Why VPNs and RSA Key ID Security Break Down When Authenticating Headless Devices; Interview with BlackRidge Technology CTO John Hayes, Part II

The world of machine-based data collection is creating an entirely new type of security problem: authenticating machines that have no formal user identity associated with them. Traditional VPNs break down in these environments while RSA Key IDs have no answer. Today BlackRidge Technology CTO John Hayes and I continue our discussion as we examine the limitations of these traditional approaches to security for headless devices and how its new BlackRidge Eclipse™ solution addresses them.

Ben: *How does Eclipse identify which machines have access and which do not?*

John: There are two main components to BlackRidge Eclipse. There is an Eclipse client which inserts identity and there is the Eclipse Gateway which is policy enforcement.

The client can be implemented in one of two ways. It can load a driver onto your laptop or desktop. Or, on machines that it does not yet support, legacy devices, and machines that, because of compliance or regulatory reasons you are not allowed to let software on it, it has what is called a client concentrator. Client concentrators insert identity directly on the wire on the behalf of the machines and devices connected to it.

Ben: *So with the Client concentrator you can attach identity to "headless" devices like industrial controls, correct?*

John: Absolutely. Machine to machine is very important to us and introduces another interesting concept.

When most people think of identity, they think of identity in terms of user identity. In industrial control and automation (machine to machine,) there is no concept of a "user."

You have a machine talking to another machine. So it really becomes machine identity and not user identity. We support both.

Ben: *I also want to be clear for folks that this is not a VPN derivative, correct? This is a completely new concept.*

John: You bring up a good point. We do get asked about VPNs. People say that VPNs have identity and that's true. But the identity only lasts as long as the VPN lasts.

So I might have an identity of the client, and you go through the VPN termination point, and once you exit the VPN termination point you no longer have that identity and are left with addresses again.

Using BlackRidge you insert the identity so even though you may go through the Eclipse policy enforcement point the identity is still there. I can actually have nested layers of policy enforcement points that are all able to determine the identity of the requester.

Ben: *Let's get a little more technical for a moment. If this is not a VPN derivative, how does Eclipse work?*

John: BlackRidge Eclipse is built using our patented Transport Access Control (TAC) technology. TAC operates technically by generating single use identity tokens and these tokens are inserted into TCP sessions starting with the very first packet. We are actually overlaying the TCP SYN and ACK fields.

The token is a cryptographic hash of keying materials associated with the identity and various clocking material. This allows us to be the policy enforcement point for a Eclipse gateway, pull out this identity token, resolve it back to an identity and then apply a policy.

One of the areas we are working on is scalability. We went from a very low number to 10,000 identities and we can scale up into the millions of identities.

But in order for that to happen, we needed to resolve statistical collisions within the namespace of the tokens. It is very possible that over time to get different identities that will show up and, for a period of time because the tokens are constantly changing, the tokens may end up being the same tokens.

We needed a way of resolving these tokens. Not only do we resolve the tokens, we have actually filed a number of patents in this area, and can resolve them at very high levels.

But as the number of identities increases, if you look at it from a statistical point of view, the probably of guessing should become easier to guess. So what we have done is actually have limits to say, *“OK, here’s your threshold.“*

We are going to make sure that the probability of somebody attacking this in a brute force fashion, regardless of the number of identities in here, is a probability threshold that they are going to have to meet. So that even though if I go from 1,000 to 10,000 to 100,000 identities, that changes the absolute probability within a given space.

This is a completely deterministic system and you can set thresholds. Right now the default threshold is at least a one in a million chance of guessing or higher.

What that means is that if you have a large number of identities in there, that threshold is not decreased. This is one of the areas we are extremely proud about and have been really working and the math folks have been having a lot of fun with as well.

Ben: *I can imagine! Another question I get asked is "How is this different than using an RSA key?"*

John: There are a lot of similarities and differences. At the highest level, what is implemented in the RSA key is essentially a one-time password that rotates every 30 seconds and requires synchronization between the RSA key that you have in your hand and some central server.

RSA keys are implemented as basically a password mechanism. You get a new key, the equivalent RSA key for us, is our token.

With BlackRidge, you end up with a key as frequently as you want. When you generate TCP sessions, there may be tens or hundreds or thousands of them every single second. And we continue to generate the new keys.

The other thing in the RSA case is the user generally has to go type it in, or cut and paste it, or do whatever you have to go do that. With BlackRidge, it's completely transparent. The user does not see it happening and you get the security without having to go through that.

The final thing is, the RSA keys as implemented today are all implemented and interpreted at the application layer. Meaning I have to establish the TCP session and then I am prompted for my user name and RSA key. This leaves open the possibility of compromising the application before or during authentication.

With BlackRidge, we intercept before you even establish the TCP session, so you do not expose the existence of your server or applications.

It also works very well for headless devices. Earlier you mentioned machine to machine. I do not know how to implement RSA keys in a machine to machine situation where you do not have the user.

In Part I of this executive interview series we examined the three practical use cases for network layer identification.

In Part III of this interview series I will discuss the management of Eclipse and how BlackRidge continues to find new use cases for this product.