

pfSense - Bug #3500

DHCP Leases List Not Showing Hostname in Some Cases

03/04/2014 04:42 PM - David Justl

Status:	Assigned	Start date:	03/04/2014
Priority:	Normal	Due date:	
Assignee:	Renato Botelho	% Done:	100%
Category:	Web Interface	Estimated time:	0.00 hour
Target version:	2.5.0		
Affected Version:	All	Affected Architecture:	

Description

The DHCP leases list shows a blank in the hostname field for some leases, even though the log shows that a hostname was received at registration. The hostnames also do appear in the `/var/dhcpd/var/db/dhcpd.leases` file, which is parsed to create the GUI list. The problem appears to be in the parsing logic, where the record separator is set to be a close brace, `}`. In some cases, a record may include a `'uid'` value that contains a close brace:

```
lease 192.168.1.10 {
starts 2 2014/03/04 21:02:04;
ends 2 2014/03/04 22:02:04;
cltt 2 2014/03/04 21:02:04;
binding state active;
next binding state free;
rewind binding state free;
hardware ethernet 00:11:22:33:44:55;
uid "\001\020x\322\367\333}";
client-hostname "MyHost";
}
```

This breaks the parsing for that record and causes the lease list to show a blank hostname.

Associated revisions

Revision 88326a6b - 06/04/2019 08:33 AM - Renato Botelho

Ticket #3500: Implement `system_get_dhcpleases()`

Revision c4c730b5 - 06/04/2019 08:33 AM - Renato Botelho

Ticket #3500: Simplify `status_dhcp_leases.php` using `system_get_dhcpleases()`

Revision 55342b60 - 06/04/2019 08:33 AM - Renato Botelho

Ticket #3500: Simplify `diag_arp.php` using `system_get_dhcpleases()`

Revision 00e3e638 - 06/04/2019 08:33 AM - Renato Botelho

Ticket #3500: Simplify `dhcpd_gather_stats.php` using `system_get_dhcpleases()`

History

#1 - 03/05/2014 01:23 AM - Phillip Davis

Had a quick look at this. The relevant original code in `/usr/local/www/status_dhcp_leases.dhcp` is:

```
$splitpattern = "BEGIN { RS=\\}"; for (i=1; i<=NF; i++) printf "%s \", \\$i; printf "\\n\\}";
```

That splits records based on any `}` in the file. Not so good when `}` appears somewhere in the data.

I thought matching `}` plus `\n` would fix it:

```
$splitpattern = "BEGIN { RS=\\}\\n"; for (i=1; i<=NF; i++) printf "%s \", \\$i; printf "\\}\\n";
```

or looking for `\n` first:

```
$splitpattern = "BEGIN { RS=\\}\\n"; for (i=1; i<=NF; i++) printf "%s \", \\$i; printf "\\}\\n";
```

or looking for "\n" then "]" then "\n":

```
$splitpattern = "BEGIN { RS=\"\\n\\n\"; } {for (i=1; i<=NF; i++) printf \"%s \", $i; printf \"}\\n\";}";
```

but everything I did seemed to only use the first char in RS.

The pfSense/FreeBSD "awk" seems to only take notice of the first char in RS. I tried to use "/usr/bin/gawk" - that should allow multiple chars in RS - but it is not anywhere to be found on pfSense distributions.

It could all be done in a PHP loop instead of using "awk". But that will use more CPU. If there is some other easy way to fix up the "awk" code that would be good.

#2 - 09/15/2015 02:08 AM - Chris Buechler

- Status changed from New to Confirmed

- Affected Version changed from 2.1 to All

#3 - 12/31/2018 04:35 PM - Totio Katunio

This bug with the } symbol in some UID fields and the missing hostname in the Web UI is still presented in version 2.4.4-RELEASE-p1 I created a small fix for this particular issue - PR: <https://github.com/pfsense/pfsense/pull/4027>

#4 - 01/01/2019 08:45 AM - Joshua Sign

Phillip Davis wrote:

It could all be done in a PHP loop instead of using "awk".

It should be better and easier, for sure. A simple file_get_contents(dhcpd.leases) and a loop on each lines can do the job. I really don't know why using an exec...

@Totio Katunio :

According to the man page about 'uid' field : it can contains any printable chars or octal representation of non-printables chars :

The client identifier is recorded as a colon-separated hexadecimal list or as a quoted string. If it is recorded as a quoted string and it contains one or more non-printable characters, those characters are represented as octal escapes - a backslash character followed by three octal digits.

So this issue is about a '}' but your solution implies usage of '#' that could also be in the uid field, isn't it ? Not sure about that, but a solution that not depend about the chars in any field may be safer.

I just test some commands line to get the same output as expected in normal behavior, wich is this output for my examples :

```
cat test_leases.txt | awk '{ gsub("#.*", ""); } { gsub(";", ""); print;}' | awk 'BEGIN{ RS=""}; } {for (i=1; i<=NF; i++) printf "%s ", $i; printf "}\n";}'
lease 192.168.0.229 { starts 1 2011/09/1 14:01:31 ends 2 2011/09/2 14:01:31 tstp 4 2011/09/4 14:00:00 binding
state active next binding state free hardware ethernet 00:0c:76:8b:c5:20 uid "\002\000\014\213\304\026" client
-hostname "paul" }
lease 192.168.0.223 { starts 3 2011/08/3 12:03:29 ends 4 2011/08/4 12:03:29 tstp 6 2011/08/6 12:00:00 binding
state active next binding state free hardware ethernet 00:07:E9:74:7D:78 uid "\021\010\014\213\304\026" client
-hostname "jeff" }
lease 192.168.0.213 { starts 5 2011/08/5 08:03:11 ends 6 2011/09/6 09:03:11 tstp 8 2011/09/8 09:00:00 binding
state active next binding state free hardware ethernet 00:07:E9:74:7D:81 uid "\001\010\014\213\104\026" client
-hostname "steve" }
lease 192.168.0.224 { starts 2 2004/09/2 14:01:31 ends 3 2005/09/3 14:01:31 tstp 5 2006/09/5 23:17:32 binding
state active next binding state free hardware ethernet 22:0c:56:8b:c4:01 uid "\001\010\024\213\304\026" client
```

```
-hostname "tim" }
}
```

I introduce the '}' and the '#' chars in some uids.
And finally compare results :

Original code : break output by the '}' and the '#'

```
cat test_leases.txt | awk '{ gsub("#.*", ""); } { gsub(";", ""); print;}' | awk 'BEGIN{ RS=""}; {for (i=1; i<=
NF; i++) printf "%s ", $i; printf "}\n";}'
lease 192.168.0.229 { starts 1 2011/09/1 14:01:31 ends 2 2011/09/2 14:01:31 tstp 4 2011/09/4 14:00:00 binding
state active next binding state free hardware ethernet 00:0c:76:8b:c5:20 uid "\002\000\014\213\304\026" client
-hostname "paul" }
lease 192.168.0.223 { starts 3 2011/08/3 12:03:29 ends 4 2011/08/4 12:03:29 tstp 6 2011/08/6 12:00:00 binding
state active next binding state free hardware ethernet 00:07:E9:74:7D:78 uid "\021\010\014\213\304\026 }
" client-hostname "jeff" }
lease 192.168.0.213 { starts 5 2011/08/5 08:03:11 ends 6 2011/09/6 09:03:11 tstp 8 2011/09/8 09:00:00 binding
state active next binding state free hardware ethernet 00:07:E9:74:7D:81 uid "\001\010\014 client-hostname "st
eve" }
lease 192.168.0.224 { starts 2 2004/09/2 14:01:31 ends 3 2005/09/3 14:01:31 tstp 5 2006/09/5 23:17:32 binding
state active next binding state free hardware ethernet 22:0c:56:8b:c4:01 uid "\001\010\024\213\304\026" client
-hostname "tim" }
```

Your workaround : broken by '#' char only

```
cat test_leases.txt | awk '{ gsub("#.*", ""); } { gsub(";", ""); print;}' | sed 's/^}/#/g' | awk 'BEGIN{ RS="#"
}; {for (i=1; i<=NF; i++) printf "%s ", $i; printf "}\n";}'
lease 192.168.0.229 { starts 1 2011/09/1 14:01:31 ends 2 2011/09/2 14:01:31 tstp 4 2011/09/4 14:00:00 binding
state active next binding state free hardware ethernet 00:0c:76:8b:c5:20 uid "\002\000\014\213\304\026" client
-hostname "paul" }
lease 192.168.0.223 { starts 3 2011/08/3 12:03:29 ends 4 2011/08/4 12:03:29 tstp 6 2011/08/6 12:00:00 binding
state active next binding state free hardware ethernet 00:07:E9:74:7D:78 uid "\021\010\014\213\304\026}" clien
t-hostname "jeff" }
lease 192.168.0.213 { starts 5 2011/08/5 08:03:11 ends 6 2011/09/6 09:03:11 tstp 8 2011/09/8 09:00:00 binding
state active next binding state free hardware ethernet 00:07:E9:74:7D:81 uid "\001\010\014 client-hostname "st
eve" }
lease 192.168.0.224 { starts 2 2004/09/2 14:01:31 ends 3 2005/09/3 14:01:31 tstp 5 2006/09/5 23:17:32 binding
state active next binding state free hardware ethernet 22:0c:56:8b:c4:01 uid "\001\010\024\213\304\026" client
-hostname "tim" }
```

A better workaround should be this command line, wich output seems to be as expected :

```
cat test_leases.txt | sed '/^#/ d' | sed '/^$/ d' | perl -ne 's/^(.+)\n$/\1 /g; print;' | tr ' ' ' ' | perl -
ne 's/\s\s+/ /g; print;' ; echo " " ;
lease 192.168.0.229 { starts 1 2011/09/1 14:01:31 ends 2 2011/09/2 14:01:31 tstp 4 2011/09/4 14:00:00 binding
state active next binding state free hardware ethernet 00:0c:76:8b:c5:20 uid "\002\000\014\213\304\026" client
-hostname "paul" }
lease 192.168.0.223 { starts 3 2011/08/3 12:03:29 ends 4 2011/08/4 12:03:29 tstp 6 2011/08/6 12:00:00 binding
state active next binding state free hardware ethernet 00:07:E9:74:7D:78 uid "\021\010\014\213\304\026}" clien
t-hostname "jeff" }
lease 192.168.0.213 { starts 5 2011/08/5 08:03:11 ends 6 2011/09/6 09:03:11 tstp 8 2011/09/8 09:00:00 binding
state active next binding state free hardware ethernet 00:07:E9:74:7D:81 uid "\001\010\014#\213\104\026" clien
t-hostname "steve" }
lease 192.168.0.224 { starts 2 2004/09/2 14:01:31 ends 3 2005/09/3 14:01:31 tstp 5 2006/09/5 23:17:32 binding
state active next binding state free hardware ethernet 22:0c:56:8b:c4:01 uid "\001\010\024\213\304\026" client
-hostname "tim" }
```

I use perl because i can't achieve this with awk regexp using crlf.
Unfortunately, i don't use dhcp over my pfsense actually, and i can't really test it, but the line 120 of the file '/usr/local/www/status_dhcp_leases.php'
can be replaced by something like :

```
exec("/bin/cat {$leasesfile} | /usr/bin/sed '/^#/ d' | /usr/bin/sed '/^\$/ d' | /usr/local/bin/perl -ne 's/^(.
.+)\n\$/\$/ $1 /g; print;' | /usr/bin/tr ';' ' ' | /usr/local/bin/perl -ne 's/\\s\\s+/ /g; print;' ; echo \"}\";
", $leases_content);
```

Corresponding in my command line above.

Regards
Josh_

#5 - 01/02/2019 12:06 PM - Totio Katunio

Well, I agree, but at least my proposal doesn't break the overall logic, the current code base already strips the # symbols, before I reuse it for the lease blocks, so it is not a regression, but an improvement (still not a complete one). This was the fastest/simplest way to fix it, without using too much more magic :)

Your solution is better, but I guess that we should test it directly on the pfSense host, since the tools there are pretty outdated (awk is a 2012 build binary).

PS: Also introducing perl, makes the pipeline execution slower but I tested it and it works ok. Maybe it will be a bit noticeable with more leases.

PS2: The usage of tr is also unsafe in the same regard as the # cleanup (';' is also a possible char for the uid)

#6 - 01/03/2019 05:11 AM - Joshua Sign

Hi Totio,

As i didn't find a way to correctly handle '\r' or '\n' with awk or sed, i switch to perl which can play with complex regexp, even if its a little bit slower (need to be tested in real conditions), at least the results are corrects.

You are right to point the problem about tr usage and the possible usage of ';' char in uid field : i just missed it.

Here is a new version of the command line : 'tr' part removed and taking account that only ';' followed by '\n' will be deleted :

```
cat test_leases.txt | sed '/^#/ d' | sed '/^\$/ d' | perl -ne 's/^(.+)({}|;)+\n\$/\$/ $1$3 /g; print;' | perl -ne 's/\\s\\s+/ /g; print;' ; echo "}" :
```

About the time to process, i put 4780 entries in my test_leases.txt :

- perl : about 0.218 seconds

- awk : about 0.124 seconds

So you are right : using perl is slower than awk.

But is it really significant for common usages ?

I supposed that many people don't have so much dhcp leases entries, but most probably 20 times less.

#7 - 01/04/2019 08:44 AM - Totio Katunio

Hi Joshua,
ok, so do you want me to close the current PR. I guess that you can make new PR with your implementation?

#8 - 01/04/2019 10:17 AM - Joshua Sign

I am not able to test this because i dont use this service.
I was just trying to help about this subject to find the best way to avoid furthers errors.

I think you should keep the PR and wait opinions of managers like Jim.
They will give you their advice and you'll be able to adapt the PR.

Regards

#9 - 06/04/2019 08:31 AM - Renato Botelho

- Status changed from Confirmed to Feedback
- Assignee set to Renato Botelho
- Target version set to 2.5.0
- % Done changed from 0 to 100

I've reimplemented the function that parses dhcpd.leases

#10 - 06/10/2019 07:36 AM - Jim Pingle

- Status changed from Feedback to Assigned

Looks OK for the most part, though I do have one weird device that doesn't match in the leases database vs what is displayed in the GUI.

GUI shows the hostname as "Nintendo.<my domain>" but the lease DB has:

```
lease x.x.x.x {  
  [...]  
  client-hostname "Nintendo 3DS";  
}
```

Since a space isn't valid in a hostname, I'm not sure what the best course of action is here. Using only the first word seems like it could be problematic. Perhaps replace spaces with dashes or remove them.