

Virus Scanning (CPP-007)

CPP-Identifier	CPP-007
CPP-Label	Virus Scanning
Author	Mattias Levlin, Johan Kylander
Contributors	Kris Dekeyser
Evaluators	Felix Burger, Maria Benauer, Fen Zhang
Date of edition completed	29.08.2025
Change history	Comments
Version 1.0 - 29.08.2025	Milestone version

1. Description of the CPP

Information packages are virus checked, with appropriate facilities for quarantine.

Inputs and outputs

Input(s)	
Data	<i>Information package(s)</i>
Documentation / guidance	Malware signature database(s)
	Guidelines for managing detected threats
Output(s)	
Metadata	<i>Provenance metadata</i>
Documentation / guidance	Scan report

Definition and scope

Virus scanning is the process of examining *Files* as proposed for ingestion into an archive for the presence of malicious software (i.e. malware) such as viruses, trojans, worms, spyware, and ransomware etc. The primary goal of virus scanning is to detect and prevent such harmful code from entering the digital archive to safeguard the integrity and trustworthiness of the preserved content. This security measure protects not only the archival system itself, but also users or other connected systems that access or receive content from the archive. Effective virus scanning is a core step of the ingest process and an essential strategic component in **Risk Mitigation** (CPP-012), ensuring that the archive remains a secure and reliable repository for digital assets. It is a means to mitigate deliberate, human-made threats to digital preservation.

Virus scanning is first and foremost applied during ingest, acting as a checkpoint before *Files* are fully accepted and integrated into a TDA's holdings. In addition, it may be triggered during preservation (archival storage) or dissemination. This may be done either to ensure that no viruses have infected the content after ingest, or to make sure that disseminated content has been checked using up-to-date signature databases.

A TDA must employ virus scanning tools and malware signature databases to ensure effective and up-to-date threat detection. This includes a process to maintain and frequently update a malware signature database that is used by the virus scanning tools. The system must provide secure workspaces and guidelines for managing detected threats, which typically involves isolating suspicious *Files* in a staging or quarantine area to prevent potential harm to the storage. The TDA can reject and remove the contaminated *Files* or *Information packages* during ingest in cases where decontamination is not a viable option.

All scanning activities, detected threats, and subsequent actions (i.e. quarantine, rejection, deletion) must be documented as part of the ingest record and preservation actions as *Provenance metadata*. This documentation contributes to the audit trail of the ingested *Files* and should be incorporated into broader **Object Management Reporting** (CPP-013).

Process description

Trigger event(s)

Trigger event	CPP-identifier
Periodic quality check of <i>Files</i>	CPP-019 (Data Quality Assessment)
Pre-access check of DIPs	CPP-025 (Enabling Access)
Ingest	CPP-029 (Ingest)

Step-by-step description

No	Supplier	Input	Steps	Output	Customer
1	CPP-029 (Ingest)	<i>SIP(s)</i>	Receive and Stage Content: Content arrives and is placed in a temporary, isolated staging area designated for pre-ingest checks	<i>SIP(s)</i> in the staging area	
2	CPP-019 (Data quality assessment) CPP-025 (Enabling Access)	<i>AIP(s)</i>	Select <i>AIP(s)</i> for virus scan and copy their <i>Files</i> to the staging area for checks	<i>AIP(s)</i> in the staging area	
3		<i>File(s)</i> in staging area	Perform Scan: Initiate a comprehensive scan of all <i>Files</i> within the staged <i>Information packages</i>	Scan report/log (indicating clean <i>Files</i> , and any detected threats with file paths and	

		Configured virus scanners		malware names)	
4		Scan report/log	Evaluate Scan Results	All <i>Files</i> reported as clean (step 7)	
				Any <i>Files</i> reported as infected or suspicious (step 5)	
5		Infected/suspicious file(s)	Handling infected or suspicious <i>Files</i> , the TDA conducts a first analysis	Decontamination recommended: Move the identified <i>File(s)</i> to a secure quarantine area, isolated from other systems and content for further analysis and potential disinfection, and inform dedicated staff members (step 6)	CPP-013 (Object Management Reporting)
		Scan report		Rejection recommended: Mark the <i>File(s)</i> (or the entire <i>SIP(s)</i> (as in case of ingest) for rejection. Notify the producer with reasons, if appropriate and/or defined by policy (end of the process)	
		Guidelines for managing detected threats			
6		Quarantined file(s)	In-Depth Analysis: The personnel analyses the threat, leading to multiple potential outcomes	False positive identified: the detected malware does not pose a threat. Whitelist the threat and	

				move the <i>Files</i> from quarantine back to the staging (loop back to step 1)	
		Notification to staff		Decontamination required and possible: The TDA disinfects the <i>Files</i> and moves the <i>Files</i> from quarantine back to the staging (loop back to step 1)	
				Decontamination required but not possible: Notify stakeholders: The TDA notifies the stakeholders that their content is at risk and that it most likely must be deleted and re-submitted. The <i>Files</i> are not moved away from the quarantine area. This is a more likely outcome for AIP <i>Files</i> that are scanned during the preservation (triggered by CPP-019). (step 7)	

7		Scan report	Record the virus scan and its <i>Outcome as a Preservation Event</i> This documentation should include: <ul style="list-style-type: none"> • Datetime of scan • Scanner software name • Virus definition file version/date • <i>Files</i> scanned • Outcome for each file (e.g., 'clean', 'infected - [virus_name]', 'quarantined', 'rejected'). • Any additional actions taken 	<i>Provenance metadata</i>	CPP-016 (Metadata Ingest and Management)
		Actions taken (quarantine, disinfection, rejection)			CPP-013 (Object Management Reporting)
8		Clean <i>File(s)</i>	Proceed with Clean Content or finalise Rejection: <ul style="list-style-type: none"> • If content is clean: release <i>Files</i> from the staging area or proceed with ingest • If any relevant content was rejected: Finalise the rejection process and archive the documentation 	Clean <i>File(s)</i> passed to the next ingest stage, or rejection process completed	CPP-013 (Object Management Reporting)
		Documentation of scan event			

Rationale(s)¹ and worst case(s)

Rational	Impact of inaction or failure of the process
Detection of malware in <i>SIP(s)</i>	Ingest of contaminated <i>Files</i> , risking destruction of the entire TDA.
Process to handle and potentially reject and delete infected <i>SIP(s)</i>	Ingest of contaminated <i>Files</i> , risking destruction of the entire TDA
Processes to maintain up-to-date malware signature databases and virus scanning tools	Ingest of contaminated <i>Files</i> , risking destruction of the entire TDA
Detection of malware in AIP(s)	Risking destruction of the entire TDA.

2. Dependencies and relationships with other CPPs

Dependencies

CPP-ID	CPP-Title	Relationship description
CPP-012	Risk Mitigation	Virus scanning is a direct risk mitigation activity against threats to content integrity and system security triggered by CPP-012.

Other relations

Relation	CPP-ID	CPP-Title	Relationship description
Required By	CPP-013	Object Management Reporting	Reports on virus scanning activities, frequency of threats, and outcomes of the actions provide essential input for operational management and risk assessment.
Required by	CPP-019	Data Quality Assessment	Virus scanning is performed as a step in the overall Data Quality Assessment process.

¹ Term derived from PREMIS.

Required by	CPP-029	Ingest	Virus scanning is one of the core processes that must be performed during ingest.
Affinity with	CPP-003	Integrity Checking	Both processes aim to ensure the "health" of <i>Files</i> . However, Integrity Checking focuses on detecting technical corruption of <i>Files</i> (e.g. bit rot), whereas virus scanning looks to mitigate human-made risks (e.g. malicious code).
Not to be confused with	CPP-004	Data Corruption Management	If a <i>File</i> is detected as infected and cannot be cleaned, it might be considered "damaged." However, CPP-004 typically applies to technical corruption or loss, rather than deliberately human-made damage such as malware-infected <i>Files</i> . In practice, infected <i>Files</i> are more likely to be replaced (by the producer) or rejected.
Not to be confused with	CPP-010	Format Validation	Both processes scan the <i>Files</i> to ensure that they are suitable for preservation. File Format Validation checks if a file conforms to its purported format specification (e.g. is this a valid PDF/A file?) while Virus Scanning checks for malware, regardless of format validity.

3. Links to frameworks

Certification

Certification framework	Term used in framework to refer to the CPP	Section
CTS Link	/	/
Nestor Seal Link	/	/
ISO 16363 Link	/	/

Other frameworks and reference documents

Reference Document	Term used in framework to refer to the process	Section
OAIS Link	Quality Assurance (within Ingest), Security	4.2.3.3 4.3.4
PREMIS Link	<i>Event</i> (with eventType 'virus check', <i>Agent</i> (the scanning software))	<i>Event</i> Entity <i>Agent</i> Entity <i>eventType</i> Controlled vocabulary (2.2)

4. Reference implementations

Example use case(s)

Virus Scan as Part of Ingest at CSC

Institutional Background	
Institution	CSC – IT Center for Science Ltd.,Finland
Hyperlink	https://www.clamav.net/
Description	
Trigger event	Ingest
Problem statement	<i>Files</i> must be scanned for viruses as part of the ingest pipeline to protect the TDA from viruses
Proposed solution	Python script to detect viruses using ClamAv virus scanner <pre>def check_virus(path): """Scan files in directory with ClamAV virus scanner. ...</pre>

Publicly available documentation

Institution	Organisation type	Language	Hyperlink
TIB – Leibniz Information Centre for Science and Technology and University Library, Germany	National library	English	https://wiki.tib.eu/confluence/spaces/lza/pages/93608618/Ingest
	Non-commercial digital preservation service		
	Research infrastructure		
	Research performing organisation		
CSC – IT Center for Science Ltd., Finland	Non-commercial digital preservation service	English	https://digitalpreservation.fi/en/services/quality_reports/2024 (Monitoring of the Digital Preservation Services: “Up-to-date status of the virus check database”)
Archivematica	Digital preservation system	English	https://www.archivematica.org/en/docs/archivematica-1.14/user-manual/transfer/scan-for-viruses/#scan-for-viruses
DANS (Data Archiving and Networked Services), Netherlands	Commercial digital preservation service	English	https://www.coretrustseal.org/wp-content/uploads/2018/04/DANS-Electronic-Archiving-SYstem-EASY-.pdf (“Virus-scans are performed periodically for ingest by the web interface and standard for all other ingest ways (like SWORD).”)
	Discipline-specific data repository		
	Discipline-agnostic data repository		