TITANIUM

martes, 15 de junio de 2021

Vulnerabilidad zero day en Primion-Digitek EVALOS Secure 8

Durante un ejercicio de hacking ético industrial a uno de nuestros clientes, detectamos una vulnerabilidad zero day crítica en una aplicación de control de accesos y personal de planta. En este caso se trata de la aplicación Evalos8 (Primion-Digitek), en concreto en la versión v1.0.1.55 del módulo Secure8. No hemos podido acceder a otras versiones anteriores de este producto para verificar si se reproduce la vulnerabilidad, por lo que estas podrían verse igualmente afectadas.

Primion-Digitek se especializa en implantación de controles de acceso, fichaje, control de personal, etc. Tienen implantados sus sistemas tanto en infraestructuras críticas, como pueden ser aeropuertos, industrias eléctricas, hospitales, así como en edificios de administración pública, entre otros.

En este caso, hemos descubierto una vulnerabilidad de tipo Blind SQL Injection en el módulo de autenticación contra el gestor administrativo de la aplicación. Mediante esta vulnerabilidad, es posible la extracción de usuarios y hashes SHA-1 de las contraseñas. A la vez, sería posible la extracción de cualquier otro dato almacenado en la base de datos.

A esta vulnerabilidad de Blind SQL Injection se le ha asignado el código CVE-2021-3604

Todo comenzó cuando empezamos a evaluar posibles vectores de ataque que bien nos permitieran realizar un bypass de la autenticación (SQL Injection), nos ofrecieran un resultado booleano diferente dependiendo del payload introducido (Blind SQL Injection), u ofreciesen un retraso de tiempo dependiendo del payload introducido (Time Based SQL Injection).

En este caso, conseguimos resultados diferentes con los siguientes payloads, indicando que el parámetro introducido estaba siendo inyectado y ejecutado en una consulta SQL:

- admin‘ or ‘a’=’a
- admin’ or ‘a’=’b

Una vez que verificamos la posibilidad de evaluar consultas booleanas, es posible la obtención de datos mediante la evaluación de una consulta de tipo ‘prueba y error’:

- ‘ or (<campo> LIKE ‘xxx%’)#

Dado que los campos son desconocidos, podemos extraer la información que necesitamos consultando la información contenida en las tablas INFORMATION_SCHEMA.TABLES y INFORMATION_SCHEMA.COLUMNS de SQLServer. De esta manera se han conseguido los nombres de columna ‘usuario’ y ‘ClaveAcceso’ de la tabla de usuarios referenciada en la consulta de autenticación.

Una vez tenemos estos datos en nuestro poder, es posible la extracción de la información de usuarios. Para ello desarrollamos un exploit que comprueba todos los caracteres del diccionario hasta obtener un usuario válido.

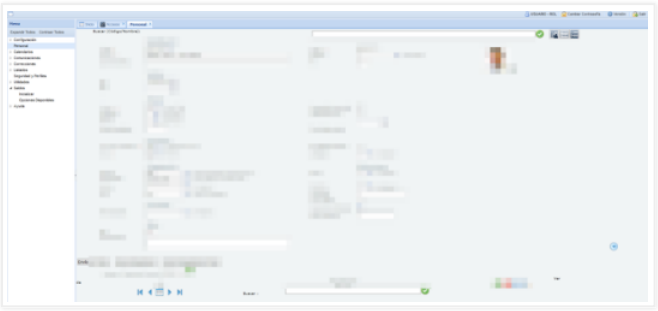
A continuación, se repite este proceso, pero con la columna 'ClaveAcceso'.

Los datos obtenidos como password son una serie numérica de 120 dígitos de longitud, y la frecuencia de los números 0, 1 y 2 es mayor al resto. De aquí podemos deducir que puede tratarse de una serie de números menores de 255, y concatenados en grupos de tres. Tras comprobar que, tomando los números en tríos de dígitos, ninguno es mayor a 255, probamos a pasarlos a hexadecimal, obtenido un posible hash de 40 caracteres. Inmediatamente, nos viene a la cabeza el tipo de hash SHA-1, el cual tiene una longitud de 40 caracteres.

Finalmente probamos a crackear el hash, obteniendo la contraseña en claro en menos de 1 segundo, ya que tenía una longitud de 1 carácter.

A partir de aquí, es trivial la construcción de un exploit válido que extraiga todos los usuarios y hashes del sistema.

Una vez dentro del sistema, es posible la edición los permisos acceso, pudiendo generar nuevos usuarios, revocar la entrada a otros, control de zonas críticas de la planta, etc. De esta manera, la seguridad física de una infraestructura crítica quedaría comprometida por completo, permitiendo el acceso a la infraestructura y a las zonas más críticas a cualquier usuario que tomase el control de esta aplicación.



Podemos obtener dos conclusiones importantes de este caso:

La primera es que es muy importante no conformarse únicamente en el análisis de vulnerabilidades conocidas durante una fase de pentest, ni con el uso de herramientas automáticas. Existen componentes que, al no ser usados de una manera más extendida, pueden tener vulnerabilidades fácilmente explotables, ya que, por desgracia, muchos desarrollos no han contemplado un desarrollo seguro de la aplicación.

La otra es que, desde una simple vulnerabilidad en un componente de software, se puede poner en grave riesgo una infraestructura crítica al completo, comprometiendo la seguridad física de la misma, por lo que es fundamental identificar estos componentes críticos en las aplicaciones a desplegar en dichas infraestructuras e, idealmente, evaluar su seguridad antes de su despliegue, con empresas como Titanium Industrial Security, que disponen de los medios humanos y materiales para hacer esto.

Enlace a la publicación de INCIBE:

https://www.incibe-cert.es/alerta-temprana/avisos-sci/vulnerabilidad-inyeccion-sql-primion-digitek-secure-8

Para finalizar, agradecer a INCIBE por la ayuda prestada en la notificación y gestión de la vulnerabilidad y felicitarlos por su reciente nombramiento como Root CNA.

Publicado por Titanium Industrial Security en 4:43

No hay comentarios:

Publicar un comentario

Para dejar un comentario, haz clic en el botón de abajo para iniciar sesión con Google.

Inicio Entrada antigua

Suscribirse a: Enviar comentarios (Atom)