Cryptography

By Trevor & Ian

quick announcement

Offensive Security Group (Wednesday, facebook link) has received permission to try to hack CS125's infrastructure.

If you're interested, talk to Erik Beitel/Beütel for a debriefing.

Please do not mess with it until you learn the constraints. This risks disciplinary action, expulsion, or a felony.

What is a crypto?

- It's a bit coin

actually tho

- theoretical - secure computation
- practical - secure communication

- authenticity - did this message come from the right person
- integrity - do messages arrive untampered
- availability - does your communication still work in presence of adversary

crypto words

- public/private key systems
- hash functions
- stream ciphers / block ciphers
- password hashing
- elliptic curves
- s/mime (email)
- WPA/WEP (KRACK from last year)
- DES / AES / RSA
- kerberos
- IPSec
- DNSSec
- x509
- openssl
- zero knowledge proofs
- searchable encryption
- homomorphic encryption
- multi-party computation
- quantum cryptography
- steganography

xor

- Binary **operator** (like plus, minus
  - 1 true, but not both
- Reversible

  Message ⊕ key = encrypted

  Encrypted ⊕ key = message

- Fundamental in cryptography
  - Used in DES, AES, etc.

XOR		Input #1	
		0	1
Input #2	0	0	1
	1	1	0

PGP - pretty good privacy

- a system for encrypting messages
- you can use it to encrypt, sign email
- used extensively in debian's package manager

- GPG (crap usability)

```
uid ian klatzco (hack the planet) <pgp@klatz.co>
sig  sig3  2CB7D015 2017-08-31 __________ 2017-12-
sig  sig3  2CB7D015 2017-12-30 __________ 2018-12-

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: SKS 1.1.6
```

diffie-hellman key exchange

- how to establish secure communication over an insecure channel

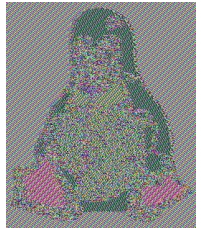
math-y explanation:
https://security.stackexchange.com/questions/45963/diffie-hellman-key-exchange-in-plain-english

https://www.youtube.com/watch?v=U62S8SchxX4 video explanation

AES / RSA

- most well-known symmetric / asymmetric crypto schemes
- factoring primes
- RSA: old, slow, unbroken, solid, modular arithmetic
- AES: has known weaknesses

- both relatively simple to understand, highly recommend the wikipedia articles



$$c \equiv m^e \pmod{n}$$ Encryption

$$c^d \equiv (m^e)^d \equiv m \pmod{n}$$ Decryption

N = (p*q) where p,q are large prime #s'

e is coprime to $\lambda(n) = \text{lcm}(p,q)$

D is multiplicative modular inverse of e (d is private key)

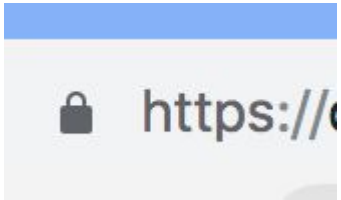
Hard to calculate d because you need to factor n, which should be > 2048 bits

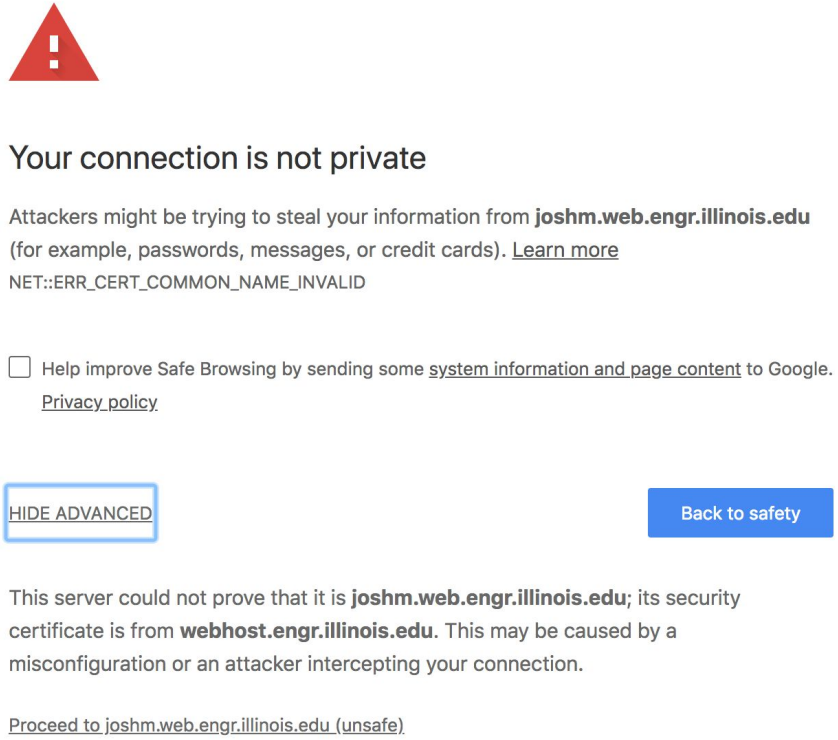
SSL & TLS (https)

- how you connect to websites
- TLS certificates - verify authenticity

heartbleed!

- client hello
- server hello
- client: change cipher spec
- server: change cipher spec
- now you have secure data

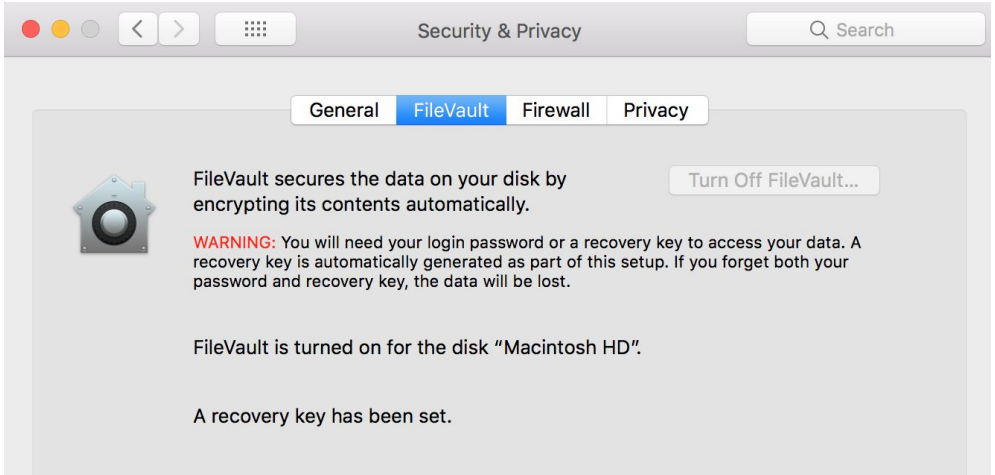




full disk encryption (FDE)

- filevault - ships on mac, enable it

- veracrypt: works on anything, a little more hardcore



if you want to learn more

get an applied cryptography textbook

do cryptopals.com

sigpwny.com time:

- SHA1 - hash
- rot13
- caesar cipher
- Keyed xor
- AES ECB

python refresher if you need one / are new