Beyond embedded: what could go wrong?

Sergey Bratus

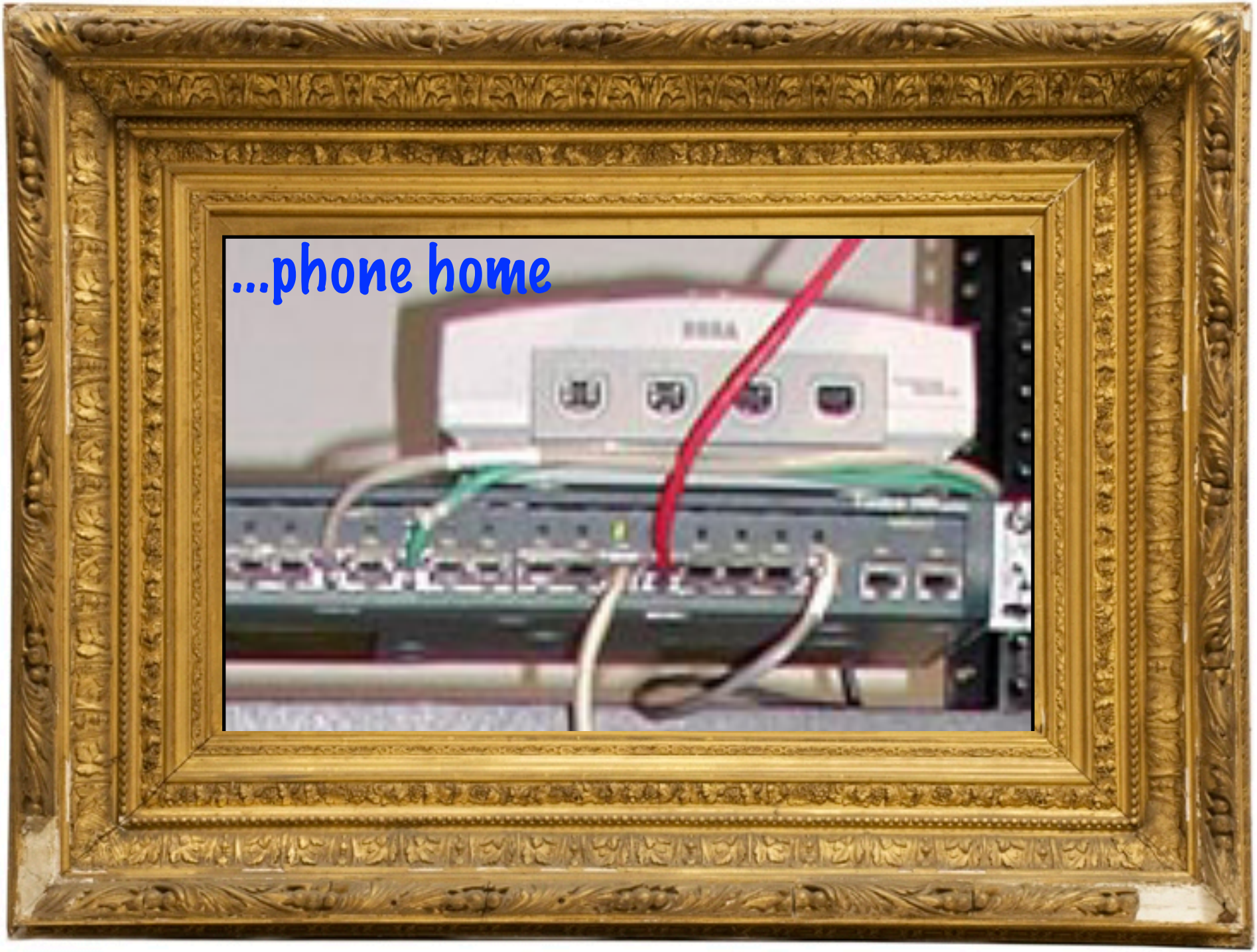
PKI/Trust Lab, Dartmouth College


The Past

The Past: Innocuous but Insidious

* Embedded systems as innocuous-looking bots/relays behind the perimeter
  * "DC phone home", BlackHat Vegas, 2002
  * The Undetectable Packet Sniffer (UPS), Defcon 11, 2004
  * iPAQ, Linksys boxes, ...

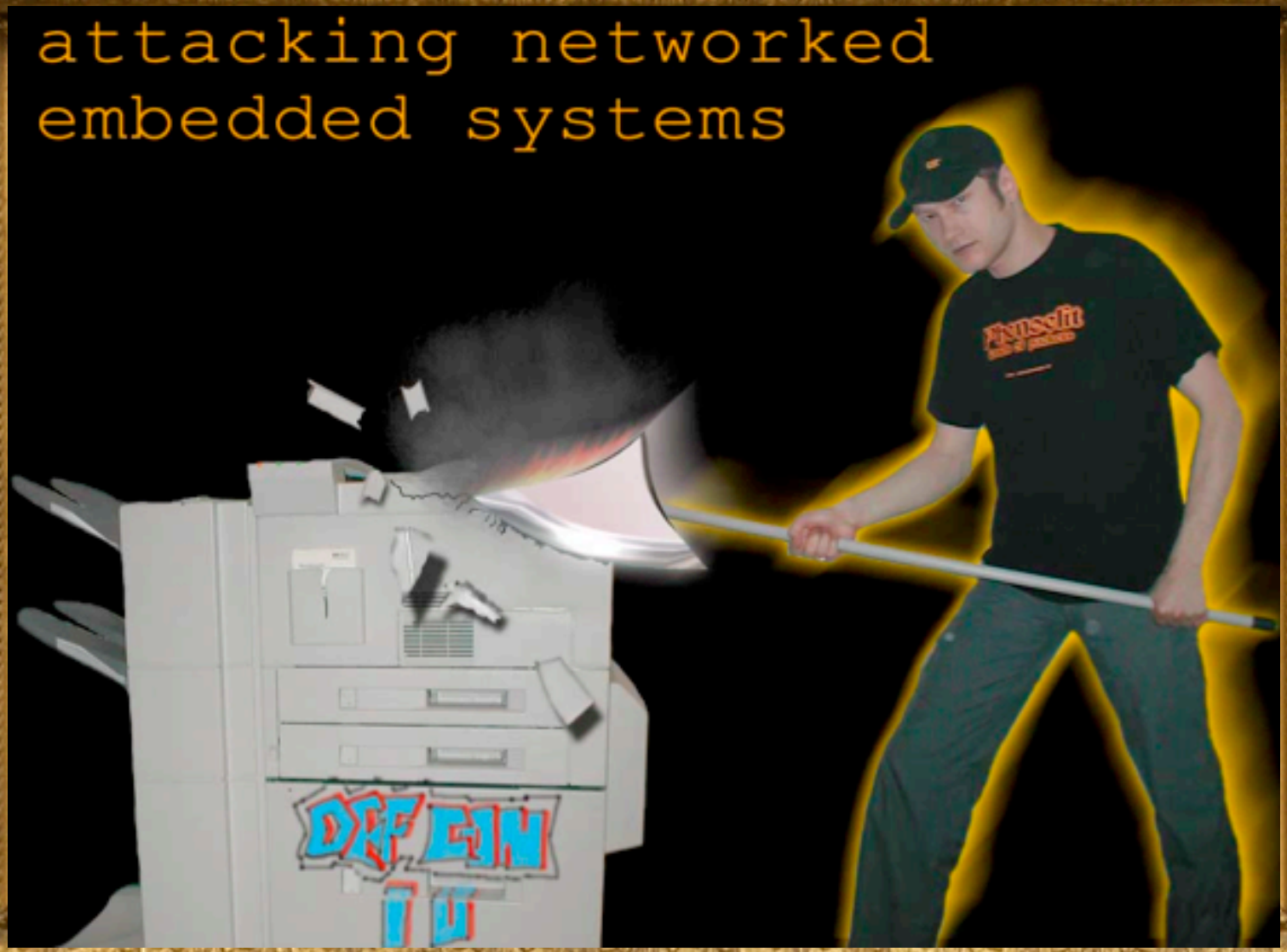

DreamCast...


...phone home

The Past: Remotely Rootable

* Custom networked operating systems are just as vulnerable
  * Switches, routers, printers
* "Attacking Networked Embedded Systems", FX & FtR, Defcon X
* Mike Lynn's "CiscoGate", IOS shellcode

attacking networked embedded systems



The Past: Naked in Public

* Sensitive functionality misconfigured in public due to lack of knowledge or neglect
  * ERNW: "Digging into SNMP in 2007: An exercise in breaking networks" HitB 2007 Dubai
  * CISCO-TAP2-MIB wiretapping/traffic interception exposed on Cisco uBR 10000 by a large ISP

ISO (1)

Organization (3)

DOD (6)

Internet (1)

ERNW
Living Security.

Management (2)

Experimental (3)

Private (4)

MIB-2 (1)

Enterprise (1)

System (1)

Interfaces (2)

Address Translation (3)

IP (4)

ICMP (5)

TCP (6)

UDP (7)

EGP (8)

CMOT (9)

Transmission (10)

SNMP (11)

Internet Activities Board (IAB) Administered

Proteon (1)

IBM (2)

Cisco (9)

HP (11)

Wellfleet (18)

Sun (42)

Apple (63)

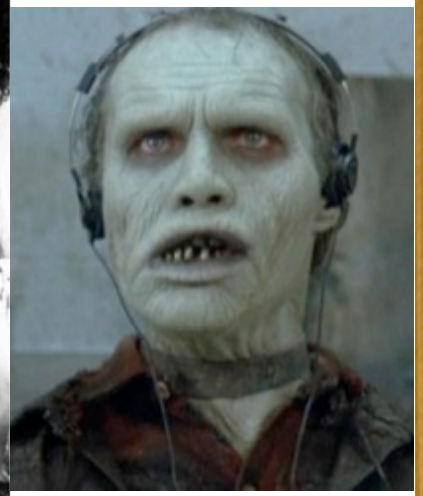
Vendor Administered

The Past: Double-dealing

* Embedded device may appear to be working fine, while doing evil on the side
  * Graeme Neilson, "Netscreen of the Dead"
  * Trojaned firmware

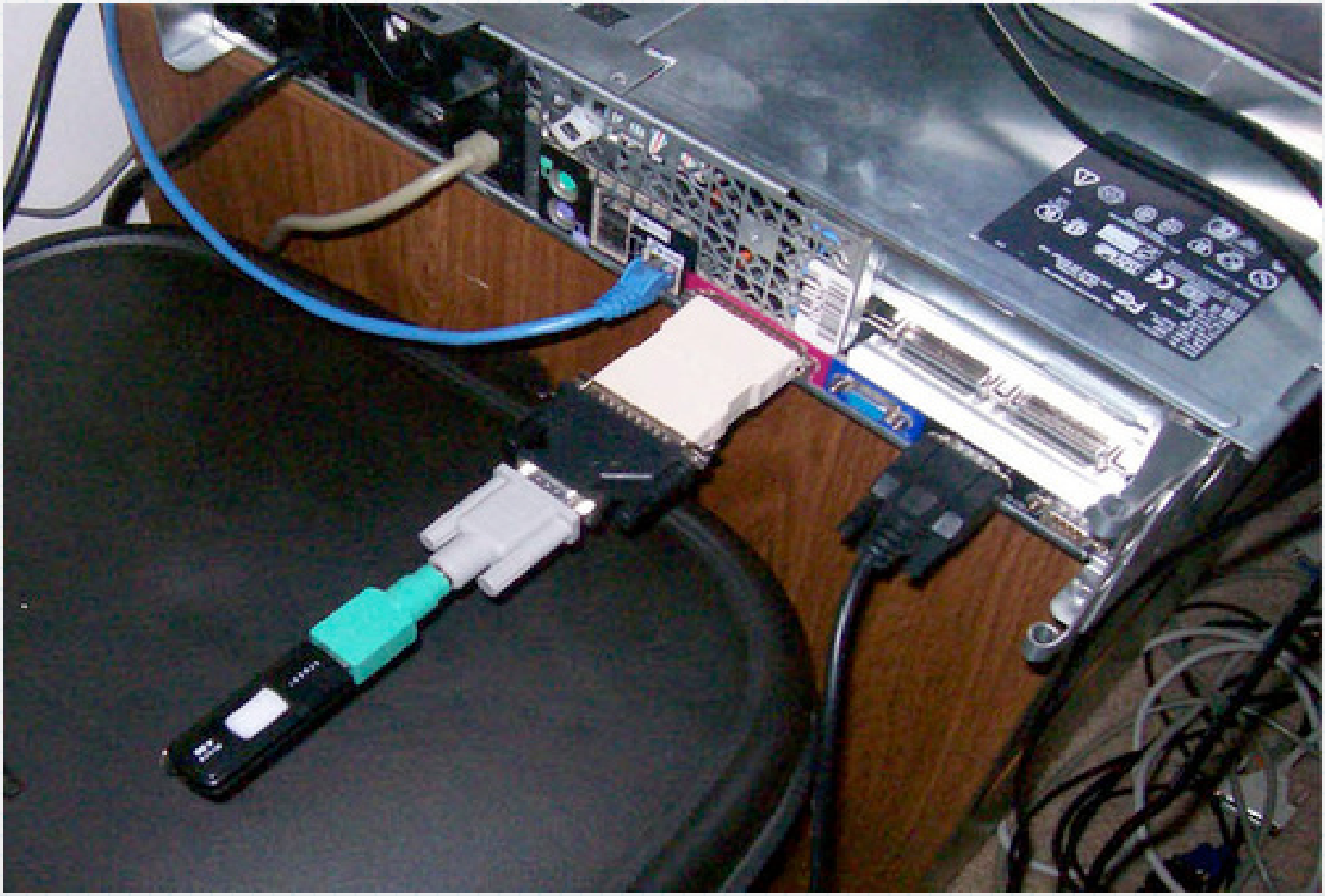
Netscreen of the Dead:

Developing a Trojaned Firmware for Juniper Netscreen Appliances



The Future

The way we build things...



The Future?

The Future?

* The illusion of "saving money with computers"
* Home energy management? / "Smart Grid" ?
* Medical devices? / Remote health care?
* you name it...

A radio-controlled defibrillator?

* Kevin Fu et al., Defcon 16
* Once past the software radio analysis, the protocol is PLAIN text
* Have a programmer, will reprogram hearts

(2b || !2b) * 100M

* To remote admin or not to remote admin?
* To trust or not to trust (the network environment)?
* To trust or not to trust (remote systems)?
* Will old engineering solutions scale up to 100M?

When we have 100M computers...

How do we extend trust to them?

How do we keep all of them trustworthy?

When we have 100M computers...

* Should they have remote administration interfaces to get configured, patched, and upgraded?
  * YES: huge network attack surface
  * NO: be prepared to lose/replace entire generations, often
    ["evolution" = "stuff dies out"]

-- Dan Geer, SOURCE Boston, '08

When we network 100M computers...

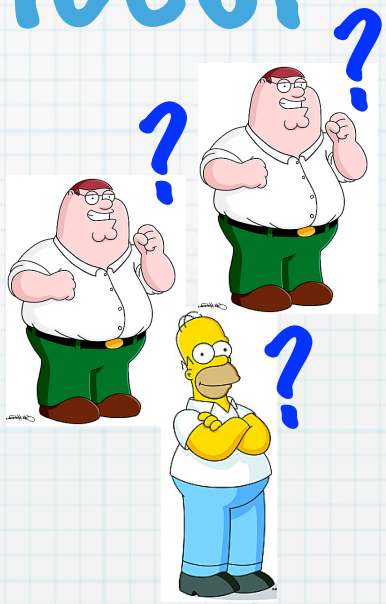
* How do we commission/config/replace them?
  * Must be easy, not require special training (e.g., in a Home Area Network)
  * "Plug it in, it just works" =>
* Devices must TRUST their network environment to learn configs from it

"Just trust the first message"

* The only way to authenticate a message is to share a secret (or public key) with the trusted origin/environment
* How will this secret get to the new device?
  * human_op * 100M =

Can we authenticate 100M devices?

* Old style auth: what you {have, know, are}/ {lost, forgot, used to be}



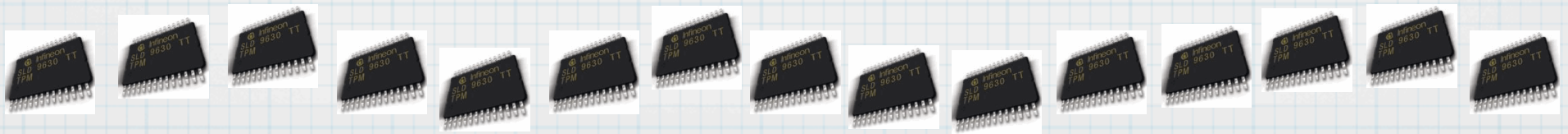
* What would managing 100M keys cost?
* PKI experience: keys may be costlier than devices!

"C", confidentiality: Crypto Chicken vs Egg

* Key material to secure link layer (L2)
* ...is exchanged via protocols in L3!
  * programming with drivers/frames rather than sockets sucks

"I", integrity: Run twice as hard to remain in place

* How much to:
  * push patches * 100M = ?
  * runtime integrity computation CPU cost * 100M = ?
  * maintain white list of trusted configs ?



...and other fun adventures...