“Beyond the ramparts: What artificial intelligence promises for cyber defense”

Presenter: László Tóth, software engineer /researcher
Department of Software Engineering
University of Szeged, Hungary

The Eighth International Conference on Fundamentals and Advances in Software Systems Integration, FASSI 2022
October 16, 2022 to October 20, 2022 - Lisbon, Portugal
About ME

László Tóth
Software engineer/researcher
University of Szeged
Department of Software Engineering
premissa@inf.u-szeged.hu

- Cyber Security
- Deep Learning
- Natural Language Processing
Agenda

- The global cyber threat
  - The evolution of the cyber attacks
  - Attack on the critical infrastructures (Stuxnet, Industroyer)
  - Ransomware attacks (WannaCry, NotPetya)

- The reasons behind the vulnerabilities
  - The human factor
  - Software bugs

- Classic solutions for protection against the cyber threats
  - The onion model
  - Firewalls and Intrusion Detection Systems
  - Static code analysis

- Applying machine learning methods in the cyber defense
  - Vulnerability prediction
  - Detecting vulnerable traffic
  - The vulnerability of the neural networks
The global cyber threat
Reported losses by FBI Internet Crime Complaint Center

LOSSES OF VICTIMS (MILLION USD)
Evolution of cyber attacks

1980s
- Simple local attacks:
  - password cracking
  - virus attack on PCs

1990s
- Attacks from the Internet
  - viruses
  - The Chameleon (polymorphic)
  - hoaxes

2000s
- Exploits
- DoS/DDoS

2010s
- polymorphic attacks
- evasion techniques
- directed attacks

Cyberwarfare
- Stuxnet 2010,
- Havex 2013,
- Industroyer 2016,
- WannaCry 2017,
- NotPetya 2017

2020s
- Sophisticated attacks
  - botnets
  - malicious codes
  - strategic attacks
  - large scale
  - multivector attacks
Changing the targets

1980s individual hosts

2010s industrial systems

Source: Pinterest

Source: Wikipedia under (Licence: CC BY-SA 3.0)
Critical infrastructures

Source: McAndrew Ian, Vishnevskaya Elena, Johnson, Michael: Artificial Intelligence in the Aviation Manufacturing Process for Complex Assemblies and Components. Licence: CC BY 3.0
Attack on critical infrastructures

- The worm was discovered in 2010. It caused substantial damage to the nuclear program of Iran.
- The worm targets Siemens PLCs through the supervisory control and data acquisition systems (SCADA).

Source: Wikipedia (Sándor Vámos), Licence: CC BY-SA 4.0
Attack on critical infrastructures

- The gas centrifuges are applied for separating nuclear materials. They are controlled by PLCs.

Attack on critical infrastructures

- Attack on the **power grid** of Kiev on 17 December 2016.
- A fifth of the city went into a blackout in an hour.
- The malware was designed to disrupt the working processes of industrial control systems.

*Source: [https://www.westmonroe.com/perspectives/in-brief/is-your-utility-prepared-for-industroyer-malware](https://www.westmonroe.com/perspectives/in-brief/is-your-utility-prepared-for-industroyer-malware)*
Ransomware attacks

- 230,000 computers were infected in 2017.
- (National Health Service GB, Telefónica Spain, Deutsche Bahn Germany, FedEx USA)
- Propagated through the **EternalBlue** exploit.
Ransomware attacks

- NotPetya began spreading on 27 June 2017.
- The malware was propagated via e-mail attachments.
- Targets the Server Message Block vulnerability (EternalBlue), like the WannaCry.
- The encryption was modified and the malware could not technically revert its changes.
The reasons behind the vulnerabilities
The human factor

"IT'S EASIER TO FOOL PEOPLE THAN TO CONVINCE THEM THAT THEY HAVE BEEN FOOLED."

~MARK TWAIN

Source: https://www.reddit.com/r/QuotesPorn/comments/avgwz6/its_easier_to_fool_people_than_to_convince_them/
Code Defects

(https://infosectests.com/cissp-study-references/domain-8-app-dev/code-defects/)

a) **Industry Average**: “about 15 – 50 errors per 1000 lines of delivered code.” He further says this is usually representative of code that has some level of structured programming behind it, but probably includes a mix of coding techniques.

b) **Microsoft Applications**: “about 10 – 20 defects per 1000 lines of code during in-house testing, and 0.5 defect per KLOC (KLOC IS CALLED AS 1000 lines of code) in released product (Moore 1992).” He attributes this to a combination of code-reading techniques and independent testing (discussed further in another chapter of his book).

c) “Harlan Mills pioneered ‘cleanroom development’, a technique that has been able to achieve rates as low as 3 defects per 1000 lines of code during in-house testing and 0.1 defect per 1000 lines of code in released product (Cobb and Mills 1990). A few projects – for example, the space-shuttle software – have achieved a level of 0 defects in 500,000 lines of code using a system of format development methods, peer reviews, and statistical testing.”
The software reliability curve

Source: Claude Y. Laporte and Alain April Software Quality Assurance
The classic solutions for protection against the cyber threats
The onion model

Source: https://www.geeksforgeeks.org/introduction-to-security-defense-models/
Source: https://eu.democratandchronicle.com/story/money/business/blogs/innovation/2016/10/04/cybersecurity-is-like-an-onion/91543960/
Firewalls

- Monitors and controls the network traffic based on predefined security rules.

**Firewall types:**
- Packet filters (ACL)
  
  1987 Digital Equipment Corporation
- Stateful firewalls (applies session tracking)
  
  1989 – 1990 AT&T Bell Laboratories
- Application firewall
  
  1993 Marcus Ranum, Wei Xu, and Peter Churchyard
- Deep packet inspection
  
  Since 2012
IDS/IPS

- Monitors the network or the system for malicious activity or policy violations.
- The logs are usually collected and analyzed using SIEM (Security Information and Event Management) system.
- NIDS vs HIDS
- Detection methods:
  - Signature-based detection
  - Anomaly-based detection
  - Stateful protocol analysis detection
Ranking IDS/IPS in 2022

1. solarwinds
2. Bro
3. OSSEC
4. SNORT
5. Suricata
6. SECURITY ONION
7. Open WIPS – NG
8. Sagan
9. McAfee Network Security Platform
10. paloalto networks
Checking software vulnerabilities

- Static code analysis
  - .NET Security Guard
  - AppSweep
  - ClodeDefense
  - DeepDive
  - FindBugs
  - SonarQube
  - SourceMeter

- Security coding rules
  - MISRA
  - SEI CERT
  - OWASP
Applying machine learning methods in cyber defense
Areas where machine learning supports the security

- **Spam filtering**
  DOI:10.11591/ijict.v9i1.pp9-18

- **Face recognition**
  *Electronics* 2020, 9, 1188. https://doi.org/10.3390/electronics9081188

- **Phishing detection**
  10.1007/978-981-33-4893-6_25.

- **Vulnerability prediction**
Areas where machine learning supports the security

• **Bug prediction**

• **Malware prediction**

• **Intrusion Detection**
Vulnerability prediction

- A vulnerability is a hole or a weakness in the application, which can be a design flaw or an implementation bug, that allows an attacker to cause harm to the stakeholders of an application. [OWASP]

- The actual vulnerabilities are language-dependent, therefore, the vulnerability detectors are designed for programming languages.

- JavaScript-based applications are proliferated and the design of the language makes it possible to write vulnerable applications.

- A large number of machine learning-based vulnerability detection processes utilize software and process metrics as the predictor features for deciding about vulnerabilities.

- The applied machine learning methods are in the set of supervised methods. In those methods, we have to collect and label both positive and negative examples.
Vulnerability prediction


Vulnerability Dataset:

- Node Security Platform: https://github.com/nodesecurity/nsp
- Snyk Vulnerability Database: https://snyk.io/vuln
Deep Water Framework


- **Applied machine learning techniques:**
  - Naive Bayes
  - Support Vector Machine
  - K-nearest Neighbors
  - Logistic Regression
  - Linear Regression
  - Decision Tree
  - Random Forest
  - Simple Deep Neural Network
  - Custom Deep Neural Network
Vulnerability prediction

<table>
<thead>
<tr>
<th>Classifier</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC</td>
<td>730</td>
<td>7046</td>
<td>32</td>
<td>230</td>
<td>96.7%</td>
<td>95.8%</td>
<td>76.0%</td>
<td>84.8% (+13.5%)</td>
</tr>
<tr>
<td>DT</td>
<td>723</td>
<td>7006</td>
<td>72</td>
<td>237</td>
<td>96.2%</td>
<td>90.9%</td>
<td>83.7%</td>
<td>82.4% (+10.8%)</td>
</tr>
<tr>
<td>KNN</td>
<td>684</td>
<td>7041</td>
<td>37</td>
<td>276</td>
<td>96.1%</td>
<td>94.9%</td>
<td>71.3%</td>
<td>71.6% (+5%)</td>
</tr>
<tr>
<td>SDNN</td>
<td>687</td>
<td>7019</td>
<td>59</td>
<td>273</td>
<td>95.9%</td>
<td>92.1%</td>
<td>80.5%</td>
<td>78.5% (+11.7%)</td>
</tr>
<tr>
<td>CDNN</td>
<td>678</td>
<td>7025</td>
<td>53</td>
<td>282</td>
<td>95.8%</td>
<td>92.8%</td>
<td>70.6%</td>
<td>71.7% (+9.4%)</td>
</tr>
<tr>
<td>SVM</td>
<td>692</td>
<td>6966</td>
<td>112</td>
<td>268</td>
<td>95.3%</td>
<td>86.1%</td>
<td>72.1%</td>
<td>78.5% (+11.7%)</td>
</tr>
<tr>
<td>LogReg</td>
<td>496</td>
<td>6906</td>
<td>172</td>
<td>464</td>
<td>92.1%</td>
<td>74.3%</td>
<td>51.7%</td>
<td>60.9% (+27.8%)</td>
</tr>
<tr>
<td>LinReg</td>
<td>570</td>
<td>6592</td>
<td>486</td>
<td>390</td>
<td>89.1%</td>
<td>54.0%</td>
<td>59.4%</td>
<td>56.6% (+24.5%)</td>
</tr>
<tr>
<td>NB</td>
<td>115</td>
<td>6779</td>
<td>299</td>
<td>845</td>
<td>85.8%</td>
<td>27.8%</td>
<td>12.0%</td>
<td>16.7% (+1.4%)</td>
</tr>
</tbody>
</table>

Intrusion detection

- Detect and identify malicious network packets.
  - The classical methods apply rules or pattern recognition methods.
  - Using machine learning, a novel malicious packet can also be recognized.
- The models focus on anomaly detection in the network traffic.
  - The simplest anomaly detection techniques apply statistical methods (Z-value, IQR).

Intrusion detection

- Multivariable anomaly detection methods (unsupervised methods).
  - K-means, DBSCAN, Local Outlier Factor, Isolation Forest
- In live traffic, labeled data are not achievable, therefore, supervised methods cannot be applied without compromise.

Source: https://www.geeksforgeeks.org/local-outlier-factor

Source: https://www.sciencedirect.com/science/article/pii/S1474034620301105
Intrusion detection

- Unsupervised and semi-supervised methods.
  - OCSVM, Autoencoder, GAN
- A classifier is to be applied on top of the Autoencoder.
LSTM Autoencoder

- **Autoencoder** is made up of **LSTM** components.
- The **LSTM** (**Long Short Term Memory**) is capable to represent sequential data.
  - The semantic relationship among the network packets can be represented.
# Comparison of the models

<table>
<thead>
<tr>
<th></th>
<th>Schneider1</th>
<th>Schneider2</th>
<th>Schneider3</th>
<th>Siemens1</th>
<th>Siemens2</th>
<th>Siemens3</th>
<th>Siemens4</th>
<th>Siemens5</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of training packages</td>
<td>29160</td>
<td>1097</td>
<td>10494</td>
<td>181612</td>
<td>33603</td>
<td>181612</td>
<td>33603</td>
<td>41888</td>
</tr>
<tr>
<td>number of normal testing packages</td>
<td>142</td>
<td>555</td>
<td>510</td>
<td>6764</td>
<td>82</td>
<td>19680</td>
<td>19680</td>
<td>19680</td>
</tr>
<tr>
<td>number of malicious testing packages</td>
<td>8654</td>
<td>12633</td>
<td>12883</td>
<td>5731</td>
<td>4525421</td>
<td>4012503</td>
<td>4951600</td>
<td>4951600</td>
</tr>
</tbody>
</table>

| LOF                  | original    | precision | 98.00%      | 96.70%     | 96%       | 47.00%    | 99.99%    | 99.99%    | 99.99%    | 99.99%    |
|                      | recall      |           | 95.70%      | 72.60%     | 97.20%    | 49.20%    | 99.99%    | 99.99%    | 99.99%    | 99.99%    |
|                      | f          |           | 97.10%      | 83%        | 96.60%    | 48.10%    | 99.99%    | 99.97%    | 99.99%    | 99.99%    |

| IF                   | original    | precision | 98.40%      | 97.10%     | 98.90%    | 45.80%    | 99.99%    | 99.99%    | 99.99%    | 99.99%    |
|                      | recall      |           | 100%        | 68.10%     | 23.10%    | 98.40%    | 100%      | 100%      | 100%      | 100%      |
|                      | f          |           | 99.20%      | 80%        | 37.50%    | 62.50%    | 99.99%    | 99.99%    | 99.99%    | 99.99%    |

| OCSVM                | original    | precision | 97.70%      | 93.90%     | 0         | 45.80%    | 99.99%    | 99.99%    | 99.99%    | 99.99%    |
|                      | recall      |           | 4.50%       | 27.20%     | 0         | 100%      | 100%      | 100%      | 100%      | 100%      |
|                      | f          |           | 8.50%       | 42.10%     | 0         | 62.90%    | 99.99%    | 99.99%    | 99.99%    | 99.99%    |

| Composite            | original    | precision | 97.70%      | 92.40%     | 0         | 45.60%    | 99.99%    | 99.99%    | 99.99%    | 99.99%    |
|                      | recall      |           | 4.40%       | 2%         | 0         | 47.80%    | 99.99%    | 99.95%    | 99.96%    | 99.99%    |
|                      | f          |           | 8.50%       | 3.90%      | 0         | 47%       | 99.99%    | 99.97%    | 99.98%    | 99.99%    |

| LOF                  | derived     | precision | 98.60%      | 96.30%     | 96.10%    | 46.20%    | 99.99%    | 99.99%    | 99.99%    | 99.99%    |
|                      | recall      |           | 89.00%      | 83.60%     | 95.50%    | 66.40%    | 99.99%    | 99.99%    | 99.99%    | 99.99%    |
|                      | f          |           | 93.80%      | 89.60%     | 96.60%    | 54.50%    | 99.99%    | 99.99%    | 99.99%    | 99.99%    |

| IF                   | derived     | precision | 98.40%      | 97.10%     | 94%       | 46.30%    | 99.99%    | 99.99%    | 99.99%    | 99.99%    |
|                      | recall      |           | 99.90%      | 64.80%     | 21.70%    | 97%       | 99.99%    | 98.60%    | 99.99%    | 99.99%    |
|                      | f          |           | 99.20%      | 77.60%     | 35.30%    | 62.60%    | 99.99%    | 99.30%    | 99.30%    | 99.30%    |

| OCSVM                | derived     | precision | 98.50%      | 94.40%     | 100%      | 46.40%    | 99.80%    | 100%      | 100%      | 100%      |
|                      | recall      |           | 99%         | 5.20%      | 1.60%     | 94.60%    | 99%       | 99%       | 99%       | 99%       |
|                      | f          |           | 95.70%      | 9.90%      | 3.10%     | 62.30%    | 99%       | 99%       | 99%       | 99%       |

| Composite            | derived     | precision | 98.60%      | 100%       | 100%      | 83.30%    | 0.06%     | 1.40%     | 2.80%     | 2.80%     |
|                      | recall      |           | 83.30%      | 0.06%      | 1.40%     | 2.80%     | 83.30%    | 0.06%     | 1.40%     | 2.80%     |
|                      | f          |           | 90.30%      | 0.12%      | 2.80%     | 2.80%     | 90.30%    | 0.12%     | 2.80%     | 2.80%     |
Neural networks can also be fooled

By adding four rectangular stickers, researchers tricked an 'artificial intelligence' system to read this 'Stop' sign as 'Speed Limit 45'.

Source: Ian J Goodfellow, EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES, ICLR2015
Thank you for your attention!