# Cryptography Pitfalls

John Downey | @jtdowney

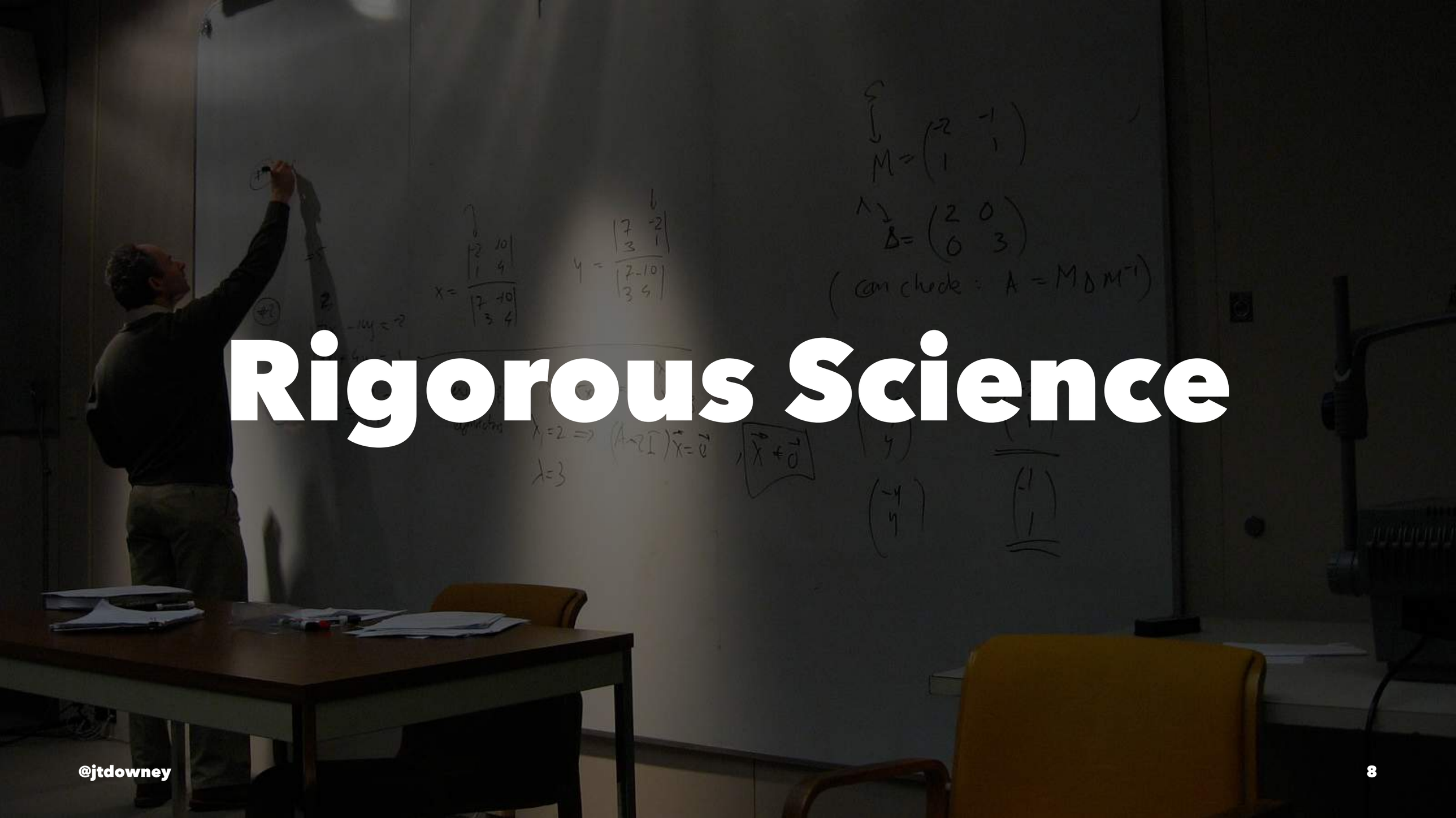
# Braintree


PayPal


Confidentiality

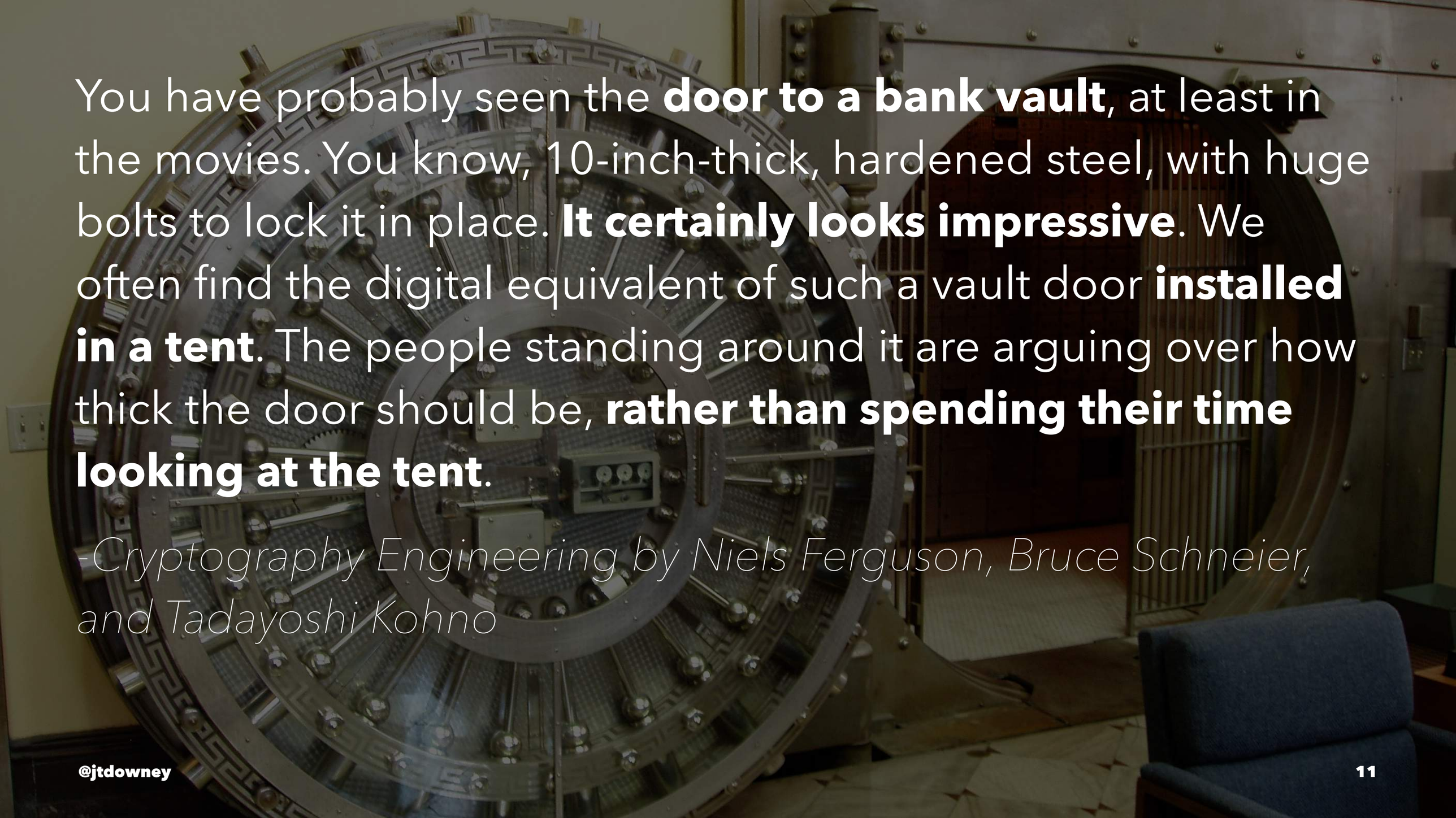

Authentication

# Identification



# Rigorous Science

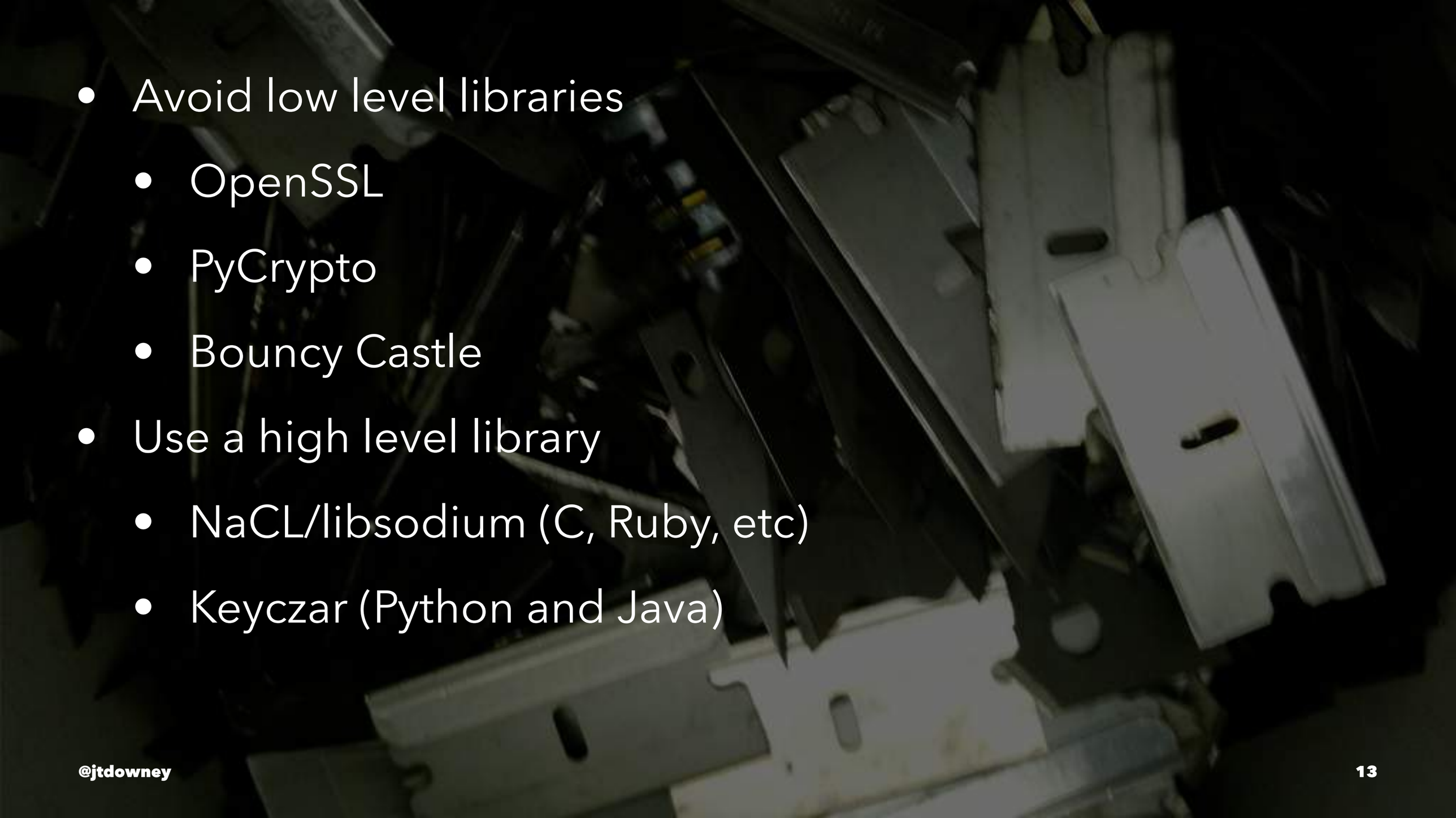
# Peer Review



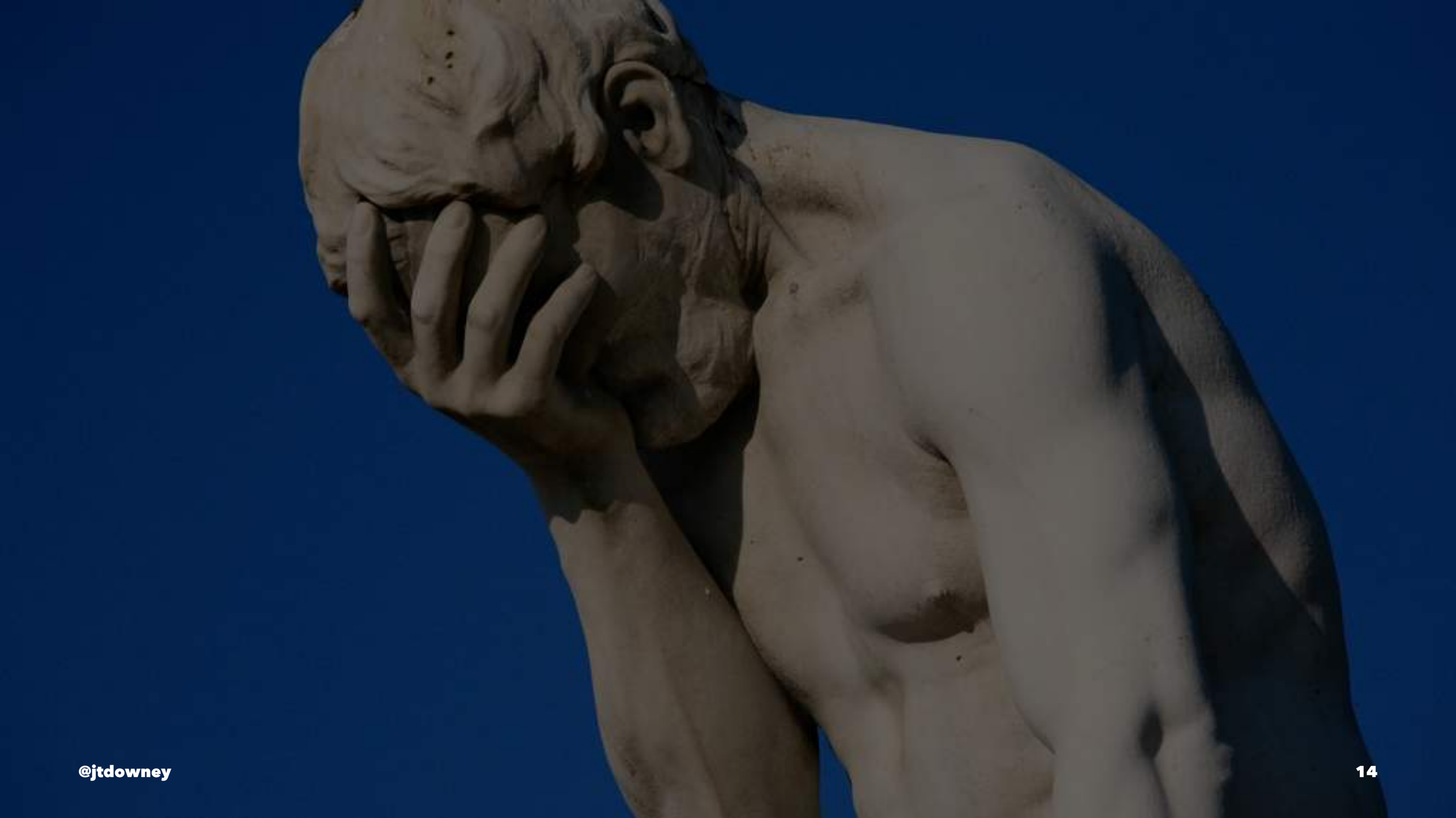
You have probably seen the **door to a bank vault**, at least in the movies. You know, 10-inch-thick, hardened steel, with huge bolts to lock it in place. **It certainly looks impressive**. We often find the digital equivalent of such a vault door **installed in a tent**. The people standing around it are arguing over how thick the door should be, **rather than spending their time looking at the tent**.

*-Cryptography Engineering by Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno*

- For data in transit
  - Use TLS (née SSL), SSH, or VPN/IPsec
- For data at rest
  - Use GnuPG



- Avoid low level libraries
  - OpenSSL
  - PyCrypto
  - Bouncy Castle
- Use a high level library
  - NaCL/libsodium (C, Ruby, etc)
  - Keyczar (Python and Java)



# Random Number Generation

- Randomness is a central part of any crypto system
- Used to generate:
  - Encryption keys
  - API keys
  - Session tokens
  - Password reset tokens

# Pitfalls

1. **Not using a cryptographically strong random number generator**
2. Broken random random number generators
3. Not using random data when it is required

# I Forgot Your Password: Randomness Attacks Against PHP Applications*

George Argyros
*Dept. of Informatics & Telecom.,*
*University of Athens,*
argyros.george@gmail.com

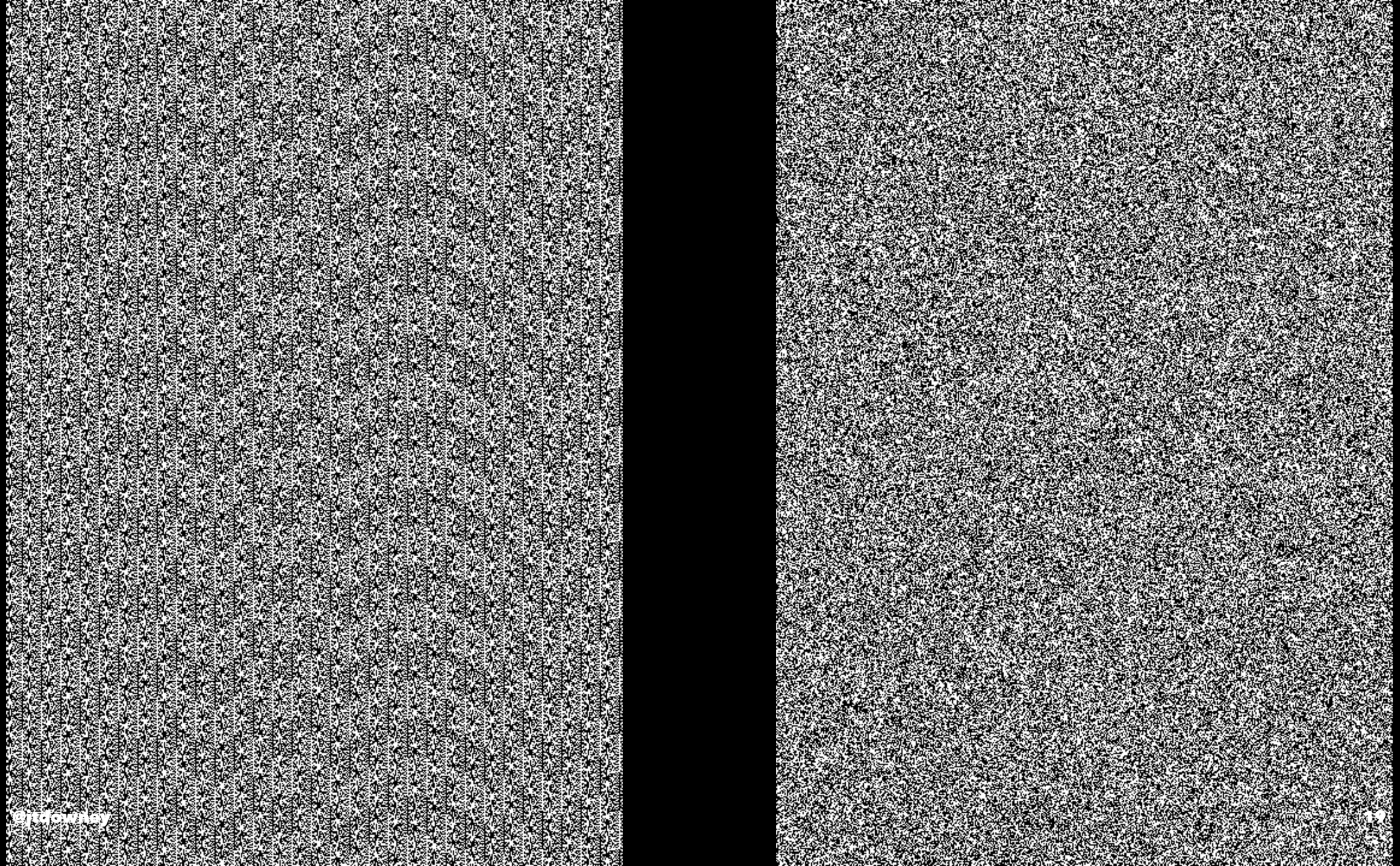
Aggelos Kiayias
*Dept. of Informatics & Telecom.,*
*University of Athens,*
aggelos@di.uoa.gr
*& Computer Science and Engineering,*
*University of Connecticut, Storrs, USA.*

## Abstract

We provide a number of practical techniques and algorithms for exploiting randomness vulnerabilities in PHP applications.We focus on the predictability of password reset tokens and demonstrate how an attacker can take over user accounts in a web application via predicting or algorithmically derandomizing the PHP core randomness generators. While our techniques are designed for the PHP language, the principles behind

PHP for example lacks a built-in cryptographically secure PRNG in its core and until recently, version 5.3, it tottaly lacked a cryptographically secure randomness generation function.

This left PHP programmers with two options: They will either implement their own PRNG from scratch or they will employ whatever functions are offered by the API in a "homebrew" and ad-hoc fashion. In addition, backwards compatibility and other issues (cf. section 2), often push the developers away even from


@rdownay

# Pitfalls

1. Not using the right random number generator
2. **Broken random random number generators**
3. Not using random data when it is required

# After Debian's epic SSL blunder, a world of hurt for security pros

## Admins: Heal thy certificates

21 May 2008 at 18:47, Dan Goodin 0 3 69

It's been more than a week since Debian patched a massive security hole in the library the operating system uses to create cryptographic keys for securing email, websites and administrative servers. Now the hard work begins, as legions of admins are saddled with the odious task of regenerating keys too numerous for anyone to estimate.

The flaw in Debian's random number generator means that OpenSSL keys generated over the past 20 months are so predictable that an attacker can correctly guess them in a matter of hours. Not exactly a



**Related topics**

Phishing, Hackers, Xss, Privacy


### Most read

Windows 10 upgrade ADWARE forces its way on to Windows 7 and 8.1

Fanbois designing Windows 10 – where's it going to end?

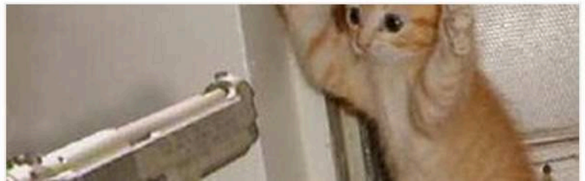
Facebook flings PGP-encrypted email at world+dog. Don't lose your private key

Force Touch tweak: Apple 15-inch MacBook Pro with Retina Display

Holy SSH-it! Microsoft promises secure logins for Windows PowerShell

### Spotlight

**Low price, big power: Virtual Private Server picks for power nerds**



```
MD_Update(&m,buf,j);
```

Don't add uninitialised data to the random number generator. This stop valgrind from giving error messages in unrelated code. (Closes: #363516)

```
/* DO NOT REMOVE THE FOLLOWING CALL TO MD_Update()! */
MD_Update(&m,buf,j);
/* We know that line may cause programs such as
purify and valgrind to complain about use of
uninitialized data.  The problem is not, it's
with the caller.  Removing that line will make
sure you get really bad randomness and thereby
other problems such as very insecure keys. */
```

# Concern mounts as Google confirms Android cryptographic vulnerability

Security fears as Symantec claims more than 360,000 Android apps are using SecureRandom service linked to bitcoin wallet flaw

**Stuart Dredge**
theguardian.com, Thursday 15 August 2013 12.58 EDT

Jump to comments (30)


Users of Android apps that generate Bitcoin wallets have been advised to update. Photograph: Linda Nylind for the Guardian



Article history

**Technology**
Android · Bitcoin · Google · Smartphones · Mobile phones

**More news**

**Related**

**5 Sep 2013**
BBC iPlayer Android app now allows downloads for offline viewing

**2 Sep 2013**
20 best Android apps this week

**23 Aug 2013**
Angry Birds Star Wars II to launch for iOS, Android



**Today's best video**

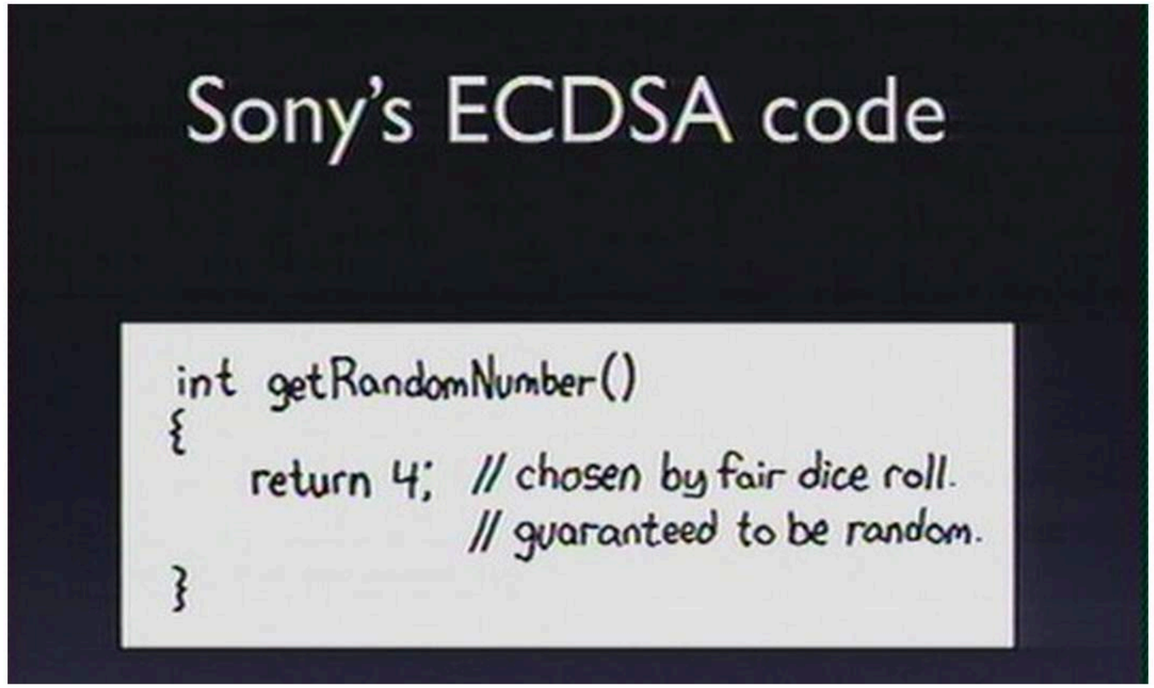

**iPhone 5C and 5S launch: spoof promo**
Innovatively breaking new boundaries but yet breaking no boundaries – and a cheap version for the 'peasants'

# Pitfalls

1. Not using the right random number generator
2. Broken random random number generators
3. **Not using random data when it is required**

# Hackers obtain PS3 private cryptography key due to epic programming fail? (update)

by Sean Hollister | | December 29th 2010 at 7:48 pm




The 27th annual Chaos Communication Conference already hacked encrypted GSM calls with a $15 cellphone, but there was a second surprise in store this morn -- the souls who

# Recommendations

- Unix-like
  - Read from /dev/urandom
- Windows
  - `RandomNumberGenerator.Create()` (.NET)
  - `CryptGenRandom` (Windows)

# Hash Functions

**DESCRIP** Phot **AL RECORD**

Age, 31 years
Height, 5 feet 7-1/8 inches
Weight, 153 pounds
Build, medium
Hair, medium chestnut
Eyes, grey
Complexion, medium
Occupation, machinist
Marks and scars, 1/2 inch scar back left hand; scar middle upper lip; brown mole between eyebrows
Mustache

John Dillinger

**er**, #14395, received **y**, Pendleton, Indiana, September 16, 1924; crime, assault and battery with intent to rob and conspiracy to commit a felony; sentences, 2 to 14 years and 10 to 20 years respectively;

As John Dillinger, #13225, received State Prison, Michigan City, Indiana, July 16, 1929; transferred from Indiana State Reformatory; paroled under Reformatory jurisdiction, May 10, 1933; parole revoked by Governor - considered as delinquent parolee;

As John Dillinger, #10587, arrested Police Department, Dayton, Ohio, September 22, 1933; charge, fugitive; turned over to Allen County, Ohio, authorities;

As John Dillinger, received County

- Often called a fingerprint
- One way
  - Not reversible (can’t find person without fingerprint DB)
- Ideally, no two people with same fingerprint (no two inputs)

# Pitfalls

1. **Using weak/old algorithms**
2. Misunderstanding checksums
3. Length extension attacks

# MD5 considered harmful today

## Creating a rogue CA certificate

December 30, 2008

**Alexander Sotirov, Marc Stevens,
Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik, Benne de Weger**

---

**Latest news**

[Dec. 31, 2008] Responses from Verisign (RapidSSL), Microsoft and Mozilla.
[Jan. 2, 2009] Responses from TC TrustCenter and RSA, and a US-CERT Vulnerability Note.
[Jan. 8, 2009] Video and audio files of the 25C3 presentation are available from CCC.
[Jan. 15, 2009] Response from Cisco.
[Mar. 11, 2009] Our paper with details on improved MD5 chosen-prefix collision construction is available.
[Apr. 30, 2009] Our paper is accepted at Crypto 2009.
[June 2, 2009] A new single block chosen-prefix collision.
[June 3, 2009] Our Crypto 2009 paper has won the best paper award.
[June 16, 2009] Full paper now available: Marc Stevens, Arjen Lenstra and Benne de Weger, "Chosen-prefix Collisions for MD5 and Applications", submitted to the International Journal of Applied Cryptography.
[August 22, 2009] The Crypto paper and the best paper award.
[January 12, 2010] Our work made it to number 1 of the Top Ten Web Hacking Techniques of 2009.
[October 28, 2010] This site now enjoys translations into Korean and Belorussian, provided respectively by vangelis and movavi.

---

**Summary**

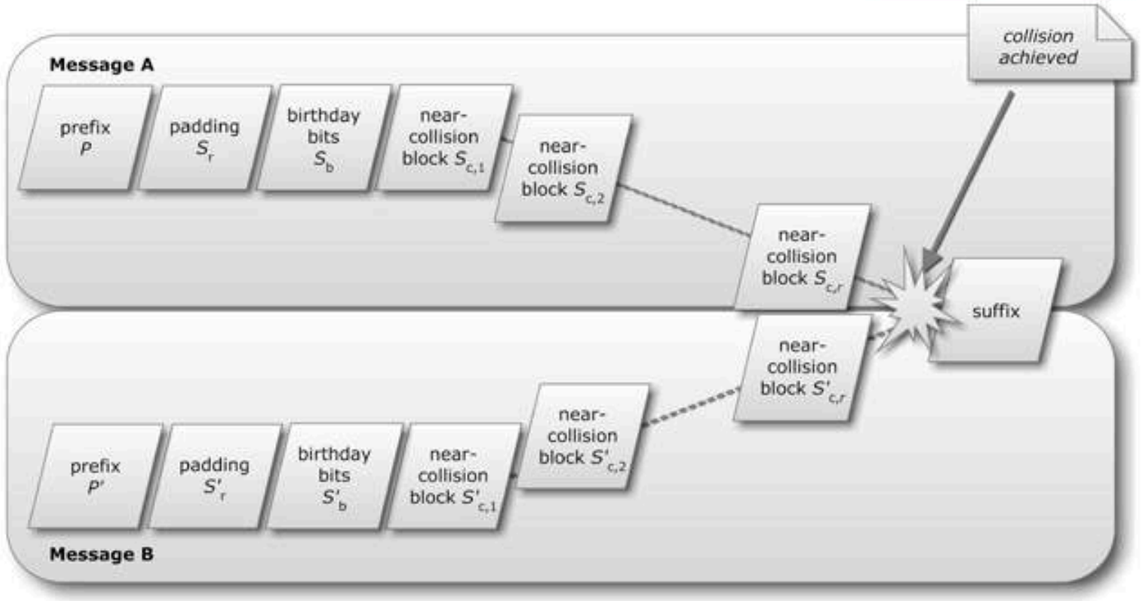
We have identified a vulnerability in the Internet Public Key Infrastructure (PKI) used to issue digital certificates for secure websites. As a proof of concept we executed a practical attack scenario and successfully created a rogue Certification Authority (CA) certificate trusted by all common web browsers. This certificate allows us to impersonate any website on the Internet, including banking and e-commerce sites secured using the HTTPS protocol.

# Crypto breakthrough shows Flame was designed by world-class scientists

The spy malware achieved an attack unlike any cryptographers have seen before.

by **Dan Goodin** - Jun 7, 2012 1:20pm CDT






Enlarge / An overview of a chosen-prefix collision. A similar technique was used by the Flame espionage malware that targeted Iran. The scientific novelty of the malware underscored the sophistication of malware sponsored by wealthy nation states.

Marc Stevens

The Flame espionage malware that infected computers

LATEST FEATURE STORY

FEATURE STORY (2 PAGES)

**The quest to save today's gaming history from being lost forever**

Changes in digital distribution, rights management increasingly make preservation tough.

WATCH ARS VIDEO


**Hands-on with the New LG G4**


UNITED STATES CYBER COMMAND

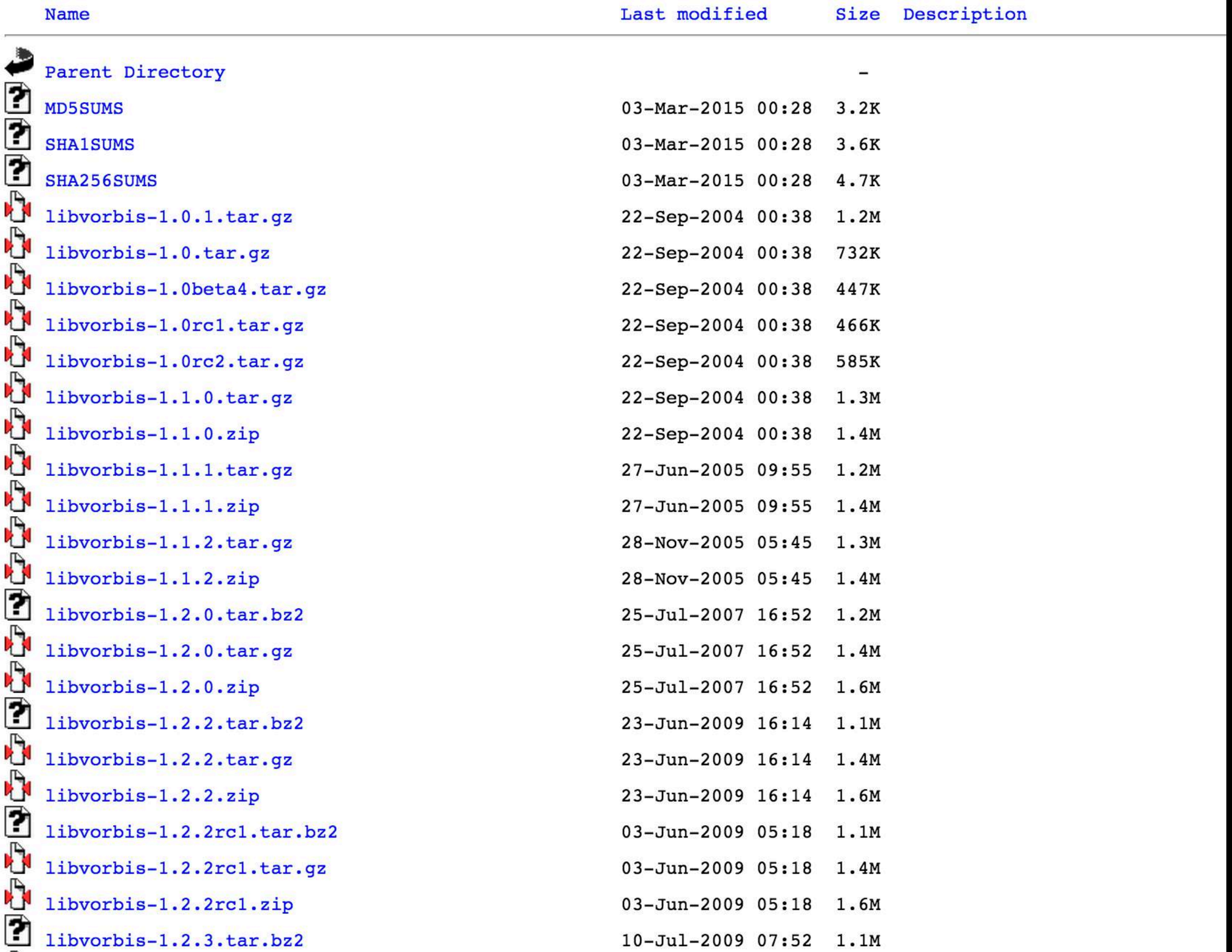
9EC4C12949A4F31474F299058CE2B22A

```
mission = """
USCYBERCOM plans, coordinates, integrates, synchronizes and conducts
activities to: direct the operations and defense of specified
Department of Defense information networks and; prepare to, and when
directed, conduct full spectrum military cyberspace operations in order
to enable actions in all domains, ensure US/Allied freedom of action
in cyberspace and deny the same to our adversaries.
"""

md5(mission)
# => 9EC4C12949A4F31474F299058CE2B22A
```

# Pitfalls

1. Using weak/old algorithms
2. **Misunderstanding checksums**
3. Length extension attacks



| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory | | - | |
| MD5SUMS | 03-Mar-2015 00:28 | 3.2K | |
| SHA1SUMS | 03-Mar-2015 00:28 | 3.6K | |
| SHA256SUMS | 03-Mar-2015 00:28 | 4.7K | |
| libvorbis-1.0.1.tar.gz | 22-Sep-2004 00:38 | 1.2M | |
| libvorbis-1.0.tar.gz | 22-Sep-2004 00:38 | 732K | |
| libvorbis-1.0beta4.tar.gz | 22-Sep-2004 00:38 | 447K | |
| libvorbis-1.0rc1.tar.gz | 22-Sep-2004 00:38 | 466K | |
| libvorbis-1.0rc2.tar.gz | 22-Sep-2004 00:38 | 585K | |
| libvorbis-1.1.0.tar.gz | 22-Sep-2004 00:38 | 1.3M | |
| libvorbis-1.1.0.zip | 22-Sep-2004 00:38 | 1.4M | |
| libvorbis-1.1.1.tar.gz | 27-Jun-2005 09:55 | 1.2M | |
| libvorbis-1.1.1.zip | 27-Jun-2005 09:55 | 1.4M | |
| libvorbis-1.1.2.tar.gz | 28-Nov-2005 05:45 | 1.3M | |
| libvorbis-1.1.2.zip | 28-Nov-2005 05:45 | 1.4M | |
| libvorbis-1.2.0.tar.bz2 | 25-Jul-2007 16:52 | 1.2M | |
| libvorbis-1.2.0.tar.gz | 25-Jul-2007 16:52 | 1.4M | |
| libvorbis-1.2.0.zip | 25-Jul-2007 16:52 | 1.6M | |
| libvorbis-1.2.2.tar.bz2 | 23-Jun-2009 16:14 | 1.1M | |
| libvorbis-1.2.2.tar.gz | 23-Jun-2009 16:14 | 1.4M | |
| libvorbis-1.2.2.zip | 23-Jun-2009 16:14 | 1.6M | |
| libvorbis-1.2.2rc1.tar.bz2 | 03-Jun-2009 05:18 | 1.1M | |
| libvorbis-1.2.2rc1.tar.gz | 03-Jun-2009 05:18 | 1.4M | |
| libvorbis-1.2.2rc1.zip | 03-Jun-2009 05:18 | 1.6M | |
| libvorbis-1.2.3.tar.bz2 | 10-Jul-2009 07:52 | 1.1M | |

# Pitfalls

1. Using weak/old algorithms
2. Misunderstanding checksums
3. **Length extension attacks**

# Length Extension Attacks

```
secret = "my-secret-key"
value = "buy 10 units at $1"
signature = sha256(secret + "|" + value)
```

# Length Extension Attacks

```
secret = "my-secret-key"
value = "buy 10 units at $1<garbage>actually make that at $0"
signature = sha256(secret + "|" + value)
```

# Length Extension Attacks

```
secret = "my-secret-key"
value = "buy 10 units at $1"
signature = hmac_sha256(secret, value)
```

# Message Authentication Code (MAC)

```
tag = hmac_sha256(key, value)
```

- `key` - shared secret
- `value` - value to protected integrity of
- `tag` - value that represents the integrity

# Flickr's API Signature Forgery Vulnerability

**Thai Duong and Juliano Rizzo**

Date Published: Sep. 28, 2009

Advisory ID: MOCB-01

Advisory URL: http://netifera.com/research/flickr_api_signature_forgery.pdf

Title: Flickr's API Signature Forgery Vulnerability

Remotely Exploitable: Yes

## *1. Vulnerability Description*

Flickr is almost certainly the best online photo management and sharing application in the world. As of June 2009, it claims to host more than 3.6 billion images. In order to allow independent programmers to expand its services, Flickr offers a fairly comprehensive web-service API that

# Recommendations

- Use SHA-256 (SHA-2 family)
- Choose HMAC-SHA-256 if you want a signature
- Stop using MD5
- Don't use SHA-1 in new projects


# Password Storage

RISK ASSESSMENT / SECURITY & HACKTIVISM

# 8 million leaked passwords connected to LinkedIn, dating website (updated)

An unknown hacker posted the lists online and asked for help in cracking them.

by **Dan Goodin** - June 6 2012, 12:05pm CDT

BLACK HAT THE WEB 132

0d2d32ea81418189eca21d1ff27fc65adb88fcd6:sm
873a5f2d901d579680fc5a5bd040ab241ac5d4a0:sa
0dde6e765f94b007f2ebed3b8fe3fcc84c7744bc:tu
e1abf2ee6113dae0b0d2ec8e8c6331b2a2308c18:st
33f059739de4286fcdd65482dc840069b62f94f9:th
dd0ea828e93ab88988691037e442c9e0d1baa6d1:sa
82ccd756877b247c989380d758c4a02bd7cccd2f:kn
2688b21ce3822ed3c923d8eb5e3454f7e4b7b2b5:al
d9ba61eef61ed406551cd9b37dee351d2d31866f:al
7f4d8f4a4128faf2f0b35b3f39a9e940310463c6:ro
1bd32f0d7301f3494050f2452faefde13b319e04:Na
e644c8ea288aead799f04e01bd01b739437052b9:es
6b6dc44810694d4cd41283b5c300a47656688462:nf
7e379328f307b5b33ff8364e453a54f9b2b9f101:thisisnotsecure
872d6b8e5de06c62ecd24d1bc6f0f6a6e35950e2:1loveMYson

A partial list of the 6.5 million passwords leaked by someone identified as dwdm. The list contains strong passwords that were unique to LinkedIn, leading to speculation that's where the passwords originated.

Dan Goodin, Ars Technica

An unknown hacker has posted more than 8 million cryptographic hashes to the Internet that appear to belong to users of LinkedIn and a separate, popular dating website.

CNET › News › Security & Privacy

# Last.fm warns users of password leak

Just like LinkedIn and eHarmony yesterday, Last.fm warns users to change their passwords following a security breach.

by Elinor Mills | June 7, 2012 10:25 AM PDT

Follow

Last.fm today urged its users to change their passwords because of a compromise that may be related to a huge password leak involving LinkedIn and eHarmony.

"We are currently investigating the leak of some Last.fm user passwords. This follows recent password leaks on other sites, as well as information posted online," a Last.fm blog post said. "As a precautionary measure, we're asking all our users to change their passwords immediately."

The blog post did not say how many users were affected or how the passwords were leaked. A Last.fm executive did not immediately respond to an e-mail and a phone message seeking comment.

Last.fm users should log in to the site and change their passwords on the settings page. The music site said it will never e-mail a direct link to update settings or ask for passwords, so if users receive such e-mails they are spam.

Yesterday, LinkedIn and eHarmony came out with similar warnings about password compromises. But the statements lacked vital details. And there are likely to be more warnings as companies discover that users' passwords are among a huge list of 6.5 million and at least one other list that were posted to a Russian hacker site earlier this week.

# Yahoo hacked, 450,000 passwords posted online

By **Doug Gross**, CNN
updated 9:31 AM EDT, Fri July 13, 2012 | Filed under: **Web**


Login information of more than 450,000 Yahoo users was hacked and posted online in a warning to the site.

**STORY HIGHLIGHTS**

- Hacker posts 450,000 Yahoo Voice account passwords
- Web page says the hack was meant as a warning to Yahoo
- Some users had painfully easy-to-crack passwords like "12345" or "password"
- **NEW:** Hack impacted users of Yahoo Voices, a Web news product

**(CNN)** -- Hackers posted online what they say is login information for more than 450,000 Yahoo users.

The hack, which of course was conducted anonymously, was meant to be a warning, according to the Web page where the documents were dumped.

"We hope that the parties responsible for managing the security of this subdomain will take this as a wake-up call, and not as a threat," a note on the page said. "There have been many security holes

# Dropbox confirms it was hacked, assures it's safe

Some account names and passwords were stolen from the file transferring company

By Joe Svetlik August 1st

0 COMMENTS

Like 78 Send Tweet 318 +1 5 Submit


Dropped the ball on this one

Dropbox **has confirmed** some account names and passwords have been stolen.

It was alerted after users started complaining about spam they were receiving to email addresses used only for their Dropbox accounts. It started an investigation, and found

**Related stories**

**More passwords compromised as Nvidia, Android Forums hacked**

# 21 eBay Urges Password Changes After Breach

MAY 14

**eBay** is asking users to pick new passwords following a data breach earlier this year that exposed the personal information of an untold number of the auction giant's 145 million customers.


In a blog post published this morning, eBay said it had "no evidence of the compromise resulting in unauthorized activity for eBay users, and no evidence of any unauthorized access to financial or credit card information, which is stored separately in encrypted formats. However, changing passwords is a best practice and will help enhance security for eBay users."

Assisted by federal investigators, eBay determined that the intrusion happened in late February and early march, after a "small number of employee log-in credentials" that allowed attackers access to eBay's corporate network were compromised. The company said *the information compromised included eBay customers' name, encrypted password, email address, physical address, phone number and date of birth*. eBay also said it has no evidence of unauthorized access or compromises to personal or financial information for **PayPal** users.

4:09 pm ET
Mar 27, 2015 BIG DATA

# Slack Discloses Breach Amid $160 Million Fundraise

ARTICLE COMMENTS

CYBERSECURITY DATA BREACH HIPCHAT SLACK STEWART BUTTERFIELD

Email Print

By DANNY YADRON and DOUGLAS MACMILLAN CONNECT

Stewart Butterfield, co-founder and chief executive officer of Slack. — Bloomberg News

Slack, maker of an eponymous office-communication tool that seeks to replace corporate email, said Friday that hackers had accessed user data that may include messages sent between users.

News of the breach comes as new investors **recently agreed** to give the much-hyped app $160 million in additional venture funding at a valuation of $2.76 billion. The company's value has risen remarkably fast, even by Silicon Valley standards, after launching a little more than a year ago.

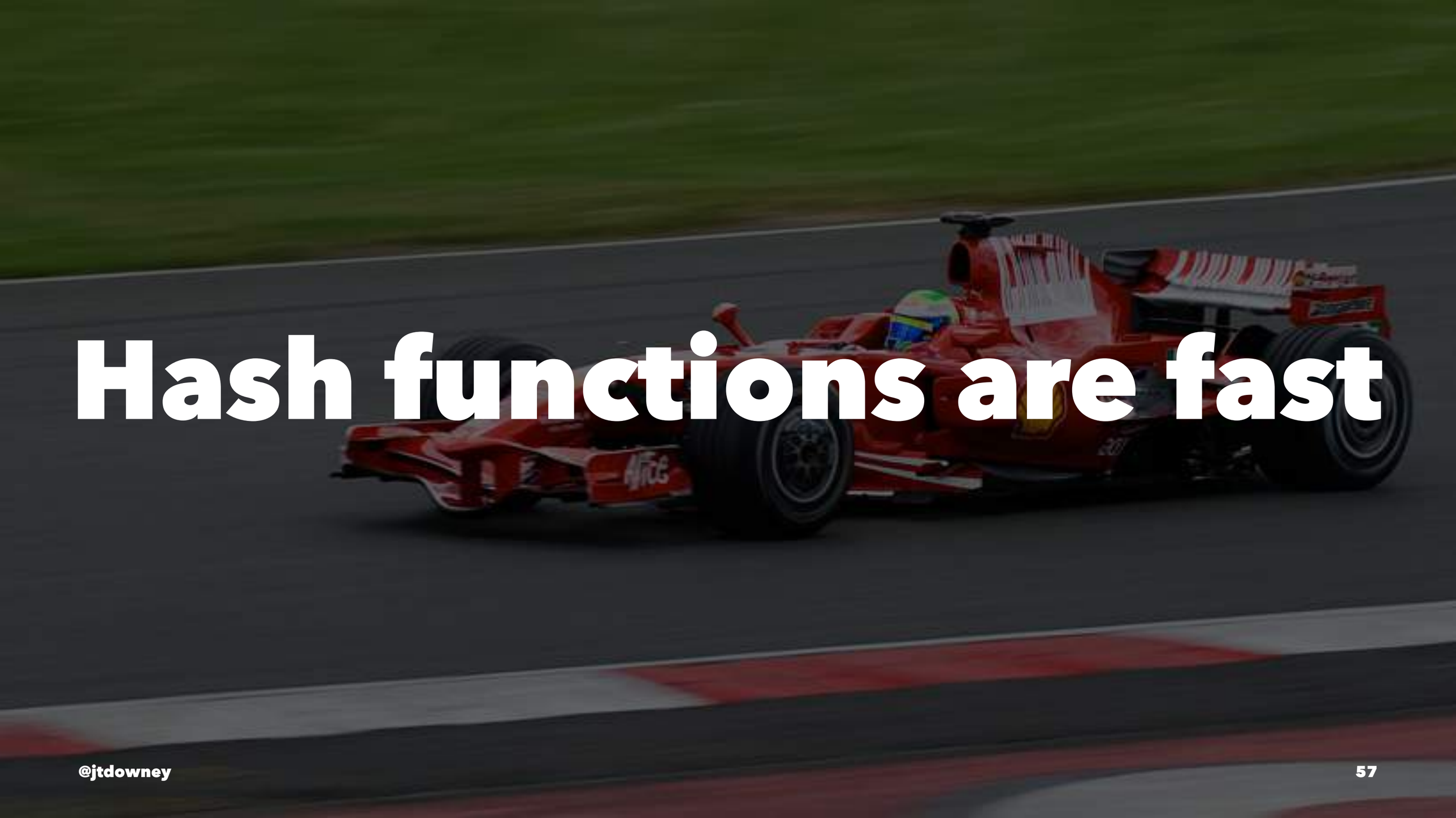
It's unclear when Slack discovered the breach or if new investors were told of it before they agreed to the deal. Paperwork for the fundraising round was signed recently, The Wall Street Journal reported Thursday.

```
sha1(password)
```

1. One-way
    - Value can be used for verification

```
sha1(salt + password)
```

1. One-way
    - Value can be used for verification
2. Randomized
    - Can largely defeat pre-computed tables
    - Forces attackers to focus on one password


Hash functions are fast

1. One-way
    - Value can be used for verification
2. Randomized
    - Can largely defeat pre-computed tables
    - Forces attackers to focus on one password
3. Slow

# Adaptive Hashing

## bcrypt, scrypt, or PBKDF2

# Recommendations

- Delegate authentication if possible
  - Facebook, Twitter, Google, Github
- Store one-way verifiers using bcrypt, scrypt, or PBDKF2


Block Ciphers

# Pitfalls

1. **Using old/weak algorithms**
2. Using ECB mode
3. Not using authenticated encryption

# Breaking Ciphers with COPACOBANA - A Cost-Optimized Parallel Code Breaker

Sandeep Kumar[1], Christof Paar[1], Jan Pelzl[1], Gerd Pfeiffer[2], and Manfred Schimmler[2]

[1] Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany
{kumar,cpaar,pelzl}@crypto.rub.de
[2] Institute of Computer Science and Applied Mathematics, Faculty of Engineering,
Christian-Albrechts-University of Kiel, Germany
{gp,masch}@informatik.uni-kiel.de

**Abstract.** Cryptanalysis of symmetric and asymmetric ciphers is computationally extremely demanding. Since the security parameters (in particular the key length) of almost all practical crypto algorithms are chosen such that attacks with conventional computers are computationally infeasible, the only promising way to tackle existing ciphers (assuming no mathematical breakthrough) is to build special-purpose hardware. Dedicating those machines to the task of cryptanalysis holds the promise of a dramatically improved cost-performance ratio so that breaking of commercial ciphers comes within reach.

This contribution presents the design and realization of the COPACOBANA (Cost-Optimized Parallel Code Breaker) machine, which is optimized for running cryptanalytical algorithms and can be realized for less than US$ 10,000. It will be shown that, depending on the actual algorithm, the architecture can outperform conventional computers by several orders in magnitude. COPACOBANA hosts 120 low-cost FPGAs and is able to, e.g., perform an exhaustive key search of the Data Encryption Standard (DES) in less than nine days on average. As a real-world application,

# Pitfalls

1. Using old/weak algorithms
2. **Using ECB mode**
3. Not using authenticated encryption

# AES - primitive

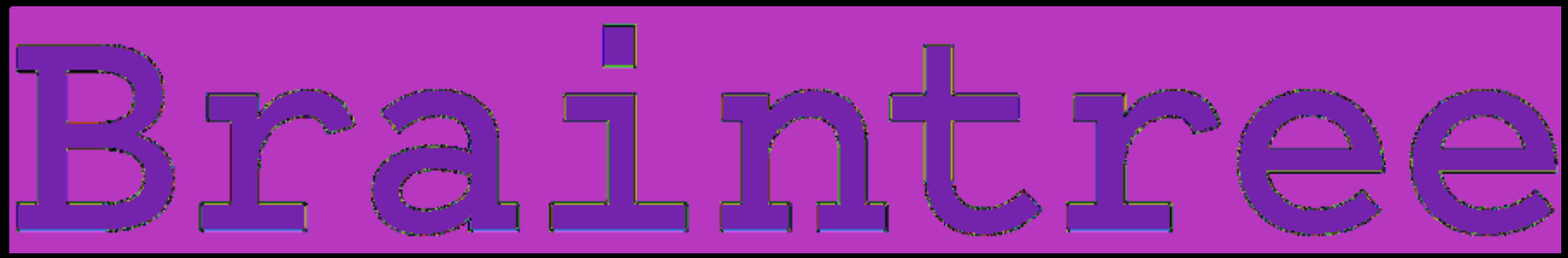
```
ciphertext = AES_Encrypt(key, plaintext)
plaintext  = AES_Decrypt(key, ciphertext)
```

- Function over:
  - `key` - 128, 192, or 256 bit value
  - `plaintext` - 128 bit value
  - `ciphertext` - 128 bit value

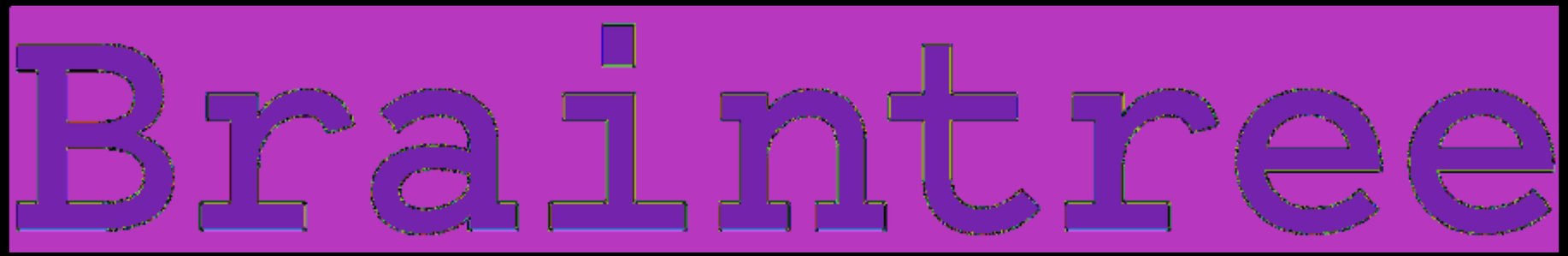
# ECB Encrypt

```
while (remaining blocks) {
  block = ... # next 64 byte (128 bit chunk)
  ouput.append(AES_Encrypt(key, block))
}
```

# Braintree



Braintree




# Pitfalls

1. Using old/weak algorithms
2. Using ECB mode
3. **Not using authenticated encryption**

# Practical Padding Oracle Attacks

Juliano Rizzo* Thai Duong†

May 25th, 2010

## Abstract

At Eurocrypt 2002, Vaudenay introduced a powerful side-channel attack, which is called padding oracle attack, against CBC-mode encryption with PKCS#5 padding (See [6]). If there is an oracle which on receipt of a ciphertext, decrypts it and then replies to the sender whether the padding is correct or not, Vaudenay shows how to use that oracle to efficiently decrypt data without knowing the encryption key. In this paper, we turn the padding oracle attack into a new set of practical web hacking techniques. We also introduce a new technique that allows attackers to use a padding oracle to encrypt messages of any length without knowing the secret key. Finally, we show how to use that technique to mount advanced padding oracle exploits against popular web development frameworks.

explained in Paterson and Yau's summary in [5], the padding oracle attack requires an oracle which on receipt of a ciphertext, decrypts it and replies to the sender whether the padding is `VALID` or `INVALID`. The attack works under the assumption that the attackers can intercept padded messages encrypted in CBC mode, and have access to the aforementioned padding oracle. The result is that attackers can recover the plaintext corresponding to any block of ciphertext using an average of $128 \star b$ oracle calls, where $b$ is the number of bytes in a block. The easiest fix for the padding oracle attack is to encrypt-then-MAC, i.e., encrypting information to get the ciphertext, then protecting the ciphertext integrity with a Message Authentication Code scheme. For more details on Vaudenay's attack and suggested fixes, please see [7, 1, 3, 4, 5].

Table of Contents

rything

lules
pto
pto.Cipher
pto.Cipher.AES
pto.Cipher.ARC2
pto.Cipher.ARC4
pto.Cipher.Blowfish
pto.Cipher.CAST
pto.Cipher.DES
pto.Cipher.DES3
pto.Cipher.PKCS1_OAEP
pto.Cipher.PKCS1_v1_5

Everything

Classes
pto.Cipher.AES.AESCipher
pto.Cipher.ARC2.RC2Cipher
pto.Cipher.ARC4.ARC4Cipher
pto.Cipher.Blowfish.BlowfishCipher
pto.Cipher.CAST.CAST128Cipher
pto.Cipher.DES.DESCipher
pto.Cipher.DES3.DES3Cipher
pto.Cipher.PKCS1_OAEP.PKCS1OAEP
pto.Cipher.PKCS1_v1_5.PKCS115_Ciph
pto.Cipher.XOR.XORCipher
pto.Cipher.blockalgo.BlockAlgo
pto.Hash.HMAC.HMAC
pto.Hash.MD2.MD2Hash
pto.Hash.MD4.MD4Hash
pto.Hash.MD5.MD5Hash
pto.Hash.RIPEMD.RIPEMD160Hash
pto.Hash.SHA.SHA1Hash
pto.Hash.SHA224.SHA224Hash
pto.Hash.SHA256.SHA256Hash
pto.Hash.SHA384.SHA384Hash
pto.Hash.SHA512.SHA512Hash
pto.Protocol.AllOrNothing.AllOrNothing
pto.Protocol.Chaffing.Chaff
pto.PublicKey.DSA.DSAImplementation
pto.PublicKey.DSA.DSAobj
pto.PublicKey.ElGamal.ElGamalobj
pto.PublicKey.ElGamal.error

Home Trees Indices Help PyCrypto.org

Package Crypto :: Package Cipher :: Module AES [frames] | no frames

# Module AES

AES symmetric cipher

AES (Advanced Encryption Standard) is a symmetric block cipher standardized by NIST . It has a fixed data block size of 16 bytes. Its keys can be 128, 192, or 256 bits long.

AES is very fast and secure, and it is the de facto standard for symmetric encryption.

As an example, encryption can be done as follows:

```
>>> from Crypto.Cipher import AES
>>> from Crypto import Random
>>>
>>> key = b'Sixteen byte key'
>>> iv = Random.new().read(AES.block_size)
>>> cipher = AES.new(key, AES.MODE_CFB, iv)
>>> msg = iv + cipher.encrypt(b'Attack at dawn')
```

## Classes

| | |
|---|---|
| | **AESCipher** |

## Functions

| | |
|---|---|
| | **new**(key, *args, **kwargs)<br>Create a new AES cipher |

## Variables

| | |
|---|---|
| | **MODE_ECB** = 1<br>Electronic Code Book (ECB). See blockalgo.MODE_ECB. |
| | **MODE_CBC** = 2<br>Cipher-Block Chaining (CBC). See blockalgo.MODE_CBC. |
| | **MODE_CFB** = 3<br>Cipher FeedBack (CFB). See blockalgo.MODE_CFB. |
| | **MODE_PGP** = 4<br>This mode should not be used. |
| | **MODE_OFB** = 5<br>Output FeedBack (OFB). See blockalgo.MODE_OFB. |
| | **MODE_CTR** = 6<br>CounTer Mode (CTR). See blockalgo.MODE_CTR. |
| | **MODE_OPENPGP** = 7<br>OpenPGP Mode. See blockalgo.MODE_OPENPGP. |
| | **block_size** = 16<br>Size of a data block (in bytes) |

# World of hurt

# Recommendations

- Prefer to use box/secret box from NaCL/libsodium
- Stop using DES
- Stop building your own on top of AES

# What if you have to use AES

- Do not use ECB mode
- Be sure to use authenticated encryption:
  - GCM mode would be a good first choice
- Verify the tag/MAC first
- Still easy to mess up in a critical way

# TLS/SSL Verification

# The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software

Martin Georgiev
The University of Texas at Austin

Subodh Iyengar
Stanford University

Suman Jana
The University of Texas at Austin

Rishita Anubhai
Stanford University

Dan Boneh
Stanford University

Vitaly Shmatikov
The University of Texas at Austin

## ABSTRACT

SSL (Secure Sockets Layer) is the de facto standard for secure Internet communications. Security of SSL connections against an active network attacker depends on correctly validating public-key certificates presented when the connection is established.

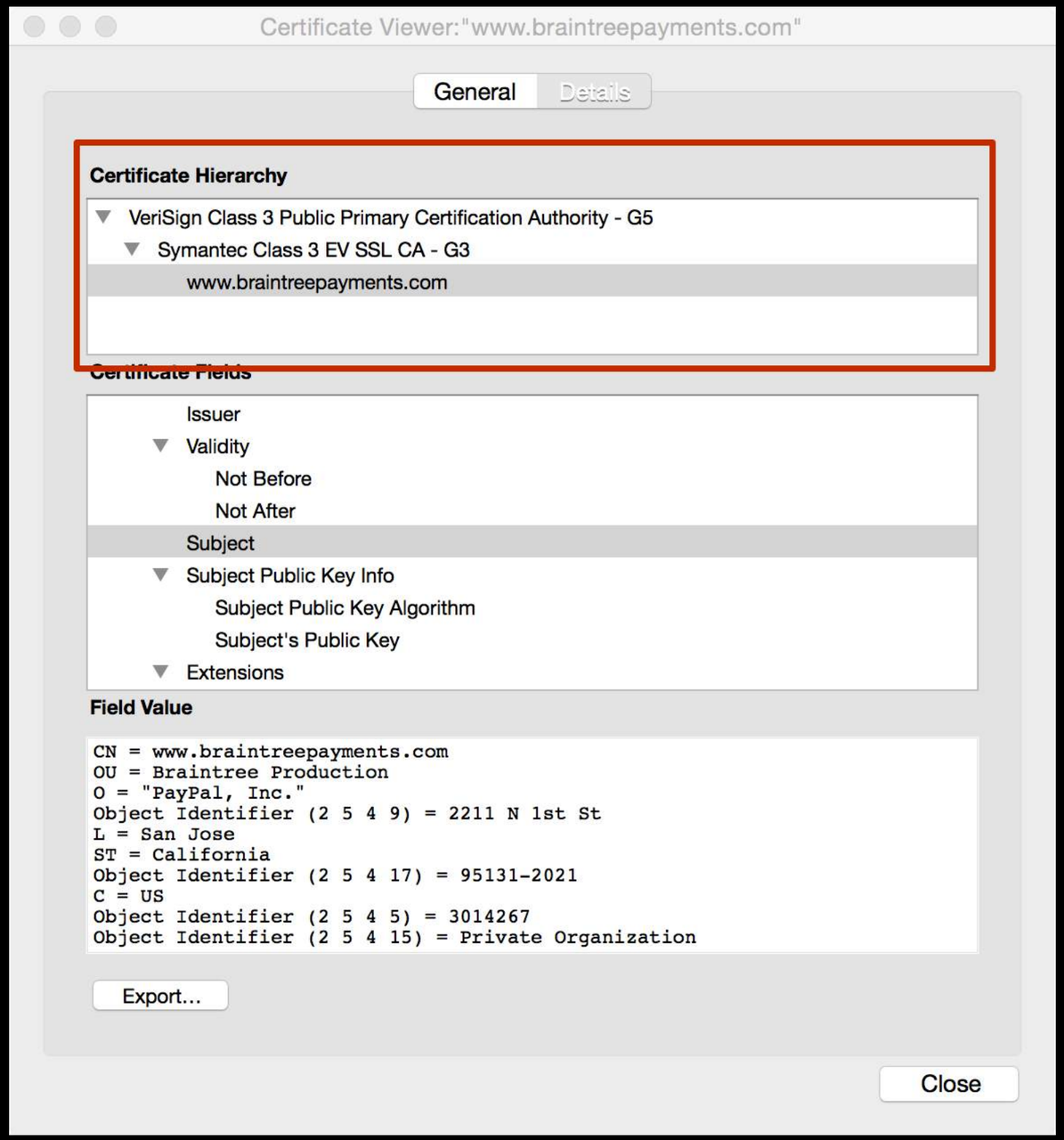
We demonstrate that SSL certificate validation is completely broken in many security-critical applications and libraries. Vulnerable software includes Amazon's EC2 Java library and all cloud clients based on it; Amazon's and PayPal's merchant SDKs responsible for transmitting payment details from e-commerce sites to payment gateways; integrated shopping carts such as osCommerce, ZenCart, Ubercart, and PrestaShop; AdMob code used by mobile websites; Chase mobile banking and several other Android apps and libraries;

cations. The main purpose of SSL is to provide end-to-end security against an active, man-in-the-middle attacker. Even if the network is completely compromised—DNS is poisoned, access points and routers are controlled by the adversary, etc.—SSL is intended to guarantee confidentiality, authenticity, and integrity for communications between the client and the server.

Authenticating the server is a critical part of SSL connection establishment.[1] This authentication takes place during the SSL handshake, when the server presents its public-key certificate. In order for the SSL connection to be secure, the client must carefully verify that the certificate has been issued by a valid certificate authority, has not expired (or been revoked), the name(s) listed in the certificate match(es) the name of the domain that the client is connecting to, and perform several other checks [14, 15].

# Pitfalls

1. **Not verifying the certificate chain**
2. Not verifying the hostname
3. Using a broken library


Certificate Viewer:"www.braintreepayments.com"
General
Details
Certificate Hierarchy
VeriSign Class 3 Public Primary Certification Authority - G5
Symantec Class 3 EV SSL CA - G3
www.braintreepayments.com
Certificate Fields
Issuer
Validity
Not Before
Not After
Subject
Subject Public Key Info
Subject Public Key Algorithm
Subject's Public Key
Extensions
Field Value
CN = www.braintreepayments.com
OU = Braintree Production
O = "PayPal, Inc."
Object Identifier (2 5 4 9) = 2211 N 1st St
L = San Jose
ST = California
Object Identifier (2 5 4 17) = 95131-2021
C = US
Object Identifier (2 5 4 5) = 3014267
Object Identifier (2 5 4 15) = Private Organization
Export...
Close

```
$ curl -k https://example.com
```

or

```
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
```

# Pitfalls

1. Not verifying the certificate chain
2. **Not verifying the hostname**
3. Using a broken library

- Hostname verification is protocol dependent
  - OpenSSL doesn't have it built in
- Also, some people just turn it off:

```
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
```

# Pitfalls

1. Not verifying the certificate chain
2. Not verifying the hostname
3. **Using a broken library**

# The Heartbleed Bug

The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet. SSL/TLS provides communication security and privacy over the Internet for applications such as web, email, instant messaging (IM) and some virtual private networks (VPNs).

The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.

## What leaks in practice?

We have tested some of our own services from attacker's perspective. We attacked ourselves from outside, without leaving a trace. Without using any privileged information or credentials we were able steal from ourselves the secret keys used for our X.509 certificates, user names and passwords, instant messages, emails and business critical documents and

## How to stop the leak?

As long as the vulnerable version of OpenSSL is in use it can be abused. Fixed OpenSSL has been released and now it has to be deployed. Operating system vendors and distribution, appliance vendors, independent software vendors have to adopt the fix and notify their users. Service providers and users have to install the fix as it becomes available for the

# ImperialViolet

## Apple's SSL/TLS bug (22 Feb 2014)

Yesterday, Apple pushed a rather spooky security update for iOS that suggested that something was horribly wrong with SSL/TLS in iOS but gave no details. Since the answer is at the top of the Hacker News thread, I guess the cat's out of the bag already and we're into the misinformation-quashing stage now.

So here's the Apple bug:

```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                 uint8_t *signature, UInt16 signatureLen)
{
OSStatus        err;
...

if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
```

# Recommendations

- Do ensure you're validating connections
- Lean on a framework/library if possible
  - But check that it also does the right thing
- Setup and automated test to validate this setting

# Trust

```
The authenticity of host 'apollo.local (10.0.2.56)' can't be established.
RSA key fingerprint is 04:63:c1:ba:c7:31:04:12:14:ff:b6:c4:32:cf:44:ec.
Are you sure you want to continue connecting (yes/no)?
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
04:63:c1:ba:c7:31:04:12:14:ff:b6:c4:32:cf:44:ec.
Please contact your system administrator.
```

AOL Time Warner Inc.
AS Sertifitseerimiskeskus
AddTrust
Baltimore
beTRUSTed
Buypass
CNNIC
COMODO CA Limited
Certplus
certSIGN
Chambersign
Chunghwa Telecom Co., Ltd.
ComSign
Comodo CA Limited
Cybertrust, Inc
Deutsche Telekom AG
Deutscher Sparkassen Verlag GmbH
Dhimyotis
DigiCert Inc
DigiNotar
Digital Signature Trust Co.
Disig a.s.
EBG Bilişim Teknolojileri ve Hizmetleri A.Ş.
EDICOM
Entrust, Inc.
Equifax
GTE Corporation
GeoTrust Inc.
GlobalSign nv-sa
Hongkong Post
Japan Certification Services, Inc.
Japanese Government
Microsec Ltd.
NetLock Halozatbiztonsagi Kft.
Network Solutions L.L.C.
PM/SGDN
QuoVadis Limited
RSA Security Inc
SECOM Trust Systems CO.,LTD.
SecureTrust Corporation
Sociedad Cameral de Certificación Digital
Sonera
Staat der Nederlanden
Starfield Technologies, Inc.
StartCom Ltd.
SwissSign AG
Swisscom
TC TrustCenter GmbH
TDC
Taiwan Government
Thawte
The Go Daddy Group, Inc.
The USERTRUST Network
TÜBİTAK
TÜRKTRUST
Unizeto Sp. z o.o.
VISA
ValiCert, Inc.
VeriSign, Inc.
WISeKey
Wells Fargo
XRamp Security Services Inc

# Certificate Pinning

# Recommendations

- Think about what organizations you really trust
- Investigate certificate pinning for your apps

# Stanford Crypto Class

https://www.coursera.org/course/crypto

# Matasano Crypto Challenges

## http://cryptopals.com

# Questions

# Images

- https://flic.kr/p/4KWhKn
- https://flic.kr/p/9F2BCv
- https://flic.kr/p/486xYS
- https://flic.kr/p/7Ffppm
- https://flic.kr/p/8TuJD9
- https://flic.kr/p/4iLJZt
- https://flic.kr/p/4pGZuz
- https://flic.kr/p/8aZWNE
- https://flic.kr/p/5NRHp
- https://flic.kr/p/7p7raq
- https://flic.kr/p/aZEE1Z
- https://flic.kr/p/7WtwAz
- https://flic.kr/p/6AN9mM
- https://flic.kr/p/6dt62u
- https://flic.kr/p/4ZqwyB
- https://flic.kr/p/Bqewr
- https://flic.kr/p/ecdhVE