はじめに

@kusano_k と申します。前々からコミックマーケット（コミケ）で本を頒布してみたいと思っていました。とはいえ、画才も無ければ、ゲームを作ることもできません。何か頒布できるものは無いかと考え、CTF を開催してその解答を本にすることを思い立ちました。本書は、コミックマーケット C92 の 1 か月前（2017 年 7 月 11 日）から、下記の URL で開催していた CTF の各問題の解説です。この CTF はコミケ開催日で終了する予定ですが、1 か月程度は問題を解けるようにしておくつもりです。その後は、奥付の URL に問題ファイルや攻撃用のサーバーの構築法を記載しようと思っています。ぜひ本書を読みながら実際に手を動かして解いてみてください。

https://ksnctfc92.sweetduet.info

CTF とは Capture The Flag の略で、コンピューターセキュリティに関する技術と知識を試す競技です（Wikipedia より）。本来は「旗取りゲーム」の名の通り、互いのサーバーを攻撃しあってフラグを奪う攻防戦形式のコンテストのことを指していたのではないかと思いますが、今ではコンピュータセキュリティに関するコンテスト全般に使われているようです。ハッキングカンファレンス DEF CON で開催されるものが最も有名です。国内では官公庁の後援も受けている SECCON CTF が広く知られていて、毎年テレビのニュースに取り上げられています。また、去年から Google も Google CTF を開催しています。

筆者は 2012 年から ksnctf という名前で、これらのコンテストに出てくるような問題を出すサイトを開設しています。一般的な CTF は数日の期間で行われ、その後はサイトも閉鎖されますが、ksnctf では特に期間を定めず、のんびりと取り組めるようにしています。CTF に参加する時間を確保するのもなかなか大変なので、こういうのも良いですよね。ここ数年は問題を追加していないにも関わらず、「CTF に出たいならまずはこのサイトで練習しろ」と紹介されることもあり、また、CTF の会場でも「ksnctf がきっかけで CTF を始めた」と言われることもありました。嬉しく思っています。

CTF 終了後は自らの解法をブログに公開する人が多く、「コンテスト名 write-up」で検索すると見つかります。筆者もコンテスト終了後には解法を書くようにしています。せっかく本にするのですから、ツールの使い方や問題で扱われているテーマを、いつもの write-up よりは少し詳しく書いてみようと思います。本書を読んで、何か一つでも得られるものがあれば幸いです。

CTF を常設していて残念なことが一つありました。作者が思いつかなかった解法で解いている人が多くいると思うのですが、その解法を作者が知れないことです。答えであるフラグそのものを公開されると困るけど、そこにいたる解法は、まあ書かれても良いんじゃないかなと思っていま

すし、「解法を書いて良いか？」と訊かれたら、そのように答えていますが……。一方、問題を解くときにはウェブ検索を活用します。「問題を解こうとググっていたら、解法が出てきてしまった」という声を聞くこともあり、なかなか難しいですね。

今回はコミケ開催までと期限を切ったので、きっと、挑戦した人たちが記事を書いてくれるでしょう。楽しみにしています。読者も、本書を読むだけではなく他の挑戦者の記事を読むと、知識が深まることと思います。

第 1 章

準備

コンテストの概要

全 9 問です。 各問題には、GATE、E1〜E7、W という ID を振りました。GATE は入場ゲート、E1〜E7 は東館、W は西館のつもりです。C92 では西館は企業ブースのみなので、1 個にまとめました。その代わり、W は複数のジャンルを絡めたかったのですが、良いネタが思いつきませんでした。

早速ネタバレをしてしまいます。E1〜E7 の各問題には「裏フラグ」が存在します。GATE と W を含め、全てのフラグを投稿すると、裏フラグ用の問題と点数が表示される「裏モード」になります。

難易度は、本腰を入れて取り組んだ人ならば表の問題を全完できる、自分の得意なジャンルならば裏のフラグも取れる、というくらいを想定していました。執筆時点では思っていたよりも裏モードに入った人が少なく、せっかく作った裏フラグの問題はそもそも見ることもできない人が多いという状況で、ちょっと残念です。表と裏で分ける必要は無かったのでは……。

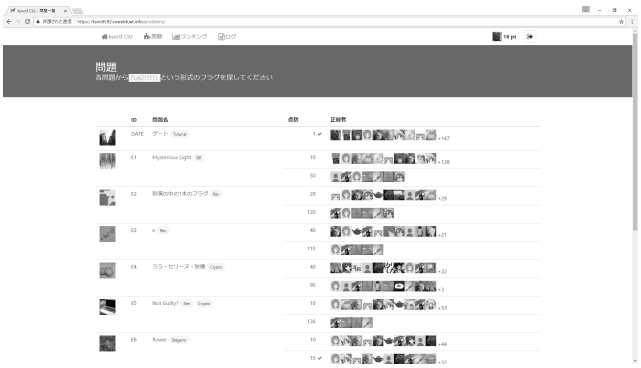

図 1.1　ksnctf C92

環境

問題と本書のプログラムやコードは下記の環境で動作することを確認しています。

プログラム・言語	バージョン
Windows	Windows 10 Pro 64 ビット
Linux	CentOS 7.2.1511 64 ビット もしくは Ubuntu 16.04 LTS 64 ビット
Python	2.7.11
C++	Visual Studio Community 2015 もしくは gcc 4.8.5
Go	1.6.3
Erlang	8.3/OTP 19

特に最新の機能や古い機能を使っているわけではないので、各言語の新しめのバージョンなら動くのではないかと思います。本書のソースコードを gcc でコンパイルするためには、`-std=c++1y` オプションを付けて、C++14 を有効にする必要があります。Python は 2 系列を使用しています。3 系列とは互換性が無いので、3 系列の Python を用いる場合にはソースコードの修正が必要になります。変換できることの確認はしていませんが、2to3[*1]が役に立つかもしれません。

本書中でシェルコマンドを入力している欄は、プロンプトが$の場合は Linux 上の bash、>の場合は Windows 上のコマンドプロンプトです。

[*1] https://docs.python.org/3/library/2to3.html

第 2 章

GATE ゲート

図 2.1　サイトの見栄えを良くするために各問題に画像を付けた。GATE の画像はアニメ中でゲートが出現した場所の写真

問題文は次の通りでした。

GATE	→	6473
SELF	→	531F
DEFENCE	→	D3F3NC3
FORCES	→	F0RC35
FLAG{WELCOME_TO_KSNCTF_C92}	→	FLAG{??????????????????????}

元ネタは「ゲート 自衛隊 彼の地にて、斯く戦えり[*1]」です。自衛隊の英称は Japan Self-Defense Forces。

[*1] http://www.gate-alphapolis.com/

まずフラグが通ったときの画面を見るとやる気が出るよね、ということで、ksnctf では 1 問目はテストとして問題文のフラグをコピペするだけで正解するようになっています。今回のコンテストでは少し捻りました。

表の左右の文字列を見比べると一部の文字が数字に置き換わっています。数字にならなかった文字はそのままです。また、同じ文字は必ず同じ文字になっています。4 個目までの文字列で置き換える文字を記録し、5 個目の文字列（フラグ）に適用すれば解けます。この程度ならば手作業でも解けますが、例えば C++ ならば次のようになります。4 個目までの文字列を連結して 1 個の文字列として扱っています。C++ では"hoge" "fuga"のように連続して文字列を書くと、1 個の"hogefuga"という文字列になります。

```
#include <iostream>
#include <string>
#include <map>

using namespace std;

int main()
{
    string s =
        "GATE"
        "SELF"
        "DEFENCE"
        "FORCES";
    string t =
        "6473"
        "531F"
        "D3F3NC3"
        "F0RC35";

    map<char, char> M;
    for (size_t i=0; i<s.size(); i++)
        if (s[i] != t[i])
            M[s[i]] = t[i];

    string flag = "FLAG{WELCOME_TO_KSNCTF_C92}";
    for (char &c: flag)
        if (M.count(c) != 0)
            c = M[c];
    cout<<flag<<endl;
}
```

```
FLAG{                    }
```

ということで、フラグは FLAG{ }です。誤答を見ていると、FLAG の部分は置き換えるのか？ とか、92 という数字を英字にする必要はあるのか？ とか、4 個目までに出てこない数字に置換できそうな文字はどうするのか？ とか、意外と悩みどころがあったようです。すみません。

この問題のように英字を似た数字や記号に置き換えた表記は「leet[*2]」と呼ばれます。ハッカーのお遊びですね。CTF においては、想定解法の leet がフラグになっているものをよく見ます。例えば、問題の想定解法が WEP の解読であれば、フラグが FLAG{w3P_1s_v3Ry_w34K!!!}など。問題の想定解法からフラグが推測されることを防ぐためでしょう。なお、ksnctf C92 では、この問題以降のフラグは leet ではなくランダム文字列です。

[*2] https://ja.wikipedia.org/wiki/Leet

第 3 章

E1 - Mysterious Light

図 3.1　雪中の温泉のイメージ。実際はただの雪山

タイトルの「Mysterious Light」は、「謎の光」を英訳したものです。「謎の光とは主に大人の事情により発せられる光である」（Pixiv 大百科[*1]より）。アニメの温泉回で、女の子の前に不自然に射してくるあの光です。<!––ここにサンプル画像が欲しいけど、怒られそうだから止めておこう–––>

「重要な部分が隠された QR コードを解読せよ」というのがこの問題の趣旨です。QR コードとセキュリティに何か関係があるのか？ 筆者もあまり関係は無いと思っていますが、SECCON ではよく見る問題だったので、出題してみました。

QR コード

本書の読者で、スマートフォンで QR コードを読み取った経験が一度も無いという人はいないでしょうから、QR コードがどのようなものかという説明は省きます。QR コードを開発したのは自

[*1] https://dic.pixiv.net/a/%E8%AC%8E%E3%81%AE%E5%85%89