Windows Active Directory Certificate Services

Table of Contents

Windows Active Directory Certificate Services (AD CS)

As of Server 2008, Certificate Service are known as Active Directory Certificate Services.

AD CS is the server functionality that allows a Public Key Infrastructure (PKI) to be built within an organization.

AD CS allows the creation and management of public key certificates.

**042 So, active directory certificate services, ADCS, runs on a server. We're going to talk about running it on the Server 2012 platform.

Windows AD CS Advantages

Can be deployed without an AD forest

Can establish Certificate Policy from the AD server and then followed as users request new certificates

Can be deployed and managed using PowerShell in Server 2012

**043 Typically, we deploy it within our domain, within an active directory forest. But I don't have to deploy it within a forest. So, the reason that I bring that up is because of small businesses. Not all organizations are going to have an entire forest. So, I can deploy it even in a smaller infrastructure if I like.

One of the things to note about PKI, I said this, it is ninety-five percent process. And so, before we ever sit down at a machine and we start actually doing this work, we ought to plan out what we're trying to accomplish with our public key infrastructure and with our certificate

services because once we can plan it out, then we can go ahead and implement those policies in that particular service. Just like everything else--

AD CS Server Roles -1

AD CS Server Roles -1

Certificate Authority

- Issues digital certificates

Web enrollment

- Use a web browser to request certificates and retrieve CRL

Online responder

- Evaluates certificate status and responds to revocation status requests

**044 I can configure this with PowerShell, as well. So, what are the components, what are the roles that we're going to find in our certificate services? We have a certificate authority. The certificate authority is responsible for the publishing of the certificates. So, it provides a server where, once a certificate is created, we publish it there. And then a third

party-- when I want to verify your public key, I can go get that public key from the server.

There's also a registration authority. We'll talk about that as kind of a subset of the certificate authority.

There is a web enrollment service. That's how you and I as individuals will request a certificate. So, I want to have a certificate so I can sign my emails. I want it signed off by a trusted third party, let's say my business. So, I can use a web interface to say my name is Mark. Please verify my identity, and then publish his certificate on my behalf.

The online responder is dealing with what is known as OCSP, online certificate statuses protocol. It's dealing with certificate revocation. We'll talk a little bit about certificate revocation in just a moment.

AD CS Server Roles -2

Network Device Enrollment Service

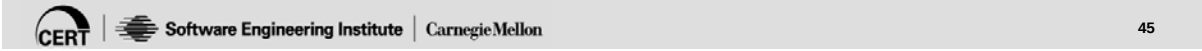
- Allows routers and other network devices that do not have a domain account to obtain certificates

Certificate Enrollment Policy Web Service

- Provides users and computers with certificate policy information

Certificate Enrollment Web Service

- Allows users and computers to enroll certificates via HTTPS



**045 Network device enrollment, if I have routers and switches and devices that don't actually participate in the domain, they're not domain-- they don't have domain accounts, well I can still have those enrolled within my certificate services through the network device enrollment service.

Where can users find the policy? A certificate authority publishes a document known as a certificate practices statement. The certificate practices statement is basically their policies about how they're maintaining their CA, how they go about verifying your identity and

those types of things. So, certificate enrollment policy web service provides users with that type of information. And then we already mentioned the web service enrollment, allows users to enroll, in this case, via HTTPS.

Windows AD CS Certificate Authority

Windows AD CS Certificate Authority

Cost effective, efficient, and secure method for managing public key certificates

Allows for the establishment of a Certificate Authority (CA)

- The CA is used to create and manage Public Key Certificates.
  - Issuing certificates
  - Revoking certificates

CERT | Software Engineering Institute | Carnegie Mellon 46

**046 They say it's a cost effective, efficient. It's-- doesn't cost you any extra. If you have Server 2012, you have the ability to run your own PKI, public key infrastructure, in your organization. So, I can create my own certificate authority. I don't have to pay third parties for a certificate.

The only issue with that is, does anybody trust me? Verisign, Entrust, Thawte, Baltimore, those are all reputable-- Microsoft fits into that category, all well known, reputable organizations that run certificate authorities.

So, if I were to download a certificate signed by Verisign, I'd feel safe that that certificate does belong to who I believe it to have belong to, Joe or Bob. But if I download a certificate, and it was signed by Billy Bob's auto parts certificate authority, well maybe I don't know who they are. I don't know what kind of reputable business they are. Maybe I don't trust that.

You can run your own CA, but will it be trusted outside of your organization? Chances are it will be trusted within. But will it be trusted widely outside of your organization. Maybe, maybe not. Sir.

Student: So, is there a service that's out there to allow your CA to be an intermediate CA of their approved CA, so that they can pass on the authenticity?

Mark Williams: So, I think you're asking could I possibly contract Verisign to--

Student: I know for sure Verisign won't do it.

Mark Williams: Huh?

Student: I know for sure Verisign doesn't run that business model. But

I'm asking if you're aware if anyone will do it?

Mark Williams: I don't know of any specifically that do it, but I know that there are some that will. You can pay them, and they will make you basically an intermediate CA beneath them. I do not believe Baltimore will. I do not believe Thawte will. You're probably going to be looking at some of the-- certainly won't be any of the big five. It will certainly be some of the others. Let's just put it that way. Yes?

Student: Couldn't you-- you could probably leverage off of Verisign to say for a certificate authority, have that be Verisign, and then everything else after that will be intermediate.

Mark Williams: That's what--

Student: You wouldn't have to ask anybody. You just sign it with their-- I mean you pay them for the SSL certificate and then reference it.

Mark Williams: Maybe. Maybe you can do it that way. I'm not sure.

Student: They don't know what you do with it after they give it to you.

Mark Williams: Yeah. That's true. All they're verifying is that you are that particular organization, and they verified that. What you do beyond that is-- well, keep in mind, certificates have-- when you get a certificate, there are going to be certain uses, depending on the level

of verification that is done. A low level certificate is only probably going to be valid for email verification.

If you want to have a certificate that you can use for financial transactions or software signing, for example, then that's a higher-level certificate. Much more verification is going to have to happen. So, I could not go to Verisign, for example, and get a low-level certificate, and then assume that I'm going to use that within my corporation to sign every other certificate within the company. It probably would not be as trusted in that fashion. Do you have a thought, Terry?

Student: I was-- are intermediate certificate authority certificates in a specific format or?

Mark Williams: The certificates are formatted the same way. Whether it's a root certificate authority certificate or an intermediate. But it will say somewhere on there it's an intermediate. It will say what its uses are for. And it will be signed by a root. Whereas a root, it will say it's a root and it's self-signed. That's one of the big differences there. Yes.

Student: Yeah I want to add on yes, the certificates you receive from the public CA's, they do contain a bit actually dictate what use it can be used for. And that's part of the-- you cannot misuse that. They made it pretty secure.

Mark Williams: You can attempt to misuse it, but will anyone trust it? Probably not.

Student: You cannot even make it work. You can-- basically, you have your own private key-- have your public key, but you will probably fail in the first step if you want to import that key and certificate pair into your own CA. It just won't complete-- fail in the first place.

Mark Williams: Right. Yeah because of the formatting says it's not supposed to be used for that functionality.

Student: Exactly.

Mark Williams: Right.

Windows AD CS CA Types

There are two core types of Certificate Authorities

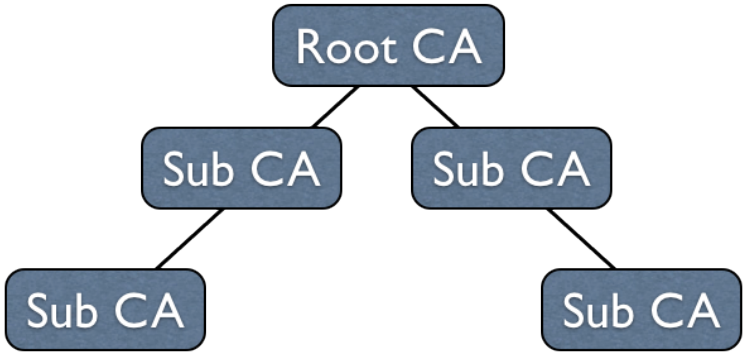
- Root Certificate Authorities
- Subordinate Certificate Authorities
  - A subcategory here is Intermediate Certificate Authorities
  - Also known as Policy or Issuing Certificate Authorities

**047 All right. So, we talked a little bit about this already. There are intermediate CAs, and we identify those as subordinate CAs. And then there are root certificate authorities.

Windows AD CS Root CA

A root CA must always be designated, whether creating a hierarchical enterprise CA structure or a stand alone CA.

The root CA is the top CA in a hierarchical structure.



**048 So, this is showing you the hierarchy. The root certification authority is going to create its own self-signed certificate. And then the root will sign for various subordinates. The subordinates will sign for other subordinates.

Eventually, what we have down here at the bottom is not a CA. So, if I were to add another little tree off of this. It would be Mark. And it would be Mike. And it would be Bob. And whoever would be the people that actually have these certificates at the lower levels. That's the whole reason we want to do this. We want to do it so that people will trust Mark, or people will trust Bob.

The alternative to this hierarchical model is what is referred to as a web of trust. And there are some organizations out there that offer certificates that follow a web of trust model.

For example, for email, Thawte-- I think it's Thawte. Thawte does a web of trust. And what that is is I can get a Thawte certificate. And Thawte really has not done anything to verify my identity.

So, here's my certificate from Mark. And then I go to a key signing party. I don't know what kind of fun you have at a key signing party. But people go to key signing parties. Or we could just do it in this environment. I could say if you have a Thought certificate, I could ask you, Bob, will you sign my certificate. So, Bob could verify that I am Mark, and Mark could verify that's Bob.

And then we could have Joe. Mark could verify that this is Joe. And Joe could verify that this is Mark. And we could have Sally. And we verify each other. And then we could have-- well, Joe and Sally, they can sign each other's certificates, right? And the idea is the more people that sign my certificate, the more people that believe that I am Mark, the more likely you are, when you see my certificate, the more likely you are to believe that I am Mark. That's the web of trust.

This is not considered to be nearly as secure as the hierarchical model. And

the reason-- I guess the big reasons it's not considered as secure is what did I do to convince Sally and Joe and Bob that I am Mark? Could have been in a bar and bought them a couple of beers. Told them my name is Fred Flintstone and please sign off on my certificate. Okay, here you go. Now, they should do certain things to verify my identity, but there's no guarantee of that because they're individuals. At least the corporations that run CAs will have process in place. I hope they have process in place to verify identity.

What might you do to verify identity before you issue or publish a certificate? For a low-level certificate, you might decide that all you're going to do is make sure they have a valid email address. So, respond to this email. And if you respond to this email appropriately, I'll believe you are who you claim to be and issue the certificate. Usually, that low-level certificate is only for email sending.

If you want to get an intermediate level certificate, we might require you give us enough information so that we can track you in publicly available databases. So, give me your name. Tell me what address you live at. Tell me who your employer is, some of those types of things, maybe even enough information so I can look you up in the credit bureau records.

If I wanted to get a high level certificate, say for a financial transactions, well now maybe we have our CA require you come in in

person and show government issued photo ID. And then once you give me the government issued photo ID, then I'll give you that high level certificate for transactions.

What if you want to get a certificate on behalf of your company? Now, we might say we'll give you a certificate on behalf of your company, if you can prove that you're authorized to do that. So, give me proof on company letterhead, signed by a board of directors, that you are authorized to act on their behalf to get a certificate.

That's the public key infrastructure. That's the policy. And, again, if I'm running my own CA, I can decide what actions are appropriate for my organization. You might decide that two person integrity is required to do verification. So, not just one person in my organization's going to verify you identity, somebody else in the organization might also verify your identity. And if they both come to the same conclusion, then we'll believe you are Fred Flintstone. And we will issue the certificate. All right?

AD CS CA Private Keys

Certificate Authority uses a Private key to digitally sign certificates.

The private key must be secured so that others NEVER have a copy!

A best practice is to store the private key on an HSM (Hardware Security Module).

- 3rd Party Root CA's may require you to have these.

If the key is stored on the computer, it is very critical to ensure the computer is secure through hardening.

**050 Private keys, the whole reason we're doing all of this is those private keys. One thing to keep in mind about private keys, we use those keys to create the digital signatures. The certificates themselves do not contain the private key. The certificates contain the public key associated with the private key.

I need to make sure that I properly protect and secure my private key. Trusted hardware's a good idea.

There's a common misconception that the certificate authority generates my private key and my

pubic key pair for me. That's not true. If you generate my private key and my public key, then it's not a private key, is it because you will have a copy of it, and I will have a copy of it?

So, I generate my own. And then I take the public key and I say here is a public key. It belongs to me. And you verify my identity. And then you have my public key. And you publish my public key. But my private key has to be exactly that, private.

Now, could I ask you, as a third party, to hold on to a copy of my private key? What would be the benefit of that? In case I lose it, it's not gone forever. It's in escrow. And there are companies that do that, private key escrow companies. Most of the certification authorities will run some escrow capability. So, if I lose my private key, I can still get it back and continue to sign emails and such. But the CA does not generate the private keys.

So, if I have my private key, where am I going to store it? I certainly don't want to store it on a thumb drive that I'm likely to lose. I don't want to store it on a laptop that's likely to get stolen.

I went to do some work at a company. They had all of their private keys stored-- well, it was the fifth floor of the building. I remember that. I remember that because the elevator, you had to have a special key just to stop on the fifth floor. I

did not have that special key. So, I could not get onto the fifth floor. They did take me there on a little bit of a tour.

And they had this big-- on the floor-- it's an open floor space. But on that floor, they had a fenced in area. It's kind of a chain link like fence. So, they had a fenced in area. And then all of the equipment that had to do with their certificate authority, their private key storage and such, all that was inside of that locked in cage.

So, they took special care to physically, and I'm sure logically, secure their certificate authority, their private key-- I lost the term, escrow server, and so forth.

They mention about an HSM, hardware security module. I might have special hardware, a special chip on my machine, that I can store-- it's not going to be on the hard drive. But it will be on a chip. It's harder to get the information, the key, off of that chip than it is to get it off of a hard drive. So, make sure your private keys are properly secured.

AD CS CA Public Keys

Used to validate the signature of a digital certificate

Available to everyone

Made available on a certificate signed by an authority higher up in the hierarchy

**051 We know that we use the public keys to decrypt the digital signature. And we use the certificates to verify the public key belongs to who it belonged to.

Root CA Self-Signed Certificate

The root CA self signs their own public key certificate.

- The structure must begin somewhere.

The root CA MUST be explicitly trusted.

- If compromised the hierarchy collapses.

The “root of trust” is critical.

- There are several “roots” like Entrust, DoD, GlobalSign, QuoVadis, Thawte, and VeriSign.
- If no one trusts you, your PKI is useless!

**052 We've already mentioned the root CA signs their own certificates, so self-signed certificates. We have to trust somewhere. And that's where the trust model begins.

Now, do the certificate authorities get it right one hundred percent of the time? No. As a matter of fact, one company, years ago, had a problem.

See, somebody went to this certificate authority and said I work for Microsoft. Please issue me a certificate for this public key that belongs to Microsoft.com. And that particular CA said okay, here you go. Here's a certificate. We're signing off

on it. You work for Microsoft. We believe you, blah, blah, blah. And it turns out that guy had nothing to do with Microsoft. And he was using that certificate to sign malware-- the private key to sign malware.

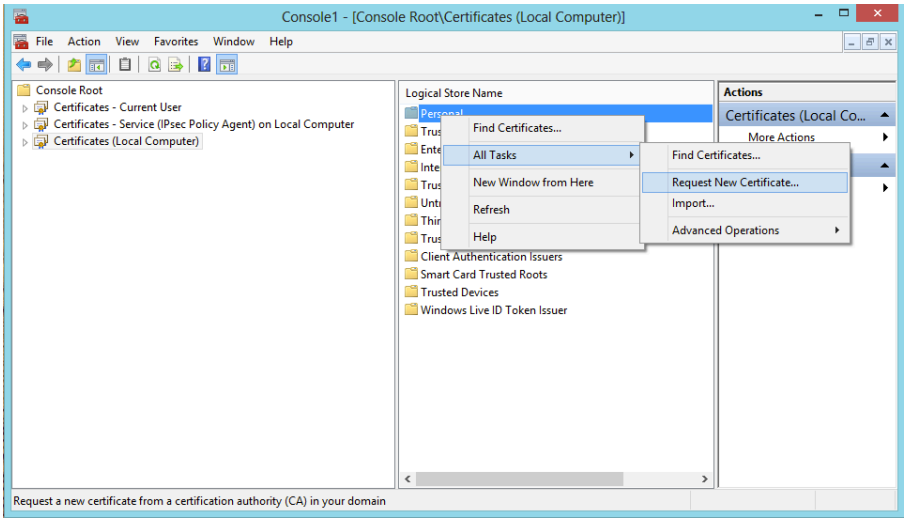
So, they didn't get it right. That particular CA and other CAs realize that that is a flaw in their process. And that's why I just told you, if you want to get a certificate on behalf of your company, you have to prove that you're authorized to act on behalf of your company. And that was one of the changes that was made to certificate practices statements.

So, the trust has to start somewhere. They don't get it right a hundred percent of the time. But they do publish their practices, so that you and I can review those practices and decide are they acceptable, are they up to par, or are they substandard practices.

Windows AD CS User Certificates

A users certificate is issued by the subordinate CAs.

A request is issued through Microsoft Management Console.



**053 All right. So, in Server 2012, we use a Microsoft management console. We have to add a snap in in order to control our certificate services. So, that's what we're showing you here.

We have the Microsoft management console up. We have the snap in already has been added. In this case, we're looking at certificates for the local computer. We have personal certificates highlighted. And the task that we're going to do in this case is go ahead and request a new certificate. So, the certificate service is already running. We're going to request a new certificate.

Installing AD CS

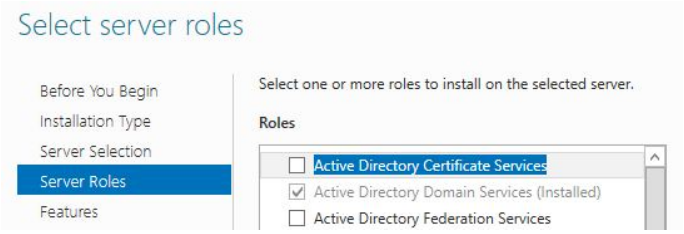
From the server dashboard

Select **Add roles and features → Role-based or feature-based installation.**

Choose the server to install CS on.

Select **Active Directory Certificate Services** check box.

Follow the wizard and choose the appropriate features.

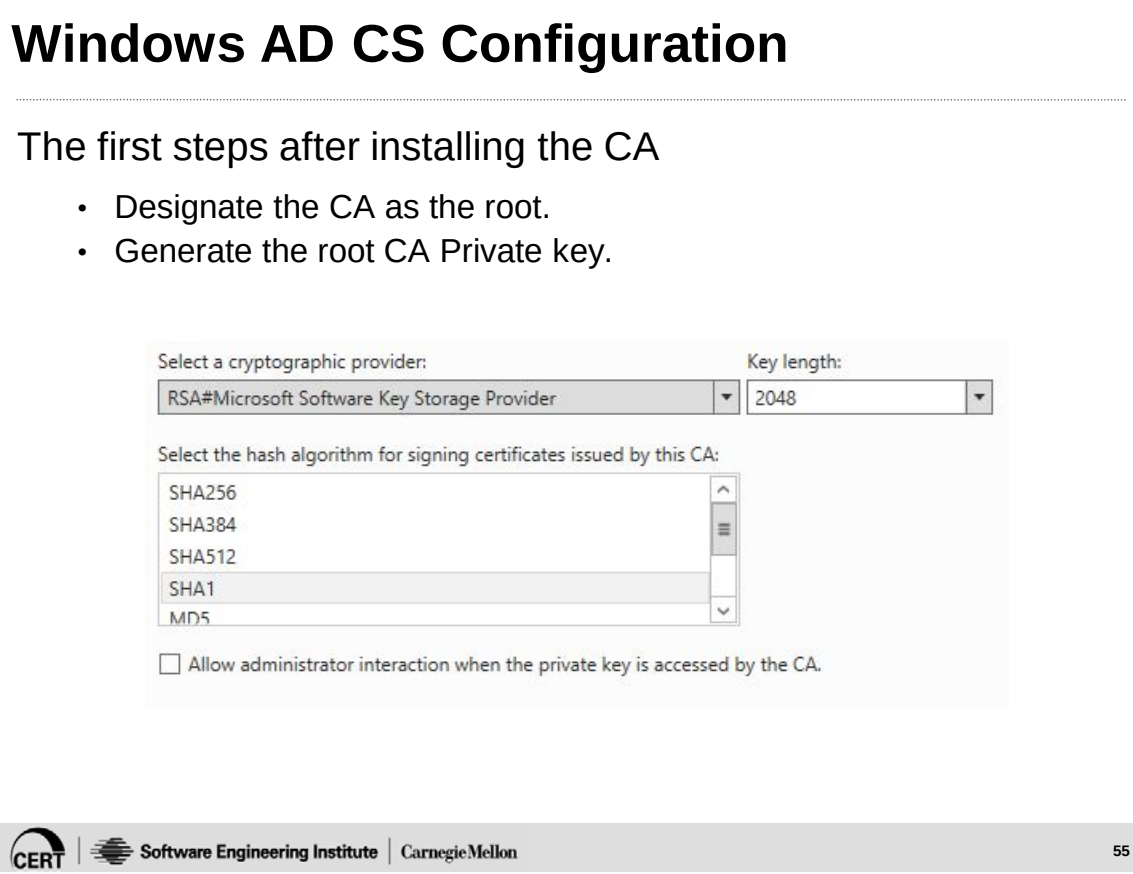


**054 Now, we have to somehow get that certificate service running. In Server 2012, there is, what is referred to as, the dashboard. The dashboard is the starting point, if you will. It's kind of like the start menu. It's the starting point for doing all things-- making that server perform many different roles.

So, in this case, we have to turn on the role of doing certificate services. So, on the dashboard, we select the server role. And we're going to check here active directory certificate services.

Now, just as a side note, we're stepping you through probably what effectively ends up being about twenty different windows and checks and boxes and such like that, just to get the certificate services running on Server 2012.

Windows AD CS Configuration

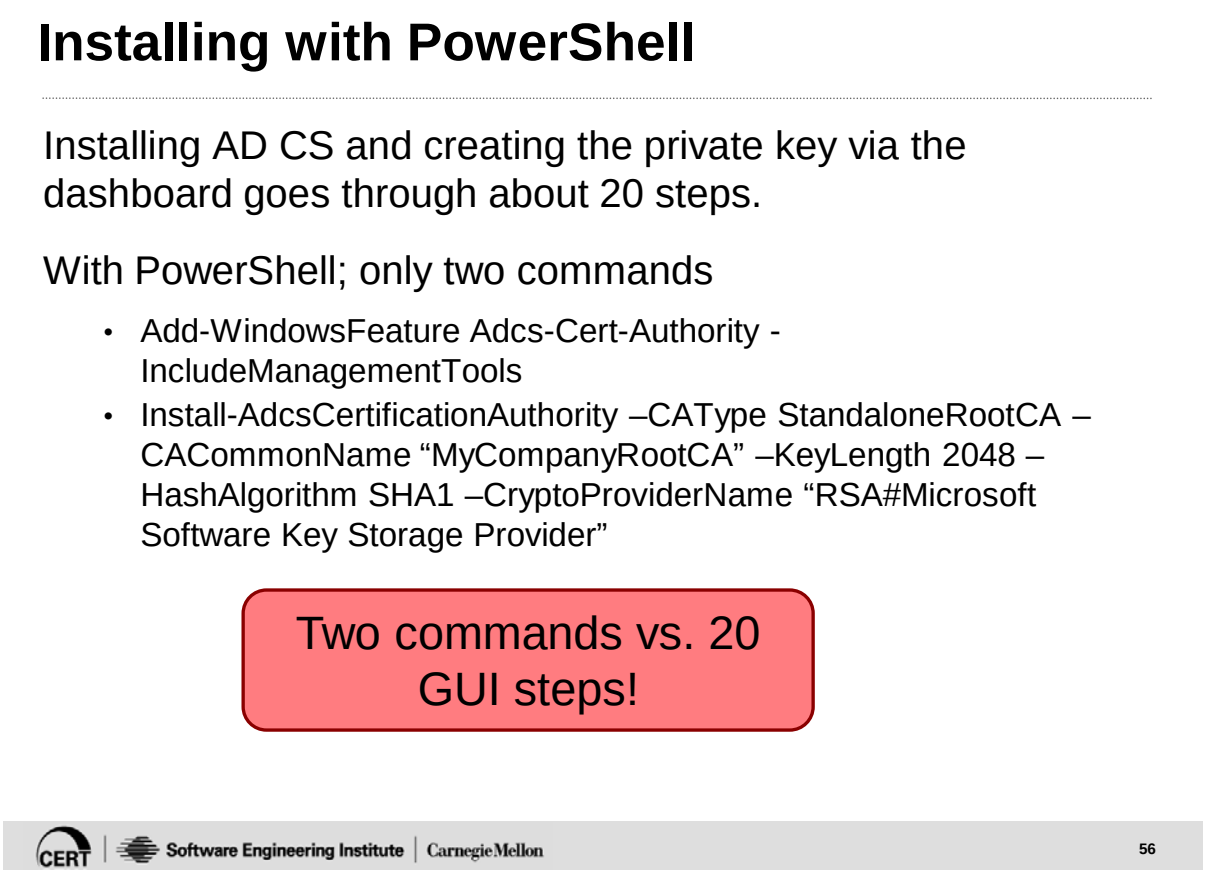


**055 After we install the certificate services, after we install the CA, now we have to go in and say who's going to be our root CA. and we have to generate the private key for that root CA. So, that's what this is selecting.

The private key, in this case, is going to be a 2048 bit key. And we're going

to use SHA1 secure hashing algorithm version one to create the message digest for that private key. So, we generate the private key.

Installing with PowerShell



**056 All right. Ultimately, after we install the certificate services, we add the-- we create the root. We create the private key for that root. And we get all of our certificate services set up and running. As I mentioned, we've gone through about twenty different windows and about twenty different checkboxes or selections.

You know what? I can do the exact same thing with PowerShell in

basically two commands. The first command you see at the top is add the Windows security feature. So, give it the role of I want to be a certificate-- do certificate services. And then this second command here, while it's a long lengthy command, it effectively creates a 2048 bit key. The name of the company is my company root CA. Here is the key. The hashing algorithm we're using is SHA1. And the crypto provider name, we give ourselves a name, RSA Microsoft Software Key Storage Provider. So, different ways of skinning the cat.

Notices

Notices