The Open Crypto Audit Project: Our Story

Kenneth White & Matthew Green
Open Crypto Audit Project

Everyone has a story. This is ours.
Agenda

• First Principles
• Post-Snowden Era
• The TrueCrypt Story
• Open Crypto Audit Project
• Secure Coding & Trust
• Looking Ahead
• Open Discussion (and swag!)
About Us
Kenneth White

- Interests: RT signals, embedded systems, analytics
- First DEFCON: DC10
- Formal training: bio-signals (EEG/ERP, MRI, PET, EKG, EOG)
- Early career: databases, *nix, RTOS, h/w drivers
- Lifecycle: FDA (cardiac safety), SEI SEPG, IA
- Defense: network security, API endpoints
- Recently: public cloud security, ML/classification, safety-critical systems, breaking crypto/networks/websites/OS'
- Now: OCAP, Linux Foundation CII, NGO security
- @kennwhite
I like to work on interesting problems
Matthew Green

- Johns Hopkins University: Computer Science
- Teaches applied cryptography
- Builds secure systems
- Trained under Susan Hohenberger & Avi Rubin
- Former senior research staff: AT&T Labs
- On-going Research includes:
  - Techniques for privacy-enhanced information storage
  - Anonymous payment systems (including ZeroCoin)
  - Bilinear map-based cryptography
- @matthew_d_green
Matthew Green

(not his actual Dachshunds)
Long journey to DEFCON (no, really)

(my actual Shepherds, semi-medicated)
“I’m here to share what I know, and learn with and from you.”

— Jack Daniel
First Principles

“If a bad guy can persuade you to run his program on your computer, it's not your computer anymore.”
— Scott Culp
First Principles

“If a bad guy can persuade you to run his program on your computer, it’s not your computer anymore.”
— Scott Culp

“Even if it has disk encryption.”
— Kenn White
Crypto 101: First Principles

Thompson: Reflections on Trusting Trust
cm.bell-labs.com/who/ken/trust.html

Culp: 10 Immutable Laws of Security
technet.microsoft.com/library/cc722487

Zimmerman: Beware of Snake Oil
www.philzimmermann.com/EN/essays/SnakeOil
Post-Snowden Era

- NYT, Propublica, Guardian: NSA spends $250M/yr to counter & undermine “the use of ubiquitous encryption across the internet”
- NIST technical standards “intentionally weakened”
- BULLRUN: NSA actively working to "Insert vulnerabilities into commercial encryption systems, IT systems, networks, and endpoint communications devices used by targets” The New York Times, 2013/09/05

See: www.eff.org/nsa-spying/timeline
“Furthermore, we will be reviewing our existing body of cryptographic work”
— National Institute of Standards and Technology, Nov 2013

Recommends that the US government “fully support and not undermine efforts to create encryption standards”
— Presidential Advisory Committee, Jan 2014

“[C]lassified [reports] have heightened concern over the possibility of a backdoor... after conducting its own review, NIST [has] removed DRBG”
— National Institute of Standards and Technology, Apr 2014
Which bring us to TrueCrypt
TrueCrypt

- File, volume, full disk encryption (FDE)
- 30M+ downloads
- Created Feb 2004 by anonymous development team
- Controversial license (Debian, Fedora, “forbidden items”)

DEF CON 22 | 2014.08.08
TrueCrypt

• Tool of choice for human rights workers, activists, attorneys, thousands of organizations, investigative/national security journalists, security professionals, and...?
AWS Import/Export
Developer Guide (API Version 2010-06-03)

Using TrueCrypt Encryption

For added security, AWS Import/Export supports data encryption using TrueCrypt for import to Amazon S3 and export from Amazon S3. TrueCrypt is an open-source disk encryption application.

TrueCrypt is the only device encryption supported by AWS Import/Export. For information about how to download, install, and use TrueCrypt, go to www.truecrypt.org.

For import to Amazon S3, you can use TrueCrypt to encrypt your data before sending it to AWS Import/Export. You will need to include your TrueCrypt password in your import manifest.

For import to Amazon EBS or Amazon Glacier, you can use any encryption method you choose. AWS does not decrypt your data for import to Amazon EBS or Amazon Glacier. We strongly encourage you to encrypt your data.

For export from Amazon S3, AWS always encrypts your data using TrueCrypt with the TrueCrypt password in your export manifest.

The following sections detail the encryption process for import to Amazon S3 and export from Amazon S3.

Encryption for Import to Amazon S3

Follow the instructions in the TrueCrypt documentation to create a new TrueCrypt volume. AWS Import/Export supports only TrueCrypt volumes created as non-system partitions or encrypted file containers. Do not use the Encrypt the system partition or the entire system drive option.

To ensure that we can decrypt your device, choose the following options when creating a TrueCrypt volume:

- Select either the Create an encrypted file container option or the Encrypt a non-system partition/drive option.

TrueCrypt

- Never thoroughly audited on Windows
- Differences reported in volume headers
- Small differences in distributed binaries vs. source
- Windows vs. Mac & Linux
- With exception of deniability volume, no formal cryptanalysis
- Deterministic build? (Xavier de Carné de Carnavalet)
- Last license review in 2008 by RedHat/Fedora/OSSI concluded “we would not be protected from a lawsuit” and “this license is non-free”
By many measures, relatively strong*

<table>
<thead>
<tr>
<th>Algorithm/Setting</th>
<th>Hashes/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>descrypt, DES(Unix), Traditional DES</td>
<td>952,300,000</td>
</tr>
<tr>
<td>SHA512</td>
<td>797,417,300</td>
</tr>
<tr>
<td>MS-SQL 2012</td>
<td>770,212,200</td>
</tr>
<tr>
<td>OSX v10.7</td>
<td>743,689,500</td>
</tr>
<tr>
<td>HMAC-SHA512 (key = $salt)</td>
<td>371,098,600</td>
</tr>
<tr>
<td>Whirlpool</td>
<td>363,711,900</td>
</tr>
<tr>
<td>HMAC-SHA512 (key = $pass)</td>
<td>194,800,700</td>
</tr>
<tr>
<td>Kerberos 5 AS-REQ Pre-Auth etype 23</td>
<td>120,452,800</td>
</tr>
<tr>
<td>AIX SHA1</td>
<td>48,099,200</td>
</tr>
<tr>
<td>md5crypt, MD5(Unix), FreeBSD MD5, Cisco IOS MD5</td>
<td>26,699,500</td>
</tr>
<tr>
<td>phpass, MD5(Wordpress), MD5(phpBB3)</td>
<td>13,736,900</td>
</tr>
<tr>
<td>1Password</td>
<td>9,249,000</td>
</tr>
<tr>
<td>TrueCrypt 5.0+ PBKDF2-HMAC-RipeMD160 boot-mode + AES</td>
<td>7,071,300</td>
</tr>
<tr>
<td>Samsung Android PIN</td>
<td>5,068,800</td>
</tr>
<tr>
<td>TrueCrypt 5.0+ PBKDF2-HMAC-RipeMD160 + AES</td>
<td>4,008,400</td>
</tr>
<tr>
<td>Password Safe</td>
<td>2,570,800</td>
</tr>
<tr>
<td>WPA/WPA2</td>
<td>1,304,800</td>
</tr>
<tr>
<td>TrueCrypt 5.0+ PBKDF2-HMAC-SHA512 + AES</td>
<td>310,323</td>
</tr>
<tr>
<td>sha256crypt, SHA256(Unix)</td>
<td>184,887</td>
</tr>
<tr>
<td>sha512crypt, SHA512(Unix)</td>
<td>98,285</td>
</tr>
<tr>
<td>bcrypt, Blowfish(OpenBSD)</td>
<td>36,141</td>
</tr>
<tr>
<td>GRUB 2</td>
<td>19,286</td>
</tr>
<tr>
<td>TrueCrypt 5.0+ PBKDF2-HMAC-Whirlpool + AES</td>
<td>17,036</td>
</tr>
<tr>
<td>OSX v10.8</td>
<td>5,571</td>
</tr>
</tbody>
</table>

*Hashes/sec on Sagitta Brutalis 290X: oclHashcat 1.00, AMD Catalyst 13.12
Accelerator: 8 x AMD Radeon R9 290X, stock clocks. Benchmark: Incremental brute force, alphanumcharset
Anonymous Dev Team

The information is out there

• Follow the money
• Follow the attorneys
• What we can share
• What we won’t share
Public Record

- State of Nevada Corporate Records
- US Trademark Office
- International Trademark Filings (UK, France, China, Russia, Czech Republic)
- Public IRS filings
- Usenet/mailing list forums
- Published academic papers
- Student theses
Public Record

Some things we chose not to share.
Why?
Remember this doxing?
Let’s not forget this:
And this:

Bitcoin Creator Returns To Internet To Say, 'I Am Not Dorian Nakamoto'
And, crucially, this:

ARThUR NAKAMOTO: Newsweek Reporter 'Is Destroying My Eldest Brother'

Newsweek is standing by Leah McGrath Goodman's assertion that Dorian Prentice Satoshi Nakamoto invented Bitcoin.

Nakamoto is denying it.

One source for McGrath Goodman's piece was Dorian's brother, Arthur.

In a very brief phone conversation with Business Insider Friday, Arthur Nakamoto indicated he'd been misquoted or misinterpreted.
Back to the Code
Conventional Wisdom: Given enough eyeballs, all bugs are shallow.
Meet Samuel Reshevsky, age 8 defeating 14 French chess masters at once, 1920
And so, it began...

AUDIT ALL THE THINGS!
The TrueCrypt Audit

- IsTrueCryptAuditedYet.com: Sept 24, 2013
- Announced on Twitter
- First contributions: Matthew & Me
- FundFill site set up
#istruecryptauditedyet? Our first pledge of $100 comes from @kennwhite towards getting TrueCrypt audited!
fundfill.com/fund/TrueCrypt...
The TrueCrypt Audit

Oct 9, 2014

• Prof. Green blogs about it
• Front page Hacker News
Why, hello there!
And so it went...

• No, we don’t take Bitcoin.
• Yes, we take Bitcoin.
• Yes, the site is mobile-friendly.
• No, we don’t take PayPal.
• /sets up IndieGoGo site.
• Yes! We take PayPal.
And so on...

“Hi, I’d like to buy 500 t-shirts, please.”

“Do you ship to Thailand?”

Where does one purchase 150 DVDs of Sneakers?
Incredible community

The TrueCrypt Audit
People, businesses, and governments all over the world use TrueCrypt to protect their privacy. We need help making it better and more secure.

Technology – Research Triangle, North Carolina, United States

Campaign Home | Updates / 0 | Comments / 0 | Funders / 190

$8,154
Raised of $25,000 Goal

59 days left

Flexible Funding
This campaign will receive all funds raised even if it does not reach its goal. Funding duration: October 14, 2013 - December 13, 2013 (11:59pm PT).

CONTRIBUTE NOW

DEF CON 22 | 2014.08.08
Fiducial responsibility is complicated
Fiducial responsibility is complicated
Then, a few days later


...  

- What do you mean you there’s $30,000 in PayPal?!
Then, a few days later


... 

- What do you mean you there’s $30,000 in PayPal?!
And thus was born the Open Crypto Audit Project

A U.S. non-profit organization, incorporated in the state of North Carolina, currently seeking federal 501c(3) tax-exempt designation
Open Crypto Audit Project

Mission

- Provide technical assistance to free open source software ("FOSS") projects in the public interest
- Coordinate volunteer technical experts in security, software engineering, and cryptography
- Conduct analysis and research on FOSS and other widely software in the public interest
Open Crypto Audit Project
Open Crypto Audit Project

Advisory Board

- Jean-Philippe Aumasson
- Nate Lawson
- Runa Sandvik
- Bruce Schneier
- Thomas Ptacek
- Jim Denaro
- Moxie Marlinspike
- Trevor Perrin
- Joseph Lorenzo Hall
And thus was born the Open Crypto Audit Project

OpenCryptoAudit.org/people
Open Crypto Audit Project

Officers & Directors
  - Matthew Green
  - Marcia Hoffman
  - Kenneth White
Our first Board meeting

DEF CON 22 | 2014.08.08
Making the connections...
The work begins

• Reached out to a few of the small handful of organizations that are capable of doing this work
• Great response from iSec Labs
• Open Technology Fund matching grant
Fast-forward
Fast-forward

Open Crypto Audit Project
TrueCrypt
Security Assessment

Prepared for:
Open Crypto Audit Project

Prepared by:
Andreas Junestam – Security Engineer
Nicolas Guigo – Security Engineer
Fast-forward

• iSec’s final security assessment:
  • Weak volume header key derivation (low kdf iteration count)
  • Sensitive information could be paged out from kernel stacks
  • Issues in the boot loader decompressor
  • Use of memset() to clear sensitive data

• Overall findings: “no evidence of backdoors or intentional flaws”
What does that mean?

- Password strength is crucial (same as always)
- Vulnerabilities discovered would likely require physical access to a mounted volume to construct exploit chains (scape key material, page files, etc)
- This is *not* a part of the TrueCrypt security model
- If your machine is compromised, disk crypto will not help you (see Culp-White Law, earlier)
- PSA: *All* major FDEs, including Bitlocker, DM-Crypt, and FileVault have *identical* attack vectors
- So far, so good.
But then...
Life is what happens when you’re busy making other plans.
TrueCrypt.org goes dark

• v. 7.2 is released, signed with developer keys (updated cert)
• Now read-only
• Archive is taken offline
• Recommendations for alternatives non-optimal
WARNING: Using TrueCrypt is not secure as it may contain unfixed security issues

This page exists only to help migrate existing data encrypted by TrueCrypt.

The development of TrueCrypt was ended in 5/2014 after Microsoft terminated support of Windows XP. Windows 8/7/Vista and later offer integrated support for encrypted disks and virtual disk images. Such integrated support is also available on other platforms (click here for more information). You should migrate any data encrypted by TrueCrypt to encrypted disks or virtual disk images supported on your platform.

Migrating from TrueCrypt to BitLocker:

If you have the system drive encrypted by TrueCrypt:

1. Encrypt the drive by BitLocker first. Open the Explorer:

![Windows Explorer Icon]
Our Response

• OCAP is continuing through with the Phase II (formal cryptanalysis) of the code
• We have created a trusted repository of source and binaries for all platforms
• Thomas Ptacek and Nate Lawson organizing Phase II
• We are considering several post-audit scenarios,
  • /possibly/ including financial support for a trusted fork
• *Many* challenges and questions remain
Secure Coding and Trust
Crypto Engineering

“There is no difference, from the attacker's point of view, between gross and tiny errors. Both of them are equally exploitable...This lesson is very hard to internalize. In the real world, if you build a bookshelf and forget to tighten one of the screws all the way, it does not burn down your house.”
— Maciej Cegłowski
(In)secure Coding: Where static analysis might help

- Unintended compiler optimizations
- Primitive type transpositions
- Pointer assignment vs. array assignments/terminators

From: [www.viva64.com/en/examples](http://www.viva64.com/en/examples) (recommend preparing a tall glass of Scotch first)
(In)secure Coding

“Source code is interesting. Everybody thinks if you have source code, you're going to be able to find everything wrong with [a system]. That’s a misconception. It’s nice to have source code so if you see something funny happening, you can check and see why – try to dig down... But for somebody to [manually] analyze millions of lines of source code, it’s just not going to happen.”

— Richard George
Former Technical Director
NSA Information Assurance Directorate
Retrospective Keynote, June, 2014
vimeo.com/97891042 [35:50]
Consider a hypothetical:

void Foo()
{
    char password[MAX_PASSWORD_LEN];
    InputPassword(password);
    ProcessPassword(password);
    memset(password, 0, sizeof(password));
}
Consider a hypothetical:

```c
void Foo()
{
    char password[MAX_PASSWORD_LEN];
    InputPassword(password);
    ProcessPassword(password);
    memset(password, 0, sizeof(password));
}
```
In Action

Credits: Program Verification Systems

(http://www.viva64.com/en/d/0208/)
void F1()
{
    TCHAR buf[100];
    _stprintf(buf, _T("Test: %d"), 123);
    MessageBox(NULL, buf, NULL, MB_OK);
    memset(buf, 0, sizeof(buf));
}

void F2()
{
    TCHAR buf[100];
    _stprintf(buf, _T("Test: %d"), 123);
    MessageBox(NULL, buf, NULL, MB_OK);
    RtlSecureZeroMemory(buf, sizeof(buf));
}
memset() didn’t
void F1()
{
    TCHAR buf[100];
    _stprintf(buf, _T("Test: %d"), 123);
    MessageBox(NULL, buf, NULL, MB_OK);
    memset(buf, 0, sizeof(buf));
}

void F2()
{
    TCHAR buf[100];
    _stprintf(buf, _T("Test: %d"), 123);
    MessageBox(NULL, buf, NULL, MB_OK);
    RtlSecureZeroMemory(buf, sizeof(buf));
}
RtlSecureZeroMemory() does
Multiple options

• Prefer secure memory/copy functions of stdlib
• Review limitations of the language/framework
• Understand compiler optimization side-effects
• GCC 4.4+ (2009) offers a pragma for function-level optimization control or prevention
  (see: gcc.gnu.org/onlinedocs/gcc-4.4.0/gcc/Optimize-Options.html)
• Learn from others’ experience
Multiple options

• Prefer secure memory/copy functions of stdlib
• Review limitations of the language/framework
• Understand compiler optimization side-effects
• GCC 4.4+ (2009) offers a pragma for function-level optimization control or prevention
  (see: gcc.gnu.org/onlinedocs/gcc-4.4.0/gcc/Optimize-Options.html)
• Learn from others’ experience
The Onion Router (TOR)

crypto.c
tortls.c
connection_or.c
onion.c
rendclient.c
tor-gencert.c

```c
int crypto_pk_private_sign_digest(....)
{
    char digest[DIGEST_LEN];
    ....
    memset(digest, 0, sizeof(digest));
    return r;
}
```
The Onion Router (TOR)

- crypto.c
- tortls.c
- connection_or.c
- onion.c
- rendclient.c
- tor-gencert.c

```c
int crypto_pk_private_sign_digest(....)
{
    char digest[DIGEST_LEN];
    ....
    memset(digest, 0, sizeof(digest));
    return r;
}
```
Network Security Services (NSS)

sha512.c

```c
SECStatus
SHA384_HashBuf(unsigned char *dest, const unsigned char *src,
                 PRUint32 src_length)
{
    SHA512Context ctx;
    unsigned int outLen;

    SHA384_Begin(&ctx);
    SHA512_Update(&ctx, src, src_length);
    SHA512_End(&ctx, dest, &outLen, SHA384_LENGTH);
    memset(&ctx, 0, sizeof(ctx);

    return SECSuccess;
}
```
Network Security Services (NSS)

sha512.c

```c
SECStatus
SHA384_HashBuf(unsigned char *dest, const unsigned char *src,
    PRUint32 src_length)
{
    SHA512Context ctx;
    unsigned int outLen;

    SHA384_Begin(&ctx);
    SHA512_Update(&ctx, src, src_length);
    SHA512_End(&ctx, dest, &outLen, SHA384_LENGTH);
    memset(&ctx, 0, sizeof(ctx));

    return SECSuccess;
}
```
void usage(void)
{
    static unsigned char *buf=NULL,*obuf=NULL;
    ....
    OPENSSL_cleanse(buf,sizeof(buf));
    OPENSSL_cleanse(obuf,sizeof(obuf));
    ....
}
void usage(void)
{
    static unsigned char *buf=NULL,*obuf=NULL;
    ....
    OPENSSL_cleanse(buf,sizeof(buf));
    OPENSSL_cleanse(obuf,sizeof(obuf));
    ....
}
On Trust
Probably not your threat model

(TS//SI//NF) Such operations involving supply-chain interdiction are some of the most productive operations in TAO, because they pre-position access points into hard target networks around the world.

(TS//SI//NF) Left: Intercepted packages are opened carefully; Right: A “load station” implants a beacon
Trust is complicated

Over 4900 unique CAs are or were transitively trusted by MSFT and/or Mozilla. That is just insane.
*Really* complicated
On Trust
<table>
<thead>
<tr>
<th></th>
<th>crontab</th>
<th>cull_incoming_pcaps</th>
<th>killall</th>
<th>smbpasswd</th>
<th>top</th>
</tr>
</thead>
<tbody>
<tr>
<td>[</td>
<td>curl</td>
<td>logger</td>
<td>md5sum</td>
<td>smoketest</td>
<td>tr</td>
</tr>
<tr>
<td>arping</td>
<td>cut</td>
<td>minidlna</td>
<td>sqlite3</td>
<td>sxnotify</td>
<td>tty</td>
</tr>
<tr>
<td>authcurl</td>
<td>dirname</td>
<td>mkfifo</td>
<td>sxstorageinfo</td>
<td>upload_events</td>
<td></td>
</tr>
<tr>
<td>awk</td>
<td>find</td>
<td>p0f-client</td>
<td>sxstrchr</td>
<td>upload_pcaps</td>
<td></td>
</tr>
<tr>
<td>basename</td>
<td>fsmon</td>
<td>pcap</td>
<td>tail</td>
<td>upload_stats</td>
<td></td>
</tr>
<tr>
<td>cgi-fcgi</td>
<td>handle_incoming_pcaps</td>
<td>send_event</td>
<td>tee</td>
<td>webfile.cgi.cgi</td>
<td></td>
</tr>
<tr>
<td>cmp</td>
<td>head</td>
<td>send_pcap</td>
<td>test</td>
<td>wget</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>xargs</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Strong crypto does not equal secure code
Forward Secrecy won’t help
Even with the best designs...
Things that make you go “hmmm”

ExclusiveArch: i686 x86_64

Summary: Utilities from the general purpose cryptography library with TLS implementation
Name: openssl
Version: 1.0.1e
Release: 4%{?buildid}%{?dist}
Epoch: 1

# We have to remove certain patented algorithms from the openssl source
# tarball with the hobble-openssl script which is included below.
# The original openssl upstream tarball cannot be shipped in the .src.rpm.
Source: openssl-%{version}-usa.tar.xz
Source1: hobble-openssl
Source2: Makefile.certificate
Source6: make-dummy-cert
Source7: renew-dummy-cert
Source8: openssl-thread-test.c
Source9: opensslconf-new.h
Source10: opensslconf-new-warning.h
Source11: README.FIPS
# Build changes
It bears repeating...

- Table 10, Queen E4. That should be Checkmate.
- Yes.
Usable Crypto is HARD
Take-Aways

- Many recent catastrophic failures are secure coding errors, not crypto errors
- Static analyzers are not enough
- Manual inspection is not enough
- Source code can result in unexpected binary code
- Subject matter experts (protocols, crypto, network) may bring more perspective than “enough” eyes
If the game is rigged, strong crypto probably won’t help you.
Looking forward
Recap: Where are we now?

- Phase I Report released April 23, 2014
- Beginning Phase II, to include:
  - Formal cryptanalysis
  - OSX & Linux review
  - Additional license work
- Partnering with Linux Foundation Core Infrastructure Initiative
- Auditing OpenSSL, possibly more
- Looking ahead!
- Trusted TC mirror: [github.com/AuditProject/truecrypt-verified-mirror](https://github.com/AuditProject/truecrypt-verified-mirror)
Final Thoughts & Goals

• Unpaid volunteers are not enough
• One-off bug bounties are not enough
• Encourage secure coding practices
• Support & create smarter test harnesses
• Develop a workable model for public code review
Open Discussion
Talk to us

@matthew_d_green
@kennwhite
@OpenCryptoAudit

admin@opencryptoaudit.org

IsTrueCryptAuditedYet.com (partly!)

OpenCryptoAudit.org

blog.cryptographyengineering.com

github.com/AuditProject/truecrypt-verified-mirror