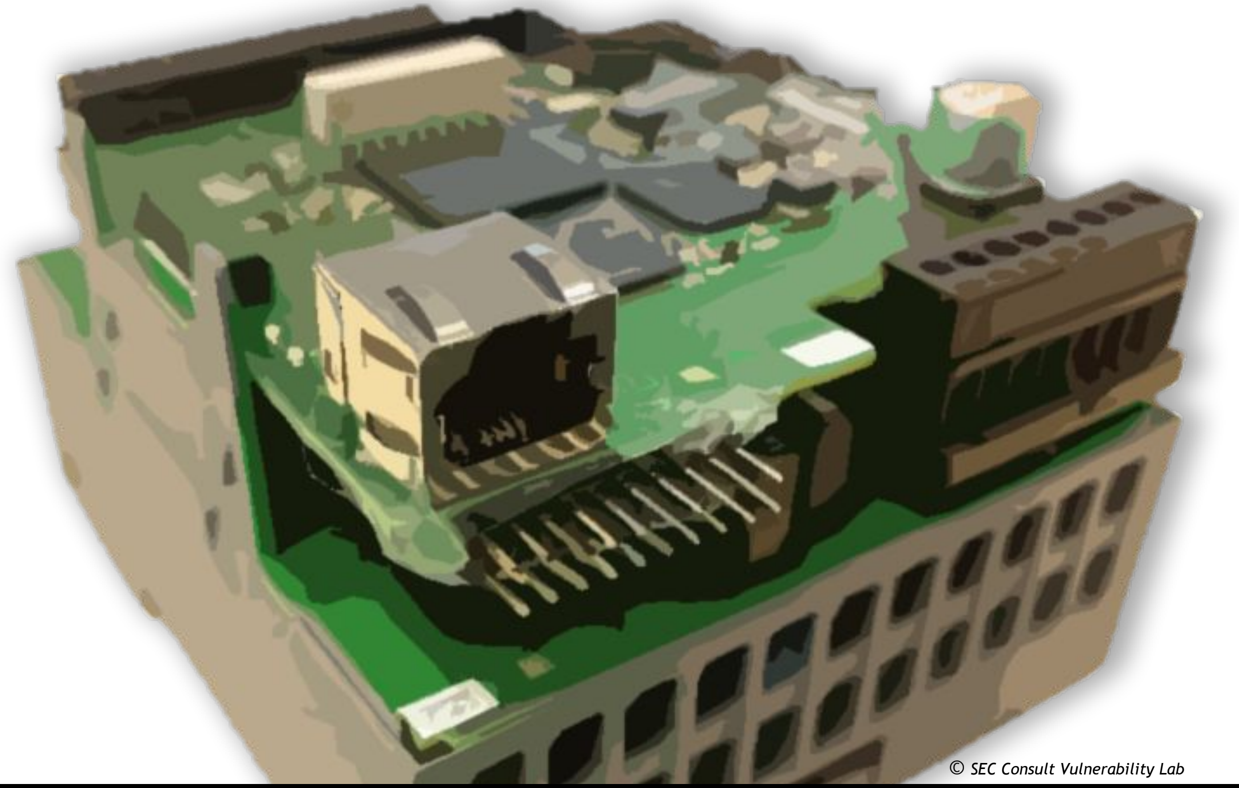
جامعـة نيويورك أبـوظـبي
NYU | ABU DHABI

MoMA LAB



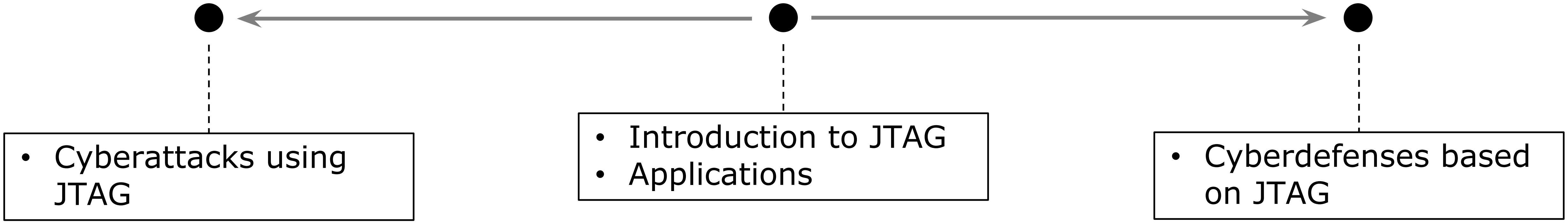
© SEC Consult Vulnerability Lab

JTAG: A Multifaceted Tool for Cyber Security

IOLTS 2019

Michail Maniatakos
New York University

Overview



- Cyberattacks using JTAG

- Introduction to JTAG
- Applications

- Cyberdefenses based on JTAG

JTAG - Joint Test Action Group [IEEE 1149.1 Standard]

- JTAG
  - Testing and debugging interconnects

- Modes
  - Extest Mode
  - Debug Mode
  - Normal Mode

- Extensions
  - IEEE 1149.6
  - IEEE 1149.7

ARM 10-Pin Interface

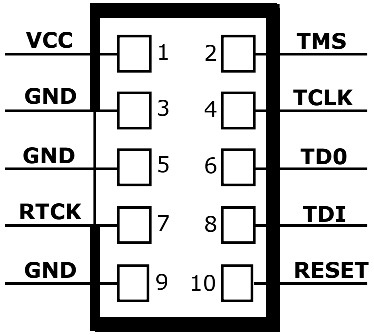


Figure 14: Typical ARM 20-Pin Interface

ARM 20-Pin Interface

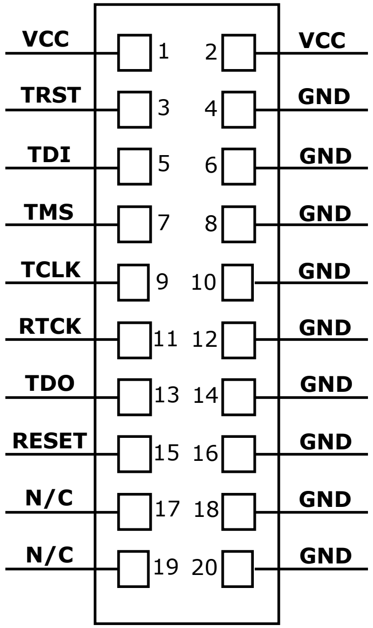


Figure 14: Typical ARM 20-Pin Interface

Applications of JTAG in Testing

- Debug
  - Embedded applications
  - Bootloader and Kernel
- Read / Write
  - Register
  - Memory
  - Programs
- Standardization

Cyberattacks using JTAG

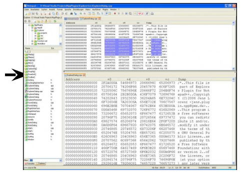
Attacks

[PLCs]

[Smartphones]

Modification Attacks on PLCs[2]

Extract Firmware image



Allen-Bradley's Controllogix 1756-L6

- Discover ARM function names
- Important locations in memory

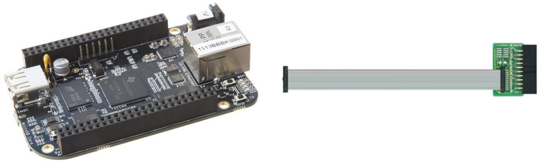
Static Analysis [IDA Pro]

- Identify execution paths
- Identify flags

Dynamic Analysis [JTAG]

- Identity object request sent can trigger a breakpoint
- DoS attack
- PLC goes into a faulty state

JTAG Combined Attack[1]



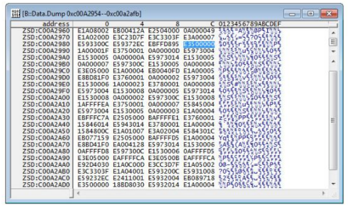
- 32 bit ARM Cortex A9
- Android v6.0.1

Modify format specifier (%pK)

[Hide kernel system call addresses]

[~$ cat /proc/kallsyms | grep setresuid]

Modify Opcode in RAM (cmp r0,#0x0)



[B::Data.Dump 0xc00A2954--0xc00a2afb]

```
     address         0        4        8        C 0123456789ABCDEF
ZSD:C00A2960  E1A08002 EB00412A E2504000 0A000049
ZSD:C00A2970  E1A0200D E3C23D7F E3C3303F E3A00007
ZSD:C00A2980  E593300C E59372EC EBFFD895 E3500000
ZSD:C00A2990  1A00001F E3750001 0A00000D E5973004
ZSD:C00A29A0  E1530005 0A00000A E5973014 E1530005
ZSD:C00A29B0  0A000007 E597300C E1530005 0A000004
ZSD:C00A29C0  E3E05000 E1A00004 EB0040FD E1A00005
ZSD:C00A29D0  E8BD81F0 E3760001 0A000002 E5973004
ZSD:C00A29E0  E1530006 1A000023 E3780001 0A000008
ZSD:C00A29F0  E5973004 E1530008 0A000005 E5973014
ZSD:C00A2A00  E1530008 0A000002 E597300C E1530008
ZSD:C00A2A10  1AFFFFEA E3750001 0A000007 E5845004
ZSD:C00A2A20  E5973004 E1530005 0A000003 E1A00004
ZSD:C00A2A30  EBFFFC7A E2505000 BAFFFFE1 E3760001
ZSD:C00A2A40  15846014 E5943014 E3780001 E1A00004
ZSD:C00A2A50  1584800C E1A01007 E3A02004 E584301C
ZSD:C00A2A60  EB077159 E2505000 BAFFFFD5 E1A00004
ZSD:C00A2A70  E8BD41F0 EA004128 E5973014 E1530006
ZSD:C00A2A80  0AFFFFD8 E597300C E1530006 0AFFFFD5
ZSD:C00A2A90  E3E05000 EAFFFFCA E3E0500B EAFFFFCA
ZSD:C00A2AA0  E92D4030 E1A0C00D E3CC3D7F E1A05002
ZSD:C00A2AB0  E3C3303F E1A04001 E593200C E5931008
ZSD:C00A2AC0  E59232EC E2411001 E5932004 EB089718
ZSD:C00A2AD0  E3500000 18BD8030 E5932014 E1A00004
```

[1]

[Inject Fault]

Request root privilege from sys_setresuid

[1] F. Majeric, B. Gonzalvo, and L. Bossuet, "JTAG Combined Attack - Another Approach for Fault Injection," in IFIP International Conference on New Technologies, Mobility and Security, Nov 2016, pp. 1–5.
[2] C. Schuett, J. Butts, and S. Dunlap, "An evaluation of modification attacks on programmable logic controllers," International Journal of Critical Infrastructure Protection, vol. 7, no. 1, pp. 61–68, 2014.

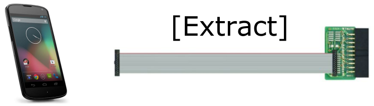
Cyberdefenses based on JTAG

Defenses

[Smartphones]

JoKER[3]

Detection mechanism for kernel rootkits



[Extract]

Raw Memory from selected regions

System call table

Exception vector table

Kernel process list

Golden Model

[Compare]

Rootkits

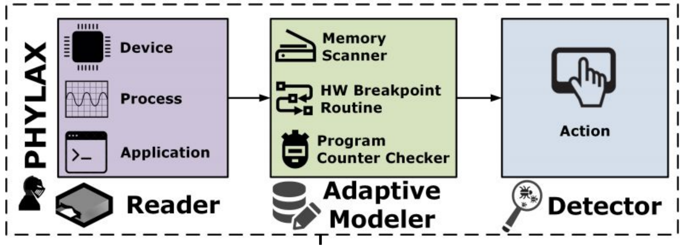
[Compare]

Protected Cache Pool (DKOM)

[PLCs]

PHYLAX[4]

Ready to deploy JTAG based online monitoring and detection mechanism



[4]

Fetch device, process and application specific parameters

Compares immutable section of the firmware to a golden model

Manages limitations on the availability of hardware breakpoints

Checks for any violations □ alert

[3] M. Guri, Y. Poliak, B. Shapira, and Y. Elovici, "JoKER: Trusted detection of kernel rootkits in android devices via JTAG interface," in IEEE Trustcom/BigDataSE/ISPA, vol. 1, 2015, pp. 65–73.

[4] C. Konstantinou, E. Chielle, and M. Maniatakos, "PHYLAX: Snapshot based profiling of real-time embedded devices via JTAG interface," in IEEE Design, Automation & Test in Europe Conference & Exhibition, 2018, pp. 869–872.

Securing JTAG[5]

- Level 0
  - Same as IEEE 1149.1

- Level 1
  - Additional authentication operation
  - Challenge-response pairs (Fuse bits)

- Level 2
  - Encrypted communication
  - Setup and communication phase
  - Chip response as key

- Level 3
  - Level 2 + MAC

[5] Rosenfeld, Kurt, and Ramesh Karri. "Attacks and Defenses for JTAG." IEEE Design & Test of Computers 27.1 (2010): 36-47.

Discussion

- Advantages
  - JTAG provides low-level access
  - It can read/write/dump data

- Challenges
  - CPU should be in test mode
  - Limited hardware breakpoints
  - Slow speed
  - Port not accessible

Thank You