Mr. PHP Analyzer

Reporte Técnico

****

Contenido

[1. Resumen 3](#_Toc500523669)

[2. Objetivo 3](#_Toc500523670)

[3. Desarrollo 3](#_Toc500523671)

[3.1. Estructura de la herramienta 3](#_Toc500523672)

[3.2. Instalación 5](#_Toc500523673)

[3.3. Ejecución 6](#_Toc500523674)

[3.3.1 Ejecución con todas las vulnerabilidades 7](#_Toc500523675)

[3.3.2 Ejecución especificando vulnerabilidad 7](#_Toc500523676)

[3.4. Resultados 7](#_Toc500523677)

## Resumen

Documento que muestra la forma de utilizar la herramienta Mr. PHP Analyzer.

## Objetivo

Dar a conocer la correcta utilización de la herramienta para que se pueda aprovechar de la mejor forma.

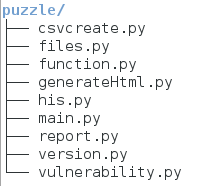
## Desarrollo

### Estructura de la herramienta



Img1. Contenido de la carpeta

* base
  + Carpeta donde se encuentra el script de la creación de la base de datos
* file
  + Carpeta que almacena un archivo “out” que contiene todos los archivos analizados con extensión php-
* install.sh
  + Script que hace la instalación de la herramienta.
* mrphpanalyzer
  + comando que se estará utilizando para ejecutar la herramienta una vez que se haya instalado.
* puzzle



Img2. Contenido de la carpeta puzzle

* + csvcreate.py
    - Script donde se hace la creación del reporte “csv” cada vez que la herramienta es utilizada.
  + files.py
    - Script donde se hace la lectura de cada uno de los archivos con extensión php.
  + function.py
    - Script donde se estará detectando las funciones obsoletas y se irán guardando en una lista para que sea utilizada por el archivo report.py. También se esta creando parte del reporte html para mostrarse con base en lo analizado.
  + generateHtml.py
    - Script que generará el archivo html, para la creación del reporte, esta recolectado toda la información de los archivos function.py y vulnerability.py.
  + his.py
    - Script que genera un archivo html que mostrara los archivos que se han generado con la herramienta.
  + main.py
    - Script que contiene la parte central de la herramienta, muestra el menú y ayuda para ejecutar la herramienta.
  + report.py
    - Script donde dependiendo si es función obsoleta o vulnerabilidades se hará el análisis para ir agregando a la base de datos los datos encontrados.
  + version.py
    - Script donde se detectará la versión del aplicativo al cual está analizando la herramienta.
  + vulnerability.py
    - Script donde se está detectando las vulnerabilidades y se irán guardando en una lista para que sea utilizada por el archivo report.py. También se esta creando parte del reporte html para mostrarse con base en lo analizado.
* README.md
  + Archivo donde se da una explicación general de lo que cuenta la herramienta.
* reportes
  + Carpeta donde se irán almacenando los reportes “cvs”.
* reportesHTML
  + Carpeta donde se irán almacenando los reportes “html”.

### Instalación

Se ejecuta el siguiente comando



Donde se creo una carpeta en /opt/



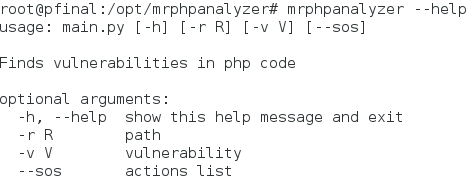
Img3. Muestra de la instalación de la herramienta

Contiene todos los archivos descritos en el punto 3.1.

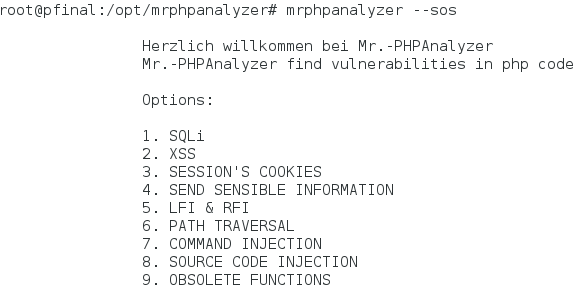
Img4. Archivos instalados

### Ejecución

El comando que estaremos utilizando es “**mrphpanalyszer**”



Img5. Mostrar ayuda



Img6. Vulnerabilidades disponibles

### Ejecución con las vulnerabilidades disponibles



Img7. Análisis con todas las vulnerabilidades a un directorio

### Ejecución especificando vulnerabilidad



Img7. Análisis especificando vulnerabilidad a un directorio

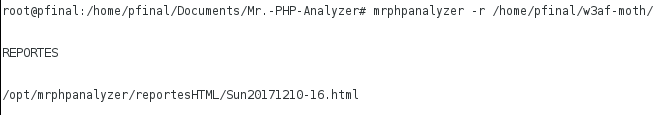
### Ejecución especificando archivo



Img6. Análisis de todas las vulnerabilidades a un solo archivo

### Resultados

Al término de cada análisis saldrá una leyenda indicando la ruta del reporte.



Que podrá ser revisando en un navegador.