Threat Hunting

written by Mert SARICA | 1 August 2017

Sometimes, after writing a blog post about a malware, I find myself asking,
"How would I detect this if I were in that situation?" and unintentionally a
process begins in the background, with this question lingering and waiting to
be answered. Once this process is completed and the question is answered, a
new blog post emerges, as seen in Figure 1-A. In this current article you are
reading, I also sought an answer to the question, "If these malicious
individuals are targeting government websites and injecting malicious
JavaScript code into the pages, how difficult can it be to detect this in
practice?", following the December 2016 blog post titled "They PWN Houses!"

As a first step, I tried to access the domain names of our government websites (with the .gov.tr extension) through search engine APIs such as Google and Bing, but I was unsuccessful due to their existing limitations. While desperately daydreaming about having access to DNS requests made to the OpenDNS service, so that I could extract the list from there, the idea of Roksit, the counterpart of OpenDNS, came to mind. I decided to contact them and ask for support regarding my security research on this matter. Thankfully, once they understood my good intentions, they shared with me a list of government domain names (~8000 in total), although not complete, which I could practically implement the idea in my mind.

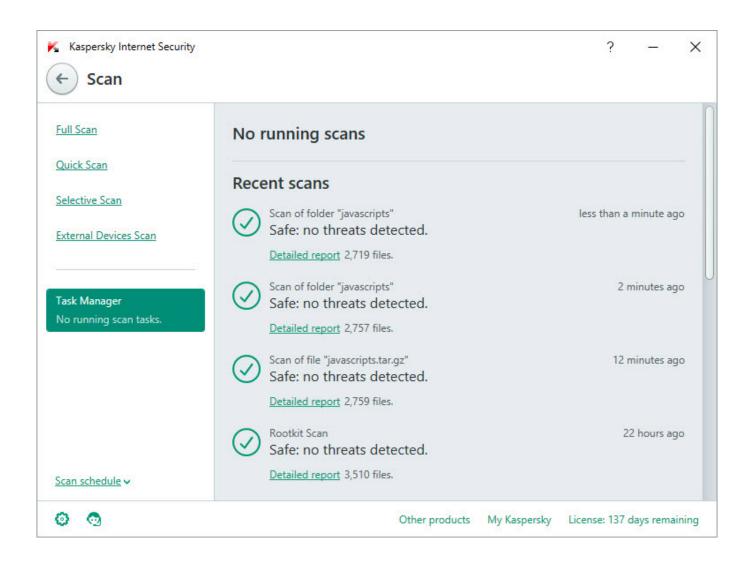
After obtaining the list, without wasting any time, I quickly developed a simple tool called JavaScript Crawler using Python, which crawls through all the websites and detects JavaScript code injected (imported) into the homepage via the current site or any other web address. It saves the detected JavaScript code along with the corresponding web addresses to the disk. Shortly after running this tool, I created a script that downloads all the identified JavaScript files from their respective addresses.

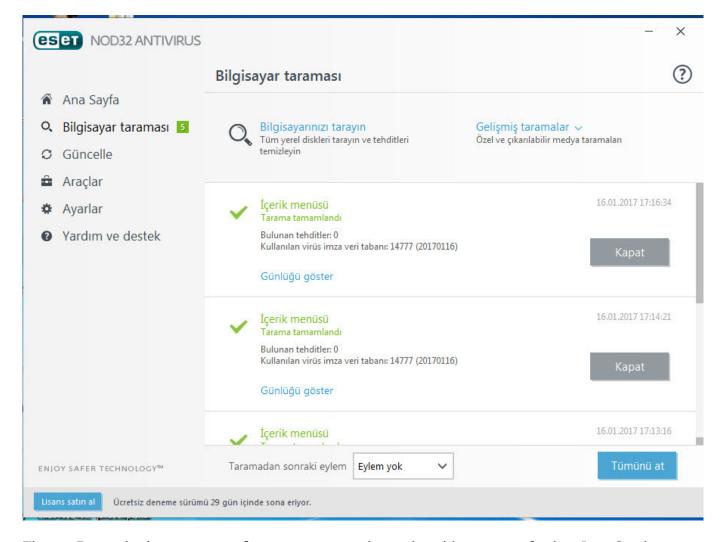
Once the JavaScript files were downloaded, I scanned them using security software such as ClamAV, ESET NOD32, and Kaspersky Internet Security Suite. Fortunately, I did not come across any malicious files during the scanning process.

```
JavaScript Crawler v1.0 [https://www.mertsarica.com]

[c] Crawling...
[c] Cromecting to: http://atam.gov.tr
[c] 1. Script tag: http://ajax.gogleapis.com/ajax/libs/jquery/l/jquery.min.js
[c] 1. Script tag: http://www.atam.gov.tr/wp-cortent/themes/v1/js/silder.js
[c] 1. Script tag: http://www.atam.gov.tr/wp-cortent/themes/v1/js/silder.js
[c] 1. Script tag: http://www.atam.gov.tr/wp-cortent/plugins/media-element-htm]5-video-and-audio-player/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/mediaelement/media
```

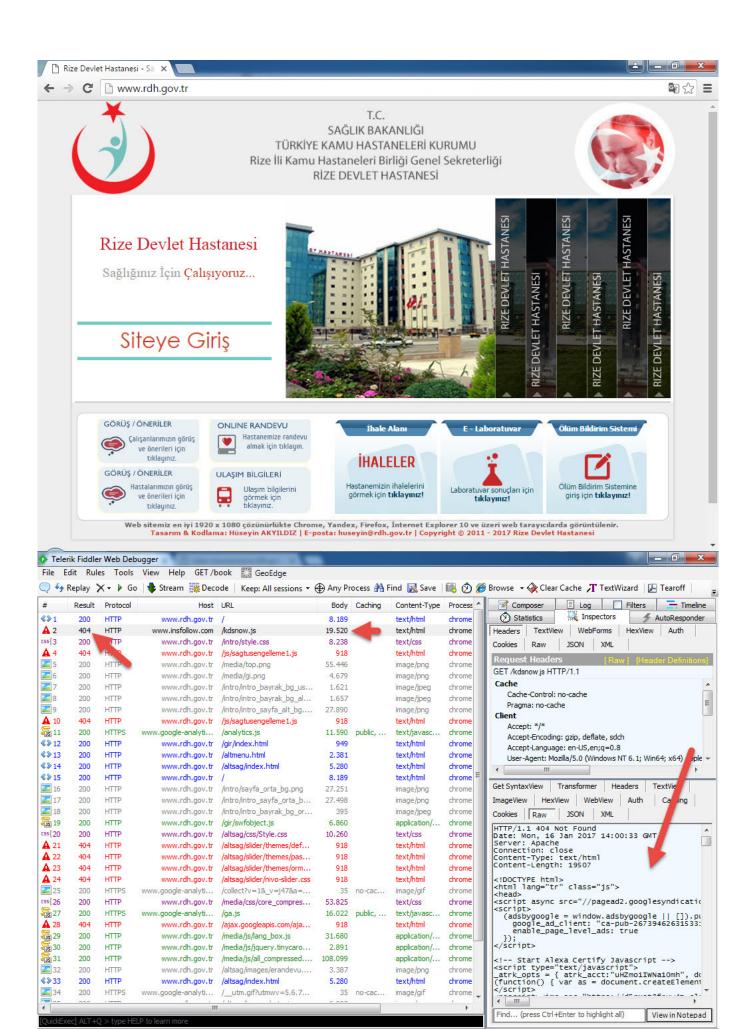
```
| Total | Descripts | Descript
```

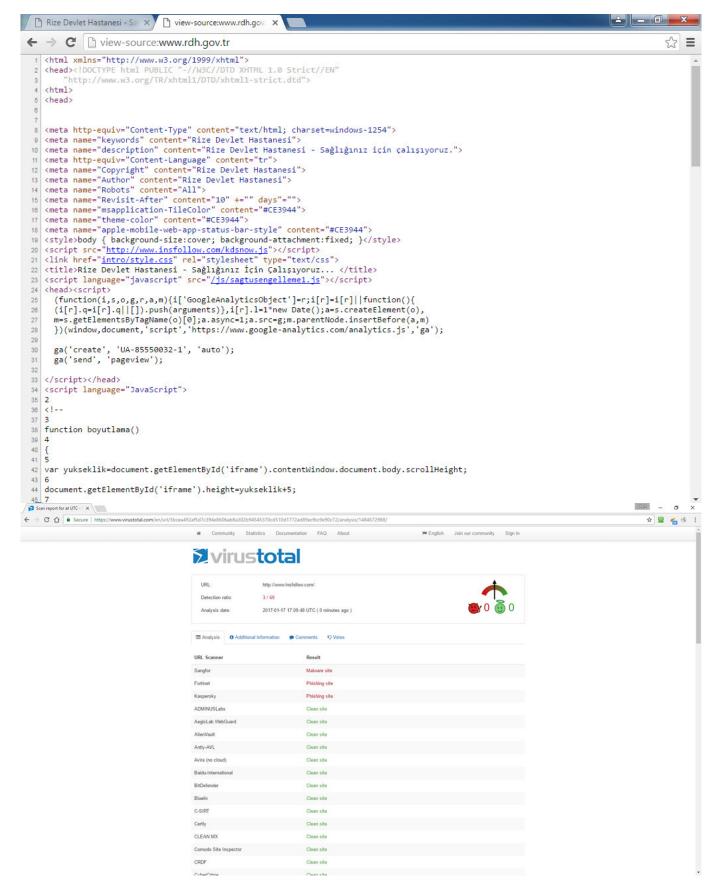




Then, I used the sort tool to arrange the web addresses of the JavaScript files listed in the log file, and filtered out well-known addresses like ajax.googleapis.com. Among the remaining addresses, one domain caught my attention: insfollow.com. When I checked which government website this domain was detected on, I found that it belonged to Rize State Hospital. I visited the website and examined its source code, where I easily identified the insfollow.com domain and the injected JavaScript file.

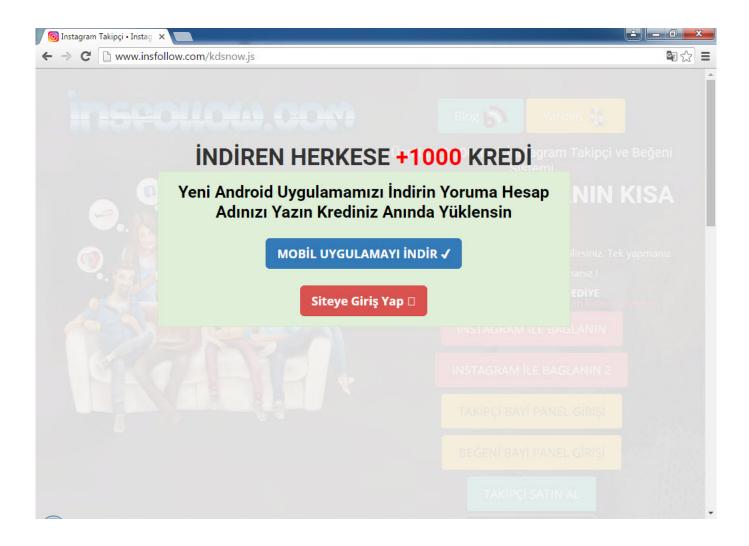
To gather more information, I submitted the insfollow.com address to VirusTotal, and it revealed that three security software detected it as a phishing site.

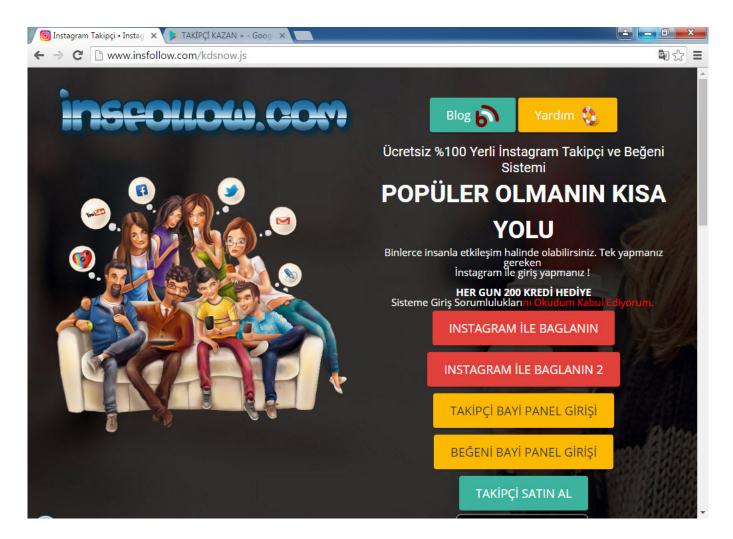




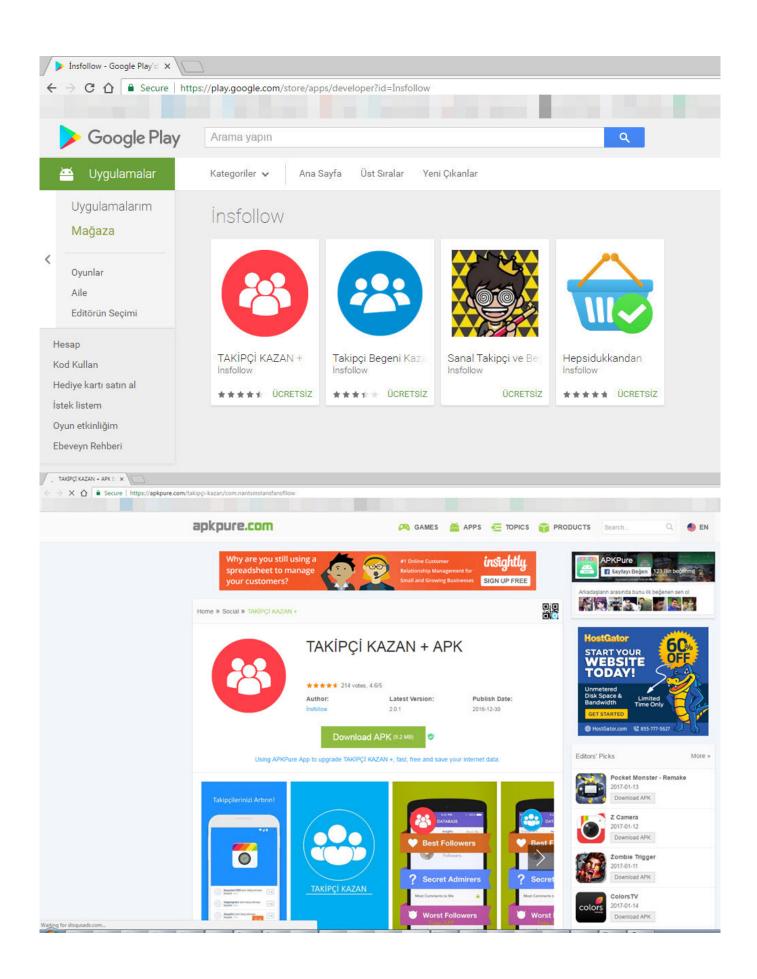
When I visited http://www.insfollow.com, I discovered that it was a website created with the purpose of selling Instagram followers, indicating that it operated under the guise of providing such services. However, based on my previous analysis of malicious websites and JavaScript codes involved in

stealing social media and network passwords (such as Token Thieves and Social Network Thieves), I decided to continue my research.



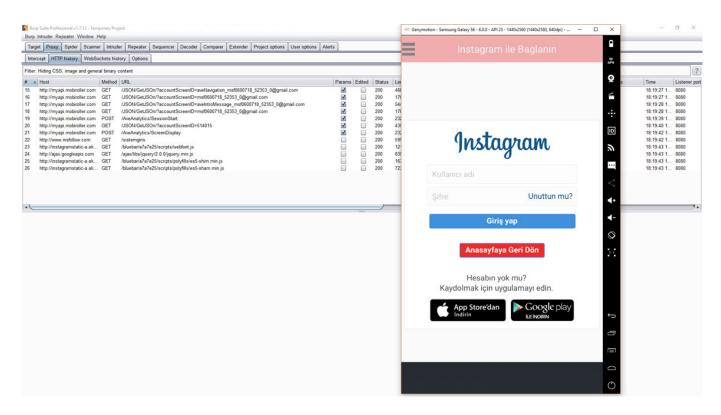


First, I downloaded the advertised "Takipçi Kazan" mobile application from the website and ran it on the Genymotion emulator. In the pop-up message window, it instructed me to log in to the application with an Instagram account. Therefore, I created a new Instagram account specifically for this purpose, knowing that I could safely expose its password.

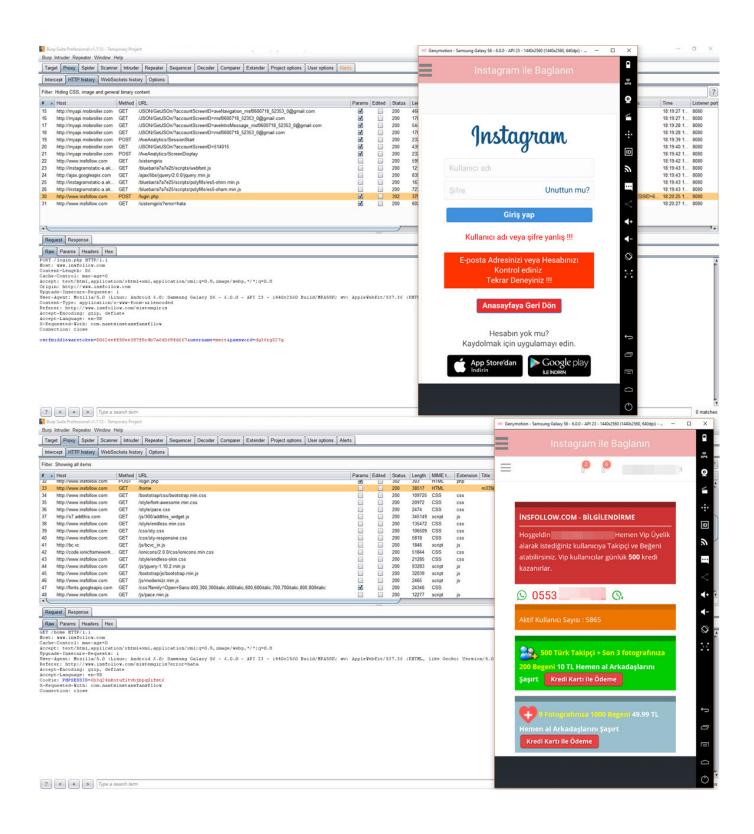


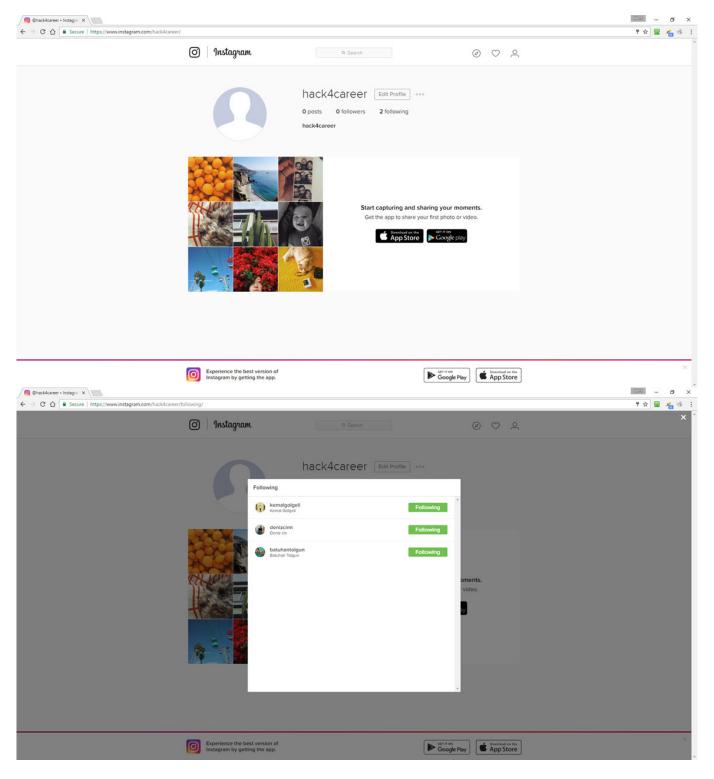


When I ran the application, I discovered that it was developed using Mobiroller, as there were requests being made to the URL http://myapi.mobiroller.com in the background. Upon further inspection of the outgoing requests, I was able to easily see the email addresses of the application developer.



To understand the behavior of the "Takipçi Kazan" application, I first entered my incorrect Instagram password. From the error message "Username or password is incorrect!!!" it was clear that the application was capturing and instantly using the entered username and password on Instagram. After entering the correct password, the application redirected me to its information and payment page. When I logged into my Instagram account afterwards, I noticed a rapid increase in the number of accounts I was following. However, it wasn't long before I was unable to log into my Instagram account, and shortly thereafter, my account was suspended by Instagram.

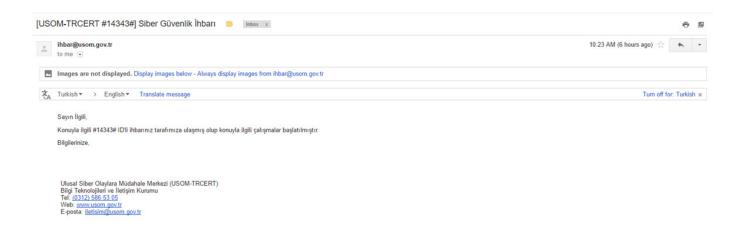




As a result of this research, I have learned that in addition to the organized groups mentioned in the "They PWN Houses!" article, social media and network thieves who create websites under the guise of follower services also target our government websites. I hope that this individual effort sheds light on the authorized institutions responsible for the security of government websites. I would like to remind social media users to be cautious when using websites and mobile applications that promise followers or likes.

Hope to see you in the following articles.

Note: I would like to express my gratitude to USOM (National Cybersecurity Intervention Center) for initiating an investigation based on my report as a responsible citizen.



Note: It has been observed that the malicious code mentioned in the blog post was removed from the website of the hospital during the time between my research and writing/publishing the blog post.