# Title

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

# This presentation will cover cybersecurity requirements.

# 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** and **requirements taxonomy** trainings.

# This training covers **cybersecurity requirements**.

# Introduction

Even in today's world of modern engineering, cybersecurity remains a mystery to many development teams, and about the only thing they like less than having to add cybersecurity controls is to deal with cybersecurity requirements.

The development view of cybersecurity requirements is that there exists a magical cybersecurity checklist and that cybersecurity is able to provide development with this checklist specific to the element they are working on at any time.

<pause>

The problem is that it's impossible to go from the total number of cybersecurity options and apply them generically to any given product, subsystem within that product, or element within that subsystem.

And so, this training is going to cover and hopefully illuminate what the relationship between the system requirements and cybersecurity requirements are, how the cybersecurity requirements get taken apart and put together, and how we apply them in a systematic fashion.

Let's start by talking about the functional requirements of the system.

# Functional Requirements to Development Tasks

If we take cybersecurity out of the picture for a moment, we can consider that a system is comprised of one or more functional requirements

<pause>

and that each of these functional requirements can be decomposed into one or more development stories.

<pause>

Further that each of these development stories can be broken down into one or more development tasks.

Now, we're talking about cybersecurity.

So, let's think for a moment how we're going to interact with this functional system.

# Functional and Non-functional Requirements

Generally speaking, there are two big categories of requirements.

Those are functional requirements and non-functional requirements.

Functional requirements are all about what the thing is supposed to do and non-functional requirements are constraints on the functional requirements.

<pause>

Now, in order to make this distinction clear, let's cover a, a real-world example.

<pause>

Let's say that we have two processes.

These could be two computers, they could be two cloud instances, or whatever.

<pause>

For the sake of argument, we have two processes A and B.

<pause>

These processes may or may not be associated with an entity.

<pause>

We're going to send a message from A to B and so our functional requirement is that A sends me a message, that's it.

What the message contains, how large the message is, what the channel being used is.

We don't care.

The function requirement is we need to be able to send a message from A to B.

<pause>

Let's look at how we could constrain this within the context of cybersecurity.

<break>

From a cybersecurity standpoint, if the message doesn't cross a trust boundary,

<pause>

here represented by this red dash line, then typically we don't care because, fundamentally, if A and B trust each other, then there's no need for cybersecurity controls.

<pause>

For the sake of argument, we're going to say that this trust boundary exists.

<break>

Well, if we're going to do cybersecurity, this typically means that we're going to be using cryptography.

And as such, there are things that we can say about how we use cryptography, what best practices should be applied, things of that nature.

<break>

Now, let's turn to something more specific.

Say what we want to do is to assure that only A and B can read that message, and that we want to use the simplest possible method to achieve this.

<pause>

We're going to use symmetric cryptography,

<pause>

which means that A and B share this key that allows them to access the message.

<break>

We can get more sophisticated if we say that there's a tunnel that we want to secure and not worry about encrypting every single message that we put through that tunnel.

Now, when we do this, typically, we're going to use asymmetric cryptography,

<pause>

which means we're going to be using certificates.

<pause>

That means we need the ability to look up the certificate for the other party, unless we save that certificate locally.

<pause>

Now, local certificate storage doesn’t scale very well in embedded systems.

What we typically tend to do is go to the net and reference the public key.

<pause>

This has its own set of problems.

<break>

Now, let’s say we decide we want that little bit of extra security.

In addition to securing the channel, we’ll also secure the message, because there's information in that message that's so important that we’re not going to trust the channel security alone.

<pause>

So, now we have the message encrypted too.

<pause>

All of these requirements that we've applied are realistic for the different situations, but which ones are functional and which ones are non-functional?

Functional and Non-functional Requirements (Summary)

Except for the first case where we assert that we want to have a message that goes from A to B, all the other requirements that we stated are non-functional because the system fundamentally only does that one thing.

<pause>

It just sends a message from A to B.

Everything else that we do is constraint.

<pause>

This shows that you can't just say “here are all of the possibilities that you could have and you know, please apply just the right ones.”

<pause>

What’s needed is known as tailoring in order to apply the correct cybersecurity requirements and do so with a sufficient level of rigor.

# Cybersecurity Requirements to Development Tasks

Returning to our previous diagram where we showed how we get from functional requirements to development tasks,

<pause>

let's look at how we get from the cybersecurity requirements to those same development tasks.

<pause>

We established in the cybersecurity requirements taxonomy,

<pause>

a set of global cybersecurity requirements based on a taxonomy where we considered the cybersecurity property, the asset, and the layer in which each requirement exists.

Now let’s take these requirements and determine how we're going to apply them to a given functional requirement.

<pause>

Let's start with a functional requirement

<pause>

and use it as a filter to determine which requirements are necessary.

<pause>

We're going to call these requirements, the tailored cybersecurity requirements because they are tailored to the functional requirement that we're dealing with.

<pause>

And we're going to call the combination of that function requirement and the applicable cybersecurity requirements that have been tailored

<pause>

an augmented functional requirement which now stands in place of the functional requirement.

<pause>

Now, in implementation, what we're actually doing is we're taking the functional requirement and we're creating dependencies on the tailored cybersecurity requirements, but this visualization gives us a better sense of, what the inputs are, and how we’re progressing.

# Overall Dependencies (1)

At a higher level, we can think of the data stores holding all of these different bits and pieces.

<pause>

We have the function requirements.

<pause>

We have the cybersecurity requirements global catalog.

<pause>

These feed into

<pause>

the tailored cybersecurity requirements catalog.

<pause>

We then use the functional requirements and the tailored cybersecurity requirements to create

<pause>

an augmented functional requirement set.

<pause>

Finally, from those and we create stories and tasks which live in the task tracking system.

# Cybersecurity Requirements-related Guidance Documents (1)

At this point, you may be thinking that there has got to be more than just these few bits of very high-level information that goes into dealing with cybersecurity requirements and applying them to real systems and situations, and you'd be absolutely correct.

<pause>

This training video is not intended to replace the underlying materials that exist, but to act as a jumping off point for working with the related materials.

<pause>

Now, the diagram that I about the detail covers the relationship between the cybersecurity requirements-related guidance documents.

<pause>

Off to the right is a legend showing the document types,

whether they're blog posts that have been created,

whether they are templates that exist to be able to apply the information within the AVCDL,

whether they're strictly reference documents,

or whether they're documents which are secondary to the AVCDL primary document.

<pause>

There are other indicators that are shown as to whether documents that are created in the process of applying the AVCDL are done by the supplier, the customer, or a combination of the two.

<pause>

And we have flows that show whether these are supplier documents, guidance documents or blog posts.

<pause>

Now let’s talk about what we’ve got.

<pause>

There are two blog posts which provide a good introduction to the space in terms of why we end up pulling a supplier in and how they relate to requirements and also why requirements are an important aspect of being able to build a case for cybersecurity.

<pause>

One is

<pause>

“Turtles all the Way Down, Security at Every Level” and

<pause>

“Traceability, Making the Case for Certification.”

<pause>

Now, there is the secondary document,

<pause>

**Security Requirements Taxonomy**, which is the subject of the prerequisite training. Derived from that is the reference document

<pause>

**Cybersecurity Requirements per Taxonomy**, which is a set of roughly 70 requirements that are in the global catalog that satisfy our observations for what will fulfill the needs within the context of the cybersecurity requirements taxonomy.

Additionally, we have the

<pause>

**Global Security Requirements** secondary document.

And when we apply both of those, we end up with this

**Global Cybersecurity Catalog**.

<pause>

At our next level, we have

<pause>

**Product Cybersecurity Requirements**, and

<pause>

**Element Cybersecurity Relevancy**.

<pause>

Finally, all of these feed into the

<pause>

**Tailored Cybersecurity Requirements**.

<pause>

Now, that's if we just are doing all of the work ourselves. As soon as we involve a supply chain,

<pause>

an entire additional set of information is needed and the AVCDL provides documentation and references and templates for that.

We have the

<pause>

“Where are You at, Level Setting Supplier Cybersecurity Maturity” blog post, which talks about the

<pause>

**Supplier Self-reported Cybersecurity Maturity**, and we provide

<pause>

a template for that. And a guidance document

<pause>

**The Supplier Self-Reported Cybersecurity Maturity Assessment**.

<pause>

We also have the

<pause>

“AVCMDS - The Autonomous Vehicle Cybersecurity Manufacturer Disclosure Statement” blog post and

<pause>

the corresponding document created by the supplier,

<pause>

the **Supplier Cybersecurity Disclosure Statement**, and

<pause>

the template that they fill out. And the guidance document

<pause>

**Autonomous Vehicle Cybersecurity Manufacturer Disclosure Statement**.

<pause>

Finally, when we have processes that need to be mapped, we have a

<pause>

Vendor Process - AVCDL Product Mapping,

<pause>

a template and an

<pause>

**Understanding Supply Chain Process Mapping** elaboration document.

<pause>

So, it's not that we just have requirements and off you go.

<pause>

There is a lot of guidance and a lot of material available to assist in getting from a global set of requirements and a hypothetical system to actual tailored cybersecurity requirements.

# Element Cybersecurity Relevancy

This brings us to probably the most important aspect of cybersecurity and cybersecurity requirements and that's cybersecurity relevancy.

You see that the title is **Element Cybersecurity Relevancy**.

<pause>

The term element is specifically chosen here because it's a level of abstraction that isn't what could be termed the item, which is an important concept in safety.

<pause>

It's not a component which could be tied to a particular supplier.

<pause>

It's not a feature. It's not a subsystem.

<pause>

It’s a pure abstraction, which is any decompositional aspect of a system that allows you to be able to reason on cybersecurity in the context of that abstract subdivision with relationship to other subdivisions.

<pause>

Let's consider whether or not we need to do anything with regard to cybersecurity, because if you don't have to do cybersecurity, that's a plus, right?

Because there isn't going to be all of the work that goes along with doing cybersecurity.

<pause>

Let's go through a series of questions to see whether or not we need to apply cybersecurity controls to a given element of the system.

<pause>

This set of questions is derived from **ISO/SAE 21434, Annex D, Figure D.1**.

<pause>

Our first question, is the element an E/E element component or system?

<pause>

E/E in this context, is an electrical or electronic item.

The element could be hardware, software, or a combination of the two. It doesn't matter what the realization of the element is.

<pause>

If the element isn't electrical or electronic, then it's not cybersecurity relevant. Say you had a bolt. It’s not cybersecurity relevant. It may be relevant to other things, but it's not relevant to cybersecurity.

So, if it's not an E/E item,

<pause> then we can just put it aside and move on.

So, if you have a component, say a sensor and that sensor is purely mechanical, then it's not going to be something that we need to worry about.

Otherwise, we're going to ask our next question.

<pause>

Is the element ASIL rated?

<pause>

That is, is this an element which has safety implications in the context of automotive?

We would consider this to be a non-QM, ASIL-rated element.

<pause>

If it is ASIL rated, then it's cybersecurity relevant and we have to deal with it.

<pause>

If it's not ASIL-rated, then we ask the next question which is,

<pause>

Does the element have interfaces which are external to the system?

<pause>

External to the vehicle in the automotive case. This applies both to active and inactive or disabled interfaces. So, is there some kind of port that hangs off of this thing?

<pause>

This also includes any user accessible interfaces within the vehicle. So, if you have USB ports that are connected into the system and they are not strictly just power, or if you have a sensor whose interface is exposed outside the vehicle, then that counts.

And if that's the case again,

<pause>

we have cybersecurity relevancy and we must address that.

<pause>

Otherwise, we're going to go on to question four, which is

<pause>

Does the element handle data that crosses a trust boundary?

<pause>

Now we talked about trust boundaries earlier.

So, we're dealing with a disparity in levels of trust because if the two end points that a communication goes through trust each other, then there is no cybersecurity that we need to apply because since they're trusted, they share information freely.

<pause>

Trust boundaries include, but aren't limited to, physical, privilege, and network.

In the case of the third question, that external vehicle interface, that's a physical boundary.

<pause>

It’s so important that we bring it up before the trust boundary case because we want that to be always handled.

<pause>

With privilege, we're thinking about things something mediated by an operating environment. Things like user and kernel processes, or different users, or different groups, things that are managed typically by access control.

<pause>

Finally, we have network boundaries. Anytime you are going from your little corner of the world across a network, you have to consider the cybersecurity implications.

And again, if our answer to that one is yes.

<pause>

Off we go to cybersecurity relevancy.

<pause>

If not, we're going to move on to our fifth question, which is,

<pause>

Does the element collect or process personally identifiable information PII or any other regulated user data?

<pause>

This question is one that covers user data that's regulated by the EU under GDPR and other jurisdictions because there exist regulatory requirements where cybersecurity is the mechanism for control.

<pause>

If it is regulated, it's cybersecurity relevant.

Otherwise, we go to our final question, which is

<pause>

Does the element directly handle sensitive data?

<pause>

Does the element is managing data which has an impact on the operational safety of the system or related activities?

<pause>

This can include executables, configuration, data, databases, unstructured data, credentials, logs

<pause>

anything that would impact the proper operation of the system. All of these assets that I've mentioned come from the security requirements taxonomy.

<pause>

If that's true again, it's cybersecurity relevant.

<pause>

And if it's not, we can call it not cybersecurity relevant and we don't need to handle it.

So, these six questions allow us to determine for any element at any resolution within the system, whether cybersecurity controls can be applied.

<pause>

It’s worth noting that element cybersecurity relevancy should be determined iteratively. This is done in order to narrow the scope of where cybersecurity controls need to be applied.

<pause>

It may be the case that a component is deemed cybersecurity relevant, but that within that component only a single function is actually in need of cybersecurity controls.

# Product-level Requirements Workflow

Now that we've established what makes an element is cybersecurity relevant, let's look at the workflow that we go through in order to create that tailored product level cybersecurity requirements catalog.

# Requirements Applicability Analysis

Our first step is the requirements, applicability analysis,

<pause>

the security SME

<pause>

and the development SME sit down with

<pause>

the element’s functional requirements and

<pause>

the global cyber security requirements catalog and

<pause>

perform a requirements applicability analysis. That is, they're going to look and see for the system that we have and the functional requirements that it has what aspects of the global catalog apply and the ones that do apply end up in

<pause>

the list of applicable requirements.

Now, it's important to recognize that the global catalog is not a magic box, it may in fact have deficiencies.

<pause>

And if the development SME and the cybersecurity SME in their discussions of the function requirements recognize the fact that you know there's something missing because the requirements that have come out of the taxonomy don't cover a particular asset for a particular cybersecurity property at a particular layer,

<pause>

then that information needs to be fed back and the global security requirements catalog needs to be updated.

# Resource Access Working Model

One of the main tools that we're going to use in order to perform this analysis is

<pause>

our resource access working model.

The basis of this in the extended CIA was covered in depth in the prerequisite video on the AVCDL requirements taxonomy. So, I'm not going to go into that here.

<pause>

To review, we have our basic interaction between a requester

<pause>

and a resource

<pause>

as mediated by the resource owner

<pause>

with data and audit logging being done on the transactions that are occurring.

Additionally, we show

<pause>

how the request and response

<pause>

and the payloads for each are handled and also

<pause>

the points where the various cybersecurity properties are applied.

We can use this model to facilitate the discussion between the cybersecurity SME and the development SME when discussing the various requirements in the global catalog and how they might apply to the element under consideration.

# Taxonomy Space

As a reminder,

<pause>

when we talk about the cybersecurity taxonomy space, we're referring to the space shown here in this diagram previously called out in the AVCDL requirements taxonomy training.

<pause>

On one axis, we have the cybersecurity property of confidentiality, integrity availability, non-repudiation, authenticity, accountability, and authorization.

<pause>

On the second axis are assets: executables, configuration, data, unstructured data, databases, credentials, logs, PII, packets, and memory. These are group into data at rest, data in motion, data in use.

Lastly, on the third axis, the locality of data is shown: whether it's in the application, protocol, network, or physical layer of our representational space.

# Identifying Requirement Needs – Application Layer

Here’s another diagram from the requirements taxonomy training.

<pause>

We're showing the application layer and an identified set of requirements areas where we're asserting that yes, for a particular cybersecurity property and a particular asset, we want to make assertions about the cybersecurity.

<pause>

It's this combination of aides

<pause>

(resource access model, taxonomy space, and layer-based requirement needs visualization)

<pause>

that helps the cybersecurity SME inform and communicate with the development SME in order to establish a subset of the global catalog applicable to the element under consideration.

# Cybersecurity Global Requirements Catalog

Here's a portion of a flattened version of the AVCDL-supplied cybersecurity global requirements catalog based on the AVCDL requirements taxonomy.

<pause>

Every requirement has unique identifier, which can be used to have discussions both internal to the organization and externally, if you're sharing those requirements with other entities,

<pause>

the requirement itself,

<pause>

the cybersecurity property dimension,

<pause>

the asset dimension and finally,

<pause>

the layer dimension.

<pause>

This file is provided in the AVCDL repository.

<pause>

It's important to note that just because the AVCDL provides this catalog doesn't mean that it's necessarily the catalog that has to be used by every organization. It's simply that these are the ones that we've identified.

Tailored Catalog Creation

Once the cybersecurity and development SMEs have settled on

<pause>

a list of applicable requirements,

<pause>

the tailored catalog is then created

<pause>

and here it's described as the product-level security requirements catalog. Other places it's referred to as the tailored catalog. They're both the same.

<pause>

Additionally, we generate a requirements report.

<pause>

We do this in order to be able to review the decisions we’ve made in order to improve the process.

Tailored and Macro Cybersecurity Requirements

The thing that makes the cybersecurity requirements taxonomy so effective is that it doesn't deal with the specifics of individual protocols or cryptographic mechanisms.

<pause>

This allows us to have a very adaptable global catalog.

<pause>

At the same time, it means that we don't have the ability directly to take the shortcuts that we normally would by saying things like we want to use HTTPS or we want to use SecOC.

<pause>

To bridge that gap, let’s introduce the concept of macro cybersecurity requirements.

<pause>

These are tailored cybersecurity requirement clusters which allow us to have a shorthand to describe a particular set of cybersecurity requirements.

<pause>

This gives us the ability to talk about these well-known mechanisms that accommodate an entire set of cybersecurity requirements, while simultaneously giving us traceability down to the cybersecurity property level.

Instead of simply saying we're going to use protocol X and hoping everyone understands what that implies. We can know that protocol X points back to fundamental cybersecurity requirements.

<pause>

So now let's look at how that will work.

Tailored and Macro Requirement Relationships

Let's look at the relationship between these various requirements.

<pause>

First, let's talk about the functional requirements. Functional requirements will have underneath them

<pause>

technical requirements. And it can be said that functional requirements implement technical requirements.

<pause>

Similarly, we can say

<pause>

that the tailored cybersecurity requirements

<pause>

refine the cybersecurity requirements. They put more meat on them, if you will.

<pause>

The macro cybersecurity requirements can be said to implement the tailored cybersecurity requirements in the same way that the functional requirement implements the technical requirements as a set.

<pause>

Then what we can say is that

<pause>

the augmented functional requirement implements the function requirement and simultaneously is constrained by each of the macro cybersecurity requirements.

<pause>

Now it's possible that there is no tailoring at all that needs to be done and it's possible that there is no macro layer that needs to be done.

And so, one could say that the augmented functional requirement is constrained by the cybersecurity requirement directly.

But, in its most expressive form, this is the diagram that shows us what can be our most complete representation of the relationship between these various elements.

<pause>

Now, let's look at an example.

# Macro Cybersecurity Requirement Example (SecOC)

For a concrete example, we're going to look at the secure onboard communications protocol, also known as SecOC.

<pause>

Specifically, we're going to look at SecOC security profile one.

<pause>

As with all of these complicated bits and pieces, references are provided at the end of the training presentation.

<pause>

Here we see the portion of the global catalog covering the network and the protocol layers.

<pause>

There are four requirements that SecOC covers.

<pause>

Those are that

<pause>

credentials are encrypted when transmitted across trust boundaries,

<pause>

communication crossing trust boundaries ensures data integrity,

<pause>

communication crossing trust boundaries is authenticated and

<pause>

custom protocols use current best practices for authentication and key exchange.

<pause>

How do we use this macro?

<pause>

We can now say that we have a cybersecurity requirement for a communications channel that uses SecOC. All someone has to do is tell us, our design has a CAN bus and we can recommend our SecOC macro requirement.

<pause>

SecOC could also be applied to Ethernet, but there are more sophisticated protocols that we use when we're dealing with Ethernet.

<pause>

I mentioned that this is SecOC security profile one. This is important because not every security profile within SecOC covers all of these requirements. However, this one does.

<pause>

So, this is how you would build a macro. You identify the underlying requirements from the global catalog and create an implements relationship with the macro requirement.

# AVCDL Framework

You'll note that the workflow we just looked at only takes us up to having what are essentially tailored requirements. They don't actually get us attached to the functional requirements.

<pause>

That's easier to talk about if we look at

<pause>

the AVCDL framework.

<pause>

You can see here in the requirements phase that we have the security requirements activity.

<pause>

This is where creation of the security requirements occurs.

<pause>

When we talk about those product-level or element-specific security requirements, we still have to do something to make that connection and we can't do that until we actually have a design.

<pause>

As shown in the diagram where we had A sending a message to B, knowing the actual design is important.

<pause>

Oh yes, we have Ethernet or we have a particular type of credential that we're going to send.

<pause>

That information is key to be able to determine which functional requirements we're going to attach are cybersecurity requirements to.

<pause>

It's the design review, where we go through and look at what is available and make that final set of connections.

<pause>

So, let's look at that workflow now.

# Secure Design Analysis Workflow

Here's our secure design analysis workflow.

<pause>

It's in two parts, the design security analysis and the security requirements attachment.

<pause>

Let's look at each.

# Design Security Analysis

In the design security analysis, what we now have available that we did not have in the requirements phase is

<pause>

the product element design.

<pause>

It's assumed that it's going to take some time for the product team to take their base requirements and come up with a design as to how they're going to implement things. And it is this implementation that we're going to apply

<pause>

our tailored security requirements catalog to,

<pause>

because it's a specific instance of the implementation of the requirements. That design analysis is once again being done by

<pause>

a security SME and a development SME

<pause>

and the outcome of that is going to be

<pause>

all of the identified security requirements which might be applied to each of the elements under consideration.

<pause>

Again, we're generating a report so that we know why we made certain decisions.

<pause>

And the outcome of this analysis is going to have the possibility of

<pause>

feeding back into both the underlying design and

<pause>

the tailored catalog because if the design shows things that we didn't know about when we were looking at the requirements, then we may need to augment the catalog. And if the analysis shows that if you're going to do certain things from a cybersecurity perspective that impacts the underlying design.

<pause>

For instance, if you needed to have a particular level of cryptographic capability that might impact the choice of a microprocessor.

# Attach Security Requirements

Finally, taking

<pause>

all of those security requirements,

<pause>

the development SME will

<pause>

attach them as dependencies to the functional requirements.

<pause>

This is what creates those augmented functional requirements.

<pause>

This is represented here as

<pause>

the updated development requirements.

<pause>

And we generate a requirements report.

# Why Bother?

At this point in time, you may be asking yourself well, this is all well and good, but why bother going to all of this trouble of deriving tailored cybersecurity requirements from a global catalog and augmenting the functional requirements.

<pause>

What does it get us at the end of the day?

<pause>

Let's go back to the diagram that shows us how we get from the cybersecurity requirement to the development task.

<pause>

An interesting thing about a development task is that it is implementing part of a development story and development stories are tracked as are their tasks within the task management system.

<pause>

And that forces you to close out all of your tasks before you can close your story.

<pause>

If you have a requirement that says you have to do something from a cybersecurity standpoint, then that requirement must then be expressed as one of those subsidiary development tasks and the story can't be closed until that requirement is satisfied.

<pause>

In this way, we're ensuring that cybersecurity is on the same level as any other of the requirements that a story is dependent upon.

<pause>

For a second reason, we're going to have to go back and look at our overall dependencies again.

# Overall Dependencies (2)

To review, we start with

<pause>

the functional requirements and

<pause>

the cybersecurity requirements global catalog.

<pause>

We take those two and we build

<pause>

tailored cybersecurity requirements.

<pause>

We take the functional requirements and the tailored cyber security requirements and build

<pause>

augmented functional requirements which feed into

<pause>

the task tracking system.

<pause>

If a problem comes up in the task, when we're testing it, that's managed by

<pause>

the issue tracking system which feeds back into the earlier stages such that the task gets refreshed.

<pause>

Similarly, we take the global requirements catalog and we generate for each of those requirements,

<pause>

some unit test fragments, things that can be used as a basis for a unit test against the implementation of the requirement.

<pause>

We can then take the tailored cybersecurity requirements catalog and those unit test fragments and build

<pause>

cybersecurity fuzz test fragments that we can use for fuzz testing.

<pause>

Now, when the task is implemented, the developer has available to them, both the unit test fragments and the fuzz testing fragments. This greatly reduces the difficulty in actually being able to verify at the task level that the cybersecurity bits are working as intended.

<pause>

Additionally, those same cybersecurity global catalog requirements feed into

<pause>

cybersecurity verification tests.

<pause>

And those verification tests feed into the cybersecurity penetration tests.

<pause>

So that, when we go into the verification phase, we have both system-level verification that we can test against and we have information that we provide for the penetration testers such that their efforts are much more efficiently dispatched.

# Cybersecurity Requirements-related Guidance Documents (2)

Returning again to our guidance documents.

<pause>

We can see that the tailored cybersecurity requirements feed into both

<pause>

the fuzz testing plan which has a corresponding secondary document,

<pause>

the fuzz testing report and

<pause>

the penetration testing plan and its corresponding document,

<pause>

the penetration testing report.

# Unit Testing and Verification Activities

So, to focus on the unit testing and verification activities, we can see that

<pause>

a development task can generate

<pause>

a number of cybersecurity unit tests and

<pause>

cybersecurity fuzz tests.

<pause>

Similarly, a verification story can have

<pause>

multiple cybersecurity verification tests and

<pause>

cybersecurity penetration tests associated with it.

# Cybersecurity Requirements to Cybersecurity Tests

Finally, let's look at how a cybersecurity requirement leads us to these testing and verification activities.

<pause>

We can see that we have a set of cybersecurity requirements.

<pause>

A given cybersecurity requirement can generate any number of

<pause>

cybersecurity unit test fragments.

<pause>

And in conjunction with

<pause>

a tailored cybersecurity requirement, we can generate specific

<pause>

cybersecurity fuzz test fragments.

<pause>

Similarly, a cybersecurity requirement can generate a number of

<pause>

cybersecurity verification tests.

<pause>

And each verification test may draw upon multiple cybersecurity requirements.

<pause>

Finally, each verification test may support multiple

<pause>

penetration tests and each penetration test may draw upon multiple verification tests.

<pause>

It's important to appreciate that there are these one-to-many and many-to-one relationships going on throughout this process.

<pause>

Additionally, having the ability to trace back through the story tracking system into the requirements management system is a big advantage when it comes to being able to completely and sufficiently test that the cybersecurity requirements are being met.

# Further Reading

Here are four documents suggested for more in-depth information.

These are:

<pause>

**Global Security Requirements**,

<pause>

**Product-level Security Requirements**,

<pause>

**Element Cybersecurity Relevancy** and

<pause>

**Design Showing Security Considerations**.

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 to the

<pause>

**Secure Design Review** or

<pause>

**Secure Settings** training if you have completed the

<pause>

**Apply Secure Design** training.

<pause>

If you have also completed the

<pause>

**Threat Prioritization** training, you can proceed to the

<pause>

**Attack Surface Analysis** or

<pause>

**Threat Modeling** 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.