Mapping Free & Open Source Deep Technical Training to the NICE Framework

(aka Making Cybersecurity Training Accessible)

Xeno Kovah

Disclaimer

- The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author.

Background

- My day job is not training, it's leading a firmware/BIOS security research team
  - See BlackHat 2013 "BIOS Security" talk
- I started OpenSecurityTraining.info to make my 8 days worth of classes available, and then I recruited MITRE colleagues to do the same
  - MITRE has been nice enough to set aside some money for me to coordinate internal training and get more material open sourced

One of the goals of OST

- Be able to train order of magnitude 100k security specialists
- That's less than 3300 people in each of the NIST NICE 31 sub-categories

There are only 2 paths

- Train trainers on common curricula (exponential bootstrapping of the first 500 or so people) + everyone has internal training
- Everybody attends centralized online training

Can we use…

- Colleges?
  - NSF SFS pays > 100k per student over 2 years for top schools
- College Online Classes?
  - *Slightly* cheaper & definitely more accessible than traditional schools, but still too expensive
- Khan Academy/Coursera/Udacity/EdX?
  - Free! (for now) But focused on general education, minimal security classes currently available

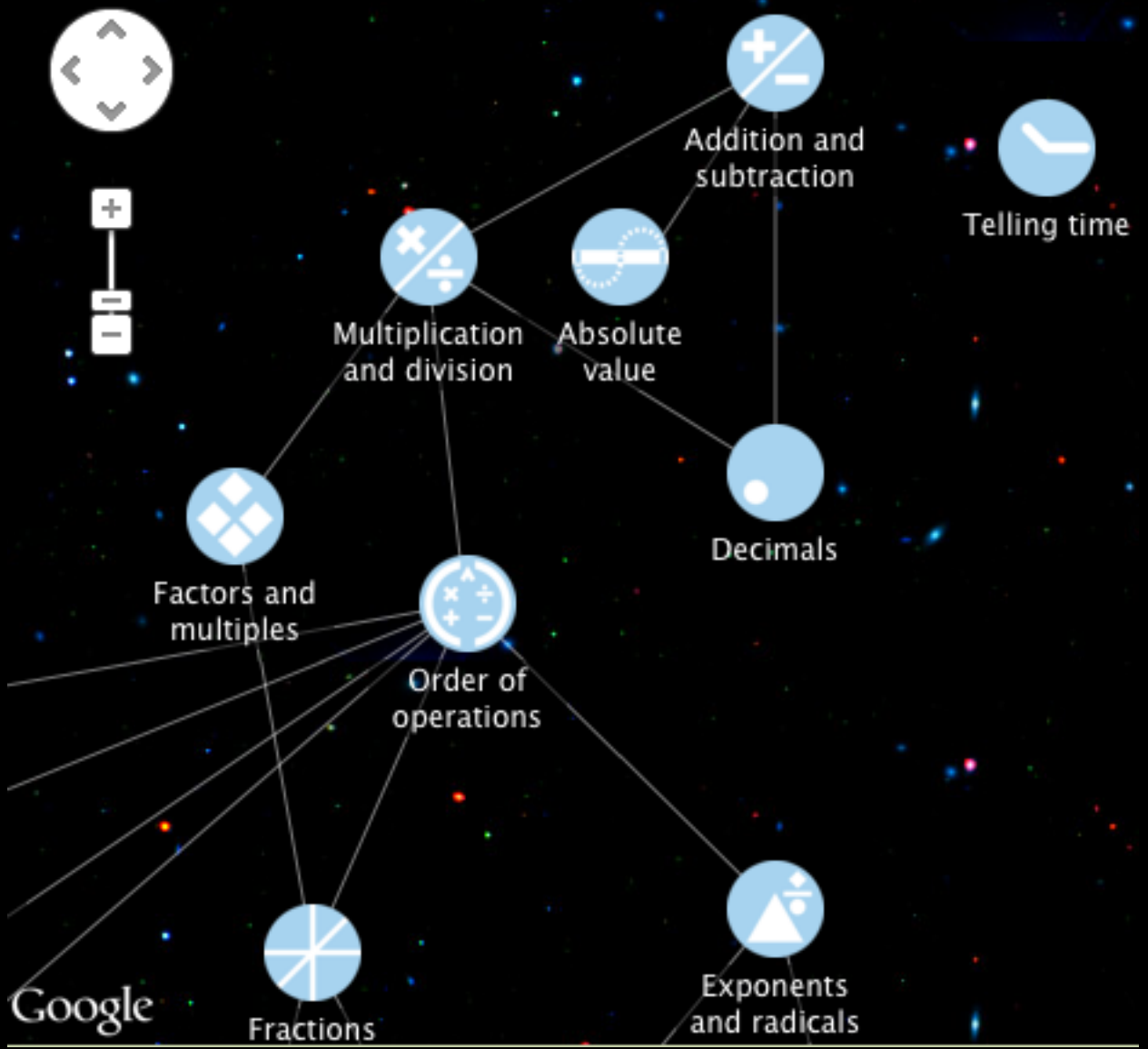
Can we use…

- Paid training?
  - How many days of training does someone need to be optimally effective? (Because you're lucky if you can find 20 days of non-overlapping training.)
    - Venue A: avg $1057/person/day
    - Venue B: avg $757/person/day
    - Venue C: avg $524/person/day
  - I think we should target $100/person/day as a goal
  - Full curricula are not available in most specialty areas
- Certifications?
  - Not deep enough at this time

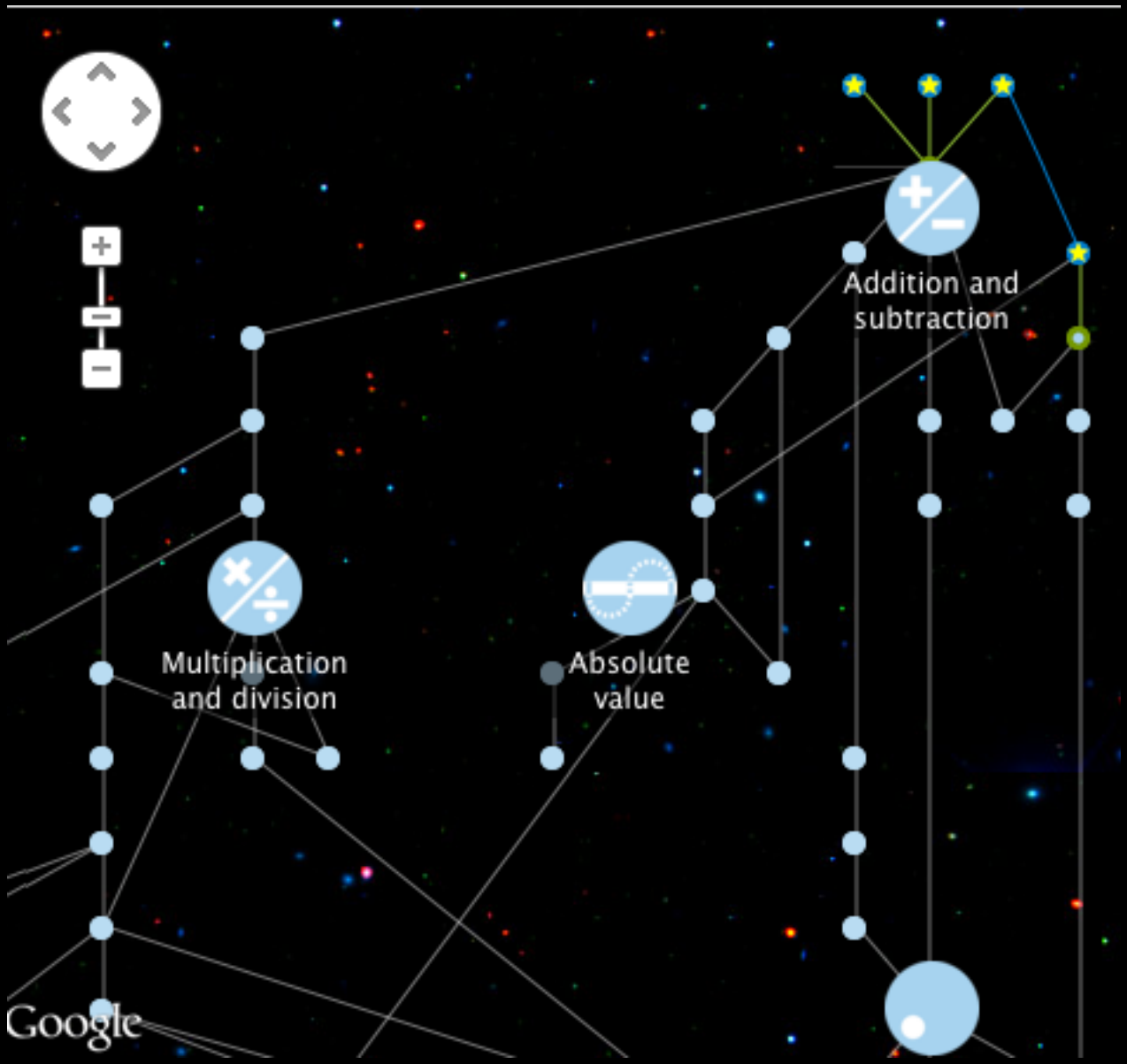
OST: Using what works

- ✓ Knowledge maps
- ✓ Open access AND Open materials
- ✓ Programatically-generated tests

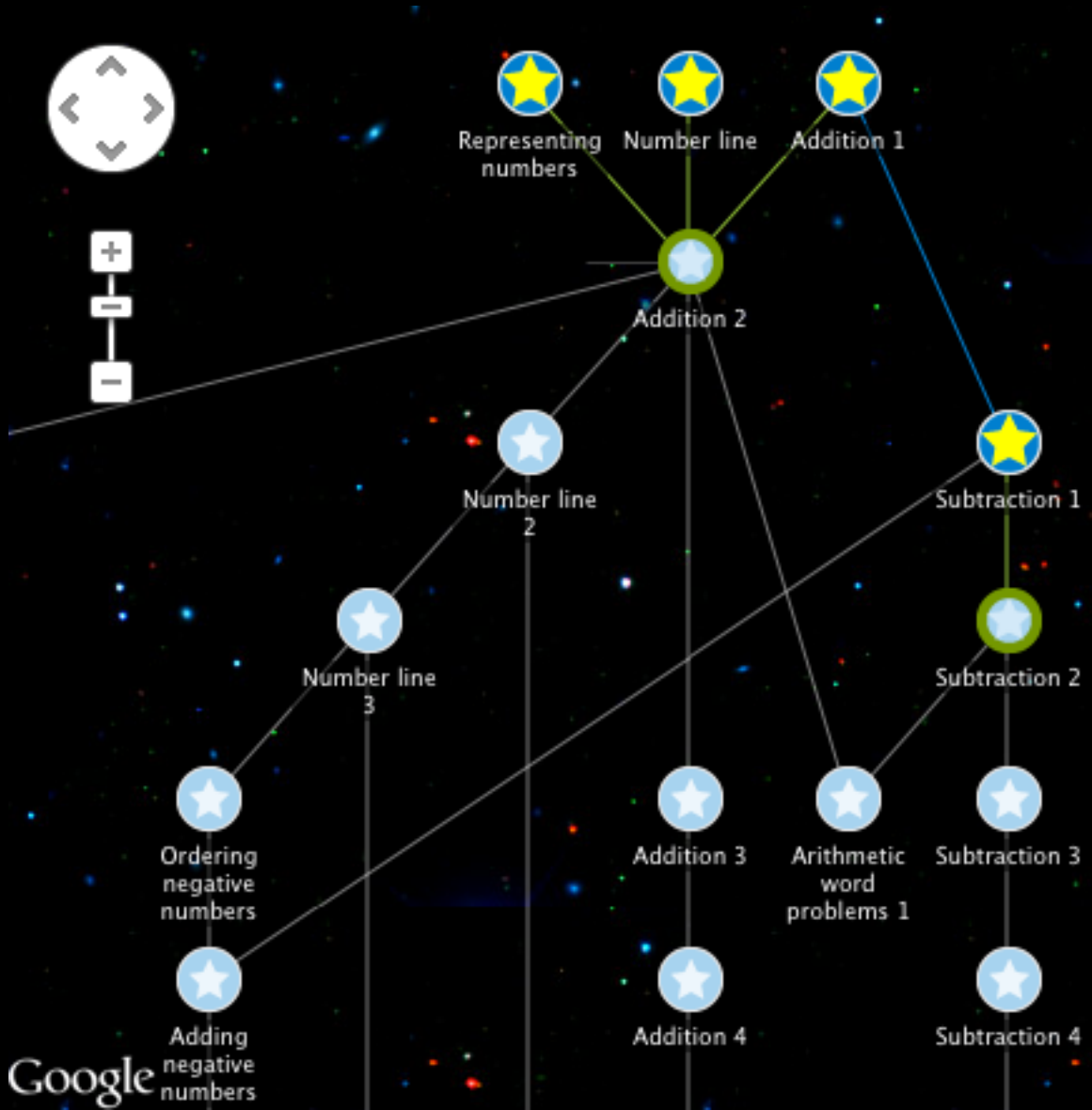
Khan Academy Knowledge Map



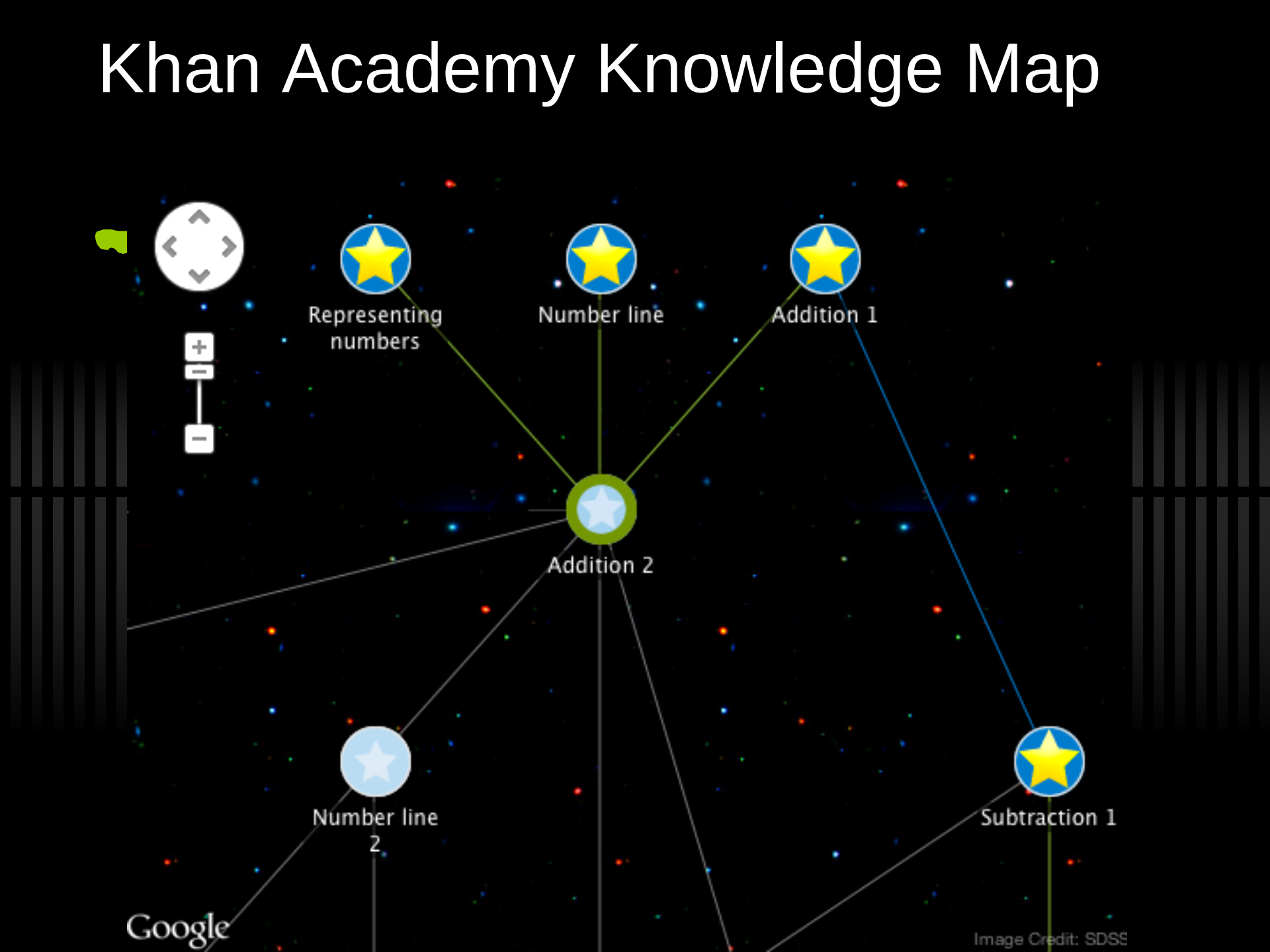
Khan Academy Knowledge Map



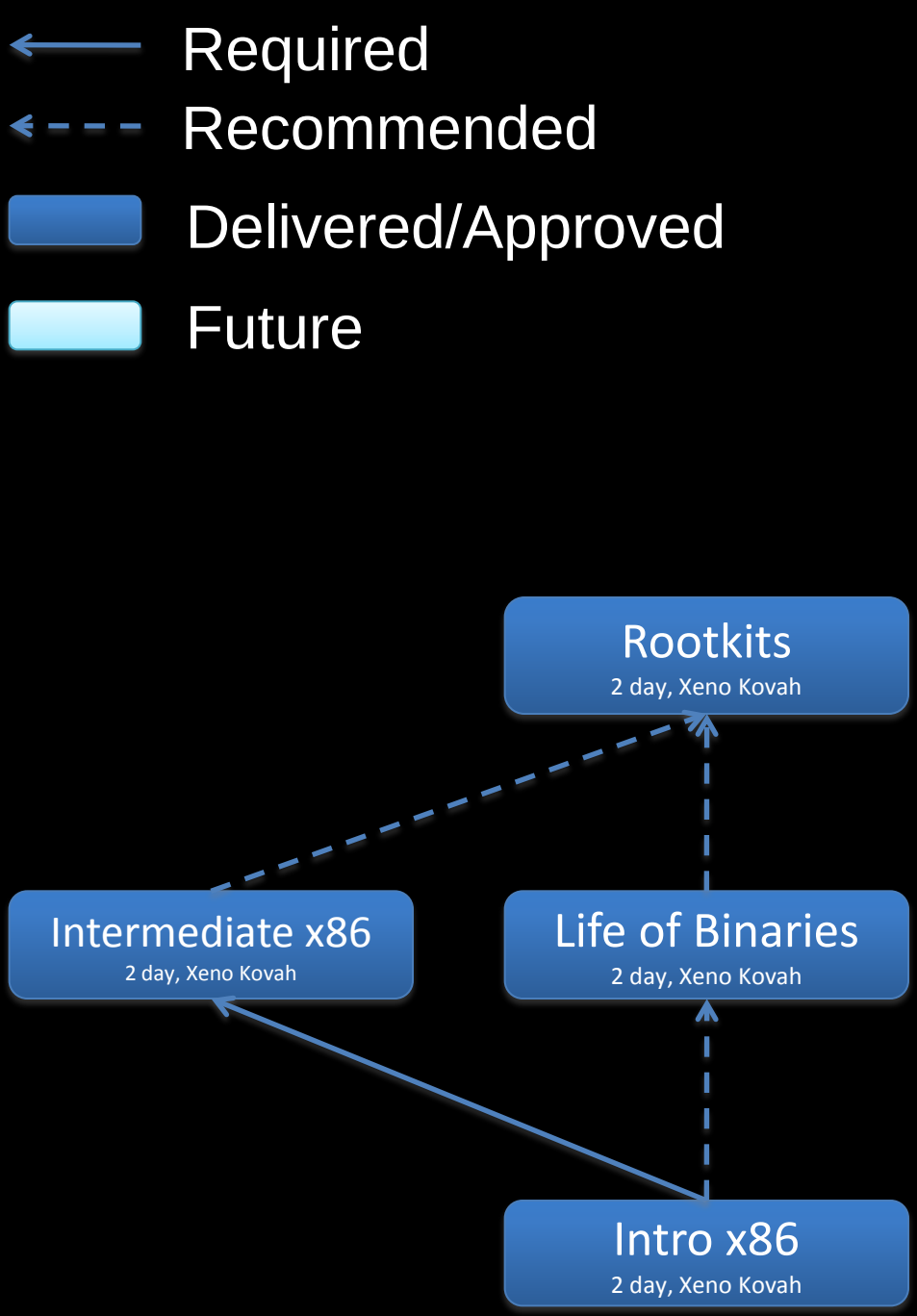
Khan Academy Knowledge Map



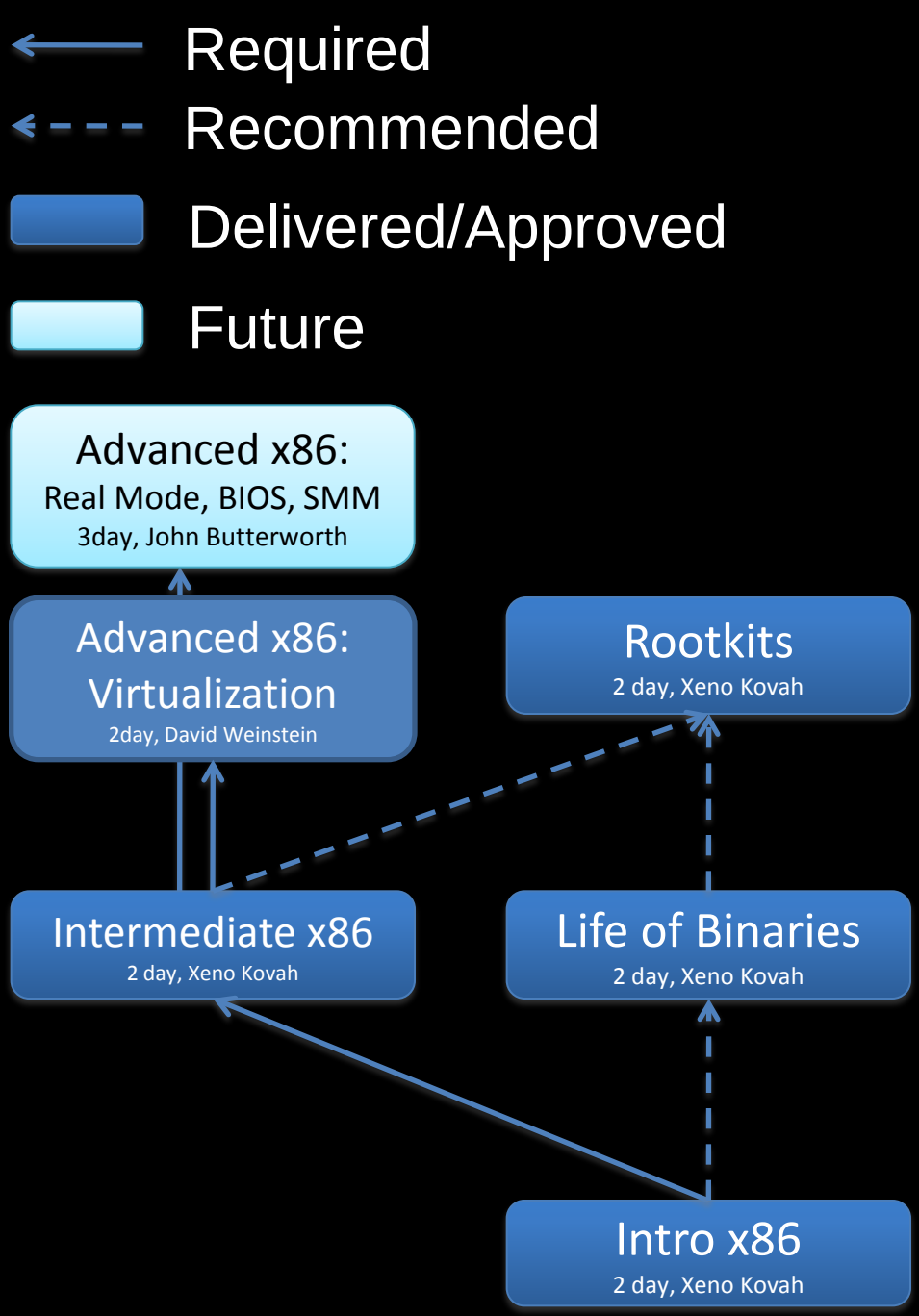
Khan Academy Knowledge Map



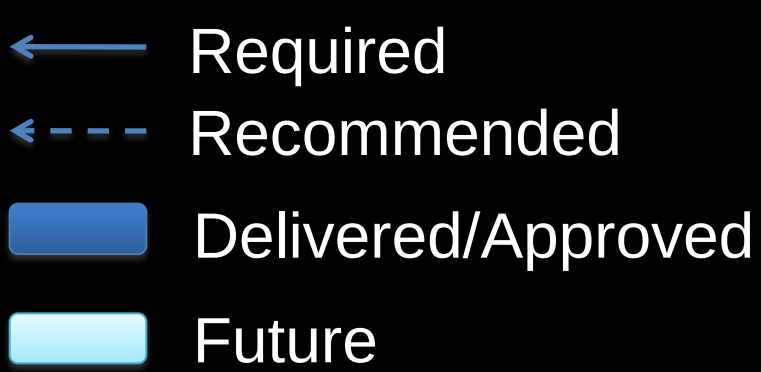
(partial) r0x0r Skill Tree

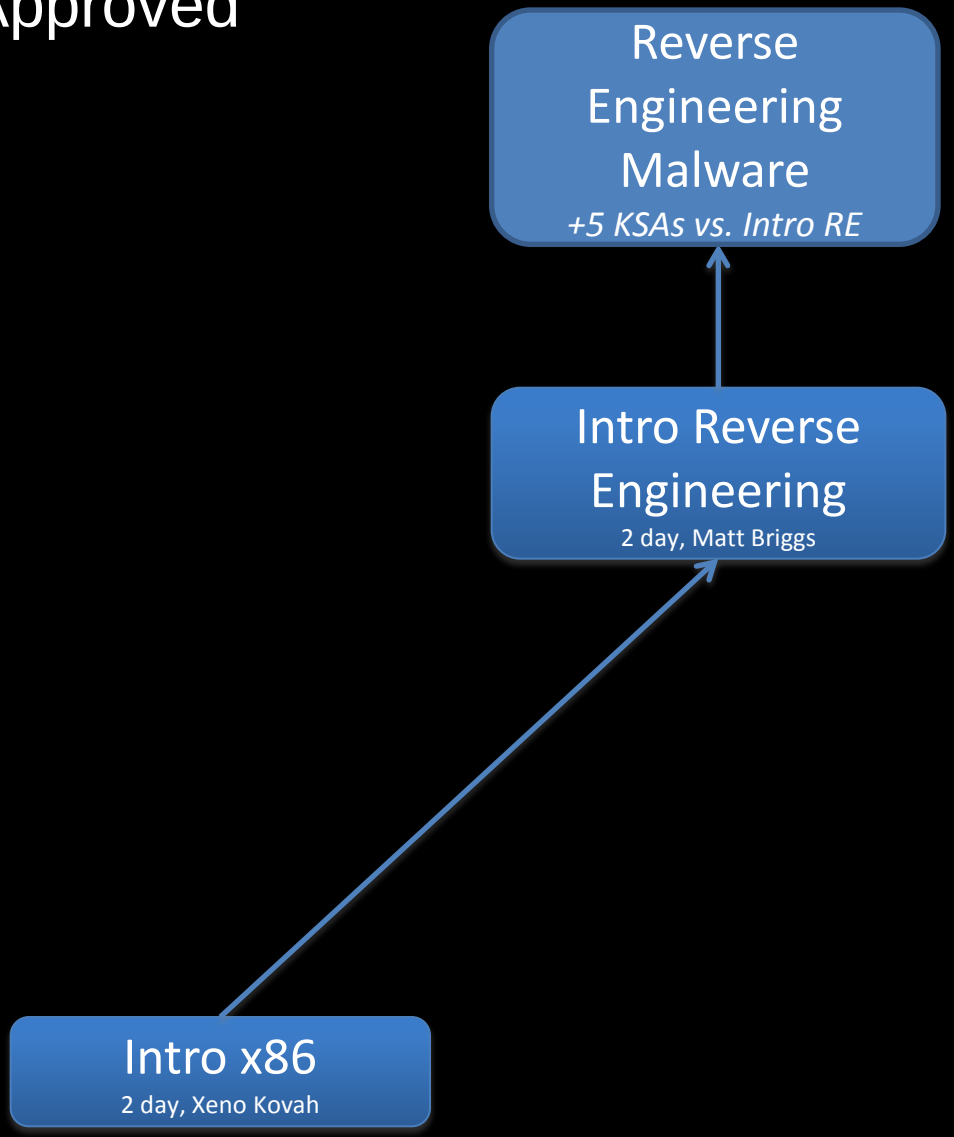


(partial) r0x0r Skill Tree

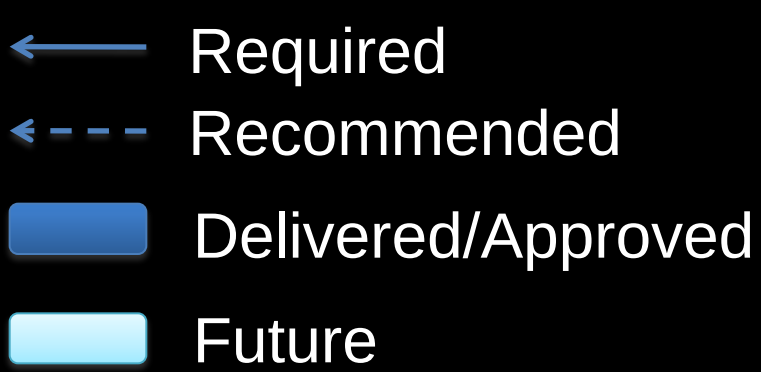


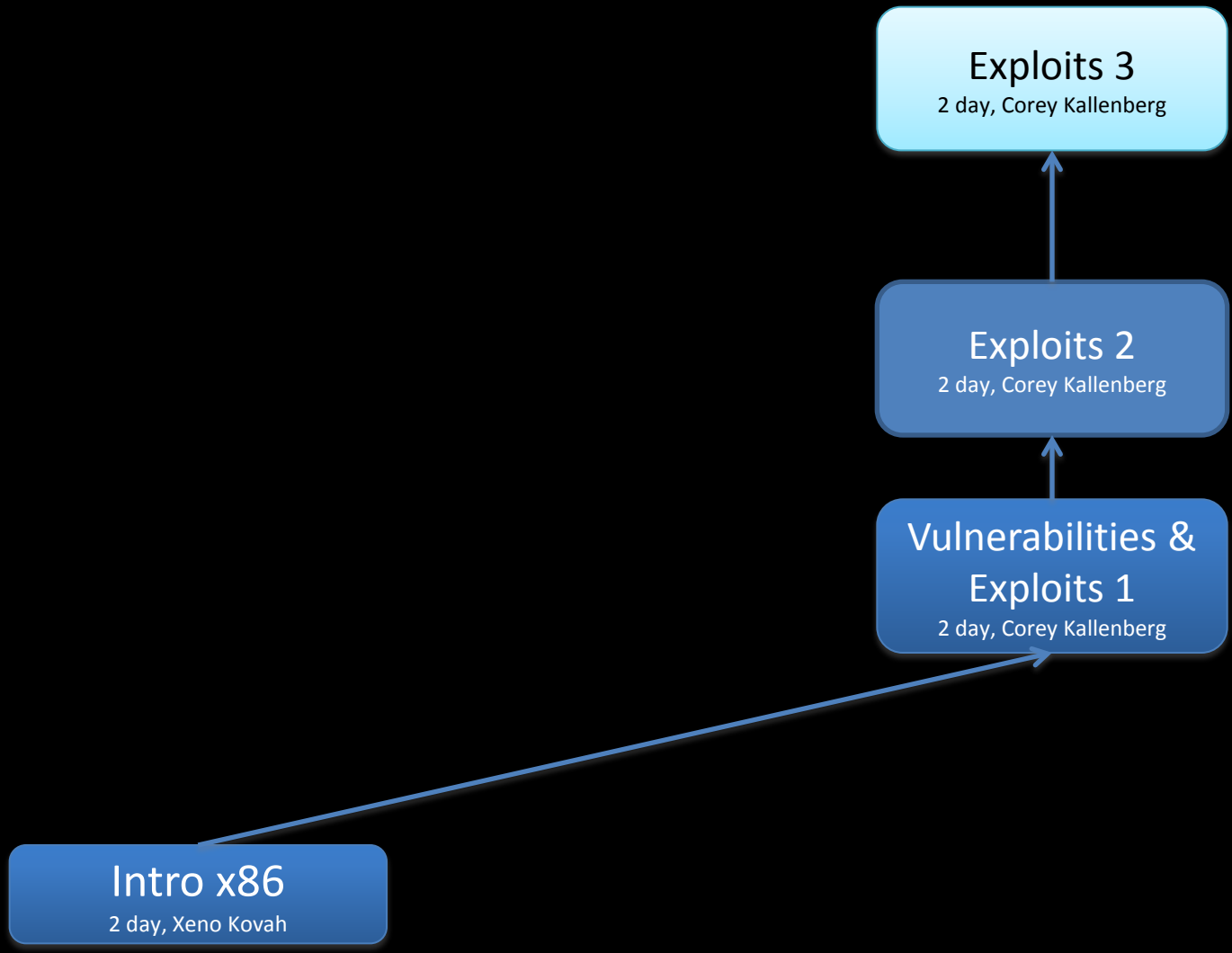
(partial) r0x0r Skill Tree



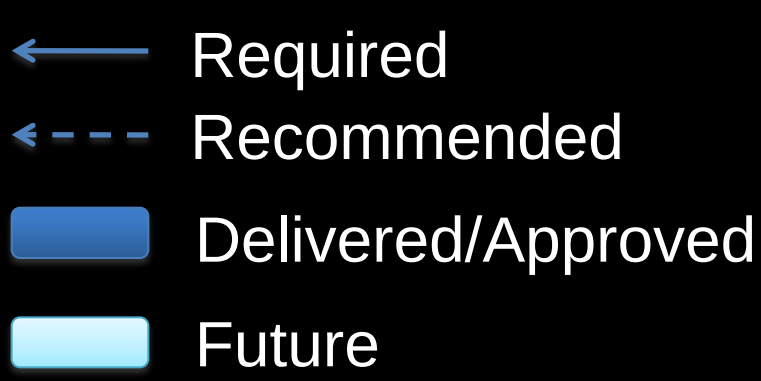


(partial) r0x0r Skill Tree





(partial) r0x0r Skill Tree



Reverse Engineering Malware
+5 KSAs vs. Intro RE

Exploits 3
2 day, Corey Kallenberg

Advanced x86: Real Mode
3day, John Butterworth

Advanced x86: Virtualization
2day, David Weinstein

Rootkits
2 day, Xeno Kovah

Intro Reverse Engineering
2 day, Matt Briggs

Exploits 2
2 day, Corey Kallenberg

Intermediate x86
2 day, Xeno Kovah

Life of Binaries
2 day, Xeno Kovah

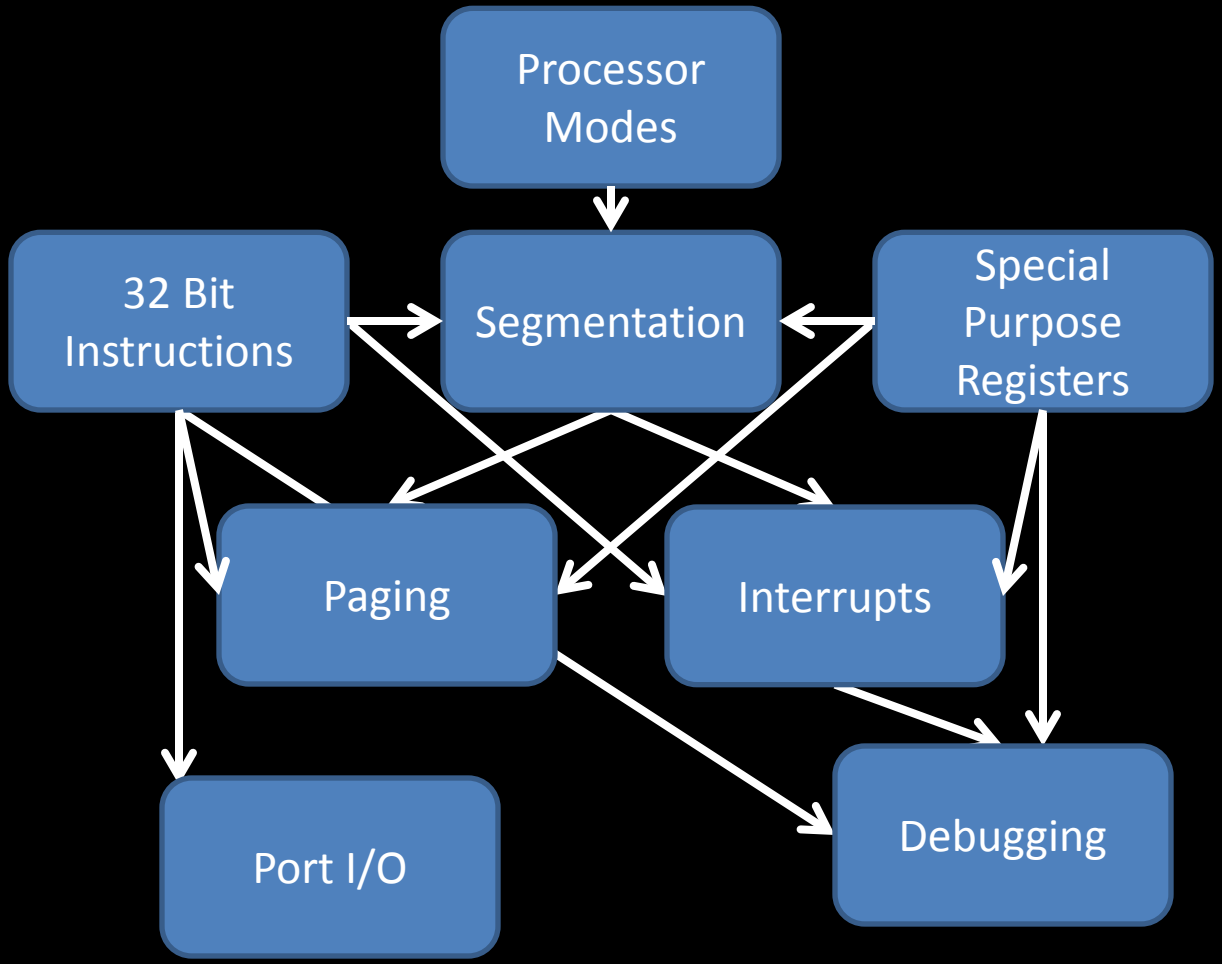
Vulnerabilities & Exploits 1
2 day, Corey Kallenberg

Intro x86
2 day, Xeno Kovah

OST goes *deep*

- There is a lot of super-shallow security training of questionable value out there right now
  - It will not create effective defenders
- We prioritize rare & deep training
  - E.g. x86 memory management, hardware support for virtualization, rootkits, trusted computing, netflow analysis, ARM assembly, cryptanalysis, static RE
  - This is where recruiting other MITRE people helps
- And we're now offering fine-grained knowledge maps for new topics as they're added

Intermediate x86

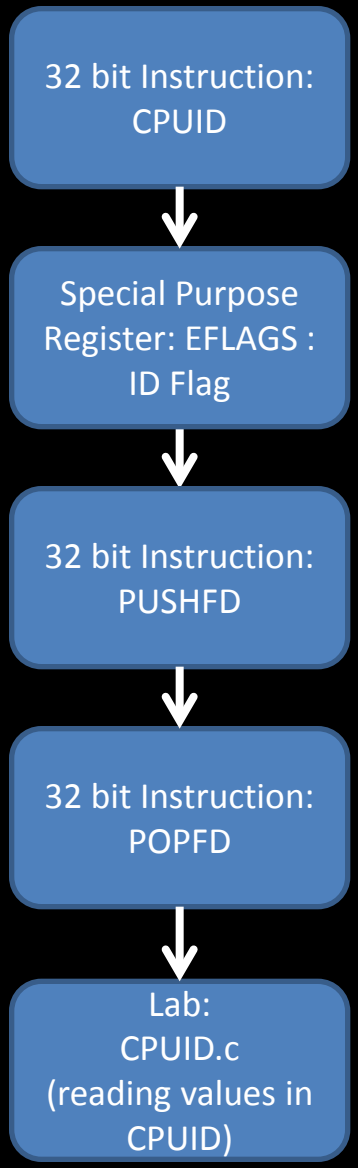

Processor Modes
32 Bit Instructions
Segmentation
Special Purpose Registers
Paging
Interrupts
Port I/O
Debugging

32 Bit Instructions

Processor Modes

Segmentation

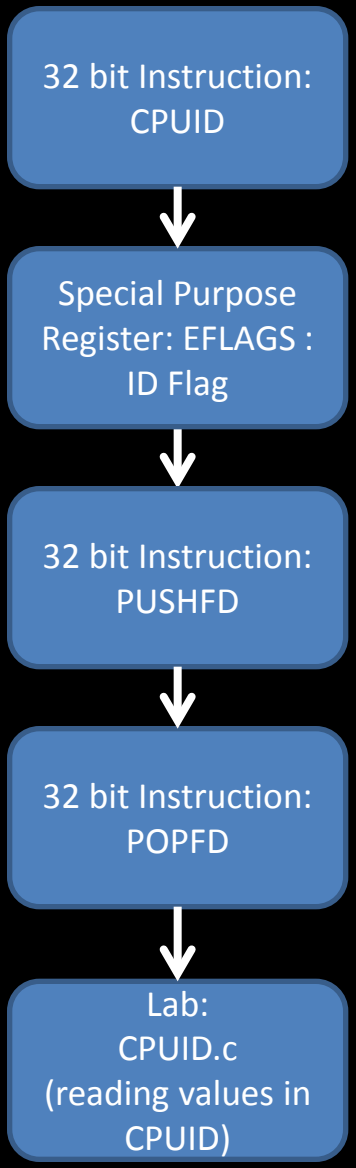
(Morning Warm Up)

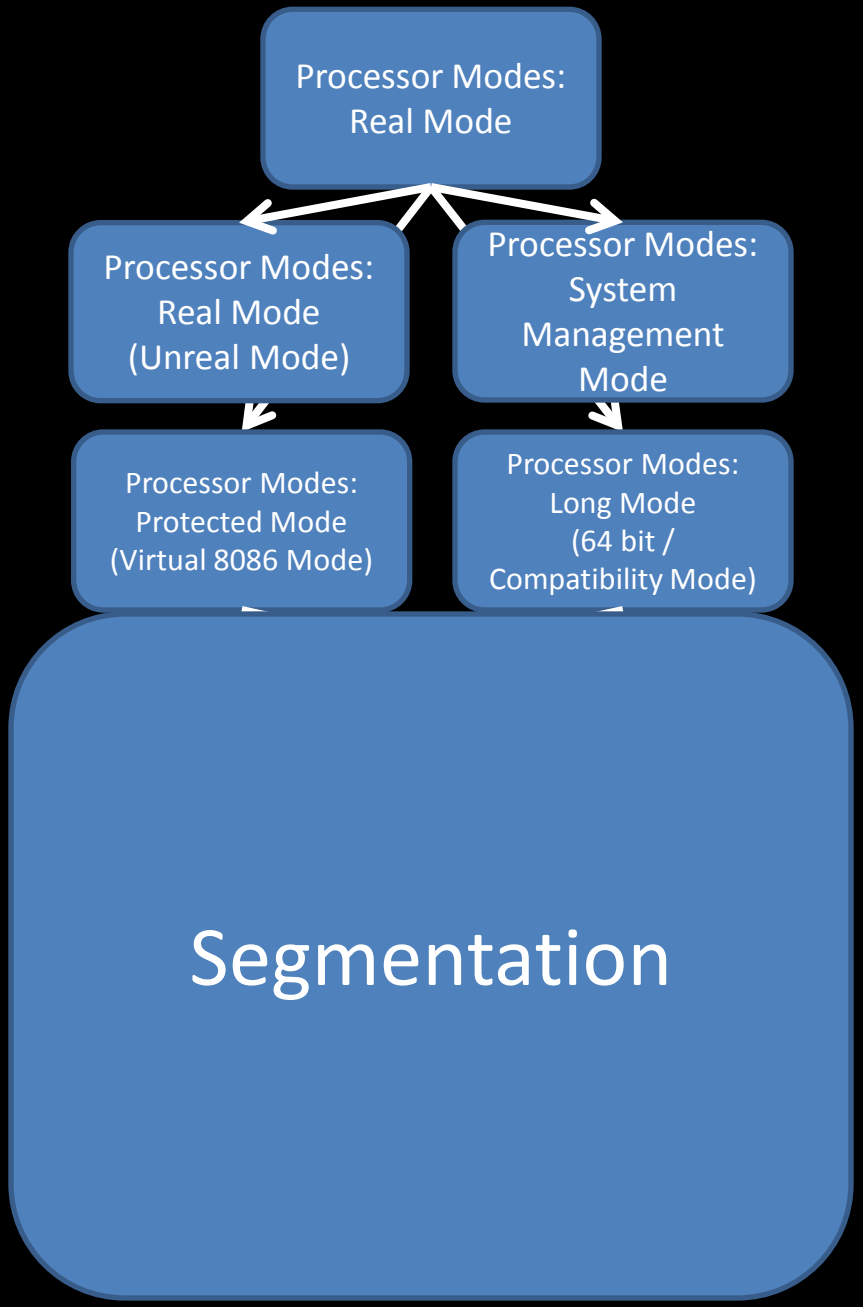


Processor Modes

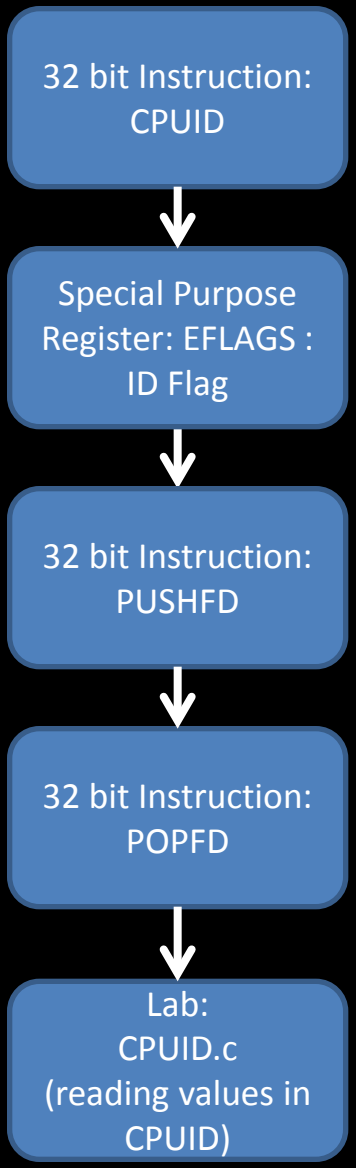
Segmentation

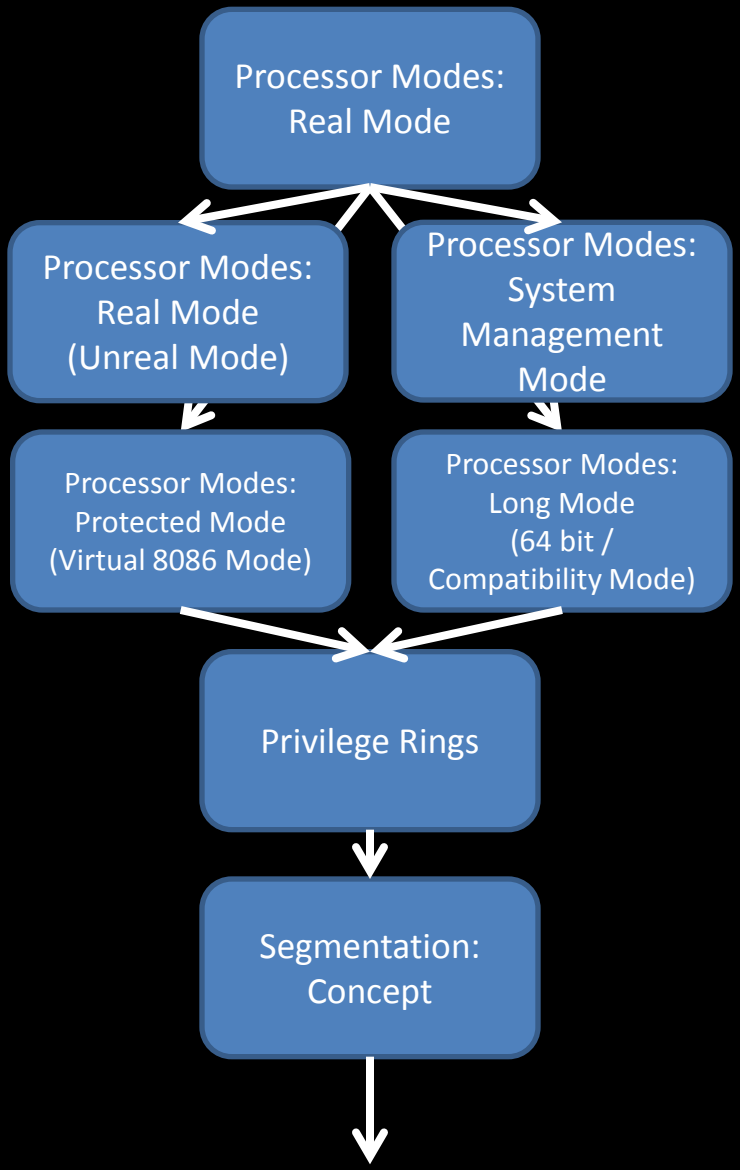
(Morning Warm Up)

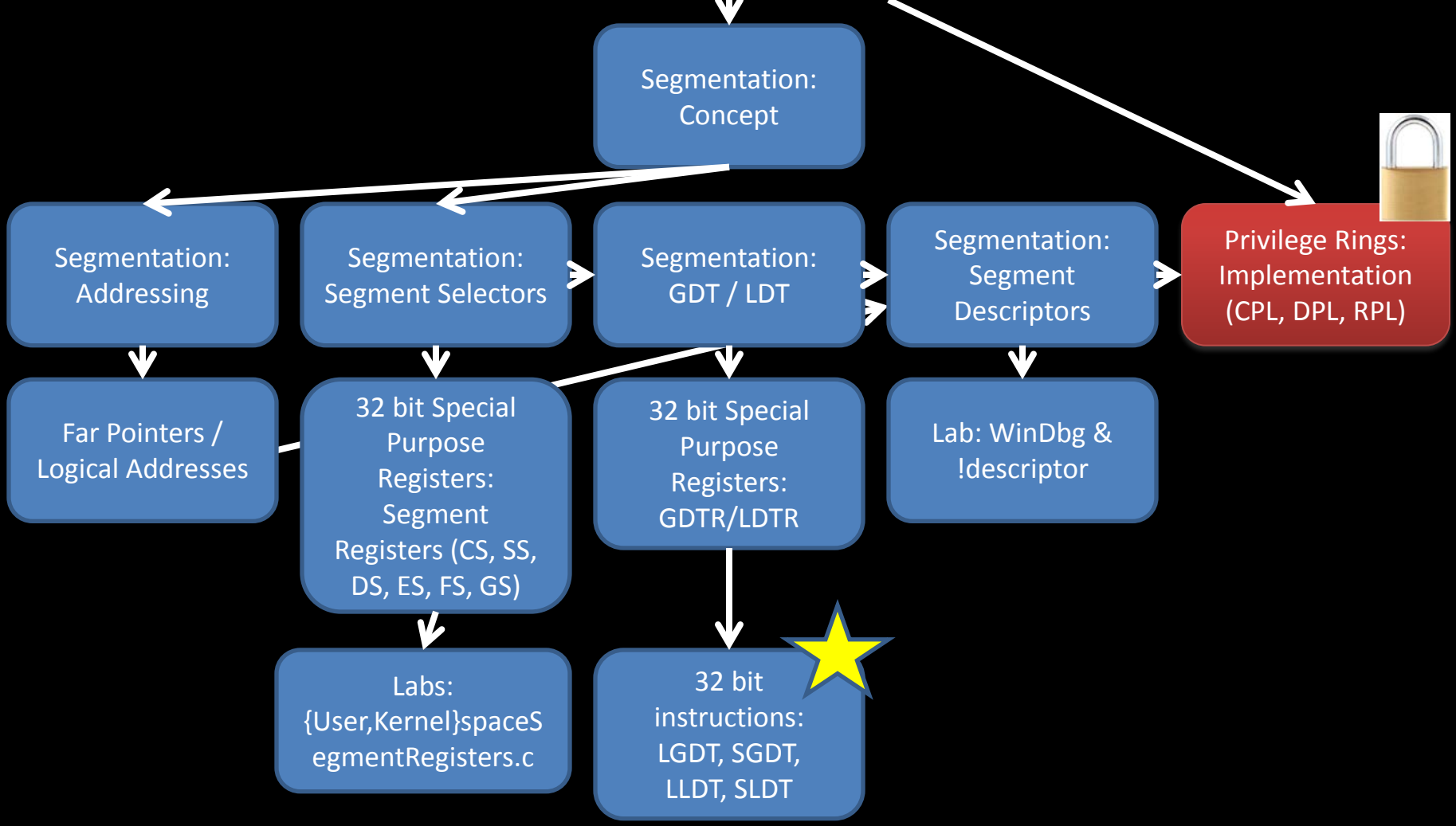


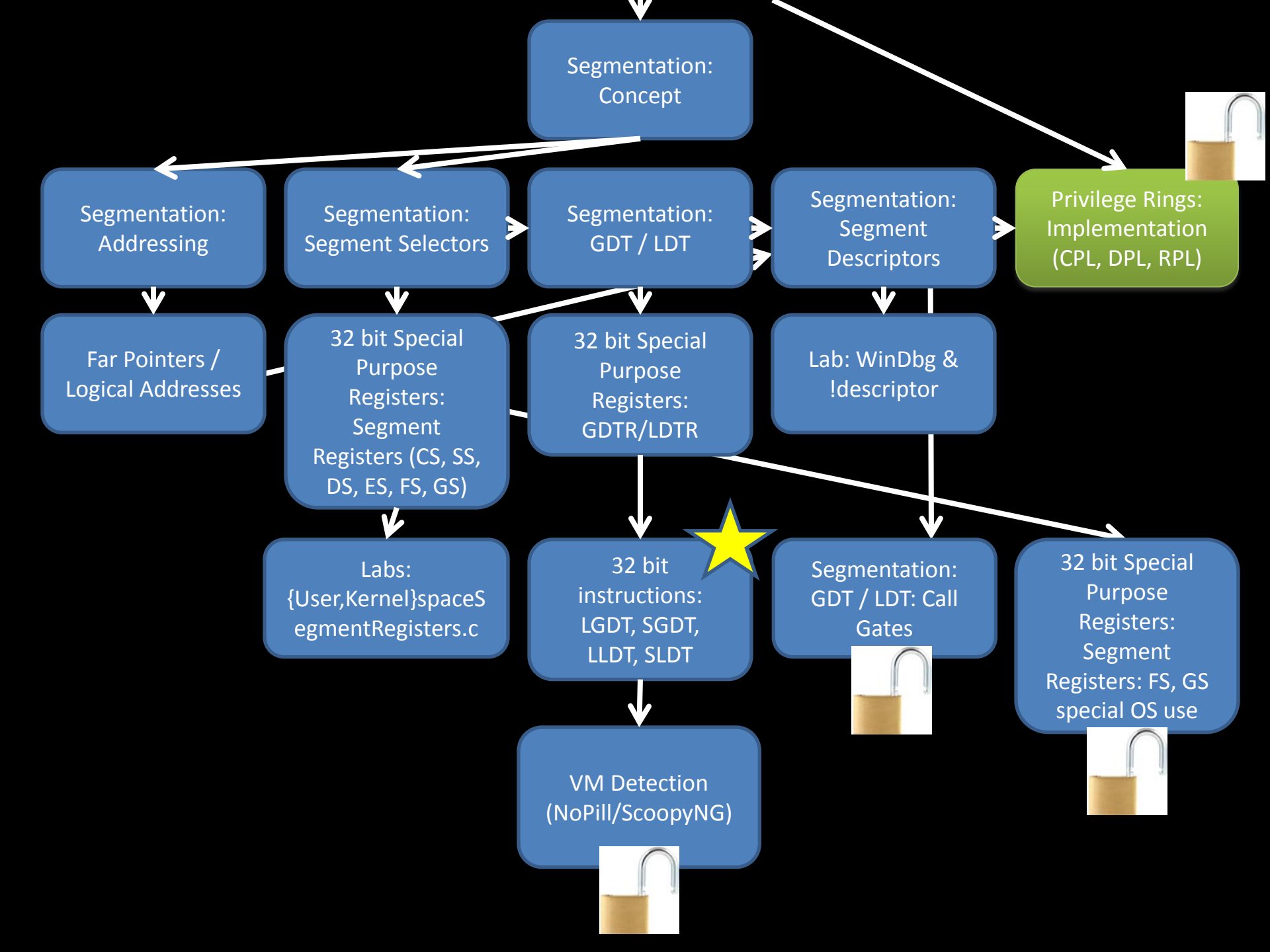


(Morning Warm Up)






Segmentation: Concept
Segmentation: Addressing
Segmentation: Segment Selectors
Segmentation: GDT / LDT
Segmentation: Segment Descriptors
Privilege Rings: Implementation (CPL, DPL, RPL)
Far Pointers / Logical Addresses
32 bit Special Purpose Registers: Segment Registers (CS, SS, DS, ES, FS, GS)
32 bit Special Purpose Registers: GDTR/LDTR
Lab: WinDbg & !descriptor
Labs: {User,Kernel}spaceSegmentRegisters.c
32 bit instructions: LGDT, SGDT, LLDT, SLDT


Segmentation: Concept
Segmentation: Addressing
Segmentation: Segment Selectors
Segmentation: GDT / LDT
Segmentation: Segment Descriptors
Privilege Rings: Implementation (CPL, DPL, RPL)
Far Pointers / Logical Addresses
32 bit Special Purpose Registers: Segment Registers (CS, SS, DS, ES, FS, GS)
32 bit Special Purpose Registers: GDTR/LDTR
Lab: WinDbg & !descriptor
Labs: {User,Kernel}spaceSegmentRegisters.c
32 bit instructions: LGDT, SGDT, LLDT, SLDT
Segmentation: GDT / LDT: Call Gates
32 bit Special Purpose Registers: Segment Registers: FS, GS special OS use
VM Detection (NoPill/ScoopyNG)

Real Maps

- ✓ Enumeration (broad & shallow) – Intro x86
- ✓ Tightly Coupled (mesh) - Intermediate x86
- ✓ Clickable – Intro Reverse Engineering

Comments on Knowledge Maps / Skill Trees

- Inspired by videogame skill trees
- Ideally, self-directed so students can pick their own path
  - But can also be used by instructors to plot out the path for a course
- Ideally, quizzable material in each capsule
  - Common feature to Khan, Coursera, EdX, etc
  - Better yet, programmatically randomized material which can be repeated indefinitely until the student maters the concept

Mapping OST classes to NICE

- Will focus on the x86 architecture/rootkits/malware analysis/exploits cluster shown before
- Will highlight some gaps in the Framework coverage that naturally appear when doing the mapping process.

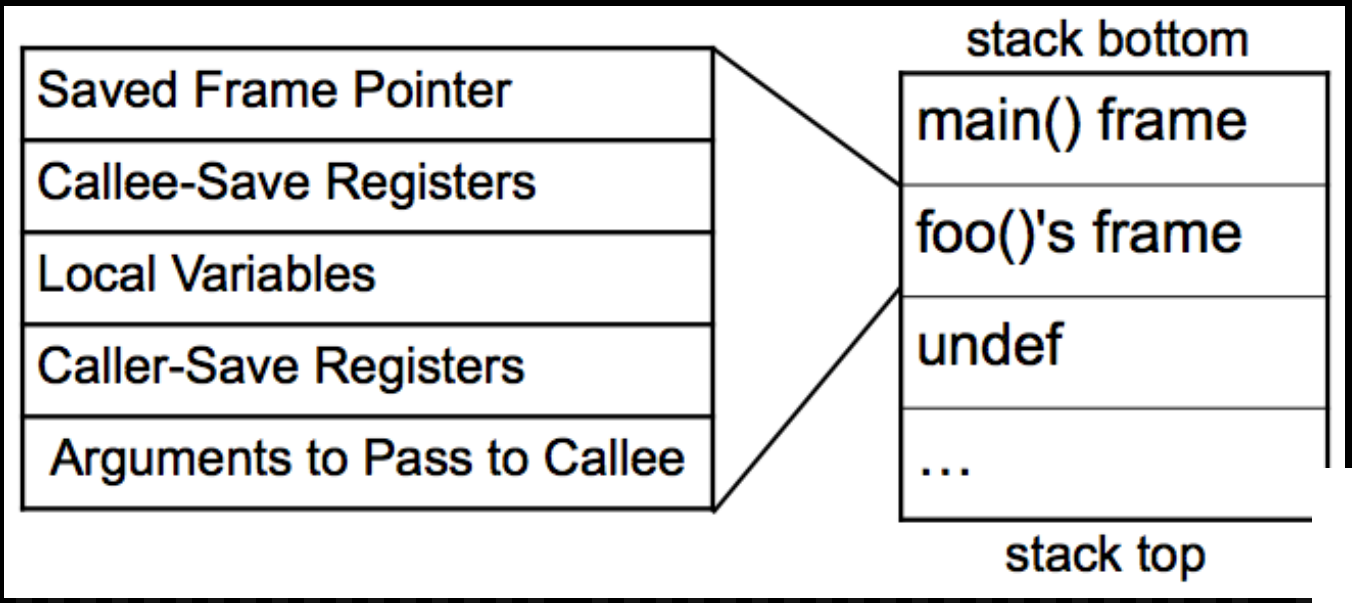
Introduction to Intel x86: Architecture, Assembly, & Applications

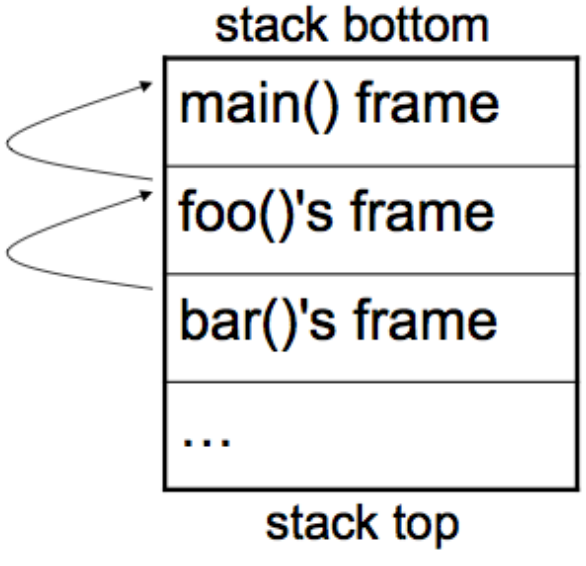
- ✓ **"20 Knowledge of complex data structures"** - too generic.
- ✓ "74 Knowledge of low-level computer languages (e.g., assembly languages)"
- ✓ **"90 Knowledge of operating systems"** - too generic
- ✓ "102 Knowledge of programming language structures and logic"
- ✓ "105 Knowledge of system and application security threats and vulnerabilities (e.g., buffer overflow, mobile code, cross-site scripting, Procedural Language/Structured Query Language [PL/SQL] and injections, race conditions, covert channel, replay, return-oriented attacks, malicious code)"
- ✓ **"113 Knowledge of server and client operating systems"** - too generic & duplicate of 90
- ✓ "116 Knowledge of software *debugging* principles"
- ✓ "117 Knowledge of software design tools, methods, and techniques"
- ✓ "168 Skill in conducting software *debugging*"
- ✓ "386 Skill in using virtual machines"
- ✓ "904 Knowledge of interpreted and compiled computer languages"
- ✓ "1088 Skill in using binary analysis tools (e.g., Hexedit, command code xxd, hexdump)"
- ✓ "1089 Knowledge of reverse engineering concepts"
- ✓ "1094 Knowledge of *debugging* procedures and tools"
- ✓ "1096 Knowledge of malware analysis tools (e.g., Oily Debug, Ida Pro)" - gdb
- ✓ "1097 Knowledge of virtual machine aware malware, debugger aware malware, and packing"
- ✓ "1115 Skill in reading Hexadecimal data"

Some thoughts right off the bat…

- "20 - Knowledge of complex data structures"
- What is a complex data structure?
  - Lets look in the eye of some beholders…

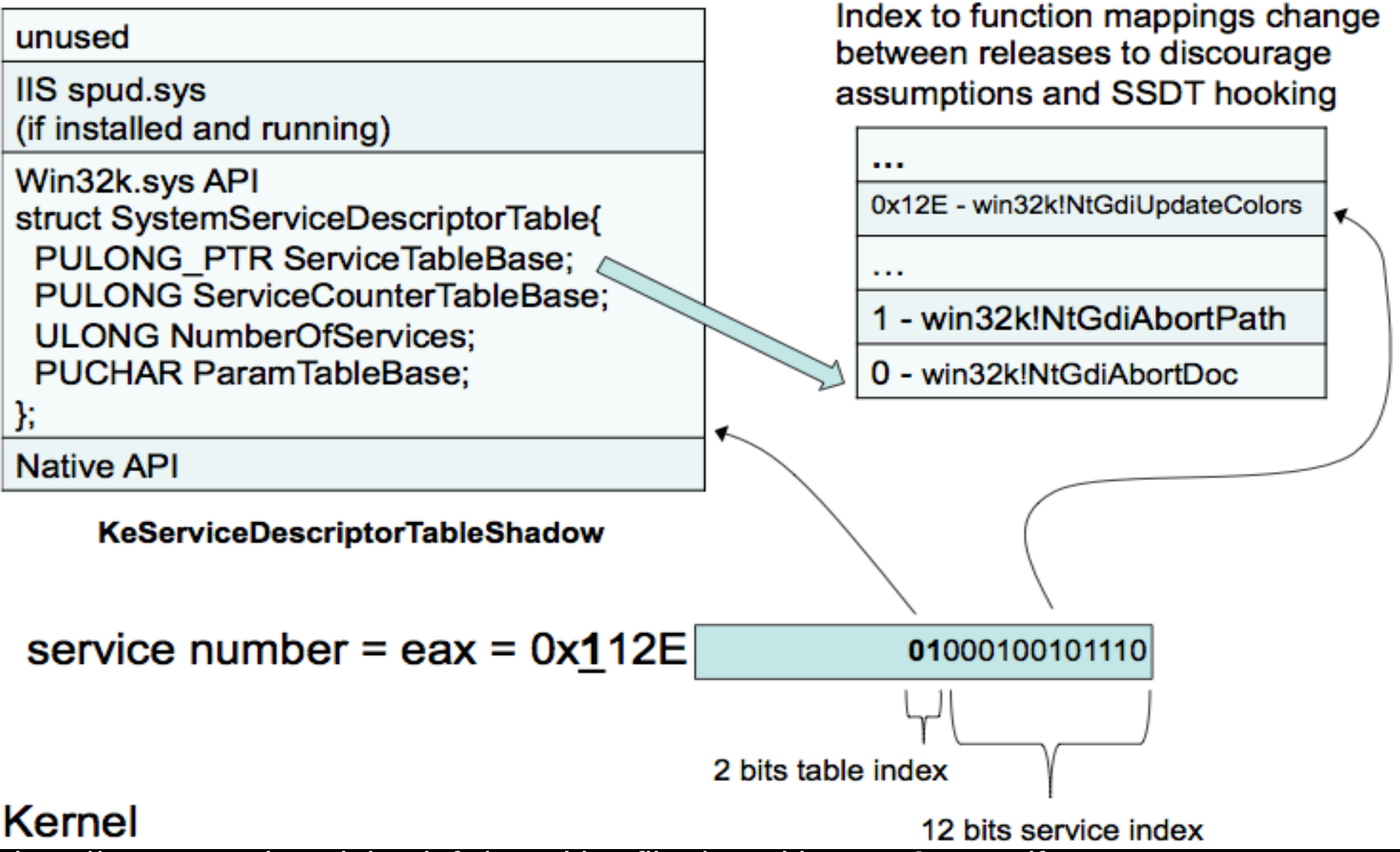
"Complex" data structure in Intro x86? (x86 stack frames)





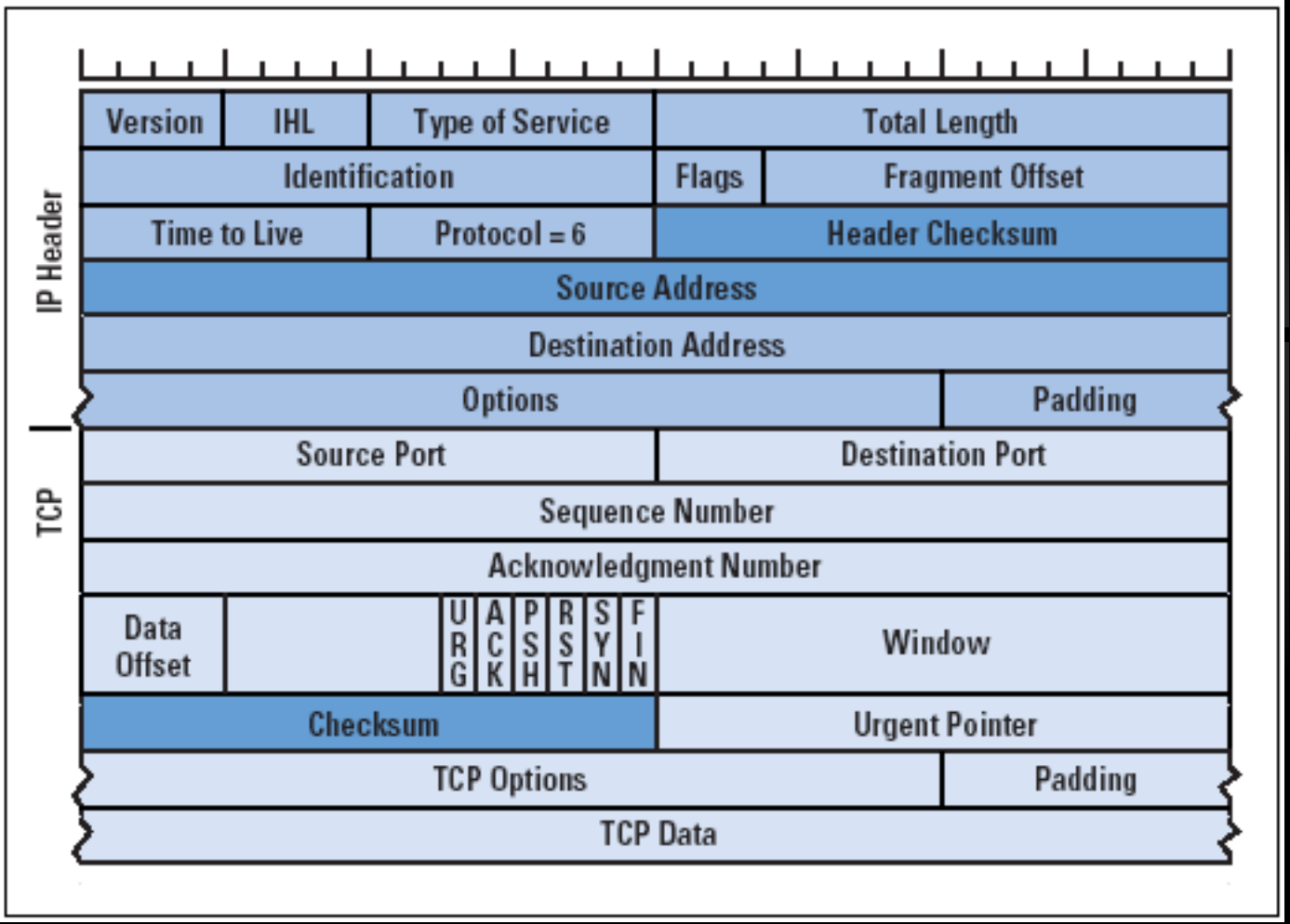
http://opensecuritytraining.info/IntroX86_files/IntroductionToIntelx86-Part1.pdf

"Complex" data structure in rookits? (Windows system call table)



"Complex" data structure in network security? (TCP packet header)

Figure 1: TCP/IP Header Fields Altered by NATs (Outgoing Packet)



http://www.cisco.com/web/about/ac123/ac147/images/ipj/ipj_7-3/anatomy_figure_1.gif

"Complex" data structure in Intermediate x86? (x86 architectural elements)

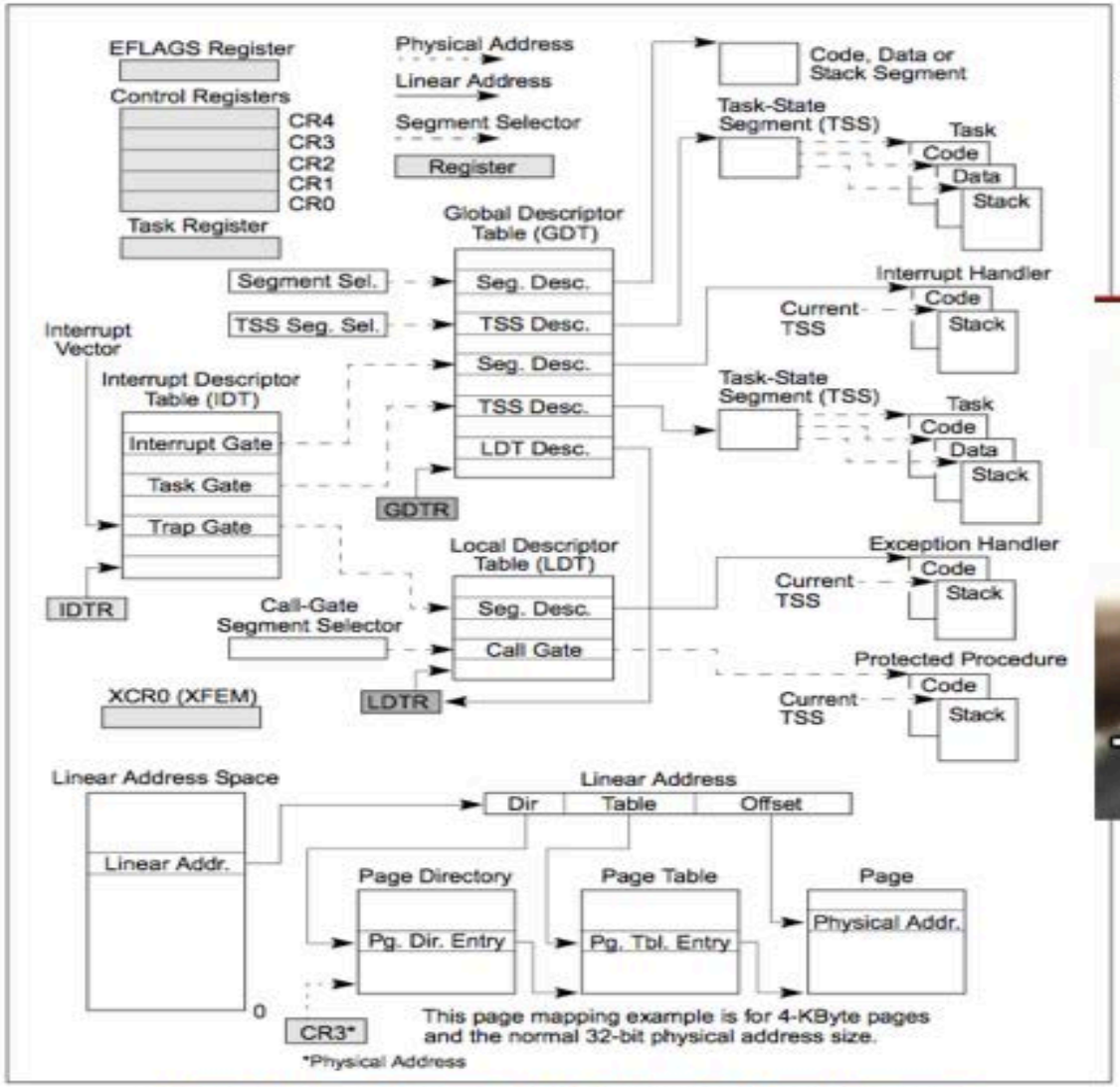
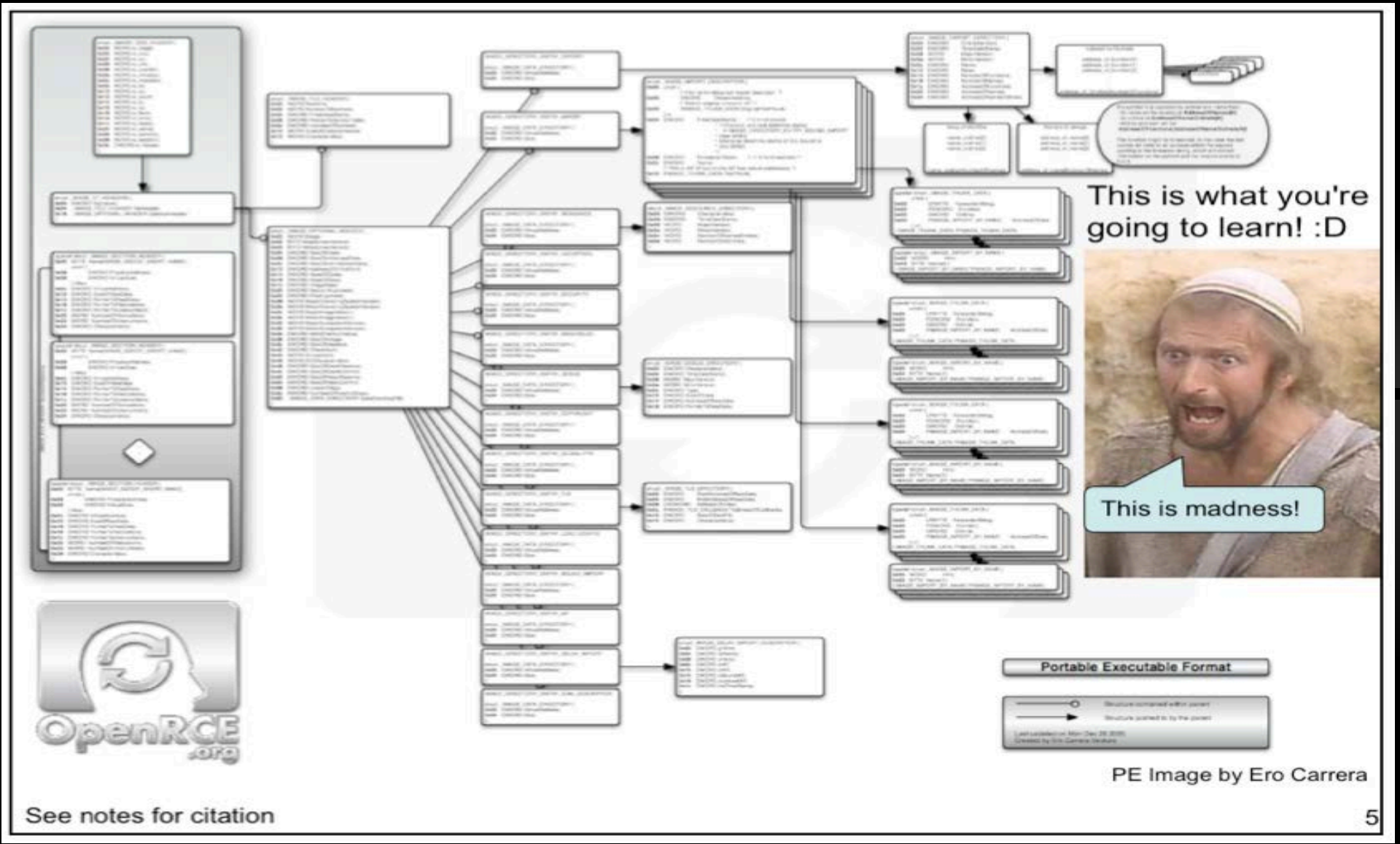


Figure 2-1. IA-32 System-Level Registers and Data Structures

That's what you're going to learn! :D

http://opensecuritytraining.info/IntermediateX86_files/IntermediateIntelx86-Part1.ppt.pdf

"Complex" data structure in Life of Binaries? (Windows PE binary format)



http://opensecuritytraining.info/LifeOfBinaries_files/2012_LifeOfBinaries2.0.pdf

"If it applies to everything, it's noteworthy for nothing."

- By specifying which *specific* "complex data structures" are relevant to the various jobs, we can help figure out which classes provide deep vs. shallow training

Some thoughts right off the bat…

- "90 – Knowledge of operating systems" & "113 - Knowledge of server and client operating systems"
- First, what's the difference?
- Second…so does that mean you know how to run programs from the command line and edit the registry and just generally be a power user?...Or does that mean you know how all the guts of the OS are programmed, and could make your own toy OS if you need to? (As good college OS classes teach.)
- Clearly we need more job-relevant specificity (and OST classes generally teach the latter, more rare info)

Some thoughts right off the bat…

- "116 - Knowledge of software *debugging* principles" & "168 - Skill in conducting software *debugging*" & "1094 Knowledge of *debugging* procedures and tools"
- Yes, debugging skills are very important (that's why we reinforce them in 11 different classes.
- But this duplication of the same information with just "Knowledge" vs. "Skill" at the beginning is common in other areas such as vulnerability assessment as well.

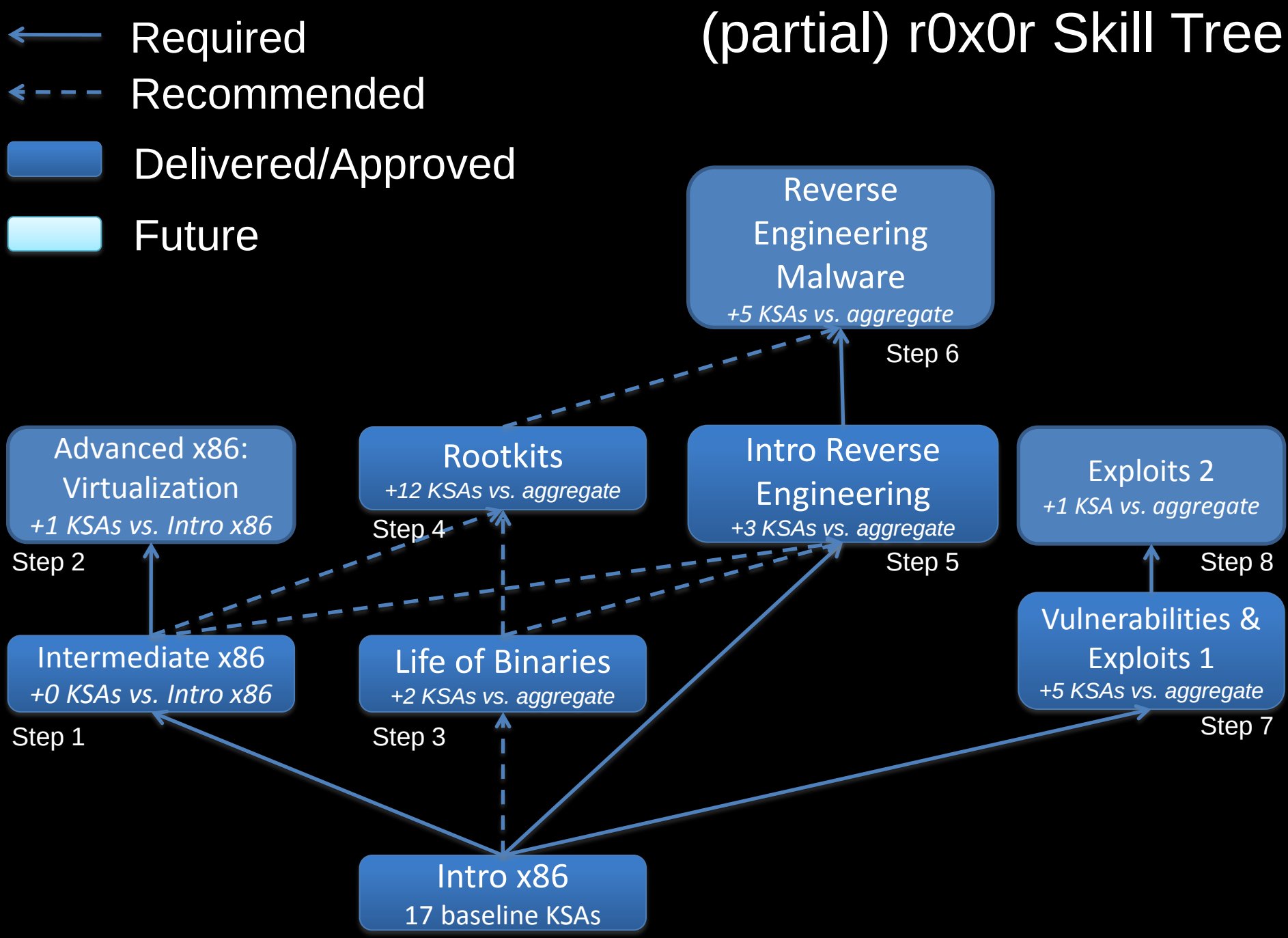
Some thoughts right off the bat…

- "386 - Skill in using virtual machines "
- Does every class that uses VMs for the labs get to claim this?!
- Because when classes self-report their KSAs for the NICCS portal they are definitely incentivized to claim as many as possible…
- So this KSA becomes nigh-meaningless

So that was just the first x86 class

- How many of those first 17 KSAs just show up again in the other classes?
- Because if each other class isn't adding a bunch of extra KSAs, what are we doing with those extra 16 days of class?

(partial) r0x0r Skill Tree



Hmm…

- Clearly if 16 days of training only gets you X more KSAs vs. 2 days, those extra classes must be pretty worthless right?
- *Or maybe some KSAs are not being captured*
- But of course also some KSAs take longer to learn and require more practice than others

A couple final observations

- The Introduction to Trusted Computing class preliminary mapping turned up only 4 KSAs – 2 generic, and 2 crypto
- A broad class like our Introduction to Vulnerability Assessment raked in 55 KSAs. Why? Seems to be in part due to overlap and duplication in the vulnerability assessment, security test & evaluation, and penetration testing categories

Conclusion 1

- OpenSecurityTraining.info classes teach or reinforce 100+ unique NICE 1.0 KSAs
- The purpose here is to share some observations based on going through the NICE mapping process.
- NICE 1.0 is a great start, but it needs more detail, especially in the malware area
  - I will be sharing many concrete recommendations with the NICE 2.0 team
- Some more research-oriented classes get short shrift (e.g. Intro Trusted Computing)
- There needs to be some way to convey broad vs. deep training classes (rather than just use KSA counts) so students can choose classes based on their needs

Conclusion 2

- Some of the KSAs that are hidden should probably be hidden. But others, like Exploitation Analysis & Threat Analysis should not. They are actually used by lots of defensive security researchers in non-classified positions. It is a disservice to the community to act like this knowledge is only ever used in a classified setting for offensive positions.
  - Luckily, people who wish to learn exactly how exploits work can find material at OpenSecurityTraining.info
  - It harms security when defenders have to treat attacks like black box voodoo magic that they have no understanding of

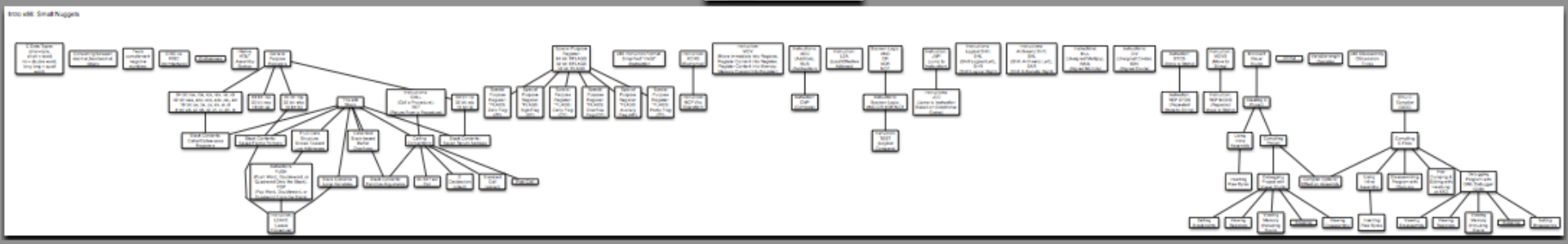
Questions?

- http://OpenSecurityTraining.info
- xkovah at gmail

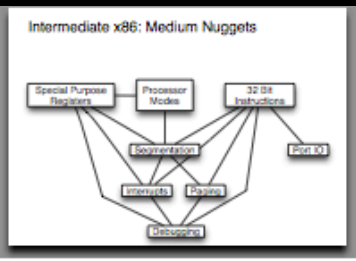
Backup

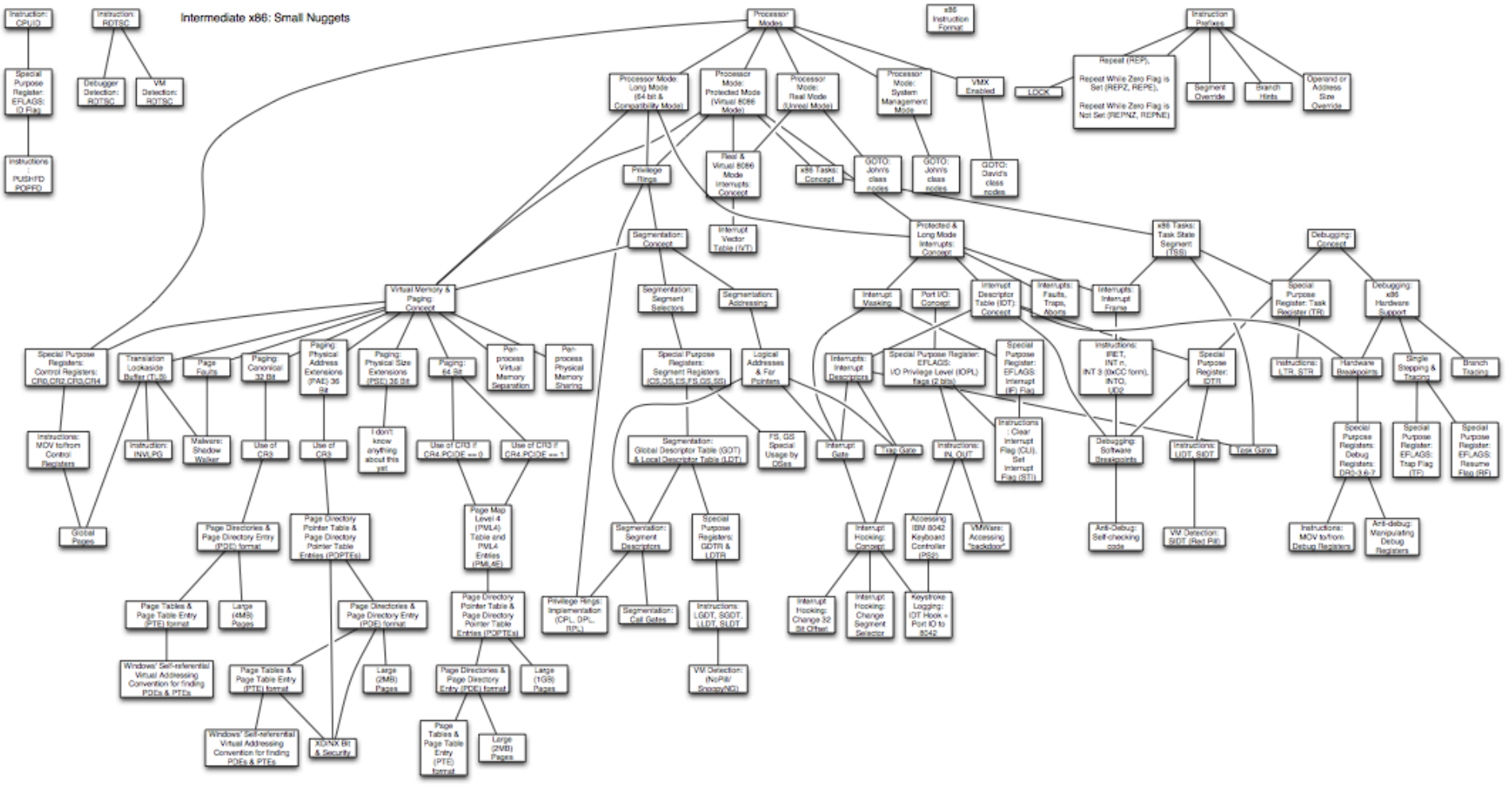
- For if the pdfs don't work

Intro x86

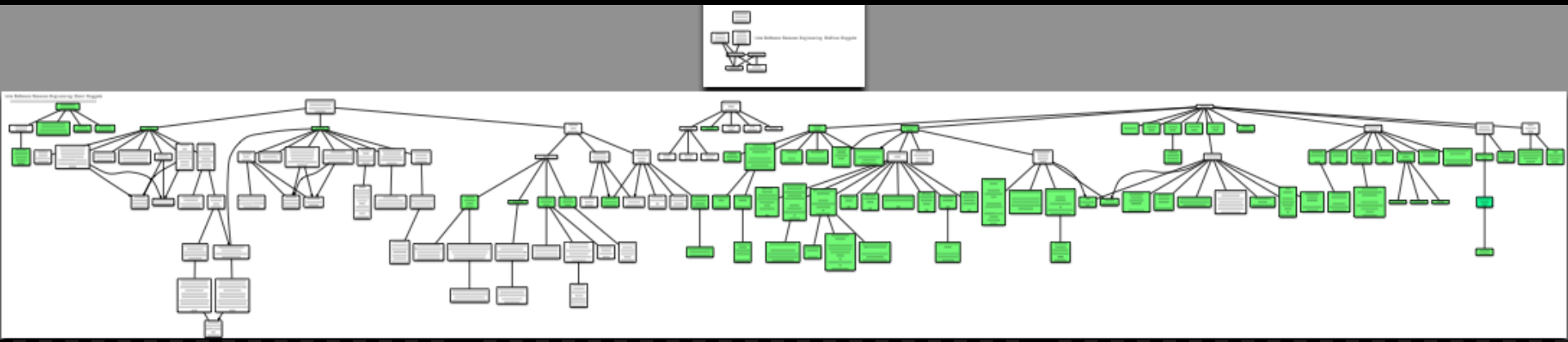


Intermediate x86

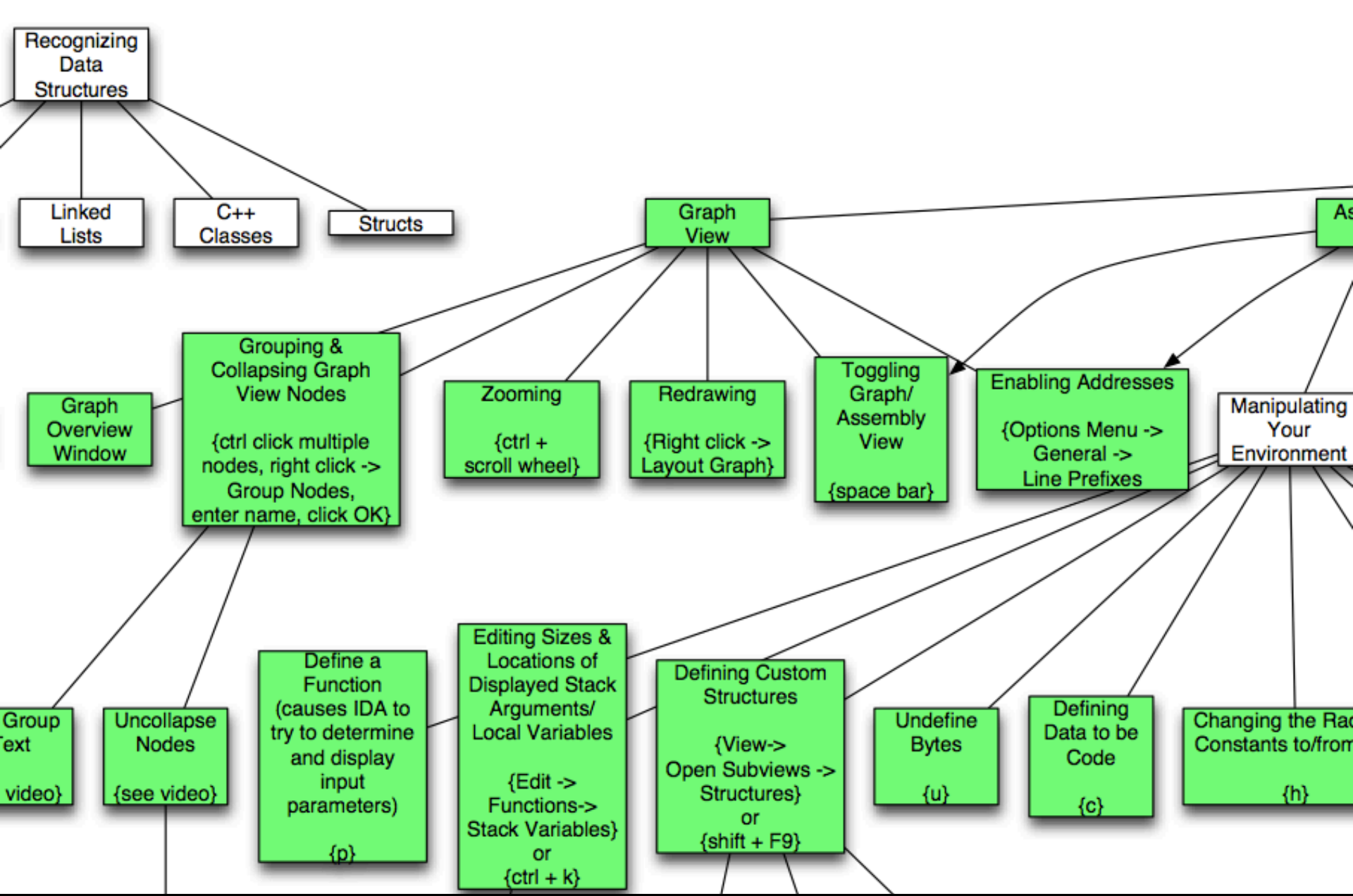


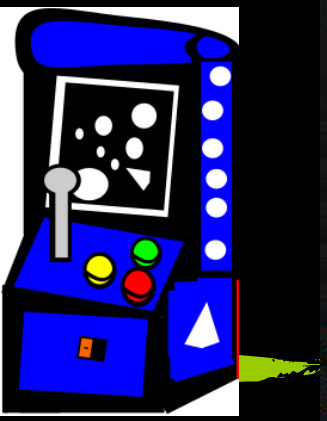


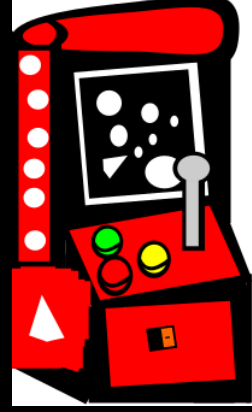
Intro RE



Intro RE (zoomed)



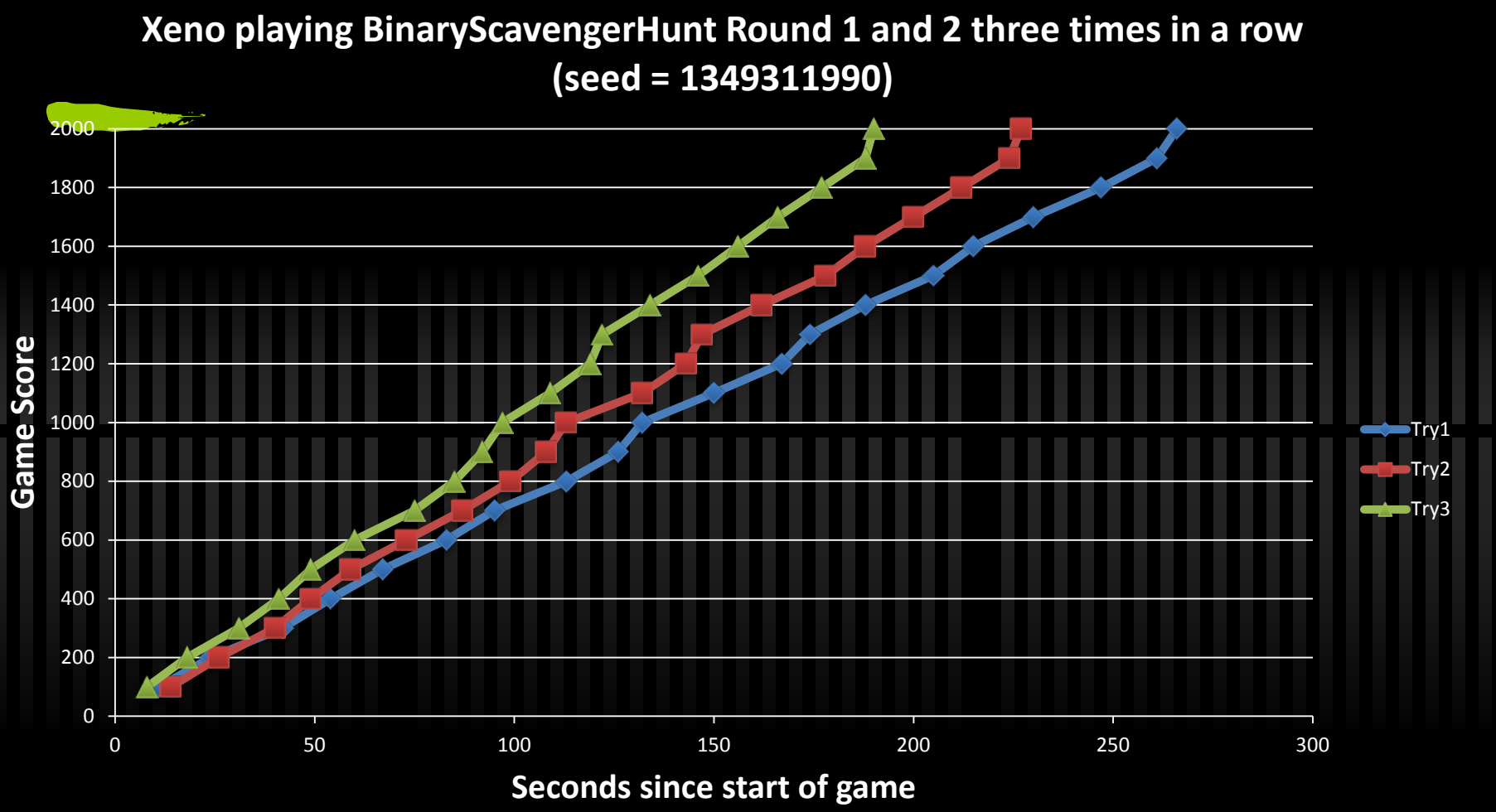




- Some learning requires grinding!
- But it can still be accelerated!

- http://code.google.com/p/roxor-arcade
  - Such timeless classics as:
  - BinDeciHex
  - The *other* ESP game
  - 1 step forward 3 steps back
  - BinaryScavengerHunt
  - May your buffer overfloweth

- Quick demo of "BinaryScavengerHunt" if I'm using my own laptop to present.


Xeno playing BinaryScavengerHunt Round 1 and 2 three times in a row
(seed = 1349311990)
Game Score
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
50
100
150
200
250
300
Seconds since start of game
Try1
Try2
Try3


Oct 19th 2012 Life of Binaries class playing Round 1 & 2 of the BinaryScavengerHunt game
Game Score
2000
1500
1000
500
0
-500
-1000
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Seconds since start of game
Student 1
Student 2
Student 3
Student 4
Student 5
Student 6
Student 7
Student 8
Student 9
Student 10
Guess who never asked any questions?