SolrSecurity

Solr Security

The authoritative guide on implementing security is in the Solr Reference Guide. This page describes security features in general, but also provides information about CVEs that have been patched or dependencies which do not require a patch for Solr.

Reported vulnerabilities (CVEs) are listed on the security news section on Solr's website.

Known false positives, which used to be listed on this wiki page, are also now listed on the Security web page.

If you believe you have discovered a vulnerability in Solr, please follow these ASF guidelines for reporting it.

Need for firewall

Even though you add SSL or Authentication plugins, it is still strongly recommended that the application server containing Solr be firewalled such the only clients with access to Solr are your own. A default/example installation of Solr allows any client with access to it to add, update, and delete documents (and of course search/read too), including access to the Solr configuration and schema files and the administrative user interface.

If there is a need to provide query access to a Solr server from the open internet, it is highly recommended to use a proxy, such as one of these.

Cross-Site Request Forgery (CSRF)

Even if a Solr instance is protected by good firewalls so that "bad guys" have no direct access, that instance may be at risk to potential "Cross-Site Request Forgery" based attacks if the following are all true:

1. Some number of "good guys" have direct access to that Solr instance from their web browsers.
2. A "bad guy" knows/guesses the host:port/path of the Solr instance (even though they can not access it directly)
3. The bad guy can trick one of the good guy into clicking a maliciously crafted URL, or loading a webpage that contains malicious javascript.

This is because Solr's most basic behavior is to receive updates and deletes via HTTP. If you have a firewall or other security measure restricting Solr's /update handler so it only accepts connections from approved hosts/clients, but *you* are approved then you could inadvertently be tricked into loading a web page that initiates an HTTP Connection to Solr on your behalf.

It's important to keep this in mind when thinking about what it means to "secure" an instance of Solr (if you have not already).

A basic technique that can be used to mitigate the risk of a possible CSRF attack like this is to configure your Servlet Container so that access to paths which can modify the index (ie: /update, /update/csv, etc...) are restricted either to specific client IPs, or using HTTP Authentication.

Streaming Consideration

If streaming is enabled, you need to make sure Solr is as secure as it needs to be. When streaming is enabled, the parameters "stream.url" will go to a remote site and download the content. Likewise, "stream.file" will read a file on disk.

Streaming is disabled by default and is configured from solrconfig.xml

```
<requestParsers enableRemoteStreaming="false" ... />
```

Indirect compromise through Tika vulnerabilities

One of the contrib modules that Solr includes is called SolrCell. This module adds the Extracting Request Handler. This component utilizes Apache Tika to parse rich documents like PDF and Microsoft Office and index the document contents into Solr.

The Tika software has had some security vulnerabilities. It would be theoretically possible for an attacker to upload a specially crafted file to be processed by Tika running inside Solr, or to trick an administrator into uploading such a file, and in that way compromise the Solr install.

For reasons not related to security, it is strongly recommended that this contrib module is never used in production. Tika can crash, and if such a crash happens in the SolrCell module, Solr will crash too. If that advice is followed, it would be very difficult to utilize Tika vulnerabilities to compromise Solr.

Solr and Vulnerability Scanning Tools

Many organizations have policies where software to be installed on the network must pass an examination by a vulnerability scanner which attempts to determine if there are known vulnerabilities in the application.

Solr includes many dependencies which may trigger warnings from a vulnerability scan but which the Solr community has determined that they are false positives. As a general rule, the Solr PMC will not accept the output of a vulnerability scan as a security report.

See the web site Security page for more details on Solr's security status. **NOTE:** The table that used to be on this page is now on the web site