

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

Как закрыть статистику (директорию) паролем в PFsense?

Нужно внести в файл `lighty-webConfigurator.conf` некоторые изменения, для этого правим файл `/etc/inc/system.inc`

Делаем так (добавили "mod_auth", ссылку на файл паролей `"/.pa.txt"` и защищаемый путь):

```
EOD;
}

$lighty_config .= <<<EOD
auth.backend = "htpasswd"
auth.backend.htpasswd.userfile = "/.pa.txt"
auth.require = ( "/lightsquid/" =>
(
"method" => "basic",
"realm" => "Statistic",
"require" => "valid-user"
)
)
EOD;

$fd = fopen("${filename}", "w");
....
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

или

```
...
## modules to load
server.modules      = (
    {${captive_portal_module}
    "mod_access", "mod_accesslog", "mod_expire", "mod_compress", "mod_redirect",
    "mod_auth", {${module}}{${captiveportal}
    )
.....

// Add HTTP to HTTPS redirect
if (${captive_portal} == false && ${config['system']['webgui']['protocol']} == "https" && !isset(${config['system']
if(${lighty_port} != "443")
    $redirectport = ":{${lighty_port}";
    $lighty_config .= <<<EOD
\${SERVER["socket"]} == ":80" {
    \${HTTP["host"]} =~ "(.*)" {
        url.redirect = ( "^/(.*)" => "https://%1{${redirectport}/${$1}" )
    }
}

EOD;
}

$lighty_config .= <<<EOD
auth.backend = "htpasswd"
auth.backend.htpasswd.userfile = "/.pa.txt"
auth.require = ( "/lightsquid/" =>
(
    "method" => "basic",
    "realm" => "Statistic",
    "require" => "valid-user"
)
)

EOD;

$fd = fopen("${filename}", "w");
....
```

- далее создаем файл с паролями /.pa.txt с помощью htpasswd.exe от apache (делал под виндой) и кладем на место

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

- перезагружаемся и проверяем

Добавлено!

Мой конфигурационный файл после изменений выглядит так:

```
<?php
/* $Id$ */
/*
system.inc
part of m0n0wall (http://m0n0.ch/wall)
```

Copyright (C) 2003-2004 Manuel Kasper <mk@neon1.net>.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
*/
```

```
/*
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
pfSense_BUILDER_BINARIES: /usr/sbin/powerd /usr/bin/killall /sbin/sysctl /sbin/route
pfSense_BUILDER_BINARIES: /bin/hostname /bin/ls /usr/sbin/syslogd
pfSense_BUILDER_BINARIES: /usr/sbin/pccardd /usr/local/sbin/lighttpd /bin/chmod /bin/mkdir
pfSense_BUILDER_BINARIES: /usr/bin/tar /usr/local/sbin/ntpd /usr/sbin/ntpdate
pfSense_BUILDER_BINARIES: /usr/bin/nohup /sbin/dmesg /usr/local/sbin/atareinit /sbin/kldload
pfSense_MODULE: utils
*/
```

```
function activate_powerd() {
global $config, $g;
if(isset($config['system']['powerd_enable'])) {
if ($g["platform"] == "nanobsd")
exec("/sbin/kldload cpufreq");
exec("/usr/sbin/powerd -b adp -a adp");
} else {
if(is_process_running("powerd"))
exec("/usr/bin/killall powerd");
}
}
```

```
function get_default_sysctl_value($id) {
global $sysctls;

if (isset($sysctls[$id]))
return $sysctls[$id];
}
```

```
function activate_sysctls() {
global $config, $g;
exec("/sbin/sysctl net.enc.out.ipsec_bpf_mask=0x00000001");
exec("/sbin/sysctl net.enc.out.ipsec_filter_mask=0x00000001");
exec("/sbin/sysctl net.enc.in.ipsec_bpf_mask=0x00000002");
exec("/sbin/sysctl net.enc.in.ipsec_filter_mask=0x00000002");
```

```
if(is_array($config['sysctl'])) {
foreach($config['sysctl']['item'] as $tunable) {
if($tunable['value'] == "default") {
$value = get_default_sysctl_value($tunable['tunable']);
mwexec("/sbin/sysctl " . $tunable['tunable'] . "=" . $value . "");
} else {
mwexec("/sbin/sysctl " . $tunable['tunable'] . "=" . $tunable['value'] . "");
}
}
}
}
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
function system_resolvconf_generate($dynupdate = false) {
    global $config, $g;

    if(isset($config['system']['developerspew'])) {
        $mt = microtime();
        echo "system_resolvconf_generate() being called $mt\n";
    }

    $syscfg = $config['system'];

    // Do not create blank domain lines, it breaks tools like dig.
    if($syscfg['domain'])
        $resolvconf = "domain {$syscfg['domain']}\n";

    if (isset($config['dnsmasq']['enable']) && !isset($config['system']['dnslocalhost'])
        $resolvconf .= "nameserver 127.0.0.1\n";

    if (isset($syscfg['dnsallowoverride'])) {
        /* get dynamically assigned DNS servers (if any) */
        $ns = array_unique(get_searchdomains());
        foreach($ns as $searchserver) {
            if($searchserver)
                $resolvconf .= "search {$searchserver}\n";
        }
        $ns = array_unique(get_nameservers());
        foreach($ns as $nameserver) {
            if($nameserver)
                $resolvconf .= "nameserver $nameserver\n";
        }
    }
    if (is_array($syscfg['dnsserver'])) {
        foreach ($syscfg['dnsserver'] as $ns) {
            if ($ns)
                $resolvconf .= "nameserver $ns\n";
        }
    }

    $dnslock = lock('resolvconf', LOCK_EX);

    $fd = fopen("{ $g['varetc_path']}/resolv.conf", "w");
    if (!$fd) {
        printf("Error: cannot open resolv.conf in system_resolvconf_generate().\n");
        unlock($dnslock);
        return 1;
    }
}
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
fwrite($fd, $resolvconf);
fclose($fd);

if (!$g['booting']) {
/* restart dhcpd (nameservers may have changed) */
if (!$dynupdate)
services_dhcpd_configure();
}

/* setup static routes for DNS servers. */
for ($dnscounter=1; $dnscounter<5; $dnscounter++) {
/* setup static routes for dns servers */
$dns gw = "dns{$dnscounter}gwint";
if (isset($config['system'][$dns gw])) {
$interface = $config['system'][$dns gw];
if (($interface <> "") && ($interface <> "none")) {
$gatewayip = get_interface_gateway($interface);
if (is_ipaddr($gatewayip)) {
/* dns server array starts at 0 */
$dnscountermo = $dnscounter - 1;
mwexec("route change -host " . $syscfg['dnsserver'][$dnscountermo] . " {$gatewayip}");
}
}
}
}

unlock($dnslock);

return 0;
}

function get_searchdomains() {
global $config, $g;

$master_list = array();

// Read in dhclient nameservers
$search_list = glob("/var/etc/searchdomain_*");
if (is_array($search_lists)) {
foreach($search_lists as $fdns) {
$content = file($fdns, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
if (!is_array($content))
continue;
foreach ($content as $dns) {
if(is_hostname($dns))
$master_list[] = $dns;
}
}
}
}
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
}
}
}

return $master_list;
}

function get_nameservers() {
global $config, $g;
$master_list = array();

// Read in dhclient nameservers
$dns_lists = glob("/var/etc/nameserver_*");
if (is_array($dns_lists)) {
foreach($dns_lists as $fdns) {
$content = file($fdns, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
if (!is_array($content))
continue;
foreach ($content as $dns) {
if(is_ipaddr($dns))
$master_list[] = $dns;
}
}
}

// Read in any extra nameservers
if(file_exists("/var/etc/nameservers.conf")) {
$dns_s = file("/var/etc/nameservers.conf", FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
if(is_array($dns_s)) {
foreach($dns_s as $dns)
if (is_ipaddr($dns))
$master_list[] = $dns;
}
}

return $master_list;
}

function system_hosts_generate() {
global $config, $g;
if(isset($config['system']['developerspew'])) {
$mt = microtime();
echo "system_hosts_generate() being called $mt\n";
}

$syscfg = $config['system'];
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
$dnsmasqcfg = $config['dnsmasq'];

if (!is_array($dnsmasqcfg['hosts'])) {
$dnsmasqcfg['hosts'] = array();
}
$hostscfg = $dnsmasqcfg['hosts'];

$hosts = "127.0.0.1 localhost localhost.${$syscfg['domain']}\n";
$lhosts = "";
$dhosts = "";

if ($config['interfaces']['lan']) {
$cfgip = get_interface_ip("lan");
if (is_ipaddr($cfgip))
$hosts .= "{$cfgip} {$syscfg['hostname']}.${$syscfg['domain']} {$syscfg['hostname']}\n";
} else {
$sysiflist = get_configured_interface_list();
foreach ($sysiflist as $sysif) {
if (!interface_has_gateway($sysif)) {
$cfgip = get_interface_ip($sysif);
if (is_ipaddr($cfgip)) {
$hosts .= "{$cfgip} {$syscfg['hostname']}.${$syscfg['domain']} {$syscfg['hostname']}\n";
break;
}
}
}
}

foreach ($hostscfg as $host) {
if ($host['host'])
$lhosts .= "{$host['ip']} {$host['host']}.${$host['domain']} {$host['host']}\n";
else
$lhosts .= "{$host['ip']} {$host['domain']}\n";
}
if (isset($dnsmasqcfg['regdhcpstatic']) && is_array($config['dhcpd'])) {
foreach ($config['dhcpd'] as $dhcpif => $dhcpifconf)
if (is_array($dhcpifconf['staticmap']) && isset($dhcpifconf['enable']))
foreach ($dhcpifconf['staticmap'] as $host)
if ($host['ipaddr'] && $host['hostname'])
$dhosts .= "{$host['ipaddr']} {$host['hostname']}.${$syscfg['domain']} {$host['hostname']}\n";
}

if (isset($dnsmasqcfg['dhcpfirst']))
$hosts .= $dhosts . $lhosts;
else
$hosts .= $lhosts . $dhosts;
```


Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
/*
 * Do not remove this because dhcpleases monitors with kqueue it needs to be
 * killed before writing to hosts files.
 */
if (file_exists("${$g['varrun_path']}/dhcpleases.pid")) {
sigkillbypid("${$g['varrun_path']}/dhcpleases.pid", "TERM");
@unlink("${$g['varrun_path']}/dhcpleases.pid");
}
$fd = fopen("${$g['varetc_path']}/hosts", "w");
if (!$fd) {
log_error("Error: cannot open hosts file in system_hosts_generate().\n");
return 1;
}
fwrite($fd, $hosts);
fclose($fd);

system_dhcpleases_configure();

return 0;
}

function system_dhcpleases_configure() {
global $config, $g;

/* Start the monitoring process for dynamic dhcpclients. */
if (isset($config['dnsmasq']['regdhcp'])) {
/* Make sure we do not error out */
@touch("${$g['dhcpcd_chroot_path']}/var/db/dhcpcd.leases");
if (file_exists("${$g['varrun_path']}/dhcpleases.pid"))
sigkillbypid("${$g['varrun_path']}/dhcpleases.pid", "HUP");
else
mwexec("/usr/local/sbin/dhcpleases -l ${$g['dhcpcd_chroot_path']}/var/db/dhcpcd.leases -d ${$config['system']}");
} else {
sigkillbypid("${$g['varrun_path']}/dhcpleases.pid", "TERM");
@unlink("${$g['varrun_path']}/dhcpleases.pid");
}
}

function system_hostname_configure() {
global $config, $g;
if(isset($config['system']['developerspew'])) {
$mt = microtime();
echo "system_hostname_configure() being called $mt\n";
}

$syscfg = $config['system'];
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
/* set hostname */
$status = mwexec("/bin/hostname " .
escapeshellarg("${syscfg['hostname']}.${syscfg['domain']}"));

/* Setup host GUID ID. This is used by ZFS. */
mwexec("/etc/rc.d/hostid start");

return $status;
}

function system_routing_configure($interface = "") {
global $config, $g;
if(isset($config['system']['developerspew'])) {
$mt = microtime();
echo "system_routing_configure() being called $mt\n";
}

$gatewayip = "";
$interfacegw = "";
$foundgw = false;
/* tack on all the hard defined gateways as well */
if (is_array($config['gateways']['gateway_item']) {
mwexec("/bin/rm ${g['tmp_path']}/*_defaultgw", true);
foreach ($config['gateways']['gateway_item'] as $gateway) {
if (isset($gateway['defaultgw'])) {
if(strpos($gateway['gateway'], ":"))
break;
if ($gateway['gateway'] == "dynamic")
$gateway['gateway'] = get_interface_gateway($gateway['interface']);
$gatewayip = $gateway['gateway'];
$interfacegw = $gateway['interface'];
if (!empty($interfacegw)) {
$defaultif = get_real_interface($gateway['interface']);
if ($defaultif)
@file_put_contents("${g['tmp_path']}/${$defaultif}_defaultgw", $gatewayip);
}
$foundgw = true;
break;
}
}
}
if ($foundgw == false) {
$defaultif = get_real_interface("wan");
$interfacegw = "wan";
$gatewayip = get_interface_gateway("wan");
@touch("${g['tmp_path']}/${$defaultif}_defaultgw");
}
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
}
$dont_add_route = false;
/* if OLSRD is enabled, allow WAN to house DHCP. */
if($config['installedpackages']['olsrd']) {
foreach($config['installedpackages']['olsrd']['config'] as $olsrd) {
if($olsrd['enabledyngw'] == "on") {
$dont_add_route = true;
break;
}
}
}

if ($dont_add_route == false ) {
if (!empty($interface) && $interface != $interfacegw)
;
else if (($interfacegw <> "bgpd") && (is_ipaddr($gatewayip))) {
log_error("ROUTING: setting default route to $gatewayip");
mwexec("/sbin/route change -inet default " . escapeshellarg($gatewayip));
}
}

if (is_array($config['staticroutes']['route'])) {
$gateways_arr = return_gateways_array();

foreach ($config['staticroutes']['route'] as $rtent) {
$gatewayip = "";
if (empty($gateways_arr[$rtent['gateway']])) {
log_error("Static Routes: Gateway IP could not be found for {$rtent['network']}");
continue;
}
$gateway = $gateways_arr[$rtent['gateway']];
if (!empty($interface) && $interface != $gateway['friendlyiface'])
continue;
$gatewayip = $gateway['gateway'];
$interfacegw = $gateway['interface'];

if (is_ipaddr($gatewayip)) {
mwexec("/sbin/route change -inet " . escapeshellarg($rtent['network']) .
" " . escapeshellarg($gatewayip));
} else if (!empty($interfacegw)) {
mwexec("/sbin/route change -inet " . escapeshellarg($rtent['network']) .
" -iface " . escapeshellarg($interfacegw));
}
}
}
}
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
return 0;
}

function system_routing_enable() {
global $config, $g;
if(isset($config['system']['developerspew'])) {
$mt = microtime();
echo "system_routing_enable() being called $mt\n";
}

return mwexec("/sbin/sysctl net.inet.ip.forwarding=1");
}

function system_syslogd_get_remote_servers($syslogcfg, $facility = ".*") {
// Rather than repeatedly use the same code, use this function to build a list of remote servers.
$facility .= " ";
$remote_servers = "";
$pad_to = 56;
$padding = ceil(($pad_to - strlen($facility))/8)+1;
if($syslogcfg['remoteserver'])
$remote_servers .= "{$facility}" . str_repeat("\t", $padding) . "@{$syslogcfg['remoteserver']}\n";
if($syslogcfg['remoteserver2'])
$remote_servers .= "{$facility}" . str_repeat("\t", $padding) . "@{$syslogcfg['remoteserver2']}\n";
if($syslogcfg['remoteserver3'])
$remote_servers .= "{$facility}" . str_repeat("\t", $padding) . "@{$syslogcfg['remoteserver3']}\n";
return $remote_servers;
}

function system_syslogd_start() {
global $config, $g;
if(isset($config['system']['developerspew'])) {
$mt = microtime();
echo "system_syslogd_start() being called $mt\n";
}

mwexec("/etc/rc.d/hostid start");

$syslogcfg = $config['syslog'];

if ($g['booting'])
echo "Starting syslog...";
else
killbypid("#{$g['varrun_path']}/syslog.pid");

if(is_process_running("syslogd"))
mwexec('/bin/pkill syslogd');
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
if(is_process_running("fifolog_writer"))
mwexec('/bin/pkill fifolog_writer');

// Define carious commands for logging
$fifolog_create = "/usr/sbin/fifolog_create -s ";
$fifolog_log = "|/usr/sbin/fifolog_writer ";
$clog_create = "/usr/sbin/clog -i -s ";
$clog_log = "%";

// Which logging type are we using this week??
if(isset($config['system']['usefifolog'])) {
$log_directive = $fifolog_log;
$log_create_directive = $fifolog_create;
} else { // Defaults to CLOG
$log_directive = $clog_log;
$log_create_directive = $clog_create;
}

if (isset($syslogcfg)) {
$separatelogfacilities = array('ntpd','racoond','openvpn','pptps','poes','l2tps','relayd','hostapd');
if($config['installedpackages']['package']) {
foreach($config['installedpackages']['package'] as $package) {
if($package['logging']) {
array_push($separatelogfacilities, $package['logging']['facilityname']);
mwexec("{ $log_create_directive } 10240 { $g['varlog_path'] } / { $package['logging']['logfile name'] }");
$syslogconf .= "!{ $package['logging']['facilityname'] } \n * . * \t \t \t \t \t { $log_directive } { $g['varlog_path'] } / { $pac
}
}
}
}
$facilitylist = implode(',', array_unique($separatelogfacilities));
/* write syslog.conf */
$fd = fopen("{ $g['varetc_path'] } / syslog.conf", "w");
if (!$fd) {
printf("Error: cannot open syslog.conf in system_syslogd_start().\n");
return 1;
}
$syslogconf .= "!ntpd\n";
if (!isset($syslogcfg['disablelocallogging']))
$syslogconf .= "*.*
{$log_directive}{ $g['varlog_path'] } / ntpd.log\n";

$syslogconf .= "!ppp\n";
if (!isset($syslogcfg['disablelocallogging']))
$syslogconf .= "*.*
{$log_directive}{ $g['varlog_path'] } / ppp.log\n";

$syslogconf .= "!pptps\n";
if (!isset($syslogcfg['disablelocallogging']))
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
$syslogconf .= "**.*                {$log_directive}{$g['varlog_path']}/ppts.log\n";

$syslogconf .= "!poes\n";
if (!isset($syslogcfg['disablelocallogging']))
$syslogconf .= "**.*                {$log_directive}{$g['varlog_path']}/poes.log\n";

$syslogconf .= "!l2tps\n";
if (!isset($syslogcfg['disablelocallogging']))
$syslogconf .= "**.*                {$log_directive}{$g['varlog_path']}/l2tps.log\n";

$syslogconf .= "!racoon\n";
if (!isset($syslogcfg['disablelocallogging']))
$syslogconf .= "**.*                {$log_directive}{$g['varlog_path']}/ipsec.log\n";
if (isset($syslogcfg['vpn']))
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "**.*");

$syslogconf .= "!openvpn\n";
if (!isset($syslogcfg['disablelocallogging']))
$syslogconf .= "**.*                {$log_directive}{$g['varlog_path']}/openvpn.log\n";
if (isset($syslogcfg['vpn']))
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "**.*");

$syslogconf .= "!apinger\n";
if (!isset($syslogcfg['disablelocallogging']))
$syslogconf .= "**.*                {$log_directive}{$g['varlog_path']}/apinger.log\n";
if (isset($syslogcfg['apinger']))
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "**.*");

$syslogconf .= "!relayd\n";
if (!isset($syslogcfg['disablelocallogging']))
$syslogconf .= "**.*                {$log_directive}{$g['varlog_path']}/relayd.log\n";
if (isset($syslogcfg['relayd']))
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "**.*");

$syslogconf .= "!hostapd\n";
if (!isset($syslogcfg['disablelocallogging']))
$syslogconf .= "**.*                {$log_directive}{$g['varlog_path']}/wireless.log\n";
if (isset($syslogcfg['hostapd']))
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "**.*");

$syslogconf .= "!-{$facilitylist}\n";
if (!isset($syslogcfg['disablelocallogging']))
$syslogconf .= <<<EOD
local0.*                {$log_directive}{$g['varlog_path']}/filter.log
local3.*                {$log_directive}{$g['varlog_path']}/vpn.log
local4.*                {$log_directive}{$g['varlog_path']}/portalauth.log
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
local7.*                {$log_directive}{$g['varlog_path']}/dhcpd.log
*.notice;kern.debug;lpr.info;mail.crit;        {$log_directive}{$g['varlog_path']}/system.log
news.err;local0.none;local3.none;local4.none;  {$log_directive}{$g['varlog_path']}/system.log
local7.none            {$log_directive}{$g['varlog_path']}/system.log
security.*            {$log_directive}{$g['varlog_path']}/system.log
auth.info;authpriv.info;daemon.info          {$log_directive}{$g['varlog_path']}/system.log
auth.info;authpriv.info |exec /usr/local/sbin/sshlockout_pf 15
*.emerg                *
```

EOD;

```
if (isset($syslogcfg['filter']))
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "local0.*");
if (isset($syslogcfg['vpn']))
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "local3.*");
if (isset($syslogcfg['portalauth']))
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "local4.*");
if (isset($syslogcfg['dhcp']))
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "local7.*");
if (isset($syslogcfg['system'])) {
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "*.notice;kern.debug;lpr.info;mail.crit;");
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "news.err;local0.none;local3.none;local4.none;");
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "security.*");
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "auth.info;authpriv.info;daemon.info");
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "*.emerg");
}
if (isset($syslogcfg['logall'])) {
// Make everything mean everything, including facilities excluded above.
$syslogconf .= "!*\n";
$syslogconf .= system_syslogd_get_remote_servers($syslogcfg, "*.");
}
}
```

```
if (isset($syslogcfg['zmqserver'])) {
$syslogconf .= <<<EOD
*.*                ^{$syslogcfg['zmqserver']}
```

EOD;

```
}
fwrite($fd, $syslogconf);
fclose($fd);
```

```
// Ensure that the log directory exists
if(!is_dir("${g['dhcpd_chroot_path']}/var/run"))
exec("/bin/mkdir -p {g['dhcpd_chroot_path']}/var/run");
```

```
// Are we logging to a least one remote server ?
if(strpos($syslogconf, "@") != false)
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
$retval = mwexec_bg("/usr/sbin/syslogd -c -c -l /var/dhcpd/var/run/log -f ${g['varetc_path']}/syslog.conf")
else {
$retval = mwexec_bg("/usr/sbin/syslogd -c -c -l /var/dhcpd/var/run/log -f ${g['varetc_path']}/syslog.conf")
}

} else {
$retval = mwexec_bg("/usr/sbin/syslogd -c -c -l /var/dhcpd/var/run/log");
}

if ($g['booting'])
echo "done.\n";

return $retval;
}

function system_pccard_start() {
global $config, $g;
if(isset($config['system']['developerspew'])) {
$mt = microtime();
echo "system_pccard_start() being called $mt\n";
}

if ($g['booting'])
echo "Initializing PCMCIA...";

/* kill any running pccardd */
killbypid("${g['varrun_path']}/pccardd.pid");

/* fire up pccardd */
$res = mwexec("/usr/sbin/pccardd -z -f ${g['etc_path']}/pccard.conf");

if ($g['booting']) {
if ($res == 0)
echo "done.\n";
else
echo "failed!\n";
}

return $res;
}

function system_webgui_start() {
global $config, $g;

if ($g['booting'])
```


Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
echo "Starting webConfigurator...";

/* kill any running lighttpd */
killbypid("${$g['varrun_path']}/lighty-webConfigurator.pid");

sleep(1);

chdir(${g['www_path']});

/* defaults */
$portarg = "80";
$crt = "";
$key = "";
$ca = "";

/* non-standard port? */
if (isset($config['system']['webgui']['port']) && $config['system']['webgui']['port'] <> "")
    $portarg = "${config['system']['webgui']['port']}";

if ($config['system']['webgui']['protocol'] == "https") {
    // Ensure that we have a webConfigurator CERT
    $cert =& lookup_cert($config['system']['webgui']['ssl-certref']);
    if (!is_array($cert) && !$cert['crt'] && !$cert['prv']) {
        if (!is_array($config['ca']))
            $config['ca'] = array();
        $a_ca =& $config['ca'];
        if (!is_array($config['cert']))
            $config['cert'] = array();
        $a_cert =& $config['cert'];
        log_error("Creating SSL Certificate for this host");
        $cert = array();
        $cert['refid'] = uniqid();
        $cert['descr'] = "webConfigurator default";
        mwexec("/usr/bin/openssl genrsa 1024 > ${g['tmp_path']}/ssl.key");
        mwexec("/usr/bin/openssl req -new -x509 -nodes -sha1 -days 2000 -key ${g['tmp_path']}/ssl.key > ${g['tmp_path']}/ssl.crt");
        $crt = file_get_contents("${g['tmp_path']}/ssl.crt");
        $key = file_get_contents("${g['tmp_path']}/ssl.key");
        unlink("${g['tmp_path']}/ssl.key");
        unlink("${g['tmp_path']}/ssl.crt");
        cert_import($cert, $crt, $key);
        $a_cert[] = $cert;
        $config['system']['webgui']['ssl-certref'] = $cert['refid'];
        write_config("Importing HTTPS certificate");
        if (!$config['system']['webgui']['port'])
            $portarg = "443";
        $ca = ca_chain($cert);
    }
}
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
} else {
$cert = base64_decode($cert['crt']);
$key = base64_decode($cert['prv']);
if (!$config['system']['webgui']['port'])
$portarg = "443";
$ca = ca_chain($cert);
}
}

/* generate lighttpd configuration */
$max_procs = ($config['system']['webgui']['max_procs']) ? $config['system']['webgui']['max_procs'] : 2;
system_generate_lighty_config("${$g['varetc_path']}/lighty-webConfigurator.conf",
$cert, $key, $ca, "lighty-webConfigurator.pid", $portarg, "/usr/local/www/",
"cert.pem", "ca.pem", $max_procs);

/* attempt to start lighthttpd */
$res = mwexec("/usr/local/sbin/lighttpd -f {$g['varetc_path']}/lighty-webConfigurator.conf");

/* fetch page to preload apc cache */
$proto = "http";
if ($config['system']['webgui']['protocol'])
$proto = $config['system']['webgui']['protocol'];
mwexec_bg("/usr/bin/fetch -o /dev/null -q {$proto}://localhost:{$portarg}/preload.php");

if ($g['booting']) {
if ($res == 0)
echo "done.\n";
else
echo "failed!\n";
}

return $res;
}

function system_generate_lighty_config($filename,
$cert,
$key,
$ca,
$pid_file,
$port = 80,
$document_root = "/usr/local/www/",
$cert_location = "cert.pem",
$ca_location = "ca.pem",
$max_procs = 1,
$max_requests = "2",
$fast_cgi_enable = true,
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
$captive_portal = false) {

global $config, $g;

if(!is_dir("${$g['tmp_path']}/lighttpdcompress"))
mkdir("${$g['tmp_path']}/lighttpdcompress");

if(isset($config['system']['developerspew'])) {
$mt = microtime();
echo "system_generate_lighty_config() being called $mt\n";
}

if($captive_portal == true) {
$captiveportal = ",\nmod_rewrite\n";
$captive_portal_rewrite = "url.rewrite-once = ( \"(.captiveportal.*)\" => \"$1\", \"(.*)\" => \"/index.php?red
$captive_portal_module = "";
$maxprocperip = $config['captiveportal']['maxprocperip'];
if($maxprocperip and $maxprocperip > 0)
$captive_portal_mod_evasive = "evasive.max-conns-per-ip = {$maxprocperip}";
else
$captive_portal_mod_evasive = "";
$server_upload_dirs = "server.upload-dirs = ( \"${$g['tmp_path']}/captiveportal/\" )\n";
exec("mkdir -p ${$g['tmp_path']}/captiveportal");
exec("chmod a-w ${$g['tmp_path']}/captiveportal");
$server_max_request_size = "server.max-request-size = 384";
} else {
$captiveportal = "";
$captive_portal_rewrite = "";
$captive_portal_module = "";
$captive_portal_mod_evasive = "";
$server_upload_dirs = "server.upload-dirs = ( \"${$g['upload_path']}/\", \"${$g['tmp_path']}/\", \"/var/\" )\n";
$server_max_request_size = "server.max-request-size = 2097152";
}

if($port <> "")
$lighty_port = $port;
else
$lighty_port = "80";

$memory = get_memory();
$avail = $memory[0];

if($avail > 0 and $avail < 65) {
$fast_cgi_enable = false;
}
}
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
// Ramp up captive portal max procs
// Work relative to the default of 2, for values that would be >2.
if($captive_portal == true) {
if($avail > 65 and $avail < 98) {
$max_procs = 1;
}
if($avail > 97 and $avail < 128) {
$max_procs = 2;
}
if($avail > 127 and $avail < 256) {
$max_procs += 1;
}
if($avail > 255 and $avail < 384) {
$max_procs += 2;
}
if($avail > 383) {
$max_procs += 3;
}
}

if($captive_portal == true) {
$bin_environment = <<<EOC
"bin-environment" => (
"PHP_FCGI_CHILDREN" => "$max_procs",
"PHP_FCGI_MAX_REQUESTS" => "500"
),
EOC;

} else if ($avail > 0 and $avail < 128) {
$bin_environment = <<<EOC
"bin-environment" => (
"PHP_FCGI_CHILDREN" => "$max_procs",
"PHP_FCGI_MAX_REQUESTS" => "2",
),
EOC;

} else
$bin_environment = <<<EOC
"bin-environment" => (
"PHP_FCGI_CHILDREN" => "$max_procs",
"PHP_FCGI_MAX_REQUESTS" => "500"
),
EOC;

if($fast_cgi_enable == true) {
$module = "\"mod_fastcgi\", \"mod_cgi\"";
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
$cgi_config = "";
$fastcgi_config = <<<EOD
#### fastcgi module
## read fastcgi.txt for more info
fastcgi.server = ( ".php" =>
( "localhost" =>
(
"socket" => "${g['tmp_path']}/php-fastcgi.socket",
"min-procs" => 0,
"max-procs" => {$max_procs},
{$bin_environment}
"bin-path" => "/usr/local/bin/php"
)
)
)

#### CGI module
cgi.assign          = ( ".cgi" => "" )

EOD;
} else {
$fastcgi_config = "";
$module = "\"mod_cgi\"";
$cgi_config = <<<EOD
#### CGI module
cgi.assign          = ( ".php" => "/usr/local/bin/php",
".cgi" => "" )

EOD;
}

$lighty_config = "";
$lighty_config .= <<<EOD
#
# lighttpd configuration file
#
# use a it as base for lighttpd 1.0.0 and above
#
##### Options you really have to take care of #####

## FreeBSD!
server.event-handler = "freebsd-kqueue"
server.network-backend = "writev"

## modules to load
server.modules       = (
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
"mod_access", "mod_accesslog", "mod_expire", "mod_compress", "mod_redirect",  
"mod_auth", {$module}{$captiveportal}  
)
```

```
## Unused modules
```

```
# "mod_setenv",  
# "mod_rewrite",  
# "mod_ssi",  
# "mod_usertrack",  
# "mod_expire",  
# "mod_secdownload",  
# "mod_rrdtool",  
# "mod_auth",  
# "mod_status",  
# "mod_alias",  
# "mod_proxy",  
# "mod_simple_vhost",  
# "mod_evhost",  
# "mod_userdir",  
# "mod_cgi",
```

```
server.max-keep-alive-requests = 15
```

```
server.max-keep-alive-idle = 30
```

```
## a static document-root, for virtual-hosting take look at the
```

```
## server.virtual-* options
```

```
server.document-root = "{$document_root}"  
{$captive_portal_rewrite}
```

```
# Maximum idle time with nothing being written (php downloading)
```

```
server.max-write-idle = 999
```

```
## where to send error-messages to
```

```
server.errorlog = "/var/log/lighttpd.error.log"
```

```
# files to check for if .../ is requested
```

```
server.indexfiles = ( "index.php", "index.html",  
"index.htm", "default.htm" )
```

```
# mimetype mapping
```

```
mimetype.assign = (  
".pdf" => "application/pdf",  
".sig" => "application/pgp-signature",  
".spl" => "application/futuresplash",  
".class" => "application/octet-stream",  
".ps" => "application/postscript",
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
".torrent" => "application/x-bittorrent",
".dvi"     => "application/x-dvi",
".gz"      => "application/x-gzip",
".pac"     => "application/x-ns-proxy-autoconfig",
".swf"     => "application/x-shockwave-flash",
".tar.gz"  => "application/x-tgz",
".tgz"     => "application/x-tgz",
".tar"     => "application/x-tar",
".zip"     => "application/zip",
".mp3"     => "audio/mpeg",
".m3u"     => "audio/x-mpegurl",
".wma"     => "audio/x-ms-wma",
".wax"     => "audio/x-ms-wax",
".ogg"     => "audio/x-wav",
".wav"     => "audio/x-wav",
".gif"     => "image/gif",
".jpg"     => "image/jpeg",
".jpeg"    => "image/jpeg",
".png"     => "image/png",
".xbm"     => "image/x-xbitmap",
".xpm"     => "image/x-xpixmap",
".xwd"     => "image/x-xwindowdump",
".css"     => "text/css",
".html"    => "text/html",
".htm"     => "text/html",
".js"      => "text/javascript",
".asc"     => "text/plain",
".c"       => "text/plain",
".conf"    => "text/plain",
".text"    => "text/plain",
".txt"     => "text/plain",
".dtd"     => "text/xml",
".xml"     => "text/xml",
".mpeg"    => "video/mpeg",
".mpg"     => "video/mpeg",
".mov"     => "video/quicktime",
".qt"      => "video/quicktime",
".avi"     => "video/x-msvideo",
".asf"     => "video/x-ms-asf",
".asx"     => "video/x-ms-asf",
".wmv"     => "video/x-ms-wmv",
".bz2"     => "application/x-bzip",
".tbz"     => "application/x-bzip-compressed-tar",
".tar.bz2" => "application/x-bzip-compressed-tar"
)
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
# Use the "Content-Type" extended attribute to obtain mime type if possible
#mimetypes.use-xattr      = "enable"

#### accesslog module
#accesslog.filename      = "/dev/null"

## deny access the file-extensions
#
# ~   is for backupfiles from vi, emacs, joe, ...
# .inc is often used for code includes which should in general not be part
#     of the document-root
url.access-deny          = ( "~", ".inc" )

##### Options that are good to be but not necessary to be changed #####

## bind to port (default: 80)
server.port              = {$lighty_port}

## error-handler for status 404
#server.error-handler-404 = "/error-handler.html"
#server.error-handler-404 = "/error-handler.php"

## to help the rc.scripts
server.pid-file          = "/var/run/{$pid_file}"

## virtual directory listings
server.dir-listing       = "disable"

## enable debugging
debug.log-request-header = "disable"
debug.log-response-header = "disable"
debug.log-request-handling = "disable"
debug.log-file-not-found = "disable"

# gzip compression
compress.cache-dir = "{$g[\"tmp_path\"]}/lighttpdcompress/"
compress.filetype = ("text/plain","text/css", "text/xml", "text/javascript" )

{$server_upload_dirs}

{$server_max_request_size}

{$fastcgi_config}

{$cgi_config}
```


Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
{${captive_portal_mod_evasive}
```

```
expire.url = (  
  "" => "access 50 hours",  
)
```

```
EOD;
```

```
$cert = str_replace("\r", "", $cert);  
$key = str_replace("\r", "", $key);  
$ca = str_replace("\r", "", $ca);
```

```
$cert = str_replace("\n\n", "\n", $cert);  
$key = str_replace("\n\n", "\n", $key);  
$ca = str_replace("\n\n", "\n", $ca);
```

```
if($cert <> "" and $key <> "") {  
  $fd = fopen("${g['varetc_path']}/${cert_location}", "w");  
  if (!$fd) {  
    printf("Error: cannot open cert.pem in system_webgui_start().\n");  
    return 1;  
  }  
  chmod("${g['varetc_path']}/${cert_location}", 0600);  
  fwrite($fd, $cert);  
  fwrite($fd, "\n");  
  fwrite($fd, $key);  
  fclose($fd);  
  if(!(empty($ca) || (strlen(trim($ca)) == 0))) {  
    $fd = fopen("${g['varetc_path']}/${ca_location}", "w");  
    if (!$fd) {  
      printf("Error: cannot open ca.pem in system_webgui_start().\n");  
      return 1;  
    }  
    chmod("${g['varetc_path']}/${ca_location}", 0600);  
    fwrite($fd, $ca);  
    fclose($fd);  
  }  
  $lighty_config .= "\n";  
  $lighty_config .= "### ssl configuration\n";  
  $lighty_config .= "ssl.engine = \"enable\"\n";  
  $lighty_config .= "ssl.pemfile = \"${g['varetc_path']}/${cert_location}\"\n\n";
```

```
// Harden SSL a bit for PCI conformance testing
```

```
$lighty_config .= "ssl.use-ssl2 = \"disable\"\n";
```

```
$lighty_config .= "ssl.cipher-list = \"TLSv1+HIGH !SSLv2 RC4+MEDIUM !aNULL !eNULL !3DES @STR
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
if(!(empty($ca) || (strlen(trim($ca)) == 0)))
$lighty_config .= "ssl.ca-file = \"${$g['varetc_path']}/${$ca_location}\"\\n\\n";
}
```

```
// Add HTTP to HTTPS redirect
```

```
// Add HTTP to HTTPS redirect
if ($captive_portal == false && $config['system']['webgui']['protocol'] == "https" && !isset($config['system
if($lighty_port != "443")
$redirectport = ":{$lighty_port}";
$lighty_config .= <<<<EOD
\\$SERVER["socket"] == ":80" {
\\$HTTP["host"] =~ "(.*)" {
url.redirect = ( "!^(.*)" => "https://%1{$redirectport}/$1" )
}
}
```

```
EOD;
}
```

```
$lighty_config .= <<<<EOD
auth.backend = "htpasswd"
auth.backend.htpasswd.userfile = "/.htpasswd"
auth.require = ( "/lightsquid/" =>
(
"method" => "basic",
"realm" => "Statistic",
"require" => "valid-user"
)
)
```

```
EOD;
```

```
$fd = fopen("${$filename}", "w");
```

```
if ($captive_portal == false && $config['system']['webgui']['protocol'] == "https" && !isset($config['system
if($lighty_port != "443")
$redirectport = ":{$lighty_port}";
$lighty_config .= <<<<EOD
\\$SERVER["socket"] == ":80" {
\\$HTTP["host"] =~ "(.*)" {
url.redirect = ( "!^(.*)" => "https://%1{$redirectport}/$1" )
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
}
}
EOD;
}

$fd = fopen("${filename}", "w");
if (!$fd) {
printf("Error: cannot open {filename} in system_generate_lighty_config().\n");
return 1;
}
fwrite($fd, $lighty_config);
fclose($fd);

return 0;

}

function system_timezone_configure() {
global $config, $g;
if(isset($config['system']['developerspew'])) {
$mt = microtime();
echo "system_timezone_configure() being called $mt\n";
}

$syscfg = $config['system'];

if ($g['booting'])
echo "Setting timezone...";

/* extract appropriate timezone file */
$timezone = $syscfg['timezone'];
if (!$timezone)
$timezone = "Etc/UTC";

conf_mount_rw();

exec("LANG=C /usr/bin/tar xzfO /usr/share/zoneinfo.tgz " .
escapeshellarg($timezone) . " > /etc/localtime");

mwexec("sync");
conf_mount_ro();

if ($g['booting'])
echo "done.\n";
}
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
function system_ntp_configure() {
    global $config, $g;

    $ntpcfg = "# \n";
    $ntpcfg .= "# pfSense OpenNTPD configuration file \n";
    $ntpcfg .= "# \n\n";

    /* foreach through servers and write out to ntpd.conf */
    foreach (explode(' ', $config['system']['timeservers']) as $ts)
        $ntpcfg .= "servers {$ts}\n";

    /* Setup listener(s) if the user has configured one */
    if ($config['installedpackages']['openntpd']) {
        /* server config is in coregui1 */
        $xmlsettings = $config['installedpackages']['openntpd']['config'][0];
        if ($xmlsettings['enable'] == 'on') {
            $ifaces = explode(',', $xmlsettings['interface']);
            $ips = array();
            foreach ($ifaces as $if) {
                if (is_ipaddr($if)) {
                    $ips[] = $if;
                } else {
                    $if = get_real_interface($if);
                    if (does_interface_exist($if))
                        $ips[] = find_interface_ip($if);
                }
            }
            foreach ($ips as $ip) {
                if (is_ipaddr($ip))
                    $ntpcfg .= "listen on $ip\n";
            }
        }
        $ntpcfg .= "\n";

        /* open configuration for writing or bail */
        $fd = fopen("${$g['varetc_path']}/ntpd.conf", "w");
        if (!$fd) {
            log_error("Could not open {$g['varetc_path']}/ntpd.conf for writing");
            return;
        }
        fwrite($fd, $ntpcfg);

        /* slurp! */
        fclose($fd);
    }
}
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
/* if openntpd is running, kill it */
while(is_process_running("ntpd")) {
killbyname("ntpd");
}

/* if /var/empty does not exist, create it */
if(!is_dir("/var/empty"))
exec("/bin/mkdir -p /var/empty && chmod ug+rw /var/empty/.");

/* start openntpd, set time now and use /var/etc/ntpd.conf */
exec("/usr/local/sbin/ntpd -s -f {${g[varetc_path]}/ntpd.conf");

// Note that we are starting up
log_error("OpenNTPD is starting up.");

}

function sync_system_time() {
global $config, $g;

if (${g['booting']})
echo "Syncing system time before startup...";

/* foreach through servers and write out to ntpd.conf */
foreach (explode(' ', $config['system']['timeservers']) as $ts) {
mwexec("/usr/sbin/ntpdate -s $ts");
}

if (${g['booting']})
echo "done.\n";

}

function system_halt() {
global $g;

system_reboot_cleanup();

mwexec("/usr/bin/nohup /etc/rc.halt > /dev/null 2>&1 &");
}

function system_reboot() {
global $g;

system_reboot_cleanup();
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
mwexec("nohup /etc/rc.reboot > /dev/null 2>&1 &");
}

function system_reboot_sync() {
global $g;

system_reboot_cleanup();

mwexec("/etc/rc.reboot > /dev/null 2>&1");
}

function system_reboot_cleanup() {
mwexec("/usr/local/bin/beep.sh stop");
require_once("captiveportal.inc");
captiveportal_radius_stop_all();
require_once("voucher.inc");
voucher_save_db_to_config();
mwexec("/etc/rc.stop_packages");
}

function system_do_shell_commands($early = 0) {
global $config, $g;
if(isset($config['system']['developerspew'])) {
$mt = microtime();
echo "system_do_shell_commands() being called $mt\n";
}

if ($early)
$cmdn = "earlyshellcmd";
else
$cmdn = "shellcmd";

if (is_array($config['system'][$cmdn])) {

/* *cmd is an array, loop through */
foreach ($config['system'][$cmdn] as $cmd) {
exec($cmd);
}

} elseif($config['system'][$cmdn] <> "") {

/* execute single item */
exec($config['system'][$cmdn]);

}
}
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
function system_console_configure() {
    global $config, $g;
    if(isset($config['system']['developerspew'])) {
        $mt = microtime();
        echo "system_console_configure() being called $mt\n";
    }

    if (isset($config['system']['disableconsolemenu']) {
        touch("${$g['varetc_path']}/disableconsole");
    } else {
        unlink_if_exists("${$g['varetc_path']}/disableconsole");
    }
}

function system_dmesg_save() {
    global $g;
    if(isset($config['system']['developerspew'])) {
        $mt = microtime();
        echo "system_dmesg_save() being called $mt\n";
    }

    $dmesg = "";
    exec("/sbin/dmesg", $dmesg);

    /* find last copyright line (output from previous boots may be present) */
    $lastcpline = 0;

    for ($i = 0; $i < count($dmesg); $i++) {
        if (strstr($dmesg[$i], "Copyright (c) 1992-"))
            $lastcpline = $i;
    }

    $fd = fopen("${$g['varlog_path']}/dmesg.boot", "w");
    if (!$fd) {
        printf("Error: cannot open dmesg.boot in system_dmesg_save().\n");
        return 1;
    }

    for ($i = $lastcpline; $i < count($dmesg); $i++)
        fwrite($fd, $dmesg[$i] . "\n");

    fclose($fd);

    return 0;
}
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
function system_set_harddisk_standby() {
    global $g, $config;
    if(isset($config['system']['developerspew'])) {
        $mt = microtime();
        echo "system_set_harddisk_standby() being called $mt\n";
    }

    if (isset($config['system']['harddiskstandby'])) {
        if ($g['booting']) {
            echo 'Setting hard disk standby... ';
        }

        $standby = $config['system']['harddiskstandby'];
        // Check for a numeric value
        if (is_numeric($standby)) {
            // Sync the disk(s)
            pfSense_sync();
            if (!mwexec('/sbin/sysctl hw.ata.standby=' . ((int)$standby))) {
                // Reinitialize ATA-drives
                mwexec('/usr/local/sbin/atareinit');
                if ($g['booting']) {
                    echo "done.\n";
                }
            } else if ($g['booting']) {
                echo "failed!\n";
            }
        } else if ($g['booting']) {
            echo "failed!\n";
        }
    }
}

function system_setup_sysctl() {
    global $config;
    if(isset($config['system']['developerspew'])) {
        $mt = microtime();
        echo "system_setup_sysctl() being called $mt\n";
    }

    activate_sysctls();

    if (isset($config['system']['sharednet'])) {
        system_disable_arp_wrong_if();
    }
}
```


Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
function system_disable_arp_wrong_if() {
    global $config;
    if(isset($config['system']['developerspew'])) {
        $mt = microtime();
        echo "system_disable_arp_wrong_if() being called $mt\n";
    }
    mwexec("/sbin/sysctl -n net.link.ether.inet.log_arp_wrong_iface=0");
    mwexec("/sbin/sysctl -n net.link.ether.inet.log_arp_movements=0");
}
```

```
function system_enable_arp_wrong_if() {
    global $config;
    if(isset($config['system']['developerspew'])) {
        $mt = microtime();
        echo "system_enable_arp_wrong_if() being called $mt\n";
    }
    mwexec("/sbin/sysctl -n net.link.ether.inet.log_arp_wrong_iface=1");
    mwexec("/sbin/sysctl -n net.link.ether.inet.log_arp_movements=1");
}
```

```
function enable_watchdog() {
    global $config;
    return;
    $install_watchdog = false;
    $supported_watchdogs = array("Geode");
    $file = file_get_contents("/var/log/dmesg.boot");
    foreach($supported_watchdogs as $sd) {
        if(stristr($file, "Geode")) {
            $install_watchdog = true;
        }
    }
    if($install_watchdog == true) {
        if(is_process_running("watchdogd"))
            mwexec("/usr/bin/killall watchdogd", true);
        exec("/usr/sbin/watchdogd");
    }
}
```

```
function system_check_reset_button() {
    global $g;
    if($g['platform'] != "nanobsd")
        return 0;

    $specplatform = system_identify_specific_platform();

    if ($specplatform['name'] != "wrap" && $specplatform['name'] != "alix")
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
return 0;

$retval = mwexec("/usr/local/sbin/" . $specplatform['name'] . "resetbtn");

if ($retval == 99) {
/* user has pressed reset button for 2 seconds -
reset to factory defaults */
echo <<<EOD

*****
* Reset button pressed - resetting configuration to factory defaults. *
* The system will reboot after this completes. *
*****

EOD;

reset_factory_defaults();
system_reboot_sync();
exit(0);
}

return 0;
}

/* attempt to identify the specific platform (for embedded systems)
Returns an array with two elements:
name => platform string (e.g. 'wrap', 'alix' etc.)
descr => human-readable description (e.g. "PC Engines WRAP")
*/
function system_identify_specific_platform() {
global $g;

if ($g['platform'] == 'generic-pc')
return array('name' => 'generic-pc', 'descr' => "Generic PC");

if ($g['platform'] == 'generic-pc-cdrom')
return array('name' => 'generic-pc-cdrom', 'descr' => "Generic PC (CD-ROM)");

/* the rest of the code only deals with 'embedded' platforms */
if ($g['platform'] != 'nanobsd')
return array('name' => $g['platform'], 'descr' => $g['platform']);

$dmesg = system_get_dmesg_boot();

if (strpos($dmesg, "PC Engines WRAP") !== false)
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

```
return array('name' => 'wrap', 'descr' => 'PC Engines WRAP');

if (strpos($dmesg, "PC Engines ALIX") !== false)
return array('name' => 'alix', 'descr' => 'PC Engines ALIX');

if (preg_match("/Soekris net45../", $dmesg, $matches))
return array('name' => 'net45xx', 'descr' => $matches[0]);

if (preg_match("/Soekris net48../", $dmesg, $matches))
return array('name' => 'net48xx', 'descr' => $matches[0]);

if (preg_match("/Soekris net55../", $dmesg, $matches))
return array('name' => 'net55xx', 'descr' => $matches[0]);

/* unknown embedded platform */
return array('name' => 'embedded', 'descr' => 'embedded (unknown)');
}

function system_get_dmesg_boot() {
global $g;

return file_get_contents("{${$g['varlog_path']}/dmesg.boot"});
}

function openntpd_get_listen_ips() {
$interfaces = get_configured_interface_with_descr();
$carplist = get_configured_carp_interface_list();
$listenips = array();
foreach ($carplist as $cif => $carpip)
$interfaces[$cif] = $carpip." (.get_vip_descr($carpip).)";
$aliaslist = get_configured_ip_aliases_list();
foreach ($aliaslist as $aliasip => $aliasif)
$interfaces[$aliasip] = $aliasip." (.get_vip_descr($aliasip).)";
foreach ($interfaces as $iface => $ifacename) {
$tmp["name"] = $ifacename;
$tmp["value"] = $iface;
$listenips[] = $tmp;
}
$tmp["name"] = "Localhost";
$tmp["value"] = "lo0";
$listenips[] = $tmp;
return $listenips;
}
?>
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

Я изменил .pa.txt на .htpasswd

Для создания под виндой нового файла .htpasswd

Загрузите утилиту htpasswd.exe (найдете в инете)

Синтаксис для создания пользователя admin:

```
C:>htpasswd -cm .htpasswd admin  
New password: *****  
Re-type new password: *****  
Adding password for user admin
```

Для изменения существующего файла:

```
htpasswd -m .htpasswd admin2
```

Автор:

29.04.11 15:56 - Последнее обновление 27.09.12 10:37

ссылка на статью: http://thin.kiev.ua/index.php?option=com_content&view=article&id=381:uyfguofcuyfufugugy&catid=50:pfsense&Itemid=81

оригинал: [тут](#)

{jcomments on}