TARTU ÜLIKOOL
Loodus- ja täppisteaduste valdkond
Arvutiteaduse Instituut

Siim Karel Koger
CVE-2009-1692
Referaat

Tartu 2019
Overview of the vulnerability

Affected products

- Internet Explorer 5, 6, 7, 8 (all versions)
- Chrome (limited)
- Opera
- Seamonkey
- Midbrowser
- Netscape 6 & 8 (9 years ago)
- Konqueror (all versions)
- Apple iPhone + iPod
- Apple Safari
- Thunderbird
- Nokia Phones : Nokia N95 (Symbian OS v.9.2), Nokia N82, Nokia N810 Internet Tablet
- Aigo P8860 (Browser hangs and cannot be restarted)
- Siemens phones
- Google Android G1 (TC4-RC30 & Firmware 1.5, Kernel: 2.6.27-00393-g6607056, Build: CRB4)
  G-SEC thanks Scott Fraser for the notice
- Blackberry 8800 & 8130 (Browser crash due to null ptr deref)
  G-SEC thanks "528-0444" & "Tyler Reguly" for the notice
- possibly more devices and products that support Javascript.

Description of the problem

G-SEC published a vulnerability in 2009 that causes browser to consume so much memory that the computer becomes virtually unusable. The impact varies between browsers and in some cases between operating systems - from null pointer dereference, which ends up eating up all the memory and crashing the browser to the reboot of the complete OS (like Ubuntu and Konqueror). It didn’t seem to have more of a negative impact other than uncomfortable user experience however.

To reproduce the problem is very simple. Using JavaScript DOM create a select element and assign a very high length value to it. From exploitation point of view this code would have been stored on a website which a user has to visit to trigger the memory usage.

```html
<script>
function bug() {
e = document.createElement("select");
```
Calling the `select()` method on a select element with a large number of options, results in continuous allocation of x+n bytes of memory exhausting memory after a while.

**Solution**

After the report of this bug, vendors apparently agreed to a limit of 10,000 elements. Apparently patches were sent out for most of the applications mentioned in the list above. However, during the research, it seems that the problem is still producible through Chrome DevTools today. After Chrome memory usage hit 5GB, the browser crashed with the following message.

I didn’t find any more information on the problem in 2019 context, but it seems as if the best solution is to just limit the options to 10,000 elements as mentioned in the article [https://bugzilla.mozilla.org/show_bug.cgi?id=460713](https://bugzilla.mozilla.org/show_bug.cgi?id=460713) and code snippet [https://bug460713.bmoattachments.org/attachment.cgi?id=345238](https://bug460713.bmoattachments.org/attachment.cgi?id=345238).
References

https://bugzilla.mozilla.org/show_bug.cgi?id=460713
http://www.g-sec.lu/one-bug-to-rule-them-all.html
https://www.exploit-db.com/exploits/9160