Authentication, Rainbow Tables, and Password Managers

[00:00:00]

[00:00:00] **G Mark Hardy:** Hello and welcome to another episode of CISO Tradecraft, the podcast that provides you with the information, knowledge, and wisdom to be a more effective cybersecurity leader. My name is G Mark Hardy. I'm your host today, and we're going to be talking about password managers and passwords in general, and then how do we maintain all those secrets and do it in a way that doesn't involve mass compromise. So before we get going, let me give you a quick word from our sponsor.

Being able to clearly articulate your vision for your security program to the board and other executives within your firm is critical to obtaining the buy in you need for your program's success. Risk3Sixty has created a presentation template that helps you structure your thoughts while telling a compelling story about where you want your security program to go. Download it today for free at risk3sixty.com/resources. [00:01:00] That's risk3sixty.com/resources.

Okay, so let's get on with our show. I just got back from DEFCON. If you remember the last show, I had a little bit of different background out there. And I've been going to DEFCON for quite a while.

And, I was privileged to become a Goon for Life. There's my 25 year badge here. This is my 28th DEFCON. And as a result, I've been able to hopefully add some value over the years and be part of a really interesting community. For those of you who have never been to DEFCON, You might've heard of its professional cousin, BlackHat, went to that as well.

And that's the week before. But if you've never been there, it's worth a listen. A lot of great talks, a lot of information comes out. And if you sign up for the Sound of Knowledge broadcast, that will give you everything that you can copy yourself, or you can download it or eventually watch it online, but.

Anyway, that was what we affectionately referred to as Hacker Summer Camp. And [00:02:00] well, now let's face it, we got to get back to work. So, password management solutions. have you ever examined what password management solution your enterprise is using? Because I'm doing that right now. We've got a contract that's coming up.

We had a three year, vendor contract. And from time to time, we look at things and say, are we getting the best value? Are there any unnecessary risks that we are assuming through this contract? And there's a lot of other questions that I'll share with you a little bit later about how you can look at things like that.

But, a lot of times we find out that sometimes we inherit a solution from a prior decision. Whomever was CISO before you, or somebody above you, or a parent corporation said we will use product X. And okay, you find use product X and it works fine and there's nothing necessarily wrong with it, but is it optimal for what you want to do?

And so the other alternative other than that is we come into a new job and we say, well, I want to use this product. Why? Well, I'm comfortable with it. I know an awful lot of people who don't like getting out of their comfort zone. And so once they've learned something, they will go kicking and screaming if they have to learn something [00:03:00] new.

You got to remember that in your cyber security world, you got to remember G Mark's Law. Half of what you know about security will be obsolete in 18 months. So if you're not dedicated to lifelong learning, this is not the career for you. But if you love it, this is a chance to expand your knowledge and learn a little bit more.

The other thought is, is a third approach might be to do a zero based decision. Now that's some budgetary term. A zero based budget means you don't take a look at anything you ever spent money on before. You don't say just because we had this we got to have it again. Zero based allows you to either, from a staffing perspective, or a budgeting perspective, build up your business case from the bottom.

I remember we did that in the Navy reserve back in 2005, did a zero based review. And as a result of either finding that there was a little bit of excess weight in there, or people didn't respond, the Navy reserve cut 16, 400 billets. Now that's a lot of people. And ironically, I was, I think the only organization that pushed back, I gained a hundred and 71, [00:04:00] because that's when I stood up the Center for Naval Leadership and we were the wrong way goldfish going the other way, putting in a reclama and picking up the floor sweepings, if you will, from it. So just because your organization is facing huge cuts, and this is the reason I bring this up many years later, is that if you go into a recession, you're going to slow down and you find out that there's going to be cuts across the board, you don't necessarily have to be cut.

You can take a leadership role, build a compelling business case. and explain why you need to continue this function. In cybersecurity, I'm thinking that's a little bit easier to do because of the importance of what we're doing. But if you can't articulate that business decision in a way that executives who control the purse strings can understand it, you may find yourself enjoying the same cuts across the board.

Okay, so that what I mean by zero based decision, hence the kind of little detraction on that, but think about it from the perspective of, if you had to start from a blank sheet of paper, what would you do? Now, what I'm not going to talk about today is privileged access management, [00:05:00] enterprise type solutions like CyberArk or BeyondTrust or companies like that.

That's probably going to be a different episode, but I want to focus on what we call password management for your end users. Now, the average user has around 80 passwords to remember. Now there may be more and chances are you have a lot more. I think I have well north of 500. It's a lot of things that I have to keep track of.

And as a result, you need a way to manage all that because unless you use the same password everywhere, and I've known some people who do that, it's a little bit tough to keep them all straight. NordPass did a study and they found out that the top passwords in this big set of passwords that they use, head plain text, were, well, no surprise, password, the first one.

What do you think was second? 1, 2, 3, 4, 5, 6. What do you think was third? We'll add some complexity. 1, 2, 3, 4, 5, 6, 7, 8, 9 And then Guest and Wow, here's a tough one, [00:06:00] QWERTY. Oddly enough, the strongest of the top 200 passwords in their list was a phone number in Kansas, 913 666 8099. Now, I never tried dialing that, and I'm not recommending you do.

I don't want this person to be bothered. Like when the song came out, 867-5309, and if everybody had that number, they probably hated it for a while because everybody was just dialing it, probably drunk, trying to figure out what was on the other end. But the advantage of having a 10 character password when the attackers don't know if it's letters or numbers or special characters, it would take about four days to brute force that.

A lot of these other passwords could drop in seconds because, first of all, they're the obvious ones that you try. So if you find that you're using obvious passwords, bad idea. Better yet would be to use super strong, ultimately

complicated, Wall to wall passwords where it goes all the way out to the maximum complexity and the maximum number of characters.

But who's gonna do that? Well, [00:07:00] I'll show you the ways that you might be able to get close to that. So let's kind of go back to basics as I like to do when I'm teaching a subject and we'll talk about authentication because that's really what passwords are doing. They're authenticating. Now if you google Authentication in the definition, you will get as many different definitions as you get entries in Google, and there's a lot of them.

So there's really no standard definition, so I figured, well, let me go grab what seems authoritative, and that would be the Merriam Webster Dictionary. So authentication, noun. Enact, process, or method of showing something, such as an identity, to be real, true, or genuine. Okay, so authentication, showing something like your identity to be real, true, or genuine.

All right, we'll go best out of three on that. But that brings up the question of identity. Okay, if I'm proving something is real or genuine, what is it I'm trying to prove? Well, an identity, to go back to the dictionary again, is a condition of being the same with something described or asserted. So if [00:08:00] I say I'm G Mark Hardy, that's my identity, but I'm asserting that, I haven't proven that yet.

It's kind of what the authentication is doing, is it's proving it. And so if I'm asserting something, what does that mean? And one last definition for the day, an assertion is a declaration that something is the case. So, I can go up to somebody in a business meeting, Hi, I'm G Mark Hardy, shake your hand, and we're good and going.

But I can't really go up to TSA and say, Hi, I'm G Mark Hardy, I have a ticket for that plane, let me get on it. They'll say, well, very nice sir, but you need to authenticate, you need to prove who you are. And typically what we prove who we are, That is to do that authentication of our identity in one of a number of what we call factors.

Now there's four types of factors. We used to have three, but we got a fourth one. Think about it if you haven't thought about that already. Something you know. That's what we're talking about today. Passwords. QWERTY123456 Whatever it is, I happen to know the password. And as a result, I prove to a certain level of assurance [00:09:00] that I am the entity that claims to be who I am.

There's also something you have. Probably the most common thing that you have, that you're kind of used to, is what? Keyring, right? And so when you have a key, a key is something you have, and you can't just stand in front of the door and say, open Sesame. Uh, well that's something you know, but you'd have to have the key to unlock it typically.

Start your car, although I guess Teslas are a little bit different thing, and other smart cars are going in that direction. Something you are, which is the biometrics. where we started out with something like a fingerprint, an iris scan, and hand geometry, and there's a lot of other things. Now they're looking at a lot of other patterns that could determine who you are.

A quick story, so back in, well, gonna be a long time ago, 1985, the definitive security show was not RSA, it was not Black Hat. It was the CSI conference and I think that was the 8th year, maybe 7th or 8th year of the CSI conference in Chicago. And I remember going there and my project, because I was working on my [00:10:00] graduate degree at George Washington University and Professor Lance Hoffman was asked to do a project.

And my project was I wanted to do an inventory of all the computer security projects, because we didn't call it cyber back then, that were out. In the market at the time and so I brought back all the literature wrote it all up and kind of categorized it Okay, so probably it was an early version of Gartner, but one of the things they had was a fingerprint scanner Now it's not like the things on your laptop today We just scan it was actually like a little device a box.

It was kind of like a mouse travel He had to put your finger in there and then it would do whatever it was going to do and scan it now I got a sense of humor and sometimes it's inappropriate But one lady was looking at this thing and the salesman wasn't around said you want to know how it works. He goes.

Yeah I said, well, it's a fingerprint scanner. It proves who you are. You just take your finger and put a little hole, go ahead. And if your identity is correct, it gives you access. And if your identity is wrong, it chops the end of your finger off. She's like, ah, and ran away. Okay. So that was probably one sale that I cost that company.

And hopefully I didn't give that person nightmares for the rest of her life. But yeah, I don't let people [00:11:00] like me close to your booth. If I'm, I don't do that anymore, but when I was younger, it was kind of fun. But what was the point? The idea that something you are was a novelty back then. Almost 40 years ago, but today it's pretty common.

We have that facial recognition, hold up your cell phone. By the way, I don't turn that on. And the reason being, I don't want to turn that on. It's that if somebody grabs you and they want to assault you or something like that and get into your phone, all they have to do is like Louis, hold his arms back and then take the iPhone and hold it to your face and you go.

So I insist on using at the minimum sequence of. Numbers, passphrase or whatever, but I don't like the biometrics personally. And that's just because you could be forced into unlocking your device without your consent if somebody's bigger than you are. And what's the fourth one I was talking about?

Something you know, something you have, something you are. How about someplace you are? GPS, right? Your phone will do that. My bank will do that. It will want me to have GPS turned [00:12:00] on. So if I'm trying to authenticate from a location, they happen to know it was my phone. They go, okay, fine. Probably low risk.

If I'm suddenly in Moldova or some other part of the world where I don't normally travel, then that would probably trigger additional checks to validate that I am in fact, really there. Someplace I am could also be. In addition to GPS, if you own an IP address, we often find that if you're from a known IP address, then we can call that reasonably safe.

I do that in terms of my policies for multi factor authentication. If our users are in the office, then that particular IP address, publicly facing by the way, of course inside they're going to have an RFC 1918 address that doesn't route out to the outside world, but publicly facing address, that's the one that Microsoft sees, if they're there, they don't have to MFA.

Why? Because we've got a secure building with secure authentication. It's two factor authentication. Get in there and then you gotta log into your system and then you get out of the Wi Fi with a secure. And if you get all that far, pretty [00:13:00] high probability you are who you say you are. And at that point in time, I'm not going to challenge my users with having to go ahead and MFA.

So something to think about. If you know that you have a secure location, some place you are, like an IP address, you might want to consider turning off MFA to make it more convenient for your users. And of course, if you think that there's a real risk with that, Send me a note and I'll take a look at it.

Now, the reason you do multiple authentication factors is you get orders of magnitude more assurance with two than one. Credit card. Well, there's three

factors on it. A number, all right, a signature, and the physical card itself. If I go to a gas station, all I need is a physical card. Nobody's checking my face, and nobody's verifying my signature, and if I didn't memorize the numbers, it doesn't matter.

If I go to Amazon, I need to know the numbers, but I don't need the physical card. And then lastly, nobody seems to check for your signature anymore, except for the postal service. That's part of their processes, check the back. So when you get multiple authentication factors, and the military will do that.

When I was ata [00:14:00] combatant command, we had to do three factor authentication to get into the command center, had to have the badge. Had to have a PIN and had to do a biometric. And if any of those three things failed, you didn't get in. And once you were in there, then of course, everybody should know who you are anyway.

And so, we do a pretty serious job of making sure that authentication is done correctly. Now, the problem with doing any type of a authentication mechanism is a false positive or a false negative. False positive is what? You misidentify a bad actor. Bad guy, come on in. False positive. Off you go. And then they start causing trouble.

So we say, well, we don't want that. Well, the problem is when you tighten down on that, you end up kind of like the whack a mole with a false negative, where you misidentify a good actor. Sorry, you blocked the CEO. The CEO wasn't able to log in because the CEO was on a business call in a different place from an airport.

You didn't allow that, and we just had the deal fall through. Now, there's a huge difference in the world. In the credit card world, there's about 16 billion dollars in false positive credit card fraud. [00:15:00] But for the denials that take place, there's almost 113 billion, at least last time I looked. A false positives, which is what?

Your credit card was declined, sir. You are no good here. Well, if you're taking somebody on a business trip, that's pretty embarrassing, and that card's going to the back of your wallet. And so, you can't have this either way because you've got to balance it because a false negative is essentially a self imposed denial of service.

If you say no to one of your people that should be in there, then you've just done a denial of service. And that could take place either in authentication or any

other process that you are using. Now, if a general in a military command orders, I said, you must allow no bad guys in here. You're getting court martialed.

All right, sir. That's easy to do. Nobody gets in. Deny all. Not very business effective, but you've accomplished that overriding objective. No bad guy gets in because nobody gets in. Alternatively, he yells, Matt, other general couldn't get in there. If you ever block anybody again who's supposed to be in here, you're court martialed.

So guess what? Take the door off the hinges. Everybody gets in. You see, you [00:16:00] can't, you've got to balance somewhere in between there and we want to figure that out. So a lot of times what we've used historically... for validation. Authentication is passwords. Now, the Roman legions used to use watchwords passwords, and they would be able to say if you knew the right phrase or whatever, you're friendly, and if not, that would be a problem.

Back in World War II, the United States and the island hopping campaigns that would take place by the Marines and things such as that. they would have past phrases and usually would have the letter L in it, like lovely lady. Why? Because if you're a native Japanese speaker and there is no L in your language, you're going to come up with something more like a rubberly righty, and it's not going to sound quite right.

Or there could be other things like who in the world series, or... who had the batting average or things such as that, basically challenge response. And that could be done, but a password prohibition, there are passwords to get into the speakeasies. You had to know the right phrase. Anybody who's ever played the old classic leisure suit Larry game, you had to know the password. Ken sent me, by the way, if you ever want to look for that. [00:17:00] But the first digital password was really established around 1961 by an MIT professor, Fernando Corbató, who provided private access to individual students to a timeshare computer as a way to keep the resources separate. And then from there, we've developed pretty much our whole idea of how we're going to be doing digital passwords.

Now, I started working on a PDP 11 back in the 70s at Northwestern, and you could have it generate randomly a password. It would be eight capital letters and digits and you figure that's 36 to the eighth power, which is about 2. 8 trillion. And when you consider that there is a fairly primitive computing back then, it would take a long time to go to 2.8 trillion attempts. However, it's being mapped

into a 16 bit register, which only had 65, 535 values. So yeah, you get a lot of collisions of possible passwords, but they only get stored.

So you didn't have that many things to search. Now, later implementations allowed more characters, but they would typically truncate to eight. So you still have the problem that you're only going to have [00:18:00] eight characters in there and, uh, you know, two bytes per character. So 2, 4, 8, 16, 32. So you get, you know, pretty good number that are possible in there.

But it's still problem is, is that we are going to have to figure out a better way to make sure that people can't, well, guess it because we say now that. People guess passwords. So Robert Morris in 1978 came up with a crypt function, which took about one second to run, which is a long time in computing because it's a lot of iterations.

And the idea though is that if I'm going to encrypt a password multiple, multiple times to get some eventual value here. If there were only 65, 000 possible values, 65, 000 seconds is less than a day. So what did he do differently? He came up with the idea of a salt. A salt is a piece of information you store in the clear with the user ID.

And then that gets added to whatever password they provide, and that whole thing is encrypted together. A 12 bit salt, which isn't that much back then, but that was 4, 096 possibilities. And now, even if you had 64 [00:19:00] bits encoded as an 11 printable character, like the Etsy password, now all those bits, we've upped that epi effort to about eight and a half years, even for a 16 bit password at the speed of PDP 11s.

And by the way, that's kind of why we started with the eight character uppercase, lowercase number of special characters. We've gone far beyond that point. The whole idea was you had to change it every 30 days or so because we figured that was kind of a minimal risk for somebody having tried brute forcing all the possible combinations.

By the way, lots of ways you can do password guessing, but brute forcing will always work, given enough time. It might be longer than the age of the universe, but you can get there. Now, if you look at your standard U. S. QWERTY English keyboard, you'll find that there's 95 keys that you can press.

I'm not talking about the special characters in the Alt 252 or something like that. Well, even only 8 characters, we'd be 95 to the 8th power or 6. 6 quadrillion combinations. And that's a pretty decent number. Now, the problem with

[00:20:00] that is, is the average person doesn't use. All 95 of those characters. We tend to really restrict ourselves down.

Letters, numbers, that's about it. Now, there was an examination back in 2015, about 10 million passwords that were aggregated in the clear. And you can look up this study online. It should be in the show notes. And it turns out that humans are a really bad source of entropy. The underscore, or a dot, or a dash, was used in less than 1 in 300 passwords.

Bang, at, star, dollar sign, question mark, less than 1 in a thousand. Thank Ampersand and percent less than one in 10, 000 and any other special character, give it up. It's just, it was so rare as to not even be on the horizon. Now, the thing is a lot of software and websites will limit your input of special characters.

Or they might convert uppercase to lowercase and they do a little bit of restricting for you. So already. The danger is, is that the passwords that [00:21:00] are universal and you can pick is a little bit smaller. The whole value here is entropy, having randomness, having a large amount of entropy. So in number theory, you're not going to be able to go ahead and say, yeah, that's the right answer.

Okay, we'll get into quantum computing some other episode. But the idea is here is that it's going to take effort. to be able to do that. And what we're trying to do is drive the level of effort beyond the value of the password. If the password is protecting a 1,000 resource and it takes 10,000 for the computer time, you're probably okay.

Now, granted, someone could go ahead and run a university attack because they're not paying for the time and come up with the answer. But in general, the whole idea of complexity is to drive the cost of an attack beyond the value of what it is you're protecting. Now, we haven't always had a clear history in that.

For example, Microsoft had the LANMAN hash, and this preceded the NTLM, the NT LANMAN. And this was back in Windows 95 or Windows 98 and things like that. And what it'd do is you'd enter in your password. Okay, good. I've got a big 20 character long scary password. I'm good. [00:22:00] Except Microsoft would first convert all your lowercase to uppercase.

And then they would truncate it at 14 characters and throw away everything else. Then if you had less than 14, we pat, they pat it with nulls. So you knew that the last one, two, three, up to six of those characters, because why not more

than six? Cause then they chopped that 14 into two pieces. You had seven character and seven character.

And then you encrypt those things. Well, how difficult is it to go ahead and try all? Combinations of 7 characters, or maybe 6 characters with a null, or 5 with 2 nulls. In fact, an 8 character password was a 7 with a 1. Not very good, is it? And so, granted that 14 characters is pretty good, but, by the way, it's still possible in Windows 10 to end up computing these old LANMAN passwords.

There is a registry key, which should be turned on, but if you want to play around with the registry, HKEY, LM, SYSTEM, CURRENT CONTROLS, SET, CONTROL, LSA, [00:23:00] and there should be an entry. No LM hash and the DWORD set to 1. If it's not there and the DWORD is set to 0, guess what? Anytime somebody inputs a password, it's not only saved in the more modern crypto, but like a Rosetta Stone, there's a copy of it saved in the ancient LANMAN.

And that was turned on by default as far back, as recently as XP. And it will still work in Windows 10. So another way around that, of course, is to go ahead and make sure that you have 15 character password minimum, at which point it won't even try to save a LANMAN because it doesn't make sense. But this is how L0phtcrack got its legs.

So L0phtcrack, L0phtcrack, was created by Loft Heavy Industries back in 1997. Now, they merged with At Stake, which became sort of a professional company, so they could make some money. They got bought out by Symantec in 2004. It's typically used for password auditing. You can use it, and then if you get a copy of the SAM or get a copy of the password database, you just bang, bang, bang, bang, bang, bang, bang on it, and you try all these inputs, and you say, Hey, guess what?

We found out the passwords. Now, it's open source today. It's no longer [00:24:00] proprietary and it will support GPU cracking. It could go a little bit faster. And there's John the Ripper and a lot of other tools that if you're a red teamer, you are familiar with these things. But one way that was come up as a defense against somebody, for example, having complexity is to say, wait a minute.

So it takes months to try all the combinations. Why don't we just pre compute them all? And then we got them all. And then we take the hash. We look it up in a ginormous table of all possible hashes, password hashes, and there's a match. And that's what it was. And that's the concept of a rainbow table.

Rainbow tables trade time for space, and saying if I got a whole lot of hard drive, and I got a lot of time to do it, then I can go ahead and compute these things in advance. Now, Philippe Echelin gets credit for coming up with this idea, although there's a little bit of work, I think, done earlier by Ron Rivest.

But, you can download these huge tables. Some of them are larger than one terabyte. Or come to DEFCON and go to the Data Duplication Village. Now, every year they do [00:25:00] this for the last several years, and it's been up to this year, six terabyte drives. Now it's up to eight terabyte drives going forward.

And every big drive sells out in Vegas for the whole area that week before. So plan ahead. But if you bring three, 8 terabyte, SATA 3, 7200 RPM drives to DEFCON, take it to the Data Duplication Village. They run about 100 drives in parallel. It takes about 14 hours to populate a drive. And you can get every talk ever given at every Hacker Conference, all the rainbow tables, and even the mainframe printout of Snoopy.

They'll put pretty much everything in there. Well, that's on the game over for these passwords because they can go up to 7 characters, 8 characters and things like that. But the concept that we heard that was developed Several years earlier by Robert Morris, the concept of a salt can defeat the rainbow table.

Why? Because what I do is I take that salt value and I prepend it to the password, and then I basically, hash or encrypt all that. And that's this thing I store. When the user comes [00:26:00] in, enters their password, I prepend the salt, run the hash on that. If the hashes match, we're good. Never, ever, ever store those passwords in the clear, unless some places do it, and it's pretty dumb.

But none the case, what happens is, is that now I could get a rainbow table for all possible six, seven, and eight character passwords, up to lowercase, lowercase, number of special characters. But now I have a special requirement. I'm going after a user, and their salt might be one, two, three, four. Well, now I got to go ahead and if I'm in a hurry, I want to say, well, hey.

I need someone to build a rainbow table that begins with 1, 2, 3, 4, and then has 8 characters. Well, that's going to take a month and it's going to cost you 2 Bitcoin. Alright, pay it off, off you go. So you get in, alright fine, maybe you can pop that one user, but the next user has a salt of 5, 6, 7, 8. That 1, 2, 3, 4 table is no good anymore.

And so the idea of salts tends to defeat the concept of rainbow tables, and it's also kind of this upper lower, who's going to win. Now, if we look at strategies

for password management, don't worry, we're getting close to the password management tool set. What's the kind of thing that we typically hear about?

Well, [00:27:00] write on a yellow sticky and stick it underneath your keyboard. So I have right here a whole bunch of yellow stickies. And in fact, somewhere here in my drawer is a little pad of something that says don't write your password on this. You know, it's some clever marketing by somebody or whatever. But the idea being is that you put it there.

I've seen situations, you probably have too, when they're talking, you know, we're here on a TV interview with the head of this emergency management agency, and there on the wall is the ID and the password of a particular site. Just stuck up there on an eight and a half by eleven piece of paper. It's like, yeah, that's not too good.

So don't do that. If you're doing a TV interview, sanitize your background first. If you write on a piece of paper and then stick it in your wallet, then it's less obvious than it being in your desk drawer at the bottom of the keyboard and someone would have to go ahead and pop your wallet to get that.

Or what's common thing to do and it's easy to do is put it in your browser. So Google, their latest version, I think Chrome is up to 1.16, is allowing you to go what? I can go ahead and store it into the browser. Well, I hear an awful lot of people going, Danger Will Robinson. Don't do that. Why? Because to get [00:28:00] access to that password vault, all I need to have is the password for the device.

Now, if I'm here on my Windows computer, I've got a fairly complex long password to log in. I don't mind. And if you knew that, then you can get my vault. But if I'm synchronizing with a phone, and my PIN is four digits or something easy, or you visual it and you had a five minute timeout, it's sitting there on the table.

Then someone's in, and they could get access to your stuff, uh, and they can see that. So your PIN would do the trick, and now they've dumped it. Now, if you want to see another risk, go to a website. Here's an experiment you can do. Go to any login screen. Let's say, like, Gmail, for an example. Put it in your ID, then get to the next screen where it has your password.

Enter in your password. Now right click where it has your password and click on Inspect. And then you'll see a whole bunch of code change input type from password to text. And when you click back on the left side, your password,

voila, you can see it. And so if someone left the machine open or some sort of phishing [00:29:00] attack, maybe you could hook it from the browser.

And so what you want to do then is going to consider to use a commercial password manager. Before we get into that, quick word from one of our sponsors.

For those valuing leadership, policy and governance. In tech risk and security, C Prime is here to help. Enhance your skills with our training and workshops, ensuring effective policy design and strategy alignment. As a tech coaching firm, C Prime offers classes for teams and executives on security analytics and risk management. Led by a C Prime expert, align expectations, prioritize and map tools for robust governance across your tech portfolio. Upgrade risk management at CPRIME.com/train and use code CPRIMEPOD for 15% off training. That's CPRIME.com/train and elevate your approach.

Now when we talk about using password managers, this requires an extra password to open, a master password, a key encrypting key, if you will. And you could also enable MFA, multi factor [00:30:00] authentication. That's a little bit different than storing in the browser.

And so let's use LastPass as an example. If you have a poor master password, then it could be potentially brute force if somebody is able to get access to all that encrypted data on the back end. You just download the backup and go after the master key through brute force. Well, that was the essence of the compromise that LastPass had in 2022.

And they fessed up eventually and they said, here's what happened and here's how the attackers got in. But it does call into question a little bit of the concern about how strong is your front door? Uh... I was a LastPass customer at that time, but my master password I think was 38 characters, good luck with that, and so as a result, I'm not too too worried about that, except...

The customer data vault that was compromised also contained unencrypted data like the website URLs that people were logging into, customers access via the password manager, the thing. So if you say, hey, wait a minute, who's using, abc. com because it's a target and you could get this old database. Wow.

These are [00:31:00] all the people that are using abc. com and that's, that's a common interest. You get company names, billing addresses, email addresses, phone numbers, and even. Customer IP address. So that's, that's a bit of a concern that those are all stored in the clear. Uh, the threat actor exploited a

vulnerable third party media software package, implanted keylogger malware on the engineer's device.

This threat actor was able to capture the employee's master password right as it's being entered after the employee authenticate with MFA and then gain the access to the corporate vault. And so I'm not dissing this particular company, I think everybody's vulnerable to certain sophisticated attacks. If the attacker thinks that it's valuable enough and I can throw enough resources at it.

So please don't think I'm picking on any one company, but I'm using this as an example of some things that could potentially go wrong. And you have to factor into risk calculus. Now, as a LastPass user, what bothers me is I get weekly alerts saying I have duplicate passwords. My question is, why are you peeking at my passwords?

Now, if passwords can only be decrypted [00:32:00] with my master password or some permutation thereof, this suggests that they are not individually salting the passwords and they're only looking at the matching ciphertext. As a result, we could see Anybody grabbing that database could also see where all the matches are, knowing if you get into one, you can get into others.

So if I could be king for a day, that's what I would change. Nonetheless. We consider one of the leading solutions out there, and they've made some fixes. I'll mention just a couple other ones. And again, this is not a sponsored episode with regard to password managers. I'm not pushing one product over another, but it's trying to give you a feel for what's out there.

KeePass is available for local storage only, and like a kitten, it's free to go home. Means it's open source, you can download it. But from what I understand, it doesn't have a built in one time password generator. There's no browser extension built into it, and there's really no easy way to manage an enterprise user base.

Looking at LastPass does have an enterprise management. I get a console, I can see everybody out [00:33:00] there, I can see how many sites they have, I can look at the relative strength of their passwords, I don't get to see the passwords. But as the administrator, I can tell when somebody is, you know, not doing so well.

And they said, you know, you need to up your game a little bit because you've got the same password on 15 sites. That way you can see. You don't see what it

is, but it gives you the warning. Now, it turns out that there have been some issues and things like that with KeePass. They had a CVSS 7. 5 vulnerability.

CVE 2022 0725, if you really want to look it up. And it turns out it was logging plain text passwords in system logs. Oops. Well, that's been fixed. And so the version 2.54, which came out June 3rd, 2023, is now been patched for that. But if you have an older version, look out, because potentially an hacker could do code execution and steal passwords.

Now, when I mention a CVE, are the common vulnerability and the exploits you find out for a given vendor or product. If you go to cve.report, not com, report, it's one of the generic TLDs, you can enter in the upper right hand corner [00:34:00] the name of a vendor or something like that, and it'll give you all things that are associated with it.

One company I've looked at is 1Password. They offer personal, business, enterprise, and developer options. And, their marketing efforts say, Hey, we've not been compromised. And they're going after other companies have had a compromise and things like that. Now you have to go back a couple of years to find a pretty bad CVE.

2020-10256 had a 9.8 CVSS score, but it was in a beta version. It basically meant that somebody could decrypt passwords. And so that's been fixed as well. Now Gartner lists nearly 40 password management tools. And you can look at their site, just for completeness. I'll mention just the top few.

LastPass, Keeper, SpecOps, ManageEngine, Dell 1Password, BravuraPass, ZohoVault, SailPoint, Bitwarden, NordPass, et al and more. But TechRepublic has a comparison table for the top choices. You can use that as part of your research to try to figure out, okay, I want to make a decision based upon what's best for my enterprise.

But [00:35:00] I want to be able to do so by understanding a little bit more and someone else has done a lot of digging, take advantage of that research. So when you're trying to say, what do I need to consider in a password management suite? Let me give you some ideas. Security. Is there any history of vendor compromise?

It may be if, not when, so understand their strategy and their architecture. Ask them questions. How do you encrypt things? How do you manage this? How do you do your DevOps? How do you ensure the secrets that you protect are

protected? Do you encrypt the encryption stuff, etc., etc.? Number two, an important element, user acceptability.

There's training involved. There's ease of use. Can users self-service if they need to fix something. Do you have to get your help desk in every time somebody has a problem? And can you administer it centrally? It's a big deal. How about recoverability? If an employee leaves or goes bad, can you recover the password vault as an administrator? And that means you better really, really trust your administrators.

Which brings up the next [00:36:00] question on auditability. In the event of a compromise, do you have a sufficiently detailed and tamper proof logs to do an investigation? So if you believe that somebody was involved in saying this person has compromised it, then Can you look it up and figure out who did what?

Jupyter One did an event over at Black Hat where there was sort of a whodunit. It was an interesting dinner theater. I'd never seen anybody do it quite this way. But the dinner theater was who... Did this compromise, but you had to use their tool to figure it out. Now they're not a password manager, but it did allow you to go through all the logs and sift through them.

So that was a vendor that had a pretty good idea about, Hey, we're going to bring everybody in here and we're going to set you up with teams and you can go solve the challenge and we can figure it out. Um, our team did figure it out by the way. So, so good for everybody that was at our dinner table. So, you know, we got nice dessert as well.

How about cost? Now notice that wasn't my first. Criterion. But it is important. Is it cost pretty much by price by seat? Is it an annual contract? And then how do they do step functions in pricing? Because [00:37:00] typically say, oh, you buy so many prices, then the seat price goes down. Which means, for example, 500 seats could actually cost less than 451, depending upon the plan.

So you want to look at that carefully. And if you're close to an edge, you might want to overbuy a little bit because your net cost goes lower. And then last one I had on my list, and you could come up with others, was availability. If their servers go down or their DDoS. Are you SOL? Are you out of luck?

Are you unable to use any access to your enterprise because none of that information is available? One of the advantages of having things locally is you're not subject to a vendor outage or you have a communications outage or

something like that. Now, there are risks of password managers, and as I say, the first one, which you discussed already, was password, I mean, sorry, vendor compromise.

Something goes wrong, and software is incredibly complicated, people make mistakes, they overlook things, and attackers figure out cool things. The other one is that a keylogger, if that's a danger, somebody installing something on the software, the nice thing about a password manager is the only thing that you're [00:38:00] really vulnerable to on a keylogger is that master password.

Because you're not typing in all the other passwords because they're just populating into the web browser. So, how do you keep that from being a risk? Use multi factor authentication. I suggest you inquire, demand all your users use MFA if they're going to use your password manager. I do. Now users are sloppy and they reuse passwords.

And they fail to invoke the complexity engine for new passwords. Because sometimes it'll say, do you want this big, long, scary thing? Now, what I think is missing out there, so here's your Shark Tank idea, is build up a database of all the major vendors that have login passwords, all the websites, get the complexity rules, build that into a back end engine and say, hey, we'll make it that complex.

If you're logging into Microsoft, give you 127 characters. You can do that, but who offers that? Oh, do you want 16? Do you want 20? You want upper or lower? You ought to know that stuff. Put it in a vendor database and maybe they'll buy your product. How's that one? And then if you lose the master password and there's no way to escrow it, you're in real trouble.[00:39:00]

Now, if we take a look at compliance. As a CISO, I get compliance checklists from potential clients and business partners. And some of them are excruciatingly detailed. Hundreds and hundreds of questions and things like that. Some of them are just hand waving. You just do that. I went through a whole process where you had to do this, and they say, Okay, submit evidence.

Get us a copy of your policies. Give us screenshots. Give us this. Give us... And they're iteration three, four, five times. There are some companies that are coming up with standardized ways so that if you subscribe to them and you go through this effort once, And then someone says, well, we want to check your risk as a vendor.

He'd say, well, check with them. They've already done the report and they've done their due diligence. Now, say not all of these are well written. And one of the concerns I see on some of these has been that many require a testing that you do frequent password changes and you change every 30 days or every 60 days or every N days.

First of all, that is contrary to the current guidance from the National Institute of Standards and Technology. As you may be aware, NIST [00:40:00] publishes Special Publication Series 800. And these are all computer security, cyber security documents. In this Special Pub 800 63B, sorry military, I always spell out there so it doesn't sound like a letter.

63B, Digital Identity Guidelines, Authentication and Lifecycle Management. Has... Portion in there. And let me read to you part of chapter and paragraph 10. 2. 1 on Memorize Secrets. It says, when users create and change Memorize Secrets, clearly communicate the Memorize Secret requirements. Meaning that need to know how long are they?

How complex are they? What are the characters you're allowed to use? Maybe some of you cannot. How's this one? Number two, allow at least. 64 characters in length to support the use of past phrases. Encourage users to make memorized secrets as lengthy as they want, using any characters they like, including spaces, thus aiding memorization.

The little girl went to [00:41:00] the store to buy some food for her cat. It's a whole lot easier to memorize than three uppercase, not, you know, uh, left parentheses, WQ, uppercase N, bang, whatever we have to, that's the entropy. That's to create the randomness. Length is always better than complexity. Why? Because if I go ahead and I take a low number and I increase the exponent, even if it's binary, 2 to the 256 is a huge number.

That's 10 to the 38th power. Okay. Actually, you know, it's 120 bits would do that. Two to 56 would be even bigger. All right. And so you can do the math work on it. It's gonna be, what, 10 to 72, doing it in my head. But it's compared to a large number of complex things, but very few. So it actually goes up a lot faster.

Do the math. It works out. Number three, do not impose other composition rules. For example, mixtures of different character types and memorize secrets. You hear that? No requirement for uppercase, lowercase numbers [00:42:00] and special characters. Just make it a big, long, freakin passphrase and be all lowercase. And if it's long enough, you're secure.

And number four, and this is key, do not require that memorized secrets be changed arbitrarily, that is periodically, unless there is a user request or evidence of authentication compromise. One and done. Why? Because people are a bad source of entropy and they take their password and they add a one, or two, or three.

Or some punctuation at the end, and that's how their password change. That's what people do. So don't require that. If you have MFA turned on, you're in really good shape to begin with. And if you have a long, complex password that someone picked out, or has managed you a password manager, So that it's picking up these uppercase, lowercase numbers, special characters.

And you have MFA, you're extraordinarily low risk at that point for somebody popping you. They're going to come in through some other way. They're not going to do through credential compromise. And, oh, by the way, if you use single factor, one time password, like the little apps and stuff like that, um, paragraph 5141 says a minimum of six digits [00:43:00] changed at least every two minutes.

You know, mine says every 30 seconds or so, but that's the minimum. So how do you build trust in an organization? If an external auditor validates the controls and approves them. So look for password management companies who have had pen tests and external reviews and on their password managers, for example.

And again, not picking anybody, but Bitwarden actually is a positive comment. Publishes their third party audits and runs an active bug bounty program with HackerOne. And that's the best level of diligence I think you can expect to find. So if you check their bug bounty record, you'll find they have a record of fixing flaws.

in a timely manner. Okay, so let's wrap up. We've covered a whole bunch of things, and one of the basics about authentication, and it's a proof of identity. Something you have a factor, something I know, something I have, something I am, some place I am. Passwords are something I know. We look at the idea of being able to create complexity so that it's harder for someone to brute force all of them.

And even if someone tries to pre compute all those brute forces [00:44:00] through something like a rainbow table, by adding a salt, which is going to be different for every user, you kind of defeat that attack as well. When we look at

the problem of users storing their own passwords, they're going to stick them on yellow stickies.

They're going to write them down. They're going to put them on the wall. They're going to put them in their wallet and they're going to lose them and all those things are potentially vulnerable. And so what we want to then have is a way to manage that centrally. A lot of the password management software out there is mature.

They've gone through a few iterations. Some of them had some issues, but they're pretty quick to fix them. And it gives you an enterprise level solution where you can manage that and ensure that your people are protecting to the higher level than they could do just manually. And the other good thing about that is, is that in the event that somebody leaves or something happens to them, if you're escrowing those master keys, you can still get access to it and you don't lose a code repository and the like.

So look at the criteria that you're going to set up. They could be different for everybody, but my thought was think about security, acceptability for users, [00:45:00] recoverability. auditability, cost, and the availability if the systems were to go down. And then if so, you've got a solution that works. Well, thank you for listening to CISO Tradecraft.

I hope you found this episode to be useful and informative. This is your host, G Mark Hardy, and until next time, stay safe out there.