基于层次聚类的警报处理方法

吴祎凡 崔艳鹏 胡建伟

(西安电子科技大学网络行为研究中心 西安 710071)

摘 要 针对入侵检测系统普遍存在冗余警报从而影响攻击类型判断的问题,文中提出了一种基于改进层次聚类的警报处理方法,其能减少冗余警报,提高攻击类型检测的准确性。该方法在层次聚类的基础上,使用警报的内容作为聚类的唯一属性值,增加了具有先验知识支撑的有效 Alert 占比来作为聚类阈值选取的标准,并改进了常规聚类直接抛弃高于阈值的类的处理方法,使用余弦相似度算法计算高于阈值的类的代表 Alert,有效避免了有用警报的丢弃。在通过合适的阈值聚类后,按照时间轴的顺序来展示时间窗口内去重且聚类后的警报结果,以便对攻击者的攻击类型进行快速判断。实验结果表明,改进后的聚类方法有较好的去冗效果。

关键词 Snort,层次聚类,警报,阈值选取,相似度计算

中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.04.032

Alert Processing Method Based on Hierarchical Clustering

WU Yi-fan CUI Yan-peng HU Jian-wei

(Network Behavior Research Center, Xidian University, Xi'an 710071, China)

Abstract Aiming at the problem that there generally exist redundant alarms in intrusion detection system and it affects the judgment of attack types, this paper processed an alert processing method based on improved hierarchical clustering, so as to reduce redundant alarms and improve the accuracy of attack type detection. On the basis of hierarchical clustering, this method uses the content of alarm as the unique attribute value of cluster, increases the percentage of effective alert with prior knowledge as the criteria for the selection of clustering thresholds, and improves the processing method of directly discarding the class whose value is higher than threshold in conventional clustering. The improved method uses the cosine similarity algorithm to calculate the representative alert above the threshold class, effectively avoiding discarding useful alarms. After clustering through suitable thresholds, the deduplicated and clustered alarm results within the time window are displayed in the order of the time axis to quickly determine the attacker's attack type. The experimental results show that the improved clustering method has better deduplicated effect.

Keywords Snort, Hierarchical clustering, Alert, Threshold selection, Similarity calculation

1 引言

随着网络技术的发展,网络安全问题日趋严重,网络攻击的防范也渐渐从被动防守转为主动防御[1]。溯源技术[2]就是一种新兴的对攻击者信息进行分析的主动防御技术,其关键点为对攻击者的攻击类型进行判断。而入侵检测(NIDS)[3]则是一种典型的攻击检测技术,只有快速、准确地检测攻击,才能更有效地进行主动防御。

Snort 是入侵检测系统中的一种,与大多数的入侵检测系统一样,都是通过监控目标区域的流量,收集并检查审计数据以确定入侵行为。当发现与规则库匹配的数据包时,就会尝试发出警报,使网络管理员快速做出反应。同样地,随着攻击技术的发展,检测规则也在不断地更新和丰富,各种各样的 NIDS 有可能因为规则冗余、配置错误、误报等产生大量的冗余报警,而这并不利于管理员针对攻击类型做出快速的判断。因此需要更加智能的手段来应对警报冗余问题,提高检测的准确性。

警报聚类可以从某种程度上减少警报的冗余数量。常规的警报聚类手段有:基于时间窗口的时间段内警报去重[4],将窗口内重复出现的警报计算为一条;基于概念聚类的方法推导过滤规则来减少误警,从而减少警报数量;基于警报序列的警报聚类,将新来的警报聚集到已聚好的类中,用聚类方法来计算多个属性间相似度[6];基于蚁群和 k-means 聚类算法的警报聚类[7],同样是针对多个属性间相似度的计算。上述聚

到稿日期:2018-03-21 返修日期:2018-06-15

吴祎凡(1994-),女,硕士,主要研究方向为网络安全,E-mail:543665029@qq.com;**崔艳鹏**(1978-),女,博士,副教授,主要研究方向为网络攻防、智能终端安全与防护,E-mail:ypcui@mail.xidian.cn(通信作者);**胡建伟**(1973-),男,博士,副教授,主要研究方向为网络安全与网络对抗。

类方法中,有的需要一些已有的攻击场景模型,有的需要多个属性来比对相似性,如 IP、时间、端口等,而且很大部分的聚类算法都是直接去除聚类中心之外的警报或者高于阈值的警报,这样可能会造成有用警报的抛弃。

本文提出了一种基于改进的层次聚类的警报聚类算法,其不需要过多的参数设置,也不需要计算多个属性的相似度,并在阈值筛选后,针对高于阈值的类,选取相似度度量值较高的警报作为类的代表警报,以防止有用警报的抛弃。对于聚类算法的评估,除了使用常规攻击类型占比之外,还使用了具有先验知识支撑的有效 Alert 占比。该方法通过使用改进后的层次聚类算法对入侵检测系统 Snort 的警报进行处理,能够很大程度地减少警报的数量,从而有效、准确地检测出攻击类型,效果较好。

2 相关原理

2.1 Snort 入侵检测系统

Snort 是一个开源、跨平台、轻量级的网络入侵检测系统,针对每一种入侵行为,提炼出其特征并照规范编写成规则,形成一个规则库,将捕获的数据包与规则库中的规则逐一匹配,若匹配成功,则认为该入侵行为成立。本文使用的是 Snort 系统的网络入侵检测模式,其体系的各个模块如图 1 所示。

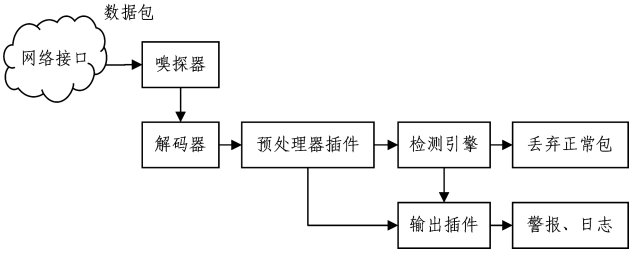


图 1 Snort 的体系结构

Fig. 1 Architecture of Snort

各个模块的功能分别为:嗅探器抓取待检测的数据包,再传给解码器和预处理插件进行数据预处理,以方便检测引擎使用模式匹配来对数据包和规则进行匹配,最后输出插件输出匹配到的警报、日志到需要的目录。

Snort 入侵检测系统除了基本的配置外,还需要其他组件的配合。Snort 默认的输出警报信息是二进制文件,可以通过 u2spewfoo 命令将其转换为文本格式,也可以通过 Barnyarnd2 插件转换并插入到 MySQL 数据库。Base 插件则可以通过读取 MySQL 的 Snort 数据库,展示简单统计后的警报信息[8]。Snort 入侵检测模式中组件之间的关系如图 2 所示。

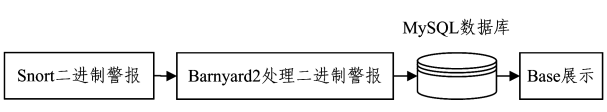


图 2 Snort 入侵检测模式

Fig. 2 Snort intrusion detection mode

2.2 层次聚类算法

层次聚类[9]是在不同层次上按一定的规则将数据聚集成若干个类,并使各个类内部的数据间具有较大的相似性,而各个类之间的数据具有较小的相似性。

层次聚类根据层次形成的顺序可以分为自下向上和自上向下,即凝聚的层次聚类算法和分裂的层次聚类算法。凝聚层次算法在初始时假设每个对象单独为一类,再通过迭代将这些类根据某种距离度量准则逐步聚合,直到所有对象最终合并成一个类。分裂层次聚类则与之相反,采取的是自顶向下的策略。

对比两种层次聚类的方法,由于警报聚类的目标是将多个相似性较高的警报聚类,从根本上减少警报的数量,因此对于警报聚类,通常使用的是凝聚的层次聚类算法。

理论上,该算法就是先计算样本之间的距离,并且每次将距离满足需求的点合并到同一个类;然后再计算类与类之间的距离,将距离满足需求的类合并为一个大类;不停地合并,直到合成了一个类。另外,即使到最后,噪音点或离群点往往还是各占一簇的。

凝聚层次聚类算法的过程描述如算法 1 所示。

算法 1 凝聚层次聚类算法

输入:样本集合 D,聚类数目或者某个条件(一般是样本距离的阈值[10],这样就可不设置聚类数目)

输出:聚类结果

1. 将样本集中的所有样本点都当作一个独立的类簇;
2. 计算两类簇之间的距离,找到距离需求的两个类簇 c_1 和 c_2;
3. 合并类簇 c_1 和 c_2 为一个类簇;
4. 重复步骤 2 和步骤 3,直到达到聚类的数目或者达到设定的条件。

计算两类簇之间的距离的方法也有很多,如最短距离法、最长距离法、平均值距离法等。本文使用的是最短距离法,即用两个待聚类中的所有数据点的最短距离代表两个聚类的距离[11]。

$$d_{\min}(c_i,c_j)=\min_{q\in c_i,q'\in c_j}|q-q'| \tag{1}$$

3 警报聚类

随着规则的不断更新,Snort 的准确率和误报率也在相应上升。由于警报的类型、数目等都无法预知,而层次聚类的距离、相似度等的计算比较容易定义,且限制条件较少,不需要预先设置聚类的数目,并且可以比较直观地看出层次关系,较其他聚类方法来说,层次聚类具有很高的灵活性,因此本文选用层次聚类算法进行警报去冗,并在此基础上进行了改进。

警报聚类是在获取待聚类警报的基础上使用层次聚类的方式对 Alert 样本进行层次聚类,在后期实验的基础上,进行白盒测试,选取合适的阈值,进而选取合适的聚类层级;接着使用聚类后的警报来过滤原始警报,按照时间轴对警报进行排列,并对时间窗口内的警报去重;最后输出按照时间轴排列的警报去冗、压缩数量后的结果。

Snort 的原始警报是一些二进制文件,可以通过 Barnyarnd2 插件将其转换为文本形式并存储在 MySQL 数据库。警报的获取则是通过联合查询 MySQL 数据库来得到。在获取待聚类的警报时,可以按照需求进行过滤,如选取特定来源 IP 和目的 IP 触发的警报、特定协议的数据包类型触发的警报、特定时间段内的警报等。

本文使用的基于层次聚类的警报处理方法的流程图如图

3 所示。其具体过程是：从 MySQL 数据库获取到需求的警报后，使用层次聚类算法生成完整的层次结构；使用阈值（阈值是经过大量实验选取的合适的聚类条件）筛选聚类层次，将层次内的叶子节点直接输出，类簇经过 Alert 相似度度量值的计算，选取代表的警报；最后输出所有警报的聚类结果。

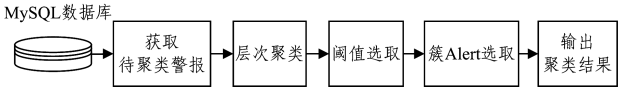


图 3　警报聚类的流程

Fig. 3　Flow of alarm clustering

3.1　警报层次聚类

层次聚类是在不同层次上按一定的规则将数据聚集成若干个类，并使各个类内部的数据间具有较大的相似性，而各个类之间的数据具有较小的相似性。基于层次聚类的警报聚类框图如图 4 所示。

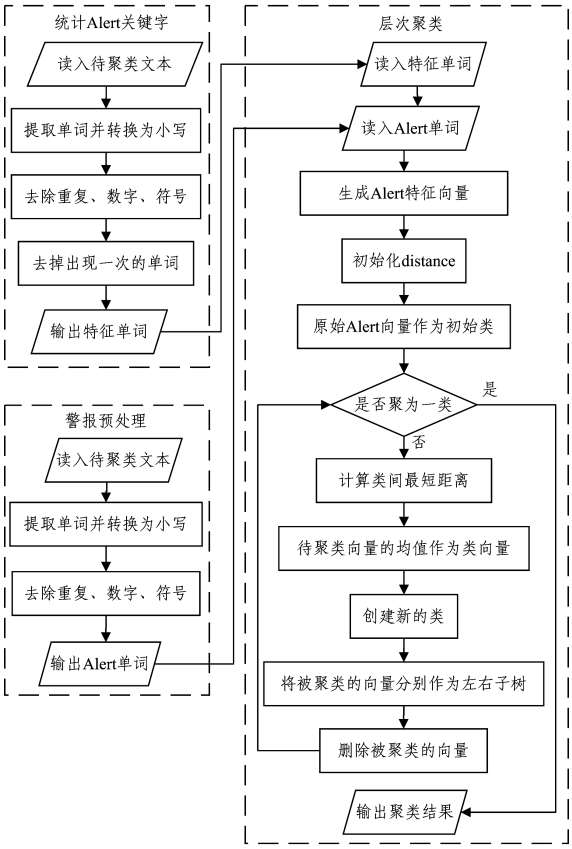


图 4　警报层次聚类流程图

Fig. 4　Flowchart of hierarchical clustering of alert

基于层次聚类算法的警报聚类流程概述为：

1）将所有待聚类的警报转换为特征向量；

2）将 N 个要聚类的报警数据分别单独归为一个类，即初始有 N 个类；

3）计算两两类间的距离，并找出距离 *distance* 最小的两个类；

4）将 *distance* 最小的两个类归并成一个新的聚类，同时总的聚类数减少 1；

5）新的聚类向量的值为 *distance* 最小的两个类向量的平均值，然后再将新的类作为父节点，*distance* 最小的两个类分别作为父节点的左子树和右子树；

6）重新计算新聚类与其他聚类间的距离；

7）重复步骤 3）和步骤 4），直到类的数目变为 1；

8）输出最后生成的树状结构。

预处理阶段主要分为特征单词的提取和特征向量的生成。层次聚类的核心部分为对距离的计算，在对比了常用的类聚类度量方法之后，选取最短距离法作为距离计算方法，即用两个待聚类中的所有数据点的最短距离代表两个聚类的距离。而初始 Alert 的 ID 标识都为正数，将新聚合类的 ID 标识为负数，并将待聚合的 Alert 作为新类的子树，最后在确定阈值之后，对选取到的层次包含的新类内部进行代表 Alert 的选取。

图 5 是对实验中 14 个 Alert 进行层次聚类后的树状图（后面的示例都是基于这 14 个 Alert），每个原始 Alert 都会有正值的 ID 标识，新聚合的类 cluster 使用负值 ID 和子树距离来标识。图中只展示了部分的层次聚类后的树结构，其余部分以此类推。

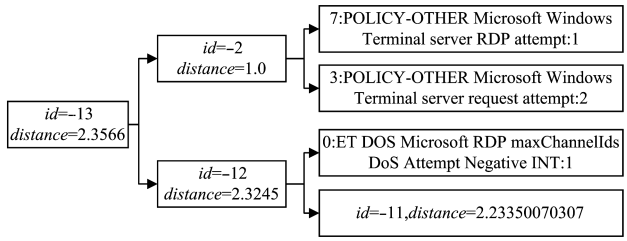


图 5　警报层次聚类结果

Fig. 5　Hierarchical clustering results of alert

3.2　有效 Alert 的选取

有效 Alert 是指可以代表攻击类型的 Alert 集合。以往的聚类使用攻击类型作为聚类阈值选取的唯一标识，但其并不准确，本文对大量典型攻击的有效 Alert 进行筛选，并将以此作为聚类阈值选取的重要依据，使得聚类结果更为准确。有效 Alert 的选取只需要在阈值选取阶段进行，筛选出合适的阈值之后，后面的聚类过程直接应用阈值即可。表 1 和表 2 是实验中待进行有效 Alert 选取的示例警报。

表 1 和表 2 是对几种典型的系统漏洞如 Ms12_020，Webdav_dll 和 IE 快捷方式图标漏洞的攻击流量进行检测而产生的警报结果。Snort 总共有 14 种（去重后）警报，警报冒号后面的数字代表的是该条 Alert 被触发的次数，因为 Snort 是依照每一个数据包进行数据检测的，所以每一个类似的攻击数据包就可能触发同一类 Alert。接下来，参考专家知识库对漏洞的解释，以及漏洞的具体利用过程，来筛选有效 Alert。

表 1　选取有效 Alert 的警报集合(1)

Table 1　Alert set to be selected by valid Alert(1)

Alert 信息
0. ET DOS Microsoft RDP maxChannelIds DoS Attempt Negative INT:1
1. INDICATOR-COMPROMISE Metasploit Meterpreter reverse HTTPS certificate:1
2. ET POLICY Suspicious Windows Executable Write Process Memory:4
3. FILE-EXECUTABLE download of executable content:4
4. FILE-OTHER Microsoft LNK shortcut arbitrary dll load attempt:1

表 2 选取有效 Alert 的警报集合(2)

Table 2 Alert set to be selected by valid Alert(2)

Alert 信息
5. ET WEB_CLIENT DLL or EXE File From Possible WebDAV Share Possible DLL Preloading Exploit Attempt:4
6. INDICATOR-SHELLCODE x86 inc ebx NOOP:1
7. POLICY-OTHER Microsoft Windows Terminal server RDP attempt:1
8. ET POLICY PE EXE or DLL Windows file download:2
9. FILE-EXECUTABLE Portable Executable binary file magic detected:2
10. ET INFO SUSPICIOUS Dotted Quad Host MZ Response:4
11. PROTOCOL-ICMP destination unreachable port unreachable packet detected:6
12. sensitive_data:sensitive data global threshold exceeded:1
13. POLICY-OTHER Microsoft Windows Terminal server request attempt:2

通过分析实验攻击的原理和过程,分别选取 *ID* 为 0,1,和 7,*ID* 为 4,*ID* 为 5 的 Alert 作为漏洞 Ms12_020,Webdav_dll 和 IE 快捷方式图标攻击的有效 Alert。

3.3 阈值选取

阈值选取的原则是聚类后的警报数目尽可能少,但包含测试所有的攻击类型和较多的有效 Alert。依据 3.2 节,使用类间的最短距离来衡量警报的相似度,因此选取合适的阈值与类距离来对比,从而选取合适的聚类层级。阈值选取的流程如图 6 所示。

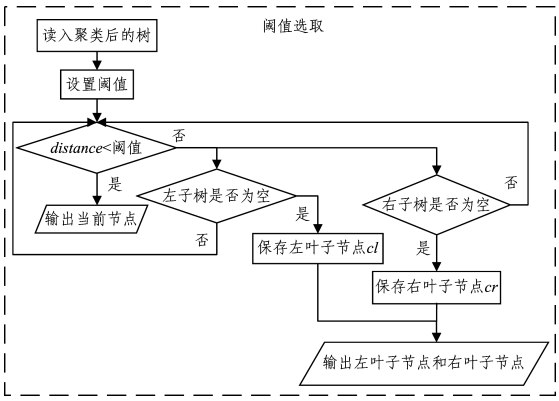


图 6 阈值选取的流程图

Fig. 6 Flowchart of threshold selection

阈值选取的原则是聚类后的警报包含测试所有的攻击类型、尽可能多的有效 Alert 和尽可能少的聚类后的警报数目。依据上一节选取的有效 Alert 的 *ID*,漏洞 Ms12_020 攻击有效 Alert 的 *ID* 为 0,1 和 7,Webdav_dll 攻击有效 Alert 的 *ID* 为 4,IE 快捷方式图标漏洞有效 Alert 的 *ID* 为 5,由阈值选取的原则可知,最合理的阈值选择是包含叶子节点 *ID* 为 0,4,5 的阈值 2.233。阈值为 2.233 时,聚类层次的警报有 7 条,其 *ID* 分别为-2,0,4,5,8,2,-7。

3.4 相似度计算

本文采用余弦相似度[12]度量向量相似度的方法来求取负值 ID 类下的所有叶子节点的相似度,并将每个叶子节点与其他叶子节点相似度的平均值作为该叶子节点相似度的度量值,最终选取度量值最大的叶子节点作为负值的类聚类后的代表 Alert。

余弦相似度是用向量空间中两个向量夹角的余弦值来衡量两个向量之间差异的大小。余弦值越接近 1,就表明夹角越接近 0°,也就是两个向量越相似。向量夹角的示意图如图 7 所示。

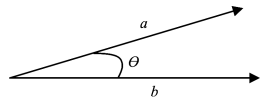


图 7 向量夹角

Fig. 7 Vectxior angle

向量空间余弦相似度理论就是基于上述原理来计算个体相似度的。

在直角坐标系中,假设向量 a 是(x_1,y_1),向量 b 是(x_2,y_2),则向量夹角的示意图如图 8 所示。

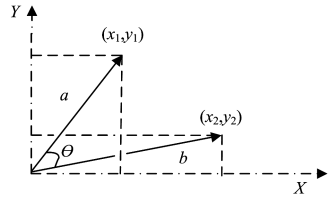


图 8 坐标系中的向量夹角

Fig. 8 Vector angle in coordinate system

那么向量 $\boldsymbol{a}$ 和向量 $\boldsymbol{b}$ 夹角的余弦计算如下:

$$\cos\theta=\frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|\times\|\boldsymbol{b}\|}=\frac{(x_1,y_1)\cdot(x_2,y_2)}{\sqrt{x_1^2+y_1^2}\times\sqrt{x_2^2+y_2^2}}$$

$$=\frac{x_1x_2+y_1y_2}{\sqrt{x_1^2+y^2}\times\sqrt{x_2^2+y_2^2}} \tag{2}$$

假设向量 $\boldsymbol{a}$ 和向量 $\boldsymbol{b}$ 是两个 n 维向量,向量 $\boldsymbol{a}$ 是$(x_1,x_2,\cdots,x_n)$,向量 $\boldsymbol{b}$ 是$(y_1,y_2,\cdots,y_n)$,则 $\boldsymbol{a}$ 与 $\boldsymbol{b}$ 夹角的余弦为:

$$\cos\theta=\frac{\sum_{i=1}^{n}(x_i\times y_i)}{\sqrt{\sum_{i=1}^{n}(x_i)^2}\times\sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{3}$$

余弦值越接近 1,表示夹角越接近 0°,也就是两个向量越相似,夹角等于 0,即两个向量相等,这就叫“余弦相似性”。

聚类后警报选取的具体流程如图 9 所示。

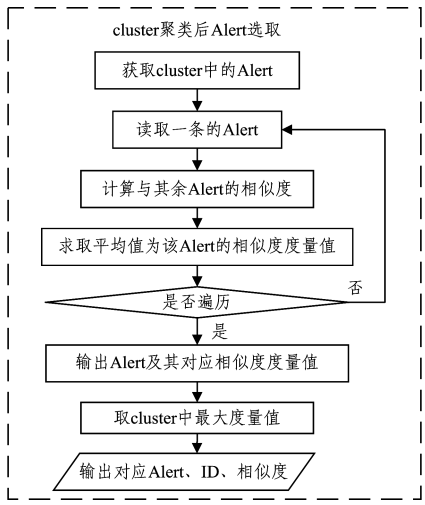


图 9 聚类后 Alert 的选取流程图

Fig. 9 Flowchart of Alert selection after clustering

依旧引用 3.3 节中阈值选取的层次聚类树的例子。阈值为 2.233 时,聚类层次的警报有 7 条,其 *ID* 分别为:-2,0,4,5,8,2,-7。提取 *ID* 为-7 的类的所有叶子节点,其 Alert 的 *ID*、内容和数量如表 3 所列。

表 3　计算相似度的警报集合和 *ID*

Table 3　Alert set and *ID* for similarity calculation

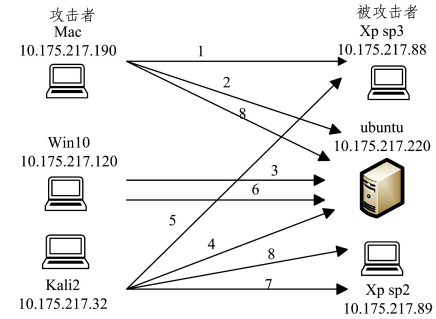
Alert 信息
1. INDICATOR-COMPROMISE Metasploit Meterpreter reverse HTTPS certificate:1
3. FILE-EXECUTABLE download of executable content:4
6. INDICATOR-SHELLCODE x86 inc ebx NOOP:1
9. FILE-EXECUTABLE Portable Executable binary file magic detected:2
10. ET INFO SUSPICIOUS Dotted Quad Host MZ Response:4
11. PROTOCOL-ICMP destination unreachable port unreachable packet detected:6
12. sensitive_data:sensitive data global threshold exceeded:1

接着，依据余弦相似度算法分别计算 *ID* 为 1 的 Alert 与其余 Alert 的相似度，并取相似度的平均值作为 *ID* 为 1 的 Alert 的相似度度量。其余 Alert 的相似度度量值的计算方法类似。通过计算得出相似度度量值最大的是 *ID* 为 6 的 Alert，其相似度度量值为 0.06835，因此选取 *ID* 为 6 的 Alert 的内容为 *ID* 为－7 的 Alert。

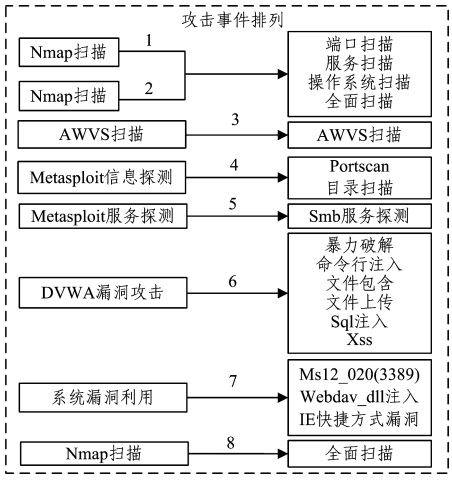
4　实验结果与分析

4.1　攻击检测

实验的攻击者有 3 台主机，操作系统分别为 Mac，Win10 和 kali2；被攻击者有 3 台主机，分别为 Ubuntu16.04，XP sp3 和 XP sp2，被攻击者选取不同的操作系统是为了还原 Web 层面的攻击以及系统层的攻击。在搭建网络拓扑的情况下还原了 14 种类型的攻击，包含信息探测、服务漏洞利用、暴力破解、SQL 注入等。具体的攻击过程如图 10 所示。



(a)



(b)

图 10　实验攻击过程

Fig. 10　Experimental attack process

图 10(a)代表攻击者和被攻击者系统及其 IP 信息，箭头指示的方向为攻击流的方向，线条上的数字代表的是事件发生的先后顺序，图 10(b)按照攻击发生的时间顺序详细列出了攻击的具体内容。

收集待回溯的原始数据包有 165956 个，Snort 检测到的攻击包数量有 40358 个，即 Alert 的个数，去重后剩下的 Alert 有 271 个，按照 Snort 的 class 划分的警报类型有 21 类。

4.2　层次聚类

从数据库提取警报，对其按照同源 IP 和同目 IP 进行划分，并依据已知的攻击类型选取有效 Alert。经过层次聚类算法处理后，绘制了 6 张聚类之后的树状图，并标识了每个节点的 *ID* 和 *distance*。

4.2.1　聚类阈值的选取

阈值选取的原则是聚类后的警报包含测试所有的攻击类型、尽可能多的有效 Alert 和尽可能少的聚类后的警报。在遵循此原则的前提下，从 6 张层次聚类后的树状图中总共获取到了 4 个阈值：2.233，1.903，1.732 和 1.414。

分别针对 6 组实验数据，按照阈值计算对应的警报聚类率、攻击类型的占比和有效 Alert 占比。原始警报有 284 个，有效 Alert 有 58 个，攻击类型有 14 种。

阈值选取标准的参数主要包含聚类率、攻击类型占比和有效 Alert 占比。

聚类率：聚类后的警报数量占原始待聚类的警报的比例。

攻击类型占比：聚类后警报包含的攻击类型占原始警报包含的攻击类型的比例。

有效 Alert 占比：聚类后警报包含的有效 Alert 数量占原始警报包含的有效 Alert 数量的比例。

使用不同的阈值合并聚类后的结果如表 4 所列。

表 4　层次聚类阈值选取后的聚类结果的对比

Table 4　Comparison of clustering results after hierarchical clustering threshold selection

（单位：%）

阈值	聚类率	攻击类型占比	有效 Alert 占比
2.233	18.6	76.9	31.0
1.903	33.0	100	51.7
1.732	39.4	100	55.1
1.413	53.1	100	77.5

阈值 1.903 和 1.732 都具有较高的警报聚类率，攻击类型占比在样本上也涵盖了所有的攻击类型。由于较低的阈值可以更好地包含有效 Alert，因此本文选取阈值 1.732 作为数据处理的阈值。

4.2.2　聚类结果展示

对实验的所有警报聚类后的结果进行展示，并没有 IP、时间、端口等属性的区分，因此提取到的警报数量比多个属性划分时的警报数量少。

从数据库获取到所有攻击不重复的 271 个 Alert，对其进行层次聚类，采用阈值 1.732 进行层级选取，并对选取到的簇进行相似度计算，选取相似度最高的 Alert。最终得到 130 条 Alert，其中包含了 28 条有效 Alert，实验测试的攻击类型完全包含在内。其警报聚类率达到 47.9%，有效 Alert 占比达到 48.2%，攻击类型占比为 100%。

为了动态还原攻击发生的时间和攻击详情以及攻击是否成功的状态，提取 Snort 数据库的所有 Alert，将对应 Alert 的数据包内容和 HTTP 状态码[13]文本进行匹配，若匹配成功，则输出对应状态码来标识攻击的状态，否则输出空格。输出格式为时间、Alert、状态码和数据包的 ID。接着使用聚类后的 Alert 对其进行筛选，留下与聚类结果相匹配的 Alert，并按照时间戳对其进行排列。

一般情况下，短时内的攻击行为会触发大量的同类型的警报。为了减少冗余的警报，便于筛选攻击类型，本文选取 2 min 为时间窗口(每隔 2 min 对时间内的警报数据进行处理)，对窗口内的警报进行去重[14]，并在展示时将最后一项作为该条 Alert 在 2 min 之内被触发的频率。

从数据库提取到的所有的 Alert 有 37 306 条，经过层次聚类的筛选后有 19 674 条，再通过时间窗口去重后得到 514 项。其部分结果如表 5 所列，表中首先展示了攻击发生的时间(精确到秒)；接着是警报的具体内容，即攻击的详情；如果攻击状态匹配到则显示攻击状态，没有匹配到则显示空格，其中攻击状态 200 OK 代表攻击行为已经成功与服务器建立连接，攻击危害较大；然后是数据包的 *ID*，便于管理员查看攻击包的详细信息；最后一项代表该条 Alert 被触发的次数。

表 5 警报聚类的处理结果

Table 5 Processing results of alert clustering

聚类结果信息
2018-01-22 15:58:34--SQL waitfor delay function in POST- possible SQL injection attempt----16045--16
2018-01-22 15:58:55--ET WEB_SERVER SQL Errors in HTTP 200 Response--200 OK--16881--2
2018-01-22 16:00:24--stream5:TCP Small Segment Threshold Exceeded------16887--1
2018-01-22 17:02:37--stream5:TCP Small Segment Threshold Exceeded------16900--14

根据表 5 所列的聚类后的结果，可以快速识别出攻击类型、攻击状态以及某时间内某种攻击出现的频率。

4.3 警报处理方法评估

4.3 节的实验通过白盒测试获取到的待处理的警报有 37 306 条，经过层次聚类结合时间窗口去重的方法得到的警报有 514 条。警报的聚类率达到了 98.6%，攻击类型占比为 100%。为了进一步评估本算法的性能，除了聚类率参数，本节还引入了检出率。检出率为被成功检测出的攻击行为占总的测试样本数的比例，即聚类后的警报为成功检测攻击的警报的占比。

为了验证本文提出的基于层次聚类的警报内容聚类方法的检测效果，本节将其与秦子燕的警报聚类系统[15]进行比较。聚类率的实验对比结果如图 11 所示，从中看出本文方法的聚类率较对比方法高出 17.3%。

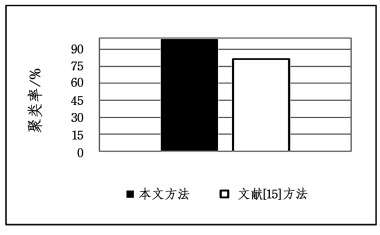


图 11 聚类率的对比

Fig. 11 Comparison of clustering rate

由于近年来警报聚类处理常常被用于警报关联处理的前一处理过程，因此从聚类率方面对比性能较为片面，例如梅海彬提出的攻击序列聚类方法[16]中聚类部分的聚类率仅有 38.84%。因此本文增加了检出率来评估警报处理方法的性能。将本文方法与徐小龙提出的基于蚁群聚类的入侵检测警报[7]过滤方法进行了比较，后者是通过结合蚁群算法和 K 均值聚类算法来对入侵检测警报进行过滤。检出率实验的对比结果如图 12 所示，从中看出本文方法的检出率较对比方法结果高出了 3.4%。

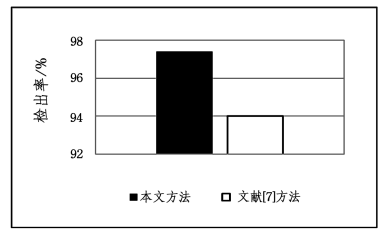


图 12 检出率的对比

Fig. 12 Comparison of detection rate

从对比实验的结果可以看出，在聚类率和检出率上，本文方法都优于对比方法。因此，本文提出的基于层次聚类的警报内容处理方法在一定程度上可以弥补传统聚类方法聚类效果和检出效果较差的缺陷，对攻击警报具有更加丰富的表征能力。

结束语 本文针对入侵检测系统的警报冗余问题，提出了改进层次聚类的算法，使用 Snort 入侵检测系统对实验的攻击结果数据进行检测，并将检测出来的警报结果进行层次聚类。该方法中进行了阈值的选取，并使用余弦相似度来计算聚类后的代表警报，选取 2 min 的时间窗口对按照时间戳排列的警报进行去重。实验结果表明，本文方法在聚类率和检出率上都有较优的性能。

参 考 文 献

[1] ZOU N. Research on Active Dynamic Network Security Defense of Network Management [D]. Changchun: Northeast Electric Power University, 2008. (in Chinese)
邹楠. 网络管理的主动式动态网络安全防御研究[D]. 长春：东北电力大学，2008.

[2] 祝世雄，陈周国，张小松，等. 网络攻击追踪溯源[M]. 北京：国防工业出版社，2015：75-79.

[3] CONG H Z. Design and implementation of high speed Network intrusion detection system based on Snort[D]. Jinan: Shandong University, 2016. (in Chinese)
丛海滋. 基于 Snort 的高速网络入侵检测系统的设计与实现[D]. 济南：山东大学，2016.

[4] JULISCH K. Clustering Intrusion Detection Alarms to Support Root Cause Analysis [J]. ACM Journal Name, 2002, 2(3): 111-138.

[5] CHEN X. Research on Intrusion Detection Alert Based on Conceptual Clustering Algorithm[J]. Journal of Air Force Radar Academy, 2004, 18(2): 28-30. (in Chinese)
陈新. 基于概念聚类算法的入侵检测警报研究[J]. 空军雷达学院学报，2004，18(2)：28-30.

[6] MEI H B. Research on discovering multi-stepattack patterns based on clustering IDS alert sequences[J]. Journal on Communications, 2011, 32(5): 63-69. (in Chinese)
梅海彬. 基于警报序列聚类的多步攻击模式发现研究[J]. 通信学报, 2011, 32(5): 63-69.

[7] XU X L. Intrusion Detection Alarms Filtering System Based on Ant Clustering Approach[J]. Electronic Technology, 2016(1): 34-37. (in Chinese)
徐小龙. 基于蚁群聚类的入侵检测警报过滤技术[J]. 电子技术, 2016(1): 34-37.

[8] Cisco. Snort- Network Intrusion Detection & Prevention System [EB/OL]. (2017-01-08) [2017-11-06]. https://www. snort. org/documents.

[9] GUO J F, ZHAO Y Y, BIAN W F, et al. Hierarchical clustering algorithm based on improved cohesion and separation[J]. Computer Research and Development, 2008, 45(1): 202-206. (in Chinese)
郭景峰, 赵玉艳, 边伟峰, 等. 基于改进的凝聚性和分离性的层次聚类算法[J]. 计算机研究与发展, 2008, 45(1): 202-206.

[10] DU Q, SUN M. Research on Intrusion Detection System Based on Improved Clustering Analysis Algorithm[J]. Computer Engineering and Applications, 2011, 47(11): 106-108. (in Chinese)
杜强, 孙敏. 基于改进聚类分析算法的入侵检测系统研究[J]. 计算机工程与应用, 2011, 47(11): 106-108.

[11] YANG B, LONG P F. Application of Condensed Hierarchical Clustering Algorithm in Intrusion Detection[J]. Journal of Electric Power Science and Technology, 2005, 20(3): 57-60. (in Chinese)
阳博, 龙鹏飞. 凝聚分层聚类算法在入侵检测中的应用[J]. 电力科学与技术学报, 2005, 20(3): 57-60.

[12] GU C Y. Text similarity calculation based on lexical semantic information[J]. Application Research of Computers, 2017, 35(2): 391-395. (in Chinese)
谷重阳. 基于词汇语义信息的文本相似度计算[J]. 计算机应用研究, 2017, 35(2): 391-395.

[13] DAVID G, BRIAN T. HTTP 权威指南[M]. 陈涓, 赵振平, 译. 北京: 人民邮电出版社, 2012: 62-69.

[14] LI H C. Alert multi-level aggregation and association method based on self-expansion time window[J]. Engineering Science and Technology, 2017, 49(1): 206-212. (in Chinese)
李洪成. 基于自扩展时间窗的告警多级聚合与关联方法[J]. 工程科学与技术, 2017, 49(1): 206-212.

[15] QIN Z Y, ZHAO Z Y. Alarm clustering for intrusion detection systems in network[J]. Journal of Computer Security, 2008(5): 27-30. (in Chinese)
秦子燕, 赵曾贻. 网络入侵检测系统中的警报聚类[J]. 计算机安全, 2008(5): 27-30.

[16] MEI H B, GONG J. Research on discovering multi-step attack patterns based on clustering IDS alert sequences[J]. Journal of Communications, 2011, 32(5): 63-69. (in Chinese)
梅海彬, 龚俭. 基于警报序列聚类的多步攻击模式发现研究[J]. 通信学报, 2011, 32(5): 63-69.