**实 验 报 告**



**课程名称 《网络攻击与防御技术》**

**学 院 计算机科学技术学院**

**专 业 信息安全**

**姓 名 黄 力**

**学 号 15307130275**

**开 课 时 间 2018 至 2019 学年第 1 学期**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | 实验项目  名 称 | Snort安装 | 成绩 |  |   **一、实验目的**  （1）通过本实验初步了解入侵检测系统的工作原理  **二、实验内容**  （1）在虚拟机（CentOS）上安装和配置snort软件并以IDS（入侵检测）模式运行该软件，在其它主机上使用X-scan、nmap等扫描软件对运行了snort的虚拟机进行扫描，分析snort给出的报警信息是否与设置的规则相符  （2）分析实验成功或失败的原因  **三、实验环境**  （1）PC机操作系统：macOS Mojave 10.14  （2）虚拟机操作系统（Parallels Desktop 13.1.1）：64位CentOS、windows10  **四、实验原理**  snort有三种用法：嗅探模式、记录模式和网络入侵检测模式。本实验使用的是网络入侵检测模式，基本原理是对网卡收发的数据包进行分析，分析条目包括IP地址、协议、端口，匹配规则集中的报警规则的数据包snort将进行报警。  本次实验的原理很简单，只需按照实验要求中的步骤安装配置snort、编写规则集，安装X-scan等扫描工具，然后对目标主机扫描并分析snort的报警信息即可。大部分的安装和配置过程参考了网络上资料（见实验步骤），snort安装在CentOS虚拟机上，X-scan安装在windows10虚拟机上，snort规则的编写只挑选了较为简单和常见的几条。  **五、实验步骤及结果**  安装过程参考了参考资料1，其中有一些步骤遇到包或库的缺失的错误我通过google搜索解决，此处不赘述此过程。  （1）安装依赖  安装命令：1、安装flex、bison：yum install flex bison -y；  2、安装libpcap、libpcap-devel：yum install libpcap libpcap-devel -y；  3、安装libdnet：wget https://nchc.dl.sourceforge.net/project/libdnet/libdnet/libdnet-1.11/libdnet-1.11.tar.gz（获取压缩包）  tar -zxf libdnet-1.11.tar.gz（解压）  cd libdnet-1.11  ./configure && make && make install    （2）安装daq  安装命令：1、wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz  tar -zxf daq-2.0.6.tar.gz  cd daq-2.0.6  ./configure && make && make install  （3）安装snort  安装命令：1、wget https://www.snort.org/downloads/snort/snort-2.9.12.tar.gz  tar -zxf snort-2.9.12.tar.gz  cd snort-2.9.12  ./configure --enable-sourcefire && make && make install  完成（1）（2）（3）步骤后的截图如下：包括了对应的压缩包和解压后的目录  此时使用snort -V查看snort是否安装成功，结果如下图：可见安装成功  （4）安装snort规则  安装命令：1、创建snort配置及规则目录：mkdir -p /etc/snort/rules  2、创建运行目录：mkdir /usr/local/lib/snort\_dynamicrules  3、将（3）中解压出的etc下的默认配置文件拷贝到snort配置目录下：cp etc/\*.conf\* /etc/snort  ；cp etc/\*.map /etc/snort  4、下载社区规则并解压到规则目录：wget https://www.snort.org/downloads/community/com  munity-rules.tar.gz；tar -zxf community-rules.tar.gz -C /etc/snort/rules  5、注释掉所有默认加载的规则文件：sudo sed -i 's/include \$RULE\\_PATH/#include \$RULE\\_PATH/' /etc/snort/snort.conf  6、启用社区规则文件：echo '' >> /etc/snort/snort.conf  echo '# enable community rule' >> /etc/snort/snort.conf  echo 'include $RULE\_PATH/community-rules/community.rules' >> /etc/snort/snort.conf  7、重设snort.conf中的变量值：sed -i 's/var RULE\_PATH ..\/rules/var RULE\_PATH .\/rules/' /etc/snort/snort.conf  sed -i 's/var WHITE\_LIST\_PATH ..\/rules/var WHITE\_LIST\_PATH .\/rules/' /etc/snort/snort.conf  sed -i 's/var BLACK\_LIST\_PATH ..\/rules/var BLACK\_LIST\_PATH .\/rules/' /etc/snort/snort.conf  8、创建自己设置的规则文件，稍后我们自己写的规则就是写入到这个文件中：touch /etc/snort/rules/local.rules  9、测试配置文件的正确性：snort -T -c /etc/snort/snort.conf，测试结果如下图：可见成功  （5）编写规则：  根据snort的规则编写格式编写了针对虚拟机的ping、http、https数据包的报警以及mysql、ssh、telnet、ftp等常用服务的入侵报警规则，规则文件为/etc/snort/rules/local.rules，因为扫描的主机（windows10）与（CentOS）在同一局域网下，简便起见，规则中的源IP地址都设置为任意IP地址（any）；规则集如下图：  （6）以IDS模式运行snort：  为便于查看报警信息，先使用命令mkdir log创建目录log存放报警信息写入的报警文件alert；再使用命令：snort -d -l ./log -c /etc/snort/snort.conf运行snort，-d表示解析应用层数据包，-l表示写入报警信息文件到特定目录，-c表示使用的配置文件。运行截图如下：  (7)安装X-scan扫描软件：  在另一台虚拟机windows10上安装X-scan扫描软件，下载地址为：https://x-scan.apponic.com/  download/，下载后解压运行xscan-gui.exe即可，运行过程中遇到了缺失npptools.dll的报错，于是我在<http://filediag.com/>上下载了缺失的nptools.dll文件并将它放在与xscan-gui.exe同一路径下，再次运行xscan-gui.exe即可成功  (8)简单设置X-scan扫描选项  1、先在language选项中设置语言为简体中文，  2、在设置->扫描参数->检测范围->指定IP范围中填入snort运行的虚拟机ip地址（10.211.55.13），如下左图：  3、在设置->扫描参数->全局设置->扫描模块勾选要扫描的模块，我勾选的模块如下右图：包括ssh、ftp等之前设置过的snort规则选项。  (9)开始扫描：  点击开始按钮开始扫描：如下左图：此外我还在主机（MAC）上使用nmap进行了扫描，命令为nmap -v 10.211.55.13，如下右图：此外我还进行了ping测试。  (10)查看报警文件中的信息：  在CentOS上使用Contol-C结束snort的运行，并使用vi log/alert查看报警文件（该文件已经随本报告一同提交）；下图为节选的一段报警信息：从图中可以看出，snort针对80端口的http数据包、22端口的ssh数据包和ping数据包都按照我们之前编写的规则进行了报警。实验成功。  **六、实验总结**  通过本次实验，我初步了解了入侵检测软件的工作原理，也成功在CentOS上安装和配置了snort、编写了简单的入侵检测规则，并最终使用X-scan、nmap等扫描工具对snort进行了扫描，成功得到snort的报警信息。本次实验的主要难点在于snort、X-scan等软件的安装和配置，由于是初次使用这些软件，所以过程中遇到了不少的麻烦，但我最后都根据参考资料的帮助和google解决了这些报错问题。  **七、参考资料**  1、https://www.cnblogs.com/lsdb/p/8023884.html  2、https://www.jianshu.com/p/113345bbf2f7 |