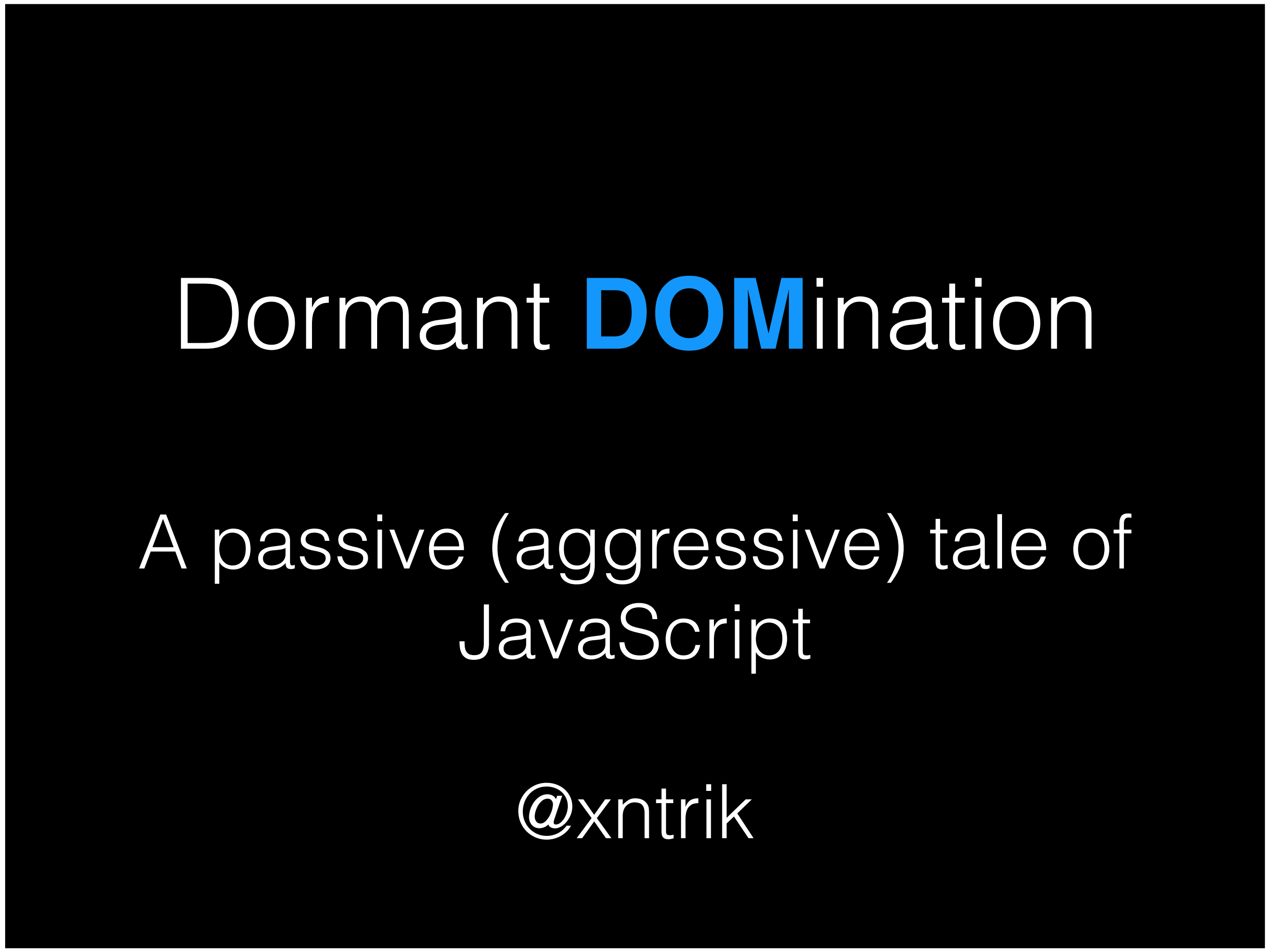


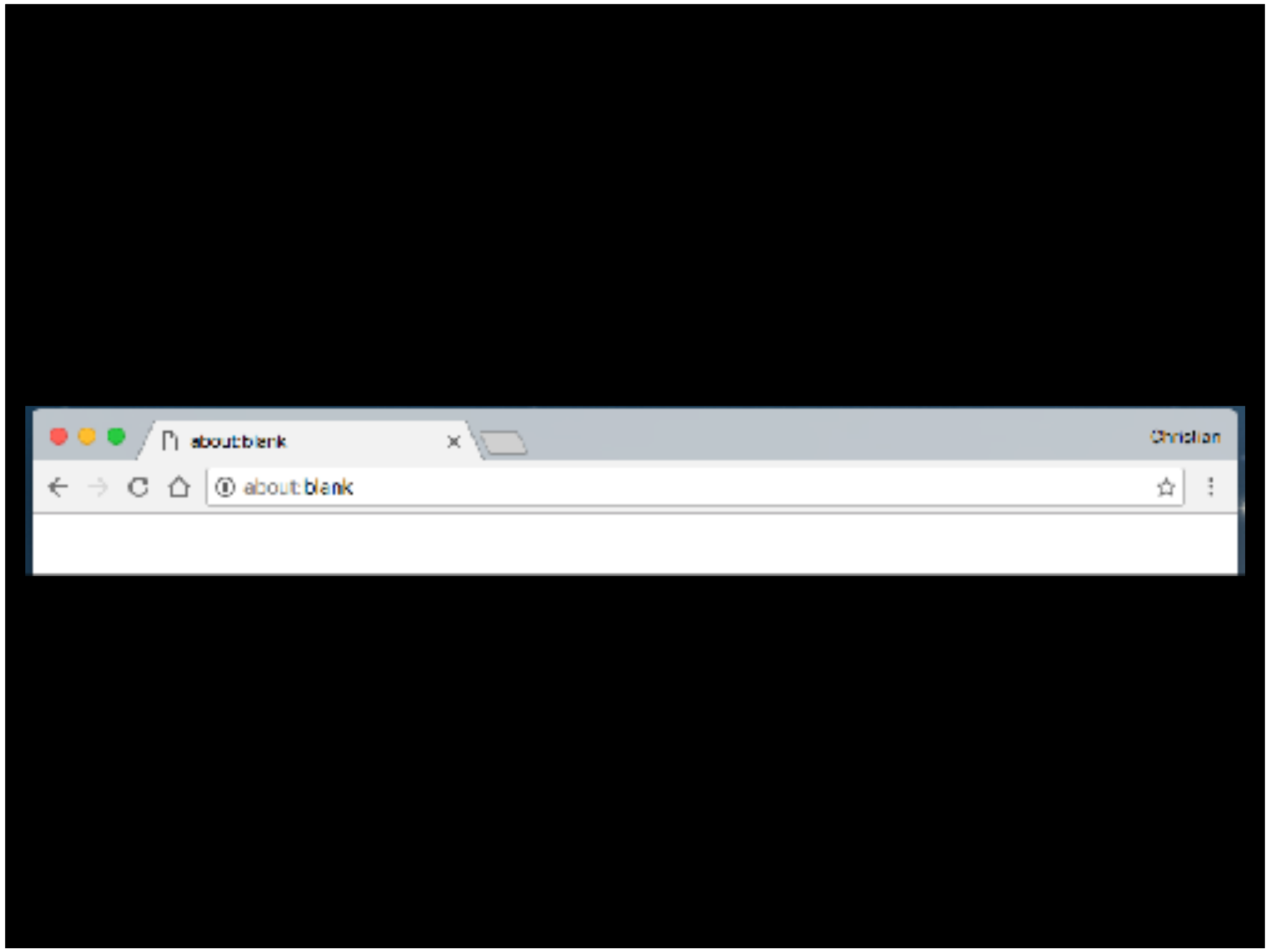
Dormant **DOM**ination

A passive (aggressive) tale of JavaScript

@xntrik

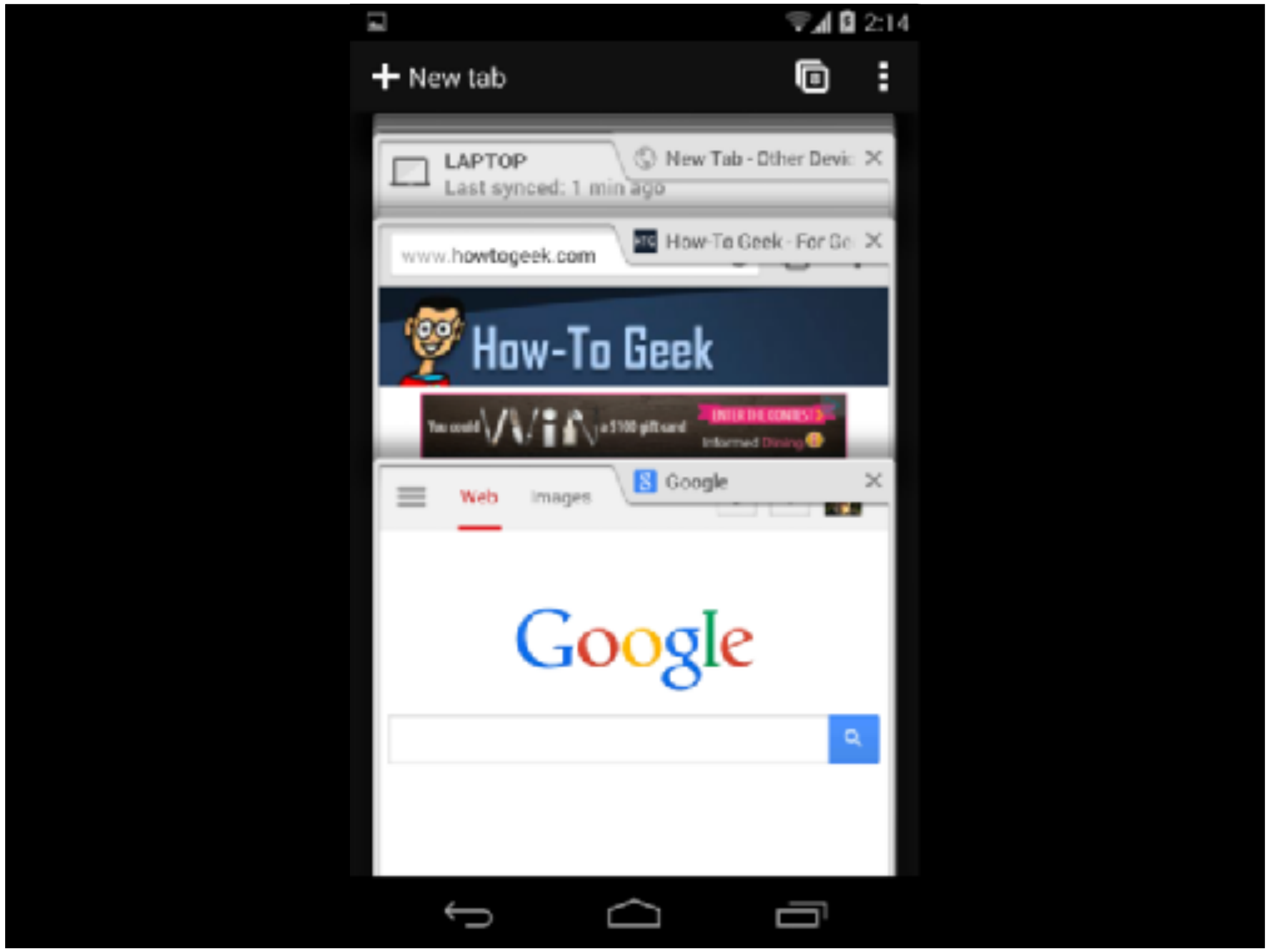
Hi everyone, my name's Christian Frichot - and this talk is all about standard JavaScript features and ways in which the 'browser-enabled' attack surface continues to change.

Now, I have to preface the rest of this talk that my work on the BeEF project, and associated fuckery with JavaScript has nothing to do with my employer, and the opinions presented are my own and don't reflect my employer.



And we obviously can’t talk about JavaScript without talking about the incredible power, and pervasiveness, of web browsers. I adore Chrome. It looks so simple, 4 buttons and a burger menu. .. that's all it is.

Think about how many different browser’s you run.

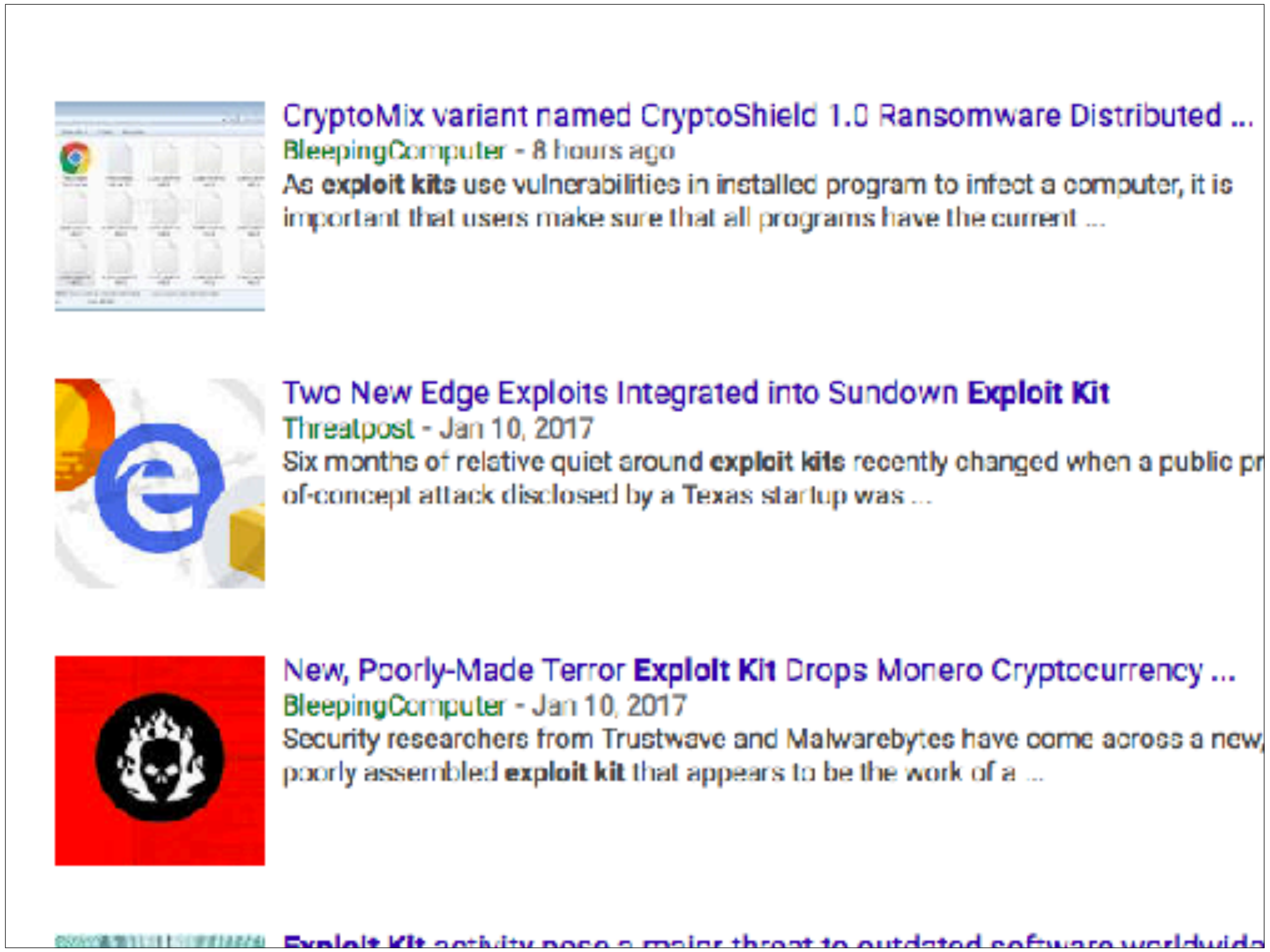


Now think about how many you run on each of your devices. Now think about how many tabs you have on each. In fact, pull out your phones, if you haven't already, and pull up Chrome or whatever browser you use, and check out the "tab" count.

And this is just on the device you carry in your pocket, let me guess that the count is MUCh higher on your laptops or desktops.

Not all bad, I mean it’s just JS right? What’s the worst thing that can happen? True.

In most circumstances it’s unlikely that malicious JS will, by itself, immediately result in arbitrary execution on a victim's computer.



CryptoMix variant named CryptoShield 1.0 Ransomware Distributed ...
BleepingComputer - 8 hours ago
As **exploit kits** use vulnerabilities in installed program to infect a computer, it is important that users make sure that all programs have the current ...

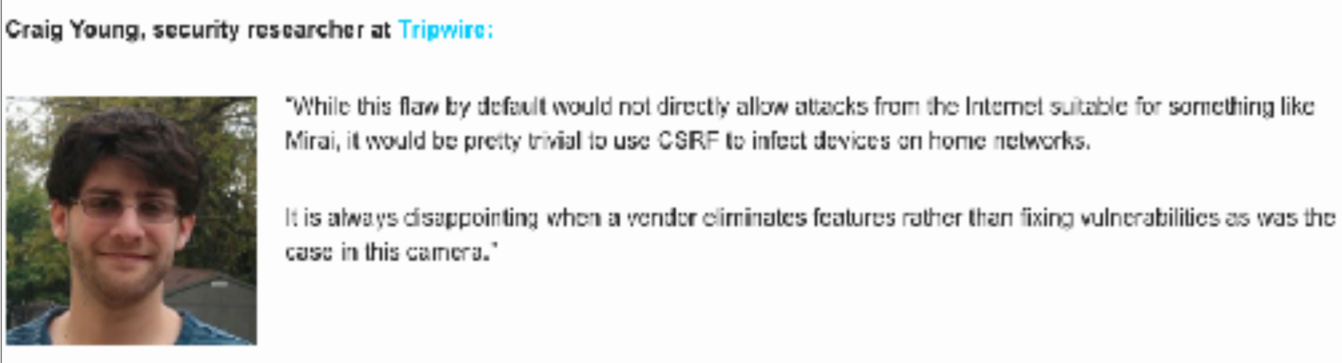
Two New Edge Exploits Integrated into Sundown **Exploit Kit**
Threatpost - Jan 10, 2017
Six months of relative quiet around **exploit kits** recently changed when a public pr of-concept attack disclosed by a Texas startup was ...

New, Poorly-Made Terror **Exploit Kit** Drops Monero Cryptocurrency ...
BleepingComputer - Jan 10, 2017
Security researchers from Trustwave and Malwarebytes have come across a new, poorly assembled **exploit kit** that appears to be the work of a ...

Exploit Kit activity pose a major threat to outdated software worldwide

But let’s not forget most exploit kits are mostly built on using JS to deliver malicious payloads to your document renderers etc.

Samsung SmartCam Security Camera Vulnerability

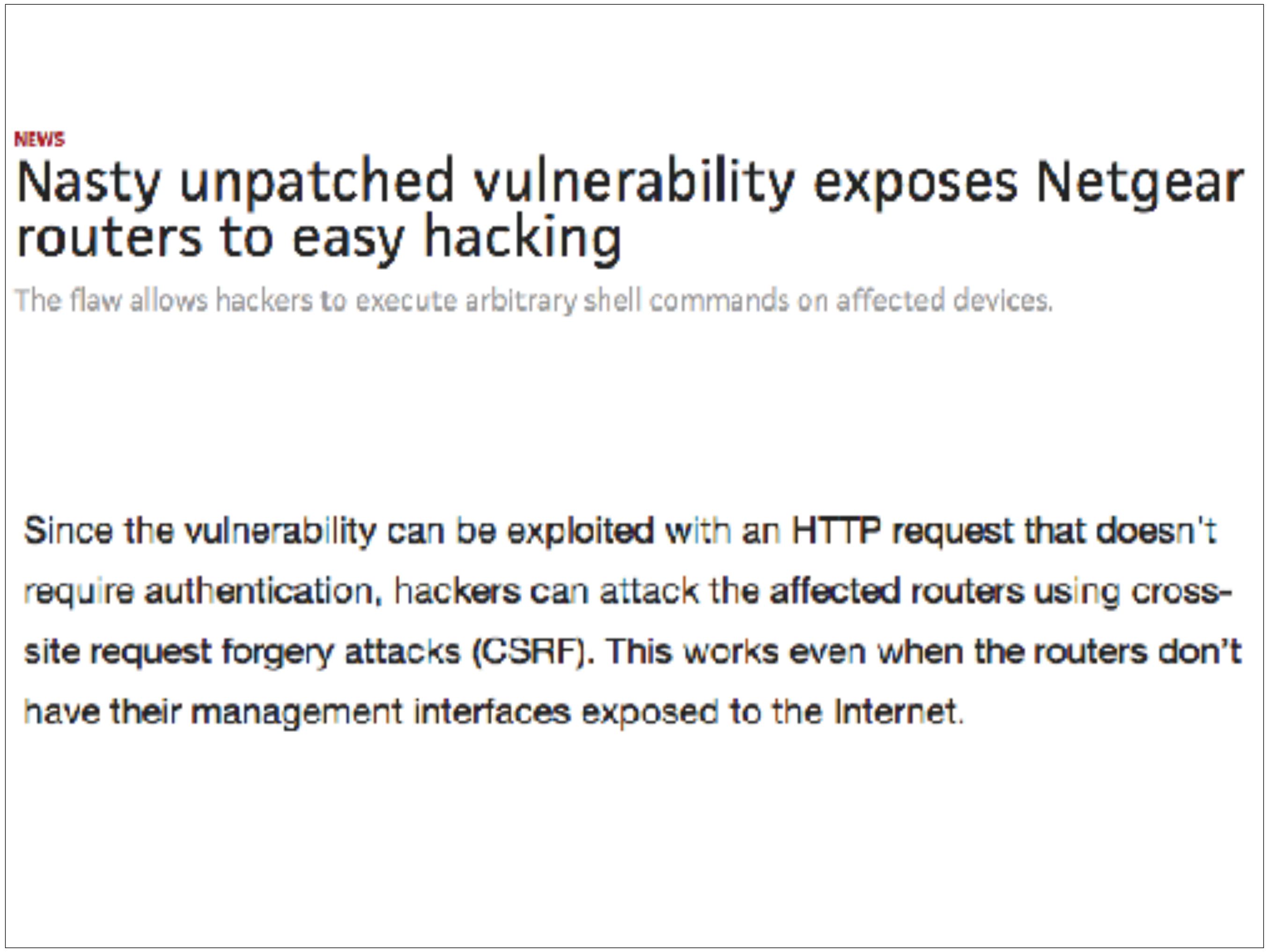


Craig Young, security researcher at Tripwire:

"While this flaw by default would not directly allow attacks from the Internet suitable for something like Mirai, it would be pretty trivial to use CSRF to infect devices on home networks.

It is always disappointing when a vendor eliminates features rather than fixing vulnerabilities as was the case in this camera."

And of course, there's all the vulnerable CSRF payloads out there



NEWS

Nasty unpatched vulnerability exposes Netgear routers to easy hacking

The flaw allows hackers to execute arbitrary shell commands on affected devices.

Since the vulnerability can be exploited with an HTTP request that doesn't require authentication, hackers can attack the affected routers using cross-site request forgery attacks (CSRF). This works even when the routers don't have their management interfaces exposed to the Internet.

The the CSRF payloads I'm more interested in are those that aren't likely to be Internet-exposed. So think of all the internal management systems you use, and how thorough your app sec testing of internal systems is.

So what's the issue here?

In-spite of the **SOP**

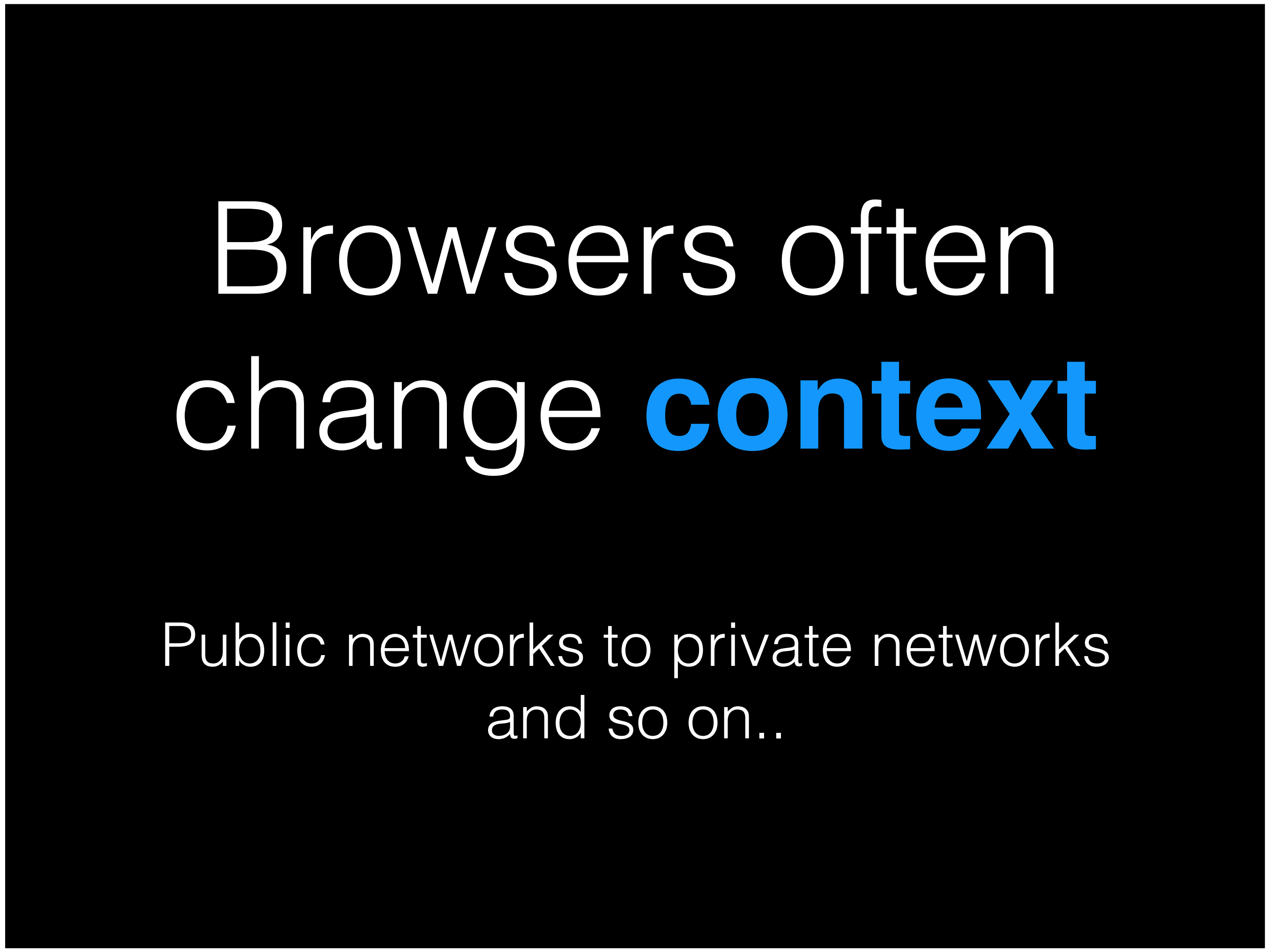
there still exists methods to send somewhat arbitrary requests to gather information about...

Where am I?

Thank you **WebRTC**

What is near me?

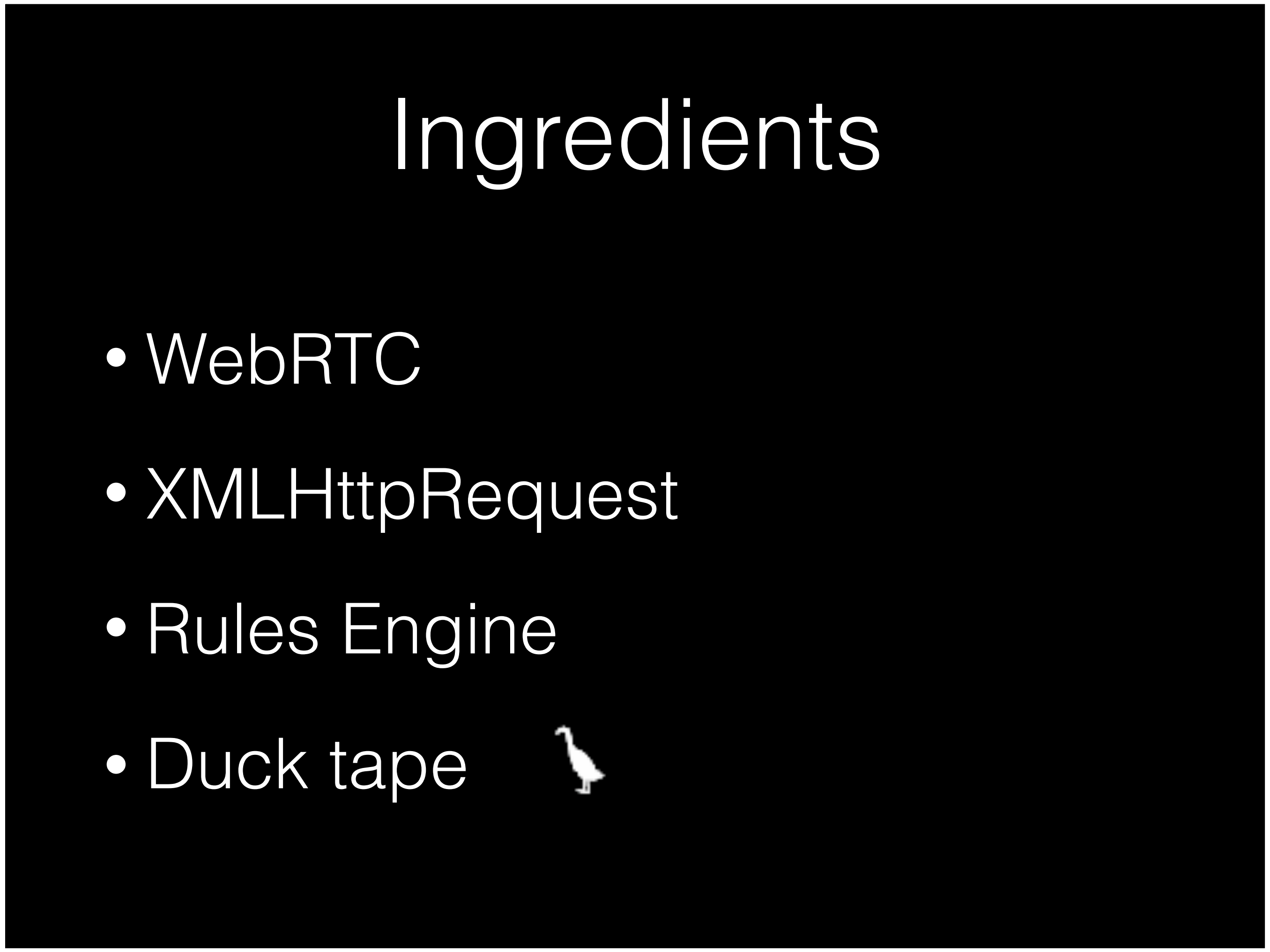
Thank you **XMLHttpRequest** + timeouts



Browsers often change **context**

Public networks to private networks
and so on..

Combined with the fact that browsers often change context..



Ingredients

- WebRTC
- XMLHttpRequest
- Rules Engine
- Duck tape 🦆

To demonstrate why this is bad .. we need to cludge some stuff together

Oh, and of course beef.

WebRTC

- window.webkitRTCPeerConnection
- onicecandidate

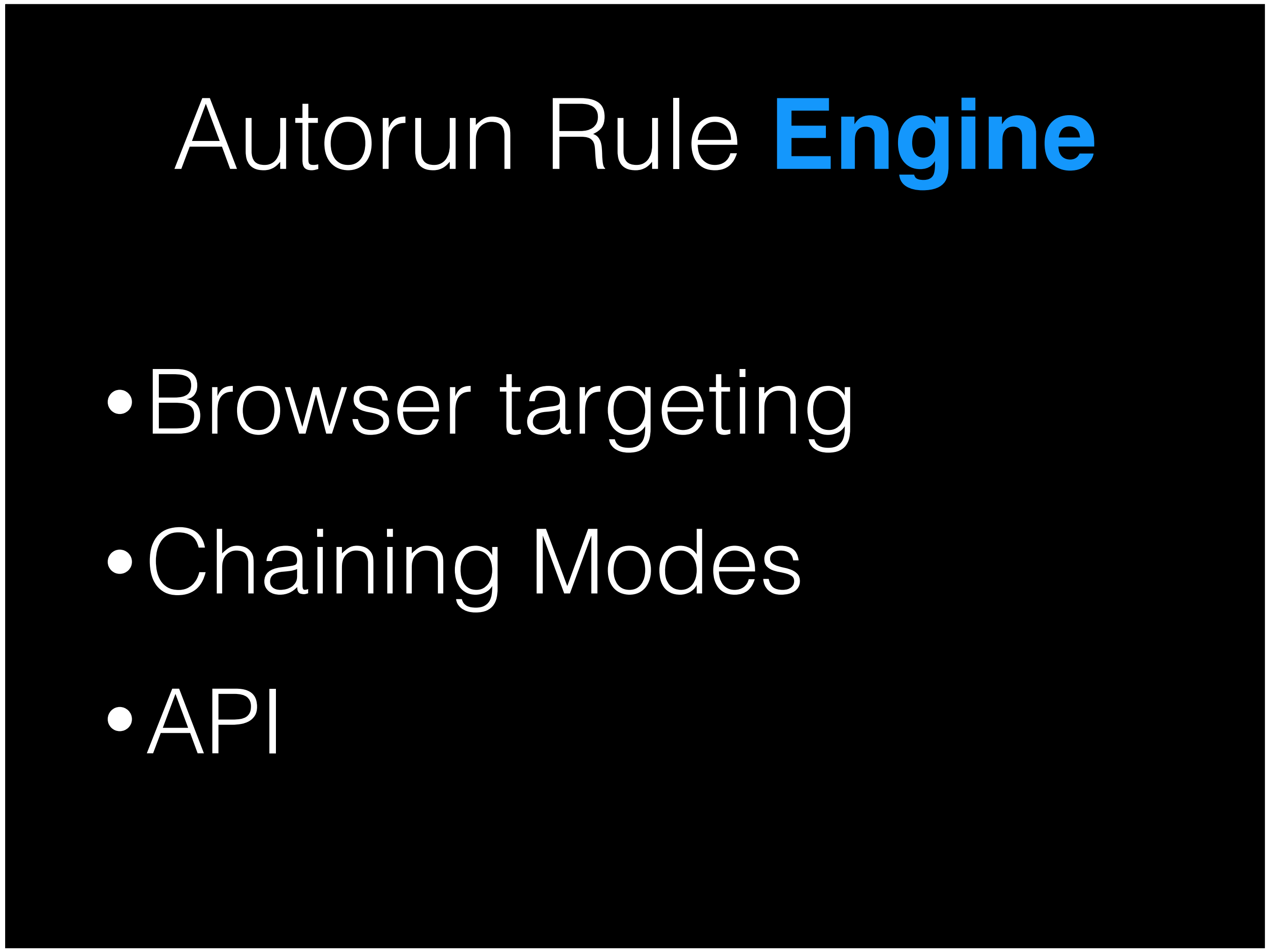
Interactive Connectivity Establishment (ICE)

{"type":"candidate","label":0,"id":"sdparta_0","candidate":"candidate:1356517016 1 udp 2113937151 **192.168.0.113** 55725 typ host generation 0 ufrag okAa network-cost 50"}

XMLHttpRequest

- xhr = new XMLHttpRequest()
- xhr.onreadystatechange
- Measure how long it takes for the state to change
  - Shorter than timeout? - Port open

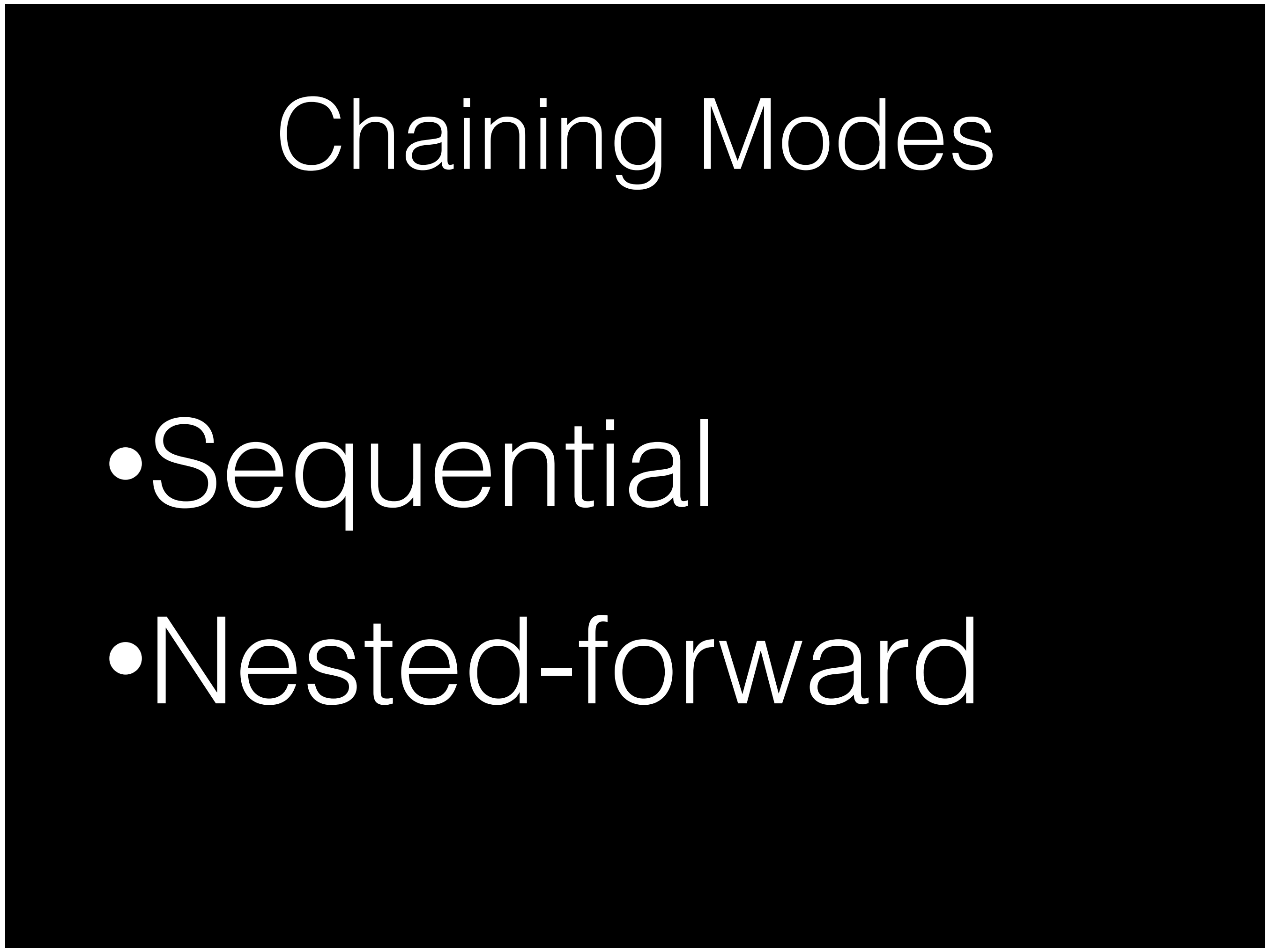
We also need a rules engine



Autorun Rule **Engine**

- Browser targeting
- Chaining Modes
- API

Beef's is called the ARE, or Autorun Rule Engine

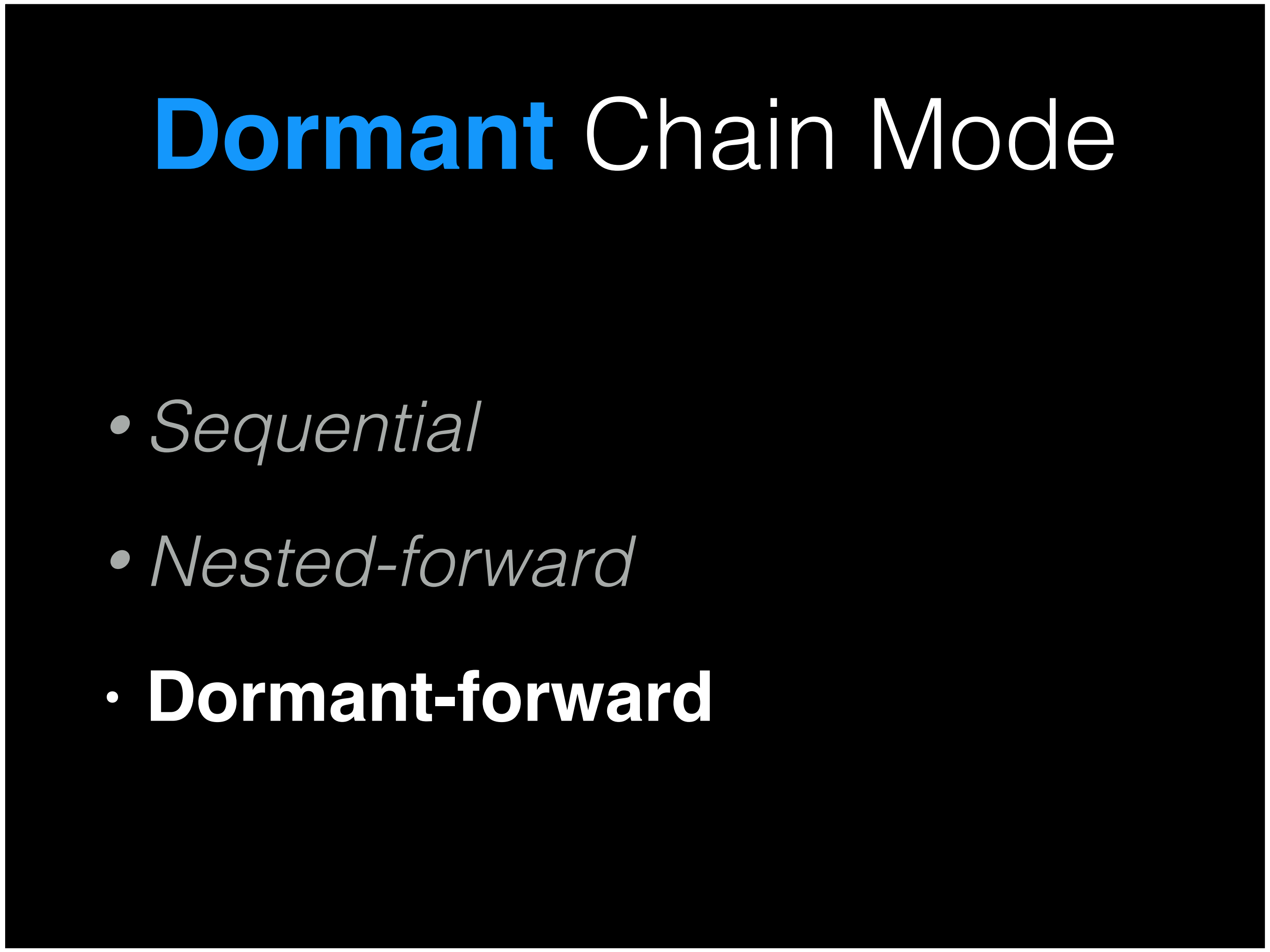


Chaining Modes

- Sequential
- Nested-forward

```
"modules": [
    {"name": "get_internal_ip_webrtc",
      "condition": null,
      "code": null,
      "options": {}
    },
    {"name": "internal_network_fingerprinting",
      "condition": "status==1",
      "code": "var s=get_internal_ip_webrtc_mod_output.split('.');var start=parseInt(s[3])-1;var
end=parseInt(s[3])+1;var mod_input =
s[0]+'.'+s[1]+'.'+s[2]+'.'+start+'-'+s[0]+'.'+s[1]+'.'+s[2]+'.'+end;",
      "options": {
        "ipRange":"<<mod_input>>",
        "ports":"80",
        "threads":"5",
        "wait":"2",
        "timeout":"10"
      }
```

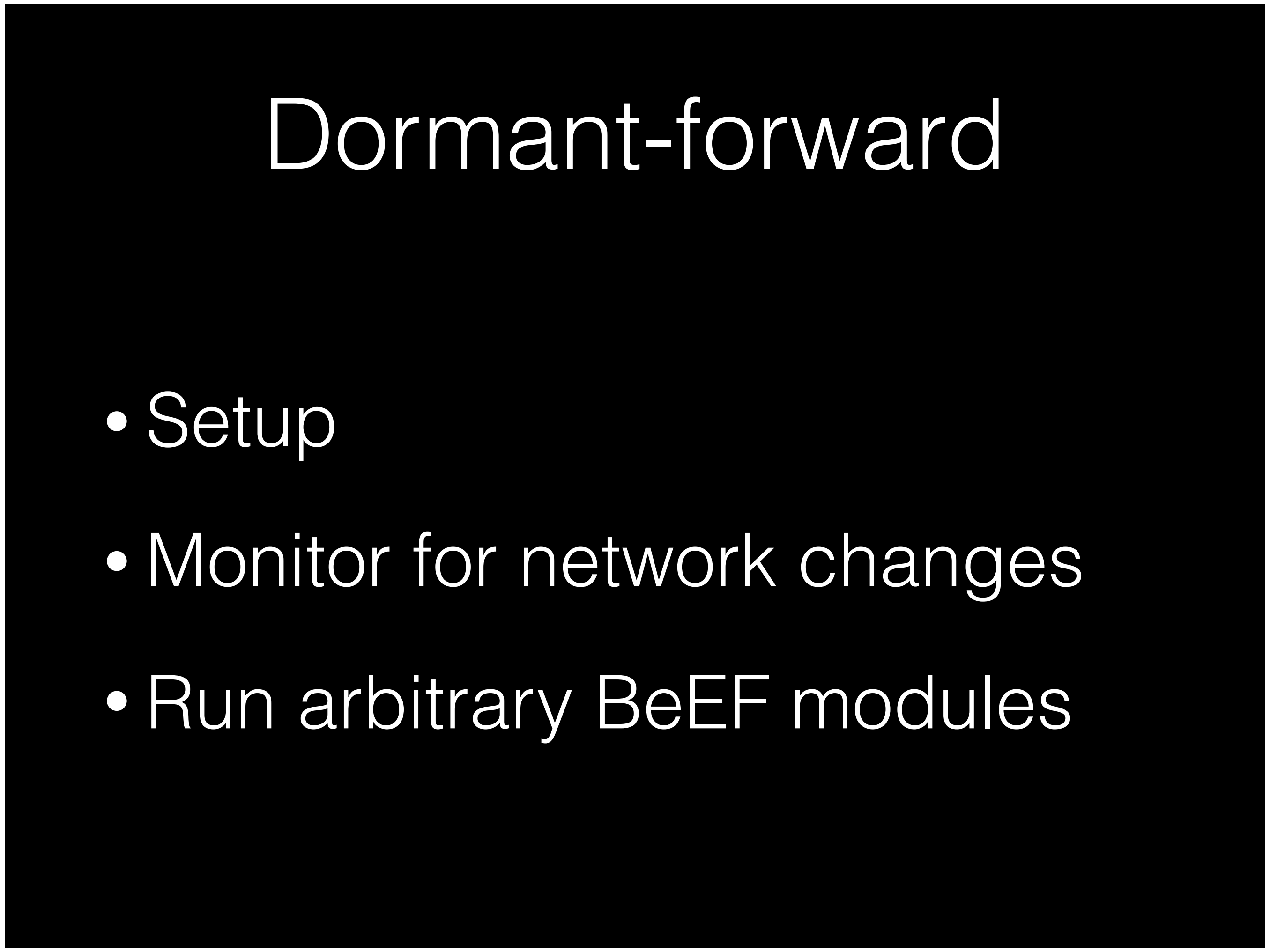
Call N modules, where module N is executed only if N-1 returns a certain status. Module N can use as input the output from module N-1 (eventually mangling it before processing it).



Dormant Chain Mode

- *Sequential*
- *Nested-forward*
- **Dormant-forward**

Wrapping this all together I've hacked together a new ARE chain_mode called dormant-forward



Dormant-forward

- Setup
- Monitor for network changes
- Run arbitrary BeEF modules

Setup

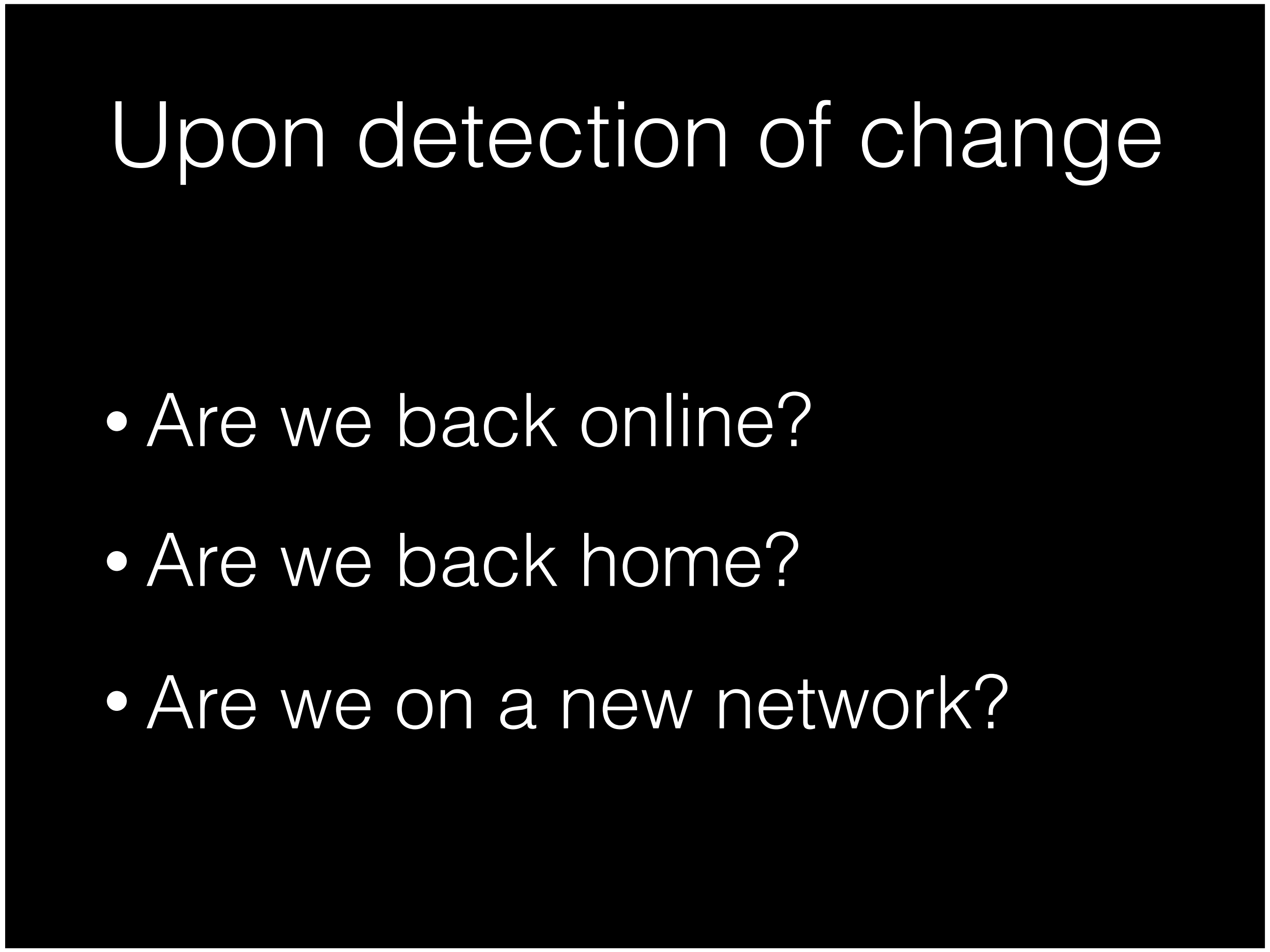
- Where am I?
  - RTC to gather LAN IP(s)
  - New /aslookup service in BeEF to get external details

Aslookup

```
dig +short
207.207.152.50.origin.asn.cymru.com TXT

 => 7922 | 50.128.0.0/9 | US | arin | 2010-10-21

dig +short AS7922.asn.cymru.com TXT

 => 7922 | US | arin |  | COMCAST-7922 -
 Comcast Cable Communications, LLC, US
```

Setup cont'd

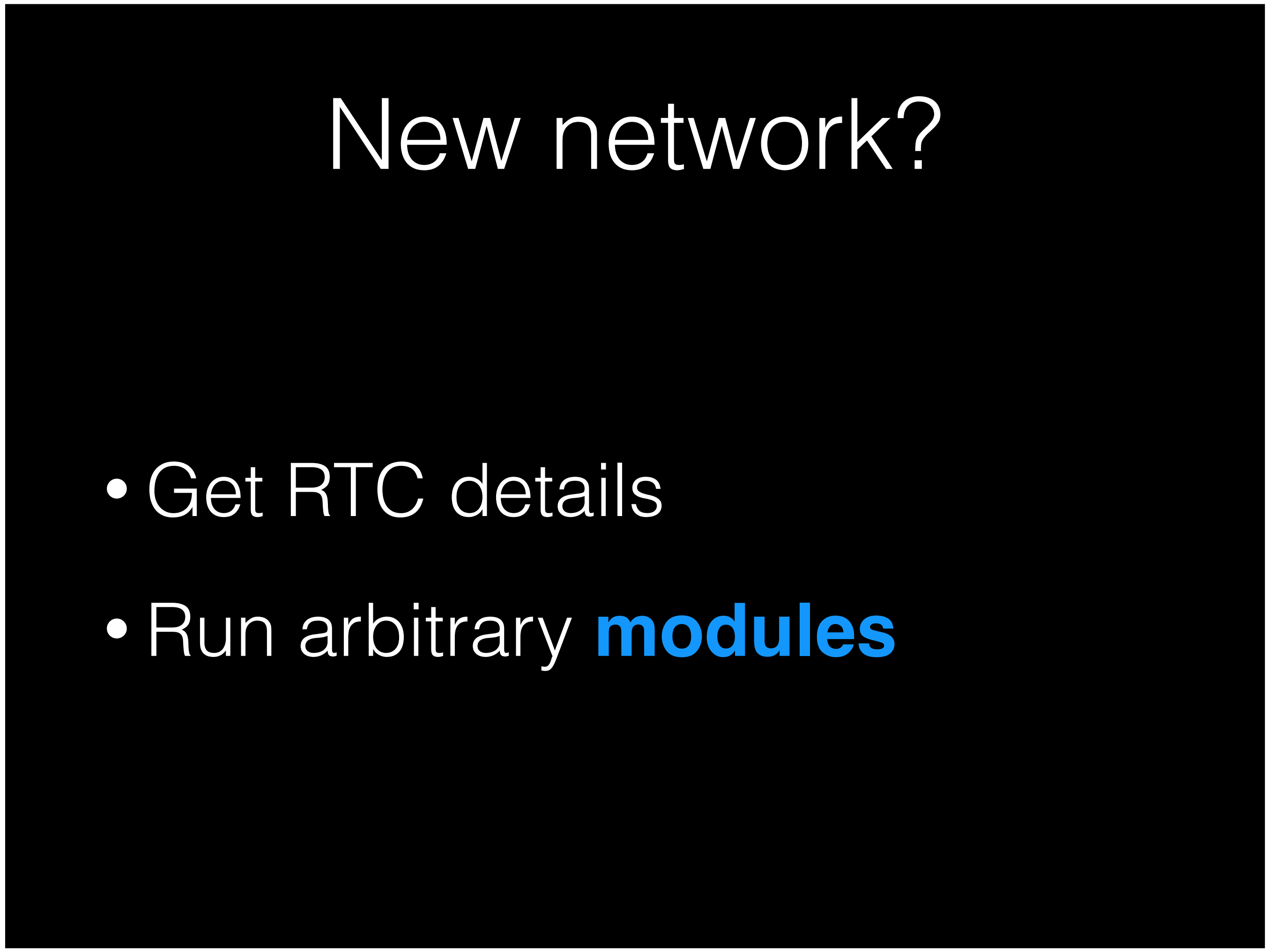
- We have internal and external details
- Kick off timers & monitor **window.navigator.onLine**



Upon detection of change

- Are we back online?
- Are we back home?
- Are we on a new network?

We use some fuzzy matching here, in case you return back to the perceived original network.

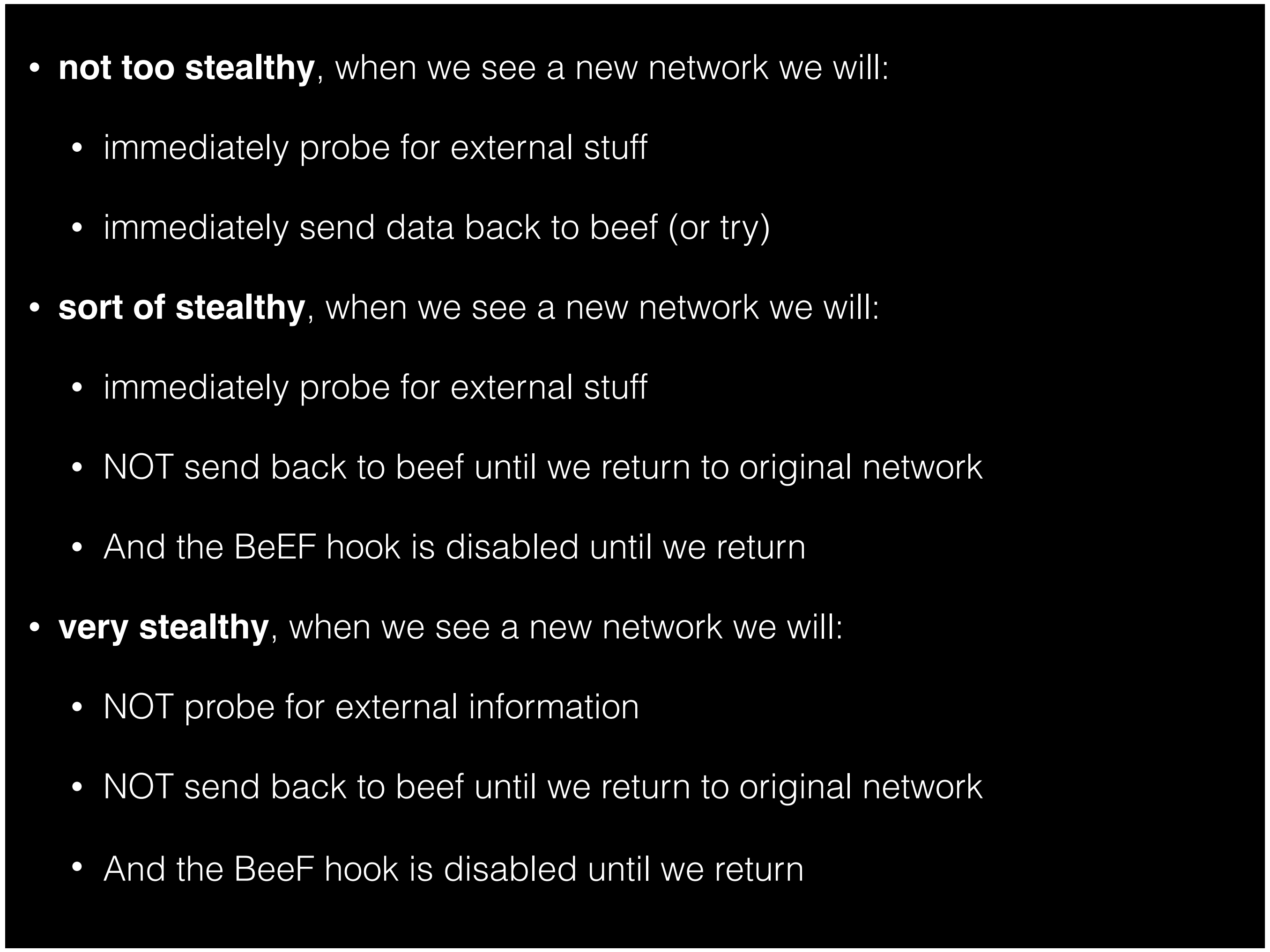


New network?

- Get RTC details
- Run arbitrary **modules**

Configurable settings

- How ***stealthy*** will we be?
- What happens when we return home



- **not too stealthy**, when we see a new network we will:
  - immediately probe for external stuff
  - immediately send data back to beef (or try)
- **sort of stealthy**, when we see a new network we will:
  - immediately probe for external stuff
  - NOT send back to beef until we return to original network
  - And the BeEF hook is disabled until we return
- **very stealthy**, when we see a new network we will:
  - NOT probe for external information
  - NOT send back to beef until we return to original network
  - And the BeeF hook is disabled until we return

When we return home

// endMode

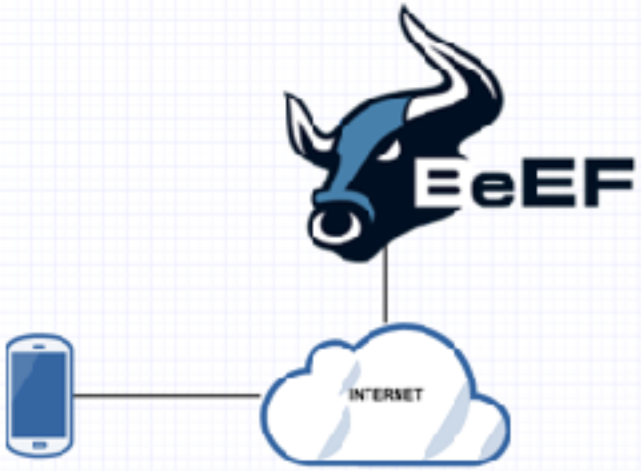
// 0 - just run indefinitely - even after returning home

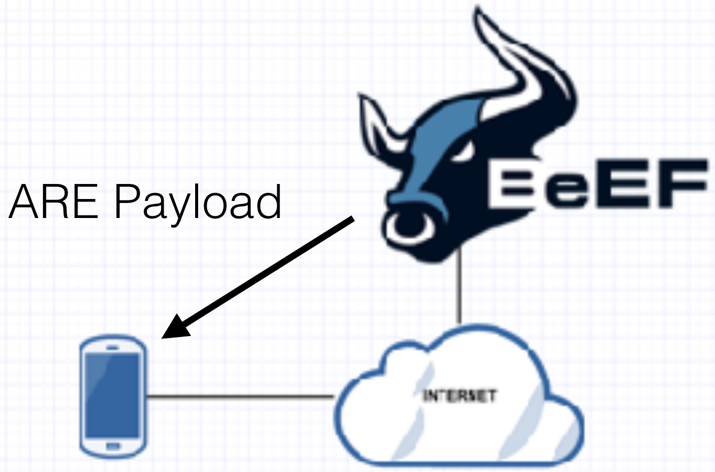
// 1 - after returning home and dumping data - kill all timers (*and humans*)

this.endMode = end_mode;

```
  "modules": [
    {"name": "ping_sweep",
      "condition": null,
      "code": "var s=outer_sequential_mod_output.split('.');var start = s[0]+'.'+s[1]+'.'+s[2]+'.117'; var end = s[0]+'.'+s[1]+'.'+s[2]+'.120'; var mod_input = start+'-'+end;",
      "options": {
        "rhosts":"<<mod_input>>",
        "threads":"6"
      }
    }
  ],
  "chain_mode": "dormant-forward",
  "stealth_mode": 2,
  "dormant_end_mode": 1
```

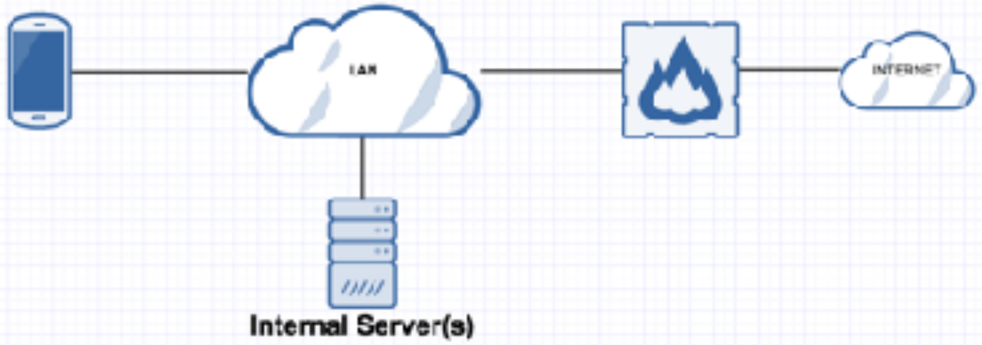
Setup

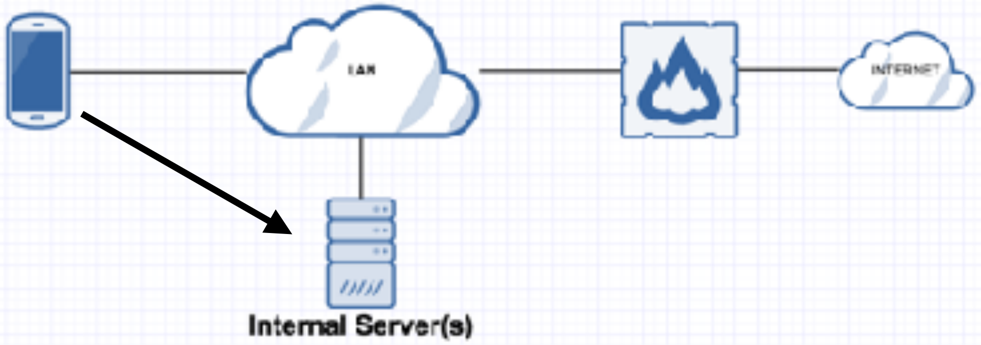




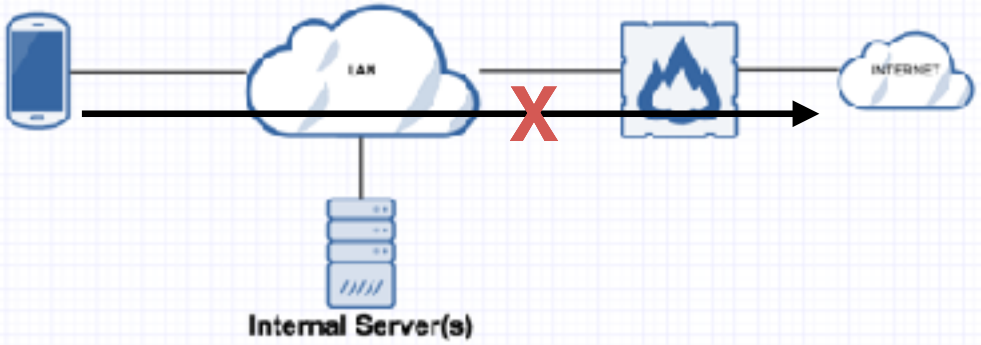
ARE Payload

New network?



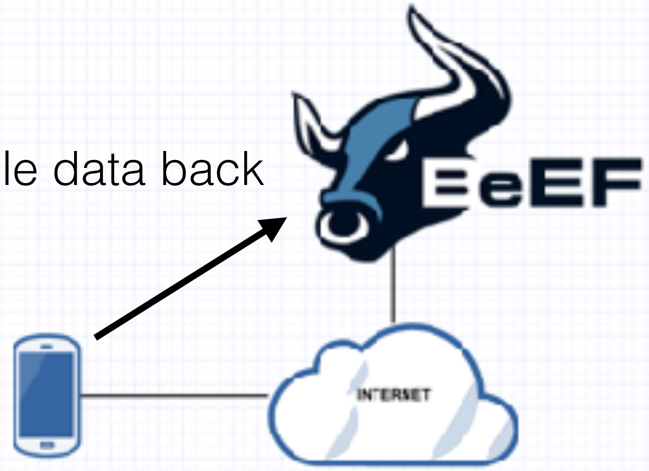


Discover internal LAN resources
PROBE internal LAN resources

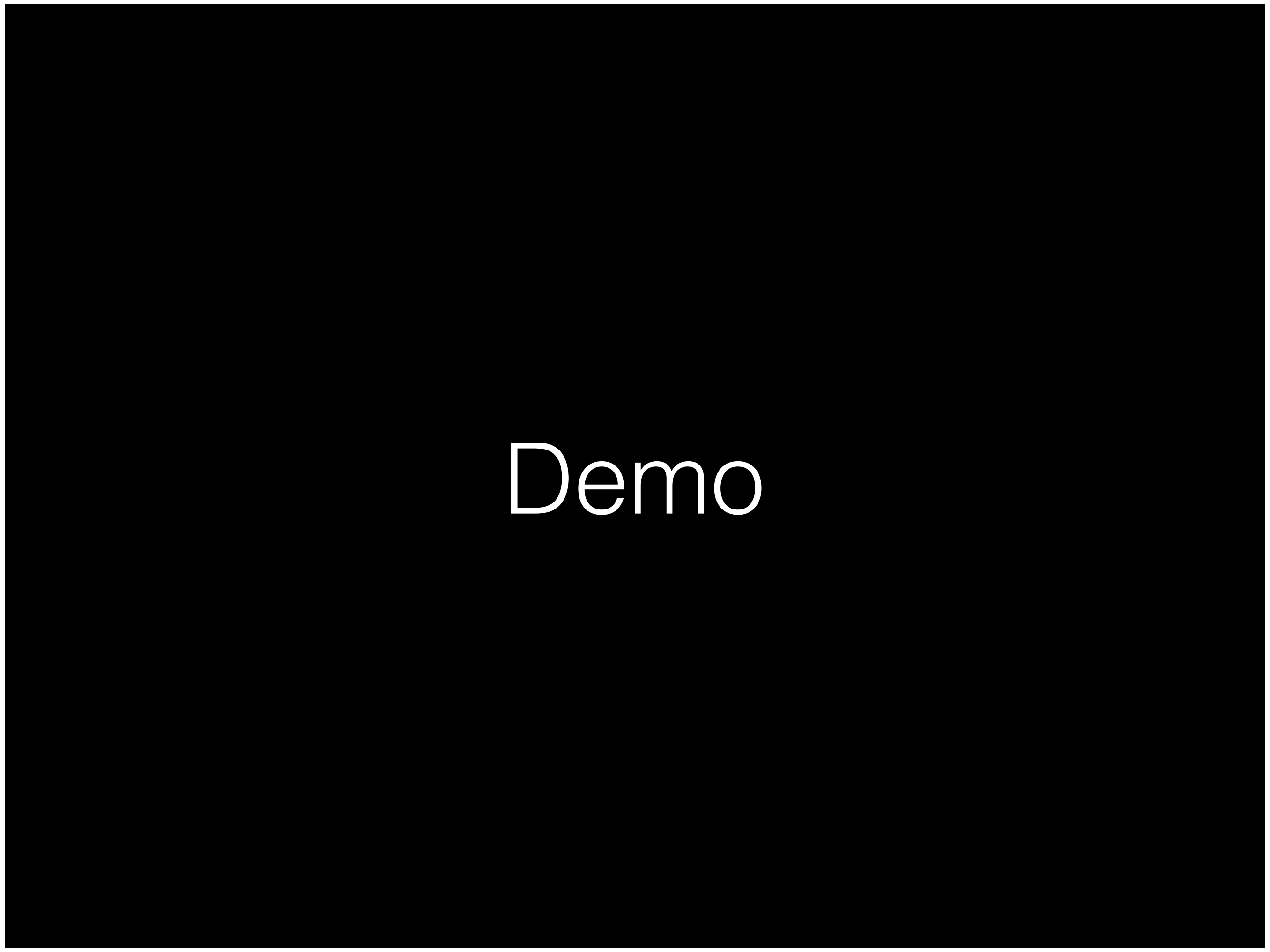


Don't talk to BeEF via the Internet

Back on the first network?



Send module data back



Demo

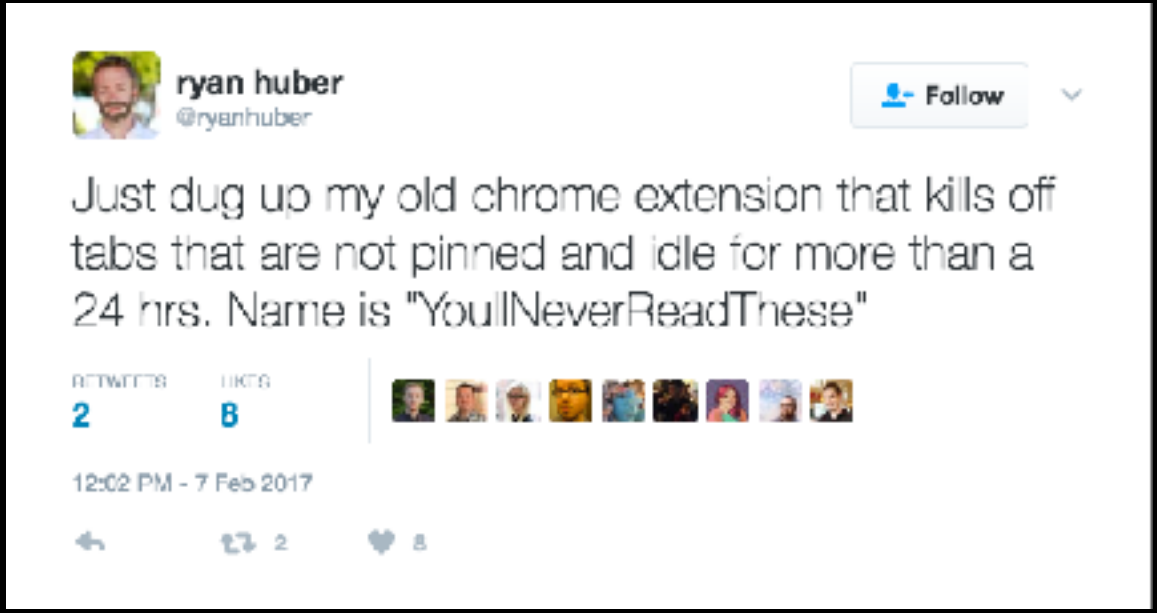
https://youtu.be/LG0FdueFtdE

[redacted]

Defensive Strategies

- Zero Trust Network
- XSS mitigations - CSP and friends ?
- CSRF mitigations - same-site cookies & friends??

Defensive Strategies



ryan huber
@ryanhuber

Follow

Just dug up my old chrome extension that kills off tabs that are not pinned and idle for more than a 24 hrs. Name is "YoullNeverReadThese"

RETWEETS 2 LIKES 8

12:02 PM - 7 Feb 2017

2 8

Future work

- Tidy up the code & push to git
- Work on separating the /aslookup service
- ARE* some modules don't work that great
- I'm sure there's more but I can't remember.

Thank you!

@xntrik

Special <3 to:

@antisnatchor
@wadealcorn
@_bcoles
@itgirljs
@SecureCloudDev
@asteriskinfosec
@inspirationduck