NAS

ONTAP Automation

NetApp
July 11, 2024

目錄

NAS

檔案安全性權限

準備好管理檔案安全性和稽核原則

您可以管理 ONTAP 叢集內透過 SVM 所提供檔案的權限和稽核原則。

總覽

使用系統存取控制清單（SACL）和自由存取控制清單（DACL）來指派檔案物件的權限。ONTAP從 ONTAP 9.9.1 開始、其餘 API 支援管理 SACL 和 DACL 權限。您可以使用 API 來自動化檔案安全性權限的管理。在許多情況下、您可以使用單一 REST API 呼叫、而非多個 CLI 命令或 ONTAPI （ ZAPI ）呼叫。

> 對於 9.9.1 之前的 ONTAP 版本、您可以使用 CLI Passthrough 功能、自動管理 SACL 和 DACL 權限。請參閱 "移轉考量" 和 "將私有CLI密碼與ONTAP REST API搭配使用" 以取得更多資訊。

我們提供幾個工作流程範例、說明如何使用 REST API 來管理 ONTAP 檔案安全服務。在使用工作流程並發出任何 REST API 呼叫之前、請務必先檢閱 "準備好使用工作流程"。

如果您使用 Python 、也會看到指令碼 "file_security_permissions.py" 如需如何自動化部分檔案安全活動的範例。

不只是使用靜態**API**、更是使用非靜態**CLI**命令**ONTAP ONTAP**

在許多工作中、使用 ONTAP REST API 所需的通話數、比等效的 ONTAP CLI 命令或 ONTAPI （ ZAPI ）通話數少。下表列出 API 呼叫清單、以及每項工作所需的 CLI 命令。

靜態**API ONTAP**	**CLI ONTAP**
「Get /傳輸 協定/file-Security /有效權限/」	「vserver安全性檔案目錄show-aive-permissions」
「POST /傳輸協定/檔案安全性/權限/」	1. 「vserver安全性檔案目錄NTFS建立」 2. 「vserver安全檔案目錄NTFS DACL add」 3. 「vserver安全檔案目錄NTFS SACL add」 4. 「vserver安全性檔案目錄原則建立」 5. 「vserver安全性檔案目錄原則工作新增」 6. 「適用Vserver安全檔案目錄」
「修補程式/傳輸協定/檔案安全性/權限/」	「vserver安全性檔案目錄NTFS修改」
「刪除/傳輸協定/檔案安全性/權限/」	1. 「Vserver安全檔案目錄NTFS DACL移除」 2. 「Vserver安全檔案目錄NTFS SACL移除」

相關資訊

- "說明檔案權限的 Python 指令碼"
- "利用REST API簡化檔案安全權限的管理ONTAP"
- "將私有CLI密碼與ONTAP REST API搭配使用"

取得檔案的有效權限

您可以擷取特定檔案或資料夾的目前有效權限。

HTTP 方法和端點

此 REST API 呼叫使用下列方法和端點。

HTTP方法	路徑
取得	/api/protocols / file-security/active-permissions/ ｛ SVM.uuid ｝ / ｛ path ｝

處理類型

同步

捲曲範例的其他輸入參數

除了所有 REST API 呼叫通用的參數之外、本步驟的捲髮範例中也會使用下列參數。

參數	類型	必要	說明
$SVM_ID	路徑	是的	這是包含檔案的 SVM 的 UUID 。
$FILE_PATH	路徑	是的	這是檔案或資料夾的路徑。

Curl範例

```
curl --request GET \
--location "https://$FQDN_IP/api/protocols/file-security/effective-
permissions/$SVM_ID/$FILE_PATH" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

Json輸出範例


```
{
  "svm": {
    "uuid": "cf5f271a-1beb-11ea-8fad-005056bb645e",
    "name": "vs1"
  },
  "user": "administrator",
  "type": "windows",
  "path": "/",
  "share": {
    "path": "/"
  },
  "file_permission": [
    "read",
    "write",
    "append",
    "read_ea",
    "write_ea",
    "execute",
    "delete_child",
    "read_attributes",
    "write_attributes",
    "delete",
    "read_control",
    "write_dac",
    "write_owner",
    "synchronize",
    "system_security"
  ],
  "share_permission": [
    "read",
    "read_ea",
    "execute",
    "read_attributes",
    "read_control",
    "synchronize"
  ]
}
```


取得檔案的稽核資訊

您可以擷取特定檔案或資料夾的稽核資訊。

HTTP 方法和端點

此 REST API 呼叫使用下列方法和端點。

HTTP方法	路徑
取得	/api/protocols / file-security/permissions/ ｛ SVM.uuid ｝ / ｛ path ｝

處理類型

同步

捲曲範例的其他輸入參數

除了所有 REST API 呼叫通用的參數之外、本步驟的捲髮範例中也會使用下列參數。

參數	類型	必要	說明
$SVM_ID	路徑	是的	這是包含檔案的 SVM 的 UUID 。
$FILE_PATH	路徑	是的	這是檔案或資料夾的路徑。

Curl範例

```
curl --request GET \
--location "https://$FQDN_IP/api/protocols/file-
security/permissions/$SVM_ID/$FILE_PATH" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

Json輸出範例

```
{
  "svm": {
    "uuid": "9479099d-5b9f-11eb-9c4e-0050568e8682",
    "name": "vs1"
  },
  "path": "/parent",
  "owner": "BUILTIN\\Administrators",
  "group": "BUILTIN\\Administrators",
  "control_flags": "0x8014",
  "acls": [
    {
      "user": "BUILTIN\\Administrators",
      "access": "access_allow",
      "apply_to": {
        "files": true,
        "sub_folders": true,
        "this_folder": true
      },
      "advanced_rights": {
```

```
        "append_data": true,
        "delete": true,
        "delete_child": true,
        "execute_file": true,
        "full_control": true,
        "read_attr": true,
        "read_data": true,
        "read_ea": true,
        "read_perm": true,
        "write_attr": true,
        "write_data": true,
        "write_ea": true,
        "write_owner": true,
        "synchronize": true,
        "write_perm": true
      },
      "access_control": "file_directory"
    },
    {
      "user": "BUILTIN\\Users",
      "access": "access_allow",
      "apply_to": {
        "files": true,
        "sub_folders": true,
        "this_folder": true
      },
      "advanced_rights": {
        "append_data": true,
        "delete": true,
        "delete_child": true,
        "execute_file": true,
        "full_control": true,
        "read_attr": true,
        "read_data": true,
        "read_ea": true,
        "read_perm": true,
        "write_attr": true,
        "write_data": true,
        "write_ea": true,
        "write_owner": true,
        "synchronize": true,
        "write_perm": true
      },
      "access_control": "file_directory"
    }
  ],
```

```
  "inode": 64,
  "security_style": "mixed",
  "effective_style": "ntfs",
  "dos_attributes": "10",
  "text_dos_attr": "----D---",
  "user_id": "0",
  "group_id": "0",
  "mode_bits": 777,
  "text_mode_bits": "rwxrwxrwx"
}
```

將新權限套用至檔案

您可以將新的安全性描述元套用至特定檔案或資料夾。

步驟 1 ：套用新權限

HTTP 方法和端點

此 REST API 呼叫使用下列方法和端點。

HTTP方法	路徑
貼文	/api/protocols / file-security/permissions/ ｛ SVM.uuid ｝ / ｛ path ｝

處理類型

非同步

捲曲範例的其他輸入參數

除了所有 REST API 呼叫通用的參數之外、本步驟的捲髮範例中也會使用下列參數。

參數	類型	必要	說明
$SVM_ID	路徑	是的	這是包含檔案的 SVM 的 UUID 。
$FILE_PATH	路徑	是的	這是檔案或資料夾的路徑。

Curl範例

```
curl --request POST --location "https://$FQDN_IP/api/protocols/file-
security/permissions/$SVM_ID/$FILE_PATH?return_timeout=0" --include
--header "Accept */*" --header "Authorization: Basic $BASIC_AUTH" --data
'{ \"acls\": [ { \"access\": \"access_allow\", \"advanced_rights\": {
\"append_data\": true, \"delete\": true, \"delete_child\": true,
\"execute_file\": true, \"full_control\": true, \"read_attr\": true,
\"read_data\": true, \"read_ea\": true, \"read_perm\": true,
\"write_attr\": true, \"write_data\": true, \"write_ea\": true,
\"write_owner\": true, \"write_perm\": true }, \"apply_to\": { \"files\":
true, \"sub_folders\": true, \"this_folder\": true }, \"user\":
\"administrator\" } ], \"control_flags\": \"32788\", \"group\": \"S-1-5-
21-2233347455-2266964949-1780268902-69700\", \"ignore_paths\": [
\"/parent/child2\" ], \"owner\": \"S-1-5-21-2233347455-2266964949-
1780268902-69304\", \"propagation_mode\": \"propagate\"}'
```

Json輸出範例

```
{
  "job": {
    "uuid": "3015c294-5bbc-11eb-9c4e-0050568e8682",
    "_links": {
      "self": {
        "href": "/api/cluster/jobs/3015c294-5bbc-11eb-9c4e-0050568e8682"
      }
    }
  }
}
```

步驟 **2** ：擷取工作狀態

執行工作流程 "取得工作執行個體" 並確認 `state` 價值是 `success`。

更新安全性描述元資訊

您可以將特定安全性描述元更新至特定檔案或資料夾、包括主要擁有者、群組或控制標誌。

步驟 **1** ：更新安全性描述元

HTTP 方法和端點

此 REST API 呼叫使用下列方法和端點。

HTTP方法	路徑
修補程式	/api/protocols / file-security/permissions/ ｛ SVM.uuid ｝ / ｛ path ｝

處理類型

非同步

捲曲範例的其他輸入參數

除了所有 REST API 呼叫通用的參數之外、本步驟的捲髮範例中也會使用下列參數。

參數	類型	必要	說明
$SVM_ID	路徑	是的	這是包含檔案的 SVM 的 UUID 。
$FILE_PATH	路徑	是的	這是檔案或資料夾的路徑。

Curl範例

```
curl --request POST --location "https://$FQDN_IP/api/protocols/file-
security/permissions/$SVM_ID/$FILE_PATH?return_timeout=0" --include
--header "Accept */*" --header "Authorization: Basic $BASIC_AUTH" --data
'{ \"control_flags\": \"32788\", \"group\": \"everyone\", \"owner\":
\"user1\"}'
```

Json輸出範例

```
{
  "job": {
    "uuid": "6f89e612-5bbd-11eb-9c4e-0050568e8682",
    "_links": {
      "self": {
        "href": "/api/cluster/jobs/6f89e612-5bbd-11eb-9c4e-0050568e8682"
      }
    }
  }
}
```

步驟 **2** ：擷取工作狀態

執行工作流程 "取得工作執行個體" 並確認 `state` 價值是 `success`。

刪除存取控制項目

您可以從特定檔案或資料夾刪除現有的存取控制項目（ ACE ）。變更會傳播到任何子物件。

步驟 **1** ：刪除 **ACE**

HTTP 方法和端點

此 REST API 呼叫使用下列方法和端點。

HTTP方法	路徑
刪除	/api/protocols / file-security/permissions/ ｛ SVM.uuid ｝ / ｛ path ｝

處理類型

非同步

捲曲範例的其他輸入參數

除了所有 REST API 呼叫通用的參數之外、本步驟的捲髮範例中也會使用下列參數。

參數	類型	必要	說明
$SVM_ID	路徑	是的	這是包含檔案的 SVM 的 UUID 。
$FILE_PATH	路徑	是的	這是檔案或資料夾的路徑。

Curl範例

```
curl --request DELETE --location "https://$FQDN_IP/api/protocols/file-
security/permissions/$SVM_ID/$FILE_PATH?return_timeout=0" --include
--header "Accept */*" --header "Authorization: Basic $BASIC_AUTH" --data
'{ \"access\": \"access_allow\", \"apply_to\": { \"files\": true,
\"sub_folders\": true, \"this_folder\": true }, \"ignore_paths\": [
\"/parent/child2\" ], \"propagation_mode\": \"propagate\"}'
```

Json輸出範例

```
{
  "job": {
    "uuid": "3015c294-5bbc-11eb-9c4e-0050568e8682",
    "_links": {
      "self": {
        "href": "/api/cluster/jobs/3015c294-5bbc-11eb-9c4e-0050568e8682"
      }
    }
  }
}
```

步驟 **2** ：擷取工作狀態

執行工作流程 "取得工作執行個體" 並確認 `state` 價值是 `success`。