CodePecker®
安 全 解 决 方 案

- 源代码缺陷分析

CodePecker 源代码缺陷分析软件

产品白皮书

2013 年 09 月

目录

1. 产品概述 1

1.1. CodePecker 产品 1

1. 2. 竞争优势 1

1. 3. 市场定位 1

2. CodePecker 的主要功能 2

2. 1. 安全缺陷检测 2

2. 2. 代码质量检测 3

2. 3. 代码缺陷类型定制 6

2. 4. 图形化结果展示和缺陷定位追踪管理 7

3. CodePecker 的特点和优势 9

4. 系统结构 11

4. 1. 系统架构示意图 11

4. 2. 服务器硬件要求 12

4. 3. 服务器软件要求 12

1. 产品概述

1.1. CodePecker 产品

CodePecker 是 CodePecker 公司采用业界领先的源代码静态分析技术开发的一款针对源代码缺陷进行静态分析检测的产品，是国内第一款成熟的源码缺陷分析产品。它能够高效的检测出软件源代码中的可能导致严重缺陷漏洞和系统运行异常的安全问题、程序缺陷，并准确定位告警，从而有效的帮助开发人员消除代码中的漏洞、减少不必要的软件补丁升级，为软件的信息安全保驾护航。

1. 2. 竞争优势

与国际上其它同类产品相比，CodePecker 产品具有很多突出的特征：

1) CodePecker 支持的语言种类多，能够分析 Java、Jsp、Php 等编程语言编写的代码；其中，在 CodePecker 最具代表性的 Java/Jsp 语言分析方面，能够对共 25 大类、169 种缺陷类型进行代码安全和质量检测。
2) 能够全面的发现软件代码中的缺陷，这其中包括软件安全漏洞，也包括软件代码质量问题，还能够发现编程中违反编程规则的情况；
3) 提供友好的图形分析界面，简化了缺陷分析操作和流程；
4) 支持分析百万行级别的源代码；
5) 快速的分析检测缺陷，检测结果的低误报率、低漏报率。
6) （即将提供）与多种主流 IDE 开发环境的集成（如 Eclipse 插件）。

1. 3. 市场定位

目标客户：大中小型企业、政府机构、军队

适用行业：电信、电力、金融、证券、IT 行业、政府事业机关、军队、其他企业

© 2013 CodePecker

2. CodePecker 的主要功能

2.1. 安全缺陷检测

软件安全性是软件产品质量中一个非常重要的方面，因为代码安全漏洞导致的网络安全事件越来越多，也越来越受重视。CodePecker 软件的安全漏洞分析部分是基于市场领先的缺陷检测能力，从已有的缺陷检测功能分离出来单独的安全漏洞检测模块。能够检测到的安全漏洞类型有 8 大类 30 小类，比较全面的覆盖了常见的安全漏洞类型，如表 2-1 所示：

表 2-1 CodePecker 检测缺陷类型

安全缺陷分类	安全缺陷类别细化
跨站脚本问题（XSS）	存储型跨站脚本
	反射型跨站脚本
数据注入问题	不可信数据存入可信存储
	数据注入
	SQL 注入
	SQL 查询注入
	LDAP 过滤被污染
未验证用户输入问题	邮件操纵
	HTTP 响应分割
	日志操纵
	数据污染
	数据污染进入本地代码
进程和路径注入问题	命令注入
	路径执行注入
	环境变量注入
	文件名操纵

© 2013 CodePecker

	路径操纵
	临时文件路径操纵
拒绝服务问题	整数溢出
	数组索引拒绝服务
	数组长度拒绝服务
	临时文件未及时删除
弱加密问题	硬编码密码
	空密码被使用
	弱密码被使用
	不安全的随机函数
信息泄露问题	敏感信息泄露
	文件名泄露
	SVN 泄露
Struts 框架安全问题	当前 Struts 版本存在漏洞

2. 2. 代码质量检测

代码自身的健壮性和可维护性是衡量应用系统质量的一个重要技术指标。由于人为因素而导致代码设计逻辑不合理，代码质量差，导致系统运行时潜在风险。为了规避这些风险，往往要对代码进行人工复审（Code Review），但是这种代码复审周期较长，覆盖点不够多，同时会消耗大量人力，如何规避代码质量风险并且有效的降低代码复查成本也是一个值得关注的问题。CodePecker 在 Java 语言检测方面能够自动快速分析 Java 和 Jsp 代码，生成代码问题报告，准确定位问题点，能够发现多种缺陷类型，目前支持的代码质量缺陷类型包含 17 大类 139 小类，如表 2-2 所示：

表 2-2 代码质量缺陷列表

代码质量缺陷分类	代码质量缺陷类别细化

© 2013 CodePecker

克隆问题	非 final 的 clone 方法
	没有实现 Cloneable 接口
	没有定义 clone 方法
	非 final 类没有定义 clone 方法
线程和同步问题	锁中调用了 sleep
	锁中调用了 wait
	锁中调用了 notify
	不一致的同步
	双检锁问题
	错误的多线程启动
	错误调用 wait 方法
执行性能问题	低效字符串比较
	空方法 finalize()未移除
	冗余的 BOOL 对象创建字串表达式
	不必要的 BOOL 对象创建 BOOL 表达式
	不必要的字符串对象创建来自字符串表达式
	不必要的空字符串对象创建
	不必要的 toString()方法
	使用相同锁对象
	同步静态方法使用相同锁对象
	字符串循环中调用 append
弱封装问题	方法 finalize()错误修饰符
	静态成员变量错误修饰符
	实例变量错误修饰符
	可变对像可能被修改

© 2013 CodePecker

	可变对象导致内部表示被暴露
	方法存储可变对象
	内置类定义
	最低的方法访问权限
	方法不是 private 的
	最低的字段访问权限
	字段能够被声明为 final
错误处理问题	空 catch 块
	不明异常
	捕获运行时异常
	在 finally 中返回
	未捕获异常
忽略返回值问题	可变对象调用方法的返回值被忽略
	新建对象被忽略
	返回值被忽略
	空对象迭代异常
	常量空指针引用
	变量空指针引用
潜在的运行时问题	返回值空指针引用
	集合返回值空指针引用
	集合类型转换异常
	集合 key 转换异常
	不同类的类型转换异常
	子类的类型转换异常
	潜在的集合并发修改异常
	尝试修改不可更改的集合
	和父类同名的公有成员

© 2013 CodePecker

	和父类同名的保护成员
	和父类同名的私有成员
可维护性问题	方法名首字母非大写
	冗余的类型转换
	不匹配的覆盖
	System.err 打印调试信息
	System.out 打印调试信息
	方法 System.gc()不期望被调用

2. 3. 代码缺陷类型定制

CodePecker 提供一百多种缺陷检测类型，同时对外提供了可选择缺陷检测类型配置操作（高级检测），用户可以根据安全需求，有针对性的选择缺陷检测类型；同时 CodePecker 也提供了多种默认缺陷检测类型（普通检测），涵盖了常见的缺陷种类。

© 2013 CodePecker

图 2-1 可配置缺陷检测类型

2. 4. 图形化结果展示和缺陷定位追踪管理

Codepecke 界面友好，操作简单，可根据业务需求来配置用户界面。检测完成后，可根据检测结果生成可视化缺陷展示图，从多个维度展示缺陷分布。

© 2013 CodePecker

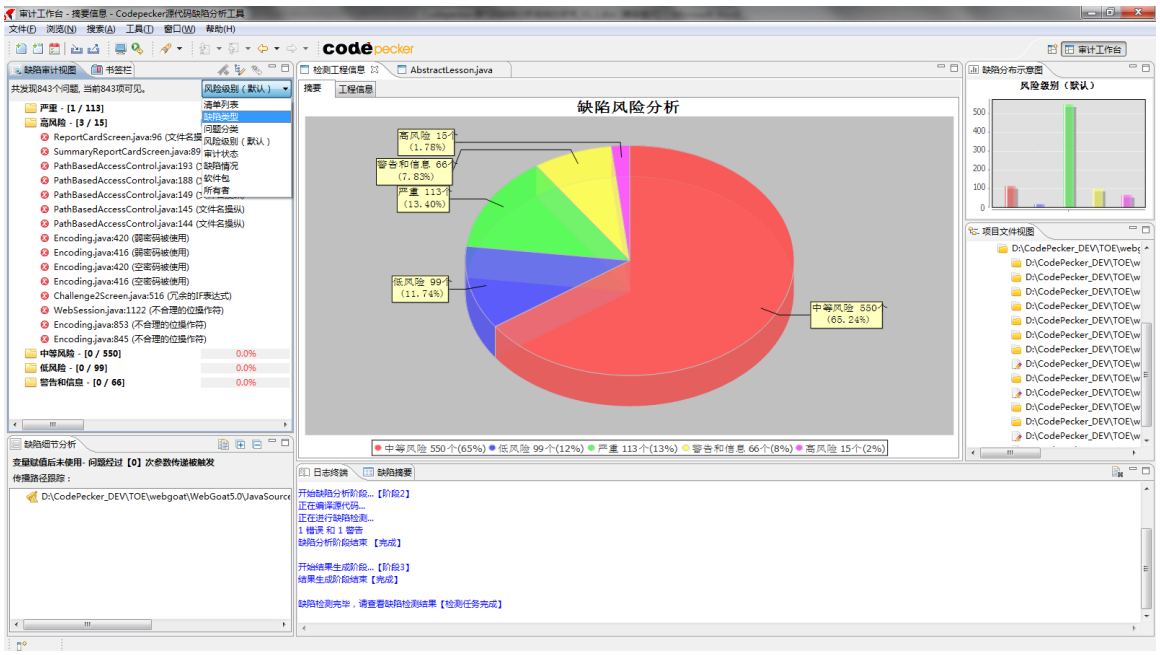


图 2-2 缺陷检测结果图

CodePecker 提供了缺陷追踪分析功能，用户可以追踪缺陷的传播，从而定位缺陷位置。在缺陷分析完毕后，可以对缺陷风险做评估和审核。CodePecker 还会对缺陷分析结果以及分析人员进行审计的结果做持久化保存，方便后续对缺陷的维护管理。

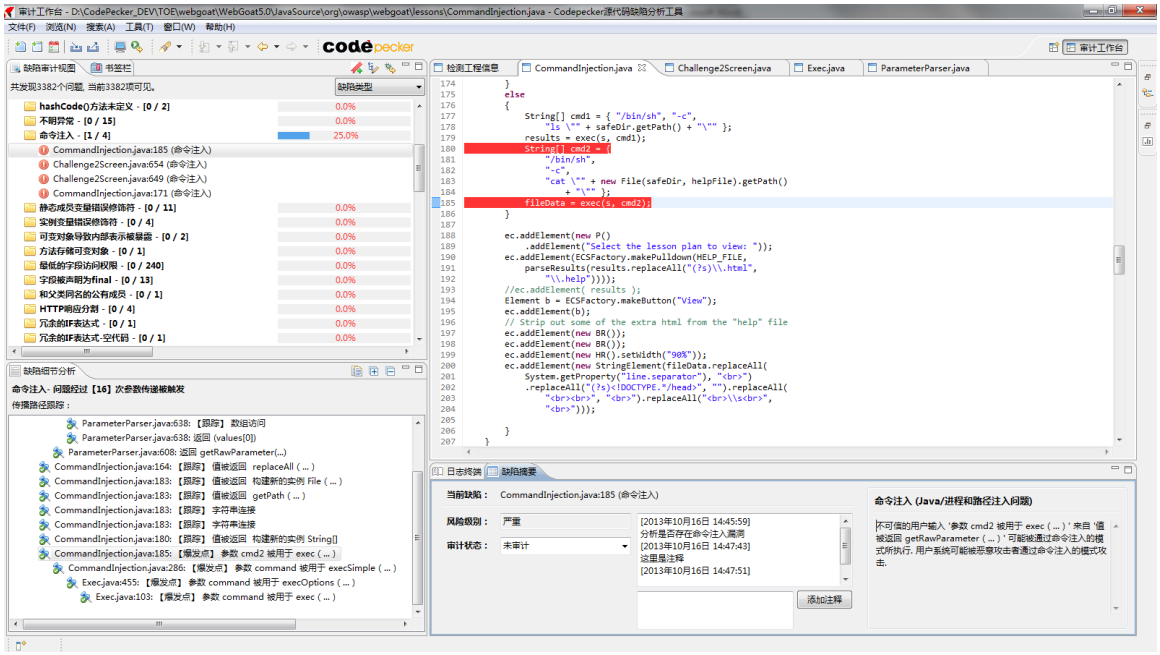
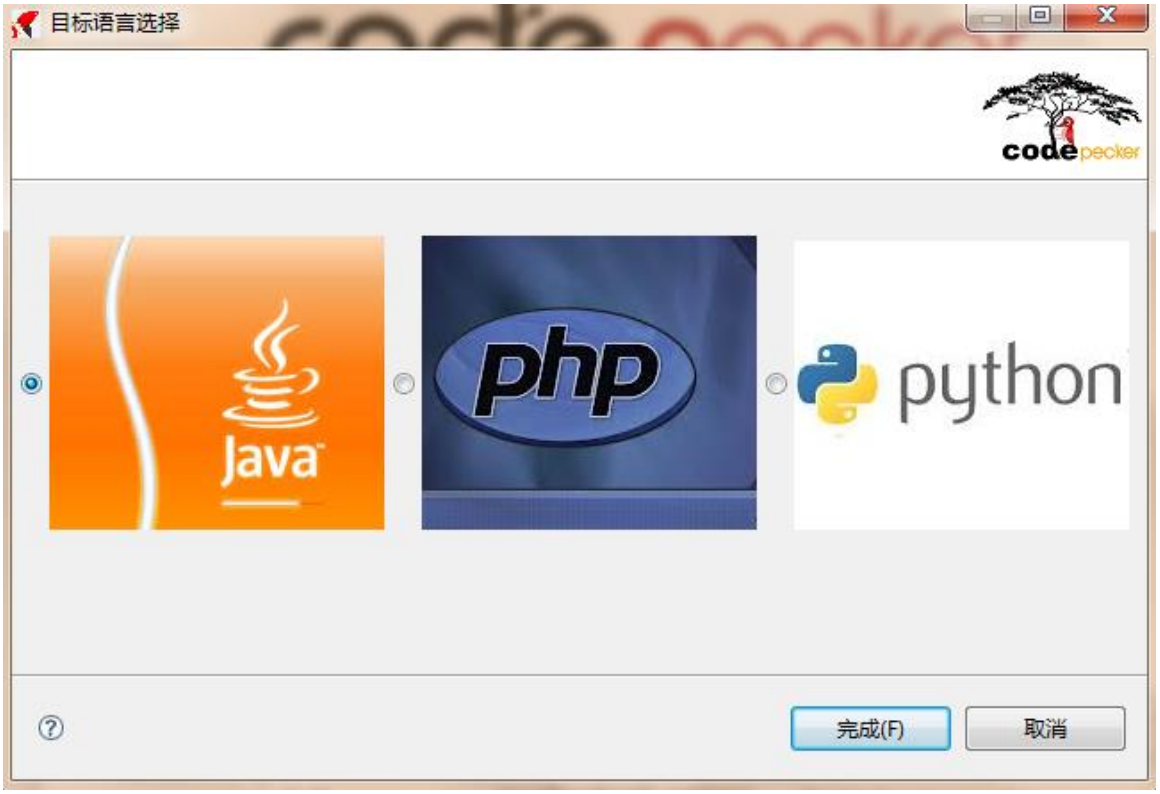


图 2-3 缺陷追踪管理图

© 2013 CodePecker

2.5. 更多语言支持

CodePecker 当前还支持对 PHP、Python 语言的检测。



3. CodePecker 的特点和优势

本系统的设计目标在于在用户系统上线之前尽最大可能的发现源代码中的安全隐患和代码质量问题，从多个视角深刻反映系统源代码的整体安全状况，对高危安全缺陷的分布、代码质量问题分布、缺陷的危害、缺陷信息细化等多视角信息进行了细粒度的统计分析，并通过柱状图、饼图等形式，直观、清晰的从总体上反映了代码缺陷分布情况，提供定位追踪代码中的问题，挖掘出系统中潜在的安全隐患，杜绝由于代码缺陷导致系统上线运行之后出现信息安全问题。

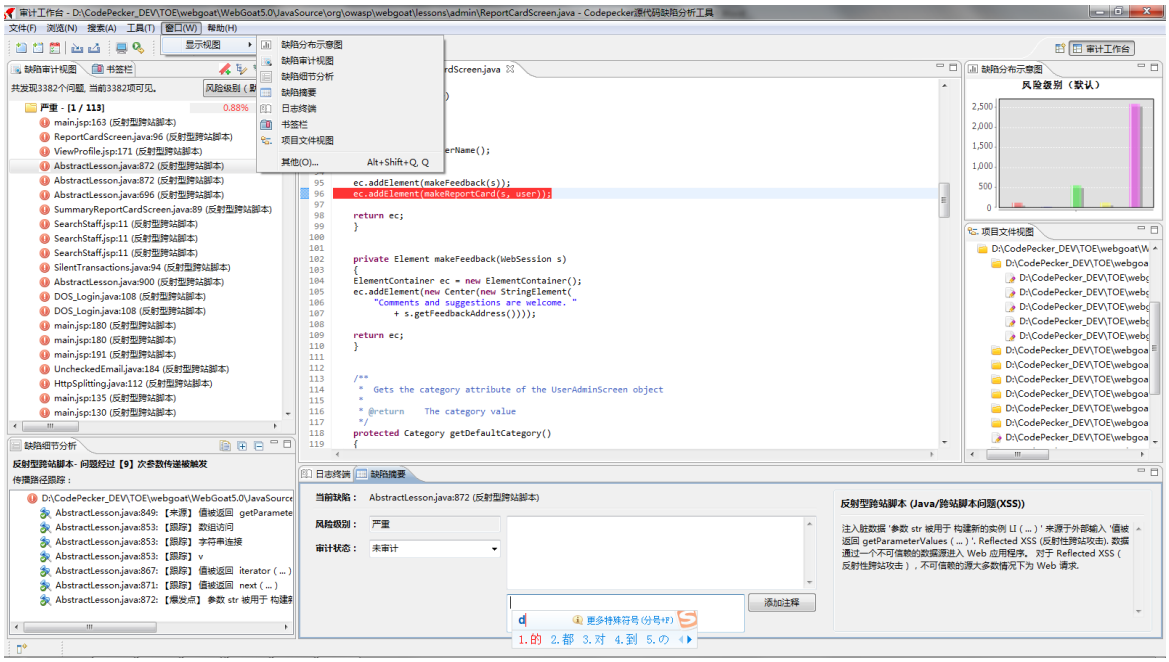
本系统的具体特点如下：

1） 友好的界面操作

CodePecker 在使用过程中，操作简单易用，不需要复杂的检测流程，检测结果简单明了，并有详细的

© 2013 CodePecker

缺陷分析，同时提供了缺陷分析的追踪定位，用户只需要简单的鼠标操作，就能够对缺陷传播定位。同时鉴于开发人员对信息安全知识的了解，在缺陷类型中会有详细的缺陷点评，即使开发人员不熟悉此缺陷类型，通过缺陷点评，也能迅速的了解掌握此类缺陷。



2） 支持检测多种缺陷和质量类型

缺陷检测覆盖常见的多种语言、多种缺陷类别，包括跨站注入、Sql 注入、拒绝服务等高危缺陷漏洞类型，也包括空指针引用，资源为释放、变量未初始化等代码质量缺陷类型。从多个维度全面覆盖代码安全问题，并积极更新最新研究结果和关注国内外最新安全研究动态，同步更新研究成果，保证缺陷知识库内容的覆盖广度和深度。

3） **自定义的缺陷类型检测规则**

系统本身提供了 25 个大类，169 种的缺陷类型检测，用户可以根据具体的需求对被检测代码做一个全量分析，也可以根据业务需求，针对自己系统关心的缺陷进行定制检测。如在大型应用系统中，存在各种级别的缺陷类型，检测结果可能偏多，会干扰错误排查，用户可只针对高危或者某几类类缺陷做有针对性的深度检测，只关注特定的缺陷类型，从而达到检测效果。

4） 高效快速的缺陷分析

通过优化的数据流分析技术、缺陷类型的智能识别、检测规则依赖关系等源码扫描技术的运用，再加上安全团队根据多年的源码检测经验和国内外安全信息缺陷结果，以及完善的缺陷检测规则，CodePecker 缺陷检测软件在源码检测速度和检测结果的准确性做到了一个很好的平衡，既保证了检测的速度，又保

© 2013 CodePecker

证了缺陷检测的质量。

5） 低误报率漏报率

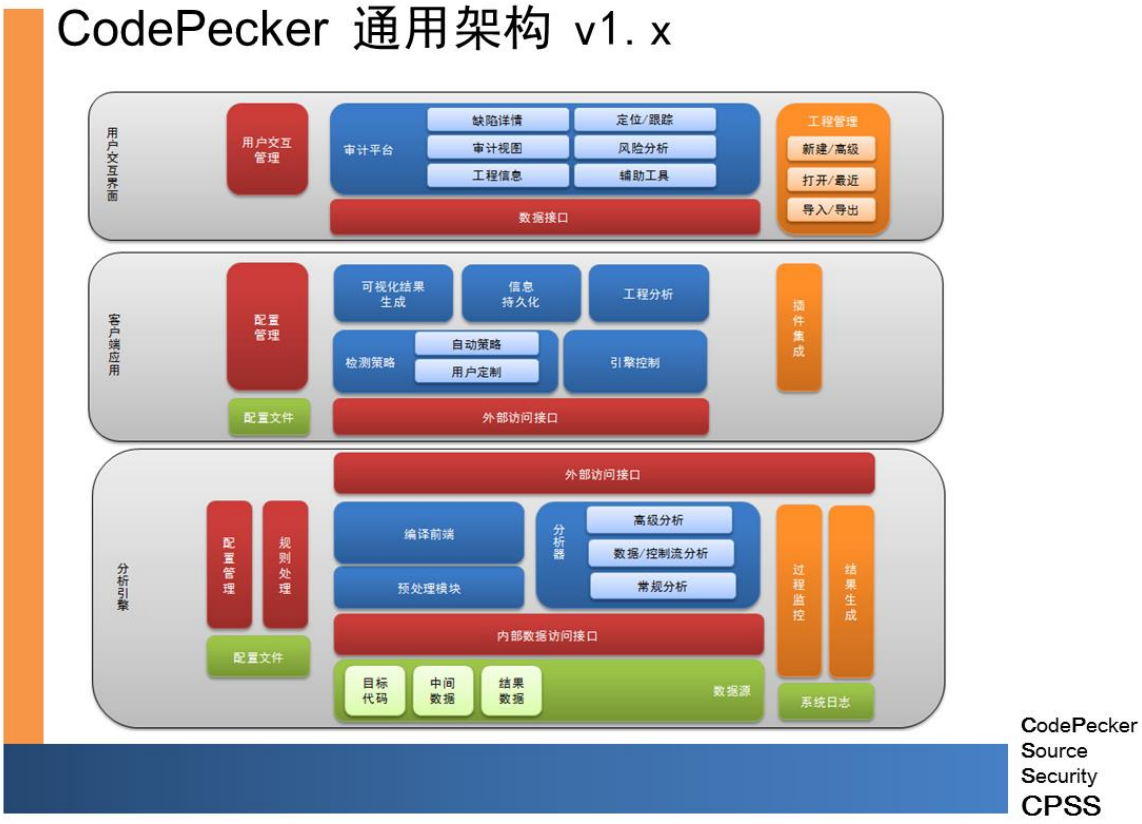
采用业内领先的深度缺陷扫描分析技术，CodePecker 软件对同样的目标系统进行检测时，能提供过程内（Intra-procedure）、过程间（Inter-procedure）等各种层次的分析，全面深入地开展缺陷检测，全面降低了检测结果中的误报率和漏报率，检测精度、准度高。

6） 权威、完备的缺陷分析知识库

团队成员有着多年的源码安全检测经验，依靠业内知名专业的安全团队的研究，CodePecker 漏洞知识库已包含多种语言多达数百条缺陷类型，每条漏洞都有详尽的描述和修补建议，同时积极与国际化接轨，大部分缺陷类型都可以映射到 CVE 和 OWASP 等权威国际安全组织公布的缺陷分类中。缺陷类型涵盖了常见操作系统、数据库、Web 工程和应用程序的绝大多数可以远程利用的漏洞以及本地安全漏洞。

4. 系统结构

4. 1. 系统架构示意图



© 2013 CodePecker

图 4-1 CodePecker 架构示意图

4.2. 服务器硬件要求

计算机处理器： Intel X86 系列或兼容 CPU，最低 PIV 2.0GHz 或相当，推荐 Xeon 2.0GHz * 2 或相当。

内存：最低 2GB，建议 4GB，推荐 8GB

硬盘：最低 IDE 80G * 1，推荐 SCSI 72GB * 2

网卡：最低 10/100 自适应网卡 * 1，推荐双网卡

4.3. 服务器软件要求

组件	版本
操作系统	Windows XP（所有版本及 Service Pack 包） Windows Vista（所有版本及 Service Pack 包） Windows Server2000/2008（所有版本及 Service Pack 包） Windows 7/8（所有版本及 Service Pack 包） 注：支持 32、64 位操作系统
JDK	JDK（1.5 及其以上的所有版本）

© 2013 CodePecker