گزارش امنیتی

شناسایی باج افزار Golang در سیستم عامل‌های لینوکس

مقدمه:

طبق تحقیقاتی که تیم تحقیقاتی شرکت امنیتی Fortinet در طی دو ماه گذشته در آزمایشگاه FortiGuard انجام داده اند یک باج افزار در Golang کشف شده است. در این آزمایشگاه با انجام مهندسی معکوس روی بدافزار موجود در Golang توانستند این باج افزار را کشف کنند. Golang که با نام Go نیز شناخته شده است یک زبان برنامه نویسی استاتیک است که در Google طراحی شده است و در جوامع توسعه دهنده بدافزار محبوبیت بیشتری دارد. در این گزارش، یک باج افزار تازه کشف شده در Golang را که سیستم‌های لینوکس را مورد هدف قرار می‌دهد، تحلیل خواهیم کرد.

بررسی اجمالی Go باینری

نمونه‌ی تجزیه و تحلیل شده یک ELF[1] مجرد[2] است. ELF یک فرمت استاندارد معمول برای فایل‌های قابل اجرا، کد شی و کتابخانه‌های مشترک است. اولین بار در مشخصات رابط باینری برنامه (ABI) نسخه سیستم عامل یونیکس با نام System V Release 4 (SVR4) و بعداً در Standard Interface Standard منتشر شد. یک اجرای مجرد می‌تواند مهندسی را سخت‌تر کند، زیرا برای بازیابی نمادها در باینری مجرد باید کارهای اضافی انجام داد. خوشبختانه، ابزار تصحیح می‌تواند در این زمینه به ما کمک کند. نرم افزار Redress ابزاری برای آنالیز Go باینری های کامپایل شده با کامپایلر Go است. داده ها را از باینری استخراج می‌کند، سپس از آن برای بازسازی نمادها و انجام تجزیه و تحلیل استفاده می‌کند. در شکل زیر نتیجه حاصل از تجزیه و تحلیل این نمونه با پارامتر "-src" است.

```
kailu-mac:redress-v0.4.0 kailu$ ./redress -src ../ransomeware/elf_cryptor
Package main: /Users/usasucks/v2
File: <autogenerated>
        init Lines: 1 to 1 (0)
File: file.go
        EncEAS Lines: 13 to 100 (87)
        EncFile Lines: 100 to 103 (3)
File: main.go
        init0 Lines: 45 to 60 (15)
        main Lines: 60 to 104 (44)
        mainfunc1 Lines: 82 to 92 (10)
        randSeq Lines: 104 to 113 (9)
        writemessage Lines: 113 to 117 (4)
        chDir Lines: 117 to 142 (25)
        check Lines: 142 to 169 (27)
        locale Lines: 169 to 174 (5)
File: rsa.go
        makesecret Lines: 29 to 38 (9)
kailu-mac:redress-v0.4.0 kailu$
```

شکل ۱ – خروجی کد منبع پس از تجزیه و تحلیل بدافزارها در Redress

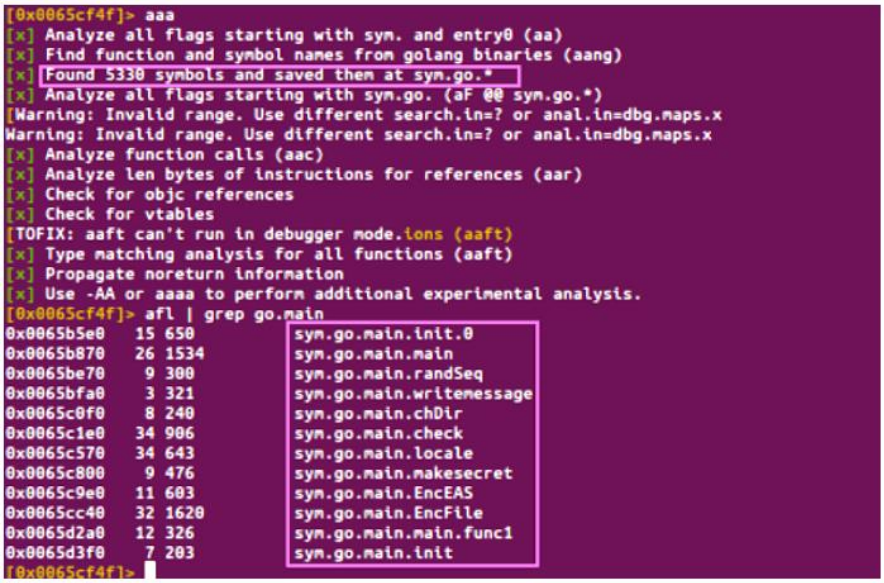
[1] Executable and Linkable Format
[2] Stripped

همانطور که در شکل(۱) مشاهده می‌کنید کد منبع بدافزار شامل سه قسمت است ۱- فایل Go ۲- توابع اجرا شده ۳- شماره خط کد آنها است. از نام برخی توابع، می‌توان حدس زد که این بدافزار باید باج افزار باشد. با این حال، تعداد خطوط در کد منبع کمی بیشتر از ۳۰۰ است. بنابراین این یک باج افزار پیچیده‌ای نیست و ممکن است در مرحله توسعه اولیه خود باشد.

در مرحله‌ی بعدی، یک اشکال زدایی[۳] پویا در اشکال زدایی شروع می‌شود. در اینجا، از Radare2 به عنوان اشکال زدا استفاده شده است. Radare2 با صدور فرمان تجزیه و تحلیل قادر به تجزیه و تحلیل Go باینری مجرد و بازگرداندن نمادها است. به همین دلیل در این پروژه به جای GDB[۴] از Radare2 به عنوان اشکال زدا در لینوکس استفاده شده است. GDB یک اشکال زدایی قابل حمل است که در بسیاری از سیستم‌های شبیه به یونیکس اجرا شده و برای بسیاری از زبان های برنامه نویسی از جمله Ada، C، C ++، Objective-C، Free Pascal، Fortran، Go و بعضی دیگر کار می‌کند.

تجزیه و تحلیل پویا درباره‌ی Go باینری

برای درک بهتر عملکرد Radare2 دستور "aaa" انجام شده است تا بتواند آنالیز اتوماتیک انجام دهد. همانطور که در شکل ۲ می بینیم، Radare2 نام عملکردها و نام‌های نمادها را خیلی خوب بازیابی و تشخیص می‌دهد. این کار کمک می‌کند تا با کارآیی بیشتری اشکال زدایی روی Go باینری انجام شود.



شکل ۲ - اصلاح اسم توابع و نمادها با استفاده از اشکال زدای Radar2

[3] debugging
[4] GNU Debugger

همانطور که می بینید، تابع init() قبل از تابع main انجام می‌شود. تابع check() در تابع init() فراخوانی می‌شود. در تابع check()، بدافزار ابتدا با ارسال یک درخواست http به hxxps: //ipapi.co/json/ اطلاعات مکان دستگاه آلوده را بدست می آورد. سپس خروجی بلاروس (BY)، روسیه (RU) و اوکراین (UA) را فیلتر می‌کند تا در صورت اجرای بدافزار در یکی از آن کشورها از اجرای خود جلوگیری کند.

```
.text:000000000065C384                call    strings_ToLower
.text:000000000065C389                mov     rax, [rsp+70h+var_60]
.text:000000000065C38E                mov     rcx, [rsp+70h+var_58]
.text:000000000065C393                cmp     rcx, 2
.text:000000000065C397                jnz     short loc_65C3A8
.text:000000000065C399                cmp     word ptr [rax], 'yb'
.text:000000000065C39E                jz      loc_65C48A
.text:000000000065C3A4                cmp     rcx, 2
.text:000000000065C3A8
.text:000000000065C3A8 loc_65C3A8:                            ; CODE XREF: main check+1B7↑j
.text:000000000065C3A8                jnz     short loc_65C3B5
.text:000000000065C3AA                cmp     word ptr [rax], 'ur'
.text:000000000065C3AF                jz      short loc_65C42D
.text:000000000065C3B1                cmp     rcx, 2
.text:000000000065C3B5
.text:000000000065C3B5 loc_65C3B5:                            ; CODE XREF: main check:loc_65C3A8↑j
.text:000000000065C3B5                jnz     short loc_65C3BE
.text:000000000065C3B7                cmp     word ptr [rax], 'au'
.text:000000000065C3BC                jz      short loc_65C3D0
.text:000000000065C3BE
.text:000000000065C3BE loc_65C3BE:                            ; CODE XREF: main check:loc_65C3B5↑j
.text:000000000065C3BE                xorps   xmm0, xmm0
.text:000000000065C3C1                movups  [rsp+70h+arg_0], xmm0
.text:000000000065C3C6                mov     rbp, [rsp+70h+var_8]
.text:000000000065C3CB                add     rsp, 70h
.text:000000000065C3CF                retn
```

شکل ۳- فیلتر کردن بلاروس (BY) ، روسیه (RU) و اوکراین (UA)

در تابع main()، ابتدا Go باینری را حذف می‌کند. سپس تابع randSeq() را برای ایجاد یک کلید AES تصادفی که در آن اندازه x20۰ در بایت است، مانند زیر فراخوانی می‌کند:

```
[0x0065b8d6]> pd 6 @ 0x65B8BF
|           0x0065b8bf      48894c2408      mov qword [var_8h], rcx
|           0x0065b8c4      e8a7cce5ff      call sym.go.os.Renove
|           0x0065b8c9      48c704242000.   mov qword [rsp], 0x20          ; rax
|                                                                          ; [0x20:8]=-1
|           0x0065b8d1      e89a050000      call sym.go.main.randSeq
|           0x0065b8d6      48c704240000.   mov qword [rsp], 0
|           ;-- rip:
|           0x0065b8de      e86daedeff      call sym.go.runtime.stringtoslicebyte      random AES key(0x20 bytes)
[0x0065b8d6]>  px 0x20 @ 0x000000c00001ab10
- offset -     0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0xc00001ab10  4b79 5050 574a 666d 4f68 4a79 5649 7852  KyPPWJfmOhJyVIxR
0xc00001ab20  7064 4f5e 656e 7679 4e5a 5951 4965 2325  pdO^envyNZYQIe#%
```

شکل ۴- ایجاد کلید تصادفی AES

در مرحله بعد، تابع makesecret() فراخوانی می‌شود که برای رمزنگاری کلید AES با یک کلید عمومی RSA استفاده می‌شود. در داخل این تابع، تابع EncryptPKCS1v15 برای رمزنگاری کلید AES با استفاده از رمزنگاری RSA داده شده است و طرح لایه بندی را از PKCS # 1 v1.5 فراخوانی می‌کند.

```
.noptrdata:00000000009060C0 aBeginPublicKey db 0Ah                  ; DATA XREF: .data:off_9270A04o
.noptrdata:00000000009060C0                 db '-----BEGIN PUBLIC KEY-----',0Ah
.noptrdata:00000000009060C0                 db 'MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAktbaBXZGkhUDs1goH1GO',0Ah
.noptrdata:00000000009060C0                 db 'CIN0XjosVi/U2nYh9uqNWPo+1ZzCLixnVCrLwxxNlr1keuJZ2f0hOjpw4j4wQzNG',0Ah
.noptrdata:00000000009060C0                 db '6IJ0um75qgcwRt4Yk4AjndOWy3u1MmZtqrrxW6On5fgch5FL54aQoPWeF47k7JB9',0Ah
.noptrdata:00000000009060C0                 db '6oF198TU+Z9/5ecO7FUcFBCg5lJ1ErOgU7YVTVZy4/LelROgRv3C5kD12VVf3g7X',0Ah
.noptrdata:00000000009060C0                 db '2AfHvasEJ2sYtXu6pAf/MmcU8H/GjNWQ6tPyN4tt/V1QsxNQ0bOEANdSSkM1+v9q',0Ah
.noptrdata:00000000009060C0                 db 'Un3w9XZGeYVgueo7QOIKtye+gvL2oHFjm8rjQwYQtpvxCL7GSgDUM59TzsTXmNNY',0Ah
.noptrdata:00000000009060C0                 db 'tlZVQ/jMQ16lnBexevAfWpX3spYFWw+vE6767Jrem8GKVIXee8yzY68L4gXZrn19',0Ah
.noptrdata:00000000009060C0                 db '+5Vp/vAlBHfQiAxYv1/WFmncEaWYGOxB6pGYbxys7Bm839Vz0lFUwHTBP+WyIL+t',0Ah
.noptrdata:00000000009060C0                 db 'OFqf7hQGfziu7xssaK2an9QTvBMn/7Q0GVo94bnkXfeds8OhJc14Rwo0Jmxny58M',0Ah
.noptrdata:00000000009060C0                 db 'HvHwAz7ZqSGw/1+AT4AaYR/52ADebIIEu0y2sQKmTXId50YaxLRgI62WSOP8srGJ',0Ah
.noptrdata:00000000009060C0                 db 'yyqmQLu5j+wJuTtS5m/nJ8mqayBQJMOQ+Ks5pa3pwfvVYd+lylrxgNaOJn07oNx+',0Ah
.noptrdata:00000000009060C0                 db 'xt/Hinpi6IFSQMB6EqUMON8CAwEAAQ==',0Ah
.noptrdata:00000000009060C0                 db '-----END PUBLIC KEY-----',0Ah,0
.noptrdata:00000000009063E2                 db    0
.noptrdata:00000000009063E3                 db    0
.noptrdata:00000000009063E4                 db    0
.noptrdata:00000000009063E5                 db    0
```

Public key for encryption

شکل ۵- کلید عمومی RSA در Go باینری

در شکل زیر داده‌ها بعد از رمزنگاری با RSA نشان داده شده است.

```
[0x0065ba34]> px 0x200 @ 0x000000c00013a200
- offset -   0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0xc00013a200  8497 daf1 5069 b2d1 b969 7fed 2713 b952  ....Pi...i..'..R
0xc00013a210  e3be cec2 9bad b97e 2389 93f8 58c6 2067  .......~#...X. g
0xc00013a220  6aa7 0dea cd2b a5a5 dd71 9a34 ec73 3735  j....+...q.4.s75
0xc00013a230  53b2 65aa 1102 11a4 16c6 eb4a a3d9 a875  S.e........J...u
0xc00013a240  8cee e967 2a48 28da 2a95 a096 4831 af36  ...g*H(.*...H1.6
0xc00013a250  4029 c249 2fff bdf7 2c9a f979 3170 c26d  @).I/...,..y1p.m
0xc00013a260  4515 5b24 83f1 759a ae6e b386 0282 1311  E.[$..u..n......
0xc00013a270  5506 e7ea e53a ea91 1830 62b8 6286 9afc  U....:...0b.b...
0xc00013a280  9c71 0cda 19fe 786b e87d 4629 d568 f955  .q....xk.}F).h.U
0xc00013a290  8d57 befa ede4 bad2 819b 0254 ee87 8530  .W.........T...0
0xc00013a2a0  aad6 57a6 37fa a986 4335 d925 1a2b f98f  ..W.7...C5.%.+..
0xc00013a2b0  55f2 0c9b 5320 6a54 e6c7 06b8 2e91 a5bf  U...S jT........
0xc00013a2c0  e8f7 89c9 6219 7432 5f52 dda0 35de 7d1b  ....b.t2_R..5.}.
0xc00013a2d0  b5b3 b4c0 023f bbb1 4c02 a695 016b 685f  .....?..L....kh_
0xc00013a2e0  acaa 8e2a d2be 82a3 067d 0059 cf0d fdf2  ...*.....}.Y....
0xc00013a2f0  8dbf 49ca 9227 4b57 ca5a 1817 7c4a d171  ..I..'KW.Z..|J.q
0xc00013a300  ec8c d5e0 13cb 064a acc4 c0ad bf60 3484  .......J.....`4.
0xc00013a310  6797 96b4 0be4 014b fb66 29ec a33f dc08  g......K.f)..?..
0xc00013a320  a93a a216 0486 f229 3351 abc0 8f1e ecaa  .:.....)3Q......
0xc00013a330  8ec3 4d24 3e6d 4035 c93c 2fdd ac95 95e2  ..M$>m@5.</.....
0xc00013a340  6ce6 c3c1 5265 57de fb6f e29f dfee 8e1e  l...ReW..o......
0xc00013a350  8328 d1e1 f910 32ba 1273 aab0 8f72 0065  .(....2..s...r.e
0xc00013a360  feb0 5de9 434d e4bd 4cd9 9b0e ced5 0796  ..].CM..L.......
0xc00013a370  940a cfc7 903e 138f 8aa6 fcd8 3134 8f2c  .....>......14.,
0xc00013a380  0c70 b90e cd72 0ccf 5c8a cdd5 cef4 03c2  .p...r..\.......
0xc00013a390  7f52 fabf fdd9 56ff a9be 1b94 7281 dc0c  .R....V.....r...
0xc00013a3a0  4903 b144 f410 6cc8 c88c 2e0b 07a8 f9bf  I..D..l.........
0xc00013a3b0  31b7 f64f d9a0 e3b4 afd0 d1b7 7dcb 3797  1..O........}.7.
0xc00013a3c0  c639 66d3 8c2c c443 e7ad 3e1c 8166 7746  .9f..,.C..>..fwF
0xc00013a3d0  95cf e7bd b048 ed83 e725 d359 9a54 ed5c  .....H...%.Y.T.\
0xc00013a3e0  45ca 13c2 f381 3cbe 4930 843c c192 3727  E.....<.I0.<..7'
0xc00013a3f0  daa7 b034 2942 0d6c 598b a8c9 38eb e18d  ...4)B.lY...8...
[0x0065ba34]>
```

Encrypted AES key

شکل ۶- کلید رمزنگاری شده AES پس از استفاده از رمزنگاری RSA

در مرحله بعد، تابع EncodeToString را در بسته Golang encoding/base64 برای رمزگذاری داده‌های قبلاً رمزگذاری شده با الگوریتم base64 فراخوانی می‌کند.

شکل ۷- کلید AES رمزگذاری شده با Base64

سپس یک بافر برای فایل README رمزگشایی شده، که در شکل ۸ نشان داده شده است.

شکل ۸- بافر فایل README رمزگشایی شده

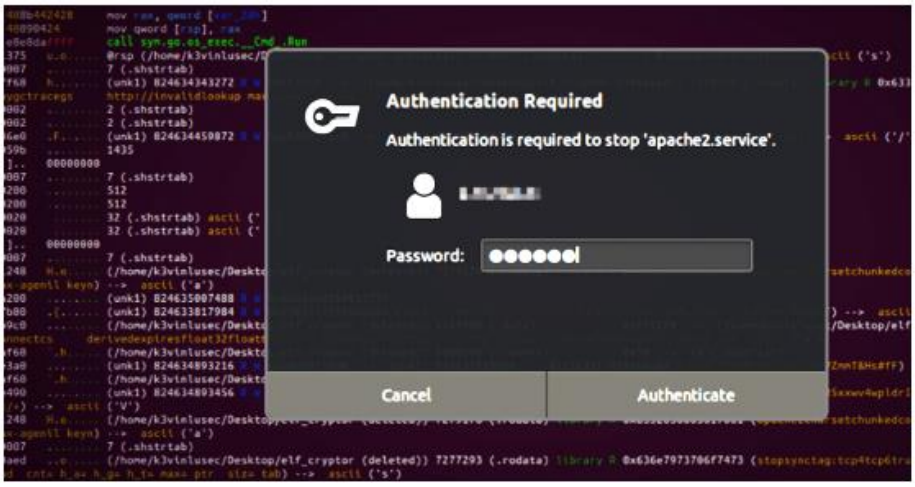
همانطور که در شکل (۸) مشاهده می‌کنید کلید رمزگذاری شده AES در فایل رمزگشایی شده readme با رمزگذاری Base64 نوشته شده است.

قبل از رمزنگاری فایل‌ها توسط باج افزار، مطابق شک زیر لیست فرآیندها را با صدور دستورات " service stop [pname]" یا "systemctl stop [pname]" از بین می برد.

```
.data:000000000092B9C0 off_92B9C0      dq offset aApache2      ; DATA XREF: .data:off_9270C0↑o
.data:000000000092B9C0                                         ; "apache2"
.data:000000000092B9C8                 dq 7
.data:000000000092B9D0                 dq offset aNginx        ; "nginx"
.data:000000000092B9D8                 dq 5
.data:000000000092B9E0                 dq offset aMysqld       ; "mysqld"
.data:000000000092B9E8                 dq 6
.data:000000000092B9F0                 dq offset aPhpFpm       ; "php-fpm"
.data:000000000092B9F8                 dq 7
.data:000000000092BA00                 dq offset aPhp5Fpm      ; "php5-fpm"
.data:000000000092BA08                 dq 8
.data:000000000092BA10                 dq offset aMysql        ; "mysql"
.data:000000000092BA18                 dq 5
.data:000000000092BA20                 dq offset aPostgresql   ; "postgresql"
.data:000000000092BA28                 dq 0Ah
```

شکل ۹- سرویس‌های متوقف شده

هنگامی که برای متوقف کردن apache2.service تلاش می‌شود است، پیغام با عنوان "تأیید هویت لازم" را ایجاد می‌کند تا به کاربر دستور دهد رمز عبور سیستم را برای انجام این عملیات وارد کند.

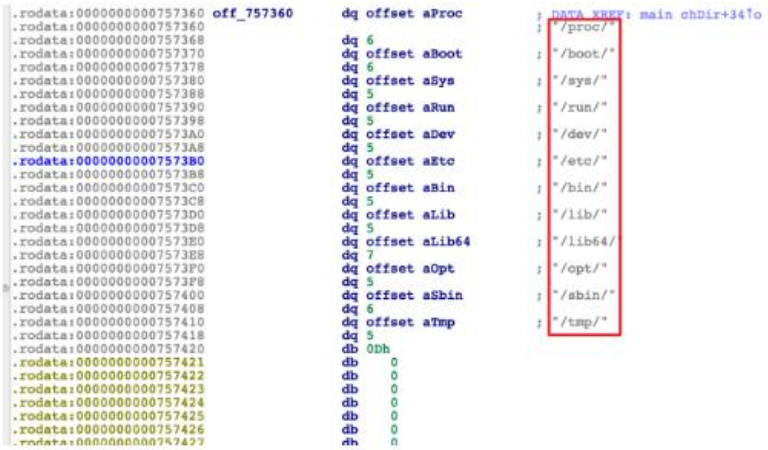


شکل ۱۰- استخراج درخواست در ping_test

سرانجام ، بدافزار با فراخوانی تابع Walk (رشته اصلی ، walkFn WalkFunc) در بسته Golang "path / filepath" و سپس فایل های رمزنگاری شده، شروع به مرور فهرست اصلی می کند.

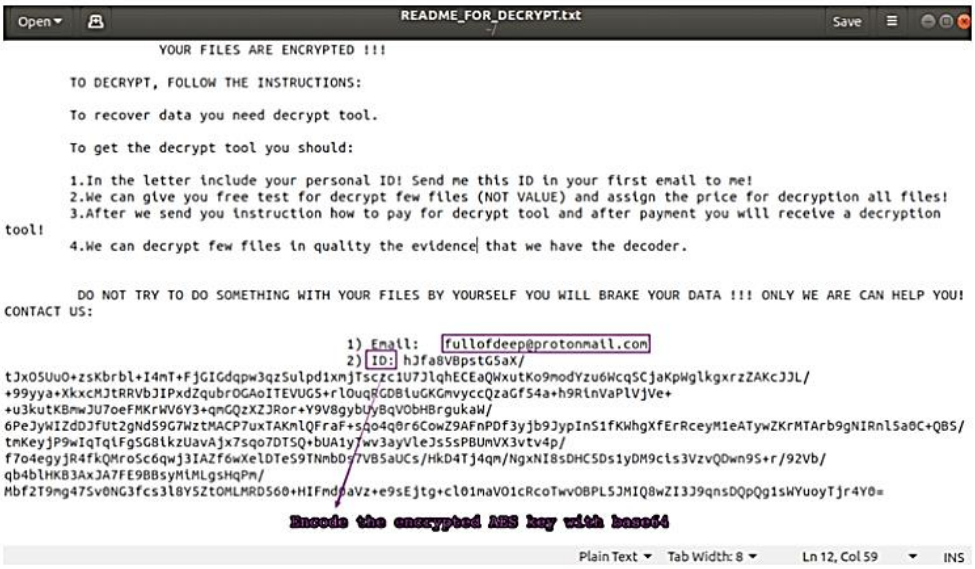
شکل ۱۱- فهرست‌های اصلی مرور و فایل‌ها رمزگذاری شده

این بدافزار همچنین دارای لیست سیاه فهرست‌ها برای رمزگذاری است. در شکل زیر لیست سیاه فهرست نشان داده شده است.



شکل ۱۲-لیست سیاه فهرست‌ها برای رمزنگاری

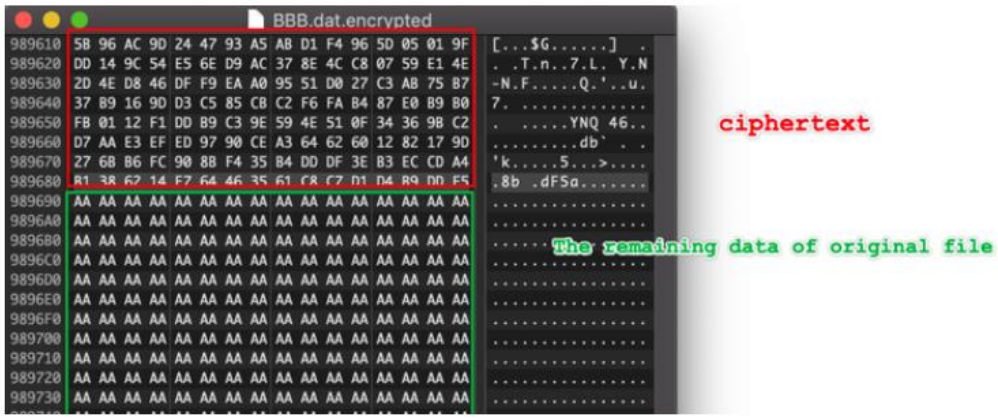
این بدافزار با استفاده از الگوریتم AES-256-CFB، فایل‌ها را رمزگذاری می‌کند، و فایل‌های رمزگذاری شده دارای اسمی هستند که نام اصلی را با پسوند ".encryelled" به هم الحاق می‌کند. فایل README برای رمزگشایی در شکل ۱۳ نشان داده شده است.



شکل ۱۳- فایل README برای رمزگشایی

تابع EncFile() برای رمزگذاری فایل‌ها استفاده می‌شود. در ابتدا اندازه فایل رمزگذاری می‌شود. اگر اندازه فایل کمتر از x9868800 (۱،۰۰۰،۰۰۰) بایت باشد، تمام داده‌های فایل را با یک الگوریتم AES-256-CFB رمزگذاری می‌کند. در غیر این صورت، اولین بایت x9868800 (۱،۰۰۰،۰۰۰) داده را می‌خواند و آنها را رمزگذاری می‌کند، سپس داده‌های باقیمانده فایل اصلی را در انتهای فایل رمزگذاری شده کپی می‌کند.

شکل ۱۴- اندازه فایل را برای رمزنگاری بررسی می‌کند



شکل ۱۵- داده‌ها پس از رمزنگاری برای فایل‌های بزرگتر از x989680۰ در بایت

نتیجه گیری

طبق تجزیه و تحلیل‌هایی که آزمایشگاه FortiGuard انجام داده است، این باج افزار زیاد پیچیده نیست و ممکن است در مرحله توسعه اولیه باشد. باید توجه داشته باشیم که بدافزارها بیشتر با Golang در حال تولید هستند و Fortinet بدافزارهایی که به این زبان برنامه نویسی جدید نوشته شده است را مرتباً کنترل و فیلتر خواهدکرد.

راه حل

فایل ELF مخرب توسط سرویس FortiGuard AntiVirus با عنوان "ELF / Cryptor.B! tr" شناسایی شده است.

مراجع

1 “Analysis of A New Golang Ransomware Targeting Linux Systems,” https://www.fortinet.com/blog/threat-research/new-golang-ransomware-targeting-linux-systems.html.