

# ECN and it's impact on Intrusion Detection

*Toby Miller* 2000-11-03

## Introduction

Recently, there has been some discussion on various mailing lists about the Explicit Congestion Notification (ECN) proposed standard and QUESO/nmap scan detection. The debate has been centered around the two reserve bits in the TCP header (bits 8 & 9) that QUESO sets in a SYN packet and those same two bits being used by ECN.

What is ECN? ECN is a standard proposed by the IETF that will cut down on network congestion and routers dropping packets. Currently, RFC 2481 states that in order to accomplish this task ECN will use four previously unused bits in both the IP header and the TCP Header.

## IP Header

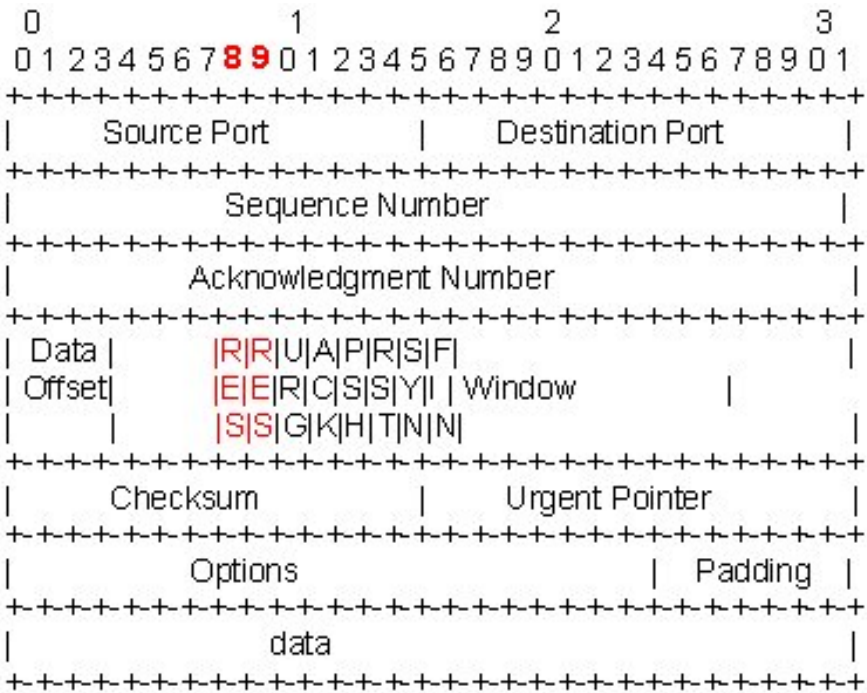
Two bits in the IP header that will be used are bits 6 (left of the low order bit) and 7 (low order bit) in the TOS field.

Over the years bits 6 and 7 have had various uses and meanings. Most recently bit 6 was used by the TOS field and was defined as "Minimize Monetary Cost" with a hex value of 0x02. Typically, NNTP would use the TOS value 0x02, however this function became obsolete by the "Differentiated Services" field for IPv4 and IPv6. Bit 7 was set to MBZ (must be zero). Again, this function became obsolete by the "Differentiated Services" field for IPv4 and IPv6. As part of ECN TOS bit 6 will now be used as an ECN capable transport (ECT). This bit will be set by the sender stating that both ends are ECN compatible. Bit 7 will now be used as a Congestion Experienced bit (CE). This bit is set by routers that detect congestion on the network.

## TCP Headers

RFC 2481 states that bits 8 and 9 in the TCP header will be used for ECN. One of the confusing aspects of this RFC is the placement of the bits. When I first read this RFC I went to my TCP/IP Illustrated Vol. 1 and began to break down the byte(s) where the two reserved bits are stored. I counted left to right; right to left and this whole process did not make sense. I e-mailed Judy Novak about this and after a few hundred e-mails (not really. Close though) I realized what the RFC was stating.

Let's take a look at the TCP Header:

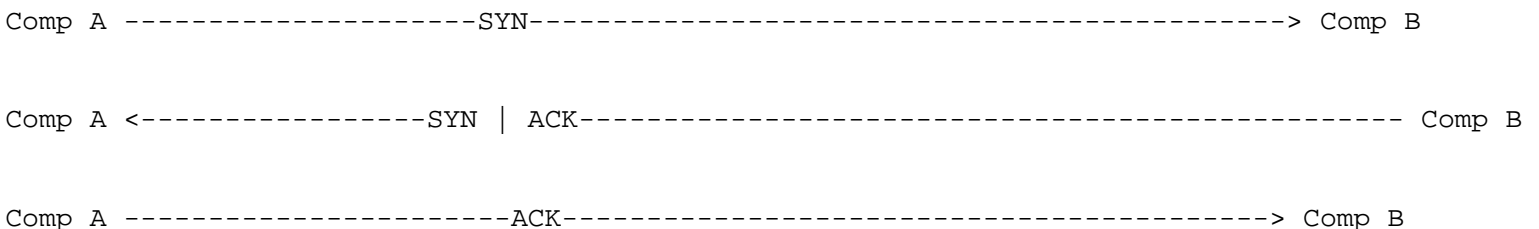


**Figure 1. TCP header <sup>1</sup>**

Figure 1 allows us to look at the header and the bit positions. Bits 8 and 9 (highlighted in red) in RFC 793 are normally reserved and should always be set to zero (0). RFC 2481 assigns a function to both bits. Bit 8 (High order bit) will be used as a Congestion Windows Reduced (CWR) flag. This flag will be used to inform the receiver that the congestion window has been reduced. Bit 9(right of high order bit) will be used as an ECN echo flag. This flag will be set to help negotiate ECN between the sender and receiver.

### Putting it together

Now that we have covered both the IP header and TCP header; lets take a look at how ECN works during the three - way handshake and during data transmissions. TCP uses a three - way handshake to make connections. The process looks like this:



ECN uses the three - way handshake to determine whether or not a sender and receiver are ECN compatible. During the initial SYN ECN will set TCP header bits 8 ( CWR flag) and bit 9 (ECN -Echo flag), if the receiver of this SYN is ECN compatible it will reply back in its SYN | ACK by setting TCP header bit 9. If the receiver is NOT compatible, the receiver will reply back by not setting any TCP header reserve bits. If this initialization is successful then the ECT flag will be set in all packets thereafter (except pure ACK's).

## What does this mean for Intrusion Detection?

It means that identifying an ECN packet vs. a QUESO or hping packet could become more difficult.

```
12:25:38.650123 attacker.com.1641 > localhost.111: S 1533993767:1533993767(0)
win 512 (ttl 64, id 64461)

          4500 0028 fbcd 0000 4006 91f5 xxxx xxxx
          xxxx xxxx 0669 006f 5b6e e327 0000 0000
          50c2 0200 7b16 0000
```

### Figure 2. TCPDUMP results

Figure 2 is a TCPDUMP trace that could have been created by hping, QUESO or ECN. What makes this packet so special? Look at the TCP flags (highlighted in blue) here we see c2. What does this mean? C2 tells us that this packet has the SYN flag along with the two reserved bits set. In the older days it normally meant that you were being probed by a tool such as QUESO or hping. Not so anymore. Since ECN now sets both reserved bits in the initial SYN, identifying a QUESO scan or forged packet will require more than a quick look. Another problem analysts will be presented with is TCPDUMP 3.5 using the -X switch.

```
12:26:38.650123 attacker.com.1641 > localhost.111: S [ECN-Echo,CWR] 380601688:380601688 (0)
win 4660 (ttl 244, id 55411)

          4500 0028 d873 0000 f406 85b4 xxxx xxxx
          xxxx xxxx 0669 006f 16af 8558 0000 0000
          50c2 1234 39e5 0000 753a 0000 e3a8
```

### Figure 3. TCPDUMP 3.5 results

Figure 3 is a portion of a TCPDUMP 3.5 output. I did not print out the information that was not related to this discussion. Figure 3 shows us that TCPDUMP 3.5 identifies this packet as an ECN packet.

This could present IDS analysts with a great number of false positives if the analysts are not aware of ECN and their network configuration.

Identifying this packet will require an analyst to look beyond the initial SYN. If the connection is made (as described above) and an analyst sees bit 6 in the TOS field (0x02) set then most likely these packets were legitimate. Otherwise, the analyst should continue to investigate.

```
15:01:47.649970 192.168.1.3.36998 > 192.168.1.1.1024: S [ECN-Echo] 22445461:22445461(0)
win 2048 (ttl 41, id 8872)

          4500 003c 22a8 0000 2906 ebbf c0a8 0103
```

```
c0a8 0101 9086 0400 0156 7d95 0000 0000
a042 0800 2a2e 0000 0303 0a01 0204 0109
080a 3f3f 3f3f
```

## FIGURE 4. NMAP Packet

We have talked about QUESO and hping setting the reserved bits, there is another program that sets at least one of the reserved bits. Figure 4 shows us a packet from a NMAP OS detect scan I ran. Here we see that NMAP will set the SYN flag and bit 9 (ECN-Echo) when trying to identify the operating system. How will this effect us in the future? Again, this means that we (IDS analysts) will need to see more then just one packet.

If you want to play with ECN the IETF has released a Linux kernel module for ECN. It is compatible with Linux kernel version 2.0.32, 2.2.5, 2.3.43.

## Conclusion

ECN could be a friend or foe, depending on how we attack this standard. I find this standard exciting because it will require us to take the "extra" step when doing analysis. This standard will also provide us with many false positives in the mean time so have fun.

## Critical Links or Papers

RFC 2481:

<http://www.ietf.org/rfc/rfc2481.txt?number=2481>

RFC 2884:

<http://www.ietf.org/rfc/rfc2884.txt?number=2884>

Everything else(including the kernel module):

<http://ftp.ee.lbl.gov/floyd/ecn.html>

ECN (Explicit Congestion Notification) in TCP/IP:

<http://www.aciri.org/floyd/ecn.html>

1. RFC 793 pp. 14 Transmission Control Protocol - <http://www.ietf.org>

*Toby Miller currently works at SYTEX Inc. based out of Pennsylvania. Toby holds a B.S in Computer Information Systems . Toby is a GIAC Certified Intrusion Analyst and a Microsoft Certified Professional. In his seven years in the computer field he has worked in many area such as Firewalls, Unix administration, NT Administration and some mainframe work.*

[Privacy Statement](#)

Copyright 2006, SecurityFocus