Host header injection leads to SSRF via resource-js endpoint

Moderate **angrybrad** published **GHSA-95wr-3f2v-v2wh** last week

Package

php **craftcms/cms** (Composer)

Affected versions	Patched versions
>= 5.0.0-RC1, <= 5.9.14	5.9.15
>= 4.0.0-RC1, <= 4.17.8	4.17.9

Description

Summary

The `resource-js` endpoint in Craft CMS allows unauthenticated requests to proxy remote JavaScript resources.
When `trustedHosts` is not explicitly restricted (default configuration), the application trusts the client-supplied Host header.

This allows an attacker to control the derived `baseUrl`, which is used in prefix validation inside `actionResourceJs()`.
By supplying a malicious Host header, the attacker can make the server issue arbitrary HTTP requests, leading to Server-Side Request Forgery (SSRF).

Details

The vulnerability exists in `AppController::actionResourceJs()`.

The function validates that the `url` parameter starts with `assetManager->baseUrl`. However, `baseUrl` is derived from the current request host. If `trustedHosts` is not configured, the Host header is fully attacker-controlled.

Attack chain:

1. Attacker sends request with controlled `Host` header.

2. Application derives `baseUrl` from the malicious Host.
3. `url` parameter is required to start with this `baseUrl`.
4. Validation passes.
5. Guzzle performs a server-side HTTP request to the attacker-controlled host.
6. SSRF occurs.

This does not rely on string parsing bypass. It relies on Host header trust.

PoC (safe reproduction steps)

Environment:

- Craft CMS 5.9.12
- Default configuration (no trustedHosts restriction)
- Docker deployment

1. Start a listener inside the container:
   python3 -m http.server 9999
2. Send a request to resource-js with a controlled Host header.
3. Observe that the internal listener receives a request (OOB confirmation).

References

`ebe7e85`

Severity

Moderate

CVE ID

CVE-2026-41130

Weaknesses

No CWEs

Credits

HuajiHD Reporter