Harold Byun:

Hi and welcome to today's webinar session. We will be starting in just a couple of minutes as people continue to log in. So please just bear with us for just a moment here and we'll get going. (silence) All right, well, while we get going here, a couple other folks just kind of getting in here. Thank you for taking the time to join us today. My name is Harold Byun I head product management for Baffle, for those of you who are familiar with is hopefully. We are a simplified encryption solution that provides data-centric protection with no code changes required. As we go through this, I would like to have you feel free to ask questions as we go along. There's a lot that we want to cover today. And I got a little note here that somebody is getting an error.

Harold Byun:

It could be a proxy problem. We've had some problems with folks who are on premise or coming through a corporate network, and your best bet there is you can listen in and there will be a live streaming media replay. And that seems to have worked for folks who get issues connecting here. So I apologize for that, but it's a network security issue or a network issue that's beyond our control. So I'm going to go through kind of this demo today it's the bare bones walkthrough of how to set up baffle, highlight some caveats here, what we're doing from the setup. It will be a bare bones set up, so you'll get to see not all of the good, the bad, the ugly, but some of the ugly and some of the good hopefully.

Harold Byun:

And this could go really, really well. And or it could not. So grab some popcorn and your beverage of choice. And hopefully this will be an interesting session for you. I'm trying to get through all of this in a half hour and open up for some questions. I'm happy to go in whatever direction people want to go in. So the agenda for today one, two, three, four, and encrypt. So we're going to basically go through four high of steps and encrypt data. We'll cover some of the architectural review during this, and then time permitting we can get into our simplified data masking and exfiltration control capability, as well as our record level encryption solution, which is something that a lot of people are using given some of the pending data privacy regulations. Q and A kind of throughout again, use the chat panel in posts, feel free to email us info@baffle.io my emails is harold@baffle.io whatever means is going to work for you.

Harold Byun:

So let's get through the high level of what we're actually planning to cover in today's session. So one, two, three, four, encrypt. And so basically when we look at how we actually are going to encrypt data, it's going to follow these four steps. These are not universal, there's multiple deployment mechanisms and methods for our solution. Our solution at a high level allows you to stand up a data protection service or a data protection abstraction layer, but it is a software. So you can deploy it on premise, you can deploy it in cloud, you can run it into hybrid mode. People have dockerized it, people are running within a Kubernetes framework. It can be fully wired into a CIC pipeline and deployment model for continuous DevOps deployment. So there's a number of different ways to go about doing this.

Harold Byun:

I'm going to walk you through a couple of those tasks today and again, we can go in whatever direction you all want to go in. So the first step is obviously configuring the baffle manager. This is probably the least exciting portion of the demo. There are some key prerequisites that we'll kind of acquire all kind of cover. Then we're going to in lay terms, connect the key store or a key management solution, could be any KMS or HSM or cloud key manager. It doesn't really matter to us. And we're going to connect a couple of databases, and then we're gonna map an application out and map the fields that we want to encrypt the keys, and we're going to encrypt the data. So that's kind of the agenda for this session. Again, in roughly 20 minutes or so, I'm going to try and get this done. We'll see how it goes.

Harold Byun:

When we look at kind of the way that this is all going to work, and I'll come back to this, it's going to start here in the upper right-hand corner with our admin console, which is the baffle manager. In many ways, you can consider what we're doing, encryption orchestration, which is we are marrying keys to data to protect data at the source. And so the way we're doing that, is we talk to a key management system or store or an HSM. We never persist your keys. We don't want your keys. We don't want your data. We just are providing a mechanism for you to protect your data. And the way we do this is we basically orchestrate the marrying of the keys with the data through our main component, which is the baffle shield. And the baffle shield is a TCP wire protocol, reverse proxy for SQL and NoSQL traffic.

Harold Byun:

And so what I'm going to do in this demo is I'm going to set up again, step one, the baffle manager, and then the last step is really step four, which is mapping the encryption keys to the application. So let's start with step one, a baffle manager config. This is the initial setup of baffle manager. Just some keynotes here as I go through this, I'm not going to spend time setting up security groups. I think that's worse than watching the grass grow. But just from our experience, working with a lot of customers, security groups, [inaudible 00:07:16] permissions, privileges, if you get that all teed up, then this thing can go in the 20 minutes that I'm going to show you. If you're tied up with security groups and everything else or privileges, or you don't have people with the right access to the right VTCs, then, that's where things take a little longer, but typically bare bones set up. I did the security groups this morning earlier.

Harold Byun:

So for this particular setup and spin up these instances. And so, you're looking at roughly an hour to two hours end to end when you get everything teed up. So we are sticking some of those prerequisites. Again, the UI is also not required. I'm showing to you, we can spin up in cloud formation templates. We have YAML and Jason files and templates, and you can upload, and you'd be spun up in three to four minutes with a deployment up and running. So I'm going to do... Let me first do the baffle manager config set up. This is again the least exciting part of this demo but we should be able to get at it. So you'll see I've spun up this brand new baffle manager, and I'm actually logged out because we don't spin up in initial mode for connecting in.

Harold Byun:

So I actually need to get into this instance. I can move this thing out of the way, I'll be able to get into this particular baffle manager instance. And so the this baffle manager instance is brand new. So this is just the first step. This is the only time you should have to get into this console, but it's to get the unlock key. So I'm just going to do that. And then I'm going to barrel through a lot of the setup because it's just not that exciting, but I think we did promise or somebody promised the bare bones set up for the solution. So I've unlocked the console, put in our domain name and our ordered name doesn't. And what I'm going to do is... This is a kind of an email notification capability. I'm not going to bother with configuring this too much, just because this is a demo. And then I set up my initial admin account.

Harold Byun:

And again, turn to you to ask any questions as we go along. So this is actually a credential key store. So we did talk to key managers and we did talk to databases. And so what we're able to do is basically specify using an actual encryption key manager, like a KMS or an HSM as the source of the credential, as of the encryption key, or what we can do is set up a local key store and we utilize a secret pass rates that you type in or randomly generated one. And then on top of that, we specify an encryption password and we're using this to actually generate our encrypted credential key store, and then encrypt any credentials that we hold. So anything that is stored with us is actually protected. And again, you can use an external key source for that is for HTTPS.

Harold Byun:

I'm going to skip that. So again, some things that were just given through in the interest of time here, but there's a lot to cover at pretty limited amount of time. So I'm going to get in, this is a brand new setup, I'm getting in and you can see it's pretty bare bones. I don't have access to a lot. We do have our back implemented. So the first thing you're going to do is that admin user is grant myself certain privileges. I'm just giving myself everything in this case. But obviously if you only wanted people to access the key store, which you don't see on the top bar today, or some of the other system administration tasks, or if you wanted to remove them from certain aspects, you could do that here log back out and get back in.

Harold Byun:

And you'll see that I now have the key store and other tabs available to me. So the next step that we're actually going to do is set up integration with a couple of key stores here, and you'll see that we've got the local credential key store that was part of the setup to keep any credentials secure. If I go back to my presentation, we are now in step two of one, two, three, four, which is connecting a couple of key stores. In this example, we're going to use Pollis or Jamal third key secure and AWS KMS in spark cloud HSM. Vormetric DSM one any of these other key management solutions.

Harold Byun:

We have a partnership going with HashiCorp. We support AWS secret store manager. It doesn't really matter to us at the end of the day. It's just a key. I'll come back to these notes in just a couple minutes and then just roll forward here. So the first thing I'm going to do is set up key secure. And so I'm going to talk to this key secure instance. I do have my cheat sheet here of IPS and I'm using... I can't remember all of these on the fly. And what we're going to do here is authenticate against this. We do use a keymap 1.1 client or higher to talk to your key management solution. [inaudible 00:13:02]

Harold Byun:

I'm using this JKS file for their certificate, and I'm going to hit enroll, it must've said password. There we go. So I've now enrolled the key secure password. And then while we're here, I'm also going to do AWS KMS. So roll that, we select our region. I'm operating on a US West two today. Deck storage can be an S3 bucket. It could be secret store manager. I'm just going to specify this, KMS test, baffled keys is just a random alias that we use to pretend any type of EK that we're managing here. And then I'm going to, again, populate credentials. We can always use an IAM role here as well, but just for particular example, I'm going to populate this and I'm going to enroll this AWS KMS connection as well. So while this is enrolling... Well, give it a couple more seconds.

Harold Byun:

Sometimes it takes a minute. So you see that we've got two key stores enrolled now, and the baffle credential store, this was step two. And our sequence of steps. We do support what we call two cure key hierarchy. So that is a CMK or a master key that is encrypting the data encryption keys at the decks. So for those of you, who've been doing encryption for quite a while and dealing with keys, so commonly referred to as MK or CMK or KEK or key encrypting key, same terminology, basically that is the parent governing scope of the master key. And that is used to encrypt the decks. The decks are never sitting out there in the clear or the data encryption keys, and the decks are actually the key material that we're using to perform the encrypt or decrypt function. And so we use all the industry standard protocols.

Harold Byun:

So KMS, PKCS, 11 protocols for HSM connections, rest APIs. We can apply the encryption material and both the column or record level. Again, that's that data centric mechanism. And so we're eliminating a lot of work that people would typically have to do to kind of wire up and figure out how to perform the key exchange, make sure you're not hard coding things, things like that, and really delivering this through this abstraction where there's just another view of the key management setup in terms of what we're doing here. So we have, again, support for virtually any key management solution, if you're using multiple flavors, if you're migrating from one to another if you've got different lines of businesses using different key management features, if you've got certain applications that are cloud scopes, certain ones that are on prem, there's a number of different scenarios where these types of mappings come into play and might be useful for your organization.

Harold Byun:

This is just another variant of this. This is out of scope. This is basically a multi key model that we use typically for multi-tenant SAS providers, but it could be any shared or co-mingled store or any data store that you're sharing with multiple third parties and you want to segment access. But this is kind of out of the scope of this, you know, is really multi-tenant or co-mingle data source. We had another webinar on this. If you're interested, it's not something that we're going to dive in too deeply called ensuring data privacy and SAS environments.

Harold Byun:

It's on our channel as well that covers that multi key, bring your own key service model. Now we're into our step three here. So the step is wiring up the database. So again, we're trying to orchestrate connecting the application to the key, to the database and make that all a move together in a unified fashion through this data protection layer. So I'll go back to my baffle manager, I'm going to go to the database tab and I'm going to enroll couple of databases. In this case, I'm going to do a SQL server on premises, Microsoft SQL server and on-prem IP.

Harold Byun:

And we can support an SSL, TLS termination if you're running that in your environment as well. And then this is going to be an RDS, MySQL environment, just to kind of show you some of the different flavors that we have. We do PostgreSQL, we can do Cassandra Mongo, Maria DB, Aurora, Oracle is in development. So expect that to be out in Q1 of next year and my SQL environment [inaudible 00:18:49]

Harold Byun:

So now I've got two databases, so that's kind of step three. And so what we're trying to do here as we go through this is basically approach this type of... I know this is marketecture, but effectively we're establishing environment where we have knowledge of where the database is sit, and we're going to create the substraction layer called baffle shield that sits below the application tier. And so we're basically invisible to the application tier, and that's how we move forward with these no code changes. And so works with microservices works very effectively with API. Any type of service mesh works with serverless functions like AWS Lambda. This is kind of a more granular view or block level view of what's going on. So the application for all intents and purposes is talking to the database tier, which is the baffle shield which I'm about to set up.

Harold Byun:

And then the application thinks it's talking to clear text columns A, B, C, and D. When in fact, it's really talking to encrypted columns, C and D in clear texting B and baffle shield maintains the privacy schema, which is the mapping that we're about to set up in step four. And so what we're doing here, is when I go into step four, I'm going to map this application and then we're going to encrypt the data. And so if I go back to baffle manager and go to the applications tab, I'm going to set up an encryption or a SQL server. And I have this brand new baffles shield that it's just an easy to host. It could be that a [inaudible 00:20:49] dockerized container. It could be wherever you... It could be bare metal, if you want it to on bare metal.

Harold Byun:

I need to upload a key here that I'm going to use for SSH. So in this particular [inaudible 00:21:06] this key and I'm going to use [inaudible 00:21:11], upload it, and then I hit next. And so I'm doing SQL server here. So I'm going to take the SQL server and I'm going to utilize a key source. In this case, I'm going to use key secure. And if you recall back to the first step we had enrolled this key, secure environment, and actually have this key secure environment right here. And you'll see that we have an initial key set of this master key and secondary key which was actually set up for... by baffle. So this is actually the deck, and then this is the master key. And so we set that up on enrollment. You can see that key two was added.

Harold Byun:

And so what I'm going to do, is I'm going to enumerate data schema, and I'm going to use this HB Mega store table, which has a set of columns. I'm going to get into this environment for you so you can see it before I do anything. And so in this environment, on the server, I have HB mega store and I'm going to select the data and you can see that this is a SQL server environment. It's whatever, potentially sensitive data in the clear it's obviously incoming data category, city, country, customer ID, customer name, order date, order ID, et cetera.

Harold Byun:

What I'm going to do here is I'm going to select certain columns that are of interest to me, customer name, product ID and then we have other key ID, what am going to do is I'm actually going to create a new deck on the fly. And so you see, I have a new key ID three, and I'm going to use key ID three for customer name. We also support deterministic and non-deterministic. And so I'm going to basically select those columns and hit next and then go. And so what's going on here? I know there's a couple other questions here, so support for Oracle yet. So it's coming in Q1 of 2020.

Harold Byun:

We already support PostgreSQL, MySQL Maria DB, or all the RDS waivers, Microsoft SQL as well, 2008 or two and higher. And so what's going on here is we are talking to in this case, this key secure implementation. And so you'll see that in this key secure, we have this master key set up and we're in the process of actually migrating the data for this application. And while this is going on, I'm also going to encrypt the MySQL environment. So I'll do MySQL and we'll go after the other, MySQL environment.

Harold Byun:

Again, just checking in, I already have that SSH queue. I'm going to connect in. Next, I'm going to select the RDS, MySQL and this case I'll use AWS KMS. What we're going to do is we're going to talk to this environment called Superstore. And Superstore it's the same dummy data set up, but if I go into the Superstore set up, which is here, and run the select, you can see, again, this data is in the clear. What I'm going to do here is I'll select city and maybe a product name and region. And we have a different set of key IDs. This has already been in use. So I'm going select a bunch from AWS KMS, hit next and hit go. And so these are in process. You'll see that what's happened here in the SQL server environment. It looks like we have completed at this point.

Harold Byun:

So if I go into the application encryption, you'll see that we have those columns that I had selected. And if I go back into the SQL server environment and hit refresh, we have encrypted city, customer name and product ID. And if I go through and connect through the baffle shield that I was using, which the way we connect and our invisible to the application is that it's effectively a connection string change or DNS host name change. What I'm going to do there, is just connect on this particular baffles shields, use this credential, and I'm now going through this baffles shield, you'll see the same schema.

Harold Byun:

And if I do the select on this particular dataset, and you can see that we are decrypting it, and that is how the application effectively is decrypting this data. Now, if I go into Microsoft Excel on my own machine, for example, I should also be able to just connect to the baffle shield and show you that we are effectively... Maybe later, we are effectively below that application level, below the ODBC interface, below the JDBC interface, there's no driver update. So if I go into get data and I go to... I think it's here. Yes, share. So I'm going to go and connecting there. We're connecting. Then I will see this, and I run this, clearly we can see that we are decrypting that data through that baffle shield, but it is actually encrypted at the data layer and at the column level within that backend database.

Harold Byun:

And so this is from my laptop going into the baffle shield sitting in AWS. And well, we're obviously kind of wired up a little bit different in this particular demo. But, just to give you the point of that in this particular example, we are below that application tier layer effectively invisible to it. So going back again to the MySQL environment. So this was running while I was actually... This looks like I might have a privilege issue. So we'll just take a quick look here. I told you that this would be interesting, so and it is in the clear, so I suspect it is a privilege issue, but just yeah, let's see, we'll give that a little more time. So while that's going, just to kind of walk you through what we're doing.

Harold Byun:

So we were in this key secure environment, and we now have that deck or key ID three, which maps to this data encryption key. And we also have the ability to rotate these keys. So I could role rotate the master. I can rotate the deck and I'll rotate this deck from three to four. And what you'll see is that, when I hit these keys, I now have moved keys that have been created. And if I go back into this key secure store, we'll see that I have a new key ID four which is ending A5D82, and there's no rotate button on key ID three. If I go back to key secure, I have A5D82, it is the new key. And then if I go back to my application, you will see that key three has been replaced with key four.

Harold Byun:

And if I go back into this application, and I was going through baffle shield, what I could do, is I could grab a bunch of queries, come back and insert, just do a new quiz here, I'm sorry. And I'm going to run this insert. So we just ran this through the battle shield. And if I go back to my original query in the clear, you'll see there's 235 records here. And if I refresh this [inaudible 00:31:01]. I must've inserted that in the wrong place. So I answered it into the wrong database. My apologies. I'll just do a couple, so you can see that [inaudible 00:31:29] for this, but I told you it might be interesting. So so I'll just run those two. [inaudible 00:31:48] My bad.

Harold Byun:

All right. So that's in there. And if I go back in, I should have gone up by two records. Let's see. It's 237 down in the lower right-hand corner. But the more important part I think is that, we're decrypting all the other data that was using key ID three, as well as the new data that has key ID four or the new deck. And so, that is one method where we can support multiple key versions across the data set environment. So those are some things that we can do as well. Let's come back to here and see if this actually went through with the permissions. It did not. So I don't have time to troubleshoot in the interest of this. I'm sure I have the privilege set up somewhere that is not being granted, but we've got plenty of customers that are running on the RDS or [inaudible 00:32:48] footprints as well.

Harold Byun:

So this is effectively the encryption orchestration that we just went through. So hopefully this gives you a good flavor of some of the capabilities there. One thing I also did not cover is a source to target migration which is moving from basically an on-premise to cloud offering, where we can encrypt it on the fly. There's another webinars as well on our chat, all that I think covers this in terms of securing your data in the cloud. Just in the interest of time as we go through the additional functionality, I'm going to cover a couple of things with the end and then hopefully we can also address any open questions. There's a couple that have come in right now as I go through this. Did the encryption process also changed the table structure?

Harold Byun:

Yes, it did. So what we do... It's a great question, to go over back over here. So what we do is, when we go into our source table and we'll see these columns, these columns are now varbinary. And so we did do a data type change, but we actually handle that for you. So we have a purpose-built migration utility.

You can use all your bulk migration, utility existing native tools if you like. Quite frankly, we think ours is better and more efficient. We have an online migration capability that can do things in a batch mode. We don't persist things or write them to a CSV file. So it's much more efficient in terms of character and coding handling, and the way we actually are migrating the data. But what we're doing during that process is we are changing the data type to a binary format for a number of reasons.

Harold Byun:

I mean, one encryption inherently generally wants to talk in a binary format. It's actually more expensive to convert it back into a ciphertext format. But we also always represent the original data schema to the application. And so that's how, when I was getting in through Excel, effectively this table does not know that... This application does not know that it's talking to a varbinary field. It thinks it's talking to a [inaudible 00:35:13] in this particular example. So we're always navigating that change for you. And that's what the function of the baffle shield was partially doing. And also kind of goes back to these earlier slides. I think it was this one. So it's effectively this, where we are taking that structure. This may open other questions. What about other DDL events? What about other alters updates? [inaudible 00:35:39] all the traffic's passing through us in the clear.

Harold Byun:

So it's passing through us in the clear wire speed. You can do all kinds of new table creates an alters all day long. It doesn't really matter to us. It's just going to pass right through 150 data. Obviously you're going to want to facilitate some type of data migration process around that. There's an additional question here. How does it typically work from a process standpoint when you want to encrypt data from staff? Someone will always have access to the baffled database to see data in the clear. depends on the architectural model. A lot of our customers are basically sealing off the application access, the SSL termination, mutual SSL authentication. We can support access control lists and then also black listing service accounts through the application tier. And that forces any privileged users straight to the backend or any attacker, more importantly, moving laterally in the network and they will only see encrypted data.

Harold Byun:

So not every user's always going to see data in the clear. A lot of them are going to see that encrypted view, which is basically this view. So your database admins can perform data maintenance or perform alters, developers can access a database, but they don't necessarily need to come through the front end. It does beg the question. What about users that do need to come through the front end? How can you control that? And so, that is a great question. And that's where the data masking piece, which is what I was going to cover next comes into play. When we go into an environment, actually, before I do this, if I go into this Superstore environment, and this is an RDS environment [inaudible 00:37:37] disconnect and reconnect, sorry, [inaudible 00:37:40]. I get in to this table. The second version is not supported to design, that's fine I don't want to. All right, there we go.

Harold Byun:

So you'll see that here we've encrypted one particular column. The rest of the, of everything is in the clear. And so if I connect in through this baffle shield, this was something that we just released earlier, in October, actually, if I go through this baffle shield, we get the same database structure, and we can also have the data encrypted on the backend and mask it on the front end. And so this is really much more of an end-to-end control model. So you'll see that in category, we masked with a string city, which was encrypted. We decided to knock decrypt and replace with access. We've generated random numerics for customer ID and a format, so good for third-party developers or application developers that need fake data. We can support timestamp and date. In this case, we did a fixed value. And then we also get a partial mass exposing the last four digits, again, without any code modification.

Harold Byun:

So another way that you can enable simplified data privacy, we can also do this based on threshold masking. How does indexing work? So this is a good question as well, especially on encrypted columns. So if it's indexing, then you can use a deterministic mode as well. So indexes on deterministic will work just the same as any plain text optimization. So there's no impact there. And then in terms of randomly encrypted data, you would want to rerun your indices after any data migration. And that that's true for database container encryption like TDE or any other data encryption solution. So we're no different in that regard. I think in many ways you need to look at this as an application level encryption equivalent, but without the code change requirements. Hopefully that answers that question there.

Harold Byun:

This was the data masking capability that I just showed you. There's other access control capabilities that we have. And this really starts taking us down the path of what we call dynamic data entitlements, which is how do we basically segment data and choose when we want to unmask it when we want to leave it master encrypted. And so, part of that is our record level encryption solution. I spoke to this a little bit earlier, again, there's another webinar on this as well, called data privacy and SAS or something to that effect. We basically took the same model that we have for column encryption, but we extended it and mapped it to a data owner ID. So now we have the ability to associate specific records with specific data owners. And this ultimately opens up the whole capability of data shredding or on-demand data masking with encrypted data underneath, which would fulfill a consumer right of revocation.

Harold Byun:

And for those of you who are keeping score on data privacy, the consumer right of revocation or data shredding, or the right to be forgotten or the right to be deleted, we could debate all day long the merits and the approaches around some of this, but this is effectively a mechanism that facilitates that. Let me give you a real quick look at what that looks like. And then we'll kind of close out with any last minute questions. Just kind of get into this environment.

Harold Byun:

What I'm going to do here is get into... This is actually an RDS and my SQL environment. And if I run this query, you can see that by going direct, the left-hand column is encrypted. Let's start there. I can't remember it. Now, if we go through the baffle shield, which is a connection string and I actually happened to be on that baffled shield right now. And I run the same query, we obviously decrypt the data as we would expect. But what we can do here is, if I simulate disabling or deleting the key, and I go back in and look at that data, you can see that we're using our masking capability to suppress the records. And that's because we don't have a key and we can't decrypt the data. So that effectively has an encrypted data value associated with a specific data owner, where there is no key present and we're facilitating a data shredding mechanism in that regard.

Harold Byun:

And so, there's a lot of that we're doing in terms of taking session IDs and mapping that to giving users and incorporating a user context to make this even more dynamic. This is one of our flagship customers, they were [inaudible 00:43:14] the financial reporting for the Fortune 500. So 75% of the Fortune 500 use them for SEC filings and 10Ks. And they use us to basically utilize an off the shelf, bring your own key service wired into their multi-tenant SAS environment. No large-scale architectural overhauls or application changes, no dedicated databases per tenant, which saves them money. Hopefully this was a, at least a good for some of the setup and how we're making things married together from an encryption and decryption standpoint. I'm happy to answer any other questions just while we're going and waiting for any additionals events and resources are coming with data privacy compliance webinar coming up a few weeks in November, we're going to be at reinvent in December, if you'd like to book a meeting with us info@baffle.io.

Harold Byun:

And then there's other resources on our website. There's a white paper on application level encryption simplified and privacy, preserving analytics. We have a report from Gartner on the privacy preserving analytics space. Do you support cloud HSM? Yes, we do support cloud HSM. So for those of you have a FIPs Level 3 requirements, we support other HSMs too. Again, it's just an industry standard protocol for PKCS 11. This is my contact info. Feel free to reach out if you have additional questions, things that we didn't cover and try to get you out of here and a half hour, I think it was a little bit aggressive but hopefully this was some useful information and happy to help you out with any additional information anytime. Thanks for your time. Bye.