Heather:

Welcome to the Hurricane Labs Podcast. I'm Heather, and today we're going to be talking about single-factor authentication and CISA adding it to their list of bad practices. Here to help me with that, I have Tom and Meredith. Thank you both for joining me. Meredith, if you want to go ahead and tell us a little bit about CISA and what this means.

Meredith:

Yeah, sure. So CISA is the Cybersecurity Infrastructure Agency, CISA, and they have this list of bad practices that they maintain in relation to cybersecurity, systems administration, and basically anything that would relate to the overall security of an organization or a network, or even just an individual device. And they basically maintain this list of bad practices as an opposite to best practices, to say, these are explicitly things you should not be doing, please don't have these in your environment, this will open up significantly more attack surfaces than not having them.

Heather:

Why don't you tell us a little bit about what CISA declared and explain why they added single-factor authentication to their list of bad practices?

Meredith:

So essentially what CISA went ahead and declared was that use of single-factor authentication anywhere across the network work, but they did specifically call out systems that were remotely administrative and added another layer of if those remotely administrative systems had remotely administrative tools. So things like controlling your domain, your users, whatever, if you had an email server, anything critical in your network, you should not be using single-factor to log into those. That comes from two different levels of a security concern. The first is, the thing that you're going to hear 40 million times in the industry, which is that users are your weakest link. If you are just using a single source of truth and a single-factor for logging in such as just a username and password, that can be stolen from a variety of techniques, whether that's attacking the user because they're the weakest, doing phishing, social engineering, whatever it may be versus using an insecure authentication method. That is why they said, okay, because that makes this essentially a single point of failure where if one person's password gets compromised, that entire thing is compromised. That is why they added that to their bad practices list versus a multi-factor authentication where if your password gets compromised, but you still need to type in something additional or provide additional information to verify you are who you are. It is infinitely harder for people on the outside to get that information and do that.

Tom:

And I think with users and passwords, you pretty much have to assume that passwords in your organization, if someone wants to compromise it they'll find a user that they can do. Now, granted, you can mitigate some of that by limiting who has administrative access, which you should be through defense in depth and all that, but even then the bar to entry for getting a user's password is lower than we'd like to hope.

Heather:

So, where can people go as far as if they want to get some recommendations and stuff like that, how can people find out what some of the better practices are for security?

Meredith:

So on the main CISA website, they basically have a link to their running list of bad practices, which also links to GitHub. And on GitHub, you'll see those list of bad practices and people from all over commenting their thoughts on why it is a bad practice, why it shouldn't be a bad practice or posing a new potential bad practice to add to the list. And there are just hundreds and thousands of people out there who spend all day debating these things. And it's good to see insight from both sides when you're debating whether or not something is a bad practice so it's a very great resource.

Tom:

I think the ability to collaborate and get feedback from different people in the industry, and especially if you're trying to implement something in an environment that maybe doesn't seem to have an obvious solution, there's almost definitely someone else in the industry that has gone through that and hopefully they'd be willing to share some experience and advice on that in order to help us all bring the bar up.

Heather:

We've been using just passwords to log in for a really long time. So, why now? Why is single-factor authentication considered such a bad thing?

Meredith:

The longer that everybody has a presence online, they get comfortable in doing the same thing over and over again. So they're much more likely to reuse the same password or not go ahead and do, yeah, complacency, not do their due diligence in securing something and attempting to basically protect themselves online, which makes them a target for phishing campaigns. And since people have gotten complacent and comfortable in what they already know, they're more likely to just open up an email and click a link, not even do their cyber security training that they've been given by their organization and just go click the link anyway, and try to sign in because it's 9AM and you haven't had your coffee yet so you may as well try to sign in to this random link and suddenly your credentials are gone. And I saw it time and time again when I used to work for help desks and I'd get a response of, oh, I just wasn't thinking. And it's not uncommon to see a person do it more than once. So the people who have gotten complacent and are also vulnerable to giving up their credentials are vulnerable to giving them up multiple times.

Tom:

Well, I think you're actually bringing up a good point there, Meredith, too, of where we could illustrate how two-factor comes in handy. Let's just say, for example, that Joe user, let's say that they have a password that is used for multiple things. So let's just say they use the same account for their email, some random company's product or service, and then their bank account. So they're using the same password for all three. Let's just assume that the email isn't the thing that gets compromised, but the random service is storing passwords in plain text in a database that's written into an S3 bucket that's worldly accessible, because no one would ever do something like that.

Meredith:

Yeah, no, that's never happened.

Tom:

Yeah. So we'll just assume that that's what happens. And you can literally just go to a page on the internet and see a list of everyone's username and password. And I'm not even sure that that would be, the password calling it getting hacked because the company is just dumb, but I'm sure that has happened somewhere in the history of the internet. But let's say without two-factor authentication, you have the email address, you have the password, you can just log probably right into that email account using that because the username for your email is almost certainly your email address, very high likelihood it's the case for the bank account as well, but once you get into an email account, you can really compromise a whole bunch of different things. And that's where shared passwords are bad because you have the password that someone gets through a breach, an exposed service, something like that, it's out there and it's easy enough for someone to try other services especially an email account. But if you add two-factor to that email, you've now created at least something else that the bad guy has to hop over in order to get access to that email account, whether that's a six digit pin you have to get from Google Authenticate or a text message, something like that, or a push from some two-factor app, it at least requires the user to do something else. Now the user could still do that and you could socially engineer it, but you significantly raise the bar for it being harder to get into an account.

Meredith:

I would agree. And even just adding, as you said, that two-factor just for your email address, since that is likely the first place that's something an attacker will pivot, that makes it so much harder to get keys to the rest of the kingdom that you have, because anything in your email, you're going to get your usernames for companies that choose to provide another bad security practice and send you your password in plain text. Those are of course all in your email as well.

Tom:

And even if they're not sending your password in plain text, you can use emails to get password resets. And that's a great way to compromise accounts once you get into the email account.

Meredith:

Yeah. So email's, essentially, your golden ticket there.

Tom:

And likewise with some two-factor stuff being SMS or text, having not a single-factor to say, swap a SIM or something like that in a cell phone, or redirect a number to someone else, that can pose equally significant issues too.

Meredith:

Yeah, I would certainly agree. And I do think that while some of that has fallen out of favor, it's still something that needs to be addressed and thought about whenever people are implementing multi-factor in places, because of the fact that that is another vulnerability that is addressed when you are using that particular form of multi-factor, which I think some may. That is why some places may choose to opt out of using the SMS multi-factor authentic.

Tom:

SMS is probably the lowest barrier to entry.

Meredith:

Either that or the automated phone calls that are of similar nature.

Tom:

Yeah. I mean, the good thing about that is, if you have a user base that doesn't necessarily have smartphones or apps that can still be a good method, although it still requires some degree of user education and all of that. And you could also say, sure, most people have cellphones, but I'm sure there could be people with landlines that want to use the voice authentication and you could create a situation where they simply can't authenticate if they're not physically at the place where that landline is. Probably not that big of a deal these days, but I could still see that being a potential case for some users as well, that might lead to pushback.

Meredith:

Yes. And I will say that some of those phone calls that you receive, they do them in one of two ways, either they send you the code just like an SMS or they ask you to press one for yes and two for no if you're authenticating. And for a lot of people, and I've noticed the same with one of the other multi-factor authentication methods that you mentioned earlier, which are the pushes from the apps or typing in codes from the apps, people are more likely to just hit yes, and think that, oh, something is trying to re-authenticate, I don't want whatever session I've got to expire. So they're more likely to just answer yes, and not think about whether or not this was somebody else.

Tom:

Yeah. So especially providing some information to the user about why they're being asked to authenticate is useful, but entering a code that's stated into a thing you're trying to log in like Google, for example, if they call you and they give you a four digit or six digit code or something, you have to actively put that into something or tell it to whoever's trying to crack into your account, which is perfectly possible too, but it involves more doing something than the alternative of actually just saying, press one.

Meredith:

And I will say that I like that route that's implemented significantly more, also any companies that are willing to provide context, especially saying the authentication was requested from Jamaica, but you're up in Maine, probably not you.

Tom:

I've actually also seen phone calls where you actually had to enter a pin in order to confirm the fact that you're logging in. Does that mean it's three factor authentication or because you know the password and know the pin, it's still two-factor?

Meredith:

It's still something you know and something you have. I think it would only become three factor authentication once you add in the something you are.

Tom:

So I guess if you use a key, be something you have something know, and then the factor, the phone that would be three factor maybe, I don't know.

Meredith:

Yeah, I guess that would work as well.

Tom:

I always joke that, oh, two-factor authentication, your username and your password. It's something you know, and something also you know. And no, that's not how it works, but...

Meredith:

I've heard a significant number of arguments for that though lately, where all you should technically need to authenticate is a username and no password, because as long as your username is secure and nobody can guess your username, why would you need a password?

Tom:

So, interesting thought about that, I am fairly convinced that you could put a Linux system on the internet, make the password the word root and make the username what your password would be and no one would ever log in. Now, granted, if there were actually people trying to do this and at some point your username would be figured out so it's not a perfect situation, but just looking at what things are scanning the internet, they're looking for root and brute forcing passwords, not trying the other way around. So in the current state of affairs, I can see that being a case where the username could be essentially a password, but you can't assume that that's private information because of... But just say your password to your email address was your email address. That's just stupid.

Meredith:

I see your point. And I would like to test this out.

Tom:

The thing about the user name?

Meredith:

Yes.

Tom:

So, the reason I say that is I actually put a honeypot on the internet, modified PAM so I could see what passwords were being used and put it all into Splunk. So in three months of this thing being on the internet, there were probably less than a couple hundred usernames that were tried and millions upon millions of passwords. Also, that's just an example of, if you do have something that you have to have publicly accessible on the internet, not using a default username can go a huge way into securing that. You still should use two-factor, but if you don't use the default administrator usernames, you're kind of adding another factor to that by making it something else that someone has to figure out. It's still not hard to figure that out so don't consider it that big of an increase in security, but having an administrator name of something that's tailored to the administrator versus admin or root is a step in the right direction for having a secret account that's more secure.

Meredith:

I would agree. And if you have a system that needs to have that default administrator account functioning for some reason, change the password, make it something that even you do don't know, as long as it's secure. Mash your keyboard for 30 seconds, that should be all you need to secure that account. Well, and turn them off the system, but I'll let that be.

Tom:

Yeah. Hard coded assumptions in programs can be annoying, but probably the same things that are the administrative tools that we're talking about for this. I guess it's also worth talking a little bit about some of the other two-factor methods that exist, so apps that exist using tokens that we've seen. I think the apps are what are taking industry by storm now. And you don't see a lot of things like RSA tokens like you used to see 15 years ago.

Meredith:

Yeah, I would agree. And I mean, even RSA's hard physical tokens that they sell, they have in app form and they've basically recreated the exact same thing and it does a very good job. A lot of these services like Duo and Google Authenticator and even the RSA one, they allow for external services that don't, by default, use their authentication for multi-factor to be added so that you can have a single, centralized point where all of your multi-factor is managed or a close to single place.

Tom:

That is true. And I think it's worth bringing up that there are ways to accomplish multi-factor authentication without having to necessarily have your product explicitly supported, because I think that's going to be a big sticking point for a lot of people thinking that this is a challenge. If you have to change how your application authenticates to add a second factor step, that's probably going to cause a lot of pushback, but if you can set up the authentication to use an LDAP proxy, for example, that goes through a two-factor mechanism and basically requires a push before it returns to successful authentication, you essentially don't have to do anything in your product other than point it to the proxy at that point and authentication will work with two-factor without the product actually supporting two-factor, which is really kind of cool. But you think about the kinds of systems where two-factor is probably most important. So just throw out an example, the control panel for a nuclear power plant probably doesn't have built-in Duo support, but it probably supports LDAP. It's probably not secure LDAP, it's probably 389 TCP in the clear, but even then, if you have a product that does that and you put a two-factor proxy in there, you basically just have to assume the passwords are compromised at that point because anyone on the network can just read them. But at least the two-factor provides some additional level of verification. Other things I think where that would come in handy are any of your out of band management for stuff, so physical equipment. Basically we go back to the IoT side of things, but you're out of band management, like your Dell iDRAC or your HP iLO, someone getting access to that basically has physical access to your system anyway. So, using a mechanism to have two-factor before they're allowed to do that, and also preventing anyone other than administrators from even touching that network, that's another thing, but you basically want to avoid the ability to have someone remotely basically have the ability to sit on a system in your data center. Likewise, things like UPSs that have the ability to potentially manage power remotely, access control systems like doors, the control panels for your fire alarm, those sorts of things, all things that probably don't support two-factor by default, but having some mechanism where the two-factor can sit between whatever native authentication they support and your two-factor of choice is a good way to handle those sorts of things.

Meredith:

I would agree. And for any of those things that you've just mentioned where your end goal, essentially, be able to access them from anywhere for a case of disaster recovery or ease of access with everybody working from home, things like your iLOs and your iDRACs, they would usually live out on the internet so make sure that they stick behind a VPN and that VPN has two-factor or the authentication you're using has some sort of verified, secure channel.

Tom:

Wait, did I just hear you say iLO and iDRAC usually lives out on the internet?

Meredith:

For a lot of places, yeah.

Tom:

Why? That's the most horrifying thing I've heard this month.

Meredith:

Right. But if your goal is be able to bring up a system when you're sysadmin is in Barbados, I don't know why I really want to go to the Caribbean today, but apparently I do, but if your sysadmin and Barbados and your server goes down and nobody else on site has the ability, sysadmin in Barbados needs to be able to bring that system up.

Tom:

Yeah, but that's a whole separate discussion. We could have a whole podcast about why you shouldn't have your out-of-band management publicly accessible. But I get how that's for things like in the power grid and stuff like that, because the equipment and the operation of it is more important than security, but even then, still, put a VPN in front of it, please.

Meredith:

Right. Put a VPN in front of it, defense in depth people, it is your friend.

Tom:

Yes. There was a vulnerability in iLO five years ago or something where anyone could authenticate with a password of 32 A's.

Meredith:

I'm sorry.

Tom:

That was just how it worked.

Meredith:

Buffer overflow the password field?

Tom:

Something like that. But yeah, if you tried to authenticate with 32 or 31, the letter A, it would just log you in as a full admin. So I know we're talking about passwords, but I don't even think two-factor would help with that. So keep random people from logging into those things first, then implement two-factor, but do both.

Meredith:

Do both, then two-factor your two-factor so you have two-factor on your two-factor.

Tom:

That might be going a little crazy.

Meredith:

Is that not what we do here?

Tom:

It happens sometimes. I think it is also worth talking about where two-factor is helpful and some ways that it can be still exploited. And I think we touched on this a little bit, but I think the first thing you're talking about, users assuming that they're authenticating when they're not, I think that's a really good thing to bring up, especially for something like an app where like Okta, Duo, those sorts of things, where it gives you a prompt. What are your thoughts on that, Meredith?

Meredith:

As somebody who has done some penetration testing, one of my favorite things to do with places where I know they have multi-factor authentication is for all the passwords I compromise, try to log in and see how many of their users will actually respond to their Duo, Okta, whatever multi-factor authentication push there is, see if they'll respond to it. I may be in the same general location, I may be states or countries away, but it's interesting to see if people are genuinely paying attention to those prompts and the details that they provide in them, such as the IP address and where you're coming from.

Tom:

Now, you know for Duo, they're just going to accidentally block you anyway, because they're used to the button being on the wrong side.

Meredith:

Yes. Yes.

Tom:

Well, I mean that is also an interesting thought where how users are trained to do a certain thing though, too. And I noticed that myself, you expect it to be in a certain spot, you hit a thing, you don't even read what shows up on the screen and then the intent of your action is not what actually happens. But I guess maybe for those that haven't had to deal with that sort of thing, it might be worth mentioning what we're talking about too. So, Duo basically released a new version of their app within the past month or so, where for the past, what? Six, seven years or something like that, the option to say, yes, I'm approving this authentication was on the left with a green check mark and the deny was on the right and then they introduced new version of the app that flopped those where the green went on the right and the red was on the left. And the first time I saw this, I thought I went insane. Muscle memory is a very strong thing. I think that also had some interesting consequences of a SOC side of things too, right?

Meredith:

That certainly did. We saw a significant uptick in false fraudulent reports from people who had inadvertently clicked on the left side, thinking, yeah, no, good to go, but not actually paying attention to what was there and as you said, just using their muscle memory.

Tom:

And also an interesting alerting use case scenario too, where assuming that that wasn't a case, a denied two-factor is significantly different, I think, than a failed just a password login from a alerting perspective.

Meredith:

Correct. A denied two-factor to us alerts us that the user is aware of something unexpected going on in their account. And we do have multiple points of monitoring for that so that if a user does mark something as either fraudulent or says, no, this wasn't me, with Duo specifically, you have the option to say either it was an accident or I believe this is fraudulent. And if we receive an alert of potential fraudulent authentication, that gives us the insight to know, okay, we should be looking at this user, making sure that this IP address was not malicious, this was malicious and reach out to their security team to have their password reset and bring it back down so that whoever's on the other end has no factors of authentication.

Tom:

Yeah, because if a user's denying multiple authentication pushes that they know aren't caused by them, that means someone has your password and they're actively trying to use it. I hate that we have to go back to the, there is user awareness training and we have to rely on users to do the right thing, but essentially you're creating another layer of complexity and something else that a bad actor has to get through, but fundamentally you're still relying on the user to do the right thing. And I kind of hate that being a thing, because like you alluded to earlier, Meredith, users aren't going to do the right things all the time. But I think it's easy as security professionals to always say like, oh, it's the user's fault, they're causing all these problems, but we need to exist in a world where we trust the users to do the right thing, hope the users will do the right thing, but provide the controls to mitigate the damage if something happens.

Meredith:

Right. And this is why it's important to have multiple ways of monitoring around your authentications so that in the event that the user's awareness of what's going on fails, we still know, hey, the user doesn't expect to or we don't expect the user to log in from Michigan when they're in Florida, that is immediately seen as suspicious to us and that's something we need to go check out. So, as you said, trusting the user is great, but goes back to one of my favorite quotes as a new red teamer is the trust, but verify.

Tom:

Are there scenarios where it's not practical to have two-factor authentication?

Meredith:

I'd say, yes. The first thing that comes to my mind is a true air gap network. If you are truly segmented from the rest of the world and you are operating in your own little bubble and every authentication, every account on there is known and trusted and there will never be transmissions with a third party and the only way that you can access those systems is physically, I'd say that multi-factor authentication isn't necessarily needed. I know that there are some government regulations around that, but in terms of practicality, it's not always going to be the most practical there.

Tom:

Yeah. I kind of feel like Stuxnet would disagree with you on that.

Meredith:

Okay, but that was a third party that was introduced to that.

Tom:

Sure, but it's still a case of basically an attack against an air gap sort of thing.

Meredith:

This is true.

Tom:

I think that that might be a good case for something like a physical RSA token or something like that, where that doesn't necessarily require anything other than clocks to be synchronized correctly. And you don't have to carry one or have an outside connection, you can still have all of your technicians that need to go in there has a heavy physical token. That all said, air gapped, if there is actually some security guard type thing and you're not having them plug random USB things into your centrifuges.

Meredith:

Oh, I'm sorry, is that bad?

Tom:

Yeah, that's bad. But yes, you could have a single-factor password on the server in your bunker, but the security authentication method for the bunker, assuming that's using something that isn't the same password, is technically two-factor authentication.

Meredith:

This is true. And with that in mind, a lot of air gaps are physically segmented off in more ways than one to where you'll have to tap in with a card and a username and password or a card and a signature and a photo. So, I guess it still is multi-factor in the end, it's just not on the end device itself.

Tom:

Yeah. It's slightly different multi-factor than what we think as multi-factor, but you are still doing something to validate who the person is before they log in. That all said, I think there are probably some cases where it is a risk, not even risk reward, but you kind of have to think of the impact of the system versus what challenges you introduce for implementing multi-factor. And I think there probably are going to have to be some accounts probably that exist that don't necessarily have multi-factor as a break glass type of situation too, especially for things like critical infrastructure and life safety type stuff where you almost have to weigh, not almost, you have to weigh human life higher than the security of the environment just because of what's protecting.

Meredith:

Correct. But you also have to weigh the security of the system in relation to human life, which is why there will never be a perfect solution.

Tom:

Yeah. But still, your normal operation should still be two-factoring, you should monitor the hell out of those accounts that are not.

Meredith:

Actually, from my perspective, you should just monitor all the things.

Tom:

Yes. But a normal user logging in is a different scenario than your this should never be used ever account.

Meredith:

Correct.

Heather:

Circling back a little bit to CISA and their bad practices list. What role does CISA's list play in the overall security world?

Meredith:

As of right now, the list is merely a strong advisement. I don't believe that there are any compliancies out here that require you adhere to that simply because CISA's bad practices list is so new, but that being said, anything that lands itself on there, as soon as that becomes worked into your standard security policies, I would expect to see that those things come out as either explicitly banned or band with some special caveats as Tom and I have mentioned through this.

Tom:

And I think there's tons of standards, tons of sources that have advice and all that, but I think driving discuss about this and pushing more organizations to improve the security of their platform, it's all going to make things better across the industry.

Heather:

Yeah, definitely. All right. Well, that's it for single-factor authentication. Be sure to catch us next time where Tom and Meredith are joined by Roxy to talk about hardening your wireless security. Until next time. Stay safe.