CVE-2022-28204: Whatlinkshere of heavily used properties in wikidata can be easily utilized as a DDoS vector

Actions

Closed, Resolved Public SECURITY

Assigned To

Lucas_Werkmeister_WMDE

Authored By

Ladsgroup
2021-12-14 21:12:52 (UTC+0)

Tags

Security

Performance Issue

MediaWiki-Special-pages (Special:WhatLinksHere)

Vuln-DoS (Tracked)

Wikidata (incoming)

Wikidata-Campsite (Team A Hearth) (Our work done)

wdwb-tech (Inbox)

Referenced Files

F34944100: T297754-squash.patch
2022-02-07 10:58:42 (UTC+0)

F34939926: T297754-2.patch
2022-02-01 19:58:14 (UTC+0)

F34939887: T297754-2.patch
2022-02-01 19:21:19 (UTC+0)

F34932879: Screenshot from 2022-01-27 14-34-27.png
2022-01-27 13:35:36 (UTC+0)

F34932876: Screenshot from 2022-01-27 14-33-32.png
2022-01-27 13:35:36 (UTC+0)

F34921141: T297754.patch
2022-01-17 19:15:37 (UTC+0)

Subscribers

- Addshore

Aklapper

gerritbot

Ladsgroup

LSobanski

Lucas_Werkmeister_WMDE

Lydia_Pintscher

View All 15 Subscribers

Description

If you click on https://www.wikidata[.]org/w/index.php?title=Special%3AWhatLinksHere&target=Property%3AP31&namespace=1&invert=1 it'll take more than thirty seconds to load and at this rate can be simply turned into a DDoS attack vector.

It's already being used by users (and that's how I found it, from slow queries logs): https://logstash.wikimedia.org/app/discover#/doc/logstash-*/logstash-mediawiki-2021.12.14?id=DoYeuH0B-N5J53KJweTL

The query:

```
SELECT  page_id,page_namespace,page_title,rd_from,rd_fragment,page_is_redirect  FROM (SELECT
pl_from,rd_from,rd_fragment  FROM `pagelinks` LEFT JOIN `redirect` ON ((rd_from = pl_from) AND
rd_title = 'P31' AND rd_namespace = 120 AND (rd_interwiki = '' OR rd_interwiki IS NULL))   WHERE
pl_namespace = 120 AND pl_title = 'P31' AND pl_from_namespace IN
(0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,120,121,122,123,146,147,640,641,828,829,1198,1199,2300,2301,2302
,2303,2600)   ORDER BY pl_from ASC LIMIT 102  ) `temp_backlink_range` JOIN `page` ON ((pl_from =
page_id))    ORDER BY page_id ASC LIMIT 51
```

Explain:

```
*************************** 1. row ***************************
           id: 1
  select_type: PRIMARY
        table: <derived2>
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 102
        Extra: Using filesort
*************************** 2. row ***************************
           id: 1
  select_type: PRIMARY
        table: page
         type: eq_ref
possible_keys: PRIMARY
          key: PRIMARY
      key_len: 4
          ref: temp_backlink_range.pl_from
         rows: 1
        Extra:
*************************** 3. row ***************************
```

```
           id: 2
  select_type: DERIVED
        table: pagelinks
         type: range
possible_keys: pl_backlinks_namespace,pl_namespace
          key: pl_backlinks_namespace
      key_len: 265
          ref: NULL
         rows: 205355406
        Extra: Using where; Using index; Using filesort
*************************** 4. row ***************************
           id: 2
  select_type: DERIVED
        table: redirect
         type: eq_ref
possible_keys: PRIMARY,rd_ns_title
          key: PRIMARY
      key_len: 4
          ref: wikidatawiki.pagelinks.pl_from
         rows: 1
        Extra: Using where
4 rows in set (0.002 sec)

ERROR: No query specified
```

It tries to scan 200M rows! What makes it even more dangerous is that Special:Whatlinkshere is not among special pages I'm planning to put a cap on (~~T297708~~)

Details

Risk Rating
Low

Author Affiliation
WMF Technology Dept

	Project	Subject
	mediawiki/core	SECURITY: Sort Special:WhatLinksHere by namespace
	mediawiki/core	SECURITY: Sort Special:WhatLinksHere by namespace
	mediawiki/core	SECURITY: Sort Special:WhatLinksHere by namespace

Customize query in gerrit

Related Objects

Search...

Task Graph | Mentions

Status	Assigned	Task
Resolved	Reedy	~~T297829~~ Release MediaWiki 1.35.6/1.36.4/1.37.2

Resolved Reedy ~~T297830~~ Tracking bug for MediaWiki 1.35.6/1.36.4/1.37.2

Restricted Task

Resolved Lucas_Werkmeister_WMDE ~~T297754~~ CVE-2022-28204: Whatlinkshere of heavily used propert...

Ladsgroup created this task. 2021-12-14 21:12:52 (UTC+0)

Restricted Application added a subscriber: **Aklapper**. · View Herald Transcript 2021-12-14 21:12:53 (UTC+0)

Reedy added a project: **Performance Issue**. 2021-12-14 21:14:10 (UTC+0)

Aklapper added a project: **MediaWiki-Special-pages**. 2021-12-15 10:30:12 (UTC+0)

Michael added subscribers: **Silvan_WMDE**, **Rosalie_WMDE**. 2021-12-15 10:45:18 (UTC+0)

Lucas_Werkmeister_WMDE added a subscriber: **noarave**. 2021-12-15 11:16:33 (UTC+0)

Michael added a subscriber: **Lydia_Pintscher**. 2021-12-15 12:00:08 (UTC+0)

Lucas_Werkmeister_WMDE added a comment. Edited · 2021-12-15 13:10:52 (UTC+0)

Hm, without the `redirect` join, the query plan doesn't look much better –

```
MariaDB [wikidatawiki]> EXPLAIN SELECT  page_id,page_namespace,page_title,page_is_redirect  FROM
(SELECT  pl_from  FROM `pagelinks` WHERE pl_namespace = 120 AND pl_title = 'P31' AND
pl_from_namespace IN
(0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,120,121,122,123,146,147,640,641,828,829,1198,1199,2300,2301,230
2,2303,2600)   ORDER BY pl_from ASC LIMIT 102  ) `temp_backlink_range` JOIN `page` ON ((pl_from =
page_id))    ORDER BY page_id ASC LIMIT 51;
+------+-------------+------------+--------+-----------------------------------+---------+--------
-+----------------------------+------------+----------------+
| id   | select_type | table      | type   | possible_keys                     | key     | key_len
| ref                        | rows       | Extra          |
+------+-------------+------------+--------+-----------------------------------+---------+--------
-+----------------------------+------------+----------------+
|    1 | PRIMARY     | <derived2> | ALL    | NULL                              | NULL    | NULL
| NULL                       | 102        | Using filesort |
|    1 | PRIMARY     | page       | eq_ref | PRIMARY                           | PRIMARY | 4
| temp_backlink_range.pl_from | 1          |                |
|    2 | DERIVED     | pagelinks  | index  | pl_backlinks_namespace,pl_namespace | PRIMARY | 265
| NULL                       | 1997287820 | Using where    |
+------+-------------+------------+--------+-----------------------------------+---------+--------
-+----------------------------+------------+----------------+
3 rows in set (0.003 sec)
```

– but it seems to run very quickly (0.029 s), despite still reporting almost 200 million rows. This is on stat1007 (or rather, my shell is on stat1007 – I think the actual db host is separate?), I don't know which host you were testing on.

Do you think it would help if we somehow inform MediaWiki core that, for this namespace / content model / whatever, it doesn't need to bother looking for redirects, because properties can't be merged/redirected?

Lucas_Werkmeister_WMDE added a comment. 2021-12-15 13:16:48 (UTC+0)

On the other hand, that wouldn't help with items, which can be redirected, and which can also be heavily used:

```
MariaDB [wikidatawiki]> EXPLAIN SELECT
page_id,page_namespace,page_title,rd_from,rd_fragment,page_is_redirect  FROM (SELECT
pl_from,rd_from,rd_fragment  FROM `pagelinks` LEFT JOIN `redirect` ON ((rd_from = pl_from) AND
rd_title = 'Q13442814' AND rd_namespace = 0 AND (rd_interwiki = '' OR rd_interwiki IS NULL))   WHERE
pl_namespace = 0 AND pl_title = 'Q13442814' AND pl_from_namespace IN
(0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,120,121,122,123,146,147,640,641,828,829,1198,1199,2300,2301,230
2,2303,2600)   ORDER BY pl_from ASC LIMIT 102  ) `temp_backlink_range` JOIN `page` ON ((pl_from =
page_id))    ORDER BY page_id ASC LIMIT 51;
+------+-------------+------------+--------+-----------------------------------+-----------------
------+---------+---------------------------------+----------+-----------------------------------
---+
| id   | select_type | table      | type   | possible_keys                     | key
| key_len | ref                             | rows     | Extra                                 |
+------+-------------+------------+--------+-----------------------------------+-----------------
------+---------+---------------------------------+----------+-----------------------------------
---+
|    1 | PRIMARY     | <derived2> | ALL    | NULL                              | NULL
| NULL    | NULL                            | 102      | Using filesort                        |
|    1 | PRIMARY     | page       | eq_ref | PRIMARY                           | PRIMARY
| 4       | temp_backlink_range.pl_from     | 1        |                                       |
|    2 | DERIVED     | pagelinks  | range  | pl_backlinks_namespace,pl_namespace |
pl_backlinks_namespace | 265     | NULL                            | 77277359 | Using where; Using
index; Using filesort |
|    2 | DERIVED     | redirect   | eq_ref | PRIMARY,rd_ns_title               | PRIMARY
| 4       | wikidatawiki.pagelinks.pl_from | 1        | Using where                           |
+------+-------------+------------+--------+-----------------------------------+-----------------
------+---------+---------------------------------+----------+-----------------------------------
---+
4 rows in set (0.018 sec)

MariaDB [wikidatawiki]> SELECT
page_id,page_namespace,page_title,rd_from,rd_fragment,page_is_redirect  FROM (SELECT
pl_from,rd_from,rd_fragment  FROM `pagelinks` LEFT JOIN `redirect` ON ((rd_from = pl_from) AND
rd_title = 'Q13442814' AND rd_namespace = 0 AND (rd_interwiki = '' OR rd_interwiki IS NULL))   WHERE
pl_namespace = 0 AND pl_title = 'Q13442814' AND pl_from_namespace IN
(0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,120,121,122,123,146,147,640,641,828,829,1198,1199,2300,2301,230
2,2303,2600)   ORDER BY pl_from ASC LIMIT 102  ) `temp_backlink_range` JOIN `page` ON ((pl_from =
page_id))    ORDER BY page_id ASC LIMIT 51;
+---------+----------------+------------------+---------+-------------+------------------+
| page_id | page_namespace | page_title       | rd_from | rd_fragment | page_is_redirect |
+---------+----------------+------------------+---------+-------------+------------------+
(snip)
+---------+----------------+------------------+---------+-------------+------------------+
51 rows in set (1 min 11.049 sec)
```

That's apparently scanning (trying to scan?) some 77 million rows, and taking over a minute, to find links to scholarly article.

Lucas_Werkmeister_WMDE added a comment. 2021-12-15 15:16:15 (UTC+0)

Hm, the original query is still very quick (again, on stat1007) if I force it to use the primary key, rather than `pl_backlinks_namespace`, for `pagelinks`.

```
MariaDB [wikidatawiki]> EXPLAIN SELECT
page_id,page_namespace,page_title,rd_from,rd_fragment,page_is_redirect  FROM (SELECT
pl_from,rd_from,rd_fragment  FROM `pagelinks` USE INDEX (PRIMARY) LEFT JOIN `redirect` ON ((rd_from
```

```
= pl_from) AND rd_title = 'P31' AND rd_namespace = 120 AND (rd_interwiki = '' OR rd_interwiki IS
NULL))   WHERE pl_namespace = 120 AND pl_title = 'P31' AND pl_from_namespace IN
(0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,120,121,122,123,146,147,640,641,828,829,1198,1199,2300,2301,230
2,2303,2600)   ORDER BY pl_from ASC LIMIT 102  ) `temp_backlink_range` JOIN `page` ON ((pl_from =
page_id))    ORDER BY page_id ASC LIMIT 51;
+------+-------------+------------+--------+--------------------+---------+---------+-------------
-----------------+------------+----------------+
| id   | select_type | table      | type   | possible_keys      | key     | key_len | ref
| rows       | Extra          |
+------+-------------+------------+--------+--------------------+---------+---------+-------------
-----------------+------------+----------------+
|    1 | PRIMARY     | <derived2> | ALL    | NULL               | NULL    | NULL    | NULL
| 102        | Using filesort |
|    1 | PRIMARY     | page       | eq_ref | PRIMARY            | PRIMARY | 4       |
temp_backlink_range.pl_from    | 1          |                |
|    2 | DERIVED     | pagelinks  | index  | NULL               | PRIMARY | 265     | NULL
| 1997365952 | Using where    |
|    2 | DERIVED     | redirect   | eq_ref | PRIMARY,rd_ns_title | PRIMARY | 4       |
wikidatawiki.pagelinks.pl_from | 1          | Using where    |
+------+-------------+------------+--------+--------------------+---------+---------+-------------
-----------------+------------+----------------+

MariaDB [wikidatawiki]> SELECT
page_id,page_namespace,page_title,rd_from,rd_fragment,page_is_redirect  FROM (SELECT
pl_from,rd_from,rd_fragment  FROM `pagelinks` USE INDEX (PRIMARY) LEFT JOIN `redirect` ON ((rd_from
= pl_from) AND rd_title = 'P31' AND rd_namespace = 120 AND (rd_interwiki = '' OR rd_interwiki IS
NULL))   WHERE pl_namespace = 120 AND pl_title = 'P31' AND pl_from_namespace IN
(0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,120,121,122,123,146,147,640,641,828,829,1198,1199,2300,2301,230
2,2303,2600)   ORDER BY pl_from ASC LIMIT 102  ) `temp_backlink_range` JOIN `page` ON ((pl_from =
page_id))    ORDER BY page_id ASC LIMIT 51;
(snip)
51 rows in set (0.029 sec)
```

I'm guessing that this isn't a good idea in general, normally that backlinks index is probably useful... would "ignore this index if the namespace filter is inverted" work as a heuristic? Or should we try to get MySQL/MariaDB to realize that the index is a bad choice?

Lucas_Werkmeister_WMDE added a comment. 2021-12-15 15:22:16 (UTC+0)

> *would "ignore this index if the namespace filter is inverted" work as a heuristic?*

(That wouldn't work in the API, where people can select an arbitrary set of namespaces – but also, I haven't been able to reproduce the slowness using the API yet, maybe because the API version of this query will never include the `redirects` join.)

Lucas_Werkmeister_WMDE added a comment. Edited · 2021-12-15 16:22:14 (UTC+0)

Hm, but the inverse namespace filtering is supposed to be a feature:

SpecialWhatLinksHere::showIndirectLinks()

```
                        if ( $invert ) {
                                // Select all namespaces except for the specified one.
                                // This allows the database to use the *_from_namespace index.
```

```
(T241837)
                                $namespaces = array_diff(
                                        $this->namespaceInfo->getValidNamespaces(), [ $namespace ]
);
                        } else {
                                $namespaces = $namespace;
                        }
```

Mentioned task: ~~**T241837: WMFTimeoutException on Commons for WhatLinksHere**~~

Lucas_Werkmeister_WMDE added a comment. 2021-12-17 10:38:11 (UTC+0)

I think we need some help from the DBAs here, I don't know how to make this query not misbehave.

In the meantime, since I assume a lot of people are about to leave for the holidays, here's an emergency patch if the DDoS vector is suddenly used more aggressively:

operations/mediawiki-config.git

```
diff --git a/wmf-config/Wikibase.php b/wmf-config/Wikibase.php
index 03373fdeb..1fb33e991 100644
--- a/wmf-config/Wikibase.php
+++ b/wmf-config/Wikibase.php
@@ -76,16 +76,19 @@
        if ( $wgDBname === 'wikidatawiki' || $wgDBname === 'testwikidatawiki' ) {
                // Don't try to let users answer captchas if they try to add links
                // on either Item or Property pages. T86453
                $wgCaptchaTriggersOnNamespace[NS_MAIN]['addurl'] = false;
                $wgCaptchaTriggersOnNamespace[WB_NS_PROPERTY]['addurl'] = false;

                // T53637 and T48953
                $wgGroupPermissions['*']['property-create'] = ( $wgDBname ===
'testwikidatawiki' );
+
+               // T297754
+               $wgSpecialPages['WhatLinksHere'] = DisabledSpecialPage::getCallback(
'WhatLinksHere', 'querypage-disabled' );
        }
```

Only deploy this if it's really necessary, though, since disabling WhatLinksHere will definitely hurt Wikidata editors too.

Ladsgroup added a comment. 2021-12-17 12:11:44 (UTC+0)

I want to mention it's on my todo list. I'll take a look soon.

sbassett edited projects, added **Vuln-DoS**; removed **Security-Team**. 2021-12-20 16:47:57 (UTC+0)

Ladsgroup added a comment. 2021-12-21 14:49:11 (UTC+0)

There is a rather simpler solution that attacks the problem in a different angle. Drop ordering by page_id.

```
wikiadmin@10.64.0.128(wikidatawiki)> explain SELECT
page_id,page_namespace,page_title,rd_from,rd_fragment,page_is_redirect  FROM (SELECT
pl_from,rd_from,rd_fragment  FROM `pagelinks` LEFT JOIN `redirect` ON ((rd_from = pl_from) AND
```

```
rd_title = 'P31' AND rd_namespace = 120 AND (rd_interwiki = '' OR rd_interwiki IS NULL))   WHERE
pl_namespace = 120 AND pl_title = 'P31' AND pl_from_namespace IN
(0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,120,121,122,123,146,147,640,641,828,829,1198,1199,2300,2301,230
2,2303,2600)  LIMIT 102  ) `temp_backlink_range` JOIN `page` ON ((pl_from = page_id)) LIMIT 51;
+------+-------------+------------+--------+-----------------------------------+-----------------
------+---------+--------------------------------+-----------+--------------------------+
| id   | select_type | table      | type   | possible_keys                     | key
| key_len | ref                            | rows      | Extra                     |
+------+-------------+------------+--------+-----------------------------------+-----------------
------+---------+--------------------------------+-----------+--------------------------+
|    1 | PRIMARY     | <derived2> | ALL    | NULL                              | NULL
| NULL    | NULL                           | 102       |                           |
|    1 | PRIMARY     | page       | eq_ref | PRIMARY                           | PRIMARY
| 4       | temp_backlink_range.pl_from    | 1         |                           |
|    2 | DERIVED     | pagelinks  | range  | pl_backlinks_namespace,pl_namespace |
pl_backlinks_namespace | 265     | NULL                           | 220663090 | Using where; Using
index |
|    2 | DERIVED     | redirect   | eq_ref | PRIMARY,rd_ns_title               | PRIMARY
| 4       | wikidatawiki.pagelinks.pl_from | 1         | Using where               |
+------+-------------+------------+--------+-----------------------------------+-----------------
------+---------+--------------------------------+-----------+--------------------------+
4 rows in set (0.002 sec)
```

And:

```
wikiadmin@10.64.0.128(wikidatawiki)> SELECT
page_id,page_namespace,page_title,rd_from,rd_fragment,page_is_redirect  FROM (SELECT
pl_from,rd_from,rd_fragment  FROM `pagelinks` LEFT JOIN `redirect` ON ((rd_from = pl_from) AND
rd_title = 'P31' AND rd_namespace = 120 AND (rd_interwiki = '' OR rd_interwiki IS NULL))   WHERE
pl_namespace = 120 AND pl_title = 'P31' AND pl_from_namespace IN
(0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,120,121,122,123,146,147,640,641,828,829,1198,1199,2300,2301,230
2,2303,2600)  LIMIT 102  ) `temp_backlink_range` JOIN `page` ON ((pl_from = page_id)) LIMIT 51;
+---------+----------------+------------+---------+-------------+------------------+
| page_id | page_namespace | page_title | rd_from | rd_fragment | page_is_redirect |
+---------+----------------+------------+---------+-------------+------------------+
...
+---------+----------------+------------+---------+-------------+------------------+
51 rows in set (0.002 sec)
```

But that would break pagination. To solution to that would be to bring back order but put it on pl_namespace and then pl_from:

```
wikiadmin@10.64.0.128(wikidatawiki)> SELECT
page_id,page_namespace,page_title,rd_from,rd_fragment,page_is_redirect  FROM (SELECT
pl_from,rd_from,rd_fragment  FROM `pagelinks` LEFT JOIN `redirect` ON ((rd_from = pl_from) AND
rd_title = 'P31' AND rd_namespace = 120 AND (rd_interwiki = '' OR rd_interwiki IS NULL))   WHERE
pl_namespace = 120 AND pl_title = 'P31' AND pl_from_namespace IN
(0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,120,121,122,123,146,147,640,641,828,829,1198,1199,2300,2301,230
2,2303,2600)   ORDER BY pl_from_namespace, pl_from ASC LIMIT 102  ) `temp_backlink_range` JOIN
`page` ON ((pl_from = page_id))    ORDER BY page_id ASC LIMIT 51;
+---------+----------------+------------+---------+-------------+------------------+
| page_id | page_namespace | page_title | rd_from | rd_fragment | page_is_redirect |
+---------+----------------+------------+---------+-------------+------------------+
...
+---------+----------------+------------+---------+-------------+------------------+
51 rows in set (0.013 sec)
```

Implementing pagination in this mode is not that hard but not super easy either. How does that sound to you? The explain says it reads a lot of rows but as long as it doesn't go to "filesort" that's fine.

Ladsgroup added a comment. 2021-12-21 14:59:04 (UTC+0)

One rather scary aspect of this DDoS vector is that since pagelinks is quite big and this force a read on the whole table, running this multiple times can easily fill the mysql's in-memory cache (innodb buffer pool) and bring everything down. It's very similar to water torture attack in DNS.

Lucas_Werkmeister_WMDE added a comment. 2021-12-21 15:13:46 (UTC+0)

Hm, that would match the API too: if I see it correctly, `ApiQueryBacklinks` already orders by `bl_from_ns` before `bl_from` if there's more than one namespace. (It may also order by target namespace and title before that, apparently.)

For users of the special page, this sounds like it would change the special page to "group" all the links from the same namespace together (i.e. first list all item-namespace links, then all talk-namespace ones, etc.). That sounds like it could even be considered a feature. (It could also break some workflows that rely on the ordering by page ID… I can't say.)

Sounds like a solid suggestion to me.

Lydia_Pintscher added a comment. 2022-01-07 15:22:38 (UTC+0)

Additionally sorting by namespace seems ok from product side.

Lucas_Werkmeister_WMDE added a comment. 2022-01-17 19:15:37 (UTC+0)

Alright, here's a patch for sorting by namespace first – review welcome:

T297754.patch 11 KB
Download

I'll quote a part of my commit message:

> *The URL changes format again, and now looks like* `&offset=0|123`*, where* 0 *is the namespace of page* 123*, and the results will be the pages in the same namespace with an ID above* 123*, or the pages in namespaces above* 0 *regardless of page ID (though still sorted). Old URLs are of course supported, and we look up the relevant namespace of the given page ID on-the-fly in that case.*

We could also keep the URL format the same, and always do the on-the-fly lookup of the namespace corresponding to the offset page ID, at the cost of an additional database query per request (though it's a very lightweight query). Does anyone have preferences for or against that? (Also, I used a `TitleFactory` to get the namespace for a page ID, is that the best way or is there something else?)

Lucas_Werkmeister_WMDE added projects: **Wikidata**, ~~**Wikidata-Campsite (Team A Hearth)**~~.
2022-01-18 10:20:36 (UTC+0)

Lucas_Werkmeister_WMDE moved this task from **To triage** to **Special:WhatLinksHere** on the **MediaWiki-Special-pages** board.

Restricted Application added a project: **wdwb-tech**. · View Herald Transcript 2022-01-18 10:20:37 (UTC+0)

Lucas_Werkmeister_WMDE moved this task from **Incoming** to **Peer Review** on the ~~**Wikidata-Campsite (Team A Hearth)**~~ board. 2022-01-18 10:20:47 (UTC+0)

Lucas_Werkmeister_WMDE added a comment. 2022-01-18 11:55:14 (UTC+0)

> > *The URL changes format again, and now looks like* `&offset=0|123`*, where* 0 *is the namespace of page* 123*, and the results will be the pages in the same namespace with an ID above* 123*, or the pages in namespaces above* 0 *regardless of page ID (though still sorted). Old URLs are of course supported, and we look up the relevant namespace of the given page ID on-the-fly in that case.*
>
> *We could also keep the URL format the same, and always do the on-the-fly lookup of the namespace corresponding to the offset page ID, at the cost of an additional database query per request (though it's a very lightweight query). Does anyone have preferences for or against that? (Also, I used a* `TitleFactory` *to get the namespace for a page ID, is that the best way or is there something else?)*

I realized that including the namespace in the `&offset=` also makes the URLs slightly more robust against page deletion.

I'm still not sure about this though. Are there other special pages that sort by namespace and page ID, and which don't use querycache(2)? How do they handle pagination?

Michael added a comment. 2022-01-18 16:29:59 (UTC+0)

> In ~~**T297754#7626820**~~, **@Lucas_Werkmeister_WMDE** wrote:
>
> *Alright, here's a patch for sorting by namespace first – review welcome:*
>
> ***T297754.patch*** *11 KB*
> *Download*

I looked at the patch and it makes sense to me code-wise. Also, the adjusted URL format is fine for me. I'm still trying it out locally.

I'm wondering though, *how* does it make a big difference in the cases where it really matters? For example, when looking at what links to "scholarly article", then almost all the millions of matches are in namespace 0 and still have to be sorted by pageid, right?

/me tries to remember their DB training.

Lucas_Werkmeister_WMDE added a comment. 2022-01-18 17:27:50 (UTC+0)

Once we sort by the “from” namespace and page ID (`pl_from_namespace, pl_from`), MySQL can use the index order (`INDEX pl_backlinks_namespace (pl_from_namespace, pl_namespace, pl_title, pl_from)` – note that `pl_namespace` and `pl_title` are constants in our query) without having to filesort, if I understand correctly.

@Ladsgroup any idea how your work in **~~T222224: RFC: Normalize MediaWiki link tables~~** will affect this, by the way? (I just saw that you created a bunch of subtasks there.)

Michael added a comment. 2022-01-18 17:27:52 (UTC+0)

Tried it out locally and seems to work, so it gets my virtual +1.

On a more general note: I've stumbled about the code below. It *is* safe, I've checked! But it still feels like the sort of code that is in principle prone to SQL injections, right? After this has been made public, maybe we can (1) add types to the signature of `showIndirectLinks()` to ensure that `$offsetNamespace` and `$offsetPageID` can truly only ever be ints, and (2) refactor this a bit more comprehensively so that only PDO ever puts any variables into actual SQL.

```
-       if ( $offset ) {
+       if ( $offsetPageID ) {
            $rel = $dir === 'prev' ? '<' : '>';
-           $conds['redirect'][] = "rd_from $rel $offset";
-           $conds['templatelinks'][] = "tl_from $rel $offset";
-           $conds['pagelinks'][] = "pl_from $rel $offset";
-           $conds['imagelinks'][] = "il_from $rel $offset";
+           $conds['redirect'][] = "rd_from $rel $offsetPageID";
+           $conds['templatelinks'][] = "(tl_from_namespace = $offsetNamespace AND tl_from $rel
$offsetPageID " .
+               "OR tl_from_namespace $rel $offsetNamespace)";
+           $conds['pagelinks'][] =  "(pl_from_namespace = $offsetNamespace AND pl_from $rel
$offsetPageID " .
+               "OR pl_from_namespace $rel $offsetNamespace)";
+           $conds['imagelinks'][] =  "(il_from_namespace = $offsetNamespace AND il_from $rel
$offsetPageID " .
+               "OR il_from_namespace $rel $offsetNamespace)";
        }
```

Ladsgroup added a comment. 2022-01-18 18:34:37 (UTC+0)

> In **~~T297754#7629225~~**, **@Lucas_Werkmeister_WMDE** wrote:
>
> ***@Ladsgroup*** *any idea how your work in* ***~~T222224: RFC: Normalize MediaWiki link tables~~*** *will affect this, by the way? (I just saw that you created a bunch of subtasks there.)*

Normalizing pagelinks would make almost all queries faster but it wouldn't matter here:

- This query at the current stage in pathological, it's several orders of magnitude worse than what can be considered okay. i.e. it's beyond salvation.
- I'm starting with templatelinks table and normalizing that will take several months, than I will look into pagelinks. So I'm sure this won't step on your toes.

Lucas_Werkmeister_WMDE added a comment. 2022-01-24 10:56:51 (UTC+0)

I know the normalization won't happen in time to resolve this task, but I'm wondering what the continuation URLs will look like after that migration is done. Will it still make sense to continue from a page ID and namespace? Will the page ID alone be more natural, so we keep that in the meantime (and do the on-the-fly lookup I mentioned above)? Or something else?

My guess for now is that it won't actually have any effect, since you're normalizing just the two columns that we *don't* sort by (`pl_namespace` , `pl_title`). I think I missed this in my earlier comment, and thought the task affected `pl_from` and `pl_from_namespace` too.

Ladsgroup added a comment. 2022-01-24 11:28:26 (UTC+0)

Yes, the normalization can't possibly have an effect on pagination in whatlinkshere. It's on the target not the source.

Lucas_Werkmeister_WMDE added a comment. 2022-01-24 18:02:49 (UTC+0)

I tried the patch on mwdebug1001 – it seems to result in an efficient database query and at a glance pagination worked as expected.

Ladsgroup added a parent task: Restricted Task. 2022-01-24 19:38:20 (UTC+0)

Lucas_Werkmeister_WMDE added a comment. 2022-01-27 11:00:27 (UTC+0)

Nahhh, something isn't quite working right in the pagination yet. It seems to work correctly within a namespace, but when you have a list with pages from two namespaces, then click "next 50", and then go back to "previous 50", you don't have quite the same list. I'll try to debug that locally.

In the meantime, I'd still appreciate feedback on the URL format:

> > *The URL changes format again, and now looks like* `&offset=0|123` *, where* 0 *is the namespace of page* 123*, and the results will be the pages in the same namespace with an ID above* 123*, or the pages in namespaces above* 0 *regardless of page ID (though still sorted). Old URLs are of course supported, and we look up the relevant namespace of the given page ID on-the-fly in that case.*
>
> *We could also keep the URL format the same, and always do the on-the-fly lookup of the namespace corresponding to the offset page ID, at the cost of an additional database query per request (though it's a very lightweight query).*

> *Does anyone have preferences for or against that? (Also, I used a `TitleFactory` to get the namespace for a page ID, is that the best way or is there something else?)*

> *I'm still not sure about this though. Are there other special pages that sort by namespace and page ID, and which don't use querycache(2)? How do they handle pagination?*

But I don' want to delay this fix forever either, so let's say that if nobody has objected to the URL change by Wednesday, 1 February 2022, I'll deploy the change as soon as I've found a fix for pagination.

Michael added a comment. 2022-01-27 11:24:54 (UTC+0)

The URL format is fine for me. I guess the alternative would be to have separate parameters for namespace and pageID? Both options have their advantages and disadvantages and they seem roughly equally good/bad to me, on balance. So I'm fine with the format you suggested.

Lucas_Werkmeister_WMDE added a comment. 2022-01-27 13:35:36 (UTC+0)

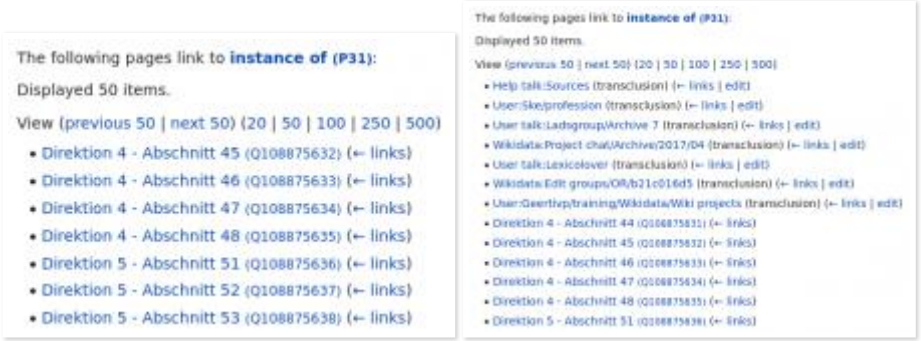
Thanks! Yeah, a separate parameter feels worse to me somehow, though I can't really put my finger on why.

> *It seems to work correctly within a namespace, but when you have a list with pages from two namespaces, then click "next 50", and then go back to "previous 50", you don't have quite the same list. I'll try to debug that locally.*

It turns out this already happens with the old code even without a namespace filter:

- offset=104103045&dir=next, first list item Q108875632 (page ID 104103046), last list item Q108875683 (page ID 104103097), "next 50" links to…
- offset=104103097&dir=next, first list item Q108875684 (page ID 104103098), last list item Q108875756 (page ID 104103158), "previous 50" links to…
- offset=104103098&dir=prev, first list item Help_talk:Sources (page ID 8279967), last list item Q108875675 (page ID 104103089); Q108875632 (the original first list item) is near the beginning of the list, the second item after a bunch of transclusion items; Q108875675 (current last list item) is near the end of the original list

Beginning of first and third list (which I'd expect to show the same items) side by side:



It looks like there's a bug when there are multiple sources of links (in this case, transclusions), which add extra entries to the "prev" version. But anyways, it seems clear to me that it's not related to the change, it's already broken and

probably not made worse.

Lucas_Werkmeister_WMDE added a comment. 2022-02-01 10:55:42 (UTC+0)

> In ~~T297754#7655889~~, **@Lucas_Werkmeister_WMDE** wrote:
>
> *But I don' want to delay this fix forever either, so let's say that if nobody has objected to the URL change by Wednesday, 1 February 2022, I'll deploy the change as soon as I've found a fix for pagination.*

There is no Wednesday, 1 February 2022, but **I've deployed the change now.**

Do we want to keep this task private until the next security release? I would think the risk to third-party wikis should be fairly low.

Lucas_Werkmeister_WMDE moved this task from **Peer Review** to **Tech Verification** on the ~~**Wikidata-Campsite (Team A Hearth)**~~ board. 2022-02-01 10:55:58 (UTC+0)

Michael added a comment. 2022-02-01 11:27:56 (UTC+0)

> In ~~T297754#7667115~~, **@Lucas_Werkmeister_WMDE** wrote:
>
> *Do we want to keep this task private until the next security release? I would think the risk to third-party wikis should be fairly low.*

Until the next security release is maybe not needed, but we should probably wait until this patch has proved itself on production and we're confident that it won't be rolled back again.

sbassett added a parent task: ~~**T297830: Tracking bug for MediaWiki 1.35.6/1.36.4/1.37.2**~~.
2022-02-01 16:00:31 (UTC+0)

sbassett assigned this task to **Lucas_Werkmeister_WMDE**. Edited · 2022-02-01 16:03:15 (UTC+0)

sbassett added a subscriber: **sbassett**.

> In ~~T297754#7667115~~, **@Lucas_Werkmeister_WMDE** wrote:
>
> *Do we want to keep this task private until the next security release? I would think the risk to third-party wikis should be fairly low.*
>
> > *! In ~~T297754#7667206~~, @Michael wrote:*
> >
> > *Until the next security release is maybe not needed, but we should probably wait until this patch has proved itself on production and we're confident that it won't be rolled back again.*

Even though it might be a low-risk Vuln-DoS, yes, this task should stay protected until the next security release (~~T297829~~). I've added it as a sub-task to the tracking bug (~~T297830~~) for the next release.

Ladsgroup added a comment. 2022-02-01 18:04:07 (UTC+0)

We have had a huge spike in really slow queries from Special:WhatLinksHere:
https://logstash.wikimedia.org/goto/386b2acf30578aac6f08b7c58048c0bd This might warrant a revert.

Lucas_Werkmeister_WMDE added a comment. 2022-02-01 19:03:33 (UTC+0)

If we just revert, then links like `offset=0|120` will break :(

Any idea why the queries are being slow?

Lucas_Werkmeister_WMDE added a comment. Edited · 2022-02-01 19:07:56 (UTC+0)

Original query of reqId 02689551-58ed-4479-98fc-ea12853ffb93:

```
MariaDB [enwiki]> EXPLAIN SELECT /* SpecialWhatLinksHere::showIndirectLinks  */
page_id,page_namespace,page_title,rd_from,rd_fragment,page_is_redirect  FROM (SELECT
tl_from,rd_from,rd_fragment  FROM `templatelinks` LEFT JOIN `redirect` ON ((rd_from = tl_from) AND
rd_title = 'Ambox' AND rd_namespace = 10 AND (rd_interwiki = '' OR rd_interwiki IS NULL))   WHERE
tl_namespace = 10 AND tl_title = 'Ambox'  ORDER BY tl_from_namespace ASC,tl_from ASC LIMIT 102  )
`temp_backlink_range` JOIN `page` ON ((tl_from = page_id))    ORDER BY page_namespace ASC,page_id
ASC LIMIT 51  ;
+------+-------------+---------------+--------+---------------------+-------------+---------+------
-----------------------+---------+-------------------------------------------------------+
| id   | select_type | table         | type   | possible_keys       | key         | key_len | ref
| rows    | Extra                                          |
+------+-------------+---------------+--------+---------------------+-------------+---------+------
-----------------------+---------+-------------------------------------------------------+
|    1 | PRIMARY     | <derived2>    | ALL    | NULL                | NULL        | NULL    | NULL
| 102     | Using temporary; Using filesort                |
|    1 | PRIMARY     | page          | eq_ref | PRIMARY             | PRIMARY     | 4       |
temp_backlink_range.tl_from  | 1       |                                                       |
|    2 | DERIVED     | templatelinks | ref    | tl_namespace        | tl_namespace | 261     |
const,const                  | 3412934 | Using index condition; Using where; Using filesort |
|    2 | DERIVED     | redirect      | eq_ref | PRIMARY,rd_ns_title | PRIMARY     | 4       |
enwiki.templatelinks.tl_from | 1       | Using where                                           |
+------+-------------+---------------+--------+---------------------+-------------+---------+------
-----------------------+---------+-------------------------------------------------------+
4 rows in set (0.005 sec)
```

Manually removing the namespace from the order:

```
MariaDB [enwiki]> EXPLAIN SELECT /* SpecialWhatLinksHere::showIndirectLinks  */
page_id,page_namespace,page_title,rd_from,rd_fragment,page_is_redirect  FROM (SELECT
tl_from,rd_from,rd_fragment  FROM `templatelinks` LEFT JOIN `redirect` ON ((rd_from = tl_from) AND
rd_title = 'Ambox' AND rd_namespace = 10 AND (rd_interwiki = '' OR rd_interwiki IS NULL))   WHERE
tl_namespace = 10 AND tl_title = 'Ambox'  ORDER BY tl_from ASC LIMIT 102  ) `temp_backlink_range`
JOIN `page` ON ((tl_from = page_id))    ORDER BY page_id ASC LIMIT 51  ;
+------+-------------+---------------+--------+---------------------+-------------+---------+------
-----------------------+---------+--------------------------+
| id   | select_type | table         | type   | possible_keys       | key         | key_len | ref
| rows    | Extra                    |
+------+-------------+---------------+--------+---------------------+-------------+---------+------
-----------------------+---------+--------------------------+
```

```
|    1 | PRIMARY     | <derived2>    | ALL    | NULL                | NULL         | NULL    | NULL
| 102     | Using filesort           |
|    1 | PRIMARY     | page          | eq_ref | PRIMARY             | PRIMARY      | 4       |
temp_backlink_range.tl_from  | 1       |                          |
|    2 | DERIVED     | templatelinks | ref    | tl_namespace        | tl_namespace | 261     |
const,const                  | 3413410 | Using where; Using index |
|    2 | DERIVED     | redirect      | eq_ref | PRIMARY,rd_ns_title | PRIMARY      | 4       |
enwiki.templatelinks.tl_from | 1       | Using where              |
+------+-------------+---------------+--------+---------------------+--------------+---------+------
------------------------+---------+--------------------------+
4 rows in set (0.001 sec)
```

(Edit: on the stats machines, i.e. from stat1007)

Lucas_Werkmeister_WMDE added a comment. 2022-02-01 19:14:03 (UTC+0)

Why is even this simple query slow on the stats machines?

```
MariaDB [enwiki]> EXPLAIN SELECT * FROM templatelinks WHERE tl_namespace = 10 AND tl_title = 'Ambox'
ORDER BY tl_from_namespace ASC, tl_from ASC LIMIT 10;
+------+-------------+---------------+-------+--------------+-----------------------+---------+---
---+------+--------------------------+
| id   | select_type | table         | type  | possible_keys | key                   | key_len |
ref  | rows | Extra                    |
+------+-------------+---------------+-------+--------------+-----------------------+---------+---
---+------+--------------------------+
|    1 | SIMPLE      | templatelinks | index | tl_namespace  | tl_backlinks_namespace | 269     |
NULL | 3203 | Using where; Using index |
+------+-------------+---------------+-------+--------------+-----------------------+---------+---
---+------+--------------------------+
1 row in set (0.001 sec)

MariaDB [enwiki]> SELECT * FROM templatelinks WHERE tl_namespace = 10 AND tl_title = 'Ambox' ORDER
BY tl_from_namespace ASC, tl_from ASC LIMIT 10;
+---------+--------------+----------+-------------------+
| tl_from | tl_namespace | tl_title | tl_from_namespace |
+---------+--------------+----------+-------------------+
|      25 |           10 | Ambox    |                 0 |
|     303 |           10 | Ambox    |                 0 |
|     309 |           10 | Ambox    |                 0 |
|     324 |           10 | Ambox    |                 0 |
|     340 |           10 | Ambox    |                 0 |
|     359 |           10 | Ambox    |                 0 |
|     572 |           10 | Ambox    |                 0 |
|     595 |           10 | Ambox    |                 0 |
|     599 |           10 | Ambox    |                 0 |
|     600 |           10 | Ambox    |                 0 |
+---------+--------------+----------+-------------------+
10 rows in set (8.744 sec)
```

(8.7 seconds!) Shouldn't this be able to use `tl_backlinks_namespace (tl_from_namespace, tl_namespace, tl_title, tl_from)` efficiently?

Lucas_Werkmeister_WMDE added a comment. Edited · 2022-02-01 19:21:19 (UTC+0)

> In ~~T297754#7668758~~, **@Lucas_Werkmeister_WMDE** wrote:

> *If we just revert, then links like `offset=0|120` will break :(*

Slightly tweaked revert patch with a bit of code to handle new-style URLs:

T297754-2.patch 10 KB
Download

Diff to original code:

git diff @~2

```
diff --git a/includes/specials/SpecialWhatLinksHere.php b/includes/specials/SpecialWhatLinksHere.php
index 4271ad76d0..f13be1538f 100644
--- a/includes/specials/SpecialWhatLinksHere.php
+++ b/includes/specials/SpecialWhatLinksHere.php
@@ -94,7 +94,7 @@ public function execute( $par ) {
                $opts->add( 'target', '' );
                $opts->add( 'namespace', '', FormOptions::INTNULL );
                $opts->add( 'limit', $this->getConfig()->get( 'QueryPageDefaultLimit' ) );
-               $opts->add( 'offset', 0 );
+               $opts->add( 'offset', '0' );
                $opts->add( 'from', 0 );
                $opts->add( 'dir', 'next' );
                $opts->add( 'hideredirs', false );
@@ -136,7 +136,16 @@ public function execute( $par ) {
                $opts->reset( 'from' );
                $dir = $from ? 'next' : $opts->getValue( 'dir' );
                // 'from' was included in result set, offset is excluded. We need to align them.
-               $offset = $from ? $from - 1 : $opts->getValue( 'offset' );
+               if ( $from ) {
+                       $offset = $from - 1;
+               } else {
+                       $offset = $opts->getValue( 'offset' );
+                       [ $offsetNs, $offsetPageID ] = explode( '|', $offset . '|' );
+                       if ( $offsetPageID !== '' ) {
+                               $offset = $offsetPageID;
+                       }
+                       $offset = (int)$offset;
+               }

                $this->showIndirectLinks(
                        0,
```

Feel free to deploy that.

Lucas_Werkmeister_WMDE added a comment. 2022-02-01 19:30:03 (UTC+0)

> In ~~T297754#7668774~~, **@Lucas_Werkmeister_WMDE** wrote:
>
> *Why is even this simple query slow on the stats machines?*

(That might just be a red herring – I re-ran it with profiling and supposedly it spent 3.158 out of 3.159 seconds “Sending data”.)

Lucas_Werkmeister_WMDE added a comment. 2022-02-01 19:50:53 (UTC+0)

```
if ( is_int( $namespace ) ) {
	$invert = $this->opts->getValue( 'invert' );
	if ( $invert ) {
		// Select all namespaces except for the specified one.
		// This allows the database to use the *_from_namespace index. (T241837)
		$namespaces = array_diff(
			$this->namespaceInfo->getValidNamespaces(), [ $namespace ] );
	} else {
		$namespaces = $namespace;
	}
```

I wonder if it would help if, when there *isn't* a namepace, we also add a namespace filter, with all the valid namespaces? Something like:

```
                        } else {
                                $namespaces = $namespace;
                        }
+               } else {
+                       // Select all namespaces.
+                       // This allows the database to use the *_from_namespace index. (T297754)
+                       $namespaces = $this->namespaceInfo->getValidNamespaces();
+               }
                $conds['redirect']['page_namespace'] = $namespaces;
                $conds['pagelinks']['pl_from_namespace'] = $namespaces;
                $conds['templatelinks']['tl_from_namespace'] = $namespaces;
                $conds['imagelinks']['il_from_namespace'] = $namespaces;
-               }

                if ( $offset ) {
                        $rel = $dir === 'prev' ? '<' : '>';
```

Lucas_Werkmeister_WMDE added a comment. 2022-02-01 19:54:01 (UTC+0)

Yeah, I think that helps.

```
SELECT /* SpecialWhatLinksHere::showIndirectLinks  */
page_id,page_namespace,page_title,rd_from,rd_fragment,page_is_redirect  FROM (SELECT
tl_from,rd_from,rd_fragment  FROM `templatelinks` LEFT JOIN `redirect` ON ((rd_from = tl_from) AND
rd_title = 'Coord' AND rd_namespace = 10 AND (rd_interwiki = '' OR rd_interwiki IS NULL))   WHERE
tl_namespace = 10 AND tl_title = 'Coord' AND tl_from_namespace IN (0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14, 15, 100, 101, 118, 119, 710, 711, 828, 829, 2300, 2301, 2302, 2303)  ORDER BY
tl_from_namespace ASC,tl_from ASC LIMIT 102  ) `temp_backlink_range` JOIN `page` ON ((tl_from =
page_id))    ORDER BY page_namespace ASC,page_id ASC LIMIT 51;
```

Taken from reqId 267f30ba-33ae-4ef1-8646-8846f083506c, but with all the namespaces added. Finished in 0.006 seconds against enwiki, where the original logged actualSeconds=59.8.

Lucas_Werkmeister_WMDE added a comment. 2022-02-01 19:58:14 (UTC+0)

Follow-up patch, intended **instead of** the revert in ~~**T297754#7668789**~~:

T297754-2.patch 1 KB
Download

We can also go with the revert first, if you want to be safer. (This patch should even still apply then, apart from some line numbers, and probably wouldn't hurt.)

There's a scap underway right now, so I won't deploy this just now.

Lucas_Werkmeister_WMDE added a comment. 2022-02-01 21:14:08 (UTC+0)

> In ~~T297754#7668891~~, **@Lucas_Werkmeister_WMDE** wrote:
>
> *Follow-up patch, intended **instead of** the revert in **~~T297754#7668789~~**:*
>
> ***T297754-2.patch*** *1 KB*
> *Download*

After testing this patch on mwdebug1001 (and with clearance by the train conductors), I've deployed this to wmf.19 and wmf.20. Let's see if it works.

Lucas_Werkmeister_WMDE moved this task from **Tech Verification** to **Our work done** on the ~~**Wikidata-Campsite (Team A Hearth)**~~ board. 2022-02-07 10:58:42 (UTC+0)

So far it looks like the patches are working well. I suggest we squash both patches into a single change for public release, since the first one on its own produces bad behavior when not filtering by namespace; to that end, the following patch **squashes** **~~T297754#7626820~~** and **~~T297754#7668891~~** with a combined commit message and fresh Change-Id (but does not include the revert at **~~T297754#7668789~~**, which we didn't end up deploying):

T297754-squash.patch 11 KB
Download

But I'll leave `/srv/patches` as it is, and if the security team prefer to push the commits to Gerrit as they are there, then that's fine with me as well.

sbassett added a subscriber: **Reedy**. 2022-02-09 18:16:34 (UTC+0)

> In ~~T297754#7669191~~, **@Lucas_Werkmeister_WMDE** wrote:
>
> *After testing this patch on mwdebug1001 (and with clearance by the train conductors), I've deployed this to wmf.19 and wmf.20. Let's see if it works.*

Thanks!

> *So far it looks like the patches are working well.*

Great.

> *I suggest we squash both patches into a single change for public release ... But I'll leave `/srv/patches` as it is, and if the security team prefer to push the commits to Gerrit as they are there, then that's fine with me as well.*

These are being tracked for the next security release at **~~T297830~~**, so we should be good there. I'm fine with squashing the patches, but **@Reedy** can ultimately make that decision as he generally organizes and completes all of the relevant backports for the security releases.

sbassett triaged this task as *Low* priority. 2022-02-09 18:18:58 (UTC+0)

sbassett changed Author Affiliation from N/A to WMF Technology Dept.

sbassett changed Risk Rating from N/A to Low.

Reedy closed this task as *Resolved*. 2022-03-20 12:46:09 (UTC+0)

Reedy mentioned this in **~~T297830: Tracking bug for MediaWiki 1.35.6/1.36.4/1.37.2~~**.

Closing for ease of tracking

Reedy added a subscriber: **gerritbot**. 2022-03-28 13:32:39 (UTC+0)

Reedy added a comment. 2022-03-28 13:40:32 (UTC+0)

It seems the backport of this to REL1_36/REL1_35 is very much dependant on **rMWb9c68590d68c: Use pagination on Special:Whatlinkshere based on offset/dir system**, which is somewhat of a "breaking change" to the parameter interface for Special:WhatLinksHere.

It seems likely that that commit introduced/exacerbated the issue, and probably isn't worth the backport to REL1_35/REL1_36.

Reedy renamed this task from *Whatlinkshere of heavily used properties in wikidata can be easily utilized as a DDoS vector* to *CVE-2022-: Whatlinkshere of heavily used properties in wikidata can be easily utilized as a DDoS vector*. 2022-03-28 13:52:49 (UTC+0)

Reedy mentioned this in **~~T297831: Obtain CVEs for 1.35.6/1.36.4/1.37.2 security releases~~**.

Reedy renamed this task from *CVE-2022-: Whatlinkshere of heavily used properties in wikidata can be easily utilized as a DDoS vector* to *CVE-2022-28204: Whatlinkshere of heavily used properties in wikidata can be easily utilized as a DDoS vector*. 2022-03-30 18:03:39 (UTC+0)

gerritbot added a comment. 2022-03-31 22:07:18 (UTC+0)

Change 775985 had a related patch set uploaded (by Reedy; author: Lucas Werkmeister (WMDE)):

[mediawiki/core@REL1_37] SECURITY: Sort Special:WhatLinksHere by namespace

https://gerrit.wikimedia.org/r/775985

gerritbot added a project: **Patch-For-Review**. 2022-03-31 22:07:20 (UTC+0)

gerritbot added a comment. 2022-03-31 22:19:56 (UTC+0)

Change 775993 had a related patch set uploaded (by Reedy; author: Lucas Werkmeister (WMDE)):

[mediawiki/core@master] SECURITY: Sort Special:WhatLinksHere by namespace

https://gerrit.wikimedia.org/r/775993

gerritbot added a comment. 2022-03-31 22:21:15 (UTC+0)

Change 775996 had a related patch set uploaded (by Reedy; author: Lucas Werkmeister (WMDE)):

[mediawiki/core@REL1_38] SECURITY: Sort Special:WhatLinksHere by namespace

https://gerrit.wikimedia.org/r/775996

gerritbot added a comment. 2022-03-31 22:49:30 (UTC+0)

Change 775985 **merged** by jenkins-bot:

[mediawiki/core@REL1_37] SECURITY: Sort Special:WhatLinksHere by namespace

https://gerrit.wikimedia.org/r/775985

gerritbot added a comment. 2022-03-31 22:49:53 (UTC+0)

Change 775996 **merged** by jenkins-bot:

[mediawiki/core@REL1_38] SECURITY: Sort Special:WhatLinksHere by namespace

https://gerrit.wikimedia.org/r/775996

gerritbot added a comment. 2022-03-31 22:57:06 (UTC+0)

Change 775993 **merged** by jenkins-bot:

[mediawiki/core@master] SECURITY: Sort Special:WhatLinksHere by namespace

https://gerrit.wikimedia.org/r/775993

Reedy changed the visibility from "**Custom Policy**" to "Public (No Login Required)". 2022-03-31 23:05:27 (UTC+0)

Reedy changed the edit policy from "**Custom Policy**" to "All Users".

Maintenance_bot removed a project: **Patch-For-Review**. 2022-03-31 23:10:38 (UTC+0)

Zabe added a subscriber: **Zabe**. 2022-03-31 23:15:24 (UTC+0)

matej_suchanek mentioned this in ~~**T301604: Pages are sorted by namespace then page ID rather than just page ID on Special:WhatLinksHere on Wikimedia wikis**~~. 2022-04-03 08:31:21 (UTC+0)

hashar mentioned this in ~~**T305440: ParseError: syntax error, unexpected '<<' (T_SL)**~~. 2022-04-05 10:20:38 (UTC+0)

Lens0021 mentioned this in **T221729: Filtering namespaces on "What links here:" times out: "PHP fatal error: entire web request took longer than 60 seconds and timed out"**. 2022-10-01 15:45:45 (UTC+0)