fimap v.1.00\_svn (My life for Aiur)

:: Automatic LFI/RFI scanner and exploiter

:: by Iman Karim (fimap.dev@gmail.com)

Usage: ./fimap.py [options]

## Operating Modes:

-s , --single 模式扫描一个单一的网址连接错误。

需要网址（- U）。此模式为默认值。

-m , --mass 质谱扫描模式。将检查每一个网址 从一个给定的列表（- L）的网络连接错误。

-g , --google 模式使用谷歌获取URL。 需要查询（Q）的Google搜索查询。

-B , --bing 使用必应获得的网址。

需要一个查询（- Q）作为必应搜索查询。

还需要一个Bing APIKey（-- bingkey）

-H , --harvest 收获一个URL模式实现新的URL。

需要一个根网址（- U）开始爬行有。

还需要（- W）写一个质量模式的网址列表。

-4 , --autoawesome 与autoawesome模式fimap会把所有

在您定义的网站上发现的窗体和标题并试图找到文件包含错误通过他们。需要一个URL (-u).

##技术:

-b , --enable-blind 在没有打印错误消息时启用盲连接错误测试。请注意，这种模式将导致大量的请求相比，默认的方法。可以使用- S，-米或-克。

-D , --dot-truncation 使点截断技术摆脱后缀如果默认模式（nullbyte毒药）失败。这种模式可能会导致吨的要求，取决于你如何配置它。默认情况下，这种模式只EST的Windows服务器。可以使用- S，- M或- G的实验。

-M , --multiply-term=X 在路径上的乘法终端符号像“。”和“/”。

## 变量:

-u , --url=URL 您要测试的网址。

需要在单一模式（- S）。

-l , --list=LIST 你想测试url-list。

质量模式所需（- M）。

-q , --query=QUERY 谷歌搜索查询。

例如: 'inurl:include.php'

需要在谷歌模式（-克）

--bingkey=APIKEY 这是你的bing apikey。你必须把这个当你想使用BingScanner（B）。

--skip-pages=X 跳过第X页从googlescanner。

-p , --pages=COUNT 定义要搜索的页的计数（-克）。

默认为10。

--results=COUNT 结果Googlescanner应该得到每页的计数。 可能的值：10、25、50或100（默认）。

--googlesleep=TIME 在几秒钟内Googlescanner应该等待谷歌在每个请求的时间。fimap将计时两请求之间如果需要达到你的冷却时间的睡眠。默认为5。

-w , --write=LIST 列表中，将写如果你选择收获模式（H）。此文件将在附加模式打开。

-d , --depth=CRAWLDEPTH 的crawldepth（递归级别）你想爬到你的目标网站收获模式（H）。默认为1。

-P , --post=POSTDATA 你想送postData。所有的变量也将被扫描的文件包含错误。 --cookie=COOKIES 定义应发送的每个请求的饼干。还将扫描文件包含错误的文件。将多个饼干与“；”字符。

--ttl=SECONDS 定义TTL（秒）的请求。默认为30秒。

--no-auto-detect 使用此开关，如果你不想让fimap自动检测模式盲目标语言。在这种情况下，你会得到一些选项，你可以选择不相信，如果fimap郎是。

--bmin=BLIND\_MIN 在这里定义的目录fimap最低数应该走在盲模式。默认值是在generic.xml定义

--bmax=BLIND\_MAX 在这里定义的目录fimap最大计数应穿过。

--dot-trunc-min=700 点开始在点截断模式的计数。

--dot-trunc-max=2000 点在点截断模式下的计数。

--dot-trunc-step=50 点截断模式下的每一个圆的步长。

--dot-trunc-ratio=0.095 最大比率检测点截断成功。

--dot-trunc-also-unix 如果点截断也应在Unix服务器上测试使用。

--force-os=OS 军队fimap测试操作系统文件。

操作系统可以是“的”或“窗口”

## 攻击工具包:

-x , --exploit 开始一个互动的会话，你可以

选择一个目标并做一些动作。

-X 同X也说明不可利用这可能可以hax0red插件。

-T , --tab-complete 使标签在开发模式的实现。需要readline模块。如果你希望能够通过远程filesdirs使用这个标签完整。对每一个“光盘”命令都需要额外的请求。

--x-host=HOSTNAME 主机上使用的漏洞。fimap不会提示您在开发模式的域，如果你设置这个值。

--x-vuln=VULNNUMBER 您要使用的漏洞。它是相同的数字类型到开发模式，在那里你选择了脆弱的脚本。

--x-cmd=CMD 你想在易受攻击的系统上执行命令。使用这个参数不止一次来执行命令一个接一个。记住，每个命令打开一个新的壳和关闭后执行n.## Disguise Kit:

-A , --user-agent=UA 应发送的用户代理。

--http-proxy=PROXY 使用此选项设置您的代理。

But read this facts:

\* 的googlescanner将忽略代理获得的URL，但pentestattack本身会通过代理。

\* 代理应该是这样的：127.0.0.1:8080格式

\*它的实验

--show-my-ip 显示您的互联网IP、当前国家和用户代理。

如果你想测试你的vpnproxy配置。

## Plugins:

--plugins 列出所有加载的插件，然后退出。

-I , --install-plugins 显示了一些官方的开发模式的插件可以安装或升级。

## 其他:

--update-def 检查和更新在配置目录中发现的定义文件。

--test-rfi 快速测试，看看如果你已经配置好的RFI。

--merge-xml=XMLFILE 如果你有一个fimap xml文件要包括你自己的fimap\_result.xml使用这。

-C , --enable-color 使色彩丰富的输出。只在在！

--force-run 忽略实例检查和跑fimap即使锁定文件存在。警告：这可能会删除你的fimap\_results.xml文件！

-v , --verbose=LEVEL 你想收到的详细等级。

LEVEL=3 -> Debug

LEVEL=2 -> Info(Default)

LEVEL=1 -> Messages

LEVEL=0 -> High-Level

--credits 显示一些学分。

--greetings 一些问候

-h , --help 表明这一缺陷。

## 实例:

1. 扫描一个单一的网址连接错误：

./fimap.py -u 'http://localhost/test.php?file=bang&id=23'

2. 扫描一个网络连接错误的网址列表：

./fimap.py -m -l '/tmp/urllist.txt'

3.扫描谷歌搜索结果的网络连接错误：

./fimap.py -g -q 'inurl:include.php'

4. 收获一个3级网页所有链接递归写的网址 /tmp/urllist

./fimap.py -H -u 'http://localhost' -d 3 -w /tmp/urllist