Welcome to

Penetration testing IV
pentest cryptography and cracking

Henrik Lund Kramshøj hlk@zencurity.dk

Slides are available as PDF, kramshoej@Github

Goals for today

Introduce some common cryptographic protocols

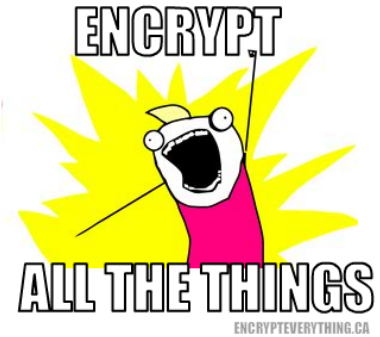
Introduce some often used pentest tools in cryptography

Increase paranoia to appropriate levels ☺

Reference some classics

Generic advice

Recommendations



- Lock your devices, phones, tables and computers
- Update software and apps
- Do NOT use the same password everywhere
- Watch out when using open wifi-networks
- Multiple browsers: one for Facebook, and separate for home banking apps?
- Multiple laptops? One for private data, one for work?
- Think of the data you produce, why do people take naked pictures and SnapChat them?
- Use pseudonyms and aliases, do not use your real name everywhere
- Enable encryption: IMAP**S** POP3**S** HTTP**S** TOR OpenPGP VPN SSL/TLS

Stop watching us!

Appropriate paranoia

par·a·noi·a
/ˌparəˈnoiə/

noun
noun: **paranoia**

1. a mental condition characterized by delusions of persecution, unwarranted jealousy, or exaggerated self-importance, typically elaborated into an organized system. It may be an aspect of chronic personality disorder, of drug abuse, or of a serious condition such as schizophrenia in which the person loses touch with reality.
   synonyms: persecution complex, delusions, obsession, psychosis More
- suspicion and mistrust of people or their actions without evidence or justification.
  "the global paranoia about hackers and viruses"

Origin

GREEK
para
irregular

GREEK
noos
mind

GREEK
paranoos
distracted

MODERN LATIN
paranoia
early 19th cent.

More

Source: google paranoia definition

Face reality

From the definition:

> suspicion and mistrust of people or their actions **without evidence or justification**. ”the global paranoia about hackers and viruses”

It is not paranoia when:

- Criminals sell your credit card information and identity theft
- Trade infected computers like a commodity
- Governments write laws that allows them to introduce back-doors - and use these
- Governments do blanket surveillance of their population
- Governments implement censorship, threaten citizens and journalist

You are not paranoid when there are people actively attacking you!

Credit card fraud and identity theft statistics

Credit Card Fraud Statistics

Share This

Statistic Verification
Source: Consumer Sentinel Network, U.S. Department of Justice
Date Verified: 7.23.2012

Credit Card Fraud Statistics Statistics	Data
Percent of Americans who have been victims of credit card fraud	10 %
Percent of Americans who have been victims of debit or ATM card fraud	7 %
Median amount reported on credit card fraud	$399
Percent of all financial fraud related to credit cards	40 %
Total amount of credit card fraud worldwide	$5.55 Billion

Source: `http://www.statisticbrain.com/credit-card-fraud-statistics/`

Identity theft statistics

Identity Theft / Fraud Statistics

Share This

Statistic Verification
Source: U.S. Department of Justice, Javelin Strategy & Research
Research Date: 6.18.2013
Identity theft is defined as the unauthorized use or attempted misuse of an existing credit card or other existing account, the misuse of personal information to open a new account or for another fraudulent purpose, or a combination of these types of misuse.

Identity Theft / Fraud Statistics	Data
Average number of U.S. identity fraud victims annually	11,571,900
Percent of U.S. households that reported some type of identity fraud	7 %
Average financial loss per identity theft incident	$4,930
Total financial loss attributed to identity theft in 2013	$21 billion
Total financial loss attributed to identity theft in 2010	$13.2 billion
Percent of Reported Identity Thefts by Type of Fraud	**Percent Reported**
Misuse of Existing Credit Card	64.1 %
Misuse of Other Existing Bank Account	35 %
Misuse of Personal Information	14.2 %

Source: `http://www.statisticbrain.com/identity-theft-fraud-statistics/`

Use protection - always

A vulnerability can and will be abused

What if I told you:

Criminals will be happy to leverage backdoors created by government

It does not matter if the crypto product has a weakness to allow investigations or the software has a backdoor to help law enforcement. Data and vulnerabilities WILL be abused and exploited.

Make sure to read *Keys Under Doormats: mandating insecurity by requiring government access to all data and communications*

`https://www.cl.cam.ac.uk/~rja14/Papers/doormats.pdf`

`http://savecrypto.org`

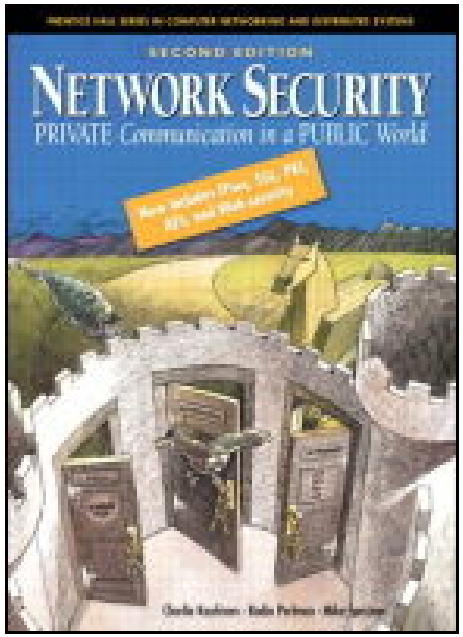
Why think of security?

> Privacy is necessary for an open society in the electronic age. Privacy is not secrecy. A private matter is something one doesn't want the whole world to know, but a secret matter is something one doesn't want anybody to know. Privacy is the power to selectively reveal oneself to the world. A Cypherpunk's Manifesto by Eric Hughes, 1993

Copied from `https://cryptoparty.org/wiki/CryptoParty`

Starting the crypto journey

Where do we start?



Private Communications in an Public World

Very nice book listing our knowledge about main protocols in use on the internet today - even though the book is from 2002! Includes: IPsec, SSL/TLS, PGP, PKI, AES m.fl.

First advice

Use technology

Learn the technology - read the freaking manual

Think about the data you have, upload, facebook license?! WTF!

Think about the data you create - nude pictures taken, where will they show up?

- Turn off features you don't use
- Turn off network connections when not in use
- Update software and applications
- Turn on encryption: IMAP**S**, POP3**S**, HTTP**S** also for data at rest, full disk encryption, tablet encryption
- Lock devices automatically when not used for 10 minutes
- Dont trust fancy logins like fingerprint scanner or face recognition on cheap devices

Chaosreader

Chaosreader

Chaosreader Report

Created at: Sun Nov 16 21:04:18 2003, Type: snoop

Image Report - Click here for a report on captured images.
GET/POST Report (Empty) - Click here for a report on HTTP GETs and POSTs.
HTTP Proxy Log - Click here for a generated proxy style HTTP log.

TCP/UDP/... Sessions

1.	**Sun Nov 16 20:38:22 2003**	30 s	192.168.1.3:1368 <-> 192.77.84.99:80	web	383 bytes	• as_html
2.	**Sun Nov 16 20:38:22 2003**	29 s	192.168.1.3:1366 <-> 192.77.84.99:80	web	381 bytes	• as_html

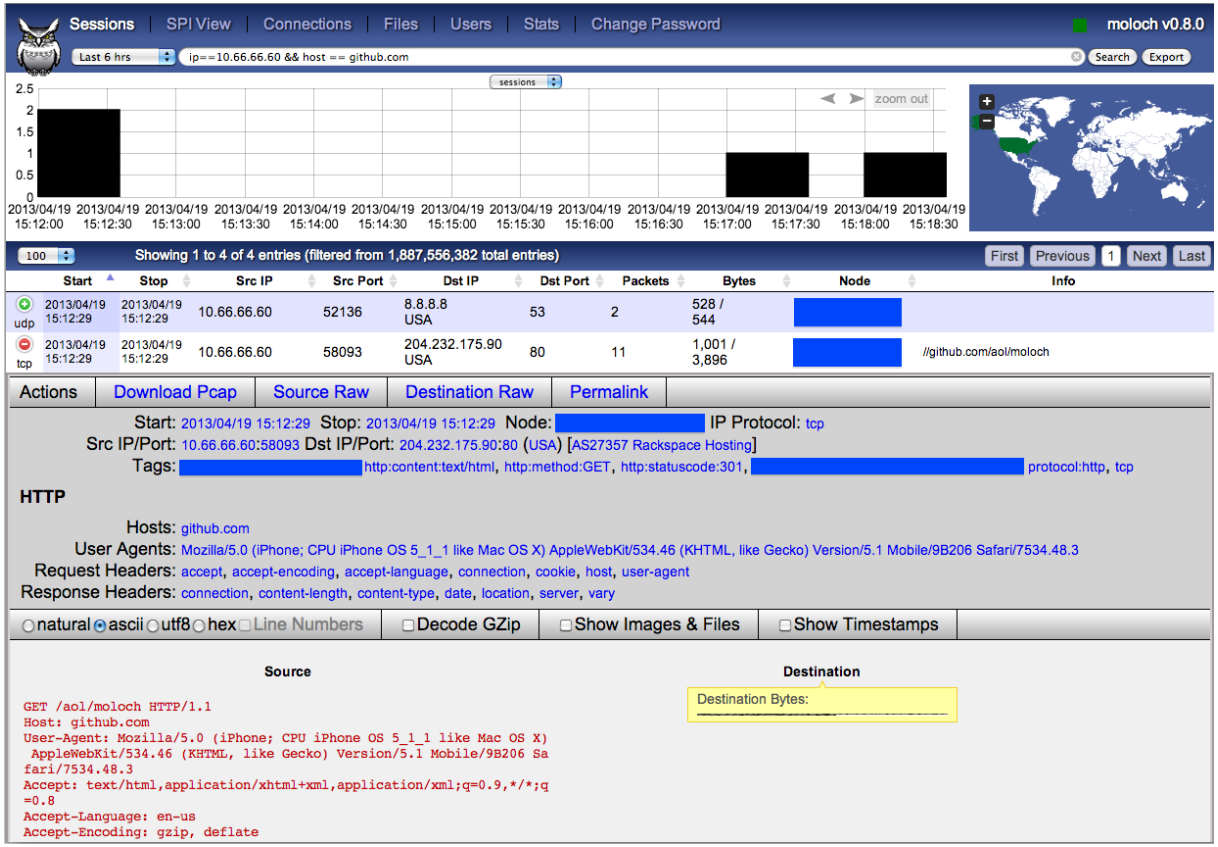
Simple but illustrative program

Read a pcap - packet capture into this tool chaosreader

Output HTML with nice index - usefull for quick demos

`http://chaosreader.sourceforge.net/`

Big data example Moloch



Picture from `https://github.com/aol/moloch`
Be your own GCHQ ... capture all, index all, search all

Cryptography

Cryptography or cryptology is the practice and study of techniques for secure communication

Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key, to ensure confidentiality, example algorithm AES

Public-key cryptography (like RSA) uses two related keys, a key pair of a public key and a private key. This allows for easier key exchanges, and can provide confidentiality, and methods for signatures and other services

Source: `https://en.wikipedia.org/wiki/Cryptography`

Kryptografiske principper

Algoritmerne er kendte

Nøglerne er hemmelige

Nøgler har en vis levetid - de skal skiftes ofte

Et successfuldt angreb på en krypto-algoritme er enhver genvej som kræver mindre arbejde end en gennemgang af alle nøglerne

Nye algoritmer, programmer, protokoller m.v. skal gennemgås nøje!

Se evt. Snake Oil Warning Signs: Encryption Software to Avoid
`http://www.interhack.net/people/cmcurtin/snake-oil-faq.html`

DES, Triple DES og AES

AES

Advanced Encryption Standard

DES kryptering baseret på den IBM udviklede Lucifer algoritme har været benyttet gennem mange år

Der blev i 2001 vedtaget en ny standard algoritme Advanced Encryption Standard (AES) som afløser Data Encryption Standard (DES)

Algoritmen hedder Rijndael og er udviklet af Joan Daemen og Vincent Rijmen.

Se også `https://en.wikipedia.org/wiki/Advanced_Encryption_Standard`

Formålet med kryptering

kryptering er den eneste måde at sikre:

fortrolighed

autenticitet / integritet

Secure protocols

Securing e-mail

- Pretty Good Privacy - Phil Zimmermann
- OpenPGP = e-mail security

Network sessions use SSL/TLS

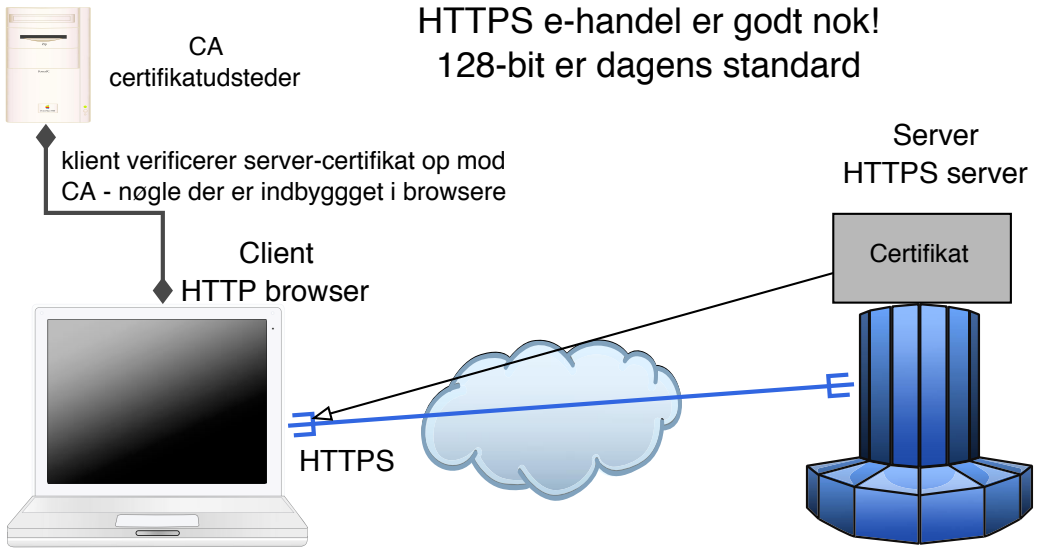
- Secure Sockets Layer SSL / Transport Layer Services TLS
- Encrypting data sent and received
- SSL/TLS already used for many protocols as a wrapper: POP3S, IMAPS, SSH, SMTP+TLS m.fl.

Encrypting traffic at the network layer - Virtual Private Networks VPN

- IPsec IP Security Framework, se også L2TP
- PPTP Point to Point Tunneling Protocol - dårlig og usikker, brug den ikke mere!
- OpenVPN uses SSL/TLS across TCP or UDP

Note: SSL/TLS is not trivial to implement, key management!

SSL og TLS



Oprindeligt udviklet af Netscape Communications Inc.

Secure Sockets Layer SSL er idag blevet adopteret af IETF og kaldes derfor også for Transport Layer Security TLS TLS er baseret på SSL Version 3.0

RFC-2246 The TLS Protocol Version 1.0 fra Januar 1999

RFC-3207 SMTP STARTTLS

SSL

The 'S' in HTTPS stands for 'secure' and the security is provided by SSL/TLS. SSL/TLS is a standard network protocol which is implemented in every browser and web server to provide confidentiality and integrity for HTTPS traffic.

Nu vi snakker om kryptering - SSL overalt?

Kan vi klare det på vores servere?

Google kan:
`http://www.imperialviolet.org/2010/06/25/overclocking-ssl.html`

Men alt for få gør det

Næste spørgsmål er så hvilke rod-certifikater man stoler på ...

Heartbleed CVE-2014-0160

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.

Source: `http://heartbleed.com/`

Heartbleed is yet another bug in SSL products

What versions of the OpenSSL are affected?
Status of different versions:

* OpenSSL 1.0.1 through 1.0.1f (inclusive) are vulnerable
* OpenSSL 1.0.1g is NOT vulnerable
* OpenSSL 1.0.0 branch is NOT vulnerable
* OpenSSL 0.9.8 branch is NOT vulnerable

Bug was introduced to OpenSSL in December 2011 and has been out in the wild since OpenSSL release 1.0.1 on 14th of March 2012. OpenSSL 1.0.1g released on 7th of April 2014 fixes the bug.

It's just a bug - but a serious one

Why is heartbleed different?

Great PR, name, web site, logo

OpenSSL is very widespread

OpenSSL has been criticized before

The spotlight is now on a lot of products, infrastructure

BOTH Open Source products and Proprietary products hurt by this

TL;DR
OpenSSL is everywhere and an example of our dependency on weak components

Key points after heartbleed

Source: picture source

`https://www.duosecurity.com/blog/heartbleed-defense-in-depth-part-2`

- Writing SSL software and other secure crypto software is hard
- Configuring SSL is hard
  check you own site `https://www.ssllabs.com/ssltest/`
- SSL is hard, finding bugs ”all the time”`http://armoredbarista.blogspot.dk/2013/01/a-brief-chronology-of-ssltls-attacks.html`
- Rekeying is hard - slow, error prone, manual proces - Automate!
- Proof of concept programs exist - good or bad?

Proof of concept programs exist - good or bad?

Some of the tools released shortly after Heartbleed announcement

- https://github.com/FiloSottile/Heartbleed tool i Go
  site http://filippo.io/Heartbleed/
- https://github.com/titanous/heartbleeder tool i Go
- http://s3.jspenguin.org/ssltest.py PoC
- https://gist.github.com/takeshixx/10107280 test tool med STARTTLS support
- http://possible.lv/tools/hb/ test site
- https://twitter.com/richinseattle/status/453717235379355649 Practical Heartbleed attack against session keys links til, https://www.mattslifebytes.com/?p=533 og "Fully automated here "
  https://www.michael-p-davis.com/using-heartbleed-for-hijacking-user-sessio
- Metasploit er også opdateret på master repo
  https://twitter.com/firefart/status/453758091658792960
  https://github.com/rapid7/metasploit-framework/blob/master/modules/auxiliar
  scanner/ssl/openssl_heartbleed.rb

Heartbleed hacking

```
06b0: 2D 63 61 63 68 65 0D 0A 43 61 63 68 65 2D 43 6F  -cache..Cache-Co
06c0: 6E 74 72 6F 6C 3A 20 6E 6F 2D 63 61 63 68 65 0D  ntrol: no-cache.
06d0: 0A 0D 0A 61 63 74 69 6F 6E 3D 67 63 5F 69 6E 73  ...action=gc_ins
06e0: 65 72 74 5F 6F 72 64 65 72 26 62 69 6C 6C 6E 6F  ert_order&billno
06f0: 3D 50 5A 4B 31 31 30 31 26 70 61 79 6D 65 6E 74  =PZK1101&payment
0700: 5F 69 64 3D 31 26 63 61 72 64 5F 6E 75 6D 62 65  _id=1&card_numbe
0710: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX   r=4060xxxx413xxx
0720: 39 36 26 63 61 72 64 5F 65 78 70 5F 6D 6F 6E 74  96&card_exp_mont
0730: 68 3D 30 32 26 63 61 72 64 5F 65 78 70 5F 79 65  h=02&card_exp_ye
0740: 61 72 3D 31 37 26 63 61 72 64 5F 63 76 6E 3D 31  ar=17&card_cvn=1
0750: 30 39 F8 6C 1B E5 72 CA 61 4D 06 4E B3 54 BC DA  09.l..r.aM.N.T..
```

- Obtained using Heartbleed proof of concepts - Gave full credit card details
- ”can XXX be exploited- yes, clearly! PoCs ARE needed
  without PoCs even Akamai wouldn’t have repaired completely!
- The internet was ALMOST fooled into thinking getting private keys from Heartbleed was not possible - scary indeed.

Analysis of the heartbleed bug

- analyse af problemet i koden
  `http://blog.existentialize.com/diagnosis-of-the-openssl-heartbleed-bug.html`
- IDS regler Detecting OpenSSL Heartbleed with Suricata
  `http://blog.inliniac.net/2014/04/08/detecting-openssl-heartbleed-with-suric`
- god beskrivelse af hvordan man kan fixe hurtigere hvis man har automatiseret infrastruktur
  `https://www.getpantheon.com/heartbleed-fix`
- Mange blogindlæg om emnet - eksempelvis
  `http://blog.fox-it.com/2014/04/08/openssl-heartbleed-bug-live-blog/`
- ”nse script ssl-heartbleed.nse committed to nmap as rev 32798. ”
- You can now use Masscan to scan the whole internet for the Hearbleed vulnerability in under 6 minutes `https://twitter.com/jedisct1/status/453679529710460928`
  og `https://github.com/robertdavidgraham/masscan/commit/23497c448b0a1c7058e84`

Heartbleed Conclusions

Nothing new, but more focus on problems?
Really is there something new in this?

Software has bugs - stay vigilant, implement defense in depth

Software need funding - especially software used in our critical systems

Security needs proof of concepts and open communication
Akamai fix that wasn't good enough!

TL;DR Fund more security audits, stop using untested/unaudited software

Weak DH paper

Weak Diffie-Hellman and the Logjam Attack

Good News! Your browser is safe against the Logjam attack.

Diffie-Hellman key exchange is a popular cryptographic algorithm that allows Internet protocols to agree on a shared key and negotiate a secure connection. It is fundamental to many protocols including HTTPS, SSH, IPsec, SMTPS, and protocols that rely on TLS.

We have uncovered several weaknesses in how Diffie-Hellman key exchange has been deployed:

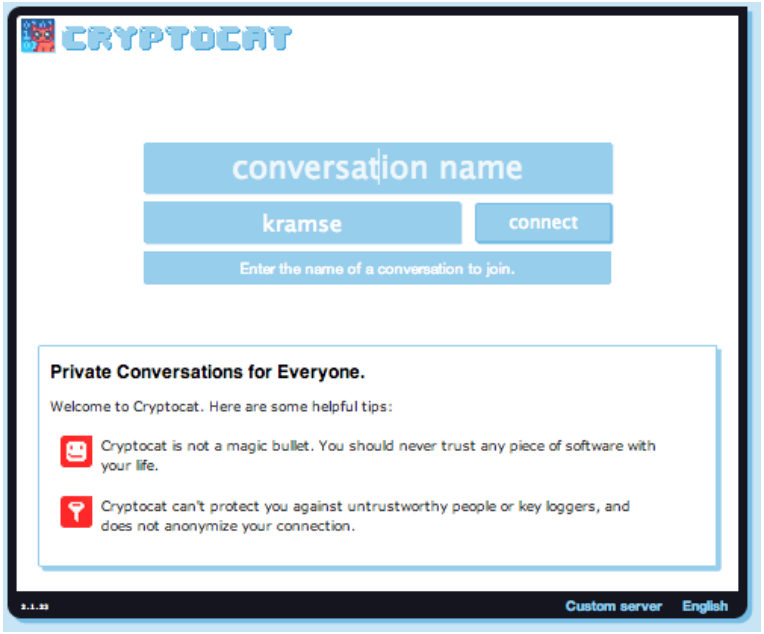
1. **Logjam attack against the TLS protocol.** The Logjam attack allows a man-in-the-middle attacker to downgrade vulnerable TLS connections to 512-bit export-grade cryptography. This allows the attacker to read and modify any data passed over the connection. The attack is reminiscent of the FREAK attack, but is due to a flaw in the TLS protocol rather than an implementation vulnerability, and attacks a Diffie-Hellman key exchange rather than an RSA key exchange. The attack affects any server that supports `DHE_EXPORT` ciphers, and affects all modern web browsers. 8.4% of the Top 1 Million domains were initially vulnerable.

2. **Threats from state-level adversaries.** Millions of HTTPS, SSH, and VPN servers all use the same prime numbers for Diffie-Hellman key exchange. Practitioners believed this was safe as long as new key exchange messages were generated for every connection. However, the first step in the number field sieve—the most efficient algorithm for breaking a Diffie-Hellman connection—is dependent only on this prime. After this first step, an attacker can quickly break individual connections.

   We carried out this computation against the most common 512-bit prime used for TLS and demonstrate that the Logjam attack can be used to downgrade connections to 80% of TLS servers supporting `DHE_EXPORT`. We further estimate that an academic team can break a 768-bit prime and that a nation-state can break a 1024-bit prime. Breaking the single, most common 1024-bit prime used by web servers would allow passive eavesdropping on connections to 18% of the Top 1 Million HTTPS domains. A second prime would allow passive decryption of connections to 66% of VPN servers and 26% of SSH servers. A close reading of published NSA leaks shows that the agency's attacks on VPNs are consistent with having achieved such a break.

Source: `https://weakdh.org/` and `https://weakdh.org/imperfect-forward-secrecy-ccs15.pdf`

Audits



Truecrypt audit

`https://isecpartners.github.io/news/2014/04/14/iSEC-Completes-Truecrypt-Audit.html`

Cryptocat audit

`https://blog.crypto.cat/2013/02/cryptocat-passes-security-audit-with-flying-colors/`

Bettercrypto.org

```
ssl_prefer_server_ciphers on;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # not possible to do exclusive
ssl_ciphers 'EDH+CAMELLIA:EDH+aRSA:EECDH+aRSA+AESGCM:EECDH+aRSA+SHA384:EECDH+\
  \aRSA+SHA256:EECDH:+CAMELLIA256:+AES256:+CAMELLIA128:+AES128:+SSLv3:!aNULL:!\
  \eNULL:!LOW:!3DES:!MD5:!EXP:!PSK:!DSS:!RC4:!SEED:!ECDSA:CAMELLIA256-SHA:AES256\
  \-SHA:CAMELLIA128-SHA:AES128-SHA';
add_header Strict-Transport-Security max-age=15768000; # six months
# use this only if all subdomains support HTTPS!
# add_header Strict-Transport-Security "max-age=15768000; includeSubDomains";
```

Listing 2.6: *SSL settings for nginx*
[configuration/Webservers/nginx/default]

Overview

This whitepaper arose out of the need for system administrators to have an updated, solid, well researched and thought-through guide for configuring SSL, PGP, SSH and other cryptographic tools in the post-Snowden age. ... This guide is specifically written for these system administrators.

`https://bettercrypto.org/`

sslscan

```
root@kali:~# sslscan --ssl2 web.gratisdns.dk
Version: 1.10.5-static
OpenSSL 1.0.2e-dev xx XXX xxxx

Testing SSL server web.gratisdns.dk on port 443
...
  SSL Certificate:
Signature Algorithm: sha256WithRSAEncryption
RSA Key Strength:    2048

Subject:  *.gratisdns.dk
Altnames: DNS:*.gratisdns.dk, DNS:gratisdns.dk
Issuer:   AlphaSSL CA - SHA256 - G2
```

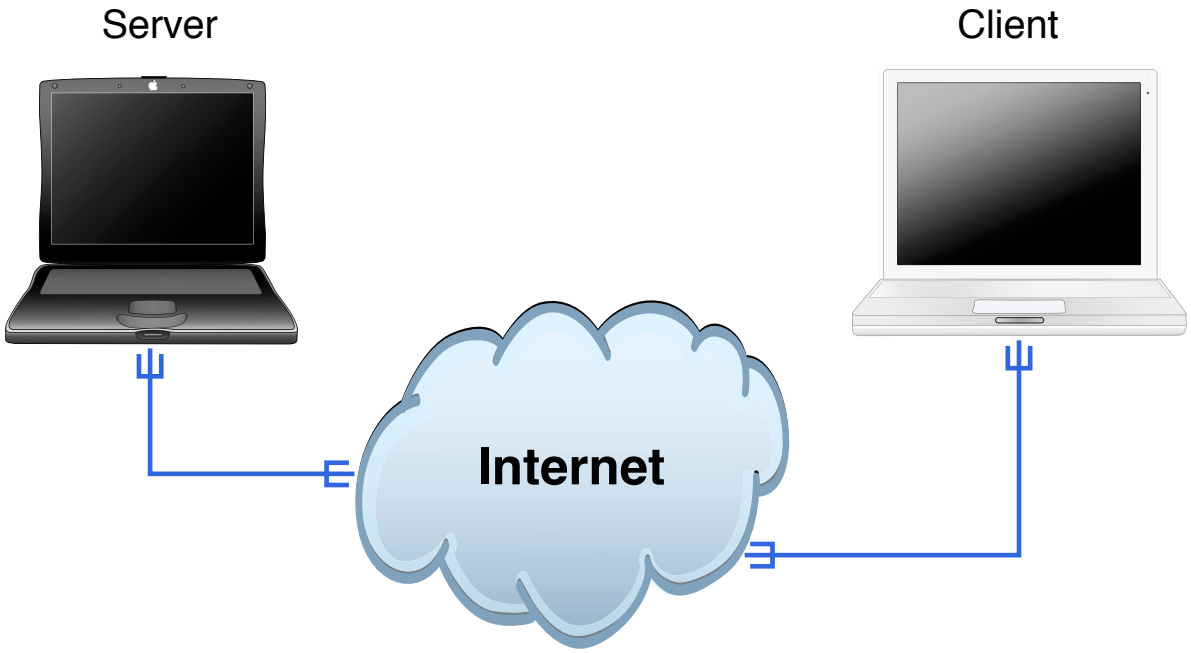
Source: Originally sslscan from http://www.titania.co.uk but use the version on Kali

sslyze

```
root@kali:~# sslyze --sslv2 web.gratisdns.dk:443
...
 CHECKING HOST(S) AVAILABILITY
 -----------------------------
   web.gratisdns.dk:443                       => 91.221.196.204:443

 SCAN RESULTS FOR WEB.GRATISDNS.DK:443 - 91.221.196.204:443
 ----------------------------------------------------------

Unhandled exception when processing --sslv2:
utils.ctSSL.errors.ctSSLFeatureNotAvailable - SSLv2 disabled.

 SCAN COMPLETED IN 0.09 S
 ------------------------
```

Source: Originally iSECPartners sslyze but development moved to:
`https://github.com/nabla-c0d3/sslyze`

Demo: Playtime - try sslscan and sslyze



Playtime - try sslscan and sslyze

Note: neither sslscan nor sslyze should be considered attacks - but may result in public shaming if bad security found - like SSLv2 and SSLv3

PRNG

Debian OpenSSL [edit]

In May 2008, security researcher Luciano Bello revealed his discovery that changes made in 2006 to the random number generator in the version of the OpenSSL package distributed with Debian GNU/Linux and other Debian-based distributions, such as Ubuntu, dramatically reduced the entropy of generated values and made a variety of security keys vulnerable to attack.[10][11] The security weakness was caused by changes made to the openssl code by a Debian developer in response to compiler warnings of apparently redundant code.[12] This caused a massive worldwide regeneration of keys, and despite all attention the issue got, it could be assumed many of these old keys are still in use. Key types affected include SSH keys, OpenVPN keys, DNSSEC keys, key material for use in X.509 certificates and session keys used in SSL/TLS connections. Keys generated with GnuPG or GNUTLS are not affected as these programs used different methods to generate random numbers. Non-Debian-based Linux distributions are also unaffected. This security vulnerability was promptly patched after it was reported.

`https://en.wikipedia.org/wiki/Random_number_generator_attack#Debian_OpenSSL`

The random number generator is VITAL for crypto security

Check out modern CPUs and Linux response to `https://en.wikipedia.org/wiki/RdRand`

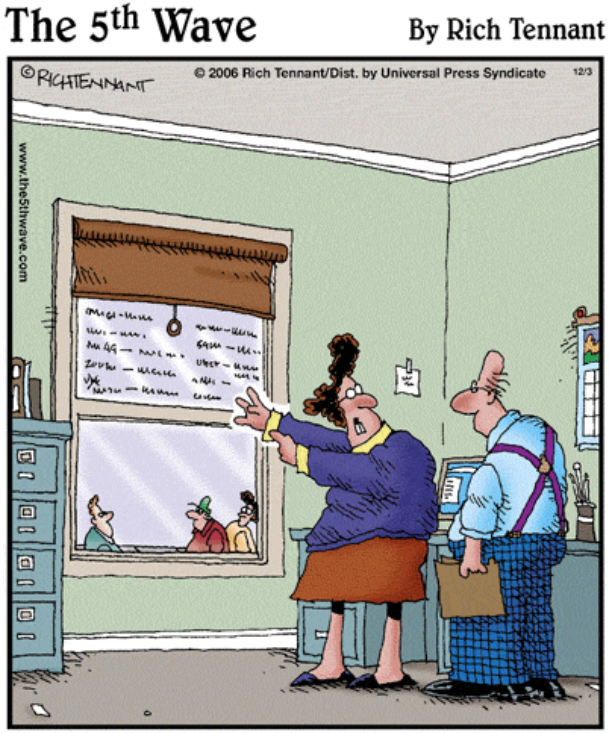
PRNG and generating WPA2 Passwords

Scrutinizing WPA2 Password Generating Algorithms in Wireless Router

#hacklu REing SOHO WPA2 generating algos by @enovella `http://archive.hack.lu/2015/hacklu15_enovella_reversing_routers.pdf` ... [UARTing; findings sobering]

Source: `https://twitter.com/daniel_bilar/status/656797778157412352`

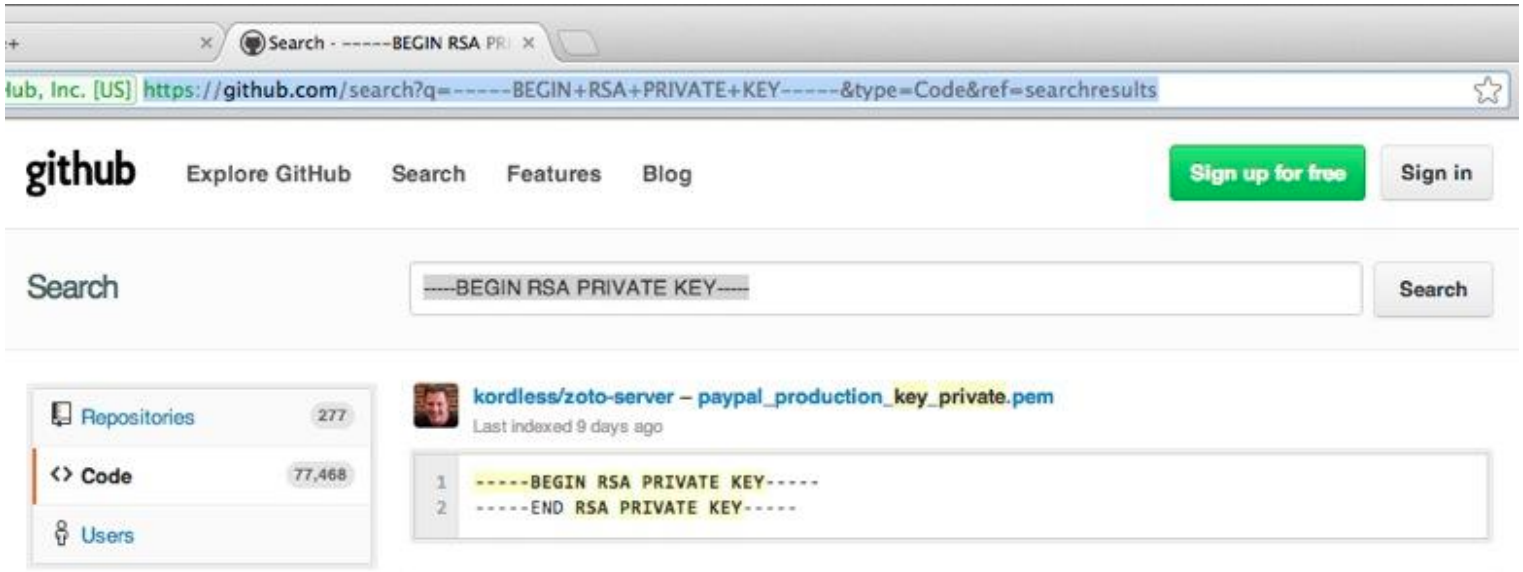
Formål: sund paranoia



"Don't be silly — of course my passwords are safe. I keep them written on my window, but then I pull the shade if anyone walks in the room."

Opbevaring af passwords

January 2013: Github Public passwords?



Sources:

https://twitter.com/brianaker/status/294228373377515522

http://www.webmonkey.com/2013/01/users-scramble-as-github-search-exposes-passwords-security-de

http://www.leakedin.com/

http://www.offensive-security.com/community-projects/google-hacking-database/

Use different passwords for different sites, yes - every site!

Simple Network Management Protocol

SNMP er en protokol der supporteres af de fleste professionelle netværksenheder, såsom switche, routere

hosts - skal slås til men følger som regel med

SNMP bruges til:

- *network management*
- statistik
- rapportering af fejl - SNMP traps

sikkerheden baseres på community strings der sendes som klartekst ...

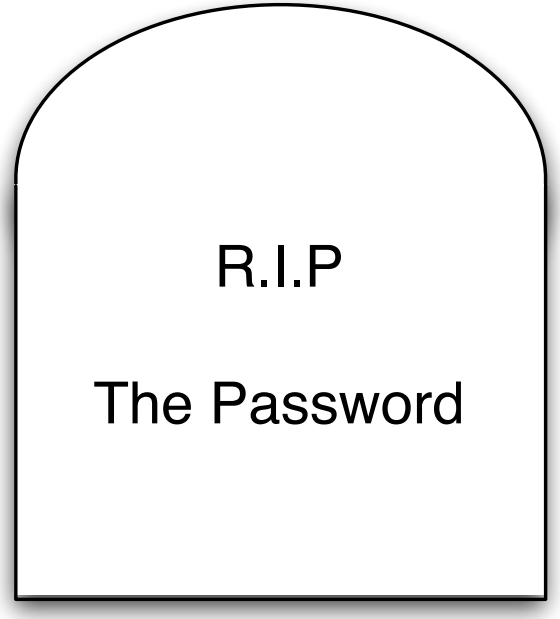
det er nemmere at brute-force en community string end en brugerid/kodeord kombination

brute force

hvad betyder bruteforcing?
afprøvning af alle mulighederne

```
Hydra v2.5 (c) 2003 by van Hauser / THC <vh@thc.org>
Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]]
[-o FILE] [-t TASKS] [-g TASKS] [-T SERVERS] [-M FILE] [-w TIME]
[-f] [-e ns] [-s PORT] [-S] [-vV] server service [OPT]

Options:
  -S         connect via SSL
  -s PORT    if the service is on a different default port, define it here
  -l LOGIN   or -L FILE login with LOGIN name, or load several logins from FILE
  -p PASS    or -P FILE try password PASS, or load several passwords from FILE
  -e ns      additional checks, "n" for null password, "s" try login as pass
  -C FILE    colon seperated "login:pass" format, instead of -L/-P option
  -M FILE    file containing server list (parallizes attacks, see -T)
  -o FILE    write found login/password pairs to FILE instead of stdout
...
```

Are passwords dead?



Can we stop using passwords?

Muffett on Passwords has a long list of password related information, from the author of crack `http://en.wikipedia.org/wiki/Crack_(password_software)`

`http://dropsafe.crypticide.com/muffett-passwords`

Google looks to ditch passwords for good

”Google is currently running a pilot that uses a YubiKey cryptographic card developed by Yubico

The YubiKey NEO can be tapped on an NFC-enabled smartphone, which reads an encrypted one-time password emitted from the key fob.”

Source:

`http://www.zdnet.com/google-looks-to-ditch-passwords-for-good-with-nfc-based-replacement-70000`

NT hashes

NT LAN manager hash værdier er noget man typisk kan samle op i netværk

det er en hash værdi af et password som man ikke burde kunne bruge til noget - hash algoritmer er envejs

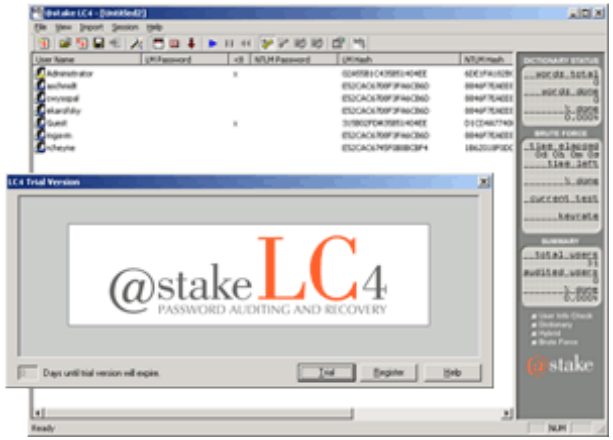
opbygningen gør at man kan forsøge brute-force på 7 tegn ad gangen!

en moderne pc med l0phtcrack kan nemt knække de fleste password på få dage!

og sikkert 25-30% indenfor den første dag - hvis der ingen politik er omkring kodeord!

ved at generere store tabeller, eksempelvis 100GB kan man dække mange hashværdier af passwords med almindelige bogstaver, tal og tegn - og derved knække password-shashes på sekunder. Søg efter rainbowcrack med google

l0phtcrack LC4



Consider that at one of the largest technology companies, where policy required that passwords exceed 8 characters, mix cases, and include numbers or symbols...

L0phtCrack obtained 18% of the passwords in 10 minutes
90% of the passwords were recovered within 48 hours on a Pentium II/300
The Administrator and most Domain Admin passwords were cracked
http://www.atstake.com/research/lc/

Pass the hash

Lots of tools in pentesting pass the hash, reuse existing credentials and tokens *Still Passing the Hash 15 Years Later* `http://passing-the-hash.blogspot.dk/2013/04/pth-toolkit-for-kali-interim-status.html`

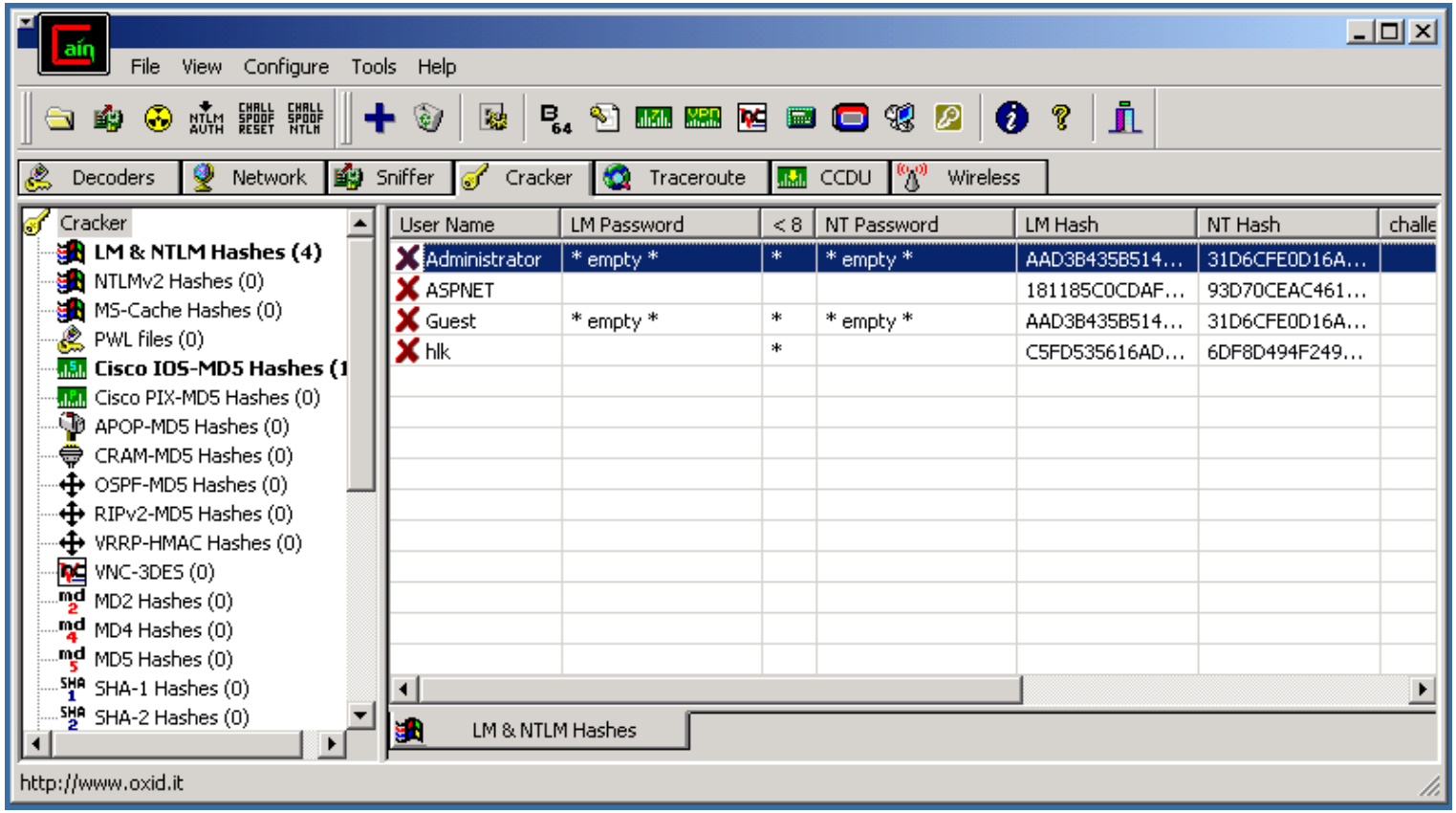
> If a domain is built using only modern Windows OSs and COTS products (which know how to operate within these new constraints), and configured correctly with no shortcuts taken, then these protections represent a big step forward.

Source:

`http://www.harmj0y.net/blog/penetesting/pass-the-hash-is-dead-long-live-pass-th`

`https://samsclass.info/lulz/pth-8.1.htm`

Cain og Abel



Cain og Abel anbefales `http://www.oxid.it`

John the ripper

> John the Ripper is a fast password cracker, currently available for many flavors of Unix (11 are officially supported, not counting different architectures), Windows, DOS, BeOS, and OpenVMS. Its primary purpose is to detect weak Unix passwords. Besides several crypt(3) password hash types most commonly found on various Unix flavors, supported out of the box are Kerberos AFS and Windows NT/2000/XP/2003 LM hashes, plus several more with contributed patches.

UNIX passwords kan knækkes med alec Muffets kendte Crack program eller eksempelvis John The Ripper `http://www.openwall.com/john/`

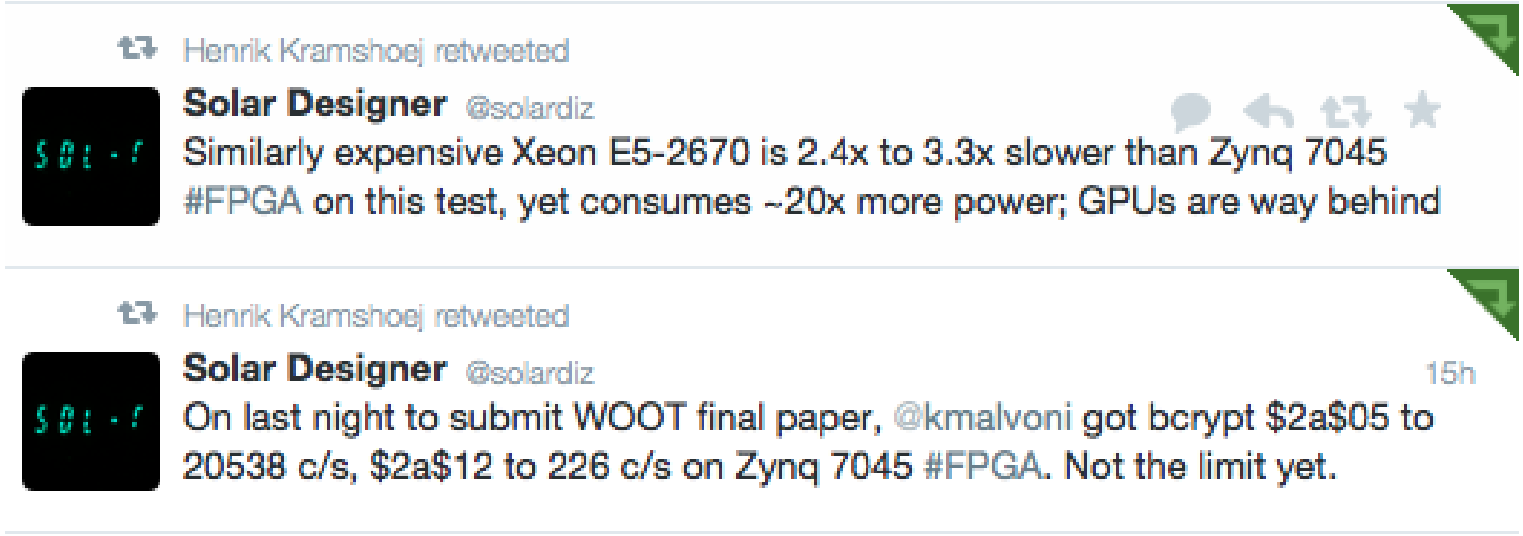
Cracking passwords

- Hashcat is the world's fastest CPU-based password recovery tool.
- oclHashcat-plus is a GPGPU-based multi-hash cracker using a brute-force attack (implemented as mask attack), combinator attack, dictionary attack, hybrid attack, mask attack, and rule-based attack.
- oclHashcat-lite is a GPGPU cracker that is optimized for cracking performance. Therefore, it is limited to only doing single-hash cracking using Markov attack, Brute-Force attack and Mask attack.
- John the Ripper password cracker old skool men stadig nyttig

Source:

`http://hashcat.net/wiki/`
`http://www.openwall.com/john/`

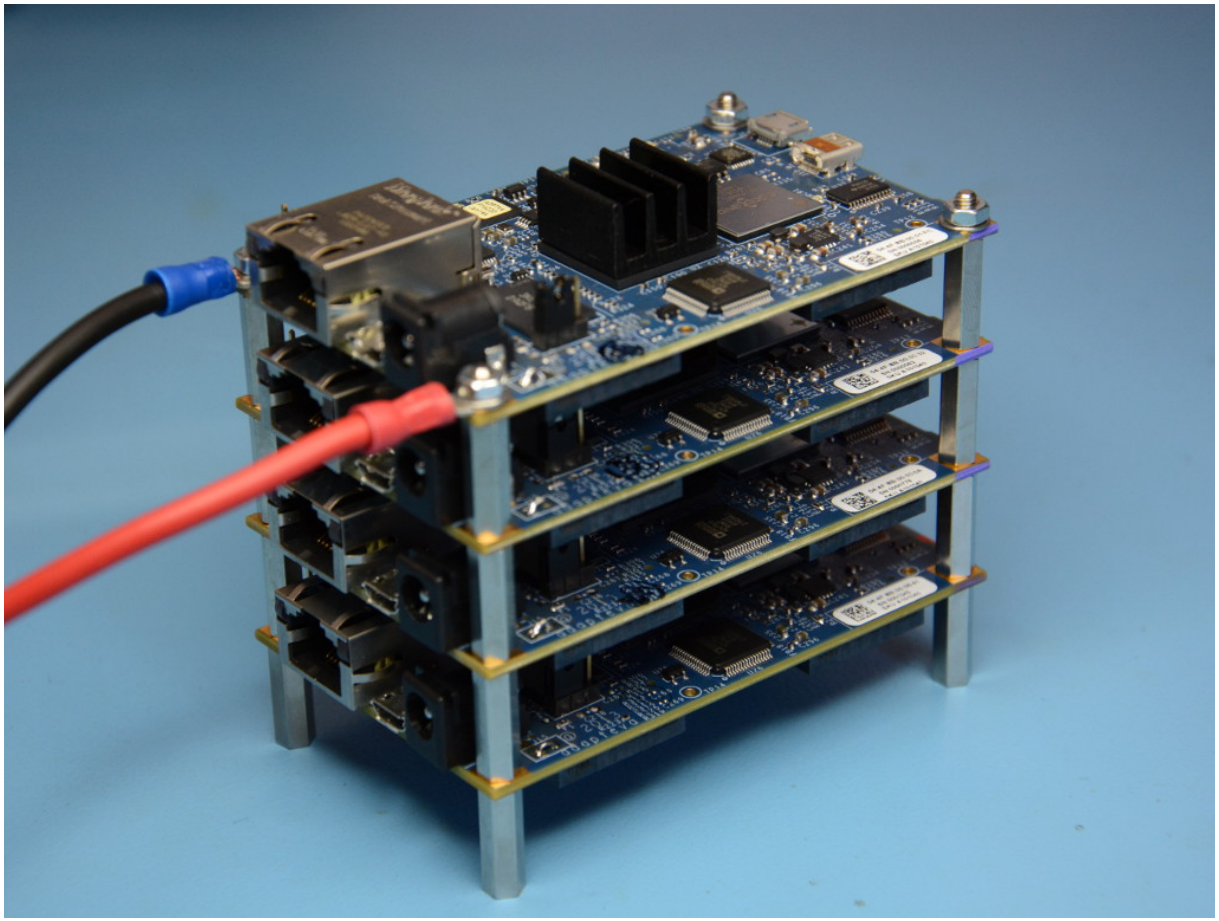
Parallella John



`https://twitter.com/solardiz/status/492037995080712192`

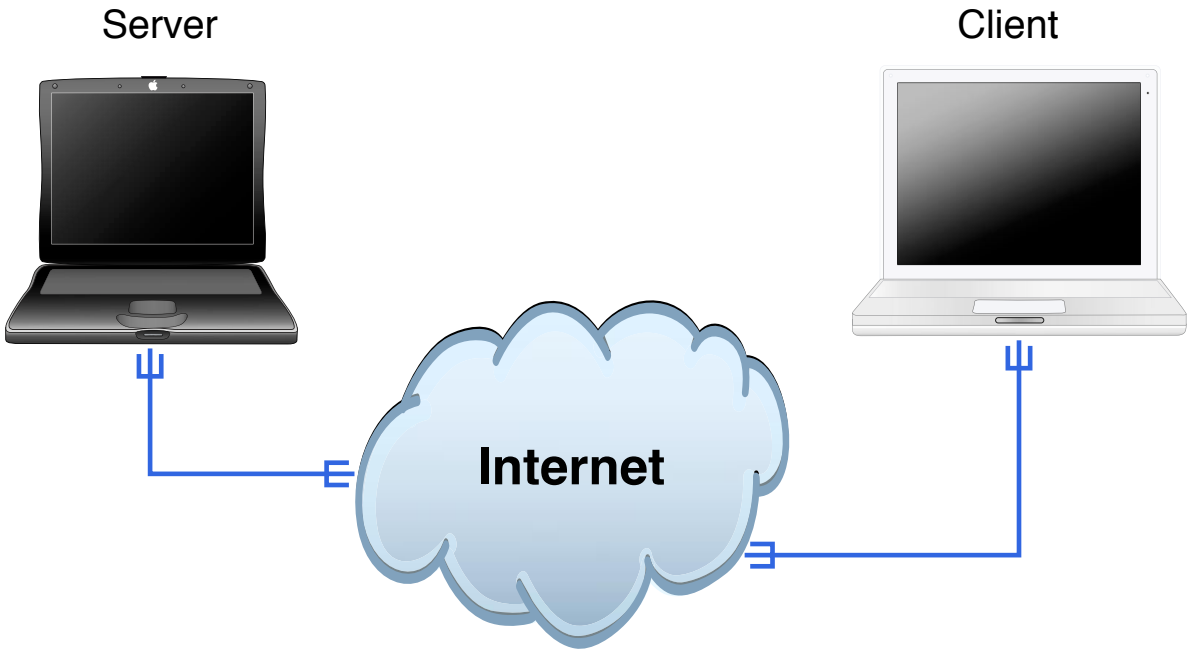
Warning: FPGA hacking - not finished part of presentation

Stacking Parallella boards



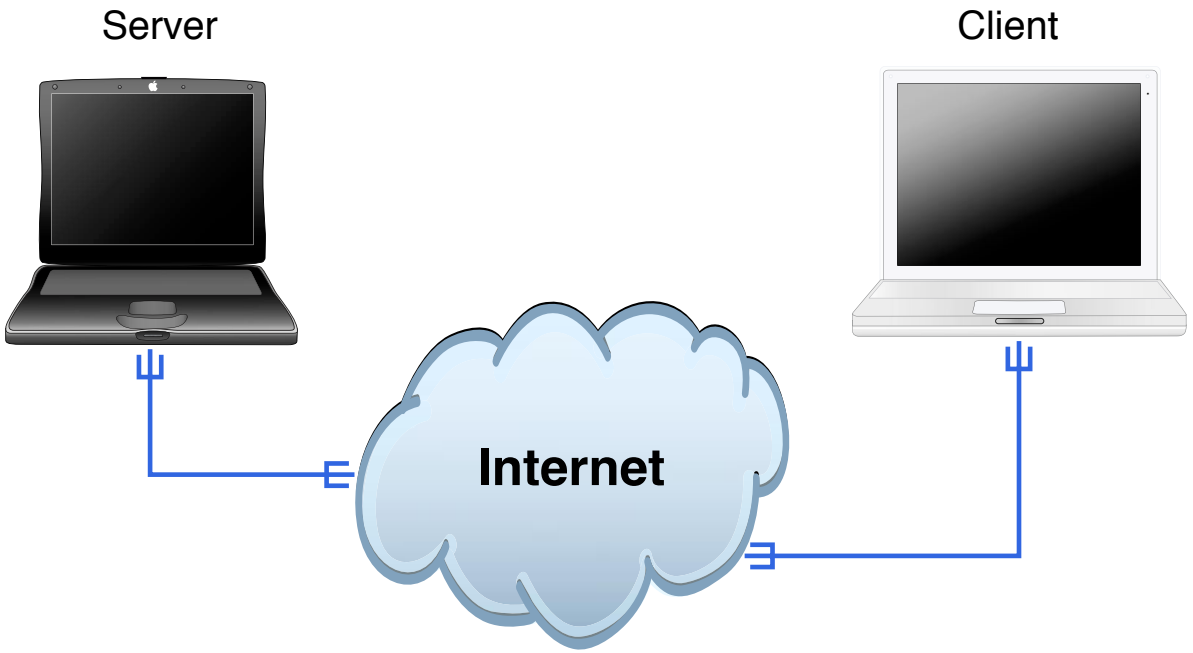
http://www.parallella.org/power-supply/

Demo: Playtime - speed test openssl speed, John speed



Playtime - speed test openssl speed, John speed

Demo: Cain/Abel, hashcat or John the Ripper



Cain/Abel, hashcat or John the Ripper

30-40 minute testing

Grab hashes from `https://hashcat.net/wiki/doku.php?id=example_hashes`

Passwords vælges ikke tilfældigt

Source: `https://wpengine.com/unmasked/`

Encryption key length

Encryption key lengths & hacking feasibility

Type of Attacker	*Budget*	Tool	*Time & Cost/Key 40 bit*	*Time & Cost/Key 56 bit*
Regular User	Minimal $400	Scavenged computer time FPGA	1 week 5 hours ($.08)	Not feasible 38 years ($5,000)
Small Business	$10,000	FPGA [1]	12 min.($.08)	556 days ($5,000)
Corporate Department	$300,000	FPGA ASIC [2]	24 sec. ($.08) 0.18 sec. ($.001)	19 days ($5,000) 3 hours ($38)
Large Corporation	$10M	ASIC	0.005 sec.($0.001)	6 min. ($38)
Intelligence Agency	$300M	ASIC	0.0002 sec.($0.001)	12 sec. ($38)

Source: `http://www.mycrypto.net/encryption/encryption_crack.html`

More up to date:
In 1998, the EFF built Deep Crack for less than $250,000
`https://en.wikipedia.org/wiki/EFF_DES_cracker`
FPGA Based UNIX Crypt Hardware Password Cracker
`http://www.sump.org/projects/password/`

WPA cracking med Pyrit

Pyrit takes a step ahead in attacking WPA-PSK and WPA2-PSK, the protocol that today de-facto protects public WIFI-airspace. The project's goal is to estimate the real-world security provided by these protocols. Pyrit does not provide binary files or wordlists and does not encourage anyone to participate or engage in any harmful activity. **This is a research project, not a cracking tool.**

Pyrit's implementation allows to create massive databases, pre-computing part of the WPA/WPA2-PSK authentication phase in a space-time-tradeoff. The performance gain for real-world-attacks is in the range of three orders of magnitude which urges for re-consideration of the protocol's security. Exploiting the computational power of GPUs, *Pyrit* is currently by far the most powerful attack against one of the world's most used security-protocols.

`http://pyrit.wordpress.com/about/`

Also check out the Reaver brute force WPS
`https://code.google.com/p/reaver-wps/`

Wi-Fi Protected Setup, WPS hacking - Reaver

How Reaver Works Now that you've seen how to use Reaver, let's take a quick overview of how Reaver works. The tool takes advantage of a vulnerability in something called Wi-Fi Protected Setup, or WPS. It's a feature that exists on many routers, intended to provide an easy setup process, and it's tied to a PIN that's hard-coded into the device. Reaver exploits a flaw in these PINs; the result is that, with enough time, it can reveal your WPA or WPA2 password.

Hvad betyder ease of use?

Source:

`https://code.google.com/p/reaver-wps/`

`http://lifehacker.com/5873407/how-to-crack-a-wi+fi-networks-wpa-password-with-reaver`

WPS Design Flaws used by Reaver

Design Flaw #1

Option / Authentication	Physical Access	Web Interface	PIN
Push-button-connect	X		
PIN – Internal Registrar		X	
PIN – External Registrar			X

WPS Options and which kind of authentication they actually use.

As the External Registrar option does not require any kind of authentication apart from providing the PIN, it is potentially vulnerable to brute force attacks.

Pin only, no other means necessary

Source:
`http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf`

WPS Design Flaws used by Reaver

IEEE 802.11/EAP Expanded Type, Vendor ID: WFA (0x372A), Vendor Type: SimpleConfig (0x01)			
M1	Enrollee → Registrar	N1 || Description || PK_E	Diffie-Hellman Key Exchange
M2	Enrollee ← Registrar	N1 || N2 || Description || PK_R || Authenticator	
M3	Enrollee → Registrar	N2 || E-Hash1 || E-Hash2 || Authenticator	
M4	Enrollee ← Registrar	N1 || R-Hash1 || R-Hash2 || $E_{KevWrapKev}$(R-S1) || Authenticator	proove posession of 1st half of PIN
M5	Enrollee → Registrar	N2 || $E_{KeyWrapKey}$(E-S1) || Authenticator	proove posession of 1st half of PIN
M6	Enrollee ← Registrar	N1 || $E_{KeyWrapKey}$(R-S2) || Authenticator	proove posession of 2nd half of PIN
M7	Enrollee → Registrar	N2 || $E_{KeyWrapKey}$(E-S2 ||ConfigData) || Authenticator	proove posession of 2nd half of PIN, send AP configuration
M8	Enrollee ← Registrar	N1 || $E_{KeyWrapKey}$(ConfigData) || Authenticator	set AP configuration

Enrollee = AP Registrar = Supplicant = Client/Attacker PK_E = Diffie-Hellman Public Key Enrollee PK_R = Diffie-Hellman Public Key Registrar Authkey and KeyWrapKey are derived from the Diffie-Hellman shared key. Authenticator = $HMAC_{Authkey}$(last message || current message) $E_{KeyWrapKey}$ = Stuff encrypted with KeyWrapKey (AES-CBC)	PSK1 = first 128 bits of $HMAC_{AuthKey}$(1st half of PIN) PSK2 = first 128 bits of $HMAC_{AuthKey}$(2nd half of PIN) E-S1 = 128 random bits E-S2 = 128 random bits E-Hash1 = $HMAC_{AuthKey}$(E-S1 || PSK1 || PK_E || PK_R) E-Hash2 = $HMAC_{AuthKey}$(E-S2 || PSK2 || PK_E || PK_R) R-S1 = 128 random bits R-S2 = 128 random bits R-Hash1 = $HMAC_{AuthKey}$(R-S1 || PSK1 || PK_E || PK_R) R-Hash2 = $HMAC_{AuthKey}$(R-S2 || PSK2 || PK_E || PK_R)

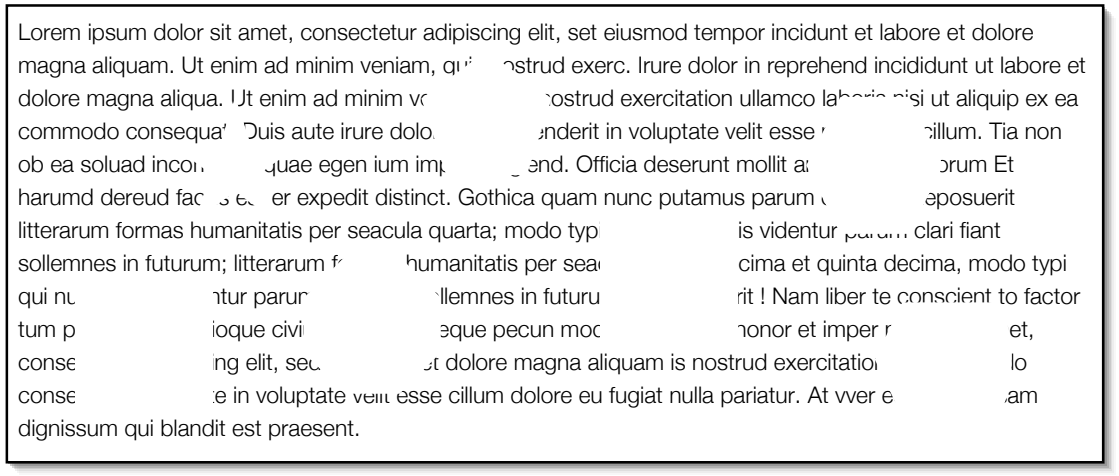
1	2	3	4	5	6	7	0
1st half of PIN				checksum 2nd half of PIN			

Reminds me of NTLM cracking, crack parts independently

Source:

`http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf`

Are your data secure - data at rest



Stolen laptop, tablet, phone - can anybody read your data?

Do you trust ”remote wipe”

How do you in fact wipe data securely off devices, and SSDs?

Encrypt disk and storage devices before using them in the first place!

Circumvent security - single user mode boot

Unix systems often allows boot into singleuser mode
press command-s when booting Mac OS X

Laptops can often be booted using PXE network or CD boot

Mac computers can become a Firewire disk
hold t when booting - firewire target mode

Unrestricted access to un-encrypted data

Moving hard drive to another computer is also easy

Physical access is often - **game over**

Encrypting hard disk

Becoming available in the most popular client operating systems

- Microsoft Windows Bitlocker - requires Ultimate or Enterprise
- Apple Mac OS X - FileVault og FileVault2
- FreeBSD GEOM og GBDE - encryption framework
- Linux LUKS distributions like Ubuntu ask to encrypt home dir during installation
- PGP disk - Pretty Good Privacy - makes a virtuel krypteret disk
- TrueCrypt - similar to PGP disk, a virtual drive with data, cross platform
- Some vendors have BIOS passwords, or disk passwords

Attacks on disk encryption

Firewire, DMA & Windows, Winlockpwn via FireWire
Hit by a Bus: Physical Access Attacks with Firewire Ruxcon 2006

Removing memory from live system - data is not immediately lost, and can be read under some circumstances
Lest We Remember: Cold Boot Attacks on Encryption Keys
`http://citp.princeton.edu/memory/`

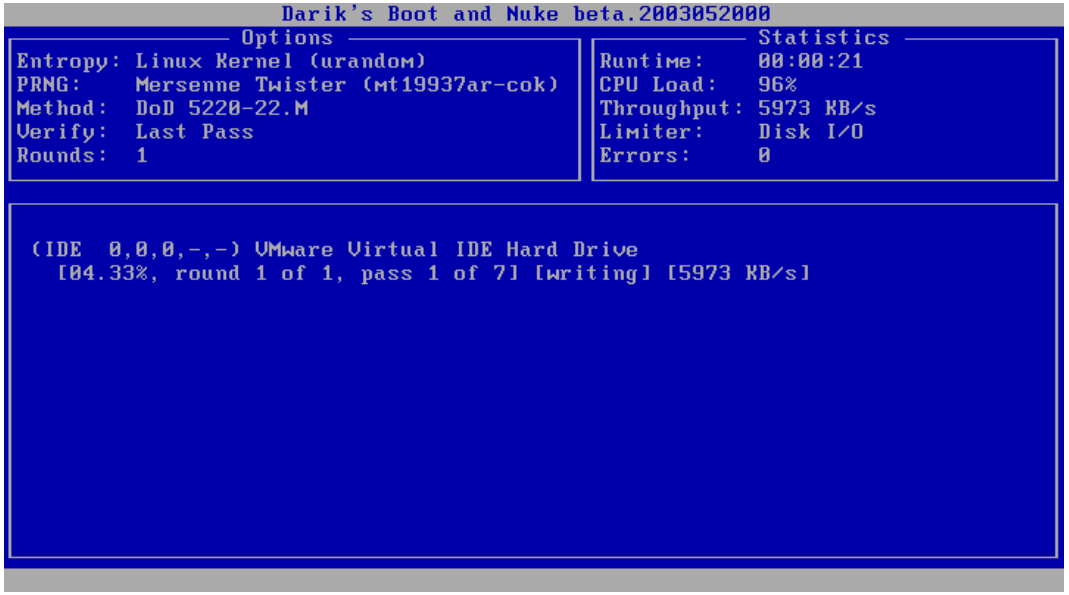
This is very CSI or Hollywoord like - but a real threat

VileFault decrypts encrypted Mac OS X disk image files
`https://code.google.com/p/vilefault/`

FileVault Drive Encryption (FVDE) (or FileVault2) encrypted volumes
`https://code.google.com/p/libfvde/`

So perhaps use both hard drive encryption AND turn off computer after use?

... and deleting data



Getting rid of data from old devices is a pain

Some tools will not overwrite data, leaving it vulnerable to recovery

Even secure erase programs might not work on SSD - due to reallocation of blocks

I have used Darik's Boot and Nuke ("DBAN") `http://www.dban.org/`

Kom igang!

- Skriv på DVD - DVD brændere i mange laptops idag
- Gem på netværket - Dropbox, husk en yderligere backup!
- Brug Duplicity på egen server, eller tilsvarende services

Mat Honan epic hacking :-(

`http://www.wired.com/gadgetlab/2012/08/apple-amazon-mat-honan-hacking/all/`

Ransomware er hot topic i 2015 :-(

Duplicity

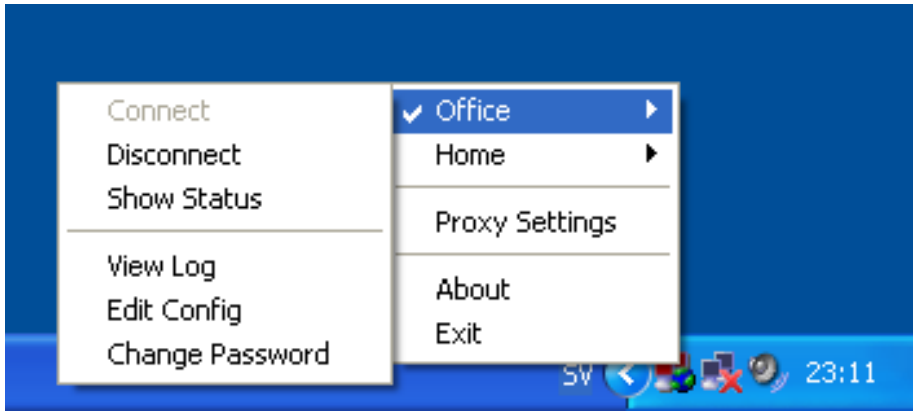
What is it?

Duplicity backs directories by producing encrypted tar-format volumes and uploading them to a remote or local file server. Because duplicity uses librsync, the incremental archives are space efficient and only record the parts of files that have changed since the last backup. Because duplicity uses **GnuPG** to encrypt and/or sign these archives, they will be safe from spying and/or modification by the server.

`http://duplicity.nongnu.org/` duplicity home page

`http://www.gnupg.org/` The GNU Privacy Guard

VPN



Virtual Private Networks are useful - or even required when travelling

VPN `http://en.wikipedia.org/wiki/Virtual_private_network`

SSL/TLS VPN - Multiple incompatible vendors: OpenVPN, Cisco, Juniper, F5 Big IP

IETF IPsec does work cross-vendors - sometimes, and is also increasingly becoming blocked or unusable due to NAT :-(

Recommended starting point OpenVPN - free and open, clients for ”anything”

IPsec IKE-SCAN

Scan IPs for VPN endpoints with ike-scan:

```
root@kali:~# ike-scan 91.102.91.30
Starting ike-scan 1.9 with 1 hosts
(http://www.nta-monitor.com/tools/ike-scan/)
91.102.91.30 Notify message 14 (NO-PROPOSAL-CHOSEN)
HDR=(CKY-R=f0d6043badb2b7bc, msgid=f97a7508)

Ending ike-scan 1.9: 1 hosts scanned in 1.238 seconds (0.81 hosts/sec).
0 returned handshake; 1 returned notify
```

Source:

`http://www.nta-monitor.com/tools-resources/security-tools/ike-scan`

crack IKE psk?

`http://ikecrack.sourceforge.net/`

`https://www.trustwave.com/Resources/SpiderLabs-Blog/Cracking-IKE-M/`

ike-scan network scanning

```
hlk@cornerstone03:~$ sudo ike-scan -M 91.102.91.0/24
Starting ike-scan 1.9 with 256 hosts
(http://www.nta-monitor.com/tools/ike-scan/)
91.102.91.14 Notify message 14 (NO-PROPOSAL-CHOSEN)
HDR=(CKY-R=94dd41cf44da082b, msgid=602c35c1)
91.102.91.30 Notify message 14 (NO-PROPOSAL-CHOSEN)
HDR=(CKY-R=e21e89d16f898aa5, msgid=ff41d51c)
91.102.91.70 Notify message 14 (NO-PROPOSAL-CHOSEN)
HDR=(CKY-R=e882d9b4477b847b, msgid=55be4339)
91.102.91.78 Notify message 14 (NO-PROPOSAL-CHOSEN)
HDR=(CKY-R=1fc54d8c3042daa3, msgid=ea705f39)
91.102.91.150 Notify message 14 (NO-PROPOSAL-CHOSEN)
HDR=(CKY-R=d5470f881de6d2d9, msgid=2bf5f5ef)
91.102.91.158 Notify message 14 (NO-PROPOSAL-CHOSEN)
HDR=(CKY-R=9f7af04bcb0152a9, msgid=44f26f01)

Ending ike-scan 1.9: 256 hosts scanned in 40.465 seconds (6.33 hosts/sec).
0 returned handshake; 6 returned notify
```

Multiple browsers

- Strict Security settings in the general browser, Firefox or Chrome?
- More lax security settings for ”trusted sites- like home banking
- Security plugins like HTTPS Everywhere and NoScripts for generic browsing

HTTPS Everywhere

HTTPS Everywhere is a Firefox extension produced as a collaboration between The Tor Project and the Electronic Frontier Foundation. It encrypts your communications with a number of major websites.

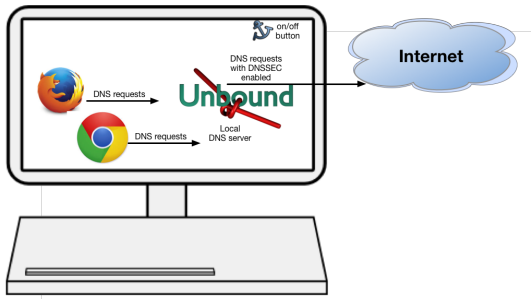
`http://www.eff.org/https-everywhere`

Playtime - metasploitable

Kør egen demo hele vejen.

- Start Kali
- Start en Metasploitable
- Hack Metasploitable, Hail Mary er fint ☺
- Find /etc/shadow filen
- Kopier denne over til jeres Kali, copy/paste burde virke
- Crack denne med John The Ripper, burde finde nogle passwords
  Ihvertfald msfadmin/msfadmin

DNSSEC trigger



Lots of DNSSEC tools, I recommend DNSSEC-trigger a local name server for your laptop

- DNSSEC Validator for firefox
  `https://addons.mozilla.org/en-us/firefox/addon/dnssec-validator/`
- OARC tools `https://www.dns-oarc.net/oarc/services/odvr`
- `http://www.nlnetlabs.nl/projects/dnssec-trigger/`

DNSSEC NSEC walk the zone

DNSSEC:NSEC vs. NSEC3

The Domain Name System Security Extensions(DNSSEC) provide two different records for securely handling non-existent names in DNS, NSEC and NSEC3. They are mutually exclusive, so operators need to pick one when deploying DNSSEC.

The problem both NSEC and NSEC3 solve is knowing when a name exists within a given zone. This is required to prevent malicious actors from sending fake negative responses to queries.

... the challenge with the plain NSEC record is that someone could use the NSEC responses to “walk the zone” and build a list of all of the records in a DNS zone.

Source:

`http://www.internetsociety.org/deploy360/resources/dnssec-nsec-vs-nsec3/`

Perhaps try `http://josefsson.org/walker/`

DANE

Objective:

Specify mechanisms and techniques that allow Internet applications to establish cryptographically secured communications by using information distributed through DNSSEC for discovering and authenticating public keys which are associated with a service located at a domain name.

DNS-based Authentication of Named Entities (dane)

`https://datatracker.ietf.org/wg/dane/charter/`

`http://googleonlinesecurity.blogspot.dk/2011/04/improving-ssl-certificate-security.html`

DNSSEC er ved at være godt udbredt - undtagen i DK

(findes på .dk zonen, men næsten ingen resolvere)

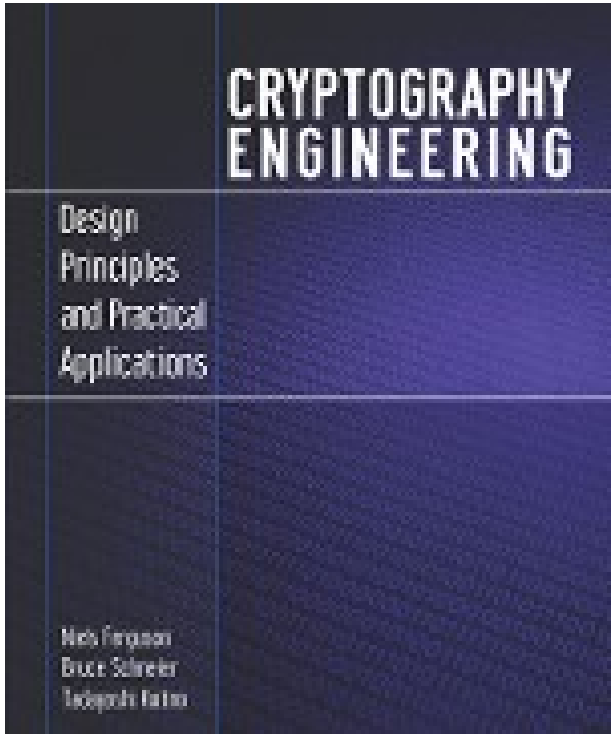
www.uncensoreddns.org

Konklusion: Kryptografi er svært

Åbent kursus på Stanford

`http://crypto-class.org/`

Kryptering: Cryptography Engineering



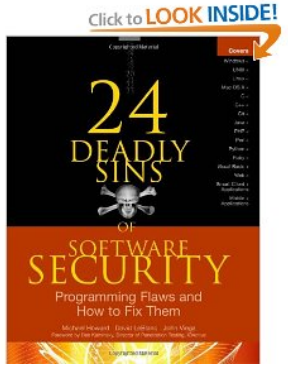
Cryptography Engineering by Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno

`https://www.schneier.com/book-ce.html`

Kryptering sikrer fortrolighed og integritet af beskederne

24 Deadly Sins of Software Security

24 Deadly Sins of Software Security af Michael Howard, David Leblanc, John Viega 2009



Obligatorisk læsning for alle udviklere

Denne bog er præcis og giver overblik på kun 432 sider

Buffer Overruns, Format String Problems, Integer Overflows, SQL Injection, Command Injection, Failing to Handle Errors, Cross-Site Scripting, Failing to Protect Network Traffic, Magic URLs Hidden Form Fields, Improper Use of SSL and TLS, Weak Password-Based Systems, Failing to Store and Protect Data Securely, Information Leakage, Improper File Access, Trusting Network Name Resolution, Race Conditions, Unauthenticated Key Exchange, Cryptographically Strong Random Numbers, Poor Usability

Open Mike night ...

Hvad glemte jeg? Kom med dine favoritter ☺

evalg, DNS censur, NemID bashing, malware sucks, Android malware, iPhone malware?

Questions?

Henrik Lund Kramshøj hlk@zencurity.dk

Need DDoS testing or pentest, ask me!

You are always welcome to send me questions later via email

Did you notice how a lot of the links in this presentation use HTTPS - encrypted