Chapter XLVII
Software Component Survivability in Information Warfare

Joon S. Park
Syracuse University, USA

Joseph Giordano
Air Force Research Laboratory, USA

ABSTRACT

The need for software component survivability is pressing for mission-critical systems in information warfare. In this chapter, we describe how mission-critical distributed systems can survive component failures or compromises with malicious codes in information warfare. We define our definition of survivability, discuss the survivability challenges in a large mission-critical system in information warfare, and identify static, dynamic, and hybrid survivability models. Furthermore, we discuss the trade offs of each model. Technical details and implementation of the models are not described in this chapter because of space limitations.

INTRODUCTION

As information systems became ever more complex and the interdependence of these systems increased, the survivability picture became more and more complicated. The need for survivability is most pressing for mission-critical systems in information warfare. When components are exported from a remote system to a local system under different administration and deployed in different environments, we cannot guarantee the proper execution of those remote components in the current run-time environment. Therefore, in the run time, we should consider component failures (in particular, remote components) that may occur due to poor implementation, during integration with other components in the system, or because of cyber attacks. Although advanced technologies and system architectures improve the capability of today's systems, we cannot completely avoid threats to them. This becomes more serious when the systems are integrated

with commercial off-the-shelf (COTS) products and services, which typically have both known and unknown vulnerabilities that may cause unexpected problems and that can be exploited by attackers trying to disrupt mission-critical services (Kapfhammer, Michael, Haddox, & Colyer, 2000). Organizations, including the Department of Defense (DoD), use COTS systems and services to provide office productivity, Internet services, and database services, and they tailor these systems and services to satisfy their specific requirements. Using COTS systems and services as much as possible is a cost-effective strategy, but such systems—even when tailored to the specific needs of the implementing organization—also inherit flaws and weaknesses from specific COTS products and services that are used. Therefore, we need reliable approaches to ensure survivability in mission-critical systems that must rely on commercial services and products in a distributed computing environment.

Definitions of survivability were introduced by previous researchers (Knight & Sullivan, 2000; Lipson & Fisher, 1999). We define survivability as the capability of an entity to continue its mission even in the presence of damage to the entity (Park, Chandramohan, Devarajan, & Giordano, 2005). An entity ranges from a single software component (object), with its mission in a distributed computing environment, to an information system that consists of many components to support the overall mission. An entity may support multiple missions.

The damage caused by cyber attacks, system failures, or accidents, and whether a system can recover from this damage (Jajodia, McCollum, & Ammann, 1999; Knight, Elder, & Du, 1998; Liu, Ammann, & Jajodia, 2000), will determine the survivability characteristics of a system. A survivability strategy can be set up in three steps: protection, detection and response, and recovery (Park & Froscher, 2002). To make a system survivable, it is the mission of the system rather than the components of the system. This implies that the designer or assessor should define a set of critical services the system must provide in order to fulfill the mission. In other words, they must understand what services should be survivable by the mission and what functions of which components in the system should continue to support the system's mission.

In this article, we focus on the survivability of mission-critical software components downloaded on the Internet. We assume that all software components are susceptible to malicious cyber attacks or internal failures. Cyber attacks may involve tampering with existing source code to include undesired functionality (e.g., Trojan horses), or replacing a genuine component with a malicious one. When using such components, particularly in mission-critical applications in information warfare, we must check to see if the component was developed by a trusted source, and whether the code has been modified in an unauthorized manner since it was created. Furthermore, we should check to see if the component is functioning in an expected way. If all these conditions are satisfied, we call it "trusted component sharing."

CHALLENGES TO SOFTWARE SURVIVABILITY IN A MISSION-CRITICAL SYSTEM

Typically, an application running at an enterprise level may span more than one organization. Figure 1 shows an example of a distributed application that spans multiple organizations. The figure depicts three organizations interconnected to form a large enterprise-computing environment. In the real world, there may be more than two or three organizations connected to form a large enterprise, and some of the organizations in the enterprise may provide specialized services that other organizations do not provide (e.g., Department of Homeland Security). In the figure, for example, components in Organizations 1 and 3 are involved in application X. In this example the application running in Organization 3 downloads necessary components for some special features that it lacks. These components are dynamically downloaded from remotely administered hosts (in Organization 1 in the example) and run locally. This situation becomes complex when one must administer components downloaded from disparate administrations. For instance,