Abstract. Active Networks (ANs) aims at incorporating programmability into the network to achieve flexibility. However, increasing flexibility results in new security risks, which cannot be handled by existing ANs systems. In this paper, we aim at analyzing the security of ANs in language level based on the active code. First, we present the notion of active packet hierarchy. Next, we abstract the AN with Seal-calculus, and security protection is represented formally in four propositions. Finally, an example is used to address security protection. Our security protection is symmetrical, which protects a host from the untrustworthy active codes that migrate to it as well as active codes from the untrustworthy host where it migrates.

Keywords: ANs, active codes, security, Seal-calculus, formalization

1 Introduction

ANs provide a general framework within which users inject programs contained in messages into a network capable of performing computations and manipulations on behalf of users. We call these programs active codes [5].

By introducing mobile active codes into the network, the network is ensured not being destroyed and influenced, which is very important for the sharing network resource and structure. Therefore, it must, at least, be as safe, secret and sound as the present systems when the active codes migrate and execute in quite a large distributed system including many management fields [6]. It is vital to tackle security problems as our key to ANs research.

In the last couple of years, a number of process calculi have been designed to model aspects of distributed programming. Among these calculi are the $\pi$-calculus [1], the Join-calculus [2], Ambient calculus [3], and Seal-calculus [4]. The goal of seal-calculus is to explore the design space of security and mobile features. It provides strong security protection mechanisms.

Despite significant efforts devoted to security research in ANs [5,6,7], the issues of security still remain unsolved. Especially, formal description of ANs security has not yet been touched at present. This paper attempts to formally describe the security requirements in ANs with Seal-calculus. We believe that it is greatly significant to describe the security of active codes formally so as to handle the ANs security more perfectly.

In this paper, we discuss security properties of ANs in language level based on the active code. We present the notion of active packets hierarchy, in which each active
packet can be a container of other active packets. We formalize the AN in Seal-calculus. The security protections basing on the formalization above are represented formally with propositions. Finally, we address security protections with an example, which demonstrate our result. And the protection provided is symmetrical, and it protects a host from active codes that migrate to it as well as active codes from the host where it migrates.

This paper is organized as follows. In Section 2, the preliminary work is introduced. The hierarchy of active packet is presented in Section 3. Next in Section 4 we give formalization of active network with Seal-calculus and its security protections. An example is presented in Section 5, and the related work is discussed in Section 6. Finally, we conclude the paper and outline future work in Section 7.

2 Preliminaries

2.1 The Security of AN

The main problem in ANs security is the security of active code. Protection mechanisms of active code security may be placed at all component boundaries with the goal to control and regulate interaction among components. Figure 1 shows the four boundaries that require security [6,7,8]:

- a. Active packets, active extensions must be protected from being attacked by each other, which prevent a computation from disrupting another information or information gained privilege.
- b. The computational environment (CE) must be protected from the potentially malicious active packets and active extensions.
- c. Active packets, active extensions must be protected from the potentially malicious CE.
- d. Protect the security from the bottom transfer network.

The remarkable thing in the case of active code security issues is the symmetry of the security concerns: both the active codes and their execution environment must be protected. The goal of this paper is to study formally how active codes can protect from each other, and how hosts can protect themselves from being attacked by active codes, and vice versa.

![Fig. 1. Active Code Security Issues](image-url)