ESSLINGEN UNIVERSITY

For people and technology.

IoT-Honeypot

Analyzing attacks against IoT-Protocols

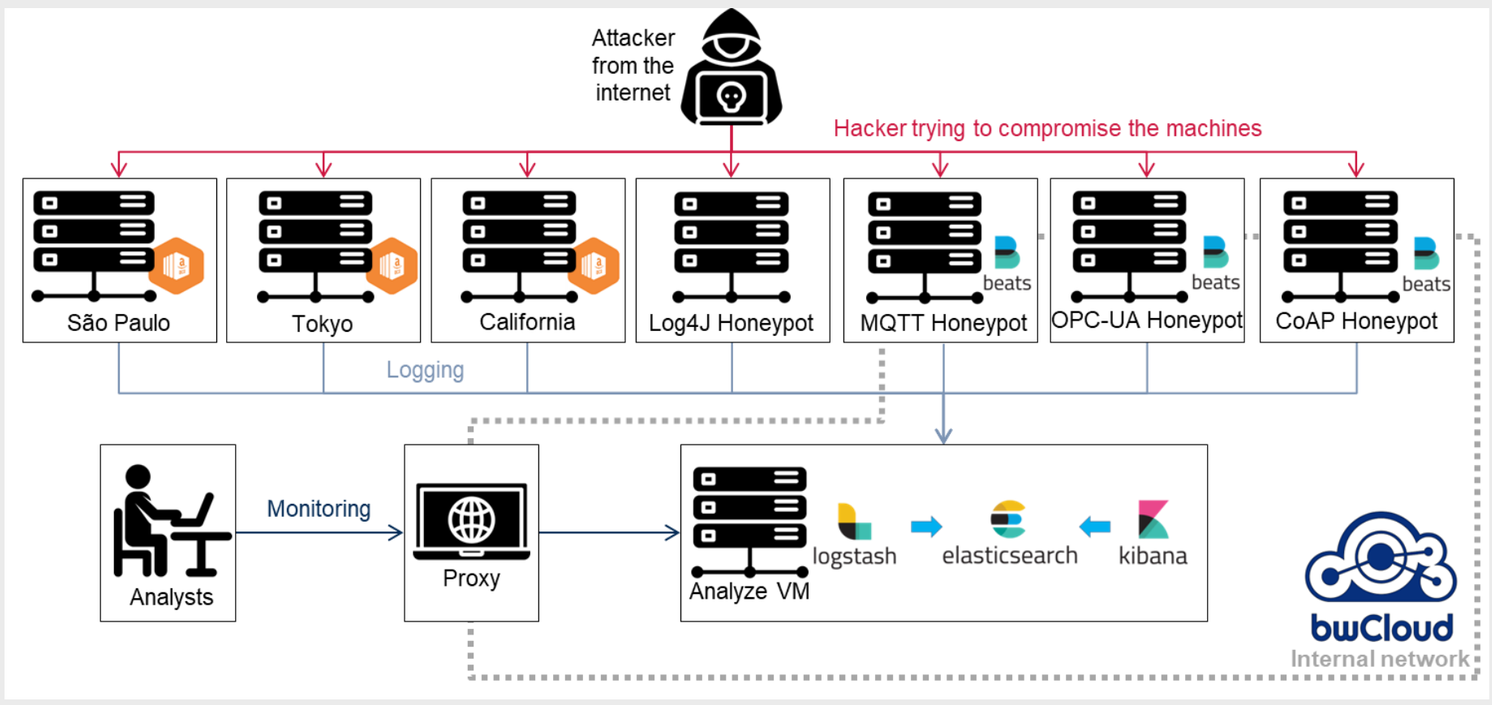
Motivation

Cybersecurity is getting more attention each year. To get an image of current attack strategies, Honeypots are deployed. Honeypots are purposely insecure systems, which will react automatically to a potential attack.

The digitalization that comes with the Industry 4.0 and the Internet of Things (IoT) is offering another potential attack surface. This project focuses on analyzing the new attack strategies for systems based on IoT protocols.

Infrastructure

Our project is split into Instances by AWS, that are spread over different continents, Honeypots for relevant IoT Protocols and a simple Log4J Honeypot due to its relevance.
All systems are logging relevant data for its protocol. The logs are then sent to the ELK Stack where it can be analyzed through the Kibana dashboard.
A proxy server prevents SSH access to the instances through the internet.
The infrastructure – except for the AWS Instances – is hosted by bwCloud.



Project Infrastructure

IoT Protocols - Overview:

MQTT	Publisher-Subscriber Architecture, Topics, Authentication/Encryption possible, TCP
OPC-UA	Client-Server/Publisher-Subscriber Architecture, Authentication/Encryption possible, TCP
CoAP	Client-Server Architecture, HTTP-Like, Encryption possible, UDP

The Message Queuing Telemetry Transport Protocol, short **MQTT**, is an easy and lightweight protocol designed for devices with low internet capacities like sensors. Therefore, it is used heavily in Machine-to-Machine communication. Secure communication is available in form of TLS over SSL or authentication (Login).[1] Our goal for this protocol is to see if attacks happen for secure MQTT (like brute force) or if unknown clients publish data to the broker.

The Open Platform Communications – Unified Architecture Protocol **OPC-UA** is one of the most important Protocol in the Industry 4.0. It allows standardized and platform independent access and communication between Machines or Systems. It offers secure communication with authentication or TLS.[2] Of the three presented, It is the most flexible protocol. We want to analyze the different implementations to show different behavior by possible attackers.

Constrained Application Protocol **CoAP** is a specialized web transfer protocol, that is designed for nodes with limited resources in constrained network. It can easily be translated to HTTP for simple web integration.[3] Since CoAP is using UDP instead of TCP, it has a few security issues. Due to misconfiguration, a CoAP Service can be vulnerable to Denial of Service (DoS) by amplification attacks. Analyzing attacks and strategies are the goal for this protocol.

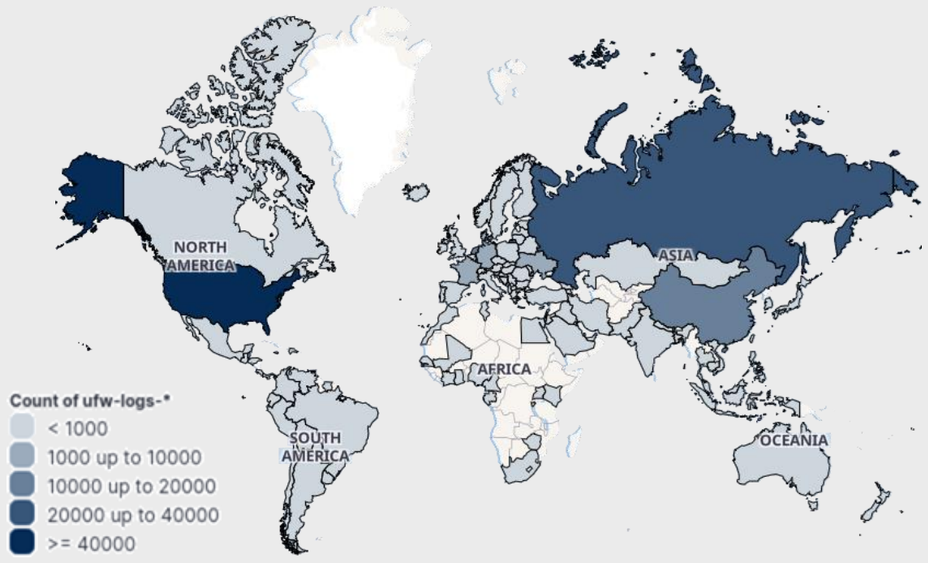
Andreas Lautner, Carmen Schäffler
Tutor: Prof. Dr. Tobias Heer, Lukas Wüsteney

ESSLINGEN UNIVERSITY

For people and technology.

Results and Geographical differences

The number of daily connections attempts to the MQTT-client are almost constant, but there are several peaks in the time between 22.12.2021 and 05.01.2022 after the Log4J vulnerability was discovered.
The quantity of daily connection attempts to the CoAP-client is similar to that. Furthermore, a security scanner marked this client as vulnerable.
The OPC-UA-client did not respond any logs about connection attempts or successful connections during our logging time.
In contrast to the increasing log quantity to the MQTT-client only a few days after the Log4J vulnerability was discovered, the connection attempts to our Log4J honeypot increased at first more than two weeks later.
Three Amazon Instances are collecting ufw-logs. The instances are located in São Paulo, Tokyo and California.
Most of the connections worldwide are made from Russia and from the United States to these instances.

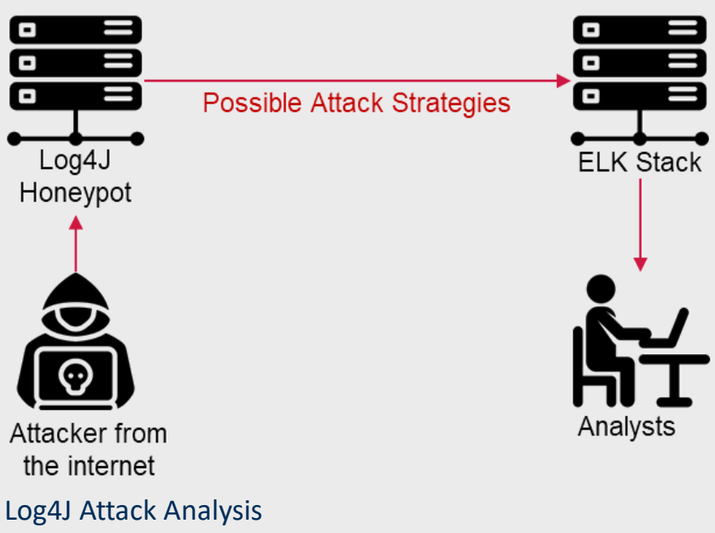


Origin of the connections made to the Amazon Instances

The most connections from Europe were tried to make from the Netherlands and Denmark. As São Paulo was the first instance we deployed, approximately 50 percent of the connections were made to São Paulo.

Log4j Vulnerability

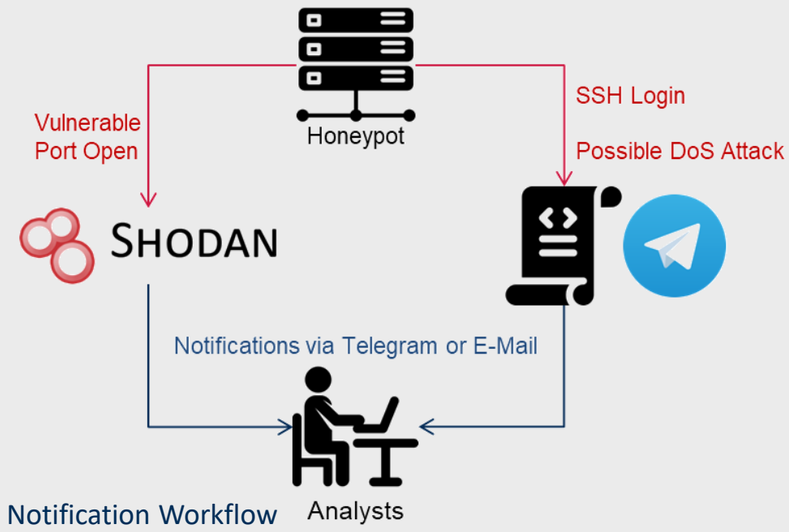
The critical Log4J vulnerability was discovered in December 2021. Because it is a rather new vulnerability, there is only few knowledge about it and attack strategies develop vastly. We created a Log4J Honeypot which gathers information on possible attacks.



Log4J Attack Analysis

Monitoring

A vulnerable system needs to be monitored. Therefore, we developed methods to get notifications via Telegram for unwanted logins via SSH and when a service has unusually much activity in a short time. In addition to that we monitor all Instances with Shodan.



Notification Workflow

Future Research

Our plan for the next semester is to improve the logging for every service to gather only relevant information. With only the relevant information, we are able to assign specific patterns to possible attack strategies. In addition to that, we want to investigate the OPC-UA protocol more, since there hasn't been much activity. We also want to exclude any possible implementation errors with the server. Because of the security issues with CoAP, we will try to furthermore ensure that it will not be vulnerable to our host bwCloud. To expand our project, we will research more about other relevant IoT protocols like **XMPP** and **AMQP**. If they are suitable, we will setup more honeypots for them and analyze attacks for more protocols. Optionally we will look more into the Log4J Honeypot and check other implementations.

References

[1] – Hillar, Mqtt Essentials: A lightweight IOT protocol: The preferred IOT publish-subscribe Lightweight Messaging Protocol 2017
[2] – Burke, OPC Unified Architecture Interoperabilität für Industrie 4.0 und das Internet der Dinge, 2016
[3] – Shelby et al., The Constrained Application Protocol (CoAP), 2014, URL: https://datatracker.ietf.org/doc/html/rfc7252

Andreas Lautner, Carmen Schäffler
Tutor: Prof. Dr. Tobias Heer, Lukas Wüsteney