17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID # 381

Efficient Ground Segments Protection against Advanced Persistent Threats

Julio Vivero[a]*

[a] *Business Partner in International Markets, GMV, Isaac Newton 11, P.T.M. 28760 Tres Cantos, Madrid,* jvivero@gmv.com
* Corresponding Author

Abstract

Ground Segments Security has become in the last years a mandatory element. Even more, this is often derived from mandatory regulations the ground segments need to comply with.

Antivirus are typically installed in Ground Segments in an attempt to comply against these regulations or to give a false sense of security. However antivirus solutions are not only useless but also annoying. Useless because if someone wants to attack a ground segment they would never use a known malware which can be detected by antiviruses. Instead, a targeted malware would be custom developed to avoid detection and execute the targeted malicious activities. Antivirus solutions would not detect such malware so they are useless to protect our ground segments. On top of that, they are a nuisance as they need to be regularly, at least weekly, updated with new signatures. This task requires special operational procedures to move the file from the Internet into the ground segment network and distribute it to the antivirus clients.

Checker, or Checker Satellite Security, is a whitelisting solution far more adequate than antiviruses to efficiently protect ground segments against advanced threats. They change the protection paradigm from a known bad to a known good. That is, unlike antiviruses, only approved operations (i.e. known good) are allowed and all the rest is discarded. Antiviruses allow everything except for known bad. This includes not only running processes in the systems, but many other elements such as incoming and outgoing connections, integrity of system and configuration files, library calls, encryption, use of external drives or USB devices to efficiently protect the ground segment against a wide range of threats. The learning process of the known-good behaviour is easily done during the validation process of every new release from the central Checker management console, The Checker central management console simplifies the process of defining the authorised elements and assigning them to the ground segment systems.

Space Norway realised the potential of Checker to provide real protection to the Ground Segment when compared with Antivirus solutions. The paper will describe the Checker solution, how it has been used to address the security needs of Space Norway and the benefits obtained.
Keywords: Cybersecurity, Whitelisting, Hardening, Malware

1. Introduction

The importance of space in communications, navigation, earth observation and many other fields is enormous for our society in civil and military terms. The conflict of Ukraine has been a wide opener for everybody on the importance of space-based services [1,2]. This reliance on space services provokes a growing interest in attacking them for ransom, hacktivists or geopolitical advantage among others. The threats to space and ground segments are both cyber and physical [3, 4].

Potential threat actors to space assets have top skills and capacities at hand. Organizations wishing to counter such skills need top cybersecurity solutions fully tailored to the particularities of space organizations and assets.

It is relatively common practice to introduce generic technical security controls into information systems without having previously specified a security policy. This may be partially due to the fact that it is not possible to define policies for complex environments (since they require an analysis of the cost and benefits as opposed to the inconveniences and limitations involved) and also to the fact that it is not easy to decide which controls are appropriate to guarantee compliance with policies. It is therefore not surprising that controls requiring heavy investments are often implemented without previously determining whether they are the most appropriate and why.

An endpoint (in a satellite control context) is a very specific environment that has a clear advantage over other endpoint environments from a security point of view: its configuration (programmes, files and resources required) and its behaviour (e.g., reading, writing or running permission) can be clearly defined and generally remain stable (with few changes) for longer periods of time than office automation environments. This configuration control and stability make it possible to clearly define security policies.

Checker takes advantage of this situation, one of its fundamental objectives being closing the gap between the definition and management of security policies for the endpoint network on the one hand and the security controls implemented in the endpoint on the other. Checker provides an environment that makes it possible to:

- Generate security policies easily,
- Automatically establish in the endpoint the controls needed to ensure compliance and
- Monitor compliance remotely and centrally

For Checker, a security policy is a set of rules for accessing and using resources that can be structured and stored in a set of files that is sent to the endpoints. It is not an abstract concept but rather a very specific concept that provides endpoints security in a manageable and comprehensible way. The rules are diverse. Some examples of the rules allowed would be “no process except myproces can modify file myfile.dat”, “only Java class myjavaapp.class can access the library mylibrary.so” or “no system external to the endpoint can access comm port 12345”.

Checker is a cybersecurity product specifically designed for controlling the operating system providing a high security environment for ground systems and can be simply and centrally managed.

2. Security Policies and Controls

The security management concept supported by Checker revolves around the basic concept of a “security policy”.

Generally speaking, we could define a security policy as a set of rules, principles and practices that determine the manner of implementing and managing security in an organisation or in a particular environment. Once a security policy has been established, it is possible to design and implement a set of mechanisms known as security controls, whose purpose is to ensure that the policies are adhered to and that any non-compliances are detected and properly managed.

Generally speaking these controls can be technical, administrative or procedural in nature. Typical examples of technical security controls are firewalls, anti-virus or intrusion detection systems.

Checker enables the creation of security policies and implements a versatile security control system (Fig. 1).

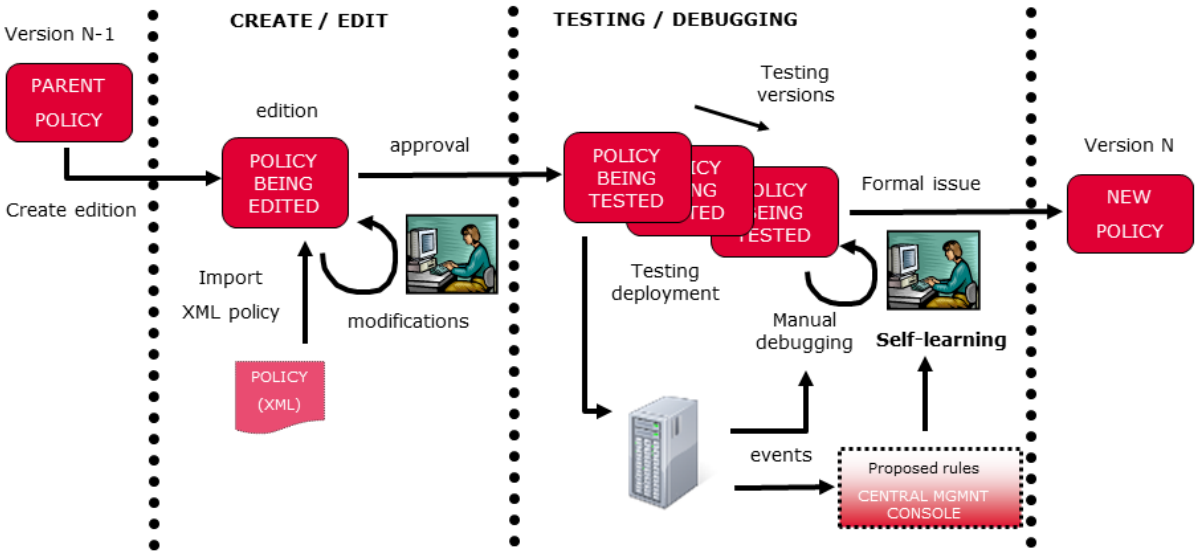

Fig. 1. Policy lifecycle

3. An adaptable Security Control

Checker provides an adaptable security control in the form of a software agent that is installed in every endpoint and enforces all aspects of the security policy. The figure below summarizes protection capabilities of the checker agent.

The fact that one single agent enforces all security has a number of advantages, because some policies are meaningful when addressing more than one aspect of security (e.g. processes running and processes communication). Usually other solutions deal with different aspects with different pieces of software (usually for historical development reasons) which is a limitation both from a security and a management point of view.

As can be seen in Fig. 2 Checker provides a full fledge of security controls for the endpoints.

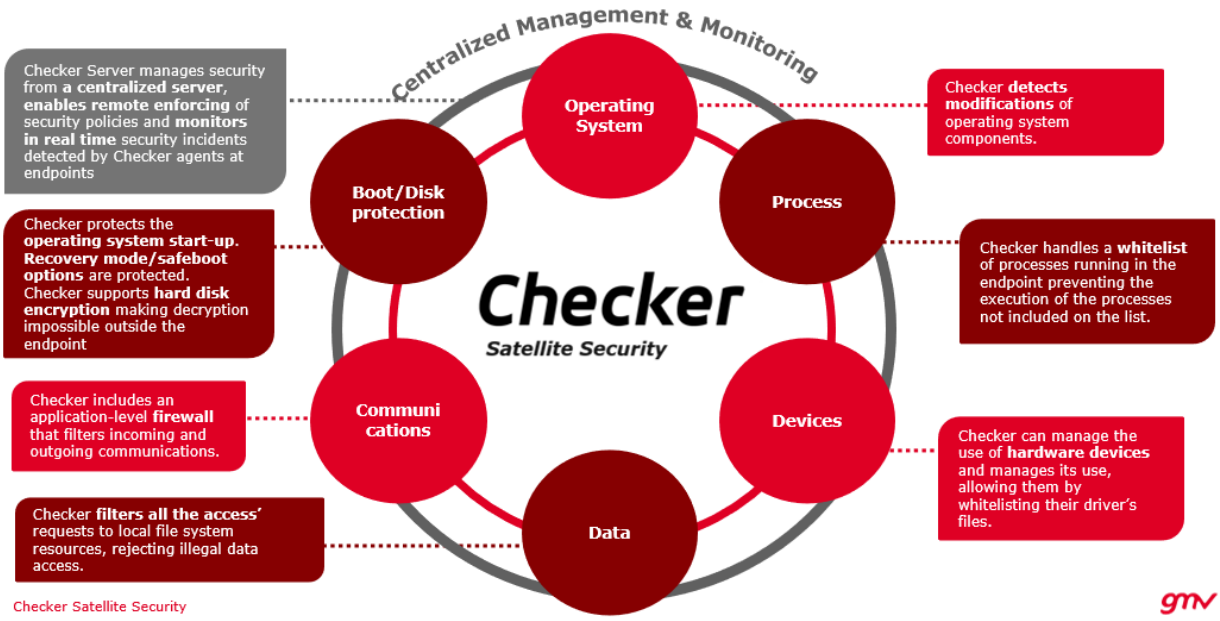

Fig. 2. Checker Satellite Security Protection

At the level of the Operating System Checker permits the enforcement of the following security controls:

- Integrity Audit: Integrity validation of critical operating system processes and resources. Any detected alteration will be reported.
- Resources use protection: Use of operating system resources (files, directories, libraries, drivers…) is granted or denied on a process basis. Non permitted use of resources will be blocked and reported.
- Multi operating system support: Checker is compatible with Windows and Linux operating systems and can be installed on top of the most popular Windows and Linux distributions. Integrity validation supports different versions of the same (authorized) program or resource.

Checker protects the execution of processes through:

- Whitelisting: List of permitted and approved processes. Processes not included in this list will be blocked and reported. **Self-learning feature** allows whitelist creation in a matter of minutes.
- Integrity Protection: Integrity validation of whitelisted process and its resources. Processes or resources illegally altered will be reported.
- Multi-version resource's support: Use of local resources (files, directories, libraries, drivers…) filtered by whitelisted process. Non permitted use of resources will be blocked and reported.

Endpoint devices are also protected by Checker by means of the following security controls:

- Hardware protection: The connection and use of new hardware is allowed or denied if the corresponding hardware device driver is listed. Unauthorized devices will be blocked and reported.
- Device access control: Use of connected hardware filtered on a process-by-process basis. Non permitted use of connected hardware will be blocked and reported
- USB flash drives control: Reliable control of authorized USB drives for easy, yet secure maintenance. Supports encryption of the USB data to protect its content and supports user authentication for authorizing USB drives.
- Keyboard and mouse control: Control the use of keyboard's keys and mouse's clicks.

Security Controls over data are also another crucial point in Checker:

- File System protection: Access to local files and directories is granted or denied on a process basis. Illegal access to restricted data will be blocked and reported.
- Integrity Protection: Integrity validation of sensitive data files. Data illegally altered will be reported.

Incoming and outgoing communications to and from the endpoint are security controlled in Checker. Filtering of incoming and outgoing communications by protocol, port, address and local process provides high level firewalling functionality matching the process whitelist. Non permitted communications will be blocked and reported.

Checker protects as well the boot process and the endpoint disk to avoid tampering with the OS before the boot:

- Operating system boot protection: Checker protects the operating system start-up with PBA (Pre-boot Authentication) based on a PIN. Recovery mode is inaccessible unless PIN is introduced. It prevents from booting into recovery mode that allows the modification of operating system files.
- Full Hard Disk Encryption: Hard disks encryption can be fully managed from the Checker console. This includes sending remote commands to encrypt or decrypt the endpoints' disks. Key management is automatic and transparent to the user.
- Zero Downtime: The encryption can be commanded from the server and the disk can undergo the encryption process while the endpoint keeps operating.
- Smart Environment detection: Endpoint environment is defined as a configurable combination of identifiers associated to the endpoint hard disk, the endpoint hardware and the endpoint network.

All the above security controls are centrally managed and monitored through the Checker Server. The Checker Server manages all Checker agents (endpoints) in the network. It provides the Checker console from where the operator can enforce new security policies (which includes whitelists, firewalls rules and more), manage security execution mode, request remote system shutdowns, etc. The Checker console includes a complete and intuitive security policy editor. Learning mode allows easy and fast security policies creation either from scratch or from already existing policies by using trusted endpoints. It also allows version control including visualization and modification of predesigned security policies. Finally, the Checker Server receives security events generated by agents in real time. These events are shown in a security dashboard, enabling easy security monitoring of the whole endpoints network. Additionally, Checker continuously monitors the integrity of all the endpoints in the network as well as any policy violation attempts and immediately alerts about these events. Security events can easily be integrated into third party monitoring systems.

3. Architecture

Checker is conceived as a distributed security administration and management environment. Its basic components are Checker Agent and Checker Server.

- Checker Agent is the software that is installed in the network's endpoints. It is a software programme with a very small footprint (i.e. occupies little space and consumes few resources). The basic purpose of the agent is to ensure compliance with the policies defined for each endpoint. The actions allowed by the policy are carried out as requested but actions that are not allowed are intercepted, prohibited and reported. The agents report attempts to violate policy in the form of real time events (alarms) to the Checker server.
- Checker Server is connected to all endpoints. The server provides the ability to create security policies and incorporates a series of aids that make this task easier and more manageable. The server can deploy different policies for different endpoints or groups of endpoints, determining when they should be applied. In addition, the server provides a monitor for tracking the security events received by the endpoints as well as a status table for rapidly identifying events in large networks (several thousand endpoints). It is also possible to view the history of events received.

The agents and server communicate with one another over the endpoint's IP network. This communication includes:

- Automatic detection of endpoint network presence and agent status.
- Sending policy files. Essentially, a policy is defined by a set of files (known as ACL files or simply ACL) that is sent from the server to the endpoint. There, the agent interprets and monitors each action that takes place in the endpoint and either allows or denies the action.
- Sending security events (alarms) from agents to server.
- Sending administrative commands to change the behaviour or configuration of the Checker Agent.

All the information exchanged by the server and the agents is encrypted for enhanced security.

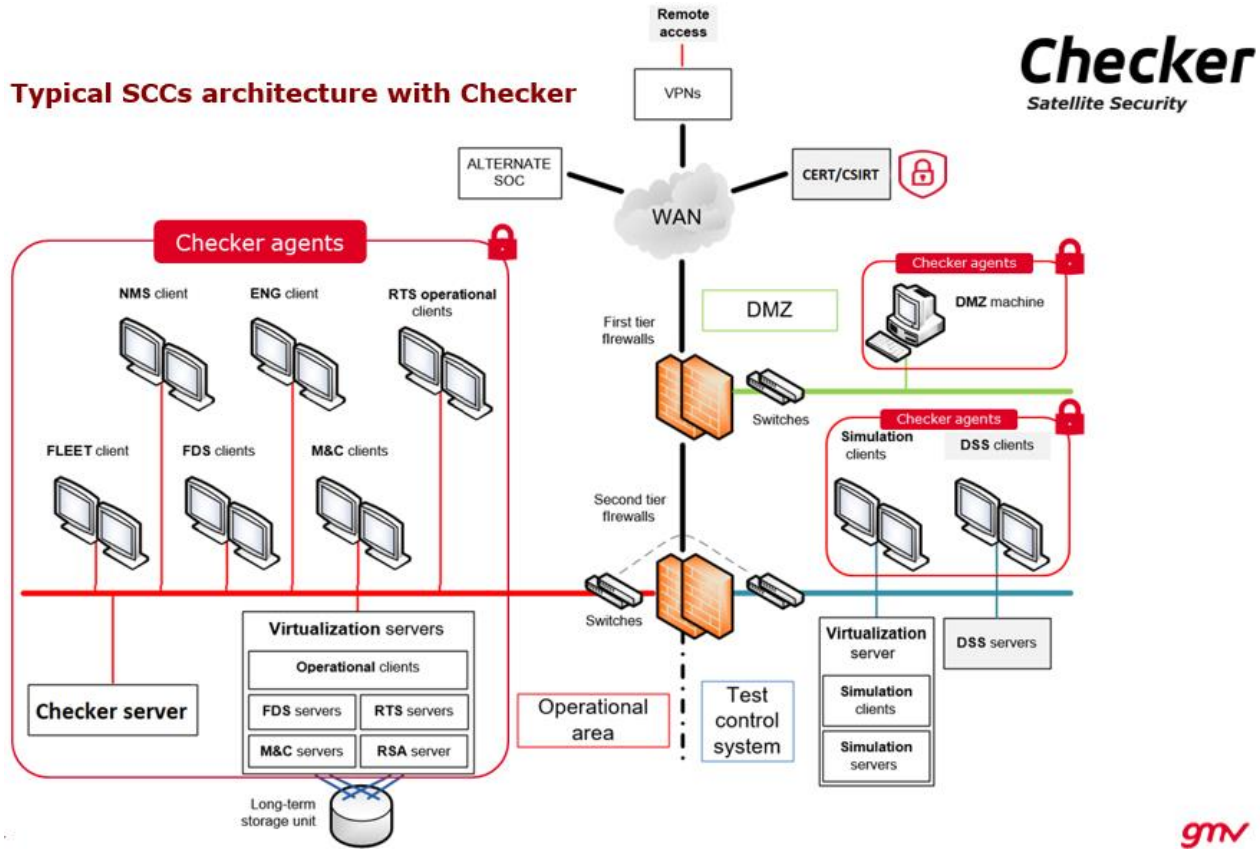

Fig. 3. Typical SCCs architecture with Checker

On Space Ground Segments two different types of endpoints are considered: Servers and Workstations. They perform different tasks and run under different environments and conditions, so in turn they require different security measures to be set up. According to this system segregation, Checker adapts its security controls as follows:

- Servers Endpoints: Server endpoints perform well-known stable functions that will be defined on the Checker security policy. They are placed in quite isolated environments and they are normally hardened. Physical access to these machines is highly restricted. This type of endpoints does not require software-based disk encryption and USB devices protections. Availability and reliability are essential in this type of endpoints.
- Workstations Endpoints: Workstations endpoints perform specialized functions that will be defined on the security policy with more flexibility than in Servers endpoints. Workstations endpoints security require hard disk encryption as regular physical access is expected. Hardware connections should be also controlled because these endpoints are not usually isolated.

Checker agent can be configured remotely from the server and work in five modes:

- ACTIVE mode, when all of the checker's functionality is available. This is normal operating mode for Checker in a working endpoint.
- INACTIVE mode. When Checker agent is deactivated, policy control is disabled and the endpoint can function freely, without limitations, as though Checker were not installed. However, communications are maintained between the agent and service for possible administration and deployment of policies and activation of the agent.
- LEARNING mode. This mode is used for tests and for debugging new policies. Its purpose is what shall or shall not be included in the Security Policy.
- OPERATOR mode. This mode is mainly used for allowing local interventions. During a (programmable) short time the security enforced by the ACL is lowered down. All actions are still monitored and reported to the central server.
- CRITICAL mode. This mode is set when the Agent detects integrity violations of the Checker system, normally due to failures in the validation of resources of the Agent. In this case, the Agent generates an

alarm that is sent to server and triggers a set of actions to recover the endpoint integrity as soon as possible.

6. Conclusions

Checker Satellite Security is designed to perfectly fit space ground segments security particularities removing the drawbacks of other typical security safeguards, namely:

- It allows a centralized management and monitoring of the whole range of security controls for the Space Ground Segments.
- It removes the need of manual procedures or external connections because no virus signatures or rules need to be downloaded from external providers.
- Lowers patching criticality and urgency. The reason is that much less services potentially vulnerable will be running on systems, and even if they are compromised the attacker will be completely tied in the actions he or she can execute in the system (only those permitted by the process whitelisting policy).

However, this solution does not only remove these drawbacks but also introduce some additional benefits:

- It does not only protect the infrastructure against malicious activities but also from unintended operational mistakes as each endpoint security policy restricts permitted actions to that corresponding to the expected functionality of the endpoint.
- Antivirus, antispyware or any other kind of malware protection software would no longer be required in your environment simply because, independently of the infection vector and independently of the malware sample (no matter whether new or old) will not be allowed to run within space systems protected with a process whitelisting solution.
- The security policy learning and definition process provides full visibility of what is happening at the ground segment, often uncovering unknown or unexpected behaviours.
- Finally, but no less important, system performance and availability properties is also enhanced because no resources are being consumed by unnecessary services and the number of anomalies caused by unpredicted software interactions are reduced.

References

[1] S. Erwing, Cyberwarfare gets real for satellite operators, SpaceNews, March 20, 2022.
Reference to a conference/congress paper:
[2] J. Beale, Space, the unseen frontier in the war in Ukraine, October 6, 2022.
Reference to a book:
[3] T. Harrison, K. Johnson, M. Young, N. Wood, A. Goessler, Space Threat Assessment 2022, CSIS Aerospace Security Project, April, 2022.
[4] B. Unal, International Security Department Chatham House, Cybersecurity of NATO's Space-based Strategic Assets, July 2019