# 威胁情报库

|  |  |  |  |
| --- | --- | --- | --- |
| 威胁情报数据 | | | |
| 序号 | 威胁数据类型 | 数据格式 | 数据量 |
| 1 | 恶意URL | 表1 |  |
| 2 | Who is 信息 | 表2 |  |
| 3 | 恶意样本 | 表3（孟工，暂无） |  |
| 4 | DNS | 表4 |  |
| 5 |  |  |  |
| 6 |  |  |  |

表1

|  |
| --- |
| **恶意URL表字段说明**   1. 数据库名： eval 2. 数据表名：malware 3. 字段：   （1）id ：int(100)，AI ，PK id序号，自增主键  （2）date ：varchar(100) 日期  **（3）url ：varchar(1000) 恶意 URL**  **（4）ip ：varchar(100) ip 地址**  （5）types ：varchar(100) 恶意 URL 类型  （6）country ：varchar(100) 国家  （7）city ：varchar(100) 城市  （8）position ：varchar(100) 位置坐标，经纬度  （9）reputation ：varchar(100) 信誉度  （10）risk：varchar(100) 风险  （11）flag ：varchar(100) 国旗  （12）asn ：varchar(100) ASN 号  （13）status ：varchar(100) 恶意 URL 状态  （14）host ：varchar(1000) 主机名  **（15）domain ：varchar(1000) 域名**  （16）registrant ：varchar(1000) 登记者  （17）files\_online ：varchar(100) 在线文件  （18）sbl ：varchar(100) 实时黑名单  （19）url\_class ：varchar(100) 恶意url等级  （20）contributor ：varchar(100) 发布者  （21）validate : varchar(100) 到期时间  （22）src：varchar(1000) 恶意网站检测来源  （23）relist：varchar(45) 是否重复写入  （24）testtime：varchar(1000) 曾检测为恶意网站时间 |

# 威胁情报库数据导入

原则：

1. 导入时增加辅助字段：\_\_ctime\_\_（创建时间）、\_\_utime\_\_（更新时间）、\_\_valid\_\_(有效标记)
2. 入库时：\_\_ctime\_\_ 与 \_\_utime\_\_默认都是当前系统时间，\_\_valid\_\_ = 'T'（'T'表示有效、'F'表示无效）
3. ~~字段名全部小写~~
4. 报警日志需要：将五元组的命名标准化，采用原来gw\_的方式，同时要参照powerv增加geo信息
5. 所有数据都用模板和mapping控制类型

# 模板

新增加一个type时，需要单独刷该type的mapping（为了保持一致，模板也刷，虽然不起作用）

刷mapping：

curl -XPUT 'localhost:9200/template\_threat\_intelligence/whois/\_mapping' -d '{

"dynamic": true,

"\_all": { "enabled": true },

"\_source": { "enabled": true },

"properties": {

"threatdb": {

"type": "string"

},

"\_\_ctime\_\_": {

"type": "date",

"format" : "dateOptionalTime"

},

"\_\_utime\_\_": {

"type": "date",

"format" : "dateOptionalTime"

},

"\_\_valid\_\_": {

"type": "string",

"index": "not\_analyzed"

},

"whois": {

"properties": {

"ASN": {

"type": "string"

},

"city": {

"omit\_norms": true,

"path": "just\_name",

"type": "multi\_field",

"fields": {

"rawcity": {

"index": "not\_analyzed",

"type": "string"

},

"city": {

"type": "string"

}

}

}

}

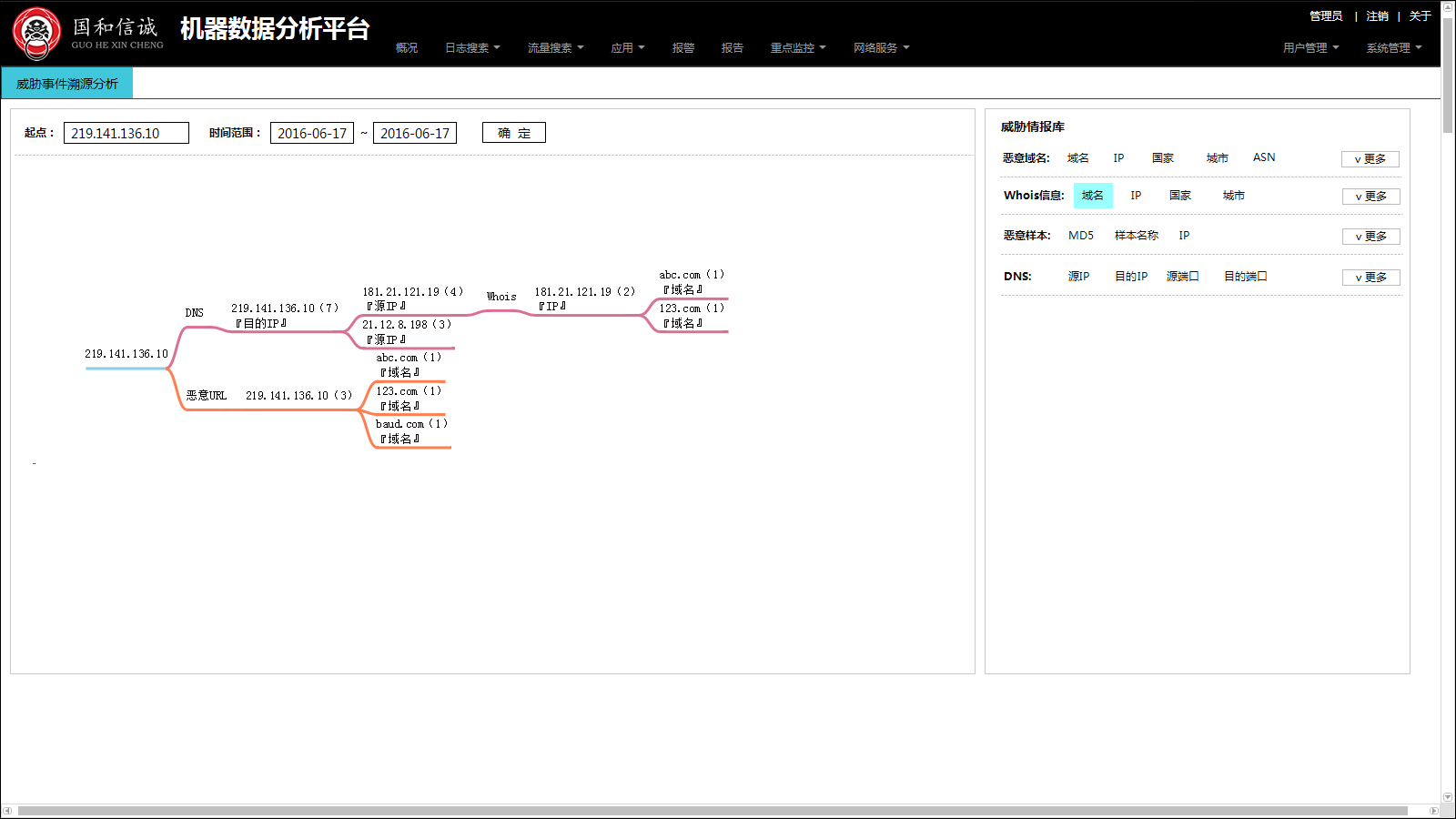
}

}

}

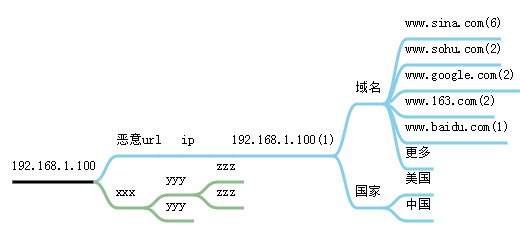
# 威胁事件溯源分析

## 界面

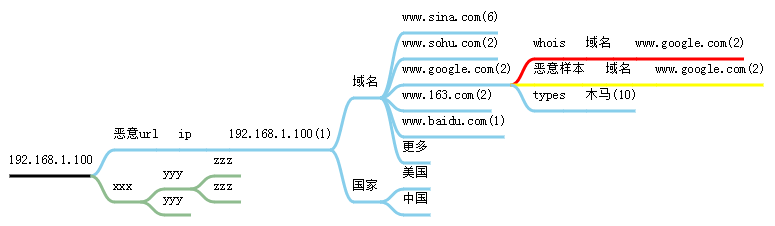


## 功能

1. 条件设置
2. 起点：任意输入IP、域名等内容；
3. 点击“确定”按钮后，将“起点”内容放到图中起点位置；
4. 威胁情报库
5. 右边的“威胁情报库”，每一栏为一个库，前面为库名称，后面为字段名，默认每个库只显示一行的字段名称，点击“更多”按钮可以显示该库的全部字段；
6. 可以同时选择多个库的多个字段，点击选中后颜色改变，可再点击取消选择，颜色恢复；
7. 溯源流程
8. 结点：分为四种
9. root结点：即起点，只显示起点信息，可以根据威胁情报库的选择进行下钻；
10. 库结点：只用来显示库名称，不能下钻；
11. 字段结点： 只用来显示字段名称，不能下钻；
12. 值结点：显示统计值，可以根据威胁情报库的选择进行下钻；右键菜单可显示与该结点相关的在库中的相关其他属性字段信息，在弹出框中显示出来（分页显示，一条一页）。
13. 下钻方法：
14. 选中图中的某个值结点；
15. 选择右边的“威胁情报库”的字段，可选择多个库的多个结点，也可选择同库的多个结点；
16. 再点击被选中字段结点的“+”图标进行下钻展示；
17. 默认每个值结点只显示5个值加上“更多”，点击“更多”，可再增加5个值；
18. 已下钻过的值结点，仍然可以选择其他库字段进行下钻（删咋办？已选中的不选了相当于删除）
19. 下钻原则：
20. “root结点”和“值结点”可以根据选择的威胁情报库的字段进行下钻；
21. 同库下钻：下钻结点必须为“root结点”或“值结点”，选择多个字段，相同库的连线用相同颜色表示。如下图所示，蓝色的为同库下钻：



1. 跨库下钻：下钻结点必须为“root结点”或“值结点”，



1. 点击每个值结点时，在各个资源库中进行全文检索，显示在资源库名边上
2. 库结点不能下钻；
3. 字段结点不能下钻；



## 表结构

t\_threatinfo\_db：威胁情报库字段表

说明：

drill=0表示此字段可以下钻，需要在右边的危险情报库中显示出来，以供选择，1则不显示

property=0表示此字段需要作为tooltip显示在所有该库的下钻字段上，用来说明，1则不显示

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **类型** | **NULL** | **索引** | **默认值** | **说明** |
| id | bigserial | N | Y，KEY |  | 主键。自生成，自增长 |
| index\_name | character varying(100) | N | Y |  | 索引名称（改已为存type名称） |
| index\_name\_[suffix](javascript:void(0);) | character varying(100) | Y | N |  | 索引名称后缀，'yyyy.mm.dd'，为空表示只有一个索引名称 |
| display\_index\_name | character varying(100) | Y | N |  | 索引显示名称（改已为存type名称） |
| field\_name | character varying(100) | N | Y |  | 字段名称 |
| display\_name | character varying(100) | Y | N |  | 字段显示名称 |
| drill | smallint | N | Y | 0 | 0：下钻字段，1：非下钻字段 |
| property | smallint | N | Y | 1 | 0：属性字段，1：非属性字段 |
| color | character varying(100) | N |  | 'black' | 颜色，用于区分库和线条的颜色 |
| desc | character varying(100) | Y | N |  | 备注 |

## 语句

1. 取威胁库：

|  |
| --- |
| select index\_name, display\_index\_name  from t\_threatinfo\_db  group by index\_name, display\_index\_name; |

1. 取威胁库字段

|  |
| --- |
| select id, index\_name, display\_index\_name, field\_name, color, case when (display\_name is null or trim(display\_name) = '') then field\_name else display\_name end display\_name  from t\_threatinfo\_db  where drill = 0; |

1. 字段统计
2. 循环每个type，每个type里再循环每个字段（drill=0的字段）
3. 同库下钻时，要带前面下钻的条件，跨库或root下钻时，只带本次条件
4. 查询条件里的字段名称格式为：type名称.字段名，如mal\_url.country
5. 统计里的字段名称格式为：type名称.raw+字段名，如mal\_url.rawcountry

|  |
| --- |
| curl -XGET 'http://localhost:9200/idx\_threat\_intelligence/**mal\_url**/\_search?search\_type=count&pretty' -d '{  "query": {  "filtered": {  "query": {  "query\_string": {  "query": "**mal\_url.ip:10.1.1.1 AND mal\_url.country:中国**", ---蓝色的表示前面同库下钻的条件，红色为本次同库下钻的条件  "default\_operator": "AND",  "default\_field": "\_all"  }  }  }  },  "size": 0  }' |

1. root结点下钻
2. root下钻是允许选择多库的，所以下面的语句可能会对多个不同type进行统计。
3. root下钻时，初始条件即为用户选择的威胁库中的字段，统计字段也为该字段。

|  |
| --- |
| curl -XGET 'http://localhost:9200/idx\_threat\_intelligence/**mal\_url**/\_search?pretty' -d '{  "query": {  "filtered": {  "query": {  "query\_string": {  "query": "**mal\_url.ip:10.1.1.1**",  "default\_operator": "AND",  "default\_field": "\_all"  }  }  }  },  "from": 0,  "aggs": {  "gp1":{  "terms": {  "field" : "**mal\_url.rawip**",  "size" : 10,  "order" : {"\_count" : "desc"}  }  }  },  "size": 0  }' |

1. 值结点同库下钻
2. 索引不变。
3. 条件为本次连续同库的所有字段条件。
4. 统计字段为选择的威胁库字段。

|  |
| --- |
| curl -XGET 'http://localhost:9200/idx\_threat\_intelligence/**mal\_url**/\_search?pretty' -d '{  "query": {  "filtered": {  "query": {  "query\_string": {  "query": "**mal\_url.ip:10.1.1.1 AND mal\_url.ip:10.1.1.1**", ----第一个为初始条件，第二个为ip字段下钻的条件，二者重复没关系  "default\_operator": "AND",  "default\_field": "\_all"  }  }  }  },  "from": 0,  "aggs": {  "gp1":{  "terms": {  "field" : "**mal\_url.rawdomain**",  "size" : 10,  "order" : {"\_count" : "desc"}  }  }  },  "size": 0  }' |

1. 值结点跨库下钻
2. 索引为选择的威胁库。
3. 条件字段为本次选择的威胁库字段名，条件值为下钻结点的值。
4. 统计字段为本次选择的威胁库字段名。

|  |
| --- |
| curl -XGET 'http://localhost:9200/idx\_threat\_intelligence/**whois**/\_search?pretty' -d '{  "query": {  "filtered": {  "query": {  "query\_string": {  "query": "**whois.domain:** **sina.com.cn**",  "default\_operator": "AND",  "default\_field": "\_all"  }  }  }  },  "from": 0,  "aggs": {  "gp1":{  "terms": {  "field" : "**whois.rawdomain**",  "size" : 10,  "order" : {"\_count" : "desc"}  }  }  },  "size": 0  }' |

1. 取值结点的属性字段内容
2. 先取某情报库的属性字段
3. 取当前选中的“值结点”的**连续同库**钻取路径为条件

|  |
| --- |
| **取属性字段：**  select index\_name|| '.' || field\_name field\_name, display\_index\_name || '.' || case when (display\_name is null or trim(display\_name) = '') then field\_name else display\_name end display\_name from t\_threatinfo\_db where index\_name = 'mal\_url' and property = 0;  **取属性字段值：**  curl -XGET 'http://localhost:9200/idx\_threat\_intelligence/**mal\_url**/\_search?pretty' -d '{  "query" : {  "filtered" : {  "query" : {  "query\_string" : {  "query": "**mal\_url.ip:10.1.1.1 AND mal\_url.domain:www.sina.com**",  "default\_operator": "AND",  "default\_field": "\_all"  }  }  }  },  "fields" : ["mal\_url.registrant", "mal\_url.src", "mal\_url.contributor"], ----字段是通过上面的sql取出来的  "from" : 0,  "size" : 100  }' |

# 概况-威胁情报库列表

## 页面



## 功能

1. 威胁情报库列表：
2. 概况页面显示威胁库的统计信息，字段包括：威胁情报库名称、数据总量、增量
3. 替代原来的“报警数列表”
4. 取数据，先从t\_threatinfo\_db表里取出“威胁情报库名称”和“索引名”，然后在es中查出威胁库的数据量
5. 5分钟定时刷新
6. 报警日志列表：
7. 按分组统计日志总量及增量；
8. 点击某组，显示该组中各dn的日志总量及增量
9. 5分钟定时刷新

## 语句

1. 取威胁库名称

|  |
| --- |
| select index\_name, display\_index\_name  from t\_threatinfo\_db  group by index\_name, display\_index\_name; |

1. 取各威胁库数据量

|  |
| --- |
| 总量：  curl -XGET 'http://localhost:9200/idx\_threat\_intelligence/mal\_url/\_search?search\_type=count&pretty'  当日增量：  curl -XGET 'http://localhost:9200/idx\_threat\_intelligence/mal\_url/\_search?pretty' -d '{  "size" : 0,  "query":{  "filtered": {  "query" : {  "query\_string" : {  "query" : "\*",  "default\_field" : "\_all",  "default\_operator" : "and"  }  },  "filter": {  "and": {  "filters": [  {  "range": {  "\_\_ctime\_\_": {  "from": "2016-08-11T00:00:00+08:00",  "to": "2016-08-11T23:59:59+08:00",  "include\_lower": true,  "include\_upper": true  }  }  }  ]  }  }  }  }  }' |

1. 按分组统计日志总量及增量：

|  |
| --- |
| **今日前的总量：**  select tb3.groupname, sum(tb1.statscnt) from t\_statsbyhost\_day tb1, t\_syslog tb2, t\_confgroup tb3  where tb1.host = tb2.logsource  and tb2.groupid = tb3.groupid  and statsdate > (now() - cast((select svalue || ' days' from t\_settings where skey = 'logs\_retain') as interval))  group by tb3.groupname;  **今日增量：**  curl -XGET 'http://localhost:9200/datagroup-2016.09.12/\_search?pretty' -d '{  "size" : 0,  "query":{  "filtered": {  "query" : {  "query\_string" : {  "query" : "\*",  "default\_field" : "\_all",  "default\_operator" : "and"  }  }  }  },  "aggs": {  "gp1":{  "terms": {  "field" : "msg.group",  "size" : 1000  }  }  }  }'  **总量：今日前的总量+今日增量** |

1. 统计某分组中的dn的日志总量及增量：

|  |
| --- |
| **今日前的总量：**  select tb2.configpath, sum(tb1.statscnt) from t\_statsbyhost\_day tb1, t\_syslog tb2, t\_confgroup tb3  where tb1.host = tb2.logsource  and tb2.groupid = tb3.groupid  and statsdate > (now() - cast((select svalue || ' days' from t\_settings where skey = 'logs\_retain') as interval))  and tb3.groupname = 'firewall'  group by tb2.configpath;  **今日增量：**  curl -XGET 'http://localhost:9200/datagroup-2016.09.12/\_search?pretty' -d '{  "size" : 0,  "query":{  "filtered": {  "query" : {  "query\_string" : {  "query" : "msg.group:\"firewall\"",  "default\_field" : "\_all",  "default\_operator" : "and"  }  }  }  },  "aggs": {  "gp1":{  "terms": {  "field" : "msg.dn",  "size" : 1000  }  }  }  }'  **总量：今日前的总量+今日增量** |

1. ~~按类别取各类报警日志的总量及增量：~~

|  |
| --- |
| **~~今日前的总量：~~**  ~~select dn, sum(statscnt) from t\_statsbydn\_day~~  ~~where statsdate > (now() - cast((select svalue || ' days' from t\_settings where skey = 'logs\_retain') as interval))~~  ~~group by dn~~  ~~order by dn;~~  **~~今日增量：~~**  ~~curl -XGET 'http://localhost:9200/datagroup-2016.09.12/\_search?pretty' -d '{~~  ~~"size" : 0,~~  ~~"query":{~~  ~~"filtered": {~~  ~~"query" : {~~  ~~"query\_string" : {~~  ~~"query" : "\*",~~  ~~"default\_field" : "\_all",~~  ~~"default\_operator" : "and"~~  ~~}~~  ~~}~~  ~~}~~  ~~},~~  ~~"aggs": {~~  ~~"gp1":{~~  ~~"terms": {~~  ~~"field" : "msg.dn",~~  ~~"size" : 10~~  ~~}~~  ~~}~~  ~~}~~  ~~}'~~  **~~总量：今日前的总量+今日增量~~** |

1. ~~取某一类中各台设备的报警日志：~~

|  |
| --- |
| **~~今日前的总量：~~**  ~~select tb1.host, sum(tb1.statscnt) from t\_statsbyhost\_day tb1, t\_syslog tb2~~  ~~where tb1.host = tb2.logsource~~  ~~and statsdate > (now() - cast((select svalue || ' days' from t\_settings where skey = 'logs\_retain') as interval))~~  ~~and tb2.configpath = 'hw:leadsec:powerv'~~  ~~group by tb1.host;~~  **~~今日增量：~~**  ~~curl -XGET 'http://localhost:9200/datagroup-2016.09.12/\_search?pretty' -d '{~~  ~~"size" : 0,~~  ~~"query":{~~  ~~"filtered": {~~  ~~"query" : {~~  ~~"query\_string" : {~~  ~~"query" : "msg.dn:\"hw:leadsec:powerv\"",~~  ~~"default\_field" : "\_all",~~  ~~"default\_operator" : "and"~~  ~~}~~  ~~}~~  ~~}~~  ~~},~~  ~~"aggs": {~~  ~~"gp1":{~~  ~~"terms": {~~  ~~"field" : "msg.host",~~  ~~"size" : 1000~~  ~~}~~  ~~}~~  ~~}~~  ~~}'~~  **~~总量：今日前的总量+今日增量~~** |

# 搜索-威胁情报库

## 页面

## 功能

1. 搜索条件：增加“威胁情报库”选项，取威胁库名称，可全选、多选、单选，默认全选；
2. 搜索：点击“搜索”按钮时，分别去各威胁库查询；
3. 查询结果：以tab标签区分各威胁库的搜索结果，点击某个威胁情报库显示相应的记录。每个tab里的数据以表格形式显示，列头为selected=1的字段，通过详细看所有字段，列表左边，为该类威胁情报库的所有字段，可以“选择/不选择”作为表格列头字段；

## 语句

1. 取威胁库名称（[参见之前的sql语句](#语句（取威胁库名称）)）
2. 取字段名称，及表格列头标记

|  |
| --- |
| select index\_name|| '.' || field\_name field\_name, display\_index\_name || '.' || case when (display\_name is null or trim(display\_name) = '') then field\_name else display\_name end display\_name, selected from t\_threatinfo\_db where index\_name = 'mal\_url'; |

1. “选择/不选择”字段作为表格列头

|  |
| --- |
| 选择：update t\_threatinfo\_db set selected = 1 where index\_name = 'mal\_url' and field\_name = split\_part('mal\_url.asn', '.', 2);  不选择：update t\_threatinfo\_db set selected = 0 where index\_name = 'mal\_url' and field\_name = split\_part('mal\_url.asn', '.', 2); |

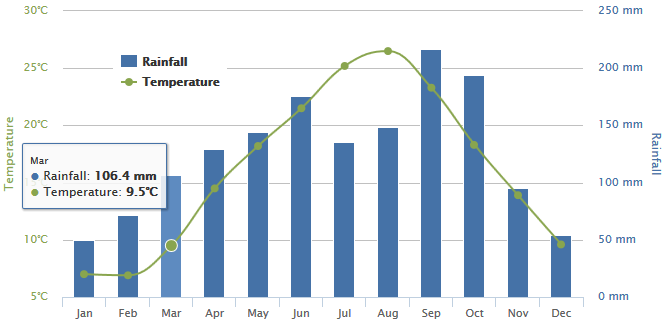
1. 威胁库查询：

|  |
| --- |
| curl -XGET 'http://localhost:9200/idx\_threat\_intelligence/**whois**/\_search?pretty' -d '{  "query" : {  "filtered" : {  "query" : {  "query\_string" : {  "query": "google.com",  "default\_operator": "AND",  "default\_field": "\_all"  }  }  }  },  "fields" : ["whois.clientip", "whois.bytes"], ----字段是通过上面的sql取出来的  "from" : 0,  "size" : 10  }' |

# 威胁态势展现

## 功能

1. 报警日志数据态势分析



1. 7日数据总量柱状及增量曲线态势展现

时间：前7日，不包括当天

图表：柱状图与曲线图的混合

柱状图：表示总量？？

曲线图：表示增量

取数据：

1. 30日数据总量柱状及增量曲线态势展现。

时间：前30天，不包括当天

图表：柱状图与曲线图的混合

柱状图：表示总量？？

曲线图：表示增量

取数据：

1. 威胁情报数据态势分析
2. 日志报警趋势展现

见地图

## 语句

1. 7日、30日报警日志

|  |
| --- |
| 取日志保留天数：  select svalue from t\_settings where skey = 'logs\_retain';  增量：  select tb1.statsdate, tb1.logscnt, sum(tb2.logscnt) from t\_stats\_day tb1, t\_stats\_day tb2  where tb1.statsdate >= tb2.statsdate  and tb1.statsdate between '2016-08-26' and '2016-09-01' ---- 7天或30天范围，不包括今天  and tb2.statsdate >= (now() - interval '15 days') ---- '15 days' 由上面的日志保留天数取得  group by tb1.statsdate, tb1.logscnt  order by tb1.statsdate; |

1. 7日、30日威胁情报库

|  |
| --- |
| 取7日（30日）前的总量：近7日（30日）的总量，需要再根据后面的每日增量自己计算  curl -XGET 'http://localhost:9200/idx\_threat\_intelligence/\_search?pretty' -d '{  "size" : 0,  "query":{  "filtered": {  "query" : {  "query\_string" : {  "query" : "\*",  "default\_field" : "\_all",  "default\_operator" : "and"  }  },  "filter": {  "and": {  "filters": [  {  "range": {  "\_\_ctime\_\_": {  "to": "2016-08-11T00:00:00+08:00", ----7日（30日）期前的日期  "include\_lower": true,  "include\_upper": false  }  }  }  ]  }  }  }  }  }'  增量数据：取7天或30天的每天的数据，需要补齐  curl -XGET 'http://localhost:9200/idx\_threat\_intelligence/\_search?pretty' -d '{  "size" : 0,  "query":{  "filtered": {  "query" : {  "query\_string" : {  "query" : "\*",  "default\_field" : "\_all",  "default\_operator" : "and"  }  },  "filter": {  "and": {  "filters": [  {  "range": {  "\_\_ctime\_\_": {  "from": "2016-08-01T00:00:00+08:00",  "to": "2016-09-01T00:00:00+08:00",  "include\_lower": true,  "include\_upper": true  }  }  }  ]  }  }  }  },  "aggs": {  "gp1":{  "date\_histogram": {  "field" : "\_\_ctime\_\_",  "interval" : "1d"  }  }  }  }' |

# 地图

## 功能

1. 地图：每隔1分钟取上一分钟的数据，源城市-->目标城市、攻击次数，然后通过t\_city\_location查出经纬度，在地图上循环展示
2. 攻击源城市TOP5：最近60分钟或今日，5分钟自动刷新
3. 受攻击城市TOP5：最近60分钟或今日，5分钟自动刷新
4. 攻击源IP TOP5：最近60分钟或今日，5分钟自动刷新
5. 受攻击IP TOP5：最近60分钟或今日，5分钟自动刷新
6. 最近60分钟，实时攻击数量，1分钟自动刷新

## 语句

1. 地图

|  |
| --- |
| 取前一分钟的源城市-->目标城市、攻击次数：  curl -XGET 'http://localhost:9200/datagroup-2016.09.12/\_search?pretty' -d '{  "size" : 0,  "query":{  "filtered": {  "query" : {  "query\_string" : {  "query" : "\_exists\_:gw\_src\_city AND \_exists\_:gw\_dst\_city AND gw\_src\_country:China AND gw\_dst\_country:China", ---- 蓝色的为固定条件，红色的根据用户选择来定，可能有一个，肯能有两个，也可能没有  "default\_field" : "\_all",  "default\_operator" : "and"  }  },  "filter": {  "and": {  "filters": [  {  "range": {  "@timestamp": {  "from": "2016-09-12T09:12:00+08:00",  "to": "2016-09-12T09:13:00+08:00",  "include\_lower": true,  "include\_upper": false  }  }  }  ]  }  }  }  },  "aggs": {  "gp1":{  "terms": {  "field" : "gw\_src\_city",  "size" : 500  },  "aggs": {  "gp2": {  "terms": {  "field": "gw\_dst\_city",  "size": 100  }  }  }  }  }  }'  取城市经纬度：  select city\_name, latitude\_number, longitude\_number from t\_city\_location where city\_name in ('Ho Chi Minh City', 'Beijing');  取国家列表：  select country\_name, country\_name\_cn from t\_country\_location; |

1. TOP5

|  |
| --- |
| curl -XGET 'http://localhost:9200/datagroup-2016.09.14/\_search?pretty' -d '{  "size" : 0,  "query":{  "filtered": {  "query" : {  "query\_string" : {  "query" : "\_exists\_:gw\_src\_city AND \_exists\_:gw\_dst\_city AND gw\_src\_country:China AND gw\_dst\_country:China",  "default\_field" : "\_all",  "default\_operator" : "and"  }  },  "filter": {  "and": {  "filters": [  {  "range": {  "@timestamp": {  "from": "2016-09-14T15:00:00+08:00", ----最近60分钟或今日  "to": "2016-09-14T16:00:00+08:00",  "include\_lower": true,  "include\_upper": false  }  }  }  ]  }  }  }  },  "aggs": {  "gp1":{  "terms": {  "field" : "gw\_src\_city", ----攻击源城市：gw\_src\_city 受攻击城市：gw\_dst\_city 攻击源IP：gw\_src\_ipv4 受攻击IP：gw\_dst\_ipv4  "size" : 5  }  }  }  }' |

1. 近60分钟实时攻击数量

|  |
| --- |
| curl -XGET 'http://localhost:9200/datagroup-2016.09.19/\_search?pretty' -d '{  "size" : 0,  "query":{  "filtered": {  "query" : {  "query\_string" : {  "query" : "\_exists\_:gw\_src\_city AND \_exists\_:gw\_dst\_city AND gw\_src\_country:China AND gw\_dst\_country:China",  "default\_field" : "\_all",  "default\_operator" : "and"  }  },  "filter": {  "and": {  "filters": [  {  "range": {  "@timestamp": {  "from": "2016-09-19T13:00:00+08:00",  "to": "2016-09-19T14:00:00+08:00",  "include\_lower": true,  "include\_upper": false  }  }  }  ]  }  }  }  },  "aggs": {  "gp1":{  "date\_histogram": {  "field" : "@timestamp",  "interval" : "10s"  }  }  }  }'; |

# 自定义场景

## 功能

1. 自定义场景定义
2. 参照“流量主机监控”的定义界面，界面上以4×5的方式显示方格，最多定义20个自定义场景的条件，可以自由拖动排序；
3. 添加：点击未定义的方格，弹出自定义对话框

添加项：名称、自定义条件、备注

自定义条件：参照“日志搜索”里的“保存搜索条件”

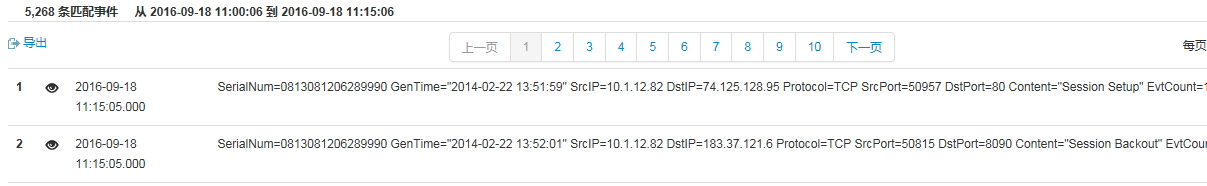
1. 修改：
2. 删除：
3. 查询跳转：点击已定义的方格，带着自定义的查询条件跳转到自定义场景查询页面；
4. 权限：访问权限和修改权限，没有修改的权限就不能增、删、改；
5. 自定义场景查询
6. 查询条件：

自定义场景条件：在“自定义场景中”定义的条件

固定条件：源IP、目的IP、源端口、目的端口、协议、源国家、目的国家、时间范围、自定义

1. 查询结果：

参照“日志查询”中的结果，去掉柱状图、去掉左边的已选字段，结果里只包括原始日志和眼睛



导出最大条数改为5万条

1. 已有功能修改：
2. “日志搜索”中的“保存搜索条件”，增加“保存为搜索场景”选项，保存时如果勾选上此项，则该条件会在搜索场景定义页面中显示，默认是不选中的；
3. 日志搜索管理页面也按1）的要求修改

## 语句

取自定义场景列表：

|  |
| --- |
| select tb1.searchid, tb1.searchname, tb1.description, tb1.searchcond, tb2.sort\_no  from t\_savedsearch tb1, t\_savedsearch\_scene tb2  where tb1.searchid = tb2.searchid  order by sort\_no; |

添加自定义场景：

|  |
| --- |
| insert into t\_savedsearch(searchname, description, searchcond, searchstart, searchend, creator, creatime, lastmodifier, lastmoditime)  values('名称', '描述', '条件', '-15minutes@second', 'now', 'admin', now(), 'admin', now()); ----蓝色为固定值  select currval('t\_savedsearch\_searchid\_seq'); ----取主键值  insert into t\_savedsearch\_scene(searchid, sort\_no) values(123, 12); ----上面取的主键值,排序号为方格的位置 |

修改：

|  |
| --- |
| update t\_savedsearch set searchname='名称', description='描述', searchcond='条件', lastmodifier='admin', lastmoditime=now() where searchid = 123; |

删除：

|  |
| --- |
| delete from t\_savedsearch where searchid = 123;  delete from t\_savedsearch\_scene where searchid = 123; |

拖动排序：

|  |
| --- |
| update t\_savedsearch\_scene set sort\_no = 8 where searchid = 123; |