

dashboard.js

```
document.addEventListener('DOMContentLoaded', function() {
    // === Timeline Chart ===
    // Shows CVEs discovered each month (trend line)
    const timelineChartEl = document.getElementById('timelineChart');
    if (timelineChartEl) {
        const ctx = timelineChartEl.getContext('2d');
        const labels = window.timelineData.labels || []; //x-axis: months,
like '2024-01'
        const data = window.timelineData.values || []; //y-axis: number of
CVEs for that month
        new Chart(ctx, {
            type: 'line',
            data: {
                labels: labels,
                datasets: [{
                    label: 'CVEs Per Month',
                    data: data,
                    borderColor: '#63a4ff', //main line color
                    backgroundColor: 'rgba(99, 164, 255, 0.1)', //fill under
the line

                    fill: true,
                    tension: 0.3, //slight curve to line
                    borderWidth: 2,
                    pointRadius: 4,
                    pointHoverRadius: 6,
                    pointBackgroundColor: '#63a4ff'
                }]
            },
            options: {
                responsive: true,
                maintainAspectRatio: false,
                plugins: {
                    legend: { display: false },
                    tooltip: { mode: 'index', intersect: false }
                },
                scales: {
                    x: {
                        grid: { display: false },
                        ticks: {
                            color: '#8b9bb4', // muted tick labels
                            //Only show year labels for January, blank
```

```

otherwise.

        callback: function(value, index, values) {
            const label = this.getLabelForValue(value);
            if (!label) return null;
            //If label month is '01', show year (2024),

else blank

            if (label.slice(5,7) === "01") return

label.slice(0, 4);

            return "";
        },
        autoSkip: false,
        maxRotation: 0,
        minRotation: 0
    }
},
y: {
    beginAtZero: true,
    grid: { display: false },
    ticks: {
        color: '#8b9bb4',
        stepSize: 500,
        //Only show y-labels divisible by 500, blank

otherwise

        callback: function(value, index, ticks) {
            return value % 500 === 0 ? value : "";
        }
    }
},
interaction: {
    mode: 'nearest',
    axis: 'x',
    intersect: false
},
// On click, send user to filtered view for that year/month
onClick: (evt, activeEls) => {
    if (activeEls && activeEls.length) {
        const chart = activeEls[0].element.$context.chart;
        const idx = activeEls[0].index;
        const label = chart.data.labels[idx];
        if (label && label.length === 7) {
            const [year, month] = label.split('-');
            window.location.href = '/vulnerabilities?year=' +
year + '&month=' + month;
        }
    }
}

```

```

        }
    }
});
}
// === Daily Trend Chart ===
if (document.getElementById('dailyTrendChart')) {
    const ctxDaily =
document.getElementById('dailyTrendChart').getContext('2d');
    const labels = window.timelineDataDaily.labels || []; // x-axis: daily
dates '2024-01-05'
    const dataValues = window.timelineDataDaily.values || []; // y-axis:
number per day

    // Only label every 7th day for some x-axis compression
    const dayLabels = labels.map((label, idx) => {
        return (idx % 7 === 0) ? label.slice(5, 10) : '';
    });

    new Chart(ctxDaily, {
        type: 'bar',
        data: {
            labels: dayLabels,
            datasets: [{
                label: 'CVEs per Day',
                data: dataValues,
                backgroundColor: 'rgba(99, 164, 255, 0.8)',
                borderColor: 'rgba(99, 164, 255, 1)',
                tension: 0.3,
                fill: true,
                pointRadius: 3,
                pointHoverRadius: 6
            }]
        },
        options: {
            responsive: true,
            maintainAspectRatio: false,
            plugins: {
                legend: { display: false },
                tooltip: {
                    mode: 'index',
                    intersect: false,
                    callbacks: {
                        // Tooltip shows actual date label on hover
                        title: function(context) {
                            const idx = context[0].dataIndex;
                            return window.timelineDataDaily.labels &&

```

```

window.timelineDataDaily.labels[idx] ?
                                window.timelineDataDaily.labels[idx] : '';
                                }
                                }
                                },
                                },
                                scales: {
                                    y: {
                                        beginAtZero: true,
                                        grid: { display: false },
                                        ticks: { color: '#90caf9' }
                                    },
                                    x: {
                                        grid: { display: false },
                                        ticks: {
                                            color: '#90caf9',
                                            maxRotation: 0,
                                            autoSkip: false,
                                            maxTicksLimit: 10
                                        }
                                    }
                                },
                                interaction: {
                                    mode: 'nearest',
                                    axis: 'x',
                                    intersect: false
                                },
                                // On bar click, go to vulnerabilities for that year/month/day
                                onClick: (evt, activeElements) => {
                                    if (activeElements && activeElements.length > 0) {
                                        const idx = activeElements[0].index;
                                        const fullDate = window.timelineDataDaily.labels[idx];
                                        if (fullDate) {
                                            const year = fullDate.slice(0, 4);
                                            const month = parseInt(fullDate.slice(5, 7));
                                            const day = parseInt(fullDate.slice(8, 10));
                                            window.location.href = '/vulnerabilities?year=' +
year + '&month=' + month + '&day=' + day;
                                        }
                                    }
                                }
                                });
                                }
                                // === Severity Doughnut Chart ===
                                if (document.getElementById('severityPie')) {

```

```

    const ctxPie =
document.getElementById('severityPie').getContext('2d');
    new Chart(ctxPie, {
      type: 'doughnut',
      data: {
        labels: ["Critical", "High", "Medium", "Low"],
        datasets: [{
          data: [
            window.severityStats.CRITICAL || 0,
            window.severityStats.HIGH || 0,
            window.severityStats.MEDIUM || 0,
            window.severityStats.LOW || 0
          ],
          backgroundColor: [
            '#f55855',
            '#f8a541',
            '#3b8ded',
            '#42d392'
          ],
          borderWidth: 4,
          borderColor: '#1a2236',
          hoverOffset: 10
        }]
      },
      options: {
        cutout: '65%', // size of center cut for doughnut
        plugins: {
          legend: { display: false }
        },
        // On slice click, go to filtered severity
        onClick: (evt, activeEls) => {
          if (activeEls && activeEls.length) {
            const chart = activeEls[0].element.$context.chart;
            const idx = activeEls[0].index;
            const label = chart.data.labels[idx];
            if (label) {
              window.location.href = '/vulnerabilities?
severity=' + label.toUpperCase();
            }
          }
        }
      }
    });
  }
// === Vendor Risk Analysis Radar Chart ===
if (document.getElementById('vendorRiskChart')) {

```

```

    const ctx =
document.getElementById('vendorRiskChart').getContext('2d');
    let topN = 10; // Default to viewing top 10 CWEs
    let weighted = false; // Weighted mode toggle

// Helper: get proper dataset depending on TopN & weighted toggle
function getRadarData() {
    let srcAll = window.cweRadarAll;
    let source = srcAll.top10 || srcAll['all'] || window.cweRadar ||
{
};

    if (topN === 5 && srcAll.top5) source = srcAll.top5;
    else if (topN === 10 && srcAll.top10) source = srcAll.top10;
    else if (topN === 'all' && srcAll.all) source = srcAll.all;
    if (weighted && window.cweRadarWeighted &&
window.cweRadarWeighted.indices) {
        let srcW = window.cweRadarWeighted;
        let codes = srcW.indices, names = srcW.labels, values =
srcW.values;

        if (topN !== 'all' && values.length > topN) {
            codes = codes.slice(0, topN);
            names = names.slice(0, topN);
            values = values.slice(0, topN);
        }
        return { codes, names, values };
    }
    let codes = source.indices || [];
    let names = source.labels || [];
    let values = source.values || [];
    if (topN !== 'all' && values.length > topN) {
        codes = codes.slice(0, topN);
        names = names.slice(0, topN);
        values = values.slice(0, topN);
    }
    return { codes, names, values };
}

// Tooltip explains: CWE code, human name, count, definition,
mitigation, link
function radarTooltip(context) {
    const code = context.label;
    const name = context.dataset.meta.names ?
        context.dataset.meta.names[context.dataIndex] : "";
    const val = context.dataset.data[context.dataIndex];
    let def = "";
    if (window.cweRadarDescriptions &&
window.cweRadarDescriptions[code]) {

```

```

        def = window.cweRadarDescriptions[code];
    }
    let mitig = "";
    if (window.cweMitigations && window.cweMitigations[code]) {
        mitig = "Mitigation: " + window.cweMitigations[code];
    }
    return [
        `CWE: ${code}`,
        `Name: ${name}`,
        `Number of CVEs: ${val}`,
        ... (def ? [def] : []),
        ... (mitig ? [mitig] : []),
        `Learn more:
https://cwe.mitre.org/data/definitions/\${code.replace\('CWE-', ''\)}.html`
    ];
}

//Redraw the radar chart (for initial draw + filter changes)
function drawRadar() {
    const { codes, names, values } = getRadarData();
    ctx.clearRect(0, 0, ctx.canvas.width, ctx.canvas.height);
    if (window.radarChartObj && window.radarChartObj.destroy)
        window.radarChartObj.destroy();
    window.radarChartObj = new Chart(ctx, {
        type: 'radar',
        data: {
            labels: codes,
            datasets: [{
                label: 'Vulnerability Frequency',
                data: values,
                backgroundColor: 'rgba(99, 164, 255, 0.2)',
                borderColor: '#63a4ff',
                pointBackgroundColor: '#63a4ff',
                pointBorderColor: '#fff',
                pointHoverRadius: 6,
                pointRadius: 4,
                pointHitRadius: 21,
                meta: { names: names }
            }]
        },
        options: {
            responsive: true,
            maintainAspectRatio: false,
            plugins: {
                legend: { display: false },
                tooltip: {

```

```

        callbacks: { label: radarTooltip }
    },
    scales: {
        r: {
            angleLines: { color: 'rgba(255,255,255,0.10)' },
            grid: { color: 'rgba(255,255,255,0.10)' },
            pointLabels: {
                color: '#a9adc1',
                font: { size: 13, weight: 'bold' }
            },
            beginAtZero: true,
            min: 0,
            ticks: { display: false }
        }
    },
    //Click node, deep-link to filtered vulnerabilities
    onClick: (evt, activeEls) => {
        if (activeEls && activeEls.length) {
            const chart = activeEls[0].element.$context.chart;
            const idx = activeEls[0].index;
            const code = chart.data.labels[idx];
            if (code) {
                window.location.href = "/vulnerabilities?q=" +
encodeURIComponent(code);
            }
        }
    }
});
});

// Change Top N radar filter
document.getElementById('radarCweCount').onchange = function() {
    topN = this.value === "all" ? "all" : parseInt(this.value, 10);
    drawRadar();
};

// Toggle weighted radar view
document.getElementById('radarWeighted').onchange = function() {
    weighted = this.checked;
    drawRadar();
};
drawRadar();

// === Info Popover for legend ===

```



```

const infoIcon = document.getElementById('legendInfoIcon');
const popover = document.getElementById('legendInfoPopover');
if (infoIcon && popover) {
    function showPopover(){
        popover.style.display = "block";
    }
    function hidePopover(){
        popover.style.display = "none";
    }
    infoIcon.addEventListener("click", showPopover);
    infoIcon.addEventListener("mouseenter", showPopover);
    infoIcon.addEventListener("focus", showPopover);
    infoIcon.addEventListener("blur", hidePopover);
    infoIcon.addEventListener("mouseleave", hidePopover);
}
}

// === Severity card click logic ===
document.querySelectorAll(".severity-card").forEach(function(card) {
    card.addEventListener("click", function() {
        var sev = card.getAttribute("data-severity");
        if (sev) {
            window.location.href = "/vulnerabilities?severity=" +
encodeURIComponent(sev);
        }
    });
    card.addEventListener("mouseenter", function() {
        card.style.boxShadow = "0 0 0 3px #63a4ff44";
    });
    card.addEventListener("mouseleave", function() {
        card.style.boxShadow = "";
    });
});
});
});

```