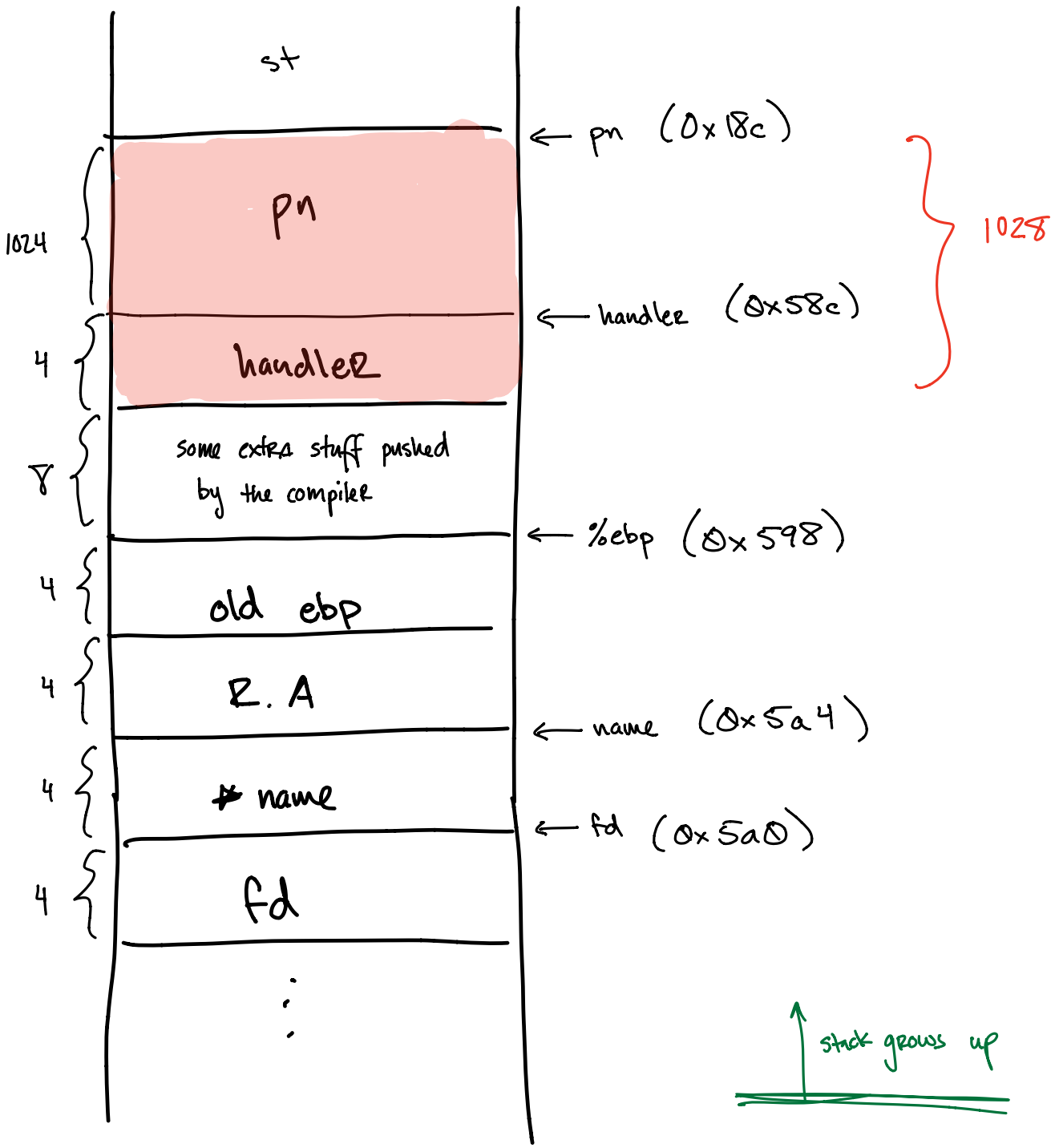
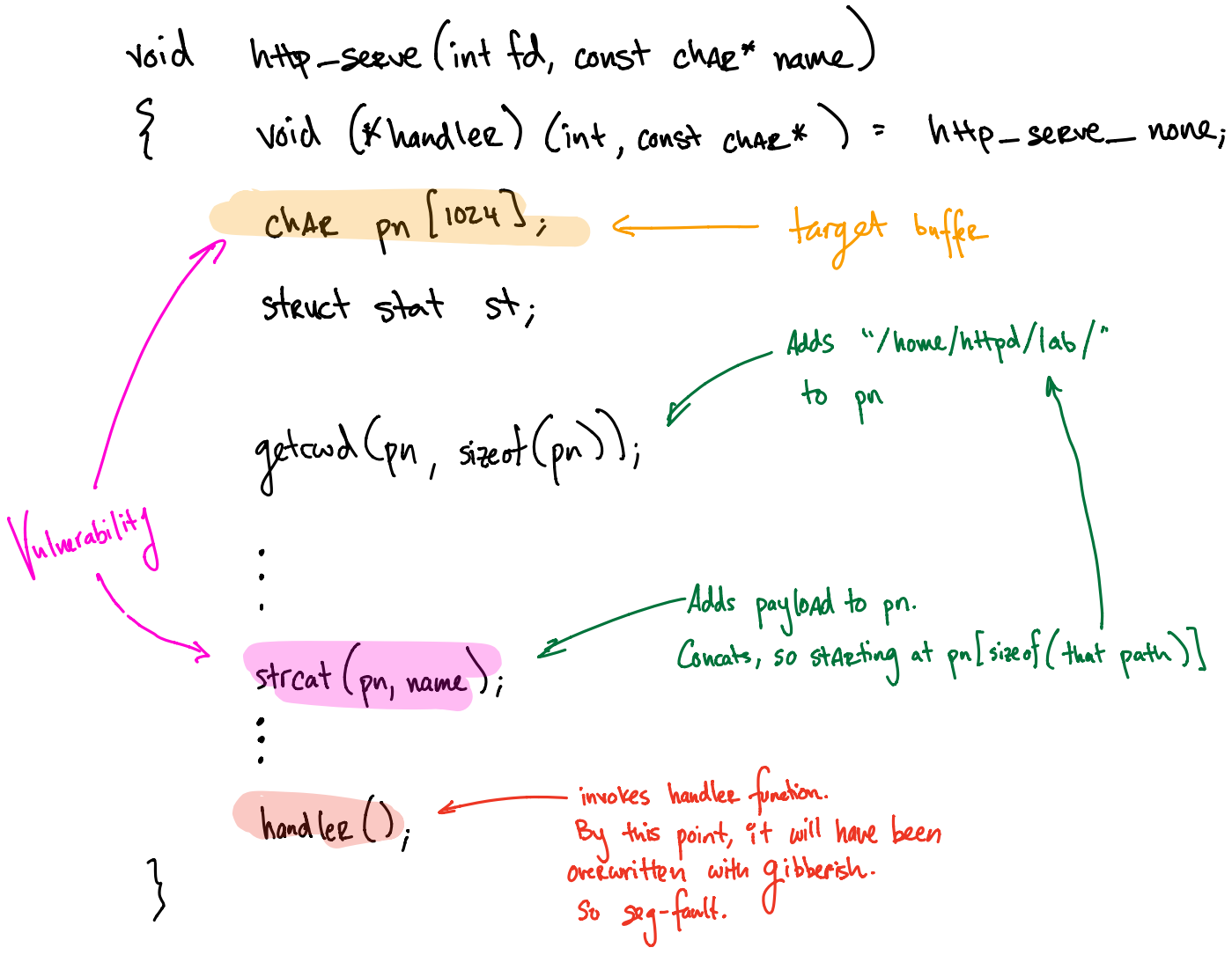
http.c : 273

Exploit-2A.py

overflow target

```
void  http_serve(int fd, const char* name)
{     void (*handler)(int, const char*) = http_serve_none;
      char pn[1024];
      struct stat st;
      ...
}
```

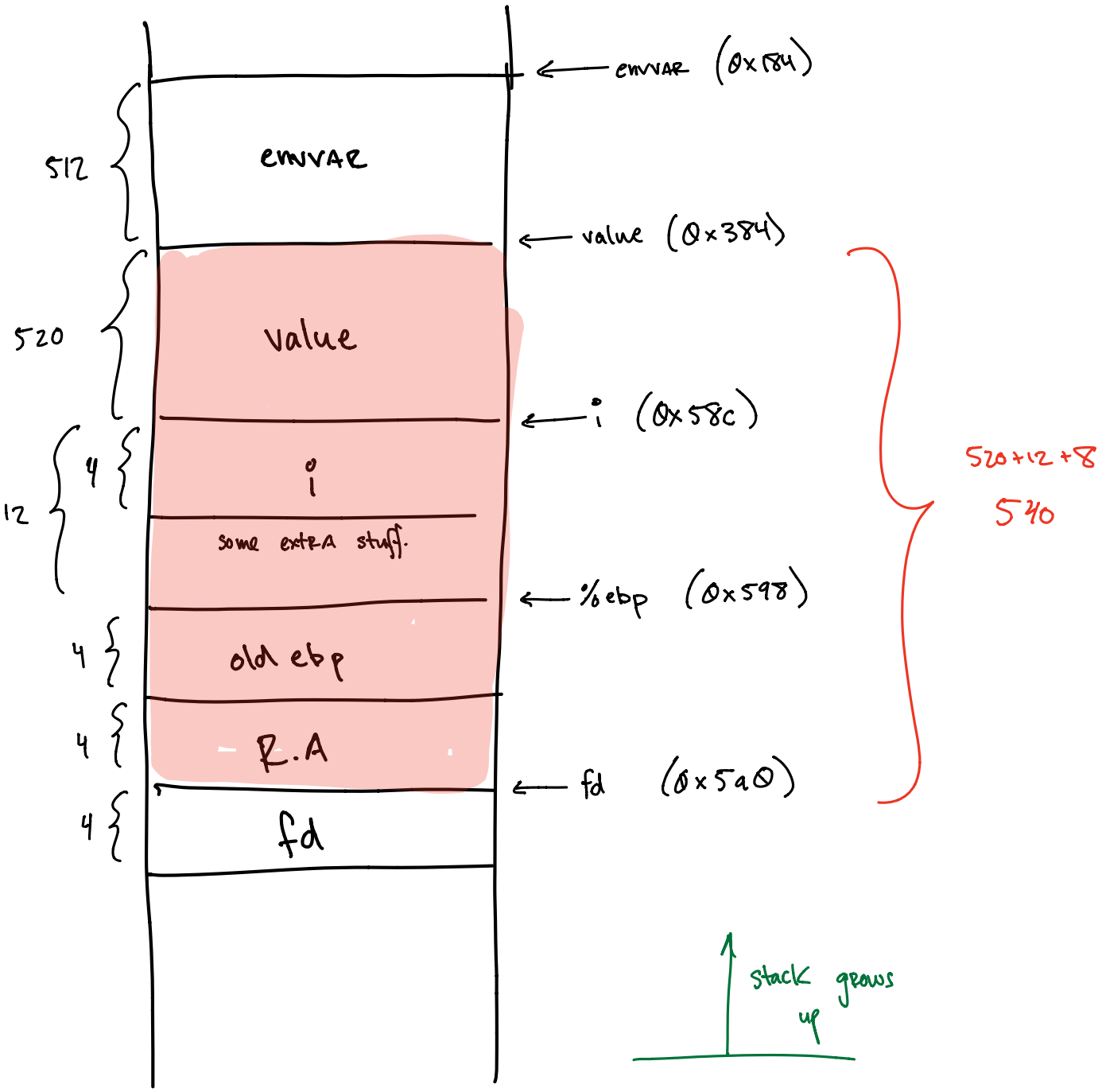



void http_serve(int fd, const char* name)
{
void (*handler)(int, const char*) = http_serve_none;
char pn[1024];
target buffer
struct stat st;
Adds "/home/httpd/lab/" to pn
getcwd(pn, sizeof(pn));
Vulnerability
Adds payload to pn.
Concats, so starting at pn[sizeof(that path)]
strcat(pn, name);
handler();
invokes handler function.
By this point, it will have been overwritten with gibberish.
So seg-fault.
}

http.c: 120

Exploit-28.py

```
const char * http_request_headers(int fd)
{
    static char buf[8192];
    int i;
    char value[512];
    char envvar[512];
    ...
}
```



```
const char *http_request_headers(int fd)
{
    static char buf[8192];
    int i;
    char value[512];
    char envvar[512];
    ...
    for (;;)
    {
        ...
        url_decode(value, sp);
        ...
    }
}
```

sp stores the payload minus 'HOST:_'.

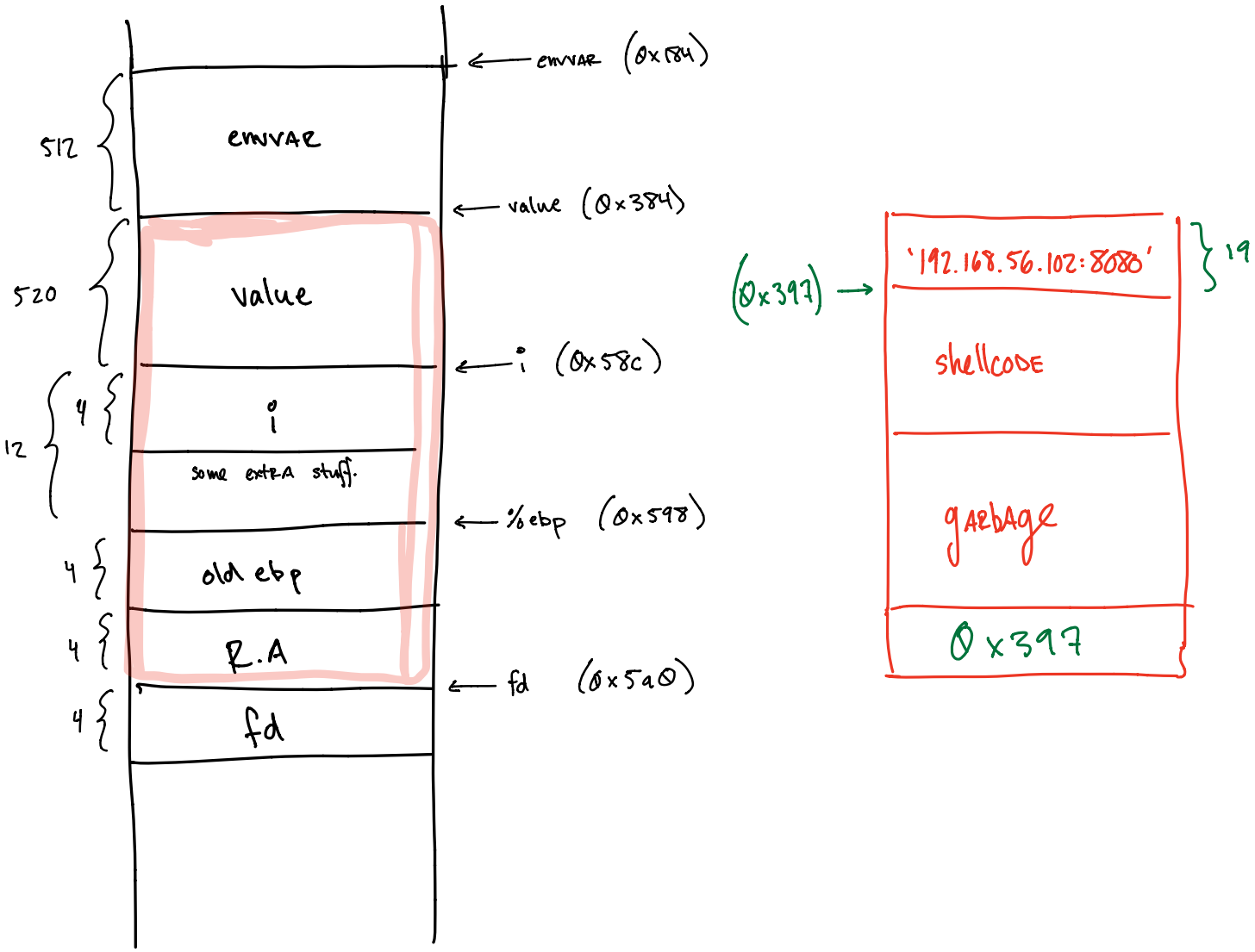
```
void url_decode(char *dst, const char *src)
{
    loop through src:
        if src[i] is '%' : do something.
        elif src[i] is '+' : do something.
        else:
            *dst = *src
            src++
            if (*dst == '\0')
                break;
}
```

payload won't include these.

Copies from src & then does a check for the null terminator.

NOT restricted by the bounds of dst.

Exploit-3.py


envvar (0x184)
512
envvar
value (0x384)
520
value
i (0x58c)
4
i
12
some extra stuff.
%ebp (0x598)
4
old ebp
4
R.A
fd (0x5a0)
4
fd
(0x397)
'192.168.56.102:8080'
19
shellcode
garbage
0x397