Cross-Layer Security Demonstrator for Future Internet

Julius Mueller[1], Abbas Siddiqui[2], and Dirk Hoffstadt[3]

[1] Technical University Berlin
Straße des 17. Juni 135, 10623 Berlin, Germany
julius.mueller@tu-berlin.de
[2] Technical University Kaiserslautern
Fachbereich Informatik, Postfach 3049, 67653 Kaiserslautern, Germany
siddiqui@informatik.uni-kl.de
[3] University Duisburg - Essen
Ellernstr. 29, D-45326 Essen, Germany
dirk.hoffstadt@iem.uni-due.de

Abstract. Cross-Layer Functional Composition represents an architectural approach of the Future Internet, in which network and service layer communicate directly, in order to optimize parameters of a particular connection. We propose a Cross-Layer architecture with a special focus on security issues. An outline of our prototypical implementation finalizes this paper.

Keywords: Cross-Layer Security, Future Internet, Functional Composition

1 Introduction

Todays Internet runs on the best-effort packet transport paradigm for several applications in many but not all domains. Several other domains like emergency, finance, government and private domains require scalability, a higher level of security, mobility, trustability, flexibility and Quality of Service (QoS).

One idea of interpreting Future Internet is a Cross-Layer architecture design with Functional Composition. Future Internet applications will signal their requirements for a particular connection out of the service layer down to the network layer, in which a functional composition framework creates and establishes a data path, suitable and highly optimized for the demanded network connection.

This paper presents a novel Cross-Layer security prototype for Future Internet using Functional Composition. The proposed architecture is implemented prototypically as a demonstrator, which is described in the following section(s).

2 Architectural Design

This section covers the G-Lab DEEP demonstrator, highlight its main components and points out security related functionality. The components are described in the order of their execution in the demonstration workflow.

The IMS client framework MyMONSTER (*User Equipment*) is equipped with an add-on providing functionalities to state an intent, which is either a standard or emergency voice call. This intent is signaled on to the Service Broker.

A *Service Broker* translates the intent and determines the information of the demanded action and formulates a request. A connection to an IP Multimedia Subsystem enforces security related functionalities like Authentication, Authorization and Accounting (AAA) of the UE and its intent. The Service Broker identifies the required services, which are necessary to satisfy the users request. Such a service may consist of a single service or several services, which might be combined as a complex service. After selecting the services, the application level requirements are derived. These requirements are signaled on to the Mediator.

The *Cross-Layer Mediator* exchanges information (e.g. applications requirements, offered functionalities at network layer) between service and network layer. A mapping of requirements is done before a suitable solution is evaluated e.g. through a cost function. [1]

Network Composition decomposes functionalities of the network stack in different functional blocks. These functional blocks are loosely coupled and provide means to exchange information between functionalities of different levels.

In order to manipulate, manage, and deliver the services, a *Functional Composition Framework (SONATE)* [2] has been developed which consists of different components. The components such as building blocks, a building block repository, and a workflow engine. In our approach building blocks are the implementation of the services which are composable. The repository holds available building blocks. Building blocks are independent but they can communicate with each other by exchanging information. The work flow engine is responsible to execute building blocks according to a given work flow. Building blocks describe themselves by holding information such as covered services, efficiency, QoS, and requirements plus constraints for the execution of a building block.

The *Attacker* module simulates and emulates Spam over Internet Telephony (SPIT), Private Branch Exchange Hijacking and Denial of Service attacks. The module acts as a potential attacker in the test environment and is configurable and controllable by a management interface.

The *Mitigation* component monitors, analyzes and blocks harmful traffic on service and network layer information. Misbehavior is derived out of collected and aggregated traffic. Furthermore, the component has to distinguish between normal use of a service and service misuse.The distributed firewall (D-CAF) is able to block attacks close to the source to avoid attacks in the network.

3 Implementation of a Prototypical Demonstrator

Our presented demonstrator is a first prototype in the scope of the BMBF G-Lab DEEP project. The demonstrator addresses security in the Cross-Layer architecture using Functional Composition to enable traffic differentiation in a voice call scenario. A voice call is established under different circumstances within this scenario. Demonstrated circumstances for a call are:

1. Successful normal voice call under normal network conditions
2. Attacker overloads the network (i.e. Denial of service)
3. Normal call fails due to the utilized network
4. Successful emergency call in an overloaded network, which uses prioritization to ensure QoS

Security related processes are performed on different steps of this demonstration and are summarized in the following text.

Each call setup is established initially via the broker, which proves the incoming UE intent, enforces AAA and signals service requirements to the mediator. The mediator is a trustful component, too and resides in the operators domain. The building blocks are selected and executed at the network level. Prioritization, error correction and encryption might secure a connection.

To establish a normal call(1), a caller and a callee will be registered to the IMS core as a normal user. While establishing a call, requirements will be dispatched to the network layer to enable building blocks required for the connection. This requirements will be formed by service broker and send it to the mediator which will further transfer it to a network composition process. Where requested services will be appended for the flow; requirements consist of required service and flow specification (i.e. to identify a flow).

A normal call doesn't have any special requirements, in case of attack which is being demonstrated by making number of unwanted calls (2) to the callee. These unwanted calls will overload the network and the quality of valid normal call will be deteriorated or entirely failed (3) to have further communication.

In case of emergency call (4), a caller and emergency unit will be registered to the IMS core. As emergency call should have a priority over other calls, special requirements will be sent to network layer to enable prioritization for the particular flow id. It is required to have user authentication and location to make an emergency call so the fabricated emergency calls can be avoided. As in case of an emergency call, a priority will be given to the emergency call flow and the attack will be mitigated by dropping other packets. Thus the quality of an emergency call will merely deteriorate.

References

1. A.Siddiqui, D.Guenther, P.Muller, C.Henke, T.Magedanz, Mediation between Service and Network Composition, EuroView 2010, Wuerzburg, Germany
2. Reuther, Bernd and Mueller, Paul, Future Internet Architecture - A Service Oriented Approach, Oldenbourg Verlag, Muenchen, 2008