Caoss Site Painting

Por The X-C3LL

Visita Http://0verl0ad.blogspot.com

./0x00 Introducción al tema

El tema que venimos a tratar en este pequeño paper es la posibilidad desde el exterior (véase desde una web, por ejemplo) realizar un "ataque" (entrecomillémoslo) hacia las impresoras que tenemos en nuestra intranet. No es un tema transcendental ni mucho menos, pero para mí es algo que me resulta bastante curioso y que está poco documentado en castellano.

Realmente el impacto no lo considero grave ni mucho menos, bueno quizás en el caso de mandar FAXs el impacto puede ser económico… y alguna denegación de servicio también se puede lograr… pero salvo estos casos, en general sólo sirve para imprimir documentos en redes ajenas (un poquito de SPAM).

El Cross Site Printing no es más que una variante del Cross Site Scripting (XSS) de toda la vida, consistente en que al visitar una web, ésta contiene un código JavaScript que enviará una petición a la impresora donde mediante código PostScript podremos llegar hasta a "dibujar". Auqnue no sólo con JavaScript, ya que podemos variar el ataque y convertirlo en uno usando las CSS (CSS Attack), cosa que ya veremos en su apartado.

Sobre la vulnerabilidad, decir que surge por una falta de previsión por parte de los fabricantes, los cuales no han pensado la posibilidad de colocar autentificaciones u algún otro sistema de identifiación, o de filtro de seguridad para evitar esto. He de decir, que en los últimos años según he estado leyendo recientemente, esto ya no es así, sino que algunos sistemas ya incorporan medidas de seguridad básicas… pero como casi todo hoy en día, tienen fallos aprovechables (podeis hechar un ojo poralgún bugtraq para ver algunas vulnes en impresoras).

./0x01 Conceptos básicos

Antes de poder meternos a fondo con el tema, necesitamos conocer ciertos conceptos básicos para poder entender más adelante todo lo que vamos a decir. Para empezar… ¿de qué forma saben las impresoras cómo es la descripción de la página que han de escribir?

Para responder a esto, debemos de saber qué son los lenguajes de descripción de páginas. Los PDL (Page Description Language) son los encargados de describir la maquetación del texto, la disposición de elementos gráficos como objetos vectoriales o mapas de bits, el formato de las páginas, etc… Podemos encontrar diferentes PDLs dependiendo de la empresa fabricante de la impresora, así encontramos por ejemplo que HP usa PCL (Printer Command Language, es muy común confundir estas siglas con las de Printer Control Languages, siendo este último un sinónimo de PDL), Epson usa ESC/P (Epson Standar Code for Printers) y ESC/P2 (versión mejorada del anterior), Samsung tiene el SPL (Samsung Printer Language), y por último nombrar a Adobe y su PostScript. Estos son los más importantes, puesto que hay muchos más.

De los PDL que hemos mencionado, nosotros nos vamos a central sobre todo en PCL y PostScript, puesto que son los que mayor transcendencia han tenido. De ambos lenguajes hablaremos más adelante con algo más de detalle, por ahora sólo hacer un par de apuntes interesantes. Sobre PostScript, decir que fue de los primeros en ser un lenguaje de programación en sí, es decir, marca la diferencia con los demás sistemas de descripción de la época, los cuales se basaban en secuencias de caracteres de escape. También decir que implementó la composición de imágenes a través de líneas horizontales y curvas, nubes de píxeles y distintos tipos de tipografía. El formato PDF (Portable Document Format) es un derivado de PostScript.

Para terminar de hablar de los PDLs, hablaremos de PCL. El PCL introducido por HP ha sido uno implementado por un gran número de fabricantes, por ello va ha ser sobre el que trabajemos. Los comandos de PCL son secuencias iniciadas por algún carácter de escape.

Ahora toca hablar sobre los protocolos que usan las impresoras en red. Bien, vamos a hablar a grandes rasgos (puesto que en materia de protocolos sólo sé aquello que me ha ido haciendo falta) de los protocolos que suelen usar las impresoras. El primero es el LPD (Line Printer Daemon) y trabaja por el puerto 515.

Después tenemos el IPP (Internet Printing Protocol) que está basado en el protocolo IP ,y construido sobre HTTP, permite la autentificación y el cifrado. Este protocolo envía una mayor cantidad de datos relacionados con el trabajo que otros protocolos. Utiliza el puerto 631.

El protocolo de recursos compartidos CIFS (Common Internet File System), anteriormente conocido como SMB (Server Message Block), también puede ser utilizado para la comunicación con impresoras dentro de una red. Este protocolo es empleado principalmente por Microsoft, aunque exite una implementación libre del protocolo llamada Samba que permite su utilización en sistemas operativos GNU/Linux. Los puertos que puede emplear son 137, 138 y 139.

Y por último, sobre el que vamos a hablar en el paper, es el protocolo Raw TCP/IP, el cual trabaja por el puerto 9100.

./0x02 Buscando Impresoras

Como ya hemos dicho, nosotros nos vamos a centrar en los ataques a través del protocolo Raw. Para realizar una búsqueda de impresoras dentro de nuestra intranet, o bien queremos hacerla hacia redes remotas (si estan mal configurada podemos acceder a las impresoras) tenemos varias opciones. La opción A es codearnos un programa sencillo en nuestro lenguaje preferido (en mi caso PERL) y lanzar sockets a direcciones IPs internas o hacer una búsqueda masiva. En cualquiera de los casos lo que debemos de hacer es tratar de ver si tiene el puerto 9100 abierto. En caso afirmativo hemos encontrado una impresora sobre la que trabajar.

Pero en este caso no estaríamos explotando ninguna vulnerabilidad ni nada similar… y sería aburrido. Recientemente he estado leyendo acerca de los Scanners de puertos en JavaScript y en CSS. Como ya dije al principio, el Cross Site Printing es una vulnerabilidad derivada de los XSS's. Luego si un usuario visita una web vulnerable a XSS, y en el código malicioso incluimos el PortScanner, podremos saber la direcciones de las impresoras que hay en su red, y ahí atacar. Este pequeño concepto lo he sacado a partir de una idea que teníamos en mente S e t h y yo sobre analisis de intranets desde el exterior para combinarlo con Malwares contra routers… pero el concepto es portable a otros escenarios como es este.

Aquí os dejo un ejemplo de un PortScanner en JavaScript:

```
var AttackAPI = {
        version: '0.1',
        author: 'Petko Petkov (architect)',
        homepage: 'http://www.gnucitizen.org'};

AttackAPI.PortScanner = {};
AttackAPI.PortScanner.scanPort = function (callback, target, port, timeout)
{
        var timeout = (timeout == null)?100:timeout;
        var img = new Image();

        img.onerror = function () {
                if (!img) return;
                img = undefined;
                callback(target, port, 'open');
        };

        img.onload = img.onerror;
        img.src = 'http://' + target + ':' + port;

        setTimeout(function () {
                if (!img) return;
                img = undefined;
                callback(target, port, 'closed');
        }, timeout);
};
AttackAPI.PortScanner.scanTarget = function (callback, target, ports,
timeout)
{
        for (index = 0; index < ports.length; index++)
                AttackAPI.PortScanner.scanPort(callback, target,
ports[index], timeout);
};
```

Mis conocimientos en JavaScript rozan casi la nulidad, por ello no he podido modificar este script para que busque impresoras en nuestra red, pero sí que tengo la idea de cómo se haría (me faltan los conocimientos sobre el lenguaje para poder expresarlas). La cosa sería crear unos bucles FOR que fueran creando diferentes IPs posibles incrementandolas (las que equivaldrían al Target) para que den lugar a las diferentes IPs de nuestra red (192.168.1.1, 192.168.1.2 y así sucesivamente) y aplicarle la función de escaneo a esas IPs, pero haciendo que únicamente revise el puerto 9100.

Si alguien logra crear un scanner basandose en estas indicaciones, por favor envíadmelo para incluirlo en esta publicación.

./0x03 El arte del Ascii

Ya sabemos cómo encontrar impresoras en nuestra red y en ajenas, y además a través de que puerto vamos a trabajar. Ahora toca hablar sobre el "trabajo en sí".

Para comunicarnos con la impresora, y que ésta imprima debemos de hacer una conexión a ésta a través del puerto 9100. Esta conexión podemos establecerla de multiples maneras, como por ejemplo a través de NetCat, Telnet, Putty o algún otro sistema que permita negociaciones de este tipo. Pero, como nosotros queremos enfocarlo hacia el ataque desde una web, usarermos para la conexión nuestro navegador. Para ello, entremos a la URI http://IPdeTuImpresora:9100. Al añadir los : y el número, indicamos a nuestro navegador que se conecte a través del puerto que queramos.

Si todo ha salido bien, no debería de cargarte ninguna web, es decir, verías todo en blanco… y en la impresora algo recién impreso: la cabecera HTTP que enviástes. Ahora observar qué ocurre si mandais una petición tipo POST con por ejemplo la siguiente frase: "Lutscher si estás leyendo esto… vuelve con nosotros!!!!" usando Live HTTP Headers. La frase aparece impresa sí, pero se encuentra bajo URLEncoded, lo que podría no ser un texto legible con facilidad.

Bien, volvamos al ejemplo de que tenemos cierta web vulnerable a XSS y combinando el scanner en JavaScript y añadiendo un formulario hidden que contenga nuestra frase, y haciendo que este se autosubmitee para que se mande a nuestra impresora (que sabremos por el scanner), podríamos hacer un poco de SPAM. Pero si

realmente queremos mandar un mensaje o un dibujo en ASCII, el URLencode resulta engorroso. Esto podemos solventarlo si el formulario lo dejamos de esta forma:

```
<FORM ACTION='HTTP://YOURPRINTER:9100' NAME="IMPRESORA"
ENCTYPE='MULTIPART/FORM-DATA' METHOD='POST'>

 <input type="hidden" value="Estas siendo Spammeado :O">

</FORM>
```

A esto le añadimos la función de auto envío y ya. Aquí juega un papel importante podríamos decir el dibujar en ASCII para poder dar cierta mejor calidad a nuestro texto (añadir dibujos y esas cosas).

./0x04 PostScript: Controlando la Impresión

Al inicio del paper ya estuvimos hablando sobre los lenguajes PDL y dijimos que nosotros ibamos a usar PCL y PostScript para controlar diversas funciones. En este apartado, vamos a conocer nociones muy básicas de Post Script (para aprender más buscad algún manual) y cómo encapsularlo dentro de PCL.

Empecemos viendo como son los comandos en PS (a partir de ahora lo usaremos como abreviatura). A diferencia de los lenguajes que todos conocemos, en PS los parámetros de los comandos han de introducirse antes que el comando, por ejemplo:

100 300 moveto

En este caso, moveto es el comando y sus parámetros son 100 y 300. Este comando sirve para mover el cursor hasta el punto de inicio del dibujo. Los dos valores que se indican antes hacen referencia a las coordenadas (utilizando como referencia píxeles, cosa que podemos modificar con otros comandos al igual que el tamaño del papel) x e y.

Como todos sabemos desde la primaria, para dibujar una línea recta necesitamos dos puntos, y si ya tenemos uno ahora nos hace falta el punto final. Para ello usaremos lineto, cuya sintaxis es la misma que la de moveto:

300 360 lineto

Donde el primer valor indica las coordenadas en el eje X y el segundo en el eje Y. Pero no solo necesitamos esto antes de hacer una línea, sino que debemos de indicar el inicio de que se va a dibujar una línea con el comando newpath y el final con stroke. Con esta información ya podemos empezar a dibujar figuras rectilíneas simples:

```
newpath
150 150 moveto
350 150 lineto
350 300 lineto
150 300 lineto
150 150 lineto
stroke
```

Aquí ya tendríamos un rectángulo. Exsiten múltiples programas que sirven para representar en pantalla el dibujo, de esta forma podremos trabajar mejor.

Para indicar el ancho de línea con la que queremos que se realice nuestro dibujo, emplearemos el comando setlinewidth:

X setlinewidth

Siendo X el grosor de la línea indicado en píxeles.

También podemos controlar la tipología con la que escribimos. Por ejemplo, para seleccionar un determinado tipo de fuente no tenemos nada más que escribir /nombredelafuente findfont . Despues seleccionamos el tamaño con scalefont (x scalefont), setear la fuente para usarla con setfont y por último escribirla y mandarla con show. En resumen:

/Times-Roman findfont
50 scalefont
Setfont
100 300 moveto
(Overload In The Net RoolZ!!) show

Por ultimo, para realizar la impresión sólo tendremos que finalizar con showpage.

Son muchos los comandos y las posiblidades (desde trazar curvas, unir puntos con curvas, usar colores, mapas de bits, etc…) y no es el objetivo de este documento centrarse en esto. Simplemten pongo algunos casos sencillos para que entendais más o menos lo básico de este lenguaje. Existen programas que convierten webs enteras a PS, lo cual nos facilitaría enormemente el trabajo.

Para encapsular un código PS dentro de PCL para que sea ejecutado por la mayoría de las impresoras, simplemente tenemos que inicialmente indicar qué lenguaje vamos a usar usando para ello la siguiente directiva:

%-12345X@PJL ENTER LANGUAGE = POSTSCRIPT

Y para indicar hasta donde termina el código, usaremos *%-12345X*. Y como último detalle decir que al inicio del código PS tenemos que poner antes del primer comando %!PS.

Si todo esto lo unimos a JavaScript, tenemos la forma perfecta de controlar la impresión. Aquí os dejo un pequeño ejemplo que he encontrado al consultar un pequeño paper de Aaron Weaver

```
var msg=String.fromCharCode(27)
 + "%-12345X@PJL ENTER LANGUAGE = POSTSCRIPT\r\n”
+ "%!PS\r\n"
 + "/Courier findfont\r\n"
+ "20 scalefont\r\n"
 + "setfont\r\n"
+ "72 500 moveto\r\n"
+ "(Your printer is mine!) show\r\n"
+ "showpage\r\n"
+ String.fromCharCode(27)
+ "%-12345X”
```

./0x05 Conclusión

Para concluir simplemente decir que proteger las impresoras con passwords u otros sitemas es algo bastante importante, ya que pese que aquí hemos visto únicamente como imprimir en casa a ajena usando para ello una web vulnerable a XSS y un poquito de JavaScript, el impacto puede llegar a ser bastante mayor. Por ejemplo, se podría llegar a causar denegaciones de servicio en la impresora, y en toda la red.

A parte de esto, hay que tener mucho cuidado a la hora de las configuraciones, para evitar que personas ajenas a la red puedan ejecutar acciones sobre ella, tal y como hemos visto.

Para contacto: Camaleon__81 -at- Hotmail -dot- com

http://0verl0ad.blogspot.com