

MICROSHARD

# Secure Implementation & Audit Guide

January 2021



**SHARDSECURE**<sup>®</sup>

# Contents

<b>Overview</b>	<b>3</b>
<b>Microshard™ Technology</b>	<b>3</b>
<b>Terminology</b>	<b>4</b>
<b>Data Security</b>	<b>5</b>
Shard size	5
Storage locations	5
Storage types	5
<b>Cluster Security</b>	<b>6</b>
Hardening	6
Maintenance	6
Network access	6
<b>Authentication/Authorization</b>	<b>7</b>
Administrative access	7
Principle of least privilege	7
Application access	7
<b>Backup / Restore</b>	<b>8</b>
Backup	8
Restore	8
Disaster recovery	8
<b>Questions for Auditors</b>	<b>9</b>

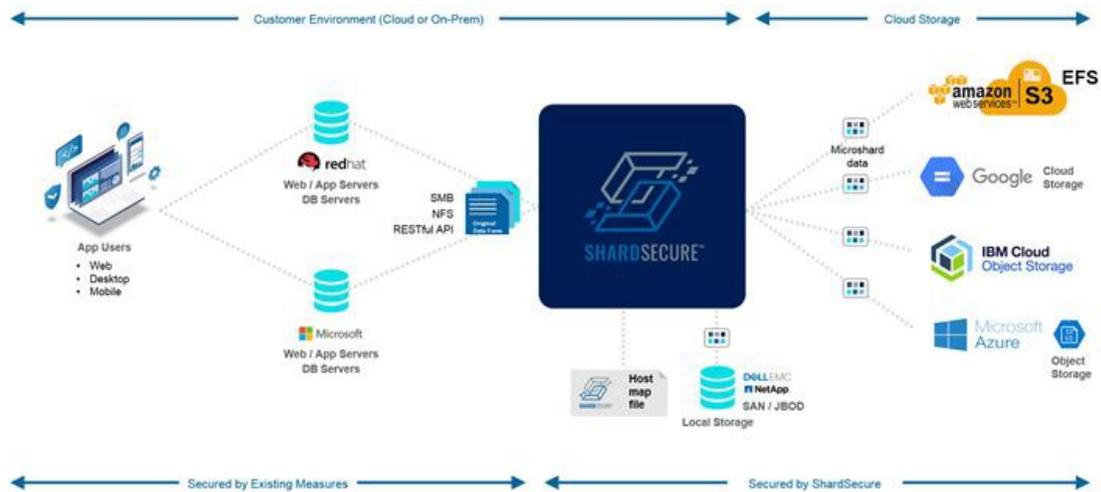
# Overview

This document is designed to aid security implementers to securely plan a Microshard deployment, as well as auditors in assessing risk and compliance with data secured by Microshard technology. There are unique concepts and considerations when auditing Microsharded data, and careful considerations to take into account when deploying Microshard Technology in your environment. As with all security tools, the details matter.

## Microshard™ Technology

Microshard™ technology breaks digital files into tiny pieces, called Microshards, that are then poisoned with false data and distributed to multiple locations, to eliminate the sensitivity of data.

Microshard data can be distributed to multiple independent locations including onsite and cloud-based storage so shards cannot be easily tracked.



# Terminology

The list below explains the terminology used in this documentation.

TERM	DESCRIPTION
Shard	A fragment of data
Microshard	A tiny fragment of data
Shard size	The size of a single fragment
Shard file system	A location where shard container files can be stored (onsite or cloud-based)
Node	An instance used to service the clients to read and/or write data
Cluster	A synced group of load-balanced Nodes
Client	A client that will access the nodes to read and/or write data, generally an application or database
Host map	Contains information used during reassembly of Microshards into their original form

# Data Security

## Shard size

One of the key tenets of Microshard, is that sensitive data is broken up into at least 3 or more pieces, before being mixed and distributed. So, security effectiveness requires shard size to be smaller than the smallest piece of sensitive data. For example, if data includes Social Security Numbers at 9 bytes, 4 bytes would be an appropriate shard size. That way every SSN would be broken into 3 pieces before being distributed to the disparate storage locations. Whereas an 8-byte shard size would not be very secure as you could have almost complete SSNs, just missing 1 digit to reconstruct. Other sensitive data, such as audio or video files, images, etc., can have much larger shard sizes. One cluster can have multiple shard sizes per policy or data type, so it is important to confirm that the appropriate shard size is being used for the appropriate data type.

### Summary

4 bytes is a good size for any data type. Larger shard size can be used for certain data types like media files.

## Storage locations

Shard file systems are just the configured storage locations. They could be AWS S3 buckets, Azure Blobs or on-prem storage. The number of storage locations is a factor. As a file gets microsharded, it will be split into as many shard file containers as there are storage locations defined. If there are 5 storage locations configured, the original file will be microsharded and mixed, then split into 5 shard file containers, and distributed to the 5 defined storage locations. As an auditor, you should ensure there are at least a few different storage locations defined, so that all the data is not stored in 1 or 2 locations.

### Summary

4 to 10 storage locations are appropriate.

## Storage types

Storage locations can exist on-premises, in cloud environments like AWS, Azure and GCP, as well as SaaS based storage services like Dropbox. For any Microshard deployment, if there are 2 or more different platforms/environments where storage locations are defined, and the principle of least privilege is used for administrator access to the storage locations, the risk of an attacker being able to compromise all the storage locations is significantly reduced.

### Summary

Use at least 2 different storage platforms/providers, and no single administrator should have access to all the storage locations.

# Cluster Security

## Hardening

Clusters should be audited for hardening procedures, including:

- No extraneous user accounts.
- No extraneous services/ports enabled.
- Host map file should be stored offline in a secure location (2 copies actually) for disaster recovery.

## Maintenance

Cluster maintenance plans should include the following:

- Frequent Backup of Pointer Database (should coincide with storage backup schedule)
- Periodic Backup of Host Map File
- Periodic vulnerability assessments
- Patch management process

## Network access

Network Access should include the following:

- Cluster should be in a private subnet with services exposed by load balancer.
- Admin access should be limited from specific management hosts / Ips.

# Authentication/Authorization

## Administrative access

Administrative access should be restricted by IP / network and include strong authentication such as MFA or certificate-based authentication.

Consider which user directory you are using for Microshard Administrator credentials. If an attacker compromises an account that has control of the user directory, they can make themselves a Microshard Administrator.

### Summary

Microshard Engine Administrator accounts should be protected as well as possible.

## Principle of least privilege

For underlying storage access, it is best to have different administrators for different cloud platforms. So, for example, if the microshard data is being stored in both AWS and Azure, no administrators should have access to all storage locations. In the event of a credentials leak, the data will remain incomplete on both platforms, reducing risk.

Microsharding gives the ability to hide the underlying data from administrators with data storage access from being able to reconstruct sensitive data, while still allowing them to perform backup and restore operations.

### Summary

Do not have any privileged accounts have access to all storage locations.

## Application access

- Application ID and Secret pairs should be created for each individual application accessing the microshard cluster.
- Application IDs should be rotated periodically.
- Applications should be restricted by IP/network to access the microshard cluster.

# Backup / Restore

The pointer database and host map file reside on the microshard engine. The pointer database is very dynamic and changes with each file write. The pointer database is responsible for the instructions to reassemble a given file. The host map file is responsible to know where the microshard containers are stored. The host map file is relatively static and only changes when storage locations or policies are modified. Both are needed to reassemble microsharded data. These files should be treated as extremely sensitive, i.e., like a private key.

## Backup

- The host map file should be backed up and stored offline in at least 2 locations in a secure location.
- The pointer database should be backed up frequently, and on the same schedule as your storage location backups/snapshots to keep them in sync.

## Restore

- In the event your storage locations are restored, you will want to restore the pointer database from a backup made at a similar timeframe as your storage locations. This will ensure the pointer database and stored data are in sync.

## Disaster recovery

- In case of a total failure, a new Microshard cluster can be installed and initialized with the host map file and a restore of the pointer database for full-service restoration.

# Questions for Auditors

## How many Storage Locations are in use?

There should be a minimum of 4 storage locations configured to ensure data is sufficiently distributed.

## Do any administrators have access to ALL storage locations?

It is best to use multiple storage platforms, and to ensure that no single administrator has access to all storage locations.

## What is the shard size of your microshards?

4 bytes is sufficient for almost any scenario. Anything larger than 4 bytes should be used only in scenarios where larger files with less data sensitivity at the byte level, e.g., media files.

## What devices have network access to the microshard engine?

Network access should be limited, similar to a database server.

## Do Admins use the same credentials for the microshard engine as for cloud access?

Depending on the security controls in place, you may want to ensure the administrators aren't using the same credentials for cloud access as Microshard engine administration.

## Where are the pointer database and host map file stored and how are they backed up?

- Host map file should be stored offline in 2 secure locations after every configuration change.
- Pointer database should be backed up regularly, in sync with the storage backup schedule.

## What controls do you have on user/application access?

This is the attack surface that remains after microsharding. Ensure there is robust security on user and application access to data.



**SHARDSECURE®**

 +1 (800) 760 9445

 [info@shardsecure.com](mailto:info@shardsecure.com)

 @ShardSecure

101 Avenue of the Americas,  
9<sup>th</sup> Floor

New York, NY 10013

United States of America