Security Clustering: A Network-wide Secure Computing Mechanism in Pervasive Computing*

Jabeom Gu[1], Sehyun Park[1]**, Jaehoon Nah[2], Sungwon Sohn[2], and Ohyoung Song[1]

[1] School of Electrical and Electronics Engineering,
Chung-Ang University, Seoul 156-756, Korea
jabeom@ms.cau.ac.kr, {shpark, song}@cau.ac.kr
[2] Electronics and Telecommunications Research Institute (ETRI)
{jhnah, swsohn}@etri.re.kr

Abstract. In this paper, we introduce a new security paradigm, called security clustering, for pervasive computing environment that enables network-wide defend against increasing evolutionary attacks on the heterogeneous network and hosts. Security clustering make use of dynamic security context exchange between cluster members and distributed information sharing to achieve scalable and efficient cooperation.

1 Introduction

The pervasive computing can be envisioned as an open network with a high degree of heterogeneity, providing advanced Internet services to mobile users [1-3]. The openness of the network may well be the most important feature that the success of many future mobile applications rests on. However, in dealing with interwork of huge set of heterogeneous components, the absence of appropriate mechanism to actively detect and put down various attacks will result in a liability to the open environment. Mobile users will face increased possibility of unwilling exposure to the significant security hazards caused by various types of attacks conducted on the network or on the device. In this paper, we propose *security clustering*, a network-wide defending mechanism against attacks. Security clustering make use of dynamic security context exchange between cluster members and distributed information sharing to achieve scalable and efficient cooperation.

The rest of this paper is organized as follows: Sect. 2 describes the environmental changes in pervasive computing and evolutionary threats. In Sect. 3 and 4, we present security clustering mechanisms, security context exchange and cooperation protocol for information sharing. We conclude the paper in Sect. 5.

* This work was supported by Korea Research Foundation Grant (KRF-2003-003-D00441).

** The corresponding author

2 Environmental Changes

On the basis of the paradigm shift of the mobile Internet, the network is expected form a loosely coupled and highly dynamic environments. The mobile users will be able to move around the network while connected to the environment: directly to other users or devices in their vicinity or indirectly (through the backbone) to the external. In this environment, the importance of user's location, service context, and various contents will be more evident than ever. But the management of such networks will have many dimensions in service provisioning, customization, and personalization, which will lead to a more complicated network revolution. Consequently, the network will face new security challenges because of the heterogeneity of the network, lack of centralized control, and presence of foreign users. Many vulnerabilities and weaknesses that have existed in the wired environment can easily be exploited in the new environment [4, 5].

The Presence of foreign users adds vulnerability on top of that openness and heterogeneity. The migration of foreign users especially have important implications on the network because they might have no pre-established secure association nor been authenticated and authorized to access the network through a decent mechanism. Users are potentially insecure in that they might conduct some kind of attacks intentionally or be victims of such attack and would act as slave for subsequent attacks. Furthermore, because they are basically mobile, the infected victims migrated into local network will be the security glitch while they are connected. This vulnerability introduces many risks to the network: First, the network becomes vulnerable to theft of data and DoS attacks. Second, network entities are effectively exposing the data on every remote system and creating thousands of unprotected entry points to the local network.

Security services in this open environment can be discussed in two different domains: *trust management* between communication entities and *system security* from various attacks such as distributed DoS (DDoS) or Interent worms. In this paper, we focus on how the pervasive computing environment can have network-wide defending mechanism against attacks. We propose *security clustering*, a distributed cooperation mechanism, to actively detect and put down various attacks conducted on the open network.

3 Security Clustering

3.1 Security Agent

The proposed secure clustering extends the agent-based management features to enable network-wide, reliable, and timely response to various and evolving attacks. The *security agent* plays a key role in managing the communication channel. The Fig. 1 shows a generalized pervasive computing environment in which we have shown three *security clusters* (C1, C2, and C3) that comprises one or more security agents and mobile users. The major function of the security agent is to establish secure inter-cluster communication channel and to exchange the *security context* with mobile users. The inter-cluster communication uses secure

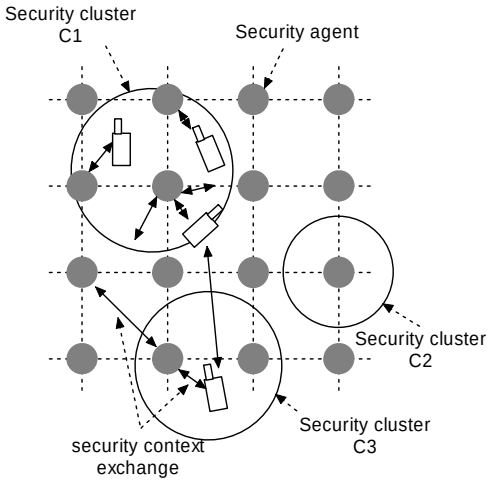

Fig. 1. Secure cluster

multicast protocol [6-8] to provide efficient and reliable information exchange mechanism to the security cluster.

3.2 Security Context Exchange

Since a lot of information for various attacks such as the records of suspicious behaviors and network events is required for constructing new signatures and uncovering relevant attacks, immediate countermeasure may not be guaranteed in many cases. The openness of pervasive computing environment seems to worsen the problem further in that there exist higher possibilities of widely conducted attacks on the open network.

The consequent main drawback of the current detection system would be the size of signature database that hinders real-time detection for various attacks. To increase the chance for successful detection, the detection system will need to collect more signatures for various attacks and its size will be much bigger that the mobile terminals can afford. Though the probability of successful detection is increased, the delayed detection procedure will lose QoS and user's interest.

To address these problems, canonical countermeasure architecture for pervasive computing should include these requirements:

- Cooperation of many network entities in scalable and robust manner
- Active detection for various attacks
- Management of attack signatures
- Authenticity of the exchanged context

To achieve these requirements, security clustering makes use of security context exchange mechanism between network entities.

The *security context* is a data structure that defines the sets of attributes or rules to describe the signatures of various attacks. The security context is differ from the *usual context* discussed in the pervasive computing, where the context means the environment, status, situation, and surroundings of a system or a user. The security context, on the other hand, is the security-specific information including the description of the on going attacks, status of the security alarm on the network, and security knowledge of each host. With the security context, one can detect the malicious activities of specific transaction or data. As shown in Fig. 1, users and agents exchange security context with each other. The secrecy of the context exchange relies on the secure multicast session that the security agents and users are involved.

4 Cooperation Protocol

4.1 Distributed Information Sharing

We use the *cooperation protocol* for exchanging security context and cooperating between multiple clusters. Because the security cluster should be able to respond timely to attacks, the volume of the security context database that each entity possess should also be minimized so that the local database scanning consume minimum horsepower and generate search result in time. For this purpose, each user that participates in the clustering possesses *differentiated* context for small amount of attack signatures. Each entity has specific policies that define the type and amount of the security context that it possesses. Therefore the security context database can be differentiated between participating entities. As a result, a host possess the *common security context*, which is the information of on going attacks or most recent security update, and small amount of the *differentiated security context* specific to that host.

A mobile user can request to the entire network for context for specific attack through the cooperation protocol. One who have proper context responses to the request.

4.2 Cooperation Protocol

We designed the *cooperation protocol* on the basis of secure multicast (shown in Fig. 2). The proposed protocol is as following.

Message Notations

- *REQ (Request)*: requests for a security context for specific fingerprints
- *RES (Response)*: response to the request with appropriate security context

Cooperation protocol

i. An initiator (a host) multicasts *REQ* message (Fig. 2(a)).
ii. Other entities of security cluster immediately perform detection procedures (Fig. 2(b)) with their security context database.

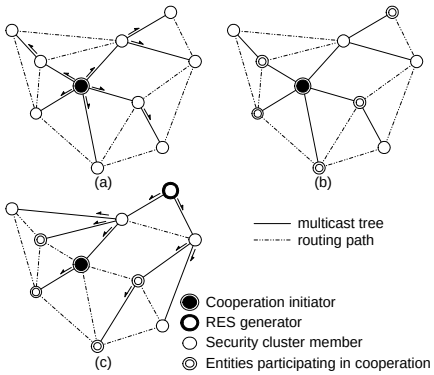

Fig. 2. Cooperation protocol

iii. Those who have no match silently ignores the request and does not respond.
iv. If one encounters appropriate context, then it multicasts the *RES* (Fig. 2(c)).
v. If one has no sufficient resources for the detection, just ignore and do not reply.

Main procedures of the cooperation protocol are as follows:

```
Collaboration_request :=
   if trigger(i) == TRUE then {
     if i ∈ common_context_pool /* common context */
       then countermeasure(i)
     else if i ∈ local_context_pool /* local differentiated context */
       then countermeasure(i)
     else {
       req = build_request(i)
       collaborative_request(req)
       if receive_response(r) then
         r -> common_context_pool /* update common context*/
         countermeasure(i) }}
Migration :=
   if migrating() == TRUE then {
     req = build_request(null) /* use null to indicate common context */
     collaborative_request(req) /* request for common context */
     if receive_response(res)
       then res -> common_context_pool } /* update common context */
Collaboration_response :=
   if receive_request(req)==TRUE then {/* is it a cooperation request? */
     if req.ctx ∈ common_context_pool then
       res = build_response(req.ctx, common_context_pool)
       send_response(res)
   } else if common_context_req() == TRUE then { /* or is it a common context
     res = build_response(common_context_pool)      request? */
     send_response(res) }
```

In the event of suspected operation or access, a host performs the *cooperation_request* procedure. The procedure check if the event corresponds to the common security context. If it does, then call *countermeasure* function. It it doesn't, the host call the *cooperate_request* function to initiate cooperation. On

successful detection, the *receive_response* returns with proper context. Mobile users migrated into the local network performs the *migration* procedure to receive the security context specific to the local network and to adapt to new environment. The hosts that participate in the cooperation performs the *cooperation_response* procedure. In this procedure, each cooperating host use their differentiated security context to generate proper response.

5 Conclusion

The purpose of the security clustering is to timely preempt the attack and quickly recover the systems on the basis of network-wide cooperative interwork. With the help of network-wide cooperation, relatively low powered mobile systems can have equivalent or higher level of security services than the ordinary single host or server that carry out self-reliant countermeasures. This will be very important feature for the highly mobile and heterogeneous environment of pervasive computing. Network-wide cooperation diminishes the migration of infected users and prevents the outbreak of attacks or viruses effectively. The continuous security context exchange and cooperation enables timely response to various attacks.

Although much work still remains to be done to design the security clustering in pervasive computing, the advantages of the security clustering suggests that it can be applied as a real-time countermeasure to the active attacks such as distributed DoS attacks and Internet worms.

References

1. M. Satyanarayanan: Pervasive computing: vision and challenges. IEEE Personal Communications, Vol. 8, Issue 4, August 2001.
2. L. Kagal, T. Finin, A. Joshi: Trust-Based Security in Pervasive Computing Environments. IEEE Computer, December 2001.
3. F. Stajano: Security for whom? The shifting security assumptions of pervasive computing. Proceedings of International Security Symposium 2002, LNCS 2609, (c) Springer-Verlag.
4. D. Moor, V. Paxson, S. Savage, C. Shannon, S. Staniford, N. Weaver: The Spread of the Sapphire/Slammer Worm. Technical Report, http://www.caida.org/analysis/security/sapphire, 2003.
5. S. R. White: Open Problems in Computer Virus Research. Virus Bulletin Conference, Munich Germany, Oct 22, 1998.
6. M.J. Moyer, J.R. Rao, P. Rohotgi: A Survey of Security Issues in Multicast Communications. IEEE Network, Vol. 13 Issue 6, Nov.-Dec. 1999.
7. C.K. Wong, M. Gouda, S.S. Lam: Secure Group Communication Using Key Graphs. Proceedings of ACM SIGCOMM'98, pp.68-99, September 1998.
8. R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas: Multicast security: A Taxonomy and Some Efficient Constructions. Proceedings of the IEEE INFOCOM'99, pp.708-716, 1999.