PROTECTING ENTERPRISE CONTENT AGAINST MALICIOUS ATTACK, CRYPTO-LOCKING OR HACKING

PREPARED BY
Xenit Solutions

xenit
return on content

IN COLLABORATION WITH
CARINGO

caringo

TABLE OF CONTENT

1. **Delivering the CIA Promise**
    - Prevention
    - Detection
    - Containment
    - Repair and Healing
2. **Risks**
    - Ransomware
    - Corruption
    - Targeted attack
    - Accidental deletion
3. **Mitigation**
    - Caringo Content Gateway
    - Backup cluster
    - Swarm versioning
    - Content integrity assurance
4. **Conclusion**
    - Building a moat with object storage protects your documents

INTRODUCTION

Malware, ransomware and crypto-locking are a daily threat, not a virtual menace. Enterprise content repositories need to defend the integrity of content in any possible manner, keeping the **CIA** promise, i.e., **Confidentiality, Integrity and Accessibility.**

In a global security approach, we look at four aspects:

- Prevention
- Detection
- Containment
- Repair and Healing

This paper describes how Xenit delivered the CIA promise to secure documents for a Belgium-based insurance company, managed through Alfresco open source Enterprise Content Management (ECM) and stored on Caringo Swarm object storage. This is not a comprehensive strategy, we focus on one layer in this white-paper. A layered approach allows you to build many moats to protect your precious data.

1. DELIVERING THE CIA PROMISE

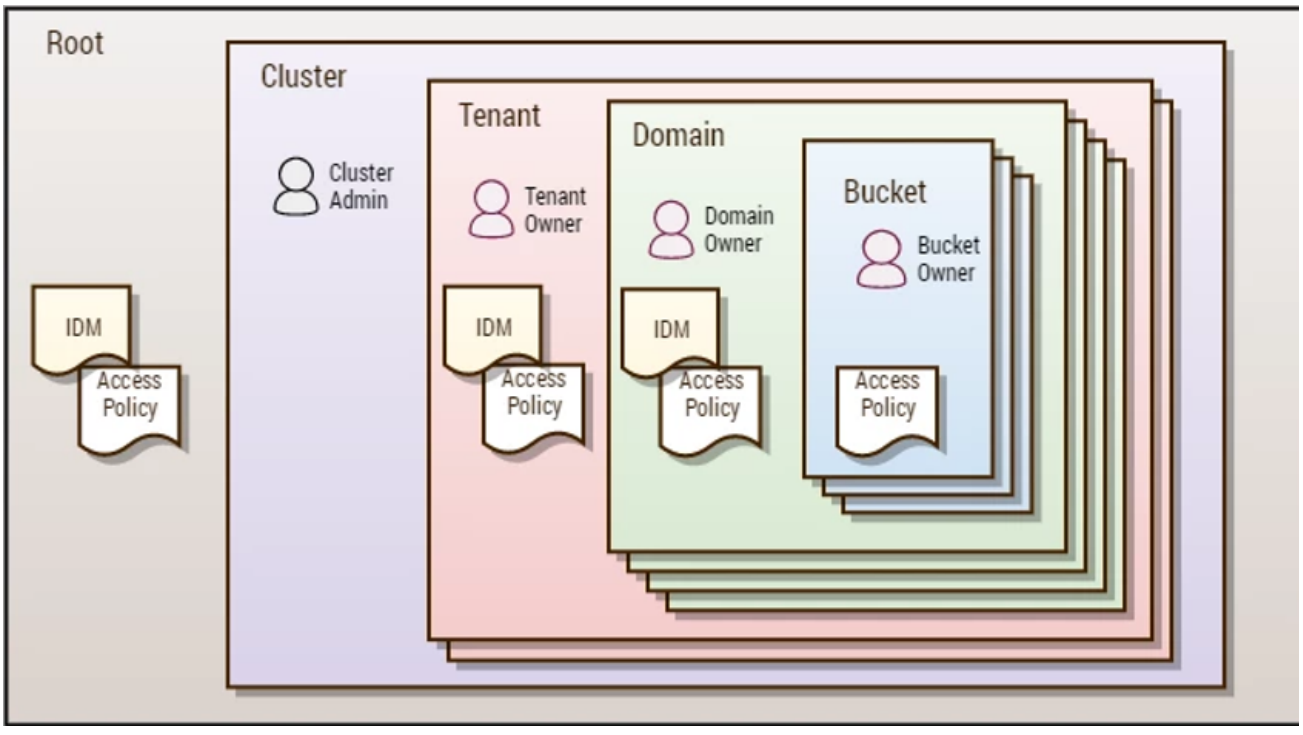
Xenit has employed a combination of processes and technology to protect enterprise content against malicious attack, crypto-locking and hacking. On the mid tier, Alfresco ECM is used to manage and store content securely on Caringo Swarm object storage. All content is encrypted with an individual encryption key.

Alfresco is an open-source Enterprise Content Management (ECM) system that manages all the content within an enterprise and provides the services and controls that manage this content. At the core of the Alfresco system is a repository supported by a server that persists content, metadata, associations, and full text indexes. Xenit created the Alfred object storage to store and retrieve Alfresco content to and from the Caringo Swarm content cluster. Today, it supports on-premise storage with Swarm and cloud storage via S3.

Caringo Swarm is a storage platform, enabled by object storage, that was developed to solve the issues associated with storing, securing and providing secure access to rapidly scaling data sets.

Data compliance and security features such as WORM (write once read many), customizable data immutability, integrity seals (upgradeable hashes of content) and support for Enterprise Identity Management and token-based authentication are standard features.
In addition to the appliance-based software deployment approach which significantly limits intrusion and attacks, data in Swarm is stored throughout a cluster with customizable protection methods in a key/value-based method, further limiting exposure. Additionally, Swarm provides a Content Gateway layer that serves as an authorization layer directing storage and access requests. This delivers an added layer of protection with detailed Identity Management (IDM) and Access Policy enforcement.

As shown in the diagram below, it is possible to define a hierarchical protection structure, from cluster, tenant, domain to the lowest bucket level. Your access and protection policies can be differentiated alongside this structure. As an example, highly confidential financial data are stored in a bucket with stringent modification and access policies. Documents shared with partners could be stored in a different tenant, shielding all internal documents from the documents that are shared with one or more partners. Above and beyond this storage protection layer, there is the enterprise access control (as exercised by Alfresco) on a business level.



• PREVENTION

As for prevention, in larger-scale archives and for INS, the storage of content is designed to strongly limit the possible expansion of a virus or ransomware attack, as the link or connection between 2 pieces of content are basically non-existent and have no common path.

For additional security, we add a layer (the Caringo Swarm Content Gateway) to control who is accessing what, and thus limit any unwanted intrusion. Swarm storage nodes are not generic computing units, and thus allow only limited interactions (as defined by the content storage protocol). Although it is possible for a fraudulent user to to retrieve, corrupt and modify content via the protocol if configuration allows, this is highly unlikely to occur if your organization follows the recommended best practices.

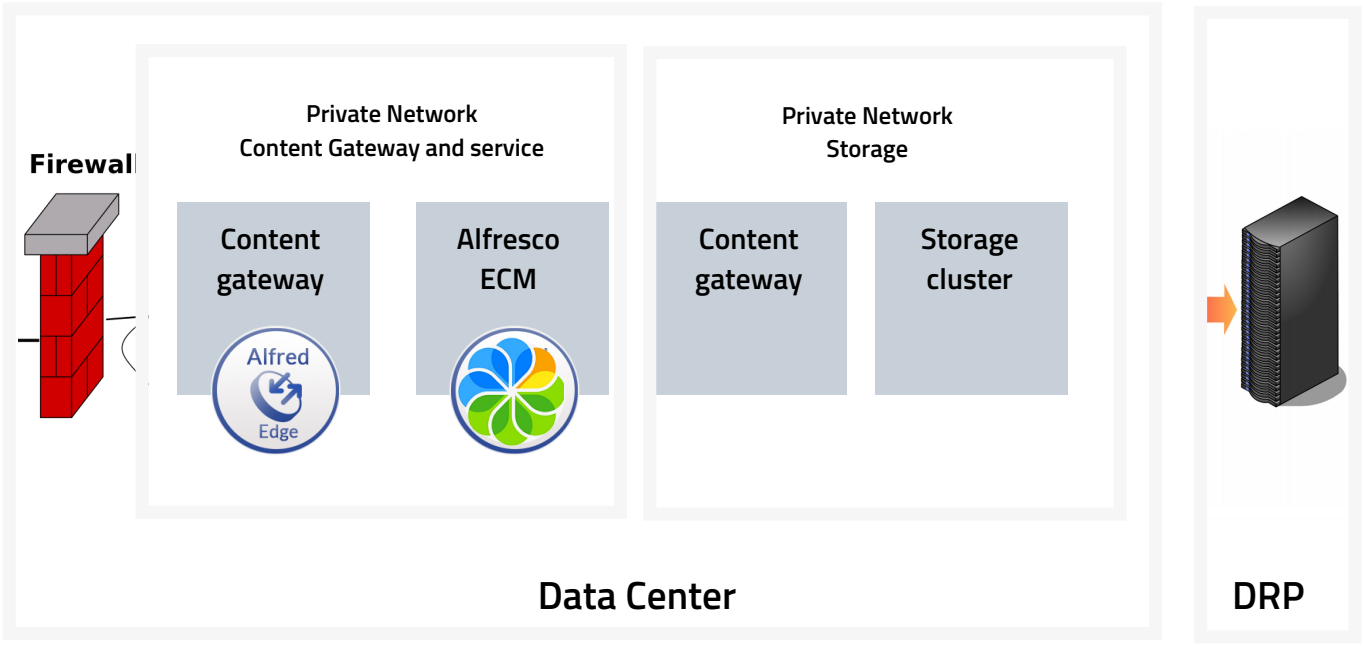
To ensure data is protected, we recommend that all access requests to content be logged, and that rules are set to intervene when suspect behaviour is detected. Suspect behaviour includes:

- Modifying content (this is very seldom required for an active archive use case)
- Modifying content in **bulk** (even more suspicious, unless a user with specific identity and authority performs such task, under controlled conditions, on a temporary basis, and on a well-defined set of content items)

• DETECTION

Detection can happen at the highest level (Application Gateway), but also at the level of the content storage, fronted by the Content Gateway.
At the business level, we can set up smart rules based upon business insight; at the storage level, we can set rather basic rules without refinement but more resistant to outside hampering.

First, all access to your content should be authenticated with the real user requiring information (not “mister portal”). Secondly, set a trigger on queries on the full repository (* like queries, anything that returns a large set of results). Thirdly, block mass modification by any external user or an internal user that works outside of her or his business mandate.



• CONTAINEMENT

Containment is what should happen after intrusion is detected. (Note: Intrusion and corruption can happen for a long time before the negative consequences become imminent. This is a worst-case scenario; we want to detect as soon as possible that our content is starting to be affected, and if this is the case, protect any copies or secondary assets so we can easily restore the data).

As improbable as any crypto-like attack is to succeed given the built-in data protection of Swarm object storage, there are steps that can be taken in the unlikely case that an infection is detected.

For example:

- Close the Swarm Content Gateway
- Shut down the Alfresco content service
- Verify the integrity of documents using a cryptographic hashkey (and thus identify documents that have been altered without an explicit and approved business operation)
- Ensure that the safeguarded copies are not overwritten by infected documents.

As the saying goes, “an ounce of prevention is worth a pound of cure.” So, let’s delve deeper into the continuous data-protection capabilities that on-premise, S3 compatible Swarm storage brings to the table. This includes life points, immutability, versionable policies, deletion prevention (undeletable) and multiple copies to bring integrity to the repository. This formidable list of features is why we trust that Swarm offers the same durability level as Amazon S3, even in the face of a cryptolocking or another type of viral attack.

• REPAIR AND HEALING

Caringo provides data resilience and built-in data protection using several features, including Elastic Content Protection (ECP) which allows selection of data durability (level of 9s) using either replication or erasure-coding. This can be set at the cluster policy or “per object” level. Any number of replicas or erasure-coding schema can be selected for each object, with the only limit being the number of physical nodes in the cluster. Replicas/EC segments are intelligently distributed for best availability/protection. Additionally, every node in the cluster runs a continuous Health Processor (HP) which automatically detects and recovers from failures (e.g., bit rot, drive failures, etc). This core functionality resides in every node so recovery times decrease as capacity grows.

In the event of a total failure or if a hard drive fails for any reason, the cluster reacts quickly and initiates a volume recovery process for each missing drive. The recovery process rapidly creates additional replicas elsewhere in the cluster of all objects that were stored on the now missing drive(s) so that each object again has two replicas. This is the default behavior. The user is free to choose any number of replicas to meet their desired level of durability.

2.RISKS

• RANSOMWARE

Ransomware attacks are often automated exploits using known vulnerabilities in computer systems. They almost always encrypt file systems.
The object storage technology of Caringo Swarm that is used to store the documents in Alfresco for INS is not exposed as a file system and does not internally use a file system. Swarm is a stripped-down Linux OS with a minimal internal file system that is NOT used to store content files.

Ransomware type of attacks generally spread across a file system-based technology, therefore, Swarm offers exceptional resistance to these types of attacks.
The ransomware attack would have to be specifically targeting Caringo Swarm and this is highly unlikely. On the client, it would be possible to retrieve a document, crypto-lock it, and store it again in the Swarm repository. However, this is far more difficult to do at scale for a ransomware attack. And, the immutable and versioning features of Swarm protect the server against the locking of content.

• CORRUPTION

The data on the disks of Caringo Swarm can be affected by bit rot, causing data corruption. However, Swarm does proactive health checks to detect and correct corrupt streams and can phone home to ensure that disks are replaced before there is an issue.

• TARGETED ATTACK

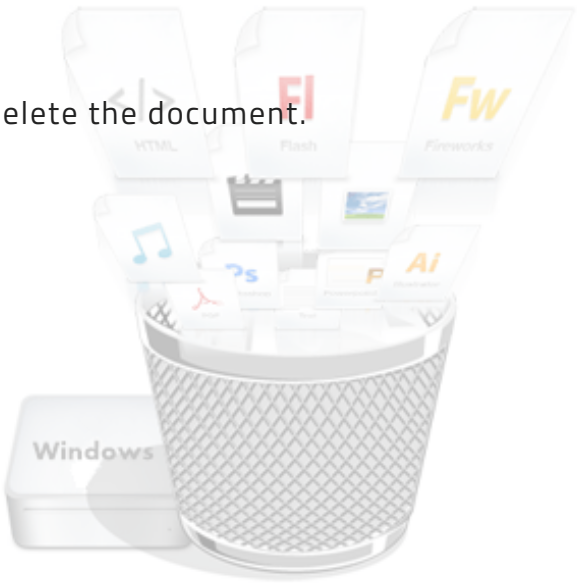
While it is not possible to completely eliminate the possibility of a breach, measures have been taken in a layered approach to make it difficult to intrude, and cause major damages. The Swarm nodes in the cluster are isolated in a VLAN, together with the Alfresco nodes and the Swarm controller. Our hoster's best practices are in place to protect all servers from attackers.

• ACCIDENTAL DELETION

There is always a possibility that users, system administrators or developers delete documents by mistake. Alfresco has a trash can on the application level. A delete on an Alfresco document would result in a document in the trash can. Documents can be easily recovered from the trash can, and putting documents in the trash can is an operation that does not interact with the content store.

Emptying the trash can is not automated, and needs to be done manually. When documents are removed from the trash can, the document is not deleted in Caringo Swarm, but marked for deletion in 50 days.

After these 50 days, the Swarm health processor will delete the document.



3.MITIGATION

• CARINGO CONTENT GATEWAY

Caringo Content Gateway acts as a proxy between Alfresco and the Swarm Cluster and allows you to implement authentication and authorization. In addition to authentication, there is also the possibility to limit the quota for bandwidth and storage. These tools could be used to limit the impact of an attack.

PROS

- No extra license cost involved
- Extra layer of security
- Provides additional insights in the Caringo Swarm cluster

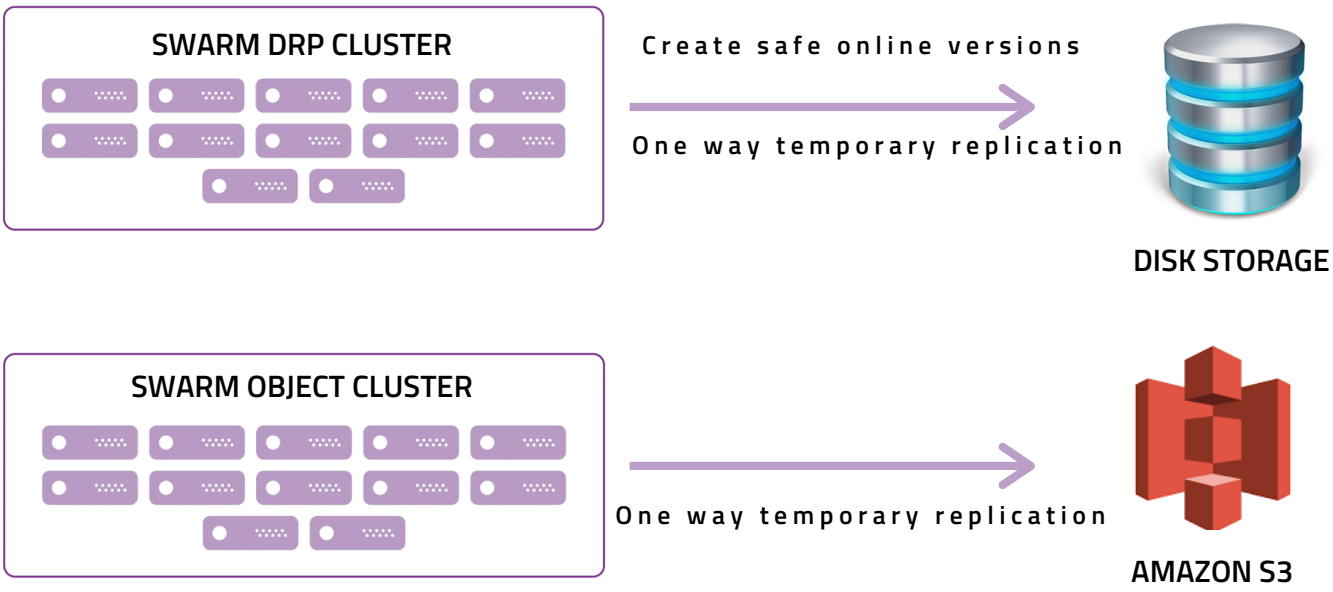
CONS

- Dependency on ElasticSearch cluster

• BACKUP CLUSTER

To protect against any attacks, bugs, or errors on the production Swarm cluster, it is possible to create a backup to another site. There are multiple possibilities:

- "One way temporary replication": Backup to a Swarm cluster (need to replicate to the same version of Swarm). Holding the documents in a disk storage facility keeps them available. With Caringo Darkive, you can optimize energy usage while storing the documents online, which facilitates retrieval considerably.
- Backup to Amazon S3 (needs Swarm 11).



In both cases, we would need to schedule when the replication is activated, rather than keep it active all the time to avoid replication deletes and overwrites. When we switch off the back-up cluster, we ensure that no contamination can happen, because there is no network connection, and no "powered" server reachable.

PROS

- Resembles a classical off-site backup. Protects against a broad range of risks, including cryptolocking and data loss

CONS

- Cost of an extra cluster or Amazon S3 monthly storage fees
- Bookkeeping to turn off and on the replication

• SWARM VERSIONING

Caringo Swarm has its own versioning feature (this is not linked to Alfresco's versioning). This would mean that an update, or even a delete of a document, would result in a new version of the document. Old versions remain available using an API.

PROS

- Easy to implement on existing content
- Versioning is policy driven, so you can select what you do and do not want to keep

CONS

- There is still an API to remove old versions
- There is still some work involved in recovering the versions of 'infected' documents.

• CONTENT INTEGRITY ASSURANCE

Swarm object storage provides methods for allowing applications to obtain and validate **integrity** guarantees on the stored data. In this context, integrity is an independently verifiable guarantee that the data returned for a given name or UUID is exactly the same data that was stored using that name or UUID, perhaps many months or years in the past. This is done by **hashing** the data using a cryptographic hash algorithm.

Content metadata is *not* included in the hash. If the application stores the name or UUID and its associated hash value, these can be used later to verify the content has not changed, either through accidental or malicious means.

It is possible to implement a similar process like Swarm does on the level of Alfresco. A health processor that loops over all the documents in the background to check the integrity, using the hash stored in the content URL.

This could help us spot issues early. A background health processor could also be useful in other scenarios, for example, to correct or cleanup metadata of documents.

PROS

- Makes it easy to spot a broad range of problems with content

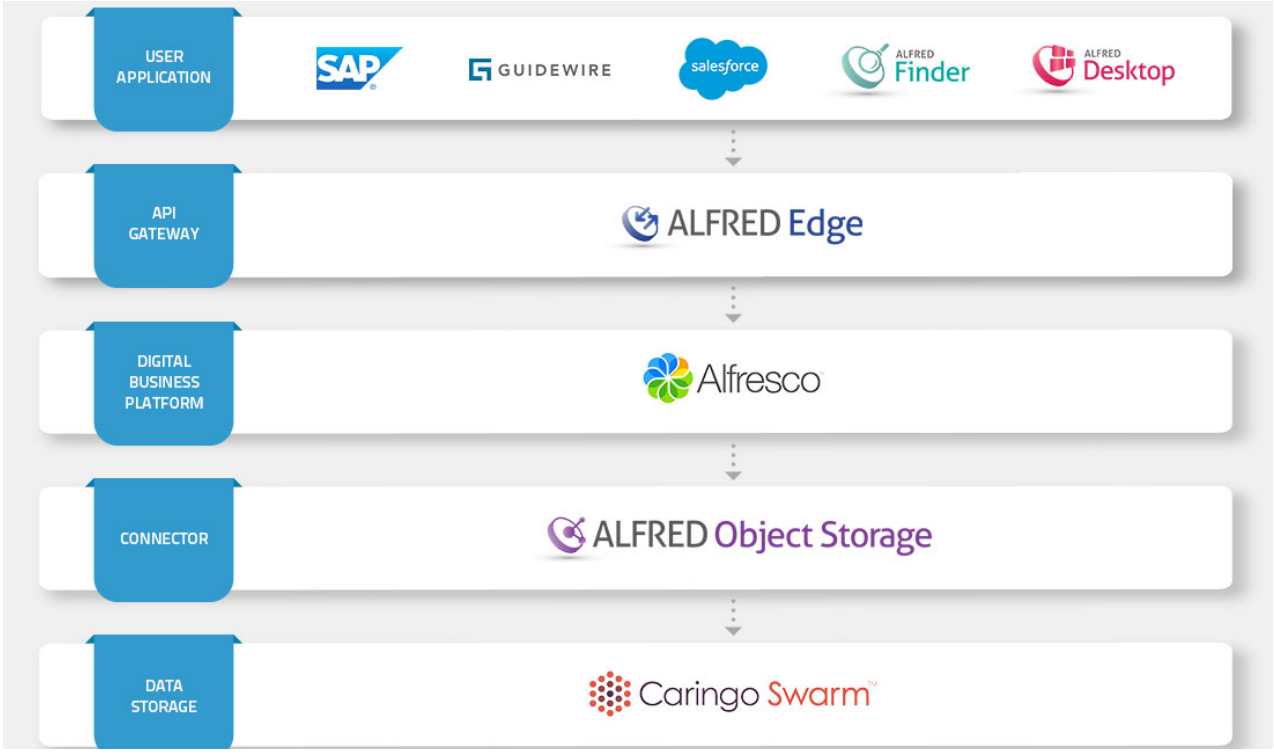
CONS

- There is an implementation cost involved.

4.CONCLUSION

- ## BUILDING A MOAT WITH OBJECT STORAGE PROTECTS YOUR DOCUMENTS

Protection at the storage level is a powerful moat against malware and cryptolockers. With Alfred Object storage, we can integrate Alfresco with a safe object store, be it Caringo Swarm on premise or S3 in the cloud, or any mix that fits your use case.



True object storage on premise or in your cloud (data center) offers multiple avenues of protection:

- **Swarm is a black box software appliance** that does not require an installation of an OS or a file system as a precursor. You cannot log into or access Swarm from outside, only interface through its RESTful protocol. There are no “under the cover” paths to the data.

- **Swarm supports Write Once Read Many** (WORM) data. So even if the virus has access to the data via the HTTP interface, it can not overwrite or perform transformations on the data.

- **Swarm’s native interface is HTTP 1.1**, which means that the storage is isolated loosely coupled from the OS and can not be accessed via an infected OS or file system and be compromised.

- **Swarm supports versioning,** which can be used to protect your non-WORM data. Even if a virus could encrypt or corrupt a file, there will always be a pristine version that can be recovered.

- **Swarm does not sit on a file system**. The disks are managed by the system and do not provide a file system or block interface which would open the door to data corruption should the virus have access privileges.

- **Swarm supports IP address blocking** so that only IP addresses of your choice can communicate with the system.

- **Encryption of data is an option**. Even if the virus gets to the data, the user has the ability to encrypt the data at rest so it can’t be read and misused. Further protection from theft.

caringo

THANK YOU

For information on building active archives with Alfresco and Swarm, please check out our website at https://xenit.eu/alfred-archive/.

The following links explain to you in depth, breadth and technicolor how Swarm protects your valuable content:

- **Blog**: Swarm Object Storage Protects Against Ransomware
- **Whitepaper**: Protecting Data with Caringo Swarm Object Storage
- **Whitepaper**: Elastic Content Protection Technical Overview

Store safe!