Poking Servers with {

whoami | head

- WebAppSec Consultant, Penetration Tester
- null Bangalore Chapter Lead
- Work at a Big4 and have conducted several Penetration Tests all over the world.
- Author of “A Beginners Approach to Windows”
- Chick Magnet [citation needed]

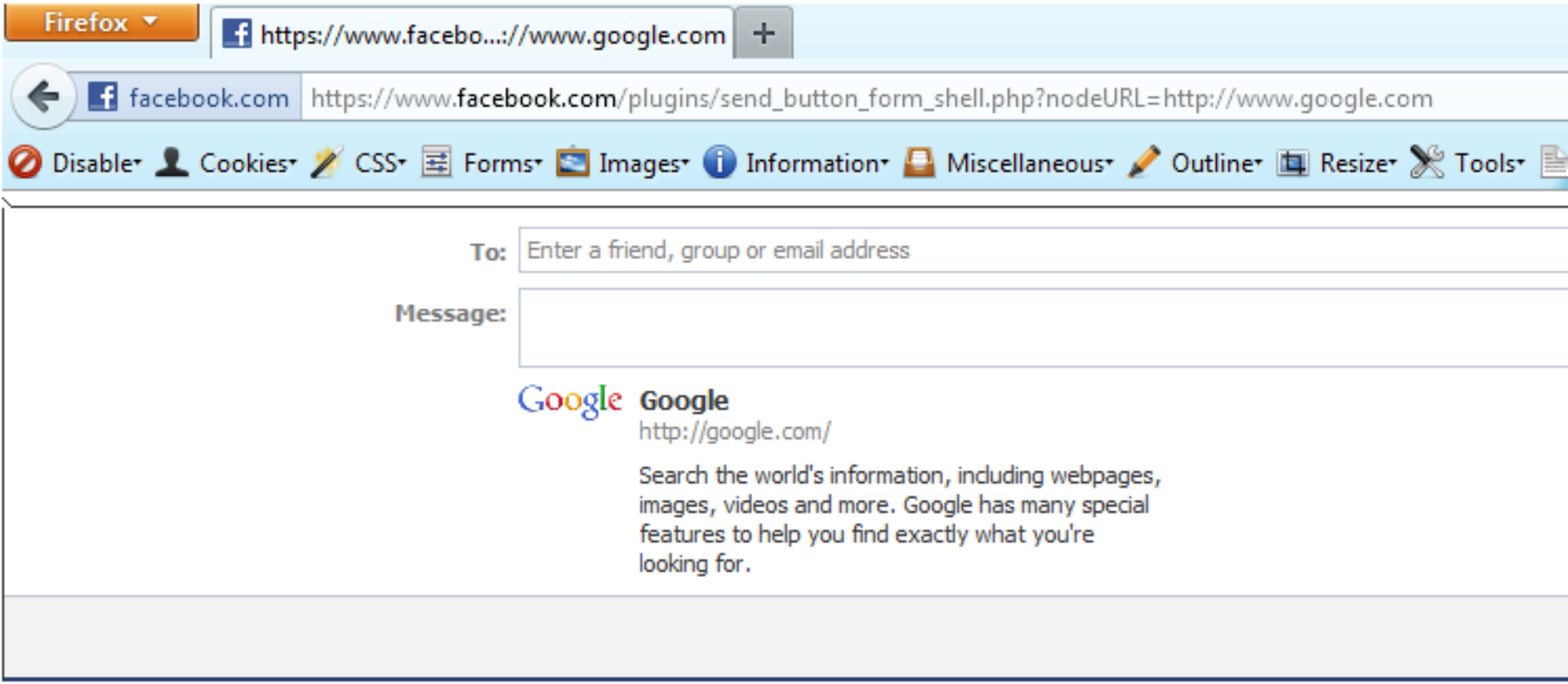
history | less

Started hunting for bugs on several bug bounty programs for

history | less

dpkg -i investigate.deb

Found a facebook.com URL which fetched the <title> from a URL I could control



uptime | cut –d " " –f2

Realized I could port scan Internet facing servers using verbose distinct errors from facebook

cat /etc/issue

Web Applications use underlying server side code to open socket connections to remote servers to download content

Error handling is seldom performed for failed socket connections at the web application level

Inadequate data handling for non HTTP data can cause applications to behave unexpectedly

mail -s 'Bug!' sec@fb.com < /dev/null

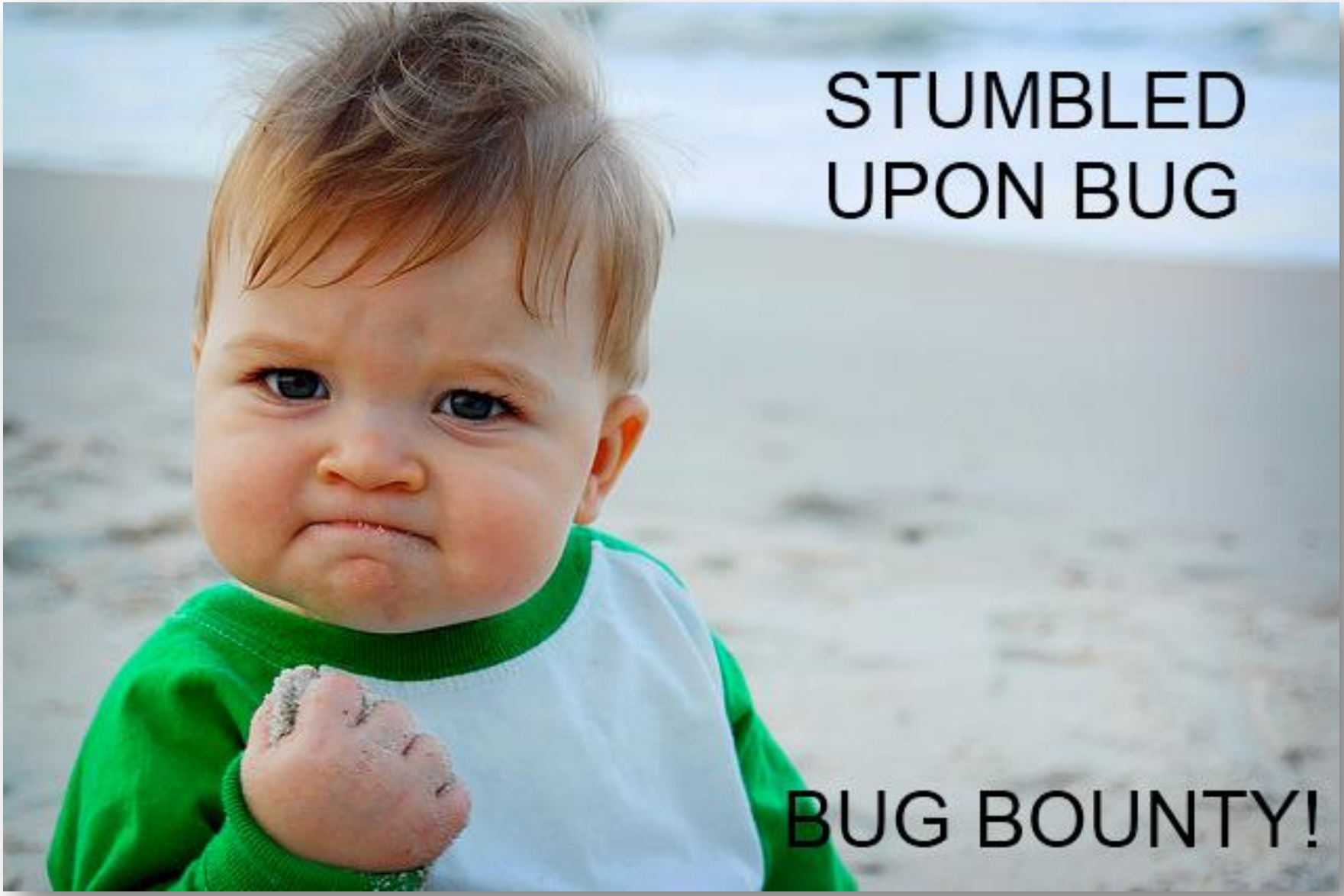
Reported the issue to Facebook who responded saying that they did not see how this was a problem

mail -s 'Bug!' sec@fb.com < /dev/null

Sent facebook a Proof of Concept python port scanner

Scanned some random servers on the Internet using the script

Facebook replied and acknowledged that this was a problem


STUMBLED
UPON BUG
BUG BOUNTY!

facebook

Search

- White Hats
- Report Vulnerability
- Bounty
- Manage Test Accounts

Information for Security Researchers

If you're a security researcher, please review our responsible disclosure policy before re
the Facebook Security Page for assistance.

If you believe you've found a security vulnerability on Facebook, we encourage you to l
our best to quickly fix the problem.

Responsible Disclosure Policy

If you give us a reasonable time to respond to your report before making any informatio
destruction of data and interruption or degradation of our service during your research,
investigate you.

Thanks!

On behalf of our millions of users, we would like to thank the following people for making

- Riyaz Walikar
- [redacted]

export vulnerability='XSPA'

XSPA – Cross Site Port Attacks

An application that allows users to download an xml file from a user controlled third party URL

XML File URL	Server Status & Body Response
http://remote_server.com/xmlfile.xml	200 OK – XML File retreived
http://remote_server.com:22/xmlfile.xml	200 OK – “Invalid XML data”
http://remote_server.com:3306/xmlfile.xml	200 OK – “Invalid XML data”
http://remote_server.com:8081	200 OK – “Connection refused!”

export vulnerability='XSPA'

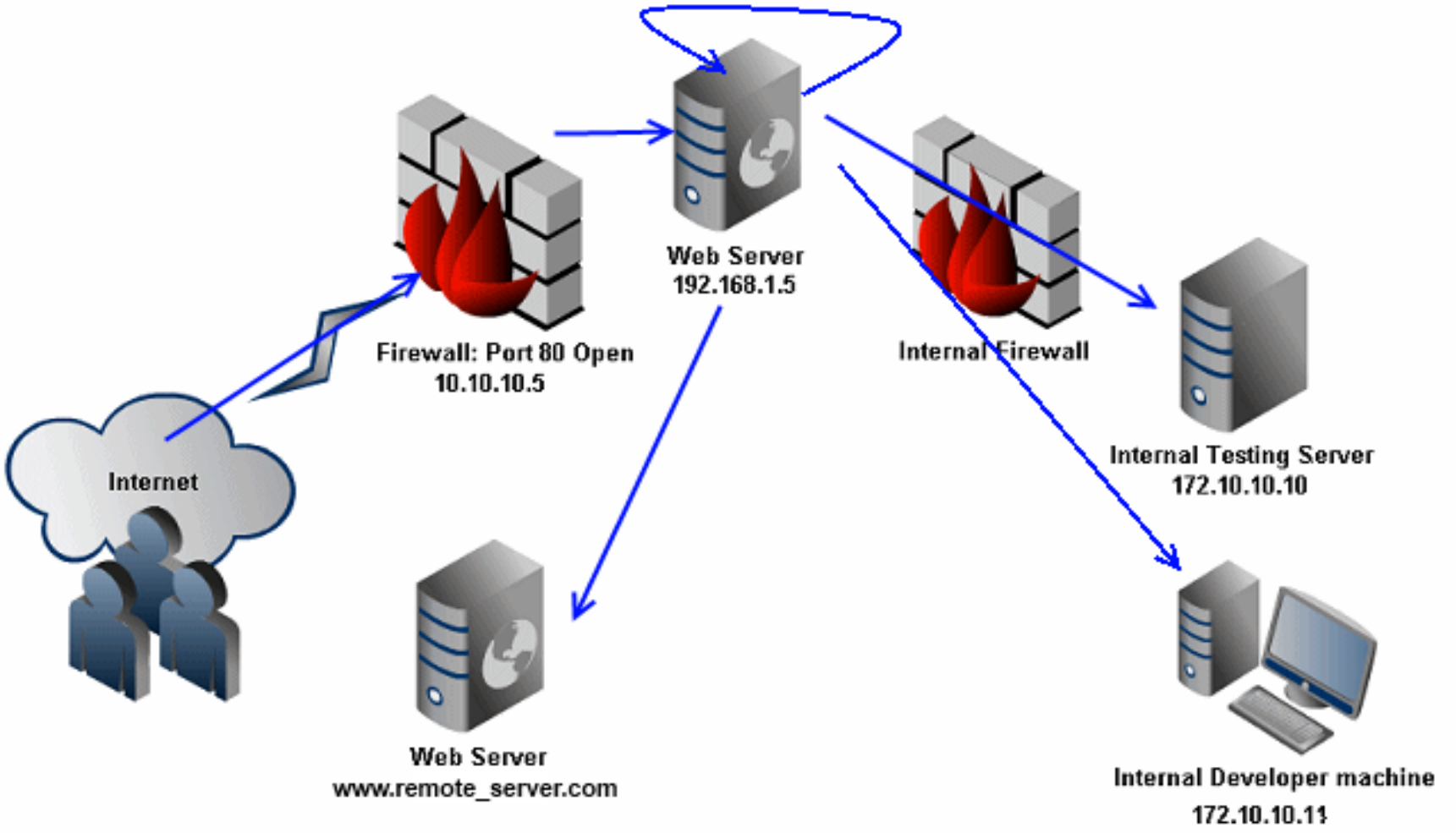
XSPA – Cross Site Port Attacks

Application displays verbose errors for failed socket connections

Application does not verify received data from the remote server, if the connection was successful

Application does not blacklist internal IP addresses/URLs

find . -print | xargs grep 'logic'



cat vulnfile.php | more

```php
<?php
    if (isset($_POST['url']))
    {
    $content = file_get_contents($_POST['url']);
    $filename = './images/'.rand().'img1.jpg';
    file_put_contents($filename, $content);
    echo $_POST['url']."</br>";
    $img = "<img src=\"".$filename."\"/>";
    }
    echo $img;
?>
```

cat vulnfile2.php | more

```
<?php
    function GetFile($host,$port,$link)
    {
    $fp = fsockopen($host, intval($port), $errno, $errstr,
    30);
    if (!$fp) {
    echo "$errstr (error number $errno)\n";
    } else {
    $out = "GET $link HTTP/1.1\r\n";
    $out .= "Host: $host\r\n";
    $out .= "Connection: Close\r\n\r\n";
    $out .= "Accept-Language: en-us,en;q=0.5\r\n";
    $out .= "\r\n";
    fwrite($fp, $out);
    $contents='';
    while (!feof($fp)) {
    $contents.= fgets($fp, 1024);
    }
    fclose($fp);
    return $contents;
    }
    }
?>
```

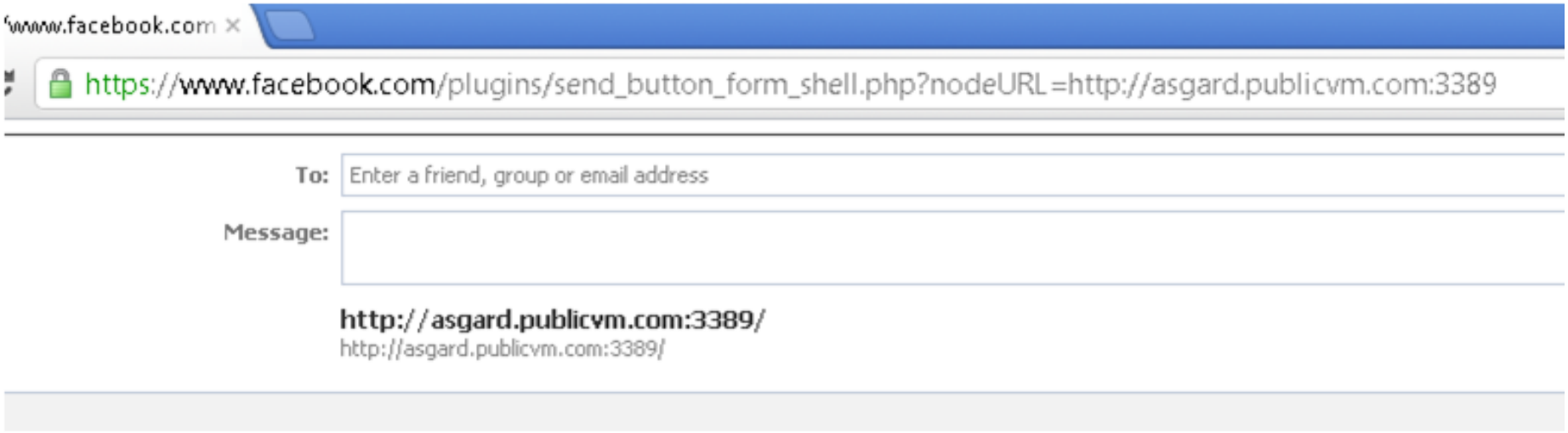
sudo demo &

cat popular_servers | ./poke

Found XSPA in

cat facebook

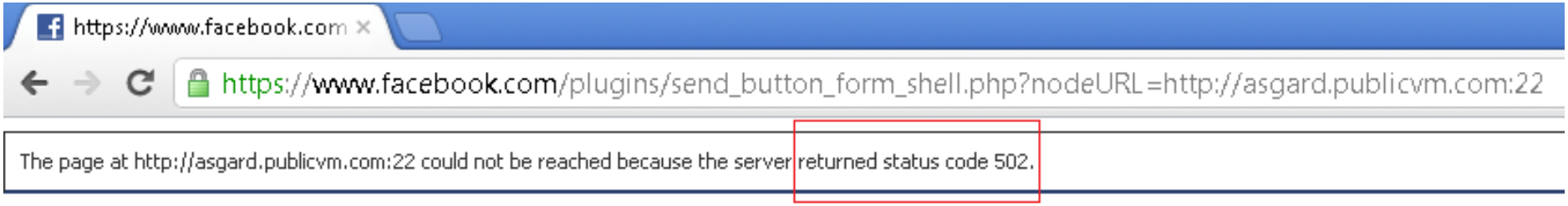
The first finding



Application specific response for open port *above* 1024

cat facebook

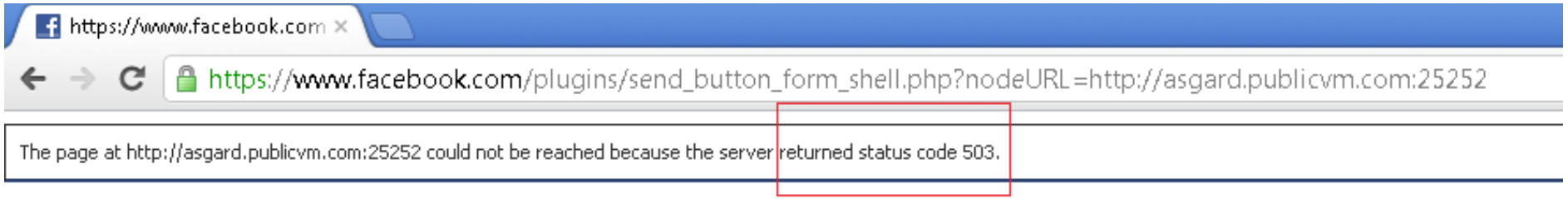
The first finding



Application specific response for open port *below* 1024

cat facebook

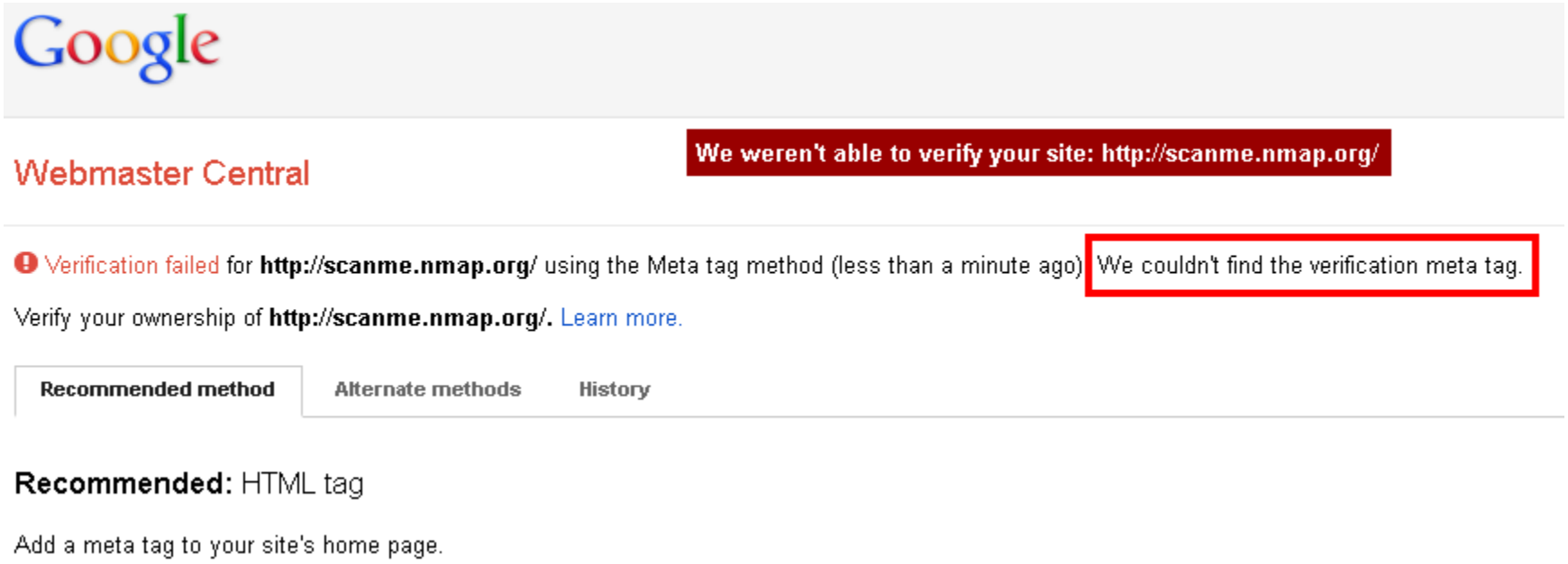
The first finding



Application specific response for closed port

cat Google

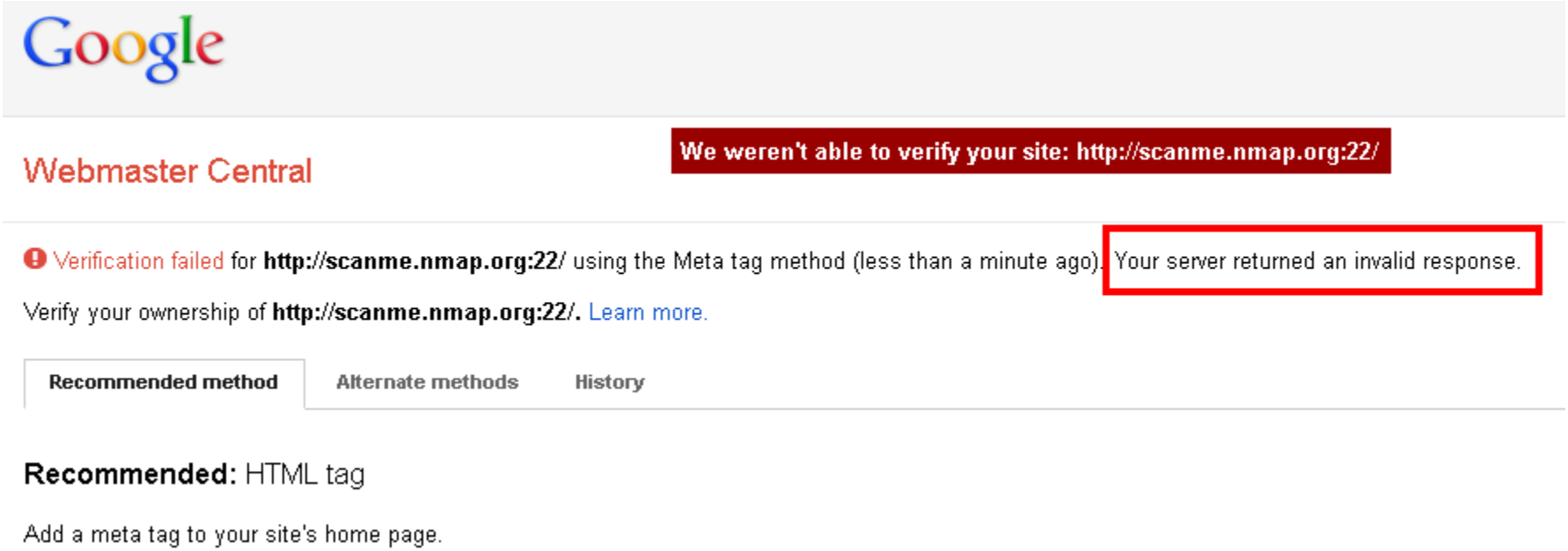
Google Webmasters - XSPA



Application specific response for open HTTP Port

cat Google

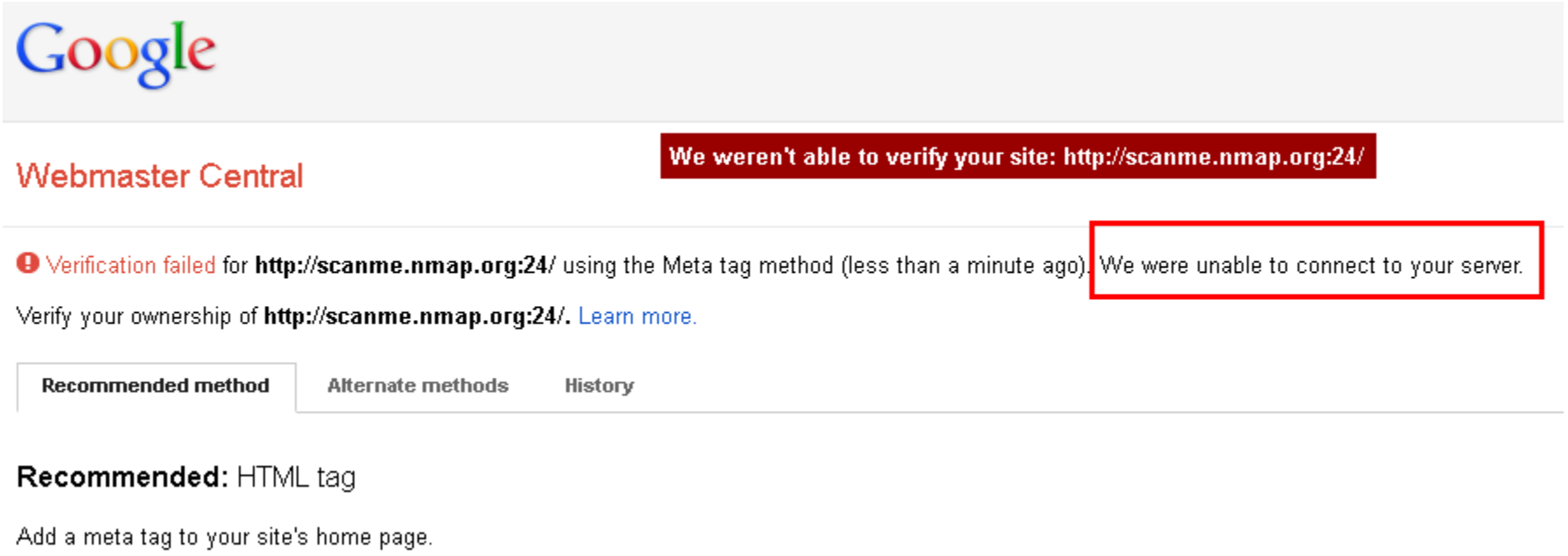
Google Webmasters - XSPA



Application specific response for open non-HTTP Port

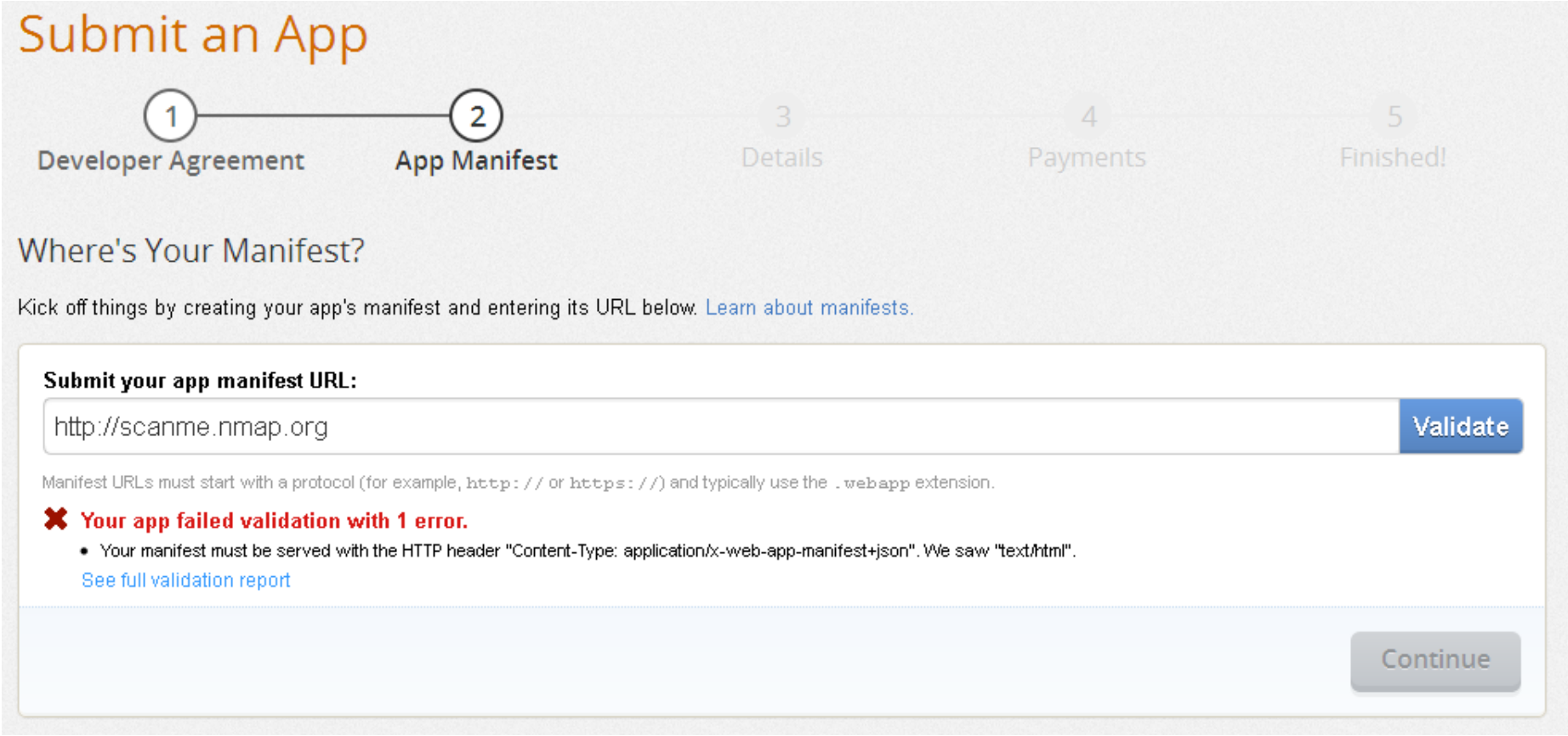
cat Google

Google Webmasters - XSPA



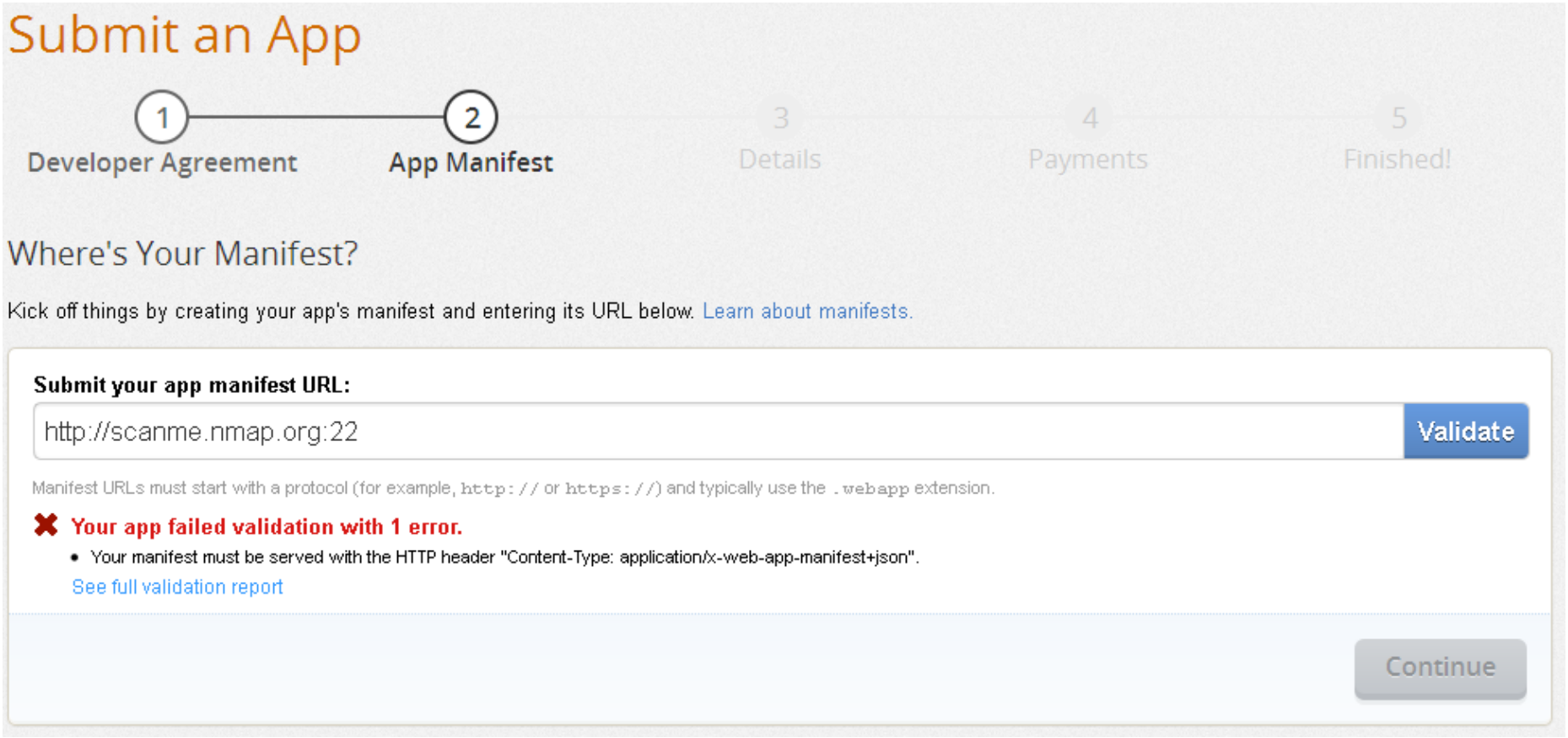
Application specific response for closed port

cat mozilla_marketplace



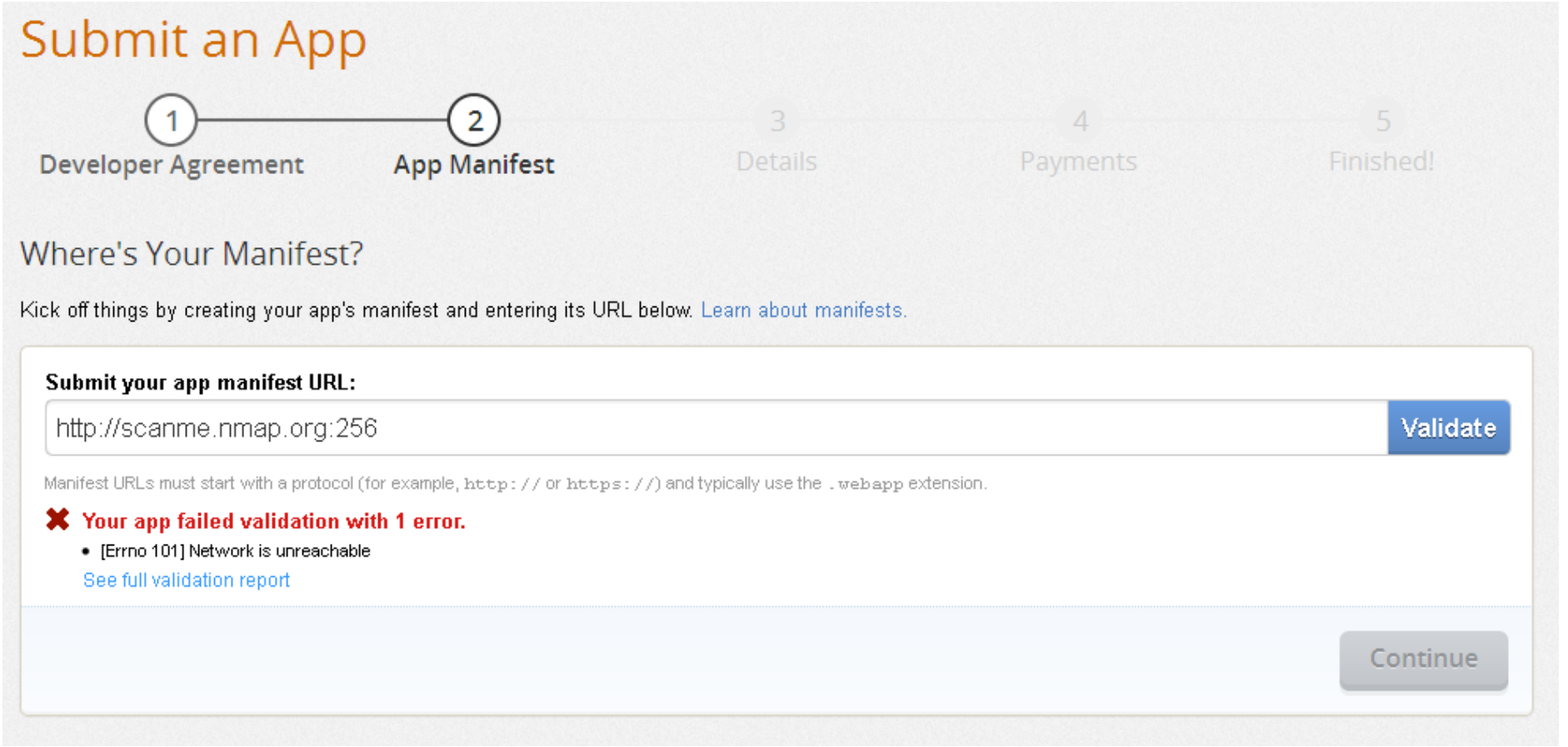
Application specific response for open HTTP port

cat mozilla_marketplace



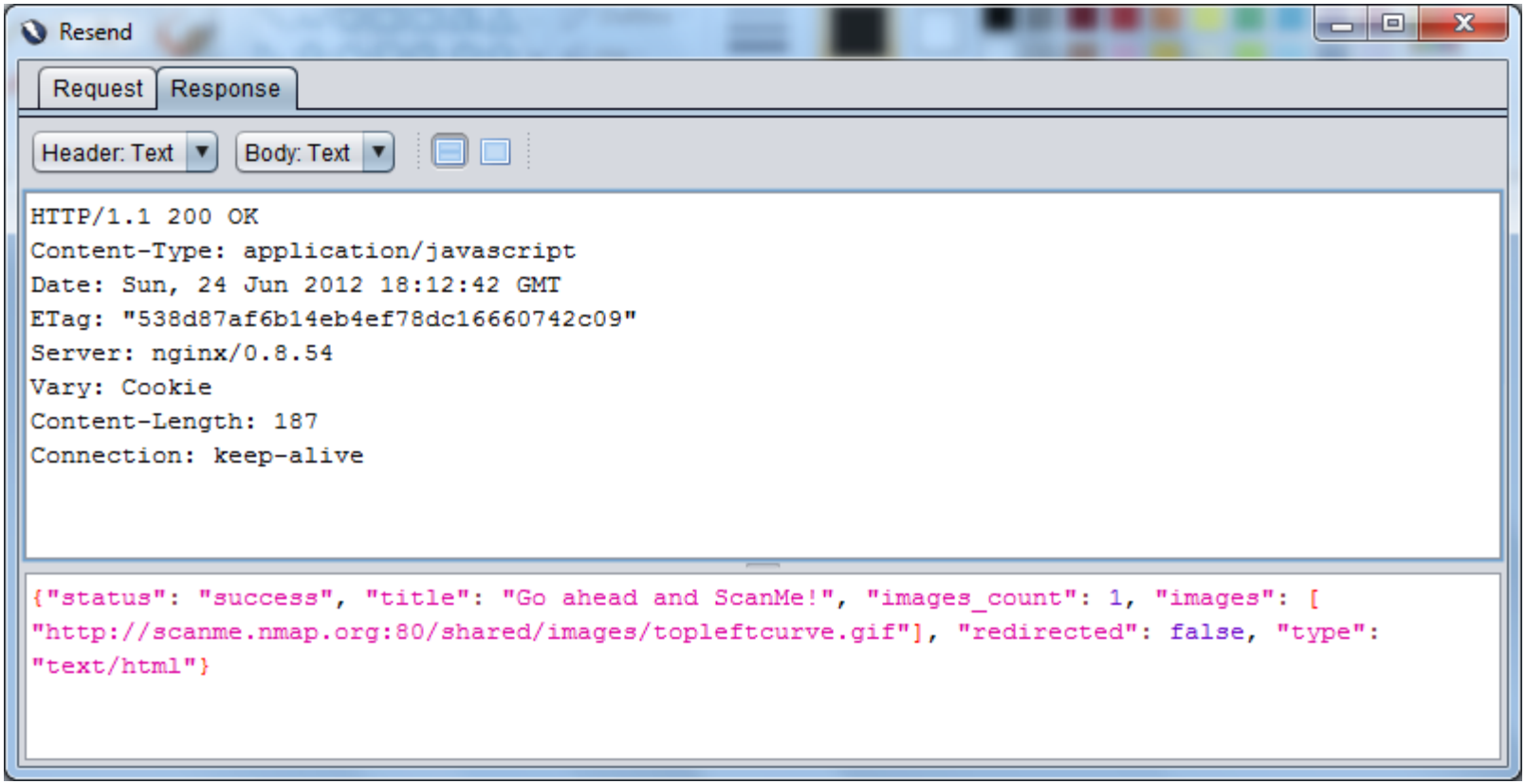
Application specific response for open non HTTP port

cat mozilla_marketplace



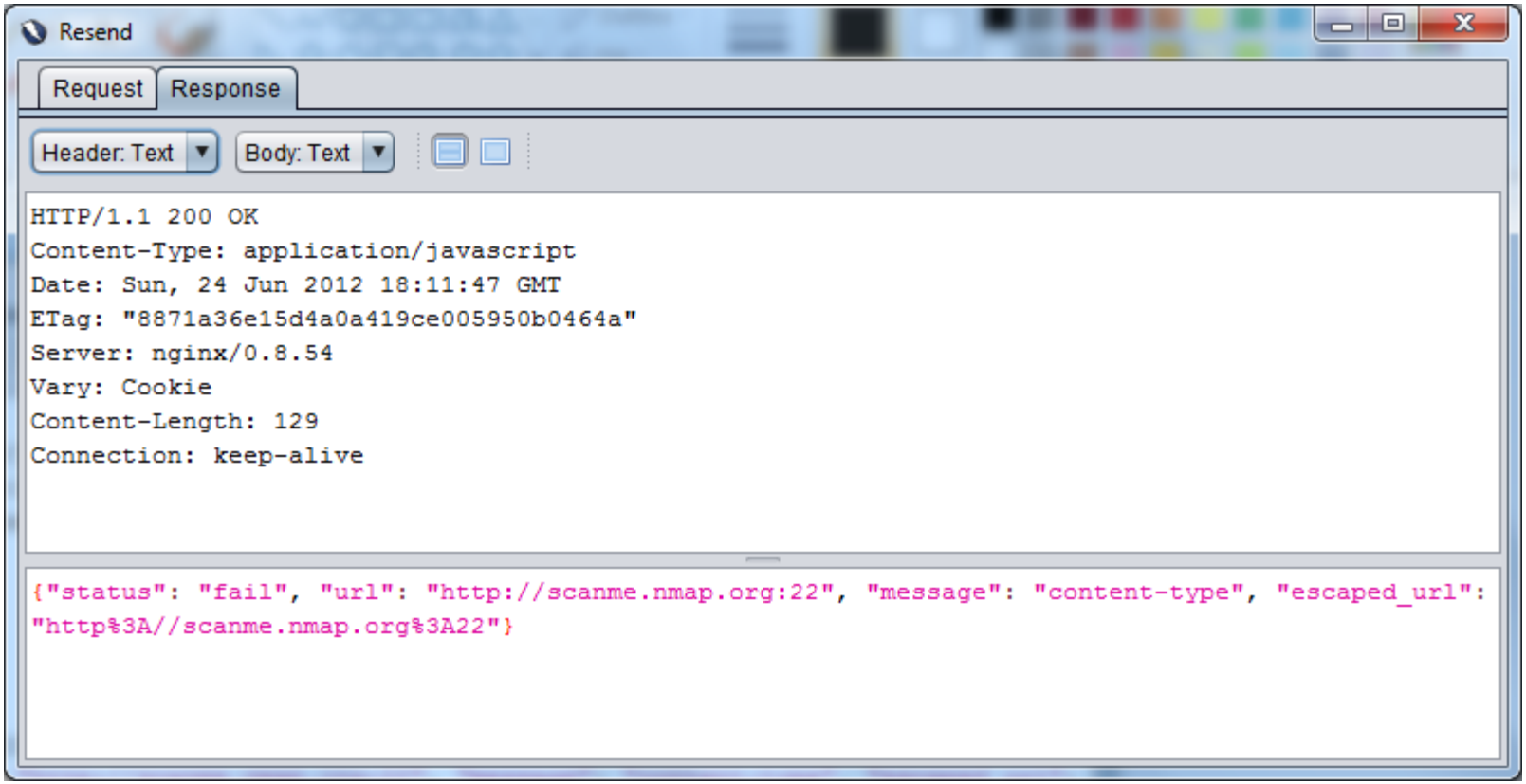
Application specific response for closed port

cat pinterest



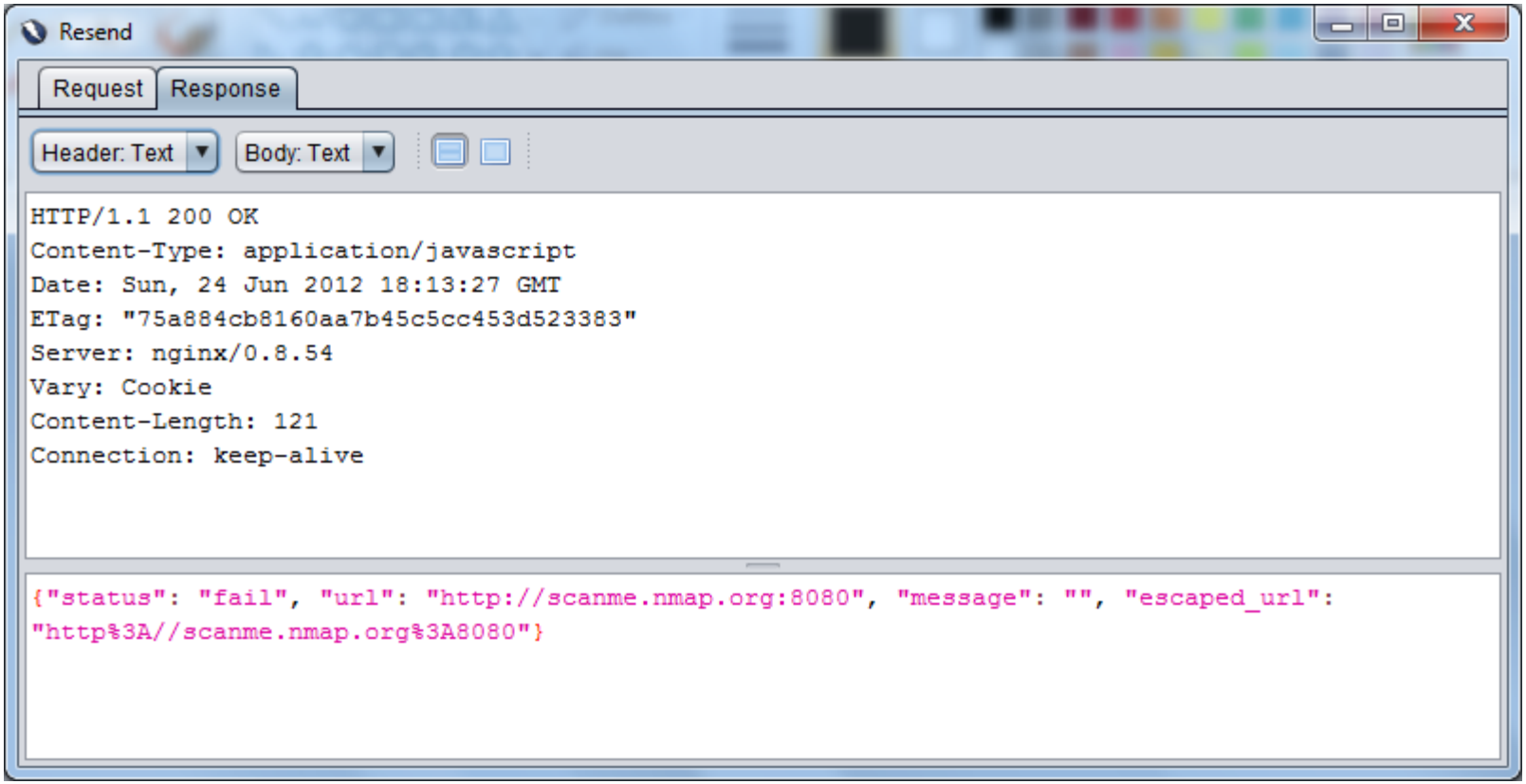
Application specific response for open HTTP ports

cat pinterest



Application specific response for open non-HTTP ports

cat pinterest



Application specific response for closed ports

ls adobe*.flv | xargs vlc

patch -p1 < /var/fixes

Basic mitigation is to force applications to make connections to remote servers to fetch data over ports 80 and 443 only

If data from other ports is required to be fetched, make sure that the data can be parsed in the format that the application expects

Do not allow connections to private IP addresses

Handle all errors/exceptions and timeouts and display generic messages regardless of the invoking condition

cat /xspa/other_attacks

Attackers can access internal applications and perform URL based attacks (SQLi, Parameter manipulation etc.)

Since the GET /<data> part is controlled by the attacker, it would be possible to attack services and use overflows to open reverse shells to attacker's computer

Limited only by your own imagination!

find / -type l

RFC 2616 - www.w3.org/Protocols/rfc2616/rfc2616.html

Rsnake's Client Side Port Scanning - http://www.sectheory.com/intranet-hacking.htm

Cross Domain XMLHTTPRequest Port Scanning - http://ha.ckers.org/weird/xhr-ping-sweep.html

Riyaz Ahemed Walikar
@riyazwalikar
http://www.riyazwalikar.com