A New Definition of Multilevel Security*

Riccardo Focardi[1] Roberto Gorrieri[2] Roberto Segala[2]

[1]Dipartimento di Informatica
Università Ca' Foscari di Venezia
focardi@dsi.unive.it

[2]Dipartimento di Scienze dell'Informazione
Università di Bologna
{gorrieri,segala}@cs.unibo.it

1 Introduction

Multilevel security has played an important role in the literature because even such a simple requirement turns out to be very complicated and subtle from the definitional point of view. In the simplest formulation of multilevel security we start with users that can be in one of two levels, either high or low, and we impose that there must be no information flow from users at the high level to users at the low level.

Most of the definitions proposed in the literature (see, e.g., [1, 2, 3, 4, 5, 7, 8]) aim at restricting the set of behaviors of a system so that any low level user cannot observe any of the operations performed at the high level. The main problem with such an approach is that the link between the definitions and the fact that there is no information flow from high to low is not always clear or easy to get.

In this paper we propose a reversed approach to the definitional problem of multilevel security. Specifically, we start with the objects that we want to rule out, i.e., covert channels, and we say that an object or a collection of objects guarantee multilevel security if it cannot be used to implement the covert channels.

In the informal proposal given above there are several points that are left open since we have not said anything about what is a covert channel, what it means to use a collection of objects to implement a covert channel, and what it means to implement a covert channel. We analyze the three points above separately.

1. What is a covert channel?

   A covert channel is a process that informally allows a high level user to send some information to a low level user. There are several possible covert

*Work partially supported by MURST Progetto TOSCA and Progetto "Certificazione automatica di programmi mediante interpretazione astratta"

channels: a channel that can communicate any message, a channel that can communicate only once, a channel that communicates a single bit, a channel that communicates a single bit eventually, and so on. The point is that we can define multilevel security parameterized on the kind of covert channel that we want to rule out. The weaker is the covert channel, the stronger is the kind of security that we guarantee.

2. What does it mean to use a collection of objects to implement a covert channel?

   We can think of the objects as a collection of interfaces that in principle permit to the low level users to communicate to the high level users. If some high level users want to communicate something to some of the low level users, then the high level users collude, and similarly the low lever users collude, to use some of the interfaces together so that at the end they can achieve their goal. Thus, to use a collection of objects to implement a covert channel means to let some high level users, some low level users, and some objects interact so that they implement the covert channel. In our setting interaction is achieved through parallel composition.

3. What does it mean to implement a covert channel?

   There are several ways that can be used to define an implementation relation. From our point of view an implementation relation is an equivalence relation between processes. We can use trace equivalence, which is not deadlock sensitive, weak bisimulation or strong bisimulation [6]. In essence, our definition of multilevel security can be parameterized over the implementation relation.

2 The new definition

In this section we present our new definition of multilevel security. We will try to keep the underlying model as much abstract as possible, thus giving only the minimal necessary elements.

Consider a set of actions $\mathcal{A}$ partitioned into two sets $\mathcal{A}_L$ and $\mathcal{A}_H$ of low level and high level actions, respectively. Let $\mathcal{E}$ be the set of processes and $\mathcal{L}(\cdot)$ be a function that returns the set of actions that occur within a process. A process should have of course a syntax and a semantics, but we do not fix particular ones here. We just require that processes are provided with a parallel composition operator which allows them to communicate one another by synchronizing through actions of the same level. Communication is indeed fundamental as we are interested in analyzing possible way of transmitting information. Let now $\mathcal{E}_L$ be the set of processes that contain low level actions only, i.e.,

$\mathcal{E}_L = \{E \in \mathcal{E} \mid \mathcal{L}(E) \subseteq \mathcal{A}_L\}$, and let $\mathcal{E}_H$ be the set of processes that contain high level actions only, i.e., $\mathcal{E}_H = \{E \in \mathcal{E} \mid \mathcal{L}(E) \subseteq \mathcal{A}_H\}$.

Let C be a process which is meant to describe a channel, and let $\sim$ be an equivalence relation between processes representing the fact that two equivalent processes always show the same observable behaviour. Our objective is to define a set $\mathcal{S}_{C,\sim}$ of interface sets that are secure with respect to C. By secure we mean that it is not possible to use some interfaces to build a process equivalent to C according to $\sim$.

The next question is what we mean when we say build a process. We imagine a scenario where some users of the interface collude to communicate. That is, a high level process and a low level process use the interface or a collection of interfaces to communicate. Thus, we want to make sure that it is not possible to build C by composing in parallel some interfaces with some high level processes and some low level processes. We give the following definition:

Definition 2.1 *Let C be a process and $\sim$ be an equivalence relation between processes. Let K be a set of processes. We say that K is a set of secure interfaces with respect to C and $\sim$ iff the following conditions are satisfied:*

- (i) $C \notin K$;
- (ii) $\mathcal{E}_L \cup \mathcal{E}_H \subseteq K$;
- (iii) *K is closed with respect to parallel composition;*
- (iv) *K is closed with respect to level-preserving renaming, i.e., renaming according to injective functions ρ such that $\rho(Act_L) \subseteq Act_L$, and $\rho(Act_H) \subseteq Act_H$;*
- (v) *K is closed with respect to $\sim$.*

Denote by $\mathcal{S}_{C,\sim}$ the set of secure interfaces with respect to C and $\sim$. ■

We now informally explain the five conditions for K to be a set of secure interfaces, defined above. Point (i) states that the covert channel C is not in K. This is quite natural since C represents the information flow that we want to avoid. Condition (ii) states that all the high and low level processes are in K: indeed all the processes that use only high or low level actions cannot contain an information flows from high to low level, thus we include all of them in every set of secure interfaces. Condition (iii) is very important and guarantees that every process that is obtainable from K is already in K. In particular, if C would be obtainable by processes in K it should be in K but this cannot be true because of point (i). Hence, (i) and (iii) guarantee that C is not obtainable by processes in K through parallel composition. Condition (iv) states that we do not take too much in account the name of the actions: two processes that differs only in the name of the actions they perform should be both secure or both insecure. Finally, condition (v) represents the fact that we are mostly interested in maximal sets of secure interfaces since any subset of a set of secure interfaces is secure. Closure under $\sim$ gives us larger sets without compromising the definition.

We now prove that a set of secure interfaces with respect to C and $\sim$ is a set of secure interfaces also with respect to C and every stronger equivalence $\sim'$:

Proposition 2.2 *If $\sim' \subseteq \sim$ then $\mathcal{S}_{C,\sim} \subseteq \mathcal{S}_{C,\sim'}$.*

Proof. *Simply need to show closure under $\sim'$ which is trivial.* ∎

So, strengthening the equivalence notion potentially add new possible sets of secure interfaces. This could look counterintuitive at a first glance. However, note that if we consider a "weak" equivalence notion $\sim$, the class of processes which are equivalent to the channel C will be potentially bigger than the one induced by a stronger $\sim'$. Thus, with a weak equivalence it will be, in a sense, easier to construct a process equivalent to the insecure channel C.

3 Separability and channel ordering

It is now quite natural to study the concept of *separability* in our setting. We first need the following closure operator:

Definition 3.1 *For a set of processes $\mathcal{F}$, let $Close_\sim(\mathcal{F})$ denote the closure of $\mathcal{F}$ under $\sim$, parallel composition, and level-preserving renaming. If $K = Close_\sim(\mathcal{F})$, then we say that K is* generated *by $\mathcal{F}$, and we call $\mathcal{F}$ a* generator *of K.* ∎

We can prove the following:

Proposition 3.2 *Fix C and $\sim$. Then*

1. *for each $K \in \mathcal{S}_{C,\sim}$, K is a generator of K;*
2. *$\mathcal{S}_{C,\sim}$ is empty iff $C \in Close_\sim(\mathcal{E}_H \cup \mathcal{E}_L)$;*
3. *if $\mathcal{S}_{C,\sim}$ is not empty, then $Close_\sim(\mathcal{E}_H \cup \mathcal{E}_L) \in \mathcal{S}_{C,\sim}$;*
4. *for each $K \in \mathcal{S}_{C,\sim}$, $Close_\sim(\mathcal{E}_H \cup \mathcal{E}_L) \subseteq K$.* ∎

In particular, the second point states that it is possible to find at least one set of secure interfaces only if the channel C cannot be decomposed into the parallel composition of processes of $\mathcal{E}_H \cup \mathcal{E}_L$, i.e., only if C communicates something. Furthermore, the set $Close_{\sim}(\mathcal{E}_H \cup \mathcal{E}_L)$ is always secure, by points 3 and 4. We denote the set $Close_{\sim}(\mathcal{E}_H \cup \mathcal{E}_L)$ by Sep and we call it the set of *separable* processes, i.e., the processes that can be obtained as a composition (possibly with renaming) of high and low level processes. Note that a high level process and low level one cannot communicate as they have no actions in common.

It is now interesting to see how we can reason on different kinds of covert channels. In particular, it would be interesting to have a notion of ordering of channels reflecting their power. The next definition states that a channel C is not more powerful than C' if it is possible to implement C using C'.

Definition 3.3 *Let C, C' be two channels. We say that C is not more powerful than C', denoted $C \leq_{\sim} C'$ if $C \in Close_{\sim}(\mathcal{E}_H \cup \mathcal{E}_L \cup \{C'\})$.* ∎

We can now prove that if a set K of interface is secure with respect to a channel C, that is is also secure with respect to every C' such that $C \leq_{\sim} C'$. Formally:

Proposition 3.4 *if $C \leq_{\sim} C'$ then $\mathcal{S}_{C,\sim} \subseteq \mathcal{S}_{C',sim}$.* ∎

Indeed a set of interfaces K which can be used to construct C' (i.e. which is not secure with respect to C'), can be easily used to construct also C. This intuitively means that C can do nothing more with respect to C' in transmitting information from high to low level.

4 Future work

We are presently studying security with respect to certain particular channels: for example a perfect channel from high to low (C_P), a channel that outputs arbitrary bits and then eventually may output the input bit forever (C_F). It is, for example, interesting to study for which equivalence relations $\sim$ we have that $C_F \leq_{\sim} C_P$.

We are also considering existing definitions of multilevel security in order to study for what channels they are secure. For properties that are closed with respect to parallel composition, level-preserving renaming and some basic equivalence $\sim$ (like, e.g., some of the properties reported in [2, 3]) the task is quite simple. It is sufficient to check if a certain channel C satisfies the property or not, in order to prove that such a property is indeed in $\mathcal{S}_{C,\sim}$, i.e., that all the processes that satisfy the property are secure interfaces with respect to C and

$\sim$. This could allow some kind of classification of existing properties, based on which covert channels they are indeed able to implement. Note that this will presumably produce a lattice of properties, as we could also consider channels that are not ordered through the $\leq_\sim$ relation.

An interesting property, naturally induced by our new proposal, is the following: Let $\mathcal{E}_{LH}$ be the processes that contain both high level and low level actions but are always enabled on the actions from Act_L. These processes can always perform any of the low level actions, and thus cannot communicate any information arising from the high level. Intuitively if we consider the set $Sep_{LH} = Close_\sim(\mathcal{E}_H \cup \mathcal{E}_L \cup \mathcal{E}_{LH})$ we should discover that Sep_{LH} is secure with respect to any channel that communicates from high to low. We are presently verifying this claim against some particular covert channels.

References

[1] R. Focardi. "Comparing Two Information Flow Security Properties". In *Proceedings of Ninth IEEE Computer Security Foundation Workshop, (CSFW'96), (M. Merritt Ed.)*, pages 116–122. IEEE press, June 1996.

[2] R. Focardi and R. Gorrieri. "A Classification of Security Properties for Process Algebras". *Journal of Computer Security*, 3(1):5–33, 1994/1995.

[3] R. Focardi and R. Gorrieri. "The Compositional Security Checker: A Tool for the Verification of Information Flow Security Properties". *IEEE Transactions on Software Engineering*, 23(9), September 1997.

[4] J. A. Goguen and J. Meseguer. "Security Policy and Security Models". In *Proceedings of the 1982 Symposium on Security and Privacy*, pages 11–20. IEEE Computer Society Press, April 1982.

[5] D. M. Johnson and F. J. Thayer. "Security and the Composition of Machines". In *Proceedings of the Computer Security Foundations Workshop*, pages 72–89, June 1988.

[6] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[7] A. W. Roscoe, J. C. P. Woodcock, and L. Wulf. "Non-interference through Determinism". In *Proceeding of European Symposium on Research in Computer Security 1994 (ESORICS'94)*, pages 33–53. Springer-Verlag LNCS 875, 1994.

[8] J. T. Wittbold and D. M. Johnson. "Information Flow in Nondeterministic Systems". In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pages 144–161. IEEE Computer Society Press, 1990.