Subject: [RFC PATCH ext3/ext4] orphan list corruption due bad inode
Posted by vaverin on Mon, 04 Jun 2007 05:19:10 GMT
View Forum Message <> Reply to Message

After ext3 orphan list check has been added into ext3_destroy_inode() (please see my previous patch) the following situation has been detected:
EXT3-fs warning (device sda6): ext3_unlink: Deleting nonexistent file (37901290), 0
Inode 00000101a15b7840: orphan list check failed!
00000773 6f665f00 74616d72 00000573 65725f00 06737270 66000000 616d726f
...
Call Trace: [<ffffffff80211ea9>] ext3_destroy_inode+0x79/0x90
[<ffffffff801a2b16>] sys_unlink+0x126/0x1a0
[<ffffffff80111479>] error_exit+0x0/0x81
[<ffffffff80110aba>] system_call+0x7e/0x83

First messages said that unlinked inode has i_nlink=0, then ext3_unlink() adds this inode into orphan list.

Second message means that this inode has not been removed from orphan list. Inode dump has showed that i_fop = &bad_file_ops and it can be set in make_bad_inode() only. Then I've found that ext3_read_inode() can call make_bad_inode() without any error/warning messages, for example in the following case:

```
...
        if (inode->i_nlink == 0) {
                if (inode->i_mode == 0 ||
                    !(EXT3_SB(inode->i_sb)->s_mount_state & EXT3_ORPHAN_FS)) {
                        /* this inode is deleted */
                        brelse (bh);
                        goto bad_inode;
...
```

Bad inode can live some time, ext3_unlink can add it to orphan list, but ext3_delete_inode() do not deleted this inode from orphan list. As result we can have orphan list corruption detected in ext3_destroy_inode().

However it is not clear for me how to fix this issue correctly.

As far as i see is_bad_inode() is called after iget() in all places excluding ext3_lookup() and ext3_get_parent(). I believe it makes sense to add bad inode check to these functions too and call iput if bad inode detected.

Signed-off-by: Vasily Averin <vvs@sw.ru>

```
diff --git a/fs/ext3/namei.c b/fs/ext3/namei.c
index 9bb046d..e3ac8c3 100644
--- a/fs/ext3/namei.c
+++ b/fs/ext3/namei.c
@@ -1019,6 +1019,11 @@ static struct dentry *ext3_lookup(struct inode * dir, struct dentry
```

```
*dentry, str

   if (!inode)
    return ERR_PTR(-EACCES);
+
+  if (is_bad_inode(inode)) {
+   iput(inode);
+   return ERR_PTR(-ENOENT);
+  }
  }
  return d_splice_alias(inode, dentry);
 }
@@ -1054,6 +1059,11 @@ struct dentry *ext3_get_parent(struct dentry *child)
  if (!inode)
   return ERR_PTR(-EACCES);

+ if (is_bad_inode(inode)) {
+  iput(inode);
+  return ERR_PTR(-ENOENT);
+ }
+
  parent = d_alloc_anon(inode);
  if (!parent) {
   iput(inode);
diff --git a/fs/ext4/namei.c b/fs/ext4/namei.c
index 4ec57be..70db358 100644
--- a/fs/ext4/namei.c
+++ b/fs/ext4/namei.c
@@ -1017,6 +1017,11 @@ static struct dentry *ext4_lookup(struct inode * dir, struct dentry
*dentry, str

   if (!inode)
    return ERR_PTR(-EACCES);
+
+  if (is_bad_inode(inode)) {
+   iput(inode);
+   return ERR_PTR(-ENOENT);
+  }
  }
  return d_splice_alias(inode, dentry);
 }
@@ -1052,6 +1057,11 @@ struct dentry *ext4_get_parent(struct dentry *child)
  if (!inode)
   return ERR_PTR(-EACCES);

+ if (is_bad_inode(inode)) {
+  iput(inode);
+  return ERR_PTR(-ENOENT);
```

```
+ }
+
  parent = d_alloc_anon(inode);
  if (!parent) {
   iput(inode);
```

Subject: Re: [RFC PATCH ext3/ext4] orphan list corruption due bad inode
Posted by Andrew Morton on Tue, 05 Jun 2007 02:03:40 GMT
View Forum Message <> Reply to Message

On Mon, 04 Jun 2007 09:19:10 +0400 Vasily Averin <vvs@sw.ru> wrote:

> After ext3 orphan list check has been added into ext3_destroy_inode() (please see my previous patch) the following situation has been detected:
> EXT3-fs warning (device sda6): ext3_unlink: Deleting nonexistent file (37901290), 0
> Inode 00000101a15b7840: orphan list check failed!
> 00000773 6f665f00 74616d72 00000573 65725f00 06737270 66000000 616d726f
> ...
> Call Trace: [<ffffffff80211ea9>] ext3_destroy_inode+0x79/0x90
> [<ffffffff801a2b16>] sys_unlink+0x126/0x1a0
> [<ffffffff80111479>] error_exit+0x0/0x81
> [<ffffffff80110aba>] system_call+0x7e/0x83
>
> First messages said that unlinked inode has i_nlink=0, then ext3_unlink() adds this inode into orphan list.
>
> Second message means that this inode has not been removed from orphan list. Inode dump has showed that i_fop = &bad_file_ops and it can be set in make_bad_inode() only. Then I've found that ext3_read_inode() can call make_bad_inode() without any error/warning messages, for example in the following case:
> ...
> if (inode->i_nlink == 0) {
> if (inode->i_mode == 0 ||
> !(EXT3_SB(inode->i_sb)->s_mount_state & EXT3_ORPHAN_FS)) {
> /* this inode is deleted */
> brelse (bh);
> goto bad_inode;
> ...
>
> Bad inode can live some time, ext3_unlink can add it to orphan list, but
> ext3_delete_inode() do not deleted this inode from orphan list. As
> result we can have orphan list corruption detected in ext3_destroy_inode().
>
> However it is not clear for me how to fix this issue correctly.
>
> As far as i see is_bad_inode() is called after iget() in all places excluding ext3_lookup() and ext3_get_parent(). I believe it makes sense to add bad inode check to these functions too and call

iput if bad inode detected.

Please avoid the 500-column paragraphs?

> Signed-off-by: Vasily Averin <vvs@sw.ru>
>
> diff --git a/fs/ext3/namei.c b/fs/ext3/namei.c
> index 9bb046d..e3ac8c3 100644
> --- a/fs/ext3/namei.c
> +++ b/fs/ext3/namei.c
> @@ -1019,6 +1019,11 @@ static struct dentry *ext3_lookup(struct inode * dir, struct dentry
*dentry, str
>
> if (!inode)
> return ERR_PTR(-EACCES);
> +
> + if (is_bad_inode(inode)) {
> + iput(inode);
> + return ERR_PTR(-ENOENT);
> + }
> }
> return d_splice_alias(inode, dentry);
> }
> @@ -1054,6 +1059,11 @@ struct dentry *ext3_get_parent(struct dentry *child)
> if (!inode)
> return ERR_PTR(-EACCES);
>
> + if (is_bad_inode(inode)) {
> + iput(inode);
> + return ERR_PTR(-ENOENT);
> + }
> +
> parent = d_alloc_anon(inode);
> if (!parent) {
> iput(inode);

Seems reasonable. So this prevents the bad inodes from getting onto the
orphan list in the first place?

What caused those inodes to be bad, anyway? Memory allocation failures?

Subject: Re: [RFC PATCH ext3/ext4] orphan list corruption due bad inode
Posted by Eric Sandeen on Tue, 05 Jun 2007 03:15:47 GMT
View Forum Message <> Reply to Message

Vasily Averin wrote:
> After ext3 orphan list check has been added into ext3_destroy_inode() (please see my previous

patch) the following situation has been detected:
> EXT3-fs warning (device sda6): ext3_unlink: Deleting nonexistent file (37901290), 0
> Inode 00000101a15b7840: orphan list check failed!
> 00000773 6f665f00 74616d72 00000573 65725f00 06737270 66000000 616d726f
> ...
> Call Trace: [<ffffffff80211ea9>] ext3_destroy_inode+0x79/0x90
>   [<ffffffff801a2b16>] sys_unlink+0x126/0x1a0
>   [<ffffffff80111479>] error_exit+0x0/0x81
>   [<ffffffff80110aba>] system_call+0x7e/0x83
>
> First messages said that unlinked inode has i_nlink=0, then ext3_unlink() adds this inode into orphan list.
>
> Second message means that this inode has not been removed from orphan list. Inode dump has showed that i_fop = &bad_file_ops and it can be set in make_bad_inode() only. Then I've found that ext3_read_inode() can call make_bad_inode() without any error/warning messages, for example in the following case:
> ...
>   if (inode->i_nlink == 0) {
>   if (inode->i_mode == 0 ||
>   !(EXT3_SB(inode->i_sb)->s_mount_state & EXT3_ORPHAN_FS)) {
>   /* this inode is deleted */
>   brelse (bh);
>   goto bad_inode;
> ...
>
> Bad inode can live some time, ext3_unlink can add it to orphan list, but
> ext3_delete_inode() do not deleted this inode from orphan list. As
> result we can have orphan list corruption detected in ext3_destroy_inode().

Ah, I see - so you have confirmed that this inode does have bad ops...?
I did notice on another orphan inode bug investigation that
ext3_inode_delete won't clear orphan from a bad_inode...

> However it is not clear for me how to fix this issue correctly.
>
> As far as i see is_bad_inode() is called after iget() in all places excluding ext3_lookup() and ext3_get_parent(). I believe it makes sense to add bad inode check to these functions too and call iput if bad inode detected.

That seems reasonable to me in any case, though as Andrew said, do you
know for sure how the bad inodes were getting on the orphan list...?

Is it possible that they were recycled after being freed while still on
the orphan list, as in my previous reply to your previous message?

-Eric

Subject: Re: [RFC PATCH ext3/ext4] orphan list corruption due bad inode
Posted by Andreas Dilger on Tue, 05 Jun 2007 03:52:32 GMT
View Forum Message <> Reply to Message

On Jun 04, 2007 19:03 -0700, Andrew Morton wrote:
> What caused those inodes to be bad, anyway? Memory allocation failures?

This can happen if e.g. NFS has a stale file handle - it will look up
the inode by inum, but ext3_read_inode() will create a bad inode due to
i_nlink = 0.

Cheers, Andreas
--
Andreas Dilger
Principal Software Engineer
Cluster File Systems, Inc.

Subject: Re: [RFC PATCH ext3/ext4] orphan list corruption due bad inode
Posted by vaverin on Tue, 05 Jun 2007 06:11:12 GMT
View Forum Message <> Reply to Message

Andrew Morton wrote:
> On Mon, 04 Jun 2007 09:19:10 +0400 Vasily Averin <vvs@sw.ru> wrote:
>> diff --git a/fs/ext3/namei.c b/fs/ext3/namei.c
>> index 9bb046d..e3ac8c3 100644
>> --- a/fs/ext3/namei.c
>> +++ b/fs/ext3/namei.c
>> @@ -1019,6 +1019,11 @@ static struct dentry *ext3_lookup(struct inode * dir, struct dentry
*dentry, str
>>
>>   if (!inode)
>>   return ERR_PTR(-EACCES);
>> +
>> + if (is_bad_inode(inode)) {
>> +   iput(inode);
>> +   return ERR_PTR(-ENOENT);
>> + }
>>   }
>>   return d_splice_alias(inode, dentry);
>> }
> Seems reasonable. So this prevents the bad inodes from getting onto the
> orphan list in the first place?

make_bad_inode() is called from ext3_read_inode() that is called from iget() only.

When we found that inode is bad we will call iput ASAP. I expect it should be
enough to exclude any chances to call any functions that can include this inode

into orphan list.

ext3_lookup
  iget
    ext3_read_inode
      make_bad_inode
  is_bad_inode? --> iput

However am I probably err?

Probably is better to add is_bad_inode check to ext3_orphan_add()?

thank you,
 Vasily Averin

Subject: Re: [RFC PATCH ext3/ext4] orphan list corruption due bad inode
Posted by Christoph Hellwig on Tue, 05 Jun 2007 06:13:15 GMT
View Forum Message <> Reply to Message

On Tue, Jun 05, 2007 at 10:11:12AM +0400, Vasily Averin wrote:
> >>   return d_splice_alias(inode, dentry);
> >> }
> > Seems reasonable. So this prevents the bad inodes from getting onto the
> > orphan list in the first place?
>
> make_bad_inode() is called from ext3_read_inode() that is called from iget() only.

Which is artefact of using the read_inode interface. Please switch from
iget to iget_locked and you can handle this case without ever inserting the
"bad" inode into the inode hash.

Subject: Re: [RFC PATCH ext3/ext4] orphan list corruption due bad inode
Posted by vaverin on Tue, 05 Jun 2007 06:49:10 GMT
View Forum Message <> Reply to Message

Eric Sandeen wrote:
> Vasily Averin wrote:
>> Bad inode can live some time, ext3_unlink can add it to orphan list, but
>> ext3_delete_inode() do not deleted this inode from orphan list. As result
>> we can have orphan list corruption detected in ext3_destroy_inode().
>
> Ah, I see - so you have confirmed that this inode does have bad ops...? I did
> notice on another orphan inode bug investigation that ext3_inode_delete won't
> clear orphan from a bad_inode...

yes, inode dump shows that i_fop = &bad_file_ops, and IMHO it's possible only
for bad inode.

>> However it is not clear for me how to fix this issue correctly.
>>
>> As far as i see is_bad_inode() is called after iget() in all places
>> excluding ext3_lookup() and ext3_get_parent(). I believe it makes sense to
>> add bad inode check to these functions too and call iput if bad inode
>> detected.
>
> That seems reasonable to me in any case, though as Andrew said, do you know
> for sure how the bad inodes were getting on the orphan list...?
>
> Is it possible that they were recycled after being freed while still on the
> orphan list, as in my previous reply to your previous message?

This incident has been occurred on Virtuozzo kernel based on latest RHEL4
2.6.9-55.el5, and it have your patch applied. btw thank you very much for this fix.

Unfortunately I do not know how this inode become bad, but inode dump in
ext3_destroy_inode shows that it is.

As far as I understand ext3_read_inode has been noticed that raw inode has
i_links_count=0 and therefore inode was marked as bad. However I have no any
ideas is it possible to have an inode on disk with i_links_count=0.

Thank you,
Vasily Averin

Subject: Re: [RFC PATCH ext3/ext4] orphan list corruption due bad inode
Posted by vaverin on Tue, 05 Jun 2007 07:31:10 GMT
View Forum Message <> Reply to Message

Christoph Hellwig wrote:
> On Tue, Jun 05, 2007 at 10:11:12AM +0400, Vasily Averin wrote:
>>>> return d_splice_alias(inode, dentry);
>>>> }
>>> Seems reasonable. So this prevents the bad inodes from getting onto the
>>> orphan list in the first place?
>> make_bad_inode() is called from ext3_read_inode() that is called from iget() only.
>
> Which is artefact of using the read_inode interface. Please switch from
> iget to iget_locked and you can handle this case without ever inserting the
> "bad" inode into the inode hash.

Sorry, I do not understand your proposal. Could you please explain your idea in
more details?

thank you,
Vasily Averin