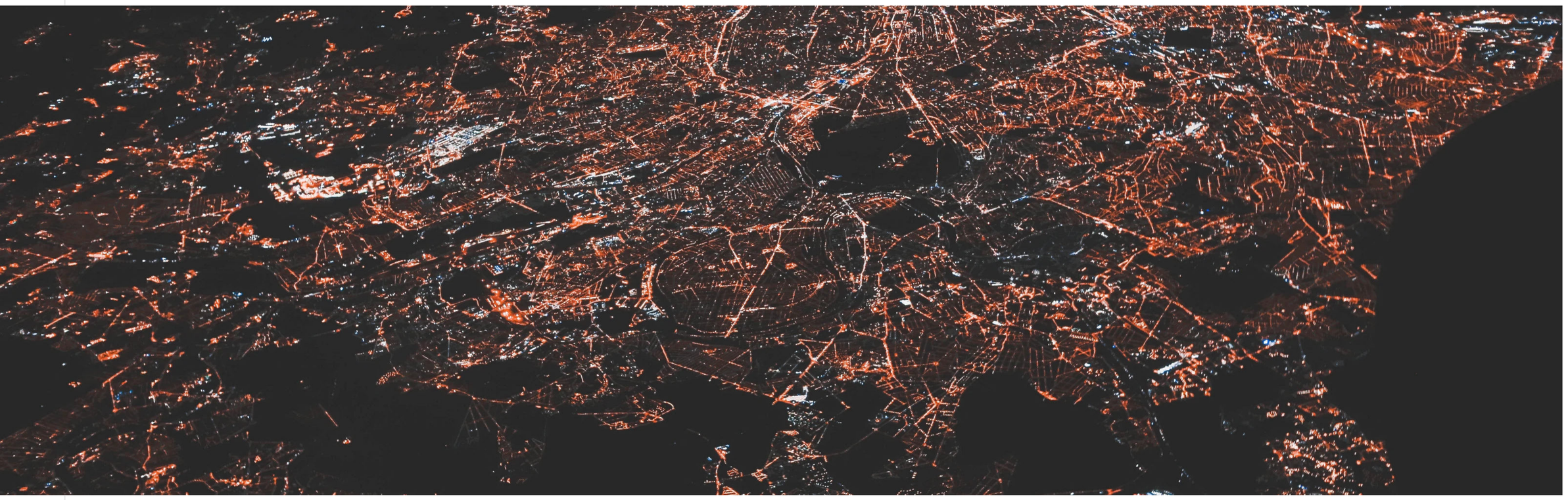


Certificate-Based Authentication Quickstart Guide

While EAP-TLS is widely known as the most secure network authentication protocol, many organizations still use legacy protocols that put their networks at high risk for Credential Theft. To help organizations protect their networks, we've compiled decades worth of field research to give Network Admins everything they need to know to get started with EAP-TLS.

Table of Contents

What is EAP-TLS? 4

How does EAP-TLS work? 4

Is EAP-TLS secure? 4

Without EAP-TLS, Credentials are Easily Stolen 5

Man-in-the-Middle (MITM) Attacks 5

Risks of WPA2-PSK 6

TTLS-PAP Sends Credentials in Plaintext 6

PEAP-MSCHAPv2 is Cracked 7

Prerequisites to Setting up EAP-TLS Authentication 7

Setting Up a Public Key Infrastructure (PKI) 8

What is a PKI? 8

Integrating your Identity Provider for Certificate Issuance 8

Generating Certificate Authorities 9

Can I use my Active Directory Certificate Services Certificate Authority? 10

Certificate Revocation Lists 10

Configuring your Existing RADIUS Server for EAP-TLS 11

Upload your Certificate Authorities and Certificate Revocation Lists 11

The Cloud RADIUS Advantage 12

Architected from the Ground-Up for EAP-TLS 12

Unique, Certificate-Based Network Access Policies 13

Certificate Issuance, Revocation and Configuring EAP-TLS Network Settings on Devices 14

BYOD Certificate Issuance through Device Onboarding 14

Configuring the Onboarding SSID 15

Issuing EAP-TLS Certificates to Managed Devices 16

Creating Gateway APIs and Keys 17

Configuring your MDM for Certificate Auto-Enrollment 17

FAQ 18

What's the difference between PEAP and EAP-TLS? 18

What's the difference between EAP-TTLS/PAP and EAP-TLS? 20

What is the Difference Between a Private and Public Key? 21

Will User Emails be Secure? 21

What RADIUS CA Should be Uploaded in the Network Profile? 22

iOS Device Restrictions 22

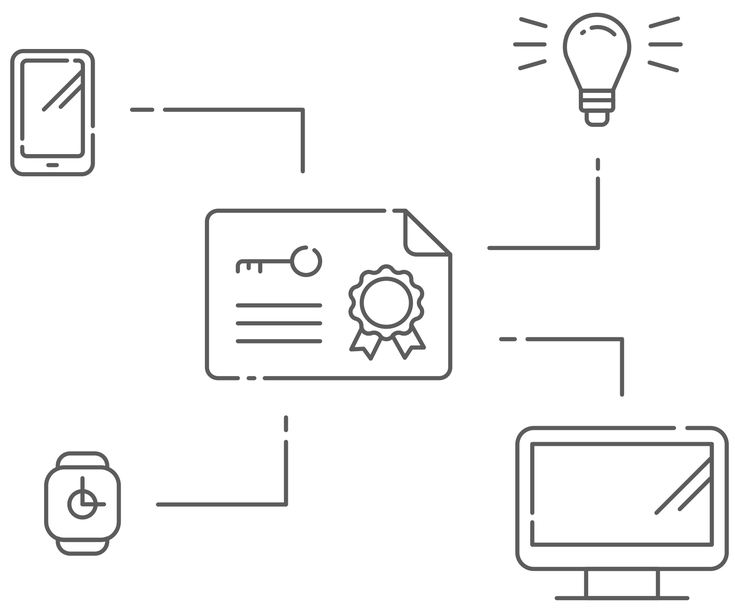
What is EAP-TLS?

Extensible Authentication Protocol - Transport Layer Security (EAP-TLS) is an IETF open standard that's defined in RFC 5216. More colloquially, EAP-TLS is the authentication protocol most commonly deployed on WPA2-Enterprise networks to enable the use of X.509 digital certificates for authentication.

EAP-TLS is considered the gold standard for network authentication security, but despite being universally recognized as ultra-secure, it's still not widely implemented. That's largely because EAP-TLS was developed before the industry had the mature device onboarding solutions necessary for smooth device configuration at an enterprise-scale.

Fortunately, EAP-TLS is much more viable with automated device onboarding solutions for BYOD and MDM like SecureW2's JoinNow MultiConnector. Streamlined onboarding makes configuration a simple task, making premium Wi-Fi security viable for a larger proportion of organizations.

How does EAP-TLS work?



Despite being the pinnacle of authentication security, EAP-TLS remains a relatively simple framework for authentication. It doesn't rely on overly complicated encryption schemes or anything like that - it's predicated on the strength of public key cryptography.

Public key cryptography is a type of asymmetric cryptography that uses public-private key pairs to establish symmetric cryptography over an unsecured channel, removing the need to securely communicate a pre-shared key beforehand.

Practically speaking, TLS architecture is similar to most other EAP-types except that it requires mutual authentication via client-side certificates. X.509 digital certificates are incredibly versatile - they dramatically enhance user experience and security and can be configured to facilitate SSO for many services.

Is EAP-TLS secure?

EAP-TLS is widely regarded as the most secure authentication protocol for 802.1X networks. The requirement for mutual certificate authentication has kept the protocol not just relevant, but dominant, for over 15 years.

One of the primary security benefits of EAP-TLS networks is the ability to perform server certificate validation. This technique renders your users all but invulnerable to common over-the-air attacks like the notorious man-in-the-middle attack.

Those hacks usually rely on spoofed access points (AP) to fool users' devices into automatically attempting to authenticate to the fake AP. The device will automatically submit real credentials without the user's knowledge, allowing the hacker to farm credentials with little effort.

Server certificate validation requires both the client and the server to validate their identity, so a device configured for EAP-TLS authentication (and thus server certificate validation) won't ever mistake a spoofed AP for the real one.

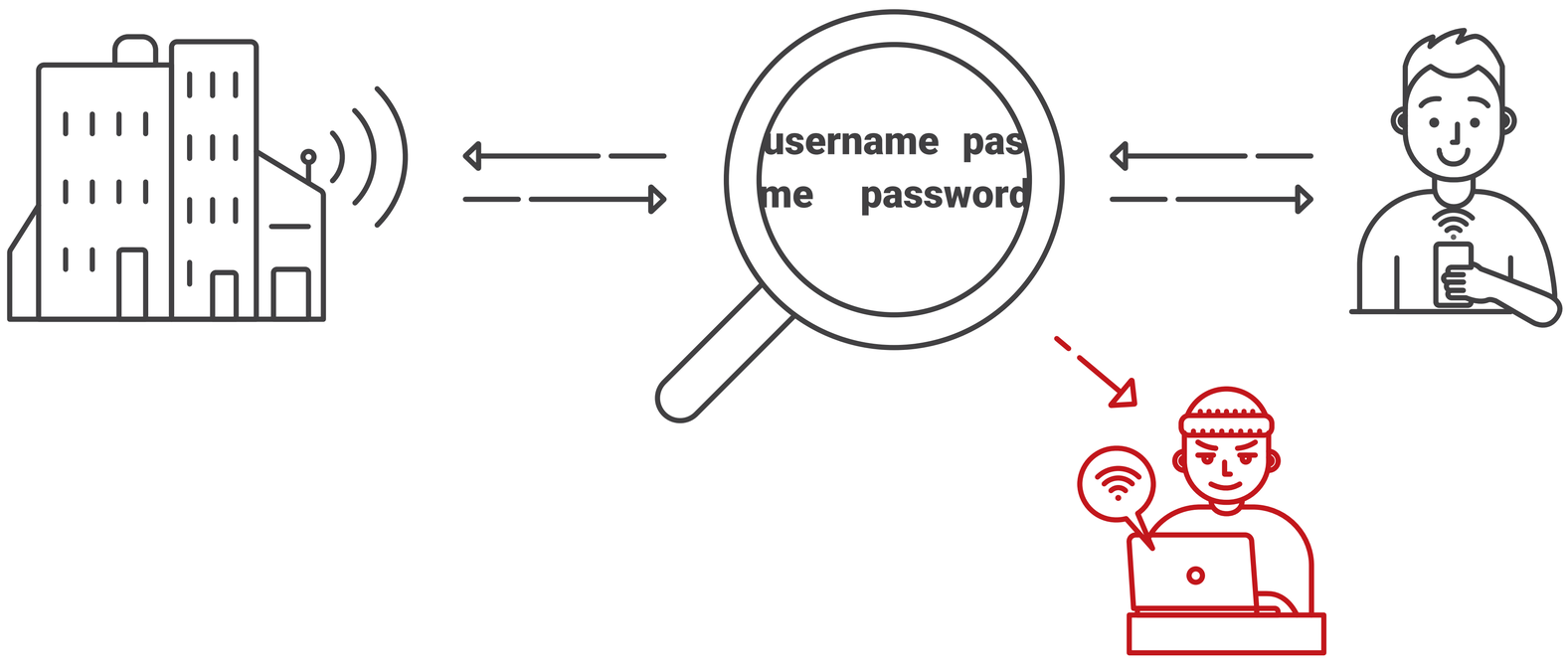
That mutual authentication requirement is fundamental to all certificate-based authentication; the true source of EAP-TLS's strength. Certificates perform similar roles in other secure applications like S/MIME and digital signature signing.

Without EAP-TLS, Credentials are Easily Stolen

According to the 2018 Credential Spill Report, an average of 1 million credentials were exposed daily in 2017, with no indication that this number will decrease.

Keeping your credentials secure is a difficult task, one that only becomes more difficult as the number of your users and applications grow.

Activities such as Wi-Fi and VPN authentication are some of the biggest contributors to credential theft, exposing tens of millions of credentials over-the-air every day. EAP-TLS is the only WPA2-Enterprise authentication protocol that is effectively immune to credential-theft because it uses ironclad X.509 digital certificates in lieu of credentials.



Man-in-the-Middle (MITM) Attacks

Man-in-the-Middle attacks are a common type of over-the-air attack simply because it's an easy attack vector. According to IBM's X-Force Threat Intelligence Index, 35% of exploitation activity involves Man-in-the-Middle attacks.

Why are MITM attacks so easy to perpetrate? For once, the fault doesn't lie with the end-user. The problem is almost always an under-secured or misconfigured network.

The simplest and most effective method to prevent MITM attacks is by configuring and requiring server certificate validation. Wi-Fi-enabled devices automatically attempt to connect to familiar SSIDs, including spoofed ones, by sending login credentials. Using EAP-TLS certificate-based authentication both positively identifies each party and also prevents private keys from being exposed at any point during authentication.

Risks of WPA2-PSK

That WPA2-PSK is not sufficient for enterprise network security is hardly a new idea, but it bears repeating. A single shared credential that protects all of the resources of a network (and organization) is a disaster waiting to happen.

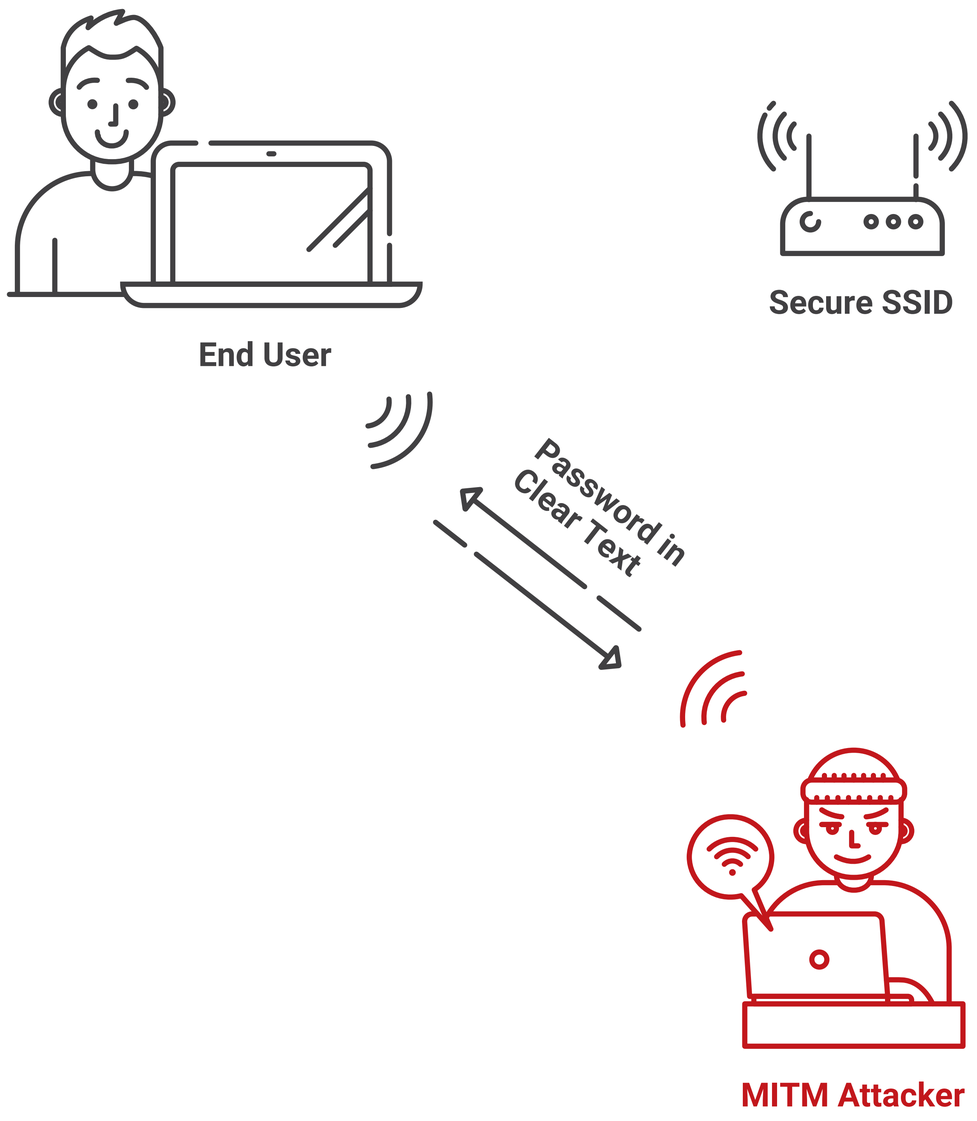
There is a persisting misconception that organizations working primarily in the cloud can scrape by with PSK since their sensitive data is not stored on the local network anyway. That false sense of security is dangerous - any level of network access is a stepping stone to deeper penetration.

At the point where a hacker has breached your office Wi-Fi, they can enact any number of Layer 2 attacks such as:

- Address Resolution Protocol (ARP) Attacks
- Content Addressable Memory (CAM) Table Overflows
- Spanning Tree Protocol (STP) Attacks
- Media Access Control (MAC) Spoofing
- Switch Spoofing
- Double Tagging
- Cisco Discovery Protocol (CDP) Reconnaissance
- Dynamic Host Configuration Protocol (DHCP) Spoofing

Given the unceasing improvements to processing power, brute force attacks are increasingly effective against PSK.

TTLS-PAP Sends Credentials in Plaintext



The primary difference between EAP-TTLS and EAP-TLS is that the former only requires server-side certificates rather than the mutual certificate authentication that characterizes EAP-TLS.

In order to compensate, TTLS uses a tunnel (hence “Tunneled” Transport Layer Security). It’s essentially an SSL wrapper encapsulating the standard EAP message. It does not encrypt the message, however.

Unfortunately, this means that EAP-TTLS communicates everything (including credentials) in clear text. And, since TTLS does not use server certificate validation, it’s particularly susceptible to being intercepted in over-the-air attacks… which completely bypass the tunnel and deliver credentials in clear text directly into the waiting hands of a hacker.

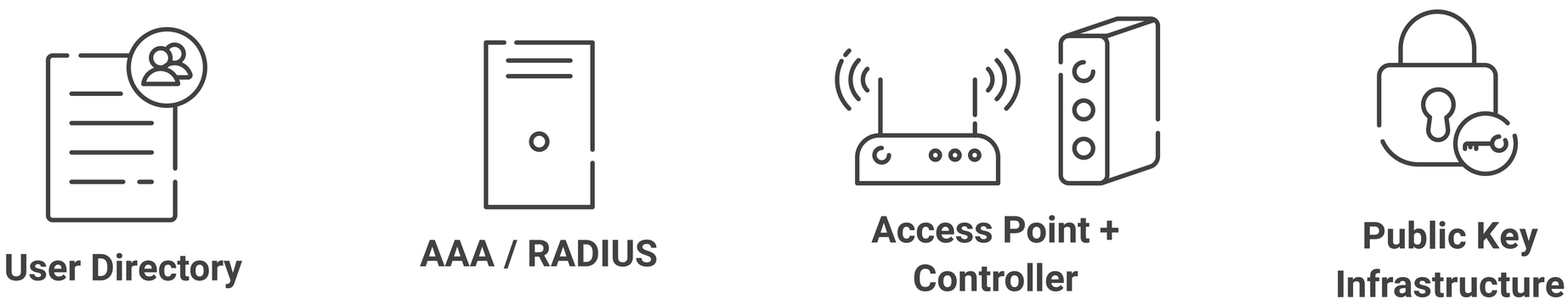
PEAP-MSCHAPv2 is Cracked

PEAP-MSCHAPv2, a widely supported standard, can be exploited to gain user login information from devices which are not properly configured to connect only to trusted RADIUS servers. A well-documented weakness in its encryption method allows the attacker to easily decrypt the packets, potentially gaining user credentials.

Unfortunately, this means MSCHAPv2 is highly susceptible to man-in-the-middle attacks and has been for more than a decade now. This is particularly crippling in an era where hackers are taking the time to be more hands-on. It's not difficult to imagine a dedicated hacker spoofing your office access point from a van parked outside your house, a scenario made all the more likely as C-suite level executives are certainly working from home during the pandemic.

EAP-TLS entirely circumvents this issue by using server certificate validation. Even in the scenario described above, it's relatively simple to use a VPN to secure a connection to your office RADIUS server for remote authentication.

Prerequisites to Setting up EAP-TLS Authentication

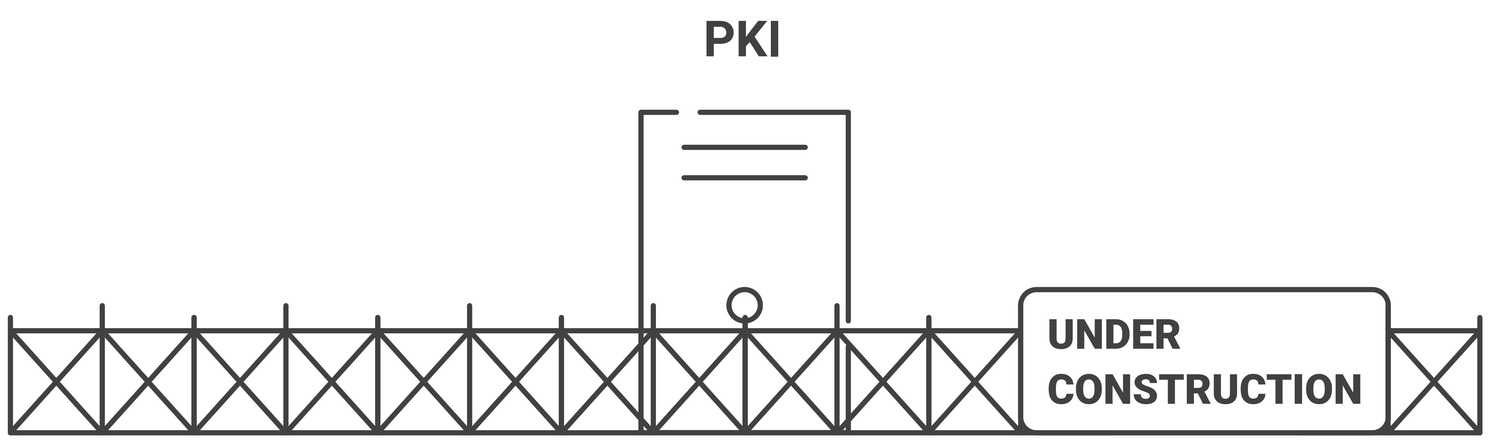


The minimum required infrastructure for EAP-TLS authentication is:

- AAA/RADIUS
- User Directory
- 802.1x Capable Access Point and Controller
- Public Key Infrastructure (PKI)

Of course, that short list makes it look much simpler than it is. Just the PKI alone is composed of more than a dozen different components, each totally vital to the process of configuring, provisioning, revoking, and otherwise managing digital certificates.

Setting Up a Public Key Infrastructure (PKI)



Before you can start configuring devices for EAP-TLS, we first need the infrastructure required to generate, manage, and revoke x.509 Digital Certificates. This infrastructure is commonly referred to as a PKI. There are ample open-source or otherwise free resources to cobble together a functional Public Key Infrastructure, however, the cost of using poorly engineered infrastructure only increases as usage grows. Before we dive deep into the best way to set up a PKI, first let's cover what a PKI is.

What is a PKI?

The purpose of a PKI is to manage the public keys used by the network for public key encryption, identity management, certificate distribution, certificate revocation, and certificate management. Once enabled, users who enroll for a certificate are identified for later authentication or certificate revocation.

The PKI allows users and systems to verify the legitimacy of certificate-holding entities and securely exchange information between them over the air. The introduction of a PKI enables stronger, certificate-based security, as well as identity services and management tools to maximize network efficiency and security.

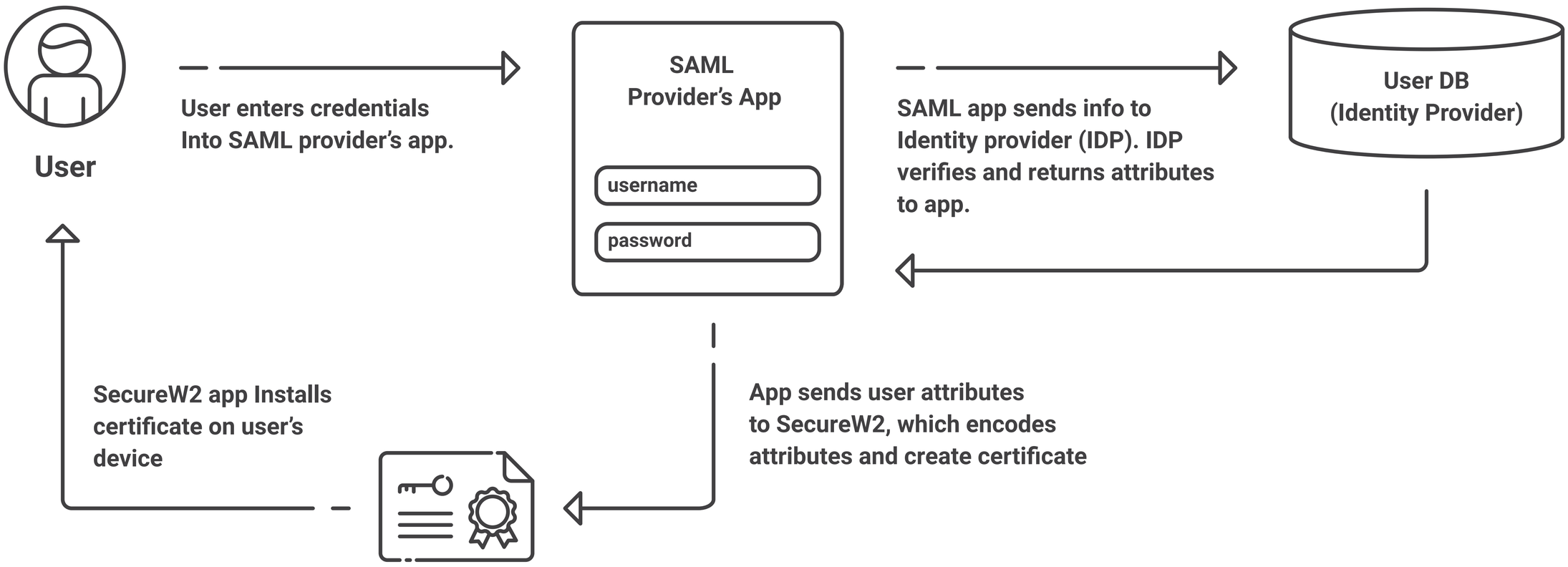
Ensuring your PKI is an effective mechanism for generating derived credentials requires a tight integration with your Identity Provider (IDP), also known as a directory. Your IDP will be used to verify identities, required before issuing a certificate, as well as performing a secondary identity lookup during the authentication phase.

Integrating your Identity Provider for Certificate Issuance

SecureW2 acts as an authority to verify user identities and issue X.509 certificates. It is able to do this, by using the SAML protocol to talk directly to an Identity Provider.

When a user enters their credentials in the SAML application, the identity provider verifies the user's identity and returns attributes for the user. These attributes serve as network rules that determine the user's access rights, which network segments/resources they can use, and more. SecureW2 encodes these attributes on the certificate it issues, and then installs the certificate on the user's device. This allows a BYOD user to easily and securely self-service themselves for a certificate.

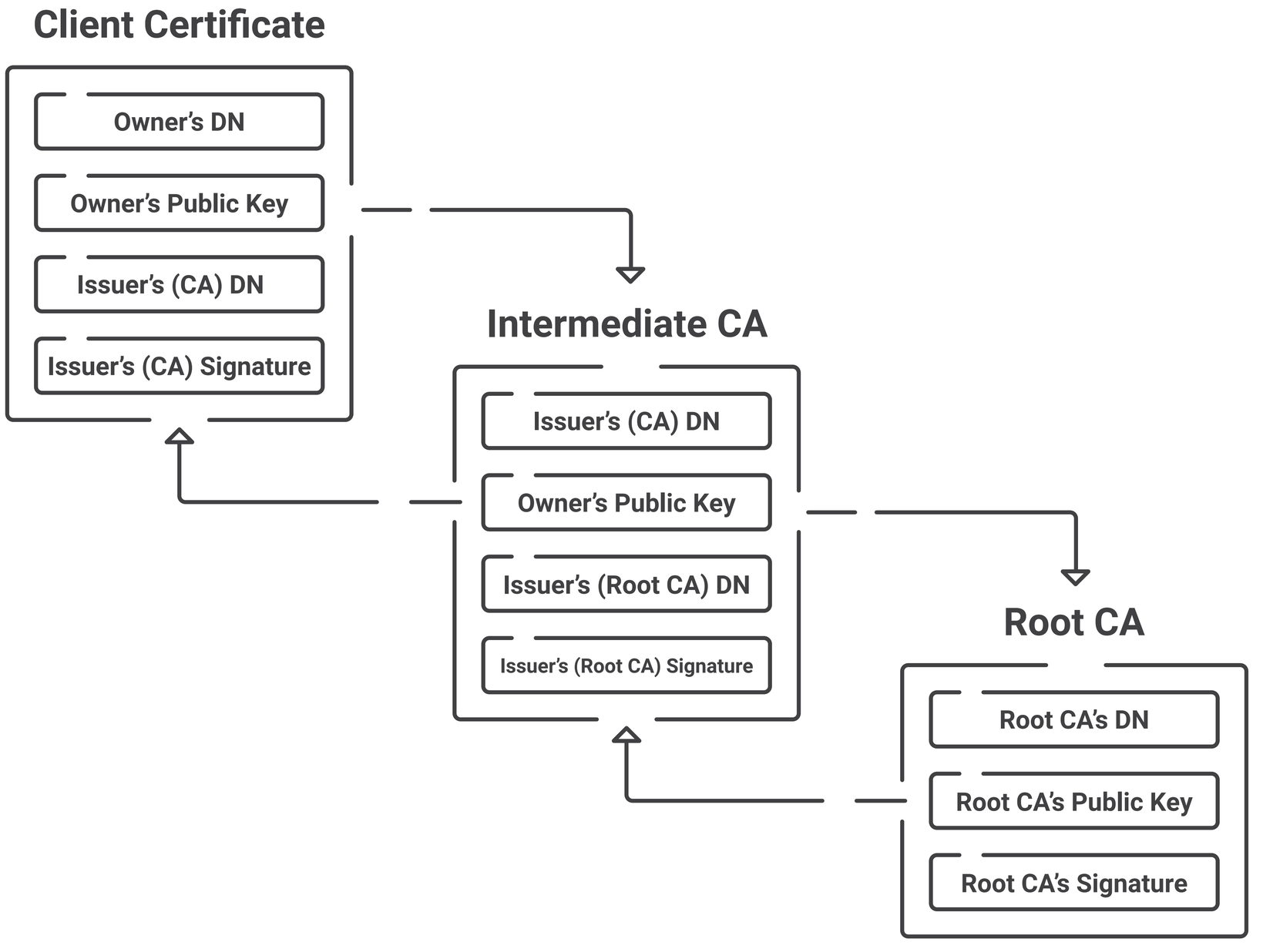
The following diagram shows the general flow of information when using SAML with SecureW2:



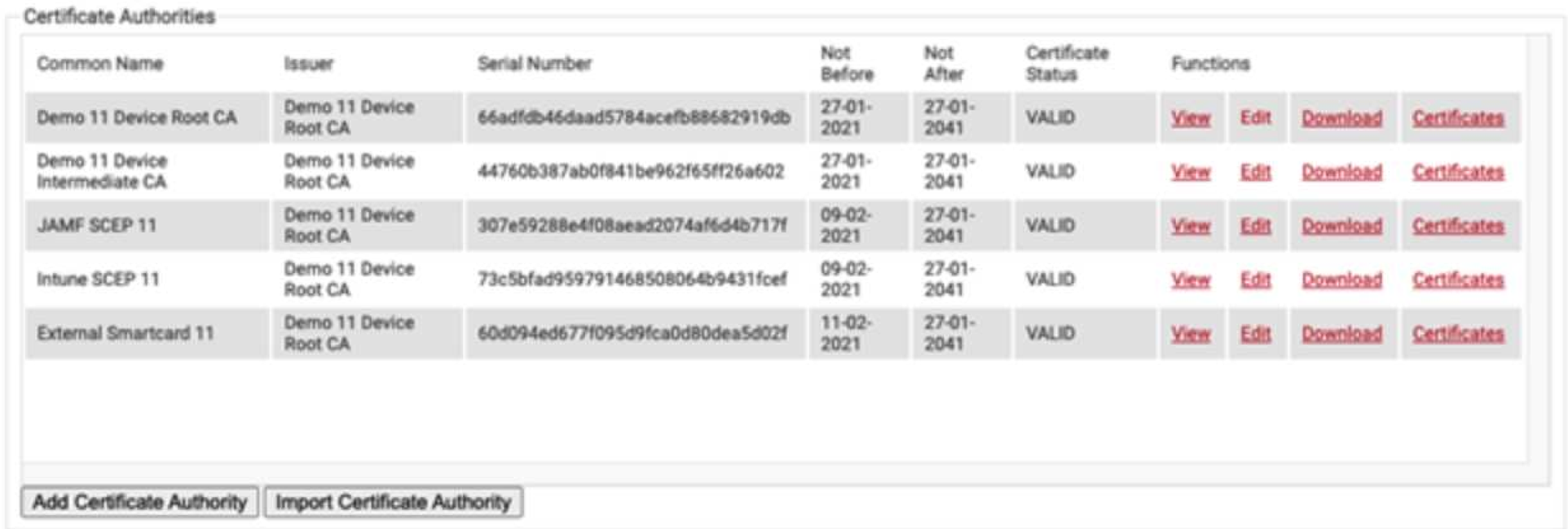
SecureW2 also allows you to integrate your Identity Provider using OAuth, which allows your Identity Provider to talk directly to the SecureW2 Cloud RADIUS during the authentication phase to perform an Identity Lookup. Not only does SecureW2 enable EAP-TLS certificate-authentication, but it further secures network authentication by looking up user, group, and device information as a secondary check.

Generating Certificate Authorities

Certificate Authorities (CA) are the heart of a PKI. They are the trusted authority that all our device's client certificates are derived from. Our authenticator, in the case of EAP-TLS is our RADIUS Server, is where our CA resides.



The CA has two very important purposes in the context of EAP-TLS authentication. The first, is it allows us to perform Server Certificate Validation. Our devices will be set up so they only attempt authentication to a RADIUS server that has a CA issued to our domain. The second, is that it will perform the public-private key exchange with our device's client certificate.



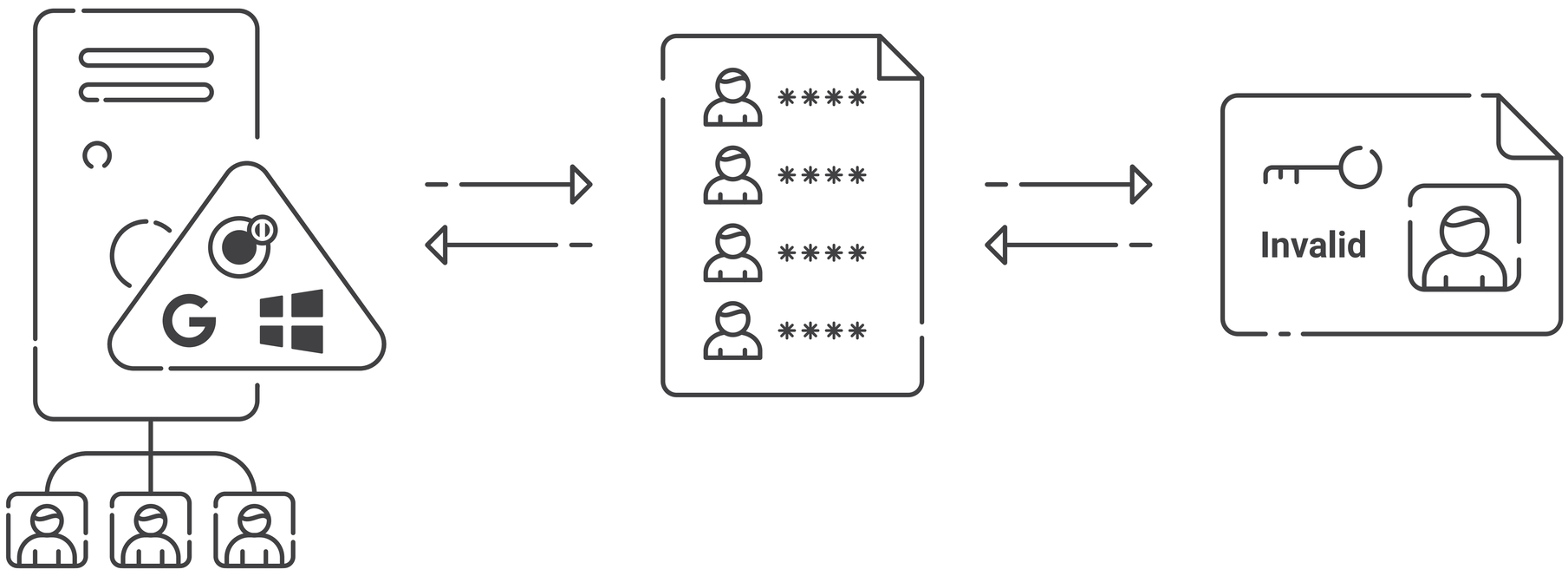
WIth SecureW2, you can easily create and import Root and Intermediate Certificate Authorities.

Can I use my Active Directory Certificate Services Certificate Authority?

An AD CS CA was historically one of the most common methods of issuing certificates. While outdated, some organizations still use their AD CS CA as the backbone of their PKI.

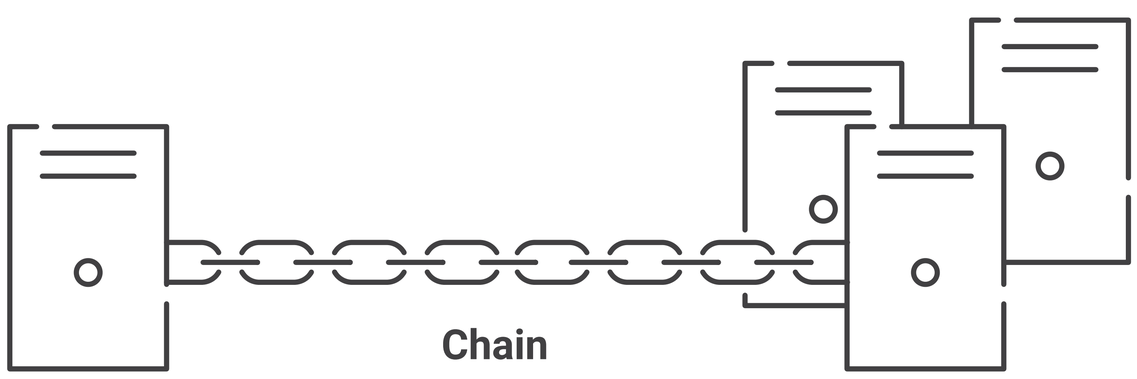
With SecureW2's PKI Services, an AD CS CA can be used in combination with our modern management and issuance services for applications such as EAP-TLS. For more information around this, please contact **sales@securew2.com.**

Certificate Revocation Lists



Certificate Revocation Lists (CRL) are lists of certificate serial numbers that belong to revoked certificates. The RADIUS server checks a device's certificate to see if it contains a serial number that exists on the CRL

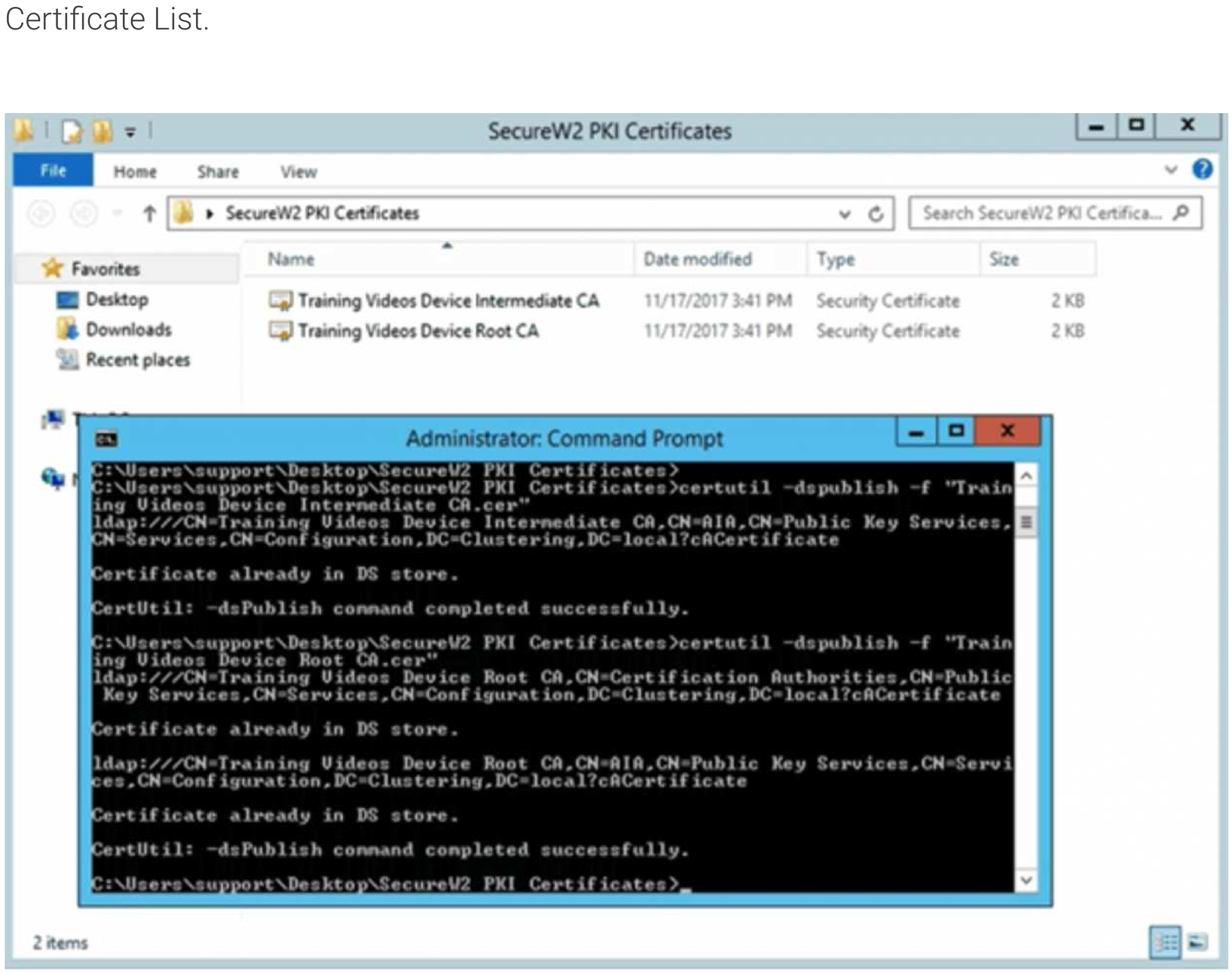
Configuring your Existing RADIUS Server for EAP-TLS



Configuring your RADIUS server for basic EAP-TLS authentication isn't extremely difficult. It consists of uploading your CA's and CRL, and then configuring Identity Lookup should your server and environment support it. However, many legacy on-premise servers that support EAP-TLS aren't perfect, many of them don't support modern cloud authentication. Below we've detailed a few.

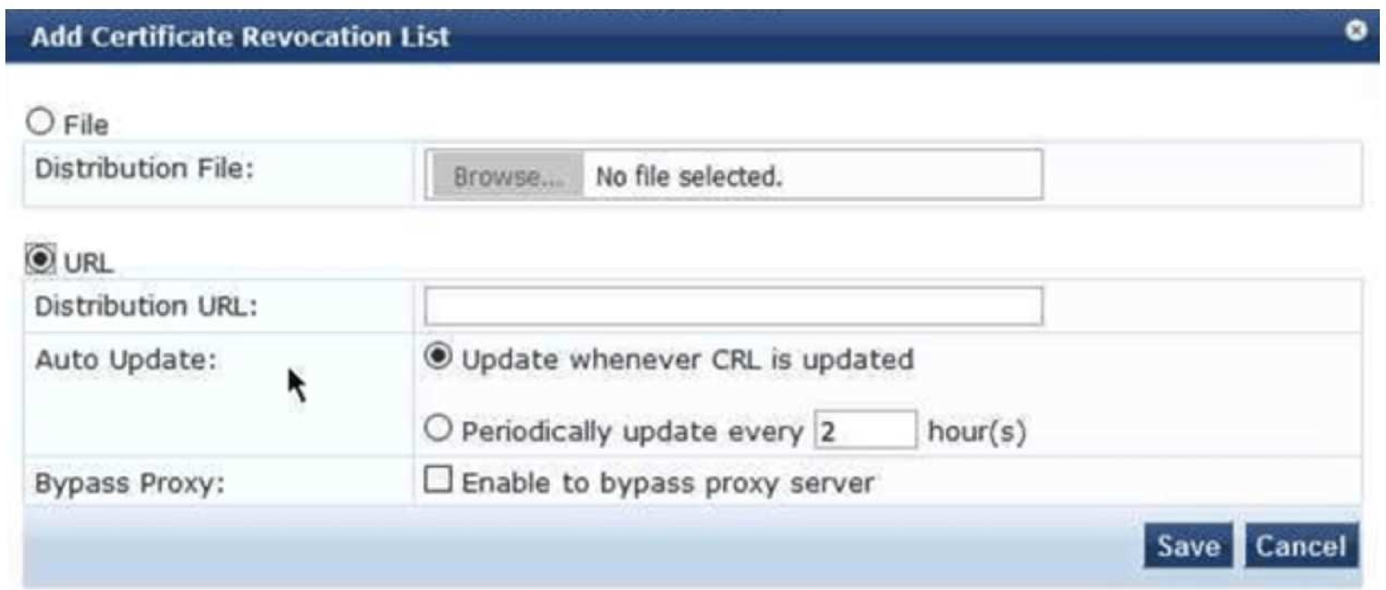
Upload your Certificate Authorities and Certificate Revocation Lists

The first part of integrating your PKI with your RADIUS server, is to upload your Certificate Authorities and CRL. Typically in RADIUS servers such as Cisco ISE or Aruba CPPM, your CA's will be exported as a .p12 file and uploaded in a Trusted Certificate List.



The screenshot above shows the CLI in Microsoft NPS where you install the Root and Intermediate CA.

Certificate Revocation Lists are easily integrated into the RADIUS server by sharing their URLs. You can easily grab the CRL by clicking view on your CA in the SecureW2 management portal. Below is a screenshot of how to enter in a CRL in Aruba CPPM RADIUS Server.

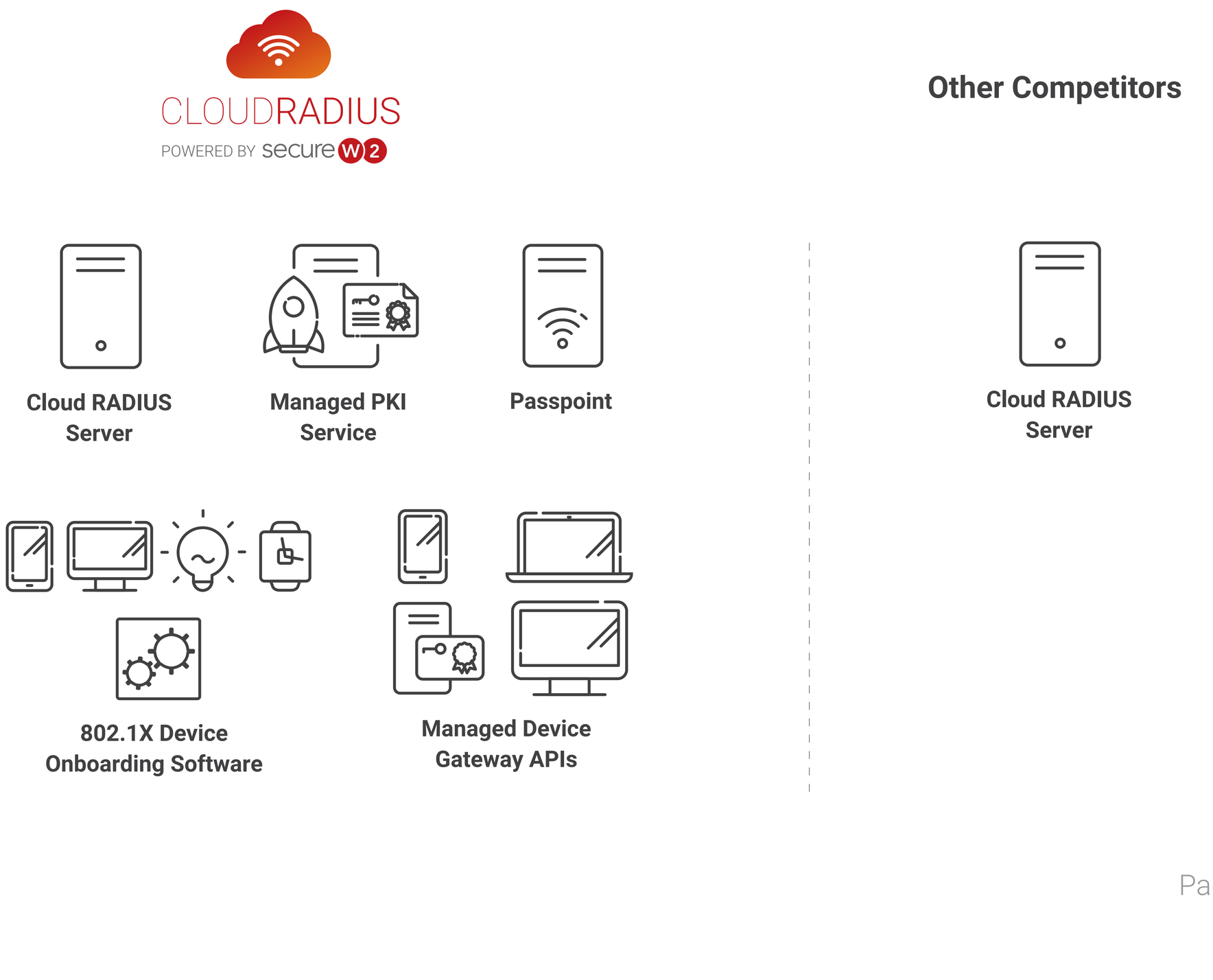


For detailed information on how to integrate with your specific RADIUS server, click here to find our integration guides for Cisco ISE, Aruba CPPM, Extreme Networks, and more.

The Cloud RADIUS Advantage

The easiest way to set up a RADIUS server for certificate-based authentication is to use a RADIUS server that was designed for it. SecureW2's Cloud RADIUS server is the only AAA server that was designed from the ground up exclusively for EAP-TLS, which provides organizations a vast number of advantages over alternative servers.

Architected from the Ground-Up for EAP-TLS

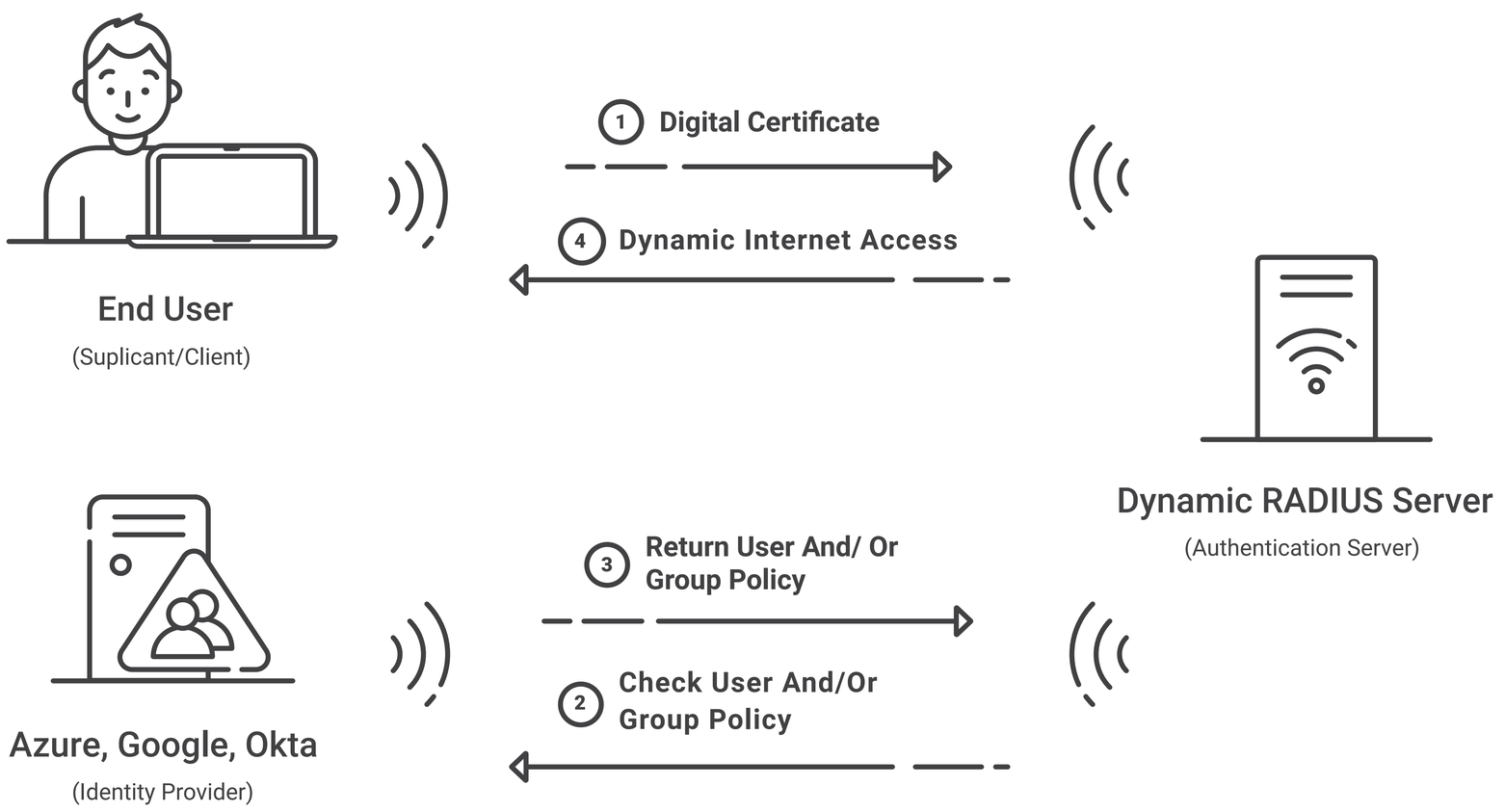


Most popular RADIUS vendors developed their product at a time when networks were primarily on-premise. Aging architecture, like Active Directory environments, have caused massive problems for organizations trying to transition to the cloud.

SecureW2's Cloud RADIUS is the only product in its class that was designed from the ground up for certificate-based cloud authentication. Other vendors use insecure authentication protocols that expose users to MITM attacks, but our solution uses your existing credentials to easily deploy certificates to end-users with no hassle.

Cloud RADIUS provides everything an organization needs to secure WPA2-Enterprise encrypted Wi-Fi using the EAP-TLS protocol for certificate-based authentication, including a Managed PKI service.

Unique, Certificate-Based Network Access Policies



Cloud RADIUS is also the industry's only certificate-based authentication solution with Role-Based Access Control that works natively with cloud directories like Okta, Azure, and G-Suite. Our Dynamic Policy Engine empowers you to automatically assign appropriate user and group policies at the moment of network authentication.

Automate access levels based off of users and groups for Wi-Fi, VPN, Applications, Desktop Logon, and much more. Tailor your policies with customizable attributes stored in the IDP, enabling a more rapid propagation of permissions updates/revocation than a CRL typically offers.

Similar to the User Lookup function of LDAP, Cloud RADIUS offers powerful lookup features for all types of identities: user, group, device, etc. The ability to dynamically reference an entity's directory entry during authentication enables more granular policy enforcement options to suit the needs of your network.

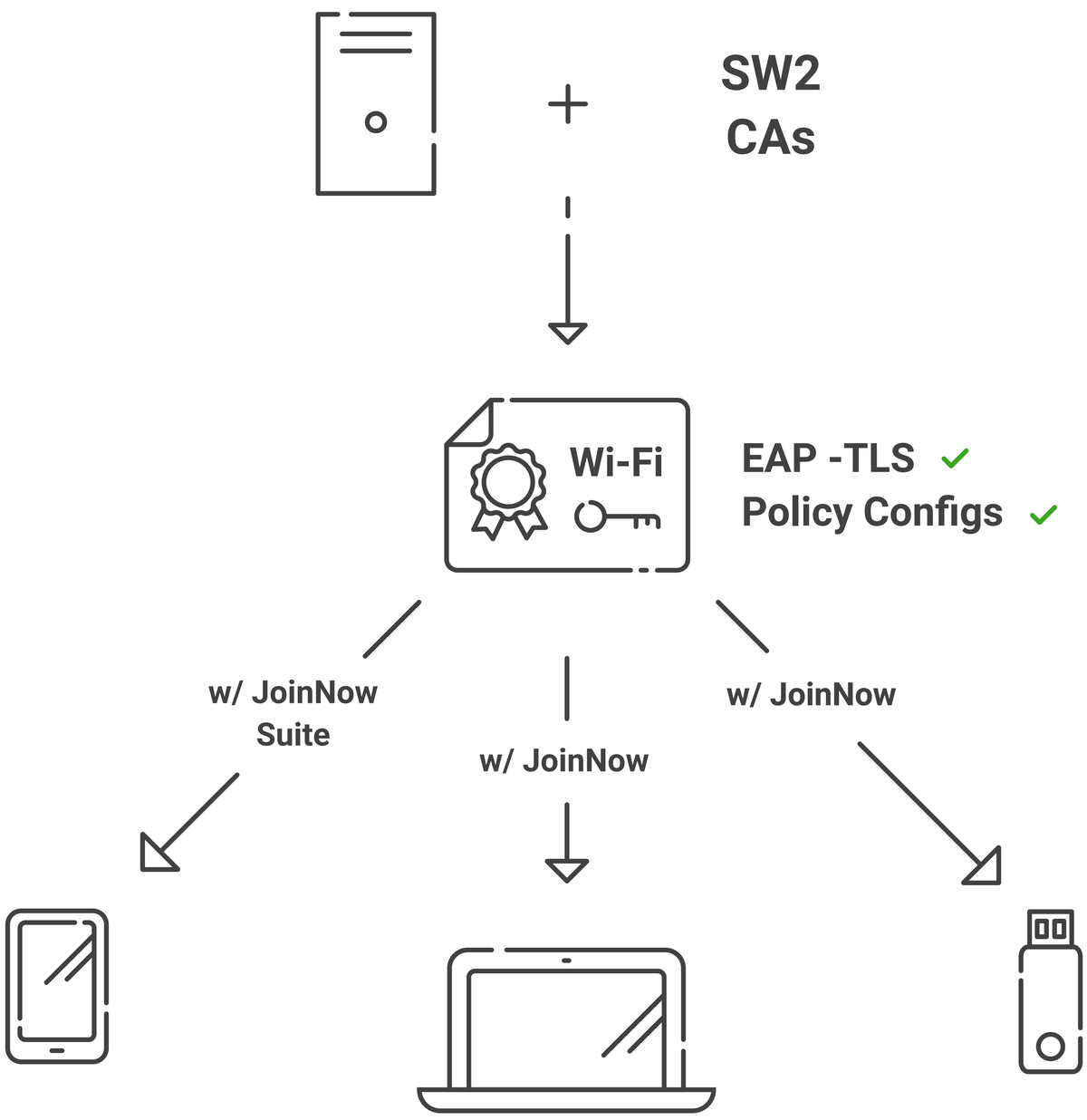
Certificate Issuance, Revocation and Configuring EAP-TLS Network Settings on Devices

Historically, the deployment and management of certificates was one of the biggest reasons why organizations would hesitate to replace their credential-based network authentication with certificates.

Today however, with the advent of the cloud alongside many advancements in cybersecurity technology, certificates are actually incredibly easy to manage. Many organizations actually switch from password authentication to certificates purely for the user experience benefits.

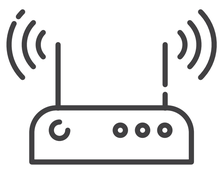
Over the past decade, we've helped organizations all across the world enroll millions of devices for certificates, and summarized the top things you should know about delivering, managing, and setting up devices for EAP-TLS certificate-based network authentication.

BYOD Certificate Issuance through Device Onboarding



In order to deploy EAP-TLS authentication for BYOD devices, you need to set up device onboarding software, which can issue certificates and configureing devices for WPA2-Enterprise network. With SecureW2, device onboarding comes standard with our PKI Services, allowing access to the industry's #1 rated BYOD Onboarding product on the market.

Onboarding a device for 802.1x and EAP-TLS is almost identical to the flow of connecting your Wi-Fi to a Hotel, or Aiport network. You connect your device to an Open SSID, a redirect happens, and then your device is configured for the network. To accomplish this, you need the following setup:



Open / Guest

Encrypted SSID

1.) Open SSID

2.) Encrypted SSID for EAP-TLS

3.) SecureW2 MultiOS Onboarding Clients or equivalent Onboarding Software

4.) SecureW2 CloudRADIUS or any 3rd party RADIUS

5.) Identity Provider

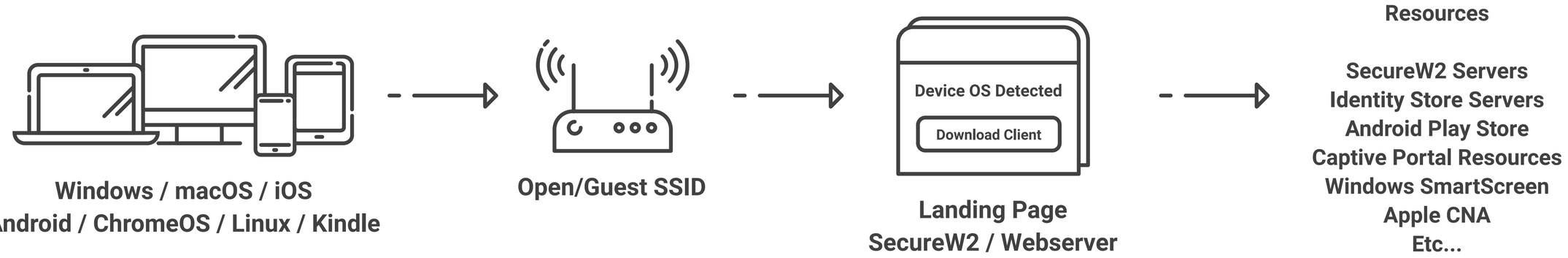
SecureW2 MultiOS Onboarding Client

SW2 Cloud RADIUS / 3rd Party RADIUS

Identity Provider

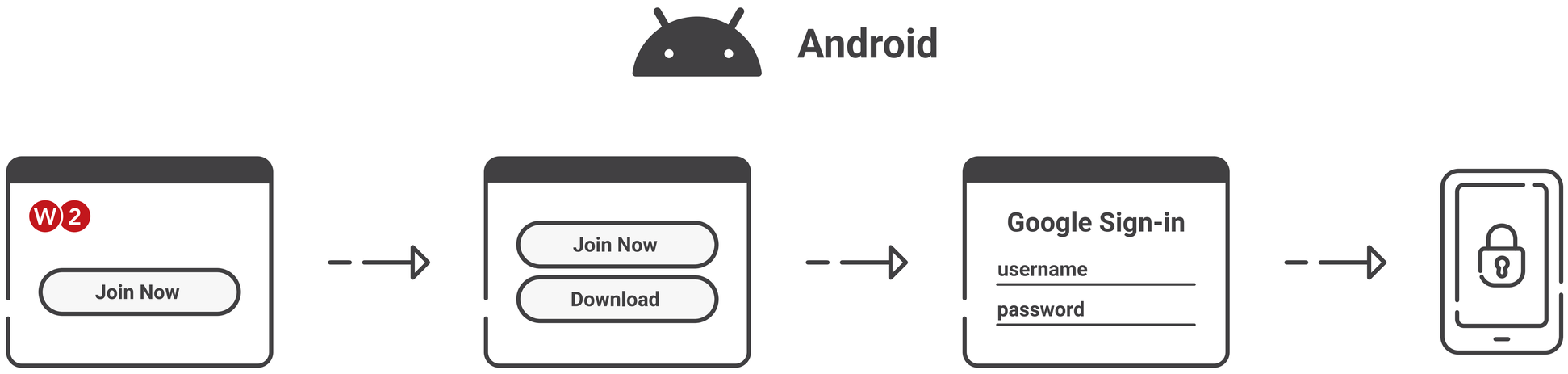
Configuring the Onboarding SSID

Configuring an onboarding SSID requires 2 major parts: configuring a redirect to the SecureW2 landing page and allowing certain resources through your network's Walled Garden. Refer to our JoinNow MultiOS Firewall Rules documentation for specific IPs we recommend adding through the Walled Garden, allowing it to communicate with SecureW2.



The landing page detects the user device's OS and deploys a client appropriate for that OS. That client will check user creds against its ID store. After authentication, the enrollment process begins. This consists of network settings and a Wi-Fi certificate installed being configured and installed on the device.

The user is now ready to authenticate against the RADIUS server on the secure SSID. The user is taken from the onboarding SSID to the secure SSID, where the certificate will be authenticated and the user is authenticated.

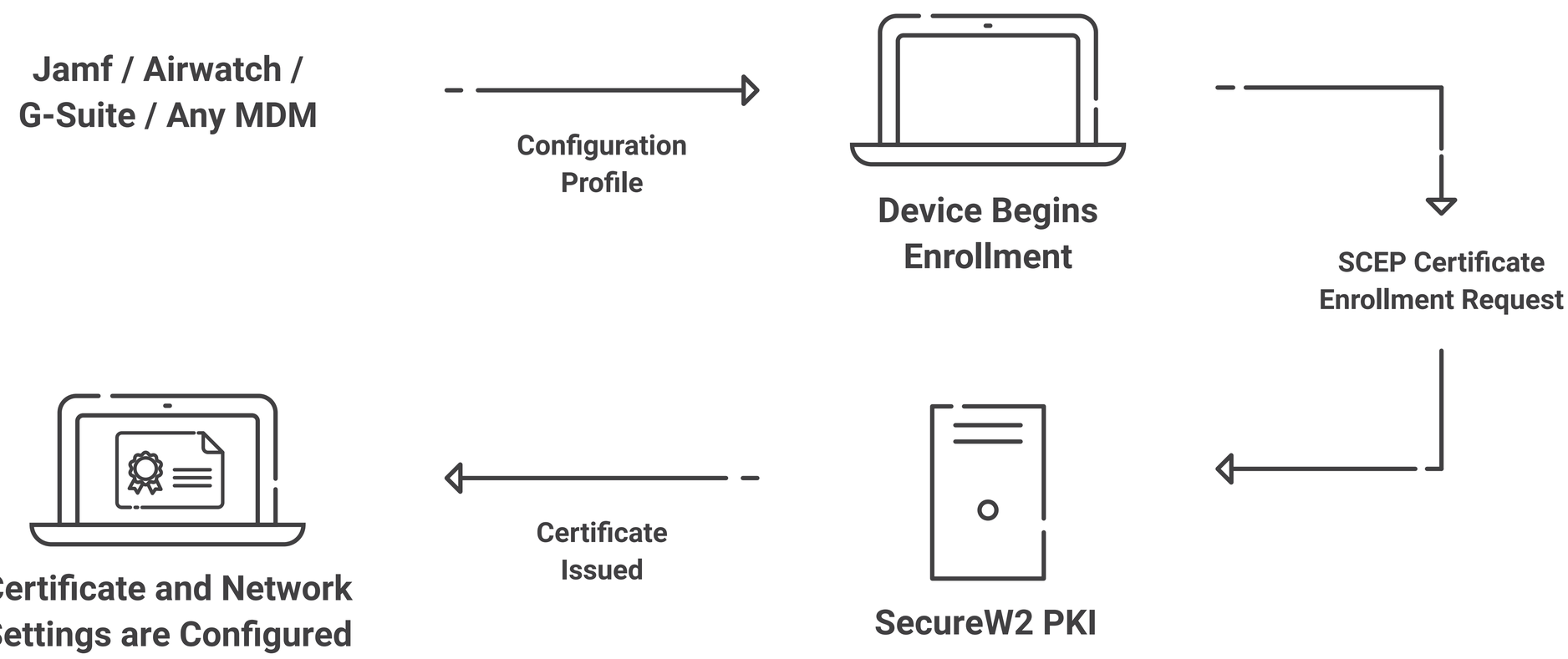


This self-service flow is significantly easier than configuring certificates and network settings manually, but there can be some caveats. One example, is that an Android Application is the most ideal way to configure end users for EAP-TLS. This is something that's a simple fix with SecureW2, as we provide the #1 Rated Android Onboarding application in the Google Play Store with our PKI services, but is something to be considered for organizations seeking to create their own Onboarding solution.

For more information on how certificates are enrolled for on our devices, **click here.**
For videos on how 802.1x is configured on every OS with our Onboarding clients, **click here.**

Issuing EAP-TLS Certificates with SCEP

To efficiently equip your managed devices with EAP-TLS certificates, the SecureW2 allows you to build a SCEP Gateway. Simple Certificate Enrollment Protocol (SCEP) allows managed devices to enroll themselves for certificates, removing the labor-intensive task of manually enrolling devices for certificates. Once a payload that contains our SCEP URL and Secret is sent out by our MDM, devices are enrolled automatically through the Gateway.



Using SecureW2's Managed Device Gateway APIs, you can easily enroll every managed device with certificates. This section covers creating a SCEP Gateway to administer EAP-TLS certificates for Jamf, Intune, Google, Meraki, and other MDMs.

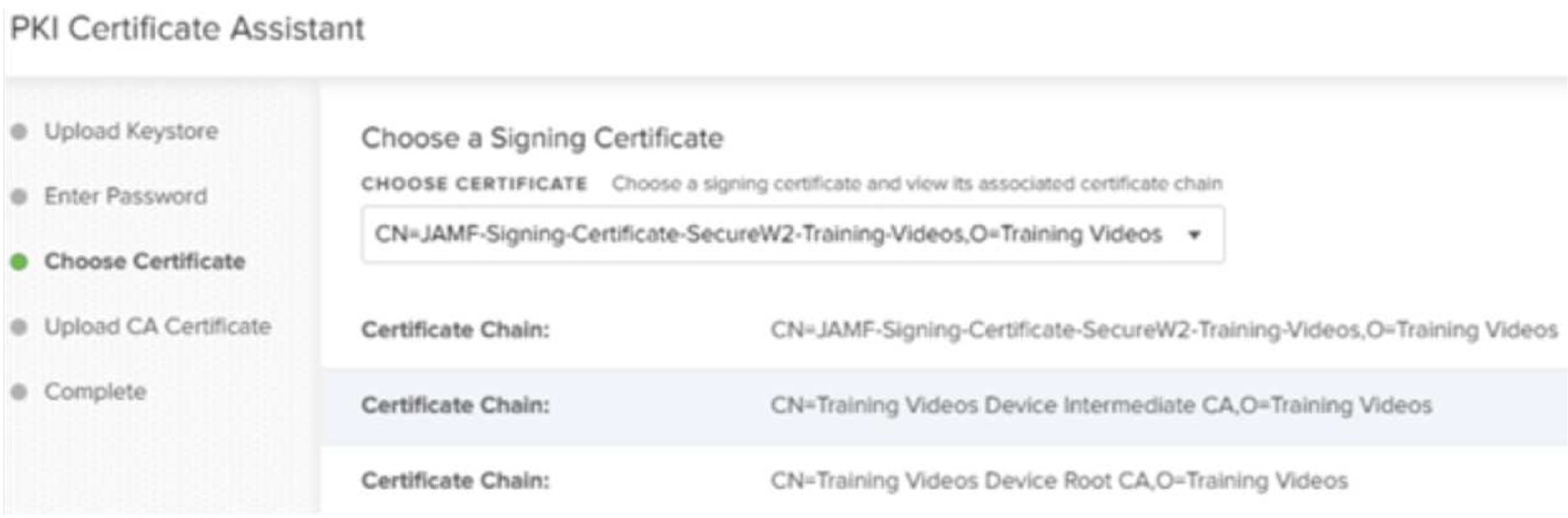
Creating Gateway APIs and Keys

Navigate to Identity Management → API Tokens in the Management portal. Once you create a new API Token, a API URL and Key will be generated for you and downloaded. This URL and key will later be pushed to your devices through your MDM to enable auto-enrollment for certificates.

Creating Gateway APIs and Keys is really simple with SecureW2, but there can be some nuances, particularly when it comes to XML vs non-XML based MDMs. For more details on our MDM integrations **click here.**

Configuring your MDM for SCEP

Once you've created a API URL and Key, you can now create a payload within your Mobile Device Management (MDM) software that will instruct your managed devices to use this URL and Key to auto-enroll themselves for certificates.



Typically with MDMs, you will create a few profiles. A profile, such as a SCEP Profile, that contains the URL and key. A couple Trusted Certificate profiles that contain your RADIUS Server and the Issuance Root/Intermediate CA. Finally a Wi-Fi profile that will instruct your device to use EAP-TLS Wi-Fi properly. If you'd like to see detailed integration guides, we have them for most MDMs, **which can be found here.**

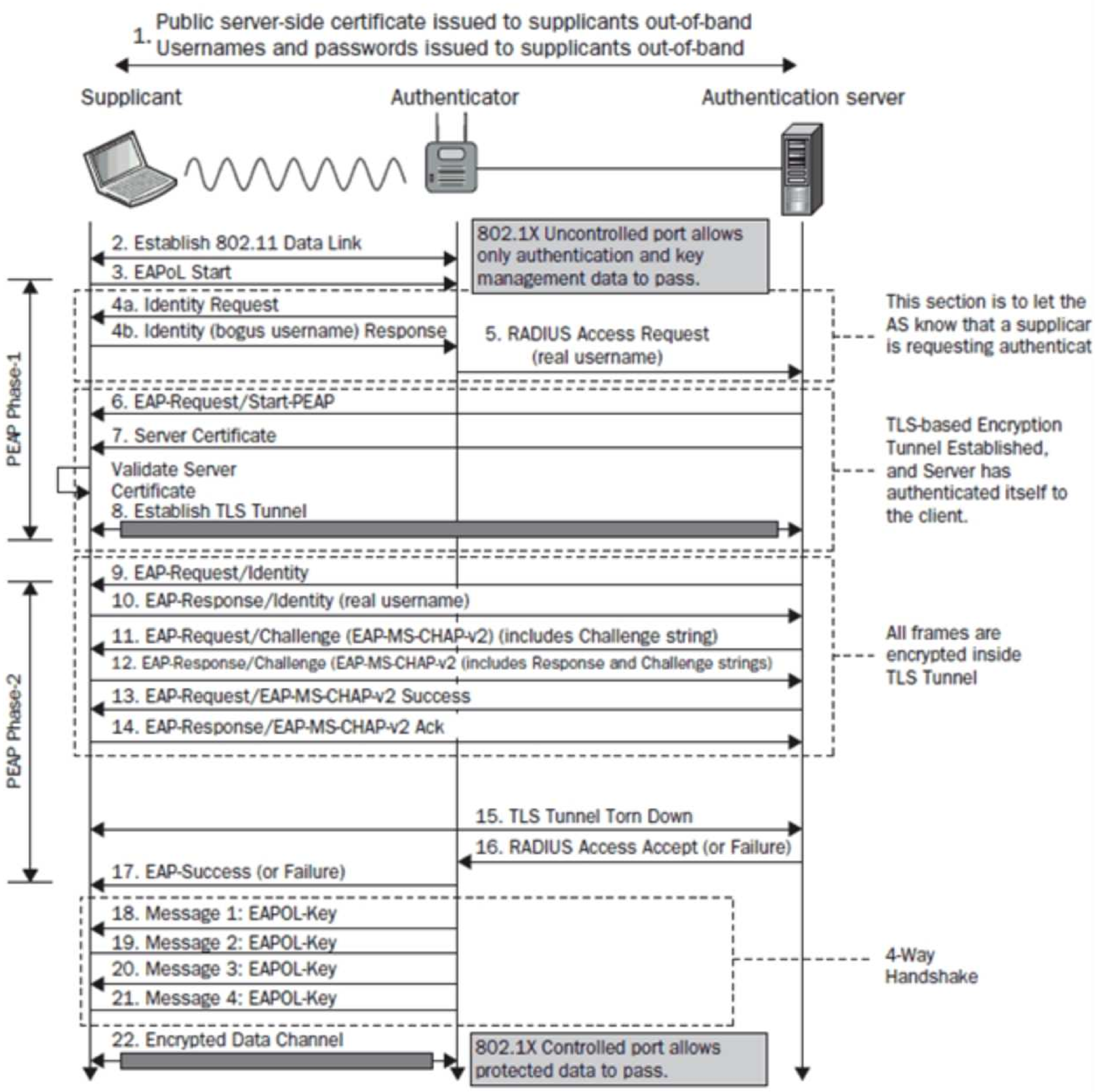
FAQ

What’s the difference between PEAP and EAP-TLS?

Both protocols are considered EAP methods, so they each send identifying information through the encrypted EAP tunnel. This encrypted tunnel prevents any outside user from reading the information being sent over-the-air.

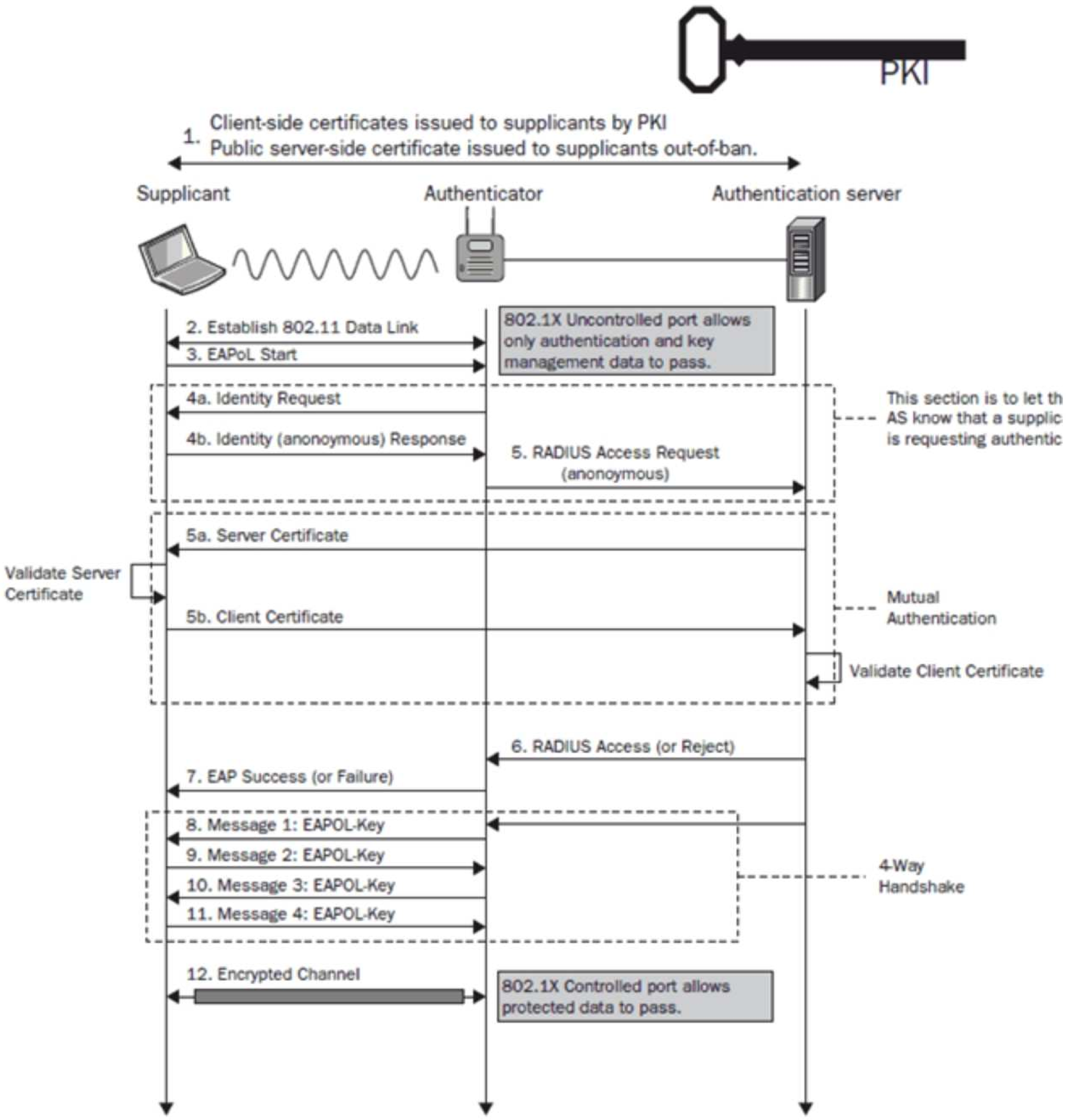
However, the process for the end user differs significantly between the two protocols. With PEAP-MSCHAPv2, the user must enter their credentials to be sent to the RADIUS Server that verifies the credentials and authenticates them for network access.

FIGURE 4.27 EAP-PEAP process



EAP-TLS utilizes certificate-based authentication. Rather than sending credentials to the RADIUS Server over-the-air, credentials are used for a one-time certificate enrollment, and the certificate is sent to the RADIUS server for authentication.. Over the course of the user’s lifetime with the organization, being able to auto-authenticate without having to memorize a password or update due to a password change policy is a huge benefit to the user experience.

FIGURE 4.29 EAP-TLS process

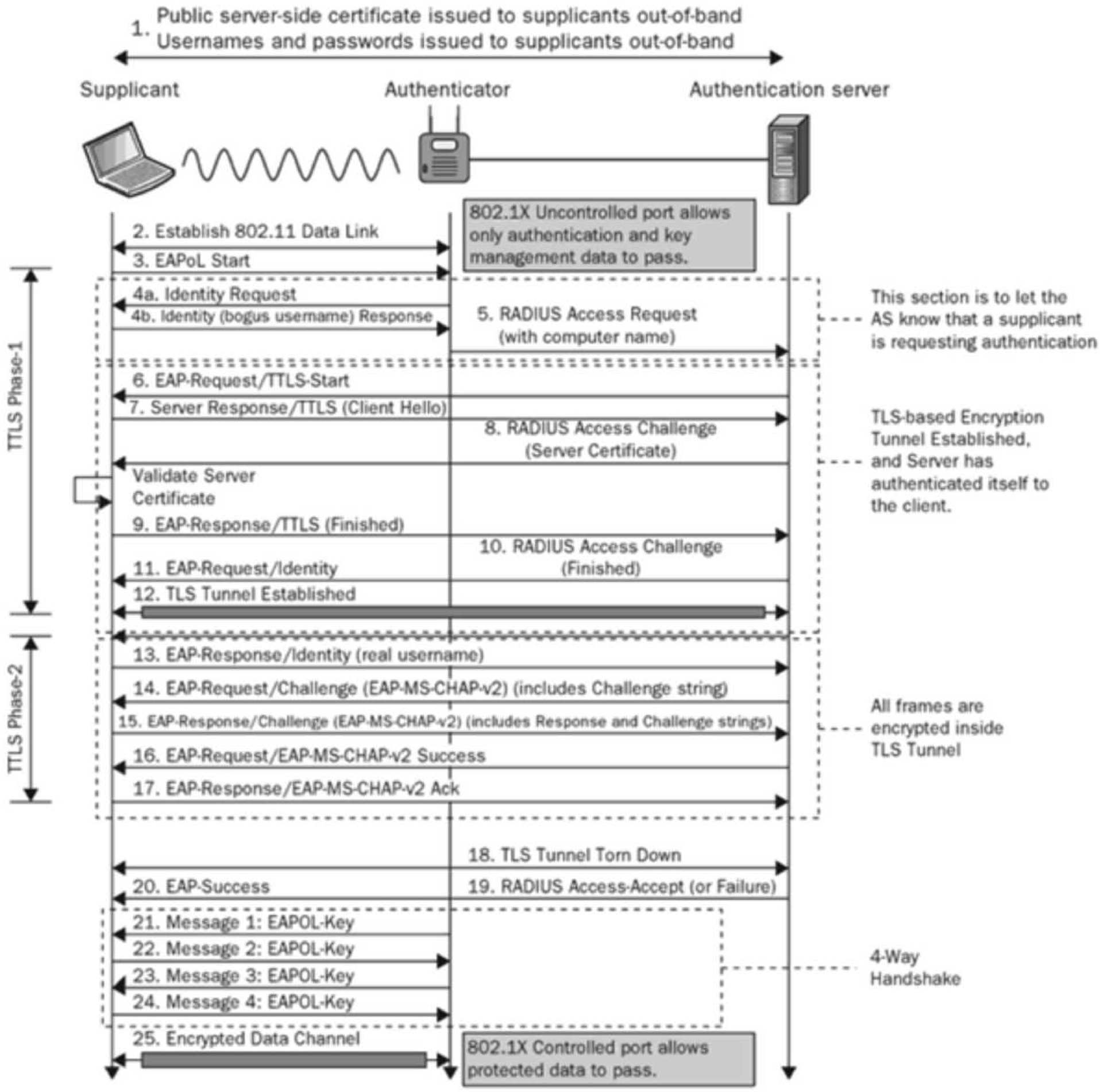


While the information exchanged between the client device, Access Point (AP), and RADIUS server may be different between EAP-TLS and PEAP-MSCHAPv2, they both undergo a TLS Handshake. This is the communication process in which the server and client exchange identifying information. The two sides will verify one another's identity, establish encryption algorithms, and agree on session keys to securely authenticate to the network.

What's the difference between EAP-TTLS/PAP and EAP-TLS?

The primary difference between EAP-TTLS and EAP-TLS is that the former only requires server-side certificates rather than the mutual certificate authentication that characterizes EAP-TLS.

FIGURE 4.28 EAP-TTLS process



What is the Difference Between a Private and Public Key?

The public key is shared around the network and can be used to encrypt messages directed to the corresponding user. We are ok with anyone and everyone knowing what our public key is. The private key is securely stored in a Hardware Security Module. The private key is generated on the device, and should never leave the device under any circumstances, as it gives access to the network. This is why SecureW2 comes with exclusive CertLock technology, which prevents private keys from being exported from devices.

Will User Emails be Secure?

User emails are input in RFC 822 SAN attribute and secured through Server Certificate Validation, which means the client will only establish a connection with the RADIUS server that has a certificate that matches your root CA.

What RADIUS CAS should be Uploaded in the Network Profile?

Included in your network profile, we upload Digicert because it's the root CA that issues RADIUS server certificates. If you use an external RADIUS Server such as ISE or CPPM, then a RADIUS server certificate will need to be uploaded in the network profile.

iOS Device Restrictions

Apple has set restrictions which can prevent your portal from capturing MAC addresses. MAC Address Randomization has crippled many organizations, who relied on MAC addresses as the sole identifying information behind a network connection. EAP-TLS solves this, as certificates contain a variety of informational attributes, which can all be tied to an IP Address.