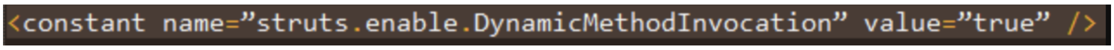
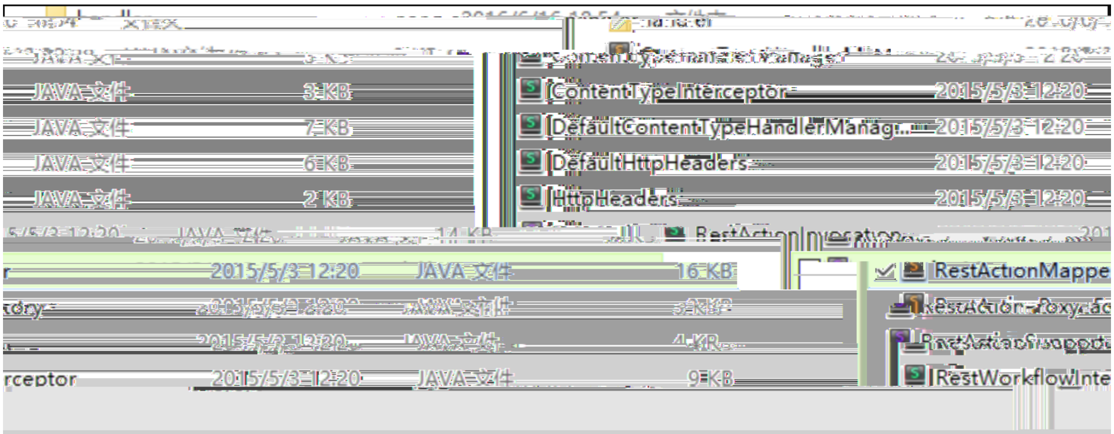
RELIABLE NETWORK
SAFE WORLD
TOPSEC

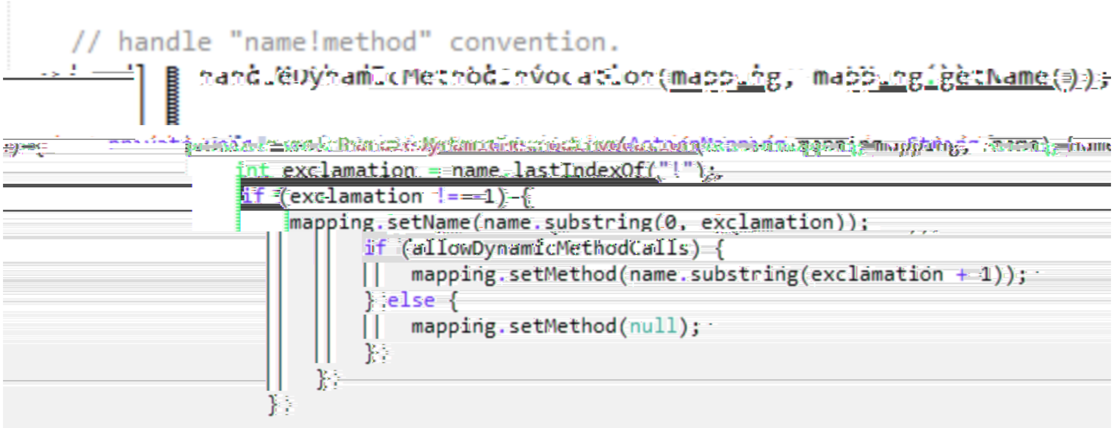
□

□

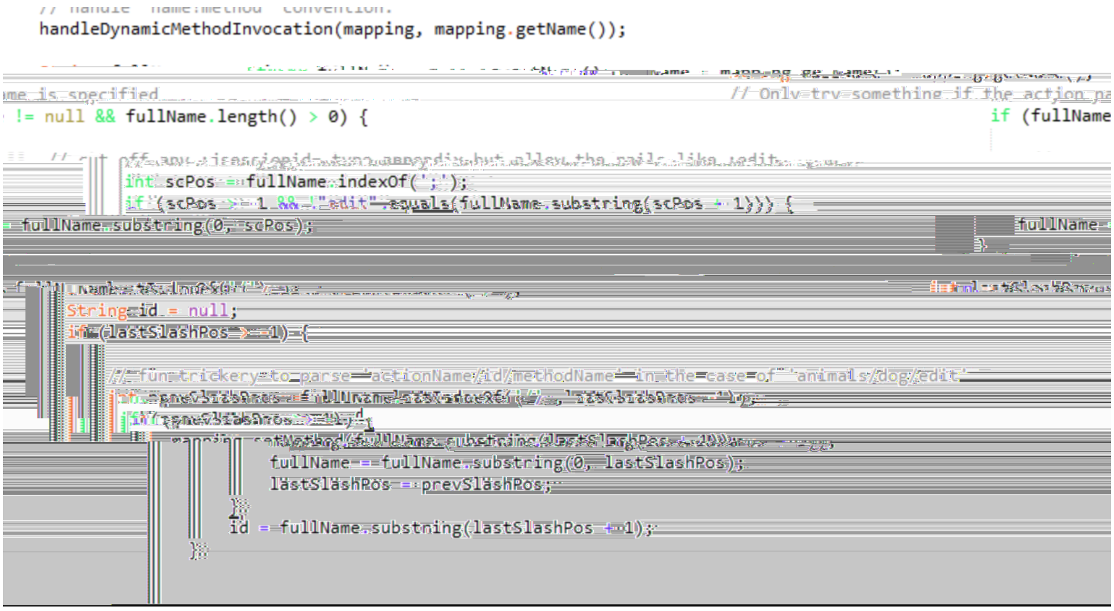


```
<constant name="struts.enable.DynamicMethodInvocation" value="true" />
```

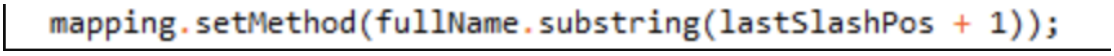

ContentTypeInterceptor
DefaultContentTypeHandlerManag...
DefaultHttpHeaders
HttpHeaders
RestActionInvocation
RestActionMappe
RestWorkflowInte
JAVA 文件
2015/5/3 12:20
3 KB
7 KB
6 KB
2 KB
14 KB
16 KB
4 KB
9 KB



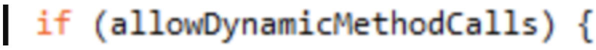
```
    // handle "name!method" convention.
    handleDynamicMethodInvocation(mapping, mapping.getName());

private void handleDynamicMethodInvocation(ActionMapping mapping, String name) {
        int exclamation = name.lastIndexOf("!");
        if (exclamation != -1) {
            mapping.setName(name.substring(0, exclamation));
            if (allowDynamicMethodCalls) {
                mapping.setMethod(name.substring(exclamation + 1));
            } else {
                mapping.setMethod(null);
            }
        }
    }
```



```
    // handle "name!method" convention.
    handleDynamicMethodInvocation(mapping, mapping.getName());

[illegible]
me is specified                                  // Only try something if the action na
 != null && fullName.length() > 0) {                                     if (fullName

    // cut off any ;jsessionid= type appendix but allow the rails-like ;edit
        int scPos = fullName.indexOf(';');
        if (scPos > -1 && !"edit".equals(fullName.substring(scPos + 1))) {
 fullName.substring(0, scPos);                                          fullName
                                                                        }

[illegible]
        String id = null;
        if (lastSlashPos > -1) {

            // fun trickery to parse 'actionName/id/methodName' in the case of 'animals/dog/edit'
            int prevSlashPos = fullName.lastIndexOf('/', lastSlashPos - 1);
            if (prevSlashPos > -1) {
                mapping.setMethod(fullName.substring(lastSlashPos + 1));
                fullName = fullName.substring(0, lastSlashPos);
                lastSlashPos = prevSlashPos;
            }
            id = fullName.substring(lastSlashPos + 1);
        }
```



```
mapping.setMethod(fullName.substring(lastSlashPos + 1));
```



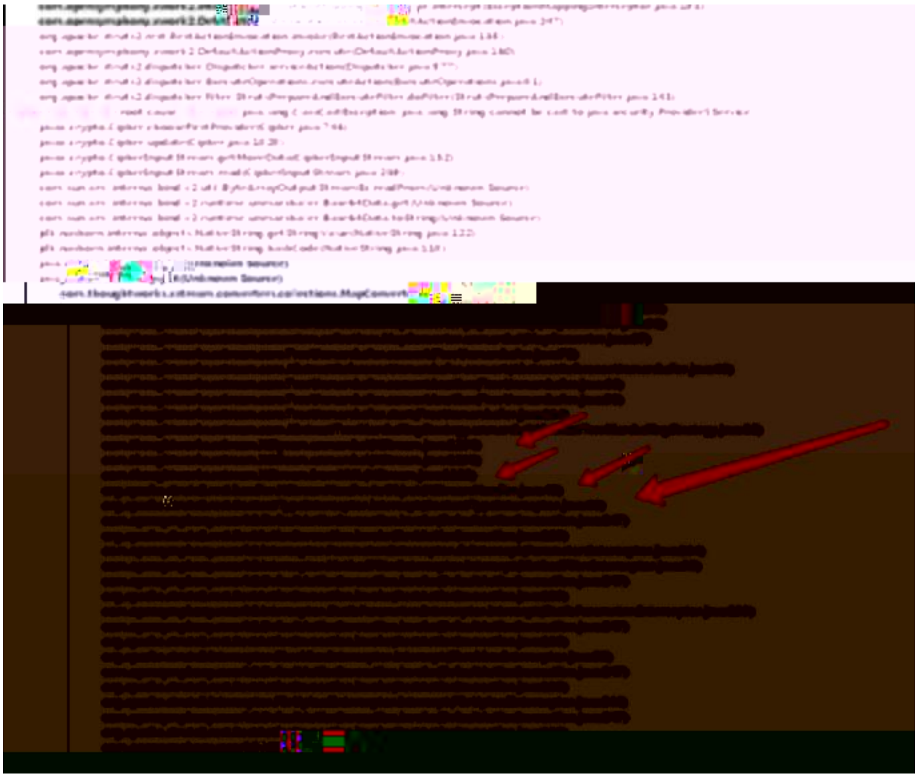
```
if (allowDynamicMethodCalls) {
```

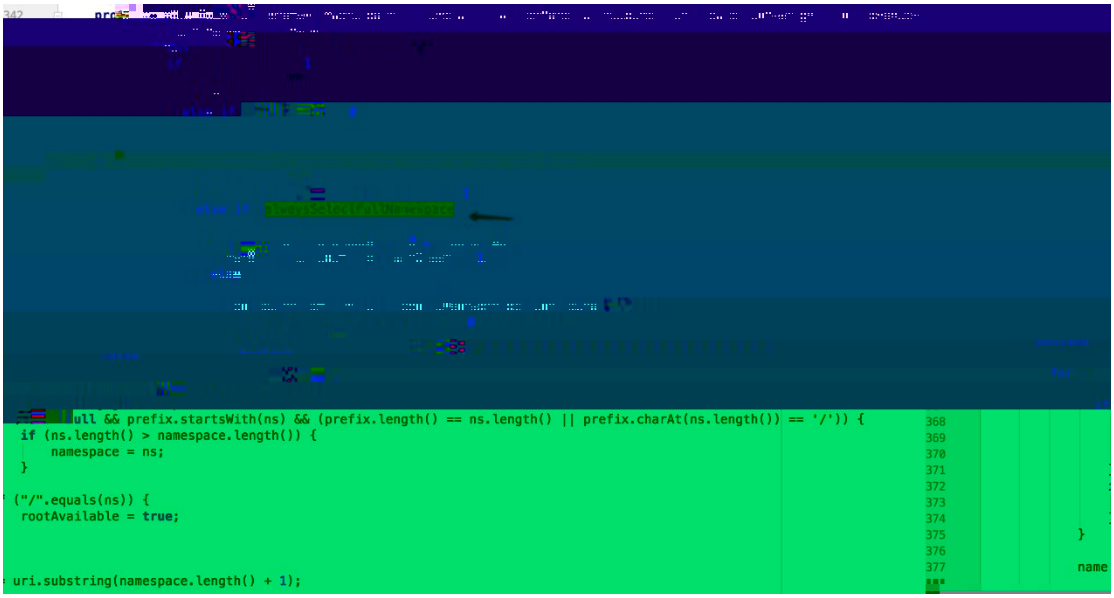
```
// fun trickery to parse 'actionName/id/methodName' in the case of 'animals/dog/edit'
```

□

```
@Inject
public void setConfiguration(Configuration config) {
    this.configuration = config;
}

public String execute() throws Exception {
[illegible]
        = null;
    try {
                                                            null);
    } catch
        throw new
    }

    //

                                   = new

                                                          quest();
    HttpServletResponse response = Servle[illegible]xt.getResponse();
        ActionForward forward = action.execute(mapping, actionForm, request, response);

        ActionMessages messages = (ActionM[illegible]
          if (                   = null) {
              for (
                  if (             != null &&
                  } else {

              }
          }
      }
```

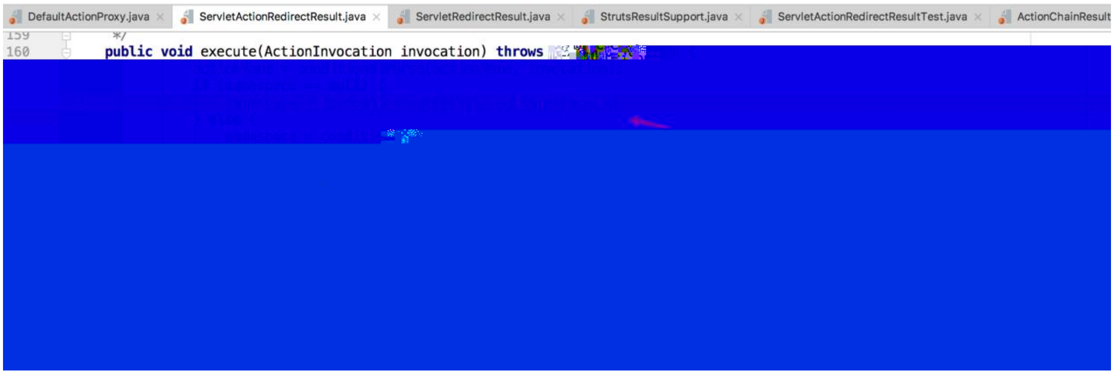
□



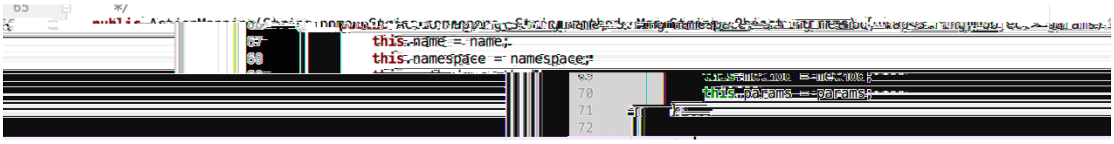


```
else if alwaysSelectFullNamespace
```

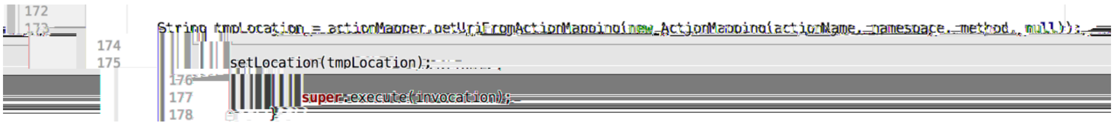
```
ull && prefix.startsWith(ns) && (prefix.length() == ns.length() || prefix.charAt(ns.length()) == '/')) {
 if (ns.length() > namespace.length()) {
     namespace = ns;
 }

 ("/".equals(ns)) {
 rootAvailable = true;


 uri.substring(namespace.length() + 1);
```



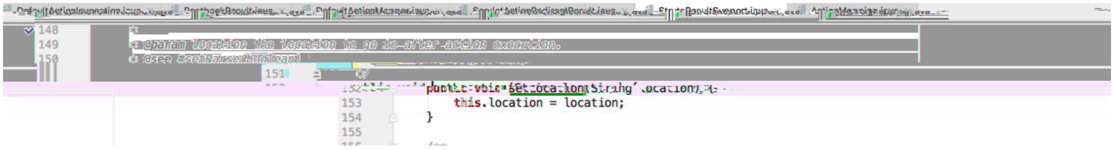
```
     */
    public void execute(ActionInvocation invocation) throws
```



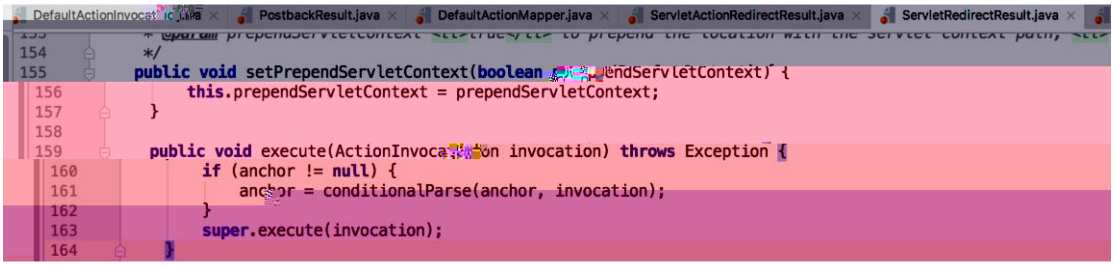
```
     */
        this.name = name;
        this.namespace = namespace;
        this.params = params;
    }
```



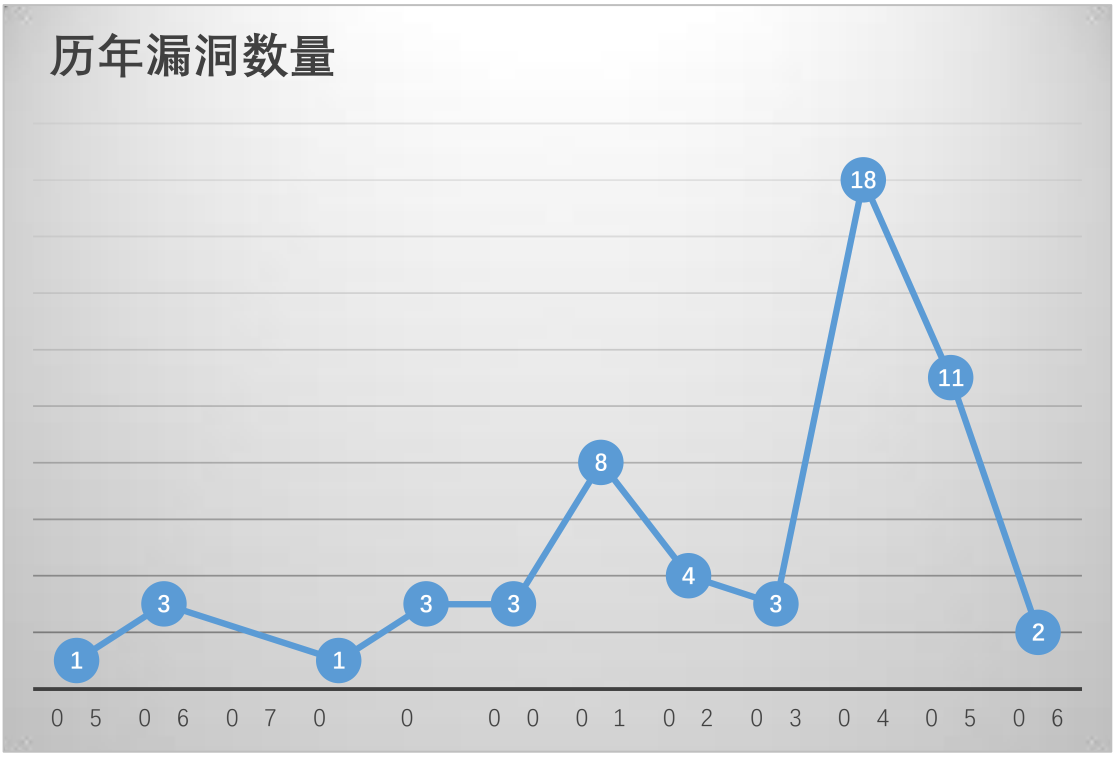
```
        String tmpLocation = actionMapper.getUriFromActionMapping(new ActionMapping(actionName, namespace, method, null));

        setLocation(tmpLocation);

        super.execute(invocation);
    }
```



```
     *
     * @param location the location to go to after action execution.
     * @see #setParse(boolean)
     */
    public void setLocation(String location) {
        this.location = location;
    }
```

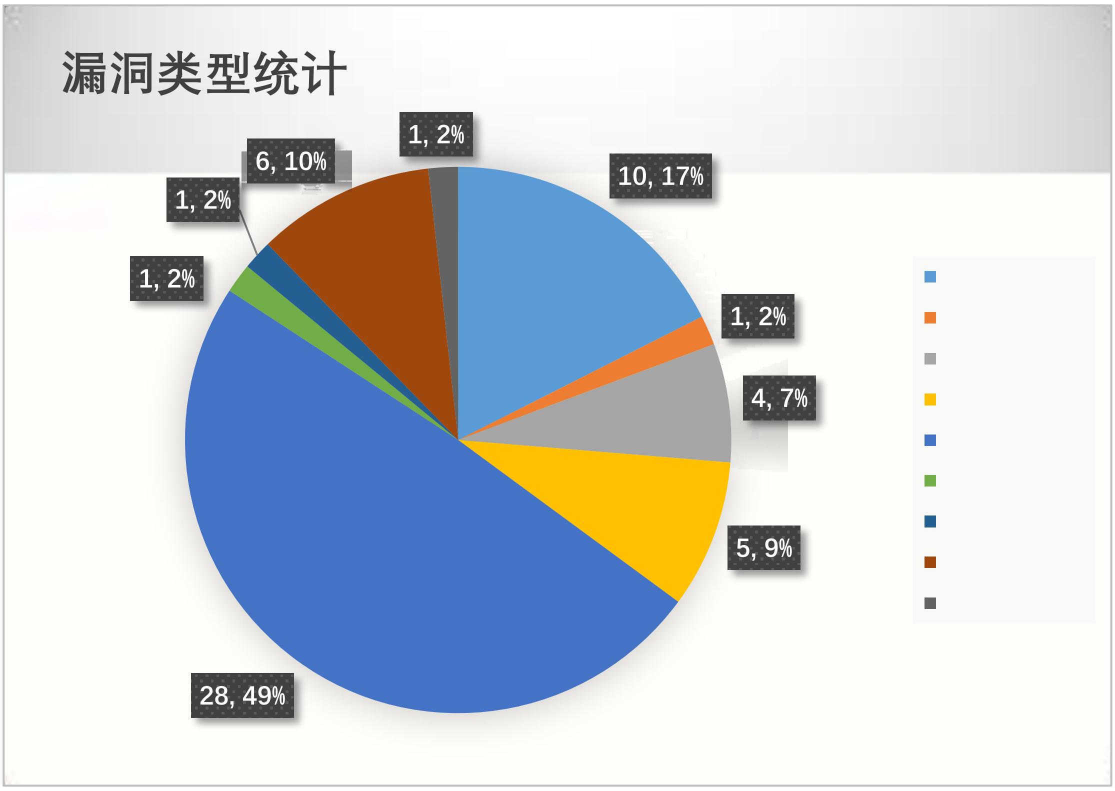


```
     */
    public void setPrependServletContext(boolean prependServletContext) {
        this.prependServletContext = prependServletContext;
    }

    public void execute(ActionInvocation invocation) throws Exception {
        if (anchor != null) {
            anchor = conditionalParse(anchor, invocation);
        }
        super.execute(invocation);
    }
```


历年漏洞数量
1
3
1
3
3
8
4
3
18
11
2
0 5
0 6
0 7
0
0
0 0
0 1
0 2
0 3
0 4
0 5
0 6

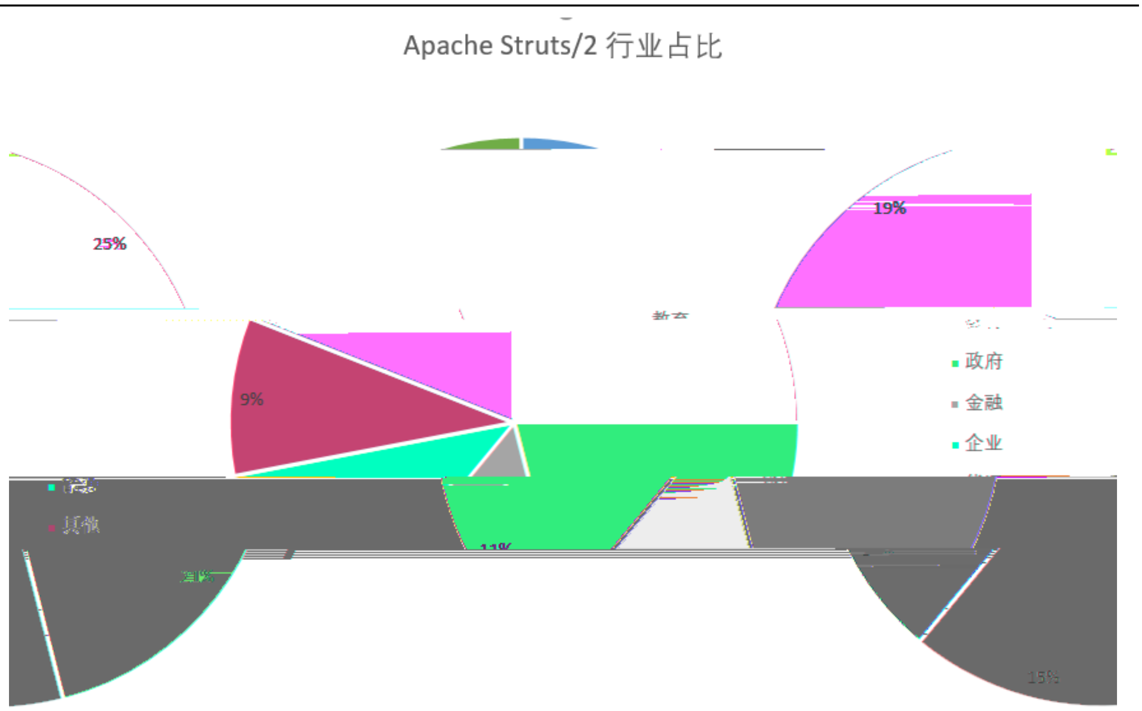
□

□


漏洞类型统计
1, 2%
6, 10%
10, 17%
1, 2%
1, 2%
1, 2%
4, 7%
5, 9%
28, 49%

□

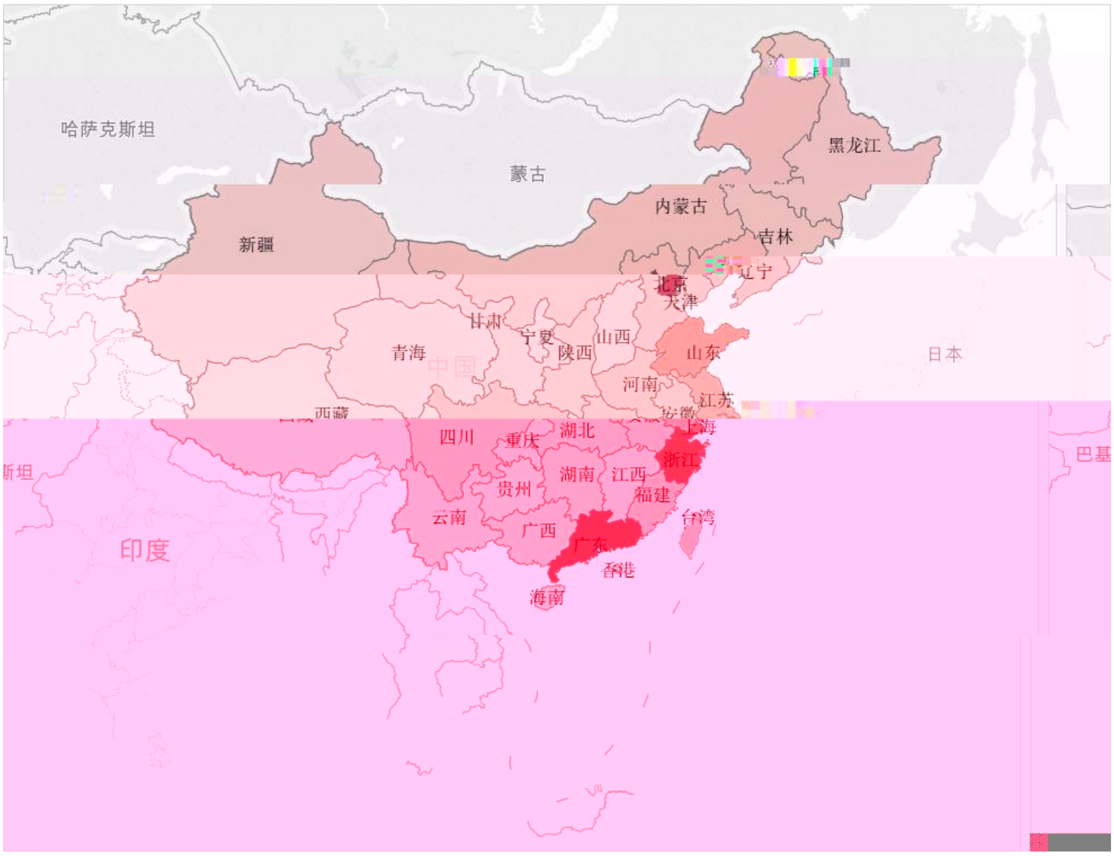
□


Apache Struts/2 行业占比
25%
19%
9%
11%
15%
政府
金融
企业
其他

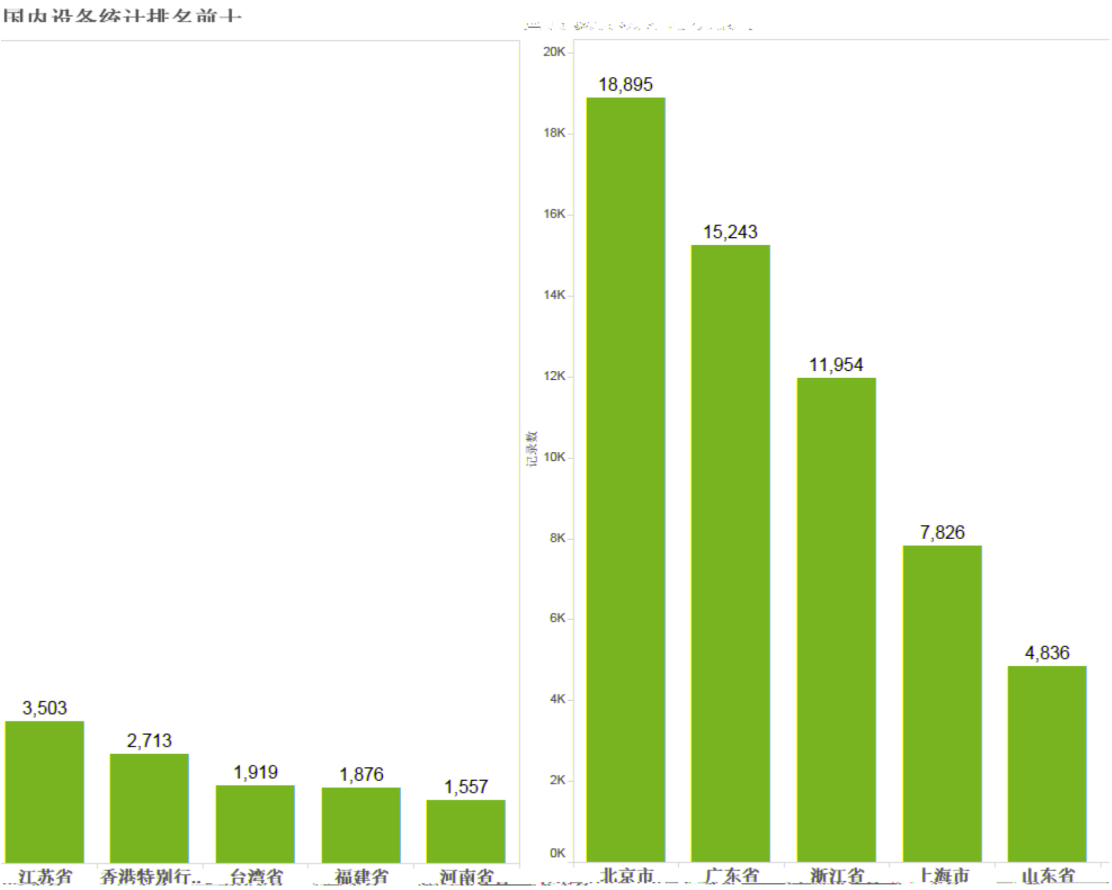
□

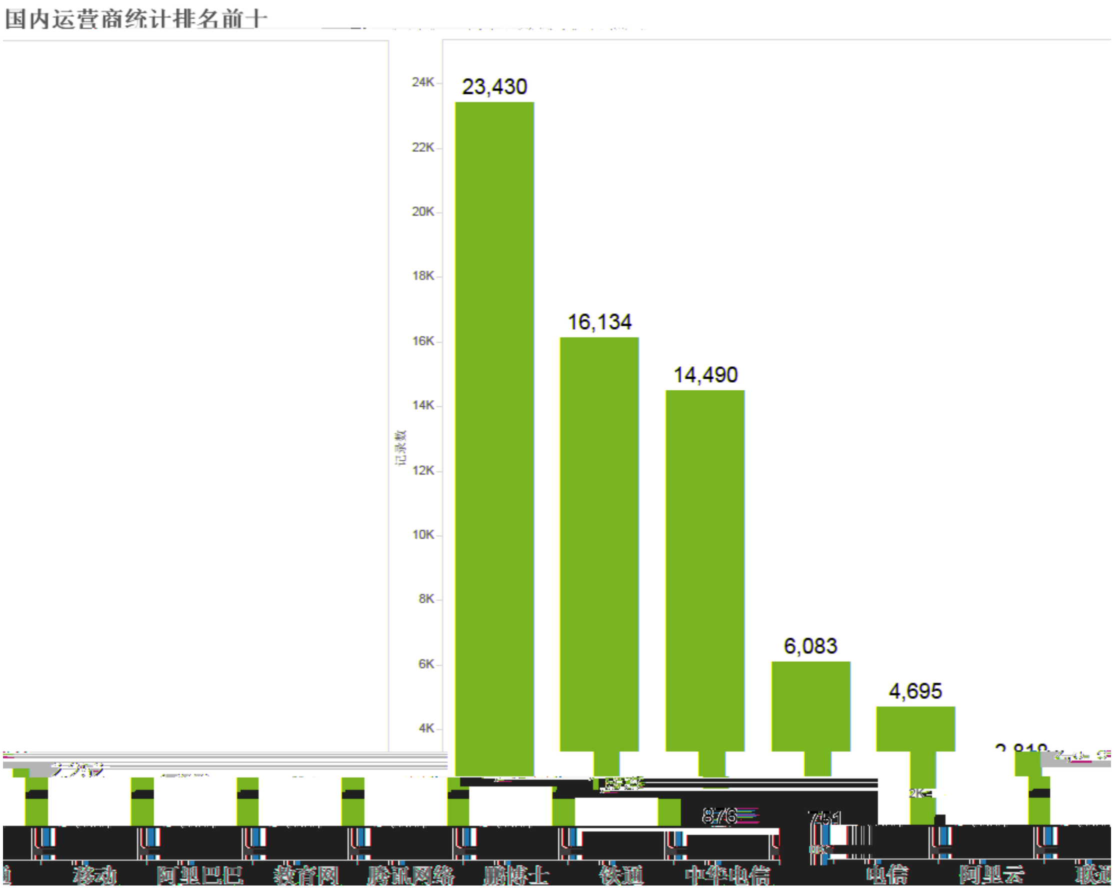
□ □

国内分布


哈萨克斯坦
蒙古
黑龙江
内蒙古
吉林
新疆
北京
天津
辽宁
甘肃
宁夏
山西
青海
中国
陕西
山东
河南
日本
江苏
西藏
安徽
上海
四川
重庆
湖北
浙江
巴基
斯坦
湖南
江西
贵州
福建
云南
台湾
印度
广西
广东
香港
海南

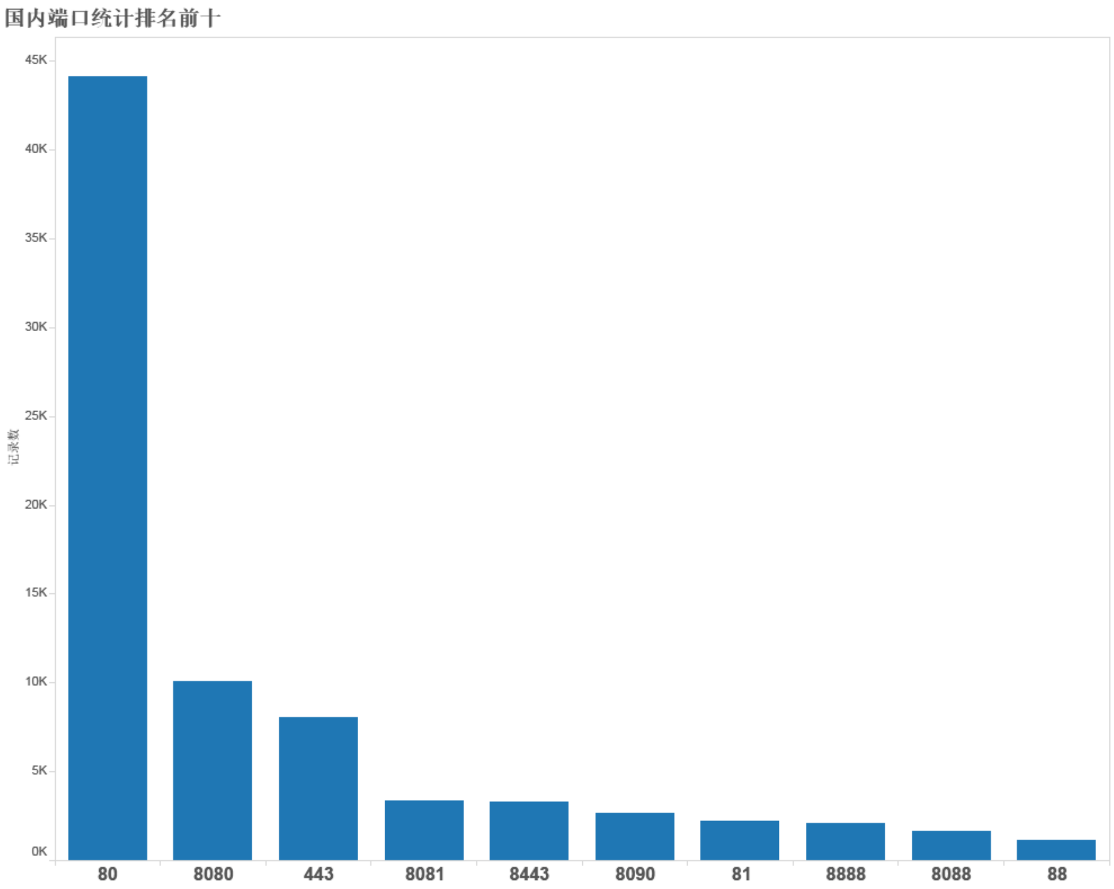
□ □


国内设备统计排名前十
记录数
20K
18K
16K
14K
12K
10K
8K
6K
4K
2K
0K
18,895
15,243
11,954
7,826
4,836
3,503
2,713
1,919
1,876
1,557
江苏省
香港特别行..
台湾省
福建省
河南省
北京市
广东省
浙江省
上海市
山东省


国内运营商统计排名前十
记录数
24K
22K
20K
18K
16K
14K
12K
10K
8K
6K
4K
2K
23,430
16,134
14,490
6,083
4,695
移动
阿里巴巴
教育网
腾讯网络
鹏博士
铁通
中华电信
电信
阿里云

□

□


国内端口统计排名前十
记录数
45K
40K
35K
30K
25K
20K
15K
10K
5K
0K
80
8080
443
8081
8443
8090
81
8888
8088
88

□