black hat USA 2022

A Fully Trained Jedi, You Are Not

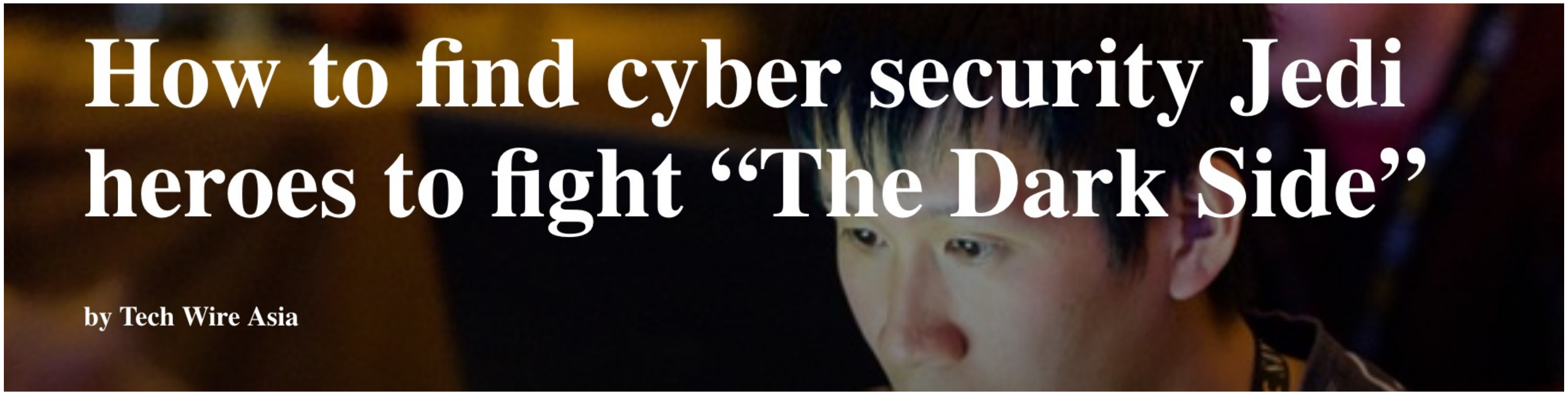
Adam Shostack

black hat USA 2022

How Many Jedi?

How Many Jedi?

We talk a lot about Jedi



How to find cyber security Jedi heroes to fight "The Dark Side"

by Tech Wire Asia

How to become a cybersecurity Jedi, Part 4: Three lessons from 'Star Wars: The Last Jedi'

It’s a Bad Goal

Expectations of heroism drive burnout

Not everyone wants to be torn from their family as a child…

… Forced to live without attachments

Even if they did, many people just don’t qualify

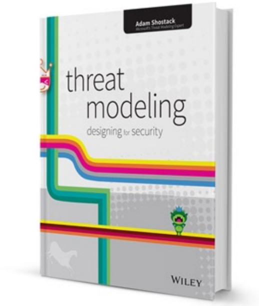
… Murdered by Sith Isn’t Good

black hat USA 2022

A Fully Trained Jedi, You Are Not

Adam Shostack

About Adam Shostack



Adam Shostack

threat modeling

designing for security

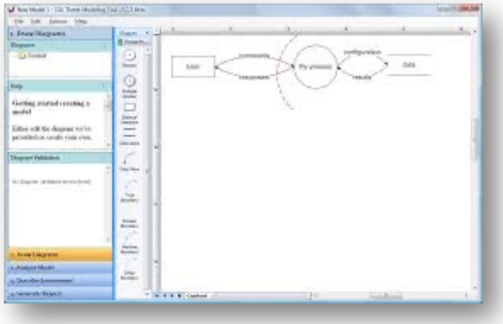
WILEY

SHOSTACK + ASSOCIATES

3 Tampering

An attacker can take advantage of your custom key exchange or integrity control which you built instead of using standard crypto

black hat



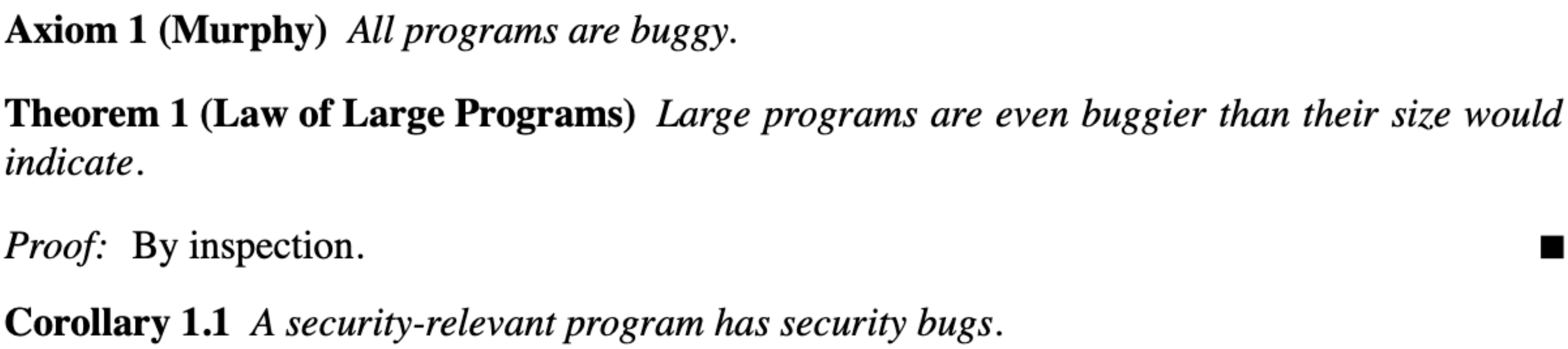
W

IriusRisk

Agenda

- The problem starts with software
- “Shifting left” isn’t working
- Reasonable Expectations
  - Bloom
  - Chunking
  - Frames

We’ve known for a while …



Axiom 1 (Murphy) *All programs are buggy.*

Theorem 1 (Law of Large Programs) *Large programs are even buggier than their size would indicate.*

Proof: By inspection. ■

Corollary 1.1 *A security-relevant program has security bugs.*

— *Firewalls and Internet Security*
(Cheswick and Bellovin, 1994)

Where do security issues come from?

Developers

Where do security issues come from?

Software engineers

Developers introduce many problems

- Code with security bugs + flaws
- Missing security features
- Unusable security features

Software

Application security	Operational security
Dev	Production

Software

Application security

Operational security

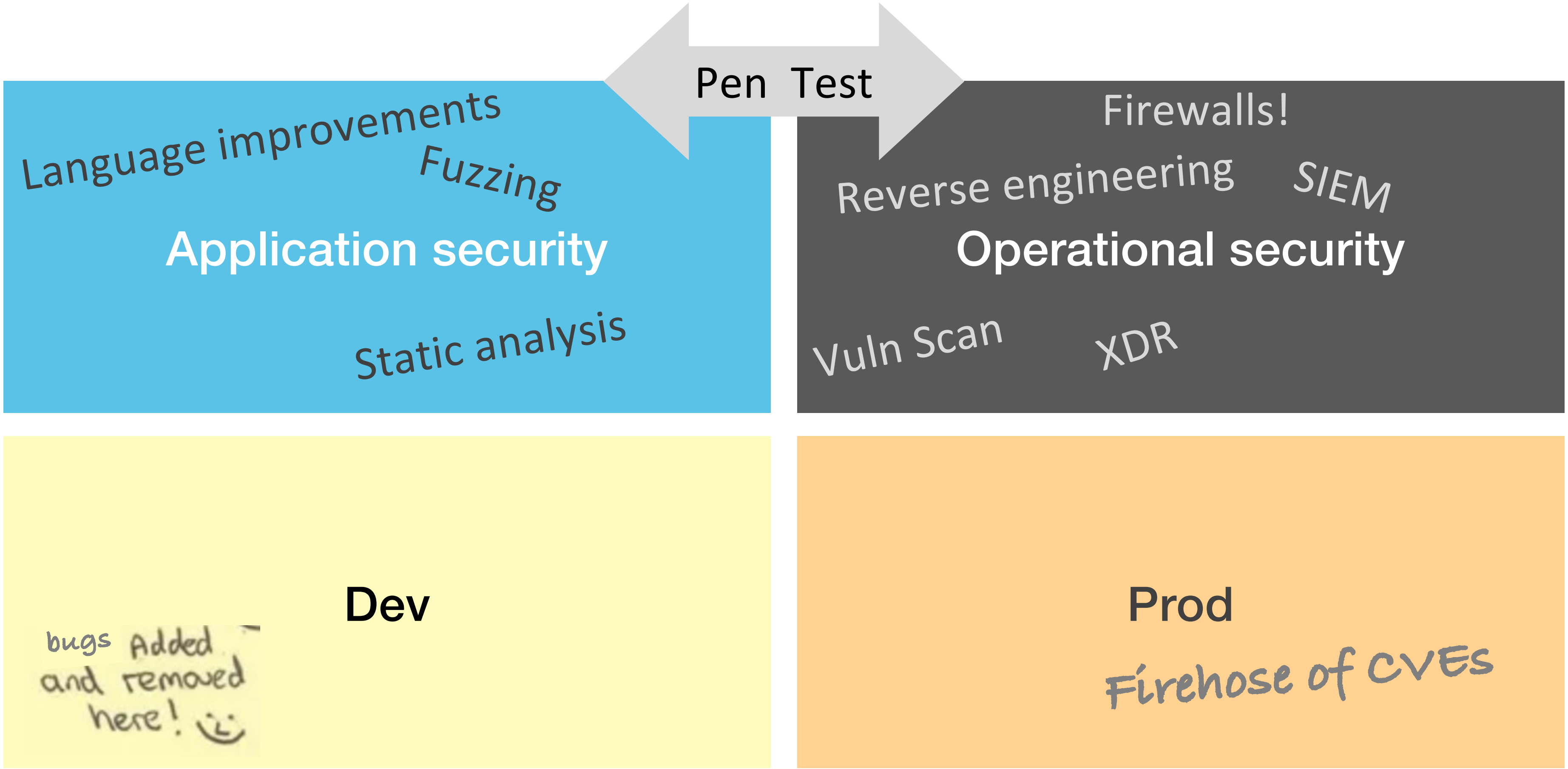
Dev

bugs Added and removed here! :)

Prod

Firehose of CVEs

Software



Pen Test

Language improvements

Fuzzing

Application security

Static analysis

Firewalls!

Reverse engineering

SIEM

Operational security

Vuln Scan

XDR

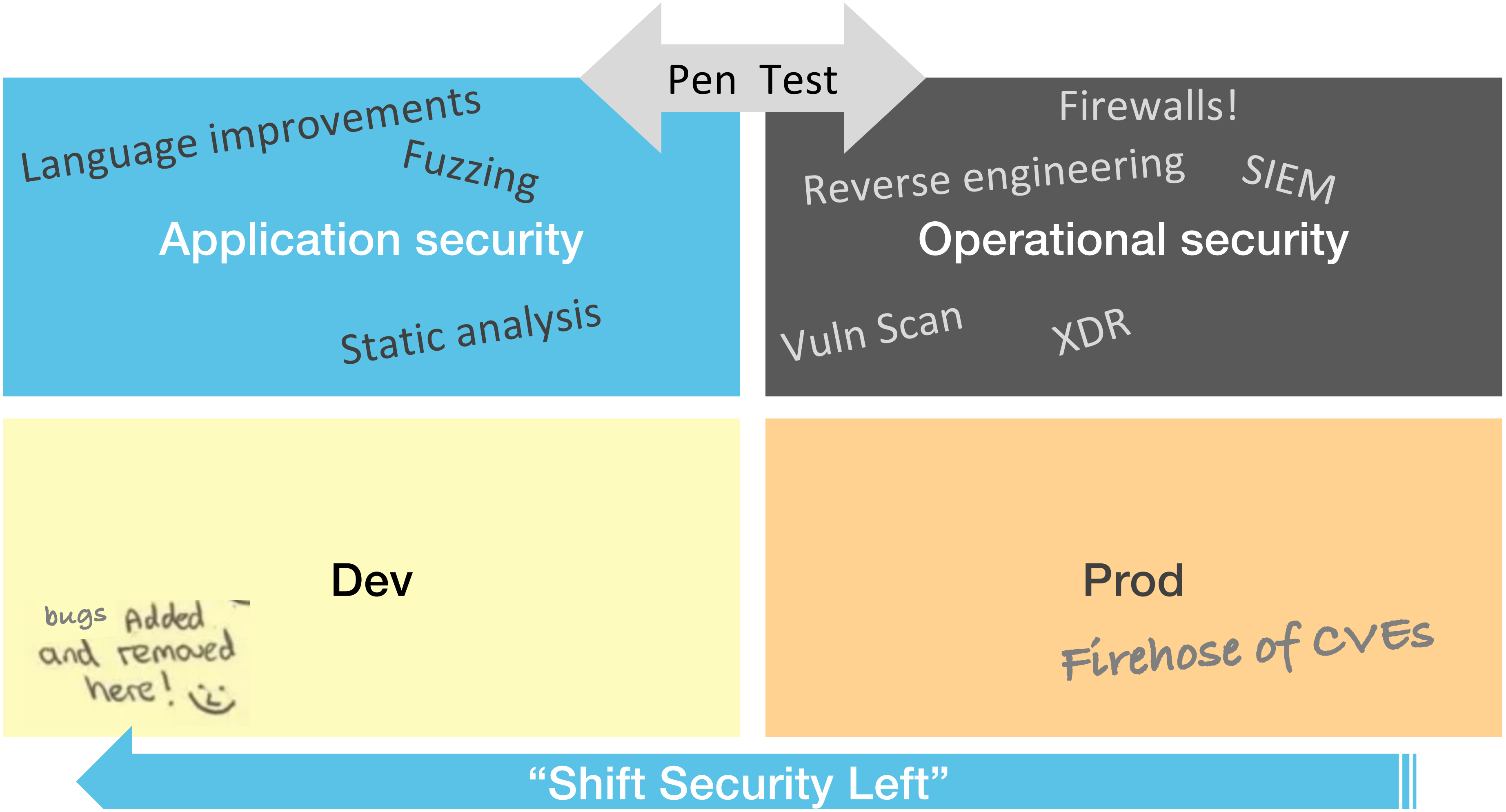
Dev

bugs Added and removed here!

Prod

Firehose of CVEs

Software



Pen Test

Language improvements

Fuzzing

Application security

Static analysis

Firewalls!

Reverse engineering

SIEM

Operational security

Vuln Scan

XDR

Dev

bugs Added and removed here! ☺

Prod

Firehose of CVEs

“Shift Security Left”

“Shift Left”

- Build security in
  - Changes to how we/they design, develop, deploy
  - Requires new skills
  - Less pen testing
    - More software engineering
- Growing popularity

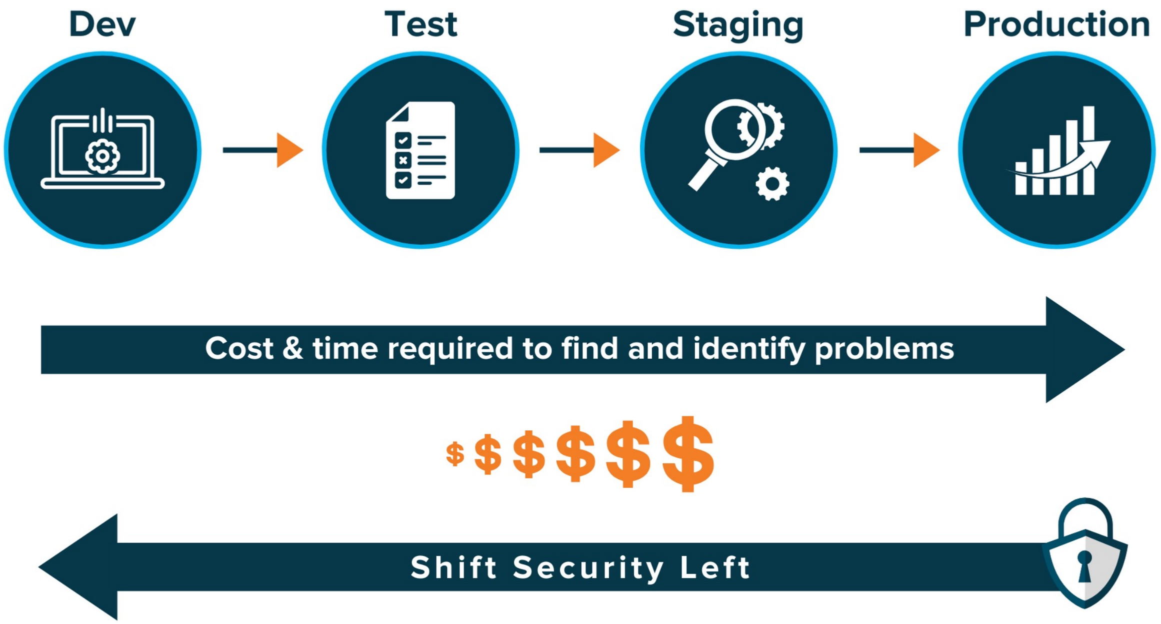


Image: Klogix

Shifting Left?

Shift left implies: Change the development process

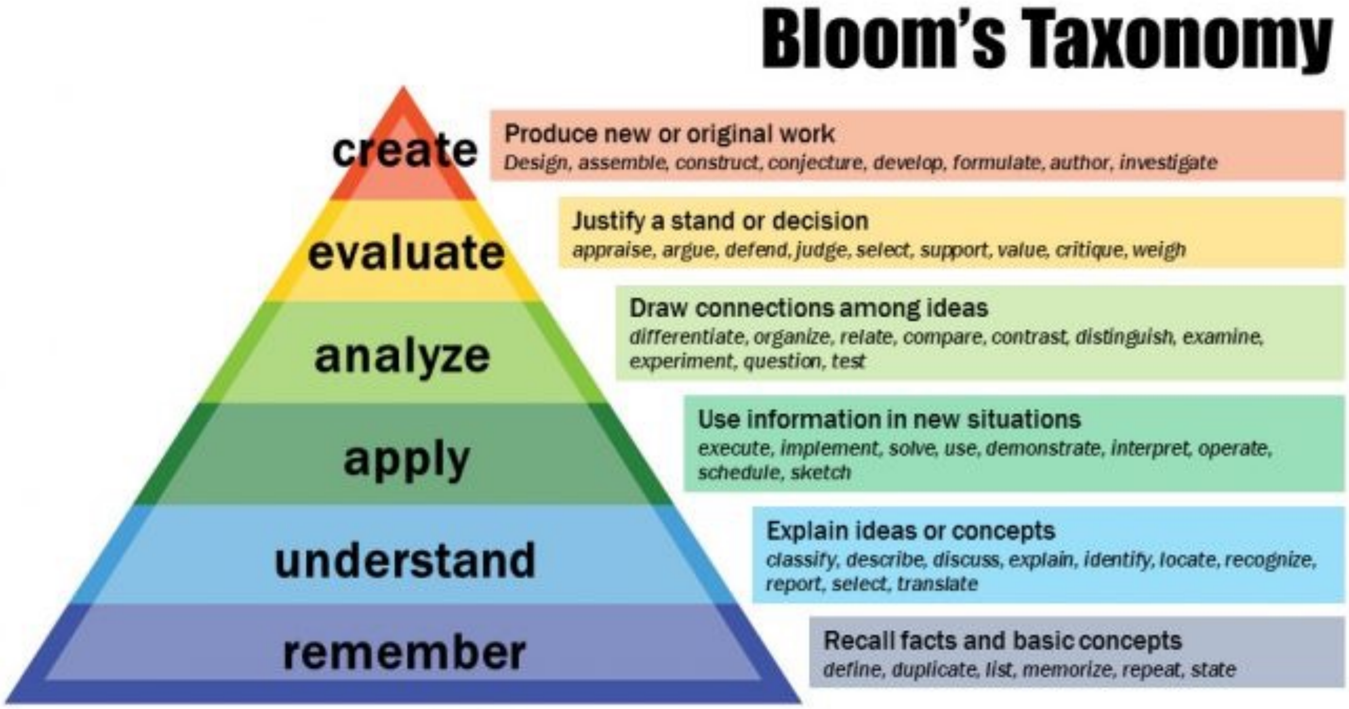
- Demands clear responsibilities
- What *exactly* is changing?
    - Deliverables
    - Tasks
    - Skills
- Risk: Are we doing this to please appsec?

Clarify

Who delivers what to whom?

How?

One tool - Bloom’s Taxonomy



Bloom’s Taxonomy

create — Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate

evaluate — Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh

analyze — Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test

apply — Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch

understand — Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate

remember — Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

Vanderbilt University Center for Teaching

- Fundamental tool in learning
- Goals + evaluations

Bloom’s Taxonomy: remember

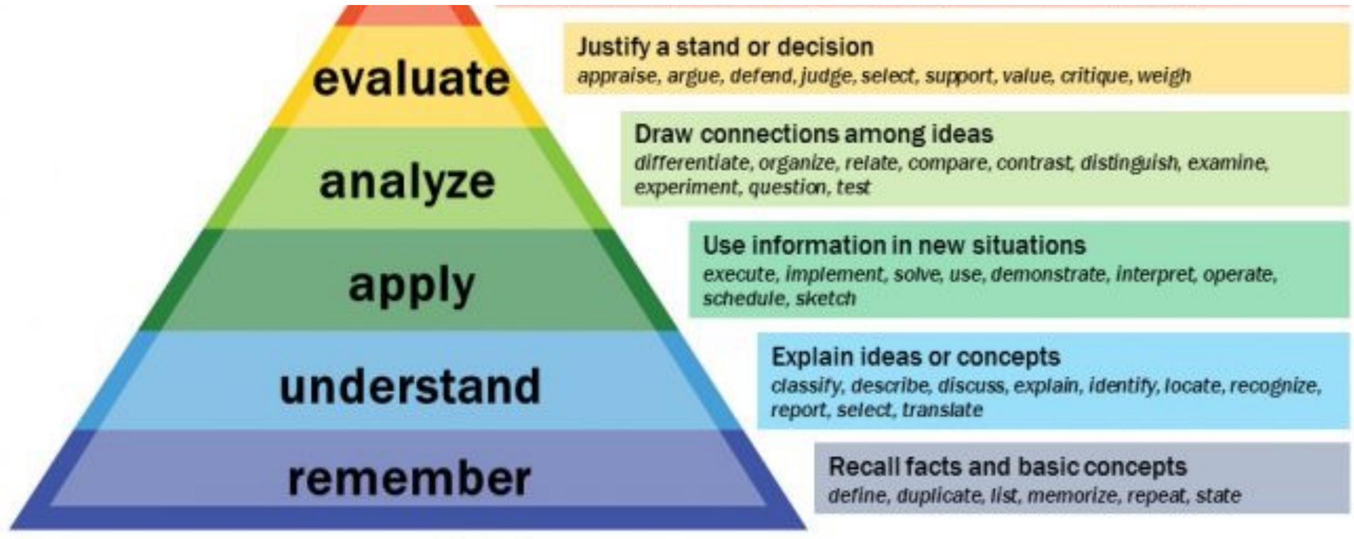
- Recall facts and basic concepts
  - Define, duplicate, list, repeat
- “Remember data sent over a network can be read by anyone”

remember
Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

Vanderbilt University Center for Teaching

Bloom's Taxonomy: Evaluate

- Justify a stand or decision
  - Argue, defend, judge, select, critique, weigh
- Does encryption protect against that threat?



Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh

Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test

Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch

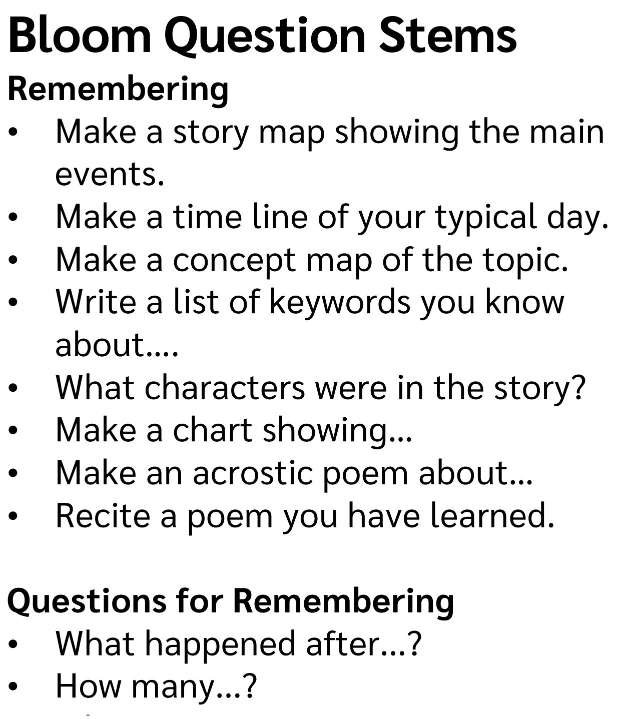
Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate

Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

Vanderbilt University Center for Teaching

Tools help us use Bloom to define skills + knowledge

- This slide’s learning goal: remember there are lots of tools to help



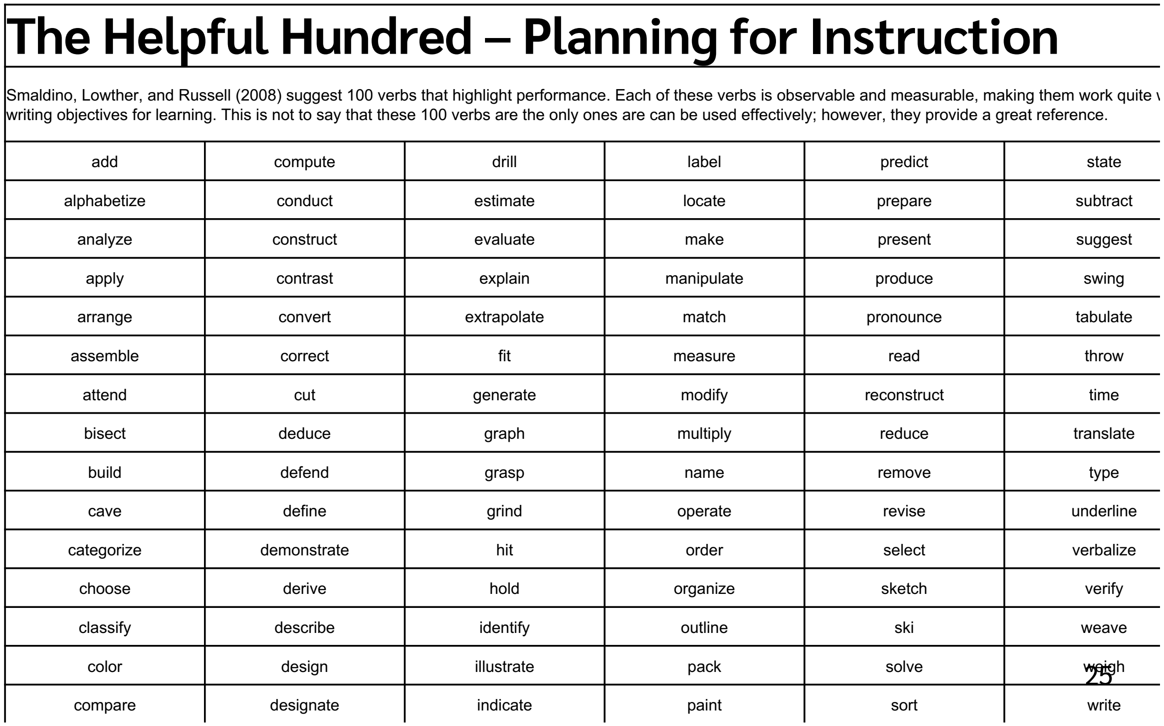
Bloom Question Stems

Remembering

- Make a story map showing the main events.
- Make a time line of your typical day.
- Make a concept map of the topic.
- Write a list of keywords you know about….
- What characters were in the story?
- Make a chart showing…
- Make an acrostic poem about…
- Recite a poem you have learned.

Questions for Remembering

- What happened after…?
- How many…?

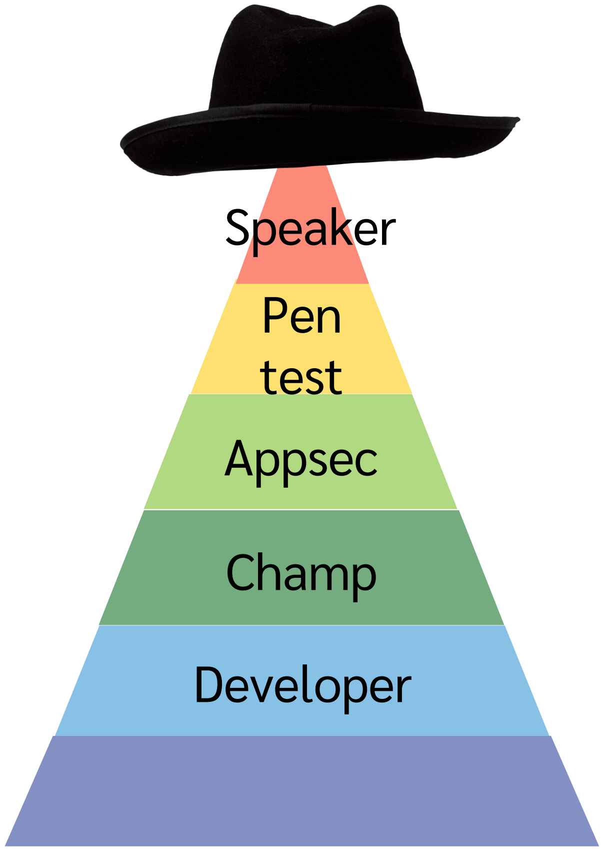


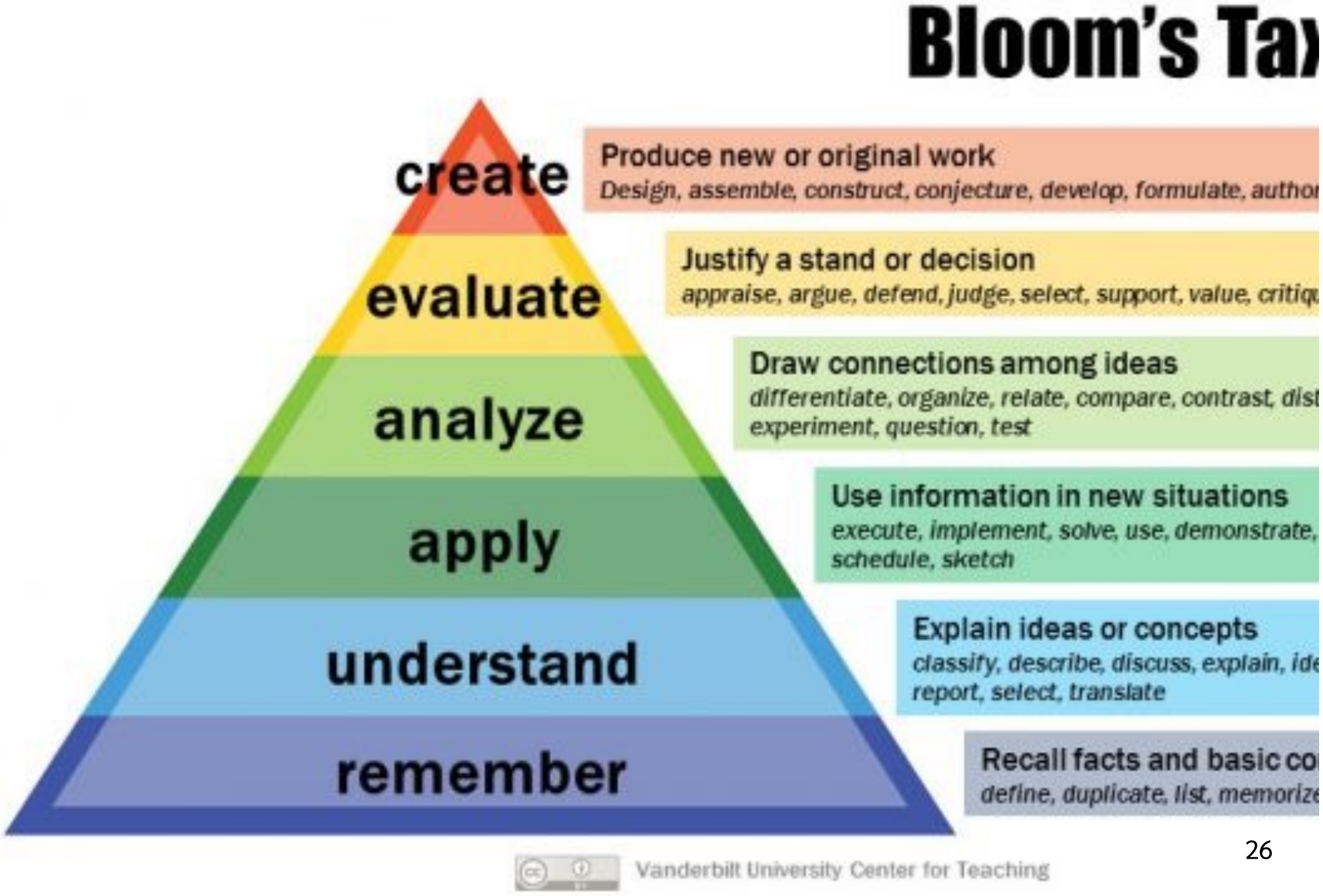
The Helpful Hundred – Planning for Instruction

Smaldino, Lowther, and Russell (2008) suggest 100 verbs that highlight performance. Each of these verbs is observable and measurable, making them work quite v
writing objectives for learning. This is not to say that these 100 verbs are the only ones are can be used effectively; however, they provide a great reference.

add	compute	drill	label	predict	state
alphabetize	conduct	estimate	locate	prepare	subtract
analyze	construct	evaluate	make	present	suggest
apply	contrast	explain	manipulate	produce	swing
arrange	convert	extrapolate	match	pronounce	tabulate
assemble	correct	fit	measure	read	throw
attend	cut	generate	modify	reconstruct	time
bisect	deduce	graph	multiply	reduce	translate
build	defend	grasp	name	remove	type
cave	define	grind	operate	revise	underline
categorize	demonstrate	hit	order	select	verbalize
choose	derive	hold	organize	sketch	verify
classify	describe	identify	outline	ski	weave
color	design	illustrate	pack	solve	weigh
compare	designate	indicate	paint	sort	write

What security work do we ask of different people?



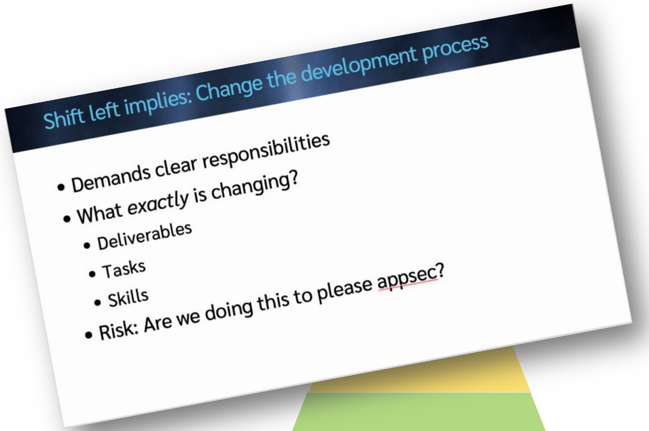

Bloom's Ta
create
Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author
evaluate
Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critiq
analyze
Draw connections among ideas
differentiate, organize, relate, compare, contrast, dist
experiment, question, test
apply
Use information in new situations
execute, implement, solve, use, demonstrate,
schedule, sketch
understand
Explain ideas or concepts
classify, describe, discuss, explain, ide
report, select, translate
remember
Recall facts and basic co
define, duplicate, list, memorize

Vanderbilt University Center for Teaching

But instead...we teach like this?

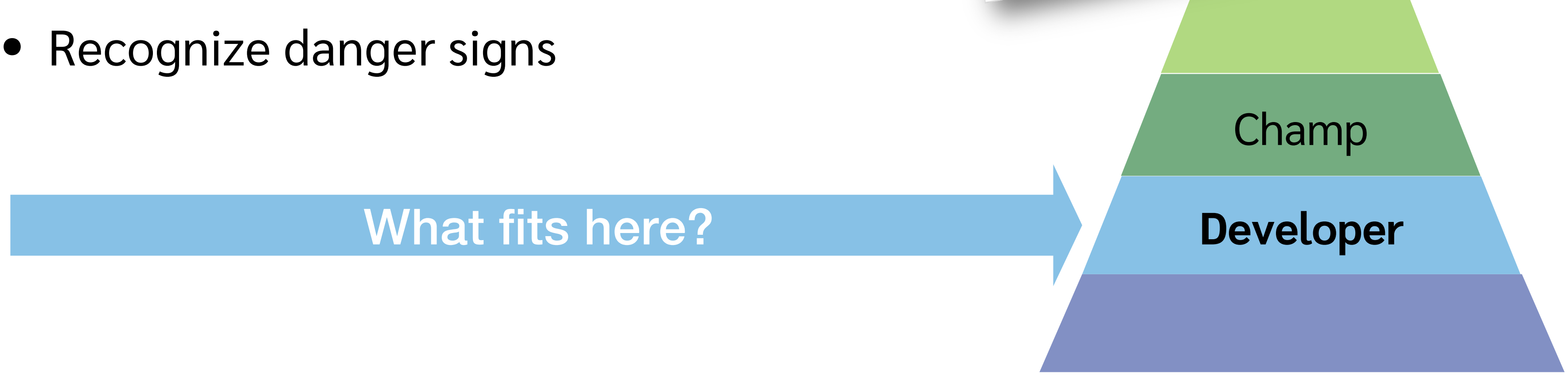
Criteria + constraints

- Align to job, aspirations
- Within reasonable training time
- Goals
  - Help people find, follow paved roads
  - Recognize danger signs



Shift left implies: Change the development process

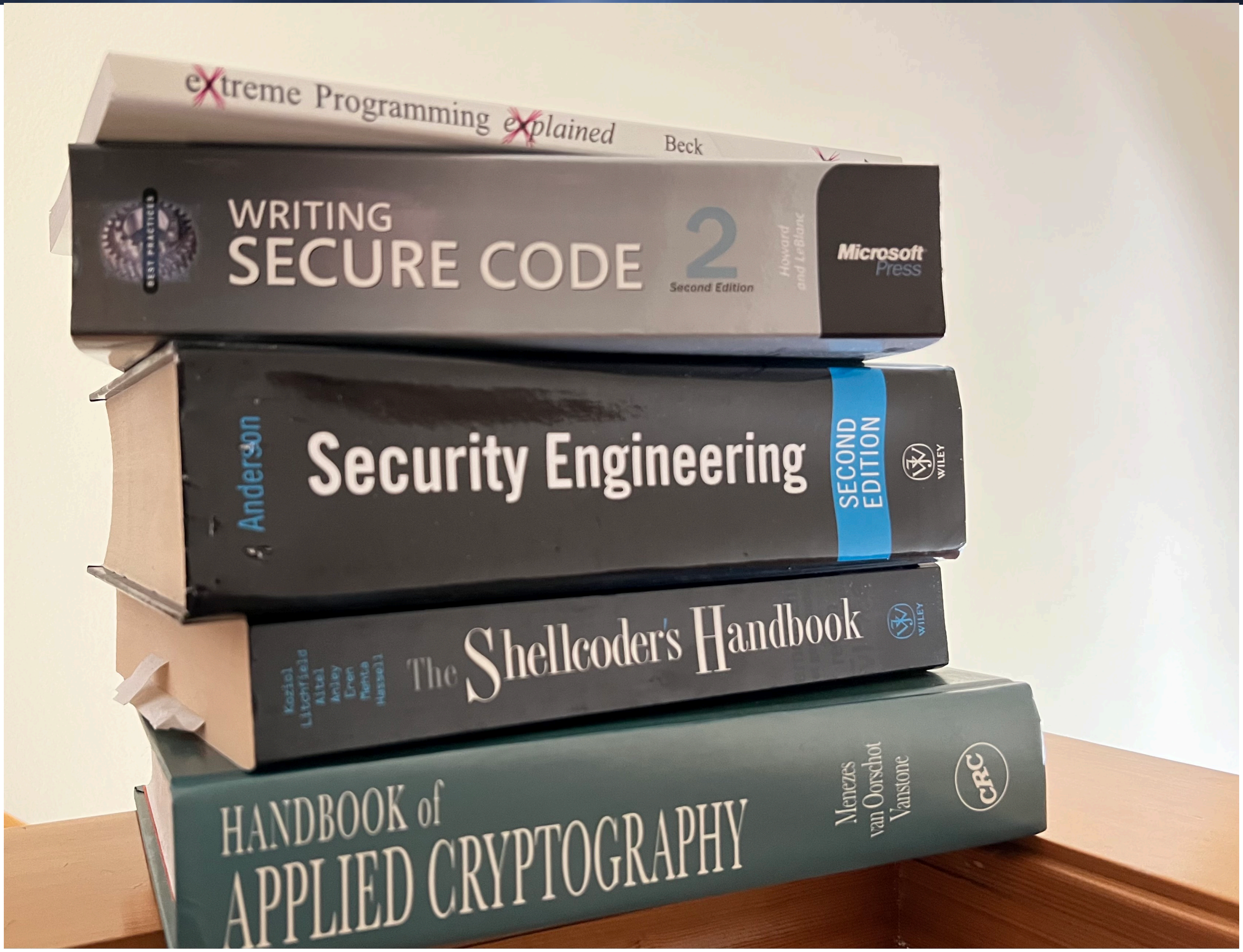
- Demands clear responsibilities
- What *exactly* is changing?
  - Deliverables
  - Tasks
  - Skills
- Risk: Are we doing this to please appsec?



What fits here?

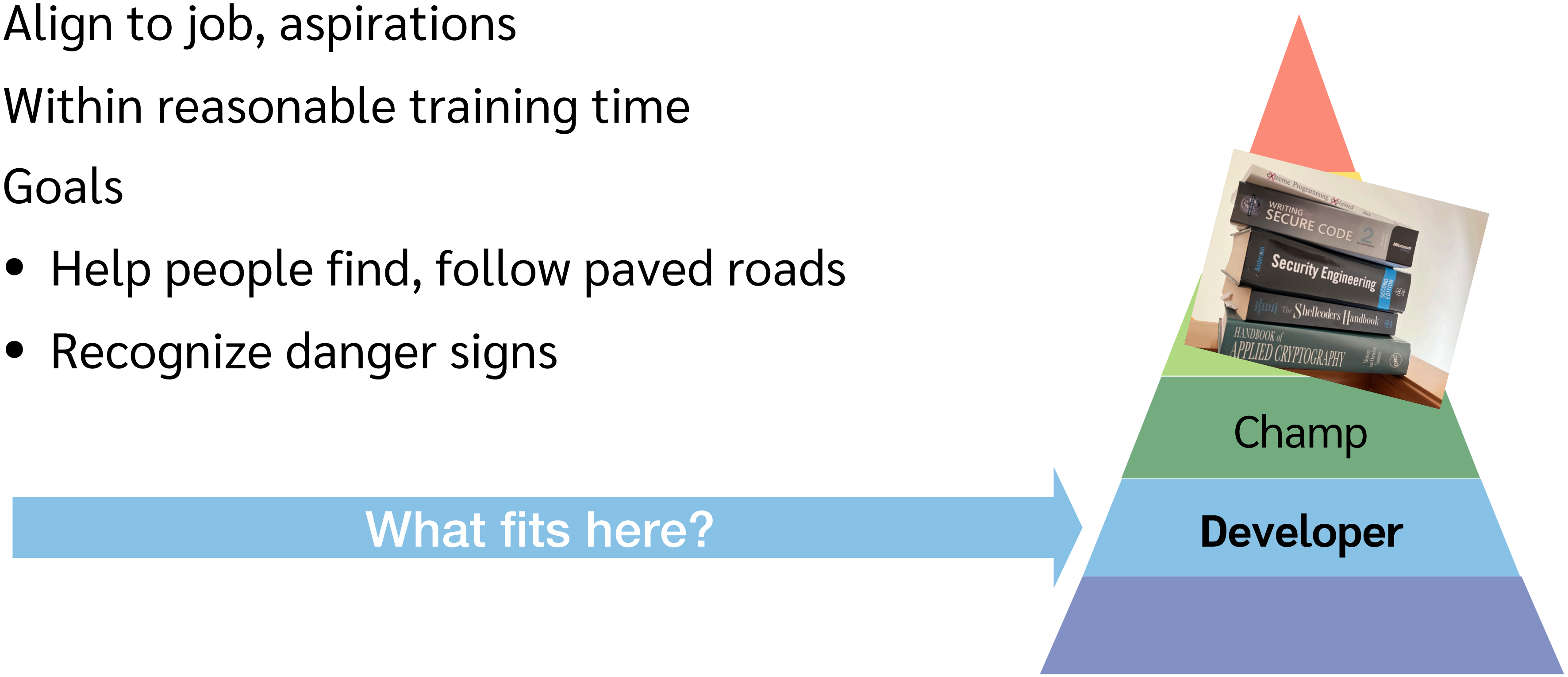
Champ

Developer


eXtreme Programming eXplained
Beck
BEST PRACTICES
WRITING SECURE CODE
2
Second Edition
Howard and LeBlanc
Microsoft Press
Anderson
Security Engineering
SECOND EDITION
WILEY
Koziol Litchfield Aitel Anley Eren Mehta Hassell
The Shellcoder's Handbook
WILEY
HANDBOOK of APPLIED CRYPTOGRAPHY
Menezes van Oorschot Vanstone
CRC

Criteria + constraints

- Align to job, aspirations
- Within reasonable training time
- Goals
  - Help people find, follow paved roads
  - Recognize danger signs



Chunking is crucial

- Our brains are really, really good at pattern recognition
  - Dealing with information in “chunks”
- Short term memory is 7 +/- 2 chunks
- 1,1,2,3,5,8,13,34,55…
- If we don’t define the chunks, our students will
  - (They may anyway!)

Categories and frames

- Exploit techniques?
- Threat actors?
- Compliance?
- Cyberwar?
- Top ten?
- Threats?

“What can go wrong” focuses our attention on threats

“What can go wrong?” is a powerful framing question

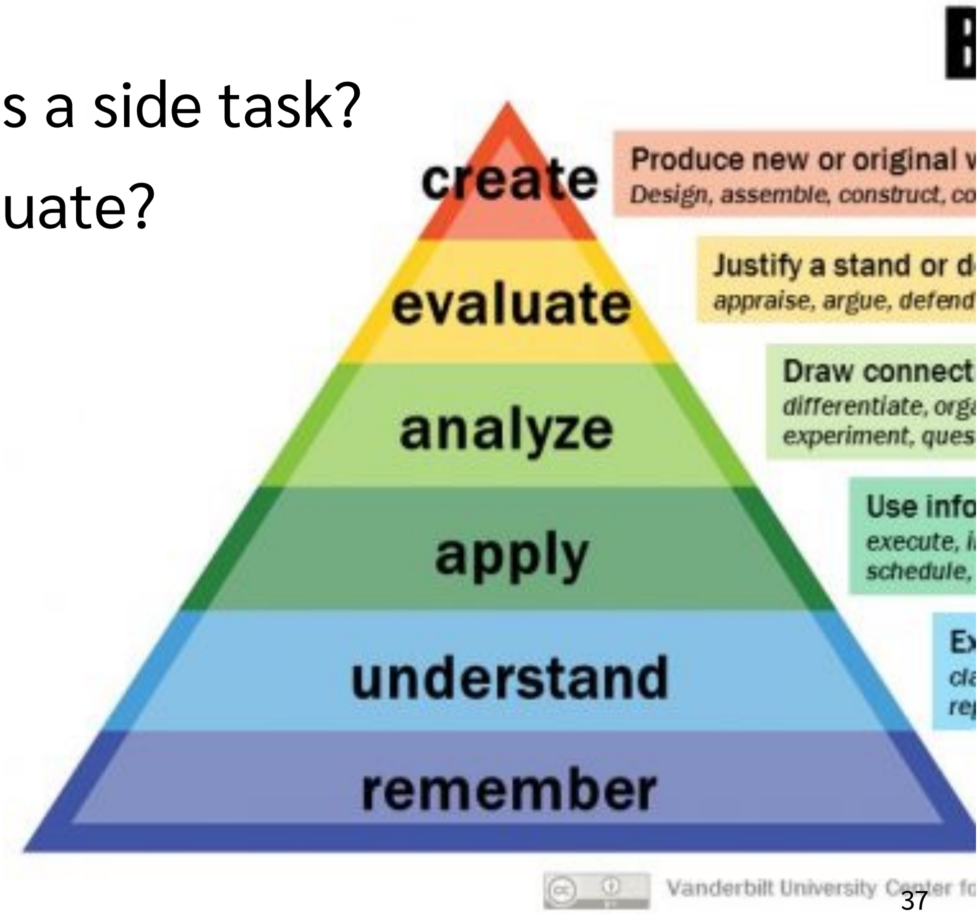
- Everyone has an answer — if you ask and encourage
    - Across industries, technical skill, execs
- Variants
    - “What keeps you up at night?”
    - “How would you attack this”
    - Lots of fast, cheap, good approaches

“What can go wrong” is an umbrella

- Open ended is easier to answer, but answers vary a lot
- Structures
  - Finance execs … ORX
  - Security … OWASP top ten
  - FDA …. inability to update
  - Compliance… see my Threat Modeling Compliance (BHAsia ’21)
- Flaws, not just bugs

What’s the single best toolset?

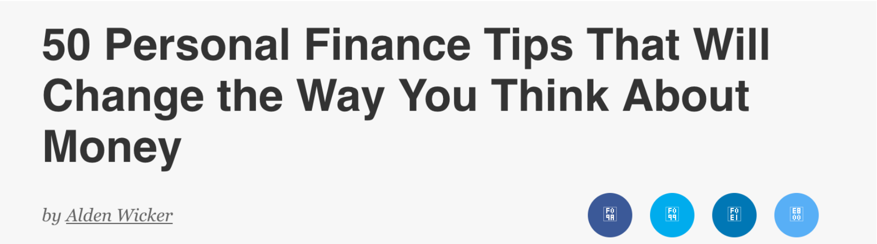
- 4 ways of doing something that’s a side task?
- I have to analyze, compare, evaluate?
  - Those are expert tasks!
- So people need experts to offer specific advice



Single best tool need: Personal finance example

- Max out your tax advantaged, matched accounts…

50?!?!!



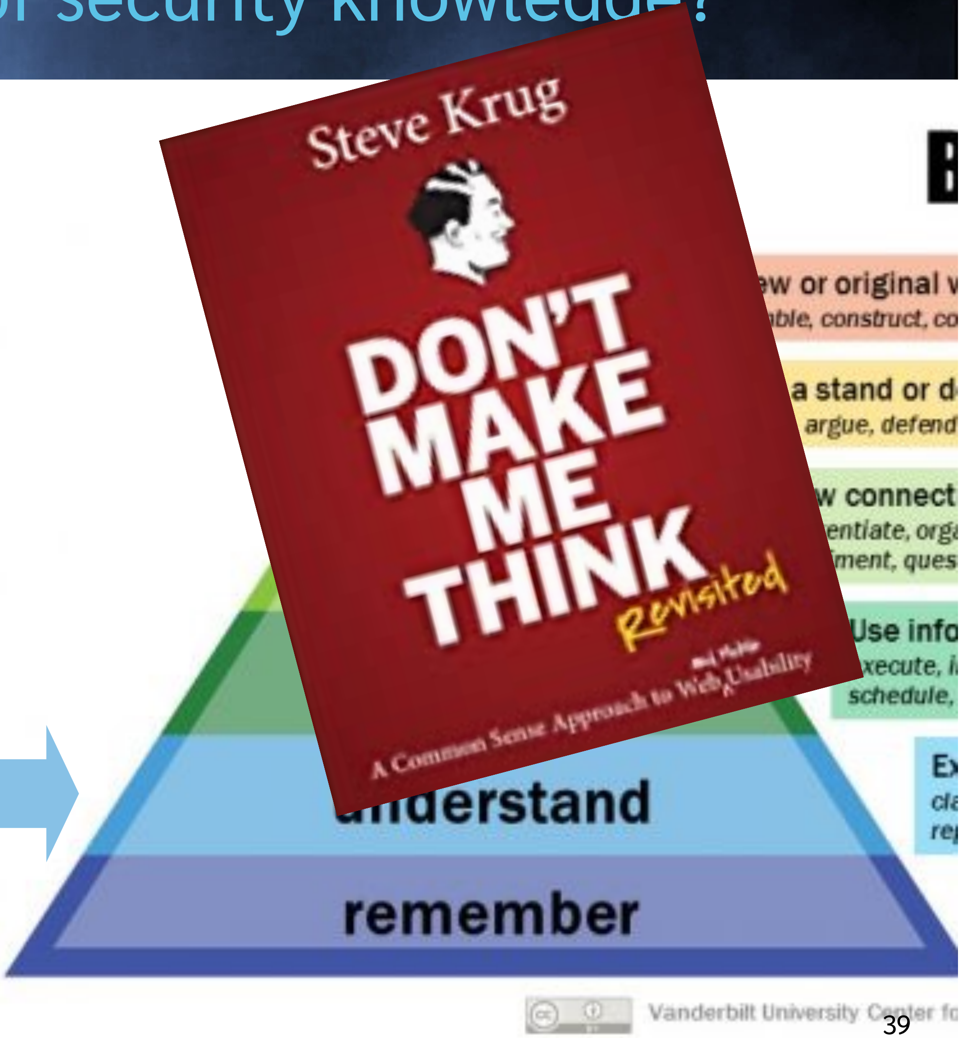
50 Personal Finance Tips That Will Change the Way You Think About Money

by Alden Wicker

- Target date funds

What’s the target date fund of security knowledge?

What fits here?



What does every engineer need to know?

- The question’s catalyzed by a couple of projects
  - Fast, Cheap + Good: An Unusual Tradeoff (whitepaper)
  - *Threats: What Every Engineer Should Learn from Star Wars*
- All of which started with a simple question…

Whitepaper available now Shostack.org/whitepapers/ | Book: Wiley, Feb, 2023

A simple question

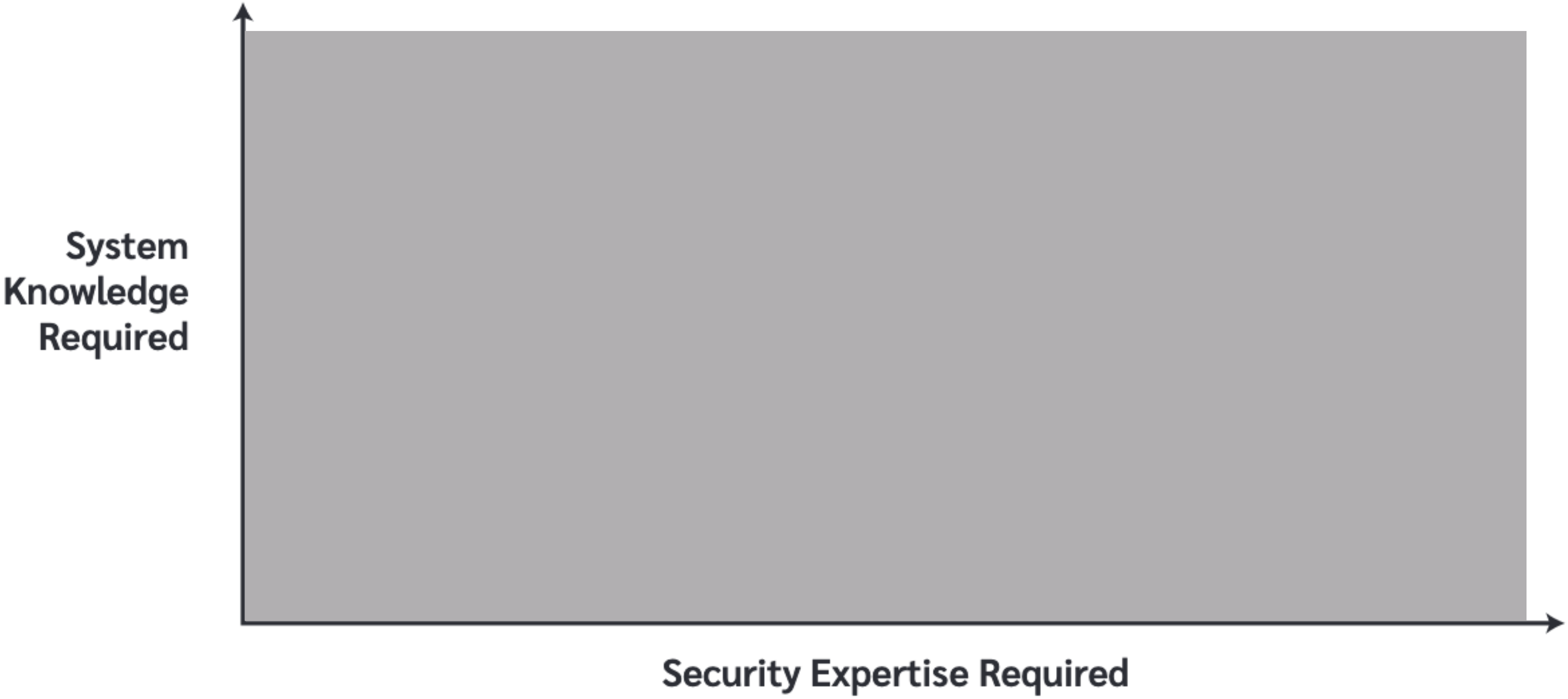
Is every flaw unique?

Another simple question

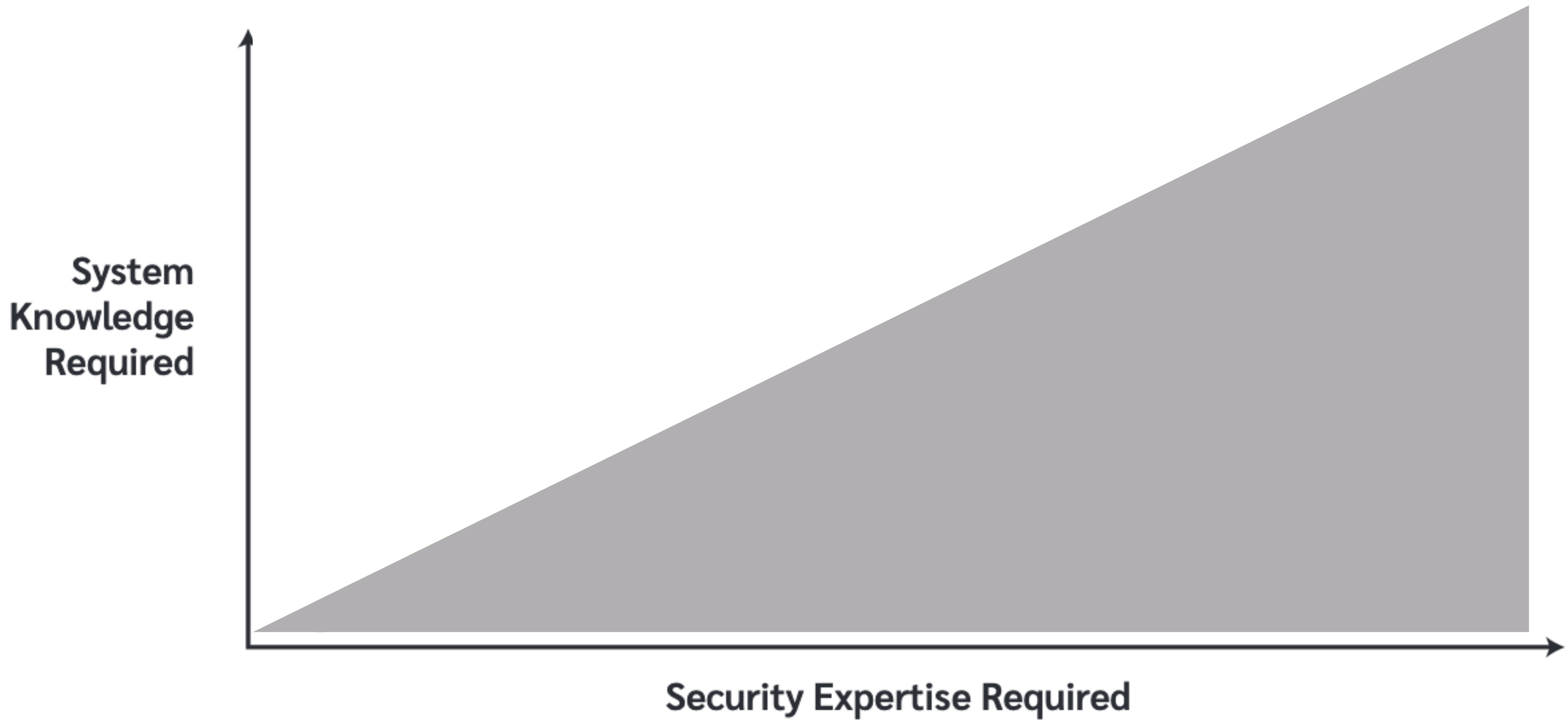
Do flaws cluster?

What do we need to know to find them?

Where are the flaws? (1)



Where are the flaws? (2)



Maybe they're lightweight?

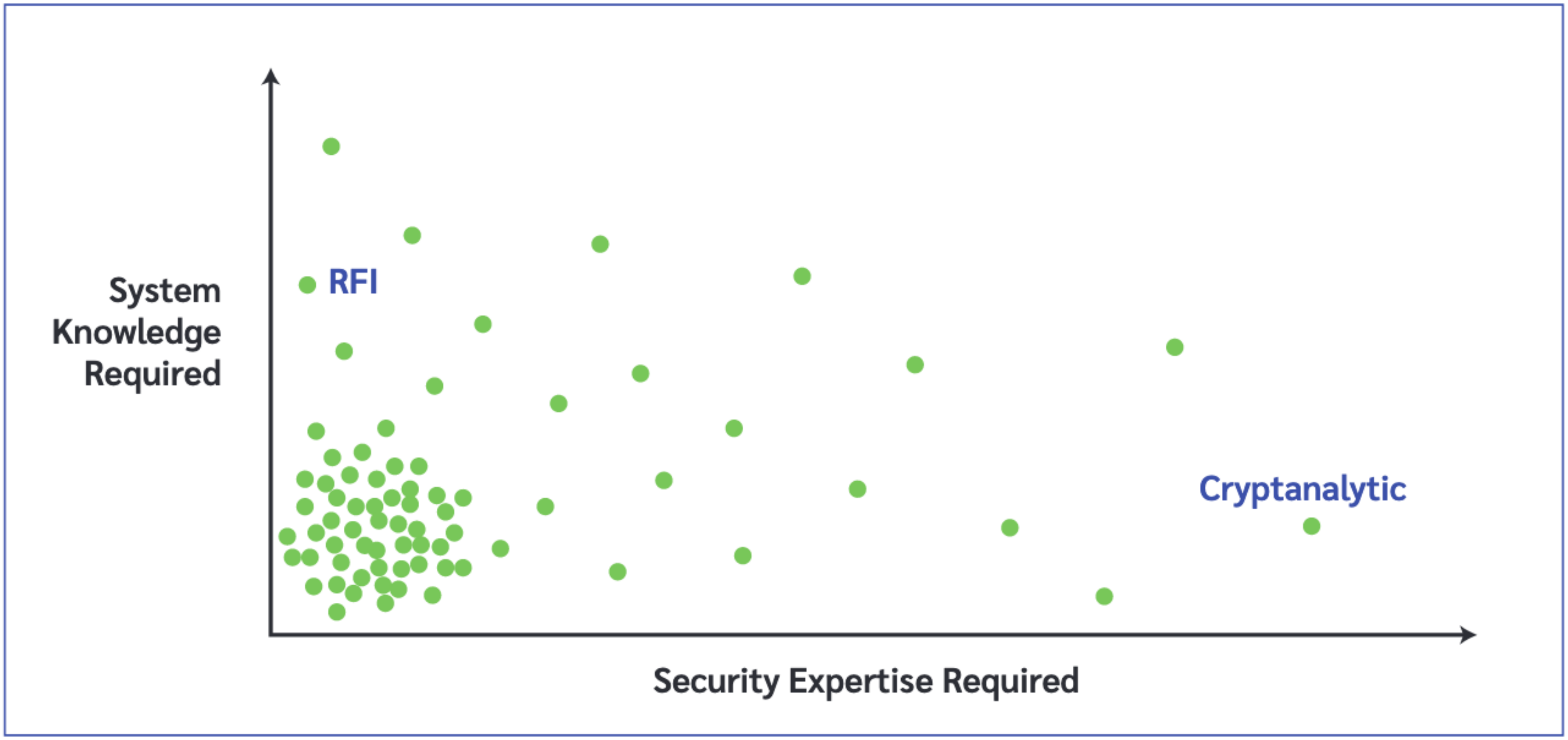


Figure 1: Knowledge required to find bugs

black hat USA 2022

If lightweight flaws are common,
we should transform
how we work with
engineers

What developers need to know: My proposal

- STRIDE threats
  - (Spoofing, Tampering, Repudiation, Info disclose, DoS, Expansion of Authority)
- Parsing + predictability generate danger
- Kill chains bring these together

What developers need to know (Samples)

- Remember that …
  - Spoofing must be addressed differently for each of
    - [machines, people] authenticating to [machines, people]
  - …Spoofing programs is easy unless the platform prevents it

Recap

- Code issues underly many (most?) security issues
- Shifting left is an admirable goal
- Only works when we’re clear about change

Rebellions are built on hope

- Normal levels of security are defined
- Developers able to build more secure systems
- Less rework, fewer escalations, more predictable delivery

Call to action

1. Define expectations: What developers know about threats
2. Help people meet them: Training, assessment
3. Measure impact

black hat USA 2022

Thank you!

Questions?
Now,
Swapcard (virtual event platform)
or
adam@shostack.org