网络安全漏洞分析知识库系统说明文档

本系统由漏洞库、补丁库、规则库、扫描器四大部分构成，下文对系统使用说明及每一模块进行详细介绍。

**1.系统登陆**

系统登陆链接：<http://172.105.233.194/test/public/>，打开链接后，点击右上角Login按钮，输入用户名：jwl@qq.com、密码：123456，点击Login按钮即可登陆系统。登陆系统后，默认进入漏洞库管理页面。

**2.漏洞库**

**2.1漏洞库管理**

如图2-1所示，漏洞库管理分为：CVE漏洞库、CNVD漏洞库以及最新漏洞。其中CVE漏洞库包含了目前为止[CVE](http://cve.mitre.org)收录的所有漏洞信息。CNVD漏洞库包含了[国家信息安全漏洞共享平台](http://www.cnvd.org.cn)收录的所有漏洞信息。“最新漏洞”菜单列出了最近最新更新的20条CVE漏洞信息。

漏洞库均实现现了对漏洞信息的增加、删除、修改操作。用户可自行尝试，不做赘述。



图2-1 病毒库管理

**2.2漏洞库自动更新**

漏洞库后台通过Python脚本，实现将远程漏洞数据定期更新到本系统漏洞库的功能，始终保证本地漏洞与远程数据一致。具体代码脚本分别为：swarm\_cve.py、swarm\_cnvd.py。具体流程如下图2-3所示。



图2-2 漏洞自动更新流程图

**2.3漏洞库更新部分代码**



图2-3 漏洞远程更新部分代码

**3.补丁库**

**3.1补丁库管理**

如图3-1所示，补丁库中包含了本系统收录的漏洞对应的补丁信息。可以对补丁信息进行修改与删除操作。注意，删除补丁信息不会删除对应的漏洞信息。



图3-1 补丁信息

**3.2补丁自动更新**

补丁库后台通过Python脚本，实现将远程补丁数据定期更新到本系统补丁库的功能，始终保证本地补丁信息与远程数据一致。具体代码脚本分别为：patch\_to\_mysql.py。具体流程如下图3-2所示。部分代码如图3-3所示。



图3-2 补丁更新流程



图3-3 补丁自动更新部分代码

**4.规则库**

**4.1规则库管理**

如图4-1为系统规则库。规则库中的每一条信息是针对某个漏洞的扫描规则。举例说明，ID为3的规则是用来扫描主机是否存在CVE-2011-2523这个漏洞。可以对每一条规则进行添加、修改、删除的操作。

**4.2 规则的来源以及添加要求**

我们知道，不同的漏洞需要专门编写针对此漏洞的发现程序，并没有一个程序能够发现目前全部的漏洞。因此，针对每一个漏洞，需要专门的漏洞发现程序。

但是，到目前为止，发现的漏洞已数以万计，我们不可能为每一个漏洞都亲自编写一个探测程序。因此，我们引入了[nmap脚本](https://nmap.org/nsedoc/categories/exploit.html)。Nmap脚本库中包含了许多针对已收录漏洞的发现脚本。因此，我们只需将此脚本信息添加的系统的规则库中即可。添加的规则每项对应的nmap脚本信息如图4-2所示。

**4.3 使用nmap自定义漏扫程序**

当然，还可以自行编写漏洞发现程序，并添加到规则库中。这就要求需要了解[nmap脚本引擎](https://nmap.org/book/nse.html)。这部分可作为功能扩展部分，如有时间，可以进一步开发。



图4-1 系统规则库



图4-2 nmap对应规则库信息

**5.扫描器**

扫描器是根据规则库中的规则，针对目标对相应的漏洞发起扫描，分为发起扫描以及扫描记录两大部分。如图5-1示。



图5-1 扫描器

**5.1 发起扫描**

如图5-2所示，选择相应的规则，点击发起扫描后，需要填写待扫描的目标。目标一般为ip，有时可能是域名，有的漏洞扫描可能不需要指定目标，此项则不填。点击发起扫描后，在扫描记录页可以查看此条扫描任务的状态。

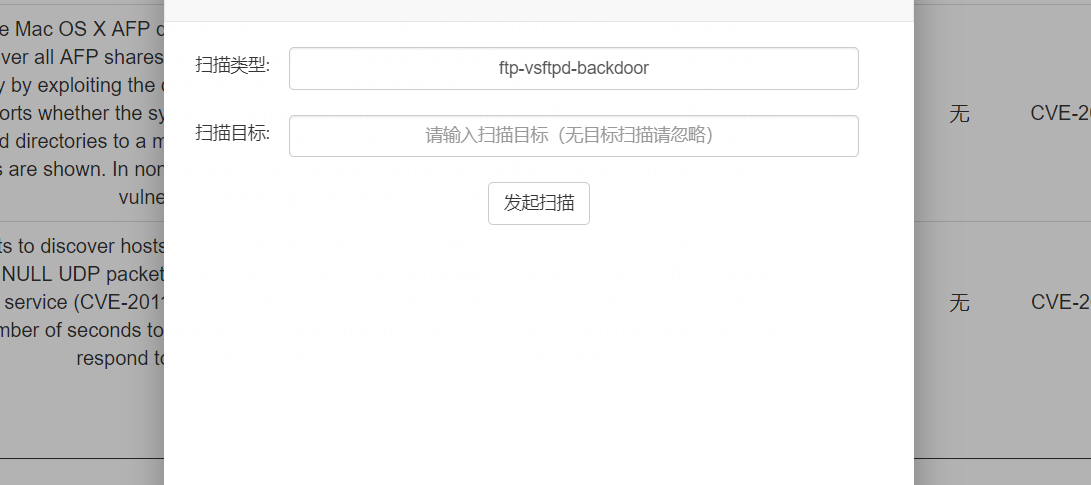


图5-2 发起扫描页面

**5.1 扫描记录**

如图5-3为扫描记录页面。这里可以查看刚刚发起的扫描任务信息，包括扫描任务的目标主机、扫描类型、扫描的漏洞、扫描状态、扫描输出结果、以及是否存在此漏洞。



图5-3 扫描记录页面

**5.3 扫描原理**

在5.1中我们介绍了规则库中的每条规则都是nmap漏洞扫描脚本的参数信息，因此，最终扫描的实现也是执行nmap命令完成的。

1. 执行扫描。对于每个发起的扫描任务，最终由scan.py去真正的执行。具体实现原理是scan.py脚本定时去读取任务列表，如果发现有新的未执行任务，则读取该任务的信息，主要是规则库中的各项参数信息以及扫描的目标ip，然后组合成一条nmap扫描命令执行扫描。
2. 漏洞判定。扫描结束后，scan.py读取扫描结果，对扫描结果信息进行关键字匹配，如果扫描结果中存在相应的关键字，则判定该目标存在该漏洞，反之亦然。对于关键字的选取，则根据每个具体的漏洞扫描脚本来具体对待。例如，“ftp-vsftpd-backdoor”这个扫描脚本的扫描结果信息中如果包含“vulnerable”这个关键字，则判断目标存在相应的漏洞。

具体流程如下图5-4所示。



图5-4 任务扫描流程

**6.论文可拓展方向**

编写自定义的漏洞扫描脚本，进一步扩充规则库。