

```
> your credentials were leaked
```



YOUR CREDENTIALS WERE LEAKED.

So what?



lure



steal



abuse

```
> whoami && whoami --the-other-one
```

Who we are



Xavier Mertens

@xme

- Freelance security consultant & incident handler.
- SANS Internet Storm Center handler; trainer & speaker.
- Runs the blog /dev/random and a personal malware lab.
- Threat-intel & detection side of this research.



Pascal Kahn

@teqagogo

- SOC and offensive/defensive tools developer.
- Built the open-source measurement rig used in this study.
- Decoys, bait and time-to-abuse measurement.
- Happy to help you replicate it.

> so your creds leak. so what?

Everybody gets phished. Three questions.

Q1

**Do they even
get used?**

Q2

How FAST?

Q3

From where?

we stopped guessing and measured all three – then asked: and then what?

> the idea

Stop defending. Start baiting.

we submit credentials we own to the phishing page...

...then watch who comes to use them.

CANARY GAMBIT

classic honeypots wait to be found – we plant the bait ourselves.

> current record

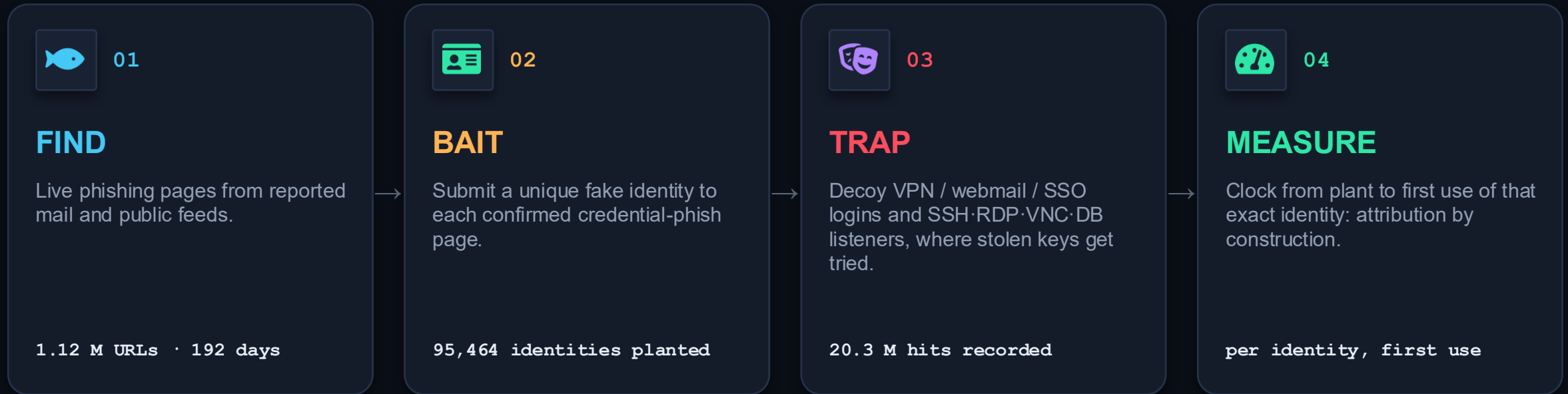
16 SECONDS

from credential planted → first abuse on our honeypot

Source: Sri Lanka. Faster than your SOC pours its first coffee.

```
> method # four steps, one question
```

The method: find · bait · trap · measure



every step serves one question:

how long between “leaked” and “used”?

fabricated identities only · no real victim data · aggregate figures · open-source implementation: last slide

```
> [bait] canary-identities --unique-per-url
```



Meet Margaux. She isn't real.

~1.9M

identities in the pool

95,464

identities planted (192 days)

1 / URL

a hit names its source

most honeypots tell you THAT you were attacked.

The canary gambit tells you WHICH page sold you out — by name.

UNAMBIGUOUS ATTRIBUTION

+ a quarter are made to look like admin / service accounts (admin@, svc-backup@)

```
> method --guardrails
```

Guardrails & limits of the method

WE DO

- fabricate every identity ourselves: no real victim data, ever
- plant only on pages confirmed as live credential phish
- quarantine payloads; nothing is ever executed
- share aggregate figures openly; raw IOCs only in trusted circles (TLP)

WE DON'T

- touch real accounts or real users' data
- crawl dark-web markets or forums: own indicators only
- claim campaign or actor attribution for the abuse set
- hack back: decoys only answer what reaches them

limits

We only see reuse that comes back to OUR decoys, so every rate is a lower bound. Geolocation of all hits is partial (2.0%); 111 of 122 abuse events are geolocated. No login attempts on the decoy portal since early July: a gap in observation, not an end of abuse.

```
> [trap] listen 443,22,3389,5900,3306
```



Where the stolen keys get tried.

decoy login portals + classic-port listeners: everything an attacker sends is recorded.

webmail

SSO

remote

VPN portal

SSH:22

RDP:3389

VNC:5900

MySQL:3306

they think they're logging into your VPN.

They're filling out our incident report.

GOTCHA

+ the decoys also hand out fake secrets in fake files (.env, configs); reuse of those is tracked too (slide 15).

```
> SELECT * FROM receipts -- 192 days, read-only
```

The receipts: 192 days of real abuse.

1,124,283

phishing URLs analyzed

95,464

identities planted

122

real abuse events

68

identities abused

00:16

fastest TTA

23,350

attacker IPs (all traffic)

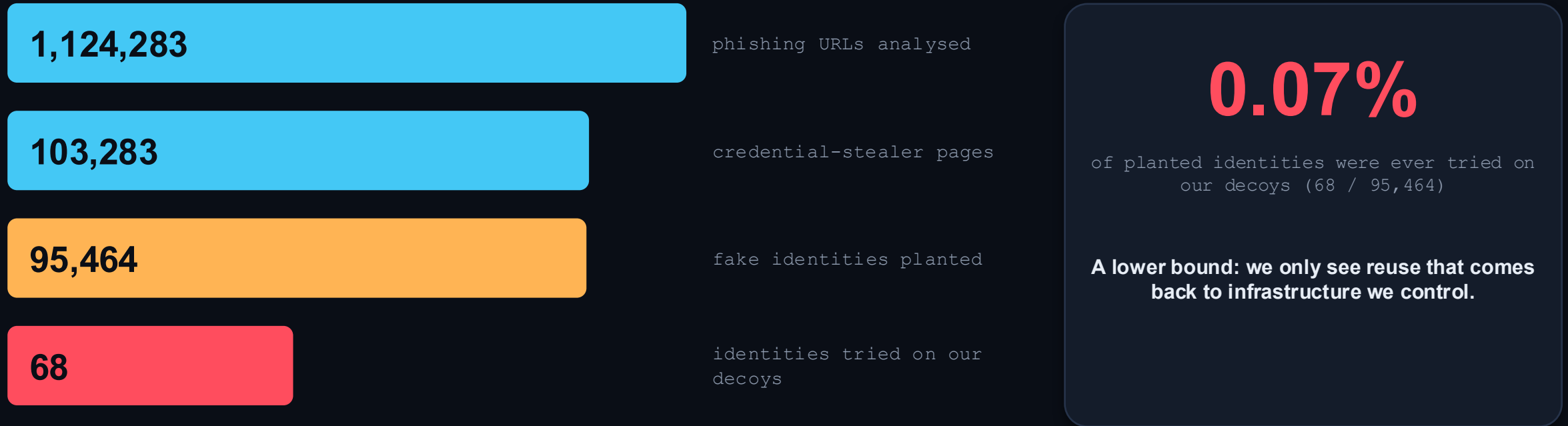
honest framing

Only 0.07% of planted identities (68 / 95,464) were ever tried on our decoys, by 82 attacker IPs. The story is the SPEED of those that were, and no single IP dominates (top = 4.9%).

2026-03-19 → 09-27 · aggregate figures · 32 proxy-masked events re-attributed via X-Forwarded-For (see notes)

> Q1: do they even get used?

Q1 — Do they even get used? Rarely. Then fast.

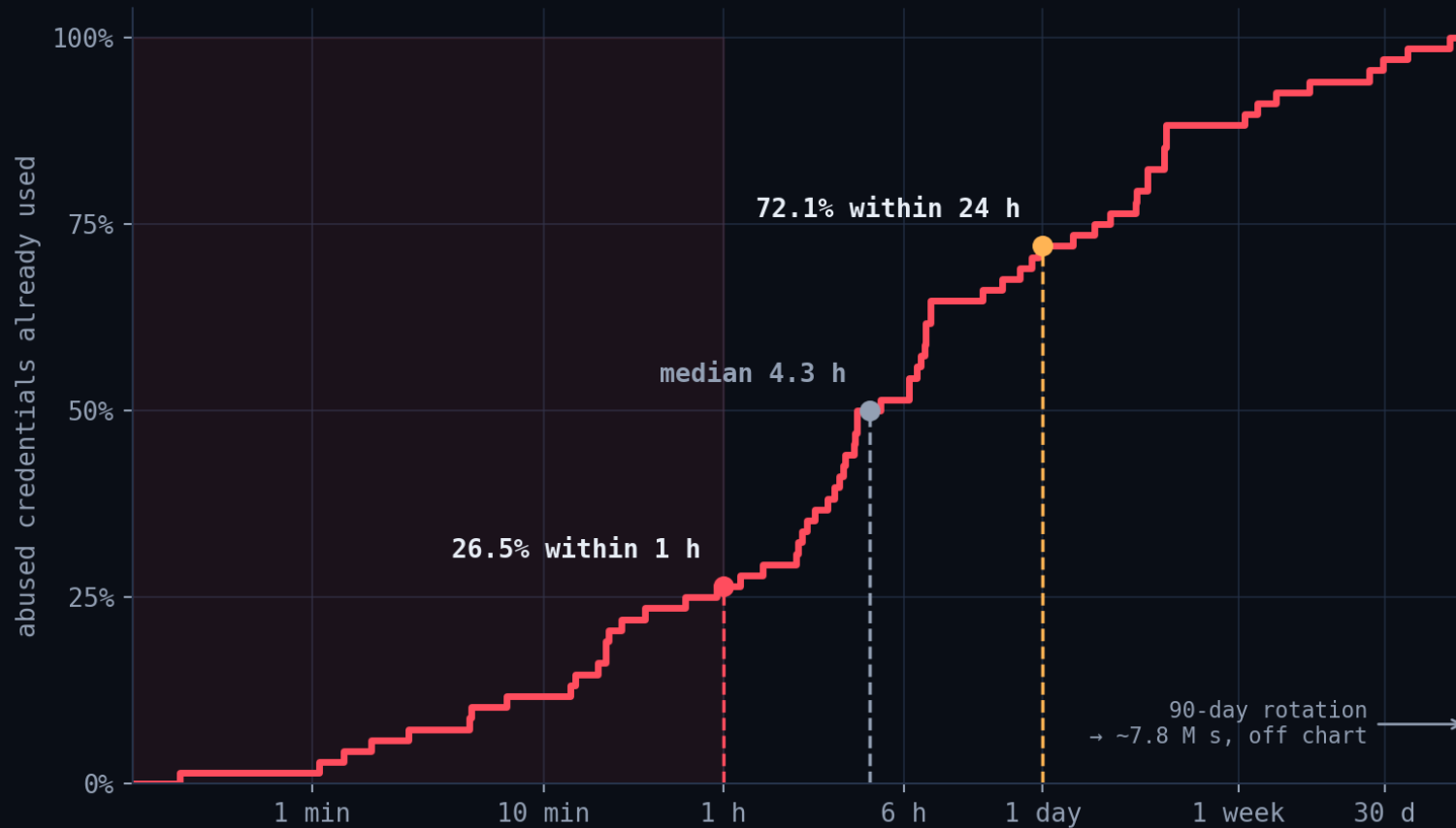


bar length: log scale · 122 abuse events · 82 attacker IPs

0.002% of the traffic (the decoy login portal) produced 100% of the credential abuse.

```
> Q2: how fast? -- first use per identity (n=68)
```

Q2 — How fast? Faster than your rotation policy.



26.5%

first used < 1 hour

72.1%

first used < 24 hours

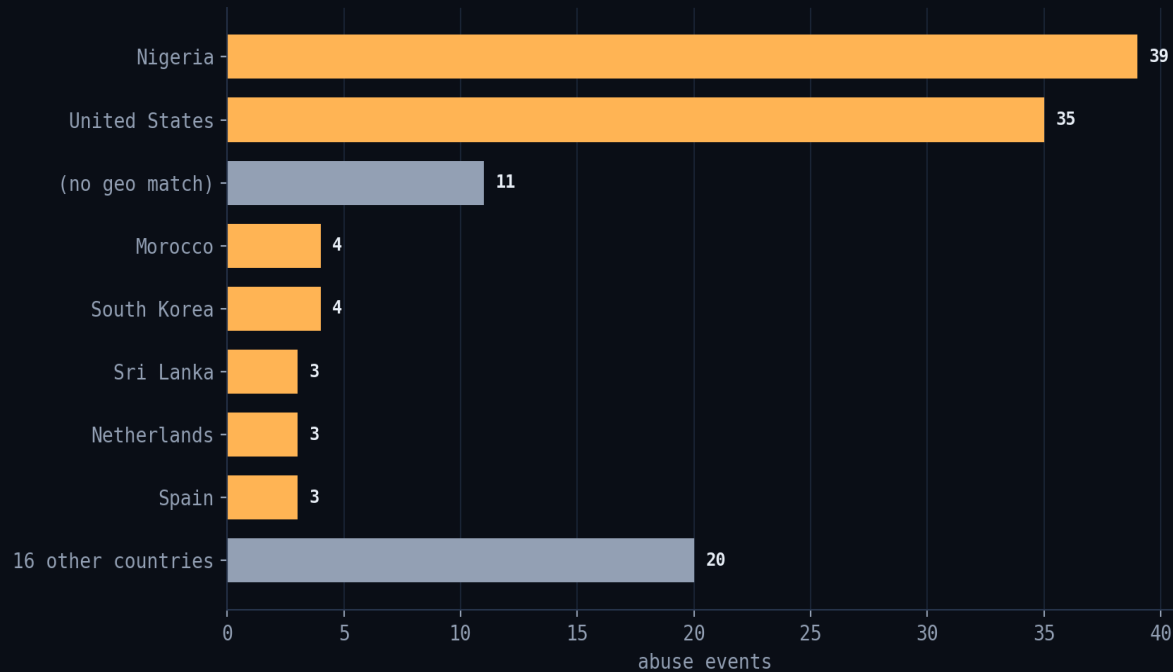
4.3 h

median, per identity

> GROUP BY country, network_type (N=122 events)

Q3 — From where, and who.

geography of abuse — volume



~60%

hosting / VPN · 0%
Tor

98%

humans in real
browsers (119/121)

97%

pasted the
password (117/121)

speed record ≠ volume leader:

Nigeria leads volume (39); Sri Lanka set the 16 s record.

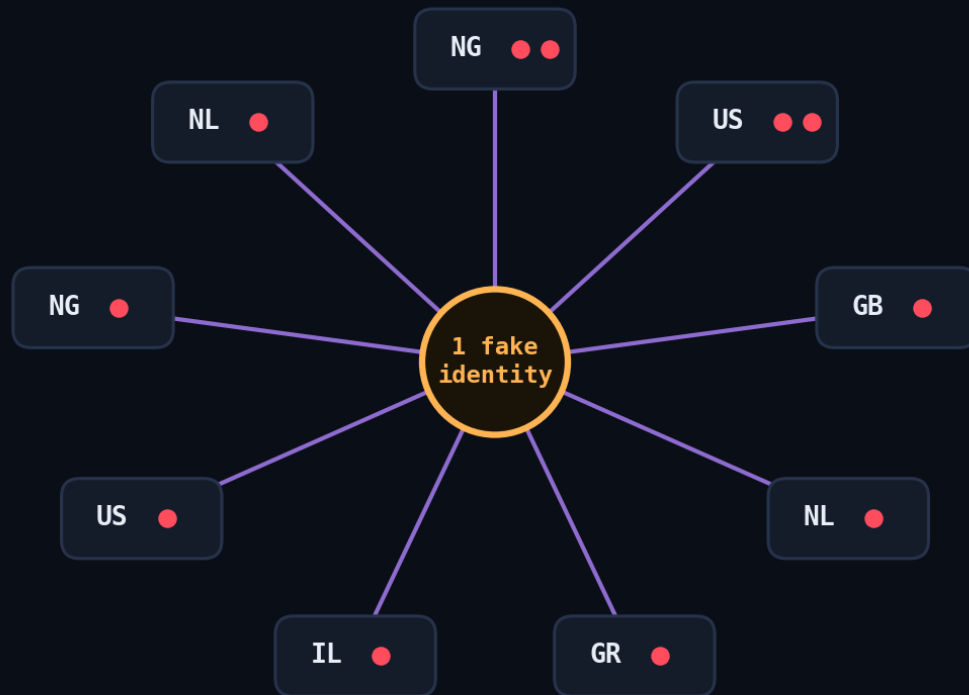
bots scan the site —

humans come back later and paste the password from a list.

no campaign attribution claimed — abused rows weren't campaign-tagged.

> ...and then what? -- per abused identity (n=68)

...and then what? Stolen credentials circulate.



network box = one ASN · red dot = one attacker IP · country codes as geolocated

16%

used from more than one network (11 / 68)

10%

used from more than one country (7 / 68)

38%

replayed more than once (26 / 68)

so what for a CSIRT:

the IOC is the account, not the phishing domain.

left: the most-circulated identity: 11 IPs · 9 networks · 6 countries · 13 events, 2026-05-22 → 06-20 · network = ASN; lower bounds

```
> SELECT ... FROM canary_tokens WHERE lure = '.env'
```

Secrets found in files get handed on.

238

fake DB passwords replayed

87%

replayed from a different IP

82%

from a different country

17.5 min

median theft → replay (fastest
9 s)

the path

decoy .env file served → fake DB password grabbed by 12 IPs → replayed on our database
decoy by 8 other IPs (one UK hosting server alone: 193 of 238)

different bait

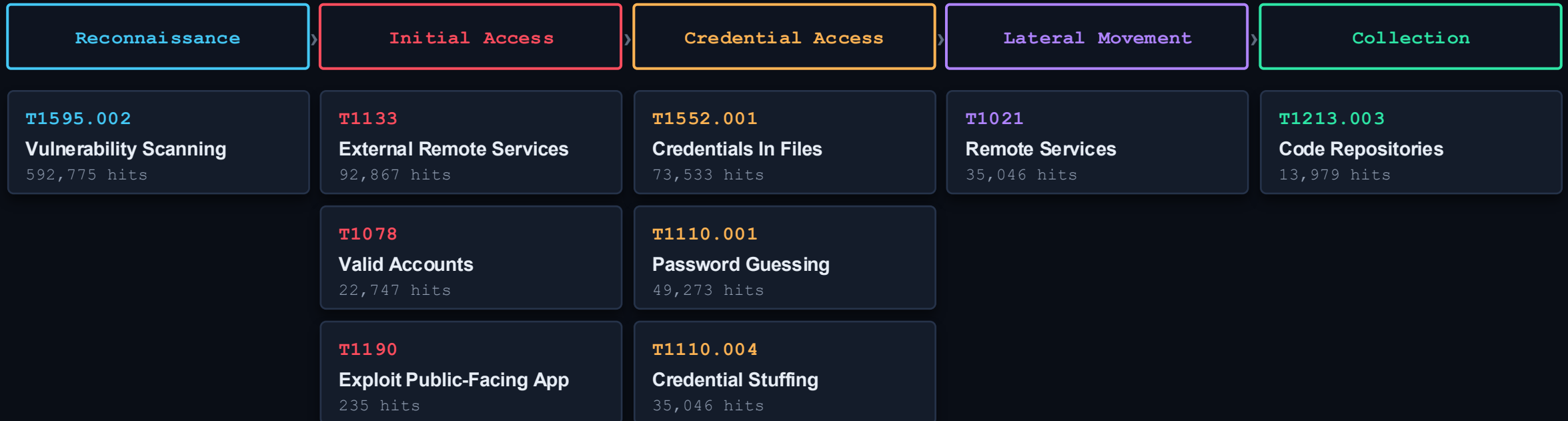
fake secrets in decoy files, NOT the phishing-planted identities. Separate funnel, never added to the 68.

ATT&CK T1552.001 Credentials In Files · triggers: May 103 · Jun 128 · Jul 6 · Aug 1 (last 2026-08-15)

a stolen secret is typically replayed by someone else, somewhere else, within about 20 minutes.

```
> GROUP BY attack_technique -- 750,862 tagged hits
```

The chain, in ATT&CK terms.



51,825 payloads captured · 0 executables · 1,783 scripts –
they came for accounts, not implants.

also: T1550.001 Application Access Token 235 · T1087.003 Email Account discovery 201 · 11 techniques in total

```
> method --beyond-our-login
```

Seeing reuse beyond our own login page.

We only see reuse that comes back to our decoys. Tripwires inside the bait could look further.

HTTP beacon

a unique link in the lure; fetched from anywhere, it calls home. No DNS, no third party.

```
status: not deployed yet · 0 planted
```

Document canary

a .docx with an external image: opening it calls home.

```
status: not deployed yet · 0 planted
```

Cloud-key bait

an AKIA...-shaped key + beacon (true global detection needs CloudTrail).

```
status: not deployed yet · 0 planted
```

admin-looking identities

25% of seeds look privileged (admin@, svc-backup@, it.admin@, aws-admin@): 847 planted since August, 0 used so far. No logins at all on the decoy portal since early July, so no privileged-vs-ordinary comparison yet.

```
method, not results: reported honestly · no sightings yet
```

```
> lessons --for-anyone-running-deception
```

Lessons for anyone running deception.

Monitor the lure path end to end

A decoy nobody can reach measures nothing. Run a synthetic login through the real front door daily; alert on a day with zero logins.

evidence: 0 logins since 07-02, decoy still up

Platform health is data quality

Headless-browser fleets fail by resource leaks, not crashes. Track launch failures, temp space and outage days next to your findings.

evidence: 12.2% launch failures; 4 outage days in Sep

Check your proxy chain before trusting source IPs

Behind a bridge or reverse proxy, the 'client' may be your own gateway. Verify forwarded-header trust hop by hop.

evidence: 70.7% gateway IPs hid 32 abuse events

Models drift without a feedback loop

A retraining pipeline nobody feeds labels to is a frozen model. Budget analyst time for labelling.

evidence: 0 labels; agreement 89.6% → 77.1%

and normalise per day:

one mailbox flood made June look like a 15× improvement in planting.

> what your CSIRT should do on Monday

What your CSIRT should do Monday morning.

- 1 Revoke on the event, not the calendar**
user says "I typed my password" → reset + revoke sessions & refresh tokens within the hour
because: 26.5% used < 1 h
- 2 Assume the session is gone too**
AitM kits steal the token: phishing-resistant MFA (FIDO2 / passkeys), admins first
because: MFA ≠ enough
- 3 Hunt the account, not the domain**
alert: same account from ≥ 2 ASNs, or hosting / VPN ranges, within 24 h of a report
because: 16% circulate
- 4 Plant your own canaries**
decoy admin-looking accounts in your IdP / AD, fake secrets in shares: any use = true positive
because: 87% handed on
- 5 Share it**
MISP / STIX 2.1 with TLP for IOCs; aggregate figures openly
because: 6 countries

```
> git clone [[REPO URL - TBD]]
```

Reproduce it: open method, open tooling.

CODE

the rig we built for this study, PhishTrack:
open source, Apache-2.0

SHARE

STIX 2.1 & MISP event exports; an
aggregate-only report (no IPs, no victim
data)

COMPARE

run it in your constituency and send us your
curve: different feeds, different bias

```
[[REPO URL - TBD]]
```

everything in this talk

can be rebuilt from open components, and we want your numbers.

> the answer

So what?

72%

of stolen credentials that get used are
used within 24 hours. the fast ones: seconds.

assume reuse

leaked = already in use

MFA ≠ enough

AitM steals the session token

steal the canary idea

bait, don't just block

your credentials were leaked.

Now you know so what. — Questions?

Q&A