Course Overview

Hi everyone. My name is Dale Meredith, and I'd like to welcome you to my course, The Issues of Identity and Access Management. Now I've been a Microsoft trainer since 1998, and it goes back a ways, as well as a cyber security trainer since about 2004. I'm also currently a consultant for several different government agencies for cyber security. Now this course is actually part of a series that supports several different certifications, including GCIH and the new CSA+. Now let me ask you a question, how many user accounts and passwords do you have in your life. Anyone, anyone, Buehler, Buehler. The average is about 118 accounts. Can you believe that? I actually did an inventory of mine and I have almost 150. In this course, we're going to talk about the differences between some of the solutions that are being used, as well as the weaknesses that they might actually cause. Now I'm not going to leave you hanging. I never do. I'll tell you some of the cool ways that we can strengthen the use of some of these solutions, such as, you know, not writing down a password on a Post-it note. Now some of the topics that we'll cover will include things like OAuth and OpenID and where the weaknesses in their technologies lays. We'll also look at things like SSO, as well as federations, and I'll even go through and show you how to set up a hacking environment that is not only quick, but easy to deploy using our auto labs so that you can test some of these weaknesses. By the end of this course, you'll be able to look at you IAM solution and see if you're protecting yourself, as well as your users. You'll also be a couple of steps closer in taking exams with the knowledge that you get from within this course, you know those CSA+, the GCIH, and ECIH. Now before beginning this course, you should have a somewhat basic knowledge of network technologies, including things like routers, and switches, and maybe even a little bit of Active Directory. After you watch this course, from here you should be able to feel comfortable in diving into some other courses within the series or even branching out and looking at the ethical hacking series. I hope that you'll join me in this adventure in learning with the Issues of Identity and Access Management course here, at Pluralsight.

It's All About Control

Series Intro-Overview

So this is kind of crazy, but let's talk about something real fast before we get into the full course itself. How many accounts do you have? And what I mean by that is want you to start thinking about all of the different accounts that you have. Let's start off with our online storage, right. Last I counted, there was like 17. There is probably more than that, but I could find 17 different services online like OneDrive, Dropbox, Google Drive, Box, Amazon Cloud Drive, iCloud, Backblaze, FlipDrive, HiDrive, IDrive, pCloud, OzzyBox, a ton of them, and how many of them do you personally have. My next question for you though is what about your email, how many email accounts do you have. I've got a couple of Google accounts, as well as I've got a couple of Hotmail, and of course, one or two business accounts. And then we start thinking about well how many banking institutes do we work with. Do you do online banking? Now I'm just talking about your local branch. I'm not talking about credit cards. What are you doing on credit cards? How many do you have? I don't think four probably is enough for most of us. But we then have to start thinking about, as far as accounts are concerned, the social networking side of things like, again Facebook, or Instagram, or Twitter, or what's the new one now, Mastodon, I think is the name of it. After we deal with that, then we have to worry about what accounts on our phones, tablets, the internet of things, man that things driving me crazy, what about your cars, or even apps that run on all your different devices. I have an app for my dishwasher believe it or not. It's so I can do diagnostics. And each one of these technologies require us to create an account. So is it any surprise to any of you that the average person has about 118 accounts. Yeah, that was kind of shocking for me too, in fact, I almost doubted it. I went and took a look at my password manager. I use a password manager. Yeah, I'm pushing 150 on the accounts, which is kind of surprising. Now when you talk about those accounts, my next question for you is how many passwords do you have. Hi. My name is Dale Meredith, and I'd like to welcome you out to The Issues of Identity and Access Management, which is also known as IAM or IAM, and this particular module is called It's All About Control. Now I've been a Microsoft certified trainer for wow since about 1998. I've also been an ethical hacker and an ethical hacker instructor for a while since like 2006, and I like to include all my different certifications here, including some I like to make up like RSVP, PDQ. Okay, those are certifications I worked hard on. If you can't tell already as we get into this course, I love having fun and having fun actually will help you in retaining information. You can try to cram it in, but they once read a study that if you make people laugh or if you laugh while learning, you'll actually retain the information longer. So my jokes aren't that fantastic because I'm not a stand-up comedian. But I want to talk to you in this particular course about several different things. We're going to go through, and of course, as I mentioned before, we're going to talk about it's all about control, and in that, we'll take a look at what is IAM or IAM. We'll then go through and take a look at how we can manage all these different security identities that we go through and create and the problem that we have with having so many of those. We'll then go through and take a look at some other authentication methods. There's several of them out there. I will go in and take a look at the pros and cons, as well as different repositories for identities, like a RADIUS Server or maybe an SQL database. Now one thing I do want to do is I want you to be able to kind of play a bit, and in order to play, we're going to help you, or at least, I'm going to try to help you just set up a lab and this is really exciting because Pluralsight and some really fantastic authors went through and created a really easy way for you to fire up a lab really fast and get rid of it really fast and pull it back if you want to. So I'll take you through the process of setting up this little playground area that we're going to have some fun in. And once we set that up, we'll then go through and take a look at all the different exploits and how to protect ourselves when it comes to IAM. Now in this particular module, we'll go through and take a look at, as I mentioned before, what is IAM or IAM. We'll also go through and talk about some of the components of IAM, which would include things like authentication. We'll also go through and take a look at two-factor authentication. And then we'll take a look at something that always throws people, it's called authorization, and what I mean by that it throws people is that they get them confused, authorization and authentication. We'll then go through and talk about role based identities, as well as some directory structures that can help us out with authentication or during this IAM process. And of course, the whole purpose here is to help us out with something called SSO, which is short for Single Sign-On and it's exactly what it sounds like. It's single sign-on and we move on. And of course, we'll take a look at the lifecycle of IAM. So if you're ready to get going, I am too. And let's jump into this.

What Is IAM?

So what exactly is IAM or IAM. Identity Access Management is actually a framework for business processes that help to facilitate the management of electronic identities. That sounds really complicated, doesn't it? I'm going to make it really easy for us to understand. Basically, it's used to initiate, capture, record, and manage user identities and their access permission to different resources. Now we all know that we typically have obviously our businesses and we have different resources that we want to give access to and we're going to have some type of IAM installed. And most of the time, we see things like Active Directory being utilized. And this service or system can help us manage and reduce the complexity and costs of handling our accounts and passwords. Now here's what's interesting is that not all of our identities are stored by our own companies. However, id management systems can also give organizations a way to control this flood that we're getting of mobile devices, what we refer to as untethered endpoints, whether those be laptops or cell phones or tablets. Now most the time, these systems we actually store on-premise or we have them within our organization, but there's been a big push and flux into this new concept of actually providing identity as a service within the cloud or we might even have a combination of the two technologies working together. A great reference for this is something called Azure's Active Directory. This particular product actually is very beneficial for identifying and giving access capabilities for applications running in Azure or running in applications that may be on-premise. Now it's not a replacement for active directory, but it's kind of a feature or a bonus feature for AD. Now as I mentioned before, Identity Access Management, or IAM, is basically a framework for, again, us to manage these different identities, at least the ones that we are in control of. Now poorly controlled processes could actually lead to regulatory and non-compliance issues, so you need to make sure that you have some type of IAM in place, and most of us do. If you don't, it is sometimes difficult to get budget set up for IAM projects because they really don't increase profitability or function. It's there as a protection for us. And how it's protecting us is it creates this whole workflow that allows us to manage the user account, what access it has, from where it has the access through the entire lifecycle of the user account itself. So now that we understand what IAM is at a basic level, let's kind of dig into some of the terms that we're going to be using quite often. Let's first start off with authentication.

Authentication

Okay, let's start off with authentication. Now I'm sure that we can all look at something and determine if it's authentic or not authentic, right. Now when it comes to digital authentication, what we're talking about here typically is a username and password. So let's break this down a bit. A username simply or typically just represents a user and that user is represented inside of some type of directory structure, in this case here, Active Directory, as an identifier and here you can see the identifying for Bruce Wayne is S-1-5-21-2235 yeah, yeah, blah, blah, blah, blah. It is actually this identifier that computers look at when they go to authenticate somebody. Now when it comes to the password itself, what we're really doing here is we're creating what they refer to as a shared secret. By combining the two together, it allows us to go through and actually confirm the identity in making sure that they were actually authentic. Now let's take a look at this in the real world. In the real world, we have our user accounts and we have our resources that we're trying to gain access to. Well what happens here is that typically, the user goes and tries to gain access to the resource and our IAM system jumps in and says hang on a second, you need to tell me who you are. The user provides their username and password, and therefore, we authenticate the user, and because the authenticated user has rights to this particular resource, they're able to go through and gain access. On the same token, even though the user has been authenticated, if they don't have rights to the resource, the IAM solution is going to go through and deny them access. So that's pretty easy, right. Authentication.

Two-factor Authentication

So where does two-factor authentication come into play? Well it's kind of like that old saying of whether two brains are better than one or two minds are better than one. Listen, we know that our authentication mechanism is not perfect, in fact, it's imperfect. And so, it'd be great if we could somehow doublecheck and make sure that the authenticated person is really and truly that particular identity. Now there is that famous statement that we've always heard when it comes to two factor authentication and that is that we typically would like to use something that you are, such as a fingerprint or a retinal scan. Something you know, which would be your password, and then of course, something you have, which could be something like a device, a key fob, my favorite, you know, obviously if I'm dealing with Dropbox and I've got a new device that I'm going to hook into my Dropbox environment, I get that nifty little text message with that code that I have to type in back into Dropbox to make sure that it's verified, that even though we've provided the username and password, it's giving me that two-factor authentication. Now guess what, with that statement, we've got a new and improved statement because technology is changing and what we're starting to see is that two-factor authentication, or we can also refer to it as multi-factor authentication, could include things like location based. For example, if I'm logging into my device in Salt Lake City, Utah, or if it then looks like I'm trying to authenticate in Antwerp, Belgium at the same time, that could create a flag for us, right. We're also starting to see a new technology that allows our systems to look at the behaviors of an individual. We're starting to really see this when it comes to credit cards, right. I've been on trips often where not only is location used, all of a sudden they see me, you know, buy a ticket to, I don't know, Cancun, they see me check into a hotel in Cancun, they see some purchases, they know that I'm at that location, but if they see me starting to purchase things that I wouldn't normally purchase, I don't know, maybe I'd purchase some Spiderman stuff because we all know that I'm a huge Batman fan, maybe that throws a flag that hey whoa this guy is buying stuff that he wouldn't normally purchase or acting a specific way with his devices. So again, combining additional factors into authentication is going to do nothing, but help us. So now I know it's not convenient, and don't get me going on my tangent about whenever we talk about convenience, we're talking about a little bit lack luster security. If I had my way, I'd make sure that we had to do what four-factor authentication, especially when it came to, or when it comes to, dealing with top secret or company resources.

Authorization

Okay, now let's talk about authorization and this is typically where I have people go wait a minute, isn't that kind of what authentication is. Well, let me ask you a question. What is the difference between authentication and authorization? It's a big difference. Authentication means that I've just proven who you are. Authorization is based off who you are, do you have rights to this resource, and that's where IAM solution typically kicks in is it looks at your authenticated identity and then checks a list on the resource to determine if you can get in. Let's go down story time with Dale. So maybe I can make this a little bit clearer because I just like talking about a couple things in my world, One of them is Batman, the other one is a fantastic group that I thoroughly enjoy. This is actually the members of Collective Soul and these guys shine. Ah, see what I did there, a little pun. Anyway, any time they come into my area or my neck of the woods, I get tickets and I always take my daughter with me. I have one particular daughter that just loves them as much as I do. So when I purchased my ticket, I provided my credit card information and when we got there into will-call, we got these nifty little bracelets. The white one authorized us to go backstage and meet and greet with the band, my boys. I may have an addiction problem, or maybe there's a restraining order against me because I've done this several times, but hey it's just the world I know. See what I did there? Now I've got to find some Batman stuff to bring in.

Role Based IDs

So let's talk about role based ids. In the previous module, we talked about how we authenticate and authorize a user, but that can be extremely overwhelming, in fact, if we were to say this was our company and here are our departments and each one of these squares represents a user, in order to provide users access to the resources that they would need, for example, we might have an HR application, a directory for the HR folks, printers for HR, as well as obviously printers and resources for sales and executives. Going through and assigning individual users access or authorization to gain access to those resources can be very daunting. Instead, it might be easier to implement something referred to as role based where we create these groups, which represent roles, and every time I say roles I get hungry because I prefer mine with honey butter. But instead, we go through and we just simply assign roles to the permissions that are needed to the resources and then as users come into the company or get moved around in the company, we just simply assign them to the role that they need or that they belong to. This makes it extremely flexible, especially consider this, let's say that you have a top-secret project that you need to have done and they're going to have access to some resources and these top-secret folks might be located in different divisions within the company. Using role-based ids, it can make it extremely easy for us to kind of control what's going on. Does this make a lot more sense than trying to do this one at a time for individuals. Imagine the company that contained thousands of employees. Now here's what's interesting is that if you remember our authentication versus authorization slide, you remember this one, well we've authenticated, and as a user, we may be actually denied access to a resource, but because we're a member of a group or because we're a member of a role that has access, we could gain access to what we need.

Directories

Next up, we have something called directories and this is very easy and we deal with this almost daily, at least most of the time. We're going to try to make our life a little bit easier. Typically, when we try to gain access to a resource, we have the resource and the user tries to, using its credentials, tries to authenticate against that resource or maybe it's an application and it checks the local security rights of that machine. Well imagine having multiple users trying to gain access to multiple different applications. It would be a horrendous nightmare. So instead, why don't we allow a different location to store the information and it's a database. It's called Active Directory or there's open directory as well. It's simply a database that's going to check the current authentication and authorization of a particular user. So now, instead, when a user needs to gain access to a resource somewhere else, we don't have to worry about placing their password or their credentials on that box. What happens is they ask for a permission to that resource, and because that resource is aware of the directory, it contacts the directory and says, excuse me, do you know about Bruce Wayne and is he current and does he have access. Well the Active Directory environment looks at the user's identity, as well as its role based identity and submits back that yes, in fact, they do have access. And of course, at that point, we're able to do what we need to do on that particular box. So again, putting everything into a centralized location makes it a lot easier for us as far as authentication and authorization is concerned. Now let's talk about SSO.

SSO

Okay, so SSO or Single Sign On. This is very easy. We experience this almost daily. Because of the fact that we're authenticating ourselves, we're providing and one of our goals is to provide this single sign on service makes it easier for us to move about. Do you remember story time with Dale and we talked about being able to go backstage with Collective Soul? Well the only reason why we were able to do that was because we were given these tickets, and these tickets, or these wristbands, allowed us to go wherever we needed to go. Now some tokens can be counterfeited. I'm sure somebody could try to counterfeit a wristband like this if they knew ahead of time, and so that's kind of where having authentic tokens comes into play and most tokens, in the case of a ticket, might have a time, as well as a date, and of course, also a specification of where I'm allowed to go. So going back to what we just saw previously with working with directories, instead of just using our SSID or our identity to give us access to the resource, instead, we go through and we allow active directory to issue us a token and it's that token that is then presented to the resource to try to gain access. And so, then what happens is that the resource then contacts the directory to find out if the token is still valid because maybe I issue a token for people that's only good for 20 days or for an hour. You know, when you go to a website and you sign on, especially like your banking website, they issue you a token and that token can expire on you, I think my bank is like 10 minutes of inactivity, but that token that I'm issued allows me to go to different departments or different sites within my banking website. I can go and fill out a loan application. I don't have to relog in, I can go over and check my balance, I can transfer money, I can apply for a loan. There are all kinds of things that I can do, again, with that particular token.

IAM's Lifecycle

Now as far as the IAM lifecycle is concerned, this is very, very easy. It sounds kind of technical, but it's really not. The lifecycle itself consists of about three different levels or three different activities. Provisioning is simply the process of creating an account for a user within that directory or within the database of that directory. Provisioning also includes the concept of adding them to different roles within the environment so that they can gain access to their resources with that token. The next step within the lifecycle is deprovisioning. Yeah, that's basically when the user gets fired or moves onto a different department, and in which case, we're either going through and deleting the user or moving the user to a different directory structure. So it's simply those three things that make up the lifecycle or the circle of life. Yeah, it's provisioning or creating the accounts, managing the accounts as they go through different transitions or their different work lifecycles, and then of course, deprovisioning the account when they leave the company or just move on. Okay, great. Now I've got the whole circle of life song going through my head.

Summary

Okay, so what did we learn in this module? Well, we went first and talked about what is IAM or IAM. Remember, it's just basically an authentication authorization framework to help us to one, authenticate. That's where we're basically proving that we are who we say we are. We also talked about two-factor authentication or multi-factor authentication and the security that it brings to our environment. Remember, more is better. And then we also talked about authorization. Remember, there is a difference between authentication and authorization. Authentication, again, is just proving who you are, where authorization is based off who you are, what are you allowed to do. We then talked about role based IDs. If you remember, role based ids allows us to assign an authorization to different groups or different roles and then just simply place those people who have authenticated to the roles they belong to. And that kind of got us into handling with directories, which is just a centralized location for us to store the accounts, as well as their tokens. And then we talked about SSO, or Single Sign On. This allows users to authenticate once and then be issued those tokens that they can present to different resources to see if they have access. And then of course, we talked about the IAM lifecycle. Again, remember that's provisioning, managing, and then of course, decommissioning. So now that you have a good understanding of what IAM is, in the next module, we'll talk about how we can actually manage all of our identities.

Managing Your Secret Identity

Overview of Your IDs

Okay, since we're all IT superheroes, we know we need to manage our secret identities and that's what we're going to talk about in this module. Actually, I won't get into the superhero stuff, but I am going to take you guys down the road or the aspect of looking at digital ids, how we handle them, where they're being stored. We'll also take a look at endpoints and the servers that are involved when we're using our credentials. And then we'll go in and take a look at services and applications as well. So let's take off our masks and our capes and take a look at our secret identities.

Digital IDs

So let's talk about digital ids and what we mean by digital identities is basically a set of claims that can be made about an individual or a group of individuals. Now it's important for network security models that people, devices, and applications allow access to network resources. So whether we like it or not, when it comes to our digital identities, everybody's got one and when I say everybody, it's not just us geeks, it's everybody, whether you're in the medical profession, whether you're working at a retail location, it doesn't matter. We have these ids everywhere. And then we have these ids that we use while we're on the internet. Now I think we can all agree here that attackers will actually target employees as a means of trying to gain access to the network, and it's also known that password based credentials are kind of a problem for us. These issues can actually be addressed by delivering training, and you hear me talk about this over and over, but training is your first line of defense and educating different individuals within your environment depending on their security rights. But here's kind of what's funny when it comes to these identities is that we have to kind of pay attention to what they refer to as administrative staff because technically research actually shows the administrative staff often use very poor credential management, not changing passwords, sharing passwords, writing them down, reusing the passwords, even reusing the same password that they use internally on external sites. And obviously, administrators are granted too many permissions, or typically, have too many privileges or abuse accounts with these super privileges on a routine basis. The other issue that we see take place quite a bit is the shared accounts issue. This is where we use one password and everybody that's involved in the IT department or in a particular department knows what that password is and they just simply use it over and over and over. A shared account actually breaks what we refer to as the principle of nonrepudiation and makes it extremely hard to do any type of audit trail to establish what took place. Now up until this point, we've been talking about just individual users, but we have something else which we refer to as roles and then we just get really hungry when I start talking about roles because I like mine with honey butter. No, not a good one. So a role is basically a type of security group where all their members perform the same functionality or need the same access to a resource. Now it's likely to require the creation of more than one group in any type of organization. And so, when you have a new user that comes on board, you just simply add them to the roles that gives them access to the information that they need. Now since we're talking about roles, we might as well talk about Role Based Access Control, also known as RBAC, which is basically the idea of restricting what tasks users can perform within an application. Now one of the things here that needs to be noted is that as we start to use these roles or these group assigned permissions, the roles themselves need to be audited periodically to make sure that the permission assigned to the role are appropriate and that the membership of each role is valid and that accounts for the personnel who left or changed the jobs have been decommissioned. So something new and exciting that's starting to happen, and we're seeing this with Server 2016, is the new and improved Just Enough Administration, or JEA. Now JEA is actually a power scripting feature that allows administrators to go through and create basically very constraint environments so that we can give access to users that normally wouldn't have access to certain PowerShell scripts. We can also go through and set up the things that they can and can't do on a particular system during a particular time frame. So we could go through and say that Billy Bob can restart the server, but only for the next 15 minutes, or restart a server I should say. Kind of exciting. I highly recommend going and watching some of our courses on Server 2016 to learn about this new technology. I don't know, maybe the reason I get excited is because every time I say JEA, I keep thinking of Jenga, which is that game where you push the sticks out, no, completely different thing.

Endpoints & Servers

So let's talk about endpoints and servers now. In order to do that, let's a have a flashback, back in time and kind of remind ourselves of how networks work. You know, we have our systems. They're all hooked into a switch of some sort and they request access to resources, whether it's a file or whether it's a service itself. Now without any type of encryption in place, everything is identifiable on the network via a MAC address and an IP address, which means that we can't actually spoof an address and make people think that we're actually a service or a server. We refer to this as spoofing an address. So how do we stop this from happening? Well it's called Rogue Machine Detection, and it's basically the process of identifying and removing machines on the network that really aren't supposed to be there, have no business on your network. We typically see this with I call them weekend geeks or hobbyists that happen to be users on your network. Now when I say machine, you need to be aware that it could mean anything from a wired client like a PC, a laptop, an appliance, a server, also wireless devices, which again, could be our PCs and laptops, as well as mobile devices. Software, as I mentioned before, we have a tendency of seeing things like Rogue DHCP servers or people trying to to fire up their own DNS servers. And this is another one that's kind of popular at this point and that is virtual systems, someone being able fire up a virtual machine, do their dirty deeds, and then get off. And so you may ask yourself, how do we stop it? Well one of the things we can do is actually visually do some inspection, physically look at ports, what's plugged into the back of the servers could actually kind of reveal some unauthorized devices and/or appliances. Personally, when I do pin tests, a lot of my assets that I use in my hacking, I just simply put a piece of tape on it and say IT department, don't remove, and who's going to touch it, so you need to be aware of what's going on. Also, something else you want to take into consideration, this is from a digital perspective, but looking at mapping out the network, as well as any new hosts that fire up. I actually do this at my home level. I use an app called Fing, F-I-N-G, which gives me all kinds of information about the wireless devices that are hooked up. I should say wireless or wired devices as they hook up and I can either deny or grant them permission. So I know this can be kind of difficult, especially in large environments, but most of us, I think, probably have some type of software out there that is doing host discovery. Now I mentioned wireless and that's something we need to look at as well as wireless needs to be monitored. It can actually show you when any time an unauthorized or malicious access point fires up or somebody turns their laptop into an access point, a very famous attack. There's a new device that's coming out, it's also by Fing, not that I bet paid by them or anything, I just when I have cool tools, I like sharing them with you. This device once it hooks into my network will actually send me messages when somebody walks by my house and they see my network. It'll actually tell me that their cell phone is looking at my network and maybe I'll have some fun with them. Yeah, you don't want to be my neighbor. Well besides wireless monitoring, we also need to be looking at network monitoring, which can give you quite a bit of information as far as unauthorized protocols being possibly used, identifying hosts that are creating a ton of traffic, and of course, I'm sure some of you guys maybe saying that's cool Dale, we've actually got a nice little intrusion detection system or a network access control system that's helping us to scan at all times and defend and that's great too. Any one of these or combination of these techniques can actually be extremely beneficial to you and your company. So you want to walk by my house?

Services & Applications

Okay, something else we need to really consider is services and applications. Because of the fact that we are utilizing services and applications and assigning permissions or access to these resources, we need to make sure that we're controlling them. You know, when you think about it, our applications and services are actually being ran and controlled by, yes, programming code, right, code that's been created by a third-party or maybe in-house. Well the problem here is that you need to make sure the code has not been modified by anyone else along the way. Now one of the best methods that we can do to protect our services and applications is implement what they refer to as execution control, which is basically the process of going through and determining which software needs to be installed or not installed through either a blacklist or a whitelist environment. One great example of this is you know when you go into, for example, Server 2012, well actually I want to go back to probably even Server 08, when you wanted to install a service, like for example, DHCP, the server product would go through and say okay look you can install that, but we also need to install these additional components or these additional programs, which are associated to execution. And as I like to say, when you install server, you get an OS and that's all you get. You have to go through and specify what you want to or not allow. Again, whitelist control just means that nothing is going to run if it's not on the list. It's like that guy in the velvet rope, right, you're not getting in the club unless you're on the list. Check again, my name is Dale. No, it says here you've been blacklisted. Yeah, that means basically I can't do anything. Now some other great examples of blacklist versus whitelist is if you were to think about your antivirus solutions, right. Most of them today are ran off of a blacklist environment. Malware that becomes known to people like what MacAfee, Semantic, F-Secure gets recorded and its signature is then placed inside of a database and it's that database that you're updating. Now because your database gets updated saying that this particular piece of malware is considered threatening, it blocks the process that matches that signature. So let's look at this from a different technology perspective. Anybody out there use a cell phone or a tablet? Well these devices actually work on the basis of a whitelist. You see, when we go to install an application, where do you get it from. Yeah, you can only install applications that come on an approved vendor store. Now this has extended out quite a bit because almost everybody has their own store now, right. Apple their store, Android or Google has their store, I know Samsung has a store, Microsoft has a store, there are all kinds of stores out there. In fact, from a corporate level, we can actually limit in some BYOD environments what applications can be installed or create our own internal stores. Now the downside to this is that most corporations will use a mixture of approaches. If you think whitelisting is the best answer, you're kind of in for a rude awakening because you can actually interfere with the productivity of users at some point and increase support times and costs. And you may be thinking, but Dale how would that happen. Oh, I don't know, maybe a user needs to install a piece of software at short notice, and if you just go after blacklisting, it's actually vulnerable to software that has not previously been identified or what we refer to as 0-day malicious attacks. Now you remember at the beginning of this module I mentioned without encryption we had these issues. Well one of the things that we can do to help secure ourselves down is to use some type of digital certificate. Client machines, as well as our servers can be issued certificates that basically create trusts. Now if you don't have access and one of the things that you will see in your immediate future is a particular product that's called OpenSSL. Now OpenSSL is just simply a software library that's used in applications that need secure communications over our networks and it helps to protect us against eavesdropping. Now OpenSSL contains an open source implementation of both yes, SSL and the TLS protocols. Now I'm not saying this is the perfect solution, in fact, one of the most notorious vulnerabilities is something referred to as heart bleed, but just like anything else, if you don't keep up with service packs and updates, you're opening yourself up to almost any type of attack.

Summary

So what did we learn here? Well, we went through and talked about, one, digital ids and what they represent and how we have them everywhere. We also talked about how endpoints and servers create an attach vector for these identity security issues, as well as services and applications. You know, I just had a thought and that is if you're trying to detect attacks on applications, you're going to see some very interesting things take place like if you see things like new accounts getting created, or unexpected outbound communications, or even memory overflows, those could all be a symptom of your application being attacked. And of course, to sum it all up, we went through and talked about how we can use encryption to help secure things down, in particular, we focused in on OpenSSL because of something you might see in your immediate future.

Other Authentication Methods

Overview of Other Methods

Okay, when it comes to IAM, let's talk about some Other Authentication Methods that we can actually use. In this particular module, we're going to go through and take a look at several options or several things. One of them is going to be starting off with understanding that there are basically just two classes when it comes to IAM. We'll also take a look at some protocols. In particular, we'll focus in on the ones that are important for you for your immediate future. You may hear me use that phrase, immediate future, it's because as a trainer I'm not allowed to tell you what may or may not be on an exam, but you may want to break out the highlighter when I make that phrase, but we'll be looking at both Radius and TACACS+. We'll then go through and talk about some of the problems and solutions for context-based authentication. And then we'll go through and focus in on looking at both single sign-on, or SSO, as well as federations. And then finally, we'll end up this module by taking a look at our self-service password resets. I know you guys all know what those are, but I want to make sure that you understand how we can make sure that we've protected ourselves. Sometimes these particular services actually end up opening some real nightmares for us. So let's get started.

The Two Classes

So we need to make sure that you understand how credentials and authorizations are processed by network servers and clients. Now technically, it's broken down to two classes and I'm going to break this down for you. I'm not going to break out my rap, break it down, but typically, falls under two different classes and the first, or I should say classes of identity repositories. The first one would be just simply known as a directory service. This is basically a service that provides information about systems, users, and other information about the organization. And then of course, we also have AAA. No, that's not the United States company that comes around when you have a flat tire to fix your tire for you. This is authentication authorization and auditing is what it's short for. This is a service and what it means is that it allows us to provide the authentication authorization auditing services for devices, which we'll talk about. But let's first go through and focus in on our directory services. As I mentioned before, the directory services is basically an identity stored database and what it does again is it's going to store information about objects, as well as their credentials and authorization. And because it's in a database, we know that one of the things that we can do with the database is to query it. Now there are several different versions of the directory service that our out there. Probably the most popular one is Windows Active Directory, but there is also OpenLDAP, there's just regular LDAP, which is a protocol, there is Apache DS, eDirectory, as well as OpenDS. Now if I didn't list your favorite, I'm sorry, there is several of them out there, but they all have the basic structure or the same function in the aspect that they typically have a hierarchy that follows a DNS structure, so you might see something like wayne. corp as our directory service hierarchy, and therefore an object would obtain the parent information. So if I had a computer that was called Bat computer, it would be batcomputer. wayne. corp. Now this opposed to using something like AAA, oh man I could have used a battery joke there, that's something that's more global than talking about our car service here. Anyway, AAA it basically again is a way for us to get devices to authenticate our switches, routers, access points, they're all going to connect to an identity store that's used to validate their request, and typically, they use a protocol, such as radius or TACACS+ to this type of authentication.

RADIUS & TRACACS+

So let's talk about those protocols, both RADIUS and TACACS+. Now RADIUS is probably the most common one that we see out there. It's short for Remote Authentication Dial-in User Service. Yay, it's a mouthful, isn't it? Now how RADIUS works is it can actually operate on either TCP or UDP and it's implemented in a client server model. Now when we talk about clients in RADIUS, we're not talking about endpoints like our desktop machines. We're talking about a client could be a device. So typically, how this works is that we go through and we have our Remote Access Server or our RAS server. This is the box that we hit from the outside and it connects to our AAA server, which then can give permission to our resources on the network. So if users are outside of our organization connect in through the internet, they're going to hit that routing remote access server, or that RAS server, and it passes the information back to the AAA server to authenticate them, and if they meet the credentials that are stored on the AAA server, then they're able to gain access to the resource. And the same can be applied to also wireless devices. They can hit our access point, which would then forward them back to the RADIUS server to be authenticated. Now how they're authenticated is that the RADIUS sends passwords that are disguised by a shared secret and an MD5 hash, meaning that the password security isn't that great, and so; therefore, any traffic that we deal with when it comes to RADIUS, we should actually encrypt using something like an IPsec tunnel or any other type of protection because obviously you could capture this shared secret and very quickly discover what the passwords are. Okay, so now that we've got that down, let's move into TACACS+. So when it comes to TACACS+, it's another acronym for us and it just simply is short for Terminal Access Controller Access Control System. Now this is a similar call to RADIUS, but it is designed to actually be a little bit more flexible. In fact, the later version added the + to it because not only did they improve it, they made it so that it would communicate over TCP instead of relying on UDP. Now this protocol is similar to RADIUS, but it's designed to be a little bit more flexible and reliable. It was originally designed or developed by Cisco, but it also supports many other third-party and open source RADIUS server implementations. Where we often see RADIUS being used for VPN implementation. TACACS is often used for authenticating and accessing or administratively accessing our devices, such as routers and switches. And to kind of give you a breakdown of the difference between RADIUS and TACACS+, you can see here that obviously RADIUS is less secure. We talked about how it uses a shared secret versus encrypting our information as we go across, and of course, the UDP versus TCP, which gives us a lot more as far as reliability is concerned. So now that we understand the protocols or the two major protocols, how do we actually secure our repositories? First and foremost, it's imperative that the communication between the client devices, such as your routers, your access points, your switches, as well as the AAA servers themselves are secure and we can implement that by making sure that we isolate the traffic, maybe on a VLAN or a subnet or even using IPSec so that we can prevent the possibility of man-in-the-middle attacks. When it comes to that shared secret that we talked about that RADIUS has, make sure that it's at least 22 random characters and they can be either alphanumeric and/or symbol based. Technically, we can up to 128 characters, so the more the merrier. Now one of the other issues that we have is that the directory services can be subjected to injection attacks, as well as software vulnerabilities. So make sure that you keep up to date as far as patching is concerned, and if you're using intrusion detection systems, make sure they're also up to date. Now as we mentioned earlier, directory services, because they're in a database, can be queried. So one of the things we can do is configure the use of a secure transmission protocol like LDAP over SSL. And one of my favorite targets is what we refer to as service accounts. We typically create these accounts because in order to do anything on a machine, we typically have to do it as a particular user. Well, for example, SQL or even SharePoint, SharePoint needs a couple of different accounts to actually interoperate with Active Directory, as well as the user environment. Now these types of accounts, they've often referred to them as being the keys to the kingdom. On a Windows network, they're the domain admin accounts as well. So one of the things you can do is make sure the network channels and the devices used for administration are isolated away from other uses as well as other users. If you've got desktops and laptops that you plan on using as administration devices to remote into these servers, they should actually be designed to run whitelisted software only and have highly restricted internet access, typically only to the sites that they need for patching or possibly accessing to drivers. Now I know that implementing all these can sometimes be overwhelming, but trust me, in the long run, it's going to make you a happy camper.

Context-based Authentication

Context-based authentication. Now most user accounts are secured by passwords and the problem with this is that password-based security is kind of easy to deploy, but it also has a ton of weaknesses, those being things like social engineering, password cracking, session hijacking, and there are tons more we can go into. Instead, what we can implement is context-based authentication, which allows authentication decisions to be made based off of information about the user, or possibly the system or systems that the user is using, or the time of day that they're doing their work, maybe restricting it based off of a timeframe, or even the group memberships or the roles that are related to the application or service that they're trying to access, or even IP addresses or the reputation of an IP address range. We can also use location based information like, again, their IP address, but more particular, probably GPS locations. Or even device-based, including information about the browser in use or other data that can give us type of fingerprint of what type of device is being used, whether it's its IP address, again, time zones, screen resolution, cookies, or installed fonts and languages. Now me being a white Caucasian dude, my company probably set up my system and they probably set it up with English as being the language with specific fonts, and for a different device to pop-up trying to authenticate as me that may have, I don't know, a different language, different fonts, Arabic, Spanish, whatever that something that I wouldn't normally have could help to protect my resources. Now let me give you an example of this. I'm sure we've all been through this scenario before, but I have a bank that I belong to and, of course, you know all my money goes to the bank, but I'm sitting here with my laptop and when I log into my bank's website, it prompts me for a password and I type my password in and I'm able to access my banking information and I'm able to do this because I've authenticated with this bank before, but I just recently got a new phone and I decided I wanted to gain access to my banking account information through my mobile app. So I made the connection to my bank and I gave them my password, but my phone was denied until I was able to answer some simple questions that I'd already set up ahead of time with my banking institution to verify that it really is me on this particular phone, and once I answered those questions, who is my favorite superhero, Batman, no. If you don't know me by now, you know that none of my passwords have anything to do with Batman or any comic book character, Superman. No, but once I answered those questions that I'd set up ahead of time, my device was then allowed to gain access because they've identified that it really is me accessing this information. Now don't be surprised that some organizations will actually use multiple types of contextual information to help to authenticate their users besides obviously, their passwords, biometrics, again either fingerprints, retinal scans, voiceprints. I just recently read an article that Google and their Google home device will start to recognize six different types of voices. It's so that the NSA can track me better. No, the concept here, and I think we'll eventually get here, is where your voice will be your password. Now the location based authentication is less common because we're a society today that does a lot of traveling, so it gets a little bit harder and harder to not pinpoint where you're at, but to look at your behaviors because I could be on vacation in, oh I don't know, San Lucas. Now there is another mechanism that's used quite a bit and that is SMS. You know when you log into, for example, Dropbox and Dropbox sends you a code and says type in the code that you just received via SMS. Well I want to talk to you here just real quick about that concept. My major gripe with this one is that SMS is not a very secure protocol. In fact, there is a lot of calls for SMS to be deprecated. Now not only have there been successful attacks against SMS based one-time passwords, but there is also a number of ways that it can be successfully targeted. Now thank heavens there is some really cool alternatives that actually exist today that you can use instead of SMS. One of them is, probably the more popular one I would say is Google authenticator. Now does this actually mean that people are going to stop using SMS as a second factor? Probably not, but it's something we need to be aware of.

SSO & Federations

Okay, let's talk about SSO, Single Sign-On, and federations. Now anytime I use the phrase federation, it is required by law that I give you the Vulcan sign for live long and prosper. No, not a good joke. How we actually want to use SSO and federations is in this way here is that we might have a user who has an account inside of their IAM service, Active Directory, in this case, and they need to access a software as a service product that's out there. Well in order to get this to work correctly, what we can do, in the old days, we'd actually have to have a login and password for each one of these devices. Instead, we actually go through and set up kind of a trust between the two. The Active Directory or where the accounts live would be known as the ID provider or IDP, where the application would be the service provider or SP and we create this trust relationship between the two. So what happens is that when the user says let's say that it's a CRM product, when they log onto that CRM product, they're actually redirected to go and verify that their account is still active, and because they're trust relationship exists, the IDP provides a token back to the user and then the user submits that token back to the SP and they're able to use the product, or in this case here, the application. Now typically, the user would have provided their authentication when they log onto the desktop or the laptop. In fact, this is the same integration that we use when we're dealing with Exchange. Exchange uses Active Directory or SharePoint. SharePoint uses Active Directory. Or if you use SharePoint for people outside your organization, again, we can go through and use a RADIUS server to provide the authentication mechanism, and once they have authenticated, they're able to gain access to those pages. Now obviously, the big advantage here is that users don't have to remember all those different passwords, and again, the simplicity comes from that trust that we created. And then guess what, there is actually a couple of technologies that serve as the core for creating this federated identity. The three major ones are SAML, OAuth, and OpenID and each one has its advantages and disadvantages. Now you may not realize this, but we're dealing with this technology every single day, especially if you have a mobile device. We have all these different accounts and passwords that we're trying to keep track of. Well using one of these technologies can give us a way or give identity providers a way to integrate with service providers so that you only have one login. Again, let me give you another example. So I recently picked up biking. I jump on my bike and I don't do road biking and I'm not a mountain bike, I guess more of a path guy, but I love going long distances and one of the apps that I use is an application called Endomondo and they would technically be referred to as my service provider. They are giving me access or they're tracking my bike rides. Now when I go to log into Endomondo, one of my options that I have is it'll redirect me and say hey why don't you use one of your social media accounts and its password, which would be our ID provider or our IDP, and again, like what we saw before because that trust relationship is there, my Twitter account, or my Facebook, or my Google+ account is going to give me a token that I then use to log onto my application, which is actually kind of slick. So what are the differences between these three? Well when it comes to the three main ones here, each one has its own advantages and disadvantages. We see here that with SAML it does authorization, as well as authentication; however, its potential security risk includes denial of services, as well as message confidentiality. It's commonly used in the enterprise authentication and authorization environment, particularly in Linux type environments. When it comes to OpenID, it doesn't do authorization, but it does do authentication and its potential security risk includes being susceptible to a cross-site request forgery, as well as a cross-site scripting attack, and of course, phishing. And most commonly, we just use it for authentication. Typically, we'll actually combine OpenID and OAuth together because OAuth provides an authorization, but it only does a partial authentication, which is where combining it with OpenID helps to provide a better solution for you. It's potential security risk include redirection, message confidentiality, and being able to impersonate a resource server. Now typically we see OAuth, and when I say OAuth, it could be OAuth 2 as well. We're going to see this being used with API and service authorization. So when we look at this environment, the question usually comes up as what are our security issues. Well the security issues are based off of or I should say the incident response is based off of all the players involved. Our identity providers are typically responsible for notifying users, as well as notifying relaying parties. Service providers need to determine what their response will be if an identity provider were to be compromised, and of course, the users responsible for obviously keeping track of his passwords. Now speaking of the user, you also have to consider what the impact would be if your account is inaccessible or if it's been compromised. How many different accounts could somebody gain access to. I mean, imagine if your Google account got compromised or if Google somehow became unavailable, what would you do. I'd start using Bing. No, different subject.

Self-service Password Reset

Aww, the self-service password reset. Yes, I like to call it the IT headache. You know the typical user that goes and changes their password or forgets their password after they've changed it, they then have to make a phone call to you and you take that one user and you multiply it by thousands of users each day, it equals an IT person who gets crazy hair or just gets really, really frustrated. So instead, one of the things that we can do is implement the password reset. Now we can typically do this via a couple different mechanisms or different ways. One of them is we could issue a temporary password and they have to change it the next time they log in or we can go through and allow the user to request and select a new password. Now there are a number of methods that we could do this, but the two most popular is doing either a challenge question, which again, might be something like what was your first school you went to or what's your pet's name. Unfortunately, attackers who know what they're doing and we see this all the time are able to typically guess what these answers are. Now if I'm presented with this is my only option, I actually kind of play around with this a bit. What I do is I come up with a pattern or I've got a particular pattern in my tiny little brain and that is if it asks me for my favorite pet's name, I actually use an answer to one of the other questions, like for example, what was my first grade teacher's name, Mrs. Johnson, that would end up being my pet's name. That might be a little bit more difficult or adds a level of complexity for the attacker to try to figure out what's going on. The other mechanism that we could use is referred to as the two-step verification. We also refer to this as the cowboy dance, no, or the guy who can't dance, we just do the two-step. I grew up in Hurricane, Utah, and we referred to it as the Hurricane two-step, that's where you shuffle your feet from left to right. You're rocking back and forth and your dance partner is just going crazy. Had a flashback to my childhood there. You know, high school. It was only five years ago for me. No. Two-step verification, the user adds a secondary communication channel, such as an email address or we talked about SMS earlier, so the user has to enter in a temporary password that's generated or a code that was sent to them via either that email or SMS message. Now I'm going to sidetrack here just for second. I'd like to see if maybe we can talk adult to adult and here it is, is that no matter what we do, we are so used to saying this is my computer and what I do on it is my business. I saw an interesting video the other day. It was an old, old video from the 1970s where IBM was talking or doing kind of a tutorial to a gentleman to explain to them how the keyboard worked on a computer and they made a really profound statement there that I think maybe we forget and what they said was this, the teacher was saying okay if you want to run the program, you type the word load. I know, this sounds really old, right. And so, the guy types in load and says okay what do I do next. And she says, you're going to find and press the Enter key and what the Enter key represents is I'm turning it over to you, meaning turning it over to the computer. Now think about that for a second. When we type something on our keyboard and we hit Enter, we are turning it over to the computer, and especially if we're hooked up to the internet, and that information goes across as 1s and 0s and anybody can pick this information up whether it's a retail location because you're shopping online or possibly a government agency, as well as yes, attackers. So don't be shocked if and when your account gets hijacked because you're accessing information on a public network. Think about that one. It's not your network. It's not my network. It is a public network, and knowing that, hopefully will help you think a little bit differently when it comes to protecting your credentials. Okay, I'll get off my soapbox now.

Summary

Okay, so in this module, we went through and first took a look at the two classes that are involved with authentication authorization. Remember, we had either directory services or we had AAA. We also talked about both RADIUS and TACACS+, which are the protocols that we use in authentication and authorization. We also talked about context-based authentication, again, authenticating based off of some type of context, be it location, be it the device that we're using, or even the time of day. We also talked about the single sign-on and federation services that we can use to make it not only easier, but also to create a seamless environment as we access different resources. And then of course, we went through and talked about self-service password resets, which is a great way of helping out the IT guy to make sure that he doesn't get that crazy hairdo. Okay, in our next module, we're going to go through and talk about identity repositories and how attackers go after them.

Identity Repositories

Module Overview

Okay, so in this module, we're going to take a look at the Identity Repositories and how attackers might come after them. We'll first take a look at what I refer to as the six areas that are potential attack vectors. Then we'll get into some detail and talk about how attackers go after LDAP, as well as going after OAuth and OpenId. And because a lot of authentication mechanisms today deal with Kerberos, we'll take a look at the different attack vectors that it presents, as well as the attack vectors in RADIUS, and of course, since most of our products are Windows based, we'll look at going after Active Directory. We'll then go through and finally take a look at how we can stop some common exploits from taking place on our network. So let's get going by taking a look at the six areas.

The Six Areas

So when it comes to the six areas, these are the six areas in my book where security issues can pop up on us, as far as identities are concerned. But before we do that, we need to understand that identity threats can be broadly classified into a handful of areas. First, there's the threat to the underlying authentication and authorization systems, you know, looking at the vulnerabilities and the way that users log in, or how they're credentials are handled, or how they're authorized. The second would include how attackers may actually go after the lifecycle of our credentials preventing them from being removed or elevating their credentials so that they have more access since they're utilizing that particular account. And the third would be focusing on accounts themselves, either through phishing or doing some other type of compromise on the system where again their credentials are being stored. Okay, so let's dive a little deeper here and take a look at these famous Dale's top six countdown. So here are the top six or the six areas that you're going to really want to pay attention to when it comes to taking your CSA+ exam, or I should say in your immediate future. Let's first start with personnel-based identity security, which includes training and awareness, continual training and continual awareness, as well as threats like social engineering, phishing attacks, or even just the straight up insider attack. You also have your endpoints and their roles in the attacks on identities. Screen captures, keyboard capture apps like loggers, local admin rights, where the passwords and how the password are stored, any tokens that may be on these devices, and even though I have a picture here of computers, please, please make sure you understand that this could also include mobile devices like tablets and phones. We also have server based exploits, which can target the system that's running the identity services or the AAA services, whatever you're using and that would include any virtual machines that you may have installed on that post. We also have to worry about our software and our applications and services that provide and they end up consuming and interacting with these identity systems. And of course, we also have to look at roles, rights, and permissions that are associated to groups or to users themselves. All six of these can actually help you as far as preparation for the exam coming up or in your immediate futures. Remember you need to consider identity security issues from each of these viewpoints, and what I mean by that is let's say that you in your immediate future get a question concerning phishing, you need to look at the aspect of well how do I fight that. Well that's going to be with training, email filtering, or even two-factor authentication. So now that we got the top six down, let's dive into how attackers go after each one of these IAM solutions.

Going After LDAP

So when it comes to looking at going after LDAP, remember that's the lightweight directory access protocol. It's used by hundreds of organizations or almost every organization for both authentication and directory information, which makes them basically argh, the motherlode. Ah, must be pirate day. But in all joking aside, this does make them a target for attackers who want to try to get the organization information or access to systems that rely on LDAP-based authentication. Now typically, attacks against LDAP will be focused in on primarily on looking at the unsecure connection methods that we use between the systems as they talk to each other. Again, if we're not using some type of encryption, we can actually open ourselves up to some pretty serious issues. Another mechanism or another focus would be the LDAP access controls that allow attackers to gather information or that would allow them to make modifications to some of the directory entries that they normally shouldn't have access to. We also have the issue just like any database, some type of injection attack, which allows attackers to gather additional information or again to make changes that they wouldn't be able to do normally, and typically, they'll exploit your web app that's built for LDAP. And finally, we have the denial of service attacks, which can stop authentication services from doing what they're supposed to be doing and create some issues for us. Now each of these attacks can actually be prevented if you just take some time and carefully design and implement your LDAP services and access methods making sure that we require secure binding methods, set the appropriate access controls, and verify them to make sure that they're working because changes are taking place all the time. Also, make sure that you use a good web application development practice and design a scalable LDAP directory service that can help us reduce the likelihood of having any time of issues with LDAP. Now believe it or not, LDAP can actually be used as an attack tool. You're like say what? Back in 2016, the connectionless LDAP services or CLDAP, which means we don't really care it's using UDP, we don't care necessarily for a response, we can just send queries. Well attackers could actually spoof an address, which would result in an amplification rate of up to 55 times higher than normal traffic. This basically creates a denial of service attack on your system.

Going After Oauth/OpenID

So how do attackers go after OAuth and OpenID? Well, that's pretty easy. And the reason why it's easy is because we are relying on each individual provider and making sure they are taking care of business or making sure that they have secured things down. Normally, we as users, we use our computers and we gain access via these providers. Well one of the most common attacks is based off the fact that some providers aren't very careful and they leave themselves open to what we refer to as open redirects. When redirects and forwarders aren't validated, untrusted user input could actually be sent to a relaying web app, which means that the user gets redirected to an untrusted site and allows for phishing scams or permits the attackers to bypass the security layers that are in place. Now because OAuth has a broad adoption for cloud services and mobile apps, it makes it a really fun target for attackers. Again, poor session management and reliance on a central shared secret file for OAuth servers, as well some companies even doing plain text sessions can create a lot of issues for us. Now most of the attacks against OpenID have been aimed at the protocol vulnerability, which was discovered in 2012. This gave the ability for attackers to forge an OpenID request, which would result in the relaying parties to allow arbitrary logins to their services. It was really kind of scary. Now we talked about earlier when we saw that sheet or the layout of the difference between them. We talked about how OAuth and OpenID can be used in conjunction to make it a little bit more secure because one is not necessarily better than the other, but combining them together gives us a better environment. One thing that we have to be worried about is the fact that OAuth2 is susceptible to what they refer to as CSRF or Cross-Site Request Forgery attacks, so the attacker can actually forge a request, let's say to transfer some money. He takes that forge request and embeds it into a hyperlink and then sends it to victims who may log onto that particular site and typically, he'll do that through an email message. What happens is the victim clicks on that particular link and inadvertently sends a request to that website and the website validates the request and transfers the money over to the attacker's account. So if you want to do me a favor and click on that dollar sign above the bank, trust me, nothing will happen.

Going After Kerberos

So going after Kerberos, well what is Kerberos. Well it's the three-headed dog that guards the gates of Hell according to Greek mythology or Fluffy if you're a Harry Potter fan. Nah, Kerberos actually relies on a central key distribution center known as a KDC, and if an attacker can compromise a KDC, it allows the attacker to basically be anybody they want to be. Now some of the most common attacks against Kerberos include yes, going after the famous administrative account attacks. Normally, administrative accounts on Windows boxes don't get locked out, and depending on your password policy and a little social engineering, I might be able to figure out what the password is. Then there is the Kerberos ticket reuse, which includes pass the ticket attacks, which basically allows the attacker to impersonate any legitimate user. And then there is also the ticket granting attacks, or the TGT focused attacks. If we're able to get a hold of these TGTs, they're often referred to as the golden ticket because they allow attackers complete access to the Kerberos connected systems. This includes the ability to create new tickets, do account changes, and even create their own back doors. So how do we protect ourselves? Well by implementing behavioral based and automated monitoring so we can detect if someone has created a golden ticket. A normal ticket will actually have a time span or a life span where a golden ticket, the attacker typically will give themselves months or years before the ticket expires, so again, monitoring that will actually help us out quite a bit. And now, because I'm talking about golden tickets, I have some Willy Wonka music going on in my head because I've got a golden ticket.

Going After RADIUS

Okay, so now let's go after RADIUS. Again, we know that RADIUS is commonly used for authenticating of network devices, as well as we use it for VPN solutions. Now this makes it extremely tempting for attackers who want to try to get in from the outside. Now typically, RADIUS attacks will focus in on session replays in which the attacker just simply replays a previous response or replays the server's response to authenticate a client without valid credentials. Attackers will also go through and target the shared secret of RADIUS. Remember, we talked earlier about the RADIUS shared secret that you should at least make it more or at least 22 characters, please, please go more because again it's a fixed secret, which makes it easier for attackers to determine what it is. There is also the aspect or the attack of going after RADIUS with denial of service, which obviously will prevent users from authenticating. Oh, I should mention, back with that shared secret, that doing simple brute force attacks against that shared secret, could glean quite a bit of information. So how do we protect ourselves? Well we can start by using TLS to protect the RADIUS authentication, instead of trying to utilize the built-in protections of RADIUS itself, encrypt the traffic in between. Don't just rely on that shared secret, please. So now that you know how to protect yourselves, I'm going to give you my RADIUS joke, you ready? What's a difference between a diameter and a RADIUS? Yes, a RADIUS. Thank you. Thank you. I'll be here all week.

Going After Active Directory

So how do attackers go after Active Directory? Well first of all, we probably are familiar with Active Directory in the aspect that it is basically the big, bad daddy of the Windows environment. It's not only an identity store, but also a AAA service for many organizations and that makes it extremely popular as an attack vector for the bad guys, and because of the fact that we typically will have multiple domain controllers that are handling Active Directory makes it hard to kind of track because you could have a domain controller in a remote location. Well, who's monitoring that particular remote location? Now the most common Active Directory attacks include first of all, malware. Now malware focused attacks that actually seek to grab credentials or even using exploit based malware to try to get into these systems. Stealing credentials via phishing attacks is another popular attack, again, very easy to do. Most phishing attacks result in about a 95% success rate, which makes it why it's so popular. Another type of attack is referred to as the privilege escalation attack where we take advantage of exploits from Windows and try to elevate the permissions of a particular account. Now one of my favorite ones to go after is actually the service accounts, which typically we install the service accounts or configure them for I don't know, again, SQL, SharePoint, System Center. We create these accounts and we typically don't come back to them, right. They're forgotten. And finally, we have the issue of domain administrator rights that are given to people that don't necessarily need all of those rights, which basically provides more attack vectors to go after or more targets to attack, and the likelihood that the Active Directory admin account won't be maintained or that its password will end up getting exposed. The other issue that we have here is that typically Windows domains include older systems. I don't know how many times I go and do a pin test and I see that companies are still using an XP machine because a particular app will only run on XP. There are also some default behaviors of Active Directory and Group Policies that make Active Directory an extremely fun target to go after. Group Policy preferences can be used to perform all kinds of actions like mapping drives or printers, but unfortunately, they also store the credentials they use in XML files and those are stored in a SysVol folder, which is shared to every domain controller, so Dale make the scary man go away, how do we protect ourselves? Well that's easy. It's done with continual training and password policies. Password policies, as in how long the password is, how often we're changing it, as well as the complexity that we require. So hopefully, you're not still scared.

Stopping Common Exploits

Okay, now let's talk about stopping common exploits. We're going to first start off with, do you remember the lifecycle that we talked about. Well if we don't follow it to the letter, we end up creating some attack vectors, things like unused accounts where the attacker can actually compromise and use without anybody ever noticing because it's not being used. Also accounts that aren't properly deprovisioned or they're left on systems because somebody missed it during a normal account removal or an end of life process. I know, that leads to abandonment issues, right. But it's real. Accounts that aren't properly deleted can often indicate that you've got some issues with a management of a system and it may not be logged or be monitored. And of course, another one of my favorites is the group membership issue. Attackers, particularly attackers from the inside, may try to leverage rights that they or others have accrued over time without knowing that they still have them, so we need to be again reviewing group membership permissions all the time. Okay, we then need to make sure that we maintain our rock and roll. Oh no, sorry, rights and rolls. There we go. Now normally, user accounts are managed by using the principle of what we refer to as least privilege, which basically means that the user should be provided with only the amount of privileges or permissions that are required to do their job. This actually helps so in case an attacker gets a hold of that account from doing any action that they wouldn't normally be able to do anyway. Now the one thing you have to be careful about is something called privilege creep. Now privilege creep can be extremely hard to track, new managers may not be aware of old user's rights or somebody changes responsibility, it gets really difficult to kind of track that, but we should be able to deploy software, there is some specialized software out there, that will help us in monitoring and managing these accounts and their permissions. One of my favorites is called cat creep. No, actually I did the required have a funny kitten photo in your presentation, as well as I couldn't find an image that represented privilege creep, so there you go. It is what it is. Now there is a couple steps that we can actually do to control the method of this madness or basically to protect us from different types of attacks against our identity in access management systems. We can first start off by protecting ourselves against impersonation attacks. Again, this is where the attacker will take the identity of a valid account. Now typically to stop impersonation attacks would require stronger session handling techniques and that's a discussion for another course, but I'll give you some resources here. If you go look at the OWASP session cheat sheet, just Google that, there are several different steps that you can go through to stop impersonation from taking place. The other is handling man-in-the-middle attacks. We stop that by implementing end-to-end encryption of a session or network links. Now the only way the attacker can still get to us is if they control one of the endpoints or have the encryption keys, but that's a little bit more difficult. We also have session hijacking. Again, this is focusing on taking over an existing session that's taking place, either by grabbing the session key or a cookie that's being used by the remote server. Again, just like our man-in-the-middle attacks, if we secure or encrypt the data, we can limit ourselves, as far as an exposure for session hijacking. Privilege escalation, we've talked about this. Again, this is where we elevate somebody else's permission or we create an account and elevate that permission up. Typically, the attacker will rely on vulnerabilities in software. So making sure that you keep up on your patched applications or making sure your services are configured correctly will help stop this type of an attack. We also have the famous rootkit attacks. My first comment here is that if you ever suspect a system has been attacked by a rootkit, format the drive. Don't try to save it. So get your data off, but the drive itself, you need to start from scratch. Fighting rootkits requires a complete suite of security practices ranging from anything from proper patching to whitelisting or even having your antivirus. Most antivirus solutions today will actually detect rootkits. Okay, and finally, I want to talk to you about passwords. I know, again, but obviously we wouldn't have to continue to talk about them if they weren't an issue for us. I get it. We've talked about this several times, as far as you're having multiple accounts with different passwords and you don't want to recycle your password, and so, you end up having a plethora, ah you didn't think I'd get through a course without talking about plethora, did you, but we end up with a plethora of passwords to try to track. Phishing attacks are always trying to trick our users, and as well as ourselves, from entering our username and passwords, and of course, we don't want to write the passwords down anywhere. So, please do me a favor and get a password manager. I'm not going to tell you which particular product to use out there, you can do your own research. But basically with a password manager all we're doing is we have a database with a master password. Now that master password, so help me if you guys go through and get a password manager and your master password is less then 8 characters long, I will hunt you down. You are defeating the whole purpose here. Like having a master password that's weak, but this database is going to keep track of all your different passwords, in fact, I'll be honest with you. I don't know my passwords, to some of the websites I go to. Because, my password manager creates them for me and saves them. And then when I need to get to that site, I just simply click on a button. It makes it really easy, but I do have an extremly long master password.

Summary

Okay, in this module, we went through and took a look at a lot of stuff here. We talked about the six areas that identities can be susceptible to. We also talked about going after LDAP. Remember, our protection is using secure binding methods, as well as setting up some access controls and then verifying them. We then talked about how attackers go after OAuth, as well as OpenID. Again, there is not a lot that you can do to protect yourself because these technologies are implemented at the per provider basis. You can check out the provider, and obviously, you are one of those keys as far as being the user is concerned. We then talked about going after Kerberos, especially when he gets off the leash, you have to run really fast. Okay, I swear that's the last joke of this module. Oh no, that was one too. Remember, we were talking that the administrator account is attacked quite often and the golden ticket, yeah you're going to start singing that song again, aren't you. I've got a golden ticket. For you, young'uns, go YouTube that, Willy Wonka Golden Ticket song. We then talked about going after RADIUS, and again, remember our protection our protection here is using TLS or some type of encryption during the transmission between the RADIUS client and the RADIUS server. And of course, we talked about going after the mother lode itself and that is Active Directory. If you remember, we talked that malware focused attacks, as well as privilege escalation attacks are extremely popular, and those wonderful service accounts. And then we ended up talking about stopping common exploits, things like making sure that we follow that lifecycle for accounts, as well as rights and rolls. Remember, using least privilege, as well as avoiding the creepy little cat, oh no, privilege creeping. I lied, that was the last joke of the module. So next up, we're finally going to get some hands on. The next module, we're going to go through and show you how to build up a lab that you get to play around with. And then in the following module, we'll go through and actually do some evil work.

Building the Lab

Module Overview

Okay, this is exciting stuff, and what I mean by that is building the lab, this particular module, I want you to kind of flag this one because we'll be able to use these labs or the setup here that we're about to show you in multiple courses, including courses maybe that you're not even watching that belong to me. But if you watch anybody else's courses, I'll actually have to hunt you down. So that's what we want to get into is showing you how to build this up. It is extremely easy and effective. So in this module, we're going to go through and make sure that you understand the Pluralsight Autolab setup. We'll also go through and talk about the lab structure, what it does for you. It's really quite amazing. It's going to build up four machines for you automatically. We'll then take a look at the Autolab requirements, as well as the aspect of some of you guys may decide to do nested virtualization. This is where we have maybe a core server or a core box that's running Hyper-V on it and we fire up an actual server inside of Hyper-V and then we load the virtualization inside the virtualized box. We'll then go through and we'll just talk about how to deploy the Autolab. I'm going to step you guys through this all the way. Trust me. This is so easy, it's not even funny. And then of course, we'll take a look at after we've deployed the Autolab on how to use the Autolab. Sounds like I'm talking about an auto shop course here, doesn't it, Autolab. We'll go through and change the oil. No, not really. So let's jump in because this is way exciting.

Preparing the AutoLab

Okay, so let's start first by looking at the Autolab itself, what it's designed to do. I have to give some real big kudos out there for a couple of Pluralsight authors that have worked very, very hard in creating this environment. Jason Helmick is probably the big guy that's heading this up, but I know a lot of authors involved with this. I think Orin Thomas is involved, Greg Shields, several of them out there and if I forgot their names, I apologize. But what it's designed to do is it's designed to create a playground for us with very simple interactions. So it's going to go through and build up a lab and it builds up like four different machines, which is kind of cool. And as they come out with more and more courses, Pluralsight's focus will be to add additional configurations, so maybe one down the road might fire up just two machines or maybe five machines. What it's designed to do then is since it creates this playground that we get to fiddle around in is it allows us to go through some of our lab steps, and in the case of security, it allows us to actually do some attacks and test things without affecting our production environment and we all know, at least, I'm a big fan of doing something, instead of just watching it being done and that's what I want to do is make sure that you guys have an environment that you get to play around with and hopefully that information will solidify a little bit more for you. So as far as the Autolab is concerned, there are some requirements here, but to give you a little bit of background before we get into the requirements, you just need to know that it's going to use a combination of Hyper-V, PowerShell, and a specific version of PowerShell, a desired state configuration, and a technology called Pester, which it goes through and based off of a single script, it goes through and creates this whole lab environment for us. Now when I say it creates it for us, I mean it really creates it. You don't have to worry about going off and trying to find ISOs or software updates. It's going to download all the stuff for us automatically, which yes, means you're going to have to have some hard drive space available, but what's really cool is that after it downloads all those things, it goes through and builds up that lab for you, and what you can do is you can not only build the lab, but you can reset it and you can actually wipe it very, very quickly. The initial setup of the lab environment itself is time consuming, but once you have all the ISOs and all the updates downloaded, restarting the lab up is extremely fast, and of course, the best part is that there is very little interaction for you to do. I'm going to demo this for you and you're going to go holy cow, that's way cool. I'm going to have to do some time lapse stuff because some of the downloads, for example, the ISOs do take a little while and I don't want to just kill time and have you watch me download, oh I don't know, a couple gig ISO file. So let's talk about the requirements. Well, the requirements are quite easy. I would recommend that your host machine has at least 16 GB of RAM. The virtual machines in this particular lab environment are going to take up about 8 GB of RAM and I always want to make sure that you have enough available for the host OS, but if you have 16 GB of RAM, you definitely can be planning for the future. Also, you'll need to make sure, I mentioned the hard drive space, you'll need to have at least about 100 GB. It may grow and shrink depending on which lab configurations you download. You'll also need to have Hyper-V and it's a specific version of Hyper-V. It's one of the newer versions that's going to come with our newer platforms and I'll talk about that here in just a second. It does require you have to have PowerShell version 5. 0 and here comes that just that second. In order to have PowerShell version 5. 0, you can make sure that your current platforms are updated, but the easiest thing to do is to run this off of Server 2016 and/or Windows 10. Now it can't be the Home Edition of Windows 10, it's got to be an Enterprise edition, at least. Now you can based off of using this with Server 2016 or Windows 10, you could actually implement nested virtualization where again, maybe you've got a Server 2016 box that's just core related or core installed and you can install a version of Server 2016 that has the full GUI and then install the lab on side of that box itself and that way there you can actually make it so that the virtualization doesn't interfere with your networking components that Hyper-V, or that I should say you may have set up with Hyper-V. So again, you can do that, but the nested virtualization does require you to have, again, a newer version of the Windows platform. So let's talk about how to prepare for the Autolab. Well, we're going to first go through, we've posted all this stuff up on GitHub and I'll pull up the link here for you. It's github. com/thejasonHelmick and then his repository is just called PS-AutoLab-Environment, or Env. Now when you download this, before you do anything, you're going to want to make sure that you set the execution policy to be bypassed because the default for PowerShell is that all PowerShell scripts have to be signed and these are not digitally signed. After we've set it up for bypass, we're just simply going to run a command called Setup-Host. ps1 and please make a note of the syntax here. Because of how PowerShell works, you have to specify the path here, in this case here, it's going to be. \Setup-Host. ps1. Afterwards, you're going to see a screen pop up, and trust me, don't try to keep up with me right now. I'm going to demo this and we'll walk through this together so you make sure you don't make any mistakes, but you'll get a nice little interface and it's going to tell you what it's going to do. It's going to tell you that it's installing Hyper-V and that it's going to download certain files for you.

Installing the AutoLab

So enough talk Dale. Let's demo. Okay. Let me show you how this is done. Okay, so here've I've got a default installation of Windows 10 Enterprise Edition. This is actually, I'm remoting into this machine. It's going to be eventually my movie server, so it's my BatPlex. But you'll notice that it is running Windows 10 Enterprise edition. I've got about 32 GB of RAM in this box and I've got oh let's see here a solid-state drive that's about 500 GB and then I've got some, I've got a storage space that I'm working with that's going to be for my movies. But that's outside the course. So what I'm going to do is I'm going to use this as a demo to fire up so you can see how easy this is and I'll be using this machine also for future demos. So I mentioned earlier that we're going to have to go download a particular product, oh let me I guess I should show you this as well is that if I come into this box here and let's look at programs and features, you'll notice that yeah, I've got a couple programs installed here, but under my Features section, you'll notice that Hyper-V is not installed and what's interesting is this script will actually go through and do this for me. So let's open up our browser here and I'm going to come in here and go to the github. com/theJasonHelmick and then the subdirectory of PS-AutoLab-Env and Jason has done a really good job of documenting what we need to be doing, we'll actually be going through these instructions, and you'll be able to come in here and see, for example, these are the different configurations that as we create newer ones for different courses, you'll just be able to download them from here and copy them into your Autolab folder. But for now, we're just going to go ahead and select, we'd like to clone or download and we're going to just download the zip file and I'm going to do a Save as and I'm going to put it in my Downloads directory and hit Save and we're going to let that download. Oh, it's already done. So let's go ahead and minimize this and we'll open up our File Explorer. I'm going to come into my Downloads section here, and in here, I should have my PS-AutoLab-Env-master and I'm going to just simply go through and extract the files that are inside this zip file, so I want to extract it just into a directory called the same name of the file as the directory and then I'm going to take this, and I'm going to cut it, and I'm going to place it in the root of the C drive. So I'm just going to paste it right there. I'm also going to, by the way, if you drill into here, you'll see that it's basically got the same structure that we saw on the GitHub. It has the configuration, the standalone, the tools, as well as some of the readme files and there is the Setup-Host1. ps1 file, which you'll notice it's right here as well. So you don't have to go through and download each of those directories from the GitHub. That zip file includes everything. But I do want to make one more change and that is I want to get rid of this master. I don't know why that's been added. We're just going to leave it here as environment. I guess it really doesn't matter because it's going to create its own directory during the installation. So now that we've got that done, the next thing we need to do is we need to open up PowerShell, but we need to open up PowerShell as an administrator. So I'm going to come down here to my Start button, I'm going to click on it and just simply type in power, and we'll start it off with Shell with an S, and I'm going to not click on, not left-click this option, I'm going to right-click on it, and select to run it as an administrator. So I'm going to hit Yes that I know that I'm making changes because PowerShell can make changes, and I've changed my background and my color of my text so it's a little bit more visible. I refer to it as the batman scheme, and from here, let me change directories and I'm going to change into the PS-AutoLab-Env, we'll clear the screen, and I need to first set the execution policy. Remember, the default for PowerShell, in fact, if I do a get-execution policy, you can see that it's set for restricted, which means we don't allow any scripts to run unless they've been digitally signed. Well these haven't been digitally signed, so I'm going to just tell it to set the execution policy for bypass and it says are you sure that you want to do this because scripts can be really damaging. I'm going to go ahead and hit Yes and now I'll clear the screen. And now, I'm just going to simply type in the setup host. Now don't just type in Setup-Host. ps1 without that. /. Now how I did it was I just typed in set and it's looking inside of the directory for anything that starts with set, and if I hit the Tab, it automatically fills in the Set-Host. ps1. I'm going to go ahead and hit Enter. It says the default location for installation is the Autolab directory on the C drive. This is where everything is going to get installed. The first directory we made was just the installation of the application. Now we're going to install the full Autolab and the C drive is fine. You can change the path if you'd like. I'm going to go ahead and hit Yes, which I should have hit no. Oh man. So let's see if I can manually, there we go. See even guys that think they know what they're doing goof up some of the time. Let's try that again. Let's do a setup. Do I want to change the default path? Heck no. Okay, so now it's telling me that it's going to go through and make sure that I have all of the PowerShell cmdlets that I need, it's going to install Hyper-V, it's going to go through and copy configurations and resources from that directory that from the zip file it's going to grab these configurations and these resources and place them in the appropriate place. It tells you here that you may have to reboot, and that's definitely going to be the issue because of the fact that Hyper-V is going to be installed and Hyper-V installation requires a reboot. So I'm going to go ahead and hit Enter and we're going to let the hamsters run around the wheel for a second and it's opening up WinRM so we can get in remotely as well, it's setting the trusted hosts so that your remote commands work, and here's where it's asking for a NuGet provider and this is one of the requirements for this Autolab feature. So I'm going to go ahead and hit Yes for this one. It tells me it's going to install it inside of my program files, package management, and you can see it's installing. It's now installing Hyper-V, it also shows you the paths that it's going to use for the configuration files, for the VM drives, for the hotfixes, for the ISOs. You'll notice those will all be inside of the Autolab directory and it's asking me do you want to reboot now. We'll go ahead and hit yes because we need to reboot and my machine is going to restart. Now again, I've remoted in to this box, so we're going to give it just a second to finish its reboot sequence. Okay, let's see if we can't get back into this box, and looks like I'm logging in. Okay, so now that we're back, let's go ahead and just double check here. I'm going to go into my programs and features. You'll notice that Hyper-V is yes, installed now, which is kind of cool. But that's not where it all ends. So we're going to jump back into PowerShell here by again typing in PowerShell and I'm going to right-click on it, make sure I go in as an administrator, and yes, I know I'm about to make changes. Now let's go ahead and go back into our directory for AutoLab because remember, that was the directory that was created during installation, and if I do a directory here, you'll notice that I've got those folders, configuration, hotfixes, ISOs, masterVirtual disks, resources, and VMVirtual disks. If I go into, for example, ISOs, there is nothing in here because it's just set up the configuration for us. We actually have to launch the configuration that we want to use, and in order to do that, we're going into the configuration folder, and if we do a ls here, you'll notice that I've got several of them. Now you can play around with different ones here, but the one I'm going to really focus in on is this one here. It's called POC-MultiRole. It's a proof of concept multirole environment. But we're going to pause here because I need to talk to you a little bit more about the Autolab setup and what we're going to be doing from this point forward.

Deploying the AutoLab

Okay, so we've installed the Autolab, but we haven't really deployed it at this point. As I mentioned, we're going to go into the directory of POC-Multirole and that's going to do a couple things for us. I'll explain the different virtual machines that this particular configuration does for us, but how we implement this is we'll go in and type in Unattended-Lab. This is a PowerShell cmdlet that was created when we did the installation of the Autolab environment and what it's going to go through and do is based off of the systems that the configuration says that it supports, it's going to go off and grab all the ISOs that it needs. It'll grab the ISO for Server 2016, for Windows 10. Now there is something here you need to note and that is obviously when you go to log in, we've already created an administrative account and it has its own password and the password is the same on all the boxes and you'll want to make a note of that. And the default password for the VMs is P@ssw0rd. You can change those if you want, but just a heads up, if we wipe this lab and reimplement it, it's going to go back to this password. So let's go in and take a look at how we actually accomplish these steps. This will take probably, this is the longest portion of the lab setup because again, we're going to be going off and downloading the ISOs from Microsoft. Now this step is going to go through and set up four of the machines that we use. We're going to be dealing with Kali Linux a little differently, but since we have Hyper-V installed, we'll just simply download the DVD for it a little bit later and bring it into our network environment. So let's jump out here and take a look. Okay, so we're back here to our box and I'm going to go ahead and switch here into the POC directory. If I do a ls, you'll notice again there's a couple of files that are in here, including some PowerShell scripts, but in order to set up the lab environment, we have to first go through and type in Unattend-Lab and hit Tab and you'll notice that the U and L got capitalized. That's actually one way that you can tell that you've got your syntax correct on your PowerShell cmdlet. I'm going to go ahead and hit Enter. It tells me here that I might have to use NuGet. And so, if it prompts me, I'm going to want to hit yes again. But you can see here it's going through and doing a whole bunch of stuff for us automatically. So there are the Mof files based off of our desired state configurations. You can see there is one for our DC, fir SL1, excuse me, for S1, N1, and CLi1, which is our client machine. It's now downloading the ISOs. I can't tell which this 1 is for, but it's 5 GB, it's probably one of the server platforms. So with the magic of film editing, I'm going to just fast forward through this. If you have any errors, you'll notice that the descriptions or part of the description here actually went through and told you that if you had some errors, it's underneath this layout right now, but it mentioned that if you have any errors, this is what you should type in and what you should be looking for. So with that being said, I'm going to go ahead and pause my video and I'll come back to you after it's done installing. I guess I should say before I leave though, if you have any problems, please feel free to post your questions concerning Autolab at the GitHub Issues tab. So I'm going to pause.

Lab Structure & New PowerShell Cmdlets

As far as the lab structure is concerned, what we're doing or what this is doing is going through and building up a couple of servers. The first one is going to be called DC1 and it's going to be running on Server Core 2016 and it's going to be an eval edition. The domain that it is scripted to create will be called Company. pri. Now the reason why we're using a core box here is because we're following Microsoft's standard and that is most of your servers should be core related or core installed and we manage everything from a client machine. We're also going to get a couple of member servers. One of them is going to be called S1, which is also running a version of Server 2016 Core edition, also it's set up for eval. Evals simply mean that you'll have about 30 to 60 days as I recall, it's one of those, to play around with the product before you have to type in a product key. And remember, this again is a member server and it's joined to the domain automatically. The other box that it's going to fire up is 1 called N1. This is actually a standalone Nano Server, which is new to the 2016 platform. Now we may or may not use these machines. I'm going to be looking at these machines as targets that I'm going to go after and the platform that I can use for targeting this box will be the other machine that it creates, the fourth machine, which is called Client1, and that is a Windows 10 Enterprise Edition, obviously it's in eval mode, and it is a member of the domain, again, the domain being company. primary. Now the other thing that the setup file or the setup program is going do is it's going to install some PowerShell cmdlets. We already know about the Unattend-Lab cmdlet, but we're also going to get a Setup-Lab. Gee, I bet you can't figure out what that does. Yeah, it's going to set up the lab for us. There is also a command to help set up the network environment, so it'll enable the internet on these virtual machines and it's a command that's just, duh, Enable-Internet. When you're ready to start the lab, what do you think you're going to type. Yep, Run-Lab and that starts the lab and applies the configurations for the first time. The other cmdlet that we're going to get out of this is a cmdlet called a Validate-Lab. Now that's exactly what it sounds like it's going to do. It's going to validate the configurations and they've actually converged for us. Now here is what's important is that if you try typing these commands in any directory outside of that configurations\POC-Multirole directory, it's not going to work, so you need to make sure that you're in that directory. Let me talk to you guys about some of the PowerShell cmdlets that are going to be installed with this lab environment. The first one is called Shutdown-Lab, yeah can you guess what that does. Actually, it's quite powerful. It not only shuts down the lab machines, but it also shuts them down in the appropriate order. There are also cmdlets called Snapshot-Lab, which will do exactly that, take a snapshot or capture the machine at a specific point in time and save it so you can revert back so in case you blow things up, you can always come back to this snapshot. If you want to rebuild the lab, maybe you do goof things up after you've made that snapshot and you go in and you play around and you goof things up and you go man I want to go back to the previous snapshot or the snapshot that I just made because I goofed things up, you can just simply use a refresh lab option. Now we know about the Run-Lab PowerShell cmdlet, that's what we typed in to apply the configuration for the very first time. And then yes, we have a Wipe-Lab, yeah it wipes it. It destroys the lab and you have to build it back up again. Now each one of these PowerShell commands needs to make sure or you need to make sure that you run them inside of that POC multirole folder, otherwise, they won't work anywhere else. And after you're done, or after you launch the lab up, you're going to simply log onto the client machine and log in as company\administrator with our funky password, if you remember that was p@ssw0rd.

Post Installation Steps

So let's jump into the lab environment here so you can actually see how this is done, okay. Okay, so we're back. You'll notice that you should have a screen similar to mine where it's gone through and it shows you that you've had a lot of successes, see you're a successful type guy. Let me scroll up here a bit. But you can see that organizational units were created, groups were created, accounts were created on the DC, if I scroll down here a bit more, you can see what's happened on S1, it obviously joined the domain, it has a DNS server that's configured for the 192. 168. 310 IP. You can also see what's happened with the Nano Server, and of course, the client machine. My big thing here is making sure that hey, I've got 0 fails, which is pretty cool. Wow, the demo gods must be smiling upon me. Now you'll also notice here at the end it tells you the VM setup and configuration is complete. It's recommended that you snapshot the VMs. And so, if I come over here and let's actually open up Hyper-V, you'll notice here that the machines are currently running and that's just the default for setting up the lab. Now what they're telling you here is to, if you want to stop the labs, you just type in Shutdown-lab and it says when the configuration is finished, you can do a checkpoint by doing a Snapshot-Lab. And of course, if you want to quickly refresh from the last checkpoint, we do a Refresh-Lab. Now I could do this through the GUI environment by just coming over here and highlighting all these and right-clicking on them and selecting to do a checkpoint, but you notice that one thing that we discover in PowerShell is a lot easier to go through and do it from a command-line interface a lot faster. So I'm just going to type in here shutdown-lab and if I bring up the GUI again for my Hyper-V clients, you'll watch them shut off one at a time there. Now the other advantage here of using the shutdown and the Start-lab PowerShell commands is that it actually brings them up and shuts them down in the appropriate order. So now that we're done doing that, we're going to go ahead and do a Snapshot-Lab and it should have created the snapshots. So you notice right now there is no checkpoints. Snapshots and checkpoints are synonymous with Microsoft because we originally called them snapshots and then they were changed to checkpoints. But if you'll notice here, if I highlight or refresh basically, I can see a lab configuration checkpoint being displayed here for me. So let me come back over here and again if I want to refresh, I can just simply type into refresh the lab, whoops, I've got to type enough in so it knows what to do. Refresh the lab and it will actually go back and rebuild up those labs from that checkpoint, but I don't want to necessarily do that. I'm just going to come back in here since I've created the snapshot, I do have to do a couple of more options here. Let's go ahead and run the lab again, and just for giggles we'll come in here and you'll notice again, remember how I told you that it shuts up, shuts down, and starts up the machines in the appropriate order. You'll notice it's bringing up the domain controller and it's giving you 60 seconds for it to come up before it launches the other machines because obviously, we need a domain controller in play to authenticate the machines, as well as the user accounts or the user as you try to log in. Okay, let me move this down here a bit so we can see what's going on. Here we go. And you're going to see as soon as this is done processing, which just counting down 60 seconds, you'll see these other machines here fire up. And typically, it doesn't matter which order they actually fire up because your most important machine at this point in this particular lab is the domain controllers, and so, you can see they're all firing up right now. So we're going to give this just a few more seconds for these machines to fire up, and oh while we're doing that, let's go ahead and do a little bit more maintenance here. Let's give internet access to our VMs and we do that with a cmdlet that was installed for us that's just simply called Enable-Internet. And once we've enabled the internet for these machines, the last thing we want to do is to do a Validate-Lab, which goes through and just simply makes sure that all the configurations have actually converged and that we're ready to go. Now this brings up those four main machines, remember the domain controller, the Member Server, the Nano Server, and the client machine or the Windows 10 box. Now the other machine that we want to bring up is a Linux box called Kali Linux. So the reason why I mention this is because in other courses within the series, as well as other security courses that we have here at Pluralsight, you might want to use a Kali box. So let me show you what we're going to do with Kali and then we'll actually install that.

More OS's: Kali Linux

Okay, so this section here, we're going to talk about how to bring in a different OS, in this case here, it's going to be Kali Linux, but just know that you can use this process to bring in other machines into this lab environment. Now here's an important note, if you run any of those lab cmdlets like to wipe the lab or to shut down the lab, those cmdlets do not affect machines or VMs that were not set up through the configuration files. So if you type in a shutdown the lab, it's going to shut down the four machines and it's going to leave your Kali Linux box up and running, and if you have a Kali Linux box and oh, I don't know you fire up an XP box so you can see what'll happen in a particular situation, it doesn't shut those down, so you've got to remember to shut those down. Same thing with capturing a snapshot or a checkpoint, you'll have to do that manually. But in order for Kali to work, we're going to go through and actually first take a look at downloading Kali Linux, and we're going to download that from kali. org directly. You can use the 32-bit version or the 64-bit. I have this tendency of using the 64-bit and I also download the full version. I am then going to go through and I'm going to create a new VM and I'm going to make sure that it has at least 2 GB of RAM. I am then going to attach the ISO, and from there, you can make a choice. Once you attach the ISO and start up the virtual machine, you can either do a default installation, and if you do, just take the defaults or you can run Kali from the DVD itself. So I know it seems like a lot of steps. We're going to go through them here right now with you so that you can kind of see how easy this is to add an additional operating system. Okay, so I'm back here in my Windows 10 box. I'm going to bring up my Internet Explorer and let's go ahead and go to download Kali Linux, we can just simply type in Kali Linux, and you can select download, and we're going to go ahead and make sure you get it from the official kali. org location, not somebody else's rendition of it, and I typically grab the ISO for the 64-bit. And if you're running a torrent, I should probably slap you for running a torrent. I know it's faster, but there are other things on torrents that are not good for you security-wise. So I would typically hit this ISO. It's asking me here if I'm going to save this. I'm going to do a save as, and I'm going to go ahead and select to save it here in my Downloads directory. So we're going to pause again the video while we, oh you know what, while that's downloading, let's not pause, let's just keep going because hey we can multitask. So what we want to do is we're going to create a virtual machine. The script that we ran before created the virtual machines for us. Well, we need to do one manually now so that we can then take that ISO that we're downloading and mount it and then install. So I'm going to come over here in Hyper-V and select to do new. I'm going to select to create a new virtual machine and I'm going to get the 4-hour wizard. It's not really 4 hours. I tell everybody that anything that I do takes at least 4 hours because any time I tell somebody it'll take me just a minute, they then anticipate that everything I do will take just a minute. Plus, typically, I find that when I say it'll take me about 20 minutes, it ends up taking me 2 or 3 hours. So yeah. I'm going to go ahead and hit Next here and it says what's the name of your virtual machine. Well I'm going to call this one Kali Linux and it says where do you want to store this machine. The default for Hyper-V is to store it inside of the data or the program's data directory. I'm going to select I'd like to use a different directory and you can go through and place it in the same directory where we installed the lab. Actually, I take that back folks. It's going to create the directory for me, so I'm going to just come back here and select Autolabs and it asks what generation of a virtual machine do I want to support. You're going to select Generation 1, trust me, because Gen 2 requires you to run a specific operating system, so we're going to stay with Gen 1. It asks how much memory do we want for startup and I'm going to put in here 2048. I'm a big fan of removing dynamic memory. This allows the memory to expand and contract, but I'm going to make it reserve the 2 GB for me. I don't want to suffer any performance issues here. It then says okay, where do you want to connect. Well remember LabNet, that's when we typed into create our network, that PowerShell command, that's what I'm going to say is I'd like to join LabNet please so I can see the other virtual machines. I'm going to hit Next and it says okay we're going to create a VHD, which is going to be representing the OS of Kali Linux and we're going to put it inside of AutoLabs\Kali Linux\Virtual Disks. Now the other drives that the Autolab created are actually inside of this master virtual disks. So you can see there is the core, there is the Nano, there is the Enterprise for Windows 10, and then we also have here is the ISOs that it downloaded. Now technically, if you wanted to, you could put your Kali ISO in here as well, but the lab is not going to do anything with it at this point. And if you ever blow away the lab, you're going to lose your ISOs. Actually, a smart learner will take this directory that has the hotfixes because there are hotfixes in here for you that it downloaded on the fly, they would actually make a backup of these or copy it to a different location, so if you blew away the lab, you could just simply copy it back over in case something happened. Anyway, so that's where it's going to store. It says this at the size that we're going to create. I'm going to go ahead and leave that for right now. That's fine. I'm only going to end up using the amount of space that I actually require for the operating system, and I'm going to go ahead and hit Next and it says okay how do you want to install an operating system. Well I'm going to be installing it from an ISO, so I'm going to select the install from a CD or DVD ROM, but I'm going to make sure that I'm doing it from an image file and let me come over here and see if I'm done downloading. Nope, I've got about 18 minutes left. So we'll come back to this. Actually, what we'll do is we'll say we're going to install an operating system later. Or you know what, I don't want to mess you guys up, so we're going to just pause here the video so we can wait for my ISO to finish downloading and I'll start talking to you as soon it's done downloading so we can finish this up. Okay, so it looks like it finished downloading. Let's go ahead and open up that folder, and sure enough, there it is. I'm just going to cut that and I am going to go ahead and place it inside of my directory here for AutoLabs, I'll put it under ISOs. I'm just going to paste it right there so it's easy for me to find. Okay, so I'm going to minimize this, we'll minimize our browser, and we're going to come back here. Remember we were saying how do I want to install an OS, well I'd like to do it via the OS ISO file. And so, I'm going to come back in here, we'll come back to the C drive, AutoLabs, ISOs, and I'm just going to click on the Kali Linux and just select Open. I'm just going to go ahead and select Next and it gives me a Summary page and I'm going to hit Finish. The hamster is running around the wheel for a second and we're back to here. I think we should be able to go ahead and minimize PowerShell here and there is my Kali Linux box. I'm going to go ahead and connect to it by right-clicking on the name, Kali Linux, and selecting to connect. This is going to open up a window, but obviously, I haven't started the machine, so there is really not much to see here. So let's go ahead and start this box. It's firing up and here's where I mentioned before that you can change how you want to run. Live means that it runs from the DVD or you can do an install or you can do a graphical install. Now to make things easy, I'm just going to go ahead and select Live and it's going to fire up Kali on the DVD. So nothing actually gets installed on this box. If you want to go through the time and install everything, just take your defaults and there we go, it looks like we had a little bit of a graphics issue there, but you could just take the defaults to install and it'll actually run a little bit faster, but I'm already running on some solid-state drives. We're going to go ahead and change this view so I can go to full screen so I can see it. So there's the desktop for Kali Linux. Now the only thing here is that by default I don't their DHCP server on the network, DC1 is just that, a domain controller with DNS. So I'm going to come in here real quick and I'm going to see if we can't figure out if it's obtained an IP address or if we need to assign one, so we'll come into wired settings here and it tells me here, oh it is getting an IP address, so there is a DHCP server. Must on the, maybe it's on the Nano Server because 192. 168. 3. 200 is a valid IP address range. So we'll go ahead and close that down. I'm going to go ahead and close this window. Now closing this window in Hyper-V does not turn the machine off. In order to do that, you can either turn it off from within Kali itself. If I double-click and open up Kali again, I can come up here and select the power option over here. Come on, there we go. I can select I'd like to power it down or, I'm going to close this down, I can right-click on it and select to shut it down. There is a difference between shutting down and turning off. Turning off is the equivalent of pulling the plug on a computer and I'm sure you're not real excited about doing that, are you. But I did want to show you this. I'm going to snap this over to this over to this side here. We're going to bring up my PowerShell command. Let me clear the screen real fast. Whoop, that's kind of weird. Okay, now here's what's interesting is that if I'm done with the lab, I did tell you before that it doesn't start the virtual machines up that are created outside of the configuration file, but it will shut them down. I may have said that it won't shut them down, but it does because it's actually executing a complete shutdown of all VMs. So if I just simply type in shutdown-lab and hit Enter, you're going to see it go through. It's going to shut down all the virtual machines, it's going to leave DC1 as the last of the virtual machines in the configuration file because that's the last machine you want to have fire down, but the Kali Linux box is outside of the configuration file so it should still shut it down or well there we go. So now it's not going to do it. There are times that it's shut it down and there are times that it hasn't shut it down, so just be aware of that. That you'll need to doublecheck and if you need to, you'll need to come in and go ahead and shut down the Kali Linux box when you're done. You would want to go through if you do a full installation, make sure you right-click on it and create a checkpoint of it so you that can always revert back to it very, very quickly. Okay, so remember that if you have any questions, feel free to post them within the course form and also check your revision questions, as well because sometimes we update some of the questions that have been answered. Just make sure that you don't duplicate basically the questions that have already been asked. If you have any questions about the AutoLab, make sure you post those at the GitHub Issues tab.

Summary

Okay, so in this module, we went through and talked about the Pluralsight AutoLab. Again, remember this lab can be used in almost any of the Pluralsight courses where you need to fire up a playground area. we talked about the lab structure itself, as well, where we talked about which machines is was going to build up for us, we also talked about the requirements. Again, make sure that you have some beefy machine there for yourself. You could probably get away with maybe 8 GB of RAM total, but man just make sure you leave at least 1 GB of RAM. You're going to be so much happier if you have a system that has 16 GB, if not 32. We also talked about that you could go through and use nested virtualization where again that's the concept of firing up a virtual box within a virtual box, so you don't goof up all your other virtual machines. We then talked about how to deploy the AutoLab, we downloaded it from GitHub, we did the installation, we launched the lab so that it went out and grabbed all the ISOs for us, all the patches, and created everything for us on the fly. And then, of course, we showed you how to go through and use the AutoLab, again, making it easy for us to totally destroy an environment or bring it back up and then also how to bring in additional operating systems if we need to. So now you've got a lab. You got some place to play around with. Shall we play? In the next module, we're going to go through and take a look at and discuss some of those attack vectors that attackers use and we'll see what kind of damage we can do.

Let's Look at the Exploits

Module Overview

Okay, so let's take a look at some of the exploits that we've talked about and that can be susceptible to our IAM environments. In this particular module, we're going to go through and take a look at first of all the concept of DNS spoofing and how easy it's done and the many places that it can actually be done at. We'll then take a look at something called session hijacking, as well as a man-in-the-middle attack and how to do privilege escalation. Now one thing I want to make sure you understand before we get going here, please don't do this in your production environment. Instead, I want to make sure that you've set up your AutoLab. If you haven't, please go back to the previous module and install the AutoLab so that you can confidentially play around with these tools without actually doing any type of damage. So fire up the VMs and let's get going here.

DNS Spoofing

Okay, so the first exploit I want to show you is called DNS spoofing and in order to understand what is we're going to be showing you here in the demo, you need to understand or maybe we'll just review how DNS works and the different attack vectors that we can hit with this one. So when it comes to DNS, if I go in and type in I would like to go to www. Pluralsight. com, my computer fires up and it actually does a couple of things. First of all, it checks itself to see if it is by chance the name Pluralsight. com. If it's not, then it goes off and looks at a file that's located on the machine directly that's called the host file. The host file is kind of a shortcut file that some administrators can use to verify or to make sure that no matter what we go to a specific IP address. It's used more for internal, but we do see it happen or we do see attackers utilize it to trick your computers and go in this different location. If it's not able to resolve the name of Pluralsight. com, it checks the cache of the local machine to see if it's been there before because typically, we do have a tendency of going to the same sites, however, this cache does get flushed every now and then. So if at this point, we still can't resolve the name, what happens then is we start to involve other servers and the main server that we typically deal with is going to be a DNS server. Now this DNS server could be one that's located within your organization or it could be a DNS server that's provided by your ISP, or it could be one of the root servers, or it could be a combination of all three of those being utilized. Maybe you forward the request to your DNS server for your company, it can't resolve it, so it forwards it to the ISP DNS, and of course, if the ISP DNS server can't resolve it, it forwards it on out to the root servers of the internet, but we're going to be focused in on a local DNS server. Once the DNS server is able to resolve the address, then it just sends that request back to the client machine and the client machine uses that IP address and goes out and directly connects to that website. Now what we're trying to do here is we're trying to poison DNS somewhere, whether it's at the local machine or at the server itself, regardless of how we accomplish this, my goal is to get you to come to my special website. So let's fire up those VMs and get our demo on, okay.

Demo: DNS Spoofing

Okay, so here we are back on our host machine, again with the AutoLab. If you didn't watch the previous module, you're going to have no idea what's happening, as well as it's not going to work for you. So the first thing I want to do is to simply come in here and type in to run the lab because it's not running right now. You're going to see it's firing up, okay, so the domain controllers finishing firing up here and here come my other virtual machines, and I'm going to go also ahead and start up my Kali Linux box, and let me double-click on that, tell it to go live, we'll minimize that, so we'll get all of our machines going up here. Okay, I'm going to go ahead and minimize down my PowerShell command for my AutoLabs. Let's go ahead and come in and log into our client machine. I'm going to do that just simply by double-clicking on the client machine. It will go ahead and take that resolution, 1280 by 720, and we'll bring this over, and I'm going to go ahead and select go to full screen mode, and we're going to go ahead and log in as the company administrator. If you remember, our password was A, excuse me, P-@-s-s-w-0-r-d, and we'll log in here. Okay, so the first thing we want to do is if I open up a command line interface and you can do this in either PowerShell or the command line. I'm just going to do the old fashion command line here, and again, let me make the font here a little bigger for us. Doom, doom, doom, doom, doom. We'll go up to about 24, and of course, we want our Batman colors, there we go. So if I first come in here and do something like I want to be able to check my DNS cache, that's one of the things. Well that is simply done with an ipconfig/display dns and you'll notice this shows me all the different sites that I've been to at this point, and there is a plethora of them in here apparently and it's probably for some of the updates that we've done. I'm going to come in here just real quick and I'm going to flush it by going through and just typing in flushdns. And now if I redo the displaydns, you'll notice there's nothing in there. So let me just ping real fast yahoo. com and you notice I get a response back and look at the address here. It's a 98. 138. 253. 109. Now you may get a different IP address from Yahoo, it's because they have multiple IPs that are responding, but I just want to show you here that what I want to do instead is I'm going to go and open up a host file. Now the host file is located in a top-secret area. It's on your C drive and this is with most Windows products. Linux, it's in a different directory, but I'm just trying to show you the theory here of being able to poison the DNS. So it's located here under Windows, under System32, I just hit the S key on my keyboard to bring me down to the Ss, and then it's under a directory called Drivers and inside of a folder called etc. Yeah, I know. It doesn't make any sense, does it, but it's been there for decades. And here's the file, it's called the host file. You'll notice there is no extension to it. If you put an extension on it, it actually gets ignored. So I'm going to open up this host file by double-clicking on it. I'm going to just open it up inside of Notepad. You can open up in Word if you wanted to or WordPad, but it's a small file. And you'll see here that they give you an example of how to use this particular file or the entries or how to create your own entries. The asterisk here, whoops, the asterisk here is simply a rim statement, meaning to ignore it. So the first interim I'm going to put in here is at 127. 0. 0. 1 is going to be equal to by hitting the Tab Space or the Tab key, yahoo. com. I'm also going to say that 127. 0. 0. 1 is equal to Facebook. com. Let me have some fun here real fast. I'm going to come over here and let me type in real fast, I'm going to do an NS lookup, meaning go find me the IP address please for msn. com and I'm going out and actually asking my DNS servers. My first server is my 3. 10 box, which is our domain controller, and it went out to the internet and it found it and here it is right here. I'm going to just grab this IP address, try that again. I'm going to hit Enter, I'm going to come back to my Notepad, and I'm going to put in that IP address is actually equaled to Google. com. Now I'm going to save this file by coming up to File and Save. I did the shortcut and hit Ctrl+S. Now I'm going to go ahead and close this down. So now, I want to make sure that I clear my cache here again, so I'll just do my up arrow, well we'll go ahead and type it in again. I was going to say up arrow keys, but it's good to review the cmdlets, or excuse me, the command interface. I'm going to do an ipconfig/flushdns. Okay, so now let's open up our Internet Explorer. If you try to open up Edge because we've logged in as the administrator, by default, Windows 10 does not allow you to run that security, think about it, because remember when grandma got her computer and she fired up here box, she became the local admin and then she went to a website that installed a ran as her. This is the solution for that. It doesn't allow you to run the Edge, but it does allow me to run Internet Explorer. Ha ha, get around it. So I'm going to come down here to Windows Accessories and Internet Explorer. You could probably try typing Internet Explorer, it just depends if the system has had time to go through and actually index all the programs. And we'll go ahead and take those. And so now, I'm going to go through and type in, I'd like to go to www. google. com. Oh, you know what, I probably should have a www in front of these two because the default now is to place the www in front. We'll go ahead and save this. Again, I'm hitting a Ctrl+F and then S for Save. We'll come down here and let's flush out our cache again, just to make sure, and now if I do a ping www. facebook. com, you'll notice the IP address I get is a 127. 00. 1 address. Same thing if I type in www. yahoo. com. And because there is no website on my computer, what's going to end up happening is if I come down here and select, see if we can get this to work. I'll go down to Windows Explorer, or excuse me, Internet Explorer. Don't use Edge because Edge, by default, if you try launching it won't because we're currently logged in as the administrative account. This is a security issue. It's a good thing. So let's see what happens here. The browser may adjust for me, but we'll find out here. We'll go to Google. com. Yeah, so it just did a search for me. If I do a, because this is now a search bar for me up here. Let's try Google. com again. It keeps taking me back, but you can see that I've actually gone through and told it, oh no wait, I said Google was equal to Bing, didn't I, or to MSN, so that did work. Let's go back and look at trying to go to yahoo, though. If I go to www. yahoo, yahoo. com, I should get a page cannot be displayed. Nope. And that's because Internet Explorer is also, this is also your search box. But if I come down here, obviously, I did Yahoo and it gave me back a 127. So this is the way that we can, and again, if I come back here and look at my cache now. If I do an ipconfig/displaydns, you'll notice that Facebook is equal to 127. 0. 0. 1. So is Yahoo. So is well there's the entry for Google going to the 13 address. This one down here is an init arpa, which is for reverse lookup, and that did resolve correctly, but I didn't change that one. So again, you can go through and see how I could quickly go through and poison, as a matter of fact, I could say that oh let's see that my pirated website that I've mimicked Citi Bank with is equal to I don't know we'll say it's 127. 0. 0. 1. Pretend like that's a valid IP address out there where I'm hosting a website, and I can come in here and put www. citibank. com. And now if you try to go into Citi Bank, you would actually be redirected to my website that looks just like Citi Bank or I'm going to have you change your password because there was a security breach and we need to make sure that you change your password. So this is a fun little file to get a hold of, but it's not the only location. If I was to come in here and able to get into your system, I'm going to come in here to the start menu and type in Admin tools because I'm going to get into my DC remotely and look at its DNS. And so, from here, I can come in my primary zone file is for Company. Pri for primary, but I could come in here and select I'd like to do a new zone file. Hit Next, it's going to be a primary zone, I can store it in Active Directory if I want. It doesn't really matter. I'm going to get in here and say I'm going to do it on all of the servers, all the DNS servers on domain controllers, and the zone is going to be called Pluralsight. com because that doesn't matter for me, hit Next, and then under Pluralsight. com, I can say I've got a new host file and it's for www and it's equal to an address of 127. 0. 0. 1, again pretending like that is in my pirated site, or excuse me, my bad guys site. I could add that to the host, and now if I come down here to my command interface and I do an nslookup for Pluralsight, see if the demo gods are good to me, oh I need to go www, don't I. Yep, you'll notice that it hit my DNS server and it resolved and said yeah I know about that. It's 127. 0. 0. 1. Does it scare you at all? That's simple DNS poisoning. Next up, we're going to talk about session hijacking.

Session Hijacking

Okay, session hijacking. So session hijacking or the concept of it is being able to take over someone's session without them knowing it and we typically do this a couple of different ways. Now the most common way to do that is by simply watching for or what ends up happening is we have a client machine that makes a connection to a server. When it makes a connection, it actually creates a session id and I just made one up here, but an attacker can actually go through and start sniffing the network and look for those IDs and then just simply go back and replay or use that ID to make the connection, and so long as the original connection is in place, typically we can get right in. So I know what you're thinking. You're thinking Dale that's not only way and I agree. There is a plethora of ways that we can accomplish that. But I'm want to show you a real quick easy example. So you still got your VMs going? Let's get our demo on.

Demo: Session Hijacking

Okay, so in order to do this demo, I'm not using the lab environment that we created because I couldn't think of an easy way to kind of explain without getting too in-depth of what I was doing, so a little bit easier to see from the command-line interface or I should say from the with GUI involved. So the one thing I'm going to do here is I'm going to first come in here and show you that I'm going to open up a PowerShell command, actually you know what, let's do this instead. Let's actually open up a command prompt. We'll type in cmd and I'm going to open this up with administrative rights. This is a virtual machine that I've created for this demo so that's why I just want you to put your keyboard and mouse down so that you can pay attention to what I'm doing here. So I'm going to change up my colors and my fonts so you guys can see this clearly. Now what I'm going to do is I'm going to just simply do a whoami so you can see who I am. I'm currently logged in as dmeredith. And if I go in and specify that I'd like to know what users are on this machine or who have remoted into this machine, you'll notice that I have both dmeredith with a current rpc, or excuse me, rdp-tcp session #8 currently going and my id is actually #2. I also have Bruce Wayne who has a session and we're going to pretend like he is an employee and I'm going to hijack his session and his current session or his terminal service is on ID 3 and it's currently disconnected, so it's still running, but it's disconnected. So what I'm going to do first is I'm going to just simply come in here and do a sc for a service control, it's a power, or excuse me, it's a command-line interface that we could use to create services, so I'm going to create a service that's going to be called hijackthis and the binpath for it is going to be equal to, I'd like to start up a command line interface and I'd like it to return this /k is to return that command back to me, and then I'm going to say please use a terminal service connection and the terminal service connection I want to return back to me is Bruce Wayne's, so it's going to be 3, and then I'm going to follow that with a destination, meaning I want to return that session back to my rdp-tcp#8 and with an end quote. And you'll notice here that it created the service with no problem. So now if I come in here and do a net start hijackthis. Did you notice that. I didn't pause the video or start it back up. It actually flashed up a service or flashed up my terminal service. You'll notice if I come in here and do a whoami, I'm now logged in as Bruce Wayne, and so I'd be able to go through and whatever programs and applications he has access to, I could take advantage of. So there are different types of session hijacks. This is just a real easy one. You can do a session hijacks based off HTTP cookies, so back in the old days, we could hijack people's Facebook accounts. A lot of things that have stopped or a lot of ways that they've stopped the cookie session hijacks from taking place is the ability that everyone is doing hopefully now is running HTTPS so that kind of encrypts it, makes it a little bit harder for us, but that's just a quick example of what we could accomplish with a standard session hijack.

Man-in-the-middle

Okay, let's take a look at the Man-in-the-Middle attack. Now your man-in-the-middle attack is simply where the attacker goes through and sits in between the two communicating hosts and captures and monitors and relays communication from one host to another, but everything passes through them. Now some of the things that we could accomplish with the man-in-the-middle attack would be to use it to modify the traffic as it passes through our black box. Now one of the easiest ways that we can implement a man-in-the-middle attack is to use some type of trojan software or malware and replace genuine software on the system. Most common that we see this taking place is when people pirate software or video courses, or even a web browser can be targeted by installing malicious plugins and scripts. Technically, that's considered a man in a browser. Now most of the time, man-in-the-middle attacks are accomplished using either the art protocol or DNS. They can be very easily defeated though if you use mutual authentication where both the server and the client exchange credentials. So you still got those VMs still running? Well let's continue on and let me show you how to do a quick man-in-the-middle attack. Okay, so here I am back on my host machine.

Demo: Man-in-the-middle Attack

I've still got my machines up and running. I'm going to go back into my client machine again, let's make this biggie size here, there we go, take up the full screen. We're going to log in as administrator, which is the default P-@-s-s-w-0-r-d, and I'm sorry if I just guessed your password in your production environment. No, okay good. So the first thing we're going to need to do is we're going to actually need to download a nifty little program that I really enjoy using. It's called Cain and Abel and it's because there is a good side to it and a bad side to it. Now when you go to download this particular product, you might get some warnings that it, if you have some antivirus installed, that it's a problem, but it really isn't so long as you get it from the right location. So I'm going to come in here and just do a quick search for cain abel and we'll do the name of the company that creates it or the site that it's at is oxid. it and there we go. You could actually just go to oxid. it. Okay, so this is the Download page for it. I'm going to come in here and go ahead and grab the one for version 2000/XP. It'll actually run on later OSs. We'll go ahead and save this to my disk. We'll save it in my Downloads directory. It's just called ca\_setup. Wait for that to download really fast. I guess fast is a relative term, it depends on your internet connection, right. We'll go ahead and get ahead of ourselves here and jump over to our Download directory, whoops. Now the one thing you'll need to note is that Cain and Abel does require you to install some other software programs. Well it's not really software program, they're drivers so that we can put the network interface card into what they refer to as promiscuous mode. Yeah, see how I got that error popping up, that's okay. Yeah, yeah, yeah, yeah and we're going to take some action. Okay, here we go. We are going to go through and say that Cain and Abel is allowed. Yeah, so I guess it's a good thing that it's picking this up. You know one of the things too that I should mention here is that I always tell IT folks is that one of your benef, or excuse, one of your first line of defenses besides training is also knowing what is on your machines. And so being able to do a software inventory and seeing if somebody has got this particular product installed, I want to know why they have it installed because it gets installed as a full suite. So I'm going to go ahead and hit Run here and I'm going to go ahead and hit Next, Next, it's going to install. And I'm going to hit Next here, we'll just let it go into the Cain folder, and hit Next again, and we're going to hit Finish. It says, yep, we need to install WinPcap. This again allows the network card to run in promiscuous mode. So we're going to go in and install that and it looks like the particular version that it's trying to install from within Cain and Abel doesn't actually support Windows 10, so we'll go ahead and hit OK here and we'll close that down. Okay, so we're just going to open up another tab here real fast and we're going to come in here and search for winpcap Windows 10 and the link we want is actually, don't take that top one because winpcap. org is the actually location. I think it is just simply install, yep, there we go. So make sure you get 4. 1. 3 and make sure you get from winpcap. org. It's very important that you get it from the right location. Go ahead and install this. Okay, I'm going to go ahead and hit Next. We're going to hit I agree, you can start it automatically at boot time, hit Finished. Okey doke. So now let's minimize everything. You notice there is a shortcut up here for Cain and Abel. I'm going to go ahead and launch it. No, it says that the firewall is enabled, so I want to go through real fast. I can hit OK, but I want to go ahead and disable the firewall on this particular box, so I'm going to come down here to Windows firewall and we're going to turn it off just because we're testing and playing. Yeah, yeah, yeah. Okay, so this is the interface here for Cain and Abel. Okay, and that reminds me. There may be something that pops up, let's see. So I'm going to go through real fast and I'm going to first tell Cain and Abel which interface to use. Now if by chance your interface comes up with showing 0s here, one of the tricks that you can do is you can come into, you'd want to shut down Cain and Abel, so I'd cancel this, exit, you're going to go into your network and sharing center. If you showed an IP address, then you don't need to do this. It's only if it comes up as 0000. I'm going to come into the properties for this network interface, I'm going to come to Internet Protocol Version 4 TCP/IPv4, and I'm going to select the Advanced button, and then I'm going to switch over to the DNS tab and whatever this is selected as, if it's already checked, then uncheck it. If it's unchecked, check it. And then we're going to come in here and when you run Cain and Abel, it should fix it. I think it's just a quirk and it should show that it's the IP address. So what I'm telling Cain and Abel here is I want to sniff off of this network interface. I'm going to go ahead and hit okey doke. If I come over here to my Arp Poisoning, I have the option of using real IP addresses and MAC addresses or I can change up the MAC address, but I'm not going to deal with any of that. I'm just going to hit OK here. I'm then going to come over here to the Sniffer's tab and I'm going to select the + button. Oh, it tells me I have to activate my sniffer. It's right, I have to activate the sniffing. So right here, this icon that looks like a network interface card, tells me that, or is my sniffing, ability to turn on my sniffing. Now they give you a warning here that we don't need to worry about for what we're doing. I'm just going to go ahead and hit OK. So it's currently sniffing. I should be able to come in here and select to add to my list. I'm going to look at all my hosts on my subnet and I'm going to select that I'm using different ARP tests to find them. Actually, we'll just select all the tests and hit OK and you'll notice that it's finding my IP addresses of my systems that are currently up and running. I should also get the 100 IP. Hmm. I did not find my 100. I thought it was 100, isn't it? DC1. Let me log in here. Oh no, I'm 10. I'm like on drugs today, folks. Sorry about that. So 10 is my server. It's also my way out of here. So let's go back here to my Cain and Abel. So yeah, so 10 is the server, 1 is, I apologize, 1 is my way out of here, it's my default-gateway. And this particular box right here happens to be, let's go full screen so you can see this again, there we go. Sorry about that, I should have done that earlier. Let me open up a command prompt real fast, do an ipconfig, and this 1 is 100. Yeah, so it's not seeing itself, which makes sense. But what I'm going to do is I'm going to go through and set up my man-in-the-middle attack. Okay, so then if I come to my ARP tab, this is where I can implement a man-in-the-middle via ARP, and how I'm going to do that is I'm going to click in this, you'll notice my + is not accessible to me, so I have to click down here in the status area and I'm going to hit add, and it's asking me okay, it tells me what ARP poisoning is going to do and what they're telling me to do here is I need to first go through and select the machines that I want to interfere with, and since I select the default-gateway, it's telling me here that I'll be interfacing with all of these machines here. I'm going to ARP poison each one of these. Now just to kind of prove the point, let me go back out. We're going to go the server box so you can see this. Let me minimize this and we'll go back to the server box, and previously, I went through and did an ipconfig. I'm going to do an ipconfig/all. I'm going to do a ping to 192. 168. 3. let's go to 60. That was one of the other addresses that were out there, right. So I get a reply back. Now I'm going to break out of that and I'm going to do an arp -a, which says, oh wow sorry, I'm giving you guys small text. Let's fix that up for you. There we go. So you notice I got the reply back, I'm going to do an arp -a, which shows me all the ARP information. Now ARP is in charge of resolving an IP address to a MAC address, and so, if I hit Enter here, it shows me that the current MAC address for 192. 168. 3. 1 is note the address here, as well as you can see, I've picked up 60 right here. So just to make a mental note of those, we'll actually scroll up and down to them. I'm going to go ahead and go back to my client machine, I'm going to go ahead and hit OK. Now before I start to poison, I want to actually come over here so that you can see this IP address or this ARP address, let's see if I do an ipconfig /all, you'll notice that my IP or my physical address is 04. So again, go back and show you here my server currently thinks that yes, 04 belongs to 100. Everybody follow me so far? Okay, so let me go back, go back to the client machine, we'll minimize that, again we'll go full screen. Oh you know, I kind of goofed up here, hang on. I should have not just hit 60. I should have hit all of these. There we go. So now we're going to be looking at 10, 50, and 60. Let's go up here and poison this thing, it tells me it's poisoning. We'll wait just a second here. Not a whole lot is going on, so let's see if we can create some, there we go. So let's see 60 is fully routed through this box now. Okay, there goes #10. It's fully routed. So since it's fully routed, let's go back and look at its interface. So again, I want you to remember that it thought that the default-gateway, it's ARP address ended in 00, right. Let's look at the ARP cache now. You'll now notice that it is 04 and 04 is yes, guess what 100, the IP address of 100. So right now, everything this machine is doing, if it tries to ping Yahoo, the information is actually passing through my Cain and Abel box, and if this machine had a GUI interface, I could try to log into Internet Explorer, I don't think I'd get so lucky as having yeah. It's a core box. You get nothing down there. Anyway, let me clear that out. Everything it's, if I had a browser on here, I would actually go through and if I started typing in passwords and username accounts and this would actually start to show me the different requests for ARP, and then on the Passwords tab, we'd actually see the passwords popping up here for LDAP passwords, HTTP passwords, FTP passwords, there is a plethora again here of passwords that it would start to track. Again, without some type of protection like appliances or software programs that are looking for network cards that are in promiscuous mode, it would be very hard for you to capture this or catch this. Let me put it to you guys another way. If I was in a public Wi-Fi and I did this, I could capture a lot of interesting traffic. I'm not saying that you should, I'm just saying what's my favorite phrase, just because you can, doesn't mean you can.

Privilege Escalation

Okay, so finally, let's talk about privilege escalation, and this is really kind of an easy subject because all we're talking about here is the concept that there are applications, as well as processes that when they execute or run, they have to do so using some type of an account or they have to have privileges to read and write data and to do what they need to do. Now depending on how the software is written, a process could actually be ran as a system account or even an account of a user that's currently logged in. Now how privilege escalation works is that an attacker can exploit the software program to be able to execute their own process and that could be anything from a trojan, a worm, a piece of malware, or even just executing some code with the same privilege as the process that is currently running. So I tell you what, let's break out our virtual machines here again and go back and do some more command line stuff, see what we can accomplish here. Actually, this is going to be another one of those where you're going want to take your hands off your keyboard and mouse again because I'm going to be doing this from that virtual machine that I created personally. Don't worry, the other virtual machines we'll be using in other courses within this series. So let's jump out and see what kind of damage or what kind of trouble I can get into.

Demo: Privilege Escalation

Okay, so here I am back in my machine that I was using, this is my other virtual machine, and I'm going to go ahead and you'll notice that I still have my dmeredith RDP session running, as well as the Bruce Wayne that's still disconnected. In fact, you can see it's been idle for almost 24 hours, so Bruce Wayne is obviously out fighting crime. So again, I'm going to go through and just as a reminder, we'll do a create. I could join the other session, but I just kind of want to make sure that we have a smooth transition all the way here. We'll do another hijackthis, oh no, let's do this. Let's go with a create a service called pwnme and the binpath, again, will be equal to "cmd. exe /k /tscon and it's going to be session number 3 with an end quote. Whoa. I didn't finish that out. I specified the connection, but I also didn't have to specify the destination, and in this case here, you'll notice that this is a different RDP session that I've come in on, so before, it was on 8, this time it's rdp-tcp#11 and we do an end quote, and then we should just simply do a net start pwnme and there is my session. So we've done this before, we've been here before, right. Okay, so I'm obviously hooked in as somebody else and let's just open up a command prompt real fast and we'll do this as an administrator, which is bwayne and our password, okay. And let's see if we can't accomplish something here. Let's do a net user Joker because again, remember I'm using Bruce Wayne's account. I'm coming in as dmeredith and we'll give a password of IHateBatman and then we'll do a /ADD. Oh, it doesn't meet the complexity requirements, IHateBatman2. There we go. So the account just got created. I can then go through and say net localgroup administrators and I'm going to add Joker with a /ADD. So now, if I come in and look at my local accounts, you'll notice here under my users, there is the joker account that was created and you'll also notice that he is a member of the administrators group as well. So again, I'm doing all this with privilege escalation where I've created, I used one account or service to create another account that actually has higher access or higher rights or higher privileges.

Module & Course Summary

Okay, so in this module, we went through and did several things. We looked at several exploits that are associated with IAM. We took a look at DNS spoofing. Again, this is where we can infect DNS either at the local level on the local machine with files or its cache or even at the DNS server itself. And by the way, this DNS spoofing also works on your mobile devices. So when you're loading apps on your phones and tablets, make sure the apps are legit because I could very easily put in there that you go to my DNS server and trick you into coming to my site. We also talked about session hijacking. Again, that's where we look for a session that's currently established to a resource and then just simply use a session ID or a cookie to overtake and pretend like that's actually the same user, but it's actually us, the attacker. We also looked at man-in-the-middle attacks. Again, this is where we place ourselves between the target or between two different targets or I should say target and a resource and we just start monitoring and picking up everything that's going across, and we can also implement session hijacks from there, as well as start cracking passwords. And then finally we went in and take a look at privilege escalation. Again, remember this is where we go through or where attackers go through and they can take an account, and by utilizing that account or service, escalate the privilege up or even create a back-door account kind of like I did with the joker account. So now that we've learned about the exploits, hopefully, you've learned a lot about Identity Access Management, or IAM. Remember that IAM is simply a technology that we use to initiate, capture, record, and manage our user's identities and the access that they're allowed to our different resources. The thing you need to consider is this is that if you have an IAM process in place that nobody is really watching, it could actually create some real issues for you, again, either from a regulatory perspective or just the perspective of you're now the latest news article on the internet about being hacked. Well I hope you enjoyed the course. Again, remember this is one in a series of courses that will help you if you're looking for, well first of all, just trying to protect yourself and your company, but also, if you're going for your GCIH, your ECIH, or even your CSA+. Again, my name is Dale Meredith, and thanks for joining me.