实训任务七：基于 Bind Log 分析用户互联网访问行为

一、目的

1、了解 Bind Log；

2、实现对 Bind Log 的数据分析。

二、学时

4 学时

三、类型

综合型

扫码看实训演示

扫码看任务步骤

四、需求

1、硬件

每人配备计算机 1 台，不低于双核 CPU、8G 内存、500GB 硬盘。

每小组配备服务器 1 台。

2、软件

Windows 操作系统，安装 puTTY 管理终端软件，安装 VMware ESXi 控制台软件。

服务器安装 VMware ESXi 7.0。

3、网络

计算机、服务器、虚拟主机使用固定 IP 地址接入局域网，并支持对互联网的访问。

4、工具

无。

五、任务要求

1、完成 Filebeat 的安装与配置；

2、实现 Bind Log 分析以达到用户互联网访问行为分析。

六、考核要求

1、提交《Bind Log 日志分析报告》；

2、提交 Bind Log 日志可视化分析成果截图/演示视频。

七、任务步骤

本任务需采集如表 7-1 所示服务器的 Bind Log 数据。

表 7-1 虚拟机列表

序号	虚拟机名称
1	YWSX-10.10.2.163-CentOS8X64-DNS

任务 1：配置 Bind

步骤 01：配置 Bind 的查询日志，其配置文件为/etc/named.conf。

配置文件：

```
logging {
        //查询日志
        channel query{
                //日志存储
                file "/var/log/named/query.log" versions 9 size 32m;
                severity info;
                print-category yes;
                print-severity yes;
                print-time yes;
        };
        category default {null;};
        category queries {query;};
        category network {null;};
        category client {null;};
        category general {null;};
}
```

步骤 02：重启 Bind

参考命令：

```
systemctl reload named
```

任务 2：使用 Filebeat 采集 Bind Log 数据

步骤 01：修改 Filebeat 配置文件

Filebeat 配置文件为/etc/filebeat/filebeat.yml。

配置文件：

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/named/query*
fields:
    indextype: dnslog-bind-10.10.2.163
  fields_under_root: true

output.Logstash:
  enabled: true
  hosts: ["10.10.2.155:5044"]
  topic: dnslog-bind-10.10.2.163
```

步骤 02：启动 Filebeat

参考命令：

```
filebeat -e -c /etc/filebeat/filebeat.yml
```

任务 3：使用 Logstash 清洗与格式化数据

步骤 01：配置 Logstash

创建 Logstash 配置文件，配置文件路径为/etc/logstash/conf.d/logstash_bind.conf，配置文件内容如下。

配置文件：

```
input {
      beats {
      port => 5044
   }
}

filter {
      if [indextype]== "dnslog-bind-10.10.2.163" {
            grok {
                  match => { "message" => "^(?<dns_time>.*)\s(?<dns_desc>\w*):\s(?<dns_lev
el>\w*):\sclient\s(?<client_hard_ip>.*)\s(?<client_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})#(?<client_po
rt>\w*)\s\((?<dns_ip_domain>.*)\):\squery:\s(?<dns_domain>.*?)\s(?<dns_class>.*?)\s(?<dns_typ
e>.*?)\s(?<dns_info>.*?)\s\((?<dns_ip>.*)\)"}
            # 如果成功匹配，则加一个匹配标签
                  add_tag => [ "matched" ]
            }
            date {
                  match => [ "dns_time" , "dd-MMM-yyyy HH:mm:ss.SSS" ]
                  target=>"dns_time_local"
            }
            mutate {
              # 移除不需要的字段
              remove_field => "agent"
              remove_field => "host"
              remove_field => "@version"
              remove_field => "ecs"
              remove_field => "input"
                  remove_field => "dns_ip_domain"
         }
            ruby {
                  path => "/etc/logstash/script/dns.rb"
                  remove_field => ["dns_time"]
            }
      }
}

output {
      if [indextype]== "dnslog-bind-10.10.2.163" {
            elasticsearch{
                  hosts => ["10.10.2.152:9200"]
                  index => "dnslog-bind-10.10.2.163-%{+YYYY.MM.dd}"
            }
      }
}
```

步骤 02：撰写清洗脚本

创建/etc/logstash/script/dns.rb，并编辑内容。

配置文件：

```
def filter(event)
        #查询时段
        begin
            dns_time_s=event.get('dns_time')
            if(dns_time_s!=""&&dns_time_s!="-"&&dns_time_s!=nil)
                dns_time=Time.new
                dns_time=Time.parse(dns_time_s)
                event.set('[dns_time_local_arr][year]', dns_time.year)
                if(dns_time.month.to_s.length==1)
                    event.set('[dns_time_local_arr][month]', "0"+dns_time.month.to_s)
                else
                    event.set('[dns_time_local_arr][month]', dns_time.month.to_s)
                end
                if(dns_time.day.to_s.length==1)
                    event.set('[dns_time_local_arr][day]', "0"+dns_time.day.to_s)
                else
                    event.set('[dns_time_local_arr][day]', dns_time.day.to_s)
                end
                event.set('[dns_time_local_arr][wday]', dns_time.wday)
                event.set('[dns_time_local_arr][yday]', dns_time.yday)

                if(dns_time.hour.to_s.length==1)
                    event.set('[dns_time_local_arr][hour]', "0"+dns_time.hour.to_s)
                else
                    event.set('[dns_time_local_arr][hour]', dns_time.hour.to_s)
                end
                event.set('[dns_time_local_arr][min]', dns_time.min)
                event.set('[dns_time_local_arr][sec]', dns_time.sec)
            end
        rescue
            event.set('error_time',dns_time_s)
        end
      return [event]
end
```

步骤 03：重启 Logstash

参考命令：

```
systemctl restart logstash
```

任务 4：在 Kibana 上创建分析模型并进行分析

步骤 01：设计分析模型

表 7-2 Bind Log 分析模型

序号	分析字段	分析模型	图表类型
1	-	日志总量	数字
2	dns_domain	域名解析总量	数字
3	client_ip	客户端 IP 总量	数字
4	local_time_arr.hour+dns_type	域名解析次数时间趋势	折线图
5	dns_type	域名访问统计排行	表格
6	dns_type	域名解析记录占比	饼图

7	dns_info	域名解析信息占比	饼图
8	local_time_arr.hour+dns_type	A 记录解析次数时间趋势	折线图
9	local_time_arr.hour+dns_type	AAAA 记录解析次数时间趋势	折线图
10	client_port	客户端端口占比	标签云

步骤 02：可视化分析

依据表 7-2 分型模型实现图表可视化，如图 7-1 所示。

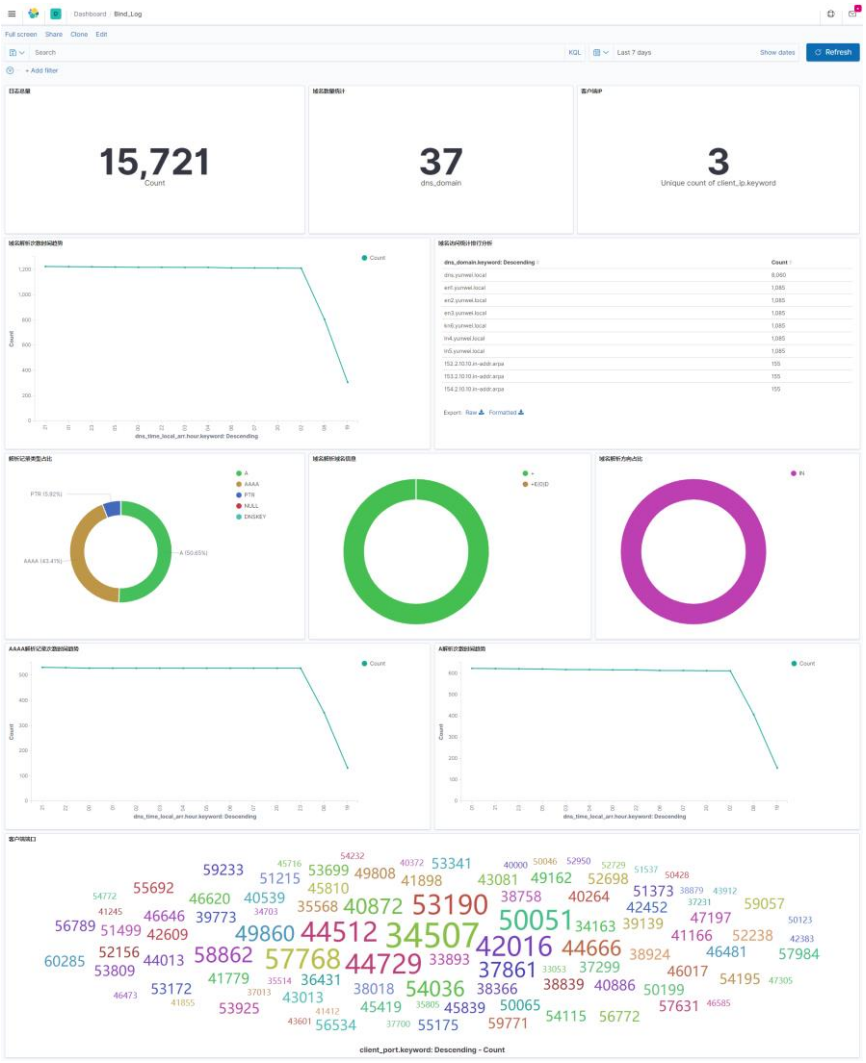

图 7-1 Bind Log 数据分析

八、实验思考

1、深入理解 Bind Log

（1）Bind Log 语法格式是什么？

（2）Bind Log 管道有哪些，分别有什么含义？

2、深入感知用户互联网访问行为

（1）用户互联网访问行为分析指标有哪些？

（2）可从哪些角度进行优化 DNS 解析？