Secure Software Systems

CYBR 200 | Fall 2018 | University of the Pacific | Jeff Shafer

Beyond the Attack

Content adapted from CS 5430 (System Security), Cornell University, Dr. Michael Clarkson

Schedule

This Week

- Tue August 28
  - Intro lecture
- Thur August 30
  - Project 1 Discussion
  - Security Policy & Aspects

Next Week

- Tue September 4
  - Threats, Harm, and Vulnerabilities
  - Goals and Requirements
- Thur September 6
  - Assurance

Beyond the Attack

https://www.forbes.com/sites/kevinanderton/2017/03/29/8-major-cyber-attacks-of-2016-infographic

8 MAJOR CYBER ATTACKS OF 2016

Hacked: US Department Of Justice

Who did it: Unknown

What was done: Information on

January

Beyond the Attack

Attacks
are perpetrated by
threats
that inflict
harm
by exploiting
vulnerabilities
which are controlled by
countermeasures.

Threats

Def: A *principal* that can cause *harm* to *assets*

- **What kinds of threats can you think of?**
  - One example:
    Hackers driven by technical challenges

Threats

- **Inquisitive people**, unintentional blunders
- **Hackers** driven by technical challenges
- **Disgruntled employees** or customers seeking revenge
- **Criminals** interested in personal financial gain, stealing services, or industrial espionage
- **Organized crime** with the intent of hiding something or financial gain
- **Organized terrorist groups** attempting to influence policy by isolated attacks
- **Foreign espionage agents** seeking to exploit information for economic, political, or military purposes
- **Tactical countermeasures** intended to disrupt specific weapons or command structures
- **Multifaceted tactical information warfare** applied in a broad orchestrated manner to disrupt a major military missions
- **Large organized groups or nation-states** intent on overthrowing a government

Harm

Def: A negative consequence to a system *asset*

- Asset?
  - Information (typically)
  - Hardware and software (potentially)
- Types of harm (the C-I-A aspects)
  - Damage to *confidentiality* (e.g., interception)
  - Damage to *integrity* (e.g., modification, fabrication)
  - Damage to *availability* (e.g., interruption)

Discussion

- **Papapavlo's Bistro and Bar**
  - Let's say they contract with opentable.com to enable online reservations
- **What does the restaurant risk in this relationship?**
  - Confidentiality?
  - Integrity?
  - Availability?

Vulnerabilities

Def: An unintended aspect of a system (design, implementation, or configuration) that can cause the system to do something it shouldn't, or fail to do something it should

- Examples: Buffer overflow, code injection, XSS, misconfiguration, bugs in access control/authentication

Vulnerabilities

- Databases:
  - NVD: https://nvd.nist.gov/
    - *National Vulnerability Database*
  - CVE: https://cve.mitre.org/
    - *Common Vulnerabilities and Exposures*
- Ignoring vulnerabilities is risky
  - *Weakest link* is all a threat needs!
- **Assumptions are also vulnerabilities!**
  - Assumptions about timing/sequence of events, failure modes, message delivery, input format, etc…

Trust vs Trustworthy

- A trusted component is *assumed* to satisfy a security policy
- A trustworthy component additionally is accompanied by evidence that it satisfies the policy
  - Goal of security practitioners: transform *trust* into *trustworthiness*

Discussion

- **Papapavlo's Bistro and Bar**
  - Let's say they contract with opentable.com to enable online reservations
- **What vulnerabilities might threats exploit to cause harm to Papapavlo's?**

Countermeasures

Def: A defense that protects against attacks by neutralizing either the threat or vulnerability involved

- Strategy:
  - **Prevent**: block attack or close vulnerability
  - **Deter**: make attack harder but not impossible
  - **Deflect**: make other targets more attractive
  - **Mitigate**: make harm less severe
  - **Detect**: as it happens or after the fact
  - **Recover**: undo harm

Types of Countermeasures

- **Physical**: something tangible (walls, locks, guards)
- **Procedural**: protocols for how people act (laws, regulations, policies, contracts)
- **Technical**: hardware and software (cryptography, access control, passwords, intrusion detection systems, ...)

Technical Countermeasures

- **Isolation**: restrict communication between components (virtual machines, sandboxes, processes, firewalls)
- **Monitoring**: a program analyzes execution and blocks bad things from happening (reference monitor, intrusion detection system)
- **Recovery**: detect and reverse effects of harm (transactions, backups, key changes)

Discussion

- **Papapavlo's Bistro and Bar**
  - Let's say they contract with opentable.com to enable online reservations
- **What countermeasures might be employed to mitigate these vulnerabilities?**

Beyond the Attack

Attacks
are perpetrated by
threats
that inflict
harm
by exploiting
vulnerabilities
which are controlled by
countermeasures.

Principles

Security Approaches

- **Prevention**: build systems that are completely free of vulnerabilities
- **Risk management**: invest wisely in countermeasures
- **Deterrence through accountability**: attribute attacks to humans and legally prosecute

Principles of Prevention

- Accountability
- Complete Mediation
- Least Privilege
- Failsafe Defaults
- Separation of Privilege
- Defense in Depth
- Economy of Mechanism
- Open Design
- Psychological Acceptability

Accountability

Accountability

Hold principals responsible for their actions

- **Authorization:** mechanisms that govern whether actions are permitted
  - *Vault locks keep out most principals*
- **Authentication:** mechanisms that bind principals to actions
  - *Vault key enables specific principals*
- **Audit:** mechanisms that record and review actions
  - *Vault security cameras monitor all principals*

Complete Mediation

Complete Mediation

Every operation requested by a principal must be intercepted and determined to be acceptable according to the security policy

- Component that does the interception and determination is the **reference monitor**
- Related to Accountability
- Restricts caching of information, including previous decisions

Least Privilege

Principals should be given the minimum privileges necessary to accomplish their task

- Limits the damage that can result from accident or malice
- aka “need to know”

Failsafe Default

Failsafe Defaults

Base decisions on the *presence of privilege*, not the *absence of prohibition*

- The default answer is "**no**"
  - Say "yes" only when there is an explicit reason to do so
- Principals who discover they don't have access will complain
- Attackers who discover they do have access won't complain!

Separation of Privilege

Separation of Privilege

Different operations should require different privileges

- Supports Least Privilege
- In tension with usability: too many operations and objects and principals

Defense in Depth

Defense in Depth

Prefer a set of complementary mechanisms over a single mechanism

- *Complementary:*
  - **Independent:** attack that compromises one mechanism is unlikely to compromise others
  - **Overlapping:** attacks must compromise multiple mechanisms to succeed

Economy of Mechanism

Economy of Mechanism

Prefer mechanisms that are simpler and smaller

- Easier to **understand**, construct, analyze
  - And thus less likely to have unknown vulnerabilities
- Applies to any aspect of system, not just security
- **Trusted computing base** (TCB): mechanisms that implement the core security functionality
  - ...keep the TCB small

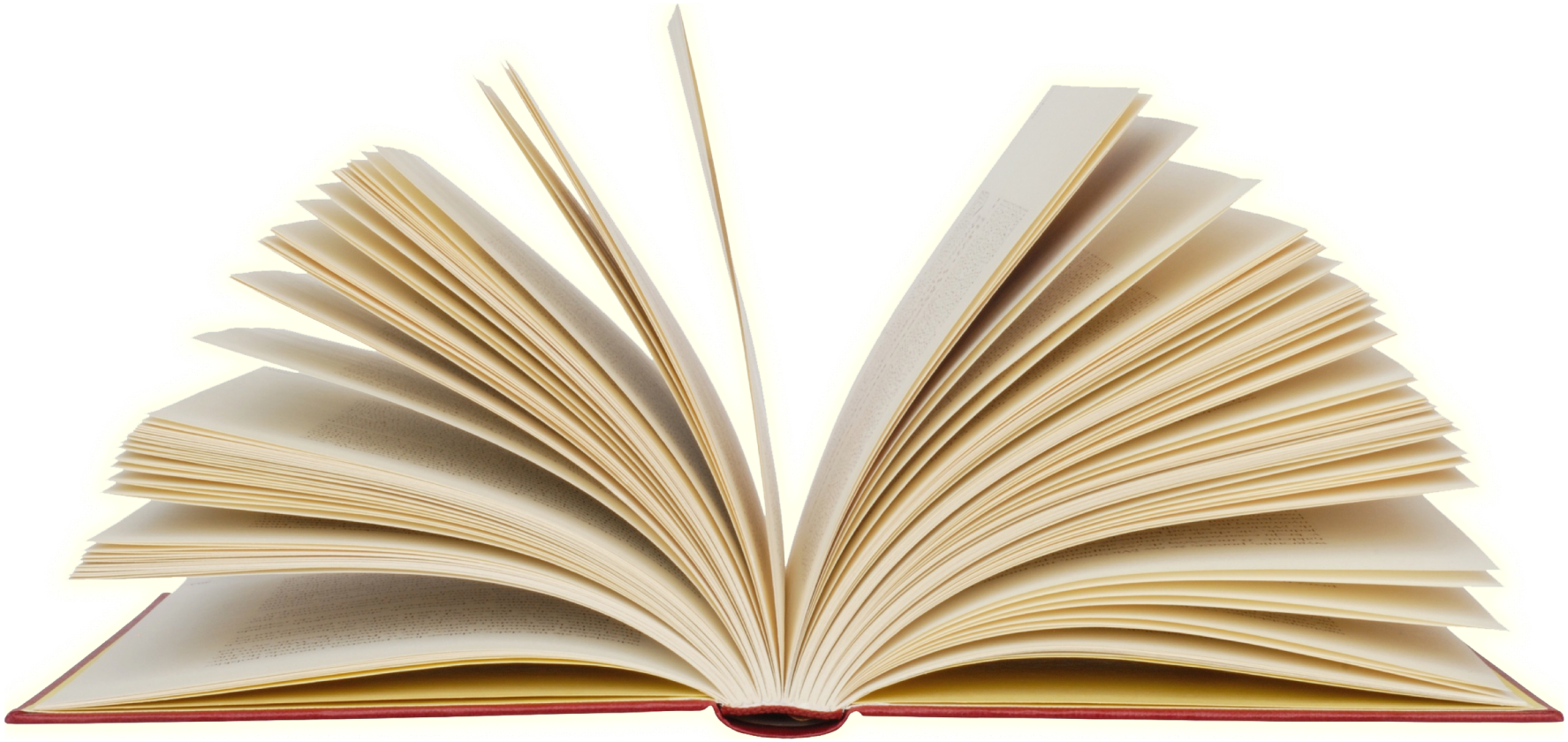
Economy of Mechanism

- **seL4 microkernel**
  https://sel4.systems/

"The world's first operating-system kernel with an end-to-end proof of implementation correctness and security enforcement..."

A microkernel is small – not a full operating system!

Open Design



Open Design

Security shouldn't depend upon the secrecy of design or implementation


```
/*      efdtt.c      Author:  Charles M. Hannum <root@ihack.net>              */
/*                                                                            */
/*      Thanks to Phil Carmody <fatphil@asdf.org> for additional tweaks.      */
/*                                                                            */
/*      Length:  434 bytes (excluding unnecessary newlines)                   */
/*                                                                            */
/*      Usage is:  cat title-key scrambled.vob | efdtt >clear.vob             */
#define m(i)(x[i]^s[i+84])<<
unsigned char
x[5],y,s[2048];main(n){for(read(0,x,5);read(0,s,n=2048);write(1,s,n))if(s[y=s[13]
%8+20]/16%4==1){int i=m(1)17^256+m(0)8,k=m(2)0,j=m(4)17^m(3)9^k*2-
k%8^8,a=0,c=26;for(s[y]-=16;--
c;j*=2)a=a*2^i&1,i=i/2^j&1<<24;for(j=127;++j<n;c=c>y)c+=y=i^i/8^i>>4^i>>12,i=i>>8
^y<<17,a^=a>>14,y=a^a*8^a<<6,a=a>>8^y<<9,k=s[j],k="7Wo~'G_\216"[k&7]+2^"cr3sfw6v;
*k+>/n."[k>>4]*2^k*257/8,s[j]=k^(k&k*2&34)*6^c+~y;}}
```

https://www.cs.cmu.edu/~dst/DeCSS/Gallery/hannum-efdtt-source.txt

Open Design

Security shouldn't depend upon the secrecy of design or implementation

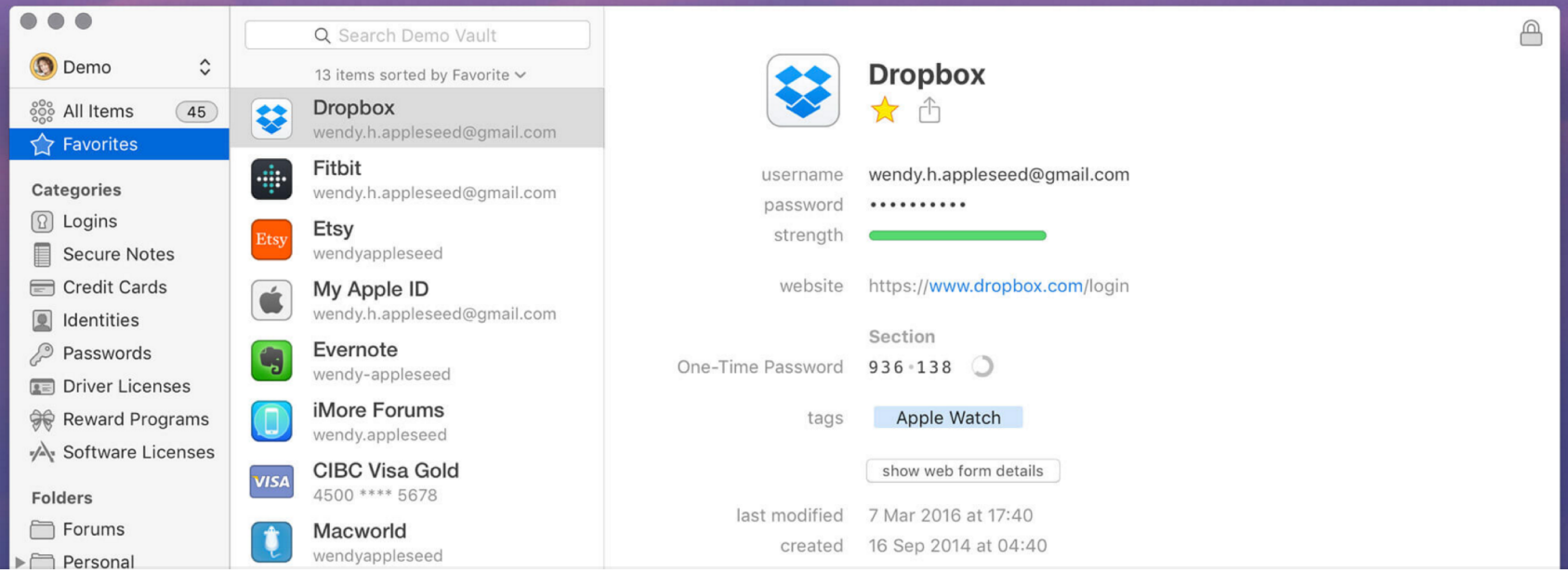
- Arguments **for** open design:
  - Secrets eventually come out: reverse engineering is possible, employees move around
  - Making details public increases chance of identifying and repairing vulnerabilities

Open Design

Security shouldn't depend upon the secrecy of design or implementation

- Arguments **against** open design:
  - Secrecy supports Defense in Depth by making it harder to find vulnerabilities
  - Lack of hard evidence that Linus' Law really holds ("given enough eyeballs, all bugs are shallow")
  - After identification, some vulnerabilities cannot quickly or easily be repaired

Psychological Acceptability



Psychological Acceptability

Minimize the burden of security mechanisms on humans

- Don't make operations (much) more difficult to complete than if security mechanisms were absent
- Don't make configuration difficult
- Produce comprehensible error messages
- ***Always a tradeoff between security and usability***