

How changing '<' to '>=' introduced a weird and hard to track bug

Date: 2022-03-22
modified: 2022-03-22
tags: Uyuni, reposync, urlgrabber, bug, proxy
description: Even small changes can have a massic and unexpected impact
category: Code
slug: how_changing_less-than_to_greater-equal_introduced_a_weird_and_hard_to_track_bug
Author: Dominik Wombacher
lang: en
transid: how_changing_less-than_to_greater-equal_introduced_a_weird_and_hard_to_track_bug
Status: published

After upgrading to [Uyuni 2022.2](#) I wasn't able to sync openSUSE Leap 15.3 and Oracle Linux 8 repositories anymore, the configured HTTP Proxy was ignored. Multiple people reported similar issues: [#4932](#) (Archive: [\[1\]](#), [\[2\]](#)), [#4850](#) (Archive: [\[1\]](#), [\[2\]](#)), [#4826](#) (Archive: [\[1\]](#), [\[2\]](#)).

During multiple troubleshooting sessions I learned a lot about the Python Codebase and the reposync tool.

It took my quite a while to track down the Issue to the `urlgrabber` package.

What happened? Method `find_proxy` in `urlgrabber.grabber.URLGrabberOptions` will identify the proxy server that should be used based on the provided URL Scheme.

Let's assume `server.satellite.http_proxy` in `/etc/rhn/rhn.conf` is set to `http://10.11.12.13:80`.

```
>>> from urlgrabber.grabber import URLGrabberOptions
>>> opts = URLGrabberOptions(proxy=None, proxies={'http': 'http://10.11.12.13:80', 'https': 'http://10.11.12.13:80', 'ftp': 'http://10.11.12.13:80'})
>>> opts.find_proxy(b'http://test.example.com', b'http')
```

That's the simplified version of the relevant Code, a Instance of `URLGrabberOptions` contains a list of proxies, one per URL Scheme, the method `find_proxy` is used to choose the right proxy from `opts.proxies` based on two parameter, `url` and `scheme`, both are passed as type bytes.

The expected result, `opts.proxy` contains the value `http://10.11.12.13:80` from type `str`, actual result, `None`.

The package `urlgrabber` is quite old and not that actively maintained. Python 3 support was added in Version 4, the last release is from October 2019.

It looks like that the above issues, that proxy is `None`, comes from an inconsistent bytes / string conversion. If you pass the scheme as string instead bytes, you get the expected result:

```
>>> from urlgrabber.grabber import URLGrabberOptions
>>> opts = URLGrabberOptions(proxy=None, proxies={'http': 'http://10.11.12.13:80', 'https': 'http://10.11.12.13:80', 'ftp': 'http://10.11.12.13:80'})
>>> opts.find_proxy(b'http://test.example.com', 'http')
>>> opts.proxy
'http://10.11.12.13:80'
```

To get the Sync of openSUSE Leap 15.3 and Oracle Linux 8 Repositories working again through a http proxy, a small change was already sufficient:

```
diff --git a/backend/satellite_tools/download.py b/backend/satellite_tools/download.py
index 3d064e5c6ce..3b7a02d5176 100644
--- a/backend/satellite_tools/download.py
```


Based on my tests, by syncing *openSUSE Leap 15.3*, *Oracle Linux 8*, *CentOS 7* and *Ubuntu 20.04* repositories, it should finally resolve all, so far known, Issues related to reposync and HTTP Proxy.

And what had all this to do with a change of '<' to '>='?

In PR [#4604](#) (Archive: [\[1\]](#), [\[2\]](#)) the version was bumped, changing `python3-urlgrabber < 4` to `python3-urlgrabber >= 4` caused all that trouble and lot of issues where syncing repositories behind a http proxy was just not possible anymore.