

Desktop Decomposition, Selective Recomposition, and the Usable Security Problem

Patrick F. Wilbur

Department of Computer Science,
Clarkson University

April 13, 2011

Copyright Notice

Copyright 2011, Patrick F. Wilbur.
Last modified: April 13, 2011 4:11 PM EDT.

Some images have been taken from Public Domain or other copyrighted sources. For more information concerning the licensing of those images, please consult the References list at the end of this document. The author of this work believes that use of those images constitutes Fair Use according to U. S. Copyright Law.

==

LICENSE:

Patrick F. Wilbur
Department of Computer Science
Clarkson University
Potsdam, NY USA

<http://pdub.net>

These slides and content are released under the Creative Commons Attribution-Share Alike 3.0 Unported license, available online at <http://creativecommons.org/licenses/by-sa/3.0/>

You may share (copy, distribute, and transmit) this work, and remix (adapt) this work, as long as you attribute this work to the author and share adapted works under the same or similar license by leaving this entire notice in place (including the original author's name/contact information/URL and this license notice).

Acknowledgments

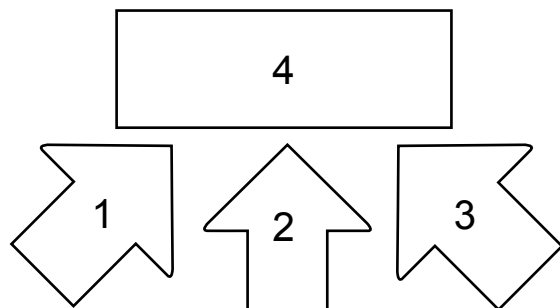
Portions of the research work included in this presentation overlap the doctoral dissertation work of Dr. Todd Dëshane (<http://todddeshane.net>) and the work of advisor, Dr. Jeanna N. Matthews.

Overview

- Introduction: usable security problem
- User intent problem
- Desktop decomposition
- Selective recomposition (recombination)
- Conclusion and future work

Overview

- Introduction: usable security problem (1)
- User intent problem (2)
- Desktop decomposition (3)
- Selective recomposition (recombination) (4)
- Conclusion and future work



Introduction

Why Computer Security is Complicated

- General-purpose *categorically includes* malicious-purpose:
 - **Instant messaging** one minute, **bagel toaster** the next
 - **Instant espionage** one minute, **data toaster** the next
- Even very-specific tools can be misused
(*Mandatory Access Control attempts to address this*)
- Specific tools with limited object access can still be harmful

Usable Security

- Infrequency of possible **false negatives** can be used in a metric to represent a system's level of security

Infrequency of possible **false positives** can be used in a metric to represent a system's level of usability

- We consider a primary goal in study of usable security to be simultaneous reduction of false positives and false negatives

User Intent

- Understanding user intent is helpful, but difficult
- Applications automate, or otherwise provide abstraction for, low-level operations on behalf of users---*Whose intent is it?*
- Difficult questions:
 - Is an action necessary to complete user's desired task?
 - Is a necessary action simultaneously able to be misused?
 - Is the task the user perceives to be desired actually not?

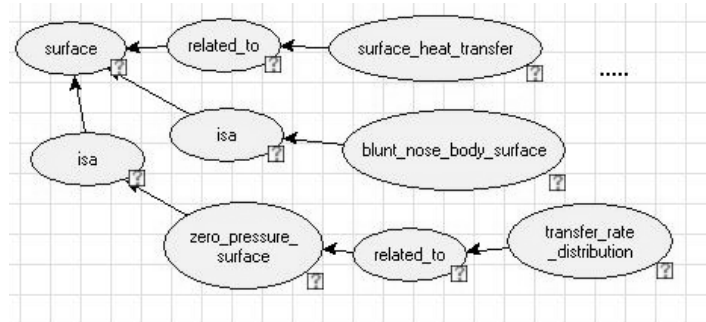
User Intent Problem

Intent Modeling

User intent can be defined as having three parts:

- **User Interests:** What is to be done?
- **User Preferences:** How will it be done?
- **User Context:** Why is it being done?

Note that User I/P/C here can be collectively referred to as a "context", such as from the perspective of factoring where we consider the context of events to include user intent.



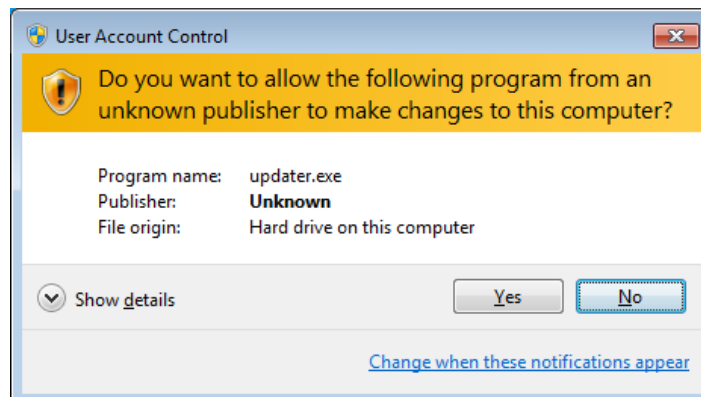
User Context Network for Search Engine Terms^[1]

Users in the Security Equation

- Improvements have been made to User I/P/C modeling
(*Nguyen's dissertation work is interesting*)
- Problem: Determining user intent is easiest through hooks built into an application's user interface, requiring modification
- What is the proper role of the user in the security equation?
- Determining user intent:
 - Actively (*asking the user*)
 - Passively

User Intent Pitfalls to Avoid

- Yes/no questioning
- Less-obvious forms of yes/no (e.g. multiple-choice questions that lead to "yes, you may continue" or "no, you may not continue" effects) (*Sunshine, et. al*)
- Opportunities for and perceived benefits to users gaming the security system
- Users making final security judgments---on average, users have an even more difficult time answering security questions



UAC Dialog in Windows 7^[2]

Are Users Useful for *Anything*?

- Sometimes users are really good at knowing what they think they want
- Other times users are really good at trying everything until they find something that they think is rewarding

User Intent Best-practices

Gaming the users:

- Align mechanisms to determine user intent with the UI elements that users need to use to make progress
- Avoid causing user to perceive a benefit in misleading system; instead, create a progress penalty for misleading

Intent: Capabilities-based Security

In the **object-capability model** of computation, objects interact by sending messages on references, which are obtained by:

- **Initial conditions:** Object A already has reference to Obj. B
- **Parenthood:** parent objects have references to child objects
- **Endowment:** Obj. A grants subset of its references to child
- **Introduction:** Obj. A sends to Obj. B its reference to Obj. C

The capability to perform an operation by an object exists through the sending of messages over a reference

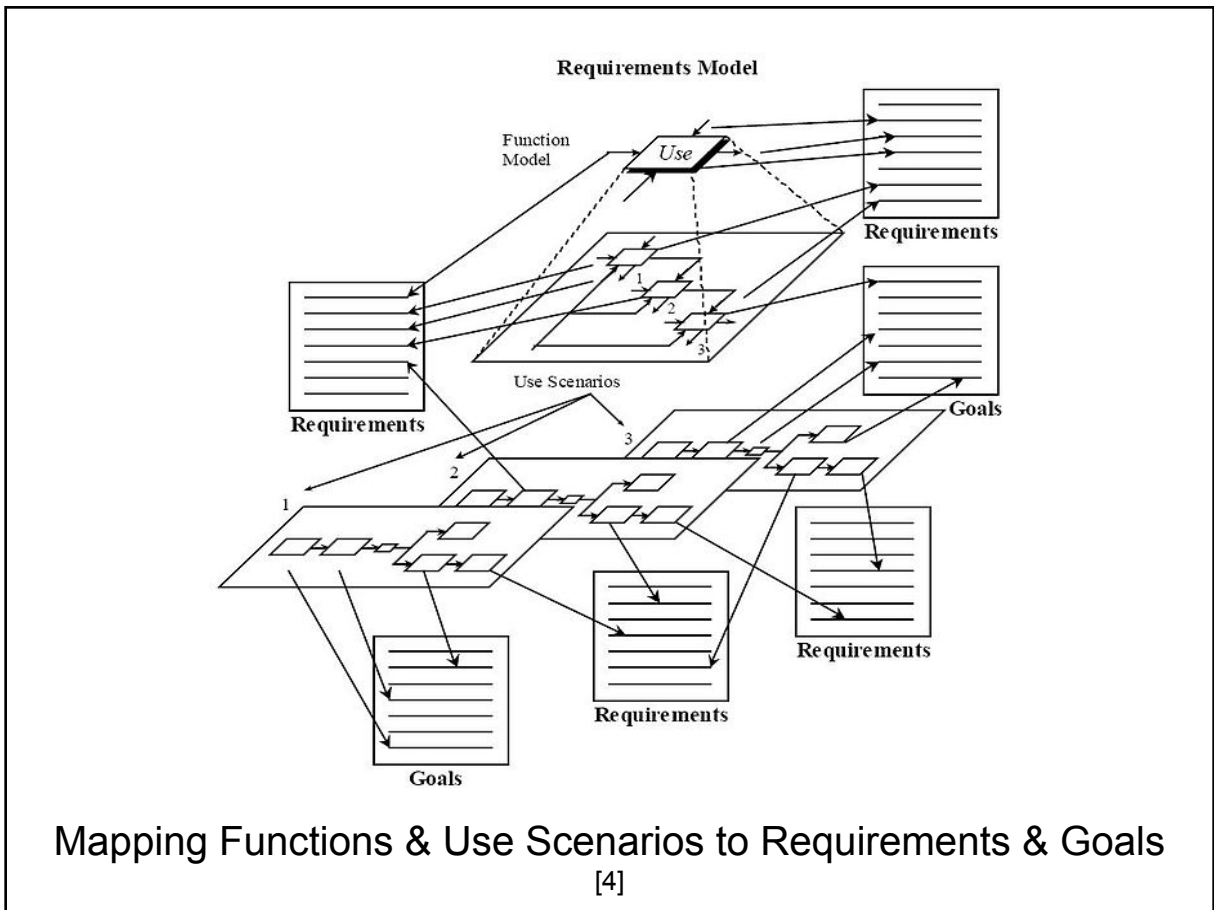
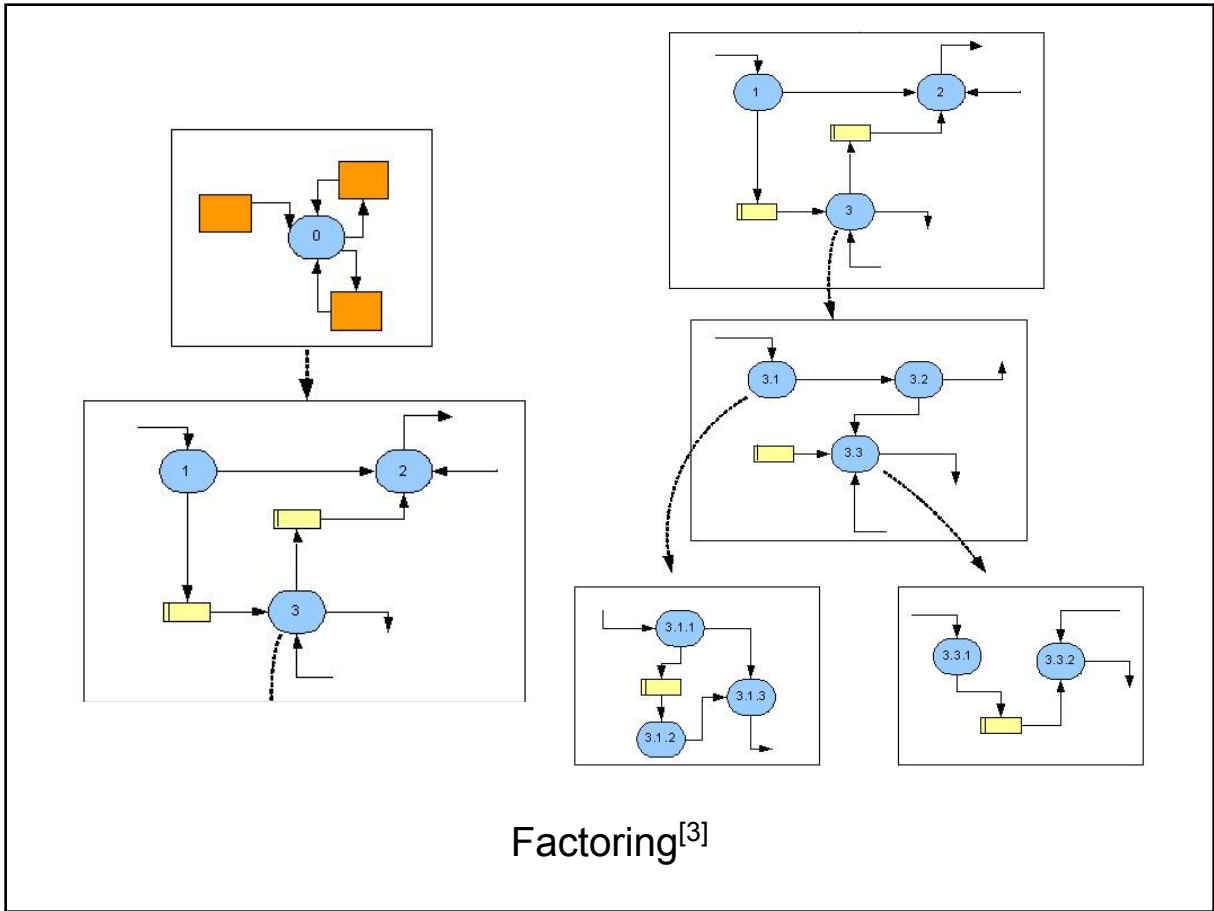
Capabilities-based Problems

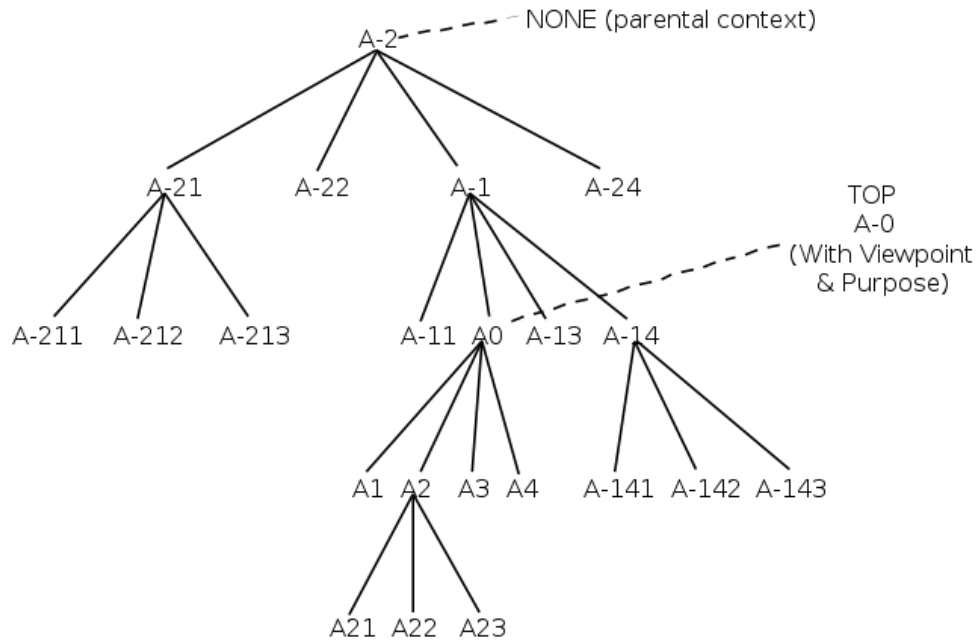
- Implementing such a system requires that applications use a specialized API for launching activities in other applications
- Some implementations do not solve known issues like seen within, for instance, the Android implementation:
 - Yes/no inquiry on app installation, users easily tricked
 - Malicious cooperation between two apps (SD data laundering)
 - Malicious misuse of otherwise reasonable capabilities

Desktop Decomposition

Factoring

- **Factoring** is the process of breaking down a complex system into subsystems
- **Functional factoring** is the process of breaking down composite processes (functions) into smaller component processes
- Such factors are easier to envision, program, and understand
- IDEF family of methods: structured system modeling/analysis

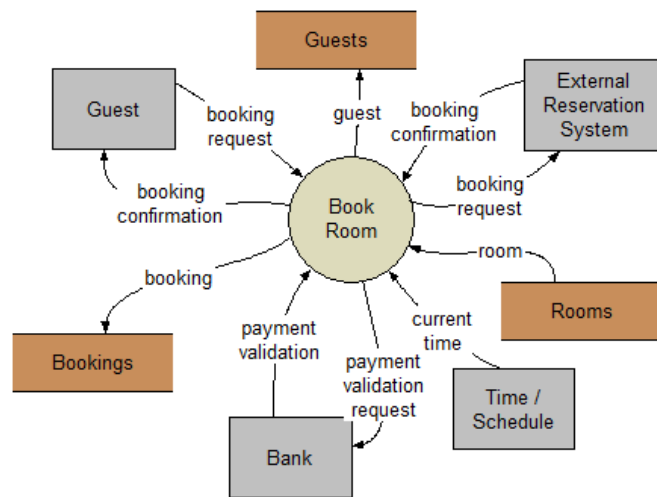




Node-Numbered Context (Positive and Negative Contexts)^[5]

Event Partitioning

- Systems analysis technique to organize requirements for large systems into collection of simpler subsys. (use cases)
- General process:
 1. Identify **actors** (external systems that interact with system)
 2. Identify **triggers** (events actors raise)
 3. Identify **data** and **points in time** that enable event detection
 4. Identify planned system **responses** to events (which complete each of the use cases)



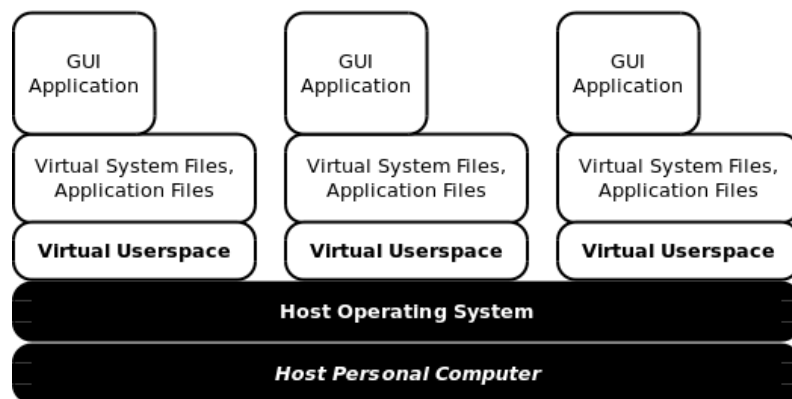
(Actors, Events, Messages, Responses) Form Each Use Case
[6]

Virtual Appliance Model for PC

- Goal is less-general-purpose computing: accomplished through **virtual appliances**
- VM on the Mainframe---predecessor to virtual appliances?
- Virtualization (system-level, OS-level) allows new security models to transitionally support existing applications
- We use a high level of tech. abstraction---combining virtualization technologies leverages compatibility, scalability

Virtual Appliance Model for PC

- Virtual userspaces
- No network, local only, intranet only, restricted domains
- `appify` (Virtual Appliance-on-Demand)



Application Segregation^[7]

Web Browser Apps (Future)

- Chrome-like frontend, where backend processes are placed in sandboxes
- Several options exist for grouping isolation:
 - Domains
 - Trusted websites versus untrusted/unknown websites
 - Internet versus intranet
- Potential for addressing XSS and other web vulnerabilities
- A project worth looking into: Caja (Google)

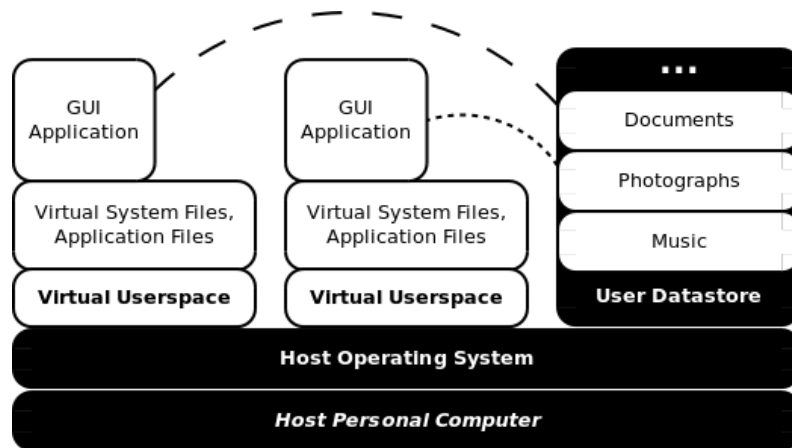
Virtual Appliance Model Benefits

- Specific-purpose appliances easier to understand, maintain, secure, and roll back if necessary
- Possibility of granting partial internet/network access
- Security rules/variables/enforcement not only outside of apps, but also apps' dependencies (OS, library versions)

Selective Recomposition

User Datastore

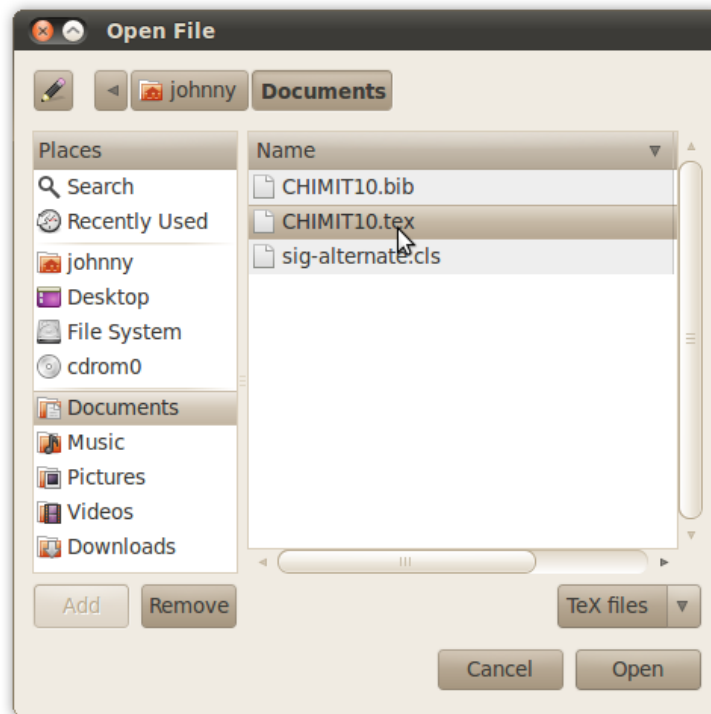
- Manages user data files, version control, libraries of files
- Temporary data, application configuration kept separate
- Copy-on-Write available for special-case uses (*testing apps*)



User Datastore^[7]

Role of User in Selective File Access

- Default access to files of a particular library/directory can be granted to associated applications, when safe to do so:
 - Time of creation
 - Applications without connectivity
- Other data access requires a specific user interaction:
 - Choose file in open dialog
 - Double-click file



Open Dialog^[7]

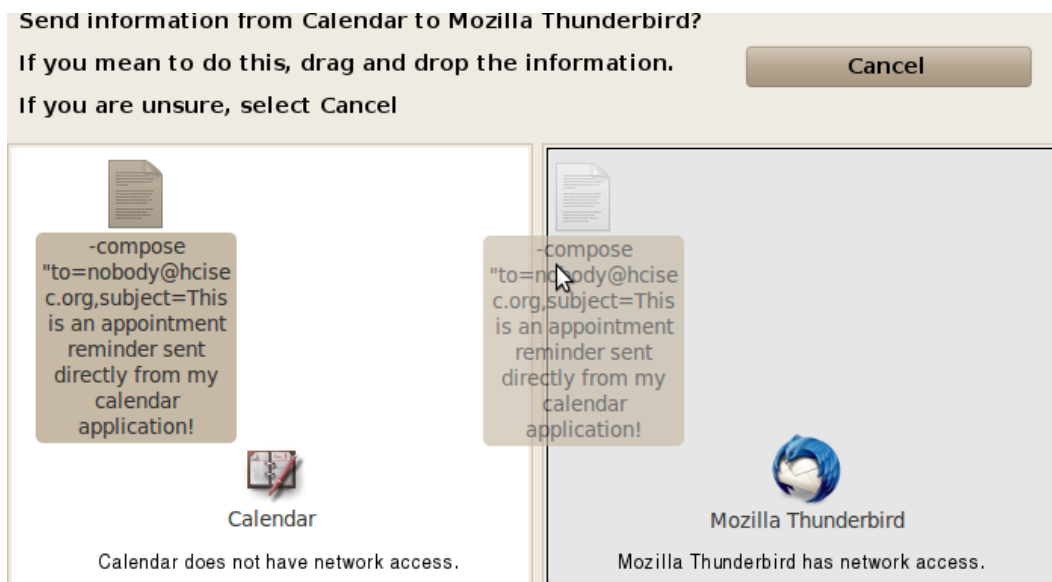
Interprocess Communication (IPC)

- Tightly-integrated application suites with similar capabilities may be sandboxed together
- Intercepting some forms of IPC through "activity proxies" might be a possibility, with thin-OS modification (*intercepting exec/shell calls, when apps launch other apps, for instance*)

Interprocess Communication (IPC)

Open issue:

- How do we selectively allow other forms of IPC across security contexts without yes/no questioning?
- Do other forms of IPC even ever need to be allowed, for typical desktop use?



Dragging-and-dropping Information^[7]

Conclusion and Future Work

Review

- Introduction: usable security problem
- User intent problem
- Desktop decomposition
- Selective recomposition (recombination)
- Conclusion and future work

Contributions

- Virtual Appliance-on-demand
- Further development in study of usable security metrics
- Further understanding of relationship between legacy software versus usable security
- Further understanding of conventional security policy limitations and relationship between policy and users

Future Work

- Investigate differences in usability/security between default ownership of file libraries versus file types by applications
- Expand profile of how real-world apps actually perform interprocess communication, why they do, and where it can simply be blocked
- Make further advancements in interprocess communication problem and the selective recomposition problem, in general (*expose more information, avoid yes/no-like questioning*)

Future Work

- Microsoft Windows virtual appliance support
- Web browser app segregation

References & Additional Resources

- *Dissertation*: Nguyen, Hien. *Capturing User Intent for Information Retrieval*. Doctor of Philosophy Dissertation. University of Connecticut. 2005. URL: <http://www.engr.uconn.edu/~hien/publications/mydissertation.pdf>
- *Paper*: Joshua Sunshine, Serge Egelman, Hazim Almuhammedi, Neha Atri, and Lorrie Faith Cranor. *Crying Wolf: An Empirical Study of SSL Warning Effectiveness*. Proceedings of the 18th conference on USENIX security symposium (SSYM'09). 2009. USENIX Association, Berkeley, CA, USA, 399-416.
- *Paper [7]*: Wilbur, Patrick F. and Todd Deshane. *Johnny Can Drag and Drop: Determining User Intent Through Traditional Interactions to Improve Desktop Security*. CHI/MIT '10: Proceedings of the 4th Symposium on Computer Human Interaction for the Management of Information Technology. November 2010. DOI: 10.1145/1873561.1873565
- *Image [1]*: Figure 2.3.1 (page 42) of Nguyen's dissertation (URL above)
- *Image [2]*: http://en.wikipedia.org/wiki/File:Windows_7_UAC.png
- *Image [3]*: <http://en.wikipedia.org/wiki/File:Fo2ufg823rhf832hfdorfg.JPG>
- *Image [4]*: http://en.wikipedia.org/wiki/File:Functions_and_Use_Scenarios_Mapping_to_Requirements_and_Goals.jpg
- *Image [5]*: http://en.wikipedia.org/wiki/File:21_Negative_Node-Numbered_Context.svg
- *Image [6]*: http://en.wikipedia.org/wiki/File>LastResortHotel_BookRoom_Process.png
- *Article*: <http://en.wikipedia.org/wiki/IDEF6>
- *Article*: [http://en.wikipedia.org/wiki/Decomposition_\(computer_science\)](http://en.wikipedia.org/wiki/Decomposition_(computer_science))
- *Article*: http://en.wikipedia.org/wiki/Event_partitioning
- *Article*: Garfinkel, Simson. *How Android Security Stacks Up*. MIT Technology Review. April 1, 2010. URL: <http://www.technologyreview.com/communications/24944/>