# 46 Rack parses encoded cookie names allowing an attacker to send malicious `__Host-` and `__Secure-` prefixed cookies

Share:

TIMELINE

jetchto99 submitted a report to Ruby on Rails. Jun 10th (3 ye

The rack cookie parser parses the cookie string using `unescape`. This allows a malicious attacker to set a second cookie with the name being percent encoded. Typically it would be expected that we cannot trust cookies and in *most* cases that's true. However in a couple of cases certain expectations are set. Cookies allow cookie prefixes on the cookie name to indicate to the browser certain attributes. In this case there are 2 special attributes we care about: `__Secure-` and `__Host-`. When the browser sends these cookies to the server certain assumptions are be made around these cookies:

1. `__Secure-` prefix: Cookies names starting with `__Secure-` (dash is part of the prefix) must be set with the secure flag from a secure page (HTTPS).
2. `__Host-` prefix: Cookies with names starting with `__Host-` must be set with the secure flag, must be from a secure page (HTTPS), must not have a domain specified (and therefore aren't sent to subdomains) and the path must be `/`

The flaw in Rack allows for a `__%48ost-` or `__%53ecure-` cookie to be set **without** the required attributes (I.e. set without HTTPS, from root domain, or from a sec page). This means a malicious cookie set by an attacker could set a `__%48ost-` cookie from a subdomain knowing that Rack would parse it as `__Host-`. Furtherm since the browser won't enforce the `HostOnly` attribute to `__%48ost-` cookies an attacker could control the `__Host-` prefixed cookie from a subdomain by settir wildcard domain on the `__%48ost-` cookie.

It should be noted that while the cookie spec recommends encoding for the value of a cookie it doesn't make any suggestions around the encoding of the name of cookie.

Here's a simple PoC test case which fails :

**Code** 1.34 KiB Wrap lines Copy Dow

```
# frozen_string_literal: true

require_relative 'helper'

describe Rack::Utils, "malicious cookie" do
  # Fails and __Host-evil reads the malicious value and sets it as the cookie
  # rather than reading the actual __Host cookie
  #
  # Furthermore, browsers enforce HostOnly for `__Host-` cookies but they would
  # not enforce it for "__%48ost" cookies so a malicious script could potentially
  # set this cookie knowing it would be parsed as the `__Host-` cookie
  #
  # Lastly, when the cookie is made it could be set with the `.example.com` domain
  # wildcard, thus a malicious script on a subdomain could set the cookie and it
  # would be parsed by the root domain
  #
  # This is due to the cookie being unescaped, thus:
  # URI.unescape("__%48ost-evil") => "__Host-evil"
  #
  # Currently fails, should be passing
  it "doesnt parse malicious __Host cookie" do
    env = Rack::MockRequest.env_for("", "HTTP_COOKIE" => "__%48ost-evil=evil;__Host-evil=abc")
    cookies = Rack::Utils.parse_cookies(env)
    cookies.must_equal({ "__%48ost-evil" => "evil", "__Host-evil" => "abc"  })
  end

  # Less of a security issue and more of a bug
  it "generic foo=bar example" do
    env = Rack::MockRequest.env_for("", "HTTP_COOKIE" => "%66oo=baz;foo=bar")
    cookies = Rack::Utils.parse_cookies(env)
    cookies.must_equal({ "%66oo" => "baz", "foo" => "bar" })
  end
end
```

An attacker could potentially set the cookie from a malicious script on a subdomain like so, bypassing any expectations around the attributes of the cookie:

**Code** 60 Bytes Wrap lines Copy Dow

```
document.cookie = "__%48ost-evil=evil; domain=.example.com";
```

I should note I work for GitHub, I'm not sure if there's any conflict with payouts in this case (and I certainly don't want/need a payout), however should you chose to payout for this I'd like the money to be donated to charity. If possible could it please be donated to NAACP Legal Defense and Education Fund their donaiton page be found here.

### Impact

An attacker can control cookies by encoding creating a second cookie with the name url encoded. This means that the `__Host-` and `__Secure-` prefixed cookies be controlled. Furthermore, a malicious attacker could set this cookie from a subdomain and have it apply to the root domain, in which case the Rack would parse t attackers cookie.

jetchto99 posted a comment. Jun 10th (3 ye

A `<cookie-name>` can be any US-ASCII characters, except control characters, spaces, or tabs. It also must not contain a separator character like the following: ( ) @ , ; : \ " / [ ] ? = { }.

fletchto99 posted a comment. Jun 11th (3 ye

I had a go at attempting to patch the issue today. I'm not sure if it's the most performant solution but I figured I'd help out where possible. I'm not too sure what's required in terms of backports but hopefully this can help kickstart a fix.

1 attachment:
**F864477:** fix.patch

tenderlove Ruby on Rails staff posted a comment. Jun 12th (3 ye

@fletchto99 what do you think about this as a patch? It's based off of your patch, but passes the test suite and should allocate fewer objects. If you think it will wo can you send me a patch using `git format-patch` so I can retain your contribution information?

1 attachment:
**F866010:** out.diff

fletchto99 posted a comment. Jun 12th (3 ye

Hey @tenderlove, here's a patch file created via `format-patch`. Feel free to touch up the commit message if rack follows any guidelines around commit messages

Cheers!

1 attachment:
**F866020:** 0001-When-parsing-cookies-only-decode-the-values.patch

jack_mccracken posted a comment. Jun 12th (3 ye

Hey @fletchto99,

I've prepared the following advisory and final patchset. Let me know if you spot any errors.

3 attachments:
**F866114:** 0001-2.1.3-When-parsing-cookies-only-decode-the-values.patch
**F866115:** 0001-2.2.2-When-parsing-cookies-only-decode-the-values.patch
**F866122:** 895727_advisory.txt

fletchto99 posted a comment. Jun 12th (3 ye

Hey Jack. The advisory looks good. I just noticed I used my GitHub email for the patches. I've updated them to use my personal email. Besides that everything loo good, thanks!

2 attachments:
**F866148:** 0001-2.1.3-When-parsing-cookies-only-decode-the-values.patch
**F866149:** 0001-2.2.2-When-parsing-cookies-only-decode-the-values.patch

tenderlove Ruby on Rails staff updated the severity to Low. Jun 15th (3 ye

tenderlove Ruby on Rails staff closed the report and changed the status to **Resolved**. Jun 15th (3 ye

This is shipped, thanks!

fletchto99 requested to disclose this report. Jun 15th (3 ye

Thanks @tenderlove and @jack_mccracken, unless there's any reservations any issues with disclosing?

jack_mccracken agreed to disclose this report. Jun 16th (3 ye

This report has been disclosed. Jun 16th (3 ye

The Internet Bug Bounty has decided that this report is not eligible for a bounty. Feb 9th (2 ye

Hi @fletchto99. No problems disclosing this, I just have completely forgotten to do that. Also since this is a low severity bug I'm not awarding a bounty (as that is w our bounty rules say).