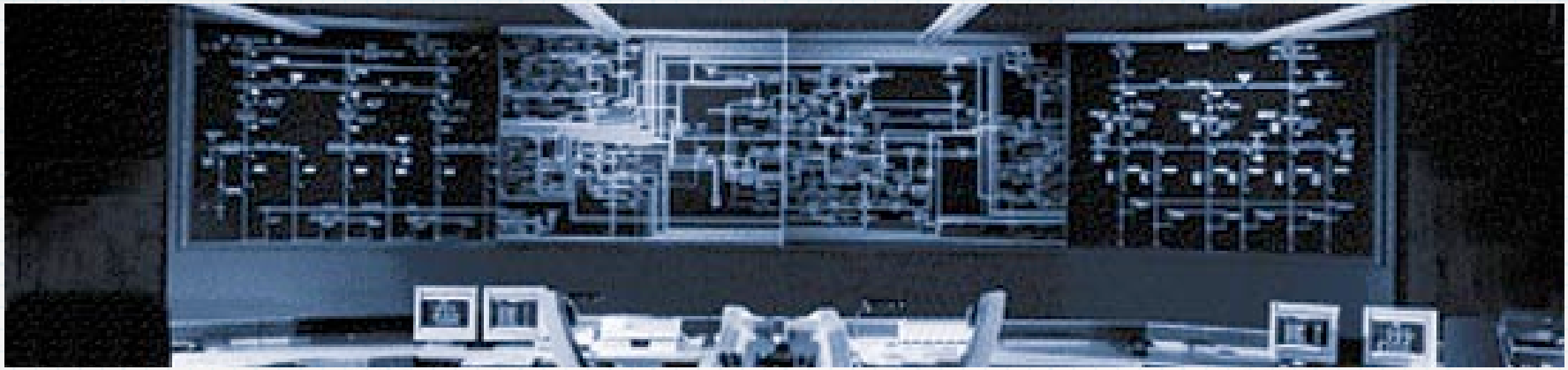


# Adventures in SCADA


Sergey Bratus
Trust Lab, Dartmouth College

Edmond Rogers ("bigezy")
a Fortune 500 utility company ->
University of Illinois' Information Trust Institute

# What this talk is not


* No Odays
* No vendors named
* No Stuxnet

# No Stuxnet ?!

| B | I | N | G | O |
|---|---|---|---|---|
| 12 | 18 | 41 | 47 | 61 |
| 7 | 26 | 39 | 54 | 70 |
| 4 | 27 | FREE 4785 SPACE | 49 | 63 |
| 5 | 23 | 35 | 58 | 73 |
| 3 | 30 | 32 | 52 | 75 |



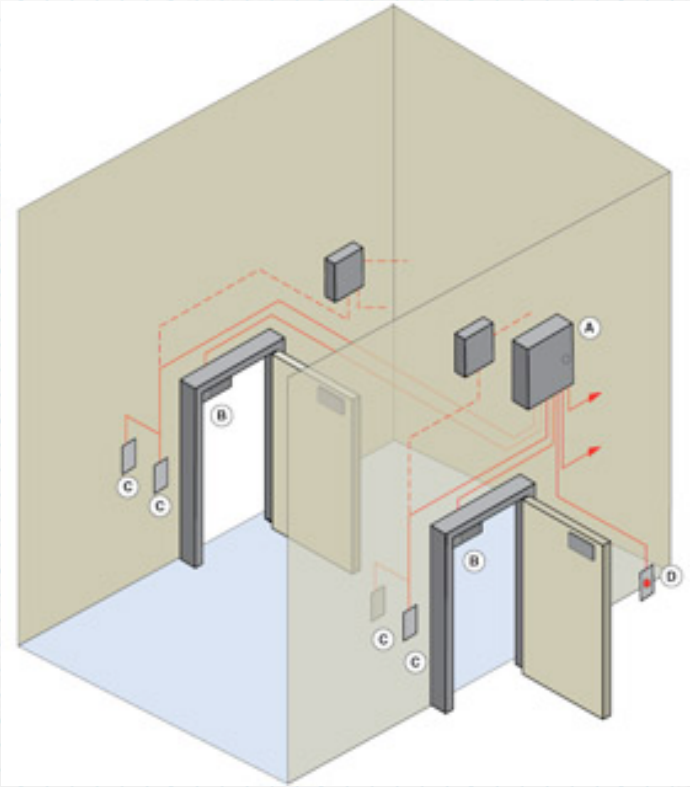
( Goto 27c3 x2:
Bruce Dang, FX )

# "SCADA in the wild"

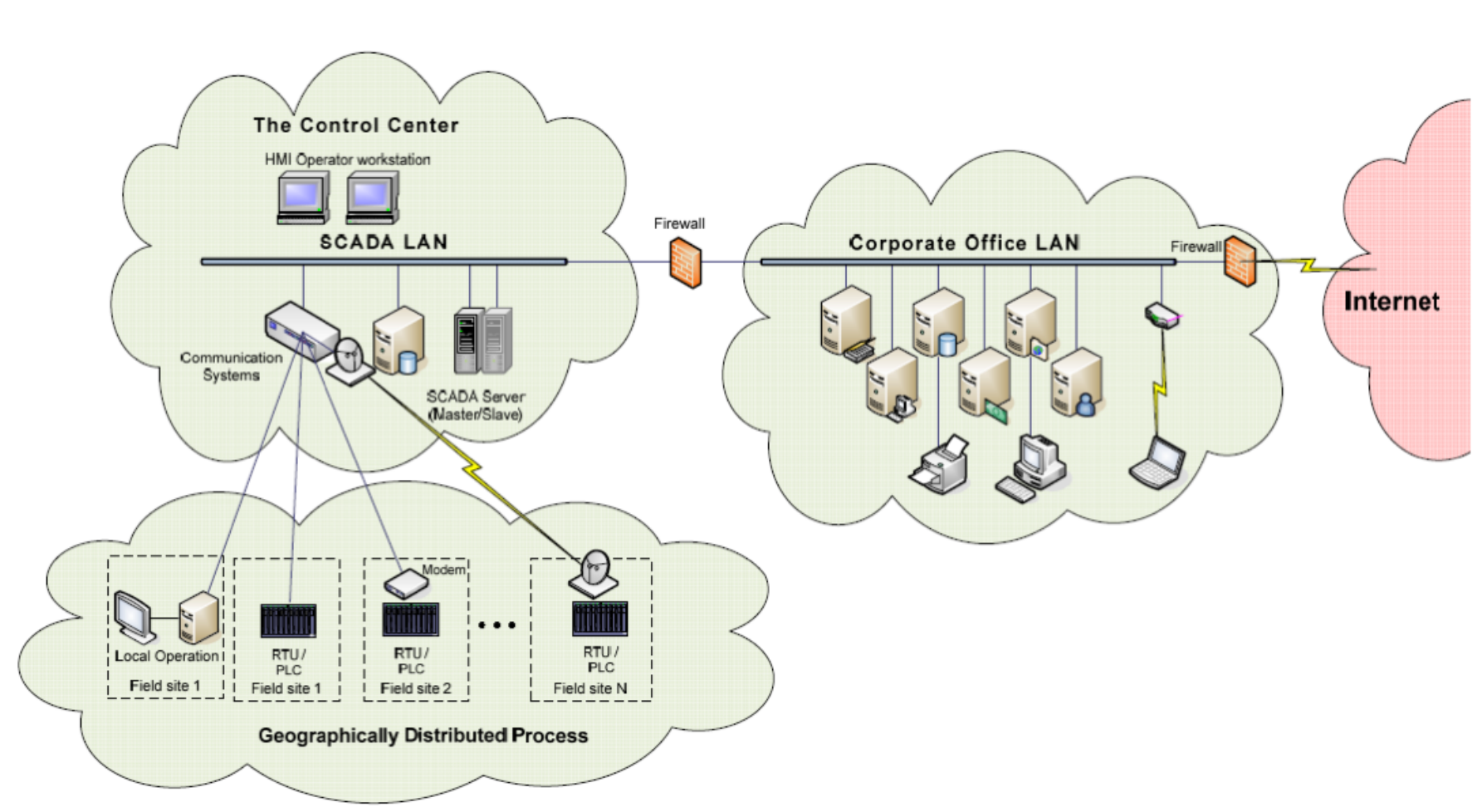
* Seeing SCADA equipment/software in its natural habitat
  * it's cruel to isolate them from their natural inputs & surroundings :)
* Seeing the operations of a control network
* Fuzzing with no target instrumentation & no protocol spec

# Bonuses

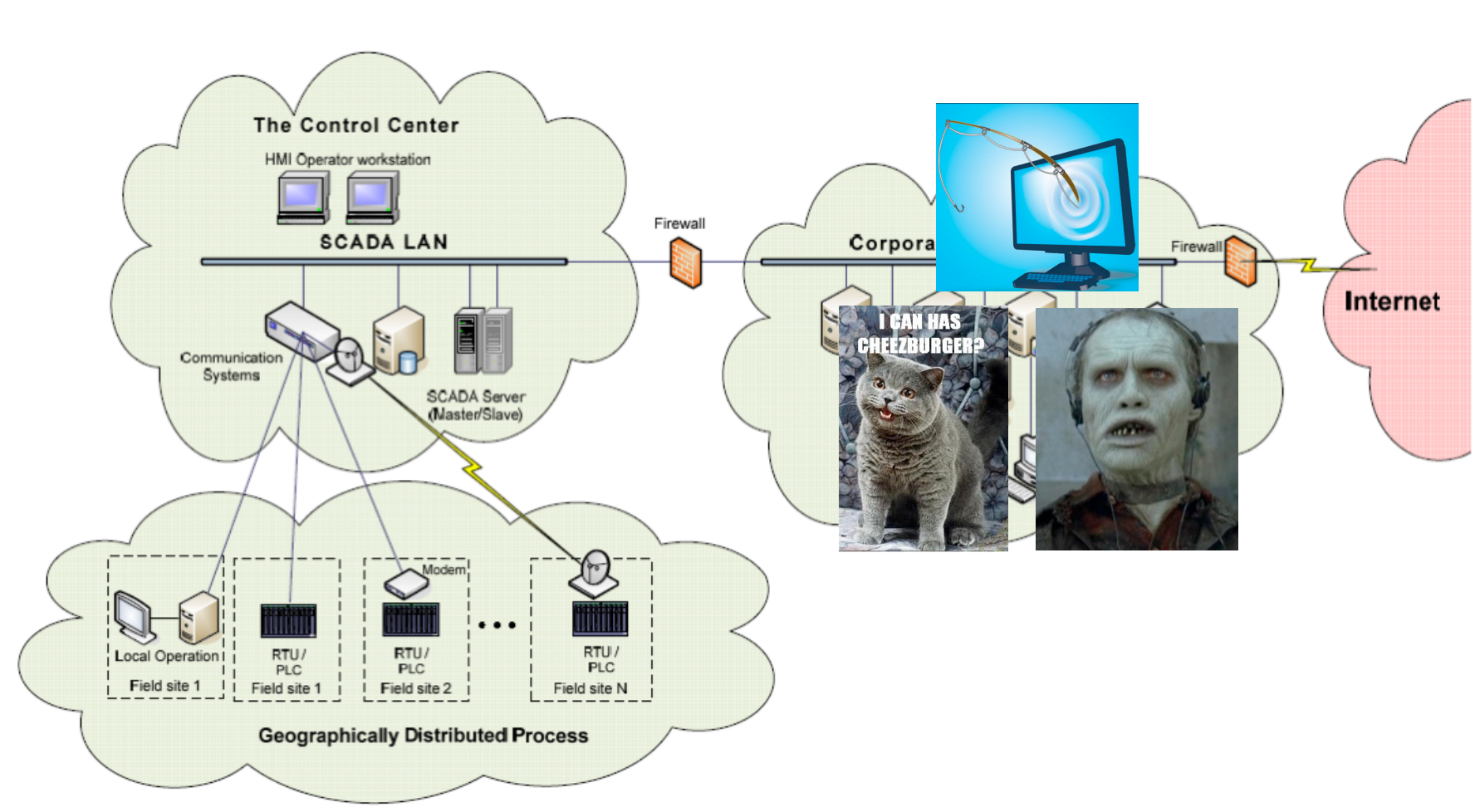
* Going through a man-trap to get to a network port
* Fuzzing across state lines
* Fuzzing $100K+ systems
* Finding out what waking up for work at 6am feels like :)



# What the jungle looks like

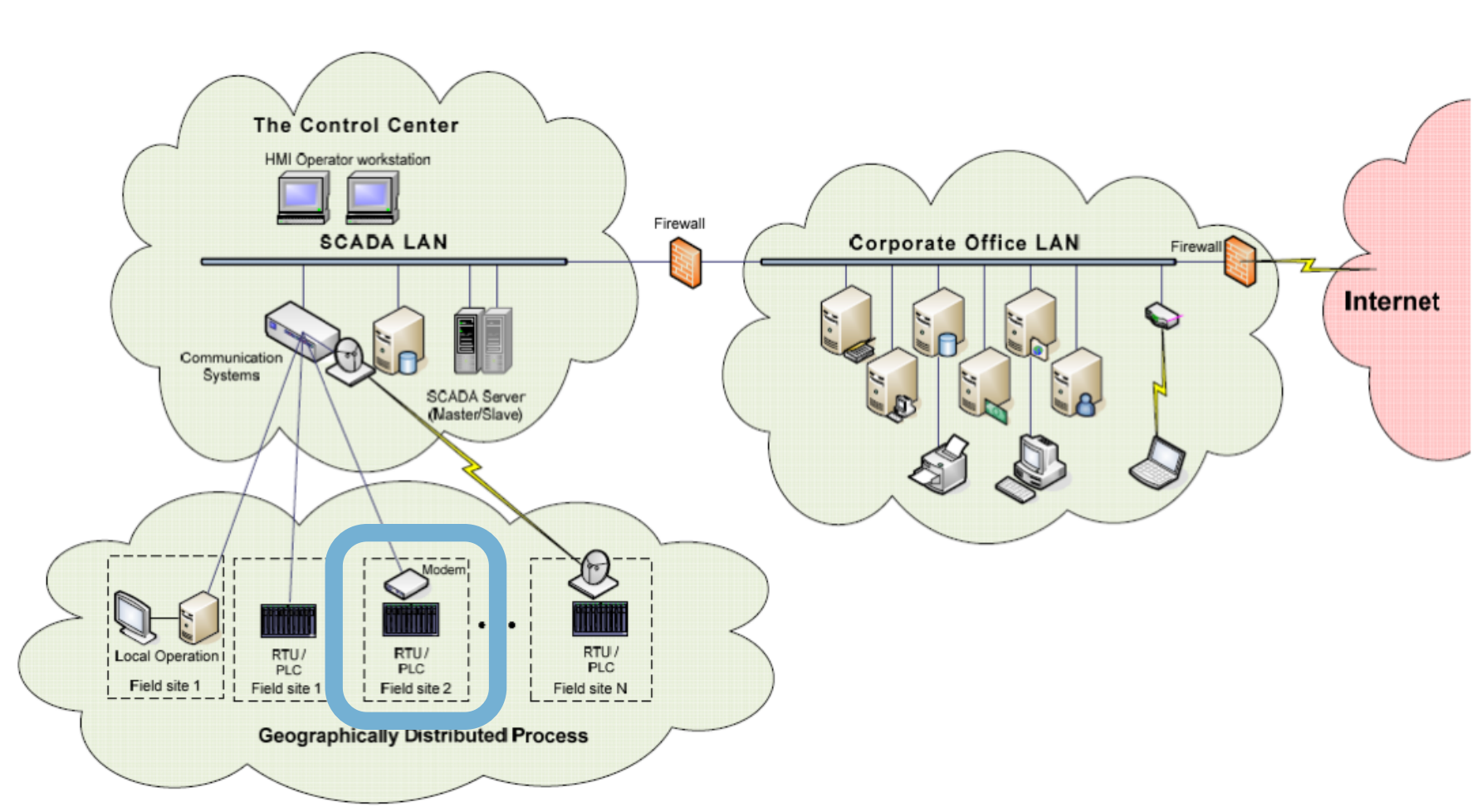


# What the jungle looks like

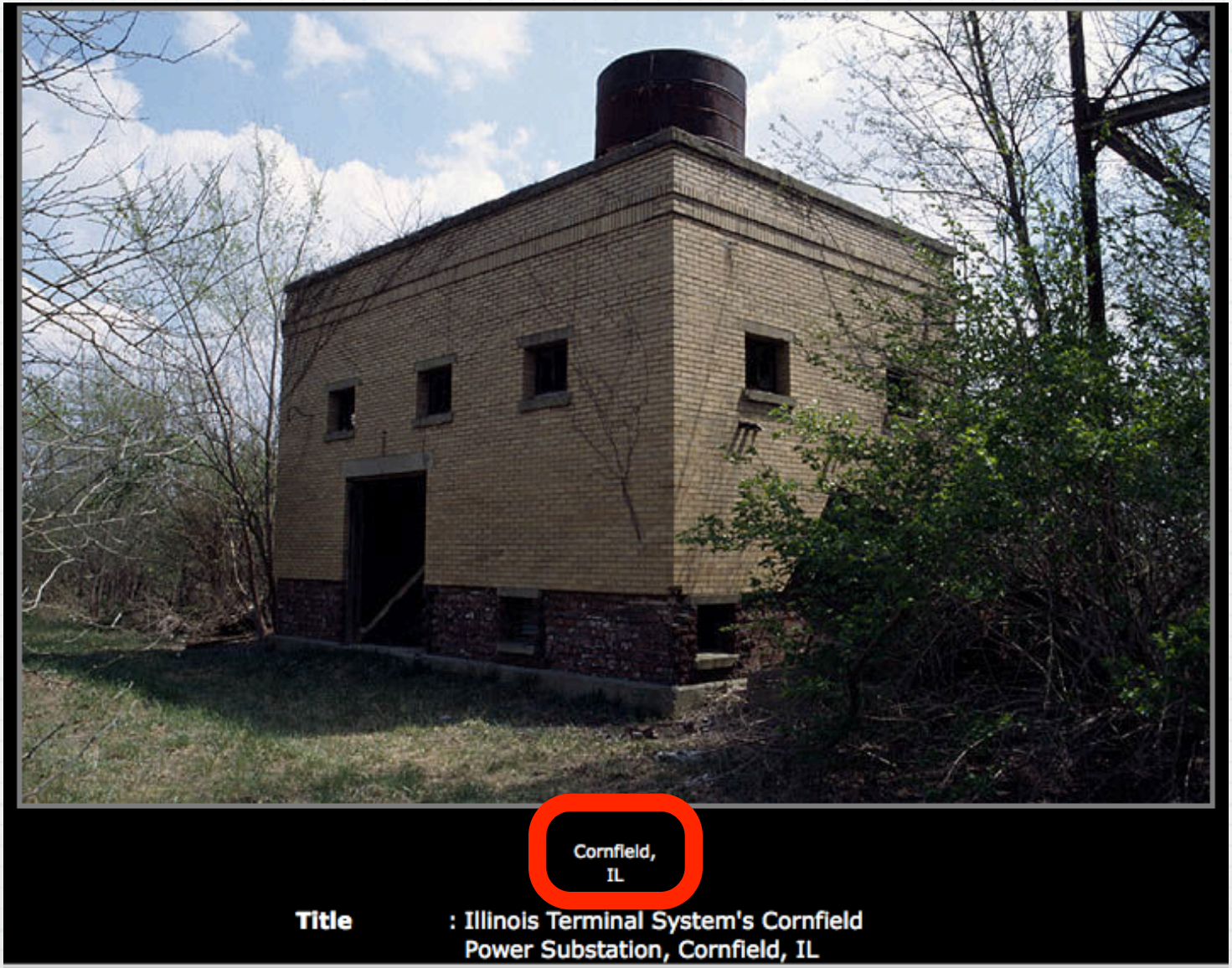


# Legacy: it's still there

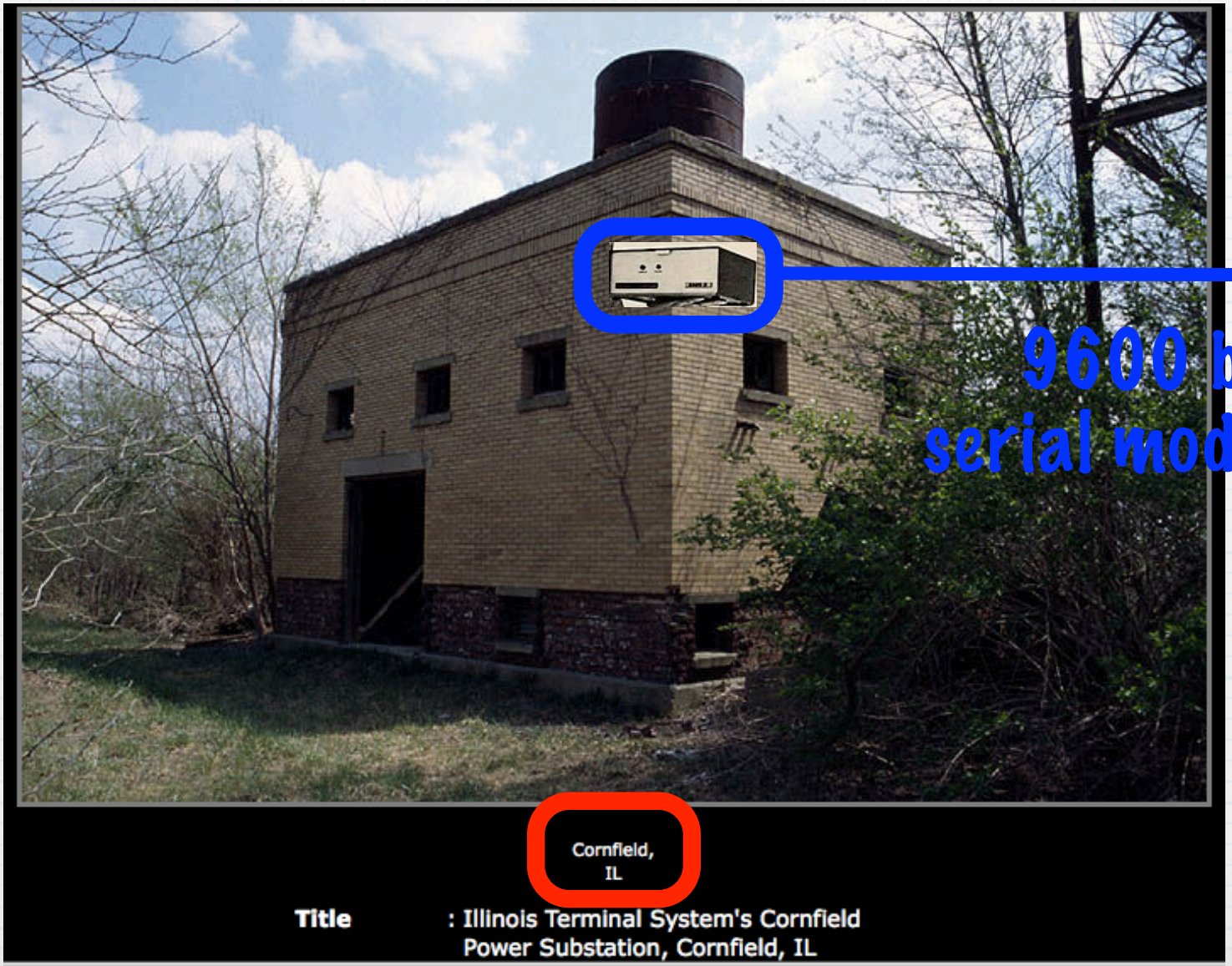
# What the jungle looks like



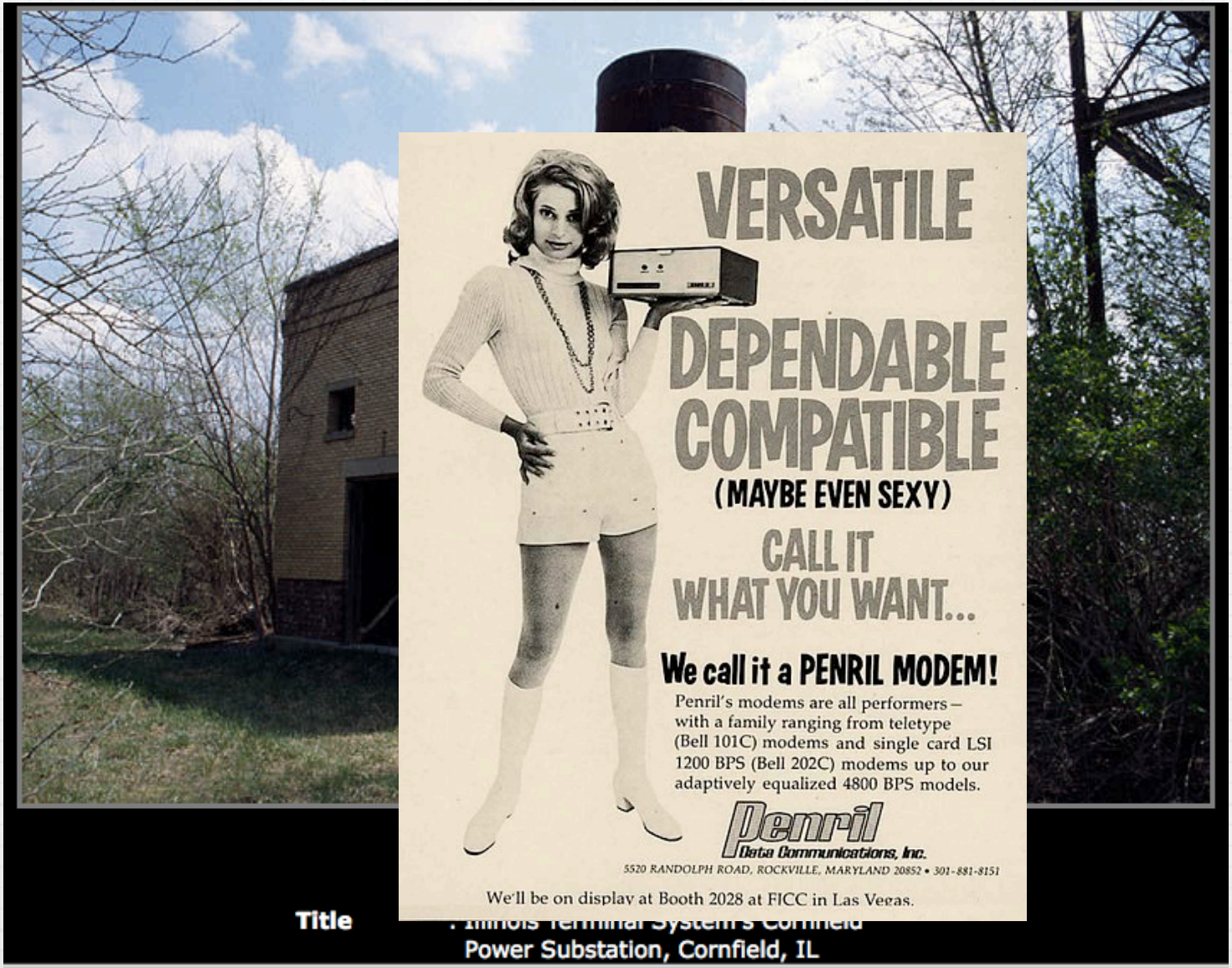
# "Substation in a corn field"



# "Substation in a corn field"

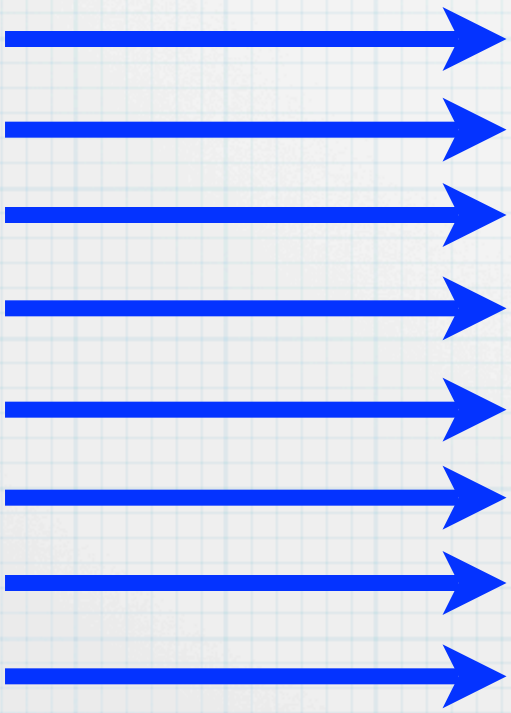


# "Substation in a corn field"



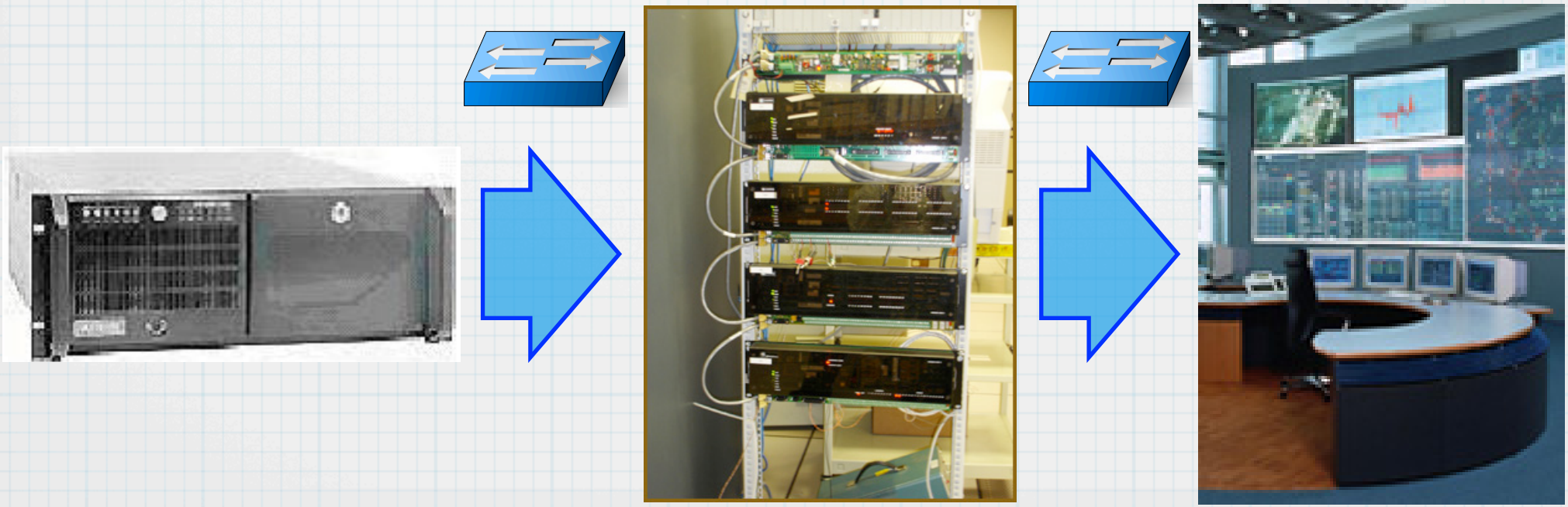
# Meanwhile, at the Control Center...

* Some 100+ modem lines terminate at the "Front End Processor" (FEP)



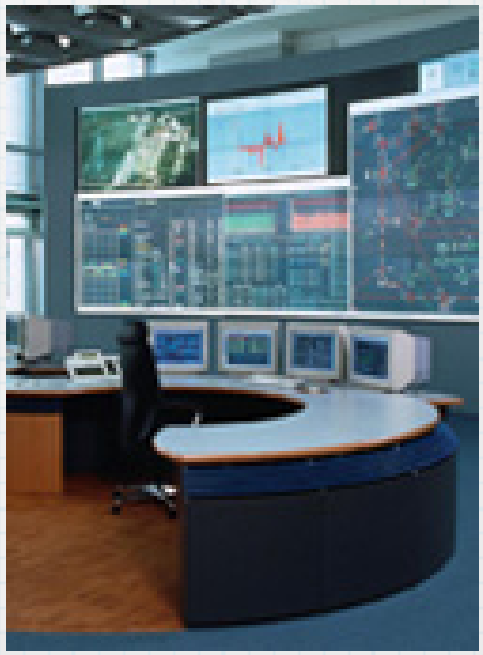
# Meanwhile, at the Control Center...

* Front End Processor connects to an Energy Management Server (EMS)
* EMS feeds data to boards/workstations



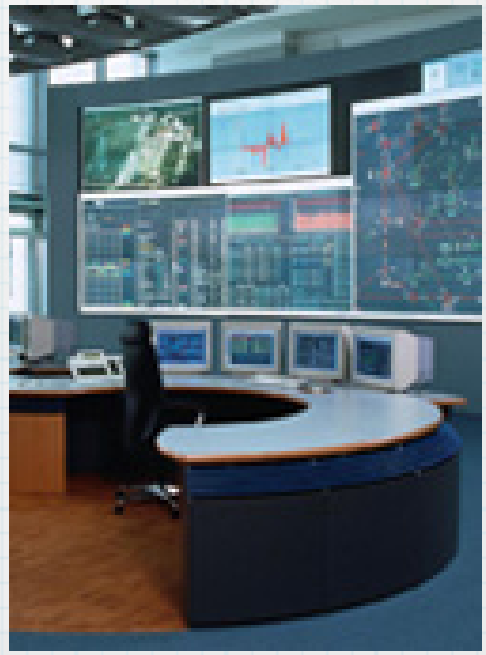
# Meanwhile, at the Control Center...

* Front End Processor connects to an Energy Management Server (EMS)
* EMS feeds data to boards/workstations



# Meanwhile, at the Control Center...

* Front End Processor connects to an Energy Management Server (EMS)
* EMS feeds data to boards/workstations



# "Power ties"

* The closer to the control center, the more proprietary the protocols get
* Sold as (expensive!) integrated solutions ($100K+ - $1M+)
* Asset owners heavily rely on vendors
  * Maintenance contracts, warranty, etc.
* But asset owners can push back, too

# SCADA owners care

* Smart asset owners suspect things might be really brittle
  * Hence serious investment into isolation of control networks (+ IPSec, too)
  * The most paranoid production network I've seen
* ...which was where we came in :)

# The cause

* Utility may spend at least as much on mitigation as on original equipment!
* This research was done to show the need for such strong and meticulous measures
* Defense in depth is only as good as the hole is deep

# Isolated Test Environment

* New devices and patches must be tested before being put into service
* Such a test environment was used as a basis: isolated from production network
* Took a lot of preparation and checking to assemble the right topology
  * with the right geographic distances

# "Fuzzing across state lines"


1: "Your fuzzer is here"  A: "your FEP is here"

(Note: these aren't the actual locations)

# Fuzzing!

Software internals

Crafted inputs

# Yeah, fuzzing SCADA...

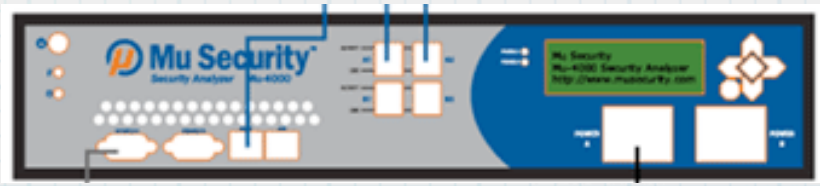
# "Fuzzing SCADA" is old...

* Ganesh Devarajan (TippingPoint)
  * DNP3 module for Sulley the fuzzer
    (Sulley released in 2007 by Amini & Portnoy)
  * Ganesh's BH 07 talk caused much media stir

* Digital Bond's ICCPSic test tools
  * released to "vetted asset owners" subscribers
  * "...will crash vulnerable ICCP servers."

* SecuriTeam's beSTORM DNP3 fuzzer
  * crashed Wireshark's DNP3 protocol dissector/parser

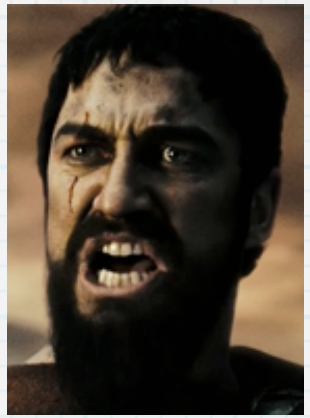
* Mu Security's fuzzer hw appliance
  * Licensed per protocol module



# Problems in the field?

* Proprietary protocols => no block-based protocol modules a-la SPIKE
* Cannot instrument the targets
  (voiding $100K+ warranties is tough)
* Who's going to restart it for us when crashed?
  * > 50% of fuzzing is framework setup

# No problems! This... is... SCADA!



* Protocol transmissions are continuous and repetitive, same structure
  * many samples of data to learn from
* Watchdogs automatically restart failed processes and systems
* Frequent keep-alive/status messages
  * easy to see when targets crash

# More SCADA goodies

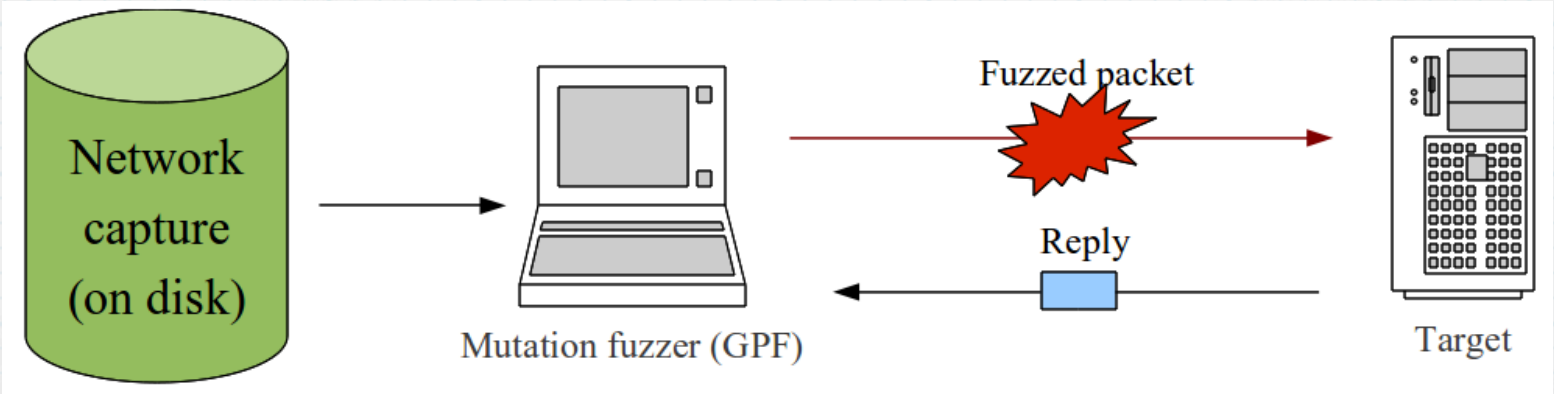
* Distinct handshake phase in protocols
  * skip it to let data connections proceed
  * then fuzz data parsing code
  * easy to recognize with packet regexps
* Similar data, similar packet structure seen over and over
  * really helps mutational fuzzing

# GPF, mutation fuzzing

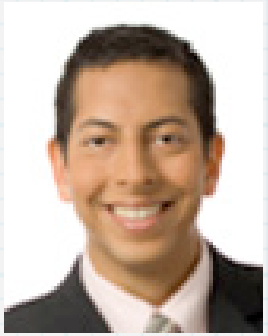
- "General Purpose Fuzzer"

VDA Labs

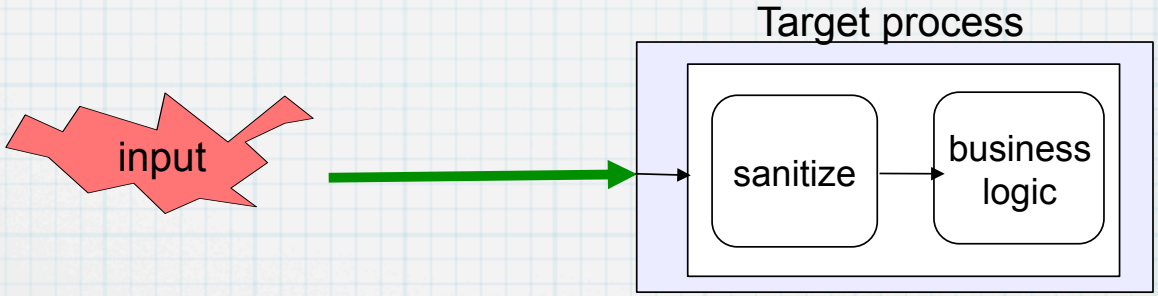
  - fuzzes saved network protocol sessions
  - useful heuristics for inserting runs of random or special bytes



# "Aitel had it right with SPIKE"



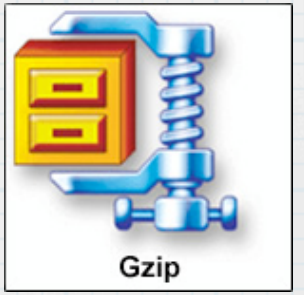
* We'd like to know the blocks of the protocol
  * must match them closely enough to cover code paths past simple sanity checks



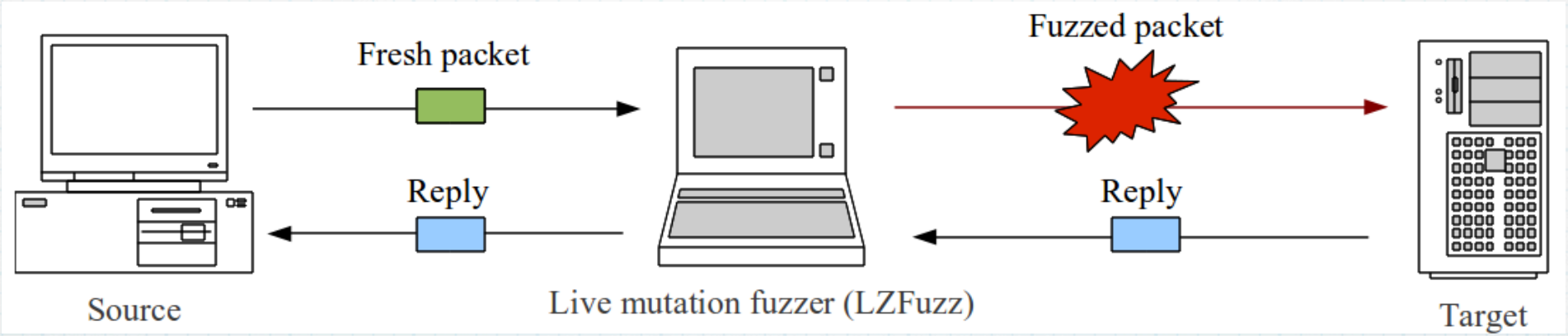

* How to guess blocks of unknown protocol?
  * well, just roughly enough to fuzz them :)

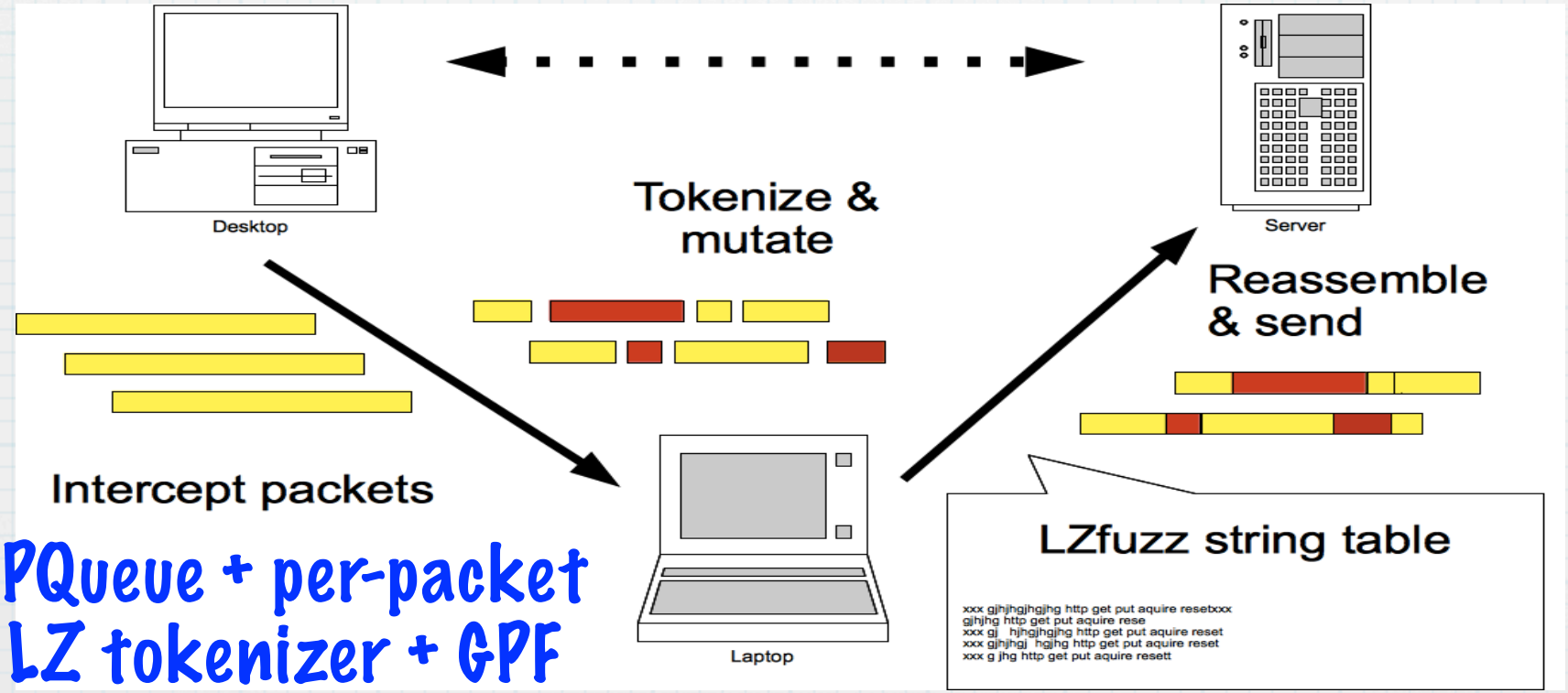
# LZfuzz, a "lazy hack"

* Guesses blocks ("tokens") based on repeated occurrence, a-la GZIP
  * runs a variant of the Lempel-Ziv compression algorithm
  * frequently repeated byte strings end up in a string table
  * seeds the table with likely tokens/blocks from packet captures
* Applies GPF's heuristic mutations to tokens:
  * long ASCII byte runs for buffers overruns
  * extra delimiters, bit flips, ...



# LZfuzz



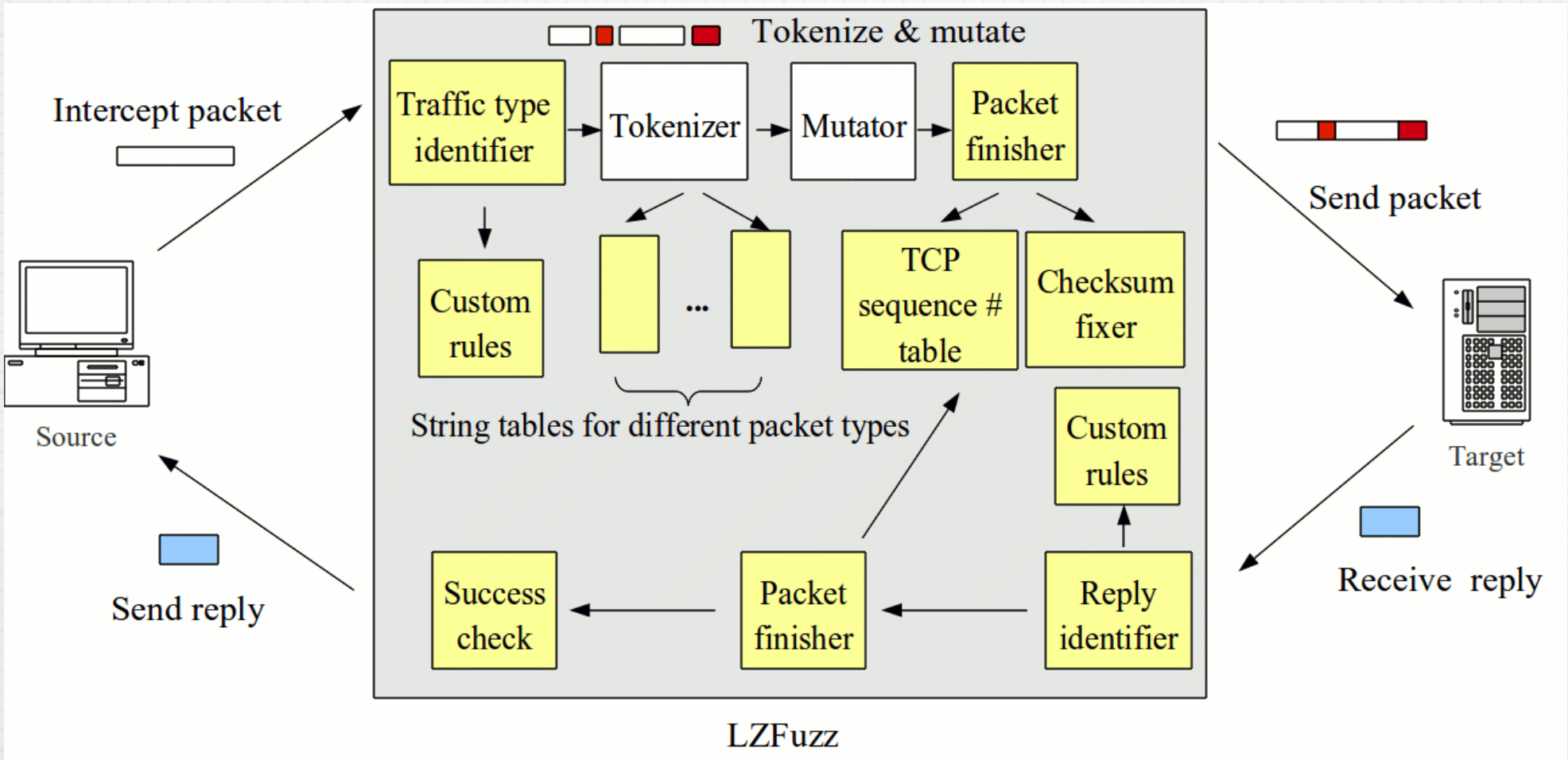

IPQueue + per-packet
LZ tokenizer + GPF

# Recap

- Cannot instrument endpoints, must infer state of target processes/OS:
  - unexpected TCP RSTs, repeated SYNs
  - special auth handshakes pre- data sessions
  - timeouts
- Must adapt & back-off to allow watchdogs to reset targets & rebuild connections
- Must hypothesize checksum kinds & places

# LZfuzz 2.0

* Connection state inference rules
* Automatic checksum detection & fix-up

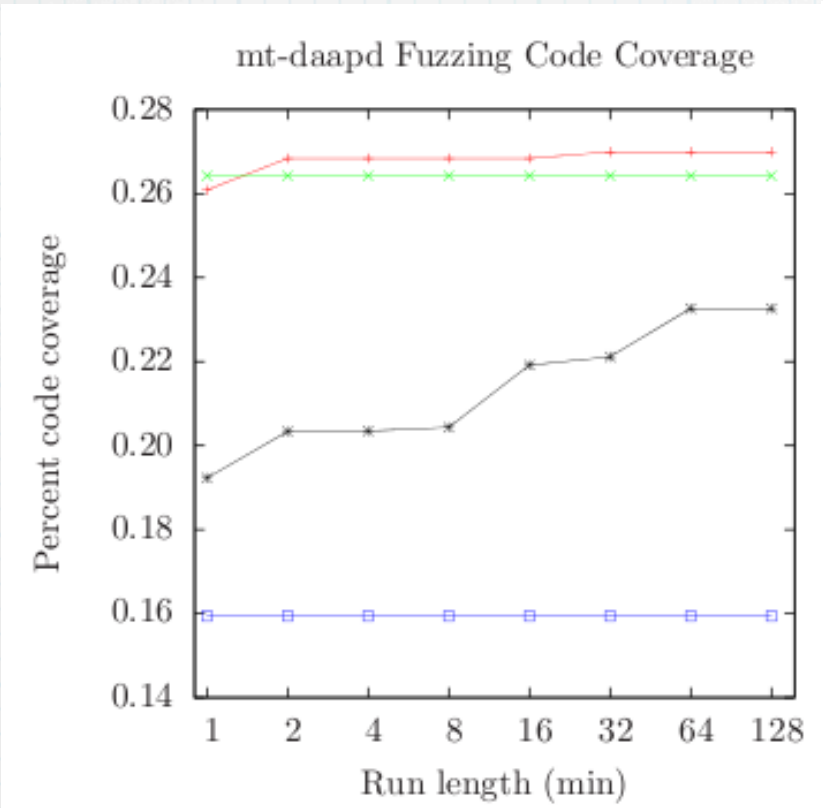


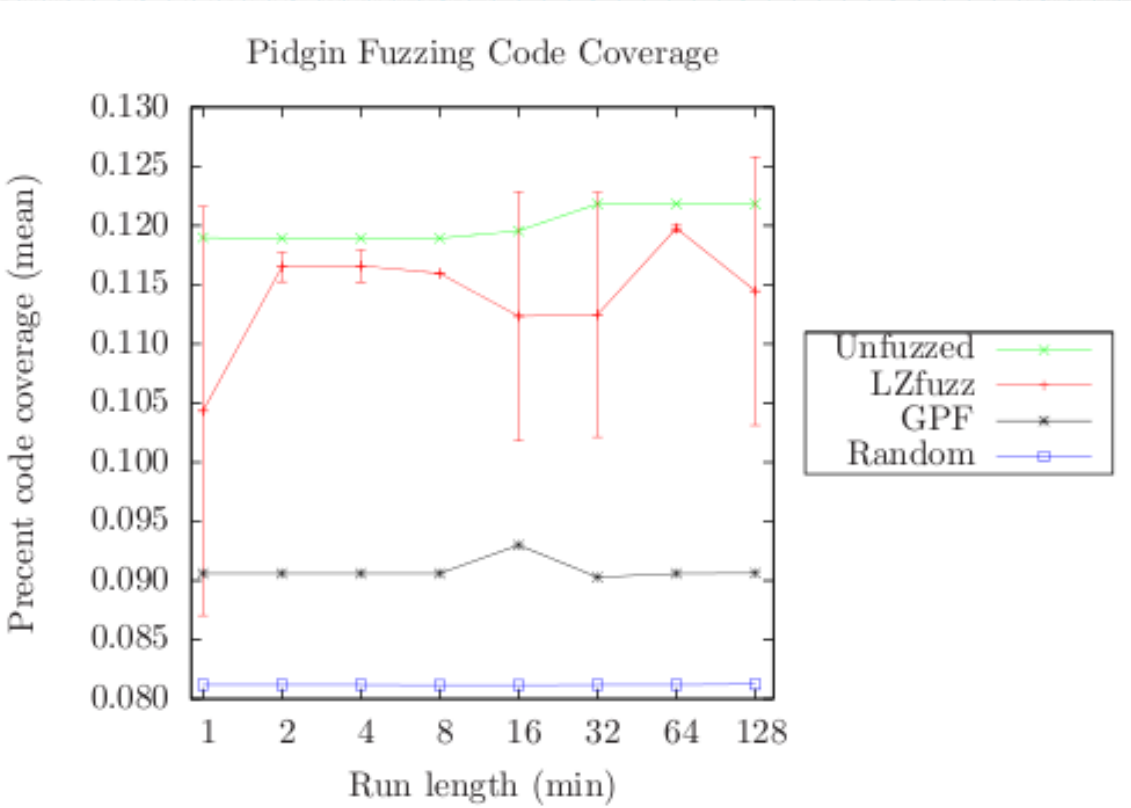
# Coverage?

* Tried non-SCADA targets:

DAAP (iTunes)

OSCAR (Pidgin)



# Validation for utility

* Mitigating controls to prevent injection of packets into the control network

* Paranoia justified

# The future?

# The future?

# The future?

* Composition is how humans do engineering
* But "Security is not composable"
  * Composing well-understood parts may yield a new system with deadly properties
* "Complexity Kills"

# “Wrong threat model”

# Smart Grid!

* It's "smartER grid", thank you very much
* "Tens of millions" of devices!
  * or 100M, whichever you feel like
* Not just "smart meters": phasors, relays, "intelligent electronic devices", ...

# (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 (e.g.,: IPv6 auto configuration)

# "Just trust the first message" vs. key mgmt

* 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?

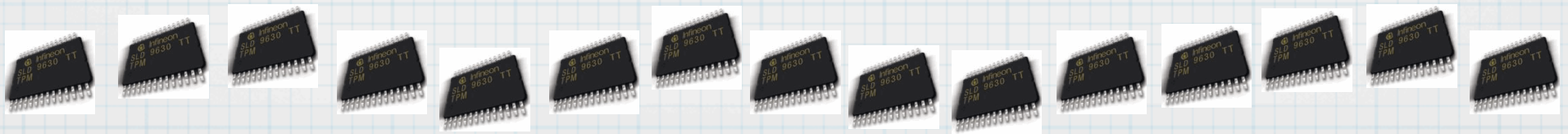
- What would managing 100M keys cost?
  - support
  - remote replacement?
- A utility's PKI experience: keys are costlier than devices!

# "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...

# Thank you!

# More Information

More research & industry interaction info:

Trustworthy Cyber Infrastructure for the Power Grid (TCIPG) project:

http://www.tcipg.org/


Disclaimer: This talk presents only the authors' positions, not those of sponsors or other organizations.