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

菊池研究室 4年
山本拓巳

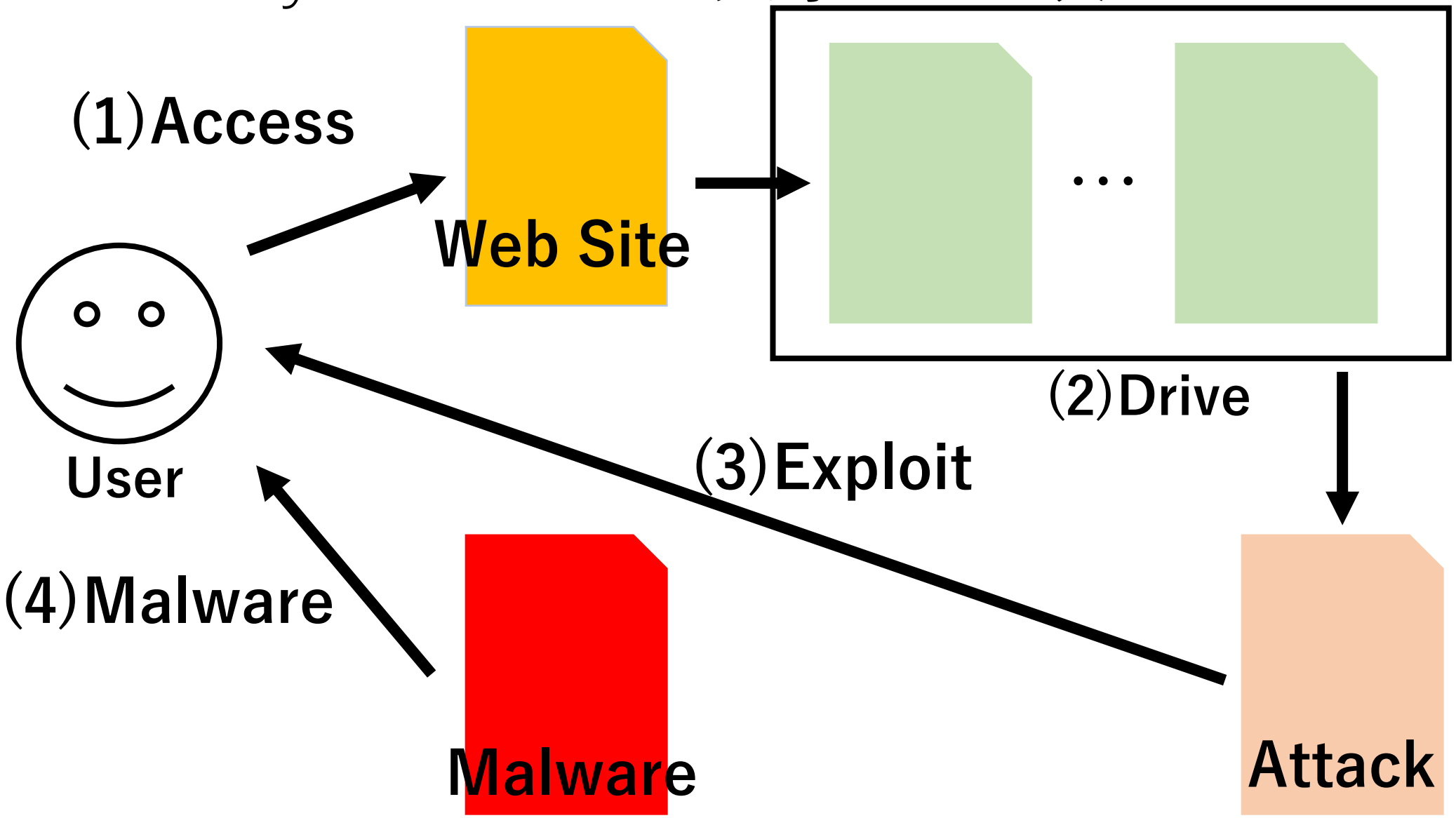
Drive-by Download攻撃

- 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

- 攻撃の際に複数のサイトを経由(Drive)させ，マルウェアをダウンロード(Download)

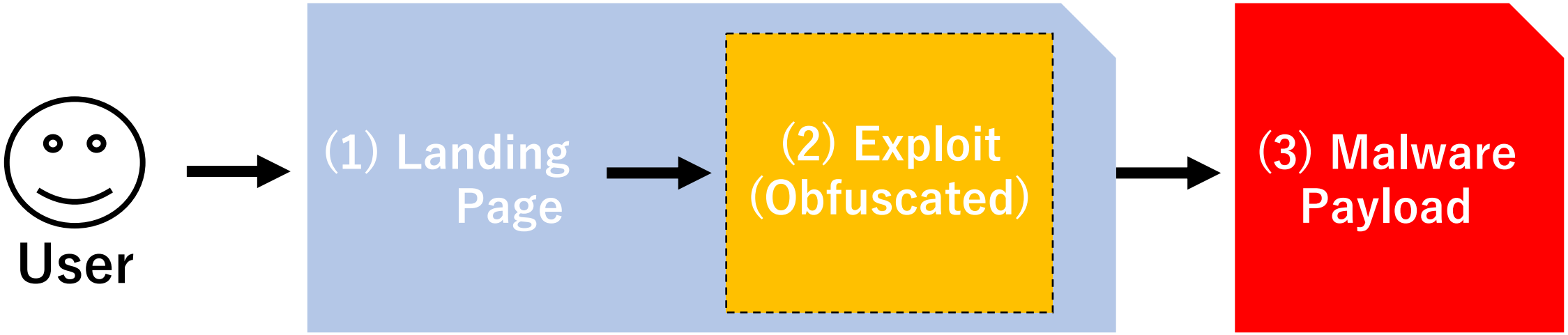
- ほとんどの場合，Exploit Kitと呼ばれる攻撃用ツールキットを使用

Drive-by Download攻撃の手順



Exploit Kit

- 攻撃サイトとマルウェア配布サイトの処理を担うツールキット
  ex)RIG Exploit Kit，Sundown Exploit Kit etc.
- 攻撃者はExploit KitのURLにユーザをリダイレクトするだけで攻撃が可能
- Exploit Kitを管理・販売する業者が存在し，攻撃者はDeep Web等で購入し使用



コードの難読化

- Exploit Kitで実行される攻撃コードは難読化されていることが多い
  - ex)意味のないコメントアウトの追加
    kiknlab → /*Oapc703ka*/kiknlab/*ha1nlDs28*/
    エンコード
    kiknlab → a2lrbmxhYg== (Base64)
    暗号化
    kiknlab → hfhkixy (シーザー暗号)

- 主に解析妨害が目的
  - 難読化によって攻撃内容の把握を困難にする

研究目的

難読化の手法とExploit Kitの種類に関連性があるか明らかにする

解決手法

- Drive-by Download攻撃の観測データを用いておおまかなトラフィックを確認し，観測されたExploit Kitを分類する

- 観測されたExploit Kitについて，難読化が施されたJavaScriptコードの解析を行い難読化手法を明らかにする

使用するデータ

- MWS Cup 2017に提出された50件のDrive-by Download攻撃観測データセット

- 高対話型クライアントハニーポット StarC(＊)を使って観測された
  - 悪性URLへアクセスした際のpcapファイル，スクリーンショット，Tempフォルダ内のexeファイル等を収集

＊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」，
(https://windy.mind.meiji.ac.jp/paper/2017/bachelor/resume/koike.pdf)

解析結果

- 50件の内全ての攻撃でExploit Kitが使用されていた
  - Terror Exploit Kit：2件，RIG Exploit Kit：48件

Exploit Kit	数
Terror Exploit Kit	2
RIG Exploit Kit	48

RIG Exploit Kit

- Landing Pageの攻撃コードが難読化されており，そのままの状態では攻撃の内容を把握することは難しい
  - Landing Pageは3つのスクリプトで構成されており，難読化の手順は全てのスクリプトにおいて同一

```
<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>
<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[BtJ
</script>
<script>aQloMZDuwt="fs*�j741�534�247�r;/�re�;�dfg�r+�xbx�-10�8�a��1�q�ile�2;�q+�6�];b�xcvx�5fs�6hfj�d5�s
/*g96g148fn*/
KgCrqIlCrS="/*�48d5�hfj5�fs*/�tion�d�var�,�cume�/*s8�d3�hfj7�6fs�bc�eat�le�]�pt�s81�5d60�hf�29f�[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>
```

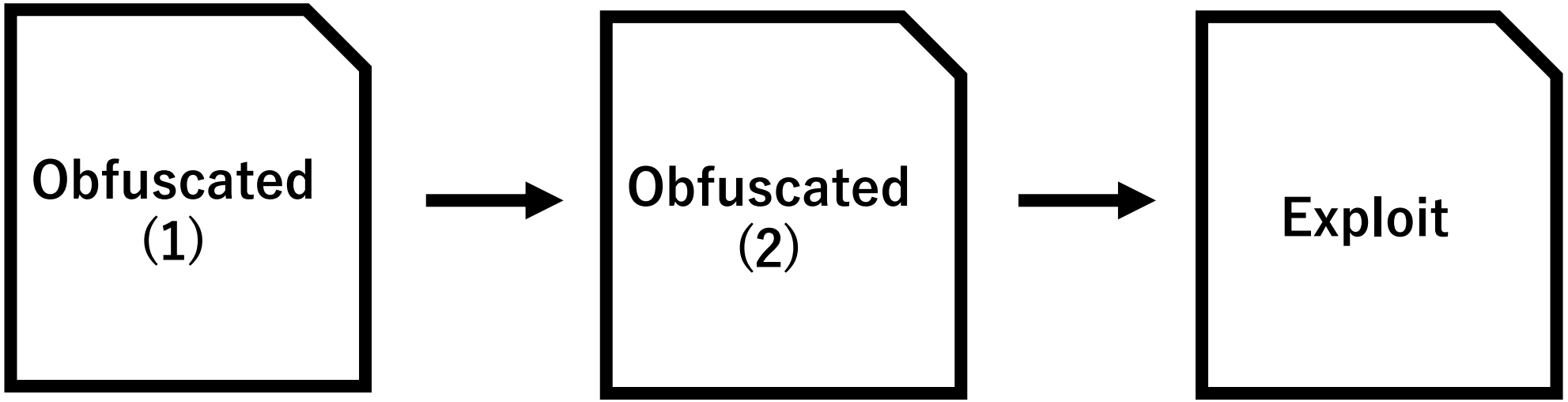
section1

section2

section3

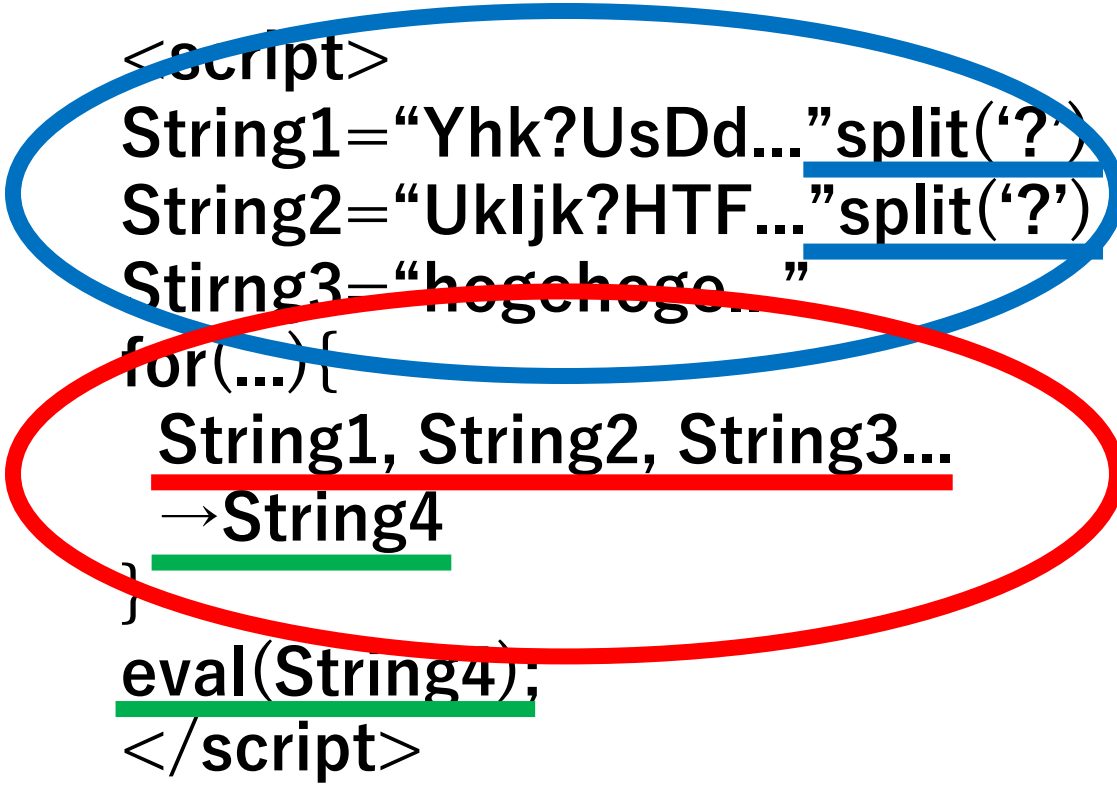
RIG Exploit Kitにおける難読化

- 実際に脆弱性を突くコードが実行されるまでに大まかに2段階の難読化が施されている
- 難読化されているだけなので，順を追って実行していけば最終的な実行内容を得ることができる



Landing Pageの難読化 (1)

- 1セクションは3つの文字列，1つのfor文で構成される
- String1,String2は後ろでsplit()されており，for文内でString1,String2のリストとString3を使った処理が行われString4を生成する
- 生成された文字列はeval()によって実行される



Landing Pageの難読化 (2)

- String4は2つの関数で構成される
- 関数Aが実行され，定義されたString XをBase64デコードして返り値に設定している
- 関数Bでは新しいスクリプトを生成し，そこで関数Aの返り値が設定されている

```
function A(){
  var X=“ugUdgrfHy...”
  var T=“ABCDEF...”
  ...
}

function B(){
  var a = A();
  c=document,
  b=c[“createElement”](“script”);
  ...
}
```

全体の難読化の傾向

- Terror Exploit Kit
  難読化は施されていなかった

- RIG Exploit Kit
  複数のサーバーのExploit Kitが観測されていたが，Landing Pageの構成や難読化手法は全て同じ

  → 難読化方法でExploit Kitの分類は可能？

まとめ

- 50件のDrive-by Download攻撃の観測データについて，Exploit Kitの種類と難読化種類の注目した解析を行った

- Exploit Kitの種類毎に難読化手法の違いはあるが，同一種類Exploit Kit同士では差がほぼなかった

- Exploit Kitは流行りの移り変わりが早く，2017年の観測データでは古い
  - 現在ではGrandSoft Exploit Kit，Fallout Exploit Kit