absio®

De-Risking Unstructured Data with Data-Centric Security and Control

720 836 1222 **sales / support**

sales@absio.com **email**
www.absio.com **web**

8740 Lucent Boulevard, Ste 101
Highlands Ranch, CO, 80129

Every day brings new reports of data breaches resulting in exposed personal information, financial fraud, theft of intellectual property, divulged military and diplomatic secrets, or humiliating personal or corporate disclosures.

The mechanics of data breaches and the financial and political profit motives behind them are well understood. What is less well understood is the characteristics of unstructured digital data that puts it at risk of being breached, and how it can be de-risked.

THE INHERENT RISK OF UNSTRUCTURED DATA

Unstructured Data Defined

The vast majority of digital data is unstructured, that is, comprised of individual files such as documents, spreadsheets, images, videos, music, email, and datasets. Structured data is information stored in large relational databases; it only accounts for a small percentage of all digital data. That percentage is shrinking rapidly due to new technologies which enable processing of unstructured data as if it were structured, while avoiding the cost of data entry. Dataset files are the results of a database query. Large relational databases are very difficult to steal, but datasets are stolen frequently. By definition, data is breached when copies of usable files fall into unintended hands.

The Making of Unstructured Data

Software applications process inputs from users and manufacture files. Even though computers and software have grown exponentially in power, speed, complexity, and reliability, applications today still manufacture files the same way they did in the 1950's. Applications digitize information created by users and add metadata which enables the file, once saved, to be used again. Such usable files can be read, played, altered, published, or transmitted to anyone, anywhere, at any time by anyone who possesses a copy of the file. When such usable files fall into unintended hands as the result of a data breach, their subsequent use cannot be seen or stopped by the user or organization from whom the data was breached.

Inherently Unsecure versus Inherently Secure Files

Usable files are inherently unsecure. To make files inherently secure, applications must be able to create files in a manner that assures they are unusable by unintended parties regardless of the means

by which an unintended party gains possession of the files. Whether the file was on a lost or stolen device, copied to physical media and exfiltrated by a malicious insider, emailed, posted to a file share, released by accident, exfiltrated by malware, or intercepted in transmission, if the file remains unusable by unintended parties, it is inherently secure.

Post-Creation Security Implementation Issues

Usable files can be secured post-creation. However, adding post-creation security through separate processes is costly, complex, and inherently error-prone. It relies on busy, fallible human beings adhering to policies dictating which files must be secured and which need no protection—and then actually taking the time to find, select, and secure the specified subset of all files to secure. Post-creation file security requires complex file security management, that is, continuous monitoring of files in servers, end devices, and virtual machines to determine their current protection status (encrypted, not encrypted). Furthermore, when usable files are shared, the sharer usually loses the capability to secure them after the fact. Sharing usable files outside of the network outsources the security of the file to the party with whom they were shared.

DE-RISKING FILES

Traditionally, cybersecurity solutions have focused on protecting the environment where files are stored, and the connections through which files are transmitted. While this form of security is necessary, a data breach by definition indicates that the environment where files are stored or the medium through which they have been transmitted has been compromised. In order to prevent data breaches from exposing information, the files themselves must be protected in storage and during transmission.

What is Data-Centric Security and Control?

Data-centric security and control is security and control applied to individual files, not to a network, device, or transmission media. Data-centric security and control is implemented in existing or new applications by software developers.

What are De-Risked Files?

Files are de-risked when they meet two criteria:

1. De-risked files are unusable when in the possession of unintended parties. Possession of unusable files does not yield information and as such, poses no risk.

2. When de-risked files are intentionally shared, the sharer can predetermine by whom, how, where, and for how long, the file can be used.

What are the Necessary Components of a Data-Centric Security and Control Ecosystem?

Data-centric security and control must necessarily de-risk files in all three states:

- Files at rest — stored on end devices and on servers on premise or in the cloud
- Files in motion — being transmitted both inside and outside an organization's network
- Files in use — decrypted by an authenticated user and open in an application on devices an organization may or may not control

Following is a list of data-centric security and control ecosystem components and the security and usage issues they address.

Automatic, Granular File Encryption

When a new file is created (first saved), the application automatically generates a unique encryption key for that specific file, generates a unique file ID, and then encrypts the file and stores it. Once created, files are never decrypted again with the exception of decryption while in use after authentication requirements have been satisfied.

Given sufficient computational time and power, even strong encryption can be broken. Because files are individually encrypted with their own unique key, successful brute force decryption only yields access to one usable file, not all the files in an encrypted folder, end device, or server. In addition, only the file being requested by an authenticated user or application is decrypted in memory at any given time. Since granular encryption at creation is automatic, all files, regardless of their content, are secured. Vulnerabilities arising from failing to secure usable files post-creation are eliminated.

As computational power increases over time, the time required to brute force decrypt a file of a given encryption strength decreases. With the advent of quantum computing, the rate at which encryption weakens will accelerate. Applications, utilizing the same increase in power, can periodically re-encrypt existing files using stronger encryption, thereby eliminating weakening encryption risk.

Unlike disk encryption, all files, except those in use, remain individually encrypted while the device is running. File security management is simplified if all files in a given environment are by default individually encrypted.

Nonsensically-Named Files and Randomized Storage Locations

As mentioned previously, given sufficient computational time and power, even strong encryption can be broken. If encrypted file content is identifiable by its name or location, brute force decryption

techniques can be used against known valuable files. In a data-centric security and control ecosystem, files are stored with nonsensical names in a randomized folder structure, disabling the targeting of specific files for brute force decryption based on their file name or folder location. From the perspective of an authenticated user or application, folder structures and file names are unchanged. When even a relatively small number of individually-encrypted, nonsensically-named files fall into unintended hands, even a nation-state actor may not have sufficient computational resources to brute force decrypt a sufficient number of files in a short enough period of time to gain useful information.

Automatically individually encrypting each file and storing it with a nonsensical name in a random location virtually eliminates risks arising from copies of files falling into unintended hands.

Endpoint Encryption Key Management

Centralized encryption key servers inherently reduce resiliency and increase latency. Key servers in the cloud and on premise are subject to attack as are connections between key server and devices being served. Successful attacks prevent access to encrypted data, and if a key server is corrupted or destroyed, access to encrypted data may be permanently lost. When encryption keys are generated and managed by the application on the device where the files are created, stored encrypted files can be used and new encrypted files can be created by authenticated users while disconnected. Eliminating key servers eliminates key server vulnerabilities. Encrypted data and encrypted keys can be stored locally, in the cloud, or both as needs dictate.

Latency is introduced when applications must connect to a key server to encrypt or decrypt. Endpoint encryption key generation and management eliminates latency arising from slow connections.

Cryptographic Usage Authentication

Strong authentication is required to ensure that only intended users are able to access and use de-risked files. Each user is programmatically assigned a unique ID. In a data-centric security and control ecosystem, the application generates and stores a nonsensically-named authentication key file for each unique user when they are created as an additional authentication factor required to access the de-risked file. Gaining a user's login credentials without the user's key file stored on a device does not enable illegitimate users to log in to a user's account.

When files are created, the creating user's unique ID is cryptographically bound to the file. The file cannot be decrypted unless the authenticated user's ID matches the user ID bound to the file. When files are shared, the application creates and binds a unique shared encryption key to the file, enabling the sharer and the user with whom the file is shared to open the file. Since each files has a unique ID, the destination of shared files is trackable.

Secure Intra-File Search

Search indexes are a common target for attackers, because they contain valuable information about the contents of files stored in a repository. In a data-centric security and control ecosystem, search indexes are comprised of cryptographic hashes of file content characters and words, not the characters and words themselves. Even though the search index is unencrypted, it is not a source of usable information. Since the search index is not encrypted, search performance is unchanged.

Decryption for Analysis or Ingestion

Organizations using data-centric security and control cannot be blinded by encryption. Files can be momentarily decrypted and then discarded for threat mitigation and data loss prevention analysis by applications installed in the analytical environment. De-risked files can be securely delivered and automatically decrypted for ingestion into databases. Log data needed for heuristic analysis tools is produced for consumption by the data-centric security and control-enabled applications.

Access and Use Controls

Encryption secures data in storage and in transit, but does nothing to protect information once it has been decrypted by an authenticated user. In a data-centric security and control ecosystem, the application cryptographically binds control metadata to each file at the time of creation. Users, at their option, can establish controls regarding any subsequent access, use, redistribution, or the lifespan of specific files. When the file is opened by an authenticated user, the control metadata is consumed by the software application, which controls the functions available for that particular file.

Application developers create controls. Example controls can include, but are not limited to, custom tags such as classification or other forms of labeling, enhanced authentication (e.g., key file, biometric, security token, application ID, device ID), access roles and/or attributes, file usage controls (print, copy, save, forward, etc.), geographic location, and file lifespan control. Users can choose how and when to set controls, or applications can be configured to implement controls by default when a new file is created or an existing risky file is de-risked.

The user or application creating or sharing files sets the controls, which are cryptographically bound to the file. Those settings control the application of the user with whom the file is shared, thus for files shared and stored by data-centric and security controlled applications, the security, or lack thereof, of the parties with whom files are shared, is irrelevant. Risk arising from faulty security on the part of the user or organization with whom the file is shared is eliminated.

IMPLEMENTING DATA-CENTRIC SECURITY AND CONTROL

Risky files are created by applications and they are de-risked by applications. Implementation begins when application purchasers, particularly large institutional purchasers, specify data-centric security and control requirements when purchasing applications. If an application is developed by internal resources (a "home grown" application), implementation begins when management specifies data-centric security and control requirements.

Based on risk mitigation priorities, data-centric security and control can be implemented in phases. Typically, the highest risks arise from usable files being breached from an organization's network devices, end devices, or cloud storage. Implementing data-centric security first significantly mitigates that risk, and the implementation from a software development standpoint requires less effort.

Implementing data-centric control requires more effort due to the need to develop control setting interfaces. Since data-centric control is a fundamentally new capability, policies for how it will be used need to be developed. Those policies guide a software developer's implementation of the controls.

BENEFITS OF IMPLEMENTING DATA-CENTRIC SECURITY AND CONTROL

Automatically Mitigates Insider Threat Copy Attacks

Copy attacks are by nature "inside jobs." To exfiltrate data, outsiders must be able to get inside by posing as insiders, or smuggling file-exfiltrating malware into the network or end devices. Legitimate insiders can accidentally or maliciously exfiltrate files. De-risked files can be made usable once exfiltrated, but only as deliberately and explicitly allowed. Unless all authentication and other constraints contained in control metadata are met, exfiltrated files will not decrypt.

Adds Secondary Transmission Security

Encrypted connections are a common target for attackers because many encrypted connections use out -of-date encryption protocols, and because the quality of all connections from endpoint to endpoint may not be under the control of the sharer. All files should be sent via encrypted internet connections to prevent files from being intercepted. However, since only nonsensically-named, individually-encrypted files are transmitted, compromised connections do not yield usable data. Risk arising from hacked or low security quality connections is eliminated.

Reduces Risk through Risky File Conversion

Data-centric and control is applied to existing files not currently subject to data-centric security and control. Applications convert existing risky files to de-risked files. If existing files are not easily classified as risky or non-risky, then all files in a given environment can be de-risked regardless of

content. This makes effective brute force decryption even more computationally difficult because the pool of de-risked files is larger, and may reduce time and money spent implementing decisions about which files to de-risk.

Inherently Secures Cloud File Storage

Generating and managing encryption keys as described above, in combination with automatic granular file encryption and nonsensically naming each file prior to sending it to the cloud, makes cloud file storage inherently secure and the geographical location of cloud storage servers irrelevant. The cloud storage provider does not have the keys to decrypt stored files, nor can the provider identify known valuable files. The cloud storage provider cannot access the data or enable others to do so. Application performance issues arising from key transport over slow connections or overtaxed central key servers are eliminated, and encryption computational costs are born by end devices, not the cloud service provider.

Enables Accurate Quantification of Data Breach Risk

The return on cybersecurity investment has been difficult to quantify, because spending and risk mitigation have not reliably correlated; one organization spends little and suffers no breach, another spends heavily and is breached. Requiring application providers to implement data-centric security and controls enables accurate quantification of progress towards de-risking organizational data: the degree to which data is secured from breach is the ratio of risky files to de-risked files. The closer that ratio is to 0%, the lower the risk. Accurate quantification may reduce cyber insurance premiums and exclusions.

Reduces Compliance Risk and Cost

Regulatory bodies are increasingly holding organizations responsible for the security of regulated data they share with external parties. Typically, such regulatory requirements are implemented through inclusion of cybersecurity requirements in operating agreements or by requiring the sharing organization to periodically assess the cybersecurity of each external party with whom they share regulated data, or both. In either case, a degree of regulatory liability or potential loss in the form of penalties or lost business accrues to the sharing organization if regulated data is breached from the party with whom the data was shared. If the sharing and external parties agree to store, transmit, and share regulated data exclusively within a common set of data-centric security and control-enabled applications, the risk to all parties is reduced, because the likelihood of breach is significantly reduced.

Arguably, de-risked data falling into unintended hands causes no harm. Regulated de-risked data falling into unintended hands may not be reportable if the regulations include an encryption safe harbor.

Preserves Cybersecurity Investment Value

Investments in perimeter security measures that obscure pathways to systems and data, thwart malware, analyze behavior, prevent memory exploits, gather and disseminates threat intelligence, and prevent denial of service attacks are still necessary. However, investments in perimeter security alone cannot assure files will never fall into unintended hands. Data-centric security and control combined with perimeter security assures that investment in both perimeter and data-centric security and control is preserved even if files do fall into unintended hands.

Demotivates Malicious Attacks

Malicious actors, both inside and outside an organization, gain nothing financially or politically by exfiltrating copies of files created and managed by data-centric security and control-enabled applications. Arguably, the most effective means of forestalling malicious attempts to breach data is to render breached files unusable, and therefore valueless.