iRule: CCN Scrubber

Joe Pruitt, 2005-27-07

This is a tie-in to my last post regarding a SSN scrubber. How would you go about scrubbing out Credit Card Numbers? This isn't as simple as searching for a nnn-nn-nnnn pattern. CCNs vary in length depending on the issuer of the card. But one thing is common: they all must pass the Luhn Formula. Info on the Luhn Formula, or MOD 10 can be found here.

This example will look matching patterns looking like credit cards and return their indexes into the payload. Then the number is run through the Luhn formula (with optimizations by unRuleY). If it is indeed a valid credit card number, it is masked with X's.

```
when HTTP_REQUEST {
  # Don't allow data to be chunked
  if { [HTTP::version] eq "1.1" } {
      if { [HTTP::header is_keepalive] } {
         HTTP::header replace "Connection" "Keep-Alive"
      }
      HTTP::version "1.0"
   }
}

when HTTP_RESPONSE {
   if { [HTTP::header exists "Content-Length"] } {
      set content_length [HTTP::header "Content-Length"]
   } else {
      set content_length 4294967295
   }
   if { $content_length > 0 } {
      HTTP::collect $content_length
   }
}

when HTTP_RESPONSE_DATA {
  # Find ALL the possible credit card numbers in one pass
  set card_indices [regexp -all -inline -indices {(?:3[4-7]\d{13})|(?:4\d{15})|(?:5[1-5]\d{14})|(?:6011\d{12})} [HTTP::payload]]

  foreach card_idx $card_indices {
    set card_start [lindex $card_idx 0]
    set card_end [lindex $card_idx 1]
    set card_len [expr {$card_end - $card_start + 1}]
    set card_number [string range [HTTP::payload] $card_start $card_end]

    set double [expr {$card_len & 1}]
    set chksum 0
    set isCard invalid

    # Calculate MOD10
    for { set i 0 } { $i < $card_len } { incr i } {
       set c [string index $card_number $i]
       if {($i & 1) == $double} {
          if {[incr c $c] >= 10} {incr c -9}
       }
       incr chksum $c
    }

    # Determine Card Type
    switch [string index $card_number 0] {
       3 { set type AmericanExpress }
       4 { set type Visa }
       5 { set type MasterCard }
       6 { set type Discover }
       default { set type Unknown }
    }

    # If valid card number, then mask out numbers with X's
    if { ($chksum % 10) == 0 } {
       set isCard valid
       HTTP::payload replace $card_start $card_len [string repeat "X" $card_len]
    }

    # Log Results
    log local0. "Found $isCard $type CC# $card_number"
  }
}
```

Click here for the forum thread.

-Joe

[Listening to: Early Morning Rain - Gordon Lightfoot - Gord's Gold (03:18)]

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com