Universidad del País Vasco Euskal Herriko Unibertsitatea
BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE MASTER

GESTIÓN DE LA IDENTIDAD MEDIANTE CRIPTOTARJETAS. DIFERENTES CASOS DE USO CON CRIPTOTARJETA YUBIKEY

Estudiante: Cuadrado Alonso, Ander

Director: Jacob Taquet, Eduardo

Curso: 2022-2023

Fecha: Bilbao, 24, agosto, 2023

Resumen

En este proyecto se desarrollan diferentes casos de uso con la tarjeta Yubico yubikey: utilizar la criptotarjeta yubikey para evitar la escalada a privilegios de super usuario en una raspberry (nodo Edge); utilizar la yubikey para autenticarse en un servicio de conexión remoto al nodo Edge y utilizar la yubikey como medio de autenticación en un entorno virtualizado de trabajo orientado a empresas. Además, se encriptará la partición root de una Raspberry pi mediante la especificación de cifrado LUKS, y que para acceder a ella (desencriptarla) habrá que utilizar una Yubico yubikey.

Para ello, se securizará un nodo edge desde cero como va a ser una Raspberry pi, actualmente fáciles de encontrar en el mercado, se aplicará sobre ella ciertos comandos y configuraciones necesarias para encriptar la información que haya dentro y posteriormente se configurará para que no se pueda arrancar a menos que no esté insertada la comentada llave. Por otro lado, se realizarán las correspondientes configuraciones referidas a la escalada de privilegios y las diferentes autentificaciones de los servicios comentados, además de configurar el entorno virtualizado.

Todo ello, recabando la información necesaria para su correcta configuración, recabando las especificaciones de los dispositivos utilizados para saber exactamente qué es lo que permiten hacer y lo que no y si, con los dispositivos utilizados para toda la instalación se pueden cumplir los objetivos principales del proyecto.

Laburpena

Proiektu honetan hainbat erabilera kasu garatzen dira Yubico yubikey kripto txartelarekin: yubikey kripto txartela erabiliz Edge nodo batean (raspberry pi batean) super erabiltzaileak dituen pribilegioetara eskalatzea sahiestea; Edge nodoarekin urruneko konexio zerbitzu bat erabiliz eskatzen duen autentifikazioa yubikey-arekin bakarrik egin ahal izatea. Gainera, Raspberry baten root partizioa enkriptatuko da LUKS zifratzearen bidez eta bertara sartzeko yubico yubikey bat erabili beharko da. Bestela ezinezkoa izango da partizioa desenkriptatu. Azkenik, enpresetara bideratutako lan ingurune birtualizatu batean yubikeyaren bitartez autentifikazioa egin ahal izatea eta yubikeya izan ezean, publikatutako aplikazioetara ezin sartzea.

Horretarako, nodo edge bat zerotik sekurizatuko da, kasu honetan Raspberry pi bat, gaur egun merkatuan erraz aurkitzen direnak, barruan dagoen informazioa enkriptatzeko beharrezkoak diren komando eta konfigurazio batzuk aplikatuko zaizkio eta ondoren, konfiguratu egingo da ez dadin abiarazi, aipatutako kripto txartela txertatuta ez badago. Bestalde, pribilegioen eta autentifikazioen igoerari dagozkion konfigurazioak egingo dira eta aurretik komentatutako birtualizazio ingurunea konfiguratuko da.

Guzti hori, beharrezko informazioa bildu ostean konfigurazio akatsik ez izateko, erabilitako dispositiboen espezifikazioak analizatuz zer baimentzen duten jakiteko eta instalakuntzarako erabilitako dispositiboekin proiektuaren helburu guztiak bete diren ikusi ahal izatea.

Abstract

In this project, different use cases are developed with the Yubico yubikey card: using the yubikey crypto card to avoid escalation to super user privileges on a raspberry (Edge node); use the yubikey in a remote connection service for authentication and connection to the Edge node and use the yubikey as a means of authentication in a virtualized business-oriented work environment. In addition, the root partition of a Raspberry pi will be encrypted using the LUKS encryption specification, and to access it (decrypt it) you will have to use a Yubico yubikey. Otherwise, you would not be able to access it.

To do this and as it has been previously mentioned, we will secure an edge node from the scratch as it going to be the Raspberry pi, the ones that are currently easy to find on the market. We will apply certain commands and settings necessary to encrypt the information inside it and later on, it will be configured so that it cannot be started unless it is not inserted the commented crypto card. On the other hand, they will be configured the corresponding settings referring to the escalation of privileges and the different authentications needed for the commented services. As well as the commented virtualized environment.

All this collecting the necessary information for the correct configuration of the raspberry, collecting also the specifications of the used devices to know exactly what they allow us to do and what not and with the devices used for the installation, if we can achieve what the project is looking for.

Índice

Índice de figuras

Índice de tablas

Memoria

Introducción

En este trabajo se describirá e implementará el uso de la criptotarjeta yubikey en diferentes servicios y aplicaciones. La encriptación de la partición root de una raspberry, más concretamente de la partición root dentro de la sdcard que ejecuta las raspberry, para que así el contenido de la misma no quede expuesto a cualquier persona. Además, la autentificación sobre la partición se realizará mediante el uso de la llave yubikey, sin la cual no se verá el contenido de la sdcard que se está utilizando y que alberga el sistema operativo de la misma. Es decir, sin la autentificación correcta el sistema operativo no procederá con el arranque y por lo tanto la raspberry queda inutilizable. Lo que añadiria un plus de seguridad a la información que se tenga dentro de la comentada partición, además de evitar diferentes ciberataques, como puedan ser los ataques de phishing.

Por otro lado, la autentificación en las conexiones SSH mediante la yubikey, hace que no todo el mundo tenga la posibilidad de acceder a los recursos de la empresa, en este caso a un nodo Edge y que de este modo se eviten ataques de fuerza bruta sobre él mismo.

Tercero evitar que, sin la posesión de la yubikey, no se pueda realizar una escalada de privilegios para obtener los del super usuario root con lo que aportaría otra capa más de seguridad.

Y por último, lo mismo sucede con la autentificación mediante la cripto tarjeta para los entornos de trabajo virtualizados como es el caso del proyecto Citrix virtual apps, desktops and workspaces. Usando la yubikey de tal forma que no se puedan acceder a las aplicaciones que están publicadas en el entorno virtualizado de trabajo sin la posesión de la configurada yubikey.

Además, en entornos de trabajo donde cada vez más se utilizan diferentes contraseñas, cada vez más largas y difíciles de recordar para acceder a cualquier dispositivo hace que el uso de este tipo de llaves sea beneficioso para gestores de sistemas operativos, entre muchos otros muchos trabajadores pertenecientes a una empresa.

Contexto

Los ataques por phishing y los ataques “man-in-the-middle” han aumentado considerablemente en los últimos diez años, lo que ha acrecentado el uso de sistemas de doble factor de autentificación. Muchos de ellos hoy en día requieren de contraseñas cada vez más robustas y difíciles de recordar, ya que las tecnologías para realizar su desencriptación avanzan según se mejoran los tiempos de respuesta para dichos casos. De hecho, pocas aplicaciones existentes que más se utilizan quedan exentas de utilizar este tipo de verificación.

Así pues, el número de contraseñas a usar y que los usuarios tengan que recordar cada vez es mayor. No obstante, muchas de las soluciones que ofrece el mercado actual requieren de un segundo dispositivo disponible, muchas veces con conectividad a Internet, en el que instalar una aplicación

extra que genere dichos dobles factores de autentificación. Estos dispositivos generalmente son teléfonos móviles o cuentas de correo, con lo que conlleva: Que se quede sin batería, que no haya cobertura, acceso a Internet, que las redes a las que se conecten no sean seguras, que extraigan el teléfono móvil, etc. Lo que significa que, sin ellos, no se tendría acceso al servicio requerido.

¿Y entonces, por qué no utilizar un sistema en el que no se tenga que depender de dichas aplicaciones y de dichos dispositivos móviles para poder acceder a los servicios, como pueden ser de correo, redes sociales, etc.? Si cada vez más hay que utilizar un número mayor de contraseñas, y cada vez más robustas y con esto último se hace referencia a que cumplan parámetros de seguridad mayores como pueden ser el uso de 12 caracteres, con caracteres alfanuméricos, con mayúsculas, minúsculas, etc. ¿Por qué no simplificar todo ello un sistema/dispositivo que no requiera de memorizar todas ellas y que solo requiera de la necesidad del mismo para acceder al servicio? ¿Por qué no aplicarlo al previo arranque de sistemas operativos, incluso antes de que se pidan el nombre y usuario registrados en una base de datos? ¿Por qué no aplicar incluso para que solamente particiones encriptadas se puedan desencriptar mediante el uso de estos dispositivos? ¿Por qué no utilizarlo también para la gestión de los servidores en remoto o para evitar la escalada de privilegios de los usuarios comunes?

Por lo que, en respuesta a todas estas preguntas, se presenta este proyecto ubicado en el sector de la ciberseguridad y enfocado a la autentificación de las personas que utilicen o requieran de diferentes servicios informáticos o incluso para la desencriptación del tipo de particiones mencionadas previamente.

Objetivos y alcance del trabajo

El alcance del proyecto es poder implementarlo en cualquier tipo de dispositivo de una empresa, e incluso poder llegar a sacarle rentabilidad en un futuro, utilizando la yubikey como dispositivo para realizar el doble factor de autentificación o en su defecto realizando la implementación completa comentada previamente. Los objetivos del proyecto serían los siguientes:

Objetivo principal

Utilizar la yubikey en los siguientes servicios o casos de uso: Evitar la escalada a privilegios de super usuario en la raspberry, utilizarse en un servicio de conexión remoto (mediante el protocolo SSH) a la

raspberry y utilizarla como medio de autenticación en un entorno virtualizado orientado a las empresas. Además, se encriptará la partición root del nodo Edge para que solamente se pueda desencriptar mediante la yubikey. Y posteriormente, dejar todo documentado de manera que, si llega el caso de sacar rentabilidad al dispositivo, es decir, a la yubikey, su puesta en producción en entornos de trabajo sea más ágil de realizar.

Objetivos secundarios

Durante el desarrollo del proyecto, obtener el conocimiento necesario y conocer el manejo completo de la yubikey para poder completar los objetivos principales de manera que se reduzca considerablemente su puesta en marcha en entornos productivos, además de usarse en los diferentes tipos de casos uso comentados previamente.

Beneficios que aporta el trabajo

Como el uso de este tipo de dispositivos se puede llegar a extender a usuarios de nivel medio, es decir, personas que trabajen con aplicaciones ofimáticas, se pueden diferenciar los siguientes tipos de beneficios:

Beneficios técnicos

El encriptado de la partición primaria de la raspberry y el uso de la yubikey para poder extraer la información que hay en ella, serían de gran beneficio para poder reducir la probabilidad de ataques “phishing” y “man in-the-middle attack”. Además, su debida documentación y las pruebas realizadas para su implementación en la raspberry, hacen que su puesta en marcha en entornos de producción sea mucho más rápida que el hecho de tener que realizar las respectivas pruebas en entornos de desarrollo.

Beneficios económicos

Como ya se ha comentado en apartados anteriores, el aumento de contraseñas y aplicaciones de doble factor de autentificación, además de contraseñas con requerimientos más estrictos, más caracteres, alfanumérica, signos, etc. hace que el uso de estos dispositivos pueda tener cabida en el mercado actual y dar una salida práctica, rentable y segura en un corto o medio plazo. Sobre todo, en empresas donde no disponen de departamentos específicos para la gestión de dispositivos IT, es decir, empresas de pequeño y mediano tamaño. Principalmente el núcleo de empresas a nivel nacional aquí en España.

Beneficios sociales

Toda medida de seguridad añadida a cualquier tipo de dispositivo es un beneficio social puesto que los datos y privacidad de los usuarios no quedan expuestos a ciberdelincuentes. Por otra parte, hoy en día con el aumento de casos de suplantación de identidad en Internet hace que este dispositivo, la yubikey, sea una medida bastante eficaz para evitar tanto los ataques comentados previamente, como para este tipo de ataques de suplantación de identidad.

Análisis del estado del arte

En este apartado se realizará un estudio a cerca de las diferentes aplicaciones que, al igual que yubikey, sirven para realizar autentificación multifactor o llamadas también 2FA. Se realizará un estudio de las más significativas, es decir, las que hoy tienen más usuarios descargados. Las aplicaciones son las siguientes:

Google authenticator

Siendo gratuita su descarga, se utiliza en diferentes páginas webs y servicios, como pueden ser Gmail, Facebook, Instagram, etc. Su fácil gestión y su sencilla interfaz de usuario hacen que sea de las más populares. Sus requerimientos mínimos para poder ejecutarla en cualquier tipo de dispositivo son los siguientes [1]: 2 Mb de RAM, compatible con todos los dispositivos Android y a partir de iOS 13.0 en adelante, en dispositivos Apple. La aplicación tiene los algoritmos TOTP y HOTP para la autentificación. La explicación de dichos algoritmos es la siguiente:

TOTP

“Time-based one-time password” [2], es un código temporal que se genera basándose en la fecha y horas actuales. Los parámetros necesarios son tener una llave secreta y una semilla, llamada en inglés “seed” que permita la generación de dichos códigos. Es decir:

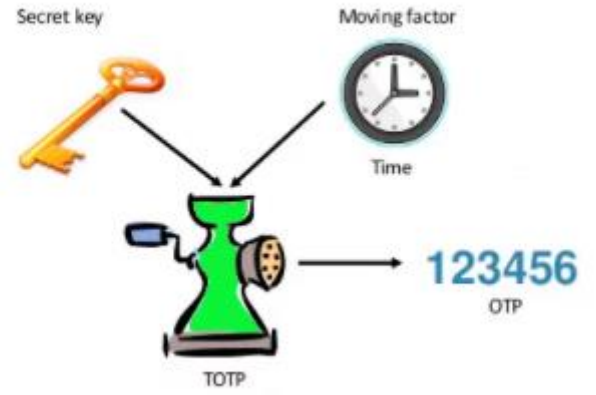


Figura 1: Diagrama de trabajo TOTP. Generación de una clave OTP [1]

Destaca también que dichos parámetros necesarios y comentados previamente son encriptados con una función hash. Una manera de explicar su funcionamiento sería con el siguiente ejemplo:

1. Un usuario intenta enrolarse en una aplicación o servicio web que lo disponga. Para que la autentificación se realice, el usuario y servidor OTP tienen que compartir una clave secreta.
2. Cuando el usuario se registra en la página web protegida mediante este método, ambos lados tienen que confirmar que disponen de la clave secreta compartida. Entonces, su token TOTP combina la semilla y el paso de tiempo actual y genera un valor HASH ejecutando una función HASH predeterminada. Este valor es esencialmente el código OTP que el usuario ve en el token.
3. Desde que la función hash y la hora son iguales para las dos partes, el servidor realiza la misma computación/calculo que el OTP del usuario.
4. Si el usuario inserta el OTP y si es idéntico que el del servidor, el acceso a la aplicación es permitido. En caso contrario, no. De manera gráfica:

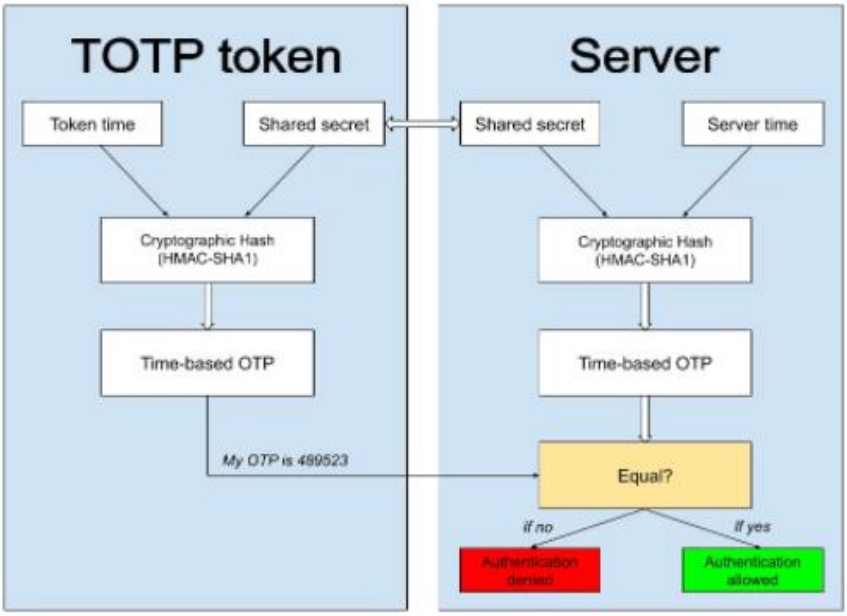


Figura 2: Flujo de funcionamiento TOTP. Comunicaciones [2]

Es más, la comentada semilla o "seed" es un string de caracteres al azar de entre 16 y 32 caracteres de longitud. Normalmente, compartir la clave secreta significa que por parte del usuario escanee un código QR proporcionado por el servidor.

HOTP

"HMAC base done-time password" [3] en vez de utilizar la hora actual y la clave secreta compartida, lo que utiliza es la clave más un contador sincronizado. Este contador suma uno a su cuenta cada vez que hay un click por parte del usuario, además de con su respectiva y validada OTP. De manera gráfica trabaja así:

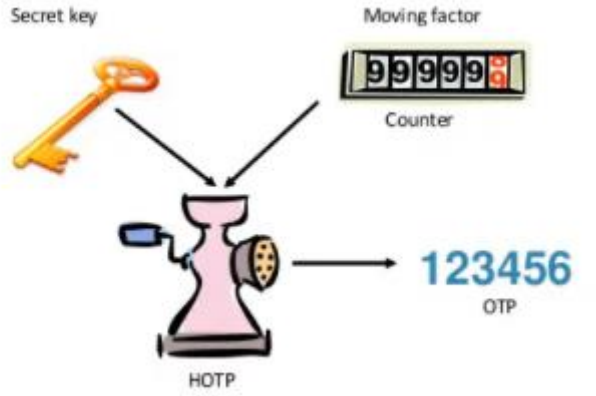


Figura 3: Diagrama de trabajo HOTP. Generación de una clave OTP [3]

El HOTP es creado previamente generando un HMAC hash desde la semilla y el contador. Y el resultado de la ecuación en un string de 160 caracteres. Posteriormente es reducido a 6 u 8 dígitos, que será el OTP expuesto en el token. Sin embargo, este algoritmo tiene un problema de sincronización puesto que haciendo click demasiadas veces se puede llegar a dar el caso de que el OTP obtenido sea no válido para el servidor, y por lo tanto no deje realizar el correspondiente registro.
Además. Google authenticator no necesita conectividad a Internet para su uso [4]. Puesto que las OTPs se generan de la misma forma, y si la generada coincide con la que tiene el servidor de autenticación se podrá registrarte en la aplicación. Además, permite tener más de una cuenta registrada en un mismo dispositivo. La garantía que ofrece esta aplicación es que si bien es cierto que el atacante puede llegar a obtener la contraseña del usuario porque esta última es débil o fácil de descifrar, necesita obligatoriamente el dispositivo móvil que tiene la aplicación para poder obtener el token que le permita acceder a la aplicación. Finalmente, los códigos generados por la aplicación cambian por motivos de seguridad cada 30 o 60 segundos, dependiendo de la configuración de la que se disponga.

Por otro lado, los datos que tiene la aplicación en el dispositivo no son transportables a otro. Cuando la aplicación se borra, las cuentas configuradas de tal manera también se borran y habría que configurarlas de nuevo en la aplicación nueva. Se recomienda además que no se tenga configuradas más de 4 cuentas para que así se puedan ver de una misma pasada los tokens generados por la aplicación. Otra de las desventajas de las que dispone también es que no tiene ningún tipo de contraseña para proteger la aplicación.

Protectimus Smart OTP

Aplicación de descarga gratuita para iOS y Android, dirigido a dispositivos que soporten Android Smart Watch [5]. Entre sus características principales se pueden destacar las siguientes:

- Tiene disponible diferentes tipos de algoritmos OTP, como pueden ser los ya mencionados antes HOTP y TOTP e incluso OCRA. Posteriormente se realizará una breve descripción del algoritmo este último algoritmo.
- Tiene la protección por código PIN. Es decir, evitar acceso no permitidos mediante la generación de códigos PIN.
- Entre 6 y 8 dígitos de longitud para los códigos OTP. Posibilitando el cambio de uno a otro, y viceversa.
- Tiene soporte para Smart Watches.
- Dispone de uno de los sistemas de autentificación por multifactor como es el caso de CWYS, que al igual que el algoritmo OCRA, también se describirá posteriormente.

No obstante, también tiene sus contras. Al igual que la mayoría de aplicaciones de este tipo no dispone de copia de seguridad cuando se cambie de dispositivo. Sin embargo, desarrolladores de la aplicación confirman que añadirán las siguientes características a la aplicación en futuras actualizaciones:

- Copia de seguridad en la nube con todos los tokens creados.
- Darán soporte para fallos en la autenticación por escáner biométrico para los dispositivos que lo tengan.
- Implementaran la posibilidad de aumentar el número de caracteres de los códigos PIN hasta un máximo de 10.
- Implementara notificaciones “push” para la función comentada previamente CWYS, con su respectivo soporte.
- Ampliaran la posibilidad de proteger la aplicación no solo con código PIN, sino también un token hardware.
- Habrá posibilidad de utilizar la app para programar hardware OTPS y para Protectimus Slim NFC.
- Se implementará el algoritmo CWYS que posteriormente se explicará.

Como se ha hecho previamente con los algoritmos TOTP y HOTP, se pasará ahora a describir los comentados CWYS y OCRA. Por lo tanto:

OCRA

Hoy en día es el algoritmo de desafío-respuesta más fiable que existe [6]. Se basa en utilizar un "challenge" o desafío de entrada para generar el código de acceso único junto con la clave secreta (seed) y un contador o la fecha y horas actuales. En este caso, la diferencia con la versión anterior es la capacidad de identificar el servidor, ya que el usuario final puede asegurar la veracidad del servidor con quien actúa. Normalmente el token es un dispositivo con un teclado o una aplicación. Se expondrá un ejemplo de caso de uso:

- La web o aplicación a la que el usuario intenta registrarse, provee de un código al usuario.
- El usuario necesita insertar el código recibido en el token que posteriormente le genera otro código.
- Y finalmente el usuario inserta este último token generado para registrarse en la correspondiente aplicación.

En la siguiente imagen se muestra como realiza la generación de la OTP a insertar en la aplicación:

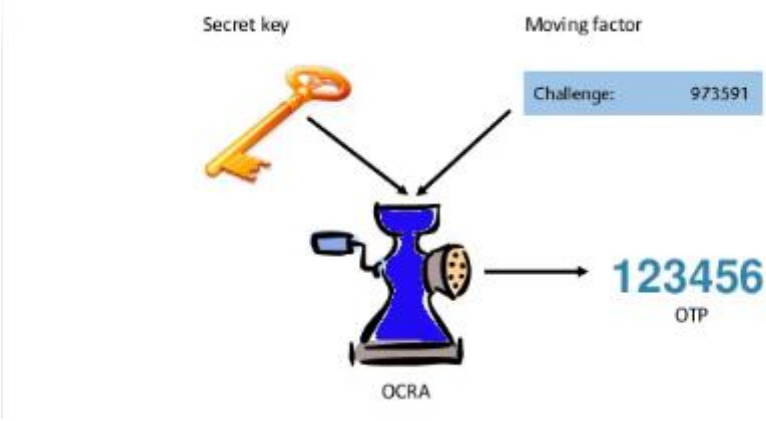


Figura 4: Diagrama de trabajo OCRA. Generación de una clave OTP [4]

Además, tiene una implementación extra. La firma de las transacciones de datos. En este método, el cliente puede verificar y confirmar la transacción mediante la confirmación de ciertos parámetros, como por ejemplo un número de cuenta, la concurrencia de la misma, la cantidad, etc. De esta manera se crearía el siguiente método: "Confirm What You See", o CWYS.

Confirm What You See

Este algoritmo usa las transacciones del usuario actuales para la generación de las OTPs. Por ejemplo:

- El usuario final necesita transferir una cantidad de dinero.
- Cuando la transacción es creada el sistema protegido, en este caso el banco, manda una petición para firmar los datos generados en la misma, y la aplicación "Protectimus" genera un código QR, que el usuario escanea y verifica si los datos mostrados concuerdan. Si todo es correcto, el usuario inserta la OTP generada por su token para realizar la transacción.

En la siguiente imagen se muestra lo explicado previamente:

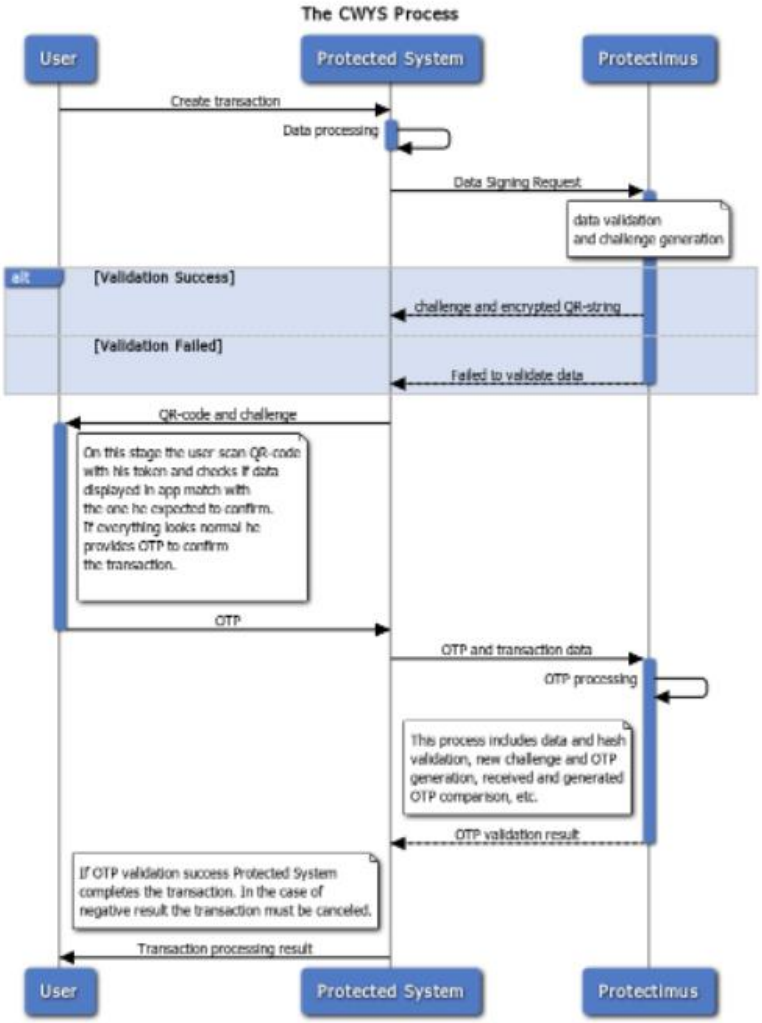


Figura 5: Diagrama de trabajo Confirn what you see [5]

Por lo que para enviar información a al servicio Protectimus, y recibir el "desafío", el usuario tiene que utilizar el método POST a la siguiente página web, ***https://api.protectimus.com/api/v1/token-service/tokens/sign-transaction*** con los siguientes parámetros:

- tokeind: el identificador del token del usuario.
- transactionData: la información para la generación del OTP.
- hash: para verificar la integridad de la información recibida. La API key del usuario es la que se usa como llave.

En respuesta se obtiene un XML o JSON con los siguientes parámetros de información:

- challenge: el "desafio" para el algoritmo de generación del OTP.
- transactionData: los detalles de la transacción, encriptados.
- tokenType: el tipo de token.
- id: el identificador del token.

Por lo que para usuarios que tienen la opción "Smart token", tiene que ser generado un código QR y tiene que ser mostrado. Para usuarios con otros tipos de tokens o para usuarios que no les sea posible escanear el código QR generado, debe mostrarse por pantalla qué deben ingresar, del modo que les sea posible, el token para la generación del OTP de registros en la aplicación.

Authy 2-Factor Authentication

Del mismo modo que las demás, sirve para que registrarse en aplicaciones o páginas webs utilizando el doble factor de autentificación [7]. Entre sus características se encuentran las siguientes:

- Tiene una versión para ordenadores. Independientemente de que se utilice un smartphone o un ordenador, los desarrolladores de esta aplicación tuvieron en cuenta ambos entornos.
- Tiene copia de seguridad en la nube. Por lo que, en caso de pérdida del dispositivo, se pueden recuperar las cuentas en las que te registraste gracias a este servicio.
- Da la oportunidad de bloquear la misma aplicación para que terceros no puedan acceder a ella.
- Dispone de sincronización en múltiples dispositivos, Por eso no se necesita escanear el código QR necesario para la creación de los tokens.

Además de las virtudes comentadas, también tiene sus defectos. Entre ellos que su sistema de autentificación se basa en el envío de SMS, lo que va en contra de los estándares de 2FA [8].

Microsoft Authenticator

Se puede usar para todos los servicios que ofrece Microsoft donde ejerce como aplicación 2FA, además de gestor de contraseñas [9]. Esta última función permite que, si se guarda una contraseña en el buscador de Microsoft Edge, pase a sincronizarse también con la aplicación del dispositivo móvil. Otra característica diferente en comparación con las anteriores es que la aplicación no permite sacar una captura de pantalla, además de que las configuraciones de inicio de sesión no se sincronizan a través de la cuenta de Microsoft. La aplicación permite crear el código de verificación OAuth de la misma forma que las aplicaciones de este tipo. Por otro lado, siempre que se intente registrar en una aplicación o servicio utilizando esta aplicación, automáticamente se enviara una notificación al dispositivo móvil para que se verifique el acceso al mismo. Entre sus características tiene diferentes tipos de autentificación: basado en SMS (no recomendada por los estándares 2FA), basado en el envío de email, generación de token por software y hardware [10]. Dispone también del factor de autentificación biométrico, entre otros. Entre sus funcionalidades, destacan la posibilidad de sincronizar con diferentes dispositivos, copia de seguridad de las cuentas registradas en la aplicación y, por último, el poder usarse la aplicación de manera offline.
Su implementación es simple e intuitiva, y dispone de implementación Web SDK como Mobile SDK. Permite también la monitorización del uso de la aplicación en una organización empresarial, siempre y cuando se utilice Microsoft Intune.

FreeOTP Authenticator

De la misma forma que las anteriores, y aunque es open-source, se puede usar en la mayoría de las aplicaciones. A diferencia de las anteriores, el coste de almacenamiento en comparación con las otras comentadas previamente es remarcable, puesto que solamente ocupa 500Kbs de almacenamiento [11]. Tiene integrados los algoritmos HOTP y TOTP. Al igual que Microsoft Authenticator, se puede usar sin Internet ya que sigue generando los correspondientes tokens. Los tokens se generan mediante el escaneo de un código QR. Por el contrario, no ofrece copia de seguridad de las cuentas registradas en la aplicación y la interfaz de usuario debería mejorar en comparación con las anteriores [12]. Su uso estaría más enfocado no a personas físicas. Por el contrario, la aplicación no está disponible para ordenadores.

Sophos Authenticator

De acuerdo con los RFC 6238 y RFC 4226, Sophos ofrece la creación de contraseñas basandose tanto en contadores como en la fecha y hora actuales [13]. Para la generación de los códigos no necesita de

conexión a Internet. Soporta los algoritmos SHA-1, SHA-256 y SHA-512. Se puede registrar una cuenta escaneando directamente un código QR. La generación por TOTP está limitado a 30 segundos. A partir de ahí se genera un nuevo código. Los códigos pueden ser de 6 a 8 dígitos de longitud. Las contraseñas se guardan de manera encriptada en el "llavero" de la aplicación. Y es fácil de usar. Los códigos generados se pueden copiar y pegar directamente clickando encima del código. Por otro lado, las desventajas que tiene son las siguientes:

- No tiene posibilidad de hacer copia de seguridad de las cuentas registradas en la aplicación.
- Diferentes usuarios afirman que la aplicación da problemas cuando se usan túneles VPN, con conexión Ipsec o cuando se utiliza en una red con un servidor de radius.

Authenticator Plus

En la versión gratuita es parecida a Google Authenticator pero en la de pago se añaden características que la mejorarían la comparación entre ambas [14]. Las características más destacadas de la aplicación son las siguientes:

- Copias de seguridad automáticas. La versión de pago permite realizar backups en diferentes plataformas en la nube, como pueden ser Dropbox, Google Drive, etc.
- Se puede realizar una organización distinta acorde a tus necesidades. Es decir, se puede crear carpetas, grupos de cuentas, etc. Permite además cambiar de temas e iconos para hacerla más amigable al usuario.
- Permite sincronizarse entre diferentes dispositivos de manera intuitiva.
- La información de la aplicación esta encriptada con el algoritmo 256-bit AES y se requiere de pin para poder acceder a la misma.
- Dispone de clave de cifrado basado en hardware.

Por el contrario, para las funcionalidades más completas de la aplicación se requiere de la versión de pago. Además, no hay soporte ni manuales y en diferentes aplicaciones como pueden ser Amazon Fire y los smartphones Kindle Fire, dan problemas para el escaneo del código de barras. Por otro lado, a partir de una versión Android, la aplicación ha empezado a dar errores. Especialmente con Google Backup, donde la función de exportar la información ha dejado de funcionar.

LastPass Authenticator

Para la autentificación, esta aplicación utiliza lo denominado “one-tap veriification” o notificaciones push. Es decir, que en la aplicación cuando se vaya a registrar en una aplicación, genere una notificación al instante. La aplicación da la oportunidad de trabajar con una cuenta LastPass o con cuentas de terceros [15]. Es compatible con los smartwhatch, recibiendo en ellos la notificación de intento de registro en la aplicación. Además, permite el backup en la nube, aunque esta característica no es automática. Otra de las características es que al igual que alguna anterior, genera códigos de 6 dígitos cada 30 segundos para poder registrarse en las aplicaciones. Tiene además soporte para programar que se reciban los códigos por SMS o escaneando códigos QR. Es compatible con el protocolo de Google Authenticator y muchos otros TOTP. De hecho, da la oportunidad de asignar “dispositivos de confianza”, para que con solo una vez aceptar el código 2FA sea suficiente. Por el contrario, tiene las siguientes desventajas:

- Para usar la opción “one-tap” se necesita haber creado previamente una cuenta en LastPass. Además, es necesario instalar una extensión en todos los navegadores donde se vaya usar la aplicación.
- Las notificaciones automáticas “one-tap” están solo disponibles para ciertas páginas webs, por lo que sería recomendable que en futuras versiones del producto se amplíe el catálogo de las páginas webs en donde usarlo.
- La longitud de los códigos generados es corta.

SoundLogin

A diferencia de todas las anteriores, esta aplicación utiliza la voz para la generación de códigos de un solo uso. Entre sus características se destacan las siguientes [16]:

- La codificación de la señal utilizada para la generación de los códigos se procesa de manera local.
- Con el uso de la voz, se evita que el usuario tenga que insertar el código de manera manual todas las veces que se intente registrar a un servicio o página web. Solo es necesario apuntar el móvil al micrófono del ordenador y la app rellenara el formulario de manera automática.
- Es la más nueva de las aplicaciones o métodos de autentificación por 2FA.
- Puede extraer las OTPs desde SMSs u otras apps, codificarlas y enviarlas de buen al navegador para posterior procesamiento. Estas aplicaciones podrían ser Google Authenticator o FreeOTP

Entre los servicios donde se pueden utilizar se destacan los siguientes:

- En redes sociales:

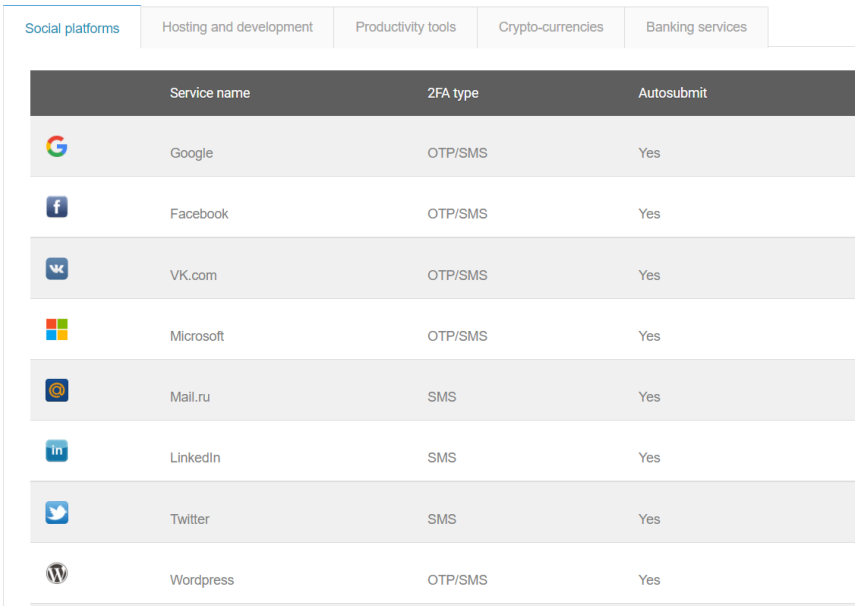


Figura 6: Casos de uso SoundLogin. Redes sociales [6]

- En entornos de programación:

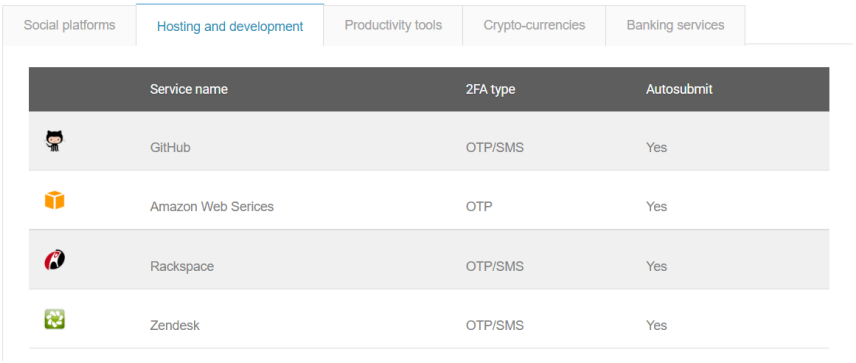


Figura 7: Casos de uso SoundLogin. Entornos de programación [7]

- En herramientas de productividad:

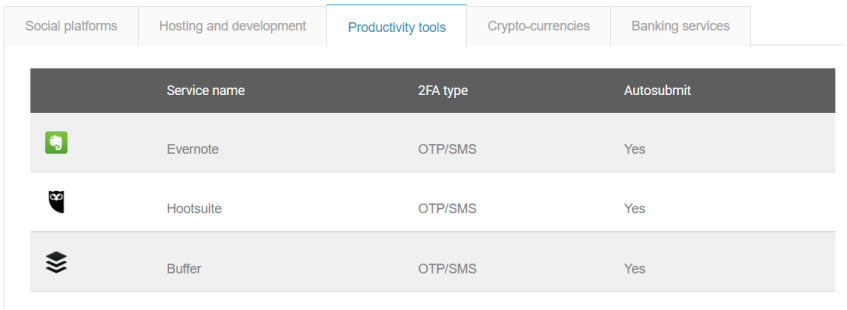


Figura 8: Casos de uso SoundLogin. Herramientas de utilidad [8]

- En servicios bancarios:

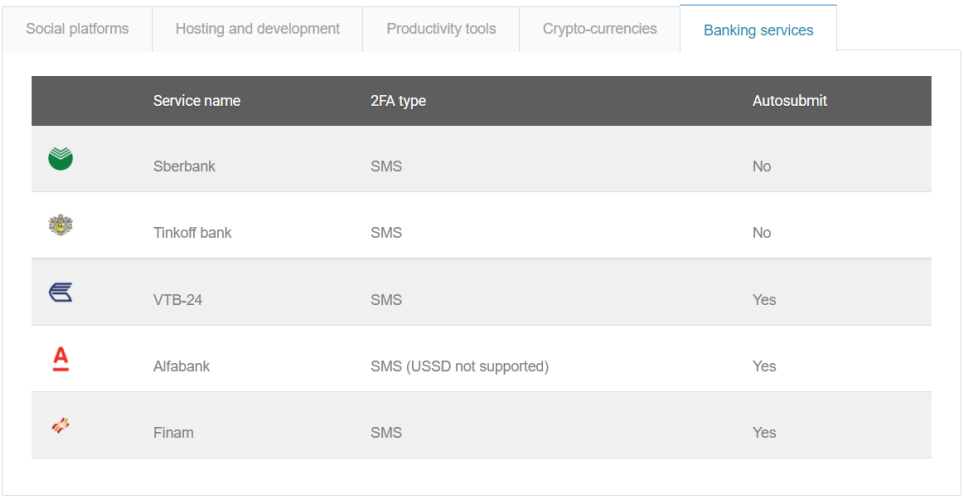


Figura 9: Casos de uso SoundLogin. Servicios bancarios [9]

En contra, es un proyecto con poco recorrido. No es recomendable tenerlo como única cuenta 2FA. No ofrece documentación "real" ni soporte [17]. Solo se puede poner en contacto rellenando exclusivamente un formulario y esperar la respuesta del equipo. Por otro lado, solo se puede trabajar con la aplicación si se dispone de micrófonos y se necesita tener instalado en el navegador una extensión de SoundLogin.

Hasta este punto se han comentado las diferentes posibilidades que hay en el mercado con respecto a las aplicaciones 2FA. Hay más, pero las comentadas hasta este punto son las más usadas a nivel mundial. A continuación, se presentará una tabla en la que se comparan muchas de las comentadas previamente para que se vean las diferentes características a la vez:

	Google Authenticator	Authy	Yubico	DUO	FreeOTP	Authenticator Plus	SoundLogin
FIDO (U2F)	✓	✓	✓	✓	✓	✓	—
Multiple Token	✓	✓	✓	✓	✓	✓	✓
Smartphones	✓	✓	—	✓	✓	✓	✓
Desktops	—	✓	✓	✓	—	—	—
Open Source	—	✓	—	✓	✓	—	—
Multiple Device Synching	✓	✓	—	✓	—	✓	—
Offline Mode	✓	✓	✓	✓	✓	✓	✓
RDM Integration	✓	—	✓	✓	—	—	—

Tabla 1: Comparación de características entre diferentes aplicaciones MFA [1]

Hay también, por otro lado, dispositivos hardware que generan los OTP tokens. Para realizar un estudio de su estado actual, se ha decidido realizarlo de manera separada de las anteriores, puesto que las anteriores son soluciones software y la siguientes son soluciones hardware. Previamente se comentará como está el mercado actual sobre su uso y compra y posteriormente se hablará sobre las que están en el mercado hoy en día.

Estado actual del mercado

Hoy en día el mercado de este tipo de OTPs esta segmentado por tipo: necesidad de estar conectado o no, si es contactless o no, usuario final enfocado a la industria (servicios bancarios, servicios de seguridad o financieros, gubernamental, empresa de seguridad) y geografía. Si se segmenta el mercado mundial por regiones, la que más rápido crece es la de Asia, y el mercado con más participación es el norteamericano [18].

En lo que se refiere al capital del mercado, en 2021 movió alrededor de 1.2 billones de dólares, y se espera que para 2027 alcance los 1.8 billones de dólares [19]. La tecnología se basa en que la primera autentificación continua necesariamente con la segunda, que se basa en el uso del dispositivo hardware. Cabe destacar las siguientes características en cuanto su uso y seguridad:

- La portabilidad de este tipo de dispositivos hardware elimina la necesidad de instalar software y hardware externos. La seguridad proporcionada por estos tokens es un factor importante que refuerza su demanda en las diversas industrias de usuarios finales en todo el mundo.
- Los usuarios de empresas, gobiernos, empresas trabajando para el sector de la salud y muchas otras identidades son parte del mercado de este tipo de dispositivos que necesitan de múltiples factores de autentificación para que se les otorgue acceso a la información o el uso de servicios. De ahí su gran demanda.
- Aplicaciones OTP basadas en software, SMS, emails, etc. son sustitutos válidos para la generación de tokens OTP, pero estos dispositivos hardware son considerados más seguros ya que no necesitan de acceso a una red para obtener la contraseña o el necesario PIN, por lo que no pueden ser objetivo de ataques que involucren a dichas redes.

Tendencias del mercado

El mercado que involucra a los dispositivos sin necesidad de conexión para el proceso de autentificación es el que más capital genera y mueve en comparación a los de otros subtipos de mercado comentados previamente. Este tipo de tokens requieren que el número que generan se copie en el campo de código de acceso, no tienen una conexión lógica ni física con la computadora cliente, no requieren de un dispositivo de entrada especial y en su lugar, usan una pantalla integrada para mostrar los datos de autenticación generados, que el usuario ingresa manualmente a través de un teclado [20].

En cuanto al uso de este tipo de dispositivos, Norte América va en cabeza debido a pronta adopción de los mismos. Este país es el que más infracciones reporta, con 2330 infracciones publicadas, muy por delante de Reino Unido, con 184 informadas.

Por otro lado, según el consejo de asesores de la Casa Blanca, la economía estadounidense perdió alrededor de 57 000 y 109 000 millones de dólares por tales infracciones relacionadas a actividades cibernéticas peligrosas, dato que refleja el aumento de estos dispositivos ya que añade una capa extra de seguridad a los usuarios para que se protejan a sí mismos [21].

Panorama competitivo

En cuanto a los proveedores de este tipo de dispositivos, hay muy pocos que acaparan todo el mercado en términos de cuota de mercado, quienes tratan de expandir su base de clientes entre los usuarios finales [22]. De hecho, están aprovechando las iniciativas estratégicas de colaboración para aumentar su cuota de mercado y por ende su rentabilidad. Datos que avalan lo comentado:

- En 2017, RSA, una empresa subsidiaria de "Dell Technologies", expandió su ecosistema para ofrecer una interoperabilidad entre servicios como "CyberArk Privileged Account Security Solution", "Microsoft Windows Hello", "Palo Alto Networks Next-Generation Firewall" y VMware Workspace ONE", por lo que de manera rápida y fácil estos usuarios podían aprovecharse un conjunto de métodos modernos de autentificación móvil de la solución propuesta por RSA.
- En 2019, la empresa "Thales" completo la adquisición de la empresa "Gemalto", conglomerado de dos empresas "Euronext Amsterdam" y "Paris: GTO", obteniendo el liderazgo mundial en cuanto las identidades digitales y su seguridad, empleando con esta adquisición a un total de 80 000 trabajadores.

Una vez comentado el panorama actual del mercado de dispositivo hardware OTP, se pasará a comentar los dispositivos más vendidos hoy en día. Para ello, se ha realizado un estudio en el que se especifican por categorías cuales han sido las mejores en su campo. A continuación, se muestra una tabla de las comentadas, se comentará qué es lo que hoy en día se está buscando en este tipo de dispositivos y posteriormente, por categoría, se hablarán sobre ellas:

Best Overall Security Key	Best Premium Security Key	Best Security Key for Bio-authentication	Best Key & Password Manager Combo	Best Open-Source Security Key
				
Yubico Yubico FIDO Security Key NFC	**Yubico** YubiKey 5 NFC USB-A	**Kensington** Kensington VeriMark	**OnlyKey** OnlyKey	**Nitrokey** Nitrokey FIDO2

Figura 10: Selección de las diferentes cripto tarjetas disponibles en el mercado y su categorización por valoración [10]

Estándar FIDO2

Actualmente, la forma de autentificarse a diferentes servicios de las diferentes plataformas se debe dar de manera casi universal. Esta universalidad la garantiza el estándar FIDO2, que lo incluyen empresas como Google, Microsoft, PayPal, American Express, MasterCard, VISA, Intel, ARM, Samsung, etc. Cabe comentar también que casi todos los dispositivos que soporten este estándar también soportan FIDO U2F, una nueva versión del mismo [23].

Por otro lado, hay quienes buscan características extra, como OTPs, encriptación de emails como lo puede realizar OpenPGP añadiendo una capa extra de seguridad al envío y recibo de emails, etc.
Dicho lo cual, se pasará a detallar las características de los dispositivos enseñados en la anterior imagen:

Yubico FIDO Security Key NFC

Basándose en la tabla expuesta anteriormente, y en comparación con las demás, es la más asequible y contiene las características que la mayoría de usuarios necesitan. Contiene los estándares U2F y FIDO

2, ya comentados previamente y tiene soporte para autentificación web, CTAP 1 y 2 y U2F. A continuación, se hace un paréntesis porque se procederá a la explicación de CTAP.

CTAP

Programa de evaluación de amenazas cibernéticas es un programa rápido y gratuito que se usa para identificar los riesgos y amenazas para la empresa y ayuda a la compresión de los mismos. El proceso comienza con la evaluación de las amenazas [24]. Posteriormente continua con la revisión de su gravedad y la creación de planes para abordar la vulnerabilidad subyacente. Finalmente, se realiza un plan de seguimiento y planes de mitigación. En la última fase del mismo, si las amenazas continúan siendo creíbles y muy probables, los equipos de seguridad utilizan indicadores en su determinación de viabilidad, también llamados RSIF. Es decir, Recurrencia, Severidad, Intensidad y Frecuencia. Este tipo de programas están enfocados en las amenazas ofensivas o dirigidas a entidades en concreto. Difiere de una evaluación de vulnerabilidad, que se ocupa de las amenazas que miden la propia capacidad defensiva de un objetivo para responder a las amenazas en su contra.

Una vez explicado lo que es un CTAP se continuará con la descripción de las características del comentado dispositivo. Además de poder trabajar con los estándares FIDO U2F y FIDO 2, incluye la posibilidad de utilizarla como NFC para registrarse o rellenar los campos requeridos. Soporta la mayoría de los MFAs del mercado y se puede usar tanto en dispositivos móviles como PCs y es resistente al agua. En cuanto a sus desventajas, la principal es que no soporta protocolos más avanzados que los comentados previamente, lo que hace que su precio sea inferior.

Yubikey 5 NFC USB-A

Además de ofrecer las características de la anterior, puesto que el dispositivo es del mismo fabricante, se pueden utilizar además de FIDO U2F y FIDO 2, protocolos más avanzados. En cuanto al ratio IP, ofrece un IP-67, es decir el nivel más alto. Esta ratio ofrece completa protección contra la entrada de polvo y partículas de aire en la yubikey, aparte de no permitir la entrada de agua u otros tipos de líquido que hace que la llave sea muy robusta y difícil de romper y sufrir degradación del entorno.

En cuanto a dónde se puede utilizar, el hecho de soportar la gran mayoría de protocolos que existen en el mercado, hace que pocos servicios queden excluidos. Usando la aplicación del fabricante se acceden a opciones más avanzadas, como puede ser el poder utilizar OpenPGP, realizar una firma segura para autenticar las comunicaciones u OTPs. Es más, con este tipo de dispositivos, se puede

enviar un email encriptado con ProtonMain, usando una PGP, pero en lugar de depender de una clave publica, puede usarse la clave que proporciona este dispositivo hardware. Dispone además de la opción de programarse para que, en vez de una generación aleatoria de contraseñas, genere siempre la misma cuando se toque el botón que dispone. Se pueden escribir hasta 32 caracteres para la generación de la contraseña, que el resto de caracteres lo completa la yubikey.

Las desventajas son, por un lado, su precio y que puede ser complicado utilizarlo en dispositivos móviles, ya que fuera a parte de los navegadores en donde sí que va bien, en muchas aplicaciones se pide que se rellenen formularios de autentificación en otra aplicación, lo que hace fallar a la yubikey.

Kensington VeriMark

Tiene un gran lector de huellas que a diferencia de las yubikey hace de función re-captcha, lo que diferencia a sus usuarios de posibles ataques maliciosos. Su diseño hardware está más orientado a que este de manera continua conectado a una de las bocas del ordenador, antes que ser transportable, y de tener que transportarse dispone de tapa. Soporta el protocolo FIDO 2 y se puede usar con Windows Hello. A diferencia de las yubikey, este dispositivo esta orientado a usarse en dispositivos Windows, ya que la configuración para usarse en IOS o Linux es más complicada.

En términos de seguridad, las huellas dactilares utilizadas para la autentificación no se guardan en la memoria de la llave si no en una plantilla dentro del sistema operativo. La captación de la huella dactilar funciona desde cualquier ángulo, por lo que el diseño del sensor marca la diferencia con las demás.

El mayor inconveniente es la falta de NFC, lo que deja fuera del mercado a los iPhone, salvo que se opte por la versión de escritorio con un cable USB. Si se quiere utilizar de esta manera, se tendrán que agregar configuraciones extra para el correcto uso del dispositivo. Su precio también es notablemente alto (alrededor de unos 60 dólares).

Onlykey

A diferencia de las demás, ofrece un administrador de contraseñas como parte hardware de la llave. Esto hace que un registrador de teclas no tenga posibilidad de obtener la contraseña que utiliza. Su uso es sencillo ya que solo hace falta pulsar una de las seis teclas para ingresar la contraseña que se requiera a un campo de texto. Además, se pueden programar para que realizando pulsaciones más largas se amplíe el número de contraseñas hasta 12. Incluso se programa para que cuando se use una de las contraseñas, se tenga que ingresar un PIN de seguridad, añadiendo una capa extra de seguridad.

Soporta 2FA, TOTP, Yubico OTP y FIDO 2 U2F, lo que permite usarse en la mayoría de apps o servicios. Se puede incluso programar un código de auto destrucción, lo que hace poner de fabrica la llave, borrando así las contraseñas que había en ella.

La desventaja más significativa es que tiene una interfaz no muy amigable. Esto hace que aquellos que no son muy expertos en tecnología tengan dificultades para usarlo. Carece de NFC o Bluetooth y es más grande en comparación con las comentadas previamente, factores a tener en cuenta.

Nitrokey FIDO2

En comparación con las anteriores no tiene las mismas implementaciones, pero es open-source y barata. Dispone soporte para una gran variedad de protocolos, como pueden ser los ya comentados FIDO U2F, FIDO 2 o WebAuthn /CTAP, lo que hace que se pueda usar en la mayoría de aplicaciones que requieran de doble autentificación.

Al ser de código abierto, todo el mundo puede mirar el firmware de la llave y asegurarse de que no tenga vulnerabilidades. Es de uso fácil y tiene un acabado profesional, además de proteger el pulsador con una capa de plástico.

Por el contrario, puede ser problemática utilizándose en otro dispositivo que no sea el ordenador. No tiene lector de huellas y tampoco dispone de un gestor de contraseñas. Además, no dispone de NFC y requiere conocimiento técnico para poder configurarla.

Una vez comentadas todas, se muestra a continuación una tabla en la que se destacan las características de cada una describiendo sus ventajas y desventajas:

	Best Overall Security Key	Best Premium Security Key	Best Security Key for Bio-authentication	Best Key & Password Manager Combo	Best Open-Source Security Key
	 **Yubico** Yubico FIDO Security Key NFC	 **Yubico** YubiKey 5 NFC USB A	 **Kensington** Kensington VeriMark	 **OnlyKey** OnlyKey	 **Nitrokey** Nitrokey FIDO2
PROS	✓ Afford able yet still has all the security features most people will need ✓ Has FIDO U2F and FIDO 2 which is used by most of the big names ✓ Protocol support for WebAuthn, CTAP 1, CTAP 2, U2F ✓ Includes NFC	✓ Wide range of protocol support ✓ Several port versions available ✓ IP67 rated and with no moving parts makes it very sturdy	✓ Excellent fingerprint reader ✓ Support for most popular forms of MFA ✓ Small and portable	✓ Can bypass keyloggers ✓ Has self-destruct emergency code ✓ Wide protocol support	✓ Open Source ✓ Relatively cheap ✓ Wide protocol support
CONS	✗ Doesn't have support for more advanced protocols	✗ Expensive for those who don't need the added features	✗ Use on non-Windows platforms can be difficult ✗ Lack of NFC	✗ UI can be a bit obtuse ✗ Bulkier than other security keys ✗ Lack of NFC	✗ Lack of NFC ✗ Requires technical knowledge

Figura 11: Pros y contras de las cripto tarjetas especificadas [11]

Análisis de alternativas

Tras realizarse un estudio de las diferentes tecnologías que hay hoy en día en el mercado y tener una idea global de lo que el mercado ofrece, en este apartado se hará hincapié en lo que a las cripto tarjetas se refiere, ya que son las tecnologías que permiten cumplir con los objetivos del proyecto. Podría hacerse también un estudio de las diferentes Raspberry pis que hay en el mercado, pero al disponer únicamente de un modelo, este estudio se descartará. Por lo tanto, se procederá a realizar la comparativa de entre todas las que se han comentado previamente, tres: la Yubico Yubico FIDO Security Key NFC, Yubico Yubikey 5 NFC USB A y la OnlyKey Onlykey. Dos de ellas pertenecen al mismo fabricante ya que para realización del proyecto se ha contado con una Yubico Yubikey 4, yubikey de una generación previa a las comentadas, pero reúne mayoritariamente las características de las comentadas. La última y aunque sea de un proveedor diferente se ha seleccionado por la gran variedad de posibilidades funcionales que ofrece. Por lo tanto:

Yubico FIDO Security Key NFC

Basada en la autentificación hardware, proporciona una gran protección contra el phishing, elimina la usurpación de cuentas y habilita los requisitos de cumplimiento para una autentificación estable y sólida [25]. Ofrece soporte para FIDO U2F y FIDO 2. Al soportar este tipo de estándares, permite la migración de las organizaciones a entornos de trabajo sin contraseña ya que la autentificación estaría basada en la autentificación mediante hardware. Puesto que está basada en la autentificación hardware, tiene tiempos de respuestas hasta 4 veces más rápidas que los métodos con dispositivos móviles o SMS. Su facilidad de uso, hace que el usuario final adopte esta tecnología sin una gran dificultad. Dispone del mecanismo NFC para la autentificación, permite la gestión de identidades y gestión de acceso en los siguientes sistemas de gestión:

- AWS Identity and Access Management (IAM)
- Centrify
- Duo Security
- Google Cloud Identity
- Microsoft Azure AD
- Okta
- Ping Identity

En entornos de producción se puede utilizar para las Google Accounts, Microsoft Accounts y Salesforce.com. Entre los gestores de contraseñas en los que se puede utilizar se encuentra 1Password, Keeper y Bitwarden Premium. Dispone además de los siguientes certificados:

- FIDO2
- FIDo Universal 2nd Factor Certified

En cuento a las especificaciones criptográficas, dispone de ECC p256, es decir, de Criptografía de Curva Elíptica (Variante de la criptografía asimétrica o de clave publica que está basada en las matemáticas de las curvas elípticas).

Yubico Yubikey 5 NFC USB A

Aun siendo del mismo proveedor, a diferencia de la anterior ofrece soporte para los siguientes protocolos [26]:

- FIDO2

- Yubico OTP
- OATH
- HOTP
- U2F
- PIV
- Open PGP

No solo dispone de autentificación por doble factor, sino que además ofrece autentificación sin contraseña y multifactor para la autentificación con el toque por parte de usuario en el botón que dispone y la inserción del necesario PIN. De la misma forma que la anterior, no requiere de baterías ni de conectividad a la red para realizar la autentificación y es fácil de usar y de integrar. Según diversos estudios realizados por Google, afirman que el uso de este tipo de criptotarjetas reducen los problemas de autentificación en un 92%. Estudio realizado con empleados de la misma empresa y más de 70 estados diferentes. Al igual que la anterior tiene habilitado el mecanismo NFC y permite la gestión de identidades y gestión de acceso en los siguientes sistemas de gestión:

- AWS Identity and Access Management (IAM).
- Centrify.
- Duo Security.
- Google Cloud Identity.
- Microsoft Azure AD.
- Okta.
- Ping Identity.
- Idaptive.
- Microsoft Active Directory.

A diferencia del anterior, un abanico de posibilidades ligeramente superior [27]. En lo que respecta a su uso en entornos de productividad y comunicaciones, son los mismos que la llave anterior y en cuanto a los gestores de contraseñas, el abanico se amplía con respecto del anterior, ya que además de los comentados para la otra llave, esta también sirve para LastPass Premium.

Respecto a las funcionalidades, el abanico se amplía en comparación con las funciones de la anterior:

- WebAuthn.
- FIDO2 CTAP1.
- FIDO2 CTAP2.
- Universal 2nd Factor (U2F).

- Smart card (PIV-compatible).
- Yubico OTP.
- OATH – HOTP (Event).
- OATH – TOTP (Time).
- Open PGP.
- Secure Static Password.

Dispone de los mismos certificados que la anterior, pero las especificaciones criptográficas, las posibilidades aumentan:

- RSA 2048.
- RSA 4096 (PGP).
- ECC p256.
- ECC p384.

Onlykey

De la misma forma que las anteriores, ofrece un soporte universal para los distintos sistemas operativos [28]: Windows, Mac OS, Android, Linux y Chrome OS. Es decir, para cualquier dispositivo que se conecte será como un teclado de ordenador. El código de Pin de protección del que dispone y que es necesario configurar, hace que en caso de perdida la llave quede inutilizada.

A diferencia de las anteriores, dispone de un doble uso, según lo que convenga. Se puede usar como gestor de claves o como token de segundo factor. Puede albergar hasta 24 cuentas únicas de manera offline, y como U2F, dispone de un número ilimitado de cuentas. Soporta Google Authenticator, OTP compatible con Yubikey y, como se ha dicho previamente, U2F. Se puede utilizar además en comunicaciones SSH, como gestor de contraseña del usuario final, tiene soporte para OpenPGP. A continuación, se mencionan características adicionales que soporta:

- Opción a borrado mediante Pin, como llama el proveedor, “Pin autodestructivo”.
- Característica de negación plausible. Es decir, tiene la posibilidad de albergar las claves que se requieran, sin que aparentemente estén ahí guardadas.
- Tiene la posibilidad de realizar un backup de la configuración de la llave, y además encriptado.
- Bloqueo automático configurable.
- Una vez insertado el Pin de seguridad, aunque un atacante pueda acceder a la misma, sin el código correcto no podrá acceder a los recursos requeridos.
- Dispone de múltiples Keyboard layouts, que se pueden cambiar al instante.

- La velocidad de escritura por pantalla se puede modificar.

Solución adoptada

Tras un análisis de las llaves seleccionadas, se ha llegado a la conclusión de que la que más se asemeja a la que se dispone para el proyecto es la Yubico Yubikey 5 NFC USB A. En un principio es la siguiente serie de Yubikeys y la más versátil para los objetivos del proyecto. Si bien es cierto que la Onlykey ofrecería más opciones, con la yubico yubikey es suficiente para lo que el proyecto concierne. Los diferentes algoritmos criptográficos que soporta, la gran versatilidad de uso del que dispone para los diferentes gestores de identidades, como pueden ser Microsoft Active Directory, Google Accounts; Además, Citrix apps and Desktops lo que utiliza para sus servicios es Microsoft Active Directory, hace que también cumpla con este requisito y uno de los objetivos del proyecto.

En cuanto a su facilidad de uso, todas las comentadas son fáciles de usar por usuarios finales. Quizá la seleccionada sea algo más fácil que la Onlykey puesto que no se tiene porque usar con la configuración de los Pines de gestión y autodestrucción. Es decir, no es una gestora de contraseñas, o no puede albergar tantas contraseñas como la Onlykey. A continuación, se expone una tabla con los criterios utilizados para la selección final la comentada cripto tarjeta, Yubikey Yubikey 5 NFC USB A:

	Versatilidad	Características extra	Implementación con Gestores de Identidad	Facilidad de uso
OnlyKey	✗	✓	✓	✗
Yubico Yubikey 5 NFC USB A	✓	✗	✓	✓

Tabla 2 : Criterios de selección yubikey

A parte de la comentada Yubikey, en el proyecto se han utilizado diferentes elementos que han sido usados tras un análisis previo. Por lo tanto, se comentará las decisiones tomadas a continuación:

- El sistema operativo utilizado en el nodo Edge, como es una Raspberry Pi, es una de las distros ofrecidas por los sistemas operativos para este tipo de elementos. Entre el abanico de posibilidades que había se optará por la siguiente distro por la compatibilidad que había para lo propuesto por el proyecto, es decir, tener instalado o que trajera de fabrica el Linux Kernel 5.0 o posteriores, que también tuviera una versión cryptsetup superior a la 2.6 y que además

soportara la implementación de AES por software. La distro en cuestión se llama: Raspbian GNU/Linux 11 (Bullseye). Además, es una distro que permite la utilización del módulo PAM, módulo que posteriormente se hablará sobre él.

- Para el cifrado de la partición root del nodo Edge, se usará un cifrado diseñado para utilizarse en las raspberrys, como es el aes-adiantum. Este tipo de cifrado en comparación con el de aes-xts ofrece una mayor velocidad de encriptación, velocidad que se muestra más adelanta en la descripción de lo realizado.
- Los sistemas operativos disponibles para el entorno de Citrix, el controlador de dominio desplegado y el servidor que alberga la solución de Citrix, utilizan un Windows Server 2022, versión Datacenter. Sistema operativo que permite una completa integración de los servicios como Active Directory, el servicio de Certificadora y el programa de Citrix.
- Para los clientes utilizados, se usa un Windows 10 (se ha descartado utilizar un Windows 11 por licenciamiento) y por si la versión usada de Citrix no soportará dicho Sistema Operativo. En cambio, para Linux se ha utilizado un Ubuntu 22.04, versión que garantiza tener la versión requerida de SSH que soporta la integración con Yubikey. Aunque versiones anteriores pueden ser compatibles, su prueba queda fuera del alcance de este proyecto.
- Si bien es cierto que hay alternativas a Citrix, como pueden ser Dropbox Business, descartada por el coste de licencia o VMWare Workspace One, descartada también porque no tiene integración con los Microsoft Endpoint Manager, además de tener que configurar una aplicación puente para poder cortar, copiar y pegar contenido entre diferentes aplicaciones, lo cual hace que además de ser más laborioso de configurar, en un futuro se tengan que mantener ambas versiones de las aplicaciones mientras que Citrix, no. Además, grandes empresas se han decantado por esta opción, la de Citrix, lo cual hace que el cliente objetivo a poder albergar la integración con Yubikey, sea considerada a tener en cuenta.
- Como se ha optado por el uso de Citrix, los agentes necesarios a instalar en los clientes finales permiten usar versiones no tan recientes y aun así ser compatibles con la solución propuesta en este proyecto. No obstante, se ha optado por las versiones más recientes de uso, el Citrix Receiver y el Citrix Workspace para Ubuntu. Este último necesario puesto que versiones anteriores tal y como comenta el propio fabricante, hace que pueda romper el sistema de arranque del propio ordenador. Bug anunciado por el propio fabricante.
- Los navegadores usados para la prueba del proyecto, entre todos los que recomienda el fabricante, se ha utilizado Chrome, Firefox y Opera. Seleccionados de esta manera para

demostrar la compatibilidad de la solución con distintos clientes finales.

- Como hypervisor para este proyecto se ha utilizado Oracle Virtualbox, puesto que no requiere de licencia y además permite el despliegue de máquinas de manera rápida y ágil de gestionar. En el mercado existen opciones más completas como por ejemplo VMWare Datacenter, o incluso utilizar proveedores de servicios cloud como pueden ser Azure o Amazon Web Services, pero vistos los costes que esto supondría para esta prueba de concepto, han quedado descartadas. No obstante, la integración del entorno de Citrix con ellos, no sería mucho más compleja, pudiendo realizar la correspondiente configuración, generar una imagen y replicarla en el entorno de destino.

Análisis de riesgos

En este apartado se comentarán los riesgos asociados al proyecto y las acciones a realizar en caso de pérdida o una configuración errónea de la yubikey. Por lo tanto:

Entre los riesgos asociados al proyecto, estarían por un lado el retraso en la entrega con la configuración completa y final de la yubikey, por problemas técnicos: una mala configuración en scripts de arranque, una configuración errónea en la propia yubikey, que el diseño propuesto no cumpla con los requerimientos de ciberseguridad de la organización y haya que añadir alguna otra configuración, etc. Puede suceder también que durante la ejecución del proyecto cualquiera de las partes tecnológicas utilizadas en el mismo tenga un bug en la versión utilizada y requiera de actualización a una versión superior. Y finalmente, que las fechas de entrega por indisponibilidad de los participantes del proyecto se tenga que retrasar. Para mitigar estas situaciones, se ha dispuesto de lo siguiente:

- En caso de no cumplimiento con los requerimientos de ciberseguridad, revisión quincenal del estado del proyecto y puesta en común con el equipo de ciberseguridad de la organización.
- En caso de necesidad de actualización de versión a una más reciente, se presupuesta una bolsa de horas extra para ello con la correspondiente ventana de ejecución.
- Finalmente, en caso de indisponibilidad de uno de los participantes por el motivo que sea, se doblarán los recursos humanos utilizados y se pondrá en común el estado del proyecto cada dos semanas. Así mismo, habrá un traspaso del conocimiento continuo a todos los participantes del proyecto, ya sean internos o externos.

Por otro lado, y en caso de pérdida o un a configuración errónea de la yubikey o de la raspberry, se tomarán las siguientes acciones:

Se tienen que tener en cuenta como se realiza la encriptación de la partición root de la Raspberry, porque de realizarse una configuración errónea, esta partición quedaría inutilizada. Por ello conviene realizar un backup de la misma previa a su encriptación. La encriptación mediante LUKS, permite también realizar un backup de los headers de la misma, por lo tanto, previa a su configuración también sería altamente recomendable realizar el correspondiente backup.

En cuanto a la escalada de privilegios a usuario root, de realizarse mal la configuración, se podria quedar sin disponibilidad de volver a ser root, por lo que conviene tener una ventana de la terminal en modo root y otra ventana de la terminal como usuario normal.

Por otro lado, quedan los riesgos por uso de este tipo de dispositivos. Es decir, en caso de perder la yubikey, se quedaría sin poder realizar el arranque de la raspberry, y por lo tanto se perdería la información que se tuviera por estar encriptada. De la misma forma, los usuarios que la pierdan tampoco podrán autentificarse en las conexiones SSH al servidor y no tendrían acceso a él.

Y por último, queda destacar que una mala configuración del uso de cripto tarjetas en entornos de trabajo como Citrix, de realizarse la migración al uso de estos dispositivos, podría dejar en el entorno de producción sin operabilidad, con lo que supondría en cuanto a costes para la empresa el hecho de no poder acceder a los mismos. Por lo que se recomienda una réplica del entorno de trabajo, realizar la configuración y correspondientes pruebas en él y si todo es correcto, aplicarlo al entorno de producción.

Diseño de alto nivel

En este apartado se expondrá el diagrama de alto nivel de las dos soluciones propuestas en este proyecto. Por un lado, se mostrará el diagrama en el que participan la yubikey, la raspberry y el ordenador físico externo utilizado. Y por otro lado se mostrará el diagrama de alto nivel del entorno de Citrix. Por lo tanto,

Conexión desde dispositivo Edge a Raspberry Pi mediante el uso de la Yubikey y utilizando el protocolo SSH

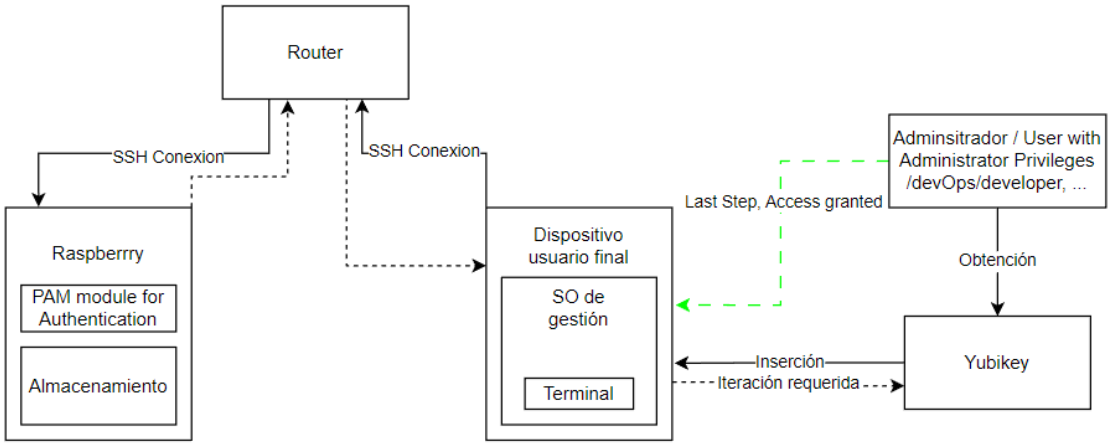


Figura 12: Diagrama de alto nivel. Conectividad por SSH a Raspberry Pi

En este diagrama participan como ya se han comentado la raspberry, el ordenador físico (portátil) y los usuarios finales. En esta primera forma de conectividad a la rasberry, se realiza a través del protocolo SSH, queriendo así simular que el usuario puede estar en cualquier zona de la oficina y puede conectarse a través de dicho protocolo. En un principio, ambos usuarios deben de tener conectividad, estar en una subred de confianza (a ser posible no en una DMZ) y tener los correspondientes usuarios creados, con los permisos adecuados en la raspberry. Se recuerda aquí también que la obtención de privilegios root solo está permitida por diseño si la yubikey está conectada directamente al servidor. En caso contrario, no es posible y habría que configurar de manera adecuada los correspondientes privilegios que se conecten de esta forma. Es decir, diferenciar claramente si son administradores de la raspberry o no.

Acceso securizado con inserción directa de la Yubikey a la Raspberry y su posterior administración

En este escenario, el correspondiente diagrama es el siguiente:

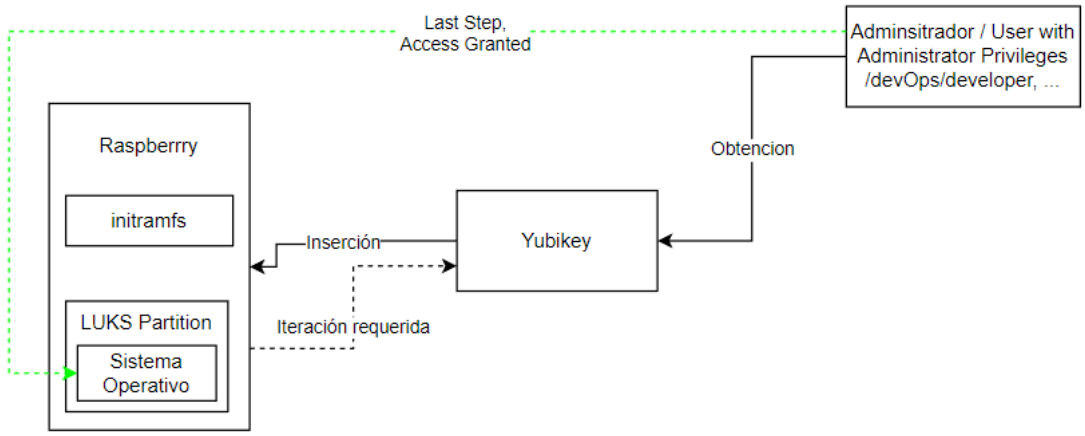


Figura 13: Diagrama de alto nivel. Administración de manera directa a la Raspberry Pi

Este escenario sería válido única y exclusivamente para el usuario administrador y el encargado de desplegar el correspondiente servicio en el nodo Edge (raspberry). Por lo tanto, tiene que ser un administrador con acceso permitido al CPD y con los suficientes privilegios para poder realizar el correspondiente despliegue del servicio orquestado. Con esta conexión sí que se puede obtener permisos root, puesto que se da por hecho que el único responsable que acceda a la raspberry es el administrador u operario encargado del mantenimiento de la servidor.

Dicho lo cual, y tras haber expuesto los diagramas de alto nivel de esta solución, más adelante se expondrá la configuración realizada para las mismas.

Entorno virtualizado de aplicaciones Citrix y autentificación de acceso al mismo mediante la criptotarjeta Yukikey

Por otro lado, se tiene la solución propuesta para el entorno de Citrix. Este entorno es más complejo, puesto que requiere de más elementos y una más compleja configuración que, de la misma forma que los escenarios anteriores, se expondrá más adelante. Por lo tanto, el diagrama para dicho entorno es el siguiente:

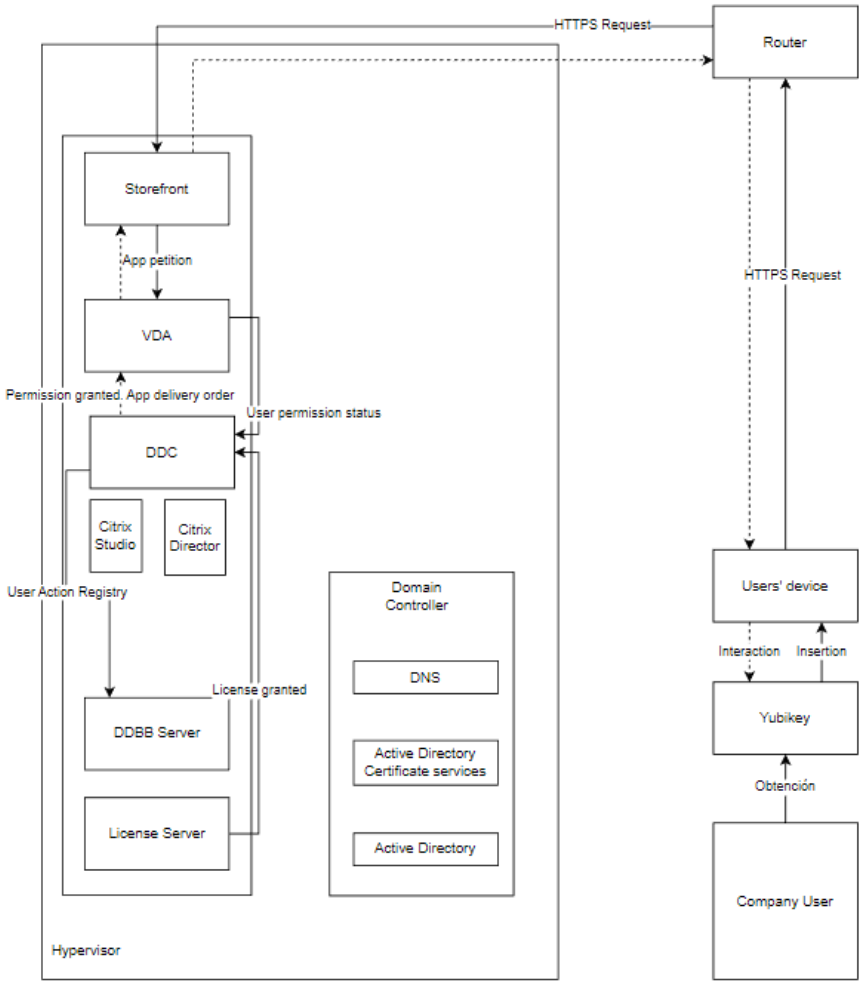


Figura 14: Diagrama de alto nivel. Entorno de Citrix

Por esclarecer las abreviaturas de los elementos que aparecen en el diagrama anterior, a continuación, se hará una breve descripción de los mismos [29]:

- **DNS**: Domain Name Server. Es el sistema de nombres de dominio que se utilizara para traducir los nombres de dominios aptos para lectura humana, en este caso, los nombres de los servidores a IPs.
    - Dominion utilizado en el entorno: citrix.com
    - Nombres de los servidores:
        - DCparaCitrix.citrix.com → Domain Controller

- maqCitrix.citrix.com → Servidor con elementos de Citrix (192.168.1.102/24)

- **Lic. Server:** Servidor de licencias que alberga tanto las licencias RDP (Remote Desktop Protocol) como las propias licencias de Citrix. En este caso, es una característica que se ha instalado en el mismo servidor de Citrix.

- **DDBB:** Servidor de bases de datos. Base de datos necesaria para el registro de los correspondientes logs de Citrix.

- **VDA:** Virtual Delivery Agent. Instalada en el servidor que está disponible para los usuarios dentro del entorno propuesto. Es la máquina que entrega las aplicaciones que se publican a los usuarios de la organización. Por otro lado, se encarga de la gestión de las sesiones entre usuarios y el correspondiente servidor. Además, se encarga de verificar que haya disponible una licencia desocupada para la sesión del usuario que la necesite y se encarga también de aplicar las correspondientes políticas a dicha sesión. Del mismo modo, comunica la información del estado de la sesión mediante el servicio Citrix bróker al DDC. Este agente tiene diferentes plug-ins y recolecta información en tiempo real sobre la sesión.

  Utilizando el puerto TCP 80, puede tener dos tipos de sesión: mono sesión, en la que un solo usuario se conecta al servidor y no permite que otros se conecten al mismo, o multi sesión, que permite conectarse a más de un usuario a la vez.

- **DDC:** Delivery Controller. Es el elemento principal de gestión para el site. Cada site tiene uno o más de un delivery controller (en este caso uno) que debe estar instalado en el "datacenter" de la organización. En este caso no se dispone de más de uno puesto que es un entorno de pruebas y no se disponen de recursos hardware suficientes para tener dos y así mejorar la disponibilidad de la solución. No obstante, se recomienda la instalación de un mínimo de dos. Como es el caso, si el despliegue incluye un hypervisor el controller se comunica con él para:

  - Distribuir aplicaciones y escritorios virtualizados.
  - Autenticación y acceso de usuarios.
  - Conectar o realizar las conexiones entre usuarios y las aplicaciones o escritorios virtuales que requieran.
  - Optimizar las conexiones de usuarios
  - Calibrar (Load Balance) dichas conexiones.

Por otro lado, el Broker service agregado al controller, analiza qué usuarios se han registrado y dónde, que recursos utiliza su sesión y si el usuario necesita reconectarse a la aplicación. Finalmente, se encarga además de las siguientes acciones:

- Su "Monitor Service" recolecta los datos históricos y los ubica en la base de datos de monitorización. Este servicio utiliza los puertos 80 y 443.
- Los datos que utiliza o recoge los servicios del controller se almacenan en la base de datos del site.
- Como ya se ha comentado anteriormente, se encarga de inicializar las sesiones y para en las conexiones de los escritorios virtualizados, parándolos o arrancándolos en función de la demanda y la configuración de administración. De hecho, en algunas versiones, que en este entorno no es el caso porque requiere de un cargo extra, te permite tener un control de perfiles para personalizar los escritorios virtualizados en función de lo que los usuarios necesiten, y así no perder el estado de la sesión una vez finalizada.

Cuando se exponga la configuración de cada uno de los elementos se comentarán más específicamente el funcionamiento de los mismos, por si llegados a este punto, quedaran dudas al respecto.

En los escenarios vistos hasta ahora, el usuario realizaría los siguientes pasos, explicados brevemente:

- En el primer escenario, el usuario inicializa una sesión mediante SSH para conectarse a la raspberry. Al inicio de esta, la sesión le pedirá que inserta la yubikey de manera obligatoria para poder autenticarse ante la raspberry. El usuario presiona el botón de la yubikey cuando esta empiece a parpadear y posteriormente comenzara la sesión SSH.
- En el segundo, el administrador de la raspberry tendrá insertada la yubikey en la raspberry. Encenderá la raspberry, porque la información de la misma estará encriptada y le pedirá la yubikey para poder arrancarla. De lo contrario, ni si quiera le dejara arrancar el proceso de arranque porque lo detendría. Una vez que esta arrancada la raspberry, siempre que necesite realizar la escalada de privilegios a usuario root, tendrá que usarla porque la raspberry se la pedirá.

- En el tercero, el usuario de la organización arrancara el Citrix Receiver o el Citrix Workspace, en su defecto que son el mismo programa solo que el Citrix Workspace es un cambio de nombre que se le dio al producto (utilizando versiones más recientes). Al inicializar el programa, este le pedirá que inserte la yubikey para poder inicializar la sesión. De no tenerla, se la pedirá hasta que la inserte, y una vez que la tenga instalada, le pedirá el código pin y el usuario para la correspondiente autentificación. Una vez dentro, solo tendrá que elegir la aplicación que desea arrancar y que este desplegada por el controller. Cabe destacar que la conectividad de este escenario es más compleja que lo comentado aquí, y que se desarrollara de manera más profunda en apartados siguientes.

Descripción de la solución propuesta. Descripción de la configuración

Las partes configuradas en este proyecto son 3: La configuración de la raspberry para que al inicio del arranque pida la yubikey. La segunda es la configuración de la propia yubikey para los diferentes casos de uso en los entornos propuestos en el proyecto. Y finalmente la configuración del entorno de Citrix para la autentificación en el mismo mediante la yubikey. La configuración de dichas partes se comentará a continuación.

Configuración de la raspberry

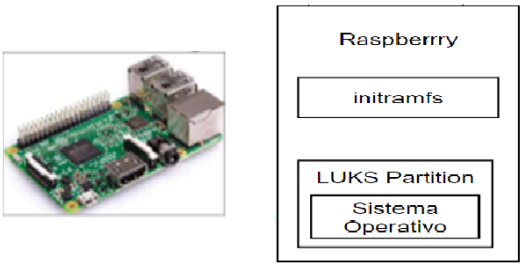


Figura 15: Raspberry. Configuración de las diferentes partes participantes

Para el arranque, hay que tocar dos veces el botón de la yubikey. Una para autenticarse con la yubikey y la segunda para poder autenticarse y desencriptar la partición encriptada

Antes de la configuración sobre el arranque, hay que configurar el propio sistema operativo y la micro tarjeta USB en donde se desplegara el mismo. El sistema operativo utilizado es Raspbian GNU/Linux 11 (bullseye).

Para evitar la interacción de cualquier archivo, sistema operativo previo o que el propio volumen de la tarjeta SSD utilizada este corrupto, se ha formateado y se ha configurado de manera que la tarjeta se convierta en una tarjeta bootable. Que sea bootable significa que desde la misma se pueda arrancar el sistema operativo comentado previamente y así dejarlo instalado para su posterior uso. Para el formateado de la tarjeta SSD, se ha utilizado las herramientas que ofrece Windows 10 y la herramienta open-source Rufus. Esta última hace que la imagen instalada en la SSD, se configure de manera automática para convertirse en bootable. Para ello, estos han sido los pasos realizados:

1. Se inserta la tarjeta SSD con un adaptador de micro-tarjetas al ordenador. Una vez que el ordenador detecte lo insertado, se abre la terminal cmd de Windows y se inserta lo siguiente:

   diskpart.exe

   Y se obtiene la siguiente ventana:

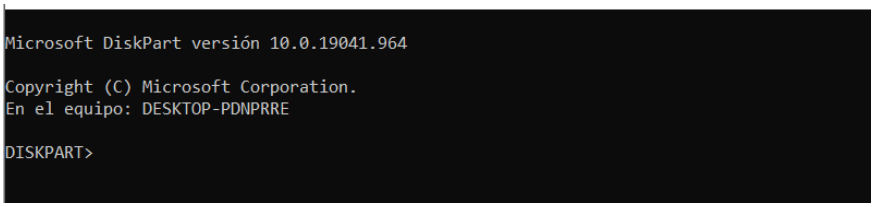


Figura 16: Output de comando por terminal Windows cmd

diskpart es una utilidad de Windows, concretamente la sucesora de fdisk, que permite realizar particiones a volúmenes mediante comandos por terminal, ya sean en USB, discos físicos, etc.

2. Como el ordenador ha detectado previamente nuestra tarjeta, se escribirá el siguiente comando por terminal para verificar que la correspondiente tarjeta esta insertada:

   list disk

   Un output de ejemplo sería el siguiente:

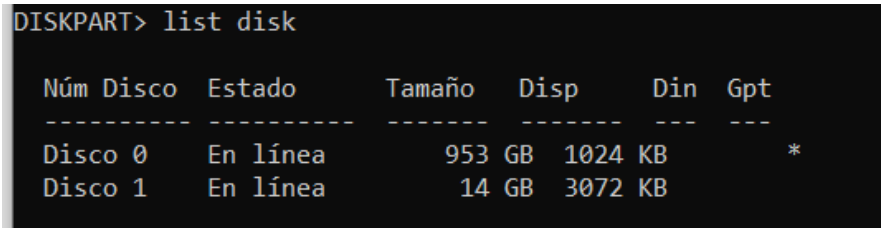


Figura 17: Lista de discos insertados al host anfitrión

Donde sí se sabe el tamaño de la SSD, que en el caso de este proyecto es de 16 GBs, se sabe que el Disco 1, es el concierne a la solución

3. Se selecciona mediante el siguiente comando:

   select disk 1

4. Se formatea insertando el siguiente comando:

   clean

5. En este punto el disco está completamente en blanco. Por lo que se creará una partición, se seleccionará y se formateará de manera que la imagen del SO comentado previamente se pueda instalar mediante el programa de Rufus. Para ello, se escriben los siguientes comandos en la misma terminal:

   create partition primary

   select partition 1

   format quick

6. En este punto, la tarjeta SSD está lista para usarse con el programa Rufus.

Comentario extra (1): De la misma forma que diskpart, Rufus es una utilidad que le ayuda a formatear y crear soportes USB de arranque, como «pendrives», tarjetas de memoria, etc. Es especialmente útil en casos donde necesite crear medios de instalación USB a partir de ISOs arrancables (Windows, Linux, UEFI, etc.), en este caso una la imagen comentada previamente para la raspberry pi; donde se necesite trabajar en un equipo que no tenga un sistema operativo instalado; donde se necesite actualizar el firmware o BIOS de un ordenador desde DOS o se quiera ejecutar una utilidad de bajo nivel.

El programa Rufus, tiene la siguiente interfaz de usuario:

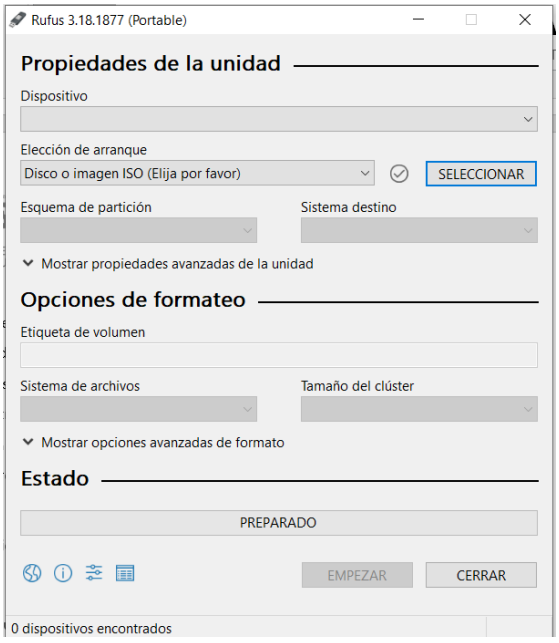


Figura 18: Consola rufus para creación bootable USB

Donde el dispositivo será la tarjeta SSD previamente lista para usarse y la imagen que se seleccione será la descargada de la página web oficial de raspberry pi:

https://www.raspberrypi.com/software/operating-systems/

Y una vez cargada, se le dará al botón de "Empezar". Tras finalizar el proceso, se sacará la tarjeta micro-SSD del adaptador, y se insertará en la raspberry.

El inicio y configuración de arranque de la raspberry es bastante intuitiva, incluso más rápido y sencillo que los sistemas operativos Ubuntu o Windows. Se seleccionará la red de Internet, se definirá el usuario con el que se comenzara y se realizaran las siguientes configuraciones, el idioma del sistema operativo y el idioma del teclado. Por lo que la raspberry en este punto queda lista para realizarse las siguientes configuraciones.

Encriptación de la partición root de la tarjeta SSD

Para la encriptación de la comentada tarjeta, se utilizará la especificación llamada LUKS. Estas siglas son Linux Unified Key Setup. LUKS especifica un formato estándar en disco, independiente de la plataforma donde se ejecute para usar en varias herramientas. El hecho de que sea independiente de la plataforma hace que facilite la interoperabilidad y la compatibilidad entre los diferentes programas lo que garantiza que la gestión de contraseñas se haga de manera segura y de una manera ordenada.

La implementación de referencia funciona en Linux, en este caso en el sistema raspbian comentado previamente y dicha especificación se basa en una versión mejorada de cryptsetup, utilizando dm-crypt como la interfaz de cifrado de disco. En Microsoft Windows, los discos cifrados con LUKS pueden ser utilizados con FreeOTFE, por ejemplo.

Usado para encriptar "block devices", los cuales proveen de un acceso en modo buffer a los elementos hardware, también pueden encriptar sistemas de ficheros, incluida la partición swap.

Comentario extra (2): La partición swap o partición de intercambio es una partición de disco estándar que se designa como espacio de intercambio por el comando mkswap. Este espacio se puede usar para crear una nueva partición en el disco cuando este está lleno. Más concretamente, se utiliza para cuando la memoria RAM del dispositivo está completamente llena y aun así se necesita de más memoria para ejecutar los procesos previstos

LUKS, por otro lado, soporta diferentes combinaciones de algoritmos de encriptación, modos de encriptación diferentes y funciones hash, entre los que se pueden incluir los siguientes:

- **AES**: Advanced Encryption Standar, que utiliza una clave de longitud variada y se denomina así en función de la longitud de la misma. AES-128, AES-192 o AES-256 [30].
- **Serpent**: Cifrador de bloques de 128 bits, fue finalista como candidato a en la competición de AES [31].
- **Twofish**: Cifrador de bloques con claves simétricas tiene una longitud variable, desde los 128 bits a 256 bits. Open-source y con diferentes capacidades y funcionalidades para reemplazar el algoritmo estándar DES [32].
- **CAST-128**: Alternativamente llamado CAST5, es un cifrador de bloques de clave simétrica, usado principalmente en diferentes versiones GPG y PGP [33].
- **CAST-256**: Cifrador de bloques con clave simétricas, no estuvo seleccionado ni si quiera entre los 5 candidatos finalistas para AES [34].

Los modos de encriptación que utiliza, por otra parte, son los siguientes:

- **ECB**: Electronic Code Book. Un modo de operación simple con un cifrado de bloque usa, mayoritariamente un cifrado de clave simétrica. En cierta medida, una manera de procesar una serie de bloques de mensajes enumerado secuencialmente [35].
- **CBC-PLAIN64**: Cipher Block Chaining. En este modo, una secuencia de bits se cifra como una sola unidad o bloque, con una clave de cifrado aplicada a todo el bloque. El encadenamiento de bloques se utiliza un vector de inicialización de cierta longitud. Al usar esto junto con una sola clave de cifrado, se puede cifrar y descifrar de forma segura grandes cantidades de texto sin formato [36].
- **CBC-ESSIV**: Mismo que el anterior, especializado para dispositivos de poca capacidad y cantidades pequeñas de datos [37].
- **XTS-PLAIN64**: XEX Tweakable Block Ciphertext Stealing. Puede cifrar o descifrar secuencias de longitud variadas En este modo los datos de salida y entrada pueden consistir en una serie de bloques de 128 bits seguidos de un bloque parcial separado que no está vacío y es inferior a 128 bits. Utilizando este método, se requieren dos claves de cifrado con 256 bits de longitud; Se denominan clave modificable y clave de cifrado, respectivamente [38].

Y entre las diferentes funciones hash que utiliza, se pueden encontrar las siguientes:

- **SHA-1, SHA-256, SHA-512**: Secure Hashing algorithm. Es una versión modificada de MD5 y se

usa para cifrar datos y certificados. Un algoritmo hash acorta los datos de entrada a una forma más pequeña que no se puede entender, usando operaciones bit a bit, adiciones modulares y funciones de compresión. Funciona de tal manera que incluso si cambia un único carácter del mensaje, se generara un hash diferente al anterior. Las diferentes -256 o -512, hacen referencia a la longitud del hash usado. Ambos se basan en la versión SHA-2, la cual mejora la versión inicial SHA-1 y es mayoritariamente usado para la generación de certificados [39].

- **RIPEMD160**: RACE Integrity Primitives Evaluation Message Digest. Grupo de funciones hash, más débiles que los comentados anteriormente. En términos generales no está catalogado como buena función hash. De hecho, en el proyecto no se ha usado, pero se menciona puesto que LUKS da la posibilidad también de utilizar dichas funciones [40].

Para finalizar con LUKS, las capacidades que tiene y las diferentes características, por defecto utiliza de todo lo comentado hasta ahora, lo siguiente:

- AES como algoritmo de encriptación
- CBC, como modo de encriptación.
- ESSIV como vector usado y basado en CBC
- SHA-256 como función hash.

Una vez comentado esto, se comenzará a describir los pasos seguidos para la encriptación de la partición root y posterior desencriptación utilizando la yubikey. Cabe mencionar que, sin la yubikey, no es posible acceder ni si quiera al sistema operativo de la raspberry, puesto que ni si quiera comenzaría el sistema de arranque de la misma. Por lo tanto:

1. Para empezar y realizar la configuración propuesta en el proyecto, se tiene que verificar que la versión de Linux kernel que utiliza la raspberry es una versión 5.0 o superior. Para ello, se abre una terminal y se escribe el siguiente comando:

   uname -s -r

   Esto dará como resultado el siguiente output en la terminal:

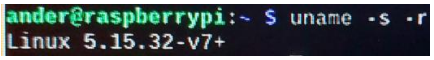


Figura 19: Output de comando por terminal Raspbian. Versión del kernel

2. De la misma forma, hay que verificar que se tenga una versión de cryptsetup igual o superior a la versión 2.0.6:

cryptsetup --version

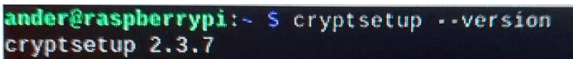


Figura 20: Version programa cryptsetup

Comentario extra (3): cryptsetup proporciona una interfaz para configurar el cifrado en dispositivos de bloque, al igual que directorio, como por ejemplo /home o particiones de intercambio, como se ha mencionado antes, particiones swap. Para ello utiliza el mapeador de dispositivos del kernel de Linux, dm-crypt. Dm-crypt es un sistema de cifrado para archivos y particiones que se utiliza para proteger los datos almacenados en un disco duro. El nombre dm-crypt se deriva de "Device Mapper" y "Crypt". Se implementa como un kernel de Linux, lo que significa que se ejecuta directamente en el núcleo del sistema operativo. DM-Crypt se basa en el estándar de cifrado establecido por el Data Ecnryption Protocol (DEP o DCP, en castellano) de la computación segura. Esto permite que los usuarios cifren y descifren sus datos de forma segura y eficiente. Está diseñado para ser compatible con los sistemas de archivos existentes de Linux, como ext2, ext3, ext4 y NTFS. Para usarse, primero se crea una partición cifrada. Esto se hace utilizando una herramienta especial como la comentada previamente. Esto crea un dispositivo de bloque de cifrado que se utiliza para cifrar la partición. Cuando se cifra una partición, todos los datos almacenados en ella se cifran automáticamente [41].

Una vez cifrada, la partición no se puede acceder sin la contraseña o dispositivo de autentificación. Esta contraseña se utiliza para cifrar y descifrar los datos almacenados en la partición. Por lo que, sin ella, los datos son inaccesibles. De la misma forma, se podría comentar que dm-Crypt es una herramienta poderosa para la protección de los datos almacenados en discos duro. Es rápido y fácil de usar, y es una solución viable y asequible para el cifrado de datos.

Dejando claro el termino, cryptsetup es compatible con LUKS al igual que es retro compatible con el formato en disco de cryptoloop, admitiendo además formatos más seguros. Este paquete incluye además lo necesario para configurar automáticamente dispositivos cifrados en el momento del arranque a través del archivo de configuración /etc/crypttab. Características adicionales son la compatibilidad con cryptoroot a través de initramfs-tools y varias formas compatibles de leer una frase de contraseña o clave. Cuando se interactúe con las herramientas initramfs, se explicará qué es.

3. Hay que instalar además todos los programas necesarios y que crean las dependencias necesarias para el correcto funcionamiento de la configuración. Por lo tanto, se escribirá por la terminal lo siguiente [42]:

   sudo apt-get install busybox cryptsetup initramfs-tools

   **cryptsetup –version de nuevo, si el comando del paso anterior no ha dado por terminal la versión que se está usando del mismo

En este punto se comentarán qué son y para qué sirven los programas instalados tras este comando:

a. **Busybox**: Es una colección de utilidades en un solo binario. De código abierto y con licencia GPL, implementa una gama de comandos extra como "applets" que combinan con la distribución de este programa. Usado también para administrar un sistema POSIX. Capaz de arrancar el proceso con el ID 1, y soportar de manera simple los servicios del sistema. Puede adoptarse este programa como recambio a los demonios init o systemd. Finalmente, una herramienta que combinándola con el kernel de Linux se puede crear un sistema sin ninguna otra dependencia [43].
b. **Initramfs-tools**: paquete que contiene herramientas para crear un initramfs para imágenes de Linux 2.6 preempaquetadas. El initramfs es un archivo cpio (un archivador de ficheros general, normalmente instala en sistemas operacionales Unix). En el momento de arranque el núcleo descomprime el archivo cpio en la RAM, lo que monta y utiliza como sistema de archivos raíz inicial. A partir de ahí, el montaje del sistema de archivos raíz real se produce en el espacio del usuario. Despues, klibc (subconjunto de la librería estándar C) se encarga de la configuración de red en el momento de arranque. Soporta además sistema raíz nfs y además, cualquier sistema de arranque compatible con initrd puede cargar un archivo como initramfs [44].

4. Una vez instalado estos programas necesarios, cabe comentar que los micro procesos de ordenadores como Raspberry Pis, no llevan incluida la aceleración AES. Por lo que dicha implementación de software es necesaria para su implementación, aunque lenta para este tipo de procesadores. La solución es la implementación de software Adiantum, que es capaz de procesar más rápido en software. La versión mínima requerida por el kernel, incluye módulos de kernel necesarios para adiantum. Se verifica que dichos módulos están presentes mediante este comando:

cryptsetup benchmark -c xchacha20, aes-adiantum-plain64

Si están, aparecerá un output por la terminal de este tipo:

```
# Las pruebas son solo aproximadas usando memoria (no hay entrada/salida de alma
cenadmiento).
#            Algoritmo |    Clave |      Cifrado |    Descifrado
xchacha20,aes-adiantum        256b      30,6 MiB/s      28,5 MiB/s
```

Figura 21: Características sobre los tipos de encriptación disponibles

En caso de que no estén o la respuesta del comando previo de algún error, se instalan de la siguiente manera:

sudo modprobe xchacha20

sudo modprobe adiantum

sudo modprobe nhpoly1305

5. En este punto se recreará initramfs, puesto que cuando se crea un nuevo kernel, dicha "recreación" es necesaria. Para ello se creara el siguiente fichero, ***/etc/kernel/postinst.d/initramfs-rebuild***, que contendrá el siguiente código, que se obtendrá

del siguiente repositorio de github ***https://github.com/Robpol86/robpol86.com/blob/master/docs/_static/initramfs-rebuild.sh***:

```
#!/bin/sh -e
```

 # Remove splash from cmdline. → Las siguientes líneas harán que no se vea la imagen siguiente en el arranque de la raspberry:

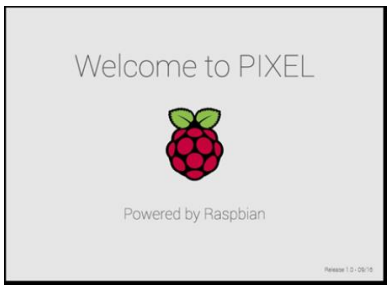


Figura 22: Imagen raspbian al encender la Rasberry con dicho Sistema Operativo insertado

```
if grep -q '\bsplash\b' /boot/cmdline.txt; then
  sed -i 's/ \?splash \?/ /' /boot/cmdline.txt
fi
```

 # Exit if not building kernel for this Raspberry Pi's hardware version → Salva guarda en caso de tener la versión del kernel menor a la comentada previamente.

```
version="$1"
current_version="$(uname -r)"
case "${current_version}" in
  *-v7+)
    case "${version}" in
      *-v7+) ;;
      *) exit 0
    esac
  ;;
  *+)
```

```
  case "${version}" in
    *-v7+) exit 0 ;;
  esac
 ;;
esac
```

 # Exit if rebuild cannot be performed or not needed. → Se verifica si hace falta la recreación del archivo initramfs. De necesitarlo se hace y si no, no.

```
[ -x /usr/sbin/mkinitramfs ] || exit 0

[ -f /boot/initramfs.gz ] || exit 0

lsinitramfs /boot/initramfs.gz |grep -q "/$version$" && exit 0  #Already in initramfs.

# Rebuild.

mkinitramfs -o /boot/initramfs.gz "$version"
```

6. Una vez creado el archivo tiene que ser ejecutable. Para ello:
   sudo chmod +x /etc/kernel/postinst.d/initramfs-rebuild
7. Además, se necesitan especificar ciertos programas en el initramfs. Para ello, en el siguiente archivo, ***/etc/initramfs-tools/hooks/luks_hooks*** se añaden las siguientes líneas de código:

```
#!/bin/sh -e
PREREQS=""
case $1 in
      prereqs) echo "${PREREQS}"; exit 0;;
esac

. /usr/share/initramfs-tools/hook-functions

copy_exec /sbin/resize2fs /sbin
copy_exec /sbin/fdisk /sbin
copy_exec /sbin/cryptsetup /sbin
```

Fuera a parte del ya comentado cryptsetup, se añaden los programas resize2fs y fdisk. A continuación, se hace un resumen de ambos:

   a. **resize2fs**: Este programa cambia el tamaño de los sistemas de archivos ext2, ext3 o ext4. Se puede utilizar para ampliar o reducir un sistema de archivos desmontado y ubicado en el dispositivo. Si el sistema de archivo está montado, se puede usar para expandir el tamaño del sistema de archivos montado, suponiendo que el kernel admita el cambio de tamaño en línea. El parámetro size, especifica el nuevo tamaño solicitado del sistema de archivos. Si no se especifican unidades, las unidades del

parámetro de tamaño serán el tamaño de bloque del sistema de archivos. Opcionalmente, el parámetro de tamaño puede tener como sufijo uno de los siguientes designadores de unidades: 's', 'K', 'M' o 'G', para sectores de 512 bytes, kilobytes, megabytes o gigabytes, respectivamente. El tamaño del sistema de archivos nunca puede ser mayor que el tamaño de la partición. Si no se especifica el parámetro de tamaño, se establecerá de forma predeterminada en el tamaño de la partición. Por otro lado, este programa no modifica el tamaño de las particiones. Si se desea ampliar un sistema de archivos, primero debe asegurarse de poder ampliar el tamaño de la partición subyacente. Este se puede hacer usando fdisk y así eliminar la partición creándola con un tamaño más grande. Sirve también para reducir los tamaños de las particiones [45].

   b. **fdisk** (en la primera forma de invocación): es un programa controlado por menú para la creación y manipulación de tablas de partición. Entiende tablas de particiones tipo DOS y etiquetas de disco tipo BSD o SUN. fdisk no entiende la tabla de particiones GUID (GPT) y no está diseñado para particiones grandes. En un caso particular, utilice GNU parted más avanzado. El dispositivo suele ser /dev/sda, /dev/sdb [46].

8. De la misma forma que el anterior archivo, este archivo también tiene que convertirse en ejecutable:

   sudo chmod +x /etc/initramfs-tools/hooks/luks_hooks

9. Por defecto, initramfs para el sistema operativo de la Raspberry PI, no incluye los módulos para LUKS y de encriptación. Por lo que se tienen que configurar en el siguiente archivo, **/etc/initramfs-tools/modules**, donde se añadirán las siguientes líneas que hacen referencia a ellos:

   algif_skcipher
   xchacha20
   adiantum
   aes_arm
   sha256
   nhpoly1305
   dm-crypt

10. En este punto se deberá construir el nuevo initramfs configurado hasta la fecha. Por lo tanto:

    sudo -E CRYPTSETUP=y mkinitramfs -o /boot/initramfs.gz

11. Se verificará que los programas comentados estén en el initramfs mediante el siguiente comando:

    lsinitramfs /boot/initramfs.gz | grep -P "(algif_skcipher|chacha|adiantum|aes-arm|sha256|nhpoly1305|dm-crypt)"

En este punto, y antes de reiniciar la raspberry, hay que configurar diferentes archivos. Estos cambios están destinados a indicarle al proceso de arranque que use un sistema de archivos raíz encriptado. Después de los cambios anteriores, la Raspberry Pi arrancará correctamente. Después de cambiar los siguientes archivos, Raspberry Pi no se iniciará en el escritorio hasta que se complete todo el proceso de encriptación de la partición raíz y configuración de LUKS que se hará más adelante. Por lo tanto, en este punto se ha hecho una copia de respaldo, por si en algún momento hay que revertir los cambios realizados. Tras ello, se realiza lo siguiente:

1. En el archivo ***/boot/config.txt***, se añadirá la siguiente line al final del mismo:

   initramfs initramfs.gz followkernel

   Este archivo contiene parámetros de configuración a cerca de la SD Card. Estos parámetros se leen al arrancar la raspberry PI. Los parámetros o instrucciones declarados aquí, ajustan la forma en cómo se detecta la pantalla, se muestra el escritorio e incluso se puede configurar para realizar overclock en la Raspberry. De hecho, si se realiza esta última acción comentada, y por el motivo que sea no se puede arrancar la Raspberry de manera correcta, mediante este archivo se puede resetear la raspberry. Se puede incluso hasta modificar el arranque por USB o incluso las entradas de audio.

   En el caso de la solución, el comando indicado especifica el archivo ramfs y la dirección de la memoria a la que hace referencia. En este caso, followkernel o 0.

2. El archive ***/boot/cmdline.txt***, contiene ciertas líneas de comando. Una de ellas hace referencia a la ubicación de la partición root. En este caso, la linea:

   root=/dev/mmcblk0p2

   Por la siguiente:

   root=/dev/mapper/sdcard

   Es decir, se especifica en este caso que la partición root está en ese lugar indicado.

   Además de la comentada modificación, al final de la línea y separado por un espacio en blanco, se añadirá lo siguiente:

   cryptdevice=/dev/mmcblk0p2:sdcard

   Donde de esta forma se le habrá especificado al sistema de arranque la partición declarada como tal, estará encriptada.

3. El dispositivo para la partición root, también tiene que ser cambiada para apuntar hacia el mapper especificado. En el archivo ***/etc/fstab*** la siguiente linea:

   /dev/mmcblk0p2

   Es sustituida por la siguiente:

   /dev/mapper/sdcard

   Este archivo es la tabla de sistema del archivo Linux. Esta tabla es una tabla de configuración diseñada para la agilizar la carga de montar y desmontar sistema de archivo en una máquina. Es un conjunto de reglas que se utilizan para controlar como se tratan los diferentes sistemas de archivos cada vez que se introducen en un sistema. Este archivo, está diseñado para configurar reglas que detectan sistemas de archivos específicos y luego tras definirlos se montan automáticamente en el orden deseado por el usuario cada que se inicia el sistema, lo que facilitaría el trabajo y reduciría errores de carga. La tabla contiene 6 columnas, donde cada una especifica un parámetro que tiene que ser configurado en el orden correcto. Las columnas

son las siguientes: device, mount point, tipo de sistema de ficheros, opciones, operación de backup y orden de supervisión del sistema.

4. En el archivo ***/etc/crypttab***, se añade otra línea al final con el siguiente contenido:

   sdcard /dev/mmcblk0p2 none luks

   Este archivo describe los bloques encriptados que son configurados en el sistema de arranque. Donde en la línea descrita previamente, los primeros valores son obligatorios y los otros dos en cambio opcionales. Dichos valores hacen referencia a lo siguiente:

   a. El primero especifica el volumen que se crea tras la desencriptación.
   b. El segundo, especifica la ruta al archivo o dispositivo de bloque subyacente.
   c. El tercero, sería la ruta que apunta a la key de la encriptación. No especificada puesto que no la hay.
   d. El cuarto hace referencia a las opciones utilizadas para la encriptación/desencriptación. En este caso, se especifica que se fuerce el luks mode, con lo que se consigue que se lean las cabeceras LUKS y que se descarten otros parámetros como cipher, hash, size (ya que vienen y se leen en las cabeceras LUKS).

De esta forma, la raspberry esta lista para reiniciarse. Tras el reinicio, de manera deliberada falla, porque no se ha configurado todavía la partición root indicada en la configuración de los archivos previos. Por lo que tras unos cuantos mensajes con error, la initramfs shell aparecerá por pantalla.

En esta shell, se verifica que se pueda utilizar cryptsetup con los modules kernel cargados previamente. El comando es el siguiente:

cryptsetup benchmark -c xchacha20,aes-adiantum-plain64

En este punto, el test es satisfactorio. Por lo que el siguiente paso es copiar la partición root de la SD Card, a una memoria USB externa. En este caso se ha utilizado una memoria de 64 GBs, es decir, lo suficientemente grande como para albergar la comentada copia.

Con esta copia lo que se hace es crear una partición encriptada, con la correspondiente perdida de datos que conlleva y posteriormente volcar los datos del USB a dicha partición. Para reducir los tiempos de ejecución de este copiado y volcado de información desde y hacia el USB, lo que se hará reducir el tamaño del sistema de ficheros al mínimo posible. Para dicha operación de verificación y posible corrección se utiizará el siguiente comando:

e2fsck -f /dev/mmcblk0p2

Tras finalizar dicha operación, se cambiará el tamaño del mismo. Donde se apuntarán el número de bloques de 4k que se han conseguido tras el reajuste:

resize2fs -fM -p /dev/mmcblk0p2

Una vez que el reajuste de tamaño se ha realizado, se obtendrá un checksum de toda la partición. Se realizará lo mismo tras la operación de copia entre la SD card y la memoria USB. Si el checksum es el mismo, la copia es correcta. Para ello se escribira el siguiente comando por terminal donde el parámetro count será el número de bloques de 4k obtenidos tras el reajuste:

time dd bs=4k count=2775095 if=/dev/mmcblk0p2 | sha1sum

Ahora, se conectará el USB para la posterior copia. Se insertará el siguiente comando por terminal:

time dd bs=4k count=2775095 if=/dev/mmcblk0p2 of=/dev/sda

Donde /dev/sda hace referencia al destino de la copia. Se realizará el cálculo del checksum, y se comparan. Si ambos checksum son iguales, se considerarán la misma copia.

time dd bs=4k count=2775095 if=/dev/sda | sha1sum

Por lo que ambos checksums son iguales. Se procede a encriptar la partición root. Para ello, se utilizarán los siguientes parámetros que deja utilizar cryptasetup:

cryptsetup --type luks2 --cipher xchacha20,aes-adiantum-plain64 --hash sha256 --iter-time 5000 –keysize 256 --pbkdf argon2i luksFormat /dev/mmcblk0p2

Dichos parámetros especifican lo siguiente:

- **type**: Especifica la versión de LUKS que se va a utilizar. En este caso, la 2.
- **cipher**: el cifrador que se usara/usaran.
- **hash**: el tipo de hash que se va a usar.
- **iter-time**: Número de milisegundos para pasar con el procesamiento de frase de contraseña PBKDF2
- **keysize**: tamaño de la clave. 256 ya que se usa sha256
- **pbkdf**: la función de derivación utilizada para el cálculo. En este caso, argo2i, ganadora del campeonato de 2015 de competición de contraseña hash.
- **luksformat**: Operación LUKS, que inicializa una partición LUKS y establece las contraseñas iniciales.

El comando pide dos veces que se inserte la contraseña, por lo que es importante que la contraseña sea larga y segura. Esta contraseña en este caso será guardada dentro de la yubikey. Una vez creado, se abrirá y se copiará el contenido del USB a la partición:

cryptsetup luksOpen /dev/mmcblk0p2 sdcard

Tras insertar la contraseña configurada previamente, se realiza la copia mediante el siguiente comando:

time dd bs=4k count=2775095 if=/dev/sda of=/dev/mapper/sdcard

Se verifica o calcula de nuevo el checksum de la copia realizada. Se ve que es correcta y a continuación, verifica, también el sistema de ficheros de la partición LUKS:

e2fsck -f /dev/mapper/sdcard

Como se ha copiado el sistema de ficheros reducido, hay que expandirlo para ajustarlo al tamaño de la SD card:

resize2fs -f /dev/mapper/sdcard

Se escribe en dicha terminal de initramfs exit y se comprubea que la raspberry arranca de nuevo correctamente. Se verifica que así lo hace.

Como no se quiere que la raspberry siempre que se inicie entre de nuevo en la terminal initramfs, lo que se hace es re-construir la initramfs de nuevo con los siguientes comandos y se reinicia:

sudo mkinitramfs -o /tmp/initramfs.gz

sudo cp /tmp/initramfs.gz /boot/initramfs.gz

Tras el reinicio, pedirá insertar la contraseña que se ha especificado previamente para desencriptar la correspondiente partición. Una vez hecho esto, queda realizar dicha autentificación/desencriptación mediante la yubikey. Para lo cual se siguen los siguientes pasos:

1. Se instalará el paquete específico para usar yubikey con las particiones LUKS:
   sudo apt-get install yubikey-luks
2. Se verifica los slots usados en la información sobre la cabecera LUKS
   sudo cryptsetup luksDump /dev/mmcblk0p2 sdcard
3. Se realiza y se copia a otro USB externo estos headers, puesto que en caso de error o una configuración errónea, no se podrá volver acceder la información encriptada y se perderan los datos de la partición root:
   *sudo cryptsetup luksHeaderBackup /dev/mmcblk0p2 sdcard *
   --header-backup-file /media/dev/sda/safe-storage/${HOSTNAME}-LUKS-header.backup-$(date -u +%Y-%m-%d_%H-%M-%S)
   Donde como nombre orientativo y único, se especifica la fecha de creación de la copia.

4. Se inicializa el modo HMAC-SHA1 challenge/response en el slot 2
   ykpersonalize -2 -ochal-resp -ochal-hmac -ohmac-lt64 -oserial-api-visible
5. Como previamente se ha descargado el paquete para utilizar la yubikey con las particiones de tipo LUKS, lo que se hará ahora es enrolarlo con dicha partición:
   sudo yubikey-luks-enroll -d /dev/mmcblk0p2 sdcard
   En este punto, el script pedirá que se inserte una contraseña para autentificarse con la yubikey. Esta contraseña es la que hace que se autentifique ante la yubikey para que posteriormente se consiga la contraseña guardada en su interior. Para ello, en este punto, se presiona el botón de la yubikey una vez, lo cual genera una cadena de caracteres de manera aleatoria. Esta cadena de caracteres, está configurada previamente cuando se configura la yubikey, y es estática, pero lo suficientemente larga como para no poder desencriptada por ataques de fuerza bruta (32 caracteres).

6. Tras presionar el botón, se inserta la contraseña generada previamente. Dicha contraseña también será de una longitud lo suficientemente grande como para no poder ser corrompida

por ataque de fuerza bruta. Así se configuro en apartados previos, y así se ha guardado en este punto.

7. Tras ello, se habilita el uso de la yubikey con la el fichero /etc/crypttab, especificándoselo de la siguiente manera:

   sdcard /dev/mmcblk0p2 none luks keyscript=/usr/share/yubikey-luks/ykluks-keyscript,discard

   Esta línea de comando hará que el proceso de arranque haga una llamada al script ***/usr/sharek/yubikey-luks/ykluks-keyscript***, que a su vez enviará la contraseña ingresada por el usuario como desafío al Yubikey y enviará la respuesta del Yubikey a LUKS para descifrar el disco. Posteriormente, se guardan los cambios realizados a este archivo.

8. Posteriormente, se actualiza el archivo initramfs mediante el siguiente comando para pasarle la configuración realizada hasta ahora:

   sudo update-initramfs -u

Tras estas configuraciones se da por cerrada la configuración de este escenario. Ahora, se pasará a comentar la configuración de la configuración realizada para evitar la obtención de los permisos root en la raspberry, así como lo referido a la conectividad a la raspberry mediante SSH.

Permisos de usuario root mediante autentificación con la cripto tarjeta yubikey

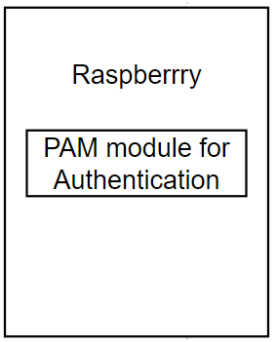


Figura 23: Módulo PAM de la raspberry. Control permisos usuario root

Los pasos en este este apartado son los siguientes:

1. Se actualizaran los paquetes que hay hasta ahora en la raspberry además de instalar las últimas versiones de los mismos y eliminar las correspondientes dependencias bloqueantes para que no haya problemas usándolos. Para ello se escribe por la terminal de la raspberry el siguiente comando:

   sudo apt-get update && sudo apt-get dist-upgrade

2. Se instalará el paquete necesario, y que contiene las librerías necesarias para configurar la yubikey:

   sudo apt-get install libpam-u2f pamu2fcfg

   Este paquete contiene el módulo PAM, que realiza el paso de autentificación U2F. Este módulo llamado PAM por ser las siglas Pluggable Authentication Module, ayuda a las aplicaciones a hacer un uso apropiado de las cuentas de usuarios en sistemas operativos de Linux. En el caso de ahora, para utilizar la conexión SSH. Separa las acciones estándar y las acciones especializadas para la autenticación de las aplicaciones. En definitiva, un conjunto de componentes que facilita la gestión de autenticación de las mismas. Los componentes de este módulo son ***/etc/pam.d***, conjunto de archivos que llaman a la libpam. Que gestiona el método de autenticación, ***/etc/security***, que permite granular los correspondientes chechk gestionados por modulos como pam_access y pam_time o como, por ejemplo, la creación de un registro centralizado, fichero que registra las incidencias que hay por usar lo comentado previamente. Este fichero se encuentra en **/var/log/secure**. Se ha añadido la instalación del paquete pamu2fcfg, puesto que en distro Linux como Ubuntu y en distros enfocadas a ordenadores, que son más completas que las que se usan en las raspberrys, esta viene dentro del paquete libpam-u2f. En el caso, propuesto y con el sistema operativo utilizado, esto no es así.

3. A continuación, se escribirá el siguiente comando por la terminal, para crear en el directorio ***.config/***, uno con el nombre Yubico:

   mkdir -p ~/.config/Yubico

   La razón para usar (-p) es para que, de no haberse creado un fichero padre, se cree al crearse el comentado. Este albergara la configuración que se realiza en el siguiente comando. No es el caso, pero normalmente se realiza para servidores que no disponen de dichos ficheros.

4. Se inserta la yubikey, y a continuación se escribe el siguiente comando por terminal:

   pam2fcfg > ~/.config/Yubico/u2f_keys

   Que es donde ira el output de la configuración que se realiza de manera automática al insertarla y ejecutar dicho comando.

5. En este punto, el led de la yubikey ha comenzado a parpadear, sinónimo de que se tiene que presionar el botón que contiene (dicho botón se aprecia en imágenes previas). Y una vez presionado la yubikey ha quedado emparejada con la raspberry. No obstante, ahora mismo la raspberry no realizara ninguna acción, puesto que no se ha configurado para actuar ante ellas, por lo que los siguientes pasos estarán destinados para dicha configuración.
6. Obtener los permisos de usuario root por terminal consiste en escribir delante de cada comando, la palabra sudo. Por lo que se controlará que cada vez que se utilice dicho comando, se necesite si o si, la yubikey. Esto hará que, como se ha comentado en apartados anteriores, se evite que los usuarios que están trabajando dentro, sin la correspondiente yubikey no se puedan autenticar para obtener dichos privilegios. Para ello, en el archivo ***/etc/pam.d/sudo***, se añadira la siguiente línea:

   auth required pam_yubico mode=challenge-response chalresp_path=/etc/yubico

Esto lo que hace es lo comentado. Encuentra el correspondiente usuario asociado en ese archivo, archivo que solamente lo puede leer y escribir usuario root, y pide que la yubikey este insertada cuando se realice la correspondiente autentificación. De no estar la yubikey insertada, por mucho que se inserte la contraseña correcta, la terminal no permitirá ejecutar dicho comando. Por lo que, para poder ejecutarlo, la yubikey tendrá que estar insertada, y el usuario poseedor de la misma tendrá que presionar el botón que contiene, que parpadea cada vez que se realiza dicha petición de autentificación.

Por otro lado, cabe destacar lo siguiente: En servidores físicos y de nuestra propia propiedad, habría que jugar con las ACLs para el acceso a las diferentes carpetas o la asignación de los grupos y la propiedad de los mismos. En servidores remotos, si se es administrador se tiene que conseguir ser usuario root para que pueda realizar cualquier tipo de administración.

Por lo tanto, la raspberry se ha configurado además de tal forma que siempre se le pida al usuario tener insertada la yubikey a la hora de utilizar el comando sudo. Para ello, se ha modificado la siguiente línea al fichero que está en el siguiente path, /etc/sudoers.d/010_pi-nopasswd, con lo siguiente:

ander ALL=(ALL) PASSWD: ALL

Es decir, que al usuario ander, que en esta configuración es el usuario root inicial con el que se ha configurado los entornos que trabajan con esto, se le pida insertar la contraseña. Esta configuración se ha modificado, porque por defecto no se pide la contraseña a dicho usuario. Lo cual, aporta más seguridad al sistema.

Posteriormente, se ha enrolado el usuario “ander” con la yubikey, puesto que es el administrador del servidor. Para ello, se escribe por terminal lo siguiente:

ykpamcfg -2

Comando que crea en la carpeta /home/ander/.yubico/ un archivo llamado challenge-SERIALDELAYUBIKEY, que nuestro caso tiene el número de serie asociado 6947656. Posteriormente, y con los correspondientes permisos, se mueve dicho archivo a la carpeta /etc/yubico/, de la siguiente forma:

sudo mv ~/.yubico/challenge-SERIALDELAYUBIKEY /etc/yubico/ander-6947656

Dicha carpeta, cabe comentar, que la ha creado el usuario root y que, por lo tanto, solo la puede modificar dicho usuario. De esta forma, solo ander podrá ejecutar comandos con los privilegios root. Es más. Si un usuario no está en el grupo denominado sudoers, y aunque este enrolado con la yubikey, tampoco tendrá permisos para poder ejecutar dicho comando.

Con lo que de esta forma se da por configurado dicho control de escalada de permisos.

Acceso mediante conexiones SSH a la Raspberry con Yubikey

Se recuerda en este punto el diagrama previamente explicado en apartados anteriores:

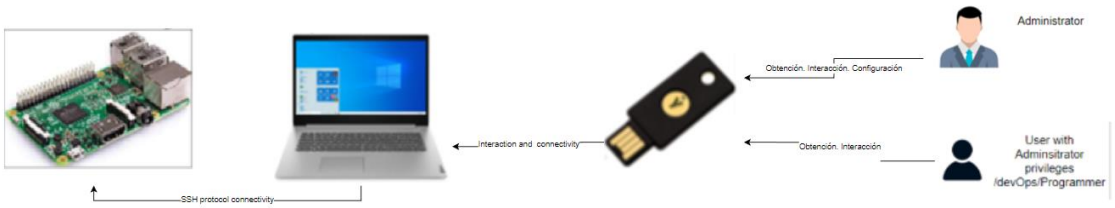


Figura 24: Configuración y conectividad SSH mediante Yubikey

Este caso consiste en que cuando uno de los usuarios de la empresa intente conectarse a la raspberry mediante el protocolo SSH, siempre que se quiera conectar con ese protocolo sin la yubikey no se le permita. Para ello, hay que hacer ciertas configuraciones tanto en la parte de la raspberry como en la parte del ordenador del usuario. Dichas configuraciones son las siguientes:

En la parte del usuario. Es decir, en su ordenador:

1. La versión mínima que acepta esta solución, es la versión 8.4p1 del programa openssh. Este programa viene por defecto en los sistemas operativos de Linux, pero si por lo que sea la versión que se está utilizando en una inferior, hay que hacer el correspondiente upgrade. En el caso propuesto, sí que ha habido que hacer una actualización de versión, pero se quiere enfocar la solución en lo propuesto previamente por lo que dichos pasos no se comentaran. Solo cabe destacar que se descargó el archivo de la versión mínima requerida y que se actualizo. Dentro de ese fichero, se abre una terminal y se inserta el siguiente comando:

   ./configure --with-md5-passwords --with-pam --with-security-key-builtin --with-selinux --with-privsep-path=/var/lib/sshd/ --sysconfdir=/etc/ssh

   Los parámetros especificados hacen referencia a lo siguiente:

   - **--with-md5-passwords**: This enables the use of MD5 passwords.
   - **--with-pam**: Este parámetro habilita la compatibilidad con Linux-PAM en la compilación.
   - **--with-security-key-builtin**: Sin este parámetro, la autentificación fallaría localmente con un error del tipo “internal security key support not enable”.
   - **--with-selinux**: Parámetro que indica que se añada soporte para SELinux. Este indicador permite que OpenSSH funcione en entornos que se ejecutan con SELinux habilitado, ya que SELinux tiene políticas de seguridad adicionales que controlan cómo interactúan las aplicaciones con el sistema. Cuando OpenSSH está configurado con este indicador, podrá usar SELinux para hacer cumplir las políticas de seguridad, si el sistema se ejecuta con SELinux habilitado.
   - **-with-privsep-path**: Path necesario para indicar la separación de los privilegios que tiene chroot.
   - **--sysconfdir=/etc/ssh**: Esto evita que los archivos de configuración se instalen en ***/usr/etc***.

   Todos necesarios para la correcta configuración del servicio SSH

2. Tras preparar la instalación con el anterior comando, se procede a ejecutar los siguientes comandos para su completa instalación:

make

sudo make install

3. Posteriormente, se genera la key que va a utilizar el usuario con el uso de SSH:
   ssh-keygen -t ecdsa-sk -C "$(hostname)-$(date +'%d-%m-%Y')-yubikey1"
   Donde el nombre -yubikey1, es indiferente.

4. Una vez creada dicha key, se copia en la raspberry con el usuario administrador:
   ssh-copy-id ander@192.168.1.119
   IP en este caso, asignada por el DHCP del router a la raspberry

En la parte del servidor, se realizará la siguiente configuración, una vez que la correspondiente key está en el mismo. Para ello:

1. En el archivo de ***/etc/ssh/sshd_config*** por defecto esta puesto que las conexiones ssh sean pidiendo usuario y contraseña. Para no permitir esa opción y solo dejar la de la yubikey se pondrá en "no" la opción de PasswordAuthentication. Es decir, dicho parámetro de esta forma:
   PasswordAuthentication no
2. Se realizará la correspondiente prueba. En este caso, y como se comentará posteriormente, si la yubikey NO esta insertada, la conexión mediante SSH directamente se cancela. En caso de estar conectada la yubikey, esta empieza a parpadear y una vez pulsado el botón se realiza la correspondiente autenticación.

Comentario extra: En caso de querer autenticarnos con un usuario diferente, este usuario también tiene que enviar la correspondiente key a la raspberry. Para poder enviar dicha key, hay que modificar el anterior parámetro a “yes” otra vez para poder enviarla, y una vez enviada, ponerlo de nuevo a “no”. Esta forma de actuar es costosa, pero muy segura puesto que solamente se permite el acceso a la yubikey de manera controlada y siempre bajo la supervisión del administrador de turno.

Esta configuración, no obstante, tiene una traba, que se ha configurado y dejado así a propósito. Y es que cuando cualquier usuario que no esté en el grupo sudoers, para obtener permisos root, tal y como se ha configurado previamente, tendrá que necesitar la yubikey. Hasta ese punto cumple con lo configurado. No obstante, esa autenticación no es posible realizarla desde el ordenador del usuario, puesto que el módulo pam de la yubikey no envía dicha respuesta generada para la autenticación con la yubikey al ordenador destino. Por lo tanto, en caso de necesitar de los correspondientes permisos, se tendrá que hacer bajo demanda y enviando la petición al administrador de la raspberry. En versiones anteriores a la comentada, sí que había dicha comunicación, pero por problemas de seguridad y brecha de seguridad del mecanismo, se eliminó.

Lo cual, aplica otra capa de seguridad extra puesto que dificulta la obtención de permisos root.

Comentario extra (4): Cuando se realiza dicha autenticación, la raspberry sí que da dicha respuesta de manera local (física), porque si se ejecuta el comando sudo en el ordenador del usuario, la raspberry si esta insertada en la raspberry, empieza a parpadear, y si se pulsa el correspondiente botón, se hace que dicha autenticación se realice de manera exitosa.

Entorno de Citrix

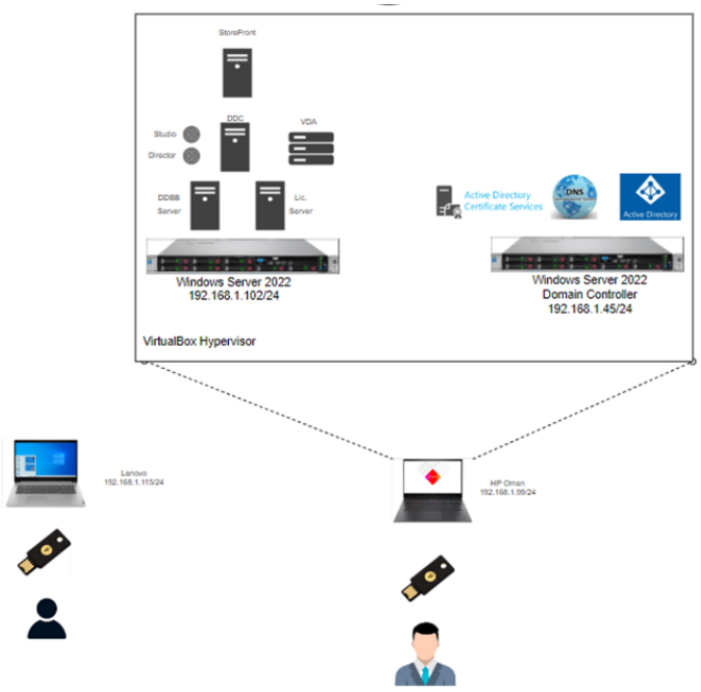


Figura 25: Configuración de los elementos del entorno de Citrix

A continuación, se describiran los pasos seguidos en la configuración del entorno de citrix. Los pasos son los siguientes:

1. En primer lugar, hay que descargar el software requerido para este entorno. Como ya se ha comentado previamente, el entorno consta de servidores de Windows 2022, el programa de Citrix (la imagen del mismo) y el hypervisor que albergará los dos servidores.
2. Tras la descarga de los mismos, se instalará en el VirtualBox. Como se sabe, virtualbox es un software de código abierto para virtualizar los servidores o infraestructura informática requerida. Por lo tanto, se configura. Los pasos para esta instalación no son muy relevantes puesto que es seguir las instrucciones del instalador. En él se especifica el directorio donde se guardarán los archivos de las maquinas virtualizadas y desde donde se va a ejecutar, es decir, el path de la variable de entorno desde donde se ejecutará el hypervisor. Además, se instalarán las correspondientes VBox Guest Additions para cada una de las máquinas. Estas permiten tener funcionalidades como copiar y pegar archivos desde el ordenador anfitrión, así como compartir una carpeta con el mismo desde dentro de la máquina virtual.
3. Para configurar los servidores, primeramente, se tiene que crear el entorno virtualizado para el mismo, para ello:

Figura 26: Cuadro de menu Oracle VirtualBox

Tras abrir el programa, se hace click en la pestaña de “Nueva”. Esto abrirá una ventana nueva.

4. En dicha ventana, se escriben los nombres correspondientes a los servidores. Estos nombres no son los nombres de los sistemas operativos internos, sino que son nombres identificativos dentro del hypervisor. Se selecciona también que serán ambos servidores Windows:

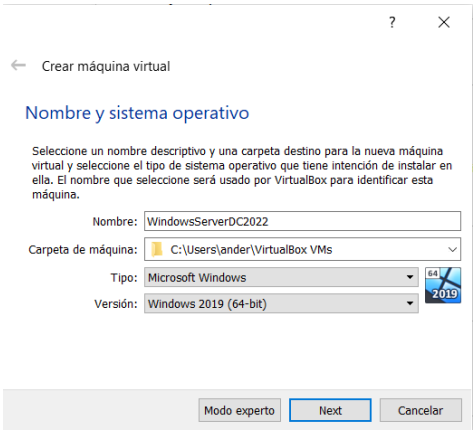


Figura 27: Ventana creación máquina virtual VBox

5. Se selecciona el tamaño de la memoria RAM. Para arrancar estos servidores y teniendo en cuenta las especificaciones técnicas o requerimientos mínimos de las mismas, se ha decidido que el controlador de dominio necesario para este entorno de pruebas es de 4096MBs y para el servidor que alberga el programa de Citrix, son de 5 GBs. Este requerimiento es principalmente para entornos de desarrollo. Al pasar a producción, el fabricante propone un mínimo de 12 GBs. A tener en cuenta también para futuras ampliaciones o paso a producción.
6. El controlador de dominio, a partir de ahora llamado DC, no necesita más que albergar los servicios de DNS, Active Directory y los servicios necesarios para ser autoridad ceritificadora, por lo que los requerimientos de almacenamiento no son tan grandes. Se dejará en 50 GBs. El servidor que contiene el programa de Citrix, llamado a partir de ahora “Servidor Citrix”, sí que necesita más, por lo que se la ha asignado una capacidad de 100 GBs.
7. Tras el despliegue de los mismos, se ajustan en ambos los núcleos para que funcionen de manera más aceptable, por lo que al DC se le asigna 1 y al servidor Citrix, 4. Cabe mencionar que cuantos más se le dé mejor. Estando limitados en este aspecto, la asignación comentada para el entorno de pruebas es suficiente.
8. Se configura además la conectividad que van a tener ambas con el sistema operativo anfitrión, y se especifica el modo “Adaptador Puente” que ofrece el hypervisor. Este modo lo que hace es lo siguiente: conecta el adaptador de red virtual de la máquina virtual, a la red física que está conectado el sistema anfitrión, lo que permite que la máquina virtual tenga conectividad hacia el exterior, o que disponga de conectividad con otros dispositivos dentro de la red donde se ubique el anfitrión.
9. Se selecciona que los sistemas de arranque sigan este orden puesto que, de no ser así el arranque de la máquina virtual es más lento:

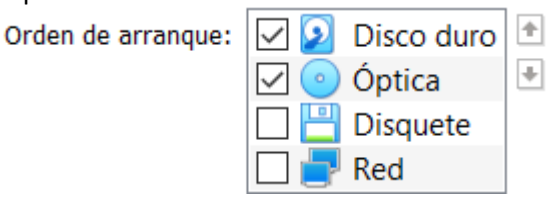


Figura 28: Orden de arranque especificado para la máquina virtual. Opciones seleccionadas

10. Se selecciona también el disco virtual que contiene el sistema Operativo de Windows descargado previamente y se encienden.
11. En la configuración inicial, se realizan las particiones del almacenamiento correspondientes y necesarios para el correcto funcionamiento de la máquina. Cabe destacar aquí, que las particiones especificadas, en caso de necesitar más capacidad de almacenamiento la compartirán con el anfitrión, de manera dinámica. En data centers de entornos de producción, este redimensionamiento se hará por la terminal de configuración del hypervisor utilizado.
12. Se especifica también el administrador local de la máquina, y se procede a añadir al DC los correspondientes roles del Active Directory, así como los necesarios para ser una autoridad certificadora. El resultado de la configuración de dichos roles queda de la siguiente forma:

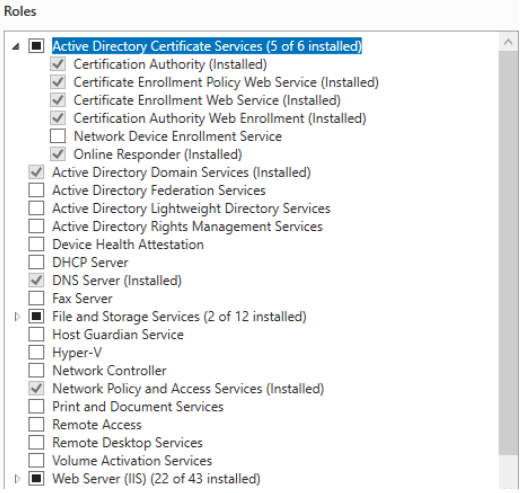


Figura 29: Roles y características servidor de Windows. Programa Server Manager

Comentario extra (5): Puesto que son roles requeridos y necesarios para la solución se detallarán a continuación que es lo que cada uno de ellos:

- **Active Directory Services (AD CS)**: Rol que permite crear autoridades certificadoras, así como los roles necesarios para la gestión de los certificados expedidos.
- **Active Directory Domain Services (AD DS)**: Rol que guarda la información de los objetos en la red y que permite exponer la información de manera disponible a los usuarios o administradores de la red. Utiliza además controladores de dominio para permitir el acceso a los usuarios de manera autenticada.
- **DNS server**: Provee de resolución de nombre para redes TCP/IP. Trabaja de manera conjunta con el AD DS
- **File and Storage services**: Instalados por defecto, que permite una mejor gestión de fichero y el almacenamiento del servidor donde está instalado.
- **Network policy and Access Services**: Provee del requerido servidor NPS, que ayuda a salvaguardar de manera segura la información.
- **Web Server (IIS)**: necesario también para expeditar los certificados que genera la autoridad certificadora instalada y hacerlos descargables desde cualquier navegador de un equipo dentro de la red o dominio.

13. Tras la comentada configuración de roles en dicho servidor y con por descontado, sus respectivos reinicios de la máquina virtual, mediante la consola de Active Directory, se crea un Administrador de dominio, necesario para la administración de los equipos de manera remota, pero dentro del mismo dominio. Este administrador permitirá acceder a las máquinas de

manera remota, en este caso, como son servidores Windows, mediante el protocolo Remote Desktop Protocol, sin tener que autenticarse con el usuario de administrador local, reservado exclusivamente, para casos excepcionales.

14. Tras esto, se procederá con la configuración del servidor de Citrix. En este caso, no necesita de los roles del DC, pero si necesita en cambio, tener un hostname y que pertenezca al dominio propagado por el DC. Para ello se siguen los siguientes pasos:
    a. Tras la verificación de que este instalado el rol de servidor IIS, necesario para publicar las aplicaciones de citrix, se abre en el panel de control, System and Security, System, las opciones avanzadas de administración y se enrola este servidor al dominio que se llamara citrix.com. De tal forma que el CNAME que se agregara al DNS sea el siguiente: *maqCitrix.citrix.com*
    b. Se le asigna dentro del rango que permite el servidor DHCP, una IP estática para evitar problemas de conectividad al servidor y se le indica que el DNS principal es el DC.
    c. Se le permite la conectividad mediante RDP para su gestión.
    d. Y puesto que Citrix a parte de su propio servidor de licencias, necesita un servidor de licencias RDP, por lo que también se le instala el correspondiente rol.

En este punto, los servidores de Windows, y el dispositivo anfitrión tienen visibilidad entre ellos y ambos servidores se pueden acceder mediante RDP. Además, la resolución de nombres también es la correcta. Por lo tanto, en este punto se pasará a comentar la configuración del programa de Citrix. Si que es cierto que en algunos de los siguientes puntos se comentara la configuración de la autoridad certificadora, como parte fundamental de la solución.

A continuación, se procederá a describir la configuración del programa Citrix. Los pasos seguidos para dicha configuración son los siguientes:

1. Tras la descarga del correspondiente .iso mediante una cuenta de proveedor de Citrix, se monta dicha imagen en el servidor de Citrix y se ejecuta con el ejecutable “AutoSelect”
2. La primera ventana que te sale es la siguiente:

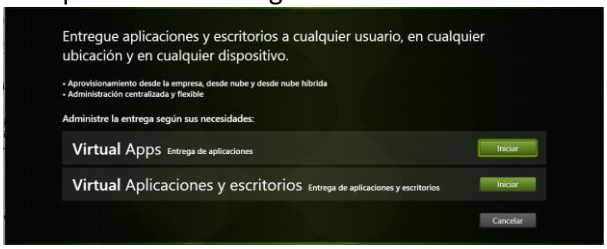


Figura 30: Tras montaje .iso de Citrix, opciones primera ventana de selección

3. En dicha pantalla se selecciona la opción Virtual Aplicaciones y escritorios, puesto que el proveedor ha concedido únicamente esas licencias de uso.
4. La siguiente pantalla muestra las opciones que se tiene para su instalación. En ellas aparecen los componentes como Citrix Director, Citrix Studio, Session Recording, Citrix License Server, Universal Print Server, Citrix Workspace Environment Management Agent, Citrix StoreFront y Federated Authenticated Service, Delivery Controller y Virtual Delivery Agent. Los que interesan y se han configurado inicialmente en este entorno son los siguientes (se comenta también una explicación sobre que hace cada uno). Cabe comentar que el VDA, se ha configurado en pasos posteriores:
    a. **Delivery Controller**: Componente del lado del servidor que es responsable de administrar el acceso de los usuarios, además de intermediar y mejorar la conexiones.

Además, tienen los servicios necesarios para la creación de escritorios virtualizados.

b. **Studio**: Consola de administración. Lugar donde se configura y administra Citrix Virtual Apps and Desktops. Incluye lo necesario para alojar las aplicaciones que se van a exponer a los usuarios. También los escritorios virtualizados. También proporciona la información sobre el seguimiento y estado de las licencias.
c. **Director**: Basada en una consola web, monitoriza el entorno y realiza tareas de soporte para los usuarios finales. Permite monitorizar de manera centralizada uno o más Citrix Virtual Apps and Desktops
d. **License Server**: Servidor (en nuestro entorno, alojado en el mismo servidor de Citrix), que administra las licencias del Citrix. Se comunica con el controlador para dar la correspondiente licencia a cada usuario para su correspondiente sesión. Componente indispensable para el entorno.
e. **StoreFront**: Autentica los usuarios y administra los lugares donde se ubican las aplicaciones o escritorios desplegados. Recolecta información sobre el estado de la sesión y el usuario y gestiona el correcto funcionamiento de las sesiones desde varios dispositivos.

5. Se selecciona el lugar por defecto para la instalación de los componentes y se prosigue con el instalador. En este punto el programa realiza un análisis del hardware donde se hospeda y si detecta que no se cumple uno o varios requisitos mínimos te alerta de las posibles consecuencias que puede haber no cumpliéndolos.
6. Adicionalmente, el programa necesita trabajar sobre un SQL Server Express, donde almacena la información recolectada. Por lo que en la siguiente pantalla del administrador se instala.
7. Citrix utiliza por defecto los siguientes puertos por lo que, en el firewall del servidor hay que configurar las correspondientes políticas:
    a. 80,443 TCP → Delivery Controller
    b. 80,443 TCP → Director
    c. 7279,27000,8083,8082 TCP → License Server
    d. 80,443 → StoreFront
8. En un momento de la instalación, el programa pide dos reinicios necesarios para instalar las dependencias extras. Se reinicia y se prosigue con la instalación.

En este punto los principales componentes quedan instalados. Se continua con la configuración en este punto, empezando por la creación de un site. Se denomina site a la implementación de un Virtual Apps and Desktops. Contienen los delivery controller que utiliza la configuración, los Virtual Delivery Agents, conexiones a hosts, catalogos de máquinas y grupos de entrega. Elementos que se describirán más detenidamente cuando se configuren. Por lo tanto:

1. En el Citrix Studio nada más conectarse aparece el Wizard de crear un site. En la primera pantalla se elige crear un site del tipo “Application and Desktop delivery Site”. De los dos modos que permite elegir, un site completo o un site vacio, se elige el completo.

2. Se le da un nombre al site, que aparecerá en el panel lateral del Studio, de la siguiente forma:

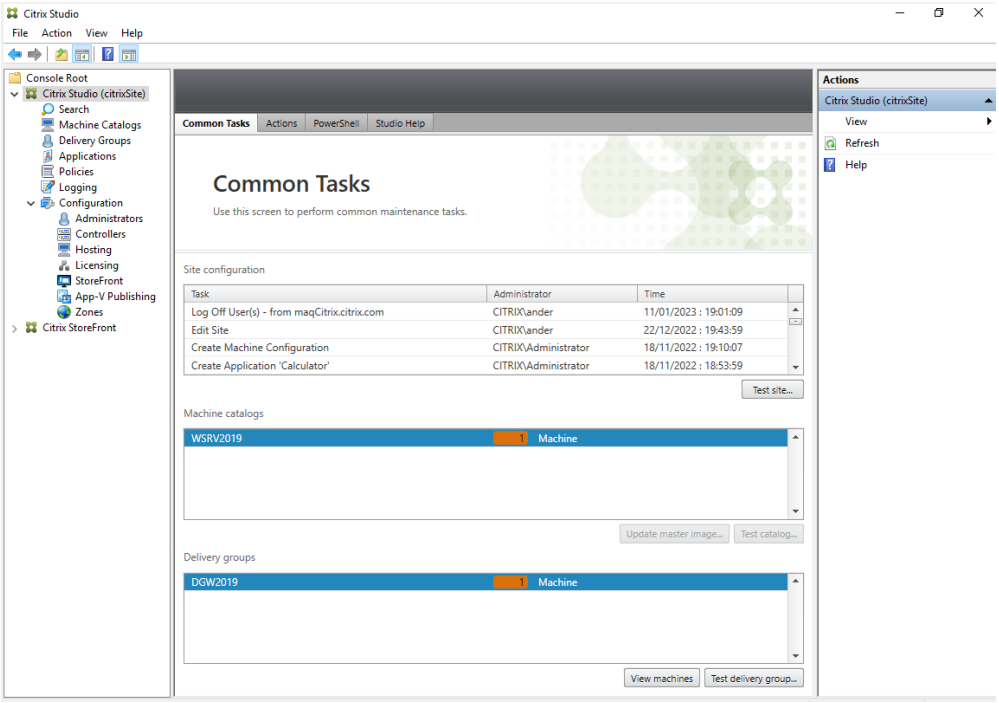


Figura 31: Ventana de consola Citrix Studio

Comentario extra (6): En una de las pantallas de configuración del site se comenta si se ha instalado la base de datos recomendad. Es cierto que previamente ya se ha instalado la base de datos que se va a usar en este proyecto, pero esta pantalla aparece por si se ha utilizado otra que no sea la especificada en los pasos previos y la opción de configurarla. Como no es el caso, se continua con la instalación.

3. Una de las partes fundamentales, casi la más importante de todas es haber obtenido, como se ha comentado previamente, las correspondientes licencias de uso. Se recuerda que este software es de pago y que, por lo tanto, las licencias hay que pagarlas. En el caso del proyecto, se han obtenido licencias válidas para un año. Dicho lo cual, donde en la imagen anterior se ve "Licensing", se hace click, y se despliega la siguiente ventana:

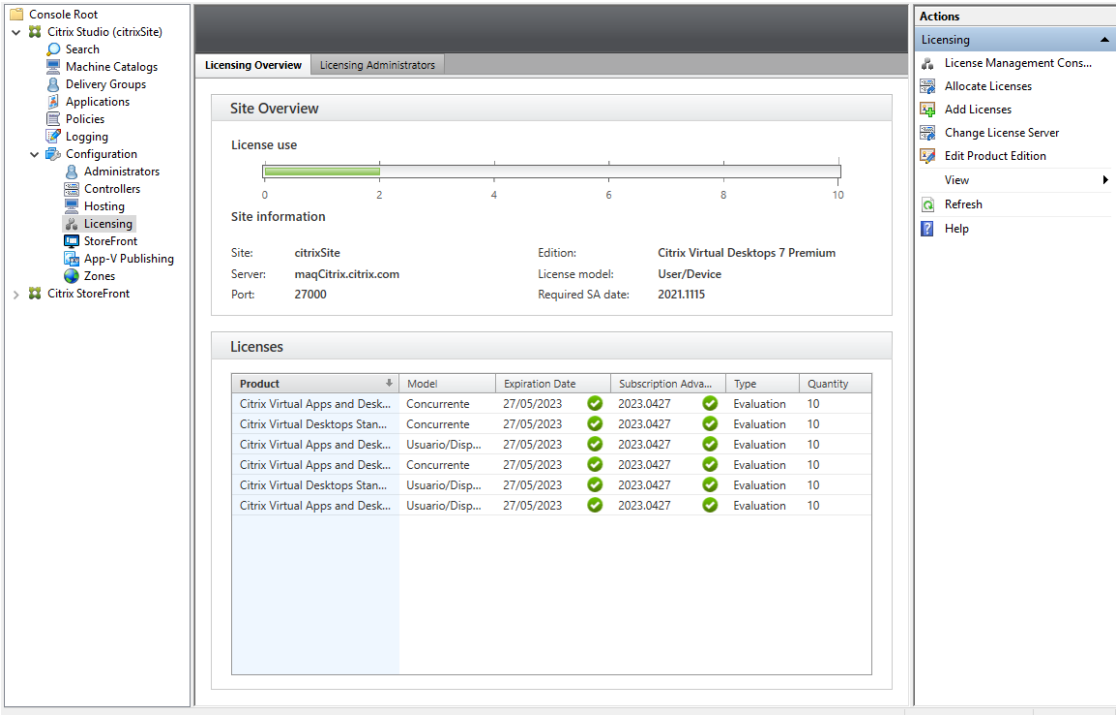


Figura 32: Opción Licensing. Estado de las licencias disponibles y su correspondiente uso

4. Se hace click en “Change License Server” y se añade el nombre FQDN del servidor de la siguiente manera:

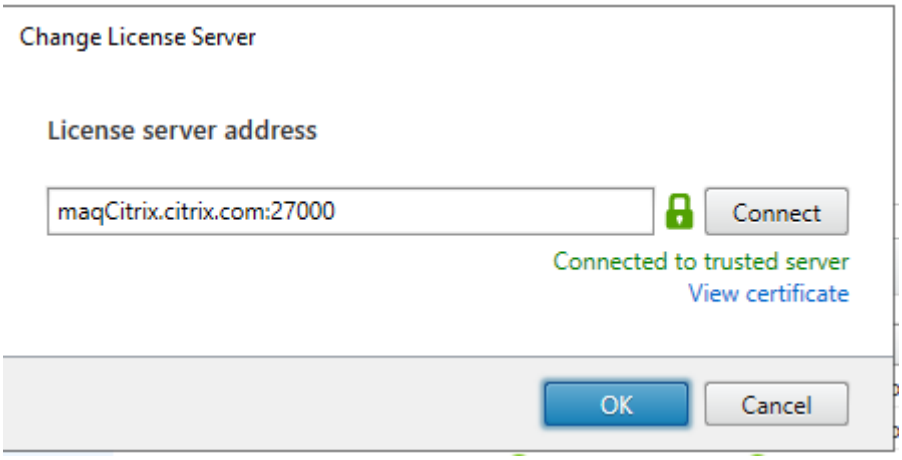


Figura 33: Prueba conectividad al servidor de licencias de Citrix

Es decir, se especifica el nombre completo del servidor de licencias, que en este caso es el mismo que contiene el programa. Y tal y como se especifica, por el puerto 27000. Hasta que no se haga una conexión satisfactoria no se puede seguir con la configuración del mismo. Por

lo que, tras la revisión de las correspondientes políticas del Firewall interno del servidor y su posterior configuración, se da por buena la conectividad.

5. Por defecto, la opción de usar una licencia existente está marcada, no obstante, como no hay licencias especificadas todavía, se añaden mediante el buscador las obtenidas. Cabe destacar que es un archivo LIC, con las necesarias para la solución.
6. En este punto solamente quedaría realizar la prueba de los test predefinidos por Citrix, para verificar que la configuración ha sido la correcta. Se puede dar el caso de que alguno de los mismo de Warning, pero se hará hincapié en ellos si en los siguientes pasos aparecen errores. En el caso de estudio, no ha sido así:

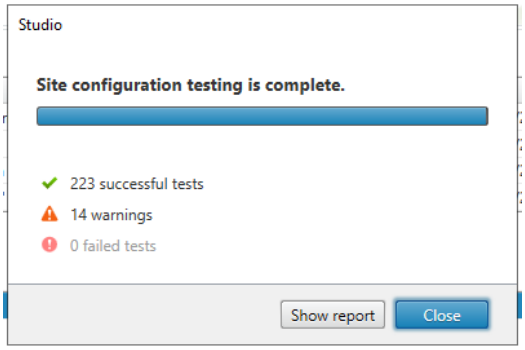


Figura 34: Ventana de comprobación de los tests realizados por defecto

Comentario extra: Los warnings que salen en este caso son porque no se ha hecho un backup de la base de datos que alberga el programa. No obstante, como la base de datos está dentro del servidor, y corre sobre el almacenamiento del mismo, durante la ejecución del proyecto se han hecho backups de los dos servidores, tanto del DC como de este. De ahí, que los warnings en este caso, sean obviados:

Created by ander
02/02/2023 18:02:51

Test name	Target Service	Date and time	Test result
Check there is a recent database backup. Check there is a recent database backup. **Test run on entire Site** ⚠ 76 days since database was created or last database backup. *Create a backup for database MAQCITRIX\sqlexpress\CitrixcitrixSiteSite.*	Host	02/02/2023 17:54:31	Warnings
Check there is a recent database backup. Check there is a recent database backup. **Test run on entire Site** ⚠ 76 days since database was created or last database backup. *Create a backup for database MAQCITRIX\sqlexpress\CitrixcitrixSiteSite.*	AD Identity	02/02/2023 17:54:31	Warnings

Figura 35: Output de los tests realizados. Estado de las pruebas realizadas

Una vez instalados, se ha proseguido con la configuración del VDA. Como se ha comentado este agente se instala en el servidor virtual o físico que se quiera poner a disposición de los usuarios de la solución. Por lo tanto:

1. Para la instalación de este agente, se tendrá que volver a montar la iso descargada y montada previamente. Tras montarla, y seleccionando la opción comentada previamente. Se selecciona la opción de "Virtual Delivery Agent for Windows Multi-session OS. Aunque la opción permita hacer Single-session o multi-session, con las licencias instaladas previamente de multi-session, el instalador verifica que son esas y solamente da la opción de instalar multi-session.
2. En la siguiente pantalla te da la opción de elegir el entorno en donde se va a instalar esta máquina. Como es este propio servidor, se seleccionará para que no tenga que tener relación con ningún programa de aprovisionamiento.
3. Se seleccionan los componentes a instalar. A parte del requerido, "Virtual Delivery Agent", da la opción de instalar el Citrix Workspace App. Esta última se instalará después en las maquinas clientes, por lo tanto, se instalará posteriormente. Es un programa para instalar en dispositivos de cliente, por lo tanto, en este servidor, en este punto de la instalación y configuración, no es necesario.
4. En la siguiente ventana, se indicaran cuales son los Delivery Controllers de la solución. Como se va a instalar en el mismo, con poner el nombre completo, como se ha hecho previamente en algún paso anterior, es suficiente:

   maqCitrix.citrix.com

5. Posteriormente, se verifican los puertos que se necesitan, entre los que se encuentran los siguientes: 80, 1494 (TCP y UDP), 2598, 8008, 1494 y 2598.
6. Tras el correspondiente reinicio del servidor, el Virtual Delivery Agent queda configurado.

Tras estos pasos, se creara un catálogo de máquinas necesario para exponer las aplicaciones que usaran los usuarios y se creara también el delivery group. Los **"machine catálogs"** son las colecciones de máquinas físicas o virtuales que se administran como una sola identidad. Las máquinas que pertenezcan al mismo, llevan consigo el mismo sistema operativo, ya sea multi-session o single-session. No obstante, los catálogos solo pueden tener o maquinas Linux o Windows, pero no las dos mezcladas en un mismo catálogo. Y el delivery grupo, por otro lado, es una colección de uno o más

catálogos de máquinas. Este grupo también tiene da la posibilidad de especificar que usuarios pueden usar las maquinas seleccionadas, además de las aplicaciones y escritorios virtuales disponibles. Por lo tanto, se describirá la configuración de ambas a continuación, empezando por el machine catalog. Comentar que se configura previamente esto, porque luego el delivery group lo selecciona. Por lo que:

1. En el panel lateral izquierdo del Citrix Studio, se selecciona “Machine Catalogs” y se le da al botón del panel lateral derecho *Create Machine Catalog*
2. Se selecciona el sistema operativo de las máquinas que se incluirán en este catálogo. No hay que confundir sistema operativo como lo pueden ser Windows, Ubuntu, etc., sino más bien a si van a ser Multi-session o single-session. Se elige multi-session siempre basandose en las licencias obtenidas.
3. Se selecciona que el despliegue no lo hará el propio Citrix si no que un software de terceros.
4. Se seleccionan las cuentas de las máquinas que ofrece el Active Directory instalado previamente, y con los dos servidores enrolados. Se elige la propia del servidor, que en este caso es la cuenta con la que se ha enrolado al Active Directory y que se llama: “MAQCITRIX”.

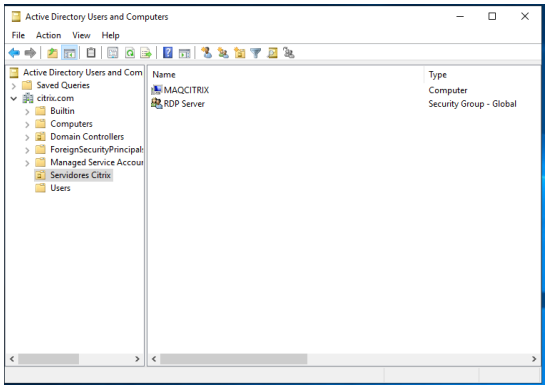


Figura 36: Consola Active Directory. OU de los servidores de Citrix

5. Tras esto, el machine catalog ya estaría creado, y por el Studio se vería lo siguiente:

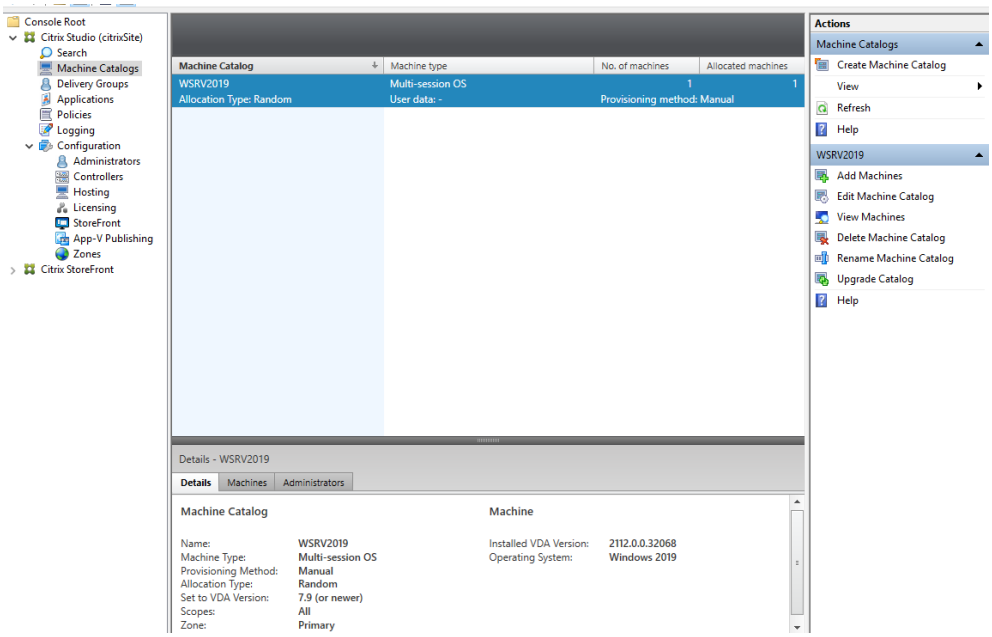


Figura 37: Resultado de la creación del Machine Catalog

A continuación, se procede a configurar el Delivery Group. Los pasos seguidos son los siguientes:

1. Se selecciona del panel de control izquierdo “Delivery Groups” y se selecciona “Create Delivery Group.
2. Se hace click en el catálogo de máquinas creado previamente.
3. Se seleccionan los usuarios y los grupos de usuarios que usaran las aplicaciones que se van a desplegar.
4. Se seleccionan que solamente puedan acceder a las aplicaciones desplegadas aquí los usuarios que previamente este autenticados.
5. Se seleccionan las aplicaciones que se quieran desplegar. En este apartado se pueden desplegar aplicaciones propias, de terceros, propias del servidor, etc. Cualquiera. Como el objetivo de este proyecto es demostrar la configuración y el proceso de autenticación mediante la yubikey, la aplicación desplegada será la calculadora de Windows, que por simpleza y como verificación de la autenticación, valdría. Como no se disponen de los recursos suficientes para desplegar escritorios virtualizados, esta opción de despliegue queda descartada.
6. Por lo que el resultado final es el siguiente:

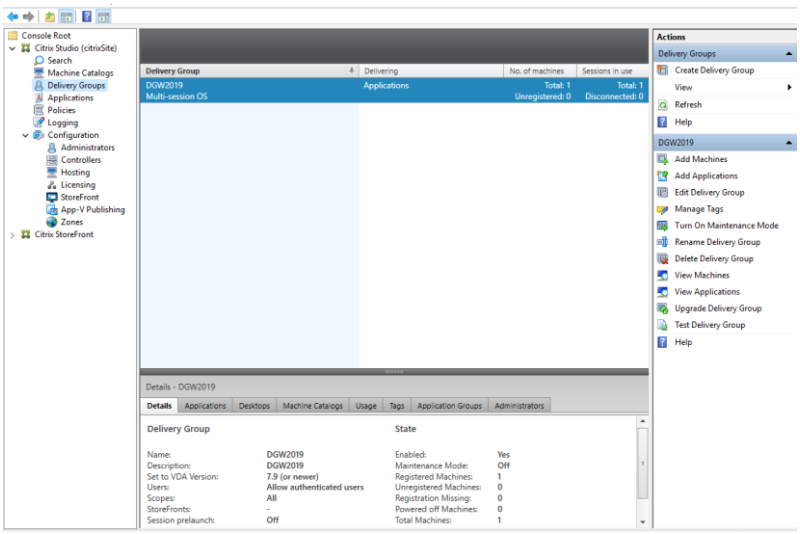


Figura 38: Delivery Groups desplegados. Output por consola de gestión Citrix Studio

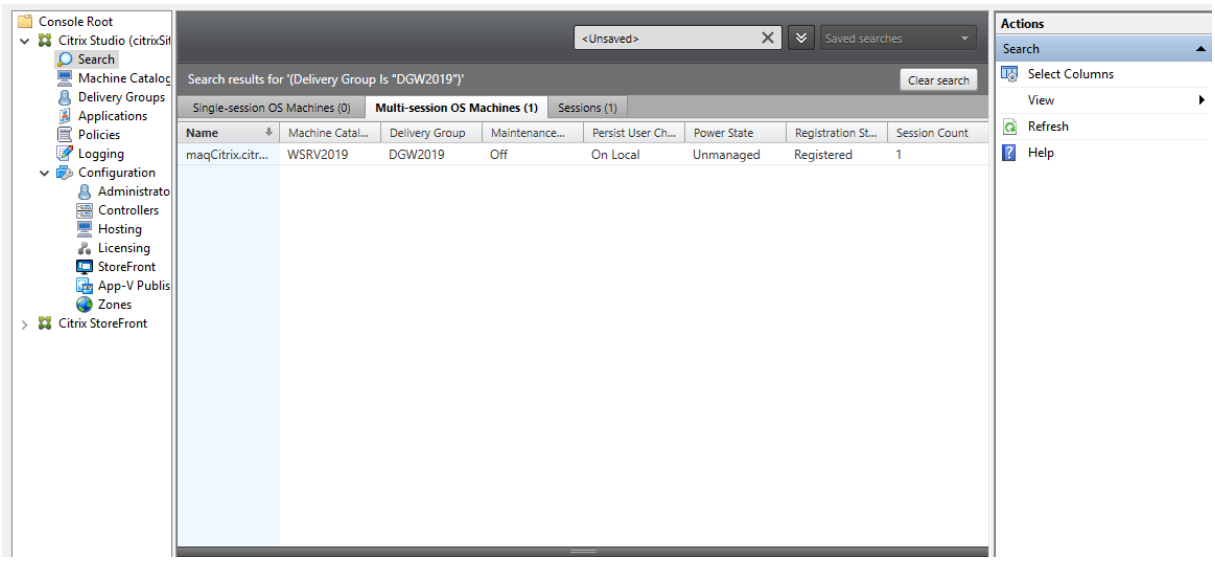


Figura 39: Ventana sobre el estado de las sesiones en activo

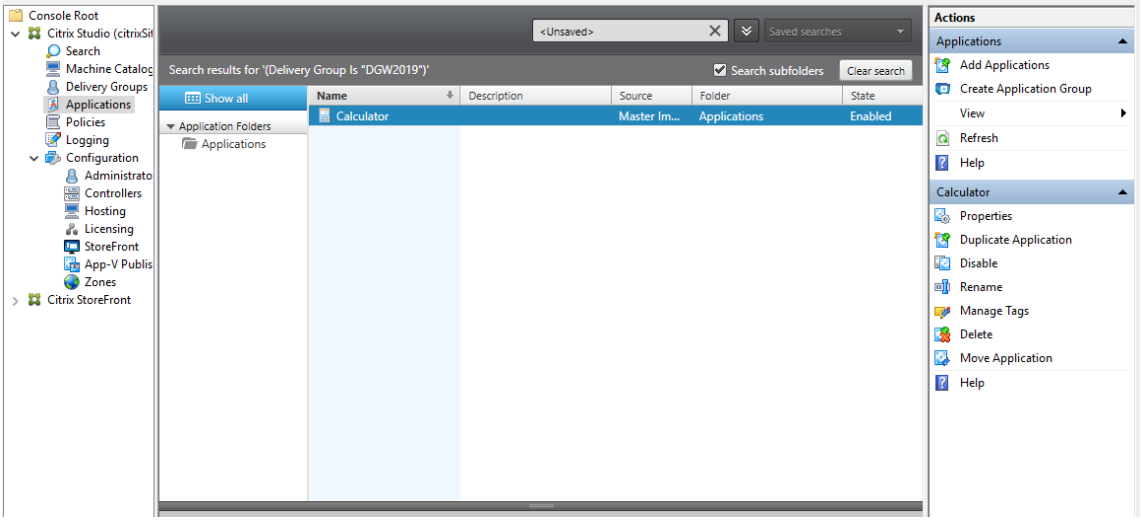


Figura 40: Grupo de aplicaciones desplegadas para grupo de usuarios

Es decir, la aplicación, el delivery group y las maquinas pertenecientes al catálogo. En este punto, ya se tienen las aplicaciones desplegadas. Y se comentan aplicaciones porque de la misma forma que se ha publicado la calculadora de Windows, se puede añadir cualquier otra al delivery group. Por lo tanto, se pasará a configurar el StoreFront. Para ello, se han seguido los siguientes pasos:

1. Este elemento de la configuración necesita de la característica de Microsoft .NET Framework, por lo que este primer paso será verificar que el servidor DC, puesto que se va a instalar ahí, contiene dicha característica:

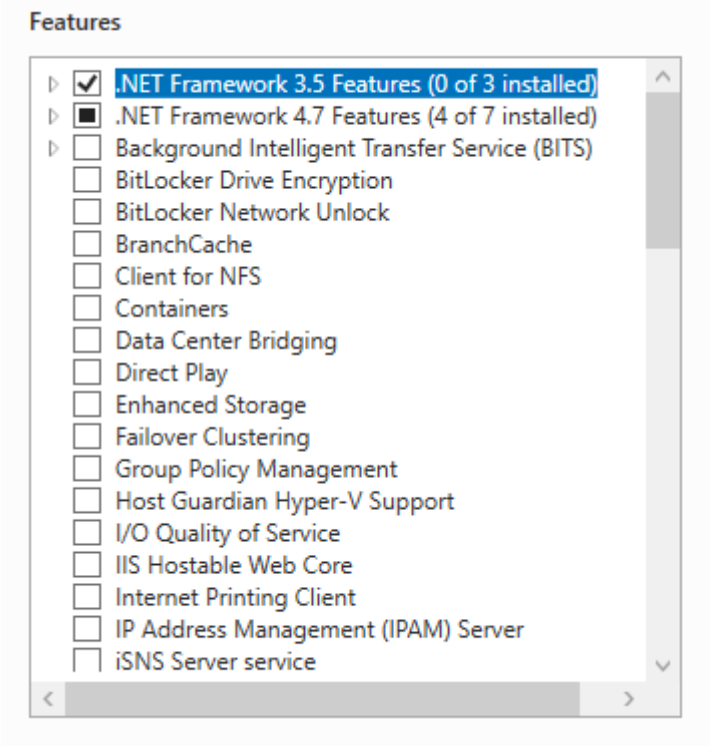


Figura 41: Características necesarias para la configuracion del servidor IIS

Como lo tiene instalado se continua.

2. En esta ocasión se necesitará del administrador local para la configuración, por lo que se cierra la sesión que se tenía abierta hasta ahora y se registra como administrador local. Una vez hecho esto, se ejecuta el instalador. Este instalador no viene con la imagen iso comentada previamente, por lo que también hay que descargar de la página el correspondiente instalador.
3. Se ejecuta, y se sigue los pasos del mismo. Esta instalación no tiene nada más de especial que seguir los pasos del instalador, verificar que los puertos que necesita son están debidamente abiertos y ver que se cumplen los requisitos mínimos necesarios.
4. Pide reinicio el instalador por lo que se reinicia y se abre el correspondiente programa, obteniendo la siguiente ventana:

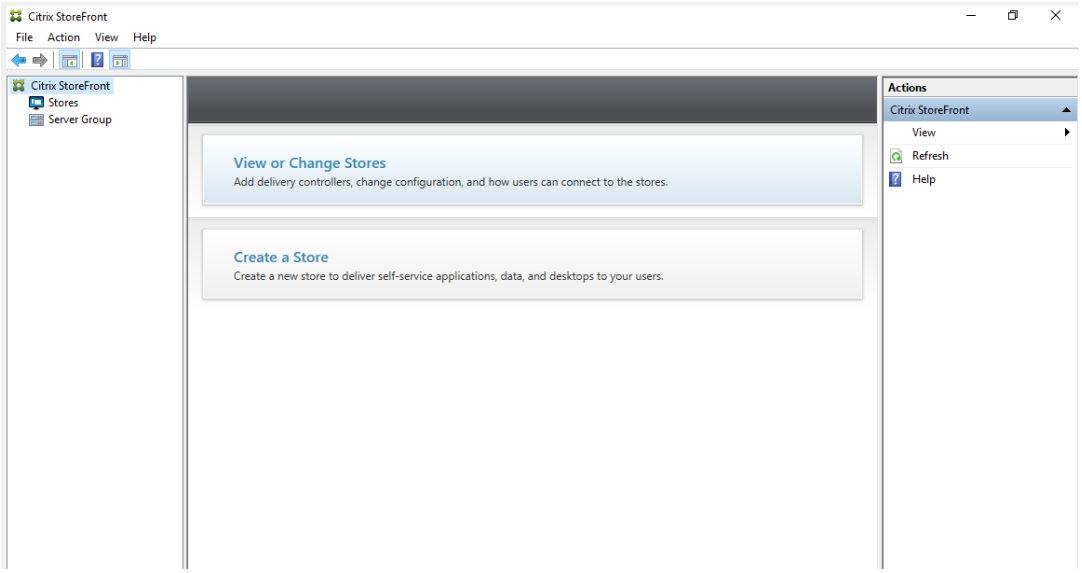


Figura 42: Consola Citrix StoreFront. Primer paso, creación site

5. Una vez ahí, se selecciona “Create Store”.
6. En la ventana que aparece, se especifica la URL del servidor que trabajará como tal, por lo que, en este caso, también será el mismo servidor. En entornos de producción, y por motivos de seguridad y disponibilidad, se recomienda que este servidor este alojado en otro servidor. No obstante, como es un entorno de desarrollo, se mantendrá en el mismo.
7. En la ventana donde se habla sobre los Delivery Controller, ahí se añaden el que se ha designado como tal, y se especifica que la conectividad con el mismo sea mediante HTTP. Esto es así puesto que esta desplegado en la intranet y dentro del dominio local, sin salida a internet. En caso contrario especificar conectividad mediante HTTPS.
8. Por lo que en este momento ya hay posibilidad de acceder al Store Front mediante página web:

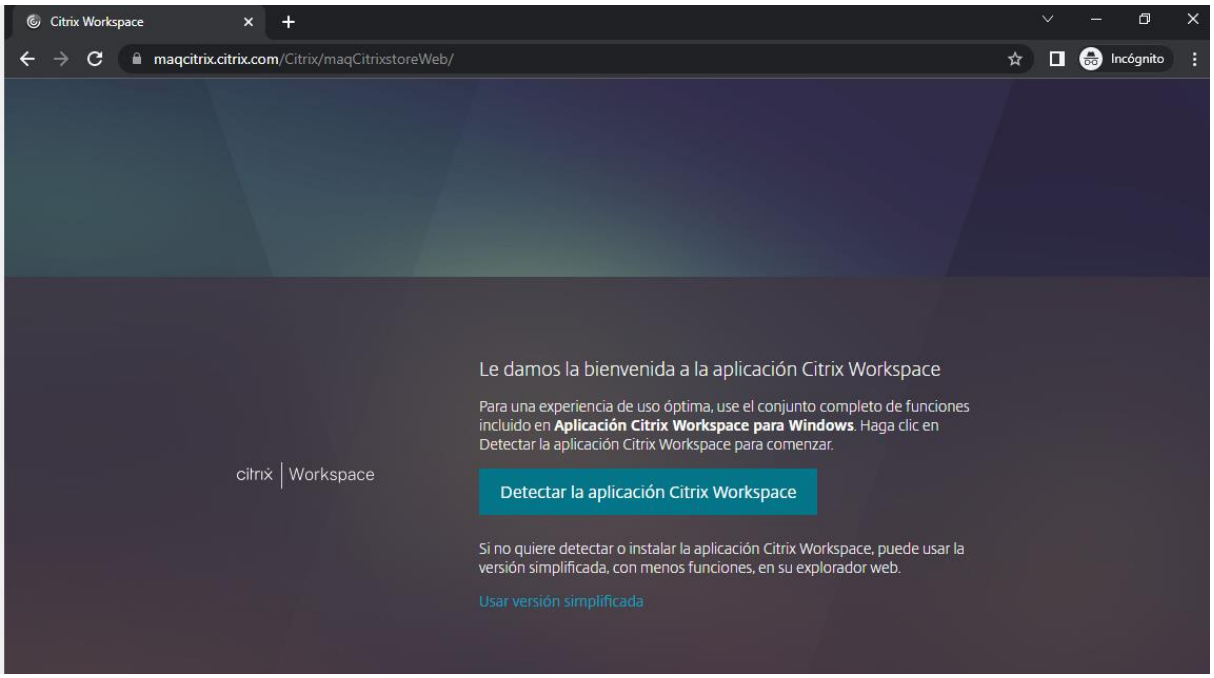


Figura 43: Resultado de la publicacion y creacion en el storeFront

Hasta este punto, se puede conectar a la aplicación desplegada mediante la web, y desde el propio servidor. A continuación, se detallarán los pasos seguidos para la configuración de lo necesario para que los usuarios de dentro del dominio puedan acceder a las aplicaciones desplegadas. Por lo tanto:

1. En primer lugar, y poder conectarse a las mismas mediante conexiones HTTPS, conexiones requeridas indispensablemente, se configurará de la siguiente manera el storeFront:
    a. En primer lugar, se especificará que, para el acceso a las misma, el acceso sea mediante Usuario y contraseña proporcionados por el Active Directoty, y se especificará que se pueda usar para la conectividad las Smart Card, Yubikey:

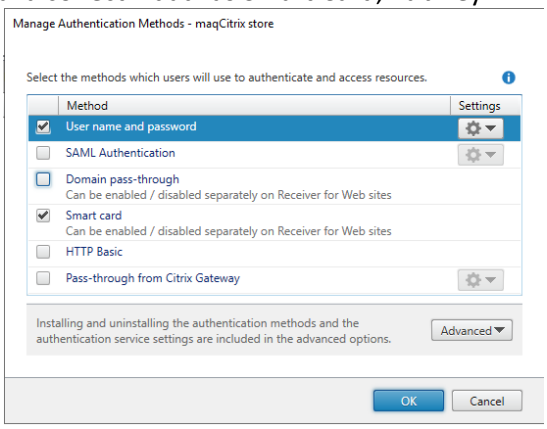


Figura 44: Modos de autenticación disponibles para Citrix. Selección de los comentados

    b. El receiver que se va usar, será el que se ha configurado de manera automática a la hora de crear el site, y que tendrá el siguiente path:

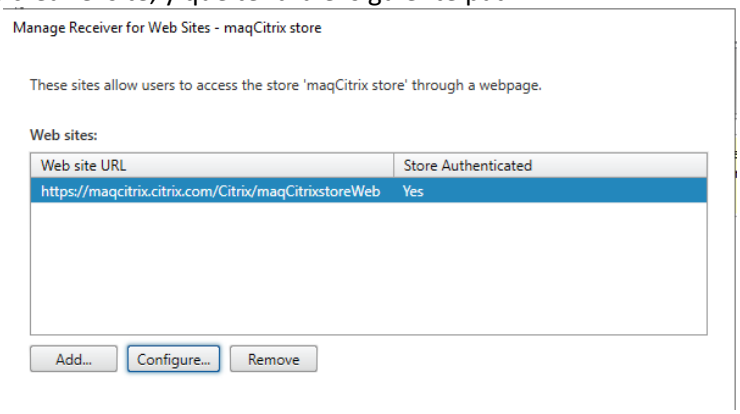


Figura 45: Path del receiver que se va a usar

Por lo que la configuración del StoreFront quedara de la siguiente forma:

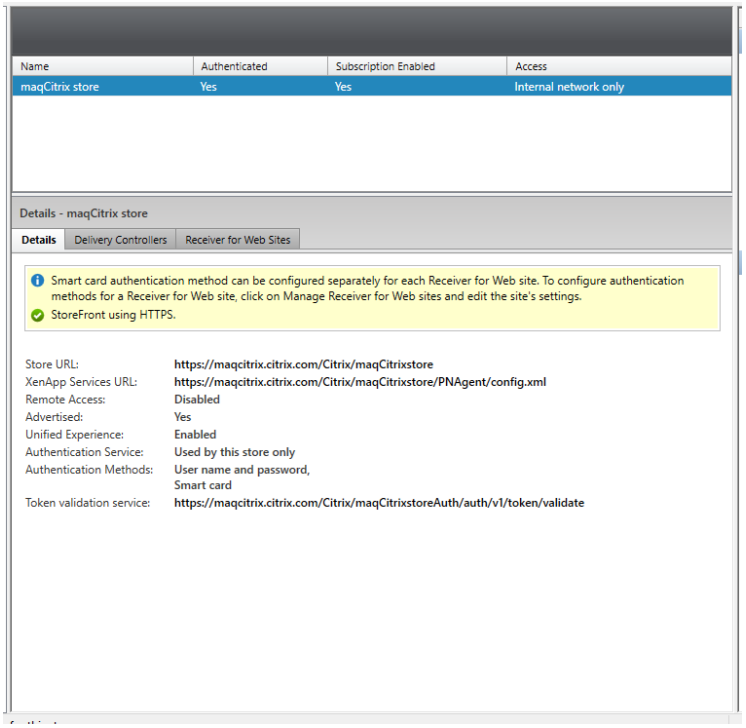


Figura 46: Resultado de la configuración realizada en la consola de Citrix StoreFront. Características y opciones elegidas de lo desplegado

Tras dejarlo así, y aunque se especifique ahí que la conectividad a la misma será mediante HTTPS, los certificados expedidos por la autoridad certificadora no están configurados todavía, por lo que al inicio de sesión y tras la correspondiente autenticación, esta sesión te dará un error y te la cerrará directamente. Por lo que ahora, se vuelve al DC, para la correspondiente configuración de dicha autoridad certificadora (https://www.youtube.com/watch?v=R4mrcju5wec):

1. Si antes en pasos anteriores se ha comentado que para la instalación de este servidor debería estar instalada la opción de Certificate Authorities, a continuación, se demostraran las características extras que se han instalado para dicho rol:

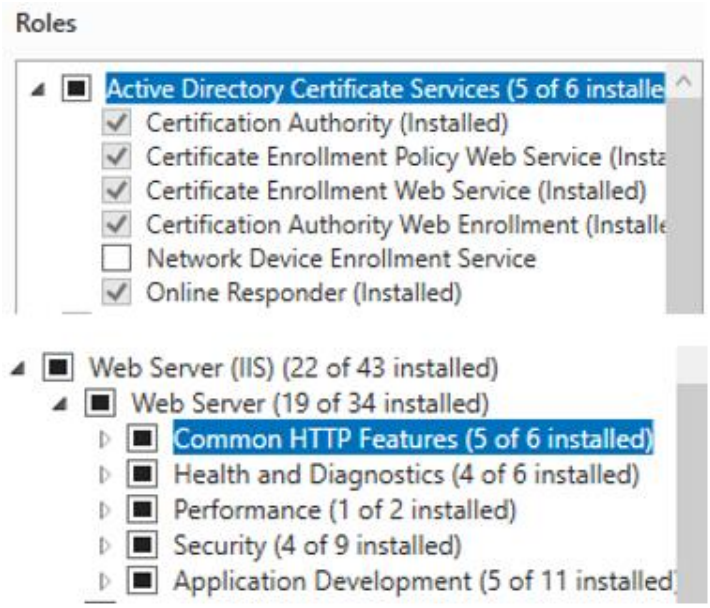


Figura 47: Roles adicionales para la configuración de la entidad ceritifcadora

2. Posteriormente, y tras el reinicio del servidor, se seleccionará la configuración "standalone" de la misma. La otra opción es "Enterprise", pero como es un entorno de desarrollo, no es necesario dicha configuración. Esta última necesita de conexión online para su uso, mientras que la standalone no tiene por qué.
3. Como se tiene instalado ya un Active Directory, se especificará que esta CA, sea root. Una CA root, es una autoridad certificadora que tiene en sí una o más roots de confianza. Como es un entorno de pruebas, se configurará de esta forma puesto que no tiene de salida a Internet, y no necesita de ser contrastada por ninguna otra.
4. En este siguiente paso, se seleccionará que se cree una private key de manera automática, puesto que es la primera CA, de dentro del dominio.
5. Se seleccionará ahora el método de encriptación para la firma de los certificados expedidos. Sera SHA256, comentado previamente.
6. Se especificará, además, un common name, common name suffix, etc. para la misma, que en este caso es el siguiente:

   CN = citrix-DCPARACITRIX-CA-1
   DC = citrix
   DC = com
7. Posteriormente, se especificará el periodo de validez de los certificados expedidos, que será en este caso de 5 años. Este periodo tan largo de validez se pone así puesto que es un entorno de desarrollo. En caso de tener más operatividad, se especificaría menos tiempo.
8. Se especificará además como se van a autenticar dichos usuarios al recibir este certificado, que será por Active Directoy.
9. Se especifica el certificado que se va a usar para las comunicaciones SSL. Es decir, el creado.

Por lo que, en este punto ya está configurado esta autoridad certificadora. Ahora bien, estos certificados son necesarios crearlos para la autentificación mediante la yubikey a las aplicaciones desplegadas, por lo que los siguientes pasos irán encaminados en la configuración de lo necesario para primeramente para poder realizar la conexiones SSL necesarias. Y dos, para la correspondiente configuración de la Yubikey. Por lo tanto:

1. En la aplicación “Certification Authority”, en la carpeta de “Certificate Templates”, se hace click con el botón derecho, y se selecciona “New” y después “Certificate Template to Issue”:

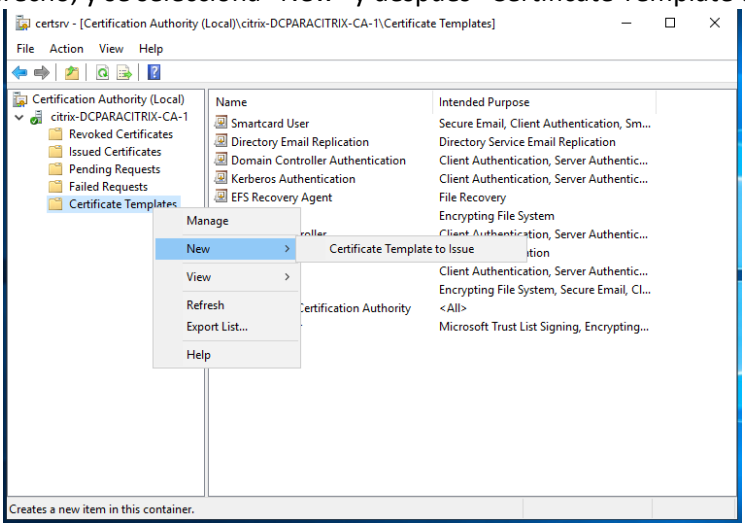


Figura 48: Creación template para tarjeta Yubikey. Pasos

2. Se elige la opción de Smartcard User. Esta opción permitirá crear un certificado para la Smart card.
3. A continuación, se elige la opción de certificates templates, botón derecho y administrar lo que abrirá la siguiente consola:

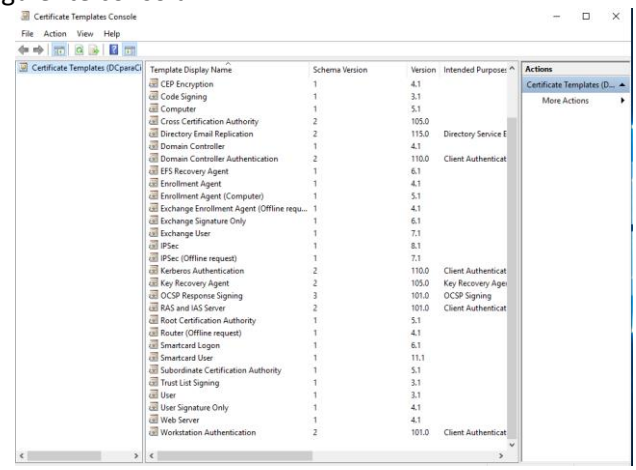


Figura 49: Ventana de la opcion "Certificate Templates". Opciones disponibles y la opción seleccionada

En ella, se clicka sobre la opción de “Smart Card user” y se va a la pestaña de seguridad. En ella, se especifica el permiso de “Enroll”, lo que hará que los certificados expedidos de este tipo, se asigne al dispositivo el permiso de enrolarse al dominio.

4. En este punto se configurará el certificado que expeditará el controlador de dominio. Para la autentificación de los usuarios con smartcard, el controlador de dominio tiene que expeditar certificados del tipo X509, para trabajar con el protocolo de Kerberos. Para ello, se abre la consola de administración de Microsoft, y una vez abierto el Snapin de los certificados, con la correspondiente opción de “Computer Account”:

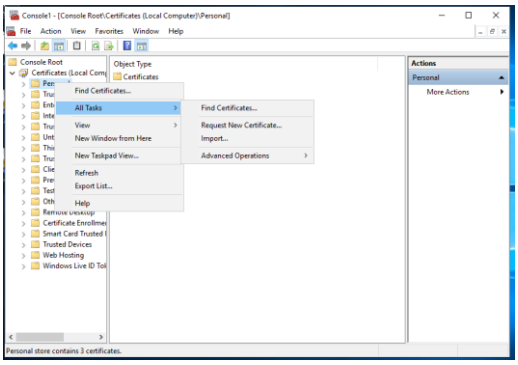


Figura 50: Selección de la opción sobre las acciones requeridas en el controlado de dominio. Configuración del tipo de certificados a propagar

Se selecciona la opción de “Request New Certificate”.

5. Posteriormente en la ventana que aparece, en los tipos de certificados que se pueden pedir, se elegiran las opciones de “Domain Controller” y “Domain Controller Authentication”. Con estas opciones lo que se consigue es poder realizar autenticaciones mediante la yubikey, además de incluso poder usarse estos certificados para comunicaciones IPsec.
6. Se le dará al botón de “Enroll” para dejar la configuración lista y proseguir con la configuración de la Yubikey:

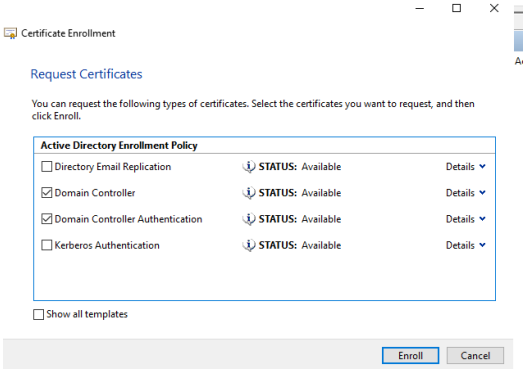


Figura 51: Tipos de certificados disponibles para pedir. Opciones elegidas "Domain Controller" y "Domain Controller Authentication"

A partir de este punto comenzaría la configuración de la Yubikey. Para ello, se ha tenido que descargar e instalar el programa “Yubikey NEO Manager”, programa que permitirá configurar y además agregar a la yubikey el correspondiente certificado para la configuración. En ocasiones, y más en las primeras versiones de la yubikey 4, la interfaz PIV y el modo CCID viene deshabilitado. Para este caso de uso, hay que verificar que este habilitado, y si no lo está habilitarlo. Esto son para lo que sirven [47]:

- PIV: Personal Identity Verification. Permite realizar operaciones de firma/descrifrado RRSA o ECC usando una clave privada almacenada en la propia Yubikeky, a través de interfaces comunes como PKCS#11-
- CCID: hace referencia a los elementos de la propia tarjeta inteligente e incluye los programas como OpenPGP, PIV (como justo encima) y YubiOATH, todos programas de autentificación.

Por lo tanto:

1. Tras abrir el programa Yubikey NEO Manager e insertar la yubikey, pedirá que se inserte una clave PIN. Se inserta uno nuevo.
2. Tras esto, se selecciona el botón de “Certificate” y se le da a la opción de “Generate new key”, disponible en la pestaña de “Authentication”.
3. Para la generación de esta private key, se elegirá el algoritmo RSA 2048, con el output de “CSR”. Además, se especificará el usuario a quien se expeditará este certificado, en el caso de entorno, ander@citrix.com. Una vez hecho esto, se guardará en una carpeta el .csr que se genera.
4. Esta operación se ha realizado en la maquina host que contiene el hypervisor, por lo que el .csr se copia al controlador de dominio.
5. Se inicia sesión en el servidor con el mismo usuario que se ha especificado antes, es decir el usuario ander@citrix.com. Y se ejecuta por la termina del servidor el siguiente comando, en la carpeta donde se encuentre el .csr:

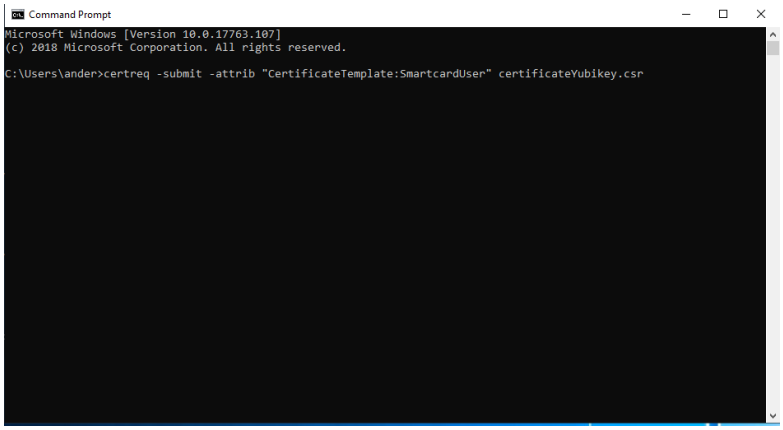


Figura 52: cmd del controlador de dominio. Generación ceritificado .csr

6. Tras insertar el comando especificado por la terminal, en la ventana que aparece se selecciona autoridad certificadora creada previamente y se guarda el archivo .csr saliente.
7. Dicho archivo se vuelve a llevar al ordenador anfitrión y se importa a la yubikey, en la pestaña de “Authentication”.
8. Tras esto, se habilita el servicio de “Smart card Service” para poder usarla.
9. Para la correspondiente verificación, se habilitará también en la configuración del RPD, la opción de Smart Card.

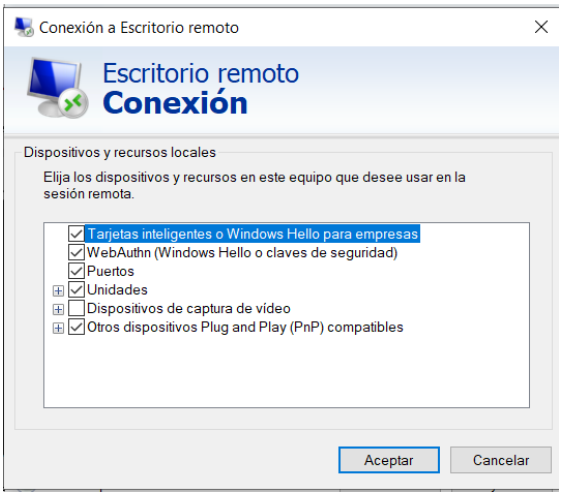


Figura 53: Dispositivos elegidos para la conexión mediante Yubikey al servidor por protocolo RDP

10. Además, para que de manera automática se configure de esta forma los ordenadores de los usuarios, se utilizara una política de grupo definida ya por defecto en la que, una vez reiniciado el ordenador del usuario, se configurara lo comentado.

Tras esto, se configurará el servidor Citrix para la conectividad mediante HTTPS.

11. Para ello, en el servidor citrix se abre la consola de configuración de Microsoft, se añade el Snapin de Certificates, con cuenta de ordenador local y se hace un request de un certificado.
12. En la pantalla que aparece, se selecciona generar un certificado del tipo “Computer”. Se guarda para tenerlo rápidamente a disposición.
13. Se abre la consola IIS. En ella, se selecciona la “Default Web Site” del servidor, que en este caso es la de Citrix, y se le da al botón del panel derecho de “Bindings”:

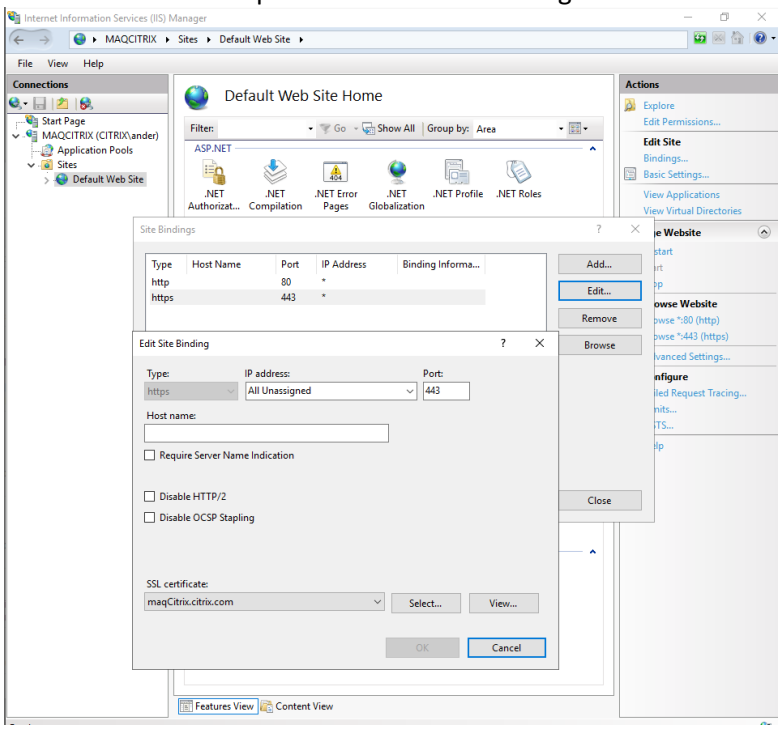


Figura 54: Consola IIS Manager. Selección del certificado configurado previamente. Especificación puerto de conexión

14. Se seleccioan el puerto 443, y se deja la IP Address, “All Unassigned” para que se acepte cualquier tipo de IP entrante. Se selecciona posteriormente el certificado que se creó anteriormente en el MMC de Windows.
15. Se va “SSL Settings” y se selecciona que solamente se permite conexiones entrantes SSL:

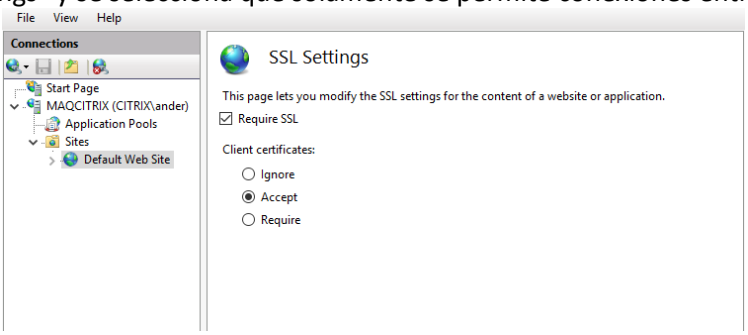


Figura 55: Configuración SSL. Requerimiento obligatorio para conectividad

Se realiza de esta manera puesto que hoy en día es un requisito indispensable de seguridad en la navegación por internet, y además porque de no utilizar el puerto 443, Citrix no te permite usar las correspondientes aplicaciones desplegadas (requerimiento indispensable).

En este punto, la correspondiente configuración por la parte de sistemas/servidores ya estaría realizada. Ahora quedaría por la parte de los dispositivos de los usuarios. En este entorno de pruebas, los servidores y el equipo anfitrión se ha enrolado previamente a la creación de la autoridad certificadora y por la tanto al estar dentro ya del dominio, los navegadores no tienen instalada la autoridad certificadora del controlador del dominio. Por lo tanto, los pasos siguientes van encaminados a comentar dicha configuración en los navegadores. Para ello, se realizan los siguientes pasos:

1. Se abre el navegador que se tenga instalado en el ordenador. En el caso del proyecto, se ha realizado la prueba en varios, pero a modo de configuración se comentará lo realizado en el de Chrome.
2. En él, se escribe por el buscador el siguiente path, teniendo en cuenta que, al estar dentro del dominio, conectividad a la autoridad Certificadora se realizara por el puerto 80:

   http://192.168.1.45/certsrv

   Donde aparecerá la siguiente página web, después de haber iniciado la sesión con un usuario de dentro del dominio:

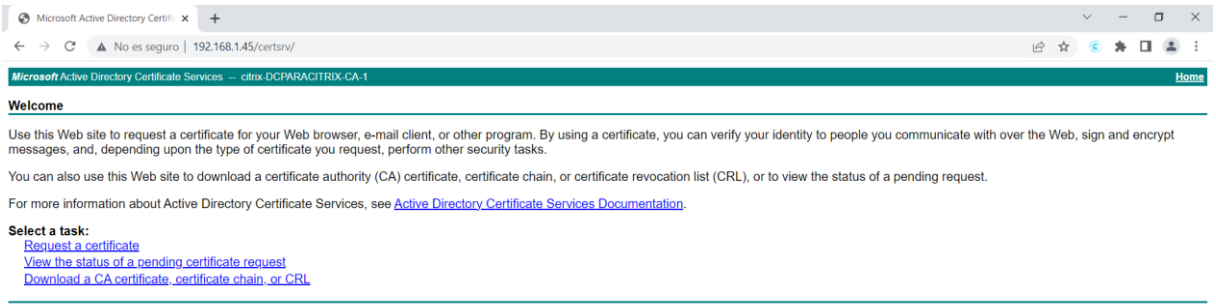


Figura 56: Web para la obtención de los certificados a instalar en el buscador de turno

3. En esas opciones, se elegirá la opción de "Download a CA certificate" y aparecerá la siguiente ventana en el navegador:

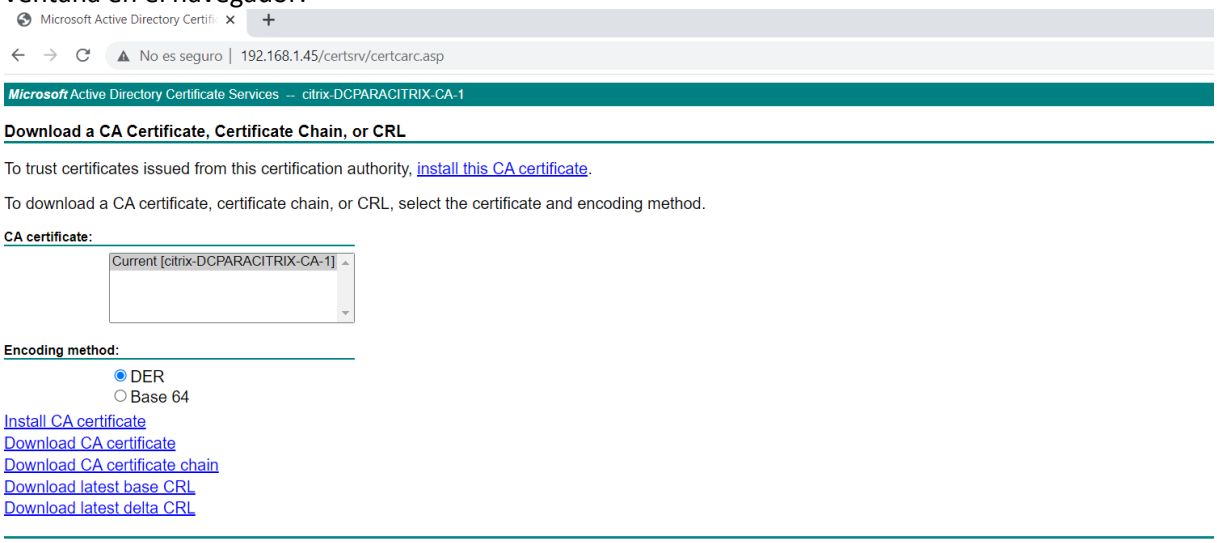


Figura 57: Segunda parte. Descarga e instalación del certificado expedido

4. Se selecciona "Install CA certificate". Se instalará. De no ser así, automáticamente se descarga en la carpeta seleccionada para ello, un archivo .cer necesario para la importación del mismo.
5. Posteriormente, hay que ir al apartado de configuración del navegador y se filtra la búsqueda por "Certificados".
6. En él, en la pestaña de "Entidades de certificación raíz de confianza", se importa el descargado hasta ahora y posteriormente se verifica que la conectividad se realiza mediante una conexión HTTPS:

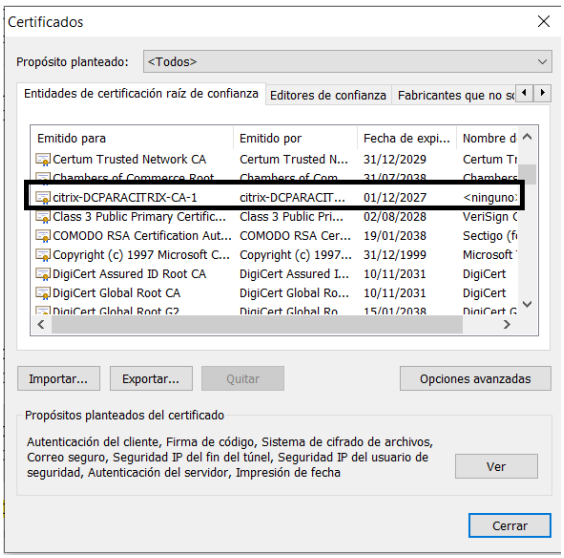


Figura 58: Resultado de la importación del certificado pedido. Añadido a "Entidades de certificación raíz de confianza"

Por lo que estaría ya configurado de manera correcta el navegador para poder usar las correspondientes aplicaciones. Y con la tarjeta inteligente insertada, se abriría el menú de selección de correspondiente de las apps:

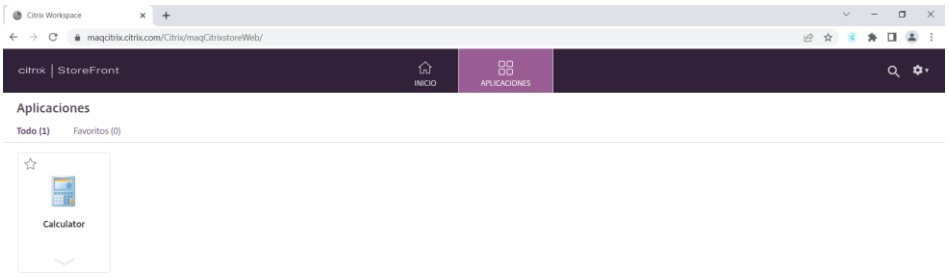


Figura 59: Tras inicio de sesión, menú de las aplicaciones disponible para este usuario

Siendo esta calculadora la que está desplegada por Citrix. De manera similar, tendría que ser para los navegadores de Firefox, Opera, etc.

Por otro lado, hay otra forma de acceder a las aplicaciones desplegadas, y es a través del programa cliente propio de Citrix. Este programa se puede descargar desde su página web de manera gratuita e instalarlo en el ordenador. En el ordenador anfitrión, se ha instalado el Citrix Receiver, donde en la configuración del mismo se ha especificado la URL donde se alberga las aplicaciones que le corresponden a la cuenta, que en este caso es el siguiente:

maqcitrix.citrix.com

Es decir, el FQDN del servidor Citrix. El programa cliente tiene el mismo proceso de autenticación, es decir, te pide también tener insertada la yubikey y saber el PIN de la misma para poder acceder a las aplicaciones. A continuación, se muestra el menú de selección del programa, que es igual que el que se puede acceder mediante la web:

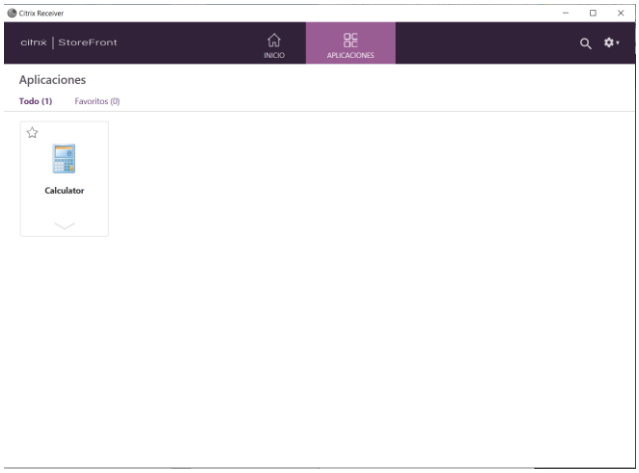


Figura 60: Menú de selección del programa cliente Citrix Receiver

Si que es cierto, que para la prueba del programa cliente en ordenador con el sistema operativo de Ubuntu, se ha utilizado el programa cliente Citrix Workspace, que no es más que las ultimas versión del Citrix Receiver. No obstante, para su configuración, hay que añadir un paso extra puesto que, en Ubuntu, no se reconocen de manera automática la conectividad hacia el servidor mediante el protocolo SSL. Para ello, hay que alojar el certificado de manera manual o si se quiere automatizar mediante script en lenguaje bash, los siguientes pasos por la terminal de Ubuntu:

1. A la hora de descargar el programa cliente Citrix Workspace, descargar una versión igual o posterior a la 22.04, ya que de descarga una anterior, incurre en el arranque de sistema operativo y lo deja inutilizable, con lo que conlleva. Importante esta elección, puesto que durante el proyecto pensando que versiones inferiores serían más estables que las ultimas, se ha tenido que recuperar el sistema operativo desde un backup.
2. Se mete en dominio el dispositivo que corre el sistema operativo de Ubuntu.
3. Se instala las dependencias necesarias para poder utilizar la yubikey en el dispositivo.
4. Se descarga el certificado correspondiente de la autoridad certificadora, ya sea para el navegador de turno y para el propio Citrix Workspace.
5. Dicho certificado, una vez instalado con la correspondiente extensión en el navegador, de no ser .crt se convertirá a dicha extensión:
   openssl x509 -inform DER -in certificate.cer -out certificate.crt
6. Se instala siguiendo los pasos del instalador el programa cliente descargado previamente. Cabe comentar aquí que hay diferentes formas de descargarlo e instalarse, pero se recomienda encarecidamente que sea el paquete full, y no el tarball que la propia página web del proveedor te permite, puesto que el tarball da bastantes problemas de dependencias.
7. E instalar dicho certificado dentro de la carpeta que se comenta a continuación. Esto permitirá que el programa cliente realice una conexión mediante SSL hacia el servidor de citrix y así

poder usar las aplicaciones desplegadas. Se recuerda en este punto el motivo del por qué hay que hacerlo así: Se recuerda que, en apartados anteriores se ha configurado el servidor IIS para única y exclusivamente permitir peticiones HTTPS. De lo contrario, se cierra la conexión. Por lo tanto, se han metido los siguientes comandos por terminal para alojar el certificado en la carpeta:

sudo su

cp /home/ander/Downloads/certificate.crt /opt/Citrix/ICAClient/keystore/cacerts/

ls -la /opt/Citrix/ICAClient/keystore/cacerts/ | grep "certificate.crt" → Para verificar que este certificado tiene los permisos 644 y el propietario del mismo es el usuario root.

/opt/Citrix/ICAClient/util/ctx_rehash

8. Una vez abierto el programa cliente, especificar también la cuenta de almacenamiento de las aplicaciones desplegadas por el servidor Citrix:

maqcitrix.citrix.com

Por lo que, de esta forma, queda completamente configurado también el programa cliente Citrix Workspace para realizar las mismas funciones que el Citrix Receiver. De esta forma quedan demostrada la configuración de ambos tipos de versiones. En este punto, se da por finalizadas las respectivas configuraciones de los diferentes entornos.

Metodología seguida en el desarrollo del trabajo

Tareas

Las tareas realizadas durante el proyecto son las siguientes diferenciadas, además en la tabla con el respectivo computo de hora:

1. Búsqueda la información necesaria para la puesta en marcha del proyecto. Obtención de la documentación sobre la yubikey. la raspberry pi y sobre los servicios que se van a configurar.
2. Realización de las configuraciones sobre la desencriptación utilizando la yubikey. A continuación, se detalla el orden en el que se ha procedido:
    a. Configuración del sistema operativo en la raspberry.
    b. Encriptación de la partición root con LUKS
    c. Configuración de initramfs
    d. Configuración yubikey para enrolar la partición con la yubikey
    e. Enrolado con la yubikey
    f. Pruebas de funcionamiento
3. Configuración para la autentificación con yubikey en conexiones SSH:
    a. Configuración en servidor
    b. Configuración en la parte de cliente

   c. Pruebas de conectividad
4. Configuración del escalado de privilegios a root mediante la yubikey.
5. Configuración de la autentificación con yubikey en Citrix
   a. Creación del entorno de pruebas. Entorno virtualizado en ordenador.
   b. Publicación de una aplicación
   c. Configuración de Citrix para la autentificación mediante Smart cards.
   d. Pruebas al respecto
6. Documentación y redacción del proyecto.

A continuación, se detalla el número de horas invertido para cada fase en la tabla comentada:

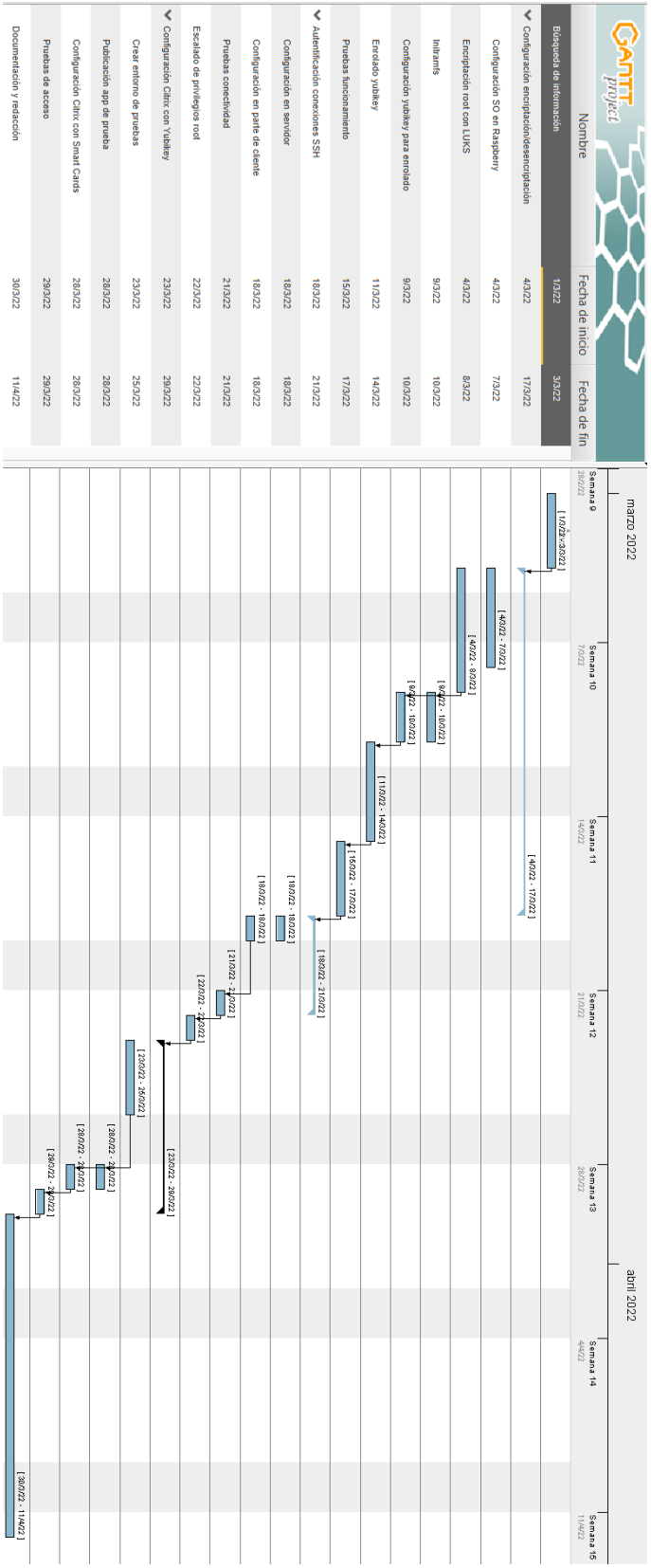
Tareas	Subtarea	Horas invertidas por subtarea	Horas invertidas
1. Búsqueda de información	.	.	24
2. Configuración encriptación/desencriptación	a. Configuración SO en Raspberry	14	92
	b. Encriptación root con LUKS	21	
	c. Initramfs	12	
	d. Configuración yubikey para enrolado	15	
	e. Enrolado yubikey	10	
	f. Pruebas funcionamiento	20	
3. Autentificación conexiones SSH	a. Configuración en servidor	8	16
	b. Configuración en parte de cliente	4	
	c. Pruebas conectividad	4	
4. Escalado de privilegios root	.	.	8
5. Configuración Citrix con Yubikey	a. Crear entorno de pruebas	25	40
	b. Publicación app de prueba	5	
	c. Configuración Citrix con Smart Cards	5	
	d. Pruebas de acceso	5	
6. Documentación y redacción	.	.	72
Total, horas			252

Tabla 3: Tareas realizadas en el proyecto. Horas computadas a cada. Horas totales

Diagrama de Gantt

En la siguiente figura se mostrará el diagrama de Gantt donde se verá reflejado el trabajo realizado hasta la fecha. Si bien es cierto que en la anterior tabla se especifican las horas invertidas en cada tarea y subtarea, se va a tomar cada 8 horas invertidas como un día trabajado en dicha tarea, puesto que el diagrama de Gantt especificará por días la resolución completa del proyecto. Por lo tanto (próxima hoja):

Figura 61: Diagrama de Gantt



Descripción de los resultados

Este apartado incluirá las capturas de las pruebas realizadas correspondientes a los entornos configurados previamente. Se empezará por enseñar las capturas correspondientes a la autenticación para el arranque de sistema de la raspberry, posteriormente lo referido a la escalada de privilegios a usuario root, después de la conectividad SSH y finalmente el acceso a las aplicaciones de Citrix, tanto mediante web, como mediante el Citrix Receiver y el Citrix Workspace. Por lo tanto:

Autenticación para el arranque de sistema de la raspberry

Primeramente, se ha tomado una imagen de los cables conectados a la raspberry y donde se ha insertado la yubikey. Dichos cables conectan a la raspberry un ratón, un teclado, una pantalla mediante el cable HDMI y la fuente de alimentación. Además, para iniciar el arranque de la raspberry, se ha dejado insertada la yubikey.

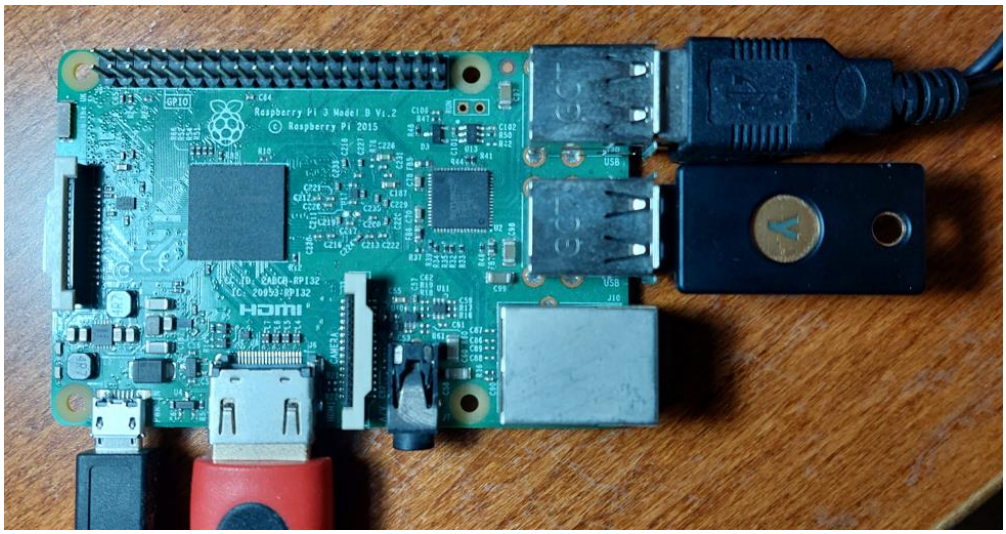


Figura 62: Disposición y cableado de la raspberry

A continuación, se ha arrancado la raspberry. En un primer intento de arranque no se ha presionado el botón de la yubikey cuando ha empezado a parpadear, para demostrar que realmente es necesaria dicha interacción para arrancar la raspberry. Y tras el fallo se ha presionado.

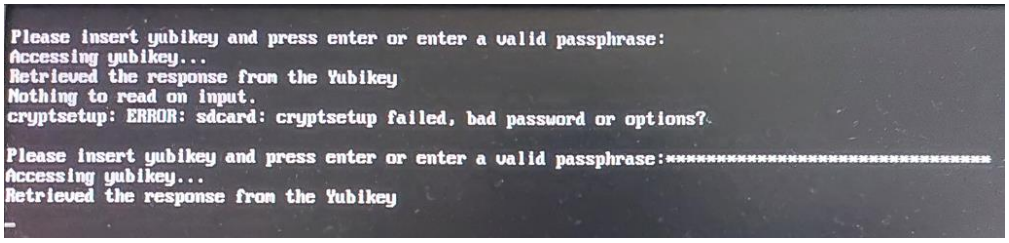


Figura 63: Autenticación previa al arranque del sistema raspbian. Terminal initramfs

Como se ve, se autentifica contra la yubikey tras presionar el botón. Luego si la contraseña es correcta, se accede a la misma, se obtiene la contraseña para desencriptar la clave LUKS y si todo es correcto, se desencripta el volumen.

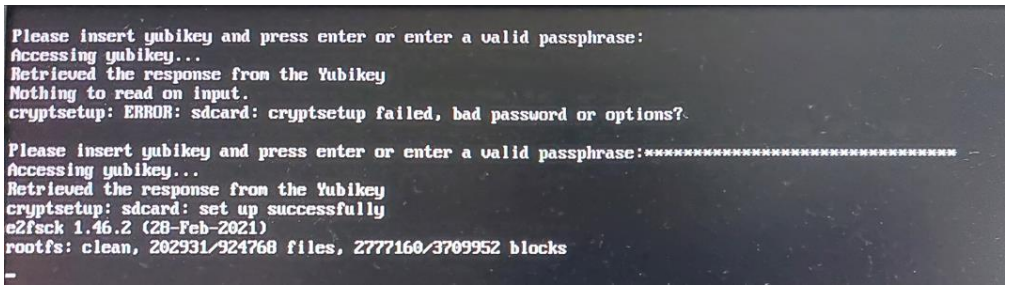


Figura 64: Resultado de la autenticación correcta mediante la yubikey

Tras desencriptarse la partición root del sistema, se continua con los siguientes pasos del arranque tal y como se ve a continuación:

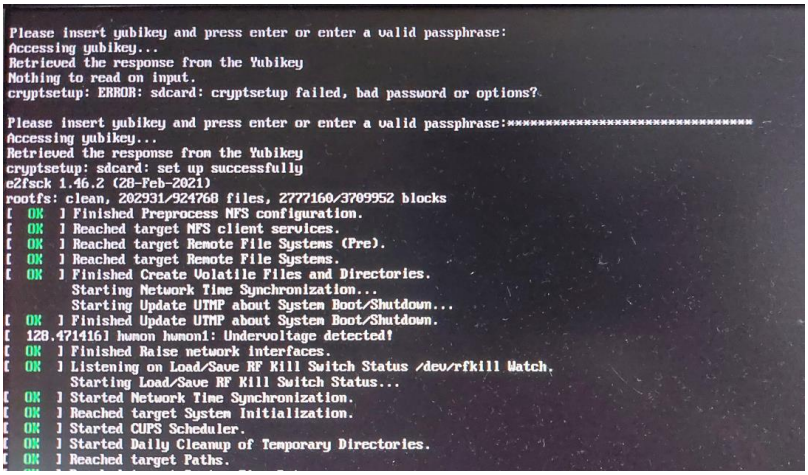


Figura 65: Continuación de los procesos de arranque tras la desencriptación

Y como no ha habido ningún fallo en dichos procesos tras la autenticación, por pantalla se ve el fondo de pantalla del sistema operativo.

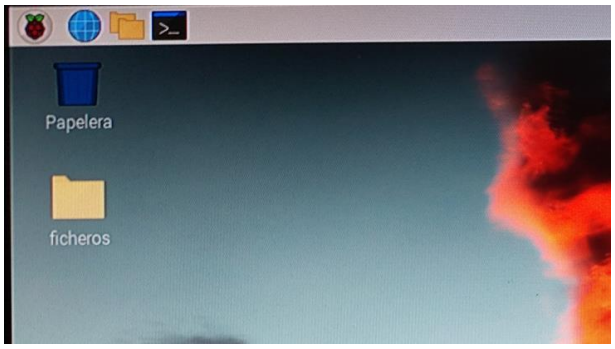


Figura 66: Tras inicio correcto, escritorio del sistema operativo albergado en la sdcard

Escalada de privilegios a usuario root

Tras el inicio correcto, se pasa a probar el siguiente objetivo. Se abre una terminal, se quita la yubikey y se prueba a ser usuario root. Como se ve en la imagen, aunque se inserte la contraseña del usuario, al no estar la yubikey, se niega el acceso a dicha escalada.

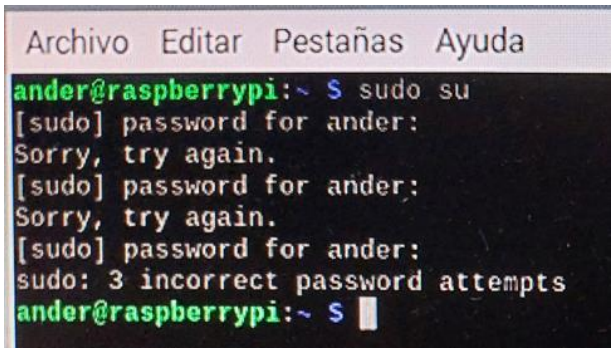


Figura 67: Intentos fallidos sin yubikey insertada

Y al de 3 intentos, se sale del proceso de autenticación.

Posteriormente, se inserta la yubikey y se prueba. En este caso, se insertan las correspondientes contraseñas, la yubikey empieza a parpadear, se presiona el botón y consigue ser finalmente usuario root.

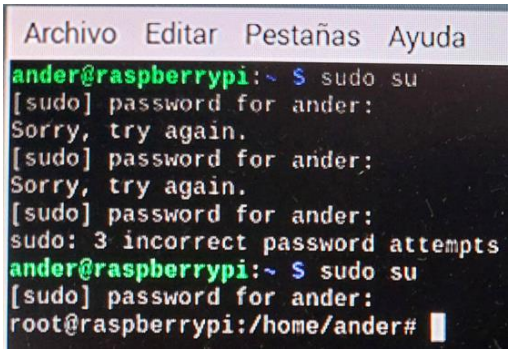


Figura 68: Autenticación con yubikey insertada. Cambio a usuario root

Ahora, se prueba con un comando a modo de ejemplo. En este, actualizar los paquetes que hay en el sistema. Se repite proceso. Se prueba sin la yubikey, error.

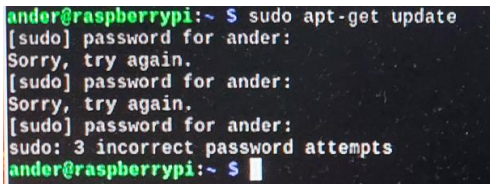


Figura 69: Intento de comandos fallido sin yubikey

Se inserta y se autentica, se prosigue con la correspondiente ejecución del comando.

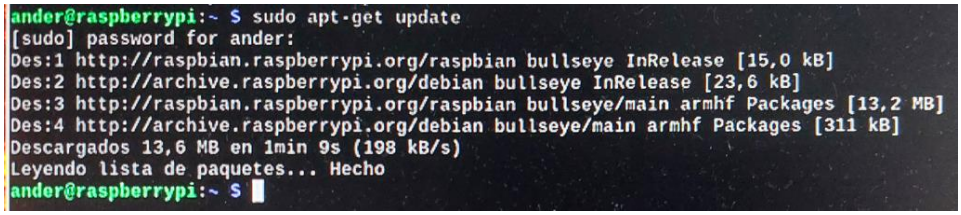


Figura 70: Resultado de comando satisfactorio con yubikey insertada

Recordar aquí que esta petición de autenticación se hace siempre que se inserte dicho comando sudo, porque en apartados anteriores así se ha configurado. De lo contrario, con hacer la autenticación una vez, suficiente. Lo que permite agregar cierta seguridad extra.

Conectividad SSH más autentificación con Yubikey

A continuación, se prueba a realizar una conexión SSH desde un ordenador externo. Primeramente, se arranca el servicio ssh, para poder recibir dichas conexiones. De manera extra, se recomienda que dichas conexiones este permitidas o filtradas por el firewall que corresponda, y simplemente permitir conexiones entrantes desde IPs de confianza.

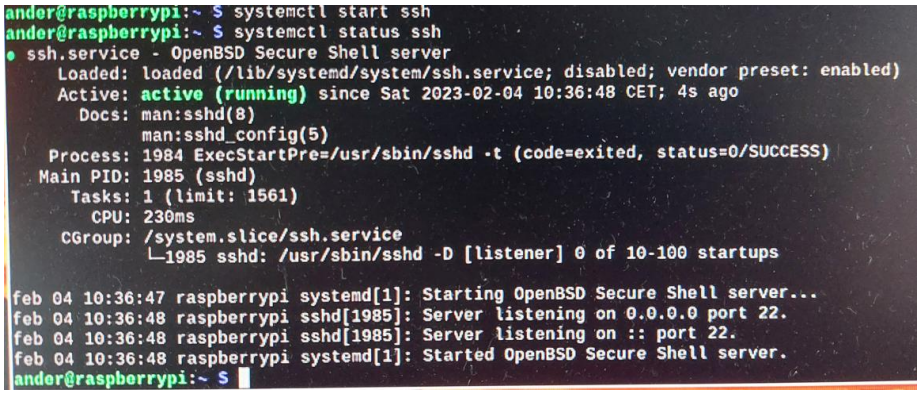


Figura 71: Arranque servicio SSH por terminal raspberry

Posteriormente, se prueba la conexión mediante SSH sin la yubikey insertada. De la misma forma que en anteriores ocasiones, el acceso no está permitido. Por lo que al de 3 intentos, la conexión o el intento de autenticación, se cierra.

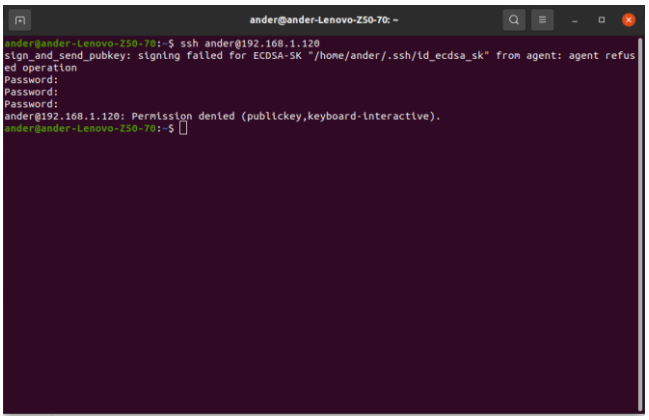


Figura 72: Intento fallido de conexión SSH por terminal de Ubuntu. Yubikey sin insertar

Tras dicha prueba, se inserta la yubikey. Se intenta conectar y efectivamente, la yubikey parpadea, se presiona el botón y la conexión se realiza de manera satisfactoria, como se aprecia en la siguiente imagen:

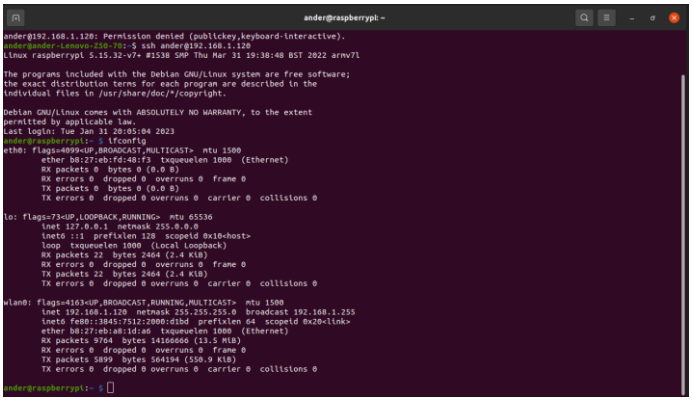


Figura 73: Intento satisfactorio conexión SSH por terminal Ubuntu. Yubikey insertada

No obstante, y tal y como se ha comentado en el apartado de la correspondiente configuración de este caso, si por lo que sea este usuario decide obtener permiso de usuario root, no puede. Por lo que necesariamente, el administrador está obligado a configurar la correspondiente ACL en la raspberry.

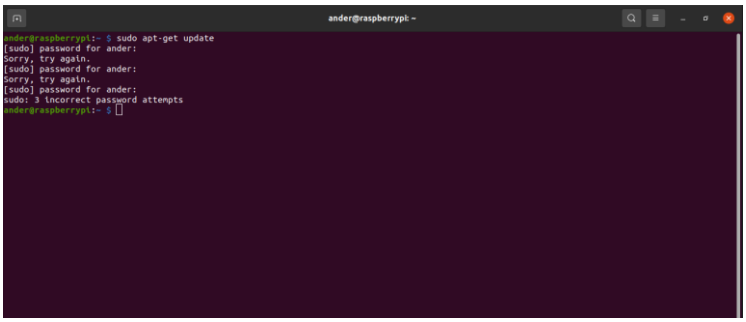


Figura 74: Intento de comando usuario root fallido mediante conexión SSH

Acceso a aplicaciones Citrix. Web más Citrix Receiver y Citrix Workspace

En este caso de uso, se diferencian tres subapartados: acceso mediante web, acceso mediante Citrix Receiver y acceso mediante Citrix Workspace.

Web

En este caso, y como se puede apreciar en la imagen, nada más insertar la URL, se pide aceptar el certificado expedido por la autoridad certificadora instalada en el dominio.

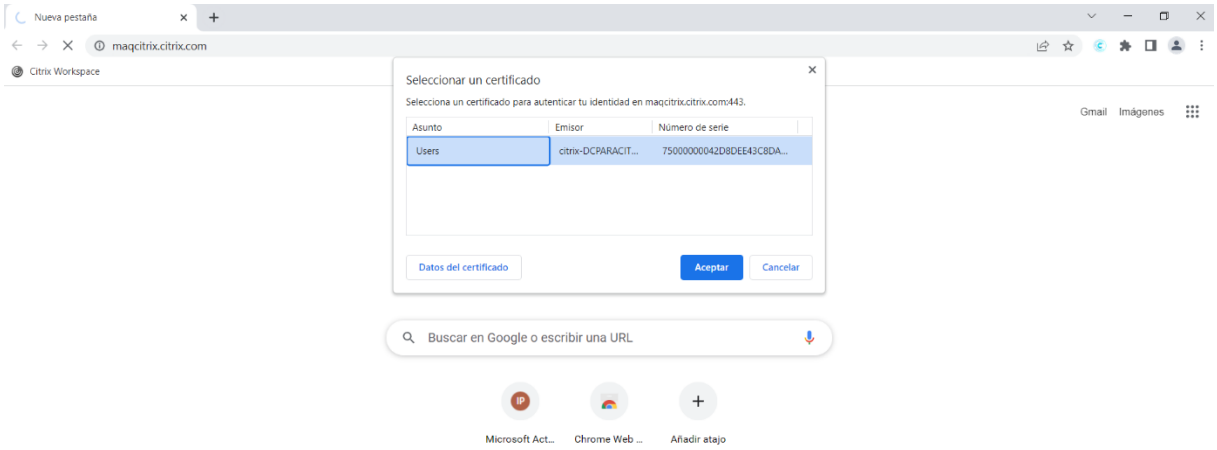


Figura 75: Petición certificado a usar para la conexión al servidor Citrix mediante navegador

Y como no se tiene en un principio la yubikey insertada, pide que se inserte la correspondiente:

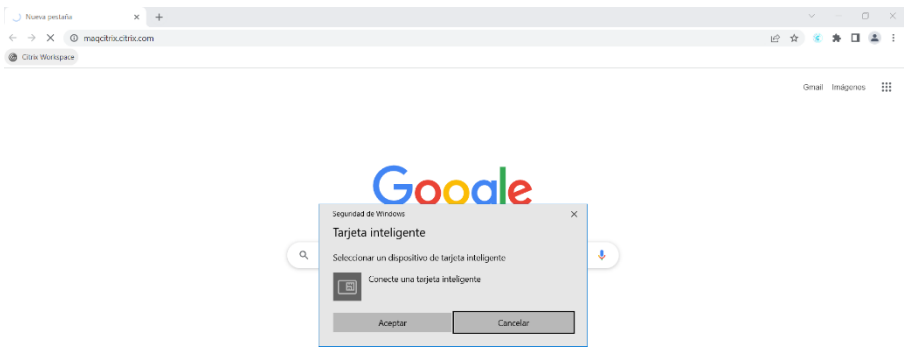


Figura 76. Yubikey sin estar insertada, petición de insertarla. Requerimiento obligatorio

Tras insertarla, se detecta:

Figura 77: Tras inserción, detección automática de la yubikey disponible

Se acepta, y pide insertar el correspondiente PIN para la autenticación. Cabe comentar que solamente es PIN, porque el certificado insertado en la yubikey para el Active Directory es de confianza y esta automáticamente asociado a un usuario de dentro del Active Directory

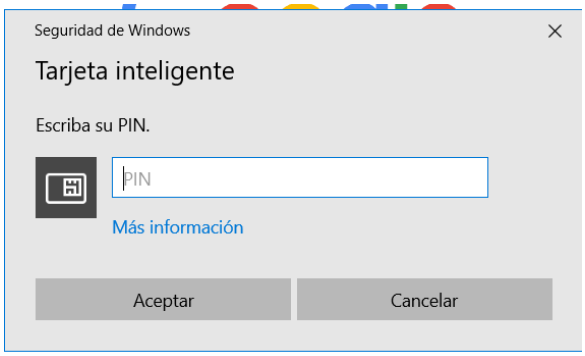


Figura 78: Autenticación e inserción de PIN para acceso. Requerimiento indispensable

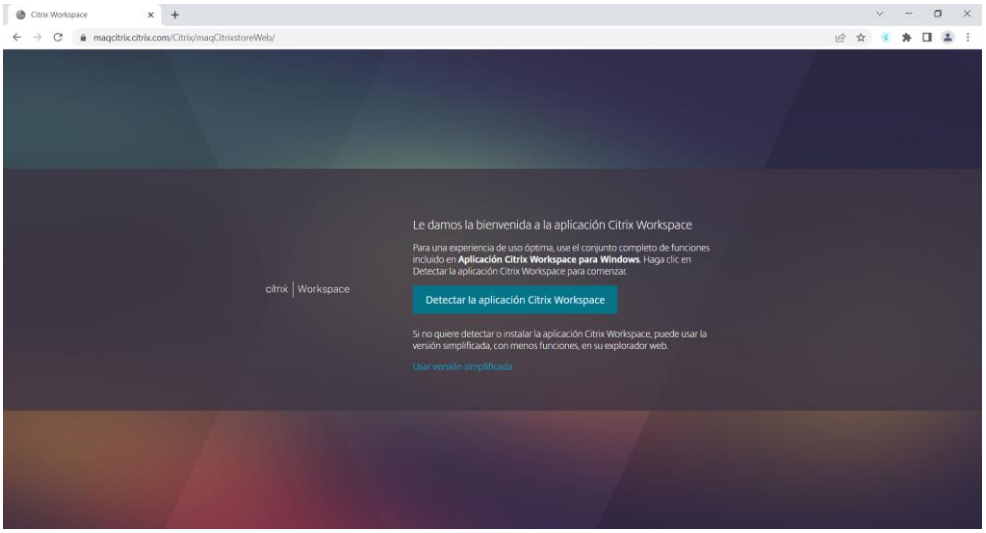


Figura 79: Resultado tras autenticación. Página web de Citrix StoreFront

Se le da a detectar la aplicación, y posteriormente, aparece el menú de las aplicaciones desplegadas exclusivamente para este usuario:

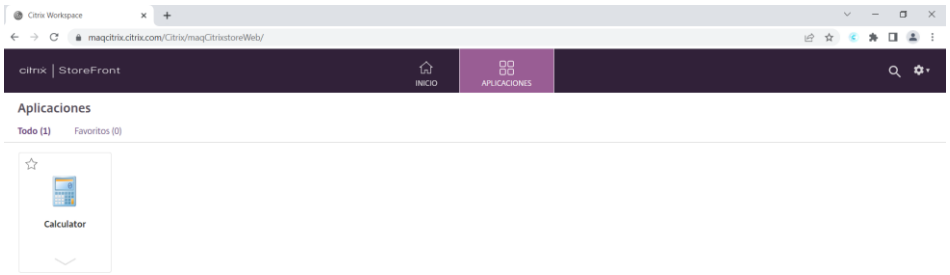


Figura 80: Menú de las aplicaciones desplegadas para este usuario

Citrix Receiver

A continuación, se utilizará el Citrix Receiver. Primeramente, no se tiene la yubikey conectada, por lo que la pide:

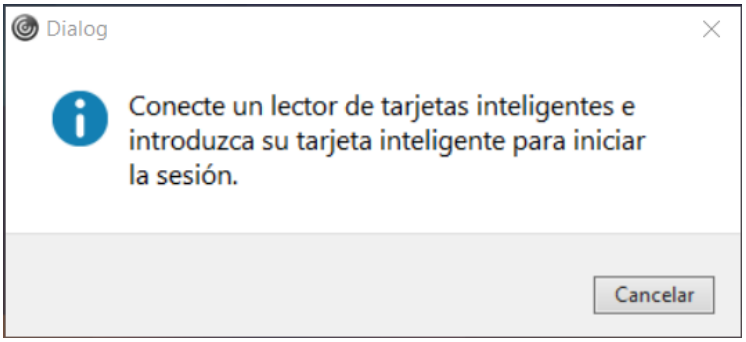


Figura 81: Programa cliente Citrix Receiver. Petición de inserción yubikey. Requerimiento indispensable

Tras insertarla, pide el PIN de la misma:

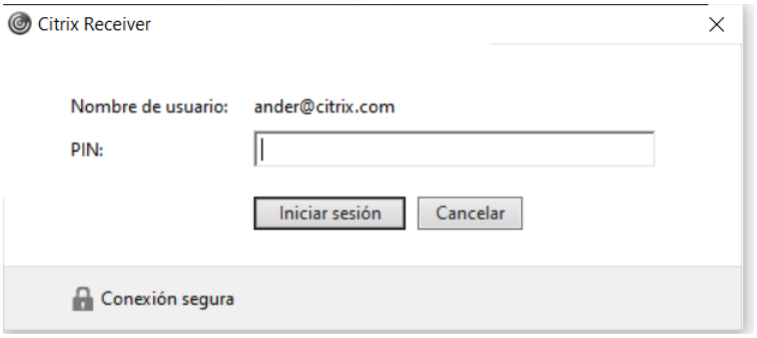


Figura 82: Tras inserción, autenticación necesaria

Y tras la correcta autenticación, aparecerá por el programa cliente el menú que le corresponde a este usuario:

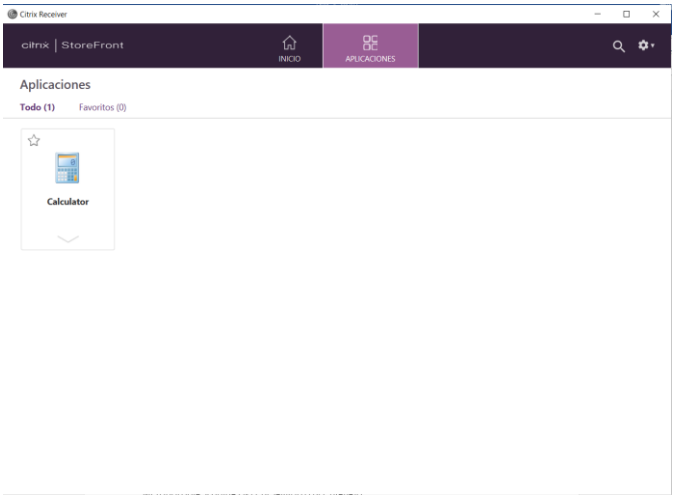


Figura 83: Menú de las aplicaciones desplegadas para este usuario

Cabe destacar, que cuando se selecciona abrir una de las aplicaciones desplegadas, pide realizar de nuevo la autenticación correspondiente puesto que abre una conexión RDP contra el servidor de Citrix. Así que siendo el usuario correspondiente se insertan el usuario y contraseña o se utiliza la yubikey para dicha autenticación.

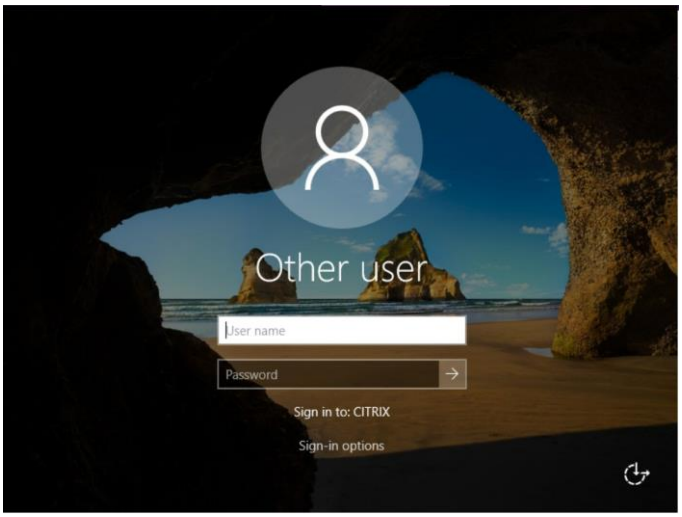


Figura 84: Conexión mediante RDP. Inicio de sesión necesaria. Dos opciones: Usuario-Contraseña, Yubikey

En caso de querer hacer la autenticación con la yubikey:

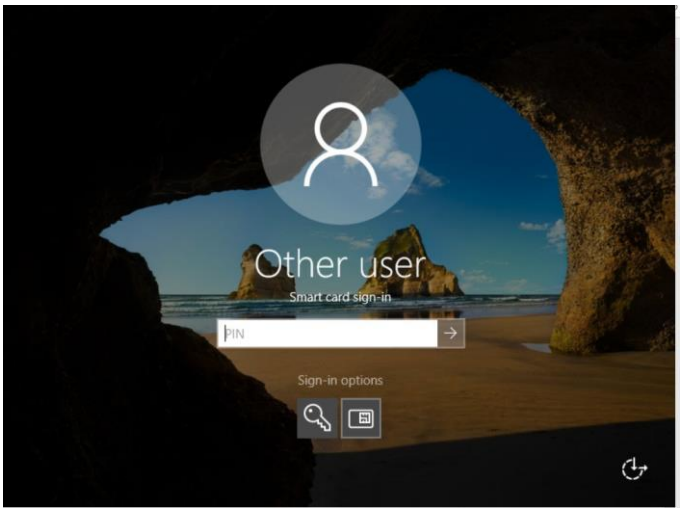


Figura 85: Autenticación mediante Yubikey

Y tras ello, se abre la correspondiente aplicación:

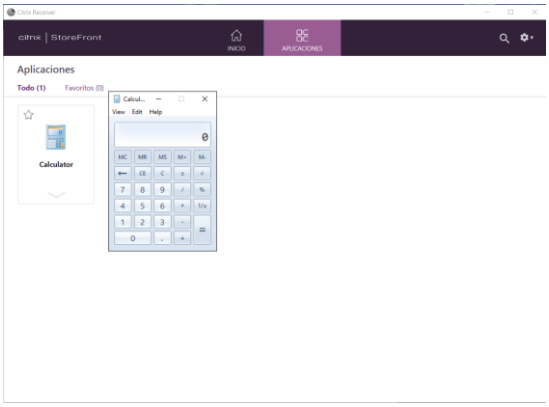


Figura 86: Aplicación abierta y lista para usarse tras autenticación correcta

Citrix Workspace

Y en Ubuntu se procederá con el mismo procedimiento solo que por lo comentado en el apartado de la configuración de la solución, se usará la última versión del Citrix Workspace. Por lo que sin la yubikey, se pide:

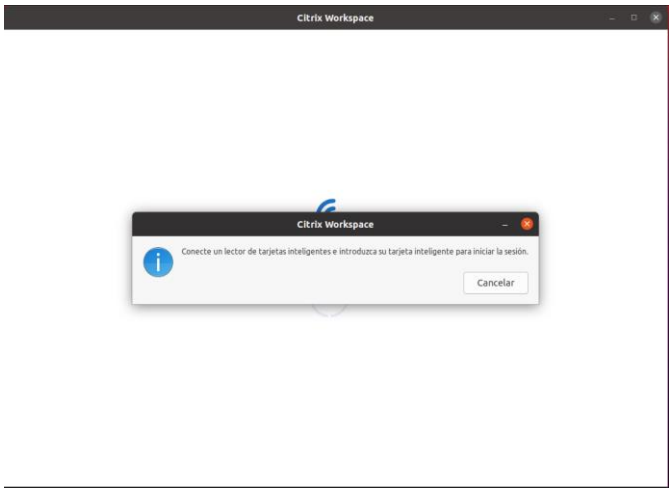


Figura 87: Ordenador con sistema operativo Ubuntu. Programa cliente Citrix Workspace. Petición para insertar Yubikey

Se autentica con la yubikey:

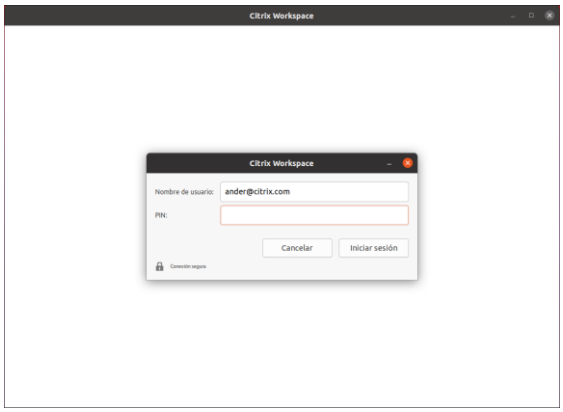


Figura 88: Autenticación requerida con la yubikey insertada

Se procede a ejecutar la correspondiente conexión al servidor Citrix:

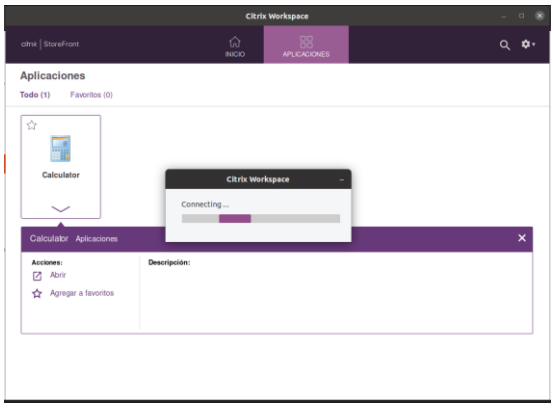


Figura 89: Esperando respuesta del servidor tras conexión

Se obtiene el menú de las aplicaciones desplegadas:

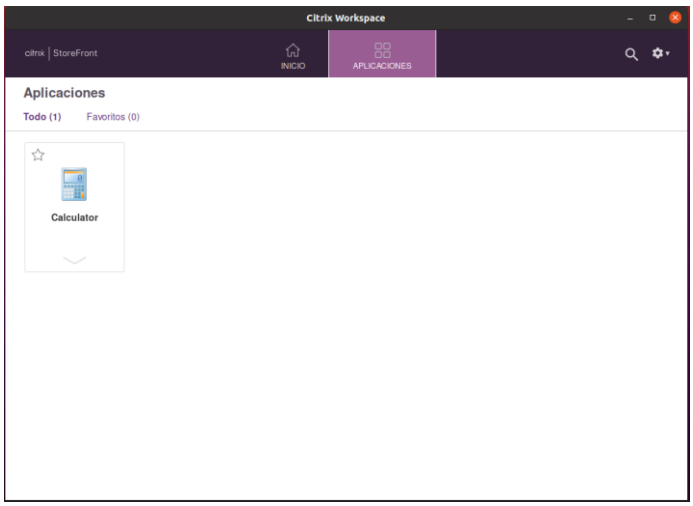


Figura 90: Menú de aplicaciones desplegadas para este usuario. Programa cliente Citrix Worksapce

Y se abre la aplicación.

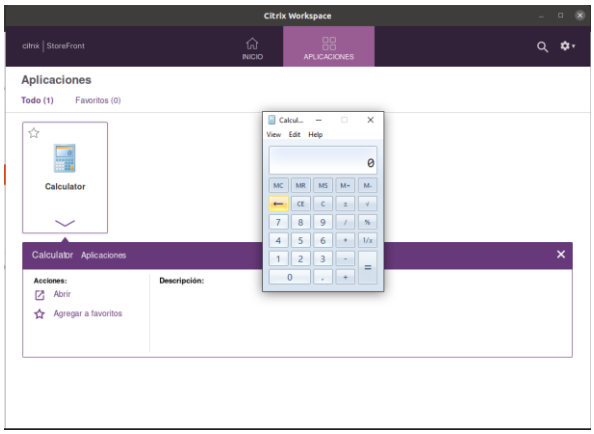


Figura 91: Aplicación abierta. Protocolo usado en este caso para ello HDX

Cabe destacar que en esta situación en vez de usarse el protocolo RDP para la conexión a la aplicación, se ha utilizado la tecnología HDX, grupo de tecnologías propias de Citrix. Un conjunto de tecnologías patentadas, construido sobre el protocolo de comunicación remota ICA. Arquitectura alternativa nativa de Citrix al protocolo RDP. Protocolo diseñado por Citrix para entregar específicamente datos de visualización grafica a los usuarios.

Y finalizando con este apartado, se enseña aquí lo registrado en el Citrix Studio cuando el usuario que está utilizando la conexión comentada arriba. Es decir, se muestra por la consola que usuario se ha

conectado, a que servidor, a que catálogo de máquinas, como está la sesión, etc. De hecho, si se hace click sobre dicha línea, aparecerán más detalles al respecto.

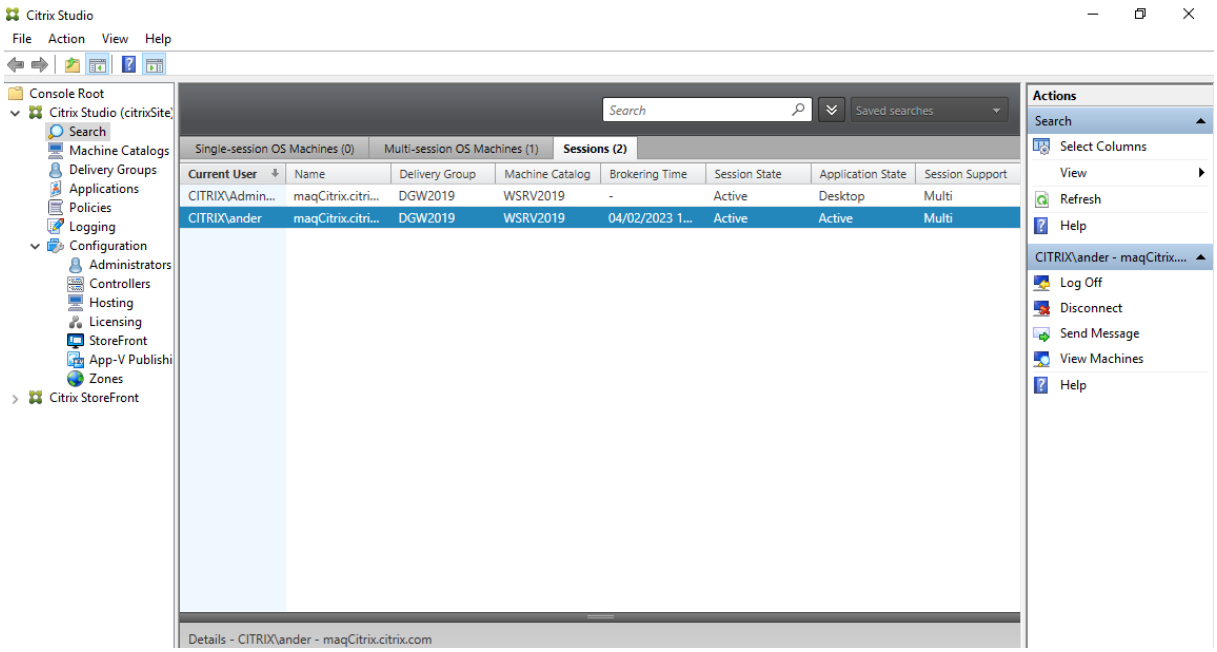


Figura 92: Estado sobre las sesiones activas en Citrix Studio

Aspectos económicos

El desglose de gastos para este proyecto se dividirá en tres subapartados, donde en el primero se tendra en cuenta el coste de los técnicos por el trabajo realizado. Cabe destacar que en todos ellos se especificará el porqué de los comentados. En el segundo se tendrán en cuenta las amortizaciones por los equipos utilizados y por último el gasto por hardware, software y mantenimiento tenidos. Por lo tanto:

1. Costes de los participantes en el proyecto

Como se observa en la siguiente tabla, se ha dividido en 4 columnas. Dichas columnas hacen referencia a las personas que han participado en el proyecto, a las horas realizadas por los mismos, donde el trabajo realizado se hace de manera paralela. Es decir, si uno de los integrantes del equipo realiza la configuración del SO de la raspberry, el otro se encargaba de la configuración del entorno de Citrix, de ahí que el cómputo total de horas sea el mismo que en el apartado anterior. En la siguiente columna se muestra el coste por hora de cada uno de los participantes y en la siguiente el total pagado por las horas trabajadas:

Participante (€/hora trabajada)	Horas	€/hora	Total (€-s)
Director del proyecto (Ing. Senior)	50	60	3000
Técnico de sistemas (Ing. Junior, 2 personas)	202	30	6060
Total	252	0	9060

Tabla 4: Horas dedicadas por trabajador. Precio hora trabajado

Los €/hora reflejados en la tercera columna son los euros/hora calculados de los salarios medios a nivel nacional teniendo en cuenta la categoría a la que corresponden las respectivas personas. Y como se puede observar el total de cada columna se ve reflejado en amarillo.

2. Amortizaciones de los equipos utilizados

Como se observa en la siguiente tabla, para trabajar los participantes dispondrán de los equipos de alta gama ofrecidos por la compañía Lenovo (ThinkPad T15g Gen 2 (15" Intel)), además de una raspberry Pi 3B v1.2 (node Edge) y una Yubico Yubikey 4. Los precios totales del producto son los que se ven en la tabla. Las horas de uso son una estimación aproximada de uso anual. La vida útil de los productos será de 5 años, y las horas laborables estarán calculadas en un año laboral normal. Por lo tanto, la tabla quedaría así:

Producto	Precio total (€)	Horas uso	Vida útil (Años)	Horas laborables
ThinkPad T15g Gen 2 (15" Intel)	3090,24	960	5	1944
raspberry Pi 3B v1.2	46,51	960	5	1900
Yubico Yubikey 4	54,99	350	5	400

Tabla 5: Precio elementos usados y horas de uso

Por lo que respecta a las amortizaciones de los productos durante esos 5 años de vida útil se les aplicará el máximo coeficiente lineal desde el momento en el que se adquirieron dichos productos. Por lo tanto, las cuotas anuales de amortización para cada equipo quedarían así:

	PC	Raspberry	Yubico
Cuota amort. Anual (€)	618,048	9,302	10,998

Tabla 6: Cuota de amortización anual asignado a cada producto

Es decir, el valor de la compra por el 20% que es el máximo coeficiente lineal. Y las amortizaciones para dichos equipos de este modo:

- Ordenadores:

Ejercicio	Valor amortizable	Cuota anual	Amortización acumulada	Pendiente
2022	3090,24	618,048	618,048	2472,192
2023	2472,192	618,048	1236,096	1854,144
2024	1854,144	618,048	1854,144	1236,096
2025	1236,096	618,048	2472,192	618,048
2026	618,048	618,048	3090,24	0

Tabla 7: Amortización de los ordenadores usados. Periodo de tiempo, 5 años

- Raspberry:

Ejercicio	Valor amortizable	Cuota anual	Amortización acumulada	Pendiente
2022	46,51	9,302	9,302	37,208
2023	37,208	9,302	18,604	27,906
2024	27,906	9,302	27,906	18,604
2025	18,604	9,302	37,208	9,302
2026	9,302	9,302	46,51	0

Tabla 8: Amortización de la raspberry usada. Periodo de tiempo, 5 años

- Yubico Yubikey:

Ejercicio	Valor amortizable	Cuota anual	Amortización acumulada	Pendiente
2022	54,99	10,998	10,998	43,992
2023	43,992	10,998	21,996	32,994
2024	32,994	10,998	32,994	21,996
2025	21,996	10,998	43,992	10,998
2026	10,998	10,998	54,99	0

Tabla 9: Amortización de la Yubico Yubikey usada. Periodo de tiempo, 5 años

3. Gastos de hardware y software utilizados

Una vez comentados los gastos de los equipos utilizados, en este apartado se hablará de los gastos por software y hardware, respectivamente. En cuanto al software, cabe destacar que para obtenerlo necesariamente tienes que estar registrado como proveedor de Citrix. Esta descarga no requiere un cargo adicional, pero si su uso bajo licenciamiento. Estas licencias, las de Citrix, son anuales y se pagan por packs. Su precio varía en función de si eres socio proveedor del programa o usuario final. En este proyecto, las licencias se han obtenido a través de un socio proveedor, por lo tanto, el coste asociado

aproximado para las 10 licencias obtenidas es el siguiente: precio unitario por licencia expedida por el proveedor 121,76 €. Las licencias de los servidores Windows también son de pago y a tener en cuenta. Estas licencias, siendo el modelo "Essentials" tienen un coste asociado de 501 $ cada una. Y se han obtenido dos, porque un servidor, tal y como se ha especificado en apartados anteriores, se usará para el controlador de dominio del entorno y la otra para albergar el programa Citrix, como servidor de las aplicaciones. Los Citrix Workspace y Citrix Receiver, no son de pago y para la descarga de los mismos no hace falta ser proveedor. La raspberry utilizada tiene un coste asociado de 46,51 € y la yubikey de 54,99 €. El sistema operativo de la raspberry es open-source y por lo tanto no tiene precio.

Por otro lado, para las pruebas realizadas, también se ha necesitado una licencia de Windows Home con un valor de 139$. El sistema operativo de Ubuntu, del mismo modo que el de la raspberry es open-source.

Análisis de rentabilidad

La rentabilidad de lo propuesto, si bien es cierto que supone un gasto por mantenimiento de los sistemas operativos, y el correspondiente soporte por el uso de la solución, es una medida de seguridad que toda empresa debería realizar para evitar "sustos" que supongan a una empresa la perdida por el servicio, en caso de estar en entornos de producción o por el robo de identidad de identidad de los administradores y lo que supondría la pérdida total de los datos albergados, en caso de que el atacante encriptase la información de toda la empresa, o la información que albergan los servidores utilizados. Cuantificar la rentabilidad de la solución en términos monetarios es difícil, y sería más bien una inversión en seguridad, disponibilidad y, probablemente de ahorro en licencias por el uso compartido de servicios expuestos en Citrix, y no tener que pagar por cada una de las licencias usadas por cada uno de los usuarios de la empresa. Un ejemplo de este último caso sería, por ejemplo, el programa Visio, de Microsoft. Este programa tiene un coste asociado de licencia de 479,00 €. Por lo que de esta forma solo pagas por una licencia, en vez de una por cada usuario. Es más. El hecho de usar la solución propuesta en este proyecto, no es solo sale rentable a la propia empresa que la implante, sino que además da un plus de seguridad y reputación a la misma, lo que puede resultar muy beneficioso para la marca de la propia empresa, o venderlo como servicio "premium" a tus clientes, por ejemplo, si eres un proveedor de servicios IT. De hecho, la adopción de este tipo de medidas ha aumentado considerablemente y las cuotas de mercado que están alcanzando este tipo de solución hace que, en un futuro no muy lejano, haga aumentar la rentabilidad de la empresa en caso de vender la solución como se ha comentado anteriormente, como servicio.

Otra cosa es que debido a la crisis mundial de hardware informático que hay actualmente por la falta de suministros, hagan que el precio para obtener las yubikeys aumente de manera considerada y que ese elevado precio, como socio proveedor de la solución haga que se reduzcan los márgenes de beneficio por la venta del servicio. Cosa a tener en cuenta si su cartera de clientes es pequeña, los clientes que administra también son pequeños y sus presupuestos en soluciones IT no son demasiado grandes.

Aun así, como inversión para pasar certificaciones de seguridad, certificaciones que actualmente se están pidiendo como requisito mínimo para poder ser proveedor de productos a mayoristas, véase el caso de una empresa que suministre un producto en concreto, por ejemplo, a otra empresa más grande de automoción, genera rentabilidad. Lo que a la postre, hace que tu negocio si exclusivamente se basa en esto último, haga que se mantenga como un negocio rentable.

Por lo tanto, se da por hecho que la rentabilidad de la solución propuesta en este proyecto queda garantizada. Si no de manera notoria numéricamente, si en materia de ciberseguridad, confianza e imagen de marca para la empresa que trabaje con ella.

Conclusiones

Tras la realización de este proyecto a quedado latente que agregar una capa de seguridad extra en cuanto a los sistemas de autenticación y gestión de la identidad es posible gracias a tecnologías como las cripto tarjetas, pero más en concreto con la Yubikey. No solamente es posible llegar a encriptar particiones de root enteras en raspberrys y no permitir ni si quiera el arranque de las mismas sin la yubikey, sino que también se pueden utilizar para entornos de trabajo virtualizados, como puede ser Citrix. Además, reduce de manera más que notable el acceso a recursos sin autorización e incluso ataques de dicho tipo de accesos mediante Internet, como ha quedado visto en caso de caso de las conexiones SSH.

De hecho, este tipo de cripto tarjetas no depende de dispositivos, como es el caso del uso de móviles como tecnologías MFA que necesiten de estar permanente conectados a Intenet o incluso necesiten de estar encendidos.

Por lo que es sin duda alguna una tecnología más recomendable para las empresas de hoy en día, no solamente por la seguridad que aporta esta tecnología, sino además de la reputación que le pueda llegar a dar a la misma empresa. Lo que a la postre, se puede perfectamente traducir en más beneficios para la misma. O al menos estar seguros de que, aunque ninguna empresa este exenta de ataques, se reduzca la superficie de ataque a la misma.

Y si se tuviera que comentar alguna mejora al respecto partiendo de la base de este proyecto sería la automatización de todas las configuraciones comentadas previamente, dirigido especialmente a entornos empresariales donde la configuración manual sería casi imposible y que conllevaría grandes cantidades de tiempo, como por ejemplo más de 30-40 usuarios dentro de la misma empresa.

Referencias

[1] https://medium.com/@tilaklodha/google-authenticator-and-how-it-works-2933a4ece8c2. Tilak Lodha

[2], [3] https://www.protectimus.com/blog/totp-algorithm-explained/. Maxim Oliynyk on Jun 24, 2020

[4] https://www.techtarget.com/searchsecurity/definition/Google-Authenticator#:~:text=Google%20Authenticator%20is%20a%20mobile,masquerade%20as%20an%20authorized%20user. Ivy Wigmore, Google Authenticator. December of 2014

[5] https://www.protectimus.com/blog/detailed-information-on-data-signing/. Denis Shokotko, March of 2015

[6] https://www.protectimus.com/blog/ocra-algorithm-explained/. Anna, Jun 2017

[7] https://authy.com

[8] https://www.binghamton.edu/its/two-fa/authy.html#:~:text=Authy%20is%20an%20application%20that,setting%20up%20and%20configuring%202FA.

[9] https://www.xataka.com/basics/microsoft-authenticator-que-como-funciona. Yubal Fernandez, 3 de febrero 2021

[10] https://learn.microsoft.com/en-us/azure/active-directory/authentication/concept-authentication-methods. 09/08/2022

[11] https://play.google.com/store/apps/details?id=org.fedorahosted.freeotp&pli=1.

[12] https://freeotp.github.io

[13] https://justuseapp.com/en/app/864224575/sophos-authenticator

[14] https://www.authenticatorplus.com

[15] https://blog.lastpass.com/es/2016/03/lastpass-authenticator-facilita-la-autenticacion-de-doble-factor/, LogMeIn Admin, marzo 16, 2016

[16] https://soundlogin.com

[17] https://webdevolutions.blob.core.windows.net/blog/pdf/updated-most-popular-2-factor-authentication-2fa-compared.pdf August 31 of 2016

[18] https://www.imarcgroup.com/otp-hardware-authentication-market

[19] https://www.reportlinker.com/p05997544/OTP-Hardware-Authentication-Market-Global-Industry-Trends-Share-Size-Growth-Opportunity-and-Forecast.html October, 2022

[20] https://www.researchandmarkets.com/reports/5682303/otp-hardware-authentication-market-global

[21] https://www.marketwatch.com/press-release/hardware-otp-token-authentication-market-scope-2023-2028-with-progress-insights-industry-share-geographical-segmentation-emerging-trends-and-key-drivers-to-2028-newst-136-tables-and-data-2022-11-21

[22] https://www.mordorintelligence.com/industry-reports/hardware-otp-token-authentication

[23] https://www.howtogeek.com/785677/best-hardware-security-keys/. Albert Bassile amd Russ Ware. February 2022

[24] https://www.hypr.com/security-encyclopedia/threat-assessment

[25] https://dl.acm.org/doi/pdf/10.1145/3167996.3167997 Sanchari Das, Gianpaolo Russo, Andrew C. Dingman. Jayati Dev, Olivia Kenny, Dr. L. Camp

[26] https://www.amazon.com/-/es/seguridad-YubiKey-verificación-resistente-impermeable/dp/B07HBD71HL?th=1

[27] https://www.ldlc.com/es-es/ficha/PB00405117.html

[28] https://onlykey.io/es/pages/how-it-works Tim Steiner

[29] https://docs.citrix.com/en-us/citrix-virtual-apps-desktops/technical-overview.html. November 2022 Citrix staff

[30] https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard Cinne Bernsteis, Michael Cobb, 2021 September

[31] https://www.cl.cam.ac.uk/~rja14/serpent.html

[32] https://www.techtarget.com/searchsecurity/definition/Twofish#:~:text=What%20is%20Twofish%3F,both%20hardware%20and%20software%20environments. Rahul Awaty, December 2021

[33], [34] http://websites.umich.edu/~x509/ssleay/rfc2144.html C. Adams. May 1997

[35] https://www.techtarget.com/searchsecurity/definition/Electronic-Code-Book#:~:text=Electronic%20Code%20Book%20(ECB)%20is,of%20sequentially%20listed%20message%20blocks. Rahul Awaty, December 2021

[36] https://www.techtarget.com/searchsecurity/definition/cipher-block-chaining#:~:text=Cipher%20block%20chaining%20(CBC)%20is,IV)%20of%20a%20certain%20length.

[37] https://www.researchgate.net/figure/Disk-encryption-via-CBC-and-ESSIV_fig2_305631645 Thomas unterluggauer

[38] https://www.kingston.com/en/blog/data-security/xts-encryption

[39] https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm Baivab Kumar Jena

[40] https://en.bitcoin.it/wiki/RIPEMD-160. June 2014

[41] https://www.kali.org/tools/cryptsetup/#:~:text=Cryptsetup%20provides%20an%20interface%20for,Key%20Setup%20(LUKS)%20support. 2022-nov-23

[42] https://rr-developer.github.io/LUKS-on-Raspberry-Pi/

[43] https://www.howtogeek.com/devops/what-is-busybox-and-where-is-it-used/ James Walker, December 2021.

[44] https://wiki.debian.org/initramfs-tools

[45] https://linux.die.net/man/8/resize2fs Theodore Ts'o

[46] https://www.techtarget.com/whatis/definition/FDISK

[47] https://docs.citrix.com/en-us/storefront/downloads/smart-card-configuration-for-citrix-environments.pdf

Fuentes de las imágenes utilizadas

[1], [2], [3], [4] https://www.protectimus.com/blog/ocra-algorithm-explained/

[5] https://www.protectimus.com/blog/detailed-information-on-data-signing/

[6], [7], [8], [9] https://soundlogin.com

[10], [11] https://www.howtogeek.com/785677/best-hardware-security-keys/

Fuentes de las tablas utilizadas

[1] https://webdevolutions.blob.core.windows.net/blog/pdf/updated-most-popular-2-factor-authentication-2fa-compared.pdf