Cautious Optimism

Stacy Prowell
Chief Cybersecurity Research Scientist
National Security Sciences Directorate
Oak Ridge National Laboratory

February 2021

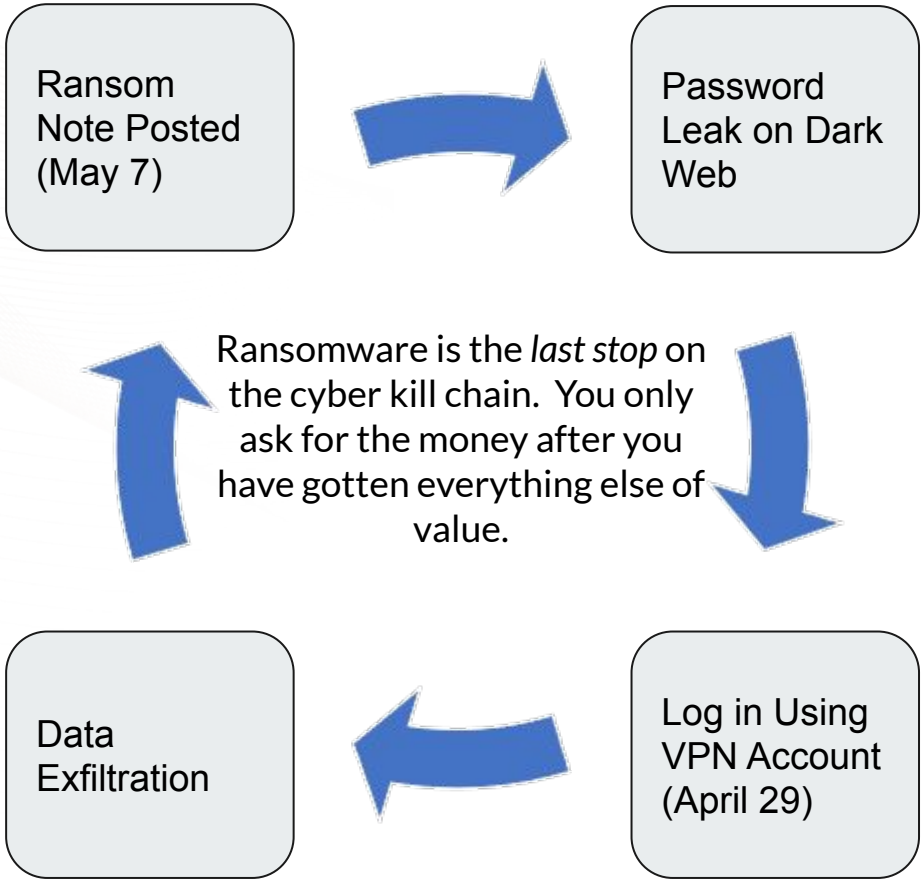
- Loss of power to 4.5 million homes and businesses
- Food and water shortages and loss of heating
- 700 deaths
- Damages estimated at $200B, the costliest disaster in Texas history.

source: https://www.flickr.com/photos/joncutrer/50975248406
Photo: Jonathan Cutrer, used under CC BY-NC 2.0

OAK RIDGE National Laboratory

Colonial Pipeline

source: https://www.eia.gov/todayinenergy/detail.php?id=47917
Image in public domain.


Ransom Note Posted (May 7)
Password Leak on Dark Web
Log in Using VPN Account (April 29)
Data Exfiltration
Ransomware is the *last stop* on the cyber kill chain. You only ask for the money after you have gotten everything else of value.

Targets

Ransomware campaigns seek targets with:

- Low tolerance for downtime
- High-value data that can be used to extort a higher ransom
- Third-party data that can be used to extort other parties or launch another campaign
- Rapid access to capital markets

Manufacturing, pipelines, etc., are the perfect targets.

from DarkReading, June 17, 2021:

One in Five Manufacturing Firms Targeted by Cyberattacks

Information-stealing malware makes up about a third of attacks, a study finds, but companies worry most about ransomware shutting down production.

Robert Lemos
Contributing Writer

source: https://www.darkreading.com/attacks-breaches/one-in-five-manufacturing-firms-targeted-by-cyberattacks

ICS Vulnerabilities?

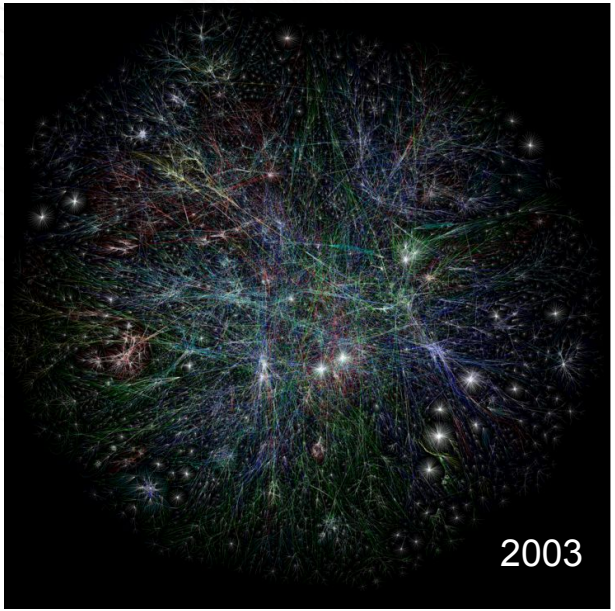
- Connected and exposed on the internet (connect everything)
- Poorly segregated (simplify the network)
- Default configurations (rapid installation, often by contractors)
- Slow password management and updates (uptime and availability are primary)
- Unknown connections
  - Contractor compromised
  - Watering hole attack

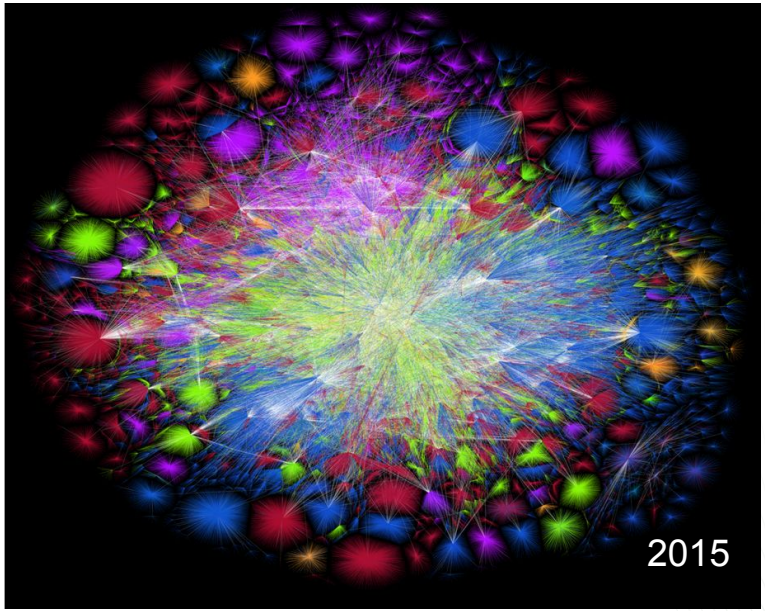
Ransomware isn't the problem.

Ransomware is a symptom of a much more serious problem.

How did we get here?

The Internet

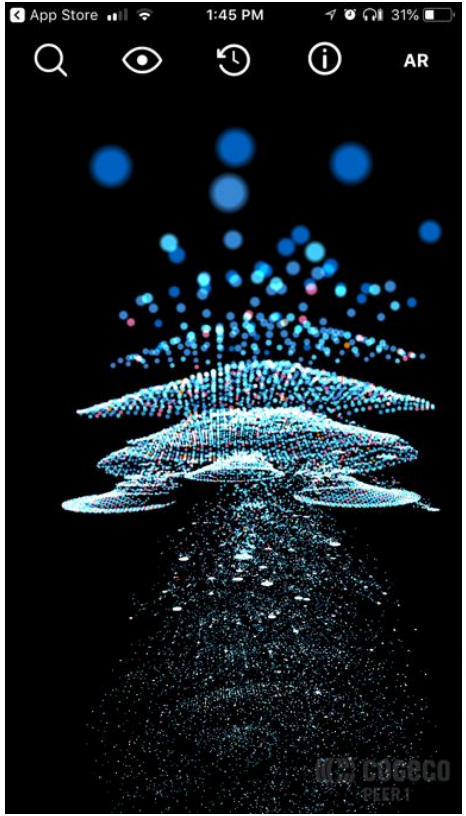




Source: The Opte Project (www.opte.org)
Images in public domain.

All about connections

- Originally **people to people** to share information
- *Then* **people to systems** to allow remote access and control
  - Meter reading / fly by wire
- *Now* **systems to systems** for autonomous control
  - Smart grid / self-driving car / demand response

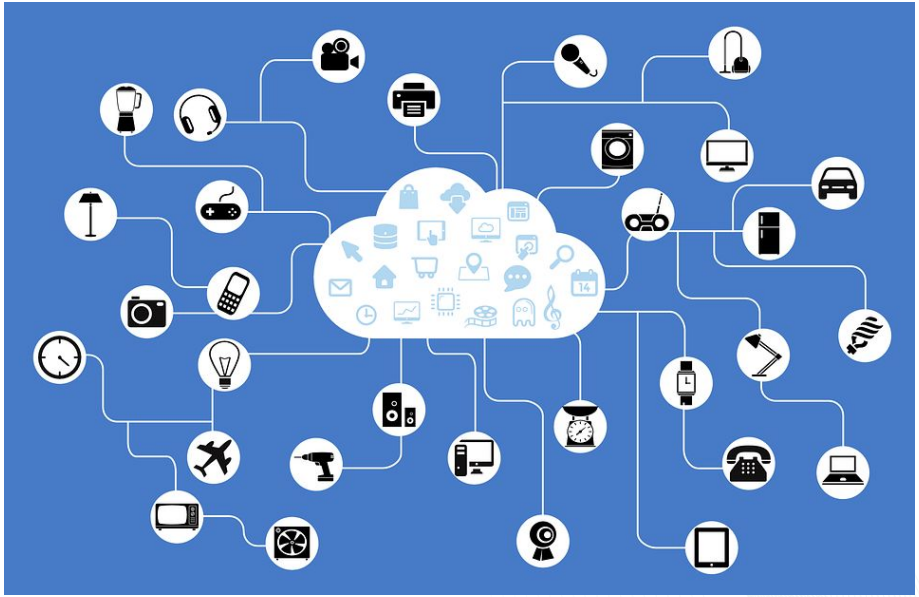


Screen capture of The Cogeco Peer 1 Map of the Internet app for iOS and Android

Open source (MIT License) app from Steamclock Software, available at: https://github.com/steamclock/internetmap

Principles of the IoT

- **Instrument all the things!**
  Unrecorded events are an opportunity to add value
- **Share all the things!**
  Sharing data enables new applications
- **Connect all the things!**
  Air gaps are a network failure



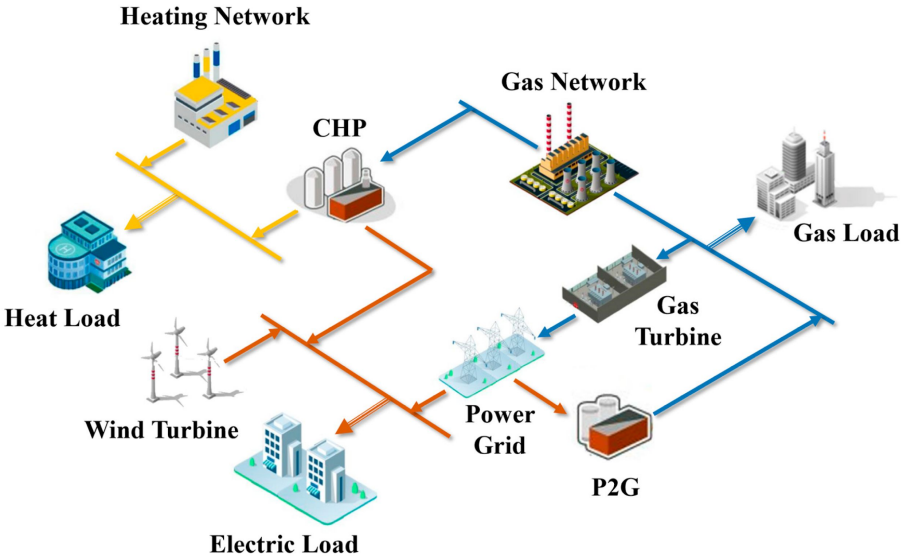
source: https://pixabay.com/vectors/network-iot-internet-of-things-782707/
Used under Pixbay License (free to use, no attribution required)

Systems Again

The electric power grid is a good example of a *convergent network*; it is built from multiple seemingly unrelated networks.

- Oil and gas
- Rail and shipping
- …

IoT Principles: instrument everything, share all data, and connect everything



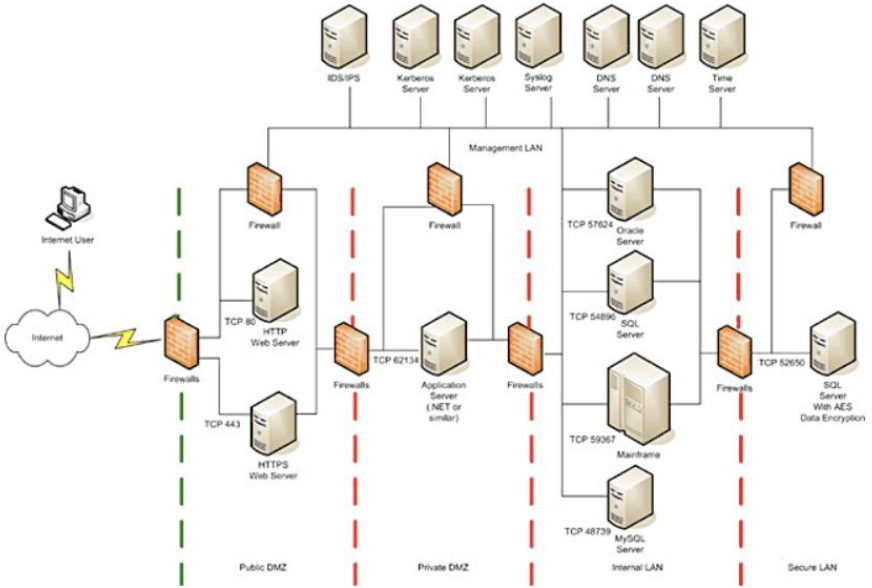
Source: Zhu, H., et al., "Distributed Optimal Scheduling of Electricity–Gas–Heating System Based on Improved Alternating Direction Method of Multipliers," Appl. Sci. 2020, 10(4), 1214; https://doi.org/10.3390/app10041214
Open access journal, educational use.

Complexity

Security systems are increasingly complex and connected, and it becomes difficult to know precisely what they are telling us.

- "False positive" problem

Information management itself becomes a security risk.



Source: https://pciguru.files.wordpress.com/2013/12/200511-theultrasecurenetworkarchitecture.pdf
Image from "The 'Ultra-Secure' Network Architecture" by PCI Guru. Used under fair use, education.

Is security hard?

Automatic Brake Demonstration

source: CNN YouTube Channel (https://youtu.be/WbW2UgmjJUA)

Volvo Cars would like to reiterate that in the depicted incident the XC60 was equipped with City Safety, however, it did not have the Pedestrian Detection functionality – this is sold as a separate package and would have been needed to mitigate or avoid the car colliding with the two men.

Even if the car was equipped with this feature, the heavy acceleration of the driver could have caused the system to be overridden and deactivated; the Volvo auto-braking (mitigation and avoidance) technology is highly advanced and in cases which the car detects that the driver intends to perform the action deliberately, it will deactivate itself.

Contrary to the title of the video, both men were alright and none of them were paralyzed nor were they a Managing Director of Volvo Cars.

Volvo Car Malaysia strongly recommends its dealers, partners and customers to never perform test towards real humans.

Public statement in public domain from Volvo Cars Malaysia

People are part of these systems

Digital devices tune out small errors while creating *opportunities for large errors*.

Every device creates its own opportunity for human error.

Exotic devices create exotic problems.

– Earl Weiner

from The Guardian, October 11, 2016:

The long read

Crash: how computers are setting us up for disaster

We increasingly let computers fly planes and carry out security checks. Driverless cars are next. But is our reliance on automation dangerously diminishing our skills?

by Tim Harford

Source: https://www.theguardian.com/technology/2016/oct/11/crash-how-computers-are-setting-us-up-disaster

Complexity

It's all lies

"Computers do exactly what you tell them, no more and no less."

It's all lies

"Computers do exactly what you tell them, no more and no less."

Computers are incredibly complex. However complex you think they are, they are more complex than that.

- **Compilers are free to re-order the work, discard it, or add to it.**

Analyzing a Program

Let's answer some questions about this program.

- How many times does it loop?

How long does it take? We'll compile it*, and then disassemble it using `objdump`.

delay.c

```
//mk: gcc -o $mkBASE $mkFILE

void delay() {
  unsigned char i, j;
  j = 0;
  while(--j) {
    i = 0;
    while(--i);
  }
}
```

* The `-c` switch to `gcc` compiles but does not link.

```
//mk: gcc -o $mkBASE $mkFILE

void delay() {
  unsigned char i, j;
  j = 0;
  while(--j) {
    i = 0;
    while(--i);
  }
}
```

gcc -c delay.c

```
delay.o:     file format elf64-x86-64


Disassembly of section .text:

0000000000000000 <delay>:
   0:   f3 0f 1e fa             endbr64
   4:   55                      push   rbp
   5:   48 89 e5                mov    rbp,rsp
   8:   c6 45 ff 00             mov    BYTE PTR [rbp-0x1],0x0
   c:   eb 0e                   jmp    1c <delay+0x1c>
   e:   c6 45 fe 00             mov    BYTE PTR [rbp-0x2],0x0
  12:   80 6d fe 01             sub    BYTE PTR [rbp-0x2],0x1
  16:   80 7d fe 00             cmp    BYTE PTR [rbp-0x2],0x0
  1a:   75 f6                   jne    12 <delay+0x12>
  1c:   80 6d ff 01             sub    BYTE PTR [rbp-0x1],0x1
  20:   80 7d ff 00             cmp    BYTE PTR [rbp-0x1],0x0
  24:   75 e8                   jne    e <delay+0xe>
  26:   90                      nop
  27:   90                      nop
  28:   5d                      pop    rbp
  29:   c3                      ret
```

```c
//mk: gcc -o $mkBASE $mkFILE

void delay() {
  unsigned char i, j;
  j = 0;
  while(--j) {
    i = 0;
    while(--i);
  }
}
```


```
delay.o:     file format elf64-x86-64


Disassembly of section .text:

0000000000000000 <delay>:
   0:   f3 0f 1e fa             endbr64
   4:   ba ff ff ff ff          mov    edx,0xffffffff
   9:   b8 00 00 00 00          mov    eax,0x0
   e:   2c 01                   sub    al,0x1
  10:   75 fc                   jne    e <delay+0xe>
  12:   80 ea 01                sub    dl,0x1
  15:   75 f2                   jne    9 <delay+0x9>
  17:   c3                      ret
```

```c
//mk: gcc -o $mkBASE $mkFILE

void delay() {
  unsigned char i, j;
  j = 0;
  while(--j) {
    i = 0;
    while(--i);
  }
}
```

gcc -c -O2 delay.c

```
delay.o:     file format elf64-x86-64


Disassembly of section .text:

0000000000000000 <delay>:
   0:   f3 0f 1e fa        endbr64
   4:   c3                 ret
```

It's all lies

"Computers do exactly what you tell them, no more and no less."

Computers are incredibly complex. However complex you think they are, they are more complex than that.

- Compilers are free to re-order the work, discard it, or add to it.
- **Even modern *assemblers* optimize your code: `mov rax, 0x21` → `mov eax, 0x21`**

It's all lies

"Computers do exactly what you tell them, no more and no less."

Computers are incredibly complex. However complex you think they are, they are more complex than that.

- Compilers are free to re-order the work, discard it, or add to it.
- Even modern *assemblers* optimize your code: `mov rax, 0x21` → `mov eax, 0x21`
- ***The processor* is free to re-order the work, discard it, or add to it.**

The processor can do work *out of order*, can do *different* (but equivalent) work, can do *extra* work, or can *not do* requested work if it determines it is not needed.

As long as the overall effect is the same, the processor is free to do *anything*.

This is important.

Whenever the processor changes the work schedule, you have the potential for a side-channel vulnerability. Modern processors *always* change the work schedule.

Store Buffering

Intermediate states are *everywhere*.

Core 1	Core 2
`mov qword [foo], 1` `mov rax, [bar]`	`mov qword [bar], 1` `mov rbx, [foo]`

Because of store buffering it is possible that neither write happens for a while, and the intermediate results *both* read old values.

It matters!

Modern processors, especially those with varying instruction lengths, are always trying to execute *faster*.

If there is something slow that must be done–a memory access, for instance–fill that time using *speculative execution*. If the result is never needed, throw it away. No harm and no foul.

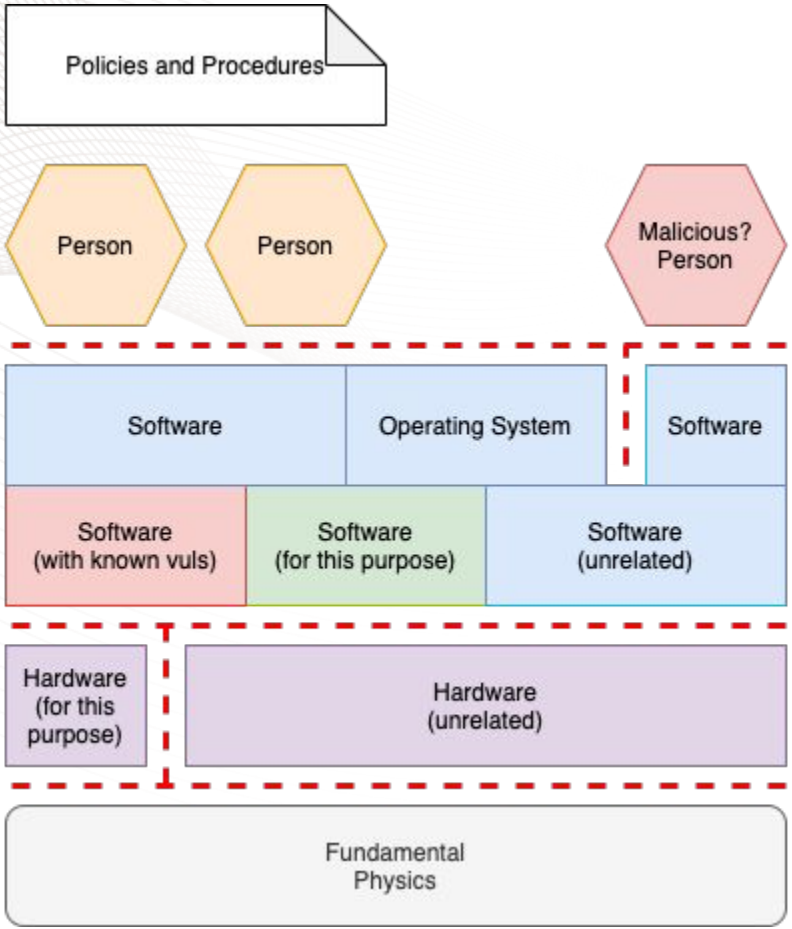
Except… Spectre. See: https://spectrum.ieee.org/how-the-spectre-and-meltdown-hacks-really-worked

The work done still has detectable effects.

Security is hard

Security is different

- Attacks are **systematic**, not random
- Security is **invisible** and hard to measure
- Security is highly **contextual**
- Security is a property of **systems**
- **Composition** is unsolved
- Security is often **binary**
- Security is an **assumed property**

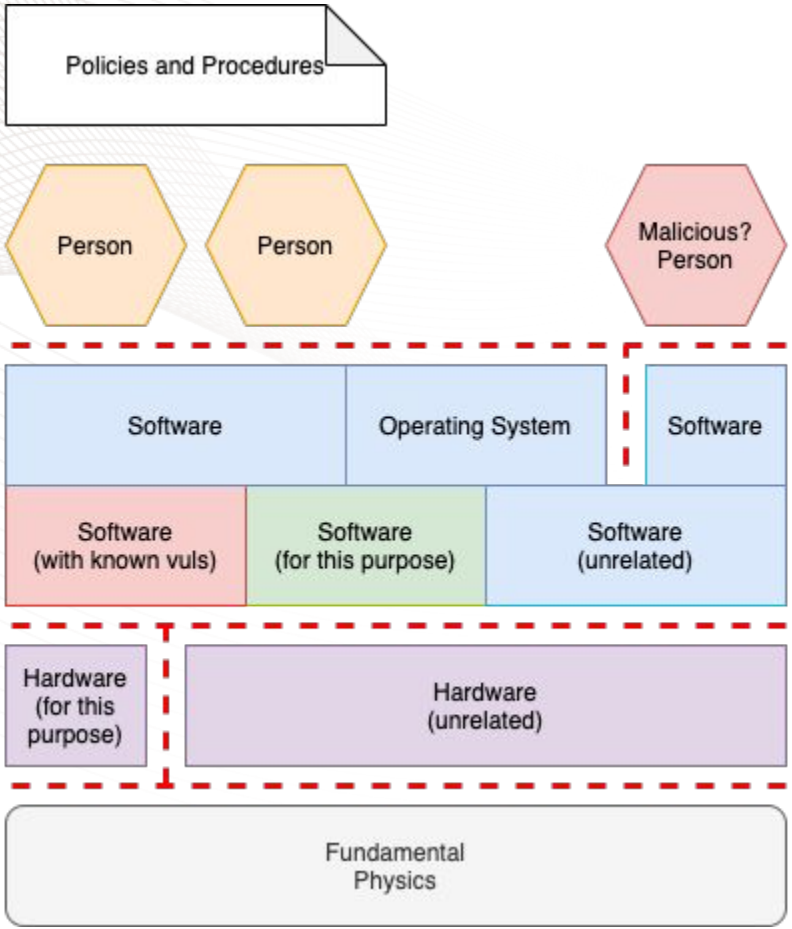


All Systems are Systems of Systems

We *do not know* how to compose secure components to yield secure systems.

An adversary will exploit vulnerability wherever it is found.

Once you have solved a problem, then next thing to do is **compose it with solutions to unrelated problems that share some of the same resources**.



Vulnerabilities at the Edges

Vulnerabilities typically occur where two systems touch (interfaces, protocols).

ICS systems have multiple boundaries with different security postures.

Adding a Feature to a Secure System…

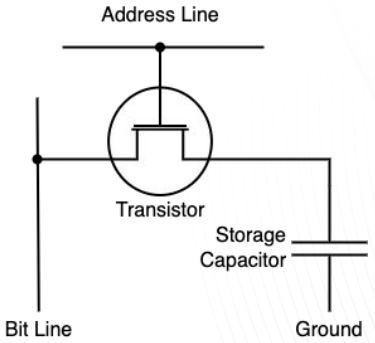
source: DefCon YouTube Channel (https://youtu.be/jmTwlEh8L7g)

Complexity

DRAM

Can we exploit *random* memory errors?

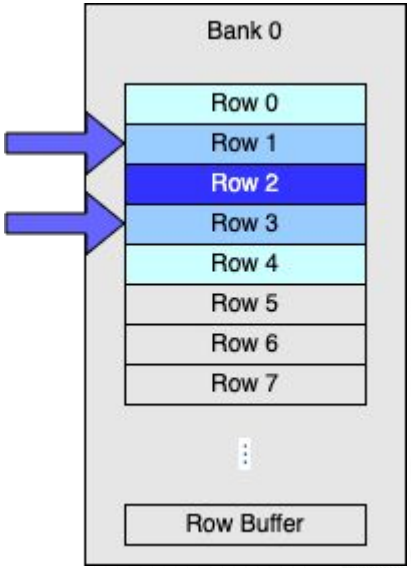
Dynamic RAM (DRAM) Cell Structure



Rowhammer

- Repeatedly write to a row…
- Cause a bit error in an adjacent row
- Improvement! Alternate between two rows

- Can’t someone else do it? Yes. Code online!




```
top:
      mov eax, [foo]     ; read from address foo
      mov ebx, [bar]     ; read from address bar
      clflush [foo]      ; flush cache for address foo
      clflush [bar]      ; flush cache for address bar
      jmp top
```

Rowhammer

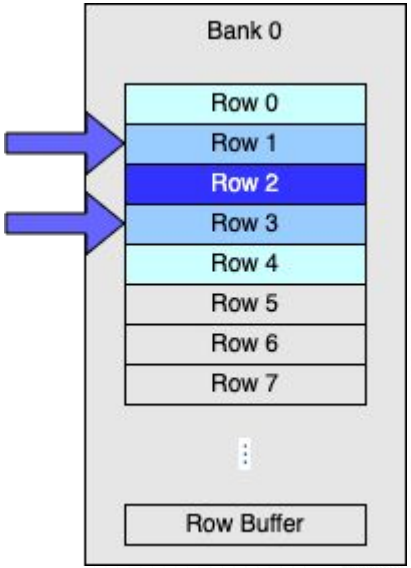
from ArsTechnica, October 23, 2016:

RISK ASSESSMENT—

Using Rowhammer bitflips to root Android phones is now a thing

Permission-less apps take only seconds to root phones from LG, Samsung and Motorola.

Dan Goodin

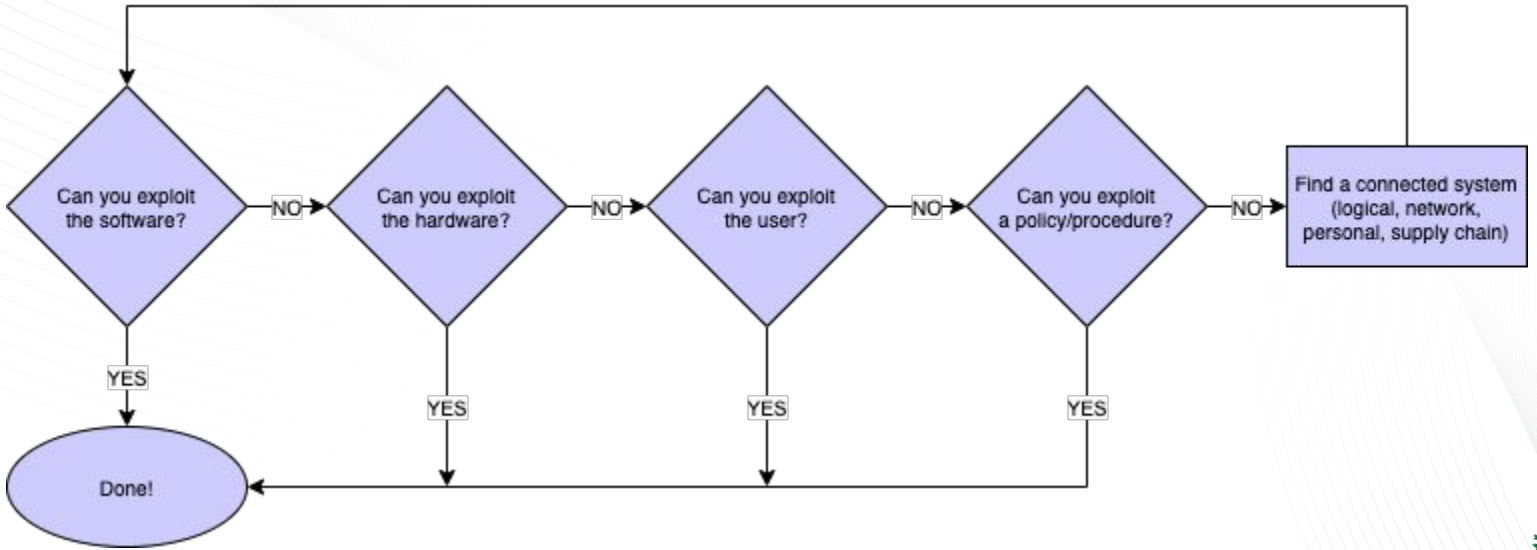


source:
https://arstechnica.com/information-technology/2016/10/using-rowhammer-bitflips-to-root-android-phones-is-now-a-thing/

RAMBleed

When you have time: https://rambleed.com/

It's all systems, all the way down...



Cautious Optimism

Physics and Side-Channels

Software running on a computer can lie to you and it is difficult to know.

Rootkits modify the operating system to subvert detection. They can even modify firmware to hide. What if malware modifies the firmware of your network card?

What if you can't trust your network card?

Loïc Duflot, Yves-Alexis Perez, and Benjamin Morin

ANSSI
French Network and Information Security Agency
51 boulevard de la Tour Maubourg, 75007 Paris

Abstract: In the last few years, many different attacks against computing platform targeting hardware or low level firmware have been published. Such attacks are generally quite hard to detect and to defend against as they target components that are out of the scope of the operating system and may not have been taken into account in the security policy enforced on the platform. In this paper, we study the case of remote attacks against network adapters. In our case study, we assume that the target adapter is running a flawed firmware that an attacker may subvert remotely by sending packets on the network to the adapter. We study possible detection techniques and their efficiency. We show that, depending on the architecture of the adapter and the interface provided by the NIC to the host operating system, building an efficient detection framework is possible. We explain the choices we made when designing such a framework that we called NAVIS and give details on our proof of concept implementation.

Keywords: firmware, NIC, network adapter, runtime verification

source: Proceedings Recent Advances in Intrusion Detection - 14th International Symposium, RAID 2011, Menlo Park, CA, USA, September 20-21, 2011. 10.1007/978-3-642-23644-0_20.

Physics and Side-Channels

It is hard to fake the *physical effects* of an operation.

For the power grid, for example, the effects of local weather can be seen in data from substations.

Leveraging EM Side-Channel Information to Detect Rowhammer Attacks

Zhenkai Zhang, Texas Tech University, **Daniel Balasubramanian**, Vanderbilt University, **Zihao Zhan**, Vanderbilt University, **Bo Li**, UIUC, **Peter Volgyesi**, Vanderbilt University, **Xenofon Koutsoukos**, Vanderbilt University

Abstract—The rowhammer bug belongs to software-induced hardware faults, and has been exploited to form a wide range of powerful rowhammer attacks. Yet, how to effectively detect such attacks remains a challenging problem. In this paper, we propose a novel approach named RADAR (Rowhammer Attack Detection via a Radio) that leverages certain electromagnetic (EM) signals to detect rowhammer attacks. In particular, we have found that there are recognizable hammering-correlated sideband patterns in the spectrum of the DRAM clock signal. As such patterns are inevitable physical side effects of hammering the DRAM, they can "expose" any potential rowhammer attacks including the extremely elusive ones hidden inside encrypted and isolated environments like Intel SGX enclaves. However, the patterns of interest may become unapparent due to the common use of spread-spectrum clocking (SSC) in computer systems. We propose a de-spreading method that can reassemble the hammering-correlated sideband patterns scattered by SSC. Using a common classification technique, we can achieve both effective and robust detection-based defense against rowhammer attacks, as evaluated on a RADAR prototype under various scenarios. In addition, our RADAR does not impose any performance overhead on the protected system. There has been little prior work that uses physical side-channel information to perform rowhammer defenses, and to the best of our knowledge, this is the first investigation on leveraging EM side-channel information for this purpose.

source: Proceedings 2020 IEEE Symposium on Security and Privacy (SP). 729-746. 10.1109/SP40000.2020.00060.

ICS Systems Have a Lot of Sensors

Correlation among sensor readings can reveal tampering.

The analog properties of digital signals can be used to "fingerprint" the signal.

Open Access Article

Use of Thermistor Temperature Sensors for Cyber-Physical System Security

by Carson Labrado [1], Himanshu Thapliyal [1,*], Stacy Prowell [2] and Teja Kuruganti [3]

[1] Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY 40506, USA
[2] Cyber and Applied Data Analytics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[3] Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
* Author to whom correspondence should be addressed.

Sensors **2019**, *19*(18), 3905; https://doi.org/10.3390/s19183905

Abstract

The last few decades have seen a large proliferation in the prevalence of cyber-physical systems. This has been especially highlighted by the explosive growth in the number of Internet of Things (IoT) devices. Unfortunately, the increasing prevalence of these devices has begun to draw the attention of malicious entities which exploit them for their own gain. What makes these devices especially attractive is the various resource constraints present in these devices that make it difficult to add standard security features. Therefore, one intriguing research direction is creating security solutions out of already present components such as sensors. Physically Unclonable Functions (PUFs) are one potential solution that use intrinsic variations of the device manufacturing process for provisioning security. In this work, we propose a novel weak PUF design using thermistor temperature sensors. Our design uses the differences in resistance variation between thermistors in response to temperature change. To generate a PUF that is reliable across a range of temperatures, we use a response-generation algorithm that helps mitigate the effects of temperature variation on the thermistors. We tested the performance of our proposed design across a range of environmental operating conditions. From this we were able to evaluate the reliability of the proposed PUF with respect to variations in temperature and humidity. We also evaluated the PUF's uniqueness using Monte Carlo simulations. View Full-Text

Keywords: Physically Unclonable Function (PUF); sensor PUF; cyber-physical systems; Internet of Things (IoT); thermistor; security

source: https://www.mdpi.com/1424-8220/19/18/3905

It is harder to lie about the weather to someone who has a window.

What's needed?

What do we need?

- Data
  - Good, synchronized multimodal data collected under experimental control
  - Threat intelligence on ICS systems
- Testbeds
  - Testbeds with replay and good resolution to lower the cost of testing attacks and solutions
- Side channel methods
  - Fingerprinting of devices for detection and attestation
- Appropriate networking
  - SND/NDN
- Better "blocking and tackling"
  - Wider deployment of known good solutions, and tailoring of those solutions to the problems specific to ICS

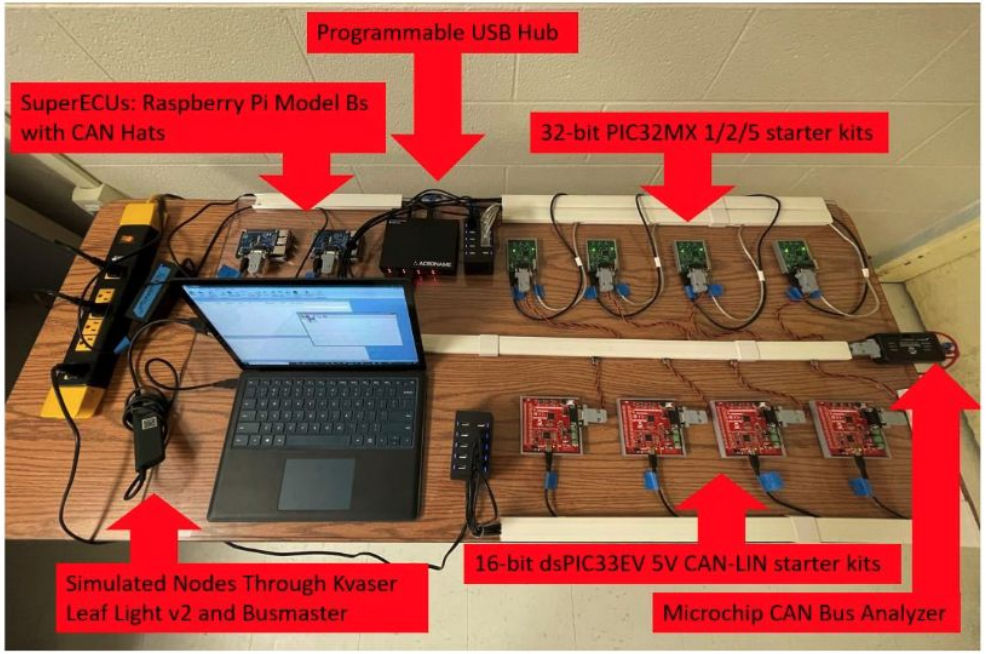


Fig. 2. Hybrid CAN Testbed

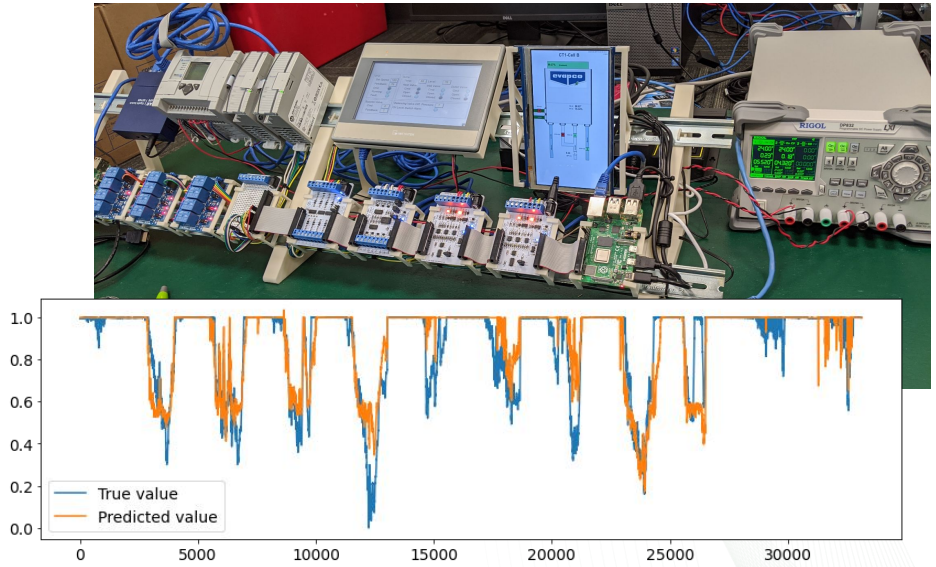
source: William Lambert, "Development of a Hybrid CAN Testbed for In-Vehicle Security Reserach," Tennessee Tech
Used with permission.

Digital Twins

Create a **digital twin** of the process and run the twin and the process in parallel, letting the twin predict the output of the actual system.

Significant deviation can be detected and investigated.

Can be inexpensive to re-create for education and training… but can be expensive to create initially.



source: Acronym Project, ORNL

Perfect security is impossible

Perfect security is impossible
Better security is possible
Resiliency is possible

Thanks!

Contact:
Stacy Prowell (prowellsj@ornl.gov and sprowell@tntech.edu)

Additional Images

Uncredited diagrams in this presentation were created by the author, Stacy Prowell.

Motivation

Defender: We want to secure the power grid to assure that it *continues to deliver power*; that it remains available and useful.

Resiliency

The ability to overcome unexpected challenges

If perfect security is impossible, then you cannot count on keeping all bad actors out (though you should certainly try). The focus should be on *resiliency*.

With people we focus on *building skills* over just addressing deficits...

For the grid, we likewise need to focus on *building alternative capabilities* for recovery and not just fixing vulnerabilities.