Breakin 'Em All

Overcoming



Anti-Cheating Mechanism

The Release of Pokémon Go



An Unexpected Journey



An Unexpected Journey



What Did You Say Your Name Was?



@taltechtreks.com

@TalSkverer

/in/reverser

taltechtreks.com

- Tal Skverer
- Decade+ of Security Research
- Head of Security @ Astrix
- Automotive Hacking, Reversing Malware, Protocol Analysis, Web Research, Identity Security
- On stage at DEFCON 31, presented Google OAuth 0-day

Badges

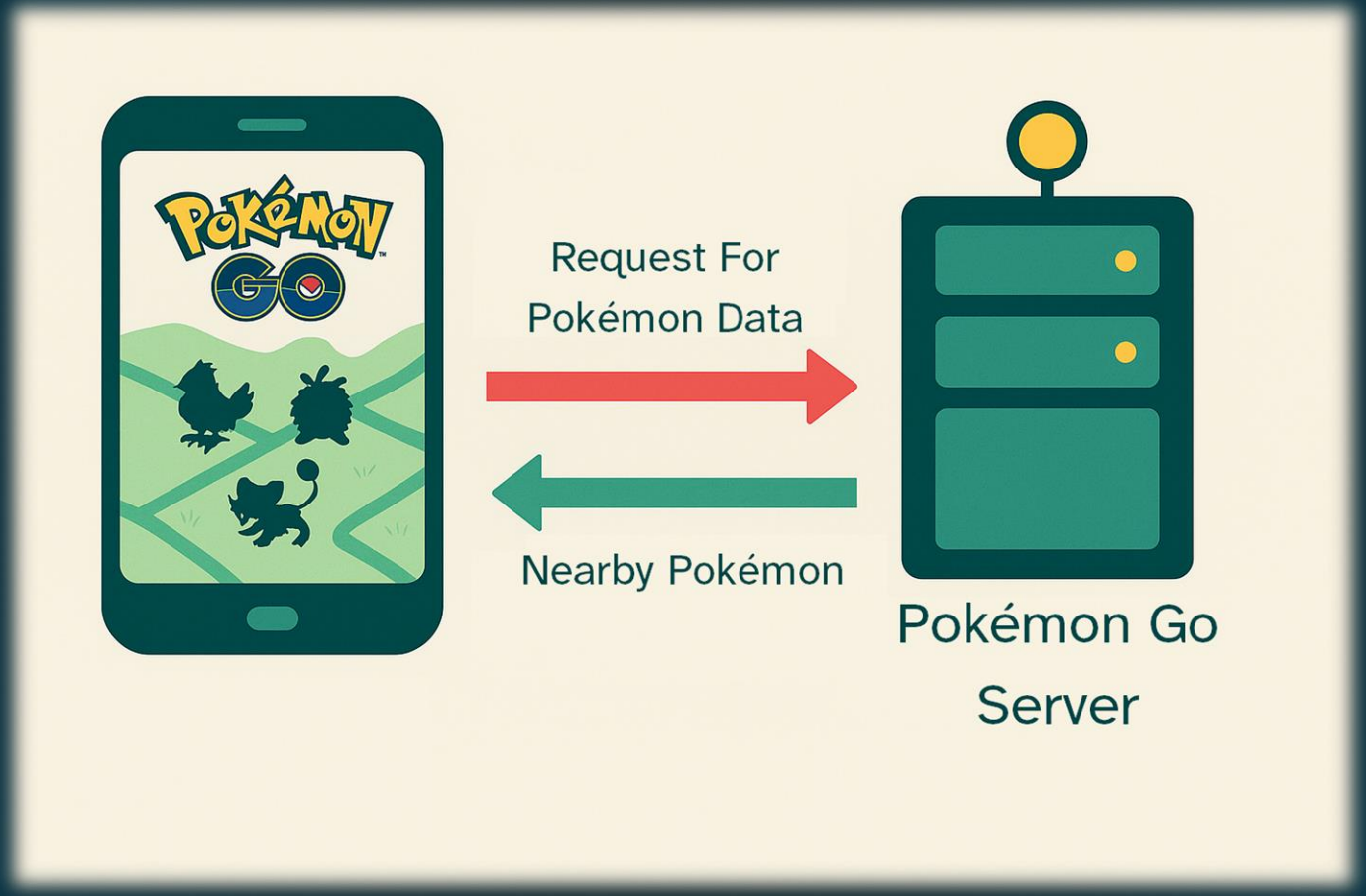




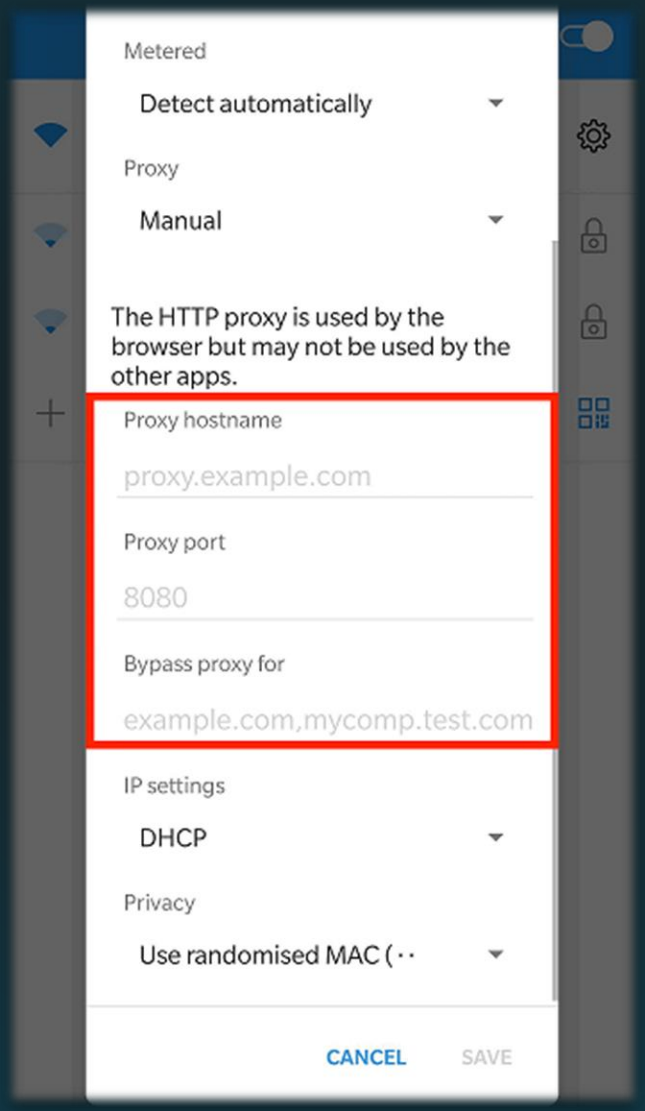
Disclaimers

- I'm speaking as an independent security researcher. My views do **not** represent my employer.
- Based on research from 9 years ago. Security has advanced greatly, so the techniques shown no longer work today. All materials presented are based on publicly-available sources (reference in final slide)

Humble Beginnings



Sniffing Traffic



Sniffing Traffic

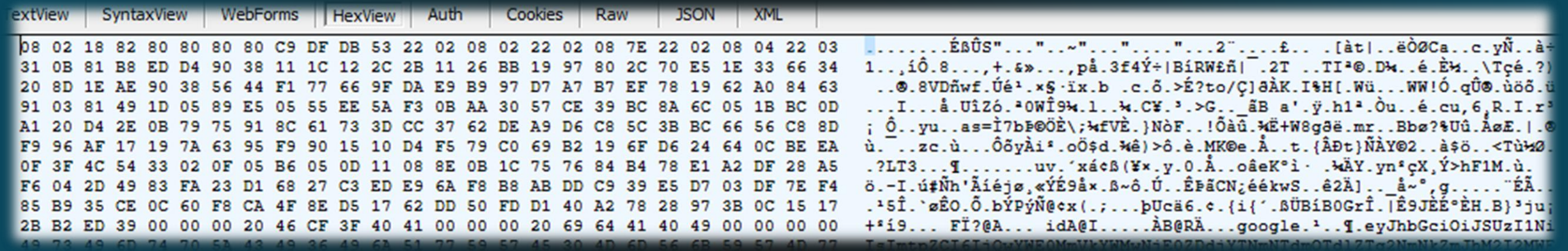
HTTPS	pgorelease.nianticlabs.com	/plfe/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc
HTTPS	pgorelease.nianticlabs.com	/plfe/552/rpc

Headers | TextView | SyntaxView | WebForms | HexView | Auth | Cookies | Raw

```
POST https://pgorelease.nianticlabs.com/plfe/rpc HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
User-Agent: Dalvik/2.1.0 (Linux; U; Android 6.0.1; A0001 Build/MHC19Q)
Host: pgorelease.nianticlabs.com
Accept-Encoding: gzip
Content-Length: 1712
```



Sniffing Traffic



Protobuf

extView | SyntaxView | WebForms | HexView | Auth | Cookies | Raw | JSON | XML

```
08 02 18 82 80 80 80 80 C9 DF DB 53 22 02 08 02 22 02 08 7E 22 02 08 04 22 03
31 0B 81 B8 ED D4 90 38 11 1C 12 2C 2B 11 26 BB 19 97 80 2C 70 E5 1E 33 66 34
20 8D 1E AE 90 38 56 44 F1 77 66 9F DA E9 B9 97 D7 A7 B7 EF 78 19 62 A0 84 63
91 03 81 49 1D 05 89 E5 05 55 EE 5A F3 0B AA 30 57 CE 39 BC 8A 6C 05 1B BC 0D
A1 20 D4 2E 0B 79 75 91 8C 61 73 3D CC 37 62 DE A9 D6 C8 5C 3B BC 66 56 C8 8D
F9 96 AF 17 19 7A 63 95 F9 90 15 10 D4 F5 79 C0 69 B2 19 6F D6 24 64 0C BE EA
0F 3F 4C 54 33 02 0F 05 B6 05 0D 11 08 8E 0B 1C 75 76 84 B4 78 E1 A2 DF 28 A5
F6 04 2D 49 83 FA 23 D1 68 27 C3 ED E9 6A F8 B8 AB DD C9 39 E5 D7 03 DF 7E F4
85 B9 35 CE 0C 60 F8 CA 4F 8E D5 17 62 DD 50 FD D1 40 A2 78 28 97 3B 0C 15 17
2B B2 ED 39 00 00 00 20 46 CF 3F 40 41 00 00 00 20 69 64 41 40 49 00 00 00 00
```

Protocol Buffers

Protocol Buffers are language-neutral, platform-neutral extensible mechanisms for serializing structured data.

What Are Protocol Buffers?

Protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data – think XML, but smaller, faster, and simpler. You define how you want your data to be structured once, then you can use special generated source code to easily write and read your structured data to and from a variety of data streams and using a variety of languages.

Protobuf - Intro

```
syntax = "proto3";

message SearchRequest {
  string query = 1;
  int32 page_number = 2;
  optional int32 results_per_page = 3;
  optional repeated SearchOption search_options = 4;
}
```



Protobuf - Intro

```python
import requests
import search_request_pb2

request = search_request_pb2.SearchRequest()
request.query = "Lord of the Rings Books"
request.page_number = 1
request.results_per_page = 10

option1 = request.search_options.add()
option1.key = "language"
option1.value = "english"

option2 = request.search_options.add()
option2.key = "sort_order"
option2.value = "ascending"

serialized_data = request.SerializeToString()

response = requests.post(
    "https://api.example.com/search",
    data=serialized_data
)
```

`protoc` - Man Page

Protobuf compiler

Examples (TL;DR)

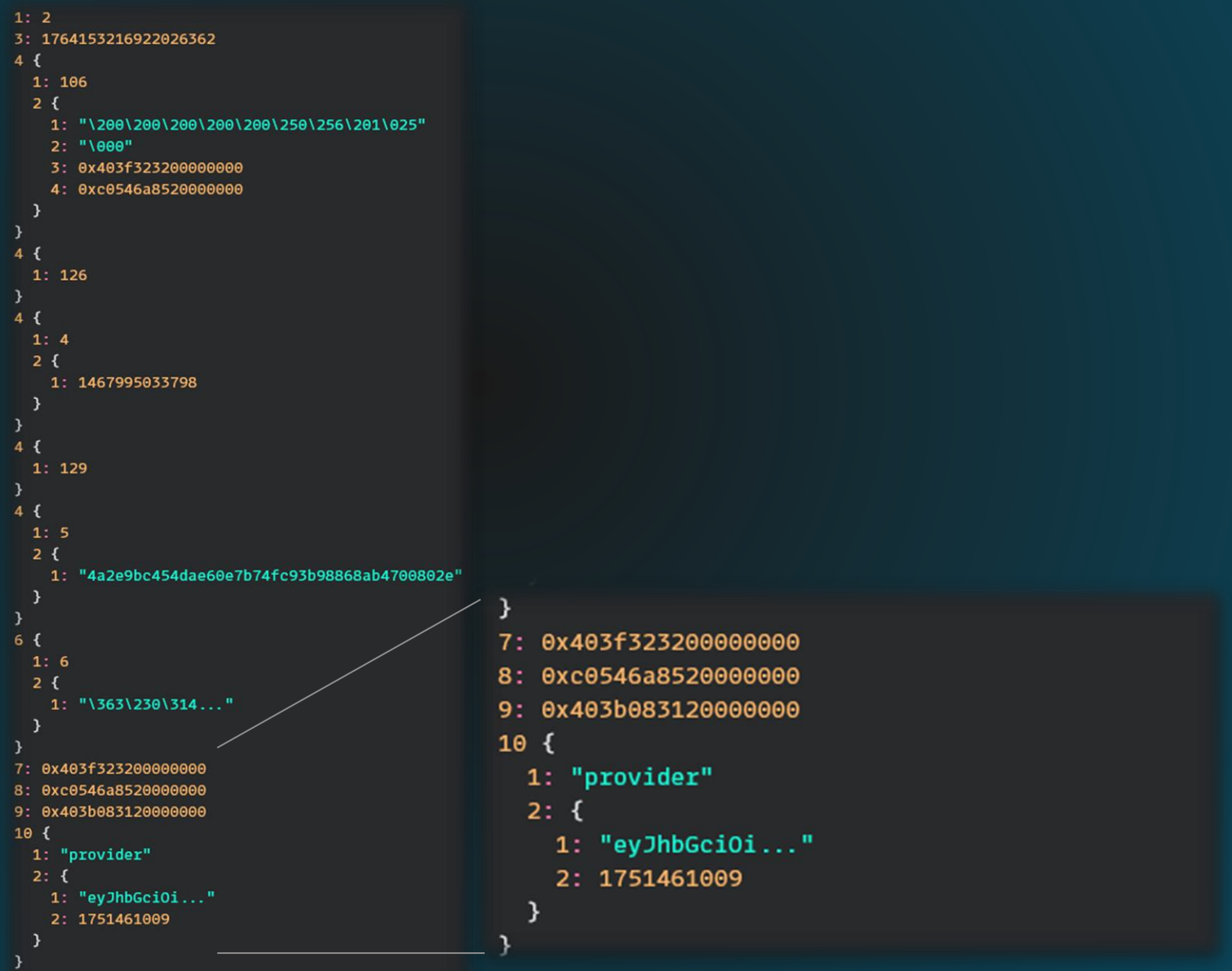
> Generate Python code from a `.proto` file:
> `protoc --python_out=path/to/output_directory input_file.proto`

> Decode a protocol message into raw tag/value pairs:
> `protoc --decode_raw < message.bin`

Protobuf - Intro

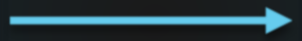
```
1: "Lord of the Rings Books",
2: 1,
3: 10
4: {
  1: "Order"
  2: "Ascending"
}
4: {
  1: "Language"
  2: "English"
}
```

Analyzing Pokémon Go Protocol



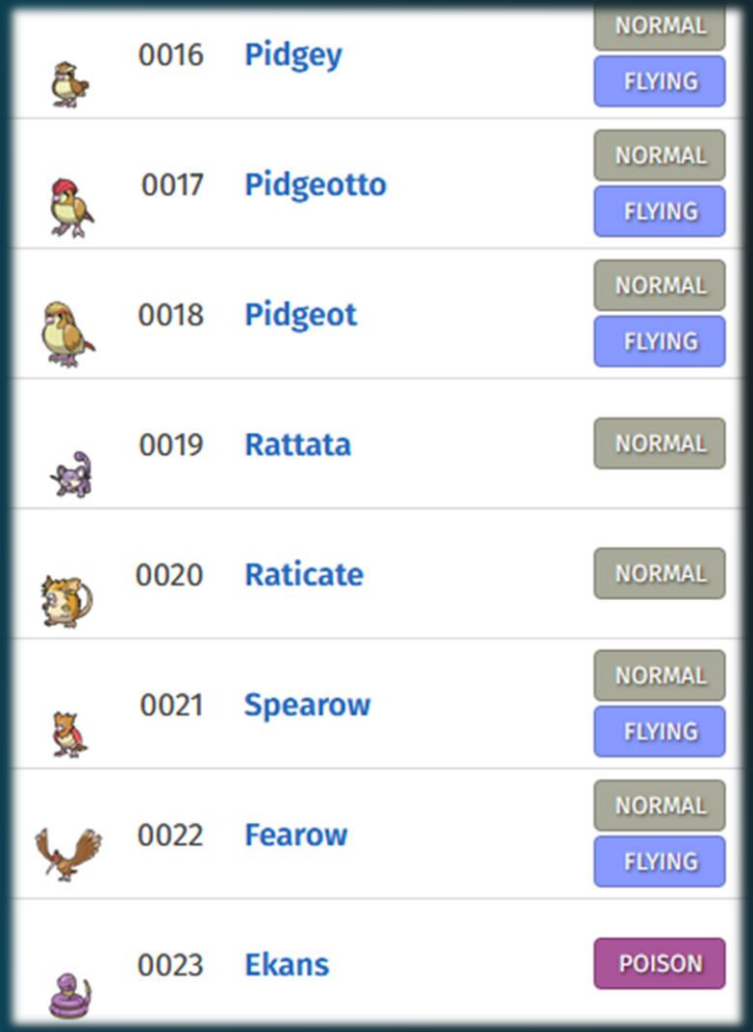
Analyzing Pokémon Go Protocol

```
status_code: 2
request_id: 1764153216922026362
requests: {
  request_type: 106
  request_bytes: -- ...
}
requests: {
  request_type: 126
}
requests: {
  request_type: 4
  request_bytes: --- ...
}
unknown6: {
  u6_unknown1: 6
  u6_unknown2: {
    u6_u2_unknown1_bytes: -- ...
  }
}
latitude: 31.19607543945312
longitude: -81.66437530517578
accuracy: 27.031999588012695
auth_info: {
  provider: "google"
  token: {
    content: "eyJhbGciOi..."
    expiration: 1751461009
  }
}
unknown12: 4590
```



```
status_code: OK
request_id: 1764153216922026362
responses: {
  response_bytes: -- ...
}
responses: {
  response_bytes: -- ...
}
responses: {
  response_bytes: -- ...
}
```

Request Type: GET_MAP_OBJECTS




```
map_cell {
  s2_cell_id: -8520171482234486784
  as_of_time_ms: 1467995224510
  fort {
    fort_id: "9feef60670b74df9871391b4b9e3ca24.16"
    last_modified_ms: 1467338329663
    latitude: 35.72228384017944
    longitude: 139.7710394859314
    enabled: true
    fort_type: CHECKPOINT
  }
  spawn_point {
    latitude: 35.72228384017531
    longitude: 139.7710394859677
  }
  wild_pokemon {
    encounter_id: 3736514205889371933
    last_modified_ms: 1467995224510
    latitude: 35.72228384017914
    longitude: 139.7710394859256
    spawn_point_id: "167fb95d5c"
    pokemon {
      pokemon_id: 16
    }
    time_till_hidden_ms: 232574
  }
  catchable_pokemon {
    spawnpoint_id: "1502bffd46b"
    encounter_id: 11920471590087242045
    pokedex_type_id: 23
    expiration_time_ms: 1467995429388
    latitude: 35.72228384013804
    longitude: 139.7710394524156
  }
```

Pokémon Go's Protocol – GET_MAP_OBJECTS





Creating a Pokéscanner

- Uses protobuf, the message is `RequestContainer`
- Request type `106` fetches map details
- Request needs a cell `ID`, responds with exact location

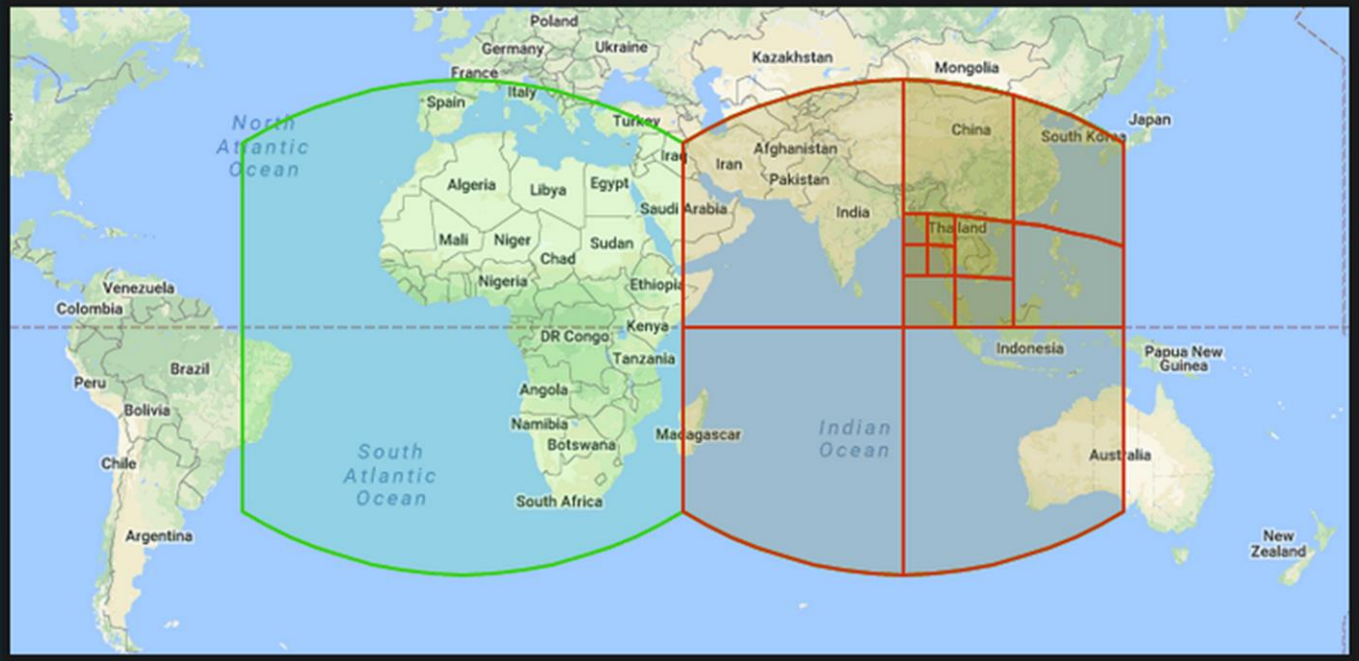
Creating a Pokéscanner

- Constructing a `RequestContainer`?
- Dealing with unknown Fields?
- Finding the Cell ID?

Creating a Pokéscanner

S2 Cells

The S2 library defines a framework for decomposing the unit sphere into a hierarchy of *cells*. Each cell is a quadrilateral bounded by four geodesics. The top level of the hierarchy is obtained by projecting the six faces of a cube onto the unit sphere, and lower levels are obtained by subdividing each cell into four children recursively. For example, the following image shows two of the six *face cells*, one of which has been subdivided several times:



S2 Geometry Calculator

Latitude

40.6341634

Longitude

-73.9932872

Level

15

S2 Cell ID

-8520171482234486 7

Copy S2 Cell ID

Open Google Maps

Creating a Pokéscanner

```
def _get_auth_ticket(self, jwt_token):
  # Creating a raw request
  req_env = RequestEnvelop_pb2.RequestEnvelop()
  req_env.status_code = 2
  req_env.rpc_id = Utils.randomize_rpc_id()
  # Must send at least one request, can be empty
  request = req_env.requests.add()
  request.request_type = Constants.GET_MAP_OBJECTS_REQUEST
  # Adding the authentication details
  req_env.auth_info.provider = "google"
  req_env.auth_info.token.contents = jwt_token
  req_env.unknown12 = 1036

  raw_data = req_env.SerializeToString()
  return NetUtil.request("POST", Constants.BASE_NIANTIC_URL, raw_data)
```

```
class GetMapObjects(object):
  def _handle_response(self, response):
    # ...

  def get(self, latitude, longitude, cell_ids):
    get_map_objects_request = GetMapObjectsRequest_pb2.GetMapObjectsRequest()
    for cell_id in cell_ids:
       get_map_objects_request.cell_id.append(cell_id.id())
       get_map_objects_request.since_time_ms.append(0)
    get_map_objects_request.latitude = latitude
    get_map_objects_request.longitude = longitude

    self.request.requests[0].message = get_map_objects_request.SerializeToString()
    data = self.request.SerializeToString()

    return self._handle_response(NetUtil.request("POST", self.url, data))
```

```
def get_map_objects(self, location):
  neighboring_cell_ids = Utils.get_neighbors(location.latitude, location.longitude)

  raw_request = self._create_raw_request()
  new_request = raw_request.requests.add()
  new_request.request_type = RequestEnums_pb2.GET_MAP_OBJECTS

  gmo = GetMapObjects.GetMapObjects(raw_request, self.url, self.logger)
  return gmo.get(location.latitude, location.longitude, neighboring_cell_ids)
```

Creating a Pokéscanner

```
{15240969903196846lL: {'catchable': False,
                       'distance': 200.0,
                       'id': 46,
                       'name': 'Paras'},
 7946361554302922531L: {'catchable': False,
                       'distance': 200.0,
                       'id': 19,
                       'name': 'Rattata'},
 4762505547286647693L: {'catchable': False,
                        'distance': 200.0,
                        'id': 19,
                        'name': 'Rattata'},
 1173204924034609801 3L: {'catchable': False,
                         'distance': 200.0,
                         'id': 46,
                         'name': 'Paras'},
 1268095952202863282 9L: {'catchable': True,
                         'disappear': 'Fri Jul 22 22:13:13 2016',
                         'distance': 200.0,
                         'id': 16,
                         'latitude': 3[redacted],
                         'longitude': 3[redacted],
                         'name': 'Pidgey',
                         'spawnpoint_id': u'1502b967f81',
                         'time_till_hidden': 624.074},
 1605898875214338378 9L: {'catchable': False,
                         'distance': 200.0,
                         'id': 41,
                         'name': 'Zubat'}}
```



A Challenger Appears



PogoDev

Unofficial PokemonGO Development

10 followers · Pallet Town · https://pogodev.org

1st day – 1am

The First Hurdle

r/pokemongodev • 9 yr. ago
keyphact MOD PogoDev Administrator

PokemonGO Current API Status

Discussion

Hi all,

As many of you have noticed, many scanners and APIs have stopped working and IOS app clients are being forced to update. The direct cause is unknown at this moment in time, but there are many people working to find a fix. It is not just you. Everything except the unmodified updated app appears to be having issues.

I've stickied this thread for discussion so as to stop the "My API is not working" and influx of re-posted links and discussions.

~~For Discord discussion for **devs** only, please use this invite: https://discord.gg/kcx5f We've decided to close this from the public in order to allow us to concentrate on the issue at hand and stop masses of people 1) stealing work and generating more effort for us by not answering questions and sending them our way 2) joining the conversation without adding much and derailing efforts.~~

Chat is open again for all to read.

Please use: https://discord.gg/dKTSHZC

> Updates

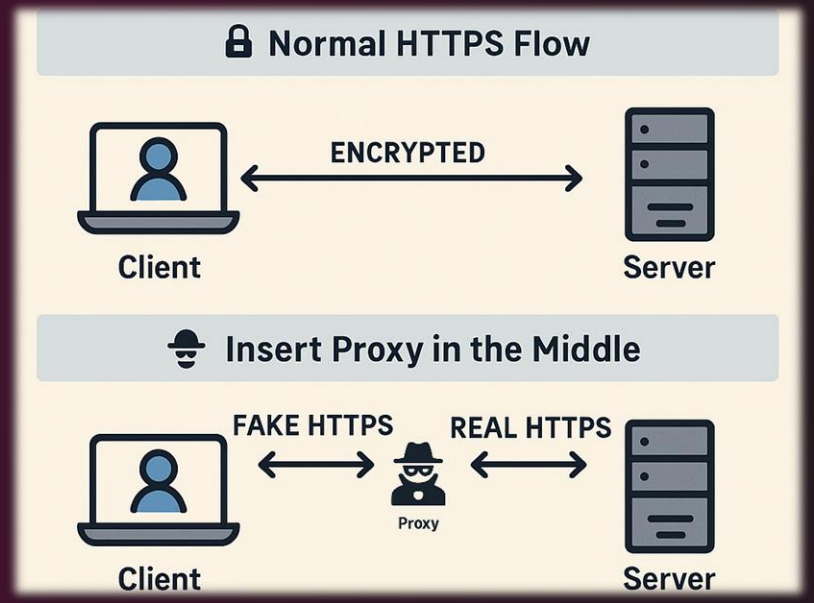
04/08/2016 - 00:49 GMT+1 : Logic and proto behind seem to have changed MapRequest, we're investigating.

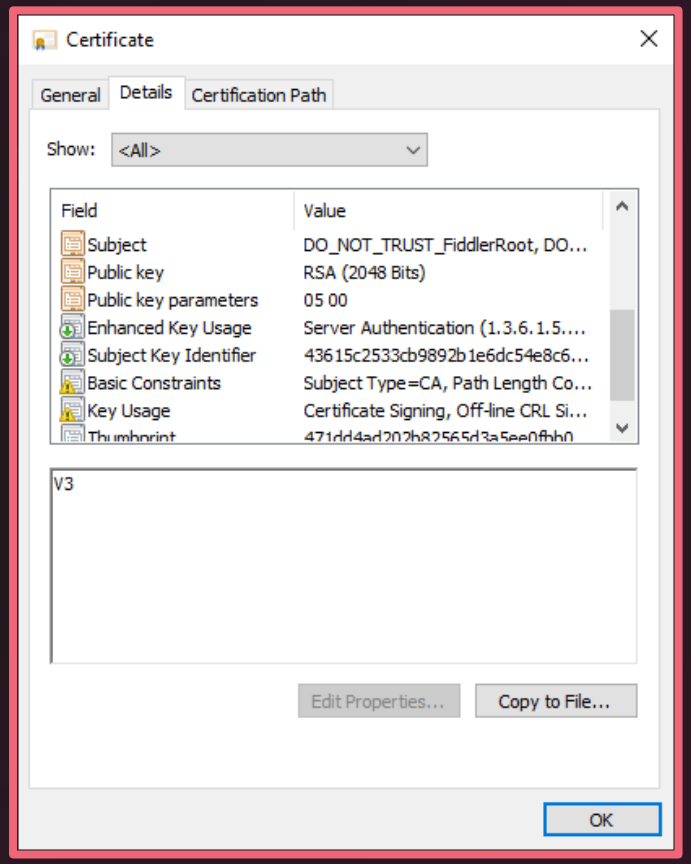
1st day – 1am

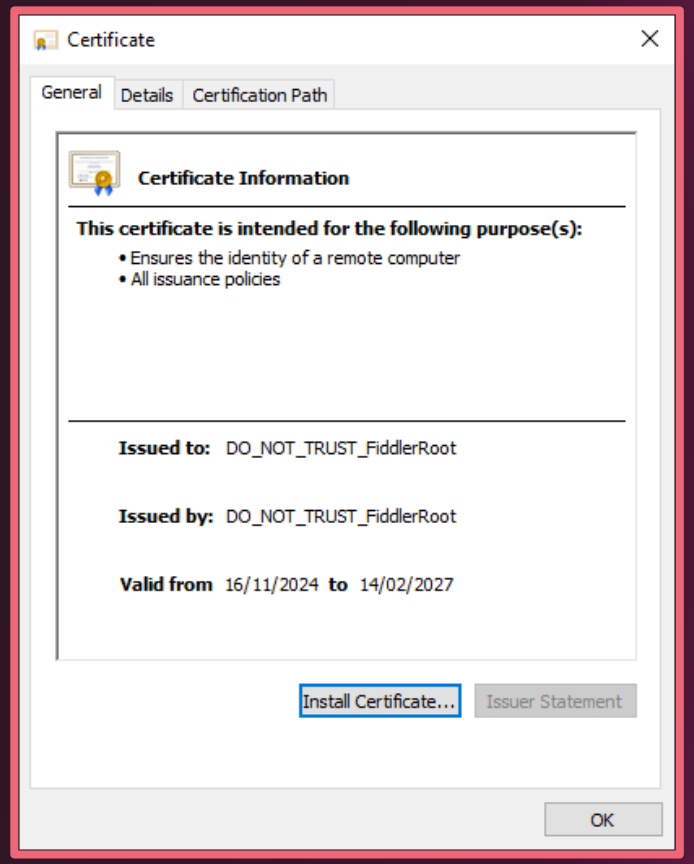
The First Hurdle

```java
public class NianticTrustManager extends ContextService implements X509TrustManager {
    private native void nativeCheckClientTrusted(X509Certificate[] x509CertificateArr, String str) throws CertificateException;

    private native void nativeCheckServerTrusted(X509Certificate[] x509CertificateArr, String str) throws CertificateException;

    private native X509Certificate[] nativeGetAcceptedIssuers();

    public NianticTrustManager(Context context, long nativeClassPointer) {
        super(context, nativeClassPointer);
    }

    @Override // javax.net.ssl.X509TrustManager
    public void checkClientTrusted(X509Certificate[] chain, String authType) throws CertificateException {
        synchronized (this.callbackLock) {
            nativeCheckServerTrusted(chain, authType);
        }
    }

    @Override // javax.net.ssl.X509TrustManager
    public void checkServerTrusted(X509Certificate[] chain, String authType) throws CertificateException {
        synchronized (this.callbackLock) {
            nativeCheckServerTrusted(chain, authType);
        }
    }
```

HTTPs Proxy - Intro







Technique: Dynamic RE



1st day – 1am

Bypassing Certificate Pinning

```
final Class NianticTrustManagerClass = lpparam.classLoader.loadClass("com.nianticlabs.nia.network.NianticTrustManager");

X509Certificate[] cert = new X509Certificate[0];

findAndHookMethod(NianticTrustManagerClass, "checkServerTrusted", cert.getClass(), String.class,
        new CheckServerTrustedHook()
);

final String ORIGINAL_CHAIN = "...";

// ...

    @Override
    protected void beforeHookedMethod(MethodHookParam param) throws Throwable {
        XposedBridge.log("Injecting Pokemon Go certificate trust chain");

        // Decode the stored original chain
        byte[] buffer = Base64.decode(ORIGINAL_CHAIN, Base64.DEFAULT);

        // Restore the original chain object and inject it into the call arguments
        param.args[0] = (X509Certificate[]) SerializationUtils.deserialize(buffer);
    }
```

Technique: Active Probing



1st day – 6am

Setting the Stage

```
message Unknown6 {
    int32 unknown1 = 1;
    Unknown2 unknown2 = 2;

    message Unknown2 {
      bytes unknown1 = 1;
    }
  }
```



Technique: Passive Analysis



1st day - 10am

Initial Discoveries

```
6 {
  1: 6
  2 {
    1: "\363\230\314\364.\201\303\312p\373\271\217\231\233\363qi8\352X\017i\343yP\245,\010PI\364\n\217Z\342L
    \201\330\317\211iT3\247\'\020I\272\t9\347i\034)\234\305\027\204\303\206\246B\263\333\265\017\340-]2\317
    \376b\250:\233\201=d\014\272\035\220\365]\265\277k\363\\K\023\307Z\327qHB{\013\251\223\300F\324\314:\332
    \255\030\327|\3032 Q\301\250\016\225\221\022\213\277q\035\306\341\371\353N\304\325\3449\234\007S\310~\031
    \205\223\225S2\243\311l\272\014\313\014\347P\344\0373\257\331\205\026\037p\236\202<\241)X\305M\215Rb\222B
    \320\020\354\261\227\353\2209\013\353\260\221\320\026g\2749\303\327e9`K\202J/\262\355T\316\335\355\t\021#
    \026\306e\356d\231\263qhzv\326\237\201\305\205,\0168\227\201#\241P\030\207\341@\026s\303\311\222p_\013S\240
    :\233\037\300\226\1776\320\237c:c\006qe\216X\3139\217\221|\354\366c^gju\214k\202\357\367\037$K\233"
  }
}
```

Android Apps - Intro

Android NDK

The Android NDK is a toolset that lets you implement parts of your app in native code, using languages such as C and C++. For certain types of apps, this can help you reuse code libraries written in those languages.

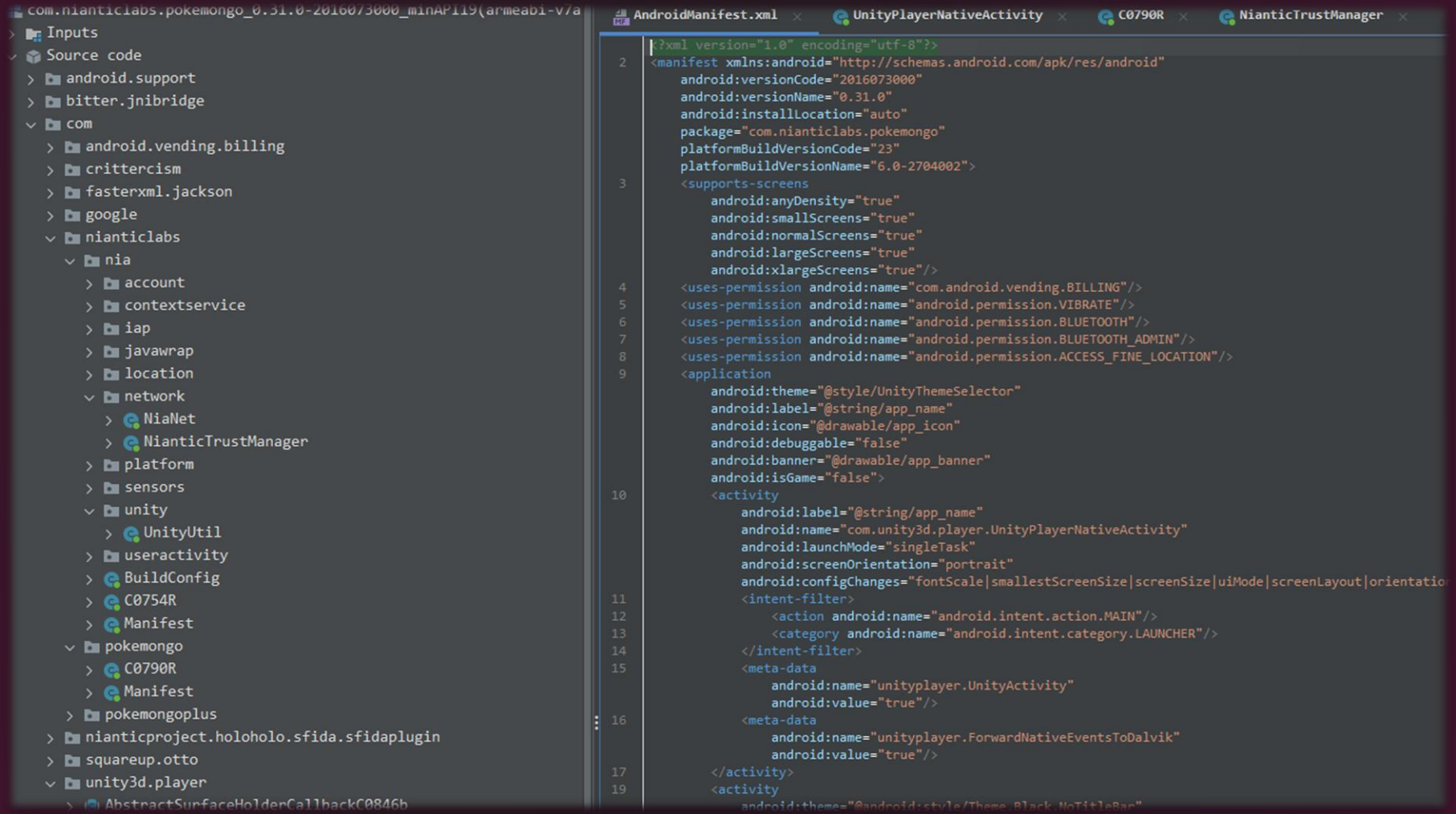
Get started

```
class MyActivity : Activity() {
  /**
   * Native method implemented in C/C++
   */
  external fun computeFoo()
}
```

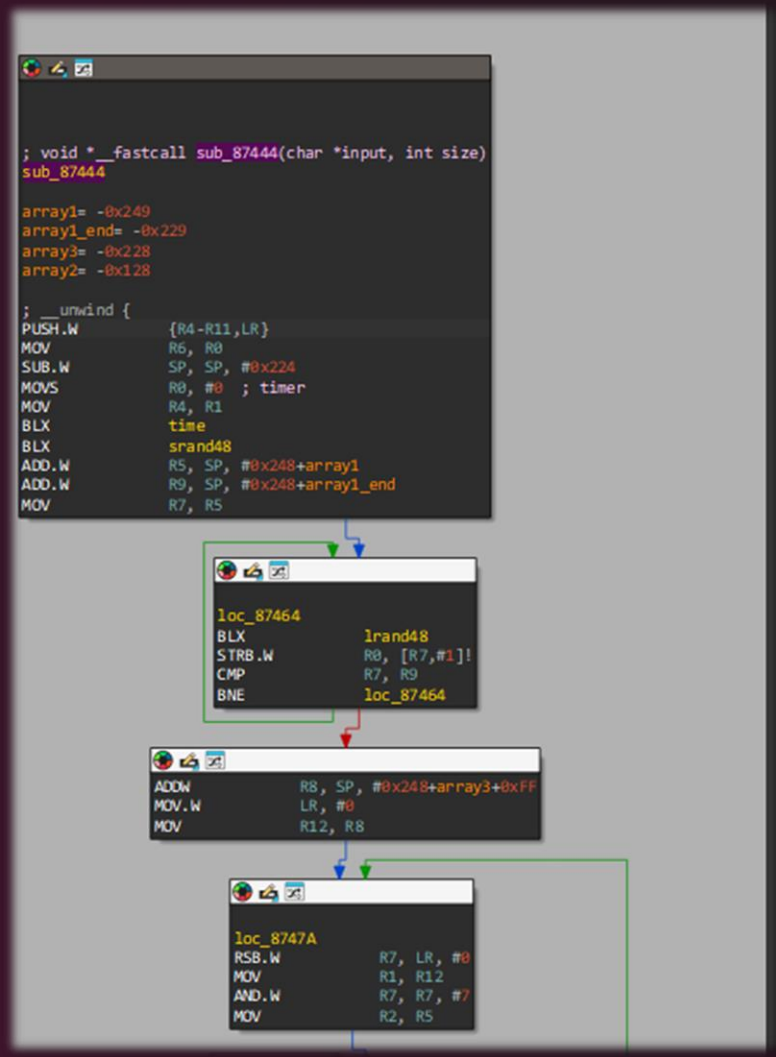
Technique: Static RE



Technique: Static RE



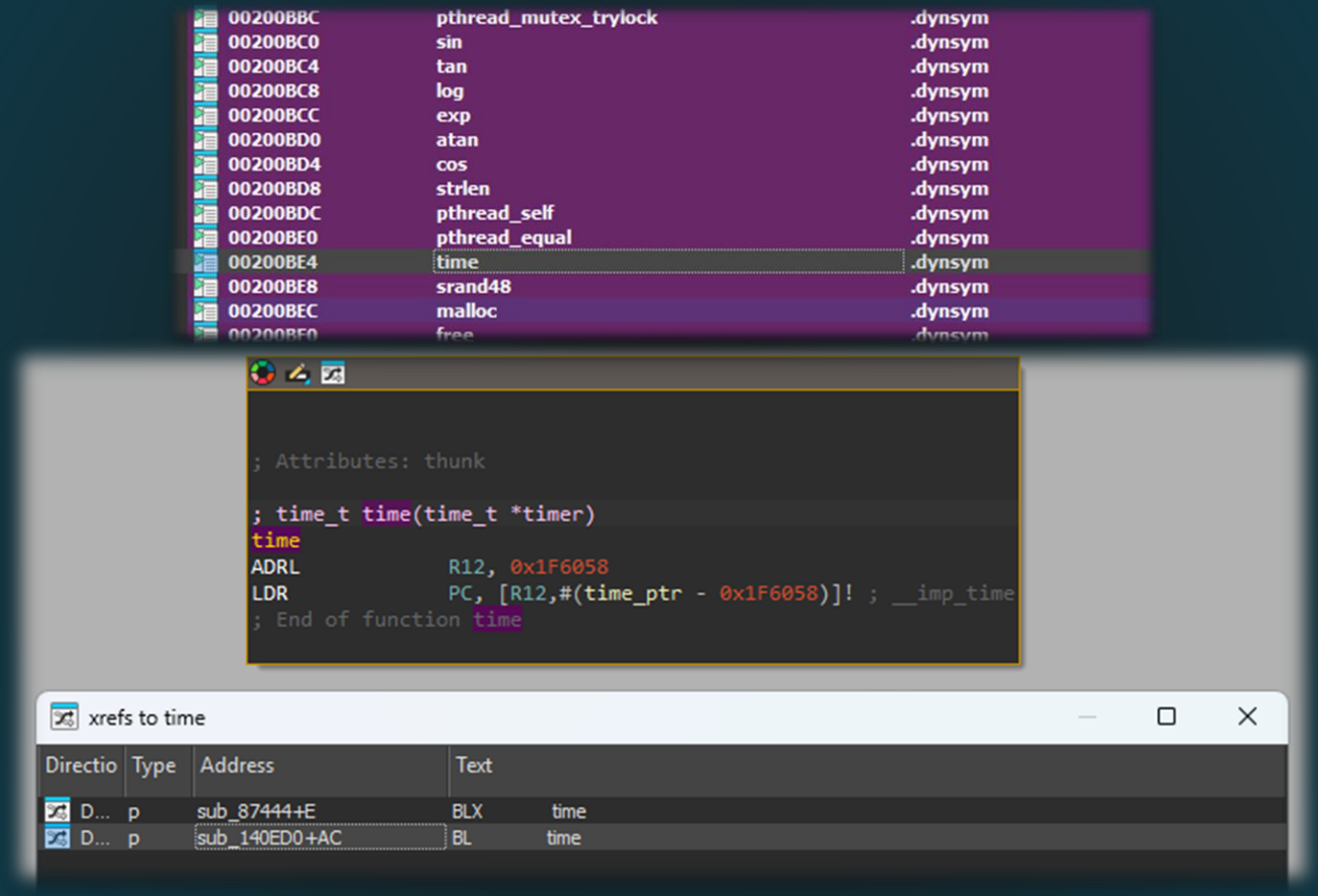
Technique: Static RE




```
1  void *__fastcall sub_87444(char *input, int size)
2  {
3    time_t time; // r0
4    char *array1_ptr_1; // r7
5    int i; // lr
6    char *array2_ptr; // r12
7    char *v8; // r1
8    char *array1_ptr; // r2
9    unsigned int current_array1_byte; // t1
10   char *input_after_header_ptr_or_current_input_block_ptr; // r10
11   int v12; // r1
12   int v13; // r2
13   int v14; // r3
14   int size_256_remainder; // r5
15   int final_size_without_header; // lr
16   unsigned int final_size; // r7
17   void *result; // r0
18   int v19; // r1
19   int v20; // r2
20   int v21; // r3
21   char *v22; // r3
22   char *v23; // lr
23   unsigned int current_block_start_index; // r11
24   char *current_input_block_ptr; // r1
25   char *array2_ptr_1; // r2
26   char current_array2_byte; // t1
27   char *v28; // r0
28   char array1[32]; // [sp+0h] [bp-249h] BYREF
29   char array1_end; // [sp+20h] [bp-229h] BYREF
30   char array3[256]; // [sp+21h] [bp-228h] BYREF
31   char array2[256]; // [sp+121h] [bp-128h] BYREF
32
33   time = ::time(0);
34   srand48(time);
35   array1_ptr_1 = array1;
36   do
37     *++array1_ptr_1 = lrand48();
38   while ( array1_ptr_1 != &array1_end );
39   i = 0;
40   array2_ptr = &array3[255];
```

1st day – 3pm

Pinpointing the Function



1st day – 8pm

The sigEncryptFunction

```
var sigEncryptFunctionPointer = nativeLibBaseAddress.add(0x87444);
console.log("Hooking sigEncryptFunction at offset: " + sigEncryptFunctionPointer);
Interceptor.attach(sigEncryptFunctionPointer, {
    onEnter: function (args) {
        console.log("Function 0x87444 entered! Parameters: " + args[0] + ", " + args[1]);
        this.inputProtobuf = args[0];
        this.inputProtobufSize = args[1].toInt32();
        console.log(Memory.readByteArray(this.inputProtobuf, this.inputProtobufSize));
    },
    onLeave: function(retval) {
        console.log("Function 0x87444 returned!");
        // As an example, we saved some values into `this` which allows us to retrieve them on return and look at their values
        console.log(Memory.readByteArray(this.inputProtobuf, this.inputProtobufSize + (256 - (this.inputProtobufSize % 256))));
        // The change in the size will be clearer soon.
    }
})
```

1st day – 8pm

The sigEncryptFunction

```
2: 9588
4 {
  1: "network"
  2: 8078
  13: 0x41fe7a3e
  14: 0x420b2364
  21: 0x42700000
  26: 3
  28: 1
}
5 {
  2: "\010\n\017..."
  3: "\000\200\234..."
  4: "\000\000@@..."
  5: "\000\000\000..."
  6 {
    0: 0x0101010101010101
    0: 0x0100000100000001
    0: 0x0000000000000001
  }
  7 {
    0: 0x0101010101010101
    0: 0x0100000100000001
    0: 0x0000000000000001
  }
  8: "\000\000\001..."
}
7 {
  1: 6287
  3: 0xbf93780000000000
  4: 0xbf693b5000000000
  5: 0xbf67b40000000000
  6: 0xc04070ca00000000
  -- ...
  19: 3
}
8 {
  1: "b79aba9100a15631"
  2: "MSM8974"
  3: "unknown"
  4: "oneplus"
  -- ...
  14: "oneplus/bacon/A0001:6.0.1/MHC19Q/ZNH2KAS254:user/release-keys"
}
9: ""
10: 285106495
20: 521243270
22: "XD\003\t\2049\264\001\270\005E\214\3570\006\205"
23: 1470491093072
24: "\257\262\242..."
```

```
}
8 {
  1: "b79aba9100a15631"
  2: "MSM8974"
  3: "unknown"
  4: "oneplus"
  -- ...
  14: "oneplus/bacon/A0001:6.0.1/MHC19Q/ZNH2KAS254:use
}
```

1st day – 8pm

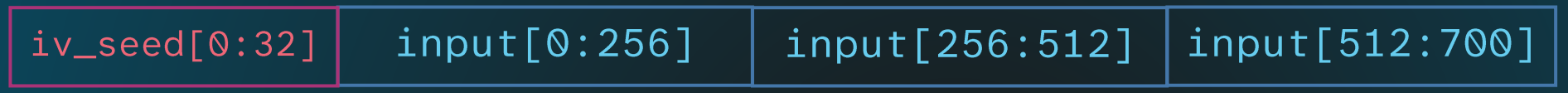
The sigEncryptFunction

input[0:256]	input[256:512]	input[512:700]

```
void sigEncryptFunction_87444(char *input, unsigned int size) {
  // ...
    srand48(time(NULL));
    for (int i = 0; i < 32; i++)
        iv_seed[i] = lrand48() & 0xff;

    memmove(input + 32, input, size);
    memcpy(input, iv_seed, 32);
```

1st day – 8pm

The sigEncryptFunction




```
void sigEncryptFunction_87444(char *input, unsigned int size) {
  // ...
    srand48(time(NULL));
    for (int i = 0; i < 32; i++)
        iv_seed[i] = lrand48() & 0xff;

    memmove(input + 32, input, size);
    memcpy(input, iv_seed, 32);
```

1st day – 8pm

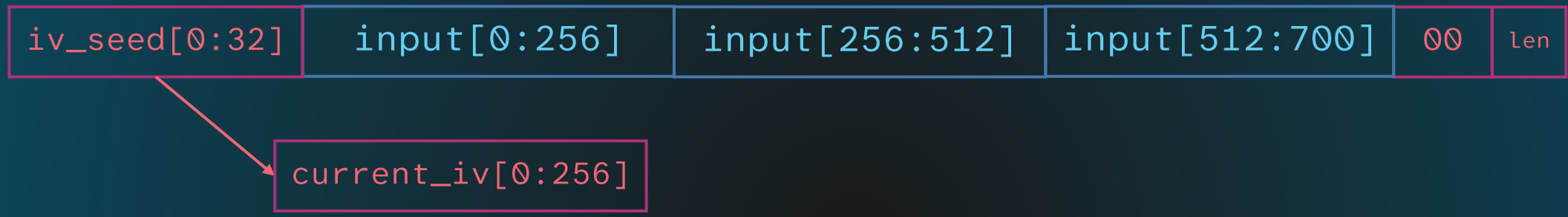
The sigEncryptFunction




```
size_remainder = 256 - size % 256;
final_size_without_header = size + size_remainder;
final_size = final_size_without_header + 32;

if (final_size_without_header > size)
{
    memset(input + 32 + size, 0, final_size_without_header - size - 1);
}
input[totalsize - 1] = size_remainder;
```

1st day – 8pm

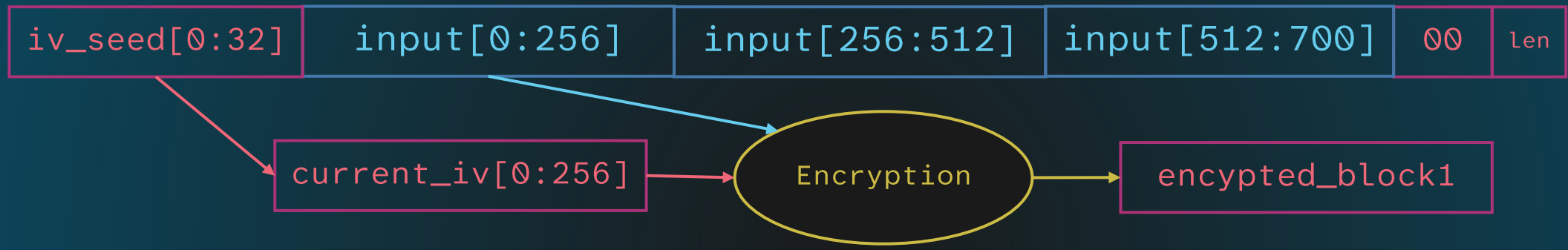
The sigEncryptFunction




```
for (int i = 0; i < 8; j++)
    for (int j = 0; j < 32; i++)
        current_iv[32*i + j] = ((iv_seed[i] >> ((-i) & 7)) | (iv_seed[i] << i))

for (int current_block = 32; current_block < final_size; current_block += 256)
{
    for (int i = 0; i < 256; i++)
        input[current_block + i] ^= current_iv[i];
    encryption_9e9d8(input + current_block, block_encryption_output);
    memcpy(current_iv, block_encryption_output, 256);
    memcpy(input + offset, block_encryption_output, 256);
}
```

1st day – 8pm

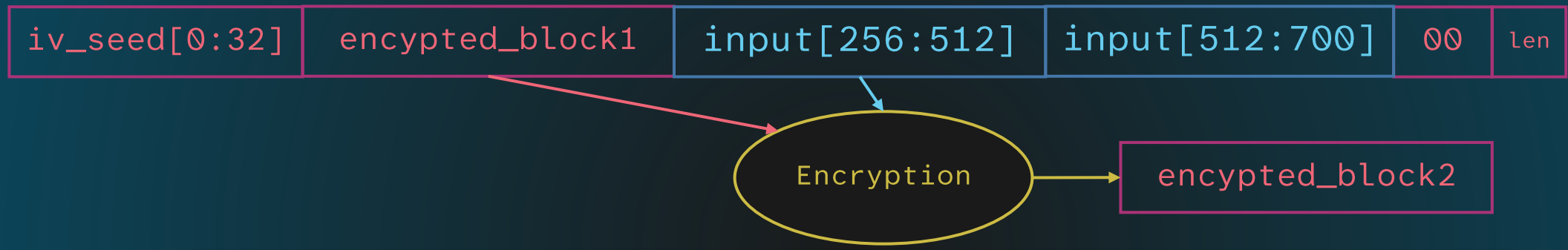
The sigEncryptFunction




```
for (int i = 0; i < 8; j++)
    for (int j = 0; j < 32; i++)
        current_iv[32*i + j] = ((iv_seed[i] >> ((-i) & 7)) | (iv_seed[i] << i))

for (int current_block = 32; current_block < final_size; current_block += 256)
{
    for (int i = 0; i < 256; i++)
        input[current_block + i] ^= current_iv[i];
    encryption_9e9d8(input + current_block, block_encryption_output);
    memcpy(current_iv, block_encryption_output, 256);
    memcpy(input + offset, block_encryption_output, 256);
}
```

1st day – 8pm

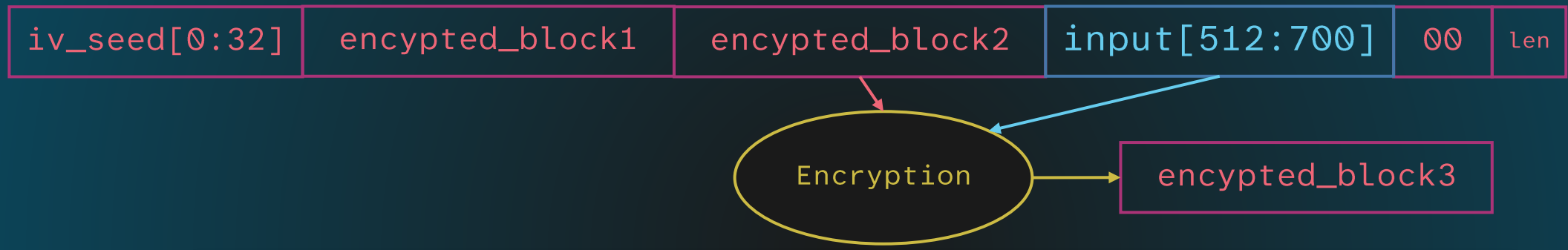
The sigEncryptFunction




```
for (int i = 0; i < 8; j++)
    for (int j = 0; j < 32; i++)
        current_iv[32*i + j] = ((iv_seed[i] >> ((-i) & 7)) | (iv_seed[i] << i))

for (int current_block = 32; current_block < final_size; current_block += 256)
{
    for (int i = 0; i < 256; i++)
        input[current_block + i] ^= current_iv[i];
    encryption_9e9d8(input + current_block, block_encryption_output);
    memcpy(current_iv, block_encryption_output, 256);
    memcpy(input + offset, block_encryption_output, 256);
}
```

1st day – 8pm

The sigEncryptFunction




```
for (int i = 0; i < 8; j++)
    for (int j = 0; j < 32; i++)
        current_iv[32*i + j] = ((iv_seed[i] >> ((-i) & 7)) | (iv_seed[i] << i))

for (int current_block = 32; current_block < final_size; current_block += 256)
{
    for (int i = 0; i < 256; i++)
        input[current_block + i] ^= current_iv[i];
    encryption_9e9d8(input + current_block, block_encryption_output);
    memcpy(current_iv, block_encryption_output, 256);
    memcpy(input + offset, block_encryption_output, 256);
}
```

1st day – 8pm

The sigEncryptFunction

iv_seed[0:32]	encypted_block1	encypted_block2	encypted_block3

```
for (int i = 0; i < 8; j++)
    for (int j = 0; j < 32; i++)
        current_iv[32*i + j] = ((iv_seed[i] >> ((-i) & 7)) | (iv_seed[i] << i))

for (int current_block = 32; current_block < final_size; current_block += 256)
{
    for (int i = 0; i < 256; i++)
        input[current_block + i] ^= current_iv[i];
    encryption_9e9d8(input + current_block, block_encryption_output);
    memcpy(current_iv, block_encryption_output, 256);
    memcpy(input + offset, block_encryption_output, 256);
}
```

2nd day – 1am

Making Encryption Work

```
14171  int encrypt(const unsigned char *input, size_t input_size,
14172      const unsigned char* iv, size_t iv_size,
14173      unsigned char* output, size_t * output_size) {
```



2nd day – 9am

Decoding the Signature

```
message LocationFix {
    string proider = 1; -- "network", "gps", "fused", ...
    uint64 timestamp_snapshot  = 2; -- In ms since start
    float altitude = 4; -- Found in iOS
    float latitude = 13;
    float longitude = 14;
    float speed = 18; -- In m/s, only in iOS. -1 for no value
    float course = 20; -- Only in iOS. -1 for no value, valid range is 0 to 360
    float horizontal_accuracy = 21;
    float vertical_accuracy = 22; -- Only in iOS
    uint64 provider_status = 26; -- 1 - no fix, 2 - acquiring/inaccurate, 3 fix acquired
    uint64 always_1 = 28
}
```

```
message DeviceInfo {
    string device_id = 1; -- As hex string
    string android_board_name = 2;
    string android_bootloader = 3;
    string device_brand = 4; -- product.brand prop
    string device_model = 5; -- product.device prop
    string device_model_identifier = 6; -- build.display.id prop. Only in Android
    string device_model_boot = 7; -- boot.hardware prop
    string hardware_manufacturer = 8; -- product.manufacturer prop
    string hardware_model = 9; -- product.model prop
    string firmware_brand = 10; -- product.name prop. In iOS - "iPhone OS"
    string firmware_tags = 12; -- build.tags prop. Only in Android
    string firmware_type = 13; -- build.type prop. In iOS - version
    string firmware_fingerprint = 14; -- build.fingerprint prop. Only in Android
}
```

2nd day – 2pm

Decoding the Signature

```
message SensorInfo {
    uint64 timestamp_snapshot = 1;  -- in ms since start
    double linear_acceleration_x = 3;
    double linear_acceleration_y = 4;
    double linear_acceleration_z = 5;
    double magnetic_field_x = 6;
    double magnetic_field_y = 7;
    double magnetic_field_z = 8;
    int32 magnetic_field_accuracy = 9;
    double attitude_pitch = 10;
    double attitude_yaw = 11;
    double attitude_roll = 12;
    double rotation_rate_x = 13;
    double rotation_rate_y = 14;
    double rotation_rate_z = 15;
    double gravity_x = 16;
    double gravity_y = 17;
    double gravity_z = 18;
    int32 status = 19; -- Set to be 3 when sensors data exists
}
```

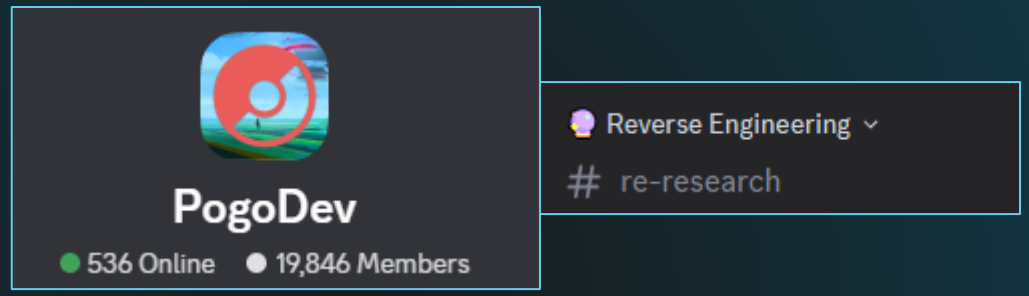
2nd day – 6pm

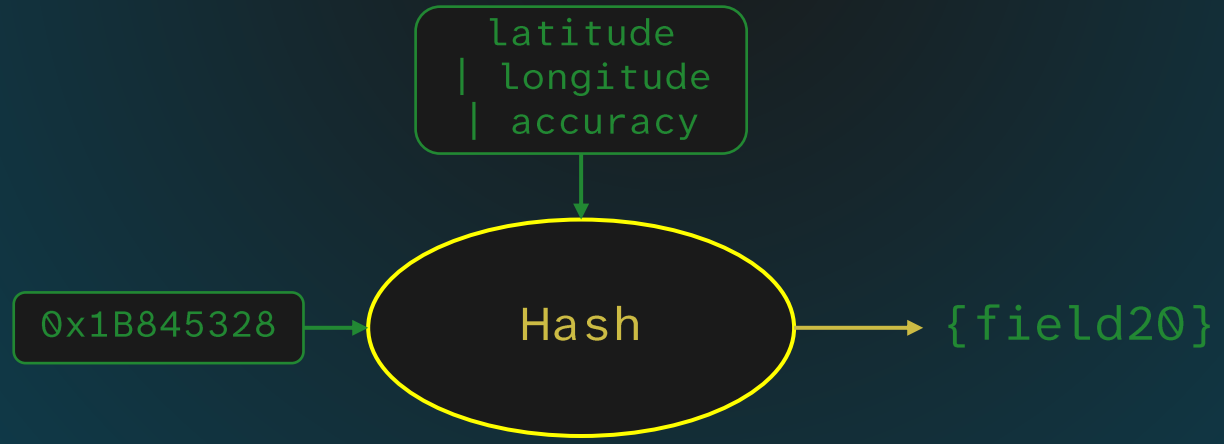
Decoding the Signature

```
message Signature {
    repeated UnknownMessage field1 = 1; -- Not required
    uint64 timestamp_since_app_start = 2;
    string field3 = 3; -- Not required
    repeated LocationFix location_fix = 4; -- Not required
    repeated UnknownMessage field5 = 5; -- Android only - not required
    repeated UnknownMessage field6 = 6; -- Not required
    repeated SensorInfo sensor_info = 7; -- Not required
    DeviceInfo device_info = 8; -- Not required
    UnknownMessage field9 = 9; -- iOS only - not required
    int32 field10 = 10;
    bool field11 = 11; -- Not required
    bool field12 = 12; -- Not required
    int32 field13 = 13; -- Not required
    int32 field14 = 14; -- Not required
    string field15= 15; -- Not required
    int32 field16 = 16; -- Not required
    string field17 = 17; -- Not required
    string field18 = 18; -- Not required
    bool field19 = 19; -- Not required
    int32 field20 = 20;
    bool field21 = 21; -- Not required
    bytes field22 = 22;
    uint64 timestamp = 23;
    repeated uint64 field24 = 24;
    int64 field25 = 25; -- Not required
    int32 field27 = 27; -- Not required
}
```

2nd day – 8pm

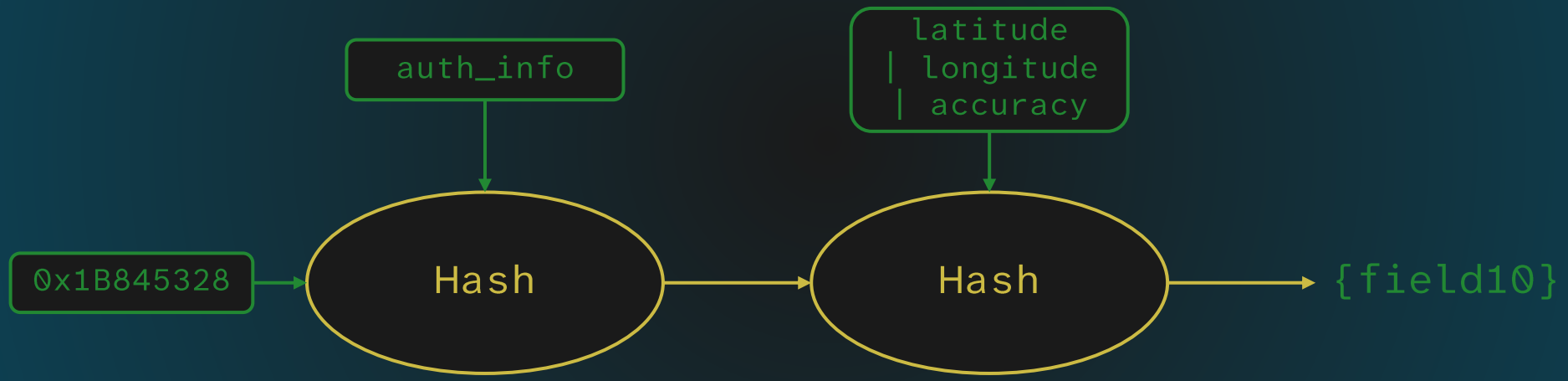
Identifying Missing Fields





2nd day – 11pm

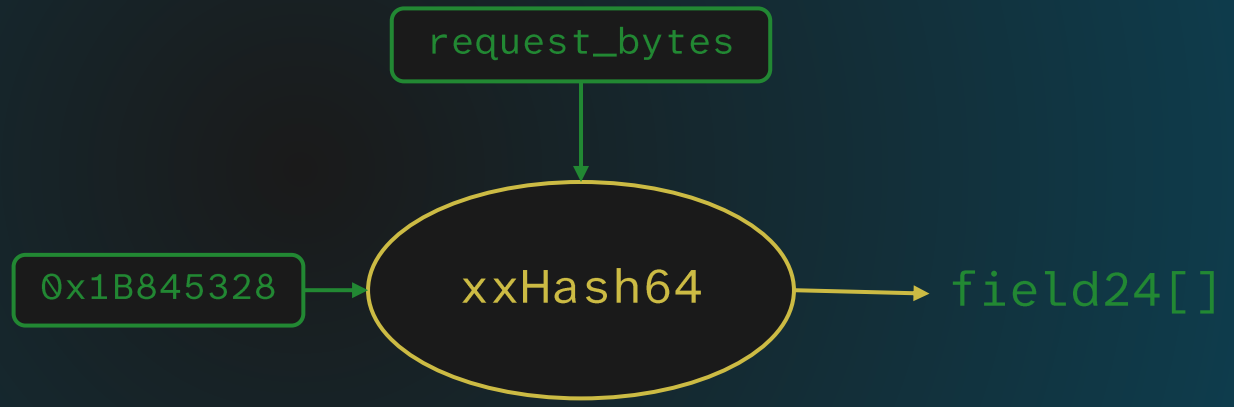
Identifying Missing Fields



3rd day – 4am

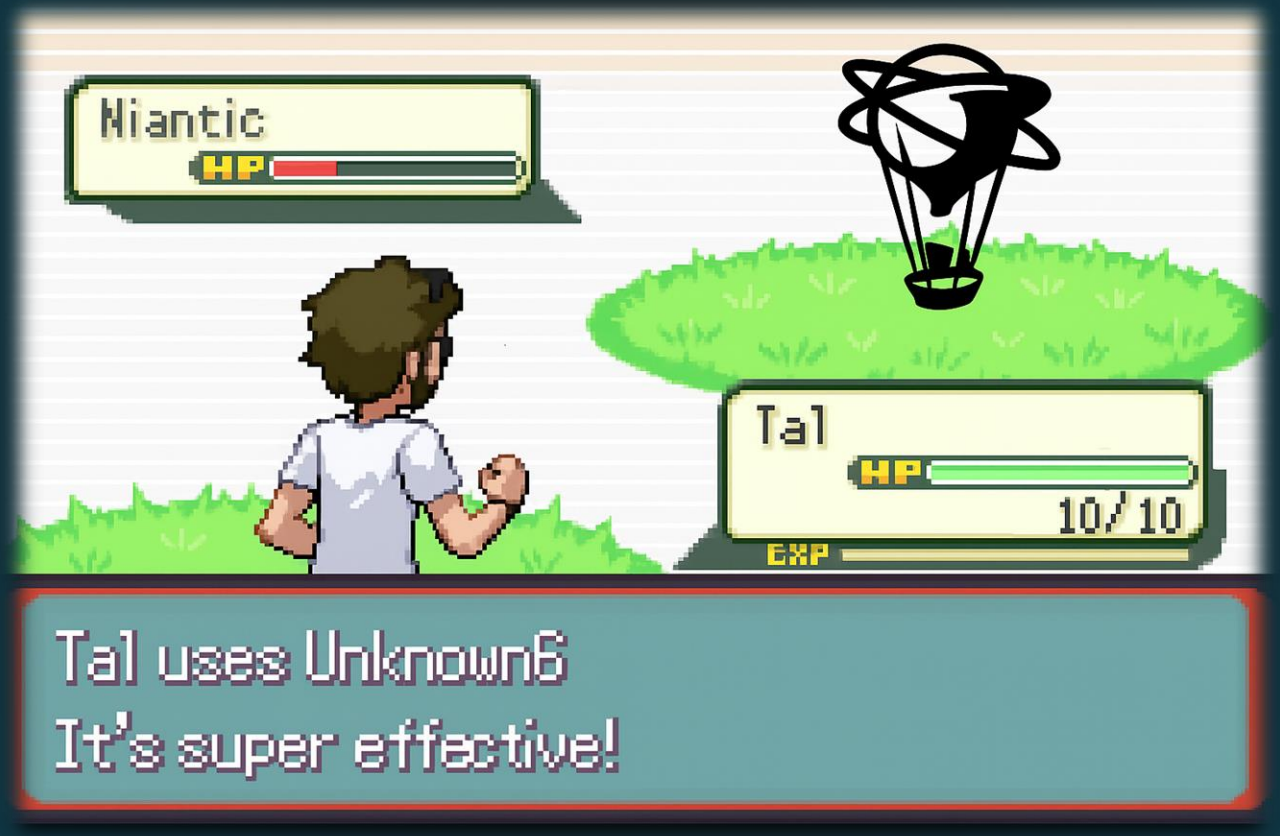
Who's This Hash?

```
status_code: 2
request_id: 1764153216922026362
requests: {
  request_type: 106
  request_bytes: -- ...
}
requests: {
  request_type: 126
}
requests: {
  request_type: 4
  request_bytes: --- ...
}
```



3rd day – 7pm

The Final Field



Finale & Considerations

```
def _create_encrypted_signature(self, latitude, longitude, accuracy, data):
    signature = Signature_pb2.Signature()

    auth_ticket_hash = xxhash.xxh32(self.request.auth_ticket.SerializeToString(), seed=0x1B845238).intdigest()

    if not accuracy:
      accuracy = "\x00" * 8
    signature.field10 = self.hash_location(auth_ticket_hash, d2h(latitude) + d2h(longitude) + d2h(accuracy))
    signature.field20 = self.hash_location(0x1B845238, d2h(latitude) + d2h(longitude) + d2h(accuracy))
    signature.field24.append(self.hash_auth(self.request.auth_ticket.SerializeToString(), data))

    signature.field22 = hashlib.md5("yeah").digest()
    signature.timestamp_snapshot = 60000
    signature.timestamp = int(time.time() * 1000)
    return self._encrypt_with_lib(signature.SerializeToString())

def _hash_location(self, seed, data):
    return xxhash.xxh32(data, seed=seed).intdigest()

def _hash_auth(self, auth_ticket, data):
    auth_ticket_hash = xxhash.xxh64(auth_ticket, seed=0x1B845238).intdigest()
    return xxhash.xxh64(data, seed=auth_ticket_hash).intdigest()

def _encrypt_with_lib(self, data):
    encrypt_dll = ctypes.cdll.LoadLibrary("encrypt.dll")
    encrypt_dll.argtypes = [ctypes.c_char_p, ctypes.c_size_t, ctypes.c_char_p, ctypes.c_size_t, ctypes.POINTER(ctypes.c_ubyte), ctypes.POINTER(ctypes.c_size_t)]
    encrypt_dll.restype  = ctypes.c_int
    output_size = ctypes.c_size_t()
    output_size.value = len(data) + 32 + (256 - len(data) % 256)
    output = (ctypes.c_ubyte * output_size.value)()

    ret = encrypt_dll.encrypt(data, len(data), "\x24" * 32, 32, ctypes.byref(output), ctypes.byref(output_size))
    return "".join([chr(x) for x in output])
```

Finale & Considerations

Team Unknown6 license GPL-3.0

From ***4th August 2016 - 00:49 GMT +1*** until ***7th August 2016 - 01:05 GMT +1***, amazing things happened. A community of Pokémon GO players from all around the world came together with the intention of restoring service to the third-party PoGO API.

We were collectively known as **Team Unknown6**, due to the first unknown we tried to decrypt. The events which followed during those three and a half days was nothing short of special, and unforgettable.

It was a lengthy battle of teamwork and effort to decrypt the U6 generation algorithm. The collaboration begun via the PokémonGoDev sub-reddit and eventually spilled onto the Discord chat servers. Seamless collaboration begun between highly experienced engineers, which lead to multiple wiki pages and Github repositories created to track and pool together progress.

```
207             sig = Signature()
208
209             sig.session_hash = self.session_hash
210             sig.timestamp = get_time(ms=True)
211             sig.timestamp_since_start = get_time(ms=True) - RpcApi.START_TIME
212             if sig.timestamp_since_start < 5000:
213                 sig.timestamp_since_start = random.randint(5000, 8000)
214
215 ...         self._hash_engine.hash(sig.timestamp, request.latitude, request.longitude, request.accuracy, ticket_serialized, sig.session_hash, request.requests)
216             sig.location_hash1 = self._hash_engine.get_location_auth_hash()
217             sig.location_hash2 = self._hash_engine.get_location_hash()
```

Finale & Considerations

References

- Team Unknown6 repository
- Review of the hackathon
- Arstechnia article on the hackathon
- Reconstructed encryption repository
- Hackathon updates Reddit Thread
- PokemonGoXposed repository
- Pgoapi repository
- PogoProtos repository