**Payconiq**

**Vulnerability Management Guidelines Guideline**

**Document Information**

|  |  |
| --- | --- |
| **Document ID** | PQI-SEC-GDL-018 |
| **Classification** | Internal Use |
| **Registered on** |  |
| **Last Modified on** | 28-02-2024 |
| **Business Owner** |  |
| **Author** | Shraddha Tugnawat |
| **Contact** | security@payconiq.com |
| **Version** | 0.2 |
| **Status** | Draft |

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Change | Author |
| 28-02-2024 | 0.1 | Draft | Michael Oki |
| 29-03-2024 | 0.2 | Final Draft | Shraddha Tugnawat |

**Documented changes**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Change | Author |
| 29-03-2024 | 0.2 | Modified overall document |  |
|  |  |  |  |

**Table of Contents**

[1. Overview 2](#_Toc164761272)

[2. Purpose 2](#_Toc164761273)

[3. Scope 2](#_Toc164761274)

[4. Governance 2](#_Toc164761275)

[5. Guidelines 2](#_Toc164761276)

[5.1 Infrastructure Vulnerability Management 3](#_Toc164761277)

[5.2 Application Vulnerability Management 4](#_Toc164761278)

[5 Related Standards, Policies and Processes 6](#_Toc164761279)

# Overview

The Vulnerability Management Guidelines outlines the comprehensive guidance for Payconiq to establish effective strategies and protocols for managing vulnerabilities within Infrastructure and Applications. These guidelines are designed to help Payconiq identify, assess, prioritize, mitigate, and monitor vulnerabilities in a systematic and proactive manner, thereby enhancing the overall security posture.

# Purpose

The purpose of this guideline is to outline the general process for maintaining vulnerability scans of Payconiq’s infrastructure and applications, evaluating the results, and taking necessary actions. The aim is to minimize the exposure of Payconiq’s infrastructure and applications to security risks, protect an asset.

# Scope

The main responsibility of vulnerability management lies with the Security team.

# Governance

* The Security team is responsible for conducting automated vulnerability scans, assessing vulnerabilities, and reporting/coordinating with relevant stakeholders for remediation.
* Application Owners and SRE are responsible for execution, monitoring of the patch process, and promptly addressing vulnerabilities within their respective systems and implementing appropriate remediation measures, as well as maintaining compliance to the schedule.
* The Head of Security is responsible for maintaining oversight over the process and ensuring that all activities defined are being carried out appropriately and as per schedule.

# Guidelines

## Infrastructure Vulnerability Management

1. Scan Setup – Amazon Inspector and DefectDojo

Infrastructure scans are automated processes conducted by Amazon Inspector. All vulnerabilities identified by Amazon Inspector are automatically logged into DefectDojo. The Security team is responsible for monitoring the vulnerability scan process's health and analysing the findings. High level details of the process can be found in [Vulnerability Management Procedure](file:///wiki/spaces/IT/pages/1120931331/Infrastructure+Vulnerability+Management+Procedure)

1. Scan Vulnerability Management guide

The infrastructure scan monitoring process includes three main steps. These are:

1. Verify Amazon Inspector scans cover full scope.
2. Review and confirm finding.
3. Generate tickets for identified vulnerabilities.

1) Verify Amazon Inspector scans cover full scope

* Navigate to AWS and check for the AWS accounts 'Production' and 'Systems' the Amazon Inspector results.
* Review the Assessment Runs section and confirm that last scan ran successfully.
* Review the Assessment Runs section and confirm there are no exclusion for the latest scan. If there are any exclusions, confirm the instances are still up and running and if so, request SRE to re-install inspector agents for these instances.
* Review the Assessment Target section and confirm that all targets scoped in [Vulnerability Management Procedure](file://wiki/spaces/IT/pages/1120931331/Infrastructure+Vulnerability+Management+Procedure) are added as a target. If not, add the missing targets to the target list.
* Manually trigger the scan to confirm that all scoped instances are included in the new scan.

2) Review and confirm finding

* Navigate to DefectDojo, Open Findings and filter for AWS Instance and HIGH and CRITICAL findings (a direct link is available [here](https://dms.payconiq.io/finding/open?title=&date=&cve=&cvssv3=&severity=High&severity=Critical&steps_to_reproduce=&severity_justification=&active=unknown&verified=unknown&false_p=unknown&duplicate=&out_of_scope=unknown&under_review=unknown&review_requested_by=&under_defect_review=unknown&defect_review_requested_by=&is_Mitigated=unknown&last_reviewed_by=&sourcefilepath=&param=&payload=&file_path=&component_name=&component_version=&static_finding=unknown&dynamic_finding=unknown&sonarqube_issue=&unique_id_from_tool=&vuln_id_from_tool=&sast_source_object=&sast_sink_object=&sast_source_line=&sast_source_file_path=&nb_occurences=&last_reviewed=&test__engagement__risk_acceptance=&has_jira_issue=unknown&jira_issue__jira_key=&has_notes=unknown&tags=&test__tags=&test__engagement__tags=&test__engagement__product__tags=&tag=&o=&test__engagement__product__prod_type=11&page_size=150)).
* For each CVE type, manually confirm from external sites (MITRE, Redhat or AWS) the actual severity of the CVE. It is possible that a CVE is marked as a high finding by Amazon Inspector, but in reality has a CVSS score of < 7.0.
* If any finding has a different severity according to these sites, please update the finding severity accordingly.
* For weekly and monthly scans, review the instances and corresponding finding severity according to [Vulnerability Management Procedure](file://wiki/spaces/IT/pages/1120931331/Infrastructure+Vulnerability+Management+Procedure)
  + For weekly scans
    - Confirm if there are any critical findings, if so mark it as true positive and create a JIRA ticket.
    - Confirm if there are any high findings for VPN instances, if so mark it as true positive and create a JIRA ticket.
  + For monthly scans
    - Confirm if there are any HIGH findings, if so mark it as true positive and perform a manual analysis.

Manual Analysis

If a vulnerability is classified as HIGH in the monthly scan, we will require further analysis. If any of the below checks fail, the Security team shall create an JIRA ticket.

* + - * Check if the latest OS image is used in the last rotation.
      * Check if the latest rotation executed successfully.
      * Check if the vulnerable component in the finding is being used in the server (for this, please liaise with SRE).

Deep analysis of the details of the vulnerability for the combination of the Attack Vector, Attack Complexity, Available Exploit Code, Privilege Required, User Interaction, and the impact on CIA. Please note that the vulnerability may be scored HIGH (> 7.0), but based on our infrastructure and business context, it may pose a higher or lower risk.

* For each true positive, request a review from SRE to confirm the relevance and severity of the CVEs to Payconiq.
* For any false positives, mark the CVE as “false positive” to prevent it from being regenerated in the future. Ensure the CVE is a false positive for all related instances and not just the instance in question because this CVE will be ignored by DefectDojo in all future reporting regardless of the instance.
* For all Info, low and medium findings, mark the findings as "Risk Accepted".
* For all high-level findings that are not related to VPN instances (in case of weekly scan), mark findings as "Mitigated". This is done to ensure if the findings persist in the next scan, they will be reopened (unlike false positive cases).
* For any true positive, if SRE can't remediate the issue in a given timeframe, tag the issue with #delayed tag.

3) Generate tickets for identified vulnerabilities

For any true positives noted during review section, a JIRA ticket should be created for future reference. After getting a confirmation from SRE that necessary updates are applied (according to [Infrastructure Vulnerability Management Procedure](file://wiki/spaces/IT/pages/1120931331/Infrastructure+Vulnerability+Management+Procedure)), it is suggested to trigger another manual scan to confirm the findings are closed.

## Application Vulnerability Management

1. Scan Setup – Veracode SAST/SCA scans and DefectDojo

Application scans are automated, integrated in current CI/CD pipelines, and conducted by Veracode. All vulnerabilities identified by Veracode are automatically logged into DefectDojo. The Security team is responsible for monitoring the health of the vulnerability scan process and analysing the findings. High level details of the process can be found in [Infrastructure Vulnerability Management Procedure](file://wiki/spaces/IT/pages/1120931331/Infrastructure+Vulnerability+Management+Procedure)

1. Scan vulnerability management guide

The application scans are categorized into SAST (Static Application Security Testing) and SCA (Software Composition Analysis) scan. Both types of scans are integrated into CI/CD pipelines and automatically conducted by Veracode for any commit in the application's master codebase.

* + - 1. SAST (Static Application Security Testing) Scan
* Verify Veracode SAST Scan Triggers Upon Any Code Commit (*should be a regular random check for all services, particularly for critical services*)
  + This step involves ensuring that the Veracode SAST scans are triggered automatically whenever there is a code commit.
  + Verification of the successful execution of Veracode scans is done through the Veracode platform, which provides insights into scan results and their status.
* Review and confirm findings
  + Navigate to the DefectDojo portal, a platform used for managing security findings and vulnerabilities.
  + Filter the findings based on the service that was previously scanned by Veracode.
  + Assess all the security findings categorized as Critical, High, and Medium.
    - distinguish between false positives (results incorrectly flagged as vulnerabilities) and true positives (legitimate security vulnerabilities).
  + Ensure that all the security findings from Veracode are correctly logged into DefectDojo.
* Generate tickets for identified vulnerabilities
  + Create Jira tickets with the identified vulnerabilities and assign them to the respective service owners.
  + The ticket should provide the complete details of the vulnerabilities, including their severity level, description, and recommended remediations.
  + Regular follow-up with the owners to ensure timely remediation of the vulnerabilities.
    - 1. SCA (Software Composition Analysis) Scan
* Verify Veracode SCA Scan Triggers Upon Any Code Commit (*should be a regular random check for all services, particularly for critical services*)
  + This step involves ensuring that the Veracode SCA scans are triggered automatically whenever there is a code commit.
  + Verification of the successful execution of Veracode scans is done through the Veracode platform, which provides insights into scan results and their status.
* Review and confirm findings
  + Navigate to the DefectDojo portal, a platform used for managing security findings and vulnerabilities.
  + Filter the findings based on the application or service that was recently scanned by Veracode.
  + Assess all the security findings categorized as Critical, High, and Medium.
* Report on a monthly basis via Jira
  + Maintain a list of all SCA vulnerabilities identified within a specific month for application. (*Current reporting is based on month wise, and all the findings are listed in spreadsheet for respective application*)
  + Create Jira tickets with the above identified SCA vulnerabilities and assign them to the respective owners.
  + The ticket should provide the details of the vulnerabilities, including their severity level, description, and recommended remediations.
  + Regular follow-up with the owners to ensure timely remediation of the vulnerabilities.

**Note**: This Regular random verification and findings review ensure that all security findings from SAST and SCA scans are accurately reported into DefectDojo, as well as that the Veracode integration in the pipeline is working as expected.

# Related Standards, Policies and Processes

* [Vulnerability Management Procedure](file:///wiki/spaces/IT/pages/1120931331/Infrastructure+Vulnerability+Management+Procedure)
* [Information Security Policy](https://payconiq.sharepoint.com/sites/Intranet/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2FIntranet%2FShared%20Documents%2FPolicies%2F02a%2E%20Information%20Security%20Policy%20v3%2E7%2Epdf&parent=%2Fsites%2FIntranet%2FShared%20Documents%2FPolicies&p=true&ga=1)
* [Secure Development Policy](https://payconiq.sharepoint.com/:b:/s/Intranet/EZkxHhNOBEREl2nEHkvsvYoBY8sEf3tmCaRtHJduf5s9CA?e=GPVscm)