fai-project系统安装定制简介

服务器通用操作系统基础设施常规配置、性能优化、安全加固

本方案系统支持优先级：debian>ubuntu>centos,这意味着低优先级会有更多的错误与未实现功能，实际生产环境服务器会根据情况调整配置，本文档描述与服务器不一致的地方以服务器配置文件为准

前期准备：

1. NTP服务器2台，提供时间服务
2. DNS服务器2台，外网dns缓存与内部名字解析
3. ZABBIX监控1台，基础监控
4. ANSIBLE主控1台，自动化运维，提供ssh公钥给被控服务器
5. 备份机存储1台，保存所有服务器镜像备份，提供ssh公钥给被控服务器
6. Rsync客户端IP列表，web后端更新代码
7. Repo代理缓存1台，提供repo（同时支持debian、ubuntu和centos）内网缓存
8. 安装过程保持网络畅通，安装时软件包更新需要访问网络
9. Nfs服务器提供静态web文件存储
10. Syslog远程日志服务器1台，集中保存所有服务器日志

常规系统增强与配置变更

1. 无论服务器有多少个网卡，第一个网卡必须为内网网段（管理网络、内部数据网络、存储网络），并且支持PXE启动，第一个网卡名字eth0
2. IP地址与MAC绑定并且DHCP分配的地址与静态设置一致,安装完成自动生成静态IP配置文件
3. 初始安装是安全的不必担心来自网络的入侵。
4. 所有服务只监听内网地址的端口,禁止监听0.0.0.0，仅有对外提供的服务可以监听外网IP

内部服务： sshd ntpd dns-cacher zabbix mysql-server redis memcache

1. 默认用户hmuser使用32位字串复杂密码，可以sudo至root
2. Root用户内置ansible控制端公钥可以直接root登录，Sshd服务端 禁止root密码登录 使用公钥或者普通用户登录后su或者sudo 禁止dns，Ssh客户端开启持久连接，加密算法使用椭圆曲线ecdsa
3. dns解析，时间同步,日志记录使用内网服务器
4. grub 关闭splash 网卡使用ethX名称， 详细输出，控制台分辨率1024x768 io调度器deadline，关闭光标闪烁，
5. apt-get 修改源（repo）自动使用局域网代理缓存 https不用代理
6. 包含sysctl内核参数调优，sysfs调优，ext4文件系统调优（性能调优以真实业务客观化观察计时测量数据为准，而非跑分与基准测试）。
7. 彩色控制台输出man ls grep
8. 默认语言en\_US.UTF-8
9. iptables-persistent保存防火墙规则
10. needrestart更新安装软件检查是否重启
11. smartmontools监视硬盘smart信息
12. nscd本地名字服务缓存（user，group,hosts,service）
13. apt-show-versions检查软件包版本
14. apt-fast多线程快速下载软件包
15. cron-apt自动检查下载更新，默认不安装任何更新
16. **~~unattended-upgrades~~**~~自动检查下载安装安全(secure)更新~~
17. ~~ipwatchd IP冲突检测~~
18. 用户默认交互shell改为zsh，bash仅运行脚本
19. kexec-tools内核热重启（不经过bios自检）
20. 性能观察sysstat，iftop（查看网卡），itop（查看中断），dnstop（查看dns解析），powertop（查看电源）~~，apachetop（查看apache）,atop（全面查看进程）, hatop(查看haproxy),jnettop（查看占用流量最多的主机/端口）, kerneltop（查看内核函数使用，需要内核开启profiling）, mytop（查看mysql）, nethogs（进程的网络使用）,~~htop（查看进程）~~, pgtop（查看postgresql）, sntop（检测主机存活）, virt-top（查看虚拟机）,Blktrace（追踪块设备io）, dnstracer（追踪dns解析）, fatrace(追踪打开文件), fxt-tools（追踪多线程）, ioapps（重放IO追踪与分析gui）, kmtrace（追踪内存泄漏gui）, latrace（追踪动态链接库）, leaktracer（追踪内存泄漏）, lft（四层路由追踪）, mutextrace（追踪互斥体）,~~mtr（路由追踪）~~, netsniff-ng（网路包嗅探）, pstack（追踪进程栈）, tcptraceroute（追踪路由）, kernelshark（GUI）, trace-cmd（追踪命令执行）, tcptrack（追踪tcp连接状态），bmon带宽监控工具, Collectl：一体化性能监控工具, Nmon：监视Linux性能,iptraf网络流量查看,glances监视系统性能，dstat系统性能~~
21. ~~性能动态追踪bcc-tools，sysdig（无法在grsecurity内核运行），Lttng-tools,lttv,lttngtop,（Linux追踪工具）~~
22. lm-sensors 监视传感器数据
23. ~~cpufrequtils CPU主频控制~~
24. irqbalance SMP处理中断

安全保护系统的设计原则

(1). 最小特权：为使无意或恶意的攻击造成的损失最低，每个用户和程序必须按需使用最小特权；  
(2). 机制的经济性：保护系统的设计应小型化、简单、明确，保护系统应该是经过完备测试或严格验证的；  
(3). 开放系统设计：保护机制应当公开，理想的情况是将安全机制加入系统后，即便是系统的开发者也不能侵入这个系统；  
(4). 完整的存取控制机制：对每个存取访问系统必须进行检查；  
(5). 基于“允许”的设计原则：说白了就是“白名单”策略，基于否定的访问控制策略；  
(6). 权限分离：实体的存取应该受到多个安全条件的约束；  
(7). 避免信息流的潜在通道；

(8). 方便使用友好的用户接口；

系统安全加固

保持更新至最新补丁只能避免已知漏洞的威胁，无法对抗0day漏洞，基于特征库的扫描性能开销巨大，同样无法对抗未知攻击，此类工具（方法）仅限于应付已知威胁。

对抗0day漏洞的前提基础是有能力解决已知漏洞威胁，公开但是未修复漏洞视作0day处理，即使具备对抗0day漏洞能力，系统依旧需要做常规安全更新与软件升级（30天内修复已知漏洞）。

安全防御加固，按照最小权限，主动免疫，多重防御，最低成本，最高收益法则实行，目标具备前瞻性，防患于未然，先发制人，能够有效对抗未知木马、未知rootkit、未知后门、0day漏洞、程序bug、以及人为操作失误的威胁（需要做概念验证，证明其有效，无法验证、验证失败或者拒绝验证认为无效措施），无需改变现有应用，最大限度保证程序按照预期的方式运行。

~~其他措施：监控、审查、取证、保密、隐匿、伪装、反击，震慑不良企图~~

暴露在外网的系统要具备相当的威慑力，迫使攻击者不敢贸然行事，针对来自中国的攻击者，系统威慑力排名如下：BSD>Linux>Windows，这样安全防护人员可以集中精力专注于高级别高技术水准的严重/致命威胁，而不是疲于奔命，应付骚扰与无实质伤害的威胁，避免针对人员的DOS攻击。

系统应当具备健壮性，有相当的抗打击能力，部分节点，组件失效或者被摧毁仍然可以提供所需功能。

安全设计应当尽可能小的影响性能，额外的开销尽可能的少，小到无法观察测量，最短的响应时间，根据出现频度不同，监视过滤拦截异常应该在1毫秒之内完成，甚至更低的时间。

整个生命周期内尽可能低的复杂度，尽量少的人力开销，便于实施、维护、升级。

堡垒最容易从内部攻破，相对于外部威胁，内部威胁更需严加防范，把服务器上的所有用户都看成是怀有恶意的使用者，所以系统必须限制使用者的权限，并且监视记录使用者行为，这是一个无关道德的问题，而是一个技术能力的问题。

强制访问控制（RBAC角色访问控制）

~~限制系统里面的用户(包括root),必须按照管理规则预期的方式工作, 规则之外,完全禁止.~~

~~系统管理特权实行三权分立~~

~~系统管理员（root/adminstrator）系统管理员负责系 统的安装、 管理和 日常维护． 如安装软件 、 增添用户账号、数据备份等 ，类似于公司的总经理~~

~~安全管理员（secadmin）安全管理员负责安全属性的设定与管理．类似于公司的监事会；~~

~~审计管理员（auditor）审计管理员负责配置系统的审计行为和管理系统的审计信息． 类似于公司的董事会 。~~

~~三个角色互相制约。攻击者破获某个或某两个管理角色 的口令时不会得到对系统的完全控制~~

伪装不能实质提升安全水准，但是可以有效阻止扫描器、ROBOT（机器人）、AI（人工智能）自动化攻击，迷惑与麻痹攻击者，加大攻击难度系数，使其犯更多错误，付出更多时间成本与代价。

~~主动防御：主动侦查可能攻击己方的对手，进行防范和报复。~~

~~自毁装置与触发条件设计~~

~~数据机密性保护~~

系统完整性保护，不被篡改，以及被篡改及时发现。

审计与取证，秘密审计不被用户察觉，无法逃脱，无法被清理/毁灭证据

~~限制Intel ME（~~*~~Intel Management Engine）~~*~~的同时加强信任链条。~~

注意：无法对抗底层攻击，固件层，hypervisor层，硬件设备层。

1. 提供2类内核，默认普通内核用于安全性要求不高的场景，grsecurity/pax加固内核用于需要高级别安全特性的场景，内核开启apparmor，禁用selinux（斯诺登“棱镜门”之后禁止一切NSA相关代码运行）
2. 复杂密码，sha512散列,密码强度，至少是大小写字母+数字+符号一共16位以上（libpam-cracklib）
3. 限制ssh用户只能内网IP登录
4. ~~拒绝root登陆远程和控制台以及su~~
5. 限制secuetty root登录
6. ~~禁止加载内核模块，kernel.modules\_disabled=1~~
7. 默认用户掩码077
8. 有且仅有adm组用户切换到root权限
9. ~~取消root用户无限大权限，限制其实际权限等同于或者弱于普通用户，对抗root提权漏洞（grsecurity+RBAC）~~
10. ~~系统全局启用基于角色访问控制RBAC，并且实施最小权限（grsecurity+RBAC）。~~
11. 隐藏pid，普通用户禁止查看其他用户进程
12. ~~隐藏或伪装操作系统指纹，避免被nmap扫描（grsecurity未完全实现，hardenedbsd未完全实现），修改tcp协议栈，避免被探测开机时间。~~
13. ~~应用沙箱firejail，fakeroot-ng~~
14. 禁止usb存储设备，firewire存储设备，禁止pc喇叭，禁止蓝牙设备
15. 启用进程记账acct，~~启用网络记账net-acct(网卡开启混杂模式，记录所有IP连接，包括对外发起连接和各种扫描器连接失败)~~
16. 禁止每日备份密码文件passwd，group,shadow,gshadow
17. 禁止安装locate和updatedb避免泄密
18. ~~Arpwatch监视arp攻击~~
19. Debsums检查已安装软件包完整性
20. Debsecan 检查CVE漏洞报告
21. **只读挂载共享内存/dev/shm（会影响部分程序运行chrome,postgresql）**
22. ~~限制ssh用户命令restricted-ssh-commands~~
23. Haveged增强随机数,~~rngd.~~
24. 禁止控制台ctrl alt del重启
25. 开启内核audit审计，审计规则: 修改关键文件,修改标志位,修改能力,修改扩展属性
26. 系统安全审计工具lynis
27. Dmesg限制，普通用户无法查看dmesg
28. Pam\_tmpdir实现用户使用独立的tmp目录
29. 禁止普通用户查看日志（尽可能多的禁止日志，直至最终完全不可见）
30. 禁止普通用户（非adm组成员）查看last登录信息
31. 用户进程资源限制Pam limit（避免fork bomb威胁,以及各种资源耗尽导致服务器宕机）
32. ~~Fail2ban根据日志执行相关屏蔽动作~~
33. ~~防火墙软件nftables（替代iptables）~~
34. ~~Suricata入侵检测（替代snort）~~
35. ~~蜜罐软件honeyd，farpd，iisemulator，tinyhoneypot~~
36. 系统完整性检查工具aide，~~AFICK，Osiris，Samhain，Tripwire，OSSEC (syscheck)，mtree~~
37. Rootkit检查工具unhide，~~Rkhunter~~
38. ~~普通用户隐藏硬件信息（使用grsecurity的RBAC禁止查看硬件命令 禁止访问procfs和sysfs实现）~~
39. ~~缓冲区溢出检查工具pscan, flawfinder，splint，bfbtester~~
40. 记录unix用户创建、删除
41. 用户命令审计追踪snoopy,日志/var/log/auth
42. ~~IMA（Integrity Measurement Architecture完整性度量体系）~~
43. ~~EVM（Extended Verification Module扩展验证模块~~）
44. ~~TPM（可信平台模块，需要硬件支持，当前不支持TPM2.0）tpm-tools,~~ ~~trouse~~rs~~, tss2~~
45. ~~加密工具gnupg, ecryptfs-utils, cryptsetup,~~
46. ~~Neopi （Webshell代码检测）~~
47. ~~psad，（根据iptables日志探测端口扫描），sxid -（检查设置 suid, sgid 的文件和目录）~~
48. ~~DNS解析审计~~
49. ~~防火墙使用全球IP地址实时黑名单~~
50. ~~Nginx+modsecurity WAF防火墙~~
51. ~~隐藏系统内关键进程，秘密实施审计、监视、记录系统行为(由grsecurity+RBAC实现~~)
52. ~~apparmor基于路径的强制访问控制（3.0支持网络访问控制，支持grsecurity内核，但是不能与grsecurity的RBAC同时开启）~~

参考：

中国国家标准GB/T 18336-2008

ISO国际标准ISO/IEC 18045:2005

美国国防部：DISA STIG标准，git clone https://github.com/hardenedlinux/STIG-4-Debian.git

FLOSS最佳实践

<https://wiki.archlinux.org/index.php/Security>

<https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet#Weak_typing>

<https://github.com/linuxfoundation/cii-best-practices-badge/blob/master/doc/criteria.md>

<https://trent.utfs.org/wiki/Hardening/Linux#Per-user_.2Ftmp>

<http://netfilter.org/projects/nftables/>

<https://www.debian.org/doc/manuals/securing-debian-howto/index.en.html>

<http://www.honeynet.org/>

<http://wiki.debian.org/SELinux>

<http://www.openwall.com/linux/>

<https://linux-audit.com/nftables-beginners-guide-to-traffic-filtering/>

<http://www.malwaremustdie.org/>

<http://iase.disa.mil/stigs/Pages/index.aspx>

<http://www.astra-linux.com>俄罗斯军队的GNU/Linux发行版

<https://www.rsbac.org/>

<https://wiki.gentoo.org/wiki/Integrity_Measurement_Architecture>

<https://lwn.net/Articles/394170/>

<https://sourceforge.net/p/linux-ima/wiki/Home/>

<http://ecryptfs.org/documentation.html>

<https://www.badips.com>

<http://www.modsecurity.org/>

<https://www.owasp.org>

<http://flint.cs.yale.edu/certikos/> CertiKOS

<https://wiki.gentoo.org/wiki/Hardened_Kernel>

<https://github.com/thestinger/linux-hardened>

<https://github.com/bartblaze/Disable-Intel-AMT>

注释

被删除线划掉的功能特性，仅仅是默认的初始安装未实现，可以根据实际情况启用。