RBS-2013-004
Critical Vulnerabilities in the MyHeritage Search Engine
ActiveX Component
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>2</td>
</tr>
<tr>
<td>About Risk Based Security</td>
<td>3</td>
</tr>
<tr>
<td>Mission</td>
<td>3</td>
</tr>
<tr>
<td>Background</td>
<td>3</td>
</tr>
<tr>
<td>Discriminators</td>
<td>3</td>
</tr>
<tr>
<td>Vulnerable Program Details</td>
<td>4</td>
</tr>
<tr>
<td>References</td>
<td>4</td>
</tr>
<tr>
<td>Credits</td>
<td>4</td>
</tr>
<tr>
<td>Introduction</td>
<td>5</td>
</tr>
<tr>
<td>Vulnerability Details</td>
<td>6</td>
</tr>
<tr>
<td>Solution</td>
<td>11</td>
</tr>
<tr>
<td>Timeline</td>
<td>12</td>
</tr>
</tbody>
</table>
About Risk Based Security

Mission
To equip clients with the technology and customized risk-based consulting solutions to turn security data into information and information into a competitive advantage.

Background
Risk Based Security, Inc., incorporated in 2011, was established to better support the users/contributors to the Open Security Foundation, OSF, with the technology to turn security data into a competitive advantage.

The OSF’s wealth of historical data, combined with the interactive dashboards and analytics offered by Risk Based Security provide a first of its kind risk identification and security management tool.

Risk Based Security further complements the data analytics with risk-focused consulting services to address industry specific information security and compliance challenges.

Discriminators
Risk Based Security offers a full set of analytics and user-friendly dashboards designed specifically to identify security risks by industry.

Risk Based Security is the only company that offers its clients a fully integrated solution – real time information, analytical tools and purpose-based consulting.

Unlike other security information providers, Risk Based Security offers companies comprehensive insight into data security threats and vulnerabilities most relevant to their industry.
Vulnerable Program Details

Vendor: MyHeritage
Product: SearchEngineQuery.dll (ActiveX component)
Tested Version: 1.0.2.0
Tested Platform: Windows XP and Windows 7

References

RBS: RBS-2013-004
OSVDB: 934981, 934992, 935003, 935014
CVE: CVE-2013-2602

Credits

Carsten Eiram, Risk Based Security

Twitter: @CarstenEiram
Twitter: @RiskBased

1 http://osvdb.org/show/osvdb/93498
2 http://osvdb.org/show/osvdb/93499
3 http://osvdb.org/show/osvdb/93500
4 http://osvdb.org/show/osvdb/93501
**Introduction**

MyHeritage\(^5\) is a genealogy website and social network, which reportedly\(^6\) has “more than 64 million members with more than 1 billion online profiles”.

MyHeritage provides various genealogy search features including an ActiveX component, which is installed on users’ systems from the MyHeritage website. The latest version of the ActiveX component is version 1.0.2.0 and was\(^7\) available at:

http://www.myheritage.com/Genoogle/Components/ActiveX/SearchEngineQuery.dll

SearchEngineQuery.dll provides the SEQueryObject ActiveX control, which is registered as “safe” and is not SiteLocked. This allows any website to instantiate the ActiveX control in e.g. Internet Explorer and repurpose the features provided by it.

The ActiveX component is part of a product called Genoogle, which according to the CEO of MyHeritage, Gilad Japhet, via e-mail communication was primarily offered to users during 2006 to 2008 until eventually discontinued in April 2012. According to the CEO’s estimates, the total user base would be in the low tens of thousands with the number of affected users to date likely being much lower.

MyHeritage, however, never set the kill-bit for the ActiveX control nor ensured the component was removed from users’ systems once Genoogle was discontinued. Since users rarely remove files from the “Downloaded Program Files” folder, which SearchEngineQuery.dll is installed to, the component is expected to still linger on many systems.

This analysis describes multiple critical vulnerabilities affecting 19 parameters across four methods provided by the SEQueryObject ActiveX control. These can be exploited to execute arbitrary code on a MyHeritage user’s system by e.g. tricking the user into viewing a malicious web page.

---

\(^5\) http://www.myheritage.com/
\(^6\) http://en.wikipedia.org/wiki/MyHeritage
\(^7\) The ActiveX component was removed after Risk Based Security contacted MyHeritage and informed about the critical vulnerabilities.
Vulnerability Details

The four methods of particular interest to this analysis are:

1. AddTokens
2. AddLastNameTokens
3. AddMultipleSearches
4. TestYourself

Each of these methods accept one or more VARIANT arguments that are treated as object pointers and dereferenced without any validation, allowing control of the program flow to be gained in a straightforward manner.

The following analysis specifically focuses on the “AddLastNameTokens” method’s “seLastNameTokensArray” parameter, however, this is only one of many affected parameters across the four methods. The complete list of the 19 affected parameters grouped by 4 vulnerable methods are:

1. AddLastNameTokens
   - seLastNameTokensArray

2. AddTokens
   - seTokensArray
   - seTokensValuesArray

3. AddMultipleSearches
   - seFrameIdArray
   - seSourceIdArray
   - seHasBreakdownArray
   - seIsIndexedArray
   - seAllConcatArray
   - seRefererURLArray
   - seMandatoryFieldsArray

4. TestYourself
   - seSourceIdArray
   - seIsIndexedArray
   - seAllConcatArray
   - seRefererURLArray
   - seQATestsArray
The “AddLastNameTokens()” method takes two arguments, “ulNumberOfLastNameTokens” and “seLastNameTokensArray”, as shown below.

When visiting a web page that instantiates the SEQueryObject ActiveX control and calls the “AddLastNameTokens()” method, the “InvokeDispatchWithNoThis()” function in mshtml.dll calls the dispatcher in SearchEngineQuery.dll.

The dispatcher checks the number of supplied parameters to ensure that no more than 16 were provided...
... before storing references to these into an array.

In case the “AddLastNameTokens()” method was called, the code branches to check that only two arguments were provided.
These arguments are both handled as VARIANTs. The first, ulNumberOfLastNameTokens, is checked to ensure it’s of vartype VT_I4 while the second, seLastNameTokensArray, is anything but VT_BYREF | VT_VARIANT.
Control is then passed to the internal function associated with the “AddLastNameTokens()” method.

This function eventually checks if a “seLastNameTokensArray” argument was supplied and, if so, dereferences it in an attempt to call a virtual function. As the value is attacker-supplied, this allows calling an arbitrary memory location to execute code.
This can be demonstrated by supplying 0x41414141 as the “seLastNameTokensArray” argument to the “AddLastNameTokens()” method. Code execution from there is straightforward.

```asm
(2f4.2b4): Access violation - code c0000005 (!!! second chance !!!)
  eax=0000d5a1 ebx=41414141 ecx=10045744 edx=1003c058 esi=00000000 edi=41414141
  eip=10016edd esp=0013b9c4 ebp=0013bb04 iopl=0 npx mp pe m x save ds:0023 es:0023 fs:003b gs:0000
  efl=00000206
*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Doc
SearchEngineQuery!DllRegisterServer+0x1041d:
  10016edd 0b07          mov     eax.dword ptr [edi]  ds:0023:41414141-????????
  0:000>
SearchEngineQuery!DllRegisterServer+0x1041d:
  10016edd 0b07          mov     eax.dword ptr [edi]
  10016edf 57           push    edi
  10016ee0 ff5004        call    dword ptr [eax+4]
  10016ee3 33db          xor     ebx,ebx
```

Similar behaviour can be observed for the other 18 parameters in the other three methods.

**Solution**

The vendor immediately removed the ActiveX component from the MyHeritage website once informed about the vulnerabilities.

RBS further advised MyHeritage to contact Microsoft in order to have the kill-bit set for the ActiveX control in the May 2013 Update Rollup for ActiveX Kill Bits. While the vendor stated that they would do so, the kill-bit was not set in the latest Update Rollup, but may be set at a later stage.

As a workaround, users can set the kill-bit for the SEQQueryObject ActiveX Control (CLSID: {6218F7B5-0D3A-48BA-AE4C-49DCFA63D400}). Refer to Microsoft KB article 240797 for additional details. As the ActiveX component is not required and is part of a discontinued product, it is recommended to also remove it from the system. Usually, the file can be found in the “Downloaded Program Files” directory.

---

8 http://support.microsoft.com/kb/240797
<table>
<thead>
<tr>
<th>Date</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013/03/05</td>
<td>Vulnerabilities discovered.</td>
</tr>
<tr>
<td>2013/03/07</td>
<td>Security contact information requested from vendor via online form.</td>
</tr>
<tr>
<td>2013/03/15</td>
<td>Security contact information requested from vendor via e-mail.</td>
</tr>
<tr>
<td>2013/03/15</td>
<td>Response from company CEO and contact details provided.</td>
</tr>
<tr>
<td>2013/03/17</td>
<td>Vulnerability details provided to CEO, CTO, and COO as per request.</td>
</tr>
<tr>
<td>2013/03/17</td>
<td>Vendor acknowledges receiving report and provides information about the product using it being discontinued.</td>
</tr>
<tr>
<td>2013/03/18</td>
<td>Considering the discontinued status of the ActiveX control, RBS recommends MyHeritage to contact Microsoft to have the kill-bit set.</td>
</tr>
<tr>
<td>2013/03/18</td>
<td>MyHeritage informs that the ActiveX control has been removed from the website and that Microsoft will be requested to set the kill-bit.</td>
</tr>
<tr>
<td>2013/05/14</td>
<td>Microsoft releases Update Rollup for ActiveX Kill Bits, but the kill-bit is not set as expected.</td>
</tr>
<tr>
<td>2013/05/20</td>
<td>Public disclosure.</td>
</tr>
</tbody>
</table>