Enabling Dynamic Network Access Control with Anomaly-based IDS and SDN

Hongda Li, Feng Wei, and Hongxin Hu

SDN-NFV Security 2019
Network Access Control with SDN

Access Control Policies

How to generate new ACP?

FlowGuard [HotSDN’14]
Dynamic Firewall [RAID’15]
Virtual Firewall [NDSS’17]

Unknown vulnerabilities
Zero-day security threat

Anomaly
Existing Anomaly-based IDS

Uncover novel security threats

Obscure outcome

A.I. Machine Learning System

User Profile

Day 1

Day 2

Day 3

Unusual (anomalous) behavior

Semantic Gap

Access Control Policies
Machine Learning Model Explanation

Input

Predictor

Outcome

CAT

DOG

Explanation
Explanation Mechanisms

**Whitebox**

**Blackbox**

**Global Explanation**

**Local Explanation**
Local & Blackbox Explanation

1. Local Approximation
2. Explanation Logic

Why $x$ is predicted as circle?
Approach Overview

SDN Switch → Mirrored Traffic → Anomaly-based IDS → SDN Flow Rule → SDN Controller → Access Control Policy → Outcome Explanation → Outcome Generator
AIDS Outcome Explanation

1. Local Approximation
2. Explanation Logic

\[ l(x) = \alpha x + \epsilon \]

\[ L(l(x), y) = \sum_{i=1}^{N} \| \alpha x_i - y_i \| \]

Feature Importance

\( x: (\text{duration}, \text{proto}_\text{type}, \text{service}, \text{flag}, \text{src}_\text{byte}, \text{dst}_\text{byte}, \ldots) \)

\( \text{FI: (97, 96, 99, 100, 95, 98, \ldots)} \)
Access Control Policy Generation

<filers, actions>

*Selects network entities*  *Defines action to take*

Networks; Hosts; Connections; Flows; Packets; Combination of above;

Allow; Deny; Redirect; Quarantine; Mirror; ...

\( \mathbf{x} : (\text{duration, proto\_type, service, flag, src\_byte, dst\_byte, ... }) \)

\( \mathbf{FI} : (97, 96, 99, 100, 95, 94, ... ) \)

Explanation
Case Study: AIDS

- Recurrent Neural Network (RNN)
  - Detect across multiple records
- NSL-KDD dataset
  - 41 raw feature
- Keras + TensorFlow for implementation
Case Study: Outcome Explanation

- Choose *Neptune attack* in dataset
  - Extensive SYN error or SYN rejection
- Two records labeled as Neptune attack

Record1: (0, tcp, private, S0, ..., 255, 20, 0.08, 0.07, 0, 0, 1, 1, 0, 0)

Record2: (0, tcp, imap4, REJ, ..., 255, 17, 0.07, 0.07, 0, 0, 0, 0, 1, 1)

**Explanation (Feature Importance)**

<table>
<thead>
<tr>
<th>Record</th>
<th>Duration</th>
<th>Proto.</th>
<th>Service</th>
<th>Flag</th>
<th>...</th>
<th>Dst SYN E.</th>
<th>Dst SYN Srv.E.</th>
<th>Dst Rej E.</th>
<th>Dst Rej Srv. E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record1</td>
<td>0</td>
<td>tcp</td>
<td>private</td>
<td>S0</td>
<td>...</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Record2</td>
<td>0</td>
<td>tcp</td>
<td>imap4</td>
<td>REJ</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Percentage of SYN Error

Percentage of Rejection Error
Case Study: Policy Generation

Outcome Explanation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Record1</td>
<td>0</td>
<td>tcp</td>
<td>private</td>
<td>SO</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Record2</td>
<td>0</td>
<td>tcp</td>
<td>imap4</td>
<td>REJ</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<filters={ip_proto=tcp, tcp_flags=syn, sip=192.168.1.2, dip=192.168.1.3}, actions=(drop)>

Access Control Policy
Conclusion and Future Work

- **Conclusion**
  - Explained the outcome of anomaly-based IDS
  - Generated network access control policy according to the explanation

- **Future work**
  - Better explanation that handles decency among records
  - Policy generation process formalization
  - More evaluation on realistic traffic and attacks
Q & A

Hongda Li (hongdal@clemson.edu)

Thank you!