enferas / header_injection_phpipam.md

Created 2 months ago

☆ Star

<> Code -o- Revisions 1

Header injection (SSRF) vulnerability in phpipam

header_injection_phpipam.md

Header injection vulnerability in phpipam https://github.com/phpipam/phpipam version v1.5.0

The path of the vulnerability:

```
<?php
//In file https://github.com/phpipam/phpipam/blob/master/app/admin/subnets/ripe-quer
//line 21
// the source is $_POST['subnet']
$res = $Subnets->resolve_ripe_arin ($_POST['subnet']);

//In file https://github.com/phpipam/phpipam/blob/master/functions/classes/class.Sub
//line 3523
public function resolve_ripe_arin ($subnet) {
    // ...
    // Note: We can bypass the check by choosing the value in this format
    // [the correct value for $subnet_check][.][injection value]
    //so reset will take the first value of tje explode and the condition will be tr
    // take only first bit of ip address to match /8 delegations
    $subnet_check = reset(explode(".", $subnet));
    // ripe or arin?
    if (in_array($subnet_check, $this->ripe)){
        // the injection in $subnet
        return $this->query_ripe ($subnet);
    }
    //...
}

// In file https://github.com/phpipam/phpipam/blob/master/functions/classes/class.Su
// line 3545
private function query_ripe ($subnet) {
    // ripe_arin_fetch method will be called
    $ripe_result = $this->identify_address ($subnet)=="IPv4" ? $this->ripe_arin_fetc
    // ...
}
// In file https://github.com/phpipam/phpipam/blob/master/functions/classes/class.Su
// line 3633
private function ripe_arin_fetch ($network, $type, $subnet) {
    // set url
    // $subnet is added to $url without sanitization
    // which can go backward in the directory ../../admin/
    $url = $network=="ripe" ? https://rest.db.ripe.net/ripe/$type/$subnet : https://

    $result = $this->curl_fetch_url($url, ["Accept: application/json"]);

    $result['result'] = json_decode($result['result']);

    // result
    return $result;
}

// In file https://github.com/phpipam/phpipam/blob/master/functions/classes/class.Su
// line 1443
// the execution for the curl
public function curl_fetch_url($url, $headers=false, $timeout=30) {
    $result = ['result'=>false, 'result_code'=>503, 'error_msg'=>''];

    //...

    try {
        $curl = curl_init();
        // Note: $url is not sanitized
        curl_setopt($curl, CURLOPT_URL, $url);
        //....

        $result['result']      = curl_exec($curl);
        $result['result_code'] = curl_getinfo($curl, CURLINFO_HTTP_CODE);
        $result['error_msg']   = curl_error($curl);

        // close
        curl_close ($curl);

    } catch (Exception $e) {
        $result['error_msg'] = $e->getMessage();
    }

    return $result;
}
```

The developers were informed of the report by sending an email on 19/06/2022.

enferas commented on Oct 3 Author

CVE-2022-41443 is assigned to this discovery.