You say Luhn, I say mod 10

Joe Pruitt, 2005-18-07

Some fun stuff has been going on in the iRules forum with regards to security.

If you are interested in securing your site from exported social security numbers, you might want to take a look at this Tech Tip derived from an iRule.

Most recently we've been having some fun with the LUHN algorithm Basically it can tell a good credit card number from a bad one. Here's the algorithm (from wikipedia):

```
function checkLuhn(string purportedCC) {
    int sum := 0
    int nDigits := length(purportedCC)
    int parity := nDigits modulus 2
    for i from 0 to nDigits - 1 {
      int digit := integer(purportedCC[i])
      if i modulus 2 = parity
         digit := digit × 2
      if digit > 9
         digit := digit - 9
      sum := sum + digit
    }
    return (sum modulus 10) = 0
}
```

Unfortunately, there aren't any TCL implementations, but that didn't stop unRuleY. Here's unRuleY's optimized version. Always one to outdo himself, he replaced the expensive modulus operator with the less CPU-intensive bitwise AND.

```
# 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
}
```

Check out the thread for the entire rule!

-Joe