# *The Argument Against Security Through Obscurity in the Non-Digital World*

Greg Newby
H2K2 Presentation
July 13, 2002

gbnewby@ils.unc.edu
http://ils.unc.edu/gbnewby

# *What are we talking about?*

- Security through obscurity is the bane of most types of computer systems
  - False sense of security
  - Limited verification
  - Vulnerabilities are known by the wrong people
- Security through obscurity means that some sort of secrecy or obfuscation is an important part of the security model. Yet, insiders and intruders will work to discover and exploit the obscure.

# *Examples of Obscurity at Work in the Physical World*

- Locks: Not permitted to own some lockpicking tools or to use them
- Cell phones: Cannot operate scanners in their frequencies
- Stores: Where are the cameras? What are the theft deterrents? Are they known or secret?
- Airports:  WTF are their procedures?
- Access to facilities (i.e., buildings, bases, campuses..)

# *The Fundamental Problem*

- We’re  talking about making systems more secure.  Security involves verification, trust, data integrity, non-repudiation, assurance and other qualities.
  - (think of levels of security in the “Orange Book”)
- Obscure systems are not made more secure by their obscurity, really.  Rather, their flaws are made more difficult to identify

# *The Alternative*

- A major alternative to security through obscurity is the scrutiny of security processes by qualified persons
- Two excellent examples of digital public scrutiny:
  - Open source
  - Crypto
- Scrutiny doesn’t need to be public, but if it is there are more people to perform it

# *Isn't this Risky?*

- People will look for exploits and will likely find them. But many people who find exploits will disclose them for system improvement, which can happen rapidly.
- In an obscure system, people will still identify exploits. But are these people more likely to have ill intent? Possibly, especially if the obscurity is heavy.

# *Security through Scrutiny of Open Source Software*

- Open source software is available to interested persons as source code, for reading and possible modification
- Better security for open source software results when more people are able to seek for vulnerabilities
- The recent Apache security problem is typical: a serious problem is discovered and evaluated by examining the source, and then fixed rapidly
  - (consider that non-open source software has a comparable number of bugs, but without as many opportunities for scrutiny and possible rapid improvement)

# *Security through Scrutiny of Crypto*

- Some of the worst crypto fiascos occurred because companies “rolled their own” crypto (usually with security through obscurity)
  - CSS for DVDs (implementation errors; subject to brute force attack)
- Crypto algorithms and procedures are not believed to be trustworthy until they have been examined and used by many experts

# *Examples of Public Scrutiny for Physical Security*

- Money (US currency), sort of
- (Tune into Robert Steele for disclosure of some federal security procedures, but such disclosure is after the fact. Did their security through obscurity work?)
- Why am I not thinking of many other examples?

# *The Simple Idea*

- Move towards open review of security procedures in the physical world
  - If review is by “qualified persons” only, not all the available brainpower might get applied
  - Open doesn’t imply complete disclosure of all details: perhaps some level of secrecy or obscurity is still desirable
  - Must be combined with an iterative design process to adjust as vulnerabilities are identified

# *General Approach to Open Review*

- Make a vulnerability matrix
- Specify the procedure
- Do a cost/benefit analysis
- Scenario building (think like a “bad guy”)
- Simulations and thought experiments
- Deploy
- Gather performance data and REPEAT

# *Sample Assessment: A Store*

- What are the risks?  Who are the players?
- What happens now?  Where does theft or other undesirable activity occur?
- Brainstorm: new procedures, current procedures
- What are losses we wish to prevent?
- Work through plans, then assess costs and determine whether costs of security plan outweigh costs due to losses
- Deploy, rinse, lather, repeat

# *Sample Assessment: Conference Attendees at H2K2*

- The risks: unauthorized entry is lost revenue, and a possible security risk
- The procedure: wear your badges
- More procedure: different color badges, t-shirts. Radios at key locations, with a clear chain of command
- The costs: Personnel; badges
- Secrecy is a part of this model: The badge design, t-shirt color, radio frequency, etc. were not widely known

# *Summary*

- Physical security is challenging and can be expensive
- Open review of security processes will almost definitely improve security
- Security systems that rely on obscurity are more susceptible to “inside” jobs and the obscurity does not prevent against targeted attack (i.e., the intruder you really want to prevent against!)
- So, let’s try open review of security procedures in the physical world, like we have in the computerized world!

# *Think About It…*

- This is just an idea.  How is it weak?  Do security processes that work well for the computerized world apply to the physical world?
- What are some of the benefits and victories of physical security through obscurity?

Get in touch: gbnewby@ils.unc.edu
http://ils.unc.edu/gbnewby