Drogi uczestniku

Nasz zespół bezpieczeństwa dostał urządzenie typu embedded do pentestów i potrzebujemy Twojej pomocy!
Większość zespołu jest właśnie na urlopie więc ów pentest został przypisany Tobie.
Urządzenie, które należy przetestować znajduje się tu:
https://hitachipowergrids-my.sharepoint.com/:f:/g/personal/rafal_golebiowski_hitachi-powergrids_com/EmQ9AsGlqNZGobpirEKU8vsB3Yr0scRwl5GLqa8d4fqCDA?e=LZeib9

W przypadku problemów z adresem IP na urządzeniu, poniższy artykuł może być pomocny https://pentester.land/tips-n-tricks/2018/06/26/How-to-get-the-IP-address-of-a-downloaded-vulnerable-machine.html

Zespół produktowy, którego urządzenie trzeba przetestować zamieścił poniższy opis

We developed a complex xml based log aggregation service which runs on port 1234. Confidentiality and Availability is not so important to this service as the clients will just resend lost packets later and there is nothing very sensitive in the messages but the Integrity should be protected as the logs are needed for audit purposes.
The service expects xml messages something similar to this one:

```
<?xml version="1.0" encoding="UTF-8"?>
<exmpleLog>
  <id>asdf1234</id>
  <severity>Info</severity>
  <msg>This is an example log message to our new, safe, log aggregator service</msg>
</exmpleLog>
```

To protect confidentiality we use very secure 3072bit RSA signature. To sign the message first we base64 encode it than we generate it's hash and sign this hash.
The only thing is since it's an embedded device we decided for performance reasons to use CRC32 as a hash algorithm. We are aware that this is not a cryptographic hash function but what is the chance that there will be a hash collision when the message has to be valid base64 and after decoding valid xml?
Here is the signed message corresponding to the above xml file:


```
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPGV4bXBsZUxvZz4KICA8aWQ+YX
NkZjEyMzQ8L2lkPgogIDxzZXZlcml0eT5JbmZvPC9zZXZlcml0eT4KICA8bXNnPlRoaXMgaXMgYW4g
ZXhhbXBsZSBsb2cgbWVzc2FnZSB0byBvdXIgbmV3LCBzYWZlLCBsb2cgYWdncmVnYXRvciBzZXJ2
aWNlPC9tc2c+CjwvZXhtcGxlTG9nPgo=
193904909563637936949965655714001491547041811806577092443823316405109961240646770
677760107372154373765421154814631105438786872855042568191989505555331881846931596
541102177789125531091096584118551339366807335856916132395643300151881652841514876
734388419832450932009979244107901315283329459237824109767363785744286542518721987
542538486395809136042354682122759436659498513887963098854556361206130803211244142
213714559390242115226162360588305007725555307588577580801241741226867918055600222
363507301888849308039540602477701801599102669120257502801066411753758467422457164
068847035481923865391717593974903486625677719096490528677336788606248165170415 27
611698007056584460360075464234319288422268835701985459248853697595891221472400472
240068412462051295999608236810485924578119734378562491678393644955305710136087191
277172948254904075116211696327147644775074751933181836380929223143204056309541174
333575328472172237537432917303782 2
```


If you send this to the service (e.g. $ cat crypt.msg | nc -nv 192.168.198.128 1234) It should answer back that the signature is correct and echo back the message.
We have some sensitive secrets on the server, we would consider it a problem if someone could steal these:

- *If someone would find a different message with the same hash the server might send back sensitive information*

- *There is a secret stored in the home directory of the user running the service in a file called "flag.txt"*
- *The last secret is in the file /root/flag.txt but reading it would need root access*

Nasz Chief Pentester wskazał, że kryptografia może nie być idealna oraz wysłał mi poniższe linki, które mogą być pomocne przy rozpoczęciu testów:
https://sar.informatik.hu-berlin.de/research/publications/SAR-PR-2006-05/SAR-PR-2006-05_.pdf
https://www.nayuki.io/page/forcing-a-files-crc-to-any-value
Rekomenduje potraktować to jako punkt startowy.

Happy Hacking!

Odpowiedzi wysyłać prosimy tu:
pl-opendaysctf@hitachi-powergrids.com