IMPERVA®

HACKER INTELLIGENCE INITIATIVE

Today's File Security is So '80s

Contents

Introduction

In today's knowledge-driven economy, modern enterprises have a fluid organizational structure in which most employees have access to most data to do their jobs. The amount of unstructured data organizations create and the number of employees that need to access it are growing exponentially. Working groups are formed organically and are cross-functional by nature which means that traditional, black and white access control to files can't keep pace with the ever-changing environment. This creates a security gap in which data contained in files can be lost, stolen or misused by malicious, careless, or compromised users.

Almost any security team will confirm that the traditional static approach to file security, centered on individually granting users access to files based on their department and function is not effective because it places too much of an administrative burden on enterprise IT teams. Setting up, maintaining and enforcing permissions to grant and deny access has proven to be ineffective when it comes to securing enterprise files.

A key problem is that file permissions are easily granted but are rarely rescinded. With permissions increasing 26% after an employee's first year and 11% annually afterward, they rapidly accumulate for every user. Yet statistics reveal that users only ever access less than 1% of the resources to which they are granted permission. In other words, the vast majority of resources to which users had access held no interest or only very temporary interest to them.

The complexity of managing enterprise-level file permissions on a daily basis makes it increasingly difficult at any given moment for security teams to keep track of who has access to what. Permission inheritance between folder structures further complicates effective oversight of unstructured data. This is one reason why so many data breaches happen as a result of insiders within an organization.

To reduce risk, static permissions should remain for folders continually in use. But without dynamic permissions management for folders having only temporary interest (where 99% of files are stored), it is unmanageable for IT teams to keep pace with constant permission changes.

Instead, the permissions model should be dynamic and based on user behavior. This serves to relieve IT staff of a near-impossible task while simultaneously protecting an organization's data assets. Backed by extensive research, Imperva has developed an improved file security approach based on how users actually access files.

Using highly granular input data collected from SecureSphere audit logs, coupled with advanced machine learning algorithms, Imperva is able to build dynamic peer groups and determine appropriate permissions based on how users actually access files within an organization. This also allows IT teams to dynamically remove permissions as changes in user interaction with enterprise files occurs over time.

Once the virtual peer groups have been established for the organization, Imperva identifies and flags suspicious folder access by unauthorized users. This allows security teams to immediately follow up on critical incidents pertaining to file access.

Why Access Control Doesn't Work

Permissions are granted, but rarely revoked

Theoretically, granting permissions should include resource owners (typically a senior person related to the relevant resource) who define the access policy, coupled with IT teams serving as the enforcer. But the trigger for revoking permissions is not well defined, thereby causing permissions to accumulate.

To better understand the implications, Imperva researched the relationship between employment duration and granted permissions within an organization having more than 1,000 employees. Figure 1 shows the strong correlation between their employment date and the number of folders each user is permitted to access. On average, permissions increase by 11% annually, with the biggest jump–26%–occurring after a single year of employment.

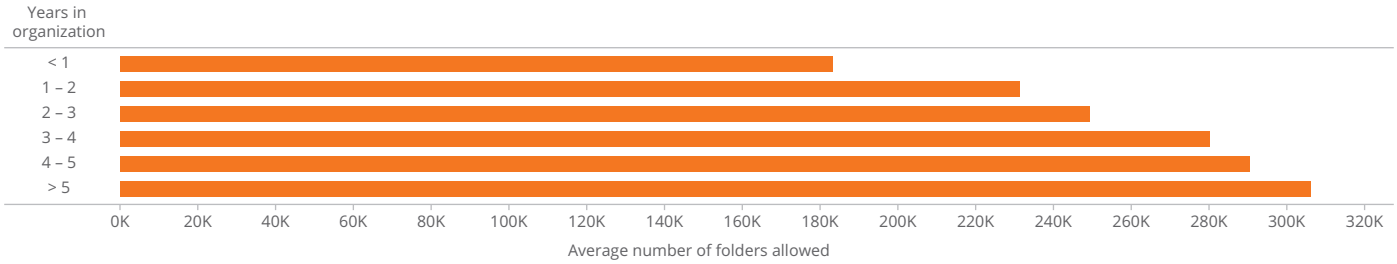

Figure 1 – The effect of employment duration on folder permissions

Users do not touch most files to which they have permitted access

In assessing the effectiveness of the traditional permissions model, Imperva compared the number of folders opened by the organization's users in a specific month to the number of folders which they had permission to access.

Figure 2 shows that 75% of the users accessed fewer than 36 folders per month; 50% opened fewer than 10 folders monthly. The number of folders opened has a very low correlation with the number of permitted folders. This reveals that typical user behavior is independent of the resource permissions they are granted.

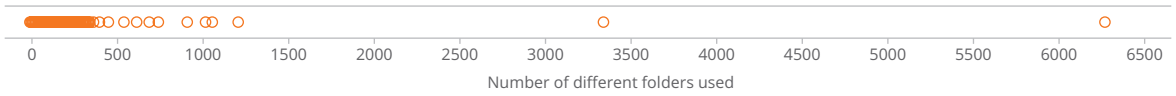

Figure 2 – Number of folders used during one month per user

On average, users are permitted access to 370,000 folders, each containing seven files. All users in our case study used less than 1% of their granted permissions (figure 3).

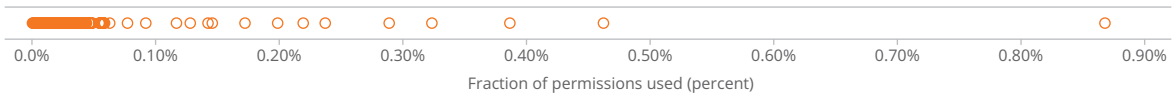

Figure 3 – Percentage of permissions used

The permissions model should be dynamic

Folder use varies over time. They can be separated into two categories:

- Folders having a continual interest. They are repeatedly used by the user and permission should be granted.
- Folders having a temporary interest. These are used over a specific duration, e.g., those containing information regarding a specific project or financial quarter. More than 99% of all shared folders are in this category.

Static permissions should remain for folders continually in use so as to reduce risk. But without dynamic permissions management for temporary interest folders (where 99% of files are stored), it is unmanageable for IT teams to keep pace with constant permission changes. A dynamic model ensures users can access such folders when required, yet also allow the organization's resources to remain protected from unauthorized access.

The Imperva Approach

Some resources in the file share should never be accessed by certain users. For example, developers do not need access to financial data. Traditional, static access controls can still be used to define resources to which a user or group should never have access. But all other files require a new security approach that is flexible, agile, dynamic and easier to maintain over time.

Using advanced machine learning algorithms, Imperva's new, research-backed, file security approach builds dynamic peer groups within an organization based on how users actually access files. By automatically identifying groups based on behavior, we can accurately define file access permissions for each user. And then based on changes in user interaction with enterprise files over time, access permissions can also be dynamically removed.

To validate this dynamic approach, several Imperva customers let us leverage production data from their SecureSphere audit logs. Containing highly granular data access activity, the log data provides full visibility regarding which files users accessed over a given duration.

The following sections present the algorithms Imperva created to solve the traditional access control problem. Statistic-backed results prove the accuracy of this automated approach.

Detecting Suspicious File Access by Building Dynamic Peer Groups

Shown in figure 4, a number of steps are required to dynamically place users in virtual peer groups according to how they access data. First, granular file access data is collected and processed. Next, a behavioral baseline is established that accounts for every file and folder accessed by each user. Based on how they access enterprise files, the dynamic peer group algorithm assigns users who may belong to different Active Directory (AD) groups into virtual peer groups. If the algorithm does not have enough information to associate a user with a specific peer group, the user is placed in a new peer group in which they are the sole member. Once virtual peer groups are established, access to resources by unrelated users can be flagged; this enables IT personnel to immediately follow up on such incidents.

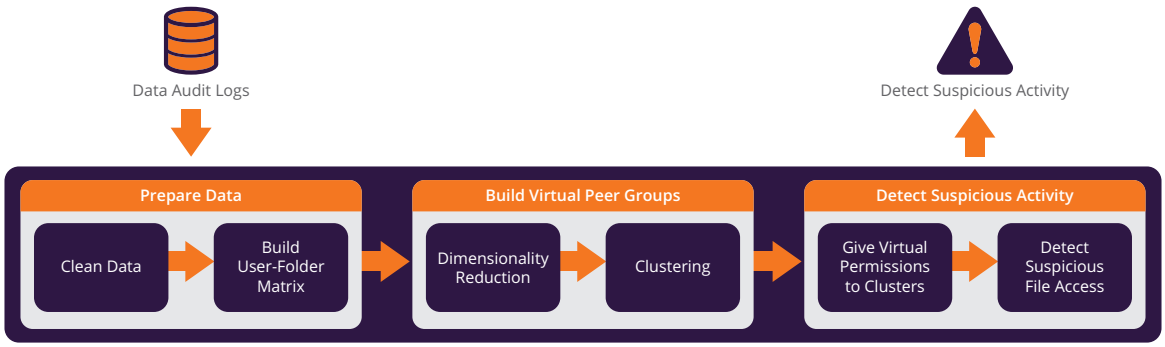

Figure 4 – Overview of suspicious file access detection process

Granular data inputs

Algorithm input comes from SecureSphere audit logs. These contain access activity that provides full visibility regarding which files users access over time. Each event contains the following fields:

NAME	DESCRIPTION
Date and Time	Date and time of file request
User Name	Username used to identify requesting user
User Department	Department to which user belongs (as registered in Active Directory)
User Domain	Domain in which the user is a member
Source IP	IP that initiated the file request
Destination IP	IP to which the file request was sent
File Path	Path of requested file
File Name	Requested file name
File Extension	Requested file extension
Operation	Requested file operation (e.g., create, delete)

Architecture

The behavioral models are created daily and simulate a sliding window on the audit data. This lets the profile dynamically learn new behavioral patterns and abandon old and irrelevant ones. Additionally, the audit files are periodically transferred to a behavior analytics engine. This improves existing behavioral models and reports suspicious events (incidents).

The behavior analytics engine is divided into the following components:

- Learning process (profilers) – Initially run over a baseline period, profilers are algorithms that profile the objects and activity in the file ecosystem and relate to normal user behavior. These include users, peer groups, and folders, as well as the correlation between the objects. Profilers are activated daily afterward, both to enhance the profile as more data becomes available, and to keep pace with environmental changes (e.g., when new users are introduced).
- Detection (detectors) – Audit data is usually aggregated over a short period (less than one day) before being processed by the detector. Activated when new data is received, detectors pass file access data from the profiler through predefined rules to identify anomalies. They then classify suspicious requests, reporting each as an incident.

Building peer groups using machine learning algorithms

To build peer groups, data must first be cleansed of irrelevant information—including files accessed by automatic processes, those that are accessed by a single user, and popular files frequently opened by many users in the organization.

Now with clean data, Imperva builds a matrix of the different users (rows) and folders accessed over time (columns). Each entry contains the number of times a user has accessed a given folder in the input data timeframe.

The matrix is very sparse because the majority of users do not access most folders. Thus, Imperva performs dimensionality reduction on that matrix to reduce both the scarcity and noise in the data. This leaves meaningful data access patterns which become the clustering algorithm input.

Imperva uses OPTICs, a density-based clustering algorithm, to divide the different peer groups within the organization into homogeneous groups called clusters. Members of a given cluster have all accessed similar folders, with a typical cluster containing about four to nine users. The process also makes certain that users in different clusters are unique.

Define virtual permissions to enterprise files

In order to define the virtual permissions model of each user, Imperva uses the notion of close and far clusters. For every cluster, Imperva determines which peer groups are close and far, based on the similarity between it and the other clusters. Distances are partitioned into two groups using a k-means algorithm; a smaller distance designates a closer cluster.

Each user is permitted access to folders accessed by others within their own cluster, or by users belonging to close clusters.

Detect suspicious file access

The detector aspect of the algorithm identifies suspicious folder access. Within a profiling period, for example, user John's access to a given folder is considered suspicious if the folder is only accessed by users belonging to clusters far from his.

Incident severity (e.g., high, medium or low) is a function of the number of users and clusters having accessed the folder during the learning period. The ratio between the first and second quantities implies severity; higher values indicate higher severity (many users grouped in a small number of clusters). Lower values (close to 1) indicate reduced confidence, as the number of users equals or approaches the number of clusters. Personal folders and files are given careful consideration when ranking severity.

Adding context to files accessed

Imperva's goal is to provide sufficient context to security teams so they can understand and validate each incident. Imperva presents typical behavior of the user who performed the suspicious file access activity. In addition, Imperva applies a label to each folder accessed during the incident; this helps SOC teams evaluate the content or relevance of the files in question.

In assigning a label to a folder, Imperva assesses the users who accessed it during the profiling period, as well as those from their peer groups. Imperva looks for the group (or groups) in Active Directory that best fits this set of users. This has two relevance aspects: called precision, the first is how many users in the set are also in the AD group. Recall is a second property; it is the number of users in the AD group also contained in the user set. The best AD group (or groups) becomes the folder label. The label provides security teams with more context about the nature of the files pertaining to an incident.

Examples Taken From Customer Data

Based on the dynamic peer group analysis algorithm, several interesting incidents were identified in customer environments. For each one, users had valid permissions to access the files and folders.

An engineering manager accessed a sensitive financial file

The first case focuses on an engineering department manager who was part of an engineering peer group comprised of others who work on similar projects. His close peer groups were also comprised of engineering department employees.

The manager attempted to access \\Finance\Contractors\Budget\FY16\Round 1 Submission, a sensitive document stored in a finance folder. The folder is associated with—and regularly used—by two peer groups: one containing finance department employees, the other comprising finance department contractors.

Because the folder was not associated with his peer group nor its close clusters, Imperva was able to identify the engineering manager's inappropriate file access.

A finance employee accessed an HR file

Another case involves an employee in finance. Clustered in a peer group with six others working on a specific project, he attempted access to personal data stored in an HR folder regarding another employee. But the folder is not associated with the user's cluster, nor with those close by. Rather, it is associated with a peer group containing HR personnel from a different location within the organization.

A researcher accessed a software classification file not related to his work

A researcher was clustered into a peer group with twelve others, in addition to R&D employees. He attempted to open a software specification document that is regularly accessed by a specific R&D team. But the folder containing the file was not associated with his peer group nor its close clusters. This is the type of incident for which we can alert the SOC team.