# 计算机网络攻防实验课

第4周

陈健

chenj@nju.edu.cn

# Linux靶机渗透测试

- 主机发现
- 服务发现
- 漏洞发现
- 漏洞利用

# 主机发现：ping

- 扫描网段
  - for octet in {1..254}; do ping –c 1 10.0.2.$octet –W 1 >> pingsweep.txt & done
  - cat pingsweep.txt | grep “bytes from”
  - cat pingsweep.txt | grep “bytes from” | cut –d “ “ –f4 | cut –d “:” –f1 > targets.txt

# 主机发现：nmap

- sudo nmap –sn –iL ranges.txt –oA pingsweep –PE
  - -sn: Ping Scan - disable port scan
  - -iL : Input from list of hosts/networks
  - -oA : Output in the three major formats at once
  - -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
- grep “Up” pingsweep.gnmap
- grep “Up” pingsweep.gnmap | cut –d “ “ –f2 > targets.txt

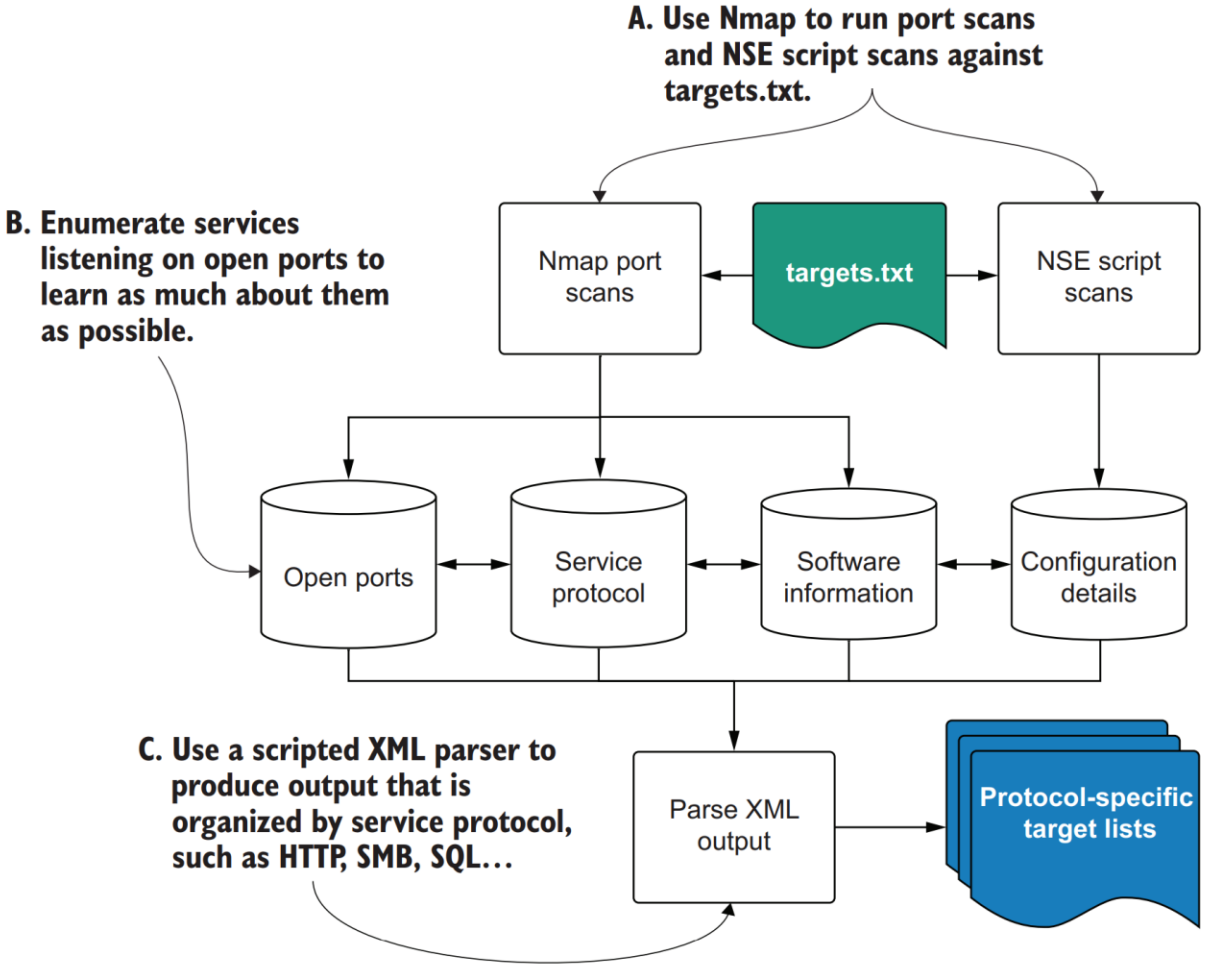
# 主机发现：RMI端口发现

- Top five RMIs
  - Microsoft Remote Desktop (RDP): TCP 3389
  - Secure Shell (SSH): TCP 22
  - Secure Shell (SSH): TCP 2222
  - HTTP/HTTPS: TCP 80, 443
- nmap –Pn –n –p 22,80,443,2222,3389 –iL ranges.txt –oA rmisweep
  - -Pn: Treat all hosts as online -- skip host discovery
  - -n/-R: Never do DNS resolution/Always resolve
  - -p : Only scan specified ports
- nmap –Pn –n –p 22,80,443,2222,3389 –iL ranges.txt –oA rmisweep --min-hostgroup 256 --min-rate 1280
- cat rmisweep.gnmap | grep open | cut –d " " –f2

# 主机发现：其他方法

- DNS brute-forcing
  - atk6-dnsdict6
  - https://github.com/blark/aiodnsbrute
- Packet capture and analysis
  - Wireshark
  - tcpdump
- Hunting for subnets
  - sudo nmap –sn 10.0-255.0-255.1 –PE --min-hostgroup 10000 --min-rate 10000

# 服务发现

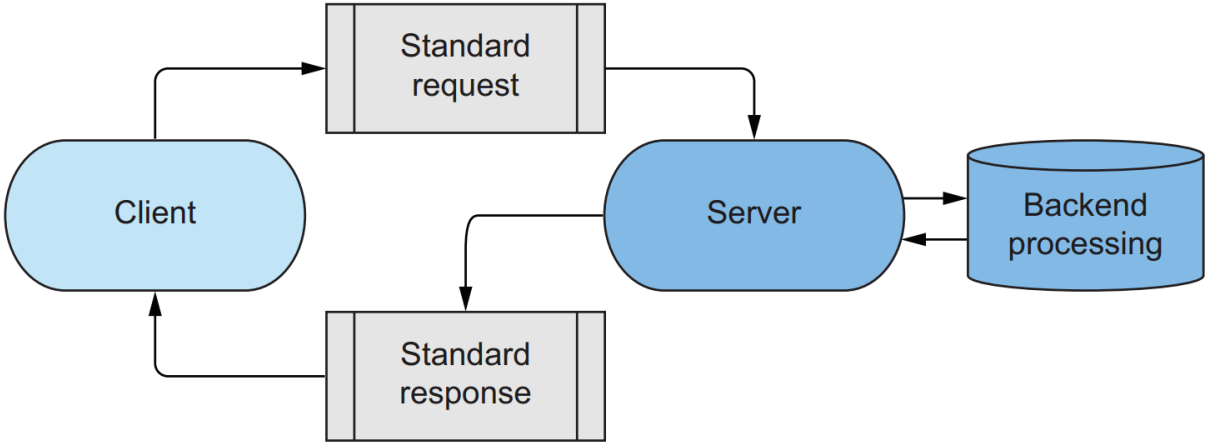

A. Use Nmap to run port scans and NSE script scans against targets.txt.
B. Enumerate services listening on open ports to learn as much about them as possible.
C. Use a scripted XML parser to produce output that is organized by service protocol, such as HTTP, SMB, SQL…
Nmap port scans
targets.txt
NSE script scans
Open ports
Service protocol
Software information
Configuration details
Parse XML output
Protocol-specific target lists

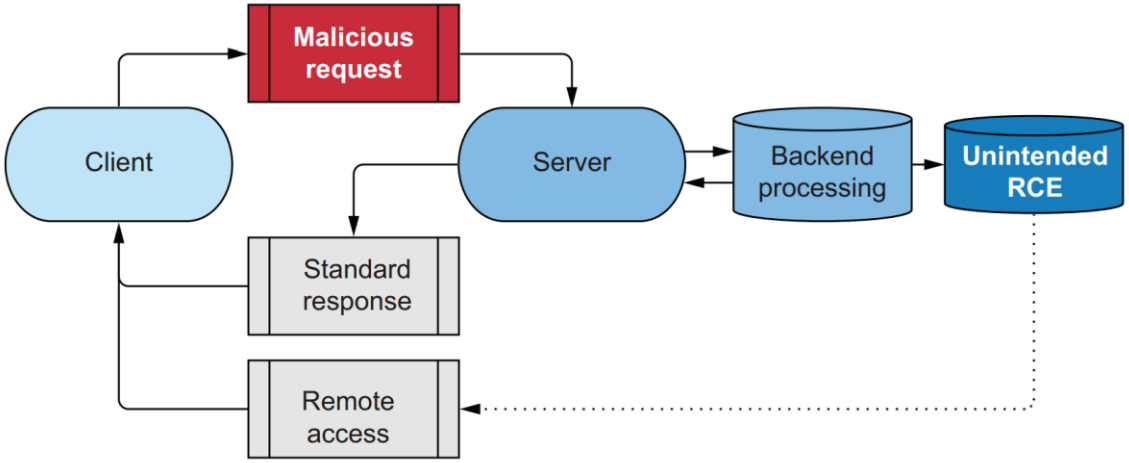
# 网络服务

A **network service** can be defined as any application or software that is listening for requests on a network port from 0 to 65535.

The **protocol** of a particular service dictates the proper format of a given request as well as what can be contained in the request response.

# 网络服务的请求和响应


Standard request
Client
Server
Backend processing
Standard response


Malicious request
Client
Server
Backend processing
Unintended RCE
Standard response
Remote access

# Network service banner

```
# curl --head 10.0.2.7
HTTP/1.1 200 OK
Date: Thu, 01 Oct 2020 07:14:26 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Content-Type: text/html
```

**This service is using the HTTP protocol**

**This is a Apache web server, Version 2.2.8**

**It's using PHP.This means the server is likely talking to a backend database server.**

# 快速端口扫描

# nmap -Pn -n -p 22,25,53,80,443,445,1433,3306,3389,5800,5900,8080,8443 -iL targets.txt -oA quick-sweep

```
Nmap scan report for 10.0.2.5
Host is up (0.00058s latency).

PORT     STATE  SERVICE
22/tcp   open   ssh
25/tcp   open   smtp
53/tcp   open   domain
80/tcp   open   http
443/tcp  closed https
445/tcp  open   microsoft-ds
1433/tcp closed ms-sql-s
3306/tcp open   mysql
3389/tcp closed ms-wbt-server
5800/tcp closed vnc-http
5900/tcp open   vnc
8080/tcp closed http-proxy
8443/tcp closed https-alt
MAC Address: 08:00:27:F6:2F:66 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.04 seconds
```

# 端口扫描

| Port | Type |
|---|---|
| 22 | Secure Shell (SSH) |
| 25 | Simple Mail Transfer Protocol (SMTP) |
| 53 | Domain name service (DNS) |
| 80 | Unencrypted web server (HTTP) |
| 443 | SSL/TLS encrypted web server (HTTPS) |
| 445 | Microsoft CIFS/SMB |
| 1433 | Microsoft SQL server |
| 3306 | MySQL server |
| 3389 | Microsoft remote desktop |
| 5800 | Java VNC server |
| 5900 | VNC server |
| 8080 | Misc. web server port |
| 8443 | Misc. web server port |

# 完整端口扫描

- nmap -Pn -n -iL targets.txt -p 0-65535 -sV -A -oA full-sweep --min-rate 50000 --min-hostgroup 100
  - -sV: Probe open ports to determine service/version info
  - -A: Enable OS detection, version detection, script scanning, and traceroute

# 提取执行的NSE脚本

```
# cat full-sweep.nmap | grep '|_' | cut -d '_' -f2 | cut -d ' ' -f1 | sort -u | grep ':'
```

```
ajp-methods:
clock-skew:
ftp-anon:
http-favicon:
http-server-header:
http-title:
nbstat:
smb2-time:
smtp-commands:
ssl-date:
```

# Nmap XML host structure

```
<host>
    <address addr="10.0.10.188" addrtype="ipv4">
    <ports>
        <port protocol="tcp" portid="22">
            <state state="open" reason="syn-ack">
            <service name="ssh" product="OpenSSH">
        </port>
        <port protocol="tcp" portid="80">
            <state state="open" reason="syn-ack">
            <service name="http" product="Apache httpd">
        </port>
    </ports>
</host>
```

# 使用Ruby解析XML输出

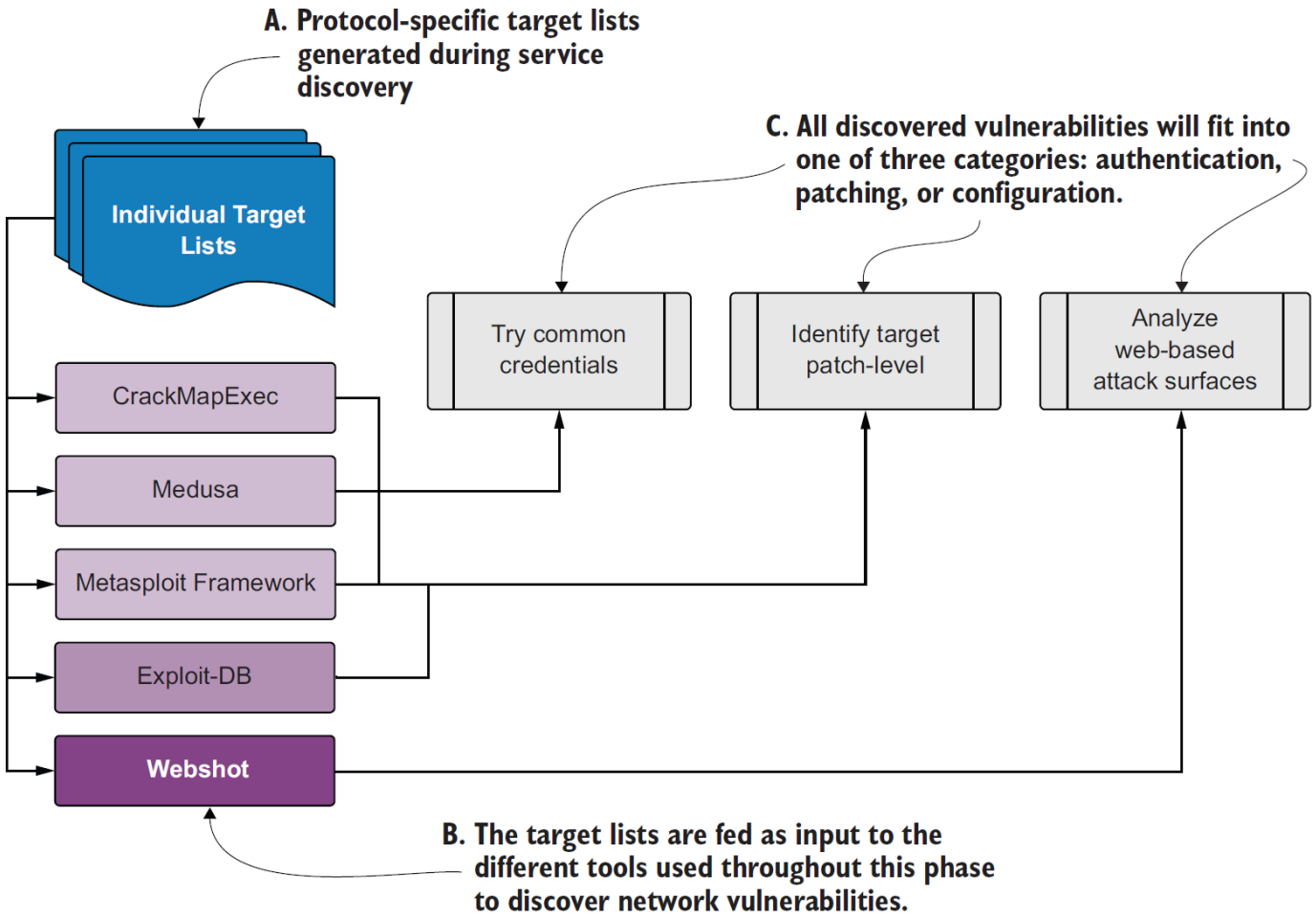
```
# git clone https://github.com/R3dy/parsenmap.git
# cd parsenmap
# bundle install
# ./parsenmap.rb full-sweep.xml
```

```
10.0.2.7        21      ftp     vsftpd  2.3.4
10.0.2.7        22      ssh     OpenSSH 4.7p1 Debian 8ubuntu1   protocol 2.0
10.0.2.7        23      telnet  Linux telnetd
10.0.2.7        25      smtp    Postfix smtpd
10.0.2.7        53      domain  ISC BIND        9.4.2
10.0.2.7        80      http    Apache httpd    2.2.8   (Ubuntu) DAV/2
10.0.2.7        111     rpcbind         2       RPC #100000
10.0.2.7        139     netbios-ssn     Samba smbd      3.X - 4.X       workgroup: WORKGROUP
10.0.2.7        445     netbios-ssn     Samba smbd      3.0.20-Debian   workgroup: WORKGROUP
10.0.2.7        512     exec    netkit-rsh rexecd
10.0.2.7        513     login
10.0.2.7        514     tcpwrapped
10.0.2.7        1099    java-rmi        GNU Classpath grmiregistry
10.0.2.7        1524    bindshell       Bash shell              **BACKDOOR**; root shell
10.0.2.7        2049    nfs             2-4     RPC #100003
10.0.2.7        2121    ftp     ProFTPD 1.3.1
10.0.2.7        3306    mysql   MySQL   5.0.51a-3ubuntu5
10.0.2.7        3632    distccd distccd v1      (GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4)
10.0.2.7        5432    postgresql      PostgreSQL DB   8.3.0 - 8.3.7
10.0.2.7        5900    vnc     VNC             protocol 3.3
10.0.2.7        6000    X11                     access denied
10.0.2.7        6667    irc     UnrealIRCd
10.0.2.7        6697    irc     UnrealIRCd
10.0.2.7        8009    ajp13   Apache Jserv            Protocol v1.3
10.0.2.7        8180    http    Apache Tomcat/Coyote JSP engine 1.1
10.0.2.7        8787    drb     Ruby DRb RMI            Ruby 1.8; path /usr/lib/ruby/1.8/drb
10.0.2.7        48115   java-rmi        GNU Classpath grmiregistry
10.0.2.7        50071   status          1       RPC #100024
10.0.2.7        50283   nlockmgr                1-4     RPC #100021
10.0.2.7        55095   mountd          1-3     RPC #100005
sudo nmap -sS -p 21,22,23,25,53,80,111,139,445,512,513,514,1099,1524,2049,2121,3306,3632,5432,5900,6000,6
667,6697,8009,8180,8787,48115,50071,50283,55095 -sV -A -vv -oA enumeration -iL ../ranges.txt
```

# Protocol-specific target lists

| Filename | Associated protocol | Associated ports |
|---|---|---|
| discovery/hosts/web.txt | http/https | 80,443,8080 |
| discovery/hosts/windows.txt | microsoft-ds | 139,445 |
| discovery/hosts/mssql.txt | ms-sql-s | 1,433 |
| discovery/hosts/mysql.txt | mysql | 3,306 |
| discovery/hosts/vnc.txt | vnc | 5800,5900 |

# 漏洞发现


A. Protocol-specific target lists generated during service discovery
Individual Target Lists
C. All discovered vulnerabilities will fit into one of three categories: authentication, patching, or configuration.
Try common credentials
Identify target patch-level
Analyze web-based attack surfaces
CrackMapExec
Medusa
Metasploit Framework
Exploit-DB
Webshot
B. The target lists are fed as input to the different tools used throughout this phase to discover network vulnerabilities.

# Following the path of least resistance

- we always want to look for the path of least resistance.
- These easy-to-spot vectors are sometimes referred to as low-hanging-fruit (**LHF**) vulnerabilities
- When targeting LHF vulnerabilities, we can avoid making too much noise on the network

# Discovering patching vulnerabilities

Discovering patching vulnerabilities is as straightforward as identifying exactly which version of a particular software your target is running and then comparing that version to the latest stable release available from the software vendor.