

[NEWS] Multiple Firewalls Ruleset Bypass through FTP Revisited

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-10/0036.html>

From: support@securiteam.com

Date: 10/09/02

From: support@securiteam.com

To: list@securiteam.com

Date: 9 Oct 2002 11:24:50 +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

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

Multiple Firewalls Ruleset Bypass through FTP Revisited

SUMMARY

A few years ago, a new attack method affected most leading firewall products and used the nature of dynamic ports.

The attack consisted of establishing a control channel connection with an FTP server behind the firewall and forcing it to send back a response that resembles a data channel command. The firewall would then add a dynamic rule to allow connection to that newly created data port for FTP data communication, and the attacker would be able to use this mechanism to connect to a restrictive port.

A new report by Mikael Olsson of <http://www.clavister.com> Clavister shows many firewalls are still vulnerable to a similar attack whose impact may be bypassing the firewall rules and accessing ports that should be blocked by the firewall.

DETAILS

This attack resembles the techniques used in early 2000 in that it forces the trusted point to output fake data channel commands in the control channel, although the approach is slightly different.

Securiteam: [NEWS] Multiple Firewalls Ruleset Bypass through FTP Revisited

This attack uses partial segment acknowledgement to cause the victim to resend control strings supplied by the attacker that a vulnerable firewall will erroneously parse as a legitimate command.

It also theoretically bypasses some of the bandaids put in place to protect against the attacks from 2000. Those attacks resulted in packets in the command channel that weren't CRLF terminated.

This one doesn't.

Attacking a server on port 5000:

Attacker: Connect to FTP server and log on (anonymously?)

Attacker: "RETR 227 Connect to me at (192,168,0,10,19,136)\r\n"

Server : "5xx No such file: 227 Connect to me at ([...])\r\n"

Attacker: Sends TCP ACK for the (here) 18 first bytes, i.e.

"5xx No such file: ", NOT the whole packet, which leaves the "227 Connect to me" string lying in the send queue of the server. This will soon be retransmitted:

Server : "227 Connect to me at (192,168,0,10,19,136)\r\n"

Notes:

The actual commands may vary depending on the FTP server. Also, please keep in mind that this is not a vulnerability in the FTP server, but rather a problem with the firewall.

Also note that all strings sent are properly CRLF terminated.

At this point, a vulnerable firewall will pick up the "227.." string and open an inbound hole to port 5000 on the vulnerable system.

Mikael Olsson notes he specifically chose a private IP in the example, because this is not a problem for the attacker: the firewall will handle address translation and send back a string containing the public IP and a different port. All one needs to do is connect to the public IP and the given port, and you'll end up on port 5000 of the server.

This does however require that you know the private IP beforehand, unless the firewall is unusually broken. A very broken firewall on the other hand might even allow you to open connections to systems other than the FTP server itself.

Attacking a client on port 5000:

This requires getting the client to attempt to retrieve a path of your choosing. This is most easily done through a web page or HTML mail with an appropriate URL / IMG SRC, e.g.:

 and having the server despool part of the incoming RETR request, which would cause the client to send out "PORT 192,...".

Again, this depends on knowing the client IP beforehand, unless the firewall is very broken and lets you open connections to any host behind the firewall.

Securiteam: [NEWS] Multiple Firewalls Ruleset Bypass through FTP Revisited

Note that the above depends on the fact that the victim TCP stack will despool partial segments. Most do – apparently only the TCP stack used in recent Linux kernels does not.

This means that Linux-based FTP servers/clients cannot themselves be used to coerce the firewalls protecting them.

This technique can very likely be abused for other protocols that use ephemeral data channels. Examples of such protocols (exploitable or not) include but are not limited to: IRC DCC, SQL*Net, H.323, SIP, Real Audio, etc.

Mikael Olsson also mentions that this may be further extended / modified to affect some firewalls that were not vulnerable to the first incarnation.

ADDITIONAL INFORMATION

The above technique was cooked up by <mailto:mikael.olsson@clavister.com> Mikael Olsson, who thanks ICSA labs for taking the time to verify it against their certified products.

This is also documented as a CERT vulnerability note
<<http://www.kb.cert.org/vuls/id/328867>>

<http://www.kb.cert.org/vuls/id/328867>

However, the current revision (53) of the vuln note talks about SACK options, which is inaccurate. No SACKs are used.

=====

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.

-
- **Previous message:** support@securiteam.com: "[EXPL] Windows Help Buffer Overflow PoC"
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)