Developing a security framework for robots

Master Thesis

UNIVERSITY OF TWENTE.

Oleksandr Shyvakov

Supervisor at UTwente: Dr. Andreas Peter

Supervisors at Deloitte: Gijs Hollestelle
Sander Maas

Faculty for Electrical Engineering, Mathematics and Computer Science
Department of Computer Science and Electrical Engineering
University of Twente

August 2017

To my family ...

Acknowledgements

This thesis would not be possible without all the guidance and support I received.

First of all, I would like to thank my supervisor dr. Andreas Peter at the University of Twente for providing me with his assistance and support.

Secondly, I am thankful to Gijs Hollestelle and Sander Maas from Deloitte Netherlands for devoting their time to supervise me, provide support and valuable insights.

Thirdly, I would like to acknowledge the robotics company and its employees for providing important information and the robot for the research.

Finally, I am glad that I had an opportunity to study within the Security & Privacy program of the EIT Digital - Master School. This program gave me an exceptional chance to study in two foreign countries. It improved my professional and social skills, helped to meet interesting people and make friends for life.

Abstract

Currently there is a significant increase in the popularity of robots. This statement holds true for both consumer and professional robots. The market is shifting towards automation and optimization. And robotics is one of the main tools which is leveraged for these purposes. However security still remains a weak point for robots. One of the reasons for it is an absence of security assessment documentation for robots. In this research we investigate what components constitute a robot and can influences its security. We obtain this information from a literature review, expert interviews and our investigation of a professional robot. Based on the obtained information we design the first security assessment framework for robots. Additionally we provide information on how to use the framework. In order to identify whether our framework provides value for security professionals we perform an experimental validation. We validate whether our framework helps to ensure that all the components that can influence robots security are assessed during security assessments (completeness). We do it by conducting an experiment which involves security professionals and a professional robot. Additionally we validate whether two independently working professionals can achieve same results with our framework even when they work independently (reproducibility). We do it by providing reasoning why it is true, making an assessment with the help of the framework ourselves and showing how the results of an assessment look like.

During our assessment of a professional robot we identify importance of securing internal networks of robots even though it was not mentioned in the literature before. We identify that communications on internal networks most often have no authentication and encryption. Consequently all communications between nodes can be disrupted or modified. Attackers can issue rogue commands and therefore impact is high. However physical access to the robot is needed to launch the attack which lowers its probability.

We identify that emerging security areas lack security assessment documentation. As a result ad hoc practices are used and it can influence the quality of security assessments. We tackle this problem on an example of robots by creating a security testing framework. Consequently it can be the first step to improve security in the robots industry by ensuring completeness and reproducibility of security assessments of robots.

Table of contents

1. Introduction

Robotics is not a new technology. However nowadays its growth is expanding and it is going to play a crucial role in the future of our society. It provides unique opportunities in multiple domains and the number of uses for robots is expanding. Robots allow organizations and individuals to lower the costs and make execution of certain tasks faster, more precise and more reliable. They are widely used in an industrial production, military combat and defense, medical care, physical security, home applications and toys. Some of them even started to gradually replace humans in different activities [1].

The number of robots is rapidly increasing. International federation of robotics expects 42 million of service robots to be sold during 2016-2019 [2]. And service robots is only one branch of robotics. So soon robots are expected to be in different aspects of our lives. But as in any other blooming innovation driven industry security is not the priority, it is way behind the technology and only an afterthought. Similar story happened with internet of things. As it can be seen the consequences were devastating because no action to secure the things was taken until hackers were able to easily gather a botnet capable of delivering a record 620 Gbps DDoS attack [3]. Obviously at that point in time it was already too late to react and try to fix things. Having millions of unpatchable vulnerable devices in the wild made the problem very hard to solve and proved the importance of security by design. If no action is taken same thing might happen with the robotics industry. However taking into account that robots are cyber-physical systems, threat landscape will be a lot wider.

1.1 Motivation

Recent incidents involving robots gained high media attention. An industrial robot which crashed a worker to death at a car plant [4] and a security robot that ran over a toddler at shopping mall [5] raised the safety question of using robots in our daily lives again. However in the case of such robots being hacked, the results can be worse and the incidents more frequent. Dallas police department used a robot to deliver a bomb and kill a suspect [6]. It raised the concern about applying lethal force through robots. However one of the most dangerous aspects is the security of such robots. They have capabilities to cause multiple injuries, deaths and destruction if they are hacked or tampered. The lethal force might be applied at the wrong place and to the wrong people. For instance Dallas police robot could have been directed into the crowd of people instead of the dangerous suspect and then commanded to explode the bomb.

1.2 Problem

Despite all the risks and possible severe consequences, little research has been done in the field of robots security. And even taking into consideration existing research, the main problem is:

- An absence of documentation that can be used as a guidance for creating secure robots and assessing the security of the existing ones.

Lack of publicly available and standardized security assessment documentation leads to ad hoc practices being used. While those practices allow to successfully perform tests most of the times, lack of documentation causes non-comparability, inconsistency and sometimes incompleteness of results.

1.3 Approaching the problem

For these reasons, we aim at creating a security framework for robots to address the aforementioned issues.

We tackle this problem for the domain of robots security by creating a framework which can be used to perform robots security assessments. It will allow to achieve more repeatable and consistent results by providing a list of components that must be assessed during the engagement and how each of them can be assessed. In order to verify that framework is able to do its job we perform its experimental validation afterwards.

1.4 Research contributions

As a result of this work the following contributions are made towards solving the problem of an absence of security documentation for robots.

1. We create the first framework for security assessments of robots. Which is an important first step to improve the state of security in the robots industry.
2. We perform a scientific validation of our framework in a form of an experiment which involved security professionals and a professional robot. Additionally we use our framework on a professional robot. In such a way we prove that our framework helps to ensure completeness and reproducibility of robots security assessments.
3. By providing such a framework for robots we tackle the problem of ad hoc practices in emerging security areas. And our framework can be used as a tool to improve this situation in the field of robots security.
4. We perform an extensive analysis of a real robot which results in a new attack on the internal network of the robot. Consequently it helped to improve the framework by adding "internal network" to the framework as a component which can influence security of the robot even though it was not mentioned in the literature before. Moreover this finding is of independent interest itself. Because literature mentions that robots have internal networks for wiring together internal components (nodes), but it misses the fact that an internal network is a security critical element which can influence robots security.

1.5 Organization of the thesis

Rest of the thesis is organized in the following manner. Chapter 2 provides background information on robots, used technologies and possible risks that they can pose. In chapter 3 state of the art overview of the field of robots security is given. However very little research has been done in this area, which further emphasizes the importance of this work. Chapter 4 gives an overview of security frameworks in an IoT domain (due to an absence of any in the domain of robotics and multiple similarities between two of them) and links it to the relevance of our research. Chapter 5 gives information about the design methodology for the framework and what steps were taken in order to build it. Chapter 6 provides the framework structure, information on different security levels and the workflow for using the framework. Chapter 7 gives overview of the experimental validation, scientific claims and their validation. Finally thesis closes with a conclusion, discussion and recommendations for the future work.

2. Background on robots

Robotics is quite a unique field. The whole concept arrived from the science fiction. And there is no universally accepted definition of what exactly constitutes a "robot" or what robots types exist. For this study, a robot is a cyber-physical system with sensors and a degree of mobility.

A robot is a machine - especially one programmable by a computer - capable of carrying out a complex series of actions automatically [7]. Robots can be also guided by humans with an external control device or the control may be embedded within. Robots may be constructed to take on a human form but most robots are machines designed to perform a task with no regard to how they look.

2.1 Operating System

Modern robots use different operating systems. Not long ago operating systems and software used for robots were closed source, developed by each company individually and accessible only within the given company. However nowadays the situation is changed with the development of widely accessible robots specific operating systems like open source Robotic Operating System (ROS). It lowered the entrance barrier into robotics for individuals and small companies. Sometimes non robot specific operating systems are adapted (e.g. Raspbian or any other Linux distribution) with some additional custom software designed to implement the required robots functionality. The fact that robots have full operating systems running means that they can be vulnerable to the same type of attacks the computer systems are vulnerable nowadays. However robots security is an even more complicated issue because of the presence of some unique advanced capabilities, like freedom of movement, physical actuators,

multiple sensors, cameras and microphones. Another feature is different modes of operation (autonomous, teleoperated) and presence of different communication channels ranging from the internet to XBee. All of it combined makes robot security way more complicated topic than the regular IT security. The attack surface is huge and as with all cyber-physical systems the impact might be very severe and even life threatening, especially in critical applications.

2.2 Communication channels

Most robots implementations require some kind of a communication channel in order to retrieve data, transmit a video stream, control and configuration packets. Depending on the robot type the following communication capabilities might be present. Robots that are located in static places can use wired communication channels like Ethernet, serial or USB. However even non-stationary robots might have wired ports for diagnostic, configuration or programming purposes. Robots that require some freedom of movement use different wireless protocols depending on their range of operation. Wi-Fi networks are used in 2 different ways. Robot can set up an access point that an operator will connect to. Alternatively it can connect to an existing W-FI network in order to extend its range to the coverage of the existing Wi-Fi network and allow easy interaction with an existing infrastructure. A radio communication and its variations like Zigbee can be also used in robots. Low range home robots or toys can use Bluetooth or infrared communication links. While long range robots used in military and critical applications use cellular or satellite networks. Some teleoperated robots that might operate via long distances can use the internet as a communication channel (e.g. surgical robots). If required by the current application some robots have redundant connections (e.g. s combination of Wi-Fi and LTE) and can switch between communication links fast if one becomes unavailable.

2.3 Sensors

Robots can be either stationary mounted or being able to move on their own. Robots capable of moving need a way to orient themselves in the space. For this reason they might have cameras, GPS receivers, proximity sensors and motion detection sensors.

Based on the functionality of the robot microphones and speakers can be present (e.g. for VOIP communications). Some robots also have physical actuators which can be abused by malicious actors in order to deal physical damage, move physical objects or even disable some hardware components on the robot itself.

2.4 Threat landscape

Different threat actors can take advantage of the described above capabilities present in robots. Ranging from terrorists whose goal might be to cause some heavy physical damage to nation sponsored attackers who might eavesdrop or confidential information theft with the help of exploited robots. A wide range of technical capabilities pose unique risks for robots. Communication protocols can be attacked in order to sniff data, inject bogus control packets or perform a DOS attack on the control link. Audio and video recording options combined with the freedom of movement transform most robots into universal, self-moving spying devices and it poses a serious privacy risk for commercial and household users. Moving capabilities mean that attackers can force the robot to leave a secure controlled area and then perform physical attacks or theft. It is a widely accepted notion that if an attacker has physical access to the computer nothing can stop him from gaining access. Robots in publicly accessible areas can also be compromised by hackers who tamper with their software or hardware.

3. Related work

The following chapter provides literature overview on the topics of robots security and security frameworks.

3.1 State of (in)-security

As it was stated before very limited research was conducted in the field of robots security. This section provides existing state of the art overview of the field.

3.1.1 Robot hacks

In their work Bonaci et al. [8] analyzed vulnerabilities in the Raven II Surgical Robot. Raven ii is a teleoperated robotic system designed to support research in advanced techniques of robot-assisted surgery. It uses open standards software including Linux and Robot Operating System. It is a remotely controlled robot. Operators can be nearby or at a completely separate location.

It was found out that there was no authentication and encryption in the communication link. So authors were able to successfully perform man-in-the-middle attacks and consequently execute the following intent modification attacks. Surgeon's Intent Reordering, a zero knowledge attack based on random reordering of intent packets going through a telemetry link from an operator to the robot. Surgeon's Intent Loss, another zero knowledge attack which is based on random intent packet drop. Surgeon's Intent Delay, a zero knowledge attack based on delaying legitimate packets for an arbitrary amount of time. Surgeon's Intent Modification, an attack based on

intercepting a legitimate packet addressed to robot, modifying the intent and sending it to the robot afterwards. Bonaci et al. were also able to perform hijacking attack. With no authentication in place the only required attribute in order to take the control of the robot was the sequence number of the packet. After the moderate time of eavesdropping on the network they were able to find out the current packet sequence number and take full control of the robot by sending any desired command. Obviously all of the performed attacks are unacceptable for a surgical robot and could lead to horrible consequences during the real surgery.

Denning et al. [9] investigated the security of 3 consumer level household robots. Multiple vulnerabilities were discovered. For instance, all communications in some robots were unencrypted and consequently leaked robots authentication credentials and recorded audio/video stream to everyone on the same wireless network. They were also able to control one of the robots with a separately bought off-the-shelf remote control. Authors also expressed a concern regarding robots which have extensive sensing capabilities (audio and video) and the privacy risk they are creating for the environments they are used in. The other concern was regarding robots mobility and ability to grasp objects with actuators and move them or just push objects around in order to deal physical damage.

In their report Cerrudo and Apa [10] analyzed different home, business and industrial robots from multiple vendors including SoftBank Robotics, UBTECH Robotics, ROBOTIS, Universal Robots, Rethink Robotics, and Asratec Corp. Authors identified about 50 vulnerabilities in the investigated robots and pointed out the following main problems:

- insecure Communications
- authentication Issues
- missing Authorization
- weak Cryptography
- privacy Issues
- weak Default Configuration
- vulnerable Open Source Robot Frameworks and Libraries

While 50 vulnerabilities is a lot, it is worth mentioning that authors stated themselves that their investigation was not even a deep and extensive security audit. So it is quite likely that there are many more undiscovered problems and vulnerabilities in analyzed robots.

Maggi et al. [11] focused their report on analyzing security of industrial robots. They found multiple vulnerabilities in a typical industrial robot (ABB IRB140). By leveraging identified vulnerabilities in:

- unsecured network and command injection,
- weak authentication,
- naïve cryptography,
- memory corruption,
- missing code signing,
- poor runtime isolation.

They managed to create 5 different remote attack vectors:

- production outcome alteration or sabotage,
- ransomware attacks on altered products,
- physical damage,
- production line process interference,
- sensitive data exfiltration.

3.1.2 Vulnerabilities in components

Morante et al. [12] investigated the security posture of two the most widely used operating systems in robots: ROS (Robot Operating System) and YARP (Yet Another Robot Platform).

Widely adopted practice in robotics is a component based software engineering. Every single component (e.g. camera recognition program) is designed as an individual piece of software which communicates with other components via predefined protocols. Both ROS and YARP operating systems function in a similar manner in order to support a component based software engineering. Such components (nodes or modules) can be located on one or multiple hosts and connected into a peer-to-peer network.

Authors mentioned that there is no authentication between nodes in ROS. Consequently such anonymous ability to read/write to nodes is a welcome area for exploitation. Another problem is the absence of any encryption while nodes communicate via TCP/IP or UDP/IP, which consequently allows attackers to read these communications.

New connection to YARP is established via a TCP handshake to the specified port. So any attacker on the same network can easily get access and abuse it. However YARP has an option for activating an authenticating mechanism, which adds a key exchange to the initial handshaking.

As it was mentioned earlier ROS has no authentication and encryption between nodes. Consequently the following attack vectors on ROS are identified by Dieber et al. [13] and McClean et al. [14]. An attacker can perform data injection by unauthorized publishing of malicious commands or false data to the node. Eavesdropping can be performed as well by unauthorized subscribing to the node. It can result in gaining intelligence on the way production process is organized or video and audio stream intercepting.

With the specifics of the ROS functioning attackers can easily perform DOS attacks by high frequency publishing to some node. The other way of performing a DOS attack is creating a new evil node with the name of the existing legitimate one. Because of the way ROS functions it will force the legitimate node to shut down even if it is located on the other host.

The TNO report [15] provides a diverse overview of risks to human safety that can be caused by robots. The most relevant part to the current research is the one about cyber-physical security. Report identifies the following possible cyber risk factors that can lead to inappropriate behavior of robot and consequently cause physical harm. Inaccurate sensor information. Robots need multiple sensors for their situational awareness, however some of these sensors might provide non-realistic information as the

result of deliberate manipulation, technical malfunction or human error (configuration). It can cause robot to misbehave appropriately.

Robots use different communication channels to communicate with their own sensors, control center and between each other. From here the following risks arise. This communication links can be abused to disrupt communication (submit false sensor values, deliver illegal commands from a fake control center, deliver wrong information to other robots during swarm operations) or completely block the link (frequency jamming, dos, overloading the channel).

Robots software can be manipulated by malware that can be delivered during the software/firmware update, reprogramming routine or via portable media. As long as there is a control center, it poses risks to robot operations. If the control center is compromised attackers can send incorrect instructions to robot or manipulate it in any way they want. So the control system security must be also taken into consideration.

3.2 Frameworks

One of the problems that security industry is facing now is a lack of standardized ways for performing security assessments in new and emerging areas.

Robots security is a relatively new field and therefore no security frameworks exist in this domain yet. For this reason we decide to focus our research on the closely related field of IoT security due to multiple similarities between two of them:

- cyber physical interaction,
- connectivity capabilities,
- possibility of being controlled remotely,
- variety of use cases ranging from households to critical infrastructure.

Below is an overview of IoT security frameworks that can be found in an open literature.

Rahman et al. [16] define security requirements for IoT devices divided into the following 4 layers: Things layer, Communication layer, Infrastructure later and data analytics layer. Based on the identified requirements authors propose a security framework. They managed to identify components that must be secured and provide very short description on how it can be done. However descriptions are very short and might give an idea of what direction can be taken, but not the steps that should be followed. This framework does not suit our needs for the following reasons. It is focused on sensor to cloud ecosystem and therefore has specific components for its use case. While our framework is focused on a wider domain of robots in general. It defines only generalized security requirements but does not mention how they can be assessed. While we aim at creating a framework that can be used for security assessments.

Leister et al. [17] suggest a framework for adaptive IoT security in eHealth applications based on scenarios and stories. Basically they propose to develop different scenarios for different use cases and then define security requirements for each of them. This approach can be suitable for creating secure devices by design. However a created set of security requirements is tailored to the specific scenario or story. And therefore it must be created from scratch for every new scenario. This approach is not suitable for tackling our problem because our framework should be universal for different robots applications and ready to use without any adjustments.

Online trust Alliance published their IoT Security & Privacy Trust Framework [18]. The framework outlines strategic principles necessary to secure IoT devices and their data. The framework is composed of four layers: security principles, user access and credentials, Privacy, Disclosures & Transparency, Notifications & Related Best Practices. Each of the layers has a list of principles which are included in this particular layer. Each principle has an indicator whether it is required or recommended to implement. The framework provides security principles that should be implemented while designing a new product in an area of connected home, office or wearable IoT. Therefore it has same limitations for the current research as [16].

Babar et al. [19] suggest a security framework for embedded IoT devices. Framework lists security principles that should be implemented in IoT devices. The following principles are mentioned: lightweight cryptography, physical security, standardized security protocols, secure operating systems, future application areas, secure storage. Each of the principles has a short description.

However this work is not a literature survey and it does not mention components that should be secured. While names of the identified security principles seems similar to our context, there is a difference in a content. Because [19] is heavily focussed on protecting embedded devices against hardware attacks. While robots have embedded devices on their internal networks, we decide to put hardware attacks on them out of scope. Because properly securing all of the internal components against hardware attacks will require high financial and time investments. At the same time it will provide little value because if security of the robot is implemented properly attacker will not be able to reach those components. Moreover it is simply not possible to apply embedded devices security principles to robots.

3.2.1 Conclusion

Analyzed papers have first steps, possible design solutions for creating an IoT security framework or include a list of components that should be taken into account for IoT devices security. However we did not manage to find an existing IoT security framework that can be used as guidance to perform complete security assessments. Therefore creating a security assessment framework is still a relevant issue for IoT devices. And due to a lack of research in the field of robots security it is even more relevant for robots.

4. Our design

As it was shown in the previous sections robots represent complex systems with multiple components and constant information exchange between these components. It makes evaluating security and consequently creating a framework designed to evaluate security of the robot as the whole very complicated task. It poses a challenge for the framework to cover all of the required components, but remain easy to understand and intuitive at the same time.

The goal of the framework is to assist during security assessments for robots by providing components that must be assessed, guidance on how it can be done and assisting in an overall evaluation for security of the robot based on assessed components.

4.1 Components that must be secured

The first step in creation of the framework is identifying all the components that might influence security of the robot and therefore must be included in the framework. In order to identify them we perform a literature review, conduct expert interviews with robot and security professionals. We also study a real robot for this purpose.

4.1.1 Literature review

We start with a literature review on known attacks and vulnerabilities in robots. Then we extract robots components that are needed for attacks to succeed. Extracted components can be found in appendix A.1. However as we mentioned before very little

research has been done in the field of robots security. Therefore the literature review provides us with limited information.

4.1.2 Expert interviews

In order to ensure that we identify all the important components which can constitute a robot and influence its security additionally we perform interviews with a software developer, robot engineer, chief technology officer (CTO), security and communications engineer at a robots company which name cannot be disclosed due to a non-disclosure agreement (NDA). The following topics are addressed during the interview:

- components used in robots and their interconnections,
- possible constraints while securing robots,
- what components the company is securing in their robots.

As the result we extract additional components that can influence robots security. Those components can be found in appendix A.2.

Afterwards we perform an interview with two senior security professionals at a large international security testing enterprise in order to verify our findings and attempt to identify any missing components that should be taken into account while securing robots. As the result we get additional components which can be found in appendix A.3.

In order to ensure the quality of the framework during our work we continuously consult security professionals on the following topics:

- feedback about the document structure and its usability,
- feedback on contents:
  - included components,
  - proposed steps for assessment of each of the components.

4.1.3 Analysing the security of a professional robot

Sanitized

4.2 Framework structure

After the first step we end up with an extensive list of robots components which can influence its security. All those components are listed in appendix A. The next step is deciding on the structure of the framework. This section provides insight on how we make design decision on the framework structure.

4.2.1 Division by layers

We face the problem that the list of components is so big and diverse that it can easily influence ease of understanding and make the framework less intuitive.

However after studying internal documentation at the large international security testing enterprise we identify that relatively complex assessments are always divided into separate stages and each stage has related activities grouped together. Therefore we decided to organize components by some criteria. Taking into account wide range of components that we identified, there are no existing solutions that can accommodate such complex systems.

After interviewing with senior security professionals we decide to divide robots security in our framework based on their nature into four existing domains.

- Physical security - robots are cyber physical systems and some of them operate in publicly accessible areas. Therefore their physical security should be taken into consideration.
- Network security - as it was stated in section 2 robots use a variety of communication channels in order to interact with the control center, other robots and an existing infrastructure. Different types of connection channels (mostly USB and Ethernet) are also used to wire together different internal components of

the robot. Computer security industry has a long history of vulnerabilities and possible attacks on different communication protocols. And therefore network security is the next layer that we include in the framework.

- Operating system security - as we mentioned before robots specific OSs are usually built on top of Linux and rarely Windows. So the security of the underlying OS should be taken into consideration in order to prevent vulnerabilities.
- Application security is important due to the presence of specific software designed to fulfil a robot's application. And it includes robots software designed to implement a robots logic and operations, control center software which might be present in the form of web or mobile device application.

We divide all the identified components by these four layers. This design decision is similar to Rahman et al. [16], Online trust Alliance [18] and Babar et al. [19]. However we use different layers due to differences identified in section 3.2. Consequently we create a structure where components are grouped based on their nature into layers and each of the layers can be approached individually. So the whole system is easier to understand and to work with.

4.2.2 Framework elements

At this stage we already have a list of components that constitute a robot divided into 4 layers. The next step is deciding on which framework structure is the most suitable taking into account its goal. The goal of the framework is to assist in performing robots security assessments. Having this in mind we decide to perform a state of the art analysis in the domain of frameworks. There are no existing frameworks in the domain of robots and therefore we decide to analyse the closest field to robotics - IoT. Analysis of IoT frameworks is present in the previous chapter. However as we identified there no suitable solutions that can be applied in our situation. Therefore we decide to refer to the large international security testing enterprise internal documentation again in order to study the way security assessment guidances are designed and structured.

We identify that all the documents are structured in the form of a table. Almost all documents are divided into components that must be assessed. Each of the components has an objective for its assessment and a list of steps that can be used to assess it. Additionally some of the documents contain notes on some of the components, which

are intended to provide additional information when needed. We also find that some of the documents include information on why the component should be secured. However it is not specified in a specific section but rather indirectly mentioned in any other section (usually an objective). Few very targeted documents also mention tools that can be used during the assessment of each component. All the documents have a designated part where observations and evaluations should be recorded for each component.

Studied internal documents went through many iterations until they reached their final form and now they are used every day in order to perform different kinds of security assessments. Therefore we can rely on them as a valuable source of information on the way assessment documentation should be created. Based on the fact that the goal of our framework is assisting in performing security assessments we decide to follow most of the identified observations in those documents and implement them in our framework. Specifically we decide to make our framework in the form of the table. We decide to include the following subsections in our security framework:

1. objective for evaluating of each of the components,
2. guidance on how it can be done,
3. notes for assessment phase if necessary,
4. risks if particular component is unsecured,
5. observation and evaluation for each component.

However what we make differently is we include a designated subsection on why each of the components should be secured in order to make it more organized and easier to locate if needed. Another difference lies in our decision to group components based on their nature into four layers as we mentioned before. It helps to systemize an extensive list of components that must be secured.

4.2.3 Security levels

At his point we have the framework design and all the components that we need grouped into 4 different layers. However the next challenge is deciding on how security of a robot should be evaluated and what metrics should be used. Based on an extensive

list of identified components, the cost to secure all of them can be quite high. However different levels of security are required for different robots applications. So it is important to find a way of differentiating between them. And therefore we decide to provide different levels of security based on the robots application and maturity of the client. We decide to suggest security levels for the following three situations:

1. A security budget is extremely limited or companies are absolutely unwilling to spend extra money on security investments. In such a case we need to provide a guidance on how to mitigate the most critical threats with the lowest investment possible. It should help to mitigate such large scale and easy to deploy attacks as ones that happened in autumn 2016, when by leveraging default or hardcoded credentials attackers were able to easily take control of hundreds of thousands of IoT devices in order to deliver massive DDoS attacks[3]. Due to mitigating only the most critical and easy to deploy attacks we call it **Trivial Defense**.

2. Typical consumer level devices that cannot pose significant risks and have relatively low prices, which limits financial resources to secure them. On this level we attempt to provide a list of components that must be secured in order to build a secure perimeter in robots defense. Consequently it allows to achieve reasonable security with relatively low investment by securing only components which are part of the perimeter. We call this security level **Perimeter Defense**.

3. Devices created for usage in critical environments and/or when human safety can be at risk. Therefore it makes higher spending on security justifiable. At this level we attempt to build multilevel defenses against attackers by providing all the critical components that can influence robots security and securing them. We call this this security level **Defense in Depth**.

5. Our security framework for robots

Following the steps outlined in the design section we create a framework for robots security. This chapter provides information on its structure, security levels and how it can be used. The framework itself can be found in appendix C.

5.1 Framework structure

Robots are complex systems and in order to make their assessment easier and more intuitive security is approached on four different layers: physical, network, operating system and application. In the framework each layer is represented by a separate table and consists of components which must be assessed on this layer (left column). Each component has one or more evaluation criteria (middle column).

"Evaluation criteria" consists of "objective" for this small evaluation, "how to" steps in order to perform evaluation, "why" describes possible impact if this criteria is not satisfied. Each evaluation criteria also has "evaluation" bullet, which describes what security level requires it.

"Results" column which is located at the right can be used to write down observations and results for each evaluation criteria.

5.2 Levels of security

The goal of the framework is to assess the security level of robots and identify what can be improved in order to increase security. According to the framework robots can

have 4 different security states. No security at all, trivial defense, medium security (perimeter defense) and high security (defense in depth).

Each security level has a list of evaluation criteria. All of which must be satisfied in order to conclude that robot has that particular security level. Even if one evaluation criteria is not satisfied, the security level cannot be granted. Each evaluation criteria in the framework has an evaluation section, where security levels that require this criteria are mentioned.

No security – when a robot does not meet one or more requirements for the trivial defense level, it is assigned no security label. Robots with no security pose significant danger and can lead to mass scale disasters due to their ease of compromise. It is highly recommended to avoid using such devices.

Trivial defense (TD) – a set of security controls designed to prevent mass scale hacks by mitigating the most commonly used and easily exploitable attack scenarios. These measures should be implemented in any robot. Robots without these security features are considered insecure and can be trivial to compromise. This security level is suitable for situations when security budget is extremely limited or companies are absolutely unwilling to spend extra money on security investments. In such a case at least the most critical threats should be mitigated with the lowest investment possible. Robots at this security level are still vulnerable to less common and more complicated attack scenarios.

Perimeter defense (PD) - provides an improvement over a trivial defense level by securing all the possible entry points for attacks. However we rank it as medium security, because it mostly relies on a single level protection. If one secured component fails, an attack can be successful. Therefore it can protect from known attacks, but will fail against unknown ones (e.g. when new vulnerability is found in the component that was believed to be secure). We believe that it is most suitable for consumer grade robots that cannot pose significant danger and have relatively low prices, which limits financial resources to secure them. It is used to cut off all attack entry points in order to mitigate the biggest percentage of attacks with the lowest investment.

Defense in depth (DID) - on this level, the goal is to achieve the highest level of security by securing as many critical components as possible and consequently building a multilevel defense. This approach requires the most financial resources. However it also provides the highest level of security. We believe that by securing all the critical

components and consequently achieving defense in depth it is possible to mitigate not only currently known attacks. Even if attackers find a way to circumvent some security mechanisms there are still going to be additional obstacles which can prevent an attack from succeeding. Thus it is applicable for critical application, when security should be a priority.

5.3 How to use the framework

An assessment can be done in two ways. In the first case all the evaluation criteria are evaluated and assessed and then results are reviewed in order to identify which security level can be assigned to the robot. It is done by reviewing which security levels meet all the evaluation criteria. If multiple levels are identified the highest one should be chosen. In the second case desired security level is identified in advance and only evaluation criteria respective for this security level are assessed. The first method gives a more complete picture while the second one requires less time.

Assessment for each evaluation criteria can be done with the help of the respective "how to" section. Whenever applicable there is also a "note" section that can specify some additional helpful information.

Once the assessment is finished and robot security level is found, framework can be used to implement required security mechanisms in order to go one level up by fixing all the failed evaluation criteria for the respective level.

6. Experimental validation

During the interview with security professionals we identify that they value two main qualities which security assessment documentation should help to achieve:

- completeness of the assessment,
- reproducibility of the assessment.

Therefore in order to validate the framework and prove or disprove its value for the real world we perform an experimental validation for both of them.

6.1 Completeness

The term "completeness of the assessment" means that all the components which can influence robots security must be assessed.

6.1.1 Scientific claim

Our scientific claim is that our framework can help to ensure completeness of robots security assessments by providing a pool of evaluation criteria that must be assessed and giving a guidance on how to approach each of them.

6.1.2 Experiment setting

A professional robot for the experiment was kindly provided by the robots company. Additionally a robot engineer from the same company was invited to manage the robot during the test and answer possible technical questions about the design. Additionally we prepare a participant template which included the following sections.

- Participants background:
  - years of experience in information security,
  - years of experience in security testing,
  - areas of expertise.
- Components that were tested.
- Components that should have been tested. But it was not possible due to time constraints.
- Observations about tested components.

We assume that if security professionals use our framework they will achieve consistent results by assessing all the identified components. For this reason all the participants perform assessment without the framework. Afterwards we compare components that subjects assess without the framework to the components that we included in the framework. It allows us to draw the following conclusions.

1. How consistent are results between participants?
2. Can participants identify all the components that are needed to be assessed during the time of the experiment?
3. Do participants identify any components that were not mentioned in the framework?

6.1.3 Results

Experiment started with the short introduction of our research, the used robot, its application and design. Afterwards we informed participants on the experiment setting

and how to use the participant's template. Then we handed out the templates and started the experiment.

Table 6.1 shows information about participants and their background.

Table 6.1 Information on participants

	Participant 1	Participant 2
Years of experience in information security	4	15
Years of experience in security testing	4	12
Area(s) of expertise	Infrastructure, web, incident response.	Infrastructure, web, code review.

The experiment lasted for two hours. During the experiment participants had occasional technical questions which were answered by the robot engineer. Below are the results in a raw form.

Sanitized

6.1.4 Evaluation of results

Participants mainly focussed on the internal network. It can be explained by their proficiency in the infrastructure security, which involves network security. Participants tested an internal network. However it is safe to assume that they would have tested an external network too. But it was not possible due to complications in setting the robot at an experiment location.

However participant 2 tested a possible attack on an external network by reviewing robot configuration settings. The robot is exchanging all the data with the cloud server via a VPN tunnel and the goal of the attack was to trick it into sending data to an attacker on a local network instead. However it was not possible due to firewall settings which were allowing outgoing connections only over the VPN tunnel. Due to such a result we added a firewall evaluation to our framework.

Another interesting result is that participant 1 wrote that he wanted to test a maintenance USB port. While participant 2 went further and mentioned hardware

testing for different hidden management interfaces. Such interfaces can be present on multiple nodes of the robot.

But overall we can see that lack of documentation caused participants to focus on their areas of expertise. A time constraint influenced the amount of identified components that should be tested. And a need to identify components to test also took additional time and therefore reduced efficiency. While 2 participants focussed on the network layer only participant 2 tested external network and participant 1 had more observations for the internal network.

Our framework contains 12 components that must be tested and participants tested 2 components (internal and external network) and mentioned another one (hardware communication ports) which they wanted to test.

While our experiment is very small in terms of people and time during which it was conducted, it is clear that having our framework could have helped participants to achieve more complete results during the limited time of the experiment. And therefore based on the experiment we can conclude that our framework can help to achieve more complete results during the assessment.

Encountered problems

In order to prove our claim originally we were planing to divide subjects into two groups. Then provide subject from one group with our framework, while subjects from the other one was supposed to rely on their knowledge and experience. Afterwards participants from both groups were supposed to perform a security assessment of a professional robot. Then we were going to draw the conclusion from comparison of the results from two groups.

However due to a low number of participants (3) in an experimental validation, it was not possible to have two groups with enough subjects in each of them. Therefore we had to change our experiment setting.

Another problem is a lack of time during which we were able to run the experiment. For this reason in the participant's template we add a section where participants can include components that they would like to asses, but were not able due to time constraints.

Due to aforementioned issues an experiment was very small in size and it would be a good idea to redo it with larger amount participants and for longer period of time.

6.2 Reproducibility

The term "reproducibility of the assessment" means that different security professionals assessing the same robot should achieve same results even when working independently.

6.2.1 Scientific claim

Our second scientific claim is that our framework can help to ensure reproducibility of security assessments. Which means that two independently working professionals will come up with the same results if they use our security framework.

6.2.2 Setting

Our framework provides a pool of evaluation criteria which must be assessed and a guidance on how it can be done. Due to such a template form it is clear that results will be the same. Therefore we perform a security assessment of the professional robot with the help of our framework and provide results of our evaluation to demonstrate possible outcome of using the framework. During our analysis of the professional robot we already assessed all the required components and their evaluation criteria following the steps that are mentioned in the framework. For this reason during this experiment we go through the framework on paper and fill in all the observations.

6.2.3 Results

The fact that we are already familiar with the robot and tested all the required evaluation criteria beforehand helps us to fill the assessment in under 30 minutes. The table **??** shows our observations and evaluation for each of the evaluation criteria.

6.2.4 Evaluation of results

Sanitized

7. Conclusion and future work

Robotics industry is rapidly expanding. However robots security still remains a weak point. One of the reasons for it an absence of documentation to asses robots security. We found that this problem is present in other emerging security areas as well. Consequently it leads to ad hoc practices being used, which can influence the quality of security assessments. We discovered that even robots that are built secure by design can have security flaws. During this master thesis we created the first security framework for robots. We experimentally validated our framework and proved hypothesis that it can help to perform more complete and reproducible security assessments of robots. Additionally we discovered importance of securing robots internal networks by finding possible attacks on it.

7.1 Discussion

During this thesis we developed the first security framework for robots. Robots security domain is in an inception phase now. And the developed framework can be used to improve robots security by improving the quality of security assessments of robots.

The framework eliminates the need to identify all the components that must be assessed during security assessments by providing a list of components and evaluation criteria for each of them. Consequently it helps to ensure that all the required components are assessed, which contributes to the completeness of the assessment. Additionally it allows security professionals to focus on the assessment itself which can improve their efficiency. A template form of the framework helps to ensure reproducibility of the assessment. Which means that two independently working

professionals can achieve same results even when working independently. Consequently it makes results more stable and comparable.

We believe that there is no security framework which is absolutely complete. And it was proven during the experiment on completeness when we discovered one missing component in our framework. Therefore our framework needs steady re-iterations before it can reach its final form. While it looks obvious that template form of the framework should help with reproducibility of assessments, we did not have an opportunity to validate it with multiple subjects. Another limitation is a low number of subjects in an experiment on completeness.

We took three sources of input for our framework: a literature review, expert interviews and our analysis of a professional robot. While it was possible to focus on one of the sources during the study, we thought that their combination should bring the most interesting and complete results.

7.2 Future work

There is still work that can be done to improve our security framework. Firstly, additional components and evaluation criteria can be added. Secondly, experimental validation of completeness of the assessment with a bigger number of participants and on a longer time span should be done.

There is still a need for more security research on robots. And our framework can be applied to several models of robots to identify their current state of security.

Attacks on internal networks of robots were not deeply researched due to time constraints. Because it involved reverse engineering of a custom written protocol for the internal network of an analysed robot. However due to performance and reliability requirements internal communications are usually unencrypted and unauthenticated. And we believe that it provides an avenue for different attacks which can be researched. Another research idea is developing a security mechanism for robots internal networks that takes into account performance and reliability requirements.

References

[1] Where machines could replace humans–and where they can't (yet) | mckinsey & company. http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/where-machines-could-replace-humans-and-where-they-cant-yet. (Accessed on 03/21/2017).

[2] Executive summary world robotics 2016 service robots | international federation of robotics. https://ifr.org/fileadmin/user_upload/downloads/World_Robotics/2016/Executive_Summary_Service_Robots_2016.pdf. (Accessed on 03/21/2017).

[3] Ddos on dyn impacts twitter, spotify, reddit — krebs on security. https://krebsonsecurity.com/2016/10/ddos-on-dyn-impacts-twitter-spotify-reddit/. (Accessed on 03/21/2017).

[4] Robot kills man at volkswagen plant in germany - telegraph. http://www.telegraph.co.uk/news/worldnews/europe/germany/11712513/Robot-kills-man-at-Volkswagen-plant-in-Germany.html. (Accessed on 03/21/2017).

[5] Stanford shopping center security robot runs over toddler harwin cheng | daily mail online. http://www.dailymail.co.uk/news/article-3686411/He-crying-like-crazy-300lb-mall-security-robot-used-catch-shoplifters-runs-one-year-old-boy.html. (Accessed on 03/21/2017).

[6] Dallas deployment of robot bomb to kill suspect is "without precedent" | ars technica. https://arstechnica.com/tech-policy/2016/07/is-it-ok-to-send-a-police-robot-to-deliver-a-bomb-to-kill-an-active-shooter/. (Accessed on 03/21/2017).

[7] robot - definition of robot in english | oxford dictionaries. https://en.oxforddictionaries.com/definition/robot. (Accessed on 03/21/2017).

[8] Tamara Bonaci, Jeffrey Herron, Tariq Yusuf, Junjie Yan, Tadayoshi Kohno, and Howard Jay Chizeck. To make a robot secure: An experimental analysis of cyber security threats against teleoperated surgical robots. *arXiv preprint arXiv:1504.04339*, 2015.

[9] Tamara Denning, Cynthia Matuszek, Karl Koscher, Joshua R Smith, and Tadayoshi Kohno. A spotlight on security and privacy risks with future household robots: attacks and lessons. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 105–114. ACM, 2009.

[10] Cesar Cerrudo and Lucas Apa. Hacking robots before skynet. Technical report, IOActive, 2017.

[11] Federico Maggi, Davide Quarta, Marcello Pogliani, Mario Polino, Andrea M Zanchettin, and Stefano Zanero. Rogue robots: Testing the limits of an industrial robot's security. Technical report, Trend Micro, Politecnico di Milano, 2017.

[12] Santiago Morante, Juan G Victores, and Carlos Balaguer. Cryptobotics: Why robots need cyber safety. *Frontiers in Robotics and AI*, 2:23, 2015.

[13] Bernhard Dieber, Severin Kacianka, Stefan Rass, and Peter Schartner. Application-level security for ros-based applications. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4477–4482. IEEE, 2016.

[14] Jarrod McClean, Christopher Stull, Charles Farrar, and David Mascareñas. A preliminary cyber-physical security assessment of the robot operating system (ros). In *SPIE Defense, Security, and Sensing*, pages 874110–874110. International Society for Optics and Photonics, 2013.

[15] Wouter Steijn, Eric Luiijf, and D Beek. Emergent risk to workplace safety as a result of the use of robots in the work place. Technical report, TNO, 2016.

[16] Abdul Fuad Abdul Rahman, Maslina Daud, and Madihah Zulfa Mohamad. Securing sensor to cloud ecosystem using internet of things (iot) security framework. In *Proceedings of the International Conference on Internet of things and Cloud Computing*, page 79. ACM, 2016.

[17] Wolfgang Leister, Mohamed Hamdi, Habtamu Abie, and Stefan Poslad. An evaluation scenario for adaptive security in ehealth. In *Proceedings of Fourth International Conference on Performance, Safety and Robustness in Complex Systems and Applications, Nice, France*, volume 2327, 2014.

[18] Iot security & privacy trust framework v2.5. https://otalliance.org/system/files/files/initiative/documents/iot_trust_framework6-22.pdf. (Accessed on 07/11/2017).

[19] Sachin Babar, Antonietta Stango, Neeli Prasad, Jaydip Sen, and Ramjee Prasad. Proposed embedded security framework for internet of things (iot). In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, pages 1–5. IEEE, 2011.

[20] Kin-Huat Low. *Industrial robotics: programming, simulation and applications*. I-Tech, 2007.

[21] Stephan Göbel, Ruben Jubeh, Simon-Lennert Raesch, and Albert Zündorf. Using the android platform to control robots. *Kassel University Germany.[Online]. Available www. innoc. at/fileadmin/user_upload/_temp_/RiE/Proceedi ngs/65. pdf*, 2011.

[22] Robert Fitch and Ritesh Lal. Experiments with a zigbee wireless communication system for self-reconfiguring modular robots. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1947–1952. IEEE, 2009.

[23] Dirk Schulz, Wolfram Burgard, Dieter Fox, Sebastian Thrun, and Armin B Cremers. Web interfaces for mobile robots in public places. *IEEE Robotics & Automation Magazine*, 7(1):48–56, 2000.

[24] Juan C Yepes, Juan J Yepes, Jose R Martinez, and Vera Z Pérez. Implementation of an android based teleoperation application for controlling a kuka-kr6 robot by using sensor fusion. In *Health Care Exchanges (PAHCE), 2013 Pan American*, pages 1–5. IEEE, 2013.

A. Sources of framework components

The following chapter provides information on where different components that can influence robots security were obtained.

A.1 Literature review

This section gives a list of components that were obtained during the literature review phase.

- Communication channels
  - Ethernet [20]
  - USB [20]
  - WiFi [21]
  - Zigbee [22, 21]
  - Bluetooth [22]
  - Internet [8, 23]
- Privacy [9, 10]
- Applications
  - Authentication [8, 10, 11]
  - Encryption [8–10]
  - Authorization [10]
  - Vulnerable open source frameworks and libraries Cerrudo and Apa [10]

  - Command injection [8, 11]
  - Insecure networks [11]
  - No code signing [11]
  - Attacks on communication protocol due to protocol insecurity [8, 11]
  - replay protection [8]
- control center application
  - mobile app [24]
  - web app [23]

A.2 Robotics experts interviews

This section provides information on which components were extracted during interviews with robotics experts.

- External physical ports
- Internal components of the robot
- External network protection
- Encryption, ports exposure
- Safety sensors
- Password complexity

A.3 Security professionals interviews

This section lists components that were obtained during interviews with security professionals.

- Monitoring and alert

  - Physical body
  - Internal and external network
- hardware ports security
- OS and firmware updates
- update signing
- 4 layer structure and suggested layers
- protocol security
- distinguishing robot acting as an access point and as a client
- feedback on usability
- reviewing

A.4 robot security assessment

The following 2 components were added to the framework after the security assessment of the robot:

- internal network
- integrity check

A.5 Internal security assessment documentation at a company

This section provides components that were added to the framework after we studied internal security assessment documentation at the large international security testing enterprise.

- table structure for the framework

- framework subsections: Objective, how to, why, note
- default passwords
- login lockout

B. Attack on an internal network

Sanitized

C. Our security framework for robots

1. Physical layer

Component	Evaluation criteria	Results
1.1 External ports	**1.1.1 Presence of external communication ports** **Objective** – identify presence of unprotected external ports **How to** • Inspect documentation / consult developers / inspect robot's body and look for accessible ports (e.g. Ethernet, USB) • Open all doors, which are not protected by locks and look for ports inside • Investigate ventilation holes and see if they are wide enough to access internal communication ports **Why?** Unprotected external ports can let attackers in physical proximity to perform a variety of attacks and serve as an entry point for them **Evaluation** **PD** 1.1.1 or 1.1.2 **DID** 1.1.1 and 1.1.2	
	1.1.2 **Security of external communication ports** **Objective** – verify if attackers can sniff or modify any critical data during communication with a docking station or by	

	connecting to the ports. **How to** • Connect to the identified communication ports • Is authentication required to use them (e.g. Network access control for Ethernet) and do accounts meet requirements from section 4.1? • Try communicating with them, attempt fizzing to discover if robot's state can be affected. • If a robot connects to a docking station to transfer some data, try to use sniffers to see how data exchange is being done (verify if some sensitive, configuration or control data is transferred in clear text) **Why?** Same as 1.1.1 **Evaluation** **PD** 1.1.1 or 1.1.2 **DID** 1.1.1 and 1.1.2	
1.2 Internal components	**1.2.1 Availability of internal components from outside** **Objective** – identify internal hardware that is accessible from outside without a need **How to** • Inspect robots body and look for accessible components (e.g. HDD, embedded devices) • Open all doors which are not protected by locks and look for accessible components inside	

	Notes All cables should also remain inside of the robot. Some components require to be partially outside of the body frame (e.g. range finding systems, WI-FI/LTE antennas) in such a case only the required part should stick out, but not the whole component. **Why?** Directly accessible internal components can be physically damaged, stolen, tampered or completely disabled **Evaluation** **PD/DID –** no internal components should be accessible from outside	
	1.2.2 Monitoring and alert capabilities **Objective** – identify whether rogue access to the internal hardware of the robot can be detected. **How to** • Identify all parts of the frame that can be opened or removed to get access to the internal components • Check whether there is an active (tamper switches) or passive (tamper evident screws and seals) monitoring capability present • In case of active monitoring capability, verify that operator receives a real-time alert and the incident is being logged and acted upon by reviewing procedures **Notes** Passive monitoring provides information upon inspection	

	whether internals were accessed or not. However, there is still a time window between inspections when exploited robots can be abused **Why?** Having no verification whether the internals of the robot were accessed or not means that attackers can easily tamper with any internal components or install a hardware Trojan unnoticed **Evaluation** **DID**	

2. Network layer

Component	Evaluation criteria	Results
2.1 Internal network	**2.1.1 Monitoring and alert capabilities** **Objective** – identify whether internal network activity is monitored and alerts are issued based on known signatures or anomalies **How to** • Enumerate internal network and find entry points (e.g. switch) • Connect to the network and attempt to perform network based attacks (e.g. ARP poisoning, denial of service on a particular node) and verify whether an operator receives	

	a real time alert and incidents are being logged and acted upon by reviewing procedures. **Notes** If it is not possible to implement full network monitoring due to hardware limitations. At least there should a capability to detect new unauthorized devices on the network. In general thresholds on IDS of the internal network should be lower than on the external network. Because normal user is usually not supposed to connect to the internal network. **Why?** Proper security controls on the internal network might be quite hard and sometimes even impossible to implement due to hardware limitations or performance requirements. If all other security measures from this document are implemented properly, unauthorized access to the internal network is very unlikely. Therefore monitoring capability should be a sufficient security control. **Evaluation** **DID**	
	2.1.2 Firewall **Objective** – identify whether internal network is separated from the external by the firewall **How to** • Inspect documentation / consult developers / inspect node which is responsible for external communications and identify whether firewall if enabled	

	• Inspect firewall settings and verify that no internal nodes are allowed to communicate to the external network unless it is necessary. • If VPN is used verify that there are rules which allow internal nodes to communicate with the outside world only via the VPN tunnel. **Why?** Firewall can help to further protect internal nodes from the outside and ensure that they cannot accidentally leak data to the external network. **Evaluation** **DID**	
2.2 External network	**2.2.1 Protocol security** **Objective** – check if used protocol is up-to-date, secure and have no known vulnerabilities **How to** • Identify all communication capabilities being present by inspecting documentation / consulting developers / manual analysis • Analyze if used protocol versions provide encryption and mutual authentication • Verify that used protocol is hardened according to industry standards. There is a suggested standard that can be used next to the protocol name. WIFI: If robot acts as an access point – SANS, Residential Wireless Network Audit Checklist	

	https://www.sans.org/media/score/checklists/ResidentialWirelessNetworkAudit.doc - If robot acts as a client, it should be able to support strong encryption standards (WPA2-PSK, WPA2-EAP) Zigbee – Homeland security, Recommended Practices Guide for Securing ZigBee Wireless Networks in Process Control System Environments, Section G, Security Best Practice Recommendations https://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/Securing_ZigBee_Wireless_Networks.pdf Bluetooth - NIST 800-121 Guide to Bluetooth Security, Table 4-2. Bluetooth Piconet Security Checklist http://csrc.nist.gov/publicat ions/drafts/800-121/sp800_121_r2_draft.pdf **Note** Amount of suggestions from the documentation that should be implemented depends on the robot application and required security level. If providing encryption on the protocol level is not possible for some reasons, VPN or application level encryption should be used. **Why?** Vulnerabilities in communication protocols can allow attackers to get unauthorized access to the external network of the robot and intercept or modify any transmitted data **Evaluation**	

	PD/DID	
	2.2.2 Network ports exposure **Objective** – identify whether only necessary network ports are exposed to the external network **How to** • Connect to the same network that is used by the robot for communication and scan all robot's TCP and UDP ports, find the open ones. If possible verify with developers if their presence is required • Attempt to identify what service is running behind an open port and its version • Verify whether identified service is still receiving security updates and has no known vulnerabilities **Note** UDP port scanning can be very slow and in case of time constraints it can be limited to the amount of most popular ports which can be scanned in a given amount of time. **Why?** More open ports mean bigger attack surface and therefore their number should be as low as possible. Services that are exposed should have no known vulnerabilities due to the ease of their exploitation. **Evaluation** **PD/DID**	
	2.2.3 Monitoring and alert capabilities	

	Objective – identify whether external network activity is monitored and alerts are issued based on known signatures or anomalies **How to** • If external network is password protected attempt common password guessing. Verify whether an operator receives a real-time alert and the incident is being logged and acted upon by reviewing procedures. • Connect to the external network • Try to perform network based attacks (e.g. network scans, ARP poisoning, denial of service) and verify whether operator receives a real-time alert and the incident is being logged and acted upon by reviewing procedures. **Why?** Properly configured external network monitoring can spot network based attacks in their inception even if other security mechanisms are compromised. **Evaluation** **DID**	

3. Firmware and Operating system layer

Component	**Evaluation criteria**	**Results**

3.1 OS	**3.1.1 Underlying OS updates** **Objective** – verify that the used operating system is still supported by the manufacturer and there is a mechanism to perform system updates • Check if the underlying OS is still maintained and receive security patches • Check whether the latest security updates are applied • Check if there is an update mechanism present **Why?** Outdated operating systems can have security vulnerabilities **Evaluation** **DID**	

3.2 Firmware	**3.2.1 Firmware updates**
	Objective – check if manufacturer firmware can be securely updated
	How to
	• Identify if there is a mechanism to deliver firmware updates
	  • Verify that updates are cryptographically signed
	  • Verify that the signature is verified prior to installation
	Why?
	If new vulnerabilities are discovered it is important to ensure that there is a way to provide updates to all the devices that are already sold to customers. However, update mechanism	

	can be circumvented by an attacker to deliver malicious update. Therefore, it is important to verify the origin of the update prior to installation. **Evaluation** **PD**	
	3.2.2 Integrity check **Objective** – identify whether the system performs an integrity check of critical components and takes action if they are not present or modified. **How to** • Consult documentation / developers to find whether integrity check for critical components is being present • Try disabling or modifying critical components (e.g. safety sensors or range finding systems) of the robot and check if operator receives a real-time alert and the incident is being logged. • Check whether robot continues to function afterwards. Its operation should be stopped as soon as any critical component is disabled or modified. (e.g. if a proximity sensor is disabled the robot should not be able to move, because it will not be able to spot obstacles and can easily do some physical damage) **Note** Critical components are components that can directly influence robot operations, functionality or safety **Why?**	

	Tampering with any of the critical components can make robot to cause physical to people and property **Evaluation** **DID**	

4. Application layer

Component	Evaluation criteria	Results
4.1 Accounts	**4.1.1 Default passwords** **Objective** – identify presence of default passwords **How to** • Review documentation / consult developers and identify whether default passwords are used • Attempt to login with commonly used passwords • If default passwords are used verify whether their change is enforced on the first use • If unique passwords are created on a per device basis, ensure that they are random and not in a sequential order **Note** When trying commonly used passwords beware of account lockouts and verify that there is a recovery mechanism present. **Why?** Default passwords are easy to find on the internet and so	

	far, remain the most popular and easy way to exploit internet connected devices. **Evaluation** **TD**	
	4.1.2 Password complexity **Objective** – verify that password complexity is enforced **How to** • Attempt to change password to a weak one and verify if change succeeded **Why?** Weak passwords take little time to guess **Note** Password complexity requirements depend on the sensitivity of the application. In general the minimum requirements that should be in place are: • Password length at least 8 characters • Enforce usage of 3 of 4 categories (lower-case, upper-case, numbers, special characters) **Evaluation** **TD/PD/DID**	
	4.1.3 Login Lockout **Objective** – identify whether the login lockout is present **How to** • Attempt to login with incorrect credentials multiple	

	times. Verify that the account has got locked out. **Why?** Having strong and non-default passwords is not enough. And brute force attempts should be prevented by implementing a login lockout mechanism. **Note** The lockout threshold depends on the sensitivity of the service. In general, it should be 5 login attempts or less. Prior to testing verify that lockout recovery mechanism is being present. Accounts can be either locked out for a specific duration of time and/or they can be recovered by physical interaction with the robot. **Evaluation** **TD/PD/DID**	
	4.1.4 **Hardcoded or backdoor accounts** **Objective** – identify presence of hardcoded / backdoor accounts **How to** • Consult documentation and developers to identify whether hardcoded / backdoor credentials are used • Analyze the source code for hardcoded / backdoor credentials **Why?** Hardcoded / backdoor credentials pose same danger as default passwords. However, their identification is usually harder due to the need for reverse engineering or	

	possession of the source code **Evaluation** **TD/PD/DID**	
	4.1.5 Cleartext passwords **Objective** – identify whether passwords are stored in cleartext **How to** • Review the source code and documentation / consult developers and identify whether passwords are stored in a cleartext **Why?** Cleartext passwords can be leveraged by an attacker for privilege escalation or lateral movement **Note** Lockout threshold depends on the sensitivity of the service. In general, it should be 5 login attempts or less. **Evaluation** **DID**	
4.2 Authorization	**4.2.1 Authorization** **Objective** – verify that resources are accessible only to authorized users or services **How to** • Login with authorized credentials and attempt to perform different actions, record the requests that are being made	

	• Log out and attempt to send same requests as an unauthenticated user. Verify whether it is successful • Log out and login again as a user with lower access rights. Attempt to send same requests again. Verify whether it is successful **Why?** Access to the restricted functions by anonymous users or users with lower access control rights diminishes all the benefits of access control **Evaluation** **TD** if resource is accessible by anonymous user otherwise **PD/DID**	
4.3 Communication	**4.3.1 Encryption** **Objective** – ensure that all sensitive data is transmitted over an encrypted channel **How to** • Intercept connection between a robot and a control center application / cloud server • Use protocol analyzer to verify that transmitted data is encrypted **Why?** If data is transmitted in a cleartext attackers can easily gather sensitive information (e.g. credentials, audio and video streams, private data) **Evaluation**	

	TD if data is transferred over the internet or publicly accessible network otherwise **PD/DID**	
	4.3.2 Replay protection **Objective** – ensure that transmitted data cannot be replayed **How to** • Intercept connection between robot and control center application / cloud server • Record control or configuration packets sent to the robot • Attempt to replay them and verify whether the desired action is executed **Why?** If replay protection is absent attackers can record legitimate packets and then arbitrary replay them to achieve desired actions **Evaluation** **PD/DID**	
4.4 3rd party libraries and components	**4.4.1 Vulnerabilities** **Objective** – verify that 3rd party software components do not have known vulnerabilities **How to** • Identify which 3rd party libraries and components are used and what are their versions • Look for known vulnerabilities in the current version	

	• Verify whether identified component is still receiving security updates and has no known vulnerabilities • Verify that the latest security updates are installed **Why?** It is quite common to blindly rely on 3rd party components. However they can easily introduce a vulnerability into the product where they are used. **Evaluation** **PD/DID**	
4.5 Privacy	**4.5.1 Privacy** **Objective** – identify whether the robot is compliant to the laws and regulations that apply **How to** • Verify that minimum Personally identifiable information (PII) is collected and transmitted over the internet • Verify that if PII is collected users are made aware of it. (e.g. in case of a video recording people can be warned by stickers or signs on the robot) • Verify that all PII is stored and transmitted in a secure manner **Note** It is not relevant to the security of the robot itself. However not complying with privacy regulations can result in financial consequences and therefore should be taken into consideration.	

	Why? Not complying with regulations could result in financial consequences **Evaluation** **DID**	
4.6 Control center application	**4.6.1 Web application** **Objective** – perform a security assessment of the web application **How to** • Identify web interface that is being used (hosted on the robot itself or a cloud server) • Use OWASP methodology to test web application against OWASP Top 10 Web application vulnerabilities **Why?** Robot can be indirectly compromised if attacker exploits a web control center application **Evaluation** **PD/DID**	
	4.6.2 Mobile phone application **Objective** – perform a security assessment of the mobile application **How to** • Identify whether robot has a mobile app that can be used to control or interact with it.	

	• Test the application against OWASP Mobile Top 10 **Why?** Robot can be indirectly compromised if attacker exploits a mobile phone control center application **Evaluation** **PD/DID**	