eBook

TLS BEST PRACTICES

digicert®

digicert®

BETTER BUSINESS STARTS WITH BEST PRACTICES

The truth is, a simple lapsed certificate can become a time-consuming headache. Unfortunately, 60% of organizations have experienced a certificate-related outage that impacted critical business applications within the last year. Fortunately, this eBook gives you a detailed but simple framework for becoming compliant with the certificate management best practices—so you can avoid becoming a statistic.

CONTENTS

IDENTIFY

Get a baseline of all certificates issued

Locate where each certificate is installed

Name owners of all certificates and domains

Identify web server O/S and application versions

Pinpoint web server cipher suites and SSL versions

REMEDIATE

Remove weak keys, cipher suites and hashes

Control wildcard certificate issuance and distribution

Deploy appropriate certificate types

Control all default vendor certificates

Ensure all web services have the latest patches installed

PROTECT

Standardize and automate issuance and renewal process

Install and renew all certificates in a timely manner

Ensure that private keys are not reused when certificates are renewed

Install certificates and private keys in a secure manner

Address certificate removal/revocation during decommissioning process

MONITOR

Scan networks for new systems and changes

Check Certificate Transparency (CT) logs for rogue certificates

Use CAA to prevent unauthorized certificate requests

GET A BASELINE OF ALL CERTIFICATES ISSUED

The best place to start is to get a baseline of all the assets in your certificate landscape. This includes identifying certificate owners, locations, domains, O/S and application versions, cipher suites and TLS versions.

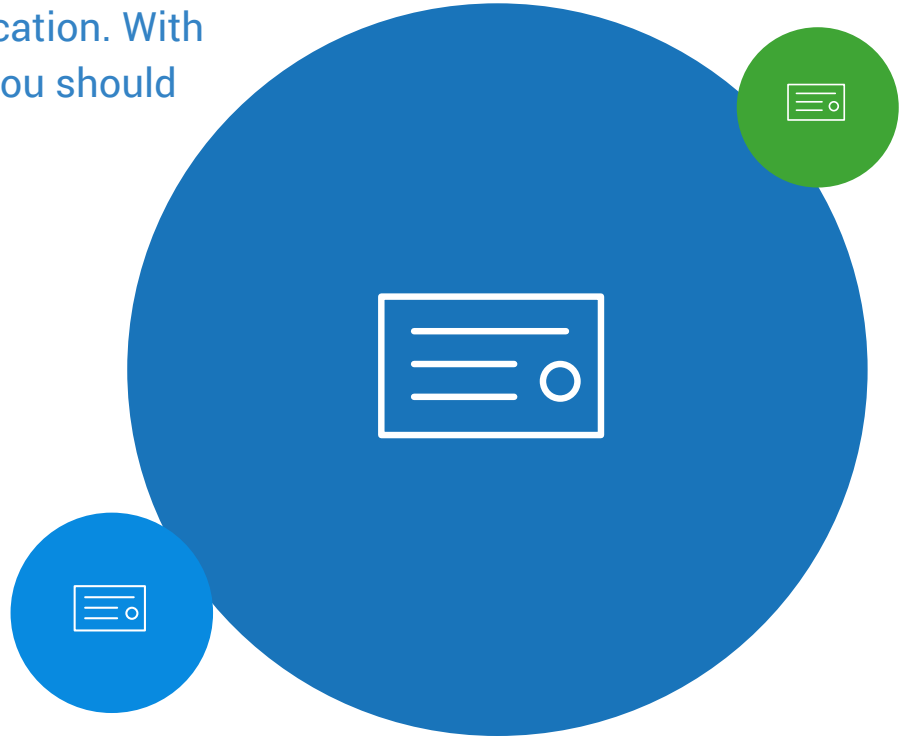
If you don't have a thorough inventory of your certificate landscape, you can open yourself up to security risks such as expiring certificates or weak keys and hashes. Your inventory should provide detailed certificate information that lists the type of certificate (DV, OV, EV, etc.) from all issuing CAs, as well as identifying problems with issuers, key lengths, algorithms, expiration dates and other certificate elements.

A good place to start is to get a list of issued certificates from your CAs. But how do you know you've captured everything? What about your internal CAs and any network devices with TLS certificates? The best method is to use a network scanner to detect TLS certificates. Many organizations are surprised by the large number of certificates they have online of which they weren't aware.

LOCATE WHERE EACH CERTIFICATE IS INSTALLED

Just because you've issued a certificate does not mean it has been installed correctly or in the right location. With your inventory of certificates as a baseline, you should add an inventory of verified server locations.

This can be manually tracked, but larger organizations prefer to use a periodic certificate discovery scan to verify that certificates are installed in their intended locations. If a rogue certificate is installed, it can allow encrypted traffic to leave the network without your knowledge.



NAME OWNERS OF ALL CERTIFICATES AND DOMAINS

Who else is buying certificates?
Are they renewing them?
Where are your gaps?

If a certificate owner leaves your organization, the certificates they owned might expire, resulting in costly outages. That's why it is vital to designate the owner of each certificate and establish a process for renewals and transfer of ownership. Domain ownership must also be verified in order for a CA to issue certificates to a public domain.

IDENTIFY WEB SERVER OPERATING SYSTEMS AND APPLICATION VERSIONS

Your inventory information should also include details of the operating system such as Windows or Linux, and applications such as Apache.

This is important because your organization could be vulnerable to exploits which attack specific versions of things like OpenSSL (i.e., Heartbleed).

PINPOINT WEB SERVER CIPHER SUITES AND SSL VERSIONS

Finally, your inventory should include web server cipher suites and SSL versions.

These items are typically configured on your web servers. Many SSL-specific attacks focus on older versions of SSL (e.g., the POODLE attack on SSL 3.0) or insecure cipher suites (e.g., the ROBOT attack on RSA encryption).

Cipher suite: A set of algorithms configured on a web server that helps to secure SSL or TLS network connections.

CONCLUSION

Your TLS inventory should include:

Certificates issued
- Certificate type
- Key size
- Algorithm
- Expiration date

Certificate locations

Certificate owners

Web server configuration
- O/S version
- Application version
- TLS version
- Cipher suites

REMOVE WEAK KEYS, CIPHER SUITES AND HASHES

Your inventory likely turned up some unresolved issues. Now, you can begin remediating those issues.

Certificates contain public keys and signatures which could be vulnerable to attacks. Certificates with key lengths less than 2048 bits or that use older hashing algorithms like MD5 or SHA-1 are no longer permitted on public web servers. However, you might find these on your internal websites. If so, it's vital that you upgrade them.

Even more important than identifying certificates with weak keys or hashes is reviewing of TLS/SSL versions and cipher suites supported on your web servers.

The following versions are outdated and vulnerable, and must be disabled:

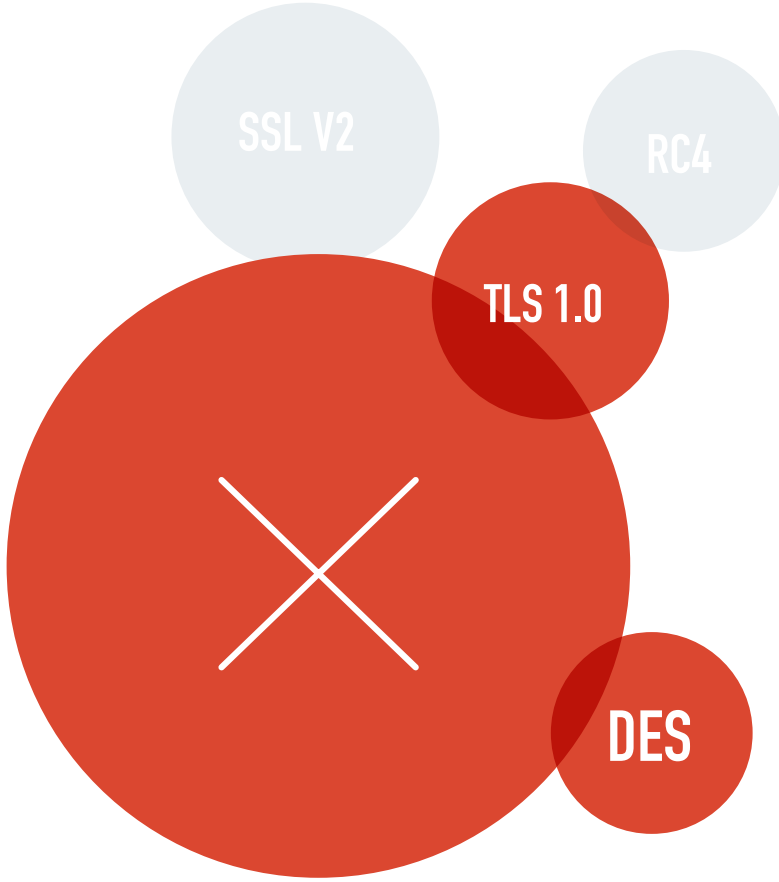
- SSL v2
- SSL v3
- TLS 1.0
- TLS 1.1

Instead, enable TLS 1.2 and TLS 1.3.

The following cipher suites are outdated and vulnerable, and must be disabled:

- DES
- 3DES
- RC4

Instead, use modern ciphers like AES.



CONTROL WILDCARD CERTIFICATE ISSUANCE AND DISTRIBUTION

Wildcard certificates offer the great advantage of one host name matching multiple host names—provided they satisfy the conditions.

That said, there are some concerns to be aware of with wildcard certificates. For one, if the private key of a wildcard certificate is stolen, attackers can then impersonate any system within that domain space. For example, stolen wildcard keys have been used for DNS poisoning or creating a rogue wireless access point within your network.

Another concern is that, if the wildcard is compromised, then you have to revoke and reissue all copies of the certificate at all locations where it has been installed. The more copies you have, the greater the headache. Unless well documented, you may not be certain that all copies have been replaced.

The best approach to avoid this is to issue each copy of a wildcard certificate with a different private key. That way, if one wildcard is compromised, you don't have to revoke all the copies.

Due to the above concerns, wildcard certificates are not permitted in Extended Validation (EV). But, assuming you have a well-controlled and documented process, there is nothing wrong with using wildcard certificates.

DEPLOY APPROPRIATE CERTIFICATE TYPES

Not all TLS certificates are created equal.

While private TLS certificates can be used for internal systems, the private root must be successfully propagated to users. If you are securing a public site, we recommend either an OV or EV certificate. DV certificates are never recommended for sites transacting sensitive information.

Extended Validation (EV)
- Rights to domain
- Thorough vetting of organization
- Highest assurance

Organization Validation (OV)
- Rights to domain
- Valid business registration
- High assurance

Domain Validation (DV)
- Rights to domain
- Low assurance

Private SSL certificates
- Must propagate private root to users

Self-signed SSL certificates
- Not trusted

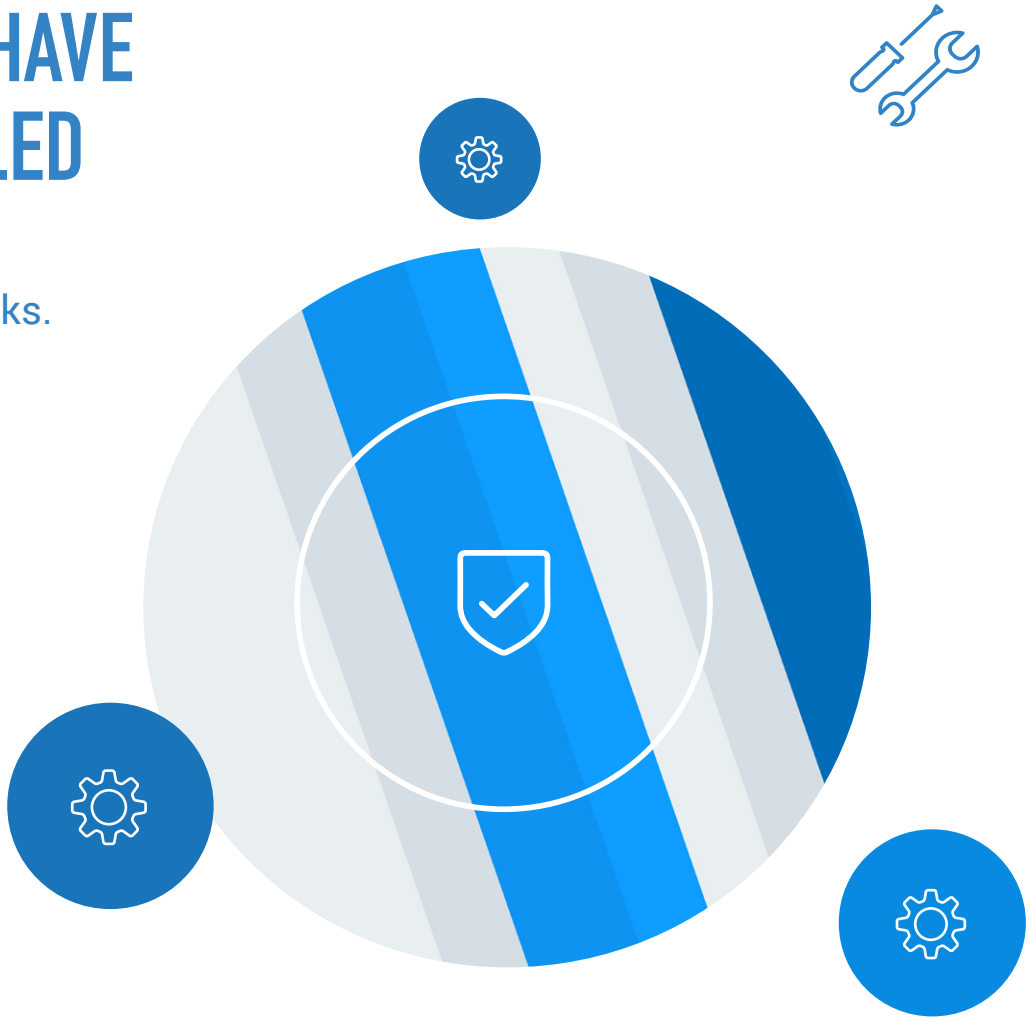
CONTROL ALL DEFAULT VENDOR CERTIFICATES

Vendor certificates are designed for ease of use, but not necessarily security.

The problem is that these vendors never intended these types of certificates to be put on a production network. Vendor certificates are typically self-signed, expired or using weak keys, and are therefore not trusted by browsers. Many organizations have thousands of vendor certificates they're not aware of. Each of these certificates should be removed and replaced by a certificate with known trust (at a minimum a private SSL certificate). To streamline this process, use the latest automation tools—including ACME protocol—to help you with the replacement and installation.

ENSURE ALL WEB SERVICES HAVE THE LATEST PATCHES INSTALLED

Patching operating systems is important to avoid some of the web's most devastating attacks.



This applies to your web servers, as well as your operating system.

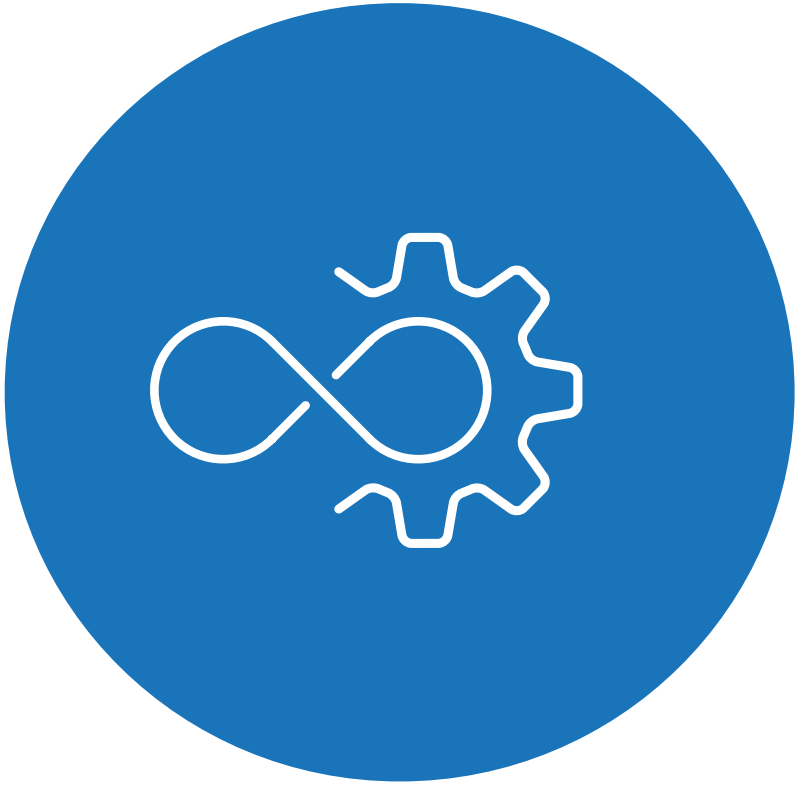
Conclusion

- Remove weak keys and hashes where possible
- Disable SSL v2, v3; TLS 1.0, 1.1
- Enable TLS 1.2, 1.3
- Disable weak cipher suites from TLS 1.2
- Control wildcard certificates
- Ensure appropriate certificate types are deployed
- Replace vendor certificates
- Ensure web servers have latest patches

STANDARDIZE AND AUTOMATE ISSUANCE AND RENEWAL PROCESS

Now that you've identified and remediated any issues, you can begin to put policies and procedures in place to mitigate future risks.

Creating a standardized process for certificate issuance and renewal will help you separate duties, prevent user errors and introduce automation to your SSL processes. For example, you can automate ACME protocol deployment in DigiCert® CertCentral using virtually any client and server type.

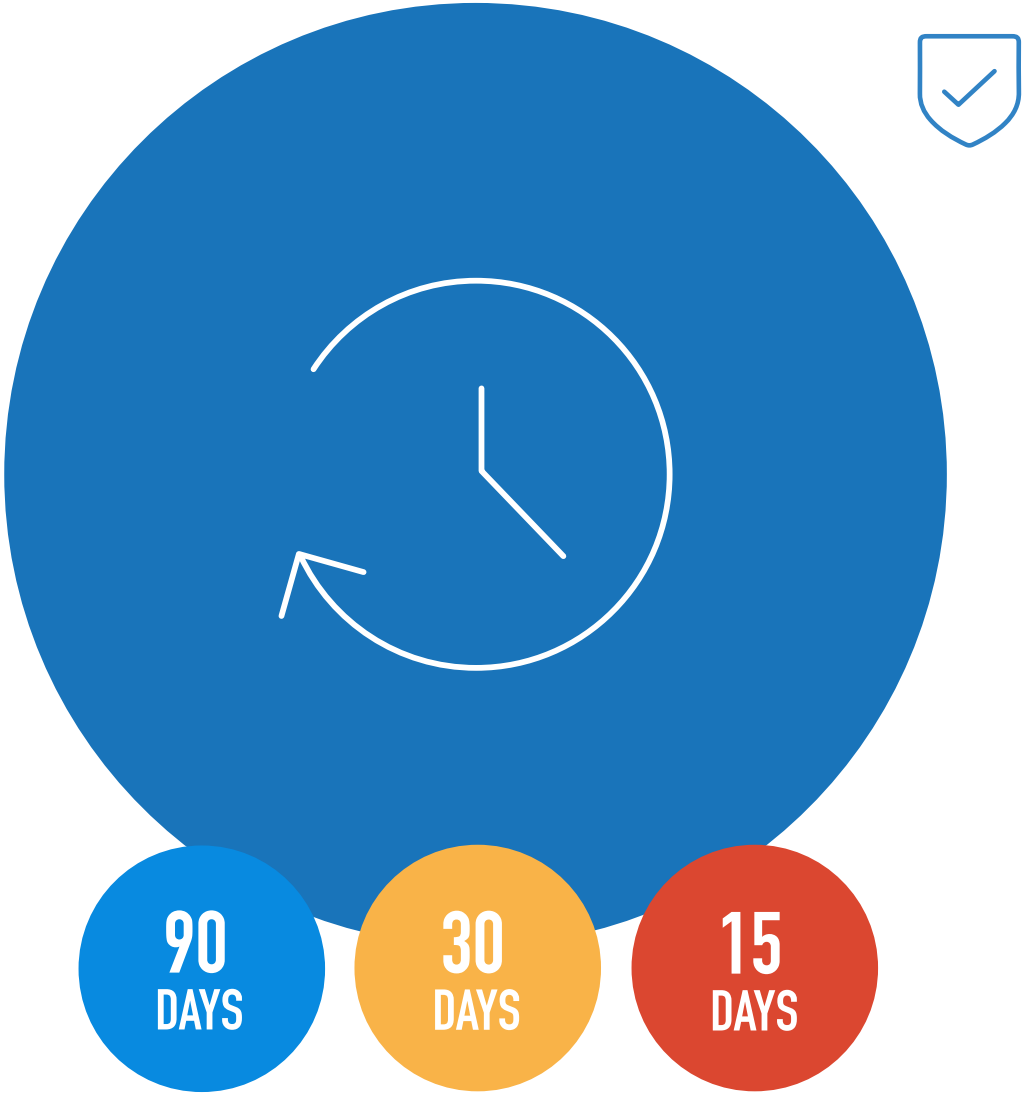


INSTALL AND RENEW ALL CERTIFICATES IN A TIMELY MANNER

Depending on your organization, you will be working under different time constraints.

We recommend renewing a certificate at least 15 days prior to the expiration date to ensure you have time for testing and rolling back to the previous certificate in the case of any issues. If you have a longer change control process, 30 days may be a more appropriate standard.

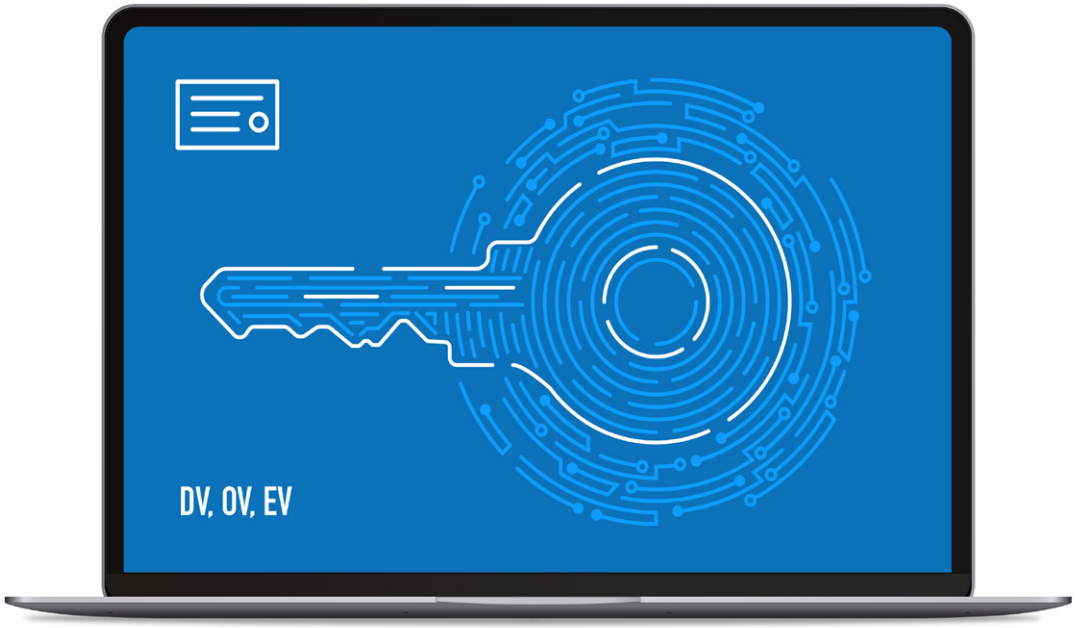
Whatever system you use should send warning notifications to users about expiring certificates. The system should notify users automatically and at regular intervals prior to expiration (e.g., 90 days, 60 days, 30 days, 15 days, etc.).



ENSURE PRIVATE KEYS AREN'T REUSED WHEN CERTIFICATES ARE RENEWED

Reusing private keys increases the risk of those keys being compromised.

As a best practice, make sure you create a new key-pair. Likewise, you should never reuse a CSR as this will automatically reuse the private key.



INSTALL CERTIFICATES AND PRIVATE KEYS SECURELY

Many organizations fail to create and store private keys in a secure manner.

- Create your private keys on a secure and trusted computer
- Give access to private keys only when absolutely necessary
- Generate a new private key whenever the owner leaves your company
- Use encrypted emails to distribute certificates and private keys
- Ensure you email system can delete and dispose of emails automatically
- Require two-factor authentication to access these systems
- Have a documented process for when the private key is exported or moved

ADDRESS CERTIFICATE REMOVAL/REVOCATION DURING DECOMMISSIONING

As part of the change control and decommissioning process for systems which reach end-of-life.

Conclusion

- Standardize and automate the issuance and renewal process
- Renew and install certificates in a timely manner
- Install certificates and private keys in a secure manner
- Never reuse private keys
- Address certificate removal/revocation in the decommissioning process

SCAN NETWORKS FOR NEW SYSTEMS AND CHANGES

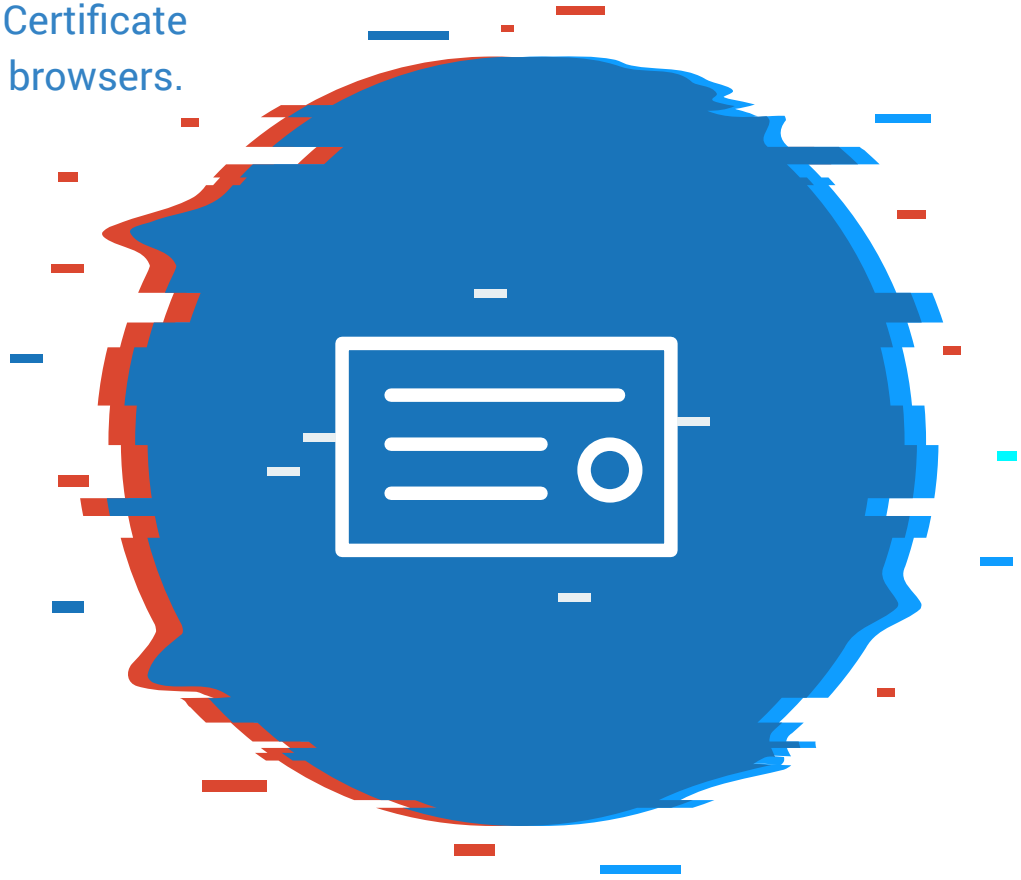
Instead of manually managing your TLS certificates, simply conduct regular checks for any risks that may crop up.

All networks are dynamic and constantly changing. That's why you need to constantly monitor for new systems or changes. This is best achieved using network scanning tools. These tools should highlight SSL security issues, certificate expirations and other network changes.

CHECK CT LOGS FOR ROGUE CERTIFICATES

Any public certificate not logged in a public Certificate Transparency (CT) log will not be trusted by browsers.

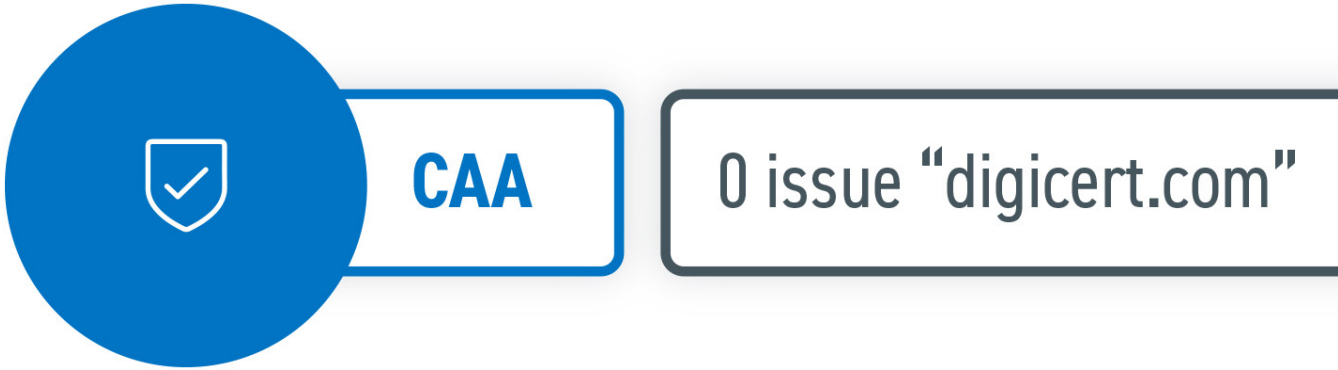
You can use a CT monitor to detect rogue certificates—much like a credit report—to quickly identify and remediate rogue certificates.



USE CAA TO PREVENT UNAUTHORIZED CERTIFICATE REQUESTS

Certificate Authority Authorization (CAA) is a DNS record used to specify which CAs are allowed to issue certificates for your domain.

In 2017, the CA/Browser Forum introduced Ballot 187 which requires all CAs to check the CAA DNS records and comply with any entries found for the domain in question. The purpose of this is to allow domain owners to declare which CAs are allowed to issue a certificate for their domain. CAA also provides a way to receive notifications in case someone requests a certificate from an unauthorized CA.



CONCLUSION

Now that you know what to do, it's time for the simplest and fastest way to get it done.

With **DigiCert® CertCentral**, you'll have all the capabilities you need to identify, remediate, protect, monitor—and, even better, customize and automate—your entire certificate ecosystem.

- Scan networks for new systems and changes
- Monitor CT logs for unauthorized certificates
- Use CAA to detect and prevent unauthorized certificate requests

CERTIFICATE MANAGEMENT TOOLS ARE COMMON. DOING EVERYTHING ON ONE PLATFORM ISN'T.

To learn how DigiCert® CertCentral makes it easy to implement best practice, visit **digicert.com/certificate-management**