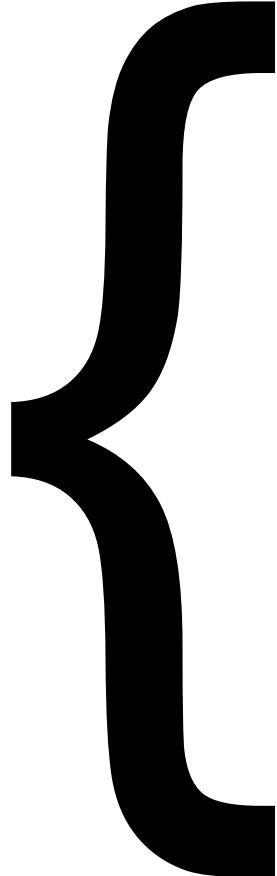
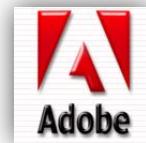


Poking Servers with



mozilla
FOUNDATION



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

The screenshot shows a Firefox browser window. The title bar is modified to read "https://www.facebook.com://www.google.com". The main content area displays a Google search result for "Google" with the URL "http://google.com/". The page includes a "To:" field for messaging and a "Message:" field below it. The title "Google" and URL are displayed in a way that suggests they were fetched from a controlled URL, as indicated by the modified title bar.

Firefox

https://www.facebook.com://www.google.com

facebook.com https://www.facebook.com/plugins/send_button_form_shell.php?nodeURL=http://www.google.com

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resizer Tools

To: Enter a friend, group or email address

Message:

Google Google
http://google.com/

Search the world's information, including webpages, images, videos and more. Google has many special features to help you find exactly what you're looking for.

```
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



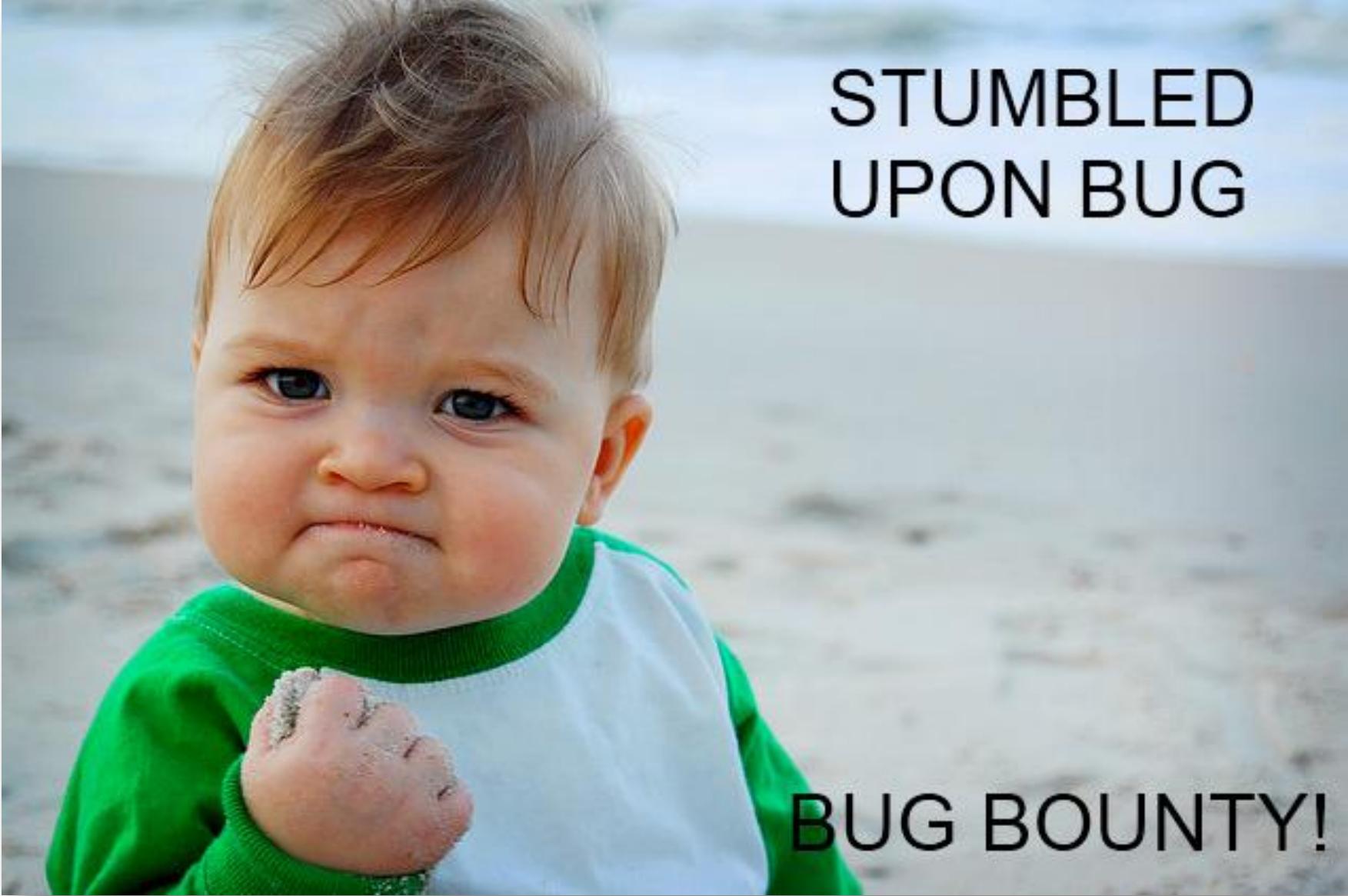
```
mail -s 'Bug!' sec@fb.com < /dev/null
```

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



A close-up photograph of a baby with light brown hair and blue eyes. The baby is wearing a green and white long-sleeved shirt. A small, dark, worm-like insect is visible on the baby's right hand. The background is a blurred beach scene with sand and water.

STUMBLED
UPON BUG

BUG BOUNTY!

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 reaching out. You can also contact us on the [Facebook Security Page](#) for assistance.

If you believe you've found a security vulnerability on Facebook, we encourage you to let us know so we can work together to quickly fix the problem.

Responsible Disclosure Policy

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

Thanks!

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

▪ Riyaz Walikar



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 retrieved
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'
```

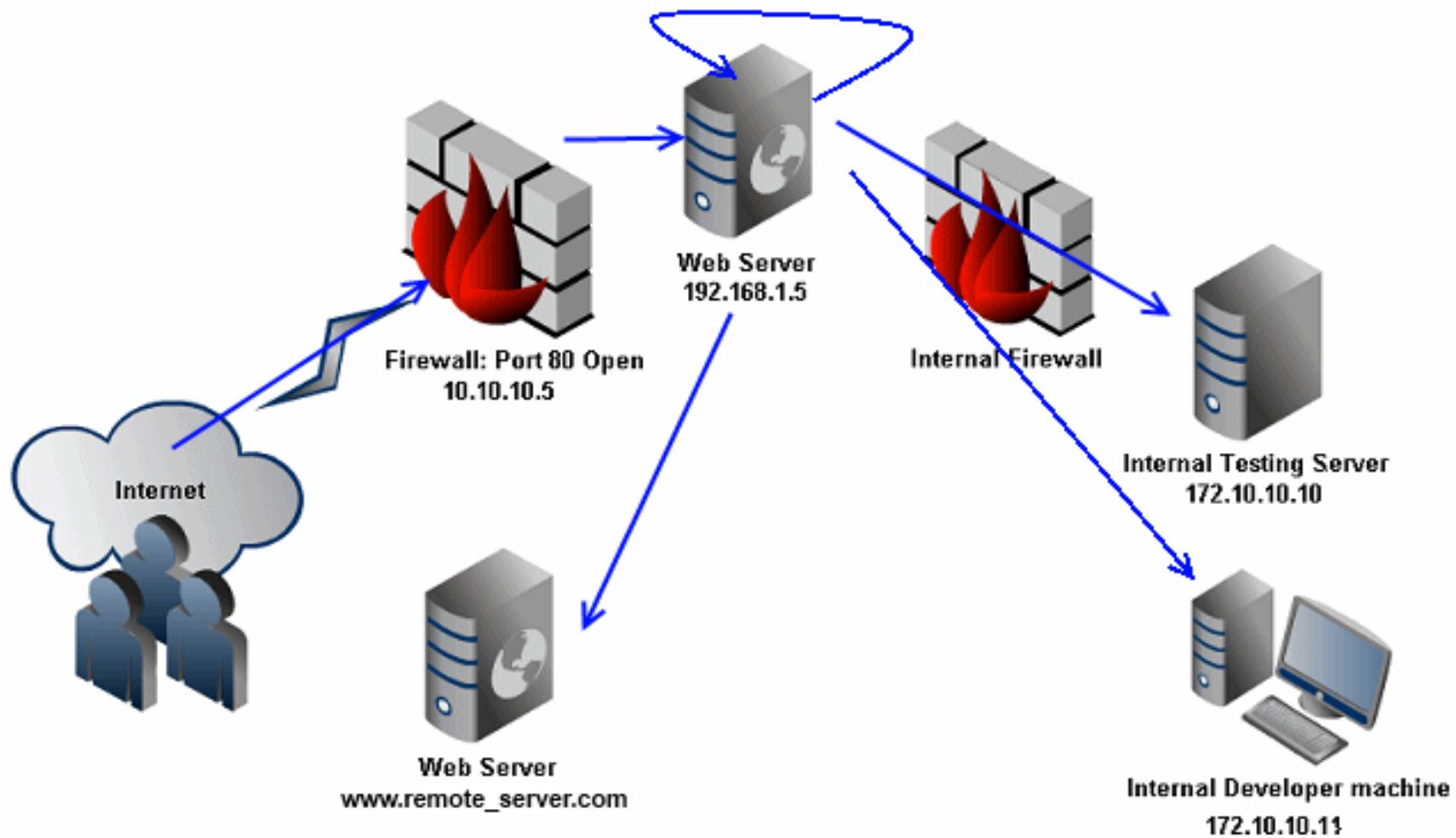
XSPA – Cross Site Port Attacks

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
    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;
        }
    }
?>
```

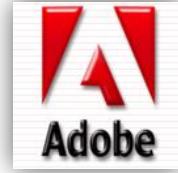


sudo demo &



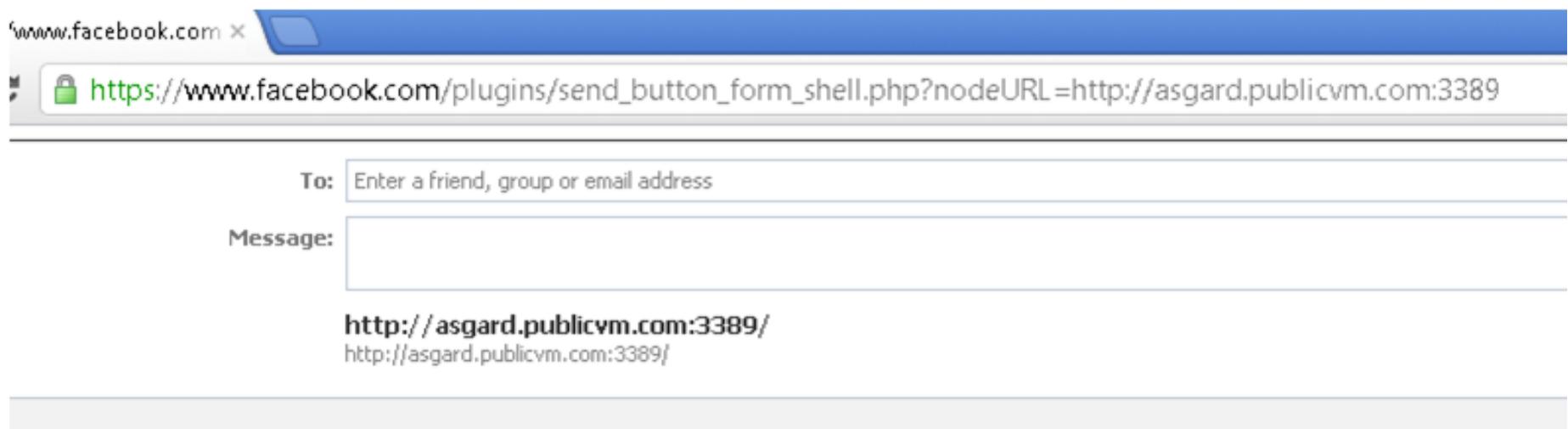
```
cat popular_servers | ./poke
```

Found XSPA in



cat facebook

The first finding

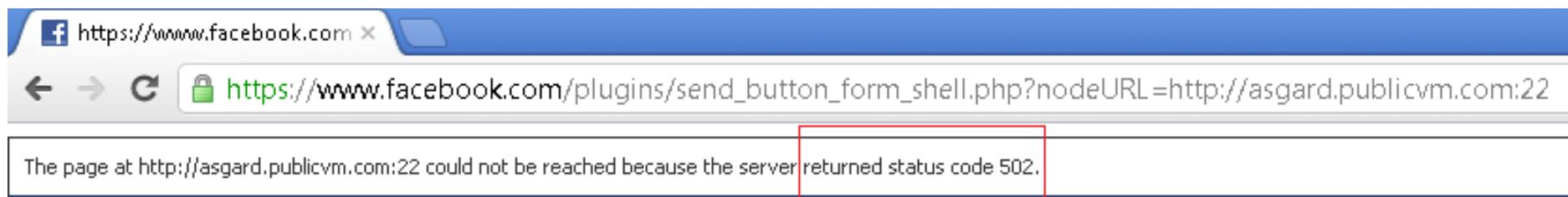


A screenshot of a web browser window. The address bar shows a URL starting with https://www.facebook.com/plugins/send_button_form_shell.php?nodeURL=http://asgard.publicvm.com:3389. The page content is a Facebook 'Send' button form. It has two input fields: 'To:' with the placeholder 'Enter a friend, group or email address' and 'Message:' with an empty text area. Below the form, the URL <http://asgard.publicvm.com:3389/> is displayed twice.

Application specific response for open port *above* 1024

cat facebook

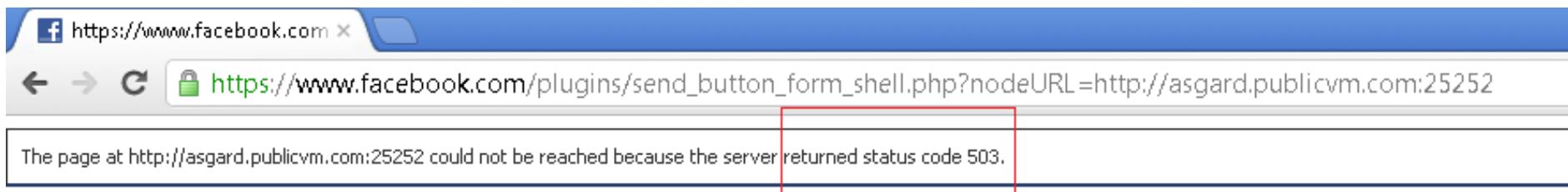
The first finding



Application specific response for open port *below* 1024

cat facebook

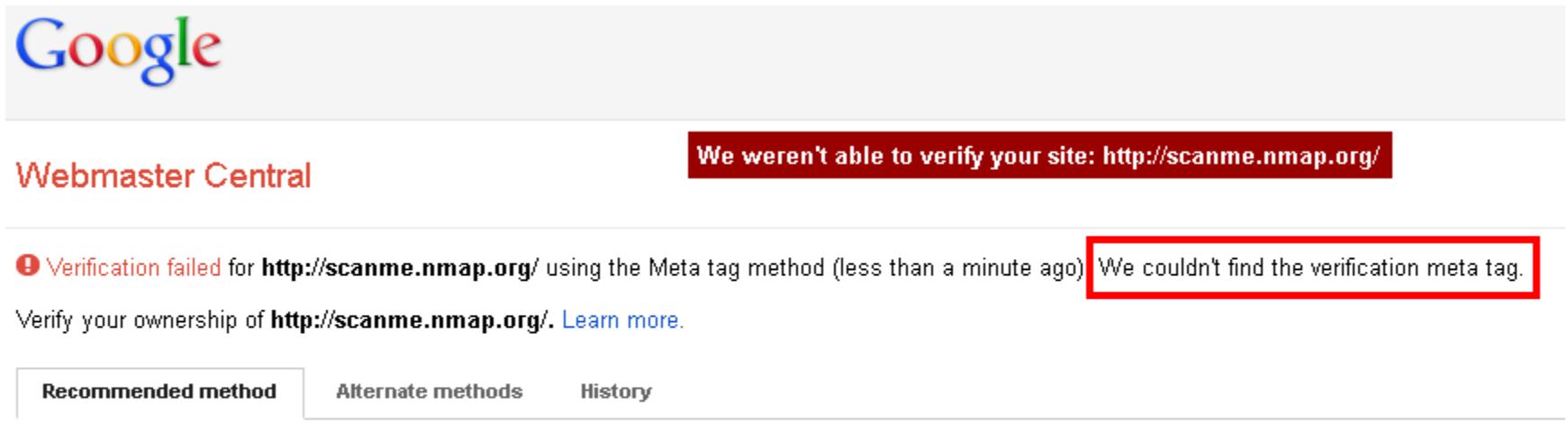
The first finding



Application specific response for closed port

cat Google

Google Webmasters - XSPA



The screenshot shows a Google search result for "Webmaster Central". The page title is "Webmaster Central". A red box highlights the error message: "We weren't able to verify your site: <http://scanme.nmap.org/>". Below the message, a note says: "Verification failed for <http://scanme.nmap.org/> using the Meta tag method (less than a minute ago). We couldn't find the verification meta tag." A link to "Learn more" is provided. At the bottom, there are tabs for "Recommended method" (which is selected), "Alternate methods", and "History".

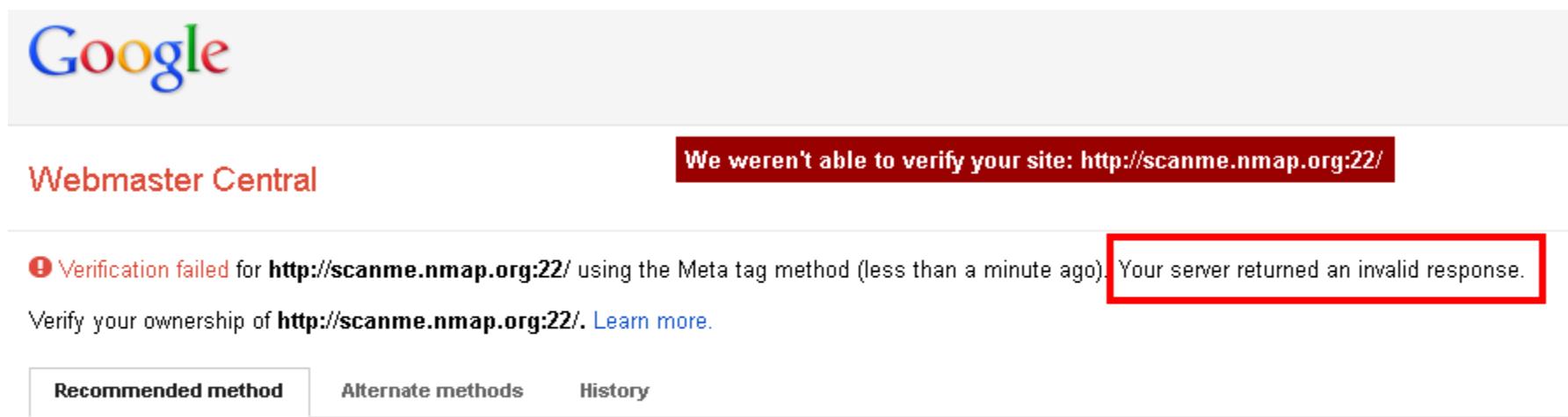
Recommended: HTML tag

Add a meta tag to your site's home page.

Application specific response for open HTTP Port

cat Google

Google Webmasters - XSPA



The screenshot shows a Google search result for "Google Webmaster Central". The page title is "Google Webmaster Central". A red box highlights an error message: "We weren't able to verify your site: http://scanme.nmap.org:22/". Below the message, a red box highlights a warning: "Verification failed for http://scanme.nmap.org:22/ using the Meta tag method (less than a minute ago). Your server returned an invalid response." A link "Learn more." is also visible. At the bottom, there are tabs for "Recommended method", "Alternate methods", and "History".

Recommended: HTML tag

Add a meta tag to your site's home page.

Application specific response for open non-HTTP Port

cat Google

Google Webmasters - XSPA

The screenshot shows a Google search result for "Webmaster Central". The page title is "Webmaster Central". A red box highlights the error message: "We weren't able to verify your site: http://scanme.nmap.org:24/". Below the message, a red box highlights a warning: "Verification failed for http://scanme.nmap.org:24/ using the Meta tag method (less than a minute ago). We were unable to connect to your server." A link "Learn more." is visible. At the bottom, there are tabs for "Recommended method" (which is selected), "Alternate methods", and "History".

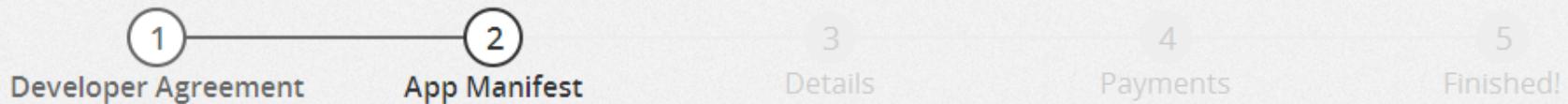
Recommended: HTML tag

Add a meta tag to your site's home page.

Application specific response for closed port

cat mozilla_marketplace

Submit an App



Where's Your Manifest?

Kick off things by creating your app's manifest and entering its URL below. [Learn about manifests.](#)

Submit your app manifest URL:

Validate

Manifest URLs must start with a protocol (for example, `http://` or `https://`) and typically use the `.webapp` extension.

✖ Your app failed validation with 1 error.

- Your manifest must be served with the HTTP header "Content-Type: application/x-web-app-manifest+json". We saw "text/html".

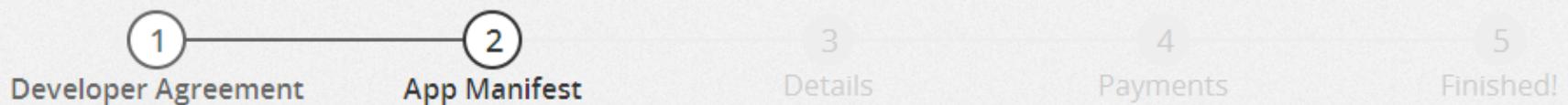
[See full validation report](#)

Continue

Application specific response for open HTTP port

cat mozilla_marketplace

Submit an App



Where's Your Manifest?

Kick off things by creating your app's manifest and entering its URL below. [Learn about manifests.](#)

Submit your app manifest URL:

Validate

Manifest URLs must start with a protocol (for example, `http://` or `https://`) and typically use the `.webapp` extension.

✖ Your app failed validation with 1 error.

- Your manifest must be served with the HTTP header "Content-Type: application/x-web-app-manifest+json".

[See full validation report](#)

[Continue](#)

Application specific response for open non HTTP port

cat mozilla_marketplace

Submit an App

1

Developer Agreement

2

App Manifest

3

Details

4

Payments

5

Finished!

Where's Your Manifest?

Kick off things by creating your app's manifest and entering its URL below. [Learn about manifests.](#)

Submit your app manifest URL:

http://scanme.nmap.org:256

Validate

Manifest URLs must start with a protocol (for example, `http://` or `https://`) and typically use the `.webapp` extension.

✖ Your app failed validation with 1 error.

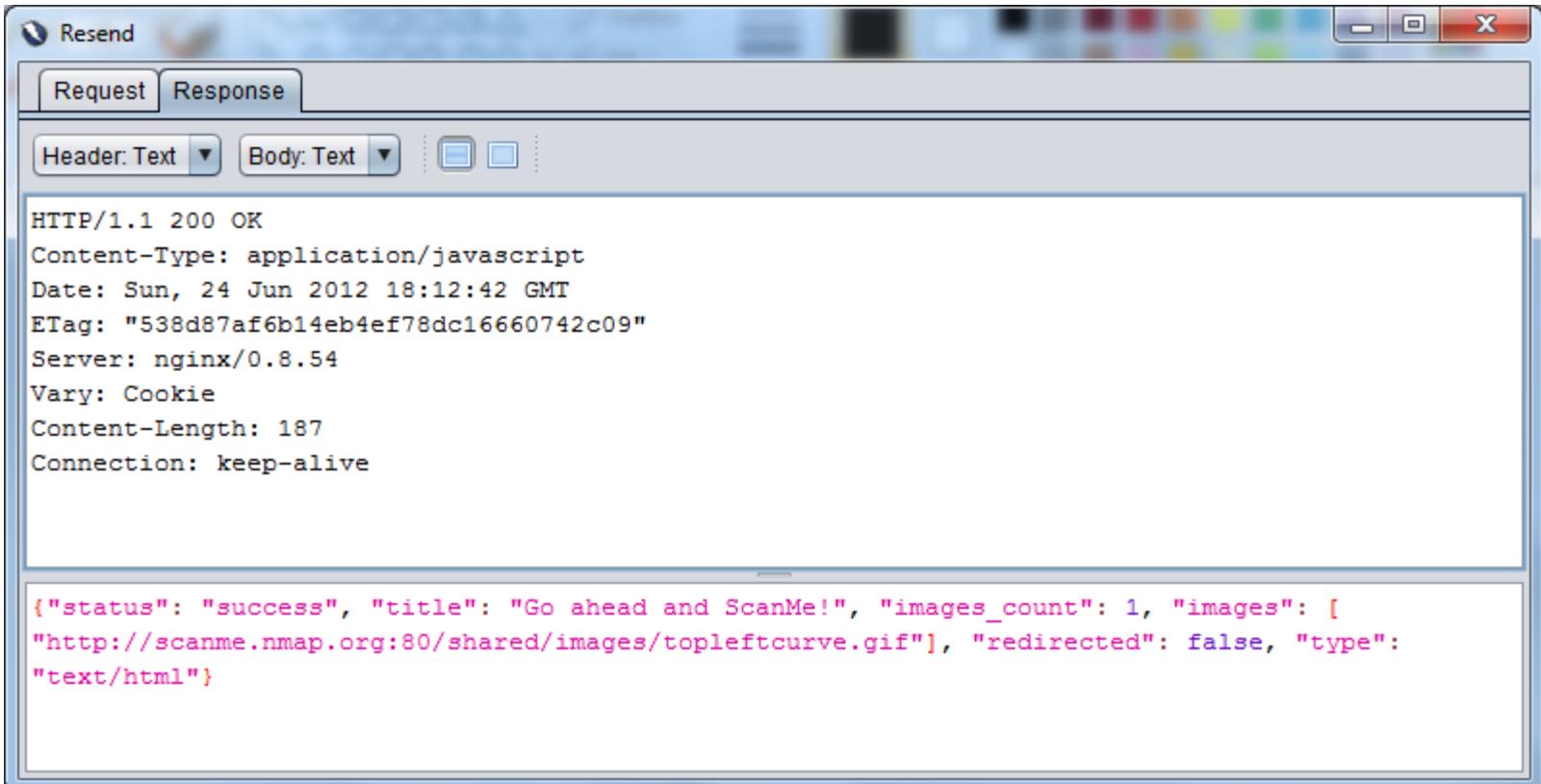
- [Errno 101] Network is unreachable

[See full validation report](#)

Continue

Application specific response for closed port

cat pinterest



The screenshot shows a browser window with a blue header bar. The main content area has a light gray header with tabs for "Request" and "Response". Below this is a toolbar with dropdowns for "Header: Text" and "Body: Text", and several small icons. The "Response" tab is active. The "Header: Text" dropdown shows the following HTTP response headers:

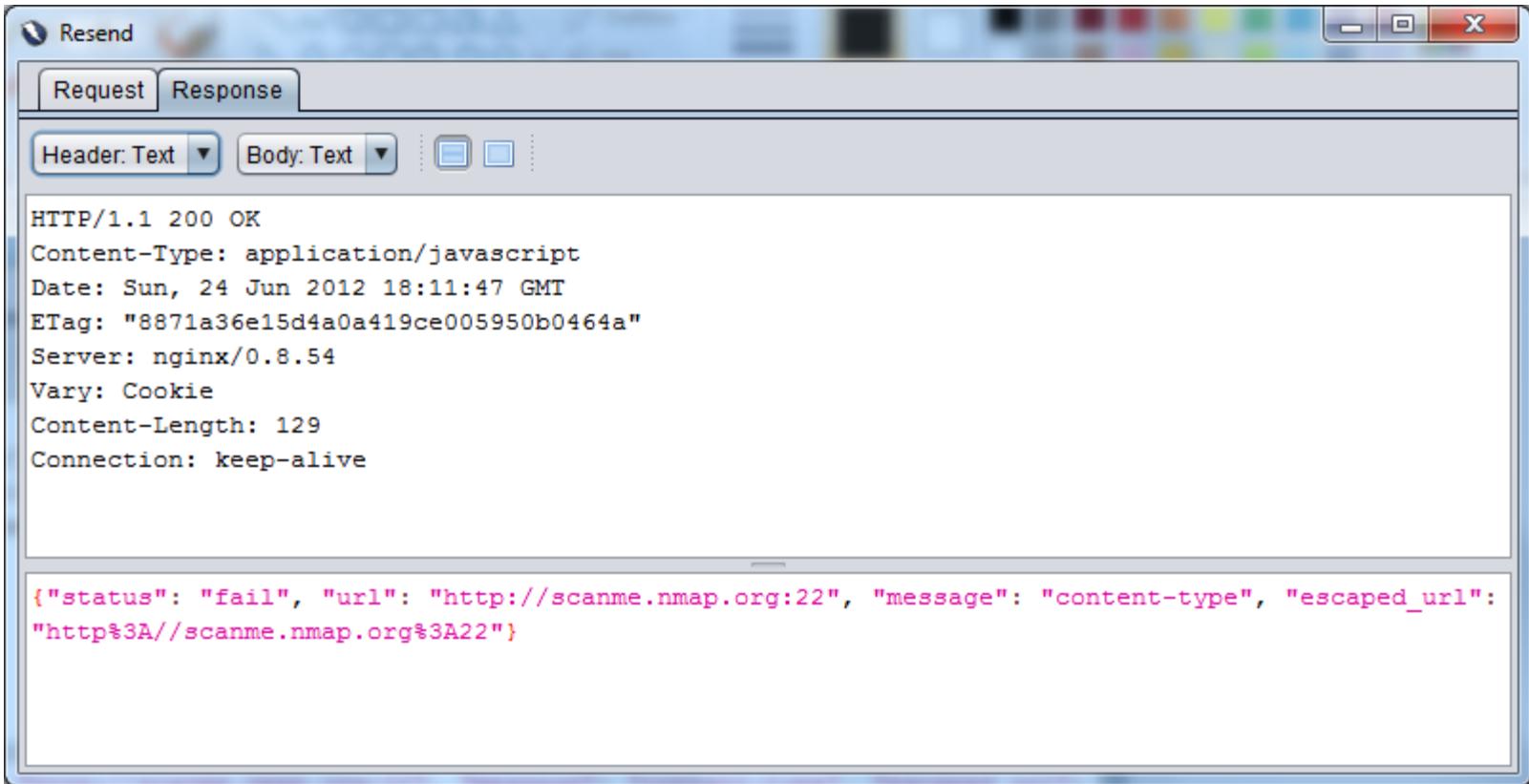
```
HTTP/1.1 200 OK
Content-Type: application/javascript
Date: Sun, 24 Jun 2012 18:12:42 GMT
ETag: "538d87af6b14eb4ef78dc16660742c09"
Server: nginx/0.8.54
Vary: Cookie
Content-Length: 187
Connection: keep-alive
```

The "Body: Text" dropdown shows a JSON object:

```
{"status": "success", "title": "Go ahead and ScanMe!", "images_count": 1, "images": [ "http://scanme.nmap.org:80/shared/images/topleftcurve.gif"], "redirected": false, "type": "text/html"}
```

Application specific response for open HTTP ports

cat pinterest



The screenshot shows a browser window with the title 'Resend' at the top. Below the title, there are two tabs: 'Request' and 'Response'. The 'Response' tab is selected. At the top of the response area, there are dropdown menus for 'Header: Text' and 'Body: Text', and a toolbar with three icons. The main content area displays an HTTP response. The header information is as follows:

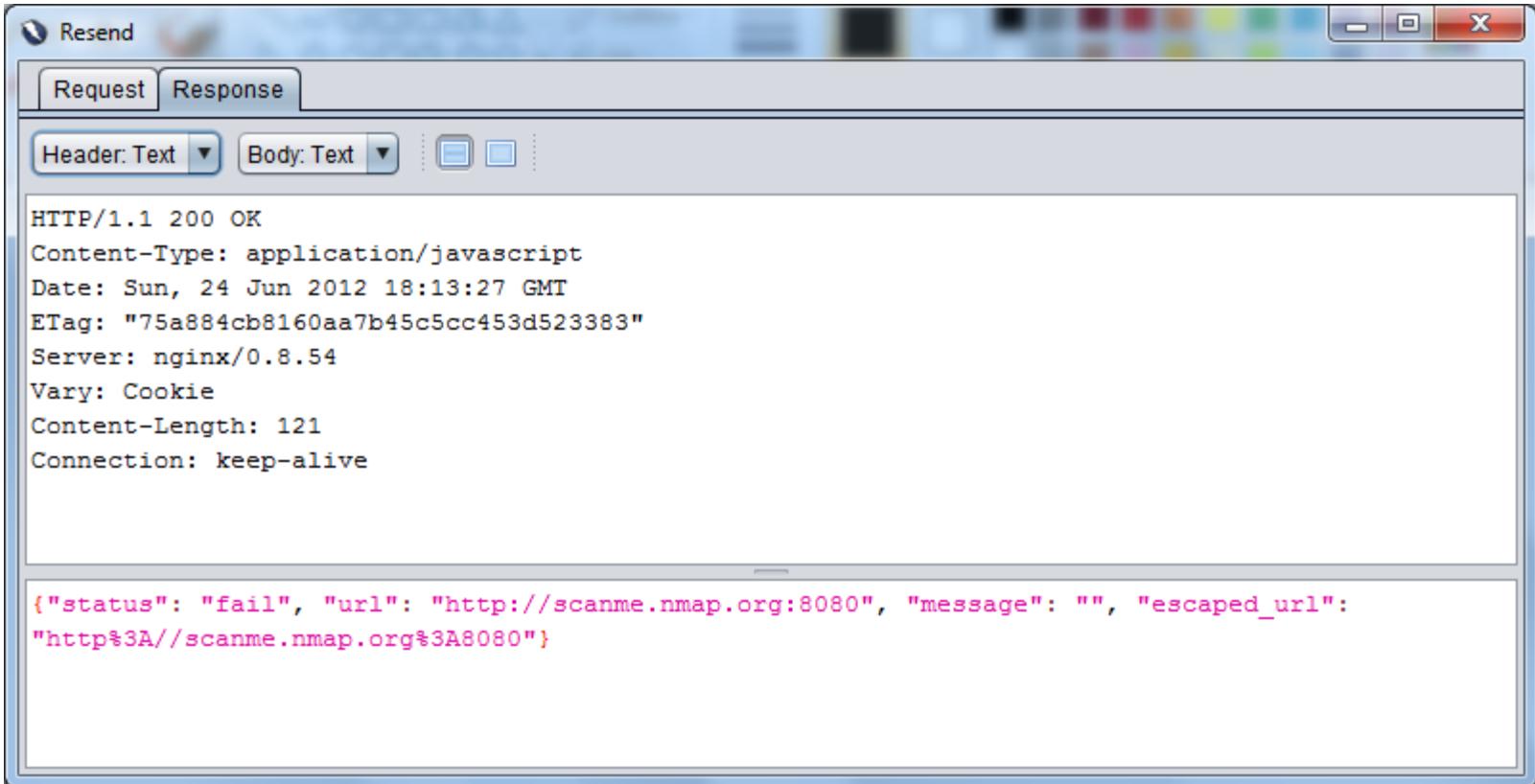
```
HTTP/1.1 200 OK
Content-Type: application/javascript
Date: Sun, 24 Jun 2012 18:11:47 GMT
ETag: "8871a36e15d4a0a419ce005950b0464a"
Server: nginx/0.8.54
Vary: Cookie
Content-Length: 129
Connection: keep-alive
```

The body of the response contains a JSON object:

```
{"status": "fail", "url": "http://scanme.nmap.org:22", "message": "content-type", "escaped_url": "http%3A//scanme.nmap.org%3A22"}
```

Application specific response for open non-HTTP ports

cat pinterest



The screenshot shows a browser developer tools Network tab with a failed request to `http://scanme.nmap.org:8080`. The request was made at `18:13:27` on `Sun, 24 Jun 2012`. The response status is `200 OK` with a content type of `application/javascript`. The response body contains a JSON object indicating a failure: `{"status": "fail", "url": "http://scanme.nmap.org:8080", "message": "", "escaped_url": "http%3A//scanme.nmap.org%3A8080"}`.

Header	Value
HTTP/1.1 200 OK	
Content-Type	application/javascript
Date	Sun, 24 Jun 2012 18:13:27 GMT
ETag	"75a884cb8160aa7b45c5cc453d523383"
Server	nginx/0.8.54
Vary	Cookie
Content-Length	121
Connection	keep-alive

```
{"status": "fail", "url": "http://scanme.nmap.org:8080", "message": "", "escaped_url": "http%3A//scanme.nmap.org%3A8080"}
```

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>

All images are the property of their respective creators.



Riyaz Ahemed Walikar

@riyazwalikar

<http://www.riyazwalikar.com>

