

这是一个创建各种搜索工具的安装脚本。当然，这并不难，但是把他们整合起来是一件非常给力的事儿，因为这样在新的机器上部署这些东西就会变得很简单了。

## 合集包括了以下工具：

类型	工具	描述
binary	<a href="#">afl</a>	目前最棒的 fuzzer.
binary	<a href="#">angr</a>	来自Sheephish的下一代二进制分析引擎
binary	<a href="#">barf</a>	二进制分析逆向工程框架
binary	<a href="#">bindead</a>	binaries静态分析工具
binary	<a href="#">checksec</a>	检查binary hardening 设置
binary	<a href="#">codereason</a>	赛门铁克的 Binary 代码分析框架
binary	<a href="#">crosstool-ng</a>	跨编译、跨结构的工具
binary	<a href="#">elfkickers</a>	一系列ELF文件的实用工具
binary	<a href="#">elfparser</a>	通过静态分析快速确定ELF二进制容量的工具
binary	<a href="#">evilize</a>	创建MD5碰撞的binary工具
binary	<a href="#">gdb</a>	python2捆绑的最新的gdb
binary	<a href="#">panda</a>	为Architecture-Neutral动态分析的平台
binary	<a href="#">pathgrind</a>	基于路径的fuzzer.
binary	<a href="#">peda</a>	gdb的增强环境
binary	<a href="#">preeny</a>	一个有用的preload集合
binary	<a href="#">pwntools</a>	有用的CTF实用工具
binary	<a href="#">python-pin</a>	捆绑python的pin
binary	<a href="#">qemu</a>	最新版本的qemu
binary	<a href="#">qira</a>	省时的debugger
binary	<a href="#">radare2</a>	像crowell的非常棒的工具
binary	<a href="#">rp++</a>	另一个gadget查询工具
binary	<a href="#">shellnoob</a>	Shellcode编写工具
binary	<a href="#">snowman</a>	反编译工具
binary	<a href="#">taintgrind</a>	一个valgrind taint分析工具
binary	<a href="#">villoc</a>	堆操作可见化操作工具
binary	<a href="#">virtualsocket</a>	一个很棒的和binaries进行交互的库
binary	<a href="#">xrop</a>	Gadget查询工具
forensics	<a href="#">binwalk</a>	固件分析工具
forensics	<a href="#">dislocker</a>	用于读取磁盘加密的工具
forensics	<a href="#">exetractor</a>	Python拆包工具 支持PyInstaller和py2exe.
forensics	<a href="#">firmware-mod-kit</a>	固件拆包/组包工具
forensics	<a href="#">pdf-parser</a>	PDF文件挖掘工具
forensics	<a href="#">scrdec</a>	Windows脚本解码工具
forensics	<a href="#">testdisk</a>	文件恢复的测试盘
crypto	<a href="#">cribdra</a>	交互式 crib dragging 工具(用于加密).

crypto [foresight](#) 预测随机数生成器的工具  
crypto [hashpump](#) 部署哈希长度扩展攻击的工具  
crypto [hashpump-partialhash](#) 支持只知道一部分哈希的哈希破解工具  
crypto [hash-identifier](#) 简单的哈希算法识别工具  
crypto [littleblackbox](#) 嵌入式设备的SSL/SSH私钥数据库  
crypto [msieve](#) 实行某种算法来影响大整数的C语言库  
crypto [pemcrack](#) SSL PEM 文件破解  
crypto [pkcrack](#) PkZip 加密破解  
crypto [python-paddingoracle](#) Padding oracle 攻击自动化  
crypto [reveng](#) CRC 查询  
crypto [ssh\\_decoder](#) 解密 ssh 通信。你需要ruby1.8  
crypto [sslsplit](#) SSL/TLS MITM.  
crypto [xortool](#) XOR 分析工具  
crypto [yafu](#) 自动整数因式分解  
web [burpsuite](#) Web 代理  
web [commix](#) 命令行注入利用工具  
web [dirs3arch](#) Web 路径扫描  
web [sqlmap](#) SQL注入  
web [subbrute](#) 能够枚举DNS记录和子域的工具  
stego [sound-visualizer](#) 可视化audio文件  
stego [steganabara](#) 另一种破解图像的工具  
stego [stegdetect](#) 图像破解工具  
stego [stegsolve](#) 图像破解工具  
android [apktool](#) Android APK分析、拆包、重新打包工具

## 使用方法

```
# set up the path
/path/to/ctf-tools/bin/manage-tools setup
source ~/.bashrc

# list the available tools
manage-tools list

# install gdb, allowing it to try to sudo install dependencies
manage-tools -s install gdb

# install pwntools, but don't let it sudo install dependencies
manage-tools install pwntools

# uninstall gdb
manage-tools uninstall gdb

# uninstall all tools
manage-tools uninstall all

# search for a tool
manage-tools search nreload
```

如果可能的话，请尽量保持工具是单独安装的（比如，安装到tool/directory）并且尽量使用git clean卸载工具（注意，你一定要小心，因为在你卸载工具时有可能导致其他工具也崩掉）。为了支持python独立，确保在安装和使用工具前创建了virtualenv（比如，mkvirtualenv --system-site-packages ctf。--system-site-package会让重新使用apt-get python包更加简单）。

## Docker

应广大用户要求，Dockerfile已经被列入其中。你可以build一个docker镜像：

```
git clone https://github.com/zardus/ctf-tools
docker build -t ctf-tools .
```

运行：

```
docker run -it ctf-tools
```

这个镜像里面包含了一些ctf工具并且可以运行，但是你可能还是需要安装这些工具。

## Vagrant

你可以构建一个Vagrant VM:

```
wget https://raw.githubusercontent.com/zardus/ctf-tools/master/Vagrantfile
vagrant up
```

连接：

```
vagrant ssh
```

## 添加工具

你可以通过以下活动添加工具

创建一个目录以要添加的工具名称命名

创建一个安装脚本

（可选择）如果需要特殊卸载步骤，你还可以创建一个卸载脚本

安装脚本会被执行。脚本会把工具安装到这个以工具命名的目录中。理想情况下，完全卸载应该使用git clean。

安装脚本应该创建一个bin目录，把可执行文件存在这里。这些可执行文件会被自动链接到主bin目录。从任何一个目录都可以启动他们。

## 许可

单独的工具具有自己的许可，你不用担心。

\* 参考来源：[GitHub](#)，FireFrank编译，转载请注明来自FreeBuf黑客与极客（FreeBuf.COM）