

Network intrusion detection with Deep Learning

Roberto Doriguzzi Corin

Cybersecurity centre

Fondazione Bruno Kessler





About me

Roberto Doriguzzi Corin

- Senior researcher at FBK
- Research focus on network security
 - Application of DL techniques to cybersecurity problems
 - Optimisation techniques applied to the provisioning of security services
 - Network softwarisation technologies (SDN and NFV)

Roberto Doriguzzi Corin, PhD
Cybersecurity centre
Fondazione Bruno Kessler (FBK)
via Sommarive, 18
38123 Povo, Trento (Italy)
Tel: (+39) 0461 312429
e-mail: rdoriguzzi@fbk.eu



Outline

- Program of the course
- Tools
- Evaluation
- Introduction to network attacks and anomalies and their detection
- Datasets



Program of the course (1)

- **1:** Presentation of the course and intro to cyber-attacks
- **2:** Fundamentals of Deep Learning in network security
- **3:** Intrusion and anomaly detection with Deep Learning
- **4:** Laboratory: implementation of a DL model with Keras
- **5:** Network traffic features: extraction and selection
 - Laboratory: ranking of packet-level features



Program of the course (2)

- **6:** Hyper-parameters tuning (laboratory)
- **7:** Evaluation of a DL model with unseen data
- **8:** Intrusion/anomaly detection in real-world applications:
Concept drift, Adversarial machine learning
- **9:** Laboratory: Deployment of a DL-based IDS on an embedded device.
- **10:** Federated Learning for cybersecurity applications



About this course

- This is **NOT** a course on Machine/Deep Learning
- This course focuses on applying Deep Learning methods to cybersecurity applications
 - You will learn fundamental concepts of Deep Learning
 - You will learn how to build/train/tune/deploy an DL-based intrusion/anomaly detection system



Tools used during the course

- Python 3.X
 - Including scientific libraries
 - Scikit-learn (<https://scikit-learn.org/>)
 - NumPy (<https://numpy.org/>)
 - Keras and Tensorflow (<https://keras.io/> and <https://www.tensorflow.org/>)
 - Pyshark (<https://pypi.org/project/pyshark/>)
 - Jupyter (<https://jupyter.org/>)
 - PyCharm (<https://www.jetbrains.com/pycharm/>)
 - Or Visual Studio Code (<https://code.visualstudio.com/>)



Attendance and Evaluation

Attendance: minimum 15 hours out of 20

Evaluation

Either:

A project focused on the implementation of a DL-based intrusion detection system

Or:

A 2-pages review of a paper.

- The review should answer questions: what?, why? and how?
- Moreover, it should highlight the limitations (if any) of the proposed solution.



Useful links

- The source code of the laboratories is publicly available at:
<https://github.com/doriguzzi/dl-for-network-security-phd>
- The repository also provides the guidelines for setting up the development environment
- Other material is also available at:
<https://github.com/Doriguzzi>, such as the source code of LUCID, a DL-based solution for DDoS attack detection



List of papers

Suggested papers:

- Adversarial Attacks Against Deep Learning-Based Network Intrusion Detection Systems and Defense Mechanisms (<https://doi.org/10.1109/TNET.2021.3137084>)
- Dos and Don'ts of Machine Learning in Computer Security (<https://www.usenix.org/system/files/sec22-arp.pdf>)
- Deep Anomaly Detection for Time-Series Data in Industrial IoT: A Communication-Efficient On-Device Federated Learning Approach (<https://doi.org/10.1109/JIOT.2020.3011726>)
- FLAD: Adaptive Federated Learning for DDoS Attack Detection (<https://arxiv.org/pdf/2205.06661.pdf>)

Alternatively:

- Propose a relevant paper of your interest



Introduction

Terms



Protection of systems, networks and programs from digital attacks

CYBERSECURITY



Hard-to-manage large volumes of data (e.g., network traffic, images, systems logs)

BIG DATA



Algorithms for automated data processing and analysis (patterns and trends)

DEEP LEARNING



Network intrusions

The term intrusion refers to an attempt by-pass the security mechanism of a computer or network with the following goal:

Compromise the confidentiality, integrity, availability of a resource



Intrusion detection

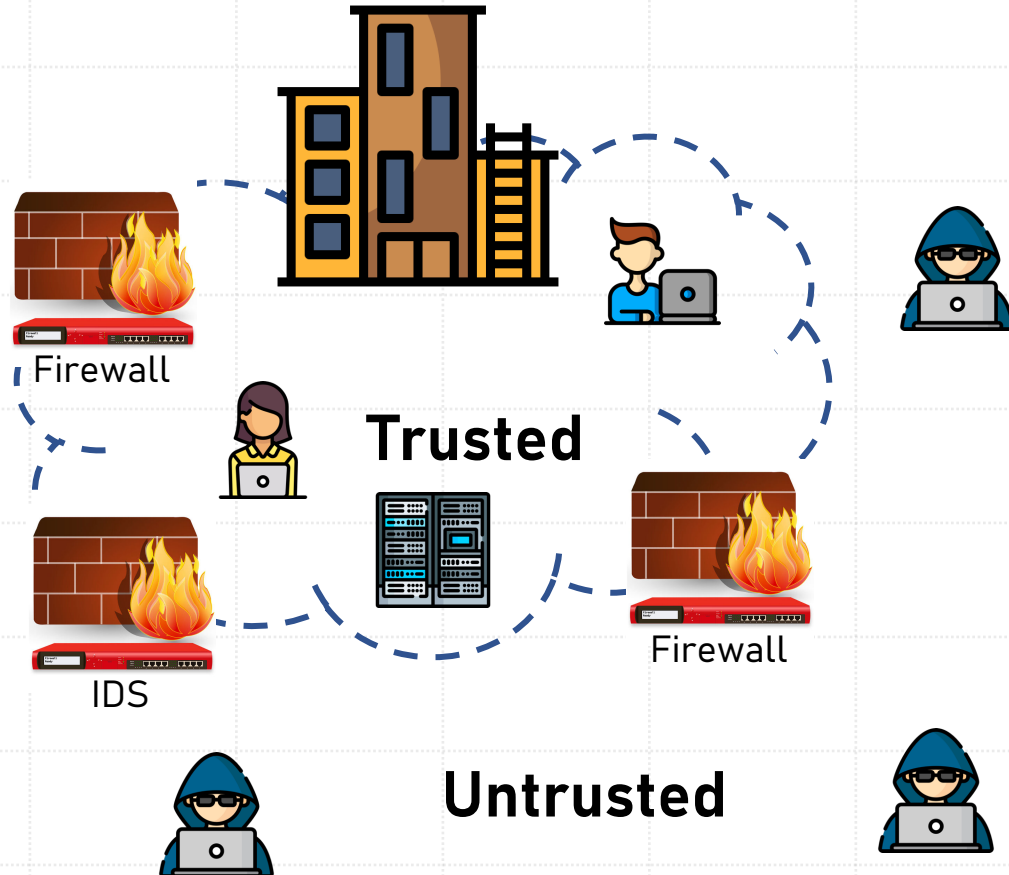
- An Intrusion Detection System (IDS) is a device or software application that monitors the network traffic for malicious activity.
 - **Network intrusion detection systems (NIDS):** A system that analyses the network traffic.
 - **Host-based intrusion detection systems (HIDS):** A system that monitors important operating system files, network activity and processes.



Intrusion detection types

- **Signature-based:** detection of possible threats by **looking for specific patterns**, such as byte sequences in network traffic, or known malicious instruction sequences used by malware. *Limitation:* cannot detect unknown attacks or variations of known attacks
- **Behavior-based:** statistical or ML methods designed to learn the **behaviour of normal and/or malicious activity** (either network or system activities or both).
 - **Anomaly detection:** the system looks for deviations from the normal patterns (typically using thresholds) to identifies malicious activity. *Limitation:* prone to false positives.
 - **Classification:** the system learns how to classify data into classes. Typically one class of normality and one or multiple classes of malicious behaviours.

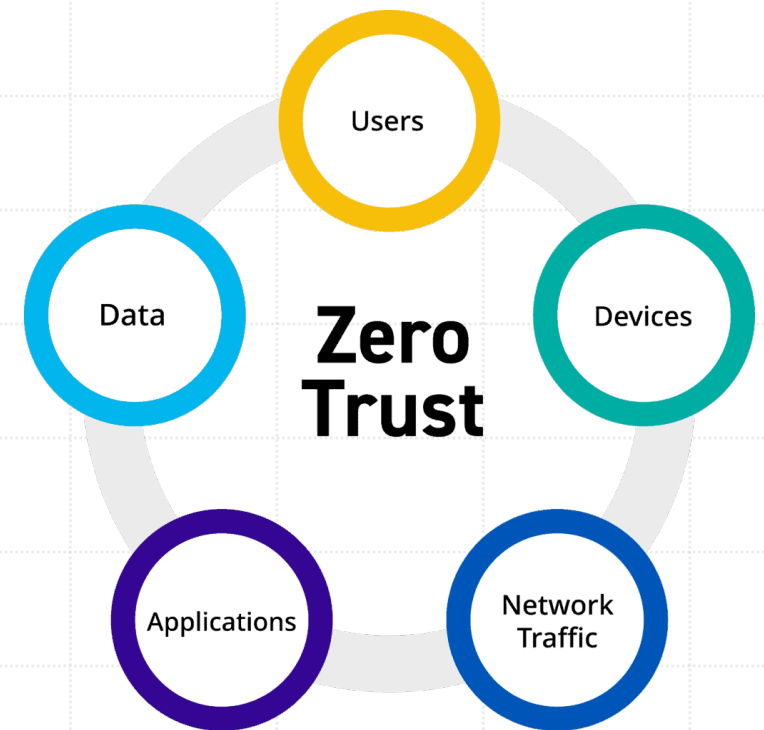
Intrusion detection approaches (1)



- **Perimeter-based:** traditional approach based on deploying dedicated security boxes to create a safe perimeter (e.g. firewalls, IDSs and De-Militarized Zones) and to centrally monitor connected nodes and devices in order to counter possible attacks.

Intrusion detection approaches (2)


- **Zero-trust:** security model based on the assumption that nothing, even inside an organisation's network can be trusted. This model does not rely less on the safety of the network perimeter, but rather on automated secure processes and technologies that can be applied directly to resources, irrespective of where they are located.





Intrusion types: overview

Understanding how cyberattack work helps in building better defences



Denial of Service attacks

A **Denial of Service attack (DoS)** is a cyberattack that

- exploits a software vulnerability or
- generates floods of traffic

with the attempt to make the victim host/network unavailable to legitimate users.

A **Distributed Denial of Service attack (DDoS)** is a type of DoS attacks that involves multiple connected online devices, collectively known as a botnet, which are used to overwhelm a target host or network.

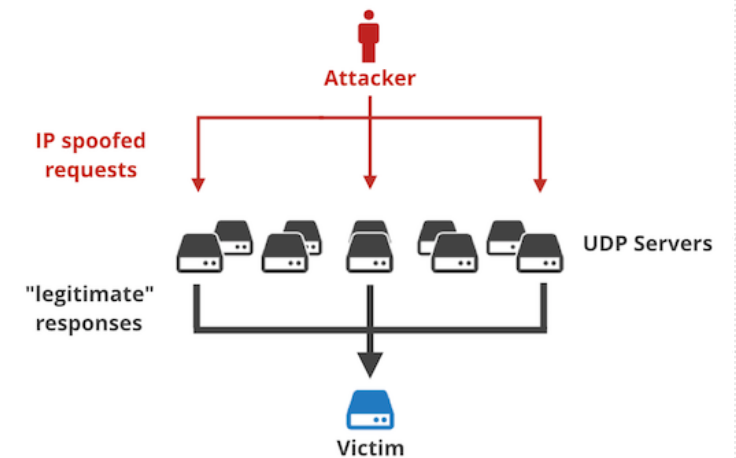
Classes of DDoS attacks

Volumetric attacks: try to overwhelm the resources of the target system (e.g., a web server) by flooding the victim with large volumes of traffic. In this case, the malicious traffic can exhaust the network, computing and memory resources of the system, preventing the normal activities

UDP-based reflection/amplification:

- UDP response/request protocols (DNS, NTP, etc.)
- Reflectors: DNS, NTP servers
- Victim's IP address

Result: congestion of victim's network with tens, hundred or more Gbps of malicious traffic



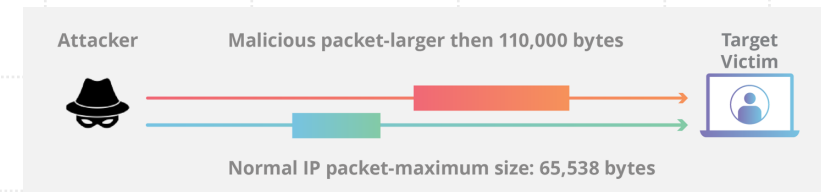
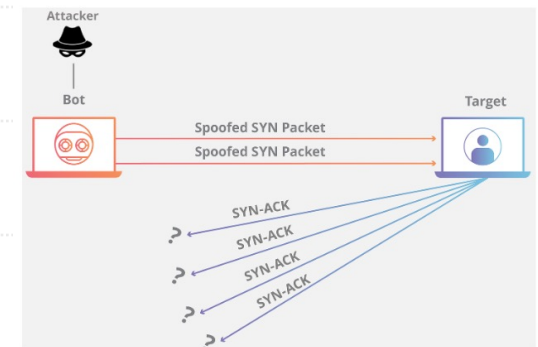
Source: cloudflare.com

Classes of DDoS attacks

Protocol attacks exploit the weaknesses or a specific behaviour of protocols. The objective of these attacks is to consume the computational resources of end hosts or crash/freeze them

SYN Flood attacks: exploit the three-way handshake of the TCP protocol. The attacker never responds to the SYN-ACK, causing half-open connection on the victim server, until the all available ports have been opened and the server cannot respond to legitimate users

Ping of death: the maximum packet size is 65535 bytes. Some TCP/IP systems were never designed to handle packets larger than the maximum. Anything that sends an IP datagram can be used for this exploit. That includes TCP and UDP transmissions.

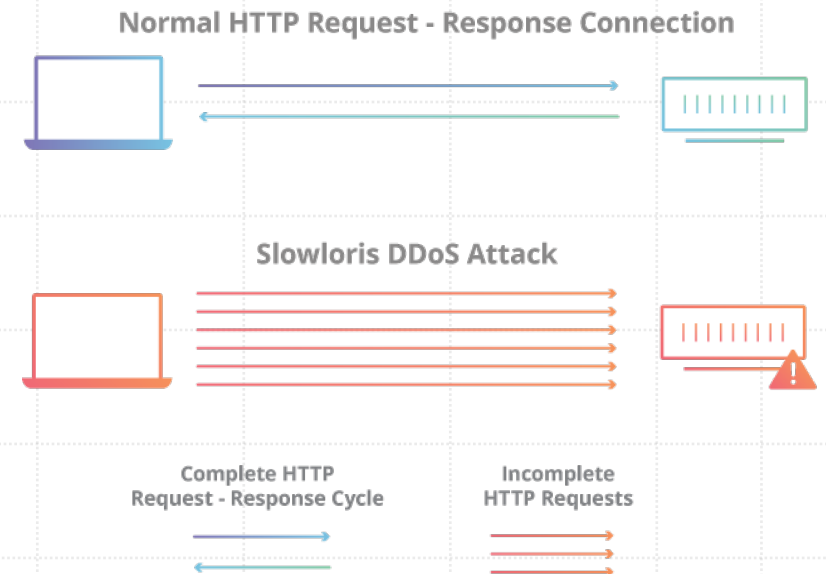


Classes of DDoS attacks

Application layer attacks, where the attacker exploits the specific logic of a web application to crash the server, making it inaccessible to the legitimate users. These attacks usually generate lower volumes of traffic compared to volumetric and protocol attacks. This aspect makes them harder to identify.

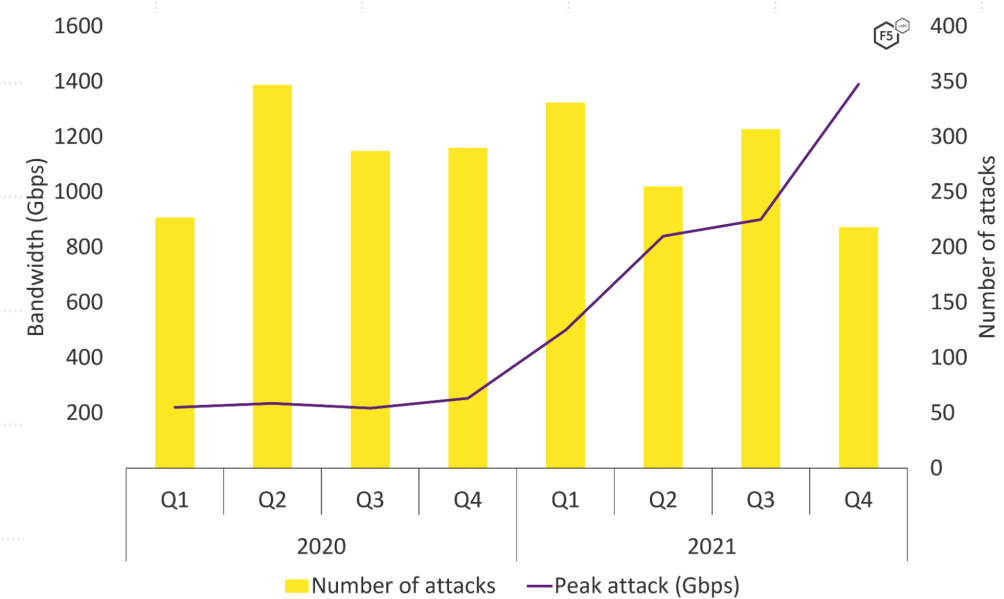
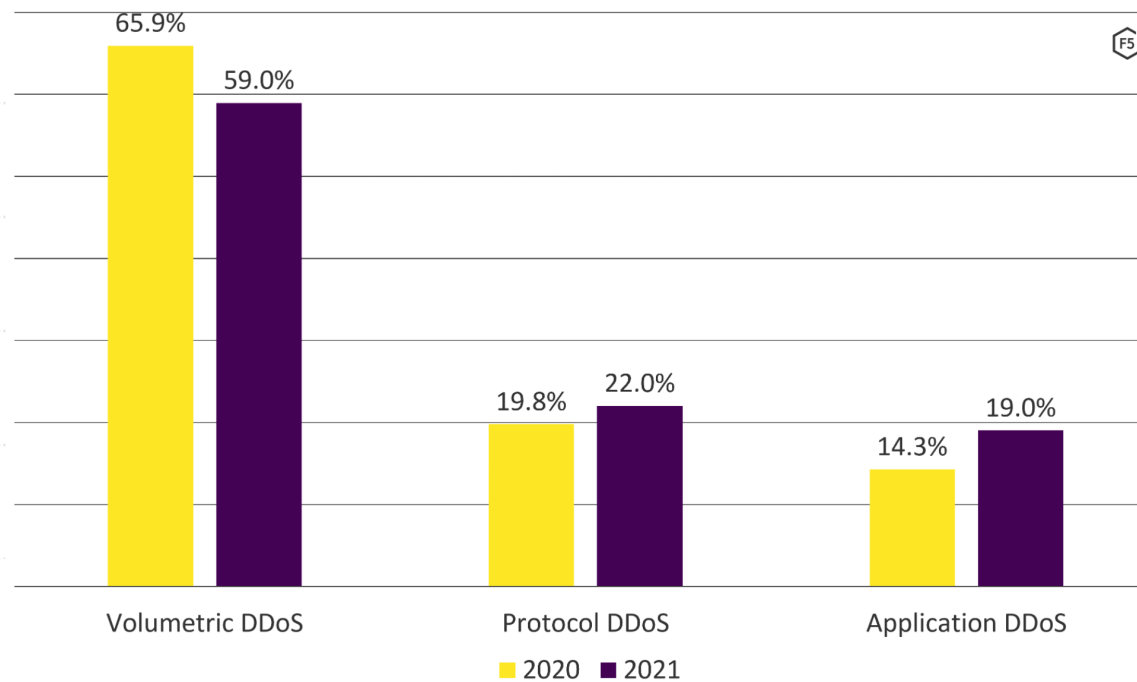
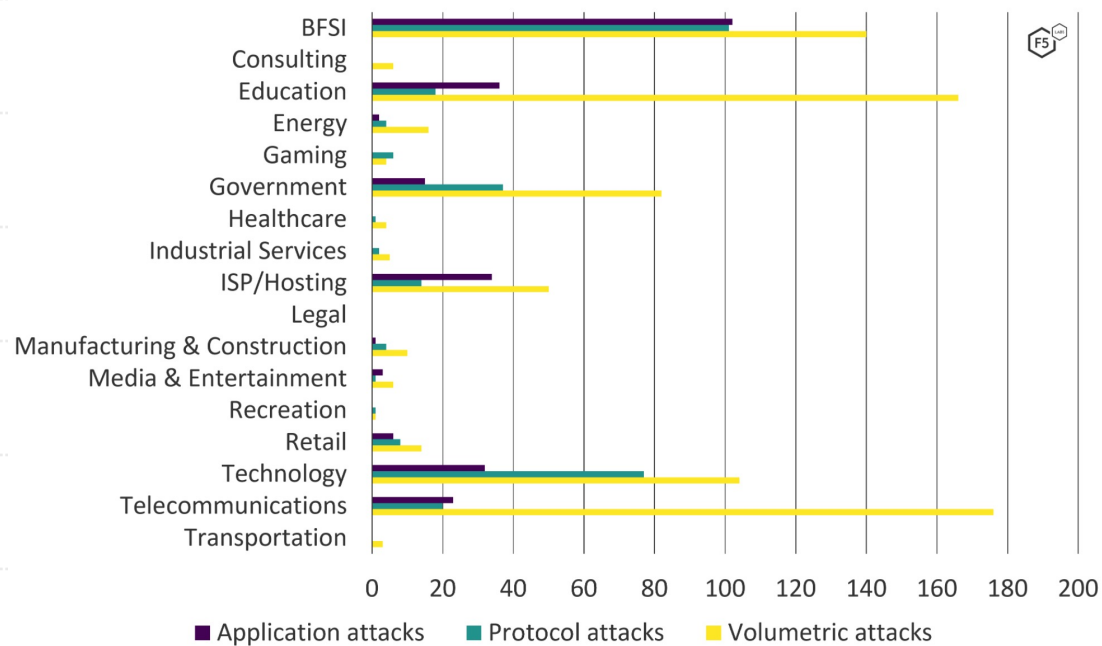
Slowloris: attacker to overwhelm a targeted server by opening and maintaining many simultaneous HTTP connections between the attacker and the target.

When the server's maximum possible connections has been exceeded (configurable, 100-200 by default), each additional connection will not be answered and denial-of-service will occur.



Source: cloudflare.com

Trend of DDoS attacks



DDoS attack types

Name	Type	Description
DNS, LDAP, MSSQL, NTP, Portmap, SNMP, TFTP, NetBIOS	Volumetric (reflection)	DDoS attacks that exploit a specific UDP-based network service to overwhelm the victim with responses to queries sent by the attacks to a server using the spoofed victim's IP address.
UDP Flood	Volumetric	Attack built with high rates of small spoofed UDP packets with the aim to consume the victim's network resources.
Syn Flood	Protocol	Attack that exploits the TCP three-handshake mechanism to consume the victim's resources with a flood of SYN packets.
Ping of Death	Protocol	The attacker sends oversized packets to the victim machine, i.e. packets larger than the maximum IPv4 packet size (65535 bytes). Any transport protocol can be used: ICMP, TCP, UDP, etc., as the issue is caused by the process that reassembles the IP packets.
Slowloris	Application	Attack based on partial HTTP requests against a Web server and on keeping those connections open as long as it can. This type of attack uses a low amount of bandwidth, with requests that mimic regular traffic.
HTTP Flood	Application	HTTP Flood attacks use legitimate HTTP GET or POST requests to force the HTTP server to allocate all its resources. It requires less traffic (compared to volumetric attacks) to bring down a server.

Other network attacks

Name	Type	Description
Brute Force	Password guessing	Attack based on trial-and-error approach to guess login info of a specific online service (HTTP, SSH, FTP, Telnet, etc.).
Port Scan	Port Scan	Port scanning is a method to understand which TCP/UDP ports of the victim are open to receive or send data. The goal of the attacker is to use this information to gain access to the victim machine, or to compromise it by sending malicious/malformed data.
Botnet	Malware	Communication between a Command & Control (C&C) server and a number of victim machines (bots) infected with malware. The traffic may include remote shell, file upload/download, key logging and other activities.
Spyware		Collects information about a device or network, then relays this data back to the attacker.
Backdoor Trojans		Creates a “backdoor” on your computer. It lets an attacker access your computer and control it.
Ransomware		Ransomware is a type of malicious software that encrypts files on a victim's computer, making them inaccessible. The attackers then demand a ransom payment in exchange for the decryption key needed to restore access to the files.
SQL injection	Code injection attacks	The attacker inserts malicious SQL code into a web form input field or a URL in order to gain unauthorized access to a database. The attacker's goal is to extract sensitive information, such as personal data or financial information, or to modify or delete data. These attacks are usually based on sending SQL queries to the server and get data or error messages that can be leveraged by the attacker.



Impact of cyber-attacks

- Cyberattack against the **Iranian nuclear program** in 2010:
 - A malware compromised Iranian PLCs, collecting information on industrial systems and causing the fast-spinning centrifuges to tear themselves apart
- Cyberattack against the **Ukrainian power grid** in late 2015:
 - A malware was able to alter the firmware of control devices, leaving thousands of users without electricity
- The volumetric DDoS attack in 2016 (600Gbps):
 - Leveraging unsecured IoT devices, targeting systems operated by DNS provider *DYN*
- The volumetric DDoS attack in 2018 towards Github(1.3Tbps):
 - Leveraged a vulnerability in *memcached*, a popular database caching tool.



Datasets

Popular datasets for research

- Canadian Institute for Cybersecurity (<https://www.unb.ca/cic/datasets/index.html>)
 - Recent dataset of DDoS, Brute force and other types of intrusions.
 - Raw traces (pcaps) produced with traffic generators + pre-processed features (csv)
- UNSW-NB15 dataset of various network intrusions (DoS, backdoors, etc.) (<https://research.unsw.edu.au/projects/unsw-nb15-dataset>)
 - Raw traces of synthetic traffic obtained with traffic generator tools + pre-processed features
- MAWI dataset of daily traces captured on the link between Japan and USA (<http://mawi.wide.ad.jp/mawi/>)
 - Anonimised IP addresses and omitted payloads



Canadian Institute for Cybersecurity

- Main page: <https://www.unb.ca/cic/datasets/index.html>
- Various types of data, including **network traffic, to memory dumps**
- Datasets of benign and malicious network traffic collected in **dedicated testbeds**
- **Benign traffic** is generated based on models of real traffic
- **Malicious data** include network intrusions (DDoS, Brute Force attacks, port scans, etc.), malware memory dumps (trojan, spyware, ransomware)

Pros:

- Freely available for testing
- Datasets of network traffic include pcap files (tcpdump/Wireshark) and labels
- Widely used in research works



Recent datasets from CIC

- Intrusion Detection Evaluation Dataset (CIC-IDS2017)
(<https://www.unb.ca/cic/datasets/ids-2017.html>)
- Intrusion Detection Evaluation Dataset (CSE-CIC-IDS2018)
(<https://www.unb.ca/cic/datasets/ids-2018.html>)
- DDoS Evaluation Dataset (CIC-DDoS2019)
(<https://www.unb.ca/cic/datasets/ddos-2019.html>)

Flow-based features (CIC-IDS2017)

ICMP = 1
TCP = 6
UDP = 17

Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	Timestamp	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets
192.168.10.5-104.16.207.165-54865-443-6	104.16.207.165	443	192.168.10.5	54865	6	7/7/2017 3:30	3	2	0	12	0
172.16.0.1-192.168.10.50-49650-80-6	172.16.0.1	49650	192.168.10.50	80	6	7/7/2017 3:56	1293792	3	7	26	0

Flow Bytes/s	Flow Packets/s	Flow IAT Mean	Flow IAT Std	Flow IAT Max	Flow IAT Min	Fwd IAT Total	Fwd IAT Mean	Fwd IAT Std	Fwd IAT Max	Fwd IAT Min	Bwd IAT Total	Bwd IAT Mean	Bwd IAT Std
4000000	666666.6667	3	0	3	3	3	3	0	3	3	0	0	0
8991.398927	7.72921768	143754.6667	430865.8067	1292730	2	747	373.5	523.9661249	744	3	1293746	215624.3333	527671.9348

Subflow Bwd Bytes	Init_Win_bytes_forward	Init_Win_bytes_backward	act_data_pkt_fwd	min_seg_size_forward	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	33	-1	1	20	0	0	0	0	0	0	0	0	BENIGN
11607	8192	229	2	20	0	0	0	0	0	0	0	0	DDoS

Packet-based features

- Extracted from the traffic traces with tools like **tcpdump** and **tshark**
- In this example, a HTTP DDoS flow is represented as a list of packets (lines)
- Each packet is represented as a list of packet-level features: (0) IAT, (1) Pkt_Len, (2) Highest_Protocol, (3) IP_Flags, (4) Protocols, (5) TCP_Len, (6) TCP_Ack, (7) TCP_Flags, (8) TCP_Win, (9) UDP_Len, (10) ICMP_Type

	0	1	2	3	4	5	6	7	8	9	10
0	0.00000	60.00000	78354535.00000	16384.00000	576.00000	0.00000	0.00000	2.00000	65535.00000	0.00000	0.00000
1	0.00000	60.00000	78354535.00000	16384.00000	576.00000	0.00000	0.00000	2.00000	65535.00000	0.00000	0.00000
2	0.00017	60.00000	78354535.00000	16384.00000	576.00000	0.00000	1.00000	18.00000	28960.00000	0.00000	0.00000
3	0.00017	60.00000	78354535.00000	16384.00000	576.00000	0.00000	1.00000	18.00000	28960.00000	0.00000	0.00000
4	0.00032	52.00000	78354535.00000	16384.00000	576.00000	0.00000	1.00000	16.00000	64.00000	0.00000	0.00000
5	0.00033	52.00000	78354535.00000	16384.00000	576.00000	0.00000	1.00000	16.00000	64.00000	0.00000	0.00000
6	0.06014	508.00000	11875138.00000	16384.00000	592.00000	456.00000	1.00000	24.00000	64.00000	0.00000	0.00000
7	0.06015	508.00000	78354535.00000	16384.00000	576.00000	456.00000	1.00000	24.00000	64.00000	0.00000	0.00000

Packet header analysis

Wireshark output

```
[Protocols in frame: eth:ethertype:ip:tcp]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]
> Ethernet II, Src: Dell_a2:c0:d3 (90:b1:1c:a2:c0:d3), Dst: Cisco_42:73:e8 (70:f3:5a:42:73:e8)
> Internet Protocol Version 4, Src: 192.168.50.1, Dst: 172.16.0.5
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0x0000 (0)
> Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x9bfd [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.50.1
    Destination: 172.16.0.5
> Transmission Control Protocol, Src Port: 80, Dst Port: 52003, Seq: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 52003
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0 (relative sequence number)
    Sequence number (raw): 3386762369
    [Next sequence number: 1 (relative sequence number)]
    Acknowledgment number: 1 (relative ack number)
    Acknowledgment number (raw): 2185688371
    1010 .... = Header Length: 40 bytes (10)
> Flags: 0x012 (SYN, ACK)
    Window size value: 28960
```

The Syn Flood use-case

Length	Info
74	15805 → 22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1285818975 TSecr=0 WS=128
74	22 → 15805 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=103691197 TSecr=1285818975 WS=128

Length	Info
1058	22342 → 80 [SYN] Seq=0 Win=10000 Len=984 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]
834	40629 → 80 [SYN] Seq=0 Win=10000 Len=760 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]
1058	22343 → 80 [SYN] Seq=0 Win=10000 Len=984 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]
834	40630 → 80 [SYN] Seq=0 Win=10000 Len=760 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]
1058	22344 → 80 [SYN] Seq=0 Win=10000 Len=984 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]
834	40631 → 80 [SYN] Seq=0 Win=10000 Len=760 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]
1058	22345 → 80 [SYN] Seq=0 Win=10000 Len=984 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]

Length	Info
797	1203 → 80 [SYN] Seq=0 Win=10000 Len=723 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]
112	4795 → 80 [SYN] Seq=0 Win=10000 Len=38 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]
655	29608 → 80 [SYN] Seq=0 Win=10000 Len=581 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]
112	4796 → 80 [SYN] Seq=0 Win=10000 Len=38 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]
797	1204 → 80 [SYN] Seq=0 Win=10000 Len=723 MSS=1452 SACK_PERM=1 TSval=422940867 TSecr=0 WS=32 [TCP segment of a reassembled PDU]



Concluding remarks

- Deep Learning is a powerful technology that can be exploited in cybersecurity to implement automated methods that **learn attack patterns** in the data (e.g., traffic, system logs) and **prevent similar malicious activities**.
- The implementation of DL solutions for cybersecurity requires a **deep analysis of the data** at hand, to avoid unpredictable results
- In science, publicly available datasets are important for initial testing and state-of-the-art comparison. However, they are **hardly suitable for real-world applications**.