基于变异的SQL注入漏洞测试技术的研究与仿真实现

汪 诚 弘

院　　系：国家保密学院

专　　业：信息安全（保密技术）

学　　号：2011212116

指导老师：赵　靖　教授

June 10, 2015

论文结构

1. 概述
2. 基于变异的测试用例生成技术
3. 自适应随机选择方法
4. 仿真测试

What is SQLi?

What is SQLi?

交通部违章车辆号牌登记程序代码

mysql =
'INSERT INTO TABLICE.ILLNO (NO, Min, Max) VALUES ('
+var1+','+var2+','+var3+')'

e.g. *var1 = ZU666, var=120, var=170,*
mysql =
'INSERT INTO TABLICE.ILLNO (NO, Min, Max) VALUES
(ZU666,120,170)'

SQLi Attack

mysql =
'INSERT INTO TABLICE.ILLNO (NO, Min, Max) VALUES (ZU666,0,0);
DROP DATABASE TABLICE'
–,120,170)'

研究背景

SQL Injection

SQLi（SQL injection）漏洞是WEB2.0和HTML5时代最危险地WEB应用程序漏洞，该漏洞主要由于数据库驱动的WEB应用程序对用户输入检查不严格，导致用户可以注入SQL命令从而任意执行恶意代码。

- 高威胁性，5年OWASP威胁榜榜首
- 普遍性和频发性，NGS公司安全白皮书显示，世界几乎没3分钟就会发生一次SQLi攻击

国内外研究现状

国际领域研究现状

目前国际领域的研究主要形成了4种主流的SQLi防御理论：动态监测（Runtime Monitor），静态分析（Static Analysis），混合防御（Hybrid Approach），漏洞测试（Vulnerability Testing）。目前国际领域公认的最可靠最高效的防御方式是**漏洞测试**的方法。

Method	**Evaluation**
RM	监测开销巨大,程序可用性降低（Zhang et.al.）
SA	误报率高，信息收集不完全（William el.al.）
HA	实现困难，过程繁琐
VT	测试效率相对较低，测试执行时间长

Table: SQLi防御机制比较

研究内容

集中解决的问题

（1）测试用例状态空间爆炸，测试漏洞的测试用例往往有上万条

（2）有效测试用例分布稀疏，即测试用例的有效度低

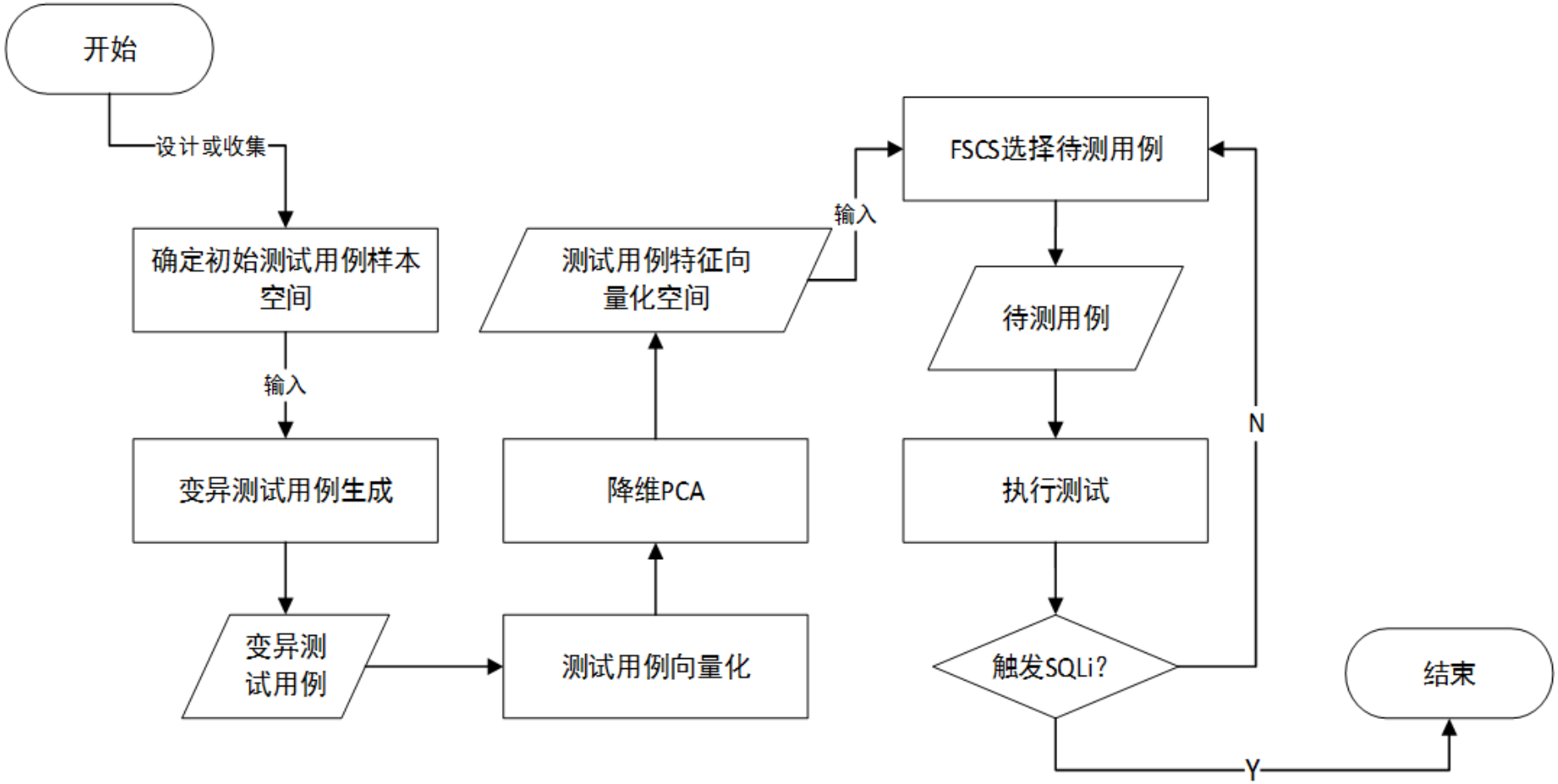
（3）执行测试的开销较高，执行每一条测试用例必须需要人工参与

提出的解决方案

Do Fewer Search Faster

MU4SQLi

MU4SQLi


开始
设计或收集
确定初始测试用例样本空间
输入
变异测试用例生成
变异测试用例
测试用例向量化
降维PCA
测试用例特征向量化空间
输入
FSCS选择待测用例
待测用例
执行测试
触发SQLi?
N
Y
结束

Mutation Based

Test Case Generation

变异测试用例生成技术

假设程序P是待测程序，测试用例空间为$T = \{t_1, t_2, ..., t_n\}$，我们将其设为初始状态空间。定义变异操作符空间$MO = \{mo_1, mo_2, .., mo_k\}$表示所有的变异操作符。我们将$T$中的每一个测试用例与$MO$中的每一个变异操作符结合，生成不重复的测试用例空间T'的过程叫做变异测试用例生成过程。

MU4SQLi中的变异操作符

序号	操作符名称	描述	类别
1	MO_or	在源输入后添加 OR 语句转义	转义操作符
2	MO_and	在源输入后添加 AND 语句进行转义	
3	MO_semi	在源输入后添加混合式语句进行转义	
4	MO_par	利用插入语，括号等截断原始语句	截断操作符
5	MO_cmt	在源输入中添加注释，将源语句截断	
6	MO_qot	利用单引号，双引号等闭合符号将语句直接闭合	
7	MO_wsp	将空格进行各种形式的转码	混淆操作符
8	MO_chr	对单引号，双引号等特殊符号进行转码	
9	MO_html	利用 HTML 转码方式进行转码	
10	MO_per	利用百分制形式转码输入	
11	MO_bool	布尔型表达式重写	

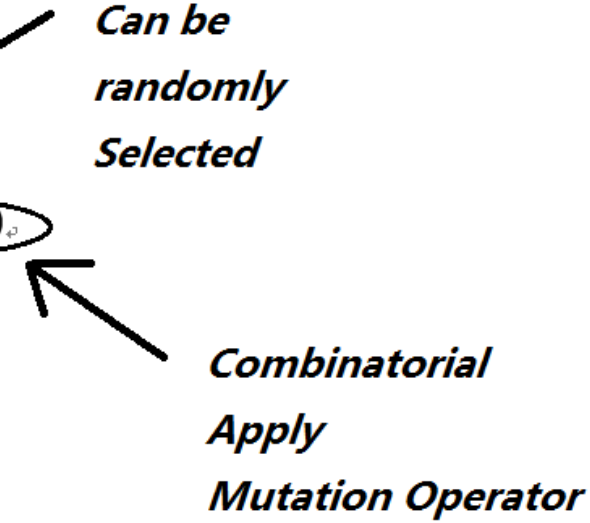
MU4SQLi变异测试用例生成算法

Algorithm 3.1 Mutation based test case Generation Algorithm

Input A set of Legal input, I

Output Test case space, T

```
1:  T={}
2:  for each input in I do
3:      while not(max_tries) do
4:          t = (apply_MO(input))
5:          if t not in T then
6:              Add t into T
7:          end if
8:      end while
9:  end for
10: return T
```



MU4SQLi测试用例生成特点

- 测试用例充分度高，能够消除认为设计的不足
- 测试用例状态空间巨大，变异程度越高

Adaptive Random

Method Based Selection

SQLi有效测试用例分布

能够触发SQLi漏洞的测试用例往往都具有很高的字串相似度。即从字串结构的角度去观察，有效测试用例呈现出聚类的特性。

No.	Successful Test Cases for CVE2014-3704
1	; or 1 = 1 waitfor delay '0: 0: TIME –
2	); or 1 = 1 waitfor delay '0: 0: TIME –
3	'; or 1 = 1 waitfor delay '0: 0: TIME –
4	"; or 1 = 1 waitfor delay '0: 0: TIME –
5	'); or 1 = 1 waitfor delay '0: 0: TIME –
...	...

Table: Successful Test Cases for CVE2014-3704

解决方案

我们提出了一种基于自适应随机**Adaptive Random** 思想的SQLi漏洞待测用例选择技术,该技术能够快速收敛到有效测试用例。下面给出具体的实施步骤:

Step 1 利用TF-IDF技术提取测试用例的特征向量(eq 3.1).

Step 2 利用PCA方法进行测试用例向量降维(eq 3.2-3.11).

Step 3 利用Cosine距离定义测试用例距离测度(eq 3.12 3.13).

Step 4 利用FSCS（Fixed Size Candidate Set）算法进行待测用例筛选.

MU4SQLi的FSCS 算法– $S_{distance}$ 选择算法

$E = \{e_1, e_2, e_3, ..., e_f\}$ 已执行测试用例集
$C = \{c_1, c_2, c_3, ..., c_\kappa\}$ 候选集合

而待测用例c_h的选择需满足以下的条件: $For\ all\ j\ \in\ \{1, 2, 3, ..., \kappa\}$,

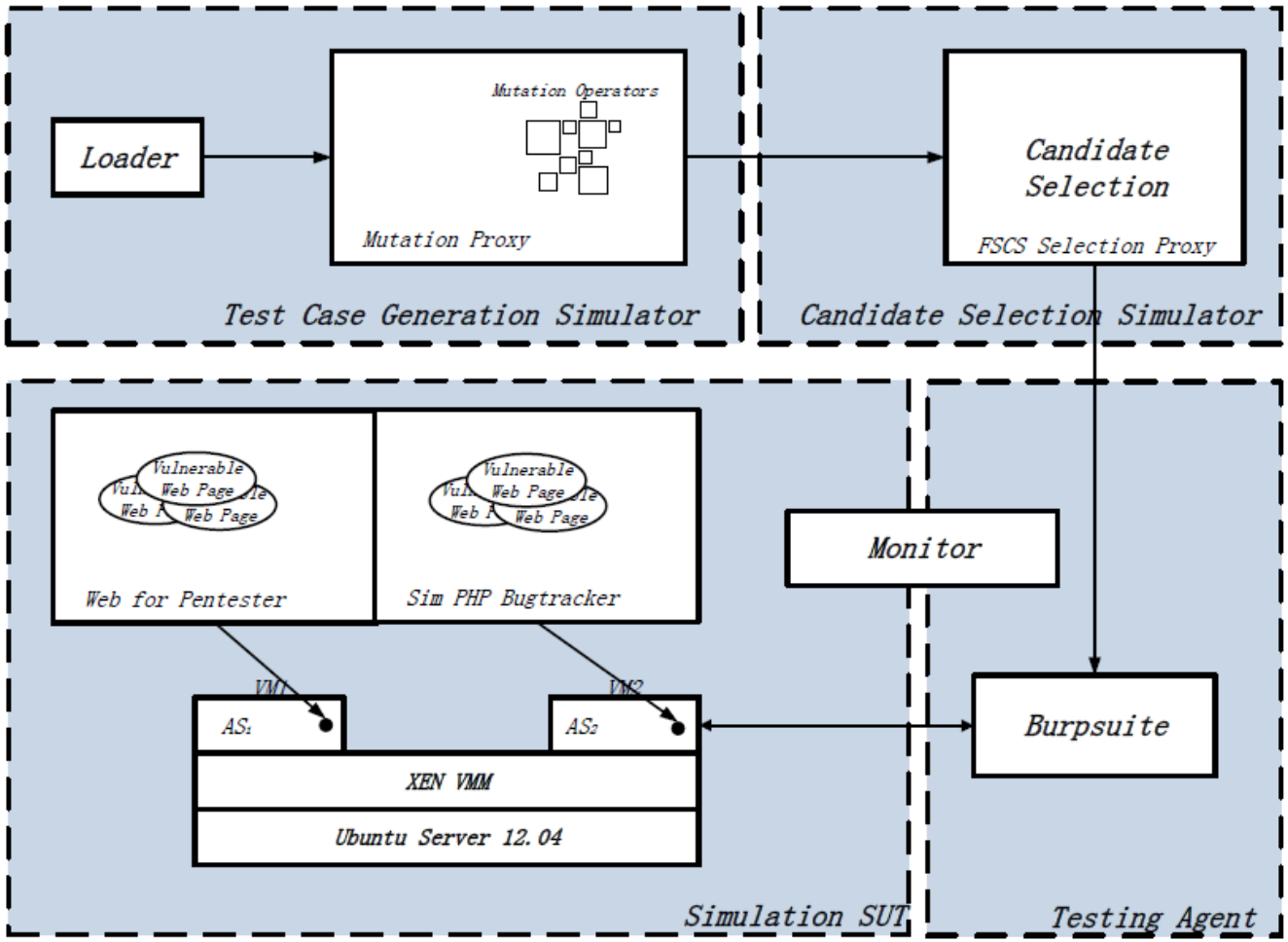
$$\min_{i=1}^{f} dis(c_h, e_i) \geq \min_{i=1}^{f} dis(c_j, e_i)$$

Simulation Evaluation

为了验证我们设计的MU4SQLi测试框架的测试用例有效度和测试效率，我们需要进行实际的比对测试。将MU4SQLi与传统的测试方法进行比对，并且比较传统测试方法在测试有效度和测试效率上运行性能。由于实际环境下的测试对于资源的消耗非常巨大。并且，对于安全测试来说Test Oracle问题（Test Oracle Problem）仍是一个困扰安全测试的关键性问题。因此我们选择利用仿真测试评估MU4SQL i

仿真平台搭建

我们的仿真平台如下图所示：



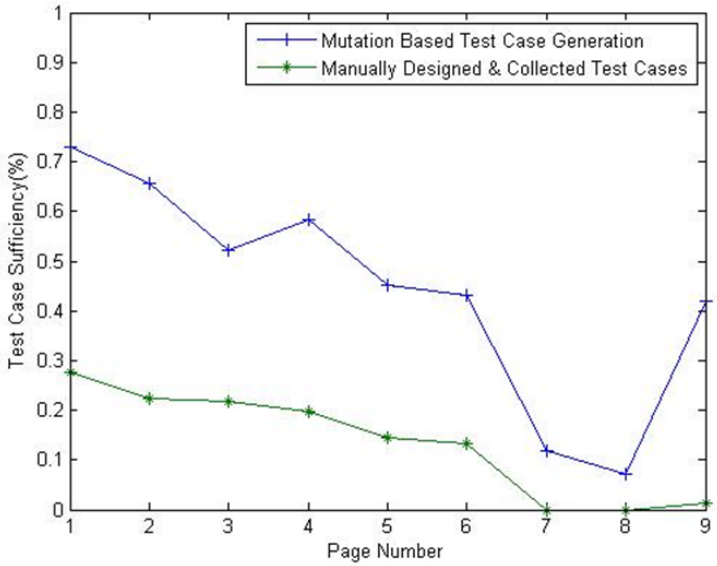
仿真平台搭建

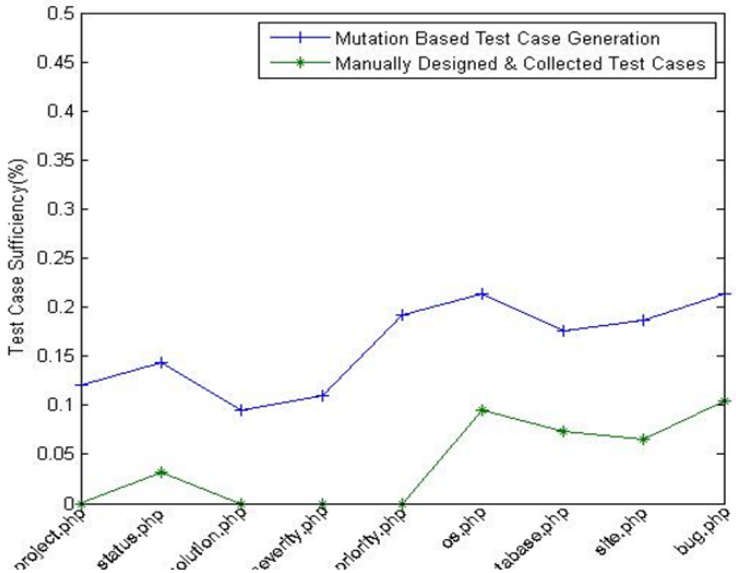
下表给出了仿真平台中各仿真模块和仿真SUT的具体配置:

序号	名　　称	描述	仿真实例	
			MU4SQLi	TT
1	测试用例生成器	模拟测试用例生成	Mutation	Fuzzdb
2	待测用例选择题	模拟待测用例选择	ART	RT
3	测试用例代理	模拟用户请求 web 页面	Burpsuite	
4	仿真 SUT-1	模拟漏洞系统	Web for Pentester	
5	仿真 SUT-2	模拟漏洞系统	Sim_PHP_Bugtracker	
6	判定器	模拟人工判定	GreeSQL	

测试用例有效性仿真结果

测试用例有效度仿真测试结果:

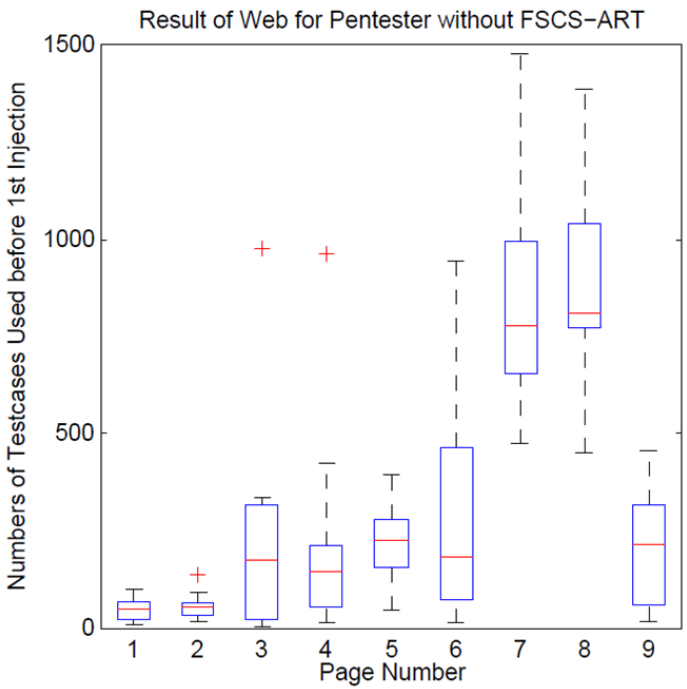


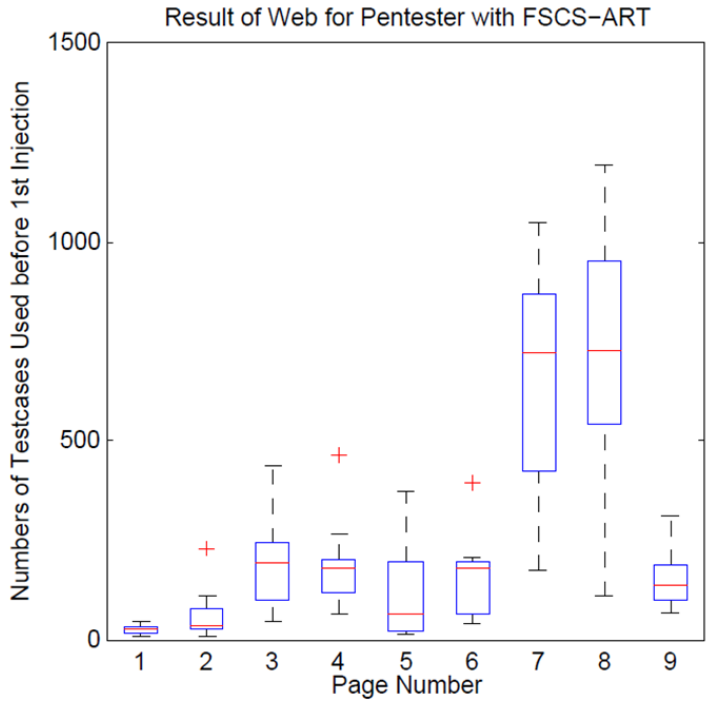


上图中左边图为Web for Pentester测试平台右边为SPB测试平台。图中绿色的线代表传统测试，蓝色的线代表基于变异的测试用例生成技术。很明显，变异生成的测试用例有效度远高于传统测试模式。

测试执行效率仿真对比测试W4P

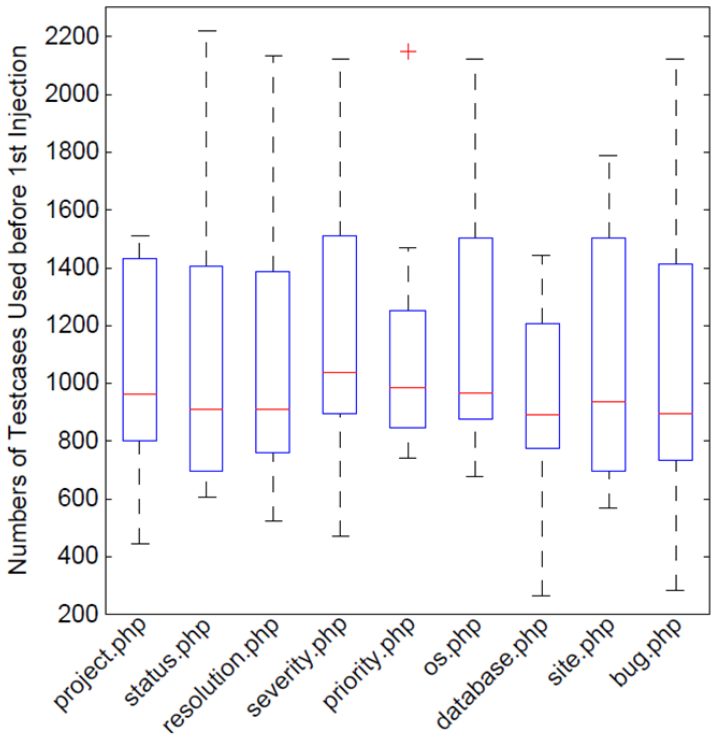
测试执行效率仿真对比测试W4P:

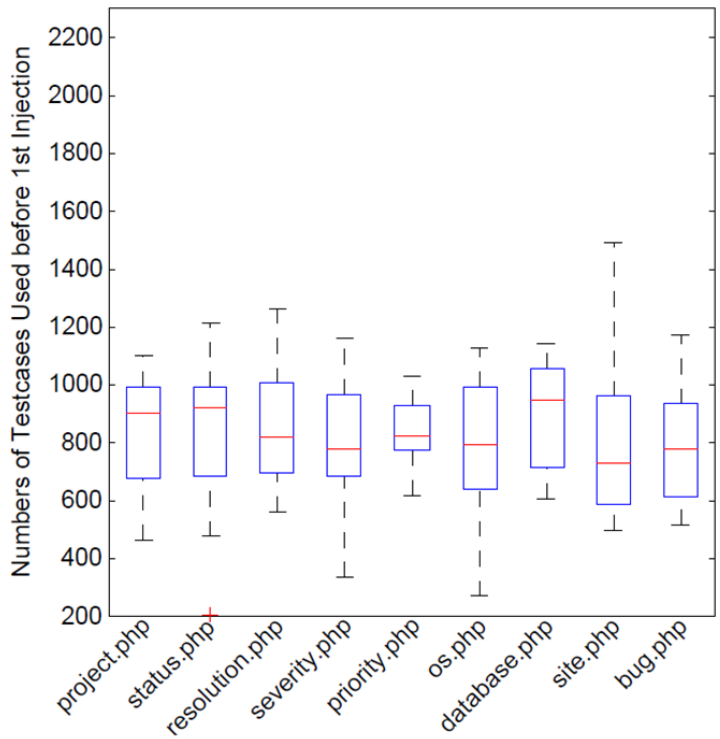




测试执行效率仿真对比测试SPB

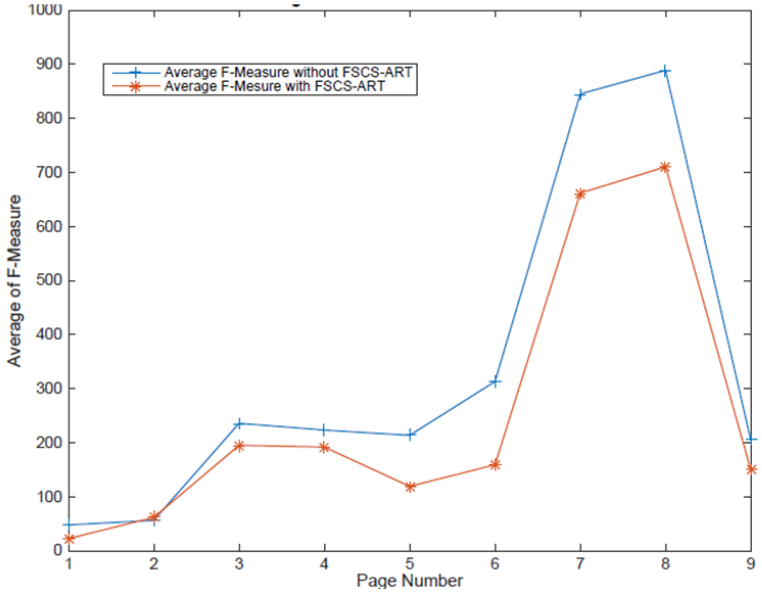
测试执行效率仿真对比测试SPB:

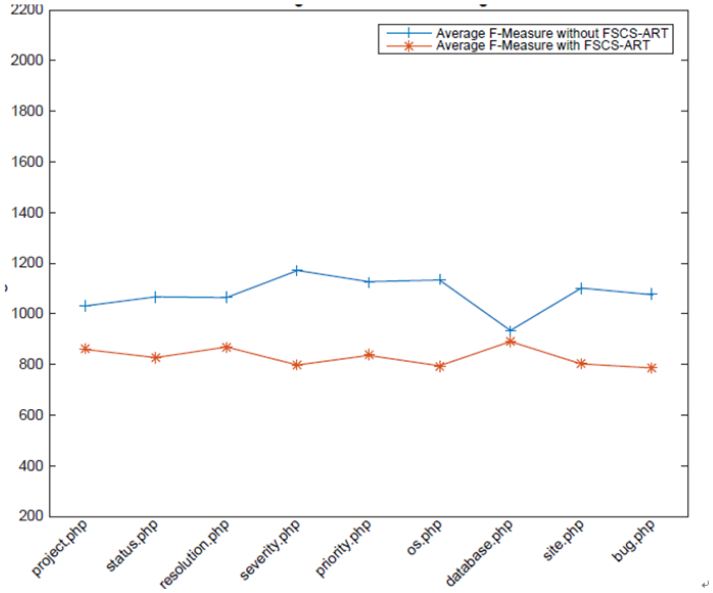




测试执行效率仿真对比测试AVG-Fmeasure

测试执行效率仿真对比测试AVG-Fmeasure:





在执行效率方面，自适应随机的方法也非常有效，从图中可以看出，我们的方法较传统测试方法提升平均20%-30%

Thank You

基于变异的SQL注入漏洞测试技术的研究与仿真实现

汪 诚 弘

院　　系：国家保密学院

专　　业：信息安全（保密技术）

学　　号：2011212116

指导老师：赵　靖　教授

June 10, 2015