Web 应用防火墙：
先打补丁，后提问题

Jonathan Werrett

Trustwave, SpiderLabs

jwerrett@trustwave.com

+852 6081 1508

OWASP

2011年11月8日

The OWASP Foundation

http://www.owasp.org

大纲

- Web应用防火墙
- 虚拟补丁
- Web应用示例
- 建立虚拟补丁
- SQL注入式攻击挑战赛的结果

Web应用防火墙

- 专用于Web应用层的安全设备
- 提供对内容的特定保护
- 可以是硬件或软件

优点

- 拥有对Web的高等级的“知识”
- 集中控制
- 成熟的防逃避机制

缺点

- 治标不治本
- 极度明确
- 无法解决业务逻辑和其他类似缺陷

Web应用防火墙

不使用WAF的理由

- ~~在相关标准中提及（比如：PCI DSS中的要求6.6）~~
- ~~为了避免主动测试~~
- ~~为了避免应付开发人员~~
- ~~你的审计人员/厂商告诉你的~~

使用WAF的理由

- 将安全与开发功能分离
- 最小化暴露时的空窗期
- 为很多应用提供“基础安全”

虚拟补丁

- 解决WAF层的具体漏洞
- “及时打补丁”

好处

- 及时打补丁
- 灵活性
- 可量测性
- 可处理遗留的代码
- 可处理外包的代码
- 减少暴露的漏洞
- 为“带外数据”打补丁
- 补丁可用性
- 减小对开发人员的依赖
- 避免“重复创建”补丁

ModSecurity

- 开源的Web应用防火墙
- 免费使用
- 最大的安装量
- 许多成熟的特点

http://modsecurity.org/

创建虚拟补丁 – 关键步骤

准备

- 确保已开始运行ModSecurity!
- 明确建立角色
- 搭建一个合适的测试环境

确认 & 分析

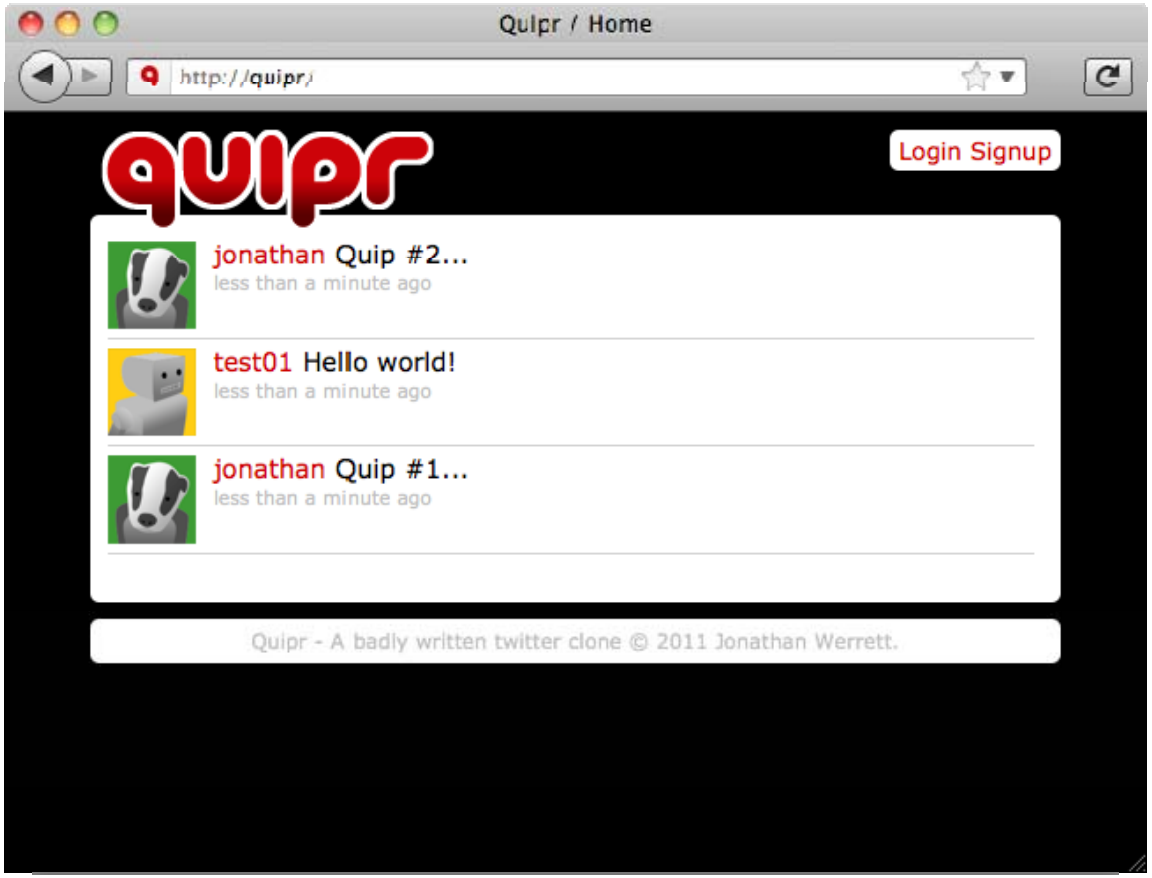
- 来源的数量(主动的评估，漏洞的通知)
- 确认关键特征。 白名单还是黑名单方法?

部署 & 测试

- 确保它不会阻止合法流量

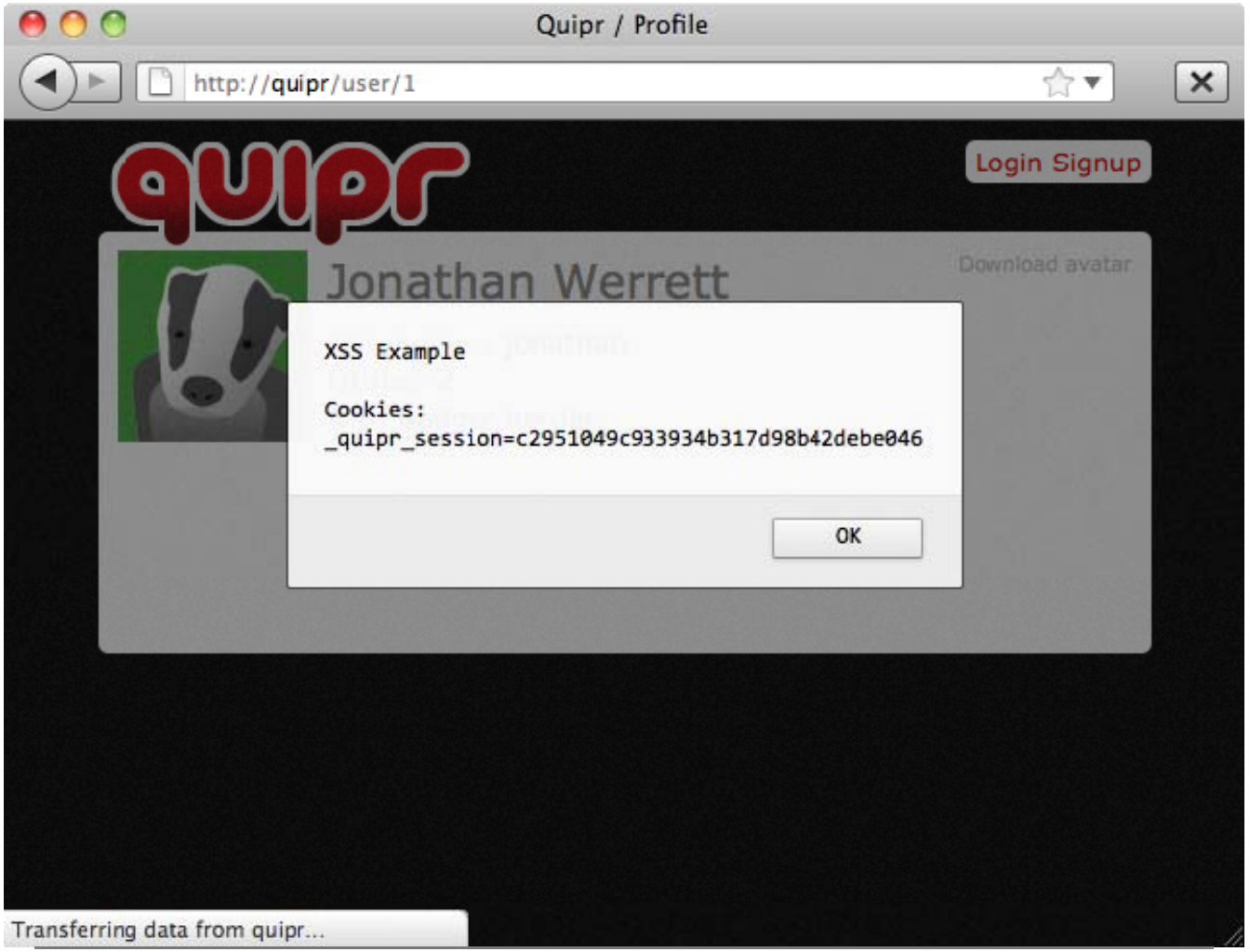
Web应用示例

http://quipr/



创建虚拟补丁 - 示例

跨站脚本攻击



创建虚拟补丁 – 示例

跨站脚本攻击

- 接受user[bio]参数的“白名单”

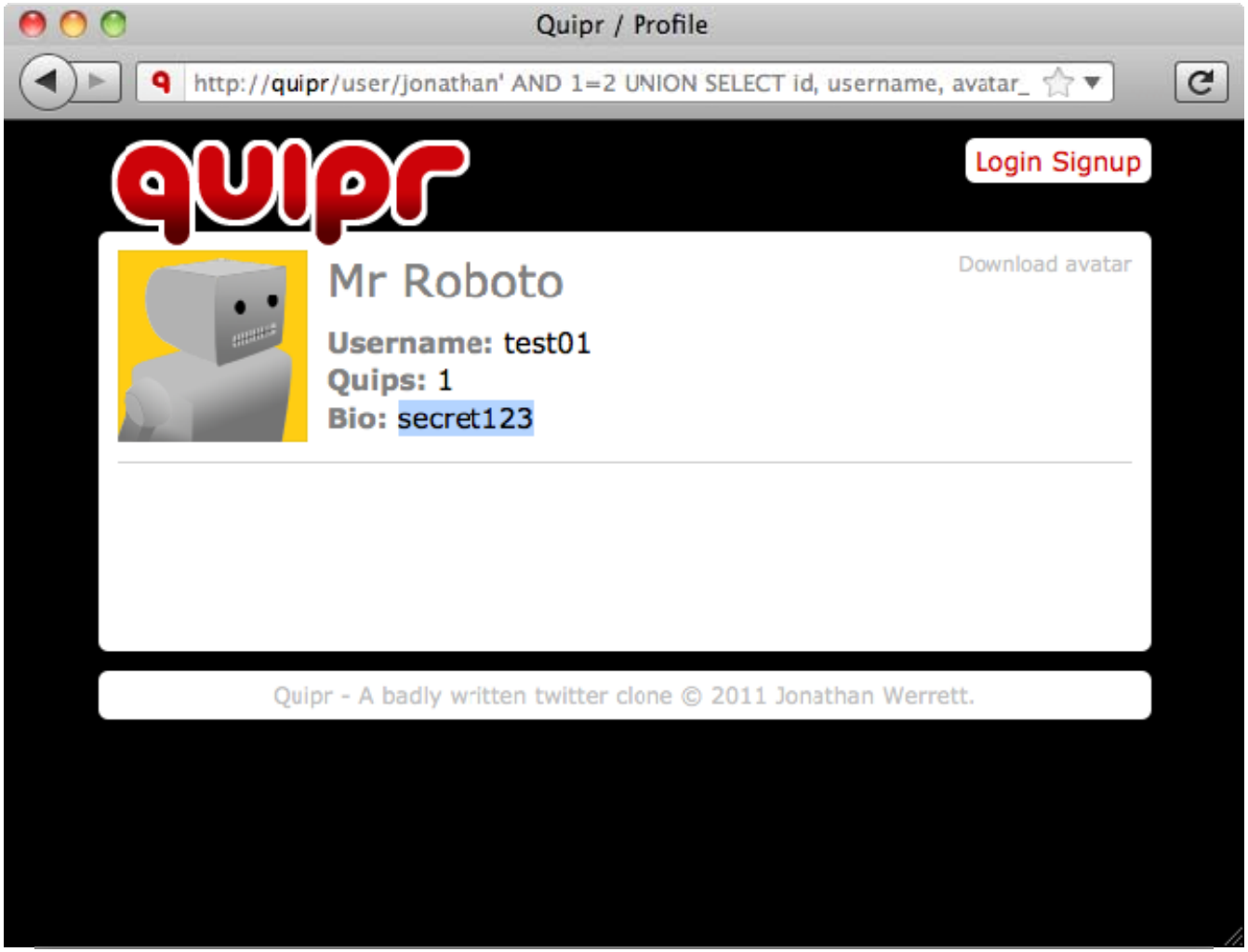
```
<Location /user/>
SecRule ARGS_POST:user[bio] "!^[\w\. ]*$"
   "phase:2,id:00001,t:none,t:urlDecodeUni,t:lowercase"
</Location>
```

- **接**受：文字、空格、破折号和句话
- 阻止：其他所有，包括标点符号<>$()‘”;

演示

创建虚拟补丁 – 示例

SQL注入式攻击



创建虚拟补丁 - 示例

SQL注入式攻击

- 最好的方式是向我们对XSS那样采用“白名单”

```
<Location /user/>
SecRule REQUEST_FILENAME "!^[\\w]*$
   "phase:2,id:00001,t:none,t:urlDecodeUni,t:lowercase"
</Location>
```

演示

创建虚拟补丁 – 示例

SQL注入式攻击

- 但是，我们还可以利用OWASP常用规则集
- 许多针对各种问题的通用规则
- 行之有效的和全面综合的
- 仅SQL注入式攻击就有179次测试
- 复杂的评分过程，而非直接的匹配比较

演示

创建虚拟补丁 - 示例

跨站请求伪造

- 为每个用户设置一个唯一的令牌

```
SecRule STREAM_OUTPUT_BODY "@rsub s/<\/body>/
<input type=\"hidden\" id=\"mstk\” value=\"%{unique_id}\">
<script>$(function(){
 $('a').each(function(){
   $(this).attr('href',this.href+'?tk='+$('#mstk').val());
 });
});
<\/script><\/body>/"
"phase:4,t:none,nolog,pass,
setsid:%{REQUEST_COOKIES._QUIPR_SESSION},
setvar:session.csrf_token=%{UNIQUE_ID}”
```

创建虚拟补丁 – 示例

跨站请求伪造

- 阻止没有令牌的请求

```
<LocationMatch "^/(user|quips)/”>

SecRule &ARGS:tk "!@eq 1"
"phase:2,t:none,log,deny,setsid:%{REQUEST_COOKIES._QUIPR_S
ESSION},msg:'No CSRF Token Detected.'”

SecRule ARGS:tk "!@streq %{session.csrf_token}"
"phase:2,t:none,log,deny,
setsid:%{REQUEST_COOKIES._QUIPR_SESSION},
msg:%{session.csrf_token}"
</LocationMatch>
```

演示

ModSecurity SQL注入式攻击挑战赛的结果

- 650位参与者
- 测试了OWASP ModSecurity 的核心规则集
- 4个厂商的Demo站点
  (Acunetix, Cenzic, HP, IBM)
- 9位“胜利者”
- 对核心规则集进行了改进

结果

- 建立黑名单非常困难
- WAF增强了“抵抗入侵”的能力，但无法做到 “彻底防止入侵”

总结

- 虚拟补丁有助于快速确保安全
- 减少了漏洞的暴露
- 为开发人员留下时间和空间以探寻最好的解决方法
- 使安全集中化，并提供一个总体的安全基础

其他阅读材料

- ModSecurity – http://modsecurity.org
- OWASP ModSecurity 的核心规则集
  https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project
- ModSecurity SQL注入式攻击挑战赛
  http://blog.spiderlabs.com/2011/07/modsecurity-sql-injection-challenge-lessons-learned.html