

鸿渐 SAST 静态代码分析工具 V5.0 用户使用手册

北京鸿渐科技有限公司



目	录
Ħ	X

	1.	1.	购买	工具简要说明	3
	1.	2.	概述		3
	1.	3.	工具	特点	3
	1.	4.	检测	能力	5
		1.4.	1.	安全漏洞	5
		1.4.	2.	运行时缺陷	5
		1.4.	3.	编码规则	6
		1.4.	4.	框架和库	7
		1.4.	5.	编译器	7
2.		安装	部署		8
	2.	1.	系统	环境要求	8
	2.	2.	Linux	系统环境下安装	9
	2.	3.	WIND	ows 系统环境下安装1	1
		2.3.	1.	安装 JDK81	1
		 2. 3. 2. 3. 	1. 2.	安装 JDK81 工具部署启动1	1 2
		 2. 3. 2. 3. 2. 3. 	1. 2. 3.	安装 JDK8	1 2 2
		 2. 3. 2. 3. 2. 3. 2. 3. 	1. 2. 3. 4.	安装 JDK8	1 2 2 2
		 2. 3. 2. 3. 2. 3. 2. 3. 2. 3. 	1. 2. 3. 4. 5.	安装 JDK8	1 2 2 3
	2.	 2. 3. 2. 3. 2. 3. 2. 3. 4. 	1. 2. 3. 4. 5. 其他	安装 JDK8 1 工具部署启动 1 Server 配置变更 1 启动 1 license 自助获取 1 系统环境下安装 1	1 2 2 3 3
3.	2.	 2.3. 2.3. 2.3. 2.3. 4. 工具 	1. 2. 3. 4. 5. 其他	安裝 JDK8 1 工具部署启动 1 Server 配置变更 1 启动 1 license 自助获取 1 系统环境下安装 1 1 1 1 1 工具部署启动 1 1 1	1 2 2 3 3
3.	2.	 2.3. 2.3. 2.3. 2.3. 4. 工具 1. 	 1. 2. 3. 4. 5. 其他 项 访问 	安装 JDK8 1 工具部署启动 1 Server 配置变更 1 启动 1 license 自助获取 1 系统环境下安装 1 1 1	1 2 2 3 3 3 3
3.	 2. 3. 3. 	 2.3. 2.3. 2.3. 2.3. 4. 工具 1. 2. 	 1. 2. 3. 4. 5. 其 使 访 数 6. 7. 7. 8. 7. 7.	安装 JDK8 1 工具部署启动 1 Server 配置变更 1 启动 1 license 自助获取 1 系统环境下安装 1 1 1 (1 1 1<	1 2 2 3 3 3 4
3.	 2. 3. 3. 	 2.3. 2.3. 2.3. 2.3. 4. 工具 1. 2. 3.2. 	 1. 2. 3. 4. 5. 其 使 访 数 1. 	安裝 JDK8 1 工具部署启动 1 Server 配置变更 1 启动 1 license 自助获取 1 系统环境下安装 1 1 1 (1 2 2 3 3 3 4 4
3.	 2. 3. 3. 	 2.3. 2.3. 2.3. 2.3. 4. 工具 1. 2. 3.2. 3.2. 	 1. 2. 3. 4. 5. 其 使 访 数 1. 2. 	安装 JDK8 1 工具部署启动 1 Server 配置变更 1 启动 1 license 自助获取 1 系统环境下安装 1 1 1 (次長 1 如目数据 1 秋陷数据 1	1 2 2 3 3 3 3 4 4 5
3.	2. 3. 3.	 2.3. 2.3. 2.3. 2.3. 4. 工具 3.2. 3.2. 3.2. 	 1. 2. 3. 4. 5. 其 使 访 数 1. 2. 3. 	安裝 JDK8 1 工具部署启动 1 Server 配置变更 1 启动 1 license 自助获取 1 系统环境下安装 1 1 (火表 1 项目数据 1 4 人员绩效数据 1	1 2 2 3 3 3 3 4 4 5 5

3.	3.1.	新建检测项目1	5
3.	3.2.	检测队列2	3
3.	3.3.	检测项目编辑2	3
3.	3.4.	缺陷结果查看24	4
3.	3.5.	代码度量20	6
3.	3.6.	函数清单20	6
3.	3.7.	检测历史2	7
3.	3.8.	报告导出	8
3.4.	项目	组23	8
3. 4. 3. 5.	项目 全局	组	8 9
 3. 4. 3. 5. 3. 	项目 全局 5.1.	组	8 9 9
 3. 4. 3. 5. 3. 3. 	项目 全局 5.1. 5.2.	组	8 9 9
 3. 4. 3. 5. 3. 3. 3. 	项目 全尾 5.1. 5.2. 5.3.	组	8 9 9 0
 3. 4. 3. 5. 3. 3. 3. 3. 	项目 全尾 5.1. 5.2. 5.3. 5.4.	组 24 配置 29 后缀名配置 29 项目组配置 30 检测配置 30 重要缺陷等级配置 30 重要缺陷等级配置 30	8 9 0 0
 3. 4. 3. 5. 3. 3. 3. 3. 3. 	项目 全局 5.1. 5.2. 5.3. 5.4. 5.5.	组 2 配置 2 后缀名配置 2 项目组配置 3 检测配置 3 重要缺陷等级配置 3 特殊函数配置 3	8 9 0 1 1

1.1. 购买工具简要说明

用户在华为云商店购买产品后,需联系本公司相关人员索要相关软件工具安装包,同时会有相关技术人员远程支持工具部署,部署成功后,下载许可证界面 sn 文件并发送给我司技术人员开通 license (一个 sn 文件对应一个 license),用户上传 license 后方可使用工具。

1.2. 概述

鸿渐静态代码分析工具(简称鸿渐 SAST)是基于北大和中科院等高校优秀毕业 生多年的研究成果、应用多种国际先进的代码分析技术,自主研发的源代码检测工 具。在不改变组织现有研发流程的前提下,与源代码管理系统(Git、SVN等)、漏洞 管理系统(如 Jira、Bugzilla、禅道等)、持续集成工具(如 Jenkins、禅道)等无缝对接, 将源代码检测流程融入企业的研发流程,实现源代码编码规则检测、运行时缺陷检 测、安全漏洞检测、度量统计、编译不通过检测、逆向架构图自动生成等功能,并 提供检测器自主研发接口,帮助组织快速构建源代码安全自主检测体系和能力。

1.3. 工具特点

SAST 静态代码分析工具相比于其他国外静态分析工具的主要特点如下:

 ① 自主研发,技术自主可控:研发团队专注于软件安全、供应链安全和漏洞 自动挖掘,拥有数十项发明专利和数十篇国际顶级会议论文。

② 更多的缺陷检测类型和精度保障:支持更多的运行时缺陷的检测,相比于 国外领先工具具有相似的误报率(15%左右)及更低的漏报率(相对漏报率约10%,能 够发现其他国外工具不能发现的或遗漏的缺陷)。

③ 详细的数据仪表及人员绩效分析:工具提供检测任务、项目组、缺陷、人

第3页共34页

员等维度的数据分析仪表,例如:项目缺陷分布、重要缺陷密度、缺陷类型及等级分布、缺陷趋势、人员绩效数据等,帮助管理者快速决策关键应用及人员管理的规划。

④ 强大的语法词法分析:支持在代码编译不通过的情况下进行质量缺陷/安
 全漏洞检测、分析,并提示编译缺少的文件,但仍不影响整体检测,最大限度降低
 测试过程成本,让用户专注于漏洞和缺陷。

⑤ 精准的跟踪编译机制: SAST 静态代码分析工具采用跟踪编译形式,通过 对代码进行跟踪编译,再对编译后的结果进行检测,使缺陷检测结果更加准确,误 漏报更低。

⑥ 反向架构图一键查看:产品提供多种反向代码架构图,例如:函数调用图 /被调用图、文件函数调用图、项目函数调用图、值跟踪图,帮助用户在脱离 IDE 的 情况下准确跟踪代码上下文,确认漏洞。

⑦ 多线程检测:由于 SAST 静态代码分析工具采用 B/S 部署模式,支持多用 户可同时进行检测,检测后将缺陷结果在服务器集中查看。

⑧ 更完善的缺陷确认机制:在缺陷检测完毕后,能够快速定位缺陷发生的行数,并且 SAST 静态代码分析工具提供变量的定义处、函数的定义处以及函数被调用点的自动定位等功能,并以统计图方式显示各等级缺陷。

⑨ 强大的适应性及快速大规模检测能力:能分析上千万行代码,效率达100万行/小时以上,并支持复杂的编译环境以及多种嵌入式的开发环境。

⑩ 国产环境适配经验丰富:产品已经参与过多个国产化适配的项目,并且获得了麒麟软件的桌面操作系统及高级服务器操作系统的 NeoCertify 认证。

第4页共34页

1.4. 检测能力

1.4.1. 安全漏洞

SAST 静态代码分析工具支持国际上通用的 OWASP 的百余种安全漏洞,包括 但不限于以下缺陷类型:

未经验证的用户输入	弱加密算法	DNS 欺骗	
命令行注入	LDAP 注入	SQL注入	
使用不安全的函数	管道劫持 字节序列错误		
线程安全	缓冲区溢出	资源泄漏	
密码权限	拒绝服务	隐私泄漏	
内存污染	非法计算	安全编程函数	

1.4.2. 运行时缺陷

SAST 静态代码分析工具支持国际上 CWE、CVE 的百余种缺陷,包括但不限于以下缺陷类型:

数组越界	变量未初始化引用	空指针解引用	
整数溢出	错误的隐式转换	不可达代码	
内存泄漏	内存重复释放	无效跳转与条件	
无效跳转	循环跳出条件异常	字符串差一错误	

不正确的指针判空	强制类型转换可能丢失精度	资源使用后未释放	
缓冲区溢出	内存释放后返回	使用释放后的内存	
非法计算	异常迭代器操作	互斥锁问题	

1.4.3. 编码规则

SAST 静态代码分析工具支持国内、国际的千余种编码规则,包括但不限于以下缺陷类型:

类别	规则名称	数量
国内	GJB 5369 2005 C 语言安全子集	138
Шгэ	GJB 8114 C/C++语言编程安全子集	204
	MISRAC 2004 英国汽车工业软件可靠性联会 C语言标准	123
国际	MISRA C++ 2008 英国汽车工业软件可靠性联会 C++语言标准	112
	MISRA C++ 2012 英国汽车工业软件可靠性联会 C++语言标准	128
CERT	CERT 出版的最新 C 编码规则集,ISO-17961 标准	45
CLKI	CERT 出版的最新 Java 编码规则集	
软件工程化	921 C-2007 载人航天 921 工程 C 语言安全子集	
	合计	1052

1.4.4. 框架和库

SAST 静态代码分析工具支持 80 多种 C/C++和 Java 等语言的框架和库,主要 内容如下:

Hibernate	IBatis	MyBatis	
Struts	Spring Framework	Spring Boot	
Axis	JEE	Restlet	
Apache Wicket	Apache Velocity	Apache Hadoop	
JSF	javax.websocket	OWASP ESAPI	
qt4	qt5	boost	

1.4.5. 编译器

SAST 静态代码分析工具支持多种内置编译器和自定义编译器,主要内容如下:

GCC	Arm_5.1.6	ARMCC	
ARMCC_504u2	c2000_6.2.7	c5500_4.4.1	
c6000_7.4.8	ccs	gcc-arm-none-eabi-4_7-2013q3	
MinGW	msp430_4.3.3	mssdks	
terboc3	tornado	vs6	
vs2010	vs08	其他自定义编译器	

2. 安装部署

2.1. 系统环境要求

注:产品带狗,建议使用物理机安装产品。

代码规模 类型	10 万行以内	10 万行-50 万行	50 万行-100 万行	100 万行-1000 万行	
建议物理可用内存	8G-16G 或以上	16G-32G 或以上	32G-64G 或以上	64G-128G 或以上	
预计占用内存	4G-8G 或以上	8G-16G 或以上	16G-32G 或以上	32G-100G或以上	
操作系统	Windows 系列/LINUX 系列(Ubuntu、中标麒麟、RedHat)/UNIX(Solaris)				
CPU	Intel 系列/AMD 系列/龙芯系列:其中 Intel/AMD I52.5GHz 或以上,龙芯 0.8GHz-1.0GHz				
硬盘容量	100G以上(视用)	户的检测代码量而定,	建议预留 50G 可用	空间)	

注: 以上内存要求是机器的物理可用内存, 可在操作系统中查看系统可用物理

内存。

① windows 平台下物理可用内存查看:可以在资源管理器中查看,如图所示:

🗐 Task Manager	A.M. A.M. 107	_	
Eile Options View			
Processes Performance App history Sta	rtup Users Details Services		
CPU 15% 1.69 GHz	Memory Memory usage		16.0 GB
7.7/15.9 GB (48%)			
Disk 0 (C:) SSD 1%		Γ	
Ethernet 以太网 4 S: 0 R: 0 Kbps			
Ethernet VMware Network Ad S: 0 R: 0 Kbps	60 seconds Memory composition		0
Ethernet VMware Network Ad	In use (Compressed) Ausilable	Sneed	2400
S: 0 R: 0 Kbps Wi-Fi S: 0 R: 0 Kbps	T.7 GB (607 MB) 8.1 GB Committed Cached 14.8/26.4 GB 7.6 GB	Slots used: Form factor: Hardware reserved:	2 of 2 SODI 145 MB
GPU 0 Intel(R) UHD Graphic 2%	Paged pool Non-paged pool 667 MB 896 MB		
Fewer details (W Open Resource	Monitor		

② LINUX 平台下物理可用内存查看:通过命令【free -n】查看可用内存,如 图所示:

For more details see free(1). defect@defect-VirtualBox:~\$ free								
	total	used	free	shared	buff/cache	available		
Mem:	7636988	1039392	3067596	22556	3530000	6273200		
Swap:	2097148	0	2097148					
defect@d	defect@defect-VirtualBox:~S							

2.2. Linux 系统环境下安装

此章节介绍 LINUX 下 SAST 静态代码分析工具安装过程,包含客户端、服务端,安装环境为 Ubuntu 18.04 中文版 32 位操作系统,32G 内存。

说明: SAST 静态代码分析工具安装需要临时获取 root 管理员用户权限,请安 装用户联系管理员获取临时权限,以下操作假设安装用户已经获取 root 权限。

产品安装步骤:

 1) 鼠标右击 SAST 静态代码分析工具安装文件,将其解压。进入解压后的 【setup】文件夹,在窗口上右键【在终端打开(T)】,打开终端窗口,运行 【chmod 777 -R*】,获取安装目录执行权限

- 2) 在终端窗口,运行【./startconfig-linux32.sh】,运行安装程序:
 - ① startconfig-linux32.sh为LINUX 32 位操作系统安装文件
 - ② startconfig-linux64.sh 为 LINUX 64 位操作系统安装文件
 - ③ startconfig-solaris.sh为 Solaris UNIX 32 位操作系统安装文件
 - ④ startconfig-solaris64.sh为 Solaris UNIX 64 位操作系统安装文件
 - ⑤ 安装帮助.txt,为安装帮助、编译分析项目助手文档
 - ⑥ 将激活码 sn.txt 放置到安装目录下即可完成安装

2.3. Windows 系统环境下安装

此章节介绍 Windows 下 SAST 静态代码分析工具安装过程,包含客户端、服务端,安装环境为 Win7 64 位操作系统,32G 内存。

2.3.1. 安装 JDK8

在"工具安装包"中找到 jdk-8u152-windows-x64.exe 并运行,按照默认配置安装即可。注:此配置是按照将 jdk8 设置为计算机全局设置来操作运行的,如果电脑中包含其他版本 jdk,为防止 jdk 冲突,可向相应技术人员获取运行时 jdk 的配置方式。

安装完成后配置环境变量:

① 我的电脑右键->属性->高级系统设置->环境变量->系统变量

② 新建:变量名:JAVA_HOME,变量名:C:\Program Files\Java\jdk1.8.0_261
 (按照实际 jdk 安装目录为准)

③ 编辑 Path: 增加;%JAVA_HOME%\bin; %JAVA_HOME%\jre\bin 编辑环境变量

	C:\Program Files\VanDyke Software\Clients\	新建(<u>N</u>)
	C:\Program Files (x86)\Common Files\Oracle\Java\javapath	
	%SystemRoot%\system32	编辑(F)
	%SystemRoot%	71074(5)
	%SystemRoot%\System32\Wbem	动地传(D)
	%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\	/刘元(世)
_	%SYSTEMROOT%\Svstem32\OpenSSH\	
	%JAVA_HOME%\bin	删除(<u>D</u>)
	%MAVEN_HOME%\bin	
	C:\Program Files\Git\cmd	
	C:\Program Files\dotnet\	上移(<u>U</u>)
	C:\Program Files\Microsoft SQL Server\130\Tools\Binn\	
	%JAVA HOME%\jre\bin	下移(Q)
	C:\Program Files\Microsoft SOL Server\Client SDK\ODBC\170\Tools\	

④ 保存后,打开 cnd 窗口,输入: java -version,出现版本信息,说明配置

成功(部分系统可能需要重启生效)



2.3.2. 工具部署启动

工具部署启动的准备工作:

- ① 创建项目主目录,随意位置,注意路径中不要包含中文和空格。
- ② 标准示例: D:\sast
- ③ 安装谷歌浏览器,本系统与谷歌浏览器兼容的最好。

2.3.3. Server 配置变更

拷贝 server 到主目录下,进入 server,修改 application.yml 配置文件,将第 80

行的路径改为本机当前路径:

/4	#目定义配置
75	□ sast:
76	🖨 analysis:
77	version: sast-1.0.0
78	url: http://sast.natapp1.cc
79	base:
80	path: E:/release/sast/parallel/server
81	project:
82	repo: \${sast.base.path}/data/sast-repo

2.3.4. 启动

在 server 目录下双击"start.bat"文件(或中文的"一键启动.bat"文件), 弹出的窗口不要关闭。窗口中出现下面字样说明启动成功:

an sast	-		×
2021-02-22 17:38:51.304 [main] INFO s.documentation.spring.web.readers.operation.CachingOperationNameGene	rator -	- Gene	ra ^
ting unique operation named: passwordChangeUsingPOST_1			
2021-02-22 17:38:51.305 [main] INFO s.documentation.spring.web.readers.operation.CachingOperationNameGene	rator :	- Gene	ra
ting unique operation named: resetPasswordUsingPOST_1			
2021-02-22 17:38:51.305 [main] INFO s.documentation.spring.web.readers.operation.CachingOperationNameGene	rator	- Gene	ra
ting unique operation named: tokenUsingPOST_1			
2021-02-22 17:38:51.306 [main] INFO s.documentation.spring.web.readers.operation.CachingOperationNameGene	rator ·	- Gene	ra
ting unique operation named: updateUserUsingPOST_1			
2021-02-22 17:38:51.306 [main] INFO s.documentation.spring.web.readers.operation.CachingOperationNameGene	rator	- Gene	ra
ting unique operation named: userInfoUsingGET_1			
2021-02-22 17:38:51.308 [main] INFO s.documentation.spring.web.readers.operation.CachingOperationNameGene	rator ·	– Gene	ra
ting unique operation named: userListUsingPOST_1			
2021-02-22 17:38:51.309 [main] INFO s.documentation.spring.web.readers.operation.CachingOperationNameGene	rator :	- Gene	ra
ting unique operation named: userListByDeptIdUsingGET_1	the Library	bl. co. bb	
2021-02-22 17:38:51.336 [main] INFO cn. redrocket. sast. web. listener. ProjectCheckStatusLinstener - 遭勁忍鍵	密赅策	做又羁	出田
極輝世州消壓」 鏟 网络射鍋估約 莆骨烷譯世州		All Vis Ma	14
2021-02-22 17:38:51.540 [main] INFO cn.redrocket.sast.web.listener.LoadFileSuffixListenerparser	config	弱机流	前開
	07.005		
2021-02-22 17:38:51.540 [main] INFO cn. redrocket. sast. SastWebApplication - Started SastWebApplication in	27.085	secon	ds
	sta o sta	C ME MA	
2021-02-22 17:38:51.598 [main] INFO cn. redrocket.sast.web.config.StartedUpRunner - mongo樂成唱拳視辭鈕之	用?获	€濮熙	
娴燥倾筠≪湄疆扮猊奶€娴媍傺垣佔娟奜			
Design of the second seco		- 1.9	
2021-02-22 17:35:31.025 [main] ERKOR CH. Fedrocket. sast. web. service. impl. ProjectGetCheckerServiceimpl - plu	gin co	m. naxs	LT.
eam inneedougs is null i 2001 09 17 20 50 50 [min] EDBOD on underslat and makes imil Desirate technology and a second second second second		. 1.9	
2021-02-22 17:35:31.025 [main] ERKOR Ch. Fedrocket. Sast. web. service. impl. ProjectGetGetGetGeterServiceImpl - plu	gin co	m. nəxs	UL ^C
eam. Lindsecougs is hull 2 2001-02, 22, 27, 29, 17, 29, 11 [main] INTO, on moducalist cost web convice sheet impl DetectorWhiteSourcesImpl _	± 01	見 がまか	CH I
2021-02-22 17:33:31.941 [marin] INFO Ch. Fedrocket. Sast. web. service. cneck. impl. DetectorwriteServiceImpl — 如 0.3b.进动时间的方面,在面积超级数据表示。	5 S	何 9年9	£
て相論取得無知為外で病與情勢的な			

2.3.5. license 自助获取

④ 获取 license:访问 SAST 应用页面,通过登录页"许可证设置"导出 sn.txt,

依据此文件向支持人员获取 license。

⑤ 导入 license: 获取 license 后,同样通过 SAST 应用页面的"许可证设置"
 导入 license。

2.4. 其他系统环境下安装

其他系统环境,例如银河麒麟、中标麒麟等环境,可凭借环境具体配置(内核、 CPU等)联系技术支持人员获取对应的安装操作手册。

3. 工具使用

3.1. 访问

确保服务是开启的,若需重新开启服务,则根据 "2.3.4 启动"章节内容进行操作。

访问服务端:打开浏览器后,在地址栏中输入以下地址及端口号,例: 【127.0.0.1:8000/】即可打开 B/S 主界面,如下图所示。默认登录用户信息:用户

第 13 页 共 34

h」 鸿渐科技

名: admin, 密码: 向指定技术人员获取。

上」鸿渐 SAST静态代码	科技 3分析工具	
A admin		
☐ ☐ 记住账号密码	许可证设置	
	 上) 次第 ある た あの は 本 の に は 能 の 報 し 、 た の の の の の の の の の の の の の	 FJ

- 3.2. 数据仪表
- 3.2.1. 项目数据

点击【仪表盘】一级菜单,默认界面是项目维度的数据分析汇总,可展示项目 相关的数据分析图:逾期未处理、重要缺陷密度、激活状态分布、缺陷分布、检测 次数趋势、项目规模排行,并且能够以项目组维度(项目组概念详见 3.4 章节)展 示数据分析,如下图所示:

○ 仪表盘 ■ 项目列表 器	组织管理 🚦 全局配置 📑 许可证	设置			②帮助 Aadmin
项目 缺陷 人员 ④ 项目批計 所有项目 ン 项目组统计 项目创建封阅: 2021-08-27 ~ 2021-09-27	☐ 语言: 全部 ∨ 展示范围:	top10 > 〇 刷新	项目总数:	12 缺陷总数:1201 平均缺陷密度:	117.28/KLoC 平均重要缺陷占比: 22.78%
逾期未处理 ③	逾期天数 2	重要缺陷密度 ③		激活状态分布 ①	未激活天数设置 2
00°0318856 60°0-02 60	ුව යු gtTest-01	100,000 80,000 40,000 0 0 0 test002 test-022	P ≥ UNDER 40.0cc 20.0cc 10.0cc domo_1 domo_1 domo_1 40.0cc	本志活日 下erti00周囲訳(
缺陷分布	状态 >	检测次数趋势	日趋势 >	项目规模排行	
Milling III 700 400 400 500 100 100 101 101 101 101 1	ی هر • • • • • • • • • • • • • • • • • • •	5 4 3 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		100,000 80,000 60,000 40,000 20,000	ی هر

第 14 页 共 34

3.2.2. 缺陷数据

点击【仪表盘】下的【缺陷】二级菜单,可展示缺陷维度的数据分析,同样也能够以项目组为维度展示缺陷数据分析,如图所示:



3.2.3. 人员绩效数据

点击【仪表盘】下的【人员】二级菜单,可展示人员绩效维度的数据分析,同 样也能够以项目组为维度展示人员绩效的数据分析,如图所示:



3.3. 代码检测流程

3.3.1. 新建检测项目

点击【项目列表】一级菜单下的【项目队列】, 可展示当前所有检测项目的基

本信息,如下图所示:

Hi	06	(表盘) 四 项目列表	民组织管理 日本	局配置 🛛 許可证设置						②帮助	Radmin
+ 8	新建项目									の刷新	▽ 筛选
		项目名称 ≑	检测状态 😄	语言类型 ≑	代码行数 ≎	缺陷总数 🗘	▽ 创建人 ≑	所属部门 🗘	创建时间 🔅	最后检测时间 💠	操作
	0	git001	未检测	C-0-	848914		admin	默认部门	2021/09/26 18:26		210
	0	test002	已完成	J	603	78	admin	默认部门	2021/09/24 11:18	2021/09/26 16:09	國土會
	0	gitTest-01	已完成	J	81457	648	admin	默认部门	2021/09/24 15:18	2021/09/24 15:19	國土會
	0	demo_1	已完成	J	492	68	admin	默认部门	2021/09/25 16:55	2021/09/25 16:59	國土會
	0	demo-01	已完成	•	492	68	admin	默认部门	2021/09/24 19:28	2021/09/24 19:30	C L D
	0	demo-02	已完成	J.	492	68	admin	默认部门	2021/09/24 19:28	2021/09/24 19:29	C L D
	0	test-022	已完成	J	492	68	admin	默认部门	2021/09/24 19:13	2021/09/24 19:14	國士會
	0	test-01	已完成	U	492	68	admin	默认部门	2021/09/24 17:43	2021/09/24 19:08	C + 0
	0	test-02	已完成	J	492	68	admin	默认部门	2021/09/24 17:28	2021/09/24 18:11	C + D
	0	Test1001跟踪测试	未检测	U.	57		admin	默认部门	2021/09/24 18:10		🗹 ± 🗇
全	选								显示第	[1-10条 共12条 < 1 2 ;	> 10条/页 >

点击左上方【新建项目】,打开新建项目界面,在该界面中可以新建项目,如

图所示:

り ② 仪表盘	四项目列表 限组织管理	B 全局配置 B 许可证设置				②帮助	Aadmin
1 项目信息			2 检测项及编译器	和置		3 检测	文件配置
* 项目名	称						
程序运行操作系统位于	数 💿 64位 🔿 32位 🔿 16位	() 8位					
*导入方:	式 💿 文件夹	选择文件夹					
	GIT						
	⊖ svn						
	○ 压缩包						
项目可见范围	围 🔿 仅个人可见						
	● 全部可见						
	○ 部分可见						
		取消	下一步 完成并检测	完成但不检测			

- 1. 基本信息:
 - 1) 新建项目: 点左侧菜单【新建项目】, 打开项目导入界面
 - 2) 项目名称: 输入导入的项目名称
 - 3) 基本信息:选择基本信息,设置项目的基本信息 (默认选择)
 - 4)项目类型:选择导入的项目类型,支持C、C++、JAVA、HTML、Kotlin、
 JavaScript、Python、Scala等
 - 5) 选择导入方式: 支持 Git、SVN、文件夹、压缩包四种导入方式
 - i. 文件夹: 如图, 可以选择将整个文件夹作为项目导入, 如图所示:

第 16 页 共 34

- ii. 压缩包:可以选择将整个压缩包作为项目导入
- iii. Git: 输入 git 地址、用户名、密码,然后单击【浏览】,选择导入的项目。
- iv. SVN: 输入 SVN 地址、用户名、密码, 然后单击【浏览】, 选择导入的项目, 填写方式同上。
- 2. 检测项及编译器配置

在项目基础信息填写步骤中点击【下一步】,出现以下配置:

 ✓ 項目信息 ▲謝道音: ✓ C/C++语言 □ Java语言 □ Python语言 □ PHP语言 □ Kotlin语言 □ 	检测项及编译器配置 JavaScript语言 HTML语言 Scala语言	3 检测文件配 스
C/C++语言		
✓ C_CPP缺陷检测项(CWE通用缺陷) 189/193 ⑦ GJB_8114(国家军用标准C/C++语言可靠性编程规范) ⑧	 ● 默认 ○ 自定义 ☑ 致命 ■ 严重 ☑ 重要 ■ 次要 ● 默认 ○ 自定义 □ 强制 □ 建议 	
 MISRA_2012(工业标准的C语言嵌入式可靠性编程规范_MISRA2012) ⑦WASP安全漏洞集 ⑦ 	 ● 默认 () 自定义 () 严重 () 重要 () 次要 () 提示 ● 默认 () 自定义 () 致命 () 严重 () 重要 () 次要 	
编译器配置		
库配置 / 添加 2		
C/C++		
编译器配置 GCC(默认) > 添加		

- 勤字1区域:可在该界面对缺陷类别、编码规则、安全漏洞、度量项进行配置:
 - ① 配置检测规则:点击相应规则名称后的 ⑦ 按钮,出现检测规则详
- 情,可酌情选择为当前项目配置哪些检测规则,忽略哪些。如图:

✓ 项目信息	启用的规则数量:189/193	Cpp类型强制转换为BSTR
检测语言: ☑ C/C C/C++语言 ☑ C_CPP缺陷检测 □ GJB_8114(国	 ▼ S_01 BSTR ▼ 【次要】Cpp类型强制转换为BSTR ▼ 【次要】C类型强制转换为BSTR ▼ S_02 C或C++警告 ▼ S_03 不可达路径 	缺陷解释: 因Cpp类的字符串和BSTR字符串的结构不同。BSTR字符串具有4个字节的长度的前缀。包含接下来的数据字符串的字节数。此外。BS 字符串还包含嵌入式的 nul 字符,并且不是强类型化的。若强制将Cpp类型转换为BSTR类型的字符串、则会导致程序基项或者可能导致内 用问题。计情可参考CWE-034; CVE-2004-1670。 可能导致的风险: 因 Cpp 类型的字符串和 DSTR 字符串的结构不同。相互转换会带来不可预如的结果。非 BSTR 指针转为 BSTR 可能会带来内存使用上的问 在某些缘下。 面本 pointeda可定 整理。但如今产生等处结果。
MISRA_2012 OWASP安全漏 编译器配置 库配置	 ▼ 5.04 不建议使用的函数 ▼ 5.06 使用已释放的内存 ■ 5.06 内存泄漏 ▼ 5.07 内联关键字作用范围 	銀騎措施 与C样式学符串不同,BSTR学符串具有4个字节的长度的前缀,包含接下来的数据学符串的字节数。此外,BSTR学符串还包含嵌入式的 null字符,并且不是强类型化的。出于这些原因,最好不要在新设计中使用BSTR。对于现有接口,请务必谨慎进行转换和使用 Sys*Alloc* SysFere 和 String* 内存分配函数。
C/C++ 编译器配置 (又、0.0 (個長等数要用部空量電量 又、0.0 高数未使用 又、10 分号的错误使用 、、11 分配与释放内存不匹配 、、11 分配与释放内存不匹配 、、12 规定条件为常数 、、13 动态链接库 、、14 宽量或值未使用 	11 (RUNOF) 2 void bstr_cast_cpp() { 3 4 wchar_t *foo = L"abC"; 5 6 BSTR bar = reinterpret_cast <bstr>(foo); //非 BSTR 变量 foo 被强制类型转换为 BSTR 类型指针 7 8 }</bstr>

2) 数字 2 区域:编译器配置。系统内置了多种编译器,可根据需要进行选

择,一般默认即可。若自定义配置,则步骤如下:

- ① 添加库配置:根据项目需要配置编译器:
 - 步骤1:在上图数字2区域的库配置处,点击【添加】按钮,根据 如下图所示内容进行配置:

▶ ○ 仪表盘 ■ 项目列表 ■ 组织管理 □ 全局配置 □	EQ 筛选		٩				新建库
	○ 系统	库名称	语言	代码行	文件数量	标篮	操作
	新建库		×	48043	502	qt4	更新 删除
检测语言:				23804	285	qt4	更新 删除
C/C++语言	* 库名称:					显示第1-2条 共2条	: < 1 > 10 条/页 ∨
☑ C_CPP缺陷检测项(CWE通用缺陷) 189/193 ③	* 库标签:						
GJB_8114(国家年用标准C/C++语百可罩性编程规范) ② MISRA_2012(工业标准的C语言嵌入式可靠性编程规范_MISRA2012) ③	导入方式:	夹 ◯ GIT ◯ SV	N				
□ OWASP安全漏洞集 ⑦		上传压缩包					
编译器配置							
库配置 > 滴加			RV 38 740 3.4				
C/C++			10 10				
编译語記畫 GCC(默认) > 添加							
	关闭						

- ▶ 步骤 2:选择"用户"
- ▶ 步骤 3: 点击"新建库"
- ▶ 步骤 4: 输入库名称
- ▶ 步骤 5:选择库配置

第 18 页 共 34

- ▶ 步骤 6:选择压缩包方式导入库文件
- ▶ 步骤 7: 点击【确认】
- ② 添加编译器配置

编译器配置:

- > 步骤 1: 在在上图数字 2 区域的编译器配置处, 点击【添加】
- 步骤 2: 点击自定义编译器处的【添加】按钮,根据如下图所示内容进行配置:

	系统自带编译器 系统宏 数据类型 标识符 系统头文件
✓ 项目信息	c5500_4.4.1
检测语言: 🔽 C/C++语言 🗌 Java语言 📄 Python语言 📄 PHP语言 📄 Kotlin.	自定义编译器 ×
C/C++语言	编译著名称
☑ C_CPP缺陷检测项(CWE通用缺陷) 189/193 ⑦	选择模板 人
GJB_8114(国家军用标准C/C++语言可靠性编程规范) ◎ MISRA_2012(工业标准的C语言嵌入式可靠性编程规范_MISRA2012) ◎	系统自带编译器
◯ OWASP安全漏洞集 ⑦	C5500_4.4.1 取消 确定 arm_5.1.6
编译器配置	c2000_6.2 msp430_4.3.3
库配置 ∨ 添加	tornado mssdks
C/C++	自定义编译 ARMCC
编译器配置 GCC(款认) > 添加	gcc_tmp ARMCC 504u2

- ▶ 步骤 3: 输入编译器名称
- ▶ 步骤 4: 下拉选择模板
- ▶ 步骤 5: 点击【确定】

宏定义添加:

▶ 步骤 6: 点击【添加】系统宏定义:

▶ j ② 仪表盘 P 项目列表 限组织管理	系统自带编译器	系统宏 数据类型 标识符 系统失文件
	MinGW	
✓ 项目信息	GCC(默认)	系统宏定义 添加 编辑 删除
检测语言: ☑ C/C++语言 □ Java语言 □ Python语言 □	c2000_6.2.7	安定义上留头
	tornado	100
C/C++语言	terboc3	s=123
☑ C_CPP缺陷检测项(CWE通用缺陷) 189/193 ⑦	vs2010	td=qwe
GJB_8114(国家军用标准C/C++语言可靠性编程规范) ⑦	c6000_7.4.8	asd=145 tg
MISRA_2012(工业标准的C语言嵌入式可靠性编程规范_MIS OWASP安全漏洞集 ③	gcc-arm-none-eabi-4_7- 2013q3	ajjj=1 sd
编译器配置	vs6	
度配要 · · · · · · · · · · · · · · · · · · ·	vs08	
Р#НО ЛИ 79% / ЛИ	自定义编译器 💿 😑	
(4)活卵肥骨 (CCC(計注)) · · · · · · · · · · · · · · · · · ·	gcc_tmp	
	test	
	关闭	

> 步骤 7: 按照提示格式输入宏定义:

永 玩日带骗咩話	系统宏 数据类型 标识符	符 系统头文件
MinGW		
GCC(默认)	系统宏定义	添加 编辑 删除
c2000_6.2.7	宏定义 上箭头	系统宏
tornado	-122	请输入宏定义。格式:key=value
terboc3	s=125	* 宏定义:
vs2010	asd=145 fg	74.1
c6000_7.4.8	asu-145 lg	ЩХ /Л 946 1.X
gcc-arm-none-eabi-4_7- 2013q3	ajjj= rsu	
vs6		
vs08		
自定义编译器 💿 🕤		
gcc_tmp		
test		

▶ 步骤 8: 点击【确认】按钮

编译器数据类型:进行配置,包括对数据类型的定义、固有类型的格式、类型大小和对齐方式的定义。

h」 鸿渐科技

系统自带编译器	系统宏数据	居类型 标识符	牙 系统头文件		
MinGW	影는 사고 이는 것				
GCC(默认)	数据失型定义				
c2000_6.2.7	右移行为	 arithmetic 	logical		
tornado	位域变量解释	 arithmetic 	logical		
erboc3	字符解释	signed	 unsigned 		
vs2010					
c6000_7.4.8	固有类型				
gcc-arm-none-eabi-4_7- 2013q3	size_t	unsigned_inttto	a 🗸]		
56	ptrdiff_t	long	×		
608	wchar_t	unsigned_char	V		
定义编译器 😗 😑					
c_tmp	类型大小和对齐				
st	基础类型		大小	对齐	操作
	char		3	10	更新
	short		16	36	更新
	int		22	22	更新

对编译器的标识符:进行配置,包括内部标识符长度和外部标识符长度,配置

完点击【更新】按钮:

系统自带编译器	系统宏 数据类型 标识符 系统头文件
MinGW	
GCC(默认)	标识符 更新
c2000_6.2.7	
tornado	内部标识符长度 5
terboc3	外部标识符长度 10
vs2010	
c6000_7.4.8	
gcc-arm-none-eabi-4_7- 2013q3	
vs6	
vs08	
自定义编译器 🔹 😑	
gcc_tmp	
test	

编译器的系统头文件:进行配置,点击【新建文件夹】,然后点击【导入文件】; 择导入方式,可选"文件"导入、"文件夹"导入、"压缩包"导入三种方式导入 头文件;点击【确定】按钮:

系统自带编译器	系统宏	数据类型	标识符	系统头文件	
MinGW					
GCC(默认)	系统头文件				新建文件夹导入文件删除
c2000_6.2.7	配置				文件夹名称
tornado					
terboc3					Tân j.k.
vs2010					
c6000_7.4.8					
gcc-arm-none-eabi-4_7- 2013q3					
vs6					
vs08					
自定义编译器 😌 🕞					
gcc_tmp					
test					
系统自带编译器	系统宏	数据类型	标识符	系统头文件	
系统自带编译器 MinGW	系统宏	数据类型	标识符	系统头文件	
系统自带编译器 MinGW GCC(翻	系统宏	数据类型	标识符	系统头文件	新建文件夹 导入文件 删除
系统自带编译器 MinGW GCC(趴 	系统宏	数据类型	标识符	系统头文件 X	新建文件夹 导入文件 删除
系統自带编译器 MinGW GCC(即 c2000_ tornadc	系统宏	数据类型	标识符	<u>系统头文件</u> ×	新建文件夹 导入文件 删除
<mark>系统自带编译器</mark> MinGW GCC(贯 c2000_ tornadc terbocc ○ 文件 terbocc ○ 文件	系统宏	数据类型	标识符	系统头文件 ×	新建文件夹 导入文件 删除
系統自帯编译器 MinGW GCC(数 c2000	系统宏 !	数据类型	标识符	系统头文件 X	新建文件夹 导入文件 删除
系統自带编译器 MinGW GCC(割) c2000_ 导入方式 tornadc 文件 terboc2 文件 vs2010 压缩包 c6000	系统宏	数据类型	标识符	系统头文件 X	新建文件夹 导入文件 删除
系統自带编译器 MinGW GCC(数) 中入方式 c2000, tornadc 文件 terboc3 文件 terboc4 文件 terboc5 文件 c6000, gcc-arr 2013qt	系统宏 !	数据类型 取 ;	标识符 消 确 1	<u>系统头文件</u> ×	新建文件夹 导入文件 删除
S統自带编译器 MinGW GCC(数) 中入方式 tornade 文件 terboc5 文件 vs2010 Gc6000 gcc-arr 2013qt	系统宏	数据类型	标识符 消 确 X	系统头文件 X	新建文件夹 导入文件 删除
S統自带编译器 MinGW GCC(創) 中入方式 c2000 tornade 文件 terboci 文件 c5000 Gcc-arr 2013q; vs6 vs08	系統宏	数据类型	标识符 消 确 x	系统头文件 ×	新建文件夹 导入文件 删除
S統自带编译器 MinGW GCC(鄧) 令入方式 c2000 tornade 文件 terbocc 文件 vs2010 正縮包 c6000 gcc-arr z013q; vs08 自定火编译器 ●	系统宏	数据类型	标识符 消 确 X	系統头文件 X	新建文件夹 导入文件 删除
S統自带编译器 MinGW GCC(鄧) 令入方式 c2000, 小方式 terbocc 文件 terbocc 文件 terbocc 文件 terbocc 文件 terbocc 文件 vs2010 正縮包 gcc-arr 2013q vs08 自定义编译器 ③ gcc_tmp	系统宏	数据类型	标识符 消 确 X	<u>系统头文件</u> ×	新建文件夹 导入文件 删除

数字1、数字2区域中的所有配置完成,点击【下一步】。

3. 检测文件配置

可设置哪些项目需要检测,哪些项目不需要检测:工具根据检测语言自动识别了需要检测的文件,如果需要变更,更改文件的选中框状态即可,如图所示:

▶ · · · · · · · · · · · · · · · · · · ·	理 國 全局配置 國 许可证设置	⑦帮助 名admin
√ 项目信息 检測文件配置 ○ 振幸	✓ 检测项及编译器配置	3 检测文件配置
文件列表 2 全选	检测文件列表	
 CMakeLists.txt server.c udp_client.c udp_server.c CMakeLists.txt CODE_OF_CONDUCT.md CodingGuidelines.md CONTRIBUTING.md CONTRIBUTING.md in conversions in binary_to_decimal.c in binary_to_hexadecimal.c in binary_to_cotal.c 	C-master/client_server/client.c C-master/client_server/do_client.c C-master/client_server/do_client.c C-master/client_server/do_server.c C-master/conversions/binary_to_decimal.c C-master/conversions/binary_to_texadecimal.c C-master/conversions/binary_to_texadecimal.c C-master/conversions/docimal_to_binary.c C-master/conversions/docimal_to_binary.c C-master/conversions/docimal_to_binary.c C-master/conversions/docimal_to_binary.c C-master/conversions/docimal_to_binary.c C-master/conversions/docimal_to_binary.c C-master/conversions/docimal_to_binary.c C-master/conversions/docimal_to_binary.c C-master/conversions/docimal_to_octal.c C-master/conversions/docimal_to_octal.c C-master/conversions/hexadecimal_to_octal.c C-master/conversions/hexadecimal_to_octal.c	
	上一歩 項目检測	

4. 创建项目完成:选择完文件、配置设置完成后,点击【项目检测】完成项目的创建。

3.3.2. 检测队列

多并发检测情况下,如果当前检测项目数超过了设定的并发数量,则后来检测项目排队,并且可以调整排队顺序,如图所示,在【项目列表】下的【检测队列】 菜单:

H	0 (1)	表盘 凹 项目列表	艮 组织管理	日 全局配置	🖪 许可证设置				②帮助	A admin
										〇 刷新
		项目名称	档	测状态		语言类型	代码行数	操作		
0	\otimes	demo-02	51	擎启动中			471	$\wedge \downarrow \rightarrow$		
0	\otimes	demo_0927	55	%		C	25756	$\wedge \downarrow \rightarrow$		
	U	demo_1	9	待中		J.	471	$\land \lor \mathrel{\dot{\rightarrow}}$		

3.3.3. 检测项目编辑

在项目列表页面,点击编辑按钮 2,按照需求编辑响应的参数:

H	Ø	仪表盘 🔳 項目:	列表 昆组织管理	日 全局配置	🛿 许可证设置					⑦蒂助	유 admin
+ 5	所建项目	1								〇 刷新	▽ 筛选
		项目名称 💲	检测状态 ≑	语言类型 🗘	代码行数 🗘	缺陷总数 👙	♡ 创建人 ≑	所属部门 ≑	创建时间 👙	最后检测时间 ↓	操作
	$\cap \otimes$	demo-02	5%	J	492		admin	默认部门	2021/09/24 19:28	2021/09/24 19:29	
	$\cap \otimes$	demo_0927	5%	0	25756		admin	默认部门	2021/09/27 15:22		
	0	git001	未检测	C-1-	848914		admin	默认部门	2021/09/26 18:26		C ± C
	0	test002	已完成	J	603	78	admin	默认部门	2021/09/24 11:18	2021/09/26 16:09	ßłÖ
	0	gitTest-01	已完成	J	81457	648	admin	默认部门	2021/09/24 15:18	2021/09/24 15:19	ßłđ
	0	demo_1	等待中	J	492		admin	默认部门	2021/09/25 16:55	2021/09/25 16:59	
	0	demo-01	已完成	J	492	68	admin	默认部门	2021/09/24 19:28	2021/09/24 19:30	ßłđ
	0	test-022	已完成	J	492	68	admin	默认部门	2021/09/24 19:13	2021/09/24 19:14	C L D
	0	test-01	已完成	J	492	68	admin	默认部门	2021/09/24 17:43	2021/09/24 19:08	C ± C
	0	test-02	已完成	J	492	68	admin	默认部门	2021/09/24 17:28	2021/09/24 18:11	CLŌ
_ 全	选								显示第1-10务	を共13条 < 1 2 ;	▶ 10 条/页 ∨

编辑内容如下图所示,并且在【检测器及编译器配置】中,可以补充缺失文件, 也可以直接通过项目信息的右上方按钮【缺失文件】进入此功能。

(缺失文件:工具在编译、解析代码过程中遇到的造成编译失败的公共库文件或自研文件。 通常工具会根据代码自动补全技术进行智能补全公共库文件,但其他文件需要手动补全,以使 检测过程更完美,检测结果更准确)

3.3.4. 缺陷结果查看

1. 缺陷查看主界面: 在项目菜单中选择我们刚才测试的项目"demo_0927", 打开项目总览主界面,如下所示:



2. 进入缺陷列表页签, 即可查看所有缺陷列表, 点击缺陷编号, 即可在中间

第 24 页 共 34

主区域内查看相关问题。如图所示:

▶ ○ 仪表金 回 項目列表 図 组织管理 図 全局配置 目	许可证设置			⑦帮助 Aadmin
demo_0927 本次检测开始时间: 2021年09月27日 15时33分; 检测时长: 41秒; 被检测	目导入方式: 压缩包			□ 缺失文件 上 导出检测结果
1X.代品 副761375代 決重 出版以用半 173 次 出出				
octal_to_hexadecimal.c ×			Pa	_关 陷信息
<pre>33 Our social/desactisat(long scalables){ 34 our social/desactisat(long scalables){ 35 our social/desactisat(long scalables); 36 our social/desactisat(long scalables); 37 our social/desactisat(long scalables); 38 our social/desactisat(long scalables); 39 our social/desactisat(long scalables); 31 our social/desactisat(long scalables); 32 our social/desactisat(long scalables); 33 our social/desactisat(long scalables); 34 our social/desactisat(long scalables); 35 our social/desactisat(long scalables); 36 our social/desactisat(long scalables); 37 our social/desactisat(long scalables); 38 our social/desactisat(long scalables); 39 our social/desactisat(long scalables); 30 our social/desactisat(long scalables); 31 our social/desactisat(long scalables); 32 our social/desactisat(long scalables); 33 our social/desactisat(long scalables); 34 our social/desactisat(long scalables); 35 our social/desactisat(long scalables); 36 our social/desactisat(long scalables); 37 our social/desactisat(long scalables); 38 our social/desactisat(long scalables); 39 our social/desactisat(long scalables); 30 our social/desactisat(long scalables); 31 our social/desactisat(long scalables); 32 our social/desactisat(long scalables); 33 our social/desactisat(long scalables); 34 our social/desactisat(long scalables); 35 our social/desactisat(long scalables); 36 our social/desactisat(long scalables); 37 our social/desactisat(long scalables); 38 our social/desactisat(long scalables); 39 our social/desactisat(long scalables); 30 our social/desactisat(long scalables); 31 our social/desactisat(long scalables); 32 our social/desactisat(long scalables); 33 our social/desactisat(long scalables); 34 our social/desactisat(long scalables); 35 our social/desactisat(long scalables); 36 our social/desactisat(long scalables); 37 our social/desactisat(long scalables); 38 our social/desactisat(long scalables); 39 our social/desactisat(long scalables); 39 our social/desactisat(long scalables); 30 our social/desactisat(long scalables); 30 our social/desactisat(</pre>	代码区		1	総裁説明 主文件(octal,to,hexadecimal(2)第[53]行周用道数(octalToHexadecimal)2)記 主文件(octal,to,hexadecimal(2)第(53)行周用道数(octalToHexadecimal)2)記 主ご論判認 在面(53)行周用道数[octalToHexadecimal]2)記介符 読記[20] (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)
分配内存未释放 在第(39)行调用函数[malloc]分配内存				
54) 55 /** 57 /* @prief Main function				
□ 傳选: 按缺陷类型聚合 > 请输入得选关键 Q	2 等选:	负责人	请输入缺陷编号	
▼ ■ ◎ 所有缺陷	● 缺陷编号 ⇒ 类型 ⇒		位置 ⇔	操作
▶ ■ 致命(2)	📄 🚾 R000015 🛛 🔚 分配内存	未释起③ 朱本昭4月3日店	octal_to_hexadecimal.c:53	未处理▼ admin▼ ① 备注
▼ ■ 严重(21)	📃 🚥 R000016 🛛 🖪 分配内存		octal_to_hexadecimal.c:50	未处理▼ admin▼ ① 备注
☑ 万氟內分水(40)(2) ☑ 变量未初始化使用-初始化不完整(1)	🗌 🚾 R000024 🛛 🖪 变量未初	始化使用-初始化不完整③	to_decimal.c:40	未处理▼ admin▼ ① 备注
✓ 污染数据用于库函数参数导致缓冲区溢出(1)	🗌 🚥 R000007 🛛 🖪 污染数据	用于库函数参数导致缓冲区溢出⑦	infix_to_postfix.c:41	未处理▼ admin▼ ① 备注
✓ 污染数据用于数组下标导致缓冲区溢出(2)	🗌 🧰 R000012 🛛 層 污染数据	用于数组下标导致缓冲区溢出③	udp_server.c:80	未处理▼ admin▼ ① 备注
外部输入作为循环跳出条件(1)	🗌 🪾 R000013 🛛 層 污染数据	用于数组下标导致缓冲区溢出②	udp_client.c:77	未处理▼ admin▼ ① 备注
 ✓ 参数类型不兼管(2) ✓ 缺少长度限制(1) 		num antinenten en en 1996		显示第1-20条 共21条 〈 【】 2 → 〔20 条/页 ∨ 〕

以上各区的功能解释如下:

- 缺陷列表区:根据查询条件查询后,缺陷的查询结果会在【缺陷树】及右 方缺陷列表中根据查询结果列表展示。
- 2) 缺陷说明及跟踪区:在【缺陷列表】中,单击一条缺陷,缺陷详细信息区 中会显示该条缺陷的详细信息,还可以列表显示缺陷的【引发位置、缺陷 的发生位置、缺陷所在文件、缺陷发生行数】,在列表中单击,【缺陷代 码主工作区】中的代码,会自动定位到发生问题的代码行,并底色标亮。
- 3) 缺陷判断区,可以对缺陷进行判断,类别包含【缺陷、误报、待定、故意 为之】,同时能对【缺陷等级】进行判定。
- 4) 缺陷知识库,在缺陷列表的某条缺陷上,单击^⑦图标,显示缺陷知识库:
 缺陷的详细描述、修复建议、代码示例等。
- 3. 代码架构图

在代码文件中选中函数或者变量并右键,可查看函数调用关系图、值跟踪图等 代码架构图,帮助用户在脱离 IDE 的环境下准确阅读代码上下文,确认问题。例如 值跟踪图,可以展示反映污点变量在程序中经过的上下文处理,如下图所示:

第 25 页 共 34

トゴ 〇 仪表金 日 第目列表 同 信約管理 四 全局配置 四 许可证设置	③帮助 Aladmin
demo_0927 本次检测开始时间: 2021年09月27日 15时33分; 检测时长: 41 步; 被检项目导入方式: 压缩包	
仪表盘 缺胞列表 度量 函数清单 历史 配置	□ 缺失文件 上 导出检测结果
[] octal_to_hexadecimal.t × [] hexadecimal.t hex值跟踪图	×
16 Finclude esticitude virtuine with for messaria and the second se	(5.66 年間)
18 Point States	Call Dependence
28 • @brief Convert a hexadecimal number to d	Every Departmenter thwoods Software
21 * @param hex Mezadecimal number to convert 1 1 hex.to.ocf(hex) 22 * @perfums A pointer to the converted octa	Legid Dependence
23 4/ C-masterconversionsnekadeoma_to_octaz.cz4	Flow Dependence
24 CONSt Char MIDE, LO, DCB CONSt Char MIDE)	ReturnDependence
26 #define MX/OCT_STR_LIN 23 /e 17,7777,7777	Sumflow Desendence
28 messet(octa), \\vec{v}, MAX_OCT_STR_L(N);	
29 unsigned long long decinal = 0; astericonversion/bracketinal to octa2 c-41 C-mastericonversion/bracketinal to o	imal to octai2.c:85
31 Int I = 0; +	-
34 if (hex == NULL) 25 {	
36 // Return an empty string charcharbey while/they ta t	10) while("hex != "
d7 return ectal; C-master/conversions/hexadecimal_to_octal2.cx43 C-master/conversions/hexadecimal_to_octal2.cx43 C-master/conversions/hexadecimal_to_octal2.cx43	cimal_to_octal2.c:41 C-master/conversions/hexadeci
39	
【 構造: 按缺胞类型聚合 v 前输入排选关键	
▼ ◎ 所有缺陷	
◆ ■設命(2) ch = ch-'à'+10 l(ch == ch-'à'+10 l(ch == 'A'))	if(ch <= Ÿ)
metrigonalization ensurements and example to call 2.c51 consteriornversionshexadecimal_to_cotal2.c53 C-mastericonversionshexadecimal_to_cotal2.c53 C-mastericonversionshexadecimal_to_cotal2.c53 C-mastericonversionshexadecimal_to_cotal2.c53 C-mastericonversionshexadecimal_to_cotal2.c53 C-mastericonversionshexadecimal_to_cotal2.c53 C-mastericonversionshexadecimal_to_cotal2.c53	-master/conversions/hexadecimal_to_octal2.c:49
→ 「無(z)」	
夏量未初始化使用-初始化不完整(1)	
河染数据用于库函数参数导致缓冲区溢出(1)	
污染物质用于数据下标号数据冲区溢出(2)	
○ 分部編入行力運転局(1) ○ 約4編入行力運転局(1) ○ 約4編入行力運転局(1)	

3.3.5. 代码度量

在检测项目的结果页进入【度量页签】, 可查看度量列表、度量结果导出, 如

图所示:

J 〇 仪表盘 回 項目列表 図 组织管理 図 全局配置 図 許	可证设置	③帮助 A	admin
demo_0927 本次检测开始时间: 2021年09月27日 15时33分; 检测时长: 41 秒; 被检项目导 仪表盘 缺陷列表 度量 函数清单 历史 配置	入方式: 压缩包	13 BAXA 🖸 BAM	测结果
· 报來		导出度	
▼ 🔚 demo_0927 度量指标		度量值	
▼ 🗁 C-master goto语句数目-文件	0	0	
▼ 🗁 client_server 代码总行数	0	158	
 udp_client.c 函数声明注释数目 	0	1	
 udp_server.c 函数声明注释比率 	0	50.00%	
 client.c 可执行代码注释率 	0	39.25%	
Server.c 可执行代码行数	0	107	
▶ Conversions 圖复杂度-模块	0	9	
➤ data_structures 基本圈复杂度-模块	0	4	
developer_tools 失部注释行数	0	7	
→ exercism 实际使用方法与可能使用方法 → C geometry 比	•	100.0%	
> immerical_methods 平均方法复杂度	•	4.5	
数据变量测试集	•	7	
文件总方法数-文件	٥	2	
注释率-文件	0	26.58%	
注釋行数	0	42	
Halstead 程序容量-文件	Θ	476.82	
Halstead 程序长度-文件	٥	89	
空注釋行數	0	0	
空行数	0	18	

1) 度量列表:根据度量查询及度量项选择,列表区显示查询的度量结果信息。

2) 度量导出:将度量结果导出成报告,支持【Excel】格式导出。

3.3.6. 函数清单

在检测项目的结果页进入【函数清单】页签,展示当前项目中的所有函数清单 及函数实现代码,并且可配置函数类型,选中之后,当前函数在"3.3.4"章节内容

第 26 页 共 34

中的"代码架构图"中会高亮显示。

〇 仅表金 四 项目列表 〇 demo_0927 本次检測开始时间: 2021年09月27日 (八表金) 以和列表 度量 品数清单	组织管理 15时33 历史	目 E 分; 检測P 配置	• 全局配置 四 许可证设置 • 计可证设置 • 长: 41 秒;被检项目导入方式: 压缩	2	🥝 操作成功!			88 ET	③帮助 文件 上	久 admin - 导出检测结果
hexadecimal_to_octal2	٩	函数:	S Q 函数类型			C_at	oi_str_to_integer.c ×	G hexadecimal_to_octal2.c ×		
 demo_0927 		序号	位置 🗘	函数签名 ⑦	标注类型	5	* The input is valid f *	from 0 to 0xFFFF_FFFF_FFFF_FFFF.		
- C-master			C-master/conversions/	hex_to_oct(const char) 业务类——交 ∨		 7 * At first, this program converts a hex string to an unsigned long long 8 * decimal, and then to an otal string. 9 * * When there is an invalid character in input string, this program stops 1 * parsing and converts the string until that character. 				
 conversions hexadecimal_to_octal2.c 		1 hexadecimal_to_octal2.c		main()						
				显示第	1-2条 共2条 〈 1 〉 10 条/页 >	12 13 14	<pre>* @see hexadecimal_to_ */</pre>	_octal.c		
						15 16 17 18	tinclude <stdio.h> /// tinclude <string.h> ///</string.h></stdio.h>	<pre>/ for printf() and fgets() / for memset()</pre>		
						19 / 20 21 22	** * @brief Convert a hex * @param hex Hexadecim * @returns A pointer t	<pre>kadecimal number to octal number. nal number to convert. to the converted octal string.</pre>		
						23 24 25 (*/ onst char *hex_to_oct((const char *hex)		

例如,将 "adaline_predict" 函数标准为特殊函数之后,再生成函数调用图时, 此函数会标紫显示(特殊函数标识和高亮颜色可在 3.5.5 章节中具体配置),如下 图所示:

H	❹ 仪表盘	□ 项目列表	艮 组织管理	日 全局配置	🛿 许可证设置	t i i i i i i i i i i i i i i i i i i i					③帮助	Radmin
demo_	0927 本次检测开始	台时间: 2021年09月	27日 15时33分;	检测时长: 41 秒; 被	检项目导入方式: 压	压缩包				(D. 1967		D 11 14 Miles III
仪表盘	缺陷列表	度量 函数清单	1 历史	配置							÷ ک	守出程测结果
© h	exadecimal_to_oct	tal2.c ×		hex to oct函数调	用关系图				₽ 缺陷信息	×		
13 14	* @see hexadec */	imal_to_octal.c		nex_to_ootmaxe	107070						1的数组 [oct	all,在第
15 16	#include <stdio< td=""><td>.h> /// for print</td><td>f() and fget</td><td>woods t</td><td>oftware tion use</td><td></td><td></td><td></td><td></td><td>节点类型 ^</td><td>自出了其定り</td><td>《范围,</td></stdio<>	.h> /// for print	f() and fget	woods t	oftware tion use					节点类型 ^	自出了其定り	《范围,
17 18	#include <strin< td=""><td>g.h> /// for memse</td><td>t()</td><td></td><td></td><td></td><td></td><td></td><td></td><td>^四55552.1 evaluation 自定义通数2021 NGDa</td><td></td><td></td></strin<>	g.h> /// for memse	t()							^四 55552.1 evaluation 自定义通数2021 NGDa		
19 20	/** * @brief Conve	rt a hexadecimal r	umber to oct							goja.net		
21	* @param nex H * @returns A p	exadecimal number ointer to the conv	to convert. erted octal								超出了数组	[octal]的
23	*/ const char +hex	_to_oct(const char	+hex)						7			
26	#define MAX_OCT	_STR_LEN 23 /* 17_ octal/MAX OCT STR	7777_7777_77 LEN1:									
28 29	memset(octa	l, '\0', MAX_OCT_S	TR_LEN); //		_				→ printf			
30 31	unsigned lo int i = 0;	ng long decimal =	0;			main C-master/conversions/hexadecimal_to_octal2.c:104	he Constant and the second second	ex_to_oct	memset			
32 33	int len;				ŀ	• •	-	-				
34 35	if (hex ==) {	NULL)										
36	// Retu	rn an empty string									-	
2 筛送	 按缺陷类型聚合 66 表 計除 	1 > 请输入等选;	ミ雄 오									
- <u> </u>	■致命(2)											
•	▲ 一般的数组组界 ■ 严重(21)	E(Z)										
	重要(10)									+	< 1 > (20 条/页 🗸
, .	— /大変(57)										1	

3.3.7. 检测历史

在检测项目的结果页进入【历史】页签,可查看项目的历次检测结果,并且能 够看到数据对比分析,如下图所示:

li ⊘	仪表盘 医项目列表 民组织管理	B 全局配置 B 许可	征设置									③鼎	め Radmin
demo_1 本次检	潮开始时间: 2021年09月27日 15时34分; 检测	时长: 18 秒 ; 被检项目导入方式	:压缩包										
仪表盘 缺环	谄列表 度量 函数清单 历史	配置											と 导出检测结果
检测编号	检测时间	代码语言	文件数	代码行数	千行代码缺陷数	缺陷总数	复发数	未处理	已确认	已解决	误报 7	「修复	操作
L002	2021年09月27日 15时34分	J	1	471 🗸	136.00	68	68 🕇	68	0	0	0 0		取消对比
L001	2021年09月25日 16时59分	J	1	492	136.00	68	0	68	0	0	0 0		± 1
											显示第1-2条 扌	ŧ2条 < 1	> 10 条/页 >
对比结果展示													
检测编号	检测时间	代码语言	文件数	代码行数	千行代码缺陷数	缺陷总	数复	[发数	未处理	已确认	已解决	误报	不修复
L002	2021年09月27日 15时34分	U	1	471(-21) ↓	136.00	68	6	8(68) 1	68	0	0	0	0
L001	2021年09月25日 16时59分	J.	1	492	136.00	68	0		68	0	0	0	0

第 27 页 共 34

3.3.8. 报告导出

在检测项目的任意结果页签,点击右上方【导出检测结果】,可根据需求自定 义导出报告的内容,格式支持 Word 和 Excel。如下图所示:

Hi	☑ 仪表盘	回 項目列表 因 组织管理	日本 (C) 全局配置 日 (C) 许可	「证设置						0	帮助 只admin
demo_0 仪表盘	927 本次检测开射 缺陷列表	时间: 2021年09月27日 15时33分 度量 函数清单 历史	;检测时长:41秒;被检项目导》 配置	方式: 压缩包						□ 缺失文件	上 导出检测结果
项目概	要统计						导出结果		语言分类	统计	文件数 🗸
	90 缺陷总数	39 文件数	2762 代码行数	32.80% 代码注释率	1.52 平均團复杂度	CPP 编程语言		8.33/KLoC 重要缺陷密度	CPP 約測1页		324
缺陷统 致命 2 新增到	计 改命缺陷: 2	产量 21 新港产量缺陷:21	重要 10 新增重要缺和: 10	次要 57 新增次要缺税: 57	總示 〇 新增提示缺陷: 0	强制 O 新增强制	缺陷: 0	建议 〇 新増建议缺陷: 0	* @c	 (C++语言(189) (C_CPP缺陷检测项(CWE通 回 S_01 BSTR(2) (Cpp类型强制转换为BST (Cpp类型强制转换为BST (C类型强制转换为BST (C) S_02 C或C++警告(44) 	用缺陷)(189) STR R
缺陷概 1 7. 2.	要统计				21 29 25	20 15 10 5 0	重要缺陷密度 千行代码重要缺陷个	ĝ		5、0.0 不可达器径(1) 5、0.0 不可达器径(1) 5、0.0 存建这使用的函数(5、0.0 内存逻篇(6) 5、0.0 向数参数编码用 5、0.0 向数参数编码用(1) 5、10 分影物继续用(1) 5、10 分影物继续用(1) 5、10 分影物继续用(1) 5、10 分易物继续用(1) 5、10 分易的继续用(1) 5、11 分配与释放内存不 5、12 判定条件为常数(1)	(14) (3) 至町(1) 職置憲(1) 四配(2)

3.4. 项目组

点击一级菜单【项目列表】下的【项目组】二级菜单,可展示项目组的管理页 面。项目组的概念:将多个检测项目(代码项目)统一归入一个项目组进行管理, 也类似产品—代码项目的概念。

设置项目组时,可先新建项目组,再点击项目组的【编辑】菜单,将检测项目 配置进项目组,如图所示:

						() KEB) Xadmi
+ 新建项目组						〇 刷新
项目组名称 ⇔	項目成员数量 ⇔		创建人 ⇔	创建时间 ⇔	操作	
master	4		admin	2021-08-17 11:04	ßŌ	
test002	2		admin	2021-08-23 15:26	ßŌ	
test001	4		123456	2021-08-17 10:53	ßŌ	
					显示第1-3条	共3条 < 1 > 10条/页 >
►i 〇 仪表盘 回 項目列表	民 组织管理 日 全局配置	日 許可证设置	×			⑦帮助 只adm
+ 新建项目组	<u> </u>					の刷新
項目組名称 ⇔	修改				×	
master					0	
test002	* 项目组名称 1	test001		_	0	
test001	項目选择 已	选项目	待选项目		.	
		🔉 Test1001跟踪测试1	demo_1		显示第1-3条	共3条 < 1 > 10条/页、
		🛿 Test1001跟踪测试 🗙 test001	demo_0927			
		x test002	demo-02			
			git001	选中加入项目组		
			test002			
			gitTest-01			
			demo-01	gitTest-01		
			test-022			
			tost=01			

3.5. 全局配置

点击一级菜单【全局配置】,所有的可配置项均在页面左侧二级菜单的功能中。 如下图所示:

·	🛙 项目列表	民 组织管理	3 全局配置	🖸 许可证设置							⊘帮助 Aadmin
③全局配置		全部									新增后缀名
后缀名配置		语言		文件类型	内容						
项目组配置 检测配置		СРР		源文件	A. *.cpp A. *.CXX A. *.C++	As t.ch	As tee As tep	*.C *.C++	As *.cxx As *.c	As top As tinl	A *CPP
重要缺陷对应等级				库文件	As thep	A *.HPP	As tH	A. *.hxx	As the	A. *.HXX	As th
特殊函数配置				配置文件	As *.dsp	As *.sin	As *.vcproj	*.vcxproj			
1070100000		00		源文件	As *.asax	As *.cs	A. *.aspx				
]	cs		配置文件	As tdl	A: *.csproj	As *.config				
		GROOVY		源文件	As *.groovy						
		JAVASCR	IPT	源文件	As ".js						
		KOTLIN		源文件	As *.kt						
		PHP		源文件	As *.php	As *.ini					
				源文件	As .html	As *.jsp	As *.smap	A. •.java			
		AVAL		库文件	Ajar	As *.class	Aclass				
				配置文件	A: *properties	As *.sin	As *.xml	As *.wsdd	A. *.classpath	As *.project	
		VB		源文件	As *.vb	Aa .VB					
		HTML		源文件	As *.html						
		SCALA		源文件	As .scala						
		PYTHON		源文件	As *.py	A. *.pxx					

3.5.1. 后缀名配置

后缀名配置可以新增非通用文件后缀名与开发语言之间的映射关系,帮助工具自动识别开发语言。新增后缀名如下图所示:

▶ ② 仪表盘 □ 项目列表	Q 组织管理 B 全局配置	🖪 许可证设置	③帮助 A.admin
② 全局配置 后继名配置	全部 · · ·		新期后缀名 ×
项目组配置	1015	文件类型 源文件	*通常発型: ・通常発型: ・ 通常内型 ✓ ・ 通常ない 通常なの 通常なPP 通常ない 通常の 通常な での 通常なPP
重要缺陷对应等级	CPP	库文件 配置文件	*文件未刻: 潮文件 ∨ - 「最低名: 」 「最低名: 」 「日本名 □ ***********************************
	cs	源文件 配置文件	×7/9: □ ABEA019 美術 高面
	JAVASCRIPT	源文件 源文件	
	KOTLIN PHP	源文件	III 144 III 1440 III 144

3.5.2. 项目组配置

可以配置当前工具是否开启项目组的概念和对应功能菜单,如下图所示:

▶j ② 仪表盘	四 项目列表 日	组织管理 🖪 全	局配置 🛛	件可证设置	③帮助	A admin
② 全局配置 后缀名配置		是否设置项目组: 🧃) 문 () 종			
项目组配置						
检测配置						
重要缺陷对应等级						
特殊函数配置						

3.5.3. 检测配置

在【检测配置】中,有3项配置:

- 1) "标注为误报的缺陷,下次是否显示"
- 2) 代码架构图中,值跟踪图的默认展示层级
- 3) 代码架构图中,项目函数调用图的默认展示层级

如下图所示:

⑦ 仪表盘 凹 项目	列表 國 组织管理 🔀 全局配置 🔀 许可证设置	②帮助	Radmin
 令 全局配置 后缀名配置 项目组配置 	标注为课报的缺陷,下次检测是否显示: ② 是 ④ 否 设置值跟踪图展示层级: 3		
检测配置 重要缺陷对应等级	设置项目函数调用图展示量级: 1		
特殊函数配置			

3.5.4. 重要缺陷等级配置

在数据仪表盘中,有"重要缺陷密度"的数据分析,那么重要缺陷包含了那些级别的问题,可以在此配置,如图所示:

Ы	⊘ 仪表盘	凹 项目列表	國 组织管理	2 全局配置	🛿 许可证设置			②帮助	유 admin
<mark>尊</mark> 全局	配置		重要缺购米刑: 🛃 致。	b 严重	雨亜	建立			
后缀名	115 M		■X#VIIX第, M 11	HP		XE IA			
项目组	16211								
检测配	300								
重要缺	帕对应等级								
特殊函	数配置								

3.5.5. 特殊函数配置

在【特殊函数配置】中,可以自定义配置重要函数的属性,添加重要函数功能如下图所示:

● 仪表盘	🗹 项目列表 🛛 限 组织管	理 🛛 全局配置	日 许可证设置			⑦帮助 Radm
② 全局配置	函数类型	0 ∠ 8	函数名 Q			添加函数
后缀名配置	● 读函数		id ¢	函数名称 💠	函数定义 💠	操作
项目组配置	- 測试		6126f5294ae1ac52d0efe668	ро	int po(int)	ßŌ
检测配置			612c973c128c8549afb5eba4	maxval	int maxval(int,int)	ßŌ
重要缺陷对应等级	○ 新增函数	数类型123	615167f5a8ffc81b0126e2fe	max	int max(int,int,int)	ßŌ
	○ 操作函数	数	611e1e8486b12b419eb42cc2	main	mian()	ßŌ
特殊函数配置			612c97f5128c8549afb5eba6	getOne	char getOne(char)	ßŌ
		EX.	611e1ea186b12b419eb42cc3	func	func(int)	ßŌ
	● 写函数		612c9799128c8549afb5eba5	duplicateZeros	void duplicateZeros(int,int)	ßŌ
	 位置函 业务类 	数 —交易			显示第1-7条 共7条 < [1 > 10 条/页 >

3.6. 权限管理

1. 创建新用户:进入首页面,点击菜单栏中的【组织管理—用户管理】,跳
 转到用户及权限管理页面,如图所示:

2 仪表盘	凹 项目列表	艮 组织管理	日本 日	许可证设置				⑦帮助 Aadmin
EQ 角色管理	清空条件	搜索用户ID	٩.					添加用户
名称	权限	用户ID	姓名	邮箱	创建时间	角色	部门	操作
普通用户	没有管理权限	test03	张三	123@122.com	2021-08-03	普通用户	默认部门	C C C
管理用户	有用户管理权限	test02			2021-06-21	普通用户	默认部门	
系统超级管理员	系统角色,不可删除	990417			2021-06-28	管理用户	默认部门	CCT
		admin	超级管理员	admin@something	g.com 2019-12-24	系统超级管理员		C C
		test			2021-06-17	普通用户	默认部门	
					2021-06-23	普通用户	默认部门	C C T
					2021-08-02	普通用户	默认部门	C C Ť
å 部门管理	0 ∠ 1				2021-08-10	管理用户	默认部门	C C T
□ 默认部门					2021-08-10	管理用户	默认部门	C C Ť
测试部门		123456			2021-08-14	普通用户	默认部门	
] 测试部门3							显示第1-10条 共19条	< 1 2 > 10 条/页 > 2

新建用户:点击图中的【新建】按钮,在【用户信息】栏中可输入新
 用户的基本信息,若不输入密码,则默认为 123,点击【保存】,即可创建用

O KAM E MAMA	Le seisvier All	SERVOTIONSE EX 1193 40	10.1m				0 1140
角色管理 清空条件		q	_				185 Au
家 权限	用户ID	新建用户			× 🕹	部门	操作
普通用户 没有管理权限	test03	-			通用 <mark>.</mark> 户	默认部门	C C Ĉ
管理用户 有用户管理权限		* 用户ID:			畫用 .户		
系统超级管理员系统角色,不可删除	990417	姓名:			墾用 <mark>,</mark> 户	服代认部门	
	admin	-			流超 <mark>及管理员</mark>		
		即相:			●用,>		
	wujuhua	*角色: 普	通用户		通用, ^会	默认部门	
	zhuhongjie	* #8(7), 5	21 BR/T		通用. ^中	默认部门	C C C
的管理 🔹 🖉 🕯	zhou	apt J.	Cevepi 1		锂用 <mark>,</mark> ≥	默认部门	
默认部门	sunyongjie				里用,户	默代人音印门	C D D
测试部门				取消	13. 通用,户		

户.

② 修改用户信息:点击图标 🗹,修改用户信息,如图所示:

▶ ○ 仪表盘	□□ 项目列表	艮 组织管理 🖪	全局配置 🛛 🛱	可证设置					⑦帮助 Aadmin
■ 角色管理	清空条件		۹				1		添加用户
名称	权限	用户ID	更新用户			×	色	部门	操作
普通用户	没有管理权限	test03					通用户	默认部门	
□ 管理用户	有用户管理权限		* 用户ID:	test03			通用户		
系统超级管理员	系统角色,不可删除	990417	姓名:	张三			理用户	默认部门	C C Ť
		admin	40.00				流超级管理员		C C
			田()和:	123@122.com			通用户		
		wujuhua	*角色:	普通用户	~		通用户	默认部门	C O Û
		zhuhongjie	+ cnnc .	##21.007/1			通用户	默认部门	C O Ť
恭 部门管理	O ∠ T	zhou	. Ibi 1:	WYNUDI 1			理用户	默认部门	C C C
□ 默认部门		sunyongjie				_	理用户	默认部门	COO
□ 测试部门					取 消	确认	通用户		
] 测试部门3								显示第1-10条 共19	9条 < 1 2 > 10 条/页 ∨

2. 角色管理: 使用管理员登录系统后, 进入【组织管理—用户管理】页面,

第 32 页 共 34

在左侧页签,即可添加、修改、删除用户角色组:

2 仪表盘	凹 项目列表	艮 组织管理	日 全局配置 日 许可	正设置				②帮助 Aadmin
≣Q 角色管理	清空条件	搜索用户ID	٩,					添加用户
名称	权限 没有管理权限 有用户管理权限 系统角色,不可删除	用户ID	姓名	邮箱	创建时间	角色	部门	操作
		test03	张三	123@122.com	2021-08-03	普通用户	默认部门	C C Ĵ
管理用户		test02			2021-06-21	普通用户	默认部门	
系统超级管理员		990417			2021-06-28	管理用户	默认部门	C C T
		admin	超级管理员	admin@something.con	n 2019-12-24	系统超级管理员		
- 命 部门管理	0 2 1	test			2021-06-17	普通用户	默认部门	
		wujuhua	吴菊华		2021-06-23	普通用户	默认部门	C C Ť
		zhuhongjie			2021-08-02	普通用户	默认部门	C C T
		zhou			2021-08-10	管理用户	默认部门	C C T
默认部门		sunyongjie			2021-08-10	管理用户	默认部门	C C T
测试部门		123456			2021-08-14	普通用户	默认部门	
测试部门3							显示第1-10条 共1	19条 < 1 2 > 10条/页 >

3. 部门管理:使用管理员登录系统后,进入【组织管理—用户管理】页面,在

H	❷ 仪表盘	🖭 项目列表	圆 组织管理	🖪 全局配置	🖪 许可证设置					②帮助	Radmin
┋ 角色	管理	清空条件	搜索用户ID	٩							添加用户
名称	权限	用户ID	姓名		邮箱	创建时间	角色	部门	操作		
普通	普通用户	没有管理权限	test03	张三		123@122.com	2021-08-03	普通用户	默认部门	C C Ť	
管理用户 系统超级管理员	有用户管理权限 系统角色,不可删除	test02				2021-06-21	普通用户	默认部门			
		990417				2021-06-28	管理用户	默认部门	ľēt		
		admin	超级管	管理员	admin@something.com	2019-12-24	系统超级管理员				
			test				2021-06-17	普通用户	默认部门		
			wujuhua	吴菊华	KI		2021-06-23	普通用户	默认部门	ľēt	
虎 部门管理	0 ∠ 8	zhuhongjie				2021-08-02	普通用户	默认部门	ľēī		
		zhou				2021-08-10	管理用户	默认部门	680		
試订	默认部门		sunyongjie				2021-08-10	管理用户	默认部门	C C Ť	
測试部门测试部门3			123456				2021-08-14	普通用户	默认部门		
									显示第1-10条 共19条 🛛	1 2 > 1	0条/页 >>

左侧下方页签,即可添加、修改、删除部门: