Gmail **Mario Mercaldi <mario.mercaldi@gmail.com>**

Fwd: HASP HL Pro

Greg Cotton <greg@quasared.net> Sat, Mar 23, 2013 at 6:00 AM
To: mariomercaldi@gmail.com

Sent from my iPhone

Begin forwarded message:

> **From:** RCE Guru <rceguru@gmail.com>
> **Date:** March 23, 2013, 5:50:42 AM EDT
> **To:** Greg Cotton <greg@quasared.net>
> **Subject: Re: HASP HL Pro**
>
> On Sat, Mar 23, 2013 at 11:35 AM, Greg Cotton <greg@quasared.net> wrote:
>> The target application calls hasp_encrypt with a static 64 byte data set (corresponding to a specific asset) and the dongle returns a static value to the game every time which is our decryption key for the game asset. As far as I understand it there may be an AES key in the vendor specific library.
> True - there is aes key - is used for crypt comunication between api and dongle key (we call it VendorAes). There is 2 other aes keys for encrypt comunicate - before and after 3.25 firmware. Last api (5.10 and 5.12) implement WBaes based comunication - personally i not know how to extract real aes from wbaes tables.
> Real AES key for hasp_encrypt and hasp_decrypt is inside dongle - you cant see anywhere it, and cant modify.
> This is reason almost all emulators SRM is table based. Actually SRM have many aes - different for every enabled feature.
>
> So you have:
> Program -> VendorAES/WBaes -> hasplms ->2 different aes (before and after 3.25 firmware) -> driver -> dongle.
> All comunication crypted with different keys. Different functions - read, decrytp, write - use different keys.
> Also these keys generate some temporary keys every request program to dongle.
>
> You can understand this well if you debug hasplms.exe from driver - 5.25. haspdinst.exe with size 5 368 720.
> Inside is all SRM api + aes keys for emulate Demo dongle. Just create HASP SL key and debug. You will find almost all algorithm inside.
>
>> Additionally, if you hasp_encrypt something using the demo program and then hasp_decrypt that same something it returns garbled.
> This i not understand.
>
>> Can you shed some light on this? I think we have modified the client library wrong. The game is statically compiled with the library so we can't simply use that instead to communicate with the dongle. Basically we have a list of 3,000 or so keys that need to be run through the dongle.
> I think you have some temporary aes keys for comunication - they useless.
>
>> Thanks!

Sent from my iPhone

On Mar 23, 2013, at 5:27 AM, RCE Guru <rceguru@gmail.com> wrote:

> Hello,
>
> we not have any experience with HASP for linux, and cant help with this case. Different result every time is normal - every request to dongle and to LM (almost every) is encrypted/decrypted with different keys. Trade-exchange of knowledge is always possible, if knowledge deserve it.
>
> Regards.
>
> RT
>
> On Sat, Mar 23, 2013 at 2:48 AM, Greg Cotton <greg@quasared.net> wrote:
>> Hi, what are your prices to emulate this dongle on Linux? (We don't 100% need an emulator, but see below please...)
>>
>> Can you make a libhasp_linux.a for this dongle? (that's really all I need)
>>
>> We tried modifying the demo .a ourselves to our dongle but I assume it's more than just the Vendor ID because hasp_encrypt returns different results every time. Do you know what else we have to do? We are another reverse engineering group, and are operating on a very strict budget. I also have full source to a MicroDog 4.0 emulator that operates on a rainbow table (which we also have a rainbow table generator for it) if you are interested in a trade of knowledge; that we made.
>>
>> Thank you!
>> -Greg