ACME Water Case Study

Shamal Faily

April 2021

ACME Water Case Study

Background

The security of the environment around Control Systems, operational personnel, sites, assets, activities, information, technological resources, and services has taken on increased importance at ACME Water for the following reasons:

- The disappearing network boundary: a highly mobile workforce works on assets and systems deployed over a diverse and challenging geographic terrain. Information Technology enables this but, in doing so, needs to provide controlled access to core applications, allowing multiple protocols access through the perimeter, reducing perimeter controls, allowing partners to deliver contractual obligations on time and to cost.
- The business climate is highly regulated, and delivery of Wastewater and Cleanwater services are critical to our customers. It is imperative ACME employees are made aware of the importance of protecting our assets, information, and reputation to meet this end.
- Law requires companies to institute reasonable, effective and consistent controls designed to prevent the disclosure and falsification of information, the safety of personnel, sites, and the protection of technological resources.
- Rapid changes face ACME in the use and dependence of computers and network technologies. It is necessary and reasonable to expect everyone to apply the proper methods of handling and safeguarding information and managing computer and network resources.

This case study is organized into ten separate electronic subfolders which contain an Exercise document with scenario and questions, a CAIRIS file and/or other applicable files, and a worked answer.

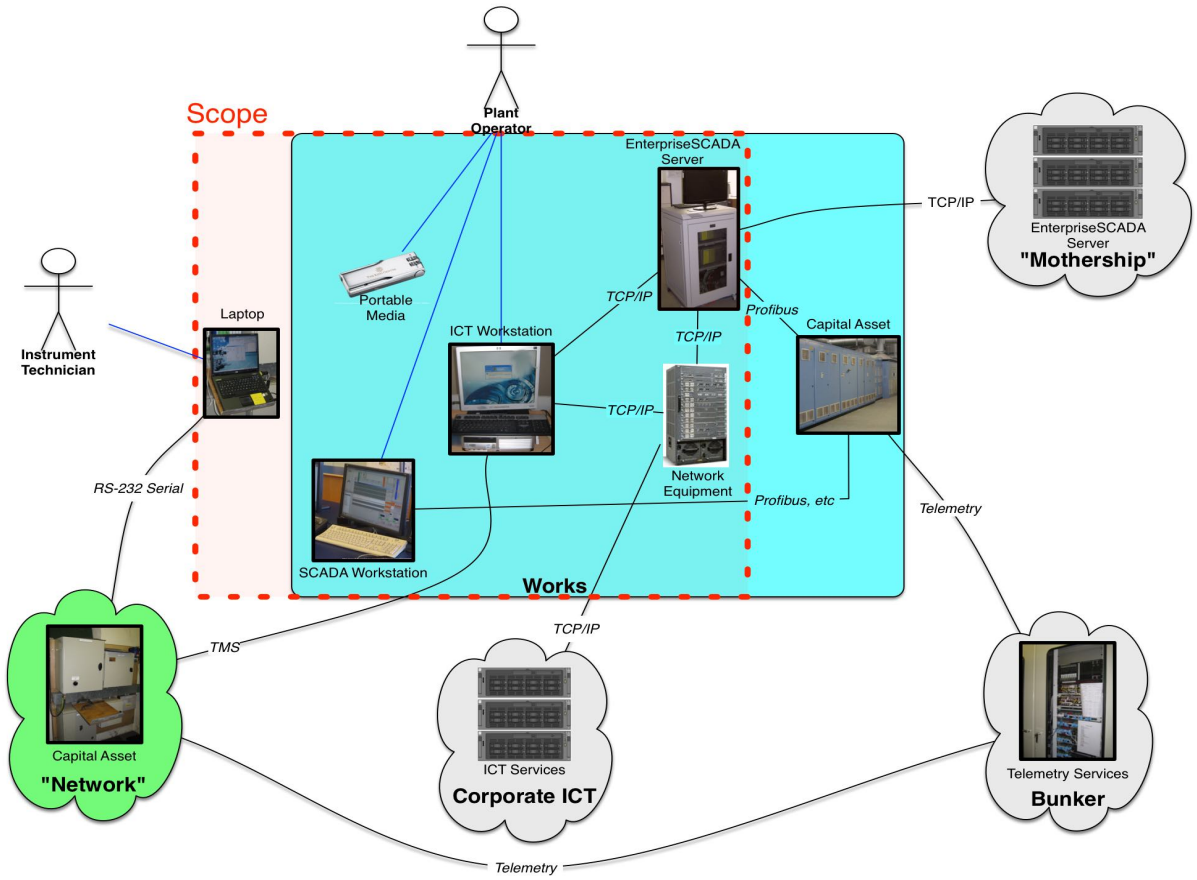
Computer Aided Integration of Requirements and Information Security (CAIRIS) is an open-source platform for building security and usability into your software. More information can be found here: https://cairis.readthedocs.io/en/latest/gettingstarted.html

The *ACME Water Requirements Specification* document is included in the case study materials to provide context for the exercises. This document includes additional information on the Environment, Personas, the Rick Argumentation Model, Document References, External Documents, Mandated Constraints, Naming Conventions, Assets, Corporate Network, Tasks which the planned system will need to be designed for, Use Cases, Responsibilities, Vulnerabilities, Attacker Profiles, Threats, Risks, Misuse Cases, and Functional Requirements.

Case Study Overview

Provide a secure operating environment for SCADA, Telemetry and Control Systems associated with assets owned and operated by ACME. The following, *Figure 1,* illustrates the scope of the environment.

Figure 1, Scope of Work in the Context of the Overall Environment



Student Instructions

Sample Instructions for Exercise #1: Introduction & Human Error

Instructions for *Exercises 1-10* can be found in their respective subfolders for this case study.

Scenario

You are a security architect at ACME Water: a water company that delivers clean and waste water services to Dorset. You have been asked to evaluate the design of a software repository used by instrument technicians at ACME to manage control software. The design of the repository is based on the following scenario:

Barry is an instrument technical at ACME water. Barry goes into the depot on a Monday morning, batch syncs his laptop. This involves plugging his laptop into the telemetry network, looking at what files have changed, and making sure he has the latest programs his area. He then picks up his schedule jobs for the rest of the week. As luck would have it, his first scheduled job is at the depot.

Barry walks 100 yards to the motor control center, locates the telemetry outstation, plugs his laptop into the outstation and loads up the program. Barry verifies his software matches up with the same software on the outstation; this is done automatically.

Barry then makes the relevant changes and commissions the change. In this case, Barry calls up the control room to make sure an alarm has been raised based on the new element setup.

Barry then saves the change to the outstation and his laptop. The software tool displays the changes and asks for verification. A software change alarm is then generated automatically and sent through to both the telemetry alarm page and the software repository.

Barry will commit this change back to the repository "as soon as he can." At the end of the day, Barry returns to a depot, fills in his paperwork and batch syncs to the repository.

Questions for Exercise #1

1. In groups, use CAIRIS to create an asset model associated with this scenario. Assign the security properties that need to be protected for each asset, together with justification for each property.

   You may find it useful to sketch your model on paper before a scribe enters the data into the platform. Alternatively, diagrams.net could be used to create the diagram, which can then be imported into CAIRIS.

2. Exchange your asset model with another group. Assess the asset model for opportunities for exploitation. To help you, you should score the model based on the assessment criteria provided.

3. In groups, use CAIRIS to create a use case for 'modifying telemetry software' based on this scenario. The actor should be 'Instrument Technician' and the system is the software repository for storing control software.

4. Exchange your use case with another group. Assess the use case for opportunities of human error leading to exploitation. To help you, you should score the model based on the assessment criteria provided.

Instructor notes

This case study and the series of all ten exercises can be used for final year undergraduate and early master-level student courses.

All ten exercises are typically used over the course of one semester.

Students will be using CAIRIS to create models.

Example solution

Sample Worked Answer For Exercise #1

1. In groups, use CAIRIS to create an asset model associated with this scenario. Assign the security properties that need to be protected for each asset, together with justification for each property. You may find it useful to sketch your model on paper before a scribe enters the data into the platform. Alternatively, diagrams.net could be used to create the diagram, which can then be imported into CAIRIS.

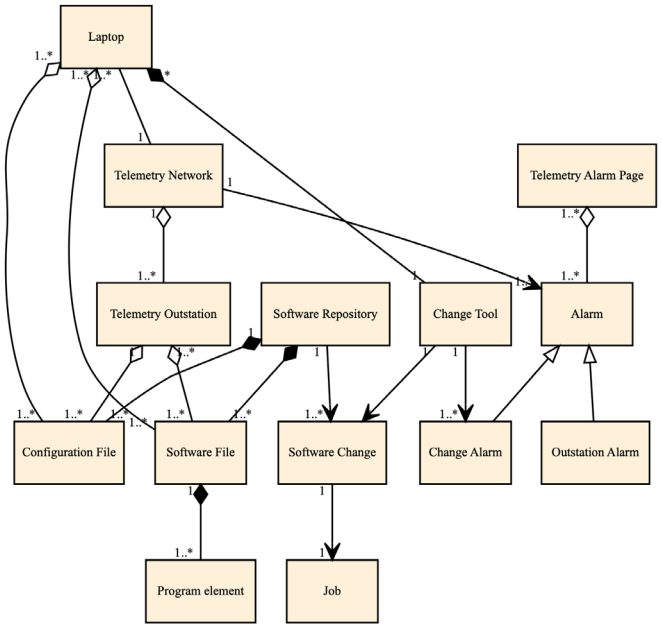
Here is a possible model based on one interpretation of the scenario:

Here are some examples of security properties associated with each asset. Integrity, Accountability, and Availability seem to be the dominant concerns here, but the values aren't always what you might expect given the criticality of the water infrastructure. Can you think of why this might be?

(Clue: Think about what constitutes Low, Medium, or High values)

Asset	Property	Value	Rationale
Laptop	Accountability	Medium	Whoever has laptop has access to files
Telemetry Network	Integrity	High	Potentially catastrophic results if alarms or config data spoofed or tampered with
Telemetry Network	Availability	High	Compromised accessibility affects technician performance.
Configuration File	Integrity	Medium	Tampering with the configuration file impacts ability to report problems with attached equipment.
Configuration File	Accountability	Low	Useful to know who made changes to configuration file.
Job	Integrity	Low	Jobs should be authorised
Job	Accountability	Low	The assigned instrument technician is responsible for a specific job, and accountable should any issues arise because of it.

Asset	Property	Value	Rationale
Telemetry Outstation	Availability	Medium	Responsible for water distribution and treatment.



2. Exchange your class model with another group. Assess the class model for opportunities for exploitation. To help you, you should score the model based on the assessment criteria provided.

Here are some assessment criteria you can use:

- *What assets are missing or incorrect?*
- *What security properties which have not been considered?*
- *What security properties appear unjustified?*
- *What associations missing or incorrect?*
- *What associations ambiguous ?*

3. In groups, use CAIRIS to create a use case for ‘modifying telemetry software ’based on this scenario. The actor should be ‘Instrument Technician ’and the system is the software repository for storing control software.

These are some of things one might expect to see in this use case. The use case steps themselves aren't that involved, but there are quite a few pre- and post-conditions, and lots of scope for things to go wrong - even if the technician is motivated and not under stress. If the context is modified such that the technician is non-motivated, tired, under pressure, etc, some of these exceptions could lead to cases for human error. For example, the wrong information or software might be included as a result of 'slip' if the modification follows shortly after several other modifications based on different software or configuration files. Alternatively, the technician might make a 'mistake' if he didn't believe that submitting the software modification report was necessary to commit the software change on the repository. Omitting that step might also be intentional if the technician is in a hurry and means to 'do this later.'

Name	Modify Telemetry Software
Actor/s	Instrument Technology
Pre-Conditions	• Software repository online • Alarm mechanisms online • Instrument Technician authenticated with Outstation • Instrument Technician authenticated with Telemetry Network • Modified software file on laptop and outstation
Steps	• Instrument Technician requests verification of software change • System sends software change alarm to Telemetry Network • System displays modified software changes to Instrument Technician • Instrument Technician submits software modification report to the system • System acknowledges software modification
Post-Conditions	• Modified software alarm received by Telemetry Network • Modified software on software repository • Software modification report on software repository
Potential Exceptions	• What if the software repository becomes unavailable? • What if the wrong software is included in the software modification form? • What if the wrong information is included in the software modification form? • What if the system acknowledgement is misinterpreted by the Instrument Technician?

4. Exchange your use case with another group. Assess the use case for opportunities of human error leading to exploitation. To help you, you should score the model based on the assessment criteria provided.

Here are some assessment criteria you can use:

- *What actors have not been considered?*
- *What goals are missing, incorrect, or ambiguous?*
- *What steps are missing, incorrect, or ambiguous?*
- *What human errors are associated with each step?*

- *What pre-conditions are missing, incorrect, or ambiguous?*
- *What post-conditions are missing, incorrect, or ambiguous?*

References

Faily S., Stergiopoulos G., Katos V., Gritzalis D. (2016) "Water, Water, Every Where": Nuances for a Water Industry Critical Infrastructure Specification Exemplar. In: Rome E., Theocharidou M., Wolthusen S. (eds) *Critical Information Infrastructures Security. CRITIS 2015*. Lecture Notes in Computer Science, vol 9578. Springer, Cham. https://doi.org/10.1007/978-3-319-33331-1_20

Faily, S., Lykou, G., Partridge, A. (2016). "Human-Centered Specification Exemplars for Critical Infrastructure Environments". *BSC Learning and Development LTD.,* In Proceedings of British HCI 2016 Conference Fusion, Bournemouth, UK.