Agile meets Security - Weekly recording of activities[(0)]

Team name:	GitGuardians
Name team member A:	Karnavat, Aadit
Name team member B:	Farag, Beshoy
Name team member C:	Guardiola, Antonio Huesa
Name team member D:	
Name team member E:	
Link to the backlogs:	https://ahuesag.atlassian.net/jira/software/projects/GI
Link to the documentation:	https://github.com/BeshoyNFarag/AgileDocuments.git
Link to the Gitlab project:	https://lv-gitlab.intern.th-ab.de/agilesec2025/team14/

Sprint 1, Week 1 - Wednesday, 23.4.2025 to Tuesday, 29.4.2025

During this week		team members have contributed 100% of the expe
During this week		team members have contributed >100% of the exp
During this week		team members have contributed <100% of the exp

#	Description of the activity
1	Research how to implement CI/CD on GitLab
2	Set up GitLab CI/CD Runner
3	Install GitLab Runner (target: Ubuntu VM)

4	Research appropriate tech stack for CI/CD + security tools
5	Virtual Machine configuration
6	Install Docker Engine on VM
7	. Create Dockerfiles for containerization
8	Search GitLab Server Access / Authentication
9	

Sprint 1, Week 2 - Wednesday, 30.4.2025 to Tuesday, 6.5.2025

During this week team members have contributed 100% of the expe
During this week team members have contributed >100% of the exp
During this week team members have contributed <100% of the exp

#	Description of the activity
1	Research for Acceptance Criteria
2	Research for Defination of done
3	checking the commands and tools required for the CI/CD pipeline of the todolist app
4	research on ubuntu VM
5	Configure properly the gitlab-runner in the VM
6	Replace base Images with custom Images for future security tools implementation
7	
8	
9	

Sprint 2, Week 1 - Wednesday, 7.5.2025 to Tuesday, 13.5.2025

During this week 3 team members have contributed 100% of the expe
During this week team members have contributed >100% of the exp
During this week 0 team members have contributed <100% of the exp

#	Description of the activity
1	Backlog refinement
2	Acceptance Criteria
3	DoD
4	Story: As a Security Champion, I want to fix 3 selected vulnerabilities in the Juice Shop and document the fixes according to secure coding guidelines.

5	Story: As a Security Champion, I want to research and test if these vulnerabilities can be automatically detected using SAST tools like Semgrep.
6	Story: As a Security Champion, I want to contribute to the Secure Coding Checklist so developers can avoid these vulnerabilities in future.
7	Story: As a DevOps engineer, I want to learn about the chosen container scannning tool for further implementation in our CI/CD pipelines
8	Story: As a DevOps engineer, I want to research which container scanner tool fits better our use case
9	Story: As a DevOps engineer, I want to implement the container scanning tool to our CI/CD pipelines
10	Story: As a DevOps, I want to choose the most appropiate SAST tool based on the projects needs
11	Story: As a DevOps engineer, I want to learn about the chosen SAST tool for further implementation in our CI/CD pipelines

---

Sprint 2, Week 2 - Wednesday, 14.5.2025 to Tuesday, 20.5.2025

During this week	3	team members have contributed 100% of the expe
During this week		team members have contributed >100% of the exp
During this week	0	team members have contributed <100% of the exp

#	Description of the activity
1	Research and learn the chosen scanning tool
2	implement the scanning tool
3	research and learn the fuzzing tool
4	implement the fuzzing tool
5	Implement the SAST tool and get report
6	Documentation
7	Weekly Scrum + Project extra planning
8	Research dependency scanning and learn how to use Dependency check and generate reports
9	Run dependency scan on Todo List app and analyze results
10	Format and prepare the data for dashboard integration

11	Reasearch the DAST

Sprint 3, Week 1 - Wednesday, 21.5.2025 to Tuesday, 27.5.2025

During this week team members have contributed 100% of the expe
During this week team members have contributed >100% of the exp
During this week team members have contributed <100% of the exp

#	Description of the activity
1	completed fuzzing stage in todolist pipeline
2	modified Dockerfile to fit the todolist Project actual requirements
3	modified todolist Project to add the fuzzing function
4	researched publishing the container and package for the todolist
5	Researched OWASP Dependency-Check and evaluated how it fits our use case
6	Documented the entire process and created a Word report for submission
7	research, decide and learn the DAST tool for our to do app
8	Implement the DAST tool for the to do app
9	research, decide and learn the SAST tool for our juice shop
10	implement the SAST tool for juice shop
11	implement dependency check for our juice shop
12	Research, decide and learn the DAST for our juice shop
13	Implement the DAST tool for the JUICE SHOP
14	Team sprint planning and sprint release
15	sprint review and feedback

Sprint 3, Week 2 - Wednesday, 28.5.2025 to Tuesday, 3.6.2025

During this week team members have contributed 100% of the expe
During this week team members have contributed >100% of the exp
During this week team members have contributed <100% of the exp

#	Description of the activity
1	completed fuzzing in todolist and juice shop pipeline
2	completed container publishing in both pipelines
3	documented fully process of fuzzing, container scanning, and container publishing
4	added SAST sonerqube and dast-zap to CI/CD pipeline in todoapp
5	added gitleaks, added semgrep to CI/CD pipelien juiceshop
6	checked container scanning in todolist pipeline
7	modified container scanning in juiceshop pipeline
8	added the deployment stage to both juiceshop and todoapp made sure its running and deployed on the vm
9	modified the dockerfiles to add the correct versión of the dependencias neede for both projects to run
10	completed package publishing in todolist pipeline
11	Implemented and tested Trivy container scanning script on the OWASP Juice Shop Docker image. Uploaded and configured custom scanning scripts (`dependency-check.sh` and `trivy-scan.sh`) in the GitLab repository.
12	Refined the sprint 3 backlog and added acceptance criteria
13	Prepared and organized project directories (`security-scanners/`) to support CI/CD integration.
14	Researched and attempted integration of OWASP Dependency-Check for scanning Juice Shop dependencies.
15	Generated a full implementation guide and documented the scanning process for future use.

Sprint 4, Week 1 - Wednesday, 4.6.2025 to Tuesday, 10.6.2025

During this week		team members have contributed 100% of the expe
During this week		team members have contributed >100% of the exp
During this week		team members have contributed <100% of the exp

#	Description of the activity

1	Cleaned the VM for storage optimization without harming the pipeline or security analysis
2	Documented all the rest of the processes
3	added acceptance criteria for the rest of the springs
4	added credentials in the yml for dockerhub for both pipelines
5	Installed and configured OWASP Dependency-Check locally on VM - 2 hrs
6	Researched and tested multiple ways to run SCA (package scanning):Tried owasp/dependency-check Docker image Identified volume mount and permission issues
7	Switched to more stable Maven plugin and Documented scanning steps and generated a weekly report
8	Documented scanning steps and generated a weekly report
9	Worked on running and verifying the scan on the todoapp- 1/2 hr
10	Started reviewing Agile/Scrum background to strengthen integration knowledge
11	Add AC for the previous springs tasks Complete previous documentation of the creation of the custom images - 1h
12	Research if the output format of the sec tools can changed into json
13	Research a way to obtain gitlab artifacts from an external application for the vulnerability dashboard -
14	Document the the json formatting and the artifacts acquisition-
15	

Sprint 4, Week 2 - Wednesday, 11.6.2025 to Tuesday, 17.6.2025

During this week team members have contributed 100% of the expe
During this week team members have contributed >100% of the exp
During this week team members have contributed <100% of the exp

#	Description of the activity
1	Set up the VD project
2	Implement basic interface
3	Implement fetching data from API function
4	Implement getter and setter of the config file

5	Implement unit tests
6	- Implement mofication of config. in the dashboard interface - 4.0h
7	Document the process
8	Cleaned up the team VM and resolved space issues blocking dependency scans.
9	Debugged and resolved CI/CD pipeline failures related to Docker authentication and image pushes.(1 hr)
10	Fixed pipeline stages for both apps to ensure full end-to-end execution (build → test → package → security → deploy
11	-Set up and verified artifact creation (HTML/JSON reports) for dependency checkers.(1hr)

Sprint 5, Week 1 - Wednesday, 18.6.2025 to Tuesday, 24.6.2025

During this week team members have contributed 100% of the expe
During this week team members have contributed >100% of the exp
During this week team members have contributed <100% of the exp

#	Description of the activity
1	Modify the container scanning pipeline stage to allow artifacts download via API - 6.0 h
2	Modify the fuzzing pipeline stage to allow artifacts download via API - 1.0 h
3	create Fuzzing Page in VD - 6.0 h
4	create Container scanning in VD - 10.0 h
5	Finish the project agile documentation
6	finish first 2 parts in technical documentation
7	change the sast and gitleaks in the pipeline to be able to pull the artifacts from gitlab api
8	add the sast to the vulnerability dashboard
9	add the dast to the vulnerability dashboard

Sprint 5, Week 2 - Wednesday, 25.6.2025 to Tuesday, 1.7.2025

During this week		team members have contributed 100% of the expe
During this week		team members have contributed >100% of the exp
During this week		team members have contributed <100% of the exp

#	Description of the activity
1	finished the techincal and agile report
2	Developed the DAST in the dashboard
3	Developed the Dependency Check in the Dashboard
4	Modified configuration section of VD to be able to fill jobnames and SAST second job's report path - 6.0h
5	Modified each VD section, so that the new configuration settings are actually used - 2.0h
6	Avoid duplication in cont. scanning of the VD - 1.0h
7	Checked pipelines result for acceptance - 2.0h
8	Implemented Dependency Scanning (SCA) for both ToDo App and Juice Shop pipelines using OWASP Dependency-Check, npm audit, and retire.js.

9	Ensured pipelines function correctly and reports are being generated, stored, and fetched accurately.- 1.5 h
	Integrated SCA reports into the Vulnerability Dashboard (JSON formatting, rendering logic via render.js and index.css). 1 h
	Assisted in finalizing technical documentation, particularly for the pipeline tools, report artifacts, and security practices.1 h

IT/boards/34

vulnerabilitydashboard.git

(0) This document must be u
(1) The time expected to be
(2) Enter the duration to the

cted time[1].
ected time.
ected time. What were the reasons for falling below 100%:

respective user story- or task-ID	Duration of the activity[2]
As a DevOps engineer, I	1.0
As a DevOps engineer, I want to configure a GitLab Runner for our projects, so that pipelines can automatically execute build and deploy jobs.	1.5
As a system administrator, I want to install and register GitLab Runner on our Ubuntu VM, so that jobs can be run in a controlled, self-hosted environment.	0.5

As a solution architect, I want to evaluate and choose the right tools (CI/CD, scanners, container engine), so that our pipeline is compatible, scalable, and secure.	0.5
As a system admin, I want to configure a VM with required packages and networking, so that it can host Docker containers and GitLab Runner securely.	1.0
As a DevOps engineer, I want to install Docker on our VM, so that I can use containers to deploy apps and run security scans in isolated environments.	0.5
As a developer, I want to write Dockerfiles for the Todo App and Juice Shop, so that they can be built and deployed in consistent container environments.	0.5
As a DevOps engineer, I want to verify and set up access to the GitLab server, so that our runner can securely connect and pull repositories to execute CI/CD jobs.	1.0
	0.0
	0.0
	0.0

Sum	6.5

---

cted time[1].
ected time.
ected time. What were the reasons for falling below 100%:

respective user story- or task-ID	Duration of the activity[2]
	1.0
	1.0
	1.5
	1.0
	0.5
	1.0
	0.0
	0.0
	0.0
	0.0
	0.0
Sum	6.0

---

cted time[1].
ected time.
ected time. What were the reasons for falling below 100%:

respective user story- or task-ID	Duration of the activity[2]
	3.0
	1.5
	1.5
	2.0

	1.5
	1.0
	0.5
	1.0
	1.0
	1.5
	1.0
Sum	15.5

---

cted time[1].
ected time.
ected time. What were the reasons for falling below 100%:

respective user story- or task-ID	**Duration of the activity[2]**
AG-53	6.0
Ag-53	5.0
AG-56	8.0
AG-56	4.0
AG-42	5.5
	4.0
	2.5
AG-62	5.0
AG-62	5.0
AG-62	3.0

yet to be decided	6.0
Sum	54.0

cted time[1].
ected time.
ected time. What were the reasons for falling below 100%:

respective user story- or task-ID	Duration of the activity[2]
	2.0
	1.0
	2.0
	4.0
	0.5
	1.0
	3.0
	3.0
	2.0
	2.0
	3.0
	2.0
	2.0
	1.0
	1.0
Sum	29.5

cted time[1].
ected time.
ected time. What were the reasons for falling below 100%:

respective user story- or task-ID	Duration of the activity[2]
	6.0
	1.0
	2.0
	6.0
	6.0
	0.5
	4.0
	4.0
	1.0
	5.0
	1.0
	1.0
	1.0
	1.0
	1.0
Sum	40.5

cted time[1].
ected time.
ected time. What were the reasons for falling below 100%:

respective user story- or task-ID	Duration of the activity[2]

	2.5
	2.0
	1.0
	3.0
	2.0
	2.0
	1.0
	0.5
	0.5
	1.0
	1.0
	4.0
	2.0
	2.0
	0.0
Sum	24.5

cted time[1].
ected time.
ected time. What were the reasons for falling below 100%:

respective user story- or task-ID	**Duration of the activity[2]**
	5.0
	2.0
	4.0
	3.0

	5.5
	4.0
	1.0
	1.0
	1.0
	1.0
	1.0
Sum	28.5

---

cted time[1].
ected time.
ected time. What were the reasons for falling below 100%:

respective user story- or task-ID	**Duration of the activity[2]**
	6.0
	1.0
	6.0
	10.0
	2.0
	3.0
	1.5
	5.0
	4.0
	0.0
	0.0
Sum	38.5

---

cted time[1].
ected time.
ected time. What were the reasons for falling below 100%:

respective user story- or task-ID	Duration of the activity[2]
finished the techincal and agile report	6.0
Developed the DAST in the dashboard	4.0
Developed the Dependency Check in the Dashboard	5.0
Modified configuration section of VD to be able to fill jobnames and SAST second job's report path - 6.0h	6.0
Modified each VD section, so that the new configuration settings are actually used - 2.0h	2.0
Avoid duplication in cont. scanning of the VD - 1.0h	1.0
Checked pipelines result for acceptance - 2.0h	2.0
Implemented Dependency Scanning (SCA) for both ToDo App and Juice Shop pipelines using OWASP Dependency-Check, npm audit, and retire.js.	1.0

Ensured pipelines function correctly and reports are being generated, stored, and fetched accurately.- 1.5 h	1.5
Integrated SCA reports into the Vulnerability Dashboard (JSON formatting, rendering logic via render.js and index.css). 1 h	1.0
Assisted in finalizing technical documentation, particularly for the pipeline tools, report artifacts, and security practices.1 h	1.0
Sum	30.5

ıpdated weekly. The current version must be submitted by the Scrum Master in Moodle on Tuesday eve

lack of communication

ɛning.