# Informe de Amenazas CCN-CERT IA-08/15

## Amenazas en BIOS

Marzo de 2015

# INDICE

# 1. SOBRE CCN-CERT

El CCN-CERT (www.ccn-cert.cni.es) es la Capacidad de Respuesta a incidentes de Seguridad de la Información del Centro Criptológico Nacional, CCN. Este servicio se creó en el año 2006 como **CERT Gubernamental Nacional español** y sus funciones quedan recogidas en la Ley 11/2002 reguladora del Centro Nacional de Inteligencia, el RD 421/2004 de regulación del CCN y en el RD 3/2010, de 8 de enero, regulador del Esquema Nacional de Seguridad.

De acuerdo a todas ellas, el CCN-CERT tiene responsabilidad en ciberataques sobre **sistemas clasificados** y sobre sistemas de las **Administraciones Públicas** y de **empresas y organizaciones de interés estratégico** para el país. Su misión, por tanto, es contribuir a la mejora de la ciberseguridad española, siendo el centro de alerta y respuesta nacional que coopere y ayude a responder de forma rápida y eficiente a los ciberataques y a afrontar de forma activa las ciberamenazas.

## 2. RESUMEN EJECUTIVO

El sistema BIOS, acrónimo de *Basic Input Output System* [Ref - 1], es una capa fundamental de los sistemas informáticos encargado de inicializar los controladores de dispositivos *hardware* del sistema, facilitar su interacción con el sistema y transferir el control al sistema operativo. Este sistema se integra en un dispositivo *firmware* dentro de las placas bases de los ordenadores, y es desarrollado tanto por los fabricantes originales de placas base, como por los vendedores independientes que integran diferentes productos proporcionando una solución única. Estos fabricantes comúnmente actualizan este *firmware* para arreglar fallos, posibles vulnerabilidades o dar soporte a nuevo *hardware*.

Actualmente, existen dos tipos de implementaciones de BIOS en el mercado. Por un lado, las implementaciones de BIOS convencionales (o tradicionales), y por otro lado, las modernas implementaciones de BIOS basadas en la interfaz UEFI (*Unified Extensible Firmware Interface*) [Ref - 2]. El uso de la interfaz UEFI garantiza unas especificaciones comunes para los desarrolladores de controladores hardware de bajo nivel para BIOS, facilitando así la interoperabilidad entre diferentes fabricantes y abstrayendo la complejidad de implementación. Sin embargo, el uso de estas especificaciones comunes puede facilitar que el *software* dañino (es decir, *malware*) cuyo objetivo sea infectar las BIOS de los equipos se pueda distribuir de una manera más rápida, infectando a un mayor número de usuarios y sistemas en menos tiempo.

Recuérdese que las modificaciones no autorizadas del sistema BIOS son una amenaza muy importante, dado que el sistema BIOS ocupa un lugar de ejecución único y con los mayores permisos sobre el *hardware* del sistema posibles. El sistema BIOS es el primer código del sistema que se ejecuta, y puede hasta modificar el contenido del disco duro, modificando por ejemplo parámetros de seguridad del sistema operativo, o incluso cambiando contraseñas de acceso al mismo. Por ello, normalmente este tipo de *malware* corresponde con Amenazas Persistentes Avanzadas (APT, del inglés Advanced Persistent Threats), que pueden acabar en una denegación de servicio (el sistema BIOS queda inservible, y por ende, no es capaz de arrancar), o una presencia persistente y resistente a reinstalaciones de sistema operativo.

Algunos de los ataques más conocidos a los sistemas BIOS son, por un lado, el ataque realizado por el virus Chernobyl, que tuvo su auge en 1998 (también conocido como CIH o Spacefiller). Este virus se instalaba en la memoria del sistema operativo Windows, e interceptaba las llamadas de acceso a ficheros, siendo capaz de propagarse además por todos los procesos en ejecución. Después, intentaba borrar el sistema BIOS sobrescribiendo la memoria ROM con valores nulos. Otra versión de este virus atacaba el Master Boot Record (MBR) del disco duro del mismo modo (sobrescribiendo con ceros), con lo que inhabilitaba la carga del sistema operativo.

Por otro lado, otro de los ataques a sistemas BIOS más actual es el que se llevó a cabo con Mebromi, que actuó en diversos períodos del año 2012. Este *software* dañino instalaba un *rootkit* tanto en la BIOS, como en el MBR. Concretamente, infectaba las BIOS convencionales del fabricante Award, modificando el MBR para alterar la carga

del sistema operativo de Windows instalando de manera persistente otro programa dañino que actuaba como troyano.

Recientemente, en una ponencia durante las VIII Jornadas STIC CCN-CERT D. Barroso describió y repasó los métodos de infección y persistencia en BIOS sucedidos a lo largo de los años [Ref - 3].

En este documento se explica, en primer lugar, en más detalle cómo funciona el sistema BIOS, tanto el tradicional como el UEFI. Después, se describen los posibles vectores de ataque al sistema BIOS, así como las soluciones de seguridad que se pueden aplicar en estos sistemas. Por último, se resumen las conclusiones sobre la seguridad de los sistemas BIOS que un administrador de sistemas de una administración pública, una empresa o un usuario particular debe de tener en cuenta para evitar que sea una potencial víctima de ataques a estos sistemas.

Para un lector que desee profundizar más en el concepto de seguridad BIOS tanto a nivel de sistemas de escritorio como a nivel de sistemas de servidores, se recomienda la consulta de las guías publicadas por el NIST [Ref – 4, Ref - 5].

## 3. SISTEMA BIOS: TIPOS Y FUNCIONAMIENTO

Algunos de los fabricantes de BIOS más conocidos son, entre otros, American Megatrends, WinBond, Phoenix, AMI, IBM, Award y Asus. Estos fabricantes normalmente integran todos los componentes de una placa base, así como los controladores de todo el *hardware* que va integrado.

La BIOS, como se ha comentado anteriormente, tiene un papel esencial en el proceso de carga de un equipo informático. Así, inicializa todo el sistema *hardware* que lo compone, gestionando y regulando incluso la temperatura de la CPU durante el arranque. Las responsabilidades/hitos de actuación de un sistema BIOS se pueden resumir en cinco grandes puntos:

- **Confianza establecida**: la BIOS es responsable de verificar la integridad de todos los componentes *hardware* en el sistema, así como de comprobar su autenticidad antes de usarlo.
- **Test de *hardware***: la BIOS tiene que inicializar y comprobar todo el *hardware* presente en el equipo antes de usarlo. La propia placa base, el *chipset*, o la memoria son elementos que se comprueban durante este test. Esta fase de test se conoce como Power-On-Self-Test (POST).
- **Carga de módulos adicionales**: algunos de los dispositivos del equipo pueden requerir la carga de controladores adicionales para poder realizar su función. Es responsabilidad de la BIOS asegurarse de que esos módulos están correctamente cargados y ejecutándose. Estos módulos pueden estar guardados tanto en el propio chip de la BIOS, como en algún otro dispositivo de almacenamiento secundario (e.g., un chip específico dentro de la placa del propio *hardware* adicional).
- **Seleccionar el dispositivo para arranque**: una vez que está todo correcto, la BIOS se encarga de detectar todos aquellos dispositivos válidos para

arrancar (e.g., disco duro, CD/DVD, o USB). Una vez que se encuentra un dispositivo válido, procede a ejecutar el "bootloader" en ese dispositivo (es decir, transfiere el control de ejecución al dispositivo seleccionado).

- **Carga del Sistema Operativo**: Tras el *bootloader*, comienza la ejecución y carga del núcleo del sistema operativo en memoria. Cuando éste ya se ha inicializado de manera correcta, el sistema BIOS cede el control al propio sistema operativo, acabando así su trabajo.

A continuación, se comentan en más detalle estos procesos en función de los diferentes sistemas BIOS actuales. En la actualidad, existen dos tipos de BIOS comerciales: la BIOS convencional [Ref - 1] (también llamada BIOS legada), y la BIOS UEFI [Ref - 2].

## 3.1 Sistema BIOS convencional

En esta sección se detalla el modelo de funcionamiento de una BIOS convencional. La Figura 1 muestra el proceso de arranque de un ordenador con una BIOS convencional, siendo el paso del tiempo representado en la progresión horizontal. Obsérvese que la primera actividad que realiza el ordenador es la inicialización de la BIOS. Este proceso, a su vez, se puede subdividir en tres procesos: el proceso "BIOS boot block", que realiza las operaciones necesarias para poder empezar el arranque del ordenador; el proceso "Power-On-Self-Test" (POST), que se encarga de comprobar, identificar e inicializar los dispositivos del sistema como la propia CPU, la memoria RAM, el sistema de interrupciones y otro tipo de controladores *hardware* que disponga la máquina en concreto. Este proceso únicamente se ejecuta tras un arranque en frío (primer arranque, o tras la pulsación del botón de reinicio del ordenador); y el proceso "Option ROMs", que se encarga de cargar en memoria y ejecutar otros controladores que se encuentren en memorias no volátiles de los dispositivos *hardware* del ordenador.

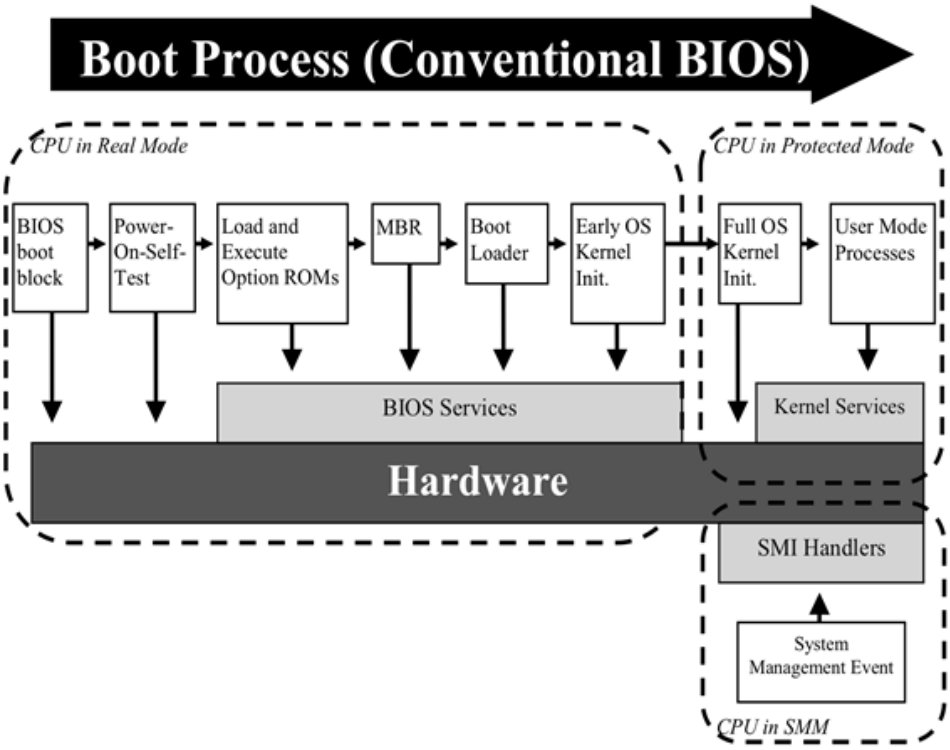



**Figura 1. Proceso de arranque de un ordenador con BIOS convencional (extraída de [Ref - 4]).**

Tras este proceso de inicialización, el ordenador carga el "Master Boot Record", o MBR [Ref - 6], que es un registro localizado comúnmente en el primer sector de un disco duro e identifica cómo y dónde se encuentra el sistema operativo a ejecutar, dentro del propio disco duro. Después, tras esta lectura se procede a ejecutar el "Boot Loader", que es un pequeño programa dependiente del sistema operativo e indica al ordenador cómo se ha de realizar la carga del propio sistema operativo. Así, el "Boot Loader" se encarga de empezar la carga del sistema operativo mediante la ejecución de código que inicializa convenientemente una versión mínimamente funcional del núcleo del sistema operativo. Esta versión se encarga de cambiar el estado de la CPU, que hasta entonces se estaba ejecutando en modo real, a modo protegido. Tras ello, se acaba de inicializar y cargar una versión completa del núcleo del sistema operativo. Una vez finaliza la carga del sistema operativo, éste toma el control de la CPU y empieza la ejecución en modo usuario. Es decir, el ordenador ya está listo para poder ser usado por un usuario cualquiera.

Analizando el proceso de carga descrito, se observa la criticidad que tiene una BIOS en el mismo: la BIOS es el primer código que se ejecuta durante un arranque, siendo además en modo real de ejecución. Por lo tanto, una BIOS comprometida puede poner en riesgo todo el sistema, ya que como se ha descrito, el código de la BIOS es el encargado de transferir el control al disco duro, para empezar su carga, y de éste al sistema operativo, para finalizar con un núcleo de sistema operativo totalmente cargado y funcional para el usuario.

## 3.2 Evolución a los sistemas BIOS UEFI

Actualmente, los sistemas de BIOS convencionales han evolucionado hacia los sistemas BIOS UEFI (Unified Extensible Firmware Interface del inglés) [Ref - 2]. UEFI es una interfaz estándar diseñada para reemplazar a los sistemas BIOS convencionales, y con el objetivo de mejorar la interoperabilidad de *software* y solucionar las limitaciones de las BIOS convencionales. Entre las soluciones que proporciona UEFI destacan, por un lado, la posibilidad de poder trabajar con discos duros modernos. En los sistemas de BIOS convencionales, como se ha descrito anteriormente, se necesitaba trabajar con el MBR. Este sistema impone una restricción física al tamaño máximo de discos duros donde puede operar, concretamente, hasta 2TiB (1TiB son 1024GiB) y no más de cuatro particiones por disco. Así, UEFI soporta un nuevo esquema de particiones denominado GPT (GUID Partition Table, del inglés). GPT permite un particionado de hasta 128 particiones por disco, con una capacidad teórica total de 8ZB (recuérdese que 1ZiB es igual a 3096TiB). Actualmente, los sistemas BIOS que trabajan con discos GPT no son capaces de arrancar en discos grandes. No obstante, esta limitación acabará siendo solucionada en el corto plazo con la propia evolución de la microelectrónica interna del ordenador. En sistemas de 64bits permite, además, trabajar con cantidades de RAM en torno a los $2^{64}$ GiB (máxima cantidad teórica que se puede direccionar con 64 bits).

UEFI también permite un mayor grado de integración entre el sistema operativo y el entorno de arranque. Por ejemplo, Windows 8 se aprovecha de esta característica en sus Opciones Avanzadas de Arranque [Ref - 7]: en un sistema UEFI, Windows 8 permite seleccionar el dispositivo desde el que se quiere arrancar, sin necesidad de configurar la BIOS directamente.

Del mismo modo, una BIOS UEFI permite un mayor control de la configuración del *hardware* de bajo nivel, como son la interfaz gráfica, el teclado o el ratón, por nombrar algunos. Otro de los beneficios de UEFI es que permite trabajar con *hardware* legado, es decir, no existen incompatibilidades con *hardware* que actualmente está funcionando en un sistema de BIOS convencional.

En realidad, UEFI es un sistema operativo mínimamente funcional que se sitúa entre el propio *hardware* del equipo y sus propios controladores. Se diferencia también de la BIOS convencional respecto a su almacenamiento: en vez de almacenarse en *firmware*, como la BIOS convencional, el código de UEFI se guarda en el directorio /EFI/ en memoria no volátil. Es decir, el sistema UEFI puede residir en una memoria NAND en la placa base, en un disco duro, o incluso en una unidad de red compartida.

Una de las mejores funcionalidades que trae UEFI es el aumento de su seguridad. Véase que los sistemas BIOS convencionales, como se ha explicado anteriormente, son un sistema crítico en un ordenador debido a su campo de actuación. Sin embargo, la BIOS de los equipos informáticos, tanto de un usuario medio como en una gran corporación, raramente son actualizadas. Esto lleva a que exista *software* dañino (es decir, *malware*) cuyo objetivo es infectar las BIOS de equipos legítimos con el fin de conseguir acceso al mismo total y transparente.

Una de estas características de UEFI y seguridad relacionada con el último sistema operativo de Microsoft, Windows 8, es el sistema denominado *Secure Boot*. Este sistema asegura que únicamente sistemas operativos autorizados puedan arrancar en el ordenador. *Secure Boot* trabaja mediante la lectura de una firma criptográfica que se encuentra en el *bootloader* del SO, verificando que esta firma se encuentra entre las claves autorizadas guardadas en el *firmware* UEFI. En los equipos modernos *Secure Boot* se encuentra activado por defecto, y suele ser un inconveniente al instalar otro sistema operativo, como GNU/Linux o MacOS X, ya que el sistema BIOS se niega a arrancar como mecanismo de seguridad.

De este modo, *Secure Boot* evita que *malware* tipo *rootkit* infecte el *bootloader* y se convierta en controlador del propio sistema operativo. Así, si un *rootkit* ha infectado el *bootloader*, su firma no será reconocida al arranque y se parará su ejecución, dejando el equipo infectado en un estado de parada permanente.

El sistema *Secure Boot* se puede deshabilitar en arquitecturas x86, permitiendo así el arranque de otros sistemas operativos como se ha comentado anteriormente. Sin embargo, en arquitecturas ARM esta inhabilitación no es posible, con lo que en plataformas como tabletas Windows RT no es posible la instalación de otro sistema operativo de entorno móvil como Android o Linux.

En la Figura 2 se muestra el proceso de arranque de estos nuevos sistemas de BIOS. Una explicación más detallada de todo el arranque de UEFI se puede consultar en [Ref - 8]. El arranque de un sistema de BIOS UEFI se divide en cuatro fases. En la primera fase, llamada *Security* (SEC), es el núcleo de la cadena de confianza en el arranque del equipo. Así, en esta fase se comprueba la integridad de todos los módulos que se van a ejecutar. Es decir, todos los módulos *firmware* necesarios para inicializar el procesador, la placa base, el *chipset*, etc., han de ser previamente comprobados. Después, sucede la fase de pre-inicialización EFI, o *Pre - EFI Initialization* (PEI). Durante esta fase se inicializan todos los componentes clave del sistema para poder empezar su ejecución: procesador, placa base, y *chipset*; y se prepara la inicialización de la siguiente fase. La siguiente fase, llamada *Driver Execution* (DXE), es la fase donde se buscan todos los controladores *hardware* que se tengan que ejecutar, acabando así el proceso de inicialización de todo el *hardware* integrado en el equipo. Es durante esta fase cuando se comprueba que los módulos *hardware* que se van a ejecutar estén autorizados, mediante el uso de firmas criptográficas de integridad.

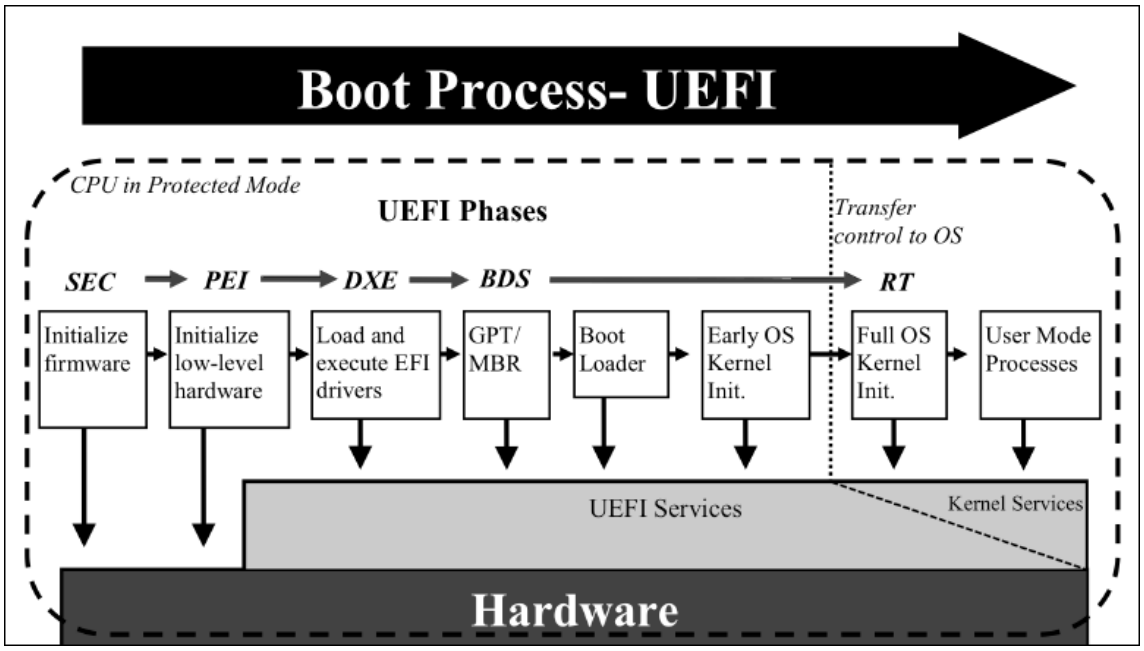



**Figura 2. Proceso de arranque de un ordenador con BIOS convencional (extraída de [Ref - 4]).**

Así, este mecanismo evita que se pueda instalar cualquier amenaza en este nivel, como por ejemplo, controladores de teclado modificados para capturar las pulsaciones de teclado, controladores de vídeo modificados para recolectar información de un usuario, o incluso controladores de acceso a disco y memoria modificados que permitieran la recuperación de información sensible almacenada en disco memoria.

Por último, la fase *Boot Device Selection* (BDS) se encarga de inicializar los dispositivos de consola del sistema (para operaciones de entrada/salida), así como otros dispositivos como interfaces remotas (e.g., Telnet, o interfaz gráfica remota sobre HTTP). También se cargan los controladores adicionales para gestionar los dispositivos de arranque, que finalmente se cargan (MBR o GPT), para finalizar con la carga del sistema operativo.

En esta fase de BDS es donde entra en juego también el sistema *Secure Boot* explicado anteriormente. *Secure Boot* supone la raíz de la cadena de confianza en el *firmware* de la máquina. La cadena de confianza se basa en certificados x509. Existe un certificado raíz de entidad autorizada (*root CA*) incluido en el *firmware* que valida el *bootloader*, quien después valida el núcleo mínimo del sistema operativo, quien valida el núcleo completo antes de su carga, etcétera. Así, se definen cuatro diferentes bases de datos para proveer una mayor flexibilidad sobre las claves:

- **DB (base de datos de firmas).** Esta base de datos contiene todas las llaves y *hashes* confiables que se pueden usar para la autenticación de cualquier aplicación o módulo que se ejecuten en el entorno UEFI.
- **DBX (lista negra de firmas).** Esta base de datos contiene una lista de claves y *hashes* no confiables, de modo que cualquier aplicación o módulo que esté

firmado por estas claves o coincida su *hash* con alguno que se encuentre en esta lista, se bloqueará su ejecución.

- **KEK (base de datos de intercambio de claves).** Esta base de datos contiene el conjunto de claves de confianza válidas para poder actualizar las bases de datos DB y DBX.
- **PK (llave de plataforma, también llamada llave pública única).** Esta clave, implementada como base de datos a pesar de ser una única clave, define la clave con la que tienen que estar firmadas las actualizaciones que afecten a la base de datos KEK.

Así, UEFI sugiere unas recomendaciones sobre las claves incluidas por defecto en estas bases de datos. Concretamente, recomienda que:

- La clave del fabricante OEM de la placa base (e.g., Award, Phoenix, o Asus, entre otros) esté en la PK.
- Las claves de los sistema operativos estén tanto en KEK como en DB.
- Los fabricantes OEM también pueden insertar ciertas claves o hashes que necesiten en KEK y en DB.

Cabe destacar que determinados fabricantes de sistemas UEFI permiten la modificación de las claves contenidas en estas bases de datos a partir del propio UEFI.

## 3.3 Diferencias entre sistemas BIOS convencionales y sistemas BIOS UEFI

A modo de resumen, se detallan aquí las diferencias entre sistemas BIOS convencionales y sistemas BIOS UEFI. Mejor funcionalidad, aumento de eficiencia, y un mecanismo de seguridad durante el arranque de un equipo informático serían las más destacables:

- **Capacidad de direccionamiento mayor**: las BIOS convencionales estaban normalmente limitadas a trabajar en un modo de 16bits, lo que suponía una limitación de máximo 1MiB en la capacidad de direccionamiento. UEFI permite la ejecución en 32 y 64 bits, con lo que se amplía considerablemente la capacidad de direccionamiento. Esto ha hecho que aparezcan programas de configuración de BIOS mucho más grandes y sofisticados.
- **Mejora de seguridad**: UEFI incorpora varios mecanismos de seguridad no presentes en los sistemas de BIOS convencionales. El sistema *Secure Boot*, comentado anteriormente, es uno de los ejemplos. También permite disponer de una infraestructura de llave pública y criptografía básica, de modo que únicamente los módulos debidamente autorizados o denegados serán ejecutados o no durante el arranque.
- **Independiente de la arquitectura de la CPU**: UEFI se desarrolló para ser completamente independiente de la arquitectura del procesador, lo que le permite ejecutarse en muchas arquitecturas diferentes (e.g., x86, IA-64, o ARM, entre otras).

- **Entorno de ejecución más potente**: UEFI permite disponer de un entorno de ejecución mucho más potente, disponiendo de características como el arranque a través de la red, el uso del ratón, control de ACPI o incluso navegar por Internet – todo ello previamente a la carga del sistema operativo.
- **Rendimiento mejorado**: el uso de una BIOS UEFI con un sistema operativo que la soporta mejora tanto el proceso de arranque como el proceso de apagado del sistema.

# 4. POSIBLES AMENAZAS A SISTEMAS BIOS

En esta sección se explican las amenazas a sistemas BIOS más comunes. Recuérdese que el sistema BIOS está escrito en dispositivos de almacenamiento no volátiles, como memorias EEPROM. Esto permite que el contenido de la ROM pueda ser sobrescrito cuando se actualizan ciertos fallos o el propio *software* de la BIOS. Sin embargo, del mismo modo esta peculiaridad puede ser aprovechada por agentes dañinos que dispongan de la suficiente habilidad como para modificar la BIOS, siempre y cuando consigan suficientes privilegios de acceso. A continuación, se distinguen entonces los diferentes escenarios de amenazas posibles.

## 4.1 Ataques iniciados por el usuario

En este escenario, el ataque es realizado por un usuario que usa un fichero no autentificado para actualizar el sistema BIOS. Nótese que este escenario puede sucederse tanto por un usuario que desconoce la intención dañina del fichero de actualización, como por un usuario que lo realiza de forma intencionada.

## 4.2 Ataques iniciados por *software* dañino

Es posible que *software* dañino sea capaz de explotar alguna vulnerabilidad en un sistema BIOS desde el propio sistema operativo. De este modo, el atacante puede llegar a instalar una versión de BIOS vulnerable y controlable, consiguiendo así tener acceso a todo el proceso de arranque del equipo comprometido, lo que pone en riesgo la seguridad de todo el sistema.

## 4.3 Ataques basados en red

En este tipo de escenario, un atacante consigue acceso y compromete directamente un servidor de actualización de sistemas BIOS. Así, es capaz de controlar e infectar de manera indiscriminada todos aquellos sistemas que se nutran del servidor comprometido.

# 5. MECANISMOS DE SEGURIDAD EN SISTEMAS BIOS UEFI

A continuación, se explican los diferentes mecanismos de seguridad aplicables en sistemas BIOS UEFI con el fin de mitigar los ataques comentados en la sección anterior. Nótese que a pesar de que el sistema UEFI propone una interfaz común y única para los fabricantes, cada uno de ellos realiza diferentes personalizaciones, con lo que la localización exacta de las opciones que se presentan aquí pueden diferir, con lo que se recomienda consultar la guía de usuario del fabricante de su BIOS para una mayor comprensión de la distribución de los menús y su contenido.

## 5.1 Instalación de contraseña de usuario y contraseña de administrador

Los sistemas BIOS UEFI permiten la introducción de dos contraseñas diferentes: "Administrator" (Administrador, algunos fabricantes lo denominan "Supervisor"), y "User" (Usuario). La contraseña de administrador (o supervisor) restringe el acceso a la pantalla de configuración de la BIOS. La contraseña de usuario permite restringir el acceso al usuario después de que el sistema BIOS ha finalizado su proceso de arranque, pero antes de que se cargue el sistema operativo. Es decir, permite controlar quién accede al sistema operativo del equipo. Nótese que después, el propio sistema operativo puede tener configurado el acceso mediante cuentas de usuario y contraseña adicionales (por lo tanto, esta es una medida de seguridad de acceso físico).

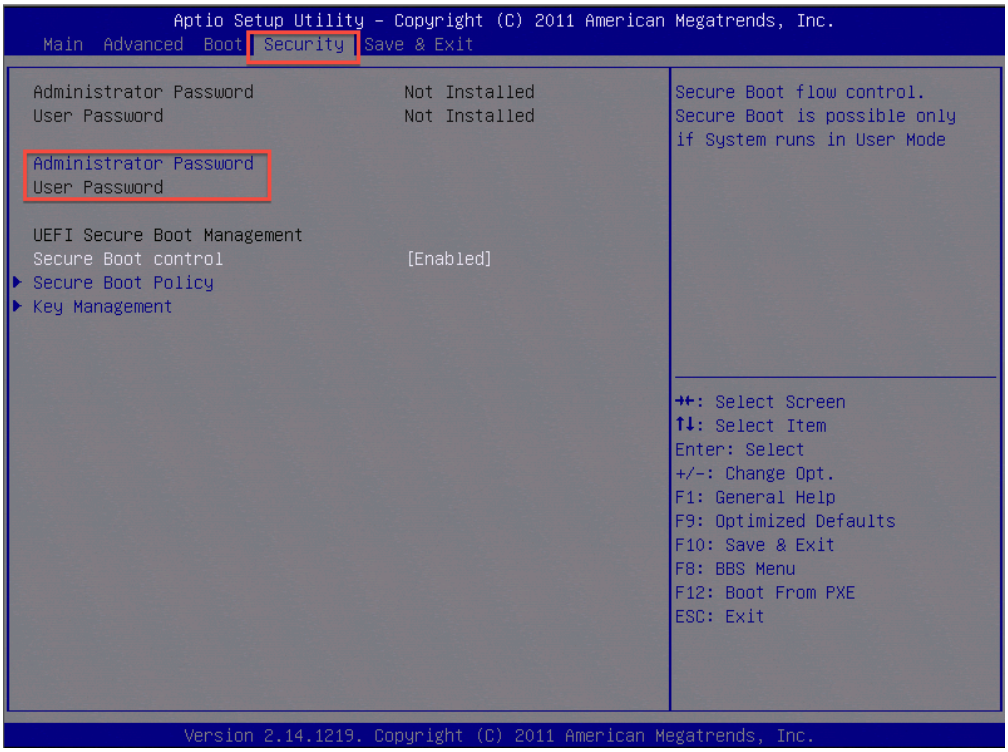
Se recomienda en primer lugar la instalación de una contraseña de administrador, para evitar que usuarios no autorizados puedan acceder al sistema BIOS y configurarlo de manera insegura. En caso de requerir una mayor seguridad en el sistema, se recomienda la instalación de una contraseña de usuario para obligar a la presencia física de los usuarios durante el arranque del sistema.

Se muestran estas opciones de configuración en dos sistemas UEFI diferentes en la Figura 3. Concretamente, se trata de dos sistemas BIOS del mismo fabricante (American Megatrends), pero diferentes placas base.

## 5.2 Habilitar la carga de módulos adicionales compatibles con UEFI

Como se ha descrito anteriormente, durante el proceso de arranque del sistema, UEFI busca todos los módulos adicionales del *hardware* y los carga en ejecución, siempre y cuando se verifique que la firma del módulo o aplicación coincide con lo esperado. Sin embargo, por compatibilidad con *hardware* legado los sistemas BIOS UEFI actuales permiten interactuar con *hardware* no UEFI, es decir, sus módulos o aplicaciones son cargados sin verificar su autenticidad ni integridad. Un ejemplo de esta configuración se muestra en la Figura 4.

Se recomienda, en la medida de lo posible, habilitar únicamente los módulos PCI compatibles con UEFI. De ese modo, únicamente se podrán cargar aquellos módulos cuya autenticación e integridad estén garantizadas.



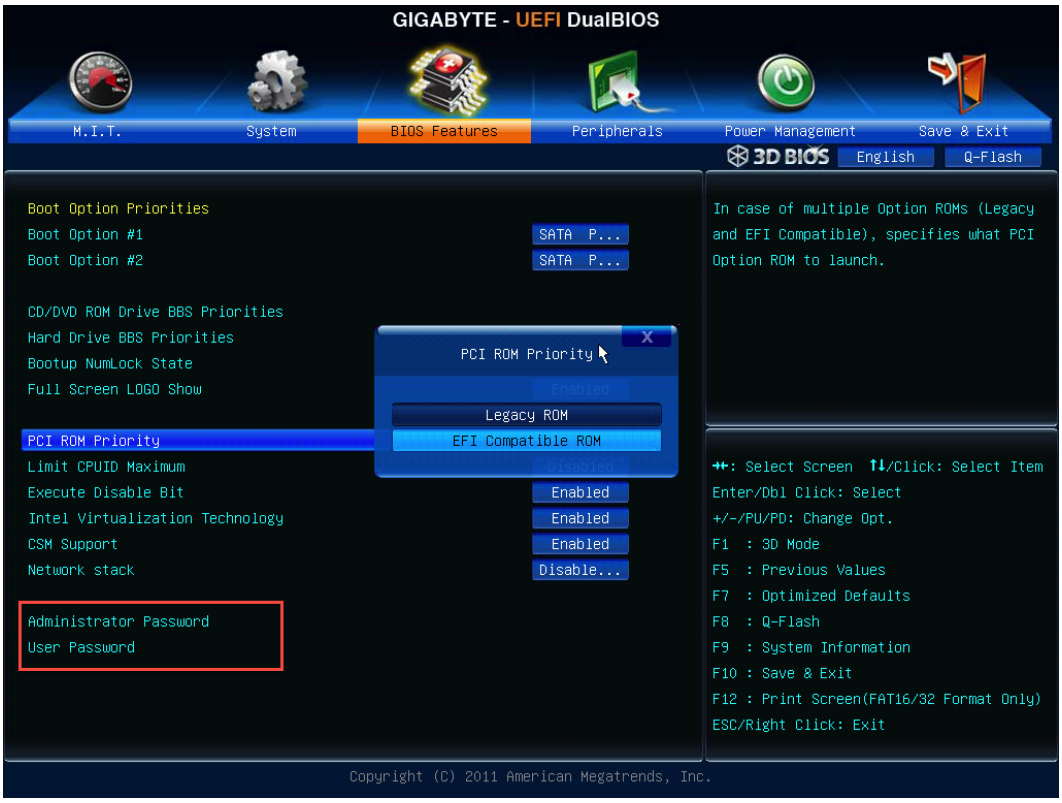




**Figura 3. Instalación de contraseñas de administrador y usuario en el sistema BIOS UEFI.**


**Figura 4. Opción de carga de módulos de hardware adicionales compatibles con UEFI.**

## 5.3 Activación de sistema *Secure Boot*

El sistema de seguridad proporcionado por *Secure Boot* se ha explicado con anterioridad en la Sección 3.2. Recuérdese que este sistema permite la ejecución del *bootloader* de sistemas operativos cuya firma esté reconocida, bloqueando su ejecución en otro caso. Este comportamiento es configurable desde el menú del sistema BIOS UEFI. En la Figura 5 se muestra esta opción de configuración en una BIOS American Megatrends.

Se recomienda que esta opción esté activada para poder evitar posibles amenazas al sistema que atenten contra el *bootloader* del sistema operativo.

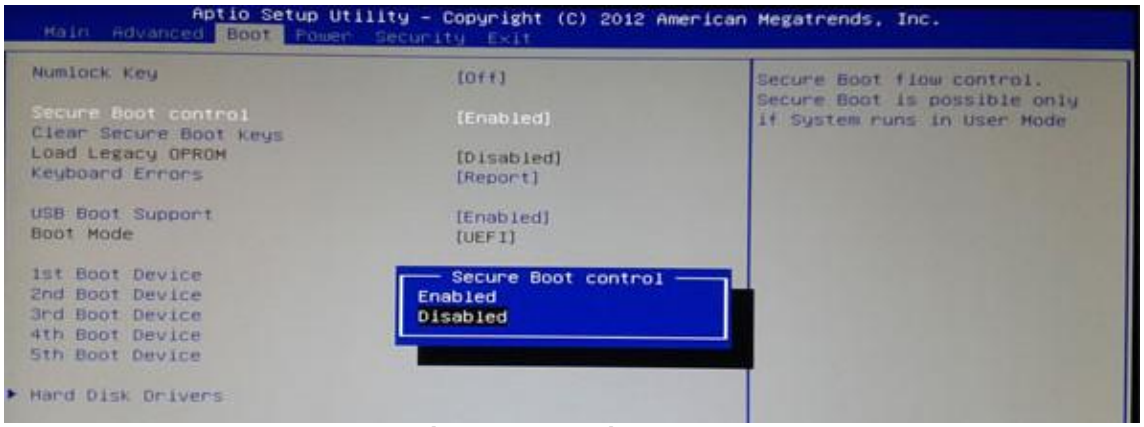



**Figura 5. Activación/desactivación del sistema Secure Boot.**

## 5.4 Modo de arranque UEFI

Como se ha comentado con anterioridad, por compatibilidad hacia atrás los sistemas BIOS UEFI soportan *hardware* legado, y por lo tanto, existen opciones de configuración dentro del sistema BIOS para habilitar este soporte. En la Sección 5.2 se ha hablado del soporte de módulos *hardware* legados. Aquí, se muestra que existen determinados sistemas BIOS UEFI que tienen la opción de habilitar el arranque del equipo como si fuera una BIOS convencional (véase la Sección 3.1). Este tipo de sistemas se denominan sistemas BIOS UEFI duales.

En la Figura 6 se muestra esta opción de configuración en un sistema BIOS UEFI de America Megatrends. Se recomienda, por seguridad, siempre que el *hardware* de que se dispone sea compatible, seleccionar el modo de arranque UEFI (explicado en detalle en la Sección 3.2).

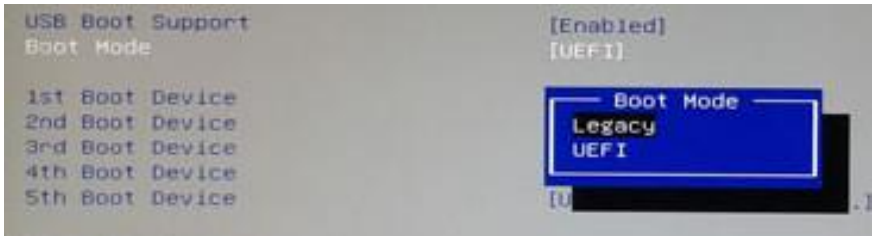



**Figura 6. Selección de modo de arranque legado/UEFI.**

## 5.5 Configuración de bases de datos de claves del sistema *Secure Boot*

El sistema de seguridad *Secure Boot*, proporcionado por los sistemas BIOS UEFI, hace uso de una serie de claves y firmas para la habilitación o inhabilitación de ejecución de ciertos módulos (véase la Sección 3.2). Algunos de los sistemas BIOS UEFI permiten configurar estas claves, consiguiendo así un soporte de *hardware* adicional configurable por el usuario (el usuario puede insertar las claves o firmas de los módulos *hardware* a los que se permite o deniega su ejecución). Esta característica añade mucha más personalización a los sistemas del usuario, pero puede ser usado por un atacante interno para la introducción de claves o firmas de módulos modificados con intenciones dañinas. Para evitar esto último, se recomienda la instalación en los equipos de contraseñas de acceso al sistema BIOS UEFI (véase la Sección 5.1).

En la Figura 7 se muestran estas opciones de configuración de claves de *Secure Boot* en un sistema BIOS UEFI de American Megatrends. En caso de no requerir ningún módulo de *hardware* incompatible con EFI, o de desarrollo propio, se recomienda dejar las claves que vienen por defecto.

## 5.6 Opciones de red UEFI

Recuérdese que el sistema BIOS UEFI permite el arranque desde diversos dispositivos, incluso desde red. En caso de no estar usando esta última característica de UEFI, y el proceso de arranque se realice desde local, se recomienda encarecidamente deshabilitar la opción relativa a red UEFI ya que a día de hoy se desconoce si esta característica puede ser explotable con intenciones dañinas.

En la Figura 8 se muestra esta opción, en una placa base Gigabyte usando un sistema BIOS UEFI de American Megatrends.

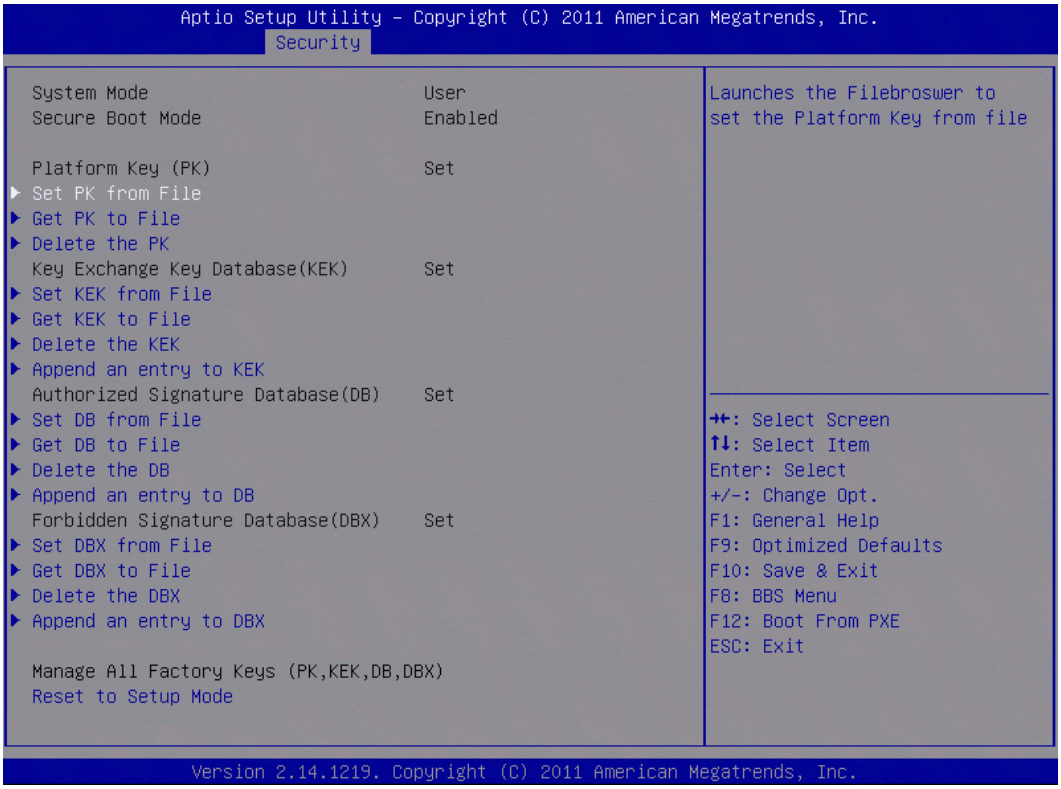



**Figura 7. Opciones para la introducción de claves relativas a *Secure Boot* (véase la Sección 3.2).**


**Figura 8. Desactivación de la pila de red usada por el sistema UEFI.**

# 6. CONCLUSIONES

Los sistemas BIOS están encargados del arranque de un sistema informático. El proceso de arranque difiere si se trata de un sistema de BIOS convencional, o un sistema de BIOS UEFI. Este último es una evolución de los sistemas BIOS convencionales, y aportan una serie de ventajas respecto a los sistemas convencionales, como una mejora del rendimiento, un diseño por capas que facilita la independencia de la arquitectura con que se esté trabajando, y una gran mejora de los mecanismos de seguridad. Estos mecanismos de seguridad se han incorporado para evitar los ataques de *software* dañino que atentan contra los sistemas BIOS. Recuérdese que estos sistema controlan el arranque del equipo, con lo que si éste se compromete, un atacante tendría control total del equipo, a una persistencia difícil de detectar y de eliminar. En este informe se han detallado los procesos de arranque de sistemas BIOS convencionales y sistemas BIOS UEFI, así como destacado sus principales diferencias. Del mismo modo, se han mostrado los posibles escenarios de ataque frente a estos sistemas, así como los mecanismos de seguridad que ofrece UEFI y que deben de configurarse para poder disponer de un sistema BIOS securizado frente a estos ataques.

# 7. INFORMACIÓN ADICIONAL

Se puede ampliar cualquier punto del informe, utilizando como vía preferente de comunicación el portal http://www.ccn-cert.cni.es

- Teléfono: +34 91372 59 74
- E-Mail: ccn-cert@cni.es

## ANEXO A. REFERENCIAS


| | |
|---|---|
| [Ref – 1] | **Basic Input/Output System (BIOS). Definición en Wikipedia**<br>Página web<br>http://es.wikipedia.org/wiki/BIOS |
| [Ref – 2] | **Extensible Firmware Interface (EFI). Definición en Wikipedia**<br>Página web<br>http://es.wikipedia.org/wiki/Extensible_Firmware_Interface |
| [Ref – 3] | **Infección en BIOS, y derivados (ponencia de VIII Jornadas STIC CCN-CERT)**<br>D. Barroso (ElevenPaths)<br>https://www.ccn-cert.cni.es/publico/VIII_Jornadas/03-Persistencia_BIOS_DavidBarroso.pdf |
| [Ref – 4] | **NIST SP800-147: BIOS Protection Guidelines**<br>D. Cooper, W. Polk, A. Regenscheid, M. Souppaya. Abril 2011<br>http://csrc.nist.gov/publications/nistpubs/800-147/NIST-SP800-147-April2011.pdf |
| [Ref – 5] | **NIST SP800-147B: BIOS Protection Guidelines for Servers**<br>A. Regenscheid. Julio 2012<br>http://csrc.nist.gov/publications/drafts/800-147b/draft-sp800-147b_july2012.pdf |
| [Ref – 6] | **Registro de arranque principal (MBR). Definición en Wikipedia**<br>Página web<br>http://es.wikipedia.org/wiki/Registro_de_arranque_principal |
| [Ref – 7] | **Opciones de Arranque Avanzadas de Microsoft Windows**<br>Página web<br>http://windows.microsoft.com/es-es/windows/advanced-startup-options-including-safe-mode |
| [Ref – 8] | **UEFI and the TPM: Building a foundation for platform trust**<br>Página web. 17 de Noviembre de 2011<br>http://resources.infosecinstitute.com/uefi-and-tpm/ |