Identifying Services v0.2

Table of Contents

1 - Summary

The guide explains how to identify services. The tools and methods used in this guide have proven helpful to me when trying to identify what services are running on a system. This paper is aimed toward people that perform penetration tests and vulnerability assessments. Keep in mind this guide shows only one method of identifying services. For security-minded individuals out there reading this guide, feel free to contact me with your preferred method(s) of identifying services.

2 – Match open ports to services

The first step is to match each open port to a service and protocol. `Amap` will use its application map mode to try and identify the service. The application map mode works by connecting to the port and sending trigger packets that try to elicit responses. `Amap` then compares the responses to a list and prints the best match. The triggers and responses can be found in appdefs.trig and appdefs.resp. Here is the command to match each open TCP port to a service.

```
# amap -1 10.10.1.2 22 25 80
amap v4.5 (www.thc.org) started at 2004-02-13 23:15:39 - APPLICATION MAP mode

Protocol on 10.10.1.2:22/tcp matches ssh
Protocol on 10.10.1.2:22/tcp matches ssh-openssh
Protocol on 10.10.1.2:80/tcp matches http
Protocol on 10.10.1.2:25/tcp matches smtp

Unidentified ports: none.

amap v4.5 finished at 2004-02-13 23:15:40
```

The only difference when matching each UDP port to a service is the -u option.

```
# amap -u -1 10.10.1.2 53
amap v4.5 (www.thc.org) started at 2004-02-13 23:16:15 - APPLICATION MAP mode

Protocol on 10.10.1.2:53/udp matches dns-djb
Protocol on 10.10.1.2:53/udp matches dns
Protocol on 10.10.1.2:53/udp matches dns-bind9

Unidentified ports: none.

amap v4.5 finished at 2004-02-13 23:16:16
```

3 – Identify the service application

Amap and Nmap will be used to identify the application behind the services as well as the application patch levels. Identifying the service applications is done best by using the following methods: application mapping and banner grabbing.

3.1 – Application Mapping

Application mapping or version detection, as called in Nmap, works by connecting to the port(s) and sending trigger packets. Many of the service applications will respond to the trigger packets. Amap and Nmap look up the responses in a list and print out the best matches. Two tools will be used for application mapping to better compare results. Amap will be shown first. Here are the commands for TCP services.

```
# amap -1 -b 10.10.1.2 22 25 80
amap v4.5 (www.thc.org) started at 2004-02-13 23:16:58 - APPLICATION MAP mode

Protocol on 10.10.1.2:22/tcp matches ssh - banner: SSH-1.99-OpenSSH_3.1p1\n
Protocol on 10.10.1.2:22/tcp matches ssh-openssh - banner: SSH-1.99-OpenSSH_3.1p1\n
Protocol on 10.10.1.2:80/tcp matches http - banner: HTTP/1.1 200 OK\r\nDate Sat, 14 Feb
2004 051659 GMT\r\nServer Apache/2.0.40 (Red Hat Linux)\r\nAccept-Ranges
bytes\r\nConnection close\r\nContent-Type text/html\r\n\r\n<!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 3.2 Final//EN">\n<!-- $Id index.shtml,v 1.220
Protocol on 10.10.1.2:25/tcp matches smtp - banner: 220 zeus.test.com ESMTP Sendmail
8.11.6/8.11.6; Fri, 13 Feb 2004 211659 -0800\r\n

Unidentified ports: none.

amap v4.5 finished at 2004-02-13 23:16:58
```

Here are the commands for mapping UDP services with Amap. The only difference here is the -u option.

```
# amap -u -1 -b 10.10.1.2 53
amap v4.5 (www.thc.org) started at 2004-02-13 23:17:52 - APPLICATION MAP mode

Protocol on 10.10.1.2:53/udp matches dns-djb - banner: y\r
CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA!vE\fROOT-SERVERSNETvF?vG?vH?vI?vJ?vK?vL?vM?vA?vB?vC?vD?
Protocol on 10.10.1.2:53/udp matches dns - banner:
Protocol on 10.10.1.2:53/udp matches dns-ms - banner:
Protocol on 10.10.1.2:53/udp matches dns-bind9 - banner: versionbind\f9.2.1

Unidentified ports: none.

amap v4.5 finished at 2004-02-13 23:17:52
```

Now, Nmap will be used to do the same thing. Here are the commands for TCP services.

```
# nmap -sS -sV -P0 -v -p 22,25,80 -n 10.10.1.2

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-02-13 23:21 CST
Host 10.10.1.2 appears to be up ... good.
Initiating SYN Stealth Scan against 10.10.1.2 at 23:21
Adding open port 22/tcp
Adding open port 80/tcp
Adding open port 25/tcp
The SYN Stealth Scan took 0 seconds to scan 3 ports.
Initiating service scan against 3 services on 1 host at 23:21
The service scan took 5 seconds to scan 3 services on 1 host.
Interesting ports on 10.10.1.2:
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 3.1p1 (protocol 1.99)
25/tcp open  smtp    Sendmail 8.11.6/8.11.6
80/tcp open  http    Apache httpd 2.0.40 ((Red Hat Linux))

Nmap run completed -- 1 IP address (1 host up) scanned in 5.254 seconds
```

Here are the commands for mapping UDP services with Nmap. The only difference here is the -sU option.

```
# nmap -sU -sV -P0 -v -p 53 -n 10.10.1.2

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-02-13 23:22 CST
Host 10.10.1.2 appears to be up ... good.
Initiating UDP Scan against 10.10.1.2 at 23:22
The UDP Scan took 12 seconds to scan 1 ports.
Adding open port 53/udp
Initiating service scan against 1 service on 1 host at 23:22
The service scan took 1 second to scan 1 service on 1 host.
Interesting ports on 10.10.1.2:
PORT   STATE SERVICE VERSION
53/udp open  domain  ISC Bind 9.2.1

Nmap run completed -- 1 IP address (1 host up) scanned in 12.112 seconds
```

3.2 – Banner Grabbing

Banner grabbing works simply by connecting to the port(s) and grabbing the banners. Amap will be used here. Here are the commands for TCP services.

```
# amap -B -1 -b 10.10.1.2 22 25 80
amap v4.5 (www.thc.org) started at 2004-02-13 23:29:13 - BANNER GRAB mode

Banner on 10.10.1.2:22/tcp : SSH-1.99-OpenSSH_3.1p1\n
Banner on 10.10.1.2:25/tcp : 220 zeus.test.com ESMTP Sendmail 8.11.6/8.11.6; Fri, 13 Feb
2004 212918 -0800\r\n

amap v4.5 finished at 2004-02-13 23:29:25
```

Here are the commands for UDP services with Amap. The only difference here is the -u option.

```
# amap -B -u -1 -b 10.10.1.2 53
amap v4.5 (www.thc.org) started at 2004-02-13 23:31:14 - BANNER GRAB mode


amap v4.5 finished at 2004-02-13 23:31:20
```

4 – Verify the services

The last step is verifying the application to the system. Application version information also will be verified. Netcat will be used to do these tasks. Verification will work by connecting to the application on the port and listening for some data. The data is normally an application banner specifying the application along with its version number. Here is the command for TCP-based applications. When connecting to port 80 I issued the GET / HTTP command and then pressed ENTER twice.

```
# nc -n -v 10.10.1.2 22
(UNKNOWN) [10.10.1.2] 22 (?) open
SSH-1.99-OpenSSH_3.1p1
^C punt!
# nc -n -v 10.10.1.2 25
(UNKNOWN) [10.10.1.2] 25 (?) open
220 zeus.test.com ESMTP Sendmail 8.11.6/8.11.6; Fri, 13 Feb 2004 21:32:37 -0800
^C punt!
# nc -n -v 10.10.1.2 80
(UNKNOWN) [10.10.1.2] 80 (?) open
GET / HTTP

HTTP/1.1 200 OK
Date: Sat, 14 Feb 2004 05:51:17 GMT
Server: Apache/2.0.40 (Red Hat Linux)
Accept-Ranges: bytes
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

The only difference with the UDP-based applications is the -u option.

```
earth# nc -u -n -v 10.10.1.2 53
(UNKNOWN) [10.10.1.2] 53 (?) open
^C punt!
```

5 – Links

Amap - http://www.thc.org/releases.php

Netcat - http://www.atstake.com/research/tools/network_utilities/

Nmap - http://www.insecure.org/nmap/