**Remediation for NERAC Regions**

**Issue #1**

Insecure SSL/TLS versions available (#1 most common vulnerability across 44 municipalities [147 Occurrences] ) [Web Security]

**Why is it an issue?**

Susceptible to man-in-the-middle attacks

**How to Resolve?**

Identify your web server software (Apache, Nginx, IIS, etc.)

Back up your current server configuration.

Modify your server's SSL/TLS configuration to disable SSL and any versions of TLS older than 1.2.

Ensure that your server supports and is configured to prioritize newer, secure protocols like TLS 1.2 and TLS 1.3.

**Issue #2**

CAA not enabled. (#2 most common vulnerability across 44 municipalities [101 Occurrences] )

[Web Security]

**Why is it an issue?**

The domain does not contain a valid Certification Authority Authorization (CAA) record. A CAA record indicates which Certificate Authorities (CAs) are authorized to issue certificates for a domain.

**How to Resolve?**

1. Identify your DNS provider: Determine which DNS provider you are using for your domain. This information is typically provided by the company where you registered your domain or where you manage your DNS settings.
2. Access your DNS management interface: Log in to your DNS provider's website or access the DNS management interface provided by your domain registrar.
3. Locate the CAA record settings: Look for an option to manage DNS records or specifically CAA records. The exact location and terminology may vary depending on your DNS provider.
4. Add a CAA record: Create a new CAA record for your domain. You will need to provide the following information:
   * Name/Host: Enter your domain name or the subdomain for which you want to add the CAA record (e.g., "danversma.gov").
   * Flag: Set the flag to 0 (zero), which indicates a critical CAA record.
   * Tag: Set the tag to "issue".
   * Value: Specify the Certification Authority (CA) that you authorize to issue certificates for your domain. You can list multiple CAs by separating them with semicolons (;). For example, if you trust Let's Encrypt and Sectigo, you can set the value to "letsencrypt.org;sectigo.com".
5. Save the changes: Once you have entered the necessary information, save the CAA record.
6. Verify the CAA record: Wait for the DNS changes to propagate, which can take some time (usually up to 24 hours). You can use online DNS lookup tools to verify that the CAA record is correctly set up for your domain.
7. Repeat the process for other domains: If you have multiple domains triggering the CAA risk, follow the same steps for each domain.

**Issue #3**

Only weak cipher suites supported in TLS 1.2 (Provisional) (#3 most common vulnerability across 44 municipalities [95 Occurrences] )

[Website Security]

**Why is it an issue?**

Weak cipher suites can potentially be broken by a well-resourced attacker, and should not be used where possible. This server also does not appear to support any secure cipher suites, which means clients will either not be able to make a secure connection, or only make a weak one.

**How to Resolve?**

1. Identify the web server software: Determine which web server software you are using, such as Apache HTTP Server, Nginx, Microsoft IIS, etc.
2. Update the web server software: Ensure that your web server software is up to date with the latest version, as newer versions often include security enhancements and stronger cipher suites.
3. Disable weak cipher suites: In your web server's configuration file, disable the weak cipher suites and only enable secure and strong cipher suites. Remove or comment out any cipher suites that are considered weak or vulnerable.

Enable Perfect Forward Secrecy (PFS): Perfect Forward Secrecy ensures that even if an attacker gains access to the server's private key, they cannot decrypt past communications. Enable PFS by configuring your web server to use cipher suites that support it.

1. Enable TLS 1.3: If your web server and client support it, consider enabling TLS 1.3, which is the latest version of the TLS protocol and provides enhanced security and performance benefits.
2. Obtain and install a valid SSL/TLS certificate: Ensure that you have a valid SSL/TLS certificate installed on your web server. Use a trusted certificate authority to obtain the certificate to establish secure connections with clients.
3. Test and verify the configuration: After making the necessary changes, thoroughly test your web server's configuration using online SSL/TLS testing tools or tools like Qualys SSL Labs. These tools can help identify any remaining vulnerabilities or misconfigurations.

**Issue #4**

HTTP Strict Transport Security (HSTS) not enforced (#4 most common vulnerability across 44 municipalities [84 Occurrences)

[Web Security]

**Why is it an issue?**

Without HSTS enforced, people browsing this site are more susceptible to man-in-the-middle attacks. The server should be configured to support HSTS.

**How to Resolve?**

1. Verify SSL/TLS Support: Ensure that your website is already using an SSL/TLS certificate and serving content over HTTPS.
2. Set HSTS Header: Add the HSTS header to all HTTP responses from your website. Include the directive "Strict-Transport-Security" with the appropriate settings, such as the maximum duration and any additional options like "include Subdomains" or "preload".
3. Test Header Configuration: Use browser developer tools or online tools to verify that the HSTS header is being sent correctly and configured properly.
4. Submit to Preload List: Consider submitting your domain to the HSTS preload list maintained by major browsers for enhanced enforcement. Follow the guidelines provided by hstspreload.org.
5. Cache Invalidation (Optional): If necessary, create a new subdomain with a short-duration HSTS header to prompt browsers to refresh their HSTS cache and enforce HTTPS for your main domain.

**Issue #5**

X-Content-Type-Options is not nosniff (#5 most common vulnerability across 44 municipalities [77 Occurrences] )

[ Web Security]

**Why is it an issue?**

Browsers may interpret files as a different MIME type than what is specified in the Content-Type HTTP header. This can lead to MIME confusion attacks.

**How to Resolve?**

1. Identify the Web Server: Determine the web server software (e.g., Apache, Nginx, IIS) being used to serve your website. The configuration steps may vary depending on the server.
2. Enable Content-Type Options: Access the configuration file for your web server (e.g., .htaccess for Apache) and add the following line to enable the "nosniff" option:

Header set X-Content-Type-Options "nosniff"

This setting instructs the browser to enforce strict MIME type checking and prevents it from guessing the content type based on the response.

1. Test Configuration: Restart your web server and verify that the "X-Content-Type-Options" header with the value "nosniff" is being sent with your HTTP responses. You can use browser developer tools or online security scanning tools to check the headers.
2. Content Validation: Ensure that your website's server-side scripts and file uploads correctly set the appropriate MIME types. This step is crucial to prevent any conflicts between the declared MIME type and the actual content.
3. Periodic Review: Regularly review your website's configuration and MIME type handling to ensure ongoing compliance. Keep your web server software up to date to benefit from the latest security enhancements.