JTAG Security and Trust

Mohammad Tehranipoor
ECE4095: Hardware Security & Trust
University of Connecticut
ECE Department

Agenda

- Introduction to JTAG
- High-Level JTAG Exploits
- Popular JTAG Exploits
- Security Options

25 December 2012

JTAG Introduction

- “JTAG” refers to IEEE Std. 1149.1, Standard Test Access Port and Boundary Scan Architecture
- IEEE Std. 1532, Boundary-Scan-Based In-System Configuration of Programmable Devices

Goals/Benefits of JTAG

- Low Cost
- Inter-circuit testing without need of physical test-probes
- Increased fault detection coverage
- Lower test time

Physical Components

- TAP
  - Test Access Port
  - Interprets JTAG protocol
  - Controlled by TMS signal
- BSR
  - Boundary Scan Registers
  - Between module and TAP

JTAG Control

- TDO
  - Test Data Output
- TDI
  - Test Data Input
- TMS
  - Test Mode Select
- TCK
  - Test Clock
- TRST
  - Resets TAP Controller
JTAG Modes

- **Bypass**
  - Connects TDI to TDO
  - One cycle delay

- **ExTest**
  - Asserts data on output pins

- **InTest**
  - Reads data from input pins

JTAG Overview

- **JTAG Benefits**
  - Low Cost
  - Ease of testing

- **Physical Components**
  - TAP, BSR

- **JTAG Pins**
  - TDI, TDO, TMS, TCK, TRST

- **JTAG Modes**
  - Bypass, ExTest, InTest

High-Level JTAG Exploits

- **Sniff TDI/TDO signals**
  - Used to intercept secrets being sent to or from a chip
  - Preceding or chip after victim chip behaves differently during bypass to intercept message

- **Modify TDI/TDO signals**

- **Control TMS and TCK signals**

Sniff TDI/TDO Signals

- Used to intercept secrets being sent to or from a chip
- Preceding or chip after victim chip behaves differently during bypass to intercept message

Modify TDI/TDO Signals

- Can modify Test Vectors and Test Responses
- Can be used to fake correct or false tests
- Attacker can either be upstream or downstream of victim based on attack

Control TMS and TCK Signals

- For many exploits, TMS and TCK signals need to be controlled
- Attacker needs to be able to overpower The signal sent by TAP
- Attacking device needs to be able to force TMS and TCK above or below logic threshold voltage
- Can be done by combining lines to make a more powerful driver or using multiple attackers to overcome TMS and TCK signals
**Xbox 360 Exploit**

- Used to override Microsoft security features
- Allows homebrew code to be run, installation of HD, game modification, ripping of games
- JTAG is used to extract secret keys needed to perform exploits and to change programming

---

**Security Options**

- Buffers in the JTAG Chain
  - JTAG system connected in “Star” pattern instead of being chained (Separate TMS and TCK)
- Encryption/Authentication for JTAG use
  - Most of the research in JTAG security would be classified under this
  - Although it would provide much better protection, like all security hardware, increases cost and space.

---

**Challenge, Response**

- Requires PUF or randomly burned fuses
- Requires Set_Challenge and Get_Response instructions in JTAG implementation
- A Challenge input is given to the JTAG module, and the module will hash this with the value of it’s fuses to create the response
- Only a known, trusted module will give a correct response
- So, can be determined if modules are trusted or not

---

**Public/Private Key Authentication**

- Tester/Updater is required to have a certificate of authentication signed by a designated third party.
- Authenticators public key is known to JTAG system
- Using the known public key, the JTAG system can decrypt the certificate and determine whether the tester/updater is trusted
- Trusted testers/updaters are allowed access to JTAG system, un-trusted are blocked

---

**User Permissions**

- A user permission level, i, allows them access to instructions with a level less than i
- Requires extra hardware to authenticate user and set permission level, and to save settings for what each permission level can and cannot do
- Ex. In memory, a permission level is saved for each module in the JTAG system. When that module is trying to be accessed, the saved level is compared to the current permission level

---

**Removal/Destruction of JTAG**

- To completely defend against JTAG attacks, one thought is to remove the JTAG hardware all together
  - Does not leave a way to in-field test
  - Can use BIST for testing
- Similarly to removal of JTAG, some companies use security fuses to disable JTAG before the hardware leaves the factory
  - Can implement different levels of disabled JTAG use
Acknowledgments