# Procedure for Elasticsearch reindex/migration

## Using the Reindex API

The following are the steps to be taken in order for Elasticsearch and its data to meet the new requirements. You should make sure that there is available disk space for at least twice the size of the type's index.

1. Create and initialize a temporary index to reindex to.
   According to the documentation for Elasticsearch 5.4, the Reindex API does not attempt to set up the destination index (copy the existing settings of the source index like mappings, shard counts, replicas etc). Therefore, the temporary/destination index should be set up prior to executing a reindex action.
   - Use the right settings when creating the new/temporary index
     **For example, to find the cspdata index settings, open the file "init_csp_index.sh" within the sxcsp repository and the deployments/elasticsearch directory. In the body of the CURL call under "Settings Configuration" you may find the index settings (if there are any specified, otherwise there will be just an empty settings block).
     Here is an example creating a temporary index named "cspdata_new" with no settings.

```
curl -XPUT  "localhost:9200/cspdata_new?pretty" -H 'Content-Type:
application/json' -d'
{
    "settings" : {
    }
}
'
```

   - Create all the mappings (including the new mapping for the given type, replacing the old mapping)
     **A quick way for figuring out the mapping content of the CURL command with the new mapping, is by dynamically creating a type on temporary index (by inserting a sample document) and retrieving its mapping using:

```
# Artefact type example
curl -XGET 'localhost:9200/temp_index/_mapping/artefact?pretty'
```

2. Reindex either by:
   - Filtering out the old type and its documents from the new index

```
# Reindexing leaving out old Taranis vulnerability type example
curl -X POST "localhost:9200/_reindex?pretty" -H 'Content-Type: application/json'
-d'
{
  "source": {
    "index": "cspdata",
      "query": {
        "bool" : {
          "must_not" : {
            "term" : {
              "_type" : "vulnerability"
            }
          }
```

```
        }
      }
  },
  "dest": {
    "index": "cspdata_new",
    "version_type": "external"
  }
}
'
```

- Using the Ingest Node feature for pre-processing documents on the new index before the actual indexing takes place.
  Using this feature you can add/remove/rename certain fields in the new index for a given type from the reindex source. This can be useful in situations where you need to keep any old documents in the new index using only the fields defined in the new mapping.

3. Delete the old index (only after reindexing successfully)

```
curl -X DELETE "localhost:9200/cspdata?pretty"
```

4. Recreate the old index with the right settings and the proper name
5. Reindex the temporary index(source) to the recreated old index(target).

```
# Reindexing back to the index with the proper name (“cspdata”) example
curl -X POST "localhost:9200/_reindex?pretty" -H 'Content-Type: application/json' -d'
{
  "source": {
    "index": "cspdata_new"
  },
  "dest": {
    "index": "cspdata"
  }
}
'
```

6. Delete the temporary index