NESSviewer

WHITEPAPER · OCTOBER 2026

From Nessus scan data to RMF package artifacts

How NESSviewer turns .nessus files into what an RMF package actually needs, from a scan you can trust to finding history, the software list, the baseline, and an eMASS POA&M update, without rebuilding the same spreadsheet every month.

STIG Solution LLC · stigsolution.com/nessviewer/

AT A GLANCE

What it is	A Windows application that reads Nessus and ACAS scan files and produces package artifacts from them
Reads	.nessus files and ZIP archives of them, the hardware baseline and test plan, the eMASS POA&M export, and remediation plans
Produces	Executive summaries, analysis views, finding history and remediation metrics, software and ports lists, the Detailed Vulnerability List, report packages in Excel and PDF, and POA&M updates for eMASS
Builds	The beta (.NET MAUI, with the baseline module) on Windows 10 1809 or later, Windows 11, or Windows Server 2019 or later, 64-bit. The current release on Windows 7 to 11 and Server 2012 to 2022, with .NET Framework 4.6 or higher.
Your data	Processed on your machine. The only outbound connection is the license check over port 443.
Licensing	14-day free trial with no license key needed to start, then 365 days from date of purchase

Summary

Nessus finds the problems. Getting from a folder of .nessus files to the artifacts an RMF package needs is still, for most teams, a spreadsheet job: export to CSV, pivot, compare against last quarter, reconcile the assets that appear under two names, hand-build the software list, and copy findings into the POA&M template. Then do it again next month.

That process works, and it is also why nobody can hand the task to anyone else, and why the software list stops matching reality about a week after it is written.

NESSviewer is a package tool that ingests scan data. It reads every scan at once with each machine's identity held consistent, tells you which hosts the scan could not really see, builds the software and port inventories that a spreadsheet cannot extract, checks the scans against your baseline, and produces the reports and the eMASS POA&M update from the findings. It keeps a history of every scan in the system's baseline, so it can say when each finding first appeared, when it was fixed, and whether the fix held, and it ranks findings by how likely they are to be exploited. This paper describes how.

The problem: the spreadsheet works until the package is due

A raw .nessus export is XML. Opened in a spreadsheet it is thousands of rows with no context, and the questions an ISSO actually has to answer are not in any single column:

- Which hosts did not scan properly, so their "clean" result means nothing?
- Which findings are old enough that they belong on the POA&M, and which are already on it?
- Which patch would fix the most machines if it went first?
- Is the endpoint agent actually installed on every server, or just most of them?
- What software and which open ports did the scans find that are not on the approved lists?
- When did this finding first appear on this host, and how long are fixes actually taking?
- Of hundreds of findings, which ones is anyone actually exploiting?

Some of this is not just slow in Excel. It is impossible:

- **Installed software is reported inside plugin text, not in a column.** A spreadsheet cannot read a paragraph.
- **Port enumeration alone** tells you something is listening on 8443. It does not tell you what.
- **The same machine appears under different names and addresses** across scans, and two different machines can share a private IP address in different enclaves.

Know what the scan actually saw

A host that fails to authenticate produces almost no findings, which looks exactly like a clean machine. NESSviewer tells the two apart.

Scan Status

Built on the ACAS best practices guide, Scan Status gives every host a verdict from the authentication and local-check plugins that fired: Good, Error, Bad, or Suspect. Credentialed and non-credentialed hosts are counted separately, and a troubleshooting panel shows why a host failed.

One machine, counted once

NESSviewer identifies each machine from what the scan itself reports about it: the AWS instance ID, BIOS UUID, Active Directory computer account, McAfee ePO GUID, fully qualified domain name, MAC address, and hostname. Two scans of the same machine are recognized as one machine even when its IP address changed between them, and two machines that share an IP address are kept apart.

- Tenable's host TAG is deliberately not used, because a cloned machine carries the same TAG as the machine it was cloned from.
- A machine scanned more than once is shown once, with its best scan, and each scan's status is judged from that scan alone, so a good rescan never hides a failed one.
- Host counts, top-host lists, asset totals, and port views count machines, not IP addresses.
- Duplicate scan files (a ZIP and its extracted contents, or a copy in a subfolder) are recognized by content and loaded once.

Presence Check

Is the endpoint agent on all 340 servers, or 337? Is prohibited remote-access software anywhere in the enclave? Presence Check answers both kinds of question, by plugin ID or software name, scoped to comparable machines. It separates "we looked and it isn't there" from "we couldn't look," so an unscanned host never counts as a pass.

Every finding from every scan, in one grid

The Analyze page pulls every finding from every loaded scan into a single grid, joined to your POA&M and remediation plan: one row per host, plugin, and port, with dozens of columns from host identity through CVSS and EPSS scoring to POA&M status.

The grid is hierarchical. Group by any column and nest the groups, so thousands of rows become severity, then plugin, then the hosts each one affects. Sort, filter, and search everything at once, open a details panel to read a finding properly, and keep as many saved views as needed: one for triage, another for POA&M work.

POA&M aging rules. With an aging rule set, the POA&M Status column says what the rule means for each finding, such as "Not on POA&M - POA&M required" or "Ongoing - not required yet," so a finding past its threshold and missing from the POA&M is obvious at a glance. It works before any POA&M is loaded, which

helps when building a first POA&M from scan results. An optional rule uses the CVSSv3 score in place of the Nessus severity.

The Dashboard summarizes where things stand: finding totals and aging, exploit likelihood, remediation performance, scan quality, totals by asset, a patch summary showing which fixes would clear the most machines, and IAVM advisories.

Guided walkthroughs. Seven short walkthroughs run on a built-in sample scan, from getting around the application to building a baseline, so a new team member can learn it without touching real data.

Which findings to act on first

A Nessus scan says what is wrong. It does not say whether anyone is exploiting it, what kind of weakness it is, or which security control it breaches. NESSviewer adds all of that to every finding from reference data it carries offline.

Dataset	Answers
EPSS (FIRST)	How likely is this to be exploited in the next 30 days?
CISA KEV	Is this confirmed exploited right now, and what is the BOD 22-01 deadline?
CWE (MITRE)	What kind of weakness is this?
CCI (DISA)	Which NIST SP 800-53 control does this finding violate?

- **EPSS answers the question CVSS doesn't.** CVSS rates how bad an exploit would be; EPSS estimates how likely one is, re-scored daily from real-world data. A CVSS 9.8 at 0.04% is a severe weakness nobody is attacking; a 7.5 at 97% is being attacked today. Each finding carries its EPSS score, percentile, and the date of the scores used, and the Dashboard counts findings at 10% or more as likely exploited, crossed against severity.
- **CISA KEV is the mandatory queue.** Findings on the Known Exploited Vulnerabilities catalog are flagged with their remediation due date and any known ransomware use. Filter on KEV and sort by due date, and that is the externally required work, in order.
- **CCIs connect a scan finding to the control.** NESSviewer maps the CWEs a plugin reports to their CCIs, each with its 800-53 control and DISA definition, so a POA&M entry can cite the control without working CWE to CCI to 800-53 by hand. Group by CCI to see which controls a set of scans is failing.
- **Decision support, not a rule change.** EPSS never changes a finding's severity, its POA&M deadline, or whether a POA&M is required. It is a reason to act sooner, not something the application acts on for you.

All four datasets ship with NESSviewer and are updated by importing the publisher's own file. Nothing is fetched at scan time.

Software and ports, from what the scans found

Software lists from plugin output. NESSviewer builds a per-asset software inventory, with versions where they can be found, from the CPE data in your scans, so new products are recognized without waiting for an update. Linux package names are normalized so the same package groups together across scans.

Ports matched to listeners. Open ports are correlated with other plugin results to show what is listening, not just that something is.

Your approved lists. Keep your own software and ports lists in NESSviewer and reconcile them against the scans, so anything found that is not on an approved list stands out.

The baseline, checked against the scans

The hardware and software list is the artifact everyone maintains by hand and nobody fully trusts. It is stale the moment it is written, and there is usually no practical way to verify it against the systems.

The NESSviewer beta carries the baseline module: the hardware and software list and the test plan. It builds the software list from what the scans actually found. Loaded beside your scans, it tells you whether you scanned everything in your system. Every host is checked against the test plan, so a machine you missed, or one you scanned that was never in scope, stands out. It also shows where a host's information has changed since the baseline was written.

- **Add or update hosts from the scans.** Hosts the scans found can be added to the baseline, and changed details offered as updates. A value you already have is never replaced with a blank.
- **No guessing.** When two scanned machines share a baseline host's IP address, the MAC address and then the hostname decide which one it is. If nothing tells them apart, NESSviewer says so ("Unclear: 2 share IP") rather than copying one machine's details onto the other.
- **Fill blank hostnames from the baseline**, so hosts the scan could not name carry their real names throughout the application.

So when a reviewer asks whether everything in the baseline was tested, the answer traces back to evidence rather than to a spreadsheet somebody last touched a year ago.

The next release of STIGrevolution brings the same module, and the two applications will share one baseline: scan data on one side, checklist data on the other.

Finding history: when it appeared, and whether the fix held

A single scan is a photograph. It carries no memory of what came before, so the question everyone asks, whether the system is getting better, cannot be answered from one scan.

With a baseline open, NESSviewer records every scan you load into it and works out, for each host and finding, when it was first identified, when it was fixed, and which scan proved the fix. Scans can be loaded in any order, each scan file is recorded once however often it is loaded, and machines are followed by identity rather than by IP address.

- **A fix has to be proved.** A finding counts as fixed only when a later scan could have seen it and didn't: the scan must be Good, and if a credentialed scan ever found the finding, the proving scan must be credentialed with local checks. A failed or uncredentialed scan never produces a false "fixed".
- **Fixes that don't hold are caught.** A finding that returns within the grace period (30 days by default) is flagged Intermittent and the fix is cancelled; one that returns later is flagged Recurred. Aging always counts from the first time a finding was identified, so half-fixing it can't make it look new.
- **Remediation performance on the Dashboard:** the share of findings fixed on time against your POA&M deadlines, median time to fix, what is overdue, fixes that didn't stick, a by-severity breakdown with the 90th percentile, the open backlog by how long each finding has been known, and new against fixed by month.
- **Evidence in the report.** The Finding History report lists every finding the history knows, with the date and the name of the scan file that proved each fix.
- **Optional POA&M aging from first identified,** for a system that has only recently come under scan coverage, instead of aging from plugin publication.

The history is kept in the baseline file itself, so it travels with the system's package rather than living in one person's settings.

Reports and the eMASS POA&M

Reports you were building by hand

- **Executive summaries:** finding totals, aging, and patch status, at the push of a button rather than the end of an afternoon.
- **Report Builder:** choose and order the sections, set the scope (severity floor, host filter, whether to include informational results) and the cover details, including a classification banner printed on the cover and on every PDF page, then generate Excel, PDF, or both. Save the setup as a template and next month is one click.
- **Detailed Vulnerability List, hardware, and host information exports.**

- **Analysis views** exported as the report rather than as raw data that still has to be shaped.
- **Finding History:** every finding with its first-identified and remediated dates, and the scan that proved each fix.

POA&M updates for eMASS

NESSviewer reads the eMASS POA&M export alongside the scans, matches POA&M items to findings, and produces the update.

- **The 2026 eMASS POA&M template** is read and written column for column, and the older layout still imports.
- **An update knows which system it belongs to.** It is linked to the system name and eMASS item IDs, and NESSviewer warns when the loaded POA&M belongs to a different system, when none of the items being updated are in it, or when the loaded scans don't contain the update's hosts.
- **The file for eMASS can't be mistaken for a working copy.** Export to eMASS writes the import file; the grid's own Excel export is named and labelled as a working view, not for eMASS.
- **Defaults fill what you choose.** Milestones that already exist in eMASS, and items being closed, are never changed.
- **Date Identified from the evidence.** New items can take their Date Identified from finding history's First Identified, or from the scan date. An existing item's dates are never changed automatically, because eMASS locks some of them and its export doesn't say which.

AI is off unless you turn it on. POA&M fill stays hidden until you connect a provider: OpenAI, Azure OpenAI, Claude, or Ollama running on your own machine. Requests go to that provider under your own account, never through STIG Solution.

Two builds, side by side

	Beta	Current release
Status	Feature-complete, in testing	The shipping application
Baseline module	Yes	No
Platform	Windows 10 1809 or later, Windows 11, Windows Server 2019 or later; 64-bit	Windows 7, 8, 10, 11; Windows Server 2012, 2016, 2019, 2022; .NET Framework 4.6 or higher

Both are available to download, so nobody is pushed onto a beta they did not choose.

Where NESSviewer fits

STIG Solution makes three applications for the same person: the one accountable for the package. STIGreviewer, free, edits and validates STIG checklists. STIGrevolution works checklists in bulk and runs the package around them, including the eMASS POA&M and Assessment Procedure results. NESSviewer does the same job for scan data, and the baseline is what will tie the two paid applications together.

Deployment, data handling, and licensing

- **Your scan data stays on your machine.** NESSviewer does not connect to a scanner. It reads the files you give it and writes the files you ask for.
- **Reference data is offline.** EPSS, CISA KEV, CWE, and CCI data ship with the application and are updated by importing a downloaded file, so risk data works on a disconnected network too.
- **One outbound connection.** The license check, to licensing.stigsolution.com over HTTPS on port 443. The licensing service runs in Microsoft Azure Government. It carries the license key, a hashed computer ID, and the application version, never your data. It runs every 7 days, and the application keeps working for up to 14 days if it cannot reach the service.
- **Disconnected networks.** NESSviewer is supported on disconnected, classified, and air-gapped networks. Contact us and we will set up licensing for your environment.
- **No Tenable software included.** NESSviewer reads the .nessus files your Nessus or ACAS scans produce. It does not include or provide any Tenable software, plugin feed, or subscription.
- **Trial and license.** A 14-day free trial with no license key needed to start. A license runs 365 days from the date of purchase.

Purchase is subject to the STIG Solution Terms of Use and the NESSviewer EULA, at stigsolution.com/terms-of-use/ and stigsolution.com/eula/nessviewer/.

About STIG Solution

STIG Solution LLC builds tools for the people accountable for RMF packages on DoD and federal systems: STIGreviewer, a free replacement for DISA STIG Viewer; STIGrevolution, for working checklists at the scale of a whole package; and NESSviewer, for turning Nessus scan data into package artifacts. We also publish a STIG revisions file every quarter, free, at stigsolution.com/free-resources/.

Questions, a demonstration, or licensing: mail@stigsolution.com. Documentation is at stigsolution.com/docs/.

This paper describes NESSviewer as of October 2026. STIG Solution is not affiliated with or endorsed by DISA or the U.S. Department of Defense. Nessus and ACAS are products of Tenable, Inc.; eMASS is a Department of Defense system. © 2026 STIG Solution LLC.