What's this about?

- A discussion about modern disk encryption systems

Who am I?

- Jacob Appelbaum
  - Some guy with great hair from Sunny Warm Never Foggy San Francisco, California, Freedom Land USA
  - Come visit! It's fun to get finger printed and photographed!
    - You dirty criminal! Give us your papers!

What's this all about?

- I'm going to discuss different disk encryption systems in their current implementation, the users rights in their given country (ie: the USA, the UK, and Germany), issues with the implementations, commentary on the community surrounding each featured implementation, threats posed by legal systems, requirements for users, as well as ideas for working around the letter of the law.

What's our supposed threat model?

- We're protecting against offline attacks
  - Laptop theft
  - Hard drive theft
  - umounted partitions for online systems
- We live in a “Free Country”
  - Do you really?
    - Know your rights!
- You have the right to remain silent.
- ...
  - You're sure about that?
    - Are you *really sure*?

What if we have other threats?

- You're in a bad place, seek help
  - We'll get there shortly, hold on tight

What rights do I have?

- I'm not a lawyer. Please consult one.
  - With that said here's an idea of the mine field ahead

The United Kingdom

- The RIP act (Regulation of Investigatory Powers Act 2000)
  - Really bad news for the British
  - http://en.wikipedia.org/wiki/Regulation_of_Investigatory
  - Not entirely in effect yet
    - Part III of the Act
      - Two years in jail for failure to comply
- What we can learn from the UK
  - How to be a modern police state in the Western World

The U.S.A.

- Criminal cases
  - Fifth amendment protections (Since 1791)
    - http://en.wikipedia.org/wiki/Fifth_Amendment_to_the_U
    - The fifth amendment: *"No person shall be held to answer for a capital, or otherwise infamous crime, unless on a presentment or indictment of a Grand Jury, except in cases arising in the land or naval forces, or in the Militia, when in actual service in time of War or public danger; nor shall any person be subject for the same offence to be twice put in jeopardy of life or limb;* ***nor shall be compelled in any criminal case to be a witness against himself****, nor be deprived of life, liberty, or property, without due process of law; nor shall private property be taken for public use, without just compensation."*

Germany

- Safe from coercion?

What's important for an implementation?

- Encryption containers in the form of files and/or (pseudo/real) devices. These containers should act as native devices for normal file systems.
- Reasonable passwords aren't just allowed, they're required
- User known passwords are transformed (read: hashed) or discarded after use (key abstraction)
- Password salting or iteration, reducing precomputed dictionary attacks
- Password changing without re-encryption of

What's important for an implementation?

- Encrypted swap partitions/files
- Your choice in symmetric ciphers (AES, blowfish, etc)
- Multiple key support
- Avoiding kernel memory lock ups
- Flipping bits in memory, keeping keys safe or destroyed
- Resistance to known attacks and obvious attacks
  - Generic dictionary
  - Plain text attacks against know file system attributes
  - Other implementation issues [...]

What's nice?

- A deniable decryption process
  - RubberhoseFS
    - Is this right for you?
- Provable impossibilities
  - M.A.I.D.
    - I'll get to this in a bit
- Knowing what you're doing
  - Pay attention to the entire system
    - Are you sure you did that right? (/tmp, /var/tmp, printer spools, logs, etc)
      - Is everything taken care of?
- A fast implementation

Who attempts to solve some of these problems?

- OpenBSD
- NetBSD
- FreeBSD
- Crypto loop in Linux
- DM Crypt in Linux
- Loop-AES in Linux
- File Vault in Mac OS X
- Others... (We're not going to talk about PGP disk, ncrypt, etc)

Who pretends to solve it and actually screws you?

- Why that's just about everyone!
  - (Apple really *really* fucks it up though!)

OpenBSD

- Encrypted swap through sysctl vm.swapencrypt.enable=1
  - Good!
- vnconfig
  - Not modular or very extensible
    - Blowfish cipher only (You're limited in your choices if everything breaks)
    - Interactive entry of password
      - Possible to supply a file
- Some of our requirements
  - Currently missing really usable key abstraction, multi-key modes, choice of ciphers
- FL/OSS (It will be improved)

NetBSD

- CDG
  - Really well thought out
    - These guys are smart (Really smart)
    - Very well written, small and compact
- All of our requirements
  - Very flexible (AES, blowfish, 3DES)
  - Sector by sector encrypting
  - Password transformation with PKCS#5 PBKDF2 (salted iterated hashing)
  - N-Factor authentication!
  - Keys in a file
- Again, these people know what they're doing

FreeBSD (GBDE)

- GBDE (GEOM Based Disk Encryption) - A bit complicated and strange
  - Written by a very smart fellow
  - Some strange mistakes?
  - It doesn't seem broken out right
  - Pass phrase and “lock file” (16 Bytes)
    - Two factor if split correctly, either stored in a file or on the raw disk.
    - Cannot decrypt unless you have both
  - Meets most of our requirements, has some issues (Passphrase, not key file)
  - Major issues with mounting /

FreeBSD (GELI)

- Builds on the ideas of GBDE
- Allows for different cryptography algorithms
- Allows for key file
- Issues with mounting / (no key file allowed)
- Similar issues to GBDE

Cryptoloop (The old Kerneli)

- Do ***not*** use this
- Does not support most of our requirements
- Vulnerable to known attacks
  - Known plaintext attacks again predictable file system attributes, never fixed (ie: rainbow table possible)
- FL/OSS
  - So you can see how poorly it's done

DM Crypt

- On disk format is the same as cryptoloop (?!)
- Does support most of our requirements
- Vulnerable to similar attacks as Cryptoloop?
- Somehow the new standard for Linux disk crypto
  - Why?
- FL/OSS

Loop-AES (Linux 2.2,2.4,2.6)

- Disclaimer: Sorta Involved but hardly
- Supports all of our requirements in Multi-key-mode-v3
- Issues with deadlocks when using journaling file system on file backed loops
  - Don't do that
- Key abstraction, salting, different hashes, your choice in cipher
- Not stock in Kernel (But works with a patch for loop.c in Linux 2.0-2.6 and a patch for util-linux)
- FL/OSS

Mac OS X File Vault

- “Unbreakable!”

Mac OS X File Vault

- Marking nonsense:
  - According to apple:
  - "At home and away, keep your valuable documents safe with powerful AES-128 encryption. FileVault automatically encrypts and decrypts the contents of your home directory on the fly. Real security comes from ***knowing nobody*** can rifle through your files without ***your permission***. FileVault uses the latest government security standard to safeguard your hard work. FileVault protects all the info in your home folder from prying eyes, so your ***trade secrets stay secret.***"
    " Eternal protection. AES gives you 3.4 x 1038

File ~~V~~(F)ault

- Supports few of our requirements
  - Depending on how it's used, it supports even less by default
- Source code for the entire system? No?
  - No.
- Flawed implementation
  - Lets all have a laugh at Apple now

File ~~V~~(F)ault

- Unbreakable!
  - Ha!
- Simple effective attacks:
  - Plain text password recovery
    - strings -8 /var/vm/swapfile* | grep -B2 -A2 "/System/Library/CoreServices/DiskImageMounter.app"
  - Dig around
    - You'll find file lists, encrypted container names, passwords, everything
- Mitigation?
  - Sure, 10.4 allows for encrypted swap
    - Do you trust your data with a bandaid?

Others

- CFS
  - Thanks to Matt Blaze for paving the way
  - An implementation as a userland NFS server
  - Well done but showing age
    - It was a nice run (DES, 3DES, Blowfish, etc)
    - slow and abandoned
- TCFS
  - Abandoned project (last patch for linux 2.0 kernel?)
    - Interesting ideas, poor implementation
    - Apparently impossible to contact the developers
    - Key management, what key management?

Others

- Rubberhose File System
  - Steganographic
  - Is this the wrong assumption for the world of today?
- Other commercial software
  - Not for me thanks. Use at your own risk.

Choose wisely

- It's up to you, do lots of research

An idea to extend your freedoms

- What if you could have a system that builds on the strengths of other tools?
- What if you could keep your data secure even in the event of a seizure by The Law?
- What if you could comply with laws that *should* compromise your data?
- What if it kept you out of jail for the contents of your encrypted containers?
- What if it was possible to demonstrate this?
- What if you couldn't even sell yourself out after a certain point?

An idea to extend your freedoms

- What if you could have a system that builds on the strengths of other tools?
- What if you could keep your data secure even in the event of a seizure by The Law?
- What if you could comply with laws that *should* compromise your data?
- What if it kept you out of jail for the contents of your encrypted containers?
- What if it was possible to demonstrate this?
- What if you couldn't even sell yourself out after a certain point?

Enter your M.A.I.D.

- Mutually Assured Information Destruction
  - A concept for a framework
    - Write your own system
  - Easy to implement in a few lines of ruby, C, Lisp or Linenoise (perl)
    - Make sure it's easy for the forensic guys to understand, they'll be the ones saving you through their inability to decrypt your data.

M.A.I.D.

- As an example keep these objects in mind:
  - Your encrypted containers
    - Devices and files
  - Authentication token(s)
    - An ssh key
      - This is tied to the password the user knows
  - Encrypted key list(s)
    - A text file encrypted to a given GPG key
      - The user never sees these passwords, they're as random as possible
  - A network to provide a connection
    - Tunnel your traffic to avoid obvious network monitoring
      - Tor has been suggested, tor has issues of it's own, your choice of implementation, your choice

M.A.I.D.

- Remote server(s) to provide services
  - Decryption of previously mentioned text file
  - In a country that's “safe” for a given amount of time
- A safe threshold of time
  - This is the amount of time you can remain silent

M.A.I.D.

- Put it together:
  - Authenticate, connect to the server (if your threshold hasn't been reached), send your key file for decryption, the returned data is never seen by the user, the device is mounted.
  - If the threshold passes, your decrypting service is revoked, keys are destroyed and the server returns an error stating so
- What's added to the threat model?
  - You're worried about arrest and physical harm
  - You will be unable to exonerate yourself

M.A.I.D.?

- Why, M.A.I.D.?
  - Who else cleans up after you when you're not around? M.A.I.D. does it all!
    - Assuming you don't screw up the implementation
    - Also assuming that you really live in a Free Country
      - Hope you're not the test case, gitmo's cold!
  - You can prove you're unable to decrypt your data
    - If you didn't keep any backups or no backups were found...
  - You can comply with The Law
    - Give up your encrypted data, keys, pass phrases, authentication tokens, server IPs
    - Perhaps not in the spirit of such unjust laws, who cares?
  - Unless your containers implementation is

What's wrong with M.A.I.D.?

- It's complex in weird ways
- It's tied to a network
- It's easy to screw up
- The server is a single point of failure
  - Split the key up
- The sever might compromise you with backups
- It's brittle if you're not careful

Final notes on ideas for the M.A.I.D. frame work

- Take the basic ideas and use them to protect yourself
- Make backups and hope no one ever finds them
- Changing your threshold after you're under a search warrant might just land you in jail
  - It depends on what the attacker knows you know
- Don't give up hope
- Open the source for your implementation. Free Software.
- Lots of issues (underlying system vulnerabilities, server issues, etc)

Thanks, Q & A, End

- Thanks to Daniel Berg, Christian Fromme and other awesome German hackers.
- Contact information:
- Jacob Appelbaum <jacob@appelbaum.net>