软件安全开发之痛

韩建

主要内容

我们面临的安全挑战

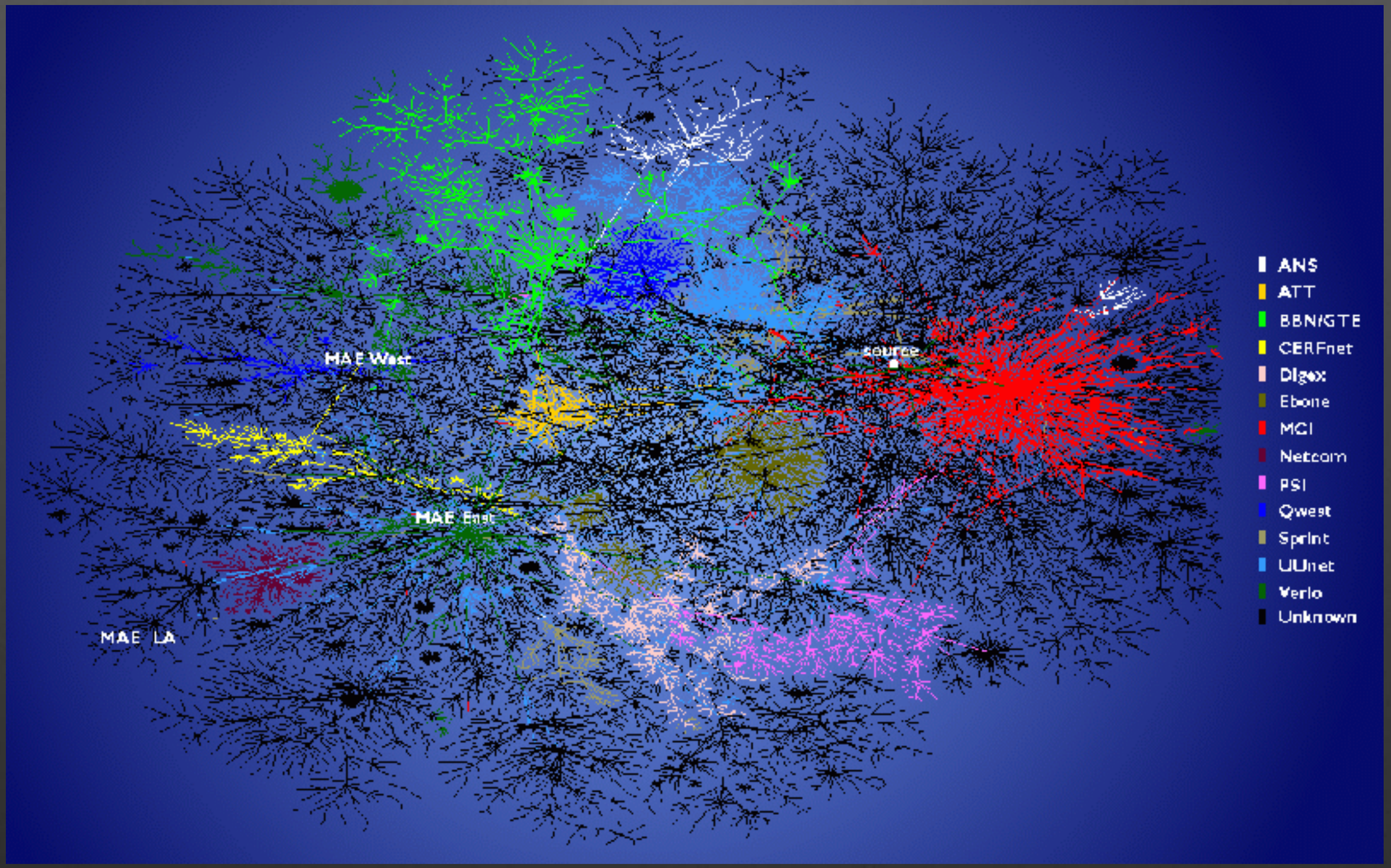
开发过程中的安全问题

如何构建安全开发体系

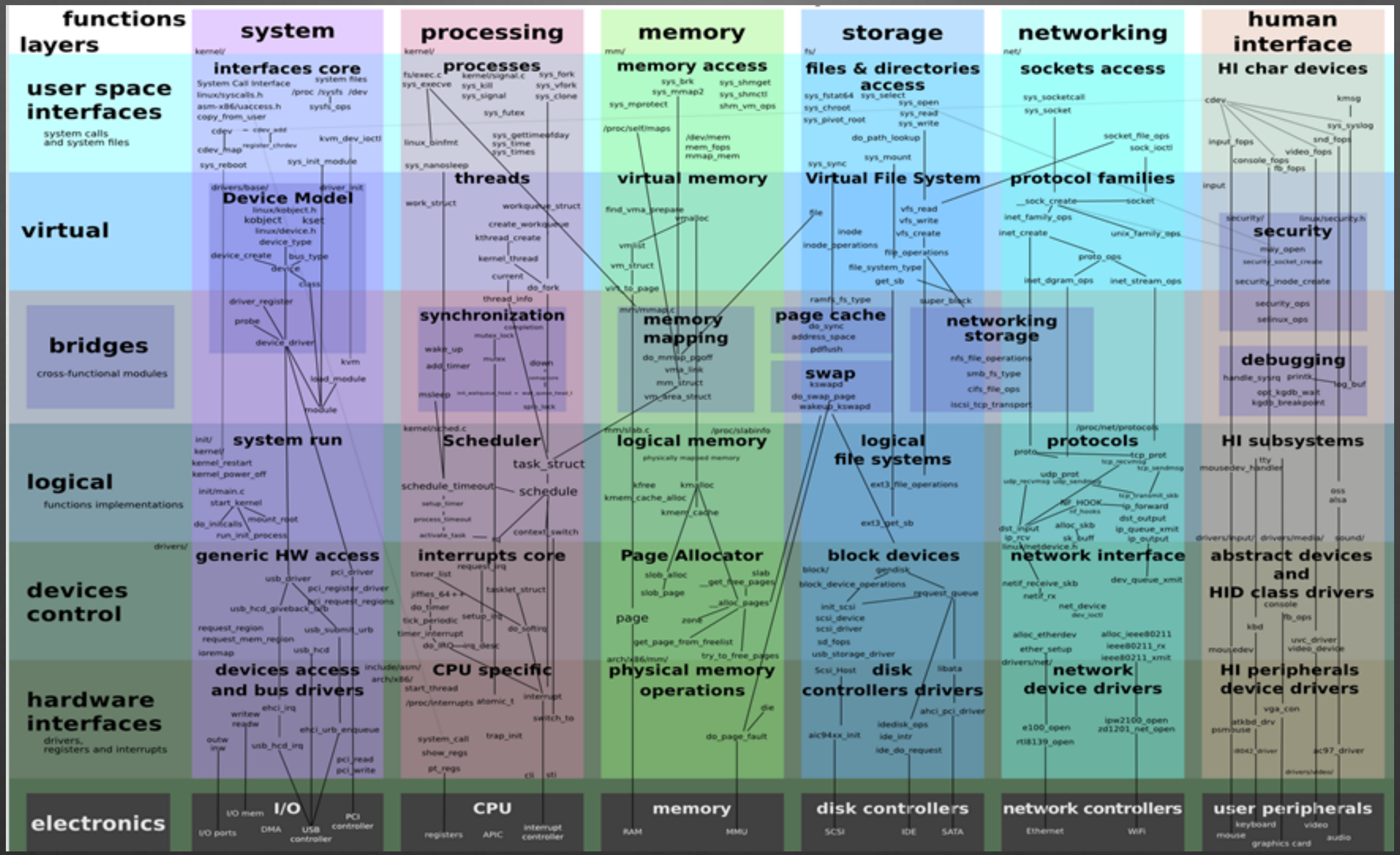
开源代码安全现状及实例分析

我们面临的安全挑战

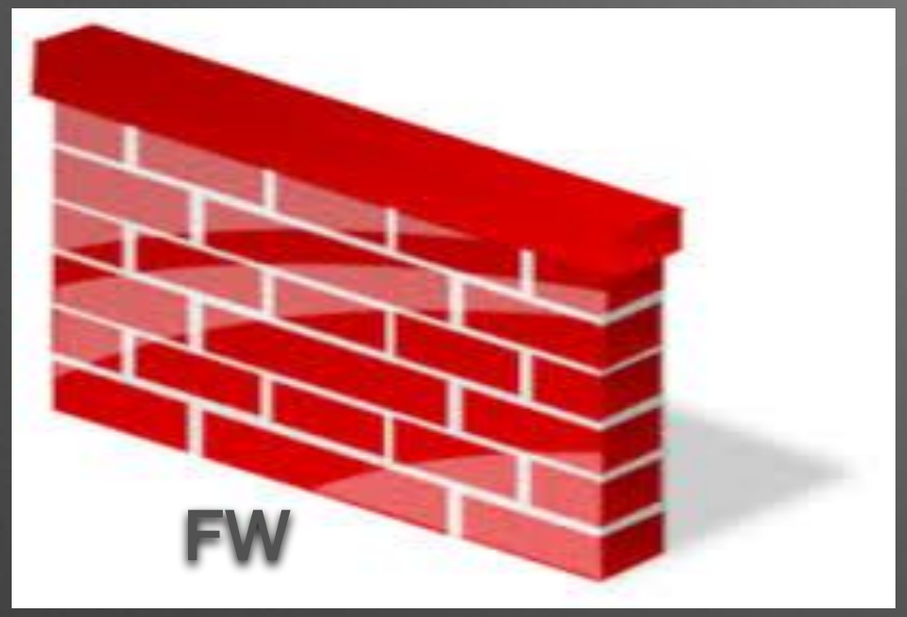
受攻击面增大


ANS
ATT
BBN/GTE
CERFnet
Digex
Ebone
MCI
Netcom
PSI
Qwest
Sprint
UUnet
Verio
Unknown
source
MAE West
MAE East
MAE LA

复杂程度提高，安全缺陷增多



我们使出洪荒之力，依然没有安全感



开发过程中的安全问题

需求分析阶段

- 关注功能，忽略安全需求
- 需求评审缺少安全的介入
- 安全目标
- ……

设计阶段

- 认证问题
- 授权问题
- 依赖客户端检测
- 关键数据保护措施
- 会话管理
- 第三方组件安全问题
- ……

设计缺陷导致的漏洞：明文存储密码

设计缺陷导致的漏洞：CSRF

订单信息

宝贝	状态	单价(元)	数量	优惠	商品总价(元)	运费(元)
[redacted]的充值卡(购买之前跟卖家联系)	-	176.00	1	无优惠	176.00	卖家包邮：0.00

实付款： 176.00 元

订单编号： 4064023[redacted]

支付宝交易号： 201006287[redacted]

卖家昵称： [redacted] 和我联系

收货信息： [redacted]

成交时间： 2010-06-28 14:47:21

- **请收到货后，再确认收货！否则您可能钱货两空！**
- 如果您想申请退款，请点击这里

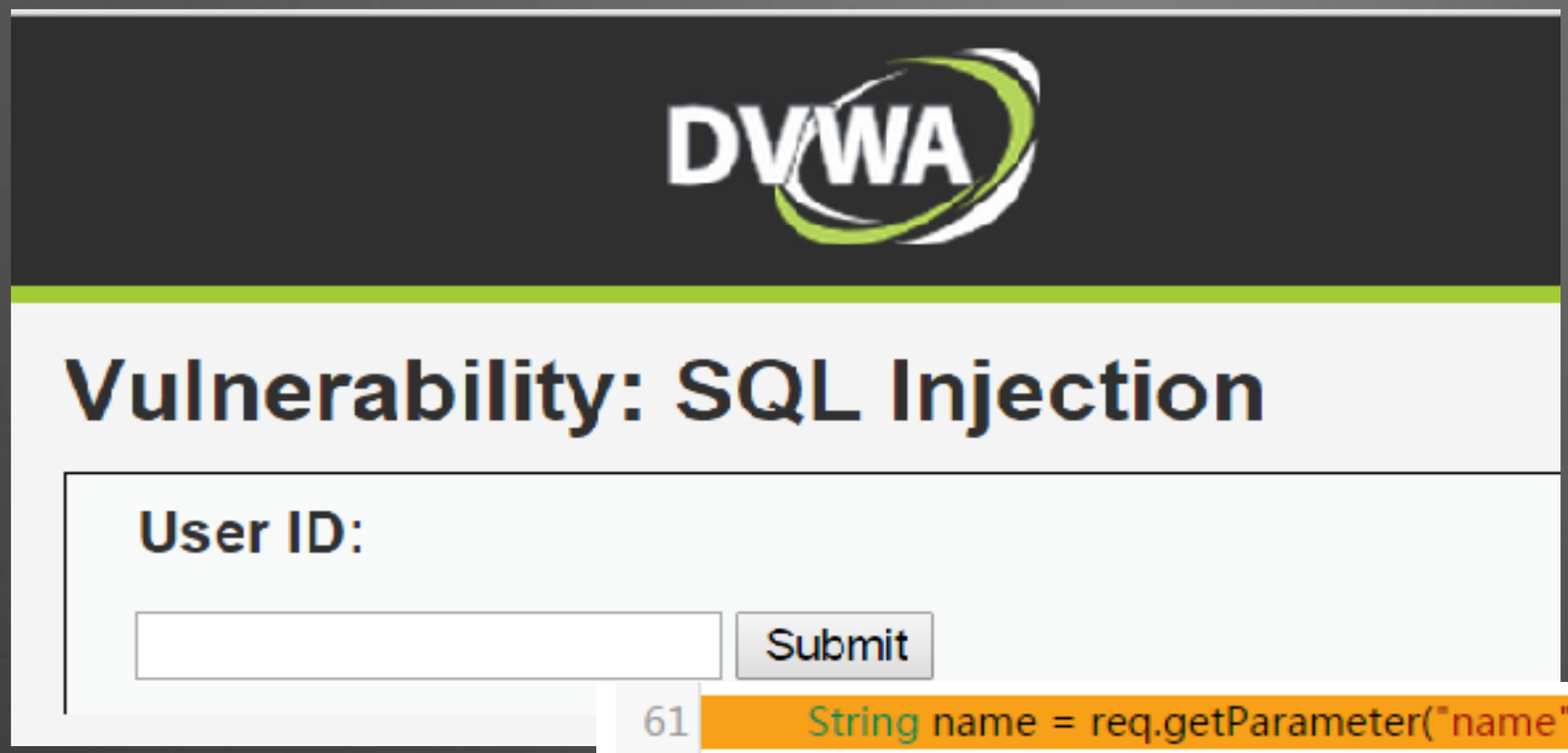
请输入支付宝账户支付密码*：

确定 找回支付密码

开发阶段

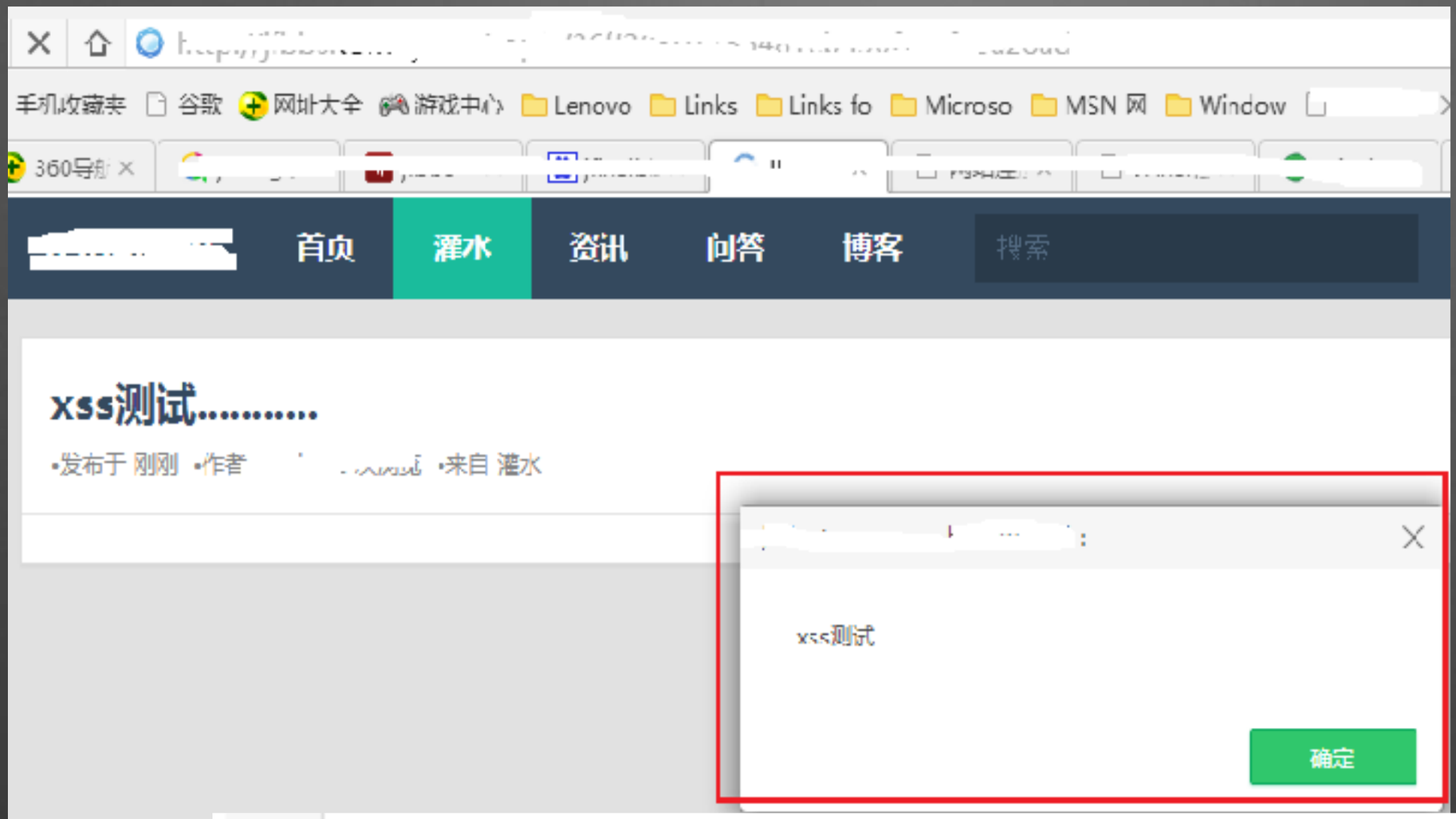
- 安全编码规范
- 持续性检测
- 差距分析
- 修复跟踪
- ……

并发出来的漏洞：SQL注入




```
61      String name = req.getParameter("name");
62      if (name == null)
63          name = "hi";
64
65      Connection conn = getConnection();
66      try {
67          PreparedStatement stmt = conn
68                  .prepareStatement("select val from mytab where name = '"
69                          + name + "'");
```

并发出来的漏洞：XSS




```
53      String username = request.getParameter("username");
54
55      if ((username != null) && (username.length() > 0)) {
56          out.println("<h4>Hello, ");
57          out.println(username); /*
```

开发出来的漏洞： Do not synchronize on objects that may be reused

```
//this bug was found  in jetty-6.1.3 BoundedThreadPool
private final String lock = "LOCK";
public void doSomething() {
        synchronized (lock) {
            // . . .
        }
    }
```

More……

- 命令注入
- 目录遍历
- 资源释放
- 空指针
- ……

测试阶段

- 更多关注功能、性能、稳定性测试
- 缺少源代码缺陷检测
- 缺少黑盒测试
- 缺少业务安全性测试评审
- 缺少第三方组件安全测试评审

部署阶段

- 缺少配置安全评审
- 系统的额外服务、端口
- 服务器默认用户、默认示例
- 权限过高的默认账户
- ……

部署不当导致的漏洞：默认口令

Back to search

Axis2 Default Administrator Password Vulnerability

Severity	CVSS	Published	Added	Modified
10	(AV:N/AC:L/Au:N/C:C/I:C/A:C)	October 12, 2010	October 13, 2010	December 03, 2013

Available Exploits

Apache Axis2 Brute Force Utility
Axis2 / SAP BusinessObjects Authenticated Code Execution (via SOAP)

admin axis2

Description

The Axis2 administrator 'admin' has a password that is set to the default value of 'axis2'. As a result, anyone with access to the Axis2 port can trivially gain full access to the machine via arbitrary remote code execution. This requires the attacker to upload a malicious webservice and for the instance of Tomcat to be restarted.

This vulnerability affects default Axis2 installations as well as SAP BusinessObjects via the web service module (known as dswsbobje) and other products that are based on Axis2

部署不当导致的漏洞： App AllowBackup

两分钟窃取身边女神微博帐号？详解Android App AllowBackup配置带来的风险

APP漏洞挖掘 BY DROIDSEC

笔者在使用自己编写的Drozer模块对国内流行的安卓手机应用进行自动化扫描后发现有大量涉及用户财产和隐私的流行安卓应用存在Android AllowBackup漏洞，已测试成功受到漏洞影响的应用包括：新浪微博，百度云网盘，美团，大众点评，去哪儿等等。

[0x00] 漏洞案例

先来看一个情景案例，某IT男一直暗恋部门某女神，一天女神手机太卡了找IT男帮忙清理手机空间，IT男高兴地答应女神两分钟搞定，屁颠屁颠的跑到自己电脑旁边连上手机，女神在一边呆呆的看着IT男敲了几行代码然后在手机上点了几下，最后果然两分钟不到就搞定了，在女神表看离开后，IT男露出了WS的笑容。

运维阶段

- 软件安全水平完好无损?
- SDK出现安全问题
- 软件本身出现安全问题
- 第三方组件产生安全漏洞

如何构建安全开发体系

软件安全开发进化史

- 2000年，NASA的一个系统安全评估项目开启了软件安全开发方法的序幕，其核心是在软件开发的每个关键点嵌入安全要素。
- AEGIS、微软的SDL、敏捷SDL、McGraw的BSI、OWASP的CLASP等软件安全开发模型先后出现。
- 在软件安全开发模型基础上，出现了ISO27034、BSIMM、SAMM等软件安全开发标准。

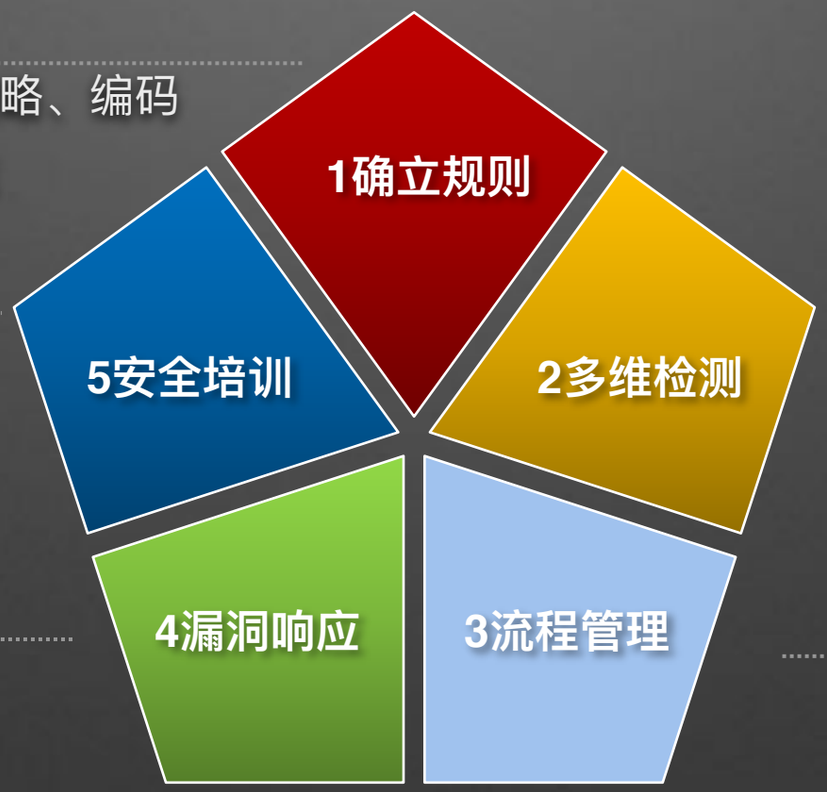
软件安全开发体系构建

基于企业目前的开发管理现状，制定合适的目标，不断演进、完善。

安全目标、策略、编码规范

定期开展安全培训

建立漏洞响应体系



合规检测、溯源检测、缺陷和漏洞检测

融入流程，管理流程

1确立规则

企业现状分析

- 现有开发流程
- 软件特点及历史安全现状

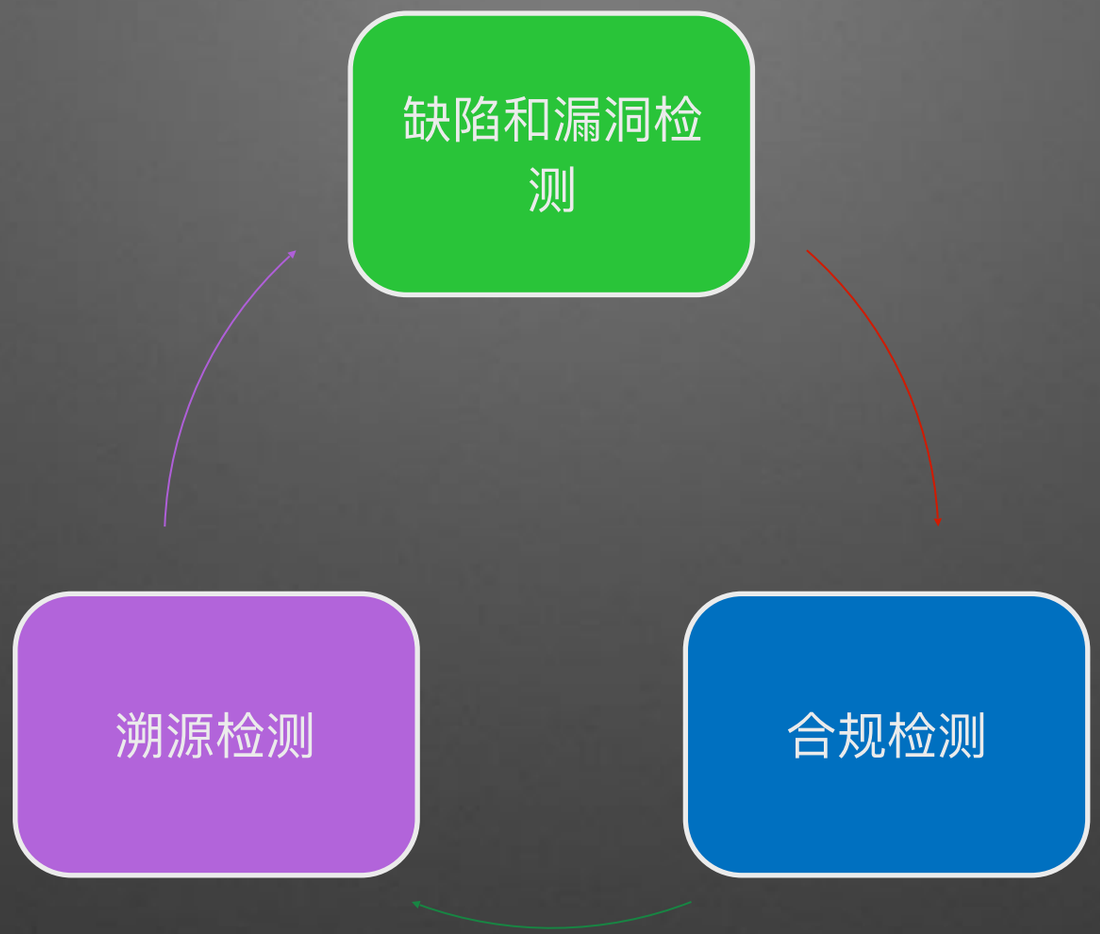
检测规则

- 根据制定的规范定制检测规则集

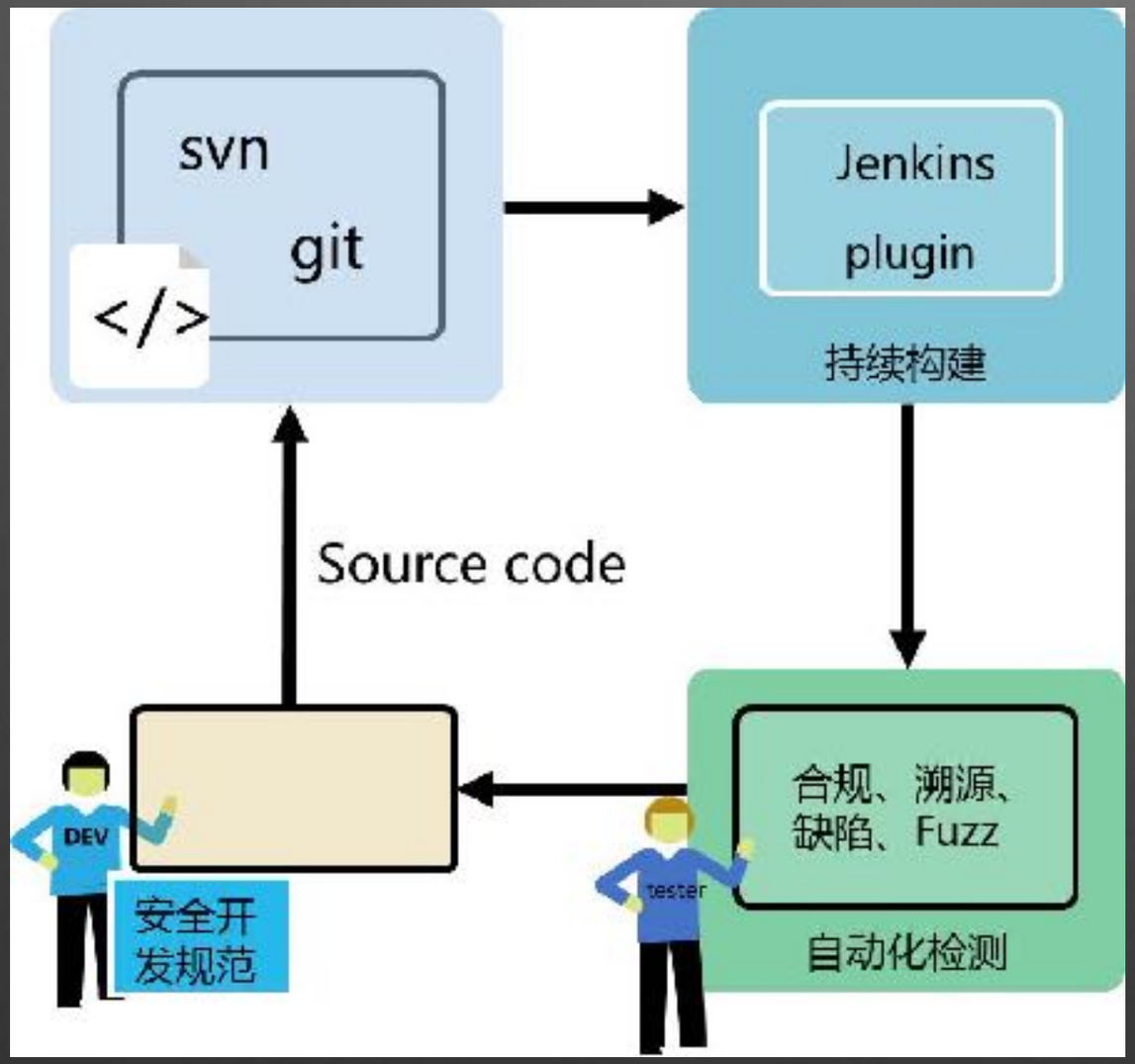
安全开发与测试规范制定

- 结合企业需求确定目标
- 制定企业安全规范

2多维检测



3流程管理



- 安全编码
- 持续构建
- 自动化检测
- 漏洞修复

4漏洞响应


上报
主动监测和接受安全问题和漏洞，启动应急处理流程，同时对漏洞上报者进行确认
验证
安全实验室和相关产品对安全问题进行验证、确认，同时评估风险等级
解决
制定安全漏洞的缓解措施，相关产品针对漏洞进行修复开发，同时制定安全预警策略
披露
在安全问题有规避措施和解决补丁的情况下，官方根据策略披露漏洞信息（范围及细节程度）
反馈
收集来自相关外部客户的建议，安全实验室组织产品进行相关的改进

5安全培训

- 定期开展安全培训
  - 宣贯政策、提高意识、培养技能
  - 面向软件开发中的各种角色（架构师、产品经理、开发工程师、测试工程师等）

	基础培训内容		专业培训内容			
安全架构师	软件安全基本技术详解	攻防技术	架构风险分析与威胁建模	编程安全（C/C++/C#/VB.NET/Java/PHP）	安全代码审核和静态分析
软件开发工程师			软件安全需求分析与设计	基于风险控制的安全测试策略	
测试和质量工程师					
测试和质量经理	软件安全基础和原理				
产品经理					

开源代码安全现状及实例分析

开源代码安全现状

- 2010年，Gartner采访了来自11个国家的547位公司负责人，在被调查的公司当中超过一半采用了开源软件作为其IT战略的组成部分。
- 2012年，Aspect Security和Sonatype公开的一份调查报告显示，最受欢迎的31个开源项目中，其不安全的版本被下载了超过4,600万次。
- 2014 OpenSSL年曝光重大安全漏洞Heartbleed。攻击者通过构造异常的数据包进行攻击，获取用户敏感信息。
- 2014年和2015年ElasticSearch分别爆出远程任意命令执行漏洞。攻击者可利用远程任意命令执行漏洞获取主机最高权限。
- 近年来Struts2频繁爆发安全漏洞。影响国内电商、银行、运营商等诸多大型网站和为数众多的政府网站。

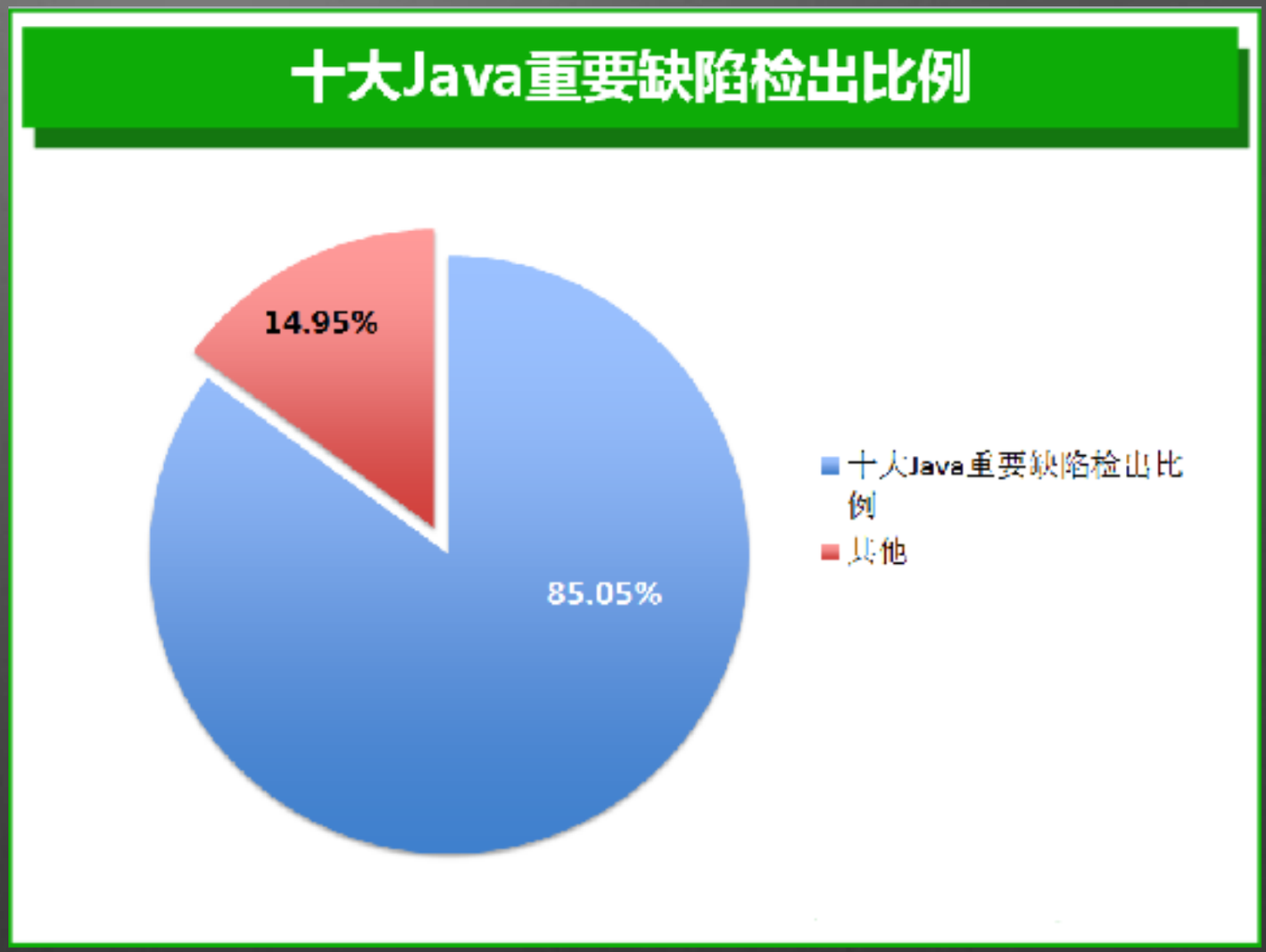
开源项目检测计划

开源项目检测计划（www.codesafe.cn）是由360代码卫士团队发起，针对开源项目进行的一项公益安全检测计划，旨在让广大开发者关注和了解开源代码安全问题，提高软件安全开发意识和技能。

注：开源项目检测计划使用的检测工具是360自主研发的源代码检测引擎“代码卫士”。

开源项目检测计划—十大Java严重缺陷统计

十大 Java 重要缺陷	缺陷总数（个）
SQL 注入	2491
跨站脚本	5011
路径遍历	17852
密码管理	21273
HTTP 消息头注入	3106
命令注入	765
资源注入	12555
资源未释放	75150
系统信息泄露	113429
跨站请求伪造	10157
总计	262089

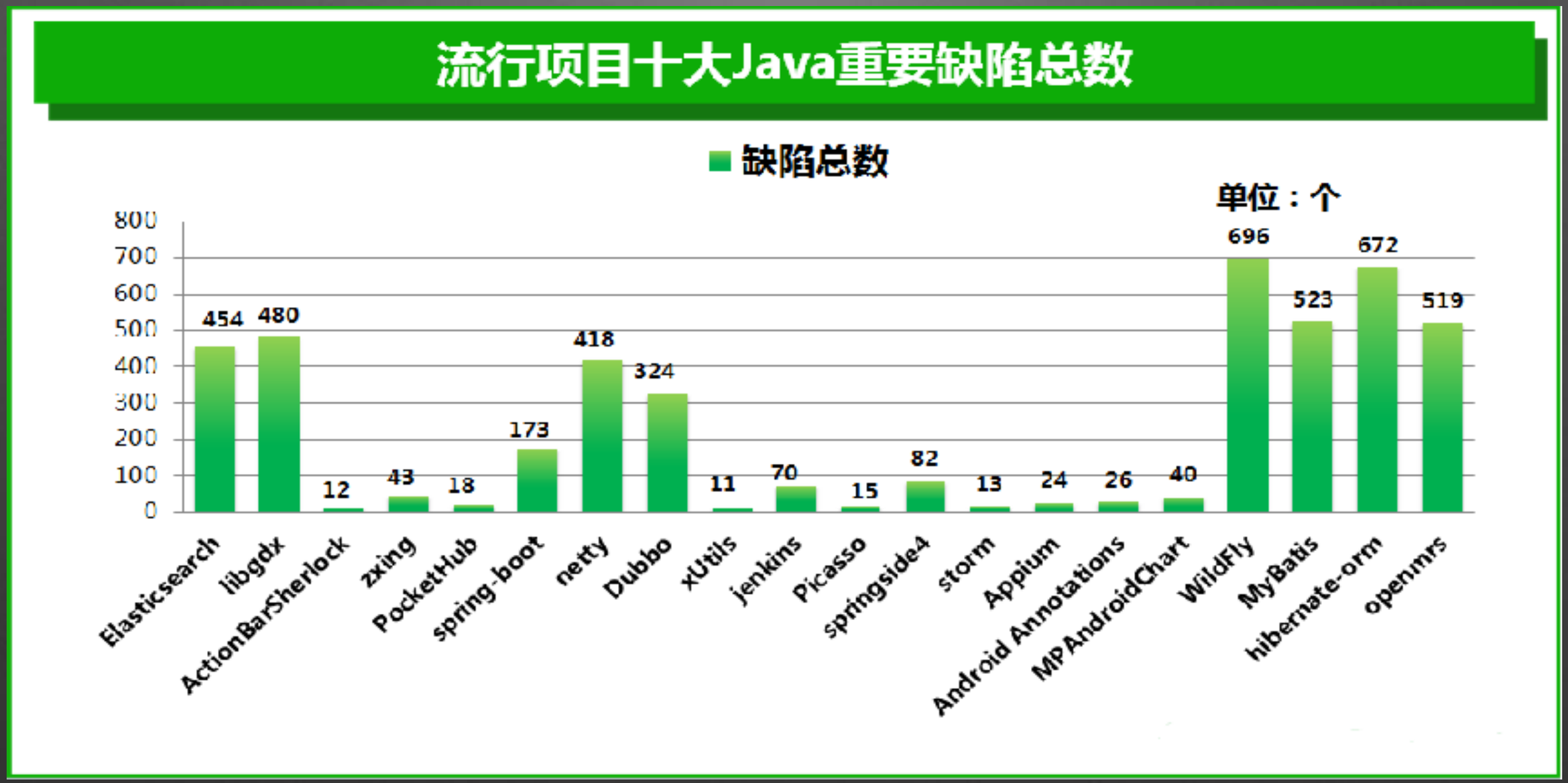


开源项目检测计划—20个流行项目

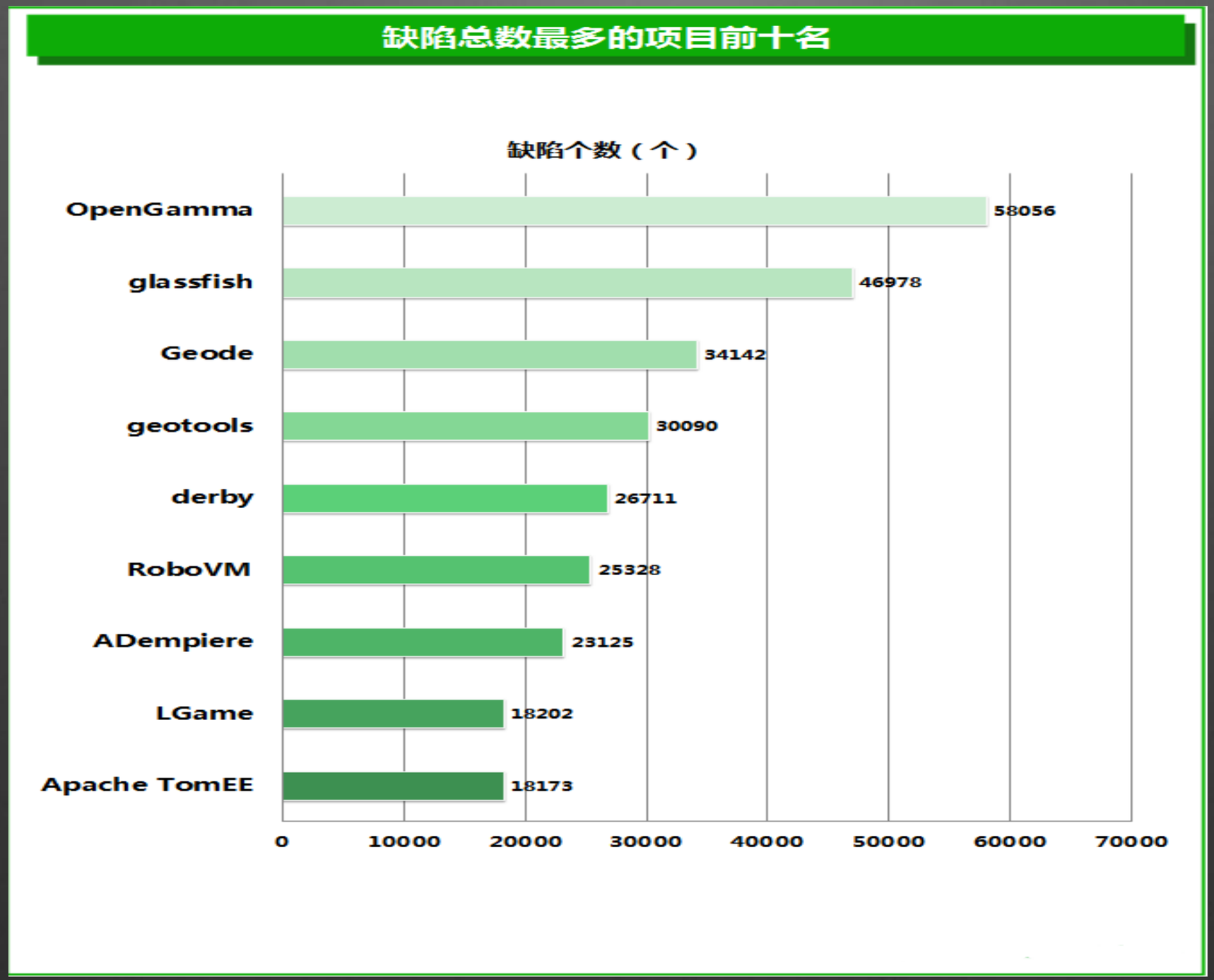
序号	项目名称	Fork	Star	Watch	版本号	缺陷总数
1	Elasticsearch	4132	12758	1303	1.5.1	1473
2	libgdx	4100	7008	936	1.5.0	1666
3	ActionBarSherlock	4096	7029	853	4.4.0	156
4	zxing	3854	6116	813	3.1.0	331
5	PocketHub	3441	6950	1054	1.9.0	135
6	spring-boot	3020	2903	501	1.3.0.M1	550
7	netty	2655	5223	813	4.0.24.Final	1237
8	Dubbo	2459	2321	758	2.5.3	903
9	xUtils	2288	3046	546	2.6.14	184
10	jenkins	2119	4532	577	1.616	364
11	Picasso	2042	7278	682	2.5.2	257
12	springside4	2038	2757	747	4.2.3.GA	373
13	storm	1748	8686	1174	0.9.0.1	297
14	Appium	1744	2600	464	1.4.10	111
15	Android Annotations	1739	6024	617	3.3.2	218
16	MPAndroidChart	1731	5194	395	2.0.8	52
17	WildFly	1432	1439	172	10.0.0.Beta1	2824
18	MyBatis	1401	1657	409	3.3.0	570
19	hibernate-orm	1172	1374	201	5.0.0.CR4	2002
20	openmrs	1161	234	82	1.9.1	2147

开源项目检测计划—20个流行项目缺陷总数统计

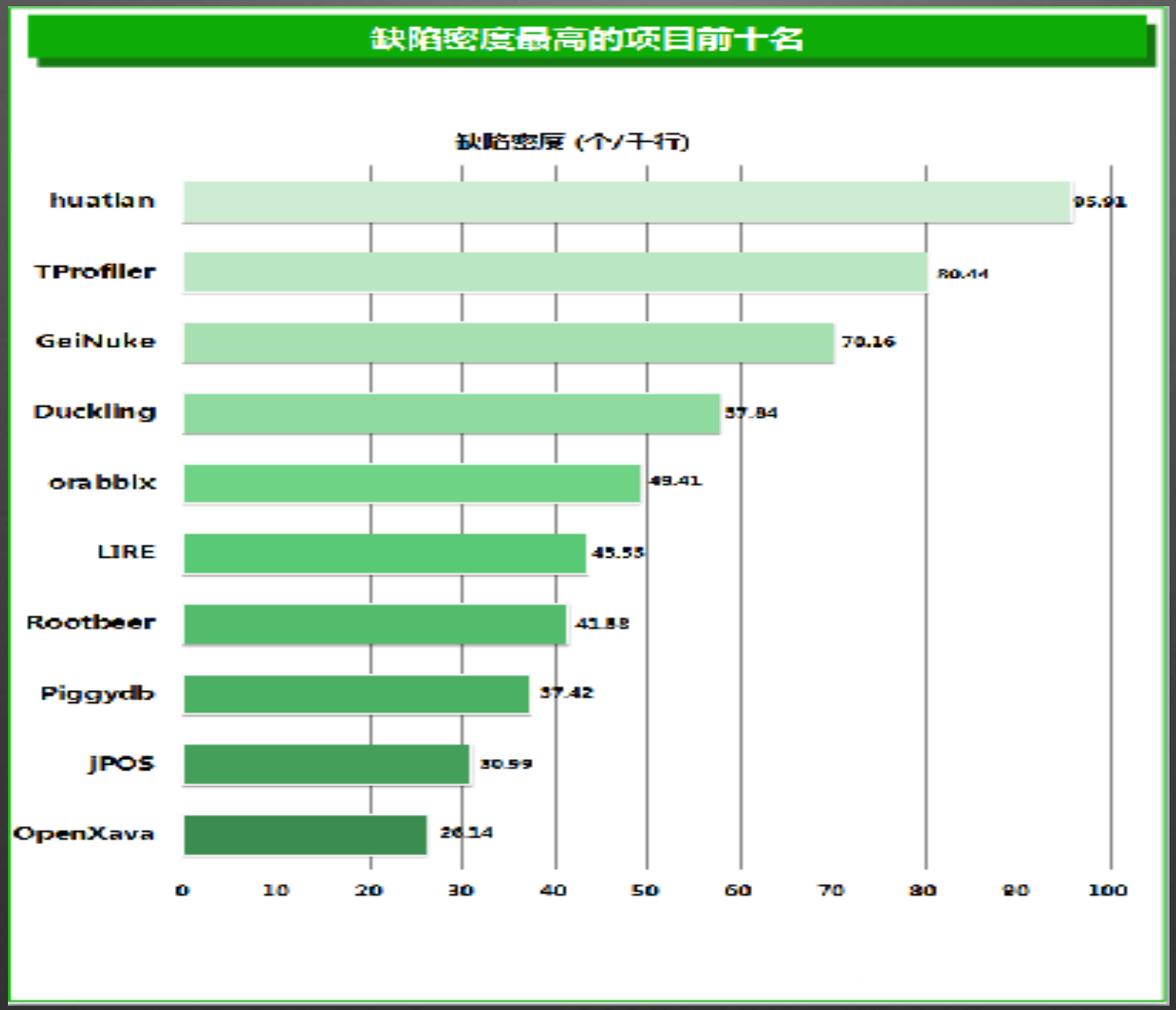
开源项目检测计划—20个流行项目十大Java重要缺陷数量统计



开源项目检测计划—缺陷数量Top 10项目



开源项目检测计划—缺陷密度Top 10项目



实例分析1—某开源论坛项目XSS漏洞

```
14 <form id="create_form" action="${baseUrl!}/topic/save" method="post">
15     <select name="sid" id="sid" class="form-control" style="width: 20%; margin-bottom: 5px;">
16         <#list sections as section>
17             <option value="${section.id}">${section.name}</option>
18         </#list>
19     </select>
20     <input type="text" placeholder="标题字数10字以上" id="title" name="title" class="form-control"
21     <input type="text" placeholder="原文地址(原创可不写)" id="original_url" name="original_url" cla
22     <div id="content" style="margin-bottom: 5px;"><textarea name="content"></textarea></div>
23     <input type="button" onclick="submitForm()" value="发布" class="btn btn-primary">
24 </form>
```

```
108         String content = getPara("content");
109         String original_url = getPara("original_url");
110         Topic topic = new Topic();
111         topic.set("id", StrUtil.getUUID())
112                 .set("in_time", new Date())
113                 .set("s_id", sid)
114                 .set("title", title)
115                 .set("content", content)
```

```
109     public String getPara(String name) {
110         return request.getParameter(name);
111     }
```

实例分析1—某开源论坛项目XSS漏洞

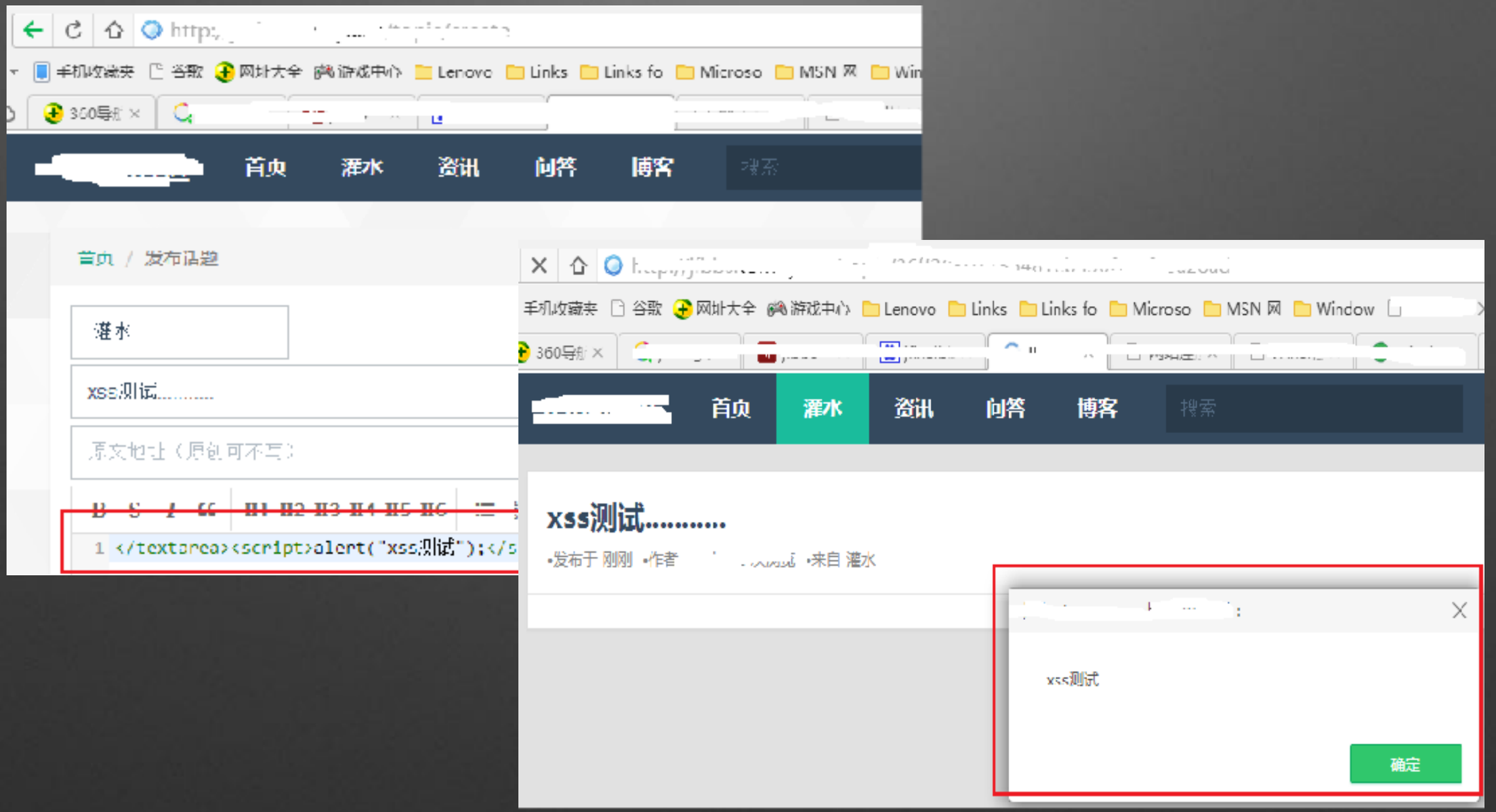
```
24          Topic topic = Topic.me.findByIdWithUser(id);
25          if (topic != null) {
26              List<Reply> replies = Reply.me.findByTid(id);
27              setAttr("topic", topic);
```

```
74      public Controller setAttr(String name, Object value) {
75          request.setAttribute(name, value);
76          return this;
77      }
```

```
62      <div class="panel-body" style="border-top: 1px #EBEBEB solid; padding-top: 10px">
63          <div id="topic_content">
64              <textarea id="_topic_content" style="display: none;">${topic.content!}</textarea>
65          </div>
66          <#if topic.reposted?? && topic.reposted == 1>
```

XSS

实例分析1—某开源论坛项目XSS漏洞



实例分析2—某开源流媒体解析工具包类型混淆漏洞

```
12909: HandleInvoke(RTMP *r, const char *body, unsigned int nBodySize)
12910: {
12911:   AMFObject obj;
12912:   AVal method;
12913:   double txn;
12914:   int ret = 0, nRes;
12915:   if (body[0] != 0x02)       /* make sure it is a string method name we start with */
12916:     {
12917:       RTMP_Log(RTMP_LOGWARNING, "%s, Sanity failed. no string method in invoke packet",
12918:       __FUNCTION__);
12919:       return 0;
12920:     }
12921:
12922:   nRes = AMF_Decode(&obj, body, nBodySize, FALSE);
```

```
01175:
01176:       nRes = AMFProp_Decode(&prop, pBuffer, nSize, bDecodeName);
01177:       if (nRes == -1)
01178:     bError = TRUE;
```

```
00653:   switch (prop->p_type)
00654:     {
00655:     case AMF_NUMBER:
00656:       if (nSize < 8)
00657:     return -1;
00658:       prop->p_vu.p_number = AMF_DecodeNumber(pBuffer);
00659:       nSize -= 8;
00660:       break;
00661:     case AMF_BOOLEAN:
00662:       if (nSize < 1)
00663:     return -1;
00664:       prop->p_vu.p_number = (double)AMF_DecodeBoolean(pBuffer);
00665:       nSize--;
00666:       break;
00667:     case AMF_STRING:
00668:       {
00669:     unsigned short nStringSize = AMF_DecodeInt16(pBuffer);
00670:
00671:     if (nSize < (long)nStringSize + 2)
00672:       return -1;
00673:     AMF_DecodeString(pBuffer, &prop->p_vu.p_aval);
00674:     nSize -= (2 + nStringSize);
00675:     break;
00676:       }
00677:     case AMF_OBJECT:
00678:       {
00679:     int nRes = AMF_Decode(&prop->p_vu.p_object, pBuffer, nSize, TRUE);
00680:     if (nRes == -1)
```

实例分析2—某开源流媒体解析工具包类型混淆漏洞

```
HandleInvoke(RTMP *r, const char *body, unsigned int nBodySize)
{
  AMFObject obj;
  AVal method;
  double txn;
  int ret = 0, nRes;
  if (body[0] != 0x02)        /* make sure it is a string method name we start with */
    {
      RTMP_Log(RTMP_LOGWARNING, "%s, Sanity failed. no string method in invoke packet",
      __FUNCTION__);
      return 1;
    }

  nRes = AMF_Decode(&obj, body, nBodySize, FALSE);
```

```
  else if (AVMATCH(&method, &av_onStatus))
    {
      AMFObject obj2;
      AVal code, level;
      AMFProp_GetObject(AMF_GetProp(&obj, NULL, 3), &obj2);
      AMFProp_GetString(AMF_GetProp(&obj2, &av_code, -1), &code);
      AMFProp_GetString(AMF_GetProp(&obj2, &av_level, -1), &level);
```

```
AMFProp_GetObject(AMFObjectProperty *prop, AMFObject *obj)
{
  *obj = prop->p_vu.p_object;
}
```

```
AMF_GetProp(AMFObject *obj, const AVal *name, int nTheNo)
{
  if (nIndex >= 0)
    {
      if (nIndex < obj->o_num)
        return &obj->o_props[nIndex];
    }
  else
    {
      int n;
      for (n = 0; n < obj->o_num; n++)
        {
          if (AVMATCH(&obj->o_props[n].p_
            return &obj->o_props[n];
        }
    }

  return (AMFObjectProperty *)&AMFProp_Invalid;
} /* end AMF_GetProp */
```

Crash

实例分析2—某开源流媒体解析工具包类型混淆漏洞

Windows平台下SMPlayer验证结果

Ubuntu自带播放器SMPlayer验证结果

Thanks!