保护 H免受强力口令破解攻击 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/290/2021_2022__E4_BF_9D_E6_8A_A4_H_E5_85_c101_290613.htm 在确保Unix类系统的安全免受外部的攻击时，我们要保证不能运行任何不必要的服务。在免费的BSD Unix系统或Linux发行版本中，我们几乎总能遇到的一个服务，即SSH进程。 在确保Unix类系统的安全免受外部的攻击时，我们要保证不能运行任何不必要的服务。在免费的BSD Unix系统或Linux发行版本中，我们几乎总能遇到的一个服务，即SSH进程。在多数情况下，这指的是OpenSSH，你可能会需要确保远程访问服务器的安全，特别是在你的服务器并没有安装监视器和键盘时更是这样。 只要你连上互联网，你就会发现：总有一些系统在那儿每天24小时都在试图通过22号端口访问你的计算机，其目的是突破你的SSH口令保护。如果你检查一下网络中与互联网直接相连的任何系统中的（或者是连接到某个开放的无线访问点的设备中的）/var/log/auth.log文件，很快你就会发现，在这个文件的末尾会出现如下的内容： NOV 05 9:03:12 local_host sshd[6906]: error: PAM: authentication error for root from remote_host 其实，为降低这种攻击的影响范围，我们可以做的事情有许多。我们完全可以减少由这种盲目攻击所造成的损害。例如，就目前的技术水平而言，一个足够复杂的口令就很难破解。当然，任何复杂的口令最终都能够被破解，不过这种可能性也许要花费一个世纪才能完成。 另一个方法是简单配置SSH，使之在另外一个端口上监听，而不是22号端口。例如，可以使其监听1138号端号。为此，可以简单地改

变/etc/ssh/sshd_config文件的端口行（即以Port开头的一行），以显示你想使用的新端口号，而且在从一个远程系统连接时，要确保使用这个端口号。在编辑完毕后，sshd_config的这一行看起来应是如下这个样子： Port 1138 这种方法可以提供额外一层的安全性，使普通的恶意破坏份子难以发现你拥有一个开放的SSH端口。OpenSSH容许采用基于密钥的认证而不是口令认证，这就进一步地限制了这种别有用心的攻击，这正如我们可以配置一个防火墙，使其拒绝除被列入优良者名单的所有连接企图。 也许最普遍的、最有效的限制强力攻击的方法是简单地禁止通过SSH对根（root）的访问。也许你想远程登录，然后执行需要根的访问权的管理任务，不过你还要保障其他人不能通过强力的口令破解攻击直接获得对你系统的访问权。 当然，你还可以容许其它的账户通过SSH访问你的系统。借助于sudu或su，你可以获得对根的访问。因为自动化的攻击难于猜测到一个未知用户的账户名称及其口令（特别是如果这个口令足够复杂的话），而且，事实上多数攻击者绝对不会试着猜测除根用户之外的其它用户的口令，主要原因在于：由于采取这种简单的措施，你的系统在面对这种自动化的攻击时，安全性得到了极大的提升。如果你不允许一般用户账户使用sudu或su来获得对根的访问，情况尤其如此。你可以指定一个特别的管理员账户，使其对系统拥有很有限的特权,不过一定要有一个其它非根账户没有的特权：通过su或sudu执行管理任务的能力，这种任务可拥有对根的特权。 但是，一旦你已经不接受通过SSH对根的访问，你就会选择安排可管理的访问，其实，允许从远程对根的访问的情况很少。要禁用通过SSH直接登录到根用户，需要再

次编辑ssdh_config文件，将PermitRootLogin设置为“ no ”：PermitRootLogin no 有一些操作系统，如FreeBSD将其作为默认值。其它的系统，如多数的Linux发行版本并非如此。任何用户都应该检查其选择的操作系统，在决定自己的系统安然无恙之前，首先要决定其默认的PermitRootLogin设置。