits [-Wwidth-expand] slang
mod1 View Problem (Alt+F8)    No quick fixes available
    
    out1        You, 4 days ago • finished basic viz data implementation and
);

// mod3 mod3_u0 (
//      .clk,
//      .in
// );

mod1 mod1_u3 (
    .*
);

endmodule
```


```
ANNOTATION
  div_sqrt_mvp_wrapper
    Submodules
      div_sqrt_top_mvp div_top_U0
      div_sqrt_top_mvp div_top_U0
    Ports
      input logic Clk_CI
      input logic Rst_RBI
      input logic Div_start_SI
      input logic Sqrt_start_SI
      input logic [C_OP_FP64 - 1:0] Operand_a_DI
      input logic [C_OP_FP64 - 1:0] Operand_b_DI
      input logic [C_RM - 1:0] RM_SI
      input logic [C_PC - 1:0] Precision_ctl_SI
      input logic [C_FS - 1:0] Format_sel_SI
      input logic Kill_SI
      output logic [C_OP_FP64 - 1:0] Result_DO
      output logic [4:0] Fflags_SO
      output logic Ready_SO
      output logic Done_SO
    Signals
      logic Div_start_S_S
      logic Sqrt_start_S_S
      logic [ C_OP_FP64 - 1 : 0 ] Operand_a_S_D
      logic [ C_OP_FP64 - 1 : 0 ] Operand_b_S_D
      logic [ C_RM - 1 : 0 ] RM_S_S
      logic [ C_PC - 1 : 0 ] Precision_ctl_S_S
      logic [ C_FS - 1 : 0 ] Format_sel_S_S
      logic Kill_S_S
      logic [ C_OP_FP64 - 1 : 0 ] Result_D
      logic Ready_S
      logic Done_S
      logic [ 4 : 0 ] Fflags_S
```

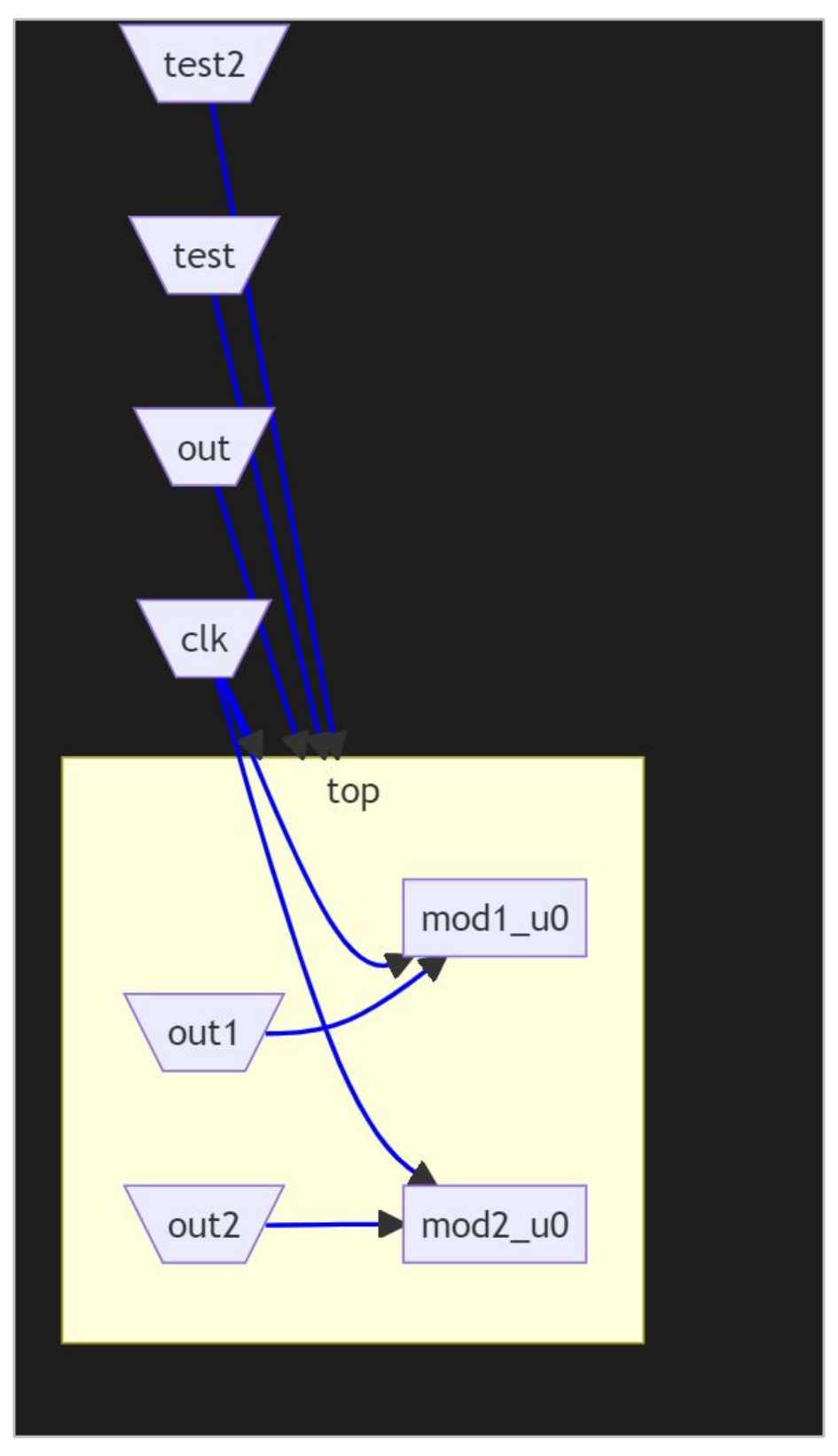
2. Design annotation:

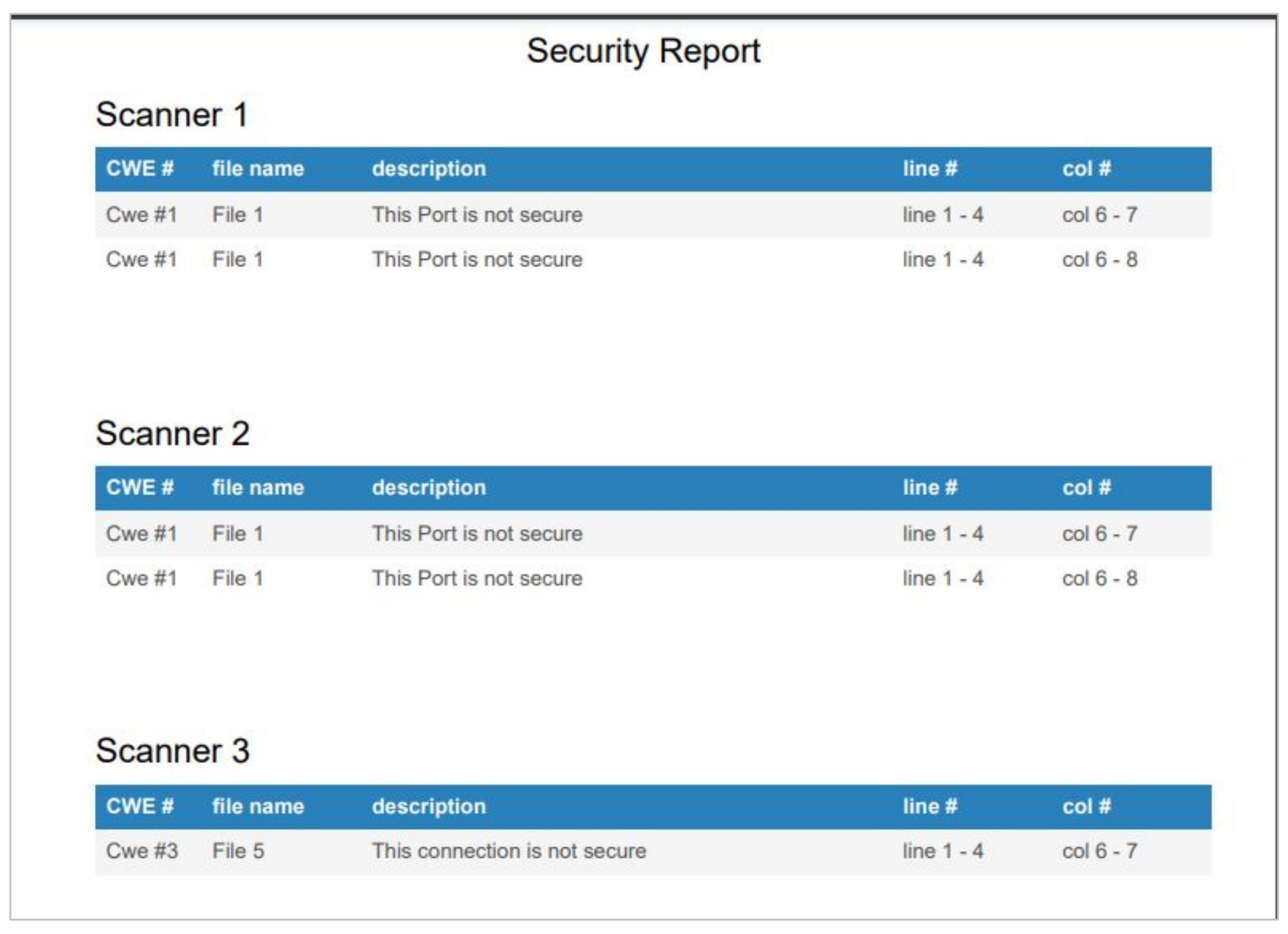





```
{
    "myExtension.possibleLabels": {
        "arithmetic": "arithmetic"
    },
    "myExtension.moduleLabels": {
        "/home/omarassaad/fpu_div_sqrt_mvp/hdl/div_sqrt_top_mvp.sv": [
            "arithmetic"
        ]
    },
    "myExtension.signalLabels": {
        "/home/omarassaad/fpu_div_sqrt_mvp/hdl/div_sqrt_top_mvp.sv:Operand_a_DI": {
            "Operand_a_DI": [
                {
                    "bit": 1,
                    "label": "arithmetic"
                },
                {
                    "bit": 2,
                    "label": "arithmetic"
                },
                {
                    "bit": 3,
                    "label": "arithmetic"
                }
            ]
        }
    }
}
```

3. Security feedback:




Security Report

Scanner 1

CWE #	file name	description	line #	col #
Cwe #1	File 1	This Port is not secure	line 1 - 4	col 6 - 7
Cwe #1	File 1	This Port is not secure	line 1 - 4	col 6 - 8

Scanner 2

CWE #	file name	description	line #	col #
Cwe #1	File 1	This Port is not secure	line 1 - 4	col 6 - 7
Cwe #1	File 1	This Port is not secure	line 1 - 4	col 6 - 8

Scanner 3

CWE #	file name	description	line #	col #
Cwe #3	File 5	This connection is not secure	line 1 - 4	col 6 - 7

1. https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password?leadSource=uverify%20wall
2. https://www.wsj.com/articles/what-is-the-log4j-vulnerability-11639446180
3. https://meltdownattack.com/
4. https://doi.org/10.1145/3508352.3549369
5. https://survey.stackoverflow.co/2022/#section-most-popular-technologies-integrated-development-environment