TR-4921: Ransomware-Verteidigung

How to enable StorageGRID in your environment

NetApp
October 09, 2024

Inhalt

TR-4921: Ransomware-Verteidigung

StorageGRID S3 Objekte vor Ransomware schützen

Informieren Sie sich über Ransomware-Angriffe und den Schutz von Daten mit StorageGRID Sicherheits-Best Practices.

Ransomware-Angriffe sind auf dem Vormarsch. Dieses Dokument enthält einige Empfehlungen zum Schutz Ihrer Objektdaten auf StorageGRID.

Ransomware heute ist die allgegenwärtige Gefahr im Datacenter. Ransomware wurde entwickelt, um Daten zu verschlüsseln und sie für Benutzer und Applikationen, die darauf angewiesen sind, nicht nutzbar zu machen. Der Schutz beginnt mit den üblichen Schutzmaßnahmen für gehärtete Netzwerke und solide Benutzersicherheitspraktiken, und wir müssen die Sicherheitsverfahren für den Datenzugriff befolgen.

Ransomware ist eine der größten Sicherheitsbedrohungen von heute. Das NetApp StorageGRID Team arbeitet mit unseren Kunden zusammen, um diesen Bedrohungen einen Schritt voraus zu sein. Mit der Verwendung von Objektsperre und Versionierung können Sie sich vor unerwünschten Änderungen schützen und sich vor böswilligen Angriffen erholen. Datensicherheit ist ein Multi-Layer-Unternehmen, dessen Objekt-Storage nur ein Teil in Ihrem Datacenter ist.

Best Practices von StorageGRID

Für StorageGRID sollten Sicherheits-Best-Practices die Verwendung von HTTPS mit signierten Zertifikaten für Management und Objektzugriff umfassen. Erstellen Sie dedizierte Benutzerkonten für Anwendungen und Personen und verwenden Sie die Mandanten-Root-Konten nicht für den Zugriff auf Anwendungen oder Benutzerdaten. Mit anderen Worten, folgen Sie dem Prinzip der geringsten Privilegien. Verwenden Sie Sicherheitsgruppen mit definierten IAM-Richtlinien (Identity and Access Management) zur Steuerung von Benutzerrechten und Zugriffskonten für spezifische Anwendungen und Benutzer. Mit diesen Maßnahmen müssen Sie dennoch sicherstellen, dass Ihre Daten geschützt sind. Bei Simple Storage Service (S3) wird bei Änderungen von Objekten zur Verschlüsselung das ursprüngliche Objekt überschrieben.

Verteidigungsmethoden

Der primäre Mechanismus zum Schutz vor Ransomware in der S3-API ist die Implementierung von Objektsperrung. Nicht alle Anwendungen sind mit Objektsperre kompatibel, daher gibt es zwei weitere Optionen zum Schutz Ihrer Objekte, die in diesem Bericht beschrieben werden: Replikation in einen anderen Bucket mit aktivierter Versionierung und Versionierung mit IAM-Richtlinien.

Wo Sie weitere Informationen finden

Sehen Sie sich die folgenden Dokumente und/oder Websites an, um mehr über die in diesem Dokument beschriebenen Informationen zu erfahren:

- NetApp StorageGRID Dokumentationszentrum https://docs.netapp.com/us-en/storagegrid-118/
- NetApp StorageGRID Enablement https://docs.netapp.com/us-en/storagegrid-enable/
- StorageGRID Dokumentations-Ressourcen https://www.netapp.com/data-storage/storagegrid/documentation/
- NetApp Produktdokumentation https://www.netapp.com/support-and-training/documentation/

Ransomware-Verteidigung mit Objektsperre

Entdecken Sie, wie die Objektsperre in StorageGRID ein WORM-Modell bietet, das das Löschen oder Überschreiben von Daten verhindert, und wie es die gesetzlichen Vorgaben erfüllt.

Die Objektsperrung verfügt über ein WORM-Modell, mit dem verhindert wird, dass Objekte gelöscht oder überschrieben werden. Die StorageGRID-Implementierung von Objektsperrung "Cohasset bewertet" unterstützt die Einhaltung gesetzlicher Vorgaben; sie unterstützt die gesetzliche Aufbewahrungspflichten, den Compliance-Modus und den Governance-Modus für die Objektaufbewahrung sowie die standardmäßigen Bucket-Aufbewahrungsrichtlinien. Sie müssen die Objektsperre als Teil der Bucket-Erstellung und -Versionierung aktivieren. Eine bestimmte Version eines Objekts ist gesperrt, und wenn keine Versions-ID definiert ist, wird die Aufbewahrung auf die aktuelle Version des Objekts platziert. Wenn für die aktuelle Version die Aufbewahrung konfiguriert ist und versucht wird, das Objekt zu löschen, zu ändern oder zu überschreiben, wird eine neue Version mit einer Löschmarkierung oder der neuen Revision des Objekts als aktuelle Version erstellt. und die gesperrte Version wird als nicht aktuelle Version beibehalten. Für Applikationen, die noch nicht kompatibel sind, können Sie möglicherweise noch Objektsperre und eine Standardkonfiguration für die Aufbewahrung auf dem Bucket nutzen. Nach der Definition der Konfiguration wird dabei eine Objektaufbewahrung auf jedes neue Objekt angewendet, das in den Bucket aufgenommen wird. Dies funktioniert, solange die Anwendung so konfiguriert ist, dass die Objekte nicht vor Ablauf der Aufbewahrungszeit gelöscht oder überschrieben werden.

Hier sind einige Beispiele für die Verwendung der Objektsperre API:

Objektsperre Legal Hold ist ein einfacher ein/aus-Status, der auf ein Objekt angewendet wird.

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

Wenn Sie den Legal Hold-Status festlegen, wird bei Erfolg kein Wert zurückgegeben, sodass er mit einer GET-Operation überprüft werden kann.

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

Um die Legal Hold-Taste zu deaktivieren, wenden Sie den AUS-Status an.

```
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"
    }
}
```

Die Objektaufbewahrung wird mit einem Zeitstempel bis beibehalten.

```
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
```

Auch hier gibt es keinen zurückgegebenen Wert auf Erfolg, so dass Sie den Status der Aufbewahrung ähnlich mit einem get Call überprüfen können.

```
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"
    }
```

Wenn für einen Bucket mit aktivierter Objektsperre eine Standardaufbewahrung eingerichtet wird, wird eine Aufbewahrungsfrist in Tagen und Jahren verwendet.

```
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
```

Wie bei den meisten dieser Operationen wird bei Erfolg keine Antwort zurückgegeben. Daher können wir ein GET durchführen, damit die Konfiguration überprüft werden kann.

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

Als Nächstes können Sie ein Objekt mit der angewandten Aufbewahrungskonfiguration in den Bucket legen.

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

Der PUT-Vorgang gibt eine Antwort zurück.

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

Für das Aufbewahrungs-Objekt wird die Aufbewahrungsdauer, die im vorhergehenden Beispiel auf dem Bucket festgelegt wurde, in einen Aufbewahrungszeitstempel für das Objekt konvertiert.

```
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"
    }
}
```

Ransomware-Verteidigung durch replizierten Bucket mit Versionierung

Erfahren Sie, wie Sie Objekte mit StorageGRID CloudMirror in einen sekundären Bucket replizieren.

Nicht alle Applikationen und Workloads werden mit Objektsperre kompatibel sein. Eine weitere Option besteht darin, die Objekte auf einen sekundären Bucket zu replizieren, entweder im selben Grid (vorzugsweise ein

anderer Mandant mit eingeschränktem Zugriff) oder auf einen anderen S3-Endpunkt mit dem StorageGRID-Plattformservice CloudMirror.

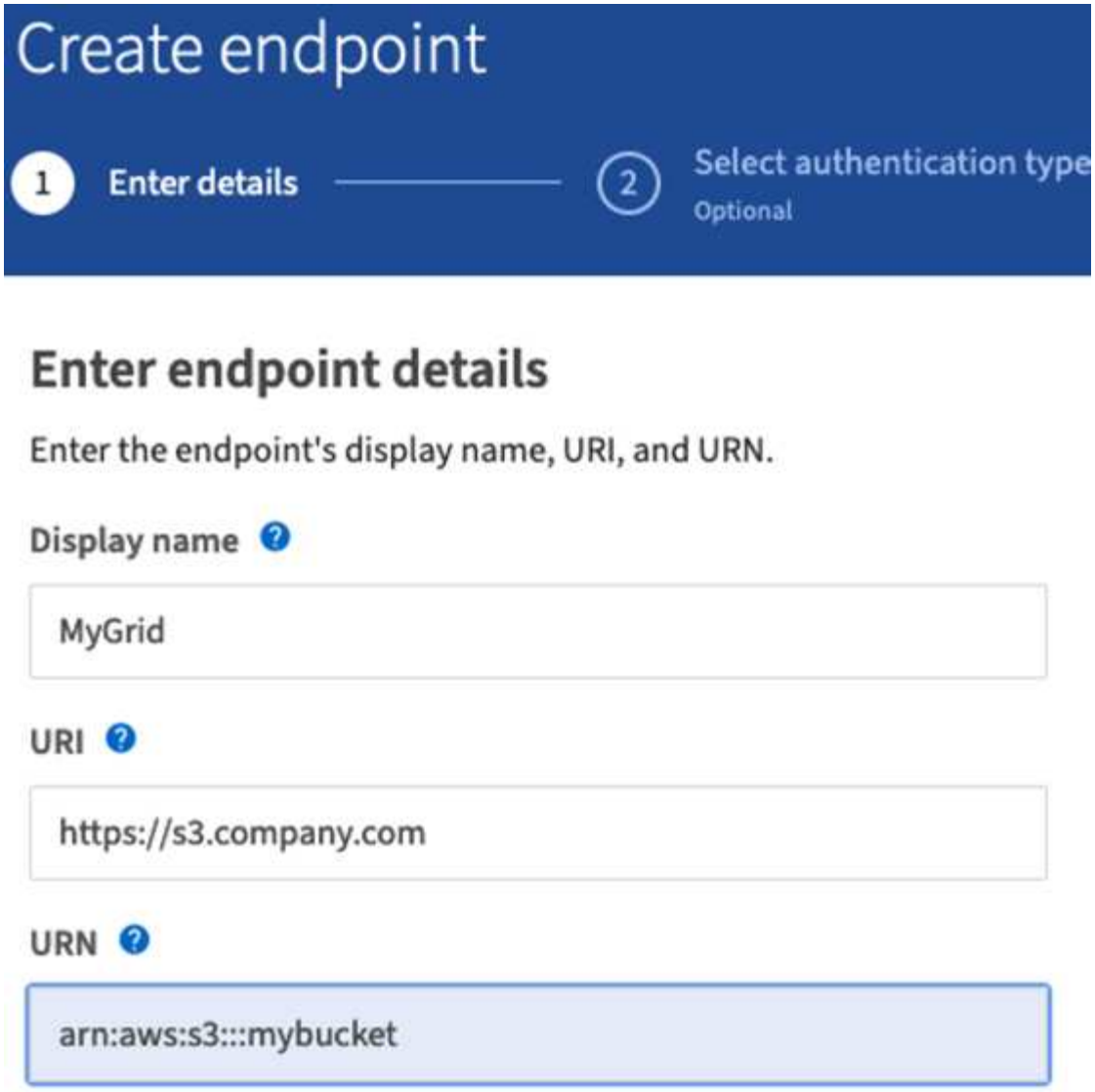
StorageGRID CloudMirror ist eine Komponente von StorageGRID, die so konfiguriert werden kann, dass Objekte eines Buckets auf ein definiertes Ziel repliziert werden, da sie in den Quell-Bucket aufgenommen werden und Löschungen nicht repliziert werden. Da CloudMirror eine integrierte Komponente von StorageGRID ist, kann sie nicht durch einen S3-API-basierten Angriff deaktiviert oder manipuliert werden. Sie können diesen replizierten Bucket mit aktivierter Versionierung konfigurieren. In diesem Szenario benötigen Sie eine automatisierte Bereinigung der alten Versionen des replizierten Buckets, die sicher zu verwerfen sind. Dazu können Sie die StorageGRID ILM-Richtlinien-Engine verwenden. Erstellen Sie Regeln, um die Objektplatzierung auf Basis der nicht aktuellen Zeit für mehrere Tage zu verwalten, die ausreichend sind, um einen Angriff identifiziert und wiederhergestellt zu haben.

Ein Nachteil dieses Ansatzes besteht darin, dass mehr Storage verbraucht wird. Dazu ist eine vollständige zweite Kopie des Buckets und mehrere Versionen der Objekte verfügbar, die einige Zeit aufbewahrt werden. Darüber hinaus müssen die Objekte, die absichtlich aus dem primären Bucket gelöscht wurden, manuell aus dem replizierten Bucket entfernt werden. Außerhalb des Produkts gibt es weitere Replizierungsoptionen, wie z. B. NetApp CloudSync, mit denen Löschvorgänge für eine ähnliche Lösung repliziert werden können. Ein weiterer Nachteil für den sekundären Bucket, bei dem die Versionierung aktiviert und nicht die Objektsperre aktiviert ist, besteht darin, dass eine Reihe privilegierter Konten vorhanden ist, die verwendet werden könnten, um Schäden am sekundären Standort zu verursachen. Der Vorteil ist, dass es sich um ein eindeutiges Konto für diesen Endpunkt oder Mandanten-Bucket handeln sollte, und der Kompromiss wahrscheinlich keinen Zugriff auf Konten am primären Standort oder umgekehrt umfasst.

Nachdem die Quell- und Ziel-Buckets erstellt und das Ziel mit Versionierung konfiguriert wurde, können Sie die Replikation wie folgt konfigurieren und aktivieren:

Schritte

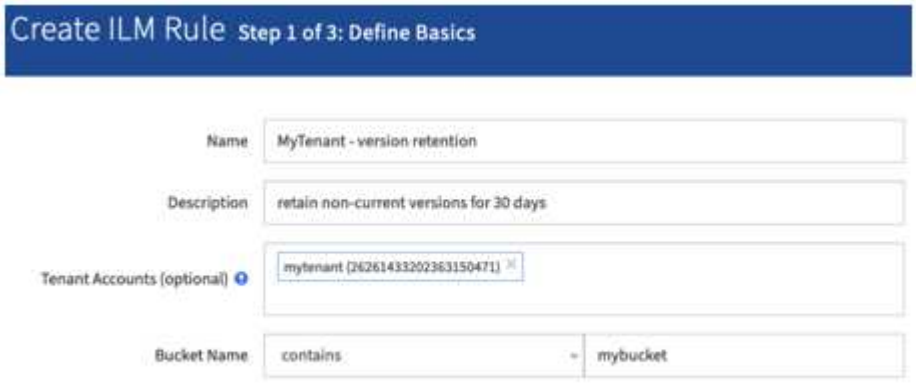
1. Erstellen Sie zum Konfigurieren von CloudMirror einen Plattformservices-Endpunkt für das S3-Ziel.

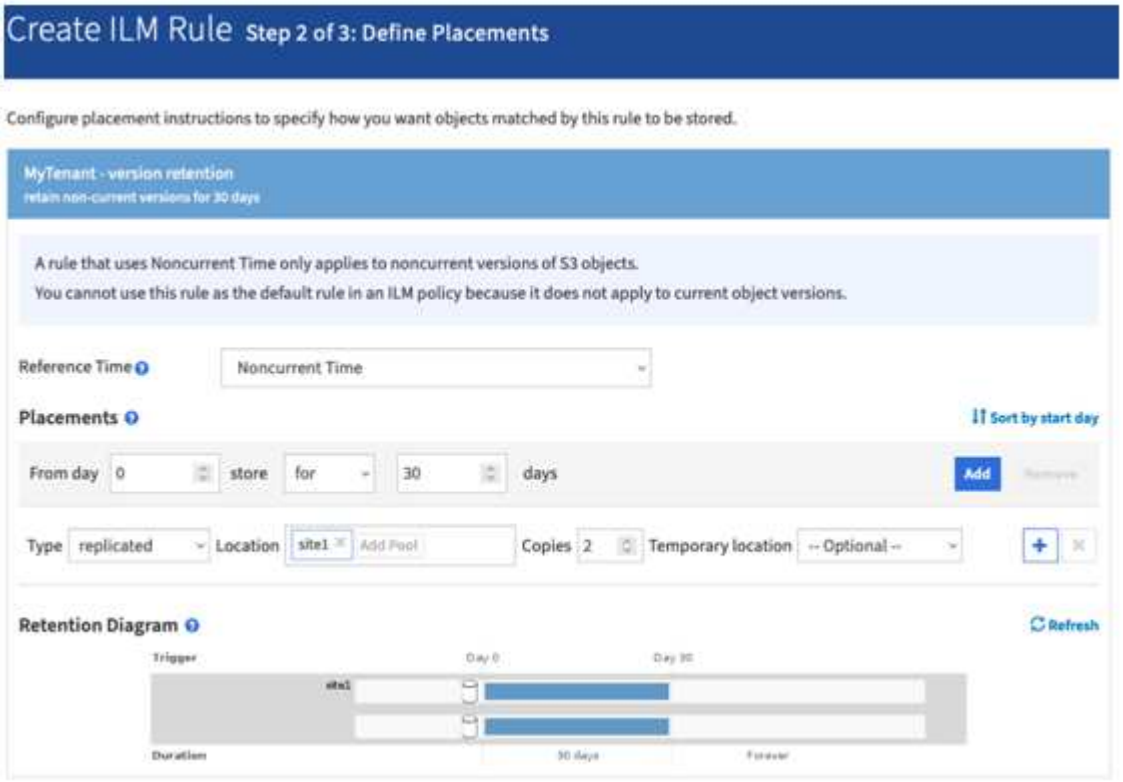


2. Konfigurieren Sie auf dem Quell-Bucket die Replikation zur Verwendung des konfigurierten Endpunkts.

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

3. Erstellen Sie ILM-Regeln, um die Storage-Platzierung und das Versionsspeicherzeitmanagement zu managen. In diesem Beispiel werden die nicht aktuellen Versionen der zu speichernden Objekte konfiguriert.





An Standort 1 sind 30 Tage lang zwei Kopien vorhanden. Sie konfigurieren außerdem die Regeln für die aktuelle Version der Objekte basierend auf der Verwendung der Aufnahmezeit als Referenzzeit in der ILM-Regel, um der Speicherdauer des Quell-Buckets anzupassen. Die Storage-Platzierung für die Objektversionen kann Erasure Coded oder repliziert werden.

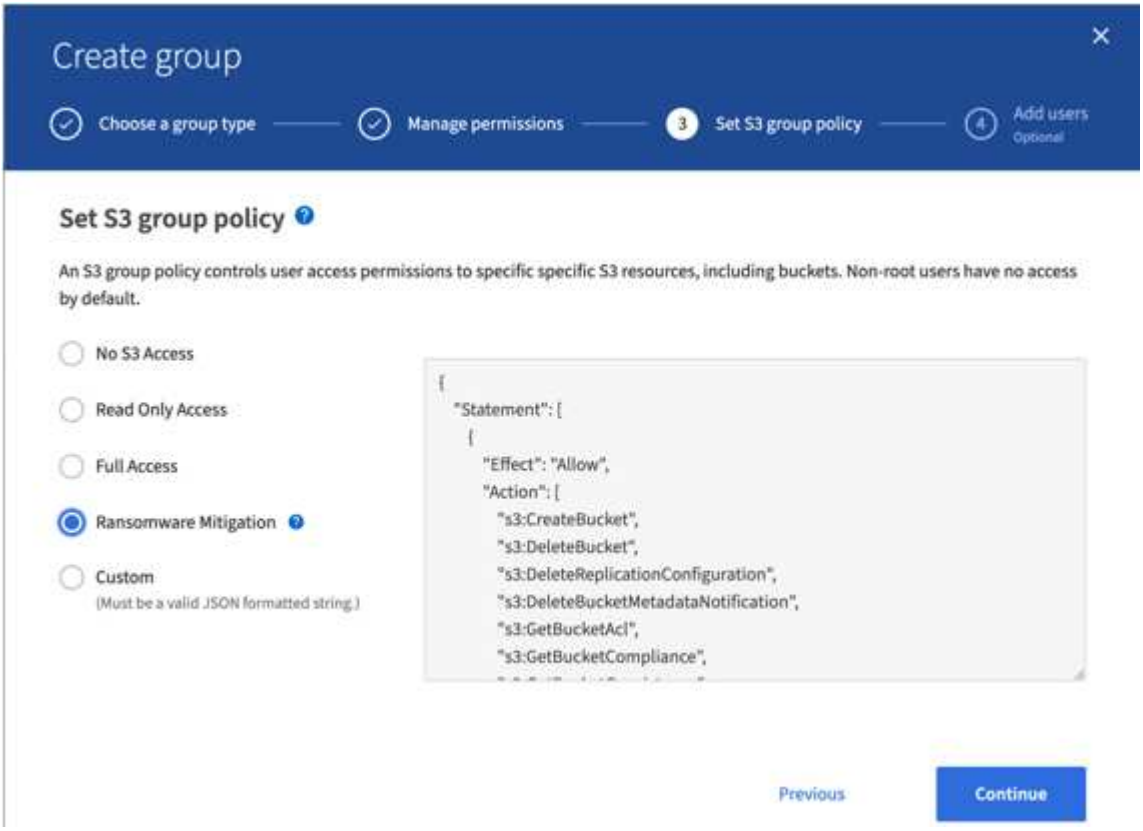
Ransomware-Verteidigung durch Versionierung mit Schutz-IAM-Richtlinie

Erfahren Sie, wie Sie Ihre Daten schützen, indem Sie die Versionierung auf dem Bucket aktivieren und IAM-Richtlinien auf Benutzersicherheitsgruppen in StorageGRID implementieren.

Eine Methode zum Schutz Ihrer Daten ohne Objektsperre oder Replikation besteht darin, die Versionierung auf dem Bucket zu aktivieren und IAM-Richtlinien auf den Benutzersicherheitsgruppen zu implementieren, um die

Fähigkeit des Benutzers zu beschränken, Versionen der Objekte zu verwalten. Im Falle eines Angriffs werden neue fehlerhafte Versionen der Daten als aktuelle Version erstellt, und die neueste nicht-aktuelle Version ist die sichere saubere Daten. Die Konten, die für den Zugriff auf die Daten kompromittiert wurden, haben keinen Zugriff auf das Löschen oder anderweitige Ändern der nicht-aktuellen Version, um sie für spätere Wiederherstellungsvorgänge zu schützen. Wie im vorherigen Szenario verwalten ILM-Regeln die Aufbewahrung der nicht aktuellen Versionen mit einer Dauer Ihrer Wahl. Der Nachteil ist, dass es immer noch die Möglichkeit von privilegierten Konten für einen schlechten Akteurangriff gibt, aber alle Anwendungsdienstkonten und Benutzer müssen mit einem restriktiveren Zugriff konfiguriert werden. Die restriktive Gruppenrichtlinie muss jede Aktion, die Benutzer oder Anwendungen ausführen sollen, ausdrücklich zulassen und alle Aktionen, die nicht ausgeführt werden sollen, ausdrücklich ablehnen. NetApp empfiehlt keine Platzhalterfunktion, da in Zukunft möglicherweise eine neue Aktion eingeführt wird. Sie sollten dann kontrollieren, ob sie erlaubt oder verweigert wird. Für diese Lösung muss die Deny-Liste DeleteObjectVersion, PutBucketPolicy, DeleteBucketPolicy, PutLifecycleConfiguration und PutBucketVersioning enthalten, um die Versionskonfiguration des Buckets und Objektversionen vor Benutzer- oder programmatischen Änderungen zu schützen.

In StorageGRID 11.7 wurde eine neue S3-Gruppenrichtlinienoption zur Risikominimierung eingeführt, um die Implementierung dieser Lösung zu vereinfachen. Beim Erstellen einer Benutzergruppe im Mandanten wird nach Auswahl der Gruppenberechtigungen diese neue optionale Richtlinie angezeigt.



Im Folgenden finden Sie den Inhalt der Gruppenrichtlinie, die die meisten verfügbaren Vorgänge explizit erlaubt und das erforderliche Minimum abgelehnt enthält.

```
{
    "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:::*"
        }
    ]
}
```