SYNACKTIV

Frinet

Reverse-engineering using Frida & Tenet

SSTIC 2024

whoami

SYNACKTIV

- **Martin Perrier**
- **Louis Jacotot**

- **Security researchers @Synacktiv**
  - Offensive Security
  - +170 ninjas
  - We are hiring!

Context

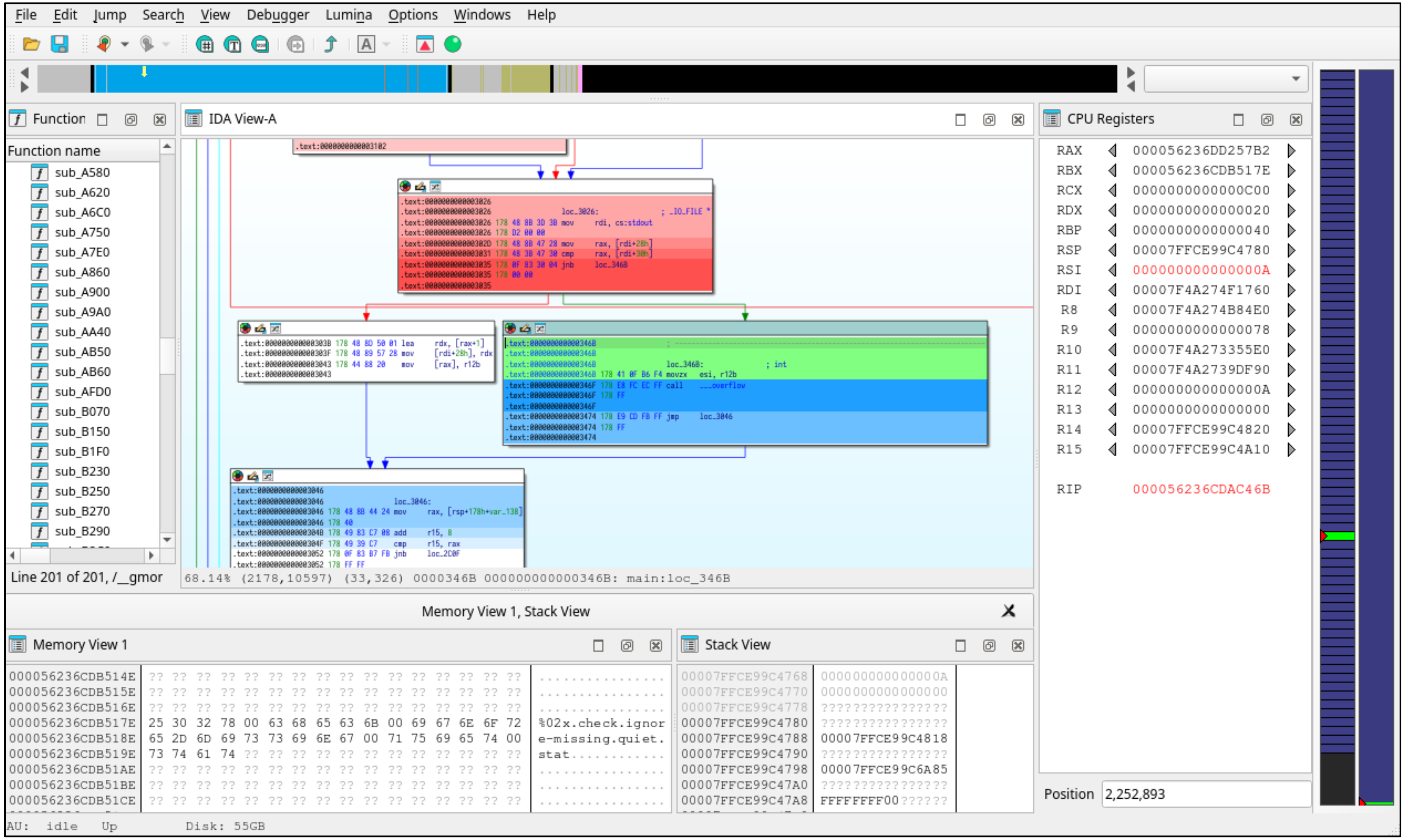
- **Native reverse-engineering approaches**
  - Static analysis
    - Disassembly
    - Decompilation
    - Understanding of actual behavior can be complex
  - Dynamic analysis
    - Debugging
    - Instrumentation
    - Focuses on details, missing comprehension of big picture
- **Combining both approaches**
  - Execution trace exploration
    - Study of complex programs
    - Root-cause analysis
    - Attack surface exploration

Tenet execution trace viewer

- **IDA Pro Plugin**
  - By Markus Gaasedelen
  - Won *Hex-Rays' 2021 Plug-In Contest*
  - Graphical interface
- **Text-based trace format**

```
rip=0x4000009c, rax=0x7f880100, rsp=0x7f8800ac
rip=0x4000009f, rbx=0x1337
rip=0x400000a2, mw=0x7f880100:1337
```

Tenet execution trace viewer



Function name
sub_A580
sub_A620
sub_A6C0
sub_A750
sub_A7E0
sub_A860
sub_A900
sub_A9A0
sub_AA40
sub_AB50
sub_AB60
sub_AFD0
sub_B070
sub_B150
sub_B1F0
sub_B230
sub_B250
sub_B270
sub_B290

Line 201 of 201, /__gmor

```
.text:0000000000003102

.text:0000000000003026
.text:0000000000003026                          loc_3026:               ; _IO_FILE *
.text:0000000000003026 178 48 8B 3D 38 mov      rdi, cs:stdout
.text:0000000000003026 178 D2 00 00
.text:000000000000302D 178 48 8B 47 28 mov      rax, [rdi+28h]
.text:0000000000003031 178 48 3B 47 30 cmp      rax, [rdi+30h]
.text:0000000000003035 178 0F 83 30 04 jnb      loc_346B
.text:0000000000003035 178 00 00
.text:0000000000003035

.text:000000000000303B 178 48 8D 50 01 lea      rdx, [rax+1]
.text:000000000000303F 178 48 89 57 28 mov      [rdi+28h], rdx
.text:0000000000003043 178 44 88 20    mov      [rax], r12b
.text:0000000000003043

.text:000000000000346B                          ; ---------------------------------------
.text:000000000000346B
.text:000000000000346B                          loc_346B:               ; int
.text:000000000000346B 178 41 0F B6 F4 movzx    esi, r12b
.text:000000000000346F 178 E8 FC EC FF call     ___overflow
.text:000000000000346F 178 FF
.text:000000000000346F
.text:0000000000003474 178 E9 CD FB FF jmp      loc_3046
.text:0000000000003474 178 FF
.text:0000000000003474

.text:0000000000003046
.text:0000000000003046                          loc_3046:
.text:0000000000003046 178 48 8B 44 24 mov      rax, [rsp+178h+var_138]
.text:0000000000003046 178 40
.text:000000000000304B 178 49 83 C7 08 add      r15, 8
.text:000000000000304F 178 49 39 C7    cmp      r15, rax
.text:0000000000003052 178 0F 83 B7 FB jnb      loc_2C0F
.text:0000000000003052 178 FF FF
```

```
68.14% (2178,10597) (33,326) 0000346B 000000000000346B: main:loc_346B
```

RAX	000056236DD257B2
RBX	000056236CDB517E
RCX	0000000000000C00
RDX	0000000000000020
RBP	0000000000000040
RSP	00007FFCE99C4780
RSI	000000000000000A
RDI	00007F4A274F1760
R8	00007F4A274B84E0
R9	0000000000000078
R10	00007F4A273355E0
R11	00007F4A2739DF90
R12	000000000000000A
R13	0000000000000000
R14	00007FFCE99C4820
R15	00007FFCE99C4A10
RIP	000056236CDAC46B

Position 2,252,893

```
000056236CDB514E ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
000056236CDB515E ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
000056236CDB516E ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
000056236CDB517E 25 30 32 78 00 63 68 65 63 6B 00 69 67 6E 6F 72  %02x.check.ignor
000056236CDB518E 65 2D 6D 69 73 73 69 6E 67 00 71 75 69 65 74 00  e-missing.quiet.
000056236CDB519E 73 74 61 74 ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  stat............
000056236CDB51AE ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
000056236CDB51BE ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
000056236CDB51CE ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??  ................
```

```
00007FFCE99C4768  000000000000000A
00007FFCE99C4770  0000000000000000
00007FFCE99C4778  ????????????????
00007FFCE99C4780  ????????????????
00007FFCE99C4788  00007FFCE99C4818
00007FFCE99C4790  ????????????????
00007FFCE99C4798  00007FFCE99C6A85
00007FFCE99C47A0  ????????????????
00007FFCE99C47A8  FFFFFFFF00??????
```

```
AU:  idle    Up         Disk: 55GB
```

Tenet trace generation

- **Existing tracers**
  - Intel Pin (x86 only)
  - QEMU Plug-In (emulation)
- **Limits**
  - Many platforms are not supported
    - Mobile devices, non-x86 architectures
  - Tracing a specific portion of code is hard
- **Need for a new tracer**
  - With support for Android, iOS, Linux, Windows… x86/64, arm(64)
  - Using already existing tooling

Frida tracer

- FRIDA
  - Dynamic instrumentation toolkit
  - Frida Stalker
    - Basic block recompilation on the fly
    - Tracing through callbacks on each instruction
    - JavaScript callback (slow!) or native code...

- **Stalker with native callback (CModule)**
  - Records register values, memory accesses
  - Support for x86/64, arm(64)
  - Outputs Tenet trace file

Frida tracer

- **Command line tool**
  - Can generate traces for Android/iOS/Linux/Windows!
  - Locally or through Frida Server over USB/network
  - Spawn/attach process
  - Trace provided function address (can be `main` entry point)
  - Works out of the box (no configuration needed in most cases)

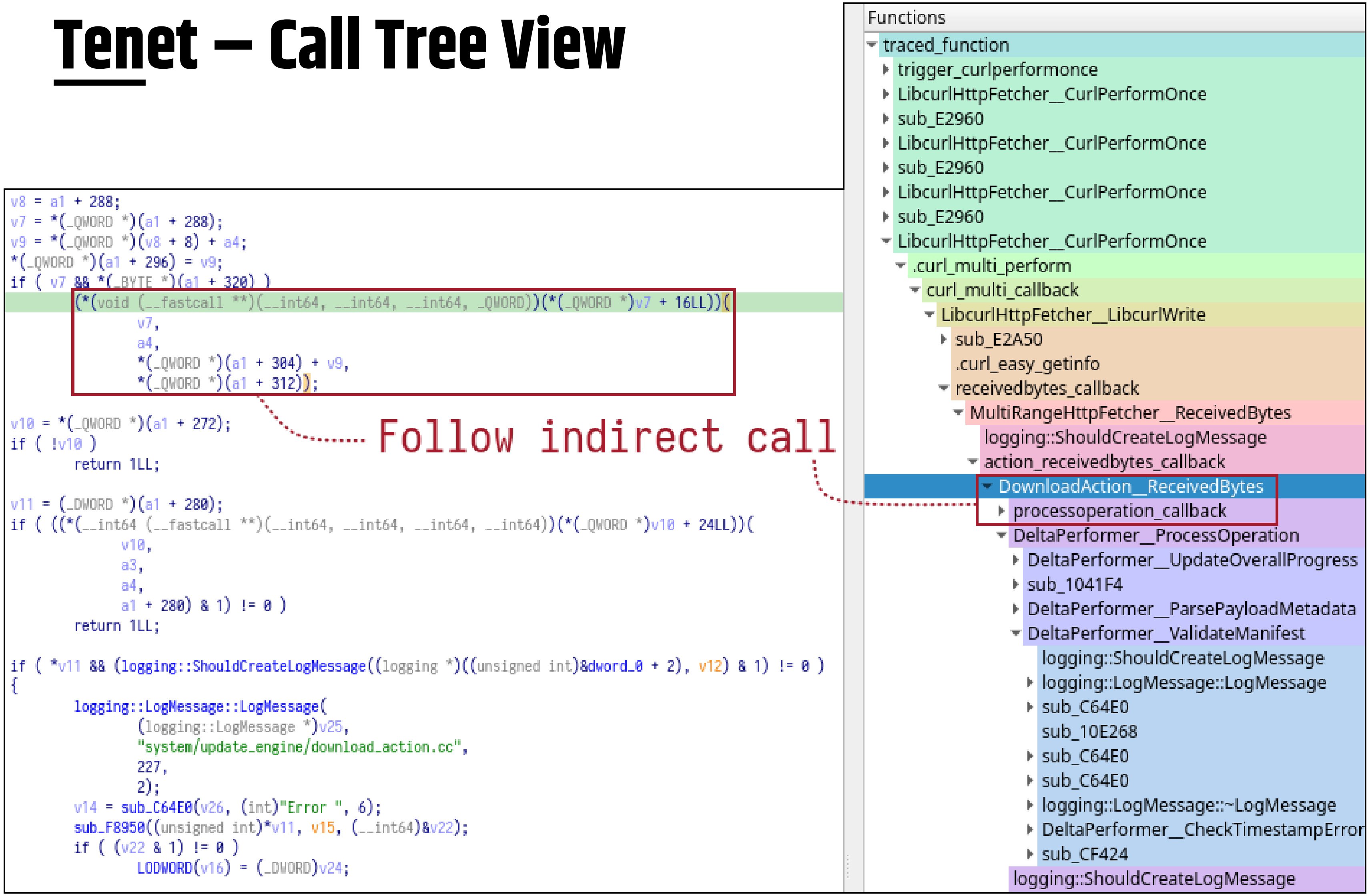
- **Example**

```
$ python3 tracer.py -U attach update_engine update_engine 0xe2fac
```

Tenet – New major features

- **Call Tree View**
  - High-level view of execution flow
    - Like a call stack, but for the whole trace
- **Memory Search**
  - Search for a pattern in space and time

Tenet – Call Tree View



```
v8 = a1 + 288;
v7 = *(_QWORD *)(a1 + 288);
v9 = *(_QWORD *)(v8 + 8) + a4;
*(_QWORD *)(a1 + 296) = v9;
if ( v7 && *(_BYTE *)(a1 + 320) )
    (*(void (__fastcall **)(__int64, __int64, __int64, _QWORD))(*(_QWORD *)v7 + 16LL))(
        v7,
        a4,
        *(_QWORD *)(a1 + 304) + v9,
        *(_QWORD *)(a1 + 312));

v10 = *(_QWORD *)(a1 + 272);
if ( !v10 )
    return 1LL;

v11 = (_DWORD *)(a1 + 280);
if ( ((*(__int64 (__fastcall **)(__int64, __int64, __int64, __int64))(*(_QWORD *)v10 + 24LL))(
        v10,
        a3,
        a4,
        a1 + 280) & 1) != 0 )
    return 1LL;

if ( *v11 && (logging::ShouldCreateLogMessage((logging *)((unsigned int)&dword_0 + 2), v12) & 1) != 0 )
{
    logging::LogMessage::LogMessage(
        (logging::LogMessage *)v25,
        "system/update_engine/download_action.cc",
        227,
        2);
    v14 = sub_C64E0(v26, (int)"Error ", 6);
    sub_F8950((unsigned int)*v11, v15, (__int64)&v22);
    if ( (v22 & 1) != 0 )
        LODWORD(v16) = (_DWORD)v24;
```

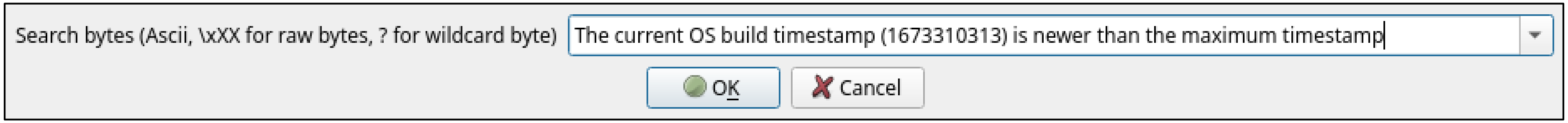
Follow indirect call

```
Functions
▾ traced_function
  ▸ trigger_curlperformonce
  ▸ LibcurlHttpFetcher__CurlPerformOnce
  ▸ sub_E2960
  ▸ LibcurlHttpFetcher__CurlPerformOnce
  ▸ sub_E2960
  ▸ LibcurlHttpFetcher__CurlPerformOnce
  ▸ sub_E2960
  ▾ LibcurlHttpFetcher__CurlPerformOnce
    ▾ .curl_multi_perform
      ▾ curl_multi_callback
        ▾ LibcurlHttpFetcher__LibcurlWrite
          ▸ sub_E2A50
            .curl_easy_getinfo
          ▾ receivedbytes_callback
            ▾ MultiRangeHttpFetcher__ReceivedBytes
                logging::ShouldCreateLogMessage
              ▾ action_receivedbytes_callback
                ▾ DownloadAction__ReceivedBytes
                  ▸ processoperation_callback
                  ▾ DeltaPerformer__ProcessOperation
                    ▸ DeltaPerformer__UpdateOverallProgress
                    ▸ sub_1041F4
                    ▸ DeltaPerformer__ParsePayloadMetadata
                    ▾ DeltaPerformer__ValidateManifest
                        logging::ShouldCreateLogMessage
                      ▸ logging::LogMessage::LogMessage
                      ▸ sub_C64E0
                        sub_10E268
                      ▸ sub_C64E0
                      ▸ sub_C64E0
                      ▸ logging::LogMessage::~LogMessage
                      ▸ DeltaPerformer__CheckTimestampError
                      ▸ sub_CF424
                    logging::ShouldCreateLogMessage
```

Tenet – Memory Search

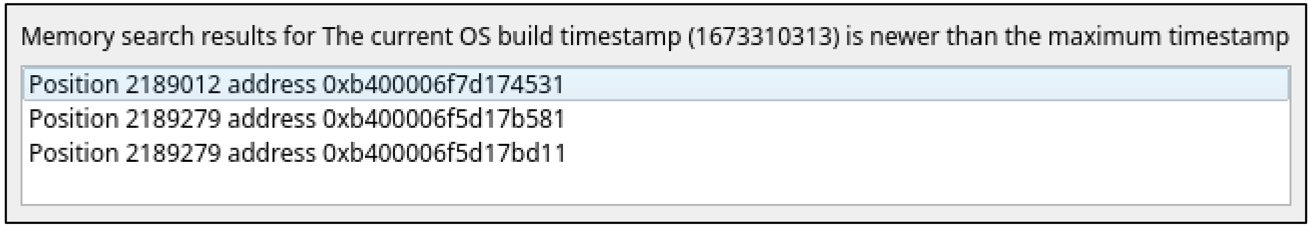
SYNACKTIV

1.



Search bytes (Ascii, \xXX for raw bytes, ? for wildcard byte) The current OS build timestamp (1673310313) is newer than the maximum timestamp

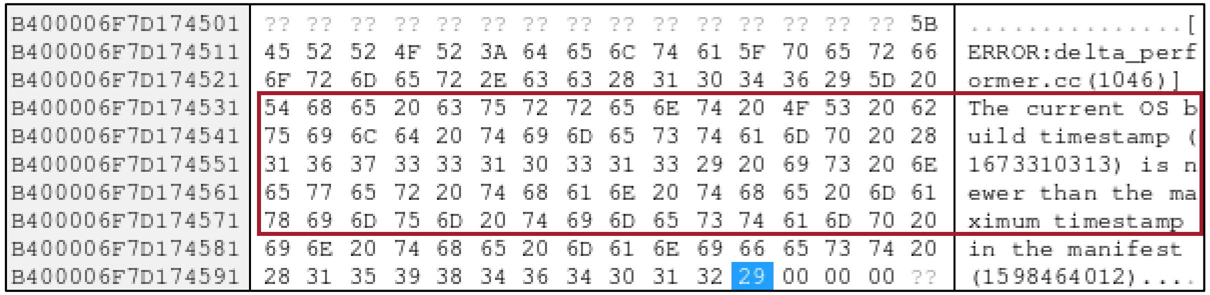
2.



Memory search results for The current OS build timestamp (1673310313) is newer than the maximum timestamp

```
Position 2189012 address 0xb400006f7d174531
Position 2189279 address 0xb400006f5d17b581
Position 2189279 address 0xb400006f5d17bd11
```

3.



```
B400006F7D174501  ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? 5B  ...............[
B400006F7D174511  45 52 52 4F 52 3A 64 65 6C 74 61 5F 70 65 72 66  ERROR:delta_perf
B400006F7D174521  6F 72 6D 65 72 2E 63 63 28 31 30 34 36 29 5D 20  ormer.cc(1046)] 
B400006F7D174531  54 68 65 20 63 75 72 72 65 6E 74 20 4F 53 20 62  The current OS b
B400006F7D174541  75 69 6C 64 20 74 69 6D 65 73 74 61 6D 70 20 28  uild timestamp (
B400006F7D174551  31 36 37 33 33 31 30 33 31 33 29 20 69 73 20 6E  1673310313) is n
B400006F7D174561  65 77 65 72 20 74 68 61 6E 20 74 68 65 20 6D 61  ewer than the ma
B400006F7D174571  78 69 6D 75 6D 20 74 69 6D 65 73 74 61 6D 70 20  ximum timestamp 
B400006F7D174581  69 6E 20 74 68 65 20 6D 61 6E 69 66 65 73 74 20  in the manifest 
B400006F7D174591  28 31 35 39 38 34 36 34 30 31 32 29 00 00 00 ??  (1598464012)....
```

Demonstration

SYNACKTIV

- **Scenario**
  - Android OTA update service (`update_engine`)
  - Let's pretend there is no public source code
  - Can we downgrade the Android version? (spoiler: no)

- **Steps**
  - 1. Get a *Pixel 4* and an OTA firmware older than the one installed
  - 2. Find the handler function address of the service (`0xe2fac`)
  - 3. Launch Frinet Tracer on this function
  - 4. Trigger an update with the old OTA firmware

Demonstration

- **Error!**

```
update_engine: [ERROR:delta_performer.cc(1046)] The current OS build timestamp
(1673310313) is newer than the maximum timestamp in the manifest (1598464012)
```

- **What if we modify the timestamp?**
  - Problem: we don't know where it is located in the OTA file!
    - Searching for `1598464012` (ascii, hex...) in the OTA file does not work
    - We do not have time to reverse-engineer the format
  - Solution: Frinet

Demonstration

Demonstration

```
0001c420: 9d17 8210 daef 9a06 d168 708c c09a fa05  .........hp.....
0001c430: 7a4a 0a46 0a19 676f 6f67 6c65 5f64 796e  zJ.F..google_dyn
0001c440: 616d 6963 5f70 6172 7469 7469 6f6e 7310  amic_partitions.
```

$\Downarrow$

```
0001c420: 9d17 8210 daef 9a06 d168 70ff ffff ffff  .........hp.....
0001c430: 7a4a 0a46 0a19 676f 6f67 6c65 5f64 796e  zJ.F..google_dyn
0001c440: 616d 6963 5f70 6172 7469 7469 6f6e 7310  amic_partitions.
```

Demonstration

- **Error!**

```
update_engine: [ERROR:payload_metadata.cc(214)] Manifest hash verification failed.
update_engine: [ERROR:delta_performer.cc(372)] Mandatory metadata signature validation failed
```

- **Modifying the timestamp did not work**
  - There is a signature mechanism
  - The next step would be to study it

Demonstration

Conclusion

- **Available now**
  - Any feedback is welcome
  - `https://github.com/synacktiv/frinet`
    - Frida Tracer
    - Modified Tenet Plug-In in subrepository
  - Incoming native backend & more
    - Python backend is too slow on larger traces
    - Rust library, with Python bindings (can be used in Python scripts without IDA)

- **Merci !**

SYNACKTIV

https://www.linkedin.com/company/synacktiv

https://twitter.com/synacktiv

https://synacktiv.com