tianocore

Reporting Issues

Bug 3356 (CVE-2021-38575) - NetworkPkg/IScsiDxe: remotely exploitable buffer overflows

Status: RESOLVED FIXED

Alias: CVE-2021-38575

Product: Tianocore Security Issues
Component: Security Issue (show other bugs)
Version: Current
Hardware: All All

Importance: Lowest normal
Assignee: Laszlo Ersek

URL:
Keywords:

Depends on:
Blocks: ~~3355~~
Show dependency tree

Reported: 2021-04-26 15:44 UTC by Laszlo Ersek
Modified: 2021-11-30 18:41 UTC (History)
CC List: 11 users (show)

See Also:
Release(s) the issue is observed: EDK II Master
The OS the target platform is running: ---
Package: NetworkPkg
Release(s) the issues must be fixed: EDK II Master

Attachments	
NetworkPkg/IScsiDxe: fix IScsiHexToBin() security and functionality bugs [v2] (25.39 KB, application/zip) 2021-05-07 07:15 UTC, Laszlo Ersek	Details
.json file_v1 (1.88 KB, application/json) 2021-09-10 18:55 UTC, kevinj	Details
.json file_v2 (1.89 KB, application/json) 2021-11-30 18:41 UTC, kevinj	Details
Add an attachment (proposed patch, testcase, etc.)	Show Obsolete (1)

Note
You need to log in before you can comment on or make changes to this bug.

Laszlo Ersek 2021-04-26 15:44:05 UTC Description

According to RFC 7143 <https://tools.ietf.org/html/rfc7143>, the iscsi target may send two hex-encoded strings to the initiator, as part of CHAP:

- CHAP_C, with the intent to authenticate the initiator,

- CHAP_R, with the intent to answer the challenge from the initiator (in case the initiator wants mutual authentication).

As of edk2 commit 61680cac5e43, these hex strings are decoded by NetworkPkg/IScsiDxe, using the IScsiHexToBin() function [NetworkPkg/IScsiDxe/IScsiMisc.c].

The function is plagued by numerous security issues:

- The IScsiHexToBin() call sites do not check the return value. According to the interface contract, EFI_BUFFER_TOO_SMALL is returned when the caller-provided binary is too small for decoding the hex string.

- The IScsiHexToBin() function never checks the BinLength parameter (which is an in-out parameter), and only ever returns EFI_SUCCESS; meaning that IScsiHexToBin() breaks its interface contract, and the decoded buffer is easy to overflow for the remote server (the iscsi target).

- The IScsiHexToBin() function peeks at HexStr (input) characters without checking the HexStr length first.

- IScsiHexToBin() does not reject an input HexStr that does not end with a full octet (i.e. one that terminates with just one nibble).

- IScsiHexToBin() silently truncates the UINTN Index to the UINT32 output BinLength.

Issues exist in the opposite direction too; minimally:

```
>|   if (((*HexLength) - 3) < BinLength * 2) {
```

here, the subtraction may underflow, the multiplication may overflow.

Laszlo Ersek 2021-04-26 15:47:50 UTC Comment 1

I'll probably fix this as part of my work on ~~bug 3355~~.

Laszlo Ersek 2021-04-26 15:48:04 UTC Comment 2

I think a CVE number would be justified.

Laszlo Ersek 2021-04-27 05:08:35 UTC Comment 3

(In reply to Laszlo Ersek from comment #0)

> - The IScsiHexToBin() function peeks at HexStr (input) characters without
> checking the HexStr length first.

Correction: this does not happen. HexStr is NUL-terminated, and so the condition

```
  (HexStr[0] == '0') && ((HexStr[1] == 'x') || (HexStr[1] == 'X'))
```

is safe, even without checking the length of the string first.

Laszlo Ersek 2021-04-29 17:25:34 UTC Comment 4

Regression test matrix template:

```
Tests with no authentication  Results
----------------------------  -------------------------
                              login result  test result
                              ------------  -----------
                              ?             ?


Tests with mutual authentication  Results
--------------------------------  --------------------------------------
secret of ... matches             CHAP_A
```

```
---------------------             -------------------
                                  offered   picked     login      test
target  initiator                 by init.  by target  result     result
------  ---------                 --------  ---------  ---------  ------
no      n/a                       ?         ?          ?          ?
yes     no                        ?         ?          ?          ?
yes     yes                       ?         ?          ?          ?
```

Laszlo Ersek 2021-04-29 17:37:51 UTC Comment 5

Created ~~attachment 713 [details]~~
NetworkPkg/IScsiDxe: fix IScsiHexToBin() security and functionality bugs

Series of 10 patches, applies on 01c0ab90beb3 ("AzurePipelines: Add support for ArmPlatformPkg", 2021-04-28).

Regression test results:

```
Tests with no authentication  Results
----------------------------  -------------------------
                              login result  test result
                              ------------  -----------
                              ok            PASS


Tests with mutual authentication  Results
--------------------------------  -------------------------------------
secret of ... matches             CHAP_A
---------------------             -------------------
                                  offered   picked     login      test
target  initiator                 by init.  by target  result     result
------  ---------                 --------  ---------  ---------  ------
no      n/a                       5         5          targ abrt  PASS
yes     no                        5         5          init abrt  PASS
yes     yes                       5         5          ok         PASS
```

Notes:

- iSCSI communication was monitored with wireshark.

Thanks,
Laszlo

```
Laszlo Ersek (10):
  NetworkPkg/IScsiDxe: wrap IScsiCHAP source files to 80 characters
  NetworkPkg/IScsiDxe: simplify "ISCSI_CHAP_AUTH_DATA.InChallenge" size
  NetworkPkg/IScsiDxe: clean up
    "ISCSI_CHAP_AUTH_DATA.OutChallengeLength"
  NetworkPkg/IScsiDxe: clean up library class dependencies
  NetworkPkg/IScsiDxe: fix potential integer overflow in IScsiBinToHex()
  NetworkPkg/IScsiDxe: assert that IScsiBinToHex() always succeeds
  NetworkPkg/IScsiDxe: reformat IScsiHexToBin() leading comment block
  NetworkPkg/IScsiDxe: fix IScsiHexToBin() hex parsing
  NetworkPkg/IScsiDxe: fix IScsiHexToBin() buffer overflow
  NetworkPkg/IScsiDxe: check IScsiHexToBin() return values

 NetworkPkg/IScsiDxe/IScsiDxe.inf |  7 +-
 NetworkPkg/IScsiDxe/IScsiCHAP.h  | 14 +--
 NetworkPkg/IScsiDxe/IScsiImpl.h  | 18 ++--
 NetworkPkg/IScsiDxe/IScsiCHAP.c  | 93 +++++++++++++++------
 NetworkPkg/IScsiDxe/IScsiMisc.c  | 65 ++++++++++----
 5 files changed, 139 insertions(+), 58 deletions(-)
```

Laszlo Ersek 2021-04-29 17:44:39 UTC Comment 6

I intend to propose an embargo end date soon.

Philippe Mathieu-Daudé 2021-05-02 16:03:15 UTC Comment 7

(In reply to Laszlo Ersek from comment #2)

> I think a CVE number would be justified.

Certainly.

Philippe Mathieu-Daudé 2021-05-02 16:10:13 UTC Comment 8

(In reply to Laszlo Ersek from comment #5)

> NetworkPkg/IScsiDxe: wrap IScsiCHAP source files to 80 characters

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: simplify "ISCSI_CHAP_AUTH_DATA.InChallenge" size

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: clean up
> "ISCSI_CHAP_AUTH_DATA.OutChallengeLength"

Why use ISCSI_CHAP_RSP_LEN instead of sizeof (AuthData->OutChallengeLength)?

Otherwise:
Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: clean up library class dependencies

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: fix potential integer overflow in IScsiBinToHex()

Predating your series, I note, while HexLength docstring has the 'output' qualifier, what is returned isn't documented.

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: assert that IScsiBinToHex() always succeeds

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: reformat IScsiHexToBin() leading comment block

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: fix IScsiHexToBin() hex parsing

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: fix IScsiHexToBin() buffer overflow

Predating your series, I note BinLength isn't documented as output (neither docstring, nor what is returned in comment).

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: check IScsiHexToBin() return values

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

For the patches fixing issues you might consider adding:
Fixes: 4c5a5e0cfecf ("Add IScsiDxe driver to NetworkPkg in order to support iSCSI over IPv6 stack and iSCSI MPIO.")

Thank you for fixing the issue!

Laszlo Ersek 2021-05-03 07:14:38 UTC Comment 9

(In reply to Philippe Mathieu-Daudé from comment #8)
> (In reply to Laszlo Ersek from comment #5)

> >   NetworkPkg/IScsiDxe: clean up
> >   "ISCSI_CHAP_AUTH_DATA.OutChallengeLength"
>
> Why use ISCSI_CHAP_RSP_LEN instead of sizeof (AuthData->OutChallengeLength)?
>
> Otherwise:
> Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

This is a great observation :) I expected it, and I'm ready with the answer.

Using ISCSI_CHAP_RSP_LEN in this patch is intentional. ~~Bug 3355~~ depends on this bug, and for one of my patches for ~~bug 3356~~, it is *semantically* important to use ISCSI_CHAP_RSP_LEN here, rather than the sizeof expression that you -- otherwise very correctly -- suggest.

The current security issue (= ~~bug 3356~~) was discovered, and is being fixed, as a part of my solution for ~~bug 3355~~. Therefore, some choices made in the present patch series are not made in isolation from the larger feature / goal.

I'd be happy to share the dependent feature patches for ~~bug 3355~~ as well. However, I have not found a good way to do so, in the TianoCore bugzilla installation. The issue is that the feature patches for ~~bug 3355~~ include such context (both semantic and actual patch context) that would break the embargo on the security problem. So normally, at this time, I would add a *private attachment* to ~~bug 3355~~. But this bugzilla instance does not support private attachments, as far as I can tell. And I wouldn't like to attach the ~~bug 3355~~ patches *here* either, as it would only cause confusion.

Another possibility would be to keep ~~bug 3355~~ a Feature Request (= not Security Issue) BZ ticket, but *still* restrict its visibility to the TianoCore Security Group, temporarily. I don't want to do it because it's risky. If someone misses my reason for that restriction (thinks that such a restriction is improper for a Feature Request) and clears the visibility restriction flag, we would again break the embargo.

NB, internally at Red Hat, I have already shared the feature patches, see: <https://bugzilla.redhat.com/show_bug.cgi?id=1935497#c14>. The patch that really depends on ISCSI_CHAP_RSP_LEN is called

  NetworkPkg/IScsiDxe: distinguish "maximum" and "selected" CHAP digest sizes

there. Let me quote the commit message of that patch here:

>| NetworkPkg/IScsiDxe: distinguish "maximum" and "selected" CHAP digest
>| sizes
>|
>| IScsiDxe uses the ISCSI_CHAP_RSP_LEN macro for expressing the size of
>| the digest (16) that it solely supports at this point (MD5).
>| ISCSI_CHAP_RSP_LEN is used for both (a) *allocating* digest-related
>| buffers (binary buffers and hex encodings alike), and (b)
>| *processing* binary digest buffers (comparing them, filling them,
>| reading them).
>|
>| In preparation for adding other hash algorithms, split purpose (a)
>| from purpose (b). For purpose (a) -- buffer allocation --, introduce
>| ISCSI_CHAP_MAX_DIGEST_SIZE. For purpose (b) -- processing --, rely on
>| MD5_DIGEST_SIZE from <BaseCryptLib.h>.
>|
>| Distinguishing these purposes is justified because purpose (b) --
>| processing -- must depend on the hashing algorithm negotiated between
>| initiator and target, while for purpose (a) -- allocation --, using
>| the maximum supported digest size is suitable. For now, because only
>| MD5 is supported, introduce ISCSI_CHAP_MAX_DIGEST_SIZE *as*
>| MD5_DIGEST_SIZE.
>|
>| Note that the argument for using the digest size as the size of the
>| outgoing challenge (in case mutual authentication is desired by the
>| initiator) remains in place. Because of this, the above two purposes
>| are distinguished for the "ISCSI_CHAP_AUTH_DATA.OutChallenge" field
>| as well.
>|
>| This patch is functionally a no-op, just yet.

The same "purpose-splitting" would actually be harder to implement and understand if in the *present* patch ('NetworkPkg/IScsiDxe: clean up "ISCSI_CHAP_AUTH_DATA.OutChallengeLength"') we used the sizeof expression rather than the macro.

So that's the reason.

> >   NetworkPkg/IScsiDxe: fix potential integer overflow in IScsiBinToHex()
>
> Predating your series, I note, while HexLength docstring has the 'output'
> qualifier, what is returned isn't documented.
>
> Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

You are right, but at least it is *somewhat* forgivable. In UEFI (even at the spec level), it is a common interface pattern to take an in-out size parameter, especially in combination with an EFI_BUFFER_TOO_SMALL return status. On input, the caller-allocated size is passed in, on output, the necessary (used) size is exposed, and the return status explains whether the transfer was actually made (i.e., whether the passed-in size was large enough). I didn't want to add more patches nor complicate these otherwise small patches.

> >   NetworkPkg/IScsiDxe: fix IScsiHexToBin() buffer overflow
>
> Predating your series, I note BinLength isn't documented as output (neither
> docstring, nor what is returned in comment).
>
> Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

Same story as above -- the interface's idea wasn't bad at all, and again there is much precedent even in the UEFI spec for this kind of size passing / size checking. (What's unfortunate is that the function didn't actually implement the interface contract.)

> For the patches fixing issues you might consider adding:
> Fixes: 4c5a5e0cfecf ("Add IScsiDxe driver to NetworkPkg in order to support
> iSCSI over IPv6 stack and iSCSI MPIO.")

Valid observation. I considered it myself, but opted against it. Here's why.

The "Fixes:" tag is useful when we can identify a *specific* earlier patch that either (a) regressed something, or (b) implemented a *well-contained* (nicely isolated) feature buggily right from the start.

Unfortunately, the commit you -- correctly -- mention is not like that. I ran "git blame" myself on the problematic code, because I wanted to understand where it came from. Commit 4c5a5e0cfecf however is a 14300 line patch, which basically dumped the -- potentially multi-month! -- original work on IScsiDxe into edk2 in a single action.

By today's standards, that's of course unacceptable. But my main point is that "Fixes: 4c5a5e0cfecf" carries very little information, as more or less *anything* we might fix in IScsiDxe nowadays would come from that huge original dump of code. Commit 4c5a5e0cfecf is not a specific, well-contained change; therefore referencing it, while fixing various bugs, is simply not informative. In my opinion anyway.

If the original 14300 lines of IScsiDxe had been carefully constructed over a series of tens of patches, then things would be different. In that case, the IScsiHexToBin() helper function would likely have been added in a much smaller patch (something like "CHAP: decode challenge from iSCSI target"), and *then* we could sensibly point a "Fixes" tag at *that* small commit now.

Put differently: this is why nowadays we require small, well isolated patches. We want bisectability, and we want "Fixes" tags that actually make sense.

> Thank you for fixing the issue!

Thank you for the review! I hope my answers are acceptable.

Riccardo Schirone 2021-05-03 11:55:03 UTC Comment 10

> NetworkPkg/IScsiDxe: fix IScsiHexToBin() buffer overflow

I'm not sure if there is a "contract" about this or not, but I noticed that BinLength here might be set to a newer value even if IScsiHexToBin ends up returning EFI_INVALID_PARAMETER due to wrong hex-characters in the input. Is this alright? I don't think it causes any real problem right now, but I'm not sure if in the rest of edk2 codebase it is expected to have values changed only when the return value is EFI_SUCCESS.

Laszlo Ersek 2021-05-04 14:24:59 UTC Comment 11

(In reply to Riccardo Schirone from comment #10)
> > NetworkPkg/IScsiDxe: fix IScsiHexToBin() buffer overflow
>
> I'm not sure if there is a "contract" about this or not, but I noticed that
> BinLength here might be set to a newer value even if IScsiHexToBin ends up
> returning EFI_INVALID_PARAMETER due to wrong hex-characters in the input. Is
> this alright?

Yes, that's alright. I can again only point at the general UEFI spec pattern: most error codes, returned by APIs, indicate that output parameters have indeterminate contents. There is a handful of (one-off) exceptions to this pattern [*], and there is one (quite common) exception *class*: namely that EFI_BUFFER_TOO_SMALL allows the caller to learn about the necessary size-on-input. This practically allows the caller to use the API for *querying* the needed size (pass in zero size at first, check for EFI_BUFFER_TOO_SMALL, allocate as needed, call API again). So that's the pattern we're sticking with here.

[*] Any such exceptional contract may be one of two kinds: a guarantee that an output parameter takes a specific new value on error return, or a guarantee that an input-output parameter does not change on error return. The rule of thumb is however that on error return, all output and all input-output parameters should be expected to have indeterminate value.

> I don't think it causes any real problem right now, but I'm
> not sure if in the rest of edk2 codebase it is expected to have values
> changed only when the return value is EFI_SUCCESS.

Yes. That's the general expectation.

Thanks!

Laszlo Ersek 2021-05-04 14:33:10 UTC Comment 12

(In reply to Laszlo Ersek from comment #11)
> (In reply to Riccardo Schirone from comment #10)

> > I don't think it causes any real problem right now, but I'm
> > not sure if in the rest of edk2 codebase it is expected to have values
> > changed only when the return value is EFI_SUCCESS.
>
> Yes. That's the general expectation.

Argh, I meant the right thing, but I didn't say what I meant. :/

To repeat:

- The general expection is that input-output and output parameters are meaningful only when EFI_SUCCESS is returned.

- The general expectation for error returns is that input-output and output parameters are indeterminate (garbled).

- In some exceptional (documented) cases, an error return may guarantee an *unchanged* parameter, or may guarantee a *particular value* for a parameter.

- The EFI_BUFFER_TOO_SMALL error return, in combination with a "size" input-output parameter, idiomatically means that "size" has been adjusted to the minimum sufficient size for calling the API successfully, given the original set of parameters (apart from "size"). On EFI_BUFFER_TOO_SMALL, all other (non-size) output and input-output parameters are usually allowed / expected to be garbled.

So, if an API fails, the caller can generally *not* expect the input-output and output parameters to preserve their values.

Riccardo Schirone 2021-05-05 04:50:20 UTC Comment 13

We would like to propose an embargo of about 1 month from today to give time to everyone to review and test patches before making this flaw public, while not delay the fix for too long.

To be precise, we propose June 8 2021 (a Tuesday) as the disclosure date.

Laszlo Ersek 2021-05-05 10:54:33 UTC Comment 14

NetworkPkg maintainers, can you please review these patches quickly? (See comment 5.) Thanks.

Maciej Rabeda 2021-05-06 11:38:11 UTC Comment 15

Sorry, I was OOO for a week. Came back today, going through the thread.

Maciej Rabeda 2021-05-06 15:01:54 UTC Comment 16

Laszlo,

Thanks for fixing this!
Review summary:

1. NetworkPkg/IScsiDxe: wrap IScsiCHAP source files to 80 characters

Coding standard - function parameter breaks & alignment:
IScsiCHAPOnRspReceived() -> IScsiGetValueByKeyFromList() calls
IScsiCHAPToSendReq() -> IScsiAddKeyValuePair() calls

2. NetworkPkg/IScsiDxe: simplify "ISCSI_CHAP_AUTH_DATA.InChallenge" size
3. NetworkPkg/IScsiDxe: clean up "ISCSI_CHAP_AUTH_DATA.OutChallengeLength"

OK, accepting explanation in your responses to Philippe.

4. NetworkPkg/IScsiDxe: clean up library class dependencies
5. NetworkPkg/IScsiDxe: fix potential integer overflow in IScsiBinToHex()
6. NetworkPkg/IScsiDxe: assert that IScsiBinToHex() always succeeds
7. NetworkPkg/IScsiDxe: reformat IScsiHexToBin() leading comment block

OK

8. NetworkPkg/IScsiDxe: fix IScsiHexToBin() hex parsing
9. NetworkPkg/IScsiDxe: fix IScsiHexToBin() buffer overflow

Comment mismatch between declaration and definition of IScsiHexToBin().

10. NetworkPkg/IScsiDxe: check IScsiHexToBin() return values

OK

Laszlo Ersek 2021-05-07 07:15:59 UTC Comment 17

Created attachment 720 [details]
NetworkPkg/IScsiDxe: fix IScsiHexToBin() security and functionality bugs [v2]

v1 was attached to comment#5.

The v2 series applies on top of commit f297b7f20010
("UnitTestFrameworkPkg: Sample unit test hangs when running in
OVMF/QEMU", 2021-05-04).

v2 addresses Maciej's review comments from comment#16. v2 incorporates
Phil's R-b tags from comment#8, except on the patches that I had to
modify. The Notes section of each individual patch summarizes the
changes (feel free to use git-range-diff too, of course, for an
incremental review).

The v2 series builds at every stage.

The v2 series has been retested for regressions using the template from
comment#4:

```
Tests with no authentication  Results
----------------------------  -------------------------
                              login result  test result
                              ------------  -----------
                              ok            PASS

Tests with mutual authentication  Results
--------------------------------  --------------------------------------
secret of ... matches             CHAP_A
---------------------             -------------------
                                  offered   picked     login      test
target  initiator                 by init.  by target  result     result
------  ---------                 --------  ---------  ---------  ------
no      n/a                       5         5          targ abrt  PASS
yes     no                        5         5          init abrt  PASS
yes     yes                       5         5          ok         PASS
```

Notes:

- iSCSI communication was monitored with wireshark.

The v2 series has been retested functionally too, with one of our
reproducers.

Thanks,
Laszlo

```
Laszlo Ersek (10):
  NetworkPkg/IScsiDxe: wrap IScsiCHAP source files to 80 characters
  NetworkPkg/IScsiDxe: simplify "ISCSI_CHAP_AUTH_DATA.InChallenge" size
  NetworkPkg/IScsiDxe: clean up
    "ISCSI_CHAP_AUTH_DATA.OutChallengeLength"
  NetworkPkg/IScsiDxe: clean up library class dependencies
  NetworkPkg/IScsiDxe: fix potential integer overflow in IScsiBinToHex()
  NetworkPkg/IScsiDxe: assert that IScsiBinToHex() always succeeds
  NetworkPkg/IScsiDxe: reformat IScsiHexToBin() leading comment block
  NetworkPkg/IScsiDxe: fix IScsiHexToBin() hex parsing
  NetworkPkg/IScsiDxe: fix IScsiHexToBin() buffer overflow
  NetworkPkg/IScsiDxe: check IScsiHexToBin() return values

 NetworkPkg/IScsiDxe/IScsiCHAP.c  | 108 ++++++++++++++++-----
 NetworkPkg/IScsiDxe/IScsiCHAP.h  |  14 ++-
 NetworkPkg/IScsiDxe/IScsiDxe.inf |   7 +-
 NetworkPkg/IScsiDxe/IScsiImpl.h  |  18 ++--
 NetworkPkg/IScsiDxe/IScsiMisc.c  |  65 ++++++++++---
 NetworkPkg/IScsiDxe/IScsiMisc.h  |  19 ++--
 6 files changed, 166 insertions(+), 65 deletions(-)
```

Maciej Rabeda 2021-05-10 09:14:50 UTC Comment 18

Reviewed-by: Maciej Rabeda <maciej.rabeda@linux.intel.com>

Laszlo Ersek 2021-05-11 10:28:15 UTC Comment 19

Thank you, Maciej!

Philippe Mathieu-Daudé 2021-05-11 13:07:45 UTC Comment 20

(In reply to Laszlo Ersek from comment #17)

> NetworkPkg/IScsiDxe: wrap IScsiCHAP source files to 80 characters

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: fix potential integer overflow in IScsiBinToHex()

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: reformat IScsiHexToBin() leading comment block

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: fix IScsiHexToBin() hex parsing

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

> NetworkPkg/IScsiDxe: fix IScsiHexToBin() buffer overflow

Reviewed-by: Philippe Mathieu-Daude <philmd@redhat.com>

Laszlo Ersek 2021-05-11 15:00:25 UTC Comment 21

Thank you for being awesome, Phil! :)

Laszlo Ersek 2021-05-11 15:14:36 UTC Comment 22

Note for backporters (given that the patches from comment 17 are supposed to reach downstreams before I post them to edk2-devel): all patches have been R-b Phil and Maciej. Don't forget to pick up those tags in your backports. I'm not refreshing the patches from comment 17 here, just for the sake of the R-b updates -- I don't want any confusion about the "latest code updates" in this BZ. I'll include the R-b's (a) in my own backports, and (b) in my posting to edk2-devel, upon public disclosure (see comment 13). Thanks.

Riccardo Schirone 2021-06-03 05:19:24 UTC Comment 23

Hi, could you assign a CVE before this becomes public in 5 days? Thanks!

Laszlo Ersek 2021-06-08 08:14:58 UTC Comment 24

Public posting:

* [edk2-devel] [PUBLIC edk2 PATCH v2 00/10]
  NetworkPkg/IScsiDxe: fix IScsiHexToBin() security and functionality bugs

Message-Id: <20210608121259.32451-1-lersek@redhat.com>
https://listman.redhat.com/archives/edk2-devel-archive/2021-June/msg00316.html
https://edk2.groups.io/g/devel/message/76198

Laszlo Ersek 2021-06-09 13:28:52 UTC Comment 25

(In reply to Laszlo Ersek from comment #24)
> Public posting:
>
> * [edk2-devel] [PUBLIC edk2 PATCH v2 00/10]
>   NetworkPkg/IScsiDxe: fix IScsiHexToBin() security and functionality bugs
>
> Message-Id: <20210608121259.32451-1-lersek@redhat.com>
> https://listman.redhat.com/archives/edk2-devel-archive/2021-June/msg00316.html
> https://edk2.groups.io/g/devel/message/76198

Merged as commit range 702ba436ed8e..b8649cf2a3e6, via <https://github.com/tianocore/edk2/pull/1698>.

Riccardo Schirone 2021-06-14 03:40:46 UTC Comment 26

Any news about a CVE for this? The later it is assigned, the less useful it becomes. In general it would be great to have it even before the flaw becomes public.

Gianluca Gabrielli 2021-06-14 10:04:59 UTC Comment 27

As Riccardo pointed out, having a CVE ID assigned is very important to increase the security level of downstream projects which are shipping/levering this package. If nobody managed to request one yet, I will do.

Riccardo Schirone 2021-06-14 11:10:26 UTC Comment 28

(In reply to Gianluca Gabrielli from comment #27)
> As Riccardo pointed out, having a CVE ID assigned is very important to
> increase the security level of downstream projects which are
> shipping/levering this package. If nobody managed to request one yet, I will
> do.

As TianoCore itself is CNA for edk2 I think they are already aware of this, but someone more involved with the project might know more.

Riccardo Schirone 2021-07-05 03:57:08 UTC Comment 29

Asking again: can we have a CVE for this please?

I will wait until tomorrow, then I will ask directly MITRE for it.

Riccardo Schirone 2021-07-08 02:05:16 UTC Comment 30

(In reply to Riccardo Schirone from comment #29)
> Asking again: can we have a CVE for this please?
>
> I will wait until tomorrow, then I will ask directly MITRE for it.

I requested the CVE directly to MITRE. No reply yet.

kevinj 2021-09-10 18:55:23 UTC Comment 31

Created attachment 825 [details]
.json file_v1

I have attached the .json file for CVE classification. Please review and provide feedback, and assign a release this issue is observed in, so I can update the .json file.

kevinj 2021-11-30 18:41:41 UTC Comment 32

Created attachment 890 [details]
.json file_v2

I have updated the .json file and submitted to MITRE for publication.