Private key "tron" leaked via Travis CI Log

Share:

State	○ Resolved (Closed)
Disclosed	**May 26, 2019 11:21pm +0530**
Reported To	Tron Foundation
Asset	https://github.com/tronprotocol/java-tron (Domain)
Weakness	Information Disclosure
Bounty	$1,000
Severity	Critical (9 ~ 10)
Participants	
Visibility	Disclosed (Full)

Collapse

SUMMARY BY RHYNORATER

While assessing the Tron Foundation's build logs on Travis-CI, it was discovered that they had leaked the SSH private key that they were using to push their code to a development asset.

The exposure can be found here: https://travis-ci.org/tronprotocol/java-tron/builds/361945077#L4101
The script which uses this key to log into development servers can be found here: https://github.com/tronprotocol/java-tron/blob/develop/deploy.sh
Paper on the methodology: https://edoverflow.com/2019/ci-knew-there-would-be-bugs-here/

This vulnerability occurred because of this line `9` of `.travis.yml` found in this commit: https://github.com/tronprotocol/java-tron/commit/ce15ea39adf28ebdd686a56cacf482772af2baf0

When Travis-CI ran this code, it converted the `[secure]` variables defined here into a SSH secret key outputted as `tron`. Because the line `cat tron` was added, this private key was outputted into the job log. From there, one can simply retrieve the secret key from the job log and log into the servers mentioned in `deploy.sh`.

Methodology

The following methodology was used to find this vulnerability:

1. List all the repos under `/tronprotocol` on travis-ci.org
2. For each of these repos, grab each of the builds
3. For each of the builds, grab the config and the job log(s)
4. Grep through the job logs and config for sensitive looking strings (see trufflehog regex and work by ed and karim)

Bounty decision

I disagree with the decided impact for this report. I was not permitted to attempt to escalate via metadata endpoints or look for GitHub keys on the test assets.
Show more

TIMELINE

rhynorater submitted a report to **Tron Foundation**. Dec 27th (5 months ago)

Hey Tron team,

It appears that via your use of secured variables in Travis-CI, you do not want the content of the `tron` private key released. However, it appears that it was leaked in one of your logs. I generated a fingerprint and scanned the internet, but couldn't find any open servers with the SSH key associated with this private key. However, I figured I'd let you know. I have placed this as critical as there is a high possibility that this could be critical, however, if this is not defined as so, I would appreciate the opportunity to self-close.

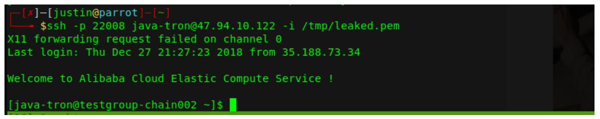
You can find the leakage at the bottom of this Travis-CI log: https://travis-ci.org/tronprotocol/java-tron/builds/361945077

Impact

Private key leakage

rhynorater posted a comment. Dec 27th (5 months ago)

Just confirmed that this leads to code execution:



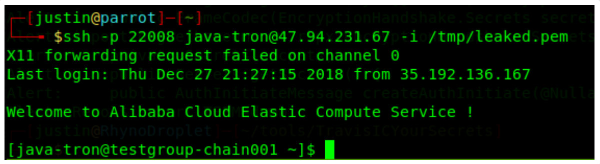
```
┌─[x]─[justin@parrot]─[~]
└──╼ $ssh -p 22008 java-tron@47.94.10.122 -i /tmp/leaked.pem
X11 forwarding request failed on channel 0
Last login: Thu Dec 27 21:27:23 2018 from 35.188.73.34

Welcome to Alibaba Cloud Elastic Compute Service !

[java-tron@testgroup-chain002 ~]$
```

1 attachment:

F397953: Leaked.png

rhynorater posted a comment. Updated Dec 27th (5 months ago)

It appears that this allows me to log into both `47.94.231.67` and `47.94.10.122`. I did nothing but login to test the keys. No commands were run.



```
┌─[justin@parrot]─[~]
└──╼ $ssh -p 22008 java-tron@47.94.231.67 -i /tmp/leaked.pem
X11 forwarding request failed on channel 0
Last login: Thu Dec 27 21:27:15 2018 from 35.192.136.167

Welcome to Alibaba Cloud Elastic Compute Service !

[java-tron@testgroup-chain001 ~]$
```

1 attachment:

F397954: Leaked2.png

rhynorater posted a comment. Dec 27th (5 months ago)

I found the servers to log into here: https://github.com/tronprotocol/java-tron/blob/develop/deploy.sh

lasagna HackerOne staff changed the status to ○ **Triaged**. Dec 28th (5 months ago)

Hello @rhynorater,

Thank you for your submission! We were able to validate your report, and we have submitted it to the appropriate remediation team for review. They will let us know the final ruling on this report, and if/when a fix will be implemented. Please note that the status and severity are subject to change.

Regards,
@lasagna

rhynorater posted a comment. Dec 29th (5 months ago)

Just as a note, my IP address is 71.176.211.78 just in case you want to vet the IPs that used the private key to authenticate.

rhynorater posted a comment. Dec 29th (5 months ago)

This is likely even more critical than just an RCE on these 2 servers due to these endpoints: https://www.alibabacloud.com/help/doc-detail/49122.htm

An attacker could have compromised every server associated with your Alibaba cloud infrastructure. I'd be happy to see if I can escalate if permitted to do so.

taiyangc posted a comment. Dec 30th (5 months ago)

@rhynorater Could you demonstrate how to attack other servers based on metadata? If not this is not that high of a priority. Thank you!

rhynorater posted a comment. Dec 30th (5 months ago)

@taiyangc I could potentially pivot to other servers using the `ram/security-credentials` endpoint in the metadata instance.

If you are not going to consider this a `critical` I think it is only fair that you allow me to access the server again to attempt to escalate. The reason why I didn't go any further once I was already in was to respect the integrity of your system.

Tron Foundation rewarded rhynorater with a **$1,000** bounty. Jan 4th (5 months ago)

Thanks for discovering this leakage. The leaked instances are part of a testing group for java-tron so no attack on TRON Foundation and/or TRON blockchain is possible. We initially would set a low severity but having you potentially accessing other machines (which are not part of the TRON blockchain) is not desired. Setting to a higher bounty for your discovery effort. Thank you for the respect.

nobodyatall closed the report and changed the status to ○ **Resolved**. Jan 4th (5 months ago)

Thanks.

rhynorater posted a comment. Jan 4th (5 months ago)

Not trying to be a pain, and I'll only comment once, but the lower threshold for a medium vulnerability is 3000. I feel like the total compromise of 2 test servers should atleast get the max for a low vulnerability. Nonetheless, I appreciate the bounty and will continue to search on your program. @nobodyatall @tronfoundation

nobodyatall posted a comment. Jan 4th (5 months ago)

@rhynorater $3000 is not the lower threshold but higher one.

rhynorater requested to disclose this report. Apr 26th (about 1 month ago)

rhynorater posted a comment. Apr 26th (about 1 month ago)

The technique used here has been released in a paper to the public. I'd like to disclose this as an example.

This report has been disclosed. May 26th (14 hrs ago)