Providing Intrusion Tolerance With ITUA[1]

Tod Courtney, James Lyons, HariGovind V. Ramasamy,
William H. Sanders, and Mouna Seri

Center for Reliable and High-Performance Computing
Coordinated Science Laboratory and
Electrical and Computer Engineering Department
University of Illinois at Urbana-Champaign
{tod, jlyons, ramasamy, whs, seri}@crhc.uiuc.edu

Michel Cukier

Center for Reliability Engineering
Department of Materials and Nuclear Engineering
University of Maryland at College Park
mcukier@eng.umd.edu

Michael Atighetchi, Paul Rubel, Christopher Jones,
Franklin Webber, Partha Pal, and Ronald Watro

BBN Technologies
{matighet, prubel, ccjones, fwebber, ppal, rwatro}@bbn.com

Jeanna Gossett
The Boeing Company

jeanna.m.gossett@boeing.com

1. Introduction

The goal of the Intrusion Tolerance by Unpredictable Adaptation (ITUA) project is to develop a middleware-based intrusion tolerance solution that helps applications survive certain kinds of attacks. The goal of this paper is to introduce several features of the ITUA project that are illustrated in a use scenario. Section 2 explains the importance of unpredictable adaptation in intrusion tolerance. Section 3 presents the assumptions made by the ITUA project. The remaining sections address the ITUA architecture. In Section 4, we give a brief overview of the overall architecture. Section 5 focuses on the group communication system. Section 6 presents the application, which will be partly ported to ITUA, thus illustrating how intrusion tolerance can be added through ITUA. This paper ends with a brief description of one scenario that uses the ITUA middleware.

2. Unpredictability in Intrusion Tolerance

Adaptation is crucial for intrusion tolerance. In order to survive an attack, applications must adapt to the damaged environment and to changes in the quality of resources. Adaptation should be provided at various levels in the system. ITUA supports both in-band and out-of-band adaptation. With in-band adaptation, inter-object interaction is intercepted, and application-level behavior is altered. With out-of-band adaptation, intrusion response and recovery actions that involve managing and configuring system resources are independent from the application's inter-object interaction. Both in-band and out-of-band adaptation is managed by the QuO [Loy98] adaptive middleware.

However, in the case of sophisticated attacks carried out in multiple stages, the resilience provided by adaptation can easily be circumvented by an adversary who can predict the adaptive response. Therefore, the ITUA project adds uncertainty in its intrusion tolerance technology so that the adaptive responses become unpredictable to the attacker. For instance, at the application level, an object could attempt to communicate with a remote object other than the one the attacker was more likely to expect. At the system resource level, after an attacker has killed a replica on a host, the replacement replica could be started on a host that is chosen in a non-deterministic manner.

3. The ITUA Model

A system implemented using the ITUA approach consists of a set of *security domains*. A security domain implements a boundary that attackers have difficulty crossing. For example, a host can be a security domain if an attacker with privileges on one host is not automatically granted privileges on other hosts. Another example of a domain is a LAN with firewalls that insulate it from other LANs.

We assume that an attacker proceeds by infiltrating security domains and by corrupting processes within those domains. In the worst case, a corrupt process can behave arbitrarily.

A key assumption in the ITUA model is that attacks will be *staged*. In a staged attack, the attacker infiltrates some domains before others. This might happen because the attacker needs time to explore the topology of interconnections between security domains, or because domains differ and attacks on them take unpredictable amounts of time. The assumption of staged attacks gives the defense some time in which to react to an attack.

Another assumption is that intrusion detection, while imperfect, is reliable and accurate enough that the system's defenses are unlikely to be overwhelmed by corrupt processes inserted undetectably. An assumption about intrusion detection is necessary because a corrupt process can behave arbitrarily. Without detection, an attacker could infiltrate domains in stages, silently corrupting processes, and then suddenly cause every process to stop simultaneously. With detection, the defense may be able to react.

4. ITUA Architecture Overview

The architecture for the ITUA infrastructure consists of replicas, managers, and subordinates located in various groups. This section will detail the main components of the ITUA architecture.

[1] This research has been supported by DARPA Contract F30602-00-C-0172.

Each host runs either a manager or a subordinate. Each security domain has one host that runs a manager; the rest of the hosts in that domain run subordinates. All managers form a process group called the *manager group*. All the subordinates in a domain and the manager of that domain form a group called a *subordinate group*. A subordinate's two principal responsibilities are *security advising* and *replication management*. A subordinate's actions and decisions are local, i.e., they involve resources associated with the host on which it runs and its network interfaces.

In the *security advisor* role, a subordinate makes use of multiple local sensor-actuator loops to 1) collect information about potential intrusions and anomalous events, 2) make a quick local reaction to the observed event, and 3) provide the security domain manager and the local replication management operation with host-specific information. We currently have implementations for two loops: the PortAttack→Firewall loop and the Tripwire→Backup loop. The PortAttack→Firewall loop detects abnormal activity on TCP ports using "snort," and adapts by dynamically changing the local firewall. The Tripwire→Backup loop monitors the file system and restores it from a save backup in case an attacker deletes/modifies files.

In the *replication management* role, subordinates are responsible for starting or killing replicas of application components. Both actions require several steps. Once replicas detect a faulty replica, they send a point-to-point message containing a proof of the failure to their respective subordinates. The subordinates then multicast the message in their subordinate groups. Finally, the managers multicast the message in the manager group so all managers become aware of the failure. The manager leader then initiates the decision (e.g., killing the faulty replica, starting a new replica). Once the decision is agreed on in the manager group, the managers send the decision (with the proof of the faulty replica) to their subordinate groups. The subordinate selected by the manager then performs its task (e.g., killing the replica on its host, starting a new replica on its host).

5. Group Communication System

There are many communicating groups in the ITUA architecture, including replication groups, subordinate groups, and the manager group. All of these process groups share some common concerns, like maintaining consistent group membership and ensuring that all proper processes in the group receive all multicast messages in the same order. Therefore, a prototype intrusion-tolerant group communication system has been developed. We devised intrusion-tolerant versions of several key group communication protocols, namely those that provide reliable multicast, total ordering within a group, and group membership operations. We then inserted implementations of these protocols into an existing crash-tolerant group communication system, C-Ensemble [Hay01].

5.1 Group Membership Protocol

The intrusion-tolerant group membership protocol [Ram02] ensures that all proper processes maintain consistent information about the membership of the group despite intrusions, provided that at most one-third of the members of the group are corrupt. It is responsible for maintaining group membership information, removing corrupt processes from the group, and joining new processes into the group. To provide these functions, the protocol relies on the reliable multicast protocol to deliver the messages it sends to maintain group membership.

We incorporated several checks into the layers of the modified C-Ensemble protocol stack to detect deviations from protocol specifications. These deviations are reported as *suspicions* to the group membership protocol, which multicasts them to the group. If more than one-third of the members of the group suspect some process to be corrupt, the group membership protocol initiates a three-phase view installation procedure, at the end of which the membership information at all correct processes will be modified to exclude all processes that are suspected by more than one-third of the group.

5.2 Reliable Multicast Protocol

In the manager group, subordinate groups, and replication groups, much of the information that is shared needs to reach all processes, even if there are intrusions and/or an unreliable network. The reliable multicast protocol [Pan01] in the group communication system addresses this issue by using message buffering, sequence numbers, positive and negative ACKs, and cryptographically signed messages. When a group member wants to multicast a message, it buffers a copy of that message, assigns to it the next available sequence number, creates a digitally signed digest of the message, and sends the {digest, sequence number, message} information to the other group members. Other group members who receive this information reply with a signed ACK to the sender. The sender collects ACKs from more than two-thirds of the group, and finally sends the message with these ACKs, which serve as a proof that a majority of the correct members have accepted this message as valid.

5.3 Total-Ordering Protocol

When using the ITUA infrastructure, application objects that are replicated to provide intrusion tolerance must maintain a consistent notion of state. To support that need, the exchange of information between the replicas is totally ordered. We devised a total ordering protocol [Pan01] for group multicasts that can be made very efficient in the special case in which all of the communicating processes in the group are replicas.

In the protocol, total ordering is obtained by associating the group members with a set of global sequence numbers. The processes then give individual sequence numbers to the messages they multicast. For all processes, messages are delivered in the order of the sequence numbers. This protocol can be seen as an example of a "born-order" protocol [Bir96] for total ordering, in which messages contain information about the order in which they should be delivered.

6. IEIST Application

We now will give a very short overview of the IEIST (Insertion of Embedded Infosphere Support Technologies) application (developed by Boeing), part of which is being ported to the ITUA architecture to test and transition ITUA intrusion tolerance technology. The goal of IEIST [Cor01] is to improve the exchange of information between deployed tactical elements and information nodes worldwide. IEIST focuses on the development of off-board software agents designed to augment embedded tactical systems and plug into the evolving Joint Battlespace Infosphere (JBI), while still providing interoperability with legacy systems and communication links. These Guardian Agents (GA) support nodes that will be re-locatable anywhere within the JBI and will allow the use of readily available off-board processing and networking resources to augment the scarce embedded resources.

A typical IEIST use case would be the interaction between an F-15 GA, a UCAV (Unmanned Combat Air Vehicle) GA, and a Discovery and Navigation service (D/N) that connects subscribers to publishers of information based on geographic region of interest. The first part of the scenario begins when a UCAV has registered as a publisher with a D/N for a particular region. An F-15 flying over the region then also registers with the D/N as a subscriber. At that time, the D/N communicates to the F-15 that the UCAV is a publisher in the region. The F-15 then subscribes to the UCAV. When the UCAV notices some activity in the region, it then sends the new information to the F-15.

This D/N service is being protected using ITUA technology. As a result, the service is replicated on different nodes located in different security domains. The goal of the use scenario presented in the next section is to show that the use of ITUA allows the D/N service to work (and thus allows the F-15 and the UCAV to communicate) even if some replicas of the service are faulty.

7. ITUA Use Scenario for IEIST

The purpose of the use scenario is to illustrate how, through the use of adaptation and unpredictability at various system levels, intrusion tolerance has been added to the parts of IEIST that have been ported to the ITUA.

The use case relates to the scenario presented in Section 6, in which the F-15 and the UCAV need to exchange some information using the D/N service. We can show that the communication is not interrupted even when intrusions occur at various levels, leading to the following effects.

- *The application level*. A D/N service replica can become arbitrarily faulty, e.g., crashing, sending the wrong information to the other D/N service replicas, or sending the right information to a wrong D/N service replica. These effects will be detected and tolerated by the other D/N service replicas using the group communication system (i.e., use of signatures, ACKs, and thresholds).
- *The manager level*. A manager becomes arbitrarily faulty and decides, for example, to kill a non-faulty replica, to start a new replica without a reason, or to exclude a security domain when no intrusion was detected. These effects are detected and tolerated by the other managers. The faulty manager will be excluded from the manager group, and the security domain managed by the faulty manager will be considered corrupted.
- *The subordinate level*. A subordinate can become arbitrarily faulty, e.g., it does not communicate the information received by a D/N replica, it sends out information for killing the wrong replica, or it kills or starts replicas on its host without receiving an order from the managers. These effects will be detected and tolerated by the subordinates in the subordinate group and the managers in the manager group. For example, the subordinate may be excluded from the subordinate group and the replicas started by this subordinate will be killed.
- *The network level*. We tolerate intrusions on the network and OS level through the use of rapid reaction loops, which take actions to confine an attacker upon getting notification from the IDSs. The loops combine different kinds of information from various IDSs, analyze the data, and take local action to tolerate the attacks.
- *The group communication level*. In addition to tolerating the effects described for the D/N service replicas, the group communication system can also tolerate delays, wrong ordering, different messages sent to different replicas, and so forth. These effects are detected and tolerated at the group communication level.

When we demonstrate, we inject some of those effects. We see that the F-15 and the UCAV continue communicating without being affected by these intrusions.

Acknowledgments

The authors would like to thank Prashant Pandey for his contribution on the reliable multicast and total-ordering protocols. We also would like to thank Richard Schantz and Joseph Loyall for their help on this paper. Finally, we thank Jenny Applequist for her careful review.

More information on the ITUA project can be found at
http://www.dist-systems.bbn.com/Projects/ITUA and
http://www.crhc.uiuc.edu/PERFORM.

References

[Bir96] K. P. Birman, *Building Secure and Reliable Network Applications*, Manning, 1996.

[Cor01] D. Corman, T. Herm, C. Satterthwaite, “Transforming Legacy Systems To Obtain Information Superiority,” in *6th ICCRTS, CCRP*, Annapolis, MD, June 2001.

[Cuk01] M. Cukier, J. Lyons, P. Pandey, H. V. Ramasamy, W. H. Sanders, P. Pal, F. Webber, R. Schantz, J. Loyall, R. Watro, M. Atighetchi, and J. Gossett, “Intrusion Tolerance Approaches in ITUA,” FastAbstract in *Supplement of the 2001 International Conference on Dependable Systems and Networks*, Göteborg, Sweden, July 2001, pp. B-64 to B-65.

[Hay01] M. G. Hayden, “Ensemble Reference Manual,” Cornell University, 2001.

[Loy98] J. P. Loyall, D. E. Bakken, R. E. Schantz, J. A. Zinky, D. A. Karr, R. Vanegas, and K. R. Anderson, “QoS Aspect Languages and Their Runtime Integration,” *Lecture Notes in Computer Science*, vol. 1511: *Proc. Fourth Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers (LCR98),* May 1998, Pittsburgh, PA. Springer-Verlag.

[Pan01] P. Pandey, “Reliable Delivery and Ordering Mechanisms for an Intrusion-Tolerant Group Communication System,” M.S. Thesis, University of Illinois at Urbana-Champaign, 2001.

[Ram02] H. V. Ramasamy, “Group Membership Protocol for an Intrusion-Tolerant Group Communication System,” M.S. Thesis, University of Illinois at Urbana-Champaign, 2002.