Anomaly Detection on Industrial Control Systems

Mohammad Reza Norouzian
Technische Universität München
Fakultät für Informatik
Lehrstuhl für IT Sicherheit
12. July 2018
Industrial Control System in the World
What type of ICS products are vulnerable:

- Group WAN
- Production network
- Supervision network / SCADA
  - Supervision consoles
  - Maintenance laptops
  - Data Historian / Scada server

Corporate network
- Corporate IT
- ERP server
- Production management

ICS
- PLC
- RTUs
- Wireless industrial networks
- PLCs

Mohammad Reza Norouzian (TUM) | Chair for IT Security
Siemens ICS Products

- Target – Siemens S7-300/400/1200 PLC
- S7 Packet

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>magic 0x32</td>
<td>pdu-type</td>
<td>reserved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>request id</td>
<td>parameters length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>data length</td>
<td>error code (only for pdu 0x03)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>parameters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- PDU-types:
  - 0x01 – Request
  - 0x02 – Acknowledgement
  - 0x03 – Response
  - 0x07 – User Data
Needs – S7 IDS rules!

- Snort rules
- Bro has no rule for S7
- Suricata no rules too!
- Just Modbus signatures

```bash
# Alert on a command that was is via s7-enumerate Redpoint NmapNSE on TCP/102
alert tcp any any -> any 102 (content:"32 07 00 00 00 00 00 00 00 08 00 08"; offset: 0; depth: 10; content:"00 01 12 04 11 44 01 00"; offset: 11; depth: 8; msg:"S7 Enumerate Redpoint NSE Request CPU Function Read S2L attempt"; id:1111301; priority:3)
# Alert on a command that was is via s7-enumerate Redpoint NmapNSE on TCP/102 from Non Authorized Hosts
alert tcp $S7_CLIENT any $S7_SERVER 102 (content:"32 07 00 00 00 00 00 00 00 08 00 08"; offset: 0; depth: 10; content:"00 01 12 04 11 44 01 60"; offset: 11; depth: 8; msg:"S7 Enumerate Redpoint NSE Request CPU Function Read S2L attempt From Non Authorized Host"; id:1111302; priority:1)
```

```bash
alert modbus !$MODBUS_CLIENT any $MODBUS_SERVER 502 (modbus: function 0x05; msg:"Modbus Write Single Coil First"; id:11; xbits:set,modbus,track ip_src)
alert modbus !$MODBUS_CLIENT any $MODBUS_SERVER 502 (modbus: function 0x07; msg:"Modbus Read Exception After Write"; id:12; xbits:isset,modbus,track ip_src)
```
Network Attacks against ICS

- Reconnaissance
- Authentication bypass
- CPU stop and start
- Brute-force
- Command injection and response
- Denial of service (DoS)
- Memory read and write logic
- Man in the middle (MITM)
- Attacks against PLC firmware
Multi Stage Attack - IUNO Scenario

- Attack ICS devices!
  - Reconnaissance
  - Authentication bypass
  - CPU stop and start (command control)
The state of the art in detecting scanners is surprisingly limited. Existing schemes have difficulties catching all but high-rate scanners and often suffer from significant levels of false positives!

What about the reconnaissance attacks for SCADA world?

- Gathering Information from the PLC with a specific commands!
- Firmware version, Serial number, module name, …
Brute Force and Command Control Attack

- Try to bypass authentication!
- Brute force with dictionary attack
- Try to stop PLC
Problem?

- Anomaly Detection on Industrial Control System (ICS)
  - Classify benign and malicious activities
    - Signature-based (Misuse) detection
    - Anomaly detection using Machine Learning

- Challenges of Using Machine Learning
  - Lack of Training Data
  - Diversity of Network Traffic
  - High Cost of Errors
Our Main Focus and Approach

• Anomaly Detection on ICS
  o Host based
    • Don’t have control on PLCs and field devices
  o Network based
    • More scalable
Industrial Network Traffic Analysis Framework

Machine Learning Anomaly based Framework
ICS Network Traffic Feature Extractor
  o Python and Tshark
  o S7 Communication Protocol, Profinet IO/RT
• Why?
  o Feed features into anomaly detection framework
• Feature Selection!
  o Identifying Intended features that help to classify benign from malicious traffic
  o It can select the best combination of features to increase accuracy and decrease FP/FN
Having Malicious Traffic

- Anomaly Detector Engine Module
- Traffic Analyzer Dissector Module
  - S7 Dissector
  - PROFINET Dissector
- Malicious Traffic Generator
Network Anomaly Detection for ICS Engine (NADICS)
Anomaly Detection Big Picture

Data Collection

Pre-processing of Data

Attribute Selection
- Attribute 1
- Attribute 2
- Attribute 3

Initialization Step

Algorithms

Training Model

Learning Step

Score Model

Predicting Output

Apply Data / Test Data

Applying Step
ML Algorithms Module

CLASSIC MACHINE LEARNING ALGORITHMS

Boosting based
- Gradient Boost
- AdaBoost

Neighbourhood based
- KNN
- LOF

Tree based
- Decision Tree
- Random Forest
- Isolation Forest

Support Vector Machines
- Linear SVM
- SVM
- Linear SVC
- One Class SVM
- C_SVC
- Nu SVC

Stochastic based
- Gaussian Process
- SGD
- Gaussian NB
- Bernoulli NB
- Multinomial NB
NADICS Sample Results

---

# NADICS MACHINE LEARNING ENGINE

---

READING CONFIG FILE... DONE.
IS PREPROCESSED DATA STORED ON DISK? TRUE.
ENCODING DATA SET... DONE in 3.579 seconds.
SAVING PREPROCESSED DATA SET TO DISK... DONE in 1.188 seconds.
TRIANGING SIZE: UNSW_NB15_training-set
TESTING SIZE: UNSW_NB15_testing-set
TRAINING SIZE: 757157
TESTING SIZE: 76270
FEATURES: 39
NORMALIZING DATA SETS... DONE in 0.278 seconds.

Classification algorithm: RandomForest
TRAINING THE MODEL... DONE in 7.6 seconds.
PREDICTING... DONE in 0.641 seconds.
Accuracy Score: 0.967
Classification report:
<table>
<thead>
<tr>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.99</td>
<td>0.94</td>
<td>0.96</td>
</tr>
<tr>
<td>1</td>
<td>0.94</td>
<td>0.99</td>
<td>0.97</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
</tr>
</tbody>
</table>

---

Time training [s] | DecisionTree | RandomForest | SGD | KNeighbors | Linear_SVC |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.8016</td>
<td>0.6131</td>
<td>0.6914</td>
<td>0.7799</td>
<td>0.8675</td>
<td>0.7883</td>
</tr>
<tr>
<td>0.8518</td>
<td>0.6969</td>
<td>0.6667</td>
<td>0.8441</td>
<td>0.8676</td>
<td>0.8169</td>
</tr>
<tr>
<td>0.9563</td>
<td>0.9669</td>
<td>0.7799</td>
<td>0.8674</td>
<td>0.7821</td>
<td></td>
</tr>
<tr>
<td>0.9562</td>
<td>0.9669</td>
<td>0.7799</td>
<td>0.8675</td>
<td>0.7883</td>
<td></td>
</tr>
<tr>
<td>Weighted precision</td>
<td>0.5975</td>
<td>0.9683</td>
<td>0.8441</td>
<td>0.8676</td>
<td>0.8169</td>
</tr>
<tr>
<td>Weighted f1 score</td>
<td>0.5962</td>
<td>0.9669</td>
<td>0.7799</td>
<td>0.8674</td>
<td>0.7821</td>
</tr>
<tr>
<td>Weighted support</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

---

Mohammad Reza Norouzian (TUM) | Chair for IT Security
Dataset Currently in Use

**TRAINING**
157,157 samples

**TESTING**
76,270 samples

**# FEATURES**
39

---

**Normal to Attack Ratio**

- **Training:**
  - Normal [\%]: 80
  - Attack [\%]: 20

- **Testing:**
  - Normal [\%]: 60
  - Attack [\%]: 40
Feature Selection

• Improving accuracy by automatically only selecting relevant features
• Requiring less data
• Reducing complexity of our model
Feature Importance

Feature Importances

src_ttl
dst_ttl
c_state_ttl
dst_load
src_tcp_win_adv_value
c_state_dst
dst_mean_sz
rate
c_state_src
c_state_src_ltm
src_mean_sz
dst_tcp_win_adv_value
c_state一趟
is_sm_ips_ports
rcp_round_trip_time
src_bytes
src_load
ackdat
c_state_ltm
dst_tcp_base_sq_num
src_interpkt_arr_time
src_tcp_base_sq_num
synack
c_state_ltm
c_state_ltm
duration
dst_pkt
src_jitter
dst_jitter
dst_interpkt_arr_time
dst_bytes
src_loss
dst_loss
trans_depth
src_pkt
c_state_http_mthd
res_body_len
isftpLogin
c_state_FS

Implemented Algorithms for Imbalances

**IMBALANCE STABILIZER**

**Oversampling**
- ADASYN
- SMOTE
- Random Over Sampler

**Undersampling**
- AllKNN
- Tomek Links
- One Sided Selection

**Adaptive Boosting Algorithm**

Source: https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/
Future ML Module Architecture
Further Improvements

• Generate more attacks
• Implement deep learning
• Learning the Normality!
Thank You!