Radboud University Nijmegen

Faculty of Science

Attack Vectors and Techniques for Unmanned Aerial Systems

Research Internship Report

Max Leijtens
Radboud University
Nijmegen, The Netherlands
m.leijtens@student.ru.nl

Privasec | DRONESEC

First Supervisor:
Mike Monnik
DroneSec
Melbourne, Australia
mike.m@dronesec.com

Second Supervisor:
Veelasha Moonsamy
Radboud University
Nijmegen, The Netherlands
email@veelasha.org

August 2020

Abstract

This report describes a framework upon which a MITRE ATT&CK® matrix for Unmanned Aerial Systems (UAS) can be built. The framework describes the actions an adversary may take while operating within a UAS or UTM system. These actions range from discovering a system to impacting operational processes. A matrix built on this framework could be used to better characterize and describe (post-compromise) adversary behaviour. By being able to predict or track adversarial attack vectors, it becomes easier to protect unmanned aerial systems against malicious actors.

Since UAS operate as traditional computer systems, they are vulnerable to the attack vectors impacting traditional IT systems. However, their physical capabilities also leave them vulnerable for several physical attack vectors, that are normally not a realistic threat for computer systems. Because UAS are not entirely covered by the Enterprise ATT&CK matrix, nor by the Industrial Control Systems (ICS) matrix, a new matrix is required to properly address the threats that an Unmanned Aerial System faces.

Acknowledgements

I wish to thank Veelasha Moonsamy for supervising my research, as well as for bringing me in touch with Mike Monnik from DroneSec. Without these two things my intercontinental research internship would not have been possible.

Furthermore, I wish to thank Privasec and DroneSec for taking on the challenge of having an international intern and allowing me to do this research.

A huge thank you to Arison Neo and Mike Monnik from DroneSec for their elaborate and constructive comments on every new development of the report. Despite the short notices, sharp deadlines, and an endless stream of question, they never failed to give me constructive criticism and mental support. I could not have written this report without their guidance and supervision.

Lastly, a thank you to Ben Baron from URSA Secure for being available for the weekly calls outside of office hours. Thank you for your advice on the first steps of developing a framework.

Contents

Abstract **1**

Acknowledgements **2**

1 Introduction **6**
1.1 MITRE ATT&CK matrices 6
1.2 UAS security 6
1.3 New matrix for unmanned aerial systems 7
1.4 Motivation for this research 7
1.5 Disclaimer 8
1.6 Document outline 9

2 Justification and methodology **10**
2.1 Justification for using two existing matrices 10
2.2 Attack selection method 10
2.2.1 Method 10
2.2.2 Assumptions on UAS 11
2.2.3 Description of UTM systems 12
2.2.4 Assumptions on UTM systems 13

3 Threat modelling **14**
3.1 Trivial 15
3.2 Informed 15
3.3 Sophisticated 15

4 ATT&CK framework **16**
4.1 Initial Access 16
4.1.1 UAS and UTM specific techniques 16
4.1.2 Existing vectors in the Enterprise matrix 17
4.1.3 Existing vectors in the ICS matrix 18
4.2 Execution 18
4.2.1 UAS and UTM specific techniques 18
4.2.2 Existing vectors in the Enterprise matrix 19
4.2.3 Existing vectors in the ICS matrix 19
4.3 Persistence 20
4.3.1 UAS and UTM specific techniques 20
4.3.2 Existing vectors in the Enterprise matrix 20
4.3.3 Existing vectors in the ICS matrix 22

4.4 Privilege escalation 23
4.4.1 UAS and UTM specific techniques 23
4.4.2 Existing vectors in the Enterprise matrix 23
4.4.3 Existing vectors in the ICS matrix 24
4.5 Defence evasion 24
4.5.1 UAS and UTM specific techniques 24
4.5.2 Existing vectors in the Enterprise matrix 25
4.5.3 Existing vectors in the ICS matrix 28
4.6 Credential access 28
4.6.1 UAS and UTM specific techniques 28
4.6.2 Existing vectors in the Enterprise matrix 29
4.6.3 Existing vectors in the ICS matrix 30
4.7 Discovery 30
4.7.1 UAS and UTM specific techniques 30
4.7.2 Existing vectors in the Enterprise matrix 31
4.7.3 Existing vectors in the ICS matrix 32
4.8 Lateral movement 33
4.8.1 UAS and UTM specific techniques 33
4.8.2 Existing vectors in the Enterprise matrix 33
4.8.3 Existing vectors in the ICS matrix 34
4.9 Collection 34
4.9.1 UAS and UTM specific techniques 34
4.9.2 Existing vectors in the Enterprise matrix 35
4.9.3 Existing vectors in the ICS matrix 36
4.10 Command and Control (C2) 36
4.10.1 UAS and UTM specific techniques 36
4.10.2 Existing vectors in the Enterprise matrix 36
4.10.3 Existing vectors in the ICS matrix 39
4.11 Exfiltration 39
4.11.1 UAS and UTM specific techniques 39
4.11.2 Existing vectors in the Enterprise matrix 39
4.11.3 Existing vectors in the ICS matrix 41
4.12 Inhibit response function 41
4.12.1 UAS and UTM specific techniques 41
4.12.2 Existing vectors in the Enterprise matrix 41
4.12.3 Existing vectors in the ICS matrix 42
4.13 Impair process control 43
4.13.1 UAS and UTM specific techniques 44
4.13.2 Existing vectors in the Enterprise matrix 44
4.13.3 Existing vectors in the ICS matrix 44
4.14 Impact 46
4.14.1 UAS and UTM specific techniques 46
4.14.2 Existing vectors in the Enterprise matrix 46
4.14.3 Existing vectors in the ICS matrix 48

5 Future work **52**
5.1 More peer reviews 52
5.2 Rewording of technique descriptions 52
5.3 Examples and proofs 52
5.4 Mitigations 52
5.5 Labelling of techniques 53
5.6 Continuous development 53

Glossary **54**

Acronyms **56**

Bibliography **59**

Chapter 1

Introduction

Unmanned Aerial Systems (UAS) are becoming more common, sophisticated and more readily available. Thanks to new technological advancements, they are now able to behave and collect data autonomously. This allows them to be used in more versatile ways for the individual, but also for businesses and law enforcement [1]. Unfortunately, their rise in popularity also makes them a more valuable target for malicious actors that could use them as new avenues for cyber attacks.

Security rarely is a priority when developing new technology and evidence shows that this has also not been the case for UAS [2]. One of the reasons for that is the fact that developing security in products is expensive and often does not have a visible pay-off. To make the threats that UAS are facing more visible, as well as making it easier (and thus cheaper) to mitigate these threats, a MITRE ATT&CK® matrix could be used.

1.1 MITRE ATT&CK matrices

This report describes a framework for a MITRE Adversarial Tactics, Techniques, and Common Knowledge (in short: ATT&CK) matrix for unmanned aerial systems. Such a matrix is a knowledge base that contains adversarial tactics and techniques on a specific set of systems. In this case, the set of systems are UAS, also known as drones, Unmanned Aerial Vehicle (UAV), or Remotely Piloted Aircraft System (RPAS).

This knowledge base can be used to map, track, and plan for adversarial behaviour. By showing the options available to an adversary, preemptive measures can be taken in order to limit their usage. Furthermore, it can be used to build behaviour based adversary emulators, which can be used to automatically test the security of the systems that are covered in a specific matrix.

Three of these ATT&CK matrices already exist: one for Enterprise systems, one for Industrial Control Systems (ICS), and one for Mobile systems [3].

1.2 UAS security

UAS have risen in popularity in the last few years for their wide range of possible applications. As the small flying computers they are, they can be used in a

variety of ways. For instance in the commercial sector, they are used for their ability to make incredible shots from hard to reach places or to deliver goods with a minimal requirement for human interaction. In the agricultural sector, they have become popular both for the monitoring of livestock, as well as for their ability to provide simple care for various types of crops [4]. They are now also employed by law enforcement to increase the range of unmanned surveillance and to allow them to effortlessly track criminals in difficult types of terrain [5, 6].

Unfortunately, security is rarely a priority when new technology is developed. Development on UAS has not been any different in this aspect. Although larger manufacturers continuously attempt to improve their security, still a large amount of UAS currently in operation have known vulnerabilities [2].

Several of these vulnerabilities can only be fixed by making hardware changes or by significantly changing their operating system. This means that these problems can only be corrected in newer models of certain UAS, which shows the importance of proper security testing before new UAS are put on the market.

Manufacturers often do not see the pay-off in the ever-present security versus innovation dilemma, which causes these preventable issues to keep appearing. Because there is not yet a standardised way in which these systems can be tested, security tests are expensive to perform. This makes them a less interesting option for manufacturers.

An ATT&CK matrix would allow for emulation of adversarial behaviour, which could be used to systematically test new UAS on known vulnerabilities. This would significantly reduce the cost of security tests and in turn improve the level of security for the usage of UAS in the entire field.

1.3 New matrix for unmanned aerial systems

Since UAS operate as a traditional computer system (e.g. Linux, Android), they are vulnerable to the attack vectors impacting traditional IT systems. However, their physical capabilities also leave them vulnerable for several physical attack vectors, that are normally not a realistic threat for computer systems. Because unmanned aerial systems are not entirely covered by the Enterprise ATT&CK matrix, nor by the ICS one, a new matrix is required to properly cover the threats that Unmanned Aerial Systems face.

1.4 Motivation for this research

The research in this report is done during a three month internship at the company DroneSec. The following paragraphs in this section are written by Mike Monnik, CTO of DroneSec, explaining the company's motivation in setting up and supporting this research.

DroneSec has constructed a research project based on internal requirements as part of a study component for Radboud Univeristy. The project outcome is an 'ATT&CK framework' for unmanned systems based on the structure provided by MITRE Corporation. The project was undertaken due to the perceived importance, global adoption and national security applications of drones (UAV/UAS/RPAS) within society. The project will be made available to the

public and aid in threat intelligence, digital security and program management capabilities to organisations utilising drones for a myriad of applications. This research will form a baseline industry standard for identifying and protecting against a variety of risks to drones, counter-drone and UAS Traffic Management (UTM) systems.

DroneSec is a UAS/Cyber Security firm based in Melbourne, Australia, with offices in Sydney and Singapore. Its parent company is Privasec, who brought in DroneSec to extend its security operations for the unmanned systems industry in 2018. Through its work on protecting friendly drones and against rogue drones, DroneSec has identified key attack vectors that could compromise the Confidentiality, Integrity and Availability of unmanned systems. Drones are cyber-kinetic systems and share fundamental similarities with computers, networks and digital systems. As a result, some attacks against them are repeatable and preventable. Similarly, DroneSec would like to identify if there are any other potential or future applications for attacks not currently known or with little research. This framework aims to map out these attacks and describe the actions an adversary may take while attempting to compromise a drone (or its interconnected UTM system). This helps characterise and describe post-compromise adversary behaviour, providing the defending team with defensive playbooks and guidelines.

The sole focus of this research was to protect unmanned system operations from disruption, reduce the potential opportunity for attackers and decrease the risk of reactive restrictions being placed on the innovation of the emerging drone industry. Other benefits to DroneSec include calibrating their Threat Intelligence assumptions, uplifting the skill of their Red Team operators and identifying requirements within their training courses. DroneSec has a partnership with URSA Secure which provided assistance to this project and the researcher.

The sole focus of this research was to protect unmanned system operations from disruption, reduce the potential opportunity for attackers and decrease the risk of reactive restrictions being placed on the innovation of the emerging drone industry. Other benefits to DroneSec include calibrating their Threat Intelligence assumptions, uplifting the skill of their Red Team operators and identifying requirements within their training courses. Additionally, the Privasec team will benefit from the project in their threat intelligence and red team operations as part of a growing risk around the control of both friendly and rogue IoT-connected systems. DroneSec has a partnership with URSA Secure which provided assistance to this project and the researcher.

1.5 Disclaimer

Instead of a complete ATT&CK matrix, this report merely describes a framework for one. ATT&CK matrices are vast projects and not able to be completed within the limited timespan of the three months that were available for this project. More work is required to turn this framework into an ATT&CK matrix. See chapter 5.

It is also important to note that neither the research, nor the researcher are affiliated to MITRE. This research merely uses the ideas of MITRE's previously developed ATT&CK matrices.

1.6 Document outline

Chapter 2: Justification and methodology. This chapter explains why techniques are taken from two already existing matrices. It also explains the method of how techniques were reviewed before they ended up in this framework.

Chapter 3: Threat modelling. This chapter describes the different adversaries that exist for UAS and how I have come to these specific threat models. It describes among other things their capabilities, skill level, and time available for attacks.

Chapter 4: ATT&CK framework. This chapter contains the framework itself, as resulted from the in chapter 2 described method.

Chapter 5: Future work. This chapter describes what future work still needs to be done on the framework before it can be published as a complete matrix.

Chapter 2

Justification and methodology

2.1 Justification for using two existing matrices

Since UAS often function as independent computer systems, many attack vectors on them are covered in the ATT&CK Enterprise matrix. However, because they also have a physical presence that can affect and be affected by the environments in which they operate, not all attack vectors are covered in this framework. Therefore this report also includes attacks from the ICS matrix, that more extensively lists physical attack vectors.

UTM systems function as an ICS when controlling UAS, yet are at the same time connected to an enterprise network. Therefore I believed that to cover all attack vectors on these systems, it was required to look at both matrices as well.

2.2 Attack selection method

In this section I describe what methodology I applied and which assumptions I used to create the Unmanned Aerial Systems (UAS) framework you find in chapter 4.

2.2.1 Method

For lack of scientific research on this topic, our initial adversarial tactics and techniques were found using news and incident reports on UAS and misbehaviour by rogue UAS operators. To these tactics I added new techniques using the scientific literature that was available at the time.

After initial attack vectors were collected, I drafted generalised assumptions about UAS and UTM systems so that I could later use these to select which attacks and tactics were viable. I tested these assumptions on the attack vectors that I had already found and adapted them to fit these. Afterwards I had them reviewed by independent UAS security experts to verify that these were accurate and could be used to make a pre-selection on the attacks in the existing matrices. You can find these assumptions in section 2.2.2.

Using these assumptions, I analysed both the Enterprise and the ICS matrix and made a selection of what I presumed were viable attacks and added these to our already existing set of attack vectors, which I then ordered into categories. Finally, I had them verified on both validity and likelihood by UAS security experts.

This method resulted in the unmanned aerial systems framework you find in chapter 4.

This work was peer reviewed by two leading organisations that specialise in UAS security. The primary reviewer and requestee of this work was DroneSec, a UAS-Cyber Security and Threat Intelligence firm based in Melbourne, Australia. The secondary reviewer and program partner was URSA Secure, a UAS Forensics and Security firm based in Washington, New York. Individually, each organisation has over 5 years experience in the specialised field of unmanned security and domain experience within cyber security.

2.2.2 Assumptions on UAS

Unmanned Aerial System (UAS) are, as the name suggests airborne computer systems without an onboard pilot. Usually they are flown by a remote pilot, but can also be autonomously controlled by the onboard computer. Their size varies from that of a coin (2 cm) to that of a full-sized airplane (40m).
Furthermore a UAS consists of the following [7, 8]:

- **Computer system:** This is a fully operational stand-alone computer system with its own processor, RAM and storage. This is the main controller of the systems on the UAS. I have assumed that these computer systems use Linux as their OS. This operating system runs software that controls what connections it allows, at what speed which motor should spin, what flight path to maintain, etc..
- **Propulsion:** Commercial UAS are usually propelled by one or more battery-powered motors, attached to propellers. These motors are controlled by the onboard computer. Larger UAS (common in military), can also be propelled by jet-engines, which are usually fuel-powered. In this case, the fuel also has to be carried on the UAS.
- **Battery:** A rechargeable battery to power the computer system and sensors on the UAS. The battery can be recharged in-flight if the UAS propulsion is done with non-electrical fuel. Otherwise it can be recharged at a charging station (usually its base station) or by removing, then charging, then replacing the battery.
- **Radio antenna:** The antenna intercepts radio waves and forwards them to the radio receiver. The antenna can also be used to transmit radio waves.
- **Radio receiver and transmitter:** This device receives radio waves and converts them into a form usable by the computer system. It can also convert computer data to radio waves that are then sent by the controller.

If a UAS is controlled remotely, the pilot has a controller with a transmitter attached to it. This can send instructions from the controller to the UAS. Since they are sent as radio waves, they can be received by anyone with an antenna

and therefore they have the possibility to be encoded as to only allow specific senders to be communicating with specific UAS.

Additionally I have also assumed that UAS come equipped with the following sensors [8, 9]:

- **Camera(s):** One or more camera's are attached to the UAS in order to record its surroundings. This recording is stored on a separate storage than the computer system's and is attached to a camera controller that has its own antenna and receiver and is able to transmit its recordings separately from the
- **Gyroscope:** A gyroscope is used to determine the rate of rotation, degree of tilt, and angular velocity of the UAS. It is the main tool used to measure and maintain the UAS's orientation. Multiple gyroscopes are in use to measure this tilt over more than one axis.
- **Accelerometer:** Measures the acceleration of the UAS over a single axis. Multiple accelerometers are in use on one system to measure multi-directional acceleration.
- **GPS/GLONASS/Magnetometer:** At least one of these systems is installed in UAS in order to allow both the computer and the controller to know the location of the system. This is especially important in UAS that are not pilot-controlled.
- **(Omni)Directional distance sensors:** UAS have ultrasonic and/or infrared sensors surrounding it, which are used to measure the distance to objects in their surroundings.
- **Barometer:** Since directional distance sensors have a limited range, UAS are equipped with a barometer to measure its altitude at greater heights.

2.2.3 Description of UTM systems

A UAS Traffic Management (UTM) system provides safe and effective control for UAS to operate in mixed airspace Beyond Visual Line-of-Sight (BVLOS) with other low-altitude aircraft. It is an air traffic management system which can manage and control a variety of brands, models, and types of unmanned aerial systems. The focus of UTM systems is the digital sharing of unmanned planned flights, and the real-time change of those flight paths if required for safety or collision prevention purposes.

UTM systems are also referred to as Urban Air Mobility (UAM) systems as they aim to support flight operations in complex urban environments. These include populated areas, city landscapes and close-proximity to power lines, meaning more obstacles to avoid, less or no visual line of sight, increased radio signal interference and changing landing positions.[10]

Not many UTM systems are currently in operation, yet they are included in this report as they are part of the ecosystem of development for UAS by the current global aviation industries. UTM systems are meant to be complementary to existing manned air traffic management systems which are already in operations for managing manned flights worldwide.

2.2.4 Assumptions on UTM systems

Using the description in section 2.2.3 I have come to the following assumptions on UTM systems.

A UTM system consists of the following parts:

- **Management server:** This is the traffic management server of the system. All data is gathered and processed here. This server also manages alterations to flight paths of UAS in its range. This server can be accessed by a local or remote desktop environment.
- **Sensors:** The system has some form of UAS detection system. This consists of radar equipment, radio frequency interceptor, and an optical tracking system, equipped with both normal and infrared light camera's. The limit of these sensors is usually the limit of the range in which a UTM system can operate.
- **Radio communication:** The system has a way to communicate over Radio Frequency (RF) with UAS within range. Usually this happens with RF and Software Defined Radio (SDR) tools, antenna's, signal extenders.
- **Counter measures:** For special cases in which an unauthorised UAS is breaking the UTM's boundaries and is not responding to radio cues that it should move out of this area, the UTM system is equipped with hardware that can partially or fully disable a UAS. Not all UTM systems come equipped with these measures, but I have assumed that they do for the development of the framework.

Such a system has an operating range of up to 10km.

This range can be extended by attaching modules to the UTM system. These UTM modules consist of the above **Sensors**, **Radio communication** and **Counter measures**. However, the server operating on these modules just processes UAS and flight data, which it sends to the main server. In return it receives commands from the main management server on how to handle the local traffic. Using a modular system would increase the operating range to about 100km.

Chapter 3

Threat modelling

In this chapter I describe the three levels of threat actors that currently exist.

These levels can be used to describe the imminence of certain techniques to be used. For every actor I describe resources at their disposal, among others, time, equipment and skill. Since equipment might become cheaper or more readily available in the future, these actor capabilities might change over time. Therefore I have also included an estimated budget for each of these actors, so that they can be adjusted accordingly.

The reason why I have included these threat models is that, when the UAS environment and thus the threat models change, this makes it easier to notice where and how it has changed. By knowing this, the threat model can be changed, and the framework updated accordingly. These models function as a baseline of current threats during the making of this version of the framework.

The creation of these threat actor models has been aided by DroneSec's incident database, which contains observations and documents of over 1,000 unique UAS incidents [11].

Category explanation

Here the categories used in the below threat models are explained.

- **Attacker:** A description of the attacker that is expected in this category.
- **Skill level:** The perceived level of technical skill the attacker has.
- **Work Hours:** A description of the amount of work hours available to the adversary. When multiple attackers are operating in a group, this amount is split among them.
- **Equipment:** A description of the equipment available to the threat actor.
- **UAS Capacity:** The amount, supply availability, and quality of UAS the adversary can afford. This/these UAS can be used to find and test vulnerabilities and exploits on.
- **Presence:** The range in which a threat actor is able to operate. This range is based on technologies also listed here.
- **Total Cost:** This describes an estimation of the total budget available for tactics used by this threat. This includes the cost of work hours, equipment and UAS capacity.

3.1 Trivial

- **Attacker:** A hobbyist modder, local disruptor, or individual hacker.
- **Skill level:** Medium; this adversary is moderately skilled, can write simple scripts, and is able to use public exploit frameworks and tools.
- **Work Hours:** Up to 10 hours.
- **Equipment:** A laptop with wireless dongle. (∼$1,500)
- **UAS Capacity:** A single, medium priced UAS (∼$500). This UAS cannot be modified or exploited in a way that might permanently damage it.
- **Presence:** Local (within WiFi range), or directly accessible from the internet.
- **Total Cost:** Up to $3,000.

3.2 Informed

- **Attacker:** Hacktivists, organised criminals, or hacking groups.
- **Skill level:** High; able to find and use undisclosed or unpublished exploits and combine multiple attack vectors.
- **Work Hours:** Up to 100 hours.
- **Equipment:** Multiple servers and high-end laptops. Radio Frequency (RF) and Software Defined Radio (SDR) tools, antenna's, signal extenders and dedicated software (e.g. HackRF One with GQRX, Yagi). UAS 0-days. (∼$5,000)
- **UAS Capacity:** Multiple high-end UAS (total value of $7,500) that can be exploited and modified in ways that might permanently damage or disable them.
- **Presence:** National. Anything accessible from the internet. Anything within range of antenna's and signal extenders (5-10km). Physical through rogue intrusion.
- **Total Cost:** Up to $20,000.

3.3 Sophisticated

- **Attacker:** Nation States, Terrorist Organisations, or Advanced Persistent Threats (APTs).
- **Skill level:** High; able to find and use undisclosed or unpublished exploits and combine multiple attack vectors. Able to develop specialised hardware to assist in the attack.
- **Work Hours:** Up to 1000 hours.
- **Equipment:** All equipment of the Informed threat. GPS/GLONASS jammers/spoofers, Protocol Manipulation Equipment, 0-day communication-protocol exploits. 0-day UTM exploits.
- **UAS Capacity:** Virtually unlimited high-end UAS and a private UTM-system.
- **Presence:** Globally. Is able to perform advanced rogue intrusions and can deploy operators worldwide.
- **Total Cost:** $100,000 and over.

Chapter 4

ATT&CK framework

This chapter contains the framework which is the result of the method explained in chapter 2. In it you will find techniques that can be found in both the ATT&CK Matrix for Enterprise [12], the ATT&CK Matrix for Industrial Control Systems [13], as well as new, UAS-specific techniques. These techniques have been ordered by tactic in the same way they are ordered in the currently existing matrices.

4.1 Initial Access

The adversary is trying to get an initial foothold within a UAS.

Initial Access consists of techniques that use various entry vectors to gain their initial foothold within a UAS. UTM systems are excluded here as these can be viewed as servers or networks, to which thus the original MITRE ATT&CK matrix already applies.

4.1.1 UAS and UTM specific techniques

Personal Identifiable Information Gathering Knowing personal information about the operator could assist in acquiring more information about the operator, which could then be used for other Initial Access tactics.

Some UAS have a unique ID (also called a UAS ID or a remote ID), which can be linked to the system's owner (due to UAS registration laws). Most UAS that have a remote ID are commercial aerial systems; custom or racing UAS often lack this ID. The identifiable information could also be garnered from other sources like wireless output from UAS, Automatic Dependent SurveillanceBroadcast (Automatic Dependent SurveillanceBroadcast) and visual identification.

Visual Identification By visually identifying the UAS, it's brand, model, or type might be found, which could be used to find known vulnerabilities to the UAS.

Signal and Protocol Analysis Analysing the underlying communication signal and protocols used in the RF band between the UAS and its controller

utilising SDR tooling in order to identify what UAS brand, model, or type is used. which could be used to find known vulnerabilities.

4.1.2 Existing vectors in the Enterprise matrix

Drive-by Compromise A drive-by compromise is when an adversary gains access to a system through a user visiting a website over the normal course of browsing. With this technique, the user's web browser is typically targeted for exploitation, but adversaries may also use compromised websites for non-exploitation behaviour such as acquiring application access tokens.

Exploit Public-Facing Application The use of software, data, or commands to take advantage of a weakness in an Internet-facing computer system or program in order to cause unintended or unanticipated behaviour. The weakness in the system can be a bug, a glitch, or a design vulnerability. These applications are often websites, but can include databases (like SQL), standard services (like SMB or Secure Shell (SSH)), and any other applications with Internet accessible open sockets, such as web servers and related services.

External Remote Services Remote services such as VPNs, Citrix, and other access mechanisms allow users to connect to internal enterprise network resources from external locations.

Hardware Additions Adversaries may introduce computer accessories, computers, or networking hardware into a system or network that can be used as a vector to gain access. While public references of usage by APT groups are scarce, many penetration testers leverage hardware additions for initial access.

Replication Through Removable Media Adversaries may move onto systems, possibly those on disconnected or air-gapped networks, by copying malware to removable media and taking advantage of Autorun features when the media is inserted into a system and executes.

Supply Chain Compromise Supply chain compromise is the manipulation of products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise. This can affect a UAS if the software within was already manipulated prior to delivery.

Trusted Relationship Adversaries may breach or otherwise leverage organizations who have access to intended victims. Access through trusted third party relationship exploits an existing connection that may not be protected or receives less scrutiny than standard mechanisms of gaining access to a network.

For UAS this includes the trust that is often given to UTM systems. By gaining access to a UTM system, this access could abused to in turn access or manipulate the UAS in range.

Valid Accounts Adversaries may steal the credentials of a specific user or service account using Credential Access techniques or capture credentials earlier in their reconnaissance process through social engineering for means of gaining Initial Access.

4.1.3 Existing vectors in the ICS matrix

Control Device Identification Adversaries may perform control device identification to determine the make and model of a target device. Video and control streams, wireless packets, RF signals and visual identification may be utilized by the adversary to gain this information. By identifying and obtaining device specifics, the adversary may be able to determine device vulnerabilities. This device information can also be used to understand device functionality and inform the decision to target the environment.

Internet Accessible Device Adversaries may gain access into industrial environments directly through systems exposed to the internet for remote access rather than through External Remote Services. Minimal protections provided by these devices such as password authentication may be targeted and compromised.

Wireless Compromise Adversaries may perform wireless compromise as a method of gaining communications and unauthorised access to a wireless network. Access to a wireless network may be gained through the compromise of a wireless device. Adversaries may also utilize radios and other wireless communication devices on the same frequency as the wireless network. Wireless compromise can be done as an initial access vector from a remote distance.

Default Credentials Adversaries may leverage manufacturer or supplier set default credentials on control system devices. These default credentials may have administrative permissions and may be necessary for initial configuration of the device. It is general best practice to change the passwords for these accounts as soon as possible, but some manufacturers may have devices that have passwords or usernames that cannot be changed.

4.2 Execution

The adversary is trying to run the code on a UAS that is already inflicted with the malicious code.

Execution consists of techniques that result in adversary-controlled code running on a remote system of the UAS or a local system on the UTM system. Techniques that run malicious code are often paired with techniques from all other tactics to achieve broader goals, like exploring a UTM network or stealing data collected from the drones.

4.2.1 UAS and UTM specific techniques

No UAS and UTM specific techniques were found for *Execution*. The techniques that were found were already covered in either the Enterprise or the ICS matrix.

4.2.2 Existing vectors in the Enterprise matrix

Execution through API Adversary tools may directly use the application programming interface (API) to execute binaries.

Graphical User Interface The Graphical User Interface (GUI) is a common way to interact with an operating system. Adversaries may use a system's GUI during an operation to search for information and execute files via mouse double-click events or other potentially difficult to monitor interactions.

Local Job Scheduling / Scheduled Task On Linux systems, multiple methods are supported for creating pre-scheduled and periodic background jobs: cron, at, and launchd. Unlike Scheduled Task on Windows systems, job scheduling on Linux-based systems cannot be done remotely unless used in conjunction within an established remote session, like secure shell (SSH).

Scripting Adversaries may use scripts to aid in operations and perform multiple actions that would otherwise be manual. Scripting is useful for speeding up operational tasks and reducing the time required to gain access to critical resources. Some scripting languages may be used to bypass process monitoring mechanisms by directly interacting with the operating system at an API level instead of calling other programs.

Signed Script Proxy Execution Scripts signed with trusted certificates can be used to proxy execution of malicious files. This behaviour may bypass signature validation restrictions and application whitelisting solutions that do not account for use of these scripts.

Third-party Software Third-party applications and software deployment systems may be in use in the network environment for administration purposes (e.g., SCCM, VNC, HBSS, Altiris, etc.). If an adversary gains access to these systems, then they may be able to execute code.

Trap The trap command allows programs and shells to specify commands that will be executed upon receiving interrupt signals. A common situation is a script allowing for graceful termination and handling of common keyboard interrupts like ctrl+c and ctrl+d. Adversaries can use this to register code to be executed when the shell encounters specific interrupts either to gain execution or as a persistence mechanism.

4.2.3 Existing vectors in the ICS matrix

Change System State Adversaries may attempt to change the state of the current system on a control device. System state changes may be used to allow for another program to take over control or be loaded onto the device.

Man in the Middle Adversaries with privileged network access may seek to modify network traffic in real time using Man in the Middle attacks. This type of attack allows the adversary to intercept traffic to and/or from a particular device on the network. If a MitM attack is established, then the adversary has the ability to block, log, modify, or inject traffic into the communication stream. There are several ways to accomplish this attack, but some of the most-common are Address Resolution Protocol (ARP) poisoning and the use of a proxy.

4.3 Persistence

The adversary is trying to maintain their foothold and control of the UAS.

Persistence consists of techniques that adversaries use to keep access within a UAS/UTM across restarts, changed credentials, and other interruptions that could cut off their access. Techniques used for persistence include any access, action, or configuration changes that let them maintain their foothold on systems, such as replacing or hijacking legitimate code or adding startup code.

4.3.1 UAS and UTM specific techniques

No UAS and UTM specific techniques were found for *Persistence*. The techniques that were found were already covered in either the Enterprise or the ICS matrix.

4.3.2 Existing vectors in the Enterprise matrix

Account Manipulation Account manipulation may aid adversaries in maintaining access to credentials and certain permission levels within an environment. Manipulation could consist of modifying permissions, modifying credentials, adding or changing permission groups, modifying account settings, or modifying how authentication is performed. These actions could also include account activity designed to subvert security policies, such as performing iterative password updates to subvert password duration policies and preserve the life of compromised credentials. In order to create or manipulate accounts, the adversary must already have sufficient permissions on systems or the domain.

Bootkit A bootkit is a malware variant that modifies the boot sectors of a hard drive, including the Master Boot Record (MBR) and Volume Boot Record (VBR).

System/Component Firmware Some adversaries may employ sophisticated means to compromise computer components and install malicious firmware that will execute adversary code outside of the operating system and main system firmware or BIOS. Malicious device firmware could provide both a persistent level of access to systems despite potential typical failures to maintain access and hard disk re-images, as well as a way to evade host software-based defences and integrity checks.

External Remote Services Remote services such as VPNs, Citrix, and other access mechanisms allow users to connect to internal UAS and UTM network resources from external locations. There are often remote service gateways that manage connections and credential authentication for these services.

File System Permissions Weakness Processes may automatically execute specific binaries as part of their functionality or to perform other actions. If the permissions on the file system directory containing a target binary, or permissions on the binary itself, are improperly set, then the target binary may be overwritten with another binary using user-level permissions and executed by the original process. If the original process and thread are running under a higher permissions level, then the replaced binary will also execute under higher-level permissions, which could include SYSTEM.

Kernel Modules and Extensions Loadable Kernel Modules (or LKMs) are pieces of code that can be loaded and unloaded into the kernel upon demand. They extend the functionality of the kernel without the need to reboot the system. For example, one type of module is the device driver, which allows the kernel to access hardware connected to the system. When used maliciously, Loadable Kernel Modules (LKMs) can be a type of kernel-mode Rootkit that run with the highest operating system privilege (Ring 0). Adversaries can use loadable kernel modules to covertly persist on a system and evade defences.

Local Job Scheduling On Linux systems, multiple methods are supported for creating pre-scheduled and periodic background jobs: cron, at, and launchd. Unlike Scheduled Task on Windows systems, job scheduling on Linux-based systems cannot be done remotely unless used in conjunction within an established remote session, like secure shell (SSH).

New Service When operating systems boot up, they can start programs or applications called services that perform background system functions.

Port Knocking Port Knocking is a well-established method used by both defenders and adversaries to hide open ports from access. To enable a port, an adversary sends a series of packets with certain characteristics before the port will be opened. Usually this series of packets consists of attempted connections to a predefined sequence of closed ports, but can involve unusual flags, specific strings or other unique characteristics. After the sequence is completed, opening a port is often accomplished by the host based firewall, but could also be implemented by custom software.

Server Software Component Adversaries may abuse legitimate extensible development features of server applications to establish persistent access to systems. Enterprise server applications may include features that allow application developers to write and install software to extend the functionality of the main application. Adversaries may install malicious software components to maliciously extend and abuse server applications.

Setuid and Setgid When the setuid or setgid bits are set on Linux or macOS for an application, this means that the application will run with the privileges of the owning user or group respectively . Normally an application is run in the current users context, regardless of which user or group owns the application. There are instances where programs need to be executed in an elevated context to function properly, but the user running them doesnt need the elevated privileges. Instead of creating an entry in the sudoers file, which must be done by root, any user can specify the setuid or setgid flag to be set for their own applications.

Shortcut Modification Shortcuts or symbolic links are ways of referencing other files or programs that will be opened or executed when the shortcut is clicked or executed by a system startup process. Adversaries could use shortcuts to execute their tools for persistence. They may create a new shortcut as a means of indirection that may use Masquerading to look like a legitimate program. Adversaries could also edit the target path or entirely replace an existing shortcut so their tools will be executed instead of the intended legitimate program.

Systemd Systemd services can be used to establish persistence on a Linux system. The systemd service manager is commonly used for managing background daemon processes (also known as services) and other system resources.

Trap The trap command allows programs and shells to specify commands that will be executed upon receiving interrupt signals. A common situation is a script allowing for graceful termination and handling of common keyboard interrupts like ctrl+c and ctrl+d. Adversaries can use this to register code to be executed when the shell encounters specific interrupts either to gain execution or as a persistence mechanism.

Valid Accounts Adversaries may steal the credentials of a specific user or service account using Credential Access techniques or capture credentials earlier in their reconnaissance process through social engineering for means of gaining Initial Access.

4.3.3 Existing vectors in the ICS matrix

Module Firmware Adversaries may install malicious or vulnerable firmware onto modular hardware devices. Control system devices often contain modular hardware devices. These devices may have their own set of firmware that is separate from the firmware of the main control system equipment.

Program Download Adversaries may perform a program download to load malicious or unintended program logic on a device as a method of persistence or to disrupt response functions or process control. Download does not necessarily has to happen over internet. Could also come from a local resource controlled by an adversary.

Project File Infection Adversaries may attempt to infect project files with malicious code. These project files may consist of objects, program organization units, variables such as tags, documentation, and other configurations needed for PLC programs to function.

4.4 Privilege escalation

The adversary is trying to gain higher-level permissions of the UAS/UTM in order to access data within.

Privilege Escalation consists of techniques that adversaries use to gain higher level permissions on a system or network. Adversaries can often enter and explore a network with unprivileged access but require elevated permissions to follow through on their objectives. Common approaches are to take advantage of system weaknesses, misconfigurations, and vulnerabilities.

4.4.1 UAS and UTM specific techniques

No UAS and UTM specific techniques were found for *Privilege escalation*. The techniques that were found were already covered in either the Enterprise or the ICS matrix.

4.4.2 Existing vectors in the Enterprise matrix

Exploitation for Privilege Escalation Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. Security constructs such as permission levels will often hinder access to information and use of certain techniques, so adversaries will likely need to perform Privilege Escalation to include use of software exploitation to circumvent those restrictions.

File System Permissions Weakness Processes may automatically execute specific binaries as part of their functionality or to perform other actions. If the permissions on the file system directory containing a target binary, or permissions on the binary itself, are improperly set, then the target binary may be overwritten with another binary using user-level permissions and executed by the original process. If the original process and thread are running under a higher permissions level, then the replaced binary will also execute under higher-level permissions, which could include SYSTEM.

New Service When operating systems boot up, they can start programs or applications called services that perform background system functions.

Process Injection Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process.

Local Job Scheduling On Linux systems, multiple methods are supported for creating pre-scheduled and periodic background jobs: cron, at, and launchd. Unlike Scheduled Task on Windows systems, job scheduling on Linux-based systems cannot be done remotely unless used in conjunction within an established remote session, like secure shell (SSH).

Setuid and Setgid When the setuid or setgid bits are set on Linux for an application, this means that the application will run with the privileges of the owning user or group respectively . Normally an application is run in the current users context, regardless of which user or group owns the application.

Sudo The sudoers file, `/etc/sudoers/`, describes which users can run which commands and from which terminals. This also describes which commands users can run as other users or groups. This provides the idea of least privilege such that users are running in their lowest possible permissions for most of the time and only elevate to other users or permissions as needed, typically by prompting for a password. However, the sudoers file can also specify when to not prompt users for passwords with a line like `user1 ALL=(ALL) NOPASSWD: ALL`.

Valid Acounts Adversaries may steal the credentials of a specific user or service account using Credential Access techniques or capture credentials earlier in their reconnaissance process through social engineering for means of gaining Initial Access.

4.4.3 Existing vectors in the ICS matrix

No additional techniques were found for *Privilege escalation* in the ICS matrix. The techniques that were found were already covered in the Enterprise matrix.

4.5 Defence evasion

The adversary is trying to avoid being detected once he has breached the UAS or UTM, allowing him to stay within for a longer period of time.

Defence Evasion consists of techniques that adversaries use to avoid detection throughout their compromise within a UAS/UTM system. Techniques used for defence evasion include uninstalling/disabling security software or obfuscating/encrypting data and scripts within the UAS or UTM system. Adversaries also leverage and abuse trusted processes to hide and masquerade their malware.

4.5.1 UAS and UTM specific techniques

Avoid (Visual) Identification By avoiding (visual) identification a compromised UAS might be able to stay undetected in a system or area for longer, which would give the adversary more time to implement other defence evasion techniques.

Modify GeoFences Bypass a UTM system's built-in geolocation restrictions by modifying the GPS coordinate parameters within files stored in the system. This could prevent the system from detecting trespassers or allow an adversarial UAS to enter restricted airspace.

Anti-Forensics Manipulation or removal of critical telemetry and file-system logs to prevent post-compromise incident analysis of the system.

Physical Remote Hacking Tool A compromised UAS (with wireless capabilities) could be utilised to conduct secondary attacks against systems similar to that of a compromised computer system.

4.5.2 Existing vectors in the Enterprise matrix

Binary Padding Adversaries can use binary padding to add junk data and change the on-disk representation of malware without affecting the functionality or behaviour of the binary. This will often increase the size of the binary beyond what some security tools are capable of handling due to file size limitations.

Code Signing Code signing provides a level of authenticity on a binary from the developer and a guarantee that the binary has not been tampered with. However, adversaries are known to use code signing certificates to masquerade malware and tools as legitimate binaries . The certificates used during an operation may be created, forged, or stolen by the adversary.

Compile After Delivery Adversaries may attempt to make payloads difficult to discover and analyse by delivering files to victims as uncompiled code. Similar to Obfuscated Files or Information, text-based source code files may subvert analysis and scrutiny from protections targeting executables/binaries.

Component Firmware Some adversaries may employ sophisticated means to compromise computer components and install malicious firmware that will execute adversary code outside of the operating system and main system firmware or BIOS. This technique may be similar to System Firmware but conducted upon other system components that may not have the same capability or level of integrity checking. Malicious device firmware could provide both a persistent level of access to systems despite potential typical failures to maintain access and hard disk re-images, as well as a way to evade host software-based defences and integrity checks.

Connection Proxy Adversaries may use a connection proxy to direct network traffic between systems or act as an intermediary for network communications to a command and control server to avoid direct connections to their infrastructure. Adversaries use these types of proxies to manage command and control communications, to reduce the number of simultaneous outbound network connections, to provide resiliency in the face of connection loss, or to ride over existing trusted communications paths between victims to avoid suspicion.

Deobfuscate/Decode Files or Information Adversaries may use Obfuscated Files or Information to hide artefacts of an intrusion from analysis. They may require separate mechanisms to decode or deobfuscate that information depending on how they intend to use it. Methods for doing that include built-in functionality of malware, Scripting, PowerShell, or by using utilities present on the system.

Disabling Security Tools Adversaries may disable security tools to avoid possible detection of their tools and activities. This can take the form of killing security software or event logging processes or other methods to interfere with security scanning or event reporting.

File and Directory Permissions Modification File and directory permissions are commonly managed by discretionary access control lists (DACLs) specified by the file or directory owner. File and directory DACL implementations may vary by platform, but generally explicitly designate which users/groups can perform which actions (ex: read, write, execute, etc.).

File Deletion Malware, tools, or other non-native files dropped or created on a system by an adversary may leave traces behind as to what was done within a network and how. Adversaries may remove these files over the course of an intrusion to keep their footprint low or remove them at the end as part of the post-intrusion clean-up process.

Install Root Certificate Root certificates are used in public key cryptography to identify a root certificate authority (CA). When a root certificate is installed, the system or application will trust certificates in the root's chain of trust that have been signed by the root certificate. Certificates are commonly used for establishing secure TLS/SSL communications within a web browser. When a user attempts to browse a website that presents a certificate that is not trusted an error message will be displayed to warn the user of the security risk. Depending on the security settings, the browser may not allow the user to establish a connection to the website.

Masquerading Masquerading occurs when the name or location of an executable, legitimate or malicious, is manipulated or abused for the sake of evading defences and observation. Several different variations of this technique have been observed.

Obfuscated Files or Information Adversaries may attempt to make an executable or file difficult to discover or analyse by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behaviour that can be used across different platforms and the network to evade defences.

Port Knocking Port Knocking is a well-established method used by both defenders and adversaries to hide open ports from access. To enable a port, an adversary sends a series of packets with certain characteristics before the port

will be opened. Usually this series of packets consists of attempted connections to a predefined sequence of closed ports, but can involve unusual flags, specific strings or other unique characteristics. After the sequence is completed, opening a port is often accomplished by the host based firewall, but could also be implemented by custom software.

Process Injection Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process.

Redundant Access Adversaries may use more than one remote access tool with varying command and control protocols or credentialed access to remote services so they can maintain access if an access mechanism is detected or mitigated.

Rootkit Rootkits are programs that hide the existence of malware by intercepting and modifying operating system API calls that supply system information. Rootkits or rootkit enabling functionality may reside at the user or kernel level in the operating system or lower, to include a Hypervisor, Master Boot Record, or the System Firmware.

Scripting Adversaries may use scripts to aid in operations and perform multiple actions that would otherwise be manual. Scripting is useful for speeding up operational tasks and reducing the time required to gain access to critical resources. Some scripting languages may be used to bypass process monitoring mechanisms by directly interacting with the operating system at an API level instead of calling other programs.

Signed Script Proxy Execution Scripts signed with trusted certificates can be used to proxy execution of malicious files. This behaviour may bypass signature validation restrictions and application whitelisting solutions that do not account for use of these scripts.

Timestomp Timestomping is a technique that modifies the timestamps of a file (the modify, access, create, and change times), often to mimic files that are in the same folder. This is done, for example, on files that have been modified or created by the adversary so that they do not appear conspicuous to forensic investigators or file analysis tools. Timestomping may be used along with file name Masquerading to hide malware and tools.

Valid Accounts Adversaries may steal the credentials of a specific user or service account using Credential Access techniques or capture credentials earlier in their reconnaissance process through social engineering for means of gaining Initial Access.

Virtualisation/Sandbox Evasion Adversaries may check for the presence of a Virtual Machine Environment (VME) or sandbox to avoid potential detection of tools and activities. If the adversary detects a VME, they may alter their malware to conceal the core functions of the implant or disengage from the victim. They may also search for VME artefacts before dropping secondary or additional payloads. Adversaries may use the information from learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviours.

Web Service Adversaries may use an existing, legitimate external Web service as a means for relaying commands to a compromised system.

Web Session Cookie Adversaries can use stolen session cookies to authenticate to web applications and services. This technique bypasses some multi-factor authentication protocols since the session is already authenticated [14].

4.5.3 Existing vectors in the ICS matrix

Rogue Master Device Adversaries may setup a rogue master to leverage control server functions to communicate with slave devices. A rogue master device can be used to send legitimate control messages to other control system devices, affecting processes in unintended ways. It may also be used to disrupt network communications by capturing and receiving the network traffic meant for the actual master device. Impersonating a master device may also allow an adversary to avoid detection.

Spoof Reporting Message Adversaries may spoof reporting messages in control systems environments to achieve evasion and assist with impairment of process controls. Reporting messages are used in control systems so that operators and network defenders can understand the status of the network. Reporting messages show the status of devices and any important events that the devices control.

4.6 Credential access

The adversary is trying to steal account names and passwords of the UAS in order to gain access to restricted data.

Credential Access consists of techniques for stealing credentials like account names and passwords to access into a UAS or UTM system. Techniques used to get credentials include keylogging onto the UI software of the UAS or credential dumping from a UTM system. Using legitimate credentials can give adversaries access to systems, make them harder to detect, and provide the opportunity to create more accounts to help achieve their goals.

4.6.1 UAS and UTM specific techniques

No UAS and UTM specific techniques were found for *Credential access*. The techniques that were found were already covered in either the Enterprise or the ICS matrix.

4.6.2 Existing vectors in the Enterprise matrix

Account Manipulation Account manipulation may aid adversaries in maintaining access to credentials and certain permission levels within an environment. Manipulation could consist of modifying permissions, modifying credentials, adding or changing permission groups, modifying account settings, or modifying how authentication is performed. These actions could also include account activity designed to subvert security policies, such as performing iterative password updates to subvert password duration policies and preserve the life of compromised credentials. In order to create or manipulate accounts, the adversary must already have sufficient permissions on systems or the domain.

Brute Force Adversaries may use brute force techniques to attempt access to accounts when passwords are unknown or when password hashes are obtained.

Cloud Instance Metadata API Adversaries may attempt to access the Cloud Instance Metadata API to collect credentials and other sensitive data.

Credentials from Web Browsers Adversaries may acquire credentials from web browsers by reading files specific to the target browser.

Credentials in Files Adversaries may search local file systems and remote file shares for files containing passwords. These can be files created by users to store their own credentials, shared credential stores for a group of individuals, configuration files containing passwords for a system or service, or source code/binary files containing embedded passwords.

Exploitation for Credential Access Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. Credentialing and authentication mechanisms may be targeted for exploitation by adversaries as a means to gain access to useful credentials or circumvent the process to gain access to systems.

Network Sniffing Network sniffing refers to using the network interface on a system to monitor or capture information sent over a wired or wireless connection. An adversary may place a network interface into promiscuous mode to passively access data in transit over the network, or use span ports to capture a larger amount of data.

This includes recording or listening in on communication sent to or from UAS. Since this is often done over RF, it can be intercepted like any other RF signal.

Acquiring Private Keys Private cryptographic keys and certificates are used for authentication, encryption/decryption, and digital signatures.

Steal Application Access Token Adversaries can steal user application access tokens as a means of acquiring credentials to access remote systems and resources. This can occur through social engineering and typically requires user action to grant access.

4.6.3 Existing vectors in the ICS matrix

No additional techniques were found for *Credential access* in the ICS matrix. The techniques that were found were already covered in the Enterprise matrix.

4.7 Discovery

The adversary is trying to figure out your environment.

Discovery consists of techniques an adversary may use to gain knowledge about the system and internal network. In this case especially other UAS or systems connected to or communicating with the compromised UAS. These techniques help adversaries observe the environment and orient themselves before deciding how to act. They also allow adversaries to explore what they can control within the UAS or UTM system and what is around their entry point in order to discover how it could benefit their current objective.
It is important to notice that this is a *post-compromise* phase.

4.7.1 UAS and UTM specific techniques

Swarm Discovery Adversaries might be able to discover other UAS or systems operating in the same swarm as the compromised UAS. (Might be covered in "System Network Connections Discovery") (Might be unique when you combine it with Network Connection Enumeration to find the role of the UAS in the swarm or UTM system.)

Signal and Protocol Analysis Analysing the underlying communication signal and protocols used in the RF band between the UAS and its controller utilising SDR tooling in order to identify what UAS brand, model, or type is used. which could be used to find known vulnerabilities.

Pilot/Control Device Discovery Adversaries may attempt to discover connected devices that have control over it/other UAS to learn more about the network they are operating in and which are interesting targets for privilege escalation.

UAS Data Platform Dashboard An adversary may use a UAS data platform dashboard GUI with stolen credentials to gain useful information from an operational UTM environment, such as analytics platforms, survey/modelling platforms, fleet hubs, or Counter-UAS (CUAS).

Reconnaissance Use cameras and other sensors to discover, explore, and map (restricted/private) areas and equipment.

4.7.2 Existing vectors in the Enterprise matrix

Cloud Service Discovery An adversary may attempt to enumerate the cloud services running on a system after gaining access. These methods can differ depending on if it's platform-as-a-service (PaaS), infrastructure-as-a-service (IaaS), or software-as-a-service (SaaS).

File and Directory Discovery Adversaries may enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system. Adversaries may use the information from this attack during automated discovery to shape follow-on behaviours, including whether or not the adversary fully infects the target and/or attempts specific actions.

Network Sniffing Network sniffing refers to using the network interface on a system to monitor or capture information sent over a wired or wireless connection. An adversary may place a network interface into promiscuous mode to passively access data in transit over the network, or use span ports to capture a larger amount of data.

Network Service Scanning Adversaries may attempt to get a listing of services running on remote hosts, including those that may be vulnerable to remote software exploitation. Methods to acquire this information include port scans and vulnerability scans using tools that are brought onto a system.

Peripheral Device Discovery Adversaries may attempt to gather information about attached peripheral devices and components connected to a computer system. The information may be used to enhance their awareness of the system and network environment or may be used for further actions.

Remote System Discovery Adversaries will likely attempt to get a listing of other systems by Internet Protocol (IP) address, hostname, or other logical identifier on a network that may be used for Lateral Movement from the current system. Functionality could exist within remote access tools to enable this, but utilities available on the operating system could also be used. Adversaries may also use local host files in order to discover the hostname to IP address mappings of remote systems.

Software Discovery Adversaries may attempt to get a listing of non-security related software that is installed on the system. Adversaries may use the information from Software Discovery during automated discovery to shape follow-on behaviours, including whether or not the adversary fully infects the target and/or attempts specific actions.

System Information Discovery An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Adversaries may use the information

from this attack during automated discovery to shape follow-on behaviours, including whether or not the adversary fully infects the target and/or attempts specific actions.

System Network Configuration Discovery Adversaries will likely look for details about the network configuration and settings of systems they access or through information discovery of remote systems. Several operating system administration utilities exist that can be used to gather this information. Examples include ARP, ipconfig/ifconfig, nbtstat, and route.

System Network Connections Discovery Adversaries may attempt to get a listing of network connections to or from the compromised system they are currently accessing or from remote systems by querying for information over the network.

System Time Discovery An adversary may gather the system time and/or time zone from a local or remote system. The information could be useful for performing other techniques, such as executing a file with a Scheduled Task, or to discover locality information based on time zone to assist in victim targeting.

Virtualisation/Sandbox Evasion Adversaries may check for the presence of a Virtual Machine Environment (VME) or sandbox to avoid potential detection of tools and activities. If the adversary detects a VME, they may alter their malware to conceal the core functions of the implant or disengage from the victim. They may also search for VME artefacts before dropping secondary or additional payloads. Adversaries may use the information from learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviours.

4.7.3 Existing vectors in the ICS matrix

Control Device Identification Adversaries may perform control device identification to determine the make and model of a target device. Management software and device APIs may be utilized by the adversary to gain this information. By identifying and obtaining device specifics, the adversary may be able to determine device vulnerabilities. This device information can also be used to understand device functionality and inform the decision to target the environment.

Discovery Adversaries may use input/output (I/O) module discovery to gather key information about a control system device. An I/O module is a device that allows the control system device to either receive or send signals to other devices. These signals can be analog or digital, and may support a number of different protocols. Devices are often able to use attachable I/O modules to increase the number of inputs and outputs that it can utilize. An adversary with access to a device can use native device functions to enumerate I/O modules that are connected to the device. Information regarding the I/O modules can aid the adversary in understanding related control processes.

Network Connection Enumeration Adversaries may perform network connection enumeration to discover information about device communication patterns. If an adversary can inspect the state of a network connection with tools, such as netstat, in conjunction with System Firmware, then they can determine the role of certain devices on the network.

4.8 Lateral movement

The adversary is trying to explore through the UAS/UTM environment to find their intended target.

Lateral Movement consists of techniques that adversaries use to enter and control remote systems on a UTM or UAS network. Following through on their primary objective often requires exploring the network to find their target (could be another UAS, a UTM system or data) and subsequently gaining access to it. Reaching their objective often involves pivoting through multiple systems and accounts to gain access. Adversaries might install their own remote access tools to accomplish Lateral Movement or use legitimate credentials with native network and operating system tools, which may be stealthier.

4.8.1 UAS and UTM specific techniques

No UAS and UTM specific techniques were found for *Lateral movement*. The techniques that were found were already covered in either the Enterprise or the ICS matrix.

4.8.2 Existing vectors in the Enterprise matrix

Remote File Copy Files may be copied from one system to another to stage adversary tools or other files over the course of an operation. Files may be copied from an external adversary-controlled system through the Command and Control channel to bring tools into the victim network or through alternate protocols with another tool such as FTP.

Remote Services An adversary may use Valid Accounts to log into a service specifically designed to accept remote connections, such as telnet, SSH, and VNC. The adversary may then perform actions as the logged-on user.

Replication Through Removable Media Adversaries may move onto systems, possibly those on disconnected or air-gapped networks, by copying malware to removable media and taking advantage of Autorun features when the media is inserted into a system and executes. In the case of Lateral Movement, this may occur through modification of executable files stored on removable media or by copying malware and renaming it to look like a legitimate file to trick users into executing it on a separate system. In the case of Initial Access, this may occur through manual manipulation of the media, modification of systems used to initially format the media, or modification to the media's firmware itself.

SSH Hijacking SSH is a standard means of remote access on Linux systems. It allows a user to connect to another system via an encrypted tunnel, commonly authenticating through a password, certificate or the use of an asymmetric encryption key pair.

Taint Shared Content Content stored on network drives or in other shared locations may be tainted by adding malicious programs, scripts, or exploit code to otherwise valid files. Once a user opens the shared tainted content, the malicious portion can be executed to run the adversary's code on a remote system. Adversaries may use tainted shared content to move laterally.

Third-party Software Third-party applications and software deployment systems may be in use in the network environment for administration purposes. If an adversary gains access to these systems, then they may be able to execute code.

Web Session Cookie Adversaries can use stolen session cookies to authenticate to web applications and services. This technique bypasses some multi-factor authentication protocols since the session is already authenticated.

4.8.3 Existing vectors in the ICS matrix

Default Credentials Adversaries may leverage manufacturer or supplier set default credentials on control system devices. These default credentials may have administrative permissions and may be necessary for initial configuration of the device. It is general best practice to change the passwords for these accounts as soon as possible, but some manufacturers may have devices that have passwords or usernames that cannot be changed.

4.9 Collection

The adversary is trying to gather data of interest to their goal.

Collection consists of techniques adversaries may use to gather information and the sources information is collected from that are relevant to following through on the adversarys objectives. Frequently, the next goal after collecting data is to steal (exfiltrate) the data. Common target sources of UAS or UTM systems include various drive types, system performances, flight logs, location, photos and video footages. Common collection methods include capturing sensor or camera data, controller input and UTM logs of drone flight profiles.

4.9.1 UAS and UTM specific techniques

Shared Swarm Data There might be data in the network of the local swarm the UAS is operating in that might be requested or collected from the swarm or the UTM system that is otherwise not available.

Sensor Capture An adversary can use the UAS's sensors to infer other information about the UAS and its surroundings. This information includes but is not limited to location, altitude, controller location, pitch, and speed.

4.9.2 Existing vectors in the Enterprise matrix

Audio Capture An adversary can leverage a computer's peripheral devices (e.g., microphones and webcams) or applications (e.g., voice and video call services) to capture audio recordings for the purpose of listening into sensitive conversations to gather information.

Automated Collection Once established within a system or network, an adversary may use automated techniques for collecting internal data. Methods for performing this technique could include use of Scripting to search for and copy information fitting set criteria such as file type, location, or name at specific time intervals. This functionality could also be built into remote access tools.

Data from Cloud Storage Object Adversaries may access data objects from improperly secured cloud storage.

Data from Information Repositories Adversaries may leverage information repositories to mine valuable information. Information repositories are tools that allow for storage of information, typically to facilitate collaboration or information sharing between users, and can store a wide variety of data that may aid adversaries in further objectives, or direct access to the target information.

Data from Local Systems Sensitive data can be collected from local system sources, such as the file system or databases of information residing on the system prior to Exfiltration. This can include telemetry data, data stored within internal or external storage devices, system files and in some cases, owner and registration information.

Data from Removable Media Sensitive data can be collected from any removable media (optical disk drive, USB memory, etc.) connected to the compromised system prior to Exfiltration.

Data Staging Collected data is staged in a central location or directory prior to Exfiltration.

Screen Capture Adversaries may attempt to take screen captures of the desktop to gather information over the course of an operation. Screen capturing functionality may be included as a feature of a remote access tool used in post-compromise operations.

Video Capture An adversary can leverage a computer's peripheral devices (e.g., integrated cameras or webcams) or applications (e.g., video call services) to capture video recordings for the purpose of gathering information. Images may also be captured from devices or applications, potentially in specified intervals, in lieu of video files.

4.9.3 Existing vectors in the ICS matrix

Detect Program State Adversaries may seek to gather information about the current state of a program on a UAS or UTM system. State information reveals information about the program, including whether it's running, halted, stopped, or has generated an exception. This information may be leveraged as a verification of malicious program execution or to determine if a UAS or UTM system is ready to download a new program.

Location Identification Adversaries may perform location identification using device data to inform operations and targeted impact for attacks. Location identification data can come in a number of forms, including geographic location, location relative to other control system devices, time zone, and current time. An adversary may use an embedded global positioning system (GPS) module in a device to figure out the physical coordinates of a device.

Role Identification Adversaries may perform role identification of devices involved with physical processes of interest in a target control system. Control systems devices often work in concert to control a physical process. Each device can have one or more roles that it performs within that control process. By collecting this role-based data, an adversary can construct a more targeted attack.

4.10 Command and Control (C2)

The adversary is trying to communicate with compromised systems to control them.

Command and Control consists of techniques that adversaries may use to communicate with the UAS or UTM systems under their control within a victim network. Adversaries commonly attempt to mimic normal, expected traffic to avoid detection.

4.10.1 UAS and UTM specific techniques

Relay Although it works better when using multiple UAS, e.g. UAS in a swarm or those controlled by by a UTM, even a single UAS can carry equipment like antennas that can be used to relay signals or set up a remote network for adversaries, or increase the range at which the adversaries can operate in the network.

4.10.2 Existing vectors in the Enterprise matrix

Commonly Used Port Adversaries may communicate over a commonly used port to bypass firewalls or network detection systems and to blend with normal network activity to avoid more detailed inspection.

Communication Through Removable Media Adversaries can perform command and control between compromised hosts on potentially disconnected

networks using removable media to transfer commands from system to system. Both systems would need to be compromised, with the likelihood that an Internet-connected system was compromised first and the second through lateral movement by Replication Through Removable Media. Commands and files would be relayed from the disconnected system to the Internet-connected system to which the adversary has direct access.

Connection Proxy Adversaries may use a connection proxy to direct network traffic between systems or act as an intermediary for network communications to a command and control server to avoid direct connections to their infrastructure. Adversaries use these types of proxies to manage command and control communications, to reduce the number of simultaneous outbound network connections, to provide resiliency in the face of connection loss, or to ride over existing trusted communications paths between victims to avoid suspicion.

Custom Command and Control Protocol Adversaries may communicate using a custom command and control protocol instead of encapsulating commands/data in an existing Standard Application Layer Protocol. Implementations include mimicking well-known protocols or developing custom protocols (including raw sockets) on top of fundamental protocols provided by TCP/IP/another standard network stack.

Custom Cryptographic Protocol Adversaries may use a custom cryptographic protocol or algorithm to hide command and control traffic. A simple scheme, such as XOR-ing the plaintext with a fixed key, will produce a very weak ciphertext.

Data Encoding Command and control (C2) information is encoded using a standard data encoding system. Use of data encoding may be to adhere to existing protocol specifications and includes use of ASCII, Unicode, Base64, MIME, UTF-8, or other binary-to-text and character encoding systems. Some data encoding systems may also result in data compression, such as gzip.

Data Obfuscation Command and control (C2) information is encoded using a standard data encoding system. Use of data encoding may be to adhere to existing protocol specifications and includes use of ASCII, Unicode, Base64, MIME, UTF-8, or other binary-to-text and character encoding systems. Some data encoding systems may also result in data compression, such as gzip.

Fallback Channels Adversaries may use fallback or alternate communication channels if the primary channel is compromised or inaccessible in order to maintain reliable command and control and to avoid data transfer thresholds.

Multi-hop Proxy To disguise the source of malicious traffic, adversaries may chain together multiple proxies. Typically, a defender will be able to identify the last proxy traffic traversed before it enters their network; the defender may or may not be able to identify any previous proxies before the last-hop proxy. This technique makes identifying the original source of the malicious traffic even

more difficult by requiring the defender to trace malicious traffic through several proxies to identify its source.

Multi-Stage Channels Adversaries may create multiple stages for command and control that are employed under different conditions or for certain functions. Use of multiple stages may obfuscate the command and control channel to make detection more difficult.

Multiband Communication Some adversaries may split communications between different protocols. There could be one protocol for inbound command and control and another for outbound data, allowing it to bypass certain firewall restrictions. The split could also be random to simply avoid data threshold alerts on any one communication.

Multilayer Encryption An adversary performs C2 communications using multiple layers of encryption, typically (but not exclusively) tunnelling a custom encryption scheme within a protocol encryption scheme such as Hypertext Transfer Protocol Secure (HTTPS) or Simple Mail Transfer Protocol Secure (SMTPS).

Port Knocking Port Knocking is a well-established method used by both defenders and adversaries to hide open ports from access. To enable a port, an adversary sends a series of packets with certain characteristics before the port will be opened. Usually this series of packets consists of attempted connections to a predefined sequence of closed ports, but can involve unusual flags, specific strings or other unique characteristics. After the sequence is completed, opening a port is often accomplished by the host based firewall, but could also be implemented by custom software.

Remote Access Tools An adversary may use legitimate desktop support and remote access software, such as Team Viewer, Go2Assist, LogMein, AmmyyAdmin, etc, to establish an interactive command and control channel to target systems within networks. These services are commonly used as legitimate technical support software, and may be whitelisted within a target environment. Remote access tools like VNC, Ammy, and Teamviewer are used frequently when compared with other legitimate software commonly used by adversaries.

Remote File Copy Files may be copied from one system to another to stage adversary tools or other files over the course of an operation. Files may be copied from an external adversary-controlled system through the Command and Control channel to bring tools into the victim network or through alternate protocols with another tool such as FTP.

Standard Application Layer Protocol Adversaries may communicate using a common, standardized application layer protocol such as HTTP, HTTPS, SMTP, or DNS to avoid detection by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.

Standard Cryptographic Protocol Adversaries may explicitly employ a known encryption algorithm to conceal command and control traffic rather than relying on any inherent protections provided by a communication protocol. Despite the use of a secure algorithm, these implementations may be vulnerable to reverse engineering if necessary secret keys are encoded and/or generated within malware samples/configuration files.

Standard Non-Application Layer Protocol Use of a standard non-application layer protocol for communication between host and C2 server or among infected hosts within a network. The list of possible protocols is extensive. Specific examples include use of network layer protocols, such as the Internet Control Message Protocol (ICMP), transport layer protocols, such as the User Datagram Protocol (UDP), session layer protocols, such as Socket Secure (SOCKS)), as well as redirected/tunnelled protocols, such as Serial over LAN (SOL).

Uncommonly Used Port Adversaries may conduct C2 communications over a non-standard port to bypass proxies and firewalls that have been improperly configured.

4.10.3 Existing vectors in the ICS matrix

No additional techniques were found for *Command and Control* in the ICS matrix. The techniques that were found were already covered in the Enterprise matrix.

4.11 Exfiltration

The adversary is trying to steal data from the compromised UAS or UTM system.

Exfiltration consists of techniques that adversaries may use to steal data from the compromised network. Once theyve collected data, adversaries often package it to avoid detection while removing it. This can include compression and encryption. Techniques for getting data out of a target network typically include transferring it over their command and control channel or an alternate channel and may also include putting size limits on the transmission. Common target sources of UAS or UTM systems include various drive types, system performances, flight logs, location, photos and video footages.

4.11.1 UAS and UTM specific techniques

Exfiltrate Compromised UAS Fly away with a compromised UAS after it has collected the sought after data.

4.11.2 Existing vectors in the Enterprise matrix

Automated Exfiltration Data, such as sensitive documents, may be exfiltrated through the use of automated processing or Scripting after being gathered during Collection.

Data Compression An adversary may compress data (e.g., sensitive documents) that is collected prior to exfiltration in order to make it portable and minimize the amount of data sent over the network. The compression is done separately from the exfiltration channel and is performed using a custom program or algorithm, or a more common compression library or utility such as 7zip, RAR, ZIP, or zlib.

Data Encryption Data is encrypted before being exfiltrated in order to hide the information that is being exfiltrated from detection or to make the exfiltration less conspicuous upon inspection by a defender. The encryption is performed by a utility, programming library, or custom algorithm on the data itself and is considered separate from any encryption performed by the command and control or file transfer protocol.

Data Transfer Size Limits An adversary may exfiltrate data in fixed size chunks instead of whole files or limit packet sizes below certain thresholds. This approach may be used to avoid triggering network data transfer threshold alerts.

Exfiltration Over Alternative Protocol Data exfiltration is performed with a different protocol from the main command and control protocol or channel. The data is likely to be sent to an alternate network location from the main command and control server. Alternate protocols include FTP, SMTP, HTTP/S, DNS, SMB, or any other network protocol not being used as the main command and control channel. Different channels could include Internet Web services such as cloud storage.

Exfiltration Over Command and Control Channel Data exfiltration is performed over the Command and Control channel. Data is encoded into the normal communications channel using the same protocol as command and control communications.

Exfiltration Over Other Network Medium Exfiltration could occur over a different network medium than the command and control channel. If the command and control network is a wired Internet connection, the exfiltration may occur, for example, over a WiFi connection, modem, cellular data connection, Bluetooth, or another radio frequency (RF) channel. Adversaries could choose to do this if they have sufficient access or proximity, and the connection might not be secured or defended as well as the primary Internet-connected channel because it is not routed through the same enterprise network.

Exfiltration Over Physical Medium In certain circumstances, such as an air-gapped network compromise, exfiltration could occur via a physical medium or device introduced by a user. Such media could be an external hard drive, USB drive, cellular phone, MP3 player, or other removable storage and processing device. The physical medium or device could be used as the final exfiltration point or to hop between otherwise disconnected systems.

Scheduled Transfer Data exfiltration may be performed only at certain times of day or at certain intervals. This could be done to blend traffic patterns with normal activity or availability.

Transfer Data to Cloud Account An adversary may exfiltrate data by transferring the data, including backups of cloud environments, to another cloud account they control on the same service to avoid typical file transfers/downloads and network-based exfiltration detection.

4.11.3 Existing vectors in the ICS matrix

Since *Exfiltration* is not included in the ICS matrix, no additional techniques were found for this tactic in this matrix.

4.12 Inhibit response function

The adversary is trying to prevent your safeguards from adequately responding to unsafe states within the UAS or UTM system.

These techniques aim to actively deter and prevent expected alarms and responses that arise due to statuses in the UAS or UTM environment. Adversaries may modify or update system logic, or even outright prevent responses with a Denial-of-Service (DoS). As prevention functions are generally dormant, reporting and processing functions can appear fine, but may have been altered to prevent failure responses in dangerous scenarios. Adversaries may use these techniques to follow through with or provide cover for Impact techniques.

4.12.1 UAS and UTM specific techniques

Modify GeoFences Bypass the UASs built-in geolocation restrictions by modifying the GPS coordinate parameters within files stored on the UAS. Alternatively, restrict the UAS from operating in legitimate areas or returning to its Base Station by enforcing geolocation coordinate parameters.

Bypass a UTM system's built-in geolocation restrictions by modifying the GPS coordinate parameters within files stored in the system. This could prevent the system from detecting trespassers or allow an adversarial UAS to enter restricted airspace.

Disable CUAS capabilities If a UTM has built in CUAS capabilities, for instance a whitelist for friendly UAS or No Fly Zone (NFZ) for unknown UAS, this functionality could be disabled on a specific or all managed areas. Specifically, this could occur if CUAS functionalities are modifiable within the UTM platform or command station, post-compromise.

4.12.2 Existing vectors in the Enterprise matrix

Since *Inhibit response function* is not included in the Enterprise matrix, no additional techniques were found for this tactic in this matrix.

4.12.3 Existing vectors in the ICS matrix

Activate Firmware Update Mode Adversaries may activate firmware update mode on devices to prevent expected response functions from engaging in reaction to an emergency or process malfunction. This mode may halt process monitoring and related functions to allow new firmware to be loaded. A device left in update mode may be placed in an inactive holding state if no firmware is provided to it. By entering and leaving a device in this mode, the adversary may deny its usual functionalities.

Alarm Suppression Adversaries may target protection function alarms to prevent them from notifying operators of critical conditions. Alarm messages may be a part of an overall reporting system and of particular interest for adversaries. Disruption of the alarm system does not imply the disruption of the reporting system as a whole.

Block Command Message Adversaries may block a command message from reaching its intended target to prevent command execution. In OT networks, command messages are sent to provide instructions to control system devices. A blocked command message can inhibit response functions from correcting a disruption or unsafe condition.

Block Reporting Message Adversaries may block or prevent a reporting message from reaching its intended target. Reporting messages relay the status of control system devices, which can include event log data and I/O values of the associated device. By blocking these reporting messages, an adversary can potentially hide their actions from an operator.

Blocking reporting messages in control systems that manage physical processes may contribute to system impact, causing inhibition of a response function. A control system may not be able to respond in a proper or timely manner to an event, such as a dangerous fault, if its corresponding reporting message is blocked.

Denial of Service (DoS) Adversaries may perform Denial-of-Service (DoS) attacks to disrupt expected device functionality. Examples of DoS attacks include overwhelming the target device with a high volume of requests in a short time period and sending the target device a request it does not know how to handle. Disrupting device state may temporarily render it unresponsive, possibly lasting until a reboot can occur. When placed in this state, devices may be unable to send and receive requests, and may not perform expected response functions in reaction to other events in the environment.

In UAS this can be done in several ways, for instance through blocking communication with the controller, by jamming the RF frequency bands, GPS/ GLONASS, or sending de-authentication packets to its Wi-Fi based communication channel. Another way is by disrupting their availability through using an Electromagnetic Pulse (EMP).

Device Restart/Shutdown Adversaries may forcibly restart or shutdown a device in the ICS environment to disrupt and potentially cause adverse effects

on the physical processes it helps to control. Methods of device restart and shutdown exist as built-in, standard functionalities. Device restart or shutdown may also occur as a consequence of changing a device into an alternative mode of operation for testing or firmware loading.

Using a (targeted) Electromagnetic Pulse (EMP) in order to temporarily disrupt one or several UAS in a certain area, forcing them to shutdown or restart.

Modify Control Logic Adversaries may place malicious code in a system, which can cause the system to malfunction by modifying its control logic. Control system devices use programming languages (e.g. relay ladder logic) to control physical processes by affecting actuators, which cause machines to operate, based on environment sensor readings. These devices often include the ability to perform remote control logic updates.

An adversary can de-calibrate a sensor by removing functions in control logic that account for sensor error. This can be used to change a control process without actually spoofing command messages to a controller or device.

Program Download Adversaries may perform a program download to load malicious or unintended program logic on a device as a method of persistence or to disrupt response functions or process control.

Rootkit Adversaries may deploy rootkits to hide the presence of programs, files, network connections, services, drivers, and other system components. Rootkits are programs that hide the existence of malware by intercepting and modifying operating-system API calls that supply system information. Rootkits or rootkit-enabling functionality may reside at the user or kernel level in the operating system, or lower.

Utilise/Change Operating Mode Adversaries may place controllers into an alternate mode of operation to enable configuration setting changes for evasive code execution or to inhibit device functionality. Programmable controllers typically have several modes of operation. These modes can be broken down into three main categories: program run, program edit, and program write. Each of these modes puts the device in a state in which certain functions are available. For instance, the program edit mode allows alterations to be made to the user program while the device is still online.

4.13 Impair process control

The adversary is trying to manipulate, disable, or damage physical control processes of the UAS or UTM system.

Impair Process Control consists of techniques that adversaries use to disrupt control logic of the drone and UTM system and cause detrimental effects to processes controlled in the unmanned environment. Targets of interest may include active procedures or parameters that manipulate the physical environment. These techniques can also include prevention or manipulation of reporting elements and control logic. If an adversary has modified process functionality,

then they may also obfuscate the results, which are often self-revealing in their impact on the performance outcome of a drone or within the UTM environment. The direct physical control these techniques exert may also threaten the safety of UAS/UTM operators and downstream users, which can prompt response mechanisms. Adversaries may follow up with or use Inhibit Response Function techniques in tandem, to assist with the successful abuse of control processes to result in Impact.

4.13.1 UAS and UTM specific techniques

Rogue/Malicious/Rooted Controller Corrupt/root the controller through malware loaded phones or by rooting the smart controller.

Manipulate Auto-Pilot Using (projected) images to trick UAS into displaying unplanned behaviour. By using false images, the UAS might make wrong decisions based on false environment input.

Modify GeoFences Bypass the UASs built-in geolocation restrictions by modifying the GPS coordinate parameters within files stored on the UAS. Alternatively, restrict the UAS from operating in legitimate areas or returning to its Base Station by enforcing geolocation coordinate parameters.

Bypass a UTM system's built-in geolocation restrictions by modifying the GPS coordinate parameters within files stored in the system. This could prevent the system from detecting trespassers or allow an adversarial UAS to enter restricted airspace.

4.13.2 Existing vectors in the Enterprise matrix

Since *Impair process control* is not included in the Enterprise matrix, no additional techniques were found for this tactic in this matrix.

4.13.3 Existing vectors in the ICS matrix

Change Program State Adversaries may attempt to change the state of the current program on a control device. Program state changes may be used to allow for another program to take over control or be loaded onto the device.

Modify Control Logic Adversaries may place malicious code in a system, which can cause the system to malfunction by modifying its control logic. Control system devices use programming languages (e.g. relay ladder logic) to control physical processes by affecting actuators, which cause machines to operate, based on environment sensor readings. These devices often include the ability to perform remote control logic updates.

An adversary can de-calibrate a sensor by removing functions in control logic that account for sensor error. This can be used to change a control process without actually spoofing command messages to a controller or device.

Modify Parameter Adversaries may modify parameters used to instruct industrial control system devices. These devices operate via programs that dictate how and when to perform actions based on such parameters. Such parameters can determine the extent to which an action is performed and may specify additional options. For example, a program on a control system device dictating motor processes may take a parameter defining the total number of seconds to run that motor.

An adversary can potentially modify these parameters to produce an outcome outside of what was intended by the operators. By modifying system and process critical parameters, the adversary may cause Impact to equipment and/or control processes. Modified parameters may be turned into dangerous, out-of-bounds, or unexpected values from typical operations.

By modifying or overwriting file-system parameters or the instructions being sent by a controller, the UAS deviates from its original pre-programmed flight path.

Another example for UAS is that some support the functionality to return to a specific location if communication with the operator is lost. An adversary could change the Return-to-Home (RTH) location, which will modify the predefined RTH point set by the operator, which would cause the UAS to return to an unexpected location, possibly allowing the adversary to steal the UAS.

Module Firmware Adversaries may install malicious or vulnerable firmware onto modular hardware devices. Control system devices often contain modular hardware devices. These devices may have their own set of firmware that is separate from the firmware of the main control system equipment.

Program Download Adversaries may perform a program download to load malicious or unintended program logic on a device as a method of persistence or to disrupt response functions or process control.

Rogue Master Device Adversaries may setup a rogue master to leverage control server functions to communicate with slave devices. A rogue master device can be used to send legitimate control messages to other control system devices, affecting processes in unintended ways. It may also be used to disrupt network communications by capturing and receiving the network traffic meant for the actual master device. Impersonating a master device may also allow an adversary to avoid detection.

Service Stop Adversaries may stop or disable services on a system to render those services unavailable to legitimate users. Stopping critical services can inhibit or stop response to an incident or aid in the adversary's overall objectives to cause damage to the environment.

Spoof Reporting Message Adversaries may spoof reporting messages in control systems environments to achieve evasion and assist with impairment of process controls. Reporting messages are used in control systems so that operators and network defenders can understand the status of the network. Reporting messages show the status of devices and any important events that the devices control.

Unauthorised Command Message Adversaries may send unauthorized command messages to instruct control systems devices to perform actions outside their expected functionality for process control. Command messages are used in ICS networks to give direct instructions to control systems devices. If an adversary can send an unauthorized command message to a control system, then it can instruct the control systems device to perform an action outside the normal bounds of the device's actions. An adversary could potentially instruct a control systems device to perform an action that will cause an Impact.

4.14 Impact

The adversary is trying to manipulate, interrupt, or destroy your systems and data.

Impact consists of techniques that adversaries use to disrupt availability or compromise integrity by manipulating business and operational processes within the UTM system or UAS. Techniques used for impact can include destroying of collected audio-visual footages, or tampering with flight data. In some cases, drone operational processes can look fine, but may have been altered to benefit the adversaries goals. These techniques might be used by adversaries to follow through on their end goal or to provide cover for a confidentiality breach.

4.14.1 UAS and UTM specific techniques

Theft of Operational Equipment Individual UAS or entire swarms may be physically moved out of the operator's control and into that of an adversaries, effectively stealing the equipment. Also see: Loss of Productivity and Revenue.

Modify Return-to-Home (RTH) Location Some UAS support the functionality to return to a specific location if communication with the operator is lost. An adversary could change the RTH location, which will modify the pre-defined RTH point set by the operator. If combined with Denial of Control it could cause a Loss of Availability if the UAS is moved out of the operators range or flown to a location controlled by the adversary.

Unauthorised Delivery A UAS can be abused to drop whatever payload is carrying without the operator's authorisation. This technique also covers the dropping of explosives.

Physical Exfiltration A UAS could be used to steal physical objects, if equipped with the appropriate payload mechanism.

4.14.2 Existing vectors in the Enterprise matrix

Account Access Removal Adversaries may interrupt availability of system and network resources by inhibiting access to accounts utilized by legitimate users. Accounts may be deleted, locked, or manipulated (ex: changed credentials) to remove access to accounts.

Data Destruction Adversaries may destroy data and files on specific systems or in large numbers on a network to interrupt availability to systems, services, and network resources. Data destruction is likely to render stored data irrecoverable by forensic techniques through overwriting files or data on local and remote drives. Common operating system file deletion commands such as del and rm often only remove pointers to files without wiping the contents of the files themselves, making the files recoverable by proper forensic methodology.

Data Encrypted for Impact Adversaries may encrypt data on target systems or on large numbers of systems in a network to interrupt availability to system and network resources. They can attempt to render stored data inaccessible by encrypting files or data on local and remote drives and withholding access to a decryption key. This may be done in order to extract monetary compensation from a victim in exchange for decryption or a decryption key (ransomware) or to render data permanently inaccessible in cases where the key is not saved or transmitted. In the case of ransomware, it is typical that common user files like Office documents, PDFs, images, videos, audio, text, and source code files will be encrypted. In some cases, adversaries may encrypt critical system files, disk partitions, and the MBR.

Defacement Adversaries may modify visual content available internally or externally to an enterprise network. Reasons for Defacement include delivering messaging, intimidation, or claiming (possibly false) credit for an intrusion.

Disk Content Wipe Adversaries may erase the contents of storage devices on specific systems as well as large numbers of systems in a network to interrupt availability to system and network resources.

Disk Structure Wipe Adversaries may corrupt or wipe the disk data structures on hard drive necessary to boot systems; targeting specific critical systems as well as a large number of systems in a network to interrupt availability to system and network resources.

Endpoint Denial of Service Adversaries may perform Endpoint Denial of Service (DoS) attacks to degrade or block the availability of services to users. Endpoint DoS can be performed by exhausting the system resources those services are hosted on or exploiting the system to cause a persistent crash condition. Example services include websites, email services, DNS, and web-based applications. Adversaries have been observed conducting DoS attacks for political purposes and to support other malicious activities, including distraction, hacktivism, and extortion.

Firmware Corruption Adversaries may overwrite or corrupt the flash memory contents of system BIOS or other firmware in devices attached to a system in order to render them inoperable or unable to boot. Firmware is software that is loaded and executed from non-volatile memory on hardware devices in order to initialize and manage device functionality. These devices could include the motherboard, hard drive, or video cards.

Inhibit System Recovery Adversaries may delete or remove built-in operating system data and turn off services designed to aid in the recovery of a corrupted system to prevent recovery. Operating systems may contain features that can help fix corrupted systems, such as a backup catalog, volume shadow copies, and automatic repair features. Adversaries may disable or delete system recovery features to augment the effects of Data Destruction and Data Encrypted for Impact.

Network Denial of Service Adversaries may perform Network Denial of Service (DoS) attacks to degrade or block the availability of targeted resources to users. Network DoS can be performed by exhausting the network bandwidth services rely on. Example resources include specific websites, email services, DNS, and web-based applications.

Runtime Data Manipulation Adversaries may modify systems in order to manipulate the data as it is accessed and displayed to an end user. By manipulating runtime data, adversaries may attempt to affect a business process, organizational understanding, and decision making.

Service Stop Adversaries may stop or disable services on a system to render those services unavailable to legitimate users. Stopping critical services can inhibit or stop response to an incident or aid in the adversary's overall objectives to cause damage to the environment.

Stored Data Manipulation Adversaries may insert, delete, or manipulate data at rest in order to manipulate external outcomes or hide activity. By manipulating stored data, adversaries may attempt to affect a business process, organizational understanding, and decision making.

Access or alter stored or transmitted data and information (such as telemetry data, flight logs, audio, and video) within the UTM system and its associated infrastructure.

System Shutdown/Reboot Adversaries may shutdown/reboot systems to interrupt access to, or aid in the destruction of, those systems. Operating systems may contain commands to initiate a shutdown/reboot of a machine. In some cases, these commands may also be used to initiate a shutdown/reboot of a remote computer. Shutting down or rebooting systems may disrupt access to computer resources for legitimate users.

Transmitted Data Manipulation Adversaries may alter data en route to storage or other systems in order to manipulate external outcomes or hide activity. By manipulating transmitted data, adversaries may attempt to affect a business process, organizational understanding, and decision making.

4.14.3 Existing vectors in the ICS matrix

The difference between Denial of something and Loss of something, is that Denial is temporary loss of the resource and Loss is (semi-)permanent loss; requiring at least a reboot or manual fix to restore.

Damage to Property Adversaries may cause damage and destruction of property to infrastructure, equipment, and the surrounding environment when attacking control systems. This technique may result in device and operational equipment breakdown, or represent tangential damage from other techniques used in an attack. Depending on the severity of physical damage and disruption caused to control processes and systems, this technique may result in Loss of Safety.

This can happen by detonating explosives attached to, or built into the UAS, or by having the UAS fly into things (e.g. plane engines).

Denial of Control Adversaries may cause a denial of control to *temporarily* prevent operators and engineers from interacting with process controls. An adversary may attempt to deny process control access to cause a temporary loss of communication with the control device or to prevent operator adjustment of process controls. An affected process may still be operating during the period of control loss, but not necessarily in a desired state.

In UAS this can be done in several ways, for instance through blocking communication with the controller, by jamming the RF frequency bands, GPS/ GLONASS, or sending de-authentication packets to its Wi-Fi based communication channel.

Denial of View Adversaries may cause a denial of view in attempt to *temporarily* disrupt and prevent operator oversight on the status of a UAS's or swarm's environment. This may manifest itself as a temporary communication failure between a device and its control source, where the interface recovers and becomes available once the interference ceases.

In UAS this can be done in several ways, for instance through blocking communication with the controller, by jamming the RF frequency bands, GPS/ GLONASS, or sending de-authentication packets to its Wi-Fi based communication channel.

Loss of Availability Adversaries may attempt to disrupt essential components or systems to prevent owner and operator from delivering products or services.

Using a (targeted) Electromagnetic Pulse (EMP) in order to (temporarily/permanently) disable one or several UAS in a certain area.

UAS can also be taken out in several ways using physical force. This can be done in for instance with the use of a weapon (shooting it down, blowing it up), other UAS, hawks, nets, or grabbing/hitting them out of the air.

A Directed-Energy Weapons (DEW) is a ranged weapon that damages its target with highly focused energy, including laser, microwaves, and particle beams. There have been instances in which these have been used to damage or take-down UAS.

Loss of Control Adversaries may seek to achieve a *sustained* loss of control or a runaway condition in which operators cannot issue any commands even if the malicious interference has subsided.

Loss of Productivity and Revenue Adversaries may cause loss of productivity and revenue through disruption and even damage to the availability and integrity of control system operations, devices, and related processes.

Loss of Safety Adversaries may cause loss of safety whether on purpose or as a consequence of actions taken to accomplish an operation. The loss of safety can describe a physical impact and threat, or the potential for unsafe conditions and activity in terms of control systems environments, devices, or processes. For instance, an adversary may issue commands or influence and possibly inhibit safety mechanisms that allow the injury of and possible loss of life. This can also encompass scenarios resulting in the failure of a safety mechanism or control, that may lead to unsafe and dangerous execution and outcomes of physical processes and related systems.

This can happen by detonating explosives attached to, or built into the UAS, or by having the UAS fly into things (e.g. plane engines). Another way this could happen is when the system comes equipped with weapons, which could be activated remotely by an adversary.

Loss of View Adversaries may cause a *sustained or permanent* loss of view where the UAS equipment will require local, hands-on operator intervention; for instance, a restart or manual operation. By causing a sustained reporting or visibility loss, the adversary can effectively hide the present state of operations. This loss of view can occur without affecting the physical processes themselves.

Manipulation of Control Adversaries may manipulate physical process control within the industrial environment. Methods of manipulating control can include changes to set point values, tags, or other parameters. Adversaries may manipulate control systems devices or possibly leverage their own, to communicate with and command physical control processes. The duration of manipulation may be temporary or longer sustained, depending on operator detection.

In UAS this can be done for instance by repeating previously sent commands via a legitimate medium, directing spoofed or prepared commands (by protocol reversing or decoding) on a protocol level, or by hiding commands in legitimate controls.

By modifying or overwriting file-system parameters or the instructions being sent by a controller, the UAS deviates from its original pre-programmed flight path.

Manipulation of View Adversaries may attempt to manipulate the information reported back to operators or controllers. This manipulation may be short term or sustained. During this time the process itself could be in a much different state than what is reported.

Operators may be fooled into doing something that is harmful to the system in a loss of view situation. With a manipulated view into the systems, operators may issue inappropriate control sequences that introduce faults or catastrophic failures into the system. Business analysis systems can also be provided with inaccurate data leading to bad management decisions.

Since the view in modern UAS is streamed over a different channel that the control, it can be manipulated without impacting or even being noticed on the control channel.

Theft of Operational Information Adversaries may steal operational information on a production environment as a direct mission outcome for personal gain or to inform future operations. This information may include design documents, schedules, rotational data, or similar artefacts that provide insight on operations.

In the case of UAS or UTM systems this could mean gaining access to hardcoded or temporarily stored data on the system. This can include telemetry data, data stored within internal or external storage devices, system files and in some cases, owner and registration information.

Chapter 5

Future work

This research created an ATT&CK framework for unmanned aerial systems. However because of the limited timespan in which this research had to be conducted, future research and development is still required.

5.1 More peer reviews

As with all research, the ATT&CK framework found in this report can be improved by being peer reviewed more. The experts that were used to peer review the current version of this framework specialised in cyber security and UAS forensics. If the report is peer reviewed by experts from other sectors of the UAS industry, the content of the framework would be more reliable and would be applicable to a wider variety of adversaries.

5.2 Rewording of technique descriptions

Most technique descriptions that have been taken from the Enterprise and ICS ATT&CK matrices are still phrased in an enterprise or ICS specific way. These would have to be reworded to make the descriptions UAS and UTM specific instead.

5.3 Examples and proofs

Some of the techniques found in this report have not been verified to also work on UAS, but are assumed to work and be applicable to UAS based on their description and the UAS model I have used. Future work could include finding proofs of concept or examples of tactics being applied by adversaries in the wild to prove that these tactics actually belong in this framework.

5.4 Mitigations

Currently the unmanned aerial systems framework consists of adversarial tactics. The other ATT&CK frameworks also include mitigations for every tactic, in order to allow the systems to be defended against these types of adversaries.

Such mitigations should also at some stage be included in the framework presented in this report.

5.5 Labelling of techniques

The Enterprise ATT&CK matrix lists per adversarial techniques, both on what platforms they work and what permissions are required to perform it. This form of labelling could also be done for the tactics found in this framework. However, instead of labelling on what operating system they work, the tactics found in this framework could be categorised based on whether they work on an individual UAS or on a UTM system, and what the minimal threat level has to be to make use of the selected procedure.

5.6 Continuous development

As UAS and their security are ever changing, does this framework need to be continuously updated to accurately reflect developments in the UAS industry. Furthermore, as time passes more adversarial tactics and procedures will be discovered and these will have to be integrated into the framework.

Glossary

ADS-B Automatic Dependent SurveillanceBroadcast is a surveillance tool which gets aircraft to periodically broadcast their location, which allows them to be tracked. 16

ATT&CK MITRE ATT&CK is a knowledge base of tactics and techniques performed by adversaries based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, government, cybersecurity products, and service community. 6

BVLOS Beyond Visual Line of Sight. This is a term that is used to signify the operation of an unmanned aircraft when it flies further than the visual range of the operator, where control and communication links between the operator and aircraft are still maintained. 12

CUAS Counter-UAS, also known as a counter-drone system or counter-UAV, is a system, or series of systems, dedicated to detect and/or disable targeted UAS, which could be necessary in restricted areas. It is also capable of blocking and distorting the communication between the aerial system and the operator. 30

DEW A directed-energy weapon (DEW) is a ranged weapon that damages its target with highly focused energy, including laser, microwaves and particle beams. 49

drone An unmanned aerial aircraft is more commonly termed as drone by the media. Glossary: UAS. 6

DroneSec An independent and agnostic security company that focuses on the drone ecosystem. DroneSec offers consultancy, physical and cyber security, training and education, and threat intelligence to industry professionals and government bodies on UAS, CUAS, and UTM systems. 7, 8, 14

MitM A Man-in-the-Middle attack, is when an adversary secretly relays the communication between two parties, which allows the adversary to read or modify the data, without either party being aware of this happening. 20

MITRE MITRE is a non-profit organization that works across governments, industries and academias. MITRE is focused on solving challenges faced in the defence and aviation sectors and work in the interest of the public to discover, invent, and lead pioneering ideas into fruition. 1, 6–8, 16

Privasec A boutique information security consulting firm providing governance and technical security assurance. Privasec is one of Australia's premier PCI, IRAP, Penetration Testing and Red Team operators, with offices throughout APAC serving the ASX200. In 2018, Privasec acquired DroneSec to extend its security operations into the unmanned systems space. 8

Radboud Univeristy Radboud University is a state funded, research oriented university located in Nijmegen, the Netherlands. The university has been included in the top 150 of universities in the world by four major university ranking tables. This document's author is one of the 22,000 students currently enrolled at this university. 7

RPAS Remotely Piloted Aircraft System. A term that is used by the International Civil Aviation Organisation (ICAO). It refers to the unmanned aircraft, the ground control systems, communications, support equipment required for the operations of the unmanned aircraft. 6, *Glossary:* UAS

RTH A Return-to-Home function automatically flies the UAS to a preset location. Usually this function is automatically activated when connection with the operator is lost. 45, 46

SDR A Software Defined Radio is a radio communication system where traditional hardware components are replaced with software simulated parts on an embedded system. This allows for a more flexible radio communication system, as well as more up and downstream possibilities[15]. 13

SMB Server Message Block is a protocol that is a network protocol for providing shared access to files between systems within a network. 17

UAS Unmanned Aerial System is an all encompassing term for everything that makes an aerial drone operate. It refers to the unmanned aircraft, the ground control systems, communications, and support equipment required for the operations of the unmanned aircraft. 6, 10, 11

UAV Unmanned Aerial Vehicle. A UAV refers to the unmanned aircraft, or drone, itself. This variant of the name for UAS is widely used in the military. 6, *Glossary:* UAS

UTM A UAS Traffic Management system provides safe and effective control for UAS to operate in mixed airspace BVLOS with other low-altitude aircraft. It is an air traffic management system which can manage and control a variety of brands, models, and types of unmanned aerial systems. The focus of UTM systems is the digital sharing of unmanned planned flights, and the real-time change of those flight paths if required for safety or collision prevention purposes. 8, 12

Acronyms

ADS-B Automatic Dependent SurveillanceBroadcast. 16, *Glossary:* ADS-B

APT Advanced Persistent Threat. 15

ARP Address Resolution Protocol. 20, 32

ATT&CK Adversarial Tactics, Techniques, and Common Knowledge. 1, 6–8, 10, 16, 52, 53, *Glossary:* ATT&CK

BVLOS Beyond Visual Line-of-Sight. 12, *Glossary:* BVLOS

CUAS Counter-UAS. 30, 41, *Glossary:* CUAS

DEW Directed-Energy Weapons. 49, *Glossary:* DEW

DNS Domain Name System. 38

DoS Denial-of-Service. 41

EMP Electromagnetic Pulse. 42, 43, 49

FTP File Transfer Protocol. 33, 38

GLONASS Global Navigation Satellite System. 12, 42, 49

GPS Global Positioning System. 12, 42, 49

GUI Graphical User Interface. 19, 30

HTTP Hypertext Transfer Protocol. 38

HTTPS Hypertext Transfer Protocol Secure. 38

ICMP Internet Control Message Protocol. 39

ICS Industrial Control Systems. 1, 6, 7, 10, 11, 18, 20, 23, 24, 28, 30, 33, 39, 41, 42, 46

IoT Internet of Things. 8

IP Internet Protocol. 31

MBR Master Boot Record. 20, 47

MitM Man in the Middle. 20, *Glossary:* MitM

NFZ No Fly Zone. 41

OS Operating System. 11

RAM Random Access Memory. 11

RF Radio Frequency. 13, 16, 29, 30, 42, 49

RPAS Remotely Piloted Aircraft System. 6, 7, *Glossary:* RPAS

RTH Return-to-Home. 45, 46, *Glossary:* RTH

SDR Software Defined Radio. 13, 17, 30, *Glossary:* SDR

SMB Server Message Block. 17, *Glossary:* SMB

SMTP Simple Mail Transfer Protocol. 38

SMTPS Simple Mail Transfer Protocol Secure. 38

SOCKS Socket Secure. 39

SOL Serial over LAN. 39

SQL Structured Query Language. 17

SSH Secure Shell. 17, 34

UAM Urban Air Mobility. Another name for a UTM system. 12, *Glossary:* UTM

UAS Unmanned Aerial System. 1, 6–18, 20, 21, 23–25, 28–30, 33, 34, 36, 39, 41–46, 49–53, *Glossary:* UAS

UAV Unmanned Aerial Vehicle. 6, 7, *Glossary:* UAV

UDP User Datagram Protocol. 39

UI User Interface. 28

UTM UAS Traffic Management. 1, 8, 10, 12, 13, 16–18, 20, 21, 23–25, 28, 30, 33, 34, 36, 39, 41, 43, 44, 46, 48, 51, 53, *Glossary:* UTM

VBR Volume Boot Record. 20

VME Virtual Machine Environment. 28, 32

Bibliography

[1] K. Best, J. Schmid, S. Tierney, J. Awan, N. Beyene, M. Holliday, R. Khan, and K. Lee, "How to analyze the cyber threat from drones: Background, analysis frameworks, and analysis tools," *Homeland Security Operational Analysis Center operated by the RAND Corporation*, 2020.

[2] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab, "Security analysis of drones systems: Attacks, limitations, and recommendations," *Internet of Things*, vol. 11, May 2020.

[3] "MITRE ATT&CK matrices." `https://attack.mitre.org/matrices/`. Accessed: 24-03-2020.

[4] A. Navas, "Measuring pasture intake with drones." `https://www.farmonline.com.au/story/6687564/measuring-pasture-intake-with-drones/`, March 2020. Accessed: 04-06-2020.

[5] D. Ghosh, "Kolkata cops to get 8 drones to keep vigil on protests." `https://timesofindia.indiatimes.com/city/kolkata/kolkata-cops-to-get-8-drones-to-keep-vigil-on-protests/articleshow/74485365.cms`, March 2020. Accessed: 04-06-2020.

[6] M. Jarram, "Drones used during police chase through streets and alleyways." `https://www.nottinghampost.com/news/nottingham-news/drones-used-during-police-chase-3900404`, February 2020. Accessed: 04-06-2020.

[7] UAV Coach, "How drones work." `https://uavcoach.com/wp-content/uploads/2016/09/How-Drones-Work-1.png`. Accessed: 11-05-2020.

[8] K. W. Chan, U. Nirmal, and W. G. Cheaw, "Progress on drone technology and their applications: A comprehensive review," *AIP Conference Proceedings 2030*, 2018.

[9] J. Flint, "What sensors do drones use." `https://3dinsider.com/drone-sensors/`, Apr 2019. Accessed: 11-05-2020.

[10] NASA, "What is unmanned aircraft systems traffic management." `https://www.nasa.gov/ames/utm/`. Accessed: 19-05-2020.

[11] DroneSec, "DroneSec Notify incident platform." `https://dronesec.com/notify`. Accessed: 28-04-2020.

[12] MITRE, "ATT&CK Matrix for Enterprise." https://attack.mitre.org/matrices/enterprise/. Accessed: 25-03-2020.

[13] MITRE, "ATT&CK matrix for Industrial Control Systems (ICS)." https://collaborate.mitre.org/attackics/index.php/Main_Page. Accessed: 06-04-2020.

[14] Z. Whittaker, "Security flaw in DJI's website and apps exposed accounts to hackers and drone live feeds." https://techcrunch.com/2018/11/08/security-flaw-in-dji-apps-exposed-accounts-to-hackers-and-drone-live-feeds/. Accessed: 03-04-2020.

[15] H. Sims, "Software defined radios - architectures, systems and functions." https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170005302.pdf. Accessed: 08-06-2020.