

# 株式会社 ITS MORE

2020年4月設立

ITS more

2020年6月30日 投稿者: SATOXITS

## 固まるSSLプロセス

基盤：どうも以前から、HTTPSのリバースプロキシのサーバプロセスが固まって溜まってしまっている現象があります。時々手動で掃除しているのですが。

開発：何でしょうね？コネクションにSSLをかぶせてるプロセスだと思いますが。

基盤：ps xlf するとこんな感じ。

```
1471      1  20    0  47100  6180 poll_s Ss  ?           0:02 ./dg9.9.1:
15067   1471  20    0 113752 10560 futex_ Sl  ?           0:00 \_ DeleGa
16298   1471  20    0 113812 10320 futex_ Sl  ?           0:00 \_ DeleGa
21228   1471  26    6 113788 10860 futex_ SNl ?           0:00 \_ DeleGa
30802   1471  20    0 113980 11472 futex_ Sl  ?           0:00 \_ DeleGa
32232   1471  20    0 113828 10600 futex_ Sl  ?           0:00 \_ DeleGa
 3942   1471  20    0 114000 11864 futex_ Sl  ?           0:00 \_ DeleGa
 5695   1471  20    0 113756 10840 poll_s S  ?           0:00 \_ DeleGa
 5724   1471  20    0 113768 10580 futex_ Sl  ?           0:00 \_ DeleGa
 5762   1471  20    0 113684 10248 poll_s S  ?           0:00 \_ DeleGa
```

基盤：現在のプロセス番号は5000番台なので、それより大きいPIDのは一巡以上前のやつです。

開発：WCHANがfutexシステムコールということなので、何かの排他制御で固まっているようですね。ていうかLinuxってプロセス番号15ビットなわけ？

基盤：まあそれはカーネルのコンフィグ次第じゃないですかね。先日までdelegate.orgだったAzureのUbuntu18でのDeleGateのログを見ると130000を超えたあたりまで行ってます。130000 == 0x1FBDD0 ですから、おそらく17ビット使ってると思います。

社長：私はfutexというようなシステムコールは知らないし、呼んだ覚えは無いのですが。

開発：ライブラリ関数から呼んでるんでしょうね。gdbで覗いてみましょうか。

基盤：gdb … 無いですね。sudo apt install gdb。OK。

開発：で、sudo gdb dg9.9.13+ -p 32232。OK。そしてbt。

社長：bt 超ナツカシス。

```
(gdb) bt
#0  0x00007f78aa0b798d in pthread_join (threadid=14015623436
#1  0x0000564c916f3ff4 in _Thread_wait (tid=525057, timeout=
#2  0x0000564c916abeac in thread_wait (tid=525057, timeout=3
#3  0x0000564c913759bf in popPFilter (Conn=0x7ffe720f4920, t
#4  0x0000564c91375ca9 in waitFilterThread (Conn=0x7ffe720f4
#5  0x0000564c912b50b2 in call_client1 (Conn=0x7ffe720f4920,
#6  0x0000564c912b68d9 in EXEC_client (Conn=0x7ffe720f4920,
#7  0x0000564c912c80a0 in StickyServer (Conn=0x7ffe720f4920,
#8  0x0000564c912c2371 in forkStickyServer (Conn=0x7ffe720f4
#9  0x0000564c912c1770 in _main (ac=15, av=0x564c934c0e40) a
#10 0x0000564c912b7a4e in _mainx (ac=15, av=0x564c934c0e40)
#11 0x0000564c912bc686 in delegate_main (ac=15, av=0x564c934
#12 0x0000564c916f01bf in alloca_call (ap=0x7ffe72110c70) at
#13 0x0000564c916ad7cc in randstack_call (strg=1, func=0x564
#14 0x0000564c912cc928 in mainX (ac=15, av=0x564c934c0e40) a
#15 0x0000564c912a3851 in main (ac=15, av=0x7ffe72110ef8) at
```

開発：SSLフィルターのスレッドの終了待ちのようですね。ログは？

```
06/30 07:47:21.13 [32232] 770+4: (5) accepted [25] -@[127.0.0.1]localho
06/30 07:47:21.13 [32232] 770+4: Proxy: host=localhost; User-Agent: Moz
06/30 07:47:21.13 [32232] 770+4: HCKA: Via=Via: 1.1 - (DeleGate/9.9.13)
06/30 07:47:21.13 [32232] 770+4: REQUEST - GET /its-more.jp/delegate/ft
06/30 07:47:21.13 [32232] 770+4: *** /its-more.jp/delegate/ftp/pub/Dele
06/30 07:47:21.13 [32232] 770+4: REQUEST +M https://its-more.jp/delegat
06/30 07:47:21.13 [32232] 770+4: *** /its-more.jp/delegate/ftp/pub/Dele
06/30 07:47:21.13 [32232] 770+4: PATH> https://its-more.jp:443!localhos
06/30 07:47:21.13 [32232] 770+4: REQUEST = [https://its-more.jp:443/] G
06/30 07:47:21.13 [32232] 770+4: XHost: (0,0,1) its-more.jp <= localhos
06/30 07:47:21.13 [32232] 770+4: ## left connected but dead [30] <= ddi
06/30 07:47:21.13 [32232] 770+4: #HT11 3 getServ 25.2s(25.1s)*0 SERVER
06/30 07:47:21.14 [32232] 770+4: ConnectToServer connected [26] {163.44
06/30 07:47:21.14 [32232] 770+4: ---OpenServer [https://its-more.jp:443
06/30 07:47:21.14 [32232] 770+4: ---OpenServer(https https://its-more.j
06/30 07:47:21.14 [32232] 770+4: ---OpenServer returns 26
06/30 07:47:21.14 [32232] 770+4: willSTLS_SV[https]: ServerFlags=0 720F
(UNIX) 07:47:21.138 [32232.80100] ##tid add[1] 4 7F78AA8FF700/80401
07/02 07:47:21.14 [32232.80401] 770+4: {t} setCloseOnFork(SSLstart,31)
07/02 07:47:21.14 [32232.80401] 770+4: {t} setCloseOnFork(SSLstart,32)
07/02 07:47:21.14 [32232.80401] 770+4: ## SSLway -- TLSxSNI: sent ru=0
07/02 07:47:21.15 [32232.80401] 770+4: ## SSLway ## 0.011729 connected/
07/02 07:47:21.15 [32232.80401] 770+4: ## SSLway server's cert. = **sub
07/02 07:47:21.15 [32232.80401] 770+4: -- SSLready[FSV]8#3[30]>>[30/-1]
06/30 07:47:21.15 [32232] 770+4: -- SSLready[FSV] << [29][30/-1] 1 0.01
06/30 07:47:21.15 [32232] 770+4: --pushPFilter (https/https) tid=401 [2
06/30 07:47:21.15 [32232] 770+4: HTTP => (its-more.jp:443) GET /delegat
06/30 07:47:21.16 [32232] 770+4: #HT11 NO-response-buffering: chunked m
06/30 07:47:21.16 [32232] 770+4: #HT11 chunked, should skip: Content-Le
06/30 07:47:21.16 [32232] 770+4: Guessed Content-Type:application/octet
06/30 07:47:21.16 [32232] 770+4: #HT11 SERVER ver[HTTP/1.1] conn[keep-a
06/30 07:47:21.16 [32232] 770+4: #HT11 --putChunk-Header: Transfer-Enco
06/30 07:47:21.16 [32232] 770+4: HTTP/1.1 200 Content-{Type:application
07/02 07:47:21.65 [32232] 770+4: Fwrite(25): partial (0/16384/16384) NB
(UNIX) 07:47:21.648 [32232.80100] +++EPIPE fwriteTIMEOUT() 0/16384 SIG*
06/30 07:47:21.65 [32232] 770+4: ## HTTP out to client TIMEOUT[25]
06/30 07:47:21.65 [32232] 770+4: ClientEOF: binary flush-1 [25 25] 8 20
06/30 07:47:21.65 [32232] 770+4: ## HTTP ERROR incomplete fwrite(16384)
06/30 07:47:21.65 [32232] 770+4: ## stop relay-2: FWRITE(16384)=0, SIGP
06/30 07:47:21.65 [32232] 770+4: Incomplete 200 H0 B1 (2707172 / 444779
06/30 07:47:21.65 [32232] 770+4: {F} SET checkClientEOF[25][25][25] flu
06/30 07:47:21.65 [32232] 770+4: HTTP transmitted: 284head+2707172/4447
06/30 07:47:21.65 [32232] 770+4: #HT11 close svsocks[29,30]
06/30 07:47:21.65 [32232] 770+4: HCKA:[0] closed -- r:by client (respon
06/30 07:47:21.65 [32232] 770+4: WaitShutdown 0/0 xpid=-1 errno=10/10 0
06/30 07:47:21.65 [32232] 770+4: disconnected [25] -@[127.0.0.1]localho
06/30 07:47:21.95 [32232] 770+4: --- thread_wait(80401,300) TIMEOUT
06/30 07:47:21.95 [32232] 770+4: -- wait threadFilter[FSV][80401] 1/3 e
06/30 07:47:22.25 [32232] 770+4: --- thread_wait(80401,300) TIMEOUT
06/30 07:47:22.25 [32232] 770+4: --popPFilter(8) ty=8 tid=401 [26][27]
```

社長：固まってるプロセスに共通点は？アクセス先とか。

基盤：実行ファイルとかPDFです。大きめのバイナリですかね。

開発：いずれも disconnected はしてるから、クライアント側とは切断していて、一方リバースプロキシ先のサーバとのSSLの切断を待ってるという状態ですね。

社長：SSLかTLSのプロトコルが変わったとかかな？10年前には見なかった現象ですが。というか、ヘビーにHTTPSのリバースプロキシはやったことがなかったかも知れ

ませんが。

開発：それって再現性ありますか？同じアクセスで固まらない場合があるかという。

基盤：同じのにアクセスしてみます。再現しないですね。固まらない。

```
06/30 11:04:25.29 [8320] 1055+10: (7) accepted [25] -@[127.0.0.1]localh
06/30 11:04:25.29 [8320] 1055+10: Proxy: host=localhost; User-Agent: Mc
i/537.36; Via: 1.1 - (DeleGate/9.9.13);
06/30 11:04:25.29 [8320] 1055+10: HCKA: Via=Via: 1.1 - (DeleGate/9.9.13
06/30 11:04:25.29 [8320] 1055+10: REQUEST - GET /its-more.jp/delegate/f
06/30 11:04:25.29 [8320] 1055+10: *** /its-more.jp/delegate/ftp/pub/Del
_13 ***
06/30 11:04:25.29 [8320] 1055+10: REQUEST +M https://its-more.jp/delega
06/30 11:04:25.29 [8320] 1055+10: *** /its-more.jp/delegate/ftp/pub/Del
_13 ***
06/30 11:04:25.29 [8320] 1055+10: PATH> https://its-more.jp:443!localhc
06/30 11:04:25.29 [8320] 1055+10: REQUEST = [https://its-more.jp:443/]
06/30 11:04:25.29 [8320] 1055+10: XHost: (0,0,1) its-more.jp <= localhc
06/30 11:04:25.29 [8320] 1055+10: ## left connected but dead [30] <= dc
06/30 11:04:25.29 [8320] 1055+10: #HT11 9 getServ 26.7s(26.6s)*0 SERVEF
06/30 11:04:25.29 [8320] 1055+10: ConnectToServer connected [26] {163.4
06/30 11:04:25.29 [8320] 1055+10: ---OpenServer [https://its-more.jp:44
06/30 11:04:25.29 [8320] 1055+10: ---OpenServer(https https://its-more.
06/30 11:04:25.29 [8320] 1055+10: ---OpenServer returns 26
06/30 11:04:25.29 [8320] 1055+10: willSTLS_SV[https]: ServerFlags=0 72(
(UNIX) 11:04:25.294 [8320.80100] ##tid add[1] 10 7F78AA8FF700/80A01
07/02 11:04:25.29 [8320.80A01] 1055+10: (t) setCloseOnFork(SSLstart,31)
07/02 11:04:25.29 [8320.80A01] 1055+10: (t) setCloseOnFork(SSLstart,32)
07/02 11:04:25.30 [8320.80A01] 1055+10: ## SSLway -- TLSxSNI: sent ru=(
07/02 11:04:25.31 [8320.80A01] 1055+10: ## SSLway ## 0.012184 connectec
07/02 11:04:25.31 [8320.80A01] 1055+10: ## SSLway server's cert. = **st
07/02 11:04:25.31 [8320.80A01] 1055+10: -- SSLready[FSV]8#9[30]>>[30/-1
06/30 11:04:25.31 [8320] 1055+10: -- SSLready[FSV] << [29][30/-1] 1 0.(
06/30 11:04:25.31 [8320] 1055+10: --pushPFilter (https/https) tid=A01 |
06/30 11:04:25.31 [8320] 1055+10: HTTP => (its-more.jp:443) GET /delega
06/30 11:04:25.32 [8320] 1055+10: #HT11 NO-response-buffering: chunked
06/30 11:04:25.32 [8320] 1055+10: #HT11 chunked, should skip: Content-I
06/30 11:04:25.32 [8320] 1055+10: Guessed Content-Type:application/octe
06/30 11:04:25.32 [8320] 1055+10: #HT11 SERVER ver[HTTP/1.1] conn[keep-
06/30 11:04:25.32 [8320] 1055+10: #HT11 --putChunk-Header: Transfer-Enc
06/30 11:04:25.32 [8320] 1055+10: HTTP/1.1 200 Content-{Type:applicatic
07/02 11:04:27.42 [8320.80A01] 1055+10: ## SSLway [FSV] (0) [31,32] read
06/30 11:04:28.56 [8320] 1055+10: HTTP transmitted: 284head+4447791/444
06/30 11:04:28.56 [8320] 1055+10: #HT11 10 putServ(29/30/26) its-more.
06/30 11:04:28.57 [8320] 1055+10/1: --- thread_wait(80A01,1) TIMEOUT
06/30 11:04:28.57 [8320] 1055+10/1: -- wait threadFilter[FSV][80A01] 1/
06/30 11:04:28.57 [8320] 1055+10/1: --- thread_wait(80A01,1) TIMEOUT
06/30 11:04:28.57 [8320] 1055+10/1: --popPFilter(8) ty=8 tid=A01 [26][2
07/02 11:04:29.42 [8320.80A01] 1055+10/1: ## SSLway [FSV] (1) [31,32] rea
07/02 11:04:31.42 [8320.80A01] 1055+10/1: ## SSLway [FSV] (2) [31,32] rea
07/02 11:04:33.42 [8320.80A01] 1055+10/1: ## SSLway [FSV] (3) [31,32] rea
07/02 11:04:35.36 [8320.80A01] 1055+10/1: ## SSLway FSV S-C:4448075/543
07/02 11:04:35.36 [8320.80A01] 1055+10/1: -- [FSV] SSLway close[31,32]
07/02 11:04:35.36 [8320.80A01] 1055+10/1: (t) clearCloseOnFork(SSLend,3
07/02 11:04:35.36 [8320.80A01] 1055+10/1: (t) clearCloseOnFork(SSLend,3
07/02 11:04:35.36 [8320.80A01] 1055+10/1: -- [FSV] SSLwayF DONT close[2
(UNIX) 11:04:35.362 [8320.80100] ##tid del[1] 10 7F78AA8FF700/80901
06/30 11:04:35.36 [8320] 1055+10/1: --popPFilter(8) ty=8 tid=901 [26][2
11:04:35[8320][8320]100 0/0/9/9 FSV thread-SSLway remaining-A
06/30 11:04:41.89 [8320] 1055+10/1: ClientEOF: request-EOF-6 [25 25] 8
06/30 11:04:41.89 [8320] 1055+10/1: HCKA:[1] closed -- d:by client(requ
06/30 11:04:41.89 [8320] 1055+10/1: disconnected [25] -@[127.0.0.1]loca
06/30 11:04:55.16 [8320] 1055+10: ##### getRIDENT 64[172.26.7.51:80][216
(UNIX) 11:04:55.162 [8320.80100] -- RES update error () (addr)
06/30 11:04:55.16 [8320] 1055+11: (7) accepted [25] -@[127.0.0.1]localh
```

開発：しかしこれ、クライアント側が切断されるまでやたら時間がかかりましたね。

```

06/30 11:18:13.58 [8789] 1069+5: #HT11 --putChunk-Header: Transfer-Encod
06/30 11:18:13.58 [8789] 1069+5: HTTP/1.1 200 Content-{Type:applicator
06/30 11:18:14.17 [8789] 1069+5: HTTP transmitted: 284head+4447791/4447
06/30 11:18:14.17 [8789] 1069+5: #HT11 5 putServ(29/30/26) its-more.jp:
06/30 11:18:14.17 [8789] 1069+5/1: --- thread_wait(80301,1) TIMEOUT
06/30 11:18:14.17 [8789] 1069+5/1: -- wait threadFilter[FSV][80301] 1/2
07/02 11:18:16.17 [8789.80301] 1069+5/1: ## SSLway [FSV] (0) [31,32] reac
07/02 11:18:18.17 [8789.80301] 1069+5/1: ## SSLway [FSV] (1) [31,32] reac
07/02 11:18:20.18 [8789.80301] 1069+5/1: ## SSLway [FSV] (2) [31,32] reac
07/02 11:18:22.18 [8789.80301] 1069+5/1: ## SSLway [FSV] (3) [31,32] reac
07/02 11:18:23.81 [8789.80301] 1069+5/1: ## SSLway FSV S-C:4448075/543
07/02 11:18:23.81 [8789.80301] 1069+5/1: -- [FSV] SSLway close[31,32]
07/02 11:18:23.81 [8789.80301] 1069+5/1: {t} clearCloseOnFork(SSLend,32
07/02 11:18:23.81 [8789.80301] 1069+5/1: {t} clearCloseOnFork(SSLend,31
07/02 11:18:23.81 [8789.80301] 1069+5/1: -- [FSV] SSLwayF DONT close[27
(UnIX) 11:18:23.808 [8789.80100] ##tid del[1] 3 7F78AA8FF700/80201
06/30 11:18:23.81 [8789] 1069+5/1: --popPFilter(8) ty=8 tid=201 [25][26]
11:18:23[8789][8789]100 0/0/2/2 SIG dangling thread-id 80301 3
(UnIX) 11:18:23.808 [8789.80100] ##TW BAD tid=80301 FFFFFFFFFFFFFFFF 3
06/30 11:18:23.81 [8789] 1069+5/1: --popPFilter(8) ty=8 tid=301 [26][27]
11:18:23[8789][8789]100 0/0/2/2 FSV thread-SSLway remaining-A
06/30 11:18:28.90 [8789] 1069+5/1: ClientEOF: request-EOF-6 [25 25] 8 (
06/30 11:18:28.90 [8789] 1069+5/1: HCKA:[1] closed -- d:by client(reque
06/30 11:18:28.90 [8789] 1069+5/1: disconnected [25] -@[127.0.0.1]local
06/30 11:18:58.93 [8789] 1069+5: StickyServer done [timeout] 5 req / 54
11:18:58[8789][8789]100 0/0/2/2 SIG dangling thread-id 80301 3
(UnIX) 11:18:58.927 [8789.80100] ##TW BAD tid=80301 FFFFFFFFFFFFFFFF 3
06/30 11:18:58.93 [8789] 1069+5: clearServ: its-more.jp:443 [0][80301/-
06/30 11:18:58.93 [8789] 1069+5: #Sig/CSC finish 4295 1777 P2 R0 E0 {57

```

基盤：データ転送自体は0.5秒くらいで終わってますが、SSLが切断されるまでに10秒とかかかっているような？

開発：なんか、スレッドidがおかしなことになってますね。32ビットと64ビットの違いとか？

社長：これって単にサーバとKeep-Aliveしてるって話のような？それならHTTP的にタイムアウトして切れるまでSSLも切れない。まずはそれを止めてみましょう。

基盤：どうやるんでしょう？

社長：さあ。マニュアルを読んで下さい。

基盤：Manual.shtmlで keep を探す… これですかね？

```

bugs:listOfBugs
listOfBugs is a list of features to be disabled to bypass possible bugs as
follows:

no-gzip ... disable Content-Encoding:gzip
no-keepalive ... disable Connection:Keep-Alive
no-keepaliveproxy ... disable Connection:Keep-Alive with client side
proxy

```

基盤：それでは HTTPCONF=bugs:no-keepalive を足して再起動。おや、ゾンビがポートを離さないののでリスタートできない。しかたが無い、親なしになった子プロセスを kill -9 で成仏させて、再起動。これ、再起動時にSIGTERMじゃなくてSIGKILLで終了させるってオプション無いんでしょうか？

社長：SIGTERMを聞いてくれなかったらSIGINTかSIGKILLするというオプションは作ったように思います。

基盤：とりあえず再起動できたので。ロード。おやログがスッキリしました。

```
06/30 11:49:56.41 [9675] 27+1: #HT11 chunked, should skip: Content-Length: 4447791^M
06/30 11:49:56.41 [9675] 27+1: Guessed Content-Type:application/octet-stream https://its-more.jp:
06/30 11:49:56.41 [9675] 27+1: #HT11 SERVER ver[HTTP/1.1] conn[keep-alive]
06/30 11:49:56.41 [9675] 27+1: #HT11 --putChunk-Header: Transfer-Encoding: chunked^M
06/30 11:49:56.41 [9675] 27+1: HTTP/1.1 200 Content-{Type:application/octet-stream Encoding:[/] I
07/02 11:49:58.49 [9675.80201] 27+1: ## SSLway [FSV] (0) [31,32] ready=0
07/02 11:50:00.49 [9675.80201] 27+1: ## SSLway [FSV] (1) [31,32] ready=0
(UNIX) 11:50:00.696 [9675.80100] closed-4[44 -1][25 25 25 27 28/44 44 44]relayResp http.c:6877
06/30 11:50:00.70 [9675] 27+1: HTTP transmitted: 284head+4447791/4447791body=>0txt+0bin->4447791/
06/30 11:50:00.70 [9675] 27+1: #HT11 close svsocks[29,30]
06/30 11:50:00.70 [9675] 27+1: ## tcCLOSED fc=44 FromC=-1 ToC=-1 CS=-1
06/30 11:50:00.70 [9675] 27+1: disconnected [44] -@[127.0.0.1]localhost:58200.-.M106073204001.v4.
06/30 11:50:00.99 [9675] 27+1: --- thread_wait(80201,300) TIMEOUT
06/30 11:50:00.99 [9675] 27+1: -- wait threadFilter[FSV][80201] 1/1 err=-1
06/30 11:50:01.30 [9675] 27+1: --- thread_wait(80201,300) TIMEOUT
06/30 11:50:01.30 [9675] 27+1: --popPFilter(8) ty=8 tid=201 [25][27] 1/1 err=-1
06/30 11:50:01.30 [9675] 27+1: CFI process none (0/1)
06/30 11:50:01.30 [9675] 27+1: StickyServer done [nonStickyProtocol(http:https:https)] 1 req / 1+
06/30 11:50:01.30 [9675] 27+1: #Sig/CSC finish 653 816 P2 R0 E0 {23 r0 t0} 1/1/1
```

開発：でもあいかわらずスレッド終了待ちがタイムアウトとかしてるのは気持ちわるいですね。

基盤：それより何より、バイナリがブラウザに表示されてしまうというコレ。



開発：Content-Type: application/octet-stream じゃいかなのですかね？

基盤：いえ、リバース先の nginx はContent-Typeを付けてくれてないようです。

```
% dget -h http://delegate.org/its-more.jp/delegate/ftp/pub/DeleGate/bin-late
http [delegate.org:80] /its-more.jp/delegate/ftp/pub/DeleGate/bin-latest9/1:
HTTP/1.1 200 OK
dget-HTTP>>> HTTP/1.1 200 OK
DeleGate-Ver: 9.9.13 (delay=0)
DeleGate-Ver: 9.9.13 (delay=0)
Server: nginx
Date: Tue, 30 Jun 2020 03:10:23 GMT
Content-Length: 4447791
Last-Modified: Fri, 31 Oct 2014 10:03:39 GMT
ETag: "43de2f-506b51c6854c0"
Accept-Ranges: bytes
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Via: 1.1 - (DeleGate/9.9.13), 1.1 - (DeleGate/9.9.13)
Connection: close

   recv#   time   total-bytes  bytes/s
     328    0.1     4447791  43943523
[10466] total: 4447791 / 4447791 [172.26.7.51:41634]
12:10:23  1: /its-more.jp/delegate/ftp/pub/DeleGate/bin-latest9/linux2.6-d
1 GET / 0.134 seconds = 7.48 GET/second 4447791 bytes (0.133/0.133/0.133)
```

開発：ああ、それで DeleGate が Gussed って出してたのか。これ、デフォルトの Content-Type で付けられるんですけどっけ？

社長：さあ。マニュアルには何と？

基盤：content-type で検索… 文字コード変換の場合と、CFIの場合にしか出てこないですね。

社長：まあ、CFIを噛ませれば良いって話ではありますが… まあ、ちょっと重くはなるでしょうね。

開発：ソースをみてくださいか。grep Gussed src/\*.c。ああ、http.cですね。どれ…

```
if( ctype = filename2ctype(head) ){
}else{
    ctype = "application/octet-stream";
}
svllog("Gussed Content-Type:%s %s://%s:%d%s\n",ctype,
DST_PROTO,DST_HOST,DST_PORT,head);
sprintf(head,"Content-Type: %s\r\n",ctype);
scanRespHead1 (Conn,RX,head);
```

開発：これって、推定した結果を反映させる気満々だったようですが。



社長：ちょっと、クライアントに返しているヘッダをログに出してもらえますか？

基盤：こういう様子です。Content-Type 返してませんね。小さなCOPYRIGHTファイルにアクセスしてコレですから、サイズとかバイナリがとかでは無く、単にオリジンサーバが返してなかったらそうなるということのようです。

```
07/02 12:50:24.38 [12923.80201] 52+1: -- SSLready[FSV]8#1[31]>>[31/-1] READY
06/30 12:50:24.38 [12923] 52+1: -- SSLready[FSV] << [30][31/-1] 1 0.012
06/30 12:50:24.38 [12923] 52+1: --pushPFilter (https/https) tid=201 [25][28] 0 542
06/30 12:50:24.38 [12923] 52+1: HTTP => (its-more.jp:443) GET /delegate/ftp/pub/De
06/30 12:50:24.39 [12923] 52+1: Gussed Content-Type:application/octet-stream http
06/30 12:50:24.39 [12923] 52+1: #HT11 SERVER ver[HTTP/1.1] conn[keep-alive]
06/30 12:50:24.39 [12923] 52+1: HTTP/1.1 200 Content-{Type:application/octet-strea
06/30 12:50:24.39 [12923] 52+1: detach respBuff: non-text data, non keep-alive
06/30 12:50:24.39 [12923] 52+1: ERASED Content-Length: 372^M
1 HTTP/1.1 200 OK
2 DeleGate-Ver: 9.9.13 (delay=0)
3 Server: nginx
4 Date: Tue, 30 Jun 2020 03:50:24 GMT
5 Last-Modified: Fri, 31 Oct 2014 06:40:10 GMT
6 ETag: "174-506b244b1c680"
7 Accept-Ranges: bytes
8 X-XSS-Protection: 1; mode=block
9 X-Content-Type-Options: nosniff
10 Via: 1.1 - (DeleGate/9.9.13)
11 Connection: close
12
```

社長：ぐう。あー、お腹がすきましたね。飲みに行きましょうか。

開発：いや、ここは逃げずにもうひと踏ん張り。

基盤：バイナリのタイムスタンプからしてソースはこれかな。一応コピーしてと… しまった！自分自身にtar cf | xf してファイルが壊れました。

開発：まあ、隣のサーバにコピーがありますし。

基盤：あ、いや。一応tarしてアーカイブしてありました。

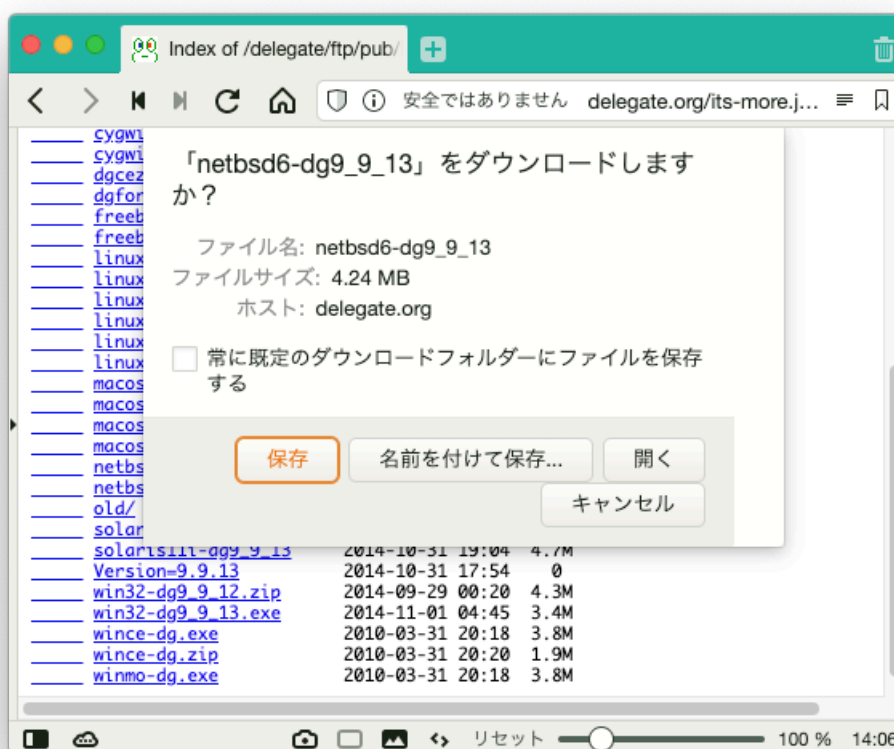
開発：で http.c を見る。Guess した Content-Type は何に使われているのか… ああ、内部的な判定に使われてるだけで出力されてないですね。じゃこれをクライアントに向けて出しちゃうと…

```
06/30 13:46:10.95 [18439] 1+1: Gussed Content-Type:application/octet-stream ht
06/30 13:46:10.95 [18439] 1+1: (^-^)/@2020-0630 added guessed Content-Type: ap
06/30 13:46:10.95 [18439] 1+1: #HT11 SERVER ver[HTTP/1.1] conn[keep-alive]
06/30 13:46:10.95 [18439] 1+1: #HT11 server KEEP-ALIVE

% dget -h http://delegate.org/wits-more.jp/delegate/ftp/pub/DeleGate/COPYRIGHT
http [delegate.org:80] /wits-more.jp/delegate/ftp/pub/DeleGate/COPYRIGHT
HTTP/1.1 200 OK
dget-HTTP>>> HTTP/1.1 200 OK
DeleGate-Ver: 9.9.13 (delay=0)
DeleGate-Ver: 9.9.13 (delay=0)
Server: nginx
Date: Tue, 30 Jun 2020 04:46:10 GMT
Content-Length: 372
Last-Modified: Fri, 31 Oct 2014 06:40:10 GMT
ETag: "174-506b244b1c680"
Accept-Ranges: bytes
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Content-Type-Comment: guessed by DeleGate (^-^)/
Content-Type: application/octet-stream
Via: 1.1 - (DeleGate/9.9.13), 1.1 - (DeleGate/9.9.13)
Connection: close

  recv#    time    total-bytes  bytes/s
    1      0.0         372      732183
13:46:10  1: /wits-more.jp/delegate/ftp/pub/DeleGate/COPYRIGHT [200][applicati
1 GET / 0.049 seconds = 20.33 GET/second 372 bytes (0.049/0.049/0.049)
```

開発：めでたし。



基盤：ブラウザもちゃんと認識してくれますね。

社長：めでたし。お昼に行きましょう。

\* \* \*

社長：あー食った飲んだで1,500円。予は満足じゃ。

開発：さたぽんのおばちゃん、店がすいてる時にビール頼むと一品サービスしてくれますね。あれはふつうの飲み屋なら、400円くらいの価値はあります。

社長：いやそれで、途中で出ていった中年男性が「また来ます」って言ったら見送りに出たおばちゃんが「あい、またきます」って返事してて、幼児の言語学習過程を見るようでした。

開発：あの男性はおばちゃんを女性として認識している感じがしましたね。

基盤：おばちゃんは実は女性だった！

社長：一般的にはそうでしょう。

開発：バグズ・ライフのてんとう虫が子供におばちゃんて呼ばれてましたね（笑）

経理：昨日のカードの使用履歴を見ますと、アマゾンVM20台一ヶ月分が一瞬で消費されてますが。

社長：いや昨日は久しぶりにボウリングの試合に出ましてね。ヒゲがちょっとウケましたw。コロナで試合が無くなってから2ヶ月？何も運動してなかったもんで、火星みたいな状態なわけですよ。それで15ポンドの鉛のように重いボールを持ったり投げたりするのって、最初は恐怖感がありましたね。スピードが16km/h出なかったりとか。だけど本番になったら出端から裏ターキーだったり。まさにビギナーズラックw。ていうかあのまま変えずに投げてたらスコア的には良かったのかも知れないけど、恥ずかしいから表に入れたくなくなってしまうわけです。すごく凸凹したけどこんな状態で200も出るもんだなと関心しました。チームも好調で、全体の2位で今季を終えました。だけど3ゲーム目にもう体力というか握力の限界みたいになってしまい。今日もまだ手首あたりにだるさが残っています。昔は30ゲームくらい平気で投げたものですが…

社長：まあそれで無事に試合を乗り切ったごほうびに、ボウリング場の向かいの台湾料理の店で祝杯を挙げたわけです。あの店では致死量まで飲み食いしても5,000円を超える心配はありません。でそこから代行なんですけど「7kmで2,500円です」で告げられた時。以前はなんとも思わなかったのですが、アマゾンVM5台ぶんかぁと思ったら、ずっしりと来ましたね。

基盤：まあそれを言えばタバコひと箱でVM1ヶ月動くわけですからね。

開発：基本、気晴らしに一番お金がかかると。

社長：それでかえりがけにウエルシアでタバコなんかを買ったのですが、支払いは非接触でできますとカードのメールにあったので、試したんですが、ダメでした。たぶんコンビニでの話なのかな。

経理：そうするともうナナコも不要になりますね。

開発：さらばななこ、そしておにぎり銀行。

社長：それで4月からずっと不愉快なのですが、レジ袋は有料になりますがどれになさいますか？と聞かれるわけです。買った分がちょうど入る袋に決まってるし、どのサイズが適切なのか、そっちで判断してくれよと思うわけです。で「ちょうど良いサイズのw」って答えるんですが、それももう飽きたというかうんざりです。

社長：ああそれで、カップ麺の棚を眺めたところ、100円以下、150円前後、200円超という3クラスがあることに気づきました。凄麺は200円、カップラーメンが150円、辛ラーメンとかが100円、そういう棲み分けのようです。

開発：そういえばトイレのLED電球が海王星状態になってしまいました。



開発：ボイジャーになった気分です。

社長：それ、買ってくるの忘れました。お出かけ前にポチポチっとして袋に入れといてくれたら良いのに。まああそこがそういう気の利いたサービスできるとは思えないですが。

基盤：というか、都会ならアマゾンで宅配してくれるのでは。

開発：いや、それをやったら足腰退化してまじ火星人になっちゃうでしょう。

\* \* \*

— 2020-0630 SatoxITS