Carnegie Mellon University
Software Engineering Institute

Chain Games: Powering Autonomous Threat Hunting

NOVEMBER 14–16, 2022

Phil Groce
Senior Network Defense Analyst

Document Markings

Get the New DM statement and number and copy it into the text box. Select “text only” to maintain font size and formatting 

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM22-0881

Introduction

Threat hunting is a critical part of cyber defense, but the amount of data available to threat hunters is overwhelming.

To develop effective autonomous threat hunting techniques, we are developing Chain Games, a set of games in which threat hunting strategies can be evaluated and refined.

What Is Threat Hunting? –1

Are attackers trying to get into the network?

- Security monitoring
- Intrusion prevention/detection systems (IDS/IPS); security event managers (e.g., SIEMs)
- Attacker tactics, techniques and procedures (TTPs) → IDS/IPS configuration

Are attackers already in the network?

- Threat hunting

Where have the attackers been in the network?

- Incident response
- Security Orchestration and Automated Response (SOAR)
- Incidents → response playbooks

What Is Threat Hunting? –2

Initiation

Are attackers trying to get into the network?

- Intrusion prevention/detection
- Stopping attacks before they start
- Impossible to stop everything

Planning/Execution

Are attackers already in the network?

- Threat hunting
- Finding attackers in a position to do harm
- Finding attackers (hopefully!) before they can do harm
- Expensive and time-consuming

Culmination

What have the attackers done in the network?

- Incident response
- Mitigating damage that has been done
- Damage that has already been done

Approaches to Autonomy

Long-term goal: autonomy

- Predication
- Investigation
- Conclusion

Short-term goal: modeling

- Quantitatively evaluating and developing strategies
- Rapid strategic development
- Tools for abstract simulation of threat hunting
- Capturing the adversarial quality of threat hunting activity

Cyber Deception Games (CDG) and Cyber Camouflage Games (CCG)

2018: Cyber Deception Games [1]

- Situates work in the Cyber Kill Chain
  - Focuses on reconnaissance
- Is a zero-sum game
- Defender is deceiver

2019: Cyber Camouflage [2]

- Is extended to general-sum games
- Defender is still deceiver

[1] Schlenker A, Thakoor O, Xu H, Fang F, Tambe M, Tran-Thanh L, Vayanos P, Vorobeychik Y, "Deceiving cyber adversaries: A game theoretic approach," in Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 892-900.

[2] Thakoor O, Tambe M, Vayanos P, Xu H, Kiekintveld C, Fang F. "Cyber Camouflage Games for Strategic Deception," in Decision and Game Theory for Security, Springer International Publishing, 2019, pp. 525-541.

Kill/Attack Chains

The Cyber Kill Chain graphic is reused with permission from Lockheed Martin Corporation. [3]

Attack behavior is often conceptualized as chains.

- Decomposes attacks
- Categorizes attack behaviors

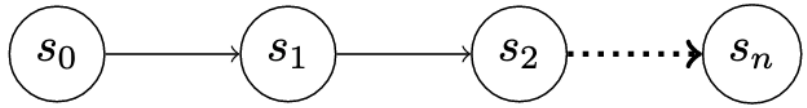
Attack graphs are a composition of attack chains.

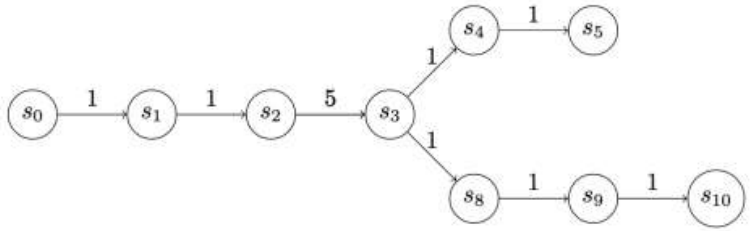
[3] Lockheed-Martin. The Cyber Kill Chain. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

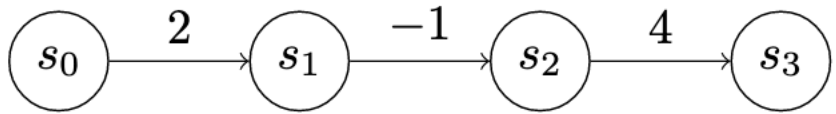
Chain Games –1

Chain Games are played on state chains.

- States represent positions in the network conveying advantage (or disadvantage) to the attacker.
- The utility and cost of occupying a state can be quantified.
- Progress through the state chain motivates the attacker; stopping progress motivates the defender.



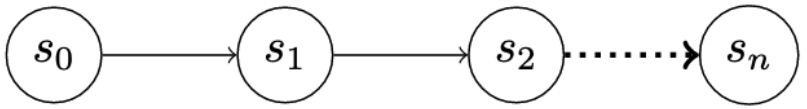


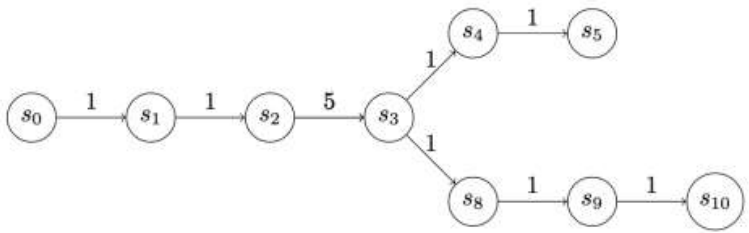


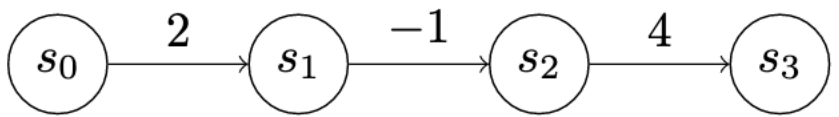
Chain Games –2

Rules

- Two players (Attacker and Defender)
- Fixed number of turns
- Simultaneous action







Chain Game Version 0

Actions

Attacker: Advance (a), wait (0)

- a costs 1; wait costs 0
- a advances to next state in chain

Defender: Defend (d), wait (0)

- d costs 1; wait costs 0
- d action negates attacker a action

Payouts

Attacker: advance value of the state

Defender: -1 * advance value

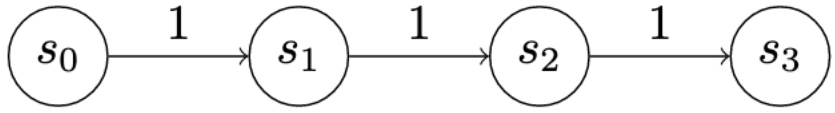
- Motivates defensive behavior

Dominant Strategies

Attacker: Always advance

Defender: Never detect

- Stopping attacker doesn't generate enough utility (no reward for detection).
- Takeaway: The full value of a strategy is its utility across all opponent strategies.



	$[\emptyset, \emptyset]$	$[a, \emptyset]$	$[\emptyset, a]$	$[a, a]$
$[\emptyset, \emptyset]$	$(0, 0)$	$(-2, 1)$	$(-2, 1)$	$(-4, 2)$
$[\emptyset, d]$	$(-1, 0)$	$(-3, 1)$	$(-1, -1)$	$(0, -3)$
$[d, \emptyset]$	$(-1, 0)$	$(-1, -1)$	$(-3, 1)$	$(-3, 0)$
$[d, d]$	$(-2, 0)$	$(-2, -1)$	$(-2, -1)$	$(-2, -2)$

Payout Matrix Over Two Turns, Uniform-Value Chain

Chain Game Version 1: Camouflage

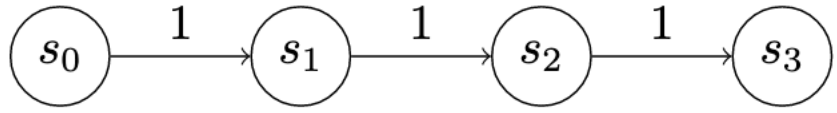
Actions

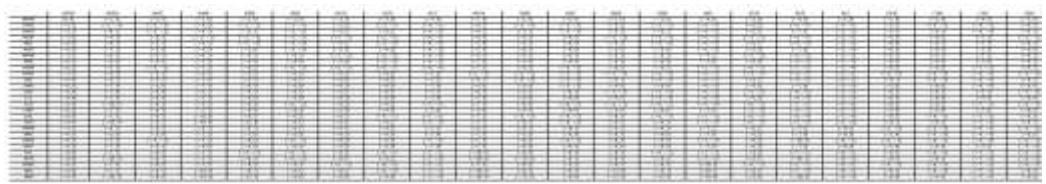
Attacker: Noisy Advance (an), Camouflaged Advance (ac)

- ac more costly than an

Defender: Weak Detect (dw), Strong Detect (ds)

- dw only detects an
- ds more costly than dw

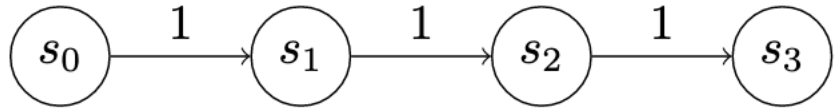


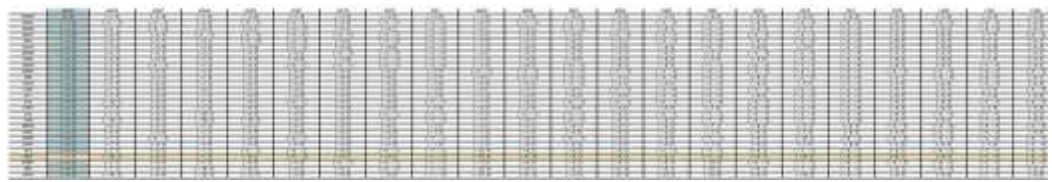


Payout Matrix Over Three Turns

Chain Game Version 1: Dominant Strategies

- One strategy dominates for each player.
- Players are **assumed to be** rational, so theoretic analysis means they only ever play one strategy.
- Each step of progress is **assumed to be equal**.
- Both of these assumptions do not hold in real-world conditions.
- Simulation gives us insight into “irrational” strategies (e.g., counterplaying the “rational” strategy).





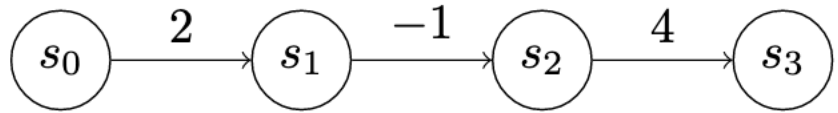
Payout Matrix Over Two Turns

Non-Uniform Value Chains

- There is no dominant pure strategy for attacker or defender.
- Non-uniform chains represent more realistic attack conditions.
- Initial infection is valuable.
- Some positions of advantage may have value that justifies taking on intermediate risk.

2: WWW	2: WWA	2: WAW	2: WAA	2: AWW	2: AWA	2: AAW	2: AAA
0	$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$	0	0	0

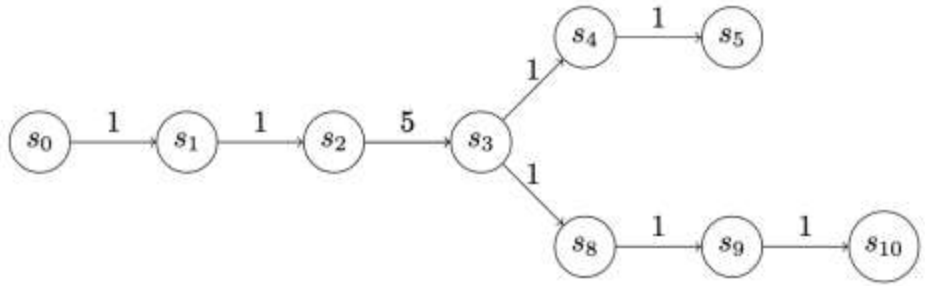
1: WWW	1: WWD	1: WDW	1: WDD	1: DWW	1: DWD	1: DDW	1: DDD
$\frac{3}{11}$	$\frac{1}{33}$	$\frac{1}{33}$	0	$\frac{1}{33}$	0	0	$\frac{7}{11}$



Attack Graphs

Chains representing multiple potential attacker behaviors

- Midpoint between state chains and attack graphs (all possible avenues of attack)
- Transition point (mapping into attacker behavior)



Simulation

Simulation is a way to model activities that are difficult to analyze exhaustively.

Simulation can model behavior that violates assumptions of rationality.

```
# Information common to all games of this type
_GAME_TYPE = pyspiel.GameType(
    short_name="chain_game_v0",
    long_name="chain game version 0",
    dynamics=pyspiel.GameType.Dynamics.SIMULTANEOUS,
    chance_mode=pyspiel.GameType.ChanceMode.DETERMINISTIC,
    information=pyspiel.GameType.Information.IMPERFECT_INFORMATION,
    utility=pyspiel.GameType.Utility.ZERO_SUM,
    # The other option here is REWARDS, which supports model-based
    # Markov decision processes. (See spiel.h)
    reward_model=pyspiel.GameType.RewardModel.TERMINAL,
    # Note again: num_players doesn't count Chance
    max_num_players=len(Players),
    min_num_players=len(Players),
    provides_information_state_string=False,
    provides_information_state_tensor=False,
    provides_observation_string=False,
    provides_observation_tensor=False,
    provides_factored_observation_string=False,
    # We can worry about parameters later
    parameter_specification={},
)
```

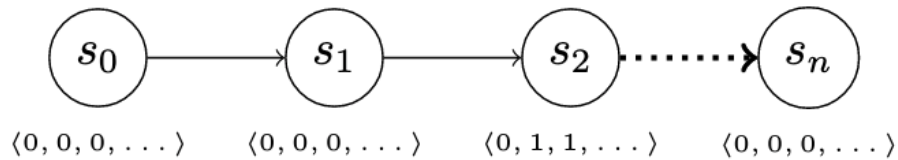
Game Specification with OpenSpiel [4]

[4] Deepmind. OpenSpiel: A Framework for Reinforcement Learning in Games. https://github.com/deepmind/open_spiel

Chain Game Versions 2..n

Evidence

- Vector associated with state
  - Indicators of attacker activity
- Camouflaged and noisy attacks change the evidence vector differently.
- Defender detect action collects evidence from vector.
- Defender remediate action(s) stop attacker advances/evicts the attacker.



Pivot to the problem domain: Map elements of the evidence vector to detections of attack activity on systems.