How Do We Fight Against Evolving Go Language Malware? Practical Techniques to Improve Analytical Skills

JSAC 2023

FFRI Security, Inc.
Tsubasa Kuwabara

Tsubasa Kuwabara

- FFRI Security, Inc.
  - Malware Analysis
  - Other security services
- Ghidra and x64dbg plugin development
  - Developed plug-ins for analyzing Go language binaries, etc.
- SecHack365, Security Camp 2019 Alumni
- Security Camp 2021 Lecturer

Table of Contents

How Do We Fight Against Evolving Go Language Malware? Practical Techniques to Improve Analytical Skills

- Current Status and Issues
- Basic Analysis
  - For those with limited experience in Go language malware analysis
- Advanced Analysis
  - For those with experience in Go language malware analysis

GO LANGUAGE MALWARE CURRENT STATUS AND ISSUES

Advantages of the Go Language in Malware Development

- Go Language
  - Easy to describe
  - Extensive library
  - Cross-compile

- Advantages for Malware Developers
  - Easy to develop
    - OSS libraries are often used in malware
  - Easy to attack multiple platforms

Future of Go Language Malware

- Multi-platform attacks
  - ElectroRAT *1
    - RATs steal digital wallet keys, etc.
    - Activities for Windows/Linux/macOS
  - Chaos *2
    - Versatile malware that exploits vulnerabilities, etc.
    - Exists for ARM/Intel/MIPS/PowerPC

- About the future
  - Many more are appearing in 2022.
    - Chaos, Nerbian, Agenda, etc.
  - Go language malware will be observed in the future

*1 https://www.intezer.com/blog/research/operation-electrorat-attacker-creates-fake-companies-to-drain-your-crypto-wallets/
*2 https://blog.lumen.com/chaos-is-a-go-based-swiss-army-knife-of-malware/

Problems with Go Malware Analysis

- Existing malware is often C/C++ binaries
- Go language binaries and C/C++ binaries are very different
  - The amount of functions is enormous
    - Even Hello World has more than 1000 functions.
    - Windows APIs and other APIs are called dynamically.
      - Difficult to understand functionality of functions
  - Unique data structure
    - Strings are represented as structs instead of null-terminated
    - Unique types such as interface{} are used
  - Unique calling convention
    - Different registers, etc. used for arguments and return values

Hello World Functions

Functions - 1063 items

It's actually about 1400, but it's not resolved.

			89
FUN_00401060	...	undefi...	1357
FUN_004015c0	...	undefi...	2104
FUN_00401e00	...	undefi...	27
FUN_00401e20	...	undefi...	17
FUN_00401e40	...	undefi...	9
FUN_00402020	...	undefi...	110
FUN_00402380		undefi	571
FUN_004024e0			345

ElectroRAT would be over 8,000.

Hello World string

```
                    s_Hello_World_Join_ControlMeetei_M_0049a709
0049a709 48 65 6c ...   ds        "Hello World\nJoin_ControlMe
```

Hello World function call

```
0048243b 48 8d 05 ...   LEA   RAX,[PTR_DAT_004b9178]
00482442 48 8d 0d ...   LEA   RCX,[DAT_0049a709]
00482449 bf 0c 00 ...   MOV   EDI,0xc
0048244e 31 f6          XOR   ESI,ESI
00482450 45 31 c0       XOR   R8D,R8D
00482453 4d 89 c1       MOV   R9,R8
00482456 e8 05 95 ...   CALL  FUN_0047b960
```

Response and problems with the tool

- Existing tool features
  - Resolving Function Names
    - Quite difficult to analyze without knowing this.
  - String extraction, etc.

- Issue
  - Difficult to use existing tools for a long time
    - Go language can change its structure with version upgrades
      - Newer versions may not be able to retrieve function names, etc.
  - Malware with obfuscation, such as malware ChaChi, exists.
    - Obfuscators for the Go language are used

- Difficult to solve by simply using existing tools as is
  - Requires modifications as appropriate

Contents

- Basic Go Language Malware Analysis
  - Go Language Specific Structures
  - Analysis Flow
  - Case Studies with Malware

- Advanced Go Language Malware Analysis
  - Data in Go language binaries referenced by existing tools
  - Supports for Go language version upgrades by modifying existing tool
  - Obfuscation measures

GO LANGUAGE MALWARE BASIC ANALYSIS

Go-Specific Structures

- Introduction to Go language specific structures before introducing analysis methods

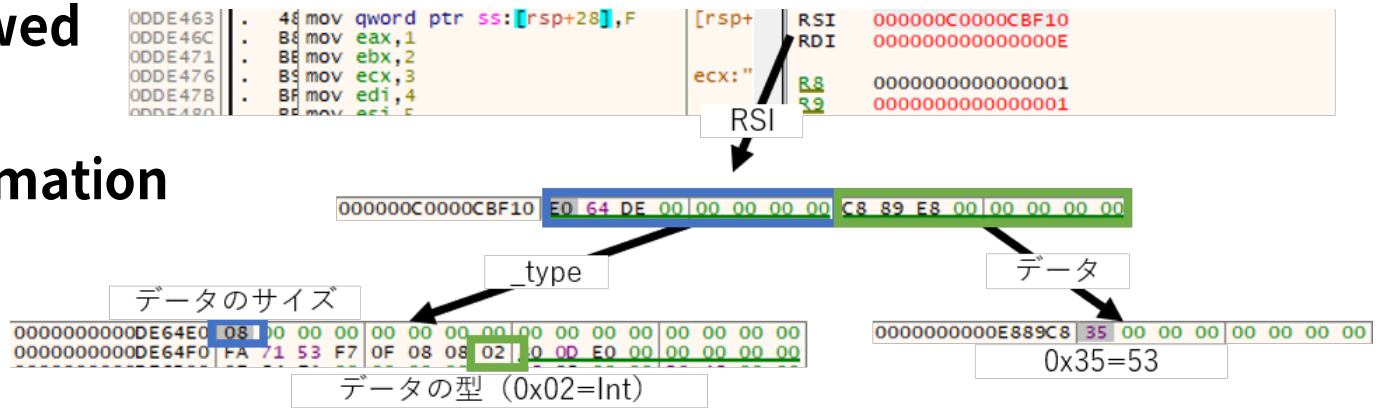
- Data Structure Related
  - string
  - interface{ }
  - slice
  - map
- Function
  - calling conventions

string & interface{ }

- string
  - Structure
    - __data: pointer to string
      - Not null terminated.
    - __len: String length
      - Its size is equivalent to sizeof(__data)

```
struct string {
    char* __data;
    int32 or int64 __len;
}
```

- interface{ }: Multiple data types allowed
  - Structure
    - tab: Pointer to data type information
    - data: Pointer to data



slice

- slice: dynamically resizable array
- Structure
  - array: Pointer to array
  - len: Length of array
    - Its size is equivalent to sizeof(array)
  - cap: Length of memory-allocated array
    - Its size is equivalent to sizeof(array)

```
struct slice {
    void* array;
    int32 or int64 len;
    int32 or int64 cap;
}
```

- Function manipulating slice
  - func growslice(et *_type, old slice, cap int) slice *1
    - Copies the slice specified in the second argument to a slice that holds more memory than the size specified in the third argument.
    - Adding a value to the slice is done on the return value of this growslice.
    - Argument structure is different starting from Go 1.20

- Variable-length arguments are represented by the same structure as slice
  - func Command(name string, arg .... .string) *Cmd *2
    - Command("cmd", "/C", "bin") == Command("cmd ", [2]string{"/C", "bin"})

*1 https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/slice.go#L178
*2 https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/os/exec/exec.go#L271

map

- map: associative array

- Function manipulating map (if key is a string)
  - func mapaccess2_faststr(t *maptype, h *hmap, ky string) (unsafe.Pointer, bool) *1
    - Specify the type of map as the first argument, the map to access as the second argument, and the key as the third argument
    - Return the value corresponding to the key and success or failure
  - func mapassign_faststr(t *maptype, h *hmap, s string) unsafe.Pointer *2
    - Specify the type of map as the first argument, the target map as the second argument, and the key as the third argument
    - Assign a value to the return value

*1 https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/map_faststr.go#L108
*2 https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/map_faststr.go#L203

Calling Conventions

- Difference from existing in Windows amd64
  - If i386, the stack is used.

	Microsoft x64 calling convention	Go Language (>=Go 1.17)	Go Language (<Go 1.17)
argument	rcx,rdx,r8,r9,stack	rax,rbx,rcx,rdi,rsi,r8,r9,r10,r11,stack	stack
return value	rax	same as above	same as above

- Argument and return value assignment
  - Go 1.17 or higher & amd64: arguments and return values are used in order from rax
  - Go 1.16 and below ¦ i386: Return values are used from the end of the stack used by the argument
- Definition in Go language source code
  - Defined in "paramIntReg<architecture name>" below.
    - https://raw.githubusercontent.com/golang/go/go1.19.1/src/cmd/compile/internal/ssa/opGen.go

Tool

- https://github.com/mooncat-greenpy/Ghidra_GolangAnalyzerExtension
  - Resolves function names
  - Resolves bytes and registers used in arguments and return values
  - Extracts data type names and field information
  - Extracts strings
  - Annotates names of the source code file corresponding to the assembly
  - Annotates source code line number corresponding to the assembly

Without the tool

```
puVar2 = (undefined8 *)FUN_00410500((ulonglong)&DAT_004a6bab);
puVar2[1] = 6;
if (_DAT_005924c0 == 0) {
  *puVar2 = &DAT_004a70e3;
}
```

With the tool

```
runtime.mapassign_faststr(&datatype.Map.map[string]string,extraout_RAX_03,"go",2);
extraout_RAX_04[1] = (char *)0x6;
if (_DAT_005924c0 == 0) {
  *extraout_RAX_04 = "golang";
}
```

- Subsequent Ghidra screenshots show results using GhidraAnalyzerExtension

Analysis Flow

- Check main (main.main) function
  - Take a brief look at assembly and decompile results
    - Functions to be called
  - Check the init function used for initialization if it exists.
- Check function name, file name, and structure sequence
  - Obvious doubtful names
  - OSS libraries used
- Estimate functions from the above information and confirm its estimation
  - Is the function as used or not or as estimated?
- Analyze in detail

Supplement
If you don't know the main function, trace it from the entry point.
Please refer to the following blog for the details
https://engineers.ffri.jp/entry/2022/04/11/141131

The init function is explained in the Appendix

Case Study: ElectroRAT

- It was active on Windows, Linux, and macOS

- Analysis Flow
  - Install and run GolangAnalyzerExtension on Ghidra
  - Check main.main
  - Estimate its functions from lists of function and file names
    - Checking structures is not covered in this talk
  - Verify the above estimation
  - Analyze in detail

ElectroRAT: Check main.main

- Excerpts of Notable Function Call
  - net/http.(*Client).Get
  - io/ioutil.readAll
  - main.registerUser
  - main.setAutostart
  - main.StartKeyLogger
  - main.socketConnect
- Guessing functions
  - Automatic execution functionality is available and main functions are based on communication with C2
- main.init is not covered in this talk
  - map is created, etc.

Decompiled main.main (tools applied without manual modification)

ElectroRAT: Check function name and file name

- Anticipated malware features
  - **Information theft**
    - github.com/gorilla/websocket
    - main.uploadFile
    - main.uploadFolder
  - **Automatic startup setting**
    - github.com/ProtonMail/go-autostart
    - main.HideFile
    - main.copyAppToStartDir
  - Terminal information acquisition
  - Software used to note content typed (typically to steal passwords)
  - Obtain screenshots
- Simplified check of the caller of the estimated function and whether the estimation is correct.
  - Only the bolded above is confirmed this time

File name list (Window → GolangAnalyzerExtension)

functions | filenames | datatypes

Filename
C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/screenshot.go
C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/socket.go
C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/startup.go
C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/uploadFile.go
C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/uploadFolder.go
C:/Users/exec/go/src/github.com/denisbrodbeck/machineid/id.go
C:/Users/exec/go/src/github.com/denisbrodbeck/machineid/id_windows.go
C:/Users/exec/go/src/github.com/gorilla/websocket/client.go

Function name list

functions | filenames | datatypes

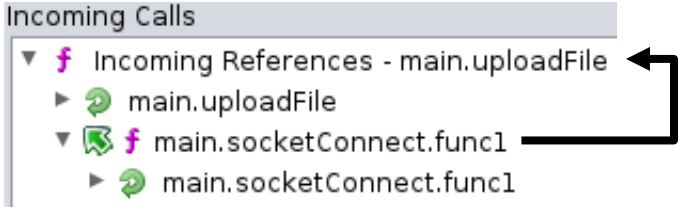
Location	Function Name	Args Size	Size
007ae860	main.getMachineID	16	176
007aebc0	main.getOsInfo	24	384
007aee50	main.getProcessList	32	752
007b03d0	main.getUserPath	16	208
007ae500	main.HideFile	32	224
007b3160	main.init	0	1328
007aed40	main.killProcessWindows	16	272
007ae910	main.main	0	688
007af140	main.registerUser	16	1056
007af560	main.RunBinary	64	384
007b0900	main.setAutostart	0	560
007afa90	main.socketConnect	16	2368

ElectroRAT: Information Theft Caller

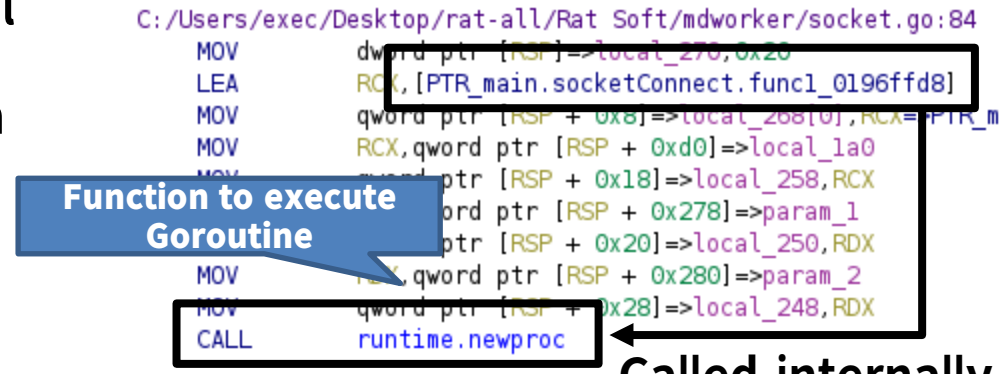
- gorilla/websocket.(*Dialer).Dial
  - (*Dialer).Dial is called first
  - Caller: main.socketConnect
    - Called from main.main
- main.upload(File¦Folder)
  - Caller: both main.socketConnect.func1
- main.socketConnect.func1
  - Cannot figure out from Call Trees
    - Because Goroutine is used to perform parallel processing
      - The function to be executed is passed via arguments
  - Caller: main.socketConnect
    - Search from address references, etc.

Call Trees in main.socketConnect

Call Trees in main.uploadFile



main.socketConnect function



Called internally

ElectroRAT: Information Theft Process

- The following will be added to the map
  - Key: "file".
  - Value: Argument of main.uploadFile
    - File path to upload
- Information theft by sending files
  - SetFiles to specify a map containing file paths
    - Transmission file settings
  - Send externally with Execute
    - POST command

Decompile result of main.uploadFile

map type

Object of map

key

```
106     runtime.mapassign_faststr
107             (&datatype.Map.map[string]string,local_148,"file",4,in_stack_ffffffffff
108     in_stack_fffffffffffe68[1] = param_4;
109     if (_runtime.writeBarrier == 0) {
110       *in_stack_fffffffffffe68 = param_3;
111     }
```

The third and fourth arguments are substituted likely file path

Function to create an object of a specified type

Type of object to communicate with

Object of map

```
 40     runtime.newobject(&datatype.Struct.resty.Request,request_obj)

116     gopkg.in/resty%2ev1.(*Request).SetFiles(request_obj,local_148,pcVar8);
```

Resty.Request

POST" string

```
128     gopkg.in/resty%2ev1.(*Request).Execute
129             (pcVar8,"POST",4,in_stack_fffffffffffe80,in_stack_fffffffffffe88,pc
130              in_stack_fffffffffffe80);
```

ElectroRAT: Auto Startup Settings

- Auto start setting to files created by main.copyAppToStartDir
  - Using OSS
    - https://github.com/ProtonMail/go-autostart

Decompile result of main.setAutostart

```
53                  /* C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/startup.go:69 */
    github.com/ProtonMail/go-autostart.(*App).IsEnabled(&app,in_stack_rsp+8);
    if ((char)in_stack_rsp+8 == '\0') {
                    /* C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/startup.go:70 */
      github.com/ProtonMail/go-autostart.(*App).Enable(&app,in_stack_rsp+8,in_stack_rsp+10);
```

Confirmation of success or failure of IsEnabled

- go-autostart
  - Using cgo
    - Go language to C language
  - A shortcut file is created in the startup folder using COM

Source code for go-autostart

```
uint64_t CreateShortcut(char *shortcutA, char *path, char *args) {
        IShellLink*   pISL;
        IPersistFile* pIPF;
        HRESULT       hr;

        CoInitializeEx(NULL, COINIT_MULTITHREADED);

        // Shortcut filename: convert ANSI to unicode
        WORD shortcutW[MAX_PATH];
        int nChar = MultiByteToWideChar(CP_ACP, 0, shortcutA, -1, shortcutW, MAX_PATH);

        hr = CoCreateInstance(&CLSID_ShellLink, NULL, CLSCTX_INPROC_SERVER, &IID_IShellLink, (LPV
```

Examples of OSS libraries used by ElectroRAT

- github.com/ProtonMail/go-autostart
  - Automatic startup setting
- github.com/gorilla/websocket
  - Socket communications
- github.com/matishsiao/goInfo
  - Obtain terminal information
- github.com/mitchellh/go-ps
  - Get process list
- etc.

- All support Windows, Linux, and macOS.
- Read the source code for the OSS library process

Frequently used OSS libraries

- Libraries for multiple platforms
  - github.com/denisbrodbeck/machineid
    - Obtain ID to identify the terminal
  - github.com/shirou/gopsutil
    - Obtain process and system-related information
  - github.com/gorilla/websocket
    - Socket communications
  - github.com/kardianos/service
    - Provide service operation
- Libraries for specific platforms, requiring a lot of work to implement in the Go language
  - github.com/go-ole/go-ole
    - Windows COM wrapper
  - github.com/lxn/win
    - Windows API Wrapper

Details

- Now we have a rough idea of the malware's functionality
- Based on these findings, a more detailed analysis is performed
  - Only the following topic is covered

- Communication with C2
  - main.socketConnect is supposed to handle the main malware
    - Because socket communication is performed by main.socketConnect after preliminaries such as infection notification to C2 by main.registerUser and automatic startup setting by main.setAutostart

Analysis of main.socketConnect

- Main processing
  - Establish communication
    - Creates Conn objects required for communication
  - Call main.socketConnect.func1
    - Passes a Conn object as the second argument

Conn object is created

```
007afeb9 e8 52 d8 ...      CALL        github.com/gorilla/websocket.(*Dialer).Dial
                                       omission
                      C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/socket.go:84
007affa7 c7 04 24 ...      MOV         dword ptr [RSP]=>local_270,0x20
007affae 48 8d 0d ...      LEA         RCX,[PTR_main.socketConnect.func1_0196ffd8]
007affb5 48 89 4c ...      MOV         qword ptr [RSP + 0x8]=>local_268[0],RCX=>PTR_main.socketConnect.func1_01
007affba 48 8b 8c ...      MOV         RCX,qword ptr [RSP + 0xd0]=>local_1a0
007affc2 48 89 4c ...      MOV         qword ptr [RSP + 0x18]=>local_258,RCX
007affc7 48 8b 94 ...      MOV         RDX,qword ptr [RSP + 0x278]=>param_1
007affcf 48 89 54 ...      MOV         qword ptr [RSP + 0x20]=>local_250,RDX
007affd4 48 8b 94 ...      MOV         RDX,qword ptr [RSP + 0x280]=>param_2
007affdc 48 89 54 ...      MOV         qword ptr [RSP + 0x28]=>local_248,RDX
007affe1 e8 9a a7 ...      CALL        runtime.newproc
```

Conn object.
Set as 2nd argument
Arguments of functions executed by Goroutine are described in the Appendix

IP address string
(Arguments of main.socketConnect)

Function to execute Goroutine

main.socketConnect.func1

- Process Flow
  - Receive command from C2
  - Parse command
  - Execute command
  - Send results to C2

Receive instructions & parse command

- Received data is converted to main.Command
  - Type field is treated as a command

Command structure (Data Type Manager)

Structure Editor - main.Command (170cb5ea1a6b4af3c27

Offset	Length	Mnemonic	DataType	Name
0	16	string.conflict	string.conflict	Type
16	8	ulonglong	ulonglong	UID
24	104	struct { Folder string; FileName string; Fil	struct { Fold...	Data

main.socketConnect.func1 function

```
007b17dc 48 8b 84 ...   MOV     RAX,qword ptr [RSP + 0x200]=>param_2
                        MOV     qword ptr [RSP]=>local_1f0,RAX
                        CALL    github.com/gorilla/websocket.(*Conn).ReadMessage
                                omission
007b18d3 48 89 04 24    MOV     qword ptr [RSP],RAX
                        LEA     RAX,[datatype.Ptr.*main.Command]
                        MOV     qword ptr [RSP + 0x8],RAX
                        MOV     RCX,qword ptr [RSP + 0x110]
007b18eb 48 89 4c ...   MOV     qword ptr [RSP + 0x10],RCX
                        CALL    encoding/json.(*Decoder).Decode
                        C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/socket.go:99
                        MOV     RAX,qword ptr [RSP + 0x110]
007b18fd 48 8b 48 08    MOV     RCX,qword ptr [RAX + 0x8]
007b1901 48 89 8c ...   MOV     qword ptr [RSP + 0x88],RCX
007b1909 48 8b 10       MOV     RDX,qword ptr [RAX]
007b190c 48 89 94 ...   MOV     qword ptr [RSP + 0x100],RDX
                                omission
                        C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/socket.go:203
                        LAB_007b1d0b
007b1d0b 0f 85 70 ...   JNZ     LAB_007b2381
                        MOV     RBX,"mmoc nuR"
                        CMP     qword ptr [RDX],RBX
                        JNZ     LAB_007b2378
                                word ptr [RDX + 0x8],"na"
                        JNZ     LAB_007b236f
007b1d30 80 7a 0a 64    CMP     byte ptr [RDX + 0xa],'d'
007b1d34 0f 84 61 ...   JZ      LAB_007b229b
                        C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/socket.go:99
007b1d3a 48 83 f9 0b    CMP     RCX,0xb
```

Use the Argument Conn.
Get incoming data

Decoder Object
Retain incoming data

Type of
&main.Command

Object of
&main.Command

Type
string

Command
decision
Run command

Execution of instructions & transmission of results

- Extracts a Command string from the main.Command structure and executes it.
- Convert results to JSON format and send to external devices

main.Command structure

Structure Editor - main.Command (170cb5ea1a6b4af3c27

Offset	Length	Mnemonic	DataType	Name
0	16	string.conflict	string.conflict	Type
16	8	ulonglong	ulonglong	UID
24	104	struct { Folder string; FileName string; Fil	struct { Fold...	Data

Data field structure

Structure Editor - struct { Folder string; FileName string; FilePath string; Name strin

Offset	Length	Mnemonic	DataType	Name
0	16	string.conflict1	string.conflict1	Folder
16	16	string.conflict1	string.conflict1	FileName
32	16	string.conflict1	string.conflict1	FilePath
48	16	string.conflict1	string.conflict1	Name
64	16	string.conflict1	string.conflict1	Command
80	16	string.conflict1	string.conflict1	FolderPath
96	8	longlong	longlong	Port

Offset from the beginning of main.Command: 0x58

main.socketConnect.func1 function

```
C:/Users/exe                                                    rker/socket.go:204
LAB_007b229b
007b229b 48 8b 48 60     MOV     RCX,qword ptr [RAX + 0x60]
007b229f 48 8b 50 58     MOV     RDX,qword ptr [RAX + 0x58]
                         MOV     qword ptr [RSP]=>local_1f0,RDX
                         MOV     qword ptr [RSP + 0x8]=>local_1f0[8],RCX
007b22ac e8 2f b7 ...    CALL    main.ExecConsole
007b22b1 48 8b 44        MOV     RAX,qword ptr [RSP + 0x18]=>local_1e0[8]
                         MOV     RCX,qword ptr [RSP + 0x10]=>local_1e0[0]

                                 omission

007b230f e8 6c 23 ...    CALL    encoding/json.Marshal
007b2314 48 8b 44 ...    MOV     RAX,qword ptr [RSP + 0x18]=>local_1e0[8]
007b2319 48 8b 4c ...    MOV     RCX,qword ptr [RSP + 0x10]=>local
007b231e 48 8b 54 ...    MOV     RDX,qword ptr [RSP + 0x20]=>local
C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/
007b2323 48 8b 9c ...    MOV     RBX,qword ptr [RSP + 0x200]=>param_2
007b232b 48 89 1c 24     MOV     qword ptr [RSP]=>local_1f0,RBX
007b232f 48 c7 44 ...    MOV     qword ptr [RSP + 0x8]=>local_1f0[8],0x1
007b2338 48 89 4c ...    MOV     qword ptr [RSP + 0x10]=>local_1e0[0],RCX
007b233d 48 89 44 ...    MOV     qword ptr [RSP + 0x18]=>local_1e0[8],RAX
007b2342 48 89 54 ...    MOV     qword ptr [RSP + 0x20]=>local_1d0,RDX
007b2347 e8 74 1c ...    CALL    github.com/gorilla/websocket.(*Conn).WriteMessage
```

RAX=&main.Command

Offset: 0x58

Get Command field

Execute the Command and return the result

Return value execution result

JSON format

Streamline analysis by comparing samples

- Introduce the comparison of the samples to be analyzed with existing samples and samples observed within the same incident

- Identical process exists
  - Analysis can be shortened
- Similar but different processes exist
  - Understanding Attack Trends
  - Investigation of the cause of infection of new samples when existing samples are protected

- Go language binaries allow you to get the file name and line number of the source code, so you can follow the differences in detail.

Comparison of old and new samples

- Compare the following samples
  - Left: e9b83d5cdefd4486b32a927d7505cdeebb43e6977759ba069d9373e46ca7d0f2 (new)
  - Right: 170cb5ea1a6b4af3c27358ba267a1309ed5118481619fc874f717262cb91fb77 (old)

functions | filenames | datatypes

Filenames are facilitated for each target environment of samples.

Filename	Filename
/home/exec/Desktop/mdworker/bin_linux.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/bin_windows.go
/home/exec/Desktop/mdworker/console.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/chrome_windows.go ⊖
⊕ /home/exec/Desktop/mdworker/downloadFile.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/console_windows.go
/home/exec/Desktop/mdworker/folderContent.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/executeBinary.go ⊖
/home/exec/Desktop/mdworker/hidefile.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/folderContent_windows.go
/home/exec/Desktop/mdworker/keycodes.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/hidefile_windows.go
/home/exec/Desktop/mdworker/machineid.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/keycodes.go
/home/exec/Desktop/mdworker/mdworker.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/keylogger_windows.go ⊖
/home/exec/Desktop/mdworker/osinfo.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/machineid.go
/home/exec/Desktop/mdworker/processKill.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/mdworker.go
/home/exec/Desktop/mdworker/processList.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/osinfo.go
/home/exec/Desktop/mdworker/registerUser.go	C:/Users/exec/Desktop/rat-all/Rat Soft/mdworker/processKill.go
/home/exec/Desktop/mdworker/re	processList.go
/home/exec/Desktop/mdworker/sc	registerUser.go

Some files have been added or deleted. See the differences in commands on the next page.

Command changes between samples

- **Analyze mainly the sections changed.**

Compare the number of lines in the socket.go file in which the command is implemented

(computer) command	Number of rows in e9b83d... (new)	Number of rows in 170cb5... (old)
Get folder content	130~ 133	100~ 103
Keylogger	**135~ 149**	**105~ 118**
Screenshot	151~ 164	119~ 128
Camera photo		129~ 134
Processes list	166~ 169	135~ 137
Download file	171~ 179	138~ 145
Download folder	181~ 189	147~ 153
Add file	**191~ 205**	
Delete file	207~ 215	154~ 161
Kill process	217~ 225	162~ 169
Chrome passwords		**170~ 176**
Run service	227~ 235	177~ 202
Run command	237~ 245	203~ 210

keylogger_windows.go was deleted but keylogger command exists

Added command related to downloadFile.go is added

Deleted command related to chrome_windows.go is deleted

GO LANGUAGE MALWARE ADVANCED ANALYSIS

Go Version Upgrade

- Tools work well suddenly stop working well
  - Version upgrades to the Go language change the data that the tool references
  - Example: ElectroRAT can be analyzed, but newer Chaos cannot be analyzed, etc.

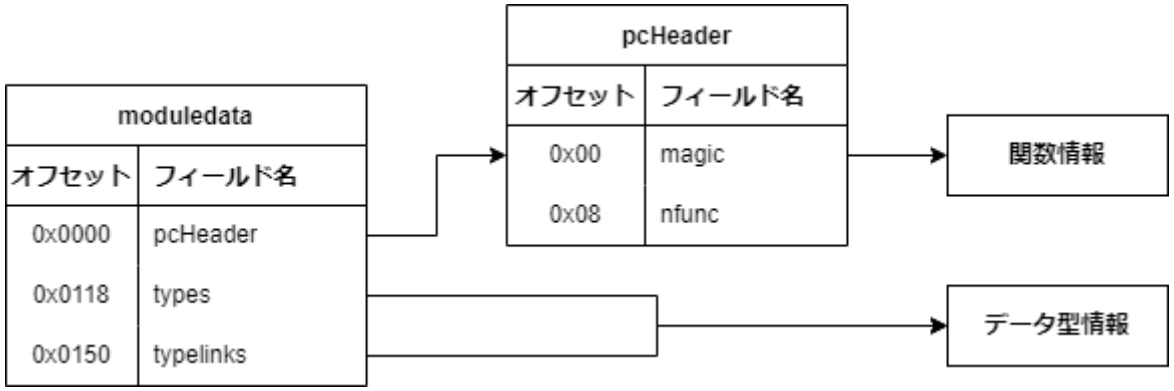
- Contents will be covered in this presentation
  - Explanation of metadata structure of Go language binaries
    - Information used by existing tools
    - Changes in past version upgrades
  - Tool modification case study

Introduction of Go Binary Metadata

- Data that the tool often refers to
  - Function Information
    - Contains information such as function names
    - Used for displaying stack traces, etc.
  - Data information
    - Includes data names and field information
    - Used to specify types in object creation, interface{}, and map

Overview of Metadata Composition

- Structure to manage the metadata
  - moduledata
    - Maintains link to pcHeader and data type information
  - pcHeader
    - Maintains links to function information



Only some fields in the binary for 64-bit are shown, and the same applies to subsequent figures.

Metadata Components

- moduledata *
  - pcHeader
    - Pointer to pcHeader
  - text
    - Pointer to code
    - Basically identical to .text section
  - types
    - Base address for type information
  - typelinks
    - Array (slice) of offsets from types to type information

```go
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

	ptab []ptabEntry

	pluginpath string
	pkghashes  []modulehash
```

* https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/symtab.go#L415-L457

Metadata Components

- moduledata *
  - pcHeader
    - Pointer to pcHeader
  - text
    - Pointer to code
    - Basically identical to .text section
  - types
    - Base address for type information
  - typelinks
    - Array (slice) of offsets from types to type information

```
415  type moduledata struct {
416          pcHeader     *pcHeader
417          funcnametab  []byte
418          cutab        []uint32
424          minpc, maxpc uintptr
425
426          text, etext          uintptr
431          end, gcdata, gcbss   uintptr
432          types, etypes        uintptr
433          rodata               uintptr
434          gofunc               uintptr // go.func.*
435
436          textsectmap []textsect
             typelinks   []int32 // offsets from types
442          pluginpath string
443          pkghashes  []modulehash
```

Offset changes due to field additions
Go 1.7 , Go 1.8 , Go 1.16 , Go 1.17 , Go 1.18 , Go 1.20

Adding types
Go 1.7

Change from pointer to offset
Go 1.7

* https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/symtab.go#L415-L457

Metadata Components

- pcHeader *
  - magic, pad1, pad2
    - Fixed value
  - ptrSize
    - Pointer Size
  - nfunc
    - Number of functions
  - textStart
    - Same as moduledata.text
  - funcnameOffset
    - Offset from pcHeader to function name sequence
  - pclnOffset
    - Offset from pcHeader to array containing link to function information

```
395 type pcHeader struct {
396         magic          uint32  // 0xFFFFFFF0
397         pad1, pad2     uint8   // 0,0
398         minLC          uint8   // min instruction size
399         ptrSize        uint8   // size of a ptr in bytes
400         nfunc          int     // number of functions in the module
401         nfiles         uint    // number of entries in the file tab
402         textStart      uintptr // base for function entry PC offsets in this modul
403         funcnameOffset uintptr // offset to the funcnametab variable from pcHeader
404         cuOffset       uintptr // offset to the cutab variable from pcHeader
405         filetabOffset  uintptr // offset to the filetab variable from pcHeader
406         pctabOffset    uintptr // offset to the pctab variable from pcHeader
407         pclnOffset     uintptr // offset to the pclntab variable from pcHeader
408 }
```

* https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/symtab.go#L395-L408

Metadata Components

- pcHeader *
  - magic, pad1, pad2
    - Fixed value
  - ptrSize
    - Pointer Size
  - nfunc
    - Number of functions
  - textStart
    - Same as moduledata.text
  - funcnameOffset
    - Offset from pcHeader to function name sequence
  - pclnOffset
    - Offset from pcHeader to array containing link to function information

Change magic
Go 1.16, Go 1.18, Go 1.20

```
395 type pcHeader struct {
396         magic          uint32  // 0xFFFFFFF0
397         pad1, pad2     uint8   // 0,0
398         minLC          uint8   // min instruction size
399         ptrSize        uint8   // size of a ptr in bytes
400         nfunc          int     // number of functions in the module
401         nfiles         uint    // number of entries in the file tab
402         textStart      uintptr // base for function entry PC offsets in this modul
403                                                                          tHeader
404                                                                          r
405                                                                          der
406                                                                          r
407         pclnOffset     uintptr // offset to the pclntab variable from pcHeader
408 }
```

Offset changes due to field additions
Go 1.16 , Go 1.18

Reference method change
Go 1.16

* https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/symtab.go#L395-L408

Locating Metadata

- Use the section
  - pcHeader is placed at the beginning of the .gopclntab section
  - .gopclntab section may not exist in Windows
    - When "-ldflags="-w -s" option is used

- Use magic in pcHeader
  - The first magic field has the value "0xfffffff0".
    - Error occurs when this field is written
      - pclnOffset, etc. can be written
  - Explore this value

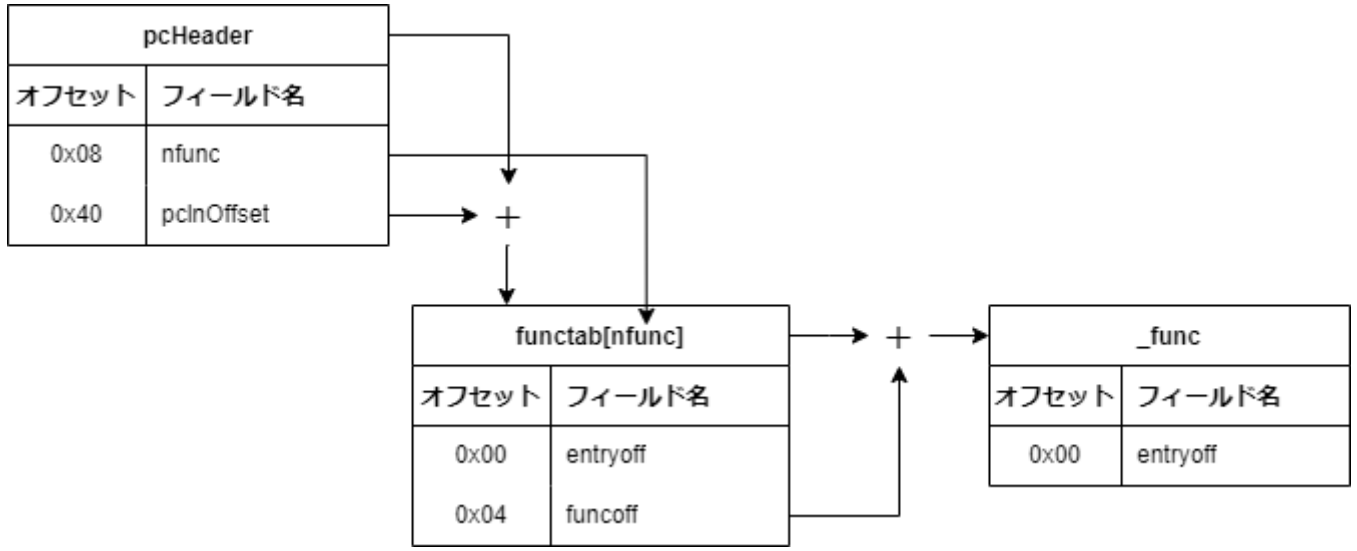
Go 1.19

- moduledata can be searched from pointers on pcHeader

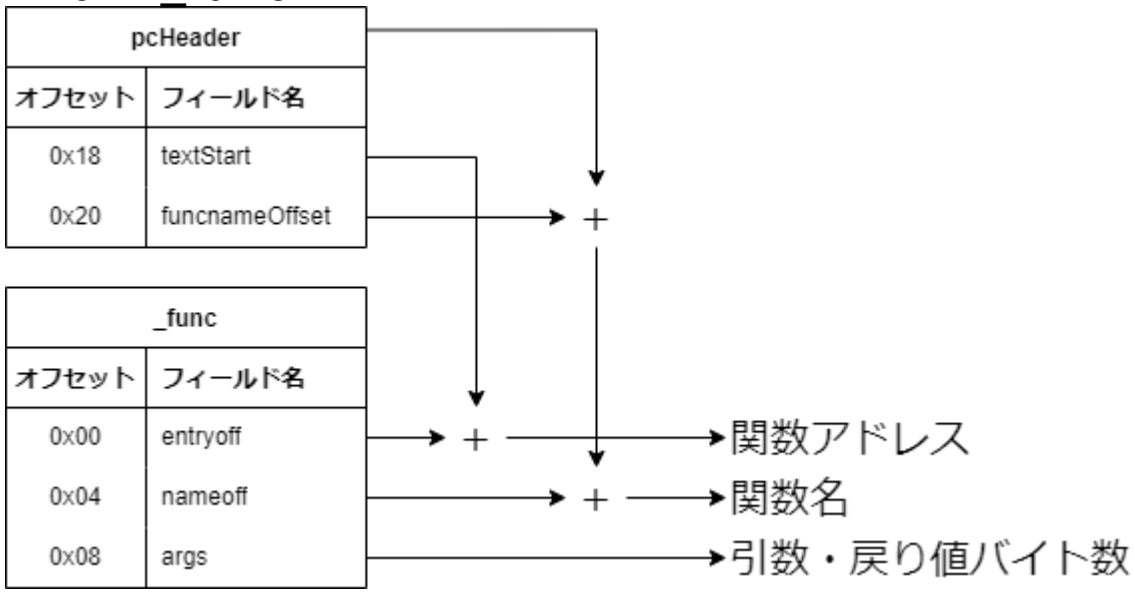
Composition of function information

- Flow of retrieving the following information from pcHeader
  - Function address
  - Function name
  - Argument and return bytes

Flow to obtain _func that holding function information



Flow of getting function information from _func



Components of Function Information

- functab *
  - entryoff
    - Offset to the first address of the function
    - Relative to pcHeader.textStart
  - funcoff
    - Offset to function information
    - Relative to the head of functab array

```
562  type functab struct {
563          entryoff uint32 // relative to runtime.text
564          funcoff  uint32
565  }
```

* https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/symtab.go#L562-L565

Components of Function Information

- functab *
  - entryoff
    - Offset to the first address of the function
    - Relative to pcHeader.textStart
  - funcoff
    - Offset to function information
    - Relative to the head of functab array

Change from pointer to offset
Go 1.18

```
562  type functab struct {
563          entryoff uint32 // relative to runtime.text
564          funcoff  uint32
565  }
```

Change in size
Go 1.18

Change of base address
Go 1.16

* https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/symtab.go#L562-L565

Components of Function Information

- _func *
  - entryoff
    - Offset to the first address of the function
    - Relative to pcHeader.textStart
  - nameoff
    - Offset to function name
    - pcHeader address + pcHeader.funcnameOffset + _func.nameoff
  - args
    - Number of bytes used in arguments and return values
  - pcsp, pcfile, pcln
    - Information to retrieve stack size, file name, and line number corresponding to the assembly
    - How to retrieve the information will be published later in our blog

```
869  type _func struct {
870          entryoff uint32 // start pc, as offset from moduledata.text/pcHeader
871          nameoff  int32  // function name
872
873          args        int32  // in/out args size
874          deferreturn uint32 // offset of start of a deferreturn call instruct
875
876          pcsp      uint32
877          pcfile    uint32
878          pcln      uint32
879          npcdata   uint32
880          cuOffset  uint32 // runtime.cutab offset of this function's CU
881          funcID    funcID // set for certain special runtime functions
882          flag      funcFlag
883          _         [1]byte // pad
884          nfuncdata uint8   // must be last, must end on a uint32-aligned bour
885  }
```

* https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/runtime2.go#L869-L885

Components of Function Information

- _func *
  - entryoff
    - Offset to the first address of the function
    - Relative to pcHeader.textStart
  - nameoff
    - Offset to function name
    - pcHeader address + pcHeader.funcnameOffset + _func.nameoff
  - args
    - Number of bytes used in arguments and return values
  - pcsp, pcfile, pcln
    - Information to retrieve stack size, file name, and line number corresponding to the assembly
    - How to retrieve the information will be published later in our blog

Change from pointer to offset Go 1.18

```
869  type _func struct {
870          entryoff uint32 // start pc, as offset from moduledata.text/pcHeader
871          nameoff  int32  // function name
872
873          args        int32  // in/out args size
874          deferreturn uint32 // offset of start of a deferreturn call instruct
875
876          pcsp      uint32
             pcfile    uint32
878                    uint32
883          _         [1]byte // pad
884          nfuncdata uint8   // must be last, must end on a uint32-aligned bour
885  }
```

Change the base address of nameoff Go 1.16

* https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/runtime2.go#L869-L885

Function information retrieval

- **Exemplified by Ghidra**

```
                    runtime.pcHeader_004bc240
004bc240 f0 ff ff ...     runtime....
  004bc240 f0 ff ff ff     uint      FFFFFFF0h      magic
  004bc244 00              db        0h             pad1
  004bc245 00              db        0h             pad2
  004bc246 01              db        1h             minLC
  004bc247 08              db        8h             ptrSize
  004bc248 95 05 00 00 00...longlong  595h          nfunc
  004bc250 b8 00 00 00 00...ulonglong B8h           nfiles
  004bc258 00 10 40 00 00...ulonglong 401000h       textStart
  004bc260 60 00 00 00 00...ulonglong 60h           funcnameOffset
  004bc268 60 bb 00 00 00...ulonglong BB60h         cuOffset
  004bc270 40 c3 00 00 00...ulonglong C340h         filetabOffset
  004bc278 e0 d1 00 00 00...ulonglong D1E0h         pctabOffset
  004bc280 00 86 03 00 00...ulonglong 38600h        pclnOffset
```

```
                                  XREF[2]:
004f4840 00 00 00 ...     runtime....
  004f4840 00 00 00 00 b0...runtime....        [0]

  004f4848 60 00 00 00 08...runtime....        [1]
    004f4848 60 00 00 00     uint      60h       entryoff
    004f484c 08 2d 00 00     uint      2D08h     funcoff
  004f4850 c0 05 00 00 60...runtime....        [2]
```

functab[nfunc]. AY_004f4840

```
004f7548 60 00 00 ...     _func
  004f7548 60 00 00 00     uint      60h       entryoff
  004f754c 18 00 00 00     int       18h       nameoff
  004f7550 10 00 00 00     int       10h       args
  004f7554 00 00 00 00     uint      0h        deferreturn
  004f7558 32 00 00 00     uint      32h       pcsp
  004f755c 43 00 00 00     uint      43h       pcfile
  004f7560 47 00 00 00     uint      47h       pcln
  004f7564 04 00 00 00     uint      4h        npcdata
  004f7568 00 00 00 00     uint      0h        cuOffset
  004f756c 00              db        0h        funcID
  004f756d 00              db        0h        flag
  004f756e 00              db[1]               _
  004f756f 07              db        7h        nfuncdata
```

```
internal/cpu/cpu.go:148
internal/cpu.processOptions
00401060 4c 8d 64 ...     LEA       R12=>local_8,[RSP + -0x8]
00401065 4d 3b 66 10      CMP       R12,qword ptr [R14 + 0x1...
00401069 0f 86 2a ...     JBE       LAB_00401599
0040106f 48 81 ec ...     SUB       RSP,0x88
00401076 48 89 ac ...     MOV       qword ptr [RSP + local_8...
0040107e 48 8d ac ...     LEA       RBP=>local_8,[RSP + 0x80]
00401086 48 89 84 ...     MOV       qword ptr [RSP + Stack[0...
```

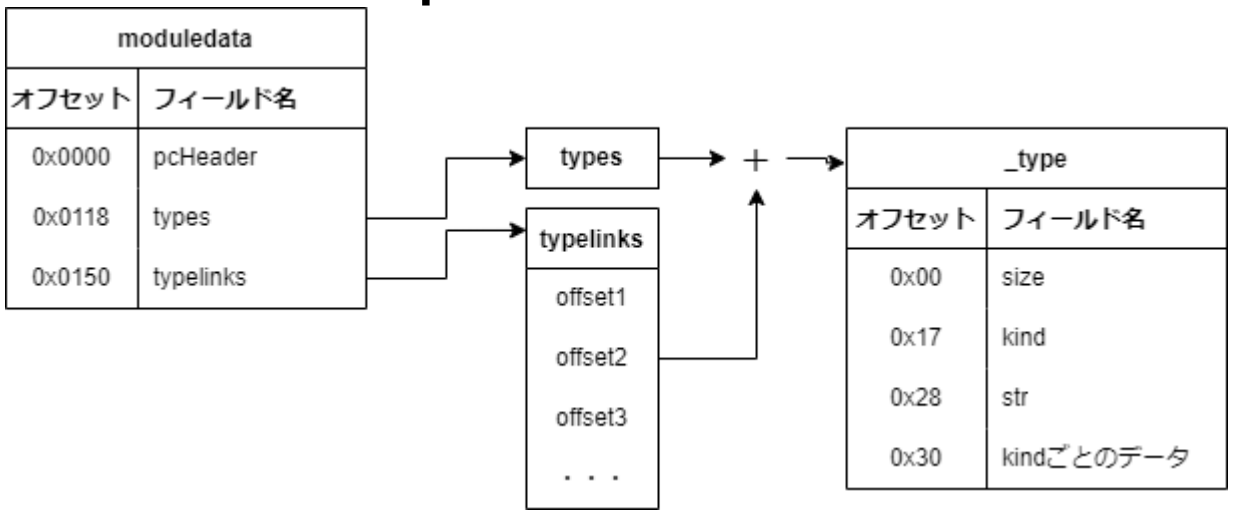
function name string

```
s_internal/cpu.Initialize_004bc2a0
004bc2a0 69 6e 74 ...     ds        "internal/cpu.Initialize"
004bc2b8 69 6e 74 ...     ds        "internal/cpu.processOptions"
004bc2d4 69 6e 74 ...     ds        "internal/cpu.indexByte"
```

Composition of data type information

- The flow for obtaining the _type structure containing each data type information from moduledata is shown in the figure below

- _type
  - Maintains information such as size and type of data type
  - Information specific to each type is added at the end of _type
  - Used in object creation and map access functions



Components of data type information

- _type *
  - size
    - Number of bytes of data type
  - kind
    - Data Type
    - Example of relationship between value and type
      - 1: Bool, 2: Int, 17: Array, 21: Map, 22: Pointer, 25: Struct
  - str
    - Offset to data type name
    - Relative to moduledata.types

```
35  type _type struct {
36          size       uintptr
37          ptrdata    uintptr // size of memory prefix holding all pointers
38          hash       uint32
39          tflag      tflag
40          align      uint8
41          fieldAlign uint8
42          kind       uint8
43          // function for comparing objects of this type
44          // (ptr to object A, ptr to object B) -> ==?
45          equal func(unsafe.Pointer, unsafe.Pointer) bool
46          // gcdata stores the GC type data for the garbage collector.
47          // If the KindGCProg bit is set in kind, gcdata is a GC program.
48          // Otherwise it is a ptrmask bitmap. See mbitmap.go for details.
49          gcdata    *byte
50          str       nameOff
51          ptrToThis typeOff
52  }
```

* https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/type.go#L35-L52

Components of data type information

- _type *
  - size
    - Number of bytes of data type
  - kind
    - Data Type
    - Example of relationship between value and type
      - 1: Bool, 2: Int, 17: Array, 21: Map, 22: Pointer, 25: Struct
  - str
    - Offset to data type name
    - Relative to moduledata.types

```
35  type _type struct {
36          size       uintptr
37          ptrdata    uintptr // size of memory prefix holding all pointers
38          hash       uint32
39          tflag      tflag
40          align      uint8
41          fieldAlign uint8
42          kind       uint8
43          // function for comparing objects of this type
44          // (ptr to object A, ptr to object B) -> ==?
45          equal func(unsafe.Pointer, unsafe.Pointer) bool
46                                                      age collector.
47                                                      a is a GC program.
                                                        ap.go for details.
49          gcdata     *byte
50          str        nameOff
51          ptrToThis  typeOff
```

Change from pointer to offset
Go 1.7

Change in method of obtaining strings
Go 1.17

* https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/runtime/type.go#L35-L52

Data type information retrieval

- **Exemplified by Ghidra**

```
                    runtime.moduledata_00518820
00518820 40 c2 4b ...     runtime.....
  00518820 40 c2 4b 00 00...runtime.....runtime.....pcHeader

  00518828 a0 c2 4b 00 00...[]uint8              funcnametab
  00518840 a0 7d 4c 00 00...[]uint32             cutab
  00518858 80 85 4c 00 00...[]uint8              filetab
  00518870 20 94 4c 00 00...[]uint8              pctab
  00518888 40 48 4f 00 00...[]uint8              pclntable

  00518938 00 40 48 00 00...ulonglong 484000h    types

  00518940 51 bc 4b 00 00...ulonglong 4BBC51h    etypes

  00518948 00 40 48 00 00...ulonglong 484000h    rodata

  00518950 e0 29 4a 00 00...ulonglong 4A29E0h    gofunc

  00518958 e0 a4 4b 00 00...[]runtim...          textsectmap
  00518970 00 bd 4b 00 00...[]int32              typelinks
    00518970 00 bd 4b 00 00...int *     UINT_ARR...__values
    00518978 37 01 00 00 00...ulonglong 137h       __count
    00518980 37 01 00 00 00...ulonglong 137h       __capacity
```

Array of typelinks

```
004bbd00 80 40 00 ...     uint[311]
  004bbd00 [0]                4080h,    40C0h,    4100h,    4140h,    9EA0h,
  004bbd18 [6]               10760h     E3E0h     BE40h     E4A0h     56C0h
```

+

_type

```
                    datatype.Ptr.*[]*runtime.bmap                XREF[2]:     004923c8(*),
00488080 08 00 00 ...     runtime.....
  00488080 08 00 00 00 00...ulonglong 8h                    size
  00488088 08 00 00 00 00...ulonglong 8h                    ptrdata
  00488090 65 75 1d fa     uint       FA1D7565h             hash
  00488094 08              db         8h                    tflag
  00488095 08              db         8h                    align
  00488096 08              db         8h                    fieldAlign
  00488097 36              db         36h                   kind          Ptr
  00488098 e8 2d 4a 00 00...void *    PTR_runtime.memequal64... equal     = 00402fa0
  004880a0 40 a0 4b 00 00...uint8 *   DAT_004ba040          gcdata        = 01h
  004880a8 66 2b 00 00     int        2B66h                 str           *[]*runtime.bmap
  004880ac 00 00 00 00     int        0h                    ptrToThis
004880b0 80              ??         80h                   datatype.Slice.[]*runtime.bmap
004880b1 9b              ??         9Bh
004880b2 48              ??         48h    H
004880b3 00              ??         00h
004880b4 00              ??         00h
```

0x36&0x1f=22

Example of Go Version Upgrade Support

- Target sample
  - Chaos
    - Hash value: ebe0f9855eb8f6bd980ed60c26e3a877dc1ace5d664e248bb0558996fe0bd06f
      - Go version: Go 1.18.1
      - OS: Linux
      - Arch: x86

- Tool (esp. software, etc.)
  - https://github.com/f0rki/r2-go-helpers
    - Script for radare2
    - Function: Resolve function name

Confirm current scope of support

- Results of applying to a binary built with Go 1.15
  - Function enumeration (afl)

```
0x0049b2c0    3 357          go.main.manyRet;
0x0049b440    3 383          go.main.goroutineFunc;
0x0049b5c0    9 1520         go.main.main;
0x0049bbc0    9 180          go.type..eq.[9]interface_{};
0x0049bc80    9 180          go.type..eq.[4]interface_{};
0x0049bd40    9 180          go.type..eq.[3]interface_{};
```

  - Assembly view of go.main.main (pdf)

```
0x0049b5c0    64488b0c25f8.   mov rcx, qword fs:[0xfffffffffffffff8]
0x0049b5c9    488d8424f8fe.   lea rax, qword [rsp - 0x108]
0x0049b5d1    483b4110        cmp rax, qword [rcx + 0x10]
0x0049b5d5    0f86cb050000    jbe 0x49bba6
0x0049b5db    4881ec880100.   sub rsp, 0x188
0x0049b5e2    4889ac248001.   mov qword [var_8h], rbp
0x0049b5ea    488dac248001.   lea rbp, qword [var_8h]
0x0049b5f2    488d05e33a02.   lea rax, qword [0x004bf0dc] ; "helloin
ode= c"
0x0049b5f9    48890424        mov qword [rsp], rax
0x0049b5fd    48c744240805.   mov qword [var_180h], 5
0x0049b606    e8b5f8ffff      call go.main.testFunc;
0x0049b60b    488b442418      mov rax, qword [var_170h]
0x0049b610    4803442410      add rax, qword [var_178h]
0x0049b615    4803442420      add rax, qword [var_168h]
```

Correction

Corrected to vaddr

```
base_addr = gopclntab['paddr']
size_addr = base_addr + 8
```

Add 4 to the argument

```
name_str_offset = get_pointer_at(bas
name_addr = base_addr + name_str_off
```

Confirmation of current scope of support

- Application results for Chaos
  - Execute script

```
[0x080ac1b0]> #!pipe python3 gohelper.py
INFO : We're gonna 'aa' first, this might take a while
[x] Analyze all flags starting with sym. and entry0 (aa)
INFO : renaming functions based on .gopclntab section
WARNING : not using function name '
' for 0x247
WARNING : not using function name '
' for 0x40
WARNING : not using function name '
' for 0xf15e0
WARNING : not using function name '
' for 0x0
```

Function Address Obviously different values are displayed.

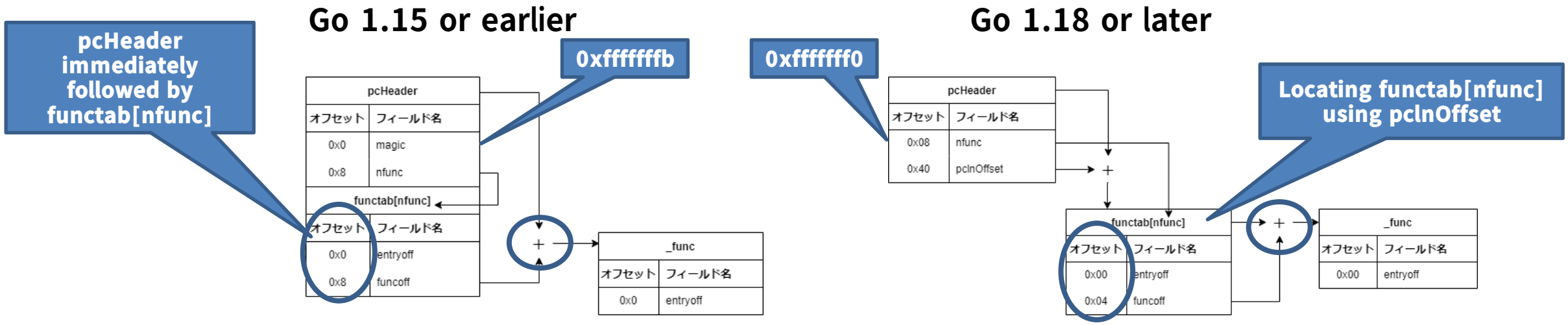
  - Function enumeration (afl)

```
[0x080ac1b0]> afl
0x080ac1b0    15 8613 -> 403  go.D32`U'>5(_G{$Dp7j?@WX[\_ab!
[0x080ac1b0]>
```

Functions are not enumerated

Impact of Version Upgrade

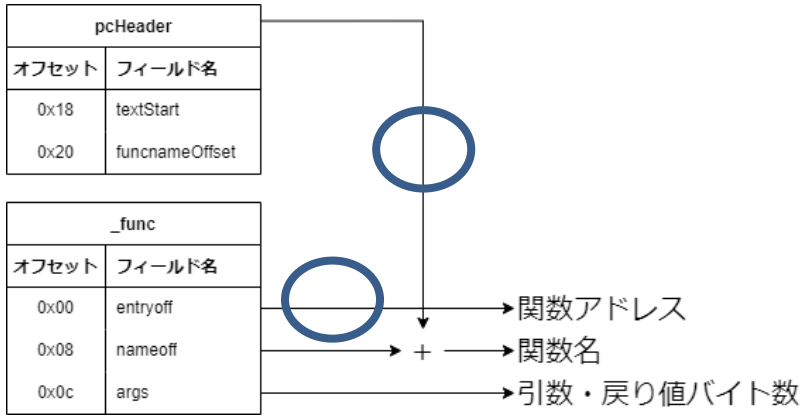
- pcHeader → _func
  - magic value changed
  - Change reference method for arrays containing links to function information
  - Change base address of functab.funcoff
  - Functab field size changed from pointer size to 0x4



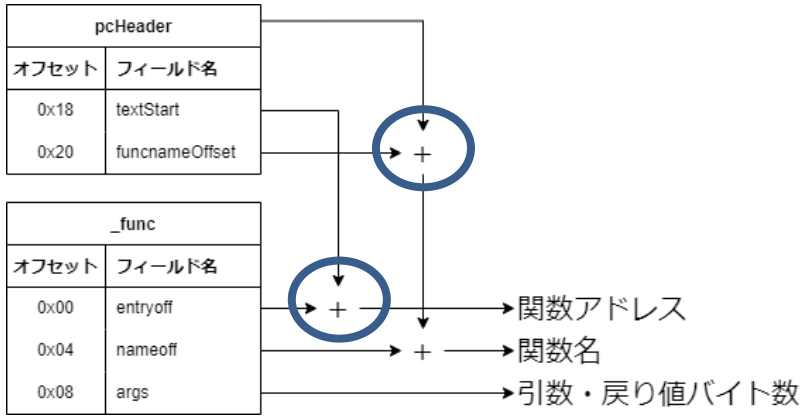
Impact of Version Upgrade

- _func → Function information
  - _func.entryoff changed from pointer to offset
  - Base address change for _func.nameoff

Go 1.15 or earlier



Go 1.18 or later



Tool modification for Chaos

- Causes of failure in extraction
  - Failure to respond to changes between Go 1.15 and Go 1.18.

- Areas that should be corrected *

```
start_addr = size_addr + PTR_SIZE
end_addr = base_addr + (size * PTR_SIZE * 2)

for addr in range(start_addr, end_addr, (2 * PTR_SIZE)):
    log.debug("analyzing at 0x{:x}".format(addr))
    func_addr = get_pointer_at(addr)
    entry_offset = get_pointer_at(addr + PTR_SIZE)

    log.debug("func_addr 0x{:x}, entry offset 0x{:x}"
              .format(func_addr, entry_offset))

    name_str_offset = get_pointer_at(base_addr + entry_offset + PTR_SIZE)
    name_addr = base_addr + name_str_offset
```

Different method of obtaining functab array position

Different functab sizes

It will be an offset, not an address.

Base address is different

* https://github.com/f0rki/r2-go-helpers/blob/d1167e4b96ba0e3c33ee8c5e578bb3cde930324e/gohelper.py#L165-L177

Applying modified tools to Chaos

- Application results for Chaos
  - Function enumeration (afl)

```
0x082d8340    6 76          go.main.readPidsFromDir.func1;
0x082d8b80   64 2177        go.main.main;
0x082d9410    6 70          go.main.main.func2;
0x082d9460    6 62          go.main.main.func1;
0x082d94a0    3 66          go.main.chaos_time;
```

  - Assembly view of go.main.main (pdf)

```
0x082d8b80    658b0d000000.   mov ecx, dword gs:[0]
0x082d8b87    8b89fcffffff    mov ecx, dword [ecx - 4]
0x082d8b8d    8d442480        lea eax, dword [esp - 0x80]
0x082d8b91    3b4108          cmp eax, dword [ecx + 8]
0x082d8b94    0f865d080000    jbe 0x82d93f7
0x082d8b9a    81ec00010000    sub esp, 0x100
0x082d8ba0    b800000000      mov eax, 0
0x082d8ba5    898424f80000.   mov dword [var_8h], eax
0x082d8bac    898424fc0000.   mov dword [var_4h], eax
0x082d8bb3    c644242300      mov byte [var_ddh], 0
0x082d8bb8    e8e31d0000      call go.main.Getmyname;
0x082d8bbd    e8ce1c0000      call go.main.Getmypwd;
0x082d8bc2    8b0424          mov eax, dword [esp]
0x082d8bc5    8b4c2404        mov ecx, dword [var_fch]
```

Support for obfuscation by gobfuscate

- What will be explained
  - Introduction of obfuscation methods with gobfuscate
  - Malware String Unobfuscation
  - Functional Estimation in Obfuscated Malware

- Target sample
  - ChaChi
    - Hash value: 8a9205709c6a1e5923c66b63addc1f833461df2c7e26d9176993f14de2a39d5b

Obfuscation with gobfuscate

- What is gobfuscate?
  - Functions: obfuscation of strings and functions, etc.
  - Obfuscation by modifying source code, etc.
  - https://github.com/unixpickle/gobfuscate

Obfuscated functions

functions | filenames | datatypes

Location	Function Name
007c5580	main.naopcgecfflmbgciiiho.func4
007c5670	main.naopcgecfflmbgciiiho.func5
007c5760	main.naopcgecfflmbgciiiho.func6
007c5830	main.(*Bjbealpgdnpanhaihjba).mlkbhekbfckondcckjam.func1
007c58a0	main.(*Bjbealpgdnpanhaihjba).mlkbhekbfckondcckjam.func2
007c5990	main.(*Bjbealpgdnpanhaihjba).pmeanjdmanemokfpkhbm.fun
007c5a70	main.(*Bjbealpgdnpanhaihjba).eiadljbonenkfjpnbgea.func1

- Here we deal with string obfuscation

String obfuscation

```
/tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:86
LAB_007be85a
    CALL    main.main.func1
    MOV     RAX,qword ptr [RSP + 0x8]
    MOV     qword ptr [RSP + 0x58],RAX
    MOV     RCX,qword ptr [RSP]
    MOV     qword ptr [RSP + 0xa8],RCX
/tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:94
    CALL    main.main.func2
/tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:78
    NOP
/tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:94
    MOV     RAX,qword ptr [RSP + 0x8]
    MOV     RCX,qword ptr [RSP]
/usr/local/go/src/flag/flag.go:644
    MOV     RDX,qword ptr [DAT_00c07008]
    MOV     qword ptr [RSP],RDX
    MOV     RDX,qword ptr [RSP + 0xa8]
    MOV     qword ptr [RSP + 0x8],RDX
    MOV     RDX,qword ptr [RSP + 0x58]
    MOV     qword ptr [RSP + 0x10],RDX
    MOV     byte ptr [RSP + 0x18],0x0
    MOV     qword ptr [RSP + 0x20],RCX
    MOV     qword ptr [RSP + 0x28],RAX
    CALL    flag.(*FlagSet).Bool
```

Should be a string

Should be a string

func Bool(name string, value bool, usage string) *bool *

* https://github.com/golang/go/blob/4a4127bccc826ebb6079af3252bc6bfeaec187c4/src/flag/flag.go#L734

gobfuscate string obfuscation

- String obfuscation methods with gobfuscate
  - Generating a string with an anonymous function
  - String is encrypted by XOR

Function generated when obfuscating "golang" with gobfuscate

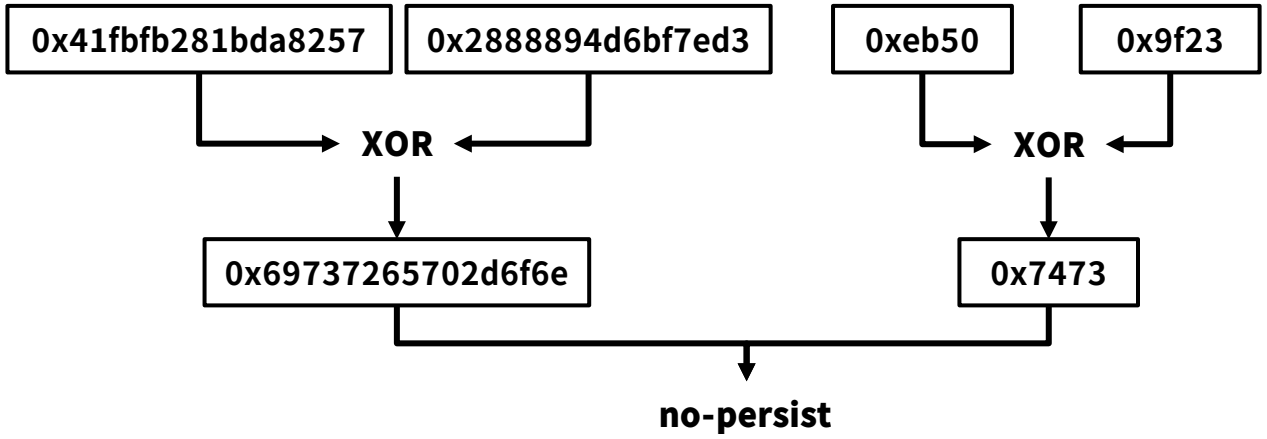
```
(func() string {
mask := []byte("\x21\x0f\xc7\xbb\x81\x86")
maskedStr := []byte("\x46\x60\xab\xda\xef\xe1")
res := make([]byte, 6)
        for i, m := range mask {
            res[i] = m ^ maskedStr[i]
        }
        return string(res)
        }())
```

An anonymous function is used.

Converts a byte sequence to a string

String obfuscation in ChaChi

- **Function: main.main.func1**



Byte-string extraction

```
                    /tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:79
007c4ab8 48 b8 a9 ...   MOV     RAX,-0x41fbfb281bda8257
007c4ac2 48 89 44 ...   MOV     qword ptr [RSP + local_1c],RAX
007c4ac7 66 c7 44 ...   MOV     word ptr [RSP + local_14],0xeb50
                    /tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:80
007c4ace 48 b8 c7 ...   MOV     RAX,-0x2888894d6bf7ed39
007c4ad8 48 89 44 ...   MOV     qword ptr [RSP + local_26],RAX
007c4add 66 c7 44 ...   MOV     word ptr [RSP + local_1e],0x9f23
                    /tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:81
007c4ae4 48 c7 44 ...   MOV     qword ptr [RSP + local_12],0x0
007c4aed 48 c7 44 ...   MOV     qword ptr [RSP + local_12+0x2],0x0
007c4af6 31 c0          XOR     EAX,EAX
                    /tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:82
007c4af8 eb 13          JMP
```

XOR a sequence of bytes.

```
                    LAB_007c4afa
007c4afa 0f b6 4c ...   MOVZX   ECX,byte ptr [RSP + RAX*0x1 + 0x3c]
                    /tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:83
007c4aff 0f b6 54 ...   MOVZX   EDX,byte ptr [RSP + RAX*0x1 + 0x32]
007c4b04 31 d1          XOR     ECX,EDX
007c4b06 88 4c 04 46    MOV     byte ptr [RSP + RAX*0x1 + 0x46],CL
                    /tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:82
007c4b0a 48 ff c0       INC
```

Number of loops

```
                    LAB_007c
007c4b0d 48 83 f8 0a    CMP     RAX,0xa
007c4b11 7c e7          JL      LAB_007c4afa
                    /tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:85
007c4b13 48 c7 04 ...   MOV     qword ptr [RSP]=>local_58,0x0
007c4b1b 48 8d 44 ...   LEA     RAX=>local_12,[RSP + 0x46]
007c4b20 48 89 4
007c4b25 48 c7 4
007c4b2e 48 c7 4
007c4b37 e8 a4 92 ...   CALL    runtime.slicebytetostring
007c4b3c 48 8b 44 ...   MOV     RAX,qword ptr [RSP + local_38]
007c4b41 48 8b 4c ...   MOV     RCX,qword ptr [RSP + local_30]
007c4b46 48 89 44 ...   MOV     qword ptr [RSP + param_1],RAX
007c4b4b 48 89 4c ...   MOV     qword ptr [RSP + param_2],RCX
```

Converts a byte sequence to a string and places it in the return value.

Anonymous Functions

- Go Language Anonymous Functions
  - Example: f := func(arg string) string { return "arg: " + arg }

- Representation in Assembly
  - Function name is in the form "<function name of implementation source>.func%d"
    - Not all function names of this form are developer-implemented anonymous functions.

String Obfuscation Removal for ChaChi

- How to implement?
  - For functions that are anonymous and only called by runtime.slicebytetostring
    - With exceptions such as the function itself and runtime.morestack_noctxt
  - Get the data stored on the stack in the second and third lines of the function
  - XOR the acquired data and convert it to a string
  - Rename the function name with the obtained string

```go
(func() string {
mask := []byte("\x21\x0f\xc7\xbb\x81\x86")
maskedStr := []byte("\x46\x60\xab\xda\xef\xe1")
res := make([]byte, 6)
        for i, m := range mask {
            res[i] = m ^ maskedStr[i]
        }
        return string(res)
        }())
```

main.main.func1 function

```
/tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:78
main.main.func1

007c4a90 65 48 8b ...    MOV     RCX,qword ptr GS:[0x28]
007c4a99 48 8b 89 ...    MOV     RCX,qword ptr [RCX]
007c4aa0 48 3b 61 10     CMP     RSP,qword ptr [RCX + 0x10]
007c4aa4 0f 86 b0 ...    JBE     LAB 007c4b5a
007c4aaa 48 83 ec 58
007c4aae 48 89 6c ...
007c4ab3 48 8d 6c ...
```

Second and third lines

```
/tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:79
007c4ab8 48 b8 a9 ...    MOV     RAX,-0x41fbfb281bda8257
007c4ac2 48 89 44 ...    MOV     qword ptr [RSP + local_1c],RAX
007c4ac7 66 c7 44 ...    MOV     word ptr [RSP + local_14],0xeb50
/tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:80
007c4ace 48 b8 c7 ...    MOV     RAX,-0x2888894d6bf7ed39
007c4ad8 48 89 44 ...    MOV     qword ptr [RSP + local_26],RAX
007c4add 66 c7 44 ...    MOV     word ptr [RSP + local_1e],0x9f23
/tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:81
```

snipped

```
/tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go:83
007c4aff 0f b6 54 ...    MOVZX   EDX,byte ptr [RSP + RAX*0x1 + 0x32]
007c4b04 31 d1           XOR     ECX,EDX
007c4b06 88 4c 04 46     MOV     byte ptr [RSP + RAX*0x1 + 0x46],CL
```

String Obfuscation Removal for ChaChi

- Multiple obfuscated strings
- Stored in memory after decryption

ChaChi's main.init function
(The label of the destination memory is set manually.)

```
007c7580 48 3b 61 10     CMP        RSP,qword ptr [RCX + 0x10]
007c7584 0f 86 81 ...    JBE        LAB_007c780b
007c758a 48 83 ec 18     SUB        RSP,0x18
007c758e 48 89 6c ...    MOV        qword ptr [RSP + 0x10]=>local_8,RBP
007c7593 48 8d 6c ...    LEA        RBP=>local_8,[RSP + 0x10]
                         /tmp/422910383/src/lnjmnamogplhgkllpmbe/constants_windows.go:...
007c7598 e8 83 a9 ...    CALL       gobfus_JavaJDBC_main.glob..func1
                         MOV        RAX,qword ptr [RSP]=>local_18
007c75a1 48 8b 4c ...    MOV        RCX,qword ptr [RSP + 0x8]=>local_10
                         /tmp/422910383/src/lnjmnamogplhgkllpmbe/constants_windows.go:3
007c75a6 48 89 0d ...    MOV        qword ptr [gobfus_JavaJDBC_len],RCX
007c75ad 83 3d 4c ...    CMP        dword ptr [DAT_00c37700],0x0
007c75b4 0f 85 40 ...    JNZ        LAB_007c77fa
007c75ba 48 89 05 ...    MOV        qword ptr [gobfus_JavaJDBC],RAX

                         /tmp/422910383/src/lnjmnamogplhgkllpmbe/constants_windows.go:...
                         LAB_007c75c1
007c75c1 e8 1a aa ...    CALL       gobfus_Java_JDBC_main.glob..func2
                         MOV        RAX,qword ptr [RSP]=>local_18
007c75ca 48 8b 4c ...    MOV        RCX,qword ptr [RSP + 0x8]=>local_10
                         /tmp/422910383/src/lnjmnamogplhgkllpmbe/constants_windows.go:...
007c75cf 48 89 0d ...    MOV        qword ptr [gobfus_Java_JDBC_len],RCX
007c75d6 83 3d 23 ...    CMP        dword ptr [DAT_00c37700],0x0
007c75dd 0f 85 06 ...    JNZ        LAB_007c77e9
007c75e3 48 89 05 ...    MOV        qword ptr [gobfus_Java_JDBC],RAX

                         /tmp/422910383/src/lnjmnamogplhgkllpmbe/constants_windows.go:...
                         LAB_007c75ea
007c75ea e8 c1 aa ...    CALL       gobfus_Oracle_JDB_service_driver_main.glob..func3
```

Unobfuscated

Unobfuscated

Unobfuscated

Function estimation of Obfuscated Malware

- Obfuscated strings in ChaChi have been resolved, but there is still obfuscated information that is difficult to parse

- Introduction to the estimation of functions and OSS libraries used to efficiently analyze obfuscated samples
  - Estimate functions
    - Collection of surface information
  - Estimate OSS libraries in use
    - Estimated from file name, line number, etc.

Obfuscated function names do not reveal what they are doing.

functions | filenames | datatypes

Location	Function Name
007c5580	main.naopcgecfflmbgciiiho.func4
007c5670	main.naopcgecfflmbgciiiho.func5
007c5760	main.naopcgecfflmbgciiiho.func6
007c5830	main.(*Bjbealpgdnpanhaihjba).mlkbhekbfckondcckjam.func1
007c58a0	main.(*Bjbealpgdnpanhaihjba).mlkbhekbfckondcckjam.func2
007c5990	main.(*Bjbealpgdnpanhaihjba).pmeanjdmanemokfpkhbm.fun
007c5a70	main.(*Bjbealpgdnpanhaihjba).eiadljbonenkfjpnbgea.func1

Estimating the function

- Check the file name sequence
- The service is supposed to have the ability to run
  - service.go file exists
    - Call OpenService in the internal process

filename sequence

functions | filenames | datatypes

Filename
/tmp/422910383/src/heimaoplnhkdaeiflhmp/lpdkaklidghllnacngmd/lmmbpplgmfkpanjncdf/console.go
/tmp/422910383/src/heimaoplnhkdaeiflhmp/lpdkaklidghllnacngmd/lmmbpplgmfkpanjncdf/service.go
/tmp/422910383/src/heimaoplnhkdaeiflhmp/lpdkaklidghllnacngmd/lmmbpplgmfkpanjncdf/service_go1.8.go
/tmp/422910383/src/heimaoplnhkdaeiflhmp/lpdkaklidghllnacngmd/lmmbpplgmfkpanjncdf/service_windows.go
/tmp/422910383/src/lnjmnamogplhgkllpmbe/chisel.go
/tmp/422910383/src/lnjmnamogplhgkllpmbe/command.go
/tmp/422910383/src/lnjmnamogplhgkllpmbe/commands_windows.go
/tmp/422910383/src/lnjmnamogplhgkllpmbe/constants_windows.go
/tmp/422910383/src/lnjmnamogplhgkllpmbe/daemon_windows.go
/tmp/422910383/src/lnjmnamogplhgkllpmbe/error.go
/tmp/422910383/src/lnjmnamogplhgkllpmbe/fake.go
/tmp/422910383/src/lnjmnamogplhgkllpmbe/is_admin_windows.go
/tmp/422910383/src/lnjmnamogplhgkllpmbe/main.go

Presumption of OSS libraries

- Investigate heimaoplnhkdaeiflhmp/lpdkaklidghllnacngmd/lmmbpplgmfkpanjncdff
  - Probably OSS libraries available on github.com
- If you can identify the above paths, you can read the source code to understand the internal processing and usage, which makes the analysis more efficient
- Attempt a search with the following unobfuscated string

SYSTEM\CurrentControlSet\Control

```
/tmp/422910383/src/heimaoplnhkdaeiflhmp/lpdkaklidghllnacngmd/lmm
    CALL        gobfus_SYSTEM_CurrentControlSet_Control_heimao... void gobfus_SYSTEM_CurrentContro...
    MOV         RAX,qword ptr [RSP]=>local_50
    MOV         RCX,qword ptr [RSP + local_48]
/tmp/422910383/src/heimaoplnhkdaeiflhmp/lpdkaklidghllnacngmd/lmmbpplgmfkpanjncdff/service_windows.go:489
    MOV         EDX,0x80000002
    MOV         qword ptr [RSP]=>local_50,RDX
    MOV         qword ptr [RSP + local_48],RAX
    MOV         qword ptr [RSP + local_40],RCX
    MOV         dword ptr [RSP + local_38],0x20019
    CALL        golang.org/x/sys/windows/registry.OpenKey        void golang.org/x/sys/windows/re...
    MOV         RAX,qword ptr [RSP + local_30]
    CMP         qword ptr [RSP + local_28],0x0
/tmp/422910383/src/heimaoplnhkdaeiflhmp/lpdkaklidghllnacngmd/lmmbpplgmfkpanjncdff/service_windows.go:498
    JNZ         LAB_0076e347
/tmp/422910383/src/heimaoplnhkdaeiflhmp/lpdkaklidghllnacngmd/lmm
    MOV         qword ptr [RSP + local_10],RAX
/tmp/422910383/src/heimaoplnhkdaeiflhmp/lpdkaklidghllnacngmd/lmmbpplgmfkpanjncdff/service_windows.go:509
    CALL        gobfus_WaitToKillServiceTimeout_heimaoplnhkdae... void gobfus_WaitToKillServiceTim...
```

WaitToKillServiceTimeout

Identify Library

- Search: "site:github.com golang WaitToKillServiceTimeout SYSTEM\CurrentControlSet\control"
  - https://github.com/kardianos/service/blob/master/service_windows.go
    - File name matches
    - Line numbers also match to some extent
      - String obfuscation shifts the number of lines
  - https://github.com/takama/daemon/blob/master/daemon_windows.go
    - File names do not match

- OSS library "github.com/kardianos/service" is used

OSS libraries used by ChaChi

- github.com/rs/xid
  - Obtain a unique ID
- github.com/fasthttp/websocket
  - gorilla/websocket fork with fasthttp support
- github.com/armon/go-socks5
  - SOCKS5 Server
- github.com/fsnotify/fsnotify
  - File system notification
- github.com/jpillora/backoff
  - Exponential backoff counter
- github.com/Jeffail/tunny
  - Library for generating and managing Goroutine pools
- etc.

SUMMARY

Reflection and Challenges

- Challenge
  - Go language malware presents several challenges that make analysis difficult

- To improve analytical skills
  - Basic analysis flow and tips
  - How to modify the metadata in Go language binaries and the tools that use it
  - Support for Go language version upgrades
  - Coutermeasures against obfuscated samples

- Future Issues
  - Poor decompile performance of Ghidra and others
  - Lack of dynamic analysis tools

Tools

- https://github.com/mooncat-greenpy/Ghidra_GolangAnalyzerExtension
  - Ghidra plugin for binary analysis made by Go language used mainly in this presentation

- https://github.com/FFRI/JSAC2023-GolangMalwareAnalysis
  - Scripts of radare2 modified in response to version upgrade
  - Ghidra script to remove string obfuscation by gobfuscate

Thank you for attending!

APPENDIX

Appendix: init Function

- init function
  - It is a function used for initialization, etc.
  - Called before the main function
  - Multiple definitions possible

- Representation in Assembly
  - Function names are in the form "<module name>.init.%d" or "<module name>.init"
    - Example: main.init.0
  - init function is called by runtime.doInit function
    - runtime.doInit function is called from runtime.main function which calls main function

Appendix : Goroutine

- Goroutine
  - Go's unique lightweight threads
  - Example: go sub_func(0x1, 0x10, 0x100, 0x1000) // execute sub_func function

- Representation in Assembly
  - Goroutine is started by runtime.newproc function
  - Argument differences by version
    - Go 1.17 or lower
      - Pass a pointer to memory with "call function pointer, argument 1, argument 2, ..." as the second argument
    - Go 1.17 or higher
      - A wrapper function for the target function is passed to newproc
        - » Some arguments are provided in the wrapper function
    - Go 1.18 or higher
      - Pass the pointer described above as the first argument

Appendix: Version Determination

- Here is how to retrieve the version of the Go language used to build the Go binaries

```
007b7020 ff          ??   FFh
007b7021 20          ??   20h
007b7022 47          ??   47h    G
007b7023 6f          ??   6Fh    o
007b7024 20          ??   20h
007b7025 62          ??   62h    b
007b7026 75          ??   75h    u
007b7027 69          ??   69h    i
007b7028 6c          ??   6Ch    l
007b7029 64          ??   64h    d
007b702a 69          ??   69h    i
007b702b 6e          ??   6Eh    n
007b702c 66          ??   66h    f
0                    ??   3Ah    :
0                    ??   08h
007b702f 00          ??   00h
007b7030 10 39 86    addr goss_go1.13.5_1863910
         01 00 00
         00 00
```

Search this byte sequence

Pointer Size

Pointer to version information

```
01863910 62 a4 94    addr s_go1.13.5_0194a462
         01 00 00
         00 00
```

Pointer to string

Character string length

```
01863918 08          ??   08h
01863919 00          ??   00h
0186391a 00          ??   00h
0186391b 00          ??   00h
0186391c 00          ??   00h
0186391d 00          ??   00h
0186391e 00          ??   00h
0186391f 00          ??   00h
```

Version string
It was built with Go 1.13.5.

```
                                 s_go1.13.5_0194a462
0194a462 67 6f 31                ds   "go1.13.5"
         2e 31 33
         2e 35
```