1 00:00:00,000 --> 00:00:08,000
, thank you. OK, great. So again, Robert McHale and the company's policy workgroup

2 00:00:08,000 --> 00:00:18,000
co-chair and also a member of Very Sincere and communities, segues focusing on security. And I do have a day job as CTO of Sunstone Secure where we do work

3 00:00:18,000 --> 00:00:28,000
with just four years and three powers and agencies on etc. So that's where I bring that personal experience, as Michaela mentioned. So today, as we've been participating

4 00:00:29,000 --> 00:00:38,000
in the Moscow community, we have seen a lot of presentations on the CSP perspective or the catalog perspective and understanding the controls and how to

5 00:00:38,000 --> 00:00:49,000
mouth those two components. We hadn't seen a lot of presentation on some from an agency perspective. How would we actually use Moscow beyond just getting documentation in a new format? So that's who I'm addressing today. But everybody is, of course, welcome. And in

7 00:00:58,000 --> 00:01:08,000
prepping for this, I promised I'd keep it kind of level one to level three. So if you've never heard of Moscow, we'll start with a little bit of an intro and then we'll kind of ramp

8 00:01:08,000 --> 00:01:18,000
up the technical detail and I promise I can keep on time. We're going to have a little demo and then everything I am showing is, of course,

9 00:01:18,000 --> 00:01:28,000
our opinions. It is not represented by ness that is not endorsed by Federation for any U.S. government agency. So please
we will post all of these materials and any examples we show on an open GitHub repo. So everybody is welcome to download them, use them,

represent them any part of this? I will say that our experience with agencies and working through the federal process is that it is very, today very document focused. So your support and some of these terms and concepts come directly from the veterans training.

You're looking for inconsistencies and in the control narratives, the diagrams and how those match up to the the narratives. You know, the system security plan.

and the three power test cases, you know, there's a lot of work that makes sure everything's in alignment. I will highlight that there are these relationships embedded in the work.

activities for an agency and how they're thinking through the process of digesting all of this documentation. And so I'll highlight those relationships.

in a little bit.

I would say it's a lot of work for an agency, I think, you know, or a lot of us work with spouse, or maybe they are spouse, work with a lot of three powers.

And I think what gets lost in the conversation and there's a lot of work on the agency side to either sponsor a bedroom or even reuse an existing
for and package, right? They have to understand the impact their systems and the interconnections they have to understand, you know, the risks.

associated with the system and the controls and how they're implemented. You know, they have to try to digest a totally new system policy and procedures and make sure that those are covering all the statements after says, you know, whether the three powers doing their job while they're accredited, you know they have to they have to take ownership that the tests were correct.

And of course, they're signing up for a long term responsibility if they're the sponsoring agency, especially that they're going to monitor and review online every month. Continuous monitoring.

So it's a lot of work. And I think, you know, as a community, we owe it to the agencies to try to present a path forward that, you know, reduces their effort and in the end, if we can make this re-use possible with a lot less effort on the agency part, we can really accelerate the timeline.

to getting agencies to sponsor new federal packages or reuse existing federal packages, and that supports innovation. And so those who may not be familiar with federal and the agency work, you know, just to cover briefly, they have to understand all the data and the data flows and the sensitive issue of that data. They have to kind of map that up with the documentation that digital identity and the strength of that identity that we have to work through. And, you know, very
cumbersome word document and excel spreadsheet process to do the analysis and review. And then they have to digest the three panels, assessment results and review. So and then of course, there are multiple layers to any complex system these days. So you're usually leveraging somebody else's into a cloud provider, another app or connecting to other federal company authorized systems. So a lot of work and then it takes a team. So, you know, there's agencies we work with. There's a number of people on each of these calls. There's a lot of work happening in the background. So there's probably, you know, half a dozen to a dozen folks actively working. If you're if they're going through the sponsorship process and then even if they're using NATO, there's still a lot of that review and GRC work that has to be done. And there's not a lot of like this letter. There's not a lot of clear definition of coverage like documentation that we're reviewing and all of the controls really covering all of the system threats. And we'll talk about threats and coverage later. Um, so again, just to summarize, you know, a lot of manual operations kind of ad hoc mapping between, you know, the CCP's interpretation of the controls and the underlying system, you know, the maybe, you know, deviations that need to be discussed there is going to inheritance
from different Asian systems

00:05:44,000 --> 00:05:54,000
and you know how those changing traceable over time. Again, the three powers using some sampling methodology, but even for a specific control or even a control part,

00:05:54,000 --> 00:06:04,000
they're recovering the intent of that control relative to the system. You need a lot of subject matter experts to assess these risks, and then you've got to worry about ongoing

00:06:05,000 --> 00:06:15,000
things like incident response preparations, the recovery programs and then every year happens again, annual requirements for better and then significant changes

00:06:15,000 --> 00:06:24,000
again. Just to summarize some of the feedback we've heard directly from agencies on why they either hesitate to sponsor an NCO

00:06:25,000 --> 00:06:34,000
or around the package or even to reduce the federal package adds burden. And typically, it's around the staffing and the time it takes

00:06:34,000 --> 00:06:44,000
. But as I note, you know, adversaries love status quo. So if we're not innovating, we're lagging behind the adversary. So

00:06:44,000 --> 00:06:54,000
was OK. Oh, and one of my favorite pet peeves is that there's those of you who work through kind of the Conran process, you know, everything kind of to R5 for vulnerabilities

00:06:54,000 --> 00:07:04,000
are detected. So we'll talk about this later. But I think we want I want to think from the agency's perspective of

00:07:04,000 --> 00:07:13,000
what we're really trying to do is understand the threats to the system and, you know, vulnerabilities and controls and how those are
interrelated. So

46
00:07:13,000 --> 00:07:23,000
we are not alone in presenting a vision for accelerating and reducing
the work. And this was from a presentation in February 2021. There's
an explicit

47
00:07:23,000 --> 00:07:33,000
goal to really condense the timeline, reduce the work and really focus
on risk management and less on checking boxes. So I think the PMO

48
00:07:33,000 --> 00:07:43,000
and certainly the folks at Nest have declared clearly that this is a
goal. So together, let's talk about how IOSCO helps roll up our
sleeves and get

49
00:07:43,000 --> 00:07:52,000
the work done faster and with more threat and risk in mind. So
clarity. So you know, we're writing things in code with IOSCO defined
set of schemas

50
00:07:52,000 --> 00:08:02,000
based on very clear models. So we're not, you know, looking at things,
we're trying to make things clear. We're literally writing code that
reads

51
00:08:02,000 --> 00:08:12,000
data and producing results. We go from kind of manual checklists and
excel templates to peers. If

52
00:08:12,000 --> 00:08:22,000
you're familiar with. We'll talk a little bit more about kit and get
ops. Everything is a automatable process in a repository that has
automated workflows

53
00:08:21,000 --> 00:08:31,000
, right? It is a team sport and everybody needs a different view of
the problem. Some only need all the details and all the
interconnections

54
00:08:32,000 --> 00:08:41,000
so we'll just need a summary of you and want that word document or the one a PowerPoint presentation. ASCO allows code to slice off the bench that it needs and present only the information at the right level of detail for that audience. And consistency is probably the most those of us who actually work with IOSCO and the schema. Everything is identified. Everything is linked together. Everything is traceable across the different models.

So we try to remove all of the inconsistencies and all the ambiguity and making sure that everything is always lined up as things change.

I think one thing that is underappreciated, perhaps, is that the current process is very linear, and IOSCO by itself doesn't necessarily change that. But it enables a more parallel process where you can now have multiple teams working on a shared set of artifacts in a great repository that is version controlled, arm back controls and auditable. And now you can have parallel branches and parallel workflows and automated, you know, GitHub actions if you're familiar with. And they're very usable, so you can very quickly clone across groups and as individuals come on and come off the project. And so you get a lot of parallel processing that all converges where it needs to. And I'll have another graphic that tries to convey a bit more of that.
later. Budget always important. So, you know, directly related to staff time. So everybody's, you know, twenty five folks on a on a call to review

. And you know, they may be subject matter experts in a particular slice of it. That's a lot of resources for each of those meetings. Whereas if we can make this more asynchronous and have subject matter experts focus on distinct parts and capture that in code or in rules which we'll talk about shortly. Then you really can

reduce that timeline in the budget. And I think a challenge as part of an agency discussion last year about the biggest challenge is staffing

. So those are leaving. Folks are not trained. So this is really what keeps a lot of agency and mission leaders up at night. So

if you can start to embrace O'Scanlon automation, you're now force reinforcing new skills and a new culture that really excites

. I hesitate to say the millennials, but the younger workforce and you know, that's what they're exposed to in their education, and that's what they're looking for in their career path. And so you're building this skills that will help you recruit that talent, retain that tone and then the asterisk courses you really do need leadership

, sponsorship and engagement for this to work for for obvious reasons. So our skill helps again with the consistency
and then the verification. So we have the ability to write validations and we won't go into those because a lot of folks who covered the validation rules and how to make sure that all of the parts of school are in the dark and the data presented is open source tools and a lot of commercial vendors working on solutions. So we'll cover a bit beyond that. But we want to kind of understand how we can map a security capability, encode in Moscow to a control implementation. So I'm sure that in the demo we want, we won't show the profiles and variables, but it's important concept. Others have talked about it that these are documents that can be tailored to an agency's risk model and risk appetite and requirements. And then we want to have that full and mapping from a component control capability threat. And then this really enables reinforces that risk management mentality. And as I'll show you, it also helps you define coverage metrics at all, at all levels and this is going to require a bit of kind of a dual perspective. So on one side of our scale is what I'll call abstract classes. So you've got catalogues, you've got component definitions, right? And those of us who might come from above to join in programming will understand this notion of an abstract class, maybe an inheritance that you
interface if you come from a Java background. And then on the other side, you've got objects, object instances, implements and methods. So

you've got this notion of a real thing that can be compiled or deployed, right? And then in a writing system, you've got an inventory and you've got the actual stuff that you're trying to use and assess. So you want to understand what is my risk, not only at the abstract level. So have

I define all of the controls that I need to define for a particular baseline or profile? And how do those all map together, but also about the implementation

level? So if I build the system according to that schema to that abstract class and implement those things that I need to implement, have I covered all the risks? And then the running system, of course, which I think a lot of us with its security background we focus on with the day to day where, you know, we're scanning things where

were enforcing controls at a technical level and configuration level. So that's kind of the thing that you work with day to day. That's the runtime

and coverage at that level is a little bit easier

. So let's say how you go. So risk assessment. So the benefits of ASCO. Right. So the validations, those kind of schema
channels a check, hey, is my ask out complete and have all the fields in here? And is it all the links? That's going to eliminate your SSP inconsistencies.

So that car's off a huge amount of work for the agency component and controls code. You know, hopefully my strong opinion is that we can deprecate the diagrams and the narrow text. They can be useful as humans consume the results of the assessments and they can certainly provide contextual information for the, you know, giving you situational awareness. But hopefully we can in the not too distant future, stop relying on them as the definition of completeness or coverage rules, which I'll I'll talk to. Not yet an official ASCO schema. I know there's a lot of work going on. I think I put the GitHub issue later on really more about the tests of the control. The control implementation, less about the consistency and the validation. So actually, how are we going to test this in code to make sure that the control and the components implementing these requirements actually match the intent? How can it demonstrate that and all of this to bring it back to that risk management is about, you know, she deposits if you're familiar with the major attack model, but there are other models that kind of rely on threat for procedures and an indicator.
of threats. So all of that helps us model these risks and the relationships between them. So how do we do this? So how are we going to actually implement this as an agency, first and foremost is again, that culture and mentality shift? You know, we should expect Moscow. We should embrace Moscow, we should use it. We shouldn't see it as kind of an appendix that, you know, maybe one day we'll use. We really need to find the champion that can say, this is where we're going. We understood the risks, the benefits, the costs and let's do that. And to support that, certainly since you have some of the sincere participants, our Communities Policy Work Group all providing open source tools, help and support community support and a lot of the folks on this call providing open source tools, community support. And obviously there are vendor commercial solutions to but things in Seattle and communities oriented. You know, everything I'm talking about is open source today. That reskilling so changing who is involved in the skill sets, so you may still have the authorizing official. Hopefully that person is the champion and the sponsor for this change. But instead of the subject matter experts being, you know, narrative oriented, now you're looking at you
know, code pipelines and code artifacts, right? And so you're going to have all the ASCO for your SSP, your SOP, your online validations, your controls

and in some code repo and just call it get on the other side, you're going to have your subject matter experts writing rules, and I'll show you some examples of that. And then there's policy rules as configuration checking rules. There's threat mapping rules becoming artifacts that allow this automated Oscar pipeline to work.

So again, hammering on the documentation issue, using algorithms and so, you know, drawing from commercial world experience

, this is already happening. So a lot of commercial implementations of compliance and security are no longer relying on documentation checklists or documented narratives on how compliance works. They're using algorithms to define, you know, how how things are interconnected, and I'll show a picture of that shortly. They're looking at the strength of those connections that are tagging things with data and and then using algorithms to measure.

. And you calculate, are we covering all the controls? What is the strength and intensity? What are the rules associated that need to pass and how specifically

can you test that? And then again, embrace this notion of automating everything. Don't see automation as a nice to have. If this is going to work.
you really need to automate as much as possible. So think of it again as compliance is code, you know, get ops and the ability to make a change in a repo that's very controlled with usable, auditable. And when you commit, that goes through a pipeline code process and makes changes available to others that need to participate or use those artifacts even down to the final stop. This is a future view where you can imagine having a final authority to operate PR that the authorizing official approves, and that can trigger downstream workflows that the system itself can use as a signal for turning on different features. If you're familiar with the notion of like feature flags. You can imagine having an NCO trigger features like this is now. These systems and services are available with the requirements all satisfied. And then again, I mentioned earlier this notion of parallel workflows those who work in software. I know the git flow process is going to be very familiar. So you instead of everybody kind of being involved at every point, you know, looking at the components, evaluating the security capabilities, how do we tested Jimmy pull on the right controls? Are the controls mapped correctly? So having everybody on those calls and on those email threads really can break this out into I'm the subject
, not in my part. I'm going to participate where I need to modify that and get merge it into the feature. Branches view well and then let my

my team take it from there. And of course, I can monitor it. I get I get notifications in the repo. I can always review the chatter in the comments. Just as all of you are probably doing

in the the Nest GitHub for asking itself. So it all becomes very codified. It all becomes pipeline and it all at the very end

that to bring in just the right people at just the right time. So a big view of a component of how I view this

problem is derived from this diet 811, which is a great publication. If you haven't read it, you should, but it really talks about a

desired state and actual state view of the world. So we're going to use, you know, in this day are they're not even talking about our scale, they're just saying we're going to document

what our desired state is, what security capabilities we need to meet these controls. And then we're going to look at the actual state and do tests to make sure that the actual

state matches. I even started one new concept is what I call declared state. As we move into a more code

systems thinking view of the world, we're going to split out. Here's my desired state I may have defined for a control catalog
, a profile, a baseline. I'm now going to translate or complement that
with declared rules and those rules from the subject matter experts
are going
to support the implement in some cases, even determine the
implementation of those controls. And then, of course, all of that
connects to the actual state
And so the crux for looking at the world in this way. You have to
have policy rules. So I'll show you a demo of rigor

, which is open source implementation of a rules engine. Regan's
language open policy is the enforcement open policy agent

is the enforcement engine. You need to have a model for risks and
threats. So threats, minor attack minded defense

is the instance that I will rely on. There are others, but you really
do have to have a working definition of the threats to your system and
then you have to have
test. And what I call query rules so that you can map your intent to
that can run against the

actual inventory of things. And as I mentioned, my my view of the
future is that, you know, ATO is code that actually can be part and
and trigger things downstream.

so what is policy is code. So for those of you who haven't seen this
and Kubernetes or other systems, it's essentially declarative rules
that state the intent of a particular. It could be, you know, authorizations are about what permissions I have, and it has a very lightweight enforcement engine that can evaluate all of the different rules logically, potentially Boolean operations and make a determination.

However, one thing that folks don't know about this policy is code. Well, I'll get to that next. One thing that isn't obvious from this policy is code engines is they can actually generate code. So I'll show you a demonstration of that later. So there has been threat guidance, just an aside quickly from the PMO on how to look at controls for an agency lens, mapping that to a risk model.

So you want to look at what is the protection? They have a scoring rubric to how you know the control protection against these risks and of course, a coverage metric about how, how well it's implemented and that allows you to prioritize the controls you're going to assess.

So that it matches your your risk profile. So now we have our desired state. We have a plan that talks about in Moscow our actual task for that zone state and we can map threats through security. KTLA Wildflower.
and for those mitigations. Can you please mute yourself? I'm sorry. Wildflower CEO is looking to set up a call for Friday 

, and then you might want to have folks mute. It's just like

, OK, thank you. So I do highlight open policy agent and that's what we use. But there are other agents. Sorry,

policy enforcement engines. Another way to implement this commercially we use this methodology is we do use a graph database and run graph queries

. And so that can actually make decisions based on the relationships and the properties of each entity in a in a graph. And then if you've if

you're really bored, you know, we've done some modeling around this and the policy work group, but you can use things like formal verification and S.A.

and solvers. And so you can really get to formal logic definitions of the rules and the underlying system. But

for today, we'll use the lightweight approach with open policy. Um,

so for under understanding how the threats are being addressed, you do need to have a query on the actual system, right? So there

are a few different approaches to this. We've used most of them and there are others that are out there in the research and open source, so you can use the
natural language in the narratives and the controlled definitions and the supplemental guidance and the policies and procedures in the component documentation.

And then you can use semantic matching or NLP entity extraction. And you know, if you will, keyword matching to try to automate that mapping so that you can say,

sure, you know, the threat language and my implementation language, do they match up so you can build statistical models and you can measure in

the properties of the threat in terms of, you know, where they operate, what entities, you know, what code they're touching, things like that.

I mentioned graph. So you can model the system as a graph. You can model the attacks as pathways through the graph, and that gives you a rich set of algorithms and queries.

You can cluster things you can look at, you know, cycles. There's a lot of rich mathematics around modeling. As a graph, you can have rules. So that's kind of where

we're going to be talking. Um, you know what can be rules of thumb that can be based on the score research that can be based on experience? And I think, you know, the emerging

future approach is to use A.I. with a combination of all of those things. And help, of course, has some machine learning underneath it. But you can use
AI to predict these relationships to model these statistics and provide a predictive model for where things are trending, even generating graph relationships to investigate. Or, as I said, clustering things or finding new powers, setting weights, tagging things, we'll talk about tagging. So I think it's really in the next five years going to be an important component. It has its own considerations around risks and bias and training and all these things. But I think it will be important. And then again, back to that get ops model, you know, we really need to embrace this notion that we're not approving by just consent versus documentation or consensus review or lack of any objection. We're really, you know, enforcing the process through some artifact and some, you know, PR a pull request. If you're familiar with get GitHub, you're you're noting the approval and all the criteria for that approval. Right? And then CSP is in three parts play an important part in this. You know, they'll have to maintain IOSCO for the components and the control implementation details the tests and you want to be able to have kind of a dry run. This is one thing that you can do in a system like Kubernetes. You have policies, code and checking configurations. You can set out all your rules in
something like open policy agent and all your configurations in the declared state.

And then you can do a dry run and say, you know, does everything satisfy the rules before you deploy it? And then, of course, you can have your runtime inventory test and there you need to detect drift. So if I say I'm declaring a particular configuration and deploy that, I need to have some mechanism to say, has it been tampered with by you?

Know, internal or external actors? And all of this allows for coverage metrics. And I want those coverage metrics that at all levels, I want that at the catalog control level. How we covered everything at the risk model, which will show shortly are the risks defined or they cover it or they implement it.

And you know, from a code perspective, how much code am I introducing each step and are the approvals? Are the test matching up?

? So let's talk about this rules engine. So again, we've talked about having desired state. We're going to have rules about that use and reason over the catalogs that controls the profiles. We're going to have what I call declared state the, you know, the component configurations and the security capability configurations. We're going to have some understanding of threats and all of this so that we can reinforce those control implementations with threat data. We're going to have
code tests, rules that test things that's going to be directly encoded and things like the SAP. And then that's going to generate from

the inventory of the song. And let's say that helps reinforce this kind of lifecycle. You know,

continuous updates. So we're going to have controls updated by catalogs and agencies and the regulatory folks. We're going to have the component that's

needing to be updated. We're going to have this kind of threat based model of the system. And then that's going to inform our testing and

as I mentioned, there is some AI work being done in the open source community. So you should definitely check out these projects. I think that's where things are going. So

decision gates, for those people to talk about that, we're working on some some research around kind of building a time machine of deaths for the entire system

. And then, you know, remediation, there's a lot of commercial and open source work about automating the remediation and generating actual code changes based on findings and programs. So as those who have done better, have you know that you need to

do online? You know that you need to report where things are on a monthly basis, but you're really kind of trying to move towards a continuous dev ops model. So you really want to
make sure that all of these things are being updated all of the time and the changes are propagating automatically through the pipelines as long as they're guided by those PR approvals.

And so I think somewhere I may have lost my mind reminders to do the demo. So I think I'll jump into the demo before we talk about the more graphic stuff. So to do that, I'm going to jump over to my Ricoh playground. So as I mentioned, Riga is a language for declaring rules and then validating those rules very effectively and very efficiently. But before you can do any validation, you have to have some data. So this is a very simplified version of inputs you might get from a catalog. And so you're going to define your controls. They're going to have statements, you have properties, those who have worked with Moscow now at the Level two three, and this will look familiar for those who have lost yet. It's just data.

And so you're looking at what controls as the system needs to employ on top of kind of the controlled data, the system will have some some capability, some security capabilities to protect, detect or respond when that kind of Gov. Kamado. So here we've got some inputs and shares the CCP's capabilities for a particular component.
and allows us to reason over different types of protection that might provide. And you know what it implements

And then, you know, for noting that there is discussion around creating a rules or test schema in Moscow. It's not official yet.

So some of this is his prototype. But I'm fairly confident that the folks on this call participating in that we will get to some implementation

of rules in the not too distant future. And so again, as a CSP, you're typically providing components. Your component

abstract class will have to be implemented and we'll have to provide evidence of these security capabilities. So now we get to the point we're actually

playing with the the policy. So in the Riga language, you know, the syntax. If you've ever worked with data log, it's very similar to data log

, but these are essentially, you know, tools that the the engine will interpret and tell you if everything in these rules sets are true,

And so this it's not it's not like Java or C code was executing this and doing things with, you know, branching logic

is basically saying is every single thing I'm certain here true. And if it is, then the output of that should reflect that. And as I mentioned,

today, and so like a Kubernetes system, open policy agent is used
really to make very binary decisions. And if you're smart communities, you can use it for

00:34:44,000 --> 00:34:50,000
admission control and say, should I put things new things into the system or not?

00:34:50,000 --> 00:35:00,000
hear a side effect of where you go and open policy agent is, you can actually generate code. So here we've we've mentioned there were basing

00:35:00,000 --> 00:35:10,000
things on on threats. So we've got threats identified in our capabilities, you know, the threats that we're going to protect against

00:35:09,000 --> 00:35:19,000
and we need to find some mitigation execution of unauthorized container. And now we're going to evaluate and. And so it's showing you that it's

00:35:20,000 --> 00:35:29,000
, you know, everything in green is true, but as a side effect, it's actually generated in the output is here's a score based on weightings that we provide right

00:35:29,000 --> 00:35:39,000
? And I'll highlight, you know, you might have in the Gupkar methodology, you know, a heat map value based on that miter attack or other threat model and

00:35:38,000 --> 00:35:49,000
how the the controls intersect with those kind of those threats. And you can see that it's loaded up different weights

00:35:49,000 --> 00:35:59,000
. And so here you can see that in addition to saying that, everything that I've declared in my rules is true, that I am interested

00:35:59,000 --> 00:36:09,000
as a subject matter expert in this particular threat. It's also calculating the score, you know, weighted or otherwise of how this particular example is covering

00:36:09,000 --> 00:36:19,000
and protecting. So you get that protection about your score and you get that coverage score. So beyond just saying now have a how's my

00:36:19,000 --> 00:36:28,000
component definition, my abstract class, my security capability from the CSP, again, in my abstract definition of what I plan to provide

00:36:28,000 --> 00:36:38,000
, is that really doing what I needed to do? You're going to you're going to subject matter expert is going to define some rules, right? So we're going to say that, you know, I'm interested in containers

00:36:37,000 --> 00:36:47,000
. I needed to have a signature capability and to be able to sign digitally sign any container that's going to be allowed into the system and

00:36:48,000 --> 00:36:57,000
in doing that, I'm going to want to evaluate that with some methodology doesn't mean this is using a particular query syntax, but it could be any any

00:36:58,000 --> 00:37:07,000
kind of evaluation syntax you're interested in. I want to find containers with signatures that are undefined, and that means that it would fail, right? So if

00:37:07,000 --> 00:37:17,000
I'm a lot of the concern is that unauthorized containers are going to be allowed into my system, my security capabilities to prevent that, my test so that

00:37:17,000 --> 00:37:27,000
at the inventory level. Am I going to be able to detect that container without a signature? And again, as a side
effect, you'll see here that it will actually generate specific rule implementations. So we haven't gotten yet to a running system.

But we're saying that in my analysis and in my south planning. And now I'm tying this down to a particular implementation. I'm going to be looking for containers with signature value undefined. And again, it's covering, you know, it's mapped to all the threats and I'm interested in. And then what are we going to do with those rule implementations? We're going to tie those two control implementations so that I think the fundamental shift in mentality here is you're letting the rules engine match things up and define if the implementation is correct, either be a coverage or threat scoring or both. And so when we evaluate the control implementation scoring, I'm saying, you know, I've got to have a particular protection threshold.

old. I've got to have a particular coverage threshold, you know, I'm interested and I want to make sure that those rules are associated with it. What's getting generated is the ask out for, you know, pseudo issue of scale in this demo for control implementation for component definition. So here it's tied it all together. It's, you know, generated the implement requirements based on the control. I'd use reference from the catalog. It's then connected. Those
to the conditions and rules for gathering evidence for that. And again, it's, you know, adhering to the idea semantics and linkages in Moscow. And then in terms of testing all this and, you know, collaborating with both the CSP and the three power unit and generate an assessment plan. And so that's really just binding those rules that are kind of abstract to actual inventory tasks and so on. Doing that now you get an assessment plan that is connected to the SSP and you can see, you know, very specific set of tasks. You have your control implementations. And so this would be severe. So then once you have a running system, you can use those queries connected. You know, they're fully traceable and connected to all the controls, controls, limitations when you gather that evidence. You know, now you've you have confidence that the system is implemented correctly. So I'm going checking time now, getting closer have a bit more to cover, but I think we'll have to move a little quickly. And so I talked about all those relationships earlier. You know, I think in graphs, I think there was a Microsoft quote, you know, the attacker's thinking, gosh and defenders thinking lists, and that's bad. So, you know, we try to think and draft. So if you
think about what isn't necessary, it's kind of a view of the system, but it's a graph of the controls and the components and how they're implemented. Work together, how to run time

inventory level. You can, you know, encode threats and and relationships, score those relationships before you can then run queries

over those draft models and you can find attack paths so that you can inform your rules doing and then you find where things are clustered together. So you

can inform your catalogs and your control implementations. And so this is kind of here's an example of a real system commercially, but I

I do reference the open source free version and the summary. So this is just a graph view of a system

where, you know, you notice where we're encoding and enriching these relationships with specific threat identifiers and metrics. So

that you really then can start to reason over attack pathways and security capabilities, both as an agency, as a three power. And certainly

as a CSP. If you're if you're designing a system, I mentioned that, you know, entity extraction, semantic search as ML, it's being used today. I think a number of

folks on the call had presented some information about that or open source. You know, there are
different models being used all Open-Source that are helpful.

Again, if you're familiar with systems like Neo4j, which again is open source and commercial, but Neo4j comes with a lot of rich graph ml capabilities.

so you can construct a lot of these models from those libraries. You don't have to invent these yourselves, and then you can start to reason about, you know, is my speeds covering everything I need to cover is my three power covering everything within a particular control test or in their sampling methodology. And then what is my overall threat exposure which tying it all back to? I think what the PMO has stated. We want to get some more that risk management mentality. And I think that's, you know, we're graph south a lot and the amount we're constantly learning or tying down all these threat models to specific queries. And that's information we run in in both federal environments.

and commercial environments. We run all of our scenario planning or incident response for disaster recovery for business continuity. Through that graph of you

, I showed previously and now are reasoning about the system, not a very linear way from the document. The checklists do you have those as artifacts

, but we're generating those from graph queries and graph reasoning. So they're releasing in the context of the system and
all the different attack failure pathways. So to sum up, I just want to leave a few some time for questions and

you know, we look at the world from an agency perspective as ASCO enabling compliance as code generating artifacts that are usable, that are verifiable as they come in, that are testable. We want to enable that continuous DevSecOps continuous data model so that you can detect drift those alerts and then really report on precise coverage metrics. And then ideally, we'd like to see that kind of PR issue, right? So we have approval gates encode in pipelines and so that we start to think of the system as developers do that there's no, you know, there's no sacred cows, there's no cattle in and what you're building that it's all reconstructed from code and rules and allows us to definitively show compliance. And at any point in time,

no matter what change, and do it very quickly and effectively with minimal human rework. And so I'll just cover this very quickly.

. You know, we need we need more support for automation. So those of us in science and related vendors who are contributing contributing to open source are trying to help. I think we need I think one of the presenters
earlier had mentioned their implementation of an API. I think that's the right approach. We need more API driven interaction

and eventually we'll need ML models that are, you know, well, synchronized, open, transparent. We'll need, you know, artifacts contributed

catalogs, components. And so, you know, folks like defense unicorns have helped create a lot of great scale examples and catalogs for things like Iron Bank and such

We need more open source definitions, and we need the vendors and ISPs to step up and and see if we're helping that we're going to be running through all seen south projects

that support communities and helping them define off scale for all their components. We need GRC, folks. If you're on the call, please, please

help us bonanza word and excel. Copy paste your tools need to support off scale both ingress and egress. Three powers

, please. We know you have accreditation requirements, but we want to help you use our scale. And so the system is self documenting and self testing

, and that you, as subject matter experts, help inform that role building. And then again, CSP is to support this effort at every stop. Agency leadership

, and they need to embrace it. And I think we've talked about some of the benefits to their hiring and retention. If they do, yeah, so there we are. We
we will put everything this presentation, all the example Rigo code, the graph queries will eventually be we had we had our KubeCon conference last week, so I got a bit behind in getting everything on the record. But it will be there the repo for anyone who's interested in not the open source graph engine that we use is called Starbase that anyone who has with does take a look. And then again, we're happy to support the community. Please, you know, reach out, schedule a call with us. We're happy to spend time with agencies. Three powers even see ISPs for sure. Let us know how we can help you on your Oscar journey and apply some of these code templates and examples in your specific use. So with that, I hopefully have enough time for questions, but I'll go ahead and stop there. Thank you very much, Robert, so we can stop the recording right now and open the floor.