UNIVERSIDAD INTERNACIONAL DE LA RIOJA

Universidad Internacional de La Rioja
Máster universitario en Seguridad Informática

Biohacking como segundo factor de autenticación

Trabajo Fin de Máster

presentado por: Rodríguez González, Giovanni Yesid

Director: González Pérez, Pablo

Ciudad: Bogotá D.C., Colombia.
Fecha: 14 de febrero de 2019

ÍNDICE

ÍNDICE DE IMÁGENES

Resumen

En un mundo digital que facilita las tareas cotidianas ya que son automatizadas o en línea crea una necesidad de garantizar que la persona que ingresa a las aplicaciones sea la persona autorizada a usar esos recursos, es por esto que en el desarrollo del presente trabajo de fin master se implementa una autenticación de doble factor en búsqueda de incrementar la seguridad en un servicio de ssh de forma que se pueda replicar en ambientes de productivos; como primer factor se implementa una tecnología de token habilitando la segunda verificación luego de pasar el segundo factor, este segundo factor fue elegido un chip NFC luego de un estudio de los diferentes dispositivos biohacking y características de la tecnología que integra, el NFC es configurado de forma que solo pueda ser leído y modificado por un lector autorizado logrando almacenar por medio de etiquetas de seguridad la contraseña y así que este dispositivo forme parte funcional del cuerpo humano.

Palabras Clave: Biohacking, transpondedor, autenticación, identidad digital.

Abstract

In a digital world that facilitates daily tasks as they are automated or online creates a need to ensure that the person who enters the applications is the person authorized to use those resources, in the development of the present document a double factor authentication is implemented in the search to increase the security in an ssh service so that it can be replicated in productive environments; as a first factor a token technology is implemented enabling the second verification after passing the second factor, this second factor was chosen an NFC chip after a study of the different biohacking devices and the characteristics of the technology it integrates, the NFC is configured form that can only be read and modified by an authorized reader, managing to store the password by means of security labels and so that this device forms a functional part of the human body.

Keywords: Biohacking, transponder, authentication, digital identity.

1. Introducción

1.1. Introducción

En la transformación digital que estamos viviendo donde nuestras transacciones bancarias no son en las sucursales si no desde un Smartphone, la educación es desde un laptop con acceso a internet, la comunicación personal es por medio de redes sociales, los pagos de impuesto y hasta tramites de gobierno son en-línea se debe contar con mecanismos que permitan identificar digitalmente las personas para poder confirmar que es quien está ingresando a las aplicaciones o sistemas, lo cual es un reto en la actualidad ya que los diferentes tipos de ataques que van desde robar las credenciales por ingeniera social hasta robar una sesiones establecidas con ataques dirigidos (APTs) con el objetivo de suplantar una identidad, lo anterior ha motivado la evolución de la autenticación que va desde usuario y contraseña hasta mecanismos biométricos que puedan “garantizar” que es el verdadero usuario.

En búsqueda de un método de autenticación robusto y confiable aparecen las combinaciones de diferentes mecanismos con el fin de aumentar las características de seguridad y suplir o remediar las fallas o vulnerabilidades de los otros mecanismos, por lo que vemos usualmente métodos de autenticación como usuario y contraseña con mensajes de texto para tener doble factor de autenticación o el método de autenticación por Single Sing-On en donde con una sola instancia de autenticación con componentes criptográficos robustos se usa para ingresar a múltiples aplicaciones, ambos métodos son más comunes para usuarios en internet o intranet, pero si vemos el ámbito empresarial en el cual esto puede ser más complejo de implementar se puede encontrar el uso de tokens mas el usuario y contraseña en los servidores para que solo sean los administradores quienes tengan el acceso a estos y no un atacante.

También están los mecanismos biométricos que permiten a partir de la huella dactilar, patrón de voz entre otros identificar a una persona, pero desde hace unos años ha comenzado aparecer una tecnología que permite tener en el cuerpo dispositivos para autenticarnos en aplicaciones, sistemas operativos e incluso la integración con accesos físicos como en domótica o la apertura de la puerta con una cerradura digital; esta tecnología permite ser insertada en partes especificas del cuerpo y sus aplicaciones actualmente están siendo exploradas, con este tipo de dispositivos es posible combinarlo con otro mecanismo de autenticación y ser usado en aplicaciones, acceso a servidores en

ambientes empresariales como factor de autenticación, el cual es crítico debido a que sus operaciones de negocio se soportan en servidores con aplicaciones o bases de datos, o usos personales como el encendido de un vehículo, en este documento se desarrollara uno de sus usos y se presentaran las características de seguridad con las que cuentan.

1.2. Justificación

Este trabajo de fin de master pretende desarrollar un piloto en donde será implementado un dispositivo de biohacking como segundo factor de autenticación en un ambiente de laboratorio con el fin de estudiar los diferentes tipos de tecnologías que permiten ser implementadas como mecanismo de autenticación, su arquitectura y características de seguridad y como estas complementa o aumenta la seguridad en búsqueda de conseguir un método de autenticación más robusto.

1.3. Planteamiento del trabajo

Se estudiará el estado actual y dispositivos biohacking que puedan ser implementados como factor de autenticación y desarrollar un piloto simulando un ambiente corporativo en donde el dispositivo permita autenticar un usuario en un sistema con el fin de incrementar la seguridad en la verificación de la identidad digital de los usuarios.

El piloto se desarrollará en un entorno simulado con el fin de recrear el uso en ámbito corporativo ya que tienen mayor necesidad de confirmar que sean las personas autorizadas quienes ingresan a los sistemas core, también son ambientes en donde los despliegues de este tipo soluciones son más complejos por compatibilidad y costos.

1.4. Estructura de la memoria

La estructura del documento está comprendida en los siguientes capítulos:

- **2. Contexto y estado del arte**: Se explicará el contexto de actual de la autenticación y biohacking basados en documentos y/o investigaciones de otros autores, identificando los factores de seguridad más importantes y las vulnerabilidades que más afectan a este tipo de tecnología para de esta forma identificar dispositivo más factible a implementar.

- **3. Objetivos concretos y metodología de trabajo:** En este capítulo se establecerán el objetivo general y objetivos específicos del TFM; también, se explicará a groso modo la metodología del laboratorio en el cual se va desarrollar el piloto.

- **4. Desarrollo especifico de la contribución:** Se describirá de forma detallada la metodología que se seguirá en la implementación del control, especificando los procedimientos de configuración, análisis de parámetros de seguridad y explicando la interacción del dispositivo con la aplicación por medio de su tecnología (NFC) fortaleciendo así la autenticación; luego de lo anterior presentar los resultados de forma objetiva para mostrar las razones por las cuales obtuvimos los resultados.

  Teniendo en cuenta lo anterior este capítulo se dividirá en 3 subcapítulos:
  4.1 Descripción detallada del experimento
  4.2 Descripción de los resultados
  4.3 Discusión de los resultados

- **5. Conclusiones:** Estarán las conclusiones obtenidas luego del desarrollo del piloto sobre la aplicación y se exponen algunas recomendaciones y métricas para futuros trabajos.

- **Bibliografía**
- **Anexos**

2. Contexto y estado del arte

Actualmente vivimos en un mundo tecnológico en donde uno de los temas que cobra mayor importancia es la identidad digital, poder estar completamente seguros de quien entra a un sistema, servidor o incluso quien hace una transacción de tarjeta sea realmente la persona autorizada; es por eso que se han desarrollado a lo largo de la historia diferentes mecanismos que permiten comprobar la identificación de una persona de forma digital o electrónica e ir evolucionando a la combinación de los mismos, hablamos de que se inició con la contraseña desde la época de los romanos en donde usaban una combinación de caracteres para poder enviar mensajes importantes a sus tropas y luego de varios siglos llegar a Fernando Corbató, quien en 1960 frente a una problemática de que varios investigadores del instituto de tecnología de Massachusetts (MIT) podían entrar a un computador y a sus archivos, implemento un método basado en contraseña para que cada investigador al acceder al computador solo entrara a sus archivos durante la conexión, para luego adaptar criptografía y evitar de que esta sea leída tan fácilmente y seguido de agregar datos aletearos dificultando su captura (eset, 2017).

Luego de ver que lo anterior no era suficiente para que un atacante obtuviera la contraseña se fueron agregando más alternativas como el On-time Password (OTP) que son contraseñas de un solo uso evitando que se pueda usar una segunda vez, la clave pública que usa criptografía asimétrica para establecer medio de comunicación seguros y autenticarse con un certificado digital, la huella digital valiéndose de la criptografía usa un hash para identificar al usuario, con tarjetas o elementos físicos que dan la identidad al portador y ya un poco más moderno la autenticación biométrica que usando características únicas del usuario (huella dactilar, iris, o facial) aseguren que es la persona autorizada (AndalucíaCERT, 2016, pág. 5).

Podemos decir que en tiempos más modernos se fueron integrando gestión de identidades con servicios AAA, que son un acrónimo en inglés que al traducir al español es Autenticación, Autorización y contabilidad, con lo que ya podemos dar a un usuario una identidad dar privilegios y seguimiento a los recursos, con lo que llevamos a un siguiente nivel la gestión de identidades acercándonos a reconocer la persona y asignar lo que el necesita según su función o rol (AndalucíaCERT, 2016, págs. 3, 4).

En la actualidad todos tenemos un número importante de credenciales o cuentas a recordar por lo que se comenzaron a implementar single sign on (SSO) que permite a los usuarios autenticarse en diferentes aplicaciones con una sola cuenta, mejorando la seguridad ya que la información que viaja por SSO es cifrada y las diferentes aplicaciones en la red pueden usar este método para cumplir normas de seguridad e identificar el usuario "inequívocamente" al usuario (Chakray Consulting S.L., 2017).

De los anteriores mecanismos de autenticación, que se presentaron de forma muy general, algunas personas podrían llegar a pensar que son robustos y no tendrán problemas de suplantación o spoofing, es más, muchos ignoran esto y acceden a los sistemas y/o aplicaciones sin tener en cuenta las recomendaciones mínimas de seguridad y un atacante podría usar uno de los siguientes métodos, más comunes, para lograr suplantarlos (Antrax, 2010):

- Fuerza bruta: consiste en obtener la contraseña intentando ingresar la correcta, esta cantidad de intentos son elevado ya que se basan en diccionarios para probar una y otra vez hasta que se logre entrar al sistema.
- Spoofing: engañando al usuario haciéndolo pensar que está ingresando sus credenciales a la página de la entidad, cuando son guiados a enlaces enviados por correo que son similares para hacer la captura de las credenciales al intentar ingresar, al suplantar su IP o MAC y consiguiendo las respuestas para tener el acceso a la aplicación.
- Malware: Con software malicioso que permite hacer captura del teclado, como keyloggers, y enviarlo al atacante.
- Hombre en el medio: por medio de la captura de información entre el cliente y el servidor pueden capturar las credenciales, haciendo un análisis de los paquetes y ver si pasan planas y no cifradas.
- IP Splicing-Hijacking: el atacante logra robar la sesión ya establecida entre el usuario y el servidor, solo espera que se efectué la autenticación para engañar al servidor de que continua con el usuario autorizado.
- Ingeniería Social: esta obtiene información del usuario quien por medio de respuestas a preguntas guiadas va dando pistas de como guiar una fuerza bruta o simplemente revela las respuestas de las preguntas de seguridad de una aplicación, como las del correo, por ejemplo.

Los anteriores son los más usados y las técnicas pueden ser tan sofisticadas como simples, es por esto que en vista de que la autenticación de factor único es altamente vulnerable nace la autenticación de doble y múltiple factor, su diferencia es básicamente en la cantidad

de factores empleados, y los factores se definen en las formas por medio de las cuales la aplicación o sistema puede comprobar que si es la persona autorizada (Tatam, 2017):

- Algo que sabe: es la autenticación más simple ya que es información que sabe el usuario como la contraseña y/o respuestas a preguntas que previamente selecciono, por lo mismo es uno de los más vulnerables ya que esta información puede ser obtenida por otro usuario malicioso, por medio de phising o ingeniería social, y hacer uso indebido de su acceso.

- Algo que tiene: es un objetó que solo el usuario debería poseer como los token, tarjetas magnéticas que permitían identificar al usuario, esta autenticación es usada como segundo factor comúnmente en el sector bancario para ingresar al portal o en las transacciones, al igual que el anterior por medio de phising con links malicioso que dirigen usuarios a el portal del atacante puede obtener el código del token en tiempo real o incluso el extraer las claves criptográficas almacenadas en la memoria de los tokens (Goodin, 2012), para el caso de la verificación en dos pasos el atacante necesita conocer el nombre, apellido y teléfono celular para explotar una de las vulnerabilidades de SS7 e interceptar el SMS enviado al usuario, este ataque se puede ver en mayor detalle en el video “Bitcoin wallet hacked via SMS interception” de Positive Technologies (Luis, 2017); también hay que considerar que en caso de que el usuario olvide su objeto no podrá acceder a la aplicación o al sistema.

- Algo que es: esta autenticación busca comprobar la identidad del usuario por medio de una característica física que sea única de cada usuario como la huella dactilar, que generalmente es usada para acceso a edificios u oficinas y en los smartphones, también está el escaneo facial (usualmente usado en smartphones), y la comparación de iris de los ojos un poco menos común que las anteriores. este tipo de autenticación requiere la recopilación de datos previa directamente de los usuarios y son comúnmente usadas para los accesos físicos en ámbitos empresariales e incrementa la dificultad de implementación en controles de accesos no físicos.

La combinación de los anteriores nos permitirá aumentar la seguridad de la autenticación ya que un atacante tendría que explotar varias vulnerabilidades, aunque sigue siendo posible, se podrá dificultar y aumentar el costo del ataque haciéndolo menos atractivo para ellos; la autenticación de dos factores nos permite combinar dos de los tipos de autenticación anteriores donde con uno de ellos complementamos el otro.

Autenticación doble factor, *Ilustración 1*. Clik2Bank. *Recuperado de https:// www.securizando.com*

Ahora si usamos autenticación de factor múltiple, es decir, combinamos más de 2 tipos de autenticación podemos tener, valga la redundancia, un método de autenticación robusto y una alta probabilidad de identificación del usuario (Securizando, 2016).

¿Y si para lo anterior usamos el cuerpo humano para implantar un dispositivo para confirmar que si es la persona que está autorizada? - podríamos pensar que estamos rayando en la ciencia ficción, pero no, si usamos biohacking tenemos un método identificación dentro de nosotros literalmente.

Biohacking es la combinación de biológico y hacking, aunque la segunda es comúnmente asociada con algo ilegal esta vez estaríamos hablando de hacking ético, ya que podríamos decir que es el uso de dispositivos electrónicos corporalmente (Frutos, 2016), y nace de un movimiento llamado Do It Yourselft (DIY) que busca modificar el sistema humano con el objetivo de mejorar las capacidades o crear nuevas funciones por medio de tecnología al alcance de todos, esto ha llevado a personas a implantarse dispositivos electrónicos y en algunos casos por medio de cirugías (Wexler, 2010), a las personas que usan el biohaking para mejorar sus cuerpos o capacidades de forma extrema se les denomina grinders, a la fecha esta tendencia no cuenta con un aval médico y sus procedimientos son bajo el riesgo de la persona por lo que la mayoría de usos son experimentales y van en aumento, a continuación se presenta los casos más conocidos y los dispositivos empleados:

- Neil Harbisson es un artista que sufre de acromatopsia por lo que solo puede distinguir el negro, blanco, gris y sus tonalidades por lo que él y su equipo desarrollaron un software que permite identificar los colores con vibraciones usando una antena insertada quirúrgicamente en su hueso occipital que hace que al identificar un color transmite una vibración a través del hueso (Portero & Linares Pedrero, 2013).

Neil Harbisson, Ilustración 2. Brain F. (2017). ¿Quién es Neil Harbisson? Recuperado de https://feelthebrain.me

- Liviu Babitz diseño un dispositivo que su implante es como un piercing en el pecho, el cual genera una pequeña vibración cada vez que mira hacia el norte e incluye una brújula, esto con el fin de saber hacia dónde vas en cualquier parte del mundo.

Liviu Babitz, Ilustración 3. Lloyd P. (2018). 'North Sense' implant. Recuperado de https://www.dailymail.co.uk

- Rich Lee promueve el biohacking y uno los grinders más polémicos debido a que cuenta con varios implantes, en sus dedos tiene imanes para identificar campos magnéticos y recoger objetos metálicos (experiencia sensorial), dos chips NFC para abrir páginas web y puertas, en el brazo tiene un chip que monitoreo su temperatura

corporal y un implante en sus oídos los cuales son unos imanes que funcionan como auriculares para procesar mejor el sonido de forma que conecta los auriculares con una bobina de cobre la cual por medio de un campo magnético hace vibrar el implante y este produce el sonido (Nye, 2018).

Rich Lee, Ilustración 4. BBC. (2018). Los "biohackers". Recuperado de https://laopinion.co

- Gabriel Licina es otro de los grinders polémicos ya que uso cloro e6 y lentes de contactos negros para ver en la oscuridad, el indico que el experimento funciono durante un tiempo.

Gabriel Licina, Ilustración 5. Abc.es. (2015). Una inyección para tener visión nocturna. Recuperado de https://www.abc.es

- Aunque las aplicaciones anteriores pueden llegar a ser extremas los implantes más populares son los chips RFID y NFC que son comúnmente usados para la apertura de puertas con chapas digitales, homologación de pagos con tarjetas de crédito, el

desbloqueo de móviles, entre otros usos; para este tipo dispositivos encontramos 2 tipos de implantes:

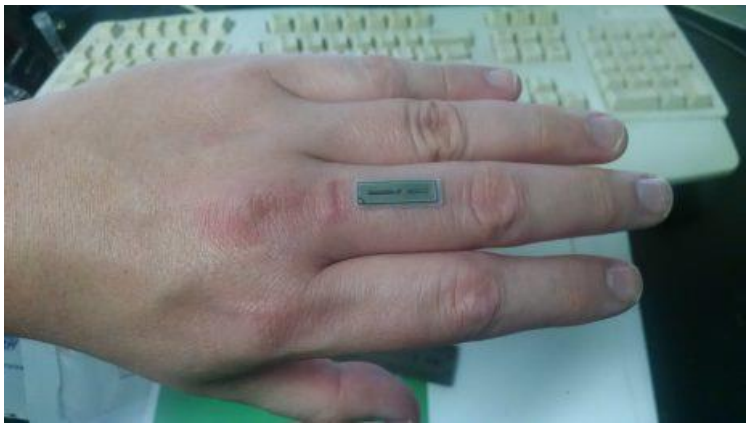
- Transpondedores tipo bioglass en los cuales el chip esta encapsulado en un vidrio y su implante es subcutáneo en la mano entre el pulgar y el índice:

Bioglass, Ilustración 6. Dangerous T. Transponders X-serie. Recuperado de https://dangerousthings.com

Meow Leudo se implanto un transpondedor bioglass para reemplazar la tarjeta que usa en las estaciones de transporte, de forma que al llegar al torniquete pone la mano en la registradora, actualmente tiene una demanda contra el gobierno australiano por imponerle un comparando por no tener un tiquete valido

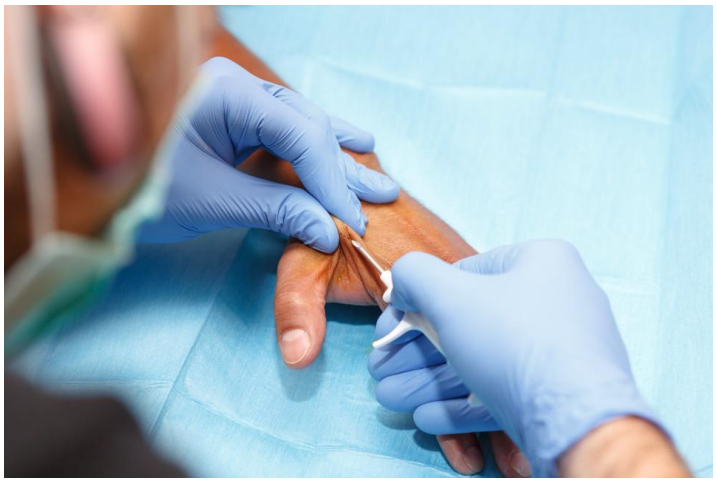
Caso de uso de transpondedor bioglass, Ilustración 7. Mackinnon O. (2017). Opal card implanted in skin. Recuperado de https://abc.net

- Transpondedores tipo flex son fabricados en un biopolímero flexible que también son de implante subcutáneo en la falange del dedo:



Chip Flex, Ilustración 8. Dangerous T. Transponders Flex-serie. Recuperado de https://dangerousthings.com

Como se mencionó anteriormente los transpondedores son los dispositivos usados para la identificación ya que tienen la capacidad de guardar información y ser usado en los sistemas para dar acceso, no solo en términos de acceso de físico o entornos digitales de uso personal; el intercambio de información se da por que los chips cuentan con tecnóloga RFID y NFC que se activan por medio de un campo magnético que excita la bobina lo que hace que los transpondedores no necesiten batería, en cuanto a los transpondedores bioglass tienen modelos con RFID y modelos con NFC y los transpondedores flex solo cuenta con tecnología NFC, otra gran diferencia es el proceso de implante aunque ambos son subcutáneos el bioglass es menos complejo ya que pueden ser insertados por medio de una aguja por un perforador profesional:



Implante bioglass, Ilustración 9. Dangerous T. Transponders x-serie. Recuperado de https://dangerousthings.com

Los transpondedores flex tienen un procedimiento más avanzado y por recomendación del fabricante *Dangerous Things* se requiere que la incrustación sea realizada por un medio o

un artista de modificación de cuerpo profesional con licencia ya que se debe hacer una incisión de 9mm y luego separarla capa dérmica para deslizar el dispositivo y una vez en posición saturar la herida sin perforar el dispositivo:

Herida de implante transpondedor flex, Ilustración 10. Dangerous T. Transponders Flex-serie. Recuperado de https://dangerousthings.com

De esta forma es posible hacer una combinación de un factor de autenticación “tradicional”, por decir los ya conocidos, y un transpondedor con información almacenado en el para un segundo factor por lo que la persona lo va llevar consigo todo el tiempo sin problemas de baterías.

3. Objetivos concretos y metodología de trabajo

3.1 Objetivo General

Incrementar la seguridad en la identificación digital de los usuarios al momento de autenticarse por medio de ssh en sistema Unix mediante la combinación tokens de basados en tiempo y un transpondedor biohacking, usando sus características de cifrado, protección y control de acceso a la memoria del mismo.

Implementar un método de autenticación con un transpondedor biohacking en un servicio de control remoto de administración de un ambiente simulado que puede ser llevado a entornos reales y/o de producción.

3.2 Objetivos específicos

- Analizar los diferentes tipos de dispositivos biohacking en la actualidad.
- Identificar el dispositivo más apropiado para implementarlo en un método de autenticación.
- Implementar autenticación de doble factor para el acceso a un sistema operativo.

3.3 Metodología de trabajo

En un ámbito empresarial el acceso a los sistemas operativos ha sido una de las grandes preocupaciones ya que es necesario saber que el usuario que accede a sus servidores es ser realmente quien está autorizado, debido a que sobre ellos corren bases datos, servicios web, aplicaciones financieras, etc., que al final son el core de negocio y no se puede permitir que alguien no autorizado afecte la operación causando pérdidas no solo económicas si no de imagen.

Se establecerá un sistema de autenticación de dos factores sobre un servidor Linux, en el cual se implementarán dos mecanismos de autenticación para acceder vía ssh emulando así un entorno real, basado en los siguientes pasos:

item	Descripción	Semana 1		Semana 2		Semana 3		Semana 4	
1	Descarga de medios								
2	Adquisición dispositivo BioHacking								
2	Instalación de servidor ubuntu								
3	Configuración básica de servidor								
4	Instalación de google authenticator								
5	Pruebas de google authenticator								
6	Configuración de Biohacking								
7	Integración de lector NFC en estación de trabajo								
8	Integración de Biohacking con acceso ssh								
9	Pruebas de autenticación con los 2 factores de autenticacion								

Cronograma de actividades, Tabla 1.

Los mecanismos se implementaran sobre un servidor Ubuntu, ya que en las empresas un porcentaje considerable de sus servidores en datacenter tiene sistemas operativos basados en Linux, al tener una licencia publica general (GPL) se puede tener acceso y crear este piloto más fácilmente que otros sistemas que requieren de algún licenciamiento y puede ser replicado en ambientes empresariales como por ejemplo RedHat, este servidor se instalara en un ambiente virtual y se dejara preparado para acceder desde una estación de trabajo.

Como primer método de autenticación se empleara *google authenticator* el cual genera códigos pseudoaletoreos por medio de un código secreto en función del tiempo, dado a que es una herramienta libre se podrá implementar sin necesidad de una adquirir una licencia y podremos acceder a ella desde internet por medio de una extensión de *google Chrome* o desde una aplicación en el Smartphone y se obtendrán las ventajas criptográficas de una autenticación por token; dado que es una herramienta que ya existe se implementara de primera en el servidor y se dejara funcionado al ser un caso de uso ya existente y probado.

Teniendo en cuenta las ventajas presentadas anteriormente de los dispositivos biohacking con NFC tipo 2, se implementará el segundo mecanismo de autenticación con un chip xNT y un lector NFC desde las estaciones de trabajo, de forma que podamos reemplazar la contraseña introducida manualmente por la interfaz NFC que permitirá junto con el token acceder al perfil al que se está autorizado, estableciendo así un método de autenticación que nos permita aumentar la capacidad de confirmar de que el usuario es en verdad quien dice ser ya que este dispositivo solo lo va a tener implantado el chip.

El Lector NFC está basado en un módulo PN532 programado para lograr hacer la lectura del biohacking de forma que sea compatible con la estación de trabajo, de esta forma la implementación en cuanto a hardware es económicamente viable.

4. Desarrollo especifico de la contribución

En el desarrollo del capítulo se presentará la implementación del chip xNT como mecanismo de doble factor de autenticación en un ambiente de laboratorio que puede ser replicado en ambientes reales con las mismas herramientas u homologas.

4.1 Descripción detallada del experimento

Para la implementación de este piloto se usaron herramientas de uso libre pero que son tecnologías usadas en entornos reales, a continuación, se describirá su elección y uso de cada una de ellas:

- VMWare Workstation: se usa el software para virtualizar el host basado en Unix sobre una estación de trabajo, este nos permite crear máquinas y asignar recursos según la capacidad del hardware físico y configurar una red virtual para lograr acceder al servidor vía ssh.

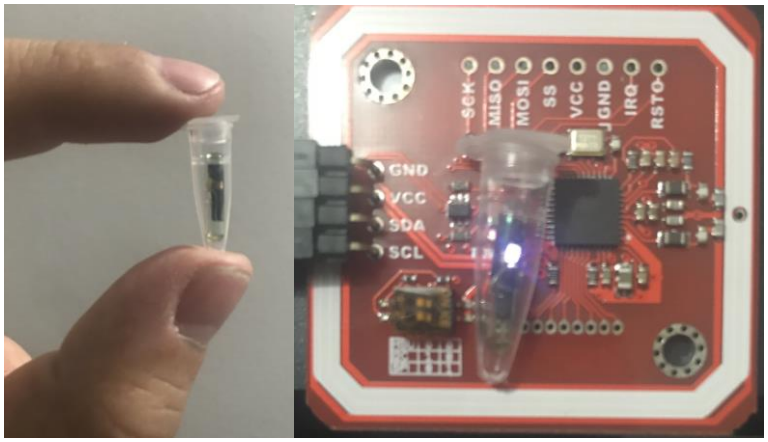
- Host Ubuntu: debido a que el piloto está enfocado a emular un entorno real se eligió Ubuntu como sistema operativo Linux debido a que tiene instalación guiada y es de los Linux con menor requerimiento de disco, memoria y CPU.

- Google authenticator: como factor de autenticación por token complementa el inicio de sesión con el chip, también permite habilitar verificación en 2 pasos, logrando tener doble factor de autenticación con verificación en 2 pasos; esta tecnología es de uso libre y es una de las que se puede homologar por soluciones empresariales, como por ejemplo SecureID de RSA, ya que tiene limitantes como que no es soportada para RedHat.

- Tarjeta de diagnóstico RFID: esta tarjeta será la que nos permite comprobar que el lector NFC está leyendo el dispositivo al encender un led, para este piloto el HF, con esto se puede descartar daños del chip o la configuración del módulo PN532.

Tarjeta de diagnóstico NFC, Ilustración 11.

- Capsula xLED-HF: dado que el chip xNT es para ser usado subcutáneamente esta cápsula contiene un led que incrementa su luminosidad para poder ubicar el lugar idóneo de lectura del implante.

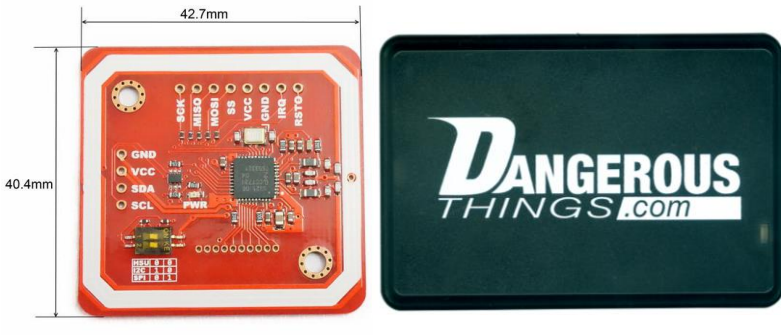


Chip xLED NFC, Led encendido en lector, Ilustración 12.

- Chip NTAG216: este es el transpondedor bioglass el cual es un cilindro de 2 mm por 12 mm y en su interior se encuntra el chip NFC.

Transpondedor NFC NTAG, Ilustración 13.

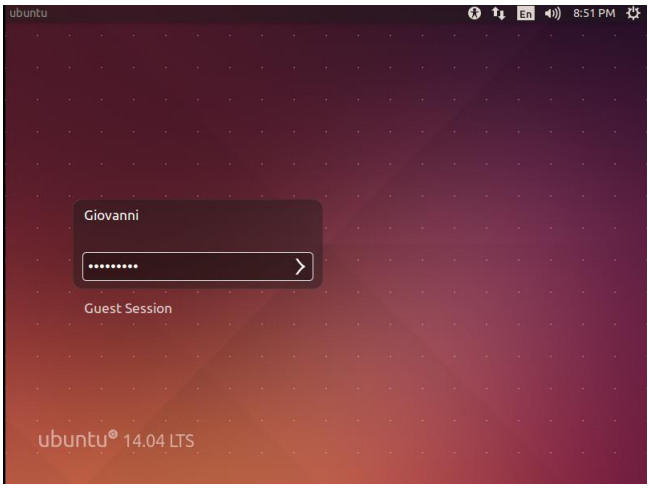
- Lector NFC: En este caso se basa en un módulo PN532 que va a un arduino UNO para hacer la lectura y escritura, adicional un lector KBR1 el cual solo es lector basado en el módulo anterior y se comporta en un sistema operativo como un lector bluetooth.



Izquierda módulo PN532 y derecha KBR1, Ilustración 14.

4.1.1 Preparación de ambiente base

El piloto se desarrollará desde la instalación del host Ubuntu, el cual va a simular el servidor Unix, dicha instalación está a detalle en el anexo A:



Host operativo, Ilustración 15.

Una vez el servidor está instalado se procede a instalar el ssh server para comenzar con la ambientación de un sistema real:

```
giovanni0@ubuntu:~$ sudo apt-get -y update && sudo apt-get -y install openssh-se
rver
[sudo] password for giovanni0:
Sorry, try again.
[sudo] password for giovanni0:
Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.2p2-4ubuntu2.6).
```

Instalación servicio ssh, Unix, Ilustración 16.

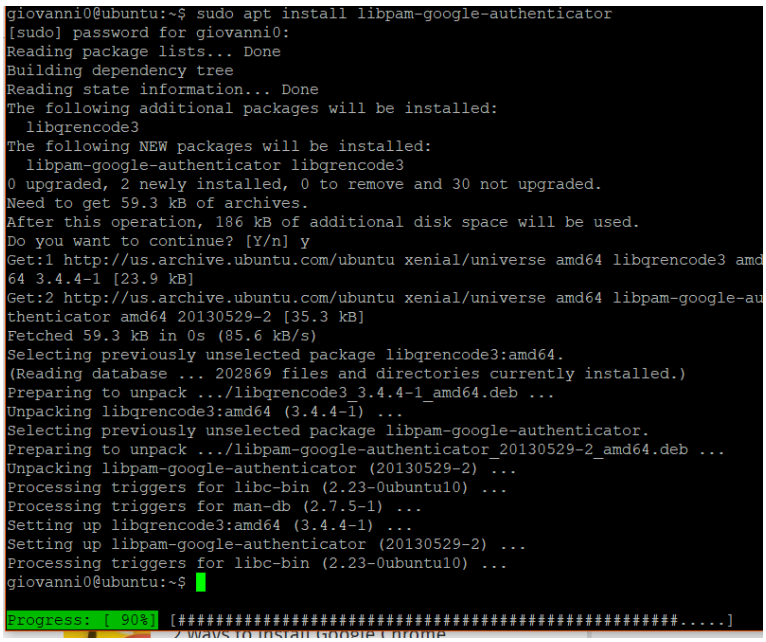
Luego de tener el host configurado y en red se procede a instalar google authenticator, el cual es un factor de autenticación basado en token ya que genera un numero de seis dígitos pseudoaleatoreamente cada 30 segundos, este desarrollo de Google usa estándares abiertos por la iniciativa para la autenticación abierta (OATH, siglas en ingles) y para el piloto se implementó la generación de códigos basado en tiempo (TOTP) relacionado al estándar RFC 6238, el cual se basa en el algoritmo HMAC –SHA1 y se agrega el valor de un contador en aumento, es decir el tiempo (hora, fecha), para

representar el mensaje HMAC en donde el valor del contador es reemplazado por el tiempo (T), como se ve en la siguiente ecuación (M'Raihi, Machani, Pei, & Rydell, 2011):

$$HOTP\ (K, T) = Truncate\ (HMAC\text{-}SHA\text{-}1\ (K, T))$$

El estándar puede usar funciones HMAC basadas en sha-256 y sha-512 en lugar de la ecuación anterior para establecer criptografía más robusta, tanto el software como el servidor deben saber el tiempo actual para la generación del número ya que esta sincronización del reloj afecta la generación, el software y el servidor deben compartir la misma clave secreta para generar un secreto compartido (M'Raihi, Machani, Pei, & Rydell, 2011).

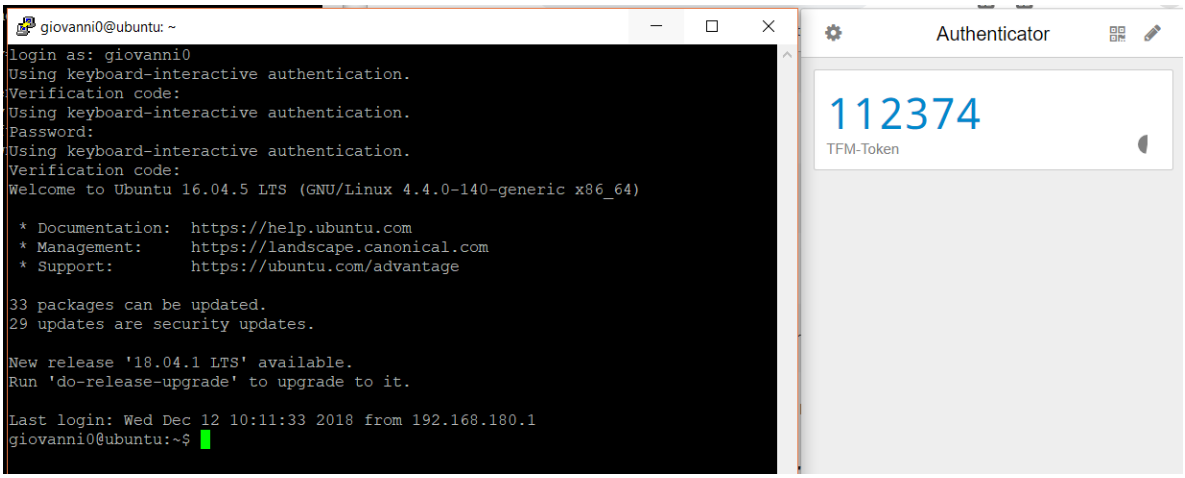
Desde el terminal de Ubuntu se descarga e instalan las librerías de google authenticator de internet y luego se ejecuta el comando google-authenticator para generar la clave secreta que puede ser un QR o un código:



Instalación google authenticator, Ilustración 17.

Generación de secreto, Ilustración 18.

En caso de no tener acceso al software en el móvil o internet, la aplicación permite tener códigos de emergencia para acceder al sistema y poder enrolar otro dispositivo, también se habilita la solicitud de un segundo código para obtener una segunda verificación como control adicional como se pude ver a continuación:

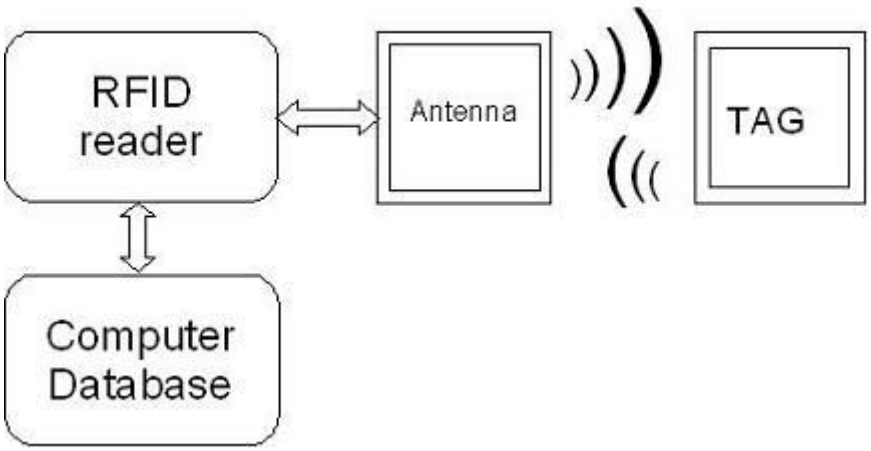


Ingreso por ssh con token, Ilustración 19.

4.1.2 Transpondedor Biohacking

Tal como se expuso en la última parte del capítulo 2 los transpondedores a implantar con fines de identificación son RFID y NFC de forma que el primer tópico a validar es las ventajas de cada una de ellas, su diferencia más grande está en las frecuencias ya que las de RFID están entre 124 KHz y 135 KHz y NFC es de 13,56 MHz, cabe aclarar que NFC es una rama de alta frecuencia de la tecnología RFID (por tanto lo podemos tomar como la mejora del mismo), tiene una comunicación más rápida con los dispositivos pero requiere un hardware especifico que puede tener una diferencia significativa en costos (Mejillon Yagual & Villamarin Zambrano, 2017).

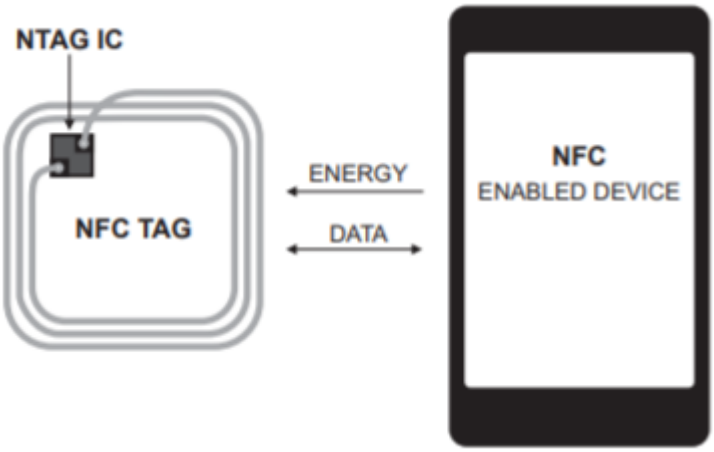
NFC fue diseñado pensando en intercambiar datos de forma segura, ya que la distancia fue reducida en un entorno de proximidad de no mayor a 10 cm, para este caso en específico el NFC es pasivo ya que es un chip que, al ser alimentado por inducción de un campo magnético como se pude ver en la figura de abajo, en este caso del lector, podrá intercambiar información dependiendo la configuración será cifrado como por ejemplo por SSL, es por esto que una de sus aplicaciones en la actualidad es el pagos bancarios en establecimientos comerciales.



Comunicación NFC, Ilustración 20.

Dado a que el chip contiene información y se va a modificar la comunicación será en modo lectura / escritura este soporta el estándar ISO/IEC 14443, la cual permite leer y escribir datos a partir de las etiquetas (Estados Unidos Patente nº US20140256251A1, 2013).

Teniendo en cuenta lo anterior se eligió un chip que cumpliera con esto, de forma que los transpondedores tipo xNT son los indicados para este piloto, dentro de la serie x se eligió el chip NTAG216 el cual tiene el mayor espacio en memoria de usuario de los NTAG21x, estas referencias están diseñadas para comunicar datos a una velocidad de 106 Kbit/s, en modo pasivo (NXP Semiconductors, 2013):



Comunicación sin contacto de chip NTAG21x, Ilustración 21. (NXP Semiconductors, 2013)

A nivel de seguridad cuenta con un UID programado de 7 bytes y 888 bytes de memoria de usuario para leer y escribir, una contraseña de 32 bits para protección de las funciones, es decir, prevenir operaciones en la memoria sin autorización, esta contraseña puede ser cambiada una vez ya que por defecto viene establecida con FFFFFFFF, también cuenta con una firma de originalidad basada en criptografía de curva elíptica y una un campo para programar el bloqueo del modo lectura (Dangerous Things).

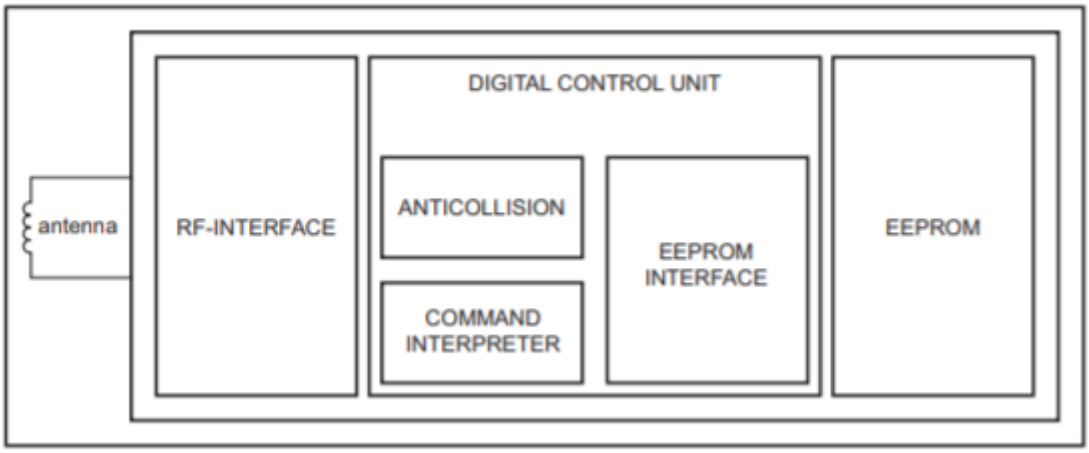


Diagrama de bloque, datasheet, Ilustración 22.

Como mejora de la tecnología RFID este chip cuenta con un algoritmo anticolisiones para operar más de una etiqueta simultáneamente, su interfaz RF ejecuta la funciones modulación/demodulación, regenerador de reloj, regulador de voltaje y rectificador, cuenta con un módulo interpretador de comandos para poder acceder a la memoria del chip; la EEPROM del NTAG216 tiene en total 924 bytes que están organizados en 26 bytes reservados para la configuración de datos y manufactura otros 34 bits que son usados para el bloqueo de solo lectura, 4 byte disponibles como capacidad del contenedor y los 888 bytes para programar y leer la memoria (NXP Semiconductors, 2013, págs. 6-7).

Para la integridad de los datos el chip cuenta con los siguientes mecanismos para la comunicación con el lector NFC:

- CRC de 16 bits por bloque
- bits de paridad para cada byte
- control de conteo de bits
- codificación de bits para distinguir entre unos, ceros y sin información
- monitoreo de canales (secuencia de protocolo y análisis de flujo de bits)

Como se indicó previamente la comunicación es iniciada por el lector NFC y la unidad de control digital del chip es quien controla la comunicación, a continuación, se describirá los estados del chip:

- Idle: luego de recibir la primera señal del lector/escritor NFC el chip se establece en este estado y saldrá del estado si recibe los comandos

REQA o WUPA, en caso de que no sea ninguno de los anteriores el chip lo tomara como un error y continuara en este estado.

- Ready 1: este estado usa los comandos de ANTICOLLISION y SELECT, resuelve los 3 primeros bytes del UID, al recibir los datos correctos pasa en cascada de nivel 1 al estado Ready 2 donde resuelve la segunda parte del UID o pasa directamente con el comando READ para omitir la anticolisión y pasar inmediatamente a activo si interpreta un dato como error regresa al estado Idle. La anticolisión en este nivel consiste si hay otros NFC y con el comando read desde la dirección 0 selecciona todos los dispositivos se produce una colisión por los diferentes números de serie.

- Ready 2: este estado resuelve la segunda parte del UID, los siguientes 4 bytes, con un nivel 2 del comando de ANTICOLLISION y es un estado que lo actica el Ready 1 pero puede ser omitido por el comando READ del estado anterior. En este nivel la anticolisión verifica que sea el único dispositivo en la comunicación verificando el byte de acknowledge e indica que el proceso de anticolisión termino pasando a Activo o volviendo a Idle.

- Active: en este estado se operan las funciones de lectura y memoria y pasa por medio del comando HLTA pasa al estado Halt si interpreta un error y luego de verificar la contraseña pasa a estado autenticado.

- Autenticado: este estado permite acceso a todas las páginas y operaciones de la memoria ya que el password fue el correcto.

- Halt: es el estado de salida de los anteriores cuando interpretan errores o llega información diferente a la esperada y solo el comando WUPA permite volver a iniciar el flujo de estados y de resto permanece sin cambios en este.

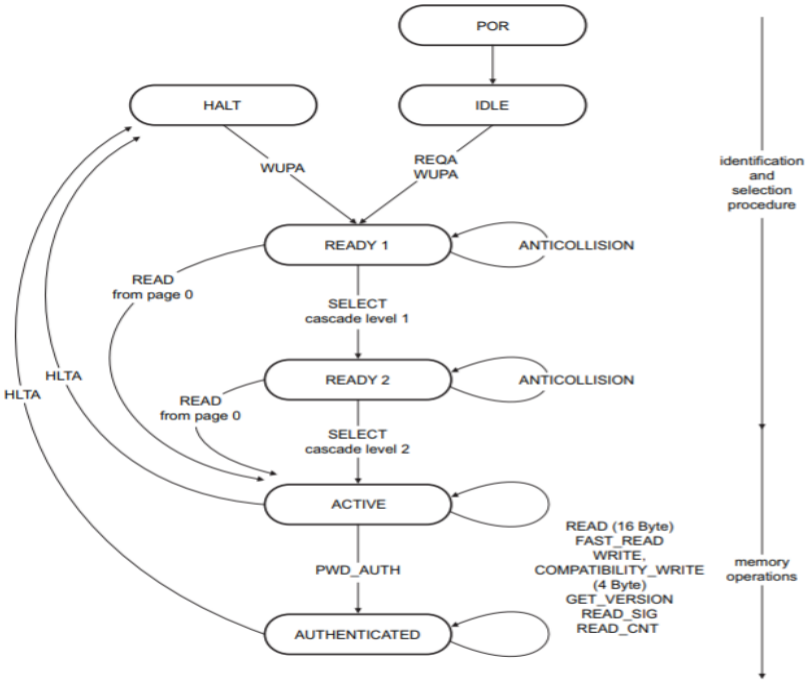


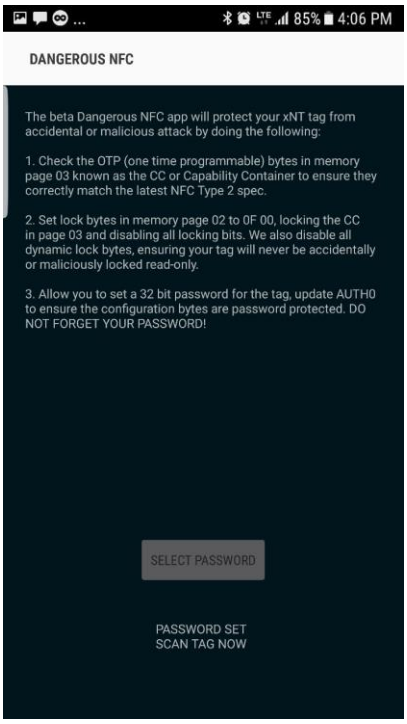
Diagrama de estados, Ilustración 23.

La memoria EEPROM tiene 231 páginas de 4 bytes cada una, en las primeras 2 paginas son el número de serie único (UID, por sus siglas en inglés) más el primer byte de la página 02 y el segundo byte de la misma página está reservado para datos internos, estas páginas están protegidas contra la escritura, los siguientes bytes de la página 02 son el mecanismo de bloqueo estático de escritura de las demás páginas, se puede programar un bloqueo individual hasta la página 09 de modo que el byte 0 de cada página va indicar si está bloqueada; los bytes de bloqueo dinámico se encuentran la página 226 en donde toma los tres primeros bytes de la página para el bloqueo de las páginas 10 en adelante.

Page Adr		Byte number within a page				
Dec	Hex	0	1	2	3	Description
0	0h	serial number				Manufacturer data and static lock bytes
1	1h	serial number				
2	2h	serial number	internal	lock bytes	lock bytes	
3	3h	Capability Container (CC)				Capability Container
4	4h	user memory				User memory pages
5	5h					
...	...					
224	E0h					
225	E1h					
226	E2h	dynamic lock bytes			RFUI	Dynamic lock bytes
227	E3h	CFG 0				Configuration pages
228	E4h	CFG 1				
229	E5h	PWD				
230	E6h	PACK		RFUI		

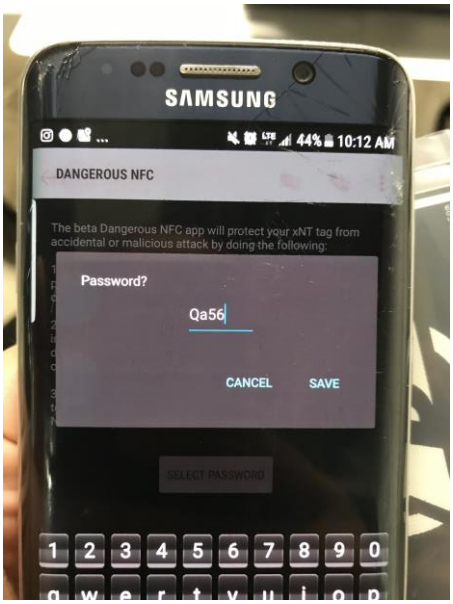
Organización de memoria NTAG216, Ilustración 24.

Una vez visto y analizado lo anterior se procede a definir los bytes de bloqueo por medio de una aplicación del fabricante llamada Dangerous NFC, la cual se puede descargar desde playstore solo para dispositivos Android con NFC:



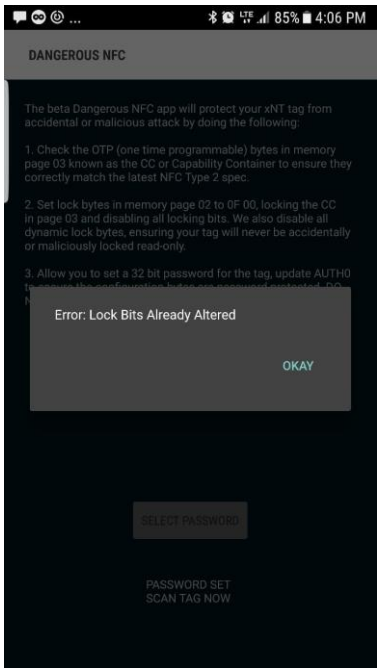
Screenshot de aplicación, Ilustración 25.

Luego de entrar a la aplicación se acerca el transpondedor a la parte de atrás del celular ya que es donde está el chip NFC del celular y se hace el cambio de la contraseña:



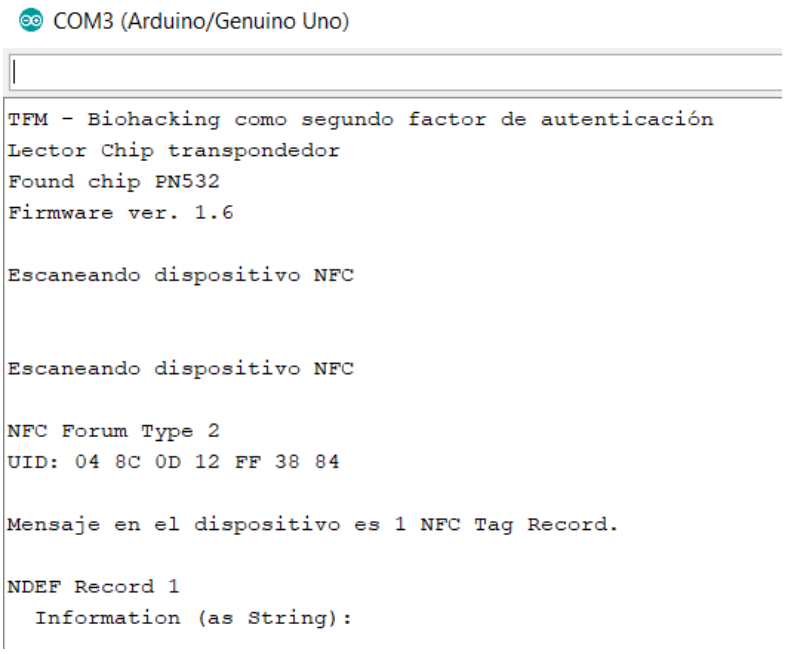
Screenshot modificación contraseña, Ilustración 26.

Luego de esto se valida que los bits ya fueron escritos en la EEPROM:



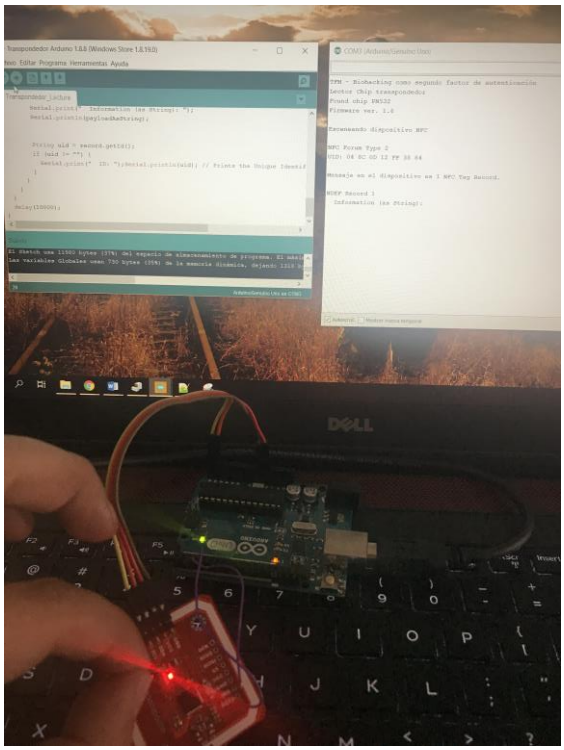
Screenshot verificación de modificación, Ilustración 27.

Configurado lo anterior se procede hacer la lectura del dispositivo por medio de comandos propios del chip, los cuales están insertados en un programa quemado en un arduino UNO al cual está conectado el módulo PN523 y obtenemos el UID en hexadecimal y se puede observar que no tiene información en las páginas:



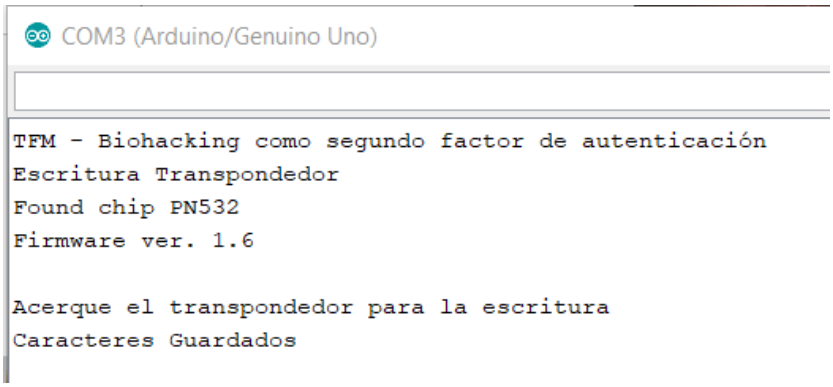
Captura de datos con PN532, Ilustración 28.

El comando con el que se hace la captura de UID es record.getId y es cargado en una variable tipo string, los códigos de los programas se encuentran en el anexo 2, básicamente fueron usadas las librerías del módulo PN532 en donde ya están las funciones creadas con las instrucciones que aparecen en el datasheet del NTAG21X.



captura de la información del transpondedor, Ilustración 29.

A modo de laboratorio se igualan las variables de UID y lo guardado en las páginas, solo se acerca el transpondedor al módulo PN532 y con el programa cargado se ejecuta la operación de escritura como se puede ver en la ilustración 30 y se verifica la información en memoria con el programa de lectura tal como se ve en la ilustración 31 ya está cargado el valor

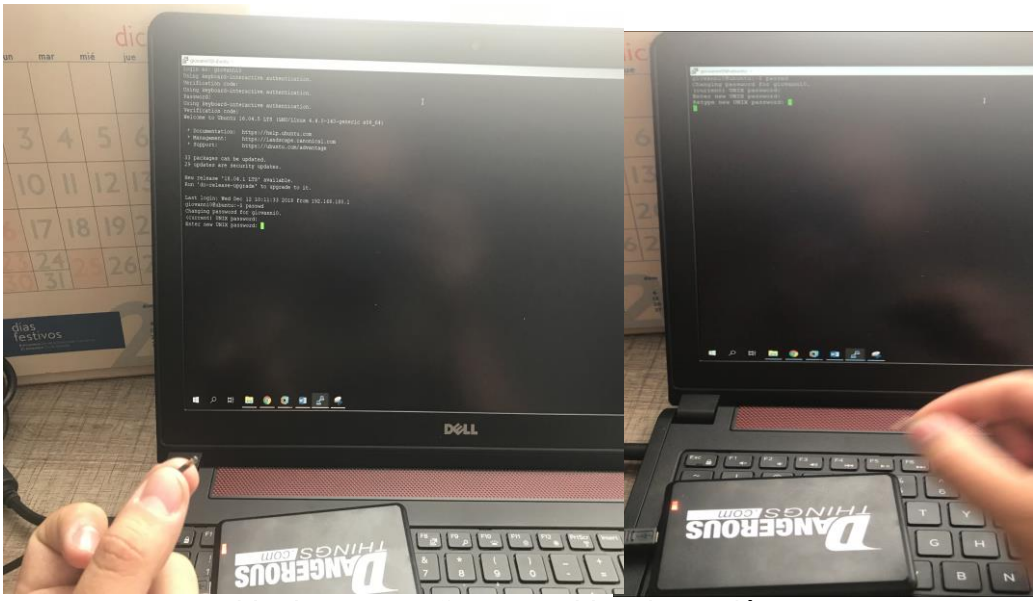


Salida de programa de escritura, Ilustración 30.

```
COM3 (Arduino/Genuino Uno)

TFM - Biohacking como segundo factor de autenticación
Lector Chip transpondedor
Found chip PN532
Firmware ver. 1.6

Escaneando dispositivo NFC

NFC Forum Type 2
UID: 04 8C 0D 12 FF 38 84

Mensaje en el dispositivo es 1 NFC Tag Record.

NDEF Record 1
  Information (as String): 048c0d12ff3884
```

Salida de programa de lectura, Ilustración 31.

Escaneando con el lector se va a cambiar la contraseña del Ubuntu, el cual va a ser el encargado de negociar con el transpondedor y entregar en claro el valor para poder establecerlo como la contraseña del usuario giovanni0:



Cambio de contraseña en servidor, Ilustración 32.

Ya con google authenticator instalado y el transpondedor integrado al servidor se tiene doble factor de autenticación, más una segunda verificación que brinda google, para ingresar por ssh y a continuación se presenta un diagrama lógico del piloto:

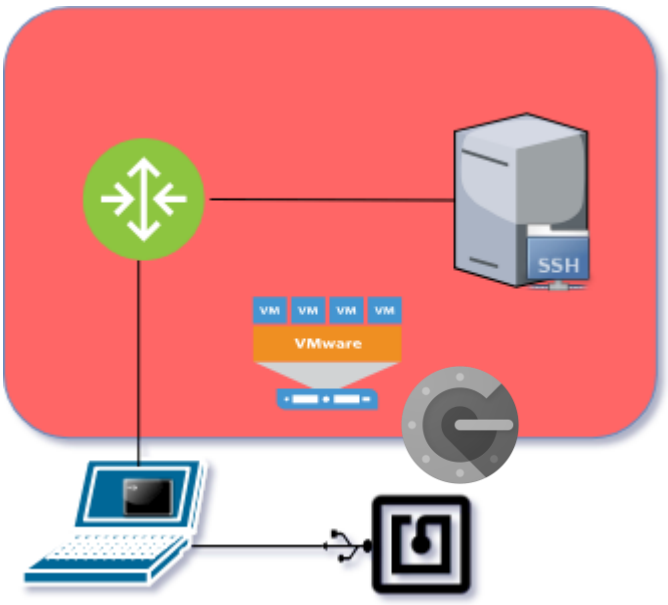
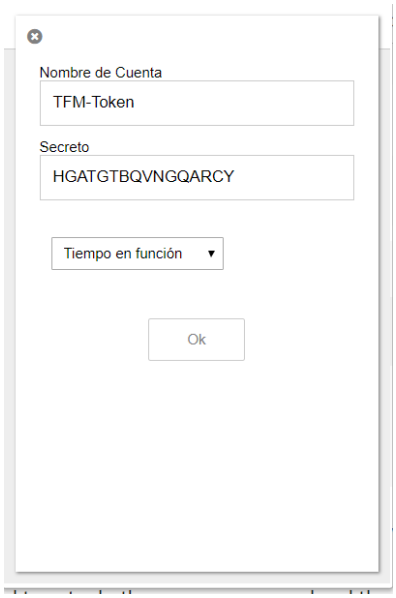


Diagrama lógico de piloto, Ilustración 33.

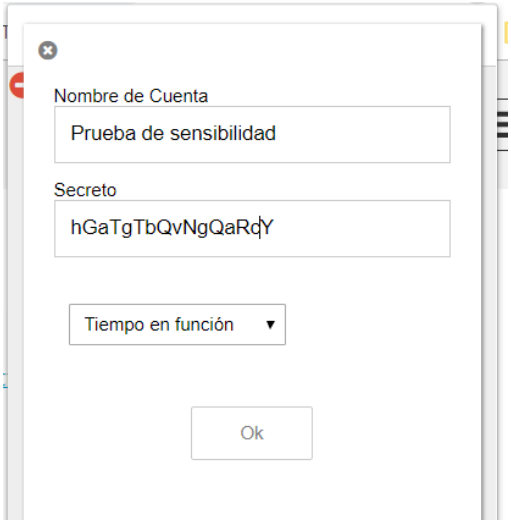
4.2 Descripción de los resultados

El primer factor de autenticación se usa el módulo PAM google authenticator, que usa para el cifrado el estándar RFC 6238 con base en la función HMAC-SHA1 y el reloj del dispositivo en donde se ejecute, este genera una clave secreta de 128 bits, y es ingresada a la extensión de Chrome:



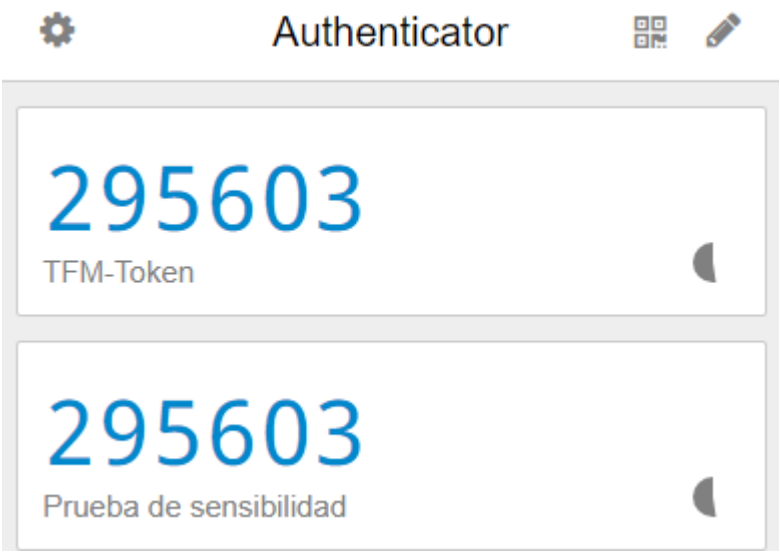
Configuración de token, Ilustración 34.

También se configura otra cuenta, pero con la mitad del código secreto en minúscula con el fin de poder determinar la sensibilidad de las mayúsculas:



Prueba de sensibilidad, Ilustración 35.

Una vez configuradas las dos cuentas se puede ver que genera el mismo código:



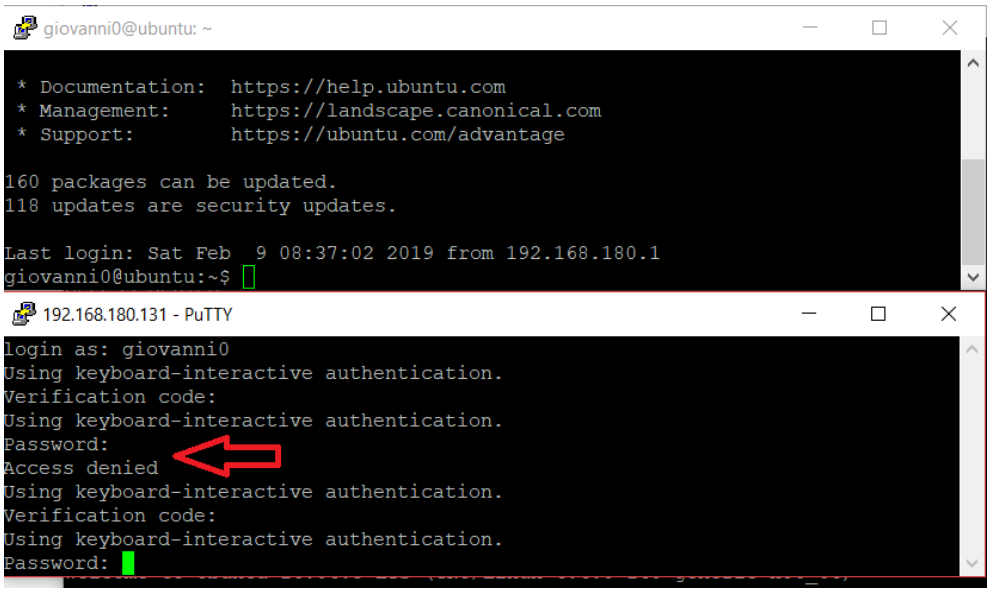
Comparación de códigos, Ilustración 36.

Google authenticator permite habilitar características de seguridad para prevenir fuerza bruta y hombre en el medio:

```
Do you want me to update your "/home/giovanni0/.google_authenticator" file (y/n)
 y

Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n)
y

By default, tokens are good for 30 seconds and in order to compensate for
possible time-skew between the client and the server, we allow an extra
token before and after the current time. If you experience problems with poor
time synchronization, you can increase the window from its default
size of 1:30min to about 4min. Do you want to do so (y/n) y

If the computer that you are logging into isn't hardened against brute-force
login attempts, you can enable rate-limiting for the authentication module.
By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting (y/n) y
```

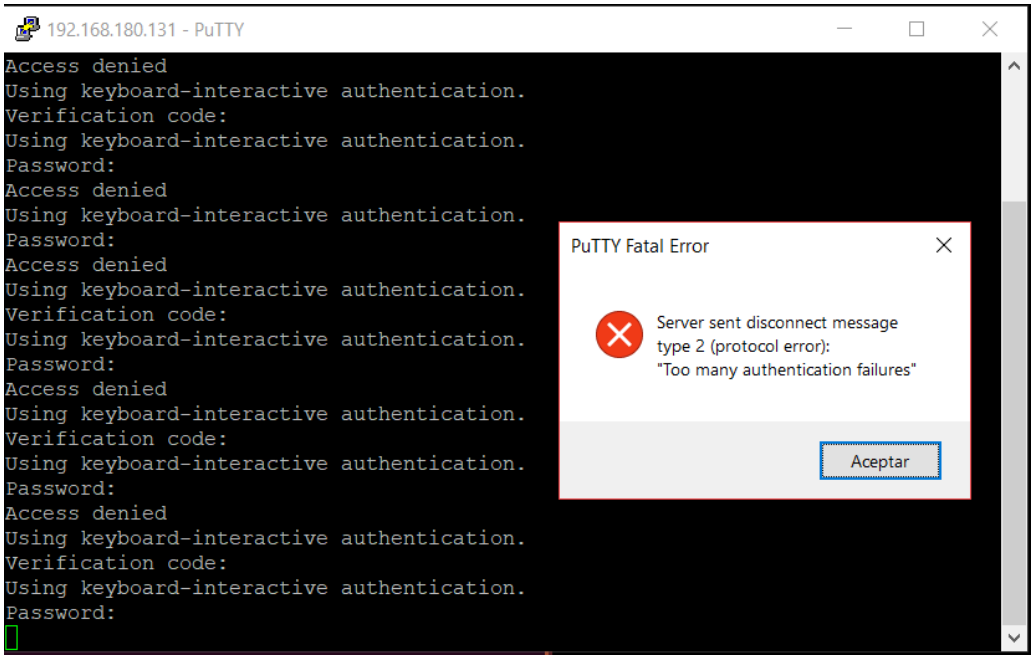
Habilitación de características de seguridad, Ilustración 37.

De esta forma se simula un hombre en el medio abriendo dos sesiones ssh e ingresando los mismos datos para validar que si este funcionado, se puede ver que permite pasar a la instancia de la contraseña y deniega el acceso:



Prueba Man in the Middle, Ilustración 38.

Se continua con una prueba de fuerza bruta y al ingresar primer código erróneo se ve que deniega al ingresar la contraseña así este bien, es decir, pero si se demora más de los 30 segundos permite más intentos de los configurados y se recibe respuesta del sistema operativo:

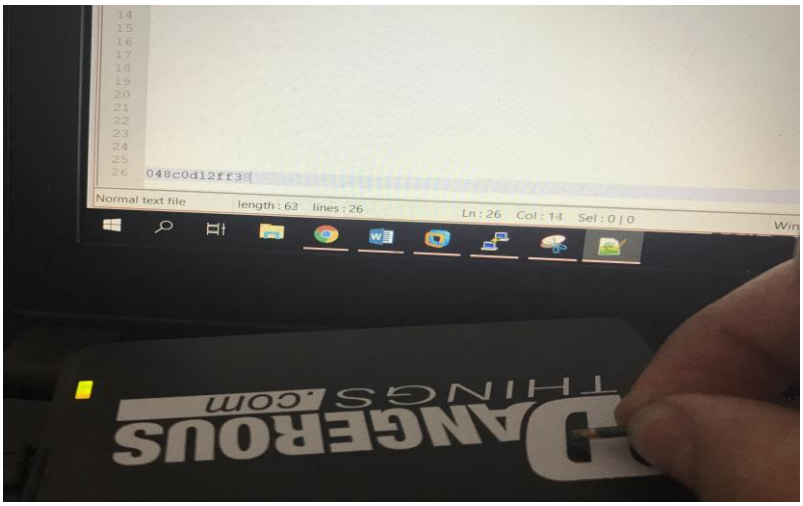


Prueba fuerza bruta, Ilustración 39.

Para que el transpondedor pueda ser leído por el PN532 o el KBR1 tiene que estar muy cerca del lector y se comprueba con la capsula led, en donde tenga su máximo de iluminación, de otra forma no es detectado:

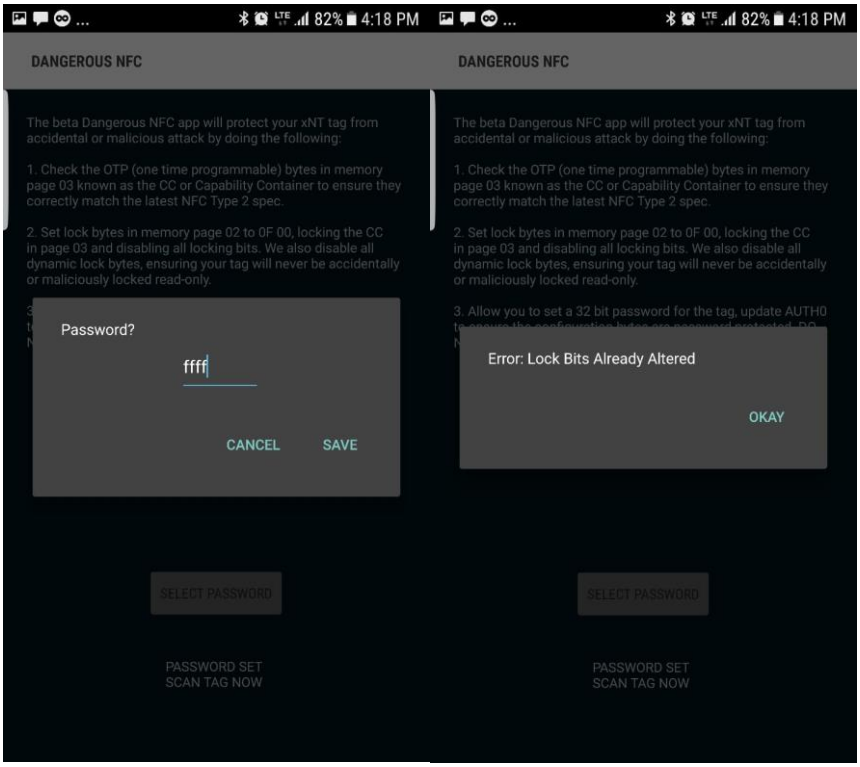
Verificación de punto ideal de lectura (izquierda mínima, derecha máxima),Ilustración 40.

En la configuración más básica de lectura de NFC es posible obtener el UID del chip mas no el contenido en la memoria, por lo que la lectura con el KBR1 nos trae de inmediato el UID del transpondedor, se puede ver que la lectura la hace en el sector de máxima emisión del lector:



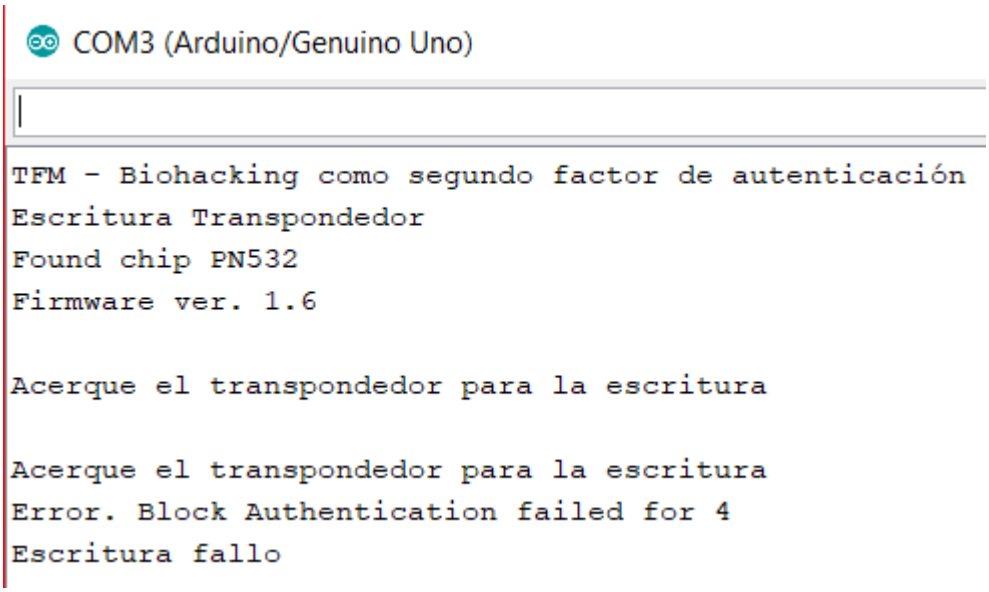
Lectura inicial del transpondedor, Ilustración 41.

Luego de cambiar la contraseña por defecto para la protección de escritura del chip, con la aplicación *Dangerous NFC,* se intenta volver a cambiar y la aplicación indica que estos ya fueron modificados, este intento se realiza luego de deshabilitar/habilitar el NFC y reinstalar la aplicación del móvil para evitar que guardara como referencia el UID del transpondedor:



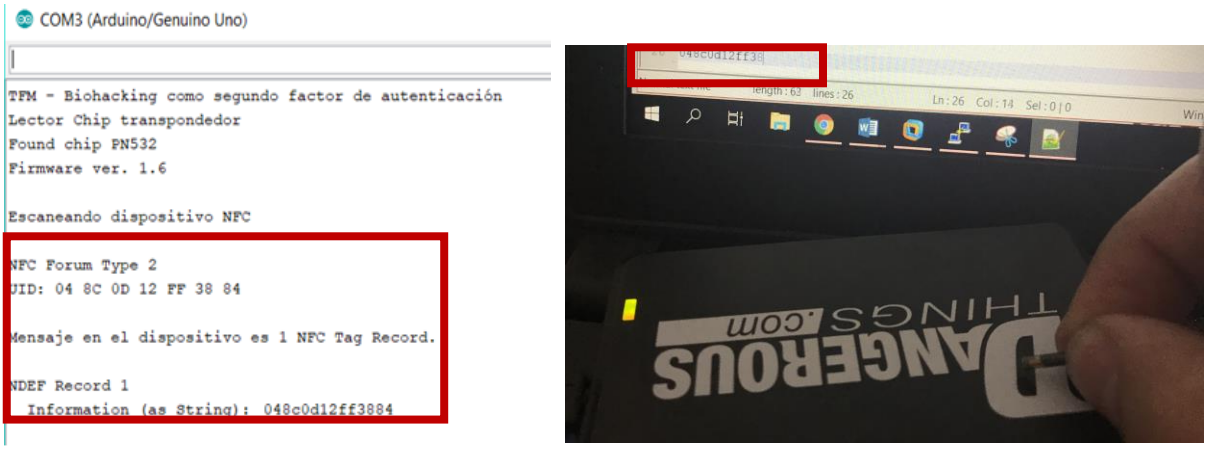
Cambio de contraseña de bloque, Ilustración 42.

Al establecer la contraseña en las librerías del PN532 se puede hacer la escritura en las páginas de las memorias, al establecer la contraseña por la aplicación se limita al código ascii y si se establece por líneas de código se puede establecer en hexadecimal ya que las librerías hacen la escritura directamente en los bits de las paginas, pero si esta se modifica falla el proceso de autenticación y no permite ejecutar operaciones sobre la memoria:



Escritura en memoria con contraseña invalida, Ilustración 43.

Ya con los parámetros de protección establecidos, a excepción de la lectura por autenticación, el transpondedor puede ser leído por cualquier lector, durante las pruebas realizadas depende del código que tenga el lector trae la información, por ejemplo, el KBR1 solo trae el UID del transpondedor o toda la memoria si con el PN532.



Lectura de transpondedor con dos dispositivos, Ilustración 44.

La diferencia anterior se basa en los registros a leer de cada uno, mientras el primero solo trae el registro del UID (ilustración 45) un código un poco más elaborado traer en una función *for* todas las páginas de la memoria (ilustración 46):

```
boolean success;
uint8_t uid[] = { 15, 15, 15, 15, 15, 15, 15 };
uint8_t uidLength;

success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, &uid[0], &uidLength);

if (success) {
  Serial.println("Tarjeta encontrada");
  Serial.print("UID Longitud: ");Serial.print(uidLength, DEC);Serial.println(" bytes");
  Serial.print("UID: "); printArray(uid, uidLength);
  Serial.println("");

  delay(1000);
}
```

Lectura de UID de NFC ISO1443a, Ilustración 45.

```
Serial.print("\nNDEF Record ");Serial.println(i+1);
NdefRecord record = message.getRecord(i);

int payloadLength = record.getPayloadLength();
byte payload[payloadLength];
record.getPayload(payload);
```

Código para lectura de memoria NFC ISO14443a, Ilustración 46.

Sin embargo, al configurar la lectura con autenticación del chip (Bloqueo dinámico y usando la página CC) se tiene que solo por medio del arduino se puede hacer la lectura del transpondedor.

4.3 Discusión de los resultados

Como se evidencio en el numeral anterior google authenticator cuenta con una clave de 128 bits, pero se ve en la prueba que no diferencia minúsculas de mayúsculas y tampoco usa números o caracteres especiales limitando sustancialmente el uso de todas las combinaciones de bits, se basa en SHA-1 cuando puede usar SHA-256 con la funciona HMAC que le permitiría tener un algoritmo de cifrado más robusto, adicional la clave secreta funciona tanto para el primer código como para el segundo código de verificación de forma que si obtenemos esta clave obtendríamos ambos códigos; aunque el algoritmo cuenta con una función de bloqueo en para fuerza bruta es algo básico ya que esta función deniega la sesión cuanto hay 4 intentos en menos de 30 segundos se pudo ver que si pasamos de los 30 segundos entra la función del sistema operativo.

Al habilitar la prevención de un ataque de hombre en el medio se comprueba que así sea desde la misma IP y MAC que se hace una segunda solicitud solo autoriza el ingreso a la primera sesión que ingresa el código del token la segunda sesión que recibe con el mismo toque la deniega al ingresar la contraseña de forma que el atacante no puede identificar que de lo que capturo es válido.

Tal como lo indica el fabricante la contraseña para proteger el chip de la escritura no autorizada es de tipo OTP, ya que después de configurarla no se puede cambiar y muestra que los bits de bloqueo ya fueron alterados, pero al realizarlo por la aplicación se evidencia que la contraseña es de complejidad baja debido que al usar 32 bits indica que solo puede usar 4 caracteres, un carácter ascii tiene 8 bits, ya que la entrada es por el teclado del móvil sin embargo si hacemos la configuración por medio del arduino con el código podemos hacer uso de los 32 bits completos, durante estas pruebas se identifica que usando las etiquetas de autenticación se puede lograr que solo sea leído por el dispositivo que tenga la contraseña y no por todos, es decir, que si no se tiene esto en cuenta a la hora de configurarlo cualquiera con un lector NFC podría sacar la información.

Para que el lector pueda establecer la comunicación el transpondedor debe estar en el punto máximo del campo magnético y a casi 1 cm de alto, esto estando expuesto directamente al lector, debido al poco espacio que tiene bobina en la capsula lo que quiere decir que al estar implantado debe hacer la mano contacto directo con el lector y el xLED que viene el kit presta una gran ayuda para establecer el punto de lectura.

Como ya se conoce, pero cabe resaltar, en la implementación se usa SSH versión 2 lo cual cifra la conexión entre el host y servidor usando AES256 y la función HMAC con SHA256, por lo que desde el ingreso del código del token hasta el código de verificación pasa cifrado el enlace y al hacer un sniffer se ve el tipo de conexión y los paquetes cifrados.

5. Conclusiones

Durante el desarrollo de los capítulos anteriores se implementa un mecanismo de doble factor de autenticación con segunda verificación, el cual se está compuesto de algo que se tiene con el token de google authenticator y con este mismo se obtiene la segunda verificación, con este factor se obtiene protección contra ataques de man in the middle y fuerza bruta ya que el código que entrega está basado en tiempo y va en función de un contador que es la hora y fecha en el servidor y del dispositivo del token, aunque desde una perspectiva de único punto de falla la clave secreta es la misma con la que se genera el código inicial y el código de verificación.

Si bien un punto sensible o de falla más común es la contraseña que establecen los usuarios ya que en su mayoría son inseguras debido a que suelen ser de fácil recordación y por medio de ingeniería social se puede obtener información con la esta se puede deducir, como fechas cumpleaños, nombre de parientes o mascotas, etc; en la actualidad la contraseña sigue siendo el factor por defecto de los mecanismos de autenticación por lo que el transpondedor configurado durante el desarrollo del piloto se almacena una contraseña en chip la cual un usuario no tiene que acordar ni relacionar con algo debido que esta almacenada y protegida por una comunicación NFC cifrada por medio de autenticación entre el lector y el transpondedor, en donde se protege la memoria contra lectura y escritura con contraseñas de forma que solo tendrá acceso los dispositivos de confianza y en caso de que un atacante quiera realizar un análisis para romper las etiquetas de protección el implante está ubicado en la mano y debe haber contacto de la piel (entre el dedo índice y pulgar) contra el lector lo que dificulta que pasen un lector sin que la persona lo note, como en la actualidad sucede con las tarjetas de crédito y los datafonos NFC.

El transpondedor es el segundo método de autenticación el cual complementa el token, dado que añade un nivel de criptografía como de componente hardware adicional siendo un tipo factor derivado de algo que se es con algo se tiene, logrando así incrementar la seguridad en la identificación del usuario, por medio de etiquetas y protocolos que protegen las páginas de la memoria, que se conecta por ssh al servidor; las tecnologías son compatibles – u homologables como el token – con diversos sistemas operativos de forma que en un ambiente empresarial se puede replicar para los usuarios que deben acceder de forma remota a los servidores, aunque debido a lo reciente del tema y que para muchos es un tema que raya con lo futurista y ciencia ficción su aplicación se puede dar en ambientes militares.

Con el chip NTAG216 se puede hacer un desarrollo a futuro para integrar certificados digitales en la memoria y poder validar por medio de estos la identidad los usuarios logrando usar una capa adicional de criptografía, también el desarrollo de un software que permita al lector identificar la aplicación a usar y poder almacenar contraseñas en diferentes paginas para cada una de las aplicaciones, está iniciando la exploración de una tecnología corporal.

6. Bibliografía

M'Raihi, D., Machani, S., Pei, M., & Rydell, J. (2011). *TOTP: Time-Based One-Time Password Algorithm.* IETF.

AndalucíaCERT. (2016). *Uso de autenticación multi-factor en sistemas y aplicaciones I.* Seguridad Andalucia Digital.

Antrax. (18 de 2 de 2010). *Underc0de*. Obtenido de https://underc0de.org/foro/hacking/ataques-de-autenticacion/

Caceres, M. E., Lun Li, Y., & Kamal, M. A. (2013). *Estados Unidos Patente nº US20140256251A1.*

Chakray Consulting S.L. (30 de Mayo de 2017). *¿Qué es el Single Sign on (SSO)? Definición, características y ventajas*. Obtenido de chakray sitio web: https://www.chakray.com/que-es-el-single-sign-on-sso-definicion-caracteristicas-y-ventajas/

Dangerous Things. (s.f.). *transpoders xseries, Dangerous Things.* Obtenido de sitio Dangerous Things: https://dangerousthings.com/shop/xnti/

eset. (4 de Mayo de 2017). *Día de la Contraseña: una breve historia de su origen*. Obtenido de Welivesecurity: https://www.welivesecurity.com/la-es/2017/05/04/dia-de-la-contrasena-origen/

Frutos, A. M. (23 de 07 de 2016). *Computer Hoy*. Obtenido de https://computerhoy.com/noticias/life/que-es-biohacking-48364

Goodin, D. (25 de 6 de 2012). *Arstechnica.* Obtenido de Scientists crack RSA SecurID 800 tokens, steal cryptographic keys: https://arstechnica.com/information-technology/2012/06/securid-crypto-attack-steals-keys/

Luis, E. R. (19 de 09 de 2017). *Urbantecno*. Obtenido de Urbantecno tecnologia: https://urbantecno.com/tecnologia/como-hackear-sms

Mejillon Yagual, F. J., & Villamarin Zambrano, M. J. (2017). *"SEGURIDAD EN TECNOLOGÍA RFID (RADIO FRECUENCY IDENTIFICACTION) Y NFC (NEAR FIELD COMUNICATION). CASO PRACTICO: CLONACION DE TARJETAS DE PROXIMIDAD".* Tesis pregrado, Guayaquil.

NXP Semiconductors. (2013). *NTAG213/215/216 NFC Forum Type 2 Tag compliant IC with 144/504/888 bytes.* Estados Unidos: Company Public.

Nye, C. (2018). Los "biohackers" que transforman su cuerpo con implantes y dietas extremas: "He creado un nuevo sentido humano". *BBC*.

Portero, A., & Linares Pedrero, A. (2013). El arte contemporáneo como proceso de ciborgizacion. *Dialnet*.

Securizando. (24 de Septiembre de 2016). *Securizando.* Obtenido de Blog y podcast con información sobre seguridad informática: https://securizando.com/definiciones/factores-de-autenticacion/

Tatam, R. (25 de abril de 2017). *¿Cuál es la diferencia entre Autenticación de Doble Factor y Multifactor?* Obtenido de Helpsystem web site: https://www.helpsystems.com/es/recursos/articulo/cual-es-la-diferencia-entre-autenticacion-de-doble-factor-y-multifactor

Wexler, A. (2010). The Social Context of “Do-It-Yourself” Brain Stimulation: Neurohackers, Biohackers, and Lifehackers. *Frontiers*.

Anexo A: Instalación de Ubuntu

1. En archivo se da clic en crear una nueva máquina virtual:

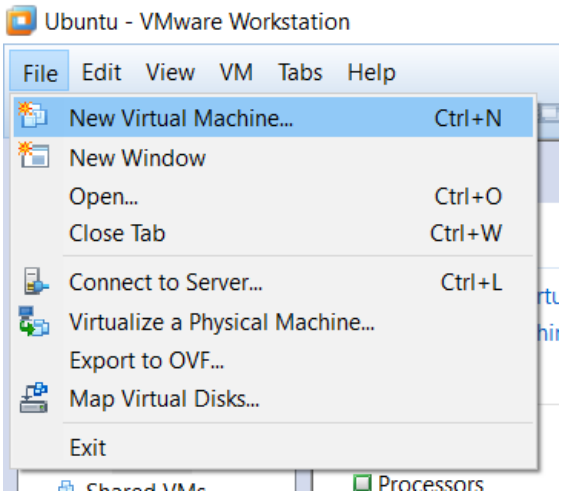


Ilustración 1

2. Seleccionar instalación típica y siguiente:

Ilustración 2

3. Escoger la opción de instalare el sistema operativo después:

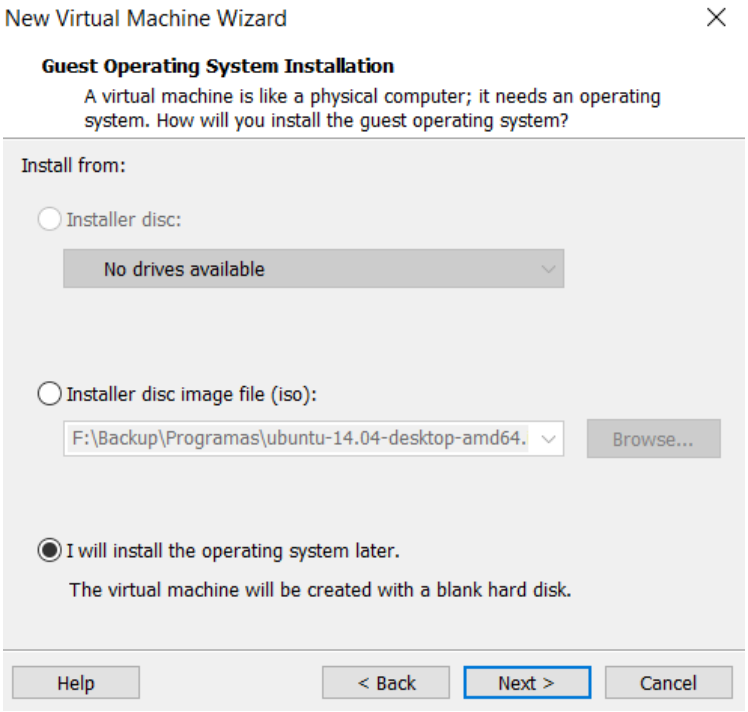


Ilustración 3

4. Seleccionar el sistema Linux y luego Ubuntu 64:

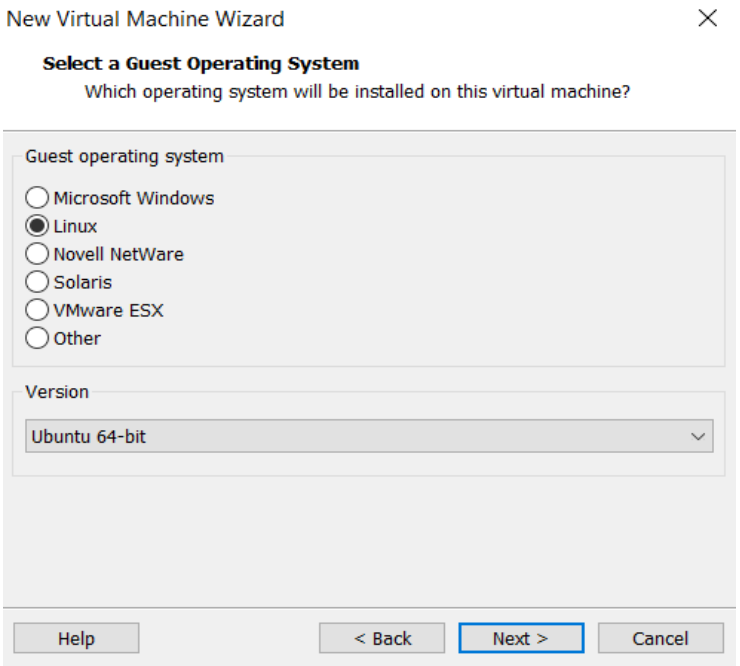


Ilustración 4

5. En nombre de la maquina se asigna uno a la maquina:

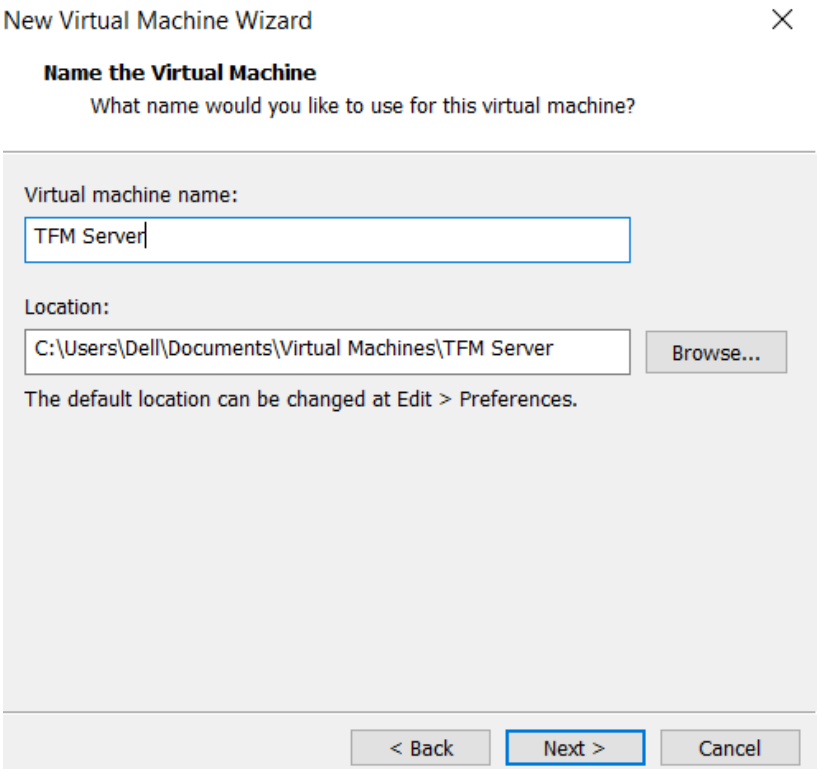


Ilustración 5

6. Para este piloto solo se requiere un servidor ssh por lo que se asigna 40 GB fr disco y se selecciona almacenar el disco en un solo archivo:

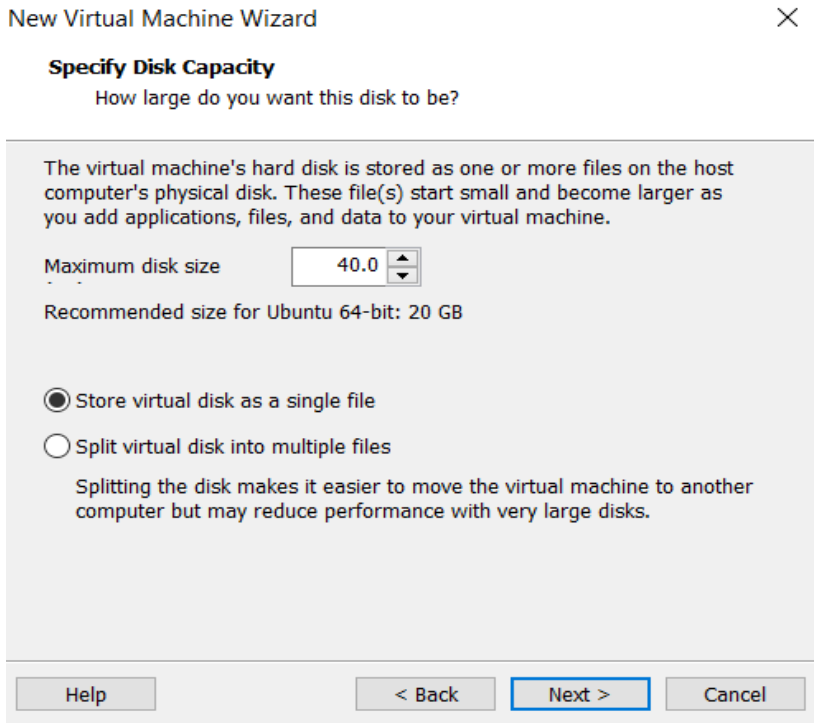


Ilustración 6

7. Se asigna un interfaz en la red de NAT de VMWare, 4 Core virtuales y 3 GB de RAM para que la maquina instale y opere más rápido:

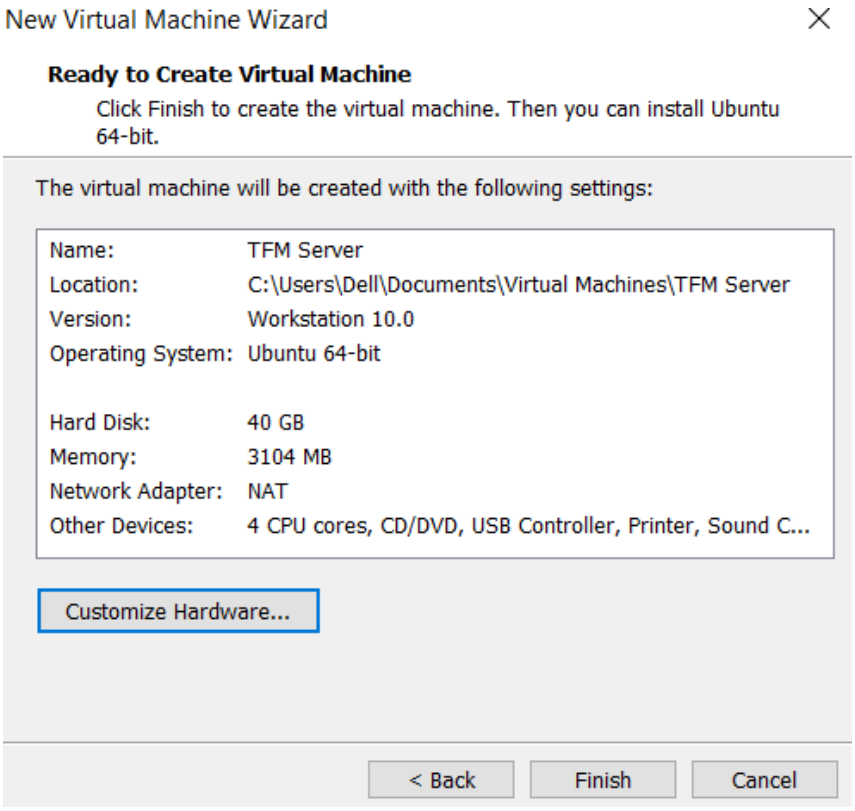


Ilustración 7

8. Luego del proceso de instalación se configura el usuario Giovanni y se ingresa por el monitor del Workstation:

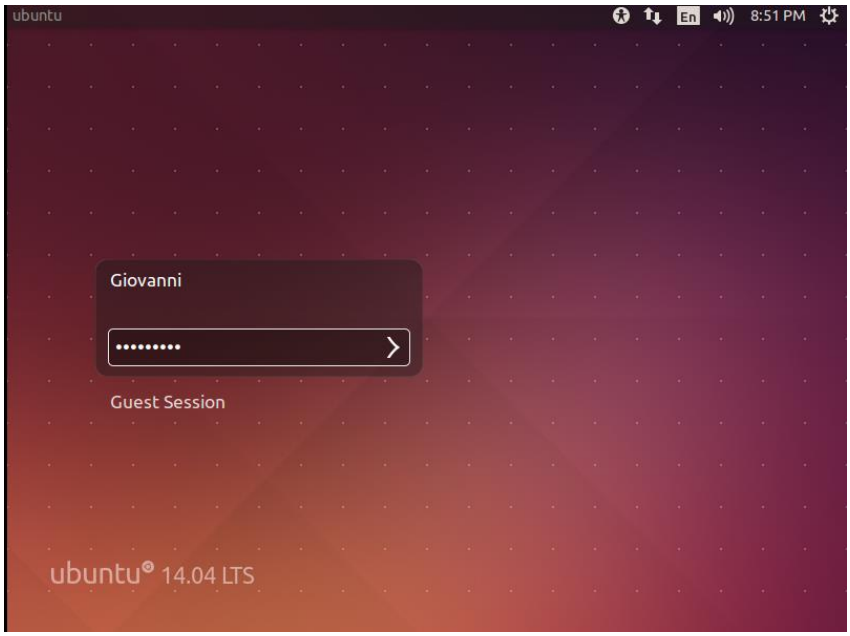


Ilustración 8

9. Desde el host físico se hace prueba y se ingresa por ssh al host virtual satisfactoriamente:

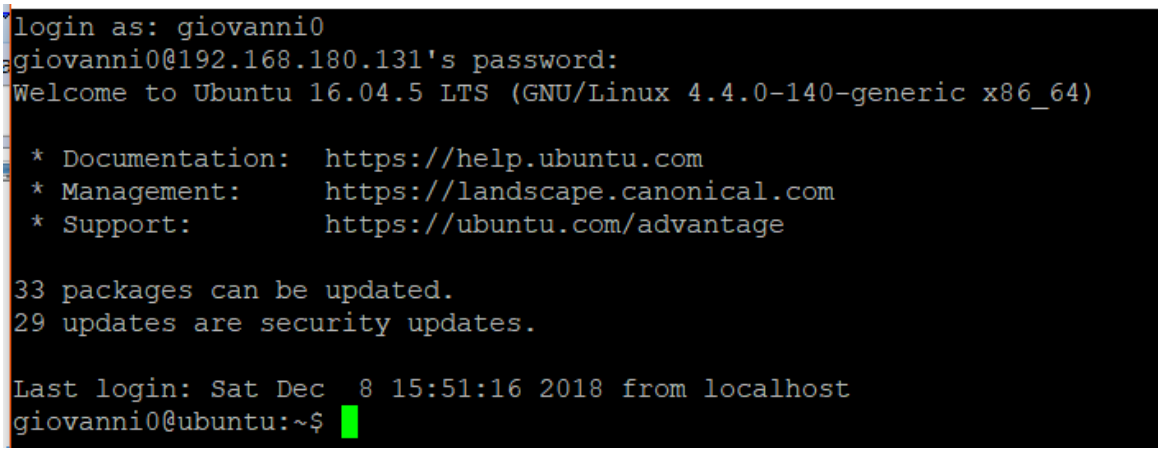


Ilustración 9

Anexo B: Instalación google authenticator

1. Una vez en el terminal del Ubuntu se instalan las librerías de pam de google

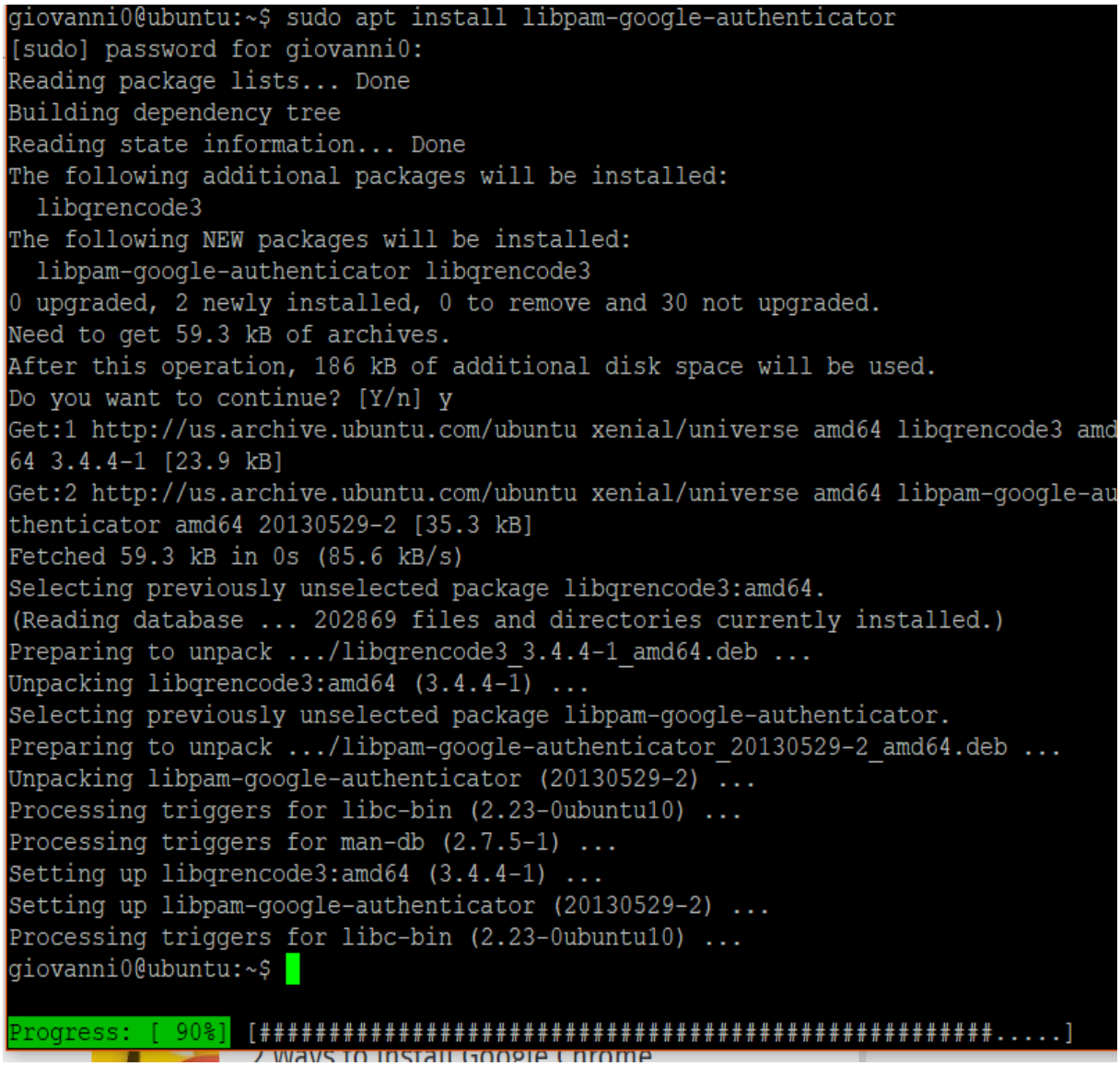


Ilustración 10

2. Una vez finalizado la instalación se procede a la configuración, en donde se generará la clave secrete y los códigos de emergencia, el QR es para leer con la cámara la clave:

Ilustración 11

3. Luego de los códigos la configuración permite habilitar o deshabilitar características de seguridad:

```
Do you want me to update your "/home/giovanni0/.google_authenticator" file (y/n)
 y

Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n)
y

By default, tokens are good for 30 seconds and in order to compensate for
possible time-skew between the client and the server, we allow an extra
token before and after the current time. If you experience problems with poor
time synchronization, you can increase the window from its default
size of 1:30min to about 4min. Do you want to do so (y/n) y

If the computer that you are logging into isn't hardened against brute-force
login attempts, you can enable rate-limiting for the authentication module.
By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting (y/n) y
```

Ilustración 12

4. Se modifica el archivo sshd_config habilitando el challenge-response y el uso del pam estableciendo *yes*:

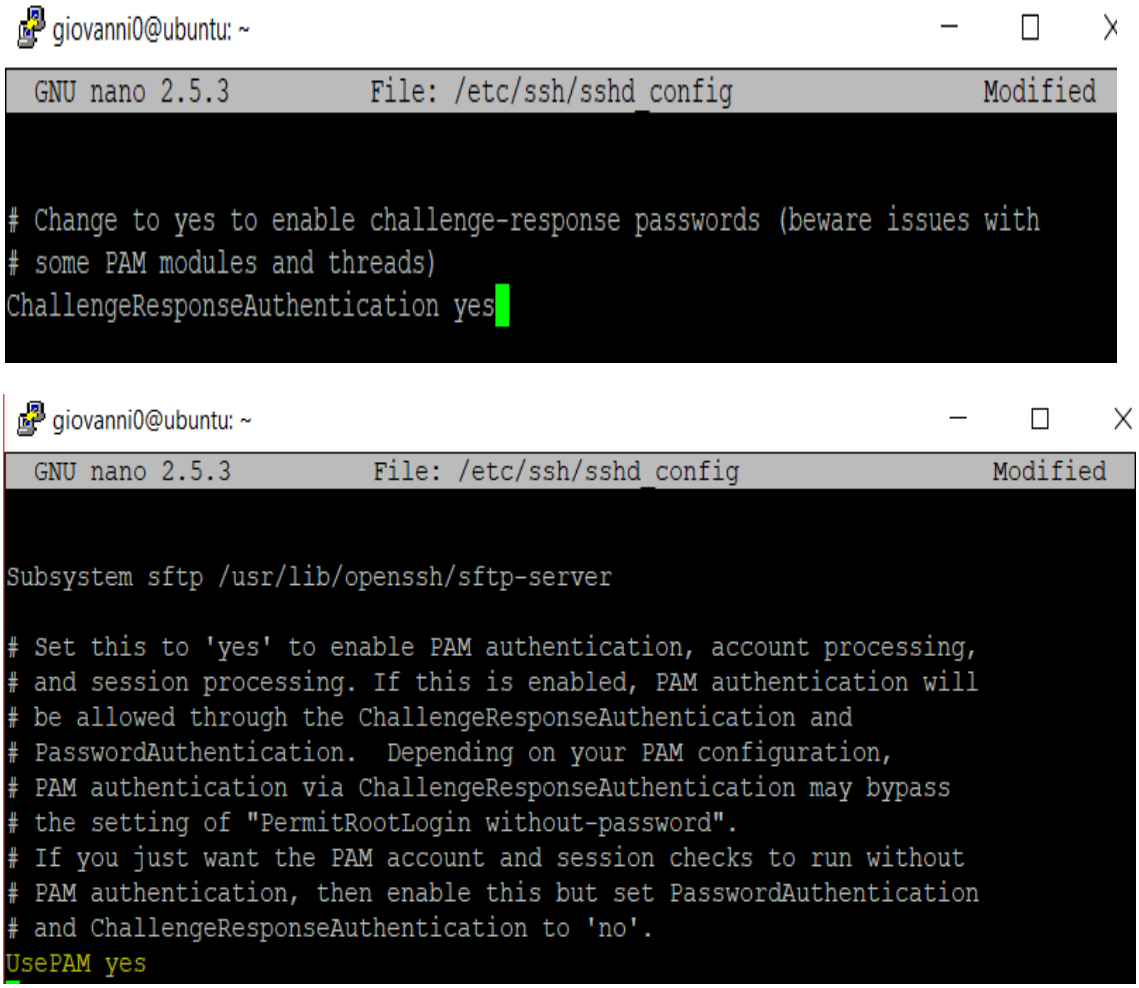


Ilustración 13

5. El plugin se activa en la consola agregando en la línea del archivo login la última línea de la imagen:

```
# Standard Un*x session setup and teardown.
@include common-session

# Print the message of the day upon successful login.
# This includes a dynamically generated part from /run/motd.dynamic
# and a static (admin-editable) part from /etc/motd.
session    optional     pam_motd.so  motd=/run/motd.dynamic
session    optional     pam_motd.so noupdate

# Print the status of the user's mailbox upon successful login.
session    optional     pam_mail.so standard noenv # [1]

# Set up user limits from /etc/security/limits.conf.
session    required     pam_limits.so

# Read environment variables from /etc/environment and
# /etc/security/pam_env.conf.
session    required     pam_env.so # [1]
# In Debian 4.0 (etch), locale-related environment variables were moved to
# /etc/default/locale, so read that as well.
session    required     pam_env.so user_readenv=1 envfile=/etc/default/locale

# SELinux needs to intervene at login time to ensure that the process starts
# in the proper default security context.  Only sessions which are intended
# to run in the user's context should be run after this.
session [success=ok ignore=ignore module_unknown=ignore default=bad]        pa
selinux.so open

# Standard Un*x password updating.
@include common-password
auth required pam_google_authenticator.so
```

Ilustración 14

6. Se copia la clave secreta en la extensión de google authenticator y se comienza a tener el código y si realiza la prueba satisfactoriamente:

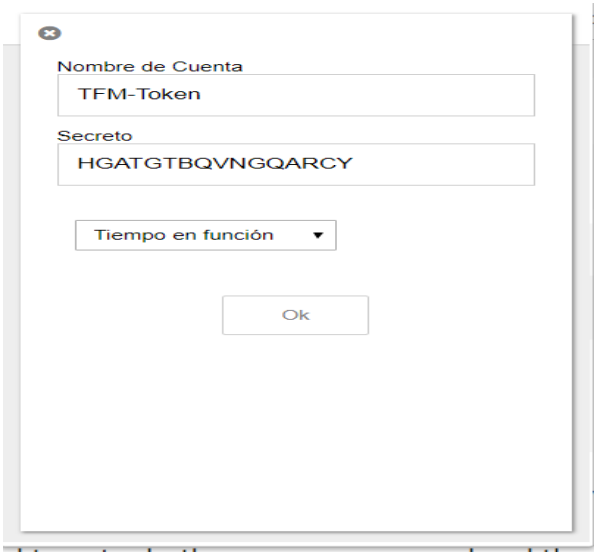


Ilustración 15

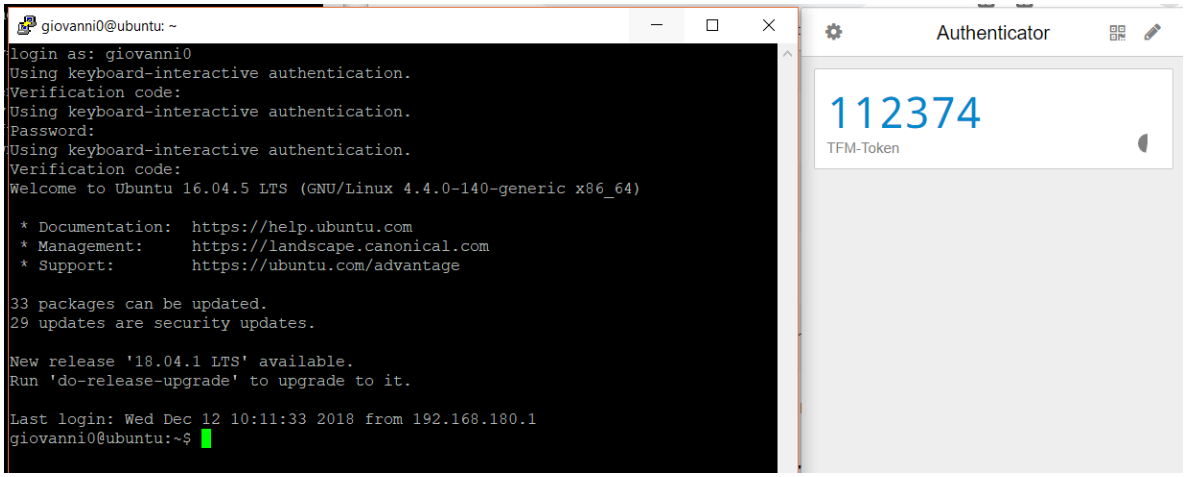


Ilustración 16

Anexo C: Código de Arduino

Las librerías y código fueron descargado de GitHub y modificados a la necesidad del TFM

1. Código de lectura

```
#include <Wire.h>
#include <PN532_I2C.h>
#include <PN532.h>
#include <NfcAdapter.h>

PN532_I2C pn532_i2c(Wire);
NfcAdapter nfc = NfcAdapter(pn532_i2c);

void setup(void) {
  Serial.begin(9600);
  Serial.println("TFM - Biohacking como segundo factor de autenticación");
  Serial.println("Lector Chip transpondedor");
  nfc.begin();
}

void loop(void) {
  Serial.println("\nEscaneando dispositivo NFC\n");

  if (nfc.tagPresent())
  {
    NfcTag tag = nfc.read();
    Serial.println(tag.getTagType());
    Serial.print("UID: ");Serial.println(tag.getUidString());

    if (tag.hasNdefMessage())
```

```
{

  NdefMessage message = tag.getNdefMessage();
  Serial.print("\nMensaje en el dispositivo es ");
  Serial.print(message.getRecordCount());
  Serial.print(" NFC Tag Record");
  if (message.getRecordCount() != 1) {
    Serial.print("s");
  }
  Serial.println(".");

  int recordCount = message.getRecordCount();
  for (int i = 0; i < recordCount; i++)
  {
    Serial.print("\nNDEF Record ");Serial.println(i+1);
    NdefRecord record = message.getRecord(i);

    int payloadLength = record.getPayloadLength();
    byte payload[payloadLength];
    record.getPayload(payload);


    String payloadAsString = "";
    for (int c = 0; c < payloadLength; c++) {
      payloadAsString += (char)payload[c];
    }
    Serial.print("  Information (as String): ");
    Serial.println(payloadAsString);
```

```
        String uid = record.getId();
        if (uid != "") {
          Serial.print("  ID: ");Serial.println(uid);
        }
      }
    }
  }
  delay(10000);
}
```

2. Código de escritura

```
#include <Wire.h>
#include <PN532_I2C.h>
#include <PN532.h>
#include <NfcAdapter.h>

PN532_I2C pn532_i2c(Wire);
NfcAdapter nfc = NfcAdapter(pn532_i2c);

void setup() {
    Serial.begin(9600);
    Serial.println("TFM - Biohacking como segundo factor de autenticación");
    Serial.println("Escritura Transpondedor");
    nfc.begin();
}

void loop() {
      Serial.println("\nAcerque el transpondedor para la escritura");
      if (nfc.tagPresent()) {
```

```
        NdefMessage message = NdefMessage();
        message.addTextRecord("048c0d12ff3884");
      boolean success = nfc.write(message);
      if (success) {
        Serial.println("Caracteres Guardados");
      } else {
        Serial.println("Escritura fallo");
      }
    }
    delay(10000);
}
```

Anexo D: Tabla de comandos de NTAG216

La siguiente tabla de comandos están los comandos que chip reconoce, pero en las librerías de PN532.h ya están las funciones para que solo sean invocadas desde el main.

Nombre de comando	ISO 14443	Comando en hexa
Request	REQA	26h
Wake-up	WUPA	52h
Anticollision	Anticollision	93h 20h
Select	Select	93h 70h
Anticollision	Anticollision	95h 20h
Select	Select	95h 70h
Halt	HLTA	50h 00h
GET_VERSION	-	60h
READ	-	30h
FAST_READ	-	3Ah
WRITE	-	A2h
COMP_WRITE	-	A0h
READ_CNT	-	39h
PWD_AUTH	-	1Bh
READ_SIG	-	3Ch