

IEPM Coding Perl Commenting

Kindly ensure that the code is properly commented. There should be comments, if not with each statement, then at least with each block of code or loop. For instance a file or a major block of code should begin with purpose and description of the code:



Perldoc?

Perhaps we should be stressing the use of perldoc also?

```
#####
#Get TULIP responsive host distances between Guthrie and Tulip
# The Input gives the IP addresses and distances of responsive TULIP hosts
# Example:
# 30.87.34.2,8551.236157
# 140.181.96.29,18488.23791
# 203.181.248.234,5148.805353
# 203.181.248.241,5148.805353
# 134.160.224.254,16684.70444
# 136.142.111.46,1685.3686
# Returns:
# A hash associated with the IP addresses with the value distance,
```

Each subroutine should also have a brief description like:

```
#####
#Executes synack to a target, and parses the results to provide
# 0 if successful or -1 otherwise. If unsuccessful then prints
#an error message to STDOUT.
# The target is provided in the argument, i.e. $_[0]
#Example of Unix synack output that is parsed by this sub:
#SYN-ACK to dns1.ethz.ch (129.132.98.12), 4 Packets
#
# connection for seq no: 0 timed out within 1.000 Secs
# connection for seq no: 1 timed out within 1.000 Secs
# connection for seq no: 2 timed out within 1.000 Secs
# connection for seq no: 3 timed out within 1.000 Secs
#
# Waiting for outstanding packets (if any).....
#
#
# ***** Round Trip Statistics of SYN-ACK to dns1.ethz.ch (Port = 22) *****
# 4 packets transmitted, 0 packets received, 100.00 percent packet loss
# round-trip (ms) min/avg/max = 0.000/0.000/0.000 (std = 0.000)
# (median = 0.000) (interquartile range = 0.000)
# (25 percentile = 0.000) (75 percentile = 0.000)
sub synack {
    my $target=$_[0];
    if($^O=~"linux") {
        my $success=0;
        ...
    }
}
```

It is always good to maintain a change history. It is of great help when applications are in production environments and someone needs to make a slight modification. So include a version number and date and the major changes made for this version, comment older version numbers and their descriptions, for example:

```
#my $version="3.73, 3/2/06, Les Cottrell";
# Replaced use of uname with $^O to improve tainting
# Increased the debug output to show the pwd
#my $version="3.74, 5/8/06, Les Cottrell";
#Enabled pings from onsite to onsite
my $version="4.0, 5/13/06, Les Cottrell";
#Added synack for SLAC
```



CVS

We are in the process of shifting everything into CVS. Putting versioning into the code is ugly and not very useful.

The rule of the thumb is to keep this thing in mind while coding.

"How would it feel if someone provided me a similar code and I had to make one small change and it took me a whole day just to point out that one line!! So if I don't want to mess up someone's day, I better put useful comments."