Revista Cubana de Ciencias Informáticas
Vol. 15, No. Especial UCIENCIA II, Septiembre, 2021
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 42-54

Tipo de artículo: Artículo original
Temática: Seguridad informática
Recibido: 30/06/2021| Aceptado: 01/10/2021

Mecanismo de bloqueo a conexiones remotas que intentan accesos por fuerza bruta para nova-ltsp

Blocking mechanism to remote connections attempting brute forceaccess for nova-ltsp

Mayra de la O Barrientos [1*] https://orcid.org/0000-0002-0962-4720

[1] Centro de Soluciones Libre. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 1⁄2, Torrens, Boyeros, La Habana, C.P.: 19370, Cuba, mdelao@uci.cu

* Autor para la correspondencia.(mdelao@uci.cu)

RESUMEN

La Distribución Cubana GNU/Linux Nova, cuenta con la plataforma de administración de clientes ligeros Nova-LTSP, que brinda un conjunto de funcionalidades para administrar estos clientes ligeros. Actualmente la herramienta carece de mecanismos de bloqueo a conexiones remotas que intentan accesos por fuerza bruta, lo que trae como consecuencia vulnerabilidad de la información almacenada en la misma. Debido a esto, el objetivo de este trabajo consiste en desarrollar un módulo que garantice la implementación de mecanismos de bloqueo a conexiones remotas en la plataforma de clientes ligeros Nova-LTSP. La propuesta de solución es guiada por la metodología de desarrollo de software Variación del Proceso Unificado Ágil para la Universidad de las Ciencias Informáticas en su escenario de historias de usuario. Además, se realiza un análisis de las diferentes herramientas, tecnologías, y lenguajes a utilizar para el modelado y desarrollo del módulo. Se realizaron pruebas funcionales, de unidad, integración, aceptación y

Revista Cubana de Ciencias Informáticas
Vol. 15, No. Especial UCIENCIA II, Septiembre, 2021
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 42-54

de seguridad para comprobar el correcto funcionamiento y calidad del módulo para la implementación de mecanismos de bloqueo a conexiones remotas que intentan accesos por fuerza bruta.

Palabras clave: bloqueo; fuerza bruta; módulo; Nova-LTSP; seguridad.

ABSTRACT

The Cuban Distribution GNU / Linux Nova, has the Nova-LTSP thin client management platform, which provides a set of functionalities to manage these thin clients. Currently, the tool lacks mechanisms for blocking remote connections that attempt brute force access, which results in vulnerability of the information stored in it. Due to this,the objective of this work is to develop a module that guarantees the implementation of blocking mechanisms to remote connections in the Nova-LTSP thin client platform. The solution proposal is guided by the software development methodology Variation of the Agile Unified Process for the University of Computer Science in its user stories scenario. In addition, an analysis of the different tools, technologies, and languages to be used for the modeling and development of the module is carried out. Functional, unit, integration, acceptance and safety tests were carried out to verify the correct functioning and quality of the module for the implementation of blocking mechanisms to remote connections that attempt brute force access.

Keywords: blockade; brute force; module; Nova-LTSP; security.

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) han tenido un desarrollo acelerado durante las últimas décadas, en este contexto el empleo de sistemas informáticos es clave para cualquier esfera de la sociedad. En el ámbito empresarial su uso se ha incrementado considerablemente, debido en gran medida, al hecho de cada vez contar con un mayor volumen de información a gestionar. La importancia de la información administrada en estos sistemas ha despertado gran interés en toda una gama de intrusos

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

cibernéticos, que se han aprovechado de numerosos problemas de seguridad existentes en las aplicaciones, para llevar a cabo toda una serie de ataques como denegación de servicio, autenticación, entre otros. Es entonces vital disminuir las posibles vulnerabilidades que pudiesen existir en estos sistemas a través de la implementación de medidas y políticas de seguridad.

La seguridad constituye uno de los temas principales en la rama de la informática a nivel mundial. Antes los problemas de seguridad estaban dados por los virus, actualmente son otros los tipos de ataques que preocupan a los usuarios como son: los spamming [a], pharming[b], hacker[c], cracker[d]. Hay que tener en cuenta que la seguridad informática es un proceso dinámico, suele siempre estar encaminado a la actualización permanente de mecanismos, métodos, técnicas y procedimientos que ayudan a contrarrestar los ataques o amenazas informáticas que cada día aparecen en Internet (Tarazona, 2016).

La seguridad es uno de los mecanismos que necesita una mejora continua, constituye uno de los factores más vulnerables que tienen los sistemas informáticos, que es cada vez más difícil de controlar. Los mecanismos de seguridad son también llamadas herramientas de seguridad y son todos aquellos que permiten la protección de los bienes y servicios informáticos. Dichos mecanismos se clasifican en diferentes tipos de acuerdo con el objetivo principal de los mismos: preventivos, consisten en prevenir la ocurrencia de un ataque informático; detectores, tienen como objetivo detectar todo aquello que pueda ser una amenaza para los bienes; correctivos, se encargan de reparar los errores o daños causados una vez que se haya cometido un ataque, modifican el estado del sistema de modo que vuelva a su estado original, y los disuasivos, se encargan de desalentar a los perpetradores de que cometan su ataque para minimizar los daños que puedan tener los bienes.

Cuba, en aras de ganar en soberanía tecnológica y seguridad, así como garantizar la informatización de todas las esferas de la sociedad, inició su incursión en el año 2002 en el desarrollo del Proyecto Futuro, nombre dado por el Comandante en Jefe Fidel Castro Rúz a la Universidad de las Ciencias Informáticas (UCI). En la actualidad la Universidad cuenta con un total de 15 centros de desarrollo de software, dentro de ellos se encuentra el Centro de Software Libre (CESOL), cuyo objetivo es el desarrollo de la distribución cubana de GNU/Linux Nova.

La Distribución Cubana GNU/Linux Nova, cuenta con la plataforma de administración de clientes ligeros Nova-LTSP, que presenta módulos para la gestión de los clientes ligeros, imágenes de los sistemas operativos y perfiles de comportamiento de hardware y software, así como la administración del protocolo de configuración dinámica de host (DHCP, del inglés Dynamic Host Configuration Protocol) y del sistema. Constituye también una herramienta que permite automatizar la administración de los clientes ligeros, a través de la tecnología Linux Terminal Server Project (LTSP).

En una entrevista realizada a varios especialistas encargados del despliegue de la herramienta Nova-LTSP, se identificó qué, aunque esta herramienta cuenta con una interfaz de monitoreo para el seguimiento de recursos de hardware, la misma carece de un mecanismo de bloqueo a las conexiones remotas que intentan accesos por fuerza bruta, lo que trae como consecuencia la vulnerabilidad de la información y los servicios que se ejecutan en el servidor. Por tanto, se define como **objetivo general**: Desarrollar un módulo para Nova-LTSP que permita la gestión de mecanismos de bloqueo a las conexiones remotas que intentan accesos por fuerza bruta.

Métodos o Metodología Computacional

La investigación realizada y el análisis a la plataforma de clientes ligeros NOVA-LTSP arrojó como resultado la inexistencia de un mecanismo de bloqueo a conexiones remotas que intentan accesos por fuerza bruta, por lo que la plataforma se encuentra vulnerable a ataques informáticos. Tal resultado implicó el desarrollo de un módulo capaz de bloquear cualquier ataque realizado a dicha plataforma.

Para su desarrollo se emplearon un conjunto de lenguajes de programación tales como Python para la programación de las funcionalidades del lado del servidor, JavaScript para manipular las interfaces del lado del cliente y lograr dinamismo en la aplicación y CSS para aplicar estilos a las vistas que se presentan al usuario. Como Entorno de Desarrollo Integrado (IDE) se utilizó el Pycharm en su versión 4.5.4, el framework Django en su versión 2.1.3, encargado de la parte lógica del negocio de la aplicación y la librería Augeas para la edición de la configuración. Para completar aspectos relacionados con el tratamiento de

imágenes y modelado UML (Unified Modeling Language, en español Lenguaje Unificado de Modelado), la herramienta CASE (Computer Aided Software Engineering, en español Ingeniería de Software Asistida por Ordenador.) Visual Paradigm respectivamente.

La metodología de desarrollo aplicada es la variación de la metodología de Proceso Unificado Ágil (AUP, del inglés Agile Unified Process) para la UCI, una variante realizada por la Universidad de las Ciencias Informáticas a la metodología ágil AUP y definida por la universidad como guía para la actividad productiva que permite estandarizar los diferentes productos de trabajo que se generan en sus centros productivos y se adapta al ciclo de vida definido para los proyectos de la UCI (Rodríguez, 2014).

Para el desarrollo de la presente investigación se aplicaron los siguientes métodos científicos:
Histórico-Lógico se utilizó en el análisis de los sistemas homólogos, de manera que permita buscar elementos que los caractericen y aspectos para fundamentar la propuesta de solución a la problemática planteada. El **Analítico-sintético** con el fin de descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución propuesta. El **Inductivo-Deductivo,** durante la especificación de la arquitectura y el uso de patrones de diseño para resolver problemas particulares del sistema en desarrollo. **Entrevista** se le realizó una entrevista al MSc Yasiel Pérez Villazón perteneciente al Centro de Software Libre con el objetivo de conocer las vulnerabilidades presentes en Nova-LTSP. **Encuesta** se aplicó para evaluar la propuesta de solución desarrollada y conocer el índice de satisfacción de los usuarios potenciales. **Observación** se utilizó a través del estudio realizado a las herramientas informáticas que existen, que permiten implementar mecanismos de bloqueo a conexiones remotas que intentan accesos por fuerza bruta, para determinar los elementos más comunes que están presentes en las mismas.

Resultados y discusión

Estudio de homólogos

Para el estudio de homólogos se empleó el método Calificación y Selección de Código Abierto (QSOS, del inglés *Qualification and Selection of Open Source software*). La finalidad de este análisis tiene como objetivo conocer si las aplicaciones analizadas cumplen con las necesidades existentes para el bloqueo de conexiones remotas que intentan accesos por fuerza bruta en Nova-LTSP.

Aplicación del método QSOS

El método propone cuatro etapas: definición, evaluación, clasificación y selección. Se establece un método de calificación de software para cuantificar y medir las posibilidades reales de implantación del software ofreciendo posibilidad de comparación al establecer criterios ponderados, en base a los cuales calificar el software y hacer una selección final de la manera más objetiva y beneficiosa (RAMOS, 2011).

Etapa de Definición: se establece el marco de referencia para la búsqueda de la información relacionada con las necesidades existentes en el proyecto de software a desarrollar. El estudio realizado en la investigación sobre aplicaciones informáticas que implementan mecanismos de bloqueo a conexiones remotas para Nova-LTSP está enmarcado en herramientas basadas en GNU/Linux. Para esto se estudiaron las siguientes herramientas:

1. DenyHost: Es una herramienta de seguridad de prevención de intrusiones basada en registros para servidores con protocolos de administración segura (SSH, del inglés *Secure Shell*) escrita en Python. Está pensada para evitar ataquesde fuerza bruta en servidores SSH mediante el monitoreo de intentos de inicio de sesión no válidos en el registro de autenticación y el bloqueo de las direcciones IP de origen.
2. Snort: Es un sistema de detección y prevención de intrusos Open Source basado en red que utiliza patrones de búsqueda, llevando a cabo registros de paquetes, análisis de protocolo y búsqueda o comparación de contenido en tiempo real. Utiliza reglas descriptivas para determinar qué tráfico debe ser monitorizado y un motor de detección diseñado modularmente para identificar ataques en tiempo real.

3. Fail2ban: Es una herramienta (aplicación) de código abierto que permite bloquear a aquellos que intenten vulnerar servicios como SSH, protocolo para transferencia simple de correo (SMTP, del inglés *Simple Mail Transfer Protocol*), protocolo de transferencia de hipertextos (HTTP, del inglés *Hypertext Transfer Protocol*), etcétera, mediante el intento de fuerza bruta. Cuando Fail2ban encuentra reiterados intentos de sesión fallidos desde una misma IP rechaza estos intentos de conexión y los bloquea con reglas de Iptables, es decir, cuando alguien intenta conectarse muchas veces Fail2ban lo bloquea y lo pone en su tabla entonces no deja que vuelva a intentar conectarse.
4. Ossec: Ossec es un sistema HIDS, es decir, un sistema de detección de intrusos que también opera como un sistema de gestión de incidentes de seguridad (SIM, del inglés *Security Incident Managament*). Ossec permite a los clientes implementar un sistema integral de detección de intrusos basado en la supervisión del host con políticas específicas de aplicaciones en el lado servidor. Como software, funciona con la mayoría de los sistemas operativos, incluyendo Linux, OpenBSD, FreeBSD, Mac OS X, Windows.

Etapa de Evaluación: consiste en realizar una caracterización del software analizado. En el epígrafe anterior se describen las herramientas informáticas estudiadas.

Etapa de Calificación: consiste en la ponderación de los criterios definidos para realizar la comparación de las herramientas analizadas. En la Tabla 1 se describen los criterios establecidos, en correspondencia con las necesidades del cliente y las características del entorno de desarrollo del proyecto Nova-LTSP.

Tabla 1 - Definición de criterios y su ponderación. Elaboración propia.

Criterios de análisis	**Puntuación**	
	No cubierto 0	Totalmente cubierto 1
Soporte para servicio	No presenta soporte para servicios	Presenta soporte para servicios
Basado en reglas	No está basado en reglas	Está basado en reglas

Basado en red	No está basado en red	Está basado en red
Bloqueo de ataque en tiempo real	No bloquea los ataques en tiempo real	Bloquea los ataques en tiempo real
Mecanismo de notificación	No presenta mecanismos de notificación	Presenta mecanismos de notificación

Etapa de Selección: se realiza la comparación de diferentes herramientas analizadas mediante los criterios definidos en la fase de calificación. En la Tabla 2 se evidencia la comparación de las herramientas informáticas que implementan mecanismos de bloqueo a conexiones remotas.

Tabla 2 - Tabla comparativa de las herramientas informáticas que implementan mecanismos de bloqueo a conexiones remotas. Elaboración propia.

Criterios de análisis	**Herramientas informáticas que implementan mecanismos de bloqueo a conexiones remotas**			
	DenyHost	Fail2ban	Snort	Ossec
Soporte para servicio	1	1	1	1
Basado en reglas	0	1	0	0
Basado en red	1	1	1	1
Bloqueo de ataque en tiempo real	0	1	1	0
Mecanismo de notificación	1	1	0	0

Al realizar la comparación mediante QSOS de las tecnologías homólogas existentes que contribuyen a la seguridad de los datos informáticos, se define como solución a emplear Fail2ban.

Fail2ban es una aplicación de Linux que permite evitar accesos no autorizados al servidor. Funciona bloqueando, o baneando, las IP que realicen varios intentos de acceso incorrectos al servidor.

Fail2ban detecta aquellas direcciones IP cuyo comportamiento resulta inusual, por ejemplo, las que han intentado acceder varias veces con una contraseña incorrecta a los archivos de registro del servidor. Un cierto número de intentos fallidos, asegurará automáticamente que ese usuario sea "baneado", es decir, que su IP sea bloqueada durante un período determinado de tiempo. El administrador también puede configurar Fail2ban para recibir notificaciones de este tipo de accesos por correo electrónico.

Fail2ban es una solución flexible y eficaz para prevenir acciones de *bots*[e], scripts u otro tipo de ataques informáticos a un servidor. Este framework hace posible el seguimiento de archivos de registro con patrones sospechosos y permite bloquear o desbloquear sus direcciones IP temporal o indefinidamente. El usuario es libre a la hora de determinar aquellos aspectos que deben ser controlados, así como los parámetros exactos que se aplicarán durante la búsqueda.

Para instalar Fail2ban, basta con instalar un paquete debe los repositorios oficiales:

```
sudo apt install fail2ban
```

Para comprobar que la instalación es exitosa, se debe ejecutar el siguiente comando:

```
sudo systemctl status fail2ban
```

Una vez instalado el paquete Fail2ban se debe configurar el fichero jail.conf rellenando los parámetros ignoreip, bantime, maxretry y findtime en la sección [Default], de forma que sean válidos para todos los servicios que Fail2ban examine. Con estos parámetros se indican las direcciones IP que se desea que no estén prohibidas (ignoreip).

Para activar el análisis y monitorización de Nova-LTSP, se debe introducir en el fichero jail.conf dentro de la sección [Default] y justo antes de la sección [ssh-iptables] la sección [navaltsp-iptables].

En esta sección se establece un máximo de 5 intentos de acceso fallidos y se indica el uso de un fichero propio de registro denominado “fail2ban” (logpath), ubicado en el directorio /var/log/novaltsp.

Revista Cubana de Ciencias Informáticas
Vol. 15, No. Especial UCIENCIA II, Septiembre, 2021
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 42-54

La configuración del parámetro maxretry en esta sección indica que éste sólo afecta a las reglas de análisis establecidas para Nova-LTSP.

Una vez terminada la configuración de la herramienta Fail2ban y puesta en marcha, ésta creará y añadirá una cadena al cortafuego por cada servicio que analice, permitiendo así rechazar las direcciones IP con signos maliciosos.

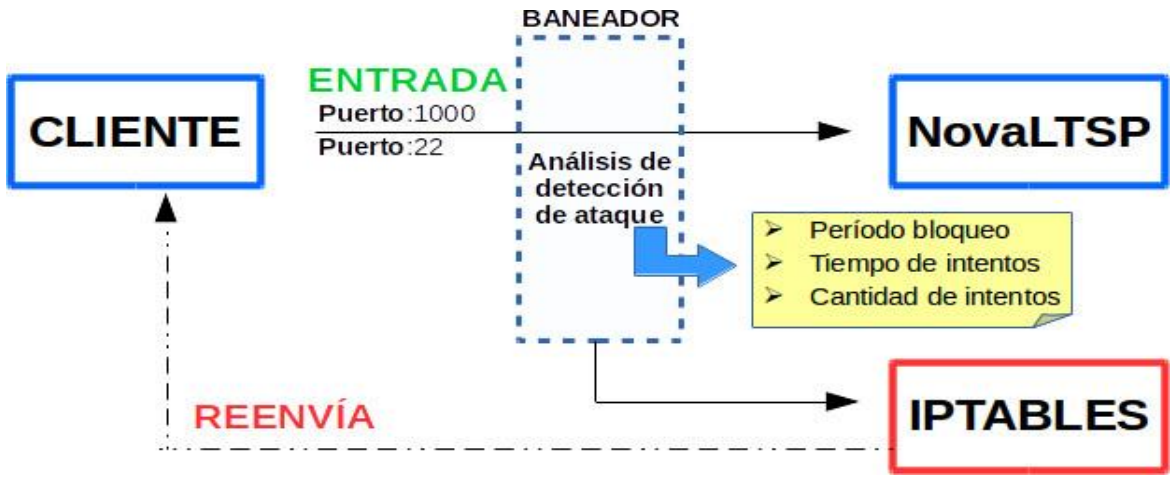

Fig.1 – Diagrama.

El cliente, persona que interactúa con la PC entra a la plataforma Nova-LTSP a través de los puerto 1000 y puerto 22. A su vez el baneador herramienta que se encarga del bloqueo del acceso a los clientes al servidor realiza un análisis de detección de ataque (período de bloque, tiempo de intentos y la cantidad de intentos), a través de iptables herramienta de cortafuegos que permite no solamente filtrar paquetes, sino también realizar traducción de direcciones de red (NAT) para IPv4 o mantener registros de log; y estas a su vez reenvían una respuesta al cliente.

Conclusiones

De manera general se puede concluir sobre la presente investigación que el análisis de los referentes teóricos y de las herramientas informáticas que implementan mecanismos de bloqueo a conexiones remotas estudiadas evidenció la necesidad de desarrollar un mecanismo que permitiera el bloqueo de conexiones

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

remotas a la plataforma de clientes ligeros Nova-LTSP. La selección de herramientas, lenguajes y tecnologías permitió la implementación de mecanismos de bloqueo a conexiones remotas que intentan accesos por fuerza bruta para Nova-LTSP. La elaboración de los artefactos propuestos por la metodología de desarrollo permitió un mejor entendimiento del trabajo, así como las características del mismo. El estudio de los sistemas homólogos permitió la selección de la herramienta fail2ban para llevar a cabo la propuesta de solución planteada. Como resultado de la implementación se obtuvo el Módulo de bloqueo a conexiones remotas a la plataforma de clientes ligeros Nova-LTSP en tiempo real.

Referencias

Bace, R. and Mell, P. Intrusion Detection Systems, NIST Special Publication on Intrusion Detection System,2015, 3: p. 1-51.

Cabeza, Y. Módulo de JAVA para la herramienta Auditoría de Código Fuente. Tesis para optar por el título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, La Habana, 2015.

Cesar H. Tarazona, T. Amenazas informáticas y seguridad de la información. Derecho Penal yCriminología,ISSN 0121-0483, ISSN-e 2346-2108, Vol. 28, Nº. 84, 2007, 137-146.

Cohn, M. Agile Estimating and Planing, ISBN 0-13-147941-5, 2005, p. 179-350.

Díaz, L.M. Módulo para el monitoreo en tiempo real de clientes ligeros desde Nova-LTSP. Tesis para optar por el título de ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, La Habana, 2017.

Gómez, D, Pérez, Y, Pérez, Y. Plataforma de Administración de clientes ligeros. VIII Taller Internacional de Tecnologías de Software Libre y Código Abierto. Universidad de las Ciencias Informáticas. La Habana, 2018.

Gómez, R. Programación Avanzada en SHELL. [En línea]. 2015. [Consultado el: 9 de marzo de 2020]. Dsiponible en [http://www.informatica.us.es/~ramon/articulos/Programacion-BASH].

Guía completa de CSS3. [En línea] 2012. [Consultado el: 20 febrero de 2021] Disponible en: [https://openlibra.com/es/book/guia-completa-de-css3].

Larman, C. UML y patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda edición. México. Prentice Hall. 1999.

OSSEC: Sistema de detección de intrusos. [En línea]. 2017. [Consultado el: 24 de marzo de 2021]. Disponible en: [https://www.mancomun.gal/es/artigo-tic/ossec-sistema-de-deteccion-de-intrusos/].

Pressman, R. Ingeniería de Sofware: Un enfoque Práctico. Quinta Edición. New York, Estados Unidos: McGraw-Hill, 2002.

Ramos G. Páez J. Análisis del método para calificación de software QSOS para la selección de software aplicable a procesos educativos. [En línea]. 2011. [Consultado el: 19 de marzo de 2021]. Disponible en: [http://scielo.sld.cu/].

Rodríguez, T. Metodología de desarrollo para la actividad productiva en la UCI. La Habana, 2014.

Sommerville, I. Software engineering, eigth edition. Harlow: Pearson Education Limited, 2007, 36p.

Wordpressg, Ingeniería de Software. [En línea]. 2012. [Consultado el: 14 de abril de 2021]. Disponible en: [https://arlethparedes.com/2012/08/27/patrones-de-arquitectura-vs-patrones-de-diseño/].

Conflicto de interés

No existe conflicto de interés de este trabajo con ninguna organización académica y/o comercial, la autora autoriza la distribución y uso del artículo.

Financiación

No se obtuvo financiamiento por parte de ninguna institución académica y/o comercial para realizar este trabajo de investigación.

a Spamming: es el hecho de enviar mensajes electrónicos (spam) no solicitados y en cantidades masivas.

b Pharming: es la explotación de una vulnerabilidad en el software de los servidores DNS (Domain Name System).

c Hacker: persona con grandes conocimientos de informática que se dedica a acceder ilegalmente a sistemas informáticos ajenos y a manipularlos.

d Cracker: el término cracker fue acuñado por primera vez hacia 1985 por hackers que se defendían de la utilización inapropiada por periodistas del término hacker.

e bots es un programa informático que efectúa automáticamente tareas repetitivas a través de Internet, cuya realización por parte de una persona sería imposible o muy tediosa