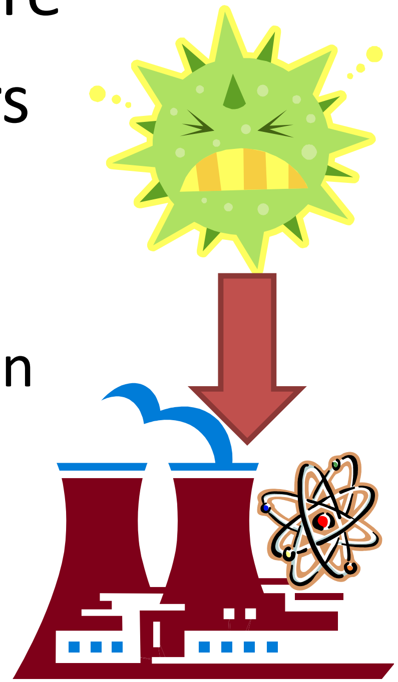
Examples of Recent Malware

Stuxnet
Equation Group
Row Hammer
Spectre (v1)

Magnus Almgren

New Era 2010: Stuxnet

- Advanced Malware
  - target specifically Programmable Logic Controllers: Siemens SIMATIC Step 7 software
  - Lots of rumors of goal and who creators
    - designed and released by a government
      - the U.S. or Israel ???
    - **Target**: Bushehr nuclear power plant in Iran (60% of infected hosts in Iran)

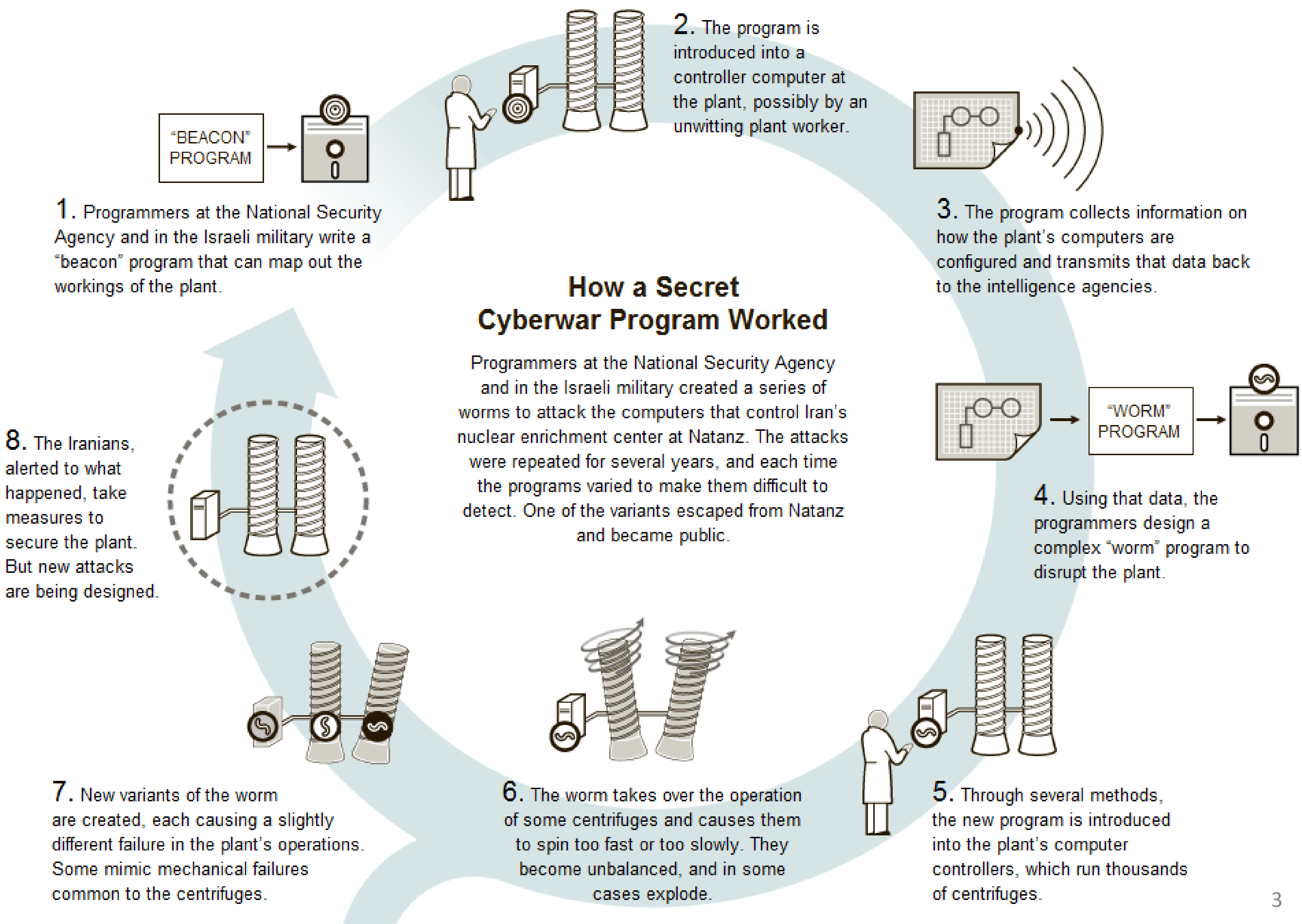


Symantec oct-2010: W32.Stuxnet Dossier (http://goo.gl/pP7S)

Published: June 1, 2012

RECOMMEND TWITTER LINKEDIN E-MAIL SHARE

Related Article »



http://www.nytimes.com/interactive/2012/06/01/world/middleeast/how-a-secret-cyberwar-program-worked.html

Stuxnet: Pandora's box ?

– Stuxnet is advanced and one of the first wild malware's targeting PLCs.
  - 6—8 people about 6 months to create.

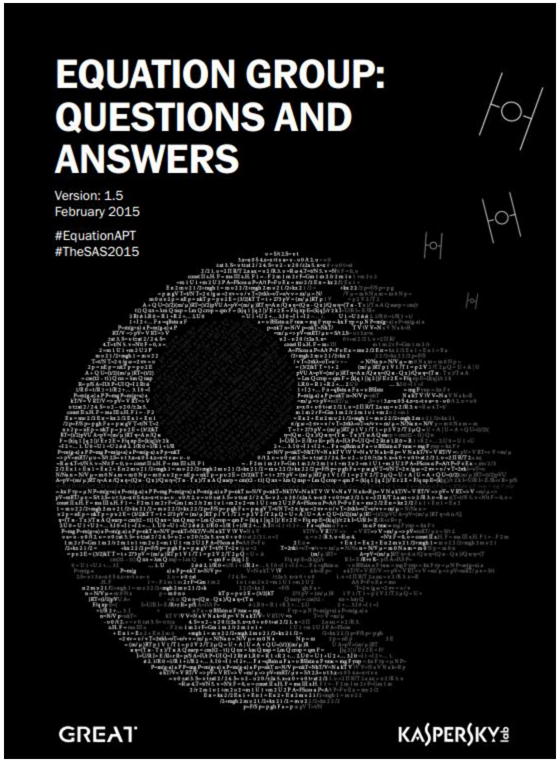
– PLCs exists in many industries
  - factory assembly lines, amusement rides, or lighting fixtures.

now blueprint to create malware targeting PLCs

- Compare this with the *Loveletter* virus (2000)
  – 2003/11 there existed 82 different variants of Loveletter.
  – It is claimed that more than 5,000 attacks are carried out every day.

> The #EquationAPT group is probably one of the most sophisticated cyber attack groups in the world #TheSAS2015

Tweet

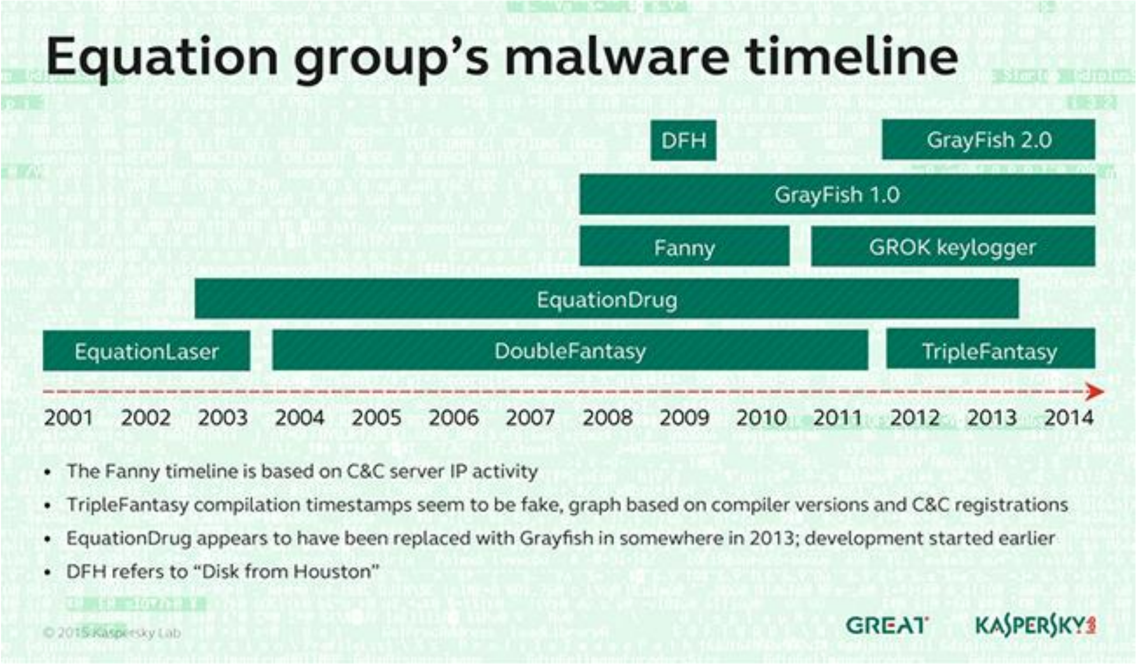


- Set of Malware Modules:
  - EQUATIONLASER,
  - EQUATIONDRUG,
  - DOUBLEFANTASY,
  - TRIPLEFANTASY,
  - FANNY, and
  - GRAYFISH

https://securelist.com/blog/research/68750/equation-the-death-star-of-malware-galaxy/
https://securelist.com/files/2015/02/Equation_group_questions_and_answers.pdf

The Equation Group

- Kaspersky Feb 2015
  - a threat actor that surpasses anything known in terms of complexity and sophistication of techniques
  - Active for almost two decades (!)
- Modular design



The Equation Group (cont'd)

- "Infect" (reprogram hard drive firmware)
  - First known malware doing this
  - Allows persistence – even if computer is reinstalled and disk reformatted, it can hide and reoccur.
  - No methods to read and analyze firmware …
  - Hidden areas ➔ store any passwords entered into the computer
- Handle "air-gapped" systems: Fanny worm
  - USB stick control
- Infection
  - online activities
  - Intercepting physical goods, and infecting (replacing them)
- Advanced functionality
  - Mapping out new systems, if "good" → download new code
  - Update modules

http://www.kaspersky.com/about/news/virus/2015/equation-group-the-crown-creator-of-cyber-espionage

The Equation Group (cont'd)

- "Infect" (reprogram hard drive firmware)
  - First known malware doing this
  - Allows persistence – even if computer is reinstalled and disk reformatted, it can hide and reoccur.
  - No methods to read and analyze firmware …
  - Hidden areas ➔ store any passwords entered into the computer
- Handle "air-gapped" systems: Fanny worm
  - USB stick control
- Infection
  - online activities
  - Intercepting physical goods, and infecting (replacing them)
- Advanced functionality
  - Mapping out new systems, if "good" → download new code
  - Update modules

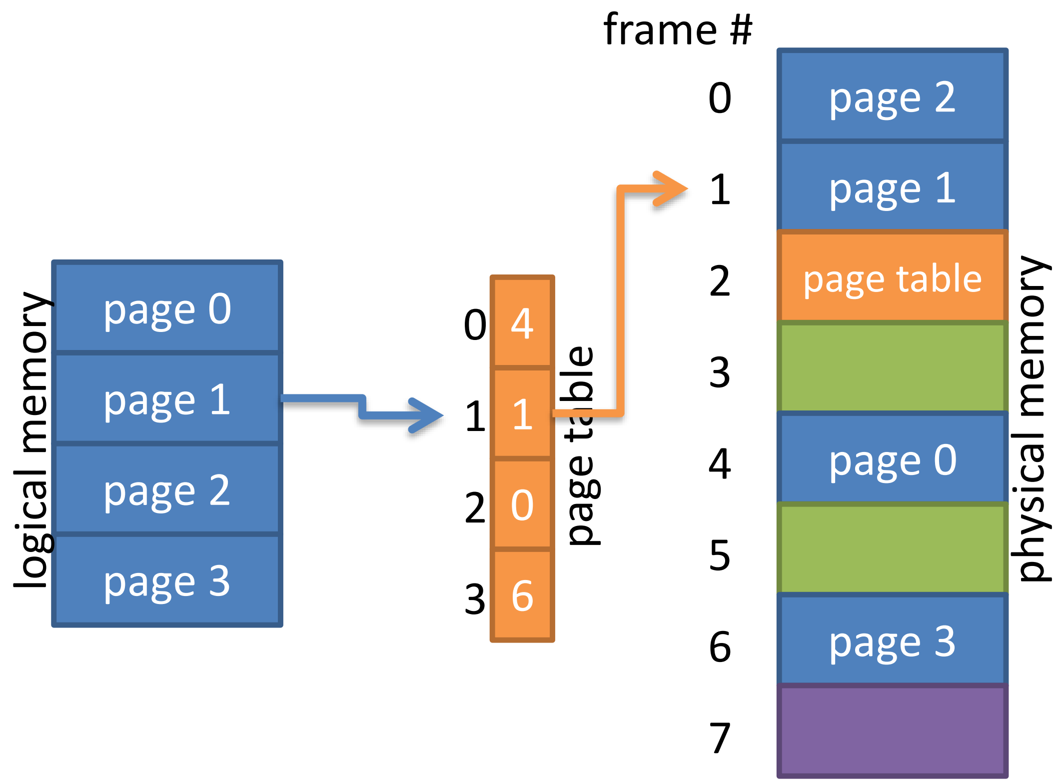
http://www.kaspersky.com/about/news/virus/2015/equation-group-the-crown-creator-of-cyber-espionage

EXPLOITING DRAM: ROWHAMMER BUG

Lecture for EDA 263
Adapted from post by Mark Seaborn, Google
http://googleprojectzero.blogspot.se/2015/03/exploiting-dram-rowhammer-bug-to-gain.html

Magnus Almgren
Department of Computer Science and Engineering
Chalmers University of Technology

Operating System Primer



- Operating System Defenses: Separation (in L09)
  - Physical, Temporal, Logical, Cryptographic

Modern web browser: sandboxed

- Sandboxing
  - Sandboxing helps prevent malware from installing itself on your computer or using what happens in one browser tab to affect what happens in another. The sandbox adds an additional layer of protection to your browser by protecting against malicious web pages that try to leave programmes on your computer, monitor your web activities or steal private information from your hard drive. *https://tools.google.com/dlpage/res/chrome/en-GB/more/security.html*

Modern web browser: sandboxed

- Sandboxing (cont')
  - Sandboxed processes that execute within a very restrictive environment. The only resources sandboxed processes can freely use are CPU cycles and memory. For example, sandboxes processes cannot write to disk or display their own windows. What exactly they can do is controlled by an explicit policy. *https://www.chromium.org/developers/design-documents/sandbox/Sandbox-FAQ*

- Can we create malware to break out of the sandbox?
  - Visiting a web page,
  - Running some javascript,
  - Possible to break out of sandbox and take control of the computer?

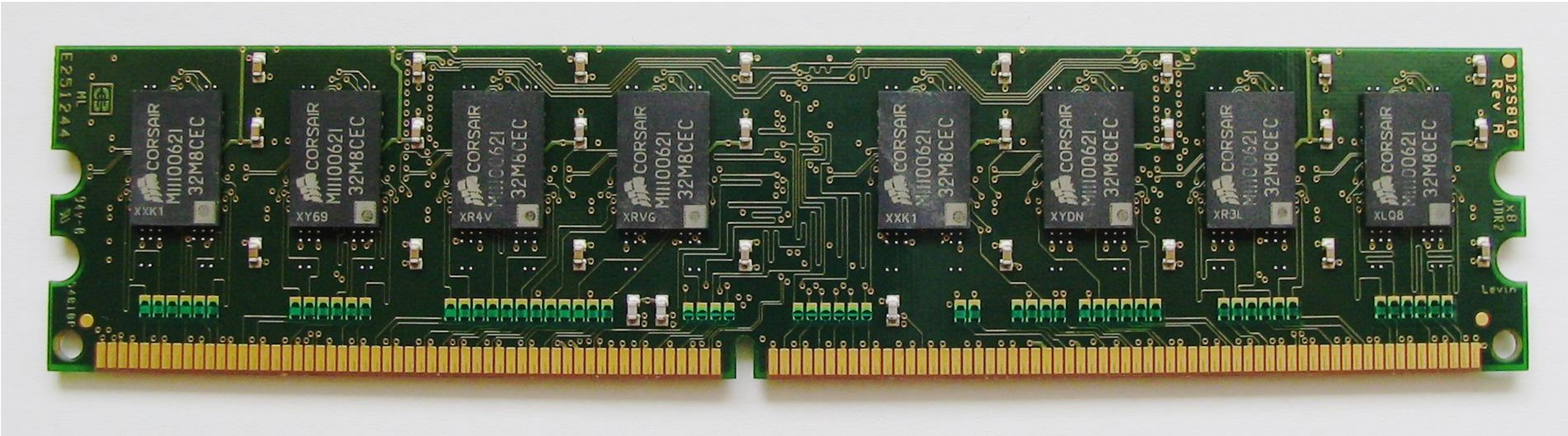
Using reliability problems in hardware to cause security issues in software

The Rowhammer exploit

> History has shown that issues that are thought to be “only” reliability issues often have significant security implications, and the rowhammer problem is a good example of this.
>
> Mark Seaborn, Google

- Assumption:
  - *Many layers of software security rest on the assumption the contents of memory locations don't change unless the locations are written to.*
- What happens if that does not hold?

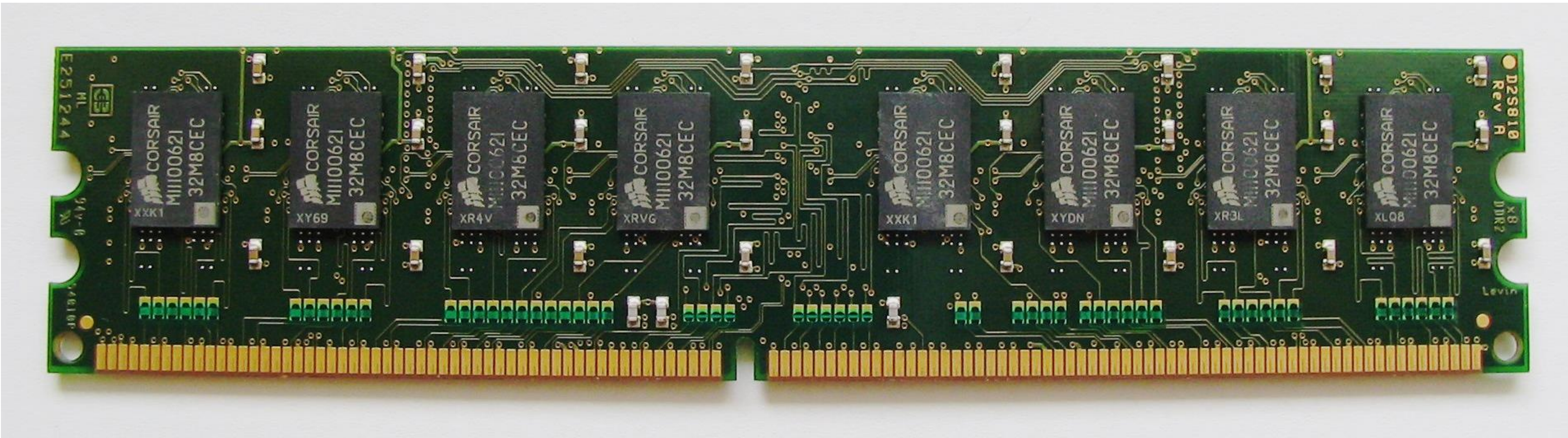
Background: Computer memory



- Dynamic random access memory (DRAM) is a type of memory that is typically used for data or program code.
- DRAM stores each bit of data or program code in a storage cell consisting of a capacitor and a transistor, and is typically organized in a rectangular configuration of storage cells.
- DRAM storage cell is dynamic in that it needs to be refreshed or given a new electronic charge every few milliseconds **(64 ms)** to compensate for charge leaks from the capacitor.

http://searchstorage.techtarget.com/definition/DRAM

Background: Computer memory



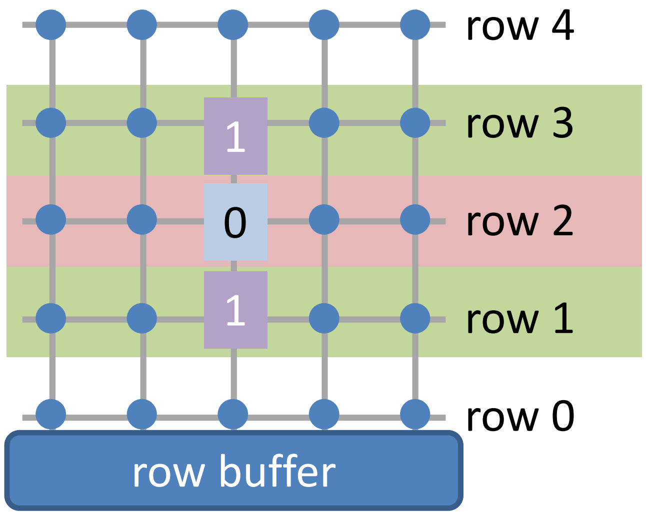
- DRAM cells getting smaller and smaller
  → the interaction between them increases.
- As a result, accessing one location in memory can disturb neighbouring locations, causing charge to leak into or out of neighbouring cells. With enough accesses, this can change a cell's value from 1 to 0 or vice versa.

http://searchstorage.techtarget.com/definition/DRAM

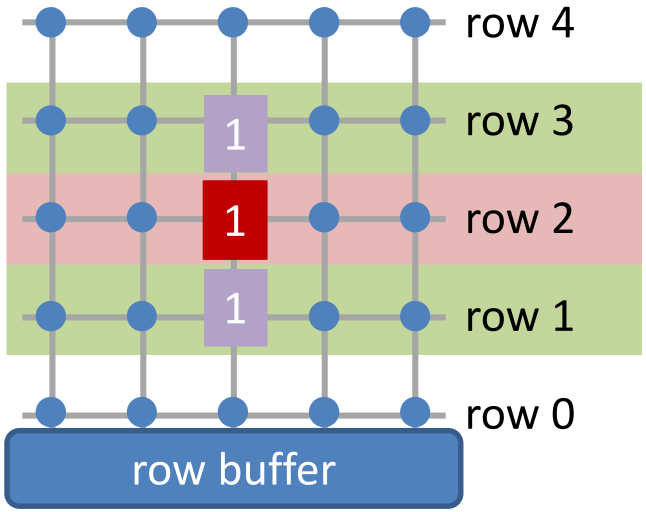
Attack

- Write to a row "*enough*" times will disturb a neighboring row
  - → Need to write "*enough*" times between the natural refresh rate.
- Problems:
  - Each DRAM bank has a "current activated row"
    - write to different "rows" within same bank
      → rowhammering
      (best if these adjacent to the one to change)
  - Avoid cache (for example to force a flush)

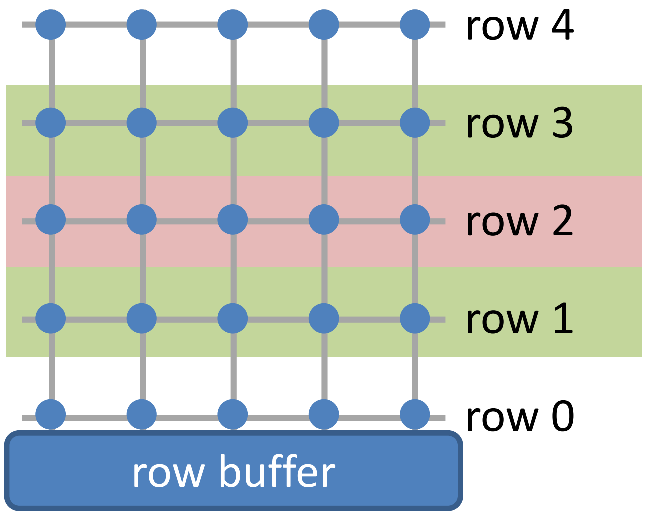
Attack in hardware



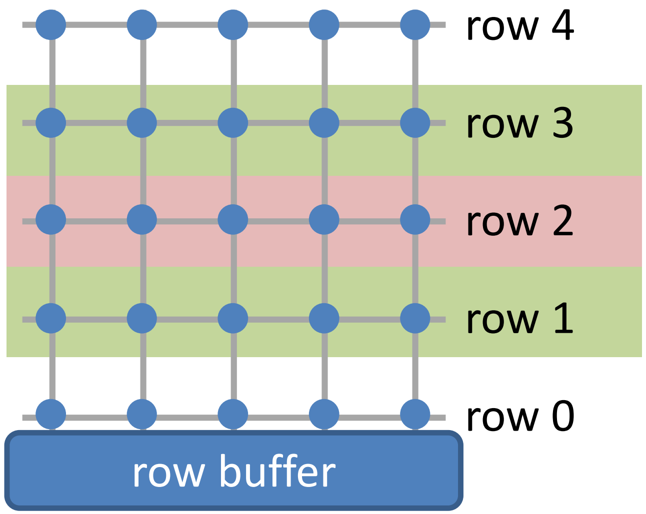
Attack in hardware

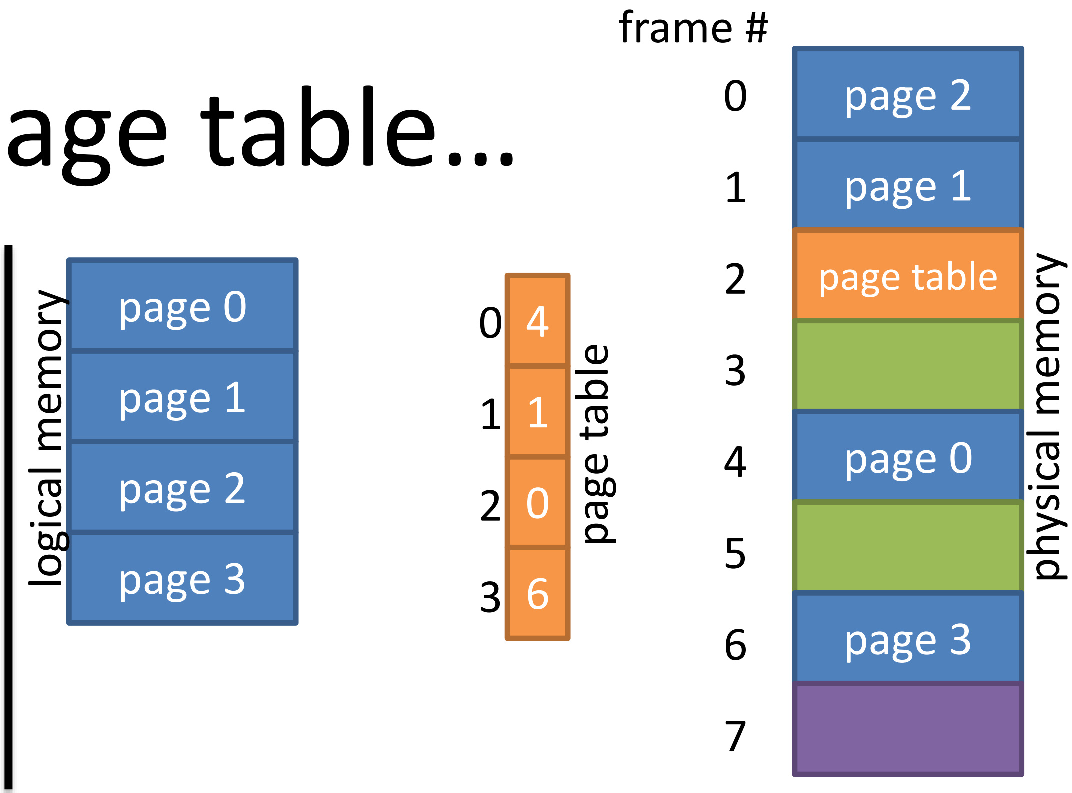


Exploit?

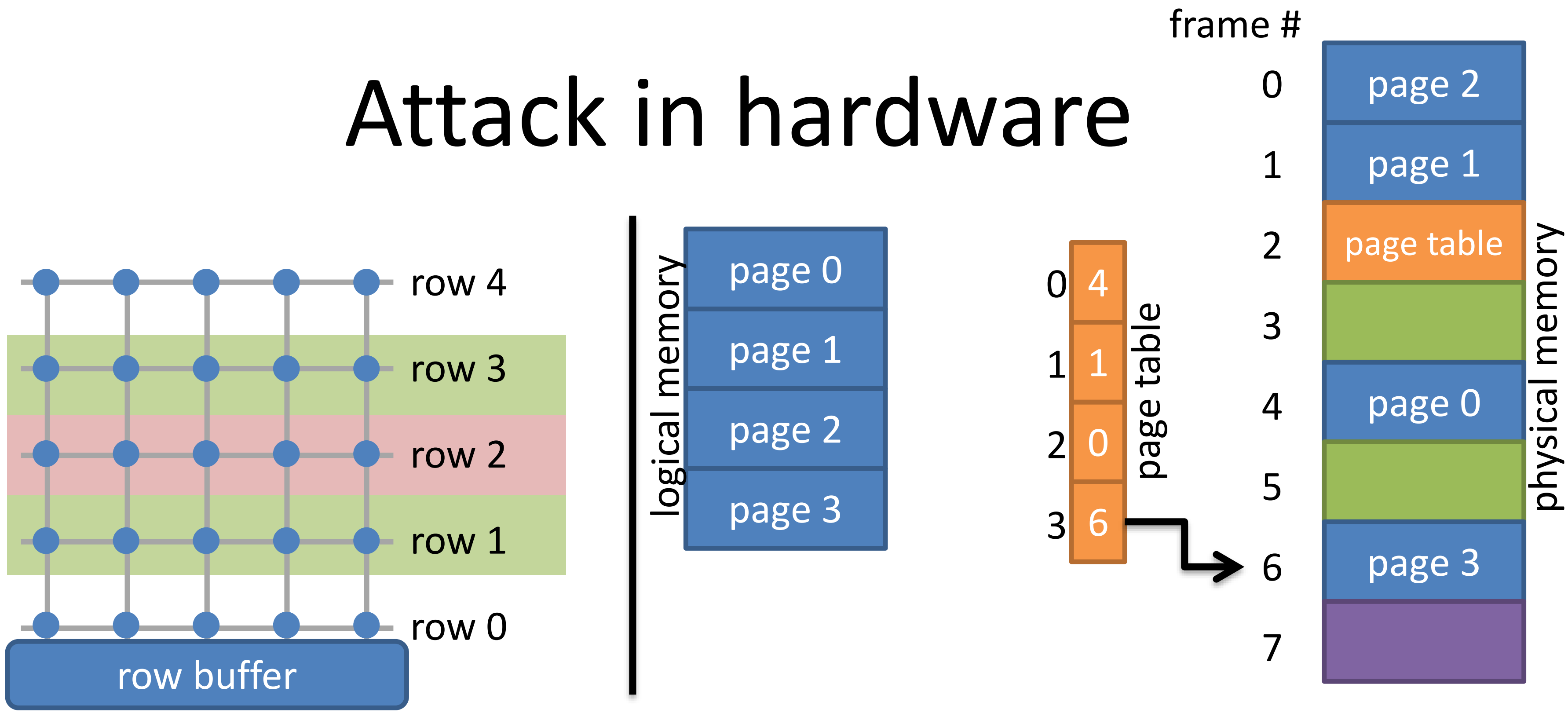


The page table…



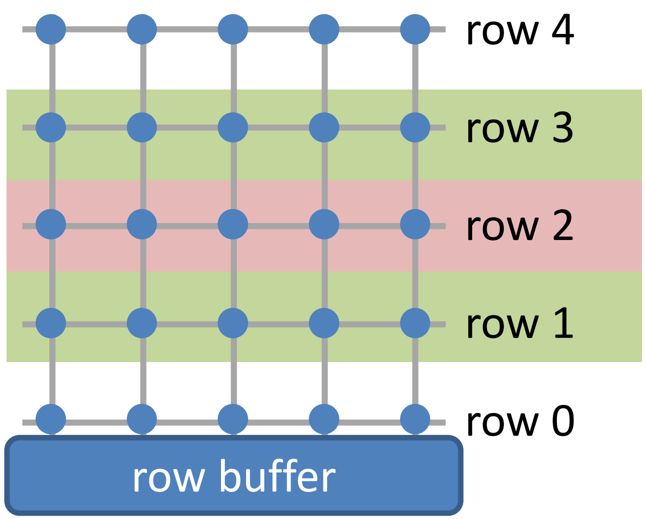


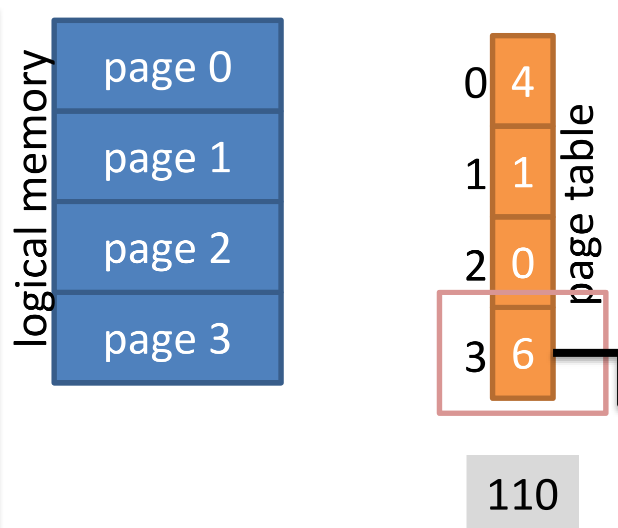
Attack in hardware

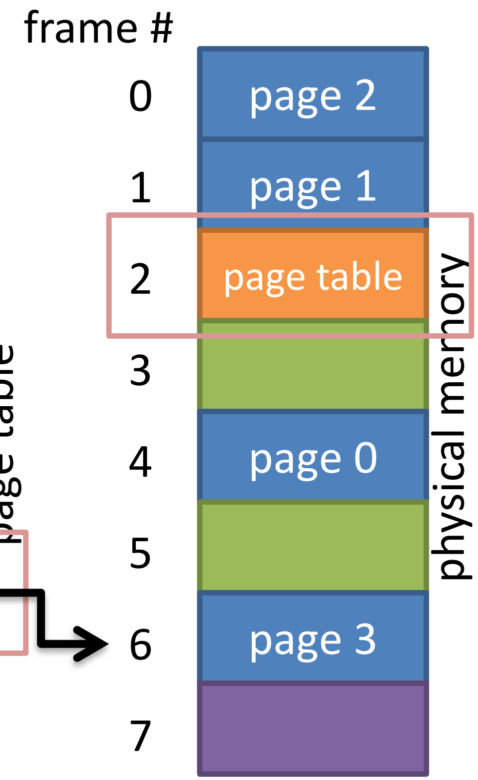


- Kernel privilege escalation
  - Induce a bit flip in a page table entry (PTE)

Attack in hardware

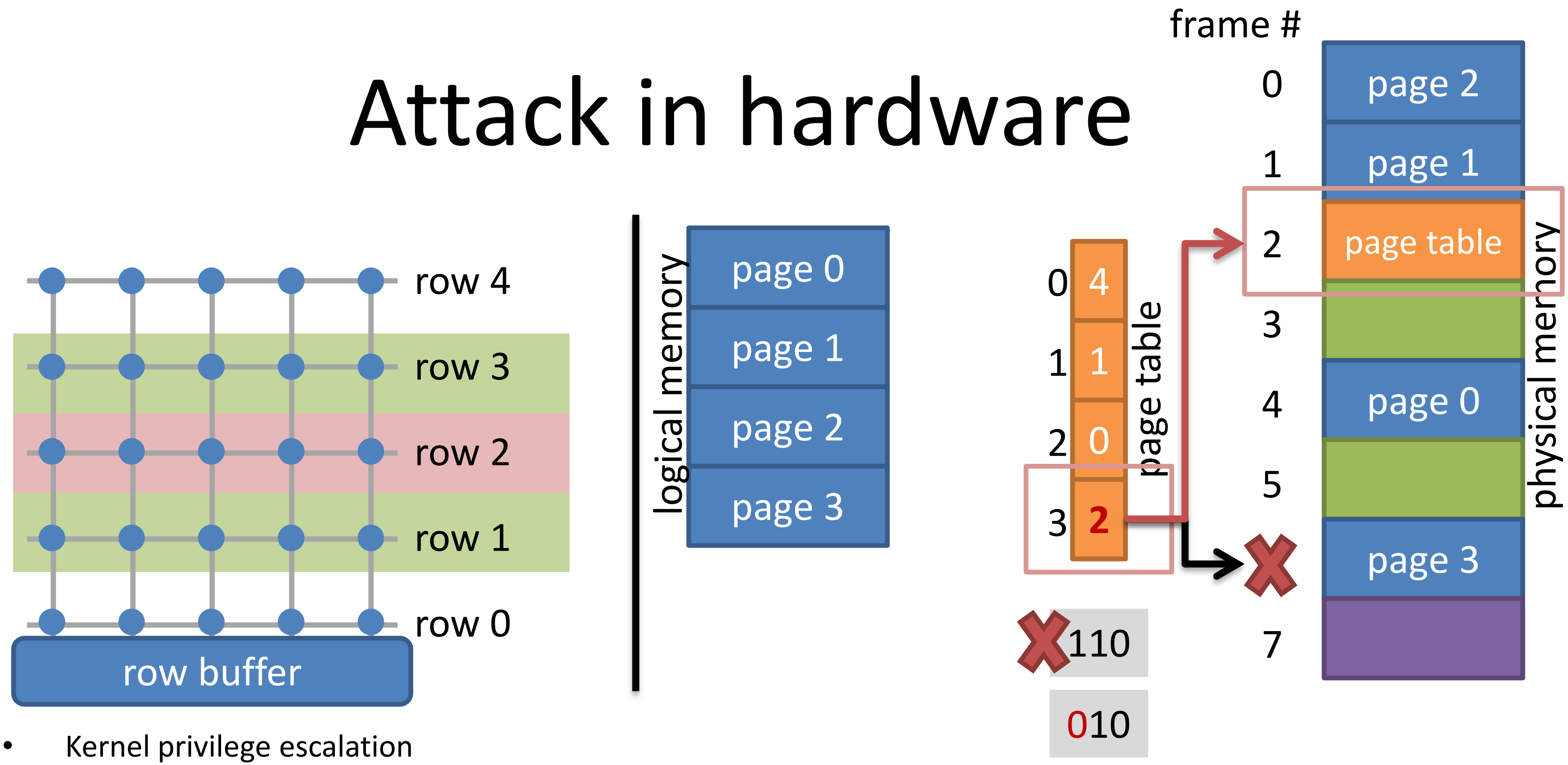






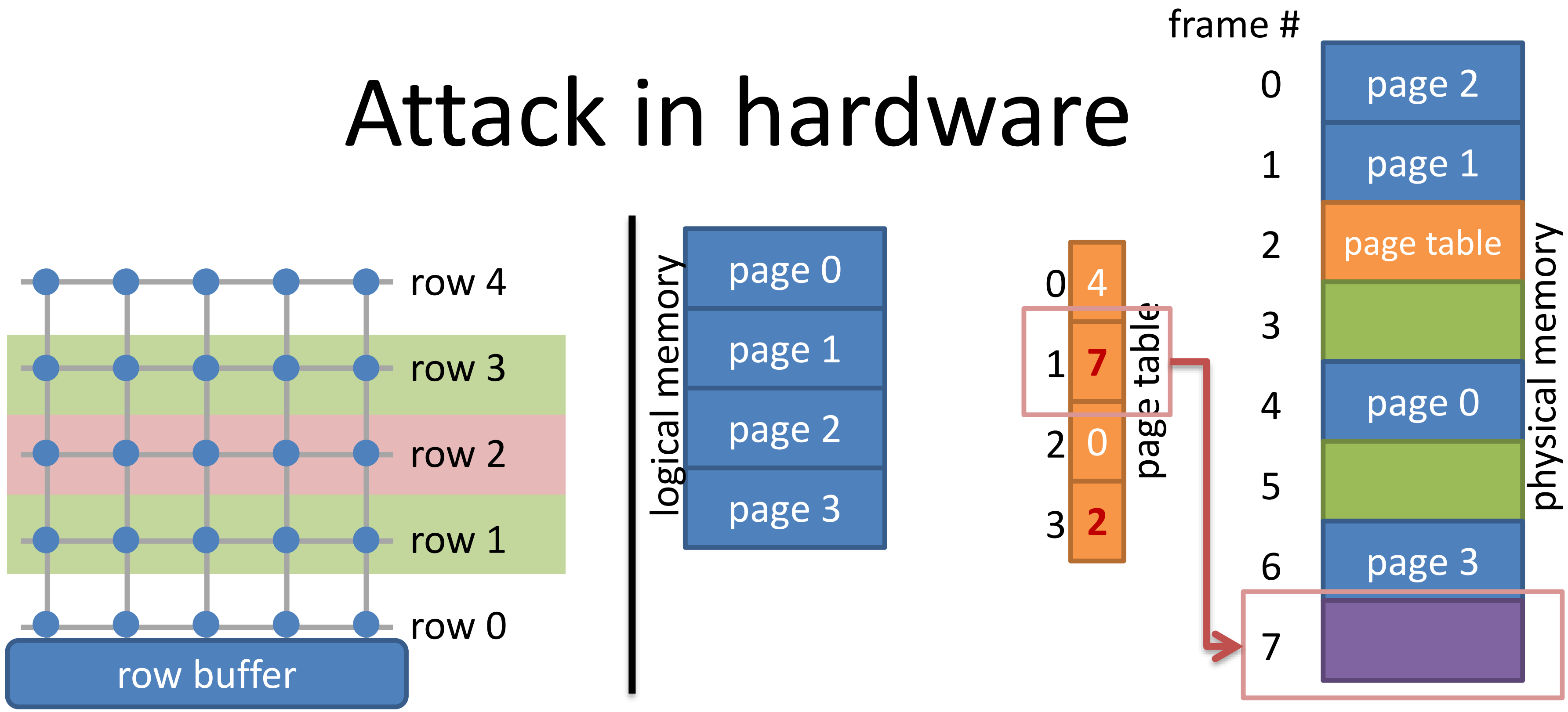
- Kernel privilege escalation
  - Induce a bit flip in a page table entry (PTE)

Attack in hardware



- Kernel privilege escalation
  - Induce a bit flip in a page table entry (PTE)
  - The PTE to then points to a physical page containing a page table of the attacking process.
  - ➔ This gives the attacking process read-write access to one of its own page tables, and hence to all of physical memory.

Attack in hardware



- Kernel privilege escalation
  - Induce a bit flip in a page table entry (PTE)
  - The PTE to then points to a physical page containing a page table of the attacking process.
  - ➔ This gives the attacking process read-write access to one of its own page tables, and hence to all of physical memory.
- Results
  - Modify a SUID-root executable such as /bin/ping, overwriting its entry point with attack shell code, and then run it.
  - Shell code will then run as root.

More information

- Cleverly combining problems in hardware with software: HW → OS → application program (known since 2012)

- http://googleprojectzero.blogspot.se/2015/03/exploiting-dram-rowhammer-bug-to-gain.html
  - https://github.com/google/rowhammer-test
- **C5: Cross-Cores Cache Covert Channel**, Clémentine Maurice, Christoph Neumann, Olivier Heen and Aurélien Francillon. 2015
- **Reverse Engineering Intel Last-Level Cache Complex Addressing Using Performance Counters,** Clémentine Maurice; Nicolas Le Scouarnec; Christoph Neumann; Olivier Heen; Aurélien Francillon. 2015

Spectre and Meltdown

Magnus Almgren

Spectre		Meltdown
Variant 1 CVE-2017-5753	Variant 2 CVE-2017-5715	Variant 3 CVE-2017-5754

Overview

"The flaws **violate** central computer science **isolation principles** that laid the foundation for modern sandboxing that protects your applications from attack by a browser; multi-user computing that protects your documents from another user logged into the same server; and multi-tenancy that protects your entire virtual machine from another virtual machine on the same metal host"

Alex Ionescu

1. bounds check bypass
2. branch target injection
3. rouge data cache load

- Sources:
- https://spectreattack.com/spectre.pdf
- https://blog.google/topics/google-cloud/answering-your-questions-about-meltdown-and-spectre/
- https://www.wired.com/story/critical-intel-flaw-breaks-basic-security-for-most-computers/?mbid=BottomRelatedStories

Concepts

- Operating System Defenses: Separation (in L09)
  - Physical, Temporal, Logical, Cryptographic
- Covert Channels & Side Channels: “leaking”unattended information
  - Cache: if variable recently used, in cache
  - Also branch prediction, etc.
- Return-Oriented Programming
  - Related to buffer overflows, redirecting the control flow
- Speculative Execution: …

Speculative Execution (1)

```
if (x < array1_size)
    y = array2[array1[x] * 256];
```

- Is x inside the array, then get value and index array2

- Where are the variables stored?
  - Register in the CPU? **fast**
  - Cache: L1, L2, L3
  - Main memory
  - Disk
  - Network **slow**

Typical orders of magnitude

execute typical instruction	**1/1,000,000,000 sec = 1 nanosec**
fetch from L1 cache memory	**0.5 nanosec**
branch misprediction	5 nanosec
fetch from L2 cache memory	**7 nanosec**
Mutex lock/unlock	25 nanosec
fetch from main memory	**100 nanosec**
send 2K bytes over 1Gbps network	20,000 nanosec
read 1MB sequentially from memory	250,000 nanosec
fetch from new disk location (seek)	8,000,000 nanosec
read 1MB sequentially from disk	20,000,000 nanosec
send packet US to Europe and back	150 milliseconds = 150,000,000 nanosec

http://norvig.com/21-days.html#answers

Speculative Execution (2)

```
if (x < array1_size)
    y = array2[array1[x] * 256];
```

- Is x inside the array, then get value and index array2

- Where are the variables stored?
  - **Main memory: 100 ns**
- We will "lose" 100 instructions while waiting for the answer.

→ Try to guess instead.

If right, have done work. If wrong, just rewind.

Speculative Execution (3)

execute code

`if (x < array1_size)`

x in cache, array1_size in main memory CM

ask for array1_size but in parallel go on

save register state

state0

state0

predict branch

execute code “internally” CM

`y = array2[array1[x] * 256];`

state1

value of array1_size arrived

did we take the right branch?

state1

state0

value of array2[array1[x]*256] arrives, and will be stored in cache

Typical orders of magnitude

Side channel!!!

execute typical instruction	**1/1,000,000,000 sec = 1 nanosec**
fetch from L1 cache memory	**0.5 nanosec**
branch misprediction	5 nanosec
fetch from L2 cache memory	**7 nanosec**
Mutex lock/unlock	25 nanosec
fetch from main memory	**100 nanosec**
send 2K bytes over 1Gbps network	20,000 nanosec
read 1MB sequentially from memory	250,000 nanosec
fetch from new disk location (seek)	8,000,000 nanosec
read 1MB sequentially from disk	20,000,000 nanosec
send packet US to Europe and back	150 milliseconds = 150,000,000 nanosec

http://norvig.com/21-days.html#answers

Speculative Execution (3)

execute code

if (**x** < array1_size)

Goal: array1[x] secret

x in cache, array1_size in main memory

CM

ask for array1_size but in parallel go on

save register state

state0

state0

predict branch

execute code “internally”

CM

y = array2[array1[x] * 256];

state1

value of array1_size arrived

did we take the right branch?

state1

state0

value of array2[array1[x]*256] arrives, and will be stored in cache

```c
/**************************************************************************************
Victim code.
**************************************************************************************/
unsigned int array1_size = 16;
uint8_t unused1[64];
uint8_t array1[160] = { 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16 };
uint8_t unused2[64];
uint8_t array2[256 * 512];

char *secret = "The Magic Words are Squeamish Ossifrage.";

uint8_t temp = 0; /* Used so compiler won’t optimize out victim_function() */

void victim_function(size_t x) {
        if (x < array1_size) {
                temp &= array2[array1[x] * 512];
        }
}
```

https://spectreattack.com/spectre.pdf

Speculative Execution (3)

execute code

x in cache, array1_size in main memory CM

ask for array1_size but in parallel go on

save register state

predict branch

execute code “internally” CM

value of array1_size arrived

did we take the right branch?

if (**x** < array1_size)

Goal: array1[x]=k secret

- array1[x] in cache (value)
- array2[…] not in cache (full)

y = array2[array1[x] * 256];

Value fetched depends on secret k

state0 | state0

state1

state1 | state0

value of array2[array1[x]*256] arrives, and will be stored in cache

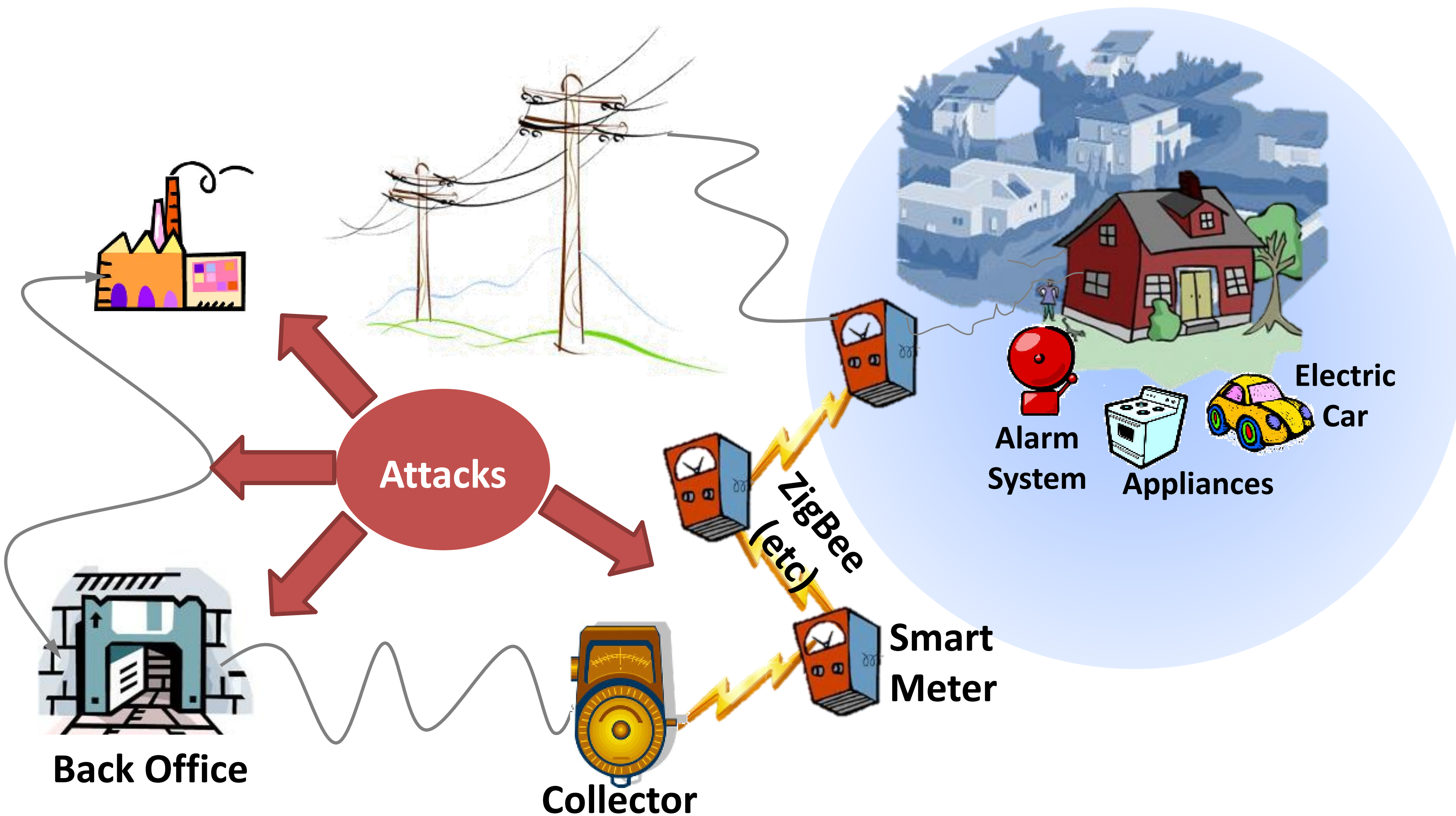
Cell Phones: sensors & actuators

- Many sensors & actuators
  - Accelerometer
  - Gyroscope
  - Magnetometer
  - GPS
  - Proximity sensor */ ambient light
  - Camera
  - Microphone
  - Speaker
  - “connections to networks”
- Need permissions to access the more sensitive ones
  - What are these?

https://upload.wikimedia.org/wikipedia/commons/7/7e/New_Mobile_CellPhone_Infos.jpg

The Smart Grid: Overview

- The Smart Grid – a modernization of the electric delivery system.
- Two-way flow of electricity & information with “intelligent nodes” to gain advantages from distributed computing.
- But – nobody knows what it will become.
  - “like Internet ~1990 – before Mosaic and Netscape.”
- Different phases:
  - First phase: Advanced Metering Infrastructure
  - Future: Important to curb greenhouse gas emissions


Attacks
Electric
Car
Alarm
System
Appliances
ZigBee
(etc)
Smart
Meter
Back Office
Collector

Why The Smart Grid?

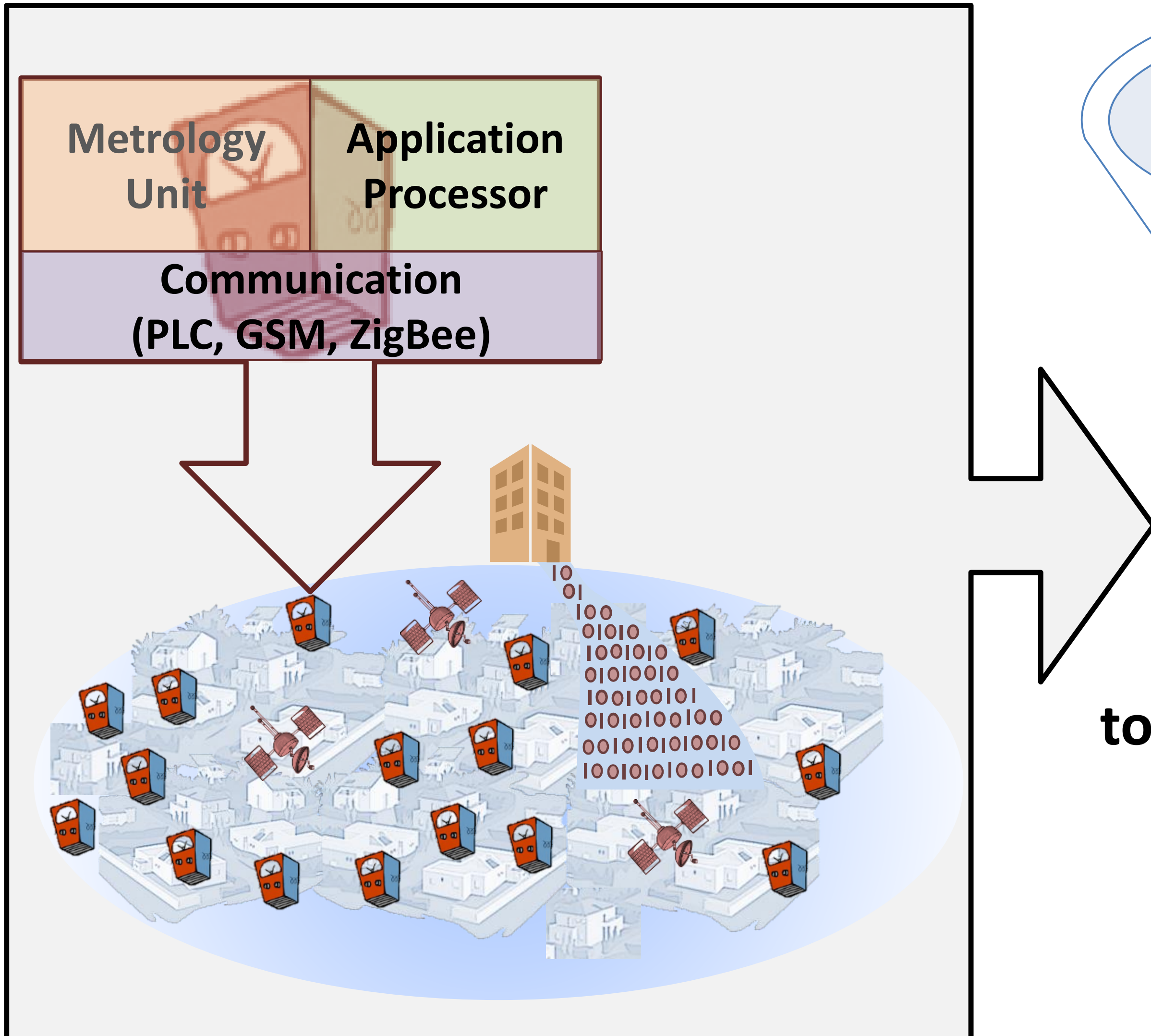
- Electrical grid ”old” & climate crisis driving green tech.
- New challenges:
  - Green power such as wind, available ***only at certain times.***
  - Generation / load no longer fixed geographically, ***but may move*** (typically the electrical car).

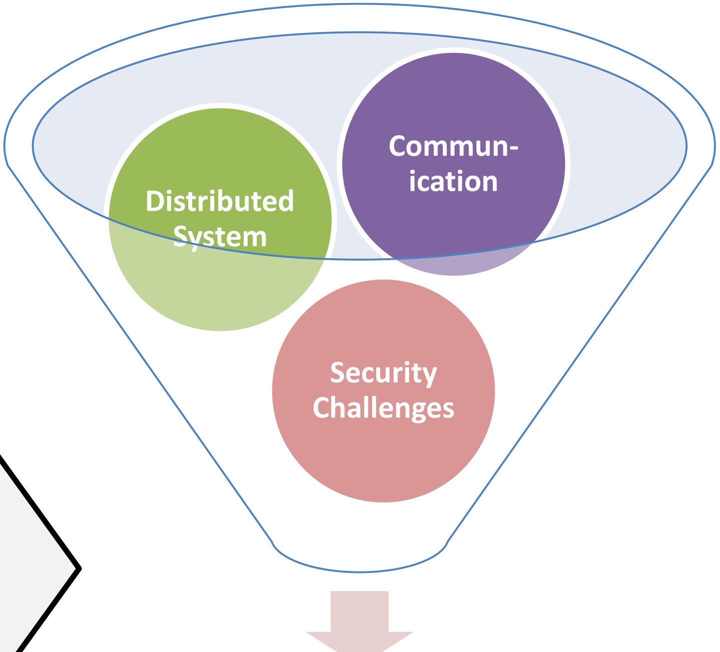
- Solution: Add ICT to upgrade the grid.
  - People talk about the ”smart grid” but what it will entail?
  - First step is the ”Smart meters,” then
  - upgrading components in the field, then
  - ”apps” to control home appliances etc.

The Smart Grid and Security: Interdisciplinary field

- Power Engineers
  - Safety is a priority
  - Know nothing about ICT, communication or security.
  - Attitude often: But we use encryption between the devices
  - Devices last for 20—50 years
- Security Experts
  - Know very little about the physical laws and the networks
  - Little comprehension for the need to keep systems running 24/7
  - Devices updated weekly, life expentancy 3—5 years
- Security problems already demonstrated in some widely deployed devices (smart meters)
  - Can be hacked but also come with privacy concerns

AMI from an ICT Perspective

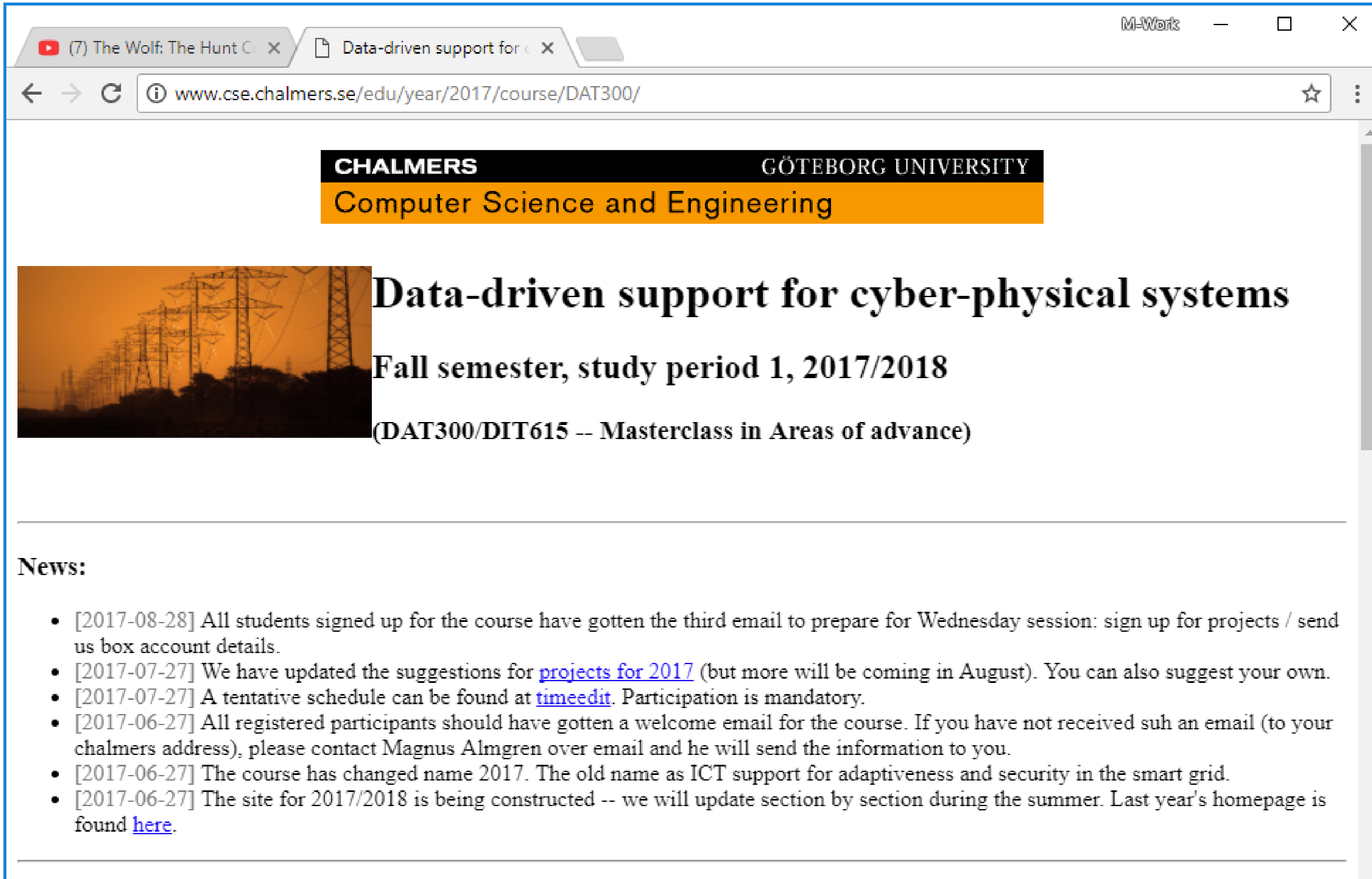




ICT Expertise Needed to Strengthen the System

Project Course: DAT300

Data-driven support for cyber-physical systems


(7) The Wolf: The Hunt C
Data-driven support for c
M-Work
www.cse.chalmers.se/edu/year/2017/course/DAT300/

CHALMERS GÖTEBORG UNIVERSITY
Computer Science and Engineering

Data-driven support for cyber-physical systems

Fall semester, study period 1, 2017/2018

(DAT300/DIT615 -- Masterclass in Areas of advance)

News:

- [2017-08-28] All students signed up for the course have gotten the third email to prepare for Wednesday session: sign up for projects / send us box account details.
- [2017-07-27] We have updated the suggestions for projects for 2017 (but more will be coming in August). You can also suggest your own.
- [2017-07-27] A tentative schedule can be found at timeedit. Participation is mandatory.
- [2017-06-27] All registered participants should have gotten a welcome email for the course. If you have not received suh an email (to your chalmers address), please contact Magnus Almgren over email and he will send the information to you.
- [2017-06-27] The course has changed name 2017. The old name as ICT support for adaptiveness and security in the smart grid.
- [2017-06-27] The site for 2017/2018 is being constructed -- we will update section by section during the summer. Last year's homepage is found here.

---

http://www.cse.chalmers.se/edu/course/DAT285B/

Data-driven support for cyber-physical systems

- Goals
  - Letting students from computer science and other disciplines be introduced to advanced interdisciplinary concepts related to the smart grid, thus
  - building an understanding of the vocabulary and important terms that may have different meanings in the individual disciplines, and
  - investigating a domain-specific problem relevant to the smart grid that need an understanding beyond the traditional ICT field.

Environment

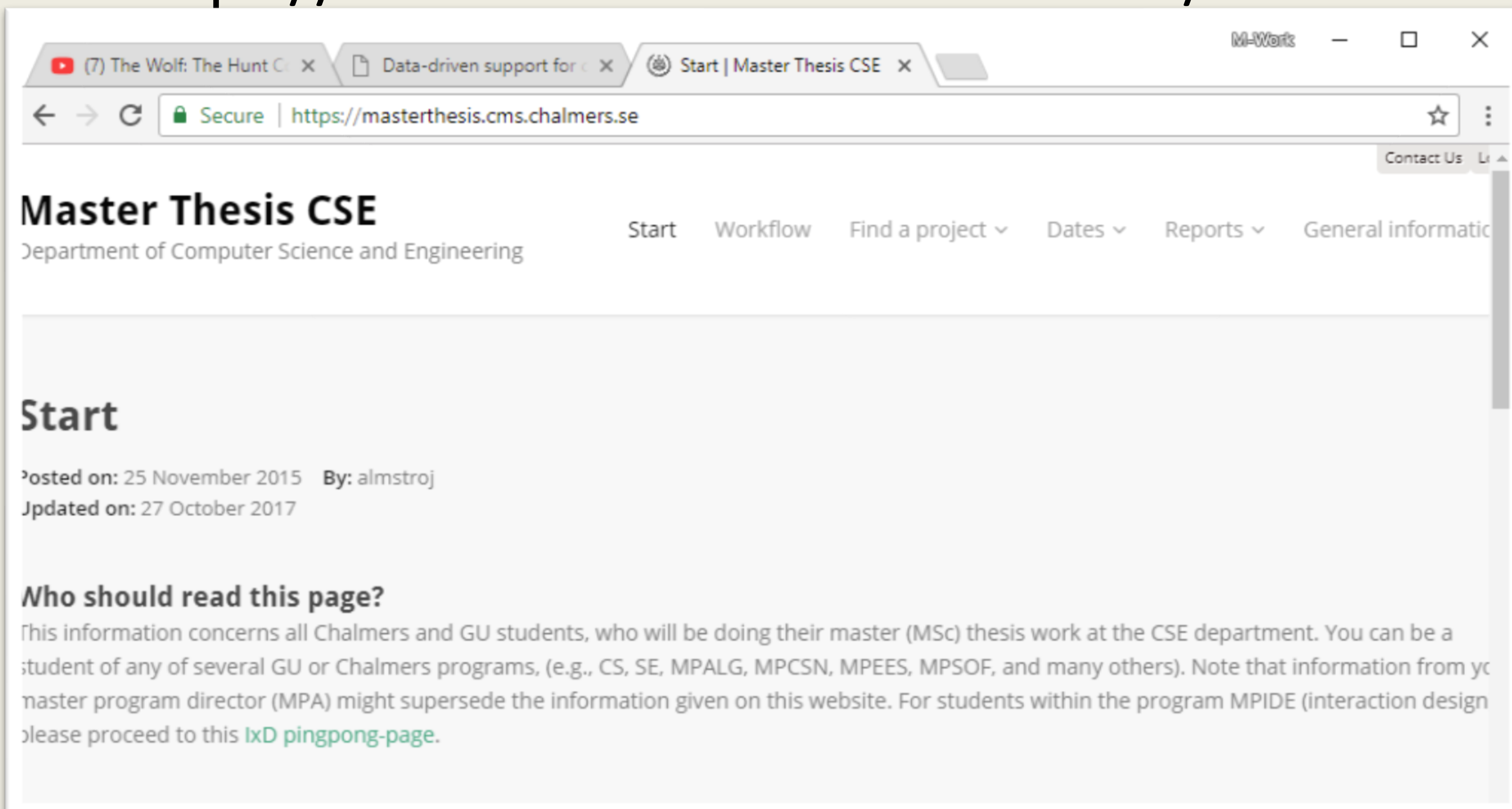
- Based on both the present and future design of the smart grid.
  - How can techniques from distributed systems be applied to large, heterogeneous systems where a massive amount of data will be collected?
  - How can such a system, containing legacy components with no security primitives, be made secure when the communication is added by interconnecting the systems?
- The students will have access to a hands-on lab, where they can run and test their design and code.

Course Setup

- The course is given on an advanced master's level, resulting in 7.5 points.
- The course setup
  - The first part of the course consists of lectures to introduce the students to each other and the two disciplines ("crash course").
  - The second part of the course will follow a seminar-style where research papers from both disciplines are actively discussed and then presented.
  - At the end of the course the students are also expected to present their respective project.

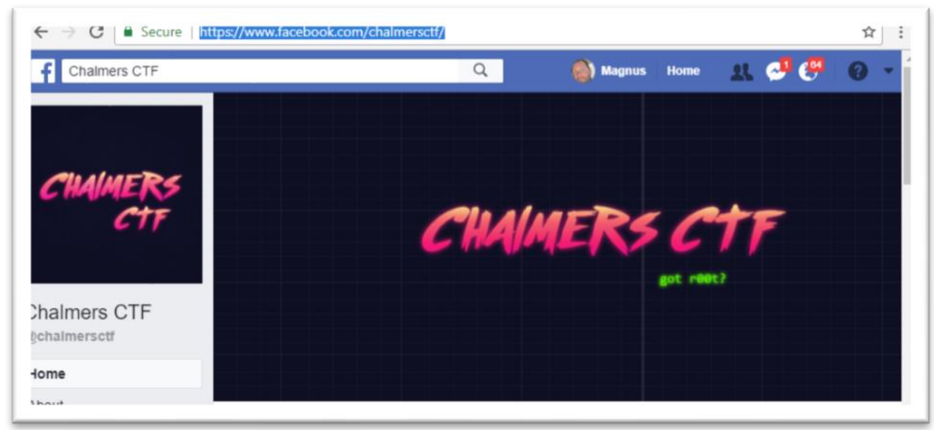
Master thesis and other projects

- Talk to us (early!)
  - https://masterthesis.cms.chalmers.se/



Capture the Flag Competition

- Team from Chalmers?



- https://www.facebook.com/chalmersctf/