# Title

# I'm Charles Wilson, Cybersecurity Engineering Technical Fellow at Motional, responsible for cybersecurity development lifecycle practice.

# This presentation will cover the AVCDL’s cybersecurity requirements taxonomy.

# Training Path

# This diagram shows the overall AVCDL training path.

# If you’re taking this training, it’s assumed that you’ve already completed the AVCDL overview training.

# This training covers the taxonomy of cybersecurity requirements.

# Introduction

The state of cyber security requirements today is pretty hit or miss.

Sometimes it hits the target, but usually we’re hard pressed to get close to the mark.

It's really inconsistent.

<pause>

What we've ended up with is just a huge pile of very specific requirements that are not attached to the functional requirements, but rather simply are added to the list of functional requirements.

Keep in mind that cybersecurity requirements are nonfunctional.

They're meant to constrain the functional requirements.

Because of that, they're not going to be constructed in the same way.

<pause>

What we'd like to have is a catalog, a library if you will of cybersecurity requirements

that we can pick and choose and attached to these functional requirements.

<pause>

We also need to have them organized in a far more systematic fashion than we've had in the past.

In order to achieve that goal, let’s start by reviewing the fundamentals of what we're asking of the requirements and where they need to be applied.

# Cybersecurity Control Points

Let's start by talking about cybersecurity control points.

Here we have two entities. A and B.

They represent arbitrary entities which communicate, store and process cybersecurity-relevant data.

Let’s consider where we would apply cybersecurity controls.

<pause>

The first place is in the data channel, where we apply controls to address data in motion.

We think about these things all the time when we talk about TLS or FTPS or other security-oriented protocols.

<pause>

The second place of control that we have is for data storage. This is data at rest.

Here we're thinking about controls applied either through an access control list type system or using mechanisms such as cryptography.

<pause>

Finally, we're going to address about data in use.

That’s when you have sensitive information that is being processed.

How do you address that?

<pause>

When we bring all these three together, we can see that we are covering the lifetime of the data.

Now let's look in more detail at the assets classes that fall into these three categories.

# Asset Classes

Let’s enumerate the asset classes for the different states that we have for data at rest, data in motion, and data in use.

<pause>

Our first class within data at rest is executables.

That is, any binary data which may run on the system.

This includes software and firmware.

Basically, anything which has a binary representation which can be interpreted as a set of instructions for the system.

<pause>

Next is configuration data.

This is the data used to establish the personality of the system.

One could consider this the metadata that gives personality to the executables.

<pause>

The next is structured data or databases.

This is data in a structured format and it's typically managed by specialized database engines.

<pause>

Next, we have unstructured data.

This is any data store not handled as a database, so binary data files fall into this class, as do plain text files.

<pause>

System credentials are something which we consider specifically because of their cybersecurity-relevance and sensitivity.

This is data intended to be used to establish and maintain the identity of an entity.

<pause>

Finally, we have log files.

This data is used to record system events specifically here, we're referring to either a data log or an audit log.

We tend to care more about audit logs within the realm of cybersecurity because of the sensitivity of the data.

<pause>

For data in motion, we're going to think about either PII, that is personally identifiable information because of its intrinsic sensitivity

<pause>

and also, more generally packets themselves.

These are the data units being used to carry messages with data in transit.

<pause>

Finally, for in use data, we're going to consider computer memory.

Here we have data actively available within executing systems.

Cybersecurity Properties

Now, let's take a look at cybersecurity properties.

These are the characteristics of the assets that we wish to ensure are maintained.

# Cybersecurity Properties (CIA)

We can't talk about cybersecurity properties without bringing up the CIA.

This triad of properties has existed for a very, very long time.

It's comprised of the properties of confidentiality, integrity and availability.

Now, although the CIA tends to held up as the gold standard, it's not and has never been universally accepted as the only basis for cybersecurity properties.

Let's take a quick look at a timeline that shows this.

# Cybersecurity Properties Timeline

When people talk about the CIA and want to mention where it comes from,

they point to a document that was written by Saltzer and Schroeder in 1975 entitled, The Protection of Information in Computer Systems.

Now, this is by no means the first time that anyone had spoken about the properties of confidentiality, integrity and availability.

We can go all the way back through documents that you see here by Conway Maxwell and Morgan in 1971 and back further to documents by Ware in 1967.

And then finally to a document titled On Distributed Communication: Security, Secrecy, and Tamper-free considerations by Baran in 1964, a full 11 years before Saltzer and Schroeder paper.

Throughout this period, there's a conversation going on as to what these things mean.

And by the time we get to 1975 we more or less settle on those three.

In 1983, when the Department of Defense released the Trusted Computer Systems Evaluation Criteria known as The Orange Book,

they didn't think that it was significant to put in a reference to the CIA model which makes one wonder.

And it’s not until 2002 that definitions for CIA show up in the United States Federal Register.

And from Saltzman and Schroeder until we get to the seminal Microsoft Security Development Lifecycle book in 2006,

a number of sources, either Parker or Firesmith or Jurjens or Mead and Hough

bring up that the CIA model itself is not sufficient to represent all the cybersecurity properties we need to consider.

And even the Microsoft SDL introduces STRIDE as an alternative to the use of CIA as the property set.

I'm not going to cover STRIDE in this discussion or why I don't think that it’s a good replacement for the CIA. That discussion will be covered in the threat modeling training material.

It’s important to appreciate that even after Microsoft’s SDL and the introduction of STRIDE, the conversations on replacing the CIA or extending the CIA continued.

And in 2007 with Calderon and Marta discussed a possible taxonomy of security requirements.

And as late as 2019 with Yu’s paper, Distributed Immutable Ephemeral - New Paradigms for the Next Era of Security have carried on the discussion.

<pause>

The two documents that I think are worth looking at in the context of extending the CIA are going to be

the 1984 ISO/OSI document also called X.800

And 2001 NIST SP 800-33.

<pause>

SP 800-33, entitled Underlying Technical Models for Information Technology Security gives us probably our best insight into what an extended set of properties might be.

So, let's look at that now.

# Security Services Model

So, here's the model for security services that NIST developed in SP 800-33.

We have the user on the left. The user could be a process or a person.

On the right, we have the resource of interest.

Between the two are a bunch of activities that mediate the interaction between the user and the resource.

If we think about this in the context of the CIA,

for confidentiality, we have the activity at the top labeled transaction privacy,

for integrity, we have proof of wholeness

and finally, for availability, we could map that to the access control enforcement to make sure that you can actually get at the thing you want to get at.

<pause>

But even after we do that, we have activities not covered by the CIA, but considered necessary.

<pause>

We've got non-repudiation.

So that actions that happen are attested to.

We have authentication

we have authorization,

And we have audit so that you can go back and see what actually happened.

<pause>

There are two additional activities which are sort of supervisory.

One is to restore the secure state and the other is intrusion detection.

Those are less property and more control aspects of the system.

So, let's take this and consider how this can inform a larger set of cybersecurity properties.

# Nested CIA

Returning again to our CIA triad,

the question becomes, is this the best way to think about these three cybersecurity properties and how they relate to each other.

<pause>

Although it is true that you could argue that they're orthogonal to each other, we can create a different representation for them.

<pause>

In this representation,

we still have data at the center of everything, but they're now nested.

Fundamentally, if you can't get at the data, nothing else matters beyond that.

If the data doesn't have integrity, even if we can access it, it’s useless.

And finally, we have confidentiality. If we have known good data, but can’t keep others from seeing it, we have a problem.

So, let's assume that we can use this as a workable model.

Let's take those properties that were talked about in NIST SP 800-33 which are referred to as the extended CIA in the UN R155 document and see how they can extend this model.

# Extended CIA

So, here's the extended CIA using the terminology from UN R155.

They form a set of concentric rings and they constrain the CIA, with each new layer being dependent upon the previous.

Now, let's look at each of the properties and give them more formal definitions.

# Cybersecurity Properties

So here are the cybersecurity properties.

<pause>

Note that these are listed in the order provided by UN R155 and not the order shown in the model.

<pause>

When we pull them all together, there are seven of them.

Confidentiality disclosure of information

Integrity data accuracy and completeness

Availability on-demand access to a resource

Non-repudiation denial of action taken or failure to acknowledge request

Basically, you acknowledge that you did what you did

Authenticity identity of the entity

accountability or audit system state history

and finally

Authorization entity's intrinsic privilege state.

# Resource Access Working Model

Script:

All of this theory is wonderful, but let's bring it down to something more concrete.

Let's develop a resource access model that we’ll be using.

<pause>

Here we have the base of our model.

We have a requester,

a resource owner.

And a resource.

<pause>

A request comes in,

the resource owner manages it,

accesses the resource,

and returns information to the requester.

<pause>

In a more sophisticated system, we log what's happening.

Since we are dealing with activities which are cybersecurity-relevant, we want to be able to log events for audit purposes.

<pause>

When we overlay this with a rough model of communication,

what we see is that we have a request that comes in.

This request decomposes as a source, a destination, a payload and some kind of integrity check.

The payload represents a command with optional data.

And that command may represent either a read, accessing the data or a write setting it.

<pause>

The data coming back from that activity is a status and optionally data representing either the value read or confirmation that the activity has taken place.

That data represents a payload in a response which, like the request, contains a source and destination and an integrity check.

<pause>

If we apply a CIA cybersecurity property set to this model,

we see that we’re only be considering the integrity check in terms of integrity,

the data, in terms of confidentiality

and the channel in terms of availability.

<pause>

However, if we include the additional properties from the extended CIA property set,

we see that the additional touch points into the data,

Combined with the controls that we can discretely apply,

Becomes far more interesting and significant.

# Combining Property and Asset

When we combine the concepts of cybersecurity property and asset,

what we we end up with a much more sophisticated basis to use when creating cybersecurity requirements.

Now, there have been systems proposed which do just this,

defining requirements in terms of properties and assets.

Unfortunately, this leaves a gap.

So, let's look at an example of that gap and then discuss how to address it.

# Same Problem – Different Floor

At first blush, the combination of cybersecurity property and asset class seems to be sufficient to be able to dispense requirements.

However, there’s a problem I’ve seen manifest while dealing with real world scenarios tracking down issues in a fairly sophisticated system which used a combination of over-the-air communications, internal ethernet communications, and multiple processes.

Rather than go into that fairly complex situation,

let's talk about a simple one which illustrates the same problem.

First, we'll start with the base communication case where we just want to send a message from a source to a destination.

<pause>

We can see that here.

We have a message and we're going to call the contents of the message a command.

It's going to go from the source to the destination.

Now, we would think that all right, regardless of, of what the source is, what the destination is, what the channel is that we could just look and say here's what the asset that we're dealing with is here's what the cybersecurity properties are, but let's expand it some more.

<pause>

That message may have data associated with it.

<pause>

We might have more sophisticated communication, where the message has an integrity check for the data.

This is a good thing.

Let's say that the source and destination aren't, as would be easiest to talk about here, just going from one process to another.

<pause>

What if we have to include a communications channel and we're going to use a protocol over that channel?

Now, we have to talk about the fact that in addition to our original source and our destination, we have the source and the destination of the communications channel.

<pause>

And in all likelihood, it's going to have integrity data.

<pause>

It's also not unlikely that our message, wrapped in protocol A is going to be contained in and sent via another channel and it's going to have a wrapper which is protocol B

<pause>

and that it may have an integrity check on it.

<pause>

Each time we add intermediation between the original source and destination, we create the opportunity for multiple problems.

Were we to only consider the end-to-end aspect, we might have a cybersecurity requirement regarding integrity, giving us a false sense of security.

In reality, each of the source and destination pairs can and should be considered separately.

<pause>

And if we were to threat model this, we could say yes, between the final source and the final destination we have an integrity check, but we also need one at the intermediary layers.

So, there's this intrinsic concept of layering going on here where we might want to apply different types of security to each layer.

And if we don't recognize that upfront, we're going to miss the opportunity to be specific about where we apply our controls.

Additionally, when we go to use threat modeling, in order to assess the design to make sure that the controls are in place, the threat modeling tool will miss these things because they will be masked by a lack of layering.

# Layers

With this example as a backdrop, let’s examine the layers we’ll need.

<pause>

In an effort to make things as simple as possible but no simpler,

we're going to only have four layers that we consider within our model for the taxonomy.

The first one, the physical layer represents the hardware.

The next layer is the network layer.

It consists of all the system mediated transports, things like HTTP, FTP and PTP.

Basically, these are protocols which have predefined interfaces which can't be changed by developers.

Next, we have the protocol layer.

The protocol layer represents all the custom data transports.

That is, any transport protocol which is created and controlled by the organization and allows them to manipulate such things as whether or not there's data integrity, whether or not there are sequence numbers, whether or not parts of the data are encrypted things which are under the control of the organization.

Finally, we have the application layer.

Here we're dealing with processes within executables which are manipulating the data.

These are our end points, if you will, as discussed in the previous example.

Taxonomy Space

So, here's what the complete taxonomy space looks like.

We've got the cybersecurity property on one axis, the asset on another axis, and the layer on the third.

It’s all well and good that we have a taxonomy space.

What do we do with it?

How do we populate it?

How do we use it?

Let’s address these questions.

Identifying Requirement Needs

With the taxonomy space established, let's look at how we use it to identify requirement needs.

<pause>

We're going to look layer by layer and consider for each combination of cyber security property and asset whether we want to make assertions about that particular combination.

The advantage that we have in this particular decomposition,

having the extended CIA as the properties on the left-hand side and the assets across the top, and going layer by layer,

is that we can both fully consider each asset class in the context of cybersecurity properties and how they are applied.

That is, each of the four layers is typically worked on by a different specialized developer applying distinct controls.

<pause>

Let's start with the application layer.

For executables, the case can be made for requirements across all seven cybersecurity properties.

Similarly, for configuration data, we can cover six of them,

and so on, until we get to memory where we've only been able to establish that there are requirements that we can assert for confidentiality and integrity.

<pause>

It's important to remember that when creating requirements

that they be represented in a way that ensures each is unique, simple, testable, and unambiguous.

In doing so, you'll be able to create a mapping between individual requirements and tests to verify their implementation.

We'll get to more on that in the requirements training.

<pause>

The next layer in the sequence is protocols,

but we found after completing the requirements identification exercise

was that the protocol and network assertion locations were the same.

It’s important to note that they were not identical in terms of the actual number and specific requirements that we have for each of these combinations,

Merely that the property-asset mapping was the same at the macro level.

<pause>

Finally, there's the physical layer.

Here we added a hardware column.

This is because hardware has nothing to do with data at rest, in motion or in use per se.

But it is a placeholder allowing us to assert specific requirements dealing with hardware itself.

<pause>

All of the requirements that established using this taxonomy have been provided in a spreadsheet in the AVCDL repo.

In that document, you can see how the decomposition of the requirements has been done and what the specific requirements are.

Use Cases

An interesting thing about the taxonomy is the way that it reflects into various use cases.

<pause>

For instance, if we look at how risk views the space, it tends to look at it from the classic property asset combination;

because where in the layers the issue occurs is not really relevant.

You tend to think in combination of for instance, there is an availability issue with the credentials.

That doesn't require us to think about the layer at all in order to make a judgment.

<pause>

Similarly, when you think about the development use case,

developers are concerned with a combination of the asset and the layer because what they're doing is they're applying a requirement.

They need to know where within the layered construct of the system the cybersecurity control needs to be applied

and what asset it needs to be applied to.

The fact that it entails a cybersecurity property isn't really relevant because the requirement tells them what it is that they are constraining.

<pause>

Finally, let’s consider verification.

Verification, thinks about attempting to either confirm or subvert a cybersecurity property.

They're going to do so at a particular layer within the structure.

What they manipulate is not really specific to or contingent upon the asset per se.

It may involve it, but it's not as important as say, looking at all of the authentication issues that occur at the application level.

Summary

Hopefully, this gives a good overview of how the AVCDL's cybersecurity requirement taxonomy came about and what its structure is.

Some takeaways are

<pause>

that the decomposition really matters.

It's the combination of asset, property, and layer and the specific subdivisions existing within those that really make the taxonomy work.

<pause>

Next is that the decomposition maps to reality.

The various proposals that have existed over time were all well and fine,

but what really matters is whether or not the application of requirements,

in the context of the taxonomy, actually does map to reality without gaps.

<pause>

Finally, it's important that whether or not you're using the requirement set provided in the AVCDL repo,

that the requirements that you're using be chosen with intent

and not simply built up from someone else's list of things

or are treated as functional requirements instead of the nonfunctional requirements that they actually are.

AVCDL on GitHub

All AVCDL materials, both in source and distribution forms, are available on our GitHub site, as shown here.

Because of the size of the repository, it's recommended that you either clone the repository or download a ZIP archive of it, if you're not familiar with using git.

Instructions for downloading a ZIP archive are linked to on the repository’s front page.

Next Steps

With this training complete, you can proceed the Security Requirements training.

References

Here are references to the source material used in the creation of this presentation.

They'll also be included in the video description.

Additionally, this presentation’s source material will be provided on the AVCDL GitHub repository.