

近年のGame Security診断 について

nevermoe

—OWASP NIGHT MEETING

自己紹介

◆ nevermoe

- 趣味：水泳、スキー、引きこもり
- 履歴：



26年前：中国で生まれ

3年前：渡日 東京大学大学院（CG研究）

八ヶ月前：XXXX株式会社セキュリティ室に新卒入社

- 仕事：Web、Mobile App、Game等のセキュリティ診断

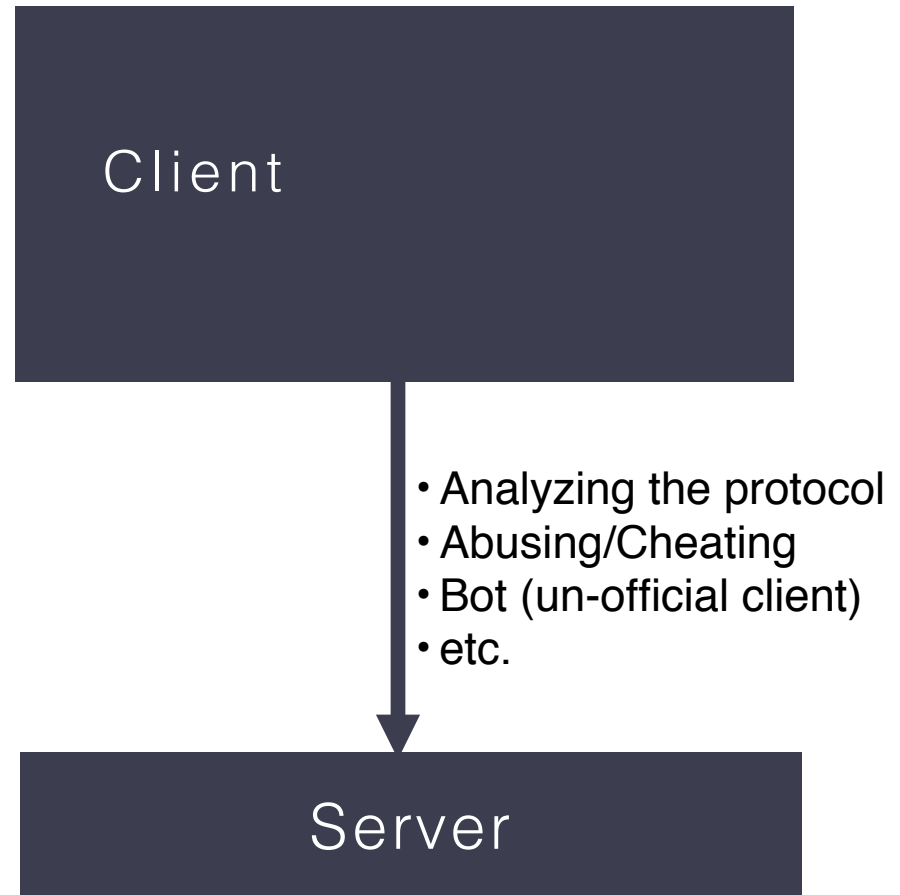
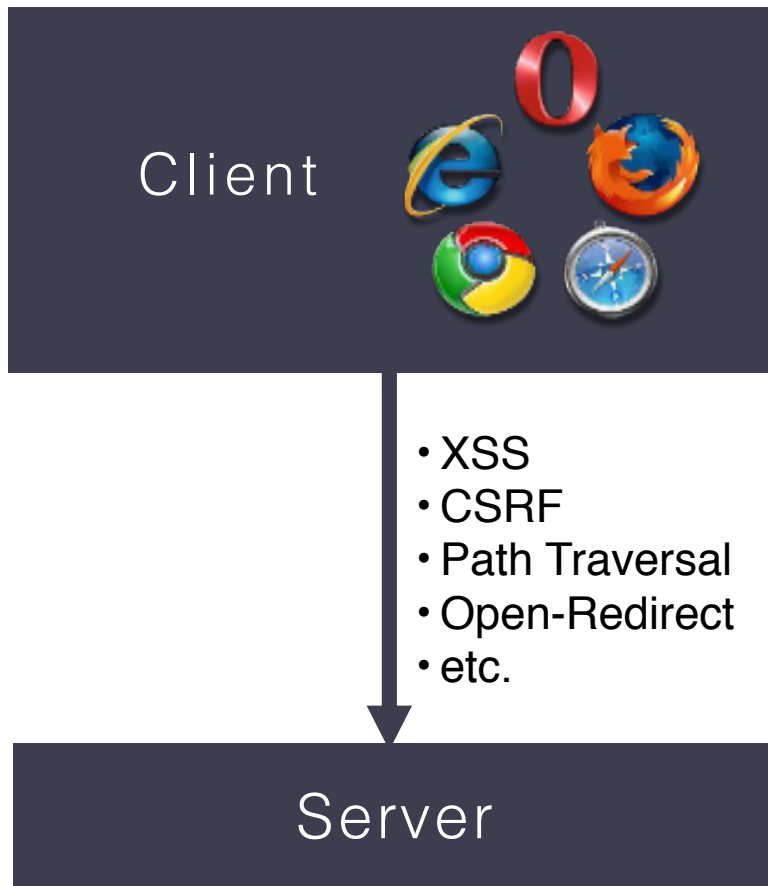
目次

- セキュリティ診断概要
- バイナリー解析
- 通信解析
- 対策&まとめ

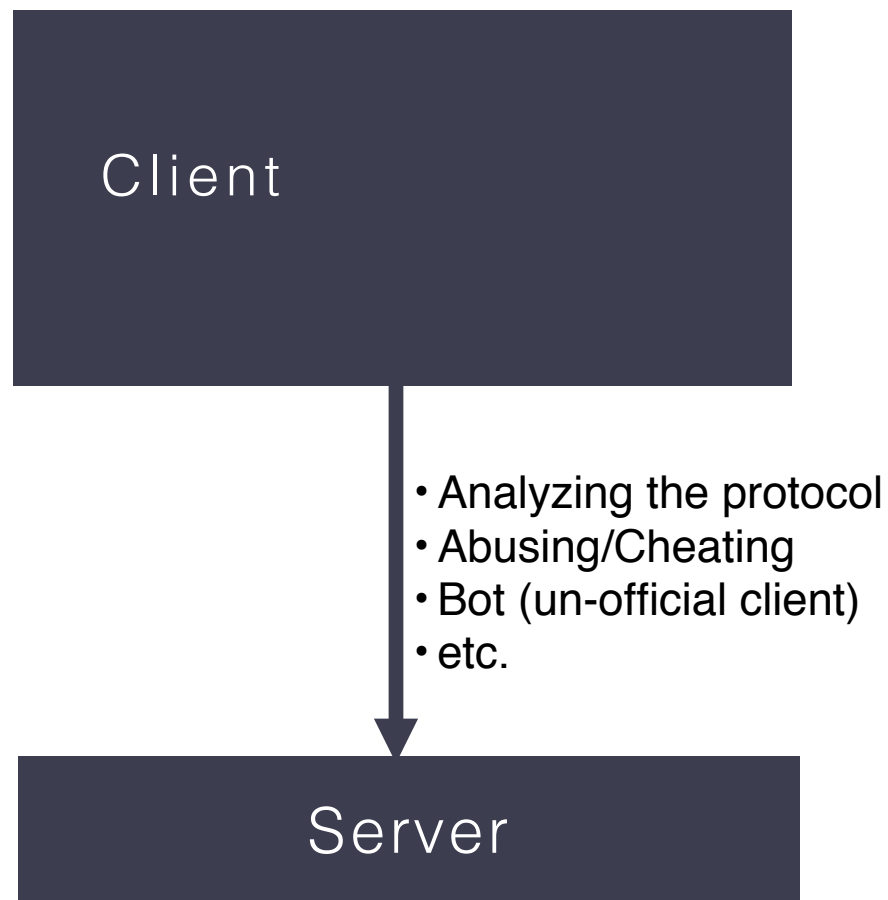
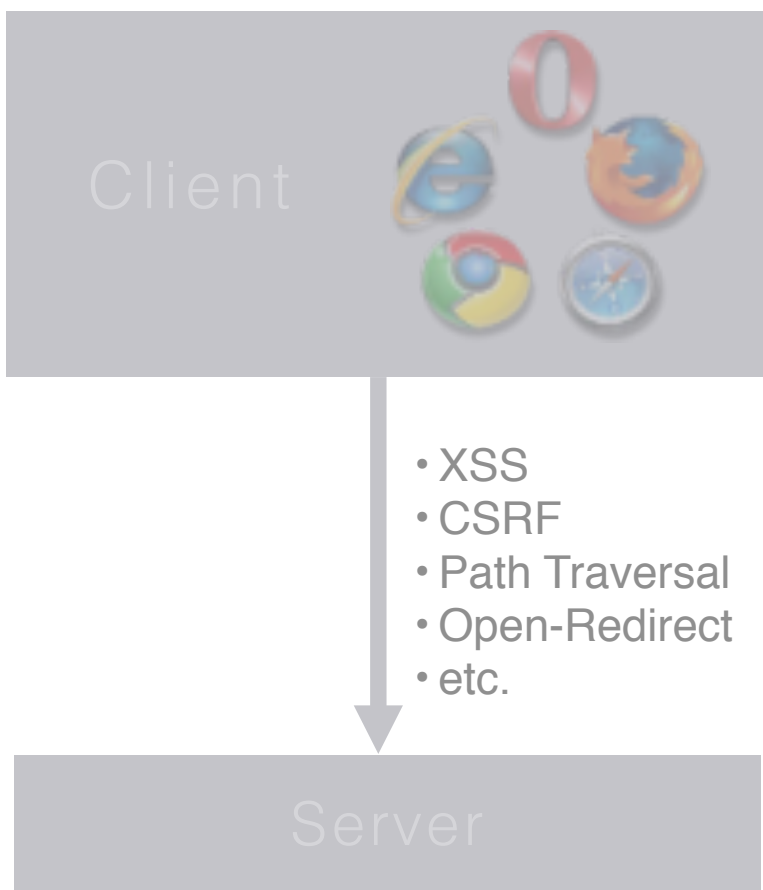
目次

- セキュリティ診断概要
- バイナリー解析
- 通信解析
- 対策&まとめ

セキュリティ診断概要



セキュリティ診断概要



目次

- セキュリティ診断概要
- バイナリー解析
- 通信解析
- 対策&まとめ

ゲームエンジン

◆ Unity3D unity

- 2Dでも、3Dでも
- Cross Platform
- sourceの一部公開

◆ Cocos2d-x

- 2Dゲームに特化
- Cross Platform
- Open Source

◆ FlashAIR/Unreal/Corona等

◆ 自作

圧倒的にシェアが高い

Unity3dゲーム現状

◆Android -> Mono

- パッキング
- 難読化



=> 難読かされた
コードをリバーシング？

◆iOS -> IL2CPP

- 取得は簡単
- パッキング、難読化一切なし



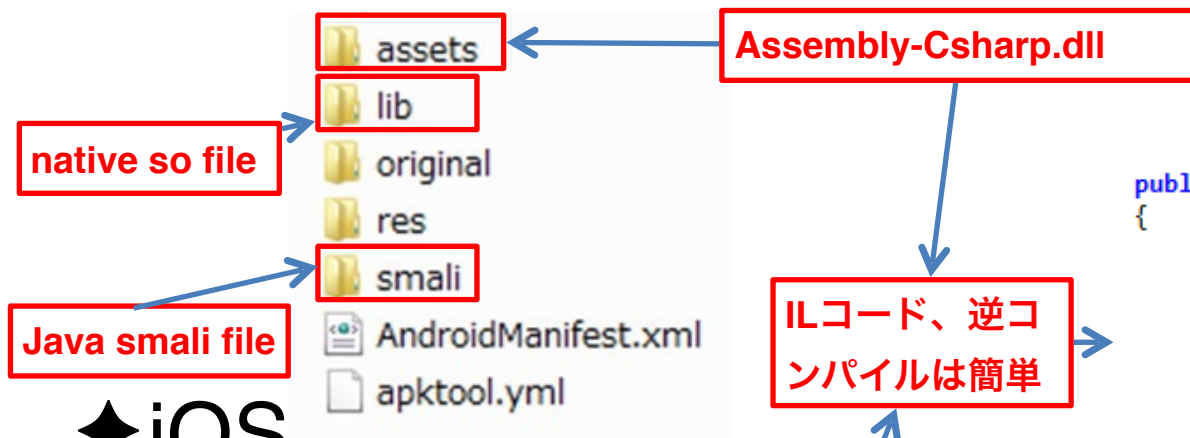
=> シンボルのない
アセンブラを
リバーシング？

★ただし、最近Androidも

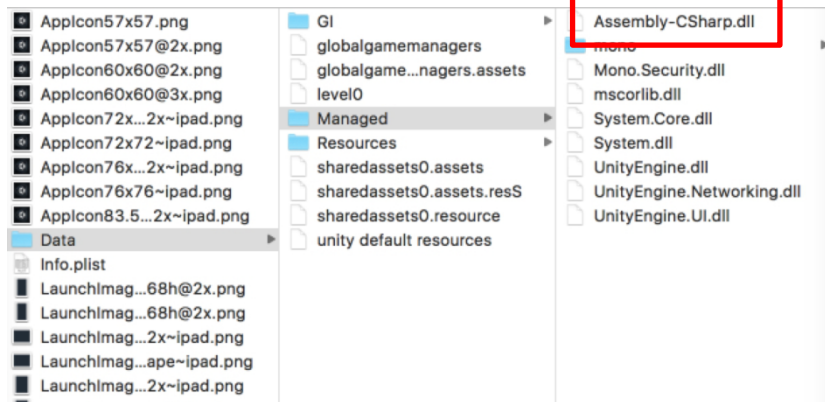
IL2CPPサポートするようになった

IL2CPP紹介 (without IL2CPP)

◆ Android



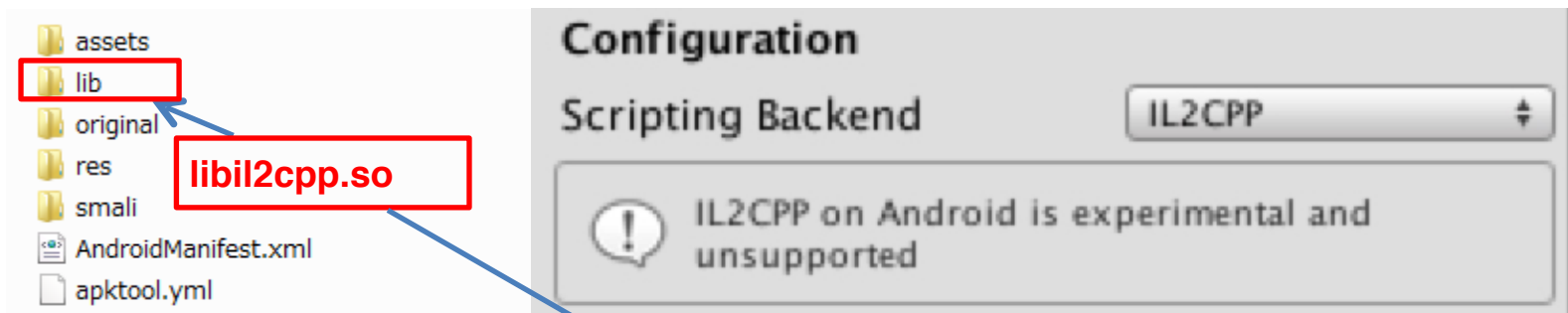
◆ iOS



```
public void TakeDamage(int amount)
{
    this.damaged = true;
    this.currentHealth -= amount;
    this.healthSlider.set_value((float)this.currentHealth);
    this.playerAudio.Play();
    if (this.currentHealth <= 0 && !this.isDead)
    {
        this.Death();
    }
}
```

IL2CPP紹介 (with IL2CPP)

◆ Android

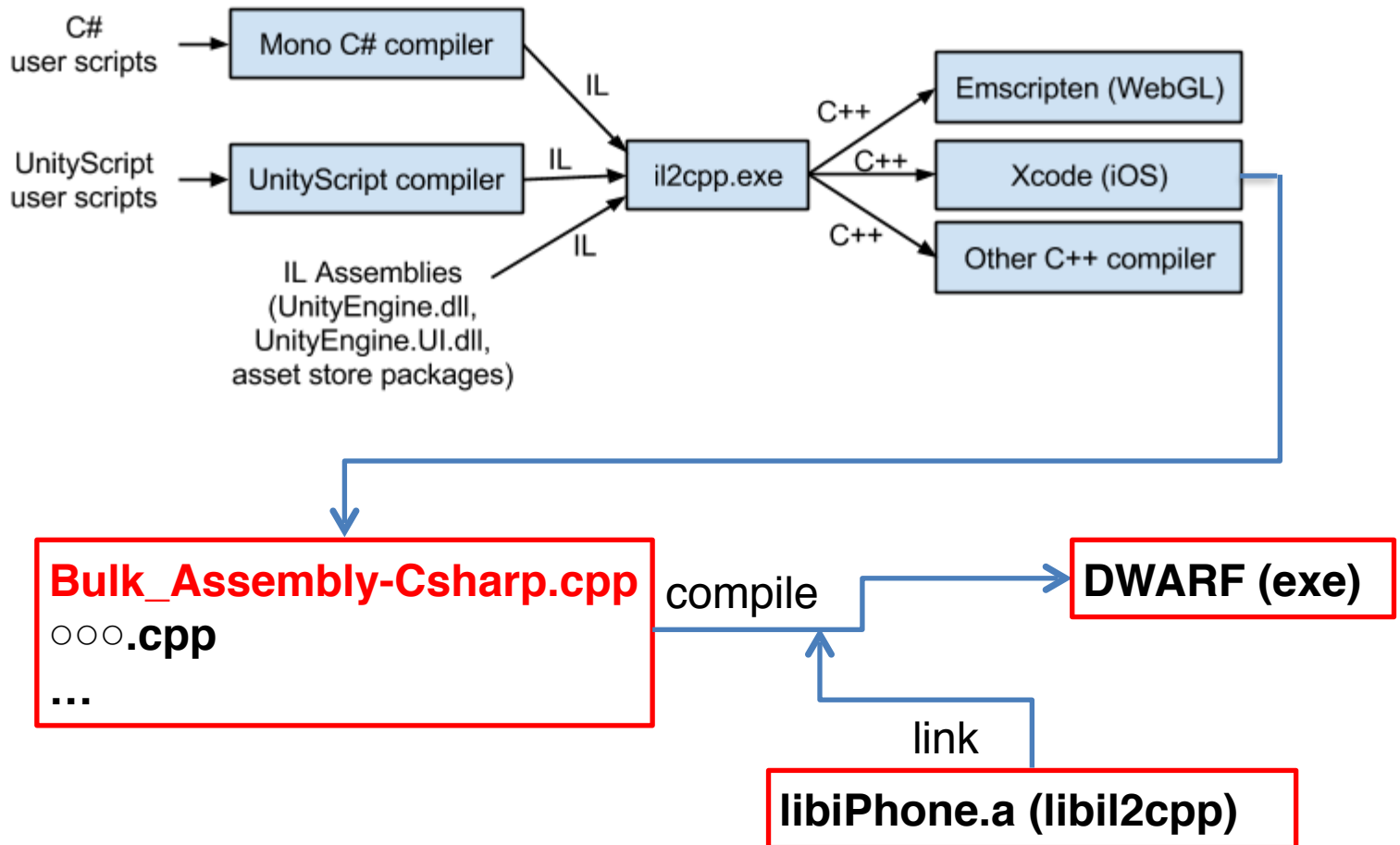


◆ iOS



```
LDR      X8, [SP,#arg_10]
STR      X8, [SP,#arg_A8]
ADRP    X8, #dword_103753684@PAGE
NOP
LDR      W9, [SP,#arg_1C]
STR      W9, [SP,#arg_B0]
LDR      W8, [X8,#dword_103753684@PAGEOFF]
MOV      W9, #1
MADD    W8, W8, W8, W9
MOV      W9, #0x24920000
MOVK    W9, #0x4925
MADD    X9, X8, X9, XZR
UBFM    X9, X9, #0x20, #0x3F
SUB     W10, W8, W9
ADD     W9, W9, W10,LSR#1
UBFM    W10, W9, #2, #0x1F
UBFM    W10, W10, #0x1D, #0x1C
SUB     W9, W10, W9,LSR#2
CMP     W8, W9
MOV     W8, #0x16
MOV     W9, #0x19
```

IL2CPP 紹介



Metadata

Bulk_Assembly-Csharp.cpp

```
extern "C" void EnemyHealth_Death_m1976714863 (EnemyHealth_t3285426352 * __this,  
const MethodInfo* method)  
{  
    static bool s_Il2CppMethodIntialized;  
    if (!s_Il2CppMethodIntialized) { ... }  
    {  
        __this->set_isDead_1l((bool)1);  
        CapsuleCollider_t720607407 * L_0 = __this->get_capsuleCollider_10();  
        NullCheck(L_0);  
        Collider_set_isTrigger_m1298573031(L_0, (bool)1, /*hidden argument*/NULL);  
        Animator_t69676727 * L_1 = __this->get_anim_7();  
        NullCheck(L_1);  
        Animator_SetTrigger_m3418492570(L_1, _stringLiteral4136222740  
        /*hidden argument*/NULL);
```

tDistance.m_EffectColor.
m_EffectDistance.m_UseGr
aphicAlpha.effectColor.e
ffectDistance.useGraphic
Alpha.Assembly-CSharp.As
sembly-CSharp.dll.EnemyA
ttack.OnTriggerEnter.OnT
riggerExit.Attack.timeBe
tweenAttacks.attackDamag
e.anim.playerHealth play
erInRange.EnemyHealth am
ount.hitPoint.TakeDamage
.Death.StartSinking.star

ordering.Internal error.
Trying to destroy objec
t that is already releas
ed to pool.Mesh can not
have more than 65000 ver
ticesPlayerPlayerDead/he
reisthelog?DeadSpawnGame
OverScore: DieShootableF
ire1_name This is not po
ssible to be called for
standalone input. Please

Data/Managed/Metadata/global-metadata.dat

シンボル復号

```
DCQ qword_1012D67A8
DCQ qword_1012D67B0
DCQ qword_1012D67B8
DCQ qword_1012D67C0
DCQ qword_1012D67C8
DCQ qword_1012D67D0
DCQ qword_1012D67D8
DCQ sub_1003212D8
DCQ sub_1003212FC
DCQ sub_1003213D4
DCQ sub_100321444
DCQ sub_10032155C
DCQ sub_1003215CC
```



```
DCQ StringLiteral_Passed_argument__args__is_invalid_size_
DCQ StringLiteral_Random_Insertion_is_semantically_invali
DCQ StringLiteral_Coroutine_container_not_configured___d
DCQ StringLiteral_Internal_error__Trying_to_destroy_objec
DCQ Locale$$GetText ; DATA XREF: __const:000000010118
DCQ Locale$$GetText_93228 ; Locale$$GetText
DCQ SafeHandleZeroOrMinusOneIsInvalid$$_ctor
DCQ SafeHandleZeroOrMinusOneIsInvalid$$get_IsInvalid
DCQ SafeWaitHandle$$_ctor
DCQ SafeWaitHandle$$ReleaseHandle
DCQ CodePointIndexer$$_ctor
```

IDA Plugin: https://github.com/nevermoe/unity_metadata_loader

Slide: https://www.nevermoe.com/wp-content/uploads/2016/11/avtokyo_jp_re.pdf

目次

- セキュリティ診断概要
- バイナリー解析
- 通信解析
- 対策&まとめ

通信改ざん対策

◆SSL Pinning △

- たくたく実装されない可能性がある
- 汎用性ツールによるbypass可能？

◆電文、パケットのHashチェック △

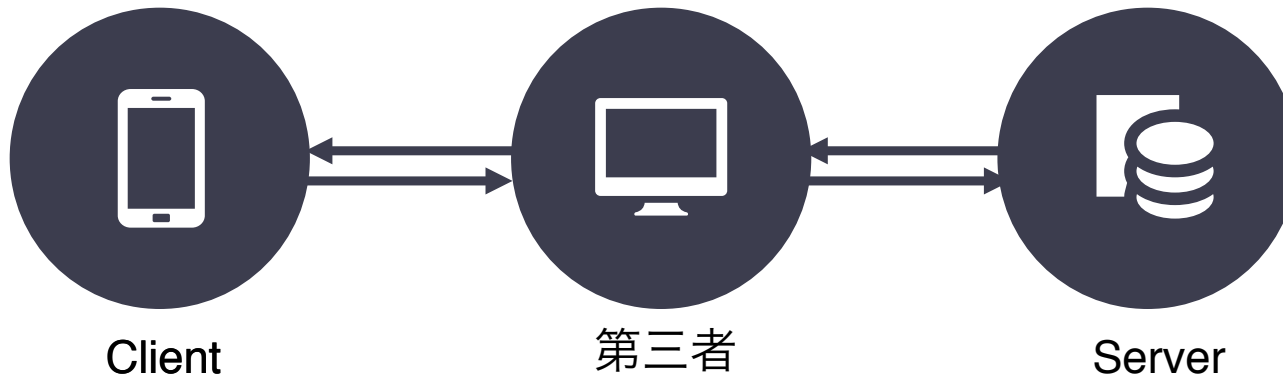
- integrityが保証されるが、通信内容が見える（protocol分析される）

◆通信暗号化 ○

- バイナリー解析しないとリバーシング難しい
- 汎用性解析ツールがない
- 現状一番使われている

SSL Pinning現状

- SSL Pinning Bypass



	Normal App (Android)	Normal App (iOS)	Game (Unity3d)	Game (Cocos2d-x)
SSL Pinning	○	○	✘	△
Bypass Tool	JustTrustMe	ssl-kill-switch	✘	✘

通信プロトコル

◆ 通常のApp

- JSON
- XML
- Protobuf: <https://developers.google.com/protocol-buffers/>
- Thrift: <https://thrift.apache.org/>
- MessagePack: <http://msgpack.org>

◆ ゲームにとって

- JSON
- MessagePack
- Protobuf

通信プロトコル比較

◆ JSON

- 人間の目で読める
- パースが簡単
- overheadが大きい

◆ MessagePack

- JSON-like
- 人間の目で読めない (バイナリー)
- パースが簡単

◆ Protobuf

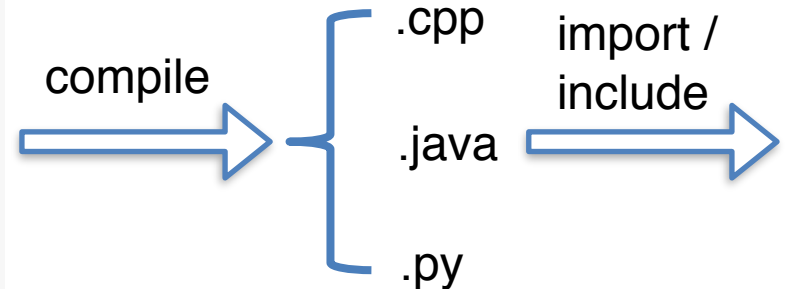
- Dense data
- 人間の目で読めない (バイナリー)
- schema (.proto)ファイルがないとdecodeが難しい
- 処理速度が速い
- 暗号化手法ではない！

Protobuf解析

◆ Protobuf

// .proto

```
message Person {  
  required string name = 1;  
  required int32 id = 2;  
  optional string email = 3;  
}
```



// .cpp

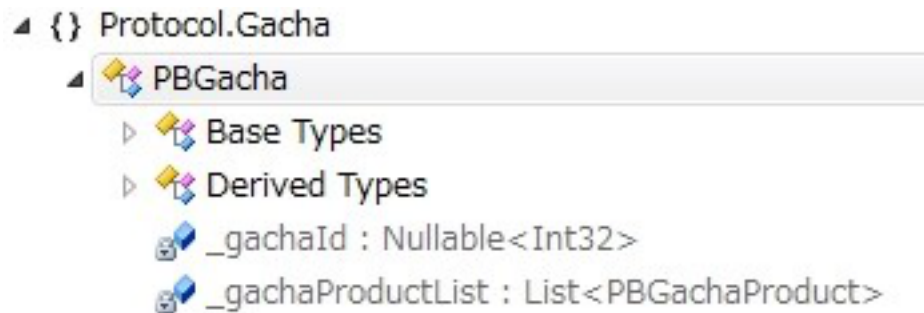
```
Person john;  
fstream input(argv[1],  
  ios::in | ios::binary);  
john.ParseFromIstream(&input);  
id = john.id();  
name = john.name();  
email = john.email();
```

// .java

```
Person john = Person.newBuilder()  
  .setId(1234)  
  .setName("John Doe")  
  .setEmail("jdoe@example.com")  
  .build();  
output = new FileOutputStream(args[0]);  
john.writeTo(output);
```

Protobuf解析

- ◆ .protoファイルあればもちろん復号できる
- ◆ java、c#のILをdecompileすればも復号可能

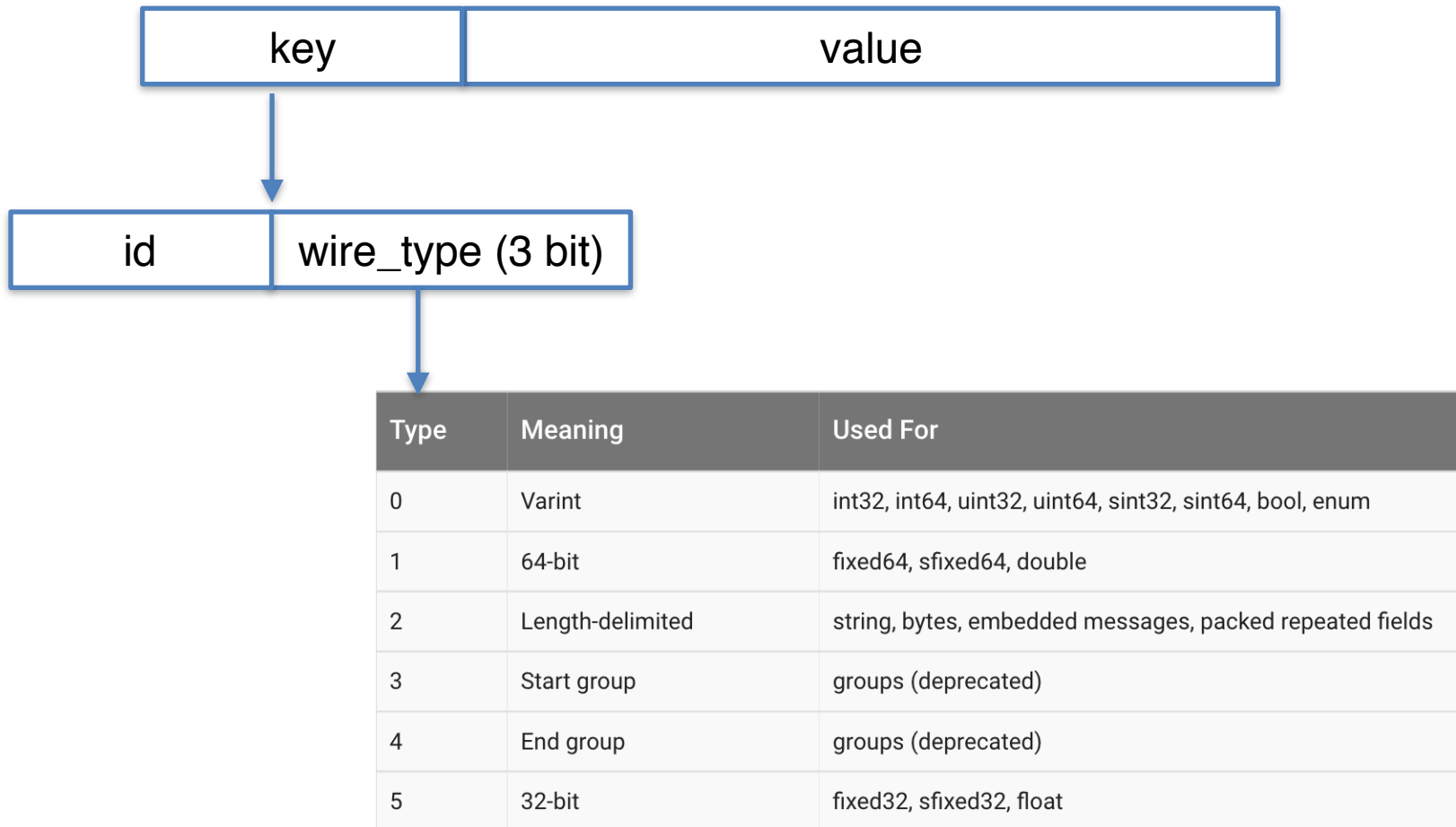


★ IL2CPPを使用すると？

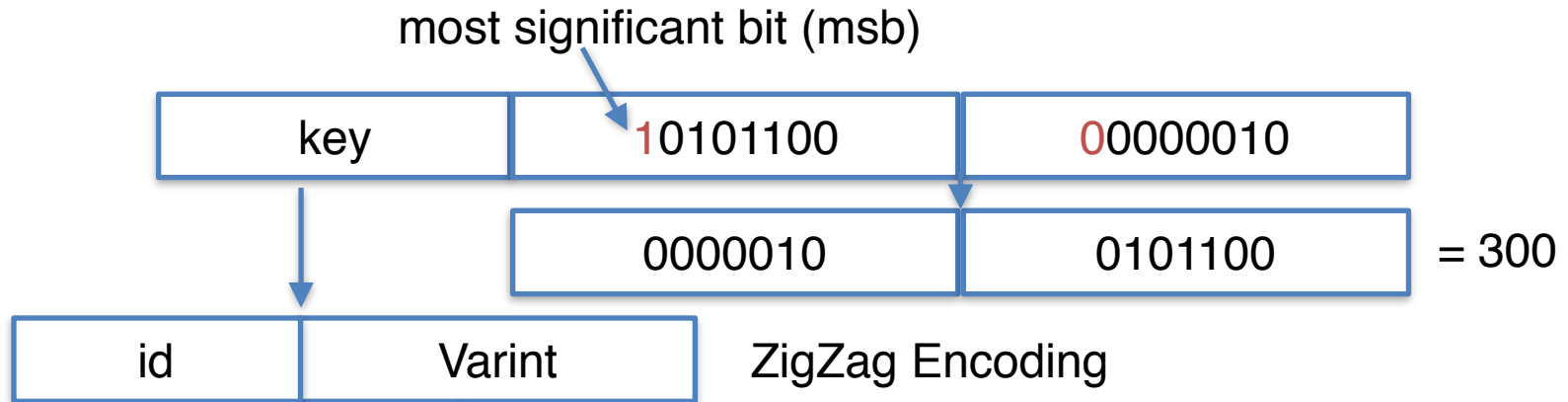


- Protobuf Encoding (<https://developers.google.com/protocol-buffers/docs/encoding>)

Protobuf解析

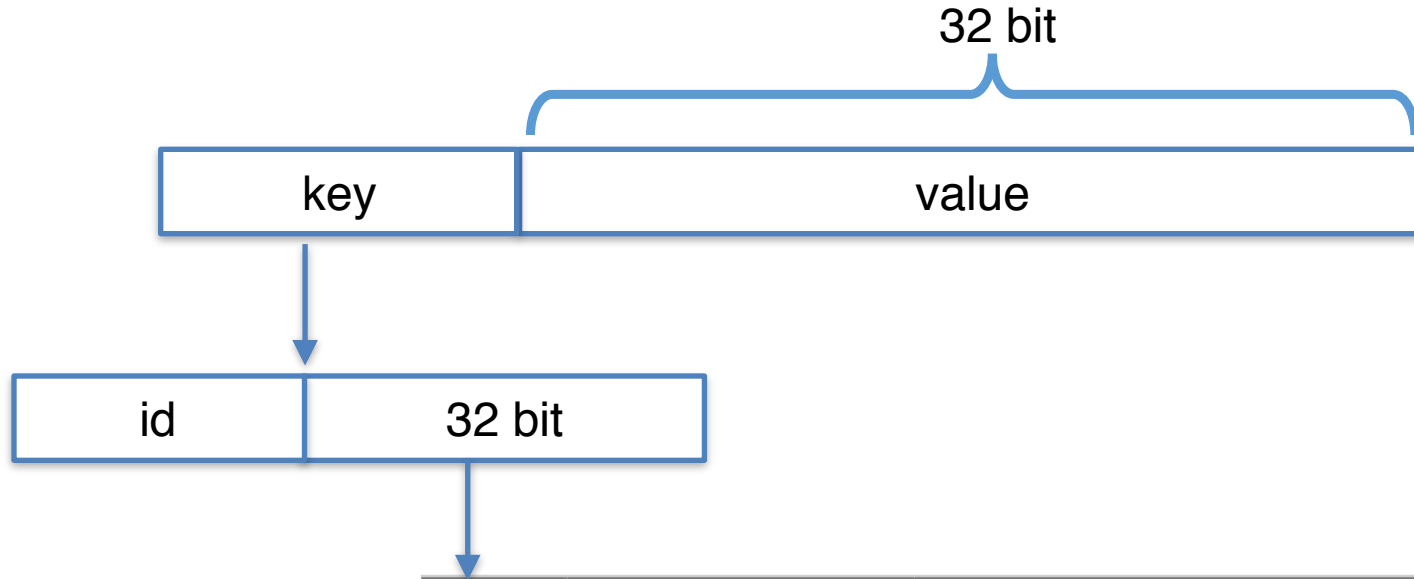


Protobuf解析



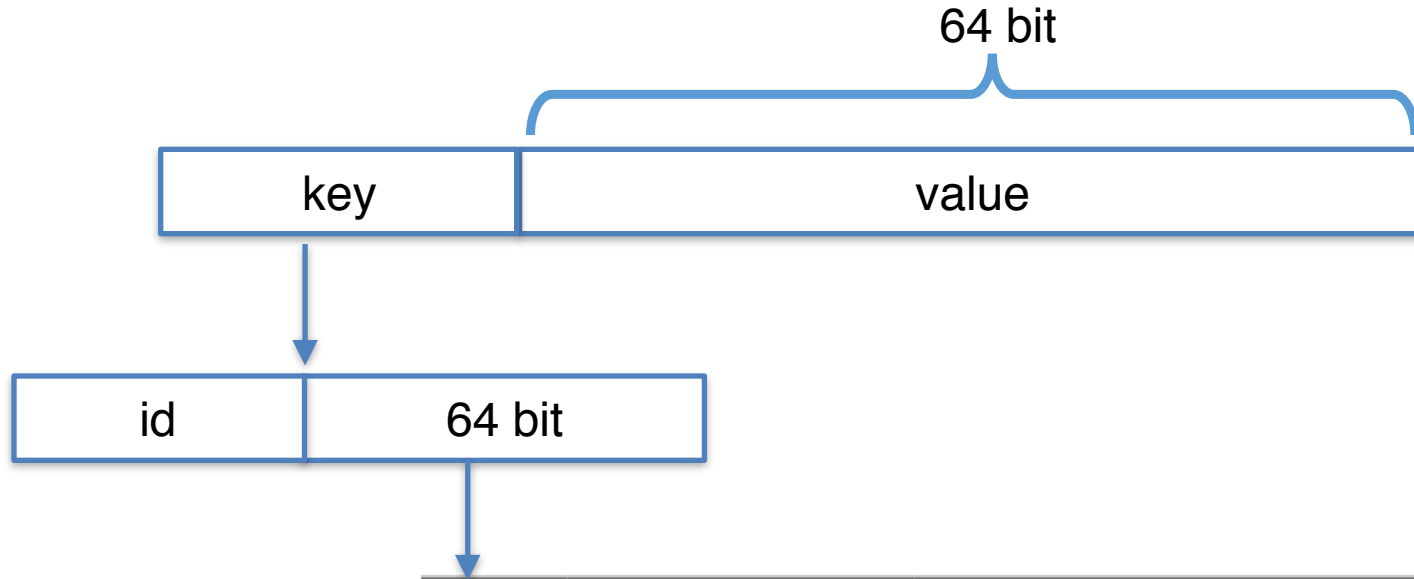
Type	Meaning	Used For
0	Varint	int32, int64, uint32, uint64, sint32, sint64, bool, enum
1	64-bit	fixed64, sfixed64, double
2	Length-delimited	string, bytes, embedded messages, packed repeated fields
3	Start group	groups (deprecated)
4	End group	groups (deprecated)
5	32-bit	fixed32, sfixed32, float

Protobuf解析



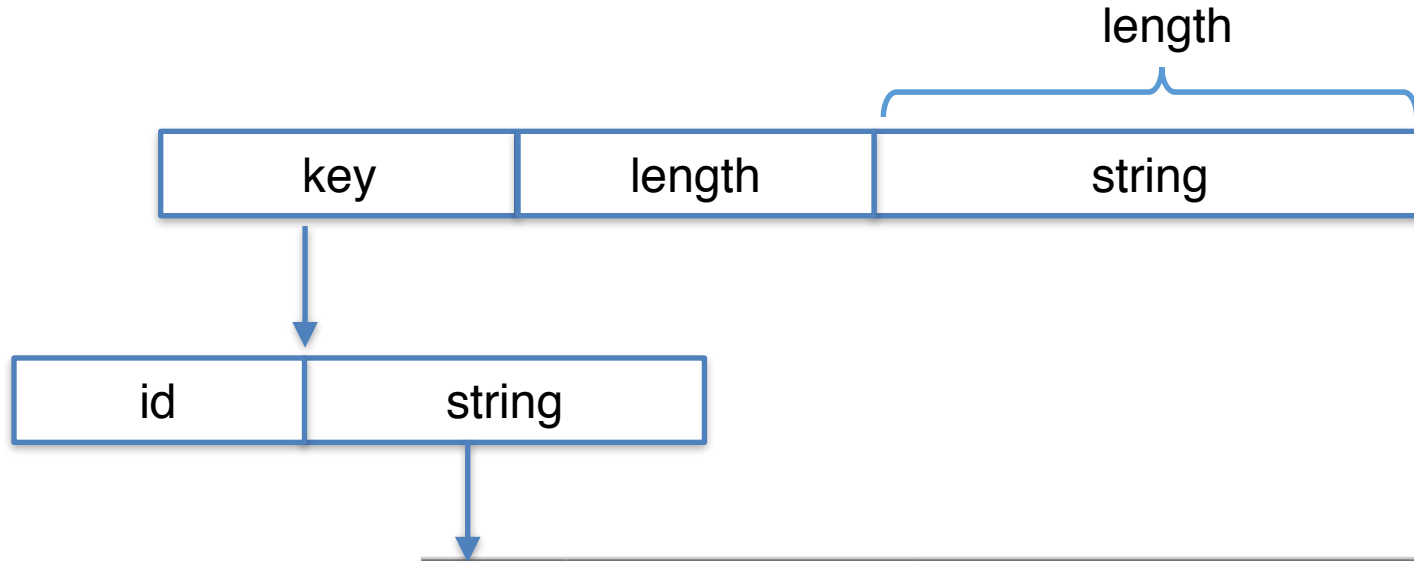
Type	Meaning	Used For
0	Varint	int32, int64, uint32, uint64, sint32, sint64, bool, enum
1	64-bit	fixed64, sfixed64, double
2	Length-delimited	string, bytes, embedded messages, packed repeated fields
3	Start group	groups (deprecated)
4	End group	groups (deprecated)
5	32-bit	fixed32, sfixed32, float

Protobuf解析



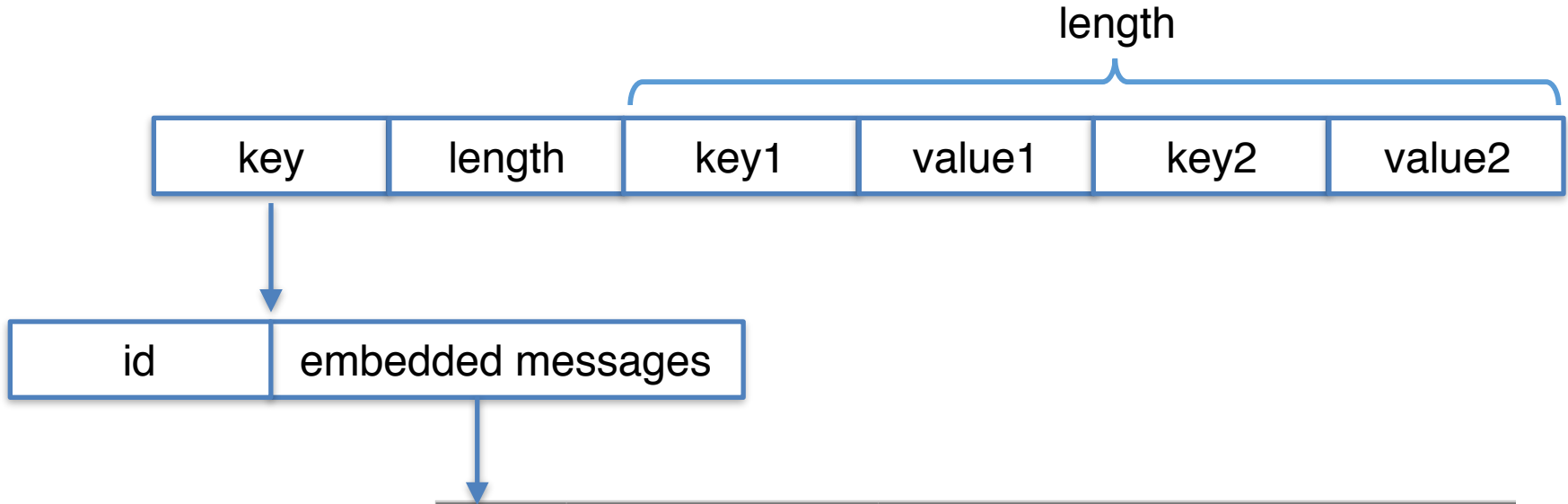
Type	Meaning	Used For
0	Varint	int32, int64, uint32, uint64, sint32, sint64, bool, enum
1	64-bit	fixed64, sfixed64, double
2	Length-delimited	string, bytes, embedded messages, packed repeated fields
3	Start group	groups (deprecated)
4	End group	groups (deprecated)
5	32-bit	fixed32, sfixed32, float

Protobuf解析



Type	Meaning	Used For
0	Varint	int32, int64, uint32, uint64, sint32, sint64, bool, enum
1	64-bit	fixed64, sfixed64, double
2	Length-delimited	string bytes, embedded messages, packed repeated fields
3	Start group	groups (deprecated)
4	End group	groups (deprecated)
5	32-bit	fixed32, sfixed32, float

Protobuf解析



Type	Meaning	Used For
0	Varint	int32, int64, uint32, uint64, sint32, sint64, bool, enum
1	64-bit	fixed64, sfixed64, double
2	Length-delimited	string, bytes, embedded messages , packed repeated fields
3	Start group	groups (deprecated)
4	End group	groups (deprecated)
5	32-bit	fixed32, sfixed32, float

Protobuf解析

- ◆ 再帰的にtry catchで当たってみる。

Protobuf解析

◆ 再帰的にtry catchで当たってみる。

```
phone_number = person.phone.add()  
phone_number.number = number  
account = phone_number.account.fnum = 666.777
```

encode

```
.A.....  
".Zd;.O..?-.....  
.@*.....0800000.  
.....&D*.....0800  
000.E.....
```

protoc --decode_raw

```
1 {  
  1: "\343\202\217\343\201\237\343\201\227"  
  2: 1234  
  4: "Zd;\3370\215\363?-\262\235\357\247\306\t@"  
  5 {  
    1: 1  
    2: "0800000"  
    3 {  
      1: 0x4426b1ba  
    }  
  }  
  5 {  
    1: 1  
    2: "0800000"  
  }  
}
```

自作decoder

```
{  
  "01:00:embedded message": {  
    "01:00:string": "\u308f\u305f\u3057",  
    "02:01:Varint": 1234,  
    "04:02:bytes": "0x5a:0x64:0x3b:0xdf:0x4f:0x8d:",  
    "05:03:embedded message": {  
      "01:00:Varint": 1  
    }  
  },  
  "05:04:embedded message": {  
    "01:00:Varint": 1,  
    "02:01:string": "0800000"  
  }  
},  
  "02:01:32-bit": 3.140000104904175  
}
```

JSON出力、編集、
Re-encodingもサポート

float型

0625

Protobuf解析

◆ burp pluginにしてみる

id、timestamp、座標？

The image shows a Burp Suite interface with two panels. The left panel displays a request, and the right panel displays a response. Both panels have tabs for 'Raw', 'Headers', 'Hex', and 'Protobuf Plain Text'. The 'Protobuf Plain Text' tab is selected in both. The response contains a Protobuf message with several fields, some of which are highlighted with red boxes. A blue arrow points from the text 'id、timestamp、座標？' to the '03:03:embedded message' field. Another blue arrow points from the text '◆ burp pluginにしてみる' to the Protobuf Plain Text view.

```
HTTP/1.1 200 OK
Server: nginx/1.11.1
Date: Thu, 13 Oct 2016 02:39:41 GMT
Cache-Control: no-cache
Via: 1.1 google
Content-Type: text/html; charset=UTF-8
Alt-Svc: clear
connection: close
Content-Length: 17458

#1826ab02fc4f495d99df28f9ecc700bf.11
#286d957c7cf240fb8af316acaab9ffdc.16
#2904cb6d97bc7f76f8f2937e44d5d12a
#2a67e3e5113240dfa2243d2a36ff386.16
#3bc0d2f08694207aee61f9f55102052.11
#45e82b2b3828a7cbcc38b542a8f48b9
#60a569be3e0c4a9898589ffid44dc66be.16
#6684f96cb4f8492ba2de25459e0c800c.16
#76630df8f0f1498e8d433ee16784a1e1.16
#9b68c5a10a396c26f7ad7de32fabcb1a
#a9f82fd0c9aed20d50936fef3061d6617
#c900f53d31af44cda5e17e08d9f8f5c7.16
#f9111aa6421540d6adcbfa8ac7ed8337a.11
#60188b58357
#125375d8158d423bb612b3e45d0b3e983.16
#236b7b154f884030b7bc4e4aca688a3e.16
#36796bf4d0ff4be19257be0d3a27a5d4.16
#36a3c163f61d4d4db92587d8c14ba5e7e.16
#4234a5d919640a88497b41278fc6c3d.16
#845dc626a88a02e201b21e1a938f4fc0
#99785e09ace64d73b41c00038cb1dc8b.16
#ba0ce8658d344d2483d7cdda58228461.16
#c0abd0185632d4c449d80a3893927cc7b.16
#f54a0499f2354570927a212b22c3a304.16
#070657d3721e49118337cc024d44a563f.16
#2158e0f7e2714a4b670d7b317db3be9.16
#35933b3e7ff7413798c9291b9708be2d.16
#372afb23cc8643719a85ea996676c2b2.16
#4d7fd7c4b664b7db74c2cf9lefe7a.16
#506374c6d3020302af6c093b24565.16
```

```
{
  "01:00:Varint": 1,
  "02:01:Varint": 6665365729551843372,
  "06:02:embedded message": {
    "01:00:Varint": 6,
    "02:01:embedded message": {
      "01:00:Varint": 1
    }
  },
  "100:03:embedded message": {
    "01:00:embedded message": {
      "01:00:Varint": 6924437638229262336,
      "02:01:Varint": 1476326381572,
      "03:02:embedded message": {
        "01:00:atstring": "1826ab02fc4f495d99df28f9ecc700bf.11",
        "02:01:Varint": 1476325148319,
        "03:02:64-bit": 35.659817,
        "04:03:64-bit": 139.702231,
        "08:04:Varint": 1,
        "09:05:Varint": 1,
        "12:06:bytes": "0xf5:0x3"
      }
    },
    "03:03:embedded message": {
      "01:00:atstring": "286d957c7cf240fb8af316acaab9ffdc.16",
      "02:01:Varint": 1476266146757,
      "03:02:64-bit": 35.659096,
      "04:03:64-bit": 139.703461,
      "08:04:Varint": 1,
      "09:05:Varint": 1
    },
    "03:04:embedded message": {
      "01:00:atstring": "2904cb6d97bc7f76f8f2937e44d5d12a",
      "02:01:Varint": 1476325466492,
      "03:02:64-bit": 35.66016,
      "04:03:64-bit": 139.701907,
      "08:04:Varint": 1,
      "09:05:Varint": 1,
      "12:06:bytes": "0xf5:0x3",
      "15:07:Varint": 4
    }
  },
  "03:05:embedded message": {
    "01:00:atstring": "2a67e3e5113240dfa2243dd2a36ff386.16",
    "02:01:Varint": 1476249282668,
    "03:02:64-bit": 35.659027
  }
}
```

source: <https://github.com/nevermoe/protobuf-decoder>

目次

- セキュリティ診断概要
- バイナリー解析
- 通信解析
- 対策&まとめ

対策&まとめ

◆パケットでも、バイナリーでも、

Non-Cryptographic手法をやめよう

- バイナリー暗号化、難読化！
- 通信暗号化！

◆**Game SecurityはDynamicで、**

一つの手法に頼ってはいけない

- 対処法：

開発 → Anti-cheat SDK → セキュリティ診断 → モニタリング

Q&A

Thank you!