Harold Byun:

Hi. Good afternoon. Good evening. Good morning. Welcome to today's webinar. This webinar is actually going to cover threat Intel sharing as a case study using a secure data sharing platform. And so we're just going to give people a couple more minutes to jump in. There are attachments and links in the attachments area. There's also the chat area, if you have questions that you want to ask as we go through the demo, and this will actually give you an overview of what we're going to present today.

Harold Byun:

Actually for some reason, this actual slide deck does not seem to be the correct deck. What I'm going to do here as we get in is I'm actually going to just share my screen and go through the deck. For some reason, there seems to be confusion over which deck, but you'll get an updated one here. Give me just a moment, a couple more people are coming in and then we'll start very quickly.

Harold Byun:

(silence)

Harold Byun:

All right. Well, thank you again for joining today's webinar on secure data sharing. This is going to be a case study on threat Intel sharing across multiple parties. My name is Harold Byun. I head up product management for Baffle, which is a data centric protection company. Today, I'm going to be walking through a number of different scenarios around threat intelligence sharing, how they can improve incident response and what we can do to help facilitate broader sharing of threat Intel data.

Harold Byun:

The agenda for today, and hopefully we can cut through this in about 30 minutes or so, might be 30 to 40 minutes, is really an overview of threat Intel sharing benefits and challenges. I'll be giving you a really quick primer on the techniques that we use to facilitate secure data sharing, mainly privacy preserving analytics, and a methodology known as multiparty compute. We'll give you the architecture overview that we're going to be going through around the secure data sharing platform, the demonstration of how this actually works, then we'll get into the weeds a little bit about how this actually works underneath the covers and Q and A, feel free to ask questions throughout as we go through things. Again, happy to answer any questions as we go along the way.

Harold Byun:

A little background about me. I've been doing security and related security technology about the architecture side prior to the vendor side for about 25 years. A lot of that focused on data containment. And so this is my latest move in my career to really drive opaque computing or secure computation on data with Baffle.

Harold Byun:

Let's kind of go into threat Intel sharing in general and the overall benefits and challenges around threat Intel sharing. If we really look at why people are looking at this as an overall potential enhancement to security response, in many ways, it's really a scenario where you're looking at the overall threat detection and response timeframe. And so this data's from the Pony Mon Institute study on the cost of a data breach.

Harold Byun:

And in their analysis, and this has been a metric that a lot of people have been tracking for quite some time, is that the incident response time in terms of the ability to detect a breach has been very challenging for a lot of organizations. And the average time to identify a data breach has been 197 days. Used to actually be longer. If you go back a few years, I think at one point it was upwards of around 200 and 206 days. And so when you look at that as a metric, what it means is that attackers are inside the environment, sitting and wait or gathering information or looking at exposure points within your network for a period of over half a year on average. And obviously, there are significant breaches where this statistic gets blown out quite a bit, but the net net is that it is far too slow in terms of the ability to actually detect an incident response or detect that a breach has actually occurred rather.

Harold Byun:

And so there's something that people are looking to compress the time to identify, and that's been a metric that a lot of the security incident response team has been tracking for quite some while. And so this is an area where people are obviously looking to compress that timeframe and then obviously drive containment, which is also excessive.

Harold Byun:

And so how can threat Intel sharing actually help? Well, it can reduce the amount of time spent on analyzing and confirming indicators of compromise related to a security incident so that the response time and the detection timeframe can actually be compressed quite a bit. In addition to that, what we can also do is facilitate a broader context of the frequency and the prevalence that a given IOC may be actually exhibiting. And so these are areas where things can definitely be enhanced.

Harold Byun:

And so the classic example that you might have, which is often done ad hoc by a lot of organizations, is you have on the left, this incident responder one or security analyst number one saying, hey, I'm seeing some strange behavior from this executable or potential malware, or this machine, or some hosts that's exhibiting strange behavior. And then they ask a peer of theirs in the industry, and this is kind of the foundation of a lot of the ISAC and ISAO or information sharing and analytics organizations. And then they ask the question, have you seen anything like this? And somebody might respond, yes, I've seen it 10 times last week. We've marked it as known bad, or people may say, I've never seen this, it's net new or an unknown or a false positive.

Harold Byun:

And so these are things that eliminate a lot of the overhead that a security analyst may have to take on in terms of finding a way to detonate potential malware, looking at the behavior over a period of time of a potential executable, looking at how it might be moving laterally or other places where you may find this in your network, and it basically can accelerate the confirmation timeframe for a given IOC. And so that's ultimately going to compress the overall incident response time that we were just talking about on the prior slide.

Harold Byun:

When we look at well, why doesn't everybody just do this? Well, the challenge with threat Intel is that I think the number one item that everybody in the industry seems to be very familiar with is that everybody consumes, but nobody really participates or shares. Part of the reason that this happens is because obviously everybody wants to consume, it's great to get this additional information and these additional data points, but nobody really wants to share because of item number two, because of the concerns over attribution of the threat Intel data.

Harold Byun:

The challenge with this is that if I share or participate, then there's the possibility that my company, as for example, let's say I work for company Acme or some made up fictitious company, and I share threat Intel that is somehow attributed to my company Acme. Then I've effectively raised my hand and said, by the way, I was attacked by this type of attack that is manifested in the form or maybe represented by these IOCs and these attack methods. And so it attributes the fact that A, I've been attacked, B, it might've been successful and C, that it eludes to what my security posture is given the nature of the methods of those types of attacks.

Harold Byun:

And so I'm really sharing a lot more information than just a given IOC value. I'm sharing a lot of information about the security posture of my organization, which increases the risk, which leads to step three, that if attackers or other parties gain access to the IOCs, then I now have a lot of information on how to bypass detection in your organization. If you're using any type of signature based method, if you're using any type of intrusion prevention or intrusion detection systems, if I know the IOCs that you're looking for, then I have a really good blueprint on how to start looking at ways to bypass that detection method.

Harold Byun:

And that ultimately begs the question of, well, what if we could ask questions of data without actually exposing the data values? What if we could actually share data without sharing it per se? And so that's kind of the notion of this secure data sharing platform.

Harold Byun:

The techniques that we're going to demonstrate and utilize in this session here is actually a category that's been dubbed privacy preserving analytics and [secure data sharing.](https://baffle.io/secure-data-sharing/) And so I'm going to give you a really quick primer on what this area of computation entails and then go into it, and then after the demonstration and walking through some of the use cases, I will go into kind of some of the technology we use around how we do this at a relatively still high level, but it'll give you a better flavor of what's going on.

Harold Byun:

Privacy preserving analytics, it's a computational method that allows for operations and processing and analysis of data without revealing the underlying data values or violating the data privacy contract. Net net of it, you can basically analyze aggregate data on encrypted data values or protected data values, return answers on your questions, but never actually reveal the underlying values that are being queried for. And kind of an extension of this is secure data sharing. This is the method that allows for this to occur across multiple parties.

Harold Byun:

And so obviously within the realm of threat intelligence sharing or any type of detection response scenario, fraud is another one, for example. This is a method that can be used across multiple parties, again, without violating the data privacy contract and it works under some basic assumptions. One, the assumption within this model is that the data store is untrusted or under active attack. And so classic examples of a data store that is untrusted or well untrusted for a lot of people in the security industry, that represents something that's sitting in the cloud, where for whatever reason, you could argue whether it's rational or not, people don't trust the cloud provider or the cloud hoster, or it's some kind of joint colo. And therefore the data store is considered untrusted under active attack. Well, you could assume that almost any host attached to any network is under active attack in today's world, whether that be in cloud or on premise.

Harold Byun:

And then other centers where this comes into play is foreign nation states. If there are scenarios where multiple nations may be wanting to share information, if there's a scenario where a company is hosting information in a foreign nation state, and for whatever reason, doesn't trust the hosting providers in that given country, those represent untrusted data stores that may be under active attack. And so that's an assumption where we assume that the data platform is effectively compromised or in a compromised state.

Harold Byun:

And so once you assume that then it makes it obviously very difficult to take the leap of faith. Well, why don't we just throw a sensitive data into that data store then? The logic and the secondary assumption is, well, the sensitive data in that data store is actually protected and there's no method to decrypt it or to decrypt another party's data in that data store. And so obviously that requires you to take the leap of faith that encryption is at work and quantum has yet to occur and all of these other factors, but we believe that there's still at least at a minimum a five-year runway and most industry analysts project a 10 to 15 year runway on quantum. If you believe that, then the data in that data store is secure.

Harold Byun:

And so the technique that we're going to be relying on is known as secure multi-party compute or SMPC or MPC, and this is a cryptographic method that allows multiple parties to jointly perform any computation over their inputs or data values. It's a technique that's been available for over 30 years. Google cloud implemented a variation of this last July, and that's the technique that we're going to be utilizing in this particular example.

Harold Byun:

How does this actually play out from an architecture standpoint when you look at a secure data sharing platform? Well, if we go back to the premise and the assumption that the data store is untrusted, we in this example have two separate compute domains and I've represented them as VPCs. They could be spread across party one and party two. It could be a neutral party. It doesn't really matter. The shared encrypted database is an encrypted data store, again, that relies on the assumption that it is untrusted and under active attack, and in this particular model is encrypted data with no encryption keys present in that compute domain.

Harold Byun:

If you have an encrypted object with no key present, then there is no way to decrypt it. This is no different than sending a backup tape to iron mountain that's encrypted. There's no key present. There's no way to decrypt the data on that tape. Although, I'm dating myself a little bit by talking about tape, but at any rate that's what's represented at this top icon.

Harold Byun:

And then the bottom icon is what we call the servlets that basically allow for the multiparty compute or computation to occur. And so these are kind of the two compute domains that we're operating around. And so if we had an organization, say organization one, what we're going to be doing here is we're going to be leveraging their key store to encrypt data as it goes into this shared encrypted data store. We're going to actually encrypt data going into the encrypted data store using their key, and then the computation will occur using SMPC servlets. And if we had an organization two, similarly they would use their own key and encrypt the data before it ever hits that encrypted data store. The data never leaves the premise of the org in the clear or the compute domain of the organization in the clear, and they're using their own keys in this model.

Harold Byun:

And so the way this plays out is if organization one, and this could extend to organization two through N. In this demonstration, we'll be showing you three participating organizations, but it really doesn't matter how many are participating. Organization one's going to publish this IOC one. It could be a hash, it could be a URL, it could be IP address, whatever it is, and that gets encrypted for simplistic purposes. We're representing that as ABCDEF. And so they want to protect that information because they don't want to broadly share it and they don't want anybody to know or be able to list out all the IOCs that they're publishing.

Harold Byun:

There are no encryption keys present in the shared database, and there's no access to those keys within that compute domain. And then organization two also publishes and IOC, and they're going to encrypt this IOC. It's actually the same IOC. We'll call it IOC one, and it gets encrypted for simplified purposes as 123456. Now, we have these two hidden values that are actually the same, but they're represented by two different encrypted values in the secure data store. And what's going to happen is organization two is going to wake up the next morning and the security analyst is going to see some kind of alert in his sock dashboard. And he's going to say, oh my gosh, this IOC one's all over the place. I need to know kind of what's going on with this IOC one. How much have we seen it? How much have other people seen it? It's not going to query the secure database looking for IOC one. And the encrypted query comes across as another randomly generated value, because again, we want to hide what questions people are asking and what answers are being given.

Harold Byun:

And so the IOC one question, does this exist, actually gets encrypted as XYZ789. And so what's actually happening in this secure threat Intel or shared encrypted data store is we're comparing whether or not XYZ789 is equal to ABCDEF and 123456. Three randomly encrypted values and running a unique comparison to see whether or not these exist from a prevalent standpoint. And so this is what effectively is keeping the data hidden, but still allows an organization to ask questions of private data values.

Harold Byun:

And so let's kind of look at what this actually looks like. What I'm going to do here is switch into my data view. And so what I've done here is I've set up a direct connection to a database, and then I have organization one, organization two and organization three. There's a three-party share, and you can see that if I run this query here, I have a bunch of tables and me focusing mostly on IP addresses and hashes, just for this particular example.

Harold Byun:

Organization one, if I go into this and run the same query, we're going to get same set of tables. Organization two, just to kind of show you that these are live queries, we'll kind of juggle back and show tables, same table set. Same thing with organization three, same table set. If I actually go into the direct database, again, assuming that this database is compromised or hosted in an untrusted store, and I did a select on IP address, as you can see that. In this case, we've hidden all the IP addresses. In the comments, I have attributed the organization's just for simplistic purposes, because this is the only way for you to easily see that these are segmented by word with hashes. If I did the same, the hash values are actually encrypted.

Harold Byun:

The data is encrypted in opaque within the data structure. If I go direct, so if I was an attacker and got access, I would not be able to see those IOC values. If I looked at this from organization one's perspective, you can see that I can see the stuff that organization one has submitted, but I can't see any of the other stuff that organization two and organization three had submitted. Similarly, if I was organization two, I can see my stuff, but I can't see anybody else's stuff. And so if I was organization three, just to play this out, I can see my stuff again, but I can't see anybody else's stuff.

Harold Byun:

And so if I look at this and then I say, well, I want to see what the prevalence of say this IP address 5555, obviously dummy data, and see what that looks like in this given environment. When I run that frequency count, I'm going to get a count of one here, but what if I actually changed this to a different value? Like 2222. I run that query. I get a count of three. And so when I go back to my distinct data set, I can see that I have a 2222 here. I cannot necessarily run or see in the clear these other data values, but I do start getting a frequency count of what actually is available across the broader pool of participants, and this can extend a little further.

Harold Byun:

One other way that I could play this out is what if I wanted to give specific context? I wanted context, I could look at the date, the number of sightings, other comments that might be available around a given IOC that I'm looking up. Similarly if I ran that, I can see that this is available from org three. And the way that this plays out, even further, just to kind of show you kind of a live data set around this, I have this here somewhere. I could also run this straight insert from org three. I'm back on org three, and you'll see that I have one occurrence of 5555, and what I'm going to do here is I'm going to insert a new entry for this IOC. And so I've inserted this 5555 and I've labeled it Harold's Nasty Attack.

Harold Byun:

And if I go back to org two and run the same query, I'm not going to be able to see that entry. It's right here, Harold's Nasty Attack, but if I was able to run context searches on that value, I would get the context around a given IOC value that's been submitted. And the same principle applies. I mean, I know people sometimes are very literal. If I do the same thing for hashes, we could do a set of hashes and hash values, again, where I can only see my own hashes.

Harold Byun:

And so if I insert another hash called Harold Howard's Malware, so I will insert this and then run that query again. I can see because I'm the submitter Harold Howard's Malware's here, but that value is going to be hidden from somebody like party number two. Or if I run that same query, I'm not going to see Howard's Malware. But if I wanted to search for that hash specifically, then I would get a return and would get context. And so this is one method, again, where it facilitates ways that we can share values across multiple parties without exposing all of the information.

Harold Byun:

Another variant of this is taking it a step further is doing something around aggregate analytics. And so this is an example where we've set up another threat Intel sharing store, and we've encrypted target IPs and threat data, as you can see in this backend. In this particular example, what we've done is we front ended the dataset with Tableau. Now, this data set obviously is exposing the data values you could restrict the dashboard, but it is running frequency analytics on a given IOC, and it's representative of how you can easily facilitate this type of privacy preserving analytics using off the shelf software.

Harold Byun:

We're able to facilitate effectively what we call a no-code model of supporting applications to operate on encrypted data values, and Tableau's kind of the validation point for us because it is running and rendering data visually that is encrypted. And there's no way that we could have modified the Tableau source code. This is kind of another variant of that use case.

Harold Byun:

And then finally, just to give you an extension of another use case. Another use case could potentially be around patient data. And so obviously, probably very relevant. Many of you are probably working from home today, but if I did something like this, where I went into patients, we can see that we've encrypted SSN, driver's information, birthplace, address, a bunch of information, passport and first and last name.

Harold Byun:

If I wanted to run a query where I wanted to say, well, show me the social security number, birthplace and has conditioned one from patients. We don't know what has condition one is. Again, I can see my data, but I can't see anybody else's data. And if I went to party number three, and I said, I want to see all the patient's SSN and has condition one, again, I can only see my data, but if I wanted to start looking at frequency counts of all the patients from a given region that have condition one, I can run that frequency count and get a return. And so there's the ability, again, to ask questions of sensitive data without revealing all the underlying information.

Harold Byun:

And so the point being here that there's a number of ways that we can support a variety of different use cases. This is another variant of what we've just shown with fraud detection. This was an example where a financial services company wanted to combine banking, online banking timeframes with cell tower data, and neither party wanted to share information on their user base and in many cases were prohibited from sharing that information, but being able to perform these types of joins on shared input data is what this model actually can facilitate.

Harold Byun:

This is another model around third party data access controls, so people who are publishing reports and want to segment what parties can view what data. There's multiple scenarios across industries, where this comes into play, and there's also multiple data leakage gaps via third parties that have also come into play. Roughly 60% of enterprise organizations have had a data leak via third party. And so obviously it remains a pretty big threat vector.

Harold Byun:

Cross-party marketing. There are a number of scenarios across organizations that don't want to share, again, their user information, but still want to yield the benefits of shared information stores.

Harold Byun:

How does that all work? And so hopefully this is of interest to you. I'm going to walk through how it works and then we'll wrap up and open with questions. The technique that we utilize is, again, known as secure multi-party compute and the way that this functions is if we, again, take each party's own key, we encrypt the data as it goes into the data store and we utilize this secure multi-party compute method.

Harold Byun:

What actually happens is when an operation is conducted on the encrypted data, the operation is sent to the SMPC servlets for calculation on the encrypted data, but the encrypted data never leaves the data store. And so this is the ability to operate on those values of data without ever actually seeing the encrypted data values, and I recognize what that sounds like. A lot of people for many years have thought that this is impossible. It is a capability that has been proven out over many, many years. We've proven it out. Microsoft has an initiative around this. Google cloud, as I mentioned earlier, has implemented this in July. We've been vetted by multiple third parties and universities on this, as well as several financial services institutions with their cryptography groups. And we're happy to get into more depth and detail around how this actually occurs.

Harold Byun:

But basically what happens is once we handle that operation in the separate compute domain, the results are returned in encrypted form and then are actually served up in a result set as we've demonstrated in the prior example. This is kind of the 10,000 foot view of what's going on underneath the covers. For those of you that are interested in more in depth analysis, we have white papers on this. There's a presentation that our CTO did at some cryptography conferences and there's additional white papers available from academics that have written on this topic pretty profusely. Happy to share that additional information, if you have an interest in that.

Harold Byun:

In kind of closing, before we get into QA, there are practical methods that are available today that can easily facilitate this type of secure data sharing. When you look at the context of how people need to share information and particularly in situations where the time to respond and the time to react are critical, whether that be for a cybersecurity incident or proliferation of some type of healthcare situation, there are a number of different scenarios where accelerating that information sharing, and it could also tie to a business go to market. It's just whatever it is in terms of the ability to unblock people from deriving business intelligence and analytics from their information in a joint model, and again, without violating the privacy and trust model.

Harold Byun:

That's the heart of what we do. There's a number of resources that are available for you. If you go to baffle.io/secure-data-sharing, there's a white paper there. It's also in the attachments for this webinar. There's a Gartner report on privacy preserving analytics, and we're also running a free trial for secure data sharing, if you have an interest in trialing that out and seeing for yourself how it could or could not work for you. We're happy to make that available, and you can do that at that page as well.

Harold Byun:

So with that, I will open this up for Q and A and take any questions that people have. There's a few questions here and again, happy to answer any additionals. Benefits require how the data is encrypted. Does this require changing how the data is encrypted or stored? No, well, the simple answer is no. I mean, it is straight encryption that is utilizing, again, the encryption key stores or key material from each respective participating party. It does not change how the data is actually stored in that secure data store and it's encrypted using ASO. Hopefully, that answers your question there.

Harold Byun:

Does this require the other party to re-encrypt their data? Well, again, the data is always encrypted as it leaves any given party. There's no re-encryption per se. There is no re-encryption. It's encrypted as it leaves the organization and the deposited into that secure data sharing platform.

Harold Byun:

And then does this require each party to use your encryption software? No, it's not required. I mean, the methodologies and the model around multiparty compute and the ability to perform computation on encrypted data have been around for awhile. Somebody could build this themselves if they want. I don't think it's an easy solution to build, and we obviously are a software company and would prefer to have you as a customer. That said, it's not to prevent somebody from building it on their own.

Harold Byun:

If you actually happen to be associated with an ISAO or an ISAC, we're also happy to have a discussion around how we can, again, facilitate broader information sharing and make the software available. And as I mentioned earlier, there's also that free trial.

Harold Byun:

How long does this take to implement? It really depends on the nature of what you're looking to do. Some of the examples that I've shown so that the Tableau example, an hour to two hours of setup, depending on, again, the infrastructure that you need to set up. If you're doing it in cloud, we can deploy our solution in a matter of minutes. The trial is available for you as well. We can also run it as a hosted service in your compute domains. We could run it as a hosted service in our domains, or you can just take the software and stand it up on your own.

Harold Byun:

How is this different than FHE? So FHE is short for [fully homomorphic encryption.](https://baffle.io/blog/why-is-homomorphic-encryption-not-ready-for-primetime/) For those of you who are heavily into the encryption space, this has been considered for many years, the holy grail of encryption, which is the ability to run any mathematical operation or computation on encrypted data. The way that this is different than FHE or fully homomorphic encryption is one, we can support any ad hoc query. One of the fundamental differences between us and FHE is homomorphic encryption requires you to kind of pass the intent of what you're actually asking via the application in order to utilize the homomorphic encryption technique to derive the answer. So meaning that if I wanted to run a search, I need to be able to pass a specialized query and equip homomorphic encryption to respond to the search encryption or a mathematical addition or multiplication.

Harold Byun:

The multiparty compute method that we've implemented allows you to perform any ad hoc query. And so what that means is that you don't need to modify the application or the application query model. That's kind of one of the fundamental differences. And then the second difference is we're about a million times faster from a performance perspective than FHE, which has been notoriously slow and not in our view viable in today's world. Maybe with Moore's Law over the next half decade to a decade, I think that they'll come closer on the performance side of things, but that remains to be seen.

Harold Byun:

Hopefully, this is of interest. Are there limitations to the types of queries? So I think I kind of just answered that. The answer is no. We can explore any ad hoc query and virtually any mathematical operation. In certain applications, the technology has been proven out for wildcard searches on. If you're familiar with wildcard search, it's just star, a portion of my name or something like that. We can support those types of searches on fully encrypted data without ever decrypting the data in memory or in process.

Harold Byun:

Do you have any government clients? We are working with several government clients. We do not claim them as customers today. We are also going through the FIPs 140-2 certification process and have implemented the standards for encryption modes. In the very near future, we expect that to change, but that's just current state of things. We wouldn't be going through FIPs 140-2 certification without kind of the government client requirements anyway, and we do have several financial services and international clients as well.

Harold Byun:

Again, so thank you all. I hope this was useful information. Again, there are the attachments and the links. Feel free to go through those resources. Feel free to reach out to us if you have any additional questions and thank you again for your time. Have a great day.