A Game between Adversary and AI Scientist

Satnam, Arunabha, Deepak, Waseem, Nirmesh, Santosh, Balamurali, Narayana
Acalvio Technologies

NULLCON
March 3, 2018
Who am I

- At Acalvio from **Day 1**
- 15+ Years in DS, ML, AI
- General Motors, Samsung Research, CA Technologies
- Author- Patents, Tech Pubs and Tech Talks
- Rock Climber
Outline

• Define the Problem

• A possible solution - research work

• Demos

• Under the hood
Problem
Can we play a game with adversary?

Can we engage with adversary?
Is adversary visible to defender?

- Extensive reconnaissance of target and defender

- Using the same tools and techniques as defender
Is he a “Returning” Adversary?

—> Compare Tools, Tactics and Procedures (TTPs)
InfoSec Game: Assumptions

- Unlike Chess, cyber game has infinite state space

→ Use Mitre ATT@CK model to define the state space
Defender’s Tools are at Perimeter

Need new tools to detect adversary INSIDE the NETWORK
Deceptions in Enterprise

- **Deceptions (D)**
- Emulations of Hosts, Applications, Database Servers, etc.
- Real VM Hosts, Applications, etc.
- Browser Cookies, Registry entries, etc.
- Vulnerability in OS/Application, Shares, etc.
AI Engine

- HIDS Log
  - Process
  - Registry
  - Bro
  - ...

AI Engine
Game: Demos
1: Recon - nmap

**Adversary**

- Adversary performs recon and nmap to find out to the neighbourhood

**Defender**

- Defender detects it and provides a few RDP credentials on the endpoints

Demo>>
2: Obfuscated PowerShell Script

- Adversary obfuscates PowerShell attack and executes in another host
- Defender detects obfuscated PowerShell commands

Demo>>
3: Credentials Dump using PowerSploit and Mimikatz

**Adversary**
- Attacker dumps credentials using PowerSploit and Mimikatz

**Defender**
- Defender detects PowerSploit and Mimikatz activities

Demo>>
Adversary

- Adversary uses DNS Tunnel using DNSCat2 to exfiltrate the credentials

Defender

- Defender detects the DNS tunnel using AI
Under the Hood
High Interaction AI Engine

Adversary

HIDS Log
- Process
- Registry
- Winevent

HISH AI Engines
- Summarisation
- Powershell Log Analyser
- DNS Tunnel Detector
HISH AI: HIDS Log Summarisation

- Summarise attacker’s activities
  - New services, processes, tasks and changes etc.
  - File system changes, registry entries, etc.
  - Shell commands, Windows event and authentication logs etc.

Summarised Notables

14:46:07 remote interactive login with username alex
14:47:20 Successful auto update of third
14:49:14 svchost started UsoClient using: UsoClient StartScan
14:50:49 wrote or created 158 files in C:\Users\alex\Downloads\
14:52:09 explorer started cmd
14:54:31 launched interactive session to local admin account
HISH-AI: Summarisation Engine

Raw Log Preprocessing → Rule & Baseline Based Filtering → Process-based Summarisation

- Domain Knowledge
- Learned Baseline
- Rules
- Baseline

<table>
<thead>
<tr>
<th>Attack Scenarios</th>
<th>Input Logs</th>
<th>Output Notables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident 1</td>
<td>60K</td>
<td>16</td>
</tr>
<tr>
<td>Incident 2</td>
<td>6K</td>
<td>5</td>
</tr>
<tr>
<td>Incident 3</td>
<td>70K</td>
<td>6</td>
</tr>
</tbody>
</table>
HISH-AI: PowerShell Log Engine

PreProcessing

Command N-GRAM

Obfuscation Detection Model

Obfuscation Prediction

Tactic Prediction

(Privilege Escalation, Lateral Movement, Exfiltration)

PowerShell Logs

Character N-GRAM

Tactic Detection Model

Classifier

Tensorflow ANN

Feature $f_1$

Feature $f_2$

Malicious scripts

Unknown scripts

Benign scripts

Classification
HISH-AI: Data exfiltration using DNS Tunnel

Enterprise Network

dns tunnel client

dns query: data.dnstunnel.com

web traffic

dns traffic

dns tunnel server (dnstunnel.com)

DNS Tunnelling Tools
Iodine, dnscat2, Ozyman
DNS tunnel detection output:
- IP and domain of tunnelling server: dnstunnel.com
- tunnel start time: 26-02-2018 19:43:37
- tunnel end time: 26-02-2018 19:53:37
Game Theory
Formally Defining A Game

Defining Game - The Normal Form

Finite 2-person normal form game: \(<N, A, u>:\n
- Players: \(N=\{\text{Adversary, Defender}\}\) is a finite set of 2 players, indexed by \(i\)
- Action set for player \(i\) — \(A_i\)
  \[a=\{a_1, \ldots, a_n\}\]
- Utility function or Payoff function for player \(i\): \(u_i\)
  \[u=(u_1, \ldots, u_n)\] is a profile of utility functions
# InfoSec Game

<table>
<thead>
<tr>
<th>Defender</th>
<th>Adversary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Carry out attack</td>
</tr>
<tr>
<td>Allow the attack</td>
<td>1,2</td>
</tr>
<tr>
<td>Block the adversary</td>
<td>2,2</td>
</tr>
</tbody>
</table>

- "Row" player is Defender, "column" player is Adversary
- Too simplistic
- How to scale it for the real world?
- How do we learn in real time?
Model as Reinforcement Learning Problem

- Break the problem into Subproblems and learn in real-time
- Model it as Reinforcement Learning Problem
Summary

• Playing a game needs “Visibility” of the adversary

• Need to surface signal in low SNR

• Fusion of Deception+AI gives a way to engage with the adversary
Questions?

@satnam74s