Privacy and security challenges for autonomous agents

A study of two social humanoid service robots

Dennis Biström*, Magnus Westerlund*, Bob Duncan† and Martin Gilje Jaatun‡§
*Arcada University of Applied Sciences, Finland
†University of Aberdeen, Scotland
‡University of Stavanger, Norway
§SINTEF Digital, Trondheim, Norway

***Abstract*—The development of autonomous agents have gained renewed interest, largely due to the recent successes of machine learning. Social robots can be considered a special class of autonomous agents that are often intended to be integrated into sensitive environments. We present experiences from our work with two specific humanoid social service robots, and highlight how eschewing privacy and security by design principles leads to implementations with serious privacy and security flaws. The paper introduces the robots as platforms and their associated features, ecosystems and cloud platforms that are required for certain use cases or tasks. The paper encourages design aims for privacy and security, and then in this light studies the implementation from two different manufacturers. The results show a worrisome lack of design focus in handling privacy and security. The paper aims not to cover all the security flaws and possible mitigations, but does look closer into the use of the WebSocket protocol and it's challenges when used for operational control. The conclusions of the paper provide insights on how manufacturers can rectify the discovered security flaws and presents key policies like accountability when it comes to implementing technical features of autonomous agents.**

***Index Terms*—Cloud computing, humanoid robot, accountability, security, privacy, DLT, agent, autonomy**

I. Introduction

Autonomous agents are often defined as physical entities that observe and interact with the environment without direct human control. Robots with humanoid characteristics have received considerable attention and are often viewed as a future human alternative to perform menial tasks [1]. The physical space in which these robots are integrated can often be considered sensitive and therefore requires us to carefully consider and monitor their trustworthiness. This paper presents an investigation into how two social humanoid service robot manufacturers implement security and privacy.

A humanoid service robot is a physical robot with human-like features built to provide customers with some form of service. The robots are considered social robots, meaning they communicate with humans. Robots are considered agents if they perceive and interact with others or the environment, and if they do not need a human operator they are considered autonomous agents. Autonomous robots rely on sensors to perceive their environment, and the sensors produce various data streams, for example numerical and textual representations of images or audio. The perceived environment may trigger the robot to perform some form of action [2]. In comparison with IoT devices, the autonomous operation of a robot may require a more sophisticated AI, however, the architectural setup and vulnerabilities are often similar to more traditional IoT setups.

Data streams can be processed locally by the robot, for example when an infrared sensor triggers an interaction with a human. Data streams are also often processed remotely. Some use cases require cloud processing power and others may require human interaction remotely, as in the case of tele-operation. Use cases that require remote processing often transmit data from the robot's sensors over the internet to a geographically indiscriminate location. In terms of security and privacy, any transmission of live sensor data is of significant concern and demands a high trust relationship toward the manufacturer and operator.

The main contribution of the paper is to shed light on the security challenges present in commercial humanoid robots. We further present suggestions as to which of the outlined challenges are worth addressing and in what order.

The outline of the paper is the following. Section 2 first details the design aims for privacy and security, and then discusses the requirements for Zero Trust architectures. Section 3 presents the use case for the studied robots, and in section 4 we define their ability to collect sensor data. Section 5 and 6 respectively report the results from the study and provide a mitigation analysis. Section 7 concludes the paper.

II. Literature review

Achieving trustworthiness for humanoid social robots demand that we consider several aspects. Trustworthiness as a term aims at being holistic in the assessment of an AI system, including consideration of system robustness and safety [3]. Evaluating trustworthiness requires that we consider the AI solution's technical, legal and ethical aspects, but also that we consider the process for data collection and the implications that may materialize if data is misused. Another concern relates to internal threats, when administrators and operators act as trusted entities with access to data and then abuse this trust.

In the subsections below we discuss and review selected literature on how to achieve both system security and privacy. First we discuss the privacy component and the design considerations on a technical and organisational level. Then we briefly review the tenets for achieving security, and finally we summarize the Zero Trust standard and its application to, and impact on, the field of social robotics.

A. Towards Privacy by Design

With the advent of inexpensive, portable cameras, Warren and Brandeis [4] tried to define privacy as ”the right to be left alone”, but that in itself is an insufficient definition in our connected world. Later definitions have linked privacy to anonymity, identity, and confidentiality [5]. These definitions often present ethical tensions, which mean they stand in conflict with each other. It may also be impossible or at least impractical to achieve a specific component of privacy, for example, the image of a human face can never be considered truly anonymous in today's connected world. Further, confidentiality can be claimed by a company but most solutions show very little or no evidence for operational confidentiality.

Robots require sensors to perceive the environment. Some of these sensors, such as an HD camera, are clearly more invasive than others, say an IR sensor, when it comes to privacy. For certain purposes we need certain sensors, and an invasion of privacy might even be considered acceptable by the user in some cases (home security camera, Face ID/Fingerprint sensor). However, the infringements are not created equal. A home security camera alerting the owner of movement automatically using a script that monitors picture composition can be considered less of a privacy infringement than a security operator looking at the bedroom live camera feed. That being said, many home security products offer a cloud solution that is impervious to most users for assessing their privacy.

Ultimately, privacy is more clear in a legal sense than in today's technical implementations. Many countries have adopted laws guaranteeing residents a right to privacy that depending on the country may then be further specified and strict. As an example, in the European Union the right to privacy has received an interpretation that states that personal data cannot be transferred to third countries without a proper impact assessment and technical measures guaranteeing that data cannot be exposed.

The implication for social robots that are integrated into sensitive environments (e.g., schools or healthcare) in the EU, suggests that in practice data cannot be transferred either to the USA or China if the respective entities have access to encryption keys (cf. GDPR art. 28.1). Further, the service robot should by default deny any incoming or outgoing connections to such third country jurisdictions, including potential cloud infrastructure operators, if there is a possibility that personal data can be misused. Hence, to ensure legal compliance, payload data, telemetry data and operational commands should only be transferred or accepted within EU boundaries. Other jurisdictions have stipulated a similar reasoning based on national security considerations (e.g., USA banning certain Chinese operators), but in the EU this has been granted as a general human right that applies not on an operator level but for all services equally.

This brings us to the concept of *accountability*, which helps define the role of being a responsible steward of other people's (often personal) data [6]. The term is focused on principles put in place to assure appropriate technical and organisational measures to be able to demonstrate activeness and effectiveness when requested. Accountability requires Transparency (how data is handled), Responsiveness (act if something is wrong), Remediability (be in a position to fix things that are wrong), and Responsibility (own mistakes). While accountability was introduced in for example the early GDPR drafts [7], almost a decade later, accountability remains a poorly followed concept.

An international privacy by design resolution was established during the 32nd International Conference of Data Protection and Privacy Commissioners (2010). Enisa has further published a set of privacy techniques for privacy by design that are also relevant in defining and achieving privacy for autonomous agents as well [8]. Many of these overlap with techniques for security by design that we discuss in the following.

B. Towards security for autonomous agents

Software security has received plenty of attention over the last decades and is often easier to define technically than privacy. A common definition of security is to refer to the CIA triad, Confidentiality, Integrity and Availability.

Access to data streams from sensors must be controlled to prevent unauthorized use of the data. Controlling who or what is authorized to use the data streams is a first step in achieving confidentiality.

Data from any sensor could be tampered with, so manufacturers must remember to employ means of ensuring and/or verifying data integrity. Another component of integrity is non-repudiation, ensuring that no party can deny that it sent or received data (eg. by using digital signatures). When using cloud backends for data collection and/or model training/inference, we suggest that extensive logging is required to verify integrity and to monitor confidentiality.

Ensuring data availability is especially important for service robots since both operators (AI or human) and customers (APIs or human) are dependant on a fully functional chain of data streams for the cause and effect required in a specific use case to take place.

A remaining challenge within software security research are attacks from trusted resources, such as internal threats or external actors gaining access by obtaining proper credentials. Additionally, the term operationally autonomous robots suggests that the security should also be autonomous.

C. Zero Trust Architectures

In defining software architectures NIST has provided clear guidelines for how to achieve both security and privacy. In

NIST's Zero Trust Architecture [9], trust is never assumed; on the contrary, it is assumed that an attacker is already present in the network. This implies that access to data is not based on where a subject is making its request from, but requires authentication of every access attempt. This brings us full circle back to the era before the introduction of the network firewall [10], when computers connected to the internet had no perimeter defence to rely on.

Zero-trust architecture is founded on seven tenets [9]:

1) Data and services are considered resources
2) Communication is secured regardless of location
3) Access to resources is granted per session
4) Access is based on dynamic policy
5) Integrity and security posture of all resources is monitored
6) Authentication and authorization is dynamic
7) As much information as possible is collected on assets, infrastructure and communication; this is used to improve security

In some sense, we could argue that zero trust is a misnomer – even though we do not assign trust to any physical location (nobody is granted access just because they happen to be communicating from a certain network segment), this also means that *only actors that we trust* (because they have been explicitly authenticated and authorized) are allowed access to any resource.

III. Use Case Description: Social Robots

In order to examine and understand the privacy and security challenges for autonomous agents interacting with users, we focus our investigation on social humanoid service robots. We chose to examine two of the robot platforms purchased for Human-Robot Interaction (HRI) research purposes at Arcada UAS in Helsinki for our review. These are 1) the CSJBot **Amy** waitress robot (Fig. 1) and a 2) **Sanbot** Elf (Fig. 2).

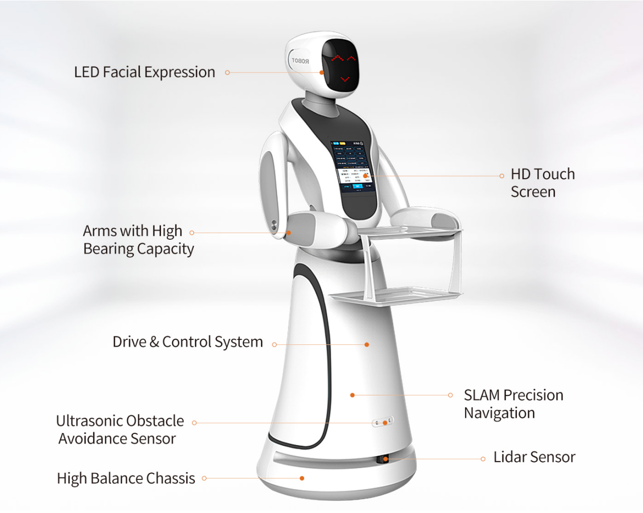

LED Facial Expression
HD Touch Screen
Arms with High Bearing Capacity
Drive & Control System
SLAM Precision Navigation
Ultrasonic Obstacle Avoidance Sensor
Lidar Sensor
High Balance Chassis

Fig. 1. CSJBot Amy and sensors

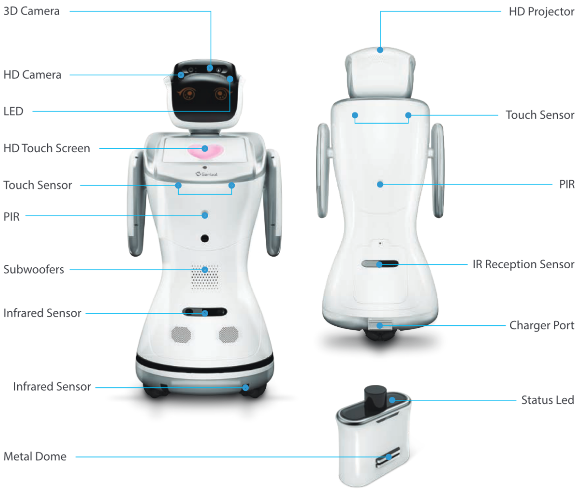


Fig. 2. Sanbot Elf and sensors

A related research project has examined use of the humanoid robots within a non-sensitive health care setting, and had settled on co-creating platforms (applications) for both robots in a preventative dental care setting, based on focus group interviews with dental professionals (n=10) [11]. The humanoids were to be placed in the waiting room at the dentists office with the purpose to converse with the patients. The robot was to guide patients through an informative conversation on how to use dental floss correctly. During development, developers grew concerned with regards to maintaining patient privacy and integrity at the dental care office. This, combined with the outbreak of the COVID-19 pandemic resulted in us opting for a simulated dental office appointment in a lab setting (n=14) followed by interviews lasting 29 minutes on average.

IV. Feature selection

For the purposes of this paper, the features (sensor data) of the robots will first be outlined. Some of the features require closer inspection, while others are less problematic. For example, a PIR proximity sensor's data stream poses minimal security or integrity issues compared to a live video or audio stream. This paper does not outline all sensors and the associated risks, but a selected few are considered worth mentioning by the authors.

The studied robots are designed for specific use, and the robot's sensors are chosen for the features that are required for their respective use case. Amy is a waitress robot, and will deliver food from the kitchen to the table, and return the dishes if desired. To accomplish its tasks, the robot needs the sensors and output devices as outlined in Table I.

The Sanbot is a ”general purpose robot”, and has several features and sensors that are not in use for some of it's use cases. The European reseller of Sanbot robots [12] lists the following use cases:

TABLE I
ROBOT FEATURES AND ASSOCIATED IMPLEMENTATION DETAILS

Feature	Implementation
A way to map and navigate its surroundings	SLAM via LIDAR
A secondary way to detect obstacles in its direct way	Ultrasonic sensors
An interface for updating today's menu	Tablet on its chest
A way to line up its charging contact with the charging station	IR sensor
A one way interface for communicating when food has arrived	A speaker
An interface to confirm the reception of food	A touch sensor in its hand

1) A robot tutor projecting slides and repeating instructions for kids
2) Greet and attract shoppers and answers their questions
3) Monitor patients in health care and record medical records
4) Security guard that patrols and uses face recognition to detect intruders

The Sanbot Elf speaks English and Chinese when shipped from the factory, and uses a cloud back-end for the language processing (NLP), see Figure 3 for a rough architectural overview. The voice recorded on the robot is locally converted to text using a text-to-speech model. This text is then sent to a cloud server for processing and the forming of an appropriate reply. The reply is received by the bot in text format, and synthesized to audio that plays on the robot. Amy, on the other hand, doesn't have a microphone.

The Sanbot uses its PIR sensor to react to when someone stands in front of or behind it. The camera is also used to detect faces, when someone leans over to talk to it. The camera has a face detection algorithm that runs locally on the robot, and when a face is detected, an image of the face is sent to the cloud server to see if it matches one of the registered ”VIP:s”. This feature is used for surveillance purposes, but also for customisation purposes where the robot can recognize its owner/-s. Amy doesn't have a camera, but uses an ultrasonic sensor for detecting obstacles.

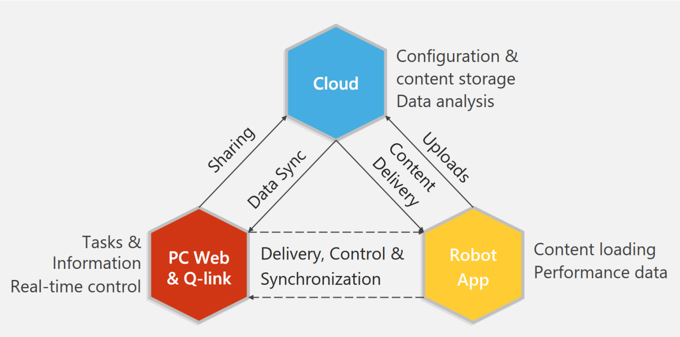
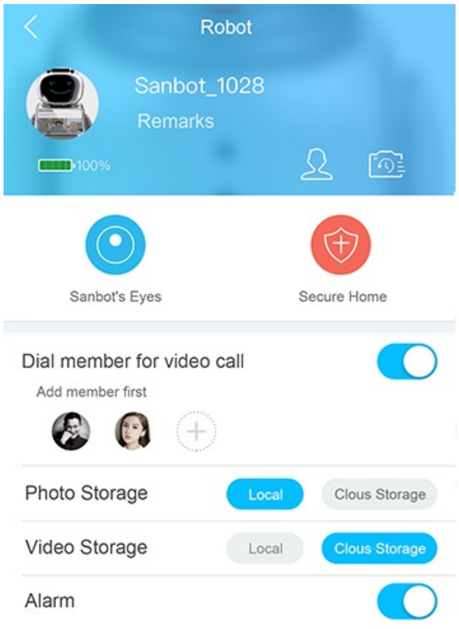


Fig. 3. Sanbot Multi Platform Service includes real-time control and cloud data storage


Robot

Sanbot_1028

Remarks

100%

Sanbot's Eyes

Secure Home

Dial member for video call: on

Add member first

Photo Storage: Local | Clous Storage (Local selected)

Video Storage: Local | Clous Storage (Clous Storage selected)

Alarm: on

Fig. 4. Most of Sanbots features are available through the mobile app

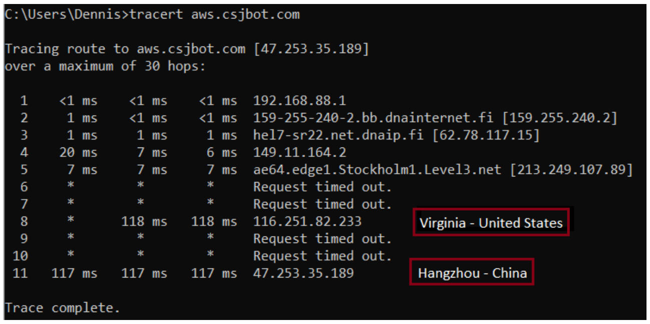
Sanbot offers a mobile app (Fig. 4), through which many of the features of the bot can be used. Remote monitoring with audio, remote control of the movement, speech interaction and app launching are all features of this app. The features are not limited to users on the same network but available from wherever in the world. Amy can also be controlled remotely, but only via a local area connection to the on-board wireless base station.

V. Results - Security and Privacy Issues

With the functionality and usage of the sensors outlined we can proceed to look closer into the data streams from and to the robots. We approach our data collection by first looking into the design liberties taken by the manufacturers, then looking at functionality related to the cloud and connectivity features.

In the manufacturing of the robots some liberties have been taken when it comes to systems designs. The Amy robots ship with default password for the built-in wireless network and the settings interface. The manual shipped with the robot does not state one should change these passwords, this can be considered to be the responsibility of the user. However, since you can control Amy by connecting to her wireless network, usage of the default password gives you access to remote control of a big and heavy physical entity.

When configuring Amy for operation, one needs to connect Amy to the internet. This is accomplished through the tablet interface. Once established, this connection is shared to all connected devices (i.e. the router is in bridged mode) so that the operator of Amy besides having access to the robot also can connect to the internet. This means that if the default



```
C:\Users\Dennis>tracert aws.csjbot.com

Tracing route to aws.csjbot.com [47.253.35.189]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.88.1
  2     1 ms    <1 ms    <1 ms  159-255-240-2.bb.dnainternet.fi [159.255.240.2]
  3     1 ms     1 ms     1 ms  hel7-sr22.net.dnaip.fi [62.78.117.15]
  4    20 ms     7 ms     6 ms  149.11.164.2
  5     7 ms     7 ms     7 ms  ae64.edge1.Stockholm1.Level3.net [213.249.107.89]
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     *      118 ms   118 ms  116.251.82.233
  9     *        *        *     Request timed out.
 10     *        *        *     Request timed out.
 11   117 ms   117 ms   117 ms  47.253.35.189

Trace complete.
```

Fig. 5. Traceroute followed by IP localisation

password of Amy has not been changed, an attacker could gain access to companies' intranets through Amy.

Amy needs to be set up (by a technician) before starting its duties. After the initial setup, all customer interaction with the bot is through the tablet interface on its chest. Some features, like remote control, are left activated after the initial setup. The manufacturer CSJbot also offers a cloud platform where one can among other things monitor video feeds from their robots. During data collection, a traceroute command was run so as to gain insights to how and possibly where the cloud platform was hosted. The traceroute command listed some IP addresses at the interface level, see Figure 5. These IP addresses could then be roughly geolocated in an attempt to learn more about how the cloud platform is hosted. Some data was collected which upon some conclusions could be drawn. The hostname of the fifth hop is a level 3 communications server. Level 3 was acquired by the American company CenturyLink in 2017 (now Lumen technologies). After this we find the traffic routed through two servers owned by Alibaba, one in Hangzhou, China and another one in the United States. [13] [14]

Csjbot has a cloud platform hosted on an unsecured site `http://aws.cjsbot.com`. The user account and password was pre-defined and given to us per mail. The password only contained one of the following: letters, numbers, special characters, and was very short in length. All characters are in one sequential pattern. The default password can however be easily changed once logged in. The only requirement for changing the password is that it needs to be four characters in length.

This cloud platform contains a control panel where besides controlling the robots we can listen to the mic and receive a live feed from the video camera on the robot. When we questioned the security of the platform, the developer informed us that "the site is safe" and that we could change the password ourselves.

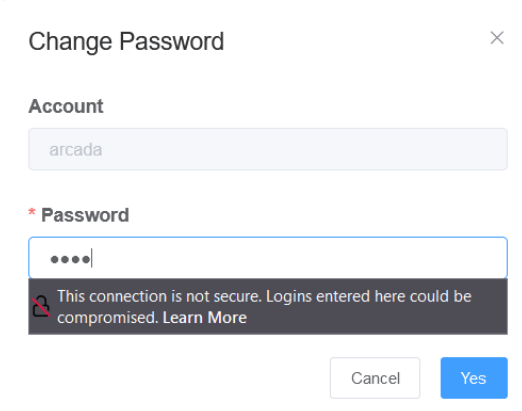
For Sanbot, the app allows you to add a robot to your account to gain access to it. This is done via a QR code on the bots. As the owner of the bot, you can add other members using just a Qlink-username. Within a month of starting to use Sanbot, someone sent us a friend request which we accepted. By accepting a friend request the friend gains access to all Sanbots you own and vice versa. The mystery friend was fortunately a kind robot enthusiast from Germany by the name of Markus. He could now monitor us in Finland and we could now monitor him whenever our Sanbots were turned on. We contacted Markus and told him he had granted us live audio and video feeds to his Sanbot whenever it was turned on. We also tried to figure out how he had found and added us as friends, whether it was my username that was easy to guess (thesourmango) or if he had tried adding the wrong Sanbot by mistake due to the bots simple naming scheme (Sanbot-2048, Sanbot-2029). Exactly how he managed to add us as friends was never established, even though we did conclude that 1) he added us by mistake and that 2) the bot was new to him.

Amy can be controlled inside a software called *robot studio* during setup of operations. Besides this, Amy runs an http server with a WebSocket endpoint, for enabling the control of Amy through a local API. The endpoint does not use any type of authentication but requires the controller to be on the same WiFi network.

VI. A CLOSER LOOK - THE WEBSOCKET INTERFACE

As we cannot dive into depth on every one of these security failures, we have chosen to examine only one of the issues in depth. Exercise of control between the system and Amy runs on an open WebSocket, which is renowned for unresolved vulnerabilities [15]. We would like to present a list of potential attack vectors, followed by a table for suggested actions to mitigate security issues with WebSocket II. Please note that the problem with most of these vulnerabilities is not the implementation of WebSockets as a communication protocol for humanoid robots as a whole, but instead the specific way that WebSockets have been implemented on this robot in particular. The secure alternative protocol WSS combined with authentication (and the other suggested improvements) can be a perfectly good alternative for the current implementation.

DoS Attacks - WebSockets allow an unlimited number of connections to reach the server, allowing an attacker to flood



Change Password

Account

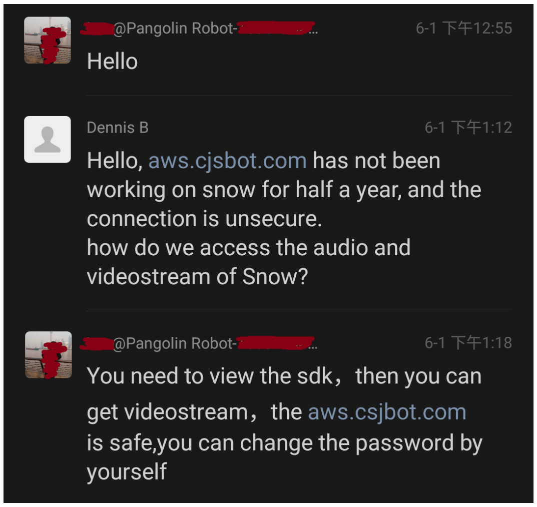
arcada

* Password

••••

This connection is not secure. Logins entered here could be compromised. Learn More

Fig. 6. The password *1234* is accepted and sent over http, Firefox warns the user about the insecure connection



[redacted]@Pangolin Robot-[redacted]... 6-1 下午12:55

Hello

Dennis B 6-1 下午1:12

Hello, aws.cjsbot.com has not been working on snow for half a year, and the connection is unsecure.
how do we access the audio and videostream of Snow?

[redacted]@Pangolin Robot-[redacted]... 6-1 下午1:18

You need to view the sdk，then you can get videostream，the aws.csjbot.com is safe,you can change the password by yourself

Fig. 7. Developer telling us a site hosted over http is safe

the system.

No authentication - The WebSocket protocol doesn't allow server-client authentication during the handshake process, leaving only HTTP connections available. This can be exploited by a Cross-Site WebSocket Hijacking attack.

Unencrypted TCP - WebSockets can be used over an unencrypted TCP channel, leading to all kinds of issues addressed by the OWASP Top 10 A6-Sensitive Data Exposure list.

Input Data Attacks - WebSockets are vulnerable to Cross-Site Scripting attacks.

Data Masking - WebSockets protocols use Data Masking to stop issues such as proxy cache poisoning, but it will prevent security tools from attempting to pattern match activities in traffic. Indeed Data Loss Prevention (DLP) software is not even aware of the existence of WebSockets, meaning they are unable to identify malicious code or data leakage.

Authorization/Authentication - The protocol does not handle Authorization nor Authentication. This must be handled separately by the implementation.

Tunneling - Since the WebSocket allows connections by anybody, this provides an open door for a Cross-Site Script using the connection as a tunnel.

Sniffing Attacks - Data transfer is carried out in plain text, meaning all data transfer is vulnerable to a man-in-the-middle attack. By using the WSS protocol at least the data is transferred via Transport Layer Security (TLS) meaning the data gets encrypted. Note that this alone does not ensure security.

VII. Analysis and Mitigations

Our discoveries prompt us to consider some of these security issues more carefully. The developers' assertion that the robot cloud-based website "is safe" is somewhat economical with the truth. It's easy for the developers to make a simple assertion — as they are personally not responsible for compliance with the EU GDPR. However, a breach on an insecure website will result in potentially serious consequences for the corporate using this system.

Let us consider the OWASP Top Ten IoT Weaknesses [16]:

1) Weak, guessable, or hardcoded passwords
2) Insecure network services
3) Insecure ecosystem interfaces
4) Lack of secure update mechanism
5) Use of insecure or outdated components
6) Insufficient privacy protection
7) Insecure data transfer and storage
8) Lack of device management
9) Insecure default settings
10) Lack of physical hardening

If we look carefully at the details provided for robot 1 (the CSJBot Amy waitress robot), we can see that it will fail on all of the OWASP Top 10 weaknesses. Robot 2 (the Sanbot Elf) fares little better.

There are many mitigations that have been developed for the OWASP Top Ten IoT vulnerabilities, and these should certainly be implemented in order to considerably improve matters further.

If manufacturers need to use a default password, as is the case for many routers, there are simple policies that can be employed to improve security and accountability. With routers most manufacturers configure the default password to be different for each manufactured device. This practice would reduce the vulnerability of Amy robots globally. Another simple improvement to accountability and security would be to force the user to change the WiFi password on the first login. The same policies are recommended for the cloud platform. In addition, the protocol should be upgraded to a secure one, using https instead of http.

Regarding the WebSocket implementation, an open endpoint without authentication is not recommended. The requirement of being in the physical vicinity of the Amy robot is good, but since Amy as a default setting routes all traffic from the parent network to the internal network, anyone connected to the parent network of Amy's hotspot is also able to access the open WebSocket endpoint. A clear improvement to the security of the system would be updating to the secure wss protocol instead of using ws, and below we list suggestions for further improving the security of the WebSocket implementation:

Matters get worse once we dig further into the detail of the systems architecture. There is the ability to listen on the mic and the possibility to record video. Neither demonstrate the use of any security protocols.

In the Q-link app, when adding members, the default permissions for the added friend should not be direct access to video and audio feed of all the robots associated with the account.

Android Debug Bridge or ADB is a powerful utility that can be used for system level access on the Android platform. When we asked the developer advice on how to connect the robot to the computer (to install an app we developed) we were advised to install an app enabling ADB over WiFi. This is very handy since we don't need to open up the chassis of the

TABLE II
WEBSOCKET MITIGATION ACTIONS

Mitigation	Description
WSS protocol	The use of the ws:// protocol does not offer secure transport. Instead, rather use WebSocket Secure (wss://) which is a much safer protocol. The big issue with WebSockets is that it is meant to be versatile, with the established connection being always open, allowing continuous sending and receipt of messages. But without authentication/authorisation vulnerability to data input attacks remain exploitable.
Client Data Validation	Construct a validation of arbitrary client data as well as what is coming in from users.
Server Data Validation	To ensure the server is not sending out compromised data, always assume messages received on the client-side are processed as instructions.
Ticket/Token Infrastructure	Authentication and authorization-based ticket/-token infrastructure of incoming requests to all WebSockets.
Prevent Tunnelling	Disallow the use of tunnelling to hinder the extraction of data for malicious purposes.
Use Rate Limiting	Restrict the volume of information the client can send and also limit the server response to the client.
Use the Origin Header	Activating this, the system will record which host the request is coming from. It will not stop the attacker from changing the origin header. While that is blocked by most modern browsers, it will not stop a determined attacker. However, it does provide an additional strand of forensic trail that can be collected.

robot to connect a USB cable. However, ADB allows for some very powerful system level commands and enabling wireless access opens up a potential back-door, especially considering most bots may be running using the previously mentioned default password. ADB over WiFi is a very powerful and indeed versatile tool that provides serious assistance in the course of developing systems, sadly security is not one of its best traits. Thus the only positive action we should take with ADB over WiFi would be to block it in use to provide a proper level of security.

VIII. CONCLUSION

The emergence of commercial social robots aimed at being introduced into sensitive environments as a potential replacement for low-skilled labor, demands products that are reliable and safe. In this paper we examine robots from two manufacturers, and while the critique is specific the problem is much broader. The critique is intended as constructive, a wake-up call for manufacturers and their customers. However, we aim at keeping the abstraction level higher than just the security concerns of these two robots.

We have presented a number of privacy and security flaws that exist in two social robots that are commercially available today. Software design choices made during development were likely done to facilitate ease of use for developers, but not addressing these in the final commercial products intended for a general market is highly problematic.

Among the discovered issues we found that unencrypted protocols were commonly used. From an integrity and privacy standpoint the use of unencrypted protocols for video and audio transmission is unacceptable. Apart from unencrypted protocols we found several of the features of the robots were implemented without considering proper data transfer and management policies, sending captured images to global cloud servers for detection and storage without further information or consent from the customer/user of the robot.

The adoption of autonomous agents in society will demand a clear design focus on practices that facilitate privacy and security. Zero Trust architectures offer a method for ensuring not only the authentication of users, but also authorization. For social robots that rely on cloud machine learning back-ends to implement perception and interactivity, they have much to gain from zero trust architectures.

We provide insights on how the manufacturers could have rectified the discovered security flaws by following well-known privacy and security by design principles. We also find the OWASP list of IoT weaknesses useful also in the evaluation of humanoid robot platform security. While the list does not consider AI security directly, it offers a relevant understanding of problems robot developers face.

Accountability also proved to be a weak point, the manufacturers have clearly not considered data protection or GDPR compliance. Changing an unsecured software architecture at a later stage when products have been sold commercially is both expensive and hard. Training staff to recognise software deficiencies and to acknowledge and affirm their correction can help companies avoid large data violation fines.

One of the biggest challenges corporates face when using any new technology is the inability to assure proper retention of the forensic trail associated with use of the assets. This is an area that we have previously addressed [17]. For autonomous agents this is highly important as no human is directly controlling or monitoring the device.

In summary, considering use cases outside a research lab for robots like the ones investigated demands a risk assessment including both legal and data security staff. Based on our investigation we identify too many open risks to be able to recommend the systems for wider use.

REFERENCES

[1] D. Feil-Seifer and M. Mataric, *Human-Robot Interaction*. Springer, 04 2009, pp. 4643–4659.

[2] M. J. Mataric, “The robotics primer (intelligent robotics and autonomous agents),” 2007.

[3] R. V. Zicari, J. Brodersen, J. Brusseau, B. Düdder, T. Eichhorn, T. Ivanov, G. Kararigas, P. Kringen, M. McCullough, F. Möslein *et al.*, “Z-inspection®: a process to assess trustworthy ai,” *IEEE Transactions on Technology and Society*, vol. 2, no. 2, pp. 83–97, 2021.

[4] S. D. Warren and L. D. Brandeis, “The Right to Privacy,” *Harvard Law Review*, pp. 193–220, 1890.

[5] A. Novak, “Anonymity, confidentiality, privacy, and identity: The ties that bind and break in communication research,” *Review of communication*, vol. 14, no. 1, pp. 36–48, 2014.

[6] M. G. Jaatun, S. Pearson, F. Gittler, R. Leenes, and M. Niezen, "Enhancing accountability in the cloud," *International Journal of Information Management*, 2016. [Online]. Available: //www.sciencedirect.com/science/article/pii/S0268401216301475

[7] EU, "REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," Legal statute, 2016.

[8] G. Danezis, J. Domingo-Ferrer, M. Hansen, J.-H. Hoepman, D. L. Métayer, R. Tirtea, and S. Schiffner, "Privacy and Data Protection by Design — from policy to engineering," ISBN 978-92-9204-108-3, ENISA, Tech. Rep., 2014. [Online]. Available: https://www.enisa.europa.eu/publications/privacy-and-data-protection-by-design/@@download/fullReport

[9] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," NIST SP 800-207, National Institute of Standards and Technology, Tech. Rep., 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

[10] S. M. Bellovin and W. R. Cheswick, "Network firewalls," *IEEE communications magazine*, vol. 32, no. 9, pp. 50–57, 1994.

[11] S. Hägglund, C. Tigerstedt, D. Biström, M. Wingrena, S. Andersson, K. Kuvaja-Adolfsson, J. Penttinen, and L. Espinosa-Leal, "Stakeholders' experiences of and expectations for robot accents in a dental care simulation. a finland-swedish case study." in *Presented in session 10: Robots in Healthcare II on rpc2022*. Aarhus university, 2022.

[12] Sanbot Innovation Technology Ltd, "Cloud-enabled Service Robot, Advanced AI robot — Sanbot Robotics." [Online]. Available: http://en.sanbot.com/product/sanbot-elf/performance

[13] GEO IPIFY, "IP Geolocation Lookup." [Online]. Available: https://geo-lookup.ipify.org/lookup-report/Mb2mbAyg5E

[14] GEO IPIFY , "IP Geolocation Lookup." [Online]. Available: https://geo-lookup.ipify.org/lookup-report/a92BXZQmke

[15] Danko Kovacic, "WebSocket Security: Top 8 Vulnerabilities and How to Solve Them," 2021. [Online]. Available: https://brightsec.com/blog/websocket-security-top-vulnerabilities/

[16] "Internet of Things (IoT) Top 10 2018," 2018. [Online]. Available: https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Top_10

[17] M. Westerlund, M. Neovius, and G. Pulkkis, "Providing tamper-resistant audit trails with distributed ledger based solutions for forensics of iot systems using cloud resources," *International Journal on Advances in Security*, vol. 11, no. 3 & 4, 2018.