Assidian Protocol

Ivan Piccaro

ivan@ivorcorp.com.ar

www.ivorcorp.com.ar

Resumen. Un protocolo encargado de administrar y automatizar todos los sistemas que no requieran intervención humana, que requieren una intervención mínima o que sean repetitivas. Este protocolo incluye 2 sistemas, uno basado en A.I. y el otro basado en inteligencia interna.

1. Introducción

Al buscar dedicar la mayor cantidad de tiempo a crear nuevos proyectos y mejorar los ya existentes hay que optimizar actividades repetitivas o con patrones similares. Para esto creamos *Assidian protocol*, el cual se basa en un sistema de doble entrada, combinando un A.I. de aprendizaje automático y un sistema basado en algoritmos, ambos con el mismo objetivo, la combinación de estas 2 mecánicas tiene de objetivo buscar el menor índice de error.
Un efecto práctico fue el paso de websites *all in one* a un website basado en *frontend*, con su *backend* en otro servidor y contando con el tercer servidor de información, esto para bifurcar las cargas hacia los servidores internos, el programador exporta una web, y ambos sistemas tratan de hacer lo mismo con otra web, ya que todas se basan en la misma estructura de archivos, de esta forma lo que a un programador le hubiera tomado horas en solo 10 minutos entrena al protocolo para poder hacerlo repetidas veces.
Este es solo un uno, el objetivo es requerir la menor participación humana en este tipo de actividades, como configurar nuevos servidores, instalar O.S, Exportar D.B, Incluso mitigar ataques de todo tipo, ya que este protocolo cuenta con control total sobre todas las cuentas administrativas de la corporación, desde bloquear una región en el administrador DNS hasta acceso *ROOT a todos los servidores*, *al tener acceso total a nuestros sistemas puede enlazarse a otros proyectos y aplicar actualizaciones automáticas a los mismos*, *basándose en actualizaciones de otros*, *principalmente con el objetivo de mejorar su seguridad y velocidad de ejecución*.

2. Exportaciones

El protocolo importa un proyecto en funcionamiento, creando una segunda capa de ejecución no visible, e implementa actualizaciones de forma recursiva a todas sus librerías internas, optimizando el código, sumando nuevas funcionalidades y creando nuevos enlaces con otros proyectos internos, esto vía los *servidores de enlace* los cuales desencriptan los archivos en el menor tiempo posible y son corromperlos, porque estos cuentan con una seguridad muy estricta que limita su lectura; luego de aplicar las actualizaciones el protocolo verifica el funcionamiento del proyecto y en caso de no demostrar ningún error en su funcionamiento el mismo se exporta hacia un servidor con el fin de ejecutarse en la capa principal luego de una simple verificación del operador, esto para verificar que no queden brechas de seguridad en la actualización. En caso de una falla existente en el código el sistema busca el archivo causante del mismo y restaura su versión anterior, en caso de funcionar ejecuta la actualización en el mismo de otra manera buscando replicar el mismo error, en caso de no funcionar se borra toda la importación hacia los servidores y sigue con otro proyecto. El protocolo tiene 2 sistemas internos, ambos funcionando de maneras distintas, pero con un mismo objetivo, de esta forma podemos aplicar un mismo cambio utilizando 2 métodos, esto permite aplicar cambios simultáneos a un mismo proyecto, pero en distintos sectores del mismo, para optimizar tiempos y no utilizar un mismo método en todos los archivos, permitiendo funcionar de distintas maneras, pero con un resultado igual en ambos casos.

3. Servidor de enlace

El protocolo utiliza una clave única basada en huellas digitales de los mismos y un algoritmo privado que enlaza internamente 2 hashes, esto para cifrar todas las conexiones y traslados de archivos entre nuestros servidores, utilizando AES-256 como método principal de encriptación y desencriptación, aunque se utilizan otros para respaldar la integridad de la información movilizada entre los servidores; la información no se envía directamente de un servidor a otro sino que se envía a través de varios servidores buscando descentralizar la información movilizada, buscando un paso a Web3 al corto plazo, con el fin de mejorar la seguridad e integridad de la información tanto privada como pública de los proyectos internos de la corporación, los creados en colaboración con otros organismos e incluso los pertenecientes a clientes.

Una vez creado el enlace entre los servidores inicia una cuenta regresiva, y cada 10 minutos toda la información se borra de todos los servidores menos el raíz y 2 de respaldo, utilizando uno de estos 3 para reestablecer la conexión creándose un nuevo vinculo bajo otro

algoritmo y otra clave; de esta forma es mucho más complejo intervenir en una de estas conexiones o llegar a obtener uno de estos archivos, porque aunque logres obtenerlo necesitaras las claves de acceso del mismo, ya que además de tener la clave del ultimo enlace tiene otras claves internas basadas en el hash del archivo original, esto permite que en caso de alterar el archivo este se corrompa evitando cualquier lectura no autorizada a la información que se encontraba dentro del mismo, al tener control total sobre todo sistema y toda cuenta de la corporación priorizamos mucho la seguridad interna.

4. Red interna

Para mantener una red privada con la mayor seguridad posible utilizamos *Servidores de enlace* aislados, aunque sincronizados utilizando una VPN para que solo sea accesible desde un único nodo, aunque si este se cae automáticamente se detectara y otro *Servidor de enlace* tomara su puerta de enlace como nodo principal, estos servidores utilizan un firewall el cual solo permite el acceso desde la VPN y al puerto del sistema, la única forma de acceder a la terminal es de forma física, y solo existe un usuario accesible, el cual solo tiene acceso a una *Restricted Shell* y el usuario *root* tiene una clave generada aleatoriamente, la cual se cambia cada 5 minutos; eliminando por completo la opción de acceder al usuario *root*, no hay acceso al servidor de ninguna forma, salvo vía la actualización del sistema la cual solo se puede hacer desde el mismo sistema y bajo su supervisión autónoma, lo cual deshabilita toda *update* que intente crear una petición hacia afuera, verificando la actualización de forma interna en una VM completamente aislada; esta se deja corriendo por un tiempo indefinido, por lo cual se decide dependiendo las líneas de código sumadas a la *update*, y solo en caso de no encontrar ningún error, petición hacia afuera, intento de acceso *root*, carga de archivos, u otra acción perjudicial para la extrema seguridad del sistema se actualizara. Además, cuenta con un enlace hacia *cleallpath protocol* el cual en caso de detectar alteraciones simultaneas en servidores o un apagón general de servidor, actuara deshabilitando todo el sistema y eliminando todas sus atribuciones al servidor.

5. Conclusión

El proyecto cuenta con ideales y sistemas de seguridad muy superiores para verificar hasta el menor detalle existente, esto porque cuenta con control total sobre los servidores de actividad, aunque corren sobre los *servidores de enlace*, los cuales no solo son accesibles desde el puerto del sistema y no cuentan con otro método de acceso, ni siquiera desde consola local, porque el usuario *root* cuenta con una clave que se modifica cada 5 minutos. Sobre sus funcionalidades, este sistema actualiza y mejora todos los proyectos existentes en nuestros servidores siguiendo patrones de programación humana o utilizando A.I.