GOTO
AARHUS 2023

#GOTOaar
Has my IoT device been hacked?

ESTABLISHING TRUST WITH REMOTE ATTESTATION

Edlira Dushku
Assistant Professor in Cyber Security
Aalborg University, Copenhagen, Denmark
edu@es.aau.dk
A bit about me

- (2016 - 2020): PhD at Sapienza University of Rome, Italy
- (2020 – 2022): Postdoc at Technical University of Denmark (DTU)
- November 2022: Assistant Professor at Aalborg University

Edlira Dushku
IoT Security, Trusted Computing
• Internet of Things Security

• Remote attestation protocols

• Open challenges
Internet of Things Security

Remote attestation protocols

Open challenges
Internet of Things (IoT) systems:

- Smart society
- Smart healthcare
- Smart transport
- Smart territory improvement
- Smart payments
- Smart buildings
- Smart energy
Internet of Things (IoT) systems

Industrial IoT

IoT for infrastructure

Consumer IoT
Internet of Things (IoT) systems
Cyberattacks on Iran — Stuxnet and Flame


About 90% of Smart TVs Vulnerable to Remote Hacking via Rogue TV Signals

Oct. 10, 2017

How Israel Caught Russian Hackers Scouring the World for U.S. Secrets

Exploiting the popular Kaspersky antivirus software, Russian hackers searched millions of computers for American intelligence keywords. Israeli intelligence tipped off American officials.

Over 8,600 vulnerabilities found…

FDA recalled half a million pacemakers…

“If you want to keep living, pay a ransom, or die…”

Casino Gets Hacked Through Its Internet-Connected Fish Tank Thermometer

Sunday, April 15, 2018  Wang Wei
IoT devices prone to cyberattacks

EASY TO EXPLOIT
- Resource-constrained devices with low-cost design
- Do not support complex security techniques

ATTRACTIVE TARGET
- Deployed in safe-critical domains
- Contain sensitive data & control physical environment

AMPLIFY THE ATTACK IMPACT
- Many interconnected devices
- Spread quickly the malware
How to improve the situation?
Option 1: Security-by-design
Security-by-design
Security-by-design

• No cybersecurity expert
• No additional time/money
• Rush to market
Option 1: Security-by-design

Difficult: Cannot guarantee that devices do not get compromised
Option 2: Malware detection

Detect compromised device (to isolate from the network)
How to detect malware presence?

Guarantee that the device is “telling the truth” even when it is infected by malware.
Remote attestation (RA)

- Two-party Security Protocol
  - **Verifier**: an external trusted entity, not always present, not possible to physically reach a device
  - **Prover**: a (potentially) compromised device

- RA allows the **Verifier** to **guarantee** the **authentication and integrity** of the software running on **Prover**

- Verify that Prover is **NOW** running the initial application
RA in Traditional systems: TPM

- **Hardware-based attestation** using a Trusted Platform Module (TPM)
- Secure crypto processor creates, stores, uses cryptographic keys
- Direct Anonymous Attestation (DAA): Makes anonymous remote attestations of host status
RA in Traditional systems: SGX

- Hardware-based memory encryption that isolates specific application code and data in memory.

- Allows user-level code to allocate private regions of memory, called enclaves, which are designed to be protected from processes running at higher privilege levels.

Intel Software Guard Extensions.
https://software.intel.com/en-us/sgx
Securing data in use and accelerating the adoption of confidential computing through open collaboration.
- Internet of Things Security
- Remote attestation protocols
- Open challenges
Overview of Remote attestation in IoT

1. Challenge (Executed by Verifier)
   Outputs a random Challenge (nonce, timestamp, memory addresses, attestation routine)

2. Attest (Executed by Prover)
   Computes a small attestation response based on internal state $S$ (e.g., checksum over memory contents) and challenge $c$

3. Verify (Executed by Verifier)
   Compares with the response received from Prover with the expected state
Typical adversary models

1. Software Adversary
   - **Remote:** Infect device(s) with malware
   - **Local:** Learn device secret, impersonate or clone, can launch side channel attack
   - **Mobile adversary:** Relocates or deletes itself

2. Hardware Adversary
   - **Stealthy Physical Intrusive:** Capture device and physically extract secrets, clone device(s)
   - **Physical Intrusive:** Capture device and modify contents/components
1. **Challenge** (Executed by Verifier)
   - Authentic, Fresh, Unpredictable

2. **Attest** (Executed by Prover)
   - Authentic, Unforgeable, Dynamic, Deterministic

3. **Verify** (Executed by Verifier)
   - Deterministic
Approaches of Remote attestation

- **Hardware design**
  Hardware-based, Software-based, or Hybrid

- **Memory**
  Static vs Control-flow attestation

- **Number of Device**
  Single Device vs Swarms (Collective)

- **Network Topology**
  Static vs Dynamic Swarms

- **Communication data**
  Swarms vs Distributed services
**History of Remote attestation**

**SOFTWARE-BASED**
- SWATT (S&P 2004)
- Pioneer (SOSP 2005)

**HYBRID-BASED**
- POSE (ESORICS 2010)
- SMART (NDSS 2012)
- TrustLite (Eurosys 2014)

**SWARMS (COLLECTIVE)**
- SEDA (CCS 2015)
- SANA (CCS 2016)
- DARPA (WISEC 2016)

**CONTROL-FLOW (RUNTIME)**
- C-FLAT (CCS 2016)
- ATRIUM (ICCAD 2017)

**DISTRIBUTED**
- RADIS (SDS 2019)
- SARA (TIFS 2020)
- ESDRA (IOT-J 2019)
- DIAT (NDSS 2019)
- PASTA (Euro S&P 2019)

**DISTRIBUTED services**

**DISTRIBUTED verifiers**

**2010**

**2015**

**2016**

**2019**

**2023**

**PRIVACY**
- ZEKRA (ASIA CCS 2023)
Hybrid attestation (Typical RA paradigm)

- Prover and Verifier share a key $k$
- Verifier expects configuration $h'$

1. Challenge $N$
2. Measure software state
   $h = \text{HASH(Software)}$
3. Authenticated Response
   $\delta = \text{MAC}_k(N \ || \ h)$
4. $\delta' = \text{MAC}_k(N \ || \ h')$

If there is a match, confirm the trustworthy state
Swarm attestation (Collective)

- Verify the internal state of a large group of devices
- Should be more efficient than attesting each node individually

Algorithm logic:
1. Verifier selects random Prover ($P_0$) initializes attestation
2. Spanning tree is created rooted at $P_0$
3. Each Prover (device) gets attested by its parent (leaves first)
4. Sub-tree roots accumulate results and reports to their parent
5. $P_0$ reports overall result to Verifier
Limitations

○ Lack of flexibility (ALL devices must participate to attestation), final result is boolean

○ Aggregators should be trusted, single point of failure

○ Network topology and attestation are static
Dynamic attestation

Program Memory Attestation schemes do not address runtime attacks
Code injection attacks

A: \( \text{if}(\text{auth}==\text{true}) \)
B: \( \text{then} \): call privileged()
C: \( \text{else} \): call unprivileged()
D: \text{terminate}
...
E: privileged{...instructions...}
F: unprivileged{...instructions...}

X: injected_code

1. Exploit Buffer Overflow
2. Inject Code
3. Alter Control Flow
**Code reuse attack**

1. **Exploit Buffer Overflow**

Adversary

Pseudo-code

- A: if(auth=true)
- B: then: call privileged()
- C: else: call unprivileged()
- D: terminate
- ...
- E: privileged{...instructions...}
- F: unprivileged{...instructions...}

Control-flow Graph (CFG)

2. **Control Flow deviation**
C-FLAT: Control-flow attestation

- Proposes a complete attestation of the run-time state of the Prover
- A single hash value that represents the entire control flow of the Prover’s state

Cumulative Hash Value: $H_i = H \left( H_{i-1}, N \right)$

- $H_{i-1}$ -- previous hash result
- $N$ -- instruction block (node) just executed

Diagram:
- $H_1 = H(0, A)$
- $H_2 = (H_1, B)$
- $H_3 = H(H_1, C)$
- $H_5 = (H_2, E)$
- $H_6 = (H_3, F)$

Authentication:
$\text{Auth} = H_4 = H(H_5, D) \text{ OR } H(H_6, D)$
Loops are a challenge!

Different loop paths
and loop iterations lead to many valid
hash values
C-FLAT Approach:

Treat loops as sub-graphs and report their hash values and # of iterations separately
C-FLAT approach

Auth = H7, <H1, {<H6a, #H6a>, <H6b, #H6b}> >
Advantages
• Better detection level: Detects runtime attacks

Disadvantages
• The protocols rely on customized hardware support
• The computations are not efficient
Internet of Things Security

Remote attestation protocols

Open challenges
• Privacy-preserving remote attestation for IoT systems

Heini Bergsson Debes, Edlira Dushku, Thanass Giannetsos, Ali Marandi,
To appear: the 18th ACM ASIA Conference on Computer and Communications Security (AsiaCCS 2023)
Energy-harvesting IoT security

• Lightweight RA operation designed specifically for Intermittent IoT system

RESERVE: Remote Attestation of Intermittent IoT devices
MD M. Rabbani, E. Dushku, J. Vliegen, A. Braeken, N. Dragoni, N. Mentens
Asynchronous Swarm attestation

Conclusions

- Introduced RA of IoT devices: Security protocol that guarantees trustworthiness

- Highlighted the need for the attestation of IoT devices. RA can serve as a fundamental building block for other security protocols.

- Presented an overview of the main RA protocols proposed in the literature (hybrid, swarm, control-flow)
Would you like to know more?

Grab a course at Master of IT

• Software construction
• Cyber Security
• IT-architecture
• Digital transformation
• Data Science
• IT-management

www.master-it-vest.dk
Thank you!

Edlira Dushku
edu@es.aau.dk