# 5-899 / 17-500 Usable Privacy and Security

## Lecture Notes for Jan 31st, 2006

### Chameleon

Chameleon is a desktop interface developed to minimize damage by malware (viruses, trojan horses and worms) through the utilization of roles for access control. The security model leverages physical aspects (such as how a house is organized) to assist end users in understanding the file organization environment in Chameleon.

Product development process of Chameleon illustrated the use of good usability practices. For example, Low-Fidelity prototyping (Li-Fi) was used extensively in conjunction with user studies to help create the interactive prototype.

Security of Chameleon is based on role based access control (RBAC). For further information, visit the following website at csrc.nist.gov/rbac.

What is RBAC? It defines that users are assigned roles based on their individual responsibilities. Since roles are created for jobs functions in an organization, users can be easily reassigned from one role to another. In addition, the roles can be granted new permissions or have their permissions revoked as well.

The five standard roles are vault, communications, default, testing and system. So far, only one Visual Basic prototype was developed. Roles are allocated before end users are giving access to their accounts.

There are difficulties in explaining to people about roles. Firstly, it is very hard. Next, window borders are subtle and easy to miss. Lastly, the desktop environment combines multiple roles simultaneously.

What happens when the system is comprised? The solution is to delete the roles that are comprised and reassign new roles to the end users.

What roles or applications should be allowed in the Chameleon environment? It makes sense to group communications together. For example, John is associated with phone, email, instant message or web browsing. In this environment, people are considered the endpoints rather than the mechanism used.

**Testing role is used when the installed programs are not trustworthy. It is almost similar to the sandbox environment. The installed applications should be checked with antispyware and antivirus tools.**

**The basic ideas of role assignments are good because it compartmentalize the access control and create different levels of trust.**

**However, there are some concerns. First, it is too easy to work around the system. End users can choose to ignore roles. Secondly, trickery can be used to compromise security as shown on page 350 in the textbook. Here, a malicious (untrustworthy) application appears to be a legitimate application window asking users to enter personal information such as user name and password. Lastly, Chameleon may be too sophisticated for the average home PC users. It may be unclear who the participants are and how to classify or assign roles to them.**

**There is also a short discussion of Microsoft's view on trusted computing. Instead of the 5 roles as described for Chameleon, Microsoft's view is 2 roles, Red and Green with no communication between the roles.**

**Is Chameleon's metaphor correct? Is it right to mix application based metaphor with file based metaphor with physical based metaphor (home)? Will this create confusion to the end users? What happens when there are multiple desktops or multiple file systems? Note: Microsoft is conducting a study on programmers using multiple monitors and multiple desktops.**

**How should the applications be secured? A useful idea is to have a toolkit with user interface widgets to secure the applications. Note: Toolkits tend to be components whereas privacy and security works in tandem (or the whole environment is involved).**

**Lastly, measurements of user interface improvements should have big impacts. An example is a faster way to reduce the time used for daily computer tasks.**

## Kazaa

**People are accidentally sharing all their files while using Kazaa and other P2P software to download and upload files through the internet.**

**For example, search on Kazaa with the search term inbox.dbx. This will produce results of Microsoft outlook files that are accidentally shared on the P2P networks.**

**How is this possible? From the usability perspective, put yourself in the shoes of the end users to see the problems.**

**List of problems:**

**1. Multiple name of similar things.**
- **My Shared Folder**
- **My Media**
- **My Kazaa**
- **Folder for Kazaa Download**

**2. Downloaded files are also shared files.**

**3. Kazaa recursively share files.**

**4. Can select folder but cannot see inside.**

**5. Multiple user interfaces for doing similar things.**

**In a file sharing study (the focus is on the shared folder), only 2 people got it correct out of 12 people.**

**How can the design be improved? Allow only the sharing of multimedia files. Secondly, provide better feedforward – tell me what is going to happen. Lastly, only allow exception to recursively shared files.**

**eMule is an open source file sharing program that is used with eDonkey and Kad. It is different from Fast Track (Kazaa file sharing). Emule is confined by hops, allows access control of sharing and setting the controls.**

**How to run user studies? A good sample size of 10 to 12 users should be used for user studies.**

**There are difficulties in building a good user interface for privacy and security. They are:**
- **What are the better design methods?**
- **What are the better tools?**
- **What would have helped Chameleon and Kazaa?**

**Cognitive walkthrough can be used flexibly to identify efficiencies and inefficiencies in the designs. For example, use grandma persona and then later use technical persona.**