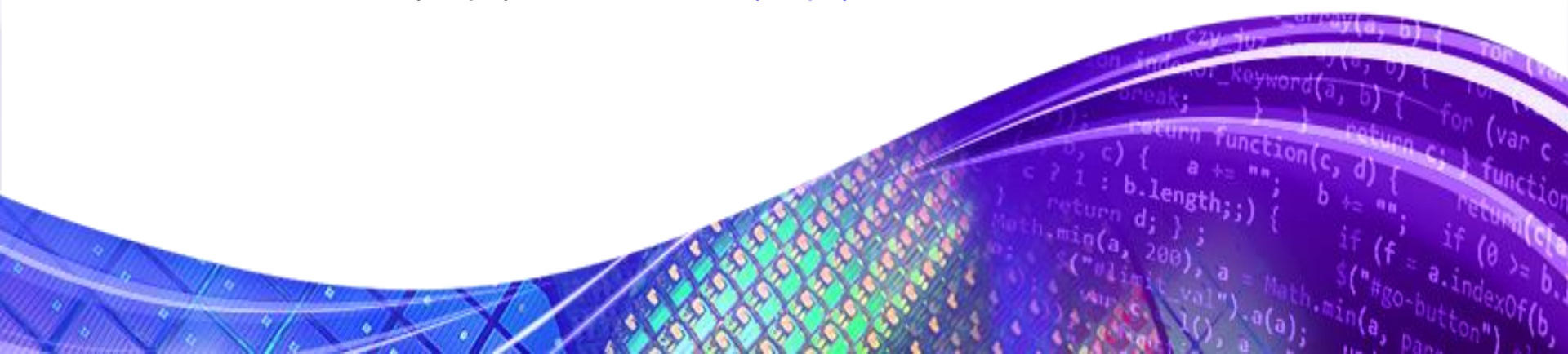


开源项目与代码的质量/安全/法律版权管控技术

韩葆 (Bob Han), Synopsys, Bao.Han@synopsys.com

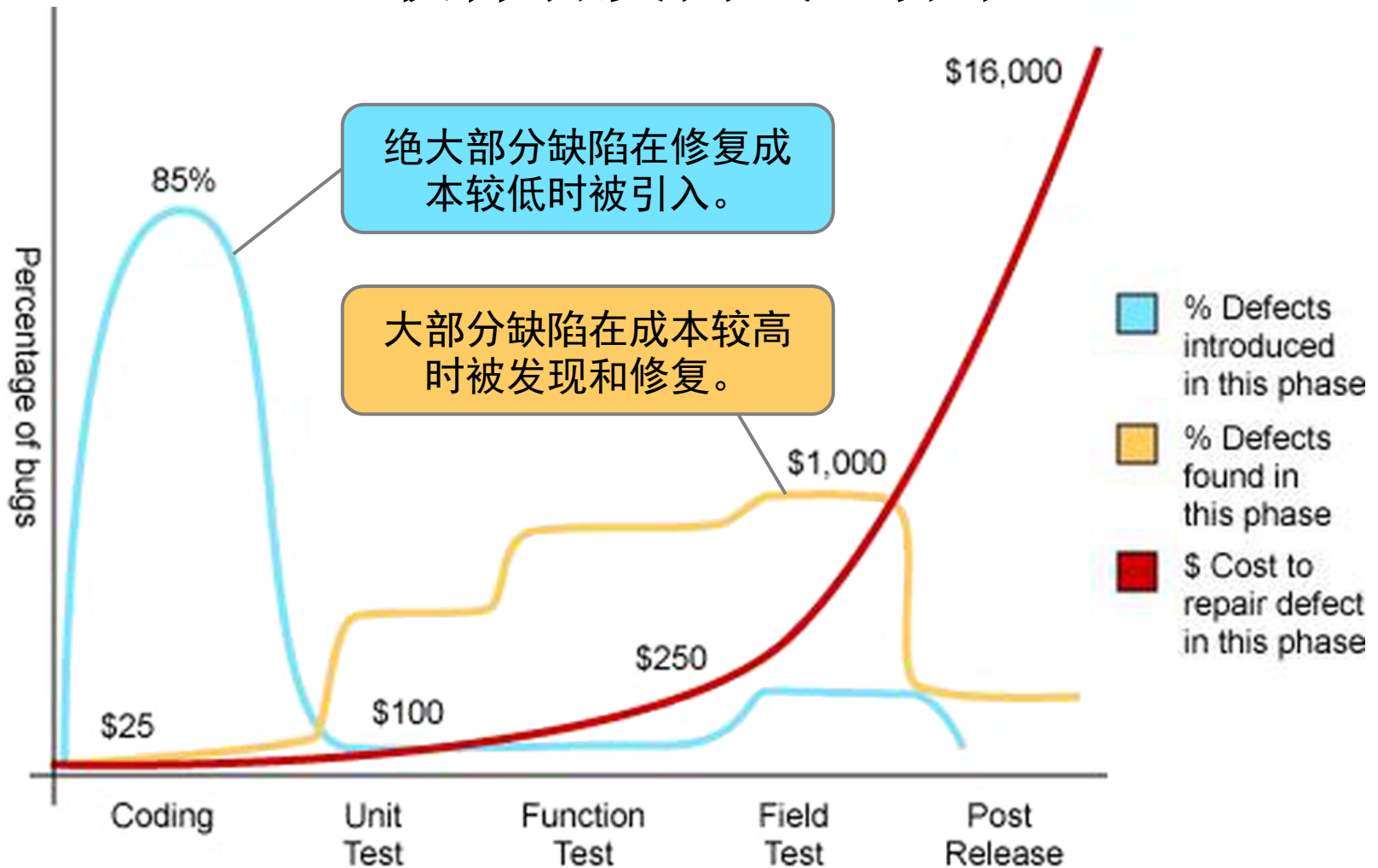


关于我-韩葆

- 程序员出身，专注软件质量与安全
- 直面新兴互联网公司与传统企业，一百多个团队，上千次交流
- Coverity 在国内第一个工程师（最佳SE），现任Synopsys Software Integrity Group 客户经理
- TID大会三届特邀讲师，ItechClub特邀讲师，Qcon大会讲师
- 联系方式：13311307163
- Email: Bao.Han@synopsys.com

开源项目的代码质量与安全管控-静态分析技术 (SAST)-Coverity

软件研发测试经济学



Source: Applied Software Measurement, Capers Jones, 1996

Coverity的静态分析方案

- 基于Meta Compilation的静态分析：
 - 由斯坦福大学教授Dawson Engler提出，在深度理解代码与程序语义的基础上检测缺陷
 - 旨在查找“真正的代码缺陷”
- 实现原理：
 - 使用可扩展的金属语言定义正确性Checker
 - 将程序的源码使用状态机进行抽象描述（State Machine Abstraction），可执行路径级别分析。
 - 使用xgcc系统匹配Checker与抽象状态机状态，找到问题所在的点。
- 可准确检测实际的Bug（内存和指针问题、资源泄露、缓冲区溢出，数组越界，心脏出血漏洞...）
 - 能够检测高达亿行级别的代码库，避免“状态爆炸”
 - 使用模型检验与符号执行技术，误报率降低至15%以下
- 算法已产品化-Coverity
 - 面向企业的Coverity 软件
 - 面向开源代码的Coverity SCAN

C语言静态分析

• Checker描述(metal 语言) 检测代码:

```
#include "extend-lang.hpp"
START_EXTEND_CHECKER( basic_leak, int_store );
ANALYZE_TREE()
{
    // Do not unmanage
    // We want the actual C functions; with un mangling we could get
    // e.g. Foo::malloc
    CallSite malloc( "malloc", /*unmanage*/false );
    CallSite free( "free", /*unmanage*/false );
    LocalVar lv;
    AnySubpart lv_sp( lv );
    if( MATCH( lv_sp = malloc( _ ) ) ) {
        SET_STATE( lv_sp, 1 );
        ADD_EVENT( lv_sp, "alloc", "Allocation assigned to " << lv_sp );
    }
    else if( MATCH( free( lv_sp ) ) ) {
        CLEAR_STATE( lv_sp );
    }
}
ANALYZE_END_OF_PATH()
{
    const ASTNode *t;
    int v;
    FOREACH_IN_STORE( t, v ) {
        COMMIT_ERROR( t, "leaked", t << " was not freed before the end of path" );
    }
}
END_EXTEND_CHECKER();
MAKE_MAIN( basic_leak )
```

```
1      #include<stdlib.h>
2      test()
3      {
```

(1) Event alloc: Allocation assigned to "p"
Also see events: [\[leaked\]](#)

```
4      void *p = malloc(10);
5      /* ... */
6      //free(p);
```

(2) Event leaked: "p" was not freed before the
end of path
Also see events: [\[alloc\]](#)

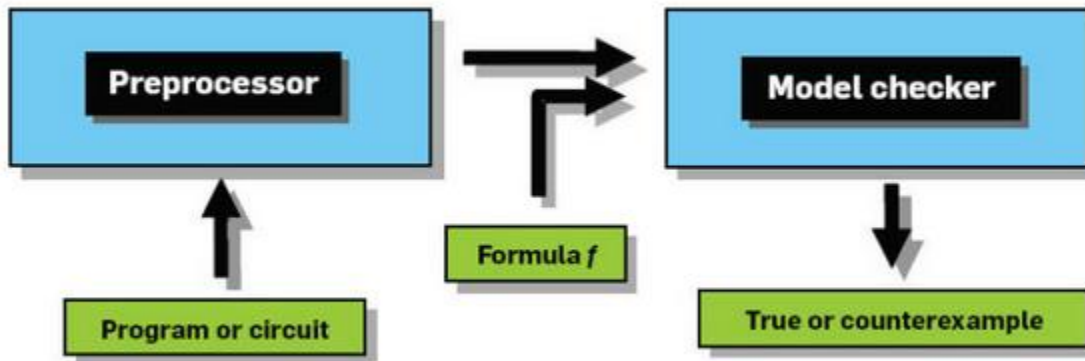
```
7      }
```

XGCC系统

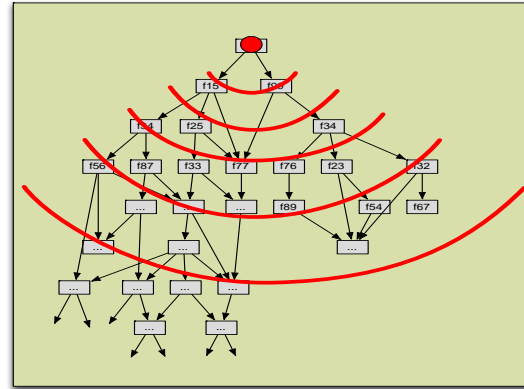
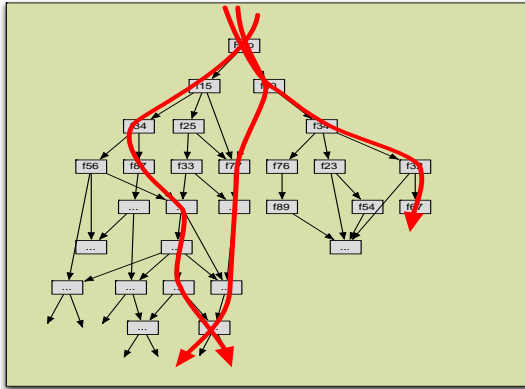
• 符号执行

- 不执行程序，用符号值表示程序变量的值，模拟程序执行
- 可以分析代码的所有/部分语义信息
- 避免状态爆炸

– 模型检验



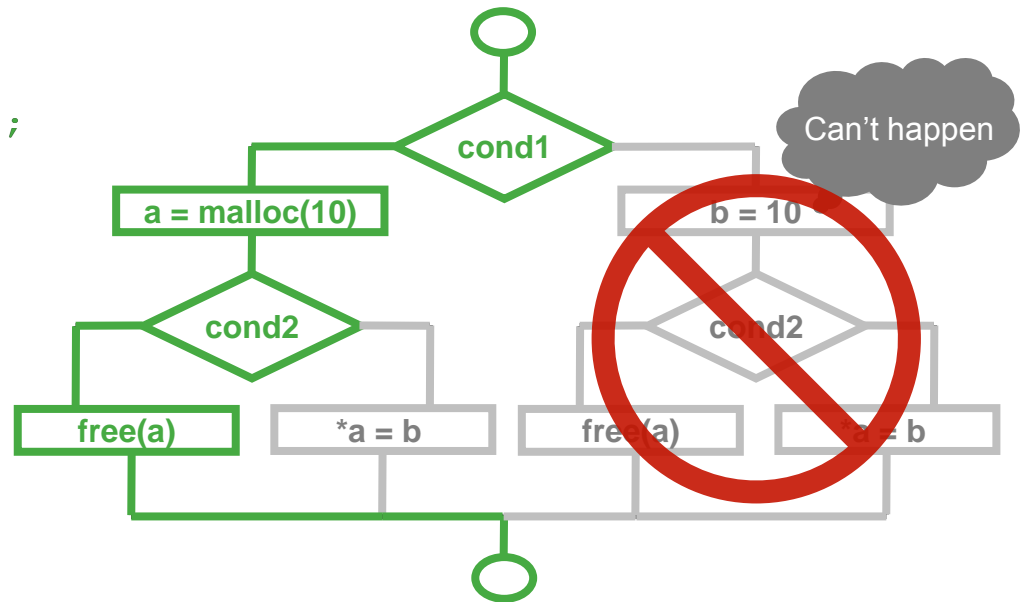
Dynamic and Static Testing



Control Flow Graph

```
if (cond1) {  
    a = malloc(10);  
} else {  
    b = 10;  
}
```

```
if (cond2) {  
    free(a);  
} else {  
    *a = b;  
}
```



C/C++/Objective-C 缺陷-Part 1

资源泄露

- 内存泄露
- Microsoft COM 内存泄露
- Object资源泄露
- 不当delete

未初始化变量

- 返回语句丢失
- 未初始化的指针/标量/数组 读写
- 类或结构体中未初始化的数据成员

并发缺陷

- 死锁
- 竞态条件 (Race conditions)
- 阻塞调用误用

算术错误

- 负变量不当使用
- 异常符号扩展
- 整数溢出
- 除零异常

内存崩溃

- 内存访问溢出
- 字符串长度计算错误
- 缓冲区溢出
- 写指针溢出
- 负数组索引写入
- 内存错误分配
- 错误的内存释放

非法内存访问

- 不正确的delete操作
- 溢出指针读取
- 越界读取
- 返回指针至本地变量
- 负数组索引读取
- 已释放指针读/写
- 不兼容的指针转换

控制流缺陷

- 逻辑/结构死代码
- Switch语句中break遗失
- 非本地资源不当使用

C/C++ 缺陷-Part 2

程序假死

- 死循环
- 双重锁或解锁丢失
- 负循环边界值
- 线程死锁
- 持锁过程中调用sleep()

空指针引用

- Null检查后引用空指针
- 直接引用返回的空指针
- Null检查前引用空指针

错误处理缺陷

- 未验证的返回值
- 未获取异常
- 负变量不当使用

代码维护性缺陷

- 多返回语句
- 无效变量

异常代码

- 复制/粘贴错误
- 格式错误

不安全的数据处理

- 不可信的循环数据源
- 使用非可信数据源读写数组/指针
- 使用非可信数据源格式化字符串

性能缺陷

- 值传递大参数
- 使用大堆栈

安全措施违反

- 缓冲区溢出
- 固定长度缓冲区写入
- 非安全函数调用
- 非安全临时文件使用
- 检查/使用时间不一致
- 用户空间指针不当使用

API错误使用

- 非安全chroot调用
- 错误的迭代器使用
- printf() 参数不匹配

C/C++ 安全问题

- CWE-20: 不当输入验证
- CWE-119: 内存缓冲区内操作不当限制
- CWE-120: 直接缓冲区拷贝 (典型缓冲区溢出)
- CWE-125: 越界读取
- CWE-129: 数组索引错误验证
- CWE-131: 缓冲区大小计算错误
- CWE-134: 不可控的字符串格式化
- CWE-170: 不当 Null 终止符
- CWE-190: 整数溢出或整数回绕
- CWE-415: 双重释放
- CWE-416: 释放后引用
- CWE-476: 空指针引用
- CWE-252: 未检验返回值
- CWE-367: (TOCTOU) 竞态条件
- CWE-369: 除零异常
- CWE-377: 不安全的临时文件
- CWE-394: 非预期状态码或返回值
- CWE-400: 不可控的资源消耗 (资源耗尽)
- CWE-401: 引用移除前不当内存释放 (内存泄露)
- CWE-590: 非堆内存释放
- CWE-188: 数据/内存布局依赖
- CWE-194: 非预期符号拓展
- CWE-195: Signed 到Unsigned 转换错误
- CWE-197: 数字截断错误
- CWE-243: 不改变工作目录调用chroot创建 Jail
- CWE-404: 不正确的资源关闭或释放
- CWE-459: 不完全清理
- CWE-465: 指针问题<<note: should be CWE-468 or something else>>
- CWE-467: 在指针上使用 sizeof
- CWE-662: 不当同步
- CWE-665: 错误初始化
- CWE-667: 不恰当的锁定
- CWE-676: 使用潜在危险函数
- CWE-681: 数据类型间不当转换
- CWE-704: 类型不当转换
- CWE-833: 死锁
- CWE-835: 死循环

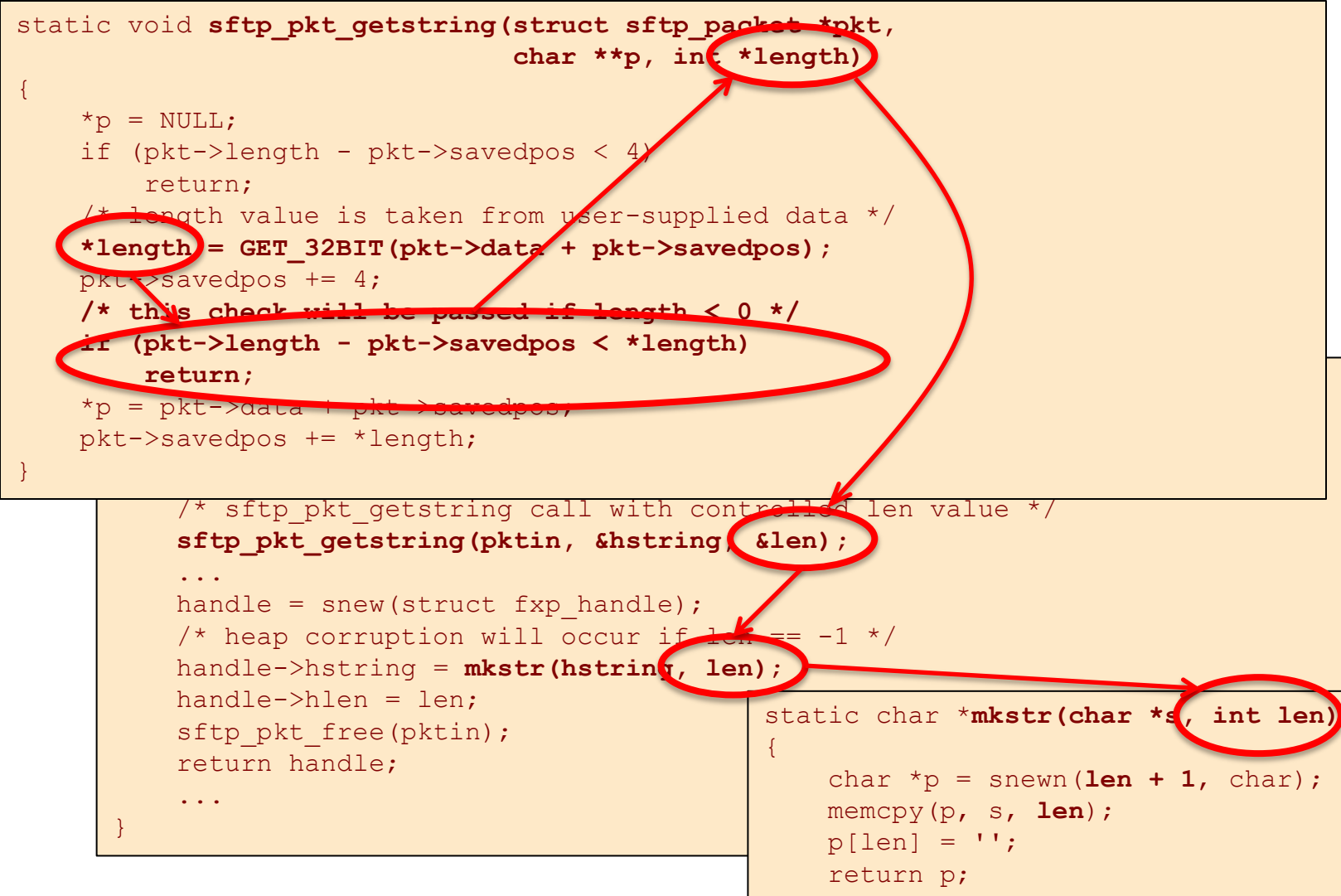
C/C++ 问题示例-跨分支识别

- PuTTY 2005年漏洞
 - CVE-2005-0467


```
static char *foo(char *s, int len)
{
    char *p = malloc((len + 1) * sizeof(char));
    memcpy(p, s, len);
    p[len]='\0';
    return p;
}
```

```
static char *mkstr(char *s, int len)
{
    char *p = snewn(len + 1, char);
    memcpy(p, s, len);
    p[len] = '\0';
    return p;
}
```

CVE-2005-0467



CVE-2005-0467



CVE LISTCOMPATIBLE PRODUCTSNEWS — FEBRUARY 27, 2012SEARCH

Common Vulnerabilities and Exposures
The Standard for Information Security Vulnerability Names

TOTAL CVEs: 49296

HOME > CVE > CVE-2005-0467 (UNDER REVIEW)

About CVE
Terminology
Documents
FAQs
CVE List
About CVE Identifiers
Search CVE
Search NVD
Updates & RSS Feeds
Request a CVE-ID
CVE In Use
CVE Adoption
CVE-Compatible Products
NVD for CVE Fix
Information
More...
News & Events
Calendar
Free Newsletter
Community
CVE Editorial Board
Sponsor
Contact Us
Search the Site

[Printer-Friendly View](#)

CVE-ID	CVE-2005-0467 (under review) Learn more at National Vulnerability Database (NVD) • Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings
Description	Multiple integer overflows in the (1) sftp_pkt_getstring and (2) fxp_readdir_rcv functions in the PSFTP and PSCP clients for PuTTY 0.56, and possibly earlier versions, allow remote malicious web sites to execute arbitrary code via SFTP responses that corrupt the heap after insufficient memory has been allocated.
References	Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete. <ul style="list-style-type: none">• IDEFENSE:20050221 Multiple PuTTY SFTP Client Packet Parsing Integer Overflow Vulnerabilities• URL:http://www.iddefense.com/application/poi/display?id=201&type=vulnerabilities• CONFIRM:http://www.chiark.greenend.org.uk/~sgtatham/putty/wishlist/vuln-sftp-string.html• CONFIRM:http://www.chiark.greenend.org.uk/~sgtatham/putty/wishlist/vuln-sftp-readdir.html• CONFIRM:http://www-1.ibm.com/support/docview.wss?uid=ssq1S1002414• CONFIRM:http://www-1.ibm.com/support/docview.wss?uid=ssq1S1002416• GENTOO:GLSA-200502-28• URL:http://www.gentoo.org/security/en/qlsa/qlsa-200502-28.xml• SECUNIA:14333• URL:http://secunia.com/advisories/14333• SECUNIA:17214• URL:http://secunia.com/advisories/17214• XF:putty-sftppktgetstring-bo(19403)• URL:http://xforce.iss.net/xforce/xfdb/19403

CVE List
About CVE Identifiers
Editorial Policies
Data Sources
Reference Key/Maps
Search Tips
Updates & RSS Feeds
Obtain a CVE Identifier
ITEMS OF INTEREST
Terminology
NVD

检测实例-HeartBleed Bug

CID	Type	Impact	Status	Count	First Detected	Owner	Classification	Severity
1201706	Untrusted value as argument	Medium	New	1	04/12/14	Unassigned	Unclassified	Unspecified

All 1 issue selected

< Page 1 of 1 >

```

1445
1446 #ifndef OPENSLL_NO_HEARTBEATS
1447 int
1448 dtls1_process_heartbeat(SSL *s)
1449 {
1450     unsigned char *p = &s->s3->rrec.data[0], *pl;
1451     unsigned short hbtype;
1452     unsigned int payload;
1453     unsigned int padding = 16; /* Use minimum padding */
1454
1455     /* Read type and payload length first */
1456     hbtype = *p++;
1457     n2s(p, payload);
1458     pl = p;
1459
1460     if (s->msg_callback)
1461         s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
1462             &s->s3->rrec.data[0], s->s3->rrec.length,
1463             s, s->msg_callback_arg);
1464
1465     if (hbtype == TLS1_HB_REQUEST)
1466     {
    
```

1. byte_swapping: Performing a byte swapping operation on `p` implies that it came from an external source, and is therefore tainted.

2. var_assign_var: Assigning: `payload = ((unsigned int)p[0] << 8) | (unsigned int)p[1]`. Both are now tainted.

3. Condition `s->msg_callback`, taking true branch

4. Condition `hbtype == 1`, taking true branch

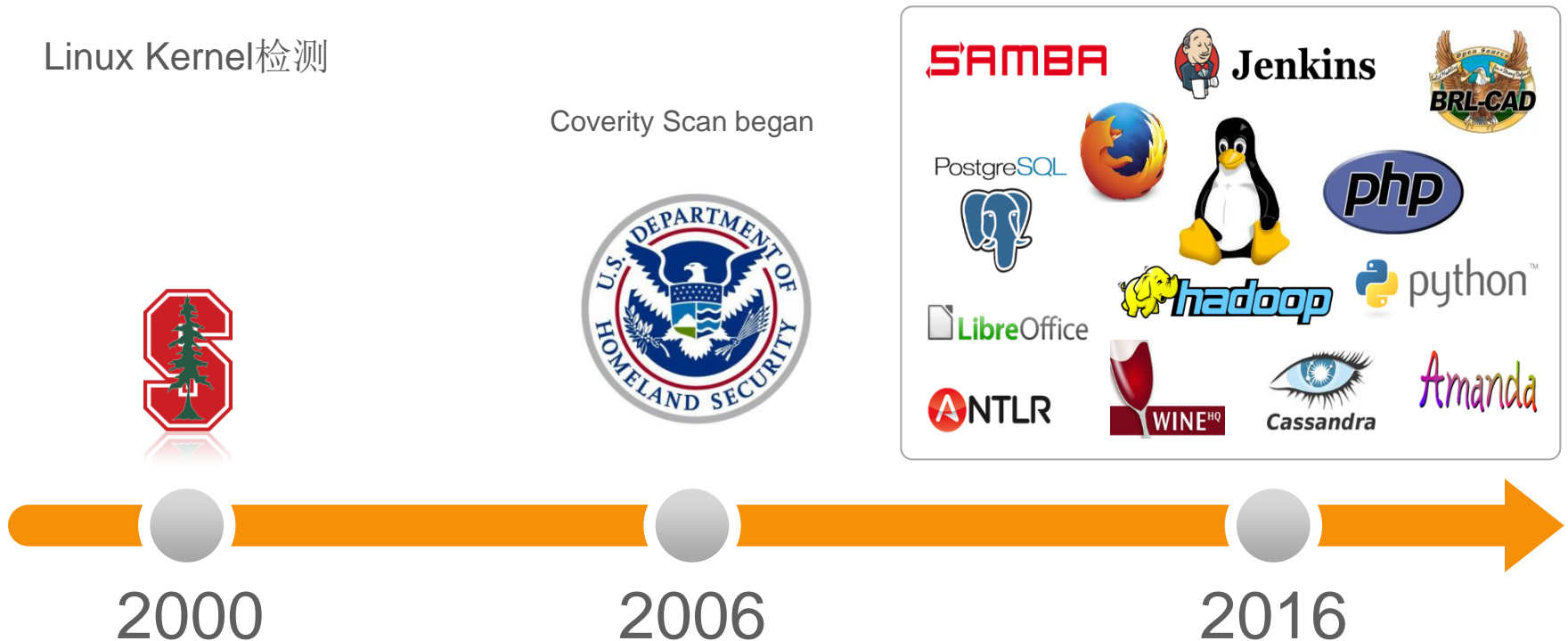
Coverity and Open-Source (Cloud Scan)

Free cloud-based service for the open source community

Linux Kernel检测

Coverity Scan began

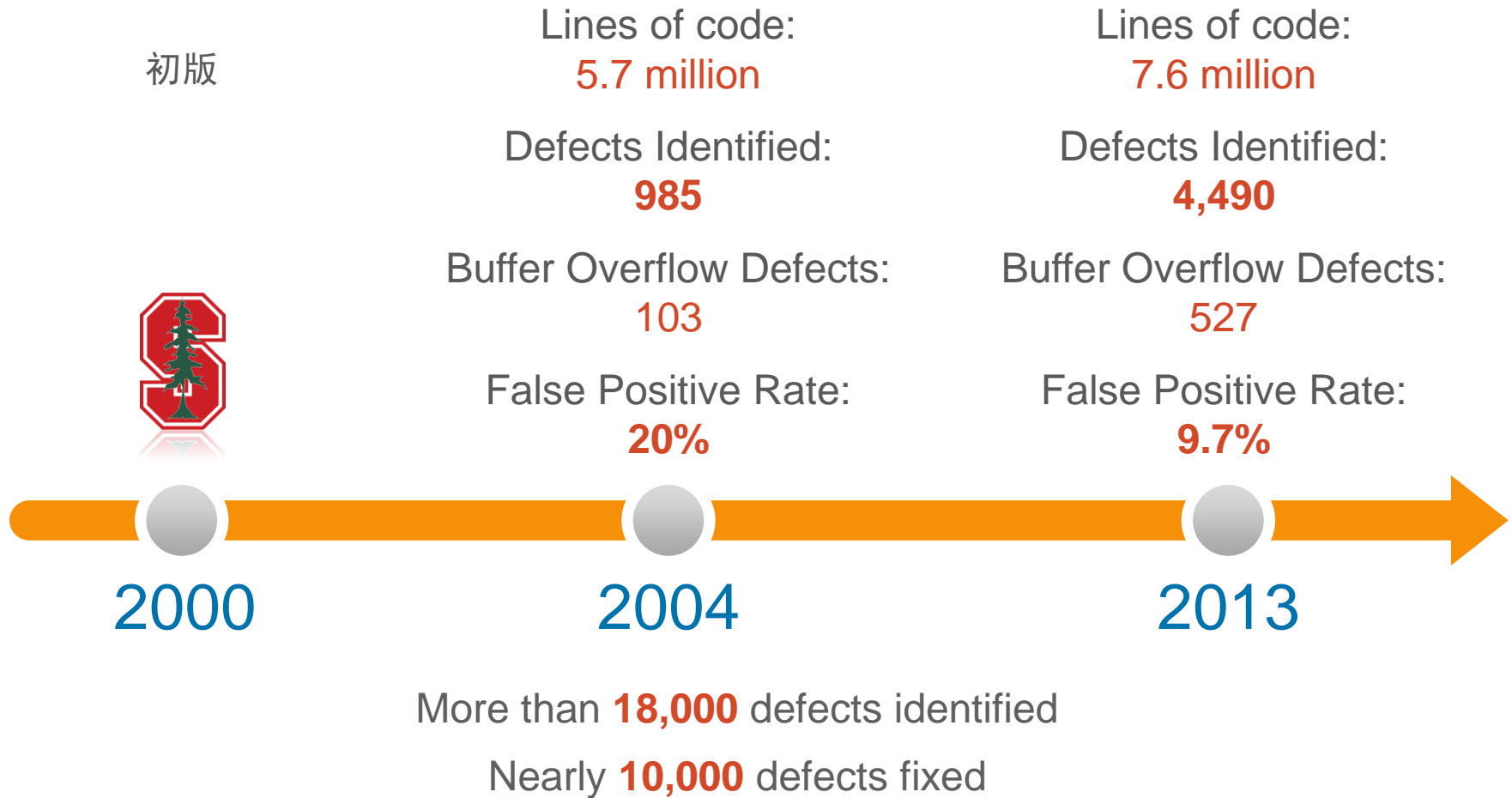
Proven developer adoption



Over 2000+ projects and 300M lines of code

Over 45,000 defects fixed by the community

Coverity and Linux



Linux 3.8 Kernel – Fixed Defects in 2013



- LOC: **7.8M**
- Defect Density: **0.66**
- False Positive rate:
7.85%
- Developers using
Coverity: **~50**

Category	# defects
API usage errors	11
Code maintainability issues	51
Concurrent data access violations	3
Control flow issues	186
Error handling issues	157
Incorrect expression	72
Insecure data handling	41
Integer handling issues	795
Memory - corruptions	723
Memory - illegal accesses	140
Null pointer dereferences	235
Performance inefficiencies	1
Program hangs	3
Resource leaks	94
Security best practices violations	21
Uninitialized variables	89
Various	1
Total	2,623

Linux- Defect Example Resource Leak



alloc_fn: Calling allocation function "kzalloc". bss_cfg is assigned

```
934
  CID 709078 (#1 of 1): Resource leak (RESOURCE_LEAK)
  alloc_fn: Calling allocation function "kzalloc".
  var_assign: Assigning: "bss_cfg" = storage returned from "kzalloc(132UL, 208U)".
935     bss_cfg = kzalloc(sizeof(struct mwifiex_uap_bss_param), GFP_KERNEL);
  At conditional (1): "!bss_cfg" taking the false branch.
936     if (!bss_cfg)
937         return -ENOMEM;
938
  noescape: Variable "bss_cfg" is not freed or pointed-to in function "mwifiex_set_sys_config_invalid_data".
939     mwifiex_set_sys_config_invalid_data(bss_cfg);
940
  At conditional (2): "params->beacon_interval" taking the true branch.
941     if (params->beacon_interval)
942         bss_cfg->beacon_period = params->beacon_interval;
  At conditional (3): "params->dtim_period" taking the true branch.
943     if (params->dtim_period)
944         bss_cfg->dtim_period = params->dtim_period;
945
```

Linux- Defect Example Resource Leak



“bss_cfg” is not freed

934

❖ CID 709078 (#1 of 1): Resource leak (RESOURCE_LEAK)
alloc_fn: Calling allocation function "kzalloc".

var_assign: Assigning: "bss_cfg" = storage returned from "kzalloc(132UL, 208U)".

935

```
bss_cfg = kzalloc(sizeof(struct mwifiex_uap_bss_param), GFP_KERNEL);
```

At conditional (1): "!bss_cfg" taking the false branch.

936

```
if (!bss_cfg)
```

937

```
return -ENOMEM;
```

938

noescape: Variable "bss_cfg" is not freed or pointed-to in function "mwifiex_set_sys_config_invalid_data".

939

```
mwifiex_set_sys_config_invalid_data(bss_cfg);
```

940

At conditional (2): "params->beacon_interval" taking the true branch.

941

```
if (params->beacon_interval)
```

942

```
bss_cfg->beacon_period = params->beacon_interval;
```

At conditional (3): "params->dtim_period" taking the true branch.

943

```
if (params->dtim_period)
```

944

```
bss_cfg->dtim_period = params->dtim_period;
```

945

Linux- Defect Example Resource Leak

“bss_cfg” out of scope and leaks



```
951     switch (params->hidden_ssid) {
952     case NL80211_HIDDEN_SSID_NOT_IN_USE:
953         bss_cfg->bcast_ssid_ctl = 1;
954         break;
955     case NL80211_HIDDEN_SSID_ZERO_LEN:
956         bss_cfg->bcast_ssid_ctl = 0;
957         break;
958     case NL80211_HIDDEN_SSID_ZERO_CONTENTS:
959         /* firmware doesn't support this type of hidden SSID */
960     default:
961         return -EINVAL;
962     }
```

At conditional (6): switch case value "NL80211_HIDDEN_SSID_ZERO_CONTENTS" taking the true branch.

leaked_storage: Variable "bss_cfg" going out of scope leaks the storage it points to.

```
> *. CID 709078: Resource leak (RESOURCE_LEAK)
>   - drivers/net/wireless/mwifiex/cfg80211.c, line: 935
> Assigning: "bss_cfg" = storage returned from "kzalloc(132UL, 208U)"
>   - but was not free
> drivers/net/wireless/mwifiex/cfg80211.c:935

Signed-off-by: Bing Zhao <bzhao@marvell.com>
---
drivers/net/wireless/mwifiex/cfg80211.c |    1 +
1 files changed, 1 insertions(+), 0 deletions(-)

diff --git a/drivers/net/wireless/mwifiex/cfg80211.c b/drivers/net/wireless/mwifiex/cfg80211.c
index 3875b1a..6c57e83 100644
--- a/drivers/net/wireless/mwifiex/cfg80211.c
+++ b/drivers/net/wireless/mwifiex/cfg80211.c
@@ -1039,6 +1039,7 @@ static int mwifiex_cfg80211_start_ap(struct wiphy *wiphy,
     case NL80211_HIDDEN_SSID_ZERO_CONTENTS:
         /* firmware doesn't support this type of hidden SSID */
     default:
+        kfree(bss_cfg);
         return -EINVAL;
```

The Fix:

<http://marc.info/?l=linux-wireless&m=134135643727424&w=2>

如何进行Java代码静态分析？

Java语言被编译成JVM bytecode - 在运行时被转换成本地可执行代码的分析

选项一

- 分析 byte-code：用户编译他们的软件，然后分析编译后的可执行文件与调试信息，分析引擎联系找到的缺陷与源代码位置
- 某些开源工具的实现原理

选项二：

- 获取所有的Java编译过程并执行分析
- Bytecode分析工作仍旧存在，但包含更多的内容

Coverity OWASP top 10: JSP/ASP/JS/Node.JS

OWASP 10	CWE映射
A1: 注入	77, 78, 88, 89, 90, 564, 917
A2: 失效认证与会话管理	259, 321, 384, 798
A3: 跨站脚本攻击 (XSS)	79, 80, 81, 82, 83, 84, 86, 87
A4: 不安全的直接对象引用	22, 23, 36
A5: 安全配置错误	4, 7, 86, 650
A6: 敏感信息泄露	321
A7: 功能级访问控制缺失	425, 862, 863
A8: 跨站请求伪造	352
A9: 使用含有已知漏洞的组件	NA
A10: 未验证的重定向和转发	938

Java 缺陷

Web 应用安全缺陷（OWASP Top 10）

- 跨站脚本攻击
- SQL 注入
- 命令行注入
- 路径遍历...

资源泄露

- 数据库连接资源泄露
- 资源泄露
- Socket & Stream 泄露

并发数据访问异常

- 变量非原子更新
- 双重检查锁定
- 数据竞态条件
- Volatile非原子更新
- Servlet 属性无效锁定
- 单例模式竞态条件

程序假死

- 线程死锁
- 死锁

空指针引用

- Null检查后引用空指针
- 直接引用返回的空指针
- Null检查前引用空指针

API 使用错误

- 无效迭代器使用
- 不可修改的集合错误
- 已释放资源调用

性能缺陷

- 低效率方法使用
- 在循环中连接字符串
- 冗余同步

逻辑错误

- 不可达代码
- 未使用变量
- 常量表达式
- 非本地资源不当使用
- 整数溢出
- 不当分号

Java 缺陷

类层次结构不一致

- 调用 `super.clone()` 或 `super.finalize()` 失败
- 父函数调用丢失
- 构造函数中使用虚函数

控制流缺陷

- 在 `Finally` 模块中返回
- `Switch` 语句中 `break` 丢失

错误处理缺陷

- 未验证的返回值

数据库操作

- 不正确的实体哈希
- `Load` 函数返回值错误验证
- 不完全持续周期
- `get()` 不当使用

代码可维护性缺陷

- 调用已过期方法
- 显式垃圾收集
- 非静态方法中设置静态变量
- 复制/粘贴错误
- 不可达代码

可疑代码

- 参数次序错误
- 格式错误

Java 安全缺陷

CWE-22: 路径遍历

CWE-23: 相对路径遍历

CWE-36: 绝对路径遍历

CWE-78: 系统命令行注入

CWE-79: 跨站脚本攻击

CWE-81: 错误页面数据泄露

CWE-89: SQL 注入

CWE-113: 应答拆分

CWE-171: 规范化错误

CWE-252: 未验证的返回值

CWE-259: 固定密码

CWE-366: 线程竞态条件

CWE-374: 可变对象传递

CWE-382: 使用 System.exit()

CWE-391: 未检查的错误条件

CWE-404: 错误关闭

CWE-476: 空指针引用

CWE-493: Final模块丢失

CWE-564: SQL 注入: Hibernate

CWE-567: 未同步的数据获取

CWE-572: 不当线程初始化

CWE-573: 规格不当调用

CWE-583: 公共 finalize() 方法

CWE-584: Finally 模块中返回

CWE-586: 直接 Finalize() 调用

CWE-595: 错误类型比较

CWE-597: 字符串比较过程中的错误操作

CWE-609: 双中检查锁定

CWE-662: 同步错误

CWE-665: 初始化错误

CWE-674: 非可控递归

CWE-833: 死锁

C# 缺陷 Powered by Eric Lippert

资源泄露

- 数据库连接资源泄露
- 资源泄露
- Socket & Stream 泄露

API 使用错误

- 已释放资源调用

并发数据访问异常

- 变量非原子更新
- 数据竞态条件

性能缺陷

- 低效率方法使用
- 在循环中连接字符串
- 冗余同步

程序假死

- 线程死锁
- 死循环

可疑代码

- 复制/粘贴错误
- 参数次序错误
- 格式错误

类层次结构不一致

- 调用 `base.close()` 或 `base.dispose()` 失败
- 父函数调用丢失

控制流缺陷

- 可疑的额外分号
- 不一致比较
- 不兼容的类型比较

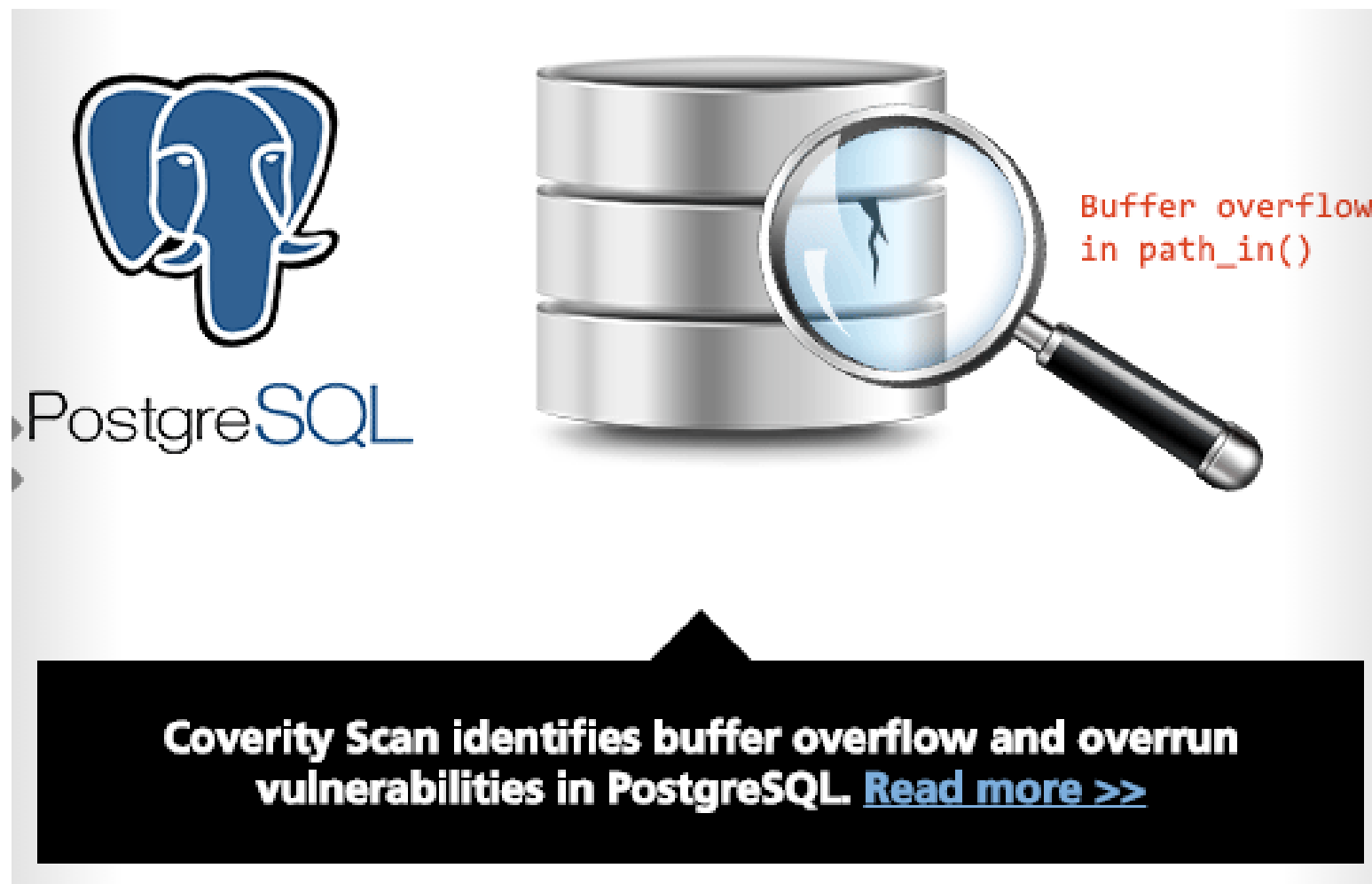
空指针引用

- Null检查后引用空指针
- 直接引用返回的空指针
- Null检查前引用空指针

算术错误

- 错误移位操作
- 不正确的表达式
- 表达式计算过程中溢出

Cloud Scan示例



Cloud Scan示例



Coverity Scan finds Remote Code Execution in Apache Roller via OGNL Injection. [Read more >>](#)

Cloud Scan示例

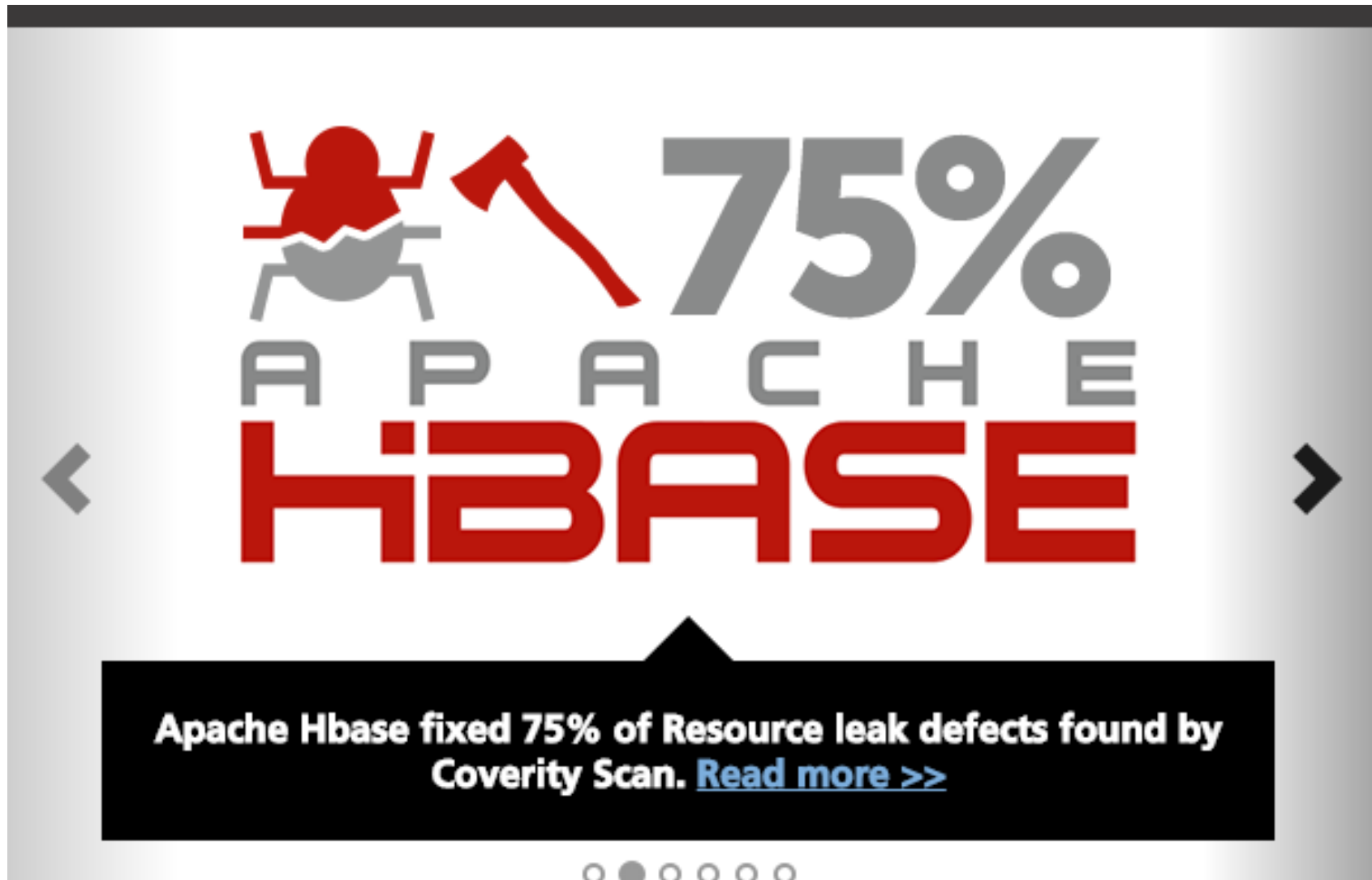
iOS

```
85     if ((err = Readynash(&SSLHashSHA1, &hashCtx)) != 0)
86         goto fail;
87     if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
88         goto fail;
89     if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
90         goto fail;
91     if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
92         goto fail;
```

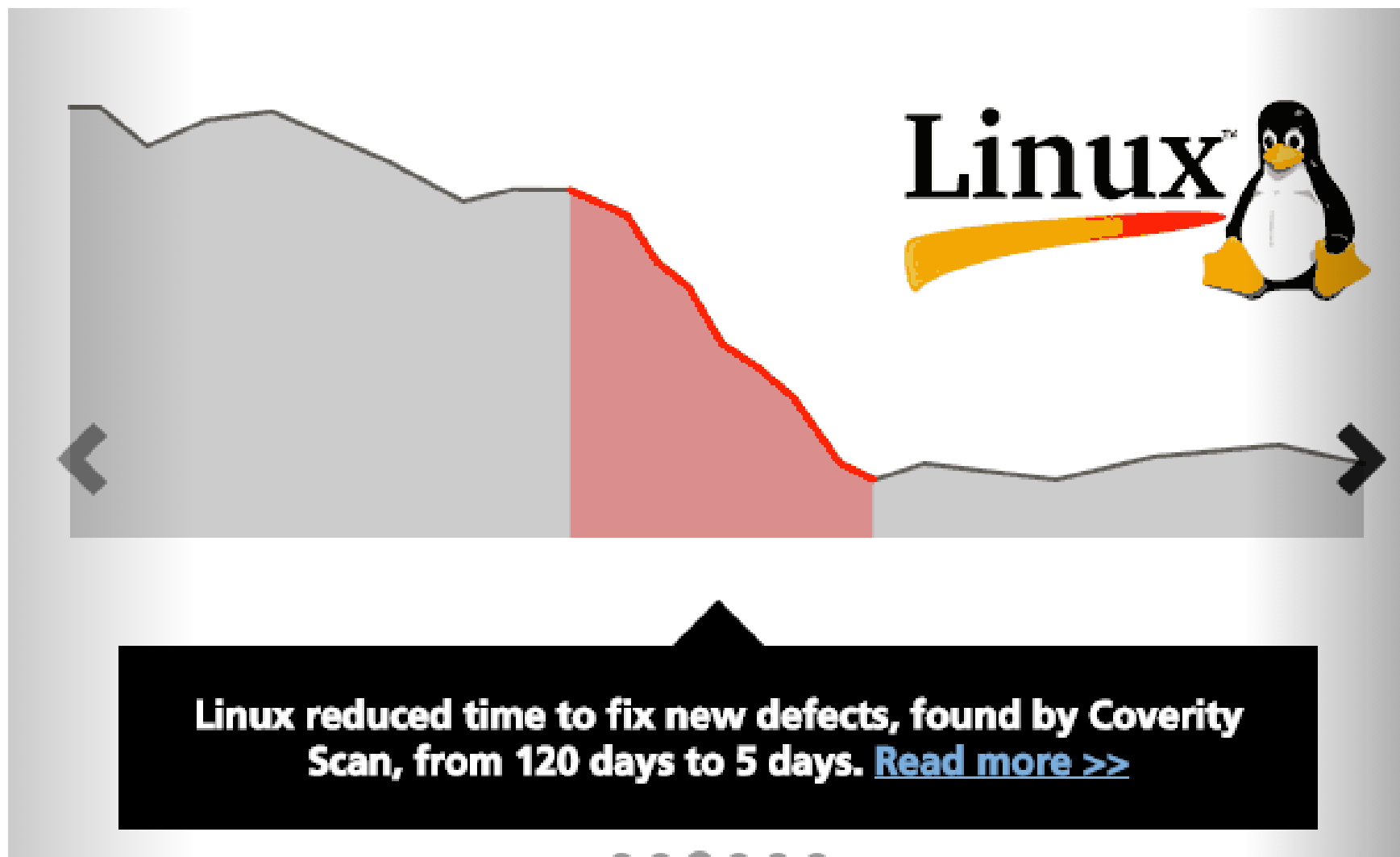
❖ CID 1186789 (#1 of 1): Structurally dead code (UNREACHABLE)
unreachable: This code cannot be reached: "if ((err = (*SSLHashSHA1.fi...".

Coverity static analysis successfully uncovers "goto fail"
SSL/TLS defect in iOS. [Read more >>](#)

Cloud Scan示例



Cloud Scan示例



检测实例-资源泄露

webgoat

问题: 按快照

Outstanding Security Risks

筛选器: 问题类型, 分类

CID	类型	影响	状态	首次被检测到	所有者	文件	分类
13137	资源泄露	高	新增	2012 年 9 月 9 日	未分配	/src/main/java/o...evelLogin2.java	未分类
13138	资源泄露	高	新增	2012 年 9 月 9 日	未分配	/src/main/java/o...evelLogin2.java	未分类
13139	资源泄露	高	新增	2012 年 9 月 9 日	未分配	/src/main/java/o...evelLogin2.java	未分类
13141	资源泄露	高	新增	2012 年 9 月 9 日	未分配	/src/main/java/...ge2Screen.java	未分类
13142	资源泄露	高	新增	2012 年 9 月 9 日	未分配	/src/main/java/o...apRequest.java	未分类
13143	资源泄露	高	新增	2012 年 9 月 9 日	未分配	/src/main/java/or...onFixation.java	未分类

1 个 (共 161 个) 问题 已选中

SoapRequest.java

SoapRequest.java

```
400
401
402 {
403     try
404     {
405         Connection connection = DatabaseUtilities.getConnection("guest", getWebgoatContext());
406         1.alloc_fn: 从分配方法 prepareStatement 返回了新资源。
407         2.var_assign: 赋值: ps = 从 connection.prepareStatement("SELECT * FROM user_data WHERE userid = ?") 返回的资源。
408         PreparedStatement ps = connection.prepareStatement("SELECT * FROM user_data WHERE userid = ?");
409         ps.setInt(1, id);
410         try
411         {
412             3.noescape: 资源 ps 在 executeQuery 中未关闭或保存。
413             ResultSet results = ps.executeQuery();
414             4.条件 results != null, 使用了 true 分支
415             5.条件 results.next() == true, 使用了 false 分支
416             if ((results != null) && (results.next() == true)) { return results.getString(field); }
417             6.导致 try 语句结束
418             } catch (SQLException sqle)
419             {
420             }
421         } catch (Exception e)
422         {
423         }
424         return null;
425     }
426 }
```

CID 13142 (2 的 1 数): 资源泄露 (RESOURCE_LEAK) 选择问题



4.条件 results != null, 使用了 true 分支

5.条件 results.next() == true, 使用了 false 分支

CID 13142 (2 的 2 数): 资源泄露 (RESOURCE_LEAK)

7.leaked_resource: 变量 ps 超出范围将泄露它引用的资源。

检测实例-SQL Injection

webgoat												
Issues: By Snapshot Outstanding Security Risks   Filters: Issue Kind, Classification												
CID	Type	Impact	Status	First Detected	Owner	Classification	Severity	Action	Component	Category	File	
11593	Cross-site scripting	High	Triaged	09/01/12	jon	Bug	Unspecifie	Fix Requ	Other	High impact security	/data00/src/webgoat/we	
11970	SQL injection	High	New	07/23/14	admin	Unclassified	Unspecifie	Undecide	Other	High impact security	/Demo/projects/webgoa	
11929	Regular expression injection	Low	New	07/23/14	admin	Unclassified	Unspecifie	Undecide	Other	Low impact security	/Demo/projects/webgoa	
11917	OS Command Injection	High	New	07/23/14	admin	Unclassified	Unspecifie	Undecide	Other	High impact security	/Demo/projects/webgoa	
11914	Filesystem path or filename man	High	New	07/23/14	admin	Unclassified	Unspecifie	Undecide	Other	High impact security	/Demo/projects/webgoa	
11873	Cross-site request forgery	High	New	07/23/14	admin	Unclassified	Unspecifie	Undecide	Other	High impact security	/Demo/projects/webgoa	
12097	Cross-site scripting	High	New	07/23/14	Unassigned	Unclassified	Unspecifie	Undecide	Other	High impact security	/Demo/projects/webgoa	

1 of 199 issues selected

Page 1 of 1

```
ThreadSafetyProblem.java
5. tainted_paul_call: org.owasp.webgoat.session.ParameterParser.getRawParameter(java.lang.String, java.lang.String) returns the tainted data.
80     currentUser = s.getParser().getRawParameter(USER_NAME, "");
81     originalUser = currentUser;
82
83     // Store the user name
84     String user1 = new String(currentUser);
85
86     Element b = ECSFactory.makeButton("Submit");
87     ec.addElement(b);
88     ec.addElement(new P());
89
90     if (!"".equals(currentUser))
91     {
92         Thread.sleep(1500);
93
94         // Get the users info from the DB
95         String query = "SELECT * FROM user_system_data WHERE user_name = '" + currentUser + "'";
96         Statement statement = connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
97                                     ResultSet.CONCUR_READ_ONLY);
98         ResultSet results = statement.executeQuery(query);
99
100         // ...
101     }
102 }
```

CID 11970 (#1 of 1): SQL injection (SQLI)

7. **sql_taint:** Insecure concatenation of a SQL statement. The value `org.owasp.webgoat.lessons.ThreadSafetyProblem.currentUser` is tainted.

Remediation for SQL injection in JDBC: Specific advice for SQL string

- Refactor the JDBC code to use the PreparedStatement API versus Statement.
- Add a positional parameter to the SQL statement using "?".
- Bind the tainted value to the parameter using the `setString` method: `PreparedStatement.setString(1, org.owasp.webgoat.lessons.ThreadSafetyProblem.currentUser)`.

[More Information](#)

8. **sql_sink:** Passing the tainted value `query` to the SQL API `java.sql.Statement.executeQuery(java.lang.String)` may allow an attacker to inject SQL.

命令行注入

问题: 按快照 | Outstanding Security Risks

筛选器: 问题类型, 分类

CID	类型	影响	状态	首次被检测到	所有者	文件	分类	严重性
13102	缺少全局异常处理程序	低	新增	2012 年 9 月 9 日	未分配	/target/WebGoa...B-INF/web.xml	未分类	未指定
13126	操作系统命令注入	高	新增	2012 年 9 月 9 日	未分配	/src/main/java/or...Jutil/Exec.java	未分类	未指定
13127	操作系统命令注入	高	新增	2012 年 9 月 9 日	未分配	/src/main/java/or...Jutil/Exec.java	未分类	未指定

1 个 (共 161 个) 问题 已选中

ParameterParser.java

3. taint_path_call: org.owasp.webgoat.session.ParameterParser.getRawParameter(java.lang.String) 将返回被污染的数据。

4. taint_path_return: 返回被污染的数据。

```
597         return getRawParameter(name);
598     } catch (Exception e)
599     {
600         return def;
601     }
602 }
603
604 /**
605  * Gets the rawParameter attribute of the ParameterParser object
606  *
607  * @param name
608  *      Description of the Parameter
609  * @return The rawParameter value
610  * @exception ParameterNotFoundException
611  *      Description of the Exception
612  */
613 public String getRawParameter(String name) throws ParameterNotFoundException
614 {
615     String[] values = request.getParameterValues(name);
616
617     if (values == null)
618     {
619         throw new ParameterNotFoundException(name + " not found");
620     }
621     else if (values[0].length() == 0) { throw new ParameterNotFoundException(name + " was empty"); }
622
623     2. taint_path_return: 返回被污染的数据。
624     return (values[0]);
625 }
626
627 // start the command
288
289 child = Runtime.getRuntime().exec(command);
290
```

◆ CID 13126 (1 的 1 数): 操作系统命令注入 (OS_CMD_INJECTION)

11. os_cmd_sink: 将被污染的值 command 传递给流程调用 API java.lang.Runtime.exec(java.lang.String) 可能允许攻击者修改该命令的目的。

💡 修复 Runtime.exec(String) 中的 OS 命令注入: 一般建议 OS command executable and arguments

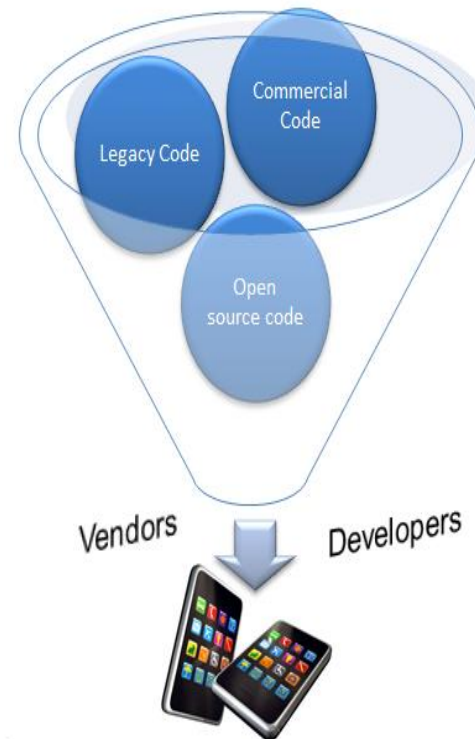
- 创建一个包含命令所有部分的数组, 并使用 Runtime.exec(String[]) 方法。
- 确保被污染的数据无法修改 OS 命令的目的。

更多信息

开源项目的代码法律版权管控技术

Problem: 未知的代码来源

- ❑ 现如今的代码很少是独立编写的
- ❑ 软件工程包含开源的或其他第三方的代码
 - 增加了软件复杂度
 - 加速开发和提高开发效率的需要
 - 便于访问开源代码
 - 供应商和外包商大量的使用
- ❑ 然而: 外部代码带来风险
 - 版权和license合规性问题
 - 安全漏洞
 - 质量损害



- Commercial licenses
- Open source licenses
- Copyright notices
- Security vulnerabilities
- Encryption requirements
- Export control restrictions
- ...

未知代码的危害

- 带来的风险
 - 未知的知识产权问题 (“我们是否有权使用我们的代码?”)
 - 未知的license使用权限
 - 未知的漏洞
 - 未知的质量
- 为公司和下游的客户供应链带来未知的法律风险
- 减少了创造合作伙伴的机会
- 降低了公司的品牌和信誉
- 阻止了潜在买家和企业并购行为



Software License Management Maturity Model

- 手动方式
 - 时间消耗
 - 高错误率的倾向
- 自动化测试
 - 加速问题的发现
 - 创建软件物料清单(or BoM)
 - 允许基于策略的代码管理
 - 可以集成到软件开发和管理流程当中
 - 可以为组织内部或组织间提供报告



Who Needs Open Source Code Management?

任何生产或使用软件的用户:

- 电信
- 网络
- 游戏
- 金融
- 移动终端
- 健康
- 能源
- 芯片供应商
- 医疗研究
- 程序控制
- 政府



解决方案与技术细节

• 产品和服务

- 检测以及管理第三方代码的开源内容(指纹匹配)
- 确保license和知识产权的合规性（自定义）
- 发现安全漏洞（质量与安全控制）
- 报告基于出口控制合规性的加密内容

Protecode Enterprise™

- 适用于企业的可扩展的扫描和管理解决方案
- 多个产品组件，企业级用户系统管理特征
- 适用于大型，分布式组织

Annual
license
fee

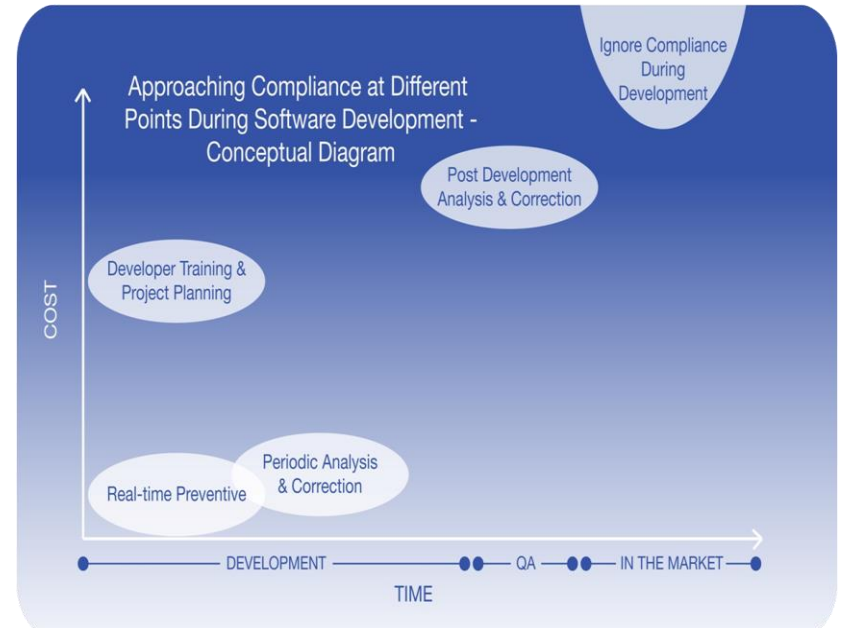
Certified Audit

- Protecode Certified™ 基于软件的一次性评估审计服务

Fee
based
on
portfolio
size

解决方案

- 软件开发各个阶段的解决方案
 - 从桌面开发到版本构建
 - 从设计到市场运作
- 基于工作流程的解决方案
 - 软件包请求/预批准方案
 - 强力的软件开发生命周期集成工具
 - 设置策略、请求、批准、扫描、确认，构建物料清单，在组织内共享结果。
- 高扩展性、智能化、易于采纳和使用
- 完整的产品组合
 - 执行一个结构化的开源软件采纳流程 ([OSSAP](#))



整体方案



- Enterprise Server (ES)

- 核心的分析和功能

- 策略、团队、组、用户管理、使用权限
- 报告管理（总览、代码片段、包、版权、license、加密、出口控制等级、版本、安全漏洞）



- Code Administrator (CA)

- 集成的 请求/评估/扫描/批准 工作流程解决方案



- Enterprise Analyzer (EA)

- 大批量分析程序
- 基于命令式或脚本化扫描



- Library Auditor (LA)

- 在代码检入SCM时进行实时的分析
- 支持 Perforce, Rational ClearCase, Git, SVN, Visual SourceSafe



- Developer Assistant (DA)

- 在开发过程中对工作站上检测到的代码进行实时分析

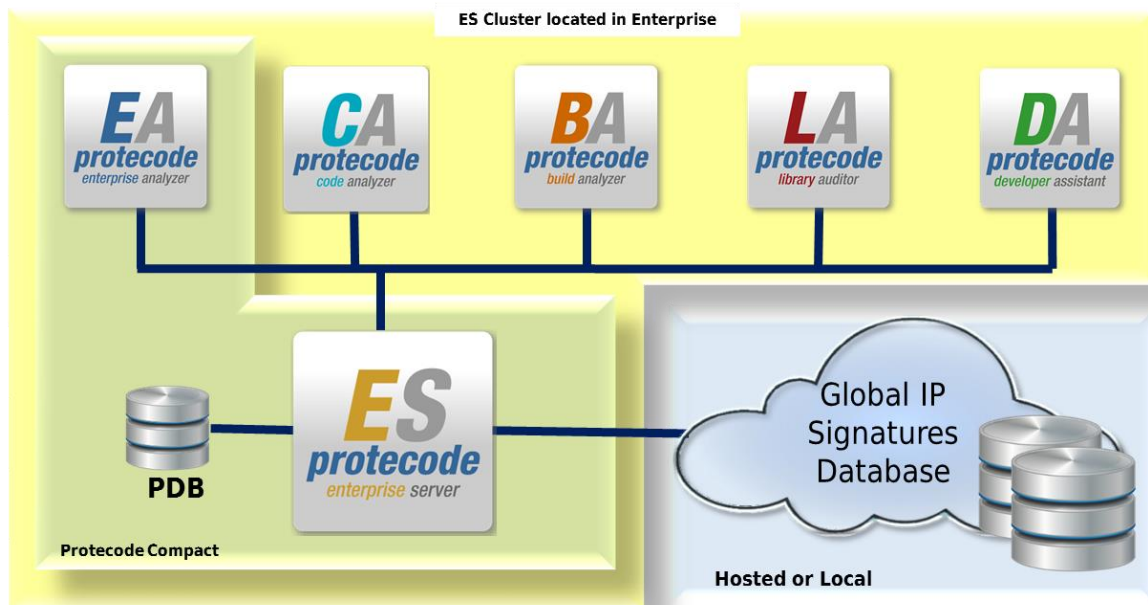


- Build Analyzer (BA)

- 在版本构建过程中进行实时分析



Products – 可扩展的架构



Enterprise Server

- 核心的知识产权分析引擎, 在程序间共享
- 管理软件物料清单 (代码数据库), 项目, 组, 用户和策略

IP Signatures Reference Database

- 引用的(开源)代码签名和源代码
- Global或者本地的
- 不会有源代码从组织泄露出去

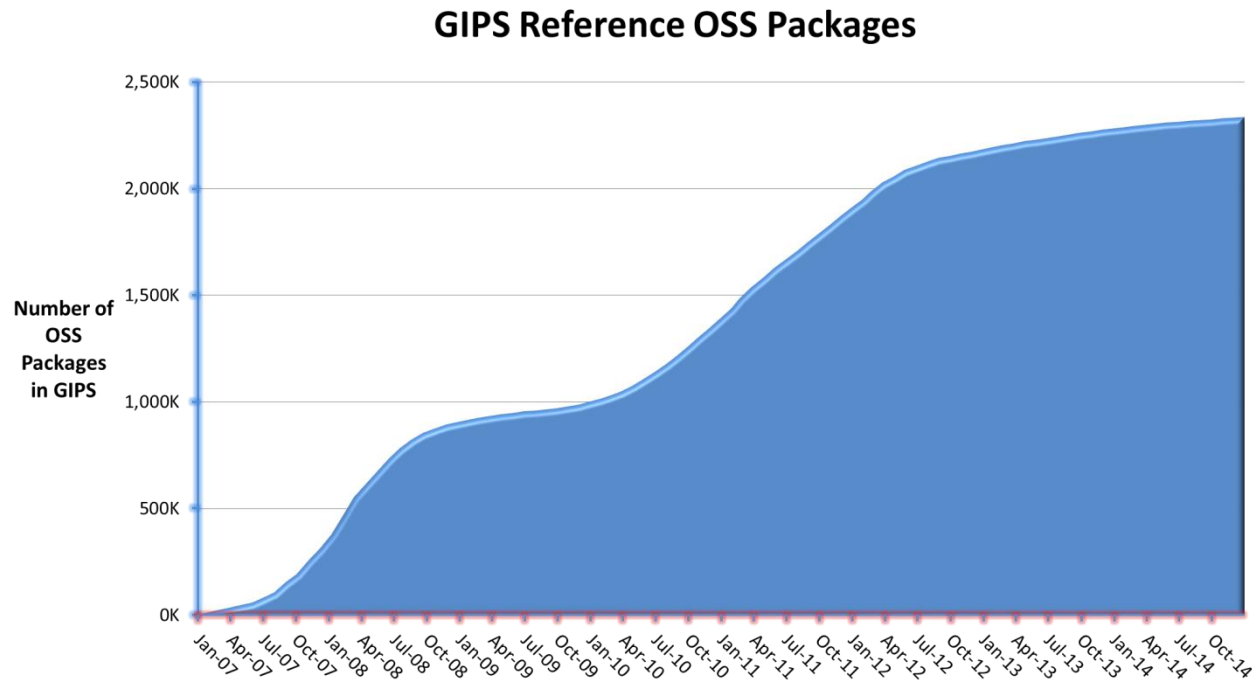
Applications

协同工作, 共享物料清单信息

- Enterprise Analyzer (EA) – 批量化代码分析
- Code Administrator (CA) – 包 预批准 流程
- Build Analyzer (BA) – 构建时进行合规性检查
- Library Analyzer (LA) – 实时分析
- Developer Assistant (DA) – 基于开发桌面的实时分析

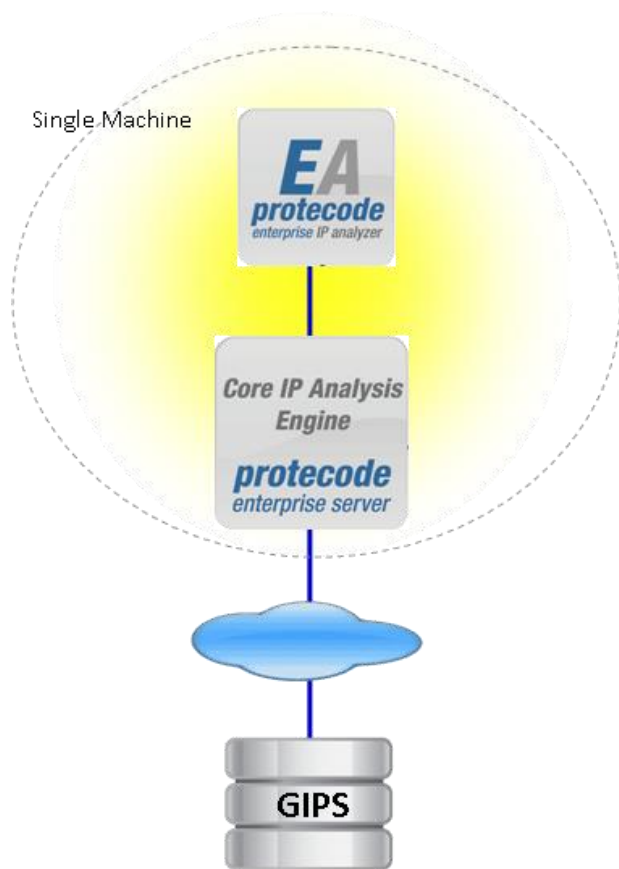
- Protecode Compact™: combined EA+ES

GIPS/EIP S

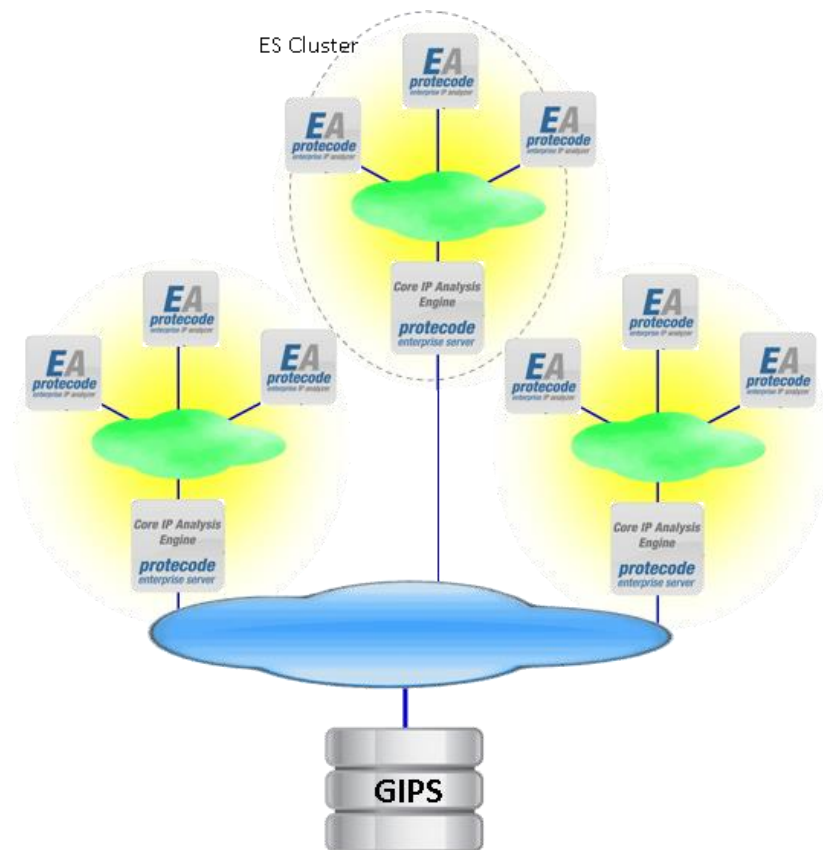


- **Global or Enterprise IP Signatures (GIPS/EIPS) Database**
 - Holds Protecode's reference public-domain software signatures
 - Crawling, download and indexing of more than 12000 sites
 - Sourceforge, Github, Google, OSOR, CodePlex, Freecode, Rubyforge, cpan, CodeProject
 - Eclipse, Apache, Mozilla, Microsoft, Universities, Public Research organizations
 - More than 2.6M open source packages, 750M files, 700k projects, >170B lines of code
 - Always evolving, GIPS updated 24 x7 available to all users instantly

可扩展性



单机模式运行



分布式、多地域式企业布局, 适用于多项目组

Q/A

