User-Centered Security:
Stepping Up to the Grand Challenge

Lotus software

Mary Ellen Zurko
IBM Software Group, WPLC/Lotus
Security Strategy and Architecture
mzurko@us.ibm.com

Psychological Acceptability

- Saltzer and Schroeder, “The Protection of Information in Computer Systems”, 1975

- “It is essential that the human interface be designed for ease of use, so that users **routinely** and **automatically apply** the protection mechanisms **correctly**. Also, to the extent that the user’s **mental image** of his **protection goals** matches the mechanisms he must use, mistakes will be minimized. If he must translate his image of his protection needs into a radically different specification language, he will make errors.”

User-Centered Security

- Zurko and Simon, “User-Centered Security”, 1992

- “**security** models, mechanisms, systems, and software that have **usability** as a primary motivation or **goal**”
  - Applying human-computer interaction (HCI) design and testing techniques to secure systems
  - Providing security mechanisms and models for human collaboration software
  - Designing security features directly desired by users for their immediate and obvious assurances

Grand Challenges in Information Security & Assurance

- Computing Research Association, 2003
- “Give end-users **security** controls they can **understand** and **privacy** they can **control** for the dynamic, pervasive computing environments of the future.”
- Almost 3 decades after psychological acceptability

Opportunities in User-Centered Security

- There is no such thing as problems, there are only opportunities
  - My boss at Prime Computer, circa 1986

- Human and social relationships to usable security

- Technical challenges best attacked with research

- Further difficulties with implementation and deployment

1. Human and Social Relationship to Security

- What is the best we can hope for when we ask humans to understand a quality of the system so complex that it cannot be understood by any single architect, developer, or administrator?

- Since humans are part of the system and the system’s security, how much responsibility should be assigned to them?

- Since usable security is so obviously a universally desirable attribute, why aren’t we applying resources to it commensurate with its desirability?

I. Understanding vs. Effectively Using Security Controls

- If we go on explaining, we shall cease to understand one another.
  - Talleyrand

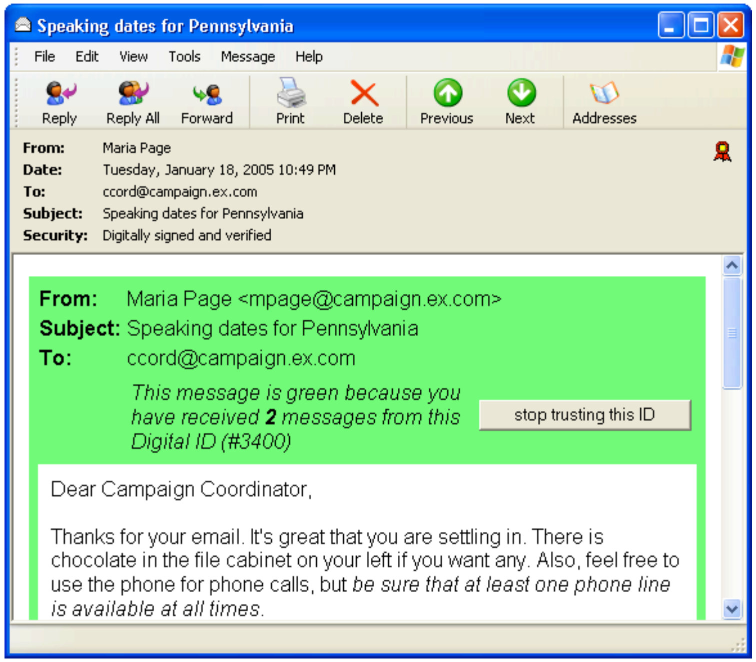
- Authentication and identification
  - Passwords, keys, tokens
- Authorization, access control, roles, digital rights management
- Auditing and logging
- Active content controls (viruses, secure languages)
- Signatures (cryptographic and otherwise)
- Encryption
- Network protection (confidentiality, integrity, replay)
- Sanitization
- Human processor attacks (scam-spam, phishing)
- Assurance
- Ethical hacking

Understandable Security

- From the security professional's point of view
- Verifiability
  - Reference Monitor
  - Security policy
- Transparency
  - Common Criteria and other external evaluation instruments
  - Evaluation by external experts

Understandable Security

- From the user interface point of view
- Graphical (and other) user interfaces
  - Visualizing security
  - Context of the user’s task pertinent to security
  - Security pertinent to the user’s task
- Documentation
  - Explicable
  - To mere mortals
  - In user interface or elsewhere



From: Maria Page
Date: Tuesday, January 18, 2005 10:49 PM
To: ccord@campaign.ex.com
Subject: Speaking dates for Pennsylvania
Security: Digitally signed and verified

From: Maria Page <mpage@campaign.ex.com>
Subject: Speaking dates for Pennsylvania
To: ccord@campaign.ex.com

*This message is green because you have received **2** messages from this Digital ID (#3400)*

Dear Campaign Coordinator,

Thanks for your email. It's great that you are settling in. There is chocolate in the file cabinet on your left if you want any. Also, feel free to use the phone for phone calls, but *be sure that at least one phone line is available at all times.*

Limits to Understandability of Computer Security

- It’s rich
  - By definition, if the system is complex
- It’s complex
  - As currently implemented
- It’s arcane
  - Active content security is the hardest

User Risk Management May Be The Better Way

- Flinn and Stoyles, “Omnivore: Risk Management Through Bidirectional Transparency”
  - What could go wrong?
  - How likely is it, and what damage would it cause to me or to others if it did?
  - How would I know if something went wrong?
  - What reason do I have to believe that it won’t?
  - Who is responsible to ensure that it doesn’t, and what recourse do I have if it does?

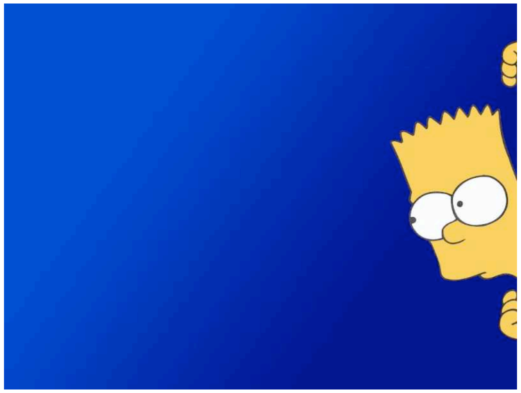
Alternative Grand Challenge

- Give all users (including developers, administrators, and end-users) **security** controls that **protect** them, their systems, and their privacy, that they can **use** appropriately in the dynamic, pervasive computing environments of the present and the future.

- Users must understand the risks, not the security controls

- Users must be able to use the security controls to manage the risks

II. User Slip-ups Are Not “User Errors”

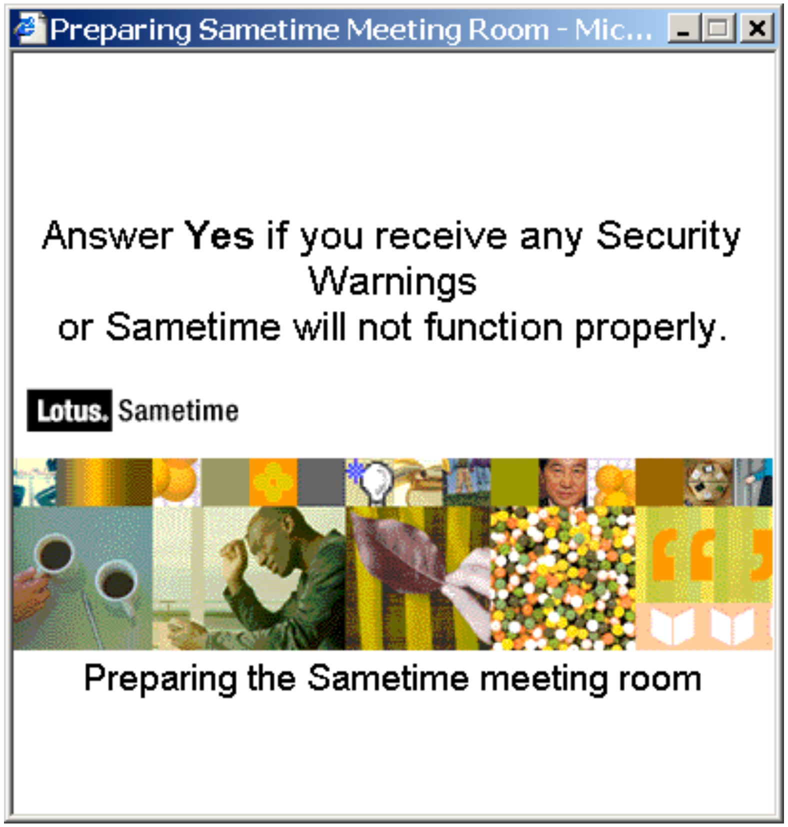


- I didn’t do it.
  - Bart Simpson, cartoon character
- Responsibility for a vulnerability indicates changes necessary to avoid the vulnerability in the future
  - “User error” points to the user or customer
  - Why did the system make the mistake attractive or easy?
  - Slip-up or misunderstanding?
- The answers point back to the software, product, or system as the source of responsibility

Responsibility and Accountability

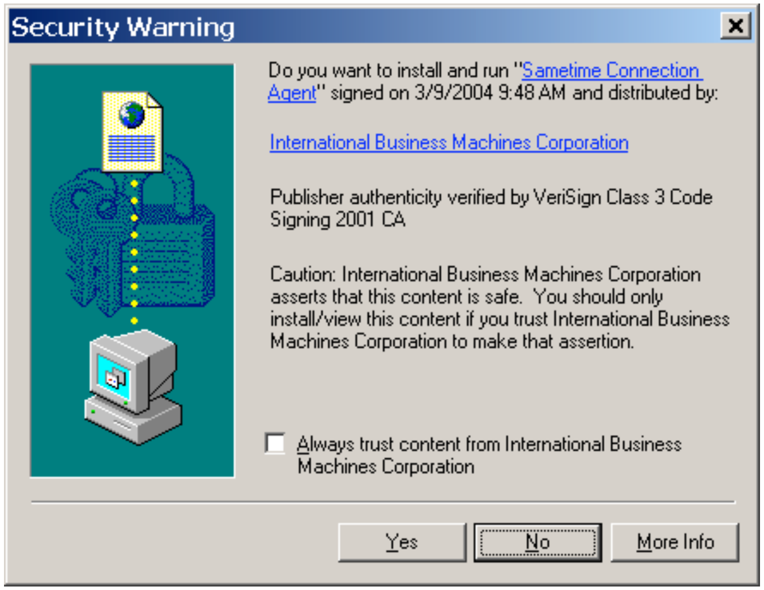
- The security architect and team responsibilities may include
  - Security features
  - Security as a quality
    - Handling vulnerabilities
  - Nothing bad happening
- Not every product has a security architect or designer
  - Every product should have someone who is responsible
  - Usable security unlikely otherwise
- Accountable for security that is deployable and usable
  - Otherwise overall security can be decreased if shifting responsibility is an attractive option

What is the Usable Security model?



Answer **Yes** if you receive any Security Warnings
or Sametime will not function properly.

Preparing the Sametime meeting room



Do you want to install and run "Sametime Connection Agent" signed on 3/9/2004 9:48 AM and distributed by:

International Business Machines Corporation

Publisher authenticity verified by VeriSign Class 3 Code Signing 2001 CA

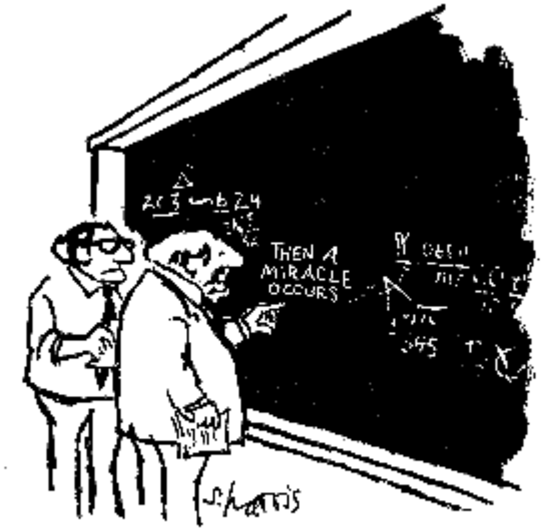
Caution: International Business Machines Corporation asserts that this content is safe. You should only install/view this content if you trust International Business Machines Corporation to make that assertion.

□ Always trust content from International Business Machines Corporation

- Just say “Yes”
- Users will and should work around security to get their job done

How to Check for Usable Security Accountability

- Error states and messages contain actionable advice
  - Bald statements and even explanations of the issue are not actionable
- Documented security processes do not contain unexplained steps (“what” without “how”)
  - “Determine if you trust …”
  - “Verify the key …”
- Unusable security at lower layers will trickle up
  - The buck stops before the user



"I think you should be more explicit here in step two."

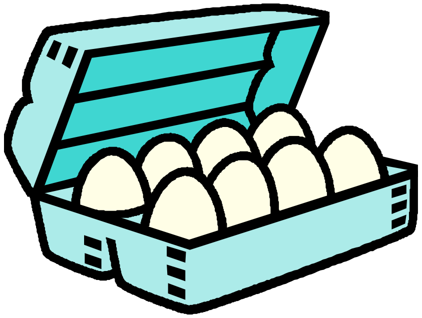
III. Marketing Usable Security

- Sell when you can, you are not for all markets.
  - As You Like It, Act 3, scene v
- Usable security is an obviously desirable attribute
  - Which clearly does not come for free
- How is the cost justified at the market level?
  - Clear need as prudent defense against concrete exploits
  - Strong customer demand
  - Low cost
    - Remains a gap and a challenge

Relationship of Usable Security to Current Exploits

- Reacting to known or theoretical breaches
- Exploits show how usable and useful the security is
  - Drive both design and bug fixing
  - Relationship of usable security features and exploits can be n x m
- Economics of triaging responses to exploits is not always optimal
  - Internal processes determine top vulnerabilities to address based on risk factors and resources available
  - Vulnerabilities made more visible will have increased risk and attention
    - Resources taken from initially more risky vulnerabilities
      - If the organization has a disciplined process
  - Truly ethical hackers need to consider the overall system impact
    - Can only do so if corporate assurance processes are transparent

Increased Customer Demand



- The desirability should be reflected by demand
  - Reactive to existing demand or
  - Proactive creation of explicit demand for an attribute already deemed desirable
- Standard marketing techniques have not been used to develop market pull for security
  - “Brown eggs are local eggs, and local eggs are fresh”
  - “Got milk?”
- Evaluation criteria that customers can use
  - Checklists with straightforward terminology
  - Exposure type categories
  - Features to look for

2. Technical Challenges Best Attacked With Research

- How can we incorporate models of user behavior into models of security, so that real user behavior is taken into account?

- How do we design systems so that security related decisions and actions are minimized, and always made by the person who has the ability to make them?

- How do we design systems so that all the parts that determine the user’s ability to interact with them securely are actually secured?

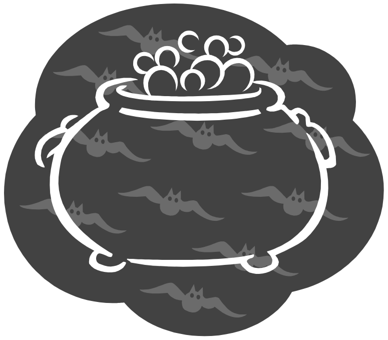
I. Users As Part of the System

- You’re either part of the solution or part of the problem
  - Eldridge Cleaver
- User models and security models are at different levels of abstraction
  - User models of specific capabilities such as memory or slips
  - Targeted user-centered models such as password handling
  - Security models can be driven by user models
    - Trust models, for example
- Users do not interact in isolation
  - Communities and authorities effect their processing
- Risks of unusable security can be integrated into threat based models

II. Who Makes the Security Decisions

- What, me worry?
  - Alfred E. Neuman, Mad Magazine
- Making a security decision correctly is not easy
  - And the easy ones can become nuisances quickly
  - Remember "Just Say Yes"
  - Recovering from not making them is also hard
- Developer to administrator to user
  - Earlier in the lifecycle takes more responsibility with less concrete data
  - Allow overrides later in the lifecycle
  - Large grained decisions means fewer to make
- Personal, fine grained control important in limited circumstances
  - Evaluators, reviewers, thought leaders, geeks
- Constraints make decisions easier
  - Trust examples - naming constraints, physical constraints

III. Assurance For the User

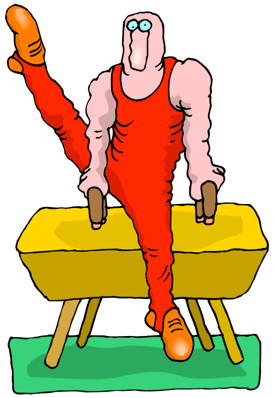


- But yet I'll make assurance double sure
  - Macbeth, Act IV, scene i
- Users make trust and security decisions based on all the information available to them
  - Including how professional the UI design is
- Traditional security assurance is pared down to the smallest possible code scope
  - Encryption alone will not make a system secure
- If we're asking the user to make security decisions, the whole UI is part of the computing base that needs to be trustworthy

3. Further difficulties with implementation and deployment

- How can we integrate the lessons from practice into our research thinking so that we achieve usable security in practice?
- How can we specify and implement reusable security components that support a user-centered security model in the system they're integrated into?

I. Integrating Research and Practice



- In theory, there is no difference between theory and practice. In practice, there is.
  - Yogi Berra

- Security weaknesses of text passwords were revealed by their use

- Usage of security mechanisms changes over time
  - Nostalgia – the days of having just one password

- Mundane development and deployment concerns can impact the feasibility of technology transfer of user-centered security research
  - Many disciplines and features vie for limited design and UI space

II. Components Contributing to Usable Security

- With these kinds of proposals, the devil is in the details
  - John B. Larson
- Reuse is good for security and usability
  - Concentrates security knowledge and functionality
  - Makes security more homogeneous and predictable
- Reuse is bad for usable security
  - Error cases are stripped of their context and relationship to users
- SSL/JSSE in a rich client example
  - User action no longer transparently tied to SSL operation
  - Should I care that the server certificate's validity time period has not begun?
- User or system actions to avoid or recover from security related errors need to be part of reuse contract or interface of the component

Current Progress in User-Centered Security

- Applying Human Computer Interaction techniques to security functionality

- Principles of Usably Secure Systems

- Process advice
  - Think about the user
- Expert application of process or principles
  - We thought about the user
- Authentication and passwords much studied

HCI Techniques for Security

- Expert evaluations
  - Usability expert evaluation of security functionality
  - Strong in visual design
  - Strong in familiar concepts (passwords)
  - No best practices (special process, checklists)
- Testing
  - Usability in the lab
    - Use of security mechanism
    - Simulated attack in some cases
  - Usability in context
    - Interviews, studies, logs
    - Attacks in context a topic for discussion

Principles of Usably Secure Systems

- Psychological acceptability – how?

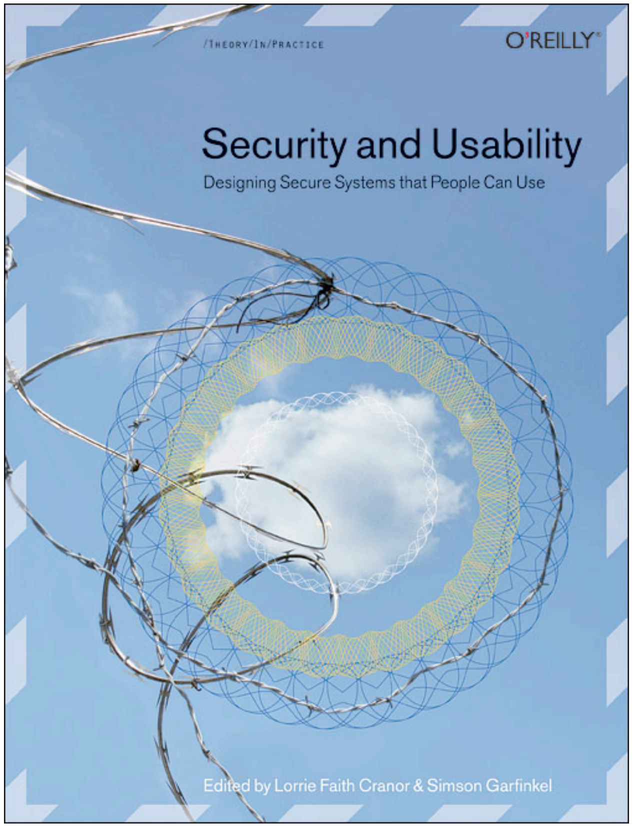
- Safe staging
  - Security decisions do not impede the flow of work
  - Security decisions can be made when the user has data to make them
- Evaluate risks of usability failures
  - Enumerate, then feedback into security model
- Integrate security into user tasks
  - Common tasks are secure by default
- Security transparency within the task
  - Highlight what’s necessary
  - Other security information available as needed
- Reliance on trustworthy authority
  - Singular or distributed

Thank you for your attention, thoughts, and questions

Symposium On Usable Privacy and Security

July 12 – 14 2006, Pittsburgh, PA

http://cups.cs.cmu.edu/soups/2006/cfp.html


/THEORY/IN/PRACTICE
O'REILLY
Security and Usability
Designing Secure Systems that People Can Use
Edited by Lorrie Faith Cranor & Simson Garfinkel

Mary Ellen Zurko
IBM Software Group, WPLC/Lotus
Security Strategy and Architecture
mzurko@us.ibm.com