

# CS1632, LECTURE 25: Security Testing

Bill Laboon



# Writing Secure Software Is Difficult; So Is Testing It!

- **Heartbleed:** ~ 66% of servers connected to the Internet vulnerable; allowed for basically untraceable eavesdropping on data in memory
- **Shellshock:** A defect in bash (default shell for OS X and most Linux distributions) which allowed arbitrary code execution. Discovered in 2014; vulnerability was introduced in 1989.
- **June 2016:** Sixteen vulnerabilities found in Windows 7 font display subsystem!



# Why Is It Difficult?

1. Adversaries are actively seeking to defeat security
2. Information about security vulnerabilities modifies behavior of adversaries
3. You need to protect all doors; they only need to find one they can open
4. Even minor vulnerabilities can have truly catastrophic consequences



# Pittsburgh - A Great City To Learn About Security!

- Many security researchers here at Pitt and CMU
  - LERSAIS in the iSchool - Laboratory for Education & Research on Security-Assured Information System
  - CyLab at CMU
  - Dr. Lee, Dr. Farnan, Dr. Garrison here in the CS department
- Software Engineering Institute
- CERT
- Many security engineering positions (esp. at banks)



# History

- Security was not a big deal in the early computing world
- Usually required physical access to a system to do anything
- Few people had necessary skills even if they did (“security through obscurity”)



But there were networked systems in the 60s and 70s...





# Phone Phreaking





# The 80s: Security Goes Mainstream





# People Were Concerned This Would Happen





# Breaking Into Computers Went Mainstream



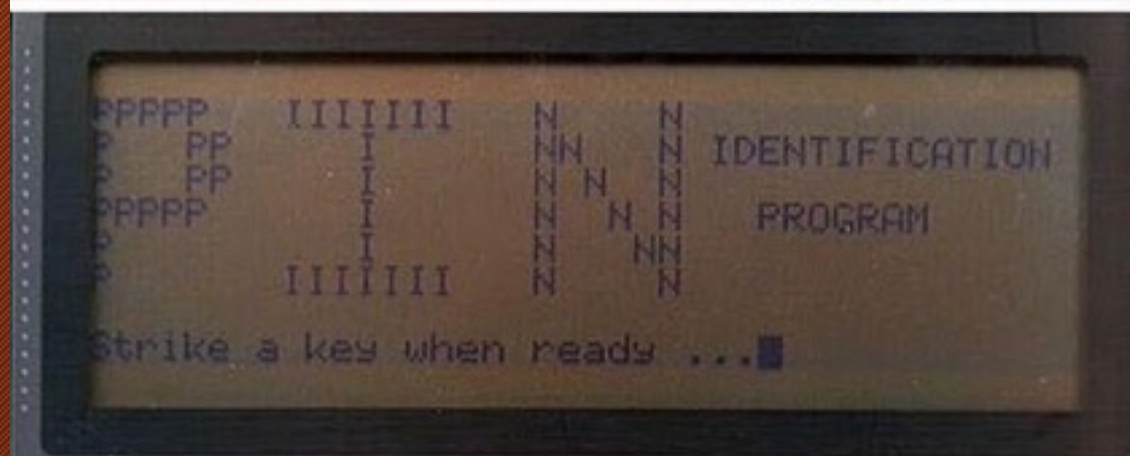


# 1988 - The Year It All Changed





Sadly, skills used less and less often for curiosity... exploiting vulnerabilities is Big Business





# The InfoSec (CIA) Triad

- A secure system needs to provide three qualities:
  - *Confidentiality*
  - *Integrity*
  - *Availability*



# Confidentiality

*No unauthorized users may read data.*



# Integrity

*No unauthorized users may write data.*



# Availability

*System is available for authorized parties to read from and write to.*



# Terminology: Kinds of Security Attacks

1. *Interruption* (attack on availability, e.g. pulling plug from network switch, DDoS)
2. *Interception* (attack on confidentiality; e.g. eavesdropping, keylogger)
3. *Modification* (attack on integrity; modifying or deleting data)
4. *Fabrication* (attack on integrity; making up or inserting data)



# Terminology: Passive vs Active Attacks

- *Passive: Do not modify system in any way*
  - Eavesdropping
  - Monitoring
  - Traffic Analysis
- *Active: Modify the system in some way*
  - Log in as a different user
  - Fill up database with garbage data
  - Modify bank account information



# Terminology: Vulnerability vs Exploit

- Vulnerability: identified weakness of a system
- Exploit: (aka "sploit") Technique or mechanism used to compromise a system using a vulnerability



# Terminology: Kinds of Malicious Code

- **Malware** - General term for malicious code (includes all kinds below)
- **Bacteria** - program that consumes system resources (e.g. fork bomb)
- **Logic bomb** - code within a program which executes an unauthorized function
- **Trapdoor** - secret undocumented access to a system or app
- **Trojan horse** - program that pretends to be another program
- **Virus** - replicates itself WITH human intervention
- **Worm** - replicates itself WITHOUT human intervention
- **Zombie** - A computer or program being run by an unauthorized controller
- **Bot network** - collection of zombies controlled by master
- **Spyware** - surreptitiously monitors your actions
- **Adware** - Shows you more ads
- **DOS (Denial of service) attacks** (e.g. via LOIC)



# Protections

- Firewalls
- Operating System Permissions
- CDNs
- Well-written code
- Proper security measures
- Cryptography
- User training

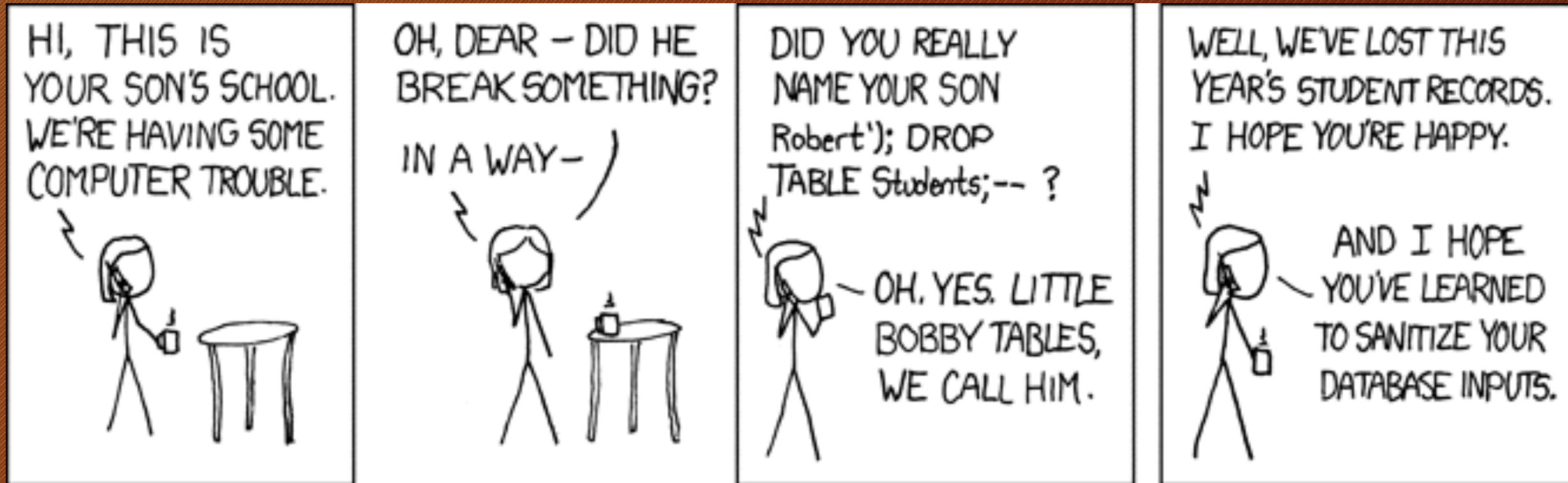


# Common Attacks

- Injection Attacks
- Broken Authentication
- Cross-Site Scripting (XSS)
- Insecure Object References
- Security Misconfiguration
- Insecure Storage
- Buffer overruns
- Social Engineering



# Injection Attacks





# Broken Authentication

- One user pretends to be another
- How?
  - Guess or crack passwords
  - “Password reset”
  - Unencrypted session IDs
- Apple iCloud leak was suspected of being this
- Sarah Palin email hack was definitely this
  - All he needed to know, he learned from Wikipedia
  - Answered security questions, reset password



# Cross-Site Scripting

- Get a third party to execute code on their system
- Similar to an injection attack, but with an intermediary
- `<html>I love Nickelback! They're so dreamy!<script>eval("evil code!!!!")</script></html>`



# Insecure Object References

- Someone can access something by knowing where it is, despite not having proper security credentials
  - <http://bank.com/?account=9844>
  - <http://bank.com/?account=9845>



# Security Misconfiguration

- You have proper security, it's just not set up correctly!
- Default passwords
- IPS, packet filtering, etc. not running
- Insecure machine on secure network



# Insecure Storage

- Secure data is stored in an unsafe way
- Example: credit card numbers being stored in a /tmp or logging directory as part of logging all transactions



# Buffer Overrun

- Trying to read or write more data than a buffer supposedly has access to - reading or writing past the end of a buffer
- This is what heartbleed was - see `heartbleed.c` in `sample_code` directory



# Social Engineering

