Global Information Assurance Certification Paper

Unidirectional Networking

GIAC Security Essential Certification Practical Assignment Version 1.4b

Jason Westmacott

0 Abstract

One of the golden rules of network security is to only enable functionality that you need. Every extra service you enable provides an attacker with another potential avenue of attack.

There are many situations in which a computer does not require a bidirectional network connection to perform its function. If a computer only needs to receive data, then it makes good security sense to disable its ability to transmit data altogether, and vice-versa.

Section 1 of this paper will begin by describing what a unidirectional network connection is and how data is sent over one.

Section 2 will discuss how having a unidirectional network connection affects the integrity, availability, and the confidentiality of a computer on a network.

Section 3 will describe some problems encountered when implementing a unidirectional network connection and suggest ways in which they may be overcome.

Section 4 will demonstrate sending data over a unidirectional fibre connection using netcat, and briefly mention some practical applications in which a unidirectional connection may be used.

Section 5 will look at some software provided by Tenix Datagate for connecting two networks over a unidirectional link enforced by a data diode.

This paper focuses specifically on unidirectional communications over a fibre network using UDP, although it is important to note that the principles discussed here are not exclusive to fibre networks or to UDP.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

1 Unidirectional Network Connections

The Merriam-Webster dictionary (Ref. 1) defines 'unidirectional' as:
1. Involving, functioning, moving, or responsive in a single direction.
2. Not subject to change or reversal of direction.

A unidirectional network connection is a connection on which a device may only transmit data or only receive data, but not both. That is, a source can transmit data to one or many destinations, but the destination(s) cannot transmit data back to the source because it is unable to receive.

The idea of unidirectional networking is not new. In April 1997 the UniDirectional Link Routing (UDLR) group was formed to *"provide a solution for the support of unidirectional links in the Internet."* (Ref. 10). In March 2001, the UDLR issued RFC 3077 describing a mechanism to emulate full bidirectional connectivity between nodes that are directly connected by a unidirectional link. (Ref. 6)

1.1 Creating a Unidirectional Link over Optical Fibre

Two fibre optic cables are used to create a fibre optic network connection, one for the transmit signal and one for the receive signal. This can be considered as either a single bidirectional connection, or as two unidirectional connections, as shown below.

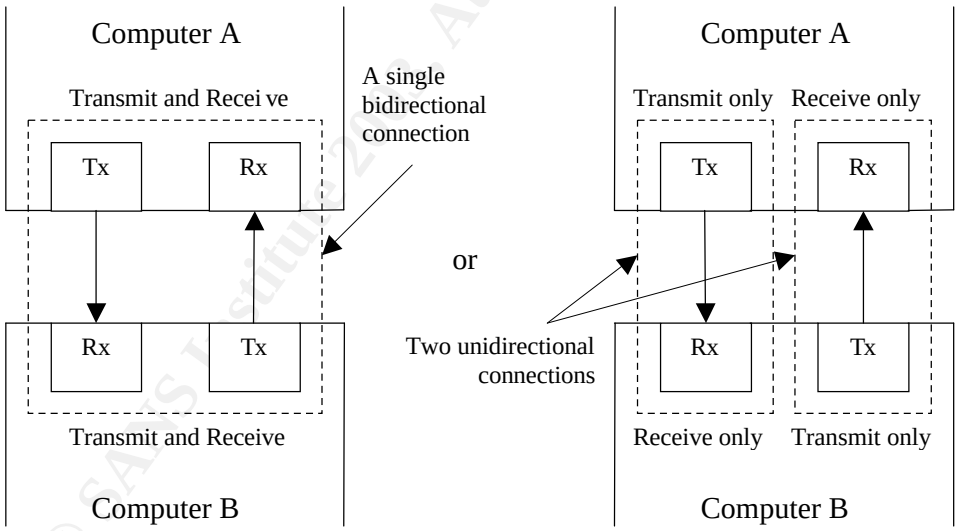

Figure 1 - Transmit and receive pairs

By disconnecting one of the transmit/receive pairs, the computers are left with a single connection on which one computer may only transmit and the other may only receive. This is an example of a unidirectional network connection.

1.2 Communicating Over a Unidirectional Link

To communicate over a one way network connection a connectionless protocol must be used. A connectionless protocol is a protocol in which there is no persistent logical connection established between the points that are communicating, and each unit of data received is treated as being independent.

Internet Protocol (IP) itself is a connectionless protocol, while Transmission Control Protocol (TCP) is a connection oriented protocol over IP. User Datagram Protocol (UDP) is a connectionless protocol over IP, and is well suited to implementing unidirectional communications.

UDP is often referred to as 'send and pray'. The transmitting computer has no idea whether or not the destination computer successfully received its transmission. It is worth noting that UDP is not always implemented in a completely unidirectional manner. The IP stack of some systems, including Linux and Solaris, will return a "protocol unreachable" message (ICMP 3/2) if no handler for the UDP protocol is present. (Ref. 14)

1.3 Enforcing a Unidirectional Link

A unidirectional connection can be enforced using a data diode, a hardware device which makes it physically impossible to transmit data in a certain direction.

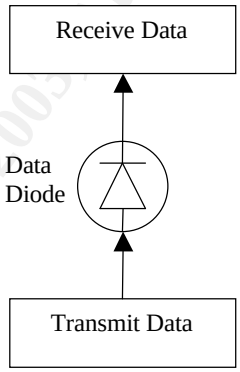

Figure 2 – Enforcing unidirectional communicati on using a data diode

The transmitting computer connects to the input of the data diode, and the receiving computer connects to the output of the data diode. The diode guarantees data cannot flow in the opposite direction.

2 Integrity, Availability, and Confidentiality

This section will briefly describe how the principals of integrity, availability, and confidentiality relate to a computer with a unidirectional network connection, specifically from the perspective of a remote attacker.

2.1 Integrity

The integrity of a computer relates to how trustworthy the information on it is. If a computer were tampered with such that it contained false information, it would be considered a breach of integrity.

2.1.1 Transmit only

If a computer cannot physically receive information it is impossible to remotely execute instructions on it. From a remote perspective there is no way to alter the data on it without physical access, so the integrity of the machine can be guaranteed.

2.1.2 Receive only

With detailed knowledge of the receiving computer, it may be possible to exploit a remote vulnerability and execute code. Although, this would be close to impossible for an attacker with no prior knowledge of the system. There is no way to perform reconnaissance to determine which services are running, or detect the operating system being used. Also, there is no way to determine if the attack was successful without physical access to the machine, as the machine is unable to transmit data back to the remote attacker. Theoretically, it is possible to remotely modify data and compromise the integrity of the machine, so the integrity of a 'receive only' machine cannot be guaranteed.

2.2 Availability

The availability of a computer relates to whether or not that computer is able to communicate over a network. If a computer cannot communicate it is said to be unavailable.

2.2.1 Transmit only

If a machine cannot receive data it cannot be targeted by availability attacks such as denial of service. As always, it is possible to attack upstream by targeting routers and other computers, but there is no way for a remote attacker to directly compromise the availability of the transmitting computer. From a remote perspective, availability can be guaranteed.

2.2.2 Receive only

If a machine can receive data it is susceptible to availability attacks such as denial of service, and availability cannot be guaranteed.

2.3 Confidentiality

The confidentiality of a computer relates to how private the information it holds is. If a computer is broadcasting passwords for the entire world to see, it is considered a breach of confidentiality.

2.3.1 Transmit only

A computer that can only transmit has no means of knowing whether the information it is transmitting has reached the intended destination. Also, an attacker can sniff traffic and compromise the confidentiality of the communications, hence confidentiality cannot be guaranteed.

2.3.2 Receive only

A computer that cannot transmit data over a network can be considered completely confidential from a remote perspective. If an attacker cannot see any information coming from the computer, there is no way to deduce any information about it. There is no way of knowing if the computer is even turned on. The only way to view information stored by such a computer is with physical access. From a remote perspective confidentiality is guaranteed.

3 Implementation

Several problems arise when a computer can only communicate in one direction. This section will describe how to overcome some of the more common problems that arise.

3.1 Domain Name Resolution

A computer that can only transmit or only receive cannot resolve domain names. It becomes necessary to use IP addresses instead of hostnames, or provide appropriate configuration such as entries in /etc/hosts.

3.2 Carrier Signal

When using 100base-FX fibre Network Interface Cards (NICs), and possibly others, a carrier signal is provided by the transmit line of one NIC to the receive line of another NIC. A fibre NIC will not transmit data unless there is a carrier signal on its receive line.

To allow a fibre NIC to transmit, a second fibre NIC can be installed with its transmit line connected directly to the receive line of the transmitting NIC, as shown in Figure 3. It is also possible to use a media converter to provide a carrier signal.

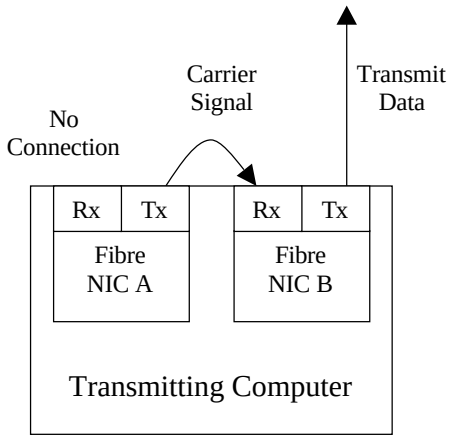

Figure 3 - Using a second NIC to provide a carrier signal

3.3 Error Control

It is impossible to guarantee error free delivery of data over a unidirectional link. Even if the receiving computer is able to deduce that data is corrupt or missing, there is no way for it to inform the source.

Data loss can be minimised by the use of redundancy, either by sending the same data more than once, or by including additional information such as Forward Error Correction (FEC) codes to reconstruct lost data if enough good data has been received.

3.4 Address Resolution Protocol

Generally, for a computer to receive a packet it must be addressed to the Media Access Control (MAC or ethernet) address of the receiving NIC. A MAC address is the unique identifier given to every NIC produced, and is used by the datalink layer (layer 2 of the Open Systems Interconnection model) to uniquely identify each NIC on a network.

Address Resolution Protocol (ARP) is used by devices such as switches, routers, or other computers to make the association between a MAC address and an IP address. A computer that can only communicate in one direction cannot use ARP to discover MAC addresses. It will either not be able to hear ARP requests at all, or be able to hear them but not respond.

Some ways to overcome this include:

1. Manually configuring the source computers ARP table with the MAC address of the destination device.
2. Put the NIC that is receiving data into promiscuous mode, such that it receives all packets regardless of address.
3. Ensure the source and destination are on the same subnet and use the network broadcast address, such that all computers within the broadcast range will read the packet.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

4 Example using netcat

The following section will demonstrate unidirectional network communications over a direct 100base-Fx fibre connection using netcat, and briefly discuss some applications in which a one way connection can be used to improve security.

4.1 Example using Netcat

Netcat (Ref. 12), a utility which reads and writes data across network connections, can be used to transfer data in a unidirectional manner over UDP. The following example sends the message 'Hello' from a source machine (10.0.1.1) to a destination machine (10.0.1.2) using RedHat Linux 7.1 over a direct one way fibre connection.

To achieve this, the destination machine is set up to listen for UDP communications on port 20000, and the source machine is used to send the text 'Hello' to the destination.

On the destination machine, netcat is set up to listen for UDP on port 20000:

```
./nc -u -l -p 20000
```

From the source machine, netcat is used to send the message:

```
echo Hello | ./nc -u -n 10.0.1.2 20000
```

On the source machine, tcpdump shows three ARP requests in an attempt to resolve the MAC address of 10.0.1.2:

```
Kernel filter, protocol ALL, TURBO mode (575 frames), datagram packet
socket
tcpdump: listening on eth1
12:55:57.659425    >    arp    who-has    10.0.1.2    tell    10.0.1.1
(0:4:76:yy:yy:yy)
12:55:58.655796    >    arp    who-has    10.0.1.2    tell    10.0.1.1
(0:4:76:yy:yy:yy)
12:55:59.655790    >    arp    who-has    10.0.1.2    tell    10.0.1.1
(0:4:76:yy:yy:yy)

3 packets received by filter
```

While on the destination machine, tcpdump shows three attempts to answer the ARP requests:

```
Kernel filter, protocol ALL, TURBO mode (575 frames), datagram packet
socket
tcpdump: listening on eth1
12:56:52.992160 B arp who-has hs_dst tell 10.0.1.1
12:56:52.992257   >   arp   reply   hs_dst   (0:4:76:xx:xx:x x)   is-at
0:4:76:xx:xx:xx (0:4:76:yy:yy:yy)
12:56:53.988472 B arp who-has hs_dst tell 10.0.1.1
12:56:53.988506   >   arp   reply   hs_dst   (0:4:76:xx:xx:xx)   is-at
0:4:76:xx:xx:xx (0:4:76:yy:yy:yy)
12:56:54.988411 B arp who-has hs_dst tell 10.0.1.1
12:56:54.988443   >   arp   reply   hs_dst   (0:4:76:xx:xx:xx)   is-at
0:4:76:xx:xx:xx (0:4:76:yy:yy:yy)
```


© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
6 packets received by filter
```

As can be seen from the tcpdump output, the source was unable to determine the MAC address of 10.0.1.2. An ARP request was sent, but the destination machine was unable to reply so the message was not sent. Normally, when a computer cannot ARP it will send to the appropriate gateway if one is configured. In this case, no appropriate gateway was configured so the message was not sent.

To overcome this, a manual entry is made to the ARP table of the source computer to map the IP 10.0.1.2 to the MAC address of its NIC:

```
arp -s 10.0.1.2 00:04:76:xx:xx:xx
```

Attempting again, tcpdump on the source machine shows a single packet was sent to 10.0.1.2:

```
Kernel filter, protocol ALL, TURBO mode (575 frames), datagram packet
socket
tcpdump: listening on eth1
12:58:26.566846 > 10.0.1.1.32800 > 10.0.1.2.20000: udp 6 (DF)

1 packets received by filter
```

On the destination machine, tcpdump shows the packet was successfully received:

```
Kernel filter, protocol ALL, TURBO mode (575 frames), datagram packet
socket
tcpdump: listening on eth1
12:59:21.892010 < 10.0.1.1.32800 > hs_dst.20000: udp 6 (DF)

1 packets received by filter
```

The destination machine displays the following message:

```
[root@hs_dst /root]# ./nc -u -l -p 20000
Hello
```

As can be seen, a message was successfully sent from the source (10.0.1.1) to the destination (10.0.1.2) over a unidirectional link. Note that at no stage were packets sent from 10.0.1.2 to 10.0.1.1, even though the destination did attempt to reply to ARP requests.

4.2 Practical Uses

There are many applications in which a unidirectional network connection may be used to improve the security of a computer without losing functionality.

Syslog is a commonly used logging daemon that is capable of receiving log information over a network (using UDP 514 by default). It is not uncommon for an organisation to run a dedicated syslog server to collect and store logs from many machines, and it is important that the syslog server is secure. A syslog server can be implemented with a receive only connection to make it physically incapable of

transmitting data to a remote attacker, yet still able to receive log information over a network.

Many organisations run a dedicated Network Intrusion Detection System (NIDS) to monitor their networks for suspicious behaviour. A receive only NIDS can monitor a network with little chance of being detected and/or compromised by an attacker.

A webcaster could use a transmit only connection to broadcast data to a network, such as a radio or video, without risk of being remotely compromised.

5 Connecting Networks

Using a unidirectional link, it is possible to join an insecure, or low side, network to a secure, or high side, network and guarantee the confidentiality of the secure network.

5.1 Tenix Datagate's Veto Software

Tenix Datagate provide a range of products designed for transferring data over a unidirectional link enforced by a data diode. (Ref. 9)

Unidirectional Network Bridge (UNB) software provides a *data pump* across the unidirectional link, while *data pump applications* plug in to the UNB to transfer data across it.

Examples of *data pump applications* include:

- Email Transport Application (ETA) - for email transfer from a low side network to a high side network.
- File Transport Application (FTA) - for file transfer from a low side network to a high side network.
- Data Forwarding Application (DFA) - for unidirectional TCP or UDP communications from a low side network to a high side network.
- Clipboard and File Transfer Application (CFT) - for transferring clipboard content and file data from a low side workstation to a high side workstation.

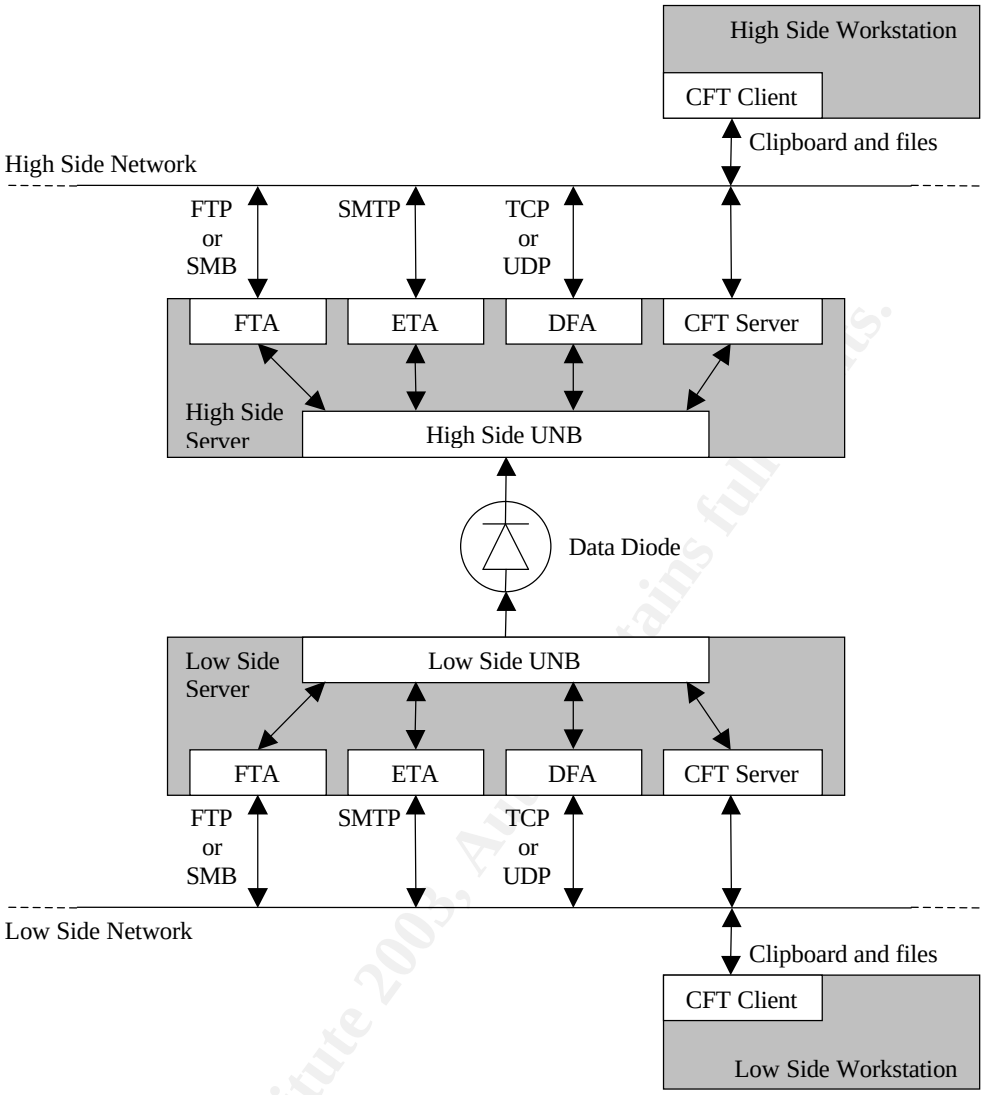

Figure 4 – Tenix Datagate's Veto Dat a Pump Applications (Ref. 9)

6 Conclusions

This paper has described what a unidirectional network connection is and how data can be sent over one. It has also discussed the implications a one way network connection has on the integrity, availability, and confidentiality of a computer.

An example using netcat was shown, demonstrating how in the absence of a suitable gateway a computer can be made to transmit over a unidirectional connection by adding an entry to the ARP table of the transmitting computer.

Some applications were discussed in which a one way connection can be used to improve the security of a computer without the loss of functionality. Finally, some software for connecting two networks over a unidirectional link was shown.

A unidirectional network connection can greatly enhance the security of a computer on a network. If a computer can only receive, or only transmit, the number of remote vulnerabilities it is susceptible to is significantly reduced. If a computer only needs to communicate in one direction, it is worth considering disabling the unnecessary direction all together.

7 References

1. The Merriam-Webster dictionary
   URL: http://www.m-w.com

2. RFC 768: User Datagram Protocol.
   URL: http://www.ietf.org/rfc/rfc0768.txt

3. RFC 791: Internet Protocol.
   URL: http://www.ietf.org/rfc/rfc0791.txt

4. RFC 793: Transmission Control Protocol.
   URL: http://www.ietf.org/rfc/rfc0793.txt

5. RFC 826: Address Resolution Protocol.
   URL: http://www.ietf.org/rfc/rfc0826.txt

6. RFC 3077: A Link-Layer Tunneling Mechanism for Unidirectional Links.
   URL: http://www.ietf.org/rfc/rfc3077.txt

7. RFC 3164: The BSD syslog Protocol
   URL: http://www.ietf.org/rfc/rfc3164.txt

8. RFC 3452: Forward Error Correction (FEC) Building Block
   URL: http://www.ietf.org/rfc/rfc0793.txt

9. Veto Uni-directional Network Bridge and Data Pump Applications White Paper
   URL: http://www.tenixdatagate.com/PDFLibrary/130.pdf

10. UniDirectional Link Routing Group
    URL: http://www.udcast.com/udlr/
    URL: http://www.ietf.org/html.charters/udlr-charter.html

11. Tenix Datagate.
    URL: http://www.tenixdatagate.com

12. Netcat.
    URL: http://www.atstake.com/research/tools/network_utilities/

13. Tcpdump.
    URL: http://www.tcpdump.org

14. Nmap Hackers: Protocol scan with nmap

http://lists.insecure.org/nmap-hackers/2000/Apr-Jun/0119.html

15. Open Systems Interconnection--Reference Model (OSI--RM)
URL: http://www.its.bldrdoc.gov/fs-1037/dir-025/_3680.htm