Linux系统安全加固

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Information** | | | | | |
| **Document Name** | | **Linux系统安全加固** | | | |
| **Document Classification** | | Technical documentation | | | |
| **Scope of Application** | |  | | | |
| **Approval and release** | | | | | |
| **Document Version** | **Audit Date** | | **Editor** | **Inspector** | Approver |
| V1.0 | 2019-03-21 | | dongfeng |  |  |
|  |  | |  |  |  |
|  |  | |  |  |  |
| **Compilation Records** | | | | | |
| **Records** | **Date** | | **Person** | | Description |
|  |  | |  | |  |

# 前言

## 术语

|  |  |  |
| --- | --- | --- |
| **英文缩写** | **英文全称** | **中文全称** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 目的

本文档用来指导加固linux系统，以保证系统安全运行。

# 协议安全

## OpenSSH

### OpenSSH MaxAuthTries限制绕过漏洞(CVE-2015-5600)

OpenSSH（OpenBSD Secure Shell）是OpenBSD计划组所维护的一套用于安全访问远程计算机的连接工具。该工具是SSH协议的开源实现，支持对所有的传输进行加密，可有效阻止窃听、连接劫持以及其他网络级的攻击。

OpenSSH 6.9及之前版本的sshd中的auth2-chall.c文件中的‘kbdint\_next\_device’函数存在安全漏洞，该漏洞源于程序没有正确限制处理单链接中的keyboard-interactive设备。远程攻击者可借助ssh -oKbdInteractiveDevices选项中较长且重复的列表利用该漏洞实施暴力破解攻击，或造成拒绝服务（CPU消耗)。

规避措施：升级openssh组件到最新版本（yum install -y openssh）

### OpenSSH 远程权限提升漏洞(CVE-2016-10010)

OpenSSH 是一组用于安全地访问远程计算机的连接工具。

OpenSSH < 7.4版本禁用特权分离后，sshd服务在实现上存在权限提升漏洞，可使攻击者通过转发Unix域套接字，获取root权限。

规避措施：升级openssh组件到最新版本（yum install -y openssh）

### OpenSSH sshd mm\_answer\_pam\_free\_ctx释放后重利用漏洞(CVE-2015-6564)

OpenSSH是SSH协议的开源实现。

非OpenBSD平台上OpenSSH 7.0之前版本，sshd内monitor.c的mm\_answer\_pam\_free\_ctx函数存在释放后重利用漏洞，这可使本地用户利用sshd uid控制，发送构造的MONITOR\_REQ\_PAM\_FREE\_CTX请求，获取提升的权限。

规避措施：升级openssh组件到最新版本（yum install -y openssh）

### OpenSSH 'x11\_open\_helper()'函数安全限制绕过漏洞(CVE-2015-5352)

OpenSSH是SSH (Secure SHell)协议的免费开源实现。SSH协议族可以用来进行远程控制，或在计算机之间传送文件。

OpenSSH 6.9之前版本，使用非ForwardX11Trusted模式时，ssh channels.c内的函数x11\_open\_helper没有检查X连接的拒绝截止日期，这可使远程攻击者通过允许时间窗口外的连接，利用此漏洞绕过目标访问限制。

规避措施：升级openssh组件到最新版本（yum install -y openssh）

### OpenSSH <=7.2p1 xauth命令注入漏洞(CVE-2016-3115)

OpenSSH是SSH协议的开源实现。

OpenSSH <=7.2p1在实现上存在xauth命令注入漏洞，可导致绕过forced-command及/bin/false。

规避措施：升级openssh组件到最新版本（yum install -y openssh）

### OpenSSH 安全漏洞(CVE-2016-1908)

OpenSSH（OpenBSD Secure Shell）是OpenBSD计划组所维护的一套用于安全访问远程计算机的连接工具。该工具是SSH协议的开源实现，支持对所有的传输进行加密，可有效阻止窃听、连接劫持以及其他网络级的攻击。

OpenSSH中存在安全漏洞，该漏洞源于OpenSSH客户端没有正确为不可信的X11转发生成身份验证cookie。攻击者可利用该漏洞与本地X服务器建立可信连接。

规避措施：升级openssh组件到最新版本（yum install -y openssh）

### OpenSSH 远程代码执行漏洞(CVE-2016-10009)

OpenSSH 是一组用于安全地访问远程计算机的连接工具。

OpenSSH < 7.4版本ssh-agent未对加载的 PKCS#11 模块进行验证，在实现上存在任意执行代码漏洞。可使攻击者在受影响应用上下文中执行任意代码。

规避措施：升级openssh组件到最新版本（yum install -y openssh）

### OpenSSH do\_setup\_env函数权限提升漏洞(CVE-2015-8325)

OpenSSH是SSH协议的开源实现。

OpenSSH 7.2p2之前版本, sshd/ session.c/ do\_setup\_env函数存在安全漏洞, 启用了UseLogin功能且PAM配置为读取用户主目录内的 .pam\_environment文件后, 本地用户通过触发/bin/login程序构造的环境, 可提升其权限。

规避措施：升级openssh组件到最新版本（yum install -y openssh）

### OpenSSH auth\_password函数拒绝服务漏洞(CVE-2016-6515)

OpenSSH是SSH协议的开源实现。

OpenSSH 7.3之前版本，sshd/auth-passwd.c/auth\_password函数未限制密码验证中的密码长度，远程攻击者通过较长的字符串，利用此漏洞可造成拒绝服务。

规避措施：升级openssh组件到最新版本（yum install -y openssh）

## Redis

Redis 默认情况下，会绑定在 0.0.0.0:6379，这样将会将 Redis 服务暴露到公网上，如果在没有开启认证的情况下，可以导致任意用户在可以访问目标服务器的情况下未授权访问 Redis 以及读取 Redis 的数据。攻击者在未授权访问 Redis 的情况下可以利用 Redis 的相关方法，可以成功在 Redis 服务器上写入公钥，进而可以使用对应私钥直接登录目标服务器。

规避措施：

1.配置bind选项，限定可以连接Redis服务器的IP，修改 Redis 的默认端口6379

2.配置认证，也就是AUTH，设置密码，密码会以明文方式保存在Redis配置文件中

3.配置rename-command 配置项 “RENAME\_CONFIG”，这样即使存在未授权访问，也能够给攻击者使用config 指令加大难度

4.Redis作者表示将会开发”real user”，区分普通用户和admin权限，普通用户将会被禁止运行某些命令，如config

# 配置规范

## 账号/认证/授权

### 3.1.1 密码策略

修改  /etc/login.defs 配置文件

PASS\_MAX\_DAYS      90               密码最长有效期

PASS\_MIN\_DAYS       10               密码修改之间最小的天数

PASS\_MIN\_LEN         8                密码长度

PASS\_WARN\_AGE     7                 口令失效前多少天开始通知用户修改密码

### 3.1.3 密码强度

修改/etc/pam.d/system-auth文件,在ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1 选3种，追加到password requisite pam\_cracklib.so后面，添加到配置文件中。

例如：password requisite pam\_cracklib.so ucredit=-1 lcredit=-1 dcredit=-1

注：ucredit：大写字母个数；lcredit：小写字母个数；dcredit：数字个数；ocredit：特殊字符个数

### 3.1.4 限制密码尝试次数

打开 /etc/pam.d/sshd  文件，在 #%PAM-1.0 的下面，加入下面的内容，表示当密码输入错误达到3次，就锁定用户150秒，如果root用户输入密码错误达到3次，锁定300秒。

“auth required pam\_tally2.so deny=3 unlock\_time=150 even\_deny\_root root\_unlock\_time300”

### 3.1.5 禁止root远程登录

修改/etc/ssh/sshd\_config文件,配置PermitRootLogin no

重启服务，systemctl restart sshd

### 3.1.6 命令行界面超时退出

以root账户执行，vi /etc/profile,增加 export TMOUT=300(单位：秒，可根据具体情况设定超时退出时间，要求不小于300秒),注销用户，再用该用户登录激活该功能。

### 3.1.7 历史命令保存条数

以root账户执行，vi /etc/profile,修改HISTSIZE=100

### 3.1.8 限制用户su到root

检查是否使用PAM认证模块禁止wheel组之外的用户su为root。

编辑su文件(vi /etc/pam.d/su)，在开头添加下面两行：

auth sufficient pam\_rootok.so

auth required pam\_wheel.so group=wheel

这表明只有wheel组的成员可以使用su命令成为root用户。

你可以把用户添加到wheel组，以使它可以使用su命令成为root用户。

添加方法为：usermod –G wheel username

### 3.1.9 删除和锁定不必要的用户

使用命令 userdel  -r 用户名  删除不必要的账号

使用命令 passwd -l  用户名  锁定不必要的账号

### 3.1.10 锁定重要文件

chattr  +i  /etc/passwd /etc/shadow

## 日志审计

### 远程日志服务器

日志服务器的好处在于，每个工作服务器将自己的日志信息发送给日志服务器进行集中管理，即使有 人入侵了服务器并将自己的登录信息悄悄删除，但由于日志信息实时与日志服务器同步，保证了日志 的完整性。以备工作人员根据日志服务器信息对服务器安全进行评测。



客户端的配置：

打开 /etc/rsyslog.conf 配置文件

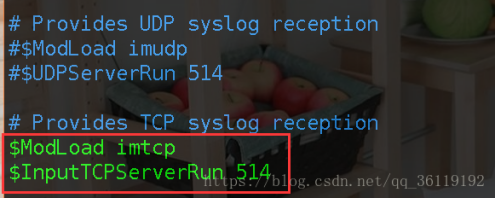
把哪种类型的日志文件发送给服务端，你就把他原来的对应的目录改成： @192.168.10.20

把所有的日志文件都发送给服务器的话，在文件最后加上： \*.\*  @@192.168.10.20:514

最后重启rsyslog服务： systemctl  restart  rsyslog

服务端的配置：

打开 /etc/rsyslog.conf 配置文件，将这里的注释给去掉



最后重启rsyslog服务： systemctl  restart  rsyslog

# 防火墙设置

## WCG防火墙

### WCG防火墙介绍

WCG没有使用系统自带的防火墙，而是使用iptables规则设置自己的防火墙。

50683端口：用于远端ssh登录

50685端口：用于web登录

500端口+4500端口：ipsec协商报文和数据报文

36412端口：WCG信令报文

2152端口：WCG数据报文

#### 清除现有的iptables规则并使计数归零

function clear\_iptables()

{

iptables -F

iptables -F -t nat

iptables -X

iptables -X -t nat

iptables -Z

iptables -Z -t nat

}

#### 设置默认iptables策略

function default\_policy\_iptables()

{

iptables -P INPUT ACCEPT

iptables -P OUTPUT ACCEPT

iptables -P FORWARD ACCEPT

}

#### 指定端口设置iptables规则

function add\_firewalld\_iptables()

{

local interface\_name=$1

iptables -A INPUT -p tcp --dport 50683 -i ${interface\_name} -j ACCEPT

iptables -A INPUT -p tcp --dport 50685 -i ${interface\_name} -j ACCEPT

iptables -A INPUT -p udp --dport 500 -i ${interface\_name} -j ACCEPT

iptables -A INPUT -p udp --dport 4500 -i ${interface\_name} -j ACCEPT

iptables -A INPUT -p sctp --dport 36412 -i ${interface\_name} -j ACCEPT

iptables -A INPUT -p udp --dport 2152 -i ${interface\_name} -j ACCEPT

iptables -A INPUT -m state --state RELATED,ESTABLISHED -i ${interface\_name} -j ACCEPT

iptables -A INPUT -p icmp -i ${interface\_name} -j ACCEPT

iptables -A INPUT -p esp -i ${interface\_name} -j ACCEPT

[[ $interface\_name ]] && iptables -A INPUT -p all -i ${interface\_name} -j DROP

}

#### 指定端口删除iptables规则

function add\_firewalld\_iptables()

{

local interface\_name=$1

iptables -D INPUT -p tcp --dport 50683 -i ${interface\_name} -j ACCEPT

iptables -D INPUT -p tcp --dport 50685 -i ${interface\_name} -j ACCEPT

iptables -D INPUT -p udp --dport 500 -i ${interface\_name} -j ACCEPT

iptables -D INPUT -p udp --dport 4500 -i ${interface\_name} -j ACCEPT

iptables -D INPUT -p sctp --dport 36412 -i ${interface\_name} -j ACCEPT

iptables -D INPUT -p udp --dport 2152 -i ${interface\_name} -j ACCEPT

iptables -D INPUT -m state --state RELATED,ESTABLISHED -i ${interface\_name} -j ACCEPT

iptables -D INPUT -p icmp -i ${interface\_name} -j ACCEPT

iptables -D INPUT -p esp -i ${interface\_name} -j ACCEPT

[[ $interface\_name ]] && iptables -D INPUT -p all -i ${interface\_name} -j DROP

}

### WCG防火墙开启和关闭

所有规则的设置都通过/root/eGW/Tools/set\_iptables进行配置。

使用方法如下：

[dongfeng@localhost Tools]$ ./set\_iptables -h

usage:

-h --help list help information

-v --version list the released version

-i --init initialze iptables

-a --add [interface] add iptables to the interface

-d --delete [interface] del iptables to the interface

例如:

eth0接口初始化iptables：./set\_iptables -i

eth0接口添加规则：./set\_iptables -a eth0

eth0接口删除规则：./set\_iptables -d eth0

## EPC防火墙