# Title

I'm Charles Wilson, Staff Product Cybersecurity Architect at Torc Robotics.

This presentation will cover an overview of threat modeling.

# 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, and additionally, that you've completed the

* Requirements Taxonomy,
* Security Requirements,
* Secure Design Principles,
* Apply Secure Design

and

* Threat Prioritization Plan training.

This training will cover an overview of Threat Modeling.

# Introduction

When thinking about cybersecurity, and especially when they hear terminology like threat modeling and penetration testing,

most people if they're not cybersecurity practitioners, tend to have this idea that cybersecurity

[break]

is some black art done in darkroom by people with hoodies and that it is impenetrable.

In reality, cybersecurity at the product level is something more akin to

[break]

an open kitchen, where we can observe all the processes that are going on, and look for areas where issues may arise that may result in harm.

Threat modeling is one of these.

We're going to be talking about how we look at the processes that exist

and determine whether there are issues in those processes because of deficiencies in design.

# Design Deficiencies vs Implementation Defects

Let’s review how the AVCDL addresses our two major concerns within cybersecurity, those being design deficiencies and implementation defects.

[break]

The combination of processes addressing both design deficiencies and implementation defects yield the highest level of certainty that we will attain sufficient cybersecurity rigor for the product.

[break]

It is within the processes addressing design deficiencies,

where we find threat modeling.

You'll note it has a duel in the verification phase where we perform a threat modeling review.

This allows us to ensure that those deficiencies that we identified in the design phase were in fact corrected in the verification phase.

# Terminology

Here’s the terminology as used within the context of the AVCDL.

It’s based on the threat modeling glossary proposed by Steven and Migues.

You'll notice that it has two distinct halves.

[pause]

We're going to concentrate on the left half.

Now, unlike the attack surface analysis, which concerns itself only with establishment of the boundaries that essentially form the attack surface,

with threat modeling, we focus more on the actual vulnerabilities that data traversing the boundaries can yield.

These issues are defects.

# What is a Threat Model?

So, what is a threat model?

Fundamentally, it can be a representation of the system's data flows, its data stores, and the interactors.

[pause]

Alternately, it can be considered a collection of views of a system.

[pause]

Finally, it's an engineering design document.

# Threat Modeling Lifecycle 1

Let's consider the NIST Guide to Data-Centric System Threat Modeling and the five stages that it breaks up the threat modeling lifecycle into.

[pause]

The first of these is to diagram.

This entails establishing what the system looks like in terms of the system’s data flows.

[pause]

The second stage is that of identification.

Here you take the diagram created in the first stage and apply a set of rules to identify deficiencies in the design.

[pause]

The third stage is that of investigation, where you determine what the fundamental issues are and how they should be corrected.

[pause]

The fourth stage is that of mitigation, where you go about the process of correcting the deficiencies.

[pause]

Finally, in the fifth stage, you validate that you correctly mitigated the deficiencies.

# Threat Modeling Lifecycle 2

So, now let's look within the context of the AVCDL and see how NIST’s threat modeling lifecycle corresponds to processes within the AVCDL.

If we take the diagram, identify, and investigate stages of the lifecycle we can see that those map onto

[pause]

threat modeling within the design phase.

[pause]

Next, we have mitigate, and it corresponds to

[pause]

all the activities within the implementation phase because those processes all contribute to the mitigation of the deficiency.

[pause]

And finally, validation is mapped to

[pause]

the threat modeling review within the verification phase where we confirm we've eliminated the deficiency.

# Threat Modeling Process

Here’s the threat modeling process.

Within the process, we undertake three individual activities, threat model creation, threat model analysis, and threat candidate triage.

Let’s examine each of these in turn.

# Model Creation

In the model creation activity,

[pause]

the development SME

[pause]

and the threat modeling SME

[pause]

take the system design documentation

[pause]

and create a threat model.

Now let’s consider a simple system and how a threat model for it might be realized.

# Simple System – Block View

Here's a block view of a simple system.

You'll note this is not an automotive system, but it is a system that anyone can relate to.

On the right, you have an interactor, an administrator who is interacting with the system via two modalities,

the first is a web browser and the second a terminal.

[pause]

On the left (the system of interest), we have a web server and a console interface that the administrator is interacting with.

Both are communicating with the core service.

That core service and terminal interface have configuration and application data.

We have multiple processes.

We have read-only and read-write data stores.

And we have different types of data flows.

# Simple System – DFD (high-level view)

We can create a high-level data flow diagram from the block view.

All entities within the block view have a one-to-one correspondence within the DFD realm at this level.

We have only two trust boundaries.

One between the core service, the web server and the console interface.

[pause]

And another between the web server browser pair and the console interface terminal pair.

It’s important to note that this is just the highest level DFD for the system.

Much more detail is exposed during the modeling activity, but that is covered in the model creation procedure.

# Model Analysis

In the model analysis activity

[pause]

The element’s threat model is taken by

[pause]

the threat modeling SME

[pause]

and analyzed with respect to the updated development requirements. Any identified deficiencies are put into

[pause]

a list of threat candidates.

# DFD Analysis

Analysis of the DFD is performed by considering each data flow crossing a trust boundary

and determining whether any of the applicable cybersecurity requirements are violated.

# Resource Access Working Model 1

As a reminder, this is the resource access working model covered in the cybersecurity requirements material.

We have a requester making a request to a resource owner to perform an operation on the resource.

Any status and data are then returned to the requester.

In some situations, the transaction may be subject to logging.

# Resource Access Working Model 2

Let’s consider this from the standpoint of the data flows themselves.

A request comes in, has source, destination, there's payload.

There may be integrity checks.

That payload is a command with optional data.

The command will either perform a write operation, in which case there's a value in,

or a read operation.

The result of the operation comes back as a write status

or read data that goes into a payload which forms a response.

# Resource Access Working Model 3

Finally, in red we see the cybersecurity properties and where controls would be applied.

It’s important to keep in mind that not all cybersecurity requirements (controls) can be reasoned upon within the context of threat modeling.

Threat Candidate Triage

The final activity in the threat modeling workflow is that of triaging.

[pause]

The threat modeling and the development SMEs

[pause]

review each of the threat candidates and determine whether they are valid.

[pause]

Non-applicable threat candidates are used to update the element’s threat model to account for their use case.

[pause]

All valid threats form the list of triaged threat candidates.

[pause]

These will be fed into the standard threat ranking system.

The threat ranking system is described in the threat prioritization plan material.

[pause]

A threat model report is generated which tells us what the disposition of the threat modeling exercise is.

Threat Modeling Feedback

Feedback is core to a cyclic implementation methodology and is the topic of the Understanding Cybersecurity Risk Freshness in an AVCDL Context elaboration document.

As we can see, threat modeling in the design phase has feedback

[pause]

which channels through the threat prioritization process and eventually

[pause]

into the issue tracking system.

[pause]

This feedback can return to the element design review, also in the design phase,

[pause]

or to the element requirements review in the requirements phase.

Threat Mitigation Verification

As mentioned earlier, we perform a threat modeling review

[pause]

in the verification phase to ensure that deficiencies identified

[pause]

by threat modeling

[pause]

in the design phase have been addressed.

V-model View

To illustrate how threat model verification fits into the V-model view,

let’s look at this diagram taken from the elaboration document entitled Understanding Verification and Validation in an AVCDL Context.

It shows how particular artifacts, created in earlier stages,

[pause]

are verified in later stages.

[pause]

In this case, we can see the threat modeling report

[pause]

created in the design phase

[pause]

is later reviewed in the verification phase.

[pause]

Additionally, you can see how particular downstream outputs of activities like threat modeling rely on various upstream outputs of activities

[pause]

like security design review.

Threat Model Review Feedback

Since it occurs in the verification phase,

[pause]

feedback from the threat modeling review may have impact in additional places when compared with that of threat modeling.

As we can see, the threat modeling review

[pause]

in the verification phase has feedback which channels through

[pause]

the threat prioritization process and eventually

[pause]

into the issue tracking system.

This feedback can return to

[pause]

the element implementation

[pause]

in the implementation phase,

[pause]

the element design review

[pause]

in the design phase, or

[pause]

the element requirements review

[pause]

in the requirements phase.

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.

AVCDL on YouTube

A set of training videos have been created to cover various aspects of the AVCDL that follow the training path found in the AVCDL primary document.

It's important to note that all AVCDL materials except for the introductory blog posts are intended for product cybersecurity engineering practitioners.

Training Path

Once you have completed all the trainings related to threat modeling, you can proceed to one of the other trainings at this level.

These are:

* Attack Surface Analysis
* Penetration Testing

and

* Vulnerability Identification

Additionally, if you’ve completed Secure Coding and its prerequisites you can also proceed to

* Static analysis
* Dynamic analysis
* Fuzz testing

and

* Secure Code Review.

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.