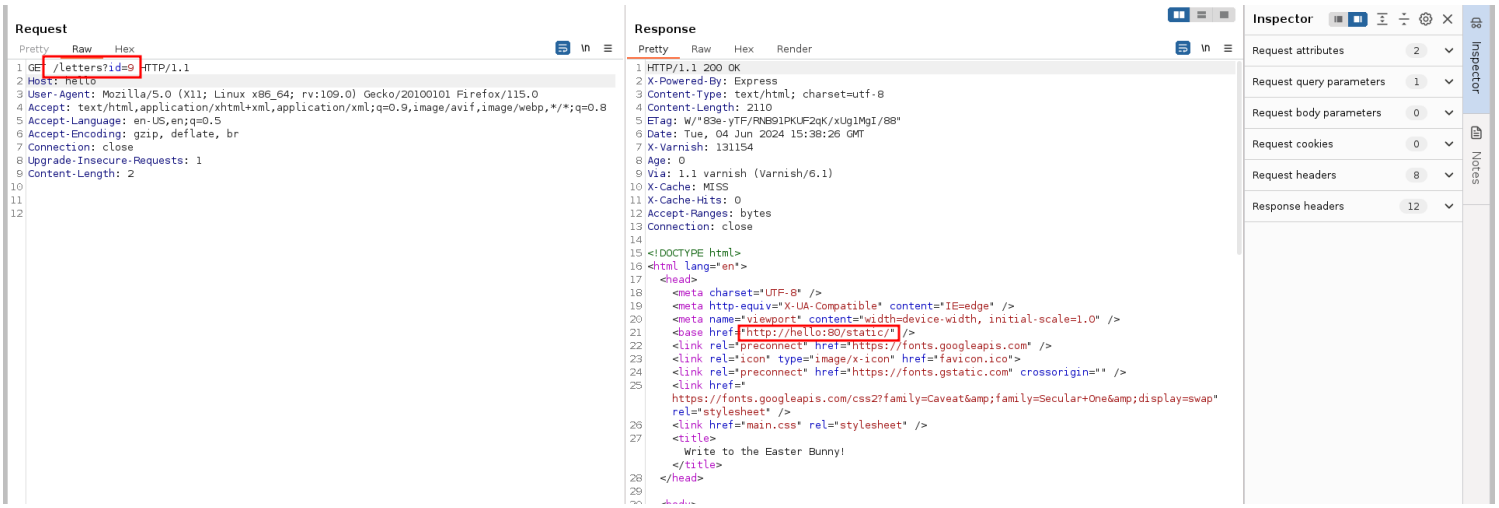
EasterBunny

1) We have a injection here




```
router.get("/", (req, res) => {
    return res.render("index.html", {
        cdn: `${req.protocol}://${req.hostname}:${req.headers["x-forwarded-port"] ?? 80}/static/`,
    });
});

router.get("/letters", (req, res) => {
    return res.render("viewletters.html", {
        cdn: `${req.protocol}://${req.hostname}:${req.headers["x-forwarded-port"] ?? 80}/static/`,
    });
});
```

HTML <base> Tag

‹ Previous | Complete HTML Reference | Next ›

Example

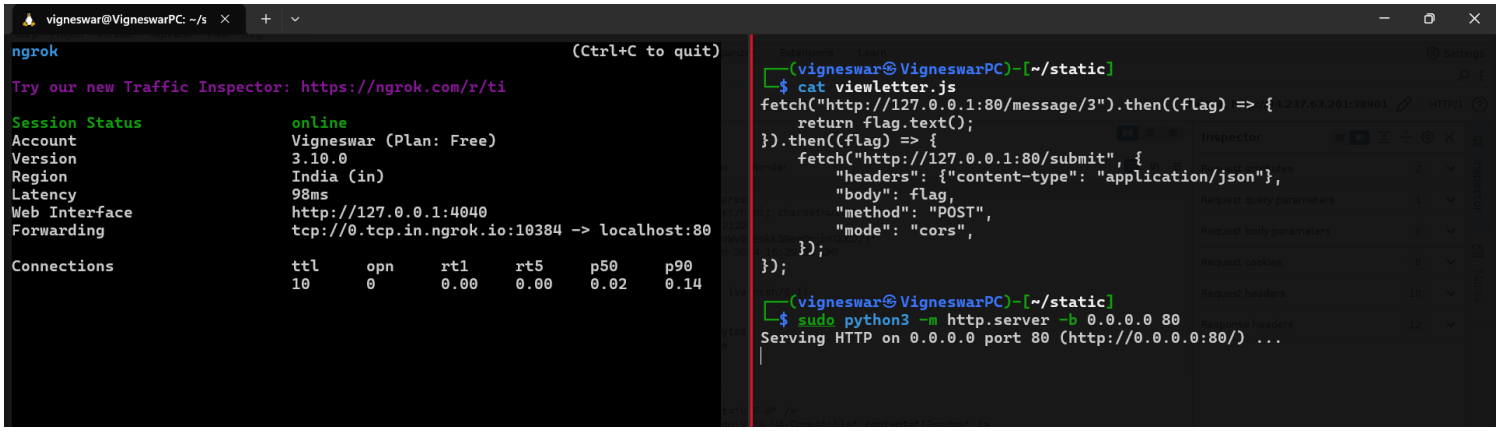
Specify a default URL and a default target for all links on a page:

```
<head>
  <base href="https://www.w3schools.com/" target="_blank">
</head>

<body>
<img src="images/stickman.gif" width="24" height="39" alt="Stickman">
<a href="tags/tag_base.asp">HTML base Tag</a>
</body>
```

Try it Yourself »

We can control the static files and load our xss payload and cache it

3) Made payload to get flag



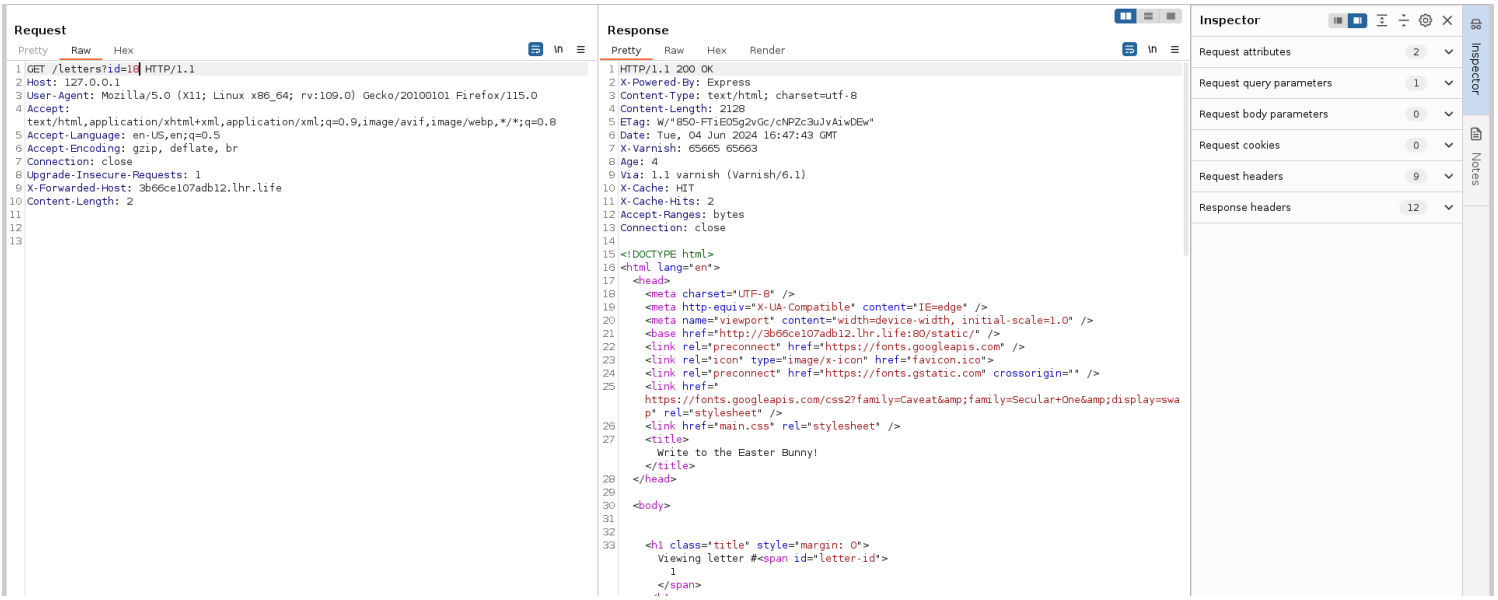
4) Cache poisoning

```
sub vcl_recv {
    set req.http.X-Forwarded-URL = req.url;
    set req.http.X-Forwarded-Proto = "http";
    if( req.http.host ~ ":[0-9]+" )
    {
        set req.http.X-Forwarded-Port = regsub(req.http.host, ".*:", "");
    }
    else
    {
        set req.http.X-Forwarded-Port = "80";
    }

    if ( !( req.url ~ "^/message") ) {
        unset req.http.Cookie;
    }
}
```

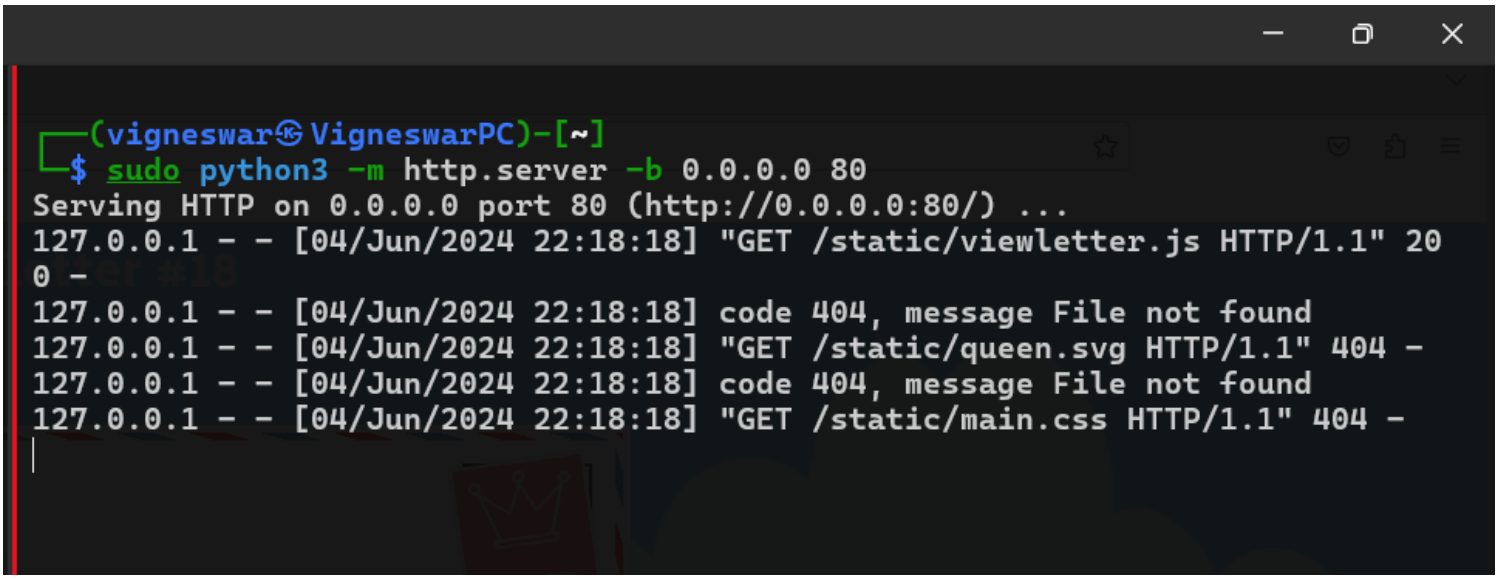
```
sub vcl_hash {
    hash_data(req.url);

    if (req.http.host) {
        hash_data(req.http.host);
    } else {
        hash_data(server.ip);
    }

    return (lookup);
}
```

5) Hosted the page

6) Poisoned the cache



7) Triggered bot



8) Got the flag