Network packet generation for Artificial Intelligence

Bernhard Gally
IT Security
FH Technikum Wien
Vienna, Austria
cs19m023@technikum-wien.at

Maximilian Wech
IT Security
FH Technikum Wien
Vienna, Austria
cs19m020@technikum-wien.at

***Abstract*—Supervised learning in artificial intelligence requires a training dataset that is already labeled. Doing so with network packets is a challenge due to the fact that even Intrusion Detection Systems are not always correct with their guessing. To overcome this problem it is necessary to generate malicious and benign packets and label them accordingly. To increase the quality of the dataset the IP addresses from the machines should not remain the same for all of the data and the malicious and benign traffic should be mixed.**

***Index Terms*—artificial intelligence, neuronal networks, computer networks, network security, packet generation**

I. Introduction

Generating network traffic that can be used to train a neuronal network brings some challenges: there are many different attack types that need to be considered, a vast variety of attack tools to generate malicious traffic are available in the Internet each with their own advantages and disadvantages and generating realistic network traffic that does not get mislabeled by intrusion detection systems just to name a few.

This paper is split into three chapters. The first chapter explains the different kinds of attacks and their targets. The second chapter focuses on tools to generate malicious traffic. The third chapter addresses the problem of generating useful network traffic.

At this point all of our considerations are based on theoretical research without practical implementation which will come to life in further work. Therefore this paper sets a direction and foundation for the next steps and shows possible solutions at a high level.

II. Cyber-attacks: types, targets, vulnerabilites

This chapter intends to provide an overview of various types of cyber-attacks, cyber-attack targets and exploitable vulnerabilities.

A. General information

In order to fulfil the task, it initially appears necessary to describe typical characteristics of attacks. Attacks aim to exploit weaknesses, such as poorly configured systems, or errors in software and to circumvent security measures. Attackers deliberately try to make services unavailable, spy on or manipulate data and disrupt normal traffic as much as possible [1]. Executing attacks generally consists of several steps: First, the attacker's motivation must be clarified (e.g. proof of your own abilities). Then it is necessary to select a target and collect information about it. As soon as this has happened, the attack method is selected and the attack is performed [2].

B. Classifying attacks

Basically, there are too many different cyber-attack methods to mention all of them. Therefore, various approaches exist in the literature to classify attacks into different groups. One way is to split attacks into active and passive ones [3], [4]. In the case of passive attacks, attempts are not made to manipulate data in network traffic, but rather to eavesdrop and collect sensitive information. The opposite is the case with active attacks, which attempt to manipulate network packets, for example [4].

At this point it also appears necessary to address the distinction between computer and network attacks mentioned in [2]. Computer attacks are spoken of when a computer system is attacked in any way. For example, attempts are made to slow down the performance of PCs or to manipulate data on hard disks. Network attacks are divided into two variants: The first option is to use the network to attack a computer system. Although these are network attacks, they can also be seen as a subset of computer attacks. The second option is when network attacks are actually directed only at the network and not at computer systems, for example by congesting a network with packets. The following figure is included to provide clarity and shows the relationship between computers and network attacks.

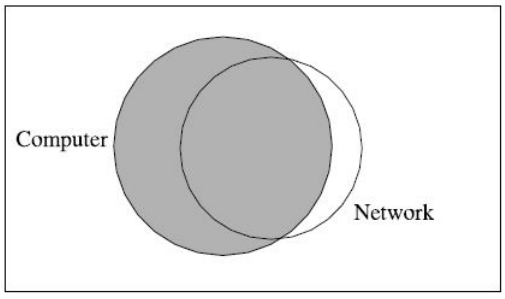

Fig. 1. Ratio of computer and network attacks [2]

Further approaches exist such as classifying into internal and external attacks, or classifying attacks based on the underlying OSI-layer.

In the literature often cited is the classification model presented in [5], which works on the basis of four dimensions and offers a holistic view of attacks. First, an attack class is assigned based on the attack vector. The attacks target is specified in the second dimension. The vulnerabilities that are exploited by the attack are then noted. Finally, the fourth dimension analyses whether an attack has a payload or an effect that exceeds the typical characteristics of the type of attack [5] . The rest of this chapter will focus on further introducing this model and consider each dimension individually.

1) First dimension: As already mentioned, in the first dimension the attack class is determined on the basis of the attack vector. The following generic classifications are listed in [5]:

Virus: this is a malware that replicates itself and, for example, infects other script code. When the infected source code is executed, the virus is also executed [6]. Basically there are many different types of viruses. An example would be File Infectors, in which a virus is embedded in a file. If this file is executed, the virus is also executed [2].

Worms: a software that spreads independently by exploiting vulnerabilities [6]. Examples include mass mailing worms, which spread via email, and network-aware worms, which only require four steps to infect a foreign host and spread themselves [2].

Trojans: a software that appears harmless at first, but also contains a hidden, malicious function [6]. One example are logic bombs, which only execute their malicious payload when a specific condition (e.g. a certain time of a day) has been reached [2].

Buffer overflows: This type of attack is based on programming errors and is very often used to attack computers or networks. Principally an attempt is made to overfill a temporary memory area. The amount of data that exceeds the size of the memory overwrites and corrupts other data. One possible goal of this attack would be to take control of a process. The main types of this attack are stack and heap buffer overflows [2].

Denial of service attacks: this means attacks on the availability of systems. An attempt is made to exhaust the resources required to run a service [6]. This attack can be host-based, network-based, or distributed. The latter version is a DDoS attack, in which several attackers attack a mutual target at the same time [2].

Network attacks: the category is divided into several subareas. If an attacker pretends to be someone else by manipulating certain addresses, this is called spoofing. This can take place on the basis of the MAC address (MAC Address Spoofing) or on the basis of the IP address (IP Spoofing). An attack in which IP spoofing is used is session hijacking. Here, an attacker tries to link into a session (based, for example, on TCP / Telnet) between two hosts and virtually replace one host. In the case of wireless network attacks, weaknesses such as too short initialization vectors of the WEP standard can be exploited. In the event of attacks on web applications, attempts are made to exploit security gaps in them. For example, if there is no strict separation between user input and program instructions, or there is insufficient input validation, an attacker can successfully send manipulated SQL statements to the underlying database. The consequences would be spying out of data or losing control of the underlying server. [2].

Physical attacks: this type of attack aims to cause physical damage to target systems (i.e. computers or networks). This includes, for example, deliberately cutting network cables or destroying computers [2]. Password attacks: here an attempt is made to find out a specific password in order to gain control over a PC, for example. This can be done in three different ways: trying out all possible password combinations (brute force), trying out a part of all possible password combinations (e.g. testing simple passwords) and determining deficiencies in password storage / encryption [2], [5].

Information gathering attacks: this type of attack does not change data; it is only about spying on (sensitive) information, which can be used for follow-up attacks [5]. A possible variant of this is sniffing of packets, to find out about IP and MAC addresses or passwords [2].

2) Second dimension: The targets of attacks are described in the second dimension of the classification model. These should be described as specifically as possible, which means that it is not sufficient, for example, to specify which server was attacked, but rather which service or which operating system is affected with which version. It should also be noted that an attack can target more than one target. Basically three groups of attack targets are defined [5]:

Hardware: this group is divided into three areas. First of all, computers and their components (e.g. processors, random access memory) should be mentioned. Network devices such as routers or switches also belong to this category. Ultimately, peripheral devices such as monitors and keyboards are also included.

Software: here a distinction is made between targets on operating systems (Windows, Linux, etc.) and various applications (email client, text processing program) that are executed on an operating system.

Network: in this group, the target is not software or hardware but the network itself, or an associated network protocol (for example, TCP).

3) Third dimension: In the third dimension of the classification model [5], vulnerabilities that can be exploited by attacks are considered. Reference is made to the industry standard CVE - Common Vulnerabilities and Exposures [7], which is used for the uniform naming of security gaps and vulnerabilities in computer systems. General vulnerability types are also defined in [8]:

Implementation: if a system is not implemented according to the design, security gaps arise.

Design: an incorrect design leads to a security-threatening implementation.

Configuration: If a system is not configured correctly, doors are opened to the attackers.

4) Fourth dimension: The last dimension analyzes whether an attack has a payload or an effect that exceeds the typical characteristics of the type of attack. This should result in a detailed view of the payload. For example, a worm's payload might be a Trojan, or the worm might just be manipulating files [5].

The classification model offers a comprehensive view of attacks. For example, for a worm that is distributed via email (dimension 1), the target Windows 95 (dimension 2) with an inadequate configuration (dimension 3) and a virus as payload (dimension 4) is determined. Further dimensions could be specified, for example what damage is caused by an attack, or how many costs arise from successfully carrying out the attack. Overall an overview is provided of how attacks can be classified, which attack targets are possible, or which vulnerabilities exist.

III. Tools

There is a large number of network tools available in the world wide web. Many of them are free to download and can be used for malicious activities but there are also many that are for defensive purposes. Generally speaking there are two major groups: tools for network attackers and tools for network defenders. [1] The tools for network attackers are the ones generating the packets thus being the ones we will focus on.

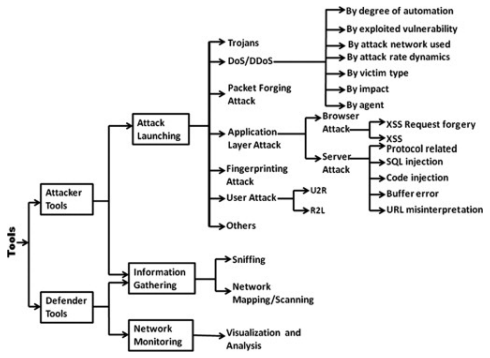

Fig. 2. Taxonomy of network security tools [1]

A. Information gathering

Attackers usually collect information before starting their attack. Understanding the environment, what operating systems are used, finding open ports and available services are only a small part of information that can be obtained.

Table I contains a list of sniffing tools that are available online. Generally speaking sniffing tools do not generate packets and are therfore undetectable. However, on a switched ethernet network segment a sniffer does generate a small amount of packets such as ARP [9]. Other types of sniffers, such as Ettercap, can execute a Man-in-the-middle attack. This causes a lot of packets to be sent through the network as it needs to answer requests in order to spoof an identity.

TABLE I
SNIFFING TOOLS [1]

Tool name	Purpose
Ethereal	Packet capturing
Tcpdump	Packet capturing
Net2pcap	Packet capturing
Snoop	Packet capturing
Angst	Sniffing
Ngrep	Capturing packet
Ettercap	Man-in-the-middle attack
Dsniff	Password sniffing
Cain & able	Password recovery
Aimsniff	Capturing packet
Tcptrace	Analysis of traffic
Tcptrack	TCP connection analysis
Argus	Analysis of audit data
Karpski	Packet analyzer
IPgrab	Display packet header
Nast	Traffic analysis
Gulp	Packet capturing/visualization
Libpcap	Packet capturing
Nfsen	Flow capturing/ visualization
Nfdump	Flow capturing/visualization

Scanning tools are often more intrusive than sniffing tools and generate lots of packets that can be detected. Their aim is to show up open ports, available services, used operating systems. With the gathered information an attacker can pick a suitable attack vector. Table II shows a few common network scanning tools.

TABLE II
SCANNING TOOLS [1]

Tool name	Purpose
Nmap	Scanning
Amap	Scanning
Vmap	Version mapping
Unicornscan	Scanning
Ttlscan	Scanning
Ike-scan	Host discovery
Paketto	Scanning

B. Attack launching

The most packets however are generated by the attacks themselves. Especially some kinds of DoS attacks are designed to overload the network with packets such that the victim cannot respond to legitimate requests.

1) Denial of service: The Internet provides many tools that are capable of executing DoS attacks and can be easily used to crash hosts or even networks. Two examples of such tools are Low Orbit Ion Cannon (LOIC) or High Orbit Ion Cannon (HOIC). Both of these tools are capable of launching DoS/DDoS attacks within a short amount of time. LOIC supports TCP, UDP and HTTP whereas HOIC only supports the HTTP protocol [1]. A list of DoS and DDoS tools is shown in Table III.

2) Packet forging: Packet forging tools are necessary to manipulate packet information. With the help of such tools it

TABLE III
DOS/DDOS TOOLS [1]

Tool name	Purpose
Jolt	DoS
Burbonic	DoS
Targa	DoS
Blas20	DoS
Crazy Pinger	DoS
UDPFlood	DoS
FSMax	DoS
Nemsey	DoS
Panther	DoS
Land & LaTierra	DoS
Slowloris	DoS
Blackenergy	DDoS
HOIC	DDoS
Shaft	DDoS
Knight	DDoS
Kaiten	DDoS
RefRef	DDoS
Hgod	DDoS
LOIC	DDoS
Trinoo	DDoS
TFN	DDoS
TFN2K	DDoS
Stachaldraht	DDoS
Mstream	DDoS
Trinity	DDoS

is possible to generate traffic with non-existant IP addresses. They also allow to individually set the flags of network packets to a desired state. An extreme example of such a packet is the "Christmas tree packet", which has all available flags set to 1. A list of packet forging tools is shown in Table IV.

TABLE IV
PACKET FORGING TOOLS [1]

Tool name	Purpose
Packeth	Packet generator
Packit	Packet analysis and injection
Nemesis	Packet crafting/injection
Tcpinject	Packet generator
Libnet	Packet injection
SendIP	Packet generator
IPsocery	Packet generator
Pacgen	Packet generator
Arp-sk	ARP packet generator
ARP-SPOOF	Packet intercept
Libpal	Packet intercept
Aicmspend	ICMP packet flooding

3) Application layer: Application layer attacks use legitimate requests to use up all the resources of the victim. Unlike DDoS attacks, they are more subtle and more difficult to detect since they are using legitimate requests.

IV. GENERATING A REALISTIC NETWORK LOAD

A problem that exists with generating a realistic network load is the unwanted burstiness of real networks. Misconfigurations can result in traffic that looks similar to flooding attacks [10] and are hard to reproduce in small sandbox networks. Existing datasets often do not include such noise which result in unrealistic datasets [10].

According to [10] it might be impossible to create a single representative background trace that contains benign and malicious traffic. Reason being that the expected traffic workload and characteristics are highly dependent on the network and its hosts.

Tools such as Trident provide multiple strategies to create benign traffic [10].

A. NIDS-based strategies

Using a wellknown network intrusion detection system (NIDS) such as Snort they groom a packet trace from a local site. This approach causes problems since benign packets that are flagged as malicious (false postives) are removed.

B. Synthetic generation strategies

Software robots emulate user behaviour and thus generate synthetic traffic. The idea behind it is that the robot only creates connections with known good hosts.

C. Trust-based strategies

The third strategy they support is by grooming a packet trace taken at a local site using connection heuristics. Examples of these heuristics are failure rates or scanning characteristics. This approach makes use of different connection characteristics of malicious and benign sources based on a model of malicious connection behaviour [10]. The advantage of using this strategy is that it is not influenced by a particular system (NIDS independence) and is based on transport level characteristics. The disadvantage is that a connection that is sufficiently similar to benign users will not be detected.

V. SUMMARY AND CONCLUSION

REFERENCES

[1] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya und J. K. Kalita, "Network attacks: Taxonomy, tools and systems", Journal of Network and Computer Applications, 2014. Jg., Nr. 40, 2014.

[2] S. Hansman, "A Taxonomy of Network and Computer Attack Methodologies", 2003.

[3] M. W. Pawar and J. Anuradha, "Network Security and Types of Attacks in Network", Procedia Computer Science, Nr. 48, 2015.

[4] A. Gagandeep and K. Pawan, "Analysis of Different Security Attacks in MANETs on Protocol Stack A-Review", International Journal of Engineering and Advanced Technology (IJEAT), 2012.

[5] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks", Computers & Security, Nr. 24, 2004.

[6] W. Stallings und L. Brown, "Computer Security: Principles and Practice", 3rd Ed., 2015.

[7] Common Vulnerabilities and Exposures. [Online] Available at: https://cve.mitre.org/.

[8] J. D. Howard, "An Analysis Of Security Incidents On The Internet", 1995.

[9] R. Spangler, "Packet Sniffer Detection with AntiSniff", 2003

[10] J. Sommers, V. Yegneswaran and P. Barford. "Toward Comprehensive Traffic Generation for Online IDS Evaluation". Technical report, Dept. of Computer Science, University of Wisconsin, August 2005.