New issue

Jump to bottom

Upload delay resulting in DoS #33

Open **mmmds** opened this issue on Feb 17, 2020 · 3 comments

mmmds commented on Feb 17, 2020

There is a vulnerability which allows to perform DoS attack against the application server. The problem lies in handling `delay` parameter when upload is initiated (`gwtupload.server.UploadServlet#parsePostRequest`). Value from this parameter is used as an argument for `Thread.sleep` invocation. Malicious user can specify even max integer value 2147483647, which would cause a thread to sleep for almost 25 days (`gwtupload.server.AbstractUploadListener#update`). Additionally, the value from `delay` parameter is assigned the field which in case of servlets behaves as a global variable. It means every further request will use this value and also will be put to sleep. Putting a thread to sleep excludes it from a limited set of available threads, so after a suitable number of upload requests (Tomcat by default has limit of 200 threads) the whole application will become unresponsive and will not accept any new requests.

https://github.com/manolo/gwtupload/blob/master/core/src/main/java/gwtupload/server/UploadServlet.java

```
protected String parsePostRequest(HttpServletRequest request, HttpServletResponse response) {

    try {
      String delay = request.getParameter(PARAM_DELAY);
      String maxFilesize = request.getParameter(PARAM_MAX_FILE_SIZE);
      maxSize = maxFilesize != null && maxFilesize.matches("[0-9]*") ? Long.parseLong(maxFilesize) : maxSize;
      uploadDelay = Integer.parseInt(delay);
    } catch (Exception e) { }
[...]
  protected AbstractUploadListener createNewListener(HttpServletRequest request) {
    int delay = request.getParameter("nodelay") != null ? 0 : uploadDelay;
    if (isAppEngine()) {
      return new MemoryUploadListener(delay, getContentLength(request));
    } else {
      return new UploadListener(delay, getContentLength(request));
    }
  }
```

https://github.com/manolo/gwtupload/blob/master/core/src/main/java/gwtupload/server/AbstractUploadListener.java

```
    // Just a way to slow down the upload process and see the progress bar in fast networks.
    if (slowUploads > 0 && done < total) {
      try {
        Thread.sleep(slowUploads);
      } catch (Exception e) {
        exception = new RuntimeException(e);
      }
```

The same way the servlet accepts `maxFileSize` parameter, but its abuse will only prevent from uploading files; the server won't suffer.

mmmds commented on Jun 7, 2020 Author

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-13128

Mik317 mentioned this issue on Jun 18, 2020

[Fix DOS issue] Updating the AbstractUploadListener.java file 418sec/gwtupload#1

Merged

jduo added a commit to jduo/gwtupload that referenced this issue on Jun 19, 2020

`Fix manolo#33, CVE-2020-13128, DOS attack due to delay param` ··· `e05acf5`

jduo mentioned this issue on Jun 19, 2020

Fix #33, CVE-2020-13128, DOS attack due to delay param 418sec/gwtupload#2

Closed

huntr-helper commented on Jun 19, 2020

A fix has been provided for this issue. Please reference: 418sec#1

This fix has been provided through the https://huntr.dev/ bug bounty platform.

huntr-helper mentioned this issue on Jun 19, 2020

Security Fix for Denial of Service (DoS) - huntr.dev #35

Merged

csware commented on Feb 14, 2021 Contributor

Fix merged, can be closed

luchua-bc mentioned this issue on Sep 19, 2021

[Java] CWE-400: Query to detect uncontrolled thread resource consumption github/securitylab#431

Closed

1 task

github-actions bot mentioned this issue 2 weeks ago

Add a PR check to ensure query IDs are unique github/codeql#11574

Merged

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants