CSE 484 : Computer Security and Privacy

Software Security [Wrap-Up]
Cryptography [Intro]

Spring 2021

Tadayoshi Kohno
yoshi@cs

Thanks to Franzi Roesner, Dan Boneh, Dieter Gollmann, Dan Halperin, David Kohlbrenner, Yoshi Kohno, Ada Lerner, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...
Admin

• Lab 1: April 21
  • That is, sploits 1-3
  • When you are ‘done,’ stop changing those files.
  • Start early! You are encouraged to finish sploits 1-3 before April 21, and start on the other sploits before the final deadline

• Wednesday:
  • Gennie Gebhart, Acting Activism Director, Electronic Frontier Foundation (EFF)
  • https://www.eff.org/about/staff/gennie-gebhart
  • Title: Surveillance Self-Defense and Security Work in Civil Society
Attacker Model

Attacker can guess \textit{CandidatePwds} through some standard interface

- Naive: Try all $256^8 = 18,446,744,073,709,551,616$ possibilities
- Is it possible to derive password more quickly?

```python
PwdCheck(RealPwd, CandidatePwd)  // both 8 chars
    for i = 1 to 8 do
        if (RealPwd[i] != CandidatePwd[i]) then
            return FALSE
    return TRUE
```
Timing Attacks

• Assume there are no “typical” bugs in the software
  • No buffer overflow bugs
  • No format string vulnerabilities
  • Good choice of randomness
  • Good design

• The software may still be vulnerable to timing attacks
  • Software exhibits input-dependent timings

• Complex and hard to fully protect against
Other Examples

• Plenty of other examples of timings attacks
  • Timing cache misses
    • Extract cryptographic keys...
    • Recent Spectre/Meltdown attacks
  • Duration of a rendering operation
    • Extract webpage information
  • Duration of a failed decryption attempt
    • Different failures mean different thing (e.g., Padding oracles)
Side-channels

- **Timing** is only one possibility

- Consider:
  - Power usage
  - Audio
  - EM Outputs
General Principles

• Check inputs
• Check all return values
• Least privilege
• Securely clear memory (passwords, keys, etc.)
• Failsafe defaults
• Defense in depth
  • Also: prevent, detect, respond

• NOT: security through obscurity
General Principles

• Reduce size of trusted computing base (TCB)
• Simplicity, modularity
  • But: Be careful at interface boundaries!
• Minimize attack surface
• Use vetted components
• Security by design
  • But: tension between security and other goals
• Open design? Open source? Closed source?
  • Different perspectives
Does Open Source Help?

• Different perspectives…

• **Positive example?**
  • Linux kernel backdoor attempt thwarted (2003)
    (http://www.freedom-to-tinker.com/?p=472)

• **Negative example?**
  • Heartbleed (2014)
    • Vulnerability in OpenSSL that allowed attackers to read arbitrary memory from vulnerable servers (including private keys)
Vulnerability Analysis and Disclosure

• What do you do if you’ve found a security problem in a real system?

• Say
  • A commercial website?
  • UW grade database?
  • Boeing 787?
  • TSA procedures?

Breakout Groups:
What would you do? What ethical questions come up?
Vulnerability Analysis and Disclosure

• Suppose companies A, B, and C all have a vulnerability, but have not made the existence of that vulnerability public

• Company A has a software update prepared and ready to go that, once shipped, will fix the vulnerability; but B and C are still working on developing a patch for the vulnerability

• Company A learns that attackers are exploiting this vulnerability in the wild

• *Should Company A release their patch, even if doing so means that the vulnerability now becomes public and other actors can start exploiting Companies B and C?*

• *Or should Company A wait until Companies B and C have patches?*
Next Major Section of the Course: Cryptography
Terminology Note: “blockchain” and “crypto”

• Rising interest, mostly in the cryptocurrency space

• For this course: crypto means “cryptography”
Common Communication Security Goals

**Privacy of data:**
Prevent exposure of information

**Integrity of data:**
Prevent modification of information
Recall Bigger Picture

• Cryptography only one small piece of a larger system
• Must protect entire system
  • Physical security
  • Operating system security
  • Network security
  • Users
  • Cryptography (following slides)
• Recall the weakest link

• Still, cryptography is a crucial part of our toolbox
History

• Substitution Ciphers
  • Caesar Cipher
• Transposition Ciphers
• Codebooks
• Machines

• Recommended Reading: The Codebreakers by David Kahn and The Code Book by Simon Singh.
History: Caesar Cipher (Shift Cipher)

- Plaintext letters are replaced with letters a fixed shift away in the alphabet.

- Example:
  - Plaintext: The quick brown fox jumps over the lazy dog
  - Key: Shift 3
    
    | ABC | DEF | GHI | JKL | MNO | PQR | STU | VWX | YZ |
    |-----|-----|-----|-----|-----|-----|-----|-----|-----|
    | A   | B   | C   | D   | E   | F   | G   | H   | I   |
    | J   | K   | L   | M   | N   | O   | P   | Q   | R   |
    | S   | T   | U   | V   | W   | X   | Y   | Z   |

  - Ciphertext: WKHTX LFNEU RZQIR AMXPS VRYHU WKHOD CBGRJ
History: Caesar Cipher (Shift Cipher)

• ROT13: shift 13 (encryption and decryption are symmetric)

• What is the key space?
  • 26 possible shifts.

• How to attack shift ciphers?
  • Brute force.
History: Substitution Cipher

• **Superset of shift ciphers:** each letter is substituted for another one.
• One way to implement: **Add a secret key**
• Example:
  - Plaintext: ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - Cipher: ZEBRASCDFGHJKLMNOPQRSTUVWXYZ
• “State of the art” for thousands of years
History: Substitution Cipher

• What is the key space?
• How to attack?
  • Frequency analysis.

26! ~= 2^{88}

**Bigrams:**

- th 1.52%
- he 1.28%
- in 0.94%
- en 0.55%
- an 0.82%
- re 0.68%
- nd 0.63%
- at 0.59%
- on 0.57%
- nt 0.56%
- ha 0.56%
- es 0.56%
- st 0.55%

**Trigrams:**

1. the  6. ion  11. nce
2. and  7. tio  12. edt
3. tha  8. for  13. tis
4. ent  9. nde  14. oft
5. ing  10. has  15. sth