DevOps Panel Discussion

Table of Contents

DevOps?

**006 Announcer: And welcome back to the SEI virtual event, Cyber COI Alignment, or Challenges and Gaps. This is our final presentation for today. It will be a DevOps panel, moderated by SEI CTO Dr. Kevin Fall. And just a reminder for everyone, if anybody's just joining us, to take a look at that Files tab where you could download the resources from today's event. And also, upon exiting today's event, please fill out our survey, as your feedback is always greatly appreciated.

Now I'd like to introduce Dr. Fall. He's a Deputy Director and Chief Technologer, Technology Officer, of the SEI, where he directs the research and development portfolio of the SEI's technical programs in cybersecurity, software architecture, process improvement, measurement

and estimating, and unique technical support to sponsors.

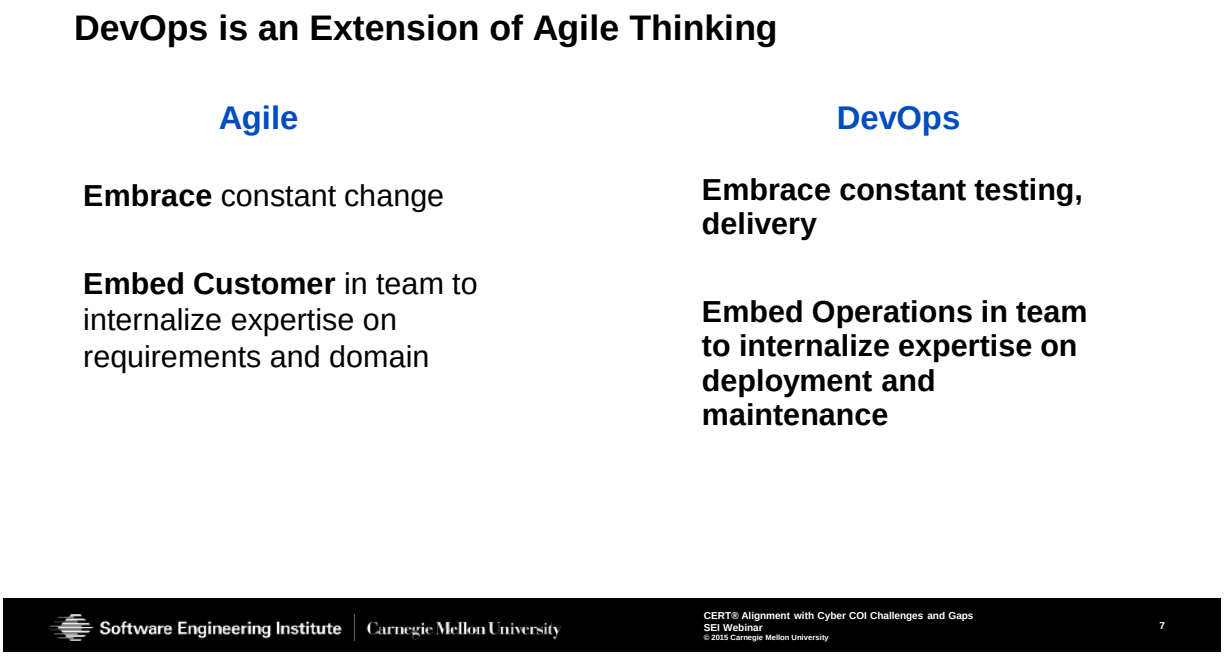
Joining him as panelists will be Joe Yankel, who is a team lead within the Secure Lifecycle Solutions team here at CERT, and also Hasan Yasar, who is the Technical Manager of the Secure Lifecycle Solutions Group here at CERT division within the SEI.

So now I'm going to turn it over to Dr. Fall to get us going.

Presenter: Okay. Thank you, Shane. Good afternoon everyone, from Pittsburgh, Pennsylvania. We're going to have a discussion here about DevOps and its relationship to cybersecurity and the dynamic threat. So thought maybe we'd just first start off by just saying and sort of level setting with what do we define DevOps to be?

Hasan Yasar: Yeah, so the DevOps is actually very industry term known, and the basic definition is the emphasized communication, collaboration, integration between software developers and information technology and operations personnel. So it's a industry term.

DevOps is an Extension of Agile Thinking

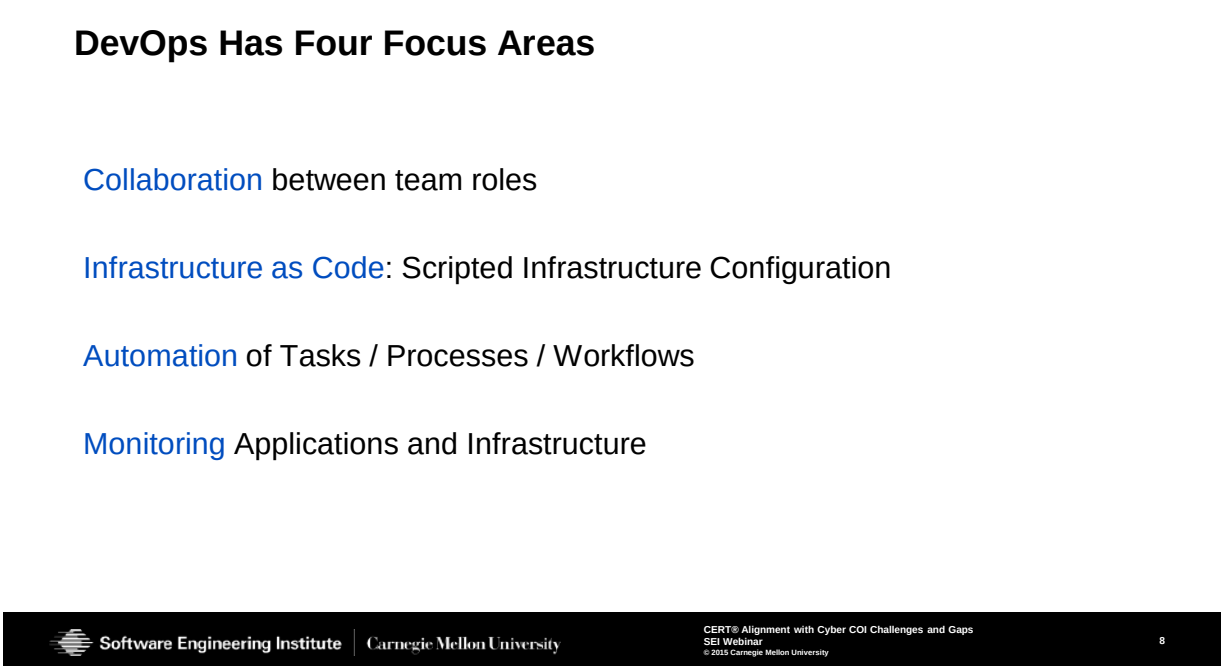


**007 However, the DevOps is a extension of HI methodologies. It requires a lot of knowledge and skills necessary to take the project from inception throughout sustaining and to be continued within a dedicated project team. So it's a practice to enable the team to achieve the level of coordination and understanding necessary to automate infrastructures. So it's a kind of like industry term known, but we would like to really cover up the process and methodologies. How can we get that process and methodologies automation, especially how industry is deploying in a faster and quicker release in a large-scale application? We would like to get the deployment techniques into the agile operation of the cyber threat or cyber maneuvers and make it faster and real, that type

of thing. We would like to get the DevOps in that realm.

So that is the basic principles of DevOps.

DevOps Has Four Focus Areas

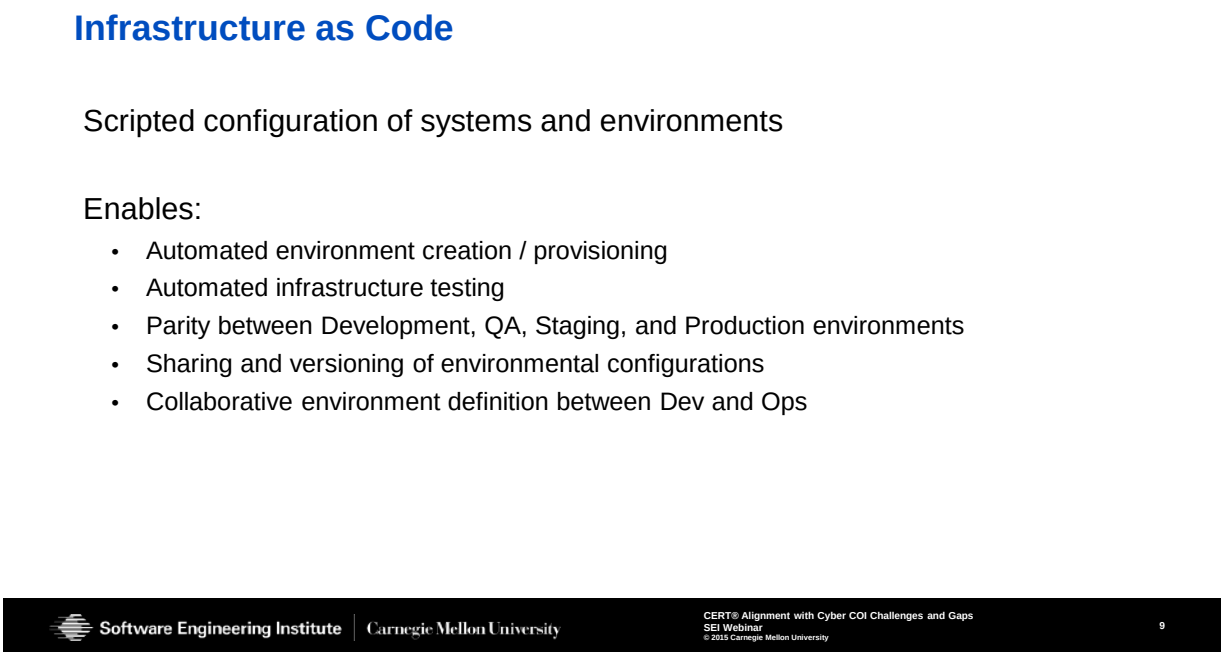


**008 And these are the collaboration, infrastructure as a code, automation and monitoring. So those are four principles of DevOps. And mainly infrastructure as code and automation will be really directly addressing the dynamic cyber threat and . And anything else, Joe, you want to add to?

Joe Yankel: No. We'll continue on with the--get a little more in-depth on infrastructure as code.

Presenter: All right. So this second item's infrastructure as code. So how does that relate to the automation aspect for DevOps?

Infrastructure as Code

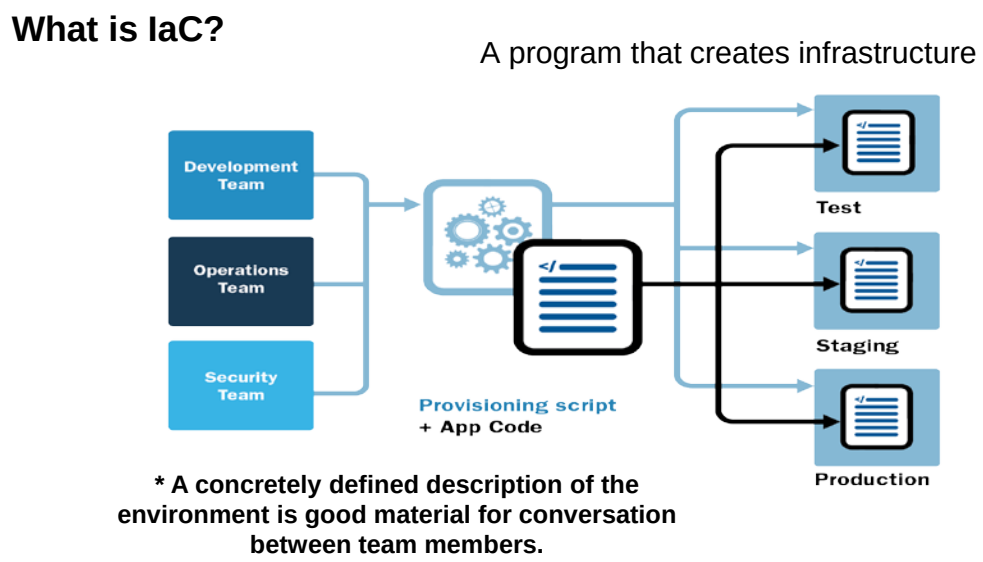


**009 Hasan Yasar: So infrastructure as code definition is kind of scripted configuration of systems and environment. So what basically means, and based on the technology changes and virtualization, that makes enable to have the configuration of environment. So what environment means, can be a virtual host, can be any applications reside on it. Can be a little testing platform. Can be infrastructured. Can be a network platform. That can be all scripted and all configured. So depends on the changes, and instead of really manual configurations.

So scripted can run automated configuration, push any VMs onto, on the fly, or can push any environment on the fly, including network configuration and some of the search configuration, possible network

changes. Then on the fly and in . So it's basically scripting all the infrastructures have a code and put in repositories and change it dynamically. That we can describe that way. So basically the main goals and technical challenges really have some sort of configuration. So if you get configurations treated as a code in a platform, it's easy to, to deploy it, and then easy to monitor, easy to versioning and then change it dynamically. So basically I'm going to jump to the folder.

What is IaC?



**013 The benefits, as they're describing, so in a general known term, it can be they will open the environment operation in security. So in a practice way, that can be multiple section of the different, the configuration of environment. Can be application, as

I said, can be a network configuration, or it can be a platform and dynamically change it. And depends on the needs.

Presenter: Okay. So given the DevOps cadence, and then with the automation and infrastructures code, we're faced with a dynamic sort of evolving cyber threat that changes regularly. How does those type of capabilities allow us to better counter some of the threats that we might face?

Hasan Yasar: That one goes to Joe.

Joe Yankel: Yeah. I could speak to that a little bit. So one of the things we're going for with DevOps is-- and infrastructures code, is we want the ability to quickly deploy our applications, continuously update them, and deliver those.

Code that does *what*, exactly?

Code that does *what*, exactly?

- Creates VMs
  - Vagrant, VirtualBox, VMware
- Provisions VMs or hardware with specific dependencies, configurations, networking, application code, Dynamic reconfiguration of networks, systems and applications
  - Docker container, Shell script, Chef, Puppet, Fabric, Ansible, etc..

© 2015 Carnegie Mellon University

**014 So in the case of maybe a dynamic changing threat, if we can detect those things then we can have the action of--if our infrastructure is checked in as code, we can then dynamically put out a new infrastructure. Immediately. So the idea here is that I make a small change in software. I don't only deploy that one application. I actually deploy my entire environment. My network, those applications, their IP addresses. They all can be pushed dynamically and basically give me a brand-new system. So systems of systems can be built on the fly. So we're going for continuous delivery. Continuous deployment. That means every change I make I can build my entire system. So we've often approached dynamic changes with just building specialized software to reconfigure itself.

Well, in DevOps, every day, we want to possibly push out a brand-new build. Using that same, those techniques, we could, essentially, we've detected a threat. Why not rebuild our environment, the entire thing? So it's kind of a different way of addressing. Something we do every day in software development using agile approaches in DevOps to get multiple builds per day, for instance. We could have a environment deployed at the push of a button. So we're really going for speed here. Currently--

Presenter: And agility.

Joe Yankel: Speed and agility. Currently it's a manual process. We detect a threat, a human actor gets involved. Makes manual configuration changes.

But with DevOps, we've already implemented a lot of tooling, automation, scripting to build this environment up. So even a minor change can give me a new environment. Well, why not trigger a new environment based off of threat detection? I've detected a threat. I've been compromised as well. I'll deploy my environment again. I my host was compromised, that host is gone. I have a new host, learning the same code or an updated version of the code. So that's where we're going with, with DevOps, in the next phases of helping the COI here.

Hasan Yasar: And I looked at a couple other things as well. As you

described, it's really getting a faster automation and deployment, how industry's addressing the application deployment in faster ways, in a large-scale user base, and taking that concept and using a threat factor. So if there is any threat changes, depends on the size, depends on the complexity, maybe we can throw out whole applications through it and--or we can just have the minor configuration. That can be IP configuration changes on a box. That can be some application configuration changes. The other things, if you know some simulated variable before, by changing the variables then we can push the new application settings probably. Or if you know the signature of any, that any threat that be captured, we can do some sort of changes on a mission itself, then we can throw another version of application or infrastructure in a faster way. As you know, that is really important. And one of the topic in cyber will address the needs in a higher speed and as quick a way as possible. To achieve that there's a quicker way and a faster way we have the automated mission behind it, which is the mission has to do that, which is a PCI , which is what's happening in industry, and take that concept, putting the cyber threat environment and change it dynamically.

Presenter: Okay. So the DevOps framework and infrastructure as code has been largely focused on

infrastructure in the sense of VMs, computing platforms, end points. Certainly the cyber threat is applicable to the network infrastructure as well, and we've got at the same time software-defined networks and other kinds of more programmable infrastructure. What's the relationship of that type of DevOps and infrastructure as code part with what could happen with more dynamically programmable network infrastructure?

Hasan Yasar: So the networking configuration, I think about any firewall box, any other switch. So if you have a versioning on that box itself and then we can change the firewall role, rules, on the fly and depends on the threat, then we can deploy new version into it, into box. It can be a maybe stand-alone mission itself, and again, changing the configuration and deploy dynamically. And that configuration is going to be treated as a code, so we have a versioning. So we can go back and if threat is changed again we can go back to previous version or a different version that we planted before and push it dynamically again into the network environment. It can be one of the server, it can be many servers.

So also other things then that's very industry things is use it right now as a document container, so we can treat as an application container, and then throw into another host maybe. Let's say if the host is compromised, then we can throw another host and

throw an application into it and it has configuration in it, so that can be another environment as a part of the OS or a part of the configuration of the network.

Anything else you want, Joe?

Joe Yankel: Ah, I think you're on... That's the track we're thinking, is that if we currently--currently it's not addressing necessarily building a network out, but why not? We can script almost any changes in network. So why not consider those to be vital pieces of our, of that infrastructure should be committed as code? Therefore changes to that could be made on the fly also. It's just one more step in the evolution of creating not just applications but systems.

Presenter: Right. It strikes me to that there's this move toward network function virtualization where you're, you know, eliminating the stand-alone boxes and moving some of that functionality into the infrastructure, which should, it would seem to me, lend itself directly to all these kind of benefits in the same way.

So are there specific, I mean, maybe along those lines, are there specific processes or tools or capabilities that you might recommend or that you're familiar with that you'd like to talk about for people?

Hasan Yasar: .

Joe Yankel: Well, virtualization's bit. It's enabled everyone to begin to quickly try these things. We don't have to rely on having so much bare metal to do the configurations. So I'd say important step. We should be virtualized at some point, and that allows us for quick reconfiguration of machines.

Next is this evaluate your current processes. I don't know. I guess, like, we see, there could be a DevOps scale. How DevOpsy are we? We're just talking about agile methods. We're talking about can we make quick changes from a requirement to having a brand-new application? How long does that take your current, your system, your project teams? How long does it take to implement a change? Well, here on some of my project teams--you asked for a requirement. If I can, if my coders can do it quick, say, an hour, well, an hour and a half I have that build up in production ready to go.

So when it comes to speed, it's the only way. First thing to do is, "How long does it take me to deploy our new system, our new software, and our current infrastructure?" You might want to ask that question. Are you even close? The answer's often no. We're nowhere close. It takes six months to get a change. So we need to look at our tool sets. We want to find out, "Can we build?"

DevOps tools

**017 Will a change in source code, will it result in a brand-new build? This is continuous delivery. We want to get to that. I'd say that's step one. We're talking tools.

Step two is far away for most folks. This is continuous deployment. That means my code change is not just delivered in some fashion as an executable or a web application. But put out there for the public for general use. That's pretty advanced, but that's where we want to get. And if we get there on all our systems we create--and this, this involves re-architecture--we want smaller services. Microservices that can be spun up and deployed. Changing a little, little bit in the way you architect does lend itself well to doing this. Large, large systems, they take a lot of time to get to the

level where I could easily change the entire system based on, say, a threat.

Hasan Yasar: And I think these are the common tools that we can use in industry standard, like the Jenkins and the integration server. Mainly too Docker is a really great tool, and have self-contained environments and deploy into the new infrastructure and new servers. And along the other things I would like to touchpoint that it's actually the, one of the technical challenges of the site and the collaboration and the coordination between the other, the multiple actors. So using the same similar techniques in software developing, like having a closed platform, common platform. Get all the developers, like, in that case, all the actors in the same collaborated platform. They can share what they're finding out and that it's possible to automate that, find their findings, and tie into the application itself. That application is deployed based on the configuration. So all that because they've initiated their, what they're finding out, threats, and then, and correlate together in a common platform and share across other team members. So that can be automated, and how is it happening in industry right now? Let's say one of my developers altogether, if you find any bugs in the system, we are collaborating through our system, tracking system like Gyro or some other things, then we're able to collaborate that. Or we can do some sort of check so we can collaborate that.

So once we collaborate and it's possible to stand up another build and push into the application directly to the production or staging environment. In that case we can literally throw the new application sets and based on threat and into that environment so we can address dynamically new threats, it's going to be, like, the speed of the application and the mission is going to . Depends on the power, depends on the network configuration. Can be a short time frame, it can be a big, but at the end it's going to be much faster than the human, instead of having one person or multiple persons. It's going to be much faster in a large-scale environment that we can see.

Presenter: Okay. Is there a...

Announcer: We have, yeah, an audience question from John asking just--we can go back to the--what were the four elements for DevOps? He got the first three of collaboration, infrastructure of code and automation.

What was the fourth again, Hasan, if we could--

DevOps Has Four Focus Areas

DevOps Has Four Focus Areas

Collaboration between team roles

Infrastructure as Code: Scripted Infrastructure Configuration

Automation of Tasks / Processes / Workflows

Monitoring Applications and Infrastructure

**008 Hasan Yasar: So in the industry known standard there is, the fourth one, is monitoring. So I'm just going to go over again those four principles. Collaboration is really key point. Collaborate all the team roles and team members. The second one, infrastructures, which we described well, and automation's another task, which we're talking about automation of the process and work flow. And the fourth one is monitoring. It's actually a good topic that we can talk and talk. If you monitor environment application itself on the server configuration. So I know there's a lot of ideas as platform or the monitoring platform in cybersecurity things. So if you tie into their application configuration, let's say if we capture anything that we expecting or we can have on alert, that can trigger some

configuration on the application itself, then either we can deploy the new one or we can understand that mission's compromised.

So if we kept infrastructure as a code and separate out the data in other server, if we know that server is compromised we know nothing happened to data, we can capture immediately and through another server, which is a monitoring capability. So it's really a good component having a monitoring throughout applications with, throughout, all the monitoring capabilities as part of the process.

Announcer: Okay. So one other question in Q, just from Joe, asking, "How often is DevOps used in the DoD environment now?" Is that something you guys see a lot or you going in and pushing that with customers now?

Hasan Yasar: It's a perfect question. I was talking with Joe this morning, actually. It's a very new concept for DoD space. It's very, very known in the industry, and how the Netflix or how the Amazon, how the Google are doing great now. We would like to get that concept into the DoD space and addressing agile operations. It's much better. So it's not really as known in DoD yet and it's still there trying to get agile operations , but it would like to get the DevOps tools and methodologies using in a cyber-maneuver techniques and addressing that needs, basically. It's -

Announcer: And you had the slide with all the tools. Is that something they have trouble getting access to those tools? Is there limitations--

Hasan Yasar: It is--

Announcer: --that they need that they can't access?

Hasan Yasar: Actually, it is not, there is no limitation. Some of them is open source. Some of them--even though some government folks, now the DoD and some civilian agents are there using already right now, it's a matter of changing their process, matter of their application, as Joe described. If the application is micro-architectural modular, yes, there is a way to get that tools and use it. It's just a knowledge base. It's not really protecting anything else, I believe.

Announcer: Okay.

Hasan Yasar: It's open source. Some of them is open source. Some of them is configurable. There are other things then that has to be picked. Based on the environments, what the DoD folks has, and then we have to use the proper tools for the environment, so we cannot really get any of the tools and use it. But if anybody using the VM, and most of the tools like the Chef and Puppet is capable to that type of work, like Chef, Puppet, it's possible.

DevOps tools

**017 And if anybody's in Linux environment, it's perfect to do well Chef and Puppet. If anybody using any, like, a Windows environment, visual team foundation serve as a part of that, it's possible too. So like other Docker container, Docker can work in a Linux environment, can work in Linux environment as well. So it really depends on the environment configurations possible to do that.

Announcer: Okay. So that cleaned up Q, so I'm going to turn to Kevin and maybe let him grill you on some of your work, come up with questions, put him on the spot.

Presenter: Okay.

Announcer: And, you know, while we have about five minutes here to

finish up. Any closing thoughts or, you know, let Kevin take it.

Presenter: So let me revisit the question I kind of got to before. So I originally came from a networking sort of point of view or background, and the evolution there had been from fixed function sort of devices and actually in the way back they were programmable, but then they were fixed function with higher performance. And now you're getting this software-defined networks, where things like routers can be changed. They can change flows, and now the next evolution that's been discussed, this network function virtualization, is to take those fixed function boxes like intrusion detection and load balancing and stick them into the rest of the infrastructure as code.

So if you start to now have the data flow elements next to the end processing elements all shared in the same racks and maybe cloud computing infrastructure, what kind of performance issues are we going to have to be worried about? What kind of security issues are we going to have to then worry about down if we have, might we someday get to the point in the DoD setting where we have multiple different kinds of networks that we can actually process together?

Joe Yankel: Oh, it's a good question. We do have to be concerned with the security. We're aware of it. I think what happens is

we skirt it. We don't actually take care of the problems. We actually know how to handle almost every situation. It's just that it isn't always--it's the last thing. It's the last thing checked and implemented. We do have to have better process. Every individual technology we look at provides us a fairly good security layer. Well vetted. Our mistakes happen, our breaches happen, on mistakes. Most of the time.

Presenter: Those errors in configuration and error ins--

Joe Yankel: Yes. By reproducing these environments, continually knowing this is how it's done, we get better. By executing the same environment application and test. In this idea of DevOps. In continually delivering I get a really good chance of guaranteeing what I see every day is what I'm going to see every day in production, not just in tests. I'm going to get the exact same type of security. That's what we're--

Presenter: So practice makes perfect.

Joe Yankel: Practice makes perfect. We need to get better practice. To go back onto one topic, how is the DoD? They're behind. We know we need to begin this in acquisition. We're still in the waterfall model. Let's get a contract. Let's decide everything right now. Two years later we'll deliver. What happens? We miss our timelines. We want to change requirements at the end. It's not the same. Technology's

switched. We need to invoke DevOps on acquisition. We need to right off the bat give our, when we're acquiring new software, have them build it in our environment. Have them continuously give the updates. So my guys who will own the software, own these services, are using them, interacting with them every day from step one.

Announcer: I attended Saturn this year. The keynote was a DoD gentleman. I forget his name, but he mentioned a lot of these tools. So I think they knew they were behind but they are, from what I see in his keynote talk, about how they're putting these tools to good use now and trying to speed up the--

Presenter: So you really want to acquire the update process at the same time you acquire the thing that you're interested in.

Announcer: Right.

Presenter: Right.

Joe Yankel: Honestly, the tooling's the easy part. The hard part is the beginning, the--

Hasan Yasar: Beginning process. Using them.

Presenter: --using them. Using them effectively. It's true. There's a lot of good, really good tools out there. But if you don't have the process behind them, that's going to be a challenge.

Hasan Yasar: Other things I would like to add at that, what you described before, your question, Kevin. And if you kept the network configuration as the code itself, as versioning, so it's easy to--

Presenter: Right.

Hasan Yasar: --the version older or network is set in a platform, so we know what we have, what type of version we have, each network appliances or ideas or services, and we can really monitor it. If there is any security problems with any of the configuration, we can fix it once when we can deploy all of them at the same time, which is automated deployment and how the industry's doing .

Presenter: And you could go further. You could deploy heterogeneous ones if you wanted.

Hasan Yasar: Right.

Presenter: Or you could dynamically perturb them in some way. Or there's a variety of things you can do with all these tools available, and I think it's--

Joe Yankel: I think it's actually a different approach to current methods of changing my network. You know, we currently write custom applications and custom configurations that dynamically change. Well, why not build? Have a random distribution of network changes.

Hasan Yasar: Mm-hm. Like, take down some services. How the Netflix doing, right, with the right now.

Presenter: Right.

Hasan Yasar: So they are taking down some services and try to make sure it's going to work. Depends on the threat. Depends on some failure. So it's a fairly likely threat. If you take the threat as a threat and take down some of the services and monitor it, how did it happen, how can we address that type of needs? How can we change dynamically?

Presenter: So sort of a moving target defense is maybe enabled both at the application side and the network infrastructure side as it moves into the end point infrastructure in a way that we haven't really done before.

Joe Yankel: Certainly.

Presenter: Yeah. That's probably right.

Announcer: Well, gentlemen, we're about to wrap up. Thank you for the conversation today. If you're interested in DevOps, Hasan's team has a number of blog posts. They've done a number of webinars, so just search the SEI website. You'll see a lot of content developed from his team that can get you going on the right path.

So that's going to wrap up our event for today. I really want to thank

everyone for spending the day with us. As I mentioned earlier, the event has been archived. We will send out an e-mail sometime in the next day or two with the location of the recording and the slides. We ask that you share that with colleagues, with your social networks, and spread the message here.

We also want to, we invite you to continue the conversation. There's a number of CERT LinkedIn groups. You can follow CERT or the SEI on LinkedIn and continue the conversations there, ask questions of our various researchers and presenters today. You have their contact information. You can always e-mail info@sei. So we want to continue this conversation from the research that was presented today, and like I said, just carry it on.

So once again, thank you for attending, everyone, and have a great day.

Thank you guys very much. Great job.

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/).

© 2015 Carnegie Mellon University.

Copyright 2015 Carnegie Mellon University

Copyright 2015 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.

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.

This material has been approved for public release and unlimited distribution except as restricted below.

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-0002555

CERT | Software Engineering Institute | Carnegie Mellon University
CERT® Alignment with Cyber COI Challenges and Gaps
SEI Webinar
© 2015 Carnegie Mellon University