```
#PROXY Protocol Receiver iRule
# c.jenison at f5.com (Chad Jenison)
# v2.0 - Added support for PROXY Protocol v2, control for v1,v2 or lack of proxy via static:: variables set in RULE_INIT
# v2.1 - Fix for skipping bytes in v2 code
when RULE_INIT {
   set static::allowProxyV1 0
   set static::allowProxyV2 1
   set static::allowNoProxy 0
}

when CLIENT_ACCEPTED {
   TCP::collect
}

when CLIENT_DATA {
   binary scan [TCP::payload 12] H* v2_protocol_sig
   if {$static::allowProxyV1 && [TCP::payload 0 5] eq "PROXY"} {
      set proxy_string [TCP::payload]
      set proxy_string_length [expr {[string first "\r" [TCP::payload]] + 2}]
      scan $proxy_string {PROXY TCP%s%s%s%s%s} tcpver srcaddr dstaddr srcport dstport
      log "Proxy Protocol v1 conn from [IP::client_addr]:[TCP::client_port] for an IPv$tcpver stream from Src: $srcaddr:$srcport to Dst:
$dstaddr:$dstport"
      TCP::payload replace 0 $proxy_string_length ""
   } elseif {$static::allowProxyV2 && $v2_protocol_sig eq "0d0a0d0a000d0a515549540a"}{
      binary scan [TCP::payload] @12H* v2_proxyheaderremainder
      binary scan [TCP::payload] @12H2H* v2_verCommand v2_remainder
      if {$v2_verCommand == 21}{
         binary scan [TCP::payload] @13H2 v2_addressFamilyTransportProtocol
         if {$v2_addressFamilyTransportProtocol == 11} {
            binary scan [TCP::payload] @16ccccccccSS v2_sourceAddress1 v2_sourceAddress2 v2_sourceAddress3 v2_sourceAddress4
v2_destAddress1 v2_destAddress2 v2_destAddress3 v2_destAddress4 v2_sourcePort1 v2_destPort1
            set v2_sourceAddress "[expr {$v2_sourceAddress1 & 0xff}].[expr {$v2_sourceAddress2 & 0xff}].[expr {$v2_sourceAddress3 & 0xff}].[expr
{$v2_sourceAddress4 & 0xff}]"
            set v2_destAddress "[expr {$v2_destAddress1 & 0xff}].[expr {$v2_destAddress2 & 0xff}].[expr {$v2_destAddress3 & 0xff}].[expr
{$v2_destAddress4 & 0xff}]"
            set v2_sourcePort [expr {$v2_sourcePort1 & 0xffff}]
            set v2_destPort [expr {$v2_destPort1 & 0xffff}]
            log "Proxy Protocol v2 conn from [IP::client_addr]:[TCP::client_port] for an IPv4 Stream from Src: $v2_sourceAddress:$v2_sourcePort to Dst:
$v2_destAddress:$v2_destPort"
                                    binary scan [TCP::payload] @14S address_size
                                    set skip_bytes [expr 16 + $address_size]
            TCP::payload replace 0 $skip_bytes ""
         } elseif {$v2_addressFamilyTransportProtocol == 21} {
            binary scan [TCP::payload] @16H4H4H4H4H4H4H4H4 v2_v6sourceAddress1 v2_v6sourceAddress2 v2_v6sourceAddress3
v2_v6sourceAddress4 v2_v6sourceAddress5 v2_v6sourceAddress6 v2_v6sourceAddress7 v2_v6sourceAddress8
            binary scan [TCP::payload] @32H4H4H4H4H4H4H4H4 v2_v6destAddress1 v2_v6destAddress2 v2_v6destAddress3 v2_v6destAddress4
v2_v6destAddress5 v2_v6destAddress6 v2_v6destAddress7 v2_v6destAddress8
            binary scan [TCP::payload] @48SS v2_v6sourcePort1 v2_v6destPort1
            set v2_v6sourcePort [expr {$v2_v6sourcePort1 & 0xffff}]
            set v2_v6destPort [expr {$v2_v6destPort1 & 0xffff}]
            set v2_v6sourceAddress
"$v2_v6sourceAddress1:$v2_v6sourceAddress2:$v2_v6sourceAddress3:$v2_v6sourceAddress4:$v2_v6sourceAddress5:$v2_v6sourceAddress6:$v2_v
6sourceAddress7:$v2_v6sourceAddress8"
            set v2_v6destAddress
"$v2_v6destAddress1:$v2_v6destAddress2:$v2_v6destAddress3:$v2_v6destAddress4:$v2_v6destAddress5:$v2_v6destAddress6:$v2_v6destAddress7:
$v2_v6destAddress8"
            log "Proxy Protocol v2 conn from from [IP::client_addr]:[TCP::client_port] for an IPv6 Stream from Src:
$v2_v6sourceAddress:$v2_v6sourcePort to Dst: $v2_v6destAddress:$v2_v6destPort"
                                    binary scan [TCP::payload] @14S address_size
                                    set skip_bytes [expr 16 + $address_size]
            TCP::payload replace 0 $skip_bytes ""
         } else {
            log "v2_proxy conn from [IP::client_addr]:[TCP::client_port] - possible unknown/malformed transportProtocol or addressFamily"
            reject
         }
      } elseif {$v2_verCommand == 20}{
         log "Proxy Protocol v2 and LOCAL command from [IP::client_addr]:[TCP::client_port]; skipping"
                                    binary scan [TCP::payload] @14S address_size
                                    set skip_bytes [expr 16 + $address_size]
         TCP::payload replace 0 $skip_bytes ""
         binary scan [TCP::payload] H* local_remainder
      } else {
         log "Proxy Protocol Protocol Signature Detected from [IP::client_addr]:[TCP::client_port] but protocol version and command not legal;
connection reset"
         reject
      }
   } elseif {$static::allowNoProxy} {
      log "Connection from [IP::client_addr]:[TCP::client_port] allowed despite lack of PROXY protocol header"
   } else {
      reject
      log "Connection rejected from [IP::client_addr]:[TCP::client_port] due to lack of PROXY protocol header"
   }
   TCP::release
}
```