

AI driven coding model for visualizing security code

[Model1]

AI driven code for cloud server security status using Matplotlib

[IoT Server Security Dash](#)

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import time

# Sample Cloud Server Names
servers = [f"Cloud_Server_{i}" for i in range(1, 6)]

def generate_security_data():
    """Simulates AI-based security risk scores."""
    return pd.DataFrame({
        "Server": servers,
        "Risk Score": np.random.randint(10, 100, size=len(servers)),
        "Anomalies Detected": np.random.randint(0, 10, size=len(servers))
    })

def plot_security_status():
    """Generates a Matplotlib bar plot for security risk scores."""
    df = generate_security_data()

    plt.figure(figsize=(10, 6))
    bars = plt.bar(df['Server'], df['Risk Score'], color='red')

    # Annotate each bar with the number of anomalies detected
    for bar, anomalies in zip(bars, df['Anomalies Detected']):
        plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height() + 2, f"{anomalies} anomalies", ha='center', fontsize=12)

    plt.title("AI-Driven Cloud Server Security Status")
    plt.xlabel("Server")
    plt.ylabel("Risk Score")
    plt.ylim(0, 110)
    plt.show()

# Simulate real-time updates
while True:
    plot_security_status()
    time.sleep(5)  # Refresh every 5 seconds
```

[Model2]

AI driven code for IoT server security status using Dash

ChatGPT의 말:

Python code snippet using Dash to create an AI-driven IoT server security status dashboard. This example assumes that you have an AI model that assesses security risks and provides a risk score for each IoT server. This dashboard provides real-time security status updates for IoT servers, displaying AI-driven risk scores and detected anomalies.

```
import dash
from dash import dcc, html
from dash.dependencies import Input, Output
import plotly.graph_objs as go
import random
import pandas as pd
import numpy as np

# Initialize the Dash app
app = dash.Dash(__name__)

# Sample IoT server data (simulated)
servers = [f"IoT_Server_{i}" for i in range(1, 6)]

def generate_security_data():
    """ Simulates AI-based security risk scores. """
    return pd.DataFrame({
        "Server": servers,
        "Risk Score": np.random.randint(10, 100, size=len(servers)),
        "Anomalies Detected": np.random.randint(0, 10, size=len(servers))
    })

# Layout of the dashboard
app.layout = html.Div([
    html.H1("AI-Driven IoT Server Security Dashboard", style={'textAlign': 'center'}),
    dcc.Interval(id='interval-component', interval=5000, n_intervals=0), # Auto refresh every 5 sec
    dcc.Graph(id='security-status-graph'),
])

@app.callback(
    Output('security-status-graph', 'figure'),
    [Input('interval-component', 'n_intervals')]
)
def update_graph(n):
    df = generate_security_data()

    fig = go.Figure()
    fig.add_trace(go.Bar(x=df['Server'], y=df['Risk Score'], name='Risk Score', marker_color='red'))
    fig.add_trace(go.Scatter(x=df['Server'], y=df['Anomalies Detected'], name='Anomalies', mode='markers',
                             marker=dict(size=10, color='blue')))

    fig.update_layout(title='IoT Server Security Status', xaxis_title='Server', yaxis_title='Score',
                       barmode='group')
    return fig

if __name__ == '__main__':
    app.run_server(debug=True)
```

[Model3]

AI driven code for cloud server security status using Seaborn

chatgpt

[IoT Server Security Dash](#)

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import time

# Sample Cloud Server Names
servers = [f'Cloud_Server_{i}' for i in range(1, 6)]

def generate_security_data():
    """Simulates AI-based security risk scores."""
    return pd.DataFrame({
        "Server": servers,
        "Risk Score": np.random.randint(10, 100, size=len(servers)),
        "Anomalies Detected": np.random.randint(0, 10, size=len(servers))
    })

def plot_security_status():
    """Generates a seaborn bar plot for security risk scores."""
    df = generate_security_data()

    plt.figure(figsize=(10, 6))
    sns.barplot(x="Server", y="Risk Score", data=df, palette="Reds")

    for index, row in df.iterrows():
        plt.text(index, row["Risk Score"] + 2, f'{row["Anomalies Detected"]} anomalies', ha='center',
        fontsize=12)

    plt.title("AI-Driven Cloud Server Security Status")
    plt.xlabel("Server")
    plt.ylabel("Risk Score")
    plt.ylim(0, 110)
    plt.show()

# Simulate real-time updates
while True:
    plot_security_status()
    time.sleep(5) # Refresh every 5 seconds
```

[Model4]

AI driven code for cloud server security status using Seaborn

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import time

# Sample Cloud Server Names
servers = [f"Cloud_Server_{i}" for i in range(1, 6)]

def generate_security_data():
    """Simulates AI-based security risk scores."""
    return pd.DataFrame({
        "Server": servers,
        "Risk Score": np.random.randint(10, 100, size=len(servers)),
        "Anomalies Detected": np.random.randint(0, 10, size=len(servers))
    })

def plot_security_status():
    """Generates a seaborn bar plot for security risk scores."""
    df = generate_security_data()

    plt.figure(figsize=(10, 6))
    sns.barplot(x="Server", y="Risk Score", data=df, palette="Reds")

    for index, row in df.iterrows():
        plt.text(index, row["Risk Score"] + 2, f"{row['Anomalies Detected']} anomalies", ha='center',
        fontsize=12)

    plt.title("AI-Driven Cloud Server Security Status")
    plt.xlabel("Server")
    plt.ylabel("Risk Score")
    plt.ylim(0, 110)
    plt.show()

# Simulate real-time updates
while True:
    plot_security_status()
    time.sleep(5) # Refresh every 5 seconds
```

[Model5]

AI driven code for cloud server security status using Plotly

[IoT Server Security Dash](#)

```
import plotly.graph_objects as go
import pandas as pd
import numpy as np
import time

# Sample Cloud Server Names
servers = [f"Cloud_Server_{i}" for i in range(1, 6)]

def generate_security_data():
    """Simulates AI-based security risk scores."""
    return pd.DataFrame({
        "Server": servers,
        "Risk Score": np.random.randint(10, 100, size=len(servers)),
        "Anomalies Detected": np.random.randint(0, 10, size=len(servers))
    })

def plot_security_status():
    """Generates a Plotly bar chart for security risk scores."""
    df = generate_security_data()

    fig = go.Figure()
    fig.add_trace(go.Bar(
        x=df['Server'],
        y=df['Risk Score'],
        name='Risk Score',
        marker_color='red'
    ))

    fig.add_trace(go.Scatter(
        x=df['Server'],
        y=df['Anomalies Detected'],
        name='Anomalies Detected',
        mode='markers',
        marker=dict(size=10, color='blue')
    ))

    fig.update_layout(
        title='AI-Driven Cloud Server Security Status',
        xaxis_title='Server',
        yaxis_title='Score',
        barmode='group'
    )
    fig.show()

# Simulate real-time updates
while True:
    plot_security_status()
    time.sleep(5)  # Refresh every 5 seconds
```