


Product ▾ Solutions ▾ Resources ▾ Open Source ▾ Enterprise ▾ Pricing



Sign in

Sign up

 BishopFox / **sliver** Public

 Notifications

 Fork 1.1k

 Star 8.5k

 Code

 Issues 187

 Pull requests 11

 Discussions


 Actions

 Projects


 Wiki

 Security

 Insights

 Files

79f2d48



Go to file

> .github

> client

> assets

> cli

> command

> alias

> armory

> backdoor

> beacons

> completers

> dllhijack

> environment

> exec

> extensions

> filesystem

> generate

> help

> hosts

> info

> jobs

> kill

> loot

> monitor

> network

> operators

> pivots

> portfwd

> prelude-operator

> privilege

> processes

> reaction


> reconfig

> registry

> screenshot

> sessions

> settings

sliver / client / command / commands.go 


History


Code


Blame

3299 lines (3052 loc) · 111 KB

Raw







1 package command

2

3 /\*

4 Sliver Implant Framework

5 Copyright (C) 2019 Bishop Fox

6

7 This program is free software: you can redistribute it and/or modify

8 it under the terms of the GNU General Public License as published by

9 the Free Software Foundation, either version 3 of the License, or

10 (at your option) any later version.

11

12 This program is distributed in the hope that it will be useful,

13 but WITHOUT ANY WARRANTY; without even the implied warranty of

14 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

15 GNU General Public License for more details.

16

17 You should have received a copy of the GNU General Public License

18 along with this program. If not, see <https://www.gnu.org/licenses/>.

19

20 -----

21 This file contains all of the code that binds a given string/flags/etc. to a

22 command implementation function.

23

24 Guidelines when adding a command:

25

26 \* Try to reuse the same short/long flags for the same paramenter,

27 e.g. "timeout" flags should always be -t and --timeout when possible.

28 Try to avoid creating flags that conflict with others even if you're

29 not using the flag, e.g. avoid using -t even if your command doesn't

30 have a --timeout.

31

32 \* Add a long-form help template to `client/help`

33

34 \*/

35

36 import (

37 "os"

38

39 "github.com/bishopfox/sliver/client/assets"

40 "github.com/bishopfox/sliver/client/command/alias"

41 "github.com/bishopfox/sliver/client/command/armory"

42 "github.com/bishopfox/sliver/client/command/backdoor"

43 "github.com/bishopfox/sliver/client/command/beacons"

44 "github.com/bishopfox/sliver/client/command/completers"

45 "github.com/bishopfox/sliver/client/command/dllhijack"

46 "github.com/bishopfox/sliver/client/command/environment"

47 "github.com/bishopfox/sliver/client/command/exec"

48 "github.com/bishopfox/sliver/client/command/extensions"

49 "github.com/bishopfox/sliver/client/command/filesystem"

50 "github.com/bishopfox/sliver/client/command/generate"

51 "github.com/bishopfox/sliver/client/command/help"

52 "github.com/bishopfox/sliver/client/command/hosts"

53 "github.com/bishopfox/sliver/client/command/info"







54 "github.com/bishopfox/sliver/client/command/jobs"

55 "github.com/bishopfox/sliver/client/command/kill"

56 "github.com/bishopfox/sliver/client/command/loot"

57 "github.com/bishopfox/sliver/client/command/monitor"

Page 1 of 45

- >  shell
- >  shikata-ga-nai
- >  socks
- >  tasks
- >  update
- >  use

```
57     github.com/bishopfox/sliver/client/command/monitor"
58     "github.com/bishopfox/sliver/client/command/network"
59     "github.com/bishopfox/sliver/client/command/operators"
60     "github.com/bishopfox/sliver/client/command/pivots"
61     "github.com/bishopfox/sliver/client/command/portfwd"
62     operator "github.com/bishopfox/sliver/client/command/prelude-operator"
63     "github.com/bishopfox/sliver/client/command/privilege"
64     "github.com/bishopfox/sliver/client/command/processes"
65     "github.com/bishopfox/sliver/client/command/reaction"
66     "github.com/bishopfox/sliver/client/command/reconfig"
67     "github.com/bishopfox/sliver/client/command/registry"
68     "github.com/bishopfox/sliver/client/command/screenshot"
69     "github.com/bishopfox/sliver/client/command/sessions"
70     "github.com/bishopfox/sliver/client/command/settings"
71     "github.com/bishopfox/sliver/client/command/shell"
72     sgn "github.com/bishopfox/sliver/client/command/shikata-ga-nai"
73     "github.com/bishopfox/sliver/client/command/socks"
74     "github.com/bishopfox/sliver/client/command/tasks"
75     "github.com/bishopfox/sliver/client/command/update"
76     "github.com/bishopfox/sliver/client/command/use"
77     "github.com/bishopfox/sliver/client/command/websites"
78     "github.com/bishopfox/sliver/client/command/wireguard"
79     "github.com/bishopfox/sliver/client/console"
80     consts "github.com/bishopfox/sliver/client/constants"
81     "github.com/bishopfox/sliver/client/licenses"
82     "github.com/desertbit/grumble"
83 )
84
85 const (
86     defaultTimeout = 60
87 )
88
89 // BindCommands - Bind commands to a App
90 func BindCommands(con *console.SliverConsoleClient) {
91
92     // Load Reactions
93     n, err := reaction.LoadReactions()
94     if err != nil && !os.IsNotExist(err) {
95         con.PrintErrorf("Failed to load reactions: %s\n", err)
96     } else if n > 0 {
97         con.PrintInfof("Loaded %d reaction(s) from disk\n", n)
98     }
99
100    // Load Aliases
101    aliasManifests := assets.GetInstalledAliasManifests()
102    n = 0
103    for _, manifest := range aliasManifests {
104        _, err = alias.LoadAlias(manifest, con)
105        if err != nil {
106            con.PrintErrorf("Failed to load alias: %s\n", err)
107            continue
108        }
109        n++
110    }
111    if 0 < n {
112        if n == 1 {
113            con.PrintInfof("Loaded %d alias from disk\n", n)
114        } else {
115            con.PrintInfof("Loaded %d aliases from disk\n", n)
116        }
117    }
118 }
```





























































































```
3226         Help:      "Install an extension from a local directory or .tar.gz file
3227 LongHelp:  help.GetHelpFor([]string{consts.ExtensionsStr, consts.Instal
3228 HelpGroup: consts.SliverHelpGroup,
3229 Run: func(ctx *grumble.Context) error {
3230     con.Println()
3231     extensions.ExtensionsInstallCmd(ctx, con)
3232     con.Println()
3233     return nil
3234 },
3235 Args: func(a *grumble.Args) {
3236     a.String("path", "path to the extension .tar.gz or directory")
3237 },
3238 Completer: func(prefix string, args []string) []string {
3239     return completers.LocalPathCompleter(prefix, args, con)
3240 },
3241 })
3242
3243 extensionCmd.AddCommand(&grumble.Command{
3244     Name:      consts.RmStr,
3245     Help:      "Remove an installed extension",
3246     LongHelp:  help.GetHelpFor([]string{consts.ExtensionsStr, consts.RmStr}
3247 HelpGroup: consts.SliverHelpGroup,
3248 Run: func(ctx *grumble.Context) error {
3249     con.Println()
3250     extensions.ExtensionsRemoveCmd(ctx, con)
3251     con.Println()
3252     return nil
3253 },
3254 Args: func(a *grumble.Args) {
3255     a.String("name", "the command name of the extension to remove")
3256 },
3257 Completer: func(prefix string, args []string) []string {
3258     return extensions.ExtensionsCommandNameCompleter(prefix, args,
3259 },
3260 })
3261
3262 con.App.AddCommand(extensionCmd)
3263
3264 // [ Prelude's Operator ] -----
```

```
3265         operatorCmd := &grumble.Command{
3266             Name:      consts.PreludeOperatorStr,
3267             Help:       "Manage connection to Prelude's Operator",
3268             LongHelp:   help.GetHelpFor([]string{consts.PreludeOperatorStr}),
3269             HelpGroup:  consts.GenericHelpGroup,
3270             Run: func(ctx *grumble.Context) error {
3271                 con.Println()
3272                 operator.OperatorCmd(ctx, con)
3273                 con.Println()
3274                 return nil
3275             },
3276         }
3277         operatorCmd.AddCommand(&grumble.Command{
3278             Name:      consts.ConnectStr,
3279             Help:       "Connect with Prelude's Operator",
3280             LongHelp:   help.GetHelpFor([]string{consts.PreludeOperatorStr, consts.C
3281             HelpGroup:  consts.GenericHelpGroup,
3282             Run: func(ctx *grumble.Context) error {
3283                 con.Println()
3284                 operator.ConnectCmd(ctx, con)
3285                 con.Println()
3286                 return nil
3287             },
3288             Args: func(a *grumble.Args) {
3289                 a.String("connection-string", "connection string to the Operato
3290             },
3291             Flags: func(f *grumble.Flags) {
3292                 f.Bool("s", "skip-existing", false, "Do not add existing sessio
3293                 f.String("a", "aes-key", "abcdefghijklmnopqrstuvwxy012345", "A
3294                 f.String("r", "range", "sliver", "Agents range")
3295             },
3296         })
3297
3298         con.App.AddCommand(operatorCmd)
3299     }
```