

## **PRET, reventando impresoras al vuelo**

Publicado por Alejandro Taibo on jueves, 2 de febrero de 2017 Etiquetas: [hardware](#), [herramientas](#), [Internet de las cosas](#), [python](#)



Todos nosotros alguna vez hemos manipulado alguna impresora, ya sea para hacer copias, imprimir documentos, escanear... También, muchos tenemos una en casa de las que desde hace unos años se comenzaron a desarrollar sin la necesidad de ser conectadas físicamente a nuestro ordenador. Poco después, se introdujo en ellas la funcionalidad de la conexión vía WiFi con la que podemos imprimir documentos inalámbricamente, con la única condición de que tanto la impresora como el ordenador desde el que queremos enviar el documento, estén conectados a la misma red.

Con estos avances a algunas personas (*trolls*) se les encendió la bombilla y, utilizando el sentido común, pensaron que estando en la misma red que una impresora, podrían imprimir cualquier documento sin el permiso del propietario para, como no, trollear y reírse un rato.

Con el paso del tiempo, para evitar lo mencionado anteriormente, se han ido implementando algunas medidas de seguridad en algunos dispositivos, como por ejemplo contraseñas que se usan para vincular dispositivos para que solo ellos puedan realizar acciones sobre dicha impresora.

Pero como todos los que leemos este blog sabemos, la seguridad es algo "ilusorio".

Y sí, esta vez le ha tocado la china a las impresoras. Hace una semana, unos estudiantes de la universidad de *Ruhr University Bochum* [presentaron su tesis](#) en la cual, desarrollaron una herramienta que han liberado hace pocos días en Github, hablo de [PRET](#), una herramienta desarrollada para probar las seguridad de las impresoras.

Esta herramienta se puede conectar a la impresora vía USB o vía WiFi y se encarga de explotar el lenguaje con el que fue desarrollada la impresora objetivo. Actualmente ataca a los lenguajes [PostScript](#), [PJM](#) y [PCL](#), los cuales se encuentran en la mayoría de impresoras láser.

Esta herramienta nos permitirá hacer cosas como: capturar documentos, modificarlos, acceder a la memoria de la impresora o incluso dañar la impresora físicamente. Esta tool ha sido programada en Python, que será un requisito imprescindible para su uso e instalación, más concretamente, deberemos

tener instalado el interprete Python2.

Cómo instalar esta herramienta:

1. Deberemos descargar la herramienta de Github: <https://github.com/RUB-NDS/PRET>.
2. En la terminal ejecutaremos: `pip install colorama pynmp`.
3. Posteriormente ejecutamos: `pip install win_unicode_console`.
4. Y por último: `apt-get install imagemagick ghostscript`.
5. Para ejecutarlo únicamente tendremos que escribir `./pret.py` y nos mostrará los siguientes parámetros.

```
pret.py [-h] [-s] [-q] [-d] [-i file] [-o file] target {ps,pjl,pcl}
```

positional arguments:

target	printer device or hostname
{ps,pjl,pcl}	printing language to abuse

optional arguments:

-h, --help	show this help message and exit
-s, --safe	verify if language is supported
-q, --quiet	suppress warnings and chit-chat
-d, --debug	enter debug mode (show traffic)
-i file, --load file	load and run commands from file
-o file, --log file	log raw data sent to the target

Los creadores de esta herramienta también nos muestran los comandos que podremos ejecutar en base al lenguaje de la impresora que estemos atacando. Serían los siguientes:

Command	PS	PJL	PCL	Description
ls	✓	✓	✓	List contents of remote directory.
get	✓	✓	✓	Receive file: get <file>
put	✓	✓	✓	Send file: put <local file>
append	✓	✓		Append to file: append <file> <str>
delete	✓	✓	✓	Delete remote file: delete <file>
rename	✓			Rename remote file: rename <old> <new>
find	✓	✓		Recursively list directory contents.
mirror	✓	✓		Mirror remote filesystem to local dir.
cat	✓	✓	✓	Output remote file to stdout.
edit	✓	✓	✓	Edit remote files with vim.
touch	✓	✓		Update file timestamps: touch <file>
mkdir	✓	✓		Create remote directory: mkdir <path>
cd	✓	✓		Change remote working directory.
pwd	✓	✓		Show working directory on device.
chvol	✓	✓		Change remote volume: chvol <volume>
traversal	✓	✓		Set path traversal: traversal <path>
format	✓	✓		Initialize printer's file system.
fuzz	✓	✓		File system fuzzing: fuzz <category>

path	-	Explore fs structure with path traversal strategies.	
write	-	First put/append file, then check for its existence.	
blind	-	Read-only tests for existing files like /etc/passwd.	
df	✓	✓	Show volume information.
free	✓	✓	Show available memory.

Y con ésto, una vez más queda demostrada la inseguridad de los IoT...

Por último comentaros también que esta herramienta ha sido creada para un proposito educativo, no ha sido diseñado para crear ningún tipo de daño... qué os veo!