



```
1  <?php
2      $reasonsfn = "blockreasons.html";
3      $fn = "/usr/local/info/blocked.csv";
4      $tabcell = "<small>";
5      $tabcellend = "</small>";
6      ob_start();
7  ?>
8  <HTML><HEAD><TITLE>Blocked Computers</TITLE></HEAD>
9  <BODY><CENTER>
10     <H3>Machines and ports blocked for security
reasons</H3>
11     <P>
12         <?php
13             $mtime = @filemtime($fn) or
14                 die ("$fn not found");
15             print "Data last updated " . date ("r", $
mtime) . "\n";
16         ?>
17     </P>
18     <TABLE border=2>
19         <?php
20             $fh = @fopen ($fn, "r") or
21                 die ("Error reading $fn");
22             $linenum = 0;
23             while ($data = fgetcsv ($fh, 2048)) {
24                 print ("<tr>\n");
25                 for ($i = 0; $i < count($data); $i++) {
26                     print ("  <td>$tabcell");
27                     print (strlen($data[$i]) ? htmlspecialcharschars($data[$i]) : "&nbsp;");
28                     print ("</td>\n");
29                 }
30                 print ("</tr>\n");
31             }
32             fclose ($fh);
33         ?>
34     </TABLE>
35 </CENTER>
36 <B>Reason codes:</B><BR>
```

```
37 <?php
38     include($reasonsfn);
39 ?>
40 </BODY>
41 </HTML>
42
```

[silversurfer:~/Sites/ednsCasDev] kcary% diff blocked.php castestblocked2.php

6a7,26

```
> // load the PHPCAS client library
> include_once('CAS.php');
>
> // start or load php session
> session_start();
>
> // initialize phpCAS
> phpCAS::client(SAML_VERSION_1_1, 'edns-test01.pepperdine.edu', 9443, '/cas');
>
> //setup handleLogoutRequests
> phpCAS::handleLogoutRequests(true, array( "edns-test01.pepperdine.edu", 9443, '/cas/logout' ) );
>
> //check for logout request
> if (isset($_REQUEST['logout'])) { phpCAS::logout(); }
>
> // don't validate the cas server
> phpCAS::setNoCasServerValidation();
>
> // force CAS authentication to view this page
> phpCAS::forceAuthentication();
```

10a31,48

```
> <B>-----<BR>
> <?php
> // parse the SAML_ATTRIBUTES array constructed by the PHPCAS client
> if (!isset($_SESSION[SAML_ATTRIBUTES])) {
>     echo "Can't Find Attributes in Session!";
> } else {
>     $attrs = $_SESSION[SAML_ATTRIBUTES];
>     $gn = $attrs['givenName'];
>     foreach ($gn as $firstname) {
>     }
>     $surname = $attrs['sn'];
>     foreach ($surname as $lastname) {
>     }
> }
> // do something wtih the values you find
> echo "Welcome " . $firstname . " " . $lastname . "<br>";
> ?>
> -----</B><BR>
```


```
1  <?php
2      $reasonsfn = "blockreasons.html";
3      $fn = "/usr/local/info/blocked.csv";
4      $tabcell = "<small>";
5      $tabcellend = "</small>";
6      ob_start();
7      // load the PHPCAS client library
8      include_once('CAS.php');
9
10     // start or load php session
11     session_start();
12
13     // initialize phpCAS
14     phpCAS::client(SAML_VERSION_1_1, 'edns-
test01.pepperdine.edu', 9443, '/cas');
15
16     //setup handleLogoutRequests
17     phpCAS::handleLogoutRequests(true, array( "edns-
test01.pepperdine.edu", 9443, '/cas/logout' ) );
18
19     //check for logout request
20     if (isset($_REQUEST['logout'])) { phpCAS::logout(
); }
21
22     // don't validate the cas server
23     phpCAS::setNoCasServerValidation();
24
25     // force CAS authentication to view this page
26     phpCAS::forceAuthentication();
27  ?>
28  <HTML><HEAD><TITLE>Blocked Computers</TITLE></HEAD>
29  <BODY><CENTER>
30      <H3>Machines and ports blocked for security
reasons</H3>
31      <B>-----<BR>
32      <?php
33          // parse the SAML_ATTRIBUTES array
constructed by the PHPCAS client
34          if (!isset($_SESSION[SAML_ATTRIBUTE
```

```

S])) {
35         echo "Can't Find Attributes in
Session!";
36     } else {
37         $attrs = $_SESSION[SAML_ATTRIB
UTES];
38         $gn = $attrs["givenName"];
39         foreach ($gn as $firstname) {
40             }
41         $surname = $attrs["sn"];
42         foreach ($surname as $lastname
) {
43             }
44         }
45         // do something wtih the values you find
46         echo "Welcome " . $firstname . " " . $lastn
ame . "<br>";
47         ?>
48         -----</B><BR>
49         <P>
50         <?php
51             $mtime = @filemtime($fn) or
52                 die ("$fn not found");
53             print "Data last updated " . date ("r", $
mtime) . "\n";
54         ?>
55         </P>
56         <TABLE border=2>
57         <?php
58             $fh = @fopen ($fn, "r") or
59                 die ("Error reading $fn");
60             $linenum = 0;
61             while ($data = fgets ($fh, 2048)) {
62                 print ("<tr>\n");
63                 for ($i = 0; $i < count($data); $i++) {
64                     print (" <td>$tabcell");
65                     print (strlen($data[$i]) ? htmlspecialcharschars($data[$i]) : "&nbsp;");
66                     print ("</td>\n");

```

```
67         }
68         print ("</tr>\n");
69     }
70     fclose ($fh);
71     ?>
72     </TABLE>
73 </CENTER>
74 <B>Reason codes:</B><BR>
75 <?php
76     include ($reasonsfn) ;
77 ?>
78 </BODY>
79 </HTML>
80
```

```
1  <?php 
2  $reasonsfn = "blockreasons.html";
3  $fn = "/usr/local/info/blocked.csv";
4  $tabcell = "<small>";
5  $tabcellend = "</small>";
6  ob_start();
7  ?>
8  <HTML>
9  <HEAD>
10 <TITLE>Unblock Page!</TITLE>
11 <META http-equiv="Content-Type" content="text/html;
12 charset=iso-8859-1"></HEAD>
13 <BODY bgcolor="#FFCCFF">
14 <H2><FONT face="Verdana, Arial, Helvetica, sans-
15 serif">Unblock Page!
16 </FONT></H2>
17 <P><FONT face="Verdana, Arial, Helvetica, sans-
18 serif">
19 <?php //HTTP Basic Auth parameter from Apache
20 print "You are: " . $_SERVER['REMOTE_USER'] . "!
21 \n<br>";
22 ?>
23 </FONT><FONT face="Arial, Helvetica, sans-serif">
24 <P><FONT face="Verdana, Arial, Helvetica, sans-
25 serif">
26 <?php
27 $mtime = @filemtime($fn) or
28 die ("$fn not found");
29 print "Data last updated " . date ("r", $mtime) . "\
30 n<br>";
31 ?>
32 </FONT>
33 <TABLE border=2>
34 <?php
35
36
37
38
39
40
41 // csvjoinq (array) return string of elements of
42 array separated by commas.
43 // Elements containing special characters are
```

```

enclosed in quotes.
33 // Embedded quotes are doubled, e.g. Say "hello" now
-> "Say ""hello"" now"
34 // By John Vannoy 5/21/2009
35 function csvjoin ($stringarray) {
36     $qlist = $stringarray;
37     for ($i = 0; $i < count($qlist); $i++) {
38         if (preg_match ("/[^\-._0-9A-Za-z]/", $qlist[$i])
39 ) {
40             $qlist[$i] = "'" . str_replace ('"', "'", $ql
41 ist[$i]) . "'";
42         }
43     }
44     return join (',', $qlist);
45 }
46 $fh = @fopen ($fn, "r") or
47     die ("Error reading $fn");
48 $linenum = 0;
49 while ($data = fgetcsv ($fh, 2048)) {
50     // if (strlen($data[0] == 0 || strlen($data[1] >
51 0) { continue; }
52     print ("|\n");
53     for ($i = 0; $i < count($data); $i++) {
54         print ("  <td>$tabcell");
55         print (strlen($data[$i]) ? htmlspecialchars($dat
56 a[$i]) : "&nbsp;");
57         print (" $tabcellend</td>\n");
58     }
59     // Final column in "Unblock" button.
60     print ("  <td>");
61     if (($linenum != 0) &&
62         ( (preg_match("/^Blocked MAC at/", $data[2]) ||
63         ( preg_match("/^Blocked IP at/", $data[2]) ) ) ) &&
64         (! preg_match("/^Blocked MAC at Vernier/i", $da
65 ta[2])) ) &&
66         (! preg_match("/!/!", $data[3])) ) {
67         // Form with unblock button

|  |

```



```

64     print ('<form action="/unblocked.php"
method="post" enctype="application/x-www-form-urlencoded
' . $linenum . '">');
65     $blockdata = csvjoinq ($data); // Recreate
original line from blocked.csv
66     //$unblockdata = "0,USERID,unblock,\"$data
[0]\",$data[4],$data[5],$data[6]";
67     print ('<input name="blockdata" type="hidden"
value="' . htmlspecialchars($blockdata) . '">');
68     //print ('<input name="mac" type="hidden"
value="'0,USERID,unblock,'" . $data[0] . ',' . $data
69     print ('<input name="signature" type="hidden"
value="0">');
70     print ('<input name="userid" type="hidden"
value="' . htmlspecialchars($_SERVER[REMOTE_USER]) .
'">');
71     print ('<input name="request" type="hidden"
value="unblock">');
72     print ('<input name="time_blocked"
type="hidden" value="' . htmlspecialchars($data[0])
. '">');
73     print ('<input name="time_unblocked"
type="hidden" value="' . htmlspecialchars($data[1])
. '">');
74     print ('<input name="block_status"
type="hidden" value="' . htmlspecialchars($data[2])
. '">');
75     print ('<input name="block_heat" type="hidden"
value="' . htmlspecialchars($data[3]) . '">');
76     print ('<input name="reason" type="hidden"
value="' . htmlspecialchars($data[4]) . '">');
77     print ('<input name="ip" type="hidden" value="
' . htmlspecialchars($data[5]) . '">');
78     print ('<input name="mac" type="hidden"
value="' . htmlspecialchars($data[6]) . '">');
79     print ('<input name="wins" type="hidden"
value="' . htmlspecialchars($data[7]) . '">');
80     print ('<input name="Unblock" type="submit"

```

```
value="Unblock"></form>');
81     } else {
82         // No unblock button
83         print ("&nbsp;");
84     }
85     print ("</td>\n");
86     print ("</tr>\n");
87     $linenum++;
88 }
89 fclose ($fh);
90 ?>
91 </TABLE>
92 <P><FONT face="Verdana, Arial, Helvetica, sans-
93 serif">
94 <?php
95 print ("<a href=\"\" . htmlspecialchars($fn) . "\">Ra
96 w CSV file</a>\n");
97 ?>
98 </FONT>
99 <P><B>Reason codes:</B><BR>
100 <?php include($reasonsfn);
101 ?>
102 </BODY>
</HTML>
```

6a7,26

```
> // load the PHPCAS client library
> include_once('CAS.php');
>
> // start or load php session
> session_start();
>
> // initialize phpCAS
> phpCAS::client(SAML_VERSION_1_1, 'edns-test01.pepperdine.edu', 9443, '/cas');
>
> //setup handleLogoutRequests
> phpCAS::handleLogoutRequests(true, array( "edns-test01.pepperdine.edu", 9443, '/cas/logout' ) );
>
> //check for logout request
> if (isset($_REQUEST['logout'])) { phpCAS::logout(); }
>
> // don't validate the cas server
> phpCAS::setNoCasServerValidation();
>
> // force CAS authentication to view this page
> phpCAS::forceAuthentication();
```

13c33,55

```
< <H2><FONT face="Verdana, Arial, Helvetica, sans-serif">Unblock Page!
```

```
---
> <?php //Parse the attribute array created by PHPCAS client
>     if (!isset($_SESSION[SAML_ATTRIBUTES])) {
>         echo "Can't Find Attributes in Session!";
>     } else {
>         $attrs = $_SESSION[SAML_ATTRIBUTES];
>         $authorized = "not authorized" ;
>         $mo = $attrs["memberOf"];
>         foreach ($mo as $groupdn) {
>             if ($groupdn == "CN=grp InfoSec,CN=Users,DC=pepperdine,DC=ad,DC=pepperdine,DC=edu") {
>                 $authorized = "authorized";
>             }
>         }
>         $gn = $attrs["givenName"];
>         foreach ($gn as $firstname) {
>             //echo "Welcome " . $firstname . " ";
>         }
>         $surname = $attrs["sn"];
>         foreach ($surname as $lastname) {
>             //echo $lastname . "<br>";
>         }
>     }
> }
```

```
> ?>
> <H2><FONT face="Verdana, Arial, Helvetica, sans-serif">Unblock Page! </FONT><FONT face="Verdana, Arial, Helvetica, sans-ser
if">
```

15,17c57,59

```
< <P><FONT face="Verdana, Arial, Helvetica, sans-serif">
< <?php //HTTP Basic Auth parameter from Apache
< print "You are: " . $_SERVER['REMOTE_USER'] . "!\n<br>";
```

```
---
> <FONT face="Verdana, Arial, Helvetica, sans-serif">
> <?php //CAS Authorization Check Result
> print "$firstname $lastname, you are " . $authorized . " to unblock computers!\n<br>";
```

21c63

```
< <?php
```

```
---
```

```
> <?php
```

28c70

```
< <?php
```

```
---
```

```
> <?php
```

57c99,101

```
< // Final column in "Unblock" button.
```

```
> // Final column in "Unblock" button.
> // Check authorized flag set by this script
> // based on attributes collected by PHPCAS
61c105
< (! preg_match("/^Blocked MAC at Vernier/i",$data[2])) &&
---
> ($authorized == "authorized") &&
63,64c107,108
< // Form with unblock button
< print ('<form action="/unblocked.php" method="post" enctype="application/x-www-form-urlencoded" name="unblk' . $linenum . "'>');
---
> // Form with unblock button also PHPCAS protected
> print ('<form action="/castestUnblocked.php" method="post" enctype="application/x-www-form-urlencoded" name="unblk' . $linenum . "'>');
```



```
1  <?php
2  // load the PHPCAS client library
3  include_once('CAS.php');
4
5  // start or load php session
6  session_start();
7
8  // initialize phpCAS
9  phpCAS::client(SAML_VERSION_1_1, 'edns-
10 test01.pepperdine.edu', 9443, '/cas');
11
12 //setup handleLogoutRequests
13 phpCAS::handleLogoutRequests(true, array( "edns-
14 test01.pepperdine.edu", 9443, '/cas/logout' ) );
15
16 //check for logout request
17 if (isset($_REQUEST['logout'])) { phpCAS::logout(); }
18
19 // don't validate the cas server
20 phpCAS::setNoCasServerValidation();
21
22 // force CAS authentication to view this page
23 phpCAS::forceAuthentication();
24
25 ?>
26 <HTML>
27 <HEAD>
28 <TITLE>Unblock Results! - Authorized</TITLE>
29 <META http-equiv="Content-Type" content="text/html;
30 charset=iso-8859-1">
31 </HEAD>
32 <BODY bgcolor="#66FF66">
33 <?php
34     if (!isset($_SESSION[SAML_ATTRIBUTES])) {
35         echo "Can't Find Attributes in Session!"
36     ;
37     } else {
38         $attrs = $_SESSION[SAML_ATTRIBUTES];
39         $authorized = "not authorized" ;
```

```

36         $mo = $attrs["memberOf"];
37         foreach ($mo as $groupdn) {
38             if ($groupdn == "CN=grp
InfoSec,CN=Users,DC=pepperdine,DC=ad,DC=pepperdine,DC=ec
) {
39                 $authorized = "authorized";
40             }
41             $gn = $attrs["givenName"];
42             foreach ($gn as $firstname) {
43                 //echo "Welcome " .
$firstname . " ";
44             }
45             $surname = $attrs["sn"];
46             foreach ($surname as $lastname) {
47                 //echo $lastname . "<br>";
48             }
49         }
50     }
51 }>
52 <?php if ($authorized == "authorized") { ?>
53 <H2><FONT face="Verdana, Arial, Helvetica, sans-
serif">Unblock Results for <?php echo $firstname . "
" . $lastname ?></FONT></H2>
54 <P><FONT face="Verdana, Arial, Helvetica, sans-serif"
>Please examine the data below. Your should see 7
fields, including the IP
55 Address and MAC address of the block you were
trying to clear.</FONT></P>
56 <FONT face="Verdana, Arial, Helvetica, sans-serif">
57
58 <TABLE>
59 <?php
60     print ('<tr><td>Signature (future) : </td><td>' . $
_POST['signature'] . '</td></tr>');
61     print ('<tr><td>Your userid: </td><td>' . $_POST['u
serid'] . '</td></tr>');
62     print ('<tr><td>Request: </td><td>' . $_POST['reque
st'] . '</td></tr>');
63     print ('<tr><td>Time Blocked: </td><td>' . $_POST['

```

```

time_blocked'] . '</td></tr>');
64   print ('<tr><td>Reason: </tc><td>' . $_POST['reason
    ']' . '</td></tr>');
65   print ('<tr><td>IP Address: </tc><td>' . $_POST['ip
    ']' . '</td></tr>');
66   print ('<tr><td>Mac Address: </tc><td>' . $_POST['m
    ac'] . '</td></tr>');
67   ?>
68 </TABLE>
69 <?php
70 $unblockdata = ( $_POST[signature] . ',' . $_POST[use
    rid] . ',' . $_POST[request] . ',' . $_POST['time_
    blocked'] . ',' . $_POST['reason'] . ',' . $_POST['i
    p'] . ',' . $_POST['mac'] );
71 //print $unblockdata;
72 //system('/usr/bin/logger -p local7.notice -t
    netblock ' . escapeshellarg( $unblockdata ) );
73
74 ?>
75 <P>If this data is correct, the block should be
    cleared within ten minutes.</P>
76 </FONT>
77 <HR>
78     <?php } else { ?>
79 <H2>
80 <?php echo $firstname . " " . $lastname . ", you are
    not authorized to unblock!";
81     }
82     ?>
83 </H2>
84 <FONT face="Verdana, Arial, Helvetica, sans-serif"><P
    ><FONT size="2">Please report errors on this page to
    Kim Cary and John Vannoy.</FONT></P>
85 </FONT>
86 </BODY>
87 </HTML>
88

```