Security Engineering

Foundations:

Fallacies, 6 C’s, Measuring Security

Meeting 2

Review of Readings

- Chapter 1 in Anderson
- SSD paper “Self-encrypting deception…:
- SSD Infographic from Trusted Computing Group

Rationale

- Security is very different from other engineering objectives and is hard to measure.
- There are many myths and misconceptions about security that must be discussed and clarified
- Security is often at odds with other objectives. However some common trends plague security.

fail
EpicLosers.com

Objectives

- Understand why security is hard
- Learn how to develop threat models
- Learn how various standards and practices help but do not completely solve security problems.
- Suggest how to measure security

Prior Learning

- Let’s Review from Lesson 1: Introduction
    - Course Outline
    - Course Outcomes
    - Defining System Security
    - What is Security Engineering?
    - Security Economics

Discuss

- Why is security difficult to measure? How does it compare to other objectives like cost, power, performance, reliability?
- How do we measure each of these?
- What are the uncertainties in each?

Security "Crash Course"

"Anyone can create an algorithm that he himself can't break. It's not even hard. What is hard is creating an algorithm that no one else can break, even after years of analysis."

Bruce Schneier, Security Pundit

Security “Crash Course” (cont’d)

- Security can be non-intuitive at first:
  - **Mentality:** Bad parties can be skilled, clever, sneaky, and cunning. Not “rational” by most people’s definition. Goal is to cause *intentional* failures.
  - **Imbalance:** Bad parties only need to find *one* way to compromise the security of your system; defender must defend against *all* realistic attack vectors.
  - **Unpredictability:** Bad parties “*win*” by doing what the defenders don’t expect.

Threat Modeling

Security is about threat modeling:

- Who are the potential attackers?
- What are their resources and capabilities?
- What are their motives?
- What assets are you trying to protect?
- What might the attackers try to do to compromise those assets?

Need to answer these questions early, before you can even begin to make any conclusions about a real system

Discuss: Threat Modeling

- Consider a recent attack and identify the threat model
- Who is the threat?
- What are their resources?
- Why are they attacking?
- What assets are they after?
- What vulnerabilities are they exploiting?
- What defenses are already in place? And what new ones can be developed?

Common Fallacies of Security

- **Common Fallacy #1:** ***"A system is either secure or insecure."***

- Security is a gradient
- No such thing as a "perfectly secure system"
  - All systems are vulnerable to attacks
  - We're interested in the level of security that a system provides (recall threat model)

Common Fallacies of Security (cont’d)

- **Common Fallacy #2:** ***“There’s never been an attack in the past, so security is not an issue.”***

- Above reasoning is **intuitive** but also **incorrect**
- Problems with this reasoning:
  - It might have happened, you just don’t know because you haven’t been worrying about it.
  - Technology changes capabilities, incentives, and context so always new things attackers might do.

Common Fallacies of Security (cont’d)

- **Common Fallacy #3:** ***“We use proprietary security algorithms, so the bad guys won’t know these algorithms and our system is secure.”***
- Flaw #1: Bad guys can learn these algorithms
  - Bad guys could reverse engineer algorithms
- Flaw #2: Security through obscurity
  - Proprietary algorithms have a history of being less secure than standardized algorithms
  - If it’s proprietary, how can outsiders (public, FDA, etc.) know for sure?

Common Fallacies of Security (cont’d)

- **Common Fallacy #4:** ***“We’re secure because we use standardized security algorithms like RSA, AES, and SSL.”***

- Using standardized algorithms is good, but **far from** sufficient
- Analogy:
  - Standardized security algorithms are like standardized locks
  - Locks themselves may be strong, but security of building depends on many other things (how you key the locks, how you attach locks to door, how door frame is mounted, whether you also lock the windows, etc.)

Activity: Common Fallacies #1-4

- Consider the limitations of Fallacy number 4.

- How can security standards like AES, RSA, SSL still be broken?
- Even if they are not broken, why are they not sufficient?
- Think of some examples.

Common Fallacies of Security (cont'd)

- **Common Fallacy #5:** ***"We've addressed all known security concerns, so our system is now secure."***

- An example:
  - 2003: Identified security problems with the Diebold voting machine
  - 2004: Diebold introduced defenses to that specific attack; RABA re-evaluated and found that the fix **introduced a new security vulnerability**
  - 2007: Diebold introduced defenses to that new attack; re-evaluation found that the second fix **introduced another new security vulnerability**

Common Fallacies of Security (cont’d)

- **Common Fallacy #6: *“Our system is secure because we’ve had it analyze by third-party testing authorities.”***

- History in other fields says otherwise; consider the case of e-voting:
  - E-voting machines are regularly evaluated by third-party testers
  - But researchers are regularly finding security flaws with these systems

Common Fallacies of Security (cont’d)

- **Common Fallacy #7:** ***“Our system must provide an ‘acceptable level of security’ since we’ve had it analyzed by one or more security experts or teams.”***

- Definitely a good sign, but not sufficient
  - Different security firms have different levels of expertise

- Who defines what an **“acceptable level of security”** is?
  - Does the vendor? Do the security consultants?
  - Each of the above parties have limited vantage points

Common Fallacies of Security (cont’d)

- **Common Fallacy #8:** ***“If we increase security, we’d be forced to decrease safety and/or usability.”***

- Challenging, but not impossible

- To make educated decisions and arguments, we need to:
  - Explore solution space
  - Gauge what’s possible
  - Assess levels of security and usability provided by different solutions

Activity: Common Fallacies #5-8

- Think of 2 examples that trade off Security and Usability.

- Find one example where Security does not penalize Usability.

Common Fallacies of Security (cont’d)

- **Common Fallacy #9:** ***“We don’t need end-devices (like IMDs) to be secure because the back-end system is already secure.”***
- Expression in security community: *“Security is only as strong as the weakest link.”*
- We need to consider security of **all** aspects of the overall system

Common Fallacies of Security (cont’d)

- **Common Fallacy #10:** ***“Only sophisticated adversaries will be able to successfully attack our system.”***

- Expression in security community: *“Attacks only get better, easier to mount over time.”*

- Some adversaries will be sophisticated
  - Different actors: Sophisticated bad guys create tools that less sophisticated bad guys use

Common Fallacies of Security (cont’d)

- **Common Fallacy #11:** ***“Insiders are not going to be adversaries.”***

- Plenty of examples to the contrary (although companies don’t like to talk about it)
  - Spies
  - Greedy employees
  - Disgruntled ex-employees

Discussion: Common Fallacy #11

- Give 2 examples of insider threats.
- For each one, think of attempts to mitigate the insider threat.
- These usually involve human factors.

Common Fallacies of Security (cont’d)

- **Common Fallacy #12:** ***“We’ve thought of everything.”***
- Doesn’t apply to computer security – can never prove to yourself that you’ve thought of everything
- Even this list doesn’t cover every fallacy or example

Security Problems with Security Software

- History is full of products from security companies that have security vulnerabilities

- Conclusions:
  - Security is hard, even for security experts
  - Need for industry-wide oversight
  - Also need many people focusing on this problem

Discuss

- Why does Security have so many fallacies?
- Is there any hope?

- Security jobs will be around for a long time….

Design for Security

- Methodology
  - Identify broad classes of threats
  - Find solutions that cover broad classes.
  - Verify coverage
- What makes a secure system?
  - Modularity, clean boundaries and abstractions
  - Well-specified, open standards
  - Verifiable, verified
  - Minimal, yet still testable, expandable, etc.
  - Careful about re-use, hooks, and other “design productivity” measures…

Enemies of Security: “6 C’s” (cont’d)

- Convenience
  - Weak passwords
  - Unlocked doors, unencrypted data
  - Out-of-date software
- Complexity
  - Hard to validate. Backdoors left open.
- Connectivity
  - More attack vectors. Larger attack “surface”

Enemies of Security: “6 C’s”

- Consolidation
  - All-in-one solutions (Swiss army knives)
- Cost
  - Cost-cutting often compromises security.
  - Naïve security design can increase cost and gain nothing
- Complacency
  - What, me worry?

Discussion: How to Measure Security?

- Theoretical? **Not used in practice (yet)**
  - Computational Complexity   O()
  - Information-theoretic (useful in probabilistic systems)
  - Game-theoretic (models attacker and incentives)
- Review previous attacks and determine coverage of broad classes? **Somewhat effective but what about new attacks?**
- Security economics?   **Maybe**
  - Attempts to quantify costs of attack, defense and assets and provide incentives for security.
  - Game-theoretic formulations a hot topic

Measuring Security with FIPS 140-2 Standards

Level 1

- Very limited requirements
- All components must be "production-grade"
- Egregious kinds of insecurity absent

Level 2

- Requirements for physical tamper-evidence and role-based authentication

Level 3

- Requirements for physical tamper-resistance and identity-based authentication
- Physical/logical separation between interfaces by which "critical security parameters" enter and leave the module

Level 4

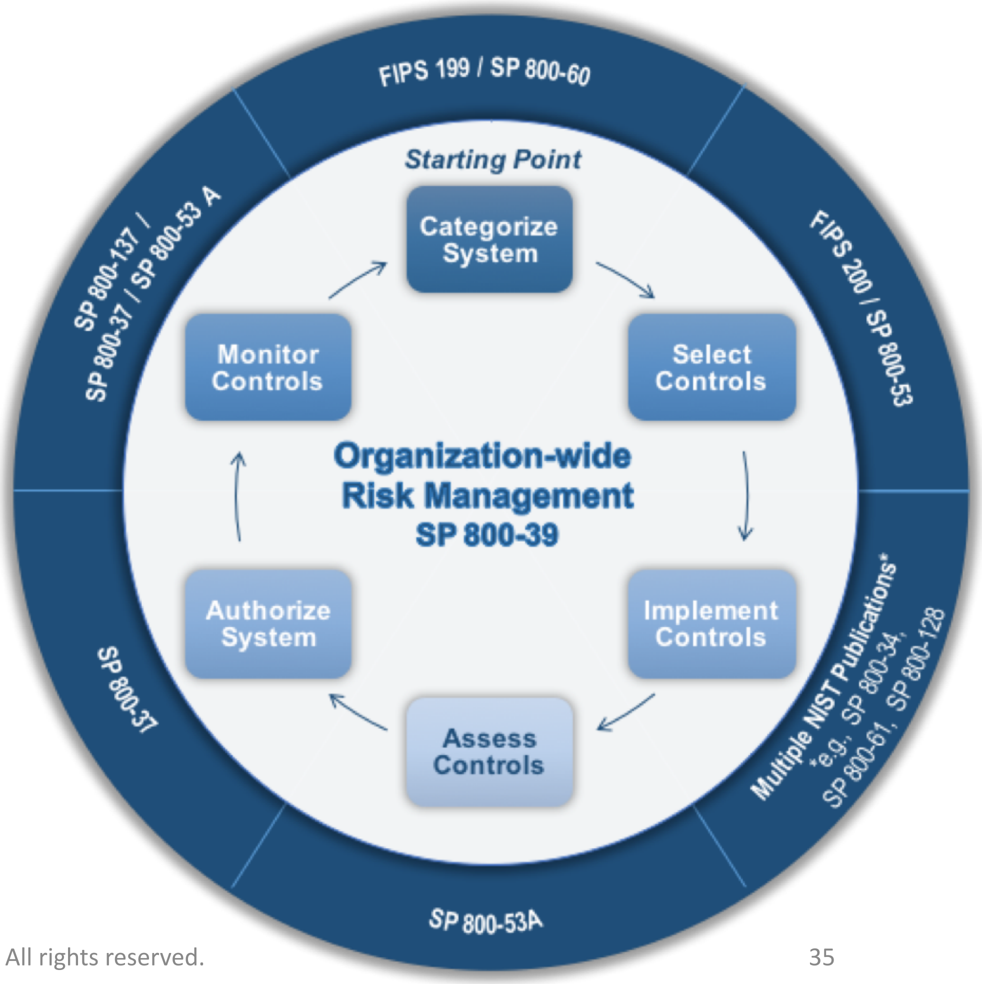
- More stringent physical security requirements
- Requires robustness against environmental attacks

Measuring Security with FIPS 140-2 Standards (cont'd)

- **Example of a** Level 4 certified product:
  - **Hewlett-Packard Enterprises, Atalla Security Products**
  - "The ACS is a multi-chip embedded cryptographic module. It consists of a secure hardware platform (a full length PCI Card) and a secure firmware loader. The purpose of the module is to load application programs, called "personalities," in a secure manner. *"FIPS-approved algorithms:* AES (Cert. #406); RNG (Cert. #200); RSA (Cert. #148); SHS (Cert. #473)

NIST Risk Management Framework

- Go to URL https://csrc.nist.gov/projects/risk-management/risk-management-framework-(RMF)-Overview



Trust Computing Group

- https://trustedcomputinggroup.org/
- https://trustedcomputinggroup.org/wp-content/uploads/Infographic-TCG-SED.pdf

Assignment 1

- Writing! Based on SSD paper
- See Moodle

Next Reading

- Before next class, visit www.autosec.org and read:
- [Comprehensive Experimental Analyses of Automotive Attack Surfaces]()

  Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, Tadayoshi Kohno.
  USENIX Security, August 10–12, 2011.
  (An earlier version of this paper was prepared as a report for the National Academy of Sciences Committee on Electronic Vehicle Controls and Unintended Acceleration, March 3–4, 2011.)

  The full paper is available in PDF form here: PDF. (FAQ)