Defensive Deception in Enterprise Networks

Mu Zhu

North Carolina State University
mzhu5@ncsu.edu

November 16, 2023

Contents

Education

- Ph.D. in Computer Science: North Carolina State University
- Master in Computer Engineering: University of Delaware
- Dual Bachelor's Degree in Electronic Engineering and Finance: Zhengzhou University

Publications

- Wan, Z., Cho, J.H., Zhu, M., Anwar, A.H., Kamhoua, C. and Singh, M., 2023, October. Deception in Drone Surveillance Missions: Strategic vs. Learning Approaches. In Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (pp. 382-387).
- Wan, Z., Cho, J.H., Zhu, M., Anwar, A.H., Kamhoua, C. and Singh, M.P., 2023. Resisting Multiple Advanced Persistent Threats via Hypergame-Theoretic Defensive Deception. IEEE Transactions on Network and Service Management.
- Anwar, A.H., Zhu, M., Wan, Z., Cho, J.H., Kamhoua, C.A. and Singh, M.P., 2022, December. Honeypot-based cyber deception against malicious reconnaissance via hypergame theory. In IEEE Global Communications Conference (pp. 3393-3398). IEEE.
- Zhu, M., Granados, A., Sharmin, N., Anjum, I., Ortiz, A., Kiekintveld, C., Enck, W. and Singh, M.P., 2022. Optimizing honey traffic using game theory and adversarial learning. In Cyber Deception: Techniques, Strategies, and Human Aspects (pp. 97-124). Cham: Springer International Publishing.
- Zhu, M. and Singh, M.P., 2022. Mee: Adaptive Honeyfile System for Insider Attacker Detection. In Cyber Deception: Techniques, Strategies, and Human Aspects (pp. 125-143). Cham: Springer International Publishing.
- Wan, Z., Cho, J.H., Zhu, M., Anwar, A.H., Kamhoua, C.A. and Singh, M.P., 2021. Foureye: Defensive deception against advanced persistent threats via hypergame theory. IEEE Transactions on Network and Service Management, 19(1), pp.112-129.
- Zhu, M., Anwar, A.H., Wan, Z., Cho, J.H., Kamhoua, C.A. and Singh, M.P., 2021. A survey of defensive deception: Approaches using game theory and machine learning. IEEE Communications Surveys and Tutorials, 23(4), pp.2460-2493.
- Anjum, I., Zhu, M., Polinsky, I., Enck, W., Reiter, M.K. and Singh, M.P., 2021, April. Role-based deception in enterprise networks. In Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy (pp. 65-76).
- Cho, J.H., Zhu, M. and Singh, M., 2019. Modeling and analysis of deception games based on hypergame theory. Autonomous Cyber Deception: Reasoning, Adaptive Planning, and Evaluation of HoneyThings, pp.49-74.

Introduction

Defensive Deception

Defensive Deception leverages **false information** to confuse, mislead, or lure the attacker.

Defensive Deception vs. Traditional Defensive Technologies

- Traditional cybersecurity: focuses on attacker actions
- Defensive deception: focuses on anticipating such actions

Objectives: Asset protection; Attack detection

Benefits and Limitations of Deception

Advantages:

- Cost-effective security scheme
- In-depth understanding threats by participating attack processing
- High deployability

Disadvantages:

- Overhead
- Disturbing legitimate user

Main Concerns

Honeyfile

Crafted decoy documents

- Research Directions:
  - Content Generation: e.g., NLP
  - Deployment: e.g., placement and number
- Benefits:
  - Simple deployment and maintaining
  - Effective detecting stealthy attack (e.g., insider attack)
- Limitations:
  - Unnecessary overhead of storage
  - Confusing legitimate user
  - Generating false positive alarm, which disturbs the defender

Main Concerns

Honeypot

Fake hosts to lure attackers

- Benefits:
  - High-interaction honeypot:
    - Sophisticated and difficult to be detected by attackers
    - Can include false information (e.g., honeyfile)
  - Low-interaction honeypot:
    - Low cost
    - Simple to deploy and maintain

- Limitation:
  - High-interaction honeypots: high cost to create and maintain
  - Low-interaction honeypots: Limited false information and easier to be detected by attackers

Main Concerns

Threat: Insider Attacks

- Traitors, who misuse their legitimate credentials; know a lot about the victim's information
- **Masqueraders**, who impersonate a legitimate user: know little about where the victim's valuable information reside

Difference: Knowledge about victim, such as file space

Main Concerns

Threat: Advanced Persistent Threats (APTs)

Meaning: Well-trained attackers who perform multiple-year threats to exfiltrate valuable and sensitive economic, proprietary, or national security information

Cyber-kill chain: Reconnaissance, Delivery, Initial intrusion, Command and control, Lateral movement, Data exfiltration

Considered action space in proposed work:

Reconnaissance: Gather information about the victim to decide whether attack or not.

Compromise: Penetrate a target device

Data Exfiltration: Harvest sensitive data and transfer them to outside (e.g., masqueraders)

Research Processes

- Before the proposal:
  - Mee: Game Theory-Based Adaptive Honeyfile System for Insider Threats
- After the proposal:
  - HoneyMee: Honeyfile System Based on Deep Reinforcement Learning
  - Hypergame-Based Hybrid Honeypot System
  - GAN-Based Honeyflow Generation for Passive Monitoring

Thesis Statement

Machine learning and game theory can enhance the efficiency of defensive deception strategies by helping the defender to more effectively allocate resources and reduce the interference to the system's normal operations.

Research Questions

Improving Defensive Deception Techniques

Caring about legitimate users

- How should the defender increase the deception attraction to the attacker?
- How should the defender effectively allocate resources?
- How should the defender reduce the impact from deception methods (e.g., confusing legitimate users)?
- How to measure the effectiveness of a defensive deception strategy?

Research of Honeyfiles: Mee and HoneyMee

How to Enhance the Current Honeyfile System

The defender can:

- Adjust the number of honeyfiles by risk assessment
- Differentiate honeyfile alarms
- Analyze suspicious behaviors across the network
- Make decision based on risk level

Mee and HoneyMee:

- Decentralized deployment: deploys honeyfiles as a way to detect suspected behaviors by any user
- Centralized control: analyzes suspicious behavior across the network to determine the number and placement of honeyfiles for each device

Threat Model: Masquerader

Assumptions about the attacker:

- Has knowledge of the users' roles, e.g., via reconnaissance
- Has ability to infiltrate any connected device
- Is unfamiliar with the file system on a compromised device
- Knows of the existence of honeyfile system, but cannot distinguish between honeyfiles and real files
- Has clear target device to search for valuable files

In one compromised device, the attacker may obtain three results:

Success: Viewing or transferring the valuable files

Failure: Not finding valuable files, i.e., wasted effort

Loss: The defender cleans or replaces the compromised device

Legitimate Users and Insider Attacker

Users:

- Familiar with file system, e.g., lower probability to touch honeyfiles
- Open, but no transfer or modify

Attackers:

- Unfamiliar with file system, e.g., higher probability to touch honeyfiles
- Open, modify, transfer honeyfiles
- Attacking devices with tendency

Sensitivity, Seriousness, and Risk

To assist the defender to choose actions

File sensitivity: How valuable a honeyfile is for the defender

Action seriousness: How much of a security threat the action is

- Weak: Open or close a honeyfile
- Strong: Edit, transfer, or zip or tar

Group of hosts:

- Groups: Based on organizational roles
- Group security level: Represents a group's security situation
- Update security estimate: Proportional to file sensitivity and action seriousness

Architecture

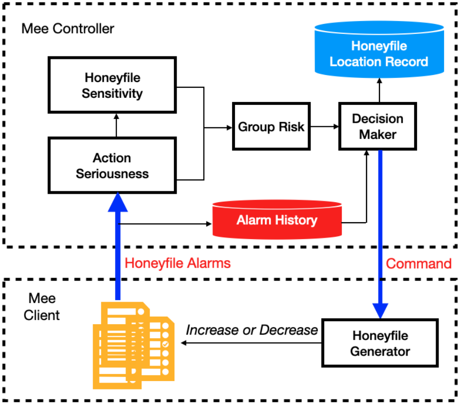
Decentralized deployment with centralized control

Client:

- Generate and remove honeyfiles
- Detect file access on honeyfile and send alarms to Mee controller

Controller:

- Analyze honeyfile alarms
- Instruct clients to adjust the number of honeyfiles in its device

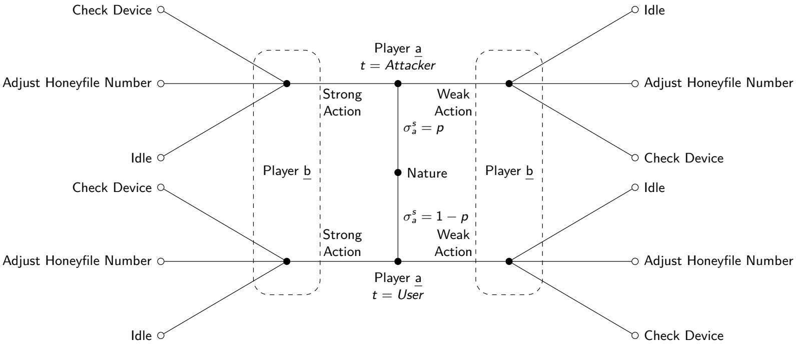


Mee: Bayesian Game-based Honeyfile System

1: From nature, Player a obtains type (attacker or user) as its private information

2: A honeyfile alarm represents an observation of the player b

3: The player b chooses an action based on a received message and its beliefs



HoneyMee: DRL-Based Honeyfile Research

Motivation

Limitation of Mee

- Simple scenario
- Limited number of players at one time slot

Continue to have:

- Mee Structure: Controller and client
- Group and group risk level
- File sensitivity, action seriousness . . .

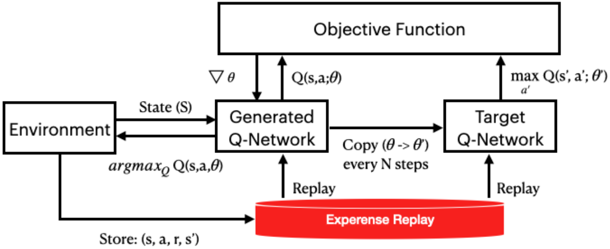
What is New?

- Complete scenario: More devices, active users and insider threats
- More details of environment
- Deep reinforcement learning: Model multiple users and attackers at one time slot

Deep Reinforcement Learning

Agent (state, action space, observation); Environment; Reward function

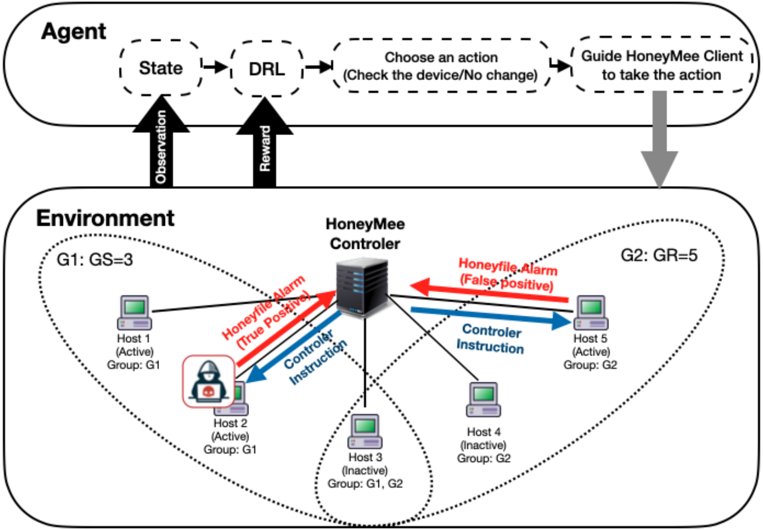
- Neural Network: Using neural networks to approximate the Q-function
- Target Network: Employing a target network that delays the update of target values to increase learning stability
- Experience Replay: Sampling a random minibatch of transitions from experience replay buffer as training data



Environment

- Device: $\langle$Condition, Security Level, Importance, Groups$\rangle$
- Active User:
  - Action Space: Login, Search, Open a File, Edit a File
  - Being Familiar with File System
- Insider Attacker:
  - Action Space: Infiltration, Search, Open a File, Edit a File
  - No Knowledge of File System

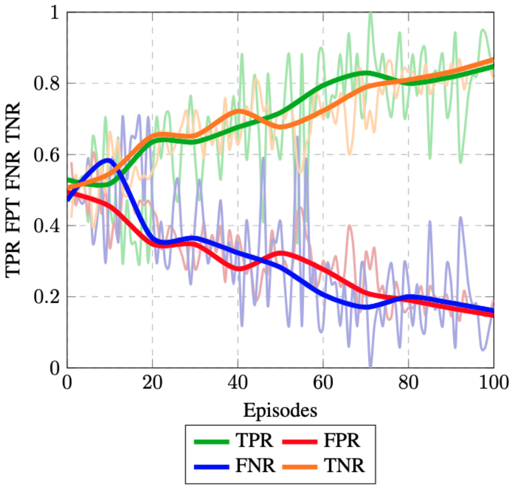
HoneyMee and DRL Scheme

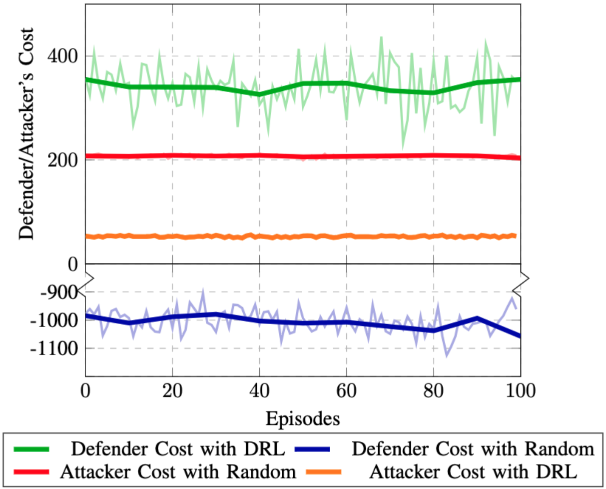


Training Settings and Results

Model Name	Dense Layer	Batch Size	Max Reward	Min Reward	Average Reward
M0	[32, 32]	16	1355	-515	299
M1	[32,32]	32	1370	-150	806
M2	[32,32,16]	32	1990	1000	1552.25
M3	[32,32,32]	32	1250	-50	-169.85
M4	**[128,64,32]**	**32**	**2935**	**1230**	**2283.45**
M5	[128,64,32,8]	32	2820	1775	2176.5

HoneyMee and DRL Scheme





Hypergame-based Hybrid Honeypot System

Deception Scheme

- High-interaction honeypot:
  - Includes vulnerable OS and applications
  - Mimics actual hosts
- Low-interaction honeypot:
  - Crafted TCP-based network flows
  - Transfer between honeypots

Models

Network Model

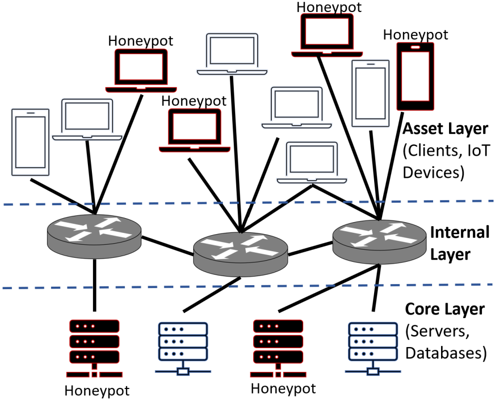
- Asset Layer: Low-value nodes (e.g., IoT devices and laptops)
- Internal Layer: Routers, switches, and nodes between the asset and core layers
- Core Layer: High-value nodes (e.g., database or web servers)

Defender Model

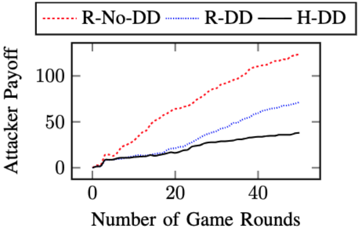
- Deploys low-interaction honeypots
- Deploys high-interaction honeypots

Attacker Model

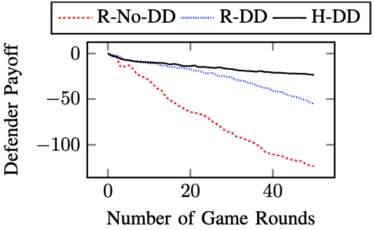
- Passive monitoring
- Active probing



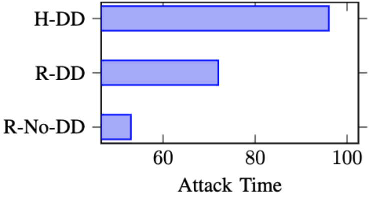
Results



(a) Attacker's payoffs under various schemes with respect to the number of game rounds.



(b) Defender's payoffs under various schemes with respect to the number of game rounds.



(c) Total attack time with different schemes.

GAN-Based Honey Traffic Generation

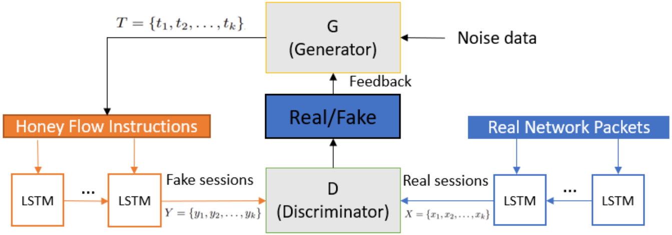
GAN

Generator (Honeypot)

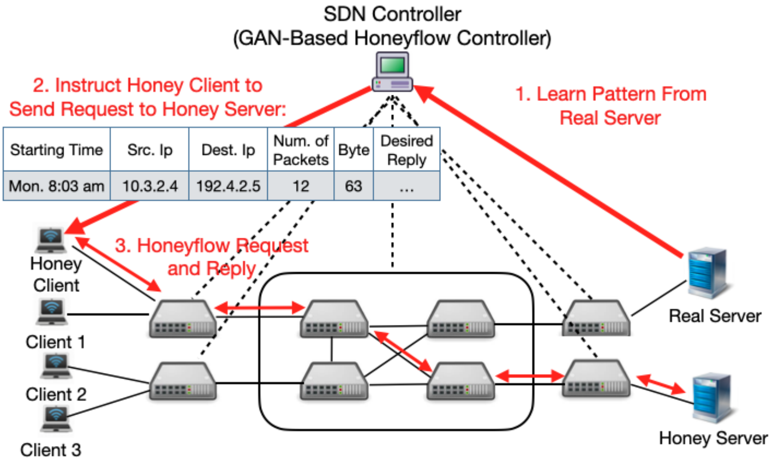
- Learning from actual server
- Generating craft fake traffic

Discriminator (Attacker)

- Learning from actual server
- Distinguishing real data from fake traffic
- Selecting device to compromise



Deception Scheme and Scenario



Conclusion

Research Question 1

How should the defender increase the deception attraction to the attacker?

Solutions: Mee, HoneyMee, and GAN-based honeyflow

- Mee and HoneyMee: adjusting the number of honeyfiles based on devices' situation
- Mee and HoneyMee: using file sensitivity to measure the appeal of honeyfiles
- GAN-based honeyflow: mimicking regular host to lure attackers

Research Question 2

How should the defender effectively allocate resources?

Solutions: Mee, HoneyMee, and hypergame-based hybrid honeypots

- Mee and HoneyMee: using groups and their associated group values to differentiate the importance of devices to the defender
- Mee and HoneyMee: Altering the quantity of honeyfiles as required
- Hybergame-based Honeypot: allocating the resources of low and high-interaction honeypot deployment

Research Question 3

How should the defender reduce the impact from deception methods (e.g., confusing legitimate users)?

Solutions: Mee and HoneyMee

- Mee and HoneyMee: Altering the quantity of honeyfiles to decrease the impact on users
- Mee and HoneyMee: Examining honeyfile alerts to reduce the number of false positives

Research Question 4

How to measure the effectiveness of a defensive deception strategy?

All research considering the metrics of measurement:

- Defender Payoff
- Attacker Payoff
- Accuracy: true/false positive rate (ROC)

Appendix

Mee: Simulation and Evaluation

Test 1: Mee's performance

- Group risk level updating
- Number of honeyfiles in each group

Test 2: Comparison between Mee and traditional honeyfile system

- Tradition Honeyfile System: With different fixed number of honeyfiles in each device
- Mee: Dynamic number of honeyfiles in each device

Test 3: Comparison between Mee and traditional honeyfile system

- With different number of attackers

Metrics of Measurement:

- Defender Payoff
- Attacker Payoff
- Accuracy: True/false positive rate (ROC)

Group Risk Update and Classification

Group Risk Update

$$\triangle \text{risk}_{\text{group}}(\text{honeyfile}, \text{action}) = \frac{\text{sensitivity}_{\text{honeyfile}} * \text{seriousness}_{\text{action}} * \text{importance}_{\text{group}}}{\text{number}_{\text{honeyfiles}}(\text{group})}$$

Group Classification

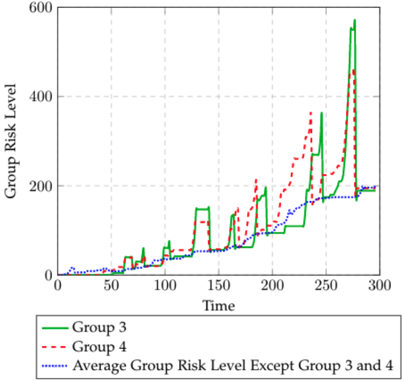
$$R_{-i} = \frac{\sum_{j \neq i} R_j}{Number\ of\ Groups - 1}$$

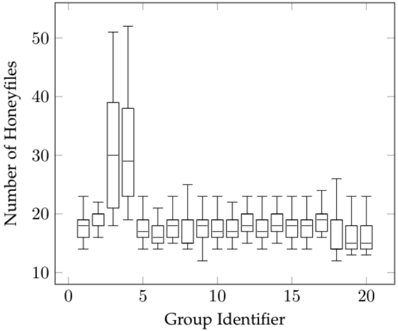
where R_{-i} represents the average group risk level except group i

$$Classification = \begin{cases} \text{Dangerous} & \text{if} \quad R_i > R_{-i} * 2 \\ \text{Medium} & \text{if} \quad R_{-i} < R_i < R_{-i} * 2 \\ \text{Safe} & \text{if} \quad R_i < R_{-i} \end{cases}$$

Test 1: Mee's Performances

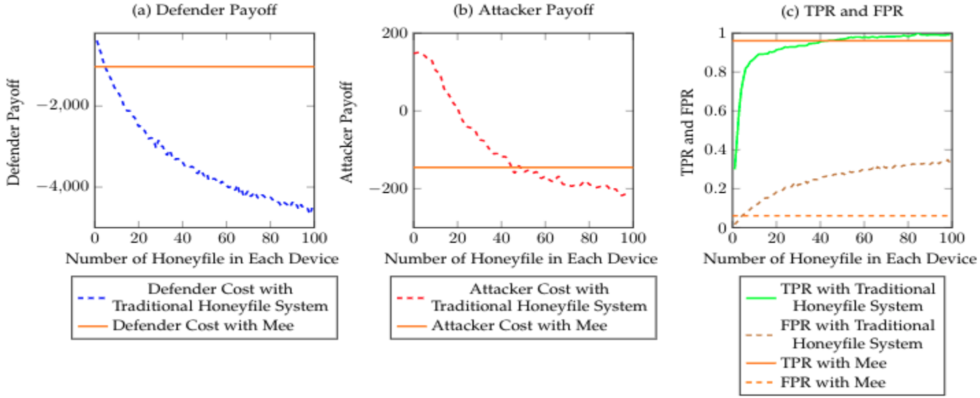
- Mee seeks to optimize resources while reducing false positives
  - Maintains group risk level
  - Adjusts the numbers of honeyfiles in various devices accordingly





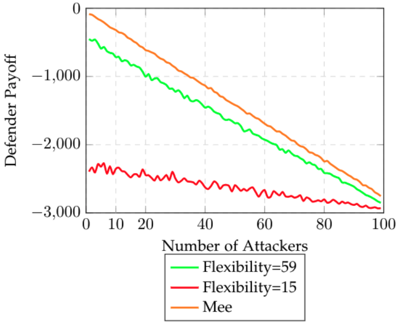
Test 2: Comparison between Mee and traditional honeyfile system

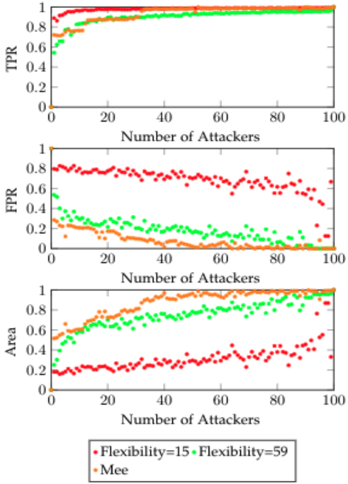
- Traditional Honeyfile System: the number of honeyfiles in one device is change from 0 to 100



Test 3: Comparison between Mee and traditional honeyfile system

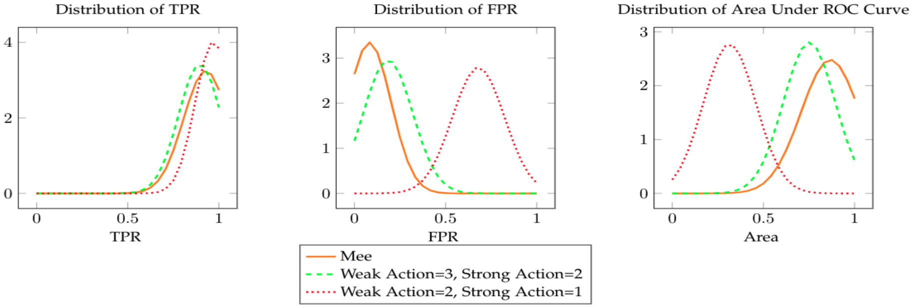
- Number of attackers is changed from 1 to 100
- Area under ROC Curve $= \text{TPR} * (1 - \text{FPR})$





Detection Improvement: Effect Size

Cohen's d values for stated pairs	True Positive Rate	False Positive Rate	Area
(Weak Action = 2, Strong Action = 1) and (Weak Action = 3, Strong Action = 2)	0.28	3.57	3.05
(Weak Action = 2, Strong Action = 1) and Mee	0.38	4.55	3.62
(Weak Action = 3, Strong Action = 2) and Mee	0.70	0.80	0.76



Hypergame-based Hybrid Honeypot System

Equations

The defender's utility when taking low-interaction honeypots

$$u_d(a_d, a_a) = \Big(\sum_{i \in a_a} -[r \cdot \hat{v}(i)] \Big) - c_d \cdot a_d + c_a, \hat{v}(i) = v(i) - W^{a_d}(i) \tag{1}$$

The defender's utility when taking high-interaction honeypots

$$u_d(a_d, a_a) = \Big(\sum_{i \in a_a} R \cdot v(i) \mathbb{1}^{\Gamma}_{\{i \in a_d\}} \Big) - C_d \cdot |a_d| + C_a \tag{2}$$

Hypergame-based Hybrid Honeypot System

Equations

Uncertainty of the defender

$$g_d(j) = \exp(-\frac{j}{\mu}) \tag{3}$$

Uncertainty of the attacker

$$g_a(\gamma, \Gamma) = \exp(-\frac{\gamma + \Gamma}{2}) \tag{4}$$

Hypergame-based Hybrid Honeypot System

Equations

The defender's HEU

$$DHEU(a_d, g_d) = (1 - g_d) \cdot EU\left(a_d; \bar{\mathbf{a}}\right) + g_D \cdot EU(a_d; \mathrm{a}_w) \tag{5}$$

The attacker's HEU

$$AHEU(a_a, g_a) = (1 - g_a) \cdot EU\left(a_a; \bar{\mathbf{d}}\right) + g_a \cdot EU(a_a; \mathrm{a}_w) \tag{6}$$

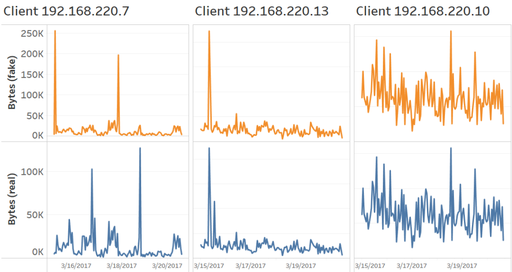
GAN-Based Honey Traffic Generation

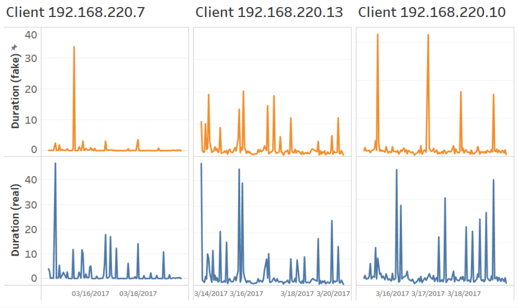
Data Set and Features

Dataset: CIDDS-001 (includes flow-based network packets represented with network attributes)

Attribute	Type	Example
data first seen	timestamp	2018-03-13
duration	continuous	0.12
transport protocol	categorical	TCP
source IP address	categorical	192.168.100.5
source port	categorical	52128
destination IP address	categorical	8.8.8.8
destination IP port	categorical	80
bytes	numeric	2391
packets	numeric	12
TCP flags	binar/categorical	.A..S.

Results


Client 192.168.220.7
Client 192.168.220.13
Client 192.168.220.10
Bytes (fake)
250K
200K
150K
100K
50K
0K
Bytes (real)
100K
50K
0K
3/16/2017
3/18/2017
3/20/2017
3/15/2017
3/17/2017
3/19/2017
3/15/2017
3/17/2017
3/19/2017


Client 192.168.220.7
Client 192.168.220.13
Client 192.168.220.10
Duration (fake)
40
30
20
10
0
Duration (real)
40
30
20
10
0
03/16/2017
03/18/2017
3/14/2017
3/16/2017
3/18/2017
3/20/2017
3/16/2017
3/17/2017
3/18/2017

Acknowledgments

The Army Research Office supported this work under award W911NF-17-1-0370. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes without standing any copyright notation.

Thank You

Mu Zhu (mzhu5@ncsu.edu)