

Zombie Awakening: Stealthy Hijacking of Active Domains through DNS Hosting Referral

Researched by: Eihal Alowaisheq, Siyuan Tang, Zhihao Wang, Fatemah Alharbi, Xiaojing Liao, XiaoFeng Wang

Presented by: Jordan He

Intro



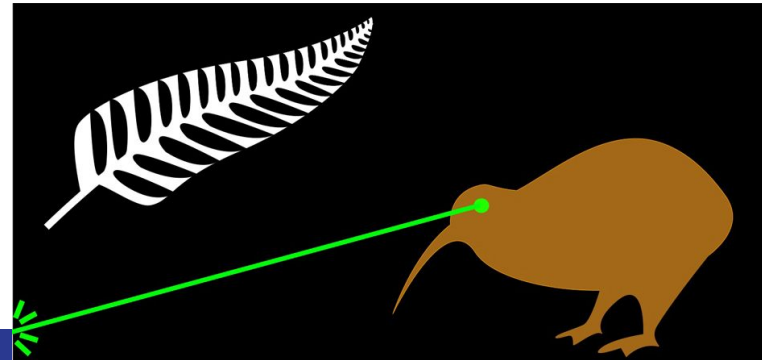
Background knowledge

DNS resolves domain names to their IP addresses

DNS runs on a hierarchical system

Take `www.laserkiwi.nz`, it can be split into

- Root (.)
- top level domain (.nz)
- second level domain (laserkiwi.nz)



More on DNS

Nameservers responsible for different levels (DNS zone) are recursively discovered and queried by a Client side resolver

Resolvers cache received DNS records for efficiency

Resource Records (RRs) organise resolution information.

A DNS response contains multiple RRs, or sections



DNS Response

Answer section is most important, contains the A record, or the IP of the domain

A domain can have more than one nameservers as seen in authority section of example.

The resolver choose one with shorter round trip time or randomly picks one nameserver

```
;; ANSWER SECTION:
example.com.          46294  IN  A    93.184.216.34

;; AUTHORITY SECTION:
example.com.          43020  IN  NS   a.iana-servers.net.
example.com.          43020  IN  NS   b.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.  118512 IN  A    199.43.135.53
b.iana-servers.net.  35275  IN  A    199.43.133.53
```

Image from Eihal et al. 's paper

Domain hijacking

Resolver stores a bunch of records (every record has domain names and IPs)

When it receives a DNS query, find the record the client wants and returns the result (the IP address)

But if the record itself is manually tampered with, then the result returned would be false



Hosting providers

DNS hosting provider - a service providing authoritative nameservers to help customers of the service to manage their domains' DNS records

How to use - customer creates an account with provider and adds a domain. Then provider assigns a set of nameservers to manage the domain and respond to its DNS queries

This name rings a bell?



What are the risks?

But if a customer stops using the hosting service

Provider's nameservers stop responding to queries for her domain

If the NS records that point to these deactivated hosting nameservers are not cleaned up, then these records become stale

Most providers do not validate their customers' ownership of domains they claim under their account



Basic ideas

Assume there is at least one stale NS record in a domain's SLD zone that points to a DNS hosting service not verifying domain ownership

Both nameserver at the hosting service and the targeted resolvers are exploited to divert traffic to attacker's destination



Stale NS record at SLD zone

A domain usually has multiple NS records

SLD zone has an “A record” pointing to domain’s actual IP address

But the SLD zone could also have a stale NS record pointing at a provider that no longer manages the domain

Vulnerable stale NS records called zombie referrals (Zrefs)

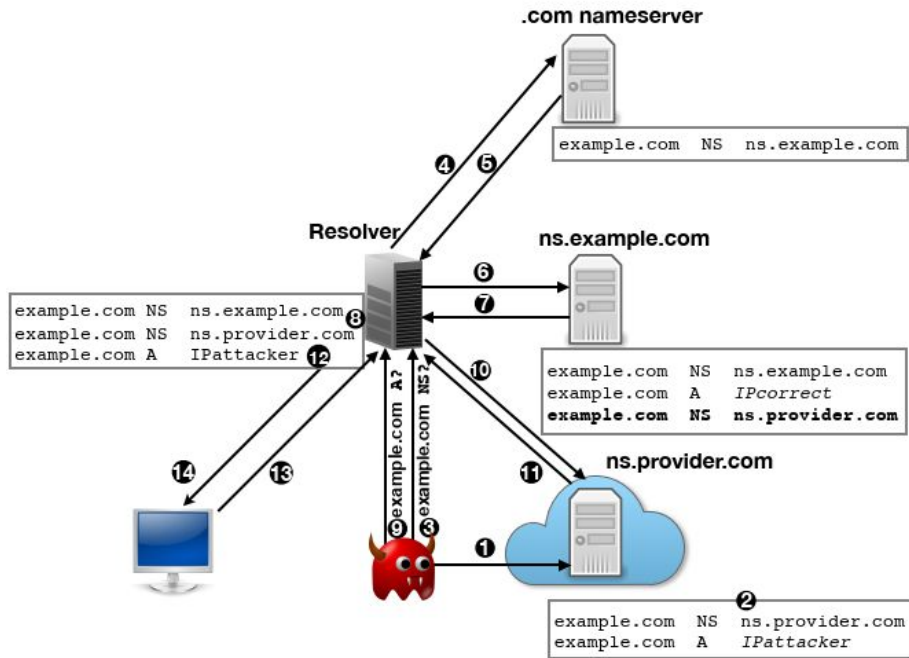


Problem

Stale NS records in SLD lead to so called Zombie awakening attack (ZAW). A domain hijacking attack that exploits said records



Diagram



(b) DNS configuration and resolution for a hijacked domain

Image from Eihal et al. 's paper


Looks complicated but...

Basic idea is: claim the vulnerable domain at the provider and poison the targeted resolver's cache with the wrong IP via the stale NS record

Three main participants: attacker, resolver, innocent client


Steps

Attacker

- Sets up an account at the hosting provider directed by the stale NS record of the domain, example.com
 - Claim the domain and set its RRs at the same nameserver on the stale record, ns.provider.com
 - Here, A record for the domain created that points to attacker's IP
 - Strategically queries the resolver for the NS records of the domain
- 


Steps

Resolver

- Resolver sends a request to the TLD (.com) nameserver if no answer found in cache
 - Further, this nameserver contacted for NS RRs in the SLD zone
 - Cached at the resolver side are the stale record pointing to ns.provider.com and the record with ns.example.com
 - NS RR from SLD zone replace those in TLD because referral from TLD has lower trust than authoritative response (IP address) from SLD
- 

Steps

Back to attacker

- Query resolver for domain's A record. Stale record has 50% chance to be used to find the A record
 - If so, the attacker IP is returned by ns.provider.com
 - Resolver caches this false IP until its TTL expires. Attacker can repeatedly query the resolver until the intended IP is received
- 

Steps

Client

- When she contacts the resolver to resolve the domain, the resolver gives the attacker's IP
- Client's traffic redirected to that IP
- Attack successful!



How to find hijackable domains?

Researchers have a mechanism called “Zrefinder” to find zombie referrals

- Identify public domains with stale NS records in SLD zone (potentially vulnerable domains or “PVDs”)
- Analyse which PVDs are linked to DNS hosting services
- Check if these services have policies to prevent PVDs from being attacked



Diagram

Will focus on two key steps in this workflow in the next few slides.

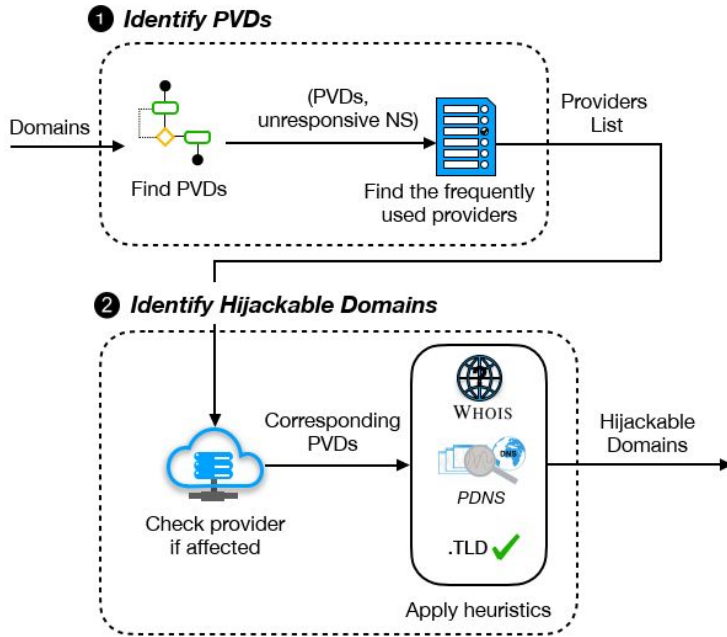


Figure 3: Workflow of ZreFinder to identify hijackable domains, where PVDs represents the potential vulnerable domains.

Image from Eihal et al. 's paper

Potentially vulnerable domains (PVDs)

- Locates a domain's NS records in SLD zone, check whether nameservers it points to still resolve the domain's DNS query
- Compare the NS RR Set for TLD and SLD for each domain
 - identify nameservers that appear on NS records in SLD set but not in TLD set
- For each such nameserver, check if its NS record is stale
- If target domain is indeed not configured at the nameserver, then it is considered a PVD



Non-hijackable domains?

Resolvable by another nameserver at provider	Registration data at provider	TLD restriction at provider
<p data-bbox="79 372 606 536">If domain has active NS record and zombie referral under same provider, ie. domain is active under another account at said provider.</p> <p data-bbox="79 583 490 615">If so, probably not hijackable</p>	<p data-bbox="678 372 1238 583">Provider may check domain ownership if the hosting service is operated by a registrar, ie. if a company operates both domain registration and DNS hosting.</p> <p data-bbox="678 626 1089 659">If so, probably not hijackable</p>	<p data-bbox="1277 372 1831 536">Domains with specific TLDs not served by some providers, so PVDs with these TLDs cannot be exploited via those providers</p>



What about the resolvers?

- Can public recursive resolvers be manipulated to enable a Zaw attack?
 - Need a “zombie resolution path” (claim of the domain with an IP pointing to attacker’s dest.)
 - Need to use this path to answer queries on the domains at resolvers
- Three sets of resolvers were analysed in the paper
 - Over 80% of popular public resolvers operators affected
 - Over 60% of open resolvers on a public list vulnerable to cache poisoning
 - Resolver operating through DHCP also found vulnerable



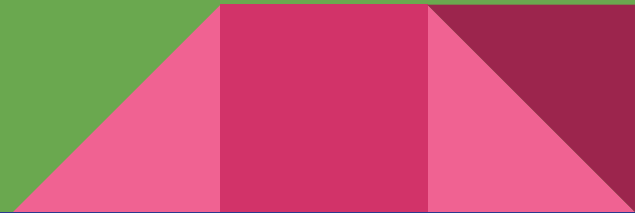
Attack complexity

- Measured in three dimensions
 - “Cache miss factor”
 - Chance of an affected resolver to pick a zombie referral
 - If hijackable domains have DNSSEC deployed
- Cache miss factor?
 - Attack only happens if a PVD’s legit IP address is not in the targeted resolver’s cache
 - High frequency of cache-miss on about half of the resolver operators surveyed



Solution

Is there one?



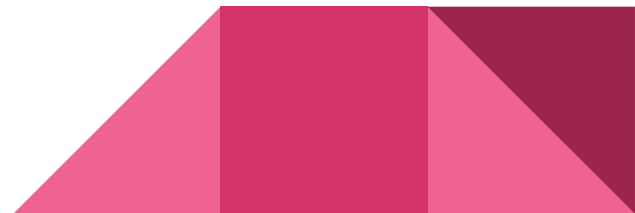
Mitigations

The researchers did not propose a one-click-solves-all solution in their paper.

Instead a series of recommendations were given, possibly because the problem is multi-dimensional in nature (DNS hosting providers, resolvers, clients...).

With providers, one major reason zombie referral exists is when a DNS hosting provider's SLD zone data is imported to a new DNS server.

Their recommendations focus on two areas.



Hosting providers

- For domain ownership verification
 - add randomly generated NS record, based on customer's identity info, at TLD (if using fixed nameservers)
 - check all possible resolution paths for each domain (if randomly assign nameservers)
- For DNS records
 - alert customers to delete current DNS records copied, including NS record for the added domain, during service transfer.
 - TTL values of these NS records could also be decreased and they would be cleared out faster, reducing security risks

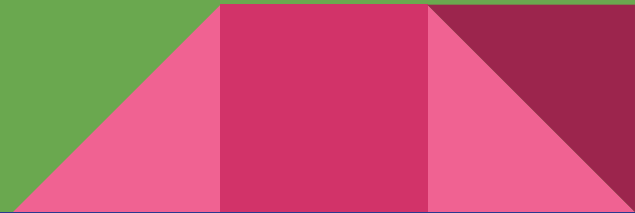


Mitigate cache poisoning

- Improve caching rules to elevate the importance of TLD zone
 - bailiwick rule (protects the cache against false response)
 - credibility rule (when to overwrite records in cache)
- More domains should be properly DNSSEC-signed and more resolvers should be DNSSEC validating



Criticism



Limitations of proposed mitigations

What can the clients (ie. you and me) who initiate the domain query do?

DNSSEC has some downsides



How to overcome such limitations

Clients:

Use Domain Name Servers operated by reputable companies such as Google and avoid less reputable ones. Not just against ZAW attack but also domain hijacking in general. ISPs in some countries have monetary motivation to do that too.



How to overcome such limitations

DNSSEC:

One problem with this protocol is lack of confidentiality (encryption).

Anyone who can monitor your internet traffic can see which domain you queried for via DNSSEC. Huge privacy risks.

Could consider newer protocols like DNS over TLS (DoT) or DNS over HTTPS (DoH) that solves this problem.



Conclusion

Zombie awakening attack, a relatively complex and sneaky way of domain hijacking using stale NS records.

No direct solution but recommendations to mitigate risks instead.

Medium risk - hosting providers can greatly lower risks of attack, if they bother to implement these recommendations and address their limitations



Further readings

Eihal et al. 's paper -

<https://dl-acm-org.ezproxy.auckland.ac.nz/doi/pdf/10.1145/3372297.3417864>

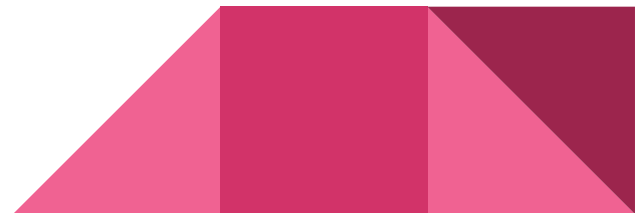
Useful links:

DNS Wars - <https://blog.apnic.net/2019/11/04/dns-wars/>

Benefits of DoH -

<https://bitsup.blogspot.com/2018/05/the-benefits-of-https-for-dns.html>

Can I Take Your Subdomain? - <https://www.usenix.org/system/files/sec21-squarcina.pdf>



Ending

