Security: I Think We Can Win!

Bill Cheswick

(Your institution name here?)

ches@cheswick.com,
http://www.cheswick.com/ches/talks/APPsec2013.pdf

Introduction

- Thinking about security since the Nixon administration
- Starting to get a long view of things
- Generalizations, Grumbles, Hand-waving
- Not a grant proposal, No perfect solutions, No universal solutions
- References are on the slides, see my web page for a PDF of the talk

I think we can win

- Meaning build an affordable computing platform that can’t be compromised by any user error not involving a screw driver
- Winning doesn’t mean that your machine can’t misbehave on the Internet

Introduction

- I *love* living in the future
- Velcro, 12-hour nasal spray, surgical “lasers”, routine rockets to LEO, astonishing computers
- Sick and tired of computer and network security problems
- Hacked for CPU seconds!
- Already a lot of good security work done
    - Time sharing, Multics
    - Spooks

Sick and Tired

- APT are not Advanced, but certainly Persistent and Threats
- Most of the attacks are on the same kinds of weaknesses: we are not making much progress
- Consarn it, I am becoming an old timer!

Long view: it is still early in the computer revolution

- I know, I know, we aren't talking UNIVAC or "minicomputers" any more.

- The order of things: make it work, then worry about security: **(It Works!)**

  - (Very bad prognosis for Obamacare data handling)

- rlogin, NFS, X windows, MSFT before 2001.

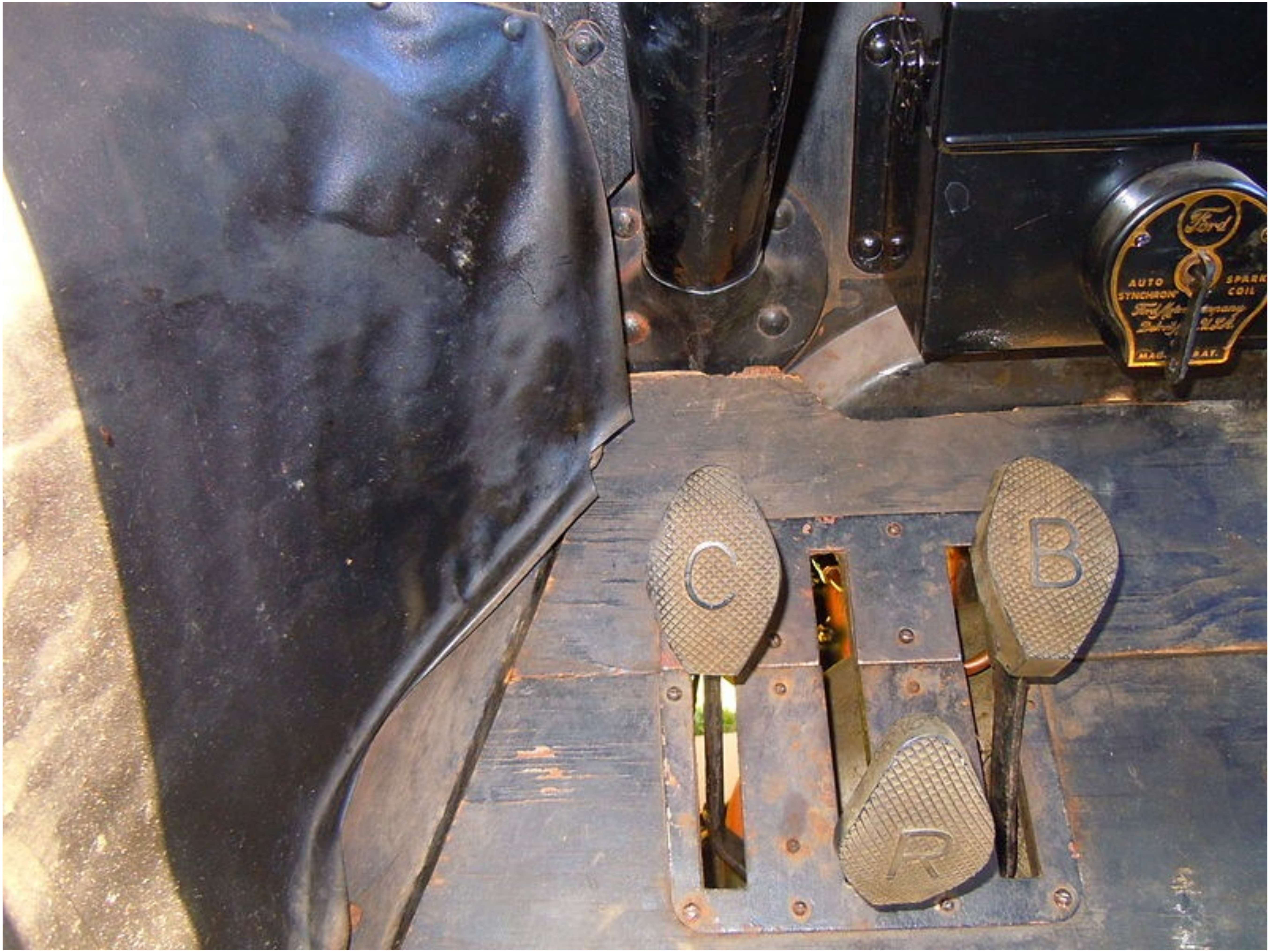
- But look where we are in UIs: I thought we might get stuck with MSFT menus, like the QWERTY keyboard

The car metaphor

- I didn’t like it: apples and oranges
- Now I do: grapes and raisins
- Consider the Model T:

Ford Model T (1913)

- 20 hp
- runs on gasoline, kerosine, and ethanol
- rear wheel drive
- two speeds, plus reverse
- grey, green, blue, and red (1909 - 1913)
- 1913 model (shown) was $550 (four months pay for an assembly line worker.
- Electric start!



Ford
AUTO SYNCHRON SPARK COIL
Ford Motor Company
Detroit, U.S.A.
C
B
R

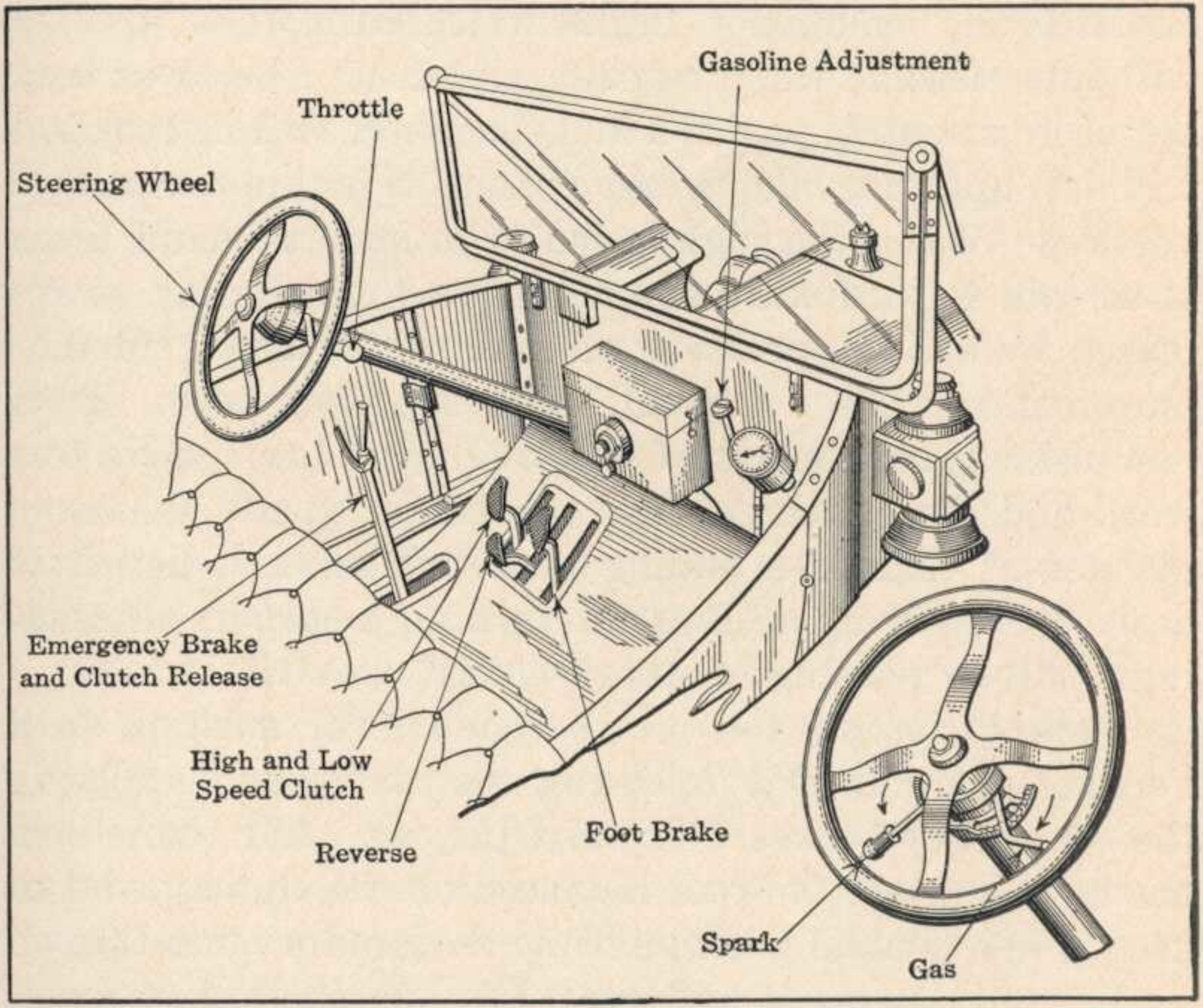


Fig. 42.—The Control System of the Ford Model T Car.

Some old-timey auto stuff

- Fading terms: choke, flood the engine, friction point, vapor lock, double-clutch
- My mother had a car you had to back up steep hills because there wasn’t a fuel pump
- First seat belts (two-point) common in mid-1960s

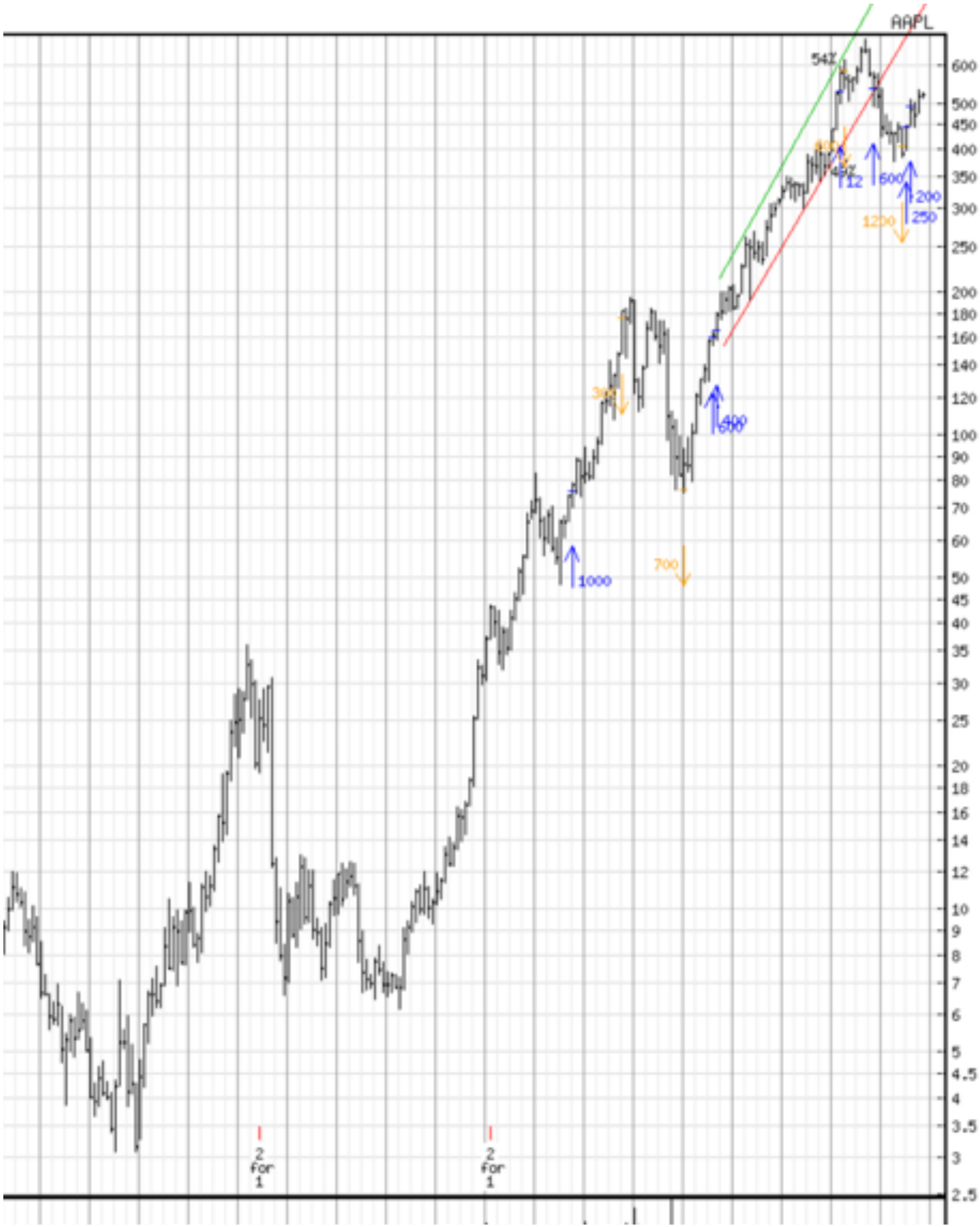
You don’t have to be a mechanic to drive your car, and you shouldn’t have to be a programmer or security expert to use your computer safely.

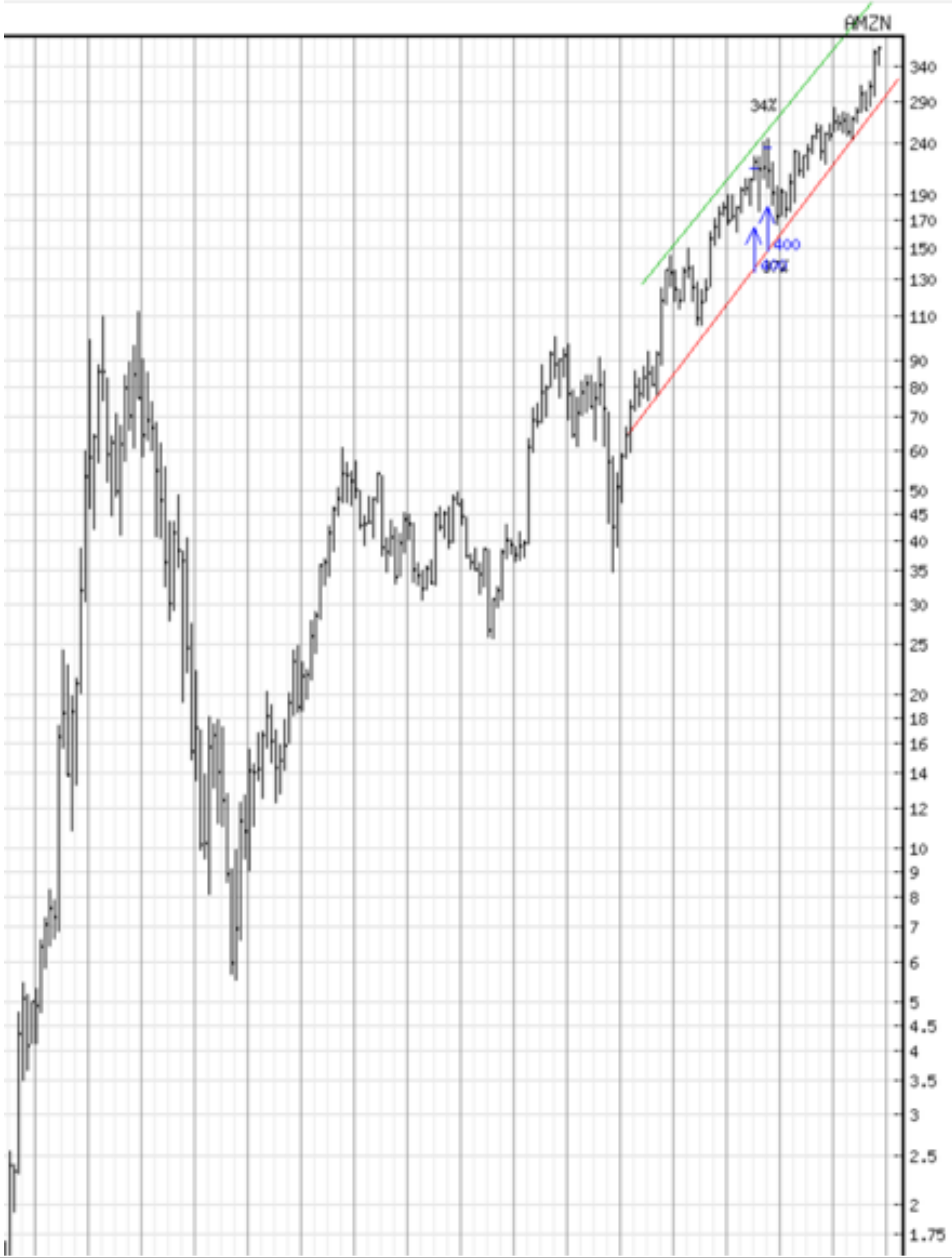
New car troubles

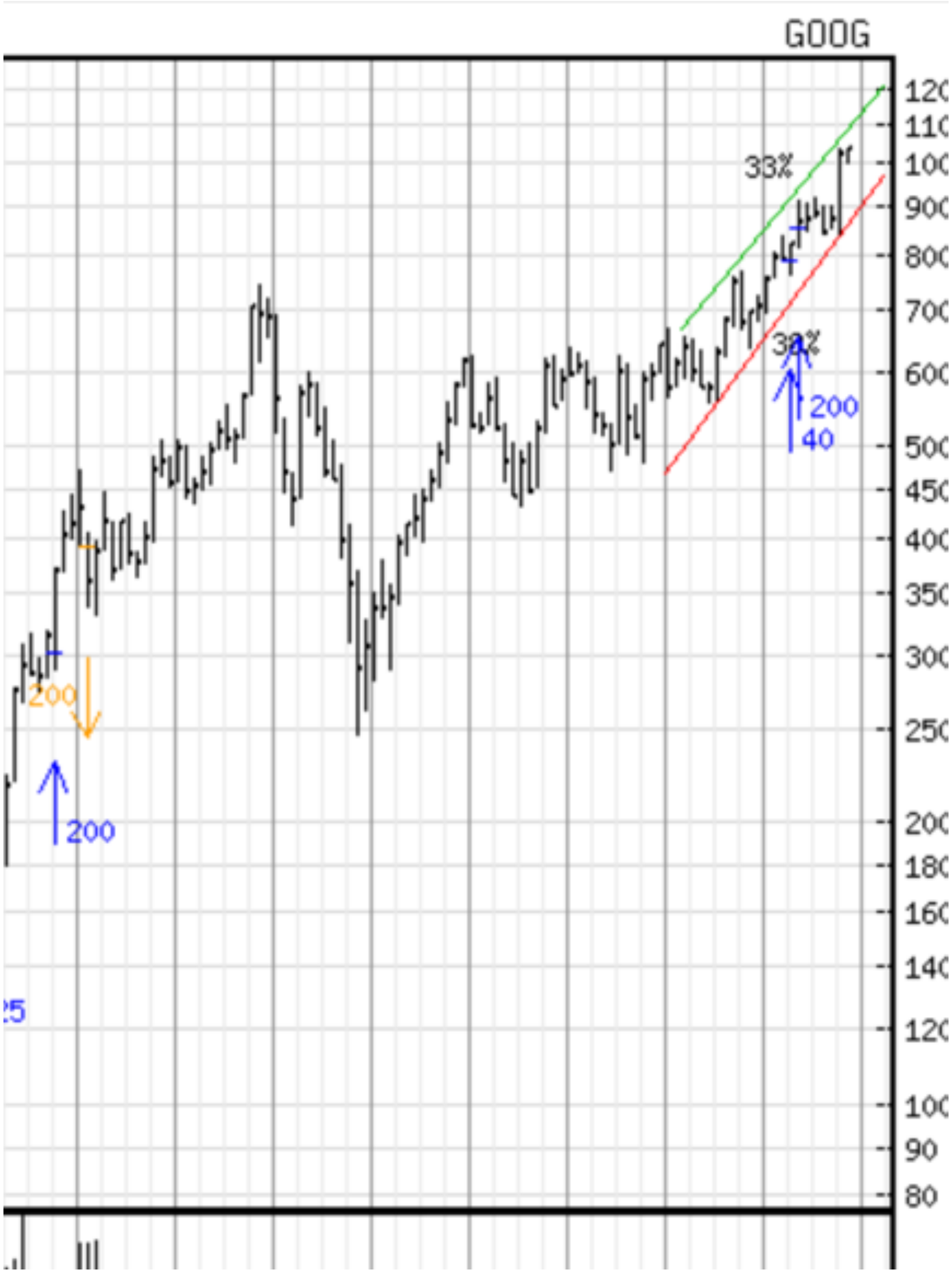
- Note: cars now need the second kind of firewall
- Attacks on the CAN bus **(It Works!)**
  - attacks through Bluetooth, evil mp3 files, etc.
  - web search for *“can bus security”*
- Here we go again

What is the current state of affairs? Great!

- It's great!
    - banking?
    - retirement accounts
    - shopping and commerce?


AAPL
54%
600
45%
12
600
200
250
1200
300
400
600
1000
700
2
for
1
2
for
1
600
500
450
400
350
300
250
200
180
160
140
120
100
90
80
70
60
50
45
40
35
30
25
20
18
16
14
12
10
9
8
7
6
5
4.5
4
3.5
3
2.5


AMZN
34%
400
340
290
240
190
170
150
130
110
90
80
70
60
50
45
40
35
30
25
20
18
16
14
12
10
9
8
7
6
5
4.5
4
3.5
3
2.5
2
1.75


GOOG
33%
32%
200
40
200
200
25
120
110
100
900
800
700
600
500
450
400
350
300
250
200
180
160
140
120
100
90
80

What is the current state of affairs? Lousy!

- Spies are all in our business
- Huge advantage to the attackers
- Crappy client operating systems
    - leaky sandboxes
    - feature-driven
- A visit to grandma's house

What isn't working?

- Checklists
  - They certainly will catch oversights, but you are not secure when you are done
- laws, general and specific
  - general: nice guidelines, but exactly how much protection does HIPAA demand
  - specific: see *ChecklistsI,* above

Not working

- PCI
  - see above, plus it misses things
- User education
  - It doesn’t fix bad engineering, and it is bad engineering to assume that it will. See below.

Not working

- Virus checking: it helps, but it will never be a win
    - It solves the wrong problem, and ultimately requires solving the halting problem.
- Strong user passwords
    - More poor engineering: it just doesn't work by itself, and isn't needed when used with the right authentication tools.

Not working

- shared and dynamic libraries
- changes the ground on which you stand
- implemented to save memory and load time
- *not* worth it
- "sshd day zero bug" this year was shared library replacement attack.
- Make all your binaries static!

Not working: PKI

- The trusted CA list is way out of hand
- Try CertPatrol on Firefox to see what is going on

What Might A Secure World Look Like?

What victory might look like

- The OS can’t be changed or subverted, regardless off app run and user action
- Apps cannot taint the OS or other apps
- Apps can be limited to signed, approved choices
- Random software (i.e. web java, etc.) can be run but can’t taint apps or OS
- Reputation-based PKI?
- Ubiquitous end-of-end crypto

Windows OK

- There is nothing you can click, tap, or say that will corrupt your computer.
- It should be intuitively obvious when you are not visiting a Fortune 500 web site, or a place you have never searched before.
- Offers standard services

Windows OK

- Good for grandma, most employees, most students (who aren't gamers), spooks on classified networks

Do we have this already?

- Jeff Jones (MSFT) said Win 7 was much safer than corresponding Linux
- Maybe Win 8, too
- Seems like an awfully large hunk of software to declare victory, and maybe they haven’t.

Maybe iOS...

- Certainly Apple tried hard to design security into iOS, and they had a fresh start, sort of
- How can we tell? Measure security…

Things that don’t work very well: measuring security

When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge of it is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced it to the stage of science.

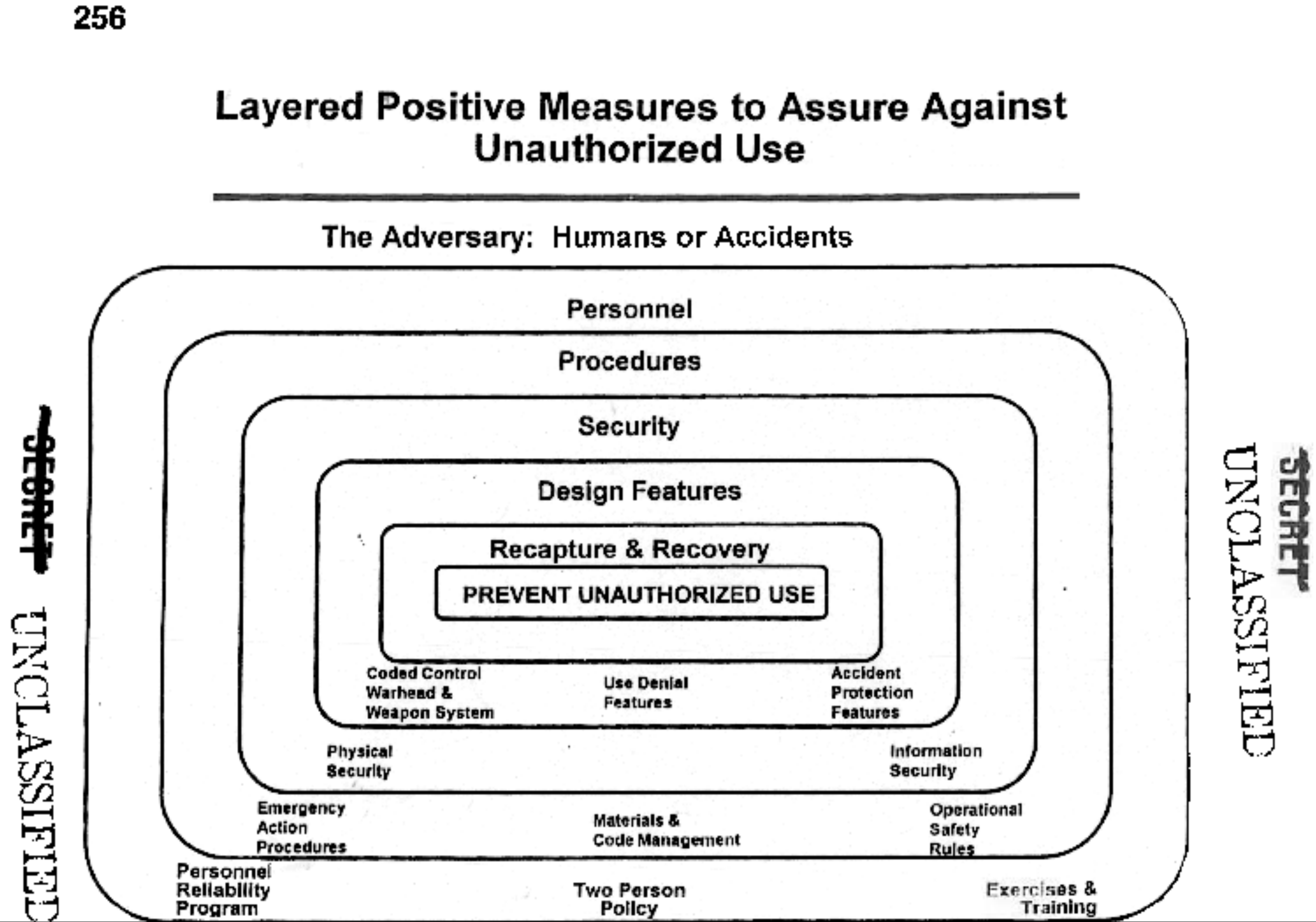
— Lord Kelvin

Measuring security

- The general wanted a number
- TruSecure and Counterpane
- Microsoft’s(?) “attack surface”* concept

* Pratyusa K. Manadhata, *An Attack Surface Metric”*,
CMU-CS-08-152, November 2008

Where to measure?



256

Layered Positive Measures to Assure Against Unauthorized Use

The Adversary: Humans or Accidents

Personnel

Procedures

Security

Design Features

Recapture & Recovery

PREVENT UNAUTHORIZED USE

Coded Control Warhead & Weapon System

Use Denial Features

Accident Protection Features

Physical Security

Information Security

Emergency Action Procedures

Materials & Code Management

Operational Safety Rules

Personnel Reliability Program

Two Person Policy

Exercises & Training

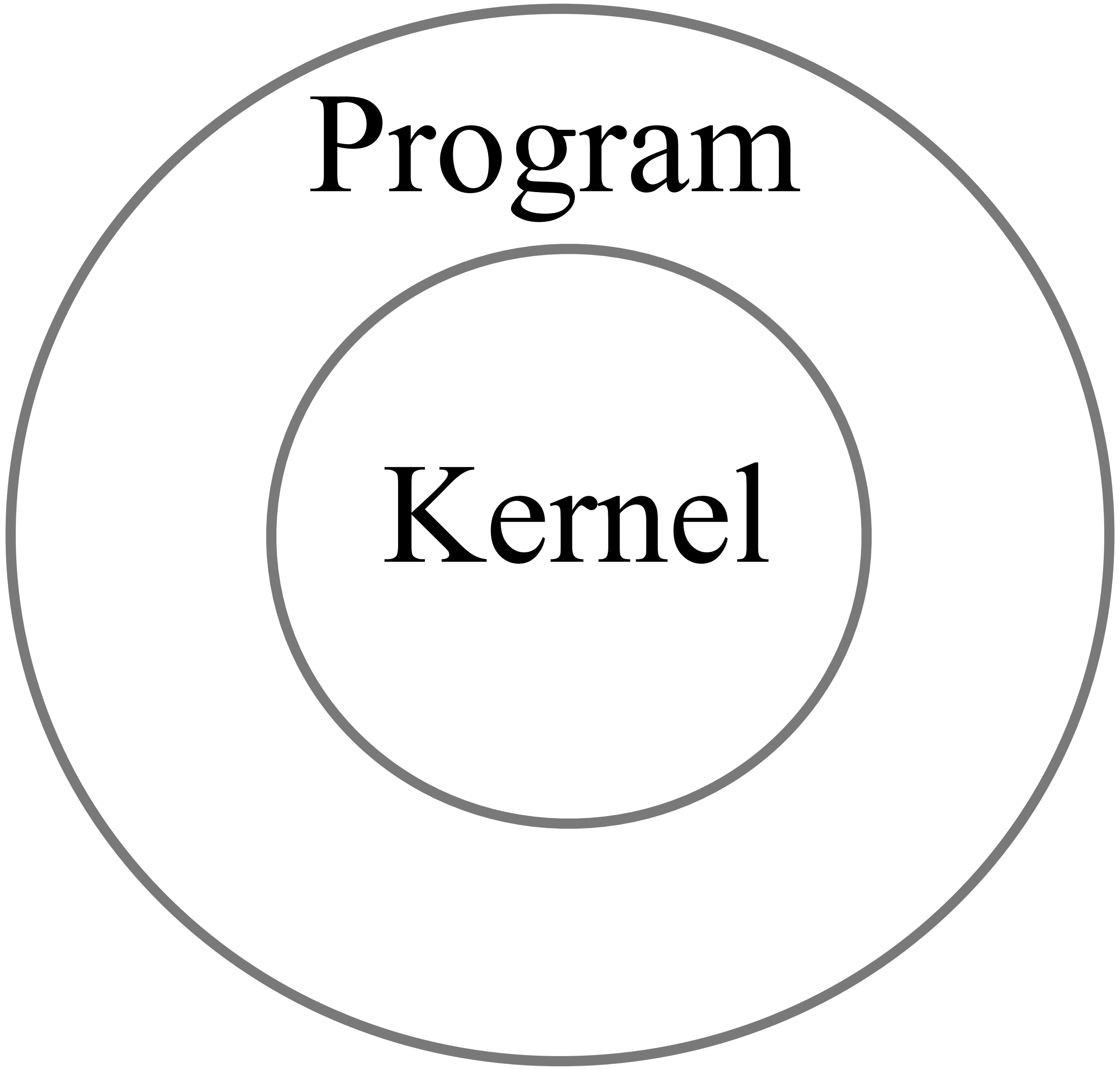
~~SECRET~~ UNCLASSIFIED

~~SECRET~~ UNCLASSIFIED

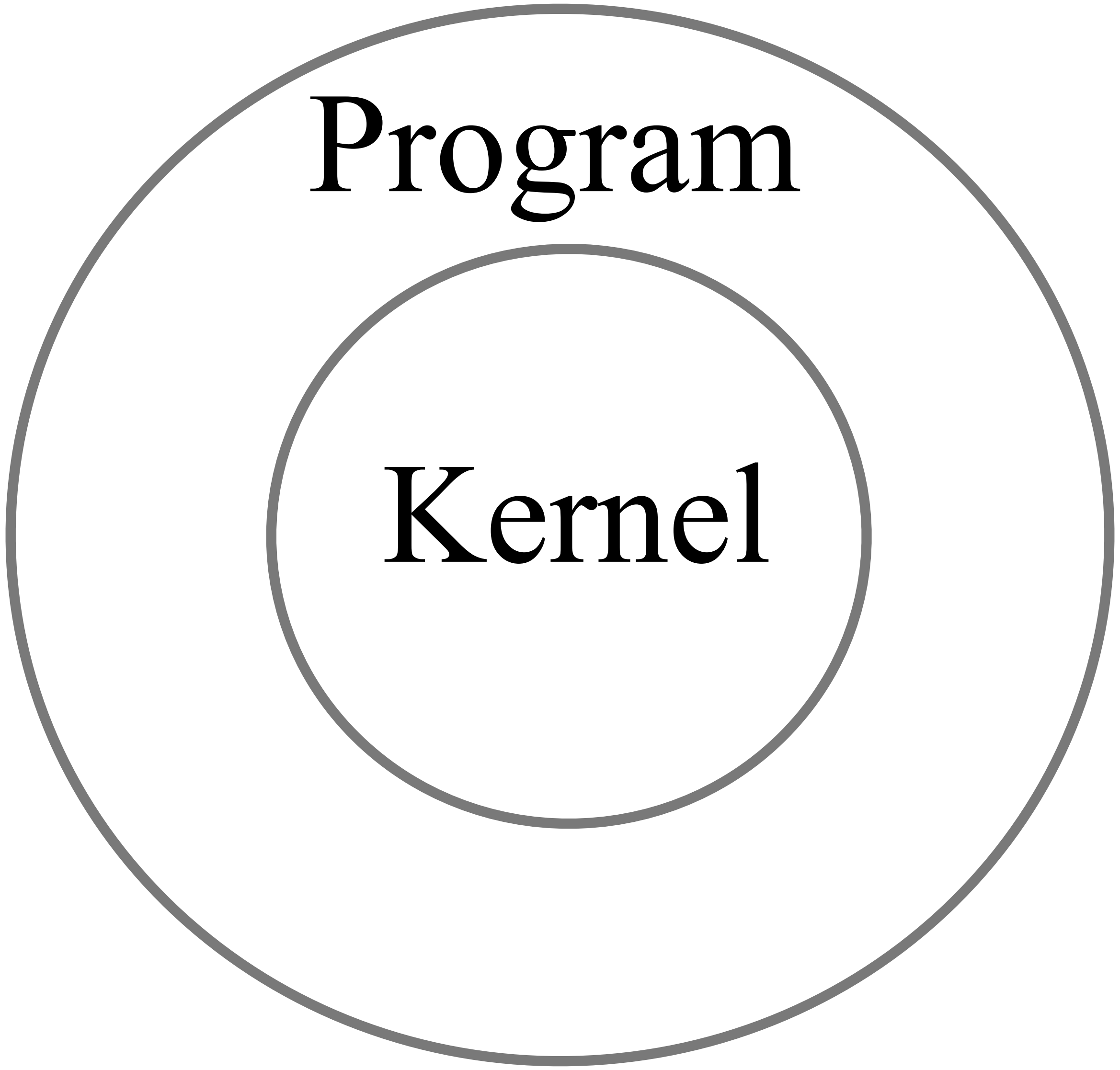
Spooks

- Valuable friends, and good resources
- Don’t ask them about secrets, ask them about lessons
- They have been thinking very hard and very long (> 50 years) about the issues raised in this talk

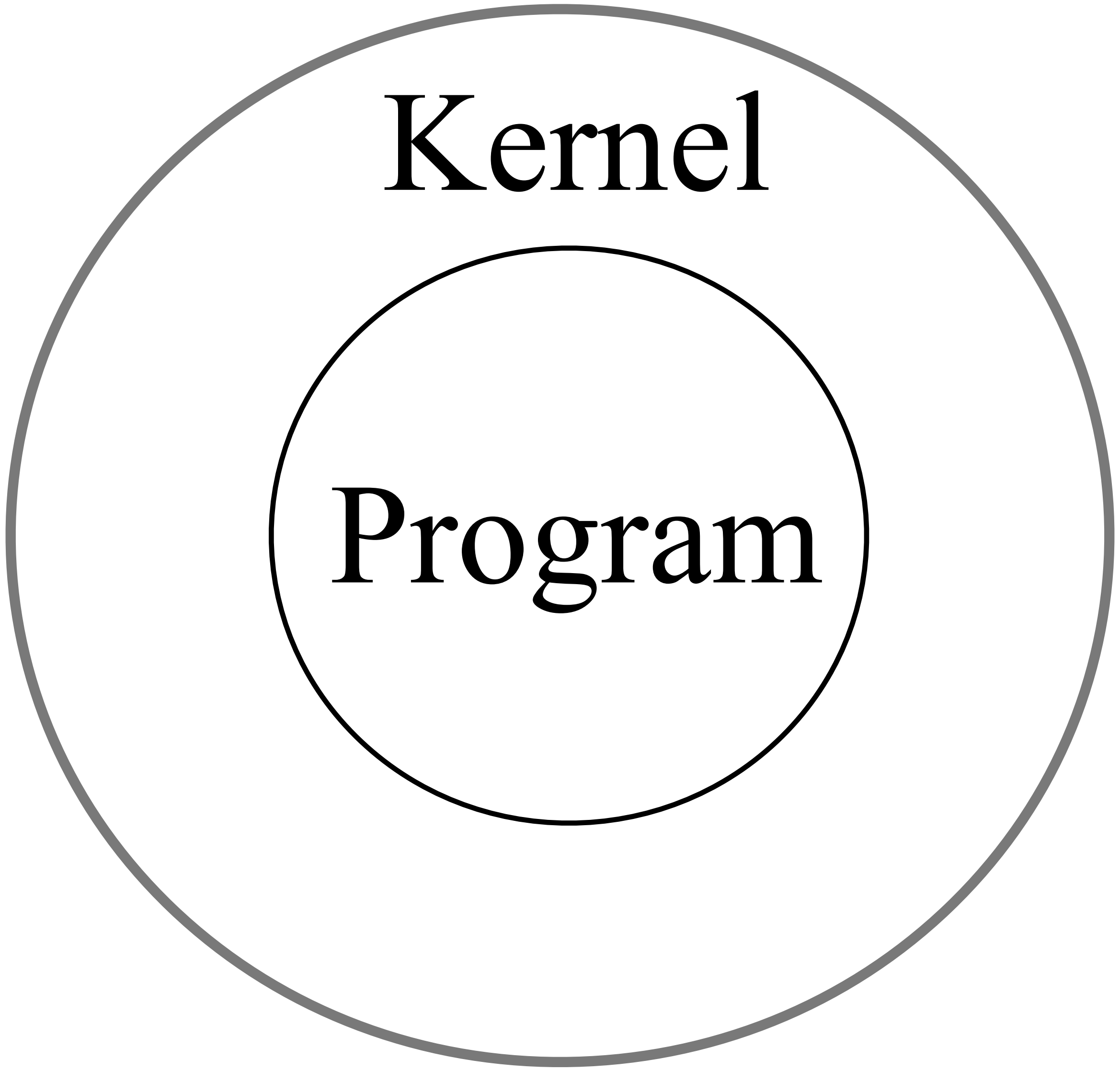
Textbook diagram



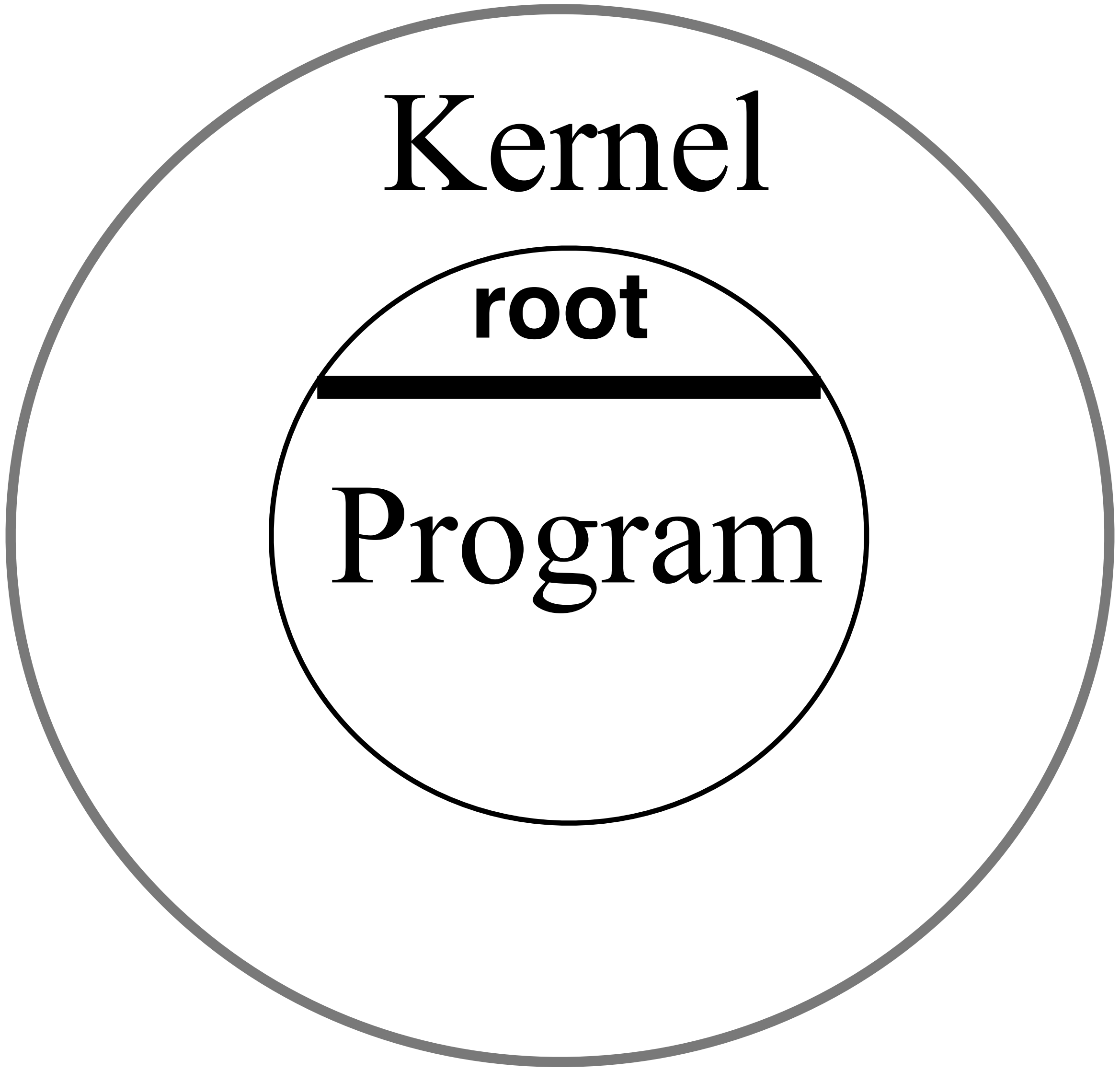
Wrong!



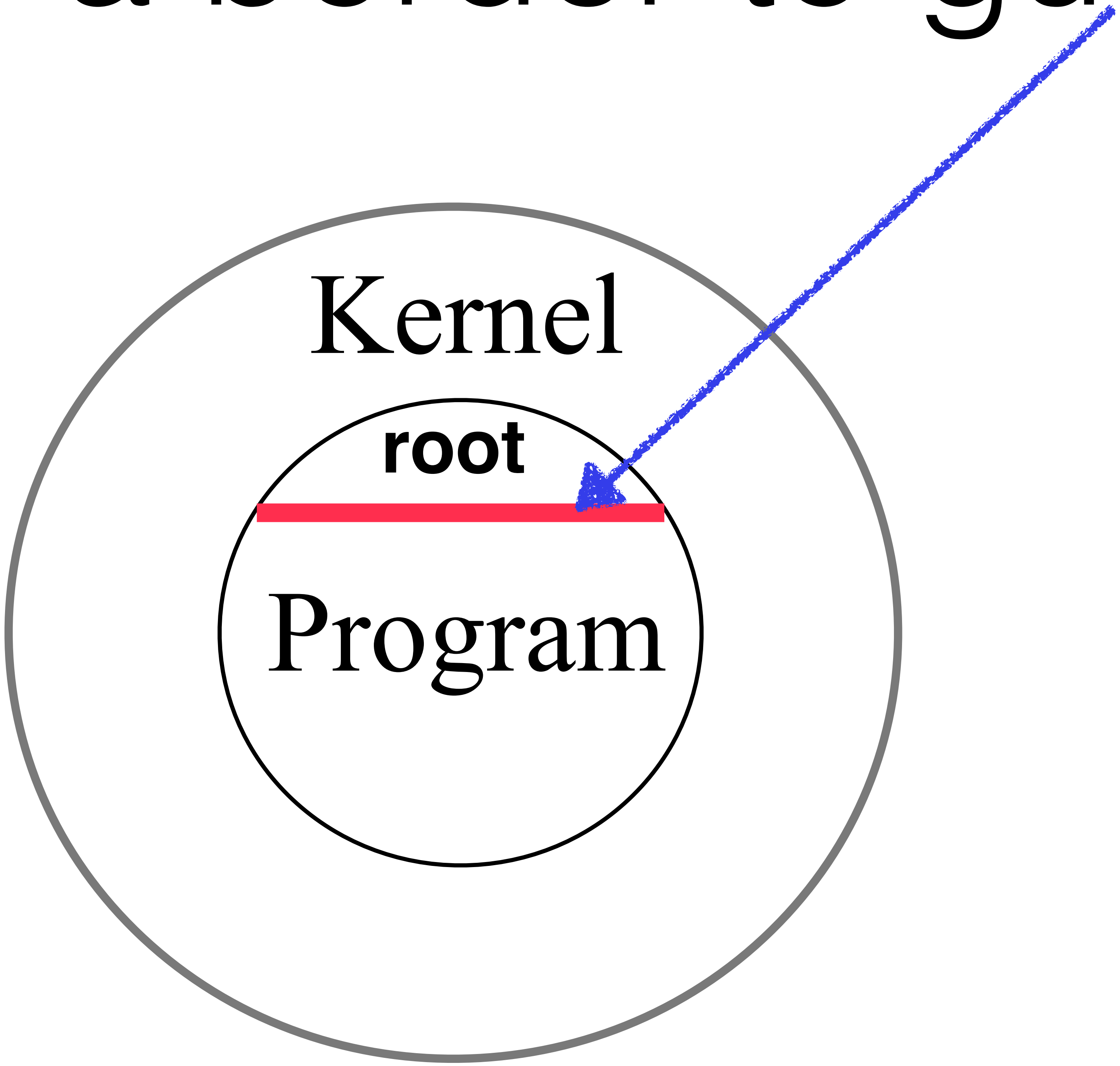
The kernel talks to the world, the user level nestles inside



root is a special user, and a hacking goal



Here's a border to guard



Rate a Unix system’s security from one of its users

```
find / -perm -4000 -user root –print | wc -l
```

```
/bin/rcp
/sbin/ping
/sbin/ping6
/sbin/shutdown
/usr/X11R6/bin/Xwrapper
/usr/X11R6/bin/xterm
/usr/X11R6/bin/Xwrapper-4
/usr/bin/keyinfo
/usr/bin/keyinit
/usr/bin/lock
/usr/bin/crontab
/usr/bin/opieinfo
/usr/bin/opiepasswd
/usr/bin/rlogin
/usr/bin/quota
/usr/bin/rsh
/usr/bin/su
/usr/bin/lpq
/usr/bin/lpr
/usr/bin/lprm
/usr/bin/chpass
/usr/bin/login
/usr/bin/passwd
/usr/bin/at
/usr/bin/ypchsh
/usr/bin/ypchfn
/usr/bin/ypchpass
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/yppasswd
/usr/bin/batch
/usr/bin/atrm
/usr/bin/atq
/usr/local/bin/screen
/usr/local/bin/sudo
/usr/local/bin/lppasswd
/usr/sbin/mrinfo
/usr/sbin/mtrace
/usr/sbin/ppp
/usr/sbin/pppd
/usr/sbin/sliplogin
/usr/sbin/timedc
/usr/sbin/traceroute
/usr/sbin/traceroute6
```

44

```
/bin/rcp
/sbin/ping
/sbin/ping6
/sbin/shutdown
/usr/X11R6/bin/Xwrapper
/usr/X11R6/bin/xterm
/usr/X11R6/bin/Xwrapper-4
/usr/bin/keyinfo
/usr/bin/keyinit
/usr/bin/lock
/usr/bin/crontab
/usr/bin/opieinfo
/usr/bin/opiepasswd
/usr/bin/rlogin
/usr/bin/quota
/usr/bin/rsh
/usr/bin/su
/usr/bin/lpq
/usr/bin/lpr
/usr/bin/lprm
/usr/bin/chpass
/usr/bin/login
/usr/bin/passwd
/usr/bin/at
/usr/bin/ypchsh
/usr/bin/ypchfn
/usr/bin/ypchpass
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/yppasswd
/usr/bin/batch
/usr/bin/atrm
/usr/bin/atq
/usr/local/bin/screen
/usr/local/bin/sudo
/usr/local/bin/lppasswd
/usr/sbin/mrinfo
/usr/sbin/mtrace
/usr/sbin/ppp
/usr/sbin/pppd
/usr/sbin/sliplogin
/usr/sbin/timedc
/usr/sbin/traceroute
/usr/sbin/traceroute6
```

18

Some should not be root, or setuid

```
/sbin/ping
/sbin/ping6
/usr/X11R6/bin/xterm
/usr/X11R6/bin/Xwrapper-4
/usr/bin/crontab
/usr/bin/su
/usr/bin/lpq
/usr/bin/lpr
/usr/bin/lprm
/usr/bin/login
/usr/bin/passwd
/usr/bin/at
/usr/bin/chsh
/usr/bin/atrm
/usr/bin/atq
/usr/local/bin/sudo
/usr/sbin/traceroute
/usr/sbin/traceroute6
```

Ta da!

```
/usr/X11R6/bin/Xwrapper-4
/usr/bin/su
/usr/bin/passwd
/usr/bin/chsh
/usr/local/bin/sudo
```

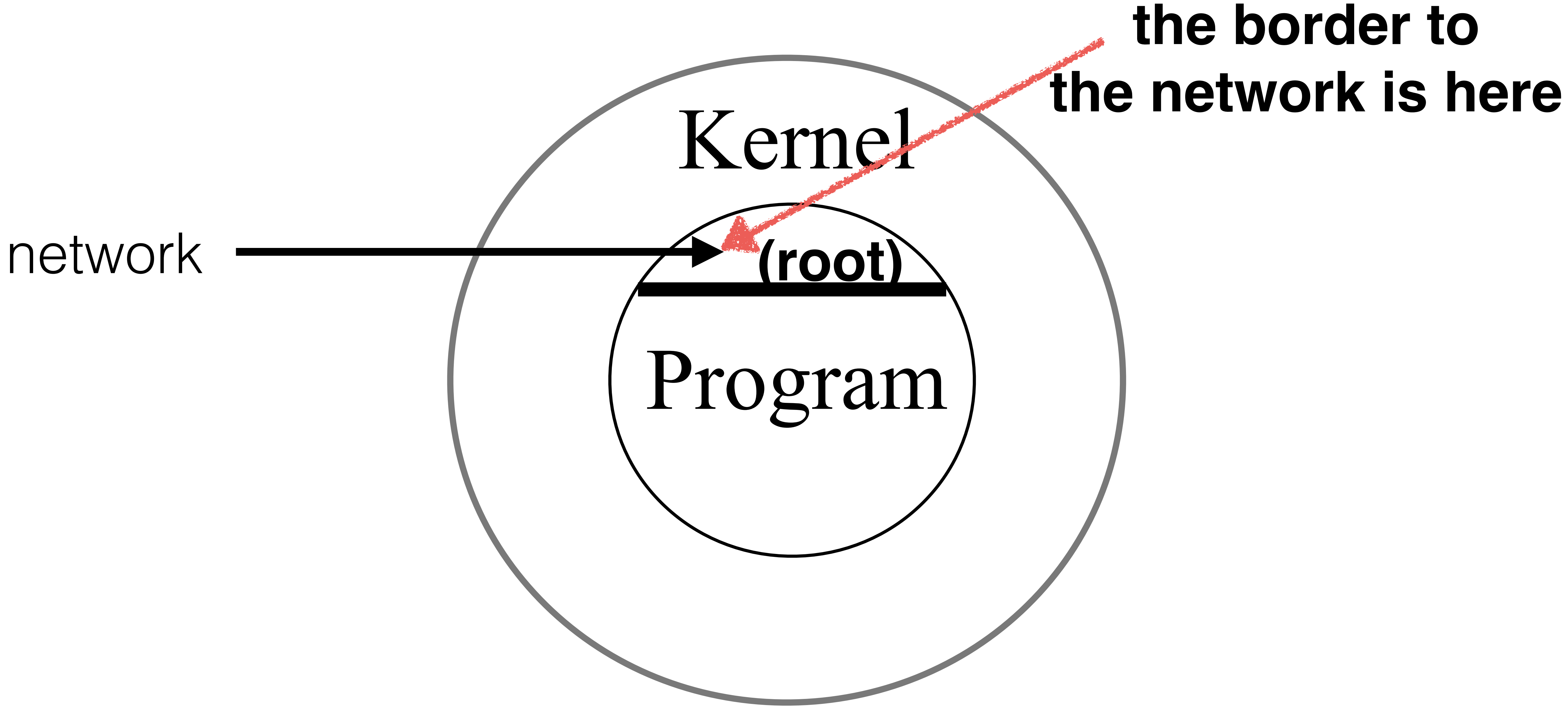
5

```
AIX 4.2                 & 242   & a staggering number \\
BSD/OS 3.0              & 78            \\
FreeBSD 4.3             & 42    & someone's guard machine\\
FreeBSD 4.3             & 47    & 2 appear to be third-party\\
FreeBSD 4.5             & 43    & see text for closer analysis \\
HPUX A.09.07            & 227   & about half may be special for this host
\\
Linux (Mandrake 8.1)    & 39    & 3 appear to be third-party \\
Linux (Red Hat 2.4.2-2) & 39    & 2 third-party programs \\
Linux (Red Hat 2.4.7-10)        & 31    & 2 third-party programs\\
Linux (Red Hat 5.0)     & 59        \\
Linux (Red Hat 6.0)     & 38    & 2--4 third-party \\
Linux 2.0.36            & 26    & approved distribution for one
university \\
Linux 2.2.16-3          & 47             \\
Linux 7.2               & 42        \\
NCR Intel 4.0v3.0       & 113   & 34 may be special to this host \\
NetBSD 1.6              & 35        \\
SGI Irix 5.3            & 83        \\
SGI Irix 5.3            & 102           \\
Sinux 5.42c1002         & 60    & 2 third-party programs\\
Sun Solaris 5.4         & 52    & 6 third-party programs\\
Sun Solaris 5.6         & 74    & 11 third-party programs\\
Sun Solaris 5.8         & 70    & 6 third-party programs\\
Sun Solaris 5.8         & 82    & 6 third-party programs\\
Tru64 4.0r878           & 72    & \\
```

Rate a Unix system's network security

```
netstat -an | wc -l
```

Network services



This Mac

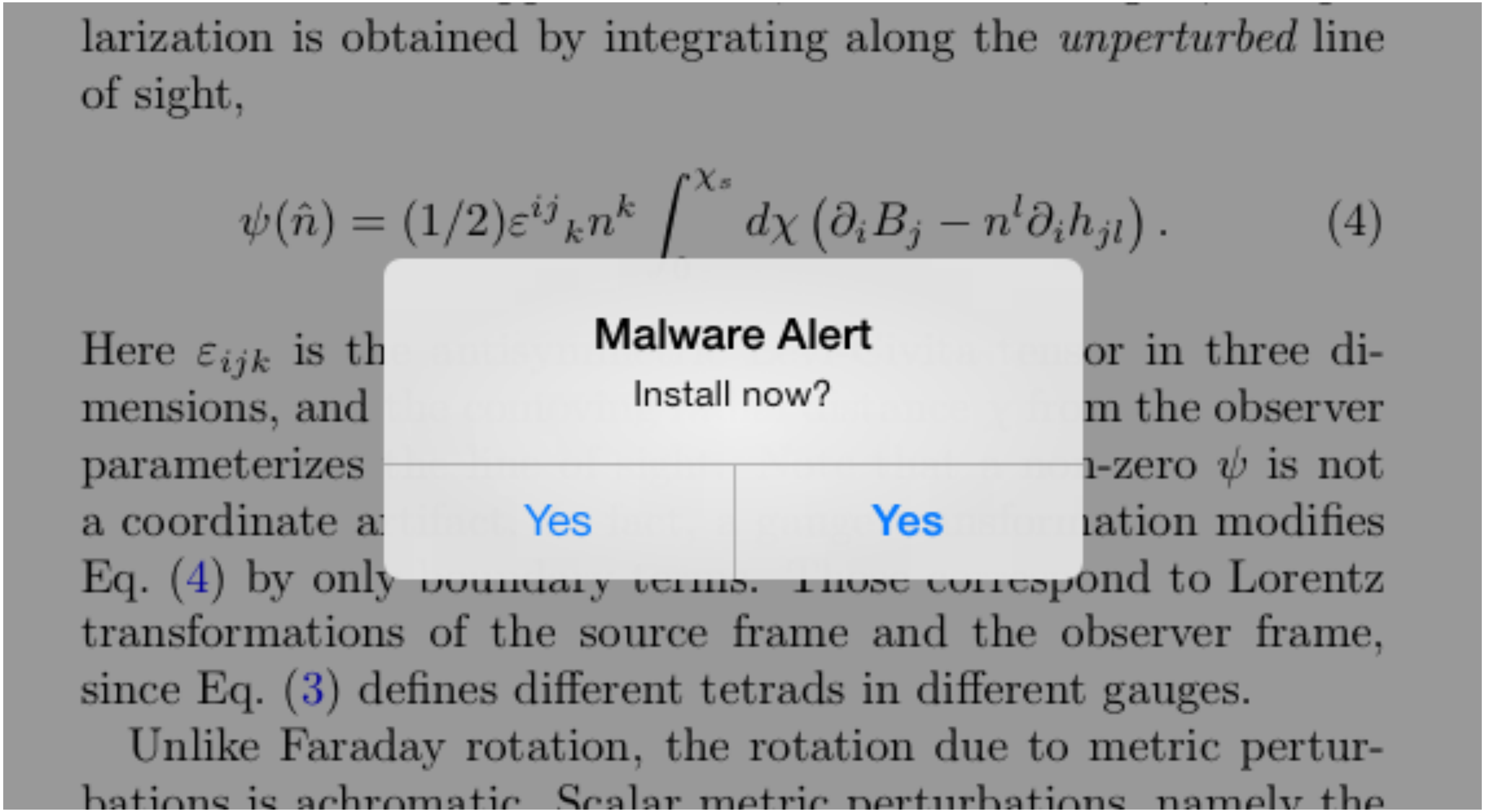
```
tcp4       0      0  *.17500                *.*                    LISTEN
tcp4       0      0  *.22                   *.*                    LISTEN
tcp6       0      0  ::1.631                *.*                    LISTEN
udp6       0      0  *.55117                *.*
udp6       0      0  *.50868                *.*
udp6       0      0  *.54355                *.*
udp6       0      0  *.52357                *.*
udp6       0      0  *.61402                *.*
udp6       0      0  *.52228                *.*
udp6       0      0  *.54159                *.*
udp6       0      0  *.51012                *.*
udp4       0      0  *.61549                *.*
udp4       0      0  *.57704                *.*
udp4       0      0  *.62703                *.*
udp4       0      0  *.59177                *.*
udp4       0      0  *.52971                *.*
udp4       0      0  223.223.223.35.123     *.*
udp4       0      0  *.17500                *.*
udp4       0      0  *.5353                 *.*
udp4       0      0  *.138                  *.*
udp4       0      0  *.137                  *.*
```

Measuring security

Adobe Reader	$5,000 - $30,000
MAC OSX	$20,000 - $50,000
Android	$30,000 - $60,000
Flash or Java Browser	$40,000 - $100,000
Microsoft Word	$50,000 - $120,000
Windows	$60,000 - $120,000
Firefox or Safari	$60,000 - $150,000
Chrome or Internet	$80,000 - $200,000
iOS	$100,000 - $250,000

Andy Greenberg, *Shopping For Zero-Days: A Price List For Hacker's Secret Software Exploits*. Forbes, 23 March 2012.
http://www.forbes.com/sites/andygreenberg/2012/03/23/shopping-for-zero-days-an-price-list-for-hackers-secret-software-exploits/

Apple security?



larization is obtained by integrating along the *unperturbed* line of sight,

$$\psi(\hat{n}) = (1/2)\varepsilon^{ij}{}_{k} n^{k} \int^{\chi_s} d\chi \left(\partial_i B_j - n^l \partial_i h_{jl}\right) . \tag{4}$$

Here ε_{ijk} is th or in three dimensions, and m the observer parameterizes -zero ψ is not a coordinate a ation modifies Eq. (4) by only boundary terms. Those correspond to Lorentz transformations of the source frame and the observer frame, since Eq. (3) defines different tetrads in different gauges.

Unlike Faraday rotation, the rotation due to metric perturbations is achromatic. Scalar metric perturbations, namely the

Malware Alert

Install now?

Apple security?

- I love these devices, so I learned Rejective C and usually follow their UI advice slavishly.
- NextStep is from the late 1980s, which is okay in itself, but
    - retain count stuff went away (mostly) only a couple years ago when ARC came
    - It’s not just my software that crashes
- I don’t see how anyone can have confidence that their non-trivial program is correct in this system.

Apple iOS clients

- My best bet for the most secure clients at the moment, but it is scary
- Android: “the problem with folk songs is that they are written by the people.” — Tom Lehrer
- Lesson: it takes discipline to write good software. Maybe Apple’s experts can do this.

What Works

Lessons and Suspicions
(you may disagree)

Small is better: software

- It is harder to design, build, understand, debug, document, and audit complex systems
- In current open software environment, there is ongoing pressure to add features
- Norman’s IAG

Small is better

- Plan 9/Inferno operating system compiled in under 20 seconds
- Very few system calls
- Very few graphics calls
- For a taste of the approach, check out the *go* language from the same folks, at Google
    - A smart phone written in *go* would be very interesting

Small is better: simpler hardware?

- Most people have extremely modest computation and feature requirements, most of the time
  - Wordstar ran on computers 30 years ago

Pentium complexity

- Rings 3 and 0
- System Management Mode*
- Virtual machine interface
- Microcode?!
- How bad can a compromised CPU be?

* Duflot, Loïc, Daniel Etiemble, and Olivier Grumelard.
Using CPU system management mode to circumvent operating system security functions.
CanSecWest/core06 (2006). http://cs.usfca.edu/~cruse/cs630f06/duflot.pdf

Hardware problems: what can go wrong

- Subverted CPU
  - King, Samuel T., *et al*. *Designing and Implementing Malicious Hardware*. *LEET* 8 (2008): 1-8. https://www.usenix.org/legacy/events/leet08/tech/full_papers/king/king_html/
- Doping
  - Becker, Georg T., Francesco Regazzoni, Christof Paar, and Wayne P. Burleson. "Stealthy Dopant-Level Hardware Trojans⋆." http://people.umass.edu/gbecker/BeckerChes13.pdf

CPU speed

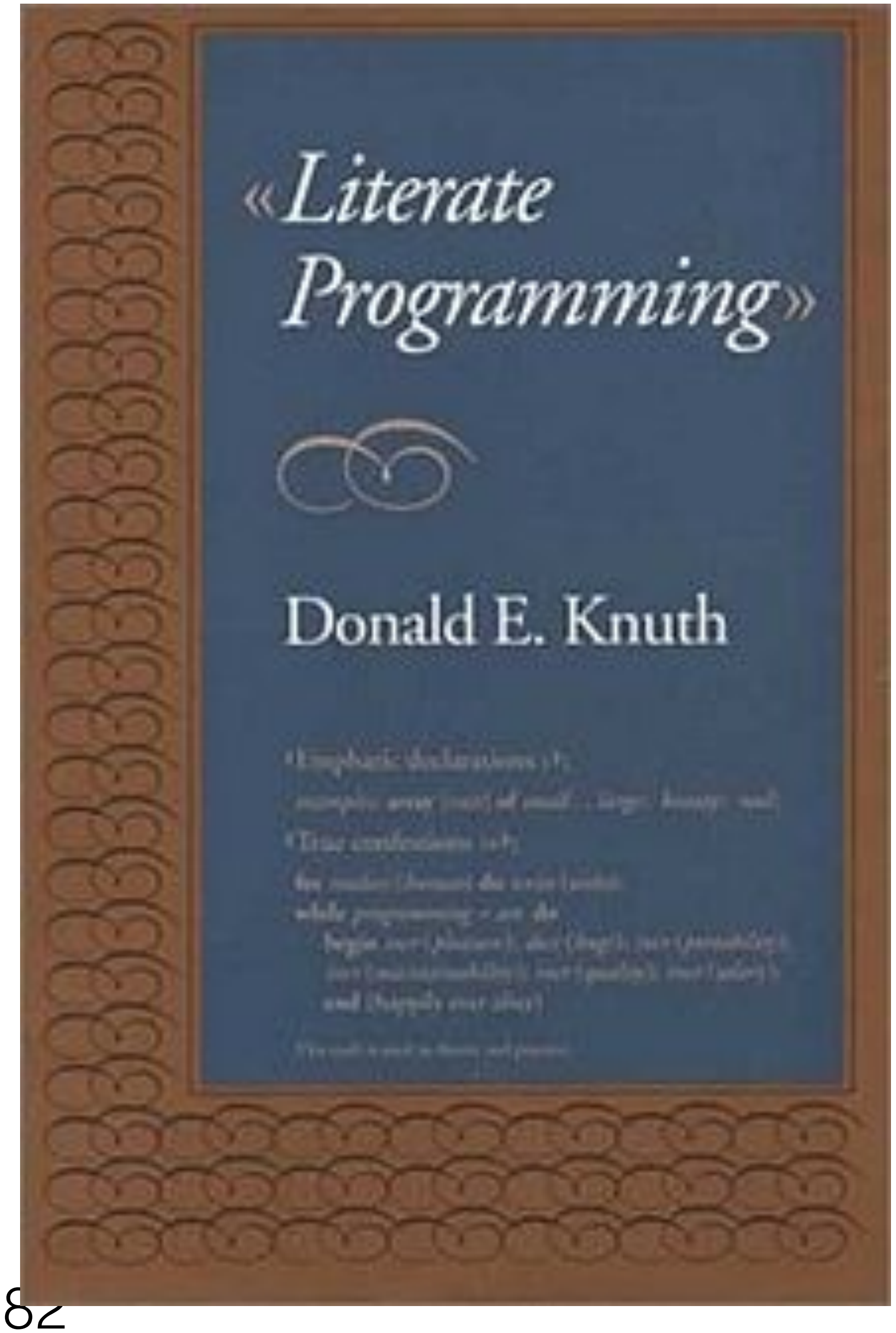
- No 100GHz Intel Octium processor
- Plenty of power for client crypto
- We could eschew a lot of CPU complexity for audit-ability and reliability
- We could use cores as separate machines, instead of coprocessors. Separate cache and memory, too.
- ARM processors?

What works: personal responsibility for the code

- Knuth’s personal checks
- Dockmaster: if someone breaks it, you are fired

Works: Literate programming

- You write a document that explains the program, algorithms, etc., with code embedded in an order natural to the description, not what the compiler wants.
- *weave* and *tangle* generate a document and a program
- Imagine a kernel lovingly described and written in this form.



«Literate Programming»

Donald E. Knuth

[illegible]

What works: software "annealing"

- Sendmail
- Postfix, in beta for a year
- ssh and its protocol

Strong type checking

- My experience with BASIC, FORTRAN
  - Dykstra, then Pascal
- Too bad C won: my choice was Modula 3 or Oberon, perhaps
- Small is still beautiful.

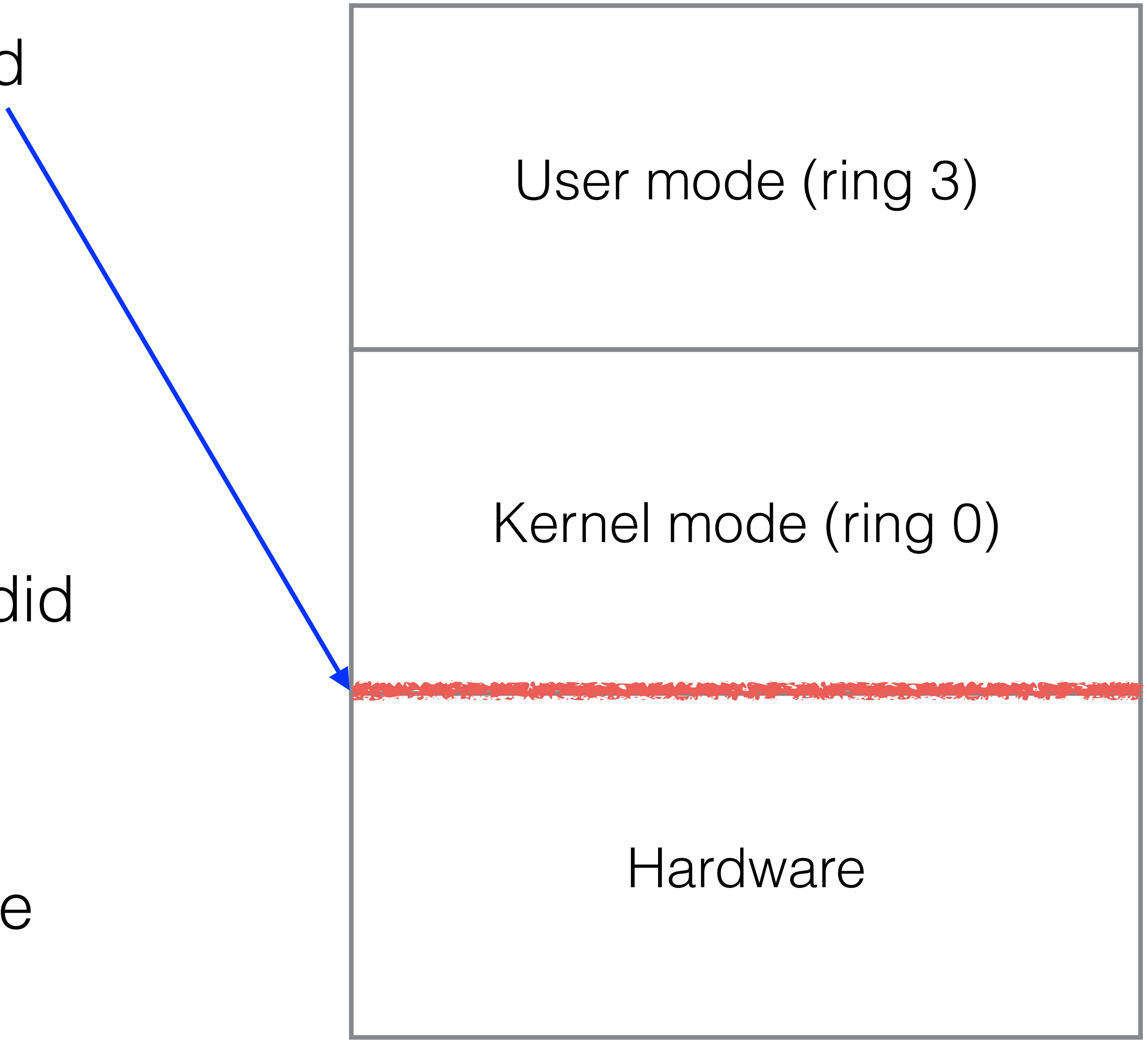
Are VMs okay?

- Yes, but there is a very weird security line there

- Kernels and the hardware have always been intimate pals

- If we throw away that trust, did we find all the hardware weaknesses?

- Also, DOM0 is an awful large entity to trust.



What works: 4 digit PINS!

- Why? Limited tries
- Robust history of success
- Only a few PINS need to be illegal

What works: trusted path

- how do you know you are talking to the trusted operating system?
- ctrl-alt-delete was an example
- out-of-band PIN
- make standard screens slightly taller than movie aspect ratio (16:9), and dedicate a bottom strip to trusted system messages

Building a computer from scratch

Goal: be like a wise man who built his house on the rock

- Trusted hardware
- Trusted firmware
- Trusted OS
    - trustable sandbox

The hardware is a problem

- Relies on the trustability of the design and fabrication
- Changes to circuits by malefactors or National Security Letters
- Confident auditing of the final chips is worthwhile, but is very hard
- Good news: CPUs could be quite cheap

Software layers

- Proved correct: BIOS, kernel, compiler, libraries, sandboxes
- Peter Neumann and others have been working on this since at least the 1970s.
- Expensive, but cheap when amortized over the whole user community.

Sandboxes have to be rock-solid

- Data may be need to be saved in a specific way between instantiations
  - Browser cache, history, cookies, etc. This is a tough problem
- Applications that want to break the sandbox will not work on the machine
- Such a machine is not for every one, but you probably don’t want to do banking on another one

Some special purpose systems already try to do this

- aerospace and aircraft
- medical devices, but many use ancient Windows software as a trusted computing base **(It Works!)**
- Controller hardware, esp. since Stuxnet.

Other solutions, if your hardware is ok

- Live CDs and thumb drives.
  - Bank with a CD/thumb drive from the bank
  - Provenance is an obvious attack

Who Are You Gonna Call?

- Hyper-careful industrialists
    - Dean Kamen (insulin pumps, wheelchairs)
    - Elon Musk (rockets)

Where Might the Solutions Come From?

- Spooks
  - NSA
    - yes, NSA. Remember Linux SE?
  - DISA/contractors in support of NIPRNET/SIPRNET/JWICS and others

Academia

- A lot of solutions aren’t real-world ready, but many are
- There are grants available in these areas
    - The politicos would *love* viable solutions

“Legacy” suppliers

- MSFT, AAPL, INTC, AMD, etc.
- a no-virus guarantee?

Yeah but

- People make buggy code
- Programming bugs imply security bugs
- There is no evidence that our code is getting less buggy
- General computing has many requirements, and they change too often
- Karger/Thompson: On Trusting Trust

Yeah but

- Governance is a big concern
- Did your hardware provider get a National Security Letter?
- National debate and resultant policy, enforced

Yeah but

- Still have DDoS
- People can still be fooled
  - phishing

I think we can win

- It is our hardware, and our software
- We have the home-field advantage
- Correct software can be implemented, if we are very careful

Security: I Think We Can Win!

Bill Cheswick

(Your institution name here?)

ches@cheswick.com,
http://www.cheswick.com/ches/talks/APPsec2013.pdf