# IT 4500 : Information Security

## Day 1

### Dr Joe Francom

### Spring 2014

---

### Intro

Syllabus Website Canvas LabSim

---

### Activity 1

Use the following sites (or others) to find 2-3 security breaches (5min):

- http://datalossdb.org/
- google.com
- http://www.us-cert.gov/
- http://arstechnica.com/security/

Take a few notes on the breaches you find. Can you determine what happened? Why it happened? What the organization did to remedy it? How much it cost them to fix it? Was it preventable? Be prepared to share your findings.

# IT 4500 : Information Security

## Day 2

### Dr Joe Francom

### Spring 2015

---

### Review

- Summarize what you learned from the s1.0 and s1.1 lectures

---

### Activity 1

Skim through this article and see if you can answer the following questions: (Look in the results and analysis section)

- Where do most security incidents occur?
- By whom are they perpetrated? What countries?
- What are the different categories that these incidents are frequently grouped into?
- Any other things that jump out to you or you find interesting.

Be prepared to present your findings.

---

### Activity 2

Let us do some simple reconnaissance. Find some different web tools that will gather information for you about the `cit.cs.dixie.edu` website. We are not trying to hack anything at the moment, just gathering information. Answer the following:

- what information did you find?
- What could you do with this information?
- Is there any way to prevent this information from getting out?
- Can you figure out what version of Apache is running? How about what ports are open? What OS version is it running?

(Hint: start by doing a Google search for `online reconnaissance tools`)

---

### Activity 3

Find 3 current vulnerabilities for a program of your choice(wordpress has some if you can't think of other programs):

- http://osvdb.org/
- http://www.us-cert.gov/ncas/current-activity/
- http://secunia.com/community/advisories/search/

Answer the following:

- Why do vulnerabilities exist?
- What information is provided about the vulnerability?
- How is it suggested to be fixed?
- Other relevant info?

---

## Conclusion

Exim example

# IT 4500 : Information Security

## Day 3


### Dr Joe Francom


### Spring 2014

---

### Review

- Summarize what you learned from the s2.1-2.3 lectures
- What is out-of-band authentication?

---

### Activity 1

http://arstechnica.com/security/2014/10/google-offers-usb-security-key-to-make-bad-passwords-moot/

---

### Activity 2

http://confidenttechnologies.com/demos/secure-second-factor-demo https://www.keylemon.com/ (facial recognition)

---

### Activity 3

- http://www.youtube.com/watch?v=lSLxbobQ6Fg
- http://www.youtube.com/watch?v=H3TheqOaXas#t=167
- http://www.keyboard-biometrics.com/online-demo.html
- http://www.govivace.com/
- http://www.biochec.com/

# IT 4500 : Information Security

## Day 4

### Dr Joe Francom

### Spring 2014

---

### Review

- Summarize what you learned from the section 2 lectures

---

### Activity 1

What is freeRadius?

---

### Activity 2

Locate the logs on your computer that show successful and failed authentication. On a Windows system, they can be found by right-clicking My Computer and selecting Manage. Then, in the computer Management dialog box, expand the Event Viewer in the left column. Expand Windows Logs and select Security. On a Ubuntu Linux system (and many other Linux distributions), the authentication logs can be found in /var/log/auth.log.

Locate any failed logon attempts.

Document how you would configure the maximum size of a log file and what the system will do when that maximum size is reached.

---

### Activity 3

- An example of Identity Managment from IBM
- Experimenting with Identity management (if you want to play with it, can't all do as a class or we overwhelm servers)
    - http://www.microsoft.com/en-us/server-cloud/products/forefront-identity-manager/try.aspx
    - I was successful using a windows IE and linux ff
    - what is provisioning?

---

### Activity 4

First, research what apparmor is:

- http://askubuntu.com/questions/236381/what-is-apparmor
- http://ubuntuforums.org/showthread.php?t=1008906
- https://help.ubuntu.com/community/AppArmor

Questions:

How does this relate to our current topics?

---

### Activity 5

- Try apparmor
  - Here
- Using one time passwords
  - https://www.digitalocean.com/community/tutorials/install-and-use-otpw

---

## Activity 5

More Linux user security:

- chage
- ulimit (cant really test this, but see man page for what it does)
- `/etc/security`

---

## Activity 6

Encryption?? Labs and help?

# IT 4500 : Day 5

## Overview

### Dr Joe Francom

---

# Review

What parts of CIA triangle does encryption help with? Why are public algorithms more secure than secret ones?

---

# Activity 1

### Steganalysis & steganography

- Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity
- Steganalysis is the art and science of detecting messages hidden using steganography;

---

# Steganalysis & steganography

How does it work? Remember that each pixel can be represented with a combination of 3 bytes (rgb). We can take a few of those bytes out and put our text in there with only marginal deterioration in image quality.

Example: Download and install steghide

```
apt-get install steghide #not easy to do on kali
```

- Grab an image (supports jpg, perhaps others)
- May be interesting to record what the image size is before and after.

```
steghide embed -cf picture.jpg -ef secret.txt
steghide extract -sf picture.jpg
steghide info received_file.wav
```

---

# Steganalysis & steganography

So, what might this be used for?

Steganography is NOT the same as cryptography. Cryptography scrambles the message so that it cannot be viewed, stego hides the existence of data. Can hide in file header fields, between sections of metadata. Can use images, sound, movies...

---

# Activity 2

Attacking passwords

- Brute force attack (cryptographic)
- Dictionary Attack

---

# Activity 3

Man in the Middle attacks

- MITM
- replay attack

---

# Activity 4

Hashing

- md5
- sha

# IT 4500 : Day 6

## Overview

**Dr Joe Francom**

**Spring 2015**

---

# Review

What things stood out to you in section 4?

---

# Activity 0

Man-in-the-middle attackage using ettercap gui.

---

# Activity 1

Social engineering toolkit - spoofed email - bad site

---

# Activity 2

Phishing emails list

---

# Activity 3

Forensic example http://cit.dixie.edu/it/4500/projects/lab4.php

---

# IT 4500 : Day 7

## Overview

### Dr Joe Francom

### Spring 2014

---

# Review

What did you learn in sections 6.1-6.3?

---

# Activity 1

- Spoofing packets with scapy

---

# Activity 2

- Armitage

---

# Activity 3

- OpenVAS or other scanner

---

# Activity 5

DOS Attacks

- http://www.youtube.com/watch?v=R8k-cJnrIrc
- http://www.cvedetails.com/version/142323/Apache-Http-Server-2.2.22.html
- msfconsole
- use auxiliary/dos/http/apache_range_dos
- set RHOSTS 144.38.x.y
- exploit

# IT 4500 : Day 8

## Overview

**Dr Joe Francom**

**Spring 2015**

---

# Review

What else did you find interesting from Section 6?

---

# Activity 1

In groups of 2 or 3, find examples of the following that you can demonstrate to the class. You will have 15-20 minutes to find, install, figure out how they work then you can show us what you learned:

- Firewalls (linux or windows)
- Proxy servers
- HTTP content filters (windows site restrictions)
- VPN solutions (how about logmein or other related?)
- Virus blockers
- Antiphishing software
- Related technologies (K9 browser, opendns)

---

# IT 4500 : Day 9

## Overview

### Dr Joe Francom

### Spring 2014

---

## Review

Summarize section 7

---

## Activity 1

http://cit.dixie.edu/it/4500/sources/rootkits.php

---

## Activity 2

Intrusion detection and prevention

Look at snort. (network-based)

Divide into groups again, research and install an IPS/or IDS (host-based)

---

## Activity 3

---

## Activity 4

# IT 4500 : Day 10

### Dr Joe Francom

### Spring 2014

---

## Review

- Malware
- What is the difference between a virus and a worm?
- How are trojans and botnets related?
- What does it mean for software to be quarantined?
- Why should you show file extensions?

---

## Review 2

- Did you learn anything new about password attacks?
- How to mitigate rainbow table attacks?
- How do you reduce the attack surface of a device to harden it?
- Hotfix vs patch?

---

## Activity 1

- Malwarebytes
- Some sample files? here
- Windows defender?
- Is malwarebytes an anti-virus?
- Find/research/install an antivirus. Be prepared to discuss your results

---

## Activity 2

- Honeypots

---

## Activity 3

- Botnet research
  - find some examples of different botnets (Zeus is one to look for to get started)
  - How easy/hard is it to install? (YOu need not install it, just look for instructions)

---

## Activity 4

# IT 4500 : Information Security

## Secure Programming

### Dr Joe Francom

---

# Buffer Overflow

### What it is?

A buffer overflow occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold. Since buffers are created to contain a finite amount of data, the extra information - which has to go somewhere - can overflow into adjacent buffers, corrupting or overwriting the valid data held in them. - http://searchsecurity.techtarget.com/definition/buffer-overflow

Video

---

# More

The attacker will need to be able to identify a buffer overflow vulnerability in some program and understand how that buffer will be stored in the process memory, and know what other adjacent memory items he could write to.

- Could DOS the system

To do this, could use a technique called fuzzing.

---

# Fuzzing

Fuzzing involves sending malformed strings into application (and web) input and watching for unexpected crashes. There are lots of interesting tutorials as to how to do this. Many times after finding this malformed input, you would then use assembly to figure out how to deliver shellcode (or how to deliver the exploit).

Fuzzing does have practical usage in software development, but it is also a tool used by hackers to find vulnerabilities in applications.

Many fuzzing tools: ComRaider (activeX), see fuzztools on security distro.

---

# Fuzzing

-Video

- http://blog.chromium.org/2012/04/fuzzing-for-security.html

---

# Why are programs vulnerable to buffer overflows?

- poor programming
- computer doesn't care... up to programmer to make sure memory is contained
- low level languages

---

# How to protect against buffer overflow

- compile time defenses

- choose a higher level language (downsides?)
- graceful failures
- Address space layout randomization (by OS)
- NX bit

---

# Examples

- Vulernability Database
- This is interesting
- bufbomb example = make program do something it wasn't originally designed to do
- Exim example = access to shellcode

---

# Secure Programming

- SDLC (esp maintenance and testing)
- Least privilege principle
- Never trust user input (input validation)
- Reduce attack surface area (remove unused functions)
- Fail securely (try/catch)
- Patch and fix issues correctly

---

# Software security

Software security is closely related to software quality and reliability.

Defensive programming: intended to ensure the continuing function of a piece of software in spite of unforseeable usage of said software. Sometimes this is called "Secure Programming". Never make any assumptions about code

---

# What to do?

Most programmers focus on solving a problem, (the algorithm involved) rather than considering every point of failure. Often make assumptions about the input and the environment it executes in. Defensive programmers should:

- validate all input
- handle potential failures

Software security has to be implemented in design phase of the development rather than as an afterthought.

---

# Handling program input

One of the most common failures in software security.

The input is not always explicitly known. Making assumptions about the input can be bad. You should always handle the input cautiously to make sure that it isn't invalid.

Two key areas of concern for input are:

- size of input
- meaning and interpretation of input

Don't make any assumptions, validate everything.

---

# Inputs

Do the inputs conform to what you are expecting? ie. If you are expecting a filename, does the input follow the typical filename pattern?

Injection attack: Essentially this is where an attacker provides non-expected input in order to do something that they shouldn't. It influences the control or execution of a program. Commonly seen in PHP or other webscripts

Expected input = expected results, but what if the attacker injects a command, i.e. `xxx; echo success; ls -l finger *`