

## ServiceNow GRC: Automated "NIST ComplianceGuard" Dashboard

---

### Project Overview

In this project, I architected and deployed a custom Governance, Risk, and Compliance (GRC) application within the ServiceNow platform. The objective was to transition an organization from manual, spreadsheet-based compliance tracking to an automated, centralized system of record based on the NIST SP 800-53 framework. This report documents the technical implementation, from data ingestion to executive dashboarding.

### Scenario

An mid-size enterprise is currently managing its NIST 800-53 compliance posture using static Excel spreadsheets. This manual process has led to several critical issues:

- Lack of Visibility: Executives cannot see real-time risks.
- Data Stagnation: Compliance data is outdated the moment the spreadsheet is saved.
- Human Error: Risk scoring relies on manual data entry rather than automated logic.
- Slow Response Time: High Mean Time to Respond (MMTR) due to delays between detecting failures and notifying the IT department.

### Task

As their GRC Developer, my task was to build an automated application (project name "NIST ComplianceGuard") in ServiceNow that would:

1. Centralize Data: Migrate NIST controls from CSV into a structured database.
2. Automate Logic: Enforce objective risk scoring using system rules.
3. Bridge GRC & IT: Automatically create incident tickets and send email notifications to designated IT employees when compliance issues arise.

### Tools Used

- Platform: ServiceNow (Yokohama)
- Core Modules: System Definition (Tables & Columns), System Import Sets, Flow Designer, Reporting, Dashboards
- Framework: NIST SP 800-53 (Security and Privacy Controls). *\*\*Note: Can be replaced with other frameworks (e.g. SOC 2, ISO 27001, etc.)*

# Phase 1: Architecture and Setup

## Action 1: Manual Table Architecture

I utilized the System Definition module to architect the database schema manually.

- Navigation: System Definition > Tables > New
- Configuration:
  - Name: NIST ctrl [u\_nist\_ctrl]
  - Extends Table: Task [task]
  - Some custom columns created include:
    - Control ID (String, Max Length 100)
    - Risk Level (Choice List: Low, Medium, High, Critical)
    - Compliance Status (Choice List: Compliant, Non-Compliant)

### Why I did this:

By extending the core Task table, my custom application automatically inherited enterprise features like SLA tracking, assignment groups, and the "Work Notes" audit trail. This ensures that every time a control is modified, the system logs who changed it and when, creating a permanent audit chain of custody.

The screenshot shows the ServiceNow Table - New Record interface. At the top, there are fields for Label (NIST ctrl), Name (u\_nist\_ctrl), and Extends table (Task). On the right, there are checkboxes for Application (Global), Create module, Create mobile module, Add module to menu (set to '-- Create new --'), and a New menu name field (NIST GRC). Below this, the 'Table Columns' section is visible, displaying a list of columns with their labels, types, and properties. The columns listed are: Control ID (String, max length 100), Control Name (String, max length 100), Compliance Status (Choice, max length 100), Risk Level (Choice, max length 100), Short Description (String, max length 250), Description (String, max length 1000), and Owner (String, max length 100). The 'Submit' and 'Cancel' buttons are at the bottom left of the form.

Figure 1.1 The Table Definition screen showing the u\_nist\_ctrl table extending the Task

## Action 2: ETL (Extract, Transform, Load)

To migrate the Microsoft Excel file, I downloaded the spreadsheet as a .CSV file and utilized System Import Sets.

- Navigation: System Import Sets > Load Data
- Configuration:
  - Data Source: NIST\_Data\_Set.csv
  - Transform Map: Created a map named NIST Final Map
  - Field Mapping: Utilized Mapping Assist to link source CSV headers to target table fields (e.g. Risk Level → Risk Level)
  - Coalesce Field: Set Control ID → True

### Why I did this:

The "Coalesce" setting is critical for data integrity. By setting the Control ID (e.g., "AC-1") as the unique key, I made sure that re-running the import would update existing records rather than creating duplicates.

As for the NIST Data Set, I downloaded the official NIST SP 800-53 Rev 5 dataset and hand-picked 15 out of over 1,000 security controls. This was to capture the complete lifecycle of a threat: Prevention (Access Control, MFA, Firewalls), Detection (Logging, Vulnerability Scanning), and Response (Incident Handling). By structuring the data model around these "Heavy Hitter" controls, which map directly to major frameworks like ISO 27001, SOC 2, and HIPAA, I made sure the final dashboard would visualize a holistic view of the organization's cyber hygiene across the core functions of Protect, Detect, and Respond.

The screenshot shows a ServiceNow list view titled "NIST ctrls". The table has columns: Control ID, Control Name, Description, Risk Level, Owner, and Compliance Status. The data is as follows:

Control ID	Control Name	Description	Risk Level	Owner	Compliance Status
AT-2	Security Awareness Training	Provide basic security awareness training...	Medium	HR Manager	Compliant
RA-5	Vulnerability Scanning	Monitor and scan for vulnerabilities in...	Critical	Network Team	Compliant
AC-1	Access Control Policy	Document and disseminate access control...	Low	System Admin	Compliant
IA-5	Authenticator Management	Manage system authenticators (passwords/...	High	Security Team	Compliant
SC-7	Boundary Protection	Monitor and control communications at th...	Critical	Network Team	Compliant
AU-6	Audit Record Review	Review and analyze system audit records ...	Medium	Audit Team	Compliant
AC-6	Least Privilege	Employ the principle of least privilege...	High	Security Team	Compliant
PE-2	Physical Access Authorizations	Develop and maintain a list of individua...	Low	Facilities	Compliant
IA-2	Identification and Authentication	Uniquely identify and authenticate organ...	Critical	Justin Min	Non-Compliant
AU-2	Event Logging	Identify the types of events that the sy...	Critical	System Admin	Non-Compliant
SI-4	System Monitoring	Monitor the system to detect attacks and...	Medium	SOC Team	Non-Compliant
AC-3	Access Enforcement	Enforce approved authorizations for logi...	High	Security Team	Non-Compliant
IR-4	Incident Handling	Implement an incident handling capability...	High	SOC Team	Non-Compliant
CM-8	System Component Inventory	Develop and document an inventory of sys...	Medium	IT Ops	Non-Compliant
AC-2	Account Management	Manage information system accounts to in...	Critical	Justin Min	Non-Compliant

Figure 1.2 The populated NIST Controls list view after successful ETL execution

## Phase 2: Automation & Logic

### Action 1: Business Rule Logic

To remove human error from risk assessment, I implemented a Business Rule to run server-side logic.

- Navigation: System Definition > Business Rules
- Name: Auto-Escalate Failure
- Condition: Compliance Status CHANGES TO Non-Compliant
- Action: SET Risk Level TO Critical

### Why I did this:

I used a "Before-Update" Business Rule to intercept the data transaction. The moment a user changes a control to "Non-Compliant", the system forces the Risk Level to Critical. This ensures that no matter who is editing the record, a control failure is always treated as a top-priority risk.

You may notice there is no rule like "*Status CHANGES TO Non-Compliant → SET Risk Level TO Critical*". This is an intentional feature, as you would NOT want it to automatically reset just because someone changed the status dropdown. A control that goes Critical stays Critical until a human manually reviews and resets it. This prevents accidental de-escalation "flapping" (status jumping up and down).

The screenshot shows the ServiceNow interface for creating a new Business Rule. The top navigation bar includes 'servicenow', 'All', 'Favorites', 'History', 'Workspaces', and 'Admin'. The title bar says 'Business Rule - New Record'. The main form has the following fields:

- Name:** Auto-Escalate Failure
- Table:** NIST ctrl [u\_nist\_ctrl]
- Application:** Global
- Active:**
- Advanced:**

Below the basic settings, there are tabs for 'When to run', 'Actions', and 'Advanced'. The 'When to run' tab is selected, showing the following configuration:

- When:** before
- Order:** 100
- Operations:** Insert  (checked), Update  (checked), Delete  (unchecked), Query  (unchecked)
- Filter Conditions:** Add Filter Condition, Add OR Clause. A condition is shown: Compliance Status changes to Non-Compliant.
- Role conditions:**

At the bottom left is a 'Submit' button.

Figure 2.1 Displays the Business Rule configuration

## Action 2: Workflow Automation (Flow Designer)

I architected an automated workflow to bridge the gap between Compliance and IT Operations utilizing ServiceNow's Flow Designer feature.

- Navigation: Process Automation > Flow Designer
- Trigger: Record Created/Updated where Risk Level is Critical.
  - Action A (IT Service Management): Utilized the Create Record action to generate an Incident ticket. I used data pills to dynamically map the Control ID and Description from the GRC record into the Incident's "Short Description" field.
  - Action B (Notification): Configured a Send Email action to alert the Control Owner immediately.

### Why I did this:

This reduces the "Mean Time to Respond" (MTTR). By automating the creation of an Incident, we can make sure that a compliance failure immediately enters the IT remediation queue, allowing for 24/7 monitoring and removing the need for manual emails or phone calls.

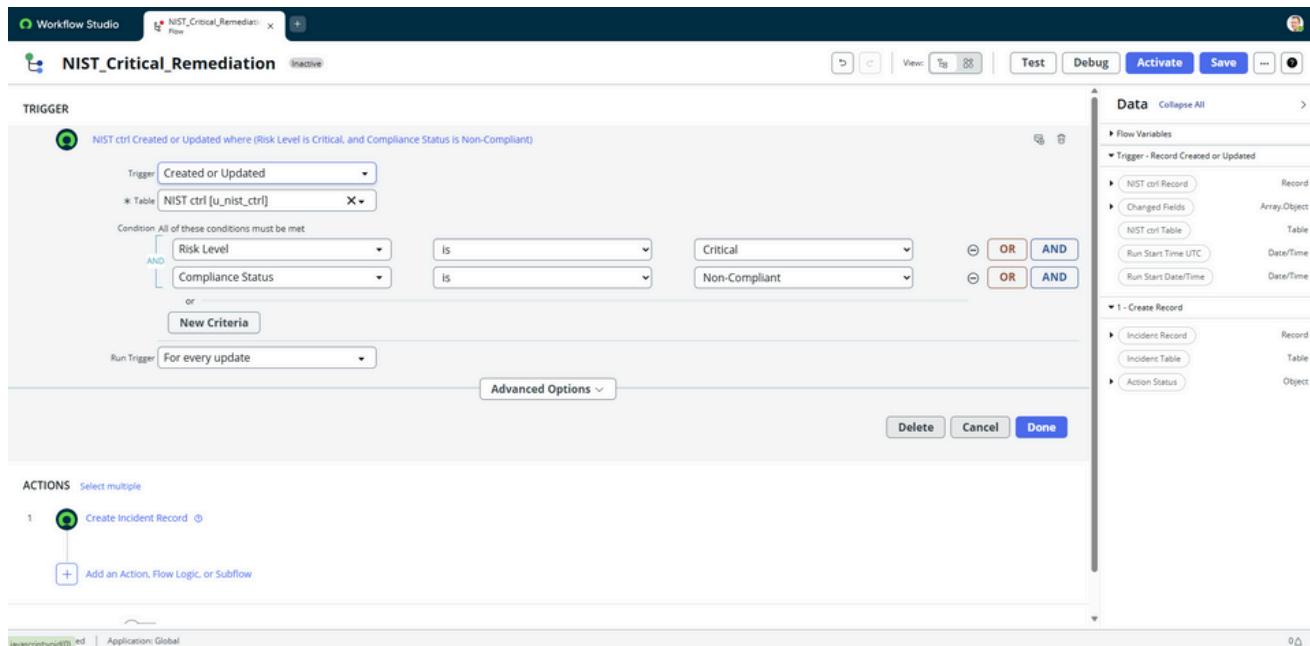


Figure 2.2 The configuration for the Trigger function in Flow Designer

The screenshot shows the Workflow Studio interface for a flow named "NIST\_Critical\_Remediation". The main area displays a single step labeled "Create Incident Record". The "Action Properties" section shows "Create Record" selected. The "Action Inputs" section contains fields for "Table" (Incident [incident]), "Fields" (Short description, Caller, Description, Impact, Urgency), and "Values" (all set to "1 - High"). The right sidebar lists available actions categorized under "Data", "Flow Variables", and "Trigger - Record Created or Updated".

Figure 2.3 The configuration for the automated Incident ticket creating action

The screenshot shows the Workflow Studio interface for the same flow. A new step "Send Email" has been added. The "Action Properties" section shows "Send Email" selected. The "Action Inputs" section includes "Target Record" (Drag and drop record data pill), "Table" (Select a Table), "Include Watermark" checked, "To" (justinmin687@gmail.com), "CC", "BCC", and "Subject" (URGENT: Control Failure -). The "Body" section contains rich text with the message: "Attention, a critical control failure has been detected. An incident has been created automatically." The right sidebar shows the continuation of the "Data" and "Trigger" sections from the previous screenshot.

Figure 2.4 The configuration for the automated email notification

## Phase 3: Visualizations & Analytics

### Action 1: Executive Dashboarding

I aggregated the data into a “CISO Command Center” using the Reporting module and Dashboards.

- Compliance Overview (Pie Chart): Provides a macro-level view of the organization’s pass/fail ratio.
- Risk Distribution (Bar Chart): Utilized “Stacked” configurations to show risk severity overlaid with compliance status.
- Critical Action Items (List): A filtered data view showing only active records where Risk = Critical.

### Why I did this:

This transforms raw data into a clear and convenient data visuals for stakeholders. Executives no longer need to parse spreadsheets; they can view the dashboard to immediately identify “Hot Spots” in the security posture. Actionable intelligence updated in real-time.

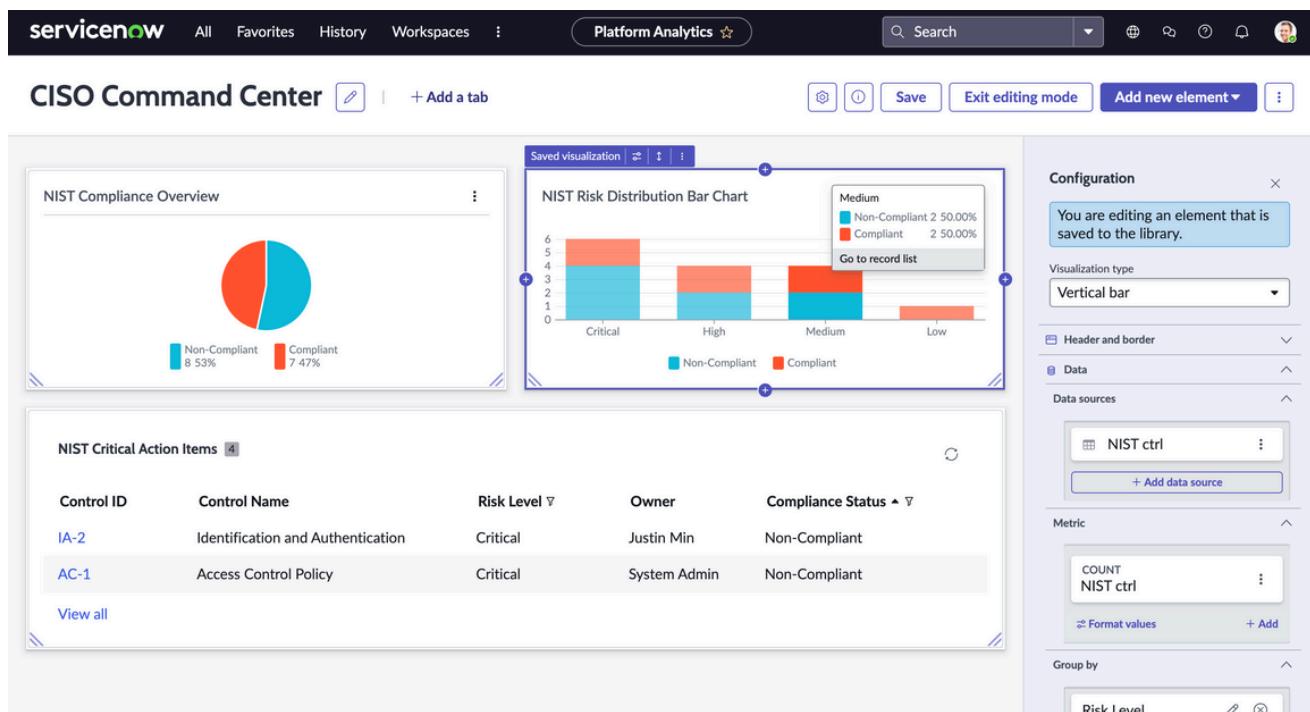


Figure 3.1 Interactive and customizable dashboard displaying Pie Chart, Bar Chart, and List View

## Phase 4: Validation (User Acceptance Testing)

### Action 1: Smoke Testing & Verification

In order to validate system integrity and make sure the application functions as intended, I performed a “Smoke Test” using a simulation record to verify the automation pipeline.

1. Baseline Injection: Created a dummy control (SIM-999) with “Compliant” status and “Low” risk.
2. Event Trigger: Manually updated the record to “Non-Compliant” to simulate a sudden audit failure.
3. Verification of Success Criteria:
  - Logic: Verified Risk Level automatically escalated from Low to Critical.
  - Automation: Verified Incident INC0010006 was created and linked to the control.
  - Notification: Verified email transmission in the sys\_email.list logs.
  - Visibility: Verified SIM-999 instantly appeared on the CISO Dashboard.

### Why I did this:

Thorough testing proves that the application handles the full lifecycle of a security event correctly, ensuring reliability before deployment to production.

All	Control ID	Control Name	Description	Risk Level	Owner	Compliance Status
	AT-2	Security Awareness Training	Provide basic security awareness trainin...	Medium	HR Manager	Compliant
	RA-5	Vulnerability Scanning	Monitor and scan for vulnerabilities in ...	Critical	Network Team	Compliant
	IA-5	Authenticator Management	Manage system authenticators (passwords/...	High	Security Team	Compliant
	SC-7	Boundary Protection	Monitor and control communications at th...	Critical	Network Team	Compliant
	SIM-999	Simulation Test Protocol	This is a test injection to verify the a...	Low	Justin Min	Compliant
	AU-6	Audit Record Review	Review and analyze system audit records ...	Medium	Audit Team	Compliant
	AC-6	Least Privilege	Employ the principle of least privilege,...	High	Security Team	Compliant
	PE-2	Physical Access Authorizations	Develop and maintain a list of individua...	Low	Facilities	Compliant
	IA-2	Identification and Authentication	Uniquely identify and authenticate organ...	Critical	Justin Min	Non-Compliant
	AC-1	Access Control Policy	Document and disseminate access control ...	Critical	System Admin	Non-Compliant
	AU-2	Event Logging	Identify the types of events that the sy...	Critical	System Admin	Non-Compliant
	SI-4	System Monitoring	Monitor the system to detect attacks and...	Medium	SOC Team	Non-Compliant
	AC-3	Access Enforcement	Enforce approved authorizations for logi...	High	Security Team	Non-Compliant
	IR-4	Incident Handling	Implement an incident handling capabilit...	High	SOC Team	Non-Compliant
	CM-8	System Component Inventory	Develop and document an inventory of sys...	Medium	IT Ops	Non-Compliant
	AC-2	Account Management	Manage information system accounts to in...	Critical	Justin Min	Non-Compliant

Figure 4.1 Dummy Control SIM-999 injected with “Compliant” Status and “Low” Risk Level

Control ID	Control Name	Description	Risk Level	Owner	Compliance Status
AT-2	Security Awareness Training	Provide basic security awareness trainin...	Medium	HR Manager	Compliant
RA-5	Vulnerability Scanning	Monitor and scan for vulnerabilities in ...	Critical	Network Team	Compliant
IA-5	Authenticator Management	Manage system authenticators (passwords/...	High	Security Team	Compliant
SC-7	Boundary Protection	Monitor and control communications at th...	Critical	Network Team	Compliant
AU-6	Audit Record Review	Review and analyze system audit records ...	Medium	Audit Team	Compliant
AC-6	Least Privilege	Employ the principle of least privilege...	High	Security Team	Compliant
PE-2	Physical Access Authorizations	Develop and maintain a list of individua...	Low	Facilities	Compliant
IA-2	Identification and Authentication	Uniquely identify and authenticate organ...	Critical	Justin Min	Non-Compliant
AC-1	Access Control Policy	Document and disseminate access control ...	Critical	System Admin	Non-Compliant
AU-2	Event Logging	Identify the types of events that the sy...	Critical	System Admin	Non-Compliant
<input checked="" type="checkbox"/> <span style="color: #0070C0;">SIM-999</span>	Simulation Test Protocol	This is a test injection to verify the a...	Critical	Justin Min	Non-Compliant
SI-4	System Monitoring	Monitor the system to detect attacks and...	Medium	SOC Team	Non-Compliant
AC-3	Access Enforcement	Enforce approved authorizations for logi...	High	Security Team	Non-Compliant
IR-4	Incident Handling	Implement an incident handling capabilit...	High	SOC Team	Non-Compliant
CM-8	System Component Inventory	Develop and document an inventory of sys...	Medium	IT Ops	Non-Compliant
AC-2	Account Management	Manage information system accounts to in...	Critical	Justin Min	Non-Compliant

Figure 4.2 Dummy Control SIM-999 “Non-Compliant” Status change automatically raising Risk Level to “Critical”

Number	Opened	Short description	Caller	Priority	State	Category	Assignment group	Assigned to	Updated	Updated by
INC0010006	2026-01-07 19:51:12	Critical NIST Alert: SIM-999	(empty)	1 - Critical	New	Inquiry / Help	(empty)	(empty)	2026-01-07 19:51:12	system
INC0010005	2026-01-07 19:45:05	Critical NIST Alert: AC-1	(empty)	1 - Critical	New	Inquiry / Help	(empty)	(empty)	2026-01-07 19:45:05	system
INC0010004	2026-01-07 18:40:38	Critical NIST Alert: AC-1	(empty)	1 - Critical	New	Inquiry / Help	(empty)	(empty)	2026-01-07 18:40:38	system
INC0010003	2025-12-31 07:34:01	Critical Yokohama Alert: SIM-999	(empty)	1 - Critical	New	Inquiry / Help	(empty)	(empty)	2026-01-07 18:58:36	system
INC0010002	2025-12-31 06:21:07	Critical Yokohama Alert: IA-2	(empty)	1 - Critical	New	Inquiry / Help	(empty)	(empty)	2026-01-07 18:58:32	system
INC0010001	2025-12-31 06:08:14	Critical Yokohama Alert: AC-2	(empty)	1 - Critical	New	Inquiry / Help	(empty)	(empty)	2026-01-07 18:58:37	system
INC0009009	2018-08-30 01:06:16	Unable to access the shared folder.	David Miller	4 - Low	New	Inquiry / Help	(empty)	(empty)	2018-12-12 23:30:24	admin
INC0009005	2018-08-31 21:35:21	Email server is down.	David Miller	1 - Critical	New	Software	(empty)	(empty)	2018-12-12 23:18:55	admin
INC0009004	2018-09-01 06:13:30	Defect tracking tool is down.	David Miller	3 - Moderate	Closed	Software	(empty)	(empty)	2025-12-19 20:46:19	system
INC0009003	2018-08-30 02:17:32	Cannot sign into the company portal app	David Miller	3 - Moderate	Closed	Inquiry / Help	(empty)	(empty)	2018-12-12 23:39:53	admin
INC0009002	2018-09-16 05:10:00	My computer is not detecting the headphone	David Miller	3 - Moderate	Closed	Hardware	(empty)	(empty)	2025-12-19 20:46:19	system

Figure 4.3 Dummy Control SIM-999 appears in Incident list and ticket is created

The screenshot shows a ServiceNow dashboard titled "NIST Critical Action Items". The top navigation bar includes links for All, Favorites, History, Workspaces, and a search bar. Below the header, a breadcrumb trail indicates the current view: All > Risk Level = Critical > Compliance Status = Non-Compliant. The main content area displays a table with the following columns: Control ID, Control Name, Description, Risk Level, Owner, and Compliance Status. The table contains five rows of data:

Control ID	Control Name	Description	Risk Level	Owner	Compliance Status
IA-2	Identification and Authentication	Uniquely identify and authenticate organ...	Critical	Justin Min	Non-Compliant
AC-1	Access Control Policy	Document and disseminate access control ...	Critical	System Admin	Non-Compliant
AU-2	Event Logging	Identify the types of events that the sy...	Critical	System Admin	Non-Compliant
SIM-999	Simulation Test Protocol	This is a test injection to verify the a...	Critical	Justin Min	Non-Compliant
AC-2	Account Management	Manage information system accounts to in...	Critical	Justin Min	Non-Compliant

At the bottom of the table, there are navigation buttons for page 1 of 5.

*Figure 4.4 Automatically appears and immediately updates CISO Command Center Dashboard*

The screenshot shows a ServiceNow dashboard titled "Emails". The top navigation bar includes links for All, Favorites, History, Workspaces, Admin, and a search bar. Below the header, a breadcrumb trail indicates the current view: All > Recipients starts with justin. The main content area displays a table with the following columns: Created, Recipients, Subject, Type, Notification type, and User ID. The table contains two rows of data:

Created	Recipients	Subject	Type	Notification type	User ID
2026-01-07 19:51:12	justinmin687@gmail.com	URGENT: Control Failure - SIM-999	send-ready	SMTP	(empty)
2026-01-07 19:45:05	justinmin687@gmail.com	URGENT: Control Failure - AC-1	send-ready	SMTP	(empty)

At the bottom of the table, there are navigation buttons for page 1 of 2.

*Figure 4.5 Email is sent to assigned owner of control*

*\*\*Note: Due to the restrictions of the ServiceNow Personal Developer Instance (PDI), outbound email is intercepted by the system to prevent spam. The generated email was verified in the System Email Logs (sys\_email.list) with a status of 'Send-ready', confirming the Flow executed correctly even though the email was not delivered to the external inbox."*

## Conclusion

This project successfully engineered a holistic GRC solution, transforming the organization's approach to compliance from a reactive, manual effort into a proactive, automated operation. By leveraging the full stack of ServiceNow's capabilities, from database architecture to executive visualizations, I delivered a system that ensures:

- Audit Readiness: Every control modification is tracked via the extended Task table architecture.
- Operational Velocity: The integration of GRC logic with ITSM workflows reduced the response time to critical failures from days to milliseconds.
- Executive Visibility: The "CISO Command Center" provides leadership with a single source of truth, eliminating the reliance on static, outdated spreadsheets.

Ultimately, this application proves that ServiceNow is not just a ticketing system, but a powerful development platform capable of enforcing complex business logic and driving enterprise security.

## Reflection

This project provided me with deep, hands-on experience in the ServiceNow "Yokohama" environment, specifically regarding the strategic architecture of enterprise applications. I learned that extending the core Task table is far more efficient than building from scratch, as it allows developers to leverage existing features like SLAs and Audit Trails. Additionally, working with Transform Maps solidified my understanding of data integrity; specifically, the concept of "Coalescing" taught me how to design imports that are safe to run repeatedly without corrupting the database.

Beyond the technical configuration, this project sharpened my architectural mindset regarding security logic. I discovered the importance of "Latch Logic" when my Business Rule prevented the risk level from auto-resetting—a feature I now understand is critical for preventing accidental de-escalation in a security context. Integrating the GRC application with the Incident Management module via Flow Designer also highlighted the power of cross-module workflows, teaching me how to map data across different scopes to create a seamless user experience.

Finally, the troubleshooting process during User Acceptance Testing was a major growth moment. Identifying that the email notification failure was due to a Data Type mismatch (String vs. Reference) and PDI spam restrictions gave me real-world debugging experience that goes beyond documentation. I am now confident in my ability to take raw business requirements, from database design to executive reporting, and engineer a fully functional, automated business application on the ServiceNow platform.