

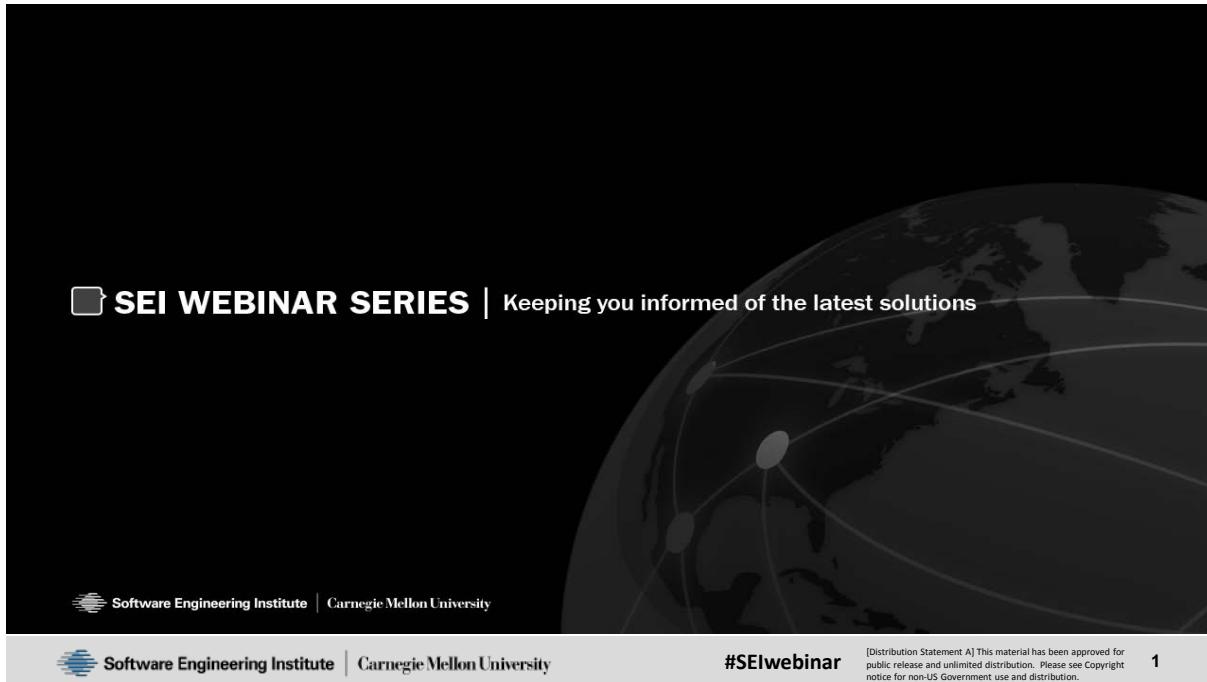
# Secure Requirements Engineering Part 2

## Table of Contents

SEI WEBINAR SERIES   Keeping you informed of the latest solutions.....	3
Carnegie Mellon University.....	3
Copyright 2016 Carnegie Mellon University.....	4
Security Requirements Engineering.....	4
Topics .....	5
Software Assurance (SwA) .....	6
Software Assurance: Lifecycle Focus .....	7
Software Security Requirements .....	8
Polling Question .....	9
Security Requirements Engineering: Key Activities 1 .....	10
Focus of this Module.....	13
Security Requirements Engineering.....	15
Polling Question .....	16
Security Engineering Risk Analysis (SERA) .....	17
SERA Approach: Focus on Mission Impact.....	18
SERA Method: Four Tasks .....	20
Pilot Example: Wireless Emergency Alerts (WEA) 1 .....	21
Establish Operational Context (Task 1).....	23
SERA Task 1: Operational Views .....	24
SERA Task 1: WEA Operational Models .....	26
SERA Task 1: Data Security Goals (Excerpt) .....	27

Identify Risk (SERA Task 2).....	28
SERA Task 2: Threats Selected for Analysis.....	29
SERA Task 2: R1 Threat Sequence.....	30
SERA Task 2: Enablers .....	31
SERA Task 2: R1 Stakeholder Consequences .....	32
SERA Task 2: Amplifiers.....	33
Analyze Risk (SERA Task 3).....	34
SERA Task 3: R1 Risk Analysis.....	35
Develop Control Plan (SERA Task 4).....	36
SERA Task 4: Prioritized Risk Spreadsheet.....	37
SERA Task 4: Controls.....	38
SERA Task 4: CMSP Cybersecurity Guidelines.....	39
SERA Task 4: Controls with Requirements Implications .....	40
Security Requirements Engineering and SERA .....	41
Polling Question.....	44
Key Points.....	47
SEI WEBINAR SERIES   Keeping you informed of the latest solutions.....	52

## SEI WEBINAR SERIES | Keeping you informed of the latest solutions



 Software Engineering Institute | Carnegie Mellon University

 Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

1

## Carnegie Mellon University

# Carnegie Mellon University

This video and all related information and materials ("materials") are owned by Carnegie Mellon University. These materials are provided on an "as-is" "as available" basis without any warranties and solely for your personal viewing and use.

You agree that Carnegie Mellon is not liable with respect to any materials received by you as a result of viewing the video, or using referenced websites, and/or for any consequences or the use by you of such materials.

By viewing, downloading, and/or using this video and related materials, you agree that you have read and agree to our terms of use ([www.sei.cmu.edu/legal/](http://www.sei.cmu.edu/legal/)).

Distribution Statement A: Approved for Public Release; Distribution is Unlimited

© 2016 Carnegie Mellon University.

 Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

2

## Copyright 2016 Carnegie Mellon University

Copyright 2016 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0003493



Software Engineering Institute

| Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

3

## Security Requirements Engineering

# Security Requirements Engineering

Christopher Alberts

CERT® Division

Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

Software Engineering Institute

| Carnegie Mellon University

#SEIwebinar

© 2016 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

\*\*022 Presenter: So, our  
next talk is going to be security  
requirements engineering by Chris  
Alberts. Chris Alberts is a principal

engineer in the CERT division at SEI where he leads applied research projects in software assurance and cybersecurity. His research interests include risk analysis, security requirements engineering, measurement analysis, modeling and simulation, and assessment. And he has published two books and over forty technical reports and articles.

So, Chris is queued up. Again, Mark is going to stay on stage with us as a facilitator to continue the conversation. But Chris, welcome. All yours.

Presenter: Thanks.

## Topics

### Topics

Background

Security Engineering Risk Analysis (SERA) Method

Summary



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

23

\*\*023 Okay, I'm going to talk to three topics today. I'll give some background information talking about some of the basic concepts behind security requirements engineering.

Then I'll look at the security engineering risk analysis method, or SERA. That's going to be the focus of the talk. I'm going talk basically showing how we can integrate-- better integrate risk analysis into the requirements process. And then I'll summarize with a few key points.

## Software Assurance (SwA)

### Definition

- "The level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at anytime during its lifecycle, and that the software functions in the intended manner." <sup>1</sup>



### Key Aspects of SwA

- Trustworthiness – No exploitable weaknesses exist, either maliciously or unintentionally inserted.
- Predictable Execution – When executed, software functions as intended.

1. National Information Assurance Glossary CNSS Instruction No. 4009; DoDI 5200.44 p.12



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

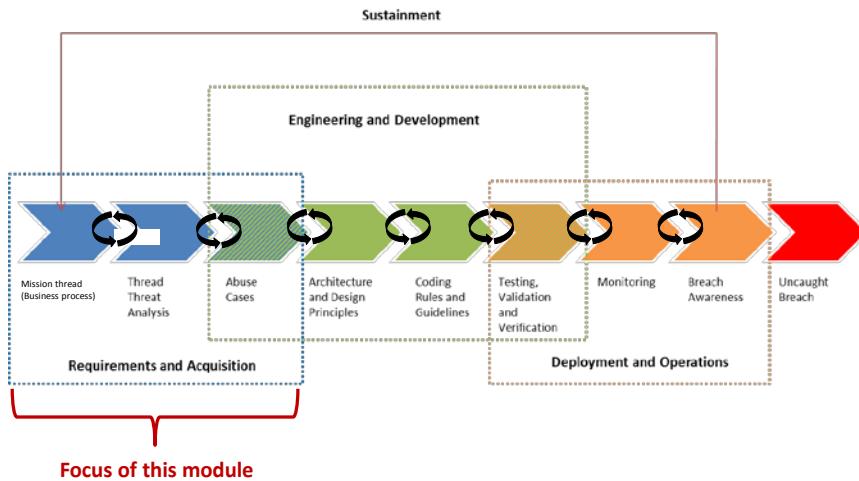
[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

25

\*\*025 So, let's start with the background. Starting with software assurance and the definition and kind of what we think about when we talk about software assurance, two key aspects, predictable execution, and there we're really looking at does the software function as intended, and then trustworthiness, are there any exploitable weaknesses in the software. And what we're trying to do is establish a level of confidence in those two key aspects. And requirements is a key piece of that.

## Software Assurance: Lifecycle Focus

### Software Assurance: *Lifecycle Focus*



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

26

\*\*026 And so, we're looking at-- this is the lifecycle model that Mark showed just a few minutes ago. And we're looking at the very early part of the lifecycle, at defining the requirements and focusing on the early acquisition aspects of software.

## Software Security Requirements

# Software Security Requirements

Features (e.g., controls or constraints) that specify how to preserve the confidentiality, integrity, and availability of critical system data<sup>1</sup>



1. Khan, M. U. A. & Zulkernine, M. "On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software," 353-358. *Computer Software and Applications Conference, 2009. COMPSAC '09. 33rd Annual IEEE International* (Volume:2 ). Seattle, WA: IEEE Press, 2009.

\*\*027 So, let's talk about what software security requirements are. I define these as features, such as controls or constraints, that specify how to preserve the confidentiality, integrity, and availability of critical data in the system. And so, you'll hear me reference CIA, confidentiality, integrity, and availability, multiple times throughout this talk and because that kind of forms the goals of what we're trying to do with software security requirements.

## Polling Question

### Polling Question

Are you experienced in developing security requirements?

Answers:

- Yes
- No

\*\*028 Presenter: Okay, a polling question again, like I said, we're going to have multiple of these throughout the day to get an idea of who's with us in the audience. And that's going to help Chris tailor some of his speaking points. But the question that's going to pop up now is, "Are you experienced in developing security requirements?" And that is a simple yes or no question. And you will have that on your screen now.

And we're going to go back to Chris.  
Go ahead Chris.

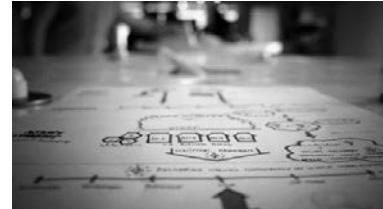
Presenter: Okay. Should I just head to the next slide?

Presenter: Yeah, go ahead to the next slide.

## Security Requirements Engineering: Key Activities 1

### Security Requirements Engineering: Key Activities<sup>1</sup>

1. Agree on definitions.
2. Identify system assets and security goals.
3. Perform security risk analysis.
4. Elicit security requirements.
5. Categorize security requirements.
6. Prioritize security requirements.
7. Inspect security requirements using a well-defined method (e.g., Fagan inspections).



1. Derived from the Security Quality Requirements Engineering (SQUARE) Method as defined in Allen, Julia H.; Barnum, Sean; Ellison, Robert J.; McGraw, Gary; & Mead, Nancy R. *Software Security Engineering: A Guide for Project Managers*. Addison-Wesley, 2008.

\*\*029 And we'll give them about fifteen or twenty seconds to vote, and I'll give you results.

Presenter: Okay, with software security requirements, here's some of the key activities. I'll kind of use these as kind of the anchor of the talk. And then I'll show you next where we're going to focus. Start by agreeing on definitions. You want to make sure that everyone is talking the same language. A lot of problems come about with respect to security because people often have different views of what terms mean and often there are different variations on terminology. So, when people come into a requirements situation, they may have different ideas about what things mean. You want to get everyone on the same page.

The second key activity is to identify system assets and security goals. So, this starts out by looking at what's the critical data that the system stores, processes, and transmits. And once you understand that, you want to know what's important about it from a confidentiality, integrity, and availability perspective. And now, you have the critical data and the security goals.

The third step then is to look at the risks. And much like Chris's previous presentation, kind of what they were doing when you think about it is they were looking at what they knew about the system. And they were starting to think about how can we attack it. Well, that's what you're doing in step three here. You're trying to think of how can we, based on what we know currently-- remember, we're early in the lifecycle. So, we don't have a full picture. But we had some logical diagrams that we can look at how things are interconnected. So, we can make some plausible guesses. And so you do the risk analysis. And then based on that you decide are there design weaknesses.

And for those design weaknesses, that feeds into step four here, elicit security requirements. You build requirements for those weaknesses. And then categorize the requirements, which essentially means you map them back to the security goals that you defined. Then develop priorities, which ones are most important, which ones are least

important because there's always tradeoffs. And you have to make sure you focus on what's important to address.

And then the last step is to inspect the security requirements. Here what you want to do is to look for weaknesses or problems with the requirements and get in and correct those flaws as early as possible.

Presenter: So, Chris to wrap up our polling question real quick, we had fifty-seven percent with no, they're not experienced in developing security requirements, and forty-three percent yes.

Presenter: Okay.

Presenter: So, hopefully that can tailor your talk a little.

Presenter: Okay.

## Focus of this Module

### Focus of this Module

1. Agree on definitions.
2. Identify system assets and security goals.
3. Perform security risk analysis.
4. Elicit security requirements.
5. Categorize security requirements.
6. Prioritize security requirements.
7. Inspect security requirements using a well-defined method (e.g., Fagan inspections).

This module examines the role of risk analysis during security requirements engineering

\*\*030 And so, here's where we're going to focus in this module. And we're going to look at primarily at steps two and three, identifying the assets and goals, and then performing the risk analysis. But I'll show you later on how what we do in these steps actually looks at some of the subsequent steps as well

Presenter: Chris, could you explain a little bit about where these steps came from?

Presenter: Sure, well there are a lot of different methods out there. I took these and derived them from a method that we developed at the SEI called SQUARE, Security Quality Requirements Engineering. And that has-- defines a set of steps. And I kind of took out the key steps that really focused on some of the, I

think, the key high points that you need to look at in security requirements engineering. Anything you want to add on SQUARE because I know you know a lot about that too?

Presenter: That's why I was going to ask you some of the method and the history behind SQUARE and how well it's been used in practice.

Presenter: SQUARE's actually a fairly mature product. It's been around for more than a decade. And it's been developed, Nancy Mead was the lead developer at the SEI. She'd worked with a number of students in the master of software engineering program at the SEI to develop the technique. And they applied it with a variety of different industry organizations. And they built several variations on it for acquisition and other specific aspects of the engineering process. And they've created some tools to support it. So, there's a lot in place for that. And it walks you through these steps and really helps guide you into applying the method.

Presenter: So, this is another example of things that we already know how to do well. It's merely taking the discipline to apply them.

Presenter: Right, and getting people to adopt them and to use them. Yes, exactly. Okay.

## Security Requirements Engineering



Security Requirements Engineering

# Security Engineering Risk Analysis (SERA)



Software Engineering Institute | Carnegie Mellon University

© 2016 Carnegie Mellon University  
[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

\*\*031 So, let's take a look at the risk aspects of security requirements engineering with the SERA method.

## Polling Question

### Polling Question

Are you experienced in assessing security risk?

Answers:

- Yes
- No

\*\*032 Presenter: And that leads us to another polling question to help Chris tailor his talk. And we'd like to know, "Are you experienced in assessing security risk?" A simple yes/no. And we can turn it back to you Chris. And I'll chime back in with the results in about a couple seconds.

Presenter: Great, thanks Shane.

## Security Engineering Risk Analysis (SERA)

# Security Engineering Risk Analysis (SERA )

### **What**

- A systematic approach for analyzing complex security risks across the lifecycle



### **Why**

- Build security into software-reliant systems
- Address design weaknesses as early as possible (e.g., requirements, architecture, design)

### **Benefits**

- Correct design weaknesses before a system is deployed
- Reduce residual cybersecurity risk in deployed systems
- Ensure consistency with risk management standards



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see 33 Copyright notice for non-US Government use and distribution.

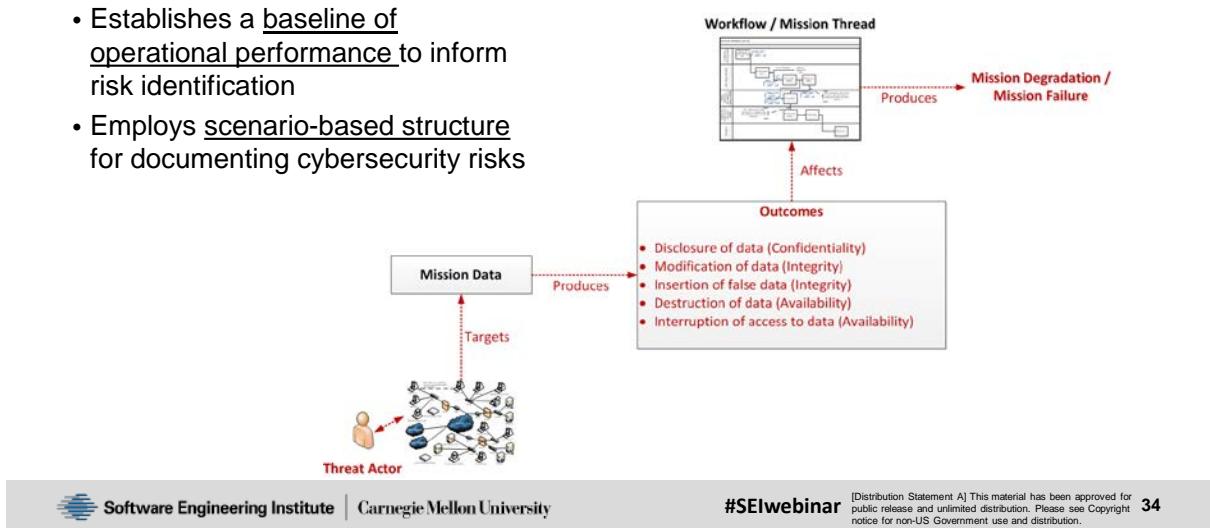
\*\*033 So, this method that we developed is a systematic approach for analyzing security risk across the lifecycle. And we're looking at trying to get at some of the complexities of risk. And I'll kind of talk to that in some subsequent slides. What we're looking is to build software-- build security into system, so starting early in the lifecycle. And we-- you can actually recursively apply this at different points in the lifecycle. And so, we want to address these design weaknesses as early as possible, create requirements for them so we can start to mitigate them, and then ultimately deploy systems with a reduced residual cybersecurity risk.

## SERA Approach: Focus on Mission Impact

### SERA Approach: *Focus on Mission Impact*

SERA analyzes the mission impact of data security breaches.

- Establishes a baseline of operational performance to inform risk identification
- Employs scenario-based structure for documenting cybersecurity risks



\*\*034 And to close out our polling real quick, we were at fifty-five percent no, not experienced in assessing security risk and forty-five percent yes.

Presenter: Okay, so I'm going to go through some of the basics of what we do in the SERA method. I'm going to start with something that's kind of different that we're trying to incorporate into our risk analysis methods than some of the techniques that we worked on ten years or so ago. And that's a scenario-based approach. So, this picture I think kind of gets to that idea. You start with a threat actor. And, in fact, we can accommodate scenarios with multiple threat actors accessing the infrastructure exploiting weaknesses to target mission data. And you want to do-- and the threat actor is trying

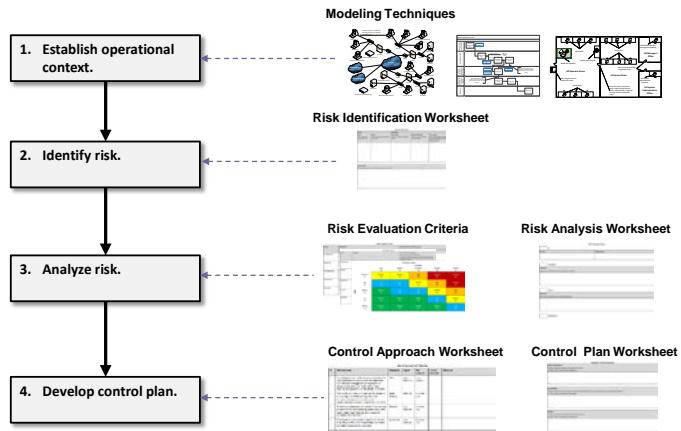
to achieve some kind of a goal. That's to some adverse outcome related to the data, disclosure of data, modification of data, affecting the availability of data. And so, what we're seeing here is that those mapped to confidentiality, integrity, and availability.

Once you do that, the question is what happens then? Well then we look at how does that affect the mission. So, in SERA our focus is on mission impact. So, we look at workflows, which are-- the other synonym for that is mission thread or business process. We map those out, look at where the data affects the business process, and just see what might happen. We use that to help us project the consequences when we're doing the risk analysis. And then those adverse consequences to the outcome can lead to mission degradation or mission failure.

And so, one of the key aspects in doing these scenarios is to first start out by understanding how the system performs under normal circumstances. Identifying what we call the baseline of operational performance.

## SERA Method: Four Tasks

### SERA Method: *Four Tasks*



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

35

\*\*035 And so, that's the first task in the four tasks that we've defined in the SERA method. First we understand the operational context. And I'll talk to each of these specifically as we move through the talk. Then we look at identifying the risk scenarios, analyzing them, and then developing control plans.

## Pilot Example: Wireless Emergency Alerts (WEA) 1

### Pilot Example: Wireless Emergency Alerts (WEA)<sup>1</sup>

WEA is a major component of the Federal Emergency Management Agency (FEMA) Integrated Public Alert and Warning System (IPAWS).

- **Initiator** – decides to issue an alert (e.g., weather alert, AMBER alert)
- **Alert originator (AO)** – sends alerts to mobile devices in the targeted area
- **FEMA** – receives and processes alerts
- **Commercial mobile service provider (CMSP)** – receives and processes alerts
- **Recipients** – receive alerts automatically

1. Alberts, C.; Woody, C.; & Dorofee, A. *Wireless Emergency Alerts CMSP Cybersecurity Guidelines* (CMU/SEI-2015-SR-020). Software Engineering Institute, Carnegie Mellon University, 2015.  
<http://www.firstresponder.gov/TechnologyDocuments/Wireless%20Emergency%20Alerts%20CMSP%20Cybersecurity%20Guidelines.pdf>



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see 36  
Copyright notice for non-US Government use and distribution.

\*\*036 So, the other thing I want to point out is all the examples that I'll show here is from a study that we recently completed on the wireless emergency alert service. And that is-- WEA, as it's called, is a major component of FEMA's integrated public alert and warning system, or IPAWS. And so, the idea here is that this I getting emergency alerts on your cellphones. So, I'm sure a lot of you have had weather alerts and things like that on your phones. And so, we did a study to look at some of the risks in this WEA service.

Just to walk you through the basic roles because some of this will come up in subsequent slides, it starts out with an initiator. So, if we're thinking about a weather alert, the initiator would be a meteorologist. And say that meteorologist says severe

weather is going to come through some geographic area like a county you're in. That meteorologist will recommend issuing an alert.

The alert originator, in this case the National Weather Service, would send the alert out. But it doesn't go directly to your phone from the National Weather Service. It goes to FEMA who processes and formats it for the commercial mobile service providers. These are the carriers like Verizon, and Sprint, and the other carriers. And then they format it and send it to the technology that they support. And it gets to your cellphones. So, that's how what they call the WEA pipeline works.

And we're going to look at the CMSP, or commercial mobile service provider, part of that in this talk. And that's what we focused on in this study. You'll notice that there's a footnote here at the bottom of the slide. That's the actual details of the study. I'm just going to be able to skim the surface in this short presentation. If you want the details, you can go to that link. Also on the materials tab, it's not there now, but tonight we'll add the final report to the tab so that you can access it from there directly as well.

## Establish Operational Context (Task 1)

### Establish Operational Context (Task 1)

The operational environment for the system of interest is characterized to establish a baseline of operational performance.

Steps	
1.1	Determine system of interest.
1.2	Select workflow/mission thread.
1.3	Establish operational views.

\*\*037 So, we have in task one, we'll look at three basic steps, determine system of interest, select the workflow or mission thread, and establish the operational views.

## SERA Task 1: Operational Views

### SERA Task 1: *Operational Views*

Mission thread / workflow

Technology (e.g., system, system of systems, architecture, network)

Use case

Data

Physical

Stakeholder

Others as needed



Software Engineering Institute

| Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved  
for public release and unlimited distribution. Please see 38  
Copyright notice for non-US Government use and distribution.

\*\*038 So, what I mean by operational views is what we want to do is we want to model what's going on in the operations. Now, if you go back to what Chris was talking about in the previous presentation of how they looked and gathered information, they were looking at how things work. And so, what we want to do when we're doing the risk analysis is we start by saying what is the mission thread, what system that we're developing, what is it supporting, what business processes.

Presenter: So, would you say the analogy here in this example these are, I would say, big objects, the mobile carriers and so on.

Presenter: Yeah.

Presenter: But they have analogs in the Jeep example where the systems are the cell components that Fiat got from various places whether it was Harman, whether it was Sprint, whether it was the ECUs that are inside there. But again we have these components, and they're all connected. And together they form some sort of operational context.

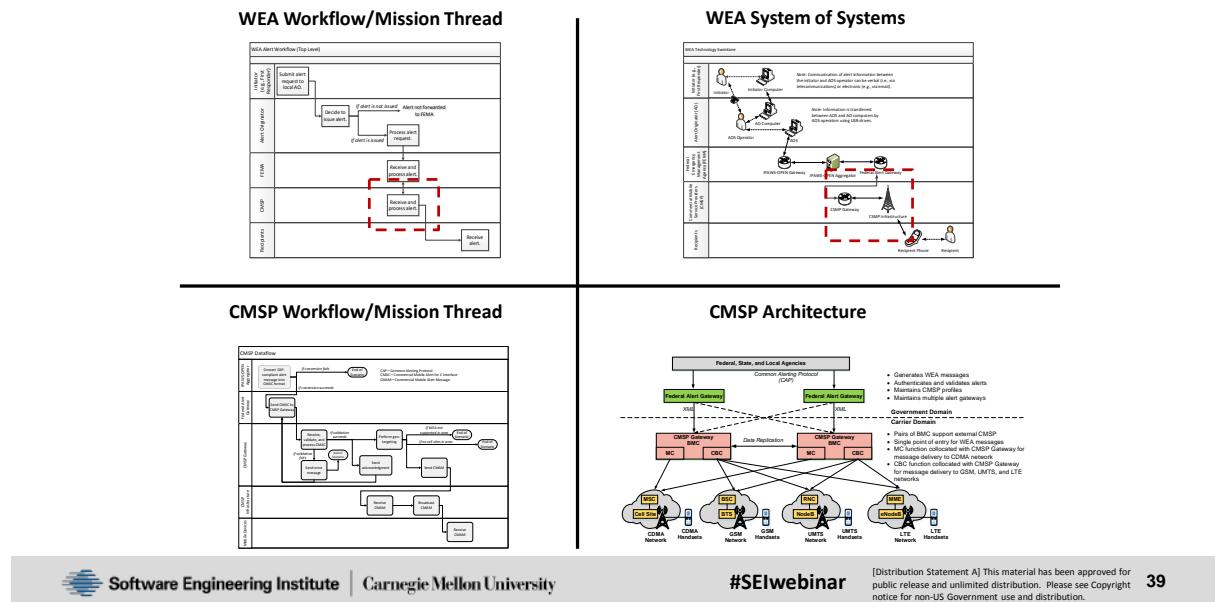
Presenter: Right. And so the same principles that I'm talking about apply at the system level with the Jeep example. And we're looking at a system of systems level in this example, but the same thinking can be applied. So, the other views we look at are things like technology views that we know about, system diagrams. Can we look at the architecture network diagrams if they apply? We always like to look at use cases, how do people legitimately use the system because that helps us determine how can we abuse the system or misuse the system. What are some of the abuse and misuse cases? We always look at the data flows because what we're trying to do is figure out how we can corrupt the data, or how we can find data that we can view or make it unavailable. We may look at physical diagrams like facility layouts if we're looking at cyber-physical attacks and things of that nature.

So, we want to really understand what the system looked like in its operational context. Again, since we're early in the lifecycle, there might be some guesswork involved.

But we do know some of this information.

## SERA Task 1: WEA Operational Models

### SERA Task 1: WEA Operational Models



\*\*039 So, I'll show you kind of the thinking of how some of the modeling-- we put the modeling together. I talked about the five roles of WEA earlier. And this slide kind of shows you. It's a swim lane diagram. Each lane represents one of the roles. And so we start with the swim lane diagram. Then we look at what systems support each of those activities. So, again, as Mark was saying, these are the big level, big systems that support a fairly-- a work flow that spans multiple organizations.

Now, what you'll see in the top left quadrant, now there's a red dashed box. That's the CMSP box, so the carriers. And what we do then

is we take a look at what's really going inside that box. Let's do a deep dive into that. And we get the detailed workflow.

Now in the same analogy, we look at the system of interest but in the system of systems diagram. And we can explode it and look at the details of the architecture. So, we're taking-- if you look at the top diagrams, that's kind of the forty-thousand-foot level. The bottom diagrams are more at the five-thousand-foot level. And you can dive down to any level of detail that you need for the analysis that you're doing.

### SERA Task 1: Data Security Goals (Excerpt)

### **SERA Task 1: *Data Security Goals (Excerpt)***

Data Asset	Form	Confidentiality	Integrity	Availability
Alert message	Electronic	There are no restrictions on who can view this data asset (public data)	The data asset must be correct and complete (high integrity).	This data asset must be available when needed (high availability).
Geo-targeting data	Electronic	There are no restrictions on who can view this data asset (public data)	The data asset must be correct and complete (high integrity).	This data asset must be available when needed (high availability).



\*\*040 So, this slide shows a couple of the key data assets, critical data assets that we identified for this study, and then the confidentiality, integrity, and availability goals that

were assigned to those assets. And so, the idea here now is if you think back to the step two or activity two of the security requirements engineering, now we know what the key assets are. And we know what's important about them.

## Identify Risk (SERA Task 2)

### Identify Risk (SERA Task 2)

Security concerns are transformed into distinct, tangible risk scenarios that can be described and measured.

Steps	
2.1	Identify threat.
2.2	Establish consequence.
2.3	Identify enablers and amplifiers.
2.4	Develop security risk scenario.



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see 41  
Copyright notice for non-US Government use and distribution.

\*\*041 And so we can move to the next task, which is identifying or starting to elicit and document the risks. So, we start with what are the threats. And then based on the threats, we look at the consequences. We then look at what enables each threat to occur and then what can make the consequences worse, the amplifiers. And then we develop the security risk scenarios.

## SERA Task 2: Threats Selected for Analysis

### **SERA Task 2: *Threats Selected for Analysis***

#### **R1. Insider Sends False Alerts**

R2. Inherited Replay Attack

R3. Malicious Code in the Supply Chain

R4. Denial of Service



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved  
for public release and unlimited distribution. Please see 42  
Copyright notice for non-US Government use and distribution.

\*\*042 Okay, in this particular example, we looked at four scenarios. And I'm going to focus on the top one, insider sending false alerts. And what we did here is-- or the basic gist of this risk is that a disgruntled insider decides to plant malicious code into the codebase for the CMSP system. And then that will repeatedly send a nonsense message to recipients in a targeted non-geographic area. So, the idea is to annoy people, get them to be angry at the carrier because this is one way a disgruntled insider who has the technical skills could get negative attitudes towards the carrier.

Presenter: Or so again in the Jeep example that Chris and I were considering, we have all these suppliers. You could have a disgruntled employee in one of the

suppliers to some of the modules that also may want to do a similar kind of advanced persistent threat, or some other kind of malicious activity.

Presenter: Right, and in fact in this study, we looked at one of those. And risk three is actually looking at malicious code in the supply chain. And again there is was somebody at- an insider in the supply chain doing the same type of thing. So, exactly, I mean that's an important piece and because most of these organizations acquire their systems from external groups.

## SERA Task 2: R1 Threat Sequence

### **SERA Task 2: *R1 Threat Sequence***

- T1. The insider is upset upon learning that he is not receiving a bonus this year and has been passed over for a promotion.
- T2. The insider begins to behave aggressively and abusively toward his coworkers.
- T3. The insider develops a logic bomb designed to replay a nonsense alert message repeatedly.
- T4. The insider uses a colleague's workstation to check-in the modified code with the logic bomb.
- T5. Seven months later, the insider voluntarily leaves the company for a position in another organization.
- T6. Twenty-one days after the insider leaves the carrier, the logic bomb is activated automatically.
- T7. The malicious code causes the carrier's WEA service to send a nonsense WEA alert repeatedly to people across the country.

\*\*043 I'm not going to go through the details of this. If you're interested in the details of any of these slides, feel free to look at the report. This is- - what we do here is we break down each thread. I kind of gave you the gist of

it a minute ago. We look at a sequence of steps, what does it take to make that happen. Starting with the first step, the insider becomes disgruntled. The last step is the malicious code sends nonsense messages repeatedly.

## SERA Task 2: Enablers

### SERA Task 2: *Enablers*

#### Threat Step

T7. The malicious code causes the carrier's WEA service to send a nonsense WEA alert repeatedly to people across the country.

#### Enabler

Insufficient capability to check message content can allow illegitimate alert messages to be broadcast automatically to designated mobile devices.

An *enabler* is a condition or circumstance (e.g., weakness, vulnerability) that facilitates a threat's occurrence.



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

44

\*\*044 And then for each threat step, we look at one or more enablers. And we define an enabler as a condition or circumstance that facilitates a threat's occurrence, or facilitates that step's occurrence. So, in this case, when it's sending out messages, you want to know is before the messages actually get sent to the recipients, are they looking and doing any filtering of the content to see if they can pick up anything that's odd or unusual and stop it before it gets sent so they can do a final check.

## SERA Task 2: R1 Stakeholder Consequences

### **SERA Task 2: R1 Stakeholder Consequences**

Recipients of the message quickly become annoyed at receiving the same nonsense message repeatedly. (Recipients)

Many recipients complain to the carrier's customer service operators. (Recipients)

A large number of recipients turn off the WEA function on their phones. Many will not turn the WEA service back on. (FEMA, Carrier)

The carrier responds to the attack. It removes the malicious code from its infrastructure. The cost to do so is considerable. (Carrier)

People leave the carrier for another carrier because of the incident. (Carrier)

People lose trust in the WEA service. (FEMA, Carrier)

\*\*045 Likewise, then with consequences, we look at the range of consequences. And on this slide, you see we look at impacts to the-- or consequences with respect to the recipients, to FEMA, the carriers. And so it starts with people becoming annoyed. They complain to their carrier. In some cases, they may, if the situation gets bad enough, they may decide they want to switch to another carrier. And it may actually cause people to lose trust in the WEA of service itself. So, a lot of various impacts that we look at.

## SERA Task 2: Amplifiers

### SERA Task 2: *Amplifiers*

#### **Consequence**

Recipients of the message quickly become annoyed at receiving the same nonsense message repeatedly.

#### **Amplifier**

Knowledge of the system's geo-targeting capability can enable the attacker to expand the geographic area being targeted and affect a greater number of recipients.

An *amplifier* is a condition or circumstance that increases the consequence triggered by the occurrence of a threat.



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

46

\*\*046 And when we look at the impacts we look at what we call amplifiers. What can make them worse? And then this example here is the geo targeting capability. That says what area should get this message. Well, if you know how to exploit that, you can actually give it to a broader range of people, and in this case, annoy more people. And so, we want to look at that because--

## Analyze Risk (SERA Task 3)

### Analyze Risk (SERA Task 3)

Each risk is analyzed in relation to predefined criteria.

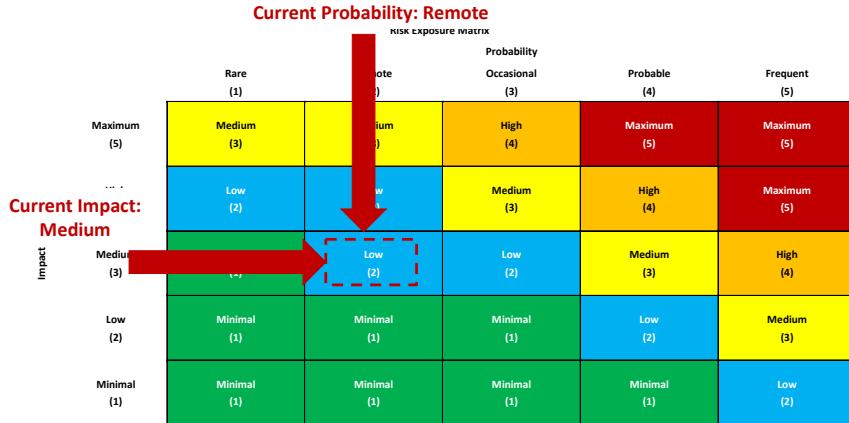
Steps	
3.1	Establish probability.
3.2	Establish impact.
3.3	Determine risk exposure.

\*\*047 When we get to the control section, amplifiers and enablers become important as helping us determine how to control the risk.

Task three, I'm going to go through this one pretty quickly. For those of you who are not familiar with a risk analysis--

## SERA Task 3: R1 Risk Analysis

### SERA Task 3: *R1 Risk Analysis*



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

48

\*\*048 We use a fairly basic risk analysis, qualitative risk analysis process. For each of the scenarios, we look at-- we subjectively estimate what we think the probability is, what the impact might be, and then look at the risk exposure.

## Develop Control Plan (SERA Task 4)

### Develop Control Plan (SERA Task 4)

Control plans are developed and documented for all security risks that are not accepted.

Steps	
4.1	Prioritize risks.
4.2	Select control approach.
4.3	Establish control actions.

\*\*049 And that's done as really a feed into task four where I want to spend the bulk of the rest of the time.

## SERA Task 4: Prioritized Risk Spreadsheet

### SERA Task 4: *Prioritized Risk Spreadsheet*

ID	Risk Statement	Imp	Prob	RE
R4	Denial of Service	Max	Rare	Med
R1	Insider Sends False Alerts	Med	Remote	Low
R2	Inherited Replay Attack	Med	Remote	Low
R3	Malicious Code in the Supply Chain	Med	Rare	Min

*Note:* A control plan will be developed for all security risk scenarios with an impact of medium or greater.

\*\*050 So, we use those impact probability and risk exposure estimates to rank the scenarios. And we key off of impact because what we want to look is we want to make sure is we consider the catastrophic or rare events where you have a very low probability, very high impact. We want to keep those in the mix and consider those as part of the mitigation because a lot of the security breaches that we hear about, you look at them. It's just like what Chris was talking about in the last segment. You think about boy, that's a lot of work. It's highly unlikely that somebody would be able to pull something like that off. Like you were saying, it's so complex. And it seems like it's hard to do. But a lot of times, the big, high-impact security attacks are the ones that are hard to do.

Presenter: And so, the thought here is it's not really about a technical argument, can you do something, or can't you do something, or even how hard it is. But what's the risk that you're willing to take? And then deciding what's important to focus on versus what might be able to be deferred.

Presenter: Right. Right. And so, we decided that we would mitigate any of the scenarios that we identified that were medium or above. And in this case, all of them fall into that category.

## SERA Task 4: Controls

### SERA Task 4: *Controls*

#### Threat Step

T7. The malicious code causes the carrier's WEA service to send a nonsense WEA alert repeatedly to people across the country.

A *control* is a safeguard or countermeasure to

- Recognize, resist, and recover from security risks
- Counteract identified enablers and amplifiers

#### Enabler

Insufficient capability to check message content can allow illegitimate CMAM messages to be broadcast automatically to designated mobile devices.

#### Control

The carrier monitors messages for suspicious content (e.g., illegitimate messages, duplicate messages) and responds appropriately.

\*\*051 So, we look at control. So, for each enabler and amplifier that we've identified, we identify a control, which is a safeguard or counter measure to counteract the enabler or the amplifier. In this case, the control

is doing monitoring of the messages for content. And so, you might be able to find some unusual content that's coming through the system. And so, we gather those.

## SERA Task 4: CMSP Cybersecurity Guidelines

### SERA Task 4: *CMSP Cybersecurity Guidelines*

The CMSP Cybersecurity Guidelines comprise 35 high-priority security controls that address the four WEA risk scenarios included in this study

Controls were identified in the following areas:

- Human Resources
- Training
- Contracting
- Physical Security
- Change Management
- Access Control
- Information Management
- Vulnerability Management
- System Architecture
- System Configuration
- Code Analysis
- Technical Monitoring
- Independent Reviews
- Incident Response
- Disaster Recovery

\*\*052 And in this case, we came up with thirty-five high priority security controls in the fifteen areas that you see on this slide. We had administrative areas like human resources and training. We had some physical security controls that we identified and also technical security controls in a variety of areas. So, with this, the idea then and taking this back now into how we feed this back into the security process, not all of these controls have design implications. Like for instance the human resource processes were operational processes, and they had nothing to do with the design of the

system, same thing about the training aspects as well.

## **SERA Task 4: Controls with Requirements Implications**

### **SERA Task 4: *Controls with Requirements Implications***

#### Access Control

- The carrier controls access to sensitive information based on organizational role.

#### System Architecture

- The carrier's WEA alerting system has a backup capability that uses a separate communication channel.

#### Technical Monitoring

- The carrier monitors messages for suspicious content (e.g., illegitimate messages, duplicate messages) and responds appropriately.
- The carrier monitors the WEA alerting system for abnormal activity and responds appropriately.



Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see 53 Copyright notice for non-US Government use and distribution.

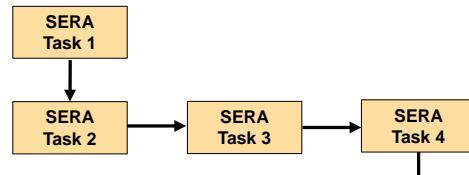
\*\*053 But these three areas for instance are examples of controls that have requirements implications. We found that the access control was important, so making sure that people were authorized to only look at the information that they should have access to based on job responsibility. The system architecture, are there backup communication channels? So, at the main communication channel-- so, in the WEA service, you're sending messages out to the community. If your main communication channel goes down, do you have a backup? And is it a non-redundant backup so that you, if there's a denial of service attack, you can mitigate that attack. And the technical monitoring, we

talked about monitoring the messages, monitoring for abnormal activity in the system. These all have implications for requirements. And--

## Security Requirements Engineering and SERA

### Security Requirements Engineering and SERA

1. Agree on definitions.
2. Identify system assets and security goals.
3. Perform security risk analysis.
4. Elicit security requirements.
5. Categorize security requirements.
6. Prioritize security requirements.
7. Inspect security requirements using a well-defined method (e.g., Fagan inspections).



\*\*054 Presenter: So, those are common themes not just for WEA or the JEEP, but understanding your security needs from an authentication authorization, who's allowed to do things.

Presenter: Right.

Presenter: The whole system reliability that is denial of service as opposed to a particular attack, something that needs to be accommodated, and what are the architectures then to support this? It's not just figuring out some particular kind of security control that goes in there. It's a much larger issue.

Presenter: Right. Right. In most cases, yes. And so what you're trying to do is input these. So, now if you're looking at access control, you should craft a requirement that says that you need to have features in the system that allow for authorization to certain resources in the system. And then not everybody can have obviously access to everything in the system. So, you start partitioning who has access to what.

And so, when you go through the whole list of controls that come out of the risk assessment, you feed them into the requirements elicitation activity in step four. And now, you can start crafting what should be built into the system. And then in step five in doing the security risk analysis, you've already identified your mitigations or controls to the threats which links it to the data. And it also is then linked to the security goals, confidentiality, integrity, and availability. So, you've done a lot of the categorization already.

The risk helps you with the prioritization then because as you look at the risks, you say which risks are most important. And then you can look at what controls are in place. You start to look at cost benefit analysis and things of that nature. And then you start to decide and make your choices as to-- again, there's a tradeoff, performance versus security, and cost versus security. And so, a lot goes into the tradeoff space.

Presenter: I assume that in these kinds of systems and also in cyber-physical system, this concept of maintenance and authority, you feel like the logical equivalent of the system administrator plays a big role because we've learned in IT you just shouldn't give the system administrator the ability to do anything anywhere. But we still have a mentality in cyber-physical systems that the mechanic, whoever is working on the system, whether it's in something like WEA where it might be the carrier or the employee of the carrier who's doing some kind of work, or in the automotive systems, the physical mechanic or the dealer have unfettered access to everything because they're trustworthy or implied to be trustworthy. And that may not in fact match the threat modeling.

Presenter: Right. And in the threat modeling, what you want to do is you want to start questioning things and say what if they aren't trustworthy, what happens then. And then you start looking and creating these scenarios. And so, that's where you start looking at some of the use cases. Here's how-- here's what these people have access to. And here's how they use the system. And then you say, "Well how can we abuse that trust that we've put in people?" And that can help you start to think about how you want to segment operational use of the system.

## Polling Question

### Polling Question

Are your organization's security requirements designed to reduce security risk in deployed software or systems?

Answers:

- Yes
- No
- Don't know

\*\*055 And I think we have one more polling question.

Presenter: Okay. And Chris's final polling question is going to be posed now is, "Are your organization's security requirements designed to reduce security risk in deployed software systems?" And while we vote for that one Chris, do you mind if we get into some questions?

Presenter: Sure.

Presenter: From the audience. So, let's see. Ted here wanted to know, "Is SERA integrated with regular requirements development for a system? Or is it done separately and then integrated later?"

Presenter: Well, it's integrated with the security requirements pieces of

the system, which should be part of your general requirements activities in the organization. And so, in general, if you're doing require-- especially if you're looking at generating security requirements, you should be doing some form of risk assessment, whether it's SERA or some other version. There are lots of risk assessments out there. But you should be doing some aspect of risk analysis. And it should be integrated into what you're doing. And that should also be integrated into your overall requirements processes.

Presenter: Okay, a couple questions again about if the materials are available to download. If you just go to the download materials tab at the bottom of your screen, you'll find everything. And the event is being archived. So, that will be available by tomorrow.

Amy wanted to know, "How do you evaluate credibility of each threat? Are insiders with security clearances less likely or more likely to become disgruntled?" Is that a question for Randy Trezak?

Presenter: Yeah, that's actually a good question for our insider threat people. I think what you want to do in terms of when you're doing a risk analysis, I would look at it-- you might look at it from two cases. What a normal person who has regular access, what could they do? But what could some of the super users or administrator type people do because they generally have access to more,

more resources, more information. And they can often do more damage. So, I kind of-- you kind of might want to play a few different ways. But what I think I would do ultimately is defer to what the data from someone like our insider threat team and what they would say where the most likely threat are to occur. I don't have a good answer for that right now.

Presenter: We'll squeeze in one more from Brandon asking, "How do you verify that the risk analysis and probabilities are reflective of reality?"

Presenter: That's a tough one. Right now, what we're doing is using security expertise to do this and having security experts. And you have multiple experts looking at it from different perspectives. And so, it's very subjective right now. And so, how can you say it reflects reality? If you have multiple experts looking at it from different perspectives, that's about as good as we see people doing these days. We don't have a lot of data that we can draw on to say that the actual quantitative probability is X or Y. And so, that's one of the difficulties and one of the areas that I think that's ripe for future research in this area is how do we refine that and improve upon that as we move forward.

Presenter: Just to wrap up the polling question, which was, "Are your organization's security requirements designed to reduce security risk in deployed software systems?" It was sixty-nine percent

yes, nine percent no, twenty-two percent don't know.

Presenter: Okay.

Presenter: Back to you for your summary.

Presenter: Okay. Okay, a couple key points.

## Key Points

### Key Points

Software assurance:

- The level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at anytime during its lifecycle, and that the software functions in the intended manner.

Software security requirements:

- Features (e.g., controls or constraints) that specify how to preserve the confidentiality, integrity, and availability of critical system data

Security Engineering Risk Analysis (SERA) Method:

- A systematic approach for analyzing complex security risks in software-reliant systems and systems of systems across the lifecycle
- Can be integrated with security requirements engineering

\*\*057 Just want to highlight that the whole focus of what we're looking at here is software assurance. And that has two aspects, predictable execution and trustworthiness. We're focusing here on trustworthiness. And the goal is to have some level of confidence that you're addressing your risks and that your software is trustworthy. In this case, risk is a good way of doing that because if you're looking at what your highest

concerns are, that gives you-- and you're mitigating them, that gives you some level of confidence in the software that you're producing.

The second is, with respect to software security requirements, those are features that preserve the confidentiality, integrity, and availability of system data. And we kind of showed you how in the SERA method in our first task, we look at data flows. We look at what's important about the data from a CIA perspective. And then that kind of helps you get the first step in terms of addressing that aspect of software security requirements. And then by applying the SERA method, you look at your risks. You prioritize them. You develop controls for the highest priority risks. And then you bring that back into the security requirements engineering process and integrate it back together.

Presenter: There's another element as well, to go back a bit to the last question that we just got. Chris doesn't really have time to delve deeply into SERA. So, we didn't cover everything. But one of the elements of creating the operational context and the threats is to expand the number of views that are considered, physical views, operational views, data flow views, process views, workflow views. And the point there is that, in many circumstances, security is done in a very siloed way. So, you ask as system administrator what's important, and they'll say configuring the firewall rules. And if

that's not done, they say there's a big risk to the system. Once you get everyone to see all of these viewpoints from who was responsible for the physical security of the system, and the maintenance, and the developments, and so on, you see all these different views. And people get-- everybody gets the same comprehensive view. And then the administrator, to sort of pull on this thread, will say, "Well, that firewall rule really isn't the most important thing to do because now I understand it's in a locked room and no one can get in anyway. And there's no outside connectivity. So, the firewall is not the real security control that's helping make the system secure." And that's a way to avoid group think and silo thinking in trying to make sure that you've got the appropriate security you need in order to accomplish what the business mission is.

Presenter: And just to build on that, when you look at the views, the one thing I wanted to point out, there's a lot of-- when you look at all the models, it looks like a lot of work. But if you're doing a good job of engineering the system, a lot of those should be available anyways. You should have what your workflows are. You should have a general sense of what the use cases are already. So, you're not necessarily generating them for this process. But you're leveraging a lot of other information that's already been generated.

Presenter: Can we squeeze in two more questions? I know we're up against it here, but just good questions. One is from Andreas asking, "This method, like others, appears very similar to failure mode and effects analysis. What are your recommendations to organizations who are familiar with FMEA and want to adopt this method in the security domain?"

Presenter: One of the key differences is you're looking at an active threat. So someone's trying to subvert the system. And a lot of times in failure modes, you're looking at how the failures can occur from a reliability perspective. So, you want to make sure that you incorporate how human actors might engage with the system to try to create risk.

Presenter: The short sort of sound bite that at least I use is attackers don't obey physics.

Presenter: Next one, "How often should the risk analysis be repeated over time as the system changes?"

Presenter: Yeah, well and so what you want to do is if you start it in the requirements as we're talking about, you'll want to take another look when you get to the architecture and look at what's changed because when you get to the architecture, you know more. And then as you go through each of the key activities in the development process, revisit it at each point. And then because you're getting-- as you move through the

lifecycle, you're getting more certainty about things. Some things that are more speculative up front, you have a better idea as you move through. And you can make better judgments.

Presenter: I think you just answered this one too. But, "Do the steps to derive security requirements change when following an Agile methodology?"

Presenter: I don't think so, although I have not applied it with an Agile methodology yet. But I don't think that steps would change.

Presenter: We actually have done some work in this area as well. And if you like, we can add that to the resource list.

Presenter: Okay, great. Chris, out of time, thank you for a wonderful presentation. Mark, thanks for your facilitation. We're going to break here until about 2:35 Eastern time. So, if you are not going to come back for the second part of today's presentation, make sure you go to that survey tab and fill out information as your feedback is always greatly appreciated. And we'll be back at 2:35 Eastern time with secure coding best practices by Bob Schiela. See you then.

## SEI WEBINAR SERIES | Keeping you informed of the latest solutions



**SEI WEBINAR SERIES** | Keeping you informed of the latest solutions

 Software Engineering Institute | Carnegie Mellon University

 Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

1