doi 10.3969/j.issn.1671-1122.2014.06.005

基于模糊测试的网络协议自动化漏洞挖掘工具设计与实现

孙哲[1]，刘大光[2]，武学礼[3]，文伟平[2]

（1. 中国工程物理研究院计算机应用研究所，四川绵阳 621900 ；2. 北京大学软件与微电子学院，北京 102600 ；3. 中国石油集团东方地球物理勘探有限责任公司，陕西长庆 710021）

摘 要 ：文章针对传统网络协议挖掘的缺陷，着重分析了传统网络协议的分析手段、漏洞类型、产生原因和挖掘方法。文章针对传统网络协议挖掘中协议的分析过程不能自动化、构造 Fuzz 的数据不符合网络协议格式规范和交互的流程导致无法深入快速地进行漏洞挖掘的缺点，提出了一种基于自动化协议分析算法、流量聚类分类算法、深度数据包检测技术、Fuzz 技术相互整合的自动化协议分析漏洞挖掘工具设计方案。文章设计了一套自动化协议分析的漏洞挖掘系统，给出了系统的工作流程和组织结构，以及各个模块的功能和相互之间的关系，实现了一个自动化协议分析漏洞挖掘系统的原型。文章的最大创新是通过自动化协议分析、流量聚类分类算法和 DPI 技术的有机结合，实现了自动化协议分析、自动形成测试路径的网络协议漏洞挖掘技术。

关键词 ：漏洞挖掘 ；协议分析 ；模糊测试 ；流量聚类

中图分类号 ：TP309 文献标识码 ：A 文章编号 ：1671-1122（2014）06-0023-08

Design and Implementation of Network Protocol Auto Vulnerability Mining Tool based on Fuzzing

SUN Zhe[1], LIU Da-guang[2], WU Xue-li[3], WEN Wei-ping[2]

(1.*Institute of Computer Application,China Academy of Engineering Physics,Mianyang Sichuan* 621900,*China*; 2. *School of Software & Microelectronics, Peking University, Beijing* 102600, *China*;3. *China Petroleum Group Dongfang Geophysical Exploration Co., Ltd.,Changqing Shanxi* 710021,*China*)

Abstract: Due to defects of traditional discovery in networking protocol, this paper analyzes methods of traditional network protocol analysis, vulnerability types, causes and discovery approaches, and disadvantages of traditional network protocol discovery. Thus, this paper proposes a design plan of automatic analysis and discovery tool based on integration of automatic protocol analysis technology, traffic clustering sorting algorithm, deep packet inspection technique and Fuzz. This paper designs a set of vulnerability discovery system for automatic protocol, which provides systematic working procedure and structure, and function of each module and their interrelations; and finally gives a system model realization, based on which vulnerability discovery is conducted to the FTP server software to verify validity and efficiency of the system design plan. The major innovation of this paper is the integration of automatic protocol analysis technology, traffic clustering sorting algorithm and DPI technology, which forms the network protocol vulnerability discovery technology that can conduct automatic protocol analysis and generate test path automatically.

Key words: vulnerability discovery; protocol analysis; fuzzing; flow clustering

0 引言

近几年来网络安全已经成为热议话题，网络安全从最早的 CIH 病毒，到后来的冲击波蠕虫病毒，到国内的熊猫烧香，再到美国的棱镜门事件，每一个安全事件的曝光都触动了公众的神经[1,2]。从前几年的 CSDN 密码泄漏事件就可以看出，不仅仅是普通百姓，就是专业的从业者对于网络安全的意识也是十分的淡薄。在信息化的时代，网络就是一个没有硝烟的战场。

收稿日期 ：2014-05-13

基金项目 ：国家自然科学基金 [61170282]

作者简介 ：孙哲（1987-），男，山东，硕士研究生，主要研究方向 ：系统与网络安全 ；刘大光（1985-），男，黑龙江，硕士研究生，主要研究方向 ：软件安全漏洞分析 ；武学礼（1975-），男，宁夏，工程师，本科，主要研究方向 ：软件测试 ；文伟平（1976-），男，湖南，副教授，博士，主要研究方向 ：网络攻击与防范、恶意代码研究、信息系统逆向工程和可信计算技术等。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

目前最火的安全概念就是 APT，这类攻击从 2003 年开始出现到 2008 年以后直线上升。APT 攻击的最重要特征就是基于一个或者几个漏洞组合进行渗透，控制计算机网络，相比 DDoS 攻击或其他类型的攻击危害范围更大、更加难以防范，因为它可以悄悄地不被用户察觉地进入对方的计算机网络从而控制计算机或者持续搜集信息。因为软件开发者的安全意识和开发水平参差不齐，重视功能和用户体验但轻视安全，导致了各类软件中的软件漏洞层出不穷。如果按照漏洞的危害划分，网络协议类型的漏洞尤为重要，因为这种类型的漏洞都是基于网络的，也就是攻击者可以远程进行攻击，而互联网是开放的，一旦漏洞被发现，那么攻击者就可以在世界的任何一个地方进行攻击而不被察觉 [3]。例如，2011 年 RSA 公司遭到入侵，攻击者给 EMC 公司的员工发送带有恶意代码的电子邮件进行钓鱼，EMC 公司的员工读取了攻击者发送的邮件，这封邮件中的恶意代码使用了最新的 flash 漏洞，绕过了所有的安全防护手段，成功地植入了木马。由此可以看出漏洞攻击在整个攻击中的重要性。

为了减少软件中的漏洞，尤其是网络远程类型的漏洞，也为了减少软件漏洞攻击的发生，安全人员应该及时查找漏洞，并且发布补丁。这对整个网络安全有着重要的意义。

1 协议分析技术及漏洞挖掘技术

1.1 协议分析方法

1.1.1 协议分析一般流程

1）抓取目标软件的网络流量。这种抓取是有目的的，会故意在输入过程中变换一些关键数据，以方便后续分析，对未公开协议尤为重要，如变换用户名密码，让流量产生对应的变化。

2）根据不同的功能进行流量划分。软件的网络协议大部分会基于命令字，或者相同功能函数产生的网络流量一般也相似，所以在协议分析中一般是按照流量的功能来划分流量的分类 [4]。

3）因为网络流量都是基于 TCP、IP 协议簇的，而 TCP、IP 协议是分层结构，那么就要按照 TCP/IP 协议来一层一层地对流量进行分析。TCP/IP 协议分为链路层、网络层、传输层、应用层。网络层记录的是网络流量的 IP 信息，但 IP 信息容易改变，所以网络层一般不在软件网络协议分析的考虑范围之内；传输层记录的是网络流量的端口信息，这个一般可以作为网络流量识别的特征，但是对于网络中 P2P 类型的软件，因为端到端的端口都在变，端口的识别就没有太大的意义。所以对于网络协议的分析，基本上可以说是应用层协议的分析，因为应用层才能够更好地区分不同软件、网络协议 [5]。

1.1.2 协议分析工具介绍

在分析网络协议的过程中，首先就需要进行网络抓包，然后按照 TCP/IP 协议格式按层来划分协议格式，最后再利用各种算法来对网络协议进行分析。

1）Wireshark。是一个网络封包分析软件。是目前使用最为广泛的网络抓包软件，其中内置了大量的已知网络协议，并且自动标识识别字段和结果，大大方便了协议分析的过程，并且随着版本更新，支持多种包格式的保存。在 Windows 平台下使用 WinPcap 包为底层进行抓包，直接和网卡交互，保证了抓包的全整型，并且提供各种过滤模式，大大方便了协议分析 [6]。

2）科来网络流量分析器。科来网络分析是一款国产的网络协议分析，最大的特点是根据不同的模型进行流量分析，并且还可以截获数据包，对数据报进行更改、重放，修改数据报和重放功能在网络协议分析中有很重要的实际意义。这也是这个网络工具的主要优势。

3）Tcpdump。这是 Linux 下最常用的网络分析工具，它支持各种命令字进行组合过滤，如与、或、非，端口的过滤等。

1.2 漏洞挖掘技术研究

1.2.1 缓冲区溢出漏洞

缓冲区溢出漏洞是一种古老的漏洞，也是最早发现的漏洞类型，产生的主要原因是程序在操作数据的过程中没有做边界检查，超长的数据导致了程序的缓冲区临界区域被覆盖，导致可执行地址被改写从而引发安全问题，这类漏洞可以直接导致系统被恶意代码控制。缓冲区是一片有限连续的内存区域，在 C 语言里最常见的就是缓冲区数组 [7]。

1.2.2 字符串溢出

字符串存在于各种命令行参数、软件交互的大部分数据和控制台输入中，加上 XML 这种标准的存在，使得程序之间的交互越来越多地使用字符串的形式，字符串管理和字符串操作的缺陷已经导致广泛的软件漏洞。

字符串漏洞主要有 4 种：无边界检查字符串复制、字

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

符串截断、空结尾错误、差一错误。

字符串漏洞是由于程序的数据输出函数对输入数据的格式解析不正确而触发的。这种字符串漏洞也属于一种缓冲区溢出，它的利用方式和触发方式都和缓冲区溢出的方式相类似。

1.2.3 格式化字符串漏洞

格式化字符串漏洞是最近新出现的攻击类型。格式化字符串漏洞的触发原因主要来自于程序员未对用户的输入进行检查。格式化字符串攻击一般可以得到任意内存数据，使攻击者可以绕过保护措施。当攻击者在不可信地址读取格式化字符串时也会触发类似漏洞。在 Windows 操作系统中字符串资源一般存放在 PE 文件中，如果攻击者可以重写、修改 PE 文件，那么就可以执行比格式化字符串更为直接的攻击,因为它们可以直接把恶意代码覆盖到正在执行的代码上。

为了解决这个问题，Windows 引入了地址空间随机化机制，在一定程度上限制了这种漏洞的发生，因为每一次程序得到的地址都是不同的，攻击者无法找到它的恶意代码的位置，从而无法执行。但是如果程序中的信息泄露，那么格式化字符串攻击可以泄露程序中地址布局信息，这样就可以使不可能发生的攻击变成可能。

1.2.4 指针覆盖漏洞

对于一个程序来说，内存中的数据往往都是程序的执行代码或者数据，程序将取出某个内存地址中的内容作为下一个将要执行的地址，此时就将保存这个程序执行地址的内存地址称为函数指针。

由于程序在运行过程中往往会将来自程序外部的数据也放到内存中，如果在存放的时候发生了写入过界，正好覆盖了函数指针，那么函数的执行流程就会发生改变。如果覆盖的地址是精心构造的，那么就有可能执行恶意代码，触发这个漏洞。这就是指针覆盖漏洞 [8]。

1.3 Fuzz漏洞挖掘技术介绍

1.3.1 Fuzz测试原理

Fuzz 技术实际就是一种测试技术，它的核心思想就是遍历所有可能的数据对程序进行测试，在测试过程中记录程序执行的状态，筛选出可能出 bug 的数据并且记录，供人继续分析。Fuzz 目前在漏洞挖掘中被应用到各个方面，如构造畸形数据的文件对文件操作的软件进行输入数据的 Fuzz，在 Fuzz 的过程中发现问题 [9]。

Fuzz 技术需要针对不同类型的漏洞进行针对化的数据构造。例如，对于缓冲区类的溢出构造畸形数据，就要构造一个超长的字符串 ；对于格式化字符串就需要构造一个格式化字符串类型的数据串进行测试，以发现程序是否能够正确执行。再如，为了发现整数溢出漏洞，就要在输入的数据里替换成最大整数或者最小整数，当程序执行时对数据进行加减操作就有可能触发漏洞。Fuzz 测试就是针对不同类型的漏洞进行数据的构造。

1.3.2 Fuzz测试分类

Fuzz 测试发现安全问题基本被分为白盒测试、黑盒测试和灰盒测试 3 种。白盒测试又称源代码审计，主要依赖于源代码的静态分析进行漏洞挖掘。黑盒测试又称功能性测试，即不涉及用户源代码，仅从程序的输入输出来判断程序问题，但不知道程序的内部细节。黑盒测试主要通过构造畸形数据，引发程序内部出错来进行漏洞挖掘。灰盒测试是介于白盒测试和黑盒测试之间的一种技术，主要通过程序的二进制代码的阅读或者动态调试获得程序片段来了解程序，同时使用黑盒测试的方法构造数据触发可能的漏洞点。灰盒测试主要针对没有源代码的程序，又需要进行精确的漏洞挖掘。灰盒测试需要漏洞挖掘者对逆向能力有很好的掌握 [10]。

1.3.3 Fuzz测试的一般步骤

Fuzz 测试通常有以下几个步骤 ：

1) 识别目标。首先根据软件自身的情况来选择测试方法，如是否有源代码，是否是远程目标，对软件的输入输出格式是否掌握，这类软件是否有过漏洞的“ 前科 ”，之前的漏洞点在哪里。

2) 识别输入。漏洞都有一个特点就是一定会接受用户的输入，包括本地程序的用户 UI 的输入，也包括远程 CS 程序网络数据报的交互，甚至包括网络连接的开始部分。而软件输入数据的识别对于后面的步骤最为重要，用户输入数据识别的程度决定了测试路径的广度和深度 [11]。

3) 生成测试数据。当对用户输入的数据有了一定了解之后，我们需要生成测试数据进行软件的交互，向目标软件提交我们生成的测试数据，触发 bug 的生成。在生成畸形数据时我们一般会根据不同的漏洞类型有针对性地生成

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

数据，如超长的字符串、格式化字符串等[12]。

4）执行测试数据。当生成了测试数据后，我们把测试数据提交给目标软件，包括本地打开测试数据文件，或者远程网络提交测试数据。针对不同类型的软件输入测试数据到软件中，这个过程一般要自动完成。

5）监视异常。当目标软件执行了我们提交的测试数据，我们需要监视目标软件的执行情况，如是否崩溃、是否抛出异常等，并把异常情况记录下来[13]。

6）分析测试结果。当测试完毕所有的畸形数据，需要人工来对测试的结果进行分析，看哪些测试的 bug 可以进行利用而转化为漏洞，哪些仅仅是软件的 bug，对安全性并不构成威胁。

2 自动化协议分析漏洞挖掘系统设计

2.1 总体设计思想

因为 TCP/IP 协议是一种高度有序的字节流，而软件在实现过程中，相同函数输入输出的数据在结构上总是类似的。基于这两点选择使用 LCS 算法进行报文协议的分析，而这个算法对于同一类流量可以取得比较好的效果，但如果样本中噪音过大，就会严重影响分析结果。使用机器学习中的聚类算法将流量首先进行分类，一个好的分类是基础，LCS 算法在分好的每一个类别里进行协议分析。聚类算法进行每一层分类的过程实际是构造一棵流量规则树的过程，当这棵树构建完毕，协议也就同时分析完毕。这棵树实际代表的是通信交互的路径，而对于下层 DPI 模块它又是一个规则树，DPI 模块根据这棵树来判断测试状态。不断根据规则树生成 Fuzz 数据直到一棵树遍历结束。所以自动化协议分析漏洞挖掘工具的难点在于：自动化提取特征算法、无监督的聚类算法、DPI 深度检测技术。下面就分别对这 3 种技术进行阐述。

2.2 特征提取算法

特征提取（longest common subsequence，LCS）算法最早应用在生物学中用来比较不同的生物的 DNA，比较两个生物的碱基对相似度。LCS 可以提取两个字符串的最大公共子串，也可以计算两个字符串的相似度。LCS 最简单的办法就是穷举所有的子序列进行比较，但是时间复杂度为 2 的指数，这对于长序列是不实际的。

从 LCS 算法描述中可以看出它是一个递归性质的问题，找出 A 和 B 的最大公共子串，可能需要找 A 和 Bn-1 或者 An-1 和 B 的最大公共子串，并且一直递归查找下去。如果 C[i,j] 是序列 Xi 和 Yi 的一个公共子串的长度，那么可以得到 LCS 算法的最优子结构。

LCS 算法的步骤描述如下：

1）构建一张表，以 A{1,2,…,I} 和 B{1,2,…,J} 为输入填入表中。

2）开始填表，填表的顺序是从左到右，从上到下依次填写，而填写的内容分为两部分，第一部分是 C[i,j] 的值，也就是截至到目前单元格最大的公共子串的长度是多少，第二个则是这个 C[i,j] 的值来源于哪个单元格。

3）从整张表的最右下端进行回溯，按照箭头所指的方向进行回溯。

4）得到 LCS 子串进行倒序得到的最大公共子串。

在计算出 LCS 字符串的长度后，我们就可以自己计算字符串之间的相似度，字符串之间的相似度定义为两个字符串最大公共子串的长度比上两个字符串的平均长度，化简通分得到下面公式：

$$simw(w_1,w_2)=\frac{2len(lcs(w_1,w_2))}{len(w_1)+len(w_2)} \quad (1)$$

在网络协议中每一个报文实际上都是一个字符串，而报文和报文之间的相似度和公共的字符串就可以视为一种特征，这就是 LCS 算法应用于协议解析的原理。

2.3 流量分类无监督算法

聚类又叫聚类分析，指的是把不同的样本分到不同的分类中，使分类的差异尽量小，而不同分组之间样本差异尽量大。聚类算法的分类方法又可以分为监督聚类算法和无监督聚类算法[14]。

监督聚类算法的代表是 K-MEANS 算法，其基本步骤描述如下：

1）从数据对象里任意选取 K 个对象作为出事聚类中心；

2）根据每个聚类对象的均值，计算每个对象与这些中心对象的距离，并根据最小距离重新对应对象进行划分；

3）重新计算每个聚类的均值；

4）计算标准测度函数，当满足一定条件，如函数收敛时，算法终止，如果条件满足则回到步骤 2）。

还有一种就是无监督聚类算法，这类算法不需要给出

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

实现具体的分类。比较典型的就是密度的聚类方法，以数据集的空间分布上的稠密程度为依据进行分类，无需预先设定簇的数量，因此特别适合针对位置内容的数据集进行分类。而网络流量的分类未知，两个样本之间相似度是可以衡量的，所以密度的聚类方法适用于网络流量的分类。

密度聚类算法的一般步骤是，只要一个区域中的点的密度大于某个阈值，就把它加到与之相近的聚类中去。对于区域内的每个对象，就是在给定的半径 e 的领域中至少要包含的对象的最小数目。这类算法可发现任意类型的聚类，且对噪声数据不敏感。代表算法有 DBSCAN、OPTICS、DENCLUE 等 [15]。

2.4 DPI深度检测技术

当流量分类完成以后也就说明规则树构造完成，之后便和目标软件进行通信，在通信过程中进行状态的识别和规则树的遍历。而数据流量的识别就需要使用 DPI 技术，这是因为规则树实际是基于应用层报文的一棵规则树，而对应用层报文的识别就需要使用 DPI 技术。下面对 DPI 技术做一个简略的介绍。

DPI 技术又叫深度检测技术，它是应用层的流量检测和控制技术，当数据流量通过防火墙或者其他流量网管时，网络流量设备会读取数据报的应用层报文进行模式匹配，当匹配到某种规则时，说明它是对应的某种预先的策略，是一种应用或者是某一种攻击。网络设备会根据用户配置和模式匹配的结果来触发相应的动作。

模式匹配分为单模匹配和多模匹配，它们都是字符串匹配的一种算法。在多模匹配中使用最多的应该是 AC 算法，它使用表的结构大大加快了字符串的匹配速度，但是这个算法的缺点是太耗内存。而单模最经典的匹配算法应该是 KMP 算法，但是 KMP 算法的效率实际上并不比 C 中的 strstr() 函数快。目前广泛使用的单模匹配算法是 BM 算法和 Sunday 算法 [16]。

3 自动化协议分析漏洞挖掘工具实现

3.1 系统总体设计

本系统主要基于分层的设计思想设计自动协议分析 fuzzing 漏洞工具，系统主要由数据包捕获模块、进程流量信息记录模块、背景流量过滤模块、流量分类及特征提取模块、构造规则树模块、fuzzing 畸形数据生成模块、流量识别模式匹配模块以及底层 TCP/IP 发包模块 8 个部分组成。

系统的工作流程是 ：系统首先启动数据报捕获模块和进程流量信息记录模块进行抓包，由用户控制停止时间，当用户停止截包时，进程流量信息记录模块同时停止工作。由背景流量过滤模块进行流量过滤，然后把目标进程的纯净流量进行包重组，提交给流量分类模块，流量分类模块根据密度聚类算法把流量进行分类，并提取出规则。由构造规则树模块把提取好的规则形成规则树，当规则树构成完毕后，系统将遍历整棵规则树。由 fuzzing 数据生成模块交给底层 TCP/IP 模块，把畸形网络报文发送给目标，底层 TCP/IP 模块接收到服务器 / 客户端返回的信息，提交给上层流量识别模块来判断走到规则树的哪个节点，从而判断生成下一个节点的报文，一直到目标发生异常，或者规则树遍历完毕结束 [17]。

3.2 主要功能模块设计

3.2.1 数据包捕获模块

数据包捕获模块主要进行数据报的捕获，可以使用 Wireshark、tcpdump 等现有的软件来实现，也可以自己调用 libpcap、WinPcap 等网络截包库来实现。本文采用了两种方式来实现数据报的截包。其中调用 libpcap 库函数进行截包，把每个数据报中的五元组信息保存下来，为进程流量信息记录模块提供输入。使用 Wireshark 自带的数据包截获保存工具，直接对网卡截包，并保存成各种类型的数据包文件，这样实现的好处是 Wireshark 作为商业软件，其实现效率和稳定性、兼容性都要比个人实现要完美得多。所以采用两种方式来实现数据包的截获，这样既兼顾了稳定性、效率，又实现了自己个性化的需求。

3.2.2 背景流量过滤模块

背景流量过滤模块主要负责从混杂的网络流量中过滤出目标进程的流量链接。因为网卡中的网络数据包非常混杂，这样对于分析网络数据非常不利。为了能够更好地分析网络流量，需要把不需要的干扰网络流量去除，所以在这里增加了背景流量过滤模块。因为程序在操作系统中是以进程的形式呈现的，所以只要把目标进程对应的网络流量过滤出来就能达到目的，而过滤网络流量主要是通过五元组，即源 IP、目的 IP 、源端口、目的端口、TCP/UDP。在 Windows 操作系统中提供了 netstat-nbo 命令用来查询网络流量和进程之间的对元关系。 把网络中的数据包对应的五

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

元组信息和这张表中的信息进行比对，对目标进程的五元组信息进行保存。

当用户停止截包，根据目标进程的五元组信息，过滤保存下来的数据报中的链接报文，把 TCP 分包的报文进行包重组，最后形成分析报文。

3.2.3 流量特征提取模块

流量特征提取模块主要是根据包的次序来进行分层，然后把同一层的报文进行特征提取。特征提取主要包括以下几个方面 ：

1) 长度特征。长度特征的提取比较简单，检查每一层的同一组内的报文长度是否相同，如果相同那么可以认为是长度特征。

2) 包内 byte_jump 特征。这类特征主要是指包内长度特征的跳转，如一个数据包的前两个字节代表的是数据包的长度。图 1 描述的数据包的前 4 个字节代表整个数据报的长度，\x00\x00\x00\x01 代表 16 个字节而数据包长度也为 16 个字节。通过不断尝试可以得到图 1 所示的特征。

图1 Wireshark报文示意图

3.2.4 构造规则树模块

规则树的构成模块是系统的关键模块，起到承上启下的作用。它其实就是一个自动状态机，由聚类算法将网络流量进行分类，每一个聚类的内部进行特征的提取。在一层聚类划分完毕后，划分下一层报文时，在上一层聚类分配的结果上递归进行分配，这样递归的最后结果就是形成了一棵 m 权规则树。

因为一个软件中的不同功能函数可能对应不同类型的报文，如一个 FTP 软件一般会有登录、注销、上传、下载、查询等不同的功能，而不同的功能对应的报文也不同。也就是说程序的一个函数对应一类报文，在不同功能之间报文提取特征是毫无意义的，所以本系统没有采用传统的根据数据报长度、发包速度等来进行分类，而是根据协议、端口、报文的相似度来进行报文的分类。对于系统的使用者来说，为了使系统识别更加准确，希望可以每次只测一种功能，这样分类会尽量少。另外，在登录时尽量使用不同的用户名和密码进行登录，这样可以让自动协议识别模块尽可能准确识别出协议的命令字和可变字段。

1) 根据 TCP、UDP 协议进行分组。这是因为一个功能一般只使用其中的一种，很少有两种同时使用。TCP 一般用于命令字的传输，UDP 报文一般用来传输语音视频对丢包不敏感的功能数据。

2) 使用端口划分。对于服务器类的软件会使用固定端口来进行监听，而一般不同的端口对应不同的功能，但是也不绝对。

3) 使用机器学习中的无监督聚类算法。使用字符串距离作为衡量标准，设定阈值 d 来进行报文的分类。这样避免了噪声对特征提取的影响，使分类结果尽量准确。但是分类结果的准确性很大程度上依赖于阈值的设定，所以阈值的选取需要人主观进行判断，需要一定的协议分析经验才能设定一个比较合适的阈值。

使用无监督聚类算法构造规则树具体过程如下 ：由报文组成一个报文矩阵，报文的每一列代表的是一条链接，矩阵的行代表了同一层的报文。第一层报文进行聚类划分，划分完毕，第二层开始划分，第二层划分的开始继承上一层的划分结果，在上一层划分的结果之上继续划分，最后就形成了一棵规则树。

3.2.5 fuzzing畸形数据生成模块

fuzzing 畸形数据生成模块主要是根据规则和 fuzzing 策略来生成 fuzzing 畸形数据。首先根据规则来生成符合协议通信的数据包，规则没有规定的部分都是可以生成畸形数据的位置。在生成了符合协议特征的数据后，开始生成畸形数据对数据报文进行填充。最后把数据报文提交给底层 TCP/IP 发包模块进行 Fuzz 测试的交互。

畸形数据生成策略 ：

1) 超长字符串。用来检测字符串溢出。

2) 整数溢出。使用 \x00、\xff 来进行填充。

3) 格式化字符串。

3.2.6 流量识别模块

流量识别模块主要用来确定规则树的位置。流量识别模块主要使用模式识别算法来进行字符串的模式匹配，确定规则是否命中。模式匹配算法采用的是 Sunday 算法和

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

strstr() 混合查找，当字符串长度大于 4 时使用 Sunday 算法来进行查找，当字符串长度小于 4 时使用 strstr() 来进行查找匹配。当规则命中时根据规则树的位置来确定提交给 fuzzing 畸形数据生成模块的规则，生成下一步要发送的畸形数据。

具体流程如下：

1) 首先根据底层 TCP/IP 提交上来的数据进行模式匹配；

2) 根据模式匹配结果确定规则树的位置；

3) 检查本节点 fuzzing 数据池是否为空；

4) 如果为空，选择规则树的下一个节点发送给 fuzzing 畸形数据模块生成畸形数据；

5) 如果不为空，提取出 fuzzing 数据池中的一个 fuzzing 样本发送给底层发包模块进行发包。

3.2.7 底层TCP/IP发包模块

该模块提供了最基础的客户端、服务端的发送接收功能，包括 TCP 和 UDP 两种协议的客户端和服务端的接收和发送，发包模块接收到目标主机发过来的报文提供给上层流量识别模式匹配模块，由流量识别模块来确定规则树的位置，并接收 fuzzing 畸形数据生成模块生成的畸形数据发送给目标主机，进行 Fuzz 测试。

具体流程如下：

1) 判断要发送的数据包类型；

2) 初始化客户端、服务端信息；

3) 发送 fuzzing 畸形数据模块提交的数据；

4) 接收对端服务器返回消息提交给模式匹配模块。

4 Auto_Fuzz 工具漏洞挖掘实验分析

4.1 实验环境

为了使本系统能够得到充分的验证，本节尽量使用工作中真实的环境进行验证，并且利用本系统的 demo 程序对目标软件进行测试，并针对测试结果给出分析验证测试的有效性。

本文的测试环境是在一个真实的局域网内进行流量抓包，在试验的后半阶段进行 Fuzz 测试时使用的环境是本机和虚拟机进行 Fuzz 交互，这样可以尽量减少其他网络流量的干扰，便于观察，提高准确性，同时也有利于漏洞现场的保存和恢复，以便于后面的分析。测试环境使用的都是 Windows 操作系统，抓包系统使用的是 Win 7，虚拟机内不只使用的是 Windows XP SP3 系统。在 Win 7 中运行着本系统的 Auto_Fuzz 程序。

4.2 某FTP软件漏洞挖掘实验

FTP 协议是一个典型的命令字的明文协议，其协议格式为“ 命令字 + 内容 ”。下面按照系统流程对模块进行一一验证。首先进行 FTP 协议的自动分析，再根据协议分析的结果构造 Fuzz 测试路径，测试数据包进行漏洞挖掘。

4.2.1 FTP协议自动分析

根据流程首先进行数据包截获。数据报截获分为两部分，第一部分是目标进程和五元组信息捕获，第二部是即时流量捕获。首先设置网卡参数为 eth1，然后开始抓包。因为网卡中通过了各种各样进程的数据包，所以流量非常大。

在过滤了背景流量，得到了想要的目标进程中的流量，就开始进行自动的协议分析。抓到的测试样本只有两个，而且这两个样本有意让它属于一个分类，因为再好的机器学习算法也没有人识别得准确。为了得到更好的测试效果，在抓取流量的过程中有意除去了相同分类的流量，实际使用中这样也会提高协议分析的准确性，因为聚类算法的作用也就是进行流量分类，提高协议分析的准确度。

在自动协议分析过程中还绘制出了一个 FTP 协议的交互的网络轨迹。例如，在链接开始服务器会给客户端返回一个“ 200+ 欢迎信息 ”的报文，然后是用户名和密码的认证。认证成功 FTP 服务端会发送 230 个状态字，代表用户已经登录成功，然后客户端根据用户自己的意愿或者设定发送 FTP 命令。一般的 FTP 客户端软件的这条网络轨迹其实就是一条测试规则，后面的 FUZZ 模块和 DPI 模块会根据这条测试路径对软件漏洞进行测试。

4.2.2 FTP协议测试路径生成和漏洞挖掘

根据自动协议分析模块分析出的结果可以看到，FTP 协议是“ 命令字 + 参数 +\r\n ”，那么根据样本中的协议轨迹进行对应 Fuzz 策略的测试。基于之前对漏洞类型的调研总结出了几种简单的 Fuzz 策略：

1) 字符串超长漏洞。在命令字之后添加一个超长的字符串进行测试。一般在后门增加 1000 个字符，因为 TCP 最大的包长为 1406，而 UDP 最大包长为 1460，一般不会超过最大包长。这里如果有分包情况的话，会直接加到 10000 的长度进行测试，看对方的缓冲区是否会溢出。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

2) 在字符串中加入一些大整数的十六进制数，看是否会对目标软件造成整型溢出。

3) 在字符串中加入格式化字符串，看目标软件是否有字符串格式化错误。

4) 在字符串中加入非正常字符，看目标软件是否会有解析错误，如！@# ￥%……&*~ 等。

4.2.3 漏洞结果分析和利用

经过 FUZZ 模块的测试，发现这个软件在 FTP 协议交互过程中有一个命令字的溢出漏洞。在 FTP 交互过程中，Fuzz 模块对用户名 USER、密码 PASS 等各个命令字进行超长字符串、格式化字符、整型数字替换等测试。在这个测试节点没有发现 bug 异常的时候，下一次就使用原始报文不做修改地进行发送，这样就保证了报文一定是符合协议的，一定可以继续交互下去。

FTP 软件界面如图 2 所示，通过测试，发现了两个漏洞：

1) 第一个是 CWD 命令后面带超长字符串溢出漏洞，CWD 缓冲区溢出长度为 272。

2) 第二个是 MKD 命令后面带超长字符串溢出漏洞，其中 MKD 的溢出长度为 269。

图2 FTP软件界面

5 结束语

本文首先对协议分析技术和漏洞类型及利用原理进行了分析和总结，参考传统的网络协议提出了自动化协议分析漏洞挖掘工具的设计方案。方案弥补了传统手动分析协议和手动写测试路径效率低下、测试路径不全、畸形数据交互无效、无法深入测试漏洞的缺点。自动协议分析漏洞挖掘系统根据自动协议分析的协议格式和样本的网络轨迹进行测试，保证了交互的正确性和测试的深入性。同时采用分层的设计思想，使得用户在使用过程中可以对自动化的结果进行微调，保证了测试的准确性，同时也让用户完成特定需求的测试，进行更深程度的网络协议漏洞挖掘。

本文工作主要体现在以下几个方面：

首先调研了协议分析的各种典型方法、技术和工具，并对各种方法的优缺点和使用范围进行了总结。调研了目前各类漏洞的类型、产生原因和利用方法，并进行了详细的阐述和总结。通过调研目前网络 Fuzz 工具，针对现有的网络 Fuzz 工具的缺点，提出了自动化协议分析技术，结合 DPI、Fuzz 等技术阐述了自动化测试网络协议测试框架的实现原理。设计了一个自动化协议分析的漏洞挖掘工具，给出了系统的详细设计、工作原理和各个模块的实现方案。使用自动化协议分析漏洞挖掘工具对一些服务器软件进行漏洞测试，并给出了测试结果并对测试结果给出了分析。（责编 马珂）

参考文献

[1] Nagappan N, Ball T. Static analysis tools as early indicators of pre-release defect density[M]. New York: ACM, 2005.

[2] Wang T, Wei T, Gu G,et al. TaintScope: A checksum-aware directed fuzzing fool for automatic software vulnerability detection[C]. In: Proc. of the IEEE Security & Privacy 2010: Oakland.

[3]Merwe J, Caceres R, Chu Y. H, Sreenan C. A Tool for Monitoring Internet Multimedia Traffic[C]. ACM Computer Communication Review, Oct. 2000, vol.30:48-59.

[4] Bush W, Pincus J, Sielaff D. A static analyzer for finding dynamic programming errors[J]. Software-Practice and Experience, 2000, 30(7): 775-802.

[5]Kurt Tutschku. A Measurement-Based Traffic Profile of the eDonkey Filesharing Service[C]. PAM,2004:12-21.

[6]Erman J,Arlitt M,Mahanti A.Traffic Classification Using Clusteing Algorithms[C]//Proc of ACM SIGCOMM'06,2006:15-20.

[7]Marshall Beddoe.The Protocol Informatics Project[EB/OL]. http://www.4tphi.net/~awalters/PI/PI.html,2004.

[8] 练琪 . 基于聚类分析的应用层流量识别研究 [D]. 长沙 ：湖南大学 , 2010.

[9] 崔月婷 . 基于分类算法与聚类算法流量识别系统的研究 [D]. 北京 ：北京邮电大学 , 2010.

[10] 彭青白 . 缓冲区溢出漏洞的挖掘与利用方法研究 [D]. 武汉 ：华中科技大学 , 2009.

[11] 李根 . 基于动态测试用例生成的二进制软件缺陷自动发掘技术研究 [D]. 长沙 : 国防科学技术大学，2010.

[12] Sen K, Marinov D, Agha G. Cute: a concolic unit testing engine for c[C]. In ESEC/FSE-13: Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering, pages 263-272, 2005.

[13] Brummayer R, Biere A. Boolector: An Efficient SMT Solver for Bit-Vectors and Arrays[M]. Berlin, Heidelberg: Springer-Verlag, 2009.

[14]Oeh lert P. Violating assumption s with fuzz ing [J]. IEEE Security& Privacy M agaz ine, 2005,3(2):58-62.

[15]BrianChess,JaeobWest 著 . 安全编程代码静态分析 [M]. 董启雄，韩平，程水敬等译 . 北京 : 机械工业出版社 ,2008.

[16] 斯顿著 . 模糊测试 ：强制性安全漏洞发掘 [M]. 龚波，冯军，徐雅丽，译 . 北京 ：机械工业出版社，2005.

[17] 姜明宇 , 马文丽 , 郑文岭 . 基于遗传算法的基因表达数据的 K-均值聚类分析 [J]. 上海生物医学工程，2006，(3):151-154.

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net