Cybersecurity challenges in the IoT era

Webinar | December 11, 2019
Using IoT platform security with mF2C to develop scalable secure edge-to-cloud applications
IoT Challenges

- Trust/Reputation of IoT
  - "Why should I trust this?"

- Legacy of IoT
  - Loads of insecure stuff already out there

- Barriers to building new applications
  - Have you ever programmed a microcontroller?
  - Limited devices (comp/mem/storage.)

- Keeping it secure
  - Potentially huge "attack surface"
  - Human factors, human bias
Platform Premises

- mF2C focuses on edge->fog->cloud applications
  - Tasks/data pushed to higher level if needed
- Build a *platform* for building applications
- Three use case applications:
  - Building sensors for emergencies (e.g. earthquakes)
  - Smart boats for boat sensor/location/harbour
  - Airport hub for traveller assistance
Generic Application
OSI (near enough) stack view

EDGE (mobile)
- IoT app
- WiFi (5G)
- REST / gRPC clients

FOG
- Implemented features
- task execution
- client/server
- LAN / WiFi

CLOUD
- Customer database
- service matching
- task execution
- server
- WAN
Platform Architecture

- WiFi (5G)
- LAN / WiFi
- REST / gRPC clients
- phone app
- recommender system
- task execution
- client/server
- database
- service matching
- task execution
- server
- WAN

Layer 0:
- Cloud agent
- Layer 0

Layer 1:
- Leader
- Backup
- mF2C_agent
- Layer 1

Layer 2:
- Leader
- Backup
- mF2C_agent
- Layer 2

Layer N:
- Leader
- Backup
- mF2C_agent
- Layer N

Layer N + 1:
- Leader
- Backup
- mF2C_agent
- Layer N + 1
Security Features

• Usually, device == 1 agent
  – Or microagent for smaller devices
• Device id tracks device lifetime
• Certification Authority in the cloud
  – Certificates to capable (fog) devices
  – Private key generated by (capable) device
  – Gateway gives access to cloud service
  – No Internet access for unauthenticated devices
• Edgier devices have private (typ. a serial bus) link to foggier devices
Zooming in further

mF2C Agent

Platform Manager

- **Service Orchestration**
  - Lifecycle management
  - SLA management

- **Distributed Exec. Runtime**
  - Task Management
  - Task Scheduling
  - COMPSs

- **Telemetry Monitoring**
  - Landscaper
  - Recommender
  - Analytics Engine

Agent Controller

- **Resource Management**
  - Discovery
  - Policies
  - Identification

- **User Management**
  - Profiling
  - Sharing model
  - Assessment

- **Event Manager**
  - GUI

- **Data Management**
  - Dataclay

- **Security**
  - Reverse Proxy
  - CAU Client
  - AC library

- **APIs**
  - Docker
  - Kubernetes
  - Swarm
  - CIMI
Addressing the challenges - Trust

• PKI for all participants
  – Distinct PKI roots for infrastructure and agents
  – Optionally distinct PKI for application
• CI/CD through Docker containers
• Trust model for security
• Application data
  – PUBLIC for unprotected
  – PROTECTED for integrity protected
  – PRIVATE for integrity and confidentiality
Addressing Challenges – Legacy

- mF2C builds entirely new applications, so no legacy?
- Some users bring own devices
- => botnet detection

- Early work on botnet detection
  – Distinguish attack from (say) emergency
  – Remote control of router/firewalls
Addressing Challenges - Barriers

• Build application on platform
  — ... however, mF2C is a research project

:-)
• Open source
• High TRL on some components
• Lots of clever people adding lots of clever features
• Some code written by professional programmers and RSEs

:(
• Platform has more features than a given app might need?
• Low TRL on some components
• Some code written by students rather than RSEs?
Addressing Challenges – Future

- mF2C updates through its CD framework
- Phone app (airport use case) through app store
- Edge hardware/firmware not addressed in project
  - (e.g. Azure Sphere..)
- Those pesky humans...
  - Make it easier to do the Right Thing
  - Need transparency for GDPR, too
Thanks!

• https://mf2c-project.eu/
• https://github.com/mF2C/
=> jens.jensen@stfc.ac.uk