ICS 35.030
CCS L 80

T

团　　体　　标　　准

T/ CIIPA 00012—2024

自主可控网络安全技术　基于网络靶场的软件自主可控能力测试指南

Autonomous and controllable cybersecurity technology—Guideline for testing the autonomous and controllable capabilities of software based on cyber range

2025-06-09 发布　　2025-06-11 实施

中关村华安关键信息基础设施安全保护联盟　发布
中　国　标　准　出　版　社　出版

目　　次

前　言

本文件按照 GB/T 1.1—2020《标准化工作导则　第 1 部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中关村华安关键信息基础设施安全保护联盟提出并归口。

本文件起草单位：奇安信科技集团股份有限公司、软极网络技术（北京）有限公司、中国工业互联网研究院、可信华泰科技有限公司、北京大学、长沙市轨道交通运营有限公司、北京市科学技术研究院、中国移动通信集团有限公司、韶关学院、中共扬州市委网络安全和信息化委员会办公室、扬州市大数据管理中心、北京柏睿数据技术股份有限公司、浙江脑动极光医疗科技有限公司。

本文件主要起草人：刘立、郑新华、赵志娟、毛庆梅、张海彬、任燕、苗德成、栾浩、杜君、雷健波、刘如才、赵菁华、王舒、王喆、郑旻、段古纳、王珊珊、腾迪、王思登、孙可侻、刘先宝、郑国忠、王雪珊、严爱明、赵晶晶、韩月华、王晓怡、张坤。

引　言

软件产品自主可控能够有效规避停服断供或安插后门等安全隐患，对于确保国家在关键领域的独立性和安全性意义重大。加强对软件自主可控能力的测试管理，掌握当前的实际水平和潜在风险，是提升自主可控能力必不可少的措施。制定本团体标准，旨在规范软件研制单位、使用单位、测评机构的测试行为，提升其测试管理能力，确保软件自主可控能力的稳步提升。

本文件围绕软件自主可控能力测试方法进行深入剖析，主要包括两大部分，一是软件自主可控能力测试的管理框架，包括测试范围、管理粒度、测试框架以及如何基于网络靶场对测试过程和测试结果进行闭环管理。二是基于网络靶场开展软件自主可控能力测试的具体方法，首先要建立资产库、组件库、漏洞库等基础库，对软件资产信息进行初始化，然后结合软件生命周期阶段和安全管理需求开展静态测试、仿真测试和实网测试，对每种测试方法从制定测试方案、搭建测试环境、执行测试任务到输出测试报告给予详细的指导，根据测试结果评定漏洞处置优先级，并对软件自主可控能力进行判定。

本文件旨在帮助软件研制单位、使用单位、测评机构提升软件自主可控能力测试过程的科学性和严谨性，加强测试结果的连续性管理，进而提升对软件产品的技术掌控能力、持续交付能力和漏洞管理能力，为数字经济的健康发展提供有力支撑。

自主可控网络安全技术　基于网络靶场的软件自主可控能力测试指南

1　范围

本文件描述了一套基于网络靶场对软件自主可控能力进行持续性测试的方法论，包括在软件生命周期各阶段进行静态测试、仿真测试、实网测试时，如何制定测试方案、搭建测试环境、执行测试任务和输出测试报告。

本文件适用于指导组织基于网络靶场开展软件资产自主可控能力测评工作，能供软件产品研制单位、使用单位、测评机构等相关方参考使用。

2　规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 20984—2022　信息安全技术　信息安全风险评估方法

GB/T 25069—2022　信息安全技术　术语

T/CSAC 001—2023　网络靶场　基于技战术模型的安全测评方法

T/ZISIA 01—2024　自主创新型网络安全技术　框架

3　术语和定义

GB/T 25069—2022 中界定的以及下列术语和定义适用于本文件。

3.1

自主可控　autonomous and controllable

产品在核心技术、原材料和零部件、生产工艺等方面不依赖于外部国家或公司，能够自主研发、生产、运营，并符合安全可控标准和监管要求。

3.2

安全可控　controllability for security

信息技术产品具备的保证其应用方的数据支配权、产品控制权、产品选择权等不受损害的属性。

[来源：GB/T 36630.1—2018，3.2]

3.3

软件资产　software asset

对组织有价值的软件资源，是自主可控测试的对象。

3.4

测试要素　test element

与测试活动相关的各种要素，包括但不限于测试目标、测试计划、测试人员、测试环境、测试方案、测试用例、测试工具和缺陷漏洞管理等。

3.5

软件成分分析　software composition analysis;SCA

用于识别、分析和追踪二进制软件的组成部分的技术。

注：SCA的目标是识别和清点开源软件的组件及其构成和依赖关系，并精准识别系统中存在的已知安全漏洞或潜在的许可证授权问题。

3.6

代码安全审计　code security audit

对代码进行安全分析，以发现代码安全缺陷或违反代码安全规范的动作。

[来源：GB/T 39412—2020，3.1.1]

3.7

代码自主率 code autonomy rate

软件代码中自主可控成分所占百分比。

3.8

开源许可证　open source license

一种具有法律效力的、允许用户自由使用、修改、复制或分发软件代码的授权条款格式合同或协议。

[来源：GB/T 43848—2024，3.3]

3.9

仿真测试　simulation test

使用网络靶场构建仿真环境，以模拟实际运行情况检测软件资产的安全性是否符合预期。

3.10

实网测试　real network test

在实际运行网络环境下对软件资产进行安全性测试。

3.11

安全众测　crowdsourcing security test

组织非特定的自然人或组织，在审计及监督下，对网络产品和系统开展漏洞发现等安全测试的过程。

3.12

网络靶场　cyber range

一种大型网络仿真测试平台。该平台支持：为网络安全技术和产品提供定量与定性评估，为网络安全培训和实验提供安全可控的设施，为模拟真实的网络攻防作战提供虚拟环境。

[来源：T/CSAC 002—2023，3.1，有修改]

4　缩略语

下列缩略语适用于本文件。

SBOM：软件物料清单（Software Bill of Materials）

CVE：通用漏洞披露（Common Vulnerabilities & Exposures ）

CWE：通用缺陷列表（Common Weakness Enumeration）

CNVD：国家信息安全漏洞共享平台（China National Vulnerability Database）

CNNVD：国家信息安全漏洞库（China National Vulnerability Database of Information Security）

5 概述

5.1 软件自主可控能力测试范围

软件自主可控能力测试包含对技术掌控能力、持续交付能力和网络安全能力的测试评估：

a） 技术掌控能力考察软件代码自主率；

b） 持续交付能力考察开源及第三方组件许可证合规性；

c） 网络安全能力考察代码缺陷、组件漏洞、资产漏洞及其他潜在的网络安全风险应对能力。

5.2 软件自主可控能力管理粒度

软件资产自主可控能力管理的粒度宜达到组件级，并明确处理组件间的依赖关系和交互。要求软件资产具备完整的 SBOM，能够清晰反映软件的构成及每个组件的情况，以实现自主可控能力评价和风险管理。

5.3 软件自主可控能力测试框架

软件自主可控能力测试宜持续开展，贯穿软件的开发阶段、测试阶段和运营阶段。

测试类型包括静态测试、仿真测试和实网测试。静态测试指在非运行环境下对软件源代码或二进制代码进行检测，包括成分分析和安全测试；仿真测试指在仿真运行环境下对运行态程序进行安全测试；实网测试指在实际网络环境下对运行态程序进行安全测试。静态测试结果包括代码来源、开源许可证有效性、组件级的代码缺陷和漏洞；仿真测试和实网测试的结果是软件资产的安全漏洞。

测试方法主要包括 SCA，代码安全审计，漏洞扫描，模糊测试、渗透测试和安全众测等。

在软件生命周期的各阶段可采用的测试类型和测试方法如表 1 所示。

表 1 软件生命周期各阶段可采用的测试类型和测试方法

生命周期阶段	开发阶段	测试阶段	运营阶段
测试类型	静态测试	静态测试/仿真测试	仿真测试/实网测试
测试方法	SCA、代码安全审计等	SCA、代码安全审计、漏洞扫描、模糊测试、渗透测试等	漏洞扫描、渗透测试、安全众测等

在软件生命周期中，各种测试均非一次性过程。在开发和测试阶段，测试活动宜迭代进行，直至达到自主可控能力基线要求；在运营阶段，可根据信息系统的安全等级要求进行周期性测试，此外当发生包括但不限于以下情况时，宜进行迭代测试：

a） 软件升级；

b） 与软件资产相关的漏洞库重大更新；

c） 出现以软件资产类型或所处行业为目标的新型攻击手段。

软件自主可控能力测试框架如图 1。

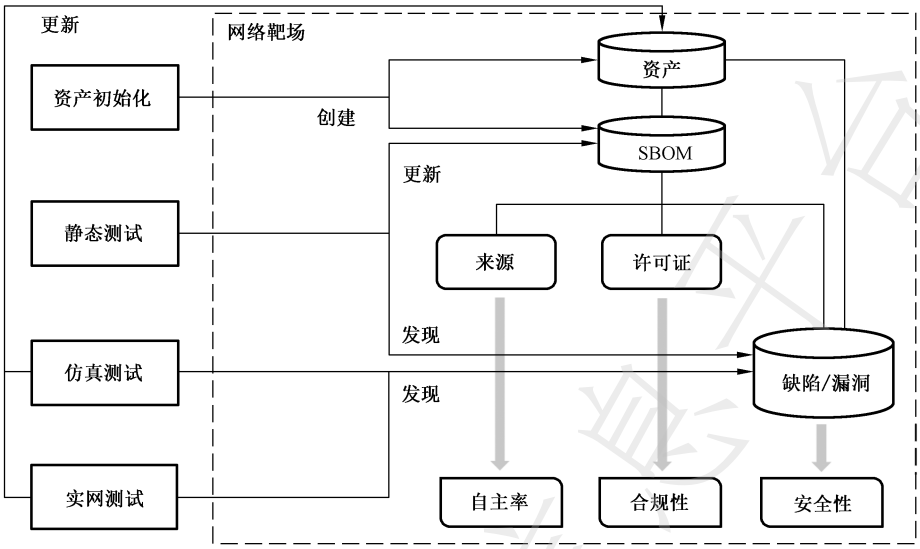

图 1　软件自主可控能力测试框架

5.4　软件自主可控能力评估

根据测试任务反馈的结果，采用定性评价、定量评价相结合的方法，评定漏洞处置优先级和软件自主可控能力等级，输出完整的测试评估报告，并为应对网络安全风险、提升软件自主可控能力制定短期计划和长期策略。

5.5　基于网络靶场的测试管理

宜根据 T/ZISIA 01—2024 的要求，基于网络靶场对软件资产自主可控能力的测试过程和测试结果进行闭环管理，包括但不限于以下内容。

a）　对测试过程进行监控和记录，能够查询复现测试环境、测试工具、测试人员动作、测试用例反馈等，支持审计回溯，从而保障测试过程的规范性，防范被测对象的机密性和完整性受损、隐瞒和不当利用测试发现的漏洞等情况。

b）　对测试结果进行关联性管理，静态测试、仿真测试和实网测试发现的安全漏洞通过软件资产的 SBOM 相互关联；对于组件级的风险，对其他采用相同组件的软件提示风险。

c）　对测试结果进行连续性管理，包括风险被发现的时间和方式、当前状态、修复情况、跟踪措施等。

6　软件资产自主可控能力测评方法

6.1　建立基础库

自主可控能力测试涉及多种测试要素，包括测试对象、测试任务、测试环境、测试资源、测试人员、测试结果等。为实现软件资产自主可控能力测试的闭环管理，需基于网络靶场平台建立基础库对各测试要素进行管理，并形成以资产为核心的测试要素关联关系，如图 2 所示。

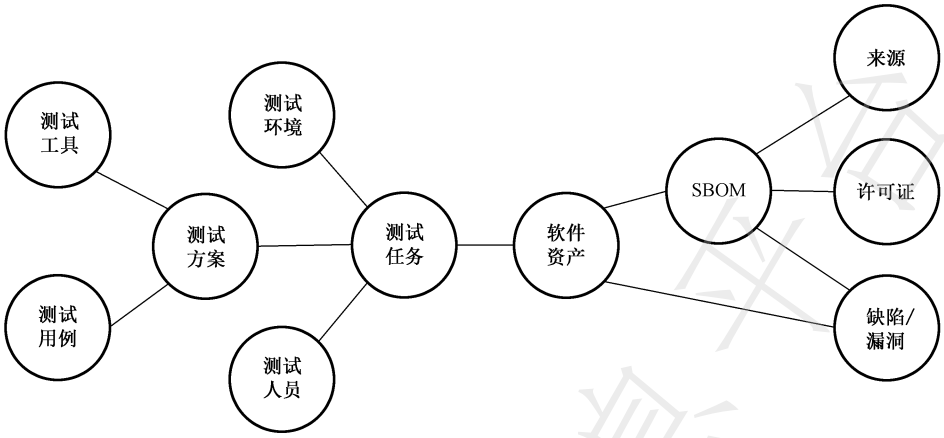

图 2　以资产为核心的测试要素关联关系

基础库包括资产库、组件库、漏洞库等，各基础库的构成参见附录 A、附录 B 和附录 C。

网络靶场需支持对基础库、各测试要素及其关联关系的管理。

6.2　资产信息初始化

初始化过程指将供应商提供的、未经测试验证的软件资产原始信息录入资产库和 SBOM，包括：

a）　软件资产的基础数据、组件关联信息和运行环境数据；

b）　资产组件的基础信息、资产关联信息和开源许可证信息。

6.3　静态测试

6.3.1　概述

静态测试包括 SCA 和代码安全审计。

SCA 分析的对象可以是源代码也可以是经过编译的二进制文件，分析目标是精确识别构成软件资产的组件，并进一步识别组件的来源、依赖关系、开源许可证及已知的缺陷和漏洞信息。

代码安全审计的对象是软件的源代码，测试目标是检测代码质量，发现代码存在的缺陷、漏洞等。

6.3.2　制定测试方案

根据资产类型和测试目标确定检测项列表，并选择相应的测试工具。

在静态测试的工具选择上，遵循如下原则：

a）　根据源代码所采用的编程语言，选取恰当的测试工具；

b）　结合所需的集成开发环境，选取恰当的测试工具；

c）　测试工具需能够被网络靶场管理和调用的接口，接收靶场下达的任务指令，并向靶场的数据采集接口实时传输测试过程数据；

d）　基于有利于全面发现代码缺陷和漏洞的原则，可采用多种测试工具和测试方法进行测试。

6.3.3　搭建测试环境

在网络靶场上搭建测试环境。关键步骤包括：

a）　虚拟节点管理和软件预装，构建测试工具的运行环境，并进行参数配置；

b）　配置测试流程及测试工具调用；

c）　配置数据采集策略。

6.3.4 执行测试任务

测试人员按照事前拟定的测试方案执行测试任务。

在测试执行过程中，通过网络靶场采集测试过程产生的相关数据，并记录测试结果，包括代码缺陷、安全漏洞、修复情况等。

6.3.5 输出测试报告

6.3.5.1 软件成分分析（SCA）

记录组件识别的结果，验证和更新 SBOM 信息。对于不具备原始组件信息的，可通过测试活动生成 SBOM。

将识别出的组件与公共组件库进行对比，获得以下结果：

a） 获取组件的来源信息，包括供应商信息和深度依赖关系，判断组件是否自主可控；

b） 获取组件的许可证信息，判断是否存在知识产权风险；

c） 发现组件中存在的已知漏洞，对于已具有修复方案进行修复，对尚无修复方案的，制定应对策略对潜在的风险进行管控。

6.3.5.2 代码安全审计

记录代码安全审计的结果，将发现的代码缺陷、安全漏洞、修复情况等记入漏洞库。

6.3.5.3 结果判定

综合所有组件的来源信息、许可证情况、缺陷和漏洞信息，依据事先制定的评价规则对技术掌控能力、持续交付能力和网络安全能力进行评价。

对于存在的停服断供风险、网络安全风险宜制定短期和长期的应对策略。

6.4 仿真测试

6.4.1 概述

仿真测试的对象是运行态程序。仿真测试目标是在软件上线前通过在仿真环境下进行安全测试，提前发现和修复在静态测试中难以发现的漏洞和潜在风险。

仿真测试采用的方式包括但不限于漏洞扫描、模糊测试和渗透测试。

6.4.2 制定测试方案

根据被测对象和测试目标构建指标树，选定测试方式，并选择恰当的测试用例、攻击技战术及测试工具。

在进行漏洞扫描时，遵循如下原则：

a） 结合被测对象的类型、已知的漏洞信息，选取恰当的漏洞扫描工具；

b） 结合历史测试数据，针对发现过的漏洞进行复测；

c） 选取恰当的测试用例和漏洞扫描工具；

d） 测试工具需能够被网络靶场管理和调用的接口，接收靶场下达的任务指令，并向靶场的数据采集接口实时传输测试过程数据。

在进行模糊测试时，遵循如下原则：

a） 测试工具能够自动生成随机的程序输入；

b） 测试工具能够检测程序的反馈；

c） 测试工具需能够被网络靶场管理和调用的接口，接收靶场下达的任务指令，并向靶场的数据采集接口实时传输测试过程数据。

在进行渗透测试时，遵循如下原则：

a） 结合被测对象的类型、已发生的攻击事件等信息，分析软件/软件组件易遭受的攻击类型，选择恰当的攻击方法；

b） 选取针对被测方所处行业的或目前活跃度较高的黑客组织，选取该组织惯用的攻击技术手法；

c） 排除技术复杂，无法在实验室环境中实现的攻击操作；

d） 选定攻击模拟软件、计划、测试用例；

e） 在工具选择上，宜尽量大程度上自动化还原整体攻击测试流程。

6.4.3 搭建测试环境

仿真测试需采用基于网络靶场搭建的仿真测试环境。测试人员宜根据测试需求，设计整体环境搭建方案，包括目标网络拓扑设计和资源规划，为搭建目标场景环境提供指导。测试环境搭建的具体流程宜参考 T/CSAC 001—2023，见图 3。

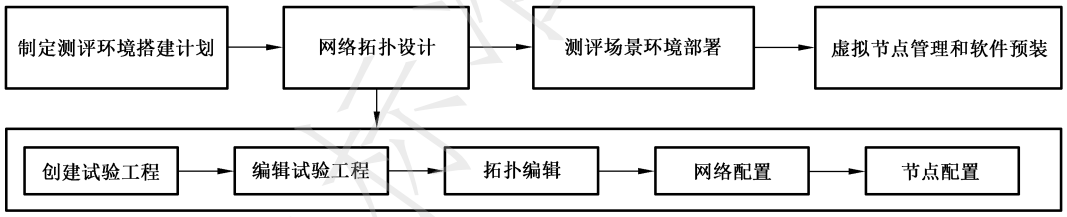

图 3 测试环境搭建流程

a） 制定测评环境搭建计划，计划内容包括但不限于：网络拓扑、资源分配、靶标设计、预装软件、任务制定、采集及检测范围、分析及评估对象等。

b） 网络拓扑设计，具体步骤包括：创建试验工程、试验工程编辑、拓扑编辑、网络配置、节点配置等。

c） 测评场景环境部署：依据测评环境搭建计划和网络拓扑设计进行部署。

d） 虚拟节点管理和软件预装。

6.4.4 执行测试任务

测试人员按照事前拟定的测试方案执行测试任务。

在执行漏洞扫描或模糊测试时，在靶场中配置测试流程，通过调用选定的测试工具，自动化执行测试任务。测试工具宜能生成满足测试任务所需格式和范围的数据。

在执行渗透测试时，由测试人员执行既定的攻击技术，模拟黑客的攻击行为，检测被测对象的响应。

在执行任务过程中，观察测试用例的反馈，完整采集测试过程产生的相关数据，包括但不限于日志类、流量信息、武器使用信息等。

6.4.5 输出测试报告

6.4.5.1 漏洞扫描

输出测试报告，详细记录测试发现的安全漏洞和潜在风险，报告宜包括如下内容：

a） 各漏洞的 CVE 编号、CWE 编号、CNVD 编号或 CNNVD 编号；

b） 漏洞的详细描述；

c） 漏洞的风险等级；

d） 漏洞的修复情况。

对于未能修复的漏洞，可持续关注公共漏洞库的更新情况，在具有修复办法后及时修复并复测。

6.4.5.2 模糊测试

输出测试报告，详细记录测试发现的安全漏洞和潜在风险，报告宜包括如下内容：

a） 漏洞的详细描述；

b） 漏洞的风险等级；

c） 处置优先级。

进一步分析定位缺陷，及时修复并复测。对于未能修复的缺陷漏洞，根据风险等级制定应对策略，并跟踪管理。

6.4.5.3 渗透测试

输出测试报告，详细记录测试发现的安全漏洞和潜在风险，报告宜包括如下内容：

a） 漏洞的详细描述，包含漏洞利用情况；

b） 漏洞的风险等级；

c） 处置优先级。

进一步分析定位风险原因，及时修复并复测。对于未能修复的缺陷漏洞，根据风险等级制定应对策略，并跟踪管理。

6.4.5.4 结果判定

综合漏洞扫描、模糊测试和渗透测试发现的所有漏洞信息，依据事先制定的评价规则对自主可控系统的网络安全能力进行评价。

依据评价结果判定软件是否达到上线运行的安全预期。同时，对于存在的网络安全风险宜制定短期和长期的应对策略。

6.5 实网测试

6.5.1 概述

实网测试是根据信息系统安全等级要求定期开展的周期性测试，测试范围内的资产均为被测对象。软件资产实网测试的对象是运行态程序，测试目的是通过周期性测试，进一步挖掘在先前测试中未暴露的漏洞和潜在风险。

实网测试采用的方式包括漏洞扫描、渗透测试和安全众测。

在实际运行网络环境下进行测试宜充分考虑对业务的影响，在确保不影响业务连续性的情况下或在约定的时间窗口内执行测试任务。实网测试宜预先制定突发情况下的应急处置方案。

6.5.2 制定测试方案

根据被测对象和测试目标构建指标树，选定测试方式，并选择恰当的测试用例、攻击技战术及测试工具。

在进行漏洞扫描时，遵循如下原则：

a） 结合被测对象的类型、已知的漏洞信息，选取恰当的漏洞扫描工具；

b） 结合历史测试数据，针对发现过的漏洞进行复测；

c） 选取恰当的测试用例和漏洞扫描工具；

d） 测试工具需能够被网络靶场管理和调用的接口，接收靶场下达的指令执行测试任务，并向靶场的数据采集接口实时传输测试过程数据。

在进行渗透测试时，遵循如下原则：

a） 选择一系列代表性技术进行测评；

b） 结合被测信息系统的类型、子系统类型、已发生的攻击事件等信息，分析系统/子系统易遭受的攻击类型，选择恰当的攻击方法；

c） 选取针对被测方所处行业的或目前活跃度较高的黑客组织，选取该组织惯用的攻击技术手法；

d） 选定攻击模拟软件、计划、测试用例；

e） 在工具选择上，宜尽量大程度上自动化还原整体攻击测试流程。

在进行安全众测时，遵循如下原则：

a） 选择的测试人员在擅长的技术方向上尽量多样化；

b） 对参与测试人员制定严密的授权管理策略；

c） 制定测试过程数据采集、存储的策略，以便进行回溯分析及后期取证；

d） 制定突发情况下的应急处置方案。

6.5.3 搭建测试环境

在实网测试中，被测对象运行在实网环境下，但网络靶场需对测试过程和结果进行监测和管理，包括测试工具调用、测试过程管理、数据采集和结果输出。

在网络靶场上配置测试环境。关键步骤包括：

a） 虚拟节点管理和软件预装，构建测试工具的运行环境，并进行参数配置；

b） 配置测试流程及测试工具调用；

c） 配置数据采集策略。

为应对实网测试中各种可能的突发事件，测试环境宜具备包含但不限于以下功能：

a） 依靠网络资源探测功能对被测对象的联通性、状态和信息等进行实时监测；

b） 依靠数据采集功能对测试环境进行实时监测，监测的指标包括：测试场景的拓扑合理性、网络连通性以及设备稳定性，并监测测试流程各步骤的状态和结果等；

c） 测试过程的复盘功能，以满足特殊情况下的测试工作审计需求。

6.5.4 执行测试任务

测试人员按照事前拟定的测试方案执行测试任务。

执行漏洞扫描时，在靶场中配置测试流程，通过调用选定的测试工具，自动化执行测试任务。

执行渗透测试时，由测试人员执行既定的攻击技术，模拟黑客的攻击行为，检测被测对象的响应。

执行安全众测时，由所有被授权的测试人员在审计监督下对被测系统进行攻击，测试人员自由选

择工具和技战术，从不同角度挖掘系统存在的安全漏洞。

在执行任务过程中，观察测试用例的反馈，完整采集测试过程产生的相关数据，包括但不限于日志类，流量信息，武器使用信息等。同时，实时监测被测对象的联通性和运行状态，一旦测试环境或被测对象发生故障或受到损害，影响关键业务连续性，宜立即停止测试活动，启动预先制定的应急处置方案。对于安全众测过程，重点检查测试人员是否有未授权的入侵网络、干扰网络正常功能、窃取网络数据等危害网络安全的行为或活动，并保存原始日志满足追溯要求。

6.5.5 输出测试报告

6.5.5.1 漏洞扫描

输出测试报告，详细记录测试发现的安全漏洞和潜在风险，报告宜包括如下内容：

a） 各漏洞的 CVE 编号、CWE 编号、CNVD 编号或 CNNVD 编号；

b） 漏洞的详细描述；

c） 漏洞的风险等级；

d） 漏洞的修复情况。

对于未能修复的漏洞，可持续关注公共漏洞库的更新情况，在具有修复办法后及时修复并复测。

6.5.5.2 渗透测试

输出测试报告，详细记录测试发现的安全漏洞和潜在风险，报告宜包括如下内容：

a） 漏洞的详细描述，包含漏洞利用情况；

b） 漏洞的风险等级；

c） 处置优先级。

进一步分析定位风险原因，及时修复并复测。对于未能修复的漏洞，制定应对策略，并跟踪管理。

6.5.5.3 安全众测

所有授权测试人员提交真实完整且描述清晰的漏洞信息，由测试组织方汇总信息输出完整的测试报告，详细记录测试发现的安全漏洞和潜在风险，报告宜包括如下内容：

a） 漏洞的详细描述，包含漏洞利用情况；

b） 漏洞的风险等级；

c） 受影响产品或服务；

d） 处置优先级；

e） 检测工具；

f） 修复办法。

及时修复测试发现的风险漏洞并复测。对于未能修复的漏洞，制定应对策略，并跟踪管理。

6.5.5.4 结果判定

综合漏洞扫描、渗透测试和安全众测发现的所有漏洞信息，依据事先制定的评价规则对自主可控系统的网络安全能力进行评价。

对于存在的网络安全风险宜制定短期和长期的应对策略。

6.6 漏洞处置优先级评定

对于测试发现的漏洞，其处置优先级与关联资产的重要性、漏洞等级、漏洞被利用的可能性相关，将这三方面的因素分别进行量化赋值。

a） 资产值：根据 GB/T 20984—2022，按照资产的重要性，取值区间为 1～5，重要性最高的资产赋值为 5，依次向下分为五个等级。

b） 漏洞值：它与漏洞等级相对应；漏洞等级，按照漏洞潜在的危害程度分为五级：超危、高危、中危、中低危和低危；相应地，漏洞值取值区间为 1～5，超危的漏洞赋值为 5，依次向下赋值。

c） 威胁值：按照漏洞易被发现和利用程度，即风险发生的可能性进行赋值，取值区间为 1～5，暴露性最强、最易于被利用的漏洞风险等级最高，赋值为 5，依次向下分为五个等级。

d） 漏洞风险值 = 资产值 × 漏洞值 × 威胁值。

风险值越高的漏洞越宜优先处置，避免造成较大的损害。由上述计算方法可知漏洞风险值的取值区间为 1～125，可按照取值的区段将处置优先级分为五个等级，见表 2。

表 2　处置优先级评定表

漏洞风险值	处置优先级
≥ 100	高
80～99	中高
60～79	中
40～59	中低
＜40	低

6.7　软件自主可控能力评定

软件自主可控能力主要考察技术掌控能力、持续交付能力和软件安全能力，将这三方面的能力分别进行量化评估。

a） 技术掌控能力值：依据代码自主率对技术掌控能力值进行评分，代码自主率＝（自主研发组件数＋来自国内的开源组件数）/软件组件总数，参考表 3。

表 3　技术掌控能力评分表

代码自主率	技术掌控能力值评分
≥95%	5
85%～94%	4
70%～84%	3
50%～69%	2
＜50%	1

b） 持续交付能力值：根据许可证合规率来确定，许可证合规率＝具有有效许可证的开源组件数/开源组件总数，可参考表 3 的形式依据开源许可证合规率制定持续交付能力评分表，评分取值区间为 1～5。

c） 软件安全能力值：软件风险值与漏洞数、每个漏洞的严重性和被利用的概率有关，漏洞数量越多、等级越高、越容易被利用，则软件风险值越高，软件安全能力值评分越低，见表 4。在进行软件安全能力值评分时，已被修复的漏洞不计算在内，仅考虑待处置漏洞引起的安全性风险。

软件风险值＝漏洞值 1 × 威胁值 1 ＋ 漏洞值 2 × 威胁值 2 ＋ 漏洞值 n × 威胁值 n，其中每个漏洞造成的软件风险值取值区间为 1～25。

表 4　软件安全能力评分表

软件风险值	软件安全能力值评分
＜ 20	5
20 ～ 39	4
40 ～ 59	3
60 ～ 79	2
≥80	1
注：每个分数等级对应的取值区段可依实际情况而定。	

d）　自主可控能力值 = 技术掌控能力值 ×持续交付能力值 ×软件安全能力值

由上述计算方法可知自主可控能力值的取值区间为 1～125，可按照取值的区段将软件自主可控能力分为五个等级，参见表 5。

表 5　软件自主可控能力评定表

自主可控能力值	软件自主可控能力等级
≥ 100	高
80 ～ 99	中高
60 ～ 79	中
40 ～ 59	中低
＜ 40	低

附 录 A
（资料性）
资产库

资产库宜包含但不限于以下维度的数据：

资产名称	
供应商	
资产类型	
型号	
版本	
描述	
资产组件信息	组件名称
运行环境信息	
资产漏洞信息	漏洞名称
历史测试数据	任务名称（关联测试时间、测试环境、测试人员、测试用例、测试工具、测试结果等）

附 录 B
（资料性）
组件库

SBOM 宜包含但不限于以下信息：

组件名称	
供应商	
作者	
时间戳	
版本字符串	
唯一标识符	
组件关系	深度依赖关系
版本哈希值	
关联资产	
许可证信息	
代码缺陷	
组件漏洞	

附 录 C
（资料性）
漏洞库

漏洞宜能够进行关联管理，组件级的漏洞宜能够关联到所有相关资产，资产级的漏洞宜通过进一步测试定位关联到组件。

漏洞库宜包含但不限于以下信息：

漏洞名称	
主机IP	
漏洞公共名称	CVE/CWE/CNVD/CNNVD
漏洞评级	参考GB/T 30279，分为超危/高危/中危/中低危/低危
处置优先级	高/中高/中/中低/低
漏洞类型	后门/发现备份文件/缓冲区溢出/命令执行/代码执行/目录遍历/目录穿越/拒绝服务攻击/文件创建/文件删除/文件下载/文件读取/文件上传/文件遍历/信息搜集/不当的输入验证/LDAP注入/本地文件包含/登录绕过/路径泄露/提权攻击/重定向/远程密码修改/解析错误/远程文件包含/安全绕过/SQL注入/服务器端包含注入/服务器端请求伪造/未授权访问/XML注入/Xpath注入/跨站脚本攻击/注入攻击/远程代码执行/信息泄露攻击/配置错误/弱口令攻击/其他攻击
漏洞状态	发现/待验证/待修复/已修复
关联资产	
关联组件	
关联网络	
关联服务	
发现工具	
记录人	
记录时间	
描述	

参 考 文 献

[1] GB/T 30279 信息安全技术 网络安全漏洞分类分级指南
[2] GB/T 36630.1—2018 信息安全技术 信息技术产品安全可控评价指标 第1部分:总则
[3] GB/T 39412—2020 信息安全技术 代码安全审计规范
[4] GB/T 43848—2024 网络安全技术 软件产品开源代码安全评价方法
[5] T/CSAC 002—2023 网络靶场 能力分级指南