

# Demonstrating Intrusion Tolerance With ITUA\*

ITUA Team:

BBN Technologies, The Boeing Company,  
University of Illinois at Urbana-Champaign, University of Maryland at College Park

## 1 Introduction

Intrusion Tolerance by Unpredictable Adaptation (ITUA) is funded under the OASIS (IPTO) project and was started in July of 2000. The goal of the ITUA project is to develop a middleware-based intrusion tolerance solution that helps applications survive certain kinds of attacks. This document describes the technology developed in the project and the capabilities of our demonstration. The demonstration shows how the mechanisms developed in the project can be combined to make a survivable application. Section 2 explains how ITUA realizes intrusion tolerance through unpredictability. Section 3 presents the ITUA architecture. Section 4 introduces the context of the demo and section 5 concludes with the use of ITUA in the demonstration.

## 2 Unpredictability in Intrusion Tolerance Technical Overview

Adaptation is crucial for intrusion tolerance. In order to survive an attack, applications must be able to adapt to damaged environments and to changes in the quality of resources. Adaptation should be provided not only at various levels in the system, but also with a range of responses. These responses can be done at many levels. Rapid reaction loops, communication mechanisms, ITUA gateways, and ITUA management entities are four mechanisms used by ITUA in order to react to changes.

Rapid reaction can occur at the host or application level. Communication between applications can also be adapted as the environment changes. Working together,

---

\*This research has been supported by DARPA Contract F30602-00-C-0172

replicated applications using ITUA gateways can locate faulty replicas and report this to ITUA replication managers. These managers can then kill the faulty replica and coordinate to replace it with a good replica. In addition to these responses we have also worked on formally and informally validating our technology.

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 are 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 expects. 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 ITUA Architecture Overview

The ITUA architecture is designed to support adaptation. It is composed of security domains, which are groups of hosts whose security is tightly coupled. ITUA managers run on hosts in these domains. The ITUA managers, and the loops they contain, are responsible for carrying out two functions: security advising and replication management.

In the security advisor role, a manager 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 managers 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 files from a secure backup in case an attacker deletes or modifies files.

In the replication management role, managers are responsible for starting or killing application replicas as well as policing themselves. In order to replicate applications the managers use ITUA gateways. These gateways ensure consistency among replicas as well as tolerate and report failures using intrusion-tolerant group communication systems. Failures are reported by the gateways to the managers, who can then take action to remove faulty replicas.

## 4 IEIST Application

We now will give a very short overview of the IEIST (Insertion of Embedded Infosphere Support Technologies) application (developed by Boeing), which will be featured in the demonstration. The goal of IEIST 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 relocatable 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, and the focus of our demonstration, is 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 re-

gion, it 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.

## 5 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 IEIST using ITUA. We will use the scenario presented in Section 4, in which the F-15 and the UCAV need to exchange some information using the D/N service in our demonstration.

In this demonstration we replicate the D/N and watch as the F-15 travels along its route. There are four copies of the D/N running and also two stand-alone/non-replicated D/Ns. These stand-alone D/Ns are used if the loops feel the stand-alone D/Ns are a better choice than the replicas, perhaps due to a common-mode failure.

Each replica is managed by an ITUA manager. Each D/N replica has a fault injector. The managers also have fault injectors, which can be used to inject failures into the system. The managers can be instructed to ignore failure notifications. This is useful for showing that failures are tolerated while not having to wait while the managers kill the faulty replica and restart it.

During the demonstration we show that the communication between the F-15 and the UCAV is not interrupted even when intrusions occur at various levels. We demonstrate responses to the following types of failures or attacks: host level, application level, gateway level, manager level, and network level.

### 5.1 Host Level Failures

The first failure will involve the host's file system. We will remove a monitored file. The loops will notice the unauthorized file system modification and replace the file, while noting the problem. The fighter will continue to execute its mission while this is occurring.

## 5.2 Application Level Failures

The next type of failure tolerance demonstrated will be D/N application failures. For the time being the managers will ignore reported faults. Using the D/N fault injector we will insert corrupt data in one of the replicas. The F-15 will not notice this problem, as the other D/N replicas will mask the failure. The replicas will report the failure to the managers.

## 5.3 Gateway Level Failures

The next failures will come from the ITUA gateways themselves. We can inject duplicate messages, unsigned messages, corrupt signatures, and even bypass the protocol. These will all be noted by the replicas and will be masked. The F-15 will continue to successfully carry out its mission.

## 5.4 Manager Responses

Next we will test that manager will kill and restart replicas. We instruct the managers to deal with failures. We will inject a fault into a replica and watch as the managers note this failure and kill the replica. The replica will be restarted on an available manager.

## 5.5 Manager Level Failures

The ability to tolerate failures of managed entities is good but what if the managers themselves misbehave? While all the replicas are running we can instruct a manager to attempt to start a new replica. The managers will note that this is not appropriate and no replica will be started.

We can also inject a failure into a replica and command the manager in charge of it not to kill the replica. The other managers will notice this failure and will note that the manager is faulty. Finally, we can inject a fault into a replica and instruct the manager to report this new fault as a previously seen fault. This will be caught by the other managers who will disregard the message.

## 5.6 Network Level Attacks

To test the PortAttackFirewall loop we will attack a manager across the network. The loops will notice this attack

and will block further communications from the attacking host.

## 5.7 Replication Failure

Finally, we will use the backup replicas. By removing all the monitored files at once we will convince the loops that all the replicas are vulnerable. The loop will move the F-15 over to using the non-replicated D/Ns.

After showing these defenses will we kill all the D/Ns and watch as the F-15 misses message and fails in its mission.

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>.