DEF CON 23





demonsaw

Crypto for Hackers

Eijah



v1.00
August 7th, 2015

Hello World

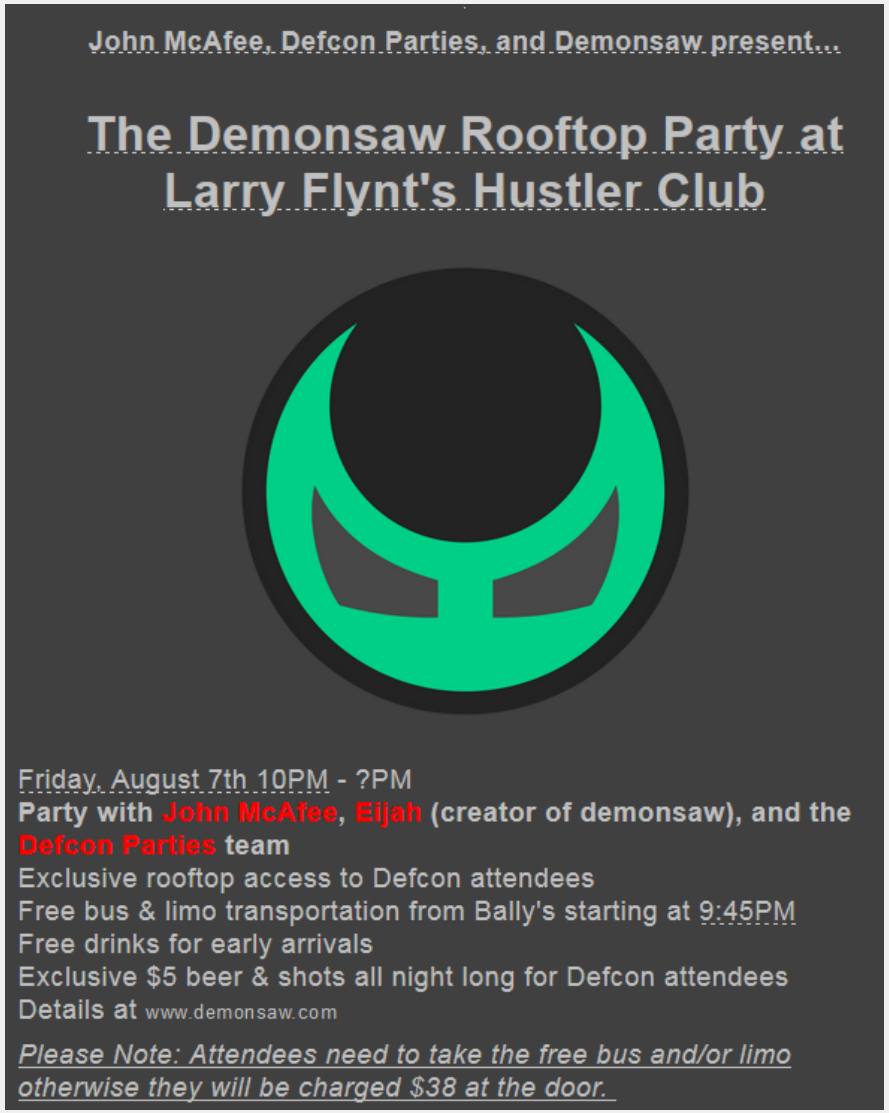
“Shall we play a game?”

– Joshua/WOPR

Free Party Tonight!

- www.demonsaw.com
- www.defconparties.com





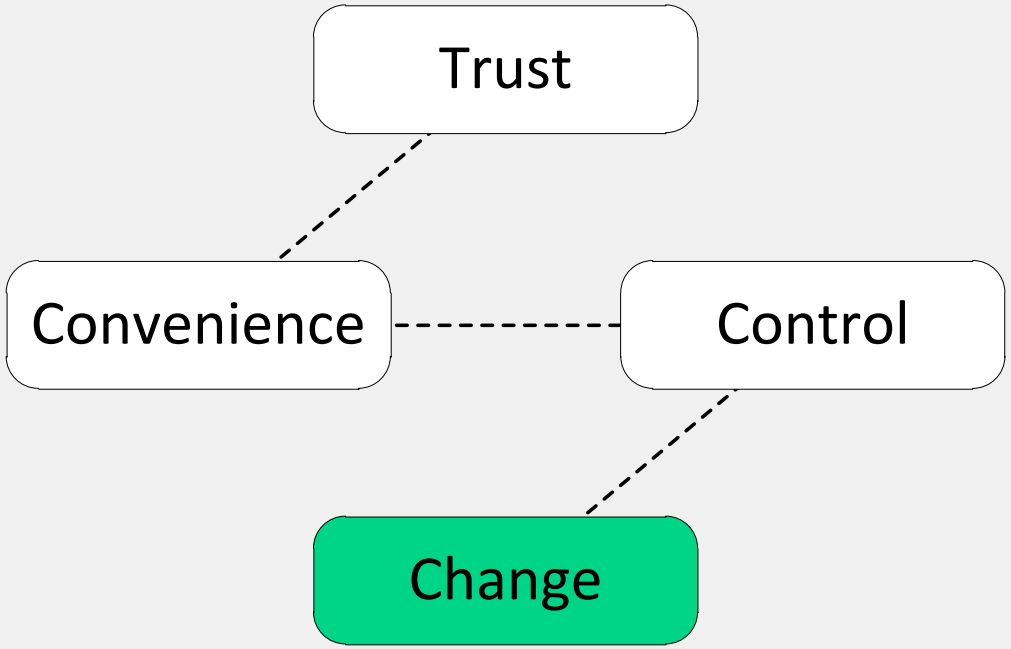
Who am I?

- Founder
- Programmer
- Hacker



Last year at Defcon…

- Saving Cyberspace by Reinventing File Sharing
  - We’re losing our right to privacy and free speech
  - The abuse of power by the strong
- Crypto is really difficult
  - Lifelong dedication
  - Can we filter out the noise?
  - Can we make it easy to understand?
- We need to take back control
- Becoming powerful again



A World Without Secrets

"Know thy self, know thy enemy. A thousand battles, a thousand victories.“

– Sun Tzu, general and author of The Art of War

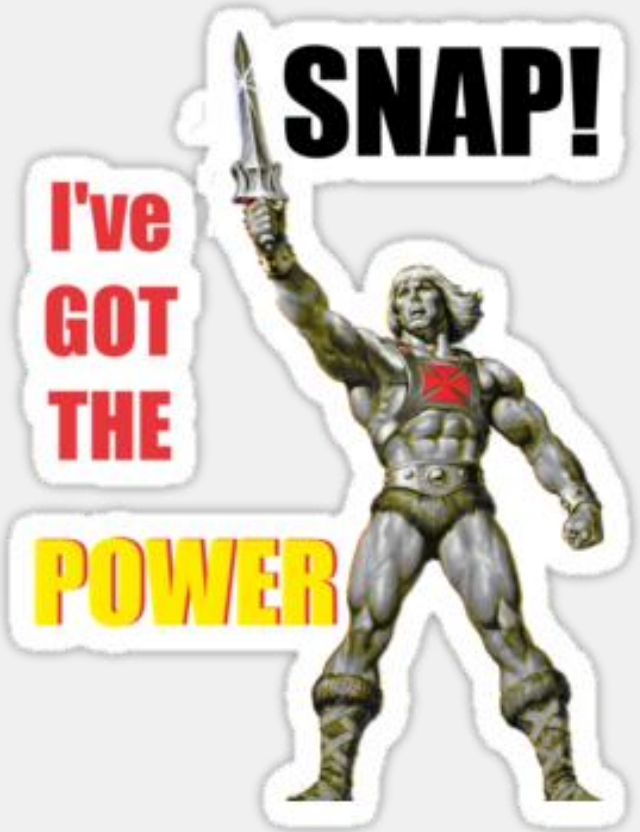
A Formidable Ally

- Crypto is our strongest weapon
- Considered a military weapon in the USA until 1992
- Many different uses
  - Protect out secrets
  - Expose the secrets of others
- Poor crypto implementations
- A common denominator to the Internet
- What is the government thinking?
  - Banning encryption?
  - Providing backdoors?



Becoming More Powerful

- Technology as the deciding factor
- We are stronger than governments & corporations
- Crypto is a complex and difficult subject
- The secret
  - Ubiquitous
  - Pervasive
  - Easy
- The will of the people
- Battle cry for a new generation of hackers



Battle Cry



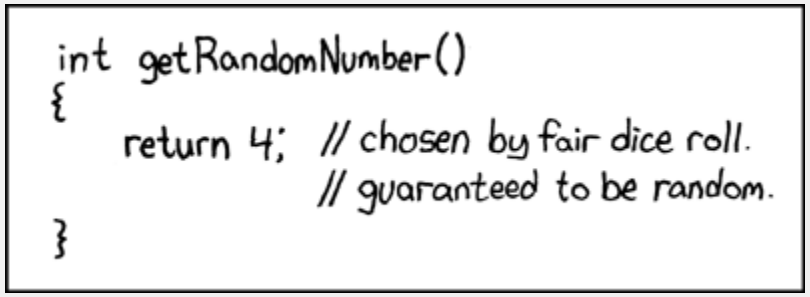
The Algorithms

“Encryption is the defense against the dark arts.”

– Edward Snowden

Terminology

- Crypto(graphy)
  - The practice and study of techniques for secure communication in the presence of third-parties.
- Cipher
  - An algorithm for performing encryption or decryption.
- Encryption, Decryption
  - The process of encoding messages or information in such a way that only authorized parties can read it (and vice versa).
- Plaintext, Cleartext
  - Unencrypted messages or information.
- Ciphertext
  - Encrypted messages or information.
- Random Number Generator (RNG)
  - Device designed to generate a sequence of numbers or symbols that lack any pattern



Terminology

- Key
  - A parameter that determines the functional output of a cryptographic cipher.
- Hash (Function)
  - A 1-way cryptographic conversion function which is considered practically impossible to invert.
- (Message) Digest
  - The output of a hash function.
- Symmetric Encryption Algorithm
  - Algorithms that use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext.
- Asymmetric Encryption
  - Algorithms that use different cryptographic keys for encryption of plaintext and decryption of ciphertext.

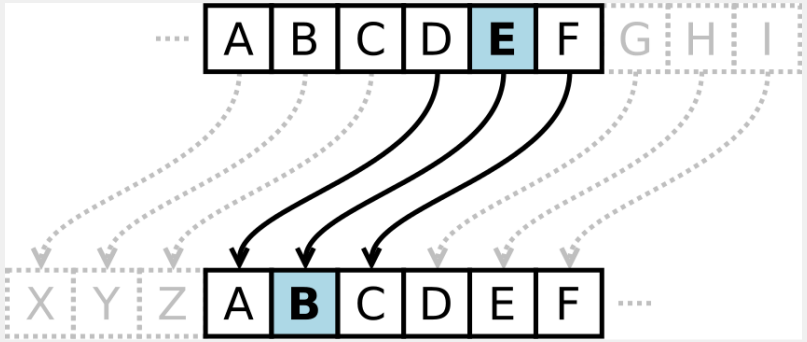


Crypto Libraries

- Crypto++
  - Free C++ class library of cryptographic schemes
  - http://www.cryptopp.com
- Demoncrypt
  - Open-source C++ wrapper around the most important Crypto++ algorithms.
  - Crypto foundation of demonsaw v2.0
  - http://www.demonsaw.com
- Algorithms
  - Ciphers (AES)
  - Hash (MD5, SHA)
  - HMAC
  - Key Agreement Schemes (Diffie Hellman)
  - Password-based Key Derivation Functions (PBKDF2)

Ciphers

- Algorithm for performing encryption or decryption
  - Plaintext → ciphertext
  - Ciphertext → plaintext
- Usages
  - Symmetric, Asymmetric
  - Everywhere (Software, hardware, networking, databases, DRM, etc.)
- Algorithms
  - Rijndael (AES), Serpent, Twofish, RC6, MARS
  - 128, 192, 256 bit keys
- Notes
  - NIST AES Standardization Process



Ciphers

AES

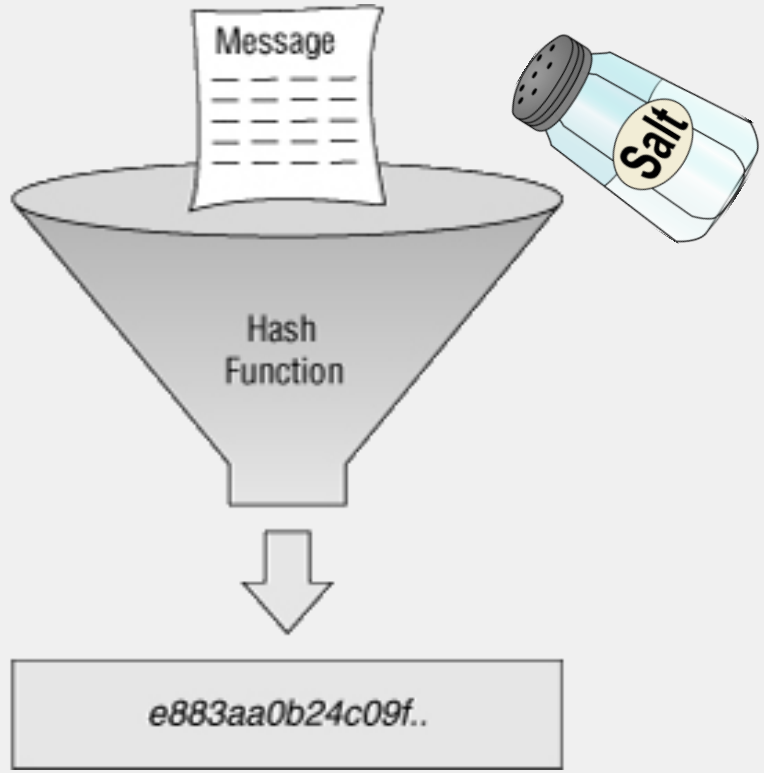



```
aes cipher;
cipher.set_key(key);
string ciphertext = cipher.encrypt("The Cloud is a lie");
cout << "CIPHERTEXT: " << hex::encode(ciphertext) << endl;
// 542f25fcf9c8564433bee34d1122fab30c349c7d7ded1967c5ff3abac42f734b

string plaintext = cipher.decrypt(ciphertext);
cout << "PLAINTEXT: " << plaintext << endl;
// The Cloud is a lie
```

Hash Functions

- One-way conversion function
  - Message → (Message) Digest
  - Practically impossible to reverse engineer
  - Fixed sized output w/ optional salt
- Usages
  - Verify file integrity, hashing passwords, digital signatures
  - Checksums, keyed data structure indexes
- Algorithms
  - MD5, SHA-1, SHA-2, SHA-3
- Concerns
  - Collisions (limited key space)
  - Rainbow attacks
  - Length extension hacks



Hash Functions

MD5

```
md5 md;
string digest = md.compute("The Cloud is a lie");
cout << "MD5: " << digest << endl;
// 759806471b250e2031c5257c01382a21
```

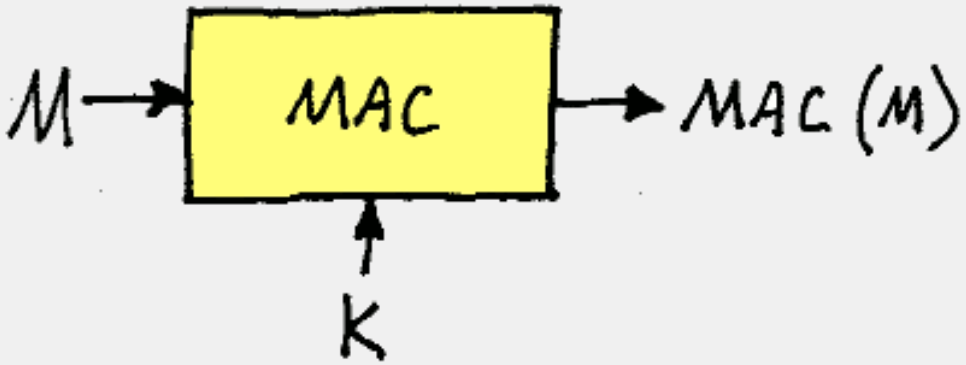


SHA256

```
sha256 sha;
string digest = sha.compute("The Cloud is a lie");
cout << "SHA256: " << digest << endl;
// 6aad0fcc90862e6b3c5cea078a0a35c0327f62671477fc5689abaa5f783c8cdf
```

Hash-Based Message Authentication Codes

- Hash function with a user-provided key
  - Message Authentication Code (MAC)
  - Provides authenticity & message integrity
  - Prevents rainbow attacks
  - Key is not in hash
- Usages
  - Can be used in place of hashes
  - Building block for other applications (PBKDF2)
  - Signing messages
- Algorithms
  - MD5, SHA-1, SHA-2, SHA-3
- Concerns
  - Strength of the input key



Hash-Based Message Authentication Codes

HMAC MD5

```
hmac_md5 hmac;
hmac.set_key("Believe in the Right to Share");
string mac = hmac.compute("The Cloud is a lie");
cout << "HMAC MD5: " << hex::encode(mac) << endl;
// 888e3e9cedd4f1a0d9a7c6e76af16afa
```

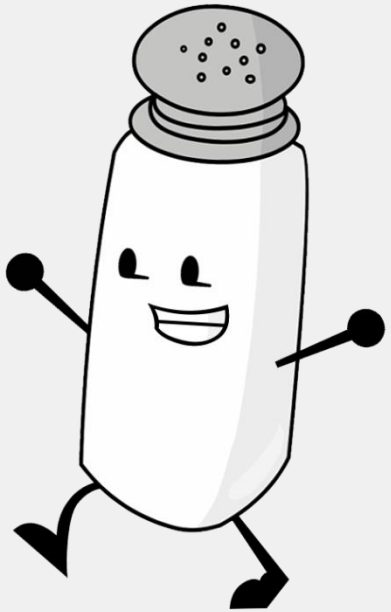


HMAC SHA512

```
hmac_sha256 hmac;
hmac.set_key("Believe in the Right to Share");
string mac = hmac.compute("The Cloud is a lie");
cout << "HMAC MD5: " << hex::encode(mac) << endl;
// cb1fa1e58e17a6b8c87e476e6725251e114b5cd51b20946facca15cc497595f1
```

Password-Based Key Derivation Functions

- Generating a key from a password/phrase
  - Convert user-friendly strings into keys
  - Choose PRF, salt, iterations, output size
- Usages
  - Mutual shared secret derivation
  - Input for other crypto algorithms
- Algorithms
  - PBKDF2
- Concerns
  - Computationally more intense
  - Character encoding is important



Password-Based Key Derivation Functions

PBKDF2 HMAC MD5

```
pkcs5_pbkdf2_hmac_md5 pbkdf;
pbkdf.set_salt("demonsaw 2.0");
string key = pbkdf.compute("Believe in the Right to Share");
cout << "PBKDF2 HMAC MD5: " << hex::encode(key) << endl;
// 0552acc1243e62c9c35acdcbc6714f30
```

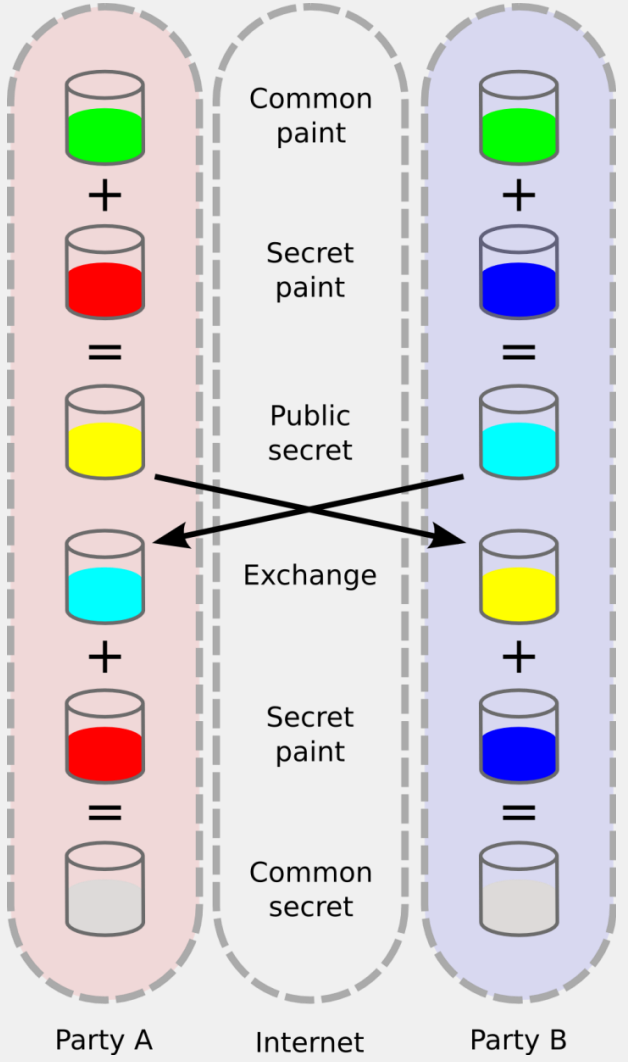


PBKDF2 HMAC SHA512

```
pkcs5_pbkdf2_hmac_sha256 pbkdf;
pbkdf.set_salt("demonsaw 2.0");
string key = pbkdf.compute("Believe in the Right to Share");
cout << "PBKDF2 HMAC SHA256: " << hex::encode(key) << endl;
// b1403d5360549dc580eb9cc3cf46bc3a5dfe871a8ada19a37a4429a8e7e85e00
```

Key Agreement Schemes

- Generates secret key common to both parties
- Benefits
  - Same shared key created, without key exchange
  - Both parties influence the outcome
  - Insecure channel communication
- Usages
  - WPS, SSL/TLS, SSH, VPN
- Algorithms
  - Diffie-Hellman
- Notes
  - Asymmetric
  - Man-in-the-Middle attack
  - Unique key sizes



Key Agreement Schemes (128 bit)

Party A



```
diffie_hellman dh_1(128);
const auto& base_1 = dh_1.get_base();
const auto& prime_1 = dh_1.get_prime();
const auto& public_key_1 = dh_1.get_public_key();

cout << "Base: " << hex::encode(base_1) << endl;
cout << "Prime: " << hex::encode(prime_1) << endl;
cout << "Public Key: " << hex::encode(public_key_1) << endl;

// Base: 03
// Prime: 952908c5e136753c1c8b44411396ecf3
// Public Key: 3f19880e8bf951f94c6dc5578242dde5
```

Key Agreement Schemes (128 bit)



Party B

```
diffie_hellman dh_2(prime_1, base_1);
const bool status_2 = dh_2.compute(public_key_1);
const auto& public_key_2 = dh_2.get_public_key();
const auto& shared_key_2 = dh_2.get_shared_key();

cout << "Public Key: " << hex::encode(public_key_2) << endl;
cout << "Shared Key: " << hex::encode(shared_key_2) << endl;
// Public Key: 0a54116dc11c41ac6c2a3a95a9c61715
// Shared Key: 29cabdd82222c74e7dc44dea9b113a21
```

Party A

```
const bool status_1 = dh_1.compute(public_key_2);
const auto& shared_key_1 = dh_1.get_shared_key();

cout << "Shared Key: " << hex::encode(secret_key_1) << endl;
// Shared Key: 29cabdd82222c74e7dc44dea9b113a21
```

Key Agreement Schemes (2048 bit)

Base

02



Prime

e85d7232623aa7988a681b914ad0d4aa1d5ba17a8be374be808dc26ed988c5b8d7e0998382a6db54ed9a75a436f4b
fa17d732d355d3835b7ec9ba131ffe9e7652235f11a9f70a27d440ccf193d6609f98e86ce17000fdb04cdf2d74da2c1e1c5f
8b7ba0fafb5617e7e29f5509b3be2d7dadc4da9f4d9f4442325e978eeb6b2521e5bc097c6ef859cfd736e6413b144b9e48a0
440e905312054315e06d12aa881833672afa467ecc40ab9002db2fa9c9fee39e4c375957f45937fca6c8c9927c8a986d3ec8
9b7059fb3cd66fccfb69a94870afe6b81b190163e152f24895e936243869063ed9d818a02b3efff23812ecba5e07a3b564f2
76e5515a4fffd45bf87

Public Key (A)

1ae20832c2fd50afab8cb387ccdada3ae620b5f22555dde17863008d4f05ede1d9b762bbb1568641af6d4c5d
5f8f2cfa441660b386e77302f8885452f786385aafaf0c55802e10890ba2da3e16df2ebf3b0dbe466fce6713463dc0a498a8
cca2ee946dcb28517b1a7bb768175aa5cdf26c3bf1c54f92a1ee7fe06dfdb058844853539115be1f76e14ab7f6230268e033
9aa687b72f47836cde28876345ca390c2425a92f3bea053168bf70adc325363d3ffc66a0be7aac5438dc43d82343586ee096
a0a142292dbad376eedb960b65975870f2c796df07df7e78b3f16961ed33ff4690d3c9cd300a3d5ac4359a8440aa3a125a61
d9aba9a20fd5e960eceb965c

Key Agreement Schemes (2048 bit)

Public (B)



```
35c8abc9cac1e9ca365005b22ab98c8d1efa5656d99286333e5de37a793ba8f8c235886fa3f5c6b6164c662b
21b18ea03b901297a6be623dc8ec5dad106b2e7cd7ab9a339187ea4142f593e414c8992408e0bdede22d3bb51517801431ef
ce5f01b4f81a15c61acd948f03f9655242537e31e668ec998ce0ef1c11e99afcf467048934290bc1358cfc41b0b742954c1c
09a539ce2d1845a735b1ad5e12359b6d271195c0fca1b3b8b7135b05857b8f9c5b66f0517c1335eeab0f6916f1848abc6c33
15811af5af417a23c531f02bde7167d268494c009d39815c27d2e8610ac021c26e6a64729c1888dad19f9b026fa752039b8b
a18ce6e0285d8060b9b70c20
```

Shared (A, B)

```
980b1baeff1745794b384f46743d31b69b830b109e28f81aca96a3c38fe8538dd64ff835b830472a266eca8f
f82ede682c57b59c83640b86e0a46edd25923f7e089da9540cb73dd2db9ddde8514c782f9cd2dd5de6a08ed7cae357e1c4ff
fb6ecf386f979b2bdd1c755fb6b1efa049d3612c7d51d39185740a2cd3e788104af2391095d4c7b4b56b2d1b6460a665458d
874c8eaee1ec57abe0f7a5335e55f41e32e1c9077582dbb542f7d4b5481d651ca2f9d748731bf9a878b84d2ce822f68e2c6f
3a6aa712da5ed31a6beb7c4dd889309353304 54fe809bed9564ffc5772d9e2caebadc749dc806990d9e682f57df33ac7d739
89fedefa71d73b18b2b26765
```

Securing our Future

“I am regularly asked what the average Internet user can do to ensure his security. My first answer is usually 'Nothing; you're screwed'.”

– *Bruce Schneier*

Demonsaw 1.0

- Defcon 22
- A good first step
- C#, Windows-only
- Learning experience
- A little buggy
- Secure, Anonymous, Free, Everywhere?

Demonsaw 2.0

- Goals
  - Everywhere (Unified code base: C++ 11, Boost, Crypto++)
  - Simplify the interface
  - Increase security
  - Add more features
- Promise
  - 100% free, no ads
  - No installs, no malware, no bundled software
  - No logging, no tracking
  - No bitcoin mining, no bullshit



Demonsaw 2.0

- Features
  - Windows, Linux, OSX, Raspberry Pi
  - Modern UI
  - Secure chat
  - Opt-in features
  - Command-line interface
  - Customizable themes
  - Customizable file/folder hashes with salts
  - Customizable Diffie-Hellman key & prime sizes
  - Social Cryptography



Social Cryptography

- New way of thinking about security
- Favors individuals & small groups
- Doesn't require anybody else
- Built on the foundation of traditional security
- Share things in common with those we interact with
- Contextual awareness via shared knowledge/experiences
- Map this contextual awareness into binary form
- Use this binary form as entropy for input into crypto routines



Demo



Summary

“A hacker to me is someone creative who does wonderful things.”

– Tim Berners-Lee

Next Steps

- demonsaw 3.0
  - Android
  - Hardware isolation on Android TEE
  - Streaming, File-sync, Torrents
  - B2B space
- Your continued support
  - Suggestions, bug fixes, beta testing
  - One person can make a difference
  - Email, Twitter



Next Steps

- Community Involvement
  - DemonBucket
  - Suggestions, bug fixes, beta testing
  - One person can make a difference
  - Email, Twitter
- Tonight's Party with John McAfee & Defcon Parties
- Believe in the Right to Share

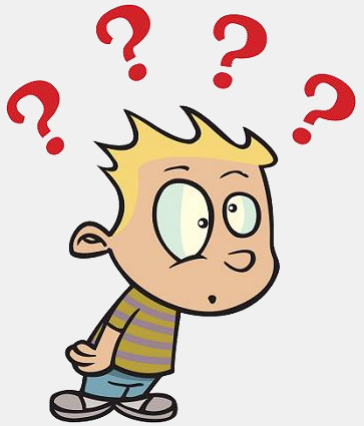


Questions?

www.demonsaw.com

eijah@demonsaw.com

@demon_saw



Eijah

References

- http://cryptopp.com/
- https://en.wikipedia.org/wiki/Export_of_cryptography_from_the_United_States
- https://en.wikipedia.org/wiki/Symmetric-key_algorithm
- https://en.wikipedia.org/wiki/Public-key_cryptography
- https://en.wikipedia.org/wiki/Advanced_Encryption_Standard_process





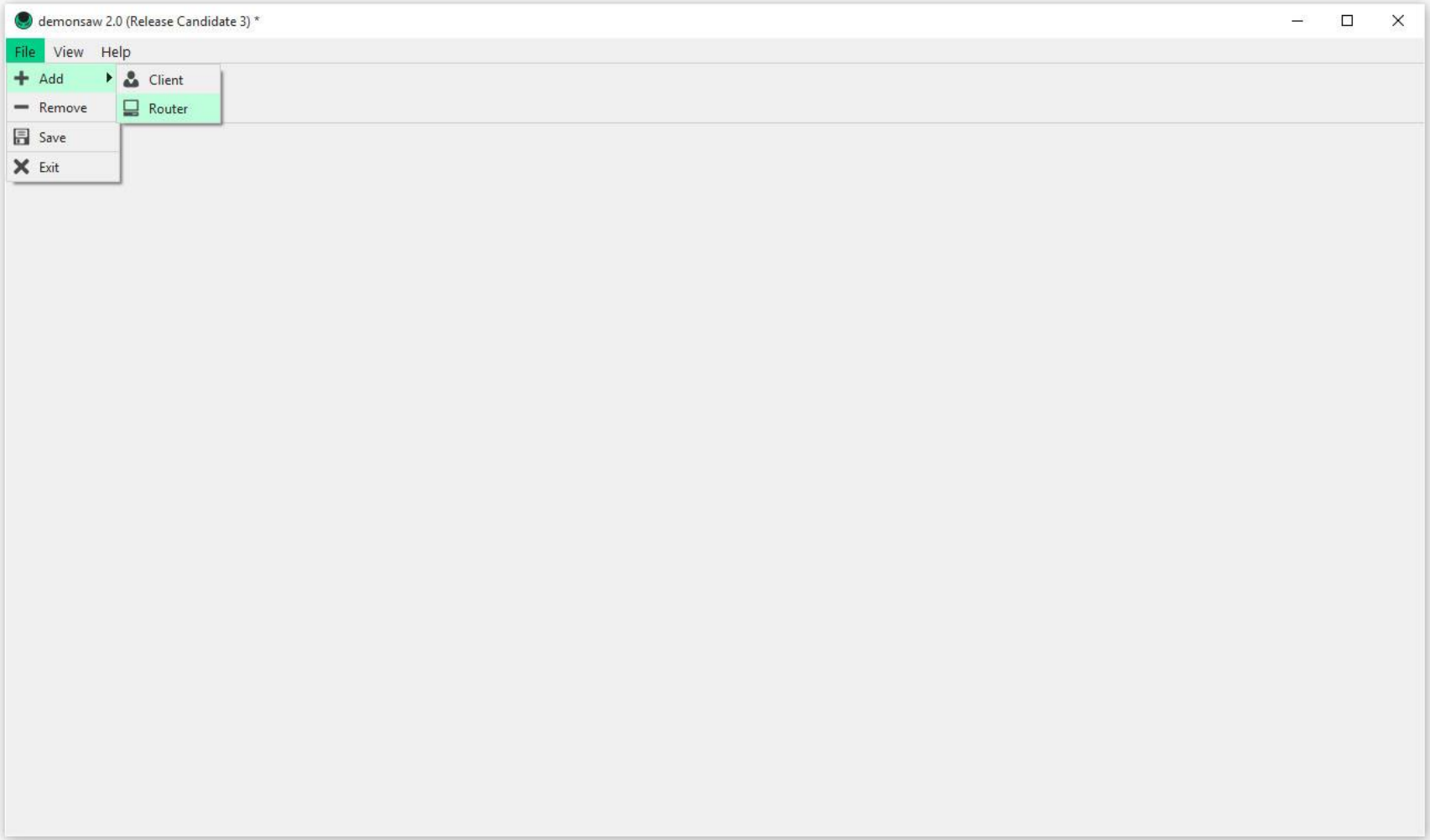
DEF CON 23



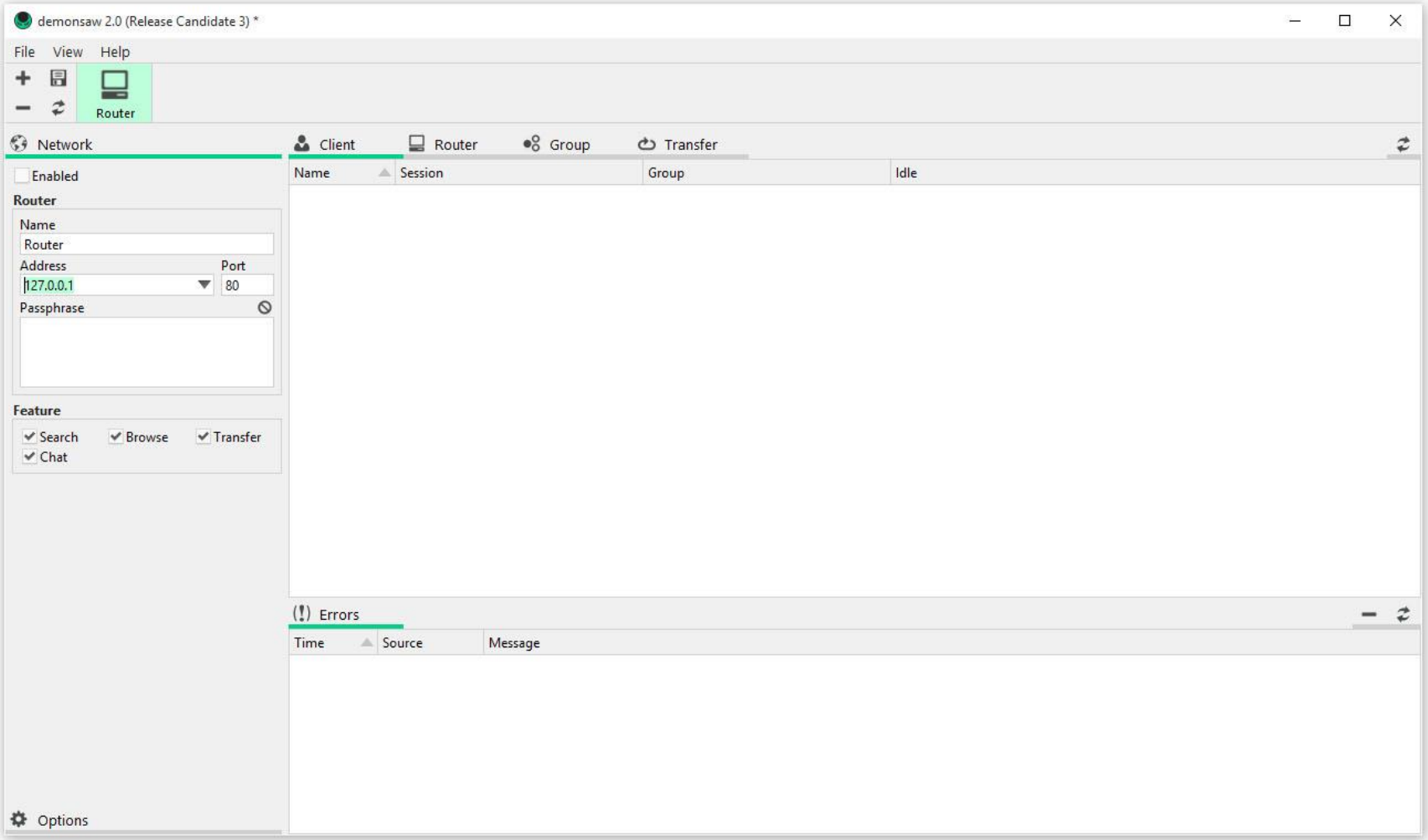


demonsaw

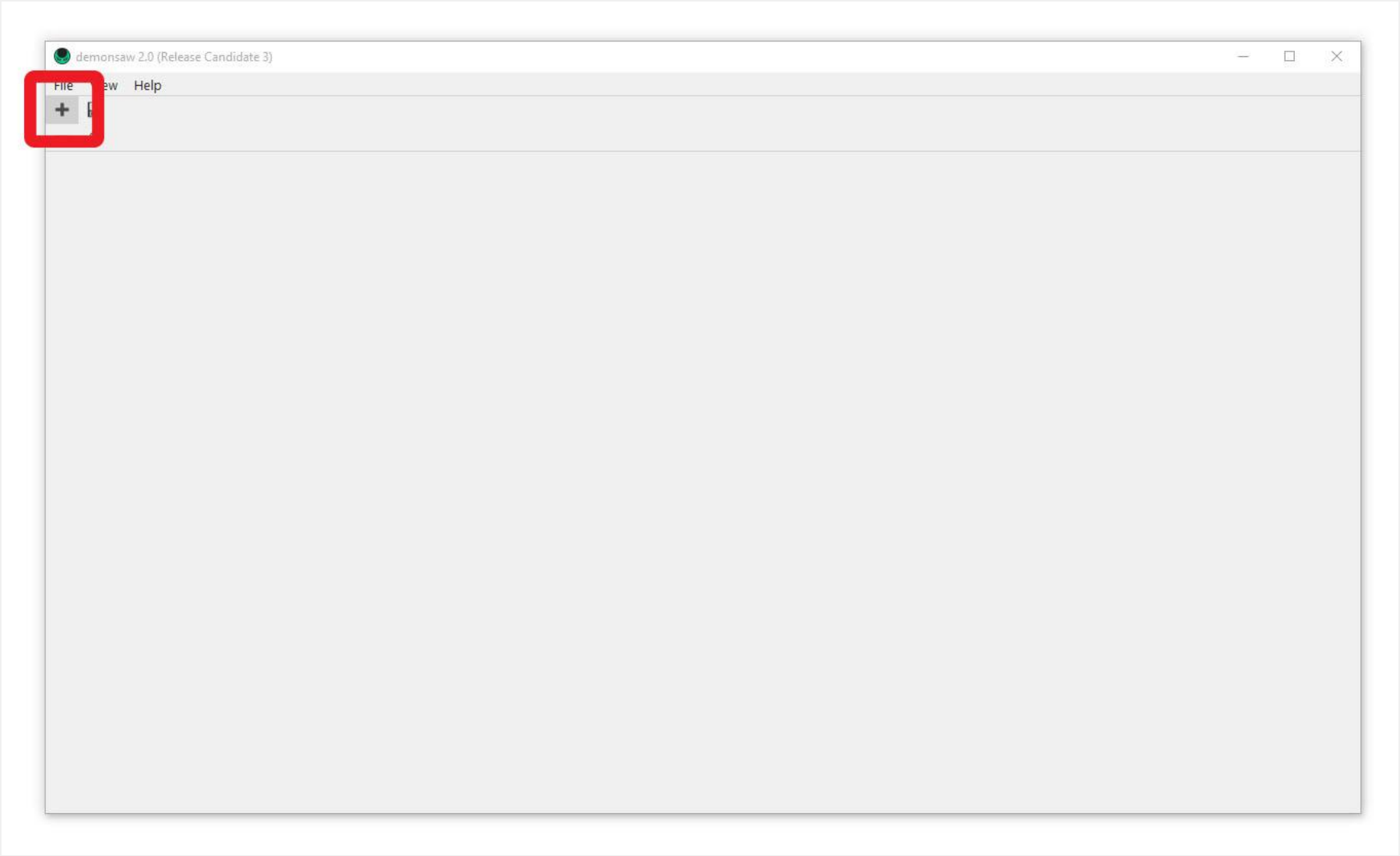
Demo – Add a Router



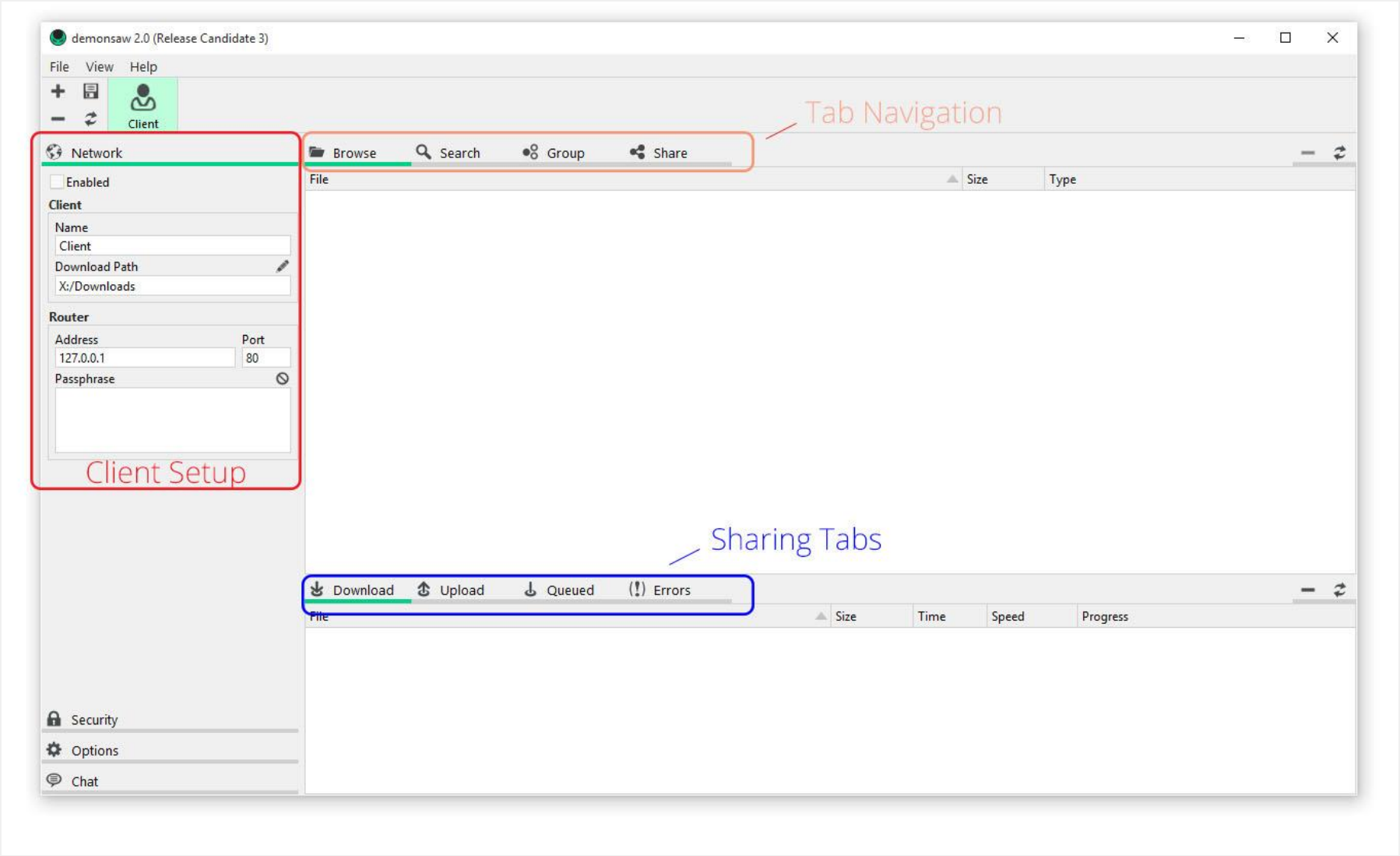
Demo – Configuring the Router



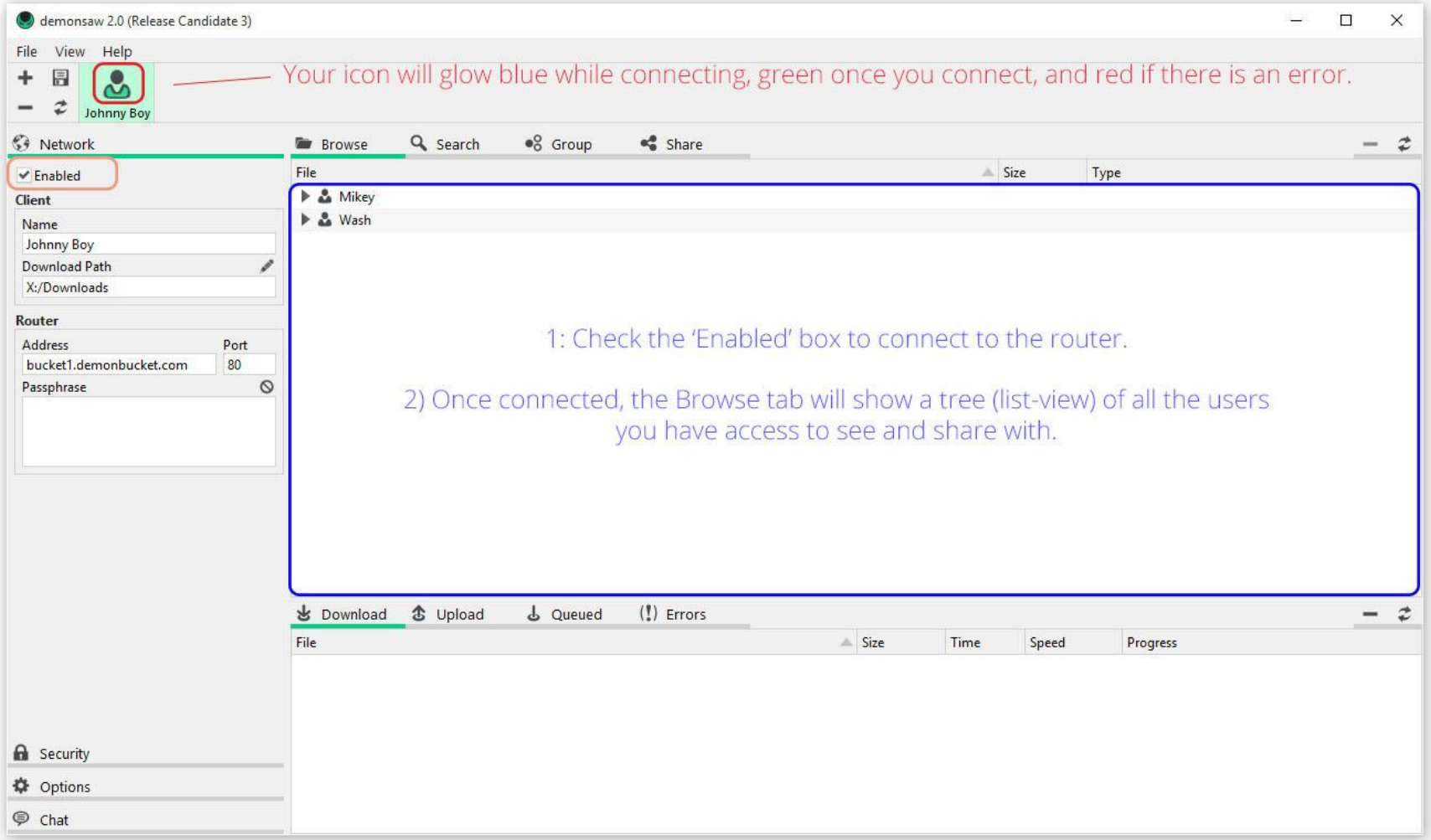
Demo – Add a Client



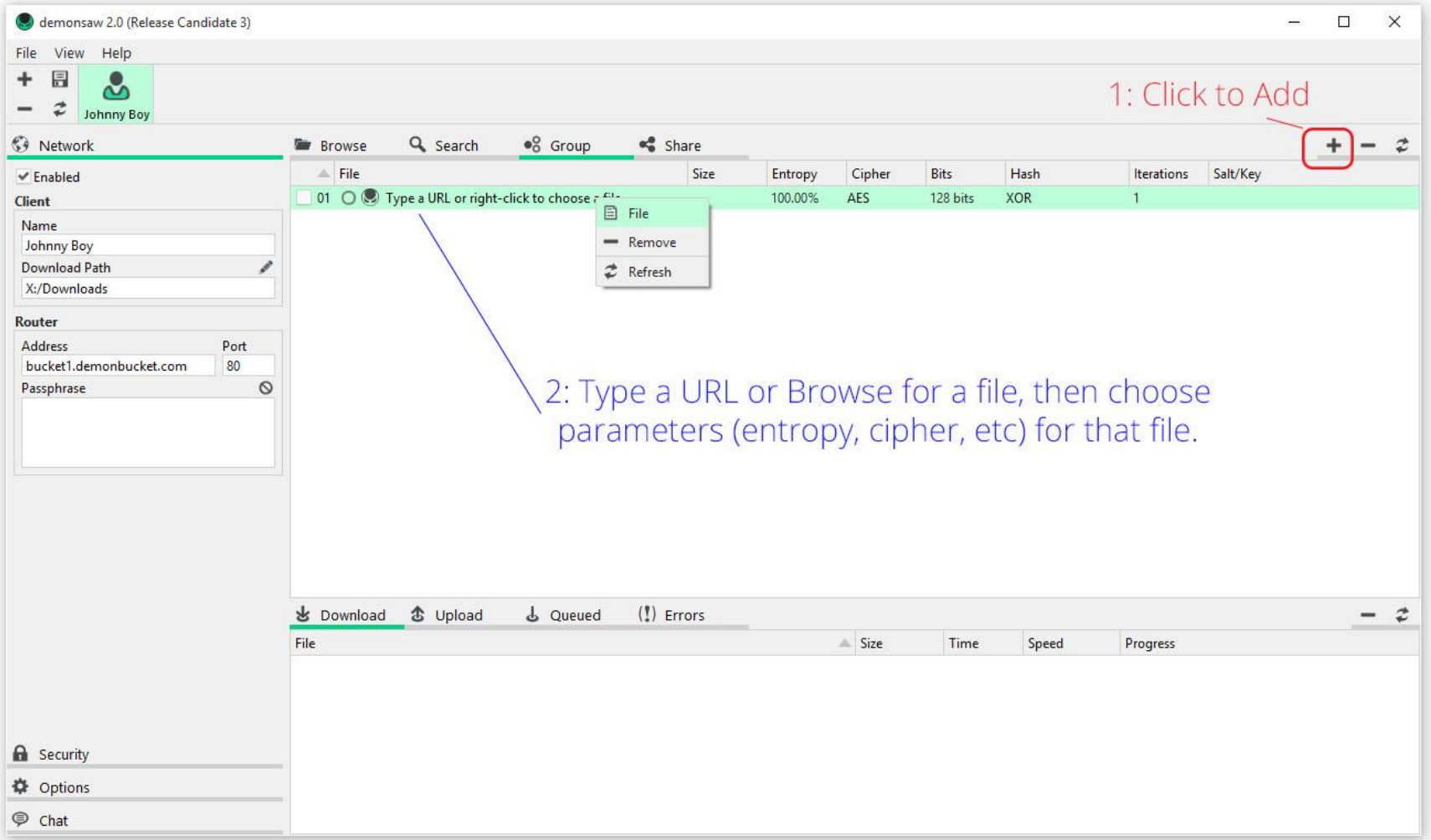
Demo – Network Options



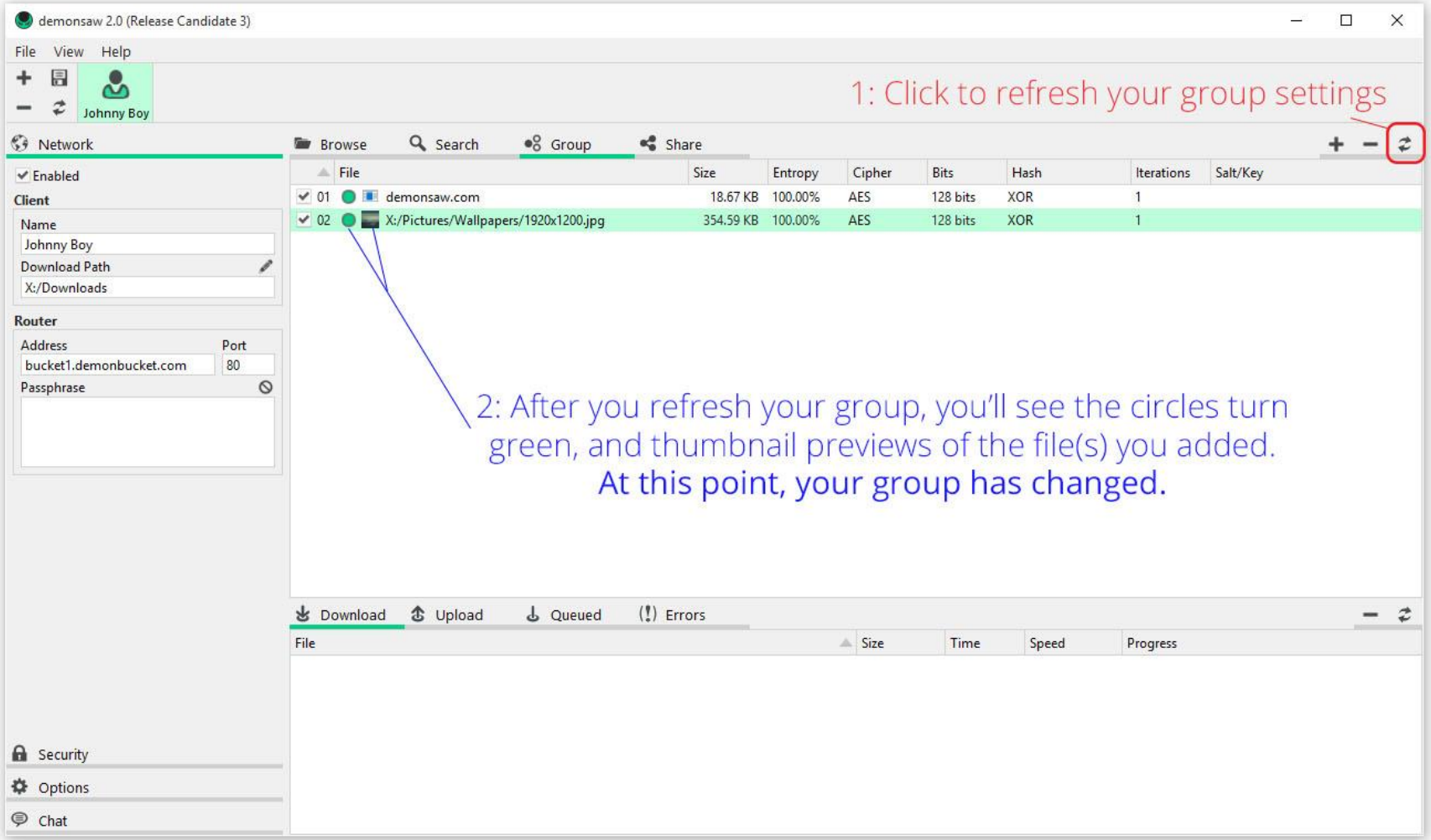
Demo - Connecting



Demo – Create/Join a Group



Demo – Social Crypto



Demo – Sharing with your Group

