



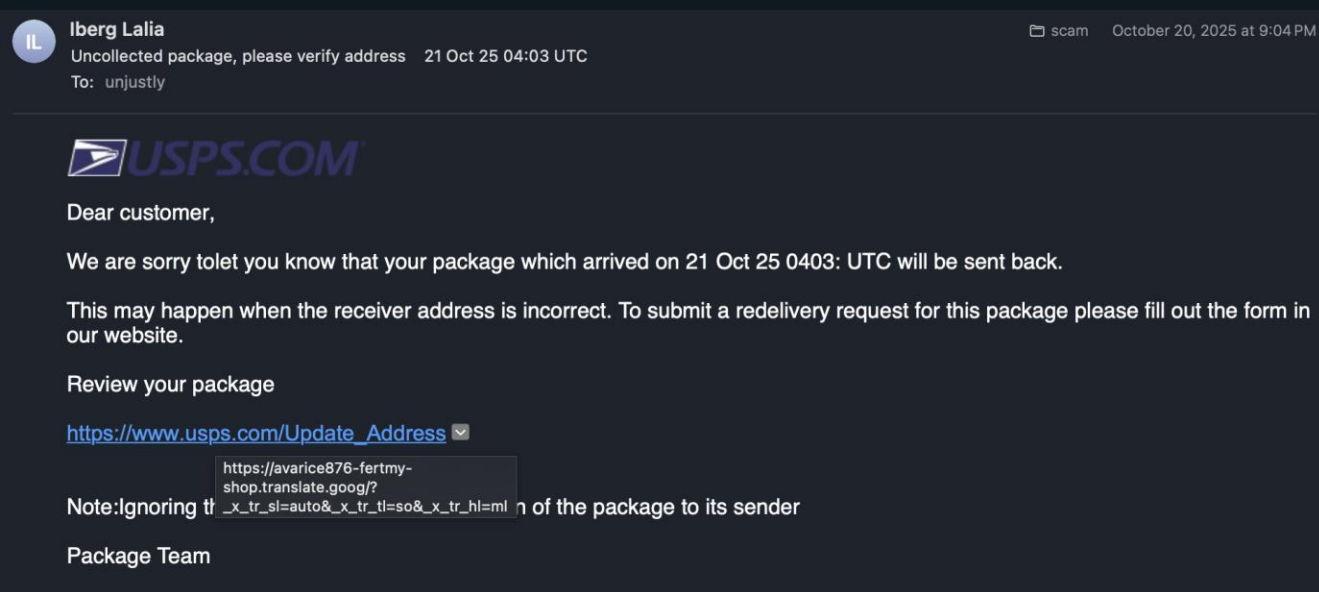
Hiding in Plain Sight:

Security Blind Spots in Composite DNS Queries

Vadym Tymchenko
Darin Johnson
Rajat Patel



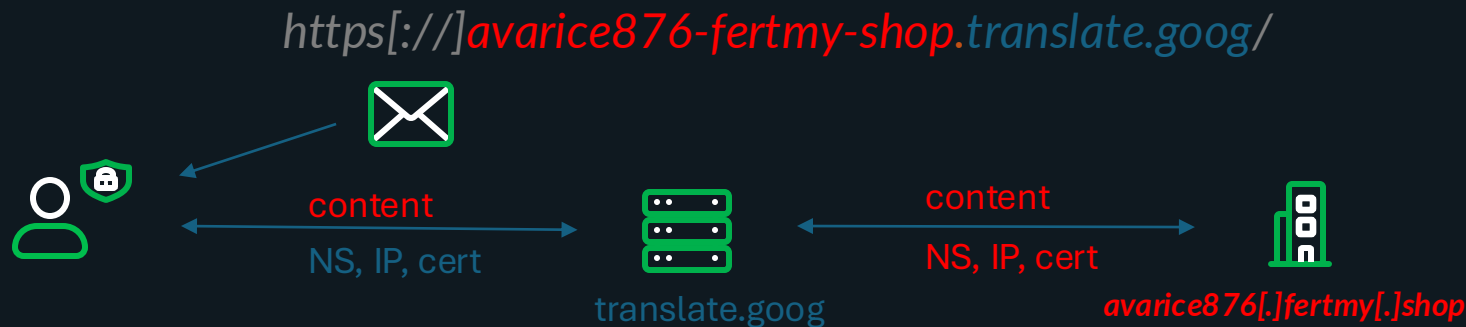
The Hook



- Looks like ordinary google-translate link
- Opens a phishing site avarice876[.]fertmy[.]shop
- Standard controls don't stop it

https://avarice876-fertmy-shop.translate.google/?_x_tr_sl=auto&_x_tr_tl=so&_x_tr_hl=ml

What Actually Happened: Composite Domain



- Destination domain is embedded in the query
- Security controls analyze only the service domain
- The service retrieves content from the destination domain

<encoded-domain>.[<suffix>.]<service-domain>

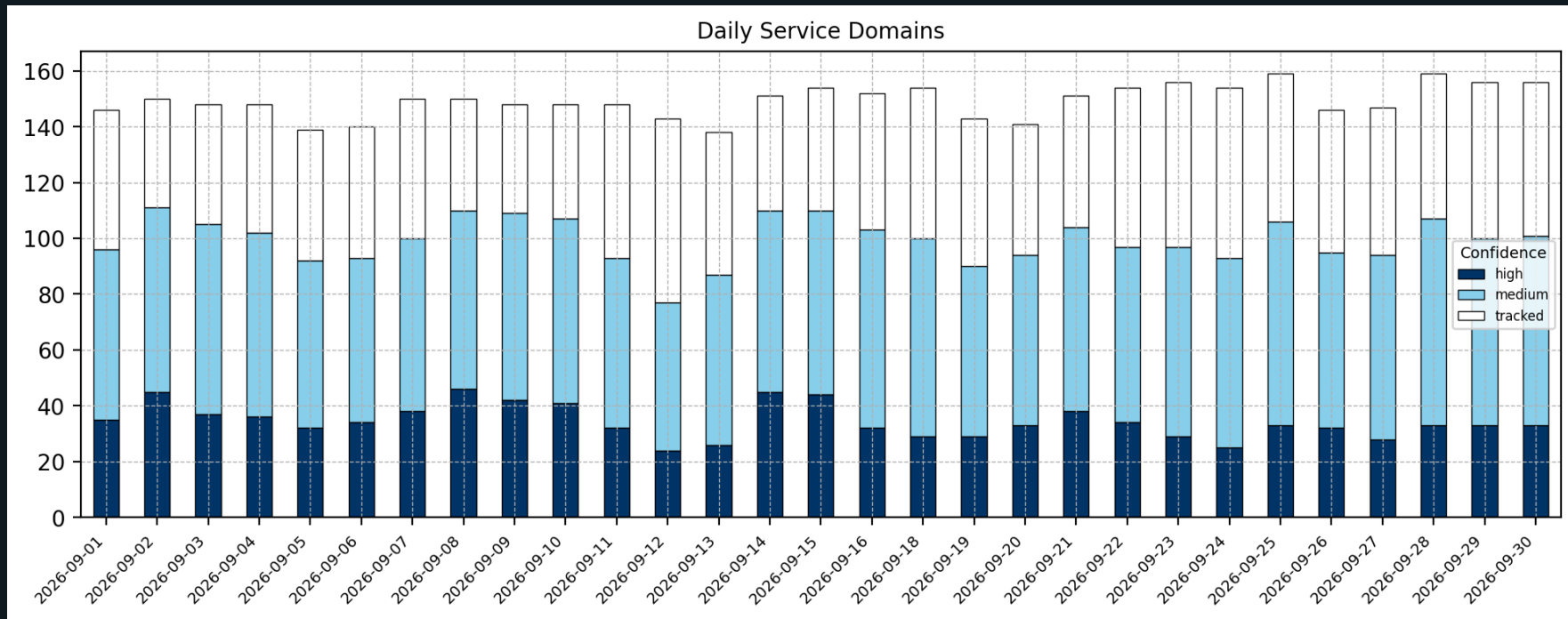
Security blind spot: the embedded domains are never validated.

Classification: Categories of Embedding Services

Category	Abuse Risk	Description	Example
Content Proxy	High	Retrieves and renders content	translate.goog
DNS Proxy	High	Resolves embedded domains	DNS Resolvers
Metadata Services	Low	Returns info <i>about</i> embedded domains	surbl.org
Static Content	Medium	Same content for all subdomain	Parked domains

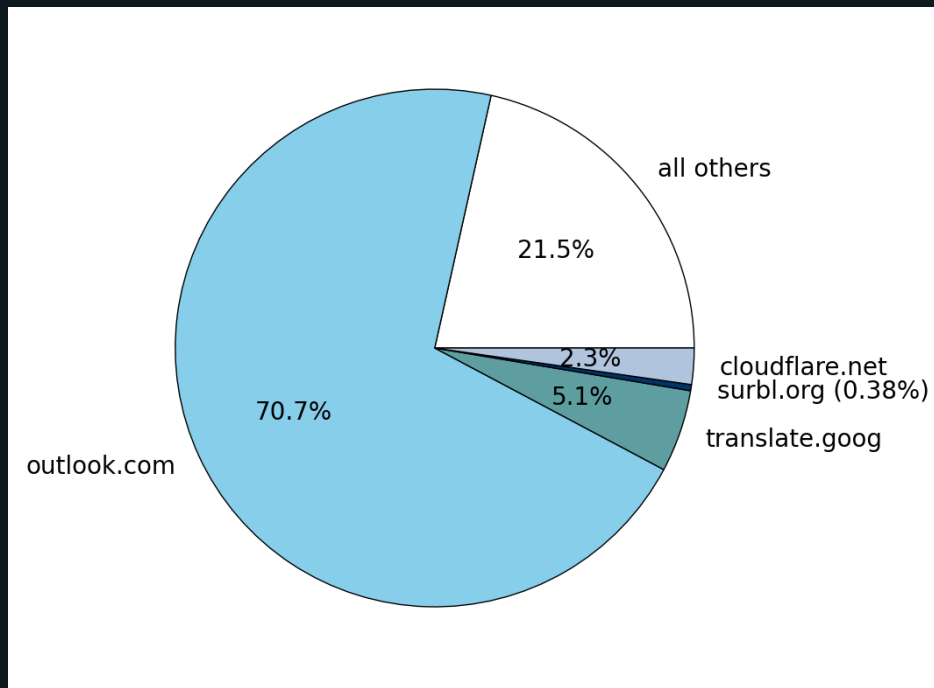
Discovery Results: Size of the Blind Spot

- Approx 100 proxies/services day

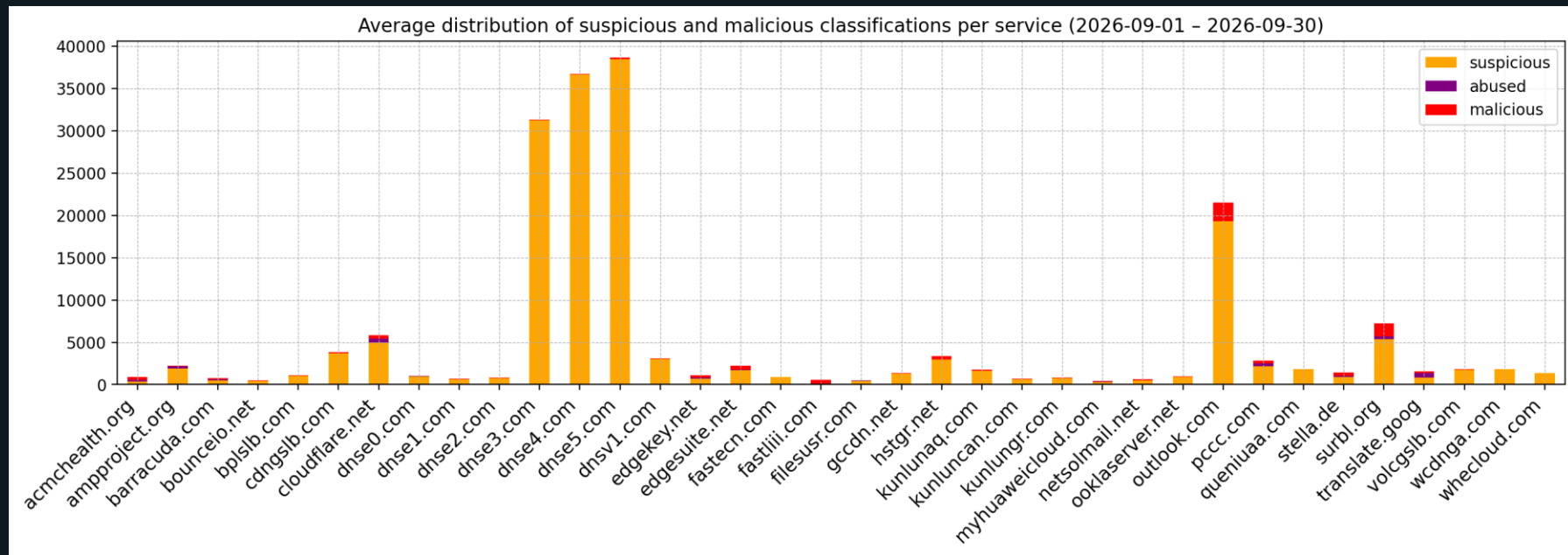


Discovery Results: Service Distribution

- Constitutes 7% of daily customer traffic
- Sept 2026
 - 2.1M distinct embedded domains
 - 5436 new domains (created within 7 days)
 - 12963 stale domains (inactive over 30 days)
- ~10K domains unobserved in direct traffic



Discovery Results: Size of the Blind Spot



Why an Reponse Policy Zones are not enough

We can't block translate.google

Adding **fertmy-shop.translate.google** to RPZ doesn't block **arvice876-fertmy-shop.translate.google**

Adding **arvice876-fertmy-shop.translate.google** doesn't block **arvice877-fertmy-shop.translate.google**

Unbound supports lua and python, so regex exists, but I also want to apply my entire RPZ against translate.google

Our Solution

For a Content Proxies and DNS Proxies, we decode the subdomain:

`arvice876-fertmy-shop.translate.google` becomes `arvice876.fertmy[.]shop`

We then reapply this domain to our RPZ's

Result: Blocks all domains in an RPZ even when they use a Content or DNS proxy.

NB: We don't apply this rule to metadata services (like Spamhaus or antivirus) as that would degrade a useful service.

Service Level vs Request-level Policy

BLOCK the service

Blocking SURBL -> breaks spam filtering
Blocking translate.goog -> breaks translation service

ALLOW the service

Translate phishing page gets through every control

Old Model

Trust service domain

Decide per service

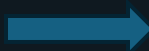
Evaluate what is visible

New Model

Analyze the embedded destination

Decide per request

Evaluate what is hidden



Takeaways

- Trusted services can carry untrusted destinations
- Service-level decisions are insufficient
- Risk depends on how each service handles the embedded domain
- Extract and evaluate embedded domains



infoblox®

GET OUR
RESEARCH



QU