DEF CON 23

demonsaw

Crypto for Hackers
The Workshop

Eijah

v1.00
August 7th, 2015

Who am I?

- Founder
- Programmer
- Hacker

Prerequisites

- Laptop
- Linux (gcc 4.7), Windows (MSVC 2013)
- Crypto++
  - www.cryptopp.com
- Boost 1_58_0
  - www.boost.org
- Workshop Files
  - www.demonsaw.com
- Qmake
  - www.qt.io
  - Optional to create your own Makefiles
- Jom
  - www.qt.io
  - Because windows doesn't support make –j 4

Lesson 0 – Overview

- C++ 11
- Boost
- Crypto++
- Demoncrypt
- 8 Examples with Boost ASIO
- Download & configure

Lesson 0 – Workshop Files

- www.demonsaw.com
- Download workshop.zip on the downloads page
- Create a directory called defcon23
- Extract the files into the defcon23 directory

Lesson 0 – Boost

- Download boost 1_58_0
  - http://sourceforge.net/projects/boost/files/boost/1.58.0/
- Extract to the defcon23 directory
- Navigate to boost_1_58_0 directory
  - bootstrap
  - bjam --layout=system threading=multi runtime-link=static link=static variant=debug architecture=x86 address-model=64 --with-system
  - Copy boost/libs contents into defcon23/lib directory

Lesson 0 – Crypto++

- Download Crypto++ 5.6.2
  - http://www.cryptopp.com/#download
- Extract to the defcon23/cryptopp directory

Lesson 1 – Caesar’s Cipher

- https://en.wikipedia.org/wiki/Caesar_cipher
- Implement the Caesar's Cipher algorithm
- Send a random string to a buddy's machine
- Write code that will crack the cipher and display the plaintext data

Lesson 2 – Hash

- https://en.wikipedia.org/wiki/Cryptographic_hash_function
- Experiment with different hash algorithms (MD5, SHA1, SHA256, etc.)
- Print out a variety of hashes from cmd-line input
- Use the security/hex class to better format the output
- Send a message and hash digest (of that message) across the network
- What are some security concerns with this approach?
- Simulate a MITM attack
- Add a salt to your hash

Lesson 3 – HMAC

- https://en.wikipedia.org/wiki/Hash-based_message_authentication_code
- Experiment with different hmac algorithms
- Print out a variety of macs from cmd-line input
- Use the security/hex class to better format the output
- Send a message and MAC (of that message) across the network
- What are some security concerns with this approach?
- Simulate a MITM attack

Lesson 4 – Cipher

- https://en.wikipedia.org/wiki/Cipher
- Experiment with some AES candidate ciphers
- Use different key sizes (128, 192, 256)
- How do we generate a key size of a specific length, i.e. 128 bits?
- Experiment with other ciphers available in block_cipher.h
- Experiment with the Initialization Vector (IV)
- How could be use the IV in the real-world?
- Send an encrypted message and MAC (of that message) across the network
- What are some security concerns with this approach?
- Simulate a MITM attack
- Is this approach safe from hackers?
- What other steps must be done to prevent mischief?

Lesson 5 – PBKDF

- https://en.wikipedia.org/wiki/PBKDF2
- Experiment with some different PBKDF functions (See pbkdf.h)
- Use different passphrases, iterations, and salts
- How does PBKDF help us derive useful cipher keys?
- Send an encrypted message over the network
- What are some security concerns with this approach?
- What are some usability concerns with this approach?

Lesson 6 – Diffie Hellman

- https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange
- Generate a base, prime, and public key
- Send this data to the server
- Read the response data
- Extract the server's public key
- Update out DH object with the server's data
- Generate a shared key (this is super secret)
- Use the shared key to send an encrypted message across the network
- Decide upon a method to convert the DH shared key into a valid AES key (128, 192, or 256 bits)
- Encrypt a message and send it to the server

Lesson 7 – Social Crypto

- Social Crypto is a new model of security designed specifically for groups and individuals
- It is infrastructure agnostic
- Built on the foundation of traditional security
- Can leverage symmetric and asymmetric keys
- Social Crypto is based on the premise that individuals and groups share common knowledge and experiences
- It then uses these as shared sources of entropy from which we derive extremely complex signatures that are fed into cryptographic routines
- Social Crypto uses networks (e.g. the Internet) as a universal and constantly changing Random Number Generator (RNG) from which deterministic and reproducible bits are created.

Lesson 8 – AACS

- I may (or may not) have hacked AACS a few years ago
- Be that as it may, let's hack the shit out of AACS together
- I've included a binary file with 3 AACS Device Keys (in a grand parent, parent, and child AES-G3 Left Branch relationship) hidden within the data somewhere
- To make things easier (and faster) I've aligned each possible key to 16 byte offsets
- Implement the AES-G3 routine, recover the keys, and tell me the hierarchical relationship
- If I really did hack AACS a few years ago, you might be using some of the actual code that I used back then

Summary

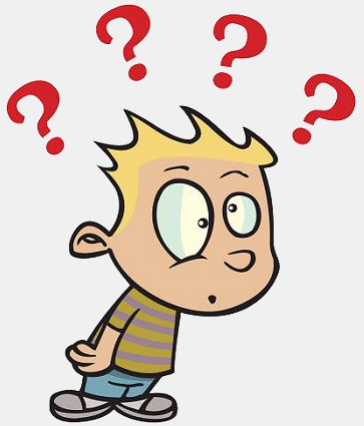
- Thank you for believing in the Right to Share
- The demonsaw promise
  - 100% free, no ads, no installs, no malware, no bundled software, no logging, no tracking, no bullshit
- Your continued support
  - Suggestions, bug fixes, beta testing
  - One person can make a difference
  - Email, Twitter
- The best is yet to come

Questions?

www.demonsaw.com

eijah@demonsaw.com

@demon_saw



Eijah