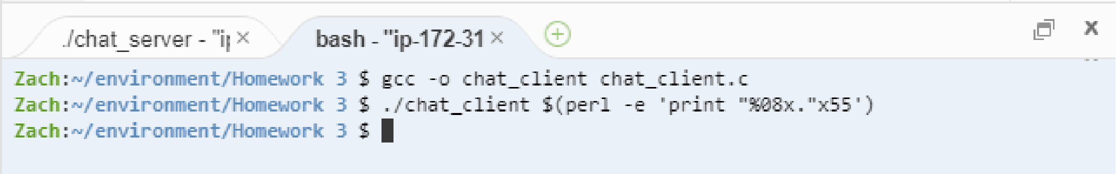
CWE-865 Demo with Mitigation

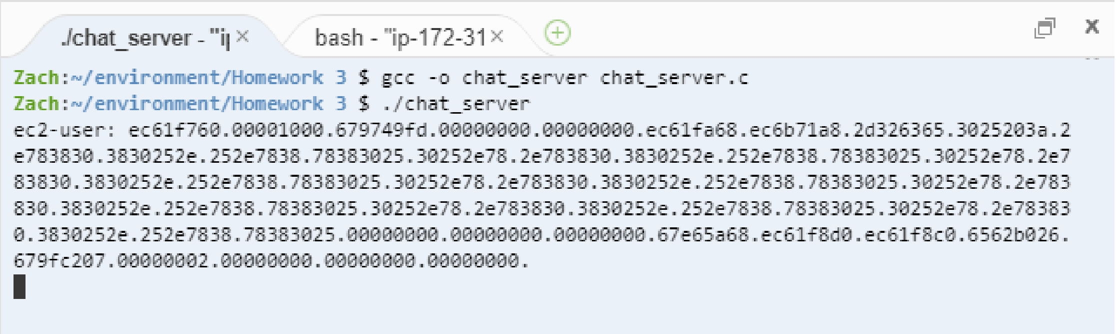
Zachary Smith
SDEV 325 6381 – Homework 3
14 September 2018

Example 1: Format String

To demonstrate the effectiveness of a CWE-134 attack on C's *printf()*, I wrote a simple network chat application. chat_server listens on IRC port 6665 for incoming messages, and reproduces them without checking for string format.



```
Zach:~/environment/Homework 3 $ gcc -o chat_client chat_client.c
Zach:~/environment/Homework 3 $ ./chat_client $(perl -e 'print "%08x."x55')
Zach:~/environment/Homework 3 $ 
```



```
Zach:~/environment/Homework 3 $ gcc -o chat_server chat_server.c
Zach:~/environment/Homework 3 $ ./chat_server
ec2-user: ec61f760.00001000.679749fd.00000000.00000000.ec61fa68.ec6b71a8.2d326365.3025203a.2
e783830.3830252e.252e7838.78383025.30252e78.2e783830.3830252e.252e7838.78383025.30252e78.2e7
83830.3830252e.252e7838.78383025.30252e78.2e783830.3830252e.252e7838.78383025.30252e78.2e783
830.3830252e.252e7838.78383025.30252e78.2e783830.3830252e.252e7838.78383025.30252e78.2e78383
0.3830252e.252e7838.78383025.00000000.00000000.00000000.67e65a68.ec61f8d0.ec61f8c0.6562b026.
679fc207.00000002.00000000.00000000.00000000.
```

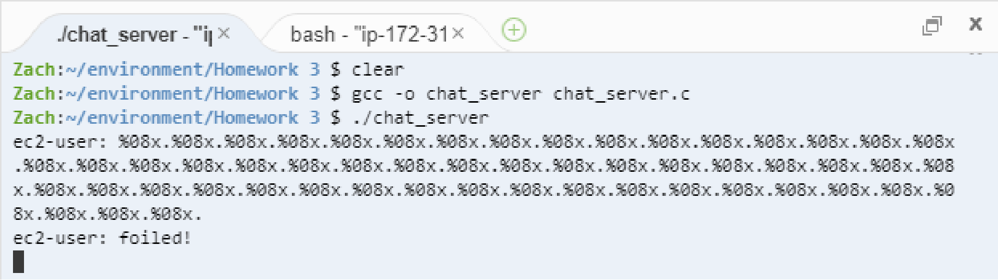
If the client were to use other format parameters like %s or %n, she/he could respectively read and write at arbitrary memory locations. This example illustrates access to lower regions of the stack. As it currently stands, chat_client can't see the output text. I assume a future combination of the two programs before they are more than a proof of concept.

The call to *recv()* in chat_server allows a message buffer of 4kB, far greater than the local *message* size, which is 512B. A client can take advantage of the size disparity to overflow *message*, making this weakness viable. I can soften the impact by assigning message buffer to a constant.

recv(client_skt, message, 4096, 0) → **recv(client_skt, message, MSGBUF, 0)**

Also, chat_server's use of *printf()* is incorrect. Instead of mindlessly printing user-controlled input, I will hardcode the output by predefining any format parameters so the user can't set his/her own.

printf(message) → **printf("%s", message)**



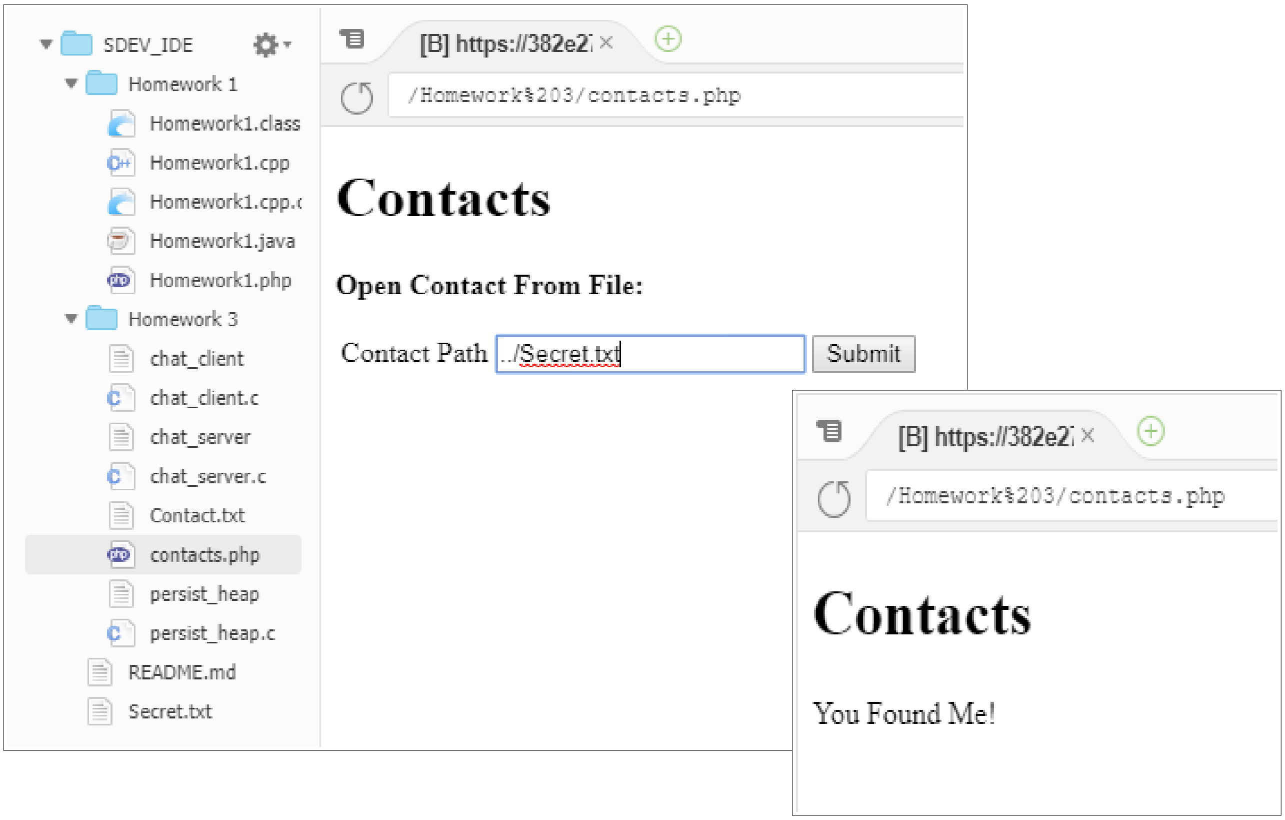
```
Zach:~/environment/Homework 3 $ clear
Zach:~/environment/Homework 3 $ gcc -o chat_server chat_server.c
Zach:~/environment/Homework 3 $ ./chat_server
ec2-user: %08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x
.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08
x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%0
8x.%08x.%08x.%08x.
ec2-user: foiled!
```

CWE-865 Demo with Mitigation

Zachary Smith
SDEV 325 6381 – Homework 3
14 September 2018

Example 2: Path Traversal

Back to PHP to demonstrate CWE-22. Contacts.php was built to retrieve a file containing contact information inside the web page's scope, such as Contact.txt. But by specifying a directory above the intended content, a visitor can read files not intended for ec2-user (default apache user).



/Homework%203/contacts.php

Contacts

Open Contact From File:

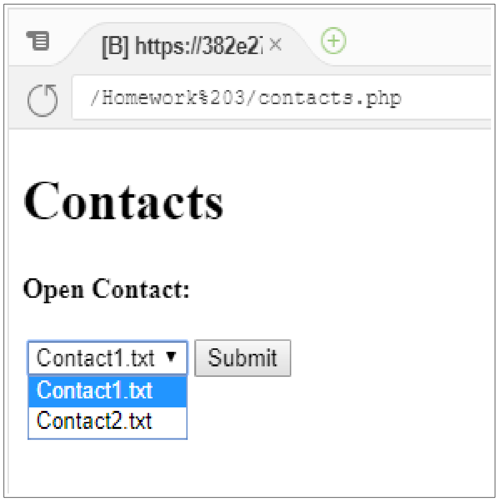
Contact Path ../Secret.txt

/Homework%203/contacts.php

Contacts

You Found Me!

Since the input is finite, I can forgo executing user controlled commands. A drop down menu is the better choice. In the future it will be better to implement a contact as a SQL entry or a JSON file, and avoid executing commands entirely. Also Cloud9 doesn't have a separate web user with limited permissions, making this kind of attack viable from other avenues. CWE-78 is an adjacent concern.



/Homework%203/contacts.php

Contacts

Open Contact:

Contact1.txt ▼
Contact1.txt
Contact2.txt

CWE-865 Demo with Mitigation

Zachary Smith

SDEV 325 6381 – Homework 3

14 September 2018

Sources

Erickson, John (2008). **Hacking: The Art of Exploitation, 2nd Ed**. San Francisco, CA: No Starch Press, Inc.

Linux IPv4 Protocol Implementation. Retrieved from: https://linux.die.net/man/7/ip.

Prodoromou, Agathoklis (June 2016). An Introduction to Web Shells. Retrieved from: https://www.acunetix.com/blog/articles/introduction-web-shells-part-1/.