“Yes, I am a criminal. My crime is that of curiosity. My crime is that of judging people by what they say and think, not what they look like. My crime is that of outsmarting you, something that you will never forgive me for.”

(“Si, soy un criminal. Mi crimen es la curiosidad, juzgar a la gente por lo que dice y piense, no por su aspecto. Mi crimen es superarte, algo por lo que nunca me vas a perdonar.”)

Extraido del Manifiesto Hacker,
The Mentor, 1986

Resumen Factual - Enigform

*** PROBLEMA: La autenticación de usuarios utiliza usuario-clave. El SSL es caro en ancho de banda y costo (excepto Cacert). Nadie le presta atención a los alertas de certificado inválido.**

*** REALIDAD: PGP. 19 años. Ampliamente utilizado en email, distribución de software, etc.**

*** Se basa en una llave pública y otra privada. Una se comparte, la otra se protege. Implementa firma, encriptación y sus contrapartes.**

*** IDEA: Utilizar el estándar OpenPGP con HTTP.**

*** BENEFICIO: Fuerte autenticación de usuarios y servers sin demasiadas complicaciones extra. Contenido cacheable. Funciona sobre SSL también.**

Resumen Factual - Enigform

* IMPLEMENTACIÓN ACTUAL: Addon para Firefox, Modulo de Apache, Plugin de Wordpress.

* Diseñado de tal manera que es fácil reescribir un plugin de autenticación HTTP básica para funcionar con Enigform. El plugin de Wordpress fue adaptado de dicha manera en menos de 10 minutos.

* Existe el potencial de lograr solicitudes HTTP encriptadas, encapsulables, funcionar como TOR incluso.

* Kevin Kiley, autor de mod_gzip ha contribuído código para implementar dicha funcionalidad.

Resumen Factual – Enigform

* Funciona perfectamente en MS Windows (Firefox, Apache para Windows, GPGME, todos disponibles). Testeado en Solaris. Funciona en cualquier plataforma que tenga GPGME, Apache, etc.

* Mediante Apache mod_proxy_http, cualquier servicio HTTP que se encuentre por detrás puede usar Enigform.

* Gratuito, Libre y Abierto

Introducción

"Que no seas paranoico... ¡No significa que no te estén persiguiendo!"

El uso más conocido del estándar OpenPGP es para enviar correo electrónico encriptado, o para firmar digitalmente un archivo cuya autenticidad debe poder ser autenticada posteriormente.

Los eMails también puede firmarse, y ambos métodos pueden combinarse.

Incluso existen implementaciones de OpenPGP para sistemas de mensajería instantánea, agregando así una muy necesaria capa de seguridad a dichos sistemas.

De HTTP no hay nada que no conozcan. Espero.

¿OpenPGP?

* Standard definido en RFC 2440 -> Compatibilidad entre implementaciones.

* Define las operaciones de Encriptado, Firmado y sus contrapartes Desencriptado y Verificación.

* Dichas operaciones son apropiadamente combinables.

* Existen Plugins para entornos de administración de archivos (Explorer, KDE/Konqueror, etc)

* Existen Plugins para sistemas de eMail basados en GUI o WEB (Outlook, Thunderbird, Gmail)

* Muy usado como alternativa más segura de MD5SUM, SHA-1/2/etc.

Un par de llaves

* Al ser instalado, requiere la creación de un keypair.

* Este par de llaves contiene un elemento público y otro privado.

* La llave pública se comparte de diversas maneras. (HTTP, HKP, eMail, reuniones de keysigning)

* La llave privada no debe caer en manos de otra persona.

* Para usar la llave privada, se requiere ingresar un passphrase, o frase secreta, que la desencripta.

* Si podemos verificar que una llave pública pertenece a cierta persona, podemos firmarla y enviársela. Otra gente ve esta relacion de confianza. Web-of-Trust.

Frase secreta

* La llave privada debe ser protegida a toda costa y contra todo riesgo. Los keypair pueden expirar.

* La frase secreta puede modificarse cuantas veces uno lo crea conveniente.

* Para la operación de firmado y/o desencriptado, se requiere el uso de la llave privada.

* Por ejemplo, para leer un eMail encriptado a nosotros, debemos introducir la frase secreta, que desencripta la llave privada.

* Para evitar ingresar la frase secreta, se utiliza un “Agente” (gpg-agent).

Avanzando

Efectivamente, una vez comprendidos los conceptos básicos de uso de OpenPGP, y la forma de utilizar la implementación (GnuPG, PGP, etc) que deseemos, podemos encontrar en OpenPGP más usos de los clásicos:

*** Plugins OpenPGP para Mensajería Instantánea.**
*** Plugins OpenPGP para IRC.**
*** Extensiones OpenPGP a sistemas de distribución de paquetes (apt, urpmi, Portage).**

*** Extensiones a otros protocolos, como HTTP... ;) ;)**

Request y Cabeceras

* HTTP 1.1 definido en RFC 2616.
* Un Request tiene un Method Line, Headers y un Body opcional.

```
== inicio request ejemplo SUPER limitado ==
GET /login.php?user=buanzo&pass=algunaclaveREsegura HTTP/1.1
Host: www.buanzo.com.ar
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.2)
Referer: http://www.buanzo.com.ar/

== fin request ejemplo ==
```

* Dicho request no tiene body.

* Las cabeceras, según RFC, se usan para enviar más información sobre el request mísmo, o el cliente.

* En una cabecera se puede agregar una firma PGP

Método GET, POST y respuesta

* El request anterior usa el verbo/metodo “GET”.

* Solicita a www.buanzo.com.ar un archivo: /login.php

* El archivo recibe parámetros (QUERY STRING): user y pass.

* Indica vía User-Agent el navegador utilizado.

* También indica desde qué URL se originó el request. (por ejemplo, llenando un formulario de login desde /)

* Un request real posee como una docena de headers.

* La respuesta es Protocolo+Estado, headers, línea en blanco y el cuerpo (opcional dependiendo del método)

Análisis de un request firmado (Enigform)

Si firmaramos dicho request usando Enigform, se vería así:

```
== inicio request ejemplo ==
GET /login.php?user=buanzo&pass=2600 HTTP/1.1
Host: www.buanzo.com.ar
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.2)
Referer: http://www.buanzo.com.ar/
X-OpenPGP-Type: S
X-OpenPGP-Sig-Fields: body
X-OpenPGP-Sig: iD8DBQFGKQDIAlpOsGhXcE0RApmdAJ4yrOLyYUT
X-OpenPGP-Digest-Algo: SHA1
X-OpenPGP-Version: GnuPG v1.4.6 (GNU/Linux)
X-OpenPGP-Agent: Enigform 0.8.4 for Mozilla Firefox

== fin request ejemplo ==
(Las Líneas largas fueron cortadas)
```

Verificación de dicho request (mod_openpgp)

Se utilizan las cabeceras agregadas para recrear un modelo de datos compatible con OpenPGP, y se agregan cabeceras especiales al request.

Dependiendo del método (GET o POST), la información a verificar será el QUERY STRING o el BODY del request.

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

username=buanzo&password=2600

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.6 (GNU/Linux)

iD8DBQFF8AlhAlpOsGhXcE0RAnxZAJ42HuV1x1qn5wlwCfcu5J
rCggK6hcww5aLkL9Gu0yXOQ=
=FZa4
-----END PGP SIGNATURE-----
```

Inicio Seguro de Sesión

* En vez de enviar usuario y clave a través de un formulario, Enigform detecta un pedido de inicio de sesión al sitio remoto.

* Mediante operaciones OpenPGP, se verifica la identidad del cliente y del servidor remoto.

* El server provee un hash de sesión encriptado para el cliente.

* Ambos almacenan dicho hash.

* De ahora en mas cada request a dicho server anexará el hash de sesión.

* Dependiendo de la validez de la sesión, el valor de la cabecera X-Auth-OpenPGP-Session será valid, expired o invalid.

Proteccion contra ataques HTTP

* Un request firmado que viaja en limpio puede ser capturado y reutilizado por un atacante ("Replay attack").

→ Por este motivo, se descartan requests correctamente firmados que no pertenezcan a una sesión válida.

→ Por el mísmo motivo, el mecanismo de inicio seguro de sesión se basa en Challenge-Response.

* En la próxima version de Enigform, se encriptan las variables de un metodo GET, no el request entero o el canal (a diferencia de SSL).

* Si se detecta que un request de cierta sesión viene de una nueva IP, se inicia una revalidacion.

SSL y HTTP-OpenPGP

* Secure HTTP (https://) es una solución que permite tráfico encriptado y autenticación de identidad de servidor y cliente.

* Requiere un tunel entre cliente y server, pasando por todos los proxies que pudiera haber.

* Casi no se le presta atención a los certificado inválido. Se usan muy poco los certificados de cliente en servicios web generales.

* Los certificados son caros, y usan modelo de Jerarquía de Confianza (CertiCUAC da un certificado, y confiamos porque CertiCUAC lo dice).

* Enigform es una extension a HTTP, por lo que no tiene problemas para correr CON SSL!

HTTP encriptado con OpenPGP

* Se puede encriptar todo un request, encapsularlo en un request “esqueleto”.

* Mediante un Input Filter de apache, se desencripta el request y se procesa el verdadero request.

* La respuesta debería encapsularse, encriptada, en una Respuesta HTTP.

* No existen Input Filters para Firefox, o IE: se complica.

* Se puede usar un sistema de proxy para encriptar y desencriptar.

* Al menos existe SSL!

Possible “Encrypted Request”

```
POST /HTTP_OPENPGP_DECRYPT HTTP/1.0
Host: localhost
Connection: Close
Content-Length: 370

-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.7 (GNU/Linux)
hQIOA9YKl/p/3dcgEAf/erCrgwG8kB3
26nA4WU2ZmB3i5ZP4aaZwKZulsBh
hfJ0mgGFIGwl+uFyQoCwXk33H5j1lJ
lqKlLHtccLuG4fAAYfviqLmuK8vpFcb
oYYkiLgJ+fDDxDwGAZ/6ryAN3tlPUyq
TDPOAzpQM0ho385J4xv1ZfQRajSRY
a/YBe4SGJyjyvDXxpUMhbftZMKDML
3E08dzok+uvoLVDBJ7wpFhYACcguX
N1cR2yQFqiuR+S6ycEo/qEL2XNM3r
=OPnL
-----END PGP MESSAGE-----
```

Ejemplo de Uso – Home Banking

* Un cliente del Banco Feliz instala el plugin Enigform.

* Crea (excepto que ya tenga) su par de llaves OpenPGP.

* Su llave pública es enviada a una red SKS, o al keyserver del banco.

* Con su tarjeta del banco, visita un cajero automático y vincula su cuenta con su llave pública. Se emite un ticket que es presentado en caja (por ejemplo).

* Ahora puede ingresar en forma segura, incluso sobre SSL, al Home Banking. Otros métodos de ingreso pueden tener menos privilegios.

Gracias por su tiempo.

¡Adiós!

Arturo 'Buanzo' Busleiman
buanzo@buanzo.com.ar
www.buanzo.com.ar
wiki.buanzo.org

“Yes, I am a criminal. My crime is that of curiosity. My crime is that of judging people by what they say and think, not what they look like. My crime is that of outsmarting you, something that you will never forgive me for.”

(“Si, soy un criminal. Mi crimen es la curiosidad, juzgar a la gente por lo que dice y piense, no por su aspecto. Mi crimen es superarte, algo por lo que nunca me vas a perdonar.”)

Extraido del Manifiesto Hacker,
The Mentor, 1986

Resumen Factual - Enigform

*** PROBLEMA: La autenticación de usuarios utiliza usuario-clave. El SSL es caro en ancho de banda y costo (excepto Cacert). Nadie le presta atención a los alertas de certificado inválido.**

*** REALIDAD: PGP. 19 años. Ampliamente utilizado en email, distribución de software, etc.**

*** Se basa en una llave pública y otra privada. Una se comparte, la otra se protege. Implementa firma, encriptación y sus contrapartes.**

*** IDEA: Utilizar el estándar OpenPGP con HTTP.**

*** BENEFICIO: Fuerte autenticación de usuarios y servers sin demasiadas complicaciones extra. Contenido cacheable. Funciona sobre SSL también.**

Resumen Factual - Enigform

*** IMPLEMENTACIÓN ACTUAL: Addon para Firefox, Modulo de Apache, Plugin de Wordpress.**

*** Diseñado de tal manera que es fácil reescribir un plugin de autenticación HTTP básica para funcionar con Enigform. El plugin de Wordpress fue adaptado de dicha manera en menos de 10 minutos.**

*** Existe el potencial de lograr solicitudes HTTP encriptadas, encapsulables, funcionar como TOR incluso.**

*** Kevin Kiley, autor de mod_gzip ha contribuído código para implementar dicha funcionalidad.**

Resumen Factual – Enigform

*** Funciona perfectamente en MS Windows (Firefox, Apache para Windows, GPGME, todos disponibles). Testeado en Solaris. Funciona en cualquier plataforma que tenga GPGME, Apache, etc.**

*** Mediante Apache mod_proxy_http, cualquier servicio HTTP que se encuentre por detrás puede usar Enigform.**

*** Gratuito, Libre y Abierto**

Introducción

"Que no seas paranoico... ¡No significa que no te estén persiguiendo!"

El uso más conocido del estándar OpenPGP es para enviar correo electrónico encriptado, o para firmar digitalmente un archivo cuya autenticidad debe poder ser autenticada posteriormente.

Los eMails también puede firmarse, y ambos métodos pueden combinarse.

Incluso existen implementaciones de OpenPGP para sistemas de mensajería instantánea, agregando así una muy necesaria capa de seguridad a dichos sistemas.

De HTTP no hay nada que no conozcan. Espero.

¿OpenPGP?

*** Standard definido en RFC 2440 -> Compatibilidad entre implementaciones.**

*** Define las operaciones de Encriptado, Firmado y sus contrapartes Desencriptado y Verificación.**

*** Dichas operaciones son apropiadamente combinables.**

*** Existen Plugins para entornos de administración de archivos (Explorer, KDE/Konqueror, etc)**

*** Existen Plugins para sistemas de eMail basados en GUI o WEB (Outlook, Thunderbird, Gmail)**

*** Muy usado como alternativa más segura de MD5SUM, SHA-1/2/etc.**

Un par de llaves

*** Al ser instalado, requiere la creación de un keypair.**

*** Este par de llaves contiene un elemento público y otro privado.**

*** La llave pública se comparte de diversas maneras. (HTTP, HKP, eMail, reuniones de keysigning)**

*** La llave privada no debe caer en manos de otra persona.**

*** Para usar la llave privada, se requiere ingresar un passphrase, o frase secreta, que la desencripta.**

*** Si podemos verificar que una llave pública pertenece a cierta persona, podemos firmarla y enviársela. Otra gente ve esta relacion de confianza. Web-of-Trust.**

Frase secreta

*** La llave privada debe ser protegida a toda costa y contra todo riesgo. Los keypair pueden expirar.**

*** La frase secreta puede modificarse cuantas veces uno lo crea conveniente.**

*** Para la operación de firmado y/o desencriptado, se requiere el uso de la llave privada.**

*** Por ejemplo, para leer un eMail encriptado a nosotros, debemos introducir la frase secreta, que desencripta la llave privada.**

*** Para evitar ingresar la frase secreta, se utiliza un “Agente” (gpg-agent).**

Avanzando

Efectivamente, una vez comprendidos los conceptos básicos de uso de OpenPGP, y la forma de utilizar la implementación (GnuPG, PGP, etc) que deseemos, podemos encontrar en OpenPGP más usos de los clásicos:

*** Plugins OpenPGP para Mensajería Instantánea.**
*** Plugins OpenPGP para IRC.**
*** Extensiones OpenPGP a sistemas de distribución de paquetes (apt, urpmi, Portage).**

*** Extensiones a otros protocolos, como HTTP... ;) ;)**

Request y Cabeceras

*** HTTP 1.1 definido en RFC 2616.**
*** Un Request tiene un Method Line, Headers y un Body opcional.**

```
== inicio request ejemplo SUPER limitado ==
GET /login.php?user=buanzo&pass=algunaclaveREsegura HTTP/1.1
Host: www.buanzo.com.ar
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.2)
Referer: http://www.buanzo.com.ar/

== fin request ejemplo ==
```

*** Dicho request no tiene body.**

*** Las cabeceras, según RFC, se usan para enviar más información sobre el request mísmo, o el cliente.**

*** En una cabecera se puede agregar una firma PGP**

Método GET, POST y respuesta

* El request anterior usa el verbo/metodo “GET”.

* Solicita a www.buanzo.com.ar un archivo: /login.php

* El archivo recibe parámetros (QUERY STRING): user y pass.

* Indica vía User-Agent el navegador utilizado.

* También indica desde qué URL se originó el request. (por ejemplo, llenando un formulario de login desde /)

* Un request real posee como una docena de headers.

* La respuesta es Protocolo+Estado, headers, línea en blanco y el cuerpo (opcional dependiendo del método)

Análisis de un request firmado (Enigform)

Si firmaramos dicho request usando Enigform, se vería así:

```
== inicio request ejemplo ==
GET /login.php?user=buanzo&pass=2600 HTTP/1.1
Host: www.buanzo.com.ar
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.2)
Referer: http://www.buanzo.com.ar/
X-OpenPGP-Type: S
X-OpenPGP-Sig-Fields: body
X-OpenPGP-Sig: iD8DBQFGKQDIA1pOsGhXcE0RApmdAJ4yrOLyYUT
X-OpenPGP-Digest-Algo: SHA1
X-OpenPGP-Version: GnuPG v1.4.6 (GNU/Linux)
X-OpenPGP-Agent: Enigform 0.8.4 for Mozilla Firefox

== fin request ejemplo ==
(Las Líneas largas fueron cortadas)
```

Verificación de dicho request (mod_openpgp)

Se utilizan las cabeceras agregadas para recrear un modelo de datos compatible con OpenPGP, y se agregan cabeceras especiales al request.

Dependiendo del método (GET o POST), la información a verificar será el QUERY STRING o el BODY del request.

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

username=buanzo&password=2600

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.6 (GNU/Linux)

iD8DBQFF8AlhAlpOsGhXcE0RAnxZAJ42HuV1x1qn5wlwCfcu5J
rCggK6hcww5aLkL9Gu0yXOQ=
=FZa4
-----END PGP SIGNATURE-----
```

Inicio Seguro de Sesión

* En vez de enviar usuario y clave a través de un formulario, Enigform detecta un pedido de inicio de sesión al sitio remoto.

* Mediante operaciones OpenPGP, se verifica la identidad del cliente y del servidor remoto.

* El server provee un hash de sesión encriptado para el cliente.

* Ambos almacenan dicho hash.

* De ahora en mas cada request a dicho server anexará el hash de sesión.

* Dependiendo de la validez de la sesión, el valor de la cabecera X-Auth-OpenPGP-Session será valid, expired o invalid.

Proteccion contra ataques HTTP

* Un request firmado que viaja en limpio puede ser capturado y reutilizado por un atacante ("Replay attack").

  → Por este motivo, se descartan requests correctamente firmados que no pertenezcan a una sesión válida.

  → Por el mísmo motivo, el mecanismo de inicio seguro de sesión se basa en Challenge-Response.

* En la próxima version de Enigform, se encriptan las variables de un metodo GET, no el request entero o el canal (a diferencia de SSL).

* Si se detecta que un request de cierta sesión viene de una nueva IP, se inicia una revalidacion.

SSL y HTTP-OpenPGP

*** Secure HTTP (https://) es una solución que permite tráfico encriptado y autenticación de identidad de servidor y cliente.**

*** Requiere un tunel entre cliente y server, pasando por todos los proxies que pudiera haber.**

*** Casi no se le presta atención a los certificado inválido. Se usan muy poco los certificados de cliente en servicios web generales.**

*** Los certificados son caros, y usan modelo de Jerarquía de Confianza (CertiCUAC da un certificado, y confiamos porque CertiCUAC lo dice).**

*** Enigform es una extension a HTTP, por lo que no tiene problemas para correr CON SSL!**

HTTP encriptado con OpenPGP

* Se puede encriptar todo un request, encapsularlo en un request “esqueleto”.

* Mediante un Input Filter de apache, se desencripta el request y se procesa el verdadero request.

* La respuesta debería encapsularse, encriptada, en una Respuesta HTTP.

* No existen Input Filters para Firefox, o IE: se complica.

* Se puede usar un sistema de proxy para encriptar y desencriptar.

* Al menos existe SSL!

Possible “Encrypted Request”

```
POST /HTTP_OPENPGP_DECRYPT HTTP/1.0
Host: localhost
Connection: Close
Content-Length: 370

-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.7 (GNU/Linux)
hQIOA9YKl/p/3dcgEAf/erCrgwG8kB3
26nA4WU2ZmB3i5ZP4aaZwKZulsBh
hfJ0mgGFIGwl+uFyQoCwXk33H5j1lJ
lqKlLHtccLuG4fAAYfviqLmuK8vpFcb
oYYkiLgJ+fDDxDwGAZ/6ryAN3tlPUyq
TDPOAzpQM0ho385J4xv1ZfQRajSRY
a/YBe4SGJyjyvDXxpUMhbftZMKDML
3E08dzok+uvoLVDBJ7wpFhYACcguX
N1cR2yQFqiuR+S6ycEo/qEL2XNM3r
=OPnL
-----END PGP MESSAGE-----
```

Ejemplo de Uso – Home Banking

*** Un cliente del Banco Feliz instala el plugin Enigform.**

*** Crea (excepto que ya tenga) su par de llaves OpenPGP.**

*** Su llave pública es enviada a una red SKS, o al keyserver del banco.**

*** Con su tarjeta del banco, visita un cajero automático y vincula su cuenta con su llave pública. Se emite un ticket que es presentado en caja (por ejemplo).**

*** Ahora puede ingresar en forma segura, incluso sobre SSL, al Home Banking. Otros métodos de ingreso pueden tener menos privilegios.**

Gracias por su tiempo.

¡Adiós!

Arturo 'Buanzo' Busleiman
buanzo@buanzo.com.ar
www.buanzo.com.ar
wiki.buanzo.org