UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

UNIVERSITY OF SZEGED

Department of Software Engineering

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

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

IARIA

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

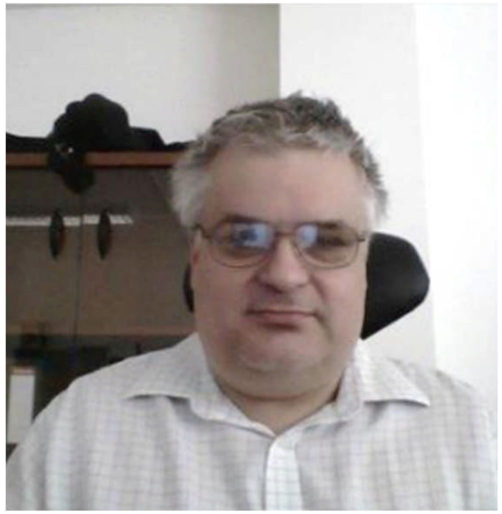
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS · SZEGEDI TUDOMÁNYEGYETEM
VERITAS VIRTUS LIBERTAS

UNIVERSITY OF SZEGED

Department of Software Engineering

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

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

UNIVERSITY OF SZEGED
Department of Software Engineering
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

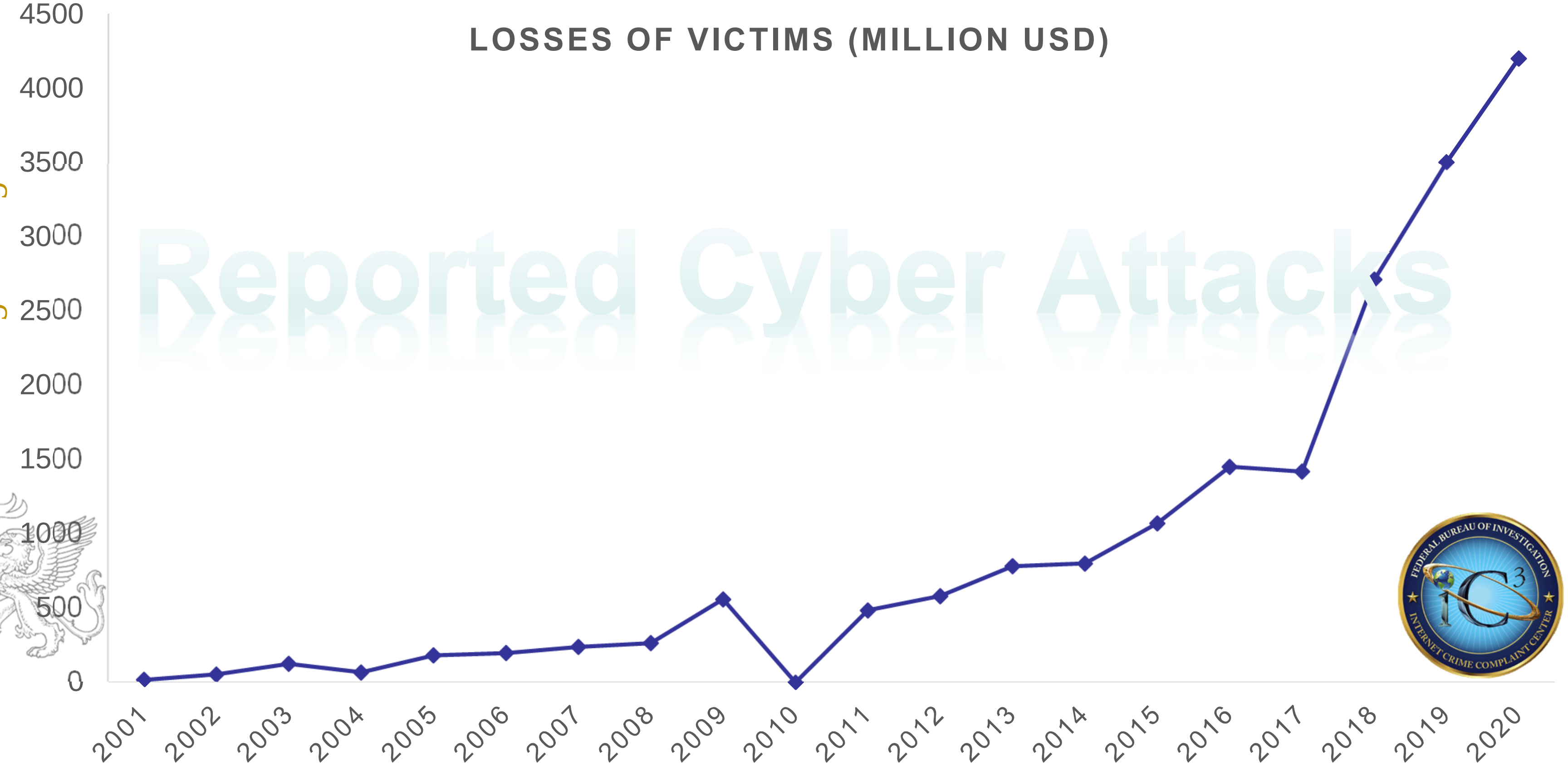
Agenda

- **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*

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
UNIVERSITY OF SZEGED
Department of Software Engineering

The global cyber threat

Reported losses by FBI Internet Crime Complaint Center



Evolution of cyber attacks

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

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

2000s
- Exploits
- DoS/DDoS

2010s
- polymorfic 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

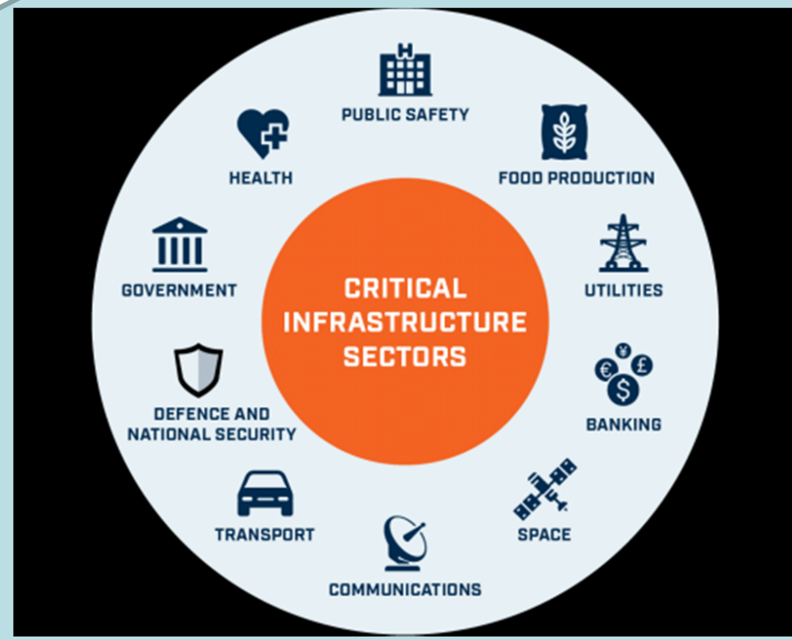
Changing the targets

1980s individual hosts

2010s industrial systems

Source:Pinterest

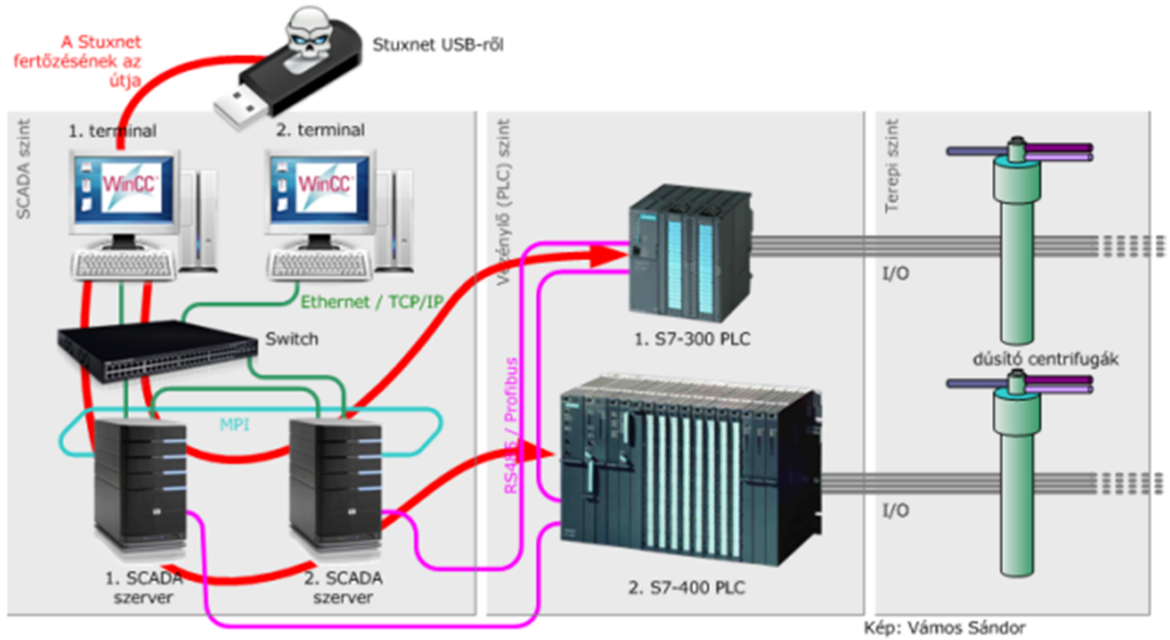
Critical infrastructures



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**).



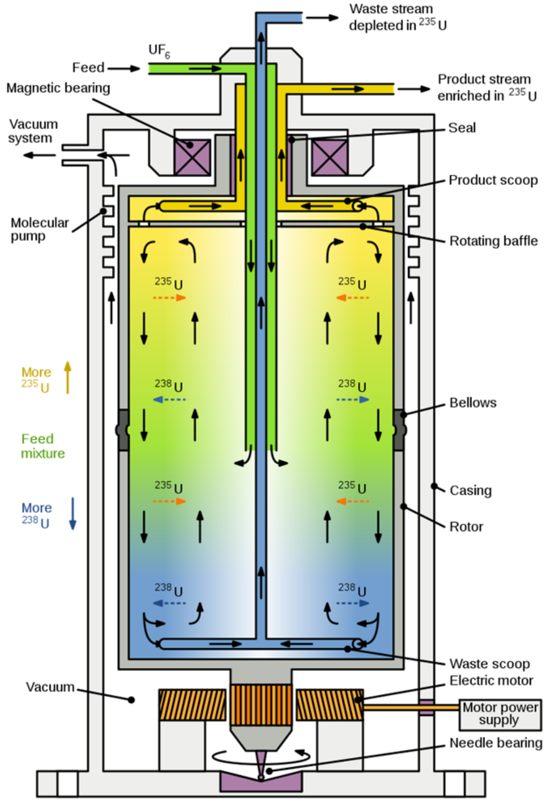
Source:Wikipedia (Sándor Vámos), **Licence**: CC BY-SA 4.0

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
UNIVERSITY OF SZEGED
Department of Software Engineering

Attack on critical infrastructures

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

Source:Wikipedia, **Licence**: CC BY-SA 2.5



Source:Wikipedia, Wikimedia Commons

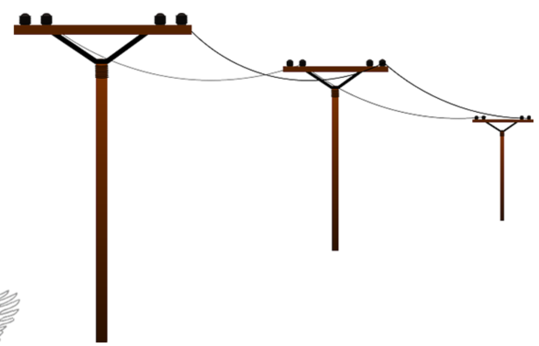
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

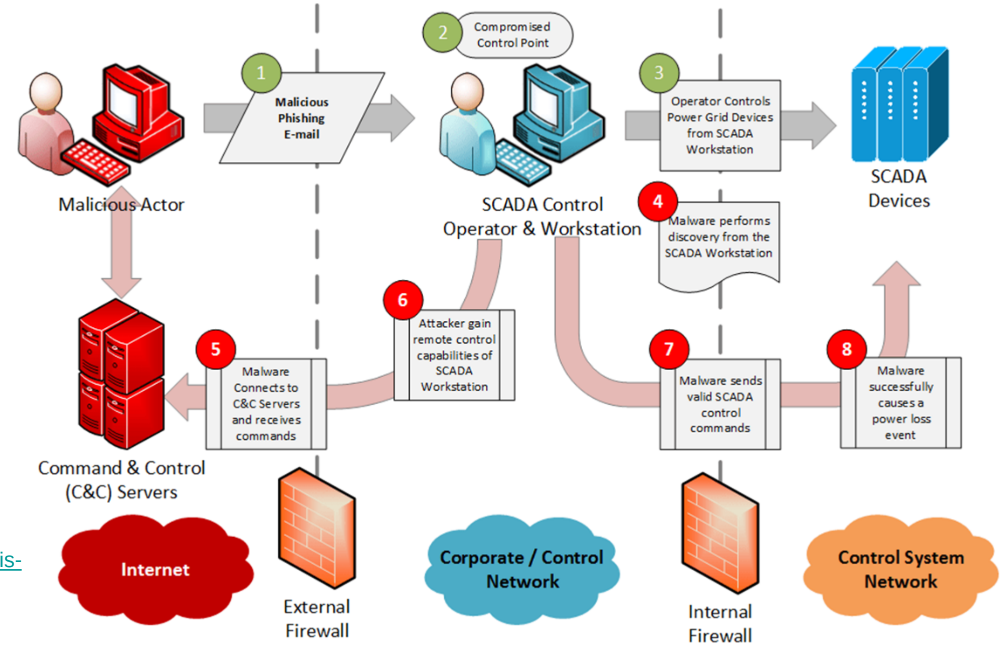
UNIVERSITY OF SZEGED

Department of Software Engineering

Attack on critical infrastructures

- ✓ Attack on the **power grid** of Kiyv 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**.



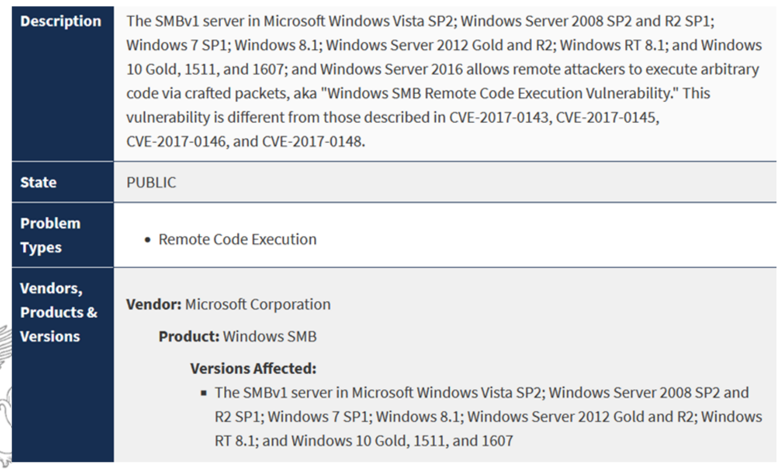


Source: https://www.westmonroe.com/perspectives/in-brief/is-your-utility-prepared-for-industroyer-malware

UNIVERSITY OF SZEGED
Department of Software Engineering
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

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.



Description	The SMBv1 server in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; Windows 8.1; Windows Server 2012 Gold and R2; Windows RT 8.1; and Windows 10 Gold, 1511, and 1607; and Windows Server 2016 allows remote attackers to execute arbitrary code via crafted packets, aka "Windows SMB Remote Code Execution Vulnerability." This vulnerability is different from those described in CVE-2017-0143, CVE-2017-0145, CVE-2017-0146, and CVE-2017-0148.
State	PUBLIC
Problem Types	• Remote Code Execution
Vendors, Products & Versions	**Vendor:** Microsoft Corporation **Product:** Windows SMB **Versions Affected:** • The SMBv1 server in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; Windows 8.1; Windows Server 2012 Gold and R2; Windows RT 8.1; and Windows 10 Gold, 1511, and 1607

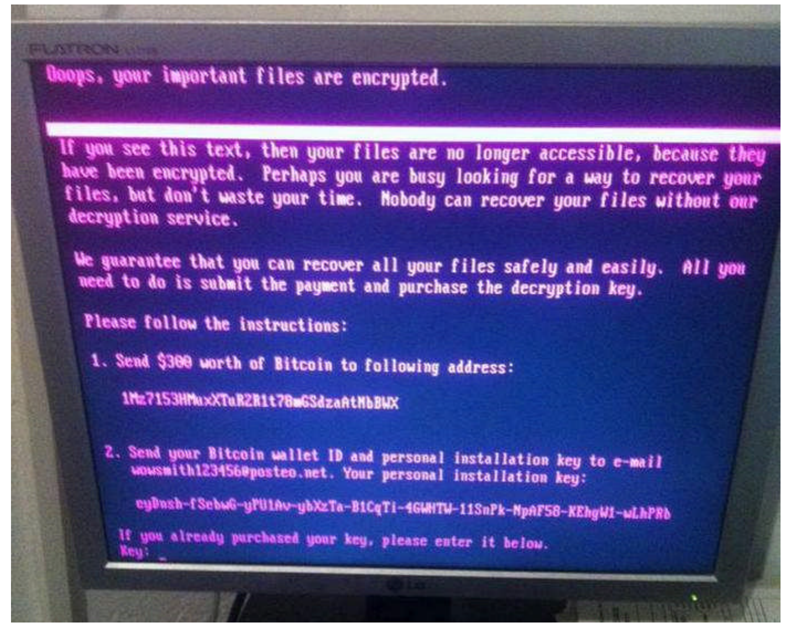
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

UNIVERSITY OF SZEGED

Department of Software Engineering

Ransomware attacks

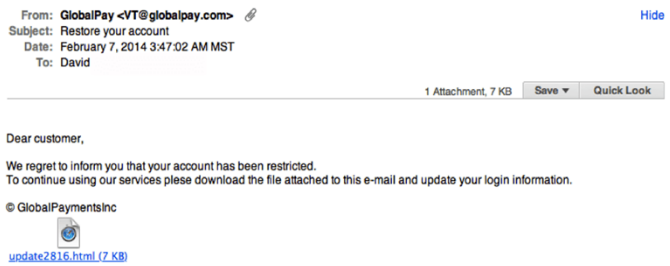
- ✓ NotPetya began spreading on 27 June 2017.
- ✓ The malware was propagated **via e-mail attachments**.
- ✓ Targets the **Server Message Block vulnerability (EternalBlue)**, like the WannaCry.
- ✓ The **encryption was modified** and the malware **could not technically revert its changes**.

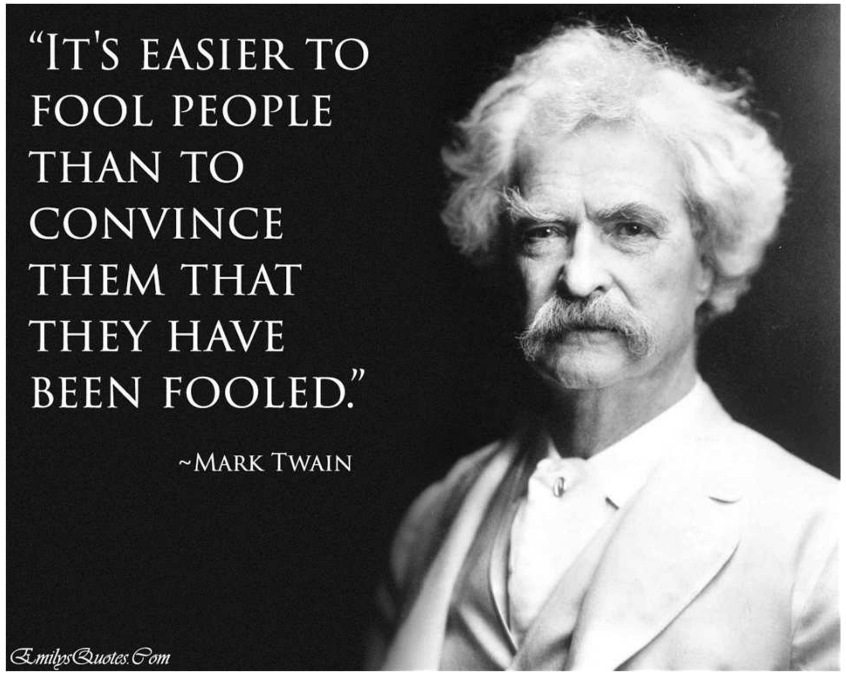


The reasons behind the vulnerabilities

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
UNIVERSITY OF SZEGED
Department of Software Engineering

The human factor





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

UNIVERSITY OF SZEGED
Department of Software Engineering
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

Code Defects

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

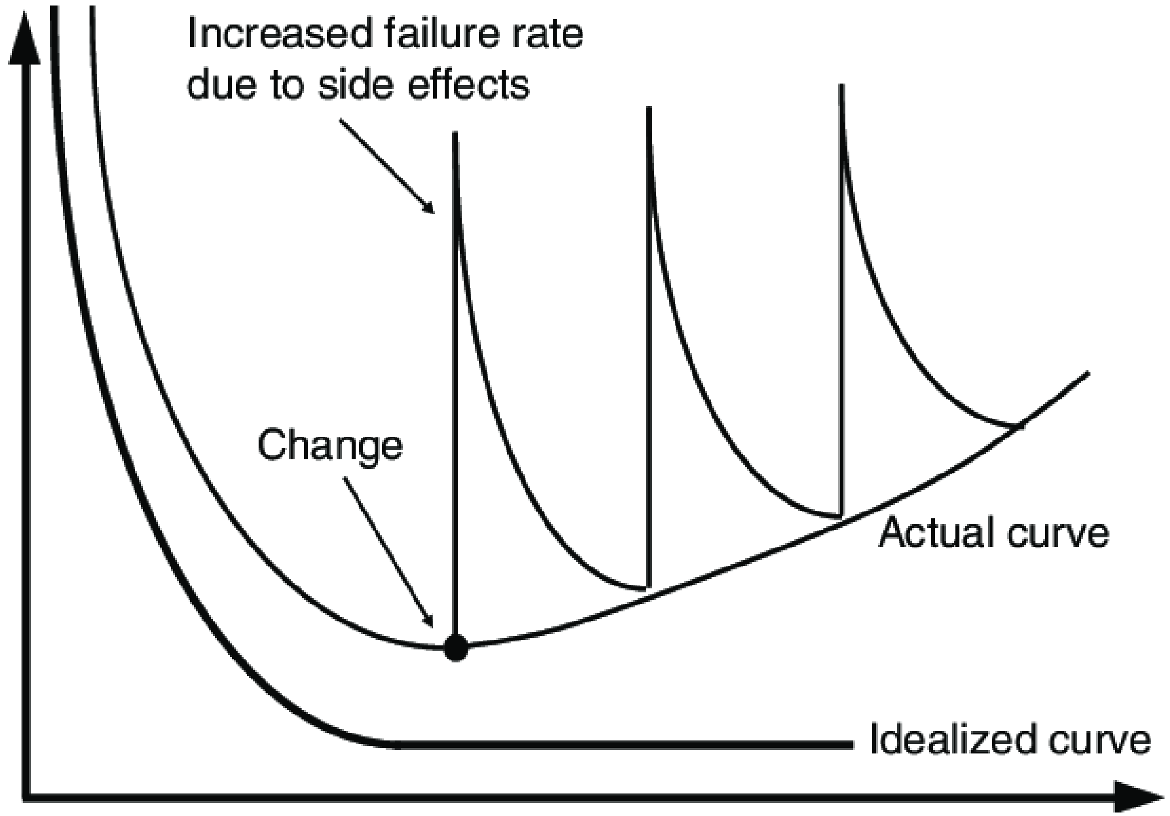
a) **Industry Average**: "about 15 – 50 errors per 1000 lines of delivered code." He further says this is usually representative of code that has some level of structured programming behind it, but probably includes a mix of coding techniques.

(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."

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
UNIVERSITY OF SZEGED
Department of Software Engineering

The software reliability curve

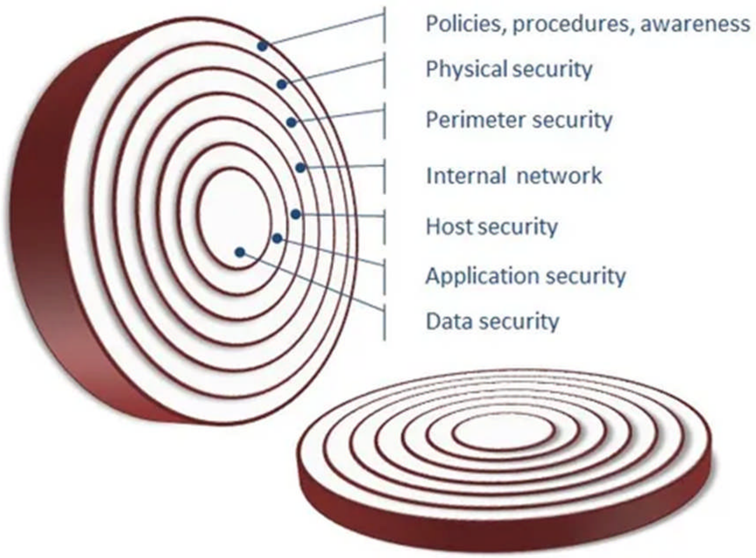


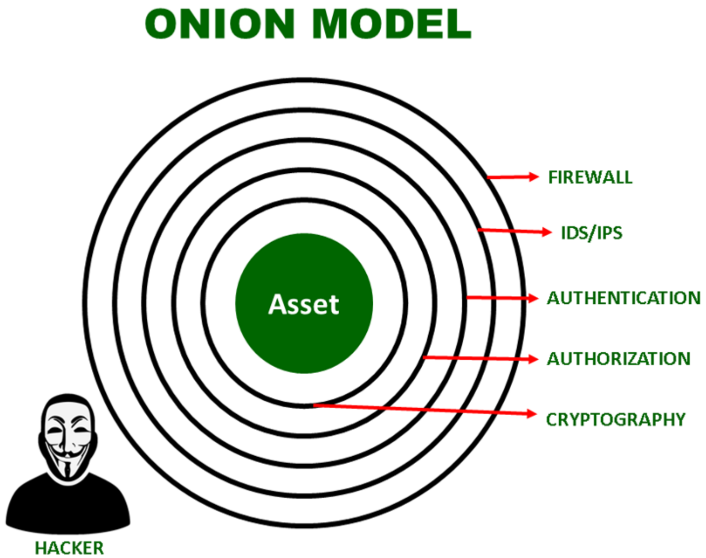
Source: Claude Y. Laporte and Alain April Software Quality Assurance

UNIVERSITY OF SZEGED
Department of Software Engineering
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

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

    The filters can be applied to the application layer.

    /1993 Marcus Ranum, Wei Xu, and Peter Churchyard/

  - Deep packet inspection

    /Since 2012/

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
UNIVERSITY OF SZEGED
Department of Software Engineering

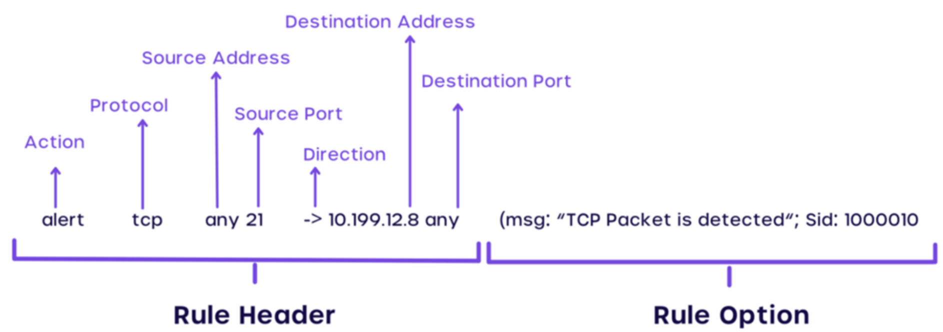
IDS/IPS

- 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

Software Testing Help

1. solarwinds
2. Bro
3. OSSEC
4. SNORT
5. Suricata
   Open Source IDS / IPS / NSM engine
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

UNIVERSITY OF SZEGED
Department of Software Engineering
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

Applying machine learning methods in cyber defense

Areas where machine learning supports the security

- **Spam filtering**

  - A.A. Ojugo, A. O. Eboka: Memetic algorithm for short messaging service spam filter using text normalization and semantic approach in International Journal of Informatics and Communication Technology, 2020 DOI:10.11591/ijict.v9i1.pp9-18

- **Face recognition**

  - Adjabi, I.; Ouahabi, A.; Benzaoui, A.; Taleb-Ahmed, A. Past, Present, and Future of Face Recognition: A Review. *Electronics* 2020, *9*, 1188. https://doi.org/10.3390/electronics9081188

- **Phising detection**

  - Abbigeri, Shivarajakumar & Pashupatimath, Anand. (2021). Detection of Phishing E-Mails: A Learning-Based Approach. 10.1007/978-981-33-4893-6_25.

- **Vulnerability prediction**

  - Viszkok, T.; Hegedűs, P. and Ferenc, R. (2021). Improving Vulnerability Prediction of JavaScript Functions using Process Metrics. In *Proceedings of the 16th International Conference on Software Technologies - ICSOFT,* ISBN 978-989-758-523-4; ISSN 2184-2833, pages 185-195. DOI: 10.5220/0010558501850195

Areas where machine learning supports the security

- **Bug prediction**

  - Aladics, T., Jász, J., Ferenc, R. (2021). Bug Prediction Using Source Code Embedding Based on Doc2Vec. In: , *et al.* Computational Science and Its Applications – ICCSA 2021. ICCSA 2021. Lecture Notes in Computer Science(), vol 12955. Springer, Cham. https://doi.org/10.1007/978-3-030-87007-2_270195

- **Malware prediction**

  - U. Adamu and I. Awan, "Ransomware Prediction Using Supervised Learning Algorithms," *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2019, pp. 57-63, doi: 10.1109/FiCloud.2019.00016.

  - Cannarile, A.; Dentamaro, V.; Galantucci, S.; Iannacone, A.; Impedovo, D.; Pirlo, G. Comparing Deep Learning and Shallow Learning Techniques for API Calls Malware Prediction: A Study. *Appl. Sci.* **2022**, *12*, 1645. https://doi.org/10.3390/app12031645

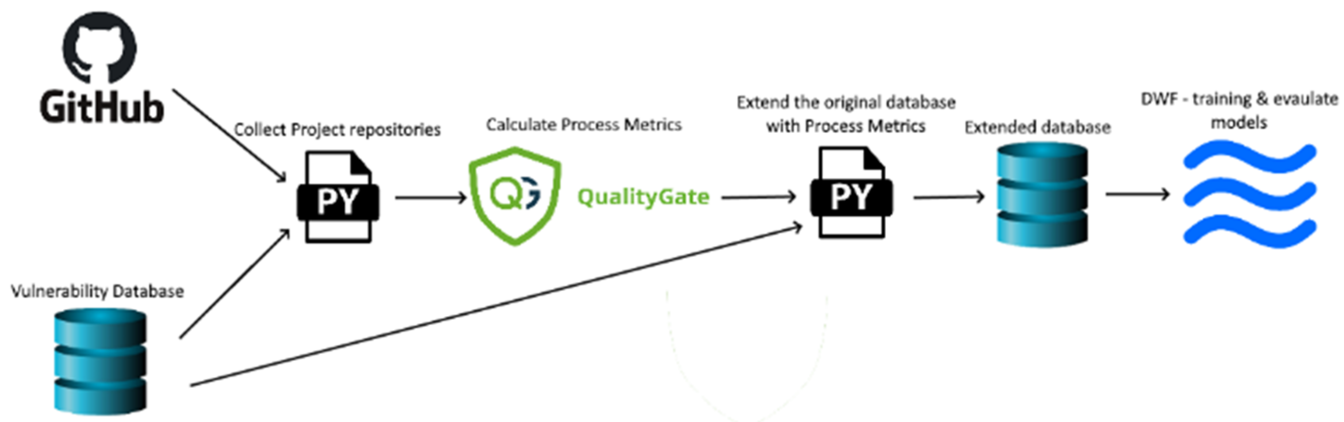
- **Intrusion Detection**

  - F. Farivar, M. S. Haghighi, A. Jolfaei and M. Alazab, "Artificial Intelligence for Detection, Estimation, and Compensation of Malicious Attacks in Nonlinear Cyber-Physical Systems and Industrial IoT," in IEEE Transactions on Industrial Informatics, vol. 16, no. 4, pp. 2716-2725, April 2020, doi: 10.1109/TII.2019.2956474.

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



Source: Viszkok, T.; Hegedűs, P. and Ferenc, R. (2021). Improving Vulnerability Prediction of JavaScript Functions using Process Metrics.

Vulnerability Dataset:

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

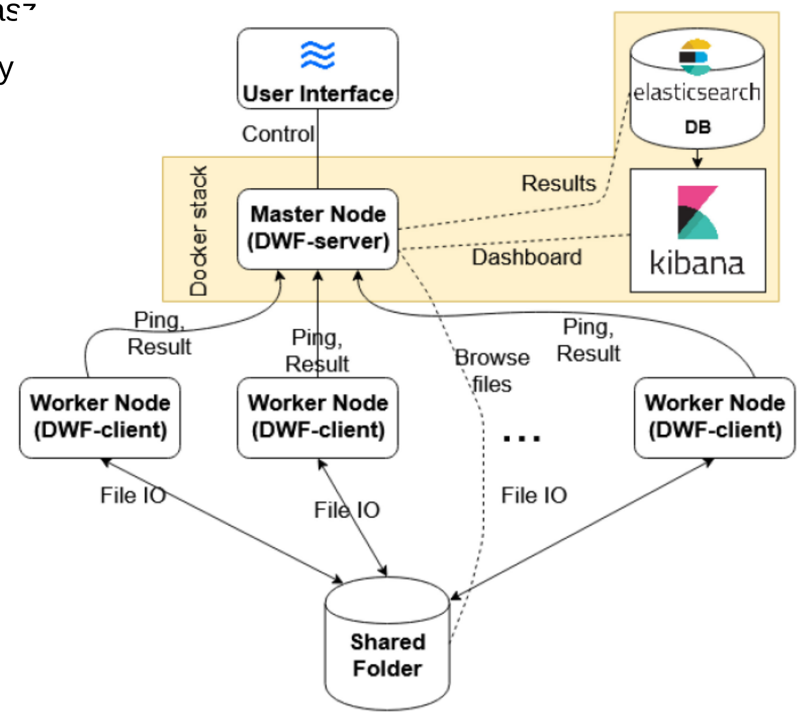
UNIVERSITY OF SZEGED

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

Department of Software Engineering

Deep Water Framework

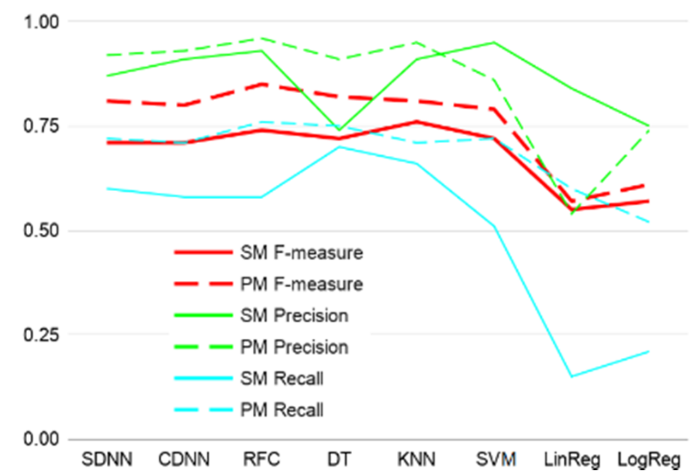
- Rudolf Ferenc, Tamás Viszkok, Tamás Aladics, Judit Jás Péter Hegedűs, Deep-water framework: The Swiss army knife of humans working with machine learning models, https://doi.org/10.1016/j.softx.2020.100551.
- **Applied machine learning techniques:**
  - Naive Bayes
  - Support Vector Machine
  - K-nearest Neighbors
  - Logistic Regression
  - Linear Regression
  - Decision Tree
  - Random Forest
  - Simple Deep Neural Network
  - Custom Deep Neural Network



Vulnerability prediction

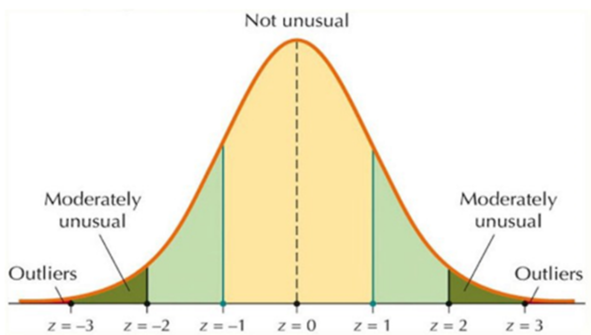
Classifier	TP	TN	FP	FN	Accuracy	Precision	Recall	F-measure
RFC	**730**	**7046**	**32**	**230**	**96.7%**	**95.8%**	**76.0%**	**84.8% (+13.5%)**
DT	723	7006	72	237	96.2%	90.9%	75.3%	82.4% (+10.8%)
KNN	684	7041	37	276	96.1%	94.9%	71.3%	81.4% (+5%)
SDNN	687	7019	59	273	95.9%	92.1%	71.6%	80.5% (+9.4%)
CDNN	678	7025	53	282	95.8%	92.8%	70.6%	80.2% (+9.4%)
SVM	692	6966	112	268	95.3%	86.1%	72.1%	78.5% (+11.7%)
LogReg	496	6906	172	464	92.1%	74.3%	51.7%	60.9% (+27.8%)
LinReg	570	6592	486	390	89.1%	54.0%	59.4%	56.6% (+24.5%)
NB	115	6779	299	845	85.8%	27.8%	12.0%	16.7% (+1.4%)

Results achieved in the article of Viszkok, T.; Hegedűs, P. and Ferenc, R. (2021). Improving Vulnerability Prediction of JavaScript Functions using Process Metrics

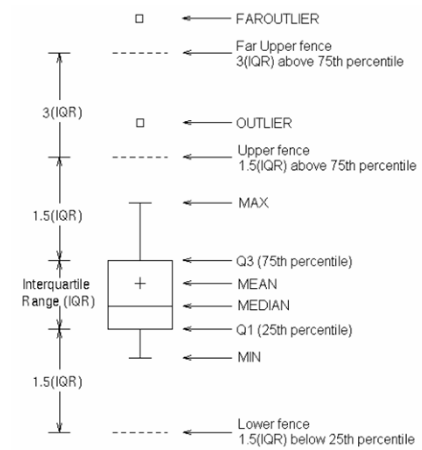


Intrusion detection

- 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).



Source: https://medium.com/@2016pceecsankalp081/top-4-best-way-to-detect-outliers-in-the-dataset-73eedd1aa12d

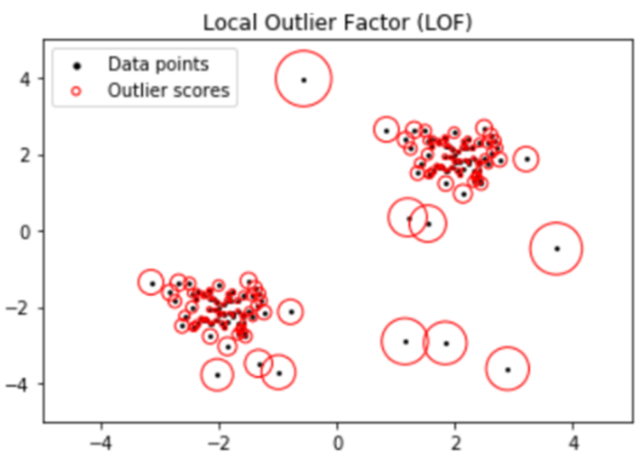


Source: https://blogs.sas.com/content/iml/2019/08/28/schematic-box-plot.html

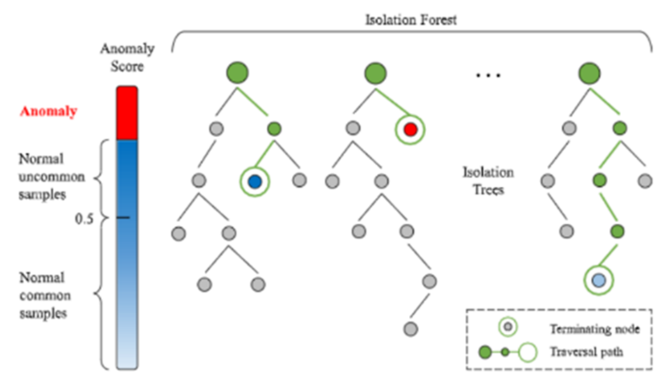
Intrusion detection

- 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

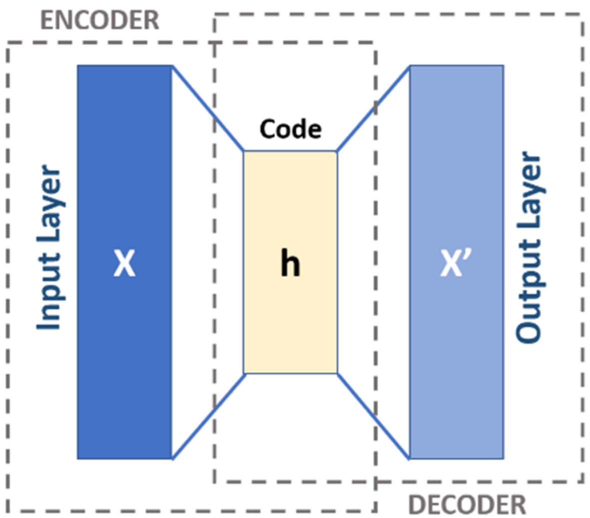


Source: https://www.sciencedirect.com/science/article/pii/S1474034620301105

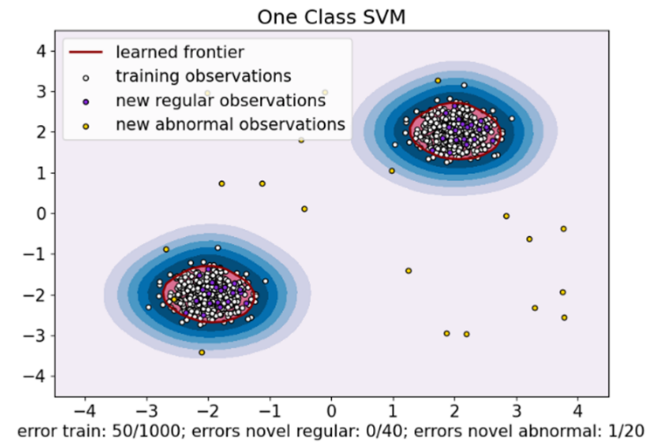
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
UNIVERSITY OF SZEGED
Department of Software Engineering

Intrusion detection

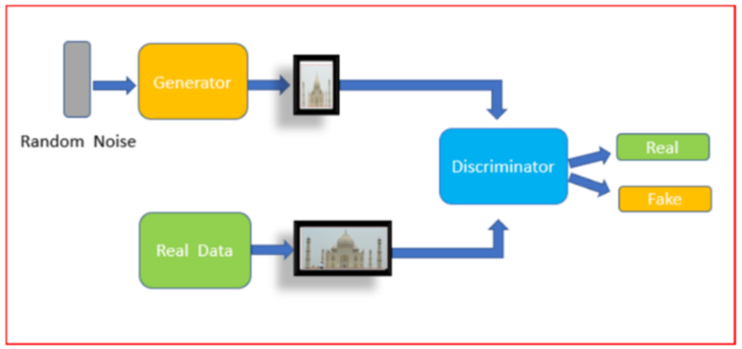
- Unsupervised and semi-supervised methods.
  - OCSVM, Autoencoder, GAN
- A classifier is to be applied on top of the Autoencoder.



Source: Wikipedia (Michela Massi) **Licence**: CC BY-SA 4.0



Source: https://scikit-learn.org/stable/auto examples/linear model/plot sgdocsvm vs ocsvm.html



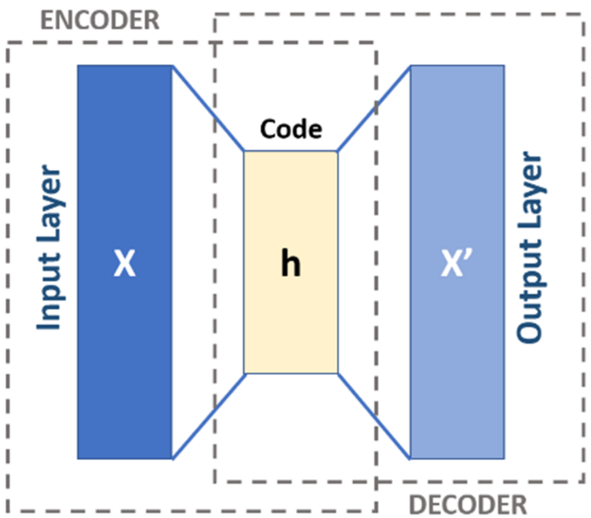
Source: https://medium.datadriveninvestor.com/generative-adversarial-network-gan-using-keras-ce1c05cfdfd3

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
UNIVERSITY OF SZEGED
Department of Software Engineering

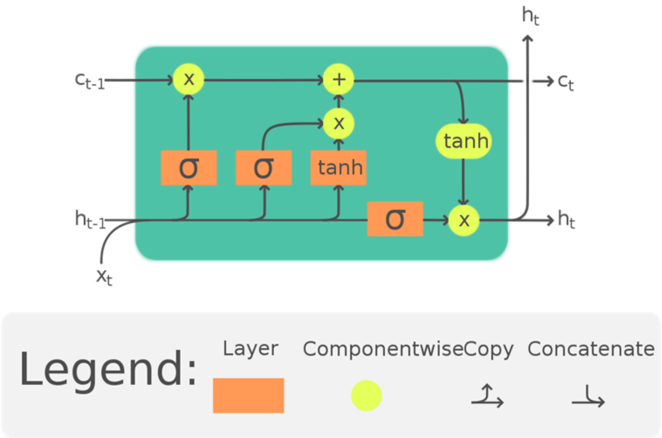
LSTM Autoencoder

- **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.



Source: Wikipedia (Michela Massi) **Licence**: CC BY-SA 4.0



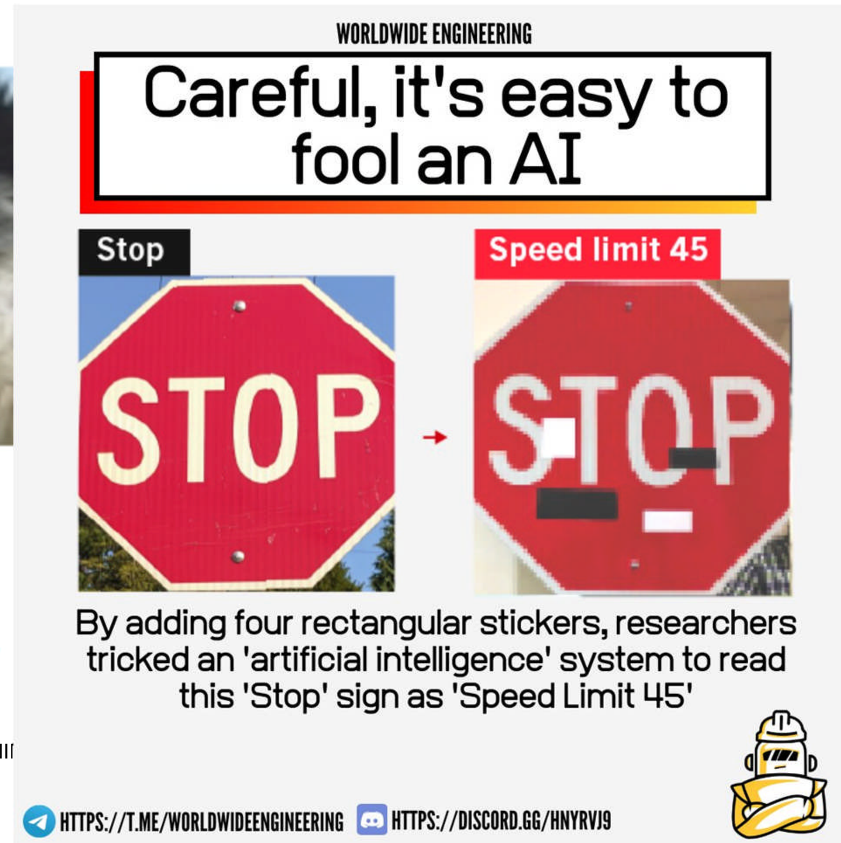
Source: Wikipedia (Guillaume Chevalier) **Licence**: CC BY-SA 4.0

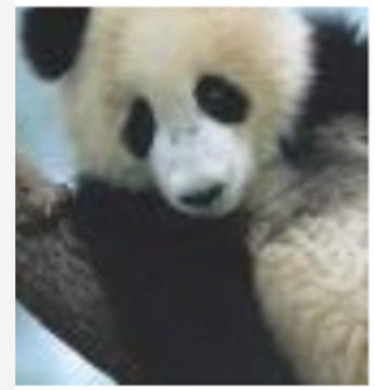
UNIVERSITY OF SZEGED
Department of Software Engineering
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

Comparison of the models

			Schneider1	Schneider2	Schneider3	Siemens1	Siemens2	Siemens3	Siemens4	Siemens5
number of training packages			29160	1097	10494	181612	33603	41888	23424	19680
number of normal testing packages			142	555	510	6764	82		30	55
number of malicious testing packages			8654	12633	12883	5731	4525421		4012503	4951600
LOF	original	precision	98,00%	96,70%	96%	47,00%	99,99%	9,99%	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%	99,96%	100%
		f	8,50%	42,10%	0	62,90%	99,99%	99,99%	99,98%	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%		
		recall	89,00%	83,60%	95,50%	66,40%	99,99%	99,99%		
		f	93,80%	89,60%	96,60%	54,50%	99,99%	99,99%		
IF	derived	precision	98,40%	97,10%	94%	46,30%	99,99%	99,99%		
		recall	99,90%	64,80%	21,70%	97%	99,99%	98,60%		
		f	99,20%	77,60%	35,30%	62,60%	99,99%	99,30%		
OCSVM	derived	precision	98,50%	94,40%	100%	46,40%	99,80%	100%		
		recall	93%	5,20%	1,60%	94,60%	0,90%			
		f	95,70%	9,90%	3,10%	62,30%	1,90%			
Composite	derived	precision	98,60%	100%	100%					
		recall	83,30%	0,06%	1,40%					
		f	90,30%	0,12%	2,80%					

Neural networks can also be fooled





Source: Ian J Goodfellow, EXPLAINI

UNIVERSITY OF SZEGED
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
Department of Software Engineering

Thank you
for your attention!