UES 安全插件使用说明

一、特别提示

1）安装安全插件将会触发 UES 集群重启，对 ES 服务造成影响．请在安装之前做好必要的准备工作，选择合适的时间进行安装．

2）安全插件安装完成之后，访问 ES 服务的代码需要修改，加上账号密码信息方可正常使用。

二、插件安装

1.支持的版本

当前支持安全插件的 UES 服务版本为 6.5.4，其他版本的 UES 暂不支持安装安全插件。

2.安全插件安装方法

1）在 UES 控制台的集群管理页面，点击“更多-插件管理”，进入插件管理页面。

2）在插件管理页面找到安全插件“ucloud_ues_security”，点击“安装”按钮。

3）在弹出的对话框中点击“确定”，开始进行安全插件的安装。

3.注意事项

1）安全插件的安装采用逐个节点进行安装并重启的方式来进行，一个节点的安全插件安装完成后，由于安全机制发生变化，将无法加入原集群。需要等待所有节点的安全插件都安装完成之后，集群才能恢复提供服务，一般情况下，拥有 3 个节点的集群安装安全插件大约需要 15 分钟时间。

2）安全插件安装完成之后，访问 ES 服务的代码需要修改，加上账号密码信息方可正常使用（默认沿用创建集群时设置的 Kibana 的账号密码作为启用安全机制后的账号密码）。示例如下：

curl -H "Content-Type: application/json" -u admin:admin -XGET http://localhost:9200/_cat/health?v

三、安全——访问控制

1.概念

术语	描述
Permission	单个动作，例如创建索引（例如 `indices:admin/create`）。
Action group	一组权限。例如，预定义的 `SEARCH` 操作组授权角色使用 `_search` 和 `_msearch` API。
Role	安全角色定义权限或操作组的范围：集群，索引，文档或字段。例如，名为的角色 `delivery_analyst` 可能没有集群权限，`READ` 与 `delivery-data-*` 模式匹配的所有索引的操作组，对这些索引中的所有文档类型的访问权限以及除外的所有字段的访问权限 `delivery_driver_name`。
Backend role	（可选）来自授权后端的其他外部角色（例如 LDAP / Active Directory）。
User	用户向 Elasticsearch 集群发出请求。用户具有凭证（例如，用户名和密码），零个或多个后端角色以及零个或多个自定义属性。
Role mapping	用户在成功进行身份验证后会担任角色。角色映射，将角色映射到用户（或后端角色）。例如，`kibana_user`（角色）到 `jdoe`（用户）的映射意味着 John Doe 在获得 `kibana_user` 身份验证后获得了所有权限。同样，`all_access`（角色）到 `admin`（后端角色）的映射意味着具有后端角色 `admin`（来自 LDAP / Active Directory 服务器）的任何用户都获得了 `all_access` 身份验证后的所有权限。您可以将每个角色映射到许多用户和/或后端角色。

2.用户和角色

角色是控制集群访问的核心方式。角色包含集群范围权限，特定于索引的权限，文档和字段级安全性以及租户的任意组合。然后，您将用户映射到这些角色，以便用户获得这些权限。Security 插件附带了许多预定义的操作组，角色，映射和用户。这些实体作为合理的默认值，是如何使用插件的很好的例子。

以下的操作如果没有特别说明，均在 Kibana 中进行。

1） 创建用户

(1)选择“Security”-“Internal User Database”-“Add a new internal user”。

(2)输入用户名和密码。

(3)如果需要，请指定后端角色和属性。后端角色与安全角色不同。后端角色是来自外部身份验证系统（例如 LDAP / Active Directory）的外部角色。如果您不使用外部系统，则可以忽略后端角色。属性是可选的用户属性，可用于索引权限或文档级安全性中的变量替换。

(4)点击提交，完成用户创建。

2) 创建角色

(1)选择“Security”-“Roles”-“Add a new role”。

(2)输入角色的名称

(3)在“Cluster Permissions”、“Index Permissions”、“DLS/FLS”、“Tenants”标签页中为该角色创建相应的权限。

(4)点击提交，完成角色创建。

3) 将用户映射到角色

创建角色后，您将用户（或后端角色）映射到它们。直觉上，人们通常认为这个过程是为用户提供一个或多个角色，但在安全插件中，过程是相反的:您选择一个角色，然后将一个或多个用户映射到该角色。

(1)选择“Security”-“Role Mappings”-“Add a new role mapping”。

(2)选择角色。如果角色是灰色的，则它的映射已存在。

(3)根据需要指定用户，后端角色（来自 LDAP 或 Active Directory 的角色）和主机（例如 *.devops.my-organization.org）。

(4)点击提交，完成角色映射配置。

3.文档级安全性

文档级安全性允许您将角色限制为索引中的文档子集。开始使用文档和字段级安全性的最简单方法是打开 Kibana 并选择 Security。然后选择 Roles，创建一个新角色，然后选择 DLS / FLS。

1) 简单角色

文档级安全性使用 Elasticsearch 查询 DSL 来定义角色授予访问权限的文档。在 Kibana 中，选择索引并在 DLS / FLS 选项卡中提供查询:

```
{
  "bool": {
    "must": {
      "match": {
        "genres": "Comedy"
      }
    }
  }
}
```

此查询指定对于角色有权访问文档，其 genres 字段必须包含 Comedy。

请注意，_search API 的典型请求包括{ "query": { ... } }查询，但在这种情况下，您只需要指定查询本身。

在 REST API 中，您将查询作为字符串提供，因此您必须转义引号。此角色允许用户读取任何索引中的任何文档，并将字段 public 设置为 true：

```
PUT _ucloud_ues/_security/api/roles/public_data
{
  "cluster" : [ "*" ],
  "indices" : {
    "pub*" : {
      "*" : [ "READ" ],
      "_dls_": "{\"term\": { \"public\": true}}"
    }
  }
```

```
}
```

2） 参数替换

存在许多变量，您可以使用这些变量根据用户的属性强制执行规则。例如，${user.name}替换为当前用户的名称。

此规则允许用户读取用户名为 readable_by 字段值的任何文档：

```
PUT _ucloud_ues/_security/api/roles/user_data

{

  "cluster" : [ "*" ],

  "indices" : {

    "pub*" : {

      "*" : [ "READ" ],

      "_dls_": "{ \"term\": { \"readable_by\": \"${user.name}\"}}"

    }

  }

}
```

存在以下替换：

术语	替换为
${user.name}	用户名
${user.roles}	逗号分隔的引用用户角色列表
${attr.<TYPE>.<NAME>}	<NAME>为用户定义名称的属性。<TYPE>是 internal, jwt 或 ldap

3） 基于属性的安全性

您可以使用 terms_set 查询的角色和参数替换来启用基于属性的安全性。

用户定义

```
PUT _ucloud_ues/_security/api/internalusers/user1
```

```
{

  "password": "asdf",

  "roles": ["abac"],

   "attributes": {

      "permissions": "\"att1\", \"att2\", \"att3\""

   }

}
```

角色定义

```
PUT _ucloud_ues/_security/api/roles/abac

{

    "indices" : {

      "*" : {

        "*" : ["READ"],

        "_dls_": "{\"terms_set\": {\"security_attributes\": {\"terms\": [${attr.internal.permissions}],

\"minimum_should_match_script\": {\"source\": \"doc['security_attributes'].values.length\"}}}}"

      }

    }

}
```

4.字段级安全性

字段级安全性允许您控制用户可以看到的文档字段。就像文档级安全性一样，您可以控制角色内的索引访问。

开始使用文档和字段级安全性的最简单方法是打开 Kibana 并选择 Security。然后选择 Roles，创建一个新角色，然后选择 DLS / FLS。

1） 包含或排除字段

配置字段级安全性时有两个选项：包含或排除字段。如果包含字段，则用户在检索文档时*仅会*看到这些字段。例如，如果包括 actors，title 和 year 领域，搜索结果可能是这样的：

```
{
  "_index": "movies",
  "_type": "_doc",
  "_source": {
    "year": 2013,
    "title": "Rush",
    "actors": [
      "Daniel Brühl",
      "Chris Hemsworth",
      "Olivia Wilde"
    ]
  }
}
```

如果排除字段，用户在检索文档时会看到*除*这些字段之外的所有字段。例如，如果排除那些相同的字段，则相同的搜索结果可能如下所示：

```
{
  "_index": "movies",
  "_type": "_doc",
  "_source": {
    "directors": [
```

```
      "Ron Howard"
    ],
    "plot": "A re-creation of the merciless 1970s rivalry between Formula One rivals James Hunt and
Niki Lauda.",
    "genres": [
      "Action",
      "Biography",
      "Drama",
      "Sport"
    ]
  }
}
```

您可以使用包含或排除来实现相同的结果，因此请选择对您的用例有意义的结果。混合这两者没有意义，也不受支持。

操作方式：登录 Kibana

(1)选择角色和 **DLS / FLS**。

(2)选择一个索引。

(3)在“Include or exclude fields”字段下，使用下拉列表选择首选选项。然后指定一个或多个字段并**“Save Role Definition”**。

2) 与多个角色互动

如果将用户映射到多个角色，我们建议这些角色对每个索引使用 include 或 exclude 语句。安全性插件使用 AND 运算符评估字段级安全性设置，因此组合 include 和 exclude 语句可能导致两种行为都无法正常工作。

例如，在 movies 索引中，如果您包含 actors，title 和 year 在一个角色中，排除 actors，title 和 genres 在另一个角色中，然后将两个角色映射到同一用户，搜索结果可能如下所示：

```
{
  "_index": "movies",
  "_type": "_doc",
  "_source": {
    "year": 2013,
    "directors": [
      "Ron Howard"
    ],
    "plot": "A re-creation of the merciless 1970s rivalry between Formula One rivals James Hunt and
Niki Lauda."
  }
}
```

3） 与文档级安全性的交互

文档级安全性依赖于 Elasticsearch 查询，这意味着查询中的所有字段必须可见才能使其正常工作。如果将字段级安全性与文档级安全性结合使用，请确保不限制对文档级安全性使用的字段的访问。

5.字段屏蔽

如果您不想使用字段级安全性从文档中删除字段，则可以屏蔽它们的值。目前，字段屏蔽仅适用于基于字符串的字段，并使用加密哈希替换字段的值。

字段屏蔽与字段级安全性在同一个每个角色，每个索引的基础上一起工作。您可以允许某些角色以纯文本格式查看敏感字段，并为其他角色屏蔽它们。带有屏蔽字段的搜索结果可能如下所示：

```
{
  "_index": "movies",
```

```
  "_type": "_doc",
  "_source": {
    "year": 2013,
    "directors": [
      "Ron Howard"
    ],
    "title": "ca998e768dd2e6cdd84c77015feb29975f9f498a472743f159bec6f1f1db109e"
  }
}
```

1） 配置字段屏蔽

登录 Kibana

(1)选择角色和 **DLS / FLS**。

(2)选择一个索引。

(3)在“**匿名**”字段下，指定一个或多个字段和“**保存角色定义**”。

2） （高级）使用备用哈希算法

默认情况下，安全性插件使用 BLAKE2b 算法，但您可以使用 JVM 提供的任何散列算法。此列表通常包括 MD5，SHA-1，SHA-384 和 SHA-512。

要指定其他算法，请在屏蔽字段后添加：

```
someonerole:
  cluster: []
  indices:
    movies:
      _masked_fields_:
```

```
      - "title::SHA-512"
      - "genres"
      '*':
      - "READ"
```

3) （高级）基于模式的字段屏蔽

您可以使用一个或多个正则表达式和替换字符串来屏蔽字段，而不是创建哈希。语法是<field>::/<regular-expression>/::<replacement-string>。如果使用多个正则表达式，结果将从左向右传递，如 shell 中的管道：

```
hr_employee:
  index_permissions:
    - index_patterns:
      - 'humanresources'
      allowed_actions:
        - ...
      masked_fields:
        - 'lastname::/.*/::*'
        - '*ip_source::/[0-9]{1,3}$/::XXX::/^[0-9]{1,3}/::***'
asdf
someonerole:
  cluster: []
  indices:
    movies:
```

```
_masked_fields_:
- "title::/./::*"
- "genres::/^[a-zA-Z]{1,3}/::XXX::/[a-zA-Z]{1,3}$/::YYY"
'*':
- "READ"
```

该 title 语句将字段中的每个字符更改为*，因此您仍然可以识别屏蔽字符串的长度。该 genres 语句将字符串的前三个字符更改为 XXX 最后三个字符 YYY。

4） 对审计记录的影响

通过读取历史记录功能，您可以跟踪对文档中敏感字段的读取访问权限。例如，您可以跟踪对客户记录的电子邮件字段的访问权限。对读取历史记录排除对掩码字段的访问，因为用户只看到了散列值，而不是字段的明文值。

6.Kibana 多租户

Kibana 中的*租户*是用于保存索引模式，可视化，仪表板和其他 Kibana 对象的空间。默认情况下，所有 Kibana 用户都可以访问两个租户：**私有**和**全局**。全局租户在每个 Kibana 用户之间共享。私有租户是每个用户独有的，无法共享。

租户对于与其他 Kibana 用户安全地共享您的工作非常有用。您可以控制哪些角色可以访问租户以及这些角色是否具有读取或写入访问权限。

您可以使用私有租户进行探索性工作，与 analysts 租户中的团队一起创建详细的可视化，并维护租户中公司领导的汇总仪表板 executive。

如果您与某人共享可视化文件或仪表板，则可以看到该 URL 包含租户：

```
http://<kibana_host>:5601/app/kibana?security_tenant=analysts#/visualize/edit/c501fa50-7e52-11e9-ae4e-b5d69947d32e?_g=()
```

1） 添加租户

在安全性插件中，租户只是角色的属性。要创建租户，请将其添加到 Kibana 中的角色。

- 读写（RW）权限允许角色查看和修改租户中的对象。
- 只读（RO）权限允许角色查看对象，但不能修改它们。

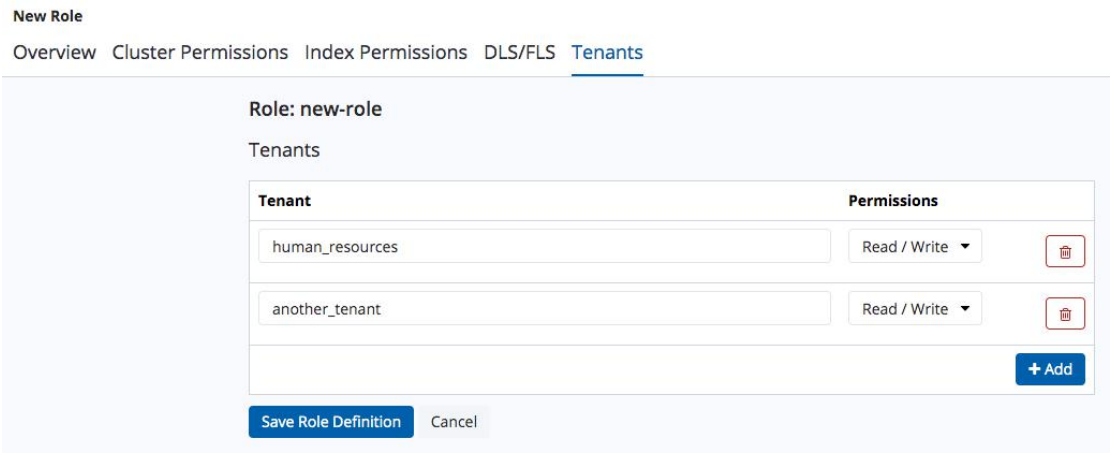
操作方法：

(1)登录 Kibana。

(2)选择**安全**和**角色**。

(3)创建新角色或选择现有角色。

(4)选择**租户**。



2） 管理 Kibana 索引

Kibana 的开源版本将所有对象保存到单个索引：.kibana。Security 插件将此索引用于全局租户，但为每个其他租户使用单独的索引。每个用户还有一个私有租户，因此您可能会看到大量索引遵循两种模式：

```
.kibana_<hash>_<tenant_name>

.kibana_<hash>_<username>
```

> 安全性插件会清除特殊字符的这些索引名称，因此它们可能不是租户名称和用户名的完美匹配。

7.权限

以下是安全性插件中可用权限的完整列表。每个权限都控制对数据类型或 API 的访问。

1） 集群

cluster:admin/ingest/pipeline/delete

cluster:admin/ingest/pipeline/get

cluster:admin/ingest/pipeline/put

cluster:admin/ingest/pipeline/simulate

cluster:admin/ingest/processor/grok/get

cluster:admin/reindex/rethrottle

cluster:admin/repository/delete

cluster:admin/repository/get

cluster:admin/repository/put

cluster:admin/repository/verify

cluster:admin/reroute

cluster:admin/script/delete

cluster:admin/script/get

cluster:admin/script/put

cluster:admin/settings/update

cluster:admin/snapshot/create

cluster:admin/snapshot/delete

cluster:admin/snapshot/get

cluster:admin/snapshot/restore

cluster:admin/snapshot/status

cluster:admin/snapshot/status*

cluster:admin/tasks/cancel

cluster:admin/tasks/test

cluster:admin/tasks/testunblock

cluster:monitor/allocation/explain

cluster:monitor/health

cluster:monitor/main

cluster:monitor/nodes/hot_threads

cluster:monitor/nodes/info

cluster:monitor/nodes/liveness

cluster:monitor/nodes/stats

cluster:monitor/nodes/usage

cluster:monitor/remote/info

cluster:monitor/state

cluster:monitor/stats

cluster:monitor/task

cluster:monitor/task/get

cluster:monitor/tasks/list

2) 索引

indices:admin/aliases

indices:admin/aliases/exists

indices:admin/aliases/get

indices:admin/analyze

indices:admin/cache/clear

indices:admin/close

indices:admin/create

indices:admin/delete

indices:admin/exists

indices:admin/flush

indices:admin/flush*

indices:admin/forcemerge

indices:admin/get

indices:admin/mapping/put

indices:admin/mappings/fields/get

indices:admin/mappings/fields/get*

indices:admin/mappings/get

indices:admin/open

indices:admin/refresh

indices:admin/refresh*

indices:admin/rollover

indices:admin/seq_no/global_checkpoint_sync

indices:admin/settings/update

indices:admin/shards/search_shards

indices:admin/shrink

indices:admin/synced_flush

indices:admin/template/delete

indices:admin/template/get

indices:admin/template/put

indices:admin/types/exists

indices:admin/upgrade

indices:admin/validate/query

indices:data/read/explain

indices:data/read/field_caps

indices:data/read/field_caps*

indices:data/read/get

indices:data/read/mget

indices:data/read/mget*

indices:data/read/msearch

indices:data/read/msearch/template

indices:data/read/mtv

indices:data/read/mtv*

indices:data/read/scroll

indices:data/read/scroll/clear

indices:data/read/search

indices:data/read/search*

indices:data/read/search/template

indices:data/read/tv

indices:data/write/bulk

indices:data/write/bulk*

indices:data/write/delete

indices:data/write/delete/byquery

indices:data/write/index

indices:data/write/reindex

indices:data/write/update

indices:data/write/update/byquery

indices:monitor/recovery

indices:monitor/segments

indices:monitor/settings/get

indices:monitor/shard_stores

indices:monitor/stats

indices:monitor/upgrade

8.默认操作组

此页面对所有默认操作组进行编目。通常，创建新操作组的最一致方法是使用这些默认组和单个权限的组合。

1） 一般

名称	描述
UNLIMITED	授予完全访问权限。可以在集群级别或索引级别使用。等于"*"。

2） 集群级

名称	描述
CLUSTER_ALL	授予所有集群权限。等于 cluster:*。
CLUSTER_MONITOR	授予所有集群监视权限。等于 cluster:monitor/*。
CLUSTER_COMPOSITE_OPS_RO	授予只读权限来执行请求，类似 mget，msearch 或者 mtv，加上权限查询别名。
CLUSTER_COMPOSITE_OPS	类同 CLUSTER_COMPOSITE_OPS_RO，但也授予 bulk 权限和所有别名权限。

MANAGE_SNAPSHOTS	授予管理快照和存储库的权限。

3) 索引级

名称	描述
INDICES_ALL	授予索引的所有权限。等于 `indices:*`。
GET	授予仅使用 `get` 和 `mget` 操作的权限。
READ	授予读取权限，例如搜索，获取字段映射 `get` 等 `mget`。
WRITE	授予对文档的写权限。
DELETE	授予删除文档的权限。
CRUD	组合 READ，WRITE 和 DELETE 操作组。
SEARCH	授予搜索文档的权限。包括 SUGGEST。
SUGGEST	授予使用建议 API 的权限。包含在 READ 操作组中。
CREATE_INDEX	授予创建索引和映射的权限。
INDICES_MONITOR	授予执行所有索引监视操作的权限（例如，恢复，段信息，索引统计和状态）。
MANAGE_ALIASES	授予管理别名的权限。
MANAGE	授予索引的所有监视和管理权限。