Stored XSS in graph tooltip from entity name

High **bramkragten** published **GHSA-mq77-rv97-285m** on Oct 14, 2025

Package

Home Assistant

Affected versions

2025.1.0 to 2025.10.1

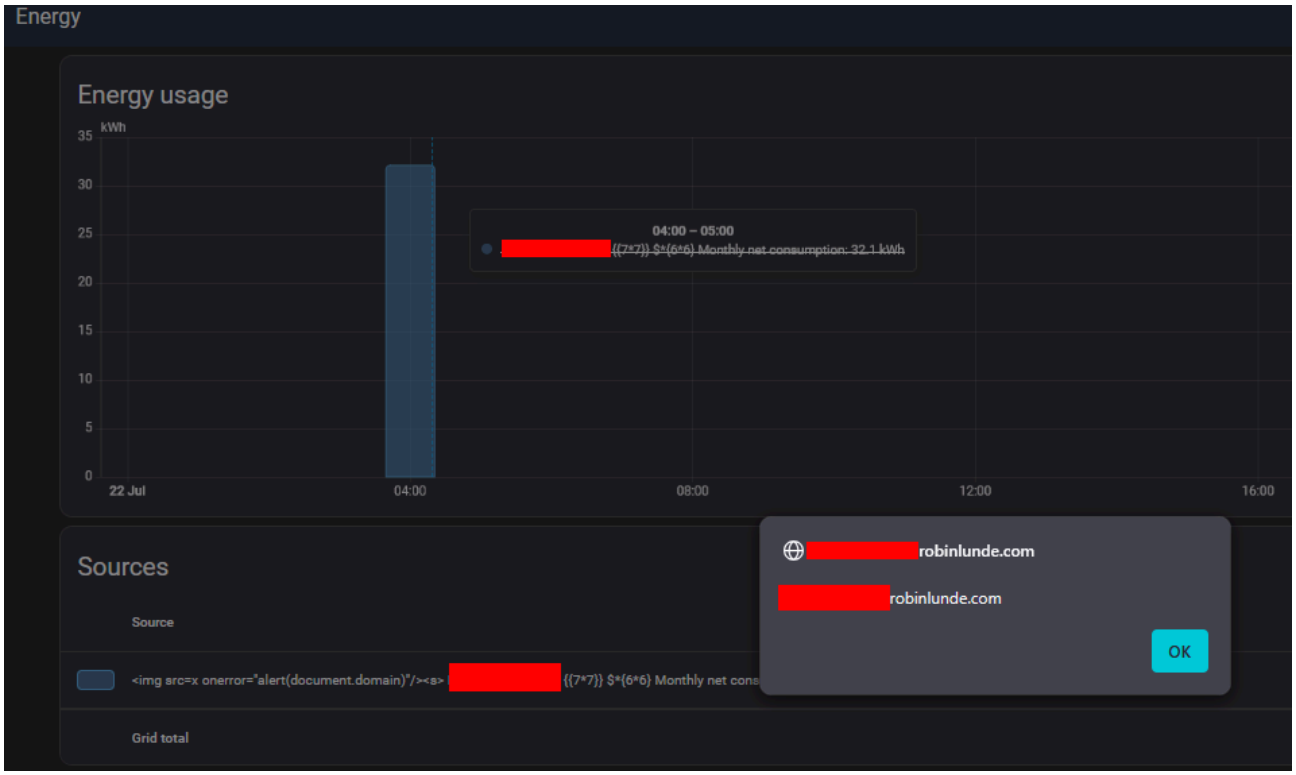
Patched versions

2025.10.2 and newer

Description

Summary

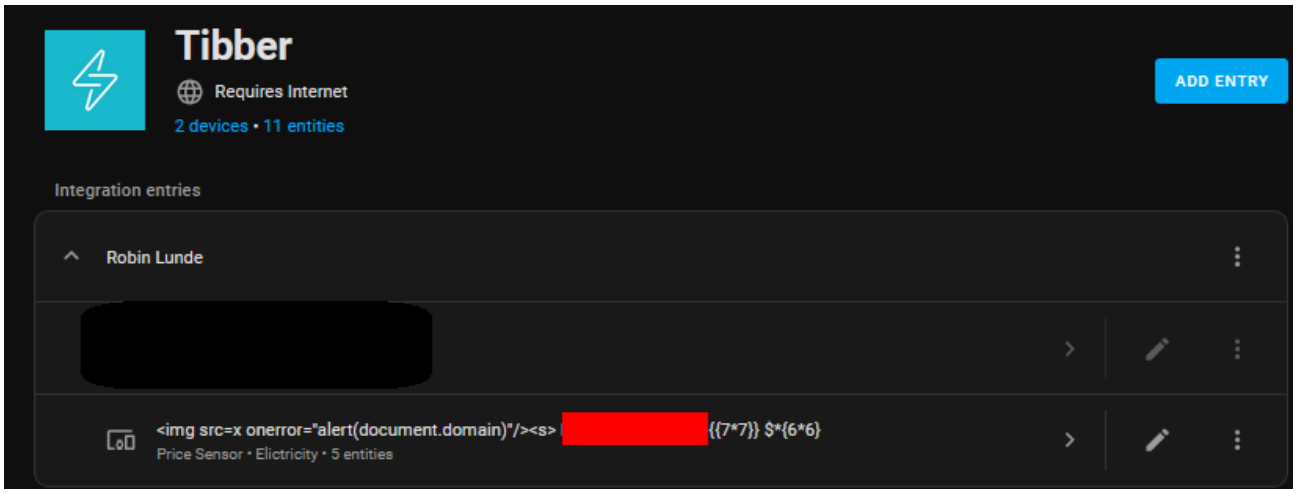
An authenticated party can add a malicious name to the Energy entity, allowing for Cross-Site Scripting attacks against anyone who can see the Energy dashboard, when they hover over any information point (The blue bar in the picture below)



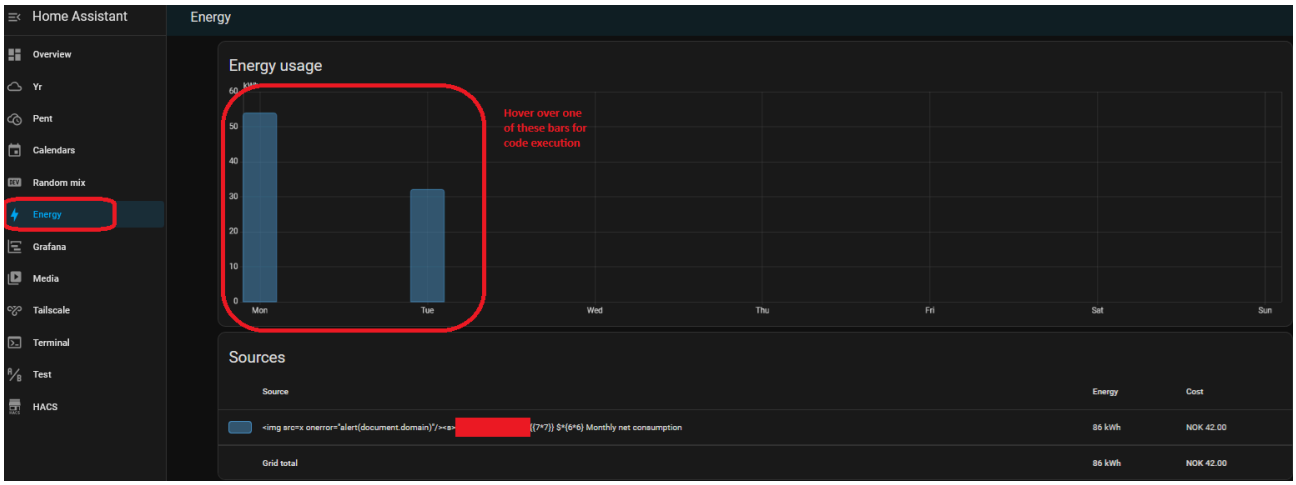
An alternative, and more impactful scenario, is that the entity gets a malicious name from the provider of the Entity (in this case the energy provider: Tibber), and gets exploited that way, through the default name.

Details

The incriminating entity in my scenario is from the Tibber integration, as shown in the screenshot below:

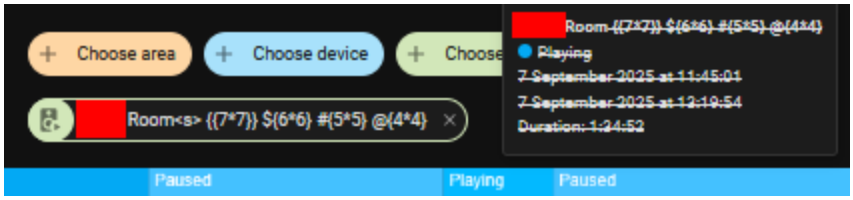


The exploit should be possible regardless of the Energy integration, as the user can name the entity themselves and as such pick a malicious name. The default name given by the Energy integration can also be taken directly from their system, and be vulnerable that way. The execution happens within the energy dashboard, when hovering over a data point:



Update found after issue was reported:

I found that the issue presents itself for any entity with a html-entity in the name, which is included and rendered in the graph view. Following is an example for a speaker:



Source code

The relevant source code is added in a comment, but copy pasted here as well:

The offending line of code rendering the payload appears to be:
https://github.com/home-assistant/frontend/blob/c13a80ce5e7ae39f0262444e2b6295a074a96732/src/panels/lovelace/cards/energy/hui-energy-devices-graph-card.ts#L110
Where the parameter marked with bold and italic is the vulnerable value:
return `${title}${params.marker} ***${params.seriesName}***: ${value}`;

From the trace below, we can see that the only change done to the friendly_name of an entity is replacing underscores with spaces (*computeObjectId(entityId).replace(/*/g, " ")_).
We can also determine that any power entity will have it's name used if there is one, and fall back to the friendly name if it cannot find one:

```
data.push({
  id: `${compare ? "compare-" : ""}${statId}-${type}`,
  type: "bar",
  cursor: "default",
  name:
    type in labels
      ? labels[type]
      : getStatisticLabel(
          this.hass,
          statId,
          statisticsMetaData[statId]
        ),
```

(https://github.com/home-assistant/frontend/blob/c13a80ce5e7ae39f0262444e2b6295a074a96732/src/panels/lovelace/cards/energy/hui-energy-usage-graph-card.ts#L467-L478)

The value comes from:

1. https://github.com/home-assistant/frontend/blob/c13a80ce5e7ae39f0262444e2b6295a074a96732/src/panels/lovelace/cards/energy/hui-energy-usage-graph-card.ts#L467-L478
   (This is the relevant call: *getStatisticLabel(this.hass, statId, statisticsMetaData[statId];*)

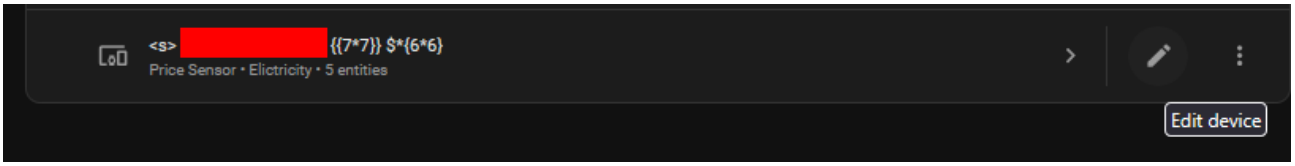
2. getStatisticLabel is defined here: https://github.com/home-assistant/frontend/blob/c13a80ce5e7ae39f0262444e2b6295a074a96732/src/data/recorder.ts#L329-L339 (This is the relevant call: *computeStateName(entity);*)

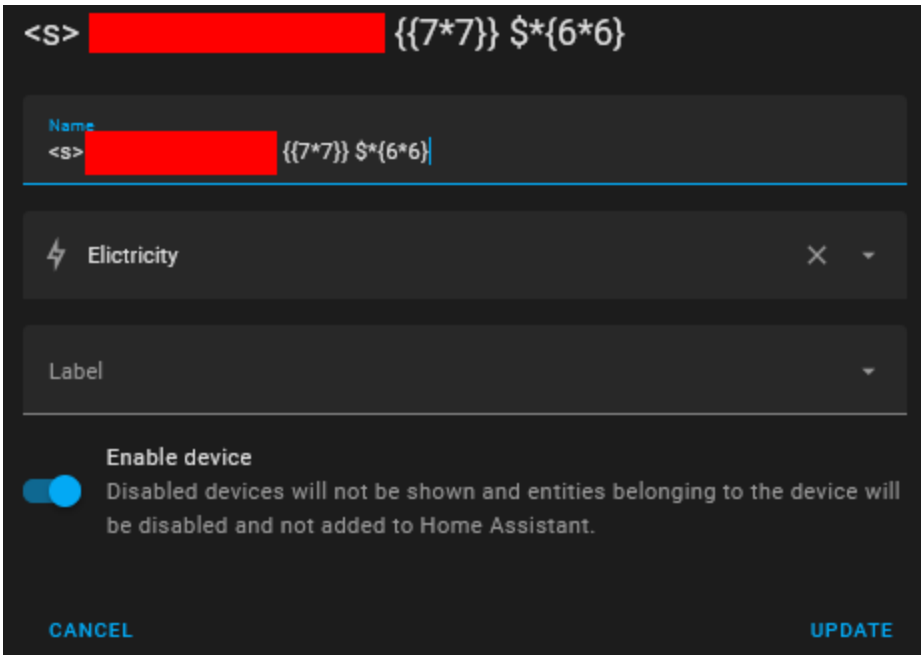
3. computeStateName is defined here: https://github.com/home-assistant/frontend/blob/c13a80ce5e7ae39f0262444e2b6295a074a96732/src/common/entity/compute_state_name.ts

---

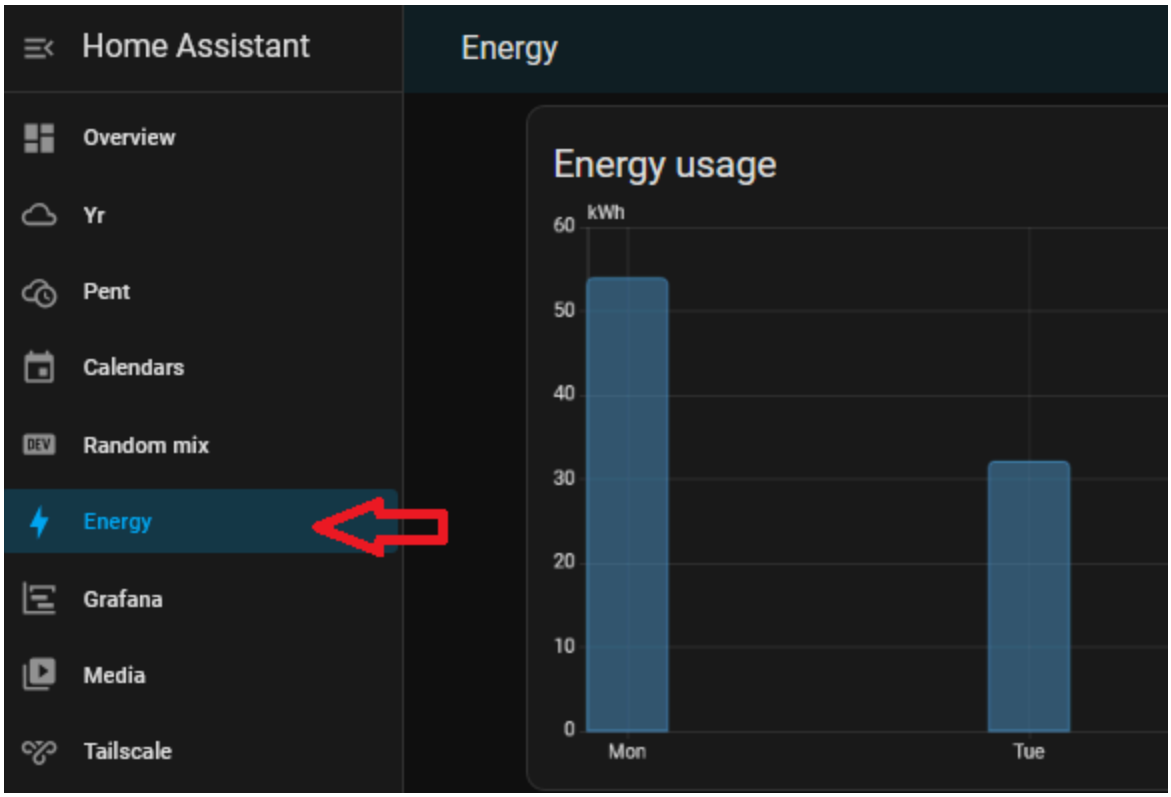
PoC

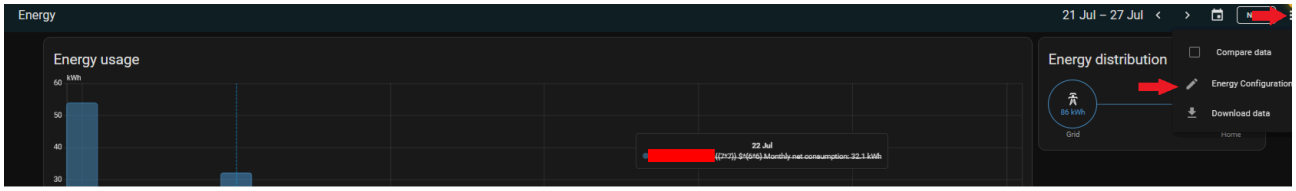
1. Set up a new energy provider with a price sensor.
2. Give the price sensor a malicious name

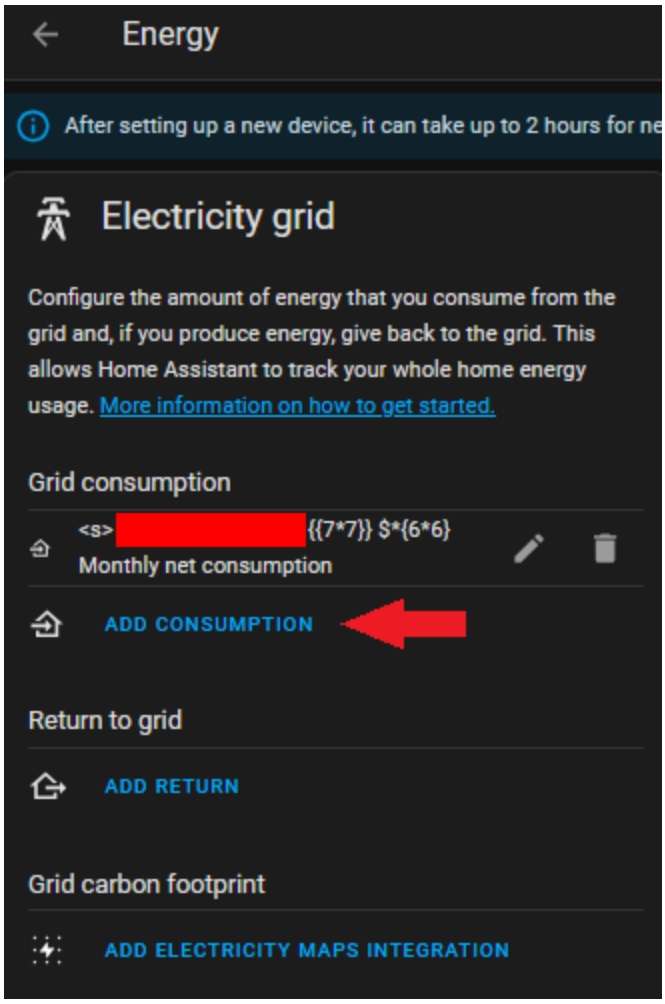




3. Configure the energy dashboard to get data from the price sensor


Home Assistant
Energy
Overview
Yr
Pent
Calendars
Random mix
Energy
Grafana
Media
Tailscale
Energy usage
kWh
60
50
40
30
20
10
0
Mon
Tue


Energy
21 Jul – 27 Jul
Energy usage
Energy distribution
Compare data
Energy Configuration
Download data



5. Look at the data and hover the data point for code to execute. (You may have to trigger data ingestion or add a false data point to be able to hover a data point when testing, you need at least one datapoint to trigger the vulnerability)

Impact

It is possible to exploit this over the internet, by using an energy provider, like Tibber, with a malicious name, and relying on the default naming in Home Assistant being used. This is actually how I found this bug:

This means that a malicious employee or someone with access to your electricity provider can attack your Home Assistant instance from your electricity provider. I am unsure if you consider this a sanitization and escaping issue in the respective integrations or not, but I believe a central fix in the form of fixing the Energy Dashboard is more appropriate, rather than to rely on every integration properly handling user input.

Severity

High

CVE ID

CVE-2025-62172

Weaknesses

▶ CWE-80

Credits

pwnpanda Reporter