Reminder:

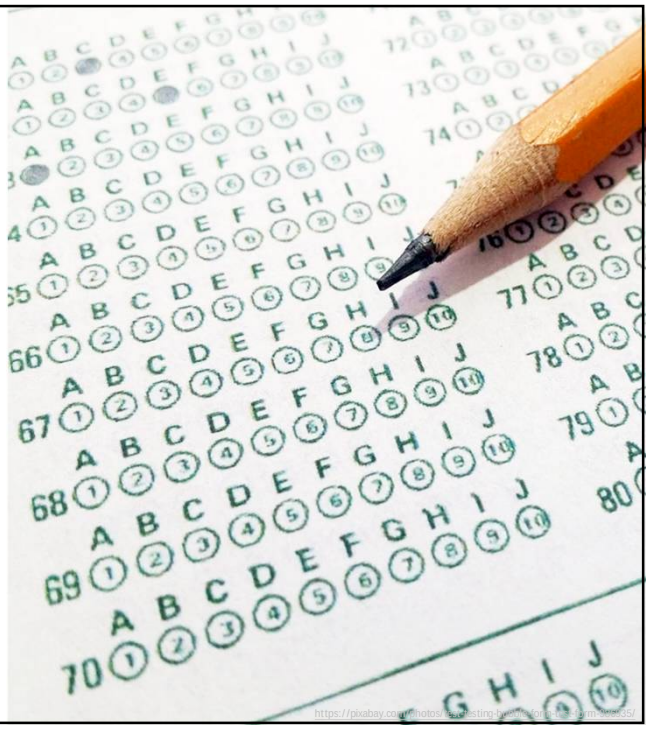
Please complete all "student evaluations of instruction."

There are several such evaluations being sent out from different offices of Dordt University.

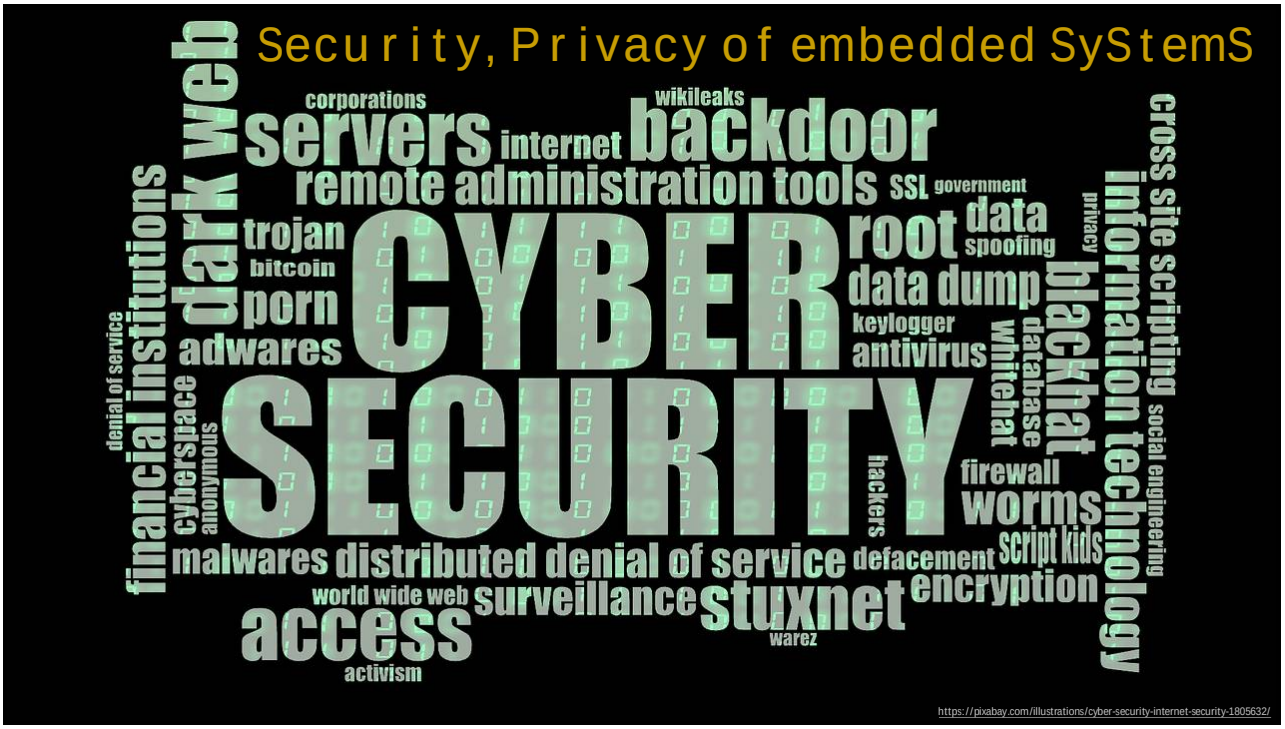
I hear that some professors are requesting students to inform them when the evaluations have been completed. I prefer not to ask that of you.

That said, in a class as small as ours 100% participation from all students in all requested evaluations is important. Even one student not participating will seriously reduce the significance of the results.

Please participate in all student evaluations of instruction.



1



2

A history lesson:

"Target to pay $18.5 Million to 47 States in Security Breach Settlement"

—New York Times headline, 5/23, 2017.
Event happened in November-December 2013

40 million credit card numbers stolen
70 million records of personal information stolen

Up-to-date and sophisticated malware detection software was in use.

"It is just an HVAC monitor. All it does is report energy use. So what if attackers can log onto it?"
(Others think the initial attack was via their accounts payable system.)

The problem: A part of exposed to the Internet for innocuousTarget's LAN was reasons. The device had no capability to deal with customer's data. The device was loaded with malware. *Then it had capabilities nobody except the attackers imagined.*

3

A history lesson:

"Hacker Can Send Fatal Dose to Hospital Drug Pumps"

—Wired Magazine headline, 6/08, 2015.
~400,000 pumps involved

Infusion pump is equipped with radio-connected smart interface for remote monitoring and control, say from the nurses' station.

Remote access needed authentication, then one can monitor and adjust dose rate. This part works properly.

The pump had a feature to allow a firmware update..
That feature needed no authentication.

Manufacturer denied there was a problem
Claimed "air gap." I.E. must plug into a hidden RS-232 port.
The hidden RS-232 port turns out not to be the only way to update the firmware. The manufacturer perhaps did not understand the full capabilities of the programmable hardware in the pump.

https://commons.wikimedia.org/wiki/File:Infusionspumpe.JPG

4

A history lesson:

"The Mirai botnet. . . Almost takes down the internet"

—CSO online magazine headline, 3/09, 2019.
Event happened October 12, 2016

Security cameras, DVD players, other such IoT devices with Internet connections having open telnet ports are "updated" with malware. Together they form a botnet capable of coordinated action.

Result: A massive DDoS attack on banks.
So much traffic that the Internet on the East Coast of the U.S. became impractically slow.

After analysis of the event the risk of open Telnet ports on low-power processors is realized.

https://www.amazon.com/Security-Wandwoo-Outdoor-Detection-Surveillance/dp/B07MF37BL8

5

Ch 17, Lee and SeShia
Security, Privacy of embedded SyStemS

Four important aspects of security and privacy

Confidentiality—secrets are kept from those who should not know them

Integrity—data cannot be tampered with

Authenticity—Identities of various actors in the system is trustworthy

Availability—The system is reliable, cannot be broken or overloaded.

Foundational to achieving the above one needs to realize. . .

A plan for what should be secret and what may be public knowledge
(Think deliberately about where the divide is—the divide must exist but where is it?)

There should never be an expectation that "the system is secure."
The system can only be made secure against anticipated threats.
A "threat model" that includes a conception of "malicious intent" is necessary.

6

Cryptography—the basic way to protect secret information

Vocabulary words:

"Text:" A message in any form. An MS-word file, a hand-written note, a Vinyl phonograph record, a video stream. . .
Within cryptography these are all called "texts."

Encryption: The process of translating a text into a secret form. (*Crypt*—Latin for secret)

Plaintext: The text that is to be protected. The text in its original form.

Ciphertext: The text in its encrypted form. The text in its encrypted form.

Decryption: The process of translating a text from its secret form to its original form.

Key: **A secret that is shared** (to some extent) to enable coordinated encryption and decryption of a text.
In a technical sense there is no essential difference between the key and the plaintext
or between the key and the ciphertext.
They are both "texts" of some sort that go into a mathematical relationship to produce a result.
In a practical sense, we interpret the key as a secret that can be practically shared.
One should assume that **the secrecy of the key is all that will protect the ciphertext**. (Kerckhoff's principle)
Specifically, one should assume, "the enemy knows the system." (Shannon's maxim)
Above, these are the ideals. Real life is a little different, but ignoring the ideals is a bad mistake.

"Security through obscurity" is not reliable. (here is something to it—but highly problematic, not ideal.
"Moving target defense" is security through obscurity via rapid replacement of systems.

7

Symmetric-Key Cryptography—same key is used for encryption and for decryption

The quintessential example of symmetric-key cryptography is the **one-time pad**. (one-time key)

Example:
Plaintext: "We shall defend our island, whatever the cost may be, we shall fight on the beaches, we shall fight on the landing grounds, we shall fight in the fields and in the streets, we shall fight in the hills; we shall never surrender." (Winston Churchill)

Step 0: Sometime far in advance of the need to communicate, give the recipient of the message the key. Use a secure and authenticated channel for this as maintaining the secrecy here is essential. The key must be at least as long as the message. The key should be random, having no recognizable pattern. In this example, the key is:
"ldjklwjeoigflkljhfosoifelkjekjfoifgpokfdlmfanlknioiuuoijlhhmnzjhouigayifwejflqwkmglsdkjfpodiovewknqiugnklfbbnp
anlkjbvmnkjllmfkrohiuguihdnvcmvnkjlvsdokbgmngskljnfldvmmmzlkvnjsanbjmadjiuovfupoijkqmnb1nythlhjlfmsfnkb
zuchkqqfbmgfmslkjgjklasdkmgsfmsfldkl"

Step 1: remove all punctuation, spaces, capitalization.
"weshalldefendourislandwhateverthecostmaybeweshallfightonthebeachesweshallfightonthelandinggroundswe
shallfightinthefieldsandinthestreetsweshallfightinthehillsweshallneversurrender"

Step 2: Assign an integer to each letter of the alphabet. a = 0, b = 1, c = 2, . . . z = 25
For each letter of the message use modulo 26 addition to add the message letter to the corresponding key letter.
w + l = h, e + d = h, s + j = b, etc.

8

Symmetric-Key Cryptography—same key is used for encryption and for decryption

The quintessential example of symmetric-key cryptography is the **one-time pad**. (one-time key)

Example so far:

Key is
"ldjklwjeoigflkljhfosoifelkjekjfoifgpokfdlmfanlknioiuuoijlhhmnzjhouigayifwejflqwkmglsdkjfpodiovewknqiugnklfbbnp
anlkjbvmnkjllmfkrohiuguihdnvcmvnkjlvsdokbgmngskljnfldvmmmzlkvnjsanbjmadjiuovfupoijkqmnb1nythlhjlfmsfnkb
zuchkqqfbmgfmslkjgjklasdkmgsfmsfldkl"

Plaintext is:
"We shall defend our island, whatever the cost may be, we shall fight on the beaches, we shall fight on the landing grounds, we shall fight in the fields and in the streets, we shall fight in the hills; we shall never surrender."

The result of step 2 is the cyphertext. In this case we get:
"hhbrlhuhsnksoyfapxzsblblldnzoayvmhuhhwfbmqbefskyttqabhwweolnrzlosmeksfiqhjrlsjkxfnpddxmncujzcprzcju
abgyvqnhigxngsoojzxqcjyousdyszblkybzzrnjmgyprrcllbdiktvrdchnfmlogzqhdccpeawnqfa"

The ciphertext may now be sent to the recipient via an insecure channel.

Step 3, To decrypt, subtract the key from the cyphertext using modulo 26 arithmentic.
h – l = w, h – d = e, b – j = s, etc.

weshalldefendourislandwhateverthecostmaybeweshallfightonthebeachesweshallfightonthelandinggroundswe
shallfightinthefieldsandinthestreetsweshallfightinthehillsweshallneversurrender

Apply a grammar checker to insert spaces, add punctuation, capitalization.

9

Symmetric-Key Cryptography—same key is used for encryption and for decryption

The quintessential example of symmetric-key cryptography is the **one-time pad**. (one-time key)

Example so far:

Step 4: Throw away the key
It is essential that the key is never reused in any form.
Reverse the key and use it again? No.
Interchange every pair of letters of the key? No.
Add "c" to every letter of the key to make a new key? No.
To protect the original message **no other cyphertext must be made from any form of the key**.

Given steps 0 and 4 are done correctly the one-time-pad type of encryption is unbreakable.

It is the only known encryption that is theoretically unbreakable!

Do it yourself:
http://rumkin.com/tools/cipher/otp.php

10

Symmetric-Key Cryptography—same key is used for encryption and for decryption

Most World-War II era cryptography systems were variations of the one-time-pad.

Usually these involved making up ways to re-use the keys so that not so much key information had to be secretly transferred in advance.

The vulnerability of re-using the key:

Suppose the plaintext message is somehow obtained, say by spying.
Score one point for the enemy—but so far, the cryptosystem is secure
Only one message has been compromised and that has nothing to do with the cryptosystem.

But now, suppose that the plaintext message can be matched up with its cyphertext.
The difference is the key. (cyphertext – plaintext modulo 26 = key) The plaintext gives away the key.
Since the key is being re-used by the encoder, the cryptosystem is cracked.
Had the key been discarded after one use the key would be worthless.
(The enemy already had the plaintext. Discovery of the key gave away no new information. = worthless)

But it is not even necessary to match a plaintext to a ciphertext. If one has many ciphertexts, all created from the same key but conveying different plaintexts, it is easily possible using statistical methods to recover the key. It is remarkably easy.

11

Symmetric-Key Cryptography—same key is used for encryption and for decryption

The present accepted best practice in symmetric-key cryptography is the "**Advanced Encryption System**" (**AES**)
NIST has oversight of AES

It uses a 128 or 196 or 256-bit-long key.

It works on text blocks of 128 bits each.
The blocks are encoded one-time-pad style, then permuted based on the key, then re-encoded, etc. several times.
The result could be brut-force attacked because the key is reused, but there are so many combinations of keys possible and the decryption effort is sufficiently high such that there is no practical way to harness enough computer time before, "the sun burns out."

Maybe this will change some day and a brute-force attack on AES will become public knowledge.
That is what has happened to each of AES' predecessors!
If you do not use a one-time-pad you are engaged in a cat-and-mouse game of sorts.

12

Symmetric-Key Cryptography—same key is used for encryption and for decryption

The matter of keys and authentication (knowing who sent the message and that the message is not corrupted).

An advantage of all symmetric key cryptosystems is that (under some conditions) they are self authenticating. Suppose a third party intercepts an encrypted text and attempts to modify the text or entirely replace the text. Not knowing the key, this action will corrupt the cyphertext and the corrupted cyphertext will decrypt to nonsense. If a message decrypts in a sensible way—say it passes a CRC check—the recipient can be sure of its authenticity and the source of it.

However the need for prior exchange of the key is a serious limitation. Once the need to secretly communicate arises, it is too late to share the key—a secure channels does not exist, or you would be better off to use it instead of using a cryptosystem in an insecure channel.

If one needs to communicate secretly in a situation where there is no secure channel for key exchange, then one must turn to a. . .

Public-Key Cryptography system.

Interestingly, a public-key system exchanges ease of authentication for the opportunity to publicly exchange keys. (Or you can give up some ease of encryption and use the system for authentication instead!)

13