Emulator - Emulator Issues #11455

Mario Kart Wii @ Wiimmfi not working with recent patches

11/12/2018 08:23 PM - Leseratte10

Status:	Fixed	**% Done:**	0%
Priority:	High		
Assignee:	delroth		
Category:			
Target version:			
Operating system:	Windows	**Relates to performance:**	No
Issue type:	Bug	**Easy:**	No
Milestone:	Current	**Relates to maintainability:**	No
Regression:	No	**Regression start:**	
Relates to usability:	No	**Fixed in:**	

Description

Game Name?
Mario Kart Wii @ Wiimmfi

Game ID? (right click the game in the game list, properties, info tab)
RMCP01, RMCE01, RMCJ01, RMCK01

MD5 Hash? (right click the game in the game list, properties, info tab, MD5 Hash: Compute)
For RMCP01, ae18681737466a6a85c7e8071c40b1d8, but that is not the original image but a Wiimmfi-patched copy, so I don't know how helpful this is. The image itself is patched properly and works just fine on a Wii.

What's the problem? Describe what went wrong.

Hello everyone. My name is Leseratte and I am one of the developers of Wiimmfi. Yesterday, for security reasons, we had to publish a mandatory update for all Wiimmfi patchers due to a security issue in Mario Kart Wii (if you are interested in background info read https://wiimmfi.de/update but I believe that is irrelevant to this bug). That patcher works without a problem on Wiis, but it does not work properly on Dolphin.

Now I basically spent the whole day today to test different Dolphin versions, settings, configurations, but was unable to get it to work flawlessly.

In my case, running Dolphin 5.0-2968 on Ubuntu 17.10, everything works fine. This is why this bug wasn't noticed during our testing. Now that we released the update, basically all Dolphin players are telling us that this doesn't work. Connecting to Wiimmfi, which should result in the server applying some RAM patches to the game, results in either error code 20100 or 20101. Network dumps show that either, the game just stops after getting the OK to continue connecting from our login server, or, that it doesn't connect to the login server properly at all.

We tried running different Dolphin builds to find out if there is one that works, but were unable to find one. We did also try running Dolphin in Interpreter mode, which, as far as I understand, should make it behave 100% like a Wii with no optimizations at all, and that seems to have fixed the problem in some cases, but most of the time it still fails and I was not able to determine any kind of pattern as to what could cause it to work / cause it to fail.

I realize that the Dolphin developers don't really have any knowledge about the patcher update / patch system, but I really hope that it is possible to somehow fix this issue anyways, because in its current state, due to this change (which works fine on console, so it is a problem in Dolphin), no Dolphin player can play MKWii online, and there were quite a few people who did that and would like to continue doing so.

Is there any kind of debugging log the affected players could create to help find this bug?

What steps will reproduce the problem?

1. Make sure you have all the Wii network stuff with certificates and whatnot set up properly: https://de.dolphin-emu.org/docs/guides/wii-network-guide/

2. Grab a Mario-Kart-Wii ISO (no matter what region)
3. Download the newest Wiimmfi patcher: https://download.wiimm.de/wiimmfi/patcher/mkw-wiimmfi-patcher-v4.zip , extract it and put your ISO into that folder.
4. Run the "patch-wiimmfi.bat" (on Windows) or the "patch-wiimmfi.sh" on Linux.
5. After some time a new ISO should appear in a "wiimmfi-images" subfolder. Boot that in Dolphin.
6. (Try to) connect to Wiimmfi.

You will receive either error 20100 or error 20101 depending on your Dolphin config, when you run in JIT Recompiler mode. When you try JITIL it doesn't boot at all. In Interpreter mode, it seems to work for some people, but not for everyone.

Is the issue present in the latest development version? For future reference, please also write down the version number of the latest development version.

Yes, it is, in 5.0-9115.

Is the issue present in the latest stable version?

Yes, it is, in 5.0.

If the issue isn't present in the latest stable version, which is the first broken version? (You can find the first broken version by bisecting. Windows users can use the tool https://forums.dolphin-emu.org/Thread-green-notice-development-thread-unofficial-dolphin-bisection-tool-for-finding-broken-builds and anyone who is building Dolphin on their own can use git bisect.)

It is present in the latest stable version.

What are your PC specifications? (CPU, GPU, Operating System, more)

I am testing on Ubuntu 17.10 with both 5.0-2968 and 5.0-9115 and it works for me, all the time. For the other users for which this fails I can only say that they use Windows and that multiple different versions of Dolphin have been tested. If the exact specifications of CPU, OS, Dolphin version are important, just tell and I'll go ask for a couple examples of Dolphin/CPU/GPU/OS combinations.

Is there anything else that can help developers narrow down the issue? (e.g. logs, screenshots, configuration files, savefiles, savestates)

Well, there aren't any special config files or savestates, tell me if you need anything and I will try to provide it. I don't know much about Dolphin internals or how such a bug might be tracked down so I don't know what information might help you guys.

History

#1 - 11/12/2018 08:27 PM - JMC4789

Is it possible it only works in Linux and the bug is on the Windows netcode in Dolphin?

I read more closely, this appears to be the case.

#2 - 11/12/2018 08:28 PM - JMC4789

- Assignee set to leoetlino

#3 - 11/12/2018 08:29 PM - Leseratte10

It seems to be, that was my guess, too. However, with that small of a sample size (1) of linux users it could just be a coincidence. I'll see if I can get some more Linux testers.

#4 - 11/12/2018 08:32 PM - JMC4789

It wouldn't be the first time Linux was the only OS that worked for something. It could also be some default setting or something.

#5 - 11/12/2018 09:40 PM - JMC4789

Can we get some accounts without the 7 day wait in order to investigate this? Or does it error out before that?

#6 - 11/12/2018 09:48 PM - Leseratte10

It errors out before that. If you get the 7-day wait error code, including a textual representation in the error message on-screen, that would already be an improvement.

In order to activate existing consoles currently within the 7-day-wait period on Wiimmfi I'd need the Wii friend code. If that is the case for you (or any other dolphin dev), you can tell me its Wii FC or its serial number. If you don't have a NAND already in the 7-day-wait period and can't get it to start due to the bug, I will be happy to provide activated NANDs to any dolphin dev who wants one in order to debug this issue (just not in public in this bug report cause then other people would download it and mess with it and get it banned).

#7 - 11/13/2018 12:29 AM - Clector

I was able to connect in Dolphin 5.0-2742 on Windows 8.1, but the latest current version (5.0-9115) always errors out.

#8 - 11/13/2018 12:32 AM - Leseratte10

Multiple people have sporadically been able to connect with various versions but it was never stable (working all the time). Can you try that a couple times, each time restarting Dolphin in-between to make sure it works stable for you? I'll later test 5.0-2742 on another Windows machine as well.

#9 - 11/13/2018 12:33 AM - Leseratte10

Correction (can't one edit comments here?): were able to connect with the Interpreter. Did you use Interpreter or JIT?

#10 - 11/13/2018 12:34 AM - Techjar

Sounds like a bisect would be extremely helpful here.

#11 - 11/13/2018 12:39 AM - MayImilae

- Priority changed from Normal to Urgent

- Milestone set to Current

- Operating system Windows added

- Operating system deleted (N/A)

#12 - 11/13/2018 12:41 AM - JMC4789

Here's an idea - at some point we made it so you had to have certain IOSes installed for certain net functions to work, where as older ones used a generic "IOS-HLE" version that didn't emulate the various intricacies. I'm wondering if maybe this is due to the older builds being less stringent.

#13 - 11/13/2018 12:47 AM - Leseratte10

So what would be the easiest way for me to get rid of the IOSes to see if, without IOS, the bug also occurs on Linux? Would it be enough to delete ~/.dolphin-emu/Wii/title/00000001/*?

#14 - 11/13/2018 12:49 AM - Clector

I tried another three times with 5.0-2742, restarting Dolphin in each try; the first time it always ends up with an error 20101, but it always connect in an stable way in all later tries.

#15 - 11/13/2018 12:50 AM - JMC4789

I wonder how hard it would be to get some dumps of communication between Wii and Dolphin and see why it's erroring out.

#16 - 11/13/2018 01:25 AM - delroth

Don't have my proper Wii NAND dump on this PC -- should the error 20101 happen before or after the "you're using a standard Dolphin NAND" error? I'm getting that one right now using latest master on Windows.

#17 - 11/13/2018 01:35 AM - Leseratte10

It should happen before. So, if your game displays the error message about the standard NAND, it should work.

Interesting that for me and delroth it works using the newest master and for clector the newest master gives errors. And the fact that the old build 2742 spits out an error once is weird as well - and also doesn't happen for me.

Communication between Wii and Dolphin? you mean between Server and Dolphin?

Does Dolphin generate any kind of debug log that might contain information about why it doesn't work properly for some?

#18 - 11/13/2018 01:37 AM - delroth

It does -- and I'd be interested in seeing a Dolphin.log with all log types at Warning level, to compare with my working Dolphin here.

#19 - 11/13/2018 01:52 AM - delroth

Shot in the dark: can we get anyone in Europe to reproduce the 20100/20101 errors, or is it happening only in places with higher latencies to Hetzner?

#20 - 11/13/2018 01:59 AM - Leseratte10

Higher latencies? Interesting theory ... my ping to our server is about 20-30 ms.

#21 - 11/13/2018 02:20 AM - Leseratte10

Used the tool "clumsy" to arbitrarily delay packets. Did a test with a delay of 500ms (so, a ping time of about 1020ms). Other than that the login took quite a bit longer, everything worked, so I'd guess ping time is not the issue.

I have asked people who get error code 20101 to provide a Dolphin log and will post that here as soon as I get one.

#22 - 11/13/2018 02:29 AM - Leseratte10

Went through the old forum posts. One person from France and one from the Netherlands reported 20101 on Dolphin as well, so, that also occurs in Europe.

#23 - 11/13/2018 08:08 AM - Leseratte10

- File dolphin.zip added

Got a log file from one of the 20101 users.

I think the important part is this:

```
05:16:385 core\ios\network\socket.cpp:334 E[IOS_SSL]: IOCTLV_NET_SSL_DOHANDSHAKE: X509 - Certificate verificat
ion failed, e.g. CRL, CA or signature check failed
05:16:385 core\ios\network\socket.cpp:357 E[IOS_SSL]: MBEDTLS_ERR_X509_CERT_VERIFY_FAILED (verify_result = 8):
 The certificate is not correctly signed by the trusted CA
```

It doesn't like our SSL certificate.

#24 - 11/13/2018 08:20 AM - Leseratte10

Maybe older builds of Dolphin did ignore that and didn't validate the cert - or did continue anyways? However that would be a bad workaround since then people could do MITM to Dolphin users and considering we are downloading and executing code I don't think that is a good idea.

#25 - 11/13/2018 08:22 AM - JMC4789

You can disable some of the checks in the INIs, like SSL and whatnot if you need to test.

#26 - 11/13/2018 08:40 AM - Leseratte10

Under Linux, there is only one entry: SSLVerifyCertificates = True
Under Windows, I had two: SSLVerifyCertificates = True and SSLVerifyCert = False

I tried to set that second setting to True to force the error, but I can still connect properly.
I'll ask one of the people with 20101 to test with both these set to False.

What is the difference though between them?

#27 - 11/13/2018 08:44 AM - JosJuice

SSLVerifyCert is historical. The latest development builds will ignore it.

#28 - 11/13/2018 10:40 AM - Leseratte10

- File dolphin_amanguver_info.log added

Just got a log file from someone else (with no change to SSLVerifyCertificates yet), and it didnn't have these validation errors, just hundreds of socket read fails like the log above. Maybe there are two independant issues?

#29 - 11/13/2018 02:47 PM - delroth

Thanks, the logs are very helpful. They both fail at the same point -- after a plain HTTP request to naswii.wiimmfi.de which returns around 400 bytes. Could you try to get a PCAP from this happening too? Filtered to "host 46.4.79.141 and port 80" should be enough for now. Because it really seems like this is the game reacting to an unexpected response from the server, somehow.

Feel free to email them to me directly in case there are any potential privacy issues. delroth@dolphin-emu.org

#30 - 11/13/2018 02:55 PM - Leseratte10

I have just sent you a mail with the dump. The HTTP request is in packet 5169 and the response in 5695. I checked the server response and that looks exactly like it should look.

#31 - 11/13/2018 02:58 PM - delroth

Haven't received anything, I guess the forwarding didn't like the email attachments... can you send directly to delroth@gmail.com instead? Thanks!

#32 - 11/13/2018 03:01 PM - Leseratte10

Done. If it still doesn't work tell me again and I'll just upload it somewhere and send you the link instead.

#33 - 11/13/2018 08:29 PM - Leseratte10

Some people are starting to report that it works with SSLVerifyCertificates = False (as I suspected from the first log I posted in #23. But obviously it is kinda bad if we make an update for security reasons and people have to disable parts of the security to make it work ...

#34 - 11/13/2018 10:44 PM - Leseratte10

Is there anything else I can do to help? Would it be helpful if I provided the CA and the server certificate we are using which apparently Dolphin doesn't want to validate correctly? Disabling the validation seems to help for quite a few people...

#35 - 11/14/2018 04:34 PM - delroth

Have we fully narrowed this down to issues with cert validation? As in, do we know of anyone that still has the issue after disabling certificates checking?

If we have good confidence it's validation issues then we'd probably need to work with someone who can reproduce the issue on their computer to figure out exactly how to debug this. There shouldn't be much difference in cert validation across machines.

#36 - 11/14/2018 04:42 PM - delroth

Another thing which would be useful and that exists in Dolphin now: could you get someone to reproduce the error with these Config/Dolphin.ini options:

SSLVerifyCertificates = True
SSLDumpRootCA = True
SSLDumpPeerCert = True

Then after the error happens, provide us with the following files in Dolphin Emulator/Dump/SSL:

main.nas.wiimmfi.de_rootca.der
main.nas.wiimmfi.de_peercert.der

That way we can do the validation check offline and maybe figure out what's going on (data corruption or something).

#37 - 11/14/2018 04:43 PM - delroth

- Status changed from New to Accepted

- Assignee changed from leoetlino to delroth

- Priority changed from Urgent to High

Also downgrading from Urgent pri since there seems to be a workaround -- not that I'm planning to ignore it, but I won't lose sleep over it :P

#38 - 11/14/2018 04:43 PM - Leseratte10

I don't know of anyone who still had (permanent) issues after disabling validation, though some people need to connect twice (and still get 20101 the first try). And, I was still unable to find any other Linux / Mac / Android users (other than myself) to confirm or deny whether the bug occurs there.

Did you find anything interesting in the dump?

I will ask some people to try with these settings and provide said files. I'll report back as soon as I get them.

#39 - 11/14/2018 05:47 PM - Leseratte10

- File cert_dump.zip added

I have received the certificate dump files from one user, and compared them to the CA and site certificate used on the Server - they are identical, so the files are not corrupted. Validation must have failed for another reason.

And even though there is a workaround, that workaround reduces the security by a lot. Basically, now that they ignore SSL validation errors anyone with MITM / anyone who convinces a user to add a different DNS server could potentially get their own malicious code to run in Dolphin ...

#40 - 11/14/2018 05:52 PM - delroth

Can you get this user's OS version, Dolphin version, and where they downloaded Dolphin from? Thanks.

(Is there a Discord where you're doing this debugging? It feels kinda inefficient to ask you to proxy info all the time.)

#41 - 11/14/2018 05:55 PM - Leseratte10

Most of it happens in the related thread in the Wii-Homebrew.com forum where all other Wiimmfi stuff is also discussed:
https://forum.wii-homebrew.com/index.php/Thread/58902-Required-Wiimmfi-Patcher-update-for-Mario-Kart-Wii/

I have asked the user to provide said info.

#42 - 11/14/2018 06:17 PM - Leseratte10

In case you didn't check that page (yet): Windows 10 64-bit, Dolphin 5.0-9132, downloaded from the official page.

#43 - 11/14/2018 06:19 PM - delroth

And I'm guessing the certificates haven't changed around the time you released your recent patch? I see the cert was generated a month ago.

This issue is super puzzling, many people on Windows 10 don't have the same certificate verification issues, and the certs are identical.

#44 - 11/14/2018 06:20 PM - Leseratte10

Well they have just been introduced then. Previously, we used HTTP for the login server. Now, because it also distributes executable code, which we don't want an attacker (Man-in-the-middle or something) to modify, we started using HTTPS.

#45 - 11/14/2018 06:22 PM - Leseratte10

If you contact me on Discord (Leseratte#6651) I can tell you some more about the internals of the patch / the changes, which I do not like to share publicly in this bug report.

#46 - 11/15/2018 01:59 AM - delroth

Alright, after 3-4h of debugging with info from Leseratte on Discord, I've got something. It's not network, it's not SSL, it's not latency to Germany. It's much better.

I'll try to stay abstract here in how things work because Leseratte has expressed concerns about describing their internals too precisely in public. Please keep that in mind if you want to dig further into this for any reason.

In general, the auth flow works like this, as seen from Dolphin network logs and Wireshark:
- An initial HTTP request is sent to ca.nas.*/ca and returns the root CA cert.
- An SSL request is sent to main.nas.* and returns some updater code which gets executed on the client.
- A third HTTP request is sent to naswii.*/ac and performs the actual authentication.

This is implemented piggy-backing on the normal Nintendo auth flow, which performs only one request but retries up to 3 times. Wiimmfi added a small amount of code to the response check code to change the connection parameters before retry, in order to use the retry as a subsequent HTTP query. So, the first /ca request succeeds, but Wiimmfi's patch transforms that into a fail and sets the URL to the one for the SSL request in order to

advance to the next step.

Part of the updater code is undoing this so that the 3rd request will be able to complete successfully, instead of returning a 3rd failure which would not be retried anymore (remember: retries up to 3 times). So after the updater code has run, the auth flow can continue normally. And that means the 3rd HTTP request to naswii.*/ac succeeds and gets interpreted as a success by the auth flow code, continuing the multiplayer connection.

Now, what is happening with Dolphin is that somehow this /ac request *still* is seen as failing, even though Wiimmfi's remote service returned a 200. Now that's super interesting, because it's completely incoherent:
- If the updater has run, then the HTTP 200 is a success, and Wiimmfi's code to turn it into a failure to piggy-back on the retries is not active anymore.
- If the updater has not run, then /ac shouldn't return a success, since it checks whether the client is up to date or not.

We're in the middle, where the updater has run, we get an HTTP 200 to prove it, and then it still gets treated as an error.

So people get booted to the main menu with error 20101. But where did this SSL wild goose chase come from?

Well, if after being booted to the main menu, you try again connecting to multiplayer... something weird happens. The patched code gets confused and sends the SSL request without having sent the /ca request first. This leads the SSL request to checking for certificates against the builtin root instead of the Wiimmfi root, which of course will fail. But if you disable checking certificates, then everything works fine and you can connect normally. Why? No clue, haven't looked enough into what happens on the second run to figure out exactly why the workaround worked as it did. Let's just say however it was pure luck, and the SSL issue had nothing to do with the actual issue, which is the first request failing. Of course for some reason people reporting the problem didn't make it clear that they had to try twice every time to get things working... probably just accepted it as a fact of life.

We actually noticed this in the logs (first request does SETROOTCA and second request does SETBUILTINROOTCA) but didn't believe it at first since nobody had mentioned having to retry :/

What's actually happening then? Well, let's talk about how the updater works. I don't know what it does, but it patches multiple places in memory by doing the following:
- write32(new_val, addr)
- dcbf addr

DCBF is the PowerPC instruction to flush L1 data cache to L2/RAM. This is useful when writing to addresses that are expected to be read by something else than the CPU's data load/store unit: if you don't dcbf, you don't have a guarantee that the write was applied further than just the L1 data cache, and other devices reading the data (for example, your GPU) will still see the old data. Not great.

However, when patching code, this is necessary but not sufficient. You need to flush data to L2 for it to be visible to the CPU's instruction fetcher, BUT you also need to make sure the address is not cached in the L1 Instruction cache. Otherwise the instruction fetcher will still read the old data from L1 ICache. So you need dcbf to flush to L2 *then* icbi to invalidate L1 and force it to re-read from L2 next time.

Wiimmfi's patcher does dcbf but no icbi. This means that if the patched code is executed very soon after patching, before the L1 ICache for that patched address gets expired, the old instruction will still be executed. As it turns out this is what happens here. I added some instrumentation to our L1I implementation to compare each read instruction to what its value in RAM is, and we see 3 different patched instructions get re-executed from ICache.

Now this is really surprising, because a decently large amount of code gets executed between the patcher and the patched code. Hell, a whole *HTTP request* gets formatted, sent over the network, received, parsed, and its payload interpreted. The L1I is only 32KB, we should definitely have expired the patched code addresses by the time we reach them again to re-execute them. Plus, Dolphin's L1I implementation is actually pretty good -- I spent some time comparing it to the spec, and really it seems decently accurate. You'd think the Wii would hit this issue too if we do, at least in some situations. But so far no reports of Wii users hitting the same problem.

But there is a huge difference between how Dolphin handles its L1I vs. the Wii. See, both on Dolphin and on the Wii, every single instruction fetches goes to the ICache. However, the Wii fetches instructions whenever it needs to execute them. On Dolphin, instruction fetch is just an input to the JITCache. Which means that while on Wii the expiry time is function of number of different instructions executed, on Dolphin, the expiry time is function of number of different instructions *recompiled*. And here's why Wiimmfi breaks: we JIT very little code between the patcher and its patchee, since we've ran all that code before already. And the patchee was one of the last things being JITed before patching, so it's still remnant in ICache.

Now, how to fix that... there is no silver bullet on the Dolphin side. First of all, Wiimmfi is buggy and should be using dcbf+icbi, and all the problems will disappear. However Dolphin's behavior will still be very different from the Wii. Ideally, we'd re-fetch all instructions through the ICache before running each block, but that's prohibitively expensive. We could clear icache on dcbf (currently we only clear jitcache), but I don't know what would break if we did that. It's not correct behavior. A more heuristic fix would be to say "well, we've run N different JIT blocks since the last instruction fetch... just invalidate icache completely, it would be completely different on real hw anyway" -- not sure how happy I'd be with that, but it would probably work and be less slow.

I'll pick phire's and booto's brains about this to see what they have to say.

#47 - 11/15/2018 03:29 AM - delroth

Oh, one unexplained question still remains: why the hell was I not able to reproduce the bug originally? At some point it started happening all the time, and before that point it never happened, but I can't pinpoint what changed to trigger the problem. Clearly it has to do with the amount of code being JITed between the patcher and the patched instruction... and it must relates to some of the game's persistent state. But I don't know what the cutoff was and why some users are or are not impacted. If you have any idea, lmk, I'm curious.

#48 - 11/15/2018 06:33 AM - Leseratte10

Yeah, that is a good question, and I have no idea why that might be. But, anyways, as long as the fix works, it doesn't really matter.

I have now changed my local copy of our update sources to include the icbi instruction for each patch, and we will probably push that out to clients on the weekend. I will let you know if that fixes all the problems. Thanks again for your help!

#49 - 11/17/2018 03:31 PM - Leseratte10

We have just pushed that fix onto the server, so now the icbi instruction is included (with no repatch needed as this is in the update loaded from the server). So now we have to wait for reports from the users if this fixed the problems or not.

#50 - 11/17/2018 04:02 PM - JMC4789

Confirmed to work today with no error codes.

#51 - 11/17/2018 09:20 PM - delroth

- Status changed from Accepted to Fixed

Files

dolphin.zip	133 KB	11/13/2018	Leseratte10
dolphin_amanguver_info.log	1.03 MB	11/13/2018	Leseratte10
cert_dump.zip	3.09 KB	11/14/2018	Leseratte10