Security Verification Toolkit

DFARS 252.204-7012 Capability Overview

Version 2.7.1 — February 2026 — Pure Bash — No Build System Required

Purpose

Automated security scanning toolkit that verifies software projects against **NIST SP 800-171** (Protecting CUI), **NIST SP 800-53**, and **FIPS 199/200**. Produces checksummed, traceable evidence with file:line references for audit and assessment packages.

Capabilities Mapped to NIST 800-171 Control Families

800-171 Family	Script	What It Does	800-53
3.4 — Configuration Management			
	`collect-host-inventory.sh`	Collects OS, hardware, network, and software inventory (CUI-protected)	CM-8
	`check-host-security.sh`	Verifies OS config: encryption, firewall, SIP, auto-updates	CM-6
	`check-power-settings.sh`	Validates screen lock, sleep, and idle timeout settings	CM-6
	`harden-system.sh`	Applies CIS-aligned hardening to improve Lynis audit score	CM-6
3.11 — Risk Assessment			
	`scan-vulnerabilities.sh`	Nmap network scan + Lynis system audit with consolidated report	RA-5
	`check-nvd-cves.sh`	Cross-references installed packages against NVD for known CVEs	RA-5
	`check-kev.sh`	Flags vulnerabilities in CISA Known Exploited Vulnerabilities catalog	RA-5
	`check-containers.sh`	Scans Docker/Podman containers for CVEs against NVD and KEV	RA-5
3.13 — System & Communications Protection			
	`check-mac-addresses.sh`	Detects MAC addresses in source code (CUI network identifiers)	SC-8
3.14 — System & Information Integrity			
	`check-pii.sh`	Scans for SSNs, credit cards, phone numbers, IPs, emails	SI-12
	`check-secrets.sh`	Detects API keys, passwords, tokens, private keys (35+ patterns)	SA-11
	`check-malware.sh`	ClamAV malware scan with file hash generation	SI-3
3.8 — Media Protection			
	`secure-delete.sh`	NIST SP 800-88 “Clear” method secure file deletion	MP-6
	`purge-git-history.sh`	Removes sensitive files from entire git history (dry-run default)	MP-6

Orchestration & Evidence Generation

Script	What It Does
`run-all-scans.sh`	Master orchestrator — runs all scans with one command; supports `-n` non-interactive mode for CI/CD
`tui.sh`	Interactive menu interface for selecting and running individual scans
`generate-scan-attestation.sh`	Produces PDF attestation with checksums, toolkit version, and commit hash
`generate-compliance.sh`	Generates compliance statement PDF from scan results
`create-airgap-bundle.sh`	Packages ClamAV + virus definitions for disconnected / air-gapped networks

Key Facts

- **No data exfiltration** — all processing is local
- **Air-gap ready** — bundled threat intel, offline ClamAV
- **Cross-platform** — macOS, Linux, Windows (PowerShell)
- **CI/CD integration** — GitHub Actions workflow included
- **Audit trail** — JSON Lines logging, SHA-256 checksums
- **MIT License** — no procurement or licensing overhead

Example Output (Redacted)

Click any link below to view redacted sample output on GitHub:

- PII Scan Output
- Secrets Scan Output
- Malware Scan Output
- MAC Address Scan Output
- Host Security Scan Output
- Host Inventory Output
- NVD CVE Scan Output
- Vulnerability Scan Report
- Scan Attestation PDF
- Checksum Manifest