

VULNERD

Team Roles — Presentation Note Card

Spring 2026 Senior Design | Florida International University

Each card below covers one team role. Use it as a quick reference before or during the presentation so everyone knows what to say, what to highlight, and how their work connects to the project.

Cloud Security Architect

Team Member: Reynaldo Rodriguez

"I built the environment everyone else tested against."

What I Did:

- Designed and deployed the entire lab environment on a Raspberry Pi 5
- Set up DVWA (the vulnerable web app) inside Docker containers
- Configured Tailscale VPN so only the team could access the lab — nothing was public
- Made sure the environment was realistic enough to produce meaningful security findings

Key Points to Mention:

- The Pi 5 proves you don't need expensive hardware to run a real security lab
- Docker kept the vulnerable app isolated from the host system
- VPN access meant no public exposure — responsible and realistic
- DVWA was set to Security Level Low so all vulnerabilities were exploitable for testing

What to Say (in your own words):

My job was to create the environment that the whole project runs on. I set up a Raspberry Pi 5 running DVWA in Docker, locked it behind a VPN, and made sure it was ready for the team to scan and attack — all without exposing anything to the internet.

Red Team Operator

Team Member: Jonathan Nava-Arenas

"I played the attacker so the team could prove what gets missed."

What I Did:

- Manually executed 8 different attacks against DVWA
- Tested: Brute Force, Command Injection, SQL Injection, XSS (3 types), File Inclusion, File Upload
- Used a browser and curl commands — no special hacking tools needed
- Documented exactly what worked, how, and why it matters

Key Points to Mention:

- Command Injection let me run system commands directly on the server — Critical risk
- SQL Injection dumped the entire user database including password hashes — Critical risk

- File Upload let me plant a backdoor that survives server restarts — Critical risk
- None of these 6 findings were caught by Nessus — that's the point of the whole project
- CSRF was intentionally skipped to avoid breaking the lab credentials

What to Say (in your own words):

My role was to think like an attacker. I ran eight manual attacks against the app — brute force logins, SQL injection, command injection, XSS, file inclusion, and a file upload that planted a backdoor. Six of those attacks found vulnerabilities that the automated scanner completely missed. That's the whole thesis of this project.



Detection Engineer

Team Member: Rickoy Cunningham

"I turned raw findings into rules the team can actually use."

What I Did:

- Built the Detection Playbook — 15 total detections with clear IF/THEN trigger logic
- 9 detections came from Nessus scans, 6 came exclusively from manual testing
- Each detection includes: what it means, how to spot it, priority level, and what to do
- Ran and configured four different Nessus scan types and analyzed what each found vs. missed

Key Points to Mention:

- 40% of all detections — including 3 of 4 Critical findings — only showed up in manual testing
- The credentialed Basic Network Scan was the most productive Nessus scan by far
- Nessus Web Application Tests failed to catch any injection vulnerabilities at all
- Filtering out Info-level plugins broke the Advanced Dynamic Scan pipeline entirely
- The playbook gives small organizations a repeatable response plan, not just a list of alerts

What to Say (in your own words):

I was the detection engineer. I ran multiple Nessus scans and built a playbook with 15 detections — simple if-this-happens-do-that rules. The big finding: 40% of our detections, including three of the four Critical ones, only came from manual testing. No automated scan caught them. That's exactly what this project was designed to prove.



Cyber Threat Intelligence Analyst

Team Member: Luisa Faria Novy E Silva

"I connected our findings to how real-world attackers actually behave."

What I Did:

- Analyzed the Nessus scan results through a threat intelligence lens
- Mapped discovered services (Apache, PHP, SSH) to realistic attacker attack paths
- Built a full attack chain diagram from initial recon through to system compromise
- Mapped findings to 5 MITRE ATT&CK techniques
- Integrated CTI analysis with the Detection Playbook to show how detection and intelligence connect

Key Points to Mention:

- Even 'informational' scan findings give attackers valuable recon data
- The attack chain shows: recon → web app attack OR SSH brute force → privilege escalation → persistence
- MITRE techniques mapped: T1595, T1046, T1190, T1110, T1059
- CTI answers the 'so what?' — it tells you how attackers would actually use what they find
- The Linux Kernel 2.6 identification creates a privilege escalation risk after initial access

What to Say (in your own words):

My role was cyber threat intelligence. I took what the scans found — Apache, PHP, SSH, an older Linux kernel — and analyzed how a real attacker would chain those together. I built an attack diagram showing the path from scanning all the way to persistent access, and mapped everything to MITRE ATT&CK. The point is: even informational findings are dangerous in the wrong hands.



AI & Automation Lead

Team Member: Anthony Whiteman

"I built the tool that makes this whole project useful to people without a security team."

What I Did:

- Built VulNerd — a web tool that turns Nessus CSV exports into plain-English report cards
- 5-stage pipeline: Collect → Parse → Analyze (Gemini AI) → Comply → Deliver
- Maps findings to compliance frameworks: HIPAA, PCI-DSS, NIST 800-53, FERPA, CIS Controls
- Delivers a graded report card, compliance dashboard, PDF export, and built-in AI chat assistant
- Built entirely in Python with a custom HTML/CSS/JS interface — no external web framework

Key Points to Mention:

- VulNerd was built for people who open a Nessus export and immediately feel lost
- The AI gives the scan a letter grade (A–F) with plain-English explanations of each finding
- The compliance dashboard shows your score by industry and includes real breach cost comparisons
- The built-in chat lets users ask follow-up questions about their specific scan results
- This is the automation piece — scanning → processing → reporting as a complete pipeline

What to Say (in your own words):

I built VulNerd, the tool that ties everything together. You upload a Nessus CSV, and it runs through five stages — parse, analyze with Google Gemini AI, map to your compliance framework, and deliver a graded report card in plain English. It gives small businesses a way to understand their vulnerabilities without needing a dedicated security analyst to translate the output.

The One Line Every Team Member Should Know

40% of our detections — including 3 of 4 Critical findings — were only found by manual testing. No automated scanner caught them. That proves why a layered approach matters.

Quick Role Reference

Role	Team Member	One-Sentence Summary
Cloud Security Architect	Reynaldo Rodriguez	Built the Raspberry Pi lab environment and locked it behind a VPN.
Red Team Operator	Jonathan Nava-Arenas	Manually attacked DVWA across 8 modules and found 6 Critical/High vulnerabilities.
Detection Engineer	Rickoy Cunningham	Built the 15-detection playbook and ran all four Nessus scan types.
CTI Analyst	Luisa Faria Novy E Silva	Mapped findings to attacker behavior and MITRE ATT&CK techniques.
AI & Automation Lead	Anthony Whiteman	Built VulNerd — the AI-powered tool that turns scan data into plain-English reports.