



OPSWAT.

FILE SECURITY PLAYBOOK

# Mastering File Upload Security with OWASP

How to Design Secure File Upload Applications with  
OWASP Principles and OPSWAT Threat Prevention



# Table of Contents

Executive Summary	04
Why Secure File Uploads Matter	05
Pillars of OWASP & OPSWAT File Upload Security	06
How OPSWAT Helps You Meet OWASP Guidelines	10
What This Means for Your Organization: A Framework for Action	20

# Executive Summary

File upload functionality is a common entry point for attackers targeting web applications. To address file-based risks, the OWASP Foundation has published a set of guidelines that define secure file upload practices. These recommendations cover everything from file type validation and size restrictions to user authorization and storage controls.

This File Security Playbook maps OWASP best practices to OPSWAT’s MetaDefender® platform. We demonstrate how our multi-layered file security technologies can help organizations prevent risks, align with compliance requirements, and strengthen the security posture of your file upload applications.

Security architects, DevSecOps teams, and compliance professionals can use this guide to strengthen file upload workflows and reduce the risk of malicious content entering enterprise systems.



# Why Secure File Uploads Matter

The OWASP Foundation, a globally recognized authority on software security, has provided clear and actionable best practices for secure file upload implementations. These guidelines help organizations:

-  Prevent common attack vectors like malware injection, directory traversal, and file overwrites
-  Reduce risk from evasive techniques, including file spoofing and extension manipulation
-  Strengthen compliance with security frameworks and regulatory standards
-  Build security into development workflows with clear technical principles

Adopting OWASP’s recommendations is not just about compliance. It’s about proactively defending applications, users, and data from threats across multiple attack vectors. By following these best practices, security teams can gain visibility, enforce stronger controls, and protect files and data wherever they are.

## OWASP Principles

1. List allowed extensions. Only allow safe and critical extensions for business functionality.
  - Ensure that input validation is applied before validating the extensions.
2. Validate the file type. Don't trust the Content-Type header as it can be spoofed.
3. Change the file name to something generated by the application.
4. Set a file name length limit. Restrict the allowed characters if possible.
5. Set a file size limit.
6. Only allow authorized users to upload files.
7. Store files on a different server. If that's not possible, store them outside of the webroot.
  - If files are publicly accessible, use a handler that gets mapped to file names inside the application [someid → file.ext].
8. Run the file through antivirus engines or a sandbox if available to validate that it doesn't contain malicious data.
9. Run applicable file types [PDF, DOCX, etc.] through a CDR [Content Disarm & Reconstruction] solution.
10. Ensure that any libraries used are securely configured and kept up to date.
11. Protect the file upload from CSRF attacks.

# Pillars of OWASP & OPSWAT File Upload Security

OWASP outlines critical best practices to protect against malicious file uploads, which are among the most common vectors for cyberattacks. However, implementing those recommendations requires more than policies.

OPSWAT MetaDefender for File Security is based on a multi-layered, scalable, and operationally efficient security architecture. This approach helps organizations operationalize OWASP's recommendations by integrating security controls directly into file upload workflows. Below are eight components to consider for secure file uploads:

1. File Extension & File Type

2. File Name & File Name Length

3. File Size

4. Authentication & Authorization
5. File Storage

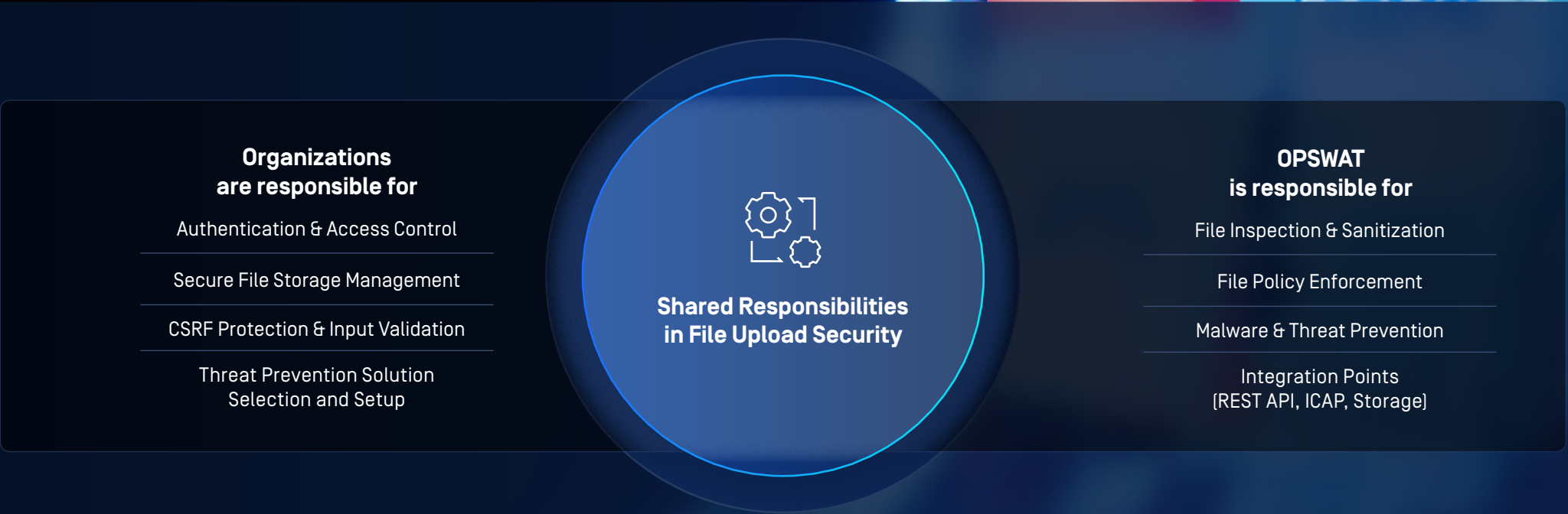
6. Malware Scanning and CDR

7. File Library

8. Protection from CSRF Attacks

## Security is a Shared Responsibility

While the MetaDefender Platform provides powerful technologies for inspecting and securing file uploads, effective protection requires more than just deploying tools. Securing file upload workflows is a shared responsibility, which requires proper integration, access controls, and ongoing monitoring by your internal security teams. Strategic file upload security comes from the collaboration between your infrastructure, your applications, and your file security technology.





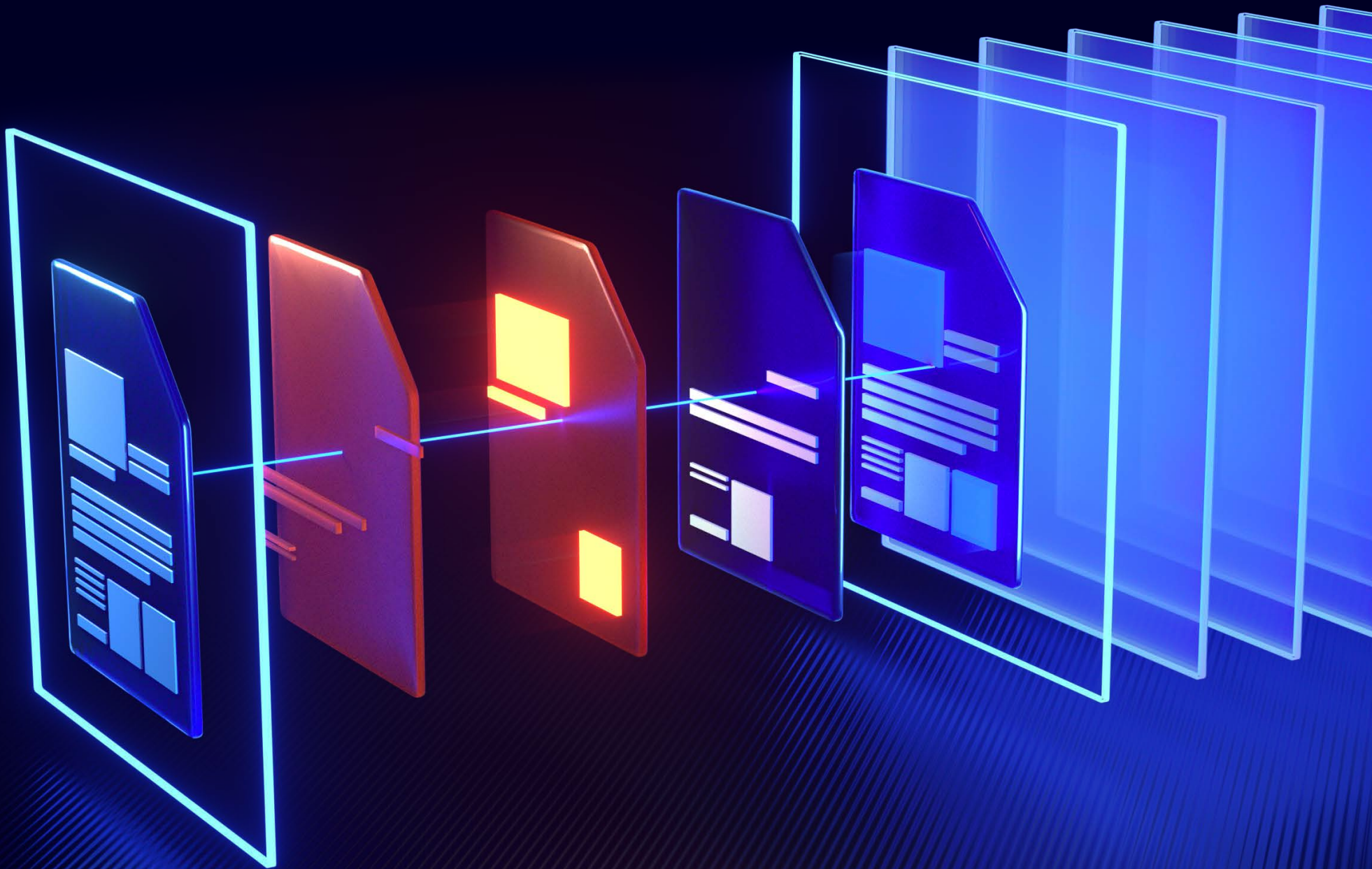
## The Role of Content Disarm and Reconstruction

In its recent File Upload Cheat Sheet publication, OWASP added a new principle to run uploaded files through a CDR solution.


“Content Disarm and Reconstruction [CDR] is a zero-trust file methodology that proactively extracts threat attack vectors from documents and media files.

IEEE

OPSWAT Deep CDR™ prevents both known and unknown file-based threats by extracting and regenerating files in milliseconds. It disarms embedded threats such as scripts, macros, and out-of-policy content, then delivers safe, fully usable files. With support for 200+ file types, customizable policies, and seamless integration across security infrastructure, Deep CDR enables secure content flows without disrupting productivity. Deep CDR is the first CDR technology to achieve a 100% protection and accuracy score from SE Labs.




# How OPSWAT Helps You Meet OWASP Guidelines

OWASP File Upload Best Practice	Why It Matters	How OPSWAT Helps	Technology Behind-the-Scenes
1. File Extension & File Type			
List allowed extensions. Only allow safe and critical extensions for business functionality.  Ensure that input validation is applied before validating extensions.	Attackers can disguise malicious files using deceptive extensions (e.g., invoice.jpg.exe). If the server only checks the extension without validating the actual file type, harmful files can bypass security controls and overwrite or execute malware on the server.  This can result in server compromise, data loss, or ransomware attacks.	OPSWAT enforces strict file extension validation and examines the true extensions to prevent spoofed or mislabeled files. <ul style="list-style-type: none"><li>• Uses AI to validate both the declared file extension and the actual file type</li><li>• Detects and flags mismatches between extension and true content type. Returns the true file type</li></ul>	<div>✔ File Type Detection</div> <div>✔ File Processing Workflow</div> <div>Use Cases<ul style="list-style-type: none"><li>• REST API (MetaDefender Core™)</li><li>• ICAP (MetaDefender ICAP Server™)</li><li>• Storage (MetaDefender Storage Security™)</li><li>• Cloud (MetaDefender Cloud™)</li></ul></div>
Validate the file type. Don't trust the Content-Type header as it can be spoofed.	File type validation should occur after decoding the file name, and must include proper filtering to prevent known bypass techniques, such as: <ul style="list-style-type: none"><li>1. <b>Double extensions</b> (e.g. .jpg, .php), where it easily circumvents the regex \.jpg.</li><li>1. <b>Null byte injection</b> (e.g. .php%00.jpg), where the extension .jpg gets truncated and .php becomes the new extension.</li><li>2. <b>Generic bad regex</b> that is not properly tested and well-reviewed, which can be bypassed due to logic gaps.</li></ul>	<ul style="list-style-type: none"><li>• Supports MIME-type detection and validation (checking the file name against the MIME information)</li><li>• Enables blocklists and allowlists based on true file type and extension.</li><li>• Policy enforcement is managed by user-defined file processing workflow and configurations</li></ul>	
<div> OPSWAT Recommended Practice</div> <div>Combine the Detection of File Extension and File Type</div>		OPSWAT treats file extension and file type validation as a unified process. This ensures that files are not only labeled correctly but also behave as expected when opened.	



2. File Name & File Name Length			
Change the file name to something generated by the application.	File names can pose risks when they contain dangerous characters or reserved names. If improperly handled, they may lead to file overwrites, path traversal, or injection attacks (e.g., XSS, CSRF). When original file names must be retained, robust input validation is critical.	OPSWAT recommends generating file names using unique identifiers and linking them to the database. However, file name handling is entirely managed by the customer's application. <ul style="list-style-type: none"><li>Customers are responsible for renaming files or validating input</li><li>OPSWAT recommends applying OWASP validation rules to prevent front-end and back-end security risks</li></ul>	Customer's Responsibility
Set a file name length limit. Restrict the allowed characters if possible.	Each system has a limit on file name length. Excessively long or poorly validated names may trigger errors, cause system crashes, or be exploited for evasion. Limiting length and restricting characters minimizes security and performance risks.	File name handling, including name length and character restrictions, is fully managed by the customer's application. <ul style="list-style-type: none"><li>OPSWAT allows flexibility in naming but recommends applying validation and maximum length restrictions</li><li>OPSWAT encourages customers to restrict names to safe characters (e.g., alphanumeric, hyphens, periods) and set reasonable limits based on their system</li></ul>	Customer's Responsibility
<div><div></div><div><div>OPSWAT Recommended Practice</div><div>Enforce Strict Policies for Both File Name and File Name Length</div></div><div>OPSWAT recommends using a unique identifier as the file name, ideally between 25–35 characters, although up to 256 characters is supported. Following industry best practices (e.g., <a href="#">Records Express Guidelines</a>) helps reduce file name-related risks.</div></div>			
3. File Size			
Set a file size limit.	Unrestricted file sizes can consume excessive system resources, overwhelm memory, and fill up storage, potentially crashing applications or disrupting services.  Large files may also bypass upload restrictions (e.g., compressed archives) and introduce hidden threats.  Controlling file size protects performance, stability, and infrastructure integrity.	MetaDefender Core allows customers to define file size limits to prevent resource exhaustion and service disruption. <ul style="list-style-type: none"><li>Blocklist rules can reject files exceeding a defined size threshold</li><li>Prevents bypass techniques such as compressed archive attacks (e.g., zip bombs)</li><li>Size-based policy configurations in MetaDefender Core and MetaDefender ICAP Server deployments</li></ul> With Archive Extraction, customers can configure the recursion depth, maximum number of extracted files, and total decompressed size, helping prevent attackers from using compressed archives to conceal oversized content, trigger ZIP bomb attacks, or bypass file size restrictions.	<div><div>✔ File Processing Workflow</div><div>✔ Archive Extraction</div></div> <div><b>Use Cases</b><ul style="list-style-type: none"><li>REST API (MetaDefender Core)</li><li>ICAP (MetaDefender ICAP Server)</li><li>Storage (MetaDefender Storage Security)</li></ul></div>

4. Authentication & Authorization			
Only allow authorized users to upload files.	<p>Without proper authentication and authorization, attackers may exploit upload portals to inject malware, access restricted areas, or overwrite critical files.</p> <p>Effectively managing user privileges helps maintain data integrity, prevent unauthorized access, and reduce the risk of security breaches.</p>	<p>OPSWAT does not manage user authentication or access control within the file upload process. Customers are responsible for implementing secure authentication and authorization mechanisms at the application or portal level.</p> <p>MetaDefender can help enforce authorization by limiting file uploads to only authorized clients and IP sources.</p>	Customer's Responsibility
5. File Storage			
Store files on a different server. If that's not possible, store them outside of the webroot.	<p>Storing uploaded files within the public webroot exposes them to potential execution by attackers via direct URL access. Isolating file storage enhances:</p> <ul style="list-style-type: none"><li>• <b>Security:</b> Reduces the risk of unauthorized access or file tampering</li><li>• <b>Performance:</b> Prevents upload-related bottlenecks on the primary server</li><li>• <b>Data organization:</b> Enables clearer access control and better file management at scale</li></ul>	<p>OPSWAT does not control how or where files are stored post-scan. However, once users have determined the file storage location, they can:</p> <ol style="list-style-type: none"><li>1. Integrate MetaDefender Storage Security for secure and policy-enforced storage.<ul style="list-style-type: none"><li>• MetaDefender Storage Security offers real-time and on-demand malware scanning, sanitization, remediation and policy enforcement</li><li>• Works across cloud and on-prem environments to protect data throughout its lifecycle</li><li>• Supports integration into enterprise workflows for seamless, secure storage</li></ul></li><li>2. Integrate MetaDefender via API or ICAP, ensuring all uploaded files are deeply scanned and sanitized before being stored, even if the storage location is within the webroot.*</li></ol> <p><small>* If you store the files inside the webroot, set them to write-only permissions. If read access is required, setting proper controls is a must (e.g. internal IP, authorized user, etc.)</small></p>	<p>Customer's Responsibility</p> <p><b>Supporting Solutions</b></p> <ul style="list-style-type: none"><li>• MetaDefender Core</li><li>• MetaDefender ICAP Server</li><li>• MetaDefender Storage Security</li></ul>



OPSWAT Recommended Practice

Implement Content Inspection at the Application Layer

While OWASP recommends storing files outside the webroot for maximum isolation, OPSWAT takes protection further. MetaDefender's advanced threat prevention technologies [Deep CDR™, MetaScan™ Multiscanning, Proactive DLP™, and more] analyze and sanitize file content before it can reach the application layer.



6. Malware Scanning and CDR			
Run the file through an antivirus or a sandbox if available to validate that it doesn't contain malicious data.	Antivirus detects known threats via signatures. Sandboxing adds an extra layer by analyzing file behavior in an isolated environment. Relying on just one can lead to missed threats or false positives. Using both significantly increases detection accuracy and reduces risk from advanced or unknown malware.	<p>OPSWAT's MetaScan Multiscanning technology combines over 30 anti-malware engines on-premises (and more than 60 in the cloud) to detect threats using a layered approach that includes signature-based, heuristic, and machine learning detection techniques.</p> <p>This dramatically improves detection accuracy, reduces false negatives, and increases resilience against evasive and polymorphic malware.</p> <p>Adaptive Sandbox performs behavior-based analysis to detect advanced threats that bypass static scanning. It identifies Indicators of Compromise (IoCs) such as exploits, network activity, and abnormal system behavior across executables, documents, and scripts. These IOCs are then extracted and made available for threat hunting, integration with SIEM platforms, and enrichment of broader detection and response workflows.</p>	<div><div>✔ Adaptive Sandbox</div><div>✔ MetaScan Multiscanning</div></div> <div>Use Cases</div> <ul style="list-style-type: none"><li>• REST API (MetaDefender Core)</li><li>• ICAP Integration (MetaDefender ICAP Server)</li><li>• Storage Scanning (MetaDefender Storage Security)</li><li>• Cloud Deployment (MetaDefender Cloud)</li></ul>
Run the file through CDR (Content Disarm & Reconstruction) if applicable (e.g., PDF, DOCX, etc.).	<p>Antivirus and sandboxing are not always enough. CDR prevents zero-day threats by sanitizing files and rebuilding files from safe components, removing malicious scripts, even those unknown to threat databases.</p> <p><b>Zero-Day Threats:</b> CDR can neutralize unknown (zero-day) malicious code in files, offering protection against new threats that have not been identified yet.</p> <p><b>File-Based Attacks:</b> CDR actively removes potentially malicious content from files like documents, PDFs, and executables, preventing them from delivering malware when opened.</p> <p><b>Reduced False Positives:</b> By focusing on removing harmful elements rather than detecting specific malware, CDR minimizes the risk of false positives, which can disrupt workflows.</p>	<p>OPSWAT's Deep CDR technology sanitizes files by removing exploitable content such as macros, scripts, and embedded code while preserving usability.</p> <ul style="list-style-type: none"><li>• Neutralizes both known and unknown threats, including zero-day attacks, without relying on signatures</li><li>• Ensures uninterrupted file processing workflows by delivering clean, functional files within milliseconds</li><li>• Deep CDR achieved a 100% rating in SE Labs' CDR test.</li></ul>	<div><div>✔ Deep CDR</div></div> <div>Use Cases</div> <ul style="list-style-type: none"><li>• REST API (MetaDefender Core)</li><li>• ICAP Integration (MetaDefender ICAP Server)</li><li>• Storage Scanning (MetaDefender Storage Security)</li><li>• Cloud Deployment (MetaDefender Cloud)</li></ul>



OPSWAT Recommended Practice

Implement Multi-Layered Security Solution

OPSWAT strongly recommends combining multiple solutions to achieve multi-layered defenses for file upload security. This defense-in-depth approach ensures the detection and removal of known, unknown, and embedded threats, and provides comprehensive protection with minimal business disruption.

The MetaDefender Platform consists of 8 distinct technologies that address specific cyberattack vectors: Deep CDR, MetaScan Multiscanning, Adaptive Sandbox, Proactive DLP, Threat Intelligence, File-Based Vulnerability Assessment, Country of Origin, and SBOM (Software Bill of Materials).

7. File Library			
Ensure that any libraries used are securely configured and kept up to date.	Third-party libraries often introduce vulnerabilities if not regularly updated. Attackers commonly exploit outdated components to infiltrate systems. Keeping libraries current with security patches is essential for reducing the attack surface and maintaining application integrity.	<p>OPSWAT helps organizations manage third-party libraries and secure their software supply chain.</p> <ul style="list-style-type: none"><li>• Detects vulnerabilities and malware in open-source and third-party libraries</li><li>• Continuously evaluates software components across the SDLC</li><li>• Provides SBOM (Software Bill of Materials) visibility and highlights affected versions for remediation</li></ul>	<p>✔ SBOM</p> <p><b>Use Cases</b></p> <ul style="list-style-type: none"><li>• Supply Chain Security (MetaDefender Software Supply Chain™)</li></ul>
8. Protection from CSRF Attacks			
Protect the file upload from CSRF (cross-site request forgery) attacks.	CSRF attacks exploit authenticated sessions to perform unauthorized actions without user consent. Attackers can trick users into submitting requests unknowingly, potentially leading to data leaks or unwanted file uploads. CSRF tokens prevent this by validating the legitimacy of each request.	<p>OPSWAT does not directly manage CSRF protection within its products. Customers are responsible for implementing CSRF tokens and securing their application logic against request forgery.</p> <p>However, this protection is typically handled by a Web Application Firewall (WAF). OPSWAT's MetaDefender ICAP Server focuses on inspecting uploaded file content at the network perimeter and can be integrated with WAFs to enhance overall web application security.</p>	<p>Customer's Responsibility</p> <p><b>Supporting Solutions</b></p> <ul style="list-style-type: none"><li>• MetaDefender ICAP Server</li></ul>



# What This Means for Your Organization: A Framework for Action

Aligning with OWASP's file upload security standards isn't just best practice; it's an essential defense strategy. OPSWAT empowers organizations to translate these guidelines into strategic, applicable, and effective protection through multi-layered technologies and flexible deployment options.

## Key Takeaways

**OWASP-Aligned Security**

OPSWAT addresses the security areas of the OWASP file upload principles through technologies like true file type detection, Deep CDR, MetaScan Multiscanning, and policy-based workflows.

**Multi-Layered Defense**

From client upload to back-end storage, OPSWAT ensures end-to-end validation, disarmament, and control, especially against unknown threats.

**Deployment Flexibility**

REST API, ICAP, or storage-based integrations support your DevSecOps, perimeter security, and cloud-native strategies.

## OPSWAT Technologies Align with OWASP Essential Practices

**True File Integrity Validation:** Validates actual file type and flags mismatches between extension, MIME type, and content.

**Advanced Threat Prevention:** Combines Deep CDR, MetaScan Multiscanning, Proactive DLP, Adaptive Sandbox, and other technologies to protect against known and unknown threats.

**Secure File Management:** Enforces file size limits, scans archive contents, integrates with secure storage, and detects software vulnerabilities via SBOM and supply chain security.

**Built for Integration:** Deploys via REST API, ICAP, or storage scanning (on-premises or in the cloud) to fit into your workflows and enforce security without disruption.

## Next Steps

- Evaluate your current upload workflows against OWASP's file upload controls.
- Explore [MetaDefender for File Security](#) solutions to integrate secure inspection at the perimeter, application layer, or storage.
- Talk to an OPSWAT expert to discuss tailored solutions for your file security strategy.

GET STARTED

# Are you ready to put OPSWAT solutions on the front lines of your cybersecurity strategy?

**Talk to one of our experts today.**

Scan the QR code or visit us at:

[opswat.com/get-started](https://opswat.com/get-started)

[sales@opswat.com](mailto:sales@opswat.com)



## OPSWAT.

Protecting the World's Critical Infrastructure

For the last 20 years OPSWAT, a global leader in IT, OT, and ICS critical infrastructure cybersecurity, has continuously evolved an end-to-end solutions platform that gives public and private sector organizations and enterprises spanning Financial Services, Defense, Manufacturing, Energy, Aerospace, and Transportation Systems the critical advantage needed to protect their complex networks from cyberthreats.

Built on a "Trust no file. Trust no device.™" philosophy, OPSWAT solves customers' challenges like hardware scanning to secure the transfer of data, files, and

device access with zero-trust solutions and patented technologies across every level of their infrastructure. OPSWAT is trusted globally by more than 1,800 organizations, governments, and institutions across critical infrastructure to help secure their devices, files, and networks from known and unknown threats, zero-day attacks, and malware, while ensuring compliance with industry and government-driven policies and regulations.

Discover how OPSWAT is protecting the world's critical infrastructure and securing our way of life; visit [www.opswat.com](https://www.opswat.com).