Security Requirements Taxonomy

Revision

Version 4
10/21/22 6:34 PM

Author

Charles Wilson

Abstract

This document describes a taxonomy for classifying cybersecurity requirements.

Motivation

This document is motivated by the need to have a systematic approach for classification of cybersecurity requirements.

License

Overview

In order to create security requirements that have sufficient specificity to allow for *risk*, *development*, and *test* to properly consider them within the context of their respective areas; it is helpful to decompose them in three dimensions. These are:

- property (why)
- asset (what)
- layer (where)

We will see that this decomposition provides us a mechanism to allow for each of the aforementioned groups to consider the requirements in an unambiguous manner.

Granularity

The granularity within each of the three dimensions is arbitrary. It is intended to allow for a meaningful decomposition of a security requirement. The ones presented here are those I believe to be useful to the task. In the case of the property dimension, the UNECE values were used. It should be noted that the choice of including both authentication and authorization within the property dimension may lead to requirement duplication in the space. This is because within the world of embedded software, authentication and authorization tend to collapse into a single attribute. As the sophistication of the operating environment increases, these attributes become distinct and therefore should be addressed with separate security requirements. Similarly, assets are highly dependent upon both the domain and platform. The choice of layers is based on a TCP/IP-centric view of the world. There is an argument to be made for a layer between the protocol and application layers where services would be considered.

Property (Why)

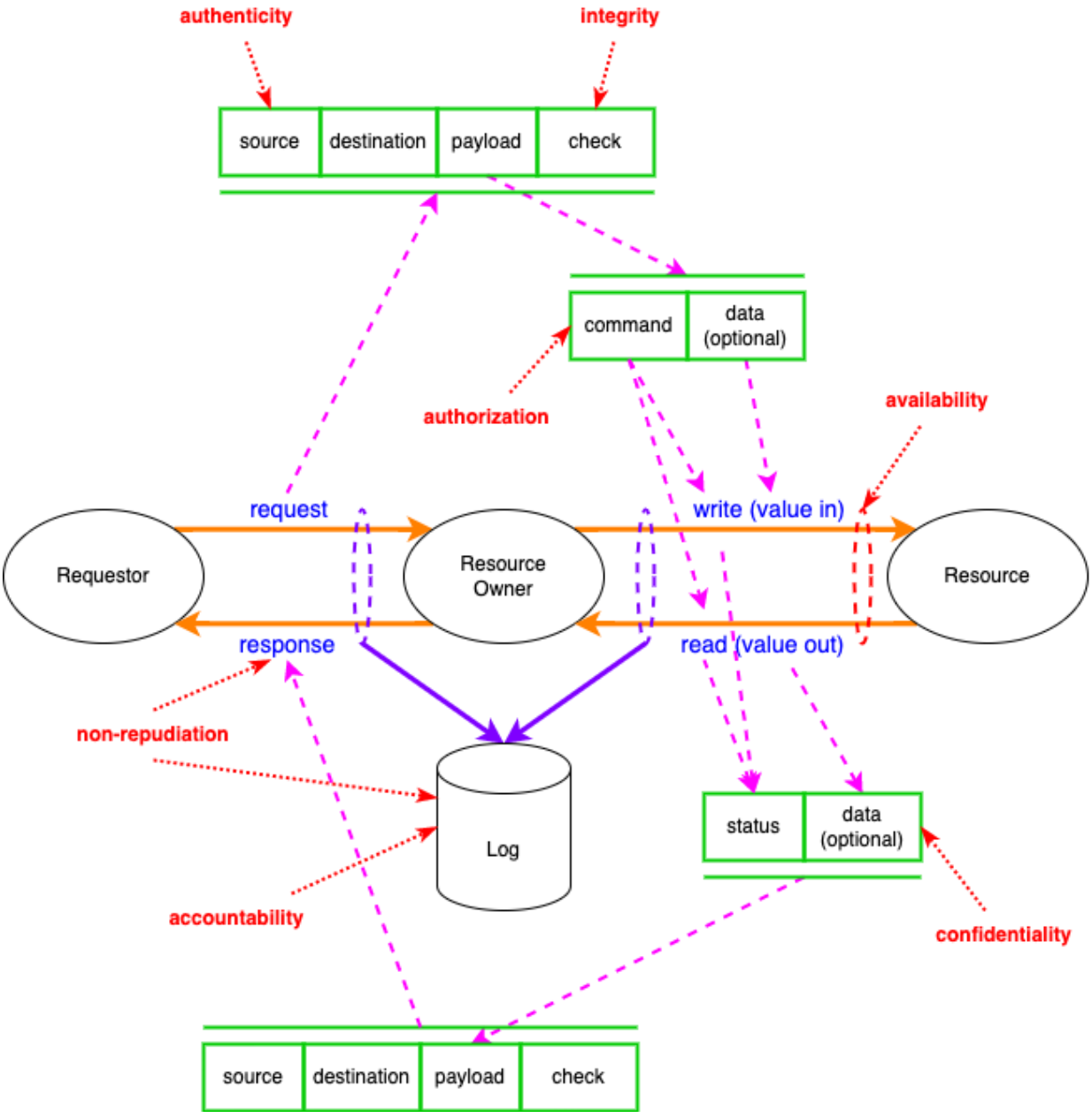
The **property** of a security requirement specifies the area within the security sphere. The specific designations are arbitrary and may be modified in order to allow for greater specificity. A typical set of designations would be:

Property	Description
Confidentiality	disclosure of information generally
Integrity	data accuracy and completeness generally
Availability	on-demand access to resources generally
Non-repudiation	denial of action taken or failure to acknowledge request
Authenticity	entity identity
Accountability	system state history (for example: for audit)
Authorization	entity privilege

Note: These are taken from the “extended CIA” specified in ***Proposal for a Recommendation on Cyber Security*** (UN ECE TRANS WP29 GRVA 2019 02e) [1] section 5.3.

Extended CIA Working Model

It is helpful to understand how the extended CIA relates to the general operational model of the interactions between a requestor, the resource owner, and the resource of interest. The following diagram illustrates these interactions and shows points of application for the extended CIA:



Note: This model is intended to be representational and not literal.

Note: Not all possible points of application of the extended CIA are shown. The **Understanding the Extended CIA Working Model** [3] elaboration document covers this material in greater detail.

Asset (What)

The **asset** of a security requirement specifies the resource under consideration.

Asset	State	Description
Executables	at rest	binary data which may be run on the system
Configuration Data		data used to establish the personality of the system
Databases		data in a structured format typically managed by specialized engines
Unstructured Data		data not handled as a database
Credentials		data specifically intended to establish and manage the identity of an entity
Logs		data used to record system events
PII	in motion	data containing Personally Identifiable Information
Packets		data in transit generally (for example: messages)
Memory	in use	data actively available within executing systems

State

The **state** of the data impacts the type of attacks which need to be considered and the mitigation used. The states are:

State	Description
at rest	data is in a stored form
in motion	data is in transit (this can be as raw information or encapsulated)
in use	data is being manipulated in the processor

Note: Assets, as described here, are types of things which are both distinct and whose securing requires specific consideration. Different assets require different mitigation strategies.

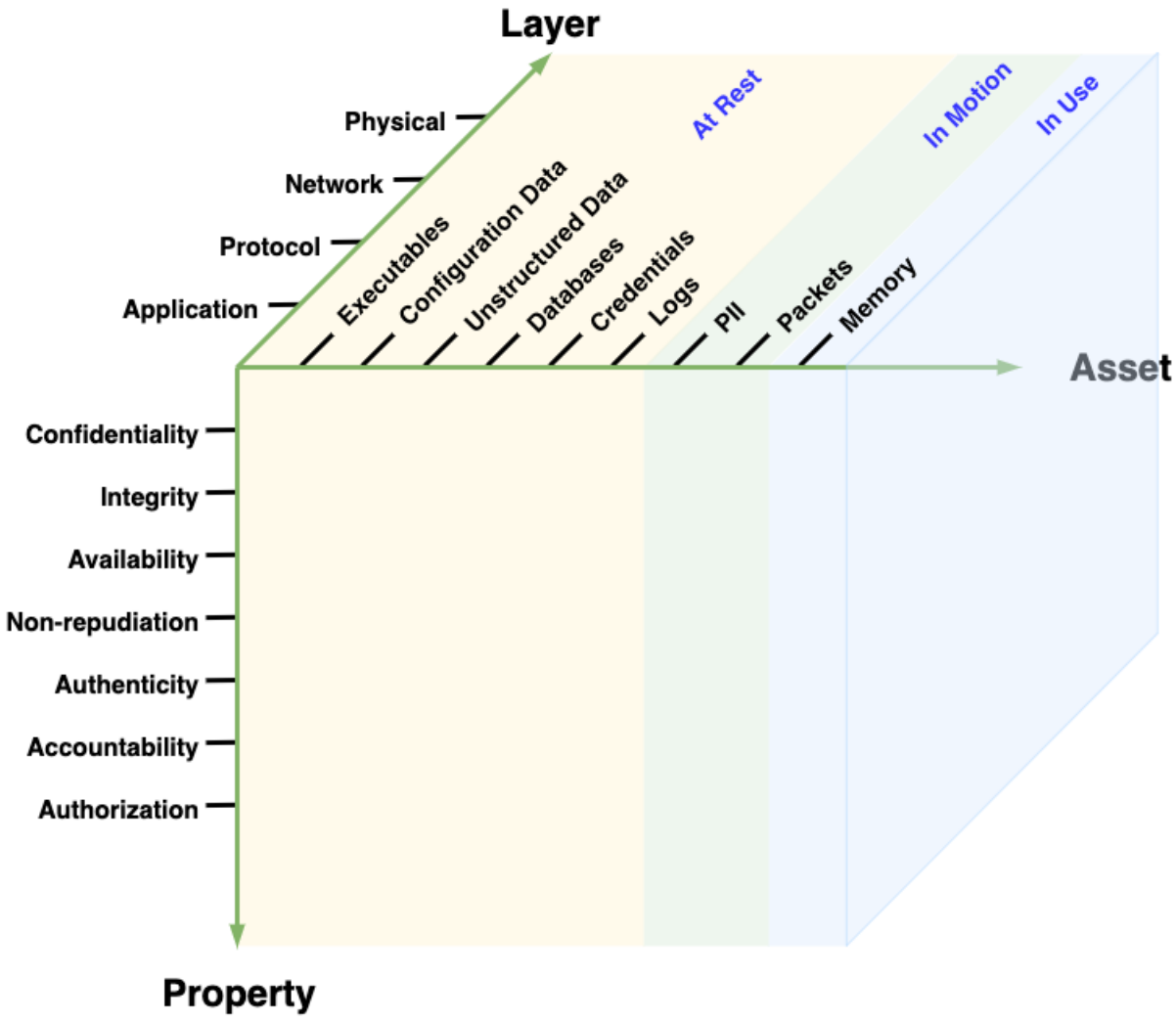
Layer (Where)

The **layer** of a security requirement roughly specifies a location within a TCP/IP abstraction. It is critical to consider the **layer** within security requirements as mitigations may be necessary at multiple layers of the abstraction. [An example of this is a denial of service attack. This type of attack is possible at any and/or all layers and is addressed differently in each.]

Layer	Description
Physical	hardware interface
Network	system-mediated transport
Protocol	custom data transport
Application	data handling within executables

Visualization

The taxonomy space can be visualized as follows:



Canonical Representation

A requirement's location within the taxonomy's space can be described with a 3-tuple textual representation. This allows for later processing. The textual 3-tuple for of a security requirement under this taxonomy would be:

[*property*.*asset*.*layer*]

This uniquely positions the requirement within the taxonomy's space. This representation could be expanded to position the requirement within the larger context of a product. One *possible* realization of this is:

[*product*.*release*.*component*.*property*.*asset*.*layer*.*unique_ID*]

Regardless of any other identification system in use for requirements, risk analysis, defect reports, etc., a requirement canonical form could be used to provide end-to-end traceability.

Consumption

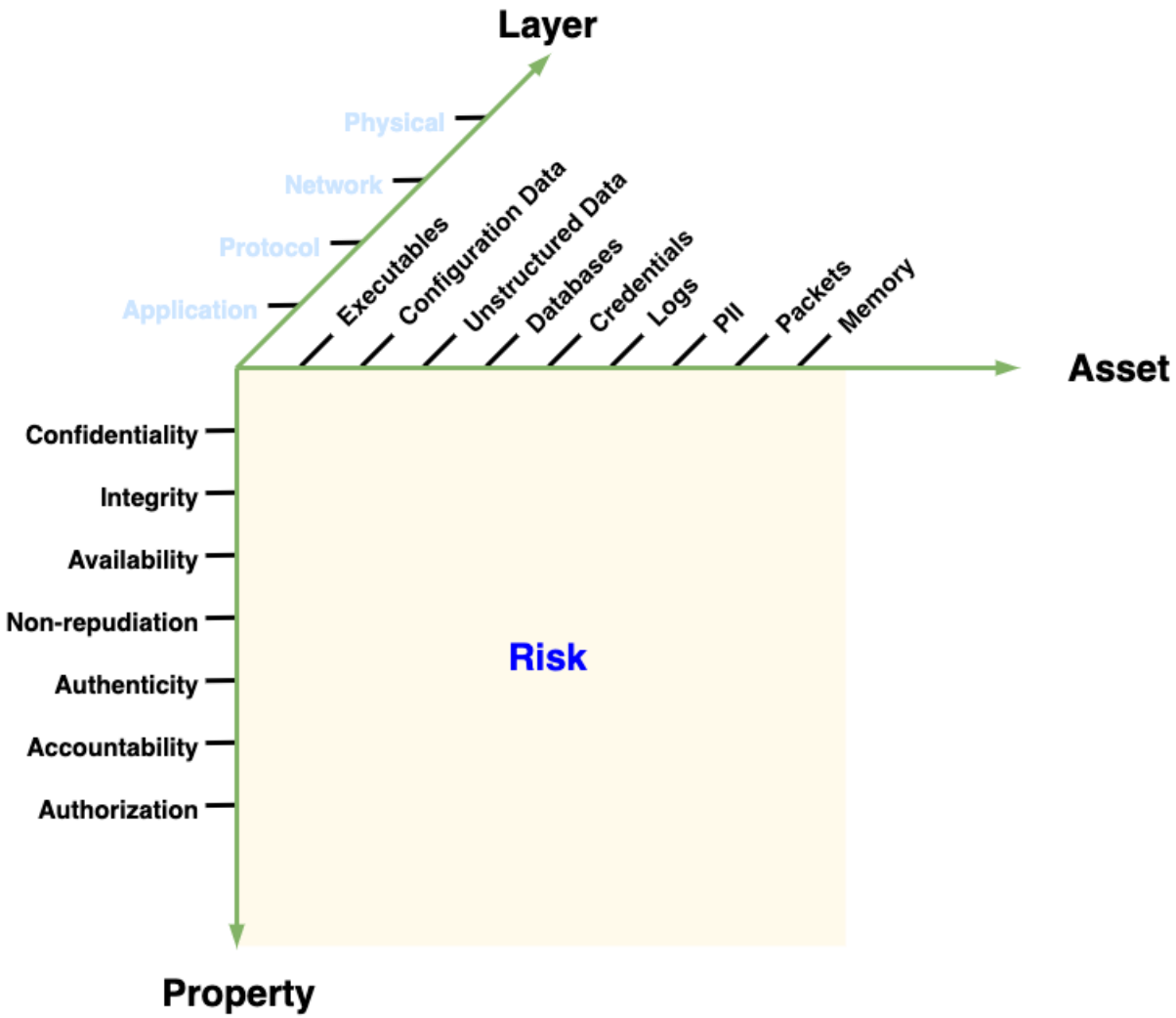
An advantage of this taxonomy is that it allows us to more accurately map areas of concern to different consumers within an organization. Specifically, it was designed to consider development, verification and risk. Each of these groups views the requirements differently. By deconstructing requirements in such a way as to allow for multiple views, it is possible to have a single set of requirements which can be queried based on the consumer. This allows a closer mapping to the unique language each consumer uses when considering requirements.

Consumer Views

When presented visually, the taxonomy can be considered as three views into the data, each with its own dominant pair of dimensions. The following are three suggested views into the data and how they are formed.

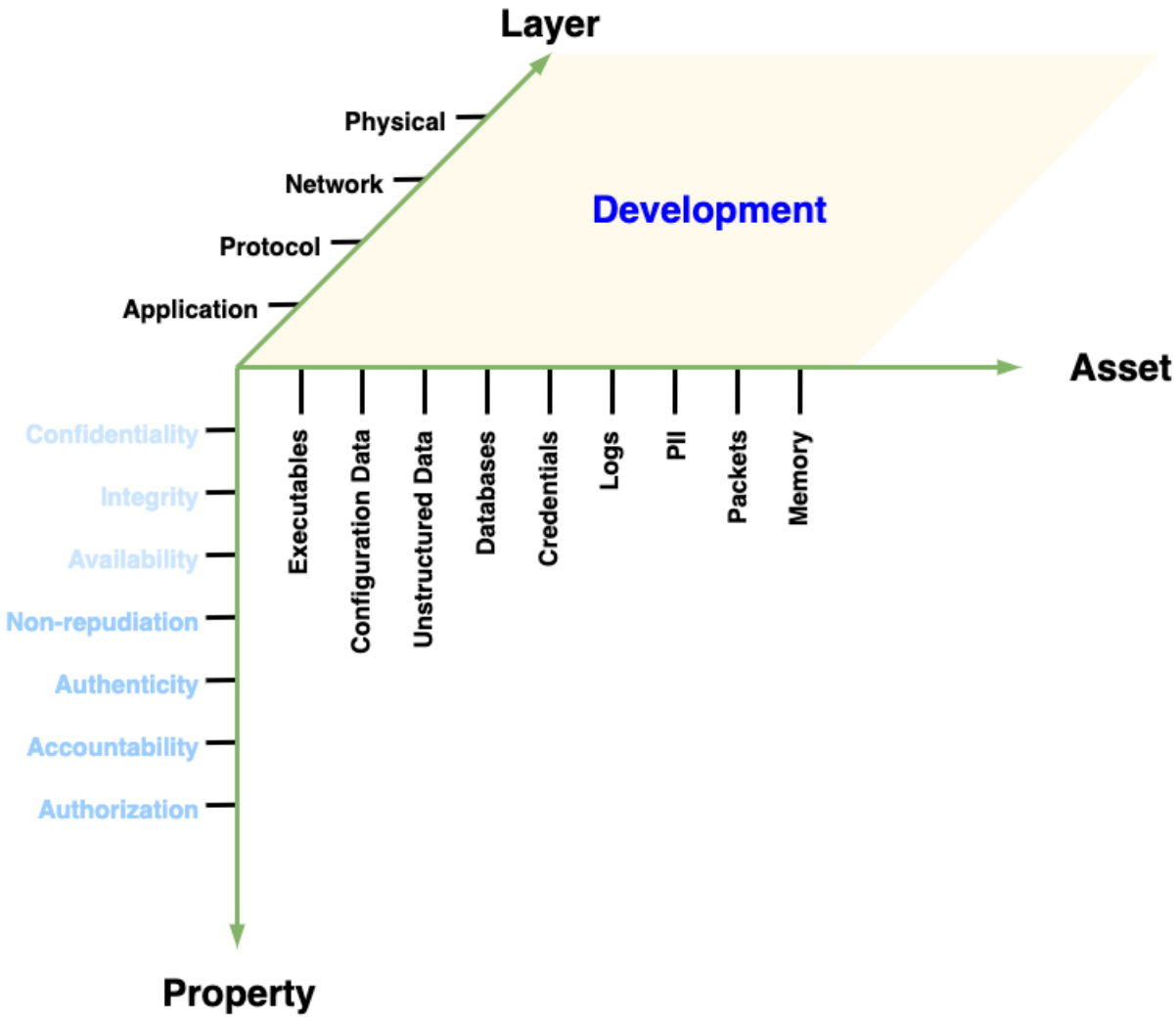
Consumer: Risk (What-Why)

The risk group has a focus on the Property-Asset view of the space. The Layer dimension is an implementation detail. The focus is on how the various properties are applied to assets. The layer is a secondary consideration. The same type of risk may occur for a given asset-property pair on various layers.



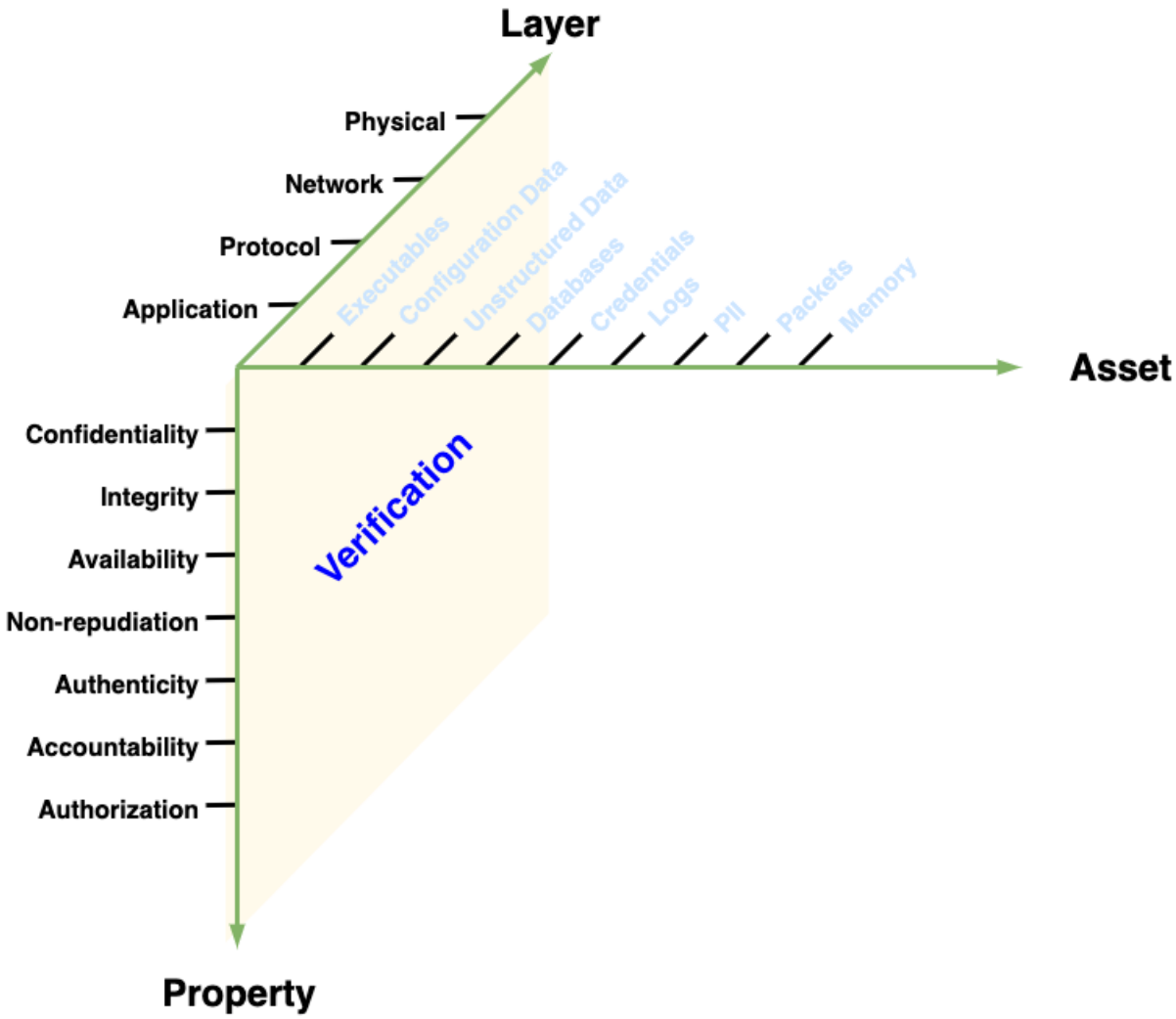
Consumer: Development (What-Where)

The development group's focus is on the Asset-Layer dimensions. Here the requirements are considered in the context of the asset and its layer. The property is secondary. Multiple properties may be applied to each asset-layer pair.



Consumer: Verification (Why-Where)

The verification group's view is the space along the Layer-Property axis pair. When creating tests, it is useful to think about the property and layer in which it is applied. Secondarily, the asset to which it will be applied. The same (kind of) test may be used on multiple assets.



Automation

By applying this taxonomy during the requirements phase, you eliminate the need to translate them later for consumption by the development and verification groups. From the risk-oriented version (layer spreadsheets), you should be able to use automation to re-slice along the other dimensions to generate development- and verification-oriented views and auto-populate the corresponding work items.

Example Usage: Requirement Gap Analysis (Consumer: Risk)

When considering a risk group-based view of the space we can represent requirements in a set of spreadsheets (one per layer) used to identify those areas needing requirements. This is a direct mapping from our above visualization for the risk group where we orient ourselves on the Property-Asset dimensions and push through the Layers dimension.

Below is an example of a set of established security requirements represented into each of the four layers. We can readily see where gaps in requirements exist.

Note: The following diagrams are based on the AVCDL **cybersecurity requirements per taxonomy** spreadsheet [2].

	Executables	Configuration Data	Unstructured Data	Databases	Credentials	Logs	PII	Packets	Memory
Confidentiality	■	■			■	■	■		■
Integrity	■	■	■	■	■	■			■
Availability	■	■	■	■	■	■			
Non-repudiation	■								
Authenticity	■	■		■	■	■			
Accountability	■	■		■	■	■	■		
Authorization	■	■	■	■	■				

Table 1: Application Layer View

	Executables	Configuration Data	Unstructured Data	Databases	Credentials	Logs	PII	Packets	Memory
Confidentiality					■		■	■	
Integrity								■	
Availability									
Non-repudiation								■	
Authenticity					■			■	
Accountability									
Authorization									

Table 2: Protocol Layer View

	Executables	Configuration Data	Unstructured Data	Databases	Credentials	Logs	PII	Packets	Memory
Confidentiality									
Integrity									
Availability									
Non-repudiation									
Authenticity									
Accountability									
Authorization									

Table 3: Network Layer View

	Executables	Configuration Data	Unstructured Data	Databases	Credentials	Logs	PII	Packets	Memory	Hardware
Confidentiality										
Integrity										
Availability										
Non-repudiation										
Authenticity										
Accountability										
Authorization										

Table 4: Physical Layer View

Example Usage: Requirement Specificity (Consumer: Risk)

You can use the taxonomy to perform an inverse gap analysis. In particular, to determine whether or not a requirement is sufficiently specific. If you attempt to map a requirement into the taxonomy space and the mapping isn't planar-contiguous, then the requirement is poorly formed. A requirement should be located in:

- a single spot in 3-space (three dimensions specified)
- a single line in 2-space (two dimensions specified)
- a plane (one dimension specified)

Note: The last case is arguably underspecified.

Note: Any requirement that does not satisfy one of these three is not sufficiently specific and needs to split into multiple requirements.

Note: Any requirement that specifies both a general as well as a more specific property (for example: integrity and tampering) is improperly scoped.

References

1. **Proposal for a Recommendation on Cyber Security** https://www.unece.org/fileadmin/DAM/trans/doc/2019/wp29grva/ECE-TRANS-WP29-GRVA-2019-02e.pdf
2. **cybersecurity requirements per taxonomy** (AVCDL working material spreadsheet)
3. **Understanding the Extended CIA Working Model** (AVCDL elaboration document)