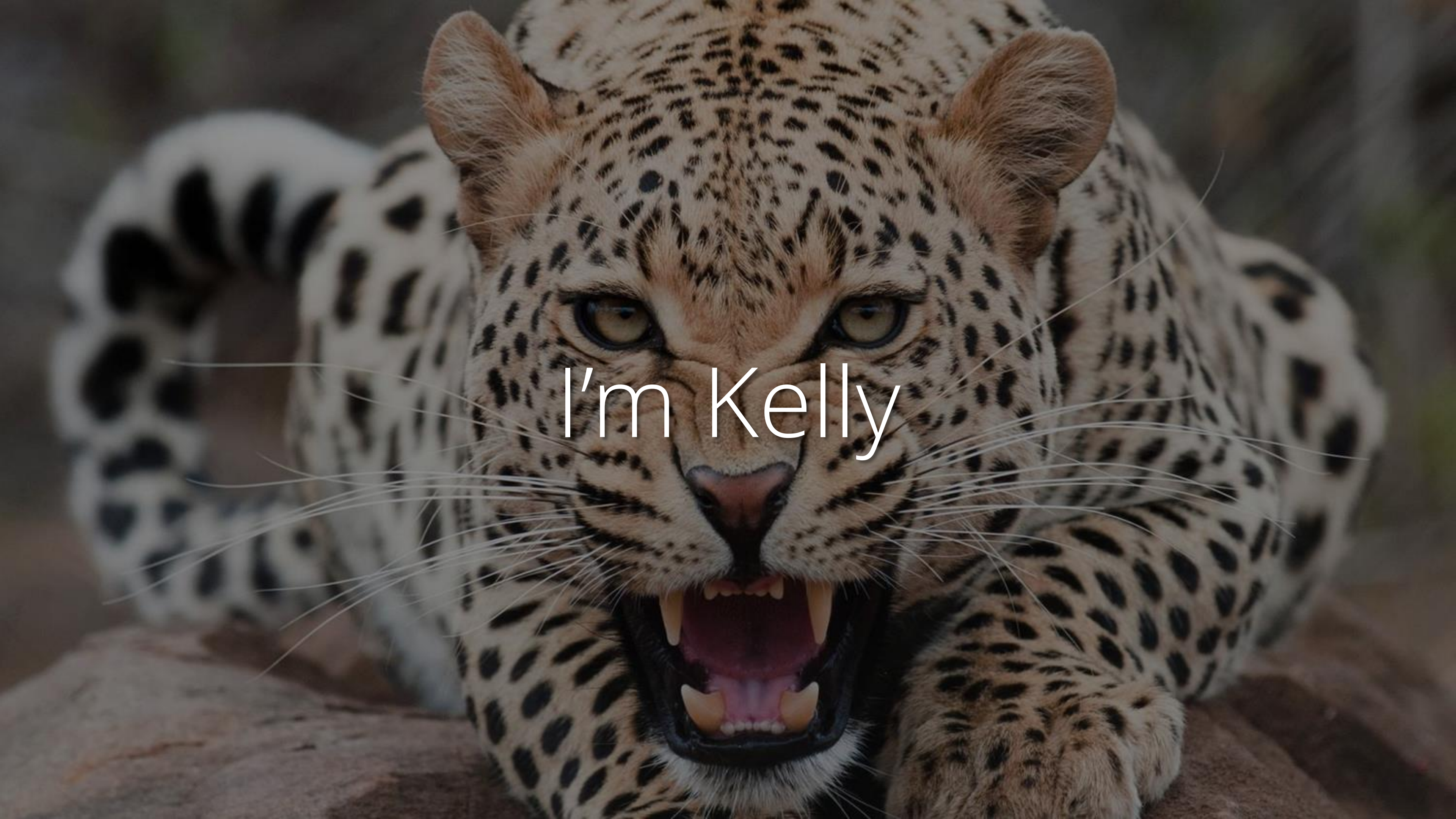
Big Game Theory Hunting

The Peculiarities of Human Behavior in the InfoSec Game

Kelly Shortridge (@swagitda_)

Black Hat 2017



I’m Kelly

This is game theory

It’s time for hunting some game theory

Do you believe bug-free software is a reasonable assumption?

Do you believe wetware is more complex than software?

Traditional Game Theory relies on the assumption of bug-free wetware

Behavioral Game Theory assumes there’s no such thing as bug-free

“Think how hard physics would be if particles could think”

—Murray Gell-Mann

“Amateurs study cryptography, professionals study economics”

—Dan Geer quoting Allan Schiffman

This is what you’ll learn:

1. Why traditional game theory isn’t even a theory and is unfit for strategy-making

2. A new framework for modeling the infosec game based on behavioral insights

3. New defensive strategies that exploit your adversaries’ “thinking” and “learning”

Let’s go hunting to find out why

I. What is Game Theory?

tl;dr – game theory is a mathematical language used to describe scenarios of conflict and cooperation

Game theory is more about language than theory

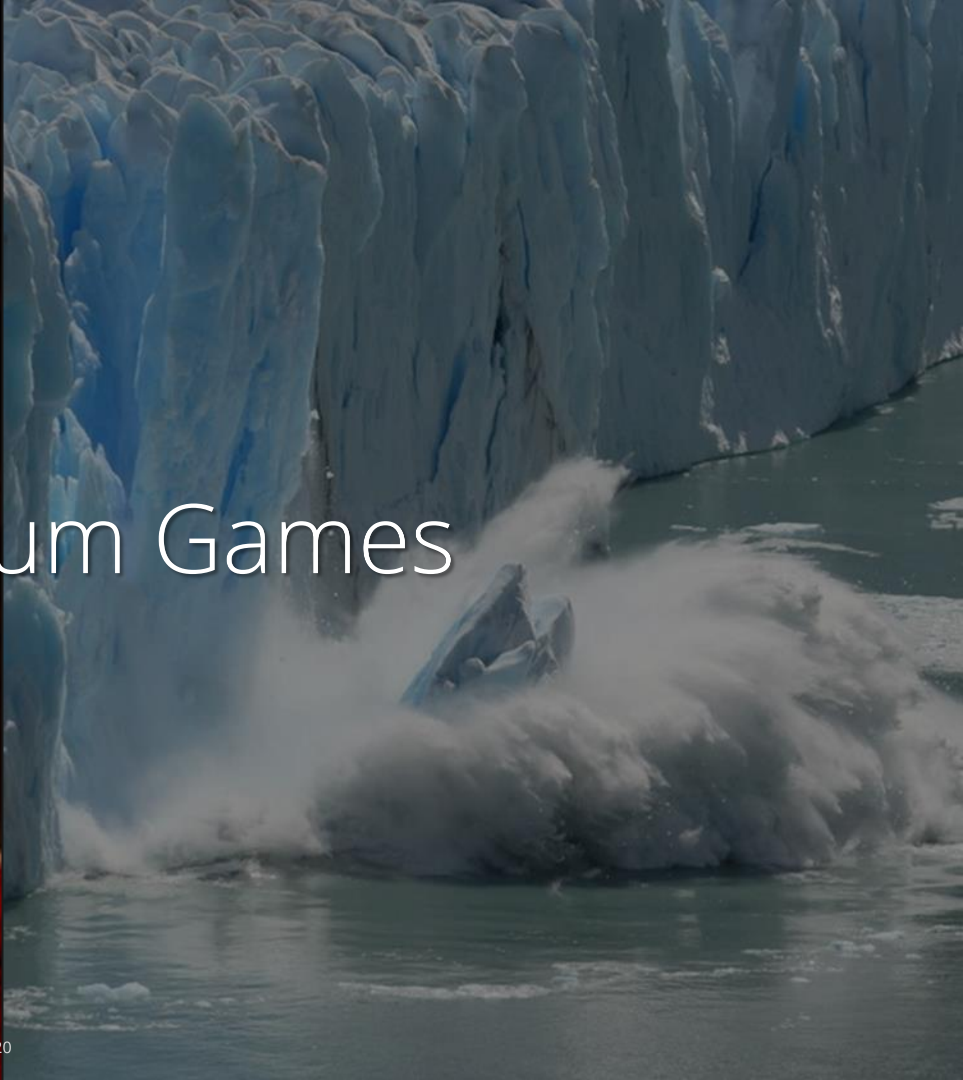
Use it as a engendering tool, not as something to dictate optimal strategies

GT applies whenever actions of players are interdependent

Strategic scenarios include many types of games, with different “solutions” for each

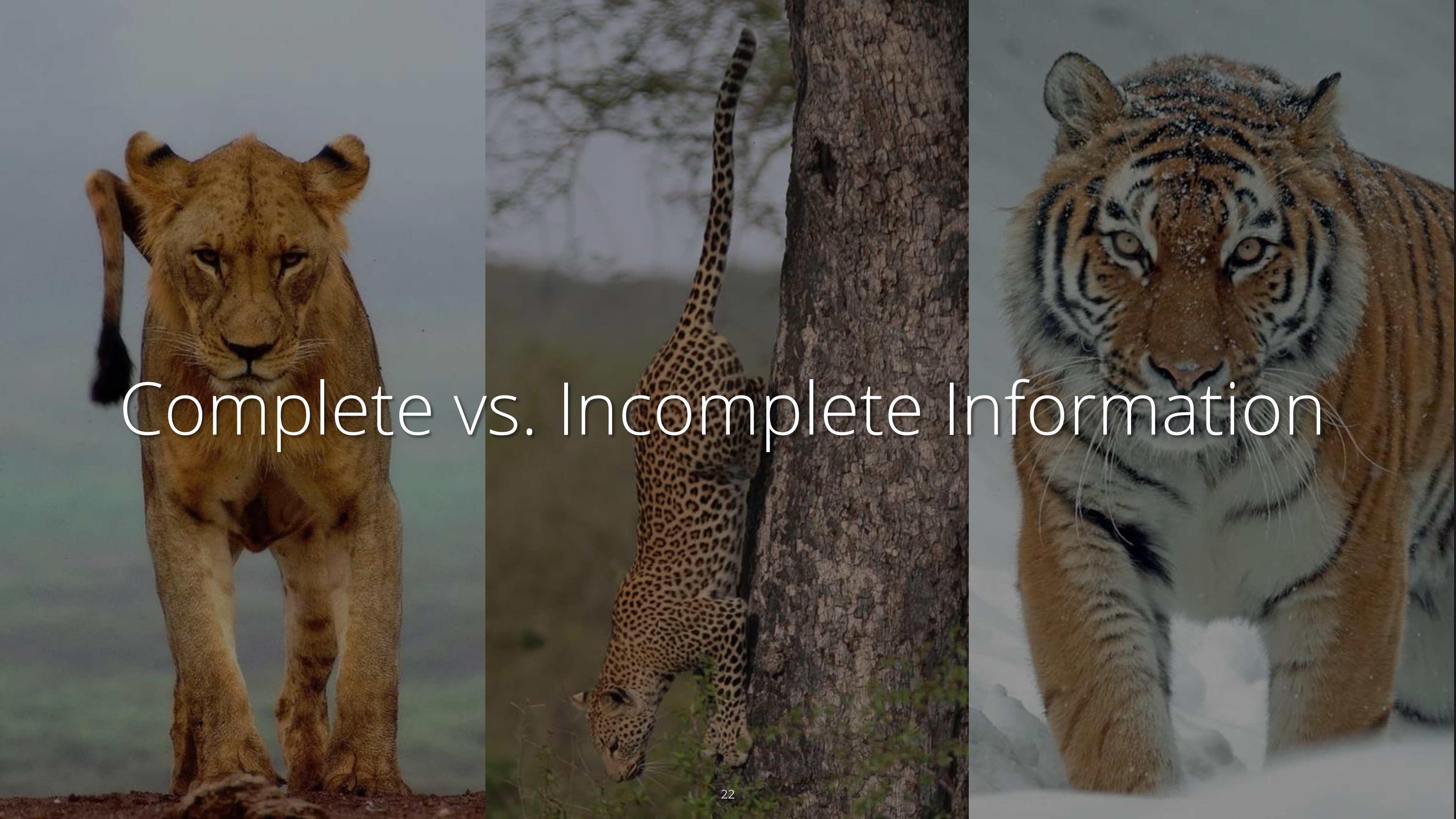
Zero Sum Games

Non-Zero Sum Games



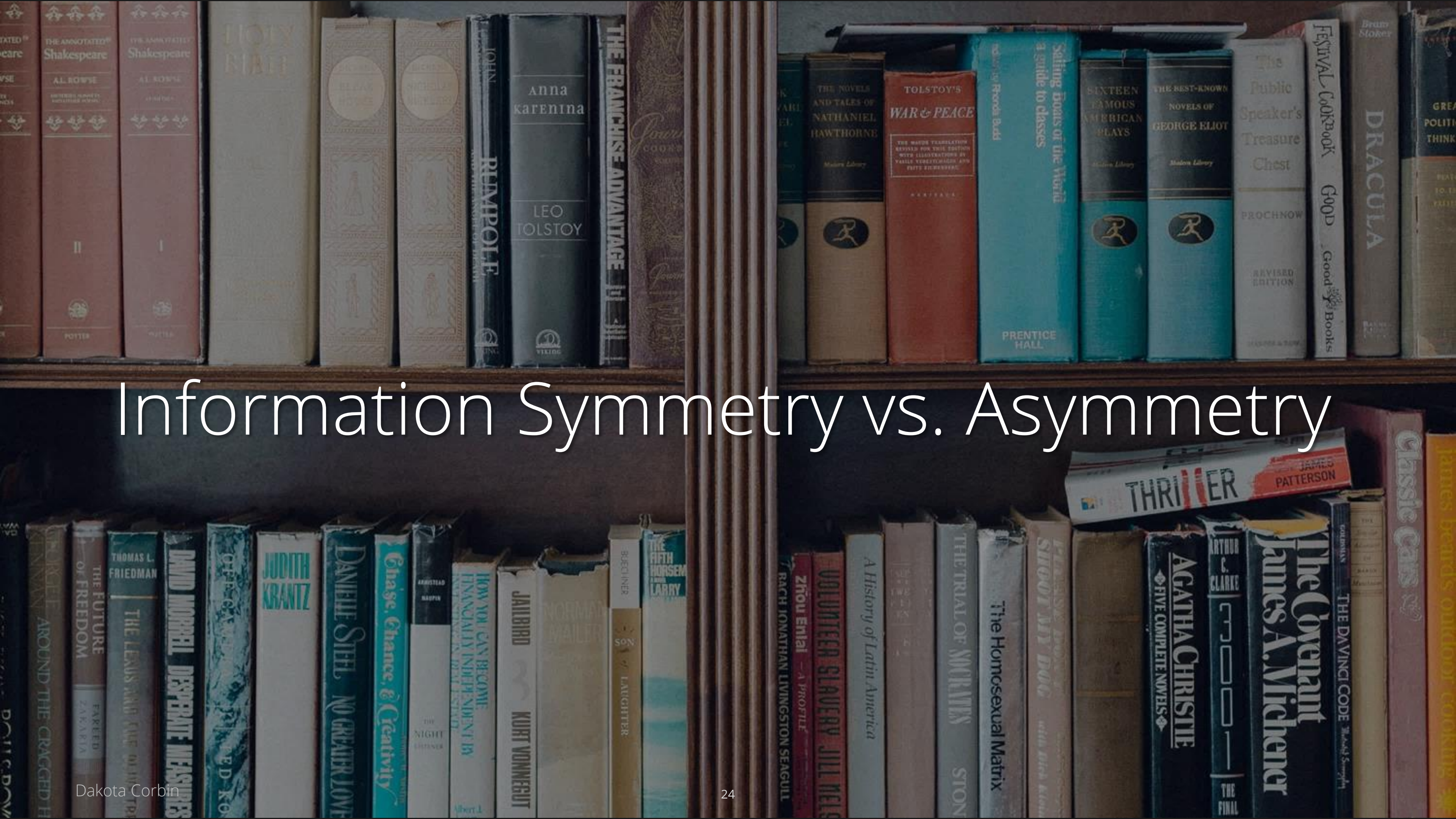
Negative Sum Games

Positive Sum Games



Complete vs. Incomplete Information

Perfect vs. Imperfect Information


The Annotated Shakespeare
A.L. Rowse
Holy Bible
Anna Karenina
Leo Tolstoy
Rumpole
The Franchise Advantage
The Novels and Tales of Nathaniel Hawthorne
Tolstoy's War & Peace
Sailing Boats of the World a guide to classes
Prentice Hall
Sixteen Famous American Plays
The Best-Known Novels of George Eliot
The Public Speaker's Treasure Chest
Prochnow
Festival Cookbook
Good Good Books
Bram Stoker
Dracula
Thriller
James Patterson
Thomas L. Friedman
The Future of Freedom
Fareed Zakaria
David Morrell Desperate Measures
Judith Krantz
Danielle Steel No Greater Love
Chaos, Chance, & Creativity
Jailbird Kurt Vonnegut
Norman Mailer
The Fifth Horseman Larry
Buechner Son of Laughter
Jonathan Livingston Seagull
Zhou Enlai A Profile
Volunteer Slavery Jill Nelson
A History of Latin America
The Trial of Socrates Stone
The Homosexual Matrix
Agatha Christie Five Complete Novels
Arthur C. Clarke
The Covenant James A. Michener
The Da Vinci Code
Classic Cars

Information Symmetry vs. Asymmetry

Defender Attacker Defender Games

Sequential games in which sets of players are attackers and defenders

Assumes people are risk-neutral & attackers want to be maximally harmful

First move = defenders choosing a defensive investment plan

Second move = attackers observe the defensive preparations & choose an attack plan

Nash Equilibrium is often used to “solve” games. This is bad.

Nash Equilibrium = optimal outcome of a non-cooperative game

Players are making the best decisions for themselves while taking their opponent’s decisions into account

Prisoner’s Dilemma

		Player 2	
		Confess	Refuse
Player 1	Confess	-2, -2	0, -4
	Refuse	-4, 0	-1, -1

Nash Equilibrium is based on a priori reasoning

Assumes rational, all-knowing players

Assumes others’ decisions don’t affect you

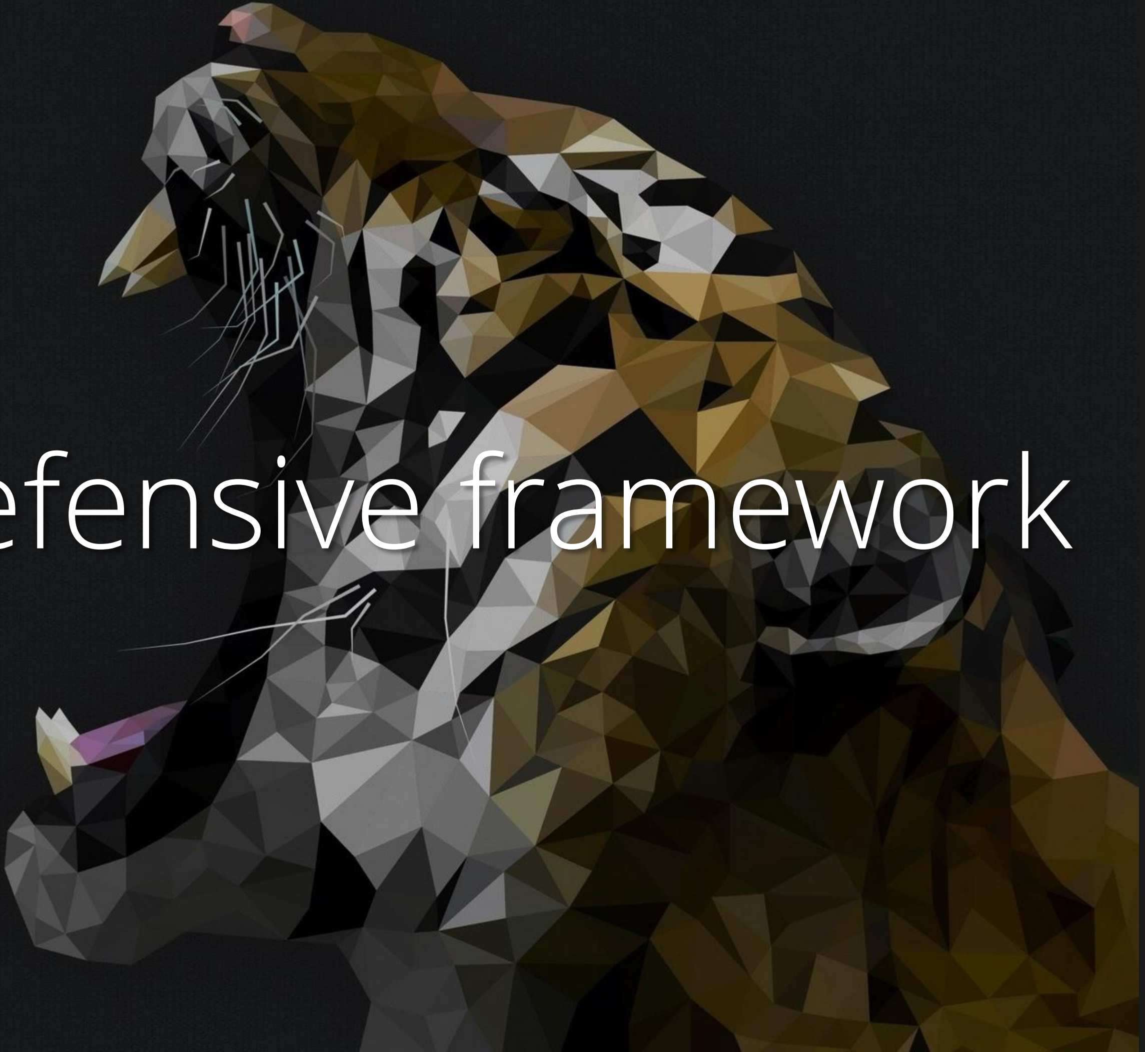
People have applied Nash Equilibrium to infosec over the years…

Defender should play extremely fast so the attacker drops out of the game

Better to invest in security than not invest, regardless of attacker strategy (wow!)

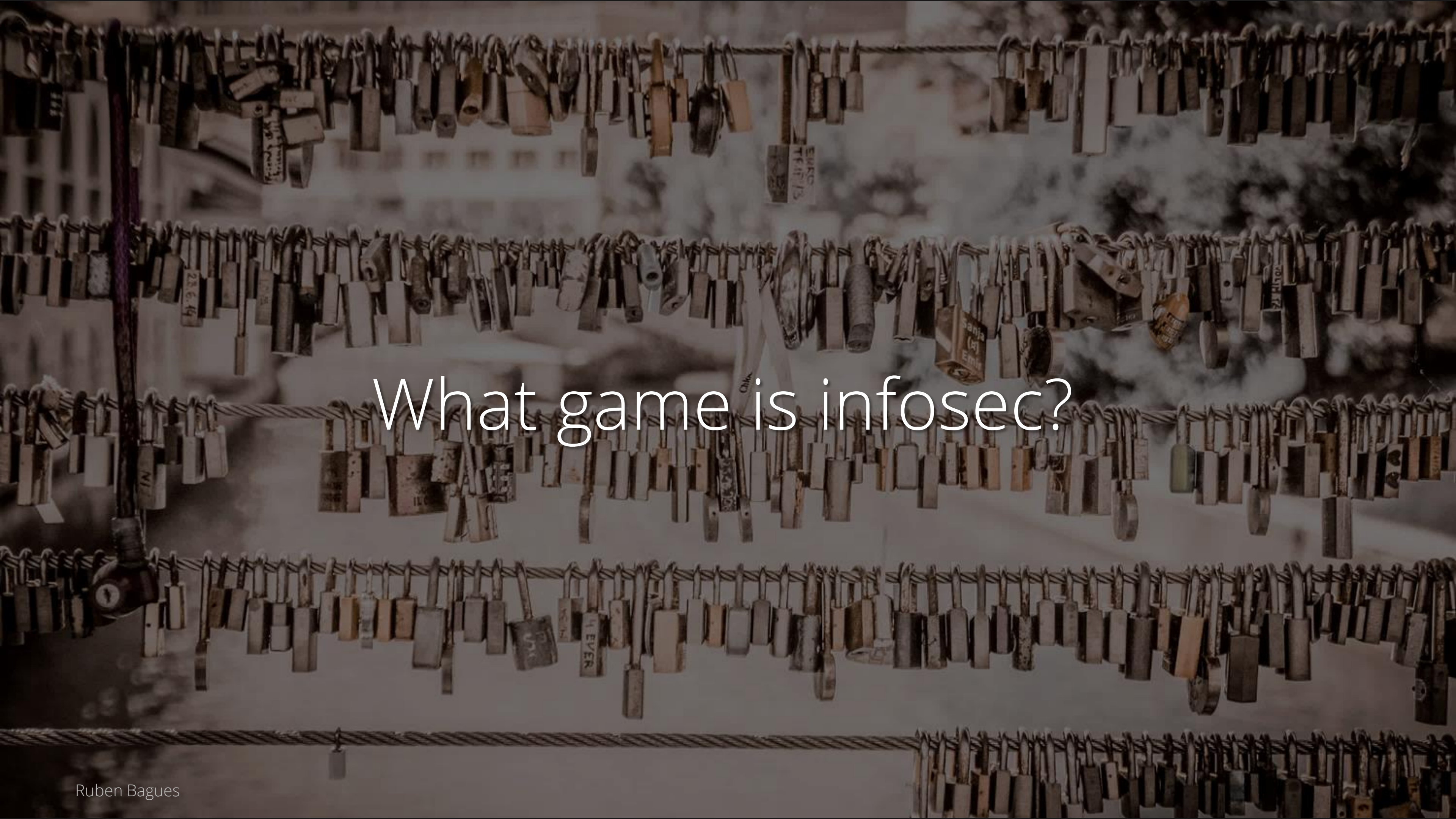
Just apply tons of mathematical equations!

II. New defensive framework



Use GT for its expressive power in describing a framework for the infosec game

Look at data outside GT, e.g. from experiments in domains similar to infosec, to select correct assumptions



What game is infosec?

Ruben Bagues

DAD game (continuous defense & attack)

Non-zero-sum

Incomplete, imperfect, asymmetrical information

Sequential / dynamic

This is a (uniquely?) tricky game

Have you heard infosec described as a “cat and mouse” game before?

Traditional Game Theory doesn’t allow for those…

…or most of the characteristics of the “infosec game”

Assumes people are rational (they aren't)

Assumes static vs. dynamic environments

Can't ever be "one step ahead" of your adversary

Deviations from Nash Equilibrium are common

“I feel, personally, that the study of experimental games is the proper route of travel for finding ‘the ultimate truth’ in relation to games as played by human players”

—John Nash

Behavior-based framework

Dayne Topkin

Experimental – how do people actually behave?

People predict their opponent’s moves by either “thinking” or “learning”

Thinking = modeling how opponents are likely to respond

Our brains work like volatile memory

Working memory is a hard constraint for human thinking

Enumerating steps past the next round is hard

Humans kinda suck at recursion

Learning = predicting how players will act based on prior games / rounds

Humans learn through “error-reinforcement learning” (trial & error)

People have “learning rates,” how much experiences factor into decision making

Dopamine neurons encode errors

Veksler & Buchler study

200 consecutive “security games” across 4 strategies

Different learning rates for attackers

Tested # of prevented attacks for each strategy

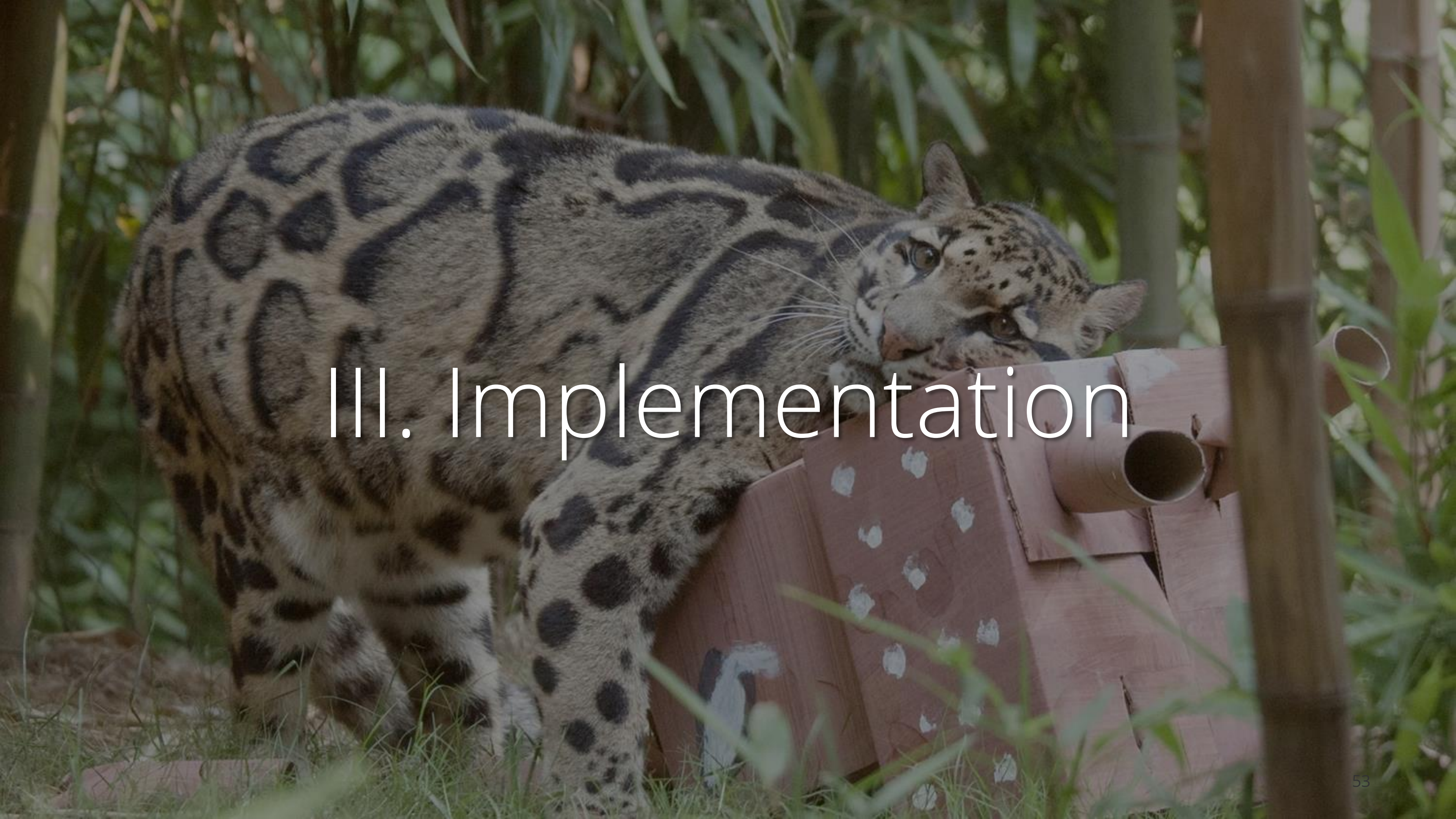
Fixed strategy = prevent 10% - 25% of attacks

Game Theory strategy = prevent 50% of attacks

Random strategy = prevent 49.6% of attacks

Cognitive Modeling strategy = prevent between 61% - 77%

Don’t be replaced by a random SecurityStrategy^{TM} algorithm



III. Implementation

1. SWOT Analysis
2. Thinking Exploitation
3. Learning Exploitation
4. Minimax
5. Looking Ahead

SWOT Analysis

Scott Webb

101: Traditional SWOT

Strengths, Weaknesses, Opportunities, Threats

Model SWOT for yourself in relation to your adversary

Model SWOT for your adversary in relation to you

> “The primary insight of GT is the importance of focusing on others – of putting yourself in the shoes of other players and trying to play out all the reactions…as far ahead as possible”
>
> – Adam Brandenburger

Strengths

- Understanding of target environment
- Motivation to not be breached

Weaknesses

- Inadequate budget
- Lack of personnel
- Limited employee training

Opportunities

- Leverage new tech to allow for tear up/down
- Increased board attention to get budget

Threats

- Attackers can use new tech for scalability
- Hard to keep up with pace of new attack surface

201: Perceptual SWOT

For you and your adversary, consider:

How can the strengths be weaknesses?

How can the weaknesses be strengths?

Self vs. Other

		Reality	
		Strength	Weakness
Perception	Strength		Threat
	Weakness	Opportunity	

		Reality	
		Strength	Weakness
Perception	Strength		Opportunity
	Weakness	Threat	

“Core rigidities” = deeply embedded knowledge sets that create problems

Compliance, fixed security guidelines

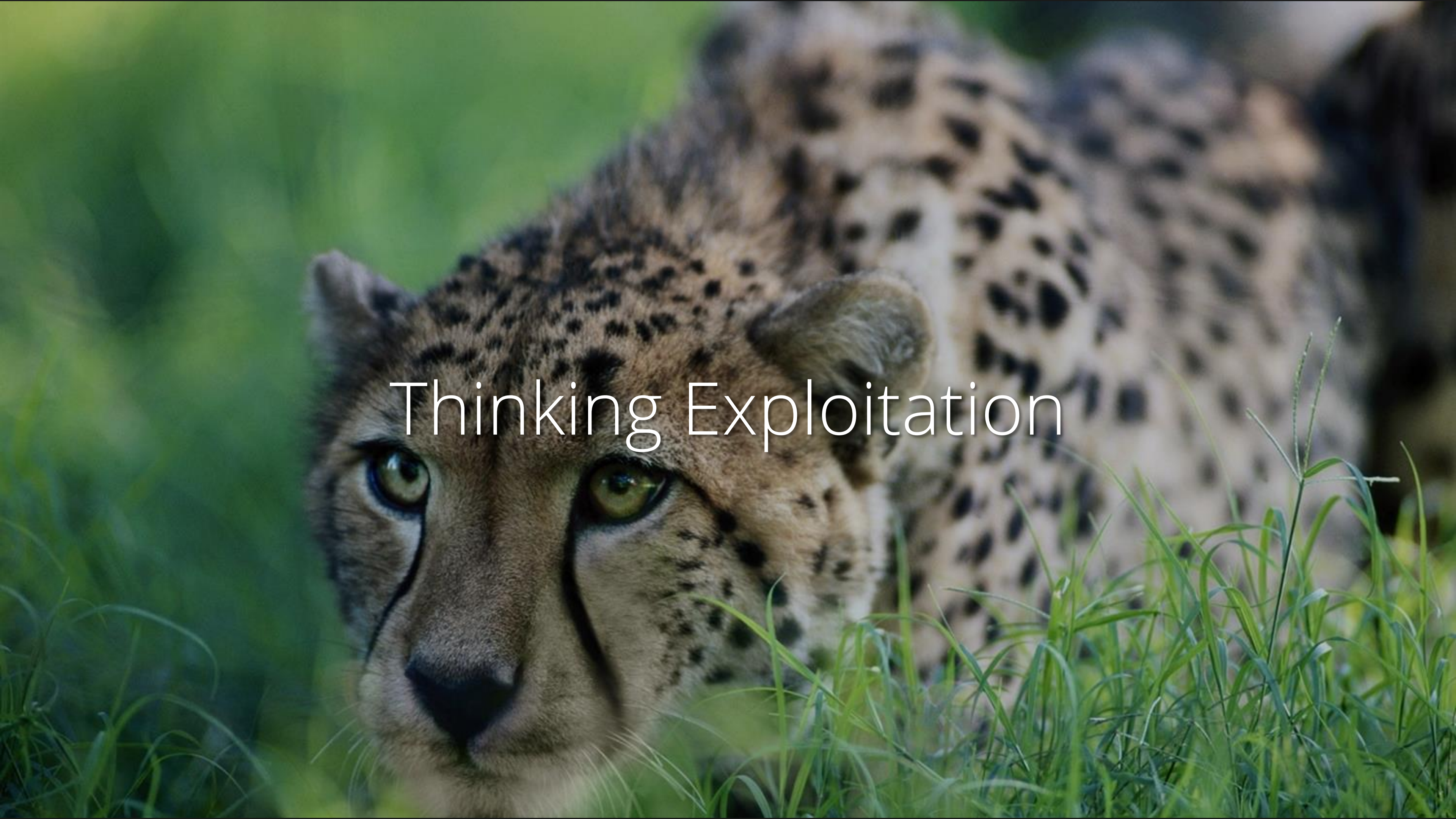
Top management can be the wrong people for an evolving environment

Attacker strength = having time to craft an attack

Leverage that “strength” with strategies leading down rabbit holes and wasting their time

Attacker strength = access to known vulns

Confuse them with fake architecture so they can’t be certain what systems you’re running



Thinking Exploitation

Thinking strategy: belief prompting

Increase players thinking by one step

“Prompt” the player to consider who their opponents are & how their opponents will react

Model assumptions around capital, time, tools, risk aversion

Your goal is to ask, “if I do X, how will that change my opponent’s strategy?”

A generic belief prompting guide:

How would attackers pre-emptively bypass the defensive move?

What will the opponent do next in response?

Costs of the opponent’s offensive move?

Probability the opponent will conduct the move?

Example: A skiddie lands on one of our servers, what do they do next?

Perform local recon, escalate to whatever privs they can get

Counter: priv separation, don’t hardcode creds

Leads to: attacker must exploit server, risk = server crashes

Decision Tree Modelling

Jessica Furtney

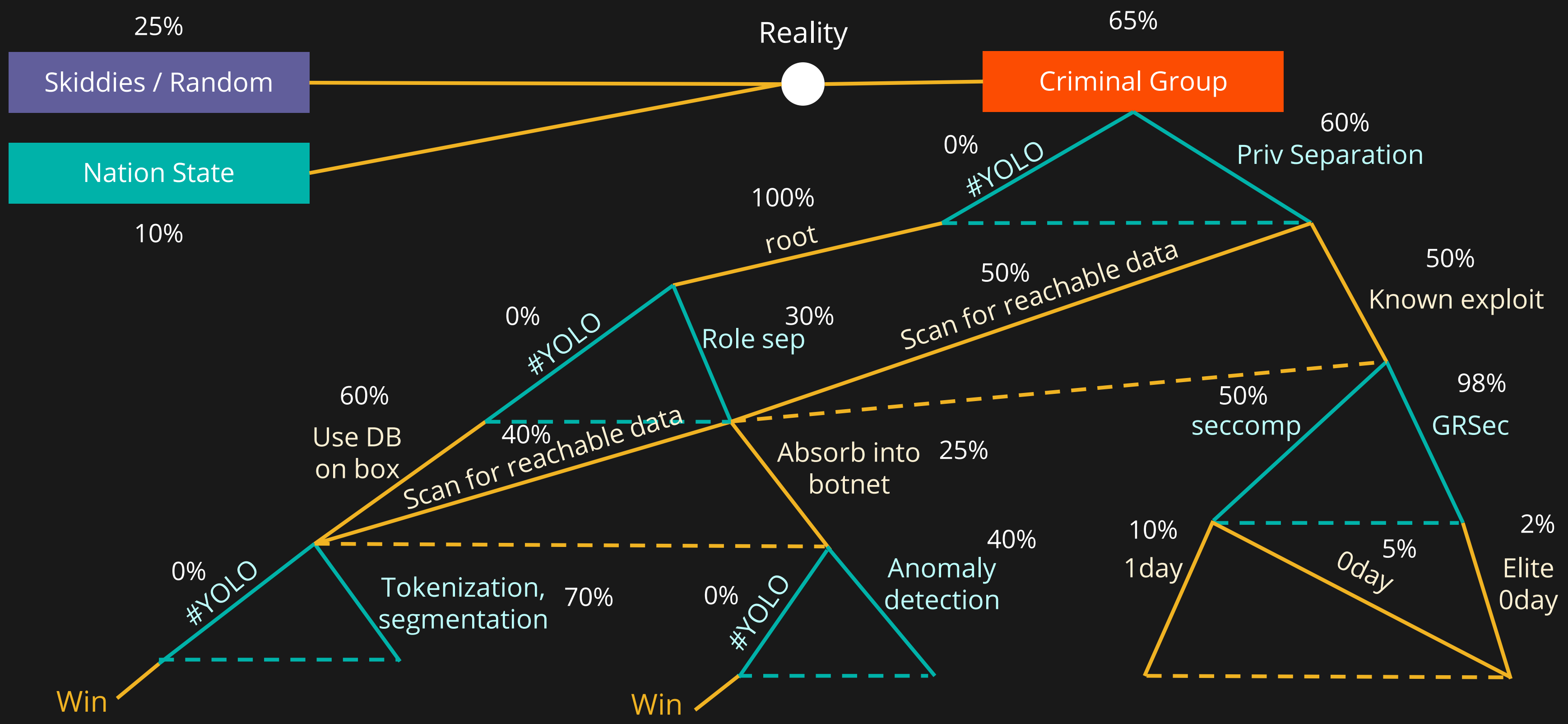
Model decision trees both for offense & defense

Theorize probabilities of each branch’s outcome

Creates tangible metrics to deter self-justification

> “Attackers will take the least cost path through an attack graph from their start node to their goal node”

– Dino Dai Zovi, “Attacker Math”


25%
Skiddies / Random
Nation State
10%
Reality
65%
Criminal Group
0%
#YOLO
60%
Priv Separation
100%
root
50%
Scan for reachable data
50%
Known exploit
0%
#YOLO
30%
Role sep
60%
Use DB
on box
40%
Scan for reachable data
50%
seccomp
98%
GRSec
Absorb into
botnet
25%
0%
#YOLO
Tokenization,
segmentation
70%
0%
#YOLO
40%
Anomaly
detection
10%
1day
5%
0day
2%
Elite
0day
Win
Win

1. Which of your assets do attackers want?
2. What’s the easiest way attackers get to those assets?
3. What countermeasures are on that path?
4. What new path will the attacker take given #3?
5. Repeat 1 – 4 until it’s “0day all the way down”
6. Assign rough probabilities

Whiteboards + camera snaps (or “DO NOT ERASE!!!!”)

Draw.io, Gliffy (plugs into Confluence)

Google Docs (> insert drawing)

PowerPoint (what I used)

Visio (last resort)

Decision trees help create a feedback loop to refine strategy

Decision trees help for auditing after an incident & easy updating

Serves as a historical record to refine decision-making process

Mitigates “doubling down” effect by showing where strategy failed

Defender’s advantage = knowing the home turf

Visualize the hardest path for attackers – how can you force them onto to that path?

Commonalities on trees = which strategies mitigate the most risk across various attacks

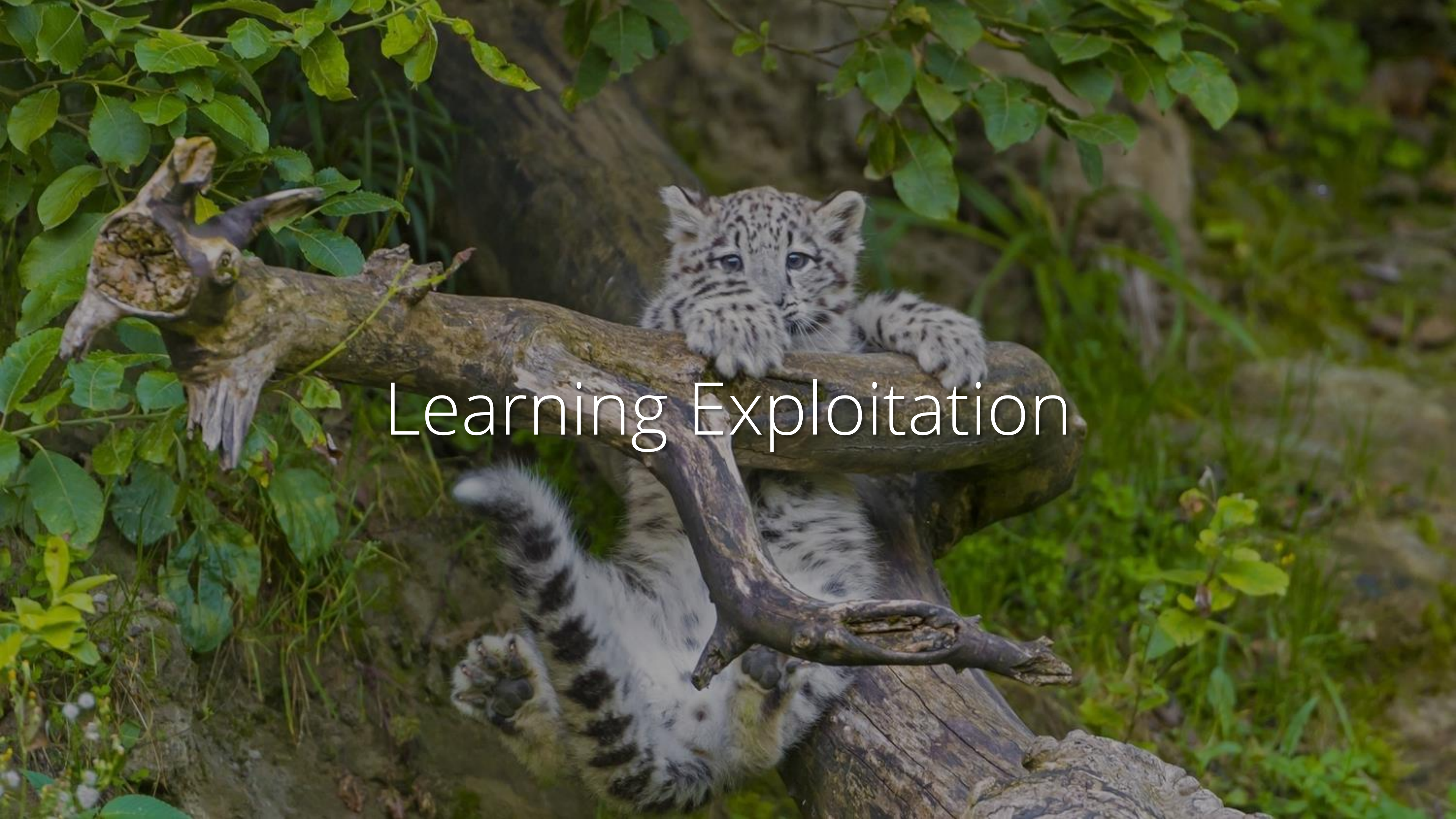
Make decision trees the new “nice report”

#WOCinTech

A new request for your pen-testers / red-team

The ask: outline which paths they did or didn’t take, and why (a decision tree w/ explanations)

Helps you see the attacker perspective of your defenses & where to improve



Learning Exploitation

Information asymmetry exploitation – disrupt attacker learning process

Learning rate exploitation – introduce unreliability and pre-empt moves

Exploit the fact that
you understand the local environment
better than attackers

Falsifying Data

Braydon Anderson

Defenders have info adversaries need to intercept

Dynamic envs = frequently in learning phase

Hide or falsify data on the legitimate system side

SEGUIN LALANDE TRESCA PONCELET BRESSE LAGRANGE BELANGER CUVIER LAPLACE DULONG CHASLES LAVOISIER AMPERE CHEVREUL FLACHAT NAVIER LEGENDRE CHAPTAL

Macron Case Study

Soroush Karimi

Allegedly used phishing tarpitting

Signed onto phishing pages & planted bogus creds and info

Obvious fakes in dumped documents

#wastehistime2016…but for hackers

Goal is to remove the attacker’s scientific method so they can’t test hypotheses

(Pretend like hashtags are a thing and tweet #wastehackertime2017 with your own ideas)

Create custom email rejection messages

Create honeydoc on the “Avallach Policy”

Have response to suspicious emails be, “This violates the Avallach policy”

Track when the doc is accessed

General strategy: create honeytokens that look to describe legitimate policies or technologies that would be useful in attacker recon

Non-Determinism

Candice Seplow

Different behaviors at different times

Can’t expect same result every time

ASLR is a non-deterministic feature, but highly deterministic in that it always works the same

I want to amplify and extend it to higher levels

Raise costs at the very first step of the attack: recon

Make the attacker uncertain of your defensive profile and environment

Attackers now design malware to be VM-aware

#WOCinTech

Good: Make everything look like a malware analyst’s sandbox

Better: Make everything look like a different malware analyst’s sandbox each time

Put wolfskins on the sheep

Mix & match hollow but sketchy-looking artifacts on normal, physical systems

RocProtect-v1 – https://github.com/fr0gger/RocProtect-V1

Emulates virtual artifacts onto physical machine

(see Unprotect Project as well)

VMwareServices.exe
VBoxService.exe
Vmwaretray.exe
VMSrvc.exe
vboxtray.exe
ollydbg.exe
wireshark.exe
fiddler.exe

\\\\.\\pipe\\cuckoo
cuckoomon.dll
dbghelp.dll

Mac addresses:
"00:0C:29", "00:1C:14",
"00:50:56", "00:05:69"

system32\drivers\VBoxGuest.sys
system32\drivers\VBoxMouse.sys

HKLM\SOFTWARE\Oracle\VirtualBox Guest Additions

C:\cuckoo, C:\IDA
Program Files\Vmware

Make the IsDebuggerPresent function call always return non-zero

Create fake versions of driver objects like \\.\NTICE and \\.\SyserDbgMsg

Set KdDebuggerEnabled to 0x03

Load DLLs from AV engines using a Windows loader with a forwarder DLL

ex64.sys (Symantec)
McAVSCV.DLL (McAfee)
SAUConfigDLL.dll (Sophos)
cbk7.sys (Carbon Black)
cymemdef.dll (Cylance)
CSAgent.sys (Crowdstrike)

Deploy lightest weight hypervisor possible for added “wolfskin”

https://github.com/asamy/ksm
https://github.com/ionescu007/SimpleVisor
https://github.com/Bareflank/hypervisor

Minimax

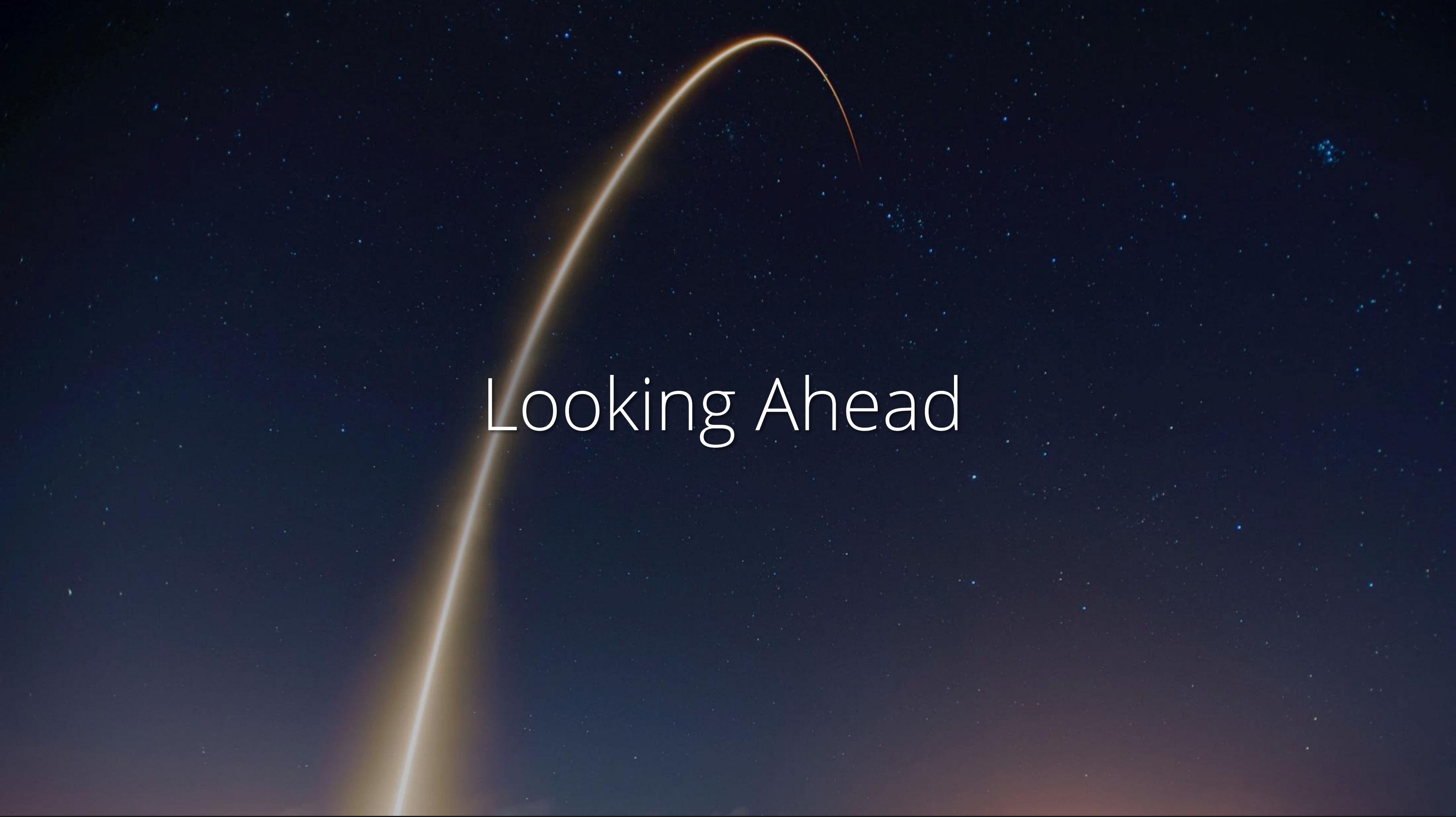
Minimax / maximin = minimize the possible loss for a worst case maximum loss scenario

Want to find the minimum of the sum of the expected cost of protection and expected cost of non-protection

Don’t have a monoculture – diversity is strongly beneficial for protection

Stochastic decisions may be better than deterministic

From *The Imitation Game*: should only act on Enigma info some of the time, not all



Looking Ahead

Fluctuating infrastructure using emerging tech in “Infrastructure 3.0”

Netflix’s Chaos Monkey
https://github.com/Netflix/SimianArmy/wiki/Chaos-Monkey

Modelling attacker cognition via model tracing

Prerequisite: how to begin observing attacker cognition

Preferences change based on experience

Models incorporate the “post-decision-state”

Higher the attacker’s learning rate, easier to predict their decisions

$\Delta U_A = \alpha\,(R - U_A)$, where:

- U_A = expected utility of an offensive action
- α = learning rate
- R = feedback (success / failure)

If α = 0.2, R = 1 for win & -1 for loss, then:

- $\Delta U_A = 0.2(1 - 0) = 0.2$

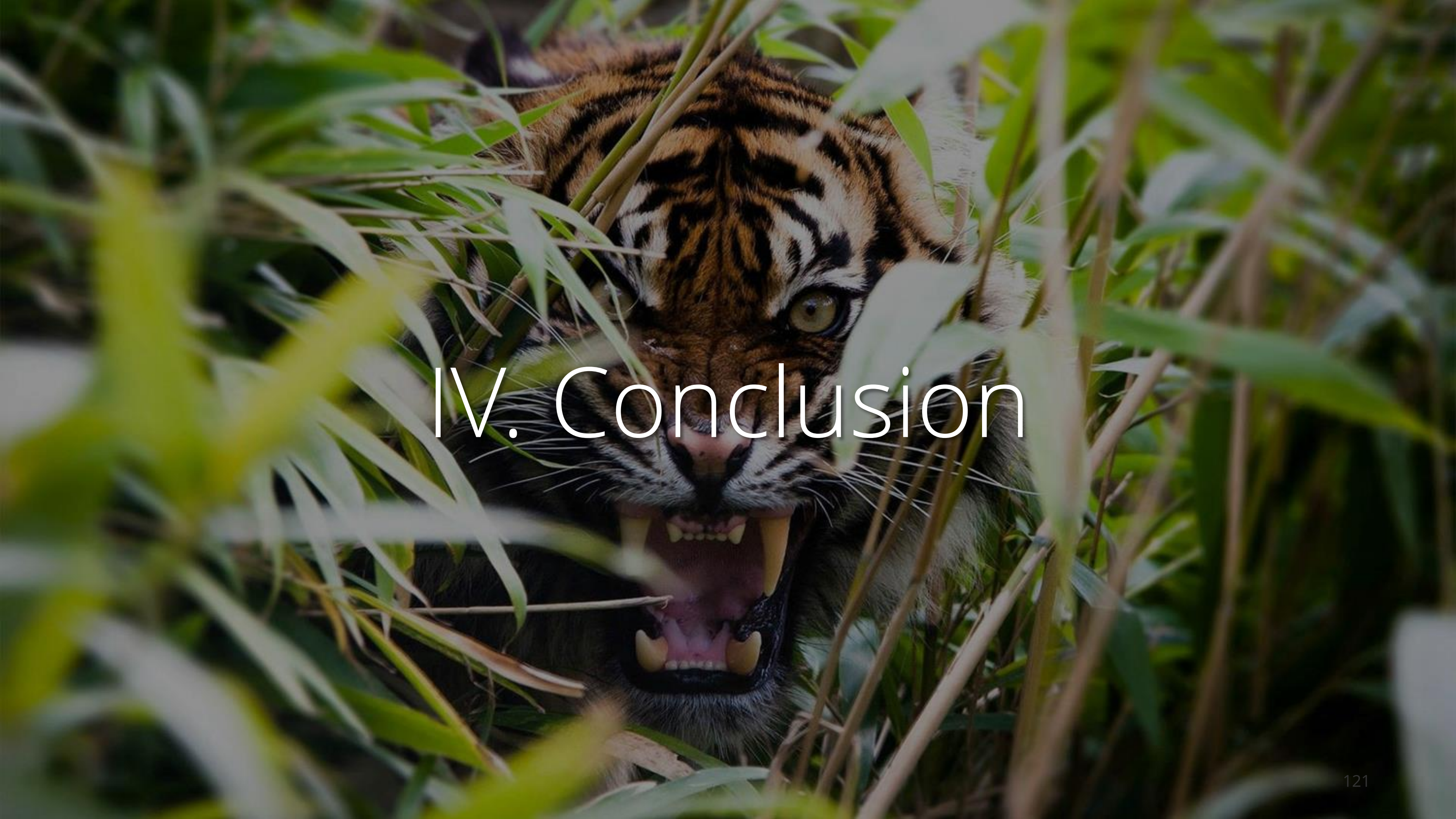
Attacker is 20% more likely to do this again

From here, you can adjust the learning rate based on data you see

Track utility values for each attacker action

For detected / blocked actions, attacker action & outcome are known variables (so utility is calculable)

Highest “U” = action attacker will pursue



IV. Conclusion

It is no longer time for some Game Theory

In fact, we've learned that GT is a language, not even a theory

Start with a SWOT analysis to gain perspective

Use thinking exploitation to improve threat modelling

Use learning exploitation to beleaguer your adversaries

Let’s work together to build strategies based on this behavioral framework

Next step – how to begin model-tracing attackers

After that – predict attacker behavior

Try these at home – make your blue team empirical

Worst case, random strategies beat fixed ones & are just as good as GT

“Good enough is good enough. Good enough always beats perfect.”

—Dan Geer

Suggested reading

- David Laibson’s Behavioral Game Theory lectures @ Harvard
- “Game Theory: A Language of Competition and Cooperation,” Adam Brandenburger
- “Advances in Understanding Strategic Behavior,” Camerer, Ho, Chong
- “Know Your Enemy: Applying Cognitive Modeling in the Security Domain,” Veksler, Buchler
- “Know Your Adversary: Insights for a Better Adversarial Behavioral Model,” Abbasi, et al.
- “Deterrence and Risk Preferences in Sequential Attacker–Defender Games with Continuous Efforts,” Payappalli, Zhuang, Jose
- “Improving Learning and Adaptation in Security Games by Exploiting Information Asymmetry,” He, Dai, Ning
- “Behavioral theories and the neurophysiology of reward,” Schultz
- “Evolutionary Security,” and “Measuring Security,” Dan Geer

@swagitda_

/in/kellyshortridge

kelly@greywire.net