Sprint 1 Report

Product Name: Intrusion Detection Tool
Team Name: SlugShield
Date: 10/21/25

Actions to Stop Doing

- Stop starting development work before clear task breakdowns are made.
  → During Sprint 1, some exploratory tasks overlapped with implementation and caused merge confusion. The team decided to define task boundaries before starting future development.

- Stop having extended Scrum discussions beyond the 15-minute timebox.
  → Some meetings went too long because technical debugging was discussed. The team will move detailed technical discussions to separate working sessions.

Actions to Start Doing

- Start using GitHub issue tracking to manage user stories and tasks.
  → This will help ensure visibility of progress, assignees, and task completion throughout the sprint.

- Start holding short pair-programming or group debugging sessions.
  → Real-time collaboration helped during setup; scheduling regular working blocks should improve progress and reduce blockers.

- Start documenting setup instructions in the README early in the sprint.
  → Several team members spent time reconfiguring local environments. Early documentation will improve onboarding for future sprints.

Actions to Keep Doing

- Keep maintaining clear communication and consistent attendance during Scrum meetings.
  → The team has been punctual and collaborative, ensuring transparency.

- Keep using time estimates for all tasks and logging total sprint hours.
  → The estimates were close to actual effort, giving a good foundation for velocity tracking.

- Keep dividing work by user story ownership.
  → Assigning each member an initial user story/task provided focus and accountability.

Work Completed

User Stories Completed

User Story ID	Description/Task	Status
US1	Description: As a user, I want the system to run locally so data stays private **Task**: Set up backend environment for project	Complete
US2	As a user, I want the IDS to monitor network traffic in real time. **Task**: ICMP flooding detection	In progress
US3	As a user, I want the IDS to log or print alerts when suspicious activity is detected. **Task**: ARP Spoofing(Man in the Middle)	Complete

User Stories Not Completed

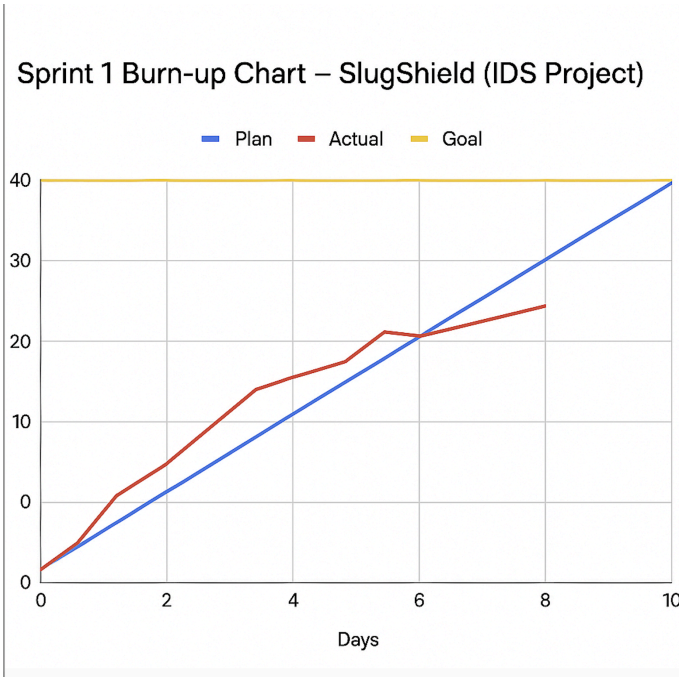
User Story ID	Description	Status
US3	Implement Port Scanning Detection	To Do
US3	Adjust threshold detection as needed	To Do

Work Completion Rate

Metric	Value
Total user stories completed	3 main stories + infrastructure & spikes
Total ideal hours completed	30 hours
Total sprint days	10 days
Average user stories per day	~0.21
Average ideal hours per day	~5.1

Team observation: The team maintained steady progress throughout the sprint, completing the planned workload slightly ahead of schedule.

Burnup chart: The final burnup chart (Sprint 1 – Local IDS Project) is posted in the lab and emailed to the TA.



Summary of Sprint 1

Sprint 1 successfully delivered the foundation of the **local intrusion detection system (IDS)**.
The team completed environment setup, ARP spoofing detection, and made strong progress on **ICMP flooding** and **SSH brute-force detection** modules. Core backend components are now operational, allowing the system to capture live packets and log alerts for suspicious activity.

Although **port scanning** and **threshold adjustment** remain in the *To Do* phase, substantial groundwork was completed for their implementation in the next sprint. The team also began **attack simulation testing**, validating early IDS responses and refining detection accuracy.

Throughout Sprint 1, the team demonstrated consistent collaboration and adaptability.
Clearer sprint boundaries and improved GitHub task tracking have led to more efficient progress management.

Next Steps (Sprint 2 Focus):

- Complete and fine-tune **port scanning and threshold-based detection**.
- Integrate a **real-time dashboard** for visualizing IDS alerts.
- Expand automated testing with more attack simulation scenarios.

Sprint 2 Report

Product Name: Intrusion Detection Tool
Team Name: SlugShield
Date: 11/3/25

Actions to Stop Doing

- The team should stop working alone on their tasks as some members may need additional support.
- Stop with the lack of documentations for files.
    - Some files are super long to read which can be time consuming for the team to fully understand in a given time.

Actions to Start Doing

- Have one in person meeting where it's dedicated to working on tasks, if help is needed then individuals can ask one another for support.

- Start utilizing design files within the directory to give a brief one sentence explanation of the overview for a file.
    - This will greatly help with understanding the purpose of each file and how each individual team member can utilize a file.

Actions to Keep Doing

- Keep maintaining clear communication and consistent attendance during Scrum meetings.
    - The team has been punctual and collaborative, ensuring transparency.

- Keep using time estimates for all tasks and logging total sprint hours.
- Documenting setup instructions in the README early in the sprint.
    - The team was relatively quick in understanding how to run the backend and frontend

Completed Work

#1 "As a user, I want to access the dashboard through my browser locally so there isn't a need for third party softwares"

- [1.1] Set up environment and make the files and directories necessary
- [1.2] Implement a simple frontend(just a blank page for testing)
- [1.3] Connects frontend to backend
- [1.4] Dashboard is reachable(host and port configuration)

#2 "As a user, I want a main overview that shows overall systems health so I can get status at a glance"

- [2.1] Implement a little summary on top of the page showing overall status– OK and ALERT; if nothing is detected then status should be OK, if something is detected then it should change dynamically without any user input
- [2.2] Implement timestamp for last time it was checked
- [2.3] Implement function that shows active alert == 0 if overall system is OK
- [2.4] Implement function that shows what was detected alongside alert == x, where x is the number of times it was detected [2 hours]

#3"As a user, I want to view alerts on a simple dashboard so I understand my system is safe."

- [3.1] If icmp flood is detected, then dashboard should update that icmp was detected
- [3.2] If ssh bruteforce login is detected, then dashboard should update that ssh bruteforce was detected
- [3.3] If arp spoofing is detected, then dashboard should update that arp spoofing was detected(partially completed, finished writing backend code necessary to connect to frontend, just need to configure to frontend)

[4.1] A chart displaying real time traffic metrics for icmp flooding (icmp packets per second)

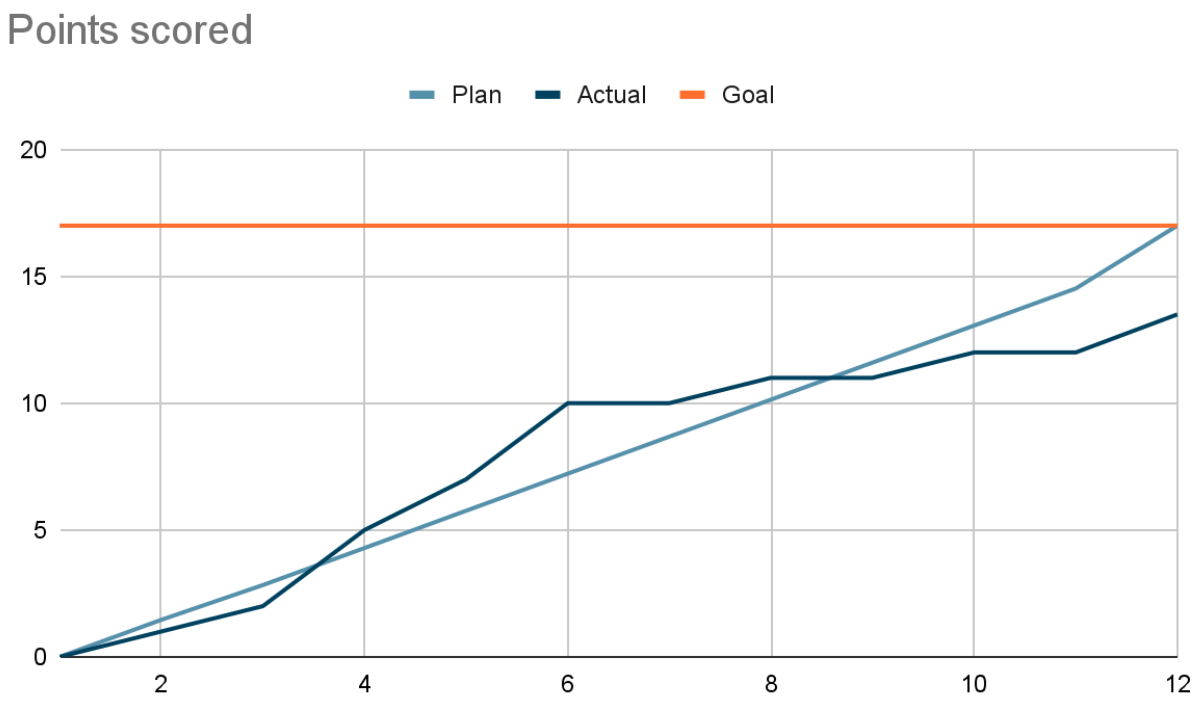
[4.2] A chart displaying real time traffic metrics for ssh brute force login (login attempts per minute)

Work Not Completed

- [2.41] Adding to this, if multiple attacks happened, then it should be a drop down menu of the multiple attacks with the alert

#4 "As a user, I want to see real time alerts that were detected along with simple charts comparing the real time alerts that were detected to normal behavior traffic so I understand what is happening and how severe it is." **(metrics can be adjusted, doesn't have to be per minute or per second– whatever you think will work best)**

- [4.3] A chart displaying real time traffic metrics for arp spoofing(changes per minute)
- [4.4] A baseline curve representing normal behavior for icmp traffic(average icmp packets per second)
- [4.5] A baseline curve representing normal behavior ssh login traffic(average login attempts per minute)
- [4.6] A baseline curve representing normal behavior for MAC mapping over time(average MAC addresses changes per minute)
- [4.7] Charts should be updating live as new packets and such are analyzed



Sprint 3 Report

Product Name: Intrusion Detection Tool
Team Name: SlugShield
Date: 11/18/25

Actions to Stop Doing

- Stop delaying UI integration features were implemented in the backend but not always linked to the dashboard quickly.
- Stop postponing documentation for newer features.
- Stop working without confirming dependencies between tasks

Actions to Start Doing

- Start keeping small notes to file changes to alert thresholds.
- Start meeting more often in person.
- Start testing features earlier in pairs, especially UI parts that rely on the back end

Actions to Keep Doing

- Keep using clear time estimates and logging hours for large user-story items.
- Keep working collaboratively when issues surface. Team members have been proactive about helping each other.
- Keep updating the README when new tools or updates are made, preventing setup issues for newer features

Completed Work

#1. “As a user, I want clear explanations for each alert so I can understand what triggered it and what it means

- [1.1] Added a dropdown menu under Recent alerts.
- [1.2] Wrote clear explanation for ICMP flooding
- [1.3] Wrote clear explanation for ARP spoofing
- [1.4] Wrote clear explanation for port scanning
- [1.5] Wrote clear explanation for SSH brute-force detection

#2 “As a user, I want to adjust the threshold for suspicious alerts so I can manage notification frequency”

- [2.1] Added adjustable panel threshold on dashboard
- [2.2] Implemented threshold control for ICMP flood detector
- [2.3] Implemented threshold control for ARP spoofing detector
- [2.4] Implemented threshold control for port scanning detector
- [2.5] Implemented threshold control for SSH brute force detector

#3 “As a user I want to pause and resume monitoring from the dashboard”

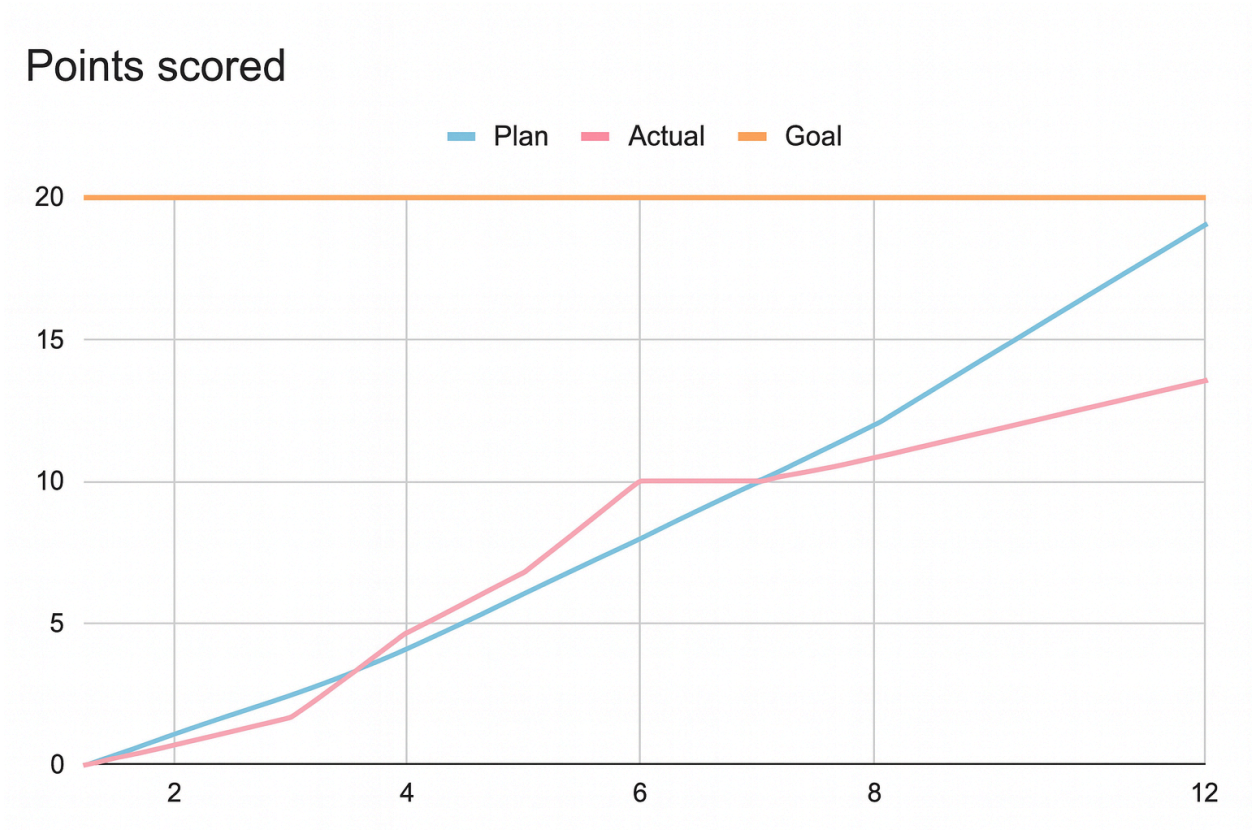
- [3.1] Implemented Pause monitoring
- [3.2] Implemented Resume monitoring

#4 “As a user I want to receive email notifications whenever an alert is detected

- [4.1] Implemented email notifications with summary for ICMP flooding
- [4.2] Implemented email notifications with summary for ARP spoofing
- [4.3] Implemented email notifications with summary for port scanning
- [4.4] Implemented email notification with summary for SSH bruteforce attempts

Work Not Completed

- UI polishing on threshold sliders
- Email formatting for certain alerts.



Sprint 4 Report

Product Name: Intrusion Detection Tool
Team Name: SlugShield
Date: 12/1/25

Actions to Stop Doing

- Stop delaying UI polish → several Sprint 4 tasks focus on aesthetic improvements like background color changes and clearer labels.

- Stop postponing fixes to time accuracy in alerts → task [2.2] emphasizes improving "recent alerts" timestamps.

- Stop leaving email notification UX unclear → task [1.4] aimed to make email entry more intuitive.

Actions to Start Doing

- Start exploring prevention-oriented features → i.e. automatically blocking suspicious IPs during ARP floods or port scans (future intrusion prevention instead of just detection).

- Start researching anomaly-based detection using lightweight ML to complement your rule-based detectors → a natural next phase after threshold improvements (task [2.1]).

- Start designing a more modern, friendly UI/UX, including a minimalist dashboard theme, improved typography, and cleaner layout structures beyond the current background/label updates (tasks [1.1–1.3]).

- Start planning real-time visualization tools, such as live packet flow graphs or alert heatmaps, to make the dashboard more interactive.

- Start modularizing notifications, allowing users to toggle which alerts trigger email notifications (building on tasks [3.1] and [3.2]).

- Start preparing a “deployment-ready” architecture, including Docker support and auto-start scripts for a Raspberry Pi or edge device

- Start planning integration with external logging tools, such as sending alerts to Splunk, ELK Stack, or a local syslog server for long-term analysis.

Actions to Keep Doing

- Keep refining alert clarity and metric accuracy.

- Keep enhancing and expanding communication features like email summaries.

- Keep improving UI/UX iteratively.

- Keep updating the README when new tools or updates are made, preventing setup issues for newer features.

Completed Work

#1. “As a user, I want a more aesthetic webpage so that it will be easy to follow.”

- [1.1] Changed background color
- [1.2] Added labels that clearly depict the functionality and data shown
- [1.3] Added additional labels across the webpage for clarity
- [1.4] Improved the email notification button and input field usability

#2 “As a user, I would like to see accurate metrics being reported so I understand the graphs.”

- [2.1] Added threshold adjustment sidebar where users can control detector thresholds
- [2.2] Improved accuracy of timestamps under “recent alerts”
- [2.3] Added explanations describing the x-axis, y-axis, and rationale behind non-zero graph baselines

#3 “As a user, I want to receive email notifications whenever an alert is detected so I can be notified when I am not near the computer”

- [3.1] Implemented email notifications for ARP flooding, including a summary of the attack
- [3.2] Implemented email notifications for port scanning, including a summary of the attack

Work Not Completed

- Any UI polish tasks not refined or tested fully
- Any metric explanations requiring more visualization improvements
- Any email notification features that remain unverified

Points Scored