

The background features several thick, light blue curved lines that sweep across the frame, creating a modern, abstract design. One large arc starts from the top left and curves towards the bottom right. Another arc is positioned in the top left corner, and a third, smaller one is in the bottom right corner.

Securing S3 Backups Against Ransomware

Introductions



Tom Manville
Director of Engineering



Michael Cade
Senior Technologist



The Need for immutable backups



Accidental
Deletion

People
make
mistakes



Policy Gaps

Confusion,
misconfiguration



Internal
Security
Threats

Malicious
Insiders



External
Security
Threats

Ransomware



Legal
&
Compliance

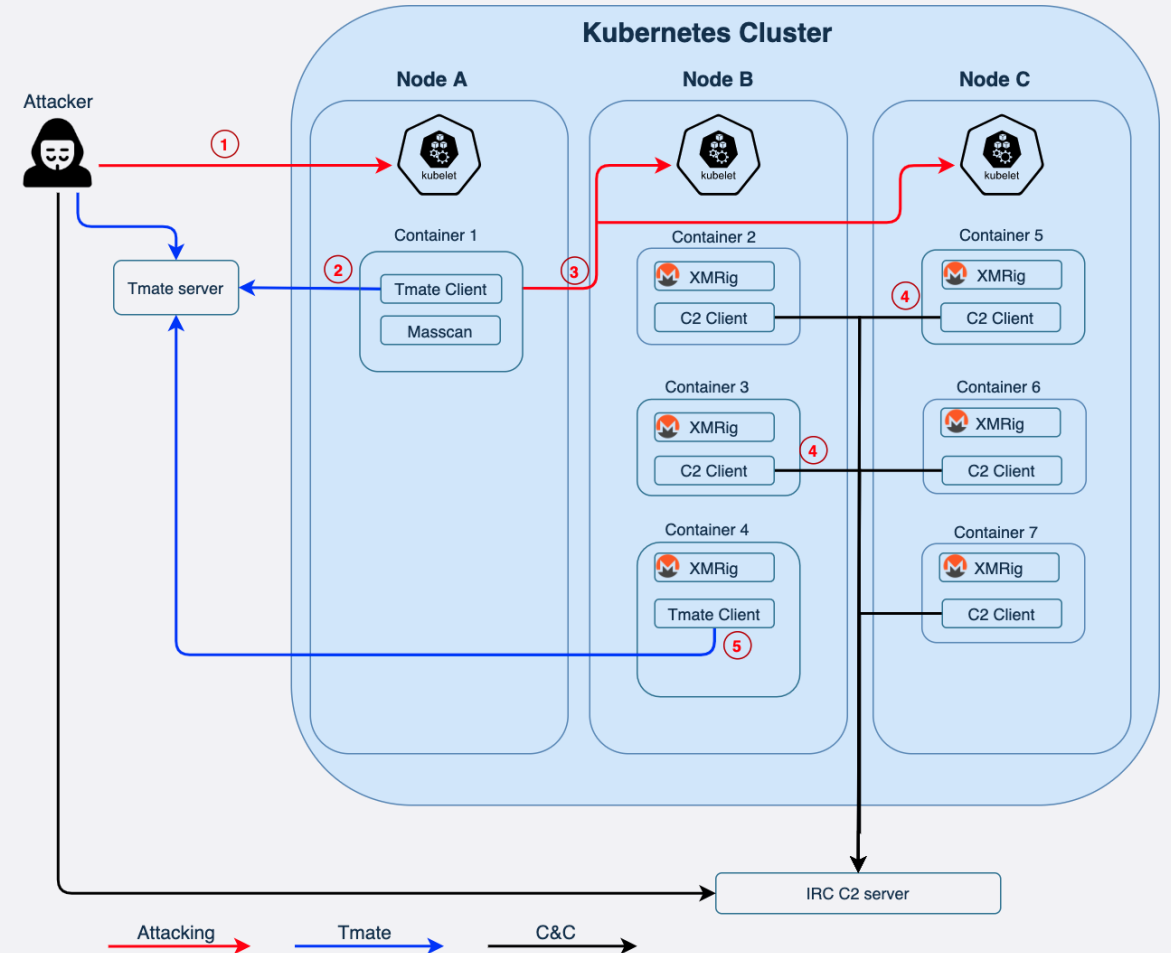
Regulatory
requirements



Malware Targets Kubernetes Clusters

“Hildegard”

- Targets cloud and container infrastructure to mine for cryptocurrency
- Deploys malicious code that targets exposed Docker daemon APIs
- Has been active since early 2021
- Potential exfiltrate sensitive data from tens to thousands of applications



Insider threat & vulnerabilities

Outsourcing network access

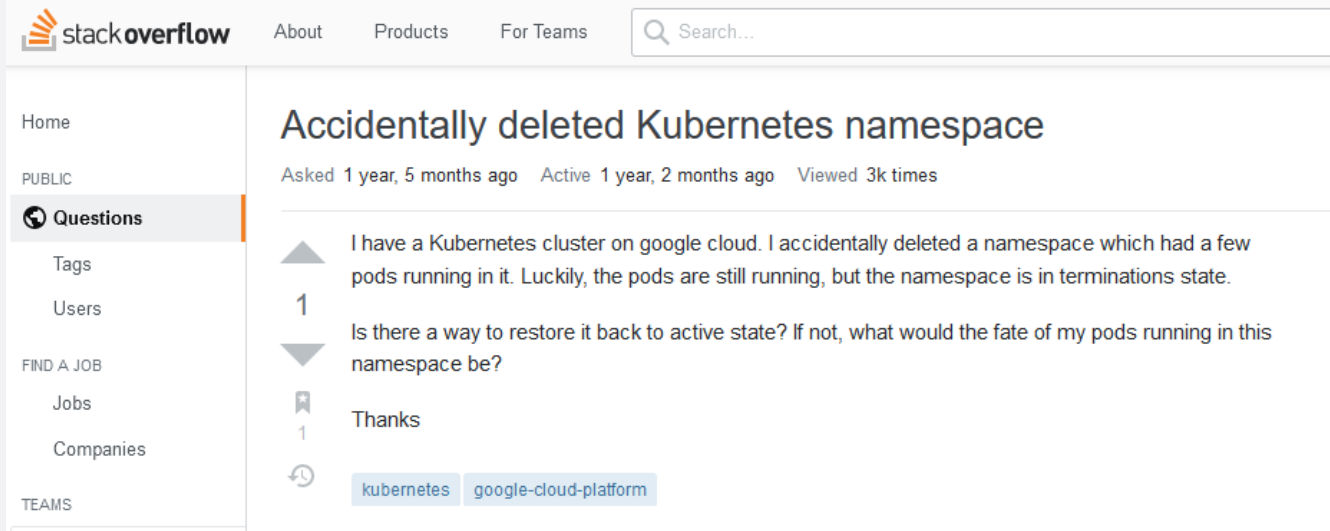
- Ransomware is hard!
 - Find an entry point
 - Compromise user accounts
 - Find Misconfigurations
 - Find vulnerabilities endpoints
- “Network Access Sellers”
 - Anywhere between \$300 and \$10,000
 - Target by industry
 - Type of access (**RDP**, Citrix, SSH)



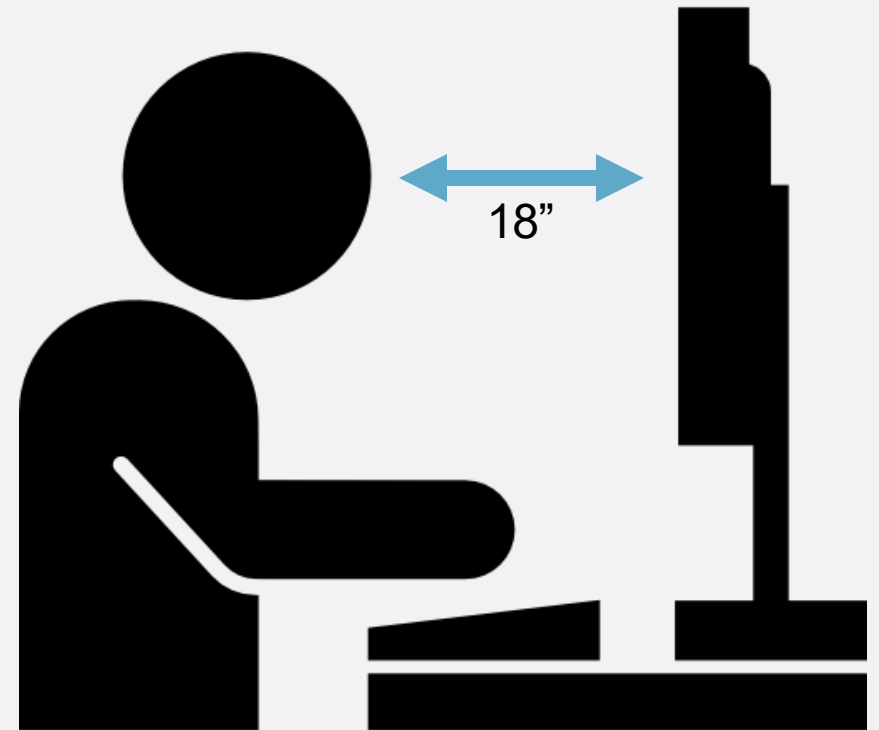
Accidental Deletion

People make mistakes

- Human errors are bound to happen.
 - Be prepared when they do.
- Someone accidentally deleted a namespace



CODE-18



Ransomware Preparedness

Better safe, than sorry

Updates

Routine Audit &
Security

Education

Backup, Backup,
Backup!



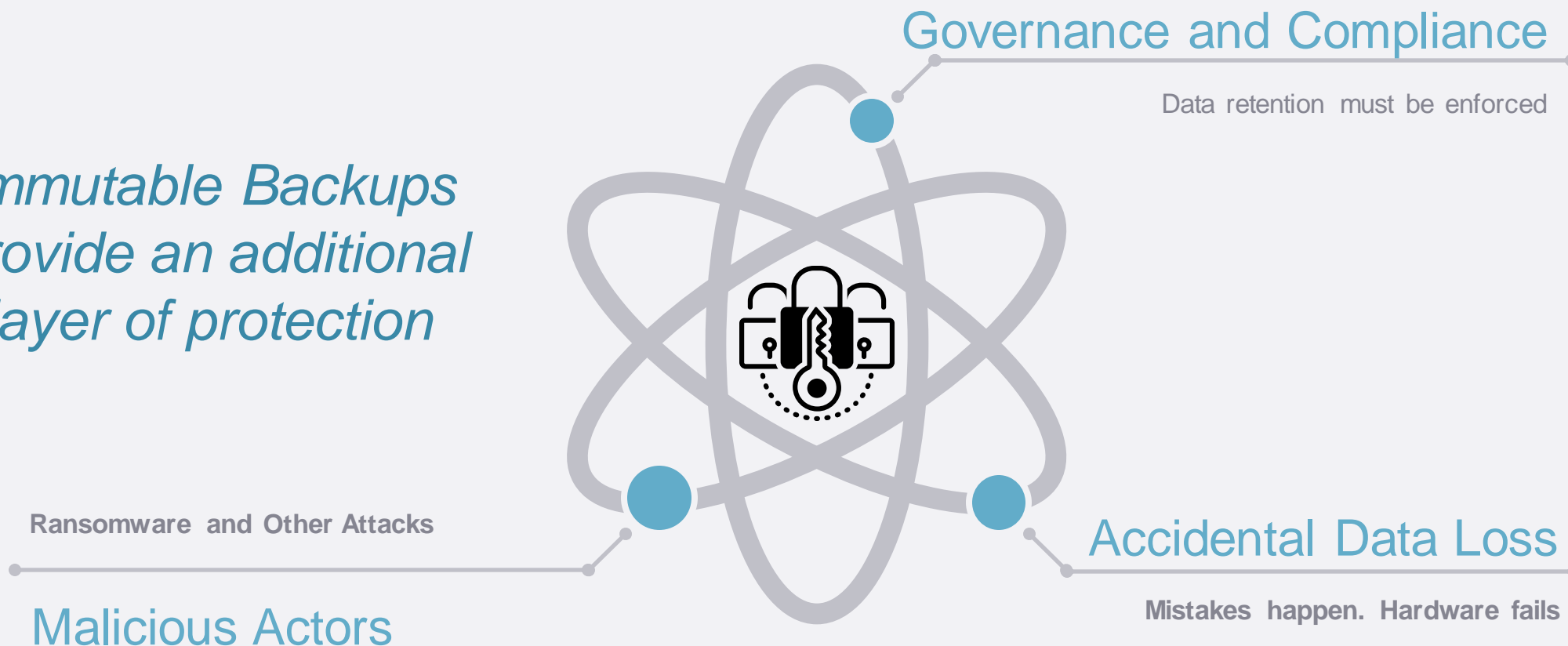
The background of the image is a composite. It features a close-up of a person's hand holding a smartphone, with the hand positioned as if about to tap the screen. This is overlaid on a blurred, high-angle view of a city skyline at night, with numerous illuminated buildings and streets. The overall color palette is dark, with blues and greys from the city lights and the person's clothing, contrasted by the warm yellows and oranges of the city lights.

Tip

Master the 3-2-1 Rule

| The Need for Immutability

*Immutable Backups
provide an additional
layer of protection*



Backups in Object Storage

- Direct database integrations
- Data protection vendors
- File system support
 - Example FOSS projects: Restic, S3FS, Kopia






Write
Once
Read
Many



| Backups with S3 Object Locks



-  Determine Immutability Requirements
-  Use Configured buckets as Backup Targets
-  Backup and Restore Data

| S3 Versioned Buckets

- Configured at the bucket level
- Required for Object Locks
- Locks apply to versions of an object



```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration Status=Enabled
```



Retention Modes

Governance



Override with

`x-amz-bypass-governance-retention:true`

Requires

`S3:BypassGovernanceRetention`

Compliance



Cannot be overridden



| Object Retention

Legal Hold



Update w/
`S3:PutObjectLegalHold`

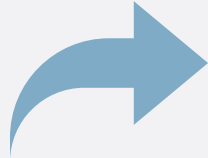
Retention Period



Cannot be modified for a set duration

Taking a Backup

```
response = client.put_object(  
    Body=b'bytes'|file,  
    Bucket='string',  
    ...  
    ObjectLockMode='GOVERNANCE'|'COMPLIANCE',  
    ObjectLockRetainUntilDate=datetime(2015, 1, 1),  
    ObjectLockLegalHoldStatus='ON'|'OFF',  
)
```



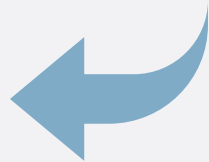
Use same API to
perform PUTs

Set ObjectLock
Parameters

Save the
returned Version ID

Restore a Backup

```
response = client.get_object(  
    Bucket='string',  
    Key='string',  
    VersionId='string',  
    ...  
)
```



Use same API to
perform GETs

Set Version
ID Parameter

Save the
returned Version ID

DeleteBackups

```
response = client.delete_object(  
    Bucket='string',  
    Key='string',  
    VersionId='string',  
    BypassGovernanceRetention=True|False,  
    ...  
)
```



Version ID is optional

No Version ID => Latest Version is a Delete Marker

Version ID => Version is Deleted

Refresh Retention

```
response = client.put_object_retention(  
    Bucket='string',  
    Key='string',  
    Retention={  
        'Mode': 'GOVERNANCE'|'COMPLIANCE',  
        'RetainUntilDate': datetime(2015, 1, 1)  
    },  
    VersionId='string',  
    BypassGovernanceRetention=True|False,  
    ...  
)
```



Call PUT

on PutObjectRetention

Set future value for
RetainUntilDate

Save the new
returned Version ID



Thank
You

Ready to take your questions