

Motivation

Building systems that are both secure and usable is a persistent challenge. We need to ensure that information can be both protected and shared. Participation in online information exchanges (commercial, personal, and civil) requires that people be able to trust the infrastructures that support these exchanges. Advances in security and cryptography technologies has provided tools that enhance the *theoretical* security of information systems -- that is, the level of secure communication and computation that is technically achievable. However, *effective* security depends not only on the technology of security, but on its application. An unusable security system is as good as no security at all.

One approach is to make security transparent -- to embed it in the system at a low enough level that users need never be aware or concerned with it. However, we believe that the variety of user needs and usage settings is so large that decisions about security cannot be hidden. End users do not simply need security, but rather need *the ability to make decisions about their security needs*. Informed decision making matches the capabilities of the system to the needs of the moment. Empirical investigations of end-user behavior

with networked and distributed systems suggests that this decision-making process poses many practical problems.

Our goal is to create a technical infrastructure which makes visible the configuration, activity, and implications of available security mechanisms, thereby allowing end users to make informed security choices resulting in increased *effective* security.

Approach

Our approach is based on two technological elements.

First, we use *visualization* technologies to allow people to inspect and assess complex information spaces, including the behavior of applications and infrastructures. Visualization technologies exploit the features of the human perceptual system to help people recognize patterns, exceptions, and correlations. In particular, we are building on the Vavoom system, which creates dynamic visualizations of the internal behavior of Java programs.

Second, we use *event architectures* to route, combine, and process information from many different system elements. This event-based approach allows our system to be both extensible and scalable, across devices and across platforms. In particular, we are using YANCEES, an extensible and configurable publish/subscribe event notification service based on plug-ins.

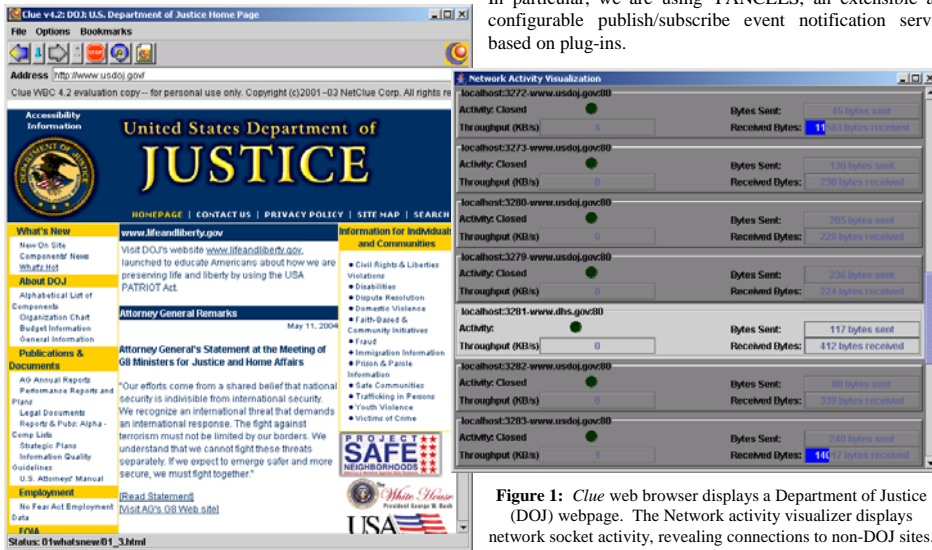


Figure 1: Clue web browser displays a Department of Justice (DOJ) webpage. The Network activity visualizer displays network socket activity, revealing connections to non-DOJ sites.

By combining these approaches, we are creating dynamic visual depictions of system behavior as the basis of informed decision-making. The challenge is to convey the information from these systems to the end user at such a time and in such a way that they can make timely and effective use of it.

Figure 1 shows a demonstration of a web browser executing on a prototype Swirl infrastructure. A security network activity visualization displays socket activity, and reveals the multiple network connections corresponding to this single web page load. The architecture for this demonstration is shown in Figure 2. The Vavoom class loader dynamically rewrites the Java bytecode to instrument it without access to the source code. The YANCEES event service integrates information from different parts of the system and provides inference mechanisms for event sequence detection.

Testbed

We are currently extending this prototype to support a more complex testbed application, a peer-to-peer collaborative workspace. This workspace is based on standard protocols (IETF Zeroconf and WEBDAV) to provide automatic

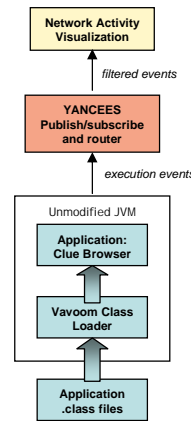


Figure 2: Architecture for the demonstration depicted in Figure 1. The Vavoom class loader instruments the Clue web browser to produce system event notifications without modification to the source. Yancees receives these events and relays them to relevant visualizers that have subscribed to different types of events.

Contact Information

Professor Paul Dourish
 Professor David F. Redmiles
 Institute for Software Research
 University of California
 Irvine, California 92697-3425
 {redmiles, jpd}@ics.uci.edu
 +1-949-824-1812, 3823}
 Fax 949-824-1715

To learn more about the Swirl project, please visit the website:
<http://isr.uci.edu/projects/swirl/>

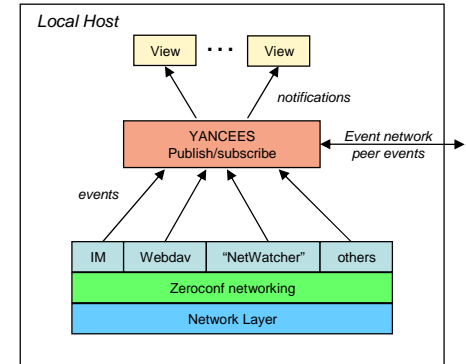


Figure 3: Future Swirl infrastructure supporting security awareness in a mobile ad-hoc networking environment.

discovery and sharing of information resources in support of collaborative work, as shown in Figure 3. The target for this testbed is collaborative groups using ad hoc, mobile wireless devices including laptops, tablets, and handheld PCs. The testbed supports a wide range of potential uses and settings, and so challenges us to create visualizations that allow users to effectively assess security needs and threats in a dynamic and complex environment. Significant issues include control over the degree of sharing, supporting dynamic network topologies, managing persistence, and balancing awareness and potential distraction.

Further Work

Our testbed application is designed to provide a realistic but constrained environment for exploring the effectiveness of our ideas and the technological challenges that they create. Beyond this initial exploration, we are investigating the ways to use dynamic visual monitors to understand a wider range of conventional security practices in everyday operating environments.

This material is based upon work sponsored by the National Science Foundation under grant number 0326105 and by the Intel Corporation. The content of the work should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of either organization.