

T.I.N.A. Nagios Ampel

Geschrieben von: Thomas Tausend

Montag, den 17. September 2012 um 13:25 Uhr - Aktualisiert Montag, den 17. September 2012 um 14:54 Uhr



TINA steht für Tausend(s) IP-gesteuerte Nagios Ampel und stellt mit einer ausrangierten Verkehrsampel den Status des von Nagios ermittelten Systemstatus dar. Die bisherigen Erfahrungen zeigen, dass die Reaktionszeiten auf Störungen durch TINA deutlich verkürzt wurden, da Störungen einerseits schneller auffallen, andererseits die Motivation der Mitarbeiter gesteigert wird, den (für alle Kollegen sichtbaren) Status möglichst schnell wieder "grün" zu bekommen... .

Einkaufsliste:

- Koukaam NETIO-230B Steckdosenleiste mit integriertem LAN-Anschluss
- ausgediente Verkehrsampel mit 220V Birnen (z.B. bei Ebay ab ca. 30 €)
- ggf. je 3x Birnen, Schuko-Stecker, ausreichend Elektrokabel

Je nach Ausführung der Verkehrsampel verfügen diese bereits über 220V-Anschlüsse. Da viele der angebotenen Ampeln als Partykeller-Beleuchtung genutzt wurden, sind evtl. die 3 Farben parallel verschalten oder über Schalter geführt. Ziel ist letztendlich pro Farbe eine eigene Zuleitung mit einem Schuko-Stecker zur Verfügung zu haben. Praktischerweise gibt es Schukostecker auch schon in verschiedenen Farben, so dass man diese auch gleich in rot/gelb/grün kaufen kann.

T.I.N.A. Nagios Ampel

Geschrieben von: Thomas Tausend

Montag, den 17. September 2012 um 13:25 Uhr - Aktualisiert Montag, den 17. September 2012 um 14:54 Uhr

Das Perl-Plugin für die Ansteuerung der Ampel:

```
#!/usr/bin/perl -w
#
# T.I.N.A: Tausend's Intelligente Nagios Ampel - nagios addon
#
# Version for Koukaam NETIO-230 LAN-controlled Power-Switch x4
#
# Copyright (C) 2012 Thomas Tausend
# by Thomas.Tausend@ARCHTechniker.de
#
# This program is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License
# as published by the Free Software Foundation; either version 2
# of the License, or (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
use POSIX;
use strict;
use Getopt::Long;
use LWP::UserAgent;
use Data::Dumper;
use File::Basename;
use Time::HiRes qw/usleep/;
use Net::Telnet;
```

```
&Getopt::Long::config('bundling');
```

```
my %PATTERNS = ( 'green' => '1000', 'yellow' => '0100', 'red' => '0010', 'orange' => '0110',
```

```
'black' => '0000', 'all' => '1111' );
```

```
my $PROGNAME = basename($0);  
my $opt_url;  
my $opt_user;  
my $opt_passwd;  
my $opt_all;  
my $opt_pattern;  
my $ip;
```

```
my $opt_hostgroup;  
my $opt_servicegroup;
```

```
my $opt_h;  
my $status;  
my $debug = undef;  
my $force = 1;
```

```
$status = GetOptions(  
    "ip=s"      => $ip,  
    "a"        => $opt_all,  
    "all"      => $opt_all,  
    "url=s"    => $opt_url,  
    "hostgroup=s" => $opt_hostgroup,  
    "servicegroup=s" => $opt_servicegroup,  
    "help|h"   => $opt_h,  
    "user=s"   => $opt_user,  
    "passwd=s" => $opt_passwd,  
    "debug"    => $debug,  
# "force"     => $force,  
    "pattern=s" => $opt_pattern,  
);
```

```
if ( $opt_h || $status == 0 || !$opt_url ) {  
    print_help();  
}
```

```
    exit 0;
}
```

```
my $device = "test"; # find_light();
```

```
my $auth = "";
```

```
if ($opt_user) {
    $opt_url =~ s/http:////;
```

```
    $auth = "http://" . $opt_user . ":" . $opt_passwd . "@";
}
```

```
# OLD ...
```

```
#
```

```
my $url =
    $auth . $opt_url
```

```

"/cgi-bin/status.cgi?host=all&type=detail&servicestatustypes=20&serviceprops=42&noheader=1
&embedded=1&hostprops=42";
```

```
$url .= "&hostgroup=" . $opt_hostgroup."&style=all"
```

```
if ( $opt_hostgroup
    && $opt_hostgroup ne ""
    && $opt_hostgroup ne "--servicegroup" );
```

```
$url .= "&servicegroup=" . $opt_servicegroup
if ( $opt_servicegroup && $opt_servicegroup ne "" );
```

```
my $ua = new LWP::UserAgent;
```

```
$ua->agent( "TLightW-Agent/0.1 " . $ua->agent );
```

```
my $req = new HTTP::Request GET => $url;
```

```
my $res = $ua->request($req);
```

```
if ( $res->is_success ) {  
    my $content = $res->content;
```

```
    my $status = 'green';  
    my $matches;  
    my @lines;
```

```
    @lines = split( /n/, $content );
```

```
    my $device_status = 'green';  
    foreach (@lines) {  
        if ( /</TR></TABLE></TD>/ ... /</TR>/ ) {
```

```
            $device_status = 'green' if ( /</TR></TABLE></TD>/ );  
            $device_status = 'yellow' if ( m/statusWARNING/ );  
            $device_status = 'red'   if ( m/statusCRITICAL/ );
```

```
            # Only Hardstate are interesting  
            if ( m/>([0-9]+)/([0-9]+)</TD>/ ) {  
                $device_status = 'green' if ( $1 ne $2 );  
            }  
        }
```

```
    }  
    $status = $device_status if ( $device_status eq 'red' );  
    $status = $device_status  
        if ( $device_status eq 'yellow' and $status eq 'green' );
```

```
    }
  }
}
$status = 'red' if ( $content =~ m/statusHOSTDOWN/ );

my $perfddata = "|green=";
my $bulb = $PATTERNS{$status};

my $result = lightshow($status);

print "OK - Status "" . $status . ""|pattern: $bulbn";

}
else {
  print "Cannot reach nagios webinterface. Please check url, user and password!nn";
  exit 2; # CRITICAL;
}

sub print_help {
  printf "nusage: n";
  printf "$PROGNAME --url <url> [--user <user>] [--passwd <passwd>] [--hostgroup
<hostgroup>] [--servicegroup <servicegroup>] nn";
  printf " --url      url to nagiosn";
  printf " --user      nagios user for the webinterfacen";
  printf " --passwd     password for the nagios usern";
  printf " --hostgroup  nagios hostgroupn";
  printf " --servicegroup nagios servicegroupn";
  printf " --debug     show debuggingn";
  printf "n";
  printf "$PROGNAME comes with ABSOLUTELY NO WARRANTYn";
  printf "This program is licensed under the terms of the ";
  printf "GNU General Public Licensen(check source code for details)n";
  printf "nn";
  exit 0;
}
```

```
}

sub cleaner {
    my $text = shift;
    $text =~ s/ //g;      # remove blanks
    $text =~ s/<html>//g; # remove HTML-Start-Tag
    $text =~ s/</html>//g; # remove HTML-End-Tag
    return($text);
}

sub lightshow {
    my $color = shift;
    my $erg;
    if ($opt_pattern) {
        $color = $opt_pattern;
    }
    print "Overwrite $color with $opt_patternn";
    # Login to Power-Switch
    $erg = new HTTP::Request GET => " http://\$ip/tgi/control.tgi?l=p:\$opt\_user:\$opt\_passwd ";
;
    my $antwort = $ua->request($erg);
    #print "aktueller Status: " . cleaner($antwort->content) . "n";

    # read Status from Power-Switch
    $erg = new HTTP::Request GET => " http://\$ip/tgi/control.tgi?p=l ";
    $antwort = $ua->request($erg);

    # Set Color to given Parameter
    # print "Ermittelte Farbe fuer Parameter '$opt_pattern': " . $PATTERNS{$color};
    $erg = new HTTP::Request GET => " http://\$ip/tgi/control.tgi?p=" . $PATTERNS{$color};
    $antwort = $ua->request($erg);
    print "Content: " . cleaner($antwort->content) . "n";
    # print "Pattern $color aus Array: " . $PATTERNS{$color} . "n";
    # Quit
    $erg = new HTTP::Request GET => " http://\$ip/tgi/control.tgi?q=q ";
    $antwort = $ua->request($erg);
    #print "Content: " . cleaner($antwort->content) . "n";
```

T.I.N.A. Nagios Ampel

Geschrieben von: Thomas Tausend

Montag, den 17. September 2012 um 13:25 Uhr - Aktualisiert Montag, den 17. September 2012 um 14:54 Uhr

```
}
```

Das Script kann z.B. unter

```
/usr/local/nagios/libexec/check_tina.pl
```

abgelegt werden.

Sobald die Steckdosenleiste konfiguriert wurde, kann der Test beginnen. Die URL der Nagios-Oberfläche und der benötigten Passwörter wird ebenfalls benötigt.

Ein Test ist mit der Zeile

```
/usr/local/nagios/libexec/check_tina.pl --url &lt;Nagios-Web-Interface-URL&gt; --user  
<nagiosuser> --passwd <nagios-password> --ip <steckdosen-IP> --pattern all
```

möglich.

Als Pattern können dabei übergeben werden:

- red
- yellow
- green
- orange (red + yellow)
- black (alles aus)
- all (alles an)

T.I.N.A. Nagios Ampel

Geschrieben von: Thomas Tausend

Montag, den 17. September 2012 um 13:25 Uhr - Aktualisiert Montag, den 17. September 2012 um 14:54 Uhr

Das folgende Script wird dann z.B. in

```
/usr/local/nagios/etc/objects/tina/tina.cfg
```

abgelegt:

```
#####  
#####  
# Configuration File for T.I.N.A.: Tausends IP Nagios Alert (or "Ampel" for us Germans #  
#####  
#####  
# Create a new hostgroup for T.I.N.A.  
  
define hostgroup{  
    hostgroup_name      T.I.N.A.          ; The name of the hostgroup  
    alias                Tausends IP Nagios Ampel    ; Long name of the group  
}  
  
define host{  
    use                  generic-switch      ; Inherit default values from a template  
    host_name            TINA  
    alias                T.I.N.A. Nagios Ampel    ; A longer name associated with the  
switch  
    address              <IP-of-Power-Switch>    ; IP address of the switch  
    hostgroups           T.I.N.A.            ; Host groups this switch is associated with  
    max_check_attempts  10                  ; Check each server 10 times (max)  
}
```

```
define service{
    use                generic-service
    host_name          TINA
    service_description status
    max_check_attempts 10 ; Check each server 10 times (max)
    check_interval     1
    normal_check_interval 1
    retry_interval     1
    check_command
check_tina!<url-to-nagios-webinterface>!<nagios-user>!<nagios-password>
}
```

Die Parameter in <xxx> müssen natürlich noch angepasst werden!

In der command.cfg - meist unter

/usr/local/nagios/etc/objects/commands.cfg

zu finden, müssen noch die folgenden Zeilen ergänzt werden:

```
# command 'check_tina'
define command{
    command_name          check_tina
    command_line          $USER1$/check_tina.pl --ip $HOSTADDRESS$ --url $ARG1$ --user
$ARG2$ --passwd $ARG3$
}
```

Es würde mich natürlich sehr freuen, wenn ich Feedback bekomme wo TINA inzwischen überall ein Zuhause gefunden hat.

T.I.N.A. Nagios Ampel

Geschrieben von: Thomas Tausend

Montag, den 17. September 2012 um 13:25 Uhr - Aktualisiert Montag, den 17. September 2012 um 14:54 Uhr

Entsprechende Erfahrungsberichte und Bilder bitte an Thomas.Tausend@ARCHItechniker.de
- Tausend(s) Dank im Voraus!