Talker Packet Sniffing using TCPDUMP
CS690 Java Talker
Jeffrey K. Brown

This is an analysis of the output of a program called TCPDUMP. TCPDUMP's purpose is to capture the packets passing a network interface that meet a certain criteria. It is often used to analyze network traffic, troubleshoot network problems and in some cases, used by malicious users and attackers to capture information about a computer system. I asked the TCPDUMP program to capture packets intended for port 2121 on my local network where I have a development version of NUTS-based Dark Ages Talker running. Through this experiment, I am backing up my assertion that many talkers send unencrypted conversation data across the Internet.

First, let me explain the environment in which I am performing this experiment. I have two computers participating, llama.universe.net running Windows 2000 and chimaera.universe.net running SuSE Linux. Chimaera will host the actual Talker server on port 2121 and permit connections. Additionally, I will connect to the Talker from Chimaera. Then, the user, Summoner, will connect to the Talker server from Llama using a popular Talker client, GMud. TCPDUMP will monitor the communication taking place from Chimaera. On the uncontrolled Internet, an attacker probably will not have direct access to one of the systems participating in a data exchange. However, once an attacker positions the monitoring equipment along the path between client and server, communication is monitored in exactly the same manner.

Next, let me explain how to use the TCPDUMP program. This is the syntax I used for this TCPDUMP command:

```
# tcpdump -i eth0 -s 0 -tlX port 2121 > tcpdump.log
```

Parts of the Command:

#	Root Shell
tcpdump	Call to the TCPDUMP program
-i eth0	We are asking TCPDUMP to listen on my Ethernet card.
-s 0	This is the "snaplen" value for TCPDUMP. In other words, this is used for the maximum number of bytes to capture for each packet. By default, TCPDUMP captures 68 bytes. To capture the entire conversation, I needed to get all the bytes in the packet, and using 0 for the length means "grab the whole thing"
-t	Do not print a timestamp. I wanted more room for the ASCII
l	Buffer stdout
X	Print the ASCII text in addition to the Hexadecimal numbers.
port 2121	Capture packets with the source or destination port of 2121
> tcpdump.log	Redirect all output to the log file, which I used to create this document.

Next, we're going to view a packet and dissect the parts, so the reader will understand what we are looking at when the experiment gets interesting. Below you will see a typical TCPDUMP packet. In this particular packet, nothing exciting is going on, so it will be a good packet to dissect.

```
llama.universe.net.1158 > chimaera.universe.net.2121: . ack 1400 win 64136 (DF)
0x0000   4500 0028 0f5d 4000 8006 67b2 c0a8 0139        E..(.]@...g....9
0x0010   c0a8 0137 0486 0849 62fc 0a7a 850e 6156        ...7...Ib..z..aV
0x0020   5010 fa88 d0e0 0000 0000 0000 0000             P.............
```

First, let's look at the header line below. We are going to pay attention to the host name and port, the direction of communication, and the destination host name and port. Other data is printed, but it is beyond the scope of this document.

```
llama.universe.net.1158 > chimaera.universe.net.2121: . ack 1400 win 64136 (DF)
```

Parts of the Header:

llama.universe.net	This is the source address or hostname. Since the computer doing the logging has the capability to look up the name, it prints the name instead of the address

.1158	This is the source port of the connection. This source port is usually defined by the computer's operating system.
>	This is a symbol showing the "direction" of the packet i.e. from source to destination.
chimaera.universe.net	This is the destination adddress or hostname.
.2121	This is the destination port of the connection. Since the Talker service runs on port 2121, connections to it must be destined for that service port.
everything else	While useful, this data is beyond the scope of this experiment, so disregard it.

Next, we will take a quick look at the payload section of the packet. There are three portions of the payload section that we will be interested in. The first column is the sequence of bytes in the payload. The Hex number shows the sequence of the first byte on that row. Next, we have the Hexadecimal representation of the packet payload. Finally, in the last column, we have the corresponding ASCII representation of the packet payload for the readers who are not as skilled at reading in Hex. The packet below does not send any data that is readable, but we will see some later.

```
Sequence         Hexadecimal Representation            ASCII Representation
 Number             of the Packet Payload               of Packet Payload
|-----|-|---------------------------------------|------|-----------------|
0x0000   4500 0028 0f5d 4000 8006 67b2 c0a8 0139        E..(.]@...g....9
0x0010   c0a8 0137 0486 0849 62fc 0a7a 850e 6156        ...7...Ib..z..aV
0x0020   5010 fa88 d0e0 0000 0000 0000 0000             P.............
```

Now we will get into the actual communication taking place. Prior to the interesting packet, I will leave my comments and explain what is taking place and why I consider it interesting.

We pick up the conversation after the Talker system has prompted the user for the username. The user types the username and transmits it from Llama to Chimaera. As you can see, the username is clearly displayed.

```
llama.universe.net.1158 > chimaera.universe.net.2121: P 1:10(9) ack 1400 win 64136
0x0000   4500 0031 0f5e 4000 8006 67a8 c0a8 0139        E..1.^@...g....9
0x0010   c0a8 0137 0486 0849 62fc 0a7a 850e 6156        ...7...Ib..z..aV
0x0020   5018 fa88 110c 0000 7375 6d6d 6f6e 6572        P.......summoner
0x0030   0a                                             .
chimaera.universe.net.2121 > llama.universe.net.1158: . ack 10 win 5840 (DF)
0x0000   4500 0028 7f0f 4000 4006 3800 c0a8 0137        E..(..@.@.8....7
0x0010   c0a8 0139 0849 0486 850e 6156 62fc 0a83        ...9.I....aVb...
0x0020   5010 16d0 b490 0000                            P.......
```

Next, the Talker asks the user to authenticate by providing a password.

```
chimaera.universe.net.2121 > llama.universe.net.1158: P 1400:1443(43) ack 10 win 5840
0x0000   4500 0053 7f10 4000 4006 37d4 c0a8 0137        E..S..@.@.7....7
0x0010   c0a8 0139 0849 0486 850e 6156 62fc 0a83        ...9.I....aVb...
0x0020   5018 16d0 9f10 0000 1b5b 3332 6d1b 5b31        P........[32m.[1
0x0030   6d4f 6b2c 2070 726f 7665 2074 6861 7420        mOk,.prove.that.
0x0040   6974 2069 7320 796f 753a 1b5b 306d 201b        it.is.you:.[0m..
0x0050   5b30 6d                                        [0m
llama.universe.net.1158 > chimaera.universe.net.2121: . ack 1443 win 64093 (DF)
0x0000   4500 0028 0f5f 4000 8006 67b0 c0a8 0139        E..(._@...g....9
0x0010   c0a8 0137 0486 0849 62fc 0a83 850e 6181        ...7...Ib.....a.
0x0020   5010 fa5d d0d7 0000 0000 0000 0000             P..]..........
```

The User, Summoner, types his Password.

```
llama.universe.net.1158 > chimaera.universe.net.2121: P 10:19(9) ack 1443 win 64093
0x0000   4500 0031 0f60 4000 8006 67a6 c0a8 0139        E..1.`@...g....9
0x0010   c0a8 0137 0486 0849 62fc 0a83 850e 6181        ...7...Ib.....a.
0x0020   5018 fa5d 252b 0000 6d65 6368 6d65 6368        P..]%+..mechmech
0x0030   0a                                             .
chimaera.universe.net.2121 > llama.universe.net.1158: P 1443:1475(32) ack 19 win 5840
0x0000   4500 0048 7f11 4000 4006 37de c0a8 0137        E..H..@.@.7....7
```

```
0x0010   c0a8 0139 0849 0486 850e 6181 62fc 0a8c        ...9.I....a.b...
0x0020   5018 16d0 6b1b 0000 1b5b 3333 6d1b 5b31        P...k....[33m.[1
0x0030   6d41 7574 6f20 5772 6170 2073 6574 1b5b        mAuto.Wrap.set.[
0x0040   306d 0a0d 1b5b 306d                            0m...[0m
llama.universe.net.1158 > chimaera.universe.net.2121: . ack 1475 win 65535 (DF)
0x0000   4500 0028 0f61 4000 8006 67ae c0a8 0139        E..(.a@...g....9
0x0010   c0a8 0137 0486 0849 62fc 0a8c 850e 61a1        ...7...Ib.....a.
0x0020   5010 ffff cb0c 0000 0000 0000 0000             P.............
```

Once authenticated, the system provides a welcome message to the user.

```
chimaera.universe.net.2121 > llama.universe.net.1158: P 1475:2761(1286) ack 19 win 5840
0x0000   4500 052e 7f12 4000 4006 32f7 c0a8 0137        E.....@.@.2....7
0x0010   c0a8 0139 0849 0486 850e 61a1 62fc 0a8c        ...9.I....a.b...
0x0020   5018 16d0 aebe 0000 0a0a 0a0d 5765 6c63        P...........Welc
0x0030   6f6d 6520 5072 696e 6365 2053 756d 6d6f        ome.Prince.Summo
0x0040   6e65 720a 0a0d 1b5b 306d 4c61 7374 206c        ner....[0mLast.l
0x0050   6f67 6765 6420 696e 206f 6e20 4672 6920        ogged.in.on.Fri.
0x0060   4f63 7420 3331 2032 313a 3339 3a33 3920        Oct.31.21:39:39.
0x0070   3230 3033 2066 726f 6d20 6c6c 616d 612e        2003.from.llama.
0x0080   756e 6976 6572 7365 2e6e 6574 0a0a 0d1b        universe.net....
```

Next, we will skip a few packets and jump ahead to some actual conversation
taking place.

Summoner, from Llama, sends a friendly greeting to the Jkb user on Chimaera.
The Talker system sends Summoner an acknowledgement by repeating what was sent.

```
llama.universe.net.1158 > chimaera.universe.net.2121: P 19:46(27) ack 2761 win 64249
0x0000   4500 0043 0f63 4000 8006 6791 c0a8 0139        E..C.c@...g....9
0x0010   c0a8 0137 0486 0849 62fc 0a8c 850e 66a7        ...7...Ib.....f.
0x0020   5018 faf9 9613 0000 6865 6c6c 6f21 2020        P.......hello!..
0x0030   486f 7720 6172 6520 796f 7520 746f 6461        How.are.you.toda
0x0040   793f 0a                                        y?.
chimaera.universe.net.2121 > llama.universe.net.1158: P 2761:2800(39) ack 46 win 5840
0x0000   4500 004f 7f13 4000 4006 37d5 c0a8 0137        E..O..@.@.7....7
0x0010   c0a8 0139 0849 0486 850e 66a7 62fc 0aa7        ...9.I....f.b...
0x0020   5018 16d0 5538 0000 596f 7520 6173 6b3a        P...U8..You.ask:
0x0030   2068 656c 6c6f 2120 2048 6f77 2061 7265        .hello!..How.are
0x0040   2079 6f75 2074 6f64 6179 3f1b 5b30 6d          .you.today?.[0m
```

Next, the Jkb user responds. The TCPDUMP packet sniffer was placed on the
Ethernet interface. Since Jkb was connected from the server machine and
does not need to traverse the Ethernet interface, messages sent by Jkb
to the Talker are not captured. Jkb's messages sent through the path that
TCPDUMP is monitoring, however, are captured and logged, as you can see below.
This is a realistic simulation because all users are not necessarily connected
over the same path through the Internet.

```
chimaera.universe.net.2121 > llama.universe.net.1158: P 2832:3051(219) ack 46 win 5840
0x0000   4500 0103 7f15 4000 4006 371f c0a8 0137        E.....@.@.7....7
0x0010   c0a8 0139 0849 0486 850e 66ee 62fc 0aa7        ...9.I....f.b...
0x0020   5018 16d0 ac55 0000 4a6b 6220 7361 7973        P....U..Jkb.says
0x0030   3a20 4669 6e65 2e20 2049 2061 6d20 7573        :.Fine...I.am.us
0x0040   696e 6720 5443 5044 554d 5020 746f 2076        ing.TCPDUMP.to.v
0x0050   6965 7720 616c 6c20 6f66 2074 6865 2070        iew.all.of.the.p
0x0060   6163 6b65 7473 2063 6f6d 696e 6720 696e        ackets.coming.in
0x0070   746f 2061 6e64 206f 7574 206f 6620 7468        to.and.out.of.th
0x0080   6520 5461 6c6b 6572 2773 2073 6572 7665        e.Talker's.serve
0x0090   7220 736f 636b 6574 2e20 2042 6563 6175        r.socket...Becau
0x00a0   7365 2074 6865 7920 6172 6520 7472 616e        se.they.are.tran
0x00b0   736d 6974 7465 6420 756e 656e 6372 7970        smitted.unencryp
0x00c0   7465 642c 2069 7420 6973 2065 6173 7920        ted,.it.is.easy.
0x00d0   746f 2070 756c 6c20 7468 6520 6461 7461        to.pull.the.data
0x00e0   2063 6f6e 7465 6e74 7320 6f75 7420 6f66        .contents.out.of
0x00f0   2074 6865 2070 6163 6b65 7473 2e0a 0d1b        .the.packets....
0x0100   5b30 6d                                        [0m
```

Summoner responds to this message and gets another acknowledgement.

```
llama.universe.net.1158 > chimaera.universe.net.2121: P 46:64(18) ack 3051 win 65535
```

```
0x0000   4500 003a 0f6b 4000 8006 6792 c0a8 0139        E..:.k@...g....9
0x0010   c0a8 0137 0486 0849 62fc 0aa7 850e 67c9        ...7...Ib.....g.
0x0020   5018 ffff 9e32 0000 5665 7279 2069 6e74        P....2..Very.int
0x0030   6572 6573 7469 6e67 210a                       eresting!.
chimaera.universe.net.2121 > llama.universe.net.1158: P 3051:3085(34) ack 64 win 5840
0x0000   4500 004a 7f16 4000 4006 37d7 c0a8 0137        E..J..@.@.7....7
0x0010   c0a8 0139 0849 0486 850e 67c9 62fc 0ab9        ...9.I....g.b...
0x0020   5018 16d0 68b6 0000 596f 7520 6578 636c        P...h...You.excl
0x0030   6169 6d3a 2056 6572 7920 696e 7465 7265        aim:.Very.intere
0x0040   7374 696e 6721 1b5b 306d                       sting!.[0m
llama.universe.net.1158 > chimaera.universe.net.2121: . ack 3085 win 65501 (DF)
0x0000   4500 0028 0f6c 4000 8006 67a3 c0a8 0139        E..(.l@...g....9
0x0010   c0a8 0137 0486 0849 62fc 0ab9 850e 67eb        ...7...Ib.....g.
0x0020   5010 ffdd c4b7 0000 0000 0000 0000             P.............
```

What I have demonstrated using the preceding experiment was the fact that Internet communication, in this case, Talkers, often send data in unencrypted form. Attackers employing tools such as TCPDUMP can capture this data and perform acts of blackmail, industrial or political espionage and disclosure of secrets. Capturing usernames and passwords could allow malicious users to commit identity theft crimes.

In conclusion, I think it would be a good idea to add some level of encryption to Talker communication. One of the goals of reimplementing the Talker in Java was because Java has some platform independent encryption libraries. Hopefully, experience with this project will allow me to provide some ways in which to implement encryption and secure the Talker communication.