SaTC: Game Theory and Distributed Systems

Mridu Nanda

June 2022

6/1/22 Session

Motivation

1. Lack of understanding of the impact of cascading attacks or mitigation on the resilience of the overall system.
2. Legacy nature and budget constraints makes completely re-architecting and strengthening impossible
3. Need to make rational decisions to strengthen parts of the system that takes into account the risks and inter dependencies among the assets.

Open questions?

1. Many existing game theory systems are static, but we require dynamic models to represent systems. How can we use game theory?
2. Can we use behavioral economics where human biases are taking into account in decision making? How can that be incorporated into traditional game theory?

Current Foundations: Game Theory

1. Common game-theoretic model has two players, where a single attacker attempts to compromise a system controlled by a single defender.
2. Game theory models have also been used to distributed denial of service attacks
3. Recent works have used game theory models to model inter dependencies between multiple defenders and effect of behavioral bias on security resource allocation. Refer to recent work, published last week (S&P).
4. Such models have also been used to study critical infrastructure security, censorship-resilient proxy distribution, and protecting networks from cascade attacks.

Current Foundations: Distributed System Security

1. Three categories: detection, diagnosis, containment and response.
2. From the attack-centric perspective, we can consider attacks that are: control plane attacks, data plane attacks, and distributed ML system attacks.

Current Foundations: Systems

1. Three categories: sandbox isolation, prevention of malware distribution, and infrastructure security.

Short Term Challenges

1. Distinguish game theory from machine learning. Claim: It is more important to look at data than at the specific algorithms. We need to learn the landscape via data.
2. How to model dynamic environments? Secure distributed systems are changing (nodes and networks both).
3. How to align incentives of different stakeholder managing interdependent systems?
4. How to leverage mechanism design?
5. How could we use game theory to model a block chain den-centralized finance system.
6. Important to quantify your security guarantees by parameterize the amount of resources available to the attackers and defenders.
7. How big a challenge is data collection and scalability. One opinion is that data wrangling is very consuming: collecting data, getting access to data, etc, is very difficult to obtain.
8. Can we model vulnerabilities in SGX (e.g., side channels) as game theoretic models? For example, would Intel be a player whose actions we cannot control. In general, we want to create models where we cannot control all parts of the ecosystem.

Long Term Challenges

1. Can we secure heterogeneous systems with some nodes possibly resource constrained?
2. Can we build secure distributed applications with partially trusted data sources?
3. Can we integrate game theory and machine learning to secure distributed systems?
4. Optimization challenges?
5. Can we build models and continually verify that the model is producing reasonable results? In contrast, we currently verify in “batches”, and then iterate on the model.
6. Can we create meta-learners to automatically figure out which models best fits our task at hand?

Questions and Comments from Audience

1. Canonical example of interdependent systems: Web e-commerce based systems. For example, when you click on a Netflix video, there are many services that are actually being used under the hood. Some other examples include power grids, and transportation systems.
2. How do we model learning in the behavioral game theory models? Learning can be in the context of humans, or also autonomous agents (e.g., machine learning models). Should we also consider how much data we can access in order to learn?
3. Can we take any lessons from the pandemic in modeling human behaviors? For example, mask adoption rates? What rules work, and what rules do not work? Can we apply epidemic theory to distributed systems, generally?
4. How to realistically evaluate the defender and attacker interaction for security? What is the correct formulation for the attacker’s payoff? How do we estimate what resources the attacker can access? Answers to these questions become muddy in the real systems.
5. In contrast, we can also consider mechanism design, so that agents play according the most desirable rules. How do we involve policy makers to help enforce desirable rules? How do we minimize the amount of supervision we need in order to build systems?

6. Security work often assumes a single (though possibly distributed) adversary. However, this does not map in the real world, since we may have multiple adversaries that have different goals, but are adversaries, nonetheless. Hyper game theory deals with imperfect information sharing (e.g., consider an information delay). We can use this to model different attackers in a single system. Multiple attackers could also be captured by a vector of payoffs, and then looking for an equilibrium point.

7. There is a knowledge asymmetry between attacker and defender, which widens as the attacker learns more from interacting with the system. How can the attacker leverage the knowledge asymmetry?

8. How to convert specific systems into the game theoretic formulation that is tractable? We can make rigorous approximations to make game theoretic models tractable (e.g., mean field theory)

9. In large automatic systems, can we take "failure" as an analog to irrationality in these systems? To model failure, we should consider stochastic models: we succeed with probability. p and fail with $1 - p$ (which traditional models, do not take in account).

10. Security critical jobs should not be put in containers, since there is a large TCB (even though this gives us some performance benefits) We should rely on hardware level virtualization instead (e.g., Intel-VTX). Even virtual machines are better than containers. Hot take: we should not always optimize for performance!

11. Most of this discussion has revolved around software; however, how does hardware fit into these game theory models? While software lives on forever, hardware does not. For example, hardware will stop working over time. In other words, we need to bring in dynamic (in this case, time dependent) behavior to model interactions. In general, game theory is at a higher level of abstraction than the software/hardware distinction.

Favorite Phrases

1. "I'm not an expert in game theory, but ... "