# DanaBot's New Tactics and Targets Arrive in Time for Peak Phishing and Fraud Season

By Authors & Contributors

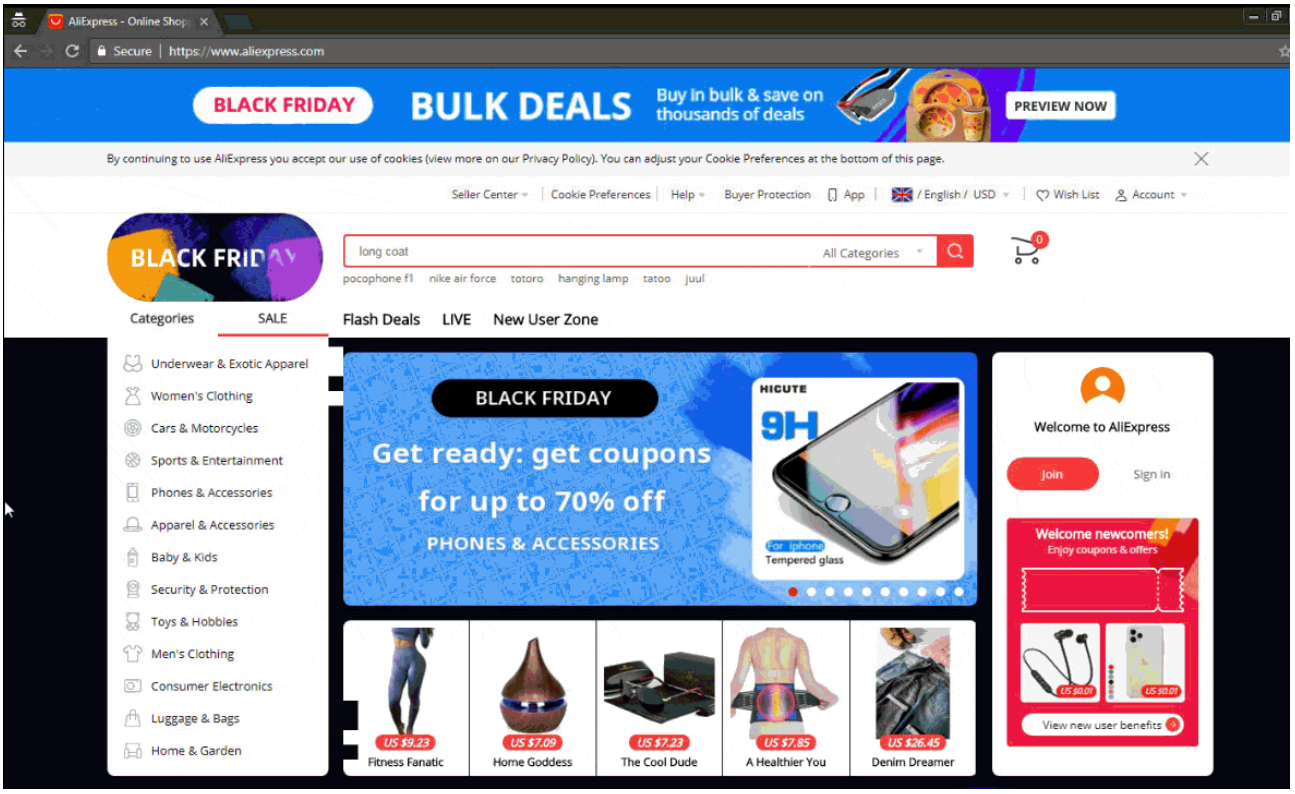
Archived: 2026-04-05 18:59:38 UTC

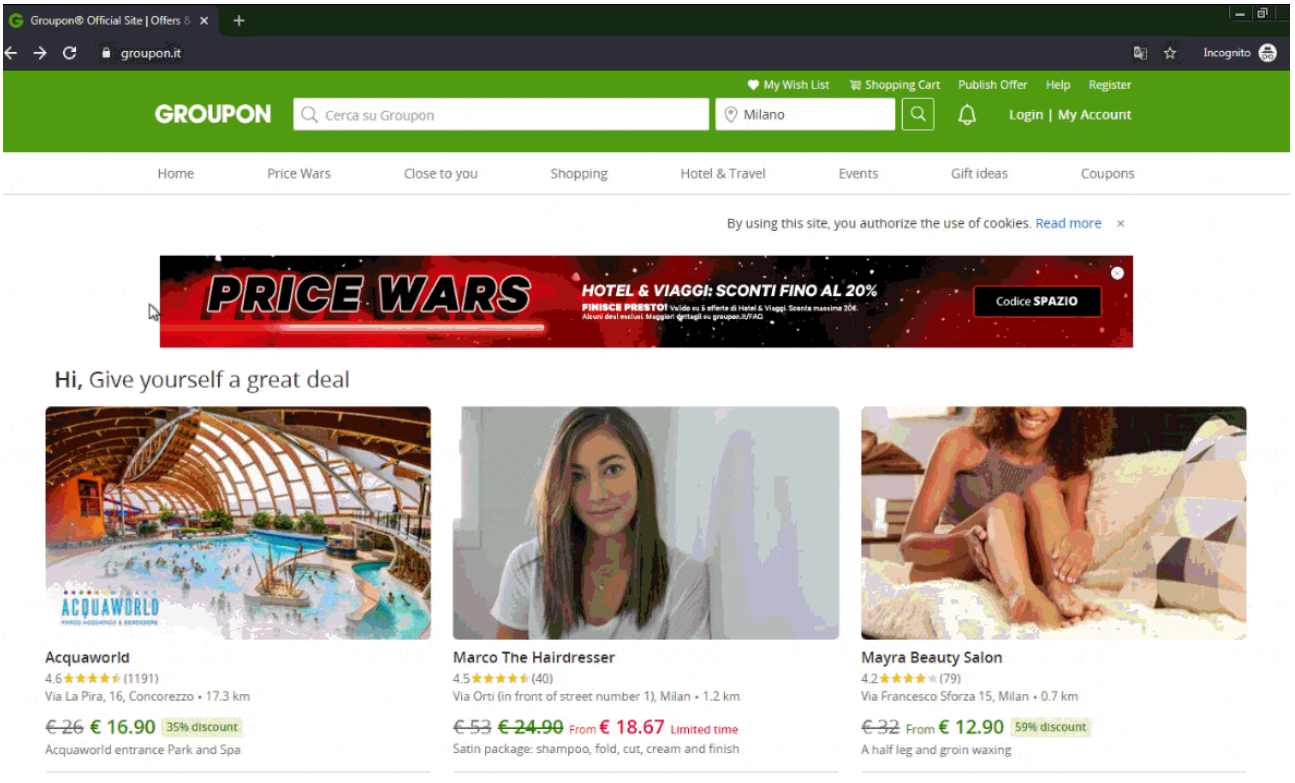
First detected in May 2018,[1] DanaBot is a powerful banking trojan that has historically focused heavily on financial services institutions in both Australia and Europe. F5 Labs has been following DanaBot since November 2018, when we began publishing campaign updates. In August 2019, we included it in our Reference Guide to the Malware Family Tree (/content/f5-labs-v2/en/archive-pages/education/banking-trojans-a-reference-guide-to-the-malware-family-tree.html). DanaBot has grown quickly since it was first detected, primarily due to its modularity and distribution methods. Similar to the Zeus banking trojan, DanaBot is known for its plug-and-play modules, which can drastically alter tactics and priorities. DanaBot has been linked to both Zeus and Gozi, two of the original banking trojans, though it is not formally part of either family. Ever since DanaBot emerged on the banking trojan scene, it has been a heavy hitter, causing significant of damage wherever it goes.

Like most of the other notable banking trojans, DanaBot continues to shift tactics and evolve in order to stay relevant. F5 malware researchers first noticed these shifting tactics in September 2019, however, it is possible they began even earlier.

- As of September 2019, DanaBot shifted its focus solely from financial services targets to include attacks on ecommerce platforms and social media sites. DanaBot has not left its banking trojan roots behind but has expanded its focus to these new targets.
- Along with adding new targets, DanaBot was seen utilizing a ransomware module, which may also indicate a change in priorities.
- To conduct these attacks, DanaBot is using two new methods for theft.
    - The first method is creating fake forms on popular websites, previously seen targeted by other high-profile banking trojans using the JavaScript Tables library, where users are prompted for credit card details. This is executed with HTML and JavaScript that originates from an external source injected to the page.
    - The second method involves using a malicious iframe and abusing the p.a.c.k.e.r. framework, which is a legitimate way to compress and obfuscate code in order to create a command and control (CNC) communication mechanism.

We observed these new DanaBot tactics tampering with popular websites such as AliExpress and Groupon. Technical details follow these images showing what users see.




## Technical Breakdown: Malicious Tables and Forms

DanaBot, like other heavy hitting banking trojans such as Zeus and Gozi, is known for its web injections, the primary way it steals credentials and money from its victims. Researchers were able to see into the DanaBot server in order to begin analyzing some of the tailor-made webinjects. This is where they are stored (see Figure 1) before the selected webinject is injected into a target when the user navigates to a particular site.


Figure 1. Screen capture from within the malicious DanaBot server where all of the tailor-made webinjects are located

Before the malicious forms pop up on a user's screen, the malicious JS tables library starts its work by checking to see if the victim is logged into a specific website on the DanaBot target list. The user's operating system must already be infected with DanaBot. If so, the malware is able to check whether a user is logged into a website on the target list by validating the login with a simple HTML element search of unique identifiers of a user.

```
_tables.start = function () {
    _tables.set('logout', function () {
        if (_tables.id('sign-out')) {
            return true;
        }
        return false;
    });
```

Figure 2. Code that checks to see if the HTML attribute id "sign-out" is in the page. If it is, DanaBot starts its malicious operation.

These operations are notable because they show the time and attention put into crafting this malware. This allows for a very targeted sniper-like attack only of users logged into specific sites. Others may have no idea DanaBot is even running on their machines.

```
if (_tables.findout(document, 'span', 'class:welcome\-title')) {
    return true;
}
return false;
```

Figure 3. Code that check if "span" element that has the class attribute that contains "welcome" and "-title" exists. If it does DanaBot creates the fake popup form.

As shown in figures 2 and 3, this simple but accurate check is an elegant solution to see if a user is logged in. The validation is unique to each target site. Once the code validates that this HTML element exist in the page, the next step of the fraud malware executes.

The malware uses the Tables JavaScript library to create fake payment request forms where users input information. In the past, the JS Tables library was used by high profile banking trojan malware operations, including Zeus and Ursnif/Gozi.

This client-side logic includes some useful utility methods, including:

- A check to see if the email or a date is valid (day, month, year), right after the user inserts those inputs into the fake message.
- A decoder method for HTML entities.

- An event attribute attacher whose purpose is to disable the enter key for form submit (forcing the victim to use the fake form with the malicious event button)
- A replacement submit buttons with the malicious logic of the fraudsters
- A JavaScript tool for URL encoding/decoding
- A validation to see if JS objects and variables are null.

```
check_day : function (dd) {
    if (parseFloat(dd) > 0 && parseFloat(dd) < 32 && (dd + '').length == 2) {
        return true;
    } else {
        return false;
    }
},

check_month : function (mm) {
    if (parseFloat(mm) > 0 && parseFloat(mm) < 13 && (mm + '').length == 2) {
        return true;
    } else {
        return false;
    }
},

check_email : function (email) {
    var re = /^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.
    return re.test(email);
},

check_year : function (yy, format) {
    switch (format) {
    case ('YY'):
        if (parseFloat(yy) >= 15 && (yy + '').length == 2) {
            return true;
        } else {
            return false;
        }
        break;
```

Figure 4. A utility function checking the date

```
input : function (input, type) {
    switch (type) {
    case ("block"):
        if (input) {
            input.onkeup = function (evt) {
                var evt = (evt) ? evt : ((event) ? event : null);
                var node = (evt.target) ? evt.target : ((evt.srcElement) ? evt.srcElement : null);
                if (evt.keyCode == 13) {
                    if (evt.stopPropagation) {
                        evt.stopPropagation();
                    } else {
                        evt.cancelBubble = true;
                    }
                    return false;
                }
            };
```

Figure 5. The utility function for disabling the enter key via the keyboard so the malicious “click” event will replace it

```
replacebutton : function (button, func) {
    var newButton = document.createElement(/image/igm.test(button.tagName) ? 'img' : button.tagName);
    for (x in button.attributes) {
        if (notnull(button.attributes[x]) && notnull(button.attributes[x].name) && notnull(button.attributes[x]
            if (button.attributes[x].name == "onclick" ||
                button.attributes[x].name == "name" ||
                button.attributes[x].name == "disabled" ||
                (button.attributes[x].name == "href" && !/image/igm.test(button.tagName)) ||
                button.attributes[x].name == "id") {
                continue;
            }
            if (button.attributes[x].name == "type" && button.attributes[x].value == "submit") {
                newButton.setAttribute('type', 'button');
            } else {
                newButton.setAttribute(button.attributes[x].name, button.attributes[x].value);
            }
```

Figure 6. The utility method showing a replacement of a legitimate button with a fake one the includes the fraudsters logic

The Tables JavaScript library and utility methods are used together to create fake forms where users enter information. F5 researchers have noticed these targeting popular ecommerce sites with a few examples shown in this article. Before DanaBot in September 2019, this tactic had not been seen. By forcing victims to input credit card information into these fake web forms, DanaBot is overtly collecting payment information. These forms are not intuitive for users to escape out of. Victims often choose the path of least resistance on a website they believe is legitimate, so they enter the requested information.

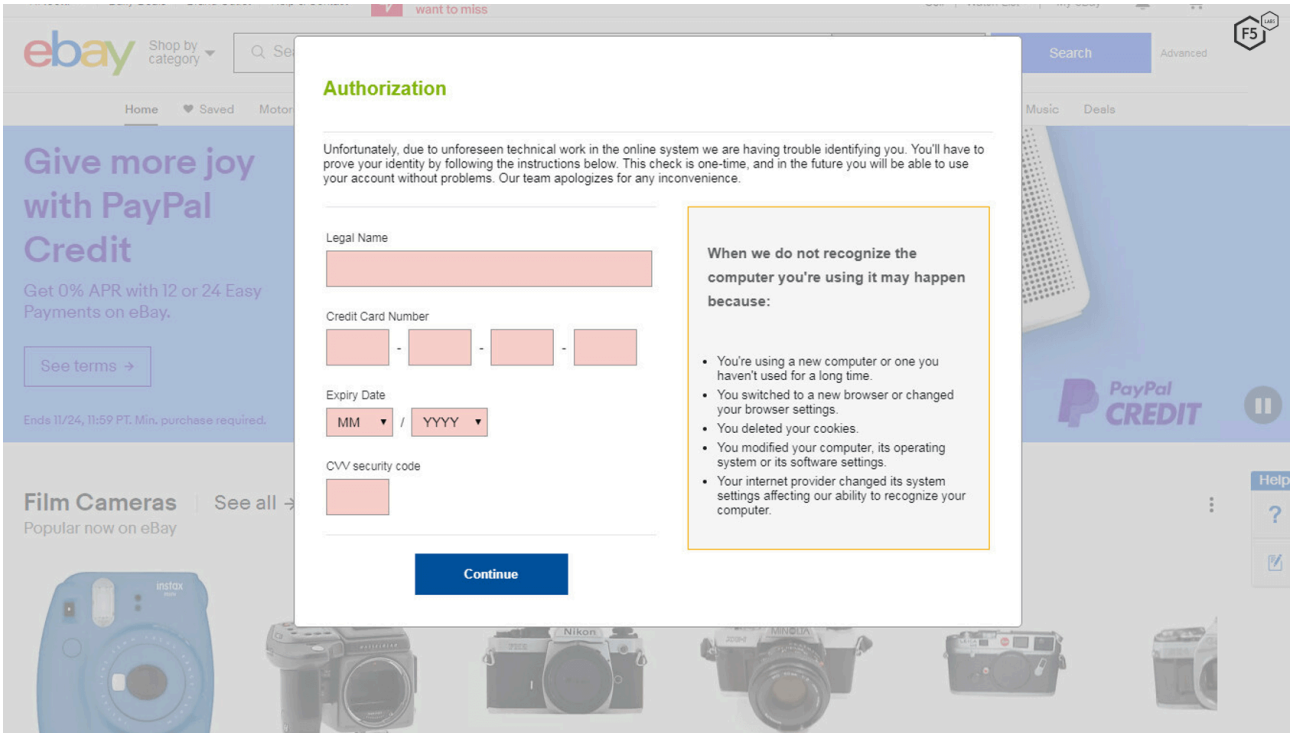



Figure 7. A malicious form DanaBot puts into eBay in order to capture financial information

This tactic is similar to other banking trojan tactics, where victims think they are entering credential or other sensitive information into websites they don't know are spoofed. For these, the entire website is not spoofed, but the addition of this form on top of a legitimate page increases the likelihood that victims will enter personal data. These examples (eBay, AliExpress, and Groupon) are significant, not only because they show a new tactic used by

DanaBot, but also because they show a shift from targeting mostly financial services institutions to popular ecommerce platforms, where users are accustomed to entering payment information.

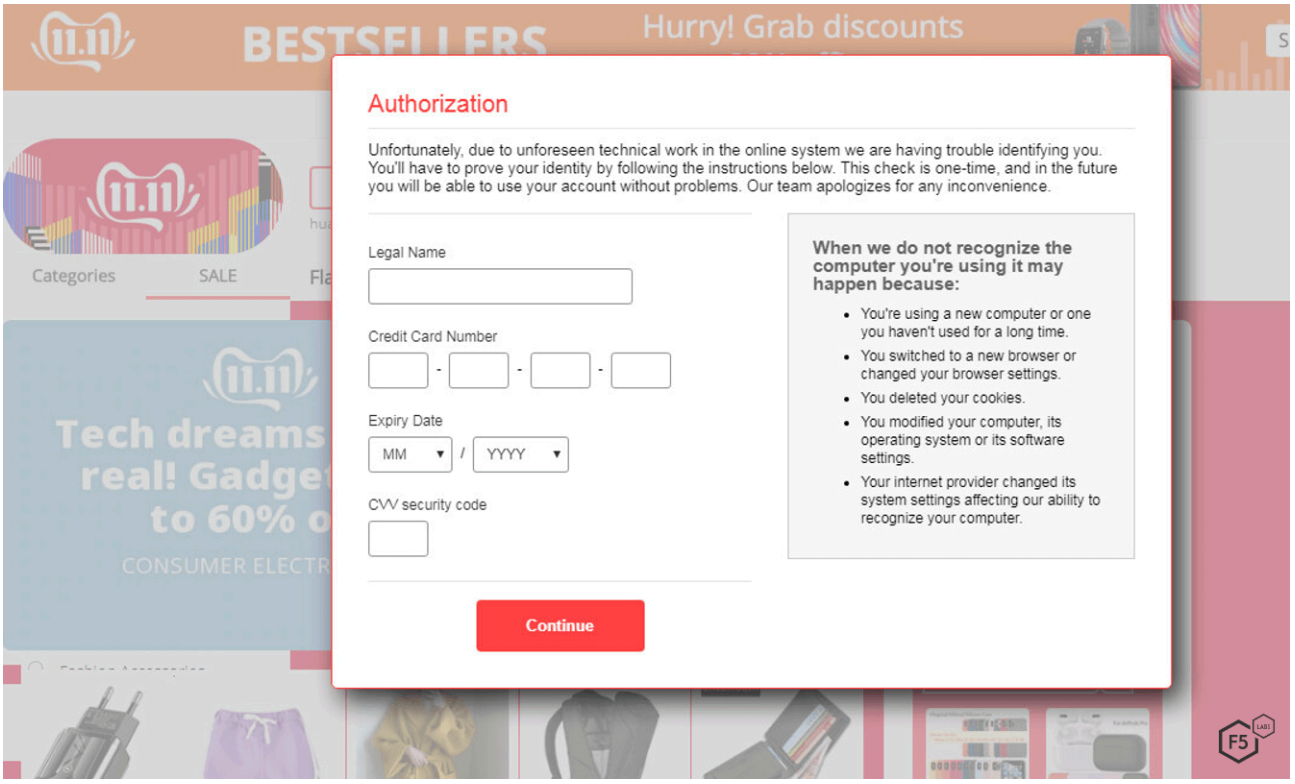



Figure 8. The malicious form on AliExpress, a popular ecommerce platform

This latest DanaBot campaign is also global, given that American websites such as Groupon and eBay are targets (see figures 7 and 8) as well as AliExpress (see figure 9), a popular global ecommerce platform from China. This further demonstrates that cybercrime is not confined to any national borders, so users must always remain vigilant, no matter where in the world they reside.

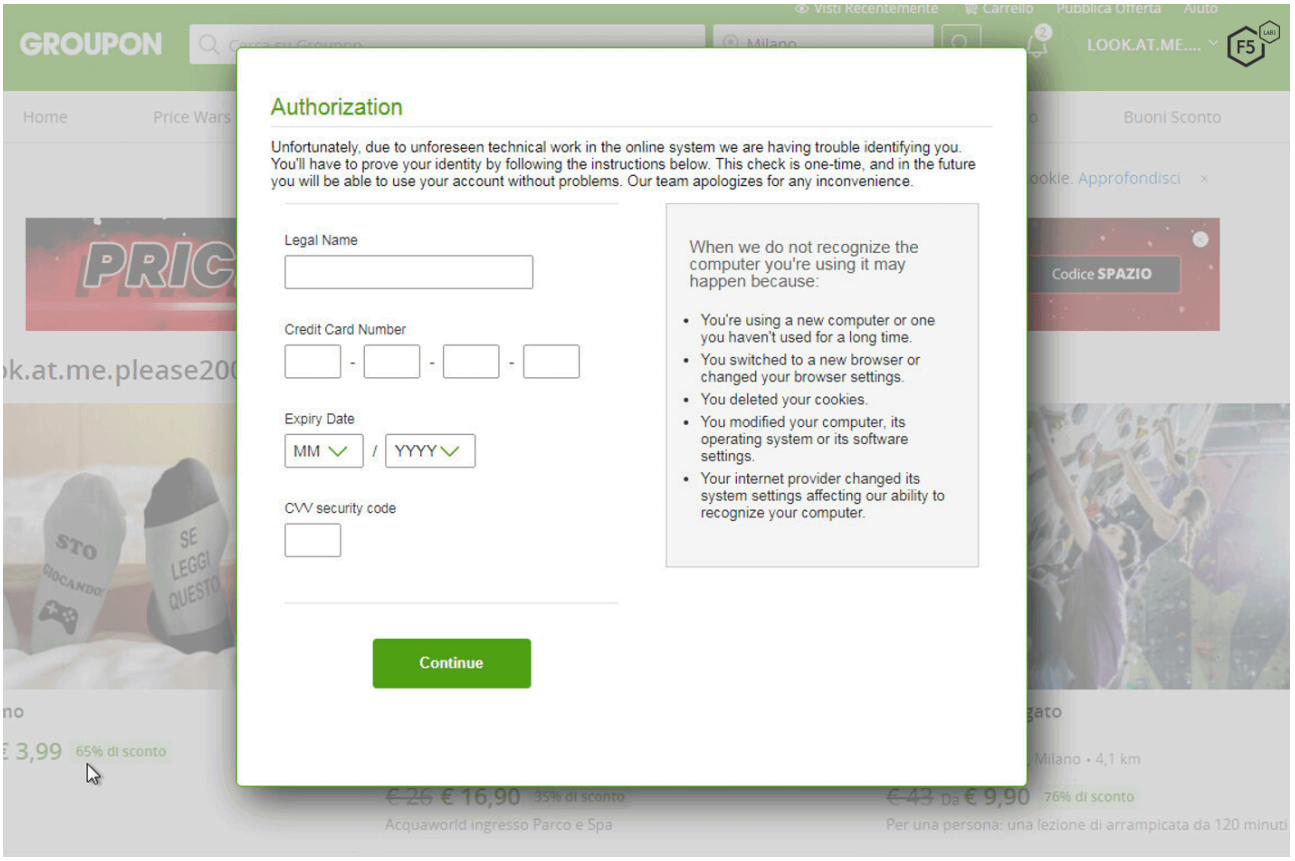



Figure 9: A malicious form on Groupon, a popular ecommerce platform

Along with ecommerce targets, DanaBot is expanding to social media and streaming websites. This includes Twitch, the world's leading live steaming platform for gamers. Users can watch and chat with others online, and there are opportunities for them to enter their credit card details to purchase Twitch Prime or to support specific channels. As such, DanaBot takes advantage of this and uses the same web injection technique to create a malicious table as used in these other examples.

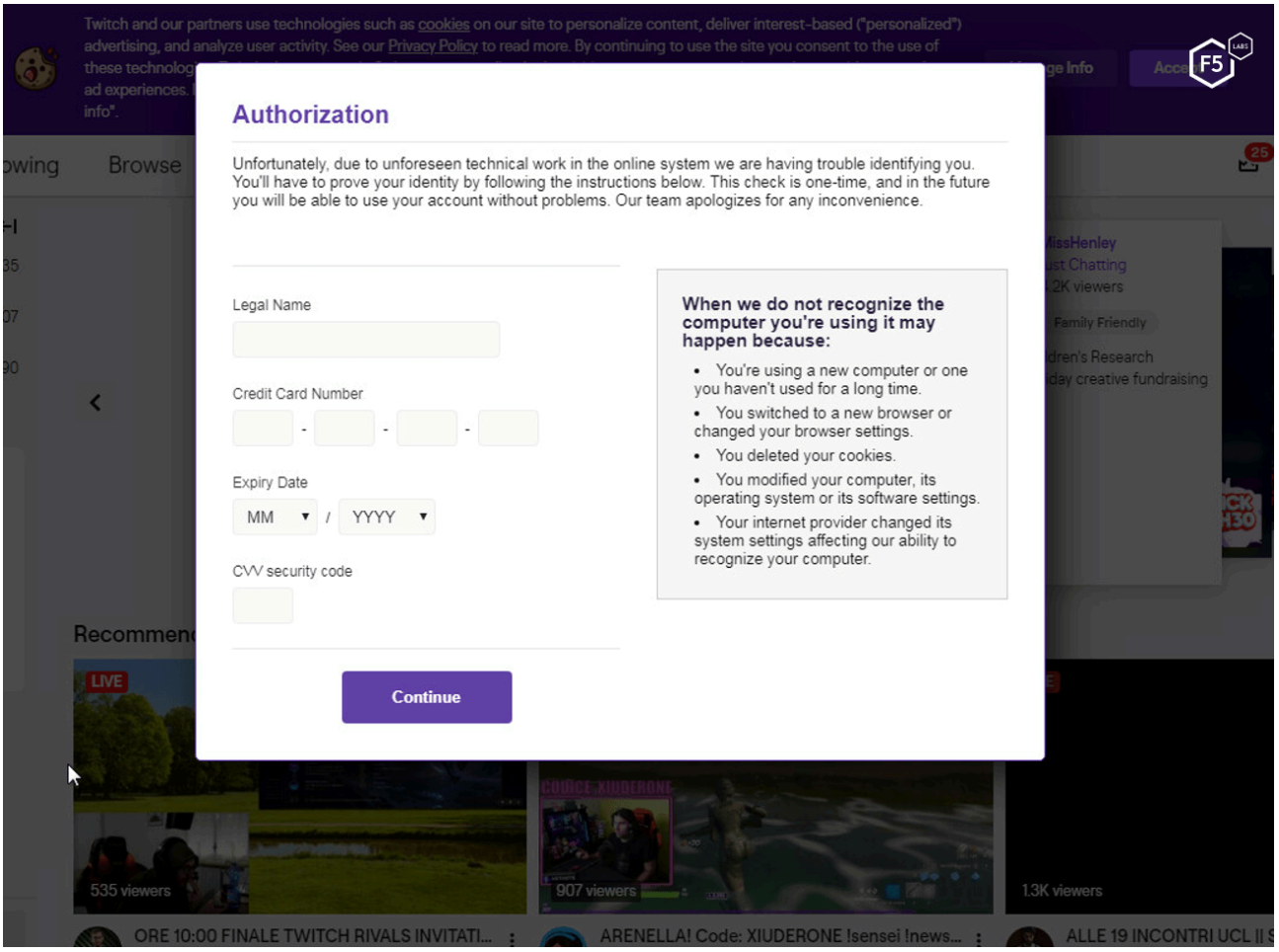



Figure 10. Malicious table on Twich, stealing credit card information

Users of these platforms in particular need to remain vigilant. If forms like these pop up, they should immediately close the window and attempt to use a clean computer to access the site. There is also an opportunity for the operators of these popular ecommerce and streaming platforms to educate users before so they don't lose personal or critical information.

## Technical Breakdown: Obfuscation and Using iFrames

Along with this new tactic used to steal users' financial information, DanaBot was also spotted as early as September 2019 abusing the p.a.c.k.e.r. framework, which is a legitimate way to compress and obfuscate code in order to create a command and control (CNC) mechanism. Using Dean Edwards' p.a.c.k.e.r compressor as the first step, DanaBot dynamically creates the second stage of the injection.[2] These two new techniques can be used together in order to trick users into entering sensitive financial information which is then communicated back to a CNC server.

Along with the legitimate p.a.c.k.e.r. compresser, the JavaScript "eval" function is used. This function is known to be vulnerable because it does not conduct any input validation and will execute anything that's passed to it. The "evil" eval function[3] takes as an argument a decompressed string, which is the output from using p.a.c.k.e.r. A script is then created that checks to see if the victim is a logged in user of that website.

After that script is created, the malware uses a malicious iframe that sends messages and receives responses. This is done using the postMessage mechanism, which enables communication with its parent window, the website itself. The script then receives messages that are being read with the first script that was created using the eval function.

The full flow of this malicious activity is as follows:

The attacker uses the p.a.c.k.e.r compressor to dynamically create a function named NCCVBVGrabLoader, which checks to see if a victim is logged in and controls the communication with the iframe (its id attribute is “pmiframe”). This is also appended by the NCCVBVGrabLoader script logic. The NCCVBVGrabLoader inputs the response from the iframe’s (id=pmiframe), which it gets from the server in order to become a script that generates the next stage.

The response is crafted using the top.postmessage functionality[4] that contains the eval function which triggers and creates the logic that injects fake html to the target website. This becomes the new controller of communication with the appended iframe.

Finally, a script is run that received in this mechanism is the list of countries and the language the payment form will be filled with.

A full gif showing what the user sees when logging in and getting the malicious pop-up is shown in figure 11.

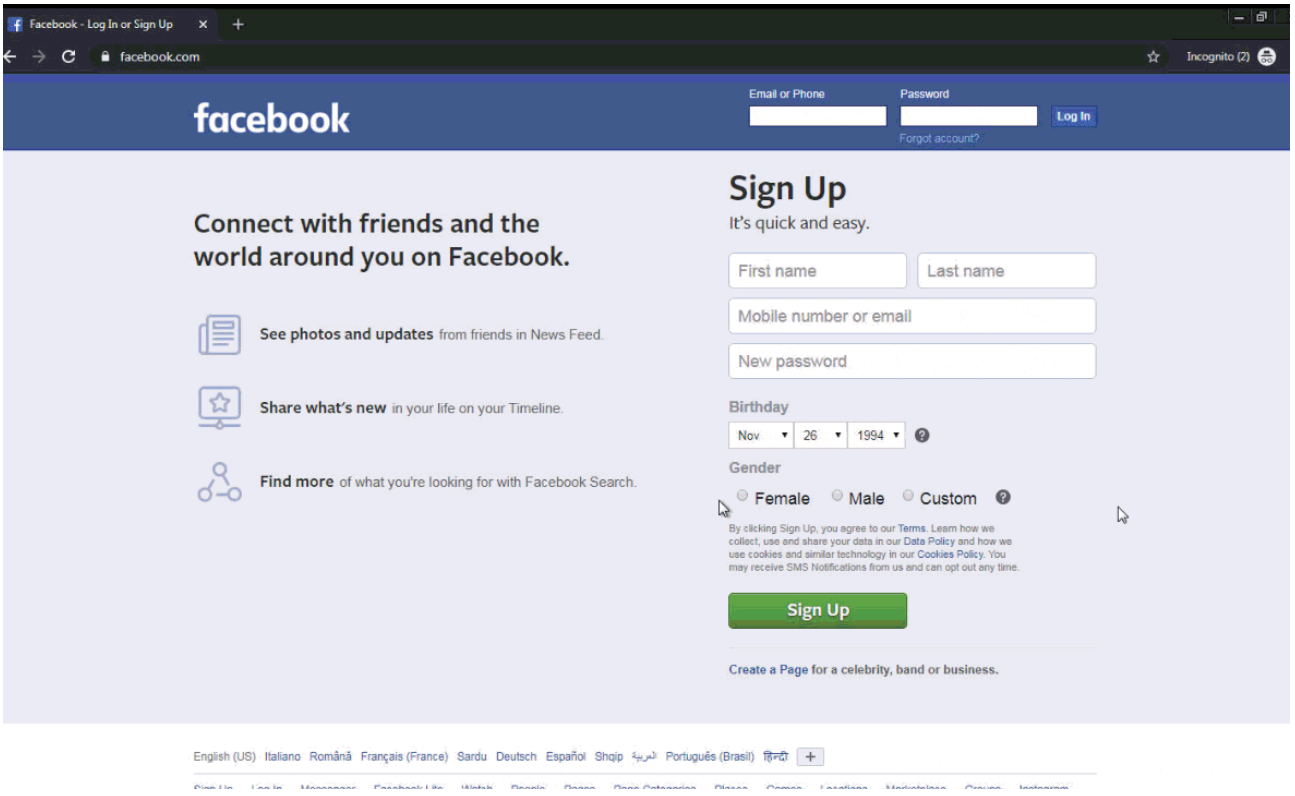



Figure 11. gif showing the user experience when this malware executes

## DanaBot Attacker Infrastructure

After utilizing either the new tactics discussed in previous sections or other webinject modules, the attacker is able to retrieve the victim's sensitive information via the CNC . Both tables and the VBV mechanism that use p.a.c.k.e.r, have their own CNC server with a dedicated panel.

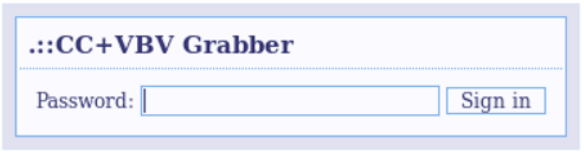



Figure 12. The login page for the vbv grabber

F5 researchers were able to log in through both of these login pages (see figures 12 and 13) and see the same data that DanaBot attackers see and use while conducting their fraudulent operations.

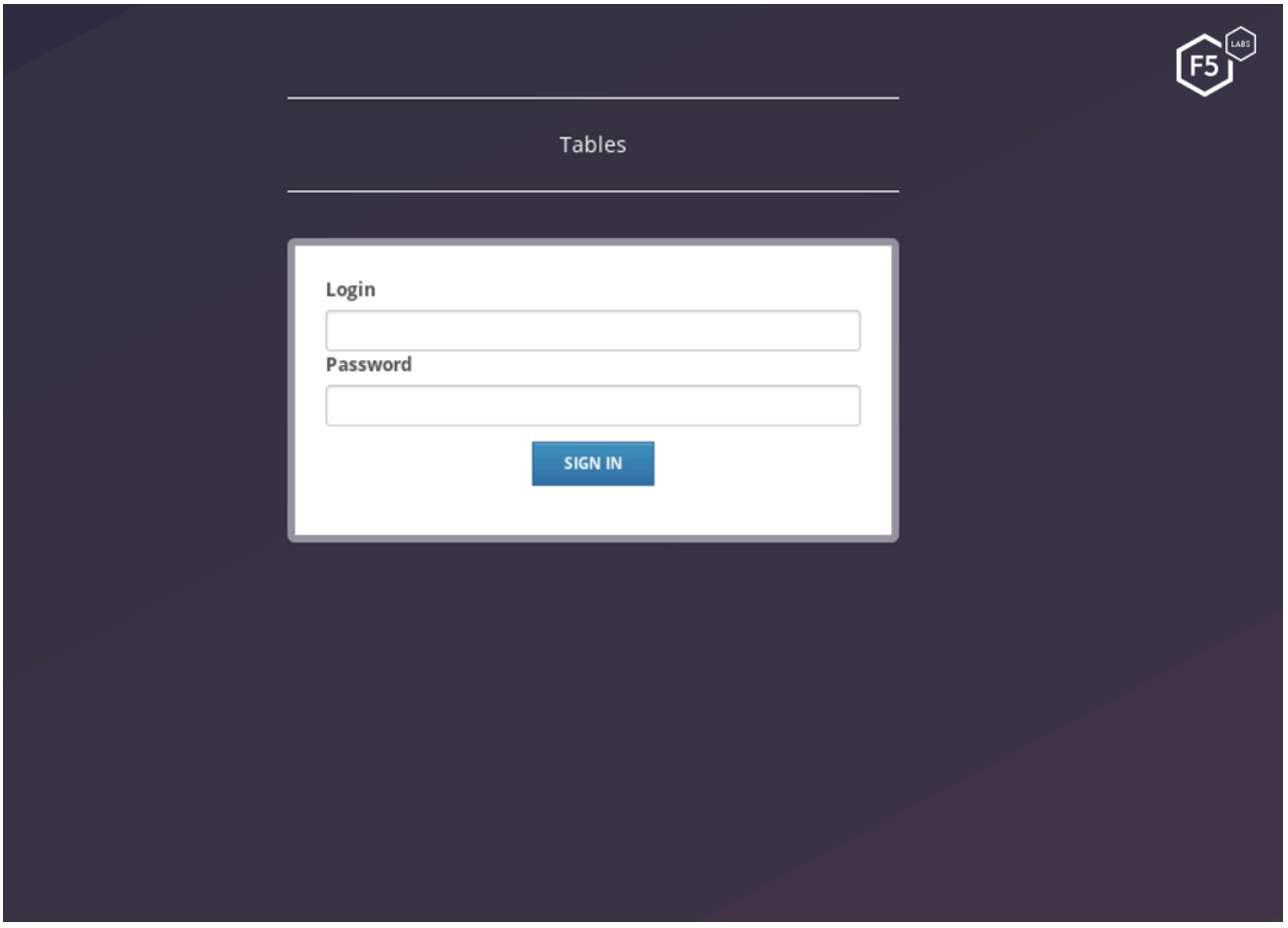



Figure 13. The Tables login page

Once logged in to the DanaBot attacker control panel, researchers were able to see the actual victim data as well as the browser BOTID. This data is blurred in figure 14 as it is sensitive information.

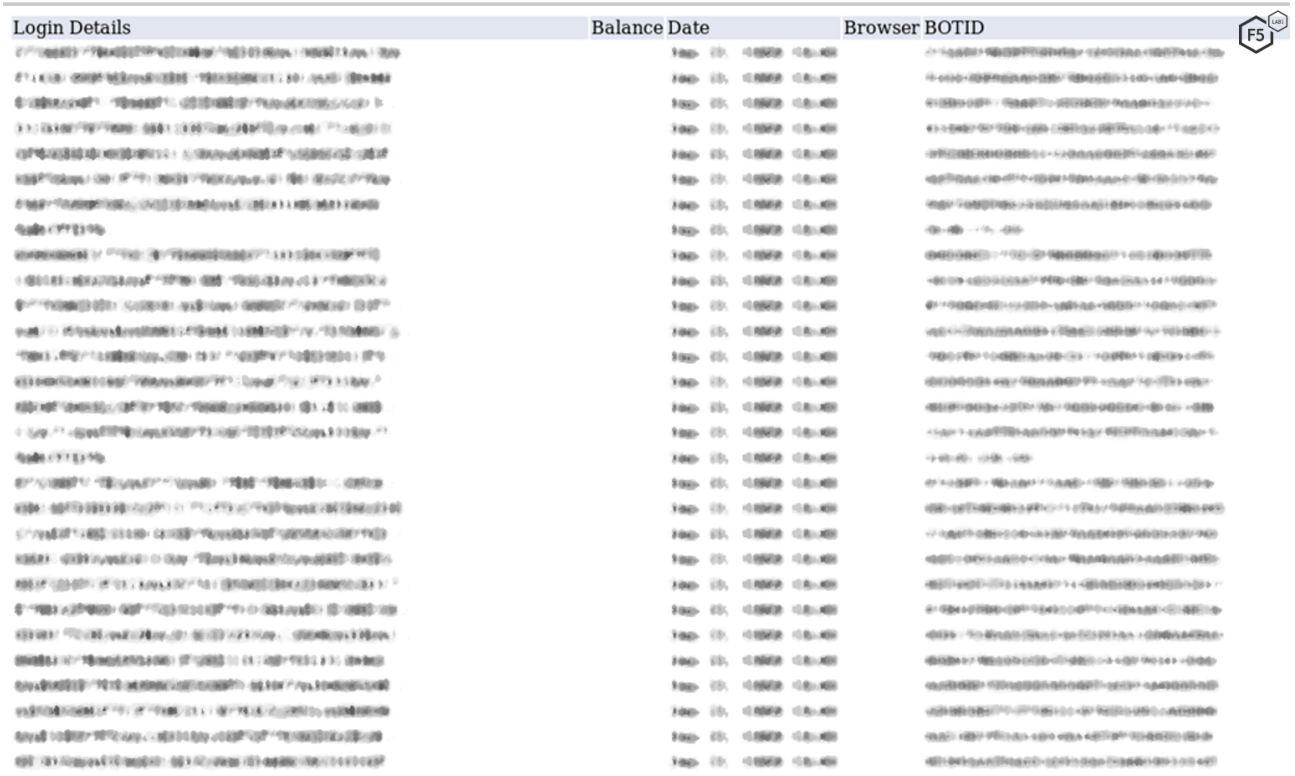



Figure 14. Actual data sent from the victim's browser to the DanaBot attacker server

Completely investigating the underlying server architecture and CNC structure of banking trojans such as DanaBot is an area of continuing research for the F5 malware team.

## Conclusion

As with all banking trojans, DanaBot actively updates its tactics, techniques, and target list to both avoid detection and maintain continual operations to optimize the attacker’s financial reward. We are not surprised to see new tactics emerge in preparation for the peak (end-of-year) holiday phishing and fraud season. What is notable, along with these new tactics, is the shift in targets from solely financial institutions to ecommerce and streaming services.

While we do not know who is maintaining this malware, the malware has been linked to Gozi and Tinba in the past, which use the same injection patterns. In addition, DanaBot has been said to be a “Zeus-like” piece of malware. Given this progression and the successful tactics used, we predict that DanaBot will continue to be a major player in the banking trojan world for the rest of 2019 into 2020.

All organizations, especially the known targets identified in this article, should make their customers aware that they are being targeted. The main purpose of these campaigns is to plant malware on their machines that is designed to steal credit card information, money, and gain access to sensitive information that may be sent back to the malware authors and used for future exploitation.

To combat the impact of fraudulent transactions occurring as the result of malware-infected customer machines, organizations should implement fraud detections within their web platforms that can detect banking trojans and

block resulting fraudulent transactions. For more details on how to combat phishing attacks that lead to fraud, see F5 Labs' 2019 Phishing and Fraud Report.

## Security Controls

Enterprises should consider implementing the following security controls (/content/f5-labs-v2/en/archive-pages/education/what-are-security-controls.html) based on their specific circumstances:

---