mF2C and
OpenFog Consortium Security

The following document compares the security implementation of the mF2C project with the security framework proposed by the Open Fog Consortium (OFC), focusing on the security developed or foreseen for the forthcoming release, mF2C's IT-1 (July 2018).

This document primarily targets people who wish to know how mF2C security aligns with OFC security. It should also serve as a feedback/discussion document for OFC participants.

About mF2C:
The EU Horizon2020-funded[1] mF2C project [2] is investigating the IoT edge-fog-cloud computing paradigm. The mF2C project aims to design an open, secure, decentralised, multi-stakeholder management framework, including novel programming models, data storage techniques, service creation, brokerage solutions, SLA policies implementation, and resource orchestration methods. The project includes support for data privacy and security. The project includes three industry-driven use-cases: UC1 - Emergency Management in Smart Cities, UC2 – "Enriched navigation" for marine vessels, UC3 - Smart Fog-Hub Services for Airports. The formal outputs of the project are published online as deliverables, and source code is available on GitHub.

OFC Security Principles and Goals

We start the discussion with a quick overview of the OFC security principles and goals and compare them to those of mF2C. As we shall see, mF2C closely follows the same principles.

OFC Principle: security in the edge and fogs must be taken seriously - too many devices are developed and deployed as insecure, with potentially devastating effects.

mF2C takes security seriously: from the initial surveying of the landscape in D2.1 to the main (as of this writing) security deliverable D2.4 (delivered in March and April 2017, respectively), as well as numerous more specialised documents ranging from data policies and privacy, security workflows, threat analyses, etc., there is a wide range of information. Moreover, the development of a sandbox environment enables testing for security holes alongside the development, and ensures that the project can collaboratively develop services that are secure. Testing in the sandbox means that

[1] mF2C is project number 730929, under the topic *ICT-06-2016 Cloud Computing*, started Jan '17 and runs till Dec. '19 (three years.)

intrusive tests can be made, which would otherwise not be permitted on shared (office, data centre, or cloud) networks.

OFC Principle: security-capable devices can be introduced into the fog in order to mitigate risks.

mF2C is developing several components to implement secure services.

- An agent for secure communications through an API
- A CA which is deployed in the cloud, in order to provide security across fogs.
- A fog-to-cloud smart gateway specific to establishing communication between the agent in the fog and the CA: we can't let any as-yet unauthenticated agent access the wider Internet, so this particular gateway lets them obtain credentials securely from the cloud CA.

In addition, several other services are foreseen (e.g. in D2.4), mainly for our second release (scheduled for the end of 2019):

- Active monitoring for intrusion in the fog. While we believe a full Security Information and Event Management (SIEM) system would be too expensive and would require too much effort, we are currently investigating alternative unsupervised solutions including one based on machine learning.
- Active protection against botnets.
- Software-defined perimeter - an early prototype was demonstrated at an internal workshop in November 2017.

OFC Principle: Security is based on the capabilities of the device.

mF2C has designed a security policy (D3.1, June 2017) which takes into account the capabilities of the device. Certificates, following experimental evidence from devices with lower computational capability, can use elliptic curve cryptography. Devices that cannot handle certificates can still communicate over private links to (preferably) dedicated devices that handle the onward security for them. For example, in Use Case 2 (UC2, smart boats), the boat sensors are not very computationally capable, but communicate over CANbus to a device like a Raspberry Pi which can aggregate data and handle the secure onward connection to the fog.

OFC Principle: Support for legacy devices.

mF2C is based on the distinction between communication within mF2C-controlled infrastructure (i.e. devices with mF2C software installed on them), and to/from the outside. The data security policy applies to both; in other words, while mF2C may need to send data to outside the mF2C fog/cloud domain, the mF2C infrastructure retains the *responsibility* for the processing of the data. The threat model takes the boundary into account, where external devices will interface to mF2C. An example is again the smart boat use case, which needs to interface to payment systems.

OFC Specific Risks of Open Environment Operation

1. Physical exposure: mF2C recognises that physical exposure is in general a serious risk, but we do not foresee physical protection for IT-1 that would protect against a resourceful adversary, for the following reasons:

   a. The focus of the mF2C project is on software: we are not building our own hardware (this is a key concept in mF2C);
   b. We wish to be inclusive, to be able to run our software on as many devices as possible.
   c. Implementing tamper-proof and tamper-evident hardware, à la Trusted Computing Platform or FIPS 140-2, would be too costly.

   mF2C has taken three approaches to mitigate the risk of physical compromise:

   a. Accept that the device may be compromised: detect it and react (D2.4).
   b. The environment in which the device is deployed can often offer physical security. UC1 and UC3 take place in a smart city (= building) and airport, respectively, both of which can offer physical security to the devices deployed within them. In the airport, the wireless access points are obviously protected against tampering, although end user's mobile phones or other wireless-capable devices would of course not be under mF2C's control; only the app deployed on the phone would be.
      Only our UC2 (smart boats) has necessarily physically exposed edge/fog devices.
   c. In the actual – as opposed to speculative – production deployment, mF2C will include the risk of physical compromise in its full risk assessment.

2. Open security boundaries. mF2C has defined a single underlying security architecture. Data must be protected according to a security policy (D3.1) which defines the data exchange between mF2C devices as well as with devices outside of mF2C. The converse is true as well: sometimes mF2C may use information or features from other security domains (e.g. smart phone app store or authenticators, or payment systems in UC2), with the user's knowledge and consent.
3. Remote management. mF2C will be able to update apps in smartphones through the conventional app store update. We recognise that updating physically remote devices in general is difficult, but one particular aspect we have explored already is remote management of firewalls. The purpose of this is to dynamically modify the network permissions of infrastructure and boundary devices - for example, a device may be compromised, but it can be isolated using firewalls. This is particularly true for "vertical" communication which predominates mF2C's first release ("IT-1"), but can also apply to "horizontal" communication.
4. Legacy brown-field devices. As in point 6 below, legacy devices are usually managed through an intermediary. We acknowledge that legacy devices are not likely to be kept up to date with security patches. We mitigate this risk through the intermediary; our proposal is to eventually have intelligent filtering (aware of content type, etc.)
5. Heterogeneous Protocols and Operation Procedures. The mF2C developments establish a framework for multi-disciplinary edge-to-cloud computing and data management. Through the use of abstractions, the shared framework implements a common data policy on multiple communications protocols and networks, enables resource optimisation and sharing. Thus, while the use cases are necessarily different, they share features and capabilities through the framework, yet can remain isolated from each other in operations through the service allocation and through different trust domains.
6. Resource Constraints among Devices. mF2C's data policy also allows for constrained devices to connect and participate: typically, these communicate over trusted (local) networks or buses (such as I^{2}C, CANbus, etc.), passing information through to a more capable device which runs mF2C software. The more capable device is responsible for

enforcing the security policy for the information it handles. For example, private data must be encrypted before being passed across untrusted networks.

7. Multi-tenancy. In mF2C we have decided not to rely on the use of trusted computing hardware (cf. item 1, but see also the separate section below) as our use-cases need to support devices without specialised trusted-compute capabilities. As we move “up” towards the cloud, resources are increasingly shared, but isolation and multi-tenancy are well established in cloud services provisioning. In future evolutions of mF2C we will increasingly share resources at the fog/edge levels (see also next item, item 8), so the risks of multi-tenancy will need to be revisited.
8. Multi-Tier IoT-Fog-Cloud Mash Up. mF2C distinguishes between “vertical” (up and down the hierarchy, e.g. fog-to-cloud) and “horizontal” communications (peer to peer). By introducing a “leader” into each fog, we enable devices to register themselves once they join a fog and start discovering other services or peers. If needed, extra security can be added by requiring communications to go through the leader (“vertically”), at the cost, obviously, of placing a higher load on the leader.

   Any communications with the cloud must go through one (or more) similar gateways which can be implemented as smart gateways. We already have one enabling fog-to-cloud communication with the CA; the leader can act as a within-the-fog gateway itself if agents discover other agents through the leader, or even communicate through the leader - these restrictions can (optionally) be implemented in the secure communications code. Future work includes further gateways that connect an agent in a fog to another agent in another fog; these are similar to the CA-gateway, except that they are established on demand.

Requirements

1. Extrinsic vs intrinsic security: standard security protocols are used (e.g. TLS, RFC 5246, and JWE, RFC 7516). The root of trust (in this case, the CA service in the cloud) is designed to enable devices to bridge from one fog instance to another, but can also easily be set up to fully separate the use cases from each other (i.e. each use case has its own CA.) Assurance levels, in the current state of the project, focus on data security requirements, but other assessments can be introduced later.
2. Protection scope: the protection scope is designed around the leader (see item 8 above), which facilitates communication within the fog (e.g. discovery of peers and lookup of their public keys), and can act as a gateway (or enable discovery of gateways) to communicate with entities outside the fog.

   We take on board the suggestion to document our CIDs[2] and SADs[3].
3. Threat models have already been developed in a generic sense; once we gain practical experiences with our use cases, we shall refine them to use case specific threats.
4. Goals and Deliverables were defined in the mF2C proposal but are regularly reviewed for relevance; in particular, mF2C has a scientific advisory board which oversees the scientific goals as well as oversight from the European Commission who will have appointed expert reviewers for the project. It should be noted that collaboration with external projects or parties, such as OFC, is an important objective of mF2C.

[2] Connectivity/Interoperability Domains, [1] section IV.B.

[3] Service/Application Domains, [1] section IV.B.

Approaches

1. Physical security: While devices lower in the hierarchy (edge, fog clients) may be physically insecure, all mF2C use cases have the ability to physically secure fog-to-cloud components[4]. This security, however, is the responsibility of the deployer, not for mF2C as we primarily focus on software. On a related note, we are not currently using Common Criteria for our evaluations; this, again, is something we may pick up later in the project.
2. End-to-end security is enabled through X.509 certificates, enabling secure communications both over secure sockets (TLS) as well as message level security (cf. JWS, JWE). mF2C software is distributed/installed with the relevant trust anchors. We support confidentiality, integrity, message origin authentication. Initially we do not support non-repudiation, as it will need a “signature” from the end user, but we recognise that it will be needed later, e.g. for financial services or some types of resource sharing.
3. Security Monitoring and Management will be implemented in a more lightweight fashion than a full SIEM system. Initially we have chosen to focus on vertical communications, thus enabling closer monitoring by higher level services. We are, however, not actively blocking peer-to-peer communications for clients in the fog.
4. Domain and Policy Management. Interoperation is enabled through the shared mF2C framework; our operational infrastructure manages the execution of tasks appropriately through resource selection and execution optimisation. As regards the application, we require that applications comply with the mF2C data security policy; as application developers for the use cases are part of the same project, it makes sense to implement application data security through them.

How mF2C can contribute to OFC security goals

- Interfaces to shared services, frameworks for edge-to-cloud applications. While we do not claim that the ones used in the initial release are perfect, we shall at least have enough to establish experiences and learn how we can make them better.
- Implementation of a trusted computing base.
- Fog node specific protection profile (or experience/feedback on the same, based on experiences gained from mF2C use cases.)
- Unless mF2C security policies/domains are too far afield, mF2C may be able to contribute to the development of protection profiles for fog nodes, or, subsequently, real-life experiences with these profiles.
- Resource sharing at the edge (based on mF2C planned work).
- Experiences with data privacy implementation in fog-to-cloud infrastructure.
- Experiences with remote management for security (e.g. firewalls, botnet controls) and automation of these tasks.

References:

[1] http://arsalanmosenia.com/papers/Openfog_preprint.pdf
[2] http:/www./mf2c-project.eu/

[4] UC1 (“smart cities”) can protect devices in buildings; UC3 (“smart hub”) can protect their wireless access points, e.g. in an airport. UC2 (“smart boats”) can physically protect only the resources in the harbour, but devices on boats may bypass the harbour and communicate directly with the cloud.