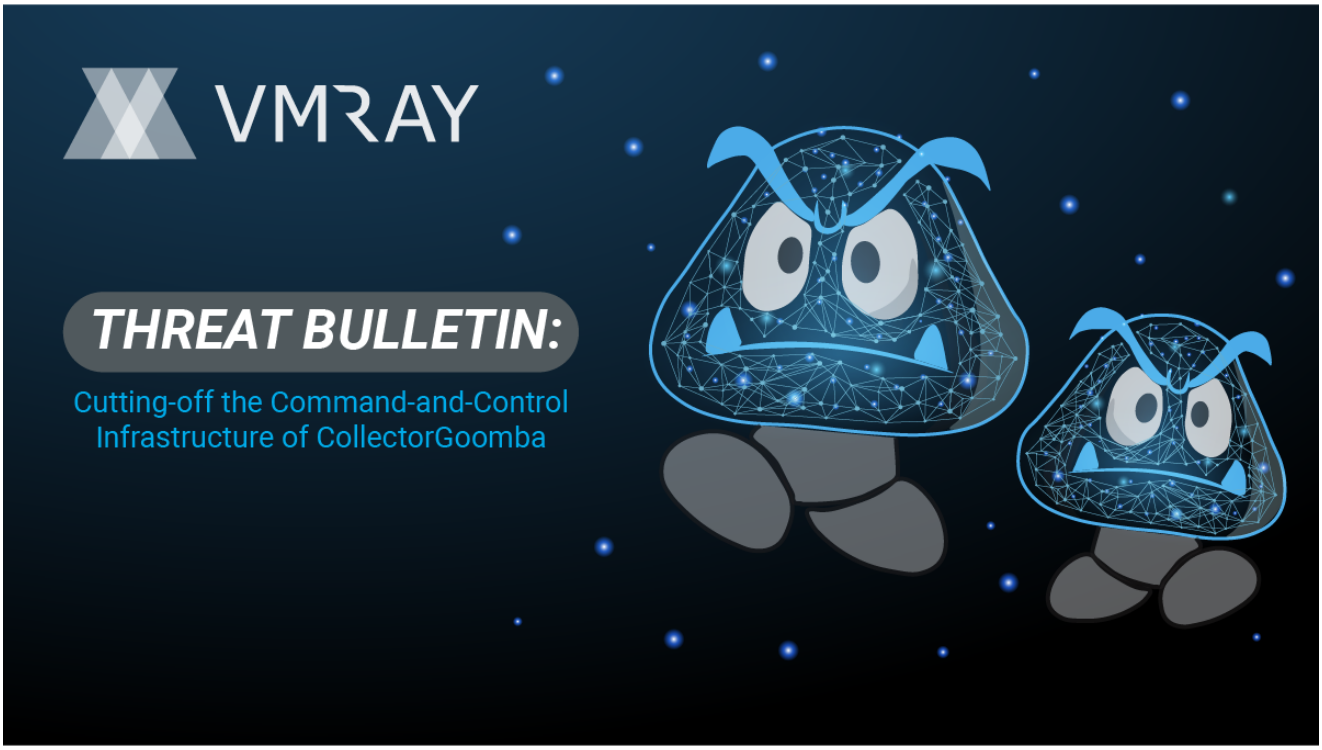
Threat Bulletin: Cutting-off the Command-and-Control Infrastructure of CollectorGoomba

vmray.com/cyber-security-blog/cutting-off-command-and-control-infrastructure-collectorgoomba-threat-bulletin/



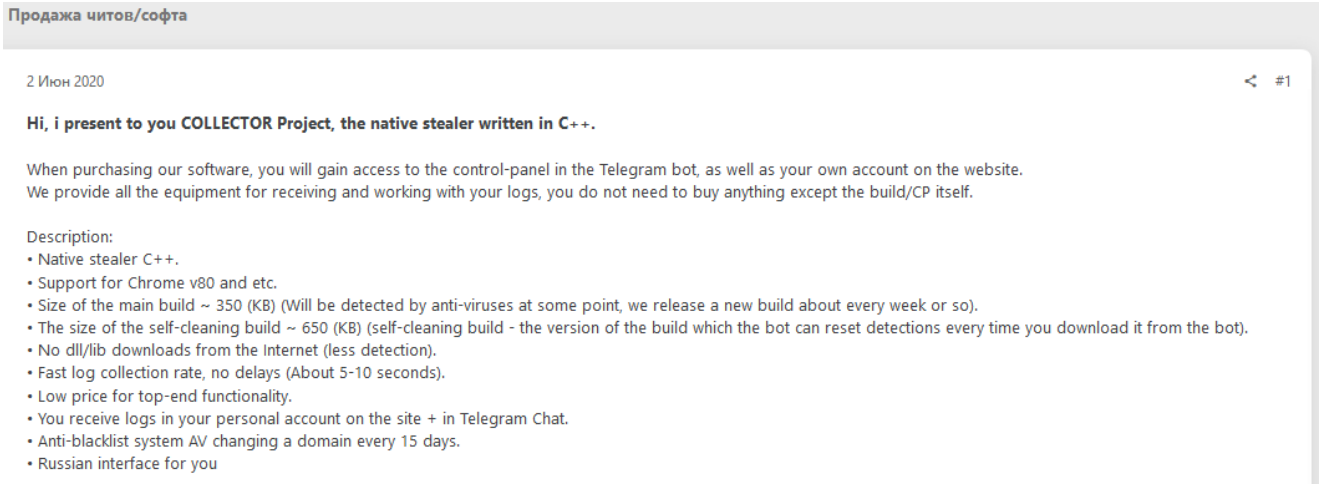
A Primer on Spyware-as-a-Service

The rise in spyware-as-a-service allows cyber-criminals to choose a specialty, whether improving spyware, infecting users, or maximizing the profit derived from stolen information. The business model for spyware-as-a-service starts with an individual or team to developing the initial spyware and standing up any necessary infrastructure that the malware relies upon. The development team can then sell its software to other, less tech-savvy cyber-criminals. Spyware-(or malware)as-a-service is economic specialization and is become more common in the cyber-criminal community – the malicious version of the software as a service in the cloud, but instead of accounting or timekeeping, something much more nefarious.

Network-based defenses are deployed by organizations to detect and prevent spyware communications. Some of these network defenses include network logging and intrusion prevention systems (IPS). The rapid evolution of malware can make portions of these defenses less effective. By detonating malware in VMRay Analyzer, defenders can dissect new malware samples and learn how to modify their defenses to keep their network protected. In this blog post, we will look at a spyware sample sold to criminal attackers. By

analyzing the data dropped during execution of the spyware sample, the VMRay Labs Team was able to identify the name that the spyware calls itself, “COLLECTOR Project”, and find forums where this malicious spyware platform is being sold to criminals.

View the VMRay Analyzer Report for CollectorGoomba


Продажа читов/софта

2 Июн 2020 #1

Hi, i present to you COLLECTOR Project, the native stealer written in C++.

When purchasing our software, you will gain access to the control-panel in the Telegram bot, as well as your own account on the website.
We provide all the equipment for receiving and working with your logs, you do not need to buy anything except the build/CP itself.

Description:
• Native stealer C++.
• Support for Chrome v80 and etc.
• Size of the main build ~ 350 (KB) (Will be detected by anti-viruses at some point, we release a new build about every week or so).
• The size of the self-cleaning build ~ 650 (KB) (self-cleaning build - the version of the build which the bot can reset detections every time you download it from the bot).
• No dll/lib downloads from the Internet (less detection).
• Fast log collection rate, no delays (About 5-10 seconds).
• Low price for top-end functionality.
• You receive logs in your personal account on the site + in Telegram Chat.
• Anti-blacklist system AV changing a domain every 15 days.
• Russian interface for you

Figure 1: Forum post advertising the spyware-as-a-service platform that this sample belongs to

Figure 1 shows an advertisement of the spyware being posted on this forum, it has since been taken down by the forum moderators. The author describes the benefits of using his product for stealing information which allows criminals to easily access the stolen information and supposed periodic changes to the spyware in order to avoid defenses (as a side-note: the history of file modifications to the command-and-control infrastructure suggest that these spyware changes were closer to a month apart than the couple weeks criminals were promised). The author also appears to be claiming that his team wrote this spyware-as-a-service strain.

Data Stolen

During execution, CollectorGoomba (referred to as Collector Project and formerly Memory Project in criminal forums) steals sensitive information from the infected computer. The spyware reads sensitive data from the user’s web browser including their web cookies, personal information, and even login details (frequently stored in the web browser’s autofill feature). Specifically, the spyware targets the data files of Google Chrome, Firefox, and Internet Explorer.

Figure 2: VMRay Function Log showing fopen() API call to read sensitive data from Google Chrome

Other applications that have their authentication details targeted by this sample include:

- Authy (2FA desktop app)
- NordVPN
- FileZilla
- Steam
- Discord
- Pidgin

Finally, the sample takes a screenshot of the victim's desktop and adds all of the stolen information to a zip archive. The theft of login credentials means that an attacker will be able to log in as the infected user – potentially using the information to further-infiltrate an organization's network.

Loss of personal information can be devastating for a user and organization as it potentially leads to identity theft, extortion, banking fraud, stalking, and many other disastrous consequences. As part of the spyware-as-a-service model, criminals commonly trade personal details of victims, meaning that the attacker can sell the information to a criminal broker who specializes in exploiting the stolen information.

Password Security Side-Note

Web browser autofill features can be very convenient for users, however, it is also dangerous to use because login credentials will be saved in clear text on the user's computer. Many password managers can be installed as browser plugins and will enable users to login with similar ease. The benefit of proper password managers is that they will encrypt the information using a master password. Unlike a web browser's autofill feature, even if spyware is able to steal the files that contain login credentials, an attacker will be unable to read the sensitive data.

Network Traffic

VMRay automates the monitoring of network traffic with the in-depth network summary is available in the VMRay Analyzer Report (Figure 3.1). In addition, the complete packet-capture can also be downloaded (Figure 3.2) to be analyzed in full detail or sent to a network monitoring tool.

CollectorGoomba makes use of high-level networking features to retrieve a text file that contains the domain of the collection server. After the target domain is acquired the spyware attempts to upload the stolen data to the spyware-as-a-service collection server. The sample uses the API functions included in Wininet.dll, a Windows library of high-level network communication functions. These functions are easy to use and make the programming of this spyware-as-a-service sample much simpler for the developer. The first network traffic that the spyware will generate is from the function InternetReadFile(), which attempts to read a text file hosted in a publicly available GitHub repository.

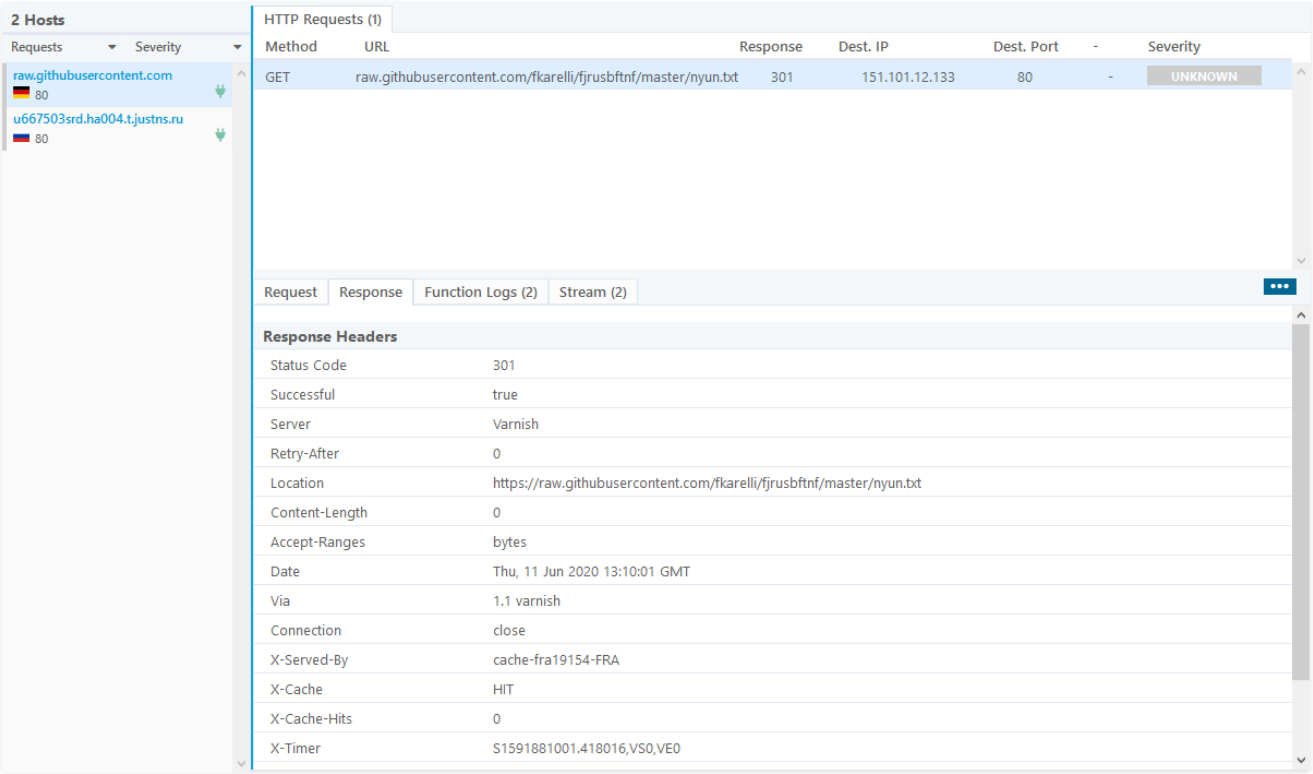


Figure 3.1: VMRay network overview of spyware sample requesting text file

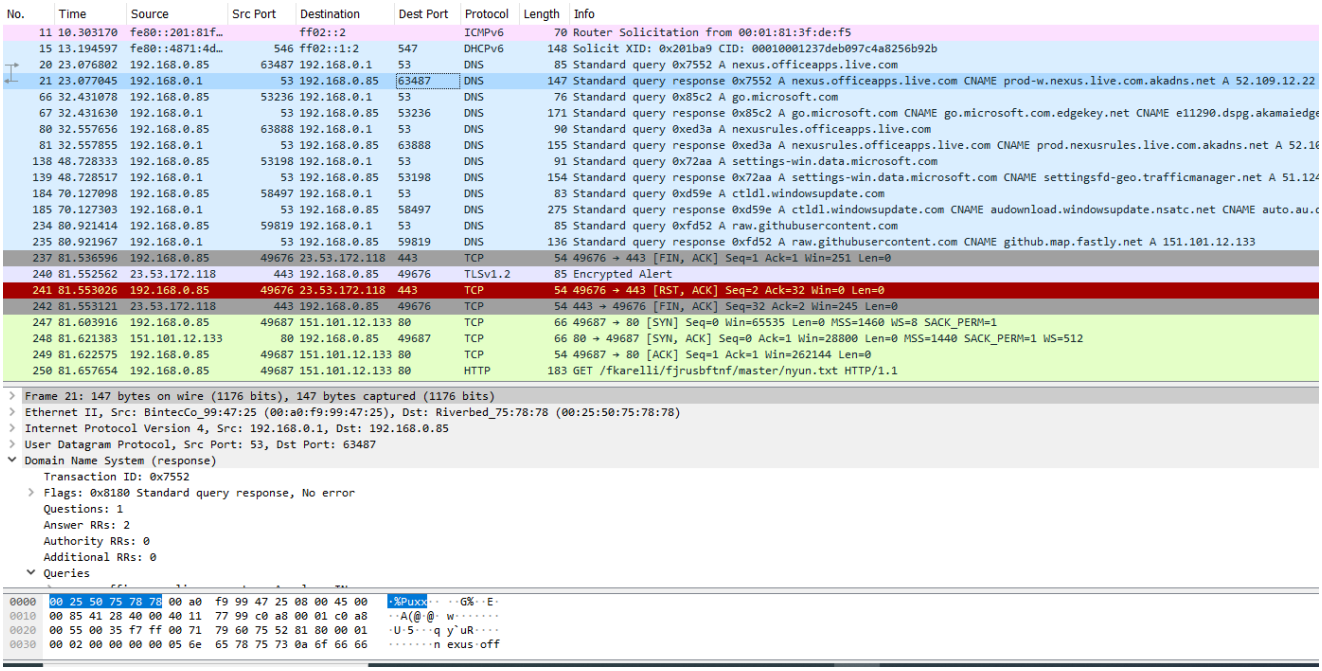


Figure 3.2: Downloaded packet capture detailing first network traffic generated by the sample

The generated network traffic will first appear in the packet capture as a DNS query for raw.githubusercontent.com – a legitimate subdomain hosted by Github that enables the direct downloading of files. After receiving the IP address for this GitHub server, the infected computer then reaches-out and download the contents of the text file nyun.txt from raw[.]githubusercontent[.]com/fkarelli/fjusbftnf/nyun[.]txt (Figure 4). The received text file contains the details of the spyware-as-a-service domain where the sample is instructed to upload the victim's stolen information.

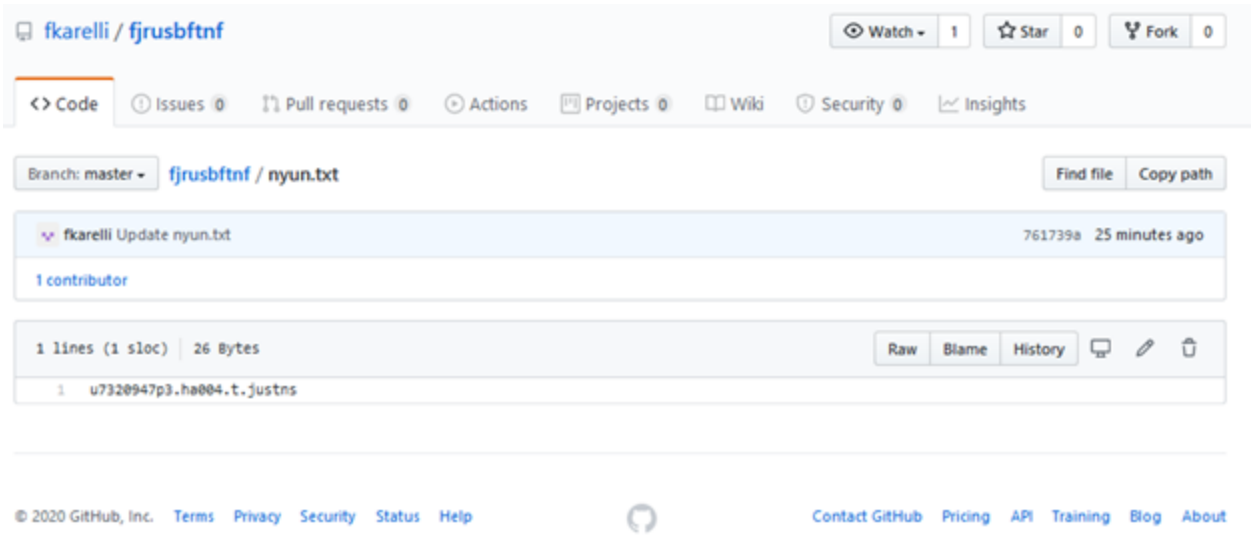


Figure 4: Screenshot of most recent text being hosted in GitHub

As can be seen in Figure 5, the program checks that it received a valid response. It confirms that the InternetReadFile() function did not return 0 (indicating that it did not have an error). Then it checks that the number of bytes returned is not 0, which would indicate that the text file was not received. If either condition is met, then the sample assumes that it failed to retrieve the domain information from nyun.txt and relies on a hard-coded domain for data exfiltration, which for this sample is u667503srd[.]ha004[.]t[.]justns.

If both the request succeeds and the return value is more than 0 bytes, then the spyware will assume that it received nyun.txt, relying on the returned text as the target domain. In the case of this execution, the contents of the received text file “u667503srd[.]ha004[.]t[.]justns” is combined with the top-level domain “[.]ru” in order to construct the fully qualified domain name u667503srd[.]ha004[.]t[.]justns[.]ru. Shortly after the sample was run, the developer for this spyware-as-a-service platform updated the text file in GitHub so that communicating spyware infections will receive “u7320947p3[.]ha004[.]t[.]justns” and will be directed to the new C2 domain u7320947p3[.]ha004[.]t[.]justns[.]ru.

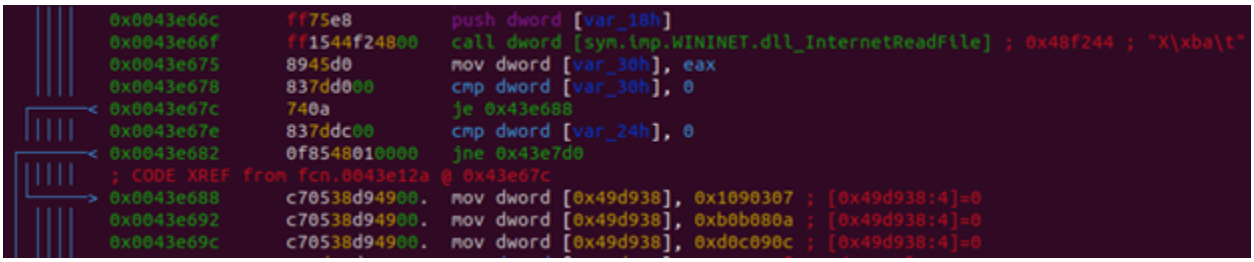


Figure 5: Decompiled Assembly where InternetReadFile() is called and the response is tested

When the spyware acquires its target domain, it exfiltrates the zip archive that contains all of the stolen data (Figure 6). The code calls high-level networking functions HttpSendRequest() and InternetWriteFile() to send an HTTP POST to u667503srd[.]ha004[.]t[.]justns[.]ru/collect[.]php. On the spyware command-and-control server, collect.php is listening for connections from spyware. According to the platform developers for this spyware-as-a-service, the attacker clients are able to connect to this server and access the data that they have stolen.

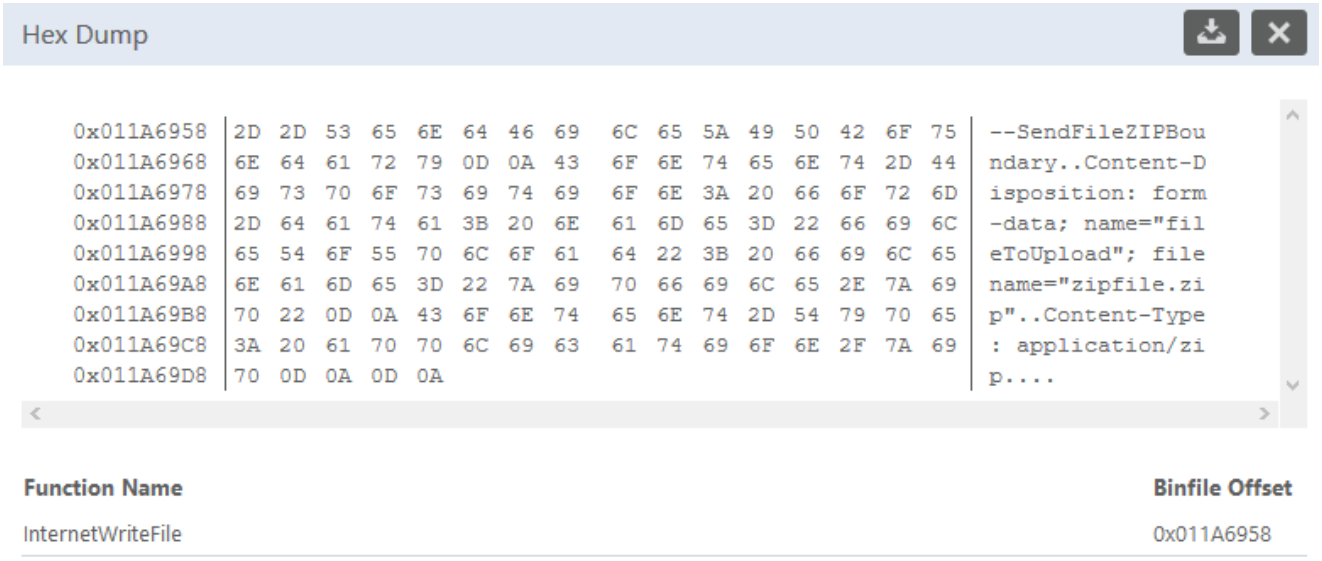


Figure 6: VMRay Data Log showing hex dump of the zip file being transmitted using InternetWriteFile()

After the stolen data is exfiltrated, the spyware deletes the temporary files it created, frees the memory it used, and finishes executing.

Command-and-Control Take Down

The VMRay Labs Team sent our findings of the C2 traffic to GitHub and were able to get the malicious repository removed. The attackers can no longer rely on this file to direct the malware to the data exfiltration server. Now when the spyware attempts to get nyun.txt from GitHub, it receives an error instead of the spyware-as-a-service domain * [.]ha0004[.]t[.]justns. CollectorGoomba has poorly programmed (as its namesake will suggest) and as can be seen in Figure 7, it attempts to upload the stolen data to “404: Not Found.ru/collect.php”. This attempt fails and the sample execution actually crashes. All of the attackers’ malware that was relying on this GitHub repository should now fail to upload the stolen data – regardless of which exfiltration server had been hard-coded into the malware.

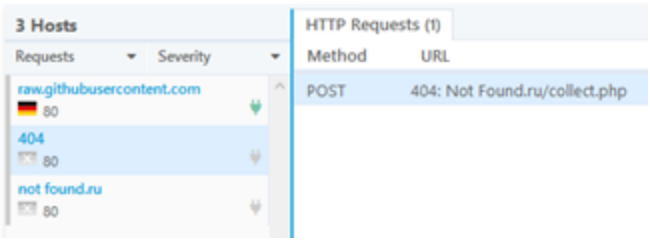


Figure 7: VMRay Network Overview showing the spyware’s attempt to upload data to an invalid domain

As early as Jun 20th, the spyware developers modified and updated their brand new code to now rely on text files hosted on upaste[.]me. However, any instance of CollectorGoomba which has been compiled before the update should still fail to upload the data it stole.

So why name is CollectorGoomba?

The spyware is malicious and can definitely harm victims, however, I understand that it is not a sophisticated credential stealer and that it was poorly programmed. During our search to confirm that this was indeed a new sample, we found several discussions on criminal forums about “COLLECTOR Project” and its predecessor “Memory Project”. Honestly, the basic coding of the spyware, malicious nature and ease of shutting it down reminded me of the basic enemies that you fight in Mario, Goombas.
Goombas can be harmful, but they are also unintelligent (historically they have very basic programming) and can be easily destroyed simply jumping on them. According to the game’s lore, however, Goombas can actually grow to become a larger threat if left alone. The new spyware strain, CollectorGoomba, is still under active development by its criminal programmers. While the spyware strain is currently easy to block and shut down – without analysis it could potentially grow to become a larger threat.

Network Defender’s Perspective

Using VMRay Analyzer defenders and researchers can see exactly how malware executes, even exporting log files as necessary. VMRay supports an add-on for Splunk, allowing analysts to submit the data generated by the VMRay Analyzer directly to Splunk for reporting and correlation with other sources. By using the analyzer, a SOC analyst can safely study the networking features of a potentially malicious program – understanding how the malware works and what indicators of compromise (IOCs) it will generate.

By analyzing this sample using VMRay, a network defender can see in the report that the sample will generate a request for raw[.]githubusercontent[.]com/fkarelli/fjrusbftnf/blob/master/nyun[.]txt and will exfiltrate data to a subdomain of justns[.]ru. By searching through an organization’s network logs, an incident responder can use this knowledge to find records of DNS queries or even the exfiltration HTTP traffic. Defenders can identify which computers on their network may have been infected. An organization’s intrusion prevention system (IPS) can also be set to monitor for these IOCs and automatically protect the affected computers. A network can also be set to blackhole the DNS queries for raw.githubusercontent.com, upaste[.]me, or *[.]ha0004[.]t[.]justns[.]ru. By configuring the network’s local DNS server to give an incorrect answer for those domain names, the spyware will be unable to connect to the command-and-control infrastructure (use caution before you block domains, your users may rely on sites such as raw.githubusercontent.com). With the C2 infrastructure safely blocked data will not be exfiltrated – similar to how older versions of CollectorGoomba will no longer be able to refer to GitHub.

IOCs

Sample:

SHA256: 49a56e067a73bb6f553b8df8a354d3b3328b8fffb64a459a1e719d86df89a322

C2 Infrastructure:

raw[.]githubusercontent[.]com/fkarelli/fjrusbftnf/master/nyun[.]txt
u7320947p3[.]ha0004[.]t[.]justns[.]ru/collect[.]php
u667503srd[.]ha004[.]t[.]justns[.]ru/collect[.]php
u667503gif[.]ha004[.]t[.]justns[.]ru/collect[.]php
u640763aha[.]ha004[.]t[.]justns[.]ru/collect[.]php
185[.]22[.]155[.]51 (observed hosting collect.php, April 2020 - June 2020)
upaste[.]me/r/4040523075fb98d9f (replaces GitHub in latest instance of spyware)

Spyware Samples Referencing C2 Infrastructure (SHA256):

0d27f5aec4935de8cf10ec74eb5c8558e57768f06ed118a0feed6fecebebaa34
0d65734eb25e7671cf618a2cd062a5f45ace06a4aeb8c3475c234daa2211f1ed
12fdab70f2ce661a0cd09c7862edb45aa9c974a564d1236dacf7ae81decd95af
1bbbf7558d64c231a1fd57b06386de9c28a914306ed1c3fe6c45b46335ef6798
31adc3a008913f0d63be55f536d936d405d7468bac97bc820c50ad4f598e7d21
36f9a4f21bafa4ade632e47d1f72d31eb0b41d647549f6f455c1ccca9242cde2
385651ce8441af1f43c9baf8fc24040a2eea53d574c193e5ba2618d09eef1050
3c04368599e361dab09da4e18f822db21c87d7a2531eab7c0e3c6baa5a0f7209
51e917806f84d3035b2d94cb3701b07ec47b3dc07a5b3e4dd38a5c552482a8bb
5216b6155c962abfd7d01ea02bbe7c7fd4e3c61f66437f1df6f2f0c128157b7a
53ad307d86d47f830a4decc093ccac947fb28a4909b7d4e8d02c909d1348d64c
5897c6061bf82ece002e1f4db3ea6e9e4ac27339223f66d85778f19fc1fb5bf6
76a0602451b6e0ab9e4f1843ae4455e8c0e1488450edd73bda7bd0a698ead565
7adfa2e759e2aac98647eea87fdaafb42127c734a524b064d44ad33471e2f7ee
7afcd27e5887e417d09657407e52e51d4f62cb070e43c9b698002750d6098129
85f5eaccf6bd35d7447b5e65171014d2de833599ed79fd3c2cecea9d946de8ae
8efa7e0b78331847d4e541e607da2ede323a506719773854c7643230b7a52994
95b98f660cd9a3940264e2d28f80ac6686384b4a5ff69f94fb5618cebfe91d6a
9b1c741dc51852aae7654a62f919c9755f4fce79077cf09316fcc764aa782d29
a4bd9a97b7dc82f254ed96749c5ab1ba8b92332e0d5dd7fc860588e252840f25
afc1dfa00008188ec3947dda0057a1e6f42330024e2b4e3fba74a5e40fea4d1f
b6cb4634459bd3eaf6bd3603d5795bbdfc30885f336d6b0b3050da3bb694d570
bca35de184fe234a061de7872a1b69b68738f900dbe1ed86db6e9514d59c270d
ce3f2b9a2704436f72efab3a30a622ec89413a9e4c157c0408474dd4573c947c
e453c1b908c18881bec40c6ae1e85bf64d12ef84d7a9a704cf957af83252af4e
e9daf12c4d651fa2ce757f1ca6209917bb272c37f1b7d65a4e6f6df32dbccae5
f120d323fc380bdfb8573dba310c9f66aefd890e0f5b624add1d1af15c51937e
fcdefe8655c9b54d44a587435ef2d19320b0deba57d52afabc3f4f1416aee2d7

Active Samples (now using upaste[.]me instead of GitHub):

639bd88a73154bd38aa18eaea3e968b76f4431ade64d25936cc7e34509075f94
c2b96838c24b59490a318b4165ae8231b9ed2f7e1b0cb61391c7816ff0f859f9

VMRay Analyzer Reports for Related Samples:

https://www.vmray.com/analyses/385651ce8441/report/overview.html
https://www.vmray.com/analyses/51e917806f84/report/overview.html
https://www.vmray.com/analyses/ce3f2b9a2704/report/overview.html
https://www.vmray.com/analyses/639bd88a7315/report/overview.html
https://www.vmray.com/analyses/c2b96838c24b/report/overview.html