

Key Space Brute Force

+ Esse é o tipo de brute force que vai esgotar todas as possibilidades até encontrar a senha. Muito útil quando o padrão utilizado é identificado

+ Exemplo didático:

PIN 4 dígitos

Senha 6 dígitos numéricos

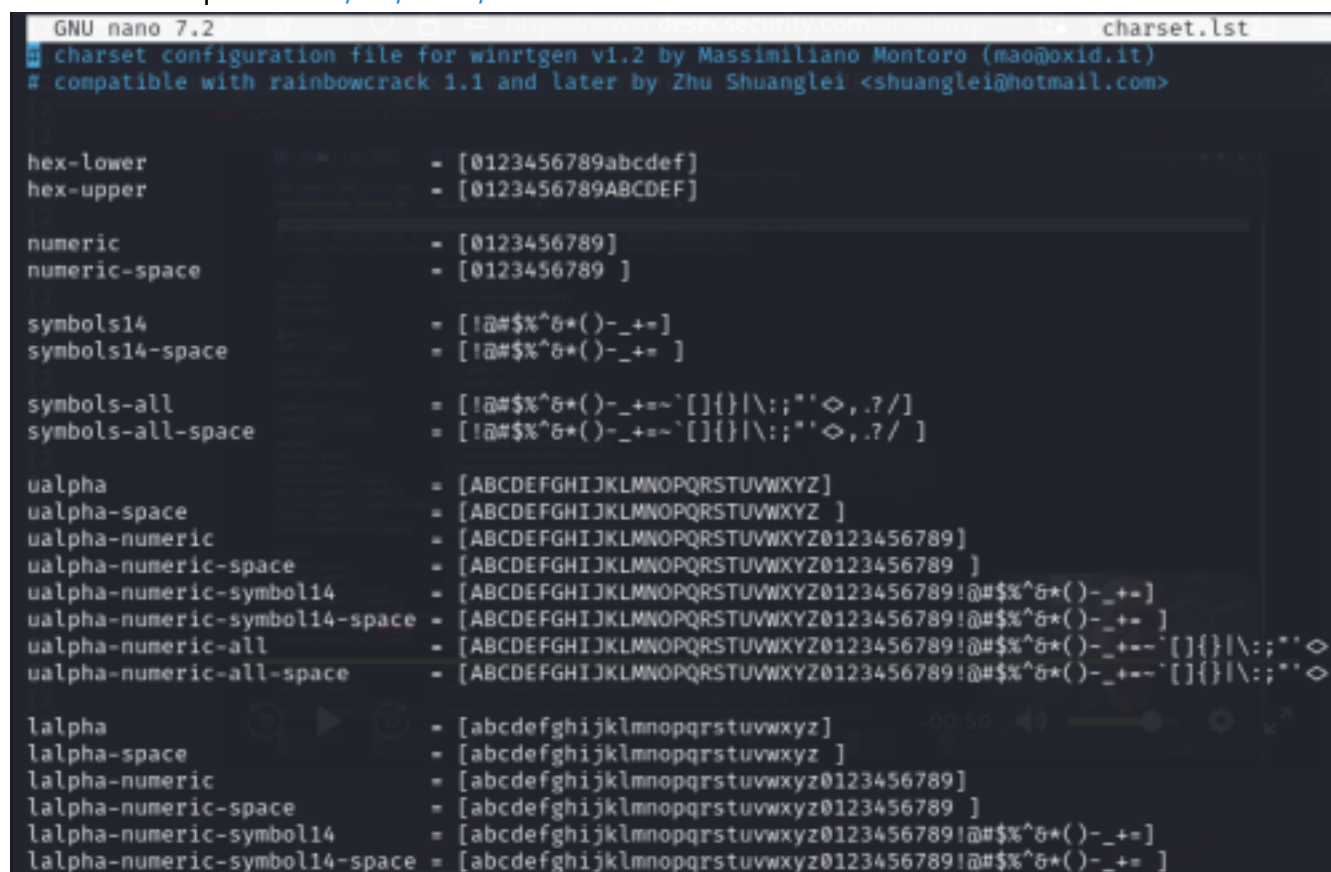
+ Exemplo real

Caso: Token (A2F3C5)

+ Para montar todas as senhas possíveis de números de 4 dígitos?

```
crunch 4 4 0123456789 -o pin
```

Outra opção é puxarmos a informação dos caracteres direto dos grupos da charset.lst que fica na [/usr/share/crunch](#)



```
GNU nano 7.2 charset.lst
charset configuration file for winrtgen v1.2 by Massimiliano Montoro (mao@oxid.it)
# compatible with rainbowcrack 1.1 and later by Zhu Shuanglei <shuanglei@hotmail.com>

hex-lower      = [0123456789abcdef]
hex-upper      = [0123456789ABCDEF]

numeric        = [0123456789]
numeric-space  = [0123456789 ]

symbols14       = [!@#%^&*()-_+=]
symbols14-space = [!@#%^&*()-_+= ]

symbols-all    = [!@#%^&*()-_+=~`[]{}|\:;'"<,./ /]
symbols-all-space = [!@#%^&*()-_+=~`[]{}|\:;'"<,./ / ]

ualpha          = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
ualpha-space    = [ABCDEFGHIJKLMNOPQRSTUVWXYZ ]
ualpha-numeric  = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
ualpha-numeric-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ]
ualpha-numeric-symbol14 = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%^&*()-_+=]
ualpha-numeric-symbol14-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%^&*()-_+= ]
ualpha-numeric-all = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%^&*()-_+=~`[]{}|\:;'"<,./ /]
ualpha-numeric-all-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#%^&*()-_+=~`[]{}|\:;'"<,./ / ]

lalpha          = [abcdefghijklmnopqrstuvwxyz]
lalpha-space    = [abcdefghijklmnopqrstuvwxyz ]
lalpha-numeric  = [abcdefghijklmnopqrstuvwxyz0123456789]
lalpha-numeric-space = [abcdefghijklmnopqrstuvwxyz0123456789 ]
lalpha-numeric-symbol14 = [abcdefghijklmnopqrstuvwxyz0123456789!@#%^&*()-_+=]
lalpha-numeric-symbol14-space = [abcdefghijklmnopqrstuvwxyz0123456789!@#%^&*()-_+= ]
```

```
crunch 4 4 -f charset.lst numeric -o pin
```