Welcome to today's EDUCAUSE webinar STINGAR. Using and sharing in higher education. Had is Brian Kelly, director of the Cybersecurity Program in EDUCAUSE. And I'll be your moderator for today's event. You're probably familiar with the interface for our webinar, but here are a few reminders. We hope you'll join us in making this session interactive. Use the chat window on the left side of the screen to submit questions, share resources, and comments. If you're tweeting, please use the #EDUCAUSEWebinar. If you have any audio issues, click on the link in the lower left-hand corner of the screen. And at any time, you can direct a private message to technical help by clicking the top right corner of the chat window, a drop-down menu will appear where you can select start with and then host. The session's recording and slides will be archived later today on the event site. And now let's turn to today's presentation. Duke University has created a system for collecting information about a tax on education networks using honey pots and sharing that information in near real-time with all participants. This data can easily be most -- through timely -- feeds and easily consumed by next generation firewalls to protect the campus network. Using container technology, the system can be deployed in a matter of minutes with minimal ongoing maintenance. We're delighted to be joined by Jesse Bowling and Joel Faison. Jesse Bowling is a security professional with over ten years in security -- and another eight years in system administration and desktop support. He's worked his entire career in higher education, advancing a mission in a variety of schools including public, private, large, and medium schools -- medium-sized schools. With a couple of stents in management over his career, he always finds his way back to direct technology work. Jesse is currently the security architect in CCERT program manager at Duke University which combines his love the creating new solutions and a thrill of incident response. Joel Faison has served in higher education information technology for nearly 25 years holding management positions for over 20 years. During those years, he's discovered how rewarding it is to be part of the intellectual transformations that occur on college campuses. Joel has developed a desire -- process efficiencies through the use of technology, a passion that motivated him throughout his 11-year tenure as chief information officer at William Pierce University in Raleigh, North Carolina. In 2010, that passion led him to return back to his alma mater in Durham, North Carolina. Earlier this year he was named the director of infrastructure and information security. Thank you all for joining us here today, and with that, let's begin. Over to you, Jesse.

Hi, everyone. I appreciate you joining today and before we get started, I do have to go through one little bit here, which is they say that some of the work that we're going to talk about here today has been partially funded by the National Science Foundation and my opinions are
not theirs and vice-versa. With that little bit out of the way, let's jump into it. I wanted to talk to you today about what the STINGAR project is. First of all, it's a clever acronym, shared threat intelligence and network gate keeping with automated response. But the purpose of STINGAR is to help higher education institutions do three things, take -- collect information about attacks on their networks, take action on those attacks, in other words to help try to block attackers before they can do damage, and to share that information as quickly as possible with other higher education institutions so that they can be protected by what we're all seeing together. So there's really three parts of this, the collect, the action, and the share. And I'm going to talk a little bit about each of those. For the collection piece, there's actually a variety of things that could be supported in the collection piece, we're focused on the use of honey pots and have spent a great deal of time developing honey pot solution. It's easy to deploy and maintain. We found that for schools that were interested in taking automated response, in other words, doing some automatic blocking based on threat intelligence they're receiving, honey pots were very easy concept to -- to sell to leadership and to other concerned parties at the institution. I think that this idea of a service that is spun up and not advertised in any way, that's made to appear intentionally vulnerable, that, in fact, is just collecting data about who's connecting to it and what they're attempting to do, I think this concept is really easy to communicate to people who might be a little bit concerned about doing automatic blocking or taking some sort of automated response based on threat intelligence data. Actually quite a bit of the software development work we've done has been to take the honey pot system that we were using at duke originally and make it as easy and dependable to deploy as possible and really easy to tear away and set back up and help enable all of these other items. For taking action, there's a piece that the system will enable folks to do. So the biggest selling part we think is to use it to build automated -- automated themes that can be used for blocking offending IP addresses. A good deal of our work has been to make sharing that data into a way that everyone can pull back a feed of all scanners that have been identified by the system and ingesting it into their firewalls with a, basically, zero false positive rate. I think the reasons for that are fairly clear. The honey potting systems, you know, by default people aren't typically accidentally connecting to them. And additionally, all of the services that these individual honey pots provide are configured in such a way that so false positives from things like spoofing are minimized. We don't present any EDP based services for instance. But even if schools are not ready to use these real-time feeds for blocking purposes, they can still be very useful, and they can get value from participating in this program by using these -- this information to do what we term look-back searches. So, in other words, you can get a list of all of the IP addresses seen talking to honey pots in the last 24 hours. And then look back over a period of time and see if you have any successful authentications from any of these IP addresses. Any successful authentications might be an indication of something happening that might indicate a compromised user account, for instance. This information is useful when doing forensic work. So if you have a compromised machine and you've extracted a list of IP addresses that were seen connecting to, you can run these through the STINGAR system and get feedback quickly about whether or not any of those IP addresses were identified by the STINGAR system as performing malicious actions. Which can help focus your forensic investigation. And from our perspective, the sharing perspective is really the most powerful piece of this model, and we'll talk about that in a little bit as well. But the sharing piece is essentially based on the collective
intelligence framework or CIF version 3. We're using that as the back end of our repository. And participants in the STINGAR project can spin up a new type of container that's populated with an API key and host information and they will automatically share the data that they're seeing into the central repository. The reason we chose the collective intelligence framework is because it has a long history in higher education and there's a number of institutions that are already using CIF to manage their threat intelligence, including -- which maintains a very large CIF instance.

We felt it good to stick with what people, generally using and it would enable them to output the data collected in these feeds in a variety of fashions. We also have another container that's focused rather than submitting the data back to the central repository, I'm pulling that data back and renting it in a useable format. So today we have a container called CHN Intel feeds that will talk to the CIF container with a CIF API key or talk to our CIF instance with an API key and gather up all the instances seen over the last three days and republish those over HTTP in a one IP address line format. It turns out this makes it trivial to ingest this feed into something like a networks firewall in their EDL list or in a Sysco power device, both which will accept a URL and pull those feeds back in and do the right thing with them. We're also always open to developing additional solutions or providing flexibility so that people can do more with this data. And we're more than happy to take future requests and so forth on our GitHub link. There's also options for raw honey pot sharing. The data that is typically shared back with -- shared back to the common CIF instance is typically it's just summary data. So you're looking at what time did this happen? What type of honey pot was touched? What was the IP address that did the attacking? And what's the anonymous partner of the person who submitted this? We do try to preserve the privacy of everyone involved and submit all partners when they come into the network are given a -- a partner ID that we can trace back to the actual institution that's in use. So if there is a need for, you know, one partner to talk to another partner about a particular collection, we're happy to broker that connection and make sure both sides are interested in speaking and then connecting them. If there are partners who wanted to do more data sharing than just that summary information, we also have a raw sharing mode wherein all the data that comes out of the honey pots could be shared directly with another partner. This is the HP feeds module that we use for communication between the honey pots and the server and is filled with more information than just the IP addresses involved in a particular attack. You can get all the -- all the username and passwords that were used in an attempt to break into the service and some of the higher interaction honey pots such as CALRY. You can see what signature was associated with their software. What commands they attempted to run. What sort of files they attempted to pull down, et cetera? So there's a wealth of information beyond that summary information that can be shared in this fashion. So, what are the benefits? Why would your institution like to participate in this project? I think there's a few things we can talk about right off the bat. One is that the solution is essentially low/no cost. I mean, the software is free and all the pieces are available on GitHub. So in that respect there's no cost. But I say there's low cost because it does require a little bit of time to learn about the system and how to manage it. We've attempted to make it as easy as possible and packaged everything up in Docker containers so that once there's an understanding of how to use Docker, it becomes very easy to set up the systems and maintain them. Typically, if you were to set up a Linux machine and install Docker, set up a Docker composed file and put this under system D control, and turn on automatic updates for the
operating system, you basically have essentially an appliance-like situation where your machine can be configured to patch itself, reboot and every time it reboots it will spin back up the Docker containers and start running. So it's a very low cost solution for people to get access to the kinds of data that we're able to provide. Flexible design, using Docker you can distribute the components horizontally if you need to -- if you have a very large honey potting system. Or if you want to run, say, one that has honey pots that are focused just on internal traffic and one that's focused on public-facing traffic, you could spread it out DPEE graphically if you need to install multiple service, multiple places. It's very flexible and easy for you to break the components apart and spread them out or have many, many instances of this where you have a specialized purpose for each of them. And then the biggest operational -- the biggest operational benefit I see is the ability to take malicious traffic off your network. To really knock down that background mediation of the Internet that all of us see. I think this is a problem that's more common in higher education in that we have large blocks of public IP address face. We typically have a more open collaborative stance towards access to our networks and so we tend to run into all the issues that that sort of comes with. But this system is a great way to do very targeted blocking of known malicious attackers in a very timed-down way and just take that stuff right off your network. One of the benefits I'll talk about in a moment is that, you know, this actually really helps save on some of your resources further into the network as well. And finally, think there's another great benefit here in that we're trying to save all of this information into a research corpus where all the partners are anonymized. But any researcher at any of the participating institutions can get the entire corpus of data fed to them so that they can do research with this data. I think that while today the amount of data that is being shared is very important operationally but maybe not rich enough for research sets, I think, one, that's going to change as we move forward and people get more comfortable with sharing. And additionally, there's still lots of interesting things that can be done by taking this dataset and marrying it with local datasets such as, say, flow record sets for an institution. And I think that it really sets you up to start prepping for some things that would normally not be possible. For instance, those doing machine learning algorithms to detect anomalies and attacks. What they need to help train those models are labeled datasets. And as everyone knows, it's very difficult to sort of label the entire, say, network activity for a University of any size. But you can use this data as a way to say, hey, we know these things are malicious so we should be able to at least detect these things within our dataset. So those are some of the off-the-cuff benefits there. The history of this is actually that in 2014 to change their approach to how we were architecting our security. So the three sort of buzzwordy things that we did were implement automated defense with threat intelligence and deception. But these marketing terms are actually, you know, fairly simple concepts that just required some work to get done. So automated defense was scraping all of our available log sources, filtering them and looking for attacks against our network. And then using that information about attacking IP addresses to do blocking at the border. We also when we started doing this realized there are other great sources of threat intelligence out there that we can pull in. And we could also use that data to start doing blocking. And, you know, it was also a very short hop from, you know, we're generating our own data to do blocking to say, hey, we could just put this in a CSE and put it on a website and share the link with some firms and they could use this data locally as well. And then of course deception is the rebranded honey pots and honey
tokens and honey accounts and things of that nature. Very early on we found honey pots were a useful way for us to understand the scope and nature of attacks against our network. So that was one of the primary focuses as we started the STINGAR project. And by 2017, we looked back and we had realized that we -- we had gained a lot of benefits here. So when we used our vendor providing threat intelligence list to do blocking at our border, we were blocking about 10 million connections a day. Now, that's settled down a bit, but we still block on the order of 250 connections a day, and that's just based on our own threat intelligence that we've generated internally and through the STINGAR project. I spoke earlier about the operational benefits that can come out of this. There's the blocking knocking down that noise, but a big one for us was that we reduced our IPS alerts to about 9 million alerts a money to under -- under a million per month. So we had about an 85% reduction in IPS alerts month over month. This -- this was really not something we sort of anticipated, but it was a great boost for us in terms of we were no longer having to pay to index all these alerts. We were no longer having to sift through all of these things when we wanted to go do investigations on a particular machine, et cetera. That was a really big sort of unexpected benefit that we got from there. And we're also sharing, at the time, about 17,000 unique IP addresses a day. At this point we're typically sharing about 60 to 70,000 unique IP addresses every day. And I'll have some more stats on that in a little bit. When we look at the architecture, as I mentioned, all of these pieces are done with containers that we polish, and we push up to Docker hub that makes them easy for people to pull them down from a shared resource. And the central components look like this. So they are CHM servers which is community honey network servers. These points act as -- they're the central server for the honey potting system. And all the individual honey pots, some of which are listed here, can be deployed from this CHN server using basically a copy/paste dash script that will come back and talk to the server, pull down the right images, write out the correct configuration files and start -- excuse me, start up a communication channel with the CHN server to get all the logs off the honey pot and into the CHN server. As well, there's some additional loggings that can happen here from the CHN server. If you want to take all of the raw logs coming out of the honey pot and put them into your own central logging system, that's very easy to do with a Docker container and a small business file. I talked about these earlier, there's functionality to pull down feeds from the shared threat repository that are very easy for you to put into a firewall or router or build host rules and things of that nature. I don't know how many folks out there are learning a black hole router, but we found this to be a very effective blocking technique as a supplement to our next generation firewall. Part of that being about timing. When we first started this, our Next-Gen firewall would only update every 30 minutes. And when we looked at the statistics of how many IP addresses that were bad, we found that it was a surprisingly low number. So, perhaps, 35% to 45% of the IP addresses that we blocked, because we saw them attacking our network, did we actually get a block in place for. That told us that these blocks were far too slow. 30 minutes was too long to let the attacker keep going and many of them were, in fact, finishing their attacks and moving on. NCSA has a GitHub page for their black hole router which is something we've implemented at Duke. And we've also implemented another container that understands how to speak to a VHR instance and submit IP addresses seen by your server directly to a black hole router. So that's just another way that people can use to protect themselves. In our case, we feed those black hole routers in for one hour which gives plenty of time for those feeds to percolate through the system
and get pulled into our next generation firewall. There is a shared threat repository that all the individual partners can submit their summary data to. And are there we can do additional sharing, say, pushing up to an IVAC or using this data to feedback to partners who want to remediate issues in their own environment. Think of infrastructure as a platform -- as a service providers. We can provide information about, hey, you know, these IP addresses coming out your networks were seen attacking honey pots and you may want to alert the customer or take some other action to help shut these things down at the source. So that's the basic architecture. The participation model, there's a few here. You could today go out to the GitHub page for community honey network and download the server in four different honey pots and use the local CHN server API to build your own feeds and put that on your own net work. I think that's great. You can get some benefit out of that and you can start to understand the scope of the problem on your own network. But, I think it doesn't -- it's not quite as powerful as some of these other models. So the next step off is that you can have a central CIF repository. And individual institutions or organizations or even, you know, departments in your institution, if it's that fragmented, could be running these servers and then contributing that data back. I think this is where the model starts to become more powerful, because there's just a limit to the number of honey pots that people are willing to sort of run and the amount of benefit that you're going to get out of seeing things from just one network. I still think that number is, you know, probably in the several hundred range. But most people are not going to want to dedicate the resources to doing this sort of thing. So the real power comes in sort of spreading that out so that more people are able to sense more different attacks and then share that data back. And that's essentially what the STINGAR system is based on, right some. You could have these smaller enclaves of people sharing. Say, for instance, if the state of North Carolina schools wanted to get together and they only want to share with one another, they could do that and then trade their data back up to the STINGAR project and individual institutions can also participate in the stipulating sting project. Or they could share their data with their, you know, whatever their IVAC model is. There's a few different models but the power comes from everyone sharing and everyone sharing that load of sensing and being able to get the benefit of visibility from other people's networks. So here's a little bit of data that we've gathered out of the project so far and then after I get there thank you section I will turn it over to Joel to talk about their own experiences. But right now, if we look at about a month of data, we've got 25 institutions that are participating in this STINGAR project. That's 25 different networks at a minimum where people are deploying at least one honey pot. We have approximately 170 honey pots out there where I count a sensor as a honey pot. I don't actually know what that number is because we don't have any sort of feedback mechanisms in the project. So there's nothing reporting back from the CHN server to us that says, hey, there are this many honey pots deployed in these places, et cetera. We don't have any visibility into that. So this number is totally based on me serving participants and saying, hey, about how many honey pots are you running this month? That sort of thing. So this is a real estimate here. But one thing I can say is that in November/December for that 30-day period, we saw over half a million unique IP addresses. And many of these IP addresses we saw more than once and we saw at more than one partner. The point of this slide is just to say that the more partners that we see, the more new IP addresses we're able to identify and allow partners to take action on. I think that's a pretty intuitive thing is that the more partners we have going, the more sensors we have and the
more networks, the more attacking IP addresses we'll be able to see. In terms of the sharing, this slide was originally a nice diagram of the overlap between different partners, which worked really well with four partners and seven partners. But somewhere around 14 partners it really started to just look like a big mess. But I think the important thing to talk about here is that at least 20% of the IP addresses that are identified in the STINGAR project are seen at more than one partner. Now, there's an important caveat here, which is that if partner one sees an IP address and puts it into the feed and that IP address doesn't get to one of the other partner networks in, say, a few minutes, typically what's going to happen is for the partners who are actively blocking, they'll never actually see that IP address come to their network. So we won't be able to record the fact that there was overlap here. We think that number's actually substantially larger, but we don't have a great way to gather that data without having partners either record the blocks that are coming out of their networks and sort of look for those IP addresses and feed that back to us, or accepting their honey pots from the blocks so that even if an IP is blocked from talking to everywhere else in the network, they can still talk to the honey pots. Another interesting thing that we are seeing is that we're seeing a bit of a range on this, but generally speaking, between -- it takes about between five and 24 hours for an IP address to be seen from one partner to the next. Which is just to say that these attackers are moving very quickly. They're cycling IP addresses quickly and it's important that if you're going to go after this class of attacker and these types of attacks, that you react in an automated fashion and do blocking automatically. Because if you're counting on having a person to review this, first of all, I think the scale is too large and you're going to be too slow and you're going to miss the value that can be eeked out from using this data. So, I think this graph is really more of a restating of the previous one. Here we have buckets of when we were looked at about 450,000 unique IP addresses how many were seen by one partner, two partners, three partners, et cetera. This is a log-a-rhythmic scale here. I think it was 450 or 460,000 unique IP addresses we saw for this particular graph. About 350,000 of them are only seen by one partner. Again, that doesn't take into account the fact that many of our partners are automatically blocking based on these feeds. But we did see there are quite a few that do show up again and again partner after partner. I think that also highlights the importance of us sharing this data as quickly as possible between partners. So, with that, I'm going to turn things over to Joel to talk a little bit about their STINGAR deployment and how their experience went. Joel.

>> All right. Thank you. Once again, thank you, Brian, for the introduction. Thanks to the participants for spending time with us today and thank you, Jesse, as well. My name is Joel Faison and I'm the director of information security at North Carolina Central University in Durham, North Carolina. So I want to have a quick disclaimer. What I will be reporting on from our STINGAR experience is going to be from the lens of a networking director, this is the role that I had up until a few weeks ago. I wanted to make that disclaimer because in networking it's not that security isn't essential, but it's a delicate balance of trying to make sure that we can move data from point "A" to point "B" as quickly as possible. And sometimes technology such as, you know, packet inspection, encryption, decryption, route lists, if not managed properly can introduce latencies into the network. So this is just a possible concern that I had going into the project. I can say that we've discovered that if there are any latencies at all, it has not been
reported by our customer base. So, with that out of the way, we first met with Jesse and the information security team at Duke back in the fall of 2017 to learn about a project that they were working on that could provide real-time threat intelligence for networks. Around the same time at our institution, we were in the early stages of developing the foundation of our information security program. So a lot of the policy development had been completed and then we were starting to look at, you know, technical controls that we could put in place. We didn't have any real-time auto blocking capabilities. We had a number of subscriptions for our edge firewall devices, but as Jesse reported, you know, sometimes there were delays in what that information could be processed. Someone could be in your system and have moved laterally in just a few short minutes. So, the STINGAR initiative in hearing the presentation sounded very intriguing to us, certainly the cost was important. And we discussed ways in which we could partner and provide feedback as the project continued to be developed. We started small. We had a admin who was working with our Docker team and he was tasked with being our resource on the project. So we started with a single on premise honey pot towards November of 2017. Over the course of 2017/2018 as the project has been developed, we have increased to five. So we're five out of that 170 or so. And we're running honey pots on SMPT, RDP looking at 8443. FTP and MySQL and Microsoft SQL. So it didn't require a lot of resources both human and technical to get out of the gate. This is some of the resources that we used. This is certainly, you know, hardware that was several generations old. But we had a single CHN server that was running CentOS. And we do have four honey pots that are hosted in Azure and we've listed the -- kind of the minimal specs and requirements at which they can operate. As you can see, it doesn't require any of the latest and greatest hardware to be up and running. So the next couple of slides I want to start talking about connections that we began blocking. And this represents 2019, actually spans from January of 2019 up through just a week or two ago in 2020. And if you look from left to right, what you'll notice, that first jump which is around February of 2019, that is when we begin blocking -- automatically dropping connections based off of partner data. So we were already blocking based off of what we were seeing from our own honey pots. But that big spike that you see is when we started blocking based off of other members of the system. The next jump that you'll see is around May of 2019. What we did there is by default I think we were blocking connections for 24 hours. So if we saw an IP address in the honey pot, we would just drop it off after 24 hours. We later changed that to four days and there was nothing scientific about it, it was, you know, more of an art versus a science. We just said, hey, let's go with four days. And what we noticed is that a lot of those connections, we just saw a huge -- a larger increase, if you will. If we continued through the rest of 2019, you can see that there was a larger volume of connections that we were blocking. Towards the end of the year there's a big spike and then you see it drops off. The reason for that is just from our own instances we dropped, and we only started blocking again based off of our data. There were some infrastructure changes that were happening at the CHN network and then due to some maintenance on our end we dropped down. But we have subsequently started blocking based off of partner data as well. So this is another representation just looking at 2019. We were blocking on average about 32 million connections per day. What we've seen in 2020 is much larger. I think as of a couple days ago we're well over 100 million dropped connections or blocked connections per day. So you can see there is value as more partners are added. Just some of the main attack ports that we would
see, most of the attackers would be either coming in on telnet or proxy telnet followed by RDP. And then we can see HTTP traffic as well. The next couple of slides I won't spend a whole lot of time on. It's just we started looking at doing Geo lookups based off of the attackers that we were seeing hitting the honey pots. If I look at a ten-day period back in July of 2019, we can see, you know, connections being blocked from China. There's that one in Russia, looks like a little bit in Ukraine. And then we do see some that are sprinkled throughout looks like the extreme east and west coast of the United States. But if I contrast that to just last week looking at a 24-hour period, we can see a lot of those same attackers or connections are still coming in but added to that we now have South America, we have parts of Africa, certainly more from Russia, and then India and looks like a who's who of central and eastern Europe. I thought that was pretty interesting. Some of the technical notes that the CIF add admin on the team observed, you'll want to make sure there's certain firewall ports that are open. Specifically, A.D. and 443 are used for the honey pots to register back to the master server. And then TCP 10,000 is used for the HP feeds message cue. So that's the honey pot data. Another thing that we observed rather quickly is SSL certs. We were using certs that required -- but later we used let's encrypt for our SSL search. Finally, there's a hard limit to the amount of IP addresses that the Next-Gen firewall can process. Right now, for our device it's 150,000. Anything over 150,000 in those entries are dropped. I can say right now looking at both our data as well as the partner data, we're seeing a little over 100,000. So we have not hit the 150 limit just yet. Some of the takeaways that we've had over the past couple of years, you know, we’ve found STINGAR to be an extremely valuable tool for our network arsenal or network defense arsenal. We know in security there's no such thing as set it and forget it. But this is very close. I talked to the sys admin and he spends less than one hour a month managing it. Essentially when he has to touch it is when there are updates or patches. He said that doesn't happen every month and when it does it's probably about three hours to do those updates. So there's minimal human and technical administration that's required. The other thing that we like about it, you know, there's zero false positives. We haven't had anyone contact us asking to be removed saying that we are erroneously blocking their connections. And, again, by nature of what a hundred nip pot is, if we have a system out there and somebody's trying to hit it, then they do not have a valid reason to be connecting on the network. It gets stronger as more partners have been added and their data's added to the collective. One thing that we are starting to do more of now is we want to start being looking it at some of the analysis of the attacker data. One thing that we've noticed is that some of those same IP addresses that are seen on the honey pots are starting to show up as some of our cloud resources. More specifically, our office 365 tenant. So that's been our STINGAR experience, just in a nutshell. Again, thank you for letting me share what we've been working on. At this point if anyone has any questions, I think there's some that are in the poll that has been submitted already.

>> Thank you both, Jesse and Joel, appreciate that great stuff. We do have questions in the chat sort of queued up. The first few are for Jesse as he was first. The first question, Jesse, was from the University of Oregon, is there any other distribution method other than Docker?

>> So currently Docker is the best supported way. Officially when we were creating this project, we wanted to enable people to do both. Docker containers as well as to do installations on hosts,
you know, if Docker was, you know, not something they were prepared to do. But we found that mostly people were really interested in just using the containers that we built in order to do so. Now, that being said in the container space you could certainly use something like singularity to one these containers as well. One of our goals for the next version of this software to come out, version 1.9, is to have good support in the way we built these containers so that it's very easy to run these things in Kubernetes as well, which I think will be helpful for people to scale with. So I guess the shorter answer is, no, it's, you know, it's mostly Docker. But there are other options coming in terms of container technology and if you've got a use case for something that's not container, I'd love to sort of talk through it with you and see -- see what we could do to help with that.

>> Thanks, Jesse. The next question was an educational service providers participate in the project?

>> Yeah. So currently we've been limiting participation to education institutions. But I think we can take a fairly broad view of that. So I would be more than happy to sit down with anyone who's interested in participating in the project and seeing if there's a way that -- if it's a good fit and if that would be -- if that would be beneficial for both parties. So we're always open to, you know, trying to get as many partners as possible because that's, you know, as we talked about in the presentation, that's where the power of this comes into play is the more people participating. But we're also mindful that we're trying to do a couple things that do make us want to focus on the higher education space. One is, you know, as we're adding people to these lists and to -- and to this workspace, we try to keep it higher education only so there's at least some level of trust or some understanding that, you know, these are the problems that we're all facing and we're coming up with solutions for those. And not sort of going off in the wrong direction in terms of development and future requests and things. But there's also the aspect of the research corpus, right? So certainly in the future we could support no higher education institutions, but it's a matter of sort of figuring out the best way to do that while continuing to provide benefits to the higher education institutions, ensuring that the Intel we're gathering isn't just pack An

packaged up and resold back to our community and things of that nature. But we're more than happy to sit down with these things. If I could jump ahead of you, Brian, someone asked how many routes did we put in our real-time black hole. So that's an interesting question. We have done -- we did do some load testing with that. We took a fairly -- a bit long in the tooth router with about 768 megabytes of ram in it and we tried to shove as many routes as possible at it using the BHR software. And what we found was that somewhere around 450,000 routes the router kicked into CPU mode. And for those familiar with routers, CPU mode is not where you want to be because you're -- everything's going to slow down and things are going to probably fall over. That being said, we used this as a stop gap, which somewhat addresses the other question about the Next-Gen firewall integration with the 30-minute lag. You know, we put the blocks in place for days at a time in our Next-Gen firewall, but we use the real-time black hole in order to react very quickly to do these blocks. So what we found in practice was that if you talk to one of the honey pots that was connecting to our CHN server, we could have a real-time black hole in place in the order of two to three seconds. And that's a very -- that's a great turnaround
for, you know, getting a block in place very quickly. That being said, we were mindful of our blocking budget. So we only kept those blocks in place for an hour, which the NCSA project on GitHub, the HR site is what you could look for that. That already had logic in place for, you know, keeping -- keeping these blocks in place for whatever time you wanted. But we defaulted to an hour for the project, which we felt was a good amount of time, a good balance between, you know, reacting very quickly and, you know, making sure that you don't go over your route budget. And in our case, based on our sensor network, which is I think is probably one of the larger ones, we're looking at typically at any point in time when we look at the number of routes that are in place for real-time black holing, we're looking at about 1500 to 2,000 routes. Obviously, there could be spikes and things of that nature, but that's very comfortably within the 440,000.

>> So another question asked was what measures in place to prevent attackers from realizing their inside honey pot? We actually have not done a lot of work in that space. We've had a lot of ideas in that space, et cetera, but what we've honestly found is that a lot of these attackers, I mean, their attacks are automated and they don't have logic built into their attack scripts and scanning scripts in order to detect Lon they're in a honey pot. So we're still getting a tremendous amount of value, even with the low interaction honey pots. These are honey pots that offer us information on a particular port and log the connection and then don't give any additional feedback. So they're basically just a port that's logging. And even with the higher interaction of honey pots such as CALRY, we don't see a lot of attackers who are actual will you actually doing anything to test whether they're in honey pot or not. Sometimes we do see human attackers logging in and they do eventually realize that they're inside a honey pot when a bunch of their standard commands don't work, et cetera. But we also offer support for people to customize their honey pot looks so you can change that services are available, what banners they present, things of that nature. It's a supportive thing but we're not publishing a lot of, say, recipes for what these look like because we don't want to start an arms race. We've been milling about ideas for how we could build repositories so that we don't all have to do the same work. But we don't have anything that's ready today in that space. But it's definitely an area of interest and an area where the people are getting benefit out of the project and they -- and they have the time and the resources to do so, it would be great to see some poll requests come back to help with that. Someone did ask a more specific --

>> I think we might have just lost Jesse's audio for a moment. Maybe we'll jump over to Joel. There's a couple of questions in the queue queued up for you around NCSC and supporting adaptive rules and a few others if you want to jump on those.

>> Sure. Thank you. So the question on NCSC ran for those outside the great state of North Carolina, that is our research and education network. And so William, which serves as the ISP for the UNC system as well as a lot of the private universities in the state. So, William, to your question, to my knowledge we have not approached NCMC ran asking if they're able to start blocking or doing any filtering based off of what we're seeing from the honey pot data. Once Jesse comes back I can see if Duke has approached him. Sorry for that answer. There's a question
from Brian on white list feature. So one of the things that we did do, we wanted to make sure from the very beginning that we weren't blocking certainly any of our own IPs. We also have a lot of hosted services so we wanted to make sure that we weren't blocking any of the IPs from those as well. And I see on the panel that the sys admin I spoke about said there's a config option that's prepopulated.

>> Hi, everyone. Can you hear me now?

>> Hey, Jesse.

>> Hey, Jesse.

>> Hi, great. Okay. Very sorry about that. I'm not sure at what point I got cut off because I could still hear but my adobe connect -- so if someone could remind me where I was. Top question, okay. So there was a question about the Next-Gen firewall and the 30-minute lag. Was it my direct API? Was there an issue with state sessioning to be cleared, et cetera. Yes is the answer. Well, really using the built-in functionality for security intelligence lists to get that data in there and we didn't want to go sort of direct API route because we didn't want to build custom code and we didn't want to have to support that aspect as well. So we focused on use the mechanisms that the vendor provided. That being said, we did have to open at least one case where we were able to show that things that were on our blacklist were still actively communicating with our infrastructure and making it past our firewall. And the vendor did ultimately uncover a bug in the way that they were handling their lists and had to provide a patch to -- to deal with that particular issue. You know, we've tried to look at some of these other mechanisms for some of these other firewalls in terms of, you know, could we help people build script that would, you know, simulate logging into a command line and adding to an ACL, et cetera. And the problem becomes that, you know, if the vendor doesn't give you this sort of functionality built in, there's a great chance that it will be done in a way that's not really optimal. You know, keeping track of what's in there and what's not and even just, you know, this was a concern our networking folks brought to us, you know, if we were to route an ACL, not only do we have to keep track of how big that ACL is getting, keep multiple ACLs to understand how big and how many ACLs a particular platform can support, just the amount of dynamism in these lists where things are falling off and being added all the time and we're updating typically every five minutes could actually be really difficult on the -- on the memory for these devices and could potentially cause a failure. So we are trying to basically stick with what the vendors provide in terms of doing these dynamic lists. And I think it's a great space for our community to sort of push back on the vendors to get more about -- to get more support and this sort of thing. I will also jump to the next question I see here, which is can other honey pots feed into this? What he were set up for today is all based on HP feeds. HP feeds was essentially a pub sub message bus that was created back before there was message buses and now there's a number of good ones out there. So I guess the short answer is not really. And the long answer is, yes, if you can get it to speak HP feeds. And I think the even longer version is we're working on a version 2 of this product that will be a complete ground-up rewrite of this and it will use entirely different mechanisms for
moving data around and getting honey pots integrated. I think that we'll have a lot more flexibility for adding in additional honey pots as we move along. And then it looks like the last one that we have here about plans to include CIF V 3. We are not directly intending to support stick taxi feeds. I think that I may be making this up, but I think there are at least in some versions out there was a way for the native CIF client to output in sticks format. And so that could be a way for you to get that data out of the project and into a format that's useable for you. That's another reason that we really liked using this CIF project as the back end is that it came with an CDK and a client that would enable a bunch of different output formats. For instance, using the CIF client you could query the STINGAR CIF instance and pull back a feed and output it in the Intel framework format. So if you wanted to use this for detection mechanism with the Zeke infrastructure, the CIF client makes that a trivial thing to do. And it can also output in Json and CSV. I think CIF can do it and if not it should. And that's an area that we can go back to those developers and see if there's any collaborations we can do.

>> Well, on behalf of EDUCAUSE and our speakers, this is Brian Kelly and I thank all of you for joining us today for an engaging session and conversation with some great questions. Before you sign off today, please click on the session evaluation link. Your comments and responses are very important to us. The session is recording, and presentation slides will be posted to the event site. Please feel free to share these resources with your colleagues. And finally, please join us for the next EDUCAUSE webinar on January 28th at 1:00 p.m. eastern time to learn about privacy in 2020. Where have we been and what's next? On behalf of EDUCAUSE, this is Brian Kelly, thanks for joining us today.

>> Thanks, everyone.

[End of Webinar]