

[REVS] Port 0 OS Fingerprinting

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-07/0112.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 07/28/03

To: list@securiteam.com

Date: 28 Jul 2003 15:33:32 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

Get Thawte's New Step-by-Step SSL Guide for Apache.

<http://ad.doubleclick.net/clk;5903117;8265118;i>

Port 0 OS Fingerprinting

SUMMARY

The following article describes a new method of fingerprinting hosts, and how to block such fingerprinting attempts.

DETAILS

Introduction:

There are 65536 TCP / UDP ports available to any normal TCP/IP stack. The range is from 0 to 65535, which is then split into multiple groups. For example 0 to 1024 is known as the reserved port range (traditionally only root can assign programs to ports in this range) and the ephemeral port range from 1025 to 65535. The ephemeral port range can also be split into two groups known as high and low port ranges. These two groups are set by the OS, but can normally be tweaked by changing specific options within the kernel.

Port 0's Normal usage

As many of you programmers will know, when you specify the source port of 0 when you connect to a host, the OS automatically reassigns the port number to high numbered ephemeral port. The same happens if you try to bind a listening socket to port 0.

The code below forces the OS to change the listening source port (`my_addr.sin_port = 0`) to another random ephemeral port.

Securiteam: [REVS] Port 0 OS Fingerprinting

```
//probably ripped from beej's guide to network programming
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <errno.h>
#define BACKLOG 1 // how many pending connections queue will hold

void main()
{
    int sockfd, new_fd; // listen on sock_fd, new connection on new_fd
    struct sockaddr_in my_addr; // my address information
    struct sockaddr_in their_addr; // connector's address information
    int sin_size;

    sockfd = socket(AF_INET, SOCK_STREAM, 0); //oops no checking

    my_addr.sin_family = AF_INET; // host byte order
    my_addr.sin_port = 0; // port 0 is reassigned
    my_addr.sin_addr.s_addr = INADDR_ANY; // auto-fill with my IP
    memset(&(my_addr.sin_zero), '\0', 8); // zero the rest of the
    struct

    if((bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct
sockaddr))) !=0){
        printf("oops: bind error as %s\n",strerror(errno));
        exit(1);
    }

    //no checking oops
    listen(sockfd, BACKLOG);

    sin_size = sizeof(struct sockaddr_in);
    new_fd = accept(sockfd, (struct sockaddr *)&their_addr,
&sin_size);
    printf("woop woop got a connection\n");
}
```

Port 0 OS Fingerprinting

As port 0 is reserved for special use as stated in RFC 1700. Coupled with the fact that this port number is reassigned by the OS, no traffic should flow over the internet use this port. As the specifics are not clear different OS's have, different ways of handling traffic using port 0 thus they can be fingerprinted.

Port 0 fingerprinting consists of seven tests. The tests are labeled P1 – P7 below.

- P1: send tcp packet from source port 0 to port 0
- P2: send tcp packet from source port X to port 0
- P3: send tcp packet from source port 0 to open port

Securiteam: [REVS] Port 0 OS Fingerprinting

P4: send tcp packet from source port 0 to closed port
P5: send udp packet from source port 0 to port 0
P6: send udp packet from source port 53 to port 0
P7: send udp packet from source port 0 to closed port

Port X in test P2 is any port not equal to 0. Port 53 is used in test P6 as it is most likely to bypass a firewall configuration.

The standard reply expected to P1, P2 and P4 should be a RST packet as the port should be closed.

The standard reply to P3 should be SYN ACK as the port is open and port 0 is a valid port as described above.

The standard reply to P5, P6 and P7 should all be ICMP port unreachable as UDP port 0 / closed port should not have a program listening on it.

Although port 0 is a valid port number various OS's handle port 0 differently.

Results

Below are a few example fingerprints. The entire list can be found at the end of the paper.

Fingerprint OpenBSD 3.2/3.3

P1(Resp=Y%Flags=AR)
P2(Resp=Y%Flags=AR)
P3(Resp=N)
P4(Resp=Y%Flags=AR)
P5(Resp=N)
P6(Resp=N)
P7(Resp=Y)

Notice that OpenBSD has a cool feature / bug whereby it does not allow incoming connections from source port 0 (test P3)

Fingerprint Linux

P1(Resp=Y%Flags=AR)
P2(Resp=Y%Flags=AR)
P3(Resp=Y%Flags=AS)
P4(Resp=Y%Flags=AR)
P5(Resp=Y)
P6(Resp=Y)
P7(Resp=Y)

Unfortunately, both MS Windows 2000 and Linux have the same port 0 fingerprint, replying to all 7 tests.

Recommendations

Although port 0 is a valid TCP / UDP port number, it is highly recommend that one should block any traffic using this port at your firewall. No

Securiteam: [REVS] Port 0 OS Fingerprinting

program should be listening on port 0 and no program should connect from port 0 thus, it should be blocked.

Port 0 fingerprinting can be tested using the gobbler-2.0.1-alpha available from <http://www.networkpenetration.com> or <http://gobbler.sourceforge.net>

Firwall Configurations

Untested IPTABLES Rules for port 0 fingerprint blocking

```
$IPTABLES -A DROP -p tcp --dport 0
```

```
$IPTABLES -A DROP -p udp --dport 0
```

```
$IPTABLES -A DROP -p tcp --sport 0
```

```
$IPTABLES -A DROP -p udp --sport 0
```

OpenBSD's Packet Filter Rules for port 0 fingerprint blocking

```
block in log quick on $EXT inet proto tcp from any port 0 to any
```

```
block in log quick on $EXT inet proto udp from any port 0 to any
```

```
block in log quick on $EXT inet proto tcp from any to any port 0
```

```
block in log quick on $EXT inet proto udp from any to any port 0
```

```
block out log quick on $EXT inet proto tcp from any port 0 to any
```

```
block out log quick on $EXT inet proto udp from any port 0 to any
```

```
block out log quick on $EXT inet proto tcp from any to any port 0
```

```
block out log quick on $EXT inet proto udp from any to any port 0
```

List of Port 0 Fingerprints

Fingerprint Mac OSX

```
P1(Resp=Y%Flags=AR)
```

```
P2(Resp=Y%Flags=AR)
```

```
P3(Resp=Y%Flags=AS)
```

```
P4(Resp=Y%Flags=AR)
```

```
P5(Resp=N)
```

```
P6(Resp=N)
```

```
P7(Resp=Y)
```

Fingerprint Gobbler 2.0 Alpha

```
P1(Resp=Y%Flags=AR)
```

```
P2(Resp=Y%Flags=AR)
```

```
P3(Resp=Y%Flags=AS)
```

```
P4(Resp=Y%Flags=AR)
```

```
P5(Resp=N)
```

```
P6(Resp=N)
```

```
P7(Resp=Y)
```

Fingerprint Linux

```
P1(Resp=Y%Flags=AR)
```

```
P2(Resp=Y%Flags=AR)
```

```
P3(Resp=Y%Flags=AS)
```

```
P4(Resp=Y%Flags=AR)
```

```
P5(Resp=Y)
```

```
P6(Resp=Y)
```

```
P7(Resp=Y)
```

Securiteam: [REVS] Port 0 OS Fingerprinting

Fingerprint MS Windows 2000

P1(Resp=Y%Flags=AR)
P2(Resp=Y%Flags=AR)
P3(Resp=Y%Flags=AS)
P4(Resp=Y%Flags=AR)
P5(Resp=Y)
P6(Resp=Y)
P7(Resp=Y)

Fingerprint VMS on Alpha

P1(Resp=Y%Flags=AR)
P2(Resp=Y%Flags=AR)
P3(Resp=Y%Flags=AS)
P4(Resp=Y%Flags=AR)
P5(Resp=Y)
P6(Resp=Y)
P7(Resp=Y)

Fingerprint OpenBSD 3.2 or 3.3

P1(Resp=Y%Flags=AR)
P2(Resp=Y%Flags=AR)
P3(Resp=N)
P4(Resp=Y%Flags=AR)
P5(Resp=N)
P6(Resp=N)
P7(Resp=Y)

Fingerprint SunOS 5.6 (can someone confirm please)

P1(Resp=N)
P2(Resp=N)
P3(Resp=Y%Flags=AS)
P4(Resp=Y%Flags=AR)
P5(Resp=N)
P6(Resp=N)
P7(Resp=Y)

Fingerprint MS NT Server 4 (Service pack ?) with checkpoint ?

P1(Resp=N)
P2(Resp=N)
P3(Resp=Y%Flags=AS)
P4(Resp=Y%Flags=AR)
P5(Resp=N)
P6(Resp=N)
P7(Resp=Y)

ADDITIONAL INFORMATION

The information has been provided by <<http://www.NetworkPenetration.com>>
Ste Jones.

=====

Securiteam: [REVS] Port 0 OS Fingerprinting

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.