Towards data-driven autonomous cyber defence for military unmanned vehicles – threats & attacks

Andreas Dybvik Kaasen & Gudmund Grov
Norwegian Defence Research Establishment (FFI) & University of Oslo
Kjeller, Norway
Gudmund.Grov@ffi.no

Federico Mancini & Magnus Baksaas
Norwegian Defence Research Establishment (FFI)
Kjeller, Norway
{Federico.Mancini,Magnus.Baksaas}@ffi.no

***Abstract*—Unmanned vehicles with varying degrees of autonomy will likely change the way military operations can be conducted, but they also introduce risks that require new ways of thinking security. In particular, the safety ramifications of cyber attacks should be seen as equally critical as the loss of classified data. Developing a cyber defence capability that can detect and manage these potentially harmful events also without human intervention thus becomes a fundamental requirement. In this paper, we commence such work by exploring how to disrupt the functionality of an actual military unmanned ground vehicle given an internal attacker, and how the resulting data can be used to design an an effective detection capability.**

***Index Terms*—Information security, Intrusion detection, Publish subscribe systems, Safety, Autonomous vehicles**

I. Introduction

Unmanned vehicles with varying degrees of autonomy are rapidly becoming an integrated part of our lives as they can efficiently perform many tasks that are either dangerous or just monotonous and physically demanding for us. They range from small and simple systems like vacuum cleaners and grass mowers, to remote controlled drones and submersible vehicles, to self-driving cars and even ferries and space vehicles.

Naturally, these characteristics are also very interesting in a military setting, where they can help increase the operational effect in a mission, while reducing risk to own personnel and more expensive equipment. Their use will likely change the ways missions are conducted and possibly enable completely new types of missions. Not surprisingly, there are many defence programs currently developing and testing military versions of such vehicles, e.g: ground vehicles for base and area protection and surveillance, drones for reconnaissance and communication operations, and water vehicles for mine hunt.

While these vehicles do have high operative potential for their cost, they also introduce new risks that need to be understood as early as possible in order to integrate the necessary security in their design at an early stage. Unlike their commercial counterparts, they are cyber-physical systems with the ability to operate autonomously in a potentially contested environment and equipped with sensitive military data and technology. From a cyber security perspective, this translates into a wider attack surface, new attack vectors, and a much greater interplay between security and safety. The question is then which security controls are needed and how they should be designed to be effective in this new context.

Recent studies provide a first overview of relevant threats and necessary security capabilities [1], [2], among which is the ability to autonomously detect attacks or malfunctions and react in a way that minimises the consequences for the mission: a *fully autonomous cyber defence* capability. The reason is that advanced autonomous vehicles will likely be employed in missions where it is not possible to remote control or monitor the system for extended periods of times, and potential compromises must be handled locally and autonomously. In this scenario, a specific threat is that an internal (privileged) component may already have been compromised before the mission, rendering classical perimeter-based security controls much less effective. This is a growing cause of concern as supply chain compromise keeps being documented and requires more advanced and targeted detection mechanisms.

> ***Threat model:*** It is assumed that an adversary has compromised a component within the system.

Anomaly detection in itself is not a new capability, but it is typically optimised for internet-based network traffic and suffers from many drawbacks, such as an unmanageable number of false alerts. In an unmanned vehicle, we may have a clearer baseline for detecting anomalies and overcome some of the usual drawbacks, but we also need to analyse a different type of data traffic and search for patterns indicating attacks other than those we see in internet-based computer networks.

In this paper we take a practical approach to this problem, using an actual military Unmanned Ground Vehicle (UGV) (§II) to explore and test new types of threats and attacks (§IV) that can constitute the basis for the design of a cyber defence capability. Several of the attacks are tested both in a simulator and on the actual UGV (§IV), with logs extracted to be used in the design of the actual detection algorithm. A proof-of-concept machine learning model is trained on this data §V and preliminary results are presented in §VI. We conclude with a discussion of the possible way forward in §VII.

II. Tor MilRem THeMIS 4.5 UGV & ROS

The work presented in this paper was conducted on *Tor*, a THeMIS 4.5 UGV from Milrem robotics[1] used by the

[1]https://milremrobotics.com/defence/

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/MILCOM55135.2022.10017692

Norwegian Defence Research Establishment (FFI) for experimental development. Tor is a remote controlled and tracked vehicle with a diesel-electric hybrid drive train shown in Figure 1. Tor uses two electric motors and a high-voltage battery pack to drive each track. A diesel generator can be used to power the high-voltage battery pack. Tor is capable of driving at a maximum speed of 20 km/ and the tracks make it possible to drive in rough terrain [3]. A rugged handheld controller and tablet attached to a military vest is used as remote control. This controller communicates with Tor through a Silvus Streamcaster 4200 Radio. Tor can also be controlled using a connected computer with an XBOX controller.

Tor has been equipped with additional sensors and software for autonomous driving capabilities. Some of the sensors used are light sensors, cameras, LiDAR, accelerometers, and GPS. Together with the autonomous software, Tor is able to gather a perception of the surroundings and can predict, plan and execute drive commands. The internal control of Tor is divided

Fig. 1. The Tor Milrem THeMIS 4.5

into two computers, Themis PC and Tor PC. Themis PC hosts the proprietary software from Milrem, including the low-level control over the vehicle's actuators. This computer is only accessible and configurable by Milrem and is the one that the rugged handheld controller communicates directly with. Tor PC hosts the sensor drivers, the autonomous software, and Themis Controller, which is a software component responsible for parsing and sending drive commands from either the XBOX controller or the autonomous software to Themis PC. Figure 2 illustrates this overall architecture.

The communication between the internal components uses the Robot Operating System, ROS. Themis PC and Themis Controller use ROS 2 [4], whilst the sensors and autonomous software use ROS 1 [5]. To allow for these versions to be able to communicate, a ROS Bridge converting the communication between ROS 1 and ROS 2 has been added. Note that ROS 1 was not developed with security in mind, meaning that e.g. communication is sent unencrypted [6], [7]. ROS 2 introduces security mechanisms through the Data Distribution Service (DDS) framework, which we return to in §VI.

The architecture of ROS is based on nodes, messages, and topics. The nodes in a ROS network communicate by

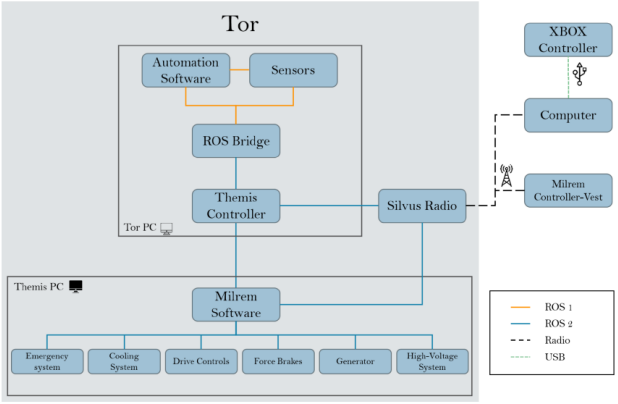

Fig. 2. The Tor Milrem THeMIS 4.5 architecture

registering as a subscriber or publisher on specific topics. A publisher can send messages on a topic which is received by all subscribers to that topic. The ROS network uses the Real-Time Publish Subscribe (RTPS) protocol to send messages between nodes. This protocol uses UDP and makes ROS 2 highly modular and scalable. To enable virtual testing and observation of node interaction and communication, Milrem provides a simulator that FFI extended [3] and that we in §IV used to test the attacks described in the next section.

III. A CATALOGUE OF THREATS AND ATTACKS

Most of Tor's tasks involve collecting, processing, storing and communicating data. In this sense, it is exposed to many of the same threats found in well-established catalogues[2]. The main difference from a pure information system, is that these tasks are performed while navigating through a demanding environment. Thus, much of the data consists of sensor measurements from physical components and control actions that need to be applied to steer the UGV. The compromise of the integrity or availability of this data could lead to serious safety concerns and to the inability of the vehicle to physically perform its mission. This is why the first step in our work has been to explore how it may be possible to disrupt Tor's functionalities under the assumed threat model, i.e, assuming that one of the ROS nodes is under the control of an opponent. This is somewhat different from detecting and preventing the loss of sensitive data, which is typically the focus of cyber security in a military setting, but equally critical.

Table I summarises the types of attacks we explored: Denial of service (DoS); Topic hijacking; Configuration & launch file tampering; and Miscellaneous attacks. More details of each attack, including assumptions and potential consequences in both the digital and physical domains are discussed below. We return to the three rightmost columns in the table later.

a) DoS: In a moving vehicle, it is critical that telemetry information is readily available for the control systems or the operator, as a slight delay in reacting to an environmental

[2]See e.g. Mitre ATT&CK, ISO 27005 Annex C, NIST 800-30 Appendix E

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/MILCOM55135.2022.10017692

TABLE I
CATALOGUE OF THREATS & ATTACKS

Attack	Description	Simulated	Applied on Tor	DDS evading
	Denial of Service (DoS)			
Network Traffic	Flood the network with fake UDP packets	✓	✓	✓
ROS Messages	Flood the ROS network with ROS Messages	✓	✓	(✓)
	Topic hijacking			
Generator Mode	Alter the generator mode	✓	✓	(✓)
High-voltage System	Alter the high-voltage system			(✓)
Drive commands/state	Alter the drive state and insert fake drive commands			(✓)
Cooling System	Alter the state of the cooling system			(✓)
Emergency Stop	Activate the emergency stop system	✓	✓	(✓)
Emergency Mode	Activate emergency mode, which overrides emergency stop			(✓)
Force Brakes	Engage or disengage the brakes	✓	✓	(✓)
	Configuration & launch file tampering			
Timeouts	Alter the timeouts in the configuration file			✓
Topic Name	Alter topic names in the configuration- & Launch file			✓
ROS Bridge	Alter which topics are sent through the ROS Bridge			✓
	Miscellaneous attacks			
Charge Mode	Deactivate charge mode			✓
Clock Skew	Interrupt the clock used by the senors			✓
CPU, RAM & Disk	Requests excessive amount of the available computer power			✓
Side/covert channels	Exploit the system's side effects			(✓)

change or a system warning could have catastrophic consequences. Likewise, critical components must be available in order to apply the control actions.

b) DoS: With access to Tor's internal network traffic, it is possible for an adversarial ROS node to flood the network with fake UDP packets (Network Traffic). The internal hardware has limited capacity related to throughput and bandwidth, and overloading these limits will increase the latency, and packages might be dropped. The messages used to flood the network could be fake, and the only purpose is to congest the traffic. Alternatively, the messages can appear legit and consume processing power and time, e.g. on Tor PC. This attack can also be performed using ROS 2 messages (ROS Messages). This allows one to target specific topics or nodes within the network. An example is the telemetry data sent from Themis PC to Themis Controller, which is a critical topic and could be targeted for this type of attack.

c) Topic hijacking: A compromised node's access rights to a topic can be used to publish fake messages. Every subscribing node will receive the fake messages as if they were real, and execute the commands. Several physical components can simply be turned off or overloaded with this type of attack and they will be challenging to detect as they are seen as legitimate traffic. For instance, Themis Controller can alter and control different components on the vehicle by publishing to the topics which Themis PC subscribes to. The following topics in particular have been used in our attacks: Generator mode, High-voltage system, Force brakes, Cooling system, Drive commands/state, Emergency stop, and Emergency mode.

The consequences of these different attacks vary in criticality and persistence based on the type of component and attack modality. Disabling functions like emergency stop or force brakes could lead to uncontrolled movement. Small changes to the direction of movement could result in a collision or a situation which the autonomous software cannot handle. Rapidly changing the state and operating mode of a component, such as switching the generator on and off continuously, could lead to mechanical stress and consequent physical damage or a reboot of the system and the loss of some data.

d) Configuration & launch file tampering: The setup of a ROS network is defined through a configuration and launch file. These files specify everything, including topic names, nodes, security features, timeouts, protocol versions, and the ROS Bridge. These features need to be configured correctly for the system to operate as intended. At start-up, the configuration file and the launch file are read and used by the system to set up the nodes and the nodes and topics relationships [8].

Access to these files by a compromised node would open the doors to a multitude of attacks. For instance, timeouts are very important for safety, as incorrect values could render Tor unresponsive. An attack against the ROS Bridge could be used to create hidden channels between two compromised nodes and exfiltrate information from Tor. However, for new changes in the configuration or launch file to be used, a restart of the system is required. This increases the difficulty of the attacks, but could e.g. be combined with an attack on the the generator to force a system restart.

e) Miscellaneous attacks: ROS is not the only attack surface an adversary may use to compromise Tor. Thus, for completeness we quickly touch on other possible attack vectors. Tor relies on a variety of sensors to be able to drive autonomously. All these sensor observations need to be synchronised for this interaction between hardware and software to run smoothly. Information gathered from e.g. cameras, LiDAR and GPS are combined based on the precise observation time. Tor PC synchronises the clocks with the Precision Time Protocol (PTP), and a *clock skew* in one sensor could lead to a wrong interpretation of the surroundings and incorrect decisions made by the autonomous software.

Charge mode is used when Tor is charging and prevents the

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/MILCOM55135.2022.10017692

vehicle from moving. If an adversary were able to disengage this mode, it would be possible to control Tor without interruptions from Themis Controller or Milrem Controller Vest as no other systems would be operative.

A *side channel* attack aims at extracting sensitive information about information being processed in a system simply by observing timing or data when specific operations are executed. The unmanned context opens up new possibilities as physical actions made by Tor can be correlated with the messages that are sent and received by the system. If patterns emerge, they can unintentionally reveal confidential information. *Covert channels* aim at manipulating naturally occurring information to communicate even without a formal communication channel, for instance by inducing high and low voltages in the processor to form binary messages. This assumes that the adversary can both induce and observe the behaviour. Here, physical moving parts, lights or sensors on Tor can be used to covertly exfiltrate information from the inside to an external observer.

Similarly to a DoS attack overloading the network, other components are also vulnerable to overload. If a compromised component demands an excessive amount of the available resources, including CPU, RAM or disk, this could affect the rest of the system.

IV. Testing the attacks on the UGV

To show practicality and feasibility of the attacks described in the previous section, a selection of them were applied on Tor. As a preliminary assessment and because some of the attacks can actually break some of the vehicle components, all attacks were first applied in the simulator (§IV-A) and only some where subsequently tested on the actual UGV (§IV-B). This is indicated in the third and fourth column of table I.

A. Simulating attacks

To simulate the attacks we used two virtual machines (VMs). One hosted the attacker sending packets and one hosted the THeMIS Simulator. Both VMs were running on the same network and were hosted on the same physical machine. This set-up allowed us to capture and analyse the traffic (see §V), which is not possible internally in the simulator.

a) DoS attacks: The DoS attack floods the network with fake UDP messages or ROS 2 messages. All ROS 2 nodes on a network use a common discovery port, in our case 7400, and it was used as a target for our UDP attack to maximise effect. After initiating the attack, latency increased drastically, resulting in Tor becoming uncontrollable. This unavailability persisted for the entirety of the attack. After the attack, the simulator returned back to normal. Flooding with ROS 2 messages did not show any particular effect.

b) ROS 2 topic hijacking: This type of attack assumes a compromised node with read or write access to a critical topic. The topics targeted by these attacks are in relation to the generation, telemetry and the brakes.

The attack against the generator consists of sending fake messages on the topic from the Themis PC with a value indicating no power. This attack will trigger a restart of the generator. The fake messages were sent at a rate of 30 messages per second. The effect was immediate, and the high-voltage system was turned off. When stopping the attack, the generator was restarted, although it got stuck in the warm-up phase as an unexpected side-effect.

The telemetry topic is used to activate the emergency stop feature, which stops Tor immediately and turns off the high-voltage system. This attack also resulted in the Themis Controller stopping all communication as it thought the operation was aborted. To continue the operation, a message containing the drive state `IDLE` will be required before the desired state can be set.

Attacking the brake topic with fake messages could cause the brakes to engage or disengage. This attack used the same setup as the previous attacks and successfully altered the state of the brakes. The Themis Controller is unaware of the fake message and continues to send drive commands to Themis PC. An adversary could then be able to stop Tor without the Themis Controller being able to disable the brakes or know why they are engaged. The potential consequences are quite easy to imagine.

B. Running attacks on the UGV

All attacks on Tor were performed from a standalone computer connected directly to the internal network and set up as a ROS node, except for DoS attacks that were performed directly at the UDP level.

a) DoS: In the DoS attacks, the compromised component is assumed to have privileged access. The attacks using UDP packets targeted the assumed weakest link in the system according to bandwidth and throughput, which is the radio connection. The attacks targeted the ports used to transfer telemetry data and the port used for the video feed. Flooding the network with UDP packets successfully interrupted the connection between the Milrem Controller Vest and Tor. There were immediate effects when the attacks started. The video feed lost the packages and froze on the last video frame, and the telemetry data was lost. When the attack was stopped, both components regained connection. Through experiments with different rates of UDP packets, a minor effect was observed with a transmission speed of 1 MB/s, which increased video latency to around 8ms. Loss of connection was achieved when latency reached 1,700 ms with a speed of 18 MB/s. The UDP packet size was also an important factor in the effect of the attack, with smaller packages being more effective. The experiments also resulted in some unexpected and previously unseen side effects on Tor, requiring a restart of the system.

Flooding the traffic with ROS 2 messages did not seem to have any effect on the controllability of the UGV, as in the tests run in the simulator. The `light` topic was targeted and Tor's lights were triggered and turned off and on during the attack - confirming the successful arrival and processing of the messages, but no latency or disturbances were registered on the network or other topics. We could not find any obvious reason for that.

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/MILCOM55135.2022.10017692

b) ROS 2 topic hijacking: The same method and topics used on the simulator were used to target Tor. The attacks were against the generator, emergency stop, and force brakes. Sending fake messages to the generator topic with 10 messages per second resulted in the generator being immediately powered down, and a series of relays were triggered. After a couple of seconds, the generator restarted and stopped shortly after. These actions repeated themselves until the attack was stopped. The attack was stopped before it was in danger of physically damaging the generator or the relays.

Targeting the emergency stop feature was done with approximately 11 messages per second. The attack caused Themis Controller to stop sending commands to Themis PC, and the control over Tor was lost. It was then possible to send the `IDLE` command from the XBOX controller, causing the XBOX controller mode to restart the operation. This attack successfully stopped Tor immediately and requires a specific action to regain normal operating mode.

When the attack against the force brakes was executed, Tor immediately stopped. The messages were published at a rate of 50 messages per second, which is higher than the rate of normal traffic. If the electric motors try to drive Tor with the force brakes applied, physical damage may occur to the motor or brakes. This hypothesis however was tested and refuted. It was confirmed that all drive commands sent to the Milrem software were ignored and not sent to the motors, so the motors do not attempt to drive when the brakes are engaged. However, the brakes could still be disengaged using the same attack method, which could lead to uncontrolled movement.

In conclusion, most attacks tested were successful, albeit reasonably straightforward. The experiments showed that some safety mechanisms were indeed built within the UGV internal control systems, as proved by the emergency stop attack. Still, more advanced attacks might probably circumvent these restrictions, if sufficiently refined.

V. Towards data-driven detection capabilities

So far we have focused on building the foundation for data-driven cyber defence capabilities by addressing threats and attacks that should be detected. Here, we briefly show feasibility of a data-driven detection mechanism by training two proof-of-concept machine learning (ML) based detection models. Note that a consequence of the threat model is that even if security mechanisms are used to encrypt ROS 2 traffic and restrict access to topics for some nodes, they will have limited effect on the compromised node[3].

We captured (internal) network traffic on Tor during attack simulation, including both benign and attack traffic. As we were in control of the attacks, we could also label each data packet as malicious (317K data packets) or benign (18K data packets). In a first experiment, we used 70% of the captured data to train a simple classifier using a decision tree. The model used features from the UDP and RTPS protocols[4] and achieved a false-negative rate of less than 1% and false-positive rate of around 5% on the test set (remaining 30%). Using exclusively benign traffic we also trained a simple anomaly detector which was used to predict the target/path of a packet. If the predicted probability of the actual path taken is below a given threshold then this is considered an anomaly. Here, both k-nearest neighbours, decision tree and random forest achieved an accuracy of 99%.

Note that this was meant to show feasibility and we have therefore not conducted a detailed analysis of the result and traffic, and in particular, of the realism of the benign traffic. We do consider it likely that more realistic traffic will considerable weaken the result and building a more robust ML-based detection capability remains future work. Nevertheless, these results show promise of our approach.

[3]See §VI for details.

[4]The features were predominantly categorical using a one-hot encoding.

VI. Limitations & related work

The work presented here contains some initial results on our journey towards novel data-driven cyber defence capabilities for UGVs, with a considerable amount of work yet to be done. We see in particular two clear limitations.

(1) We have not addressed how a component was compromised to begin with. Neither have we considered other well-known attack vectors that can produce similar results to ours, as they would not bring a much deeper insight in the security issues specific to military UGVs. A complete analysis of how different component types can be exploited by an attacker and of the preconditions needed to make the attacks effective in a real world scenario, has not being systematically performed either.

(2) A more serious limitation is that the UGV used in the attacks (§IV) does not utilise the DDS security mechanism supported by ROS 2 [8]. DDS would have hindered components from entering the system and registering as publishers or subscribers to topics, providing authentication mechanisms, access control and encryption of messages.

The reason for omitting DDS in this work was purely pragmatic as the access restrictions on the Milrem's software did not allow us to activate DDS. However, the attacks described in this paper have been carefully crafted to evade DDS's security mechanisms as far as possible. The rightmost column of table I indicates which attacks we believe will evade DDS, where ✓ indicates that DDS will have no impact while (✓) indicates that DDS may have some very limited impact. One example of this is that trusted components will already have the necessary access and keys to interpret and publish messages.

However, this will still need to be validated through practical experiments. It is also worth noting that real-time performance requirements can limit the use of DDS as some critical topics might not accept the possible time delay to encrypt and authenticate a message [9]. This delay is especially critical with real-time systems like Tor, as we showed with our tests.

Much of the related work looking into the security of unmanned and autonomous vehicles typically has self-driving cars or drones as the object of their study [10], [11], but they

Dette er en postprint-versjon/This is a postprint version.
DOI til publisert versjon/DOI to published version: 10.1109/MILCOM55135.2022.10017692

assume a different threat model and assets. For instance, many attack vectors in cars are linked to the passengers, workshops repairs and internet access, all of which are not very relevant in a military setting. Other surveys identify a wide range of security challenges for cyber-physical systems in general, but not all are relevant for unmanned vehicles. One thing that most seem to agree upon, however, and in line with our work, is the importance of developing more adequate detection and response capabilities for these systems [12].

As for research on ROS security, McClean et al. [13] addressed vulnerabilities in cyber-physical systems built with ROS that can be exploited by our attacks, but the work is not directly comparable to our approach. Michaud et al. [14] addresses vulnerabilities in DDS, but deviates from our work by focusing on real-time innovations. There is also a military version of ROS [15], but it focuses mainly on source code integrity with strong reliance on DDS.

Given that we are pursuing a ML-based data-driven approach for intrusion detection, there is a vast amount of literature which could be relevant, but as we focus on the generation of training and validation data by simulating and applying attacks (§II) rather than the ML algorithms themselves, we will not review it further here[5]. There are also other possible ways to improve security of UGVs like white or black listing, but we see them as little effective in our threat model where we assume a compromised trusted component. Signature-based detection could also detect some attacks, but will only be possible for existing attacks and, as discussed in §I, this area is lacking a catalogue of attacks and consequently, signatures of such attacks.

VII. Conclusion and future work

In this paper, we have laid the foundation for, and commenced the development of, autonomous cyber defence capabilities for unmanned vehicles with autonomous capabilities. Key contributions are: (1) an initial catalogue of relevant threats and attacks for this domain that complements the available threat catalogues for other types of cyber systems and upon which an autonomous cyber defence capabilities can be built (§III); (2) practical realism of selected attacks by applying them on an actual military vehicle (§IV); (3) feasibility of data-driven ML-based detection capabilities (§V).

Next, limitation discussed in §VI will need to be addressed, and more attacks and more realistic normal traffic will need to be generated to improve detection capabilities.

The most challenging part of our end goal is most likely to identify and apply suitable courses of actions (CoA)[6] in order to respond to detected malicious activity. Here, the detected activity will need to be analysed to form situational awareness and overall mission goals and underlying risk assessment will need to be taken into account when selected suitable CoAs. Reinforcement learning has shown some promise here [19], [20], but the work is still in its infancy.

[5]See e.g. [16]–[18].

[6]E.g. disable a sensor or component not longer trusted, ignoring certain messages, do nothing or even abort mission.

Acknowledgment

The authors would like to thank the FFI researchers Eilert André Mentzoni, and Niels Hygum Nielsen for their assistance with Tor.

References

[1] F. Mancini, B. Greve, S. Bruvoll, and J. H. Wiik, "A threat model and security capabilities for autonomous military vehicles – exploring the challenges of designing and integrating security," FFI report, (UO) 21/00428, 2021.

[2] F. Mancini *et al.*, "A security reference model for autonomous vehicles in military operations," in *2020 IEEE Conference on Communications and Network Security (CNS)*, 2020.

[3] K. Mathiassen, M. Baksaas, S. A. Græe, E. A. Mentzoni, and N. H. Nielsen, "Making the milrem themis ugv ready for autonomous operations," in *Unmanned Systems Technology XXIII*, vol. 11758. SPIE, 2021, pp. 221–240.

[4] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, 2022.

[5] A. Koubaa, Ed., *Robot Operating System (ROS)*. Springer Cham, 2016.

[6] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[7] S. Sandoval and P. Thulasiraman, "Cyber security assessment of the robot operating system 2 for aerial networks," in *2019 IEEE International Systems Conference (SysCon)*. IEEE, 2019, pp. 1–8.

[8] (2018) Dds security, an omg® dds security™ publication. [Online]. Available: https://www.omg.org/spec/DDS-SECURITY/1.1/PDF

[9] J. Kim, J. M. Smereka, C. Cheung, S. Nepal, and M. Grobler, "Security and performance considerations in ros 2: A balancing act," *arXiv preprint arXiv:1809.09566*, 2018.

[10] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *20th USENIX Security Symposium (USENIX Security 11)*. San Francisco, CA: USENIX Association, Aug. 2011.

[11] N. M. Rodday, R. d. O. Schmidt, and A. Pras, "Exploring security vulnerabilities of unmanned aerial vehicles," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016, pp. 993–994.

[12] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[13] J. McClean, C. Stull, C. Farrar, and D. Mascarenas, "A preliminary cyber-physical security assessment of the robot operating system (ros)," in *Unmanned Systems Technology XV*, vol. 8741. International Society for Optics and Photonics, 2013, p. 874110.

[14] M. J. Michaud, T. Dean, and S. P. Leblanc, "Attacking omg data distribution service (dds) based real-time mission critical distributed systems," in *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*. IEEE, 2018, pp. 68–77.

[15] J. Towler and M. Bries, "Ros military: Progress and promise," in *Proc. 2018 Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, 2018.

[16] N. F. Haq, A. R. Onik, M. A. K. Hridoy, M. Rafni, F. M. Shah, and D. M. Farid, "Application of machine learning approaches in intrusion detection system: a survey," *IJARAI-International Journal of Advanced Research in Artificial Intelligence*, vol. 4, no. 3, pp. 9–18, 2015.

[17] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[18] J. A. Khan and N. Jain, "A survey on intrusion detection systems and classification techniques," *Int J Sci Res Sci Eng Technol*, vol. 2, no. 5, pp. 202–208, 2016.

[19] A. Ridley, "Machine learning for autonomous cyber defense," *The Next Wave*, vol. 22, no. 1, pp. 7–14, 2018.

[20] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

DOI til publisert versjon/DOI to published version: 10.1109/MILCOM55135.2022.10017692