セクコン株式会社 様

Androidアプリ 診断報告

担当：Team m1z0r3

m1z0r3（みぞれ）について

- 早稲田大学の情報セキュリティの研究室の有志で活動しているCTFチーム

写真の出典:
http://www.atmarkit.co.jp/ait/articles/1507/03/news101.html

主な活動実績

2013:
m1z0r3結成
SECCON決勝大会出場

2014:
MWS Cup 準優勝
SECCON決勝大会出場

2015:
危機管理コンテスト
経済産業大臣賞

発表について

- ゲームのセキュリティ診断士として、**セクコン株式会社の開発者および経営者**に対して報告を行う

- ゲームの仕様が知らされていない中で、アプリを解析するだけで、チートの可能性等のセキュリティ上の問題がないか調査した

セクコン株式会社 様

Androidアプリ 診断報告

担当：Team m1z0r3

目次

- 診断概要
- セキュリティの重要性
- チートの基本的な手法
- 診断の総評
- 通信改ざんによるスコアのチート（SandBag1）
- 通信の偽装による大量のアカウント作成（SandBag1）
- メモリ改ざんによるチート（SUNIDRA）
- まとめ

診断概要

- 依頼について
  - 現在開発中のゲームのセキュリティ診断
  - セキュリティ上の問題を引き起こす手法、影響度、対策方法について報告を行う

- 診断環境
  - 利用した端末：Nexus 7 (Android 4.4.4)
  - 診断日時：2015/08/01 11:00 ~ 2015/08/10 23:59

セキュリティの重要性

セキュリティ問題が起こることによる被害

ユーザー	ゲームがつまらなくなる 課金が無駄になる
開発者	クラッカーへの対応に追われる リリース後の対策は、変更箇所が多くなる
経営者	ユーザーからの苦情の対応が生じる 企業の信用低下＆売り上げが下がる

チートとは

- 開発者が想定していない手法を利用して、高得点を叩き出したり、有利にゲームを進めること
- アプリやサーバーとの通信に対しクラックを行う

オンラインゲームの構成

アプリ

通信

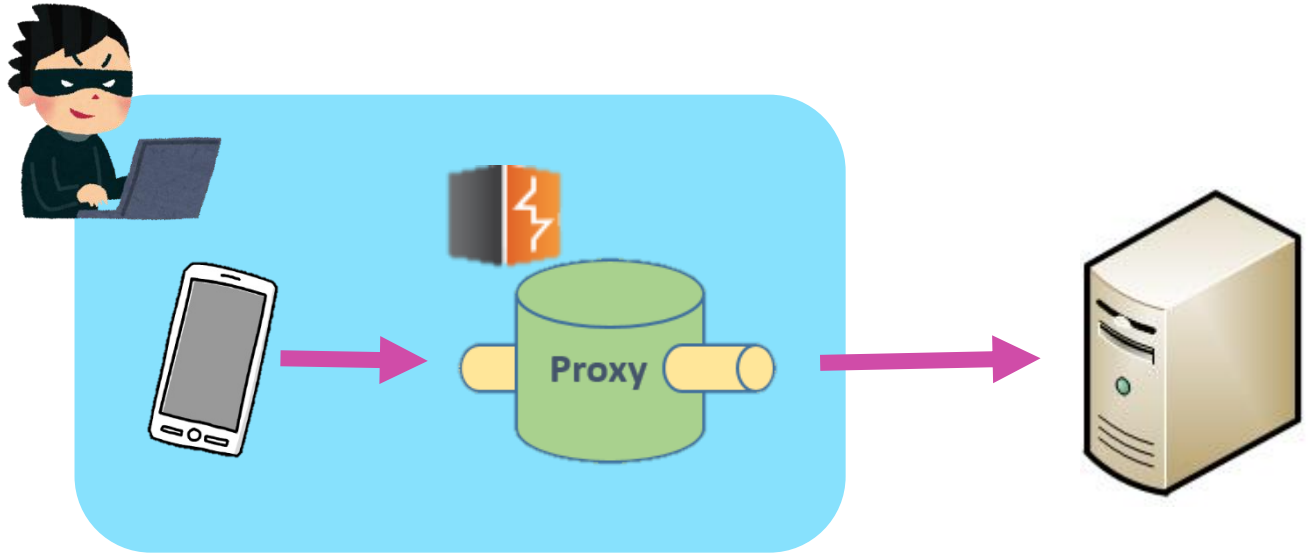
- スコア
- ユーザー情報

サーバー

チートの基本的な手法

パターン①　Proxyの利用

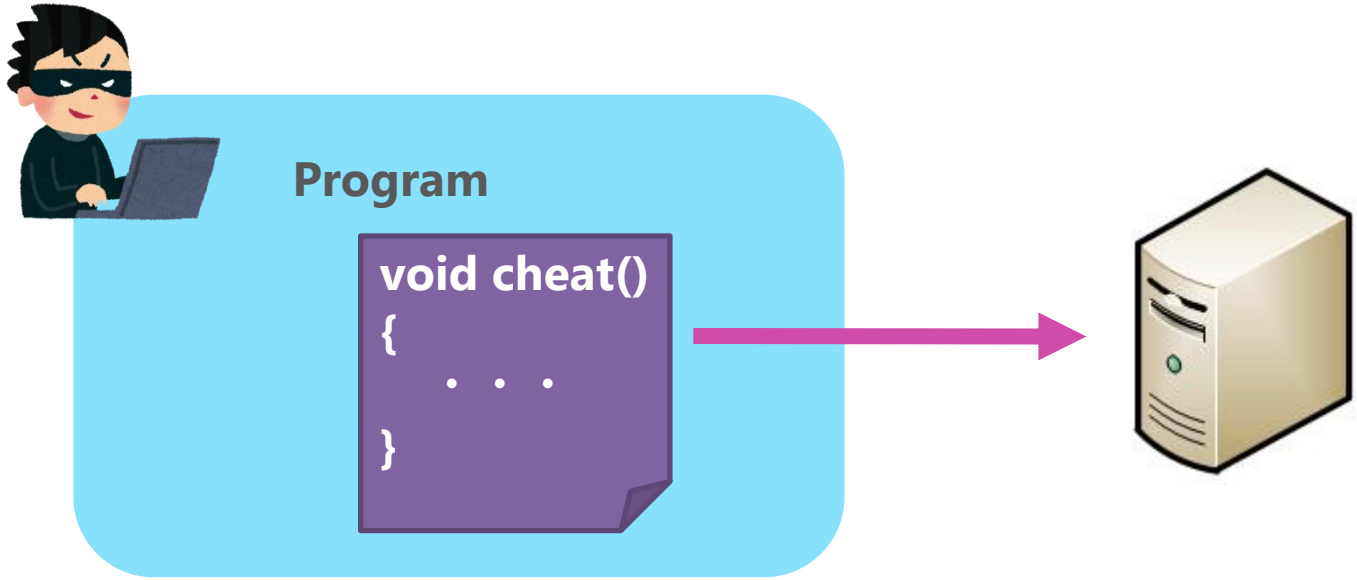
- Proxyを利用してAndroidの通信を改ざんし、サーバーに送信を行う



チートの基本的な手法

パターン②　通信の偽装

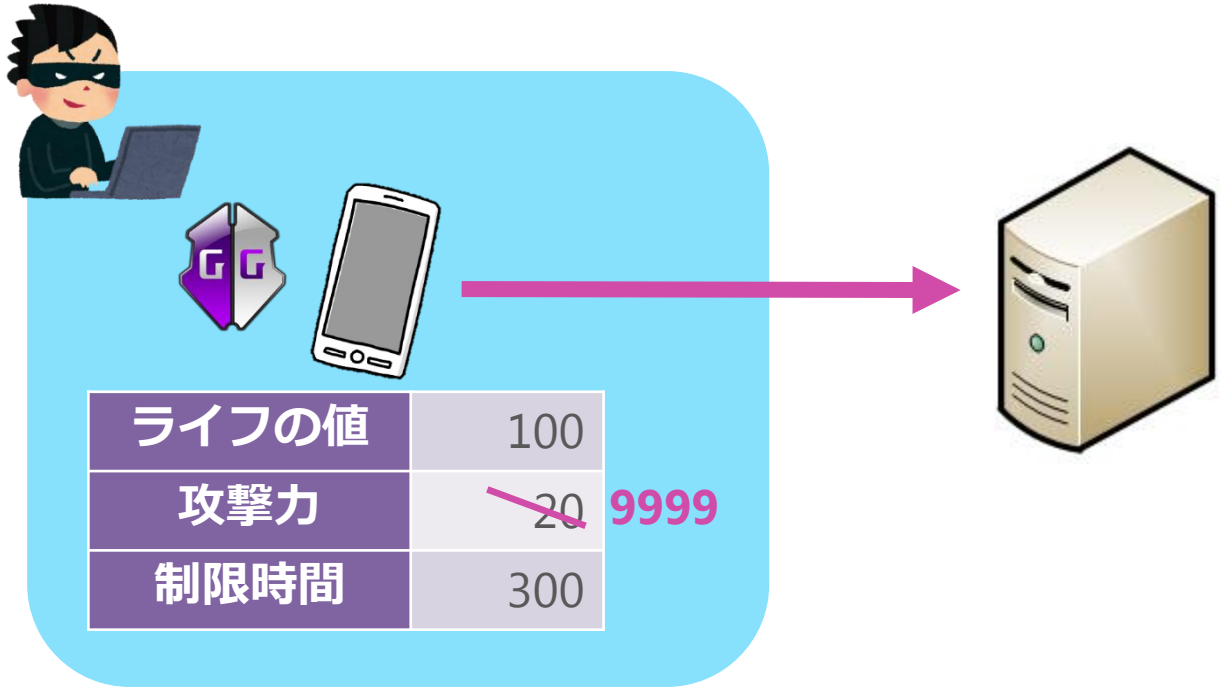
- プログラムを用いて通信を偽装し、サーバへリクエストを送る



チートの基本的な手法

パターン③　メモリの改ざん

- 端末のメモリを改ざんし、パラメタを書き換える

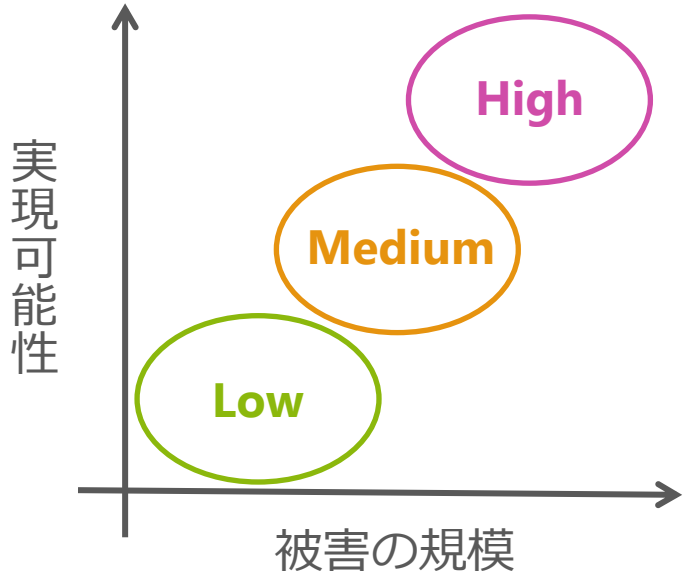


目次

- 診断概要
- セキュリティの重要性
- チートの基本的な手法
- 診断の総評
- 通信改ざんによるスコアのチート（SandBag1）
- 通信の偽装による大量のアカウント作成（SandBag1）
- メモリ改ざんによるチート（SUNIDRA）
- まとめ

セキュリティ問題の評価指標

- 問題の危険度を「**実現可能性**」と「**被害の規模**」を軸として3段階に分ける



- 攻撃の実現可能性(縦軸)：環境要因によるものや人的要因を考慮
- 被害の規模(横軸)：被害者数や被害の種類などを考慮

総評

- ゲームの面白さを損ねかねない影響度の大きな問題点が複数発見された
- いくつかの初歩的なセキュリティ対策が行われておらず、容易にチートが可能である

発見された問題一覧

アプリ	項目	危険度
SandBag1	**スコアの改ざんが容易**	High
	アカウントの大量作成が可能	High
	他ユーザーの名前を書き換えることが可能	Low
	他ユーザーの通信改ざんが可能	Low
SUNIDRA	**メモリ改ざんによるチートが可能**	High
	メモリ改ざんによるキャラクターのパラメタの改ざんが可能	Medium

SandBag1
- スコアの改ざんが容易
- アカウントの大量作成が可能

SUNIDRA
- メモリ改ざんによるチートが可能

発見された問題一覧

アプリ	項目	危険度
SandBag1	**スコアの改ざんが容易**	High
	アカウントの大量作成が可能	High
	他ユーザーの名前を書き換えることが可能	Low
	他ユーザーの通信改ざんが可能	Low
SUNIDRA	**メモリ改ざんによるチートが可能**	High
	メモリ改ざんによるキャラクターのパラメタの改ざんが可能	Medium

SandBag1
- スコアの改ざんが容易
- アカウントの大量作成が可能

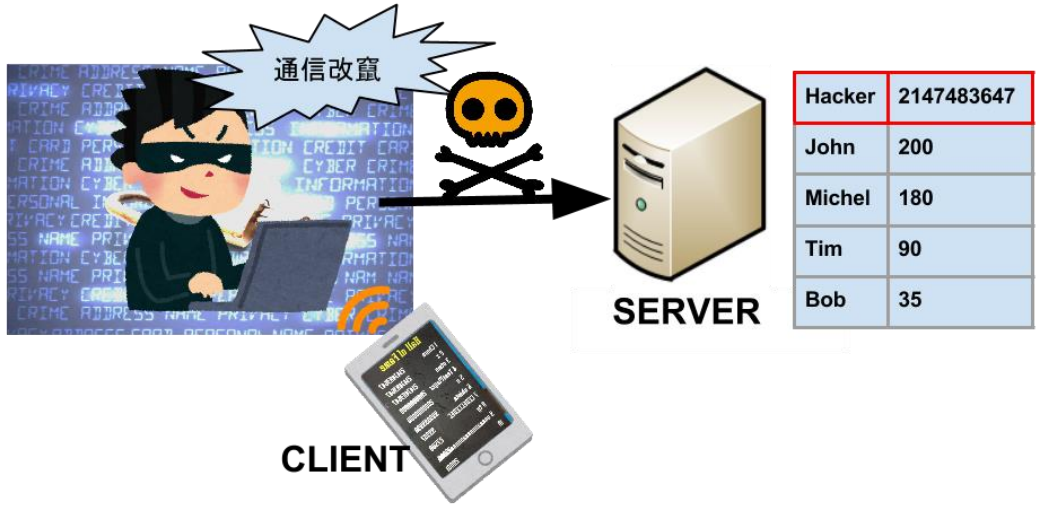
SUNIDRA
- メモリ改ざんによるチートが可能

SandBag1

スコアの改ざんが容易

【概要】スコアの改ざんが容易

- サーバーに送信されるスコアを書き換えることができ、不正にランキング上位に入ることが出来る



【再現手法】通信の改ざんが容易

- Proxyを用いて通信を中継する際にリクエストを書き換える

HTTPリクエストヘッダ

Raw | Params | Headers | Hex

```
POST /score/ranking/ HTTP/1.1
X-Unity-Version: 5.0.2f1
Content-Type: application/json; charset=UTF-8
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.4.4; Nexus 7 Build/KTU84P)
Host: api.sandbag2015.net
Connection: Keep-Alive
Accept-Encoding: gzip
Content-Length: 74

{"uuid":"e26fbae0-a53b-49e4-b83a-d0e6ddbf11d4","name":"otms","point":4400}
```

スコア情報

【再現手法】通信の改ざんが容易

- Proxyを用いて通信を中継する際にリクエストを書き換える

HTTPリクエストヘッダ

Raw | Params | Headers | Hex

```
POST /score/ranking/ HTTP/1.1
X-Unity-Version: 5.0.2f1
Content-Type: application/json; charset=UTF-8
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.4.4; Nexus 7 Build/KTU84P)
Host: api.sandbag2015.net
Connection: Keep-Alive
Accept-Encoding: gzip
Content-Length: 74

{"uuid":"e26fbae0-a53b-49e4-b83a-d0e6ddbf11d4","name":"otms
```

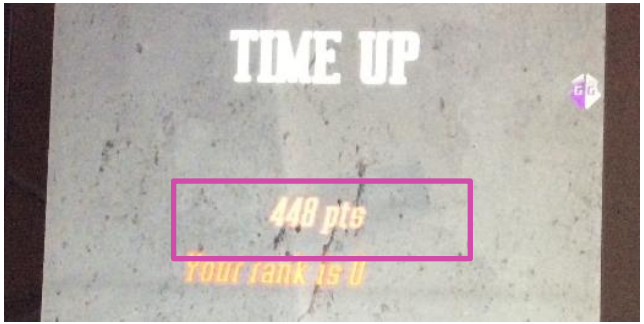
スコア情報

99999

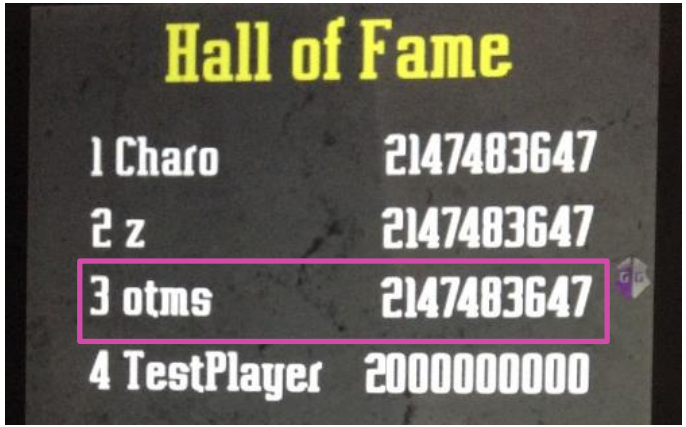
【再現手法】通信の改ざんが容易

- 送信する値を書き換えて上位にランクインできる

実際のスコア
448 pt



改ざん後のスコア
2147483647 pt

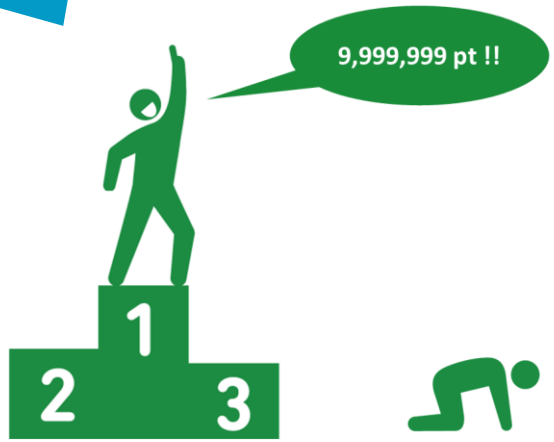


【影響】通信の改ざんが容易

SandBag1
- **スコアの改ざんが容易**
- アカウントの大量作成が可能

SUNIDRA
- メモリ改ざんによるチートが可能

通常プレイでは到達できない
スコアが１位にいる



モチベーション
の低下

【対策 1/3】通信の改ざんが容易

- サーバー側で異常な値を送信してくるアカウントのデータを削除する
  - 開発の必要はない
  - DB内の不正なユーザーを削除することで一般ユーザーに対する影響を一時的に取り除くことができる

【対策 2/3】通信の改ざんが容易

- 値の改ざんが行われていないかチェックするパラメタを追加する
  - クライアントとサーバーサイドでの開発が必要。必要な工数は少ない
  - pointから計算できるハッシュ値も同時に送るようにして、値とハッシュ値の関係が正しいかを判定する

【対策 2/3】通信の改ざんが容易

サンプルコード（クライアント側）

uuidとスコアを連結したものをハッシュ化

```
1  using System.IO;
2  using System.Text;
3
4  string MD5hash(string uuid, int point)
5  {
6      byte[] data = System.Text.Encoding.UTF8.GetBytes(uuid + ':' + point.ToString());
7      System.Security.Cryptography.MD5CryptoServiceProvider md5 =
8          new System.Security.Cryptography.MD5CryptoServiceProvider();
9      byte[] bs = md5.ComputeHash(data);
10     md5.Clear();
11     System.Text.StringBuilder result = new System.Text.StringBuilder();
12     foreach (byte b in bs)
13     {
14         result.Append(b.ToString("x2"));
15     }
16     return result.ToString();
17 }
18 MD5hash("Hoge",10)
19
```

http://dobon.net/vb/dotnet/string/md5.html

【対策 2/3】通信の改ざんが容易

サンプルコード（サーバ側）

```php
<?php
function validate($uuid, $point, $hash_str)
{
    if (md5($uuid . ":" . $point) == $hash_str)
        return true;
    return false;
}
```

スコアの正当性をチェック

【対策 3/3】通信の改ざんが容易

- 通信内容の秘匿
  - 開発コスト高い
  - 通信内容のパラメタも暗号化することで、どのようなパラメタのやりとりが行われるかを秘匿する

【対策まとめ】通信の改ざんが容易

SandBag1
- **スコアの改ざんが容易**
- アカウントの大量作成が可能

SUNIDRA
- メモリ改ざんによるチートが可能

対策方法	開発コストの低さ	ユーザビリティ	セキュリティ強度	備考
ログ監視	○	○	×	根本的な解決ではない
パラメタ追加	△	○	△	パラメタの設定方法なども考える必要がある
通信内容の秘匿	×	○	○	

SandBag1
- スコアの改ざんが容易
- **アカウントの大量作成が可能**

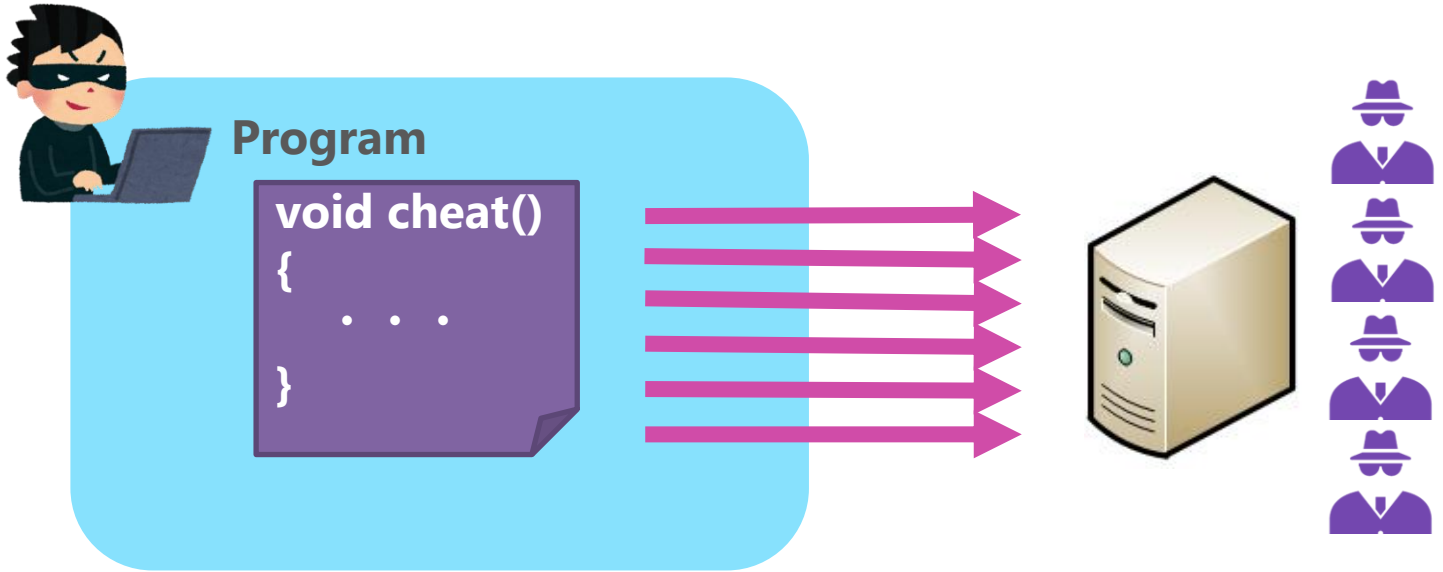
SUNIDRA
- メモリ改ざんによるチートが可能

SandBag1

アカウントの大量作成が可能

【概要】アカウントの大量作成が可能

- 偽装した通信を大量に送ることで、アカウントを量産し、ランキングを荒らすことができる



【再現手法】アカウントの大量作成が可能

- アカウントを大量に作成するためのスクリプトを用意する。

uuidの値を変化させながらリクエストを送信する

```
1 import os
2 from time import sleep
3 for i in range(0x10, 0xff):
4     os.system('http -j http://api.sandbag2015.net/score/ranking/
          uuid=e26fbae0-a53b-49e4-b83a-d0e6ddbf12' + hex(i)[2:] + '
          name=hoge point=-10')
5     sleep(1)
6
```

※httpコマンドについては、 https://github.com/jkbrzt/httpie を参照してください。

【再現手法】アカウントの大量作成が可能

- アカウントを大量に作成することでランキングが**大幅**に下がる

アカウント量産前
34位

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must
Connection: close
Content-Length: 13
Content-Type: text/html; charset=UTF-8
Date: Mon, 10 Aug 2015 05:25:22 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
Server: Apache/2.2.15 (CentOS)
Set-Cookie: PHPSESSID=dcqh8pc1l981k6m2o
X-Powered-By: PHP/5.3.3

{"rank":"34"}
```

アカウント量産後
276位

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-
Connection: close
Content-Length: 14
Content-Type: text/html; charset=UTF-8
Date: Mon, 10 Aug 2015 07:24:47 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
Server: Apache/2.2.15 (CentOS)
Set-Cookie: PHPSESSID=lqna296fkrj1qsp2il
X-Powered-By: PHP/5.3.3

{"rank":"276"}
```

※他ユーザーに影響を少なくするために、pointの値は負の値を用いている

【影響】アカウントの大量作成が可能

- ランキングの正当性が失われてしまうため、ユーザーはゲームに対しての信頼を失ってしまう

Rank	User
1	John
2	Bob
3	Tim
4	Hakcer
5	Me

アカウントの大量生成

Rank	User
1	John
2	Bob
3	Tim
4	Hakcer
5	Hacker
6	Hacker
・・・	・・・
99	Hacker
100	Hacker
101	Me

【対策 1/3】アカウントの大量作成が可能

- IPアドレスによる制限を設ける
  - 不正に大量作成を行っているユーザーがないかを監視する
  - 問題のあるアクセスを行っているユーザーがあった場合は、逐一そのIPアドレスからアクセスできないようにする

【対策 2/3】アカウントの大量作成が可能

- **CAPTCHA**の利用
  - ログインしていないユーザーもランキングに乗せたい場合の選択肢
  - ボット対策の有名な手法
  - 悪性ユーザーが大量にアカウントを作成することを防ぐことができる

引用：http://www.captcha.net/

【対策 3/3】アカウントの大量作成が可能

- ログイン機能を実装する
  - 開発コスト高い
  - 画面の追加が必要なため、ユーザービリティへの影響も大きい
  - メールアドレスやSNS連携（Twitter やFacebook）などで登録を行う
  - 登録が完了したユーザーのみランキングに反映させるようにする
  - 登録をしないユーザーは遊べるが、ランキングへの反映がされないように変える

【対策まとめ】アカウントの大量作成が可能

対策方法	開発コストの低さ	ユーザビリティ	セキュリティ強度	備考
IPアドレスによる作成制限を設ける	○	○	×	根本的な解決ではない
CAPTCHAの利用	△	×	○	
ログイン機能を実装する	×	△	○	

SandBag1
- スコアの改ざんが容易
- アカウントの大量作成が可能

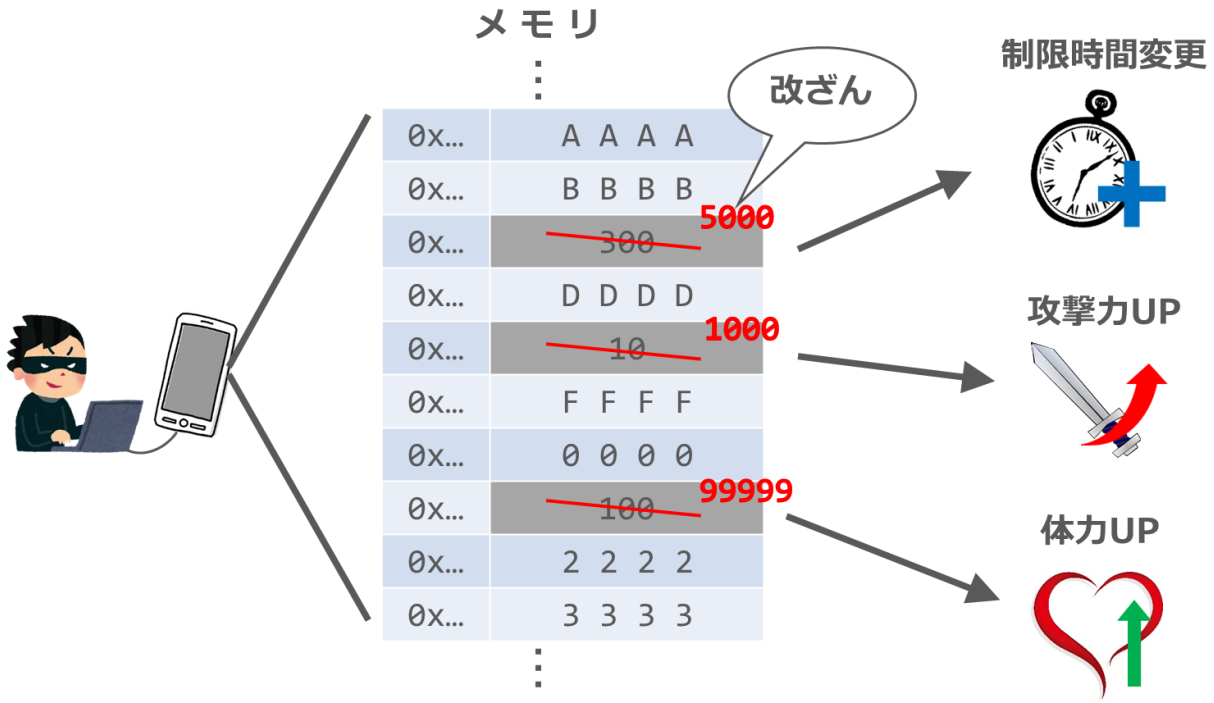
SUNIDRA
- **メモリ改ざんによるチートが可能**

SUNIDRA

メモリ改ざんによるチートが可能

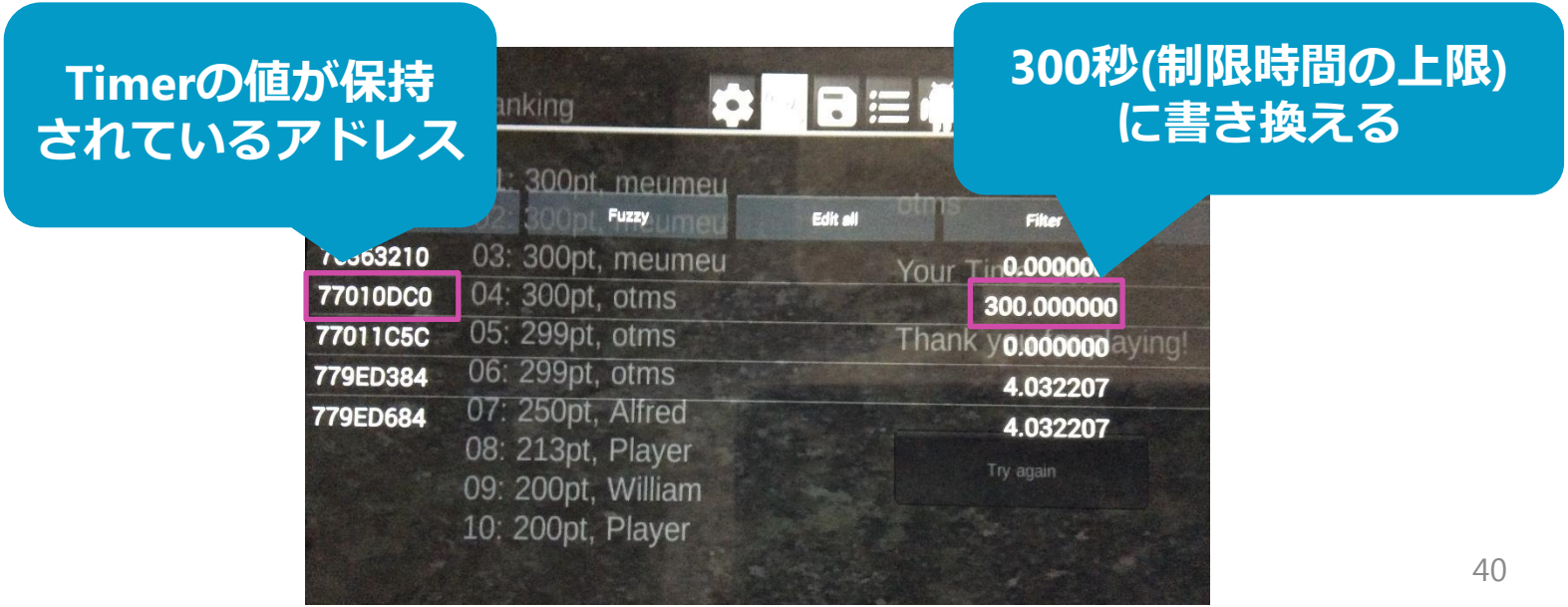
【概要】メモリ改ざんによるチートが可能

- メモリの改ざんにより、**制限時間**を改ざんすることができる



【再現手法】メモリ改ざんによるチートが可能

- Androidのメモリエディッタ（GameGuardian）を利用
- メモリ内からtimer の値を見つけて上限値に固定し、ゲームを通常にクリアする

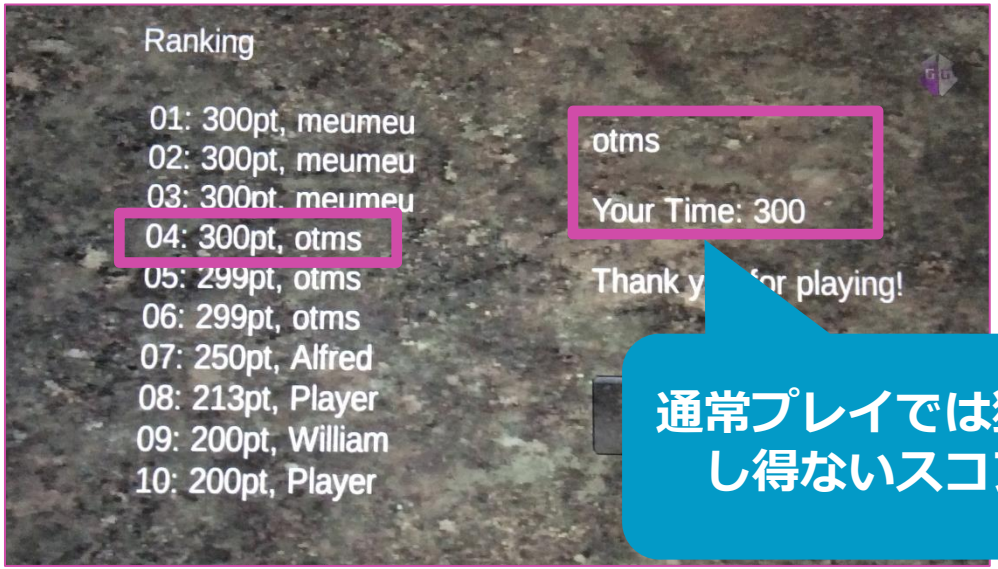


SandBag1
- スコアの改ざんが容易
- アカウントの大量作成が可能

SUNIDRA
- メモリ改ざんによるチートが可能

【影響】メモリ改ざんによるチートが可能

- ランキング上位に入ることが出来る
- 正攻法でやっているユーザーのモチベーションへ深刻な悪影響がある



【対策 1/3】メモリ改ざんによるチートが可能

- ルート化検知を導入する
  - クライアント側での開発が必要
  - 端末上でのメモリ改ざんではAndroidがルート化されている場合が多く、ルート化を検知することで対策になる
  - ルート化検知参考リンク：
    - https://blog.netspi.com/android-root-detection-techniques/
    - http://docs.unity3d.com/ScriptReference/AndroidJavaClass.html

【対策 2/3】メモリ改ざんによるチートが可能

- 値の検索を困難にする
  - クライアント側での開発が必要
  - 値をそのまま保持せず、メモリに対する検索などを妨害する
  - 例：値を定数値とXORしてメモリ上に保持

【対策 3/3】メモリ改ざんによるチートが可能

- メモリ改ざんが行われていないかチェックする
  - クライアント側での開発が必要
  - 時間の値が不正に改ざんされていないか判別する関数を、時間の更新と共に評価を行う
  - 適正な値か調べる（二重保持、ハッシュ値の保持）

SandBag1
- スコアの改ざんが容易
- アカウントの大量作成が可能

SUNIDRA
- メモリ改ざんによるチートが可能

【対策まとめ】メモリ改ざんによるチートが可能

対策方法	開発コストの低さ	ユーザビリティ	セキュリティ強度	備考
ルート化検知を導入する	△	○	△	ルート化検知の回避法が存在
値の検索を困難にする	×	○	○	
メモリ改ざんが行われていないかチェックする	×	○	○	

まとめ

- チートに関する基本的な手法の説明を行った
- SandBag1とSUNIDRAに対してセキュリティ診断を行い、セキュリティ上で特に問題と思われる点に対し、概要、再現手法、影響度、対策方法の観点で報告を行った

ご清聴ありがとうございました。

補足

ツール一覧

- Burp
  - https://portswigger.net/burp/
- GameGuardian
  - https://gameguardian.net/forum/forum/68-android/

暗号

平文

{"uuid":"e26fbae0-a53b-49e4-b83a-d0e6ddbf1200"、 "name":"otms" "point":4400}

暗号化

{"c":"eyJ1dWlkIjoiZTI2ZmJhZTAtYTUzYi00OWU0LWI4M2EtZDBlNmRkYmYxMjAwIiwgIm5hbWUiOiJv¥ndG1zIiAicG9pbnQiOjQ0MDB9"、 "j":"Q0MDB9"、 "k": "eyJ1dW"}

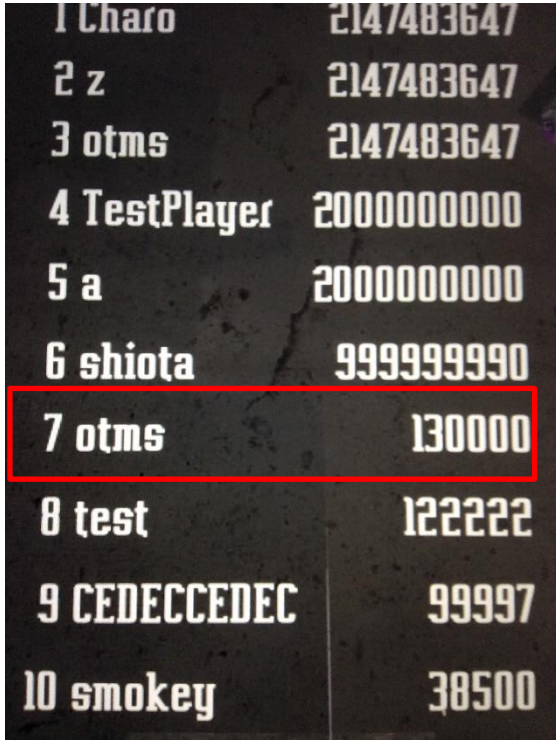
通信の改ざんが容易 再現手法

- Proxy を用いてHTTP リクエストを改ざんする。Proxy には Burp Suite Free Edition v1.6を用いた
- Android 端末のNetworkの設定でBurpを起動させているマシンをProxyとして設定し、サーバに対するリクエストをフックする
- リクエストのpointに関するパラメタを改ざんしサーバに送信する

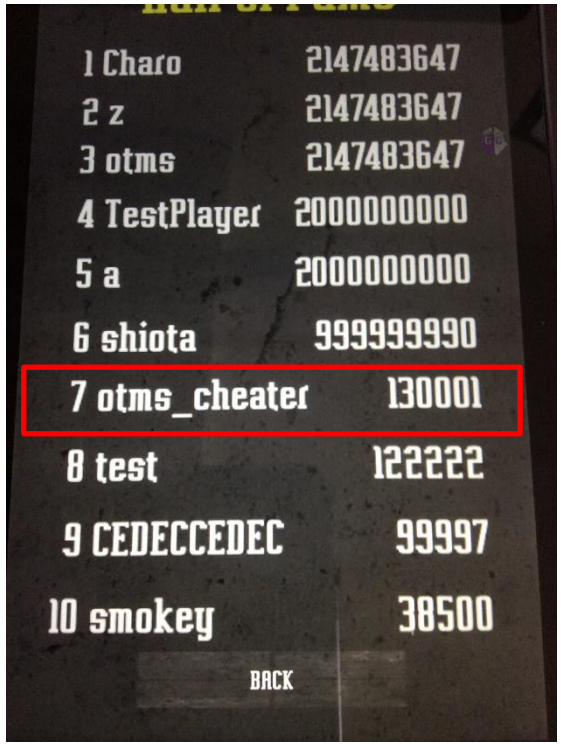
他ユーザーの名前を書き換えることが可能　概要

- 他ユーザーのuuidを入手できた場合、ユーザー名を書き換えることが出来る

書き換え前



書き換え後

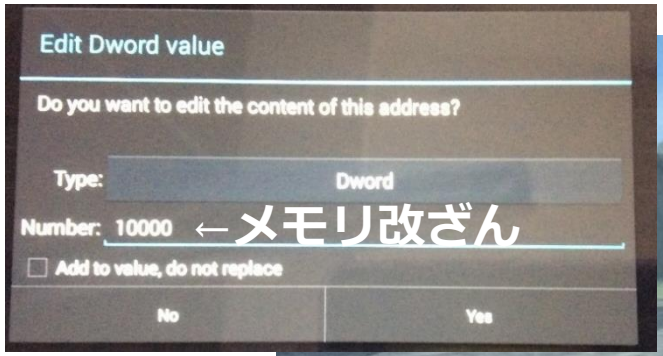


【再現手法】メモリ改ざんによるチートが可能

SandBag1
- スコアの改ざんが容易
- アカウントの大量作成が可能

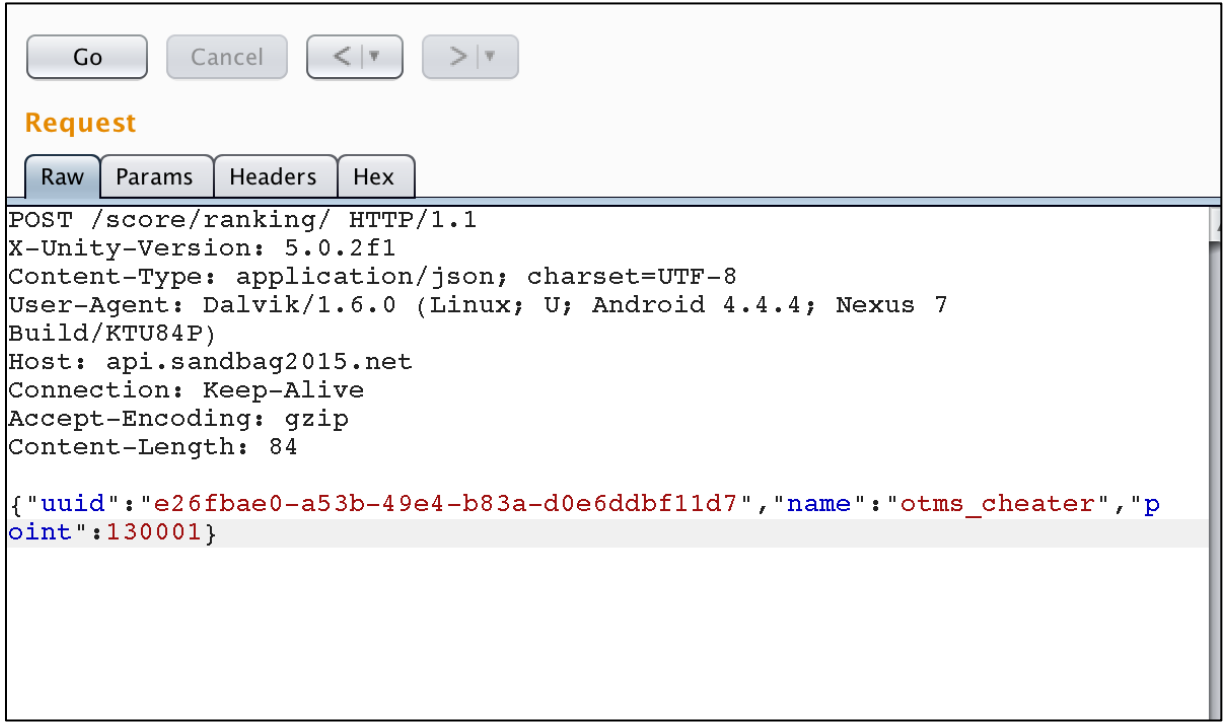
SUNIDRA
- メモリ改ざんによるチートが可能

- メモリの改ざんを行うことで、キャラクターのHPや攻撃力を改ざんすることができる



他ユーザーの名前を書き換えることが可能
再現手法

- 他ユーザーのuuidを入手できた場合、書き換えたいユーザー名とポイントの更新を行う情報をサーバーに送る。



他ユーザーの名前を書き換えることが可能
影響度

- uuidは、.NETの機能を用いて生成されており、ランダムで推測は難しく影響度が低い

- しかし、サーバーとの通信はHTTPで通信されていないため傍受されていたさいにはuuidが漏えいする危険性はある

他ユーザーの名前を書き換えることが可能
対策手法

- この攻撃にはpointの更新、uuidに対して認証が必要ないという問題1-1と1-2の条件が必要である

- そのため、これらの問題のどちらかが解決できれば、この攻撃も行えなくなる

他ユーザーの通信改ざんが可能

- サーバーとの通信がHTTPで行われているため、名前の変更やスコアの書き換えが可能である。
- 手法　ARP SpoofingなどでMITM攻撃をする。
- 影響度　低い
- 対策　HTTPS通信にする。
- 備考　サービスには深刻な影響を与える可能性は高いが、攻撃を行うための準備が難しいため、サービス全体への影響度は低い。