Abusing PHP Sockets For Fun And Profit

OWASP April 2008

Rodrigo Marcos
rodrigo.marcos@secforce.co.uk
www.secfoce.co.uk

secforce

Agenda

- Introduction
- Windows Sockets Reuse
- Apache Web Server
- PHP Socket Library
- Vectors of Attack
- Demo
- Conclusions

Introduction

- Lot's of research on Apache
- Lot's of research on PHP applications
- This talk will focus on PHP functionality from an offensive point of view
- Interesting vectors of attack re-using Windows sockets

Windows Socket Reuse

- The SO_REUSEADDR socket option allows a socket to forcibly bind to a port in use by another socket
- The behaviour is non-deterministic when used on the same interface
- However, we can take advance of Windows interface precedence: Local interface precede 0.0.0.0 and makes the attack reliable

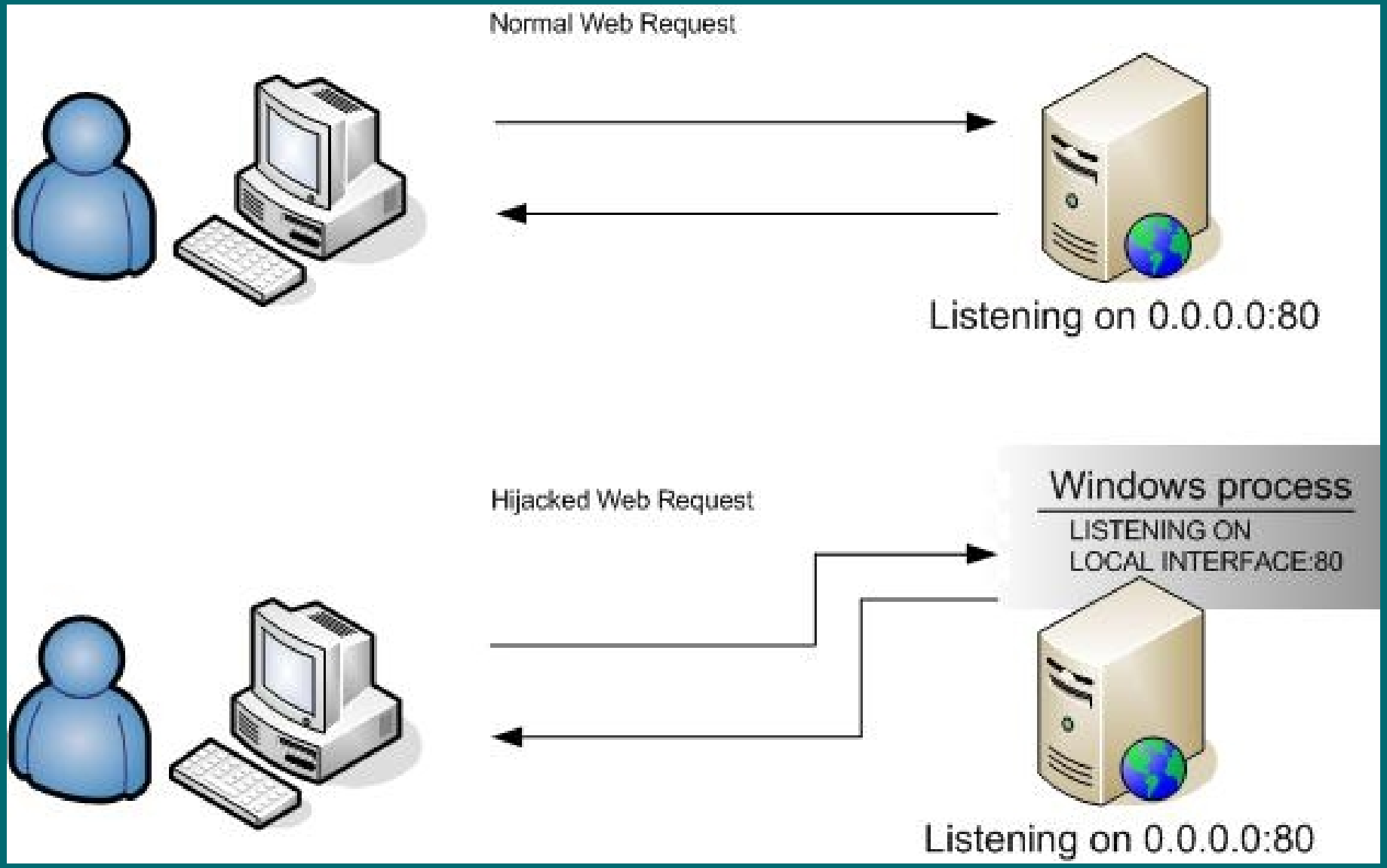
Windows Socket Reuse

- Windows Sockets introduced the SO_EXCLUSIVEADDRUSE socket option and recommends its use on server applications

  http://msdn2.microsoft.com/en-us/library/ms740621(VS.85).aspx

- Enhanced socket security was added with the release of Windows Server 2003 and makes sockets not in a shareable state by default

Windows Socket Reuse



Apache Web Server

- Apache threads by default run as:
  - www-data/httpd/apache on *nix systems (low priv)
  - SYSTEM on Microsoft Windows systems
- By default listens on 0.0.0.0
- It is not compiled with SO_EXCLUSIVEADDRUSE

Apache Web Server

- httpd-2.2.8/server/listen.c

```
#ifdef WIN32
    /* I seriously doubt that this would work on Unix; I have doubts that
     * it entirely solves the problem on Win32.  However, since setting
     * reuseaddr on the listener -prior- to binding the socket has allowed
     * us to attach to the same port as an already running instance of
     * Apache, or even another web server, we cannot identify that this
     * port was exclusively granted to this instance of Apache.
     *
     * So set reuseaddr, but do not attempt to do so until we have the
     * parent listeners successfully bound.
     */
    stat = apr_socket_opt_set(s, APR_SO_REUSEADDR, one);
    if (stat != APR_SUCCESS && stat != APR_ENOTIMPL) {
        ap_log_perror(APLOG_MARK, APLOG_CRIT, stat, p,
                      "make_sock: for address %pI, apr_socket_opt_set: (SO_REUSEADDR)",
                      server->bind_addr);
        apr_socket_close(s);
        return stat;
    }
#endif
```

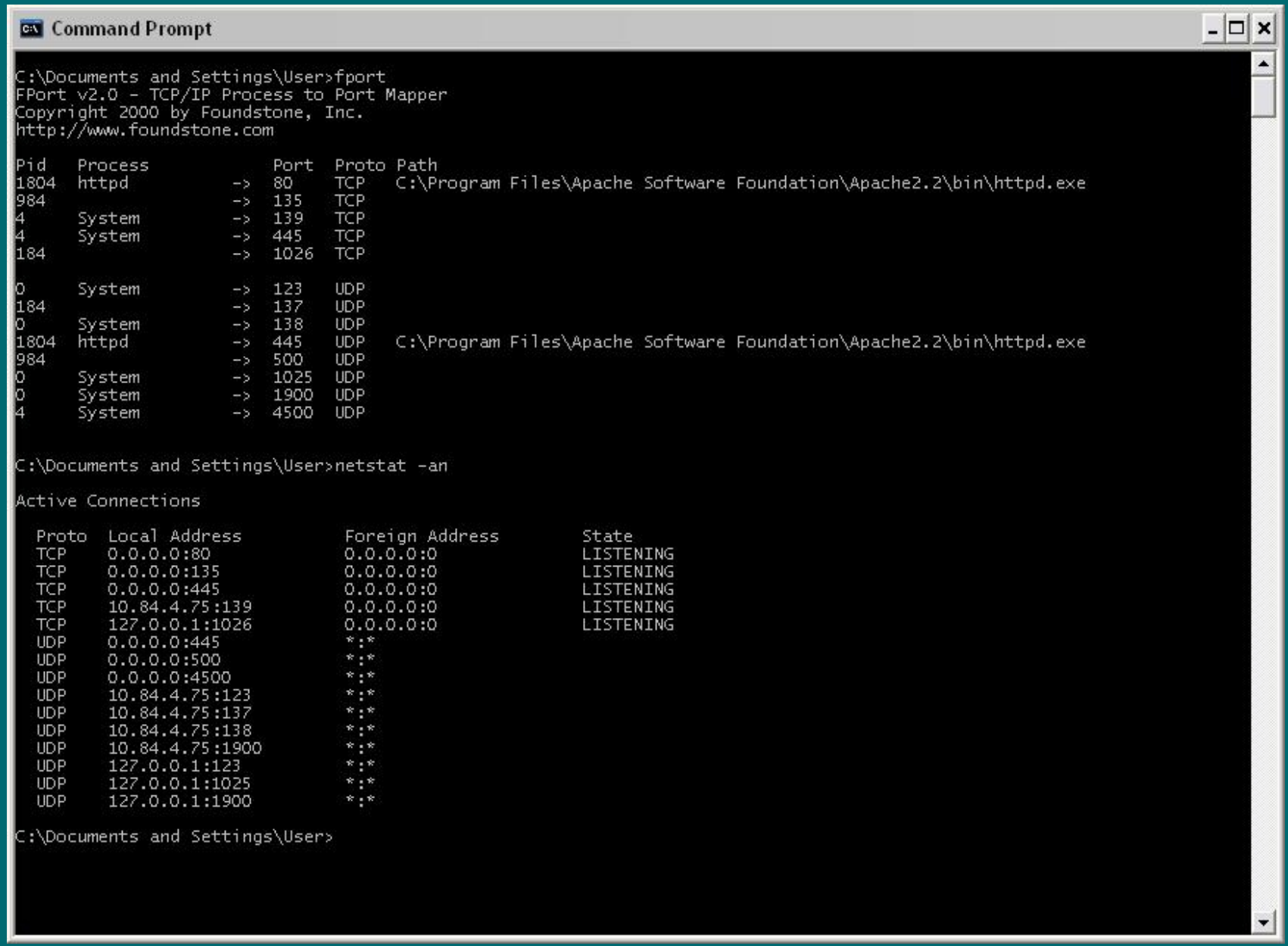
PHP Socket Library

- Low-level interface
- Powerful implementation, based on BSD sockets
- Provides server and client functionality
- Provides higher level functions (but we are not interested on those in this talk)

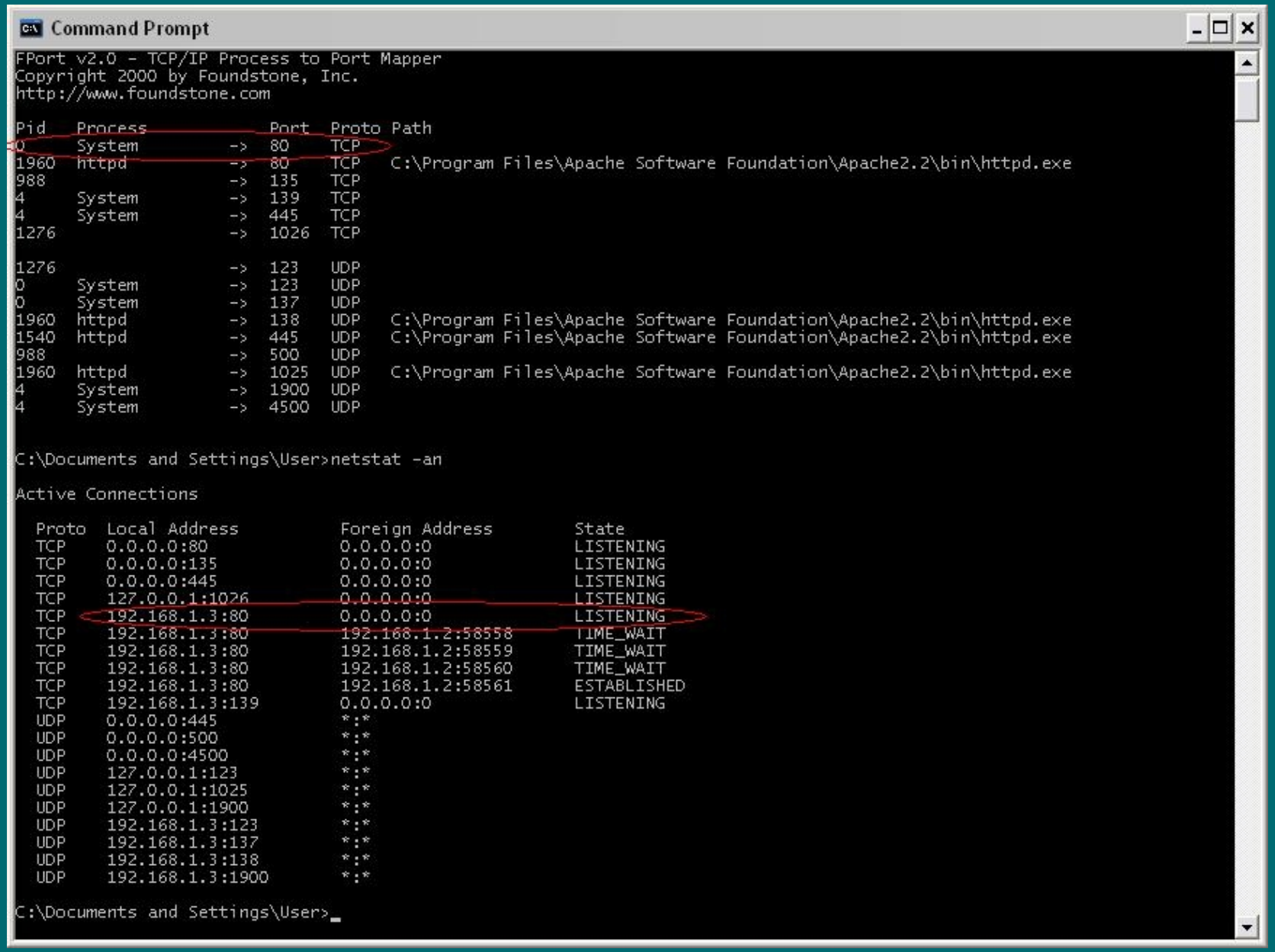
PHP Socket Reuse

- $sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP)
- socket_set_option($sock, SOL_SOCKET, SO_REUSEADDR, 1)
- socket_bind($sock, $address, $port)

PHP Socket Reuse (before)



PHP Socket Reuse (after)

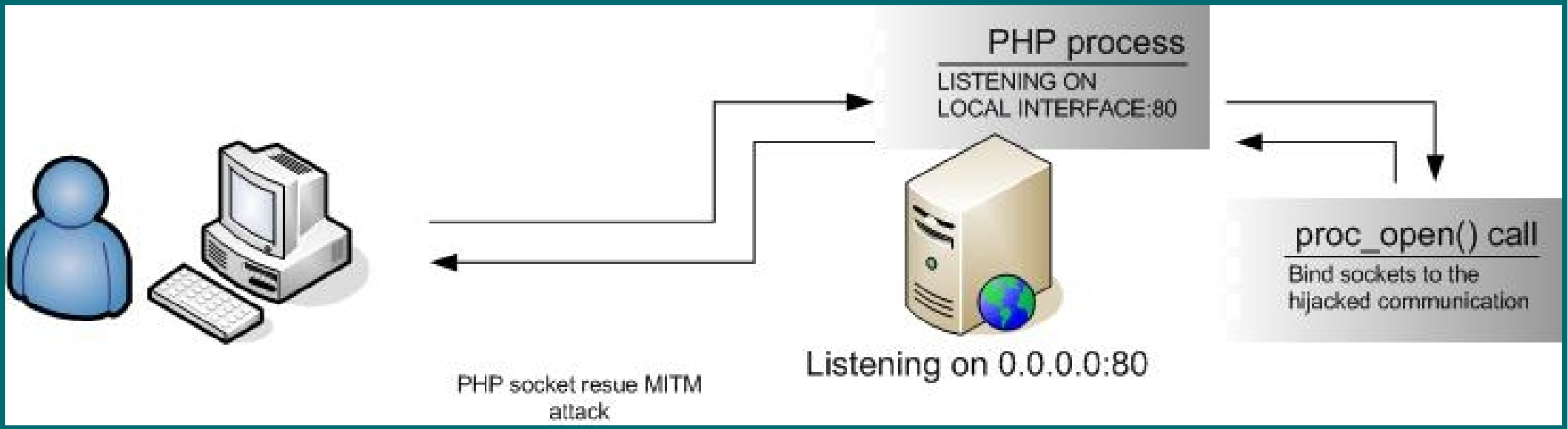


Vectors Of Attack

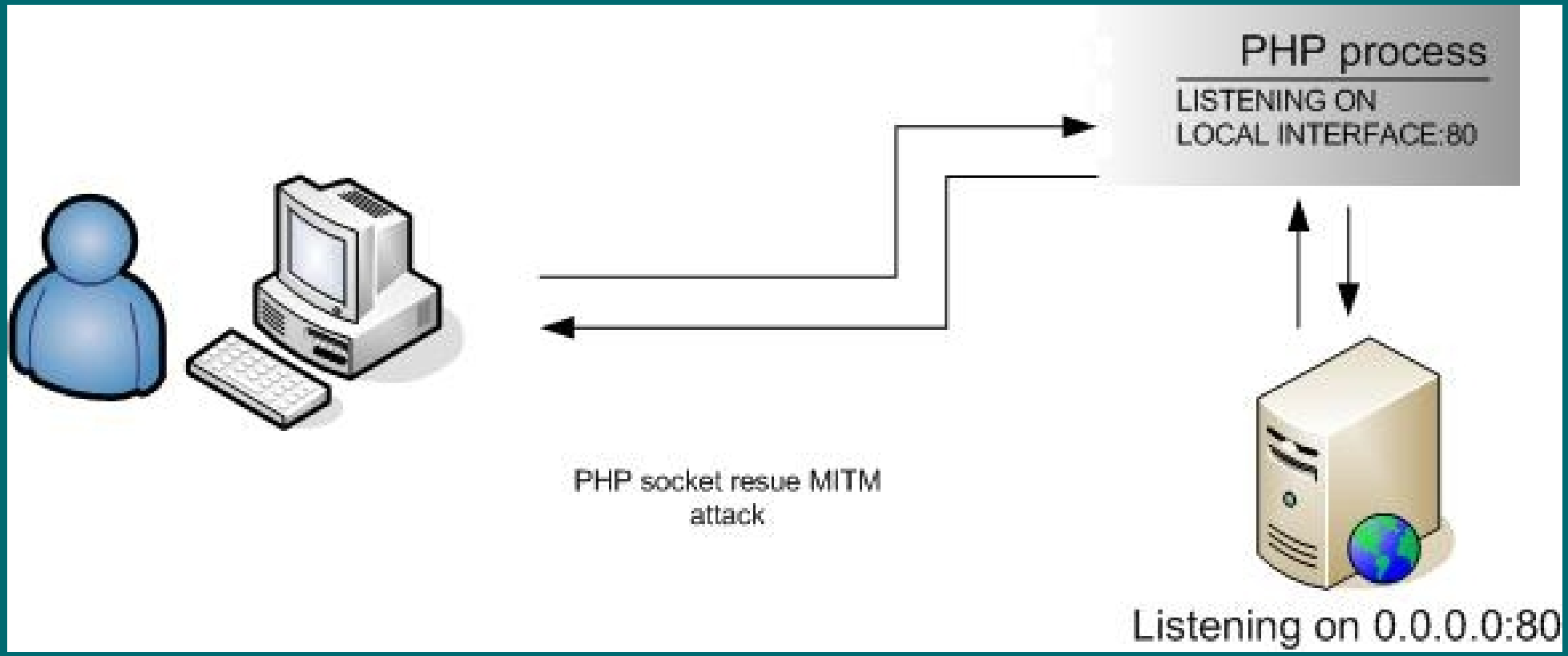
- Total control of clients
- We can send arbitrary response
  - Denial of service
  - Defacement
  - Bind to shell
  - Man-in-the-middle

  ...any other ideas?

PHP shell attack



PHP Man-in-the-middle attack



DEMO

Conclusions

- PHP provides a powerful socket library
- Sockets can be misused to perform neat attacks
- However, this attack is not that realistic as administration rights and execution of PHP code are needed

Thanks

Rodrigo Marcos
rodrigo.marcos@secforce.co.uk
www.secforce.co.uk