I Don’t Like Big Gateways (and I Cannot Lie)

spur.us/i-dont-like-big-gateways-and-i-cannot-lie/

Sean S. October 17, 2022

How IP Reputation Gets Large Gateways Wrong

Thanks to Network Address Translation (NAT), large organizations can get by with only a small number of public IP addresses. What this means is, to the backbone of the Internet, the employee watching funny cat videos on **youtube.com** will have the same IP address as the employee trying to get in on the next big sneaker drop on **nike.com**, and the employee betting on sports on **fanduel.com**, and the employee doomscrolling through **twitter.com**, and the employee viewing NSFW material on… some domain.

Implementing IP-level blocking on these large gateways can be akin to throwing the baby out with the bathwater. You may prevent the specific instance of malicious activity you are observing, but at the same time you could be blocking hundreds or thousands of legitimate users as well. In this way, a similarity exists between large gateways and VPN exits. Both serve as the egress point for a large number of users who are indistinguishable from each other at the network layer. This constitutes much of the appeal of VPNs for many users: it’s like firearm with multiple sets of fingerprints.

Meanwhile, mobile gateways can be especially troublesome. Due to the transient nature of the devices utilizing these gateways to route their Internet traffic from their phones or hotspots, the behavior you are observing may only be originating from an IP for only minutes — or even seconds — at a time.

All this to say, outright blocking these large gateways is infeasible for most organizations and services.

Residential Proxies and large gateways

Spur routinely gets support requests asking for help interpreting a specific result in our IP Context data. In the majority of cases, the IP address in question is a large gateway serving as an exit for one or many residential proxies, services that co-opt and sell the Internet connection of normal user devices. The unfortunate truth is these questions are tricky to answer generically as it heavily depends on a number of factors including:

- The business model of the service
- The use case of Spur’s IP Context data
- The nature of the abusive traffic being observed

Blocking or limiting access from datacenter proxies and static VPN exits can be relatively straightforward, but the growing use of residential proxies really muddies the water, as often just the knowledge that a residential proxy has been seen routing traffic through a given IP is not sufficient enough of a reason to take action against that IP.

As a rule of thumb, **the larger the gateway, the more likely you are to see one or many residential proxy services operating behind it.** This is just a numbers problem ultimately, as residential proxy services collectively operate on tens of millions of devices across the globe.

One proxy, many interpretations

Let's pick on a single residential proxy provider (Oxylabs is the lucky winner) and go over how you might interpret five different examples from Spur's Context API (v2) output. We won't speculate on how the proxy got on the device in the first place; that's an exercise for a different blog post.

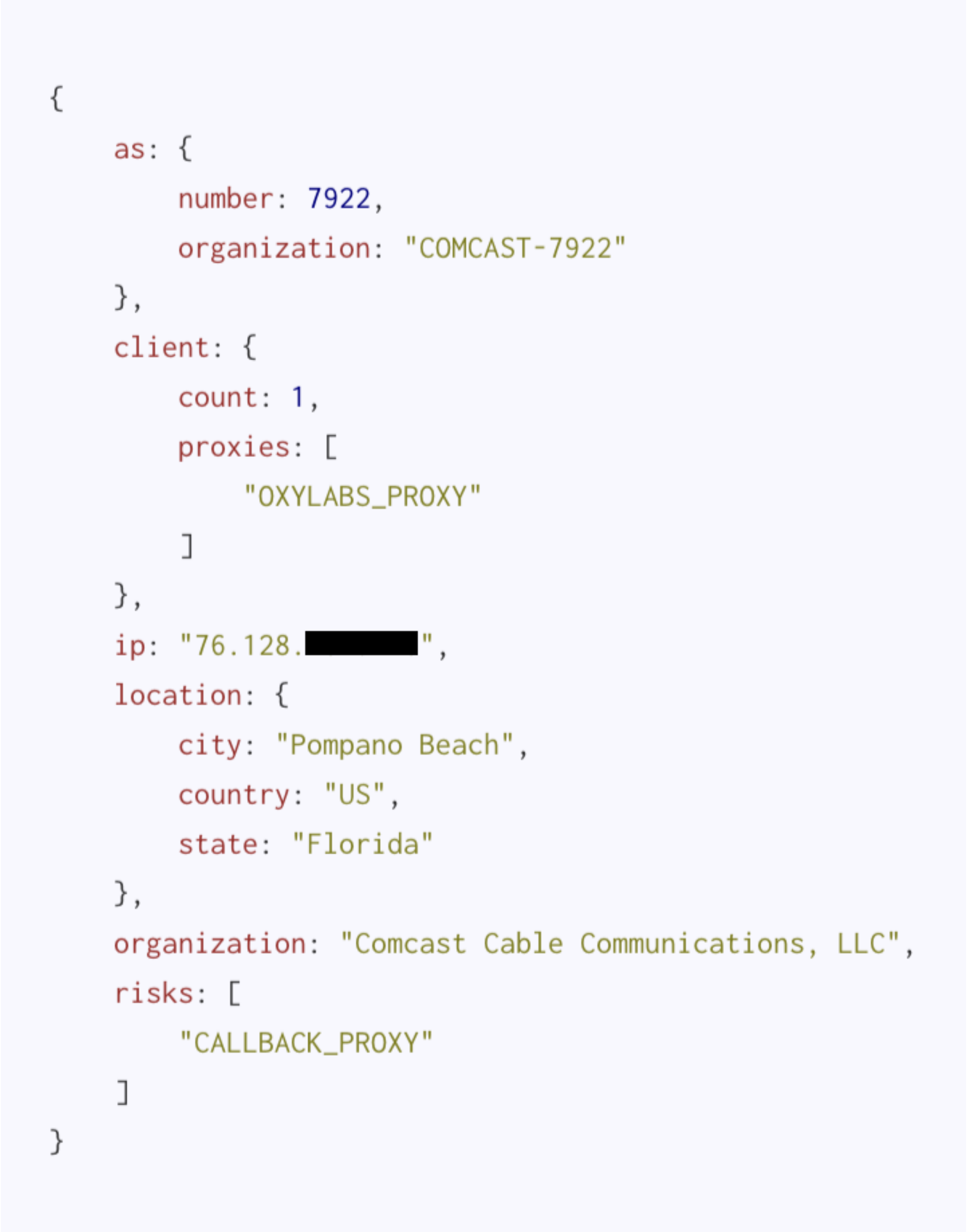
The scenarios we'll be covering:

- **Small gateway**: single-family homes, small businesses.
- **Medium gateway**: apartments, condos, small offices, libraries, schools, busy coffee shops, etc.
- **Large gateway**: office buildings, hotels, other large large institutions.
- **Datacenter**: Almost always a tunnel of some sort. Likely not a source of normal, legitimate user traffic in most cases.
- **Mobile gateway**: Could be anything, really. Not at all uncommon to see residential proxy service indicators.

In these examples, we're going to be focusing on a few key properties of the result set from Spur's Context API (v2):

- `as.organization` – Gives an indication to the general nature of the IP space in question via the Autonomous System to which the IP belongs.
- `client.count` – A *very conservative* approximation of the number of unique devices observed behind the given IP address on a daily basis.
- `organization` – This top-level property can sometimes give keen insight into the exact institution or business to whom the IP address is registered.
- `infrastructure` – Another top-level property that, if present, indicates the IP is known to be non-residential (though this doesn't mean it can't act as a gateway for residential proxies).

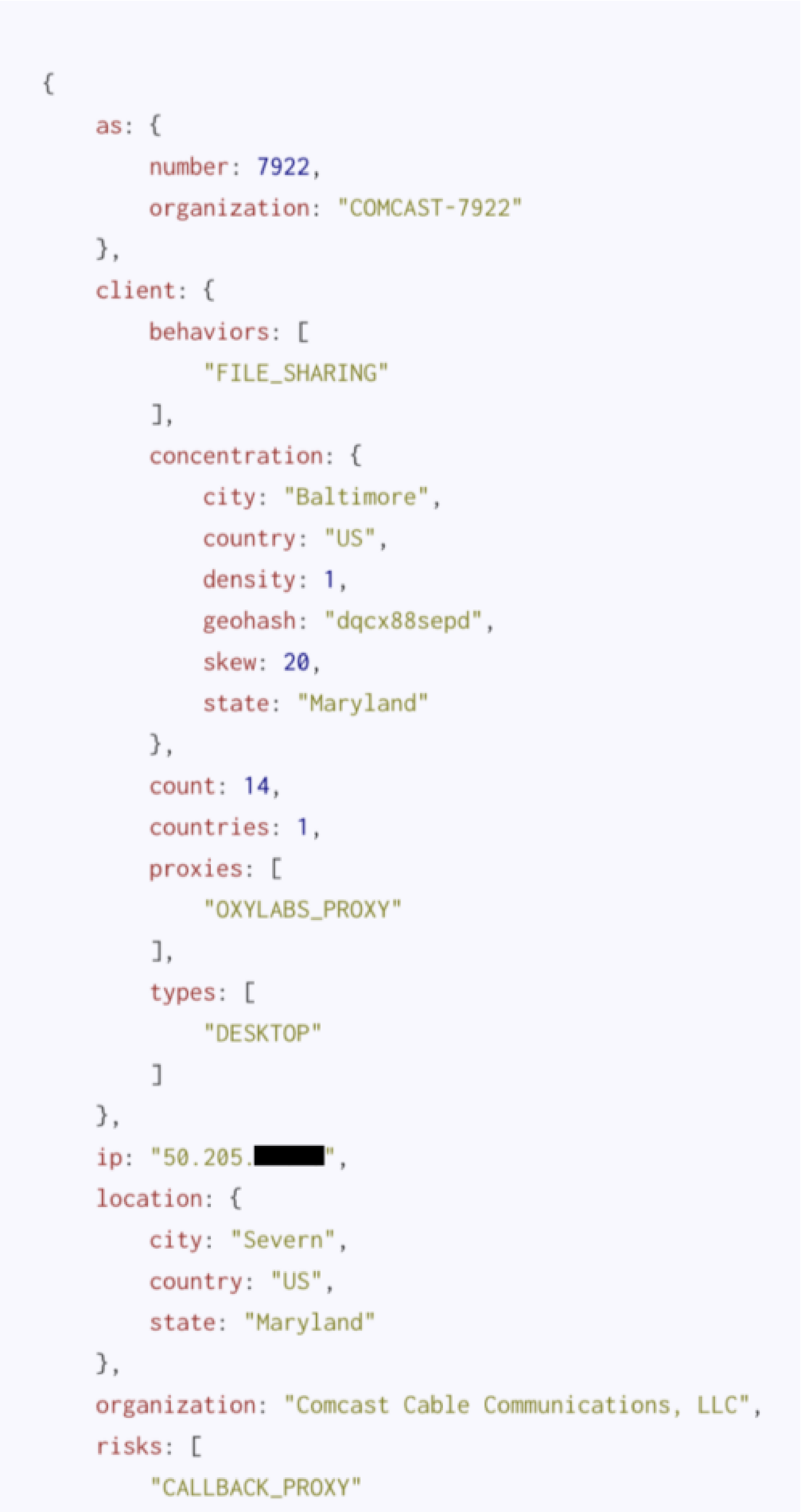
Let's start with the first three examples from consumer/business ISP space as noted by the `as.organization` property.



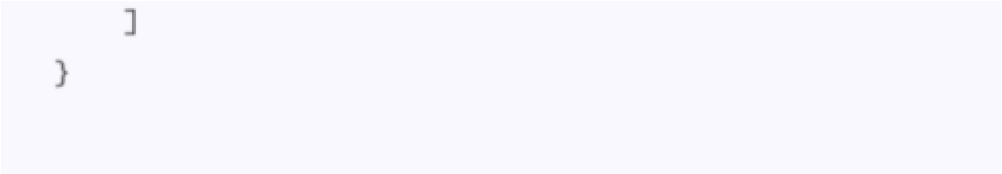
```
{
    as: {
        number: 7922,
        organization: "COMCAST-7922"
    },
    client: {
        count: 1,
        proxies: [
            "OXYLABS_PROXY"
        ]
    },
    ip: "76.128.[redacted]",
    location: {
        city: "Pompano Beach",
        country: "US",
        state: "Florida"
    },
    organization: "Comcast Cable Communications, LLC",
    risks: [
        "CALLBACK_PROXY"
    ]
}
```

Spur Context API results for a small gateway

Here we have what looks to be a **small gateway**; a Comcast customer from a single-family home or small business, with Oxylabs present. **This is the type of IP for which residential proxy services strive**: **innocuous but ubiquitous.** Without Spur's in-depth tracking of residential proxy services, anyone looking into this IP would meet a dead end through other open-source analysis. A decision to take action against (see: blocking) this IP would stem from the nature and volume of traffic originating from it. Ultimately, this could be a valid, legitimate user of your service who just also happens to be looking to get some extra cash by reselling their bandwidth.



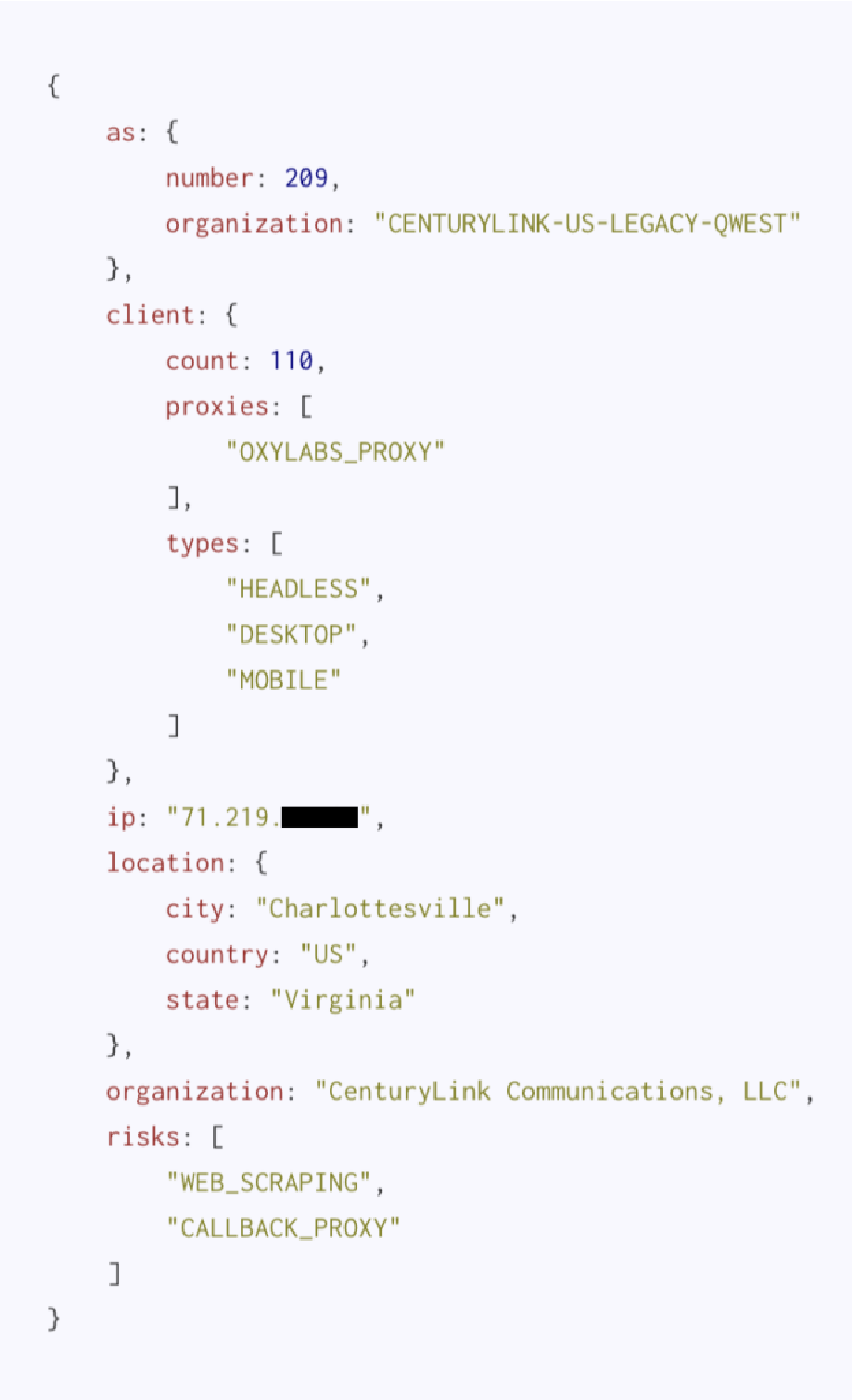
```
{
    as: {
        number: 7922,
        organization: "COMCAST-7922"
    },
    client: {
        behaviors: [
            "FILE_SHARING"
        ],
        concentration: {
            city: "Baltimore",
            country: "US",
            density: 1,
            geohash: "dqcx88sepd",
            skew: 20,
            state: "Maryland"
        },
        count: 14,
        countries: 1,
        proxies: [
            "OXYLABS_PROXY"
        ],
        types: [
            "DESKTOP"
        ]
    },
    ip: "50.205.[redacted]",
    location: {
        city: "Severn",
        country: "US",
        state: "Maryland"
    },
    organization: "Comcast Cable Communications, LLC",
    risks: [
        "CALLBACK_PROXY"
```



```
        ]
    }
```

Spur Context API results for a medium-sized gateway

We have a busier, **medium-sized gateway** here, with at least 14 devices detected behind, but still making use of Comcast IP space. This could be a small office, a school, or busy coffee shop with an open WiFi network. It starts getting harder to justify an outright block of IP addresses similar to this one. Further analysis with in-house and open source data is likely warranted. The device(s) with Oxylab installed could be regular denizens or patrons of the establishment, but rest assured there are many otherwise-normal users behind this IP on a daily basis.



```
{
    as: {
        number: 209,
        organization: "CENTURYLINK-US-LEGACY-QWEST"
    },
    client: {
        count: 110,
        proxies: [
            "OXYLABS_PROXY"
        ],
        types: [
            "HEADLESS",
            "DESKTOP",
            "MOBILE"
        ]
    },
    ip: "71.219.[redacted]",
    location: {
        city: "Charlottesville",
        country: "US",
        state: "Virginia"
    },
    organization: "CenturyLink Communications, LLC",
    risks: [
        "WEB_SCRAPING",
        "CALLBACK_PROXY"
    ]
}
```

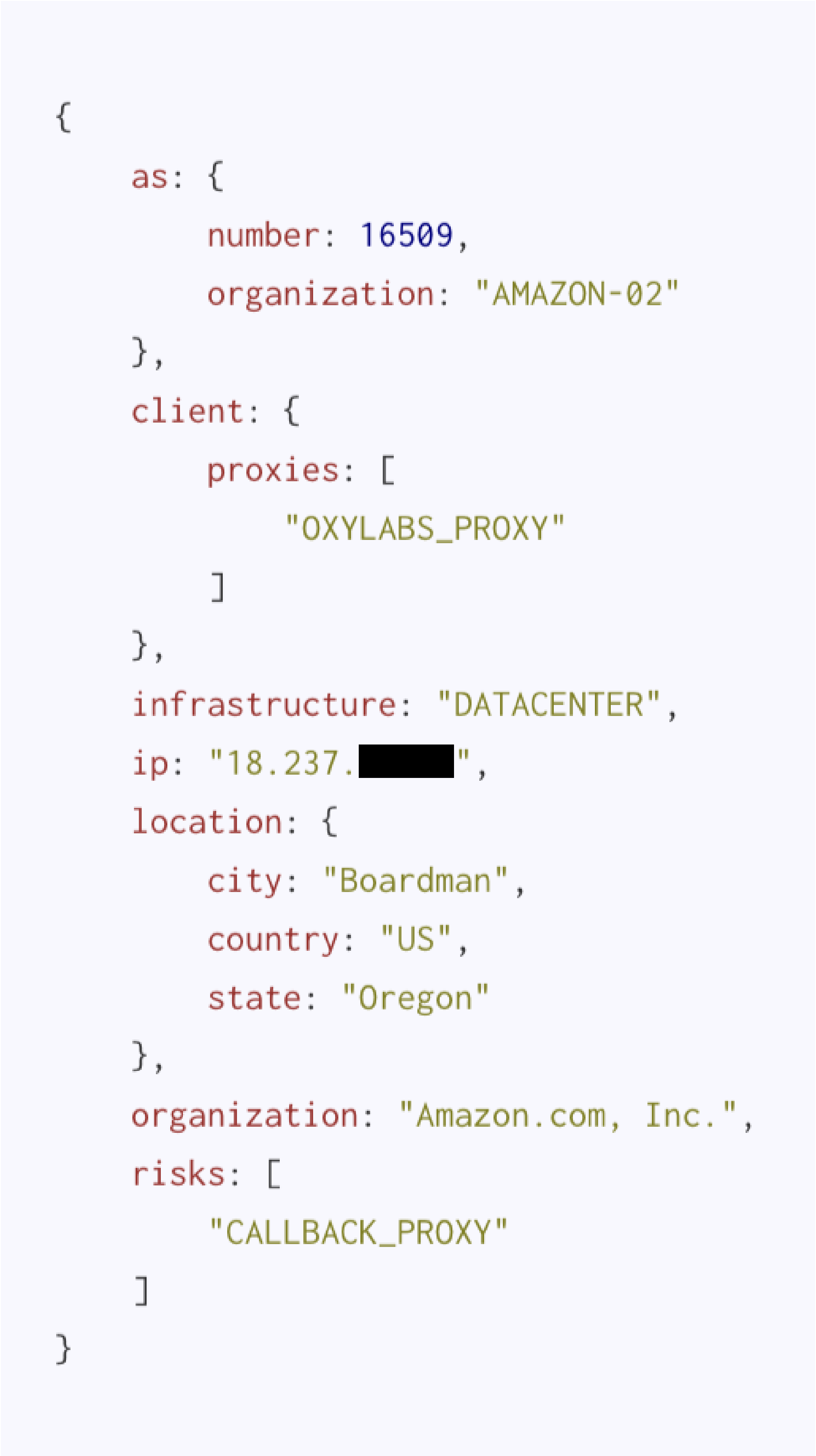
Spur Context API results for large gateway

And finally here we have the titular **large gateway** servicing at least a hundred (and probably many more) unique devices. Much like the medium-sized gateway above, deciding what to do with these can be a sticky wicket. Hopefully in this scenario you have other means of action against this user that aren’t purely IP reputation based. Thankfully, the identity of gateways of this size are often conspicuous enough to where you could discern exactly to whom or what it belongs using open source methodologies.

Non-residential gateways

When an IP belongs to commercial/consumer ISP space, taking action against them can be tricky. This is exactly the point of residential proxy services: to provide a seemingly endless pool of IP addresses so that if one is blocked, there are countless more waiting in reserve.

But normal user devices with residential proxy software installed can sometimes choose to route *their* traffic through a datacenter proxy or VPN. Additionally, residential proxy software is not strictly limited to phones and PCs, as many services also provide software for Linux servers and other network devices. And so while generally these services prefer their devices exit from residential ISP space (as per the three examples above), sometimes you’ll see exits from datacenters due to either of the aforementioned scenarios.

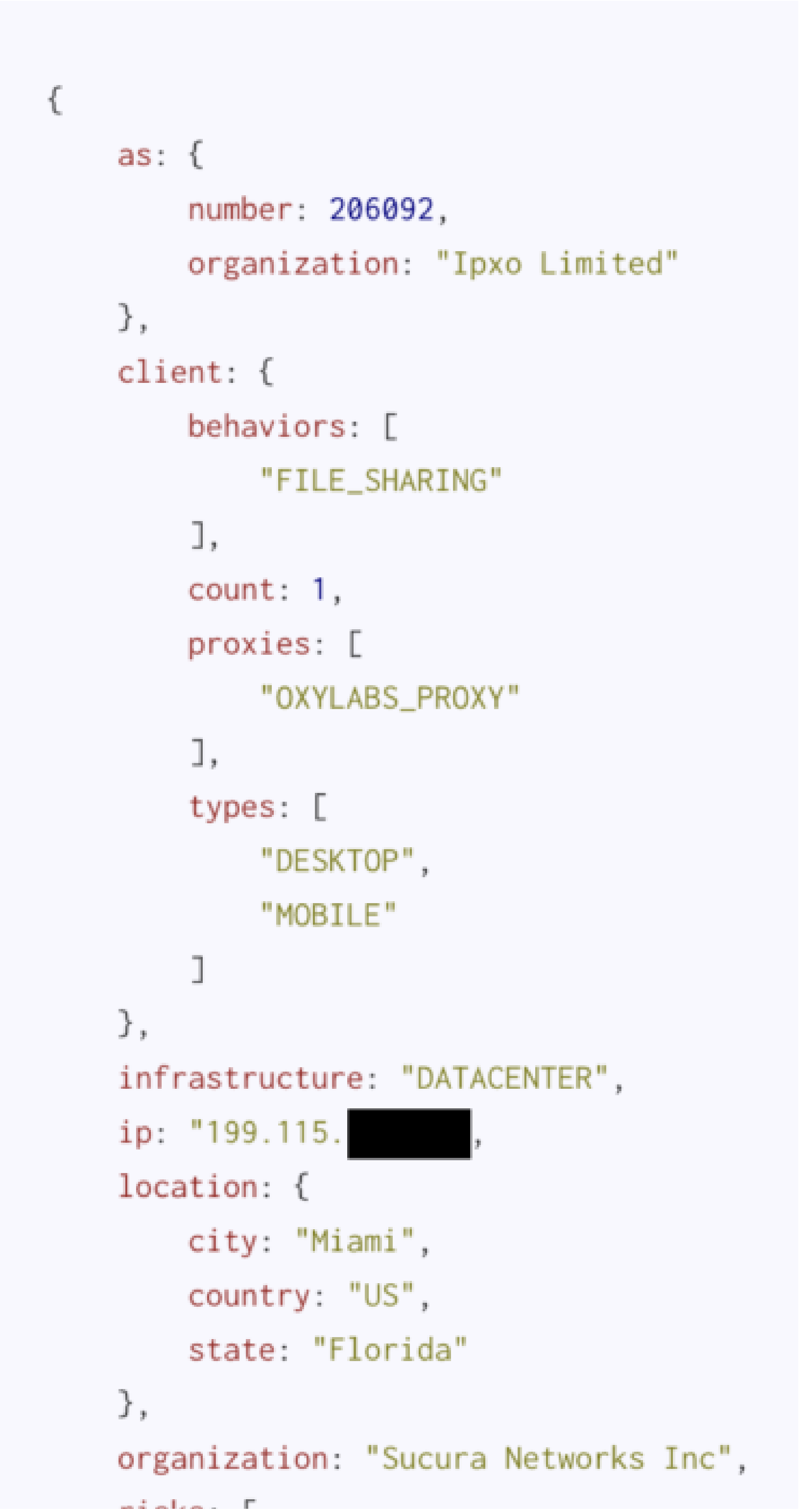


```
{
    as: {
        number: 16509,
        organization: "AMAZON-02"
    },
    client: {
        proxies: [
            "OXYLABS_PROXY"
        ]
    },
    infrastructure: "DATACENTER",
    ip: "18.237.[redacted]",
    location: {
        city: "Boardman",
        country: "US",
        state: "Oregon"
    },
    organization: "Amazon.com, Inc.",
    risks: [
        "CALLBACK_PROXY"
    ]
}
```

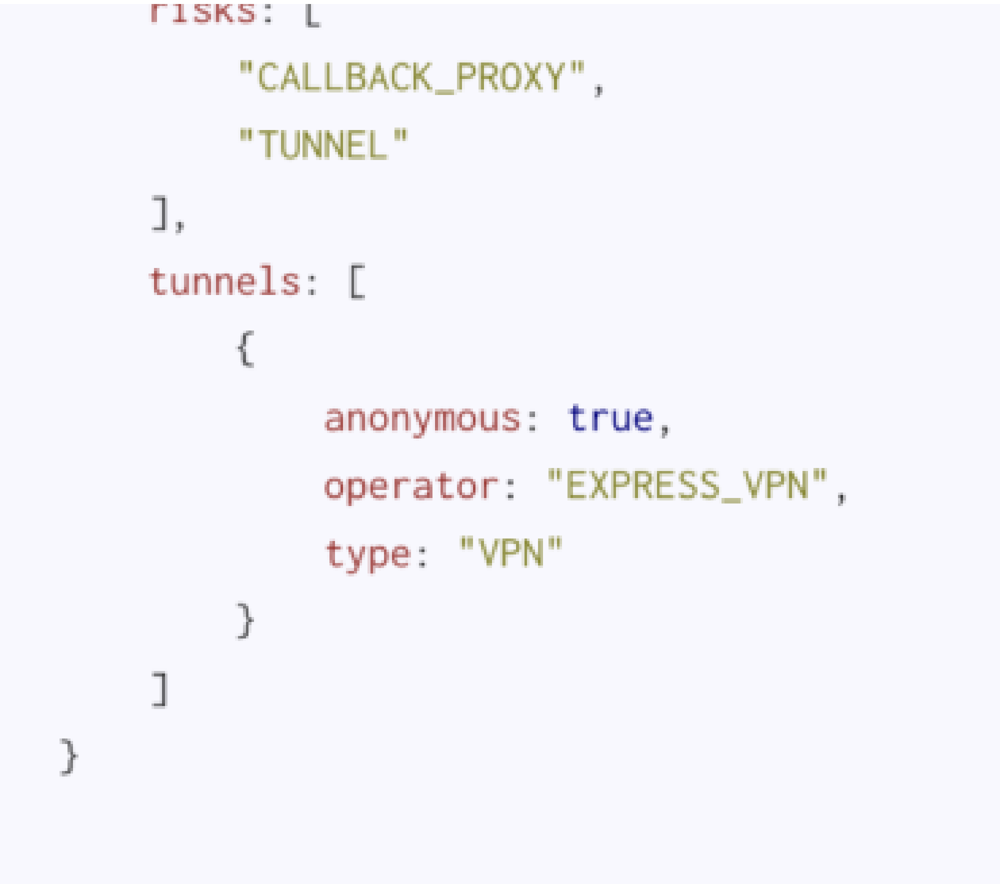
An Oxylab proxy calling out from a datacenter box belonging to Amazon

As an example, here we have a datacenter in Amazon IP space that appears to be (in addition to whatever else is does) acting as an exit node for Oxylabs. There are many scenarios that could explain this result, such as an unidentified VPN service, but regardless it tends to be *a bit* more palatable to block this IP outright. Outside of VPNs (which Spur can help identify) and a few other niche scenarios, most normal user traffic wont originate from a datacenter.

A caveat to blocking datacenter-based IP addresses is the very real likelihood that the datacenter or cloud service will eventually rotate the IP address in question to another customer, rendering the block stale. We have seen this exact scenario pop up before where a Spur Feed customer blocked a VPN exit in a datacenter, only to have the IP address be reassigned to a 3rd-party SaaS provider (of whom they were a customer) just days later, causing headaches all around. Blocklists are cumbersome and hard to maintain; agility and fresh data are key to any sort of blocklist approach.



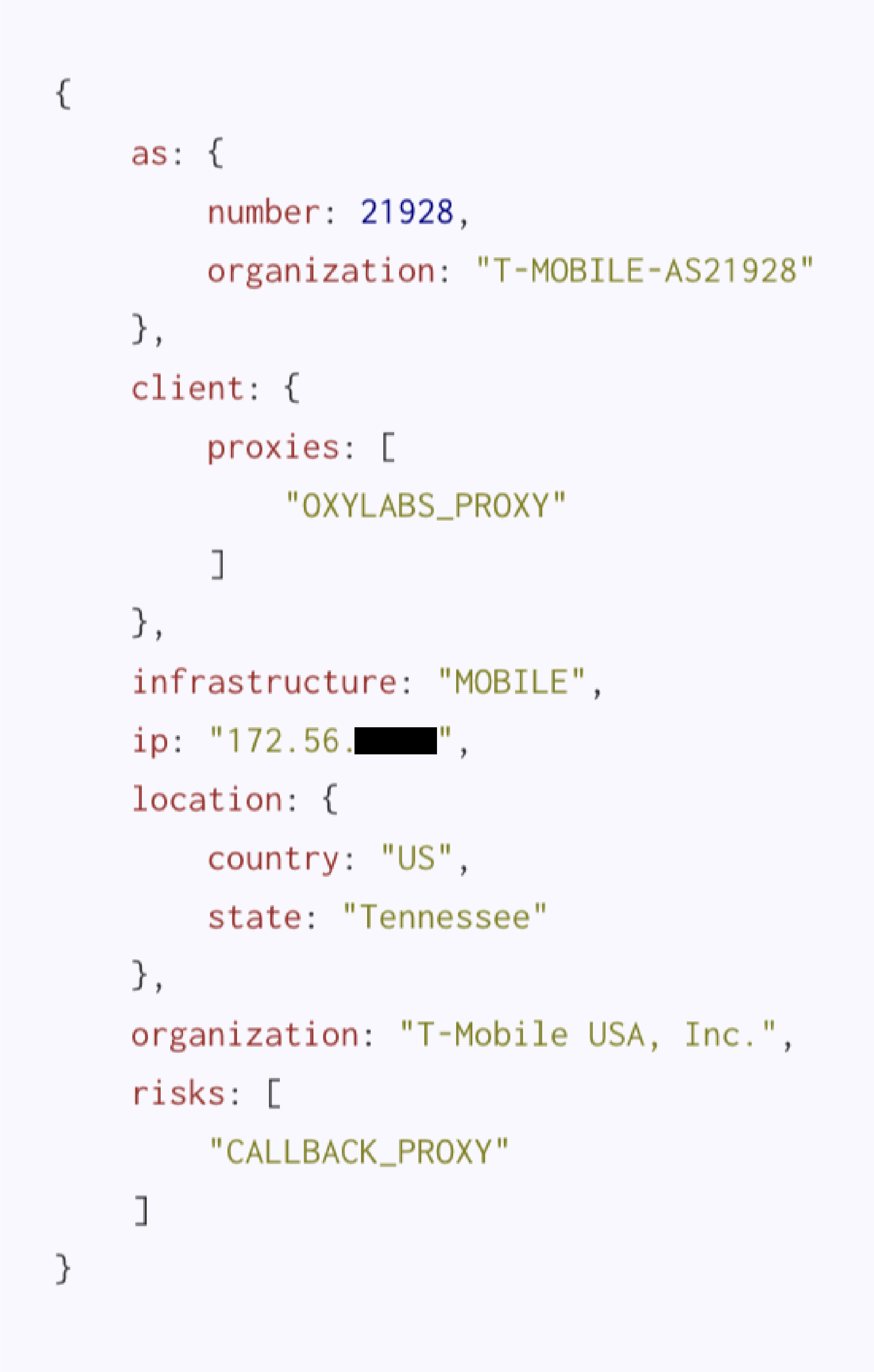
```
{
    as: {
        number: 206092,
        organization: "Ipxo Limited"
    },
    client: {
        behaviors: [
            "FILE_SHARING"
        ],
        count: 1,
        proxies: [
            "OXYLABS_PROXY"
        ],
        types: [
            "DESKTOP",
            "MOBILE"
        ]
    },
    infrastructure: "DATACENTER",
    ip: "199.115.[redacted],
    location: {
        city: "Miami",
        country: "US",
        state: "Florida"
    },
    organization: "Sucura Networks Inc",
    risks: [
```



```
    risks: [
        "CALLBACK_PROXY",
        "TUNNEL"
    ],
    tunnels: [
        {
            anonymous: true,
            operator: "EXPRESS_VPN",
            type: "VPN"
        }
    ]
}
```

An Oxylab proxy utilizing Express VPN to route its traffic

Bonus example of a residential proxy using a datacenter VPN as an exit. Sneaky! This behavior is “discouraged” by residential proxy providers for obvious reasons.



```
{
    as: {
        number: 21928,
        organization: "T-MOBILE-AS21928"
    },
    client: {
        proxies: [
            "OXYLABS_PROXY"
        ]
    },
    infrastructure: "MOBILE",
    ip: "172.56.[redacted]",
    location: {
        country: "US",
        state: "Tennessee"
    },
    organization: "T-Mobile USA, Inc.",
    risks: [
        "CALLBACK_PROXY"
    ]
}
```

Lastly, we have a mobile gateway. This IP is registered with T-Mobile and is used to route Internet traffic for mobile devices. These gateways are probably the hardest to justify a block against as they serve traffic for an enormous volume of (mostly transient) daily users. More and more proxy services are popping up or expanding their offerings to include “4G proxies” to capitalize on this fact. Oftentimes the device hosting the proxy software behind these gateways will be Raspberry Pis with 4G dongles.

Let’s talk about context

When everything looks like a nail, IP reputation is giant hammer.

There are many free and paid IP reputation lists floating around out there. This data can be very useful when performing retrospective analysis, but it often lacks critical context to make decisions in real-time. We used our daily feed data to enrich the CINS Army reputation list and found that, of the 15,000 total IP addresses present:

- 113 are medium or greater size gateways
- 600 belong to VPN services
- 4,541 of them are host to malware proxy services
- 363 are mobile gateways

Putting your fraud and abuse data into context is crucial; an overview of events juxtaposed with the freshest data possible is required to make good business decisions, as opposed to focusing on singular events and stale go/no-go blocklists.

For instance, if for a given incident, 90% of the abusive traffic is originating from the same tag (such as Oxylabs as above, maybe), that might be enough to identify the specific proxy service being leveraged by a campaign or actor. From there you can introduce more granular alerts or escalations based specifically on those high-risk services.

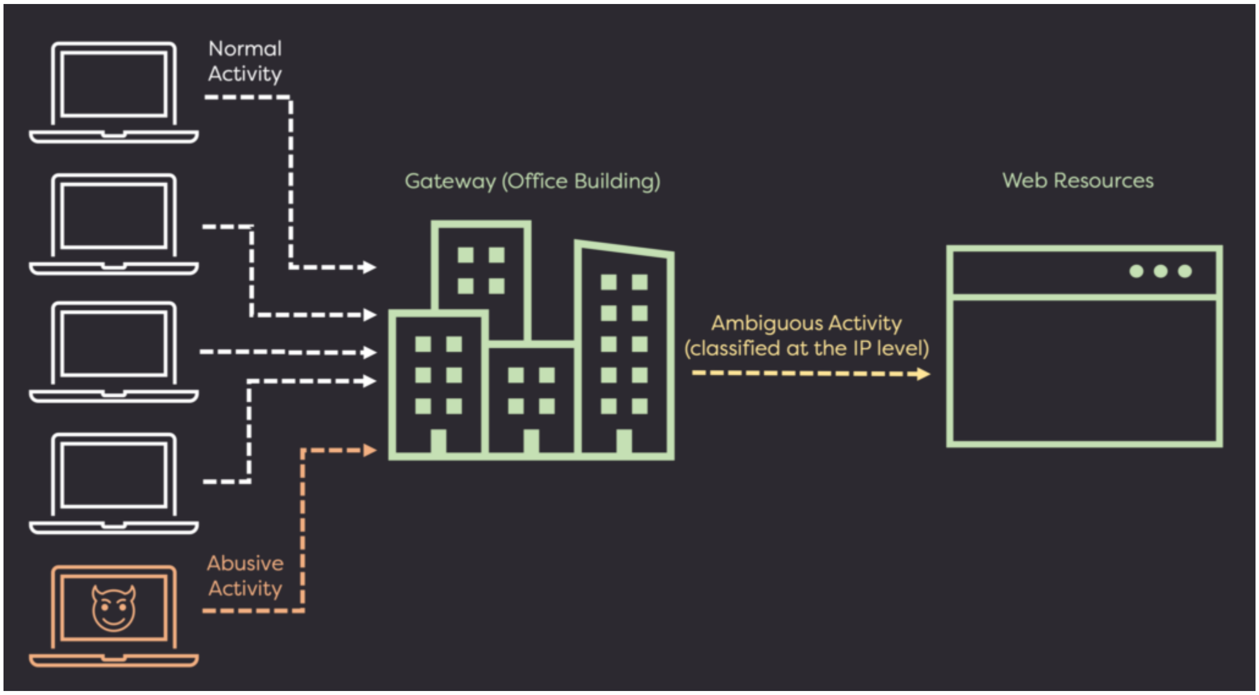
Ultimately, IP-based insights can only take you so far. In an ideal world, you’d have better insight into the nature of connections at a session-level, allowing you to flag proxied vs non-proxied traffic coming from the same IP. In this way, you’d know if that dubious traffic from you see coming from a public library is from someone sitting in the physical location, or from a residential proxy customer potentially operating from another continent.

Introducing: Monocle

Instead of condemning all the users behind a gateway for the actions of a few, wouldn’t it be better to attribute malicious activity at the session level instead of the network level? One bad apple doesn’t necessarily have to spoil the bunch.

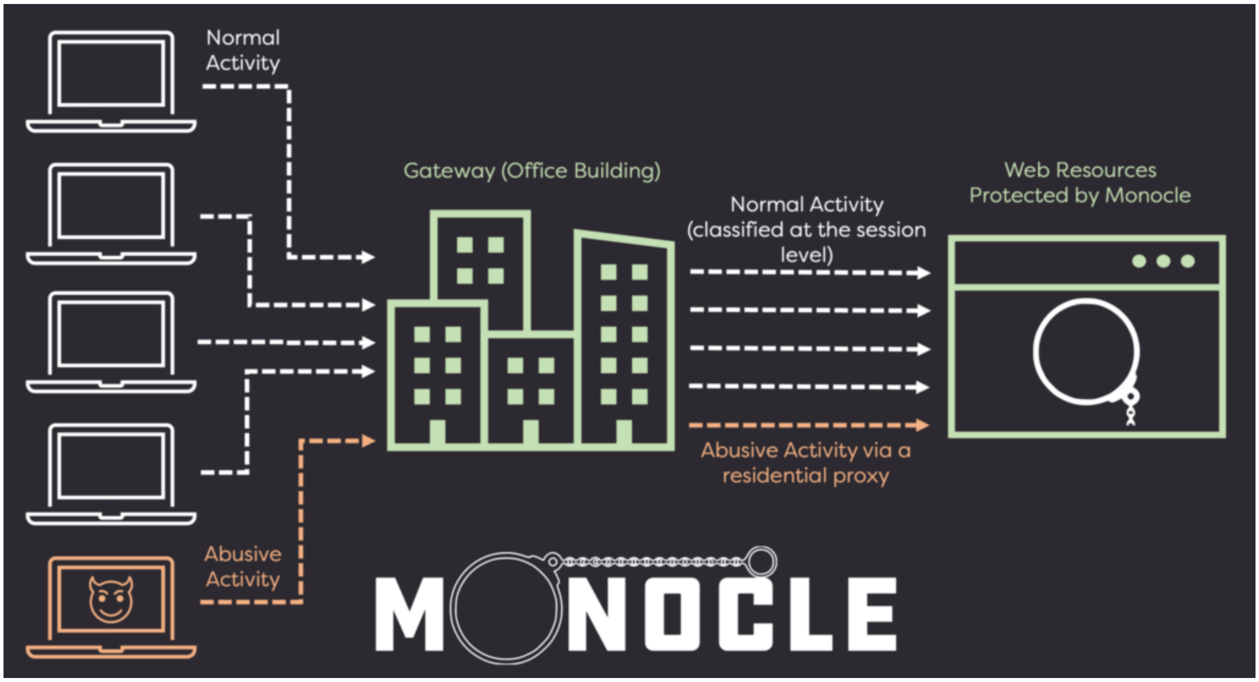
Spur’s newest offering is a lightweight, “captcha”-like JavaScript library called **Monocle** that aims to empower web applications with realtime data on the nature of connected sessions at a **device level**, not a gateway level.

Intended to work independently or alongside existing captcha services such as hCaptcha, reCAPTCHA, or the newly-announced Turnstile, Monocle can detect sessions from classic anonymizing infrastructure (such as Tor, VPNs, and Datacenter Proxies), and even from the growing threat of residential proxies.


Normal Activity
Gateway (Office Building)
Web Resources
Ambiguous Activity (classified at the IP level)
Abusive Activity

IP reputation's view of a large gateway

Where IP reputation gives murky insight into connections from large gateways, Monocle can give realtime clarity to whether or not action should be taken against a form submission or session.


Normal Activity
Gateway (Office Building)
Web Resources Protected by Monocle
Normal Activity (classified at the session level)
Abusive Activity via a residential proxy
Abusive Activity
MONOCLE

Monocle's view of a large gateway, or more specifically, the sessions from the devices behind the large gateway

Captcha services are great at targeting bot-like user behavior and device telemetry, but they fail to give insight into the nature of the session from an infrastructure level. A malicious user can pass a Captcha (or farm it off to a Captcha solving service) but still do your service harm while operating from a residential proxy or VPN. Monocle can give you that critical missing context.

Check out the Monocle product page to learn more about how Monocle can support your business in the fight against fraud and abuse!