网络安全检测的理论和实践（三）

卿斯汉 （中科院信息安全技术工程研究中心 100080）

4 网络安全检测

4.1 背景

由以上分析可知，针对网络黑客各种可能的攻击，我们需要有一个用于全面维护网络系统安全的综合集成化工具，它可以通过对目标系统进行全面的探测和模拟攻击，找出目标系统自身以及它所在网络的安全脆弱点，消除安全隐患，维护系统安全。

下面，我们设计一个模拟的网络安全检测系统，它比较全面地考虑了系统安全的各种因素，不仅对各种基于UNIX的系统在多用户环境下自身的安全进行检测分析，而且对系统所在网络可能遭受各种安全攻击的薄弱环节提出警告。另外，该系统通过可插型附件的体系结构，使其具有易于扩展的功能，使用户可将最新发布的网络或系统探测工具方便地集成到已有的软件中去，使系统的网络维护功能不断增强，跟上网络攻击和反攻击技术不断发展的步伐。

4.2 工作原理

该系统对网络安全的检测主要分为两个部分，即对本机的安全检测和在Internet网络上的安全检测。网络安全检测系统的实现基于黑客攻击的思路，采用模拟攻击的方法测试目标系统在Internet上的安全漏洞。此外，安全检测系统还能设置防火墙，对网络易受攻击的各个节点加以保护，对远程登录的可疑用户进行实时的跟踪和监控。

该系统对目标系统可能提供的各种网络服务进行全面扫描，如finger，ftp，tftp，rexd，NFS，NIS等，尽可能收集目标系统远程主机和网络的有用信息，并模拟攻击行为找出可能的安全漏洞，例如错误配置的网络服务、系统或网络应用的漏洞等。由于网络测试涉及远程机及其所处的子网，因而大部分检测、扫描、模拟攻击等工作并不能一步完成。网络安全检测系统采取的方式是，先根据这些返回信息进行分析判断，再决定后续采取的动作。或者进一步检测，或者结束检测并输出最后的分析结果，如是反复多次。总的说来，该系统的网络安全检测使用的是“检测—分析”循环结构。

该系统的设计注重灵活性，各个检测工具相对独立，为增加新的检测工具提供方便。如需加入新的检测工具，只需在网络安全检测的实现目录bin下加入该工具，并在网络检测级别类中注明，则启动程序会自动执行新的安全检测。为了实现这种灵活性，该系统的网络安全检测采用了将检测部分和分析部分分离的策略，即检测部分由一组相对而言功能较单纯的检测工具（如TCP端口扫描，UDP端口扫描，NIS扫描等等）组成，这种工具对目标主机的某个网络特性进行一次探测，并根据探测结果向调用者返回一个或多个标准格式的记录。分析部分则根据这些标准格式的返回记录决定下一轮对哪些相关主机执行哪些相关的检测程序，这种“检测—分析”循环可能进行多轮，直至分析过程不再产生新的检测为止。一次完整的检测工作可能由多轮上述的“检测—分析“循环组成。

该系统为了进一步提高灵活性采取了这样的策略：分析部分并不预先设定，而是由每个分析子过程在运行时刻根据与之相关的规则集自动生成。这些规则集用一种规范的形式定义了各子过程的实际行为，只要掌握了规则的书写方法，使规则的格式简单且语义清晰，任何用户都可随意添加自己的规则，改变分析子过程的行为，从而达到控制“检测—分析”流程的目的。

由于该系统的网络测试采用了“检测—分析”的循环结构，它每一次循环的检测结果都具有保存价值，尤其当检测比较详尽、范围较大（如检测某个子网）时，检测时间会较长，检测结果也很丰富，因此应当将结果存储起来供下一循环分析使用或供以后参考。每次进行网络检测之前，网络安全检测系统都创建一个新的数据库或选择一个先前创建的数据库，用以存放本次检测的结果。如不选择，系统会自动用缺省数据库存放之，如果选择的数据库是已经存在的，则本次检测数据将和数据库中已有的数据合并。

网络检测数据库由下述3个部分组成：

(1)主机列表all-hosts，用于存放所有检测过的主机；

(2) 事实记录列表facts，用于存放如前所述由检测部分和分析部分产生的标准格式的

返回记录；

(3) 检测项目列表todo，用于存放所有进行过的检测。

4.3 网络安全检测系统的逻辑结构

网络安全检测系统的核心逻辑结构分为以下5个主要组成部分：

(1) 策略分析部分。策略分析部分用于控制网络安全检测系统的功能，即它应当检测哪些主机并进行哪些检测。它根据系统预先设定的配置文件决定应检测哪些Internet域内的主机，并决定对测试目标机执行的测试级别（简单、中级和高级）。

(2) 获取检测工具部分。对于给定的目标系统，获取检测工具部分用于决定对其进行检测的工具。目标系统可以是一个主机，或是某个子网上的所有主机（子网扫描）。目标系统可以由用户指定，也可以由分析推断部分根据获取数据部分获得的结果产生。一旦确定了目标系统，获取检测工具部分就可以根据策略分析部分得出的测试级别类，确定需要应用的检测工具。这些检测工具正是获取数据部分的输入。

(3) 获取数据部分。对于给定的检测工具，获取数据部分运行对应的检测过程，收集数据信息并产生新的事实记录。安全检测系统能在检测循环中记录哪些检测是已执行过的，哪些检测是还未执行的，避免重复工作。最后获得的新的事实记录是事实分析部分的输入。

(4) 事实分析部分。对于给定的事实记录，事实分析部分能产生出新的目标系统、新的检测工具和新的事实记录。该部分又分为几个事实分析子过程，每个子过程分别由自己的基本规则集控制，同时该规则集又在子过程的分析中不断更新。新生成的目标系统作为获取检测工具部分的输入，新生成的检测工具又作为获取数据部分的输入，新的事实记录又再一次作为事实分析部分的输入。如此周而复始，直至不再产生新的事实记录时为止。

(5) 报告分析部分。当安全检测系统执行完网络安全检测之后，会获得关于目标系统的大量信息。报告分析部分则将有用的信息组织起来，用超文本界面显示，使用户可以通过HTML浏览器方便地查看运行的结果。

网络安全检测系统的逻辑结构及其流程图分别如图2和图3所示。

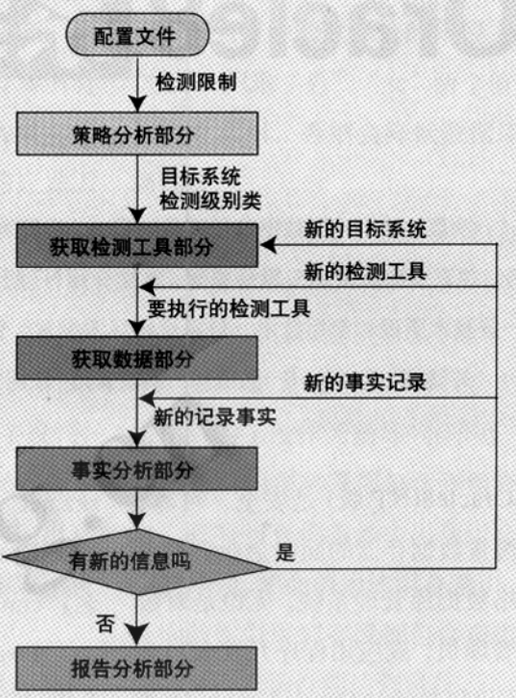

图 2 网络安全检测系统的逻辑结构

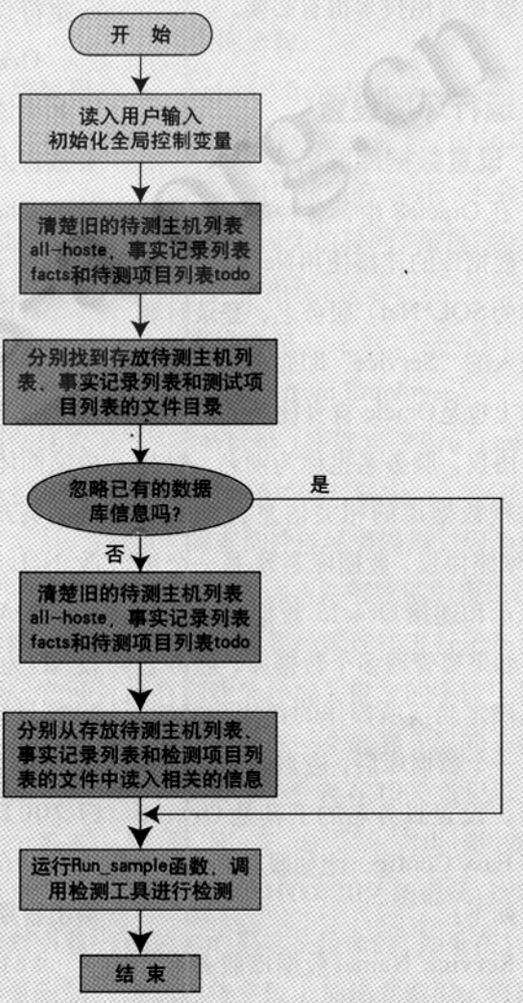

图 3 网络安全检测系统的流程图

(下接第68页)

(上接第 46 页)

4.4 Run_sample 函数的实现

Run_sample函数主要分为两大部分：即初始化部分、控制获取检测工具部分、获取数据部分和事实推断部分三大逻辑块的循环检测。

Run_sample函数的初始化部分进行每次检测前的初始化工作，即填写待测主机列表（若也要检测该主机所在的子网，则其子网上的所有主机都要填入列表），获取每个目标机的IP地址，验证其合法性。

Run_sample函数的第二部分则是循环执行以下四个函数，直到不再有新的信息产生时为止。

(1) process_targets函数：处理新的待测主机列表

① 根据检测规则初始化有关的检测数组；

② 查看每个待测试的目标主机是否处于活动状态；

③ 对每一个活动的目标主机分配相应的检测方案，放入待测项目列表todo队列中。

(2) process_todos函数：不断循环处理待测项目列表todo队列中的每一个项目，直到队列中不再有新的待测项目。

① 分别初始化用于存放新的待测项目列表的数组和临时数组；

② 每次从待测项目列表todo队列中取一个项目，分别获得检测目标（放入变量$target）、检测工具（放入变量$tool）和检测参数（放入变量$args）。当所用的检测工具符合该待测主机的检测级别类时，把以上变量传给执行调用检测的关键函数run_next_todo，实现用$tool中的检测方法按参数$args的要求对$target进行检测。

其中，run_next_todo函数是调用检测工具的接口函数，它根据调用过程传给它的检测目标（变量$target中）、检测工具（变量$tool中）和检测参数（变量$args中），调用bin目录下的检测工具。

(3) process_facts函数：不断处理事实记录队列facts，产生新的事实记录和新的待测项目todo，直到不再生成新的facts时为止。

该函数对事实记录facts中的每一个项目循环执行以下动作：① 分析推断主机的类型；② 分析推断主机所提供的服务；③ 分析推断主机的脆弱性；④ 分析推断主机的信任关系；⑤ 分析推断更新的待测项目；⑥ 分析推断更新的事实记录。

(4) save_sample_data函数：将测试结果保留到指定文件中，供下一轮测试循环使用。 (未完待续)