Increase Efficiency with Automated Auditing of Static Scans with Fortify

How auditing automation utilizing machine learning will save your organization time and money with an average of 97% accuracy
Table of Contents

Executive Summary ............................................................... 1
Introduction ............................................................................. 2
Unlocking the Power of Predictive Machine Learning .................. 5
The Value of Automating Audits ............................................... 7
Conclusion ................................................................................ 8
Executive Summary

Static Application Security Testing (SAST) tools uncover potential weaknesses in applications by scanning source, byte, and binary code. As of October of 2018, Fortify Secure Coding Rulepacks detected an industry-leading 788 unique categories of vulnerabilities across 25 programming languages spanning over 1,007,000 individual APIs. This thorough level of coverage is crucial to assess the true risk applications pose to the organization; however, raw findings are not actionable because they lack valuable contextual information. A human auditor must review raw findings to determine exploitability by considering environmental, mitigation, and business logic, specific to each application. Time spent auditing raw findings has accounted for the bulk of security and development teams’ non-value-added time for the nearly two decades that SAST has been available. With automated auditing technology from Micro Focus® Fortify, the time spent on the auditing process is drastically reduced.

Fortify Audit Assistant predicts the exploitability of raw findings with 97% average accuracy. Fortify customers using Audit Assistant have seen benefits such as a 58% reduction in manual audit times in its first year of limited adoption with internal teams. Automating audits of static application security findings proved its value with verifiable savings of time and effort. Audit Assistant amplifies the SAST return on investment in three major ways:

- Reducing the number of issues needing deep manual examination
- Identifying relevant issues and removing false positives sooner
- Scaling application security with existing resources

Benefits for Security Teams

By reducing the number of issues needing manual examination, auditors can focus their time on deeper dives with fewer findings, utilizing their skillset more effectively. The organization may retain top security talent more easily with interesting investigations and manageable vulnerability counts. Security teams can audit more applications with the same resources, leveraging machine learning to automatically remove uninteresting findings and validate high-confidence findings.

Benefits for Development Teams

The organization may incentivize top talent more easily when remediation efforts become frictionless. Development teams can efficiently focus only on mitigating the most relevant issues, start fixing high-confidence issues immediately after a static security scan is complete, and dramatically reduce the wait time introduced by human auditing of scan results. The organization may attract top development talent more easily when security is a seamless part of the culture rather than a nuisance or gatekeeper.
Introduction

Software vulnerabilities are a serious problem that the software development process is often not controlled to minimize. SAST enables organizations to identify, monitor, and reduce the business risk from an application’s source code. It has been widely recognized as a necessary component of securing the digital enterprise for nearly two decades. SAST tools report potential vulnerabilities as findings by using analysis methods such as taint, structure, or control flow analysis. Experts then audit findings using details specific to their organization, such as the context of the application and deployment. When auditors determine a potential software security vulnerability is "Not an Issue", the time spent on verification is non-value-added time. These audits are time-consuming and traditionally come at significant cost, exposing a fundamental challenge with delivering secure applications; tools and technology do not immediately produce actionable intelligence.

Auditing a software security finding as an "Issue" often includes a proof of concept attack that exploits the vulnerability without the actual harm an attacker could inflict. These vulnerabilities are risks that must be remediated through changes to the code base or mitigated via alternate controls. Findings that are audited "Not an Issue" are costly for human auditors to spend time with, whether determining why these vulnerabilities are unlikely to be exploited, that their exploitation is an acceptable risk, or in the worst-case scenario, that the same finding has previously been found "Not an Issue." Simplified examples of why findings may be determined to be "Not an Issue" include application context, organizational preference, and expertise of the examiner:

<table>
<thead>
<tr>
<th>Category</th>
<th>Audit Decision</th>
<th>Example Audit Notes</th>
</tr>
</thead>
</table>
| Application Context | Not an Issue   | Not an Issue: unreachable code snippet
adminDebug = false;
if(true == adminDebug)
administratorExecute(cmds);
High risk if compiled reachable, complies with risk profile [employee-dev-risk-03]. |

Continued on the next page
<table>
<thead>
<tr>
<th>Category</th>
<th>Audit Decision</th>
<th>Example Audit Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Context</td>
<td>Not an Issue</td>
<td>Voltage Format-Preserving Encryption protects this database</td>
</tr>
<tr>
<td></td>
<td>There are existing mitigations outside of the application.</td>
<td>High risk if the database backend works with data in the clear, complies with risk profile [consumer-commerce-risk-01]</td>
</tr>
<tr>
<td>Organizational</td>
<td>Not an Issue</td>
<td>Not an Issue: scan optimization</td>
</tr>
<tr>
<td>Preference</td>
<td>The scan is not optimized for the application.</td>
<td>1.5 Java Developer Environment scan resolved by 1.8 runtime.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High risk if runtime JDE is not 1.8, complies with risk profile [employee-dev-risk-03]</td>
</tr>
<tr>
<td>Organizational</td>
<td>Not an Issue</td>
<td>Not an Issue: monitored secured logins</td>
</tr>
<tr>
<td>Preference</td>
<td>The organization’s policies allow for limited risk.</td>
<td>Active Directory system is monitored by ArcSight in the Security Operations Center, requires two-factor authentication, and a VPN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High risk if untrusted authentication, complies with risk profile [employee-dev-risk-03]</td>
</tr>
<tr>
<td>Security Expertise</td>
<td>Not an Issue</td>
<td>Not an Issue: trusted serialized objects</td>
</tr>
<tr>
<td></td>
<td>The security practitioner determines the difficulty to exploit is too high.</td>
<td>Only trusted serialized objects are accepted and exploitation results in server fault</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High risk if upstream server compromised and exploitations stabilize, complies with risk profile [consumer-commerce-risk-01]</td>
</tr>
</tbody>
</table>

Table 1. Time spent auditing raw findings example
Each finding is mapped to one or more analysis tags representing why the finding is “Not an Issue,” traditionally through a manual auditing process. With Audit Assistant on-premises or as a service, these labels are automatically mapped to findings within specific confidences, further reducing the non-value-added time teams must undertake. Traditional strategies for uninteresting findings such as tuning the scan process, suppression through viewing templates, or deleting from reports as mandated by compliance or policy are augmented with the ability to automatically audit high-confidence predictions of both uninteresting findings and verified vulnerabilities.

The scope of the problem is best viewed by a simplified application example scan:

- 0.5 hours of scan time produces 1,000 findings
- 40 hours of audit time identifies 500 actionable issues
- 40 hours remediating issues at 5 minutes per issue

With Audit Assistant, labels are automatically mapped to findings within specific confidences, further reducing the non-value-added time teams must undertake.

Figure 1. Problems viewed as an application example
The magnitude of automating audits at scale is compelling against scan times, which are a negligible component of modern SAST. Customers regularly find 50% of the time spent on application security testing to be in auditing. This data suggests several paths for improvement, including limiting the overall number of findings, reducing the number of findings needing audit, and focusing remediation efforts on the most important issues. Audit Assistant uniquely reduces the number of findings in need of audit while determining with confidence which issues to focus mitigation efforts on first. Waiting for human audits has traditionally meant that security issues were only available days or weeks after security scans, causing friction for developers who then had to switch gears to mitigate issues in code they may not have touched ever! Auditing has been a significant bottleneck for mitigation efforts until Fortify brought the first-to-market machine-learning Audit Assistant

Unlocking the Power of Predictive Machine Learning

By leveraging hundreds of thousands of historic audits from Fortify on Demand (FoD) or an organization’s private dataset, Audit Assistant creates predictive insights into prioritizing triage efforts. Audit Assistant is free for all Fortify customers and delivered in an on-premises integration with Fortify Software Security Center (SSC) or as a service through FoD. Early customers found machine learning effective in correctly predicting audited outcomes using only anonymous issue metadata.

No source code ever leaves the scan environment, being reduced to these limited metadata values:
- Vulnerability category
- Severity
- Inputs
- Branches
- Outputs
- Programming language
- File extension
- Detecting analyzer
Without sharing a single piece of source code, Audit Assistant creates an issue fingerprint used to anonymously and securely drive predictions.

Metadata is sufficient to confidently predict an issue while completely anonymizing its originating organization, code, and the specific vulnerability it represents. Audit Assistant policies leverage audit data where known context differences result in different predictions, allowing organizations to leverage these predictions for their individual needs and use cases.

Policies refine predictions to an organization’s specific needs by using custom audit datasets and confidence thresholds that apply more accurate context to predictions. An “external” policy may make predictions aware of mitigative context, allowing the automatic labeling of injection findings as “Not an Issue” within higher confidence thresholds. A separate “internal” policy may make predictions on applications without these mitigative contexts, requiring manual review before applying labels. With baseline scans and prediction policies in place, static application security findings reap the rewards of machine-learning assisted predictions.
Each finding is locally anonymized into issue metrics, metadata-only packets that remain protected in the TLS encrypted communication channel throughout transmission. Transmitting only the anonymous issue metrics allows the organization to scale software security without sending a single line of source code or identifiable information off-premises. Machine learning algorithms use the metadata and prediction policy to return the status of a finding as not an issue, below not an issue threshold, exploitable, below exploitable threshold, and not predicted in addition to a confidence score, ranging from zero (no confidence) to one (fully confident).

In one customer sample, approximately 20% of findings were removed from scope while suggesting efficient sequences to investigate around 50% of findings in a single application.

The Value of Automating Audits

Audit assistant expands the capacity of security teams and improves the efficiency of development teams by predicting on and automatically auditing findings to mapped analysis labels. Teams easily cut audit time in half by eliminating vulnerabilities from the scope of human review. For a single application of moderate complexity, auditing 1000 findings may easily take 40 hours to identify 500 actionable vulnerabilities. A simplification of savings from audit time alone can be quantified as $X \times H \times C = S$. For a company with 200 applications ($X$), an annual savings of 20 hours per application ($H$) at an audit cost of $150 per hour ($C$) comes in at a $600,000 saving ($S$) per year. These savings have strong evidence in the three use cases below, each of which impacts the hours spent auditing ($H$).

<table>
<thead>
<tr>
<th>Number of applications</th>
<th>Hours per application</th>
<th>Audit cost per hour</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>20</td>
<td>$150</td>
<td>$600,000</td>
</tr>
</tbody>
</table>

Figure 3. Savings example in audit time using Fortify Audit Assistant
Use Case 1
Accelerate deployment cycles by automatically applying audit labels when predicted to a high degree of certainty within a known context. A large software company integrated audit assistant predictions to break a build when an issue is predicted as Exploitable with very high confidence, automating deployments with fewer, lower severity risks. This approach allows truly frictionless application security at DevOps speed for the enterprise with a trusted risk profile and the layered security to mitigate lower confidence findings. Ultimately this use case requires finely tuned auditing policies to accept defined risks for each rapid deployment. Interested teams will refer to the “Batch Train” and “Batch Predict” functions found in example projects on the Fortify GitHub for the on-premises API calls involved.

Use Case 2
Save the expense of auditor time by allowing them to focus on a subset of initial findings. A leading worldwide oil and gas company exclaimed “Scan Analytics helps our team greatly reduce time spent in analyzing false positives.” A leading financial services group in Asia reported a 37% reduction in issues requiring manual audit, predicting 3000 issues of 8000 findings, an immediate savings of $37,500. A large software company who beta tested the automated auditing capability unique to Fortify removed 20% of findings from human auditing entirely. Removing issues from human review allows for refined budgeting, improved time-to-market, and focused mitigation efforts.

Use Case 3
Obtain a force multiplier from expert auditors where their knowledge will be leveraged against future findings. Even after “rock star” auditors have advanced in their careers, their expertise provides predictive value. In this case, training audit assistant better tunes future predictions, improving the 97% average accuracy expected out-of-the-box in general or specific policies. When training is provided for the default classifier, all Fortify users reap the rewards, unlike training provided to private classifiers whose access is restricted to the enterprise that created it.

Conclusion
Organizations no longer need to accept noisy scan results as part of SAST, make costly tradeoffs between scan comprehensiveness and time-to-audit, or contend with developer pushback from uninteresting findings. Fortify Audit Assistant leverages machine learning algorithms and millions of anonymous audit decisions across billions of lines of code to make automated audit predictions. This technology amplifies return on investment by reducing the cost to audit findings substantially with uncannily accurate predictions, leveraging machine learning trained with the knowledge of thousands of application security professionals against unaudited findings. Rather than reduce the breadth of issues found through limiting what analyzers report, Audit Assistant distinguishes non-issues from vulnerabilities and automatically labels them. This innovative machine learning approach to big data analytics scales software assurance
to any sized organization without sacrificing scan depth or security integrity, all without sending a single identifying attribute or line of source code outside the scan environment.

Application security must fit seamlessly into the software development lifecycle as organizations accelerate deployments. 74% of FoD static assessments complete in less than an hour and customers are opting for fully automated audits to facilitate security at DevOps speed. Fortify on-premise customers need only obtain an authentication token then flip a configuration switch to begin requesting predictions. This easy implementation allows for simplified and immediate adoption. Additional requirements, such as breaking a build when an issue is predicted as exploitable with 99% confidence, are enabled through SWAGGER APIs for sophisticated software assurance programs.

Contact Fortify to get started and amplify your return on investment with Audit Assistant, free for all on-premise, on demand, and hybrid customers. Customers:

- Meet deliverable and deployment schedules by reducing the time spent auditing
- Focus human auditing efforts by reducing the number of issues needing deep manual examination
- Retain organizational knowledge critical to high-fidelity audits required to make sound business decisions by scaling application security with existing resources
- Reduce the repetitive, time-consuming work of issue auditing by identifying relevant issues earlier
- Lower the friction between business and security objectives by automating audits

Request your Audit Assistant instance today here!

Learn more at https://analytics.fortify.com/request