悬镜服务器卫士 V2.0.0

技术白皮书

北京安普诺信息技术有限公司

（http://www.x-mirror.cn/）

二〇一六年

目 录

1. 前　言

随着 Internet 应用的广泛深入，信息安全问题日益引起人们的高度重视。

目前常见的网络安全防范措施主要是采用防火墙、入侵检测系统、防病毒等安全产品。然而网络安全是一项系统的工程，上述产品只能针对某一方面，解决部分安全问题。如今，黑客完全有可能透过防火墙将黑客工具装入网络发动内部进攻。许多黑客也综合采用多种手段来攻击，如果只解决某一方面的安全问题，不能起到真正的安全作用，所以必需采用系统的解决方法。

信息安全的最终目的就是对信息数据的安全保护。而和信息数据安全相接近的就是操作系统的安全。当系统遭受了攻击，就可能造成重要的数据、资料丢失，关键的服务器丢失控制权等极其严重的后果。

Linux 是一种类 Unix 的操作系统。源代码开放，是在自由和开放的氛围中发展起来的。由于具有开放源代码的免费的特点。可以说，今天的这个完善并强大的 Linux 完全是一个热情、自由、开放的网络产物。由此受到越来越多用户的欢迎。随着 Linux 操作系统在我国的不断普及。采用 Linux 网络操作系统作为服务器的用户也越来越多。这因为 Linux 不仅包含 Windows 的所有功能（甚至包括域登录），而且在很多方面比 Windows 操作系统更稳定。这一点在连续工作的服务器类型的系统中表现得尤其明显。另外，Linux 属于开源操作系统。所以从可靠性上来讲，更适合政府、军事和金融等关键性机构使用。我们不难预测今后 Linux 操作系统在我国将得到更快更大的发展，Linux 面临着前所未有的发展机遇，同时，也面临着更多的安全隐患，随之而来的信息安全问题也日益突出。Linux 是一个开放式系统，可以在网络上找到许多现成的程序和工具，这既有益于用户，也方便了黑客。因为他们也能很容易地找到程序和工具来潜入 Linux 系统或者通过系统漏洞来盗取 Linux 系统上的重要信息。不过，只要我们仔细地设定 Linux 的各种系统功能，并且加上必要的安全技术措施，以及经常性的安全检查就能最大限度地保证系统的安全性。因此，详细分析 Linux 系统的安全机制，找出它可能存在的安全隐患，给出相应的安全策略和保护措施是十分必要的。

2. 研发背景

计算机安全涉及到的各方面内容中，操作系统、网络系统、数据库管理系统的安全是主要问题，其中操作系统安全尤为关键。操作系统是计算机资源的直接管理者，所有应用软件都是基于操作系统来运行的，不能保障操作系统安全，也就不能保障数据库安全、网络安全及其他应用软件的安全问题。因此构建一个安全的操作系统，成为提高计算机信息系统安全性的重要手段。

操作系统安全是计算机网络系统安全的基础。而服务器以及业务数据又是被攻击的最终目标。因此，部署安全产品，加强对关键服务器的安全控制，是增强系统总体安全性的核心一环。

近年来 Linux 系统由于其出色的性能和稳定性，开放源代码的特性带来的灵活性和可扩展性，以及较低廉的成本，而受到计算机工业界的广泛关注和应用。成为了大部分服务器操作系统的首选，但在安全性方面，Linux 内核只提供了传统 Unix 自主访问（root 用户，用户 id，模式位安全机制），以及部分的支持了 POSIX.le 标准草案中的 Capabilities 安全机制，无法满足日益紧迫的信息安全问题的需求，而 Linux 系统因为在中国的用户量相对 Windows 系统较少，导致针对 Linux 系统尤其是针对 Linux 系统的服务器的安全软件较少，并且安装使用起来也非常的繁琐和复杂。

针对以上问题，北京安普诺信息技术有限公司推出了针对 Linux 服务器的悬镜服务器卫士。目的是为 Linux 服务器保驾护航，并且用户可以简单直观的操作这款软件。

悬镜服务器卫士简称 Xmirror，综合考虑了 Linux 服务器上的安全问题，是针对 Linux 服务器提出的一套安全解决方案，悬镜服务器卫士可以提供给系统安全管理员最便捷、最直观、最有效的方法，保证 Linux 服务器最大限度的安全。

3. 产品概述

3.1 应用技术介绍

3.1.1 SSL 安全隧道

SSL 是英文 Secure Sockets Layer 的缩写，中文含义是安全套接层协议，可以在 Internet 上提供秘密性传输。Netscape 公司在推出第一个 Web 浏览器的同时，提出了 SSL 协议标准，其目标是保证两个应用间通信的保密性和可靠性，可在服务器端和用户端同时实现支持，已经成为 Internet 上保密通讯的工业标准。

SSL 能使用户/服务器应用之间的通信不被攻击者窃听，并且始终对服务器进行认证，还可以选择对用户进行认证。SSL 协议要求建立在可靠的传输层协议(TCP)之上。SSL 协议的优势在于它是与应用层协议独立无关的，高层的应用层协议(例如：HTTP，FTP，TELNET 等)能透明地建立于 SSL 协议之上。SSL 协议在应用层协议在通信之前就已经完成加密算法、通信密钥的协商以及服务器认证等工作。在此之后应用层协议所传送的数据都会被加密，从而保证通信的私密性。

SSL 协议是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及 SSL 协议，并提供丰富的应用程序供测试或其它目的使用。

悬镜服务器卫士中使用 TLSv1 协议，与 SSLv3 协议相比，安全度更高，加密性更好，证书采用动态、非文件类型，使用不同的复本，在每次运行时都有不同的公、密钥对，极大提高数据安全性。

3.1.2 Hook 技术

Hook 技术其实就是修改函数行为的一种技术，通过对函数行为的修改可以实现：文件的监控/保护 、进程的监控/隐藏等，大多数的安全软件都是基于这种技术实现的，还有些病毒木马技术也会涉及到 Hook 技术。一般的 Hook 函数的流程：

```
{目标 API} -【Hook API】 -修改函数入口前几个字节，添加跳转指令 类似 JMP DWORD Ptr Address
                                \
                                { jump 我们的代码处 }
```

```
|- a.  执行我们代码
|- b.  修复 Hook
|- c.  调用目标 API
```

通过修改目标函数的前几个字节，然后跳转到指定代码执行，等执行完定制的代码后，再调用真实的目标 API 返回结果给调用程序。

大多数的系统 API，前几个字节都是对堆栈的操作，而 Hook 技术就是利用了这几个字节实现的跳转，这是一种比较常用的 Hook 思路，还有修改函数中部或者尾部的，思路相同，只是 patch 的位置不同，还有种 Hook 的思路是通过“跳板”函数实现，定制函数和目标函数之间的关系：

```
{ Hook API }
      \
    { Jump  定制函数 }
          \
       { Jump Trampoline(跳板函数) }
              \___{ Call   目标函数   }    --- { 结果返回给调用程序 }
```

成功 Hook 函数后，跳转到定制的函数中执行，当我们定制的代码执行完后，并不是在函数内部调用原目标函数，而是调用一个叫 Trampoline（）的函数，Trampoline（）的任务就是平衡堆栈，然后执行原目标函数，最后将结果返回给调用程序。

3.1.3 强制访问控制技术

Linux 访问控制策略采用的是基于访问模式位的自主访问控制，在自主访问控制策略中，进程能执行操作的权限完全依赖于所属用户的身份。

我们说一个操作系统是安全的，系指它能满足某一给定的安全策略。安全模型则是对安全策略所表达的安全需求的简单、抽象和无歧义的描述，它为安全策略与其实现机制的关联提供了一种框架。因此，安全模型是设计安全操作系统的基础之一。Bel1-Lapadula 模型(简称 BLP 模型)是 D.Elliott Bell 和 Leonard J．LaPadula 于 1973 年创立的一种模拟军事安全策略的计算机操作模型，其目标是详细描述计算机信息系统的多级操作规则。它是最早、也是最常用的典型的计算机多级安全模型，目前已被实现和应用于许多安全操作系统中。BLP 模型的安全策略包括两部分：

自主安全策略和强制安全策略。

在军事安全管理中，通常将文件的保密级别按从高到低顺序分为绝密级 (Top Secret)、秘密级 (Secret)、机密级(Confidentia1)和公开级(Unclassified)等四个级别，同时指定文件的流通领域。为了确定是否应该允许某人读或写一个文件，要把该人的许可证同文件的密级进行比较， 要遵循“下读上写”的原安全策略是访问控制的依据，下面依据 BLP 模型来举例定义安全策略。BLP 模型模拟这一军事安全策略，将每个客体和主体都分配一个安全级。

(1)强制访问控制信息和访问控制策略为了实现强制访问控制机制，必须为已有 Linux 系统中的主客体增加额外的控制信息。同时，需要添加强制访问控制策略，作为评判访问是否能够被允许的依据。(2)访问控制增强模块在新添加的强制访问控制信息和访问控制策略的基础上，本系统采用 LKM 的方法，为已有 Linux 系统增加强制访问控制模块(访问控制增强模块)，即通过替换原有安全相关的系统调用，实现对访问控制的增强。访问控制增强模块将在原有 Linux 的安全功能之上提供一层强制访问控制来提高 Linux 系统的安全性。该模块是基础服务，将运行于核心态。(3) 访问控制信息管理模块为了管理新添加的强制访问控制信息，强制访问控制管理模块，该模块运行在用户态，为安全管理员提供管理、配置强制访问控制信息的接口命令。通过这些管理命令，可以指定、修改和删除主客体的安全级和范畴。

3.1.4 Netfiler/Iptables 过滤技术

Netfilter/Iptables(简称为 Iptables)组成 Linux 平台下的包过滤防火墙，与大多数的 Linux 软件一样，这个包过滤防火墙是免费的，它可以代替昂贵的商业防火墙解决方案，完成封包过滤、封包重定向和网络地址转换(NAT)等功能 Netfilter/Iptables IP 信息包过滤系统被称为单个实体，但它实际上由两个组件 Netfilter 和 Iptables 组成。Netfilter 组件也称为内核空间(kernel space)，是内核的一部分，由一些信息包过滤表组成，这些表包含内核用来控制信息包过滤处理的规则集。Iptables 组件是一种工具，也称为用户空间(user space)，它使插入、修改和除去信息包过滤表中的规则变得容易。通过使用用户空间，可以构建自己的定制规则，这些规则存储在内核空间的信息包过滤表中。

Netfilter/Iptables 可以对流入和流出的信息进行细化控制，且可以在一台低配置机器上很好地运行，被认为是 Linux 中实现包过滤功能的第四代应用程序。Netfilter/Iptables 包含在 Linux 2.4 以后的内核中，可以实现防火墙、NAT（网络地址翻译）和数据包的分割等功能。Netfilter 工作在内核内部，而 Iptables 则是让用户定义规则集的表结构。Netfilter/Iptables 从 Ipchains 和 Ipwadfm（IP 防火墙管理）演化而来，功能更加强大。

这里所说的 Iptables 是 Ipchains 的后继工具，但具有更强的可扩展性。内核模块可以注册一个新的规则表（Table），并要求数据包流经指定的规则表。这种数据包选择用于实现数据报过滤（Filter 表），网络地址转换（NAT 表）及数据报处理（Mangle 表）。Linux 2.4 内核及其以上版本提供的这三种数据报处理功能都基于 Netfilter 的钩子函数和 IP 表，都是相互间独立的模块，完美地集成到了由 Netfilter 提供的框架中，Netfilter 主要提供了如下三项功能：

包过滤：Filter 表格不会对数据报进行修改，而只对数据报进行过滤。Iptables 优于 Ipchains 的一个方面就是它更为小巧和快速。它是通过钩子函数 NF_IP_LOCAL_IN、NF_IP_FORWARD 及 NF_IP_LOCAL_OUT 接入 Netfilter 框架的。

NAT：NAT 表格监听三个 Netfilter 钩子函数：NF_IP_PRE_ROUTING、NF_IP_POST_ROUTING 及 NF_IP_LOCAL_OUT。NF_IP_PRE_ROUTING 实现对需要转发数据报的源地址进行地址转换，而 NF_IP_POST_ROUTING 则对需要转发的数据报目的地址进行地址转换。对于本地数据报目的地址的转换，则由 NF_IP_LOCAL_OUT 来实现。

数据报处理：Mangle 表格在 NF_IP_PRE_ROUTING 和 NF_IP_LOCAL_OUT 钩子中进行注册。使用 Mangle 表，可以实现对数据报的修改或给数据报附上一些外带数据。当前 Mangle 表支持修改 TOS 位及设置 skb 的 nfmard 字段。

Iptables 的基础知识:

1、规则（rules）它其实就是网络管理员预定义的条件，规则一般的定义为“如果数据包头符合这样的条件，就这样处理这个数据包”。规则存储在内核空间的信

息包过滤表中，这些规 则分别指定了源地址、目的地址、传输协议（如 TCP、UDP、ICMP）和服务类型（如 HTTP、FTP 和 SMTP）等。当数据包与规则匹配 时，Iptables 就根据规则所定义的方法来处理这些数据包，如放行（accept）、拒绝（reject）和丢弃（drop）等。配置防火墙的主要工 作就是添加、修改和删除这些规则 。

2、链（chains） 它是数据包传播的路径，每一条链其实就是众多规则中的一个检查清单，每一条链中可以有一条或数条规则。当一个数据包到达一个链时，Iptables 就会从链中第一条规则开始检查（即：检查的顺序：从上到下），看该数据包是否满足规则所定义的条件。如果满足，系统就会根据该条规则所定义的方法处理该数据包；否则 Iptables 将继续检查下一条规则，如果该数据包不符合链中任一条规则，Iptables 就会根据该链预先定义的默认策略来处理数据包。

3、表（tables） 它提供特定的功能，Iptables 内置了 4 个表，即 filter 表、nat 表、mangle 表和 raw 表，分别用于实现包过滤，网络地址转换、包重构(修改)和数据跟踪处理 。

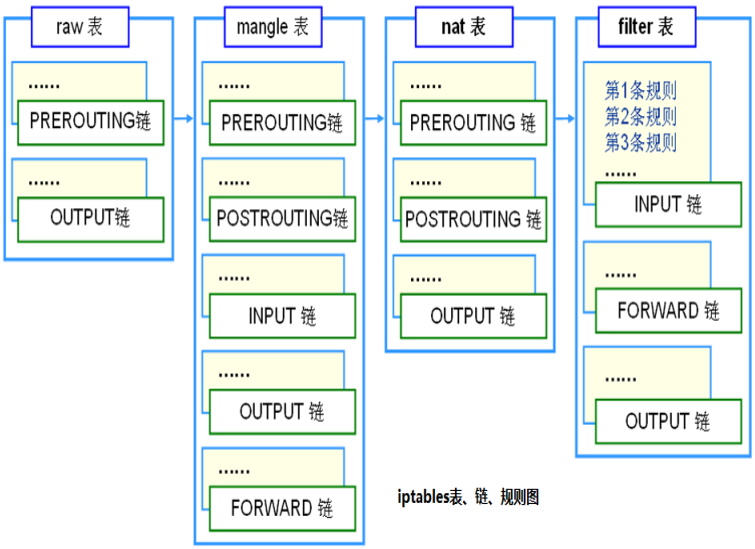

图 3-1　Iptables 表、链、规则图

3.1.5 安全会话技术

悬镜服务器卫士软件中安全会话 Session 技术的实现类似于 Web 服务器的 Session 机制，对于前端每次正确的登录，后端都会生成一个唯一的 Session code，前端登录后的每次请求都要携带这个 Session code，Session code 有一个生效时间，

这个生效时间的意思是前端在这个时间内没有做任何请求，那么这个 Session code 就会失效。

3.2 Xmirror For Linux 体系结构

3.2.1 C/S 架构

C/S 结构，即大家熟知的客户机和服务器结构。它是软件系统体系结构，通过它可以充分利用两端硬件环境的优势，将任务合理分配到 Client 端和 Server 端来实现，降低了系统的通讯开销。目前大多数应用软件系统都是 Client/Server 形式的两层结构，由于现在的软件应用系统正在向分布式的 Web 应用发展，Web 和 Client/Server 应用都可以进行同样的业务处理，应用不同的模块共享逻辑组件；因此，内部的和外部的用户都可以访问新的和现有的应用系统，通过现有应用系统中的逻辑可以扩展出新的应用系统。这也就是目前应用系统的发展方向。

3.2.2 系统体系结构

Xmirror 包括安全体检、应用防护、防火墙、资源监控、日志审计、安全设置等多项功能，其结构如图 3-2 所示。

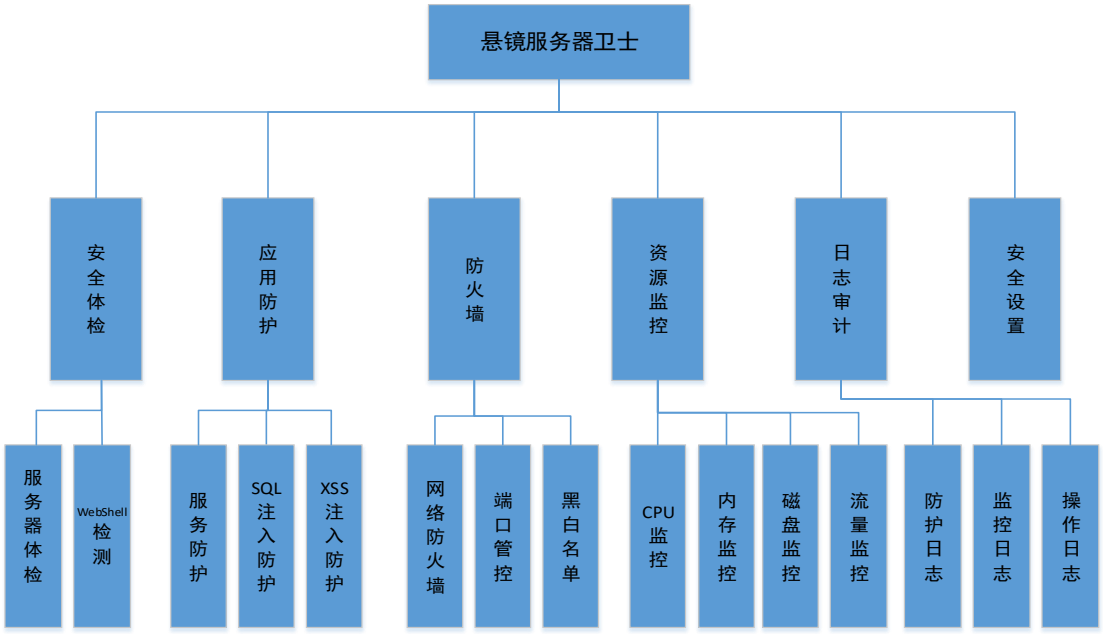

图 3-2 悬镜服务器卫士结构图

3.3 Xmirror For Linux 主要功能简介

3.3.1 安全功能

这一部分主要是对系统主要功能做一个简单的介绍。

- 对于强制的访问控制功能，它主要包括有文件强制访问控制、进程强制访问控制、服务强制访问控制。
- 登录防爆破功能：在特定时间窗口下允许最大登录错误次数，如果在这段时间内错误次数超过最大登录错误次数，软件将对该登录 IP 进行一段时间的屏蔽。登录防爆破功能是防止前端或者 Web 端通过暴力破解获得登录口令，进一步进入软件。
- 防火墙功能：网络防火墙、端口管控、黑白名单。网络防火墙细分为网络层攻击防护、应用层攻击防护、局域网连接防护。网络层攻击防护主要是针对 DDOS 攻击的防护；应用层攻击防护主要是 CC 攻击；局域网连接防护主要有被动 PING 命令保护、服务器自身上网阻断、被动 SSH 连接阻断。端口管控是用于控制服务器可疑端口的使用情况，可以对可疑端口进行禁用操作或手动添加禁用端口操作。针对高级用户提供添加自定义规则的超级黑名单和超级白名单功能。
- 远程监管功能：CPU 监控，内存监控，磁盘监控，流量监控。
- 应用防护功能：服务防护、SQL 注入防护、XSS 注入防护。服务防护功能是对服务器代理端的重要服务进行防护；对于 SQL 注入防护以及 XSS 注入防护中的规则项，其中 Get 防护项的含义是对 Get 类型请求中的所有的键值对的键和值以及去除域名的 URI 进行检测，检测到不合法的项进行拦截；Post 防护项的含义是对 Post 类型请求的所有键值对的键与值（除文件外）以及文件上传时的文件名进行检测，检测到不合法项进行拦截；Cookie 防护项是对所有 Cookie 键值对的键和值（除_utm、_pk_ref 外）的 URI 进行检测，检测到不合格项进行拦截；Header 防护项是对请求 Header 里的 X-Forwarder-For、User-Agent 和 Referer 字段的值进行检测，检测到不合法项进行拦截；Xml 防护项是对 Xml 根以下标签的值进行检测，检测到不合法项进行拦截。
- 安全检测功能：Linux 服务器体检，WebShell 检测。

表 3-1 功能列表

功能项	一级子项目	二级子项目
安全体检	安全基线配置	检测+修复
	安全策略	检测+修复
	网络优化	检测+修复
	WebShell 检测	自动检测+自定义检测
应用防护	服务防护	Apache/Samba/Mysql/Sshd/Ftpd/Snmp 服务等
	SQL 注入防护	详细的规则项：Get、Post、Cookie、Headers、Xml，包括默认配置+自定义配置。
	XSS 注入防护	详细的规则项：Get、Post、Cookie、Headers、Xml，包括默认配置+自定义配置。
防火墙	网络防火墙	CC 攻击防护
		DDOS 攻击防护
		被动 PING 阻断
		服务器自身上网阻断
		被动 SSH 连接阻断
	端口管控	自定义配置
	超级黑白名单	自定义配置
资源监控	CPU 监控	阈值管理
	内存监控	阈值管理
	磁盘监控	阈值管理
	流量监控	阈值管理
日志审计	防护日志	图形日志：近期历史攻击记录分布
		图形日志：近期历史操作记录

		图形日志：今日功能操作记录
		详细日志：DDOS 防护日志
		详细日志：被动 PING 阻断日志
		详细日志：自身上网阻断日志
		详细日志：被动 SSH 连接阻断
		详细日志：黑白名单日志
		详细日志：应用层防护日志
		详细日志近期日志
	监控日志	图形日志：近期资源监控告警分布
		图形日志：近期资源监控警告
		图形日志：今日资源监控警告
		详细日志：CPU 监控日志
		详细日志：内存监控日志
		详细日志：硬盘监控日志
		详细日志：流量监控日志
	操作日志	图形日志：近期历史操作记录分布
		图形日志：近日拦截攻击记录
		图形日志：近期历史攻击记录
		详细日志：连接情况
		详细日志：体检日志
		详细日志：网络防火墙日志
		详细日志：资源监控日志
		详细日志：应用防护日志
		详细日志：近期日志

3.3.2 参考标准

根据模块设计的需要以及有关法规的要求，依据的相关法律规范和标准包括但不限于：

1） 《信息系统安全等级保护基本要求》（GB/T 22239-2008）：应用类建设标准，第三级

2） 《信息系统安全保护等级定级指南》（GB/T 22240-2008）：应用类定级标准，第三级

3） 《信息系统通用安全技术要求》（GB/T 20271-2006）：应用类建设标准

4） 《计算机信息系统安全保护等级划分准则》（GB 17859-1999）：基础类标准，第三级

5） 《信息系统安全等级保护实施指南》（GB/T 25058-2010）：基础类标准，第三级

6） 《信息系统安全管理要求》（GB/T 20269-2006）：应用类管理标准

3.3.3 远程监控

通过 C/S 架构实现管理端和服务器代理端的远程监控，并且一个服务器代理端可以同时被多个管理端管理。

3.4 Xmirror For Linux 模块组成和各模块功能

3.4.1 安全体检

3.4.1.1 安全基线背景

NISF（National Institute of Standards and Technology）美国国家标准与技术研究所。《联邦信息安全管理法案》（FISMA）制定于 2002 年，是一套促进和发展美国的主要安全标准与准则的测评。FISMA 指定了 NIST 具有制定信息安全标准和指导方针的作用。

安全内容自动化协议（SCAP：Security Content Automation Protocol）产生的相关背景，对于企业级日常信息安全维护工作，执行系统基线安全配置、对系统安全配置进行实时监控、检查补丁安装情况、定期进行漏洞扫描，这些工作有点复杂，

需要进行维护的系统数量巨大且不同，对于新的威胁需要快速的反应，此外很多高层的方法需要落实到低层的技术细节上，于是，SCAP(Security Content Autmation Protocol)安全内容自动化协议，它列举了与安全有关的软件配置问题，它包括了XCCDF。

2005 年 2 月，美国国家安全局 NSA 和美国国家标准与技术研究所 NISF 联合发布了 XCCDF(eXtensible Configuration Description Format，可扩展的配置检查清单描述格式)规范，用于为安全检查清单、安全基准和其它安全配置指南建立一个通用平台，该规范有利于资源的重用，具有良好的可扩展性，以此促进安全措施的广泛应用。用于描述安全配置列表、基准点相关文档。一般用于描述目标系统安全配置规则。这个规范引起了行业和政府安全专家、分析人员、审核人员和安全管理产品开发者等强烈的关注，并得到众多厂商的支持。

FDCC(Feferal Desktop Core Configuration)，联邦桌面核心配置计划。是一项美国行政管理和预算局的命令。要求所有的政府机构为其全部安装有 Windows XP 和 Vista 计算机按照标准进行大约 300 条系统加固配置。目的是加强政府信息系统的抗攻击能力，并有效降低维护成本。它的目标是到 2008 年 2 月 4 日止在美联邦政府 45 万多台计算机上部署整合了安全配置的标准桌面操作系统，以减少数百万联邦计算机中的安全漏洞和非法配置，同时降低采购和运行成本。

其针对的不仅仅是桌面计算机，同时也包含笔记本计算机，FDCC 实际上是 SCAP 协议的一个典型应用。

从法规（FISMA）到机构（NIST）到协议（SCAP)到(规则)XCCDF 到应用（FDCC）最后到安全基线，安全基线就是这样产生的。

3.4.1.2 安全基线应用

木桶效应，一个木桶无论多高，它盛水的高度取决于其中最低的那块木板。一个信息系统的安全防护水平取决于防护能力最差的个体水平，由此引入了安全基线（Security Baseline）。安全基线的元素包括：操作系统组件的配置，例如：Internet 信息服务（IIS）自带的所有样本文件必须从计算机上删除；权限和权利分配，例如：计算机上的 administrator 密码每 30 天换一次。

国内的安全基线形势：中国移动公司下发的“中国移动公司基线安全配置指南”，要求总部及所有分公司内部业务终端进行合规配置。绿盟的 BVS 实际上早期是为移动公司定制开发的项目。其中也参考了 FDCC 的思想。中国国家信息中心提出了“中国政务终端安全桌面核心配置标准研究”（CGDCC），不过基本上也只是 FDCC 的翻译版。

本系统的安全基线模块根据国家等保三级的标准，对 Linux 操作系统的主机应当遵循的操作系统安全性设置标准，标准类似于安全基线的概念。通过对用户口令、远程登录限制、syslog.conf 的配置审核、ICMP 配置文件、SSH 配置文件等二十多项的配置文件和系统状态的扫描对整个 Linux 系统给予全方面的加固，从而保证系统的基础安全性。由这些一系列的安全策略、标准组成了安全体检的依据。

3.4.1.3 WebShell 概述

对于 WebShell 的理解，“Web”显然需要服务器开放 Web 服务，“Shell”取得对服务器某种程度上操作权限。WebShell 常常被称为匿名用户(入侵者)通过 Web 服务端口对 Web 服务器有某种程度上操作的权限，由于其大多是以网页脚本的形式出现，也有人称之为网站后门工具。

WebShell 是一种常见的网页后门，它常常被攻击者用来获取 Web 服务器的操作权限。攻击者在进行网站入侵时，通常会将 WebShell 文件与 Web 目录下的长长网页文件放置在一起，然后通过浏览器访问 WebShell 文件，从而获取命令执行环境，最终达到控制网站服务器的目的。当网站服务器被控制后，就可以在其上任意查看数据库、上传下载文件以及执行任意程序命令等。WebShell 与正常网页具有相同的运行环境和服务端口，它与远程主机通过 WWW(80)端口进行数据交换的，一次能够很容易地避开杀毒软件的检测和穿透防火墙。

总体，WebShell 的作用，一方面，WebShell 常常被站长用于网站管理、服务器管理等等，根据 FSO 权限的不同，作用有在线编辑网页脚本、上传下载文件、查看数据库、执行任意程序命令等。 另一方面，被入侵者利用，从而达到控制网站服务器的目的。这些网页脚本常称为 Web 脚本木马，目前比较流行的 ASP 或 PHP 木马，也有基于.NET 的脚本木马。 Xmirror 系统应用防护中 WebShell 检测，

所要应对的就是后者。

3.4.1.4 WebShell 攻击原理

WebShell 有以下 5 种攻击方式：

1）首先通过系统前台上传 WebShell 脚本到网站服务器，此时，网站服务器会向客户端返回上传文件的整体 URL 信息；然后通过 URL 对这个脚本进行访问，使其可以执行；最终导致攻击者能够在网站的任意目录中上传 WebShell，从而获取管理员权限。

2）攻击者利用管理员密码登录进入后台系统，并借助后台管理工具向配置文件写入 WebShell，允许任意脚本文件上传。

3）通过数据库的备份和恢复功能和获取 WebShell。在数据库备份时，可以将备份文件的扩展名更改为.asp 类型。

4）系统中其他站点遭受攻击，或者搭载在 Web 服务器上的 Ftp 服务器遭受攻击后，被注入了 WebShell，这些都会导致整个网站系统被感染。

5） 攻击者利用 Web 服务器漏洞直接对其进行攻击，从而获得控制权限。

WebShell 的感染过程描述：

1）攻击者首先通过 SQL 注入、跨站脚本攻击等方式得到上传权限，然后将 WebShell 上传至服务器。

2）通过 WebShell 完成对服务器的控制，实现植入僵尸木马、篡改网页以及获取敏感信息等恶意功能。

3）植入攻击木马，使其作为攻击“肉鸡”对整个网站进行感染。

3.4.1.5 Linux 下 WebShell 攻击

WebShell 反弹命令行 Shell 的方式在 Linux 从操作系统下 Web 服务器入侵提权过程中得到了广泛应用。在 Linux 下，WebShell 可以执行命令，然后溢出却必须在交互环境中进行，否则即使提权成功，也不能获得完美利用。因此，为了完成

WebShell 攻击，只需反弹一个 Shell 命令行窗口，在命令行终端下执行溢出并进行提权。

多数 Linux 操作系统下的 PHP WebShell 都通过反弹连接功能获得一个继承当前 WebShell 权限的 Shell 命令行窗口。在使用反弹连接功能前，需要首先使用 NC 工具对一个未使用的端口进行监听，然后选择反弹连接方式。

3.4.1.6 WebShell 检测

对于 WebShell 的检测，北京安普诺网络安全团队通过自主研发，设计出一种综合多种检测方法的综合检测方案。

特征检测系统中核心是特征提取，选取特征的好坏直接关系到检测结果的优劣。因此，在进行特征选取时，首先应对 Web 页面本身进行充分考虑，使得选取的特征能够很好地表现出静态页面。其次，选取的特征还应该具有动态特点，可以体现出页面所进行的操作。

如果提取网页的全部特征进行处理，则无法检测变形的 WebShell，也会因为特征过多而对效率产生影响。如果检测特征过少，则有可能产生误报。通过将多个 WebShell 库综合在一起，同时，又加入公司积累的特征码，综合构建了一个非常强大的 WebShell 特征库，这个特征库构成了 WebShell 检测的依据。

通过对文件进行特征库匹配、文件 base64 编码后特征库匹配、可疑特征码匹配等多种手段进行扫描，从而保证扫描准确性并且降低误报率。扫描后，可以具体的提供 WebShell 的名称、类型等详细信息，以供用户参考。并且针对反馈，用户还可以有选择的执行“清理”、“添加信任”等功能，以此实现对 WebShell 的检测、发现、处理等操作。

特征检测是比较常见的 WebShell 检测方法，base64 编码后特征匹配相对于普通特征码检测的优势在于匹配的准确性更高误报率更低。但是特征检测的局限性还是存在的，就是同时降低误报率和漏报率是很难实现的，所以还需要别的手段对文件进行综合的检测。

为此我们提出了 Delttime（文件创建时间间隔），Fnum（文件数量阈值）的

概念。以特征属性、Delttime 属性、Fnum 属性作为 Webshell 动态检测算法的主要 3 个输入参数，根据不同参数对检测结果的影响大小来决定其在算法中的权重。实践证明该算法检测效率相对传统特征值检测大幅降低检测误报率，有一定的可行性。目前正在申请国家发明专利。

与此同时，为了方便用户使用，软件给用户提供“快速扫描”、“自定义扫描”功能。

下表中为悬镜服务器卫士测试时 WebShell 查杀率反馈，具体数据参看下表：

表 3-2 悬镜服务器卫士 webshell 检测功能测试反馈

样本	文件个数	Webshell 个数	Webshell 检测项	可疑项	检测率（可疑与 Webshell 项）	WebShell 误报率（可疑文件误报率）	检测时间
360 样本	96	96	82	6	85.4%(91.7%)	0.00%（0.00%）	0 分 8 秒
Webshell 集合-1	1240	884	646	16	76.5%(78.4%)	0.00%（0.00%）	1 分 30 秒
Webshell 集合-2	538	535	421	8	78.7%(80.2%)	0.00%（0.00%）	0 分 44 秒
Webshell 集合-3	206	206	149	6	72.3%（75.2%）	0.00%（0.00%）	0 分 21 秒
wordpress 4.3.1	1229	0	1	1	无	0.08%（0.08%）	0 分 45 秒
DedeCMS-V5.7-UTF8-SP1（CMS）	2474	0	0	6	无	0.00%（0.24%）	0 分 24 秒
EmpireCMS_7.2（CMS）	1729	0	0	0	无	0.00%（0.00%）	0 分 32 秒
phpcms_v9.5.10_UTF8（CMS）	1287	0	0	0	无	0.00%（0.00%）	0 分 14 秒
KesionCMS X1.5 Free	2915	0	14	5	无	0.48%（0.17%）	0 分 40 秒
KesionEshop X1.5 Free	3052	0	14	5	无	0.46%（0.16%）	0 分 55 秒
KesionEshop X1.5 Free	1237	0	0	1	无	0.00%（0.08%）	0 分 23 秒

安普诺渗透测试系统	536	0	0	0	无	0.00%（0.00%）	0 分 5 秒

注：该测试在 Centos6.5/RHEL6.5 32 位、单核 CPU 1G 内存中测试，表格中 WebShell 误报率是指非 WebShell 的文件被检测出为 WebShell 的概率，可疑文件误报率指的是非可疑文件被检测出为可疑文件的概率。

3.4.2 应用防护

3.4.2.1 服务防护

典型服务防护采用虚拟化技术，在现有 Linux 操作系统空间中根据不同的用户应用分别创建出多个虚拟空间，实现用户与用户之间的隔离，服务与服务之间的隔离，该虚拟空间被称作“安全域”，每个安全域均具备增强型 RBAC、TE、BLP 安全机制。这些被保护的系统典型服务具有各自对应的安全域，实现用户与系统之间的隔离。对于原有应用或服务来说都是完全透明的，这也意味着对于原有应用的业务不会有丝毫改变。

具体来说，设置服务防护后本系统可对受保护的相关服务的端口监听、端口外联、文件读写、用户目录访问等操作进行主动安全域隔离，全面提升系统服务的完整性和安全性。

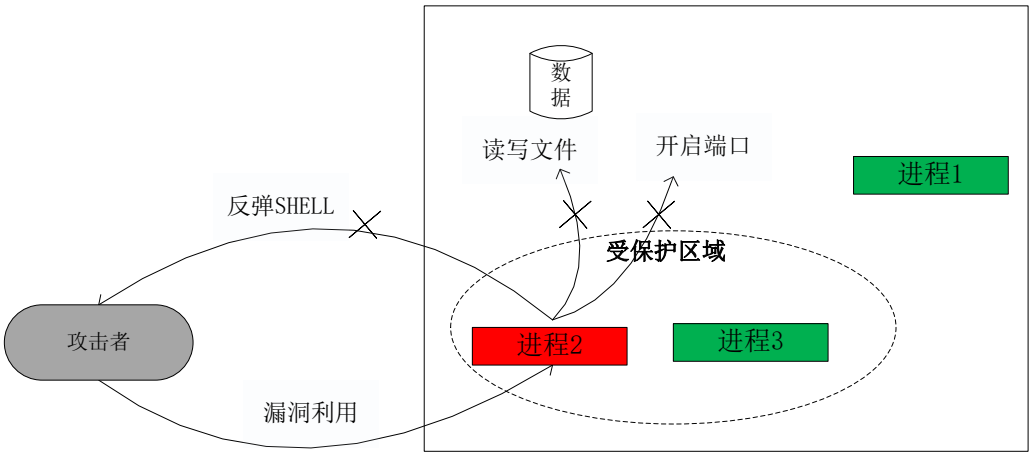

图 3-3　典型服务防护原理图

如上图所示，当前系统内运行着三个常用进程，分别是进程 1、进程 2 和进程 3。若进程 2 具备读写系统任意文件，开启任意端口和外联任意端口等特权（具备

针对系统进行危险操作的权限），当进程 2 存在的安全漏洞被利用时，则入侵者可将进程 2 作为跳板通过端口反弹进一步控制系统（例如借助 webshell 等），拿到系统控制权或者该进程 2 的特权后，非法读取或篡改系统上的重要文件或系统配置数据，在非授权情况下对系统监听端口进行变更等。典型服务防护模块正是基于此类场景的攻击的防护，可有效缓解此类安全威胁带来的风险。

3.4.2.2 SQL 注入防护

随着 B/S 模式应用开发的发展，使用者用该种模式编写应用程序的程序员也越来越多。但是由于这个行业的门槛不高，程序员的水平及经验参差不齐，相当大一部分程序员在编写代码的时候，没有对用户输入数据的合法性进行判断，使应用程序存在安全隐患。用户可以提交的一段数据库查询代码，根据程序返回的结果，获取某些他想得知的数据，这就是所谓的 SQL Injection，即 SQL 注入。对于 SQL 注入防护主要是匹配 POST、GET 请求中的关键字实现。

SQL 注入防护被细分为以下 14 条规则：⑴基本 SQL 注入检测；⑵增强 SQL 注入检测；⑶多关键字检测；⑷盲注检测；⑸认证绕过检测；⑹存储过程注入；⑺十六进制绕过；⑻数据库名检测；⑼SQL 同义语句检测；⑽敏感字符利用检测；⑾SQL 注释检测；⑿字符异常利用检测；⒀整数溢出攻击检测；⒁基于 MongDB 检测。这 20 项规则又分别从 POST、GET、COOKIE、HEADERS、XML 五个方面进行防护。

3.4.2.3 XSS 注入防护

XSS 注入防护被细分为以下 9 条规则：⑴基于 XSS 检测；⑵事件触发型 XSS 检测；⑶URI 利用型 XSS 检测；⑷常用关键字过滤检测；⑸强力关键字过滤检测；⑹JS 执行函数检测；⑺CSS 型 XSS；⑻IE 相关 XSS；⑼UPDF XSS。这 9 条规则又分别从 POST、GET、COOKIE、HEADERS、XML 五个方面进行防护。

3.4.3 防火墙

3.4.3.1 防火墙的简介

- 防火墙是指设置在不同网络或网络安全域之间的一系列部件的组合，它能增强机构内部网络的安全性。它通过访问控制机制，确定哪些内部服务允许外部访问，以及 允许哪些外部请求可以访问内部服务。它可以根据网络传输的类型决定 IP 包是否可以传进或传出内部网。
- 防火墙通过审查经过的每一个数据包，判断它是否有相匹配的过滤规则，根据规则的先后顺序进行一一比较，直到满足其中的一条规则为止，然后依据控制机制做出相应的动作。如果都不满足，则将数据包丢弃，从而保护网络的安全。
- 防火墙可以被认为是这样一对机制：一种机制是拦阻传输流通行，另一种机制是允许传输流通过。一些防火墙偏重拦阻传输流的通行，而另一些防火墙则偏重允许传输流通过。

3.4.3.2 防火墙的功能

- 可以保护易受攻击的服务；
- 控制内外网之间网络系统的访问；
- 集中管理内网的安全性，降低管理成本；
- 提高网络的保密性和私有性；
- 记录网络的使用状态，为安全规划和网络维护提供依据。

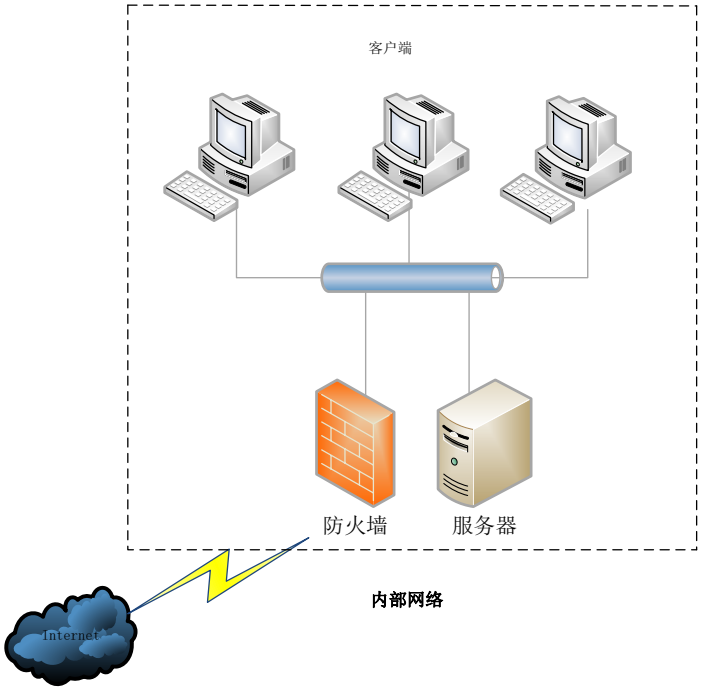

图 3-4 网络防火墙的位置

3.4.3.3 防火墙的类型

防火墙技术根据防范的方式和侧重点的不同而分为很多种类型，但总体来讲可分为：包过滤防火墙、代理服务器，两种类型：

1、包过滤防火墙（网络层），工作原理：

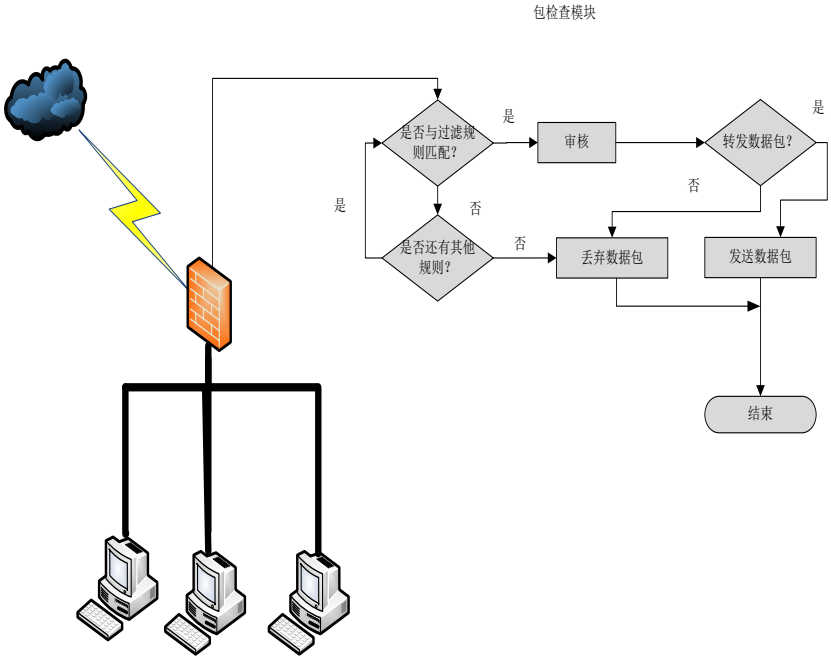

图 3-5 包过滤防火墙工作原理

通过检查数据流中的每个数据包的源地址、目的地址、源端口、目的端口、协议状态等因素来确定数据包是否允许通过。

2、代理服务器防火墙（应用层）工作原理，代理服务型防火墙是在应用层上实现防火墙功能的。它能提供部分与传输有关的状态，能完全提供与应用相关的状态和部分传输的信息，它还能处理和管理信息，例如：当代理服务器收到用户对某个网站的访问请求，先检查这个请求是否符合控制规则，如果符合 ACL，则替用户取回所需要的信息再转给用户 。

3.4.3.1 Iptables 优点

Netfilter/Iptables 的最大优点是它可以配置有状态的防火墙，这是 Ipfwadm 和 Ipchains 等以前的工具都无法提供的一种重要功能。有状态的防火墙能够指定并记住为发送或接收信息包所建立的连接的状态。防火墙可以从信息包的连接跟踪状态获得该信息。在决定新的信息包过滤时，防火墙所使用的这些状态信息可以增加其效率和速度。这里有四种有效状态，名称分别为“ESTABLISHED”、“INVALID”、“NEW”和“RELATED”。状态“ESTABLISHED”指出该信息包属于已建立的连接，该连接一直用于发送和接收信息包并且完全有效。“INVALID”状态指出该

信息包与任何已知的流或连接都不相关联，它可能包含错误的数据或头。状态“NEW”意味着该信息包已经或将启动新的连接，或者它与尚未用于发送和接收信息包的连接相关联。最后，“RELATED”表示该信息包正在启动新连接，以及它与已建立的连接相关联。Netfilter/Iptables 的另一个重要优点是，它使用户可以完全控制防火墙配置和信息包过滤。您可以定制自己的规则来满足您的特定需求，从而只允许您想要的网络流量进入系统。另外，Netfilter/Iptables 是免费的，所以这无疑是一个不错的解决方案。

3.4.3.2 系统防护架构

本系统是采用这样的设计思路，分为两层进行防护：一、网络层防护（Iptables 规则实现），包括 DDOS、PING、SSH、HTTP 这些项的防护；二、 应用层防护（Modsecurity 模块规则实现 Apache），包括 CC、DDOS、SQL injection、XSS injection 这些项的防护。

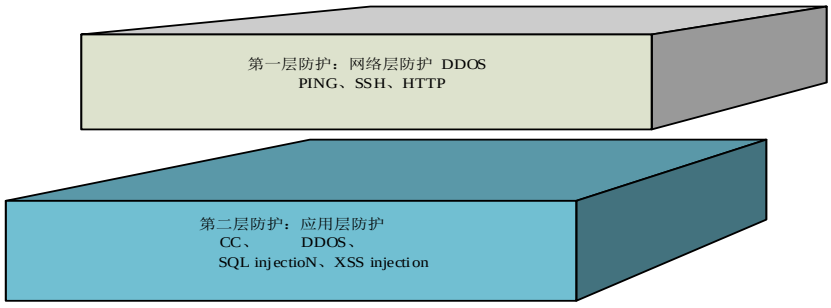

图 3-6 防火墙设计架构图

防火墙功能模块中包括了：网络防火墙、端口管控、黑白名单。

网络防火墙中包括：CC 攻击防护、DDOS 攻击防护、XSS 注入防护、SQL 注入防护、PING 命令防护、外网连接防护、SSH 防护。

3.4.3.3 网络防护层——DDOS 攻击防护

DDOS 攻击防护，DDOS(Distributed Denial of Service)分布式拒绝服务攻击指

借助于客户/服务器技术，将多个计算机联合起来作为攻击平台，对一个或多个目标发动 DDOS 攻击，从而成倍地提高拒绝服务攻击的威力。通常，攻击者使用一个偷窃帐号将 DDOS 主控程序安装在一个计算机上，在一个设定的时间主控程序将与大量代理程序通讯，代理程序已经被安装在网络上的许多计算机上。代理程序收到指令时就发动攻击。利用客户/服务器技术，主控程序能在几秒钟内激活成百上千次代理程序的运行。针对 DDOS 攻击的问题，也是通过添加 Linux 中自带的网络防火墙规则（Iptables 规则），来应对的。

3.4.3.4 网络防护层——PING 命令防护

PING 是用于检测网络连接性、可到达性和名称解析的疑难问题的主要 TCP/IP 命令。PING 最主要的用处就是检测目标主机是否可连通。黑客要入侵，就得先锁定目标，一般都是通过使用 PING 命令来检测主机，获取相关信息，然后再进行漏洞扫描。如何不受别人的攻击？那就是阻止别人 PING 自己的电脑，让攻击无从着手。PING 命令防护，通过杜绝“外界”利用 PING 命令，查看服务器，来防御一些潜在的威胁，从而达到防护的效果。

3.4.3.5 网络防护层——SSH 防护

SSH 协议简介，SSH(Secure Shell)传输层协议是底层的安全传输协议，是在 1995 年由 Tatu Ylonen 正式提出，其目的是要在不安全的网络上提供安全的远程登录和其他安全网络服务。SSH 使用多种加密方式和认证方式，解决了传统服务中存在的数据加密、身份认证问题。SSH 有成熟的公钥/私钥体系，为客户端和服务端之间的会话提供加密通道，解决了数据在网络上以明文传输的不安全问题。SSH 加密口令及数据可以在不安全的网络上透明的提供强认证和安全通信。SSH 是实现网络安全的非常轻型的工具，不占用客户机及服务器的大量资源，便于用户理解（至少在简单应用水平上是这样的），灵活且功能强大，集各种功能及易于使用等特点于一身。

- 端口管控功能采用的是白名单机制，除了添加到白名单中的端口外，其他端口均处于屏蔽、拦截状态，端口管控功能中显示的信息是端口、协议、

绑定 IP、占用进程、进程 ID、进程描述、状态，同时用户也可以手动添加端口到白名单中，手动的去刷新获取到的信息列表。

- 黑白名单主要是由用户来决定的，主要可以输入的项是规则目标，它分为进入控制和出行控制；协议类型，它有 TCP、UDP、ICMP、全部这些类型的选择；源 IP；目的 IP；源端口；目的端口；这些项组成了一条规则项，可以添加白名单也可以添加黑名单规则。

3.4.3.6 应用层攻击防护——CC 攻击防护

CC 攻击防护，CC 攻击是流量攻击的一种，CC 的含义就是模拟多个用户不停地进行访问那些需要大量数据操作，既是需要大量 CPU 时间的页面，造成服务器资源的浪费，CPU 长时间处于 100%，永远都有处理不完的连接直至网络拥塞，正常的访问被中止。CC 攻击基本上都是针对端口的攻击，属于硬性流量的攻击。

如果服务器（网站）被入侵了，一般都是服务器或者网站存在漏洞，被黑客利用并提权入侵的，导致服务器中木马，网站被挂黑链，被篡改，被挂马。针对 CC 攻击的问题，系统提出了一套防 CC 攻击的方案，通过添加 Linux 中自带的网络防火墙规则（Iptables 规则），来应对 CC 攻击。

3.4.4 远程资源监控

远程对主机的 CPU 使用率、内存使用率、磁盘使用率、网络流量进行定时监控。用户可以对监控参数进行设置，通过设置 CPU 使用率、内存使用率、磁盘使用率、流量的上限和定时监控间隔时间。如果在开始定时监控时，服务器的资源使用上限超过了设置的参数，就会报警并且保存到日志，从而方便用户管理。

3.4.5 日志审计

日志审计中包括了防护日志本身、监控日志以及操作日志。

防护日志是对服务器所受到的攻击以及拦截进行记录，包括的有图形日志和详细日志。图形日志中包括走势图与分布图，分布图中包括近期历史攻击记录分布环状图与拦截次数柱状图；走势图中包括近期历史攻击记录与近日拦截攻击记录。详

细日志主要记录的项是 DDOS 防护日志、被动 PING 阻断日志、自身上网阻断日志、被动 SSH 连接阻断、端口管控日志、应用层防护日志，每一条日志包含的项是时间、功能、主机名、ID、协议、源地址、目的地址和 MAC 地址。

监控日志中主要记录的是 CPU 监控、内存监控、硬盘监控、流量监控，每一条日志包含的项是时间、监控类型、警告信息。除此之外，监控日志中还包括走势图与分布图，走势图中包括近期资源监控告警与今日资源监控告警；分布图中包括近期资源监控告警环状图以及近期告警次数柱状图。

操作日志中是对用户进行的操作行为进行记录，主要记录的项是连接情况、体检日志、网络防火墙日志、资源监控日志、应用防护日志。每一条日志由时间、IP、功能模块、描述、执行结果这些项组成。以上组成了详细日志，除此之外，操作日志中也包括图形日志，图形日志分为走势图与分布图，走势图中主要包括近期历史操作记录与今日功能操作记录；分布图中包括近期历史操作记录分布环状图与操作次数柱状图。

3.4.6 版本更新

悬镜服务器卫士版本更新功能实现代理端与管理端的在线更新功能。代理端与管理端初始连接通信时，版本更新功能会自动检测当前安装版本是否为最新版本，如果为最新版本会提示用户当前版本为最新版本，如果版本更新检测后，当前安装版本不是最新版本，则会提示用户存在最新版本可以安装。

在更新版本进程启动时，代理端与管理端连接断开。如果代理端与管理端在更新服务器上同时存在新的版本，且新版本之前相互兼容，两者都会更新，如果只有一端存在更新版本，且当前安装的版本与最新版本兼容同样会提示用户进行更新。

版本更新功能除了具有版本更新之外，还具有备份、回滚、恢复的功能，在更新过程中如果出现异常现象，版本更新会启动回滚恢复到备份的状态，以保证用户的正常使用。

除了自动检测更新外，更新功能还具有手动检测更新的功能。用户在管理端与代理端连接的状态下，点击检测更新按钮，能够启动更新检测功能，之后，版本更新功能会检测比对版本，在存在新的可兼容的版本的情况下，提示用户进行选择更

新。

3.5 软件安全防护

悬镜服务器卫士是一款服务器加固产品，除了对服务器系统进行关键点的安全加固之外，对自身的安全防护也做了充分加固。

从登录控制进行基于 OpenSSL 的安全隧道通信外，到软件自身防护中动态验证码机制再到访问控制，再到之后的进程保护/文件保护，最后又针对自己的系统进行渗透测试，这些安全防护形成了一个闭环，层层防护，以确保在对关键服务器安全加固的基础上自身系统的安全性。

3.5.1 登录防护

悬镜服务器卫士采用的是 C/S 结构，在管理端可以通过服务器 IP、 系统用户名、密码与代理端进行认证，实现连接通信，这种方式的优点是用户不需要注册，省去了用户注册的环节，同时也更便于用户的操作与管理。

采用这种方式也会存在它的劣势，如果管理端与代理端之间的通信采用明文通信，在信息传输过程中，会导致系统账户信息泄露，轻则暴露系统账户名、登陆口令等信息，重则用户数据被盗用，非授权用户长期入侵系统并盗取用户个人数据。为了防止信息传递中的安全隐患，悬镜服务器卫士运用 OpenSSL，为管理端与代理端之间的通信进行加密。

3.5.2 动态验证码

软件本身作为系统安全的一部分，系统对软件自身安全的也有一定的防范措施，为防止利用特定程序对某一个特定用户用暴力破解方式进行不断的登录尝试，来突破软件登录认证的恶意攻击，系统还引入了动态验证码机制。如果密码输入错误次数超过三次就会出现验证码，如果再输入错误两次，账户将会被锁定。

验证码是现在很多网站通行的方式（比如招商银行的网上个人银行，腾讯的 QQ 社区等），虽然登录麻烦一点，但是对一些软件来说这个功能还是很有必要，对于系统的安全也有一定的实质作用。

验证码，是将一串随机产生的数字或符号，生成一幅图片，图片里加上一些噪点（防止 OCR），由用户肉眼识别其中的验证码信息，输入表单提交网站验证，验证成功后才能使用某项功能。

系统在登录认证中引入验证码最主要的作用是防止对认证机制的暴力破解，验证码机制无疑是一个很好的选择。

3.5.3 安全设置

安全设置又被称为安全名片，设置安全名片访问服务器，限制客户端的访问。

需要设置的项主要包括：主机名、用户名、主机 IP、磁盘序列号。

安全名片也是访问控制的范畴，以与“黑名单”概念对应的白名单为规则，有很多软件都应用到了黑白名单规则，例如：操作系统、防火墙、杀毒软件、邮件系统、应用软件等，凡是涉及到控制方面几乎都应用了黑白名单规则。黑名单启用后，被列入到黑名单的用户不能通过。如果设立了白名单，则在白名单中的用户会优先通过，安全性和快捷性都大大提高。在白名单之列的系统具有与远程服务器通信的权利，不在白名单之列的则不允许远程登录服务器。提取系统的指纹、IP、用户名、硬盘序列号等来作为判断的依据。

3.6 Web 服务器防护

Web 网站防护是悬镜服务器卫士的最为核心的设计，从应用防护中的 WebShell 检测，到防火墙中的 DDOS 攻击防护、CC 攻击防护、网站漏洞防护等功能，都起到对 Web 网站的防护。

悬镜服务器卫士主要支持的 Web 服务器是 nginx、apache，具体支持版本如下表所示

表 3-3 软件支持的 Web 服务器版本

WEB 服务器	服务器版本
Apache	2.2.x 2.4.x
Nginx	0.x.x 1.x.x

对于上表中的 Web 服务器版本，在 Linux 系统上无论是以源码编译方式还是

yum 安装的方式，只要是利用这些 Web 服务器配置的网站，悬镜服务器卫士均可以检测到并对其起到防护作用。

同时，对利用 LAMPP（Apache+MySQL+PHP+PERL）集成安装包安装配置的网站也具有防护的作用。

4. 产品特点

4.1 更全面的 Linux 系统安全检测和保护

相比其他的 Linux 安全软件，Xmirror 从内核级的强制访问控制，到操作系统本身的配置文件和系统状态的检测，再到应用安全的 WebShell 检测和应对外部攻击的防火墙，Xmirror 对 Linux 服务器进行了全方位立体的保护，使服务器更安全。

4.2 更人性化的用户体验

Xmirror 给用户提供了更加美观的界面和更加简便的操作。Xmirror 尽可能的帮助用户判断和处理安全问题，从而减少用户的操作量和减少使用门栏。

4.3 部署实施操作简单

一般 Linux 系统的软件非常不好安装和部署，Xmirror 只需进行简单的配置即可运行；若在系统组建之间有防火墙或其他访问控制设备，建议与网管人员配合在防火墙上配置相应规则，实现安全通信，可以自由设定相应的端口，避免在 Web 服务器上开启其他端口。

5. 运行环境

OS	Version
代理端：Linux	Centos（redhat）5.x　x86/x64、 Centos（redhat）6.x　x86/x64、 Centos（redhat）7.x　x64 Ubuntu server 12.04、14.04、15.10（单独发行版本）

管理端：Windows	Windows 7 x86/x64 、Windows 8 x86/x64、 Windows 8.1 x86/x64、Windows 10 x86/x64

6. 产品购买

6.1 商务流程

- 客户提出试用申请，通过审核后，双方签订保密协议，发出试用版本，并协助联调测试；
- 客户试用满意，双方达成合作意向，签订正式服务合同；
- 提供正式版本和客户 ID，并协助客户完成产品的集成和上线。

6.2 联系方式

张涛 [技术经理]
联系电话：18510237215
Mail: zhangtao@anpro-tech.com QQ：510691049
公司地址：北京市海淀区安宁庄西路 Imoma 写字楼 1-303 室 邮编：100085