文章编号：2095-2163(2020)04-0162-03 中图分类号：TP309 文献标志码：A

基于自动化的渗透测试

高宏佳[1]，李世明[1,2]
（1 哈尔滨师范大学 计算机科学与信息工程学院，哈尔滨 150025；
2 上海市信息安全综合管理技术研究重点实验室，上海 200240）

摘 要：为有效应对网络威胁，提高系统的安全性，本文根据人工渗透原理及过程，提出一种基于自动化的渗透测试方法，对试运行的系统进行渗透测试，尽早检测系统中存在的安全问题。本方法通过对目标主机的自动扫描、自动判断、自动渗透等攻击行为，完成了系统的漏洞检测。实验表明：该方法可提高渗透测试的总体工作效率，节约了时间成本。

关键词：渗透测试；漏洞扫描；网络攻击

Penetration test system based on Automation

GAO Hongjia[1], LI Shiming[1,2]
（1 College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China;
2 Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai, 200240, China）

【Abstract】 In order to effectively threaten the network and improve the security of the system, this paper conducts penetration test on the system in trial operation, and detects the security problems in the system as early as possible. According to the principle and process of artificial penetration, this paper proposes a penetration test method based on automation. This method completes the vulnerability detection of the system through the automatic scanning, automatic judgment and automatic penetration of the target host. The experiment shows that this method can improve the overall work efficiency of penetration test and save the time cost.

【Key words】 Penetration Testing; Vulnerability Scanning; Network Attack

0 引 言

开放的互联网上，网络安全事件频发，各种网络攻击行为已经严重威胁着网络服务的安全运行。如何进行预防和在线检测成为网络安全防护的研究热点。为最小化安全风险、减少被入侵概率，有必要在系统正式运行前进行渗透测试，事先进行攻击模拟。为降低在渗透测试过程中大量人工操作成本，有效提高测试效率，本文提出了基于自动化思想的渗透测试策略，来提高渗透测试效率。国内外也有相关研究成果被提出，如基于 Agent 智能模型思想的渗透测试系统[1]。历年各级各类网络攻防赛、黑客大赛等也为渗透测试自动化做出了贡献，促进了该研究热点的研究和发展。

1 渗透测试攻击

渗透测试攻击，是通过模拟恶意黑客的攻击手段，实现对被测系统的安全性测试与评估。它主要包括前期信息采集、交互获取操作系统、目标 IP、开放端口、提供服务 SQL 漏洞及注入点等等。随后对所获取的信息进行审核处理，获取外网 IP 以及内网 IP 并进行渗透测试、提权等操作。最后，通过大量的分析，自动得出渗透测试报告，报告中对所有的漏洞，包括 HTTP 请求头泄露、文件泄露、XSS 泄露、点击劫持、是否 SQL 注入成功、是否设置 X-content-type-options 头等情况，以及所有的端口开放情况、服务情况进行整理、对主机的漏洞进行评级。最终給出的报告有利于对主机进行安全分析，尽量减少入侵风险。

2 自动化渗透测试

2.1 总体设计

本文自动化渗透攻击核心总体为：(1)输入目标主机域名及参数；(2)输入参数执行自动化渗透；(3)生成渗透测试报告。执行流程如图 1 所示。

通过分析部分渗透攻击技术，结合并分析漏洞特点及复杂性，现阐述本系统重要模块如下：

(1)信息收集。渗透测试首先从搜集攻击目标的基本信息开始，经过一系列分析、排除、进一步搜集的迭代过程。本文创新性利用 python 代码编程进行自动搜集网站 Title 和 banners，通过对外网和内网的探测来获取目标主机等有效信息，为后面的测试做好前期准备[2]。

作者简介：高宏佳(1999-)，女，本科生，主要研究方向：网络与信息安全、漏洞挖掘；李世明(1976-)，男，硕士，副教授，主要研究方向：网络与信息安全、物联网技术、数据挖掘。

收稿日期：2020-02-03

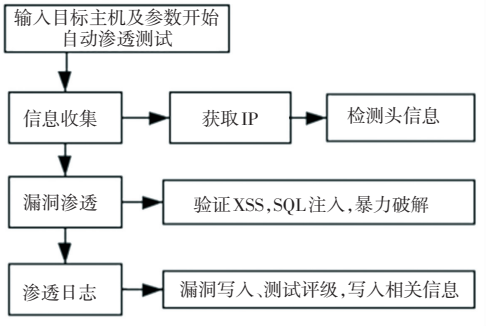

图 1　渗透攻击流程

Fig. 1　Penetration attack process

(2)漏洞扫描渗透。为简化实验,本文利用已知漏洞对目标 web 应用可能存在的利用点进行检测与利用[3];通过检测并成功利用漏洞,可以得出渗透测试的凭证,并依据 HTTP 协议的通用性,改变 HTTP 协议的内容以达到 web 应用系统渗透检测利用的目的,对其存在已知漏洞进行渗透攻击[4]。

(3)渗透日志。渗透日志是记录在渗透过程中渗透过程所获得的信息(如端口扫描、弱口令等)、对渗透过程以及攻击过程中出现的其它问题加以整理并进行文档化记录。

2.2　自动化渗透测试的实现

本文自动渗透测试利用 python 语言编程,来搜集网站 Title 和 banners,对内网设备探索,快速确定目标,Title 可由 BeautifulSoup 直接获取。以利用 python 语言进行 SQL 盲注为例[5],采取二分查找算法实现信息检索[6],伪代码描述如下:

INPUT: $I=\{ip_i \mid ip_i \in IPset, i \in Z^+\}$ // *IPset*:被攻击目标 IP 集合

OUTPUT: $O=\varnothing \cup \{<u_i, p_i> \mid u_i \in U, p_i \in P, i \in Z^+\}$ //输出结果

//其中, U,P 分别为破解时用的用户名字典和密码字典

connecting ip_i

Initialization //初始化

define chardataU[U_MAX] //

define chardataP[P_MAX] //

dataU = U; // 将用户名字典拷贝到数组中

dataP = P; // 将密码字典拷贝到数组中

Asc_sort(dataU);　//对用户名字典数据按升序排序

Asc_sort(dataP);　//对密码字典数据按升序排序

int low, high, mid;

char u_i , p_i ;

int low=0,high,mid;

high=MAX-1;

/*用户名检测*/

Low=0;

high= U_MAX -1;

while(low! =high)

{

mid=(low+high)/2;

if (dataU[mid]! = u_i)

{

if(dataU[mid]> u_i)

high=mid-1;

else if(dataU[mid]< u_i)

low=mid+1;

}

else　return u_i ;

} /*密码检测*/同上

通过模型得到盲注实例,部分关键代码如下:

```
url_dump = 'http://%s/? id=-1\' union select {inc} if( ord( mid( ( select group_concat( {column_name} ) from {table_name} ), {pos}, 1)) {op} {ord}, sleep(2.5), 1) ;--+' % url_part
req_count = 0
sleep_count = 0
def binary_search(url, low, high):
gt = '>'
lt = '<'
eq = '='
while low <= high:
mid = (low + high) // 2
if judge(url % (eq, mid)):
return mid
elif judge(url % (gt, mid)):
low = mid + 1
else:
high = mid - 1
return -1
def dump_binsearch(column_count, table_name, column_name):
data = ''
pos = 1
inc = make_union(column_count)
while True:
```

```
url = url_dump.format(inc = inc, pos = pos, op = '%s', ord = '%d', table_name = table_name, column_name = column_name)
ord = binary_search(url, low, high)
if ord > -1:
data += chr(ord)
pos += 1
continue
else:
return data
```

3 实验结果与分析

3.1 自动化渗透测试实验环境

本仿真实验测试硬件配置环境为 CPU 酷睿 I5，2.3GHz 主频，系统内存 8GB，自动渗透测试系统采用 python3.7 开发，采用 kali linux 虚拟机作为测试主机：

测试是基于 python 的 kali+linux 渗透测试需要而搭建的 selenium + firefox + geckodriver + beautifulSoup 环境。

其中：

(1) Selenium 是一款 Web 程序测试工具，不受浏览器、语言等限制，能够完成自动化测试功能。

(2) geckodriver+firefox

Geckodriver 采用 WebDriver 兼容模式，WebDriver 协议描述以及 HTTP API 编程接口，可实现 Marionette 远程协议通信。

(3) BeautifulSoup

将 Beautiful Soup 添加到 python 环境中，将获得更多库函数。并具有分析搜索结果、数据抓取、unicode 编码转化、UTF-8 编码输出等功能，起到 python 解释器作用。

3.2 实验结果

实验过程中通过直接输入目标主机的 IP 地址和相应参数，自动进行信息收集、渗透攻击和日志生成。

攻击靶机过程中获取了目标主机类型、端口、服务、网络类型、网络安全和漏洞等信息，实验结果如表 1 所示。

3.3 实验分析

本项目通过对目标主机进行渗透实验得出渗透测试中易受攻击的渗透点、可利用代码、开放端口、SQL 注入点、文件上传漏洞、接触漏洞等。此外渗透测试可能带来一些风险，如因频繁扫描、违反安全策略等可能出现系统崩溃、运行出错或账号锁定，造成渗透测试被迫中止。此外，也可采用适当技术措施来降低风险[7]。如对测试任务分解、约束扫描测试策略、规则测试危险时间段等。本实验的弊端在于渗透时间过长，渗透测试需要大量的前期信息搜集，对于防护较好的目标主机需要更长的渗透时间。

表 1 渗透攻击信息表

Tab. 1 Penetration attack information

目标主机	是否成功	开放端口	获取信息
49. 235. 44.81	成功	22,80,443	可匿名上传下载、爆破、嗅探、提权
192.168. 1.135	成功	139, 22, 445, 10000	可劫持漏洞，不含 X-content-type-option 头
192.168. 1.131	成功	80, 81, 82, 84, 90,1433,444	发现 robots 目录、泄露 admin 目录、HTTP 头含有 XSS 漏洞、可劫持漏洞

4 结束语

本项目可以自动完成网络下的安全扫描、探测、利用等工作，具备全面、低干扰的特点。利用 python 进行信息采集搜集网站 Title 和 banners，对内网设备探索，快速确定目标。具有查找注入点等应用漏洞、破解数据库结构、爆破弱口令、远程指令执行、外网注入、内网扫描等功能，能够发现指定目标的漏洞点，对漏洞进行综合利用，最终获取带有指定关键字的文件。减轻安全测试给业务带来的干扰。

由于本文的研究还处于探索自动渗透测试阶段，有诸多研究没有开展。在后继工作中，还将扩展对渗透测试自动化更多的研究，进一步优化信息收集、渗透测试工具利用优化等。

参考文献

[1] 李浩杰，裘国永. 基于自动化渗透测试的分析[J]. 电子设计工程，2015，23(22)：25-28.

[2] 严俊龙. 基于 Metasploit 框架自动化渗透测试研究[J]. 信息网络安全，2013(02)：53-56.

[3] Balasubramanian N, Askarunisa A, Ruba A. SOS-WS Host Shield: A sketch-based Service Oriented Shield against web application business layer IDS attacks [J]. Computer Communications, 2020, 153: 626-631.

[4] RODRÍGUEZ G E, TORRES J G, FLORES P, et al. Cross-site scripting (XSS) attacks and mitigation: A survey[J]. Computer Networks, 2020, 166: 106960.

[5] ZHOU T, ZANG Y, ZHU J, ET AL. NIG-AP: a new method for automated penetration testing [J]. Frontiers of Information Technology & Electronic Engineering, 2019, 20(9): 1277-1288.

[6] Wes Masri, Sam Sleiman. SQLPIL: SQL injection prevention by input labeling[J]. Security and Communication Networks, 2015, 8(15).

[7] Gupta, Sharma. Detecting attacks in high-speed networks: Issues and solutions [J]. Information Security Journal: A Global Perspective, 2020, 29(2).