TR-4921：勒索软件防御

How to enable StorageGRID in your environment

NetApp

July 05, 2024

目录

TR-4921：勒索软件防御

保护StorageGRID S3对象免遭勒索软件的攻击

了解勒索软件攻击以及如何利用StorageGRID安全最佳实践保护数据。

勒索软件攻击呈上升趋势。本文档就如何保护StorageGRID上的对象数据提供了一些建议。

如今、勒索软件已成为数据中心面临的一个始终存在的威胁。勒索软件旨在对数据进行加密、使依赖该数据的用户和应用程序无法使用该数据。保护从强化网络和可靠用户安全实践的常规防御开始、我们需要遵循数据访问安全实践。

勒索软件是当今最大的安全威胁之一。NetApp StorageGRID团队正在与我们的客户合作、以防范这些威胁。通过使用对象锁定和版本控制、您可以防止不必要的更改并从恶意攻击中恢复。数据安全是一项多层风险、您的对象存储只是数据中心的一部分。

StorageGRID最佳实践

对于StorageGRID、安全最佳实践应包括使用HTTPS和签名证书进行管理和对象访问。为应用程序和个人创建专用用户帐户、不要使用租户root帐户进行应用程序访问或用户数据访问。换言之、遵循最小特权原则。使用具有定义的身份和访问管理(IAM)策略的安全组来管理用户权限以及特定于应用程序和用户的访问帐户。实施这些措施后、您仍然必须确保数据受到保护。对于简单存储服务(S3)、在修改对象以对其进行加密时、可以通过覆盖原始对象来实现。

辩护方法

S3 API中的主要勒索软件保护机制是实施对象锁定。并非所有应用程序都与对象锁定兼容、因此本报告中介绍了另外两个保护对象的选项：复制到启用了版本控制的另一个存储分段以及使用IAM策略进行版本控制。

从何处查找追加信息

要了解有关本文档中所述信息的更多信息，请查看以下文档和 / 或网站：

- NetApp StorageGRID文档中心 https://docs.netapp.com/us-en/storagegrid-118/
- NetApp StorageGRID支持 https://docs.netapp.com/us-en/storagegrid-enable/
- StorageGRID文档资源页面 https://www.netapp.com/data-storage/storagegrid/documentation/
- NetApp产品文档 https://www.netapp.com/support-and-training/documentation/

使用对象锁定进行勒索软件防护

了解StorageGRID中的对象锁定如何提供WORM模型来防止数据删除或覆盖、以及如何满足法规要求。

对象锁定提供了WORM模型、可防止删除或覆盖对象。StorageGRID实施对象锁定 "评估的协资产" 有助于满足法规要求、支持合法保留、合规模式和对象保留监管模式以及默认分段保留策略。您必须在创建分段和版本控制过程中启用对象锁定。对象的特定版本被锁定、如果未定义版本ID、则保留将放置在对象的当前版本上。如果当前版本配置了保留、并且尝试删除、修改或覆盖对象、则会创建一个新版本、其中包含删除标记、或者对象的新

修订版作为当前版本。 锁定的版本将保留为非当前版本。对于尚不兼容的应用程序、您仍可以使用对象锁定以及存储分段上的默认保留配置。定义配置后、此操作会对放入存储分段的每个新对象应用对象保留。只要将应用程序配置为在保留时间过去之前不删除或覆盖对象、此操作就有效。

以下是使用对象锁定API的几个示例：

对象锁定合法保留是应用于对象的简单开/关状态。

```
aws s3api put-object-legal-hold --bucket mybucket --key myfile.txt --legal
-hold Status=ON --endpoint-url https://s3.company.com
```

设置合法保留状态成功后不会返回任何值、因此可以通过GET操作进行验证。

```
aws s3api get-object-legal-hold --bucket mybucket --key myfile.txt
--endpoint-url https://s3.company.com
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

要关闭合法保留、请应用关闭状态。

```
aws s3api put-object-legal-hold --bucket mybucket --key myfile.txt --legal
-hold Status=OFF --endpoint-url https://s3.company.com
aws s3api get-object-legal-hold --bucket mybucket --key myfile.txt
--endpoint-url https://s3.company.com
{
    "LegalHold": {
        "Status": "OFF"
    }
}
```

设置对象保留时、会使用保留到时间戳。

```
aws s3api put-object-retention --bucket mybucket --key myfile.txt
--retention '{"Mode":"COMPLIANCE", "RetainUntilDate": "2022-06-
10T16:00:00"}'  --endpoint-url https://s3.company.com
```

同样、成功时也不返回任何值、因此您可以通过GET调用来验证保留状态。

```
aws s3api get-object-retention --bucket mybucket --key myfile.txt
--endpoint-url https://s3.company.com
{
    "Retention": {
        "Mode": "COMPLIANCE",
        "RetainUntilDate": "2022-06-10T16:00:00+00:00"
    }
```

为启用了对象锁定的存储分段设置默认保留期限时、保留期限以天和年为单位。

```
aws s3api put-object-lock-configuration --bucket mybucket --object-lock
-configuration '{ "ObjectLockEnabled": "Enabled", "Rule": {
"DefaultRetention": { "Mode": "COMPLIANCE", "Days": 1 }}}' --endpoint-url
https://s3.company.com
```

与大多数操作一样、成功后不会返回任何响应、因此、我们可以执行GET以验证配置。

```
aws s3api get-object-lock-configuration --bucket mybucket --endpoint-url
https://s3.company.com
{
    "ObjectLockConfiguration": {
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 1
            }
        }
    }
}
```

接下来、您可以在应用保留配置的情况下将对象放入存储分段中。

```
aws s3 cp myfile.txt s3://mybucket --endpoint-url https://s3.company.com
```

Put操作确实会返回响应。

```
upload: ./myfile.txt to s3://mybucket/myfile.txt
```

在保留对象上、上例中为分段设置的保留持续时间将转换为对象上的保留时间戳。

```
aws s3api get-object-retention --bucket mybucket --key myfile.txt
--endpoint-url https://s3.company.com
{
    "Retention": {
        "Mode": "COMPLIANCE",
        "RetainUntilDate": "2022-03-02T15:22:47.202000+00:00"
    }
}
```

使用具有版本控制的复制存储分段进行勒索软件防护

了解如何使用StorageGRID CloudMirror将对象复制到二级存储分段。

并非所有应用程序和工作负载都能与对象锁定兼容。另一种方法是、将对象复制到同一网格中的二级存储分段(最好是访问受限的不同租户)、或者使用StorageGRID平台服务CloudMirror的任何其他S3端点。

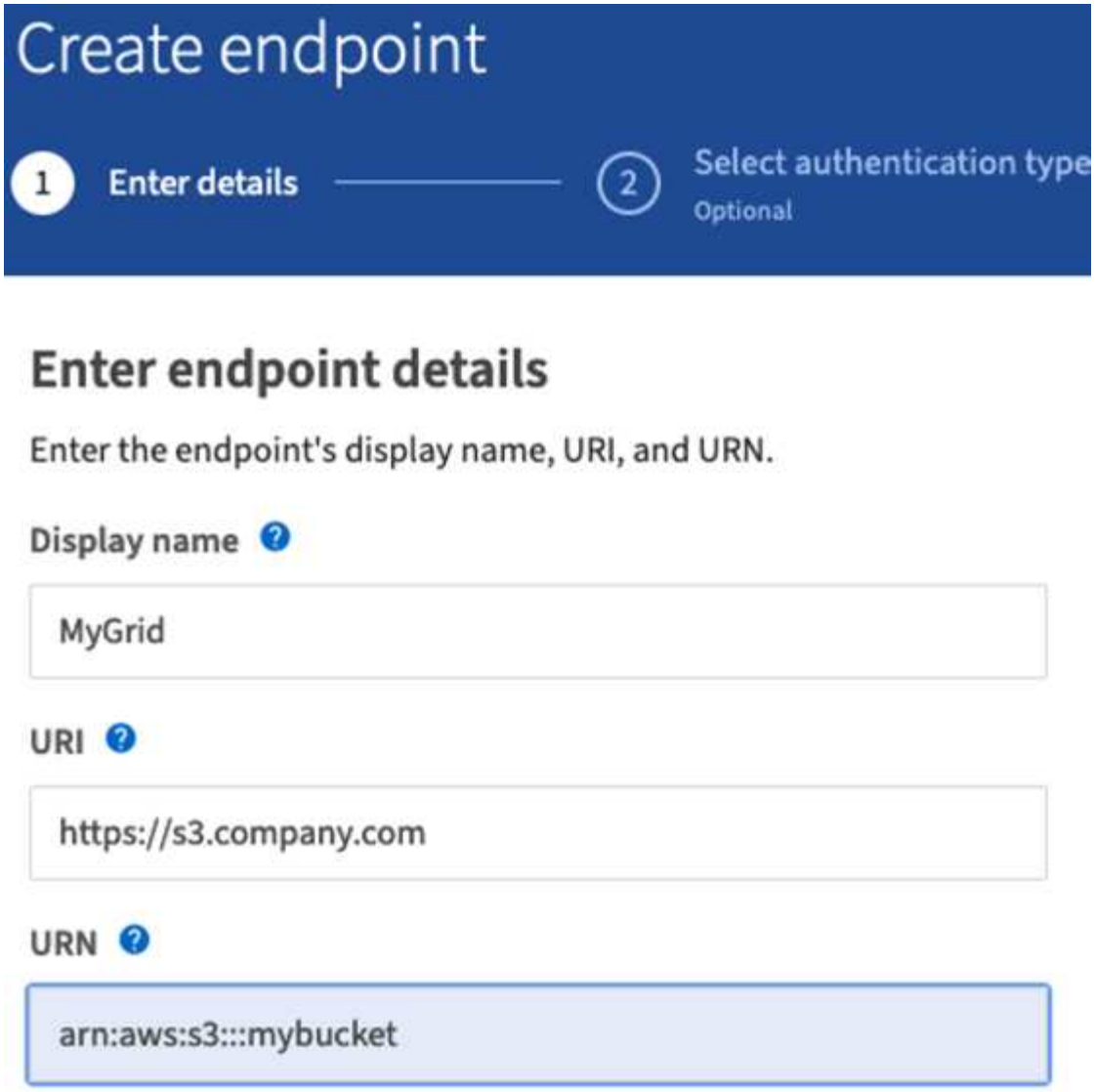
StorageGRID CloudMirror是StorageGRID的一个组件、可以配置为在将某个存储分段的对象移入源存储分段时将其复制到定义的目标、而不会复制删除。由于CloudMirror是StorageGRID的一个集成组件、因此不能关闭它、也不能被基于S3 API的攻击所操纵。您可以在启用版本控制的情况下配置此复制分段。在这种情况下、您需要对复制的存储分段的旧版本进行一些可安全丢弃的自动清理。为此、您可以使用StorageGRID ILM策略引擎。创建规则、根据非当前时间管理对象放置、持续数天、足以识别攻击并从攻击中恢复。

这种方法的一个缺点是、它会通过保留存储分段的完整第二个副本以及多个版本的对象一段时间来消耗更多存储。此外、必须从复制的存储分段中手动删除主存储分段中特意删除的对象。产品之外还有其他复制选项、例如NetApp CloudSync、可以为类似的解决方案复制删除。二级存储分段启用了版本控制而未启用对象锁定的另一个缺点是、存在许多特权帐户、这些帐户可能会导致二级位置损坏。其优势在于、它应该是该端点或租户存储分段的唯一帐户、而这种损害可能不包括对主位置上的帐户的访问、反之亦然。

创建源分段和目标分段并为目标配置版本控制后、您可以按如下所示配置和启用复制：

步骤

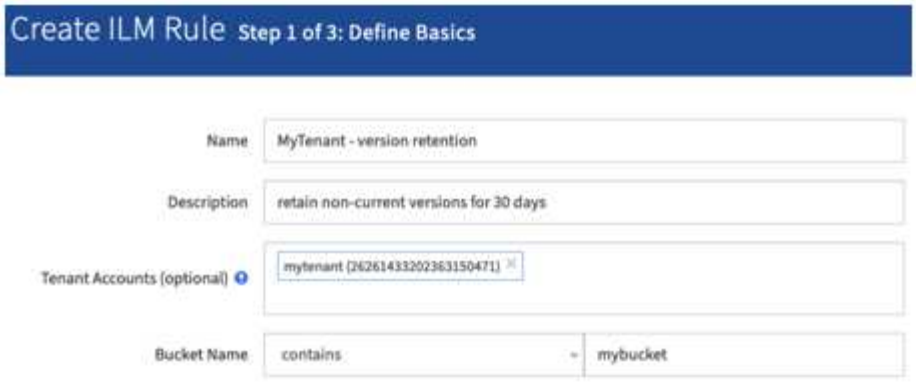
1. 要配置CloudMirror、请为S3目标创建一个平台服务端点。

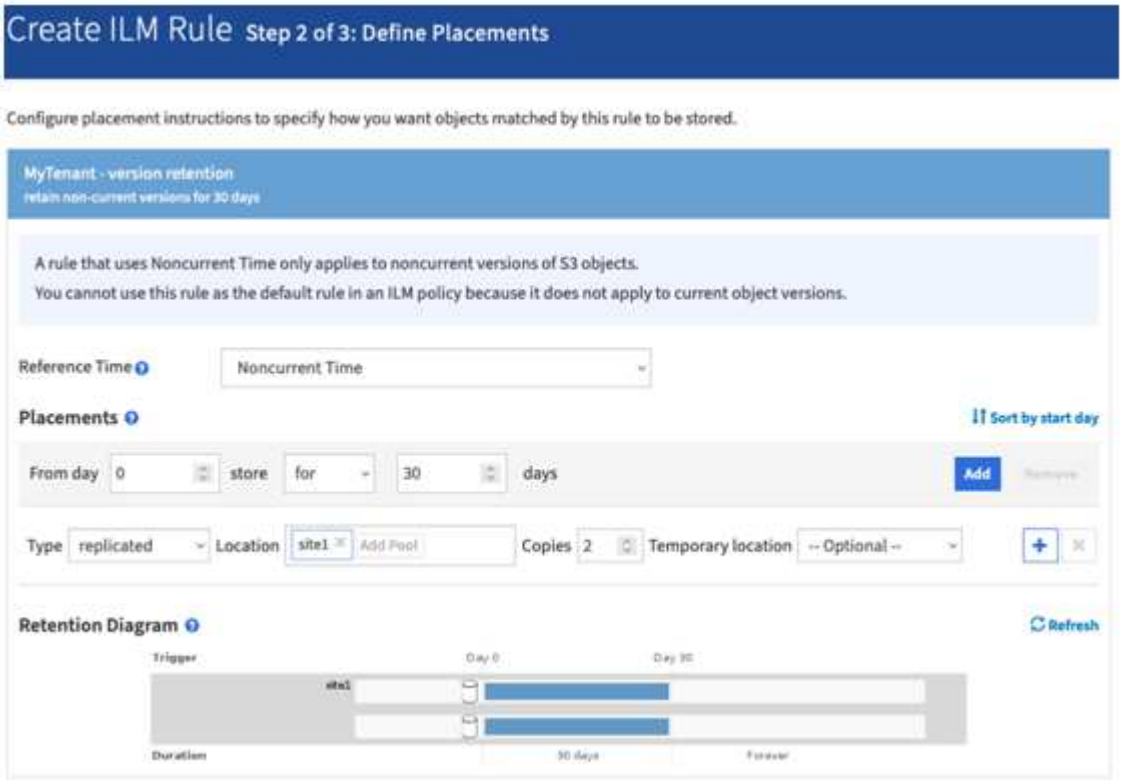


2. 在源存储分段上、配置复制以使用配置的端点。

```
<ReplicationConfiguration>
    <Role></Role>
    <Rule>
        <Status>Enabled</Status>
        <Prefix></Prefix>
        <Destination>
            <Bucket>arn:aws:s3:::mybucket</Bucket>
            <StorageClass>STANDARD</StorageClass>
        </Destination>
    </Rule>
</ReplicationConfiguration>
```

3. 创建ILM规则以管理存储放置和版本存储持续时间管理。在此示例中、配置了要存储的对象的非最新版本。





站点1中有两个副本、保留30天。此外、您还可以根据在ILM规则中使用加载时间作为参考时间来为当前版本的对象配置规则、以匹配源存储分段存储持续时间。可以对对象版本的存储放置进行卷或复制。

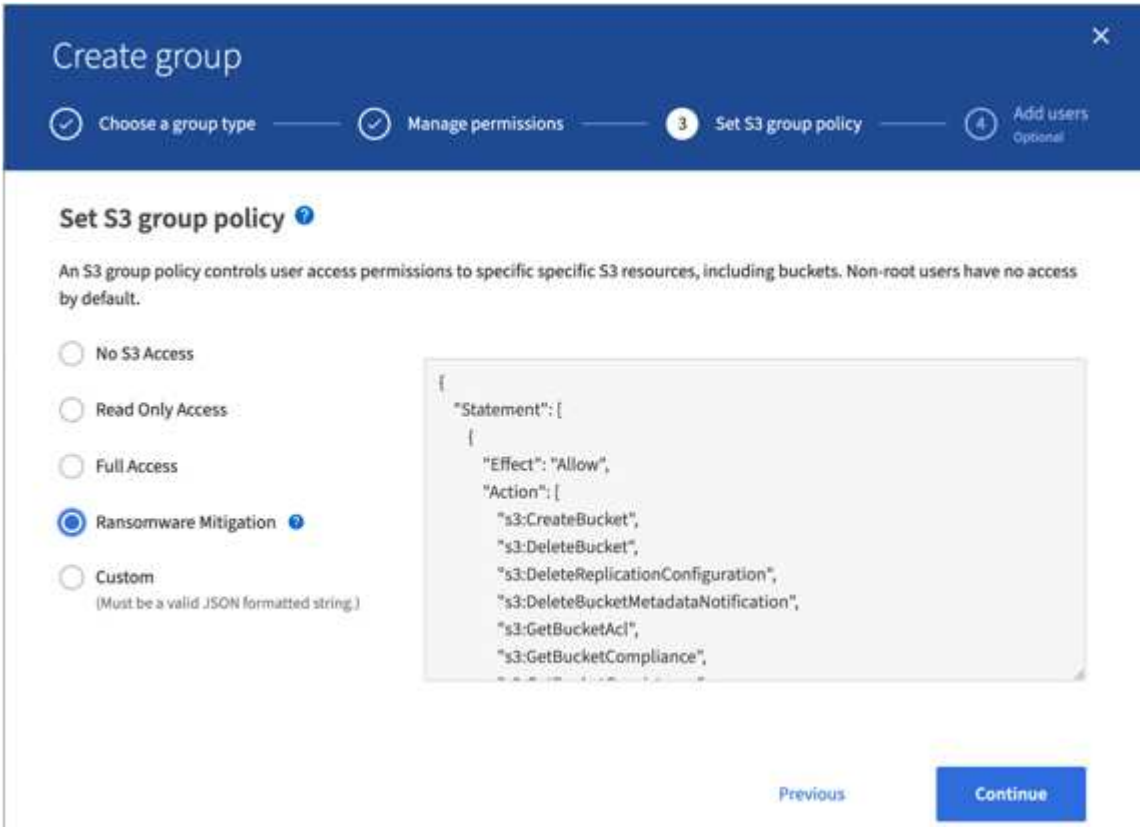
使用版本控制和保护性**IAM**策略进行勒索软件防御

了解如何通过在StorageGRID中对存储分段启用版本控制并对用户安全组实施IAM策略来保护您的数据。

在不使用对象锁定或复制的情况下保护数据的一种方法是、在存储分段上启用版本控制、并在用户安全组上实施IAM策略、以限制用户管理对象版本的能力。在发生攻击时、系统会创建新的错误数据版本作为当前版本、而最新的非最新版本是安全清理数据。为获得数据访问权限而泄露的帐户无权删除或以其他方式更改用于保护数据以供日后还原操作的非最新版本。与上一种情形一样、ILM规则可在您选择的持续时间内管理非最新版本的保留。缺点是、仍然可能存在针对恶意攻击者攻击的特权帐户、但必须为所有应用程序服务帐户和用户配置限制性更强的访问。限制性组策略必须明确允许您希望用户或应用程序能够执行的每个操作、并明确拒绝您不希望用户

或应用程序能够执行的任何操作。NetApp建议不要使用通配符allow、因为将来可能会引入新的操作、您需要控制是允许还是拒绝该操作。对于此解决方案、拒绝列表必须包括DeleteObjectVersion、PutBucketPolicy、DeleteBucketPolicy、PutLifecycleConfiguration和PutketVersioning、以防止用户或编程更改分段和对象版本的版本控制配置。

在StorageGRID 11.7中、引入了一个新的S3组策略选项"Ransmans要 缓解"、以便于实施此解决方案。在租户中创建用户组时、在选择组权限后、您可以看到此新的可选策略。



下面是组策略的内容、其中包括显式允许的大多数可用操作以及所需的最小拒绝值。

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:DeleteBucket",
                "s3:DeleteReplicationConfiguration",
"s3:DeleteBucketMetadataNotification",
                "s3:GetBucketAcl",
                "s3:GetBucketCompliance",
                "s3:GetBucketConsistency",
                "s3:GetBucketLastAccessTime",
                "s3:GetBucketLocation",
                "s3:GetBucketNotification"
"s3:GetBucketObjectLockConfiguration",
```

```
                "s3:GetBucketPolicy",
                "s3:GetBucketMetadataNotification",
                "s3:GetReplicationConfiguration",
                "s3:GetBucketCORS",
                "s3:GetBucketVersioning",
                "s3:GetBucketTagging",
                "s3:GetEncryptionConfiguration",
                "s3:GetLifecycleConfiguration",
                "s3:ListBucket",
                "s3:ListBucketVersions",
                "s3:ListAllMyBuckets",
                "s3:ListBucketMultipartUploads",
                "s3:PutBucketConsistency",
                "s3:PutBucketLastAccessTime",
                "s3:PutBucketNotification",
"s3:PutBucketObjectLockConfiguration",
                "s3:PutReplicationConfiguration",
                "s3:PutBucketCORS",
                "s3:PutBucketMetadataNotification",
                "s3:PutBucketTagging",
                "s3:PutEncryptionConfiguration",
                "s3:AbortMultipartUpload",
                "s3:DeleteObject",
                "s3:DeleteObjectTagging",
                "s3:DeleteObjectVersionTagging",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectLegalHold",
                "s3:GetObjectRetention",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion",
                "s3:GetObjectVersionAcl",
                "s3:GetObjectVersionTagging",
                "s3:ListMultipartUploadParts",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectLegalHold",
                "s3:PutObjectRetention",
                "s3:PutObjectTagging",
                "s3:PutObjectVersionTagging",
                "s3:RestoreObject",
                "s3:ValidateObject",
                "s3:PutBucketCompliance",
                "s3:PutObjectVersionAcl"
            ],
            "Resource": "arn:aws:s3:::*"
```

```
        },
        {
            "Effect": "Deny",
            "Action": [
                "s3:DeleteObjectVersion",
                "s3:DeleteBucketPolicy",
                "s3:PutBucketPolicy",
                "s3:PutLifecycleConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```