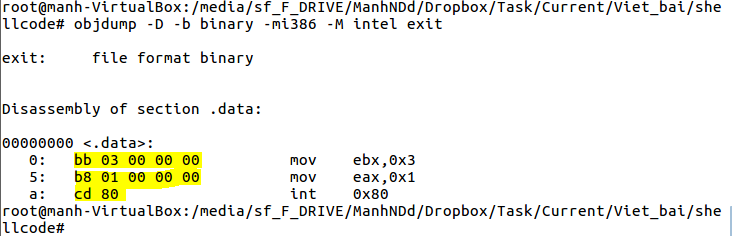
Hôm trước mình có một bài viết hướng dẫn cách viết shellcode trên Linux. Cơ mà một số bạn vẫn chưa hiểu bản chất shellcode, nên hôm nay mình viết một bài giải thích cụ thể hơn.

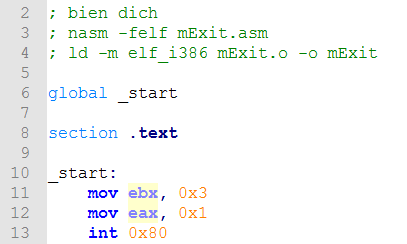
Shellcode thực chất chỉ là một dãy các lệnh thực thi – hay là một đoạn mã máy, được viết dưới dạng một xâu. Nó không phải là một chương trình thực thi hoàn chỉnh. Hãy cùng phân biệt một file ELF và một con shellcode:

Như trong hình dưới đây, ta dump ra con shellcode exit:

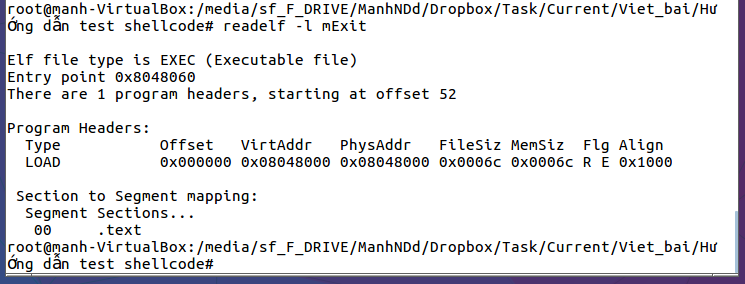


Con shellcode exit này đơn giản chỉ là xâu “\xbb\x03\x00\x00\x00\xb8\x01\x00\x00\x00\xcd\x80”. Nó chỉ gồm có 12 byte.

Còn khi bạn lập trình assembly hay lập trình C, thì đều là tạo ra một file thực thi có định dạng ELF. Nó có kích thước tới hàng KBbyte đến MByte. Ví dụ, với chương trình asm rất nhỏ sau:



Khi mình biên dịch ra (dùng các lệnh ở dòng 3, 4), nó cũng có kích thước tới 488Byte và là một file cấu trúc ELF hoàn chỉnh. Dùng lệnh readelf ta thấy nó có một segment text duy nhất:



Do sự khác biệt trên, khi mình chạy testShellcode (con shellcode trên là 32bit, nên chạy testShellcode32. Ở đây, testShellcode là chương trình mình nhắc tới trong bài viết shellcode <https://whitehat.vn/threads/huong-dan-viet-shellcode.8472/>), ta không thể sử dụng nó để chạy file mExit được.

Trang shell-storm có database của rất nhiều shellcode Windows, Linux, trên đủ các kiến trúc 32bit, 64bit, ARM, Intel,… Mình hay dùng con shellcode gọi /bin/sh dài [23 bytes](http://shell-storm.org/shellcode/files/shellcode-827.php) *của Hamza Megahed:*



File trên cũng có hoạt động tương tự như file testShellcode của mình, nhưng nó truyền hẳn shellcode cho chương trình ở dạng xâu, còn mình thì đọc vào từ file.

Sau đây, mình sẽ demo việc test một con shellcode bằng chương trình testShellcode.

[Video demo]