CUJOAI

FIRST
June 2022

Reversing Golang Binaries with Ghidra

Dorka Palotay
Senior Threat Researcher, CUJO AI

Who am I

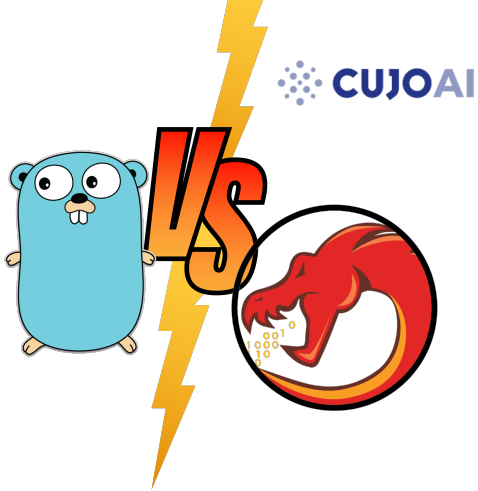
Background

Dorka Palotay (@pad0rka):

- Senior Threat Researcher at CUJO AI
- BSc in Applied Mathematics
- MSc in Security and Privacy – Advanced Cryptography
- Worked at financial and security companies as well
- Malware researcher and reverse engineer
- Member of last4ofus CTF team
  (with Filip Savin, Zoltan Balazs, Albert Zsigovits)
  - 2020 First CTF winner
  - 2021 First CTF second place

Research topic

The quest



Background:

- IoT malware research -> more and more (IoT) malware families are written in Go

Issue:

- Reverse engineering Go binaries is challenging
  - Huge file size
  - Unusual string handling
  - No symbol names due to stripping
- Ghidra open-source development is in early stage compared to other tools
  - Only a few open-source scripts are available, solving only parts of the problem

Goal:

- Making reverse engineering Go binaries with Ghidra easier

Steps:

- Understand Go and the differences from usual languages
- Get familiar with Ghidra's features (In this research we used Ghidra 10.0)
- Create our own scripts: https://github.com/getCUJO/ThreatIntel

Golang

Introduction

CUJOAI

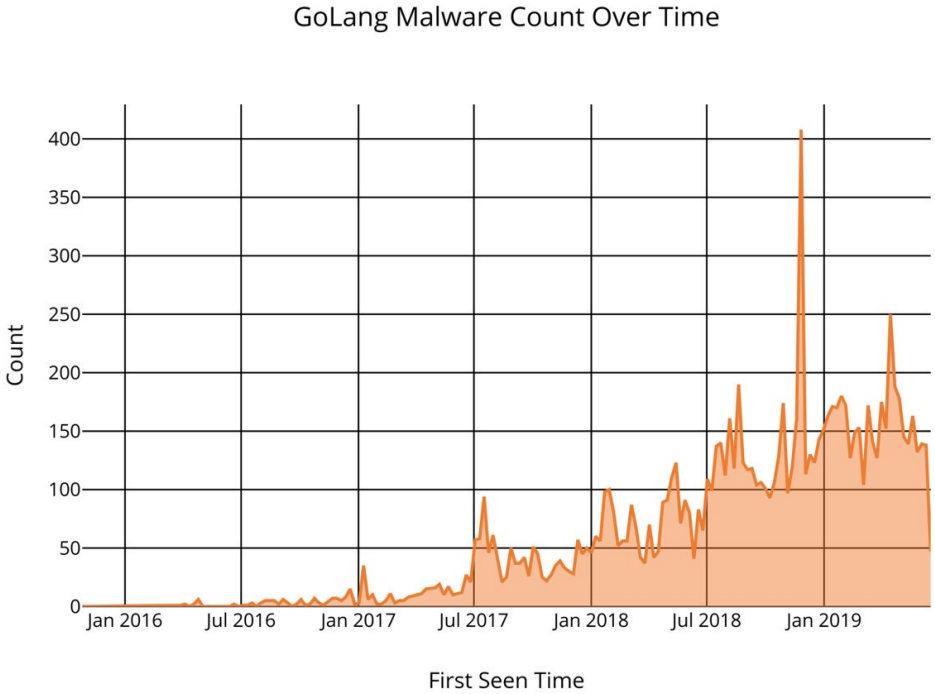
- Go (also called Golang) is an open source programming language
- Designed by Google in 2007
- Made available to the public in 2012
- Current version is Go 1.18
- https://golang.org/

- Go comes out top of the languages most developers want to learn[1]
- Advantages:
  - Simple and clear documentation
  - Easy to learn, ease of coding
  - Compiled language (faster than Python)
  - Cross compiling (Windows, Linux, macOS)
  - Scalability and concurrency
  - Garbage collection – automatic memory management

1: https://www.zdnet.com/article/developers-say-googles-go-is-most-sought-after-programming-language-of-2020/

Malware families

A surge in Go malware



https://unit42.paloaltonetworks.com/the-gopher-in-the-room-analysis-of-golang-malware-in-the-wild/

Static linking

Big Bad Binaries

CUJOAI

- Go binaries are statically linked by default
- All the necessary libraries are included in the executable image
- No dependency issues
- Large size
    - Difficult malware distribution
    - Anti – virus products have difficulty to detect
    - Reverse engineering can be more time consuming

Hello World - Unstripped

C vs Go

CUJOAI

- C

```c
#include <stdio.h>

int main()
{
    printf("Hello, World!\n");
    return 0;
}
```

gcc -o world_c world.c

ELF 64-bit LSB shared object,
x86-64, version 1 (SYSV),
dynamically linked,
not stripped

size: 16,3 kB

- Go

```go
package main

import "fmt"

func main(){
    fmt.Printf("Hello, World!\n")
}
```

go build -o world_go world.go

ELF 64-bit LSB executable,
x86-64, version 1 (SYSV),
statically linked,
not stripped

size: 2,0 MB

Hello World in Ghidra

C vs Go

CUJOAI

Functions - 19 items

Name	Location	Function Signature	Function Size
_init	00101000	int _init(EVP_...	27
FUN_00101020	00101020	undefined FUN_...	13
__cxa_finalize	00101040	thunk undefine...	11
puts	00101050	thunk int puts...	11
_start	00101060	undefined _sta...	47
deregister_tm_clones	00101090	undefined dere...	34
register_tm_clones	001010c0	undefined regi...	51
__do_global_dtors_aux	00101100	undefined __do...	54
frame_dummy	00101140	thunk undefine...	9
main	00101149	undefined main()	27
__libc_csu_init	00101170	undefined __li...	101
__libc_csu_fini	001011e0	undefined __li...	5
_fini	001011e8	undefined _fini()	13
_ITM_deregisterTMCloneTable	00105000	thunk undefine...	1
puts	00105008	thunk int puts...	1
__libc_start_main	00105010	thunk undefine...	1
__gmon_start__	00105018	thunk undefine...	1
_ITM_registerTMCloneTable	00105020	thunk undefine...	1
__cxa_finalize	00105028	thunk undefine...	1

Functions - 1790 items

Name	Location	Function Signat...	Function Size
internal/cpu.Initialize	00401000	undefined int...	78
internal/cpu.processOptions	00401060	undefined int...	1877
internal/cpu.indexByte	004017c0	undefined int...	53
internal/cpu.doinit	00401800	undefined int...	1029
internal/cpu.cpuid	00401c20	undefined int...	27
internal/cpu.xgetbv	00401c40	undefined int...	17
type..eq.internal/cpu.CacheLinePad	00401c60	undefined typ...	6
type..eq.internal/cpu.option	00401c80	undefined typ...	165
type..eq.[15]internal/cpu.option	00401d40	undefined typ...	139
runtime/internal/sys.OnesCount64	00401de0	undefined run...	119
runtime/internal/atomic.Cas64	00401e60	undefined run...	26
runtime/internal/atomic.Casuintptr	00401e80	thunk undefin...	5
runtime/internal/atomic.Storeuintptr	00401ea0	thunk undefin...	5
runtime/internal/atomic.Store	00401ec0	undefined run...	12
runtime/internal/atomic.Store64	00401ee0	undefined run...	14
internal/bytealg.init.0	00401f00	undefined int...	34
cmpbody	00401f40	undefined cmp...	569
runtime.cmpstring	00402180	undefined run...	30
memeqbody	004021a0	undefined mem...	318
runtime.memequal	004022e0	undefined run...	36
runtime.memequal_varlen	00402320	undefined run...	35
indexbytebody	00402360	undefined ind...	279
internal/bytealg.IndexByteString	00402480	undefined int...	24
runtime.memhash128	004024a0	undefined run...	89
runtime.strhashFallback	00402500	undefined run...	98
runtime.f32hash	00402580	undefined run...	282
runtime.f64hash	004026a0	undefined run...	284
runtime.c64hash	004027c0	undefined run...	110
runtime.c128hash	00402840	undefined run...	110

19 functions vs 1790 functions

Binaries: world_c, world_go

Stripped Binaries

CUJOAI

- Discard debugging symbols
- Reduced size
- No names for routines and variables
- More difficult debugging and reverse engineering
- Malware files are usually stripped

Hello World - Stripped

C vs Go

CUJOAI

- C

```c
#include <stdio.h>

int main()
{
    printf("Hello, World!\n");
    return 0;
}
```

gcc -o world_c_strip **-s** world.c

ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, **stripped**

size: 14,1 kB

- Go

```go
package main

import "fmt"

func main(){
    fmt.Printf("Hello, World!\n")
}
```

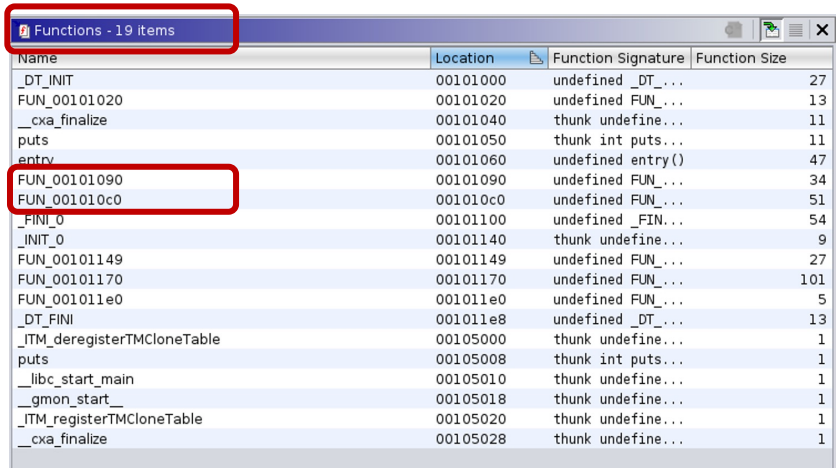
go build -o world_go_strip – **ldflags "-s"** world.go

ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, **stripped**

size: 1,3 MB

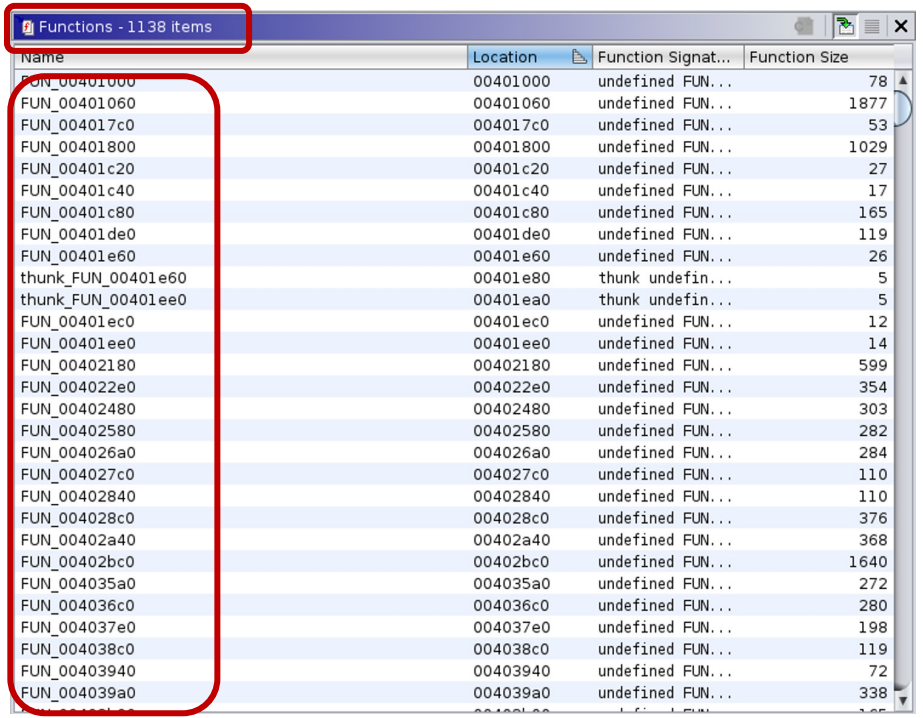
Hello World Stripped in Ghidra

C vs Go



Functions - 19 items

Name	Location	Function Signature	Function Size
_DT_INIT	00101000	undefined _DT_...	27
FUN_00101020	00101020	undefined FUN_...	13
__cxa_finalize	00101040	thunk undefine...	11
puts	00101050	thunk int puts...	11
entry	00101060	undefined entry()	47
FUN_00101090	00101090	undefined FUN_...	34
FUN_001010c0	001010c0	undefined FUN_...	51
_FINI_0	00101100	undefined _FIN...	54
_INIT_0	00101140	thunk undefine...	9
FUN_00101149	00101149	undefined FUN_...	27
FUN_00101170	00101170	undefined FUN_...	101
FUN_001011e0	001011e0	undefined FUN_...	5
_DT_FINI	001011e8	undefined _DT_...	13
_ITM_deregisterTMCloneTable	00105000	thunk undefine...	1
puts	00105008	thunk int puts...	1
__libc_start_main	00105010	thunk undefine...	1
__gmon_start__	00105018	thunk undefine...	1
_ITM_registerTMCloneTable	00105020	thunk undefine...	1
__cxa_finalize	00105028	thunk undefine...	1



Functions - 1138 items

Name	Location	Function Signat...	Function Size
FUN_00401000	00401000	undefined FUN...	78
FUN_00401060	00401060	undefined FUN...	1877
FUN_004017c0	004017c0	undefined FUN...	53
FUN_00401800	00401800	undefined FUN...	1029
FUN_00401c20	00401c20	undefined FUN...	27
FUN_00401c40	00401c40	undefined FUN...	17
FUN_00401c80	00401c80	undefined FUN...	165
FUN_00401de0	00401de0	undefined FUN...	119
FUN_00401e60	00401e60	undefined FUN...	26
thunk_FUN_00401e60	00401e80	thunk undefin...	5
thunk_FUN_00401ee0	00401ea0	thunk undefin...	5
FUN_00401ec0	00401ec0	undefined FUN...	12
FUN_00401ee0	00401ee0	undefined FUN...	14
FUN_00402180	00402180	undefined FUN...	599
FUN_004022e0	004022e0	undefined FUN...	354
FUN_00402480	00402480	undefined FUN...	303
FUN_00402580	00402580	undefined FUN...	282
FUN_004026a0	004026a0	undefined FUN...	284
FUN_004027c0	004027c0	undefined FUN...	110
FUN_00402840	00402840	undefined FUN...	110
FUN_004028c0	004028c0	undefined FUN...	376
FUN_00402a40	00402a40	undefined FUN...	368
FUN_00402bc0	00402bc0	undefined FUN...	1640
FUN_004035a0	004035a0	undefined FUN...	272
FUN_004036c0	004036c0	undefined FUN...	280
FUN_004037e0	004037e0	undefined FUN...	198
FUN_004038c0	004038c0	undefined FUN...	119
FUN_00403940	00403940	undefined FUN...	72
FUN_004039a0	004039a0	undefined FUN...	338

Binaries: world_c_strip, world_go_strip

19 functions vs 1138 functions

Recover function names

strings

```
> strings world_c | grep -o ".\{0,10\}main.\{0,10\}"
ibc_start_main
ibc_start_main@@GLIBC_2.
main
```

```
> strings world_c_strip | grep -o ".\{0,10\}main.\{0,10\}"
ibc_start_main
```

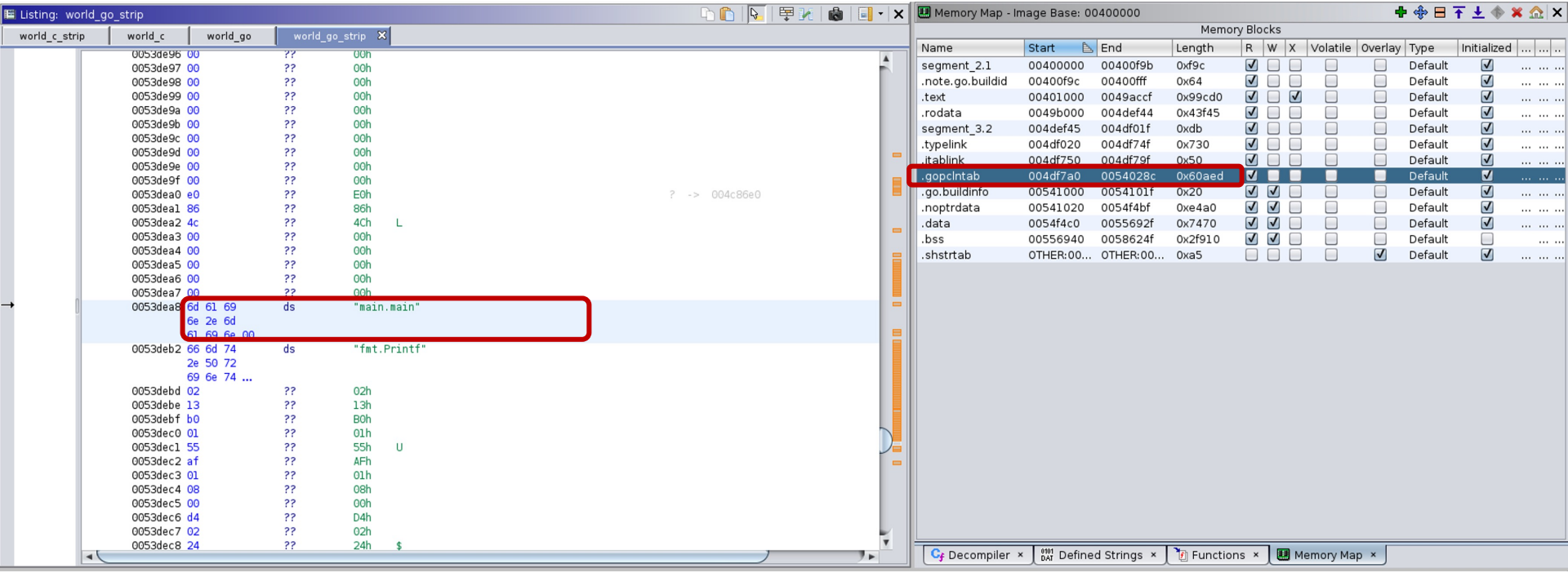
```
> strings world_go | grep -o ".\{0,10\}main.\{0,10\}"
hasmain
edruntime.main not on m0
 p stateremaining pointe
 out of domainpanic whil
e space remainingreflect
routines (main called ru
runtime.main
runtime.main.func1
runtime.main.func2
main.main
main..inittask
runtime.main_init_done
runtime.mainStarted
runtime.mainPC
runtime.main
runtime.main.func1
runtime.main.func2
main.main
```

```
> strings world_go_strip | grep -o ".\{0,10\}main.\{0,10\}"
hasmain
edruntime.main not on m0
 p stateremaining pointe
 out of domainpanic whil
e space remainingreflect
routines (main called ru
runtime.main
runtime.main.func1
runtime.main.func2
main.main
```

Binaries: world_c, world_go, world_c_strip, world_go_strip

Recover function names

pclntab

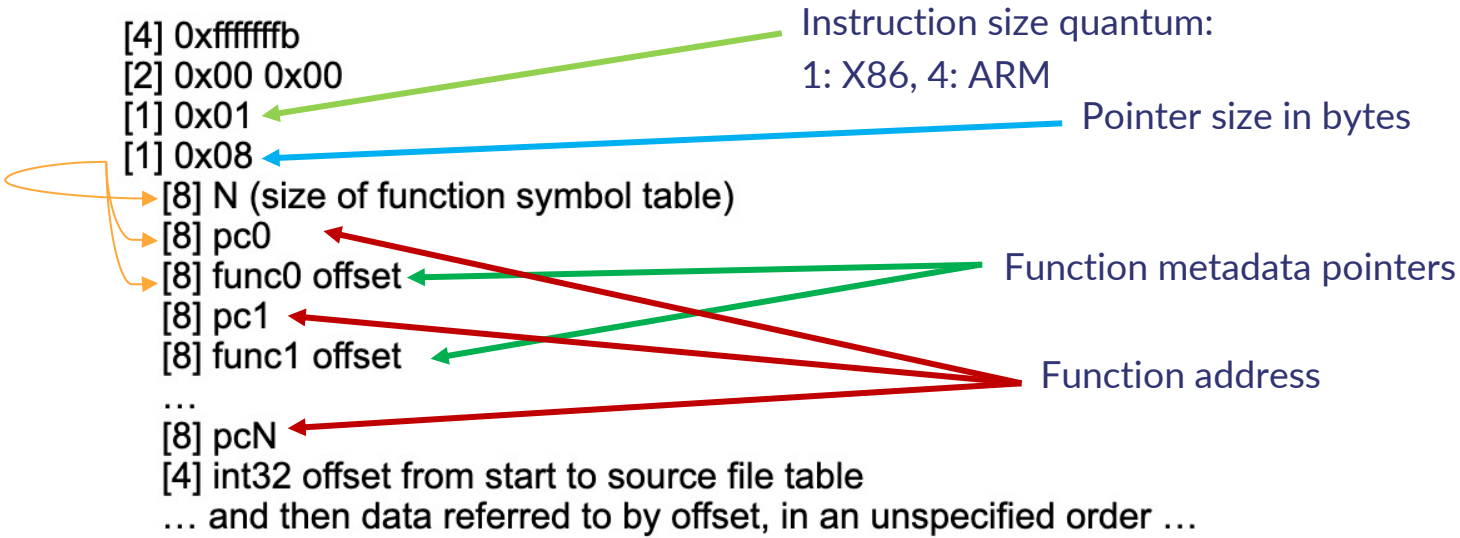


Binary: world_go_strip

Recover function names

pclntab

- Detailed documentation of pclntab[1] is available



1: https://docs.google.com/document/d/1lyPIbmsYbXnpNj57a261hgOYVpNRcgydurVQIyZOz_o/pub

Recover function names

pclntab

- Function metadata

```
struct        Func
{
        uintptr        entry;   // start pc
        int32 name;             // name (offset to C string)
        int32 args;             // size of arguments passed to function
        int32 frame;            // size of function frame, including saved caller PC
        int32        pcsp;                  // pcsp table (offset to pcvalue table)
        int32        pcfile;            // pcfile table (offset to pcvalue table)
        int32        pcln;                  // pcln table (offset to pcvalue table)
        int32        nfuncdata;           // number of entries in funcdata list
        int32        npcdata;            // number of entries in pcdata list
};
```

Recover function names

pclntab (from go 1.16 and go 1.18)

CUJOAI

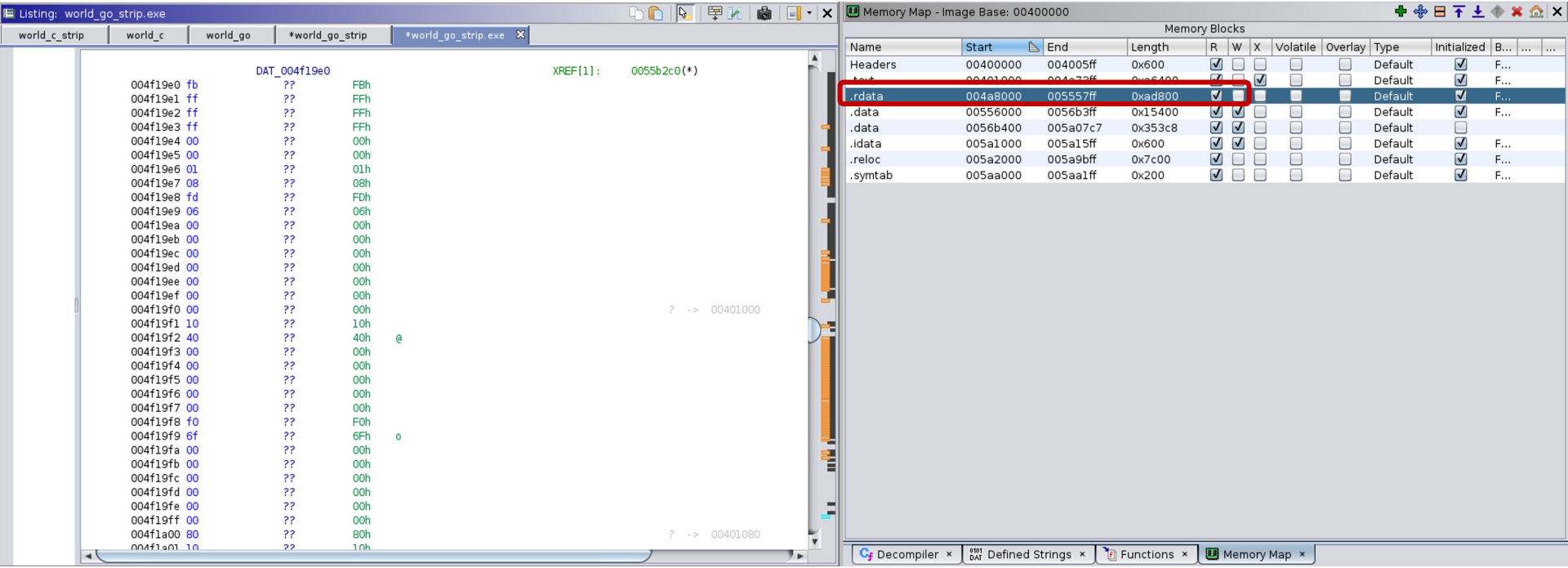
```
// pcHeader holds data used by the pclntab lookups.
type pcHeader struct {
        magic          uint32  // 0xFFFFFFFA
        pad1, pad2     uint8   // 0,0
        minLC          uint8   // min instruction size
        ptrSize        uint8   // size of a ptr in bytes
        nfunc          int     // number of functions in the module
        nfiles         uint    // number of entries in the file tab
        funcnameOffset uintptr // offset to the funcnametab variable from pcHeader
        cuOffset       uintptr // offset to the cutab variable from pcHeader
        filetabOffset  uintptr // offset to the filetab variable from pcHeader
        pctabOffset    uintptr // offset to the pctab variable from pcHeader
        pclnOffset     uintptr // offset to the pclntab variable from pcHeader
}
```

```
// pcHeader holds data used by the pclntab lookups.
type pcHeader struct {
        magic          uint32  // 0xFFFFFFF0
        pad1, pad2     uint8   // 0,0
        minLC          uint8   // min instruction size
        ptrSize        uint8   // size of a ptr in bytes
        nfunc          int     // number of functions in the module
        nfiles         uint    // number of entries in the file tab
        textStart      uintptr // base for function entry PC offsets in this module, equal to
        funcnameOffset uintptr // offset to the funcnametab variable from pcHeader
        cuOffset       uintptr // offset to the cutab variable from pcHeader
        filetabOffset  uintptr // offset to the filetab variable from pcHeader
        pctabOffset    uintptr // offset to the pctab variable from pcHeader
        pclnOffset     uintptr // offset to the pclntab variable from pcHeader
}
```

Recover function names

pclntab in Windows

- Not a separate section -> Look for the structure



Binary: world_go_strip.exe

Recover function names

Idea

Function name recovery steps:

- Locate pclntab structure
- Extract function addresses
- Find function name offsets

```
//
// .gopclntab
// SHT_PROGBITS  [0x4df7a0 - 0x54028c]
// ram:004df7a0-ram:0054028c
//
DAT_004df7a0

004df7a0 fb ??  FBh
004df7a1 ff ??  FFh
004df7a2 ff ??  FFh
004df7a3 ff ??  FFh
004df7a4 00 ??  00h
004df7a5 00 ??  00h
004df7a6 01 ??  01h
004df7a7 08 ??  08h
004df7a8 ef ??  EFh
004df7a9 06 ??  06h
004df7aa 00 ??  00h
004df7ab 00 ??  00h
004df7ac 00 ??  00h
004df7ad 00 ??  00h
004df7ae 00 ??  00h
004df7af 00 ??  00h
004df7b0 00 ??  00h
004df7b1 10 ??  10h
004df7b2 40 ??  40h  @
004df7b3 00 ??  00h
```

```
main.main

0049ac60 64 48 8b 0c 25 f8 ff ff ff   MOV   RCX,qword ptr FS:[0xfffffff8]
0049ac69 48 3b 61 10                  CMP   RSP,qword ptr [RCX + 0x10]
0049ac6d 76 5a                        JBE   LAB_0049acc9
0049ac6f 48 83 ec 58                  SUB   RSP,0x58
0049ac73 48 89 6c 24 50               MOV   qword ptr [RSP + local_8],RBP
```

```
004e6690 60 ??  60h  `
004e6691 ac ??  ACh
004e6692 49 ??  49h  I
004e6693 00 ??  00h
004e6694 00 ??  00h
004e6695 00 ??  00h
004e6696 00 ??  00h
004e6697 00 ??  00h
004e6698 a8 ??  A8h
004e6699 e6 ??  E6h
004e669a 05 ??  05h
004e669b 00 ??  00h
004e669c 00 ??  00h
004e669d 00 ??  00h
004e669e 00 ??  00h
004e669f 00 ??  00h
```

0x4df7a0 + 0x5e6a8 = 0x53DE48

```
0053de48 60 ??  60h  `
0053de49 ac ??  ACh
0053de4a 49 ??  49h  I
0053de4b 00 ??  00h
0053de4c 00 ??  00h
0053de4d 00 ??  00h
0053de4e 00 ??  00h
0053de4f 00 ??  00h
0053de50 08 ??  08h
0053de51 e7 ??  E7h
0053de52 05 ??  05h
0053de53 00 ??  00h
0053de54 00 ??  00h
0053de55 00 ??  00h
0053de56 00 ??  00h
0053de57 00 ??  00h
```

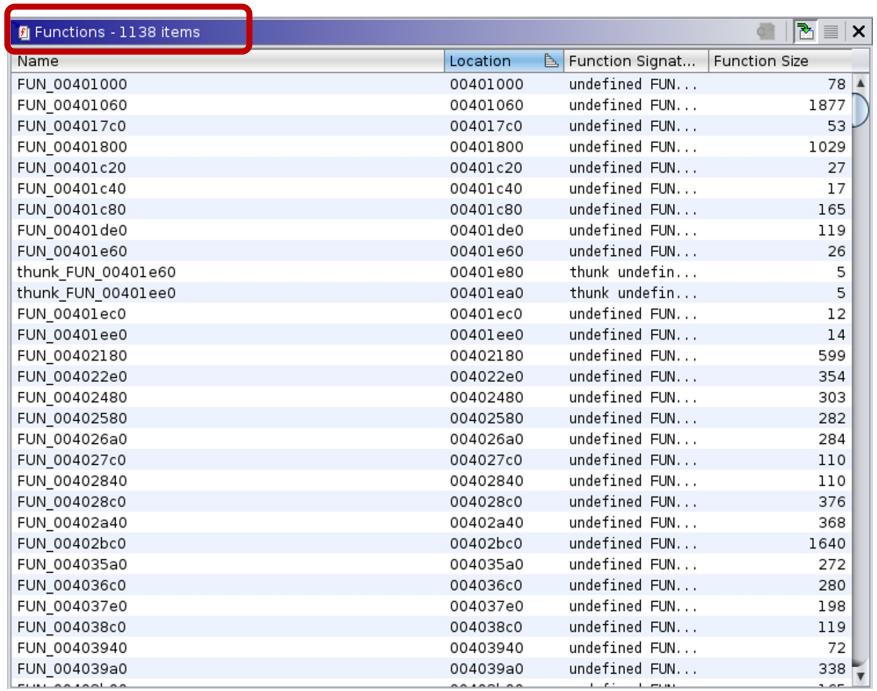
0x4df7a0 + 0x5e708 = 0x53DEA8

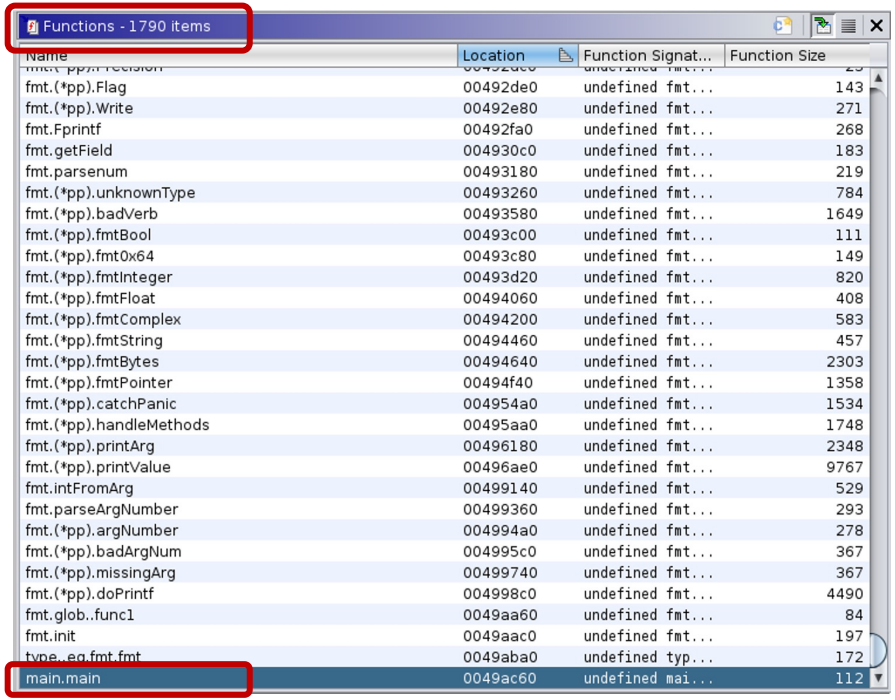
```
0053dea8 6d 61 69 6e 2e 6d 61 69 6e 00   ds   "main.main"
```

Binary: world_go_strip

Recover function names

Executing our script



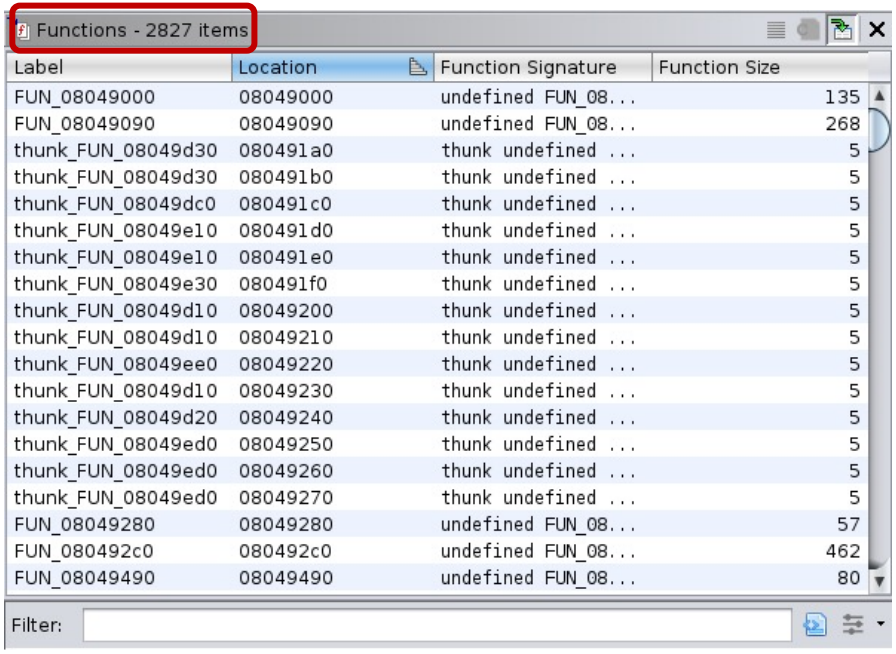


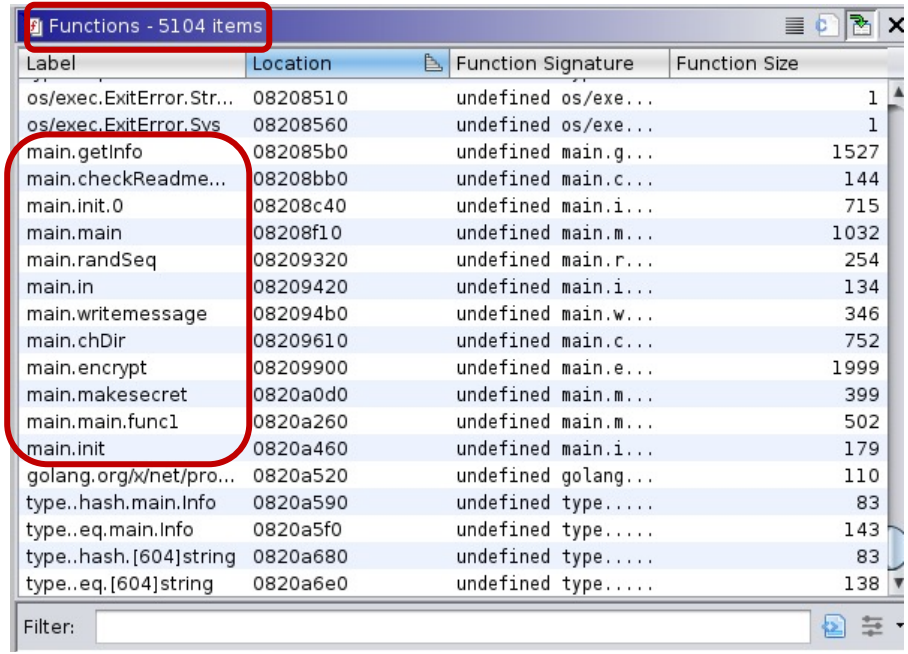
Binary: world_go_strip

Recover function names

Real world example – eCh0raix

CUJOAI





Binary: eCh0raix – x86

Recover function names

Challenges

- Undefined function name strings

```
                     *************************************************************
                     *                          FUNCTION                         *
                     *************************************************************
                     undefined FUN_08184fa0(undefined4 param_1, undefined4 pa...
     undefined         AL:1           <RETURN>
     undefined4        Stack[0x4]:4   param_1                                 XREF[1]:     08184fc7(R)
     undefined4        Stack[0x8]:4   param_2                                 XREF[2]:     08184fd8(R),
                                                                                           0818501d(R)
     undefined4        Stack[0xc]:4   param_3                                 XREF[2]:     08184ff0(R),
                                                                                           0818500b(R)
     undefined4        Stack[0x10]:4  param_4                                 XREF[1]:     08184fdf(R)
     undefined4        Stack[0x14]:4  param_5                                 XREF[1]:     08184ff7(R)
     undefined4        Stack[0x18]:4  param_6                                 XREF[1]:     08184ffe(W)
     undefined4        Stack[-0x4]:4  local_4                                 XREF[1]:     08184fc3(R)
     undefined4        Stack[-0x8]:4  local_8                                 XREF[1]:     08184fbb(*)
                     FUN_08184fa0                                        XREF[2]:     0818502f(c),
                                                                                      log.init:08186012(c)
08184fa0 65 8b 0d       MOV        ECX,dword ptr GS:[0x0]
         00 00 00 00
08184fa7 8b 89 fc       MOV        ECX,dword ptr [ECX + 0xfffffffc]
         ff ff ff
```

```
083aa0e4 6c             ??         6Ch    l
083aa0e5 6f             ??         6Fh    o
083aa0e6 67             ??         67h    g
083aa0e7 2e             ??         2Eh    .
083aa0e8 4e             ??         4Eh    N
083aa0e9 65             ??         65h    e
083aa0ea 77             ??         77h    w
083aa0eb 00             ??         00h
```

```
func_name = getDataAt(name_address)

#Try to define function name string.
if func_name is None:
    try:
        func_name = createAsciiString(name_address)
    except:
        print "ERROR: No name"
        continue
```

Binary: eCh0raix – x86

Hello World Strings in Ghidra

C vs Go

CUJOAI

Defined Strings - 70 items

Location	String Value	String Representat...	Data Type
.strtab::000000db	__GNU_EH_FRAME_HDR	"__GNU_EH_FRAME_...	ds
.strtab::000000ee	_GLOBAL_OFFSET_TABLE_	"_GLOBAL_OFFSET_...	ds
.strtab::00000104	__libc_csu_fini	"__libc_csu_fini"	ds
.strtab::00000114	_ITM_deregisterTMCloneTable	"_ITM_deregisterTM...	ds
.strtab::00000130	puts@@GLIBC_2.2.5	"puts@@GLIBC_2.2...	ds
.strtab::00000142	_edata	"_edata"	ds
.strtab::00000149	__libc_start_main@@GLIBC_2.2.5	"__libc_start_main...	ds
.strtab::00000168	__data_start	"__data_start"	ds
.strtab::00000175	__gmon_start__	"__gmon_start__"	ds
.strtab::00000184	__dso_handle	"__dso_handle"	ds
.strtab::00000191	_IO_stdin_used	"_IO_stdin_used"	ds
.strtab::000001a0	__libc_csu_init	"__libc_csu_init"	ds
.strtab::000001b0	__bss_start	"__bss_start"	ds
.strtab::000001bc	main	"main"	ds
.strtab::000001c1	__TMC_END__	"__TMC_END__"	ds
.strtab::000001cd	_ITM_registerTMCloneTable	"_ITM_registerTMCl...	ds
.strtab::000001e7	__cxa_finalize@@GLIBC_2.2.5	"__cxa_finalize@@G...	ds
00100001	ELF	"ELF"	ds
00100318	/lib64/ld-linux-x86-64.so.2	"/lib64/ld-linux-x86-...	ds
00100471	libc.so.6	"libc.so.6"	ds
0010047b	puts	"puts"	ds
00100480	__cxa_finalize	"__cxa_finalize"	ds
0010048f	__libc_start_main	"__libc_start_main"	ds
001004a1	GLIBC_2.2.5	"GLIBC_2.2.5"	ds
001004ad	_ITM_deregisterTMCloneTable	"_ITM_deregisterTM...	ds
001004c9	__gmon_start__	"__gmon_start__"	ds
001004d8	_ITM_registerTMCloneTable	"_ITM_registerTMCl...	ds
00102004	Hello, World!	"Hello, World!"	ds
00102061	zR	"zR"	ds

Defined Strings - 6544 items

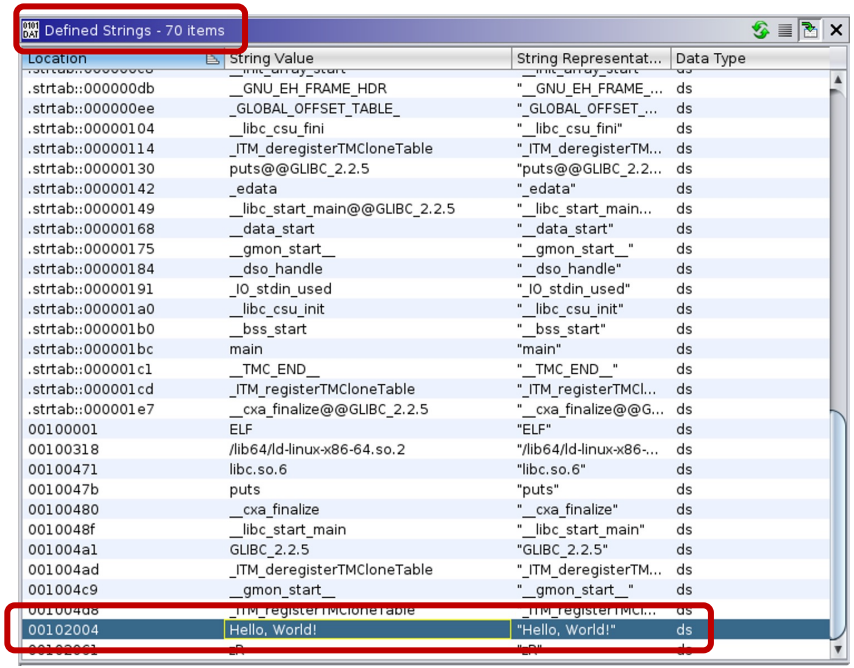
Location	String Value	String Representation	Data Type
.shstrtab::00000001	.text	".text"	ds
.shstrtab::00000007	.noptrdata	".noptrdata"	ds
.shstrtab::00000012	.data	".data"	ds
.shstrtab::00000018	.bss	".bss"	ds
.shstrtab::0000001d	.noptrbss	".noptrbss"	ds
.shstrtab::00000027	__libfuzzer_extra_counters	"__libfuzzer_extra_coun...	ds
.shstrtab::00000042	.go.buildinfo	".go.buildinfo"	ds
.shstrtab::00000050	.note.go.buildid	".note.go.buildid"	ds
.shstrtab::00000061	.elfdata	".elfdata"	ds
.shstrtab::0000006a	.rodata	".rodata"	ds
.shstrtab::00000072	.typelink	".typelink"	ds
.shstrtab::0000007c	.itablink	".itablink"	ds
.shstrtab::00000086	.gosymtab	".gosymtab"	ds
.shstrtab::00000090	.gopclntab	".gopclntab"	ds
.shstrtab::0000009b	.symtab	".symtab"	ds
.shstrtab::000000a3	.strtab	".strtab"	ds
.shstrtab::000000ab	.debug_abbrev	".debug_abbrev"	ds
.shstrtab::000000b9	.zdebug_abbrev	".zdebug_abbrev"	ds
.shstrtab::000000c8	.debug_frame	".debug_frame"	ds
.shstrtab::000000d5	.zdebug_frame	".zdebug_frame"	ds
.shstrtab::000000e3	.debug_info	".debug_info"	ds
.shstrtab::000000ef	.zdebug_info	".zdebug_info"	ds
.shstrtab::000000fc	.debug_loc	".debug_loc"	ds
.shstrtab::00000107	.zdebug_loc	".zdebug_loc"	ds
.shstrtab::00000113	.debug_line	".debug_line"	ds
.shstrtab::0000011f	.zdebug_line	".zdebug_line"	ds
.shstrtab::0000012c	.debug_pubnames	".debug_pubnames"	ds
.shstrtab::0000013c	.zdebug_pubnames	".zdebug_pubnames"	ds
.shstrtab::0000014d	.debug_pubtypes	".debug_pubtypes"	ds

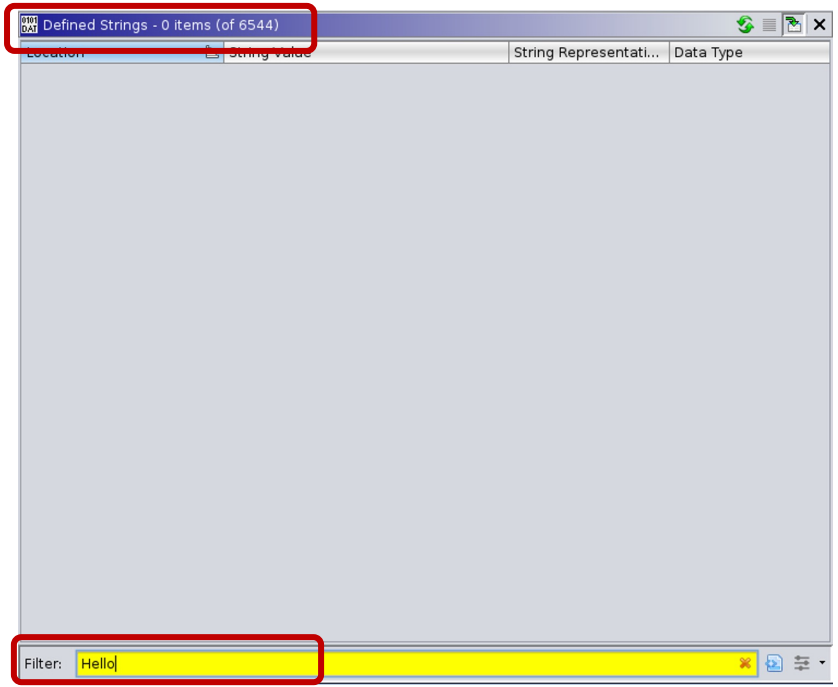
Binaries: world_c, world_go

70 defined strings vs 6544 defined strings

Hello World Strings in Ghidra

C vs Go





No “Hello” in Go

Binaries: world_c, world_go

Hello World Strings

C vs Go

CUJOAI

C:

“Hello, World!” is easy to find

```
> strings world_c | grep Hello
Hello, World!
```

Go:

“Hello, World!” is part of a huge string

```
> strings world_go | grep Hello
entersyscallgcBitsArenasgcpacertracehost is downillegal seekinvalid slotlfstack.pushmadvdontneedmheapSpecialmspanSpecialnot pollableraceF
iniLockreleasep: m=runtime: gp=runtime: sp=short bufferspanSetSpinesweepWaiterstraceStringsuname failedwirep: p->m= != sweepgen  MB) work
ers= called from  failed with  flushedWork  heap_marked= idlethreads= is nil, not  nStackRoots= s.spanclass= span.base()= syscalltick= wo
rk.nproc=  work.nwait= , gp->status=, not pointer-byte block (3814697265625GC sweep waitGunjala_GondiHello, World!Masaram_GondiMende_Kika
kuiOld_HungarianSIGKILL: killSIGQUIT: quitbad flushGen bad map statedebugCall2048exchange fullfatal error: level 3 resetload64 failedmin
too largenil stackbaseout of memorysrmount errortimer expiredtraceStackTabtriggerRatio=value method xadd64 failedxchg64 failed}
```

Binaries: world_c, world_go_println

String Representation

C vs Go

CUJOAI

C

- sequence of characters terminated with a null character

Go

- sequence of bytes with a fixed length
- not null terminated
- str – sequence of bytes
- len – number of bytes
- https://golang.org/src/runtime/string.go
- Large string blobs from concatenated strings until null character
- Ghidra has a hard time defining strings in Go binaries

```
type stringStruct struct {
        str unsafe.Pointer
        len int
}
```

Idea: help Ghidra to find string structures

- Static vs dynamic allocation
- Per architecture (different instruction set)
- Multiple solution within one architecture
- Possible changes per Go version

Dynamically allocated string structure

x86

- String structures can be allocated runtime
- Several different scenarios
- Let's look at the Hello World examples again

```
                        s_Hello,_World!_00102004                        XREF[1]:     main:00101151(*)
00102004 48 65 6c        ds         "Hello, World!"
         6c 6f 2c
         20 57 6f ...
```

```
                     *************************************************************
                     *                          FUNCTION                         *
                     *************************************************************
                     undefined main()
     undefined         AL:1           <RETURN>
                     main                                    XREF[4]:     Entry Point(*),
                                                                          _start:00101081(*), 00102040,
                                                                          001020e8(*)
00101149 f3 0f 1e fa     ENDBR64
0010114d 55              PUSH       RBP
0010114e 48 89 e5        MOV        RBP,RSP
00101151 48 8d 3d        LEA        RDI,[s_Hello,_World!_00102004]              = "Hello, World!"
         ac 0e 00 00
00101158 e8 f3 fe        CALL       puts                                        int puts(char * __s)
         ff ff
0010115d b8 00 00        MOV        EAX,0x0
         00 00
00101162 5d              POP        RBP
00101163 c3              RET
```

Binary: world_c

Dynamically allocated string structure

x86

```
                    main.main                                       XREF[4]:     Entry Point(*),
                                                                                 runtime.main:00434ac7(c),
                                                                                 0049acce(c), 004c5cb8(*)
0049ac60 64 48 8b        MOV        RCX,qword ptr FS:[0xfffffff8]
         0c 25 f8
         ff ff ff
0049ac69 48 3b 61 10     CMP        RSP,qword ptr [RCX + 0x10]
0049ac6d 76 5a           JBE        LAB_0049acc9
0049ac6f 48 83 ec 58     SUB        RSP,0x58
0049ac73 48 89 6c        MOV        qword ptr [RSP + local_8],RBP
         24 50
0049ac78 48 8d 6c        LEA        RBP=>local_8,[RSP + 0x50]
         24 50
0049ac7d 48 8b 05        MOV        RAX,qword ptr [os.Stdout]                          = ??
         0c bd 0b 00
0049ac84 48 8d 0d        LEA        RCX,[go.itab.*os.File,io.Writer]                   =
         95 26 04 00
0049ac8b 48 89 0c 24     MOV        qword ptr [RSP]=>local_58,RCX=>go.itab.*os.File,i... =
0049ac8f 48 89 44        MOV        qword ptr [RSP + local_50],RAX
         24 08
0049ac94 48 8d 05        LEA        RAX,[DAT_004bf224]                                 = 48h    H
         89 45 02 00
0049ac9b 48 89 44        MOV        qword ptr [RSP + local_48],RAX=>DAT_004bf224       = 48h    H
         24 10
0049aca0 48 c7 44        MOV        qword ptr [RSP + local_40],0xe
         24 18 0e
         00 00 00
0049aca9 48 c7 44        MOV        qword ptr [RSP + local_38],0x0
         24 20 00
         00 00 00
0049acb2 0f 57 c0        XORPS      XMM0,XMM0
0049acb5 0f 11 44        MOVUPS     xmmword ptr [RSP + local_30[0]],XMM0
         24 28
0049acba e8 e1 82        CALL       fmt.Fprintf                                        undefined fmt.Fprintf(
         ff ff
```

Binary: world_go

Dynamically allocated string structure

x86

```
                main.main                                XREF[4]:   Entry Point(*),
                                                                    runtime.main:00434ac7(c),
                                                                    0049acce(c), 004c5cb8(*)
0049ac60 64 48 8b        MOV       RCX,qword ptr FS:[0xfffffff8]
         0c 25 f8
         ff ff ff
0049ac69 48 3b 61 10     CMP       RSP,qword ptr [RCX + 0x10]
0049ac6d 76 5a           JBE       LAB_0049acc9
0049ac6f 48 83 ec 58     SUB       RSP,0x58
0049ac73 48 89 6c        MOV       qword ptr [RSP + local_8],RBP
         24 50
0049ac78 48 8d 6c        LEA       RBP=>local_8,[RSP + 0x50]
         24 50
0049ac7d 48 8b 05        MOV       RAX,qword ptr [os.Stdout]                    = ??
         0c bd 0b 00
0049ac84 48 8d 0d        LEA       RCX,[go.itab.*os.File,io.Writer]             =
         95 26 04 00
0049ac8b 48 89 0c 24     MOV       qword ptr [RSP]=>local_58,RCX=>go.itab.*os.File,i... =
0049ac8f 48 89 44        MOV       qword ptr [RSP + local_50],RAX
         24 08
0049ac94 48 8d 05        LEA       RAX,[DAT_004bf224]                           = 48h    H
         89 45 02 00
0049ac9b 48 89 44        MOV       qword ptr [RSP + local_48],RAX=>DAT_004bf224 = 48h    H
         24 10
0049aca0 48 c7 44        MOV       qword ptr [RSP + local_40],0xe
         24 18 0e
         00 00 00
0049aca9 48 c7 44        MOV       qword ptr [RSP + local_38],0x0
         24 20 00
         00 00 00
0049acb2 0f 57 c0        XORPS     XMM0,XMM0
0049acb5 0f 11 44        MOVUPS    xmmword ptr [RSP + local_30[0]],XMM0
         24 28
0049acba e8 e1 82        CALL      fmt.Fprintf                                  undefined fmt.Fprintf(
         ff ff
```

```
                DAT_004bf224
004bf224 48      ??      48h    H
004bf225 65      ??      65h    e
004bf226 6c      ??      6Ch    l
004bf227 6c      ??      6Ch    l
004bf228 6f      ??      6Fh    o
004bf229 2c      ??      2Ch    ,
004bf22a 20      ??      20h
004bf22b 57      ??      57h    W
004bf22c 6f      ??      6Fh    o
004bf22d 72      ??      72h    r
004bf22e 6c      ??      6Ch    l
004bf22f 64      ??      64h    d
004bf230 21      ??      21h    !
004bf231 0a      ??      0Ah
```

Length

Binary: world_go

Dynamically allocated string structure

x86

- Search for these instructions and define strings

```
#x86
#LEA REG, [STRING_ADDRESS]
#MOV [ESP + ..], REG
#MOV [ESP + ..], STRING_SIZE
```

```
08208bdc 8d 05 0e        LEA        EAX,[DAT_0827de0e]
         de 27 08
08208be2 89 44 24 0c     MOV        dword ptr [ESP + local_10],EAX=>DAT_0827de0e
08208be6 c7 44 24        MOV        dword ptr [ESP + local_c],0x17
         10 17 00
```

```
#x86_64
#LEA REG, [STRING_ADDRESS]
#MOV [RSP + ..], REG
#MOV [RSP + ..], STRING_SIZE
```

```
0049ac94 48 8d 05        LEA        RAX,[DAT_004bf224]
         89 45 02 00
0049ac9b 48 89 44        MOV        qword ptr [RSP + local_48],RAX=>DAT_004bf224
         24 10
0049aca0 48 c7 44        MOV        qword ptr [RSP + local_40],0xe
         24 18 0e
         00 00 00
```

Binary: eCh0raix – x86, world_go

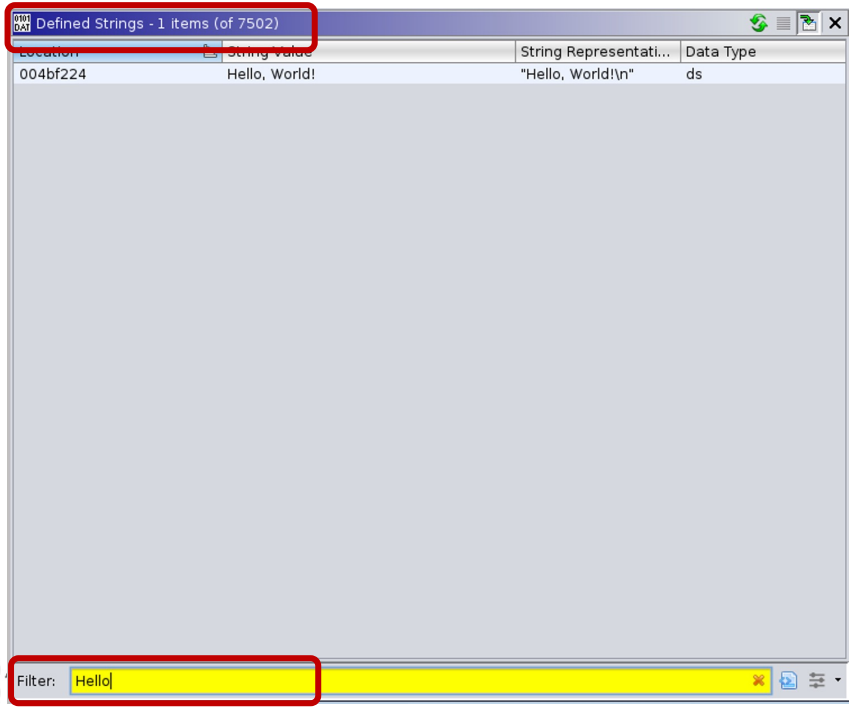
Dynamically allocated string structure

x86

- Results after executing the script

```
                    main.main                                       XREF[4]:     Entry Point(*),
                                                                                 runtime.main:00434ac7(c),
                                                                                 0049acce(c), 004c5cb8(*)
0049ac60 64 48 8b        MOV         RCX,qword ptr FS:[0xfffffff8]
         0c 25 f8
         ff ff ff
0049ac69 48 3b 61 10     CMP         RSP,qword ptr [RCX + 0x10]
0049ac6d 76 5a           JBE         LAB_0049acc9
0049ac6f 48 83 ec 58     SUB         RSP,0x58
0049ac73 48 89 6c        MOV         qword ptr [RSP + local_8],RBP
         24 50
0049ac78 48 8d 6c        LEA         RBP=>local_8,[RSP + 0x50]
         24 50
0049ac7d 48 8b 05        MOV         RAX,qword ptr [os.Stdout]                          = ??
         0c bd 0b 00
0049ac84 48 8d 0d        LEA         RCX,[go.itab.*os.File,io.Writer]                   =
         95 26 04 00
0049ac8b 48 89 0c 24     MOV         qword ptr [RSP]=>local_58,RCX=>go.itab.*os.File,i... =
0049ac8f 48 89 44        MOV         qword ptr [RSP + local_50],RAX
         24 08
0049ac94 48 8d 05        LEA         RAX,[s_Hello,_World!_004bf224]                     = "Hello, World!\n"
         89 45 02 00
0049ac9b 48 89 44        MOV         qword ptr [RSP + local_48],RAX=>s_Hello,_World!_0... = "Hello, World!\n"
         24 10
0049aca0 48 c7 44        MOV         qword ptr [RSP + local_40],0xe
         24 18 0e
         00 00 00
0049aca9 48 c7 44        MOV         qword ptr [RSP + local_38],0x0
         24 20 00
         00 00 00
0049acb2 0f 57 c0        XORPS       XMM0,XMM0
0049acb5 0f 11 44        MOVUPS      xmmword ptr [RSP + local_30[0]],XMM0
         24 28

                    s_Hello,_World!_004bf224                        XREF[2]:     main.main:0049ac94(*),
                                                                                 main.main:0049ac9b(*)
004bf224 48 65 6c        ds          "Hello, World!\n"
         6c 6f 2c
         20 57 6f ...
```



Binary: world_go

Dynamically allocated string structure

x86

- After executing our script the number of defined strings grew from 9719 to 11213

```
                    main.checkReadmeExists                          XREF[2]:     08208c3b(c),
                                                                                 main.init.0:08208cda(c)
08208bb0 65 8b 0d       MOV         ECX,dword ptr GS:[0x0]
         00 00 00 00
08208bb7 8b 89 fc       MOV         ECX,dword ptr [ECX + 0xfffffffc]
         ff ff ff
08208bbd 3b 61 08       CMP         ESP,dword ptr [ECX + 0x8]
08208bc0 76 74          JBE         LAB_08208c36
08208bc2 83 ec 1c       SUB         ESP,0x1c
08208bc5 c7 04 24       MOV         dword ptr [ESP]=>local_1c,0x0
         00 00 00 00
08208bcc 8b 44 24 20    MOV         EAX,dword ptr [ESP + param_1]
08208bd0 89 44 24 04    MOV         dword ptr [ESP + local_18],EAX
08208bd4 8b 44 24 24    MOV         EAX,dword ptr [ESP + param_2]
08208bd8 89 44 24 08    MOV         dword ptr [ESP + local_14],EAX
08208bdc 8d 05 0e       LEA         EAX,[DAT_0827de0e]
         de 27 08
08208be2 89 44 24 0c    MOV         dword ptr [ESP + local_10],EAX=>DAT_0827de0e
08208be6 c7 44 24       MOV         dword ptr [ESP + local_c],0x17
         10 17 00
         00 00
08208bee e8 dd c1       CALL        runtime.concatstring2
         e7 ff
```

```
                    main.checkReadmeExists                          XREF[2]:     08208c3b(c),
                                                                                 main.init.0:08208cda(c)
08208bb0 65 8b 0d       MOV         ECX,dword ptr GS:[0x0]
         00 00 00 00
08208bb7 8b 89 fc       MOV         ECX,dword ptr [ECX + 0xfffffffc]
         ff ff ff
08208bbd 3b 61 08       CMP         ESP,dword ptr [ECX + 0x8]
08208bc0 76 74          JBE         LAB_08208c36
08208bc2 83 ec 1c       SUB         ESP,0x1c
08208bc5 c7 04 24       MOV         dword ptr [ESP]=>local_1c,0x0
         00 00 00 00
08208bcc 8b 44 24 20    MOV         EAX,dword ptr [ESP + param_1]
08208bd0 89 44 24 04    MOV         dword ptr [ESP + local_18],EAX
08208bd4 8b 44 24 24    MOV         EAX,dword ptr [ESP + param_2]
08208bd8 89 44 24 08    MOV         dword ptr [ESP + local_14],EAX
08208bdc 8d 05 0e       LEA         EAX,[s_/README_FOR_DECRYPT.txt_0827de0e]
         de 27 08
08208be2 89 44 24 0c    MOV         dword ptr [ESP + local_10],EAX=>s_/README_FOR_DECRYPT.txt_0827de0e
08208be6 c7 44 24       MOV         dword ptr [ESP + local_c],0x17
         10 17 00
         00 00
08208bee e8 dd c1       CALL        runtime.concatstring2
         e7 ff
```

Binary: eCh0raix – x86

Dynamically allocated string structure

ARM – before executing the script

```
#ARM, 32-bit
#LDR REG, [STRING_ADDRESS_POINTER]
#STR REG, [SP, ..]
#MOV REG, STRING_SIZE
#STR REG, [SP, ..]
```

```
001e35bc 68 23 9f e5     ldr        r2,[PTR_DAT_001e392c]
001e35c0 10 20 8d e5     str        r2=>DAT_0025f560,[sp,#local_90]
001e35c4 44 20 a0 e3     mov        r2,#0x44
001e35c8 14 20 8d e5     str        r2,[sp,#local_8c]
001e35cc 18 00 8d e5     str        r0,[sp,#local_88]
001e35d0 1c 10 8d e5     str        r1,[sp,#local_84]
001e35d4 44 cc f9 eb     bl         runtime.concatstring3
```

Length

```
                         PTR_DAT_001e392c
001e392c 60 f5 25 00     addr       DAT_0025f560
```

```
                         DAT_0025f560                    XREF[2]:   main.main:001e35c0(*),
                                                                    001e392c(*)
0025f560 0d              ??         0Dh
0025f561 0a              ??         0Ah
0025f562 0d              ??         0Dh
0025f563 0a              ??         0Ah
0025f564 44              ??         44h    D
0025f565 6f              ??         6Fh    o
0025f566 20              ??         20h
0025f567 4e              ??         4Eh    N
0025f568 4f              ??         4Fh    O
0025f569 54              ??         54h    T
0025f56a 20              ??         20h
0025f56b 72              ??         72h    r
0025f56c 65              ??         65h    e
0025f56d 6d              ??         6Dh    m
0025f56e 6f              ??         6Fh    o
0025f56f 76              ??         76h    v
0025f570 65              ??         65h    e
0025f571 20              ??         20h
0025f572 74              ??         74h    t
0025f573 68              ??         68h    h
0025f574 69              ??         69h    i
0025f575 73              ??         73h    s
```

Binary: eCh0raix – ARM

Dynamically allocated string structure

ARM – after executing the script

```
#ARM, 32-bit
#LDR REG, [STRING_ADDRESS_POINTER]
#STR REG, [SP, ..]
#MOV REG, STRING_SIZE
#STR REG, [SP, ..]
```

Binary: eCh0raix – ARM

Dynamically allocated string structure

Challenges

- Different instruction sets
- Can be implemented in different ways within the same architecture
- Easy to break intentionally
- From Go 1.17 new way of passing function arguments and results using registers instead of the stack

```
                        DAT_0028bbff                            XREF[6]:    ddos.sshgo:001fd740(*),
                                                                            ddos.sshgo:001fd744(*),
                                                                            ddos.sshgo:001fd788(*),
                                                                            ddos.sshgo:001fd7a4(*),
                                                                            ddos.sshgo:001fd7c0(*),
                                                                            ddos.sshgo:001fd7dc(*)
0028bbff 6c          ??         6Ch    l
0028bc00 69          ??         69h    i
0028bc01 6e          ??         6Eh    n
0028bc02 75          ??         75h    u
0028bc03 78          ??         78h    x
0028bc04 5f          ??         5Fh    _
0028bc05 61          ??         61h    a
0028bc06 72          ??         72h    r
0028bc07 6d          ??         6Dh    m
```

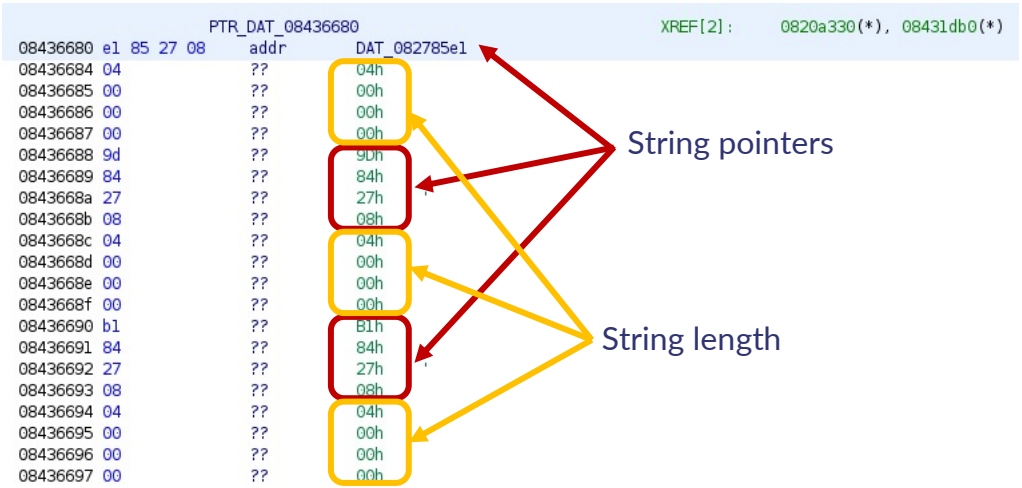
```
001fd734 21 01 80 d2     mov        param_2,#0x9
001fd738 e1 4b 00 f9     str        param_2,[sp, #local_c0]
001fd73c 62 04 00 d0     adrp       param_3,0x28b000
001fd740 42 fc 2f 91     add        param_3=>DAT_0028bbff,param_3,#0xbff
001fd744 e2 4f 00 f9     str        param_3=>DAT_0028bbff,[sp, #local_b8]
001fd748 e1 53 00 f9     str        param_2,[sp, #local_b0]
```

Binary: Kaiji – ARM

Statically allocated string structure

Idea

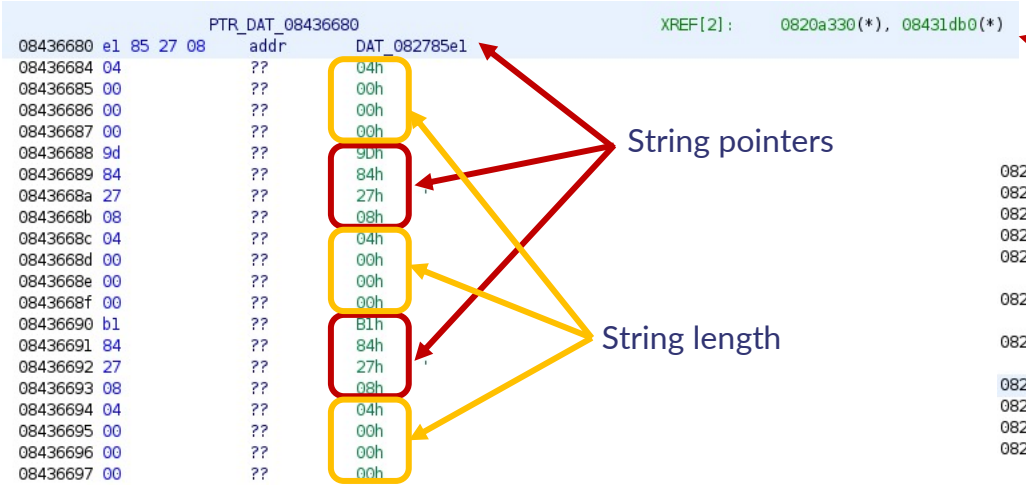
- Look for pointer to string followed by possible length value
- To eliminate FPs limit string length and search for printable characters only
- Check only in data sections
- Not architecture specific



Binary: eCh0raix – x86

Statically allocated string structure

Example – before executing the script



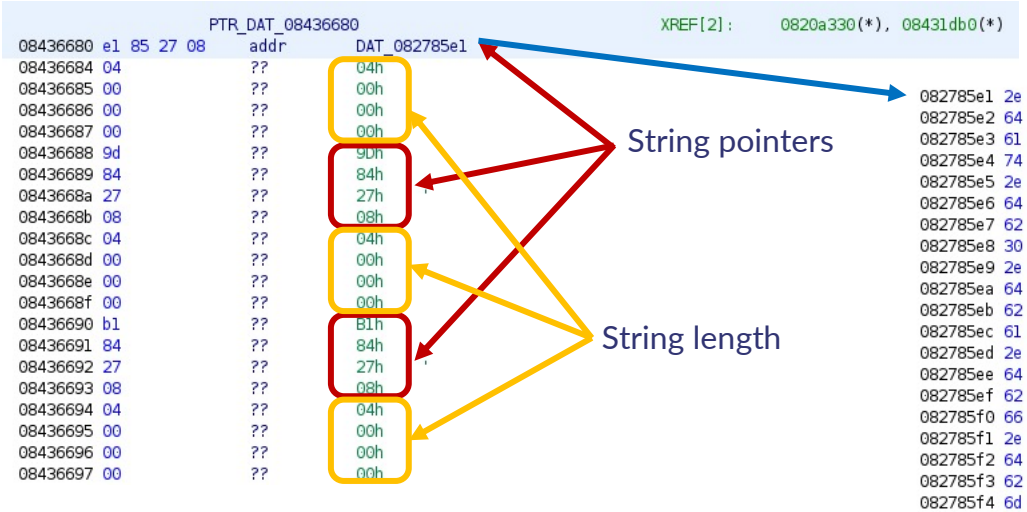
One pointer was successfully identified as it is directly referenced from the code

```
0820a30f 8b 44 24 20     MOV        EAX,dword ptr [ESP + 0x20]
0820a313 89 04 24        MOV        dword ptr [ESP],EAX
0820a316 8b 44 24 1c     MOV        EAX,dword ptr [ESP + 0x1c]
0820a31a 89 44 24 04     MOV        dword ptr [ESP + 0x4],EAX
0820a31e 8b 05 b0        MOV        EAX,dword ptr [PTR_PTR_DAT_08431db0]
         1d 43 08
0820a324 8b 0d b4        MOV        ECX,dword ptr [DAT_08431db4]
         1d 43 08
0820a32a 8b 15 b8        MOV        EDX,dword ptr [DAT_08431db8]
         1d 43 08
0820a330 89 44 24 08     MOV        dword ptr [ESP + 0x8],EAX=>PTR_DAT_08436680
0820a334 89 4c 24 0c     MOV        dword ptr [ESP + 0xc],ECX
0820a338 89 54 24 10     MOV        dword ptr [ESP + 0x10],EDX
0820a33c e8 df f0        CALL       FUN_08209420
         ff ff
```

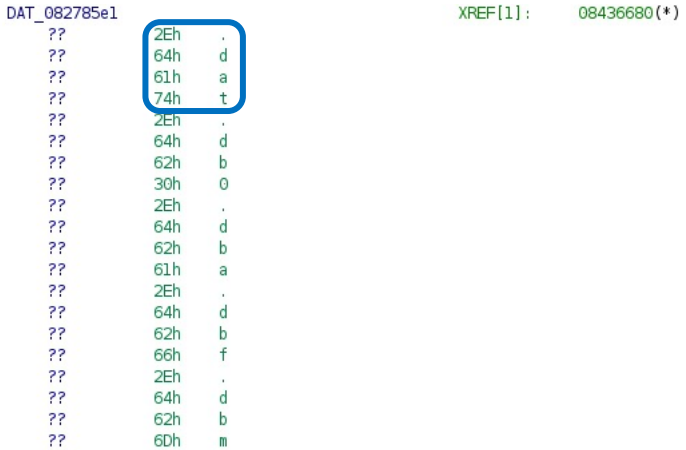
Binary: eCh0raix – x86

Statically allocated string structure

Example – before executing the script



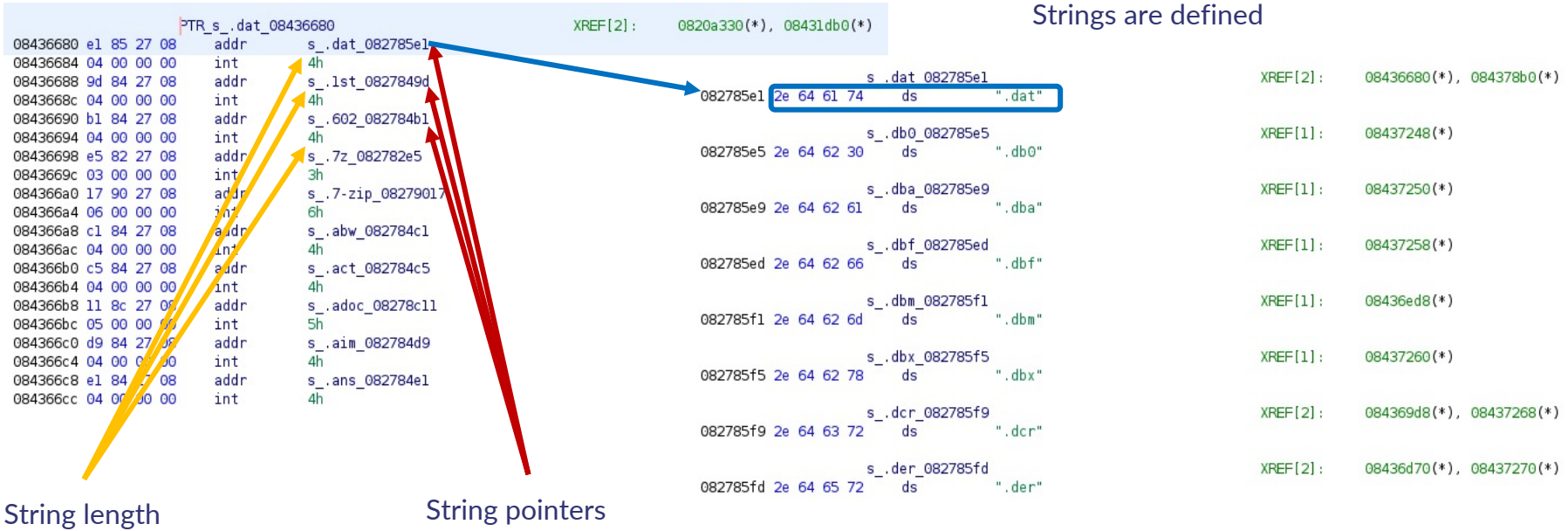
Strings are not defined



Binary: eCh0raix – x86

Statically allocated string structure

Example – after executing the script



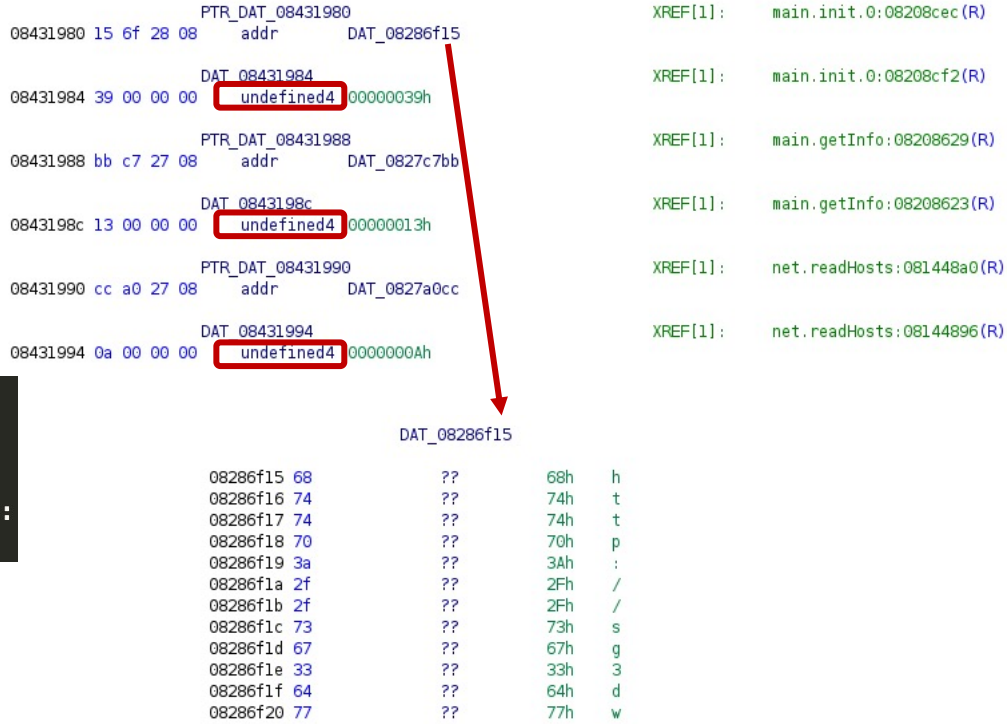
Binary: eCh0raix – x86

String recovery challenges

Falsely defined data types by Ghidra

- undefined4 or undefined8 (depends on pointer size)
- Already defined data types cannot be redefined
  (undifined4 and undifined8 are defined data types)
- First the data type has to be removed
- Then the new data type can be defined

```
if getDataAt(length_address) is not None:
    data_type = getDataAt(length_address).getDataType()
    #Remove undefined data to be able to create int.
    #Keep an eye on other predefined data types.
    if data_type.getName() in ["undefined4", "undefined8"]:
        removeData(getDataAt(length_address))
```

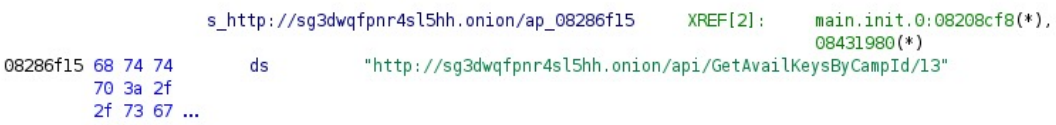


Binary: eCh0raix – x86

String recovery challenges

Falsely defined data types by Ghidra

- undefined4 or undefined8 (depends on pointer size)
- Already defined data types cannot be redefined
  (undifined4 and undifined8 are defined data types)
- First the data type has to be removed
- Then the new data type can be defined

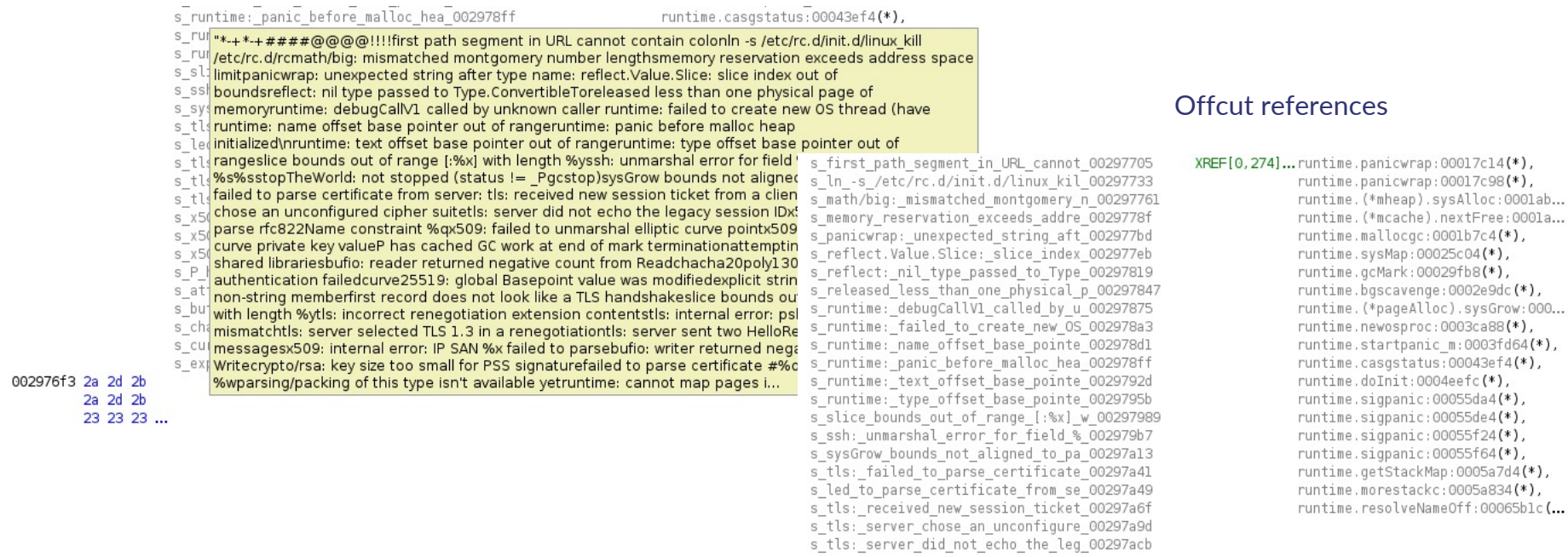


Binary: eCh0raix – x86

String recovery challenges

Falsely defined data types by Ghidra

- A large string blob (containing multiple strings) defined as one string



Offcut references

Binary: Kaiji – ARM

String recovery challenges

Falsely defined data types by Ghidra

- A large string blob (containing multiple strings) defined as one string

```
                    s_runtime:_panic_before_malloc_hea_002978ff
                    s_runtime:_text_offset_base_pointe_0029792d
                    s_runtime:_type_offset_base_pointe_0029795b
                    s_slice_bounds_out_of_range_[:%x]_w_00297989
                    s_ssh:_unmarshal_error_for_field_%_002979b7
                    s_sysGrow_bounds_not_aligned_to_pa_00297a13
                    s_tls:_failed_to_parse_certificate_00297a41
                    s_led_to_parse_certificate_from_se_00297a49
                    s_tls:_received_new_session_ticket_00297a6f
                    s_tls:_server_chose_an_unconfigure_00297a9d
                    s_tls:_server_did_not_echo_the_leg_00297acb
                    s_x509:_failed_to_parse_rfc822Name_00297af9
                    s_x509:_failed_to_unmarshal_ellipt_00297b27
                    s_x509:_invalid_elliptic_curve_pri_00297b55
                    s_P_has_cached_GC_work_at_end_of_m_00297b83
                    s_attempting_to_link_in_too_many_s_00297bb2
                    s_bufio:_reader_returned_negative_c_00297be1
                    s_chacha20poly1305:_message_authen_00297c10
                    s_curve25519:_global_Basepoint_val_00297c3f
                    s_explicit_string_type_given_to_no_00297c6e
002976f3 2a 2d 2b       ds          "*-+*-+####@@@@!!!!first path segment in URL cannot
         2a 2d 2b
         23 23 23 ...
```

Defined Strings - 10814 items

Location	String Value	Data Type	Byte Count	Offcut Reference Count
0022073d	certificateAuthorities	ds	23	1
00220ec1	ReplaceAllLiteralString	ds	24	1
00220ef5	responseMessageReceived	ds	24	1
00220f29	verifyServerCertificate	ds	24	1
00221561	hashForClientCertificate	ds	25	1
00221e1e	asn1:"explicit,tag:1"	ds	22	1
00221e53	handlePostHandshakeMessage	ds	27	1
00222552	secureRenegotiationSupported	ds	30	1
00222ebd	asn1:"optional,tag:2"	ds	23	1
00290069	ckunpa	ds	6	1
002903f7	queuefinalizer during GC	ds	24	1
00330cff	runtime.dropg	ds	14	1
00460248	-----END	ds	12	1
00460258	-----BEGIN	ds	16	1
0029bb9c	000102030405060708091011 1...	ds	969	2
002e9100	expand 32-byte k	ds	20	3
002e91a0	expand 32-byte k	ds	20	3
00293a08	3552713678800500929355621...	ds	170	4
0028b3b3	= is not mcount= minutes nallo...	ds	225	23
002976f3	*-+*-+####@@@@!!!!first pat...	ds	4517	95

Binary: Kaiji – ARM

Type extraction

Types

- Description for types is available within the binary
- Basic types: string, bool, numeric types (e.g. int8) etc.
- Composite types: pointer, struct, func, interface etc.
- https://golang.org/src/reflect/type.go

```
type miner.Process struct{
        pid int
        name string
        path string
        cmdline string
        buf []uint8
}
```

```
type exploit.exploiter interface {
        check(*exploit.Session) int
        init() []uint16
        run(*exploit.Session) bool
}
```

```
func(string, string, *tls.Config) (net.Conn, error)
```

```
// A Kind represents the specific kind of type that a Type represents.
// The zero Kind is not a valid kind.
type Kind uint

const (
        Invalid Kind = iota
        Bool
        Int
        Int8
        Int16
        Int32
        Int64
        Uint
        Uint8
        Uint16
        Uint32
        Uint64
        Uintptr
        Float32
        Float64
        Complex64
        Complex128
        Array
        Chan
        Func
        Interface
        Map
        Pointer
        Slice
        String
        Struct
        UnsafePointer
)
```

Type extraction

Example

- shell/miner.NewProcess function
- Call to runtime.newobject – memory allocation

```
006cab44 48 8b 4c 24 10     MOV     param_4,qword ptr [RSP + local_b0[8]]
006cab49 48 89 4c 24 58     MOV     qword ptr [RSP + local_60],param_4
006cab4e 48 8d 15 6b ce 0f 00  LEA  param_3,[DAT_007c79c0]                  = 50h    P
006cab55 48 89 14 24        MOV     qword ptr [RSP]=>local_b8,param_3=>DAT_007c79c0  = 50h    P
006cab59 e8 c2 54 d4 ff     CALL    runtime.newobject                       undefined runtime.newobject(unde...
006cab5e 48 8b 44 24 08     MOV     RAX,qword ptr [RSP + local_b0[0]]
006cab63 48 8b 8c 24 c0 00 00 00  MOV

                                    DAT_007c79c0                            XREF[2]:    shell/miner.NewProcess:006cab4e(...
                                                                                        shell/miner.NewProcess:006cab55(...
007c79c0 50        ??    50h    P
007c79c1 00        ??    00h
007c79c2 00        ??    00h
007c79c3 00        ??    00h
007c79c4 00        ??    00h
007c79c5 00        ??    00h
007c79c6 00        ??    00h                                                  ? -> 00400000
007c79c7 00        ??    00h
007c79c8 40        ??    40h    @
007c79c9 00        ??    00h
007c79ca 00        ??    00h
007c79cb 00        ??    00h
007c79cc 00        ??    00h
```

Binary: sysrv

Type extraction

rtype

CUJOAI

```
// rtype is the common implementation of most values.
// It is embedded in other struct types.
//
// rtype must be kept in sync with ../runtime/type.go:/^type._type.
type rtype struct {
        size       uintptr
        ptrdata    uintptr // number of bytes in the type that can
        hash       uint32  // hash of type; avoids computation in h
        tflag      tflag   // extra type information flags
        align      uint8   // alignment of variable with this type
        fieldAlign uint8   // alignment of struct field with this t
        kind       uint8   // enumeration for C
        // function for comparing objects of this type
        // (ptr to object A, ptr to object B) -> ==?
        equal     func(unsafe.Pointer, unsafe.Pointer) bool
        gcdata    *byte    // garbage collection data
        str       nameOff // string form
        ptrToThis typeOff // type for pointer to this type, may be
}
```

```
                        LONG_007c79c0
007c79c0 50 00 00       long        50h
         00 00 00
         00 00
007c79c8 40 00 00       long        40h
         00 00 00
         00 00
007c79d0 4d 48 2e 23    int         232E484Dh
007c79d4 07             db          7h
007c79d5 08             db          8h
007c79d6 08             db          8h
007c79d7 19             db          19h
007c79d8 00 5a f9       addr        DAT_00f95a00
         00 00 00
         00 00
007c79e0 4c 5a 84       addr        DAT_00845a4c
         00 00 00
         00 00
007c79e8 85 15 01 00    int         11585h
007c79ec e0 29 05 00    int         529E0h
```

Binary: sysrv

Type extraction

rtype

CUJOAI

```
// rtype is the common implementation of most values.
// It is embedded in other struct types.
//
// rtype must be kept in sync with ../runtime/type.go:/^type._type.
type rtype struct {
        size       uintptr
        ptrdata    uintptr // number of bytes in the type that can
        hash       uint32  // hash of type; avoids computation in h
        tflag      tflag   // extra type information flags
        align      uint8   // alignment of variable with this type
        fieldAlign uint8   // alignment of struct field with this t
        kind       uint8   // enumeration for C
        // function for comparing objects of this type
        // (ptr to object A, ptr to object B) -> ==?
        equal     func(unsafe.Pointer, unsafe.Pointer) bool
        gcdata    *byte   // garbage collection data
        str       nameOff // string form
        ptrToThis typeOff // type for pointer to this type, may be
}
```

```
                        LONG_007c79c0
007c79c0 50 00 00       long        50h
         00 00 00
         00 00
007c79c8 40 00 00       long        40h
         00 00 00
         00 00
007c79d0 4d 48 2e 23    int         232E484Dh
007c79d4 07             db          7h
007c79d5 08             db          8h
007c79d6 08             db          8h
007c79d7 19             db          19h
007c79d8 00 5a f9       addr        DAT_00f95a00
         00 00 00
         00 00
007c79e0 4c 5a 84       addr        DAT_00845a4c
         00 00 00
         00 00
007c79e8 85 15 01 00    int         11585h
007c79ec e0 29 05 00    int         529E0h
```

```
0073f585 01             ??          01h
0073f586 00             ??          00h
0073f587 0e             ??          0Eh
0073f588 2a 6d 69       ds          "*miner.Process"
         6e 65 72
         2e 50 72 ...
```

Binary: sysrv

Type extraction

Finding type descriptions

- Moduledata - records information about the layout of the executable image
- type, etype – memory location storing type information
- Typelinks - stores offsets of type descriptions (from type)

- PE and ELF binary differences
  - ELF - .typelink section, type = .rodata section
  - PE – parse moduledata to find the necessary addresses
- Version differences
  - Pclnatb header updates (from 1.2, changes in 1.16, 1.18)
  - Moduldata structure update
    (from 1.5, changes in 1.7, 1.8, 1.10, 1.16)
  - Type name struct update (1.18)

```
type moduledata struct {
        pcHeader     *pcHeader
        funcnametab  []byte
        cutab        []uint32
        filetab      []byte
        pctab        []byte
        pclntable    []byte
        ftab         []functab
        findfunctab  uintptr
        minpc, maxpc uintptr

        text, etext           uintptr
        noptrdata, enoptrdata uintptr
        data, edata           uintptr
        bss, ebss             uintptr
        noptrbss, enoptrbss   uintptr
        end, gcdata, gcbss    uintptr
        types, etypes         uintptr
        rodata                uintptr
        gofunc                uintptr // go.func.*

        textsectmap []textsect
        typelinks   []int32 // offsets from types
        itablinks   []*itab
```

Binary: sysrv

Type extraction

Example – executing our script

```
type miner.Process struct{
        pid int
        name string
        path string
        cmdline string
        buf []uint8
}
miner.Process
```

```
                    DAT_007c79c0
007c79c0 50         ??          50h    P
007c79c1 00         ??          00h
007c79c2 00         ??          00h
007c79c3 00         ??          00h
007c79c4 00         ??          00h
007c79c5 00         ??          00h
007c79c6 00         ??          00h
007c79c7 00         ??          00h
007c79c8 40         ??          40h    @
007c79c9 00         ??          00h
007c79ca 00         ??          00h
```

```
007c79c0 50         ??          50h    P
007c79c1 00         ??          00h
007c79c2 00         ??          00h
007c79c3 00         ??          00h
007c79c4 00         ??          00h
007c79c5 00         ??          00h
007c79c6 00         ??          00h
007c79c7 00         ??          00h
007c79c8 40         ??          40h    @
007c79c9 00         ??          00h
007c79ca 00         ??          00h
```

```
006cab44 48 8b 4c 24 10    MOV     param_4,qword ptr [RSP + local_b0[8]]
006cab49 48 89 4c 24 58    MOV     qword ptr [RSP + local_60],param_4
006cab4e 48 8d 15 6b ce 0f 00  LEA     param_3,[miner.Process]                    = 50h    P
006cab55 48 89 14 24       MOV     qword ptr [RSP]=>local_b8,param_3=>miner.Process  = 50h    P
006cab59 e8 c2 54 d4 ff    CALL    runtime.newobject                          undefined runtime.newobject(unde...
006cab5e 48 8b 44 24 08    MOV     RAX,qword ptr [RSP + local_b0[0]]
006cab63 48 8b 8c 24 c0 00 00 00  MOV     param_4,qword ptr [RSP + param_7]
```

Binary: sysrv

Type extraction

Example II. – before script execution

```
                        main.getInfo                                    XREF[2]:     08208ba2(c),
                                                                                     main.init.0:08208cff(c)
082085b0 65 8b 0d       MOV         ECX,dword ptr GS:[0x0]
         00 00 00 00
082085b7 8b 89 fc       MOV         ECX,dword ptr [ECX + 0xfffffffc]
         ff ff ff
082085bd 8d 44 24 f4    LEA         EAX=>local_c,[ESP + -0xc]
082085c1 3b 41 08       CMP         EAX,dword ptr [ECX + 0x8]
082085c4 0f 86 d3       JBE         LAB_08208b9d
         05 00 00
082085ca 81 ec 8c       SUB         ESP,0x8c
         00 00 00
082085d0 c7 84 24       MOV         dword ptr [ESP + param_3],0x0
         98 00 00
         00 00 00 ...
082085db c7 84 24       MOV         dword ptr [ESP + param_4],0x0
         9c 00 00
         00 00 00 ...
082085e6 c7 84 24       MOV         dword ptr [ESP + param_5],0x0
         a0 00 00
         00 00 00 ...
082085f1 c7 84 24       MOV         dword ptr [ESP + param_6],0x0
         a4 00 00
         00 00 00 ...
082085fc 8d 05 20       LEA         EAX,[DAT_0824bd20]                                  = 10h
         bd 24 08
08208602 89 04 24       MOV         dword ptr [ESP]=>local_8c,EAX=>DAT_0824bd20         = 10h
08208605 e8 66 97       CALL        runtime.newobject                                   undefined runtime.newobject(
         e4 ff
0820860a 8b 44 24 04    MOV         EAX,dword ptr [ESP + local_88]
0820860e 89 44 24 40    MOV         dword ptr [ESP + local_4c],EAX
08208612 8d 0d 23       LEA         ECX,[s_tcp_08278423]                                = "tcp"
         84 27 08
08208618 89 0c 24       MOV         dword ptr [ESP]=>local_8c,ECX=>s_tcp_08278423       = "tcp"
0820861b c7 44 24       MOV         dword ptr [ESP + local_88],0x3
         04 03 00
         00 00
08208623 8b 0d 8c       MOV         ECX,dword ptr [INT_0843198c]                        = 13h
         19 43 08
08208629 8b 15 88       MOV         EDX,dword ptr [PTR_s_192.99.206.61:65000_08431988]  = 0827c7bb
         19 43 08
0820862f 89 54 24 08    MOV         dword ptr [ESP + local_84],EDX=>s_192.99.206.61:6...  = "192.99.206.61:65000"
08208633 89 4c 24 0c    MOV         dword ptr [ESP + local_80],ECX
08208637 c7 44 24       MOV         dword ptr [ESP + local_7c],0x0
         10 00 00
         00 00
```

```
                        DAT_0824bd20                                    XREF[2]:     main.getInfo:082085fc(*)
                                                                                     main.getInfo:08208602(*)
0824bd20 10             ??          10h
0824bd21 00             ??          00h
0824bd22 00             ??          00h
0824bd23 00             ??          00h
0824bd24 0c             ??          0Ch
0824bd25 00             ??          00h
0824bd26 00             ??          00h
0824bd27 00             ??          00h
0824bd28 15             ??          15h
0824bd29 e7             ??          E7h
0824bd2a c0             ??          C0h
0824bd2b 27             ??          27h     '
0824bd2c 07             ??          07h
0824bd2d 04             ??          04h
0824bd2e 04             ??          04h
0824bd2f 19             ??          19h
0824bd30 28             ??          28h     (                                   ? -> 0820c828
0824bd31 c8             ??          C8h
0824bd32 20             ??          20h
0824bd33 08             ??          08h
0824bd34 fc             ??          FCh                                         ? -> 082ba0fc
0824bd35 a0             ??          A0h
```

Binary: eCh0raix - x86

Type extraction

Example II. – after script execution

```
                        main.getInfo                                    XREF[2]:     08208ba2(c),
                                                                                     main.init.0:08208cff(c)
082085b0 65 8b 0d       MOV         ECX,dword ptr GS:[0x0]
         00 00 00 00
082085b7 8b 89 fc       MOV         ECX,dword ptr [ECX + 0xfffffffc]
         ff ff ff
082085bd 8d 44 24 f4    LEA         EAX=>local_c,[ESP + -0xc]
082085c1 3b 41 08       CMP         EAX,dword ptr [ECX + 0x8]
082085c4 0f 86 d3       JBE         LAB_08208b9d
         05 00 00
082085ca 81 ec 8c       SUB         ESP,0x8c
         00 00 00
082085d0 c7 84 24       MOV         dword ptr [ESP + param_3],0x0
         98 00 00
         00 00 00 ...
082085db c7 84 24       MOV         dword ptr [ESP + param_4],0x0
         9c 00 00
         00 00 00 ...
082085e6 c7 84 24       MOV         dword ptr [ESP + param_5],0x0
         a0 00 00
         00 00 00 ...
082085f1 c7 84 24       MOV         dword ptr [ESP + param_6],0x0
         a4 00 00
         00 00 00 ...
082085fc 8d 05 20       LEA         EAX,[main.Info]                                 = 10h
         bd 24 08
08208602 89 04 24       MOV         dword ptr [ESP]=>local_8c,EAX=>main.Info        = 10h
08208605 e8 66 97       CALL        runtime.newobject                               undefined runtime.newobject(unde...
         e4 ff
0820860a 8b 44 24 04    MOV         EAX,dword ptr [ESP + local_88]
0820860e 89 44 24 40    MOV         dword ptr [ESP + local_4c],EAX
08208612 8d 0d 23       LEA         ECX,[s_tcp_08278423]                            = "tcp"
         84 27 08
08208618 89 0c 24       MOV         dword ptr [ESP]=>local_8c,ECX=>s_tcp_08278423   = "tcp"
0820861b c7 44 24       MOV         dword ptr [ESP + local_88],0x3
         04 03 00
         00 00
08208623 8b 0d 8c       MOV         ECX,dword ptr [INT_0843198c]                    = 13h
         19 43 08
08208629 8b 15 88       MOV         EDX,dword ptr [PTR_s_192.99.206.61:65000_08431988]  = 0827c7bb
         19 43 08
0820862f 89 54 24 08    MOV         dword ptr [ESP + local_84],EDX=>s_192.99.206.61:6...  = "192.99.206.61:65000"
08208633 89 4c 24 0c    MOV         dword ptr [ESP + local_80],ECX
08208637 c7 44 24       MOV         dword ptr [ESP + local_7c],0x0
         10 00 00
         00 00
```

```
type main.Info struct{
        RsaPublicKey string
        Readme string
}
main.Info                                                   XREF[2]:     main.getInfo:082085fc(*),
                                                                         main.getInfo:08208602(*)
bd20 10         ??          10h
bd21 00         ??          00h
bd22 00         ??          00h
bd23 00         ??          00h
bd24 0c         ??          0Ch
bd25 00         ??          00h
bd26 00         ??          00h
bd27 00         ??          00h
bd28 15         ??          15h
bd29 e7         ??          E7h
bd2a c0         ??          C0h
bd2b 27         ??          27h     '
bd2c 07         ??          07h
bd2d 04         ??          04h
bd2e 04         ??          04h
bd2f 19         ??          19h
bd30 28         ??          28h     (                                   ? -> 0820c828
bd31 c8         ??          C8h
bd32 20         ??          20h
bd33 08         ??          08h
bd34 fc         ??          FCh                                         ? -> 082ba0fc
bd35 a0         ??          A0h
```

```
type main.Info struct{
        RsaPublicKey string
        Readme string
}
```

Binary: eCh0raix - x86

Challenges

Problems to solve

- Continuous version changes
- Go version identification
    - Currently string based

```
> strings go_mal_v118_elf | grep "go1\."
go1.18.1
/usr/lib/go-1.18/src/vendor/golang.org/x/crypto/internal/poly1305/bits_go1.13.go
go1.18.1
```

```
> strings sys.x86_64_unp | grep "go1\."
        stack=[_ABSTIME_ACLITEM_CSTRING_MACADDR_NUMERIC_POLYGON_REGOPER_REGPROC_REGROLE_REG
TYPE_RELTIME_TSQUERY_TSRANGE_VARCHAR_wpnoncea1111111a1234567a1b2c3d4a1s2d3f4a@123456aA12345
6aB123456ab123456abc#1234abc123!@abc@1234abc@123@abcd1234abcd@123abcdefghaddress admin123ad
min520adminsunalefsym;angelicaangrtvb;angzarr;asdf1234asdfasdfasdfghjkasympeq;b42207_1backs
im;bad instbeEfFgGvbecause;bemptyv;between;bigcirc;bigodot;bigstar;bnequiv;boxplus;ccupssm;
cemptyv;cgocheckcirscir;cisco123coloneq;congdot;continuecudarrl;cudarrr;culARRp;curarrm;dat
abasedbkarow;ddagger;ddotseq;default:demptyv;diamond;digamma;disableddotplus;dump.rdbdwangl
e;echo -n epsilon;eqcolon;equivDD;fc00::/7filenameflushallftpadminfunctiongesdoto:go1.10.7g
tquest;gtrless;harrcir;hijackedhost keyhttp/1.1if-matchif-rangeinfinityintprod;invalid io_e
rrorisindot;it123456larrbfs;larrsim;lbrksld;lbrkslu;ldrdhar;lesdoto;lessdot;lessgtr;lesssim
;locationloopbacklotimes;lozenge;ltquest;luruhar;maltese;minusdu;mysql123napprox;natural;ne
arrow;nexists;nistp256nistp384nistp521no anodeno-cacheno_proxynotinva;notinvb;notinvc;notni
va;notnivb;notnivc;npolint;npreceq;nsqsube;nsqsupe;nsubset;nsucceq;nsupset;nvinfin;nvltrie;
nvrtrie;nwarrow;olcross;omicron;orderof;orslope;os:Linuxp@55w0rdp@ssw0rdp@sswordpa55wordpas
s1234passw0rdpassw0rdpasswordpertenk;planckh;pluscir;plussim;plustwo;postgresprecsim;q1w2e3
r4qQ123456qq111111qq112233qq123123qq123456quatint;questeq;qwe123!@qwe123..qwer1234qwerasdfq
werqwerrarrbfs;rarrsim;rbrksld;rbrkslu;rdldhar;readfromreadlinkrealine;recvfromredis - redi
s123responseroot.123root123!root1234root@123rootrootrotimes;ruluhar;runnableruntime.scaveng
esearrow;sendfileshutdownsignal: simplus;simrarr;socket:[strconv.subedot;submult;subplus;su
brarr;succsim;supdsub;supedot;suphsol;suphsub;suplarr;supmult;supplus;swarrow;testusertext/
xmltimeout:timezonetopfork;triplus;tritime;unixgramunknown(uparrow;upsilon;uwangle;vzigzag;
weblogicxx123456xxxx1234zaq1@WSXzigrarr;zxcv1234 (forced) (normal) -> node= blocked= defers
c= in use)
```

Other researcher's work

Links

IDA Pro

- https://github.com/sibears/IDAGolangHelper
- https://github.com/strazzere/golang_loader_assist
- https://github.com/sentinelabs/alphagolang

radare2 / Cutter

- https://github.com/f0rki/r2-go-helpers
- https://github.com/JacobPimental/r2-gohelper/blob/master/golang_helper.py
- https://github.com/CarveSystems/gostringsr2

Binary Ninja

- https://github.com/f0rki/bn-goloader

Ghidra

- https://github.com/felberj/gotools
- https://github.com/ghidraninja/ghidra_scripts/blob/master/golang_renamer.py

Other

- https://go-re.tk/

Files used during the presentation

Hashes

CUJOAI

File name	SHA-256
world.c	761301bb14ea3b678650fc1b6da768f009387ee726712e291d57e2d7985613d0
world.go	7cb3316a7b89eb996e8dbb0d0fb277136cd588cc54642f3b09aa84cd177cb3a2
world_c	76a5c4ef9277b97660f2c412e67ff2c3826e699913db86cd333e8f1d4fb5b8a3
world_c_strip	486a93362a6a8bc3b449fd6ba07656011c687ed31a19091c329a434bff4d75bb
world_go	d0d4781de4ffd5fbe18d59328eccd373a782eecdf55a2c5199b7dc6598cfb99e
world_go_strip	9b975bd9406a8b79a414195e184be0c82bb1593979577f0344c797f9bcd4ad0b
world_go.exe	9e36291f5fc67fdb9e5e17b636d34b39f2cc39f328916a9012a8f8d545e9d0c8
world_go_strip.exe	c5b66623942a0cea6df30541e92afe93172be7bb4dbdd42a1fa354e9edd79a1d
world_go_println	fa00f5ad2aa79a6245a28516bc285ae8c36f075d818787aadff6f3e850e2ec5c
eCh0raix - x86	154dea7cace3d58c0ceccb5a3b8d7e0347674a0e76daffa9fa53578c036d9357
eCh0raix - ARM	3d7ebe73319a3435293838296fbb86c2e920fd0ccc9169285cc2c4d7fa3f120d
Kaiji - x86_64	f4a64ab3ffc0b4a94fd07a55565f24915b7a1aaec58454df5e47d8f8a2eec22a
Kaiji - ARM	3e68118ad46b9eb64063b259fca5f6682c5c2cb18fd9a4e7d97969226b2e6fb4
sysrv	c543f137a9e9380203ab12b29662b10810afe7e10c2af24b3b0cf0c3669193a1

References, additional reading

Other Go malware research

CUJOAI

- https://rednaga.io/2016/09/21/reversing_go_binaries_like_a_pro/
- https://2016.zeronights.ru/wp-content/uploads/2016/12/GO_Zaytsev.pdf
- https://carvesystems.com/news/reverse-engineering-go-binaries-using-radare-2-and-python/
- https://www.pnfsoftware.com/blog/analyzing-golang-executables/
- https://github.com/strazzere/golang_loader_assist/blob/master/Bsides-GO-Forth-And-Reverse.pdf
- https://github.com/radareorg/r2con2020/blob/master/day2/r2_Gophers-AnalysisOfGoBinariesWithRadare2.pdf
- https://securelist.com/extracting-type-information-from-go-binaries/104715/

CUJOAI

FIRST
June 2022

Dorka Palotay
Senior Threat Researcher, CUJO AI
@pad0rka

CUJO AI Labs
https://github.com/getCUJO/ThreatIntel
@CujoaiLabs