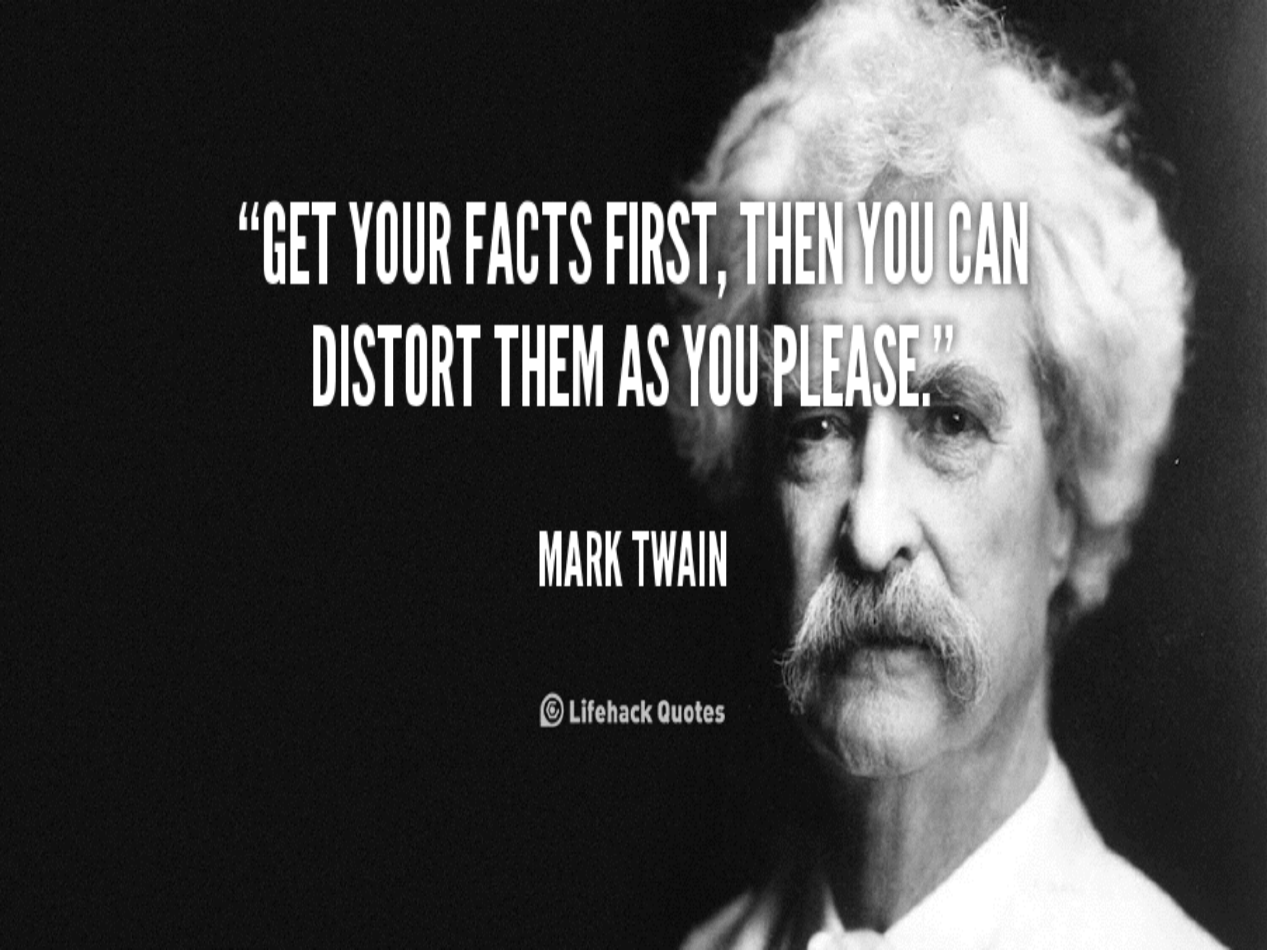
The Board Room guide to hacking

Vincenzo Iozzo

[redacted]



"GET YOUR FACTS FIRST, THEN YOU CAN DISTORT THEM AS YOU PLEASE."

MARK TWAIN

Are you compromised?

Yes

Why is everyone compromised?

1. Your network is a replicable monoculture

2. Compromising is a one-way street: You can't "un-compromise" something

3. The internet and your network are a graph of trust: compromising is viral and exponential

4. Your defense is reactive and slow, it must be proactive and fast

LOOK IN THE
MIRROR...
THAT'S YOUR
COMPETITION.

Monoculture

- The attacker can download the same software you have and attack it until he finds a way in.
- An attacker can replicate an almost-exact copy of your machine and go at it until he finds an “in”
- Once the attacker is on a machine he can experiment and explore the trusted neighbors until he finds an “in”

NO
EXIT

“Un-compromise”-able

- A maxim: there’s always a deep enough level in a machine that is not defended/defendable

- It used to be the kernel, now it’s the bios, the firmware, the hardware, the secret co-processor, you name it

- You can’t “un-compromise” because it’s impossible to know what’s compromised

Graphs of trust

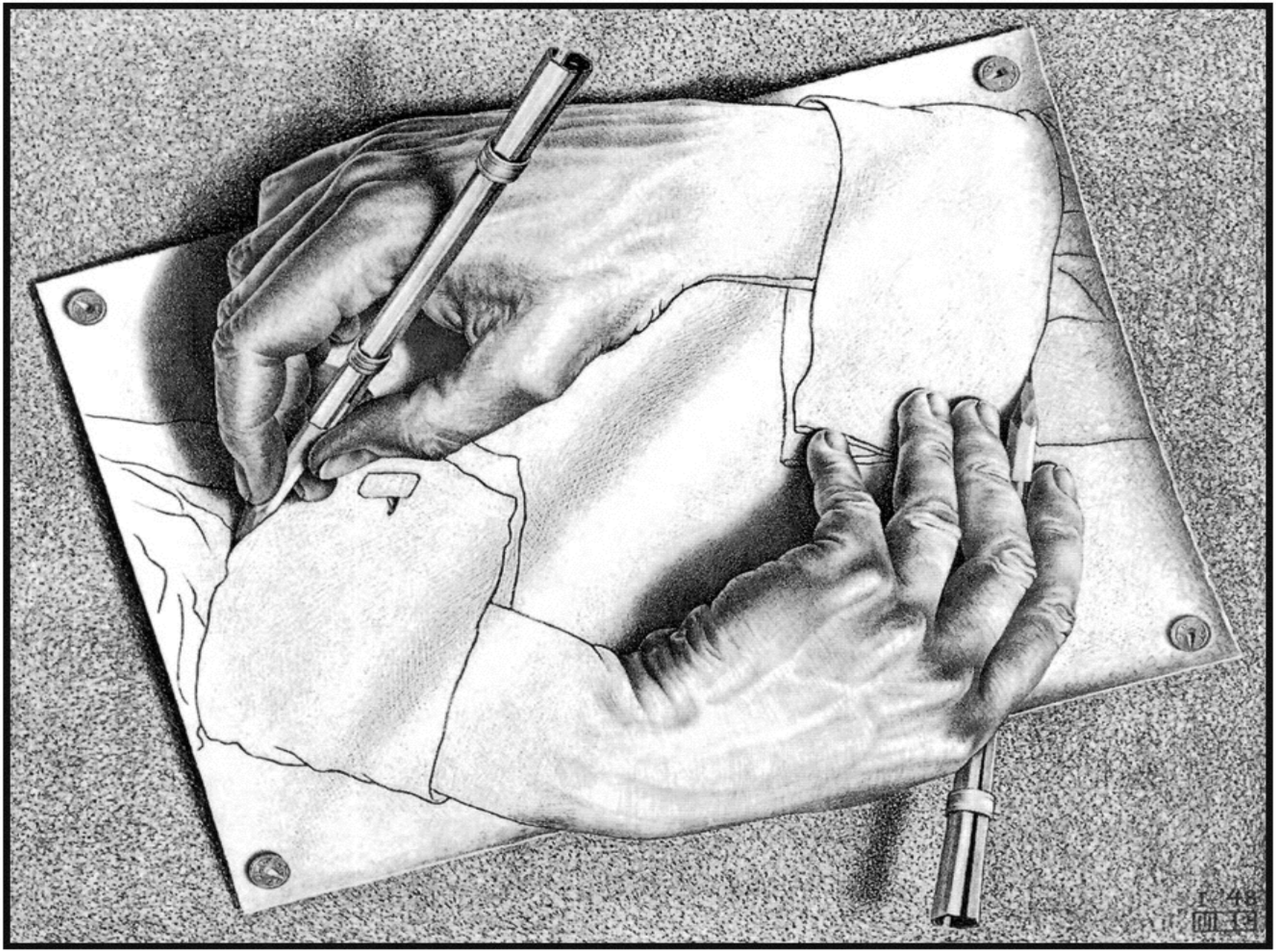
- A lot of security today happens at the “perimeter”, once you’re in it’s game over. This is called “lateral movement”

- Implicit trust: we trust somebody else servers to download executables, we trust certificate authorities keys, we trust our partner servers

- This means that your threat model is in large part outside of your control

Reactive and slow

- Most security tools today work by identifying an attack somewhere else and then try to protect everyone else
- This is reactive in nature and ineffective: most attacks stay latent for a very long time
- Even with almost-real time detection, the attacker needs to beat you at the race just once

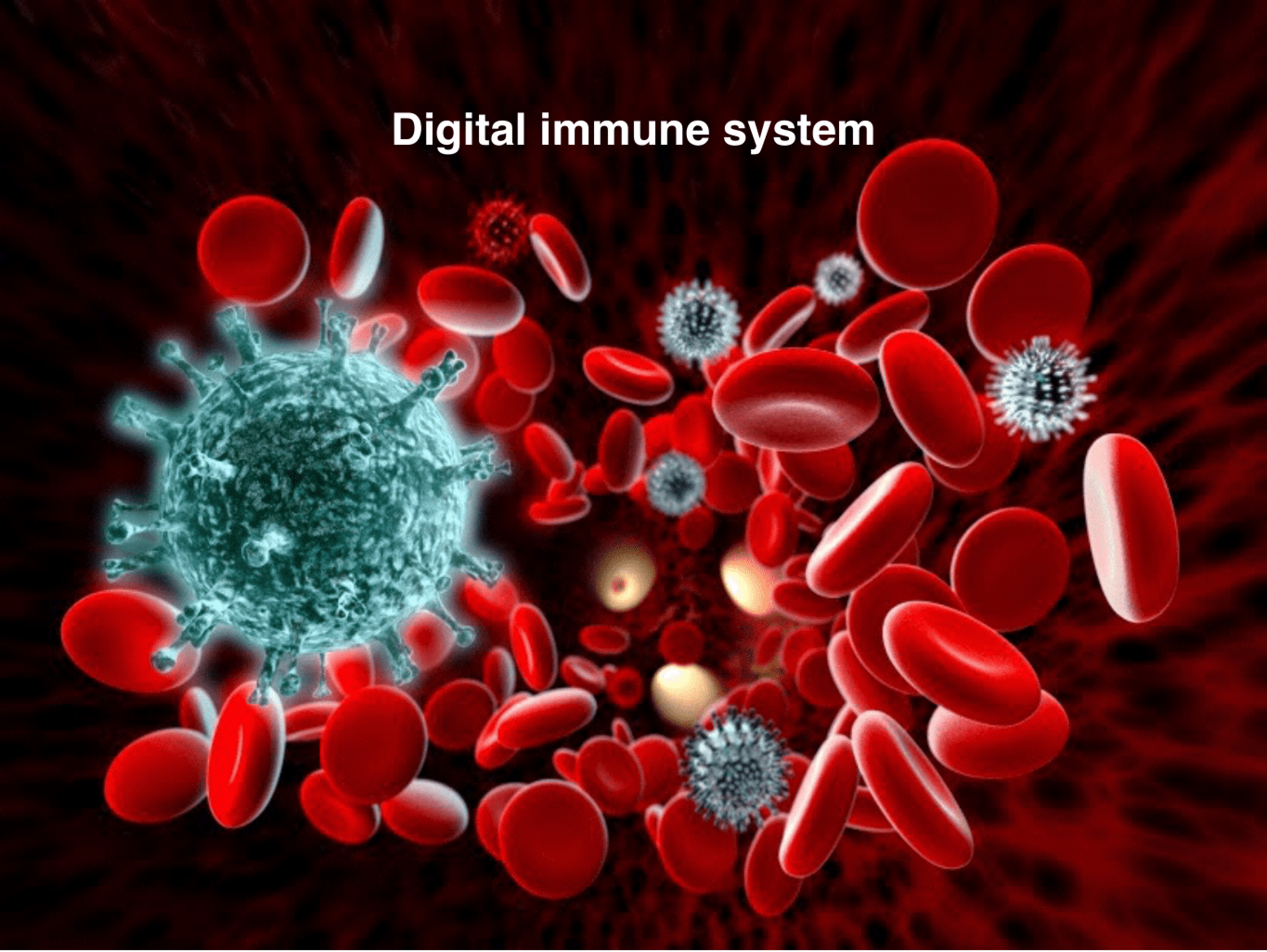

I-'48
MCE

The recursive guide to compromise anything

1. Compromise a machine (exploit, social engineering, backdoor, physical access)

2. The maxim: there's always a deep enough level in a machine that is not defended/defendable. **Go there and stay put**

3. For every node in the graph that trusts your machine, go to 1 and be fast



Digital immune system

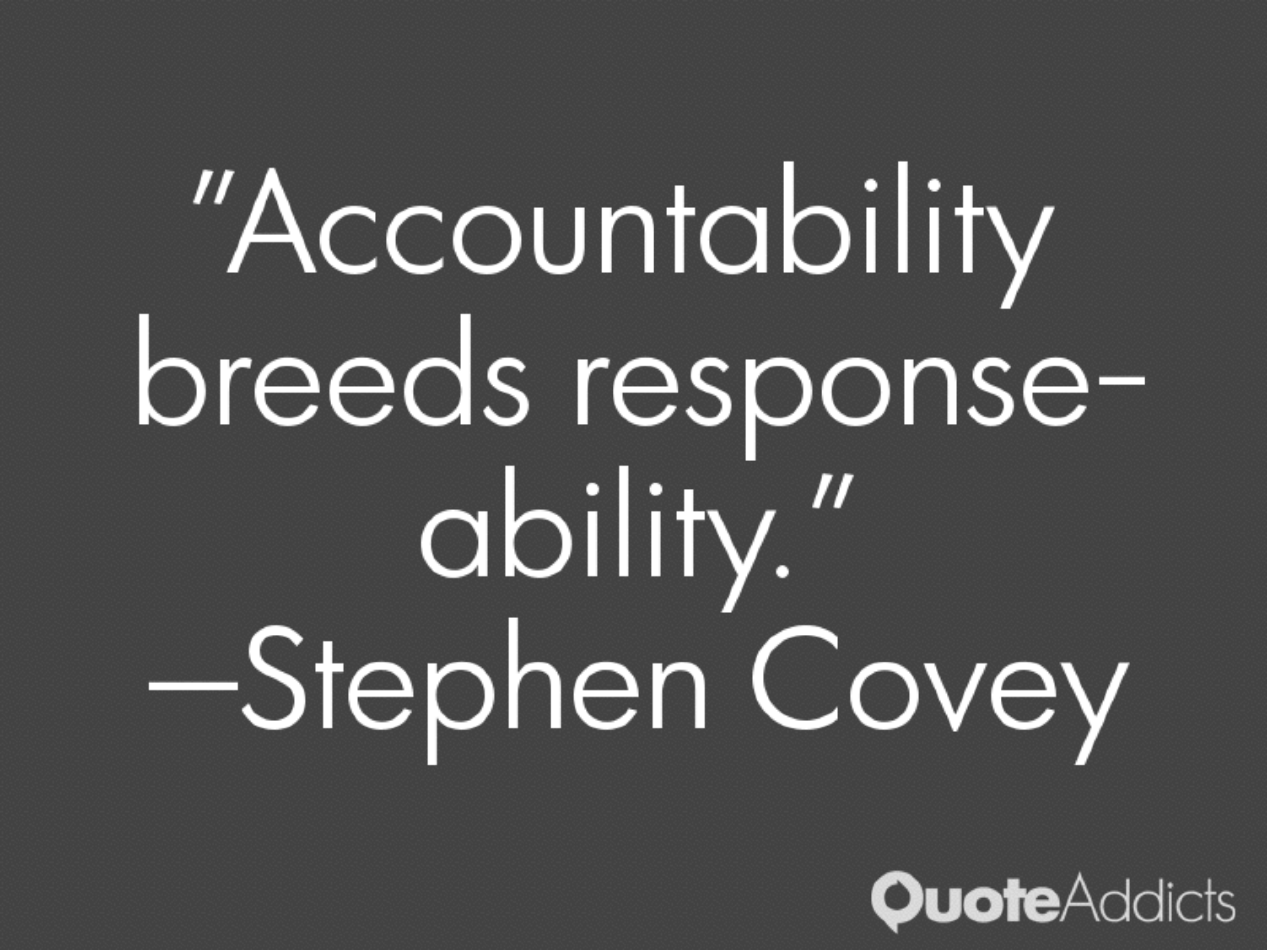
Digital immune system

- We have the technology to build 80% of the digital immune system
- We need network effects and board-level decisions to make the remaining 20% true
- This will not solve computer security but it will leap it ahead by a lot

“Shape-shifting” software

- No two copies of the same app, (kernel, firmware, etc etc) should behave the same way **at the micro level**

- Code should adapt to its users/owners, detect and log anomalous behavior on a distributed ledger



"Accountability breeds response-ability."
—Stephen Covey

Code Signing

- Every piece of code that is executed on a machine should be signed by a trusted entity
- We can’t trust a single company/machine: create a distributed ledger of valid signatures for every piece of code

1:36
SELF-DESTRUCT

Self-destructing machines

- Every machine should have a “known-good” state to revert to
- Every time a machine is thought to be compromised it should be destroyed immediately and reverted back to the “known-good” state

Adaptive network structure

- The trusting neighbors of a machine must be able to shut down communication with the allegedly compromised machine
- The trusting neighbors should be able to adapt their network topology to use a mirror copy of the compromised machine

The AI future

- In the future a lot of offensive security will be AI/ML-driven

- In the future security will be much faster and much more complicated

- We can’t have proper defense against that without these building blocks

Q&A