

## Recent Posts

 [threatresearch.ext.hp.com/tips-for-automating-ioc-extraction-from-gootloader-a-changing-javascript-malware/](https://threatresearch.ext.hp.com/tips-for-automating-ioc-extraction-from-gootloader-a-changing-javascript-malware/)

May 4, 2022

[HP Threat Research Blog](#) • Tips for Automating IOC Extraction from GootLoader, a Changing JavaScript Malware



### Tips for Automating IOC Extraction from GootLoader, a Changing JavaScript Malware

The threat actors behind GootLoader are always making adjustments to this family of JavaScript malware, which affects indicator of compromise (IOC) extraction using [our decoder script](#). Whenever the GootLoader decoder breaks we try to adapt it to the new version of the malware to help the security community. In this post, we share the process of debugging and fixing the script, showing the common steps we usually take.

First, we look at all extracts from the regular expressions (regex) because this is usually the reason why the decoder breaks. To do so, we add print statements before and after each regex evaluation starting from the bottom of the script. The following image shows our first check.

```
else:
    # New GootLoader Version 2022-04
    content = content.replace("\")+("\", "").replace("\")+("\", "").replace("\")+("\", "").replace("\")+("\", "").replace("\")+("\", "")
    print(content)
    domains = re.findall(breacket_regex, content.split(";")[0], re.MULTILINE)
    print(domains)
```





substitute the expression based on the values in our dictionary and get the final expression consisting of variables that refer to code fragments. The adjusted code sequence looks like this:

```
print(clean_content)
matches = re.findall(code_order.replace("NUM_REP", str(len(code_parts)-1)), clean_content.replace(" ", ""), re.MULTILINE)
print(matches)
order = list()
if len(matches) > 0:
    for m in matches:
        order = m.split("+")
else:
    # New GootLoader Version 2022-05
    code_fragments = dict()
    result_element = ""
    matches = re.findall(re_code_order, clean_content.replace(" ", ""), re.MULTILINE)
    for expr in matches:
        stmt = expr.replace(" ", "").split("=")
        code_fragments[stmt[0]] = stmt[1].split("+")
        result_element = code_fragments[stmt[0]]

    for element in result_element:
        order += code_fragments[element]
```

Figure 8 – Modified code sequence to decode the new GootLoader version.

To keep the script still compatible for older GootLoader versions, we add an If statement. If we do not get regex matches for the simple statement, we use the code which evaluates compound statements. Finally, we remove the print statements which we inserted for debugging and run the repaired script to get the following output:

```
:/tmp/gootloader$ python3 decode.py -d 20220502/
OK - 20220502/loader_zip.js
Found URLs: (3)
> Wrote file: urls.txt
Found Domains: (3)
> Wrote file: domains.txt
:/tmp/gootloader$ cat domains.txt
lakeside-fishandchips.com
kristinee.com
learn.openschool.ua
:/tmp/gootloader$ cat urls.txt
https://lakeside-fishandchips.com/test.php?sgjngbizjfwgs=
https://learn.openschool.ua/test.php?sgjngbizjfwgs=
https://kristinee.com/test.php?sgjngbizjfwgs=
```

Figure 9 – GootLoader decoding script output.

This is the typical process we go through to adapt the decoding script to new GootLoader versions. As the threat actors behind GootLoader have been making version changes more frequently lately, we have had to make changes to our script more frequently as well. We hope that this explanation is helpful for analyzing GootLoader and making adjustments of our decoder script.

## Tools

---

You can download the latest GootLoader decoder script here:

<https://github.com/hpthreatresearch/tools/blob/main/gootloader/decode.py>

## IOCs

---

New GootLoader version:

0a7c07fc84fd9f5b91bde6822b865f9647ca4ece67e8a4a646ce8d405187dc8b

hxxps://lakeside-fishandchips[.]com/test.php?sgjngbizjfwgs=

hxxps://learn.openschool[.]ua/test.php?sgjngbizjfwgs=

hxxps://kristinee[.]com/test.php?sgjngbizjfwgs=

Older GootLoader version:

765fbca3b6b1a922b442bc7304454e752e8bf231e2abe5060ace55db72c78d68

hxxps://kristinee[.]com/test.php?zemyrwgzcsnjur=

hxxps://kep[.]org/test.php?zemyrwgzcsnjur=

hxxps://korsakovmusic[.]com/test.php?zemyrwgzcsnjur=

Tags

[automate](#) [gootloader](#) [ioc](#)