How Credential Stuffing is Evolving

And where do we go from here?

Jarrod Overson
Director of Engineering at Shape Security

CREDENTIAL STUFFING

cre·den·tial stuff·ing

/krəˈden(t)SHəl ˈstəfiNG/

The testing of previously breached username and password pairs across sites to find accounts where passwords have been reused.

Photo by Nine Köpfer on Unsplash

Who am I?

And should you trust me?

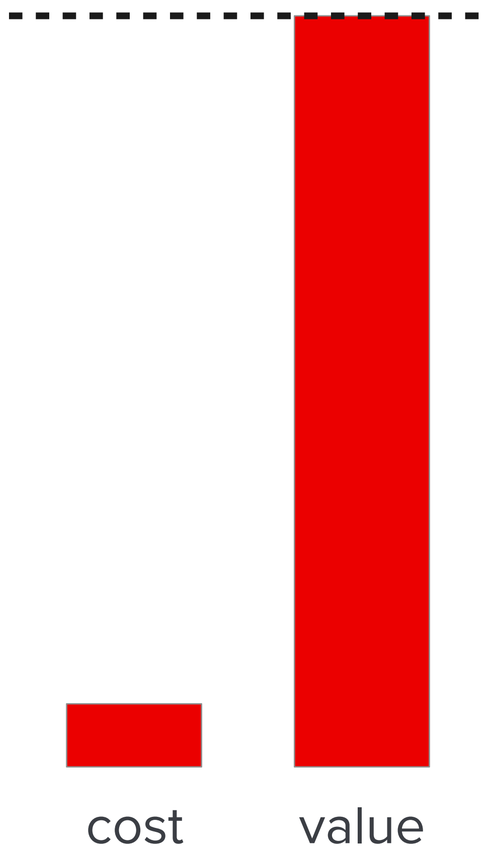
- Director of Engineering at Shape Security
- Google Developer Expert.
- Old school video game hacker.
- @jsoverson everywhere

1. **Why credential stuffing is evolving**
2. How credential stuffing has evolved
3. Where do we go from here?

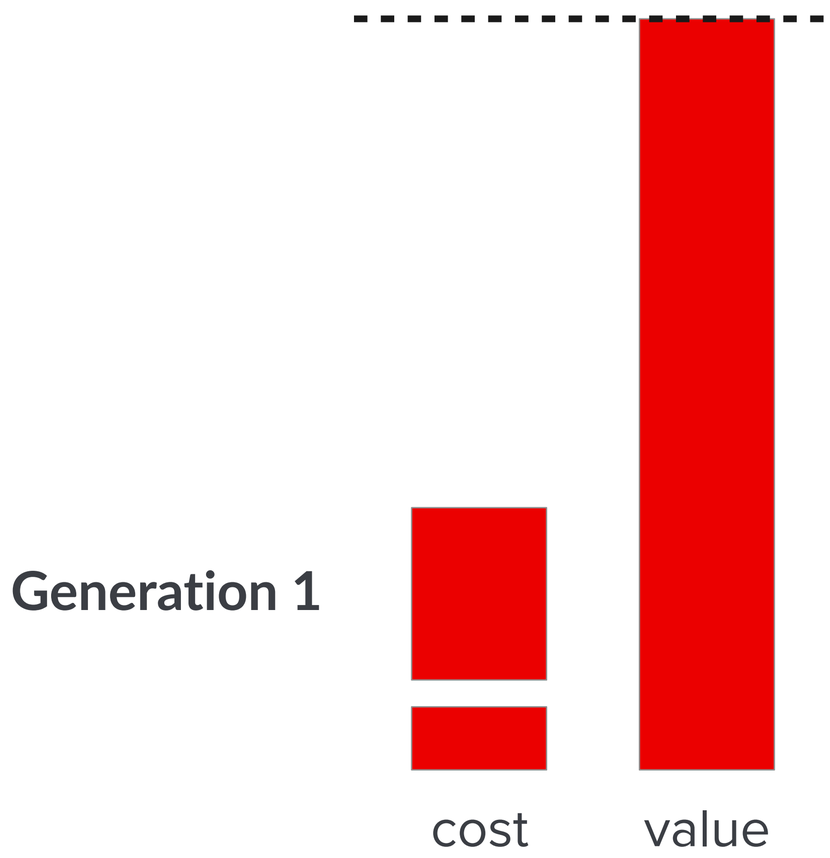
1 Why credential stuffing is evolving

The same reason anything evolves. Incentive + adversity.

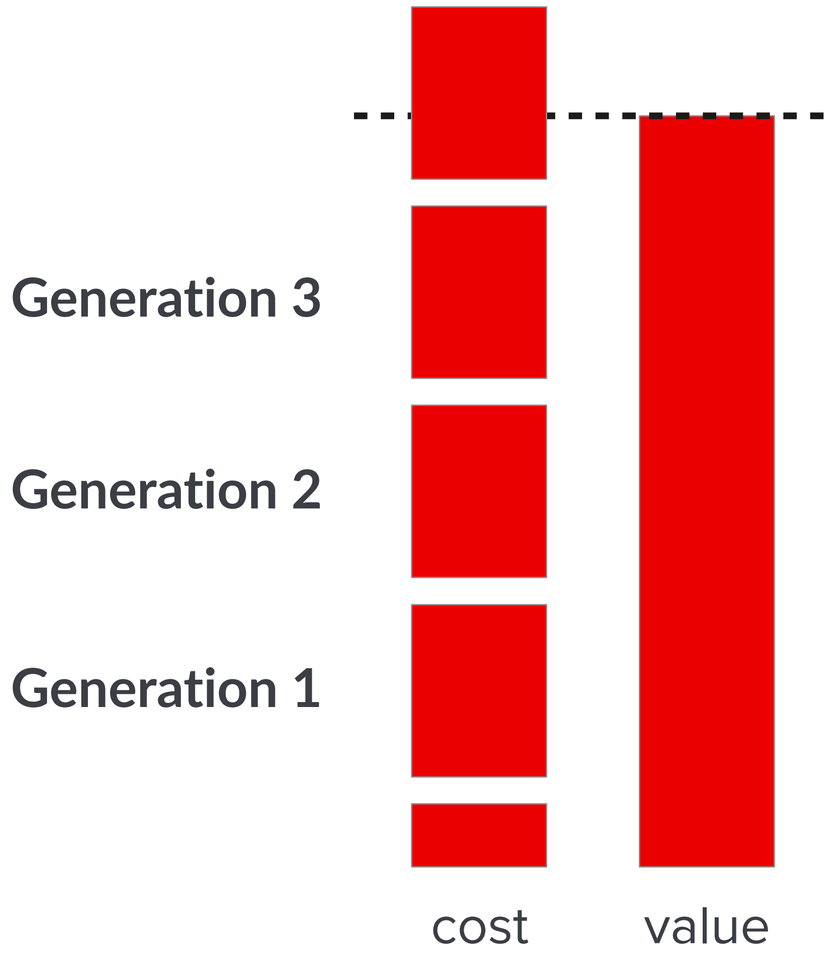
If there are no defenses in place, the cost is nearly zero.



Any defense increases the cost by forcing a generational shift.

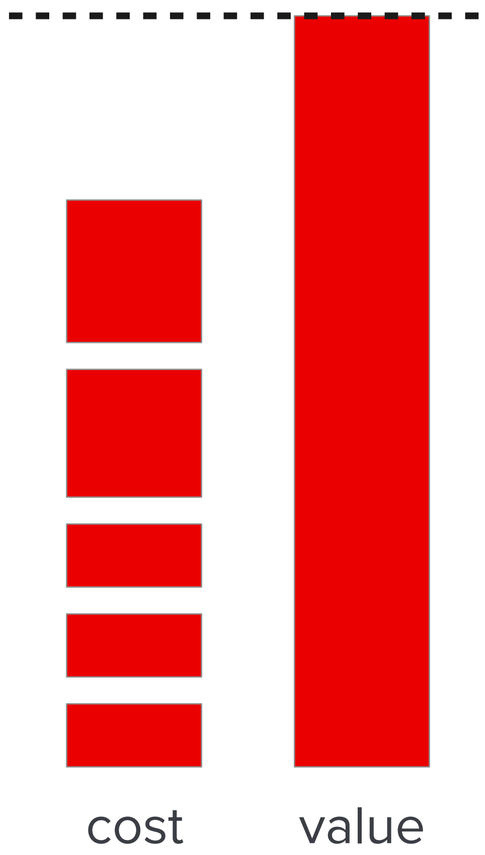


Enough defenses will tip cost/value in your favor

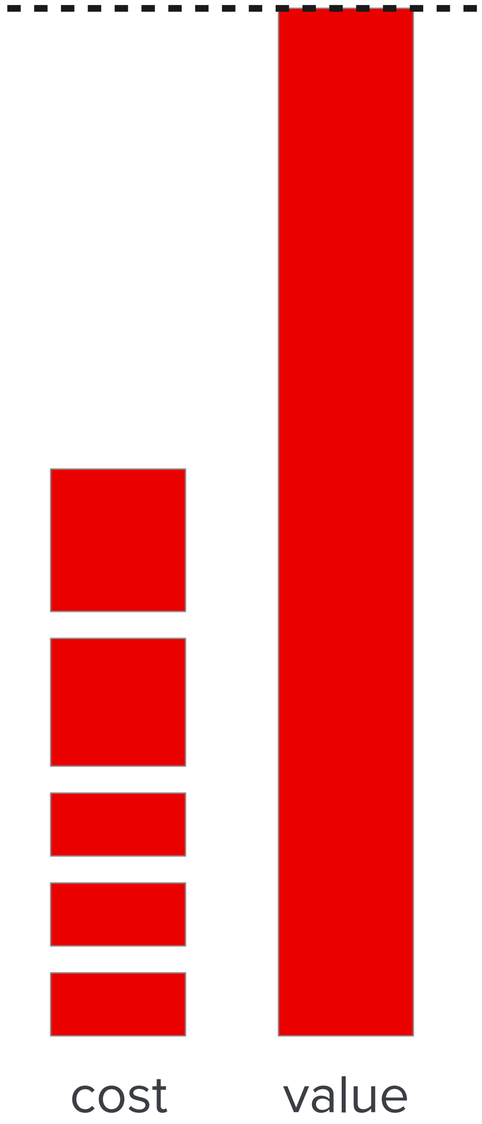


The cost of entry for all technology decreases over time.

All technology gets cheaper as it becomes better understood and more generalized.



While the value of successful attacks only goes up.



MANUAL WORK

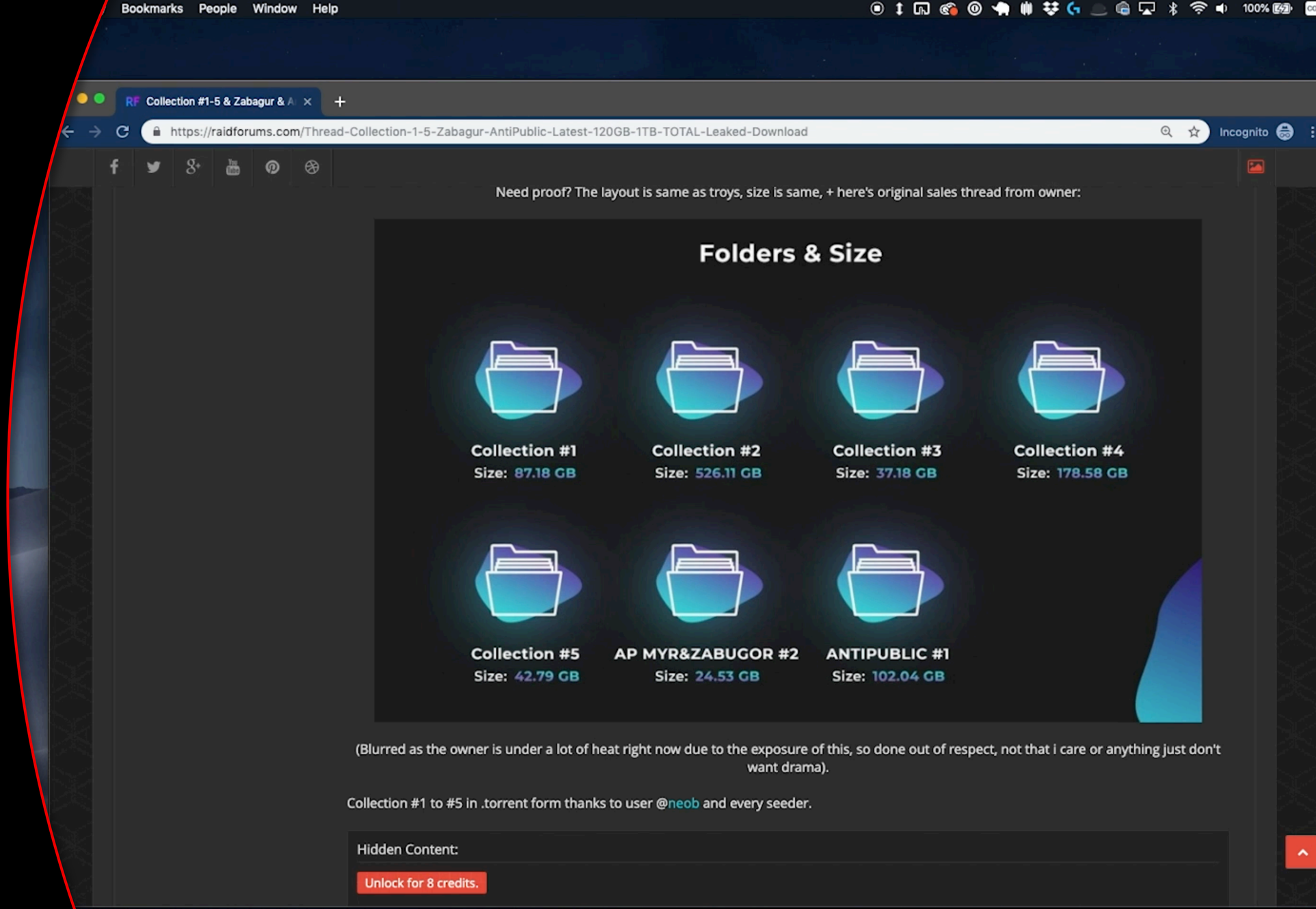
AUTOMATION

CREDENTIAL STUFFING: A HOW-TO GUIDE

1. Get Credentials
2. Automate Login
3. Defeat Existing Defenses
4. Distribute Globally

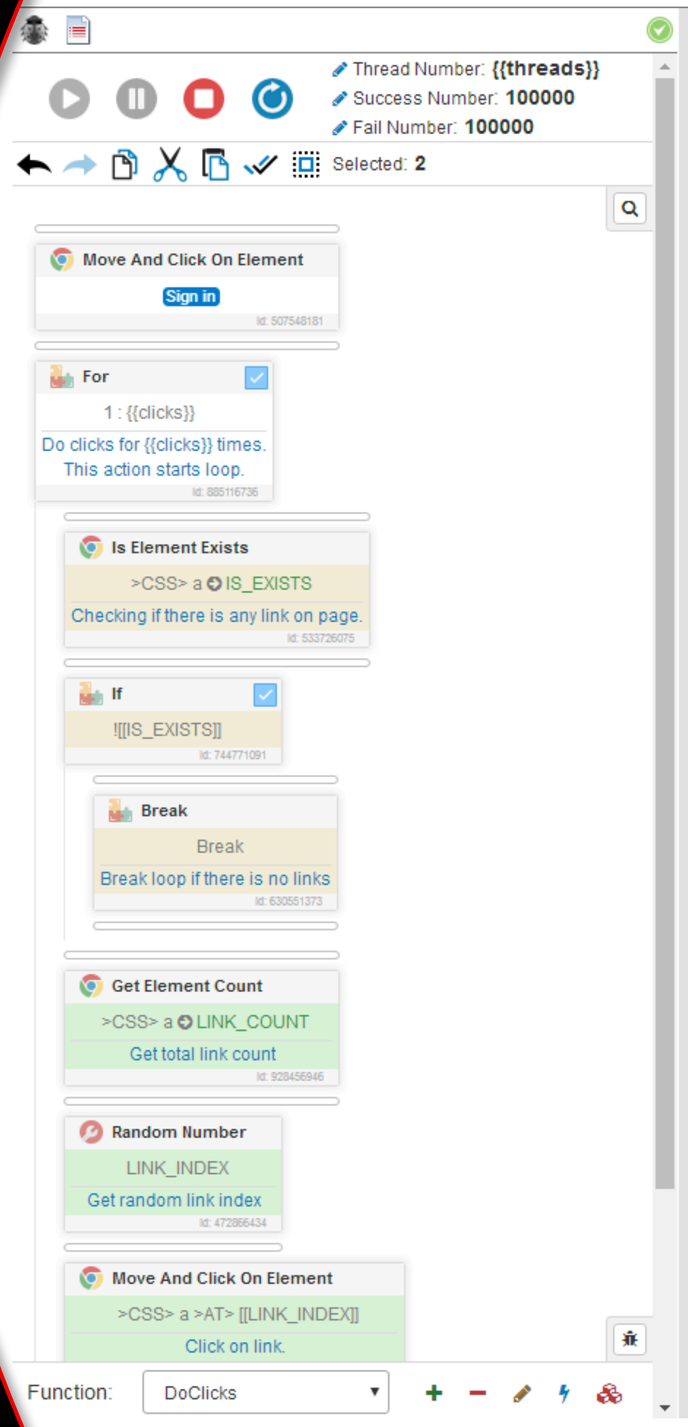
CREDENTIAL STUFFING

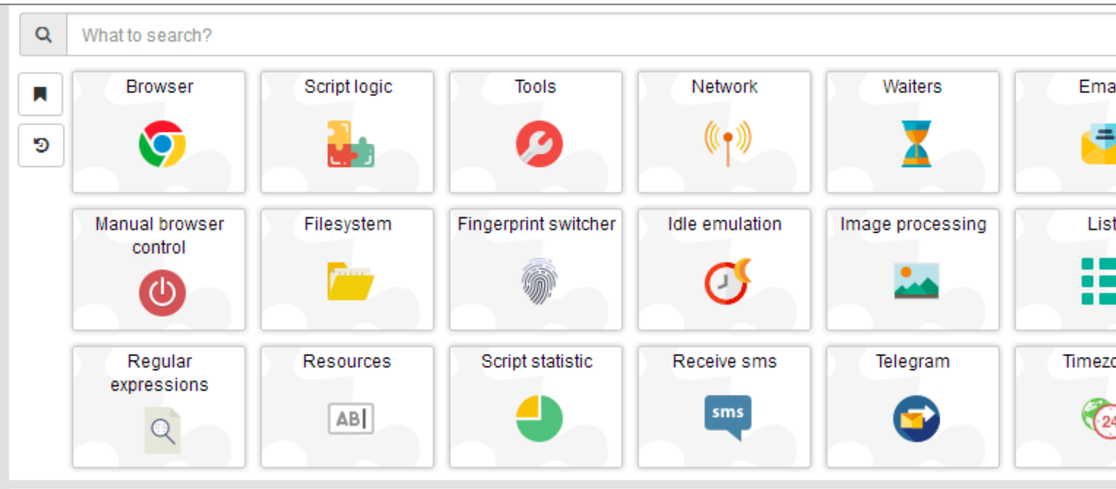
1. Get Credentials

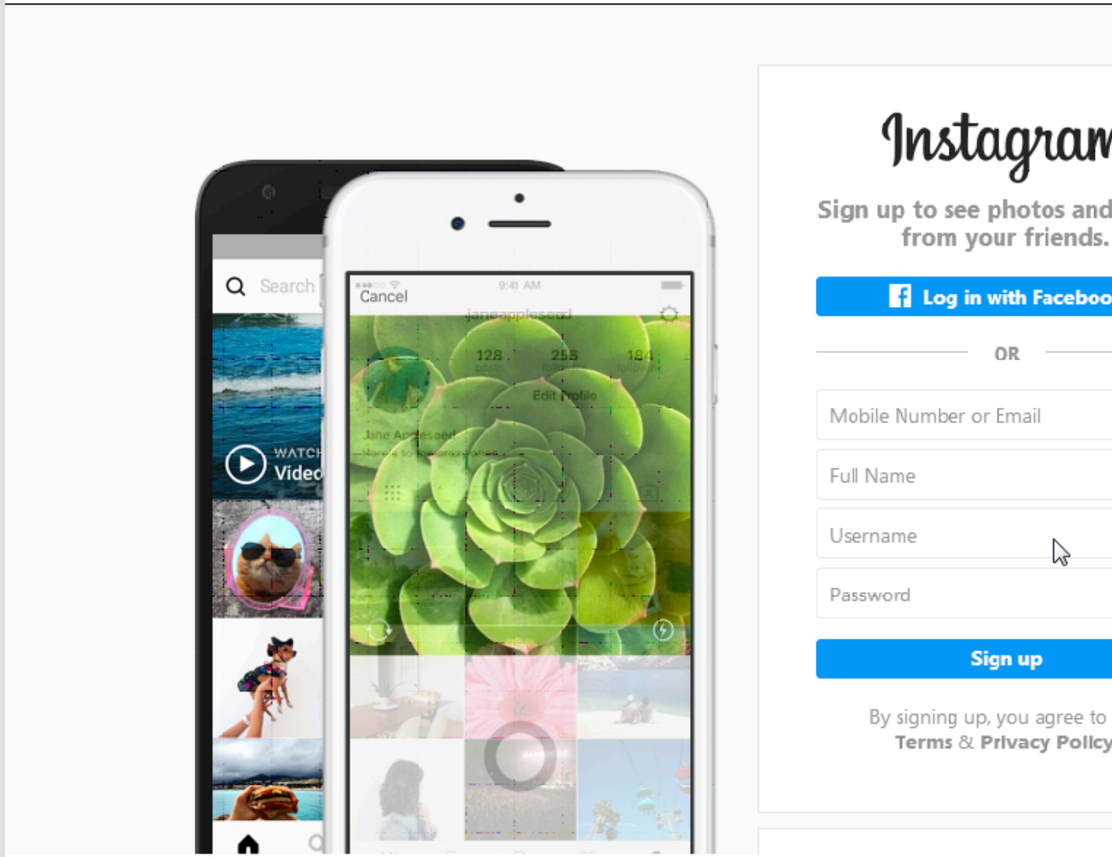


CREDENTIAL STUFFING

1. Get Credentials
2. Automate Login

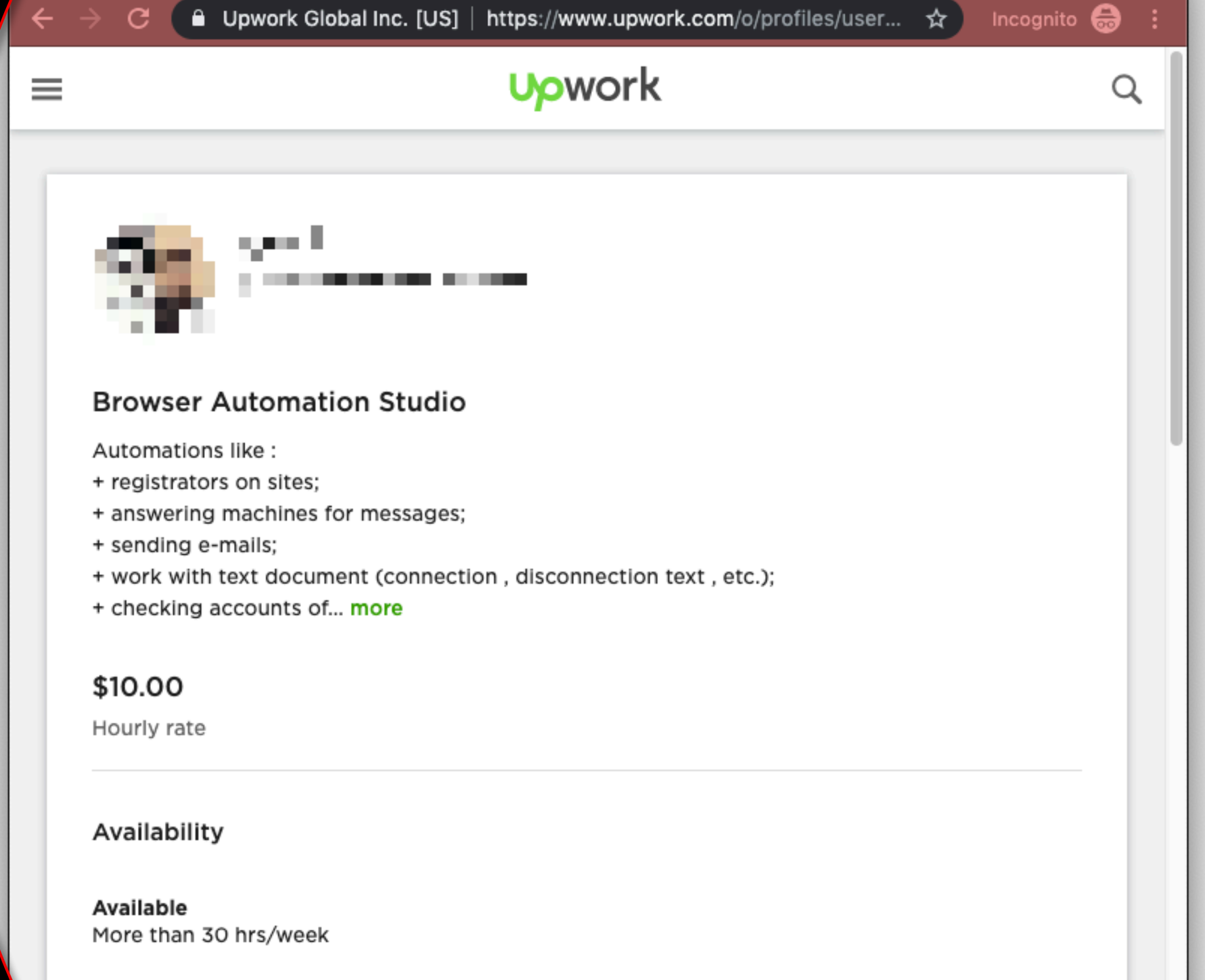






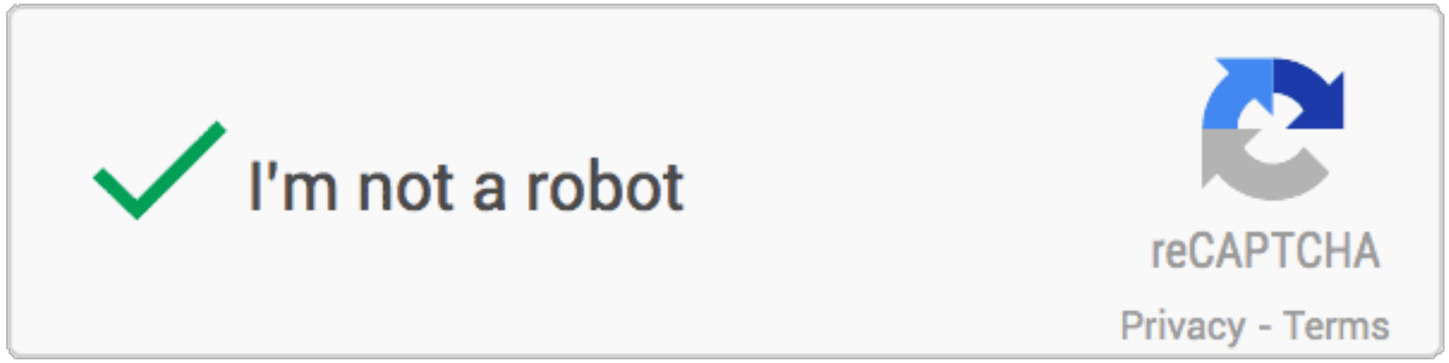
CREDENTIAL STUFFING

1. Get Credentials
2. Automate Login



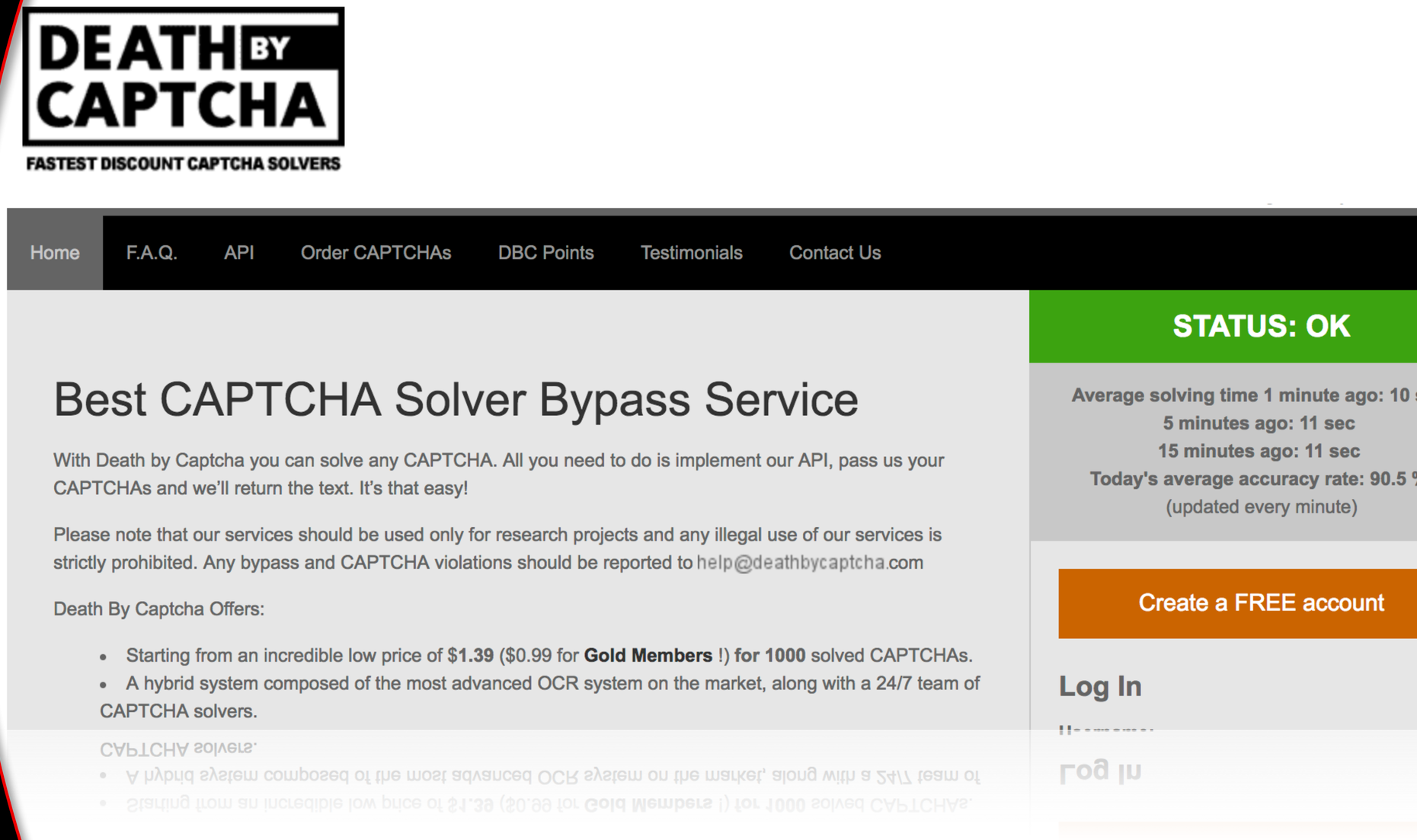
CREDENTIAL STUFFING

1. Get Credentials
2. Automate Login
3. Defeat Defenses

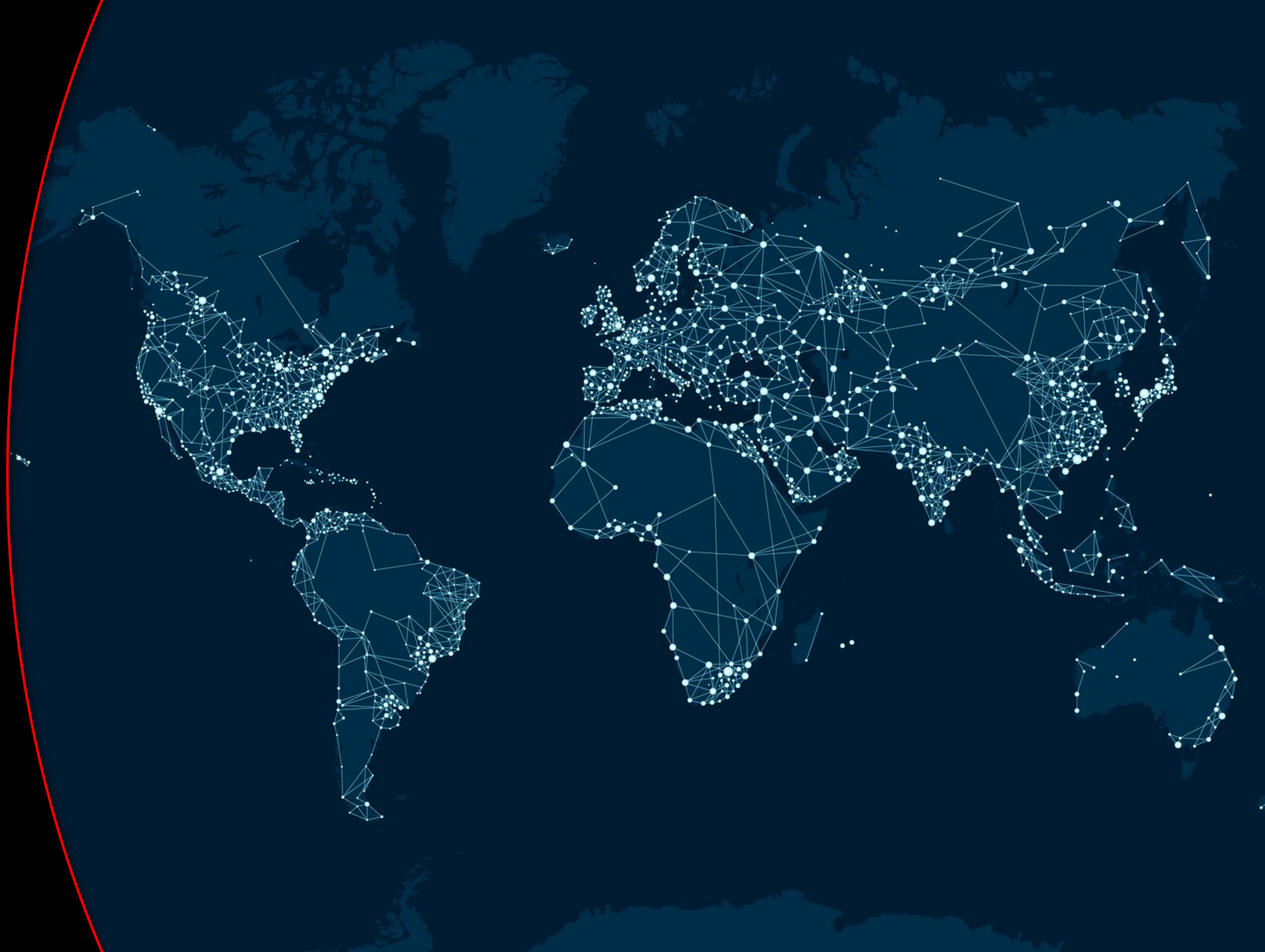


CREDENTIAL STUFFING

1. Get Credentials
2. Automate Login
3. Defeat Defenses



CREDENTIAL STUFFING



1. Get Credentials
2. Automate Login
3. Defeat Defenses
4. Distribute

100,000 ATO attempts can be tried for less than $200 USD

$0

2.3 billion credentials

$0-50

For tool configuration

$0-139

For 100,000 solved CAPTCHAs

$0-10

For 1,000 global IPs

<$0.002

per ATO attempt.

The rate of return is between 100% and 150,000%+

$2 - $150+

Typical range of account values.

0.2% - 2%

Success rate of a typical credential stuffing attack.

$0.002

Cost per individual attempt.

$$\frac{\text{Value} * \text{Success Rate}}{\text{Cost}} - 100\% = \text{Rate of Return}$$

1. Why credential stuffing is evolving
2. **How credential stuffing has evolved**
3. Where do we go from here?

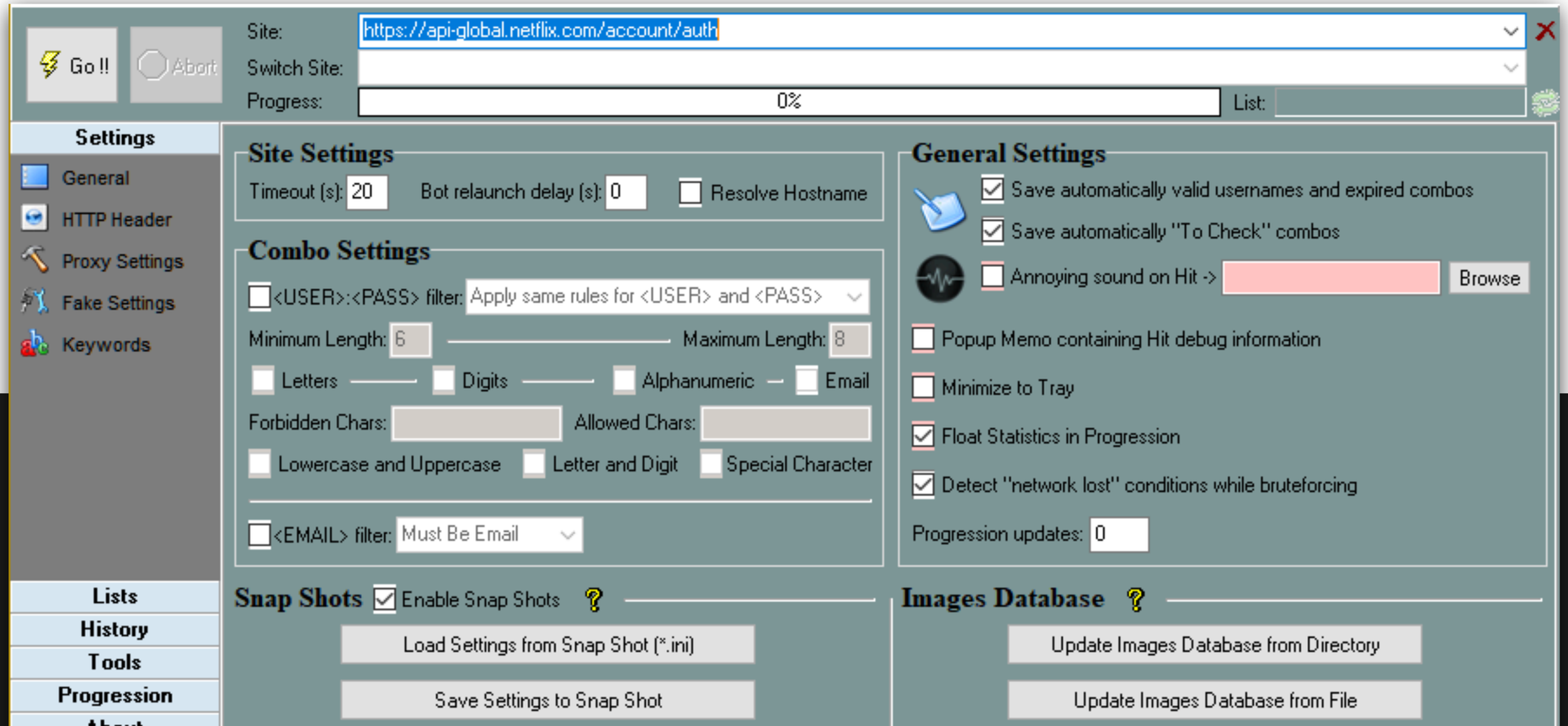
Before Modern Era

Generation 0: Basic HTTP requests with common tools

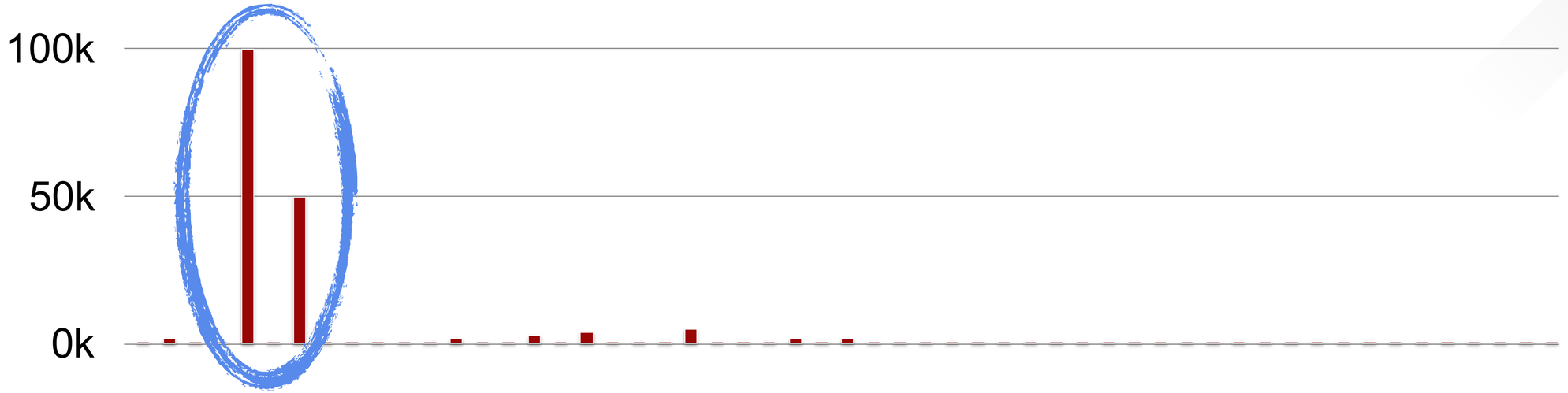
```
$
```

SentryMBA

- Performs basic HTTP requests.
- Extensible and highly configurable.
- Tailored towards specific attack use cases.



Early defense: IP Rate limiting.



Free Proxy List

FREE PROXY | WEB PROXY | SOCKS PROXY | BUY PROXY | COMPANY

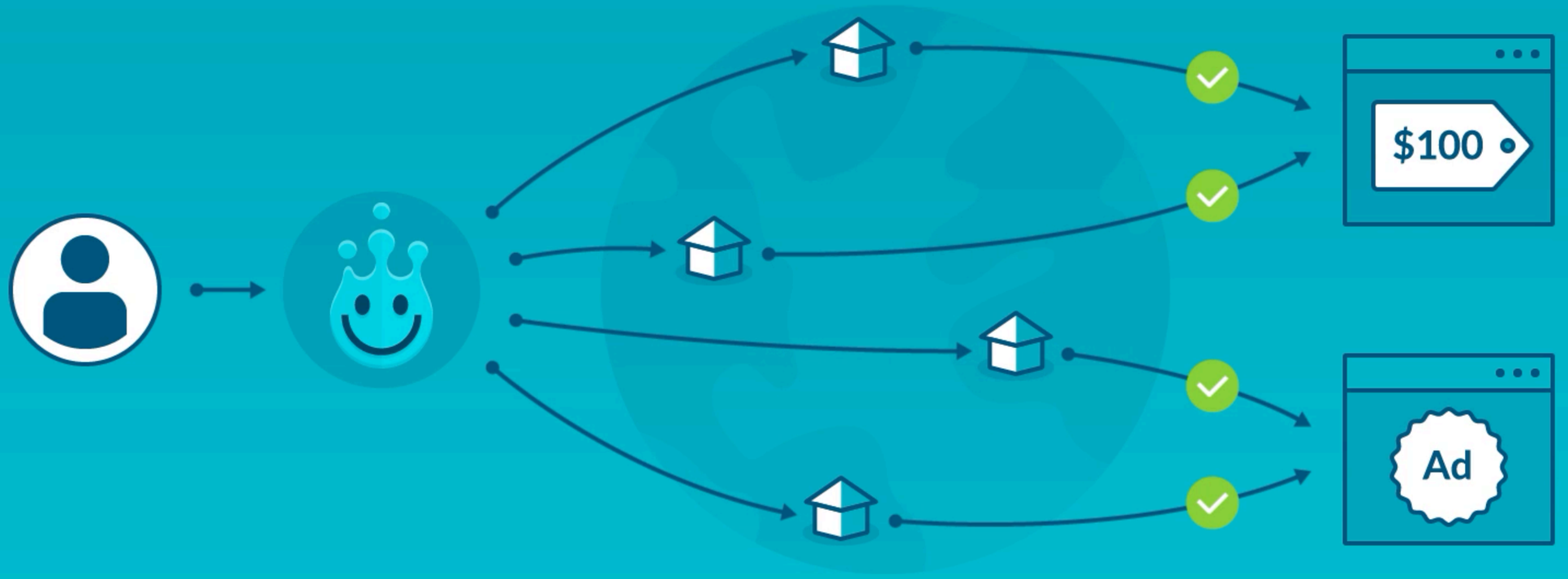
Show 20 entries

Search all columns:

IP Address	Port	Code	Anonymity	Https
185.122.44.218	36805	IT	elite proxy	yes
41.39.125.250	23500	EG	elite proxy	yes
197.211.245.50	53281	ZW	elite proxy	yes

Iteration 1 : Rotate through proxies

Luminati — Residential proxy network

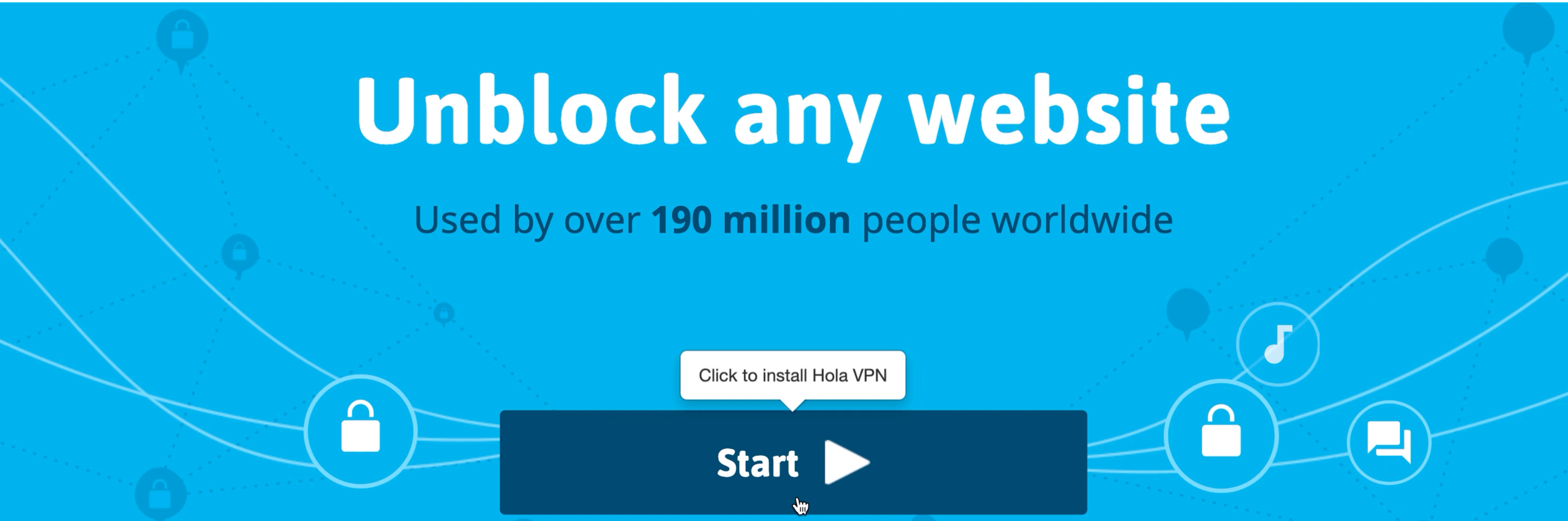


Every country and every city in the world

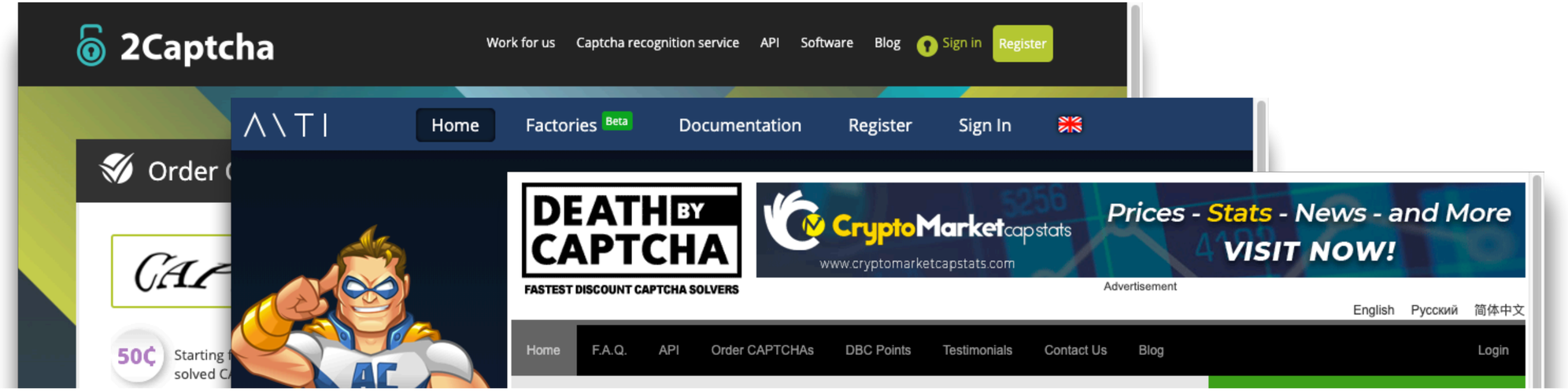
hola! VPN

Unblock any website

Used by over **190 million** people worldwide

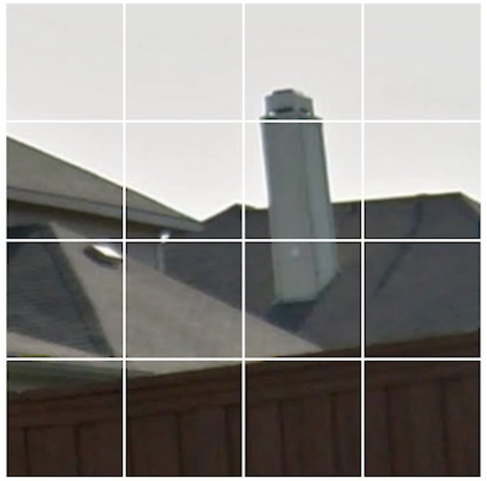


Defense: Text-based CAPTCHAs



Iteration 2: Use CAPTCHA Solvers.

Defense: Dynamic sites and JavaScript heavy defenses.



PhantomJS

Full web stack
No browser required

trifleJS

Headless automation for Internet Explorer

Iteration 3: Scriptable WebViews

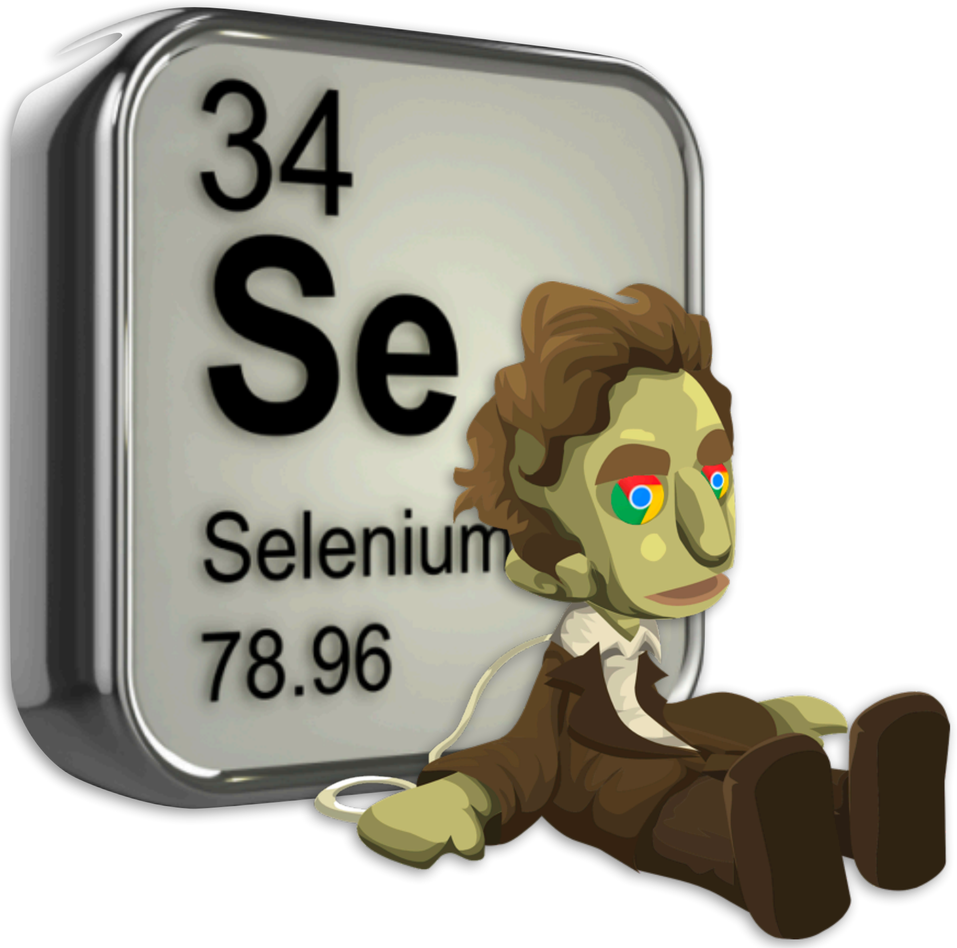
Defense: Header Fingerprinting & Environment Checks

```
GET / HTTP/1.1
Host: localhost:1337
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/
*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.
(KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
```

```
GET / HTTP/1.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X) AppleWebKit/534.34 (KHT
like Gecko) PhantomJS/1.9.8 Safari/534.34
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Connection: Keep-Alive
Accept-Encoding: gzip
Accept-Language: en-US,*
Host: localhost:1337
```

Modern Era

Iteration 4: Scriptable Consumer Browsers



Selenium & Puppeteer

Selenium is a free, open source testing tool that scripts popular browsers.

Puppeteer is a Google project that automates Firefox and Chromium based browsers.

Defense: Browser Fingerprinting

Browser Fingerprinting

High-entropy data points are collected to produce an acceptably unique fingerprint.

Data points like screen size, fonts, plugins, hardware profiles, et al.

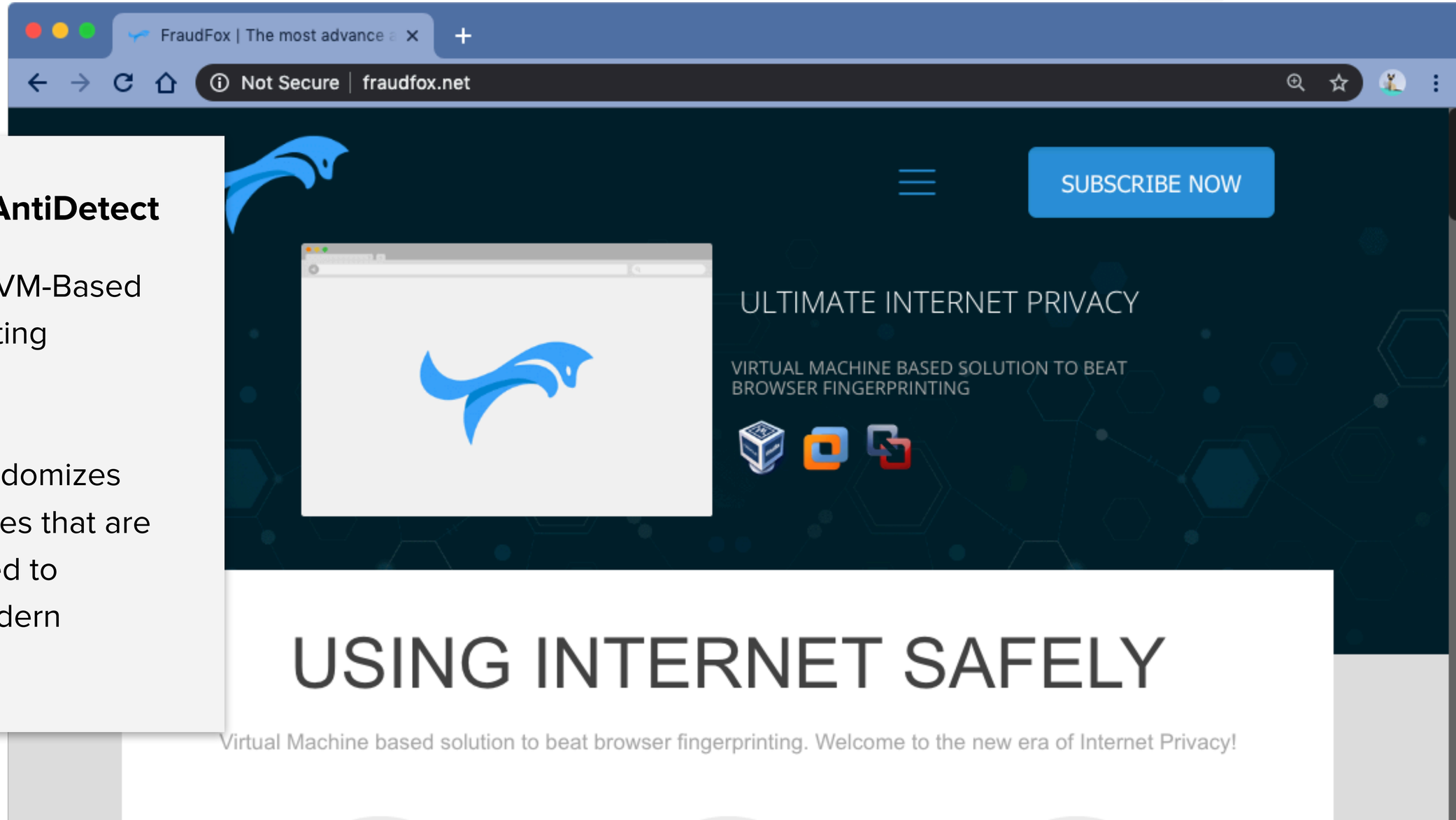
This identifies the source of traffic even when tunneling through proxies.

Iteration 5: Randomizing Fingerprint Data Sources

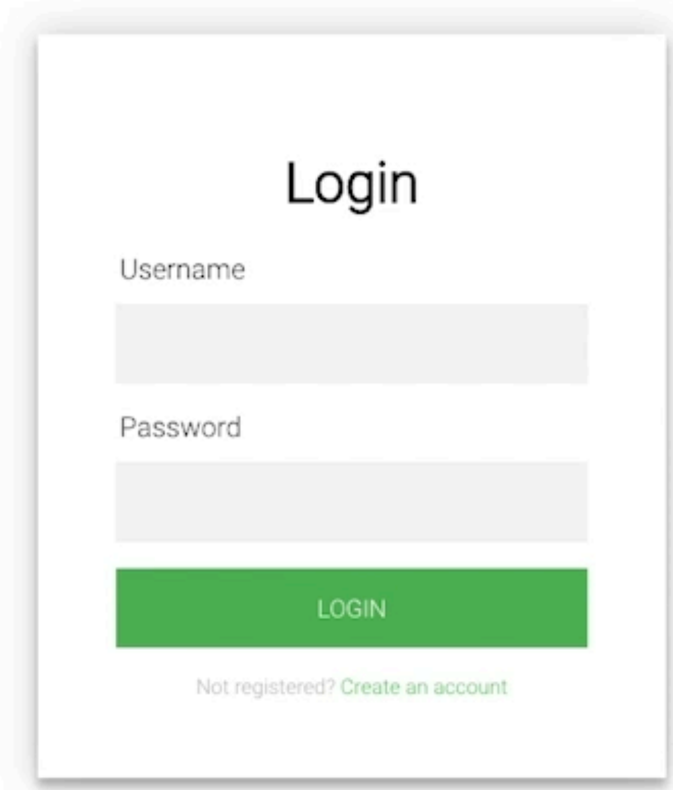
FraudFox & AntiDetect

FraudFox is a VM-Based anti-fingerprinting solution.

AntiDetect randomizes the data sources that are commonly used to fingerprint modern browsers.



Defense: Behavior Analysis for Negative Traits



Behavior Analysis

Naive bots give themselves away by ignoring normal human behavior.

Humans don't always click in the upper left hand corner and don't type out words all at once.

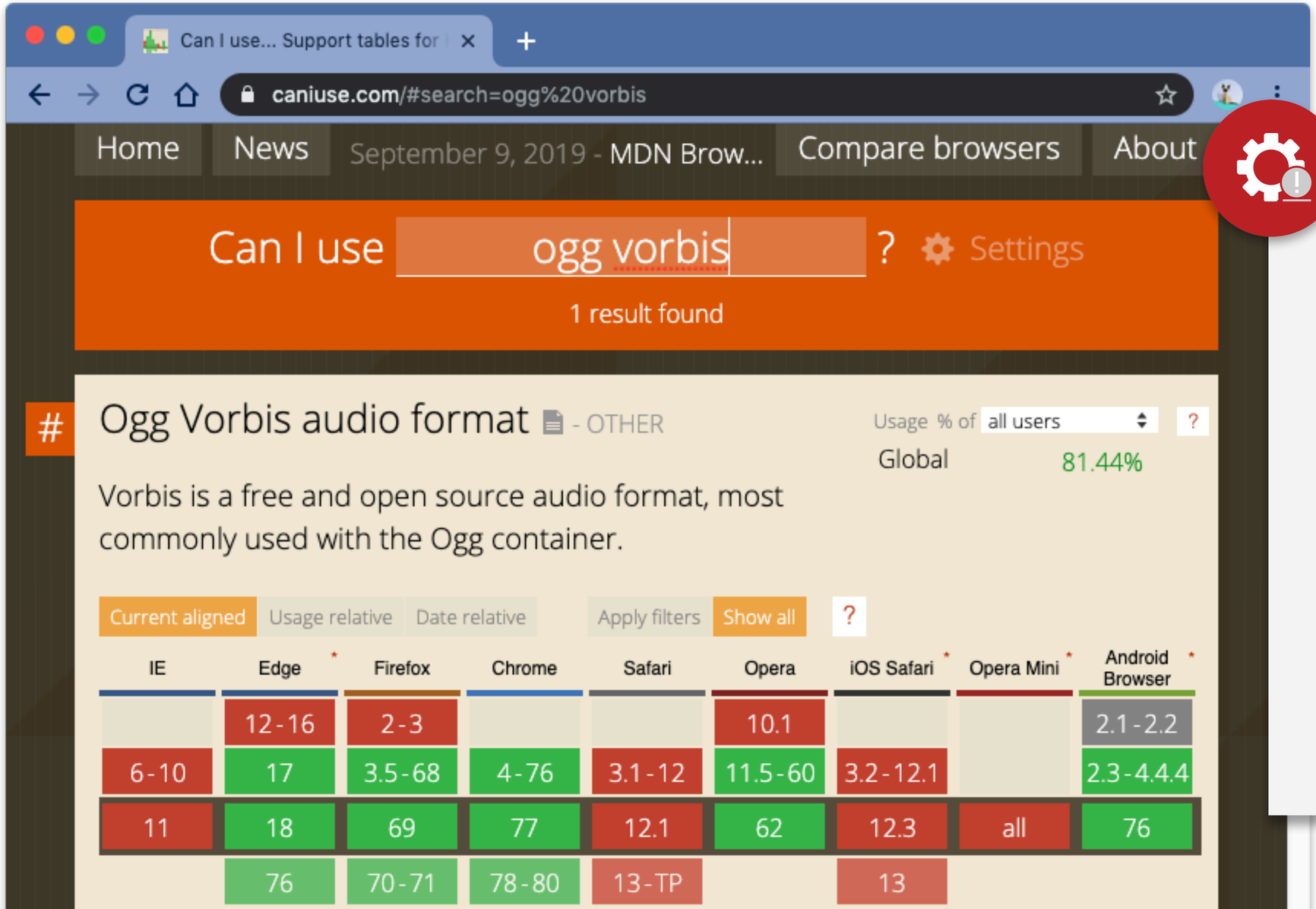
Capturing basic behavior can make naive automation easy to knock down.

Iteration 6: Human Behavior Emulation

Browser Automation Studio

BAS is an automation tool that combines CAPTCHA solving, proxy rotation, and emulated human behavior.

Defense: Browser Consistency Checks



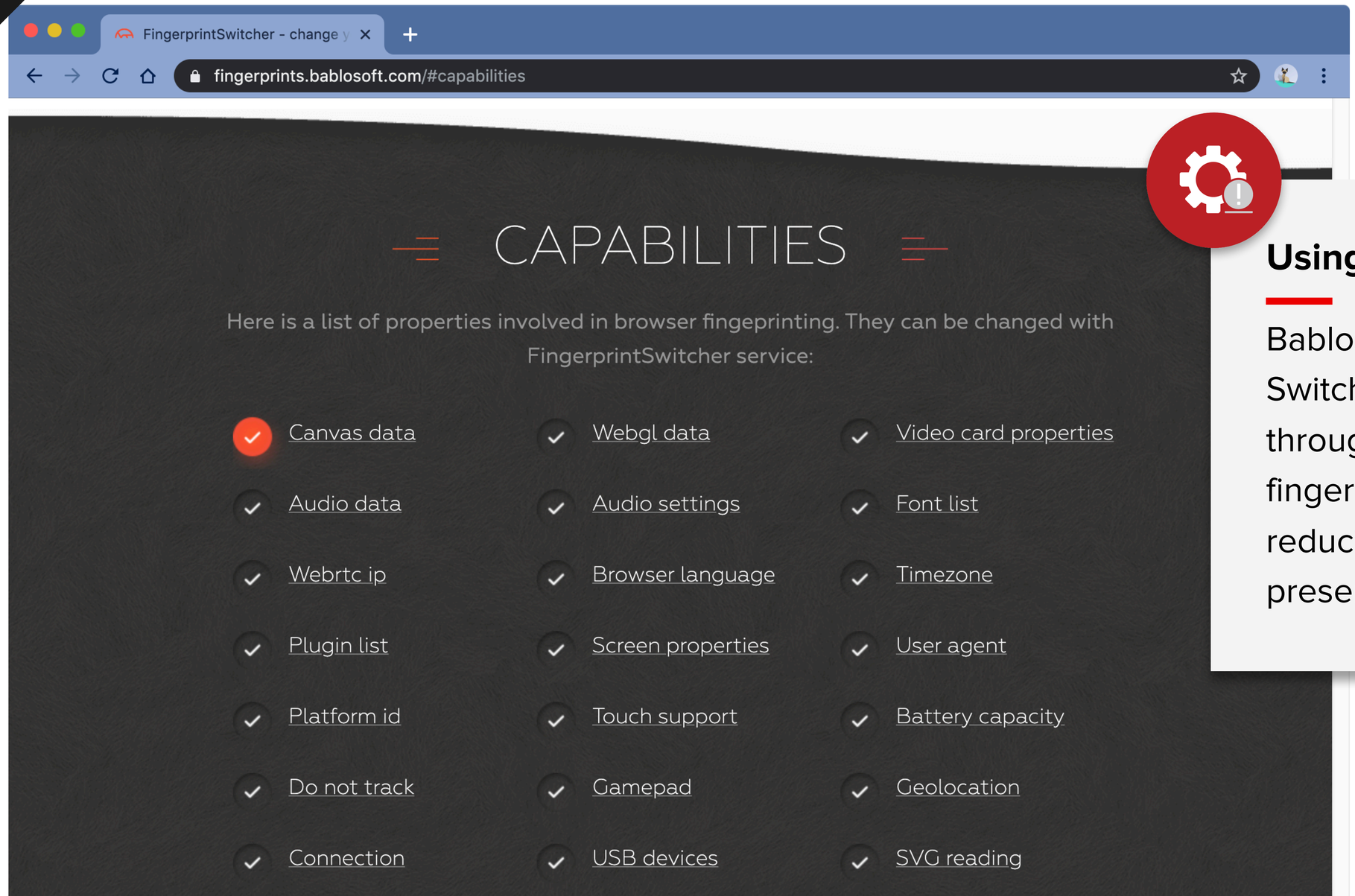
Validating Fingerprint Data

Good Users don't lie much.

Attackers lie a lot. They use a handful of clients but need to look like they are coming from thousands.

Those lies add up.

Iteration 7: Use real device data



Using Real Values

Bablosoft's Fingerprint Switcher allows a user to cycle through a real browser's fingerprintable data points, reducing the number of lies present in the data.

This keeps going but the direction is clear.

The end goal is perfect emulation of humans and their environments.

We're calling these Imitation Attacks

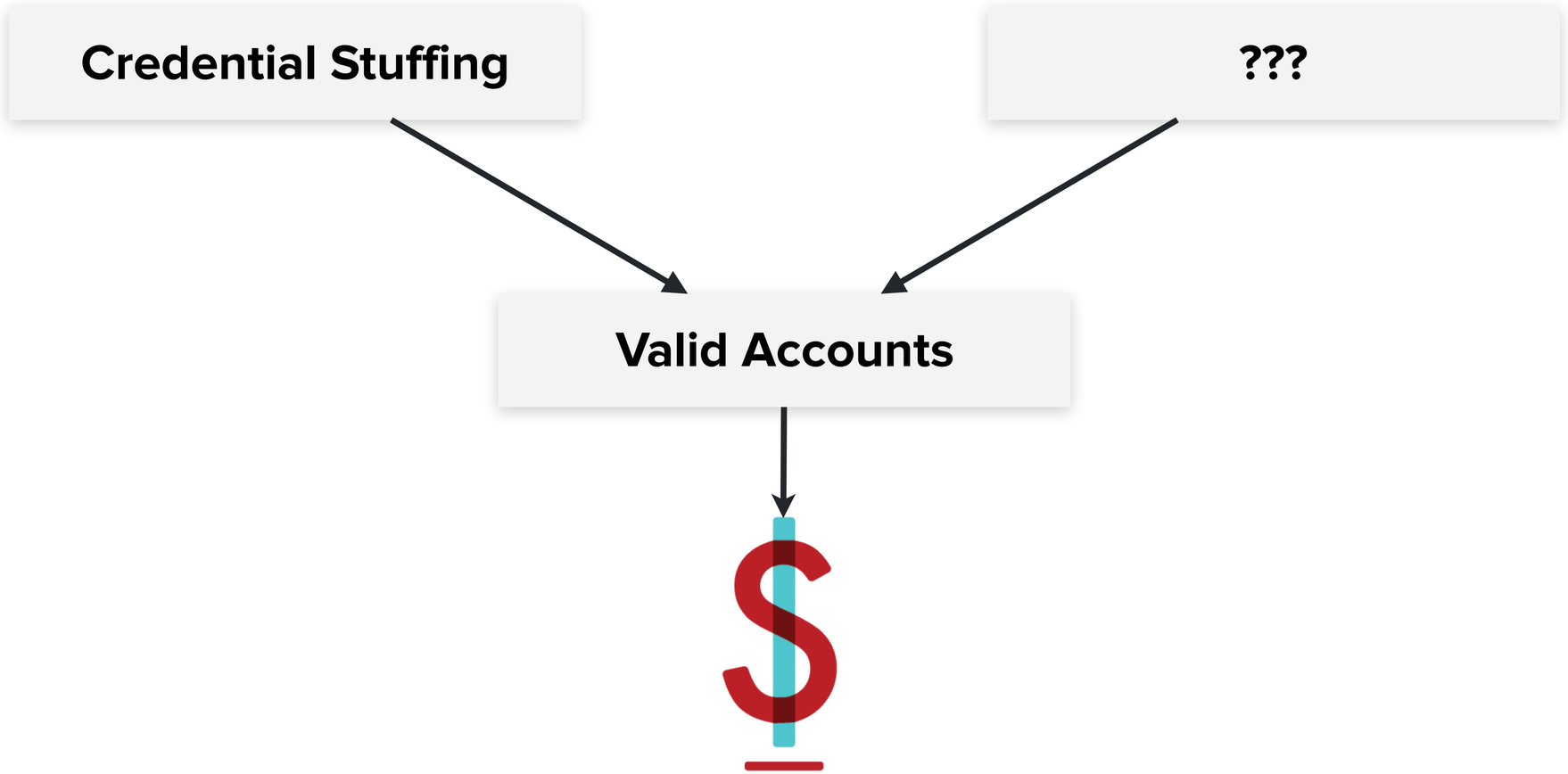
Imitation attacks indicate sophisticated fraud from dedicated adversaries.
The aim is to blend in and bypass risk & automation defenses.

Not all automation is an imitation attack, not all imitation attacks are automated.

1. Why credential stuffing is evolving
2. How credential stuffing has evolved
3. **Where do we go from here?**

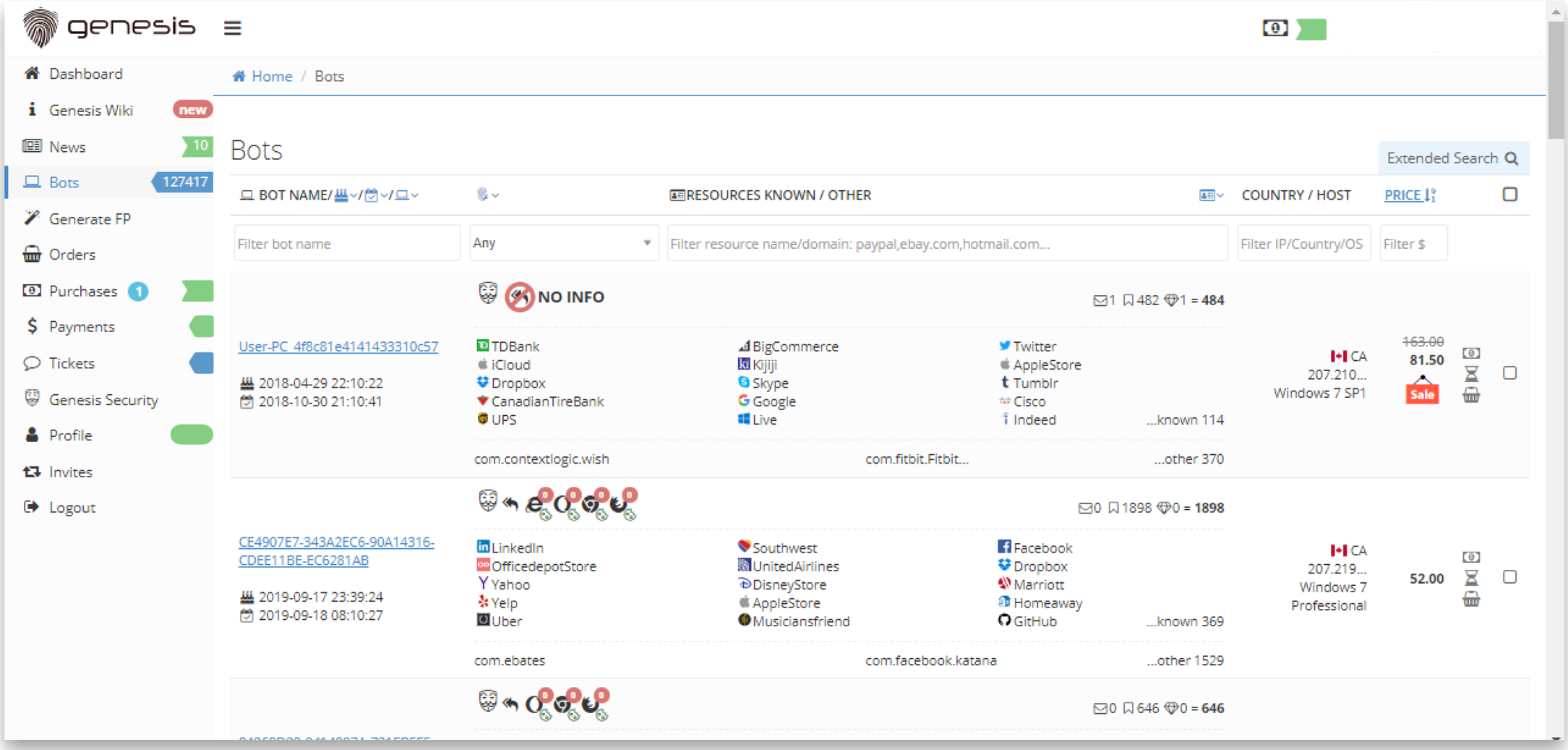
The value in our accounts is not going away.

As we raise the cost of credential stuffing there is greater incentive to diversify attacks.

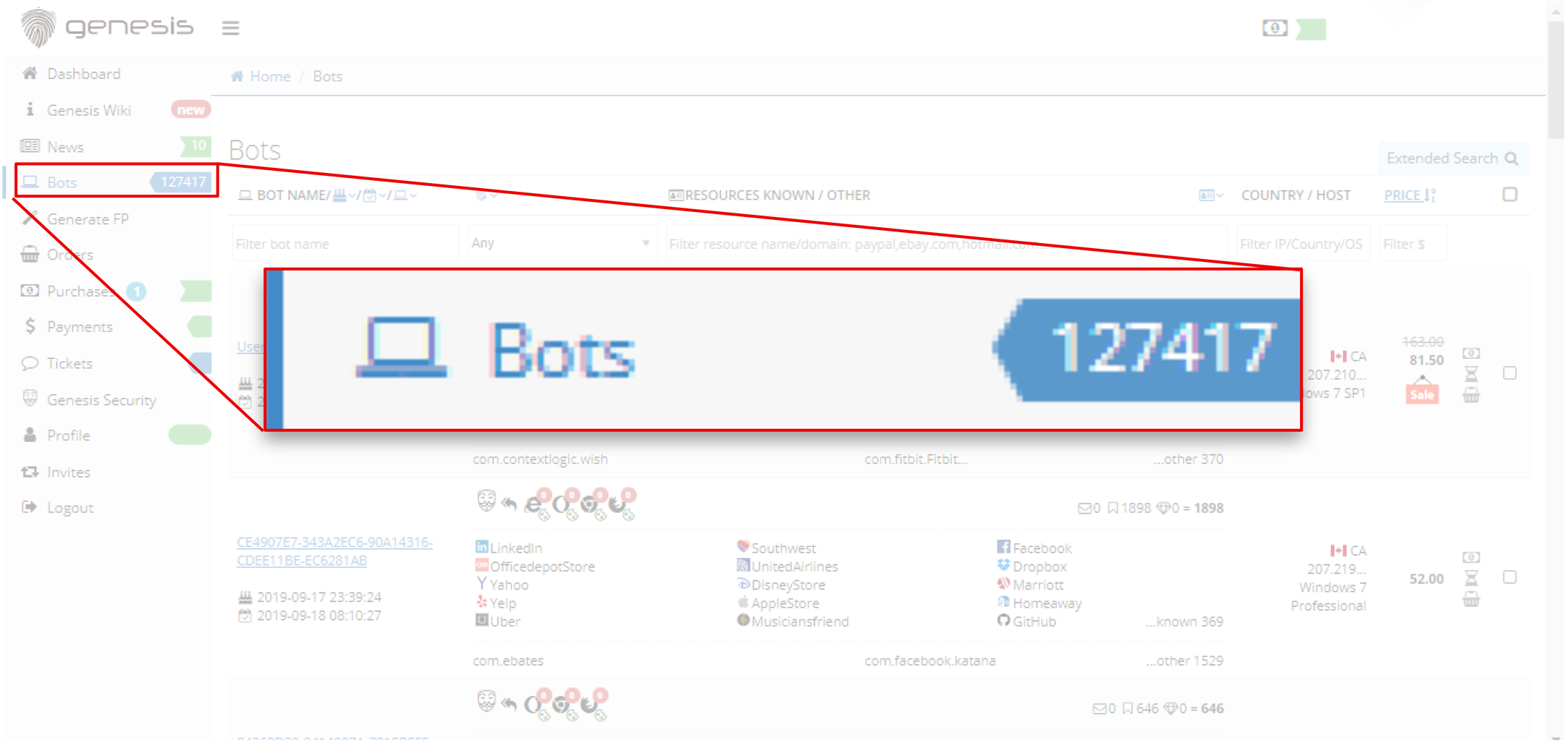


Genesis is an early example of what's next.

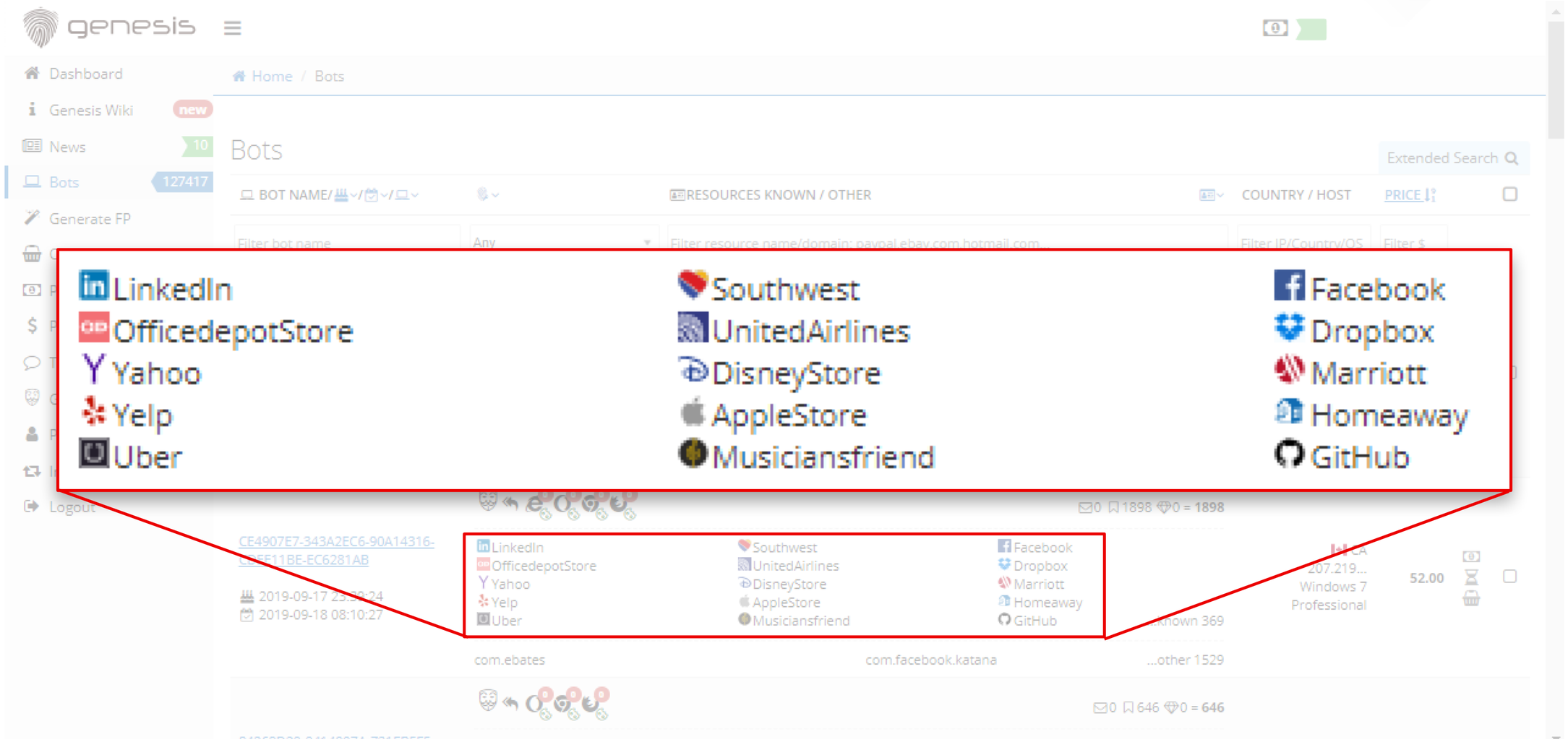
Malware that resides on the victim to scrape account and environment details.



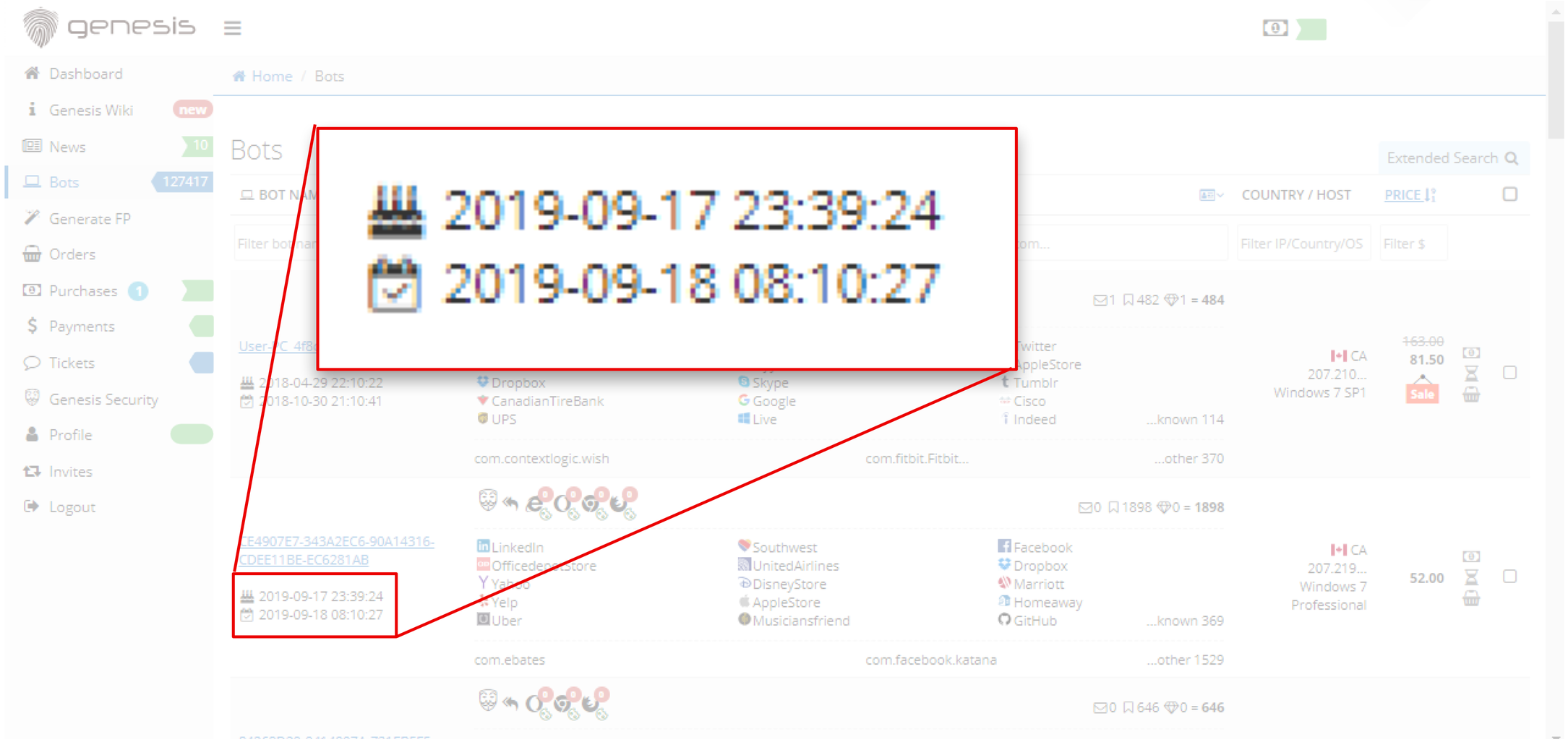
Thousands of infections and growing.



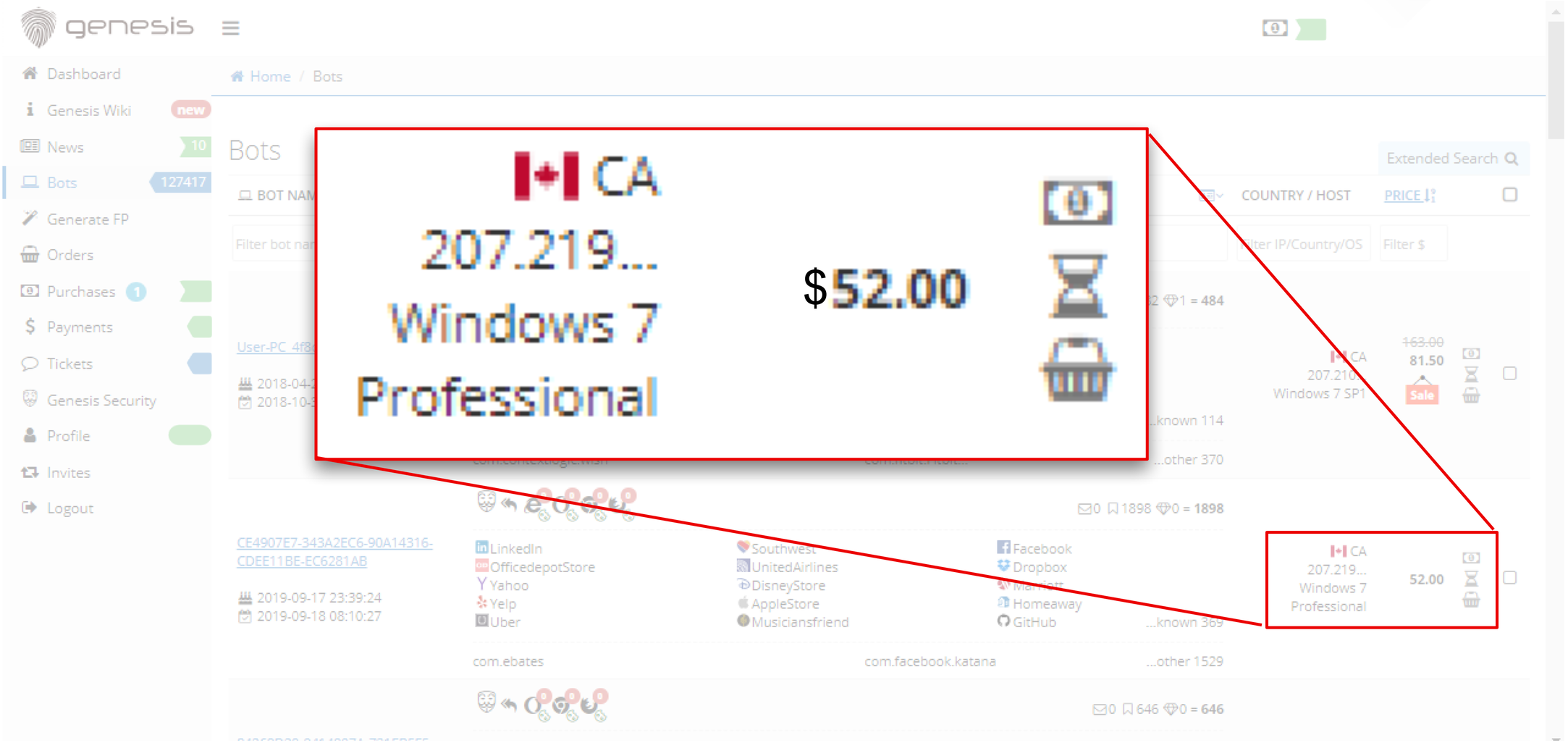
Advertises the high profile accounts the bot has already scraped.



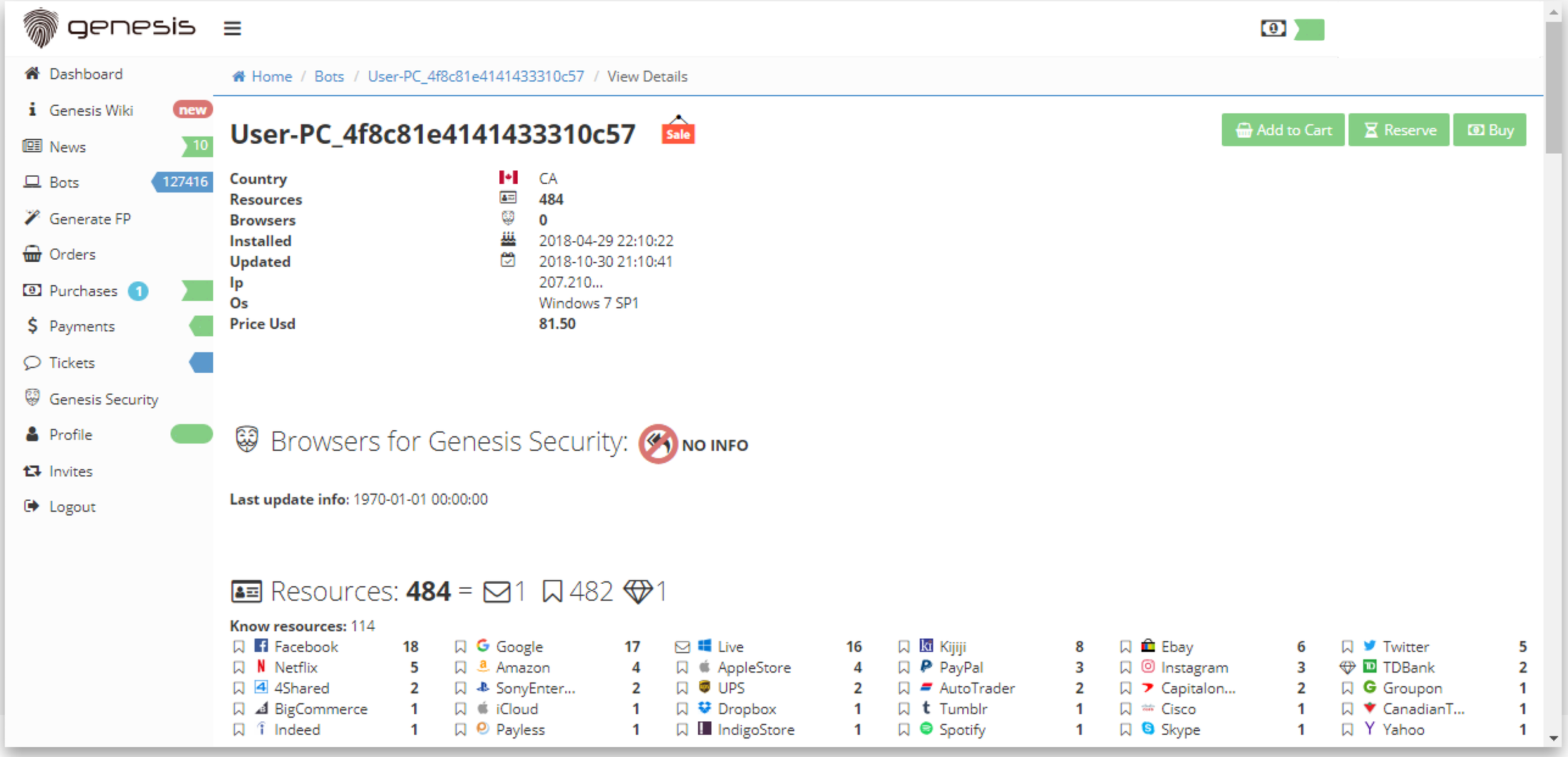
Regularly updates its records with newly acquired accounts.



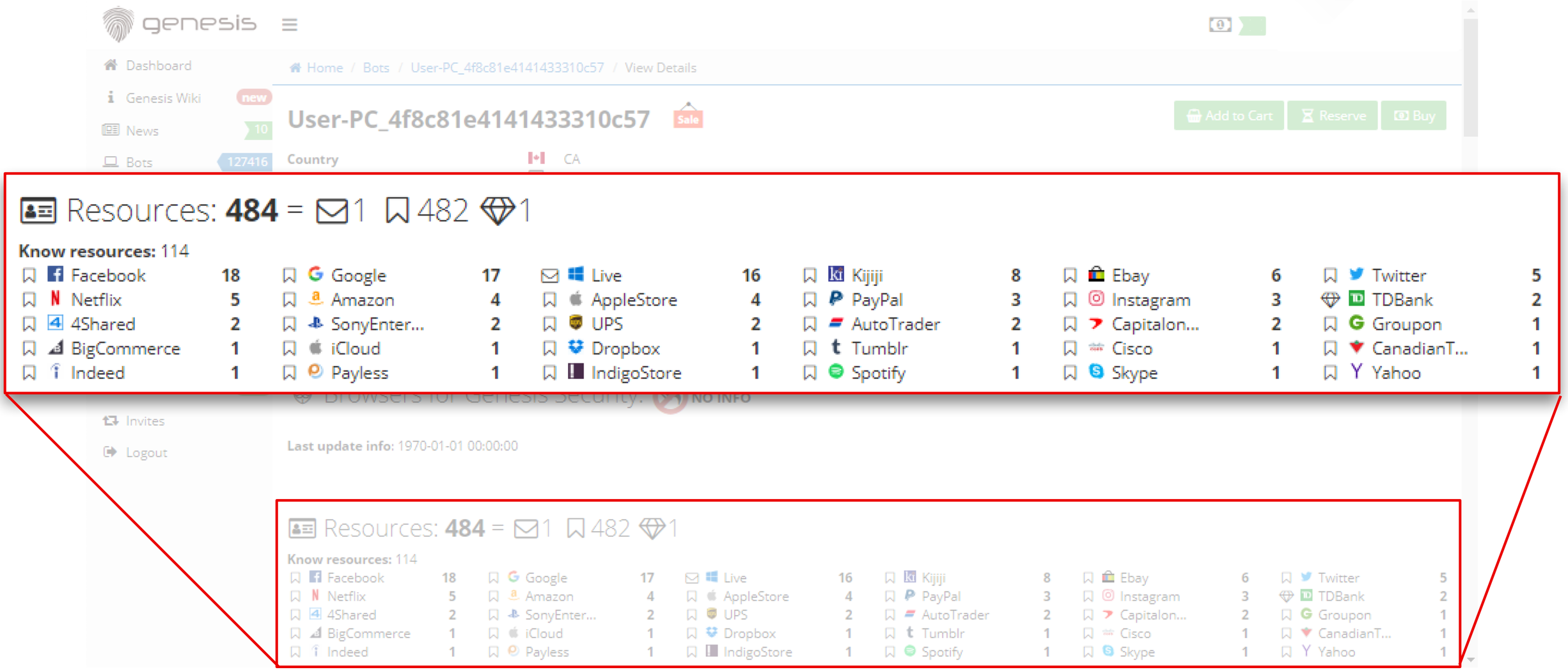
Each bot and its data is sold as one unit



Bot detail page

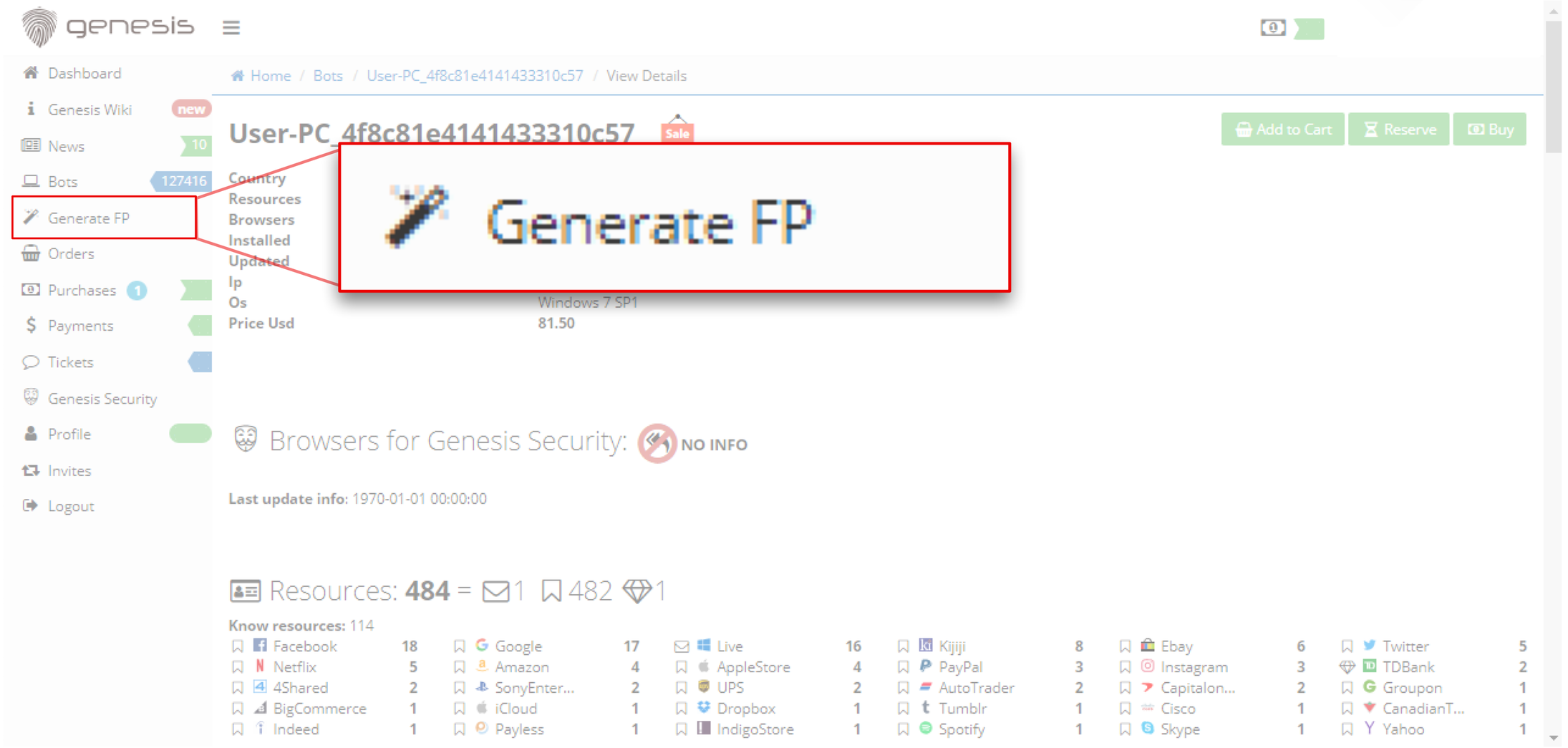


Bots have hundreds of scraped resources and accounts.

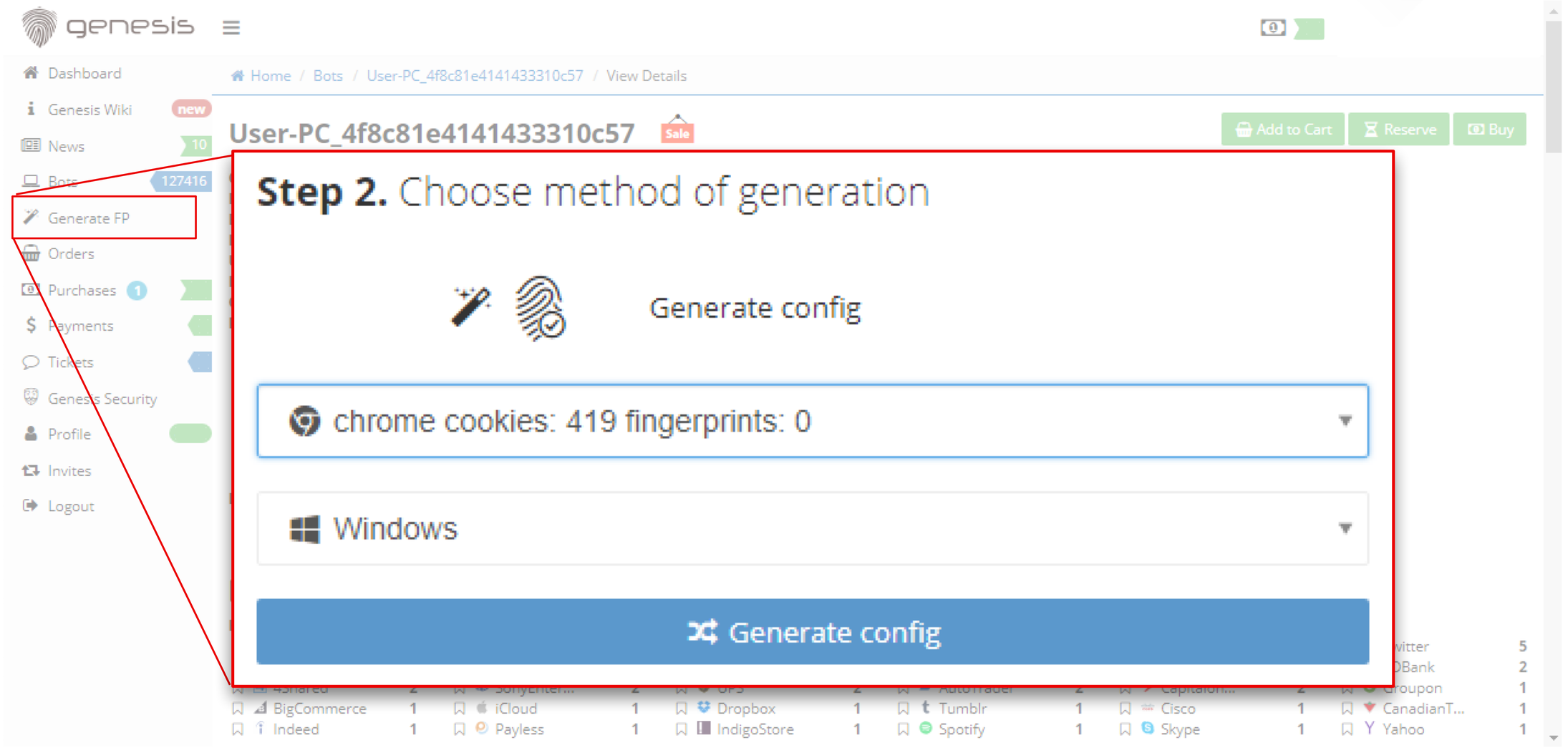


Genesis can generate the fingerprints of your exact target.

This bypasses many risk-scoring mechanisms that look for activity from new devices.

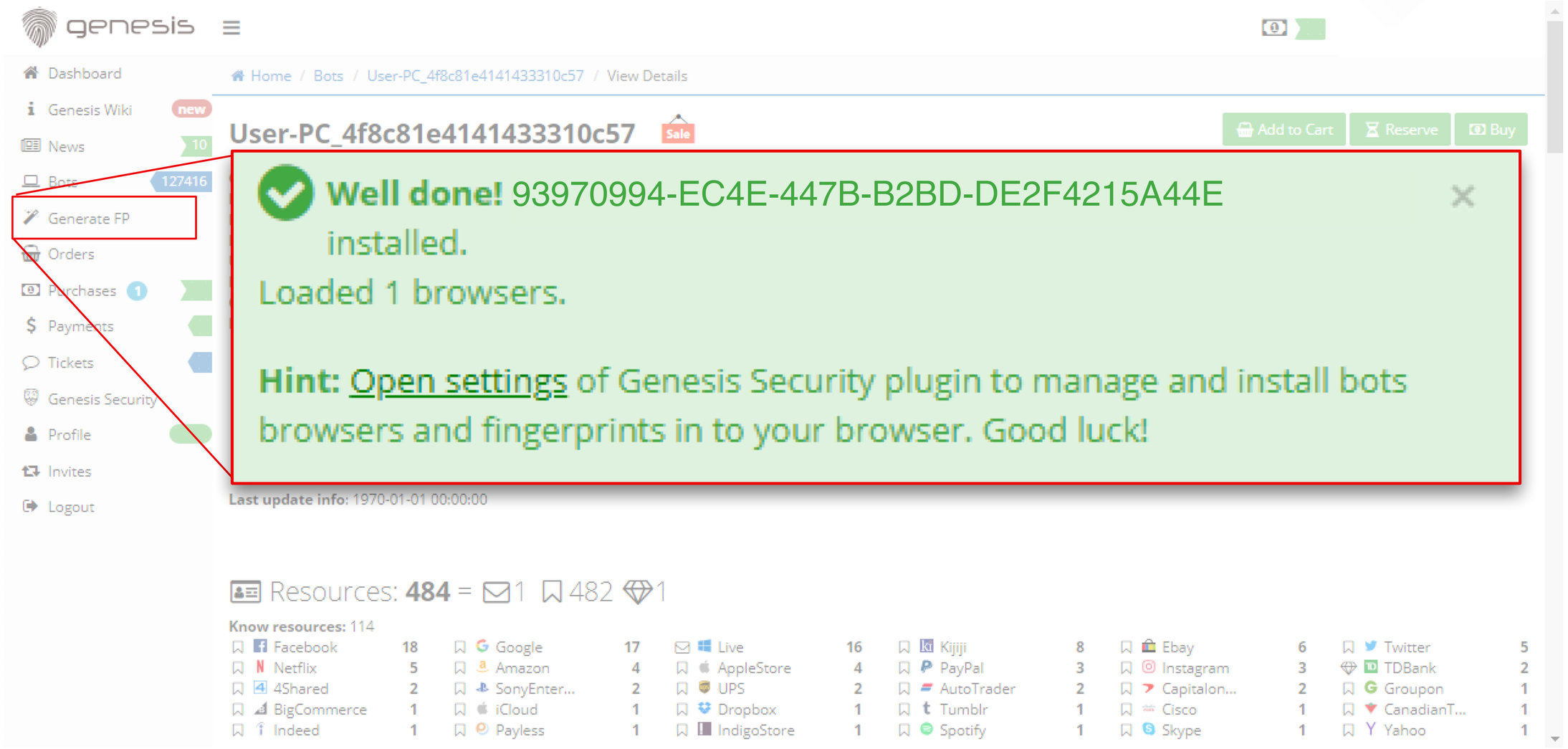


Select the fingerprint you are looking for



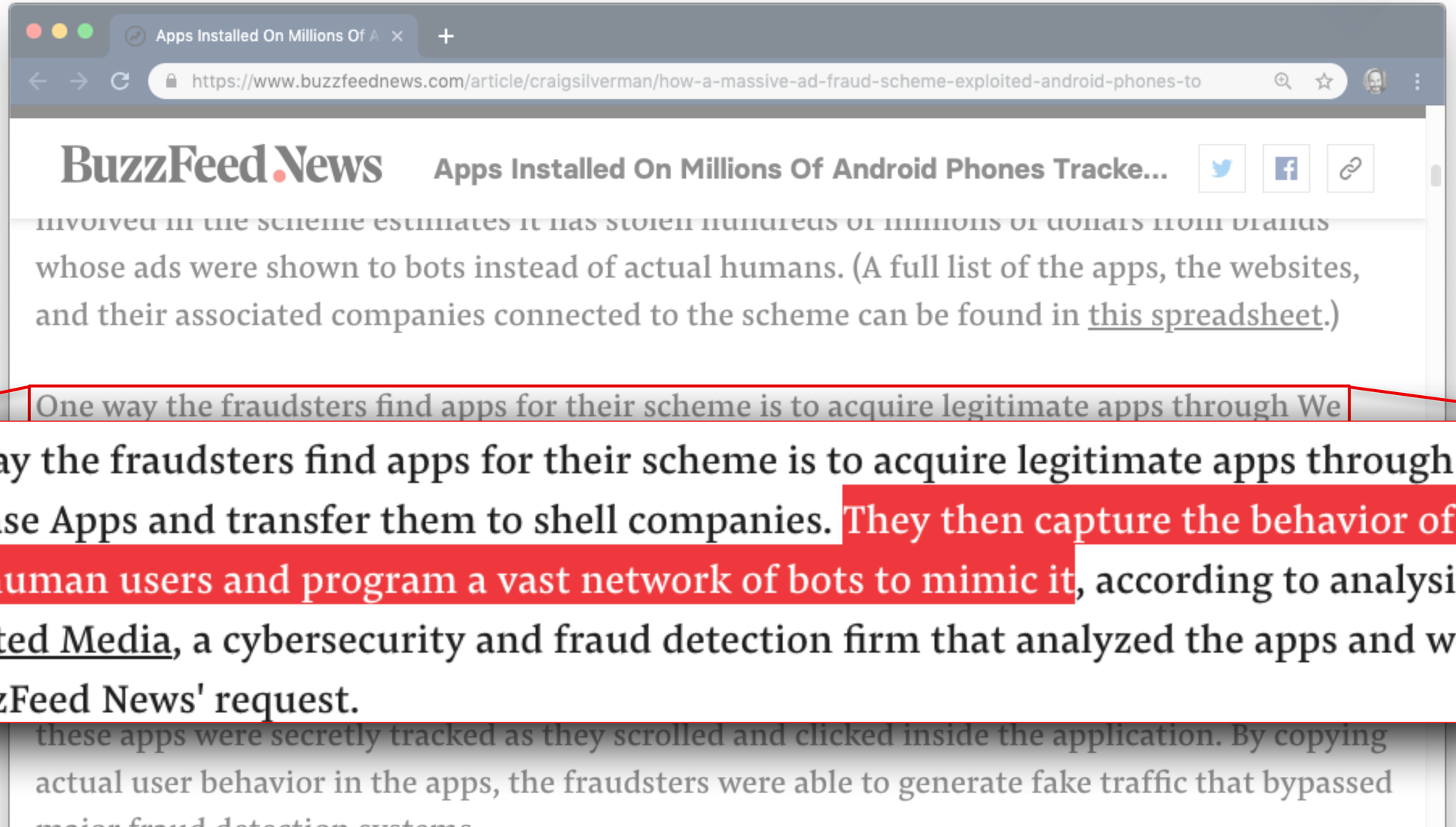
And load it into the Genesis Security Plugin

Voila! You are now your target.



Malware that scrapes, learns, imitates, and proxies through its victim is next

We've started seeing the signs in ad fraud.



One way the fraudsters find apps for their scheme is to acquire legitimate apps through We Purchase Apps and transfer them to shell companies. They then capture the behavior of the app's human users and program a vast network of bots to mimic it, according to analysis from Protected Media, a cybersecurity and fraud detection firm that analyzed the apps and websites at BuzzFeed News' request.

This is a human problem, not a technical problem.

There are no silver bullet solutions against humans. (Except literal silver bullets, but...)

Advanced credential stuffing is sophisticated fraud. It is more than simple automation. **Fraud teams aren't staffed for this, they need help.**

Imitation attacks are designed to blend in. **Look deeply even if you think you don't have a problem.**

Attackers are economically driven. We need to attack the economics. **Every defense will fail if the value is still there.**

THANK YOU

Jarrod Overson
@jsoverson on twitter, medium, and github.