Essential Crypto for Pen Testers (Without the Math)

Joshua Wright, InGuardians
Senior Security Analyst
josh@inguardians.com

Webcast – April 26 2010

Outline

- ➡ Introduction
- Essential crypto skill development
- Applying crypto analysis
- Summary and conclusion

Crypto and Pen Testing

- Many pen testers skip over crypto in assessments
  - Math, algorithms, more math, etc.
- With some essential skills, you can attack cryptographic mistakes
  - Expanding your skill repertoire
  - New opportunities to exploit systems

What We're Targeting

- It is uncommon to identify crypto flaws in widespread protocols (TLS, PGP, etc)
- There is a lot more crypto to attack out there
  - Less-common but critical standards
  - Proprietary applications
  - *Other* wireless protocols
  - Removable storage drives
  - Custom web-app session cookies, etc.
  - Database field encryption

Outline

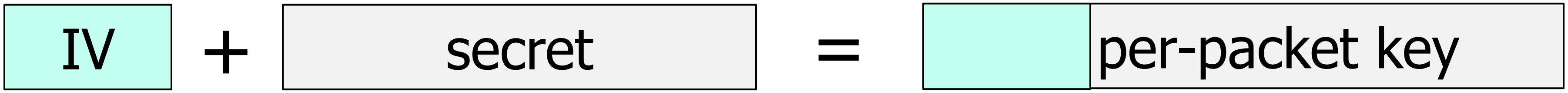
- Introduction
- ➡ Essential crypto skill development
- Applying crypto analysis
- Summary and conclusion

Stream Ciphers

- Encrypt one bit at a time
- Encrypted length is the same as the plaintext
  - 63 bytes ciphertext means 63 bytes plaintext
- Examples include RC4, A5/1, E0
- Cipher generates a keystream
- Keystream is XOR'd with plaintext to produce ciphertext
  - Sorry, that was a little math

Critical Evaluation: IV Handling?

- Law of Stream Ciphers: Can never use the same key twice
- We accomplish this by mixing a per-packet value with each key
  - Initialization Vector (IV)
  - IV is not a secret (usually sent in packet)
- Must rotate key before IV's repeat



Block Ciphers

- Encrypt data one fixed-length block at a time
- Must pad the last few bytes to an even block length
  - 8-byte block length with 64 bytes ciphertext is 57 – 64 bytes plaintext
- Examples include: AES, DES, 3DES, Blowfish

Block Cipher Modes

- Block ciphers introduce a "mode"
  - Some block cipher modes provide better security than others
- Any block cipher can be used with various modes (AES-CTR, 3DES-CBC)
  - Key Vendor Question: "What block cipher mode do you use?"
- We'll look at ECB, CBC modes

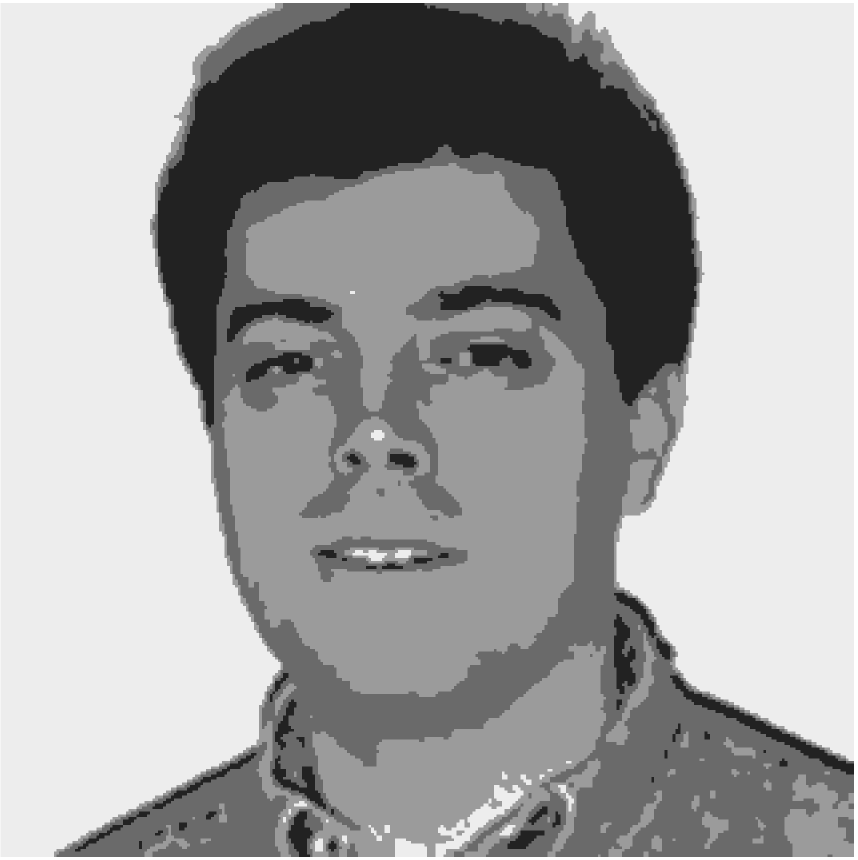
ECB Mode

- Electronic Cookbook Mode
- Encrypts each block with the same key
  - Critical issue: same plaintext blocks encrypt to matching ciphertext blocks
  - Attacker can identify repetitious blocks of plaintext
  - Commonly an issue with lots of 0's
- Reveals interesting content about plaintext

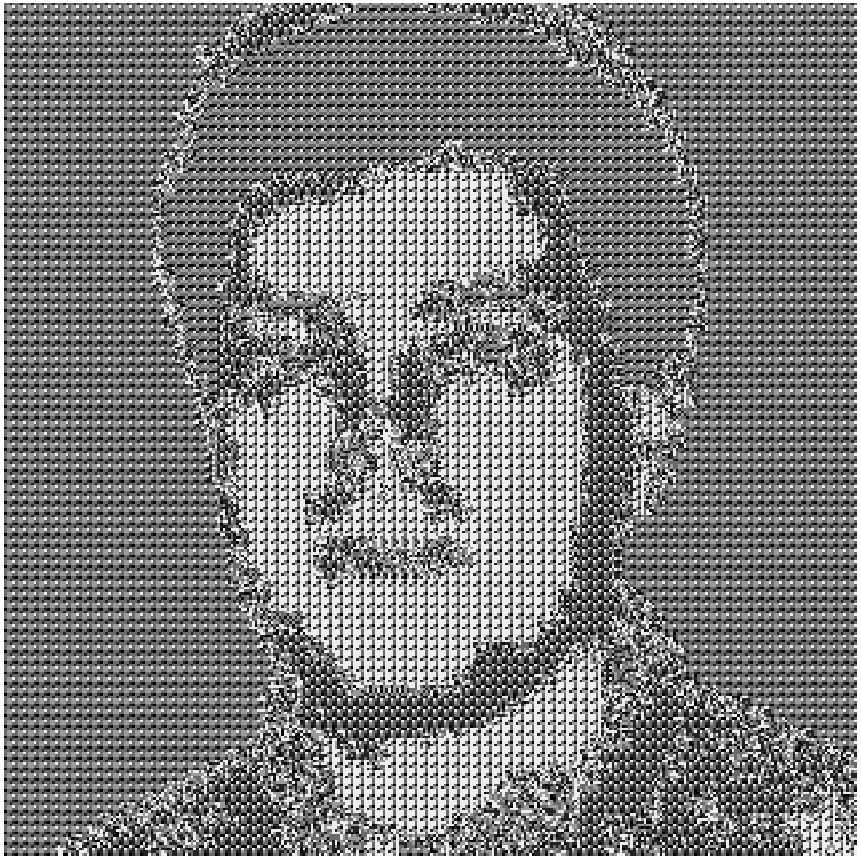
```
$ xxd -p aes-256-ecb-encrypted-secrets.bin
b2e5d275b8a9d7fd05ee7b58a1e242f1890a6a8b763c4ddb97f642c5f7d8
edb5b2e5d275b8a9d7fd05ee7b58a1e242f1f04eab49bff6e46fb8b5fd99
```

Levi Johnston?

Unencrypted



AES-ECB-256 Encrypted



CBC Mode

- Cipher Block Chaining Mode
- Encrypt a block using the key
- Encrypted block is then XOR'd with the next plaintext before encrypting
- Adds variety to each block
  - Solves the ECB same-plaintext = same-ciphertext problem
- What about the first block then?

CBC IV

- CBC uses an IV as the first "plaintext" block to encrypt
  - Encrypted IV is XOR'd with first byte of real plaintext
  - IV "should" not repeat
- Repeating IV reveals plaintext patterns

Three encrypted packets – what's the problem here?

```
$ openssl enc -aes-128-cbc -in packet1 -K $KEY -iv $IV | xxd -p
0a940bb5416ef045f1c39458c653ea5ad172ce43bf147f4dffa206c1d372ddca
$ openssl enc -aes-128-cbc -in packet2 -K $KEY -iv $IV | xxd -p
06cf727e3dc3bd52ce98916d71dd233bfc60a567fea20a5e3191ab952c4a6491
$ openssl enc -aes-128-cbc -in packet3 -K $KEY -iv $IV | xxd -p
0a940bb5416ef045f1c39458c653ea5ad172ce43bf147f4dffa206c1d372ddca
```

More At The Summit

- At the pen-test summit, we'll get into more detail
  - Also covering CTR mode, common issues in stream and block ciphers
- We're short on time today, so let's jump into an analysis example

Outline

- Introduction
- Essential crypto skill development
- ➡ Applying crypto analysis
- Summary and conclusion

Network Traffic Sample

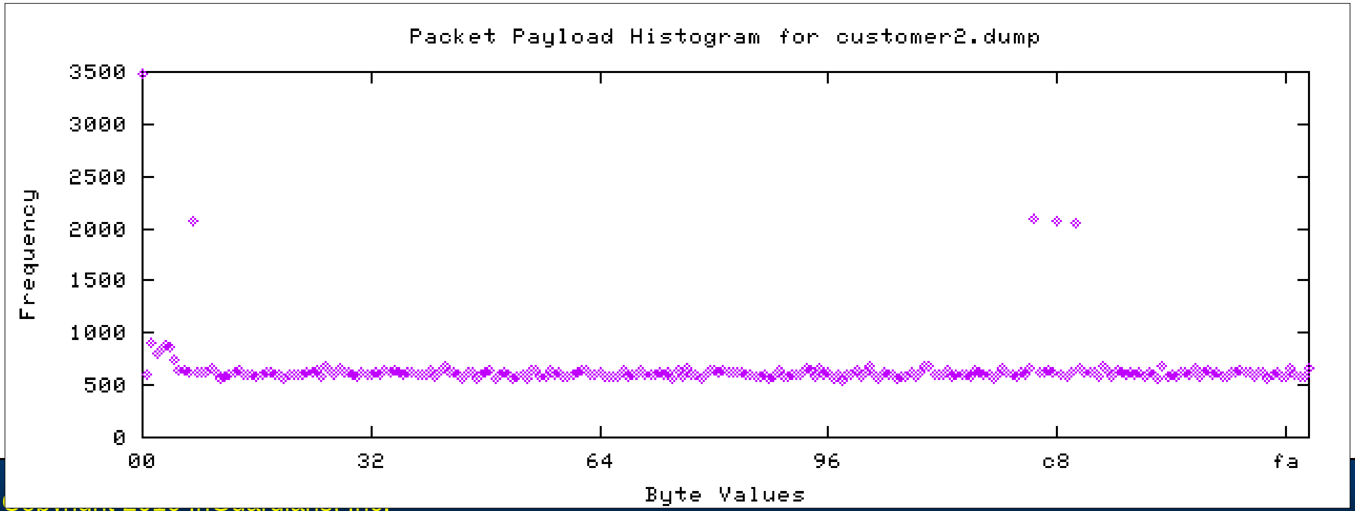
```
Packet 1  591f5377 5cd731c7 9bc02d08 8bac34
Packet 2  31b98481 e1
Packet 3  eb3c6307 1cb1cdc4 a3e1a69c 6c3f71f9
Packet 4  d8a3390c fb48aa61
Packet 5  591f5377 5cd731c7 9bc02d08 8bac34
Packet 6  204f0eb3 f1
```

- Proprietary wireless protocol traffic
- Header information removed, packets shortened for space, simplicity
- We need to evaluate this implementation

pcaphistogram

- Is this traffic encrypted at all?
- Histogram: plot frequency of each byte of encrypted payload
  - Encrypted data should have roughly equal distribution of byte values

```
$ pcaphistogram customer2.dump | gnuplot
$ display customer2.png
```



Lots more crypto visualization tools at the summit!

Evaluating the Data

- Frame lengths 15, 5, 8 bytes
  - Indicative of a stream cipher
- Some repetition in encrypted packets
  - Lack of unique IV for each packet
  - That's a big no-no, especially with stream ciphers
- Commutative property of XOR

P1 XOR P2 = C1 XOR C2

Stream Cipher IV Collision

- Known-ciphertext attack opportunity
- Able to create a packet with text we specify and observe the encrypted counterpart

1	Normal Encryption	Plaintext	⊕	Keystream	=	Ciphertext
2	Normal Decryption	Ciphertext	⊕	Keystream	=	Plaintext
3	Commutative XOR	Our Packet (known p'text)	⊕	Ciphertext	=	Keystream
4	Pwned	Unknown Ciphertext	⊕	Keystream	=	Plaintext FTW!

Stream Cipher IV Collision

- We could generate our own traffic for this system
- Identified corresponding ciphertext

```
plainknown =    (     0x80, 0x11, 0x39, 0xa5, 0x00, 0x00, 0x00, 0x00,
                      0xff, 0xff, 0xff, 0xff, 0x00, 0x44, 0x00, 0x43 )
cipherknown =   (     0x59, 0x1f, 0x53, 0x77, 0x5c, 0xd7, 0x31, 0xc7,
                      0x9b, 0xc0, 0x2d, 0x08, 0x8b, 0xac, 0x34, 0x26 );
cipherunknown = (     0xeb, 0x3c, 0x63, 0x07, 0x1c, 0xb1, 0xcd, 0xc4,
                      0xa3, 0xe1, 0xa6, 0x9c, 0x6c, 0x3f, 0x71, 0xf9 );
for i in xrange(0,len(plainknown)):
       cipxor = cipherknown[i] ^ cipherunknown[i]
       print("%02x"%(cipxor ^ plainknown[i])),
print("")
```

```
C:\dev>python ivcoltest.py
32 32 09 d5 40 66 fc 03 c7 de 74 6b e7 d7 45 9c
```

Outline

- Introduction
- Essential crypto skill development
- Applying crypto analysis
- ➡ Summary and conclusion

Conclusion

- Useful to build skill set for basic crypto analysis
  - Very little math required
- Stream ciphers and IV reuse
- Block ciphers and modes
- Walkthrough – IV collision and known plaintext attack

Crypto w/o Math at the Summit …

- More critical crypto skill development
- Tools you can use!
  - Stuff from my previously-unreleased stash and other public tools
  - More data visualization examples
- More attack examples
  - Database storage, HTTP cookies, standards-based protocols
- Critical questions for your vendor's crypto implementation
- Recommendations – where to go from here!

Questions?

- Pen Test Summit 2010, June 14-15
  - Baltimore, MD (Hilton across from Camden)
- Another awesome line-up this year
  - Vinnie Liu, Dan Kaminsky, HD Moore, Jonathan Ham, Paul Asadoorian, Jeremiah Grossman, Joshua Wright, Larry Pesce, Jabra, Johnny Cache and more!
- Come for the content, attendee interaction, networking and more!

9 days left for early-bird sign-up savings of $350!
www.sans.org/pen-testing-summit-2010