
MAKING
SECURITY
SHINY

Tom Webster

"SamuraiLink3"

SamuraiLink3.com

SamuraiLink3@gmail.com

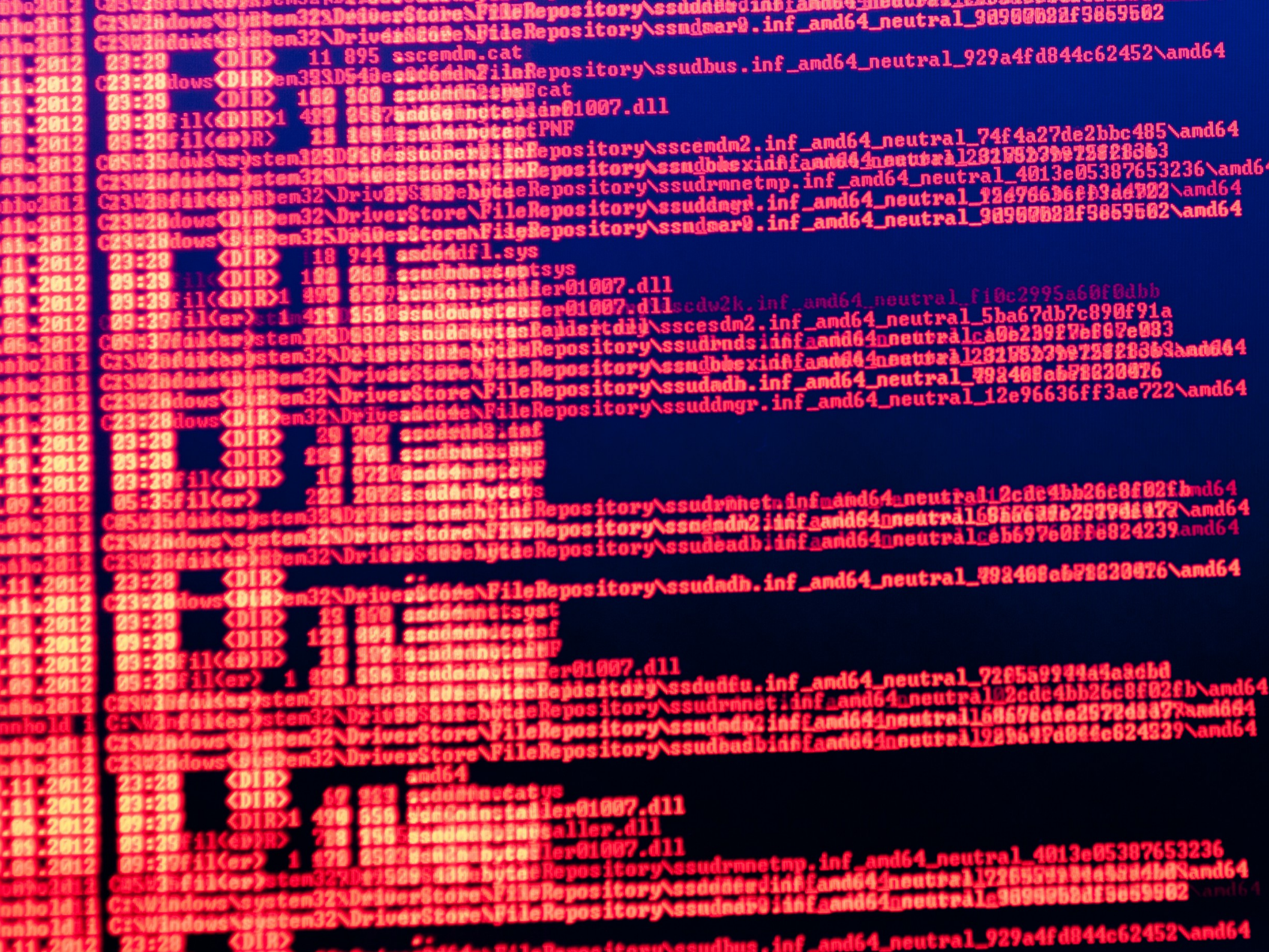
Gplus.to/samurailink3

@SamuraiLink3

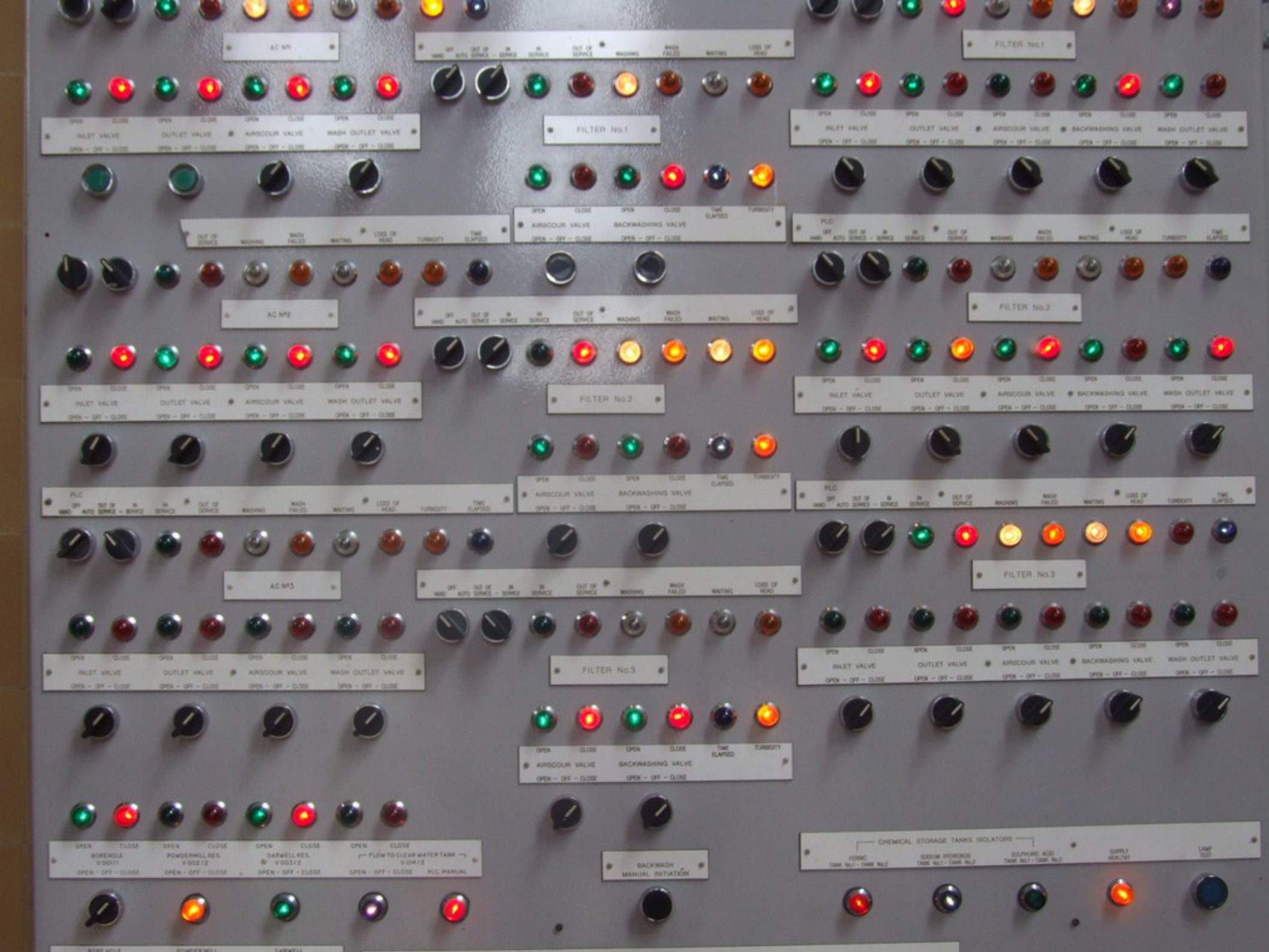
DISCLAIMER

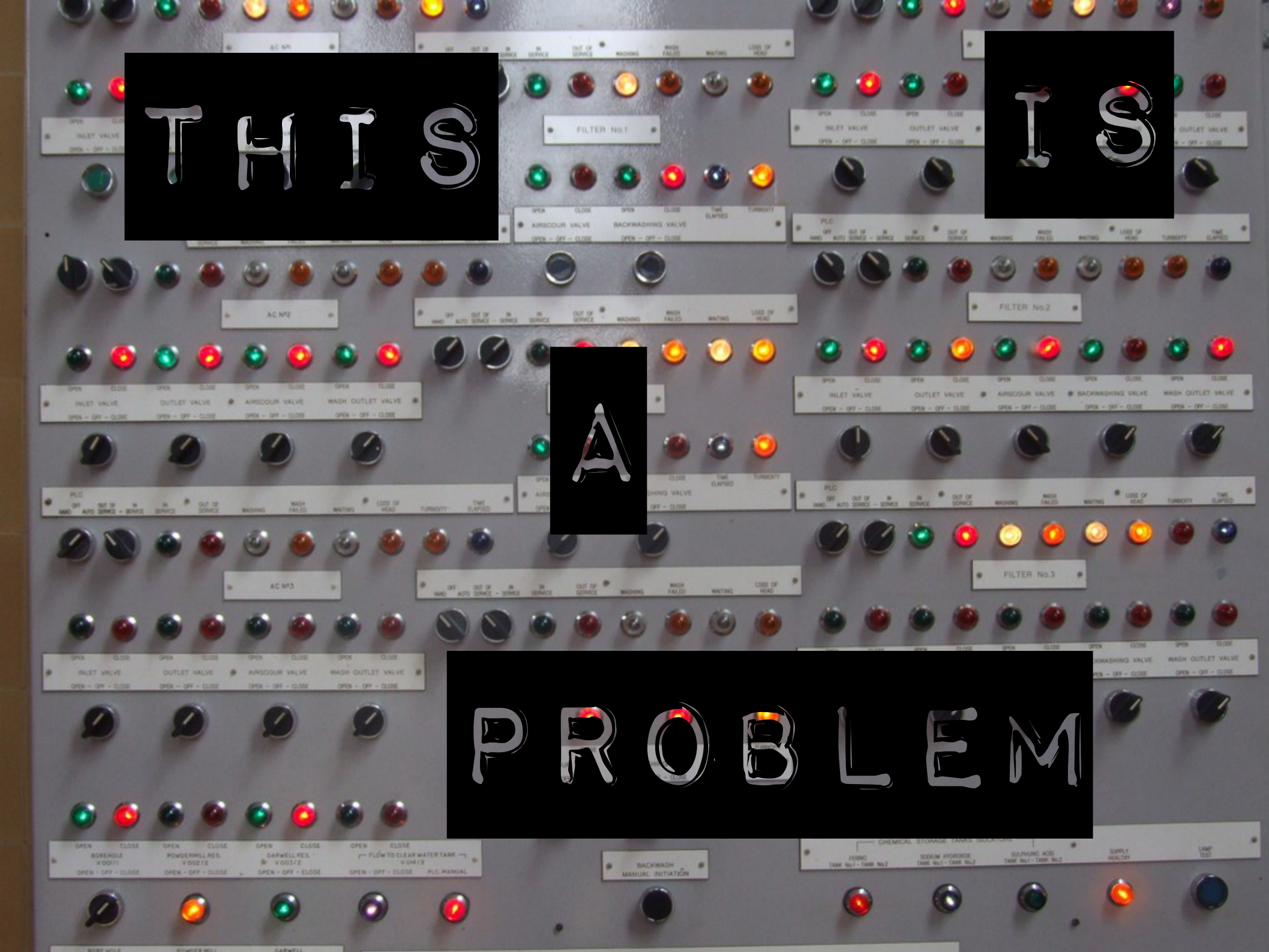
This isn't about the security of the following products, just how the interface and user experience is implemented. Certain providers in the list have had major demonstrated downfalls, attacks, and general laziness when it comes to implementing good crypto, they are here to serve as an example of how to provide a user interface, nothing more. Don't hurt me.

Most security programs look like this:

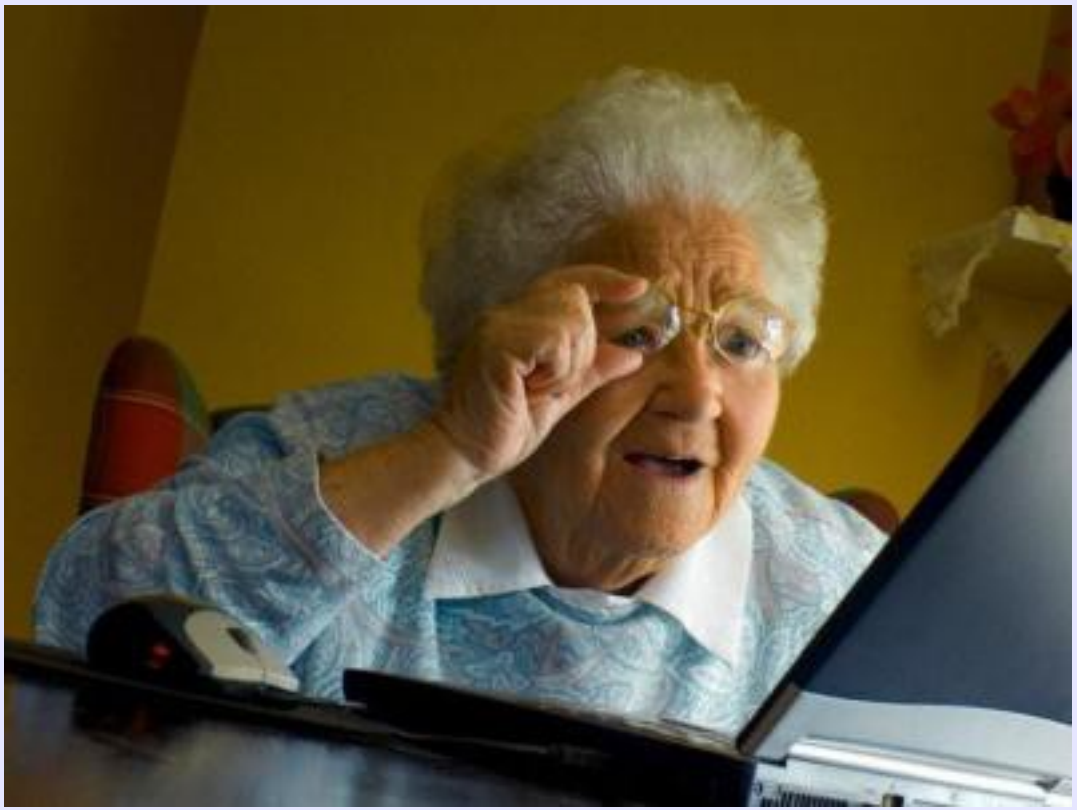


Or this:


AC N°1
FILTER No.1
INLET VALVE
OUTLET VALVE
AIRSCOUR VALVE
WASH OUTLET VALVE
BACKWASHING VALVE
PLC
AC N°2
FILTER No.2
AC N°3
FILTER No.3
BOREHOLE V 001/1
POWDERMILL RES. V 002/2
DARWELL RES. V 003/2
FLOW TO CLEAR WATER TANK V 014/2
BACKWASH MANUAL INITIATION
CHEMICAL STORAGE TANKS ISOLATORS
FERRIC
SODIUM HYDROXIDE
SULPHURIC ACID
SUPPLY HEALTHY
LAMP TEST


THIS
IS
A
PROBLEM

WHY DO WE CARE?



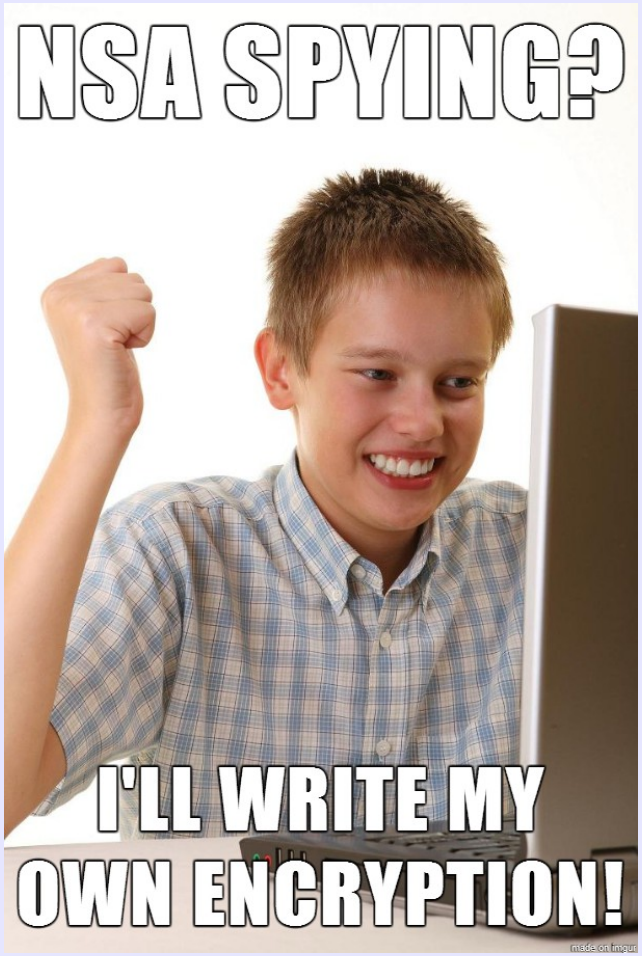
With the advent of the Snowden leaks, people are finally starting to take security seriously.

We're about to see a major influx of 'security' apps/programs/sites/oils that a pretty to look at, easy to use, and, best of all, utterly ineffective or malicious.

People are going to create 'security' applications and market them to the general public like we've never seen before.

WHY DO WE CARE?

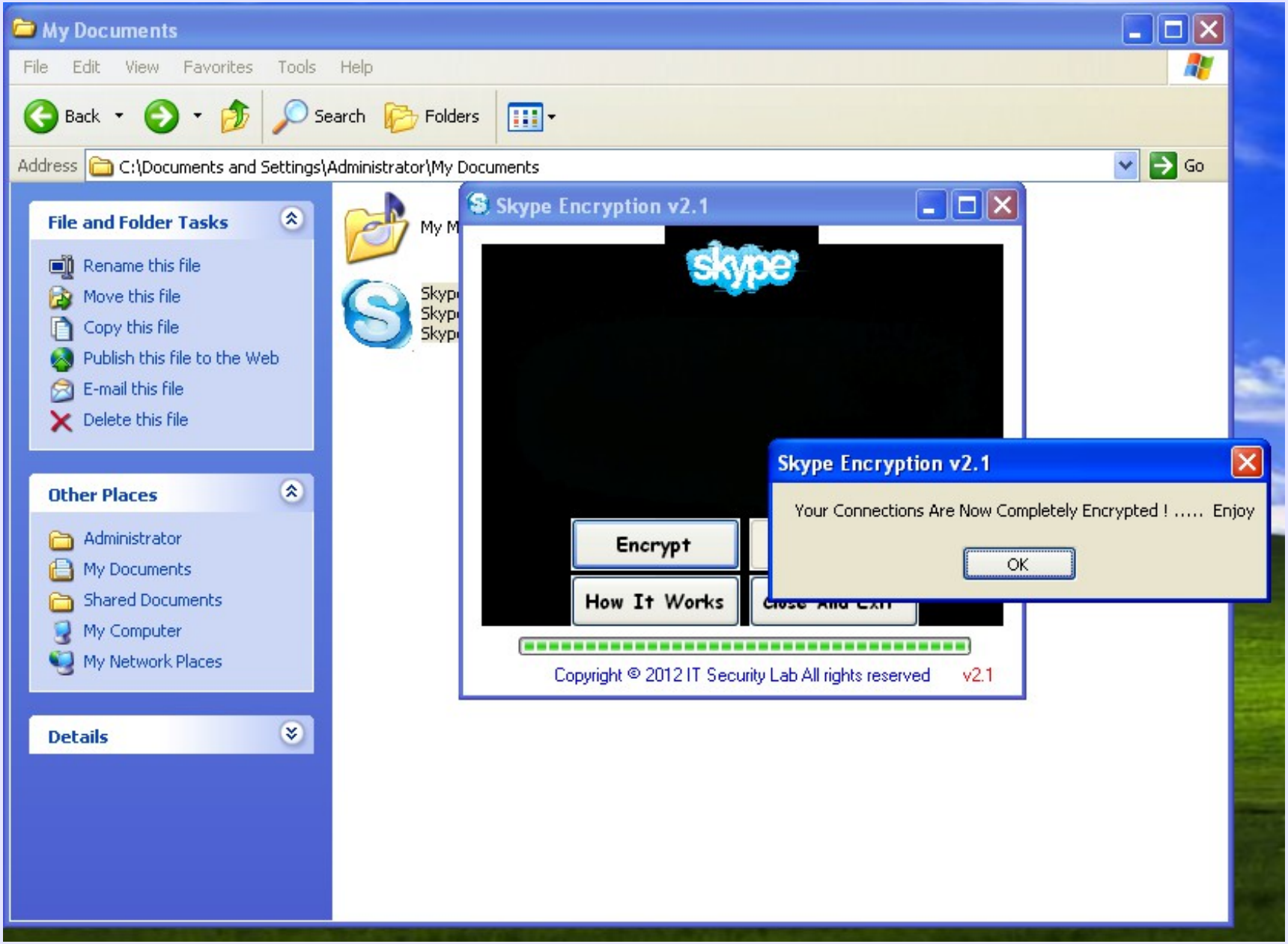
Including these guys:



Reality: These guys will sell and give away a lot of software to our mothers and co-workers.

WHY DO WE CARE?

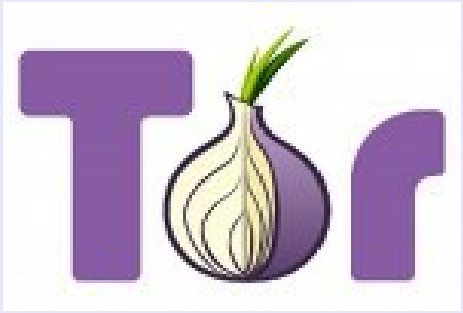
This is already happening:

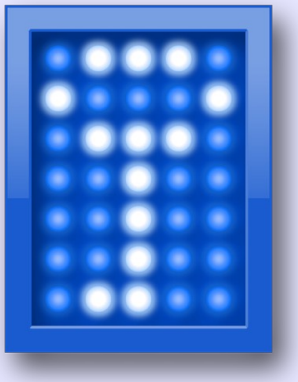


https://www.eff.org/deeplinks/2012/05/fake-skype-encryption-tool-targeted-syrian-activists-promises-security-delivers

WHY DO WE CARE?

Let's make sure the real champions stay on top:





PGP®

WHAT CAN WE DO?


Union
NCH AND ATM

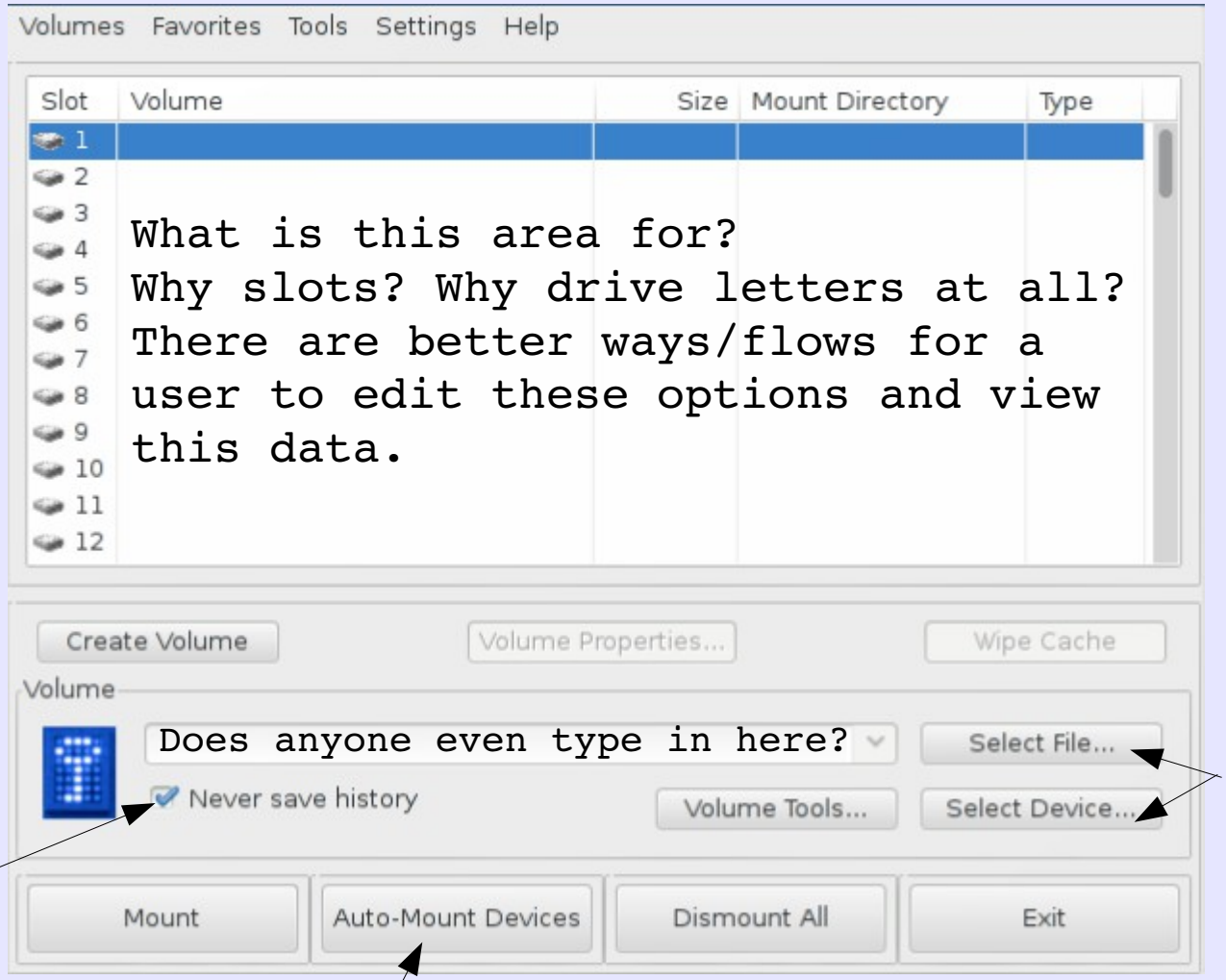
MAKE


MAKE
SECURITY


MAKE
SECURITY
SHINY

EXAMPLES

TrueCrypt



Why is this even an option?

This should be a hidden option.

Redundant: This choice should present itself after 'Mount' is clicked.

EXAMPLES

TrueCrypt

Errata:

Bad:

- Very text heavy
- Language, language, language.
- In your applications:
  Talk to people, not security auditors
- In your documentation:
  Talk to security auditors, user's don't read anything anyway...
- 'Easy buttons' don't exist.

Good:

- Wizard for volume creation is good.

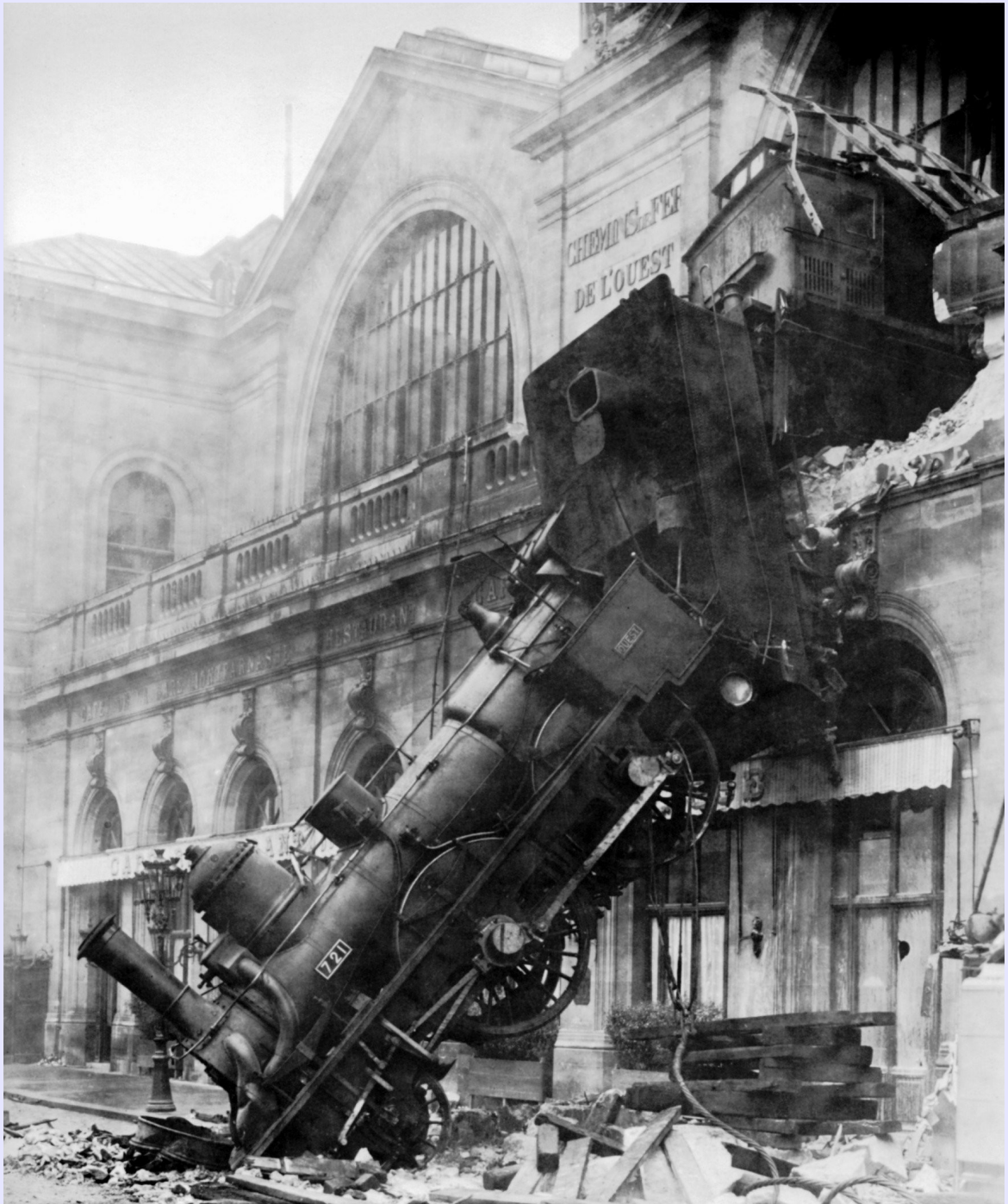
LESSONS LEARNED

1. BE UP FRONT

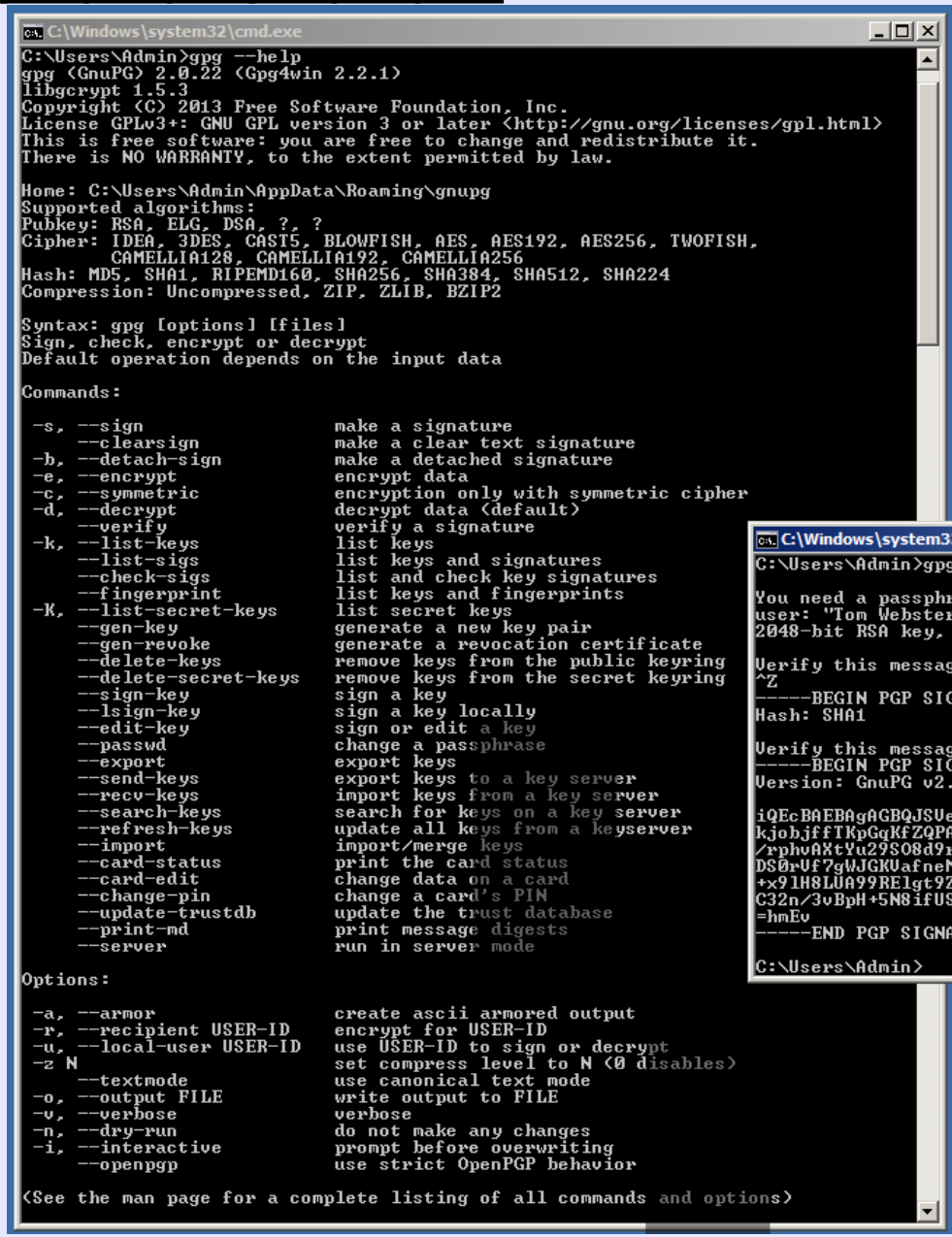
2. USE PLAIN LANGUAGE

EXAMPLES

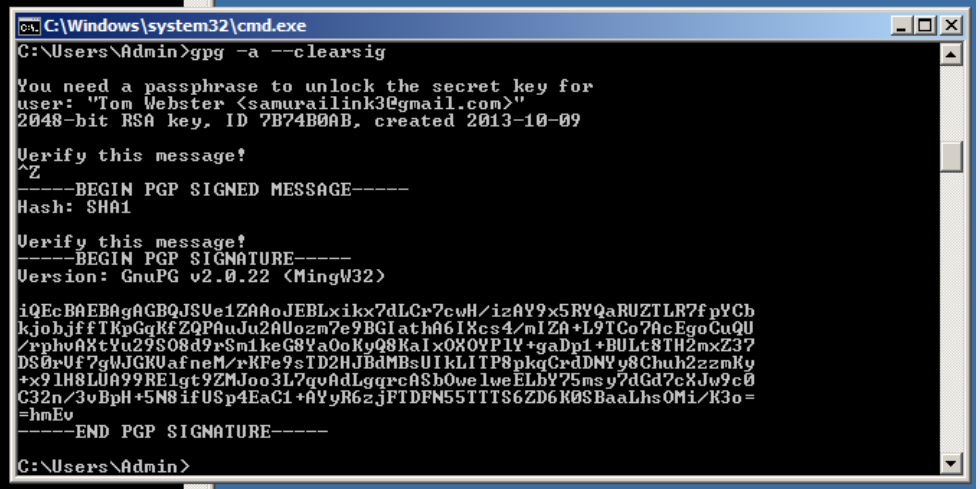
GPG



EXAMPLES



You're making this too easy...



EXAMPLES

Errata:

Bad:

- Command Line
- Command Line
- Command Line

Good:

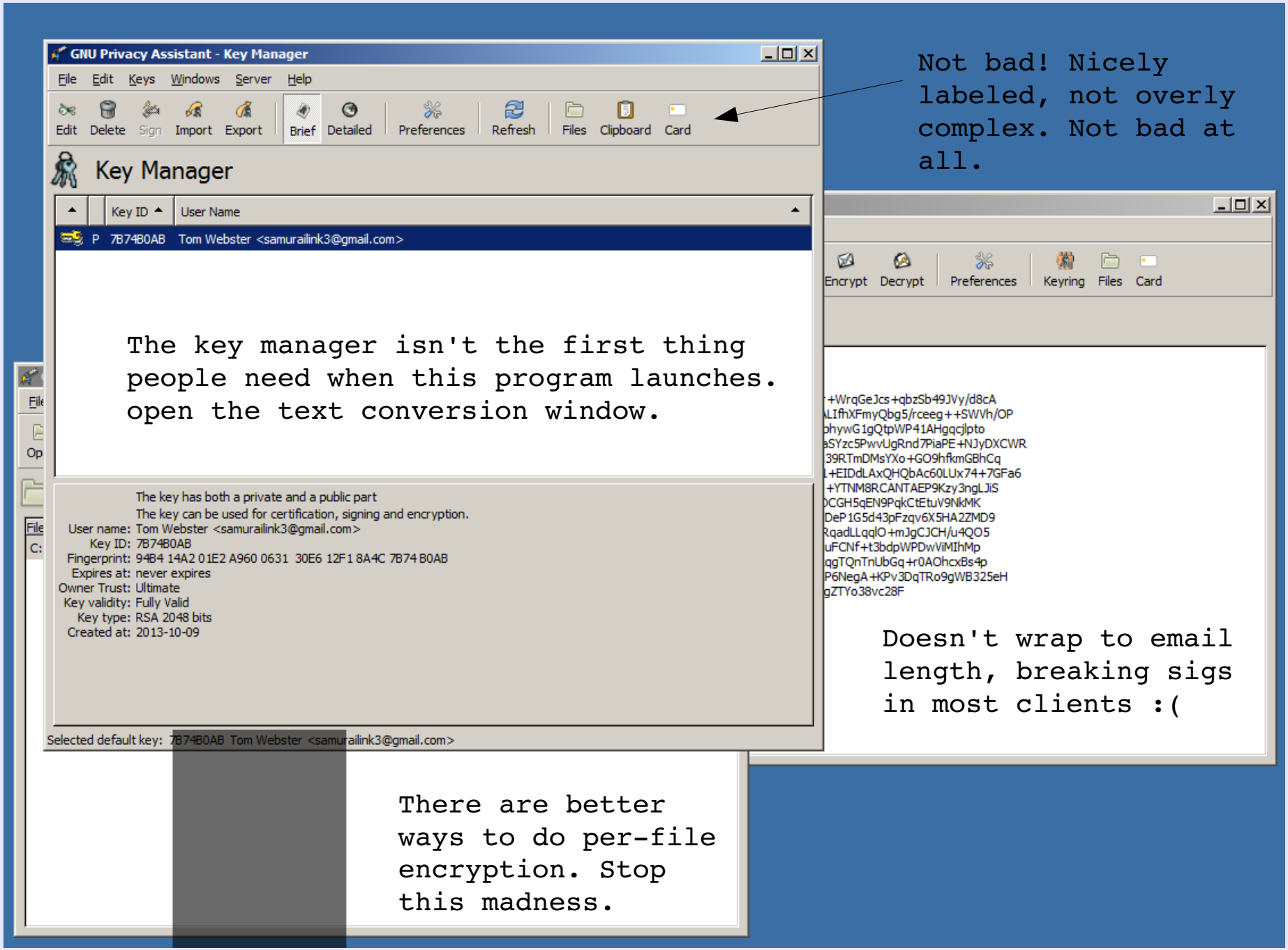
- Nothing. At all. Ever. Stop it.

Honestly, I use bash daily and I love my terminal, but normal people don't stand for this.

EXAMPLES

GPA

A 'little' better...



EXAMPLES

Errata:

Bad:

- Looks like it's from 1996
- File encryption area confuses users and offers no explanations at all
- Key-manager-as-default puts off users who just want to send a secure message
- If you think this will copy/paste into an email client nicely, you're gonna have a bad time.

Good:

- The buttons up top are nice

LESSONS LEARNED

3. PEOPLE USE SOFTWARE, DEVELOPERS USE LIBRARIES

EXAMPLES

SpiderOak

This one needs some conversation.

EXAMPLES

SpiderOak

Errata:

Good:

- It looks pretty good
- It looks easy to use

Bad:

- It's a damn liar
- Uses confusing wording when setting up syncs or shares
- Offers little to no explanation of features in application
- You actually have to read the documentation on their site.

  Like some sort of savage.

LESSONS LEARNED

4. NEVER BETRAY EXPECTATIONS

5. DON'T TEACH UNLESS IT OFFERS A CLEAR BENEFIT

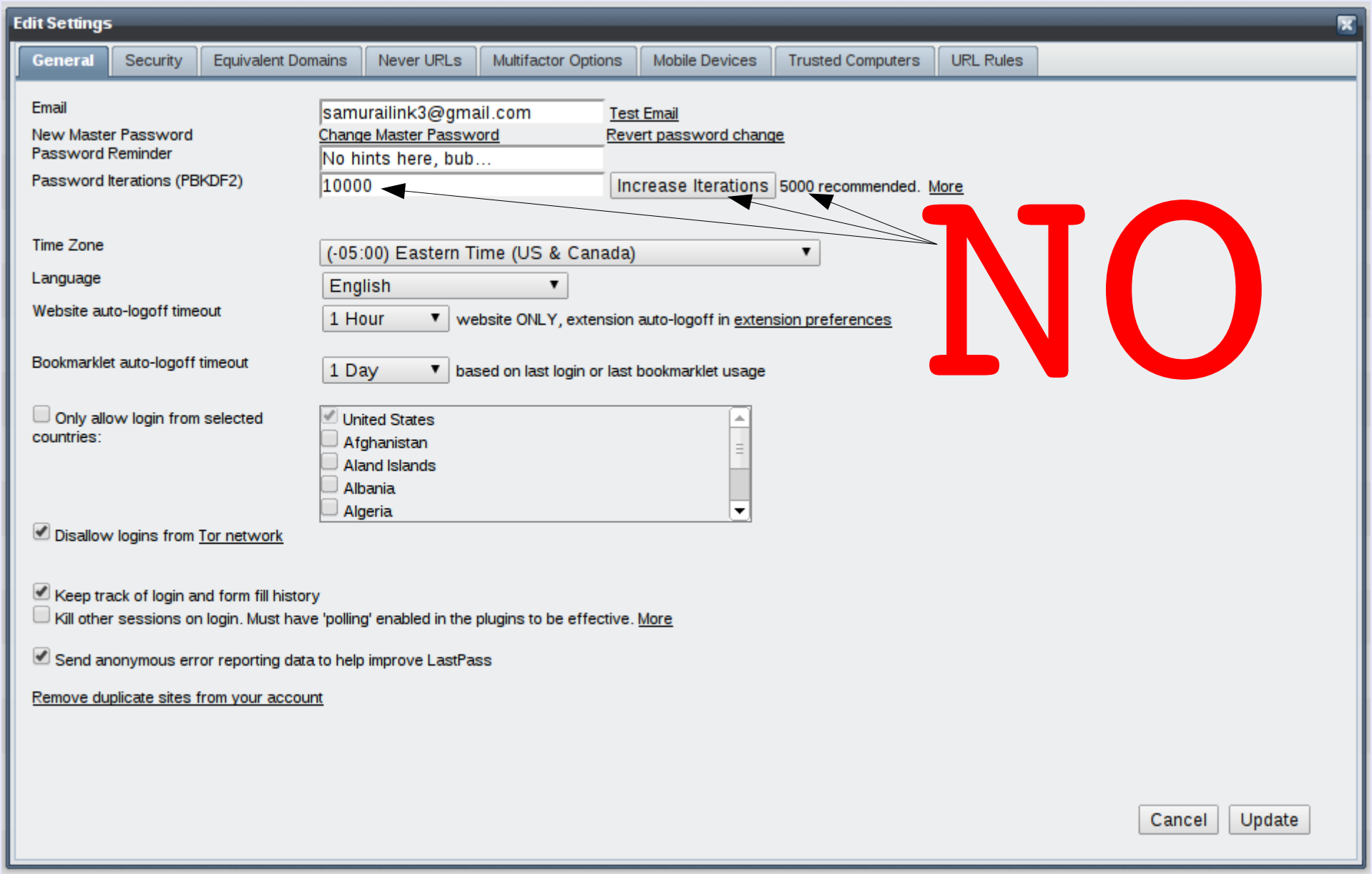
EXAMPLES

LastPass

But...

EXAMPLES

LastPass



EXAMPLES

LastPass

Errata:

Good:

- Great password entry screen
- Simple, easy, to the point
- No overburdened with security jargon
- Technical details that crypto nerds care about is in the documentation.

Bad:

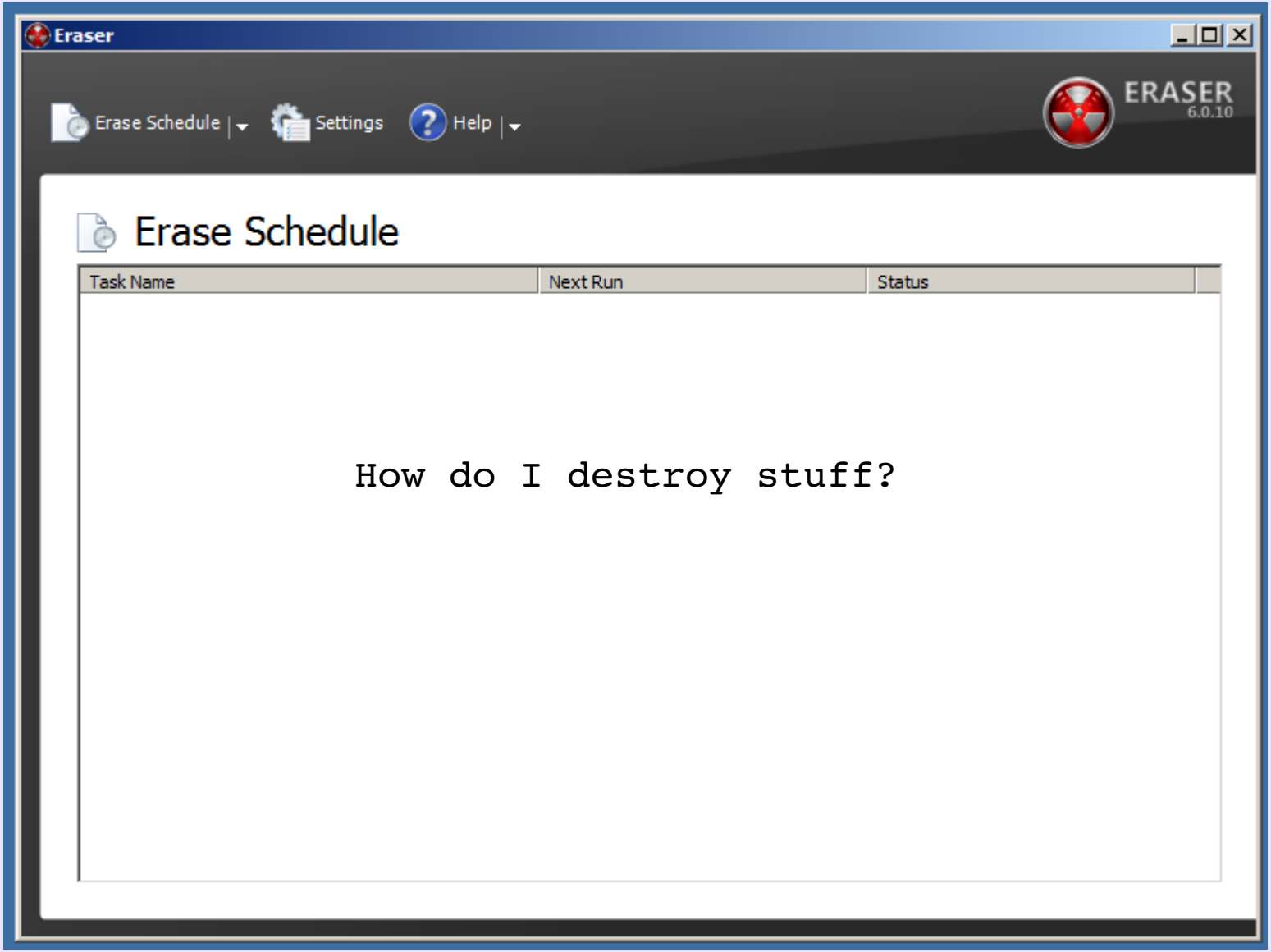
- Don't ever let the user hurt themselves. Ever. No matter what.
- Setting a low PBKDF2 value will make brute force attacks easier.
- If the recommendation is changed to a higher value than the user has set, no prompt or notification is issued.
- Be opinionated. Pick a value, use it.

LESSONS LEARNED

6. NEVER LET USERS HURT THEMSELVES. EVER.

EXAMPLES

Eraser



EXAMPLES

Errata:

Good:

- Simplified Interface

Bad:

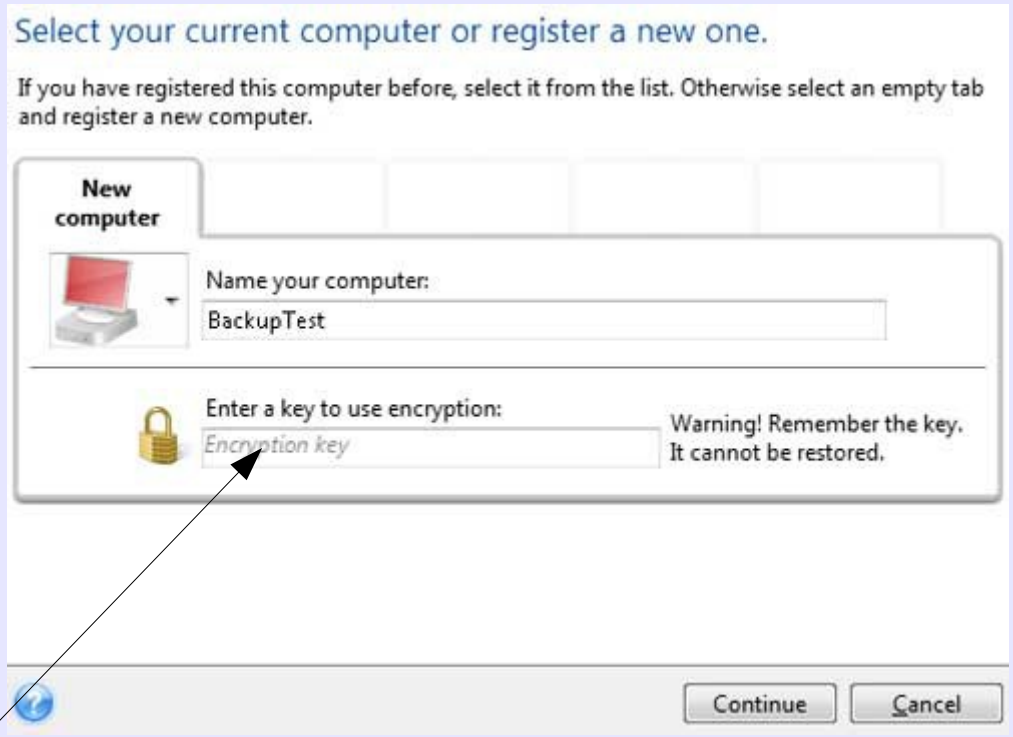
- Simplified in the exact wrong way
- 95% of your users want to do a simple thing quickly and easily. Enable that.
- Why put the scheduler first? Most people are looking to shred a single file.

LESSONS LEARNED

7. BUILD FOR THE 99%

EXAMPLES

Carbonite



Keep it simple. You could have pages upon pages of options and buttons here, but you don't need them. Most of your users don't want them. Be opinionated. Make the right choice for your user. Tuck your advanced options away.

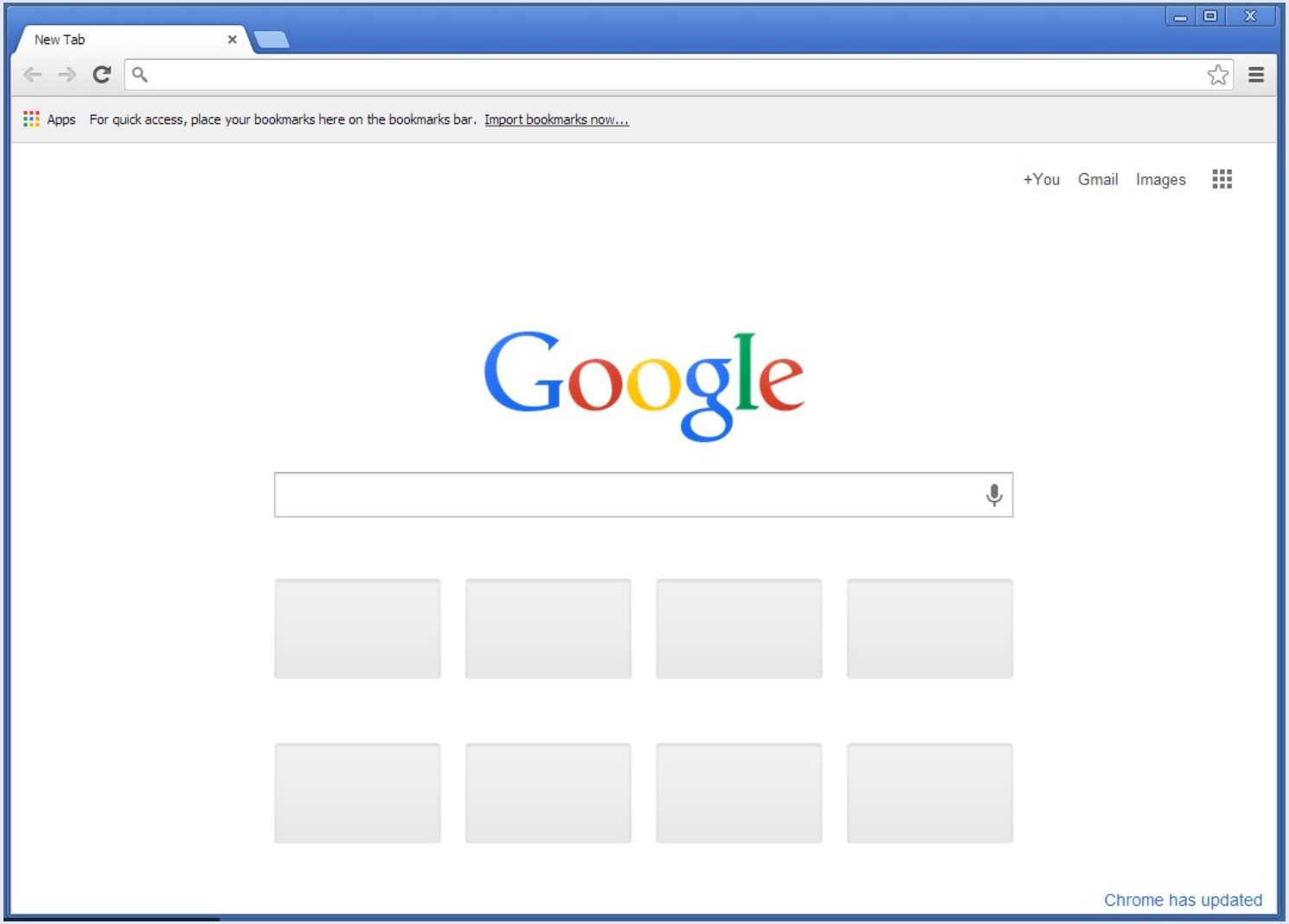
LESSONS LEARNED

8. MAKE OPINIONATED SOFTWARE

EXAMPLES

Google Chrome

Possibly the best blend of prowess and usability out there.

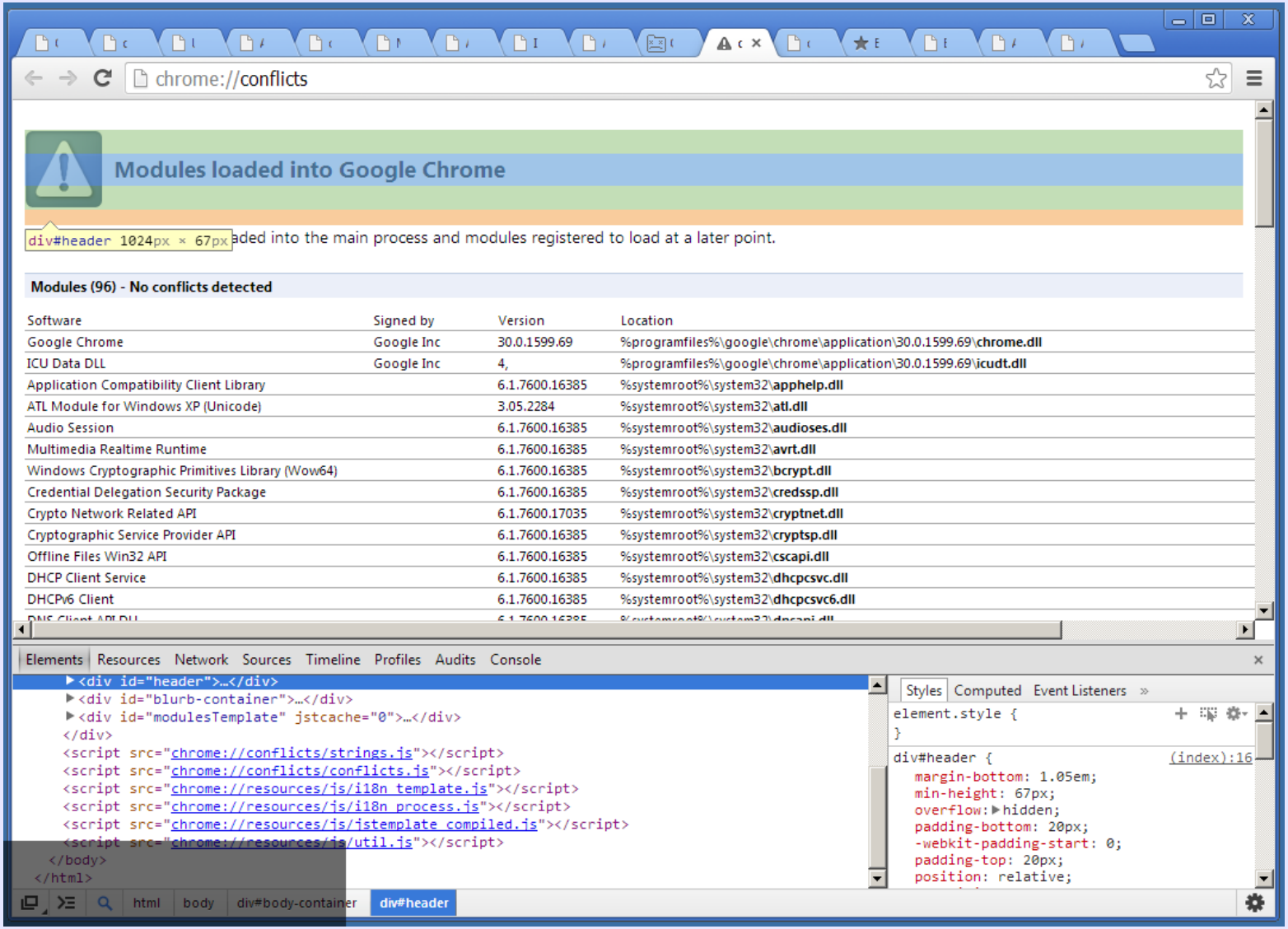


Simple. Clean. Chrome gives you only what you need up front and not a bit more. But! Dig just a little deeper, and Chrome becomes one of the best web-dev/debugging/perf testing suites ever created. Ever.

EXAMPLES

Google Chrome

Possibly the best blend of prowess and usability out there.



Everything from OpenGL testing to live CSS edits, Chrome's developer tools give you an incredible amount of control over how your browser operates.

LESSONS LEARNED

9. IF YOU NEED TO INCLUDE ADVANCED OPTIONS, HIDE THEM.

EXAMPLES

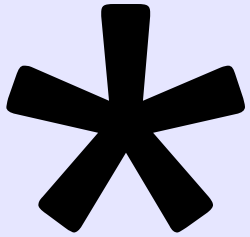
Mega.co.nz

LESSONS LEARNED

10. KEEP IT SIMPLE

EXAMPLES

SSL/TLS



* SSL/TLS actually doesn't have a defined interface.
  It is the most pervasive form of encryption used
  today, but it doesn't have an official GUI, logo,
  or desktop application.

If your software can protect people without them
  knowing it, you've won.

LESSONS LEARNED

11. IF YOU CAN, HAVE NO INTERFACE AT ALL

THE ULTIMATE LESSON

0. MAKE IT WORK

This should go without saying, if you're building a crypto app that you want people to use, actually make it secure, make it work. People who do 5 minutes of Googling avoid Hushmail like the plague because they find out they aren't really secure at all. If you build an application, make it right, make it in public (open source), and report any and every breach you have. This is our industry, don't screw it up.

THANKS!

Slides, Fonts, Colors, Software, and Image credits to appear on samurailink3.com/talks.