University of Amsterdam

Thesis for master in Security & Network Engineering

Large-scale Indirect DNS Hijacking Detection Using HTTPS Scan Data

December 20, 2020

Student:
Niels Warnars

Supervisor:
C. Veenman

Abstract

In DNS hijacking attacks, adversaries take control of a targeted organisation's domain at the domain registrar or DNS provider and redirect traffic to an attacker-controlled server. In this research, we investigate whether DNS hijacks can potentially be detected indirectly by attempting to identify the attacker-controlled servers using internet-wide HTTPS scan data. Based on previously reported hijacking incidents, we determined multiple properties that were characteristic of several previous hijacking attacks. Based on these characteristics, we implemented a detection method that filters internet-wide scan data in an attempt to identify the attacker-controlled servers. The research shows that indirect DNS hijacking detection using internet-wide scan data has potential, but is not possible without additional selective filtering. Our implementation yielded too many false positives. Only with additional filtering on potential high-profile domains and more data enrichment, we identified artefacts of likely hijacking attacks against several government websites.

1 Introduction

In recent years, numerous DNS hijacking attacks have been documented in which attackers gained access to the DNS settings of a victim organisation's domain at the domain registrar and redirected traffic to an attacker-controlled (Man-in-the-Middle) server for information harvesting purposes [1] [2] [3] [4] [5].

Detecting this type of attack for individual domains is trivial. It only requires periodic DNS lookups and comparisons with a predefined set of expected answers. This approach has its limitations; It requires knowledge of domains to monitor. Aimé, researcher at Kaspersky Labs, described this as the "main issue" of DNS hijacking monitoring, requiring "[lots] of work, scrapping, scripting" [6].

For threat researchers, it is beneficial to know what organisations become the victim of DNS hijacking attacks on a global scale. Preferably, a more generic method should be available, without too many hardcoded values. In this research, we study whether DNS hijacks can be detected on a large (internet-wide) scale using a completely different approach. In this paper, we propose an indirect DNS hijacking detection method using HTTPS scan data that can identify attacker-controlled servers. The existence of these malicious servers is an indicator of an ongoing hijacking attack.

2 Research questions

This research is focused on indirectly identifying DNS hijacking attacks via the attacker-controlled (MitM) servers using internet-wide HTTPS scan data. The central research question is therefore defined as:

Can DNS hijacking attacks be detected indirectly by identifying the attacker-controlled (MitM) servers using internet-wide HTTPS scan data?

To answer the main research question, it would be desirable to determine the characteristics of previous DNS hijacks. We can use these characteristics to design filter algorithms for a DNS hijack detection system. The resulting sub-questions are therefore formulated as follows:

- What properties characterise previously-documented DNS hijacking attacks?
- How can internet-wide HTTPS scan data be filtered to potentially identify new attacker-controlled (MitM) servers?
- How do different filtering methods compare with regard to coverage?

3 Related work

Reports about previous DNS hijacking incidents can mainly be found in news articles and in industry publications, whereas only limited research into the large scale real-world detection of hijacking attacks is available.

3.1 Previous hijacking incidents

Throughout the years numerous organisations have been the victim of DNS hijacking incidents, in which attackers changed DNS records of the victim's domain at the domain registrar and redirected all traffic to an attacker-controlled (MITM) server.

For example, Fox-IT [1] became the victim of a DNS hijacking attack and subsequently published an analysis of all stages of the attack. During the compromise, traffic to one of the company's portals was routed through a Man-in-the-Middle server, possibly to intercept sensitive information. And also, in 2019, CrowdStrike [2], FireEye [3] and Cisco Talos [4] [5] published reports about dozens of DNS hijacking attacks that among others targeted governmental institutions and IT companies. In these attacks, traffic to a targeted website would also be redirected to an MitM server. The reports provide extensive background information and technical details, though no detection methods were published. It is therefore unknown how CrowdStrike, FireEye and Cisco Talos identified the hijacking attacks.

3.2 DNS hijacking detection

Besides reports about DNS hijacking incidents, several persons performed attempts at (real-time) detection of hijacking attacks. Monitoring for DNS changes of known individual domains is straightforward. For example, in [7] Aimé, researcher at Kaspersky, describes a system that is used to monitor for DNS changes of a predefined list of target domains. For his thesis, Braun [8] attempted to monitor for DNS hijacking cases on a larger scale. For a period of three months in 2015 and 2016, Braun monitored for DNS changes of several thousand (sub)domains belonging to corporations in the aerospace industry. His research provides insights into the results and limitations of large scale monitoring.

4 Background

The idea of our to-be-investigated detection method is based on the existence of multiple servers for a targeted domain at the moment a DNS hijacking attack takes place. To exploit a successful DNS hijack, the adversary will initialise a potentially new internet-facing server that will receive the traffic destined to the targeted domain. If we can detect these malicious servers, that have a valid certificate of a targeted domain, using bulk HTTPS scan data, this could be an indicator of an ongoing DNS hijack.

To prevent immediate detection of an ongoing attack, the malicious servers sometimes only act as an MitM server that intercepts the traffic between the targeted server and the end-user. Because of its stealthiness and impact, we specifically look into this type of hijacking attack.

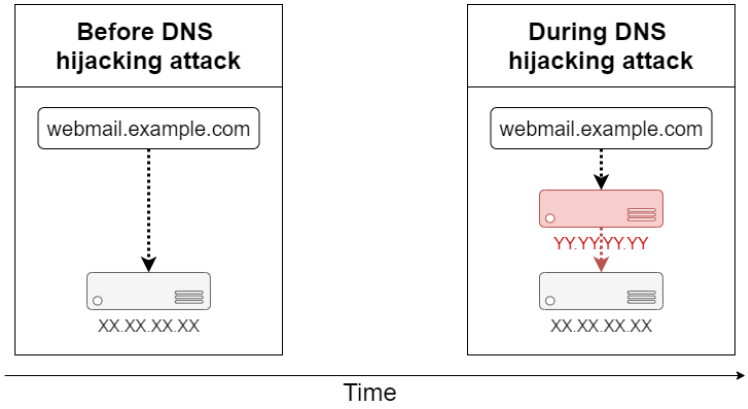

Figure 1: High-level overview of DNS hijacking attack

As a replacement for DNS data, we obtain domain name information from certificates' Subject Alternative Name (SAN) extensions. Observing domain name information in a TLS certificate present on a web server does not guarantee that the domain is resolving to that server; However, it can serve as an indicator that a server administrator had control over the observed domain for which the certificate was issued. This indicator is even stronger for CA-issued certificates. Under normal circumstances, entities that do not have ownership of a domain should not be able to obtain browser-trusted certificates for that domain. In this research, self-signed certificates are therefore deemed unsuitable and are out of scope. Our detection method can thus not identify DNS hijacking attacks that involve attacker-controlled servers or targeted servers with self-signed certificates.

5 Methodology

To determine whether we can detect the malicious servers used in DNS hijacks, the research is divided into three parts; In the first part, we analyse several previously documented DNS hijacking attacks and document the properties that characterised these old attacks. In part two, a multi-layer filtering system is constructed and tested that relies on the properties identified in part one. And in part three, the research will finalise with an attempt at identifying previously unknown hijacking attacks using historic scan data from between January 2018 and November 2020.

5.1 Data sources

We relied on multiple open and (semi-)commercial data sources to perform our analyses and research. In the analysis of historic DNS hijacking incidents, we used several third-party data providers that allow for historic data look-ups. For the construction of the detection system, we used internet-wide HTTPS scan data and IP location data. Table 1 shows the data sources we depended on:

Data provider	**Data type**
crt.sh [9]	Historic certificates
RiskIQ [10]	Historic certificates
RiskIQ [10]	Historic passive DNS
VirusTotal [11]	Historic Passive DNS
Shodan [12]	Historic scan records
Rapid7 [13]	Historic scan records
MaxMind GeoLite2 [14]	IP location information

Table 1: Overview of data sources used in research

5.2 Lab set-up

The experiments require sufficient storage and memory resources due to the size of the scan data sets. Therefore, we parse and pre-process all scan data sets on a storage server and analyse the processed data sets on a high-memory data analysis server. The analysis machine is capable of caching multiple processed scan data sets into memory for parallel processing of multiple scans.

Component	**Resources**
CPU	2x Intel Xeon E5 quad-core
Memory	32GB
Storage (HDD)	24TB

Table 2:
Specifications of storage server

Component	**Resources**
CPU	Intel Xeon E5 hexa-core
Memory	256GB
Storage (SSD)	1TB

Table 3:
Specifications of data analysis server

5.3 Analysis of historic DNS hijacking incidents

In previous reports from CrowdStrike [2] and Cisco Talos [4] [5], dozens of hijacking attacks have been documented. We analyse these incidents using historic passive DNS, HTTPS scan and certificate data from [9] [10] [11] [12] and review the following artifacts related to the targeted and attacker-controlled servers:

1. The autonomous systems
2. The countries of hosting
3. The type of certificates
4. The returned HTTP responses

5.4 Construction of detection system

The detection system relies on the properties identified in the analysis of historic DNS hijacking incidents. Different combinations of filtering steps are implemented and compared to find an optimal trade-off between data reduction and the risk of false negatives. At each

filtering stage, we document statistics about the number of hits. Based on these statistics, we determine the usability of the filtering steps.

The implementation and evaluation of the filtering system relies on Rapid7's [13] internet-wide (sonar.ssl and sonar.https) scans of port 443 from 2020-10-19 and 2020-11-02. Hosts present in the 2020-10-19 dataset serve as the initial base state. Metadata of all base state servers are stored into memory. We store a machine's subject alternative names, HTTP signatures, ASN and country of hosting. Newly initialised servers only present in the 2020-11-02 scan are ran through the detection system, as schematically shown in figure 2. If a new server (for an existing domain) matches certain characteristics, it is flagged as suspicious. In that case, it is potentially a newly initialised attacker-controlled MitM server to be used in a DNS hijack. The adversary starts a new attack server at a certain moment in time. We try to catch this new server instance shortly after it has come online with HTTPS scan data.

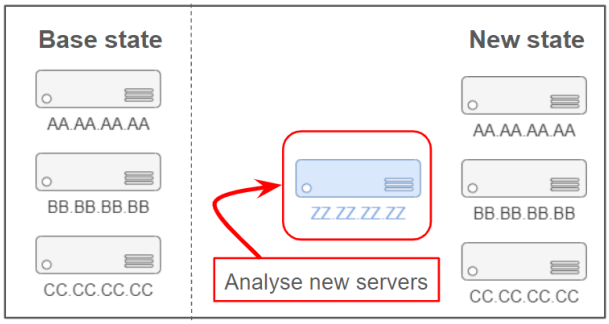

Figure 2: Basis of detection system

5.4.1 Data pre-processing

The detection system uses parsed HTTPS scan data sets. For each IP in a scan we obtain the host's autonomous system, country of hosting, parsed TLS certificate and parsed HTTP response.

We relied on the TLS certificates for obtaining domain name information. To ensure the integrity of the subject alternative names, we can only use CA-issued certificates. Therefore our certificate pre-processing script also determines whether a certificate was likely issued by a browser-trusted Certificate Authority. Our detection method only looks at servers with browser-trusted certificates. Servers with self-signed certificates are ignored.

For each HTTP response in the scan data, we generate multiple hash values. One can use the hash values to construct HTTP signatures that can identify a server. We generate:

- A header hash, derived from the server's HTTP response headers.
- A body hash, derived from the server's HTTP response body.
- A page structure hash, derived from the page title and DOM tree structure in the HTTP response body. This accommodates for changing values in the HTML page, like CSRF tokens.

In case a 301/302 redirect is observed, the body and page structure hashes are set to the hash of the redirection URL. We take into account that the redirection URL may contain changing values like an IP address or unique URL parameters.

5.4.2 Data pre-processing code and tooling

For the research, we parsed over 460 million certificates and 2.4 billion HTTP responses. Given this amount of data and the limited time frame of this research, the preparation phase of this research was dedicated to finding optimal data parsing and processing methods that required a minimum amount of time and computing resources. We determined the most suitable and optimal implementations for multiple data processing components. Table 4 shows for each data component what parsing or lookup implementations were most suitable/optimal for this research:

Data components	Parsing / Lookup implementation
Certificates	ZCertificate [15]
HTTP responses	Custom
HTML pages	Selectolax [16]
Autonomous System	pyasn [17]
Server locality	pyasn with MaxMind GeoLite2 data [14]

Table 4: Overview of data parsing / lookup implementations used in research

5.5 Evaluation by hunting for new hijacking attacks

The research ends with an evaluation of the detection method by hunting for signs of previously unknown hijacking attacks in historic scan data from Rapid7's Project Sonar [13]. We analyse 73 scans from between 2018-01-02 and 2020-11-02. Our implementation will highly likely still create false positives. Therefore, we manually analyse the final hits. To reduce the number of hits we will manually analyse, the search is limited to:

- Suspicious new Outlook servers for existing domains. Outlook is regularly used by organisations as webmail solution and can be a high-value target for a DNS hijack.
- Suspicious new servers for domains belonging to governmental organisations. We filter on domains belonging to government organisations as they are often high-profile targets, and can in certain cases be easily identified by a '.gov.*' top-level domain.

6 Results

We divide the results over three parts; An analysis of previously documented DNS hijacking incidents, the construction of a detection system, and an evaluation of the detection method by hunting for historic hijacking incidents.

6.1 Analysis of historic DNS hijacking incidents

To understand the attributes that characterise DNS hijacking attacks, we studied the properties of 50 hijacking attacks documented in [1] [2] [4] [5] for which enough information was available to analyse. For some other documented hijacking cases, we could not determine what domains were targeted, because no passive DNS or certificate data was available. Additionally, some reported attacker-controlled servers were used as DNS servers and are therefore out of scope. Our focus is on HTTPS MitM servers. With data from RiskIQ [10], VirusTotal [11], Shodan [12] and Certificate Transparency [9] we studied characteristics of the old attacks that can serve as basis for future filtering algorithms.

6.1.1 Hosting locations

The 50 hijacking attacks we analysed made use of 26 unique MitM servers. Most servers were either hosted at large known legitimate hosting providers, or in autonomous systems belonging to smaller relatively unknown parties. No clear pattern exists that one could use in the detection system.

AS Label	AS Number	Count
DigitalOcean, LLC	AS14061	9
Choopa, LLC	AS20473	6
myLoc managed IT AG	AS24961	2
DataShack, LC	AS33387	2
Zemlyaniy Dmitro Leonidovich	AS42159	2
BelCloud Hosting Corporation	AS44901	2
ReliableSite.Net LLC	AS23470	1
Linode, LLC	AS63949	1
KANARTEL	AS33788	1

Table 5:
Autonomous systems of MitM servers (n=26)

Next, we assessed how the autonomous systems and countries of hosting for the malicious servers compare to the location of the targeted servers. In 50 out of 50 cases, the attacker-controlled servers were located in a different autonomous system then the targeted servers. And in 45 of 50 cases, the MitM servers were hosted in a different country than the targeted servers.

6.1.2 SSL Certificates used in DNS hijacking attacks

Our detection system relies on SSL certificates for obtaining domain name information. Therefore, we investigated what type of certificates were used in previous attacks. In at least 40 of the 50 cases, newly issued Domain Validated (DV) certificates were used. These certificates were either used on the MitM servers at the moment of the attacks or were issued in the time frame the DNS compromises occurred. In at least 22 / 50 cases, RiskIQ data [10] shows that stolen certificates were used on the MitM servers.

We consider the usage of compromised SSL certificates on MitM servers out of scope and do not look at these certificates in the construction of the detection system. The usage of stolen certificates means the adversary has already breached the target's defences. Therefore, we assume that an actor can only use new DV certificates.

6.1.3 HTTP responses returned by MitM servers

For each separate hijacking attack, we tried to determine what HTTP responses were returned by the MitM servers at the moment the attacks occurred. In at least 24 out of 50 incidents, the MitM servers exposed a fully cloned/proxied HTTP response that would also be returned by the servers that originally hosted the targeted websites. This included 21 confirmed cases in which both the MitM and targeted server returned a similar response. In three cases the MitM servers returned a response that would likely also be returned by the targeted server.

6.2 Construction of detection system

In part one of the research, we derived several properties from previously documented hijacking incidents [1] [2] [4] [5] that can be incorporated as filters in our detection system. Most notably are:

- The initialisation of new servers
- The issuance of new DV certificates for existing (sub)domains
- New autonomous systems for existing domains
- New countries of hosting for existing domains
- Full proxying / cloning of the HTTP responses

In this section, we study the impact of different filtering methods that can be applied in the detection system. In case multiple filter variants are possible, we make a trade-off between count reduction and the risk of false positives.

6.2.1 Filtering on new servers for existing domains

For the construction of the filtering method, we studied the impact of different data reduction steps. Our base state is an internet-wide HTTPS scan of October 19, 2020. The next scan available is a scan performed on November 2, 2020. Each newly initialised server in the November 2020 scan is deemed suspicious if it has the same characteristics as an MitM server.

Filtering on newly initialised servers yield over 2.6 million hits. Filtering on the next characteristic of servers used in DNS hijacks, new servers with a new DV certificate for existing (sub)domains, reduces the hit count by 95%. The detection system uses filters 1, 2 and 3.

Data set	**Count**
IPs with CA-issued cert present in 2020-10-19 scan	25.529.372
IPs with CA-issued cert present in 2020-11-02 scan	25.665.581

Table 6: Hosts with CA-issued certificates in HTTPS scan data sets

Filter	**Count**
Filter 1: New servers in 2020-11-02 scan	2.612.405
Filter 2: Filter 1 + new CA-issued DV cert	602.648
Filter 3: Filter 2 + existing FQDN - (Takes into account wildcard certificate matches)	114.210

Table 7: Impact of filtering on new servers for existing domains

6.2.2 Filtering on new autonomous systems

Subsequently, we consider the impact of filtering on the next characteristic: New autonomous systems for servers hosting existing domains. One can compare a new server's ASN to ASNs observed under exact matching FQDNs / wildcard domains or to ASNs used for second-level domains. These filtering methods reduce the hit count with 78%, respectively 86%.

For the detection method, we use filter 4b, that compares a new server's ASN to ASNs used for second-level domains. We did not look into the number of false negatives the filter creates. This depends on the size of autonomous systems and the number of servers that

host websites of a given second-level domain. We made a trade-off and chose for the filter with the highest count reduction.

Filter	Count
Filter 4a: Filter 3 + new ASN for existing domain - ASNs under matching FQDN / wildcard domain	24.800
Filter 4b: Filter 3 + new ASN for existing domain - ASNs under second-level domain	14.955

Table 8: Impact of different ASN filtering methods

6.2.3 Filtering on HTTP response

Next, we consider the impact of filtering on HTTP responses. Previously, we determined that HTTP responses are regularly passed on one-to-one by the MitM servers. This includes the full response headers and page body. One can identify (parts of) the HTTP responses using the earlier calculated header, page and page structure hashes. For each HTTP entry, we generate multiple different signatures to evaluate the impact of different HTTP response filtering methods.

As shown in table 9, none of the different filtering methods reduces the hint count by a more than a factor two compared to any other HTTP filter. We use filter 5d in the detection system. This filter is based on the 'headers hash + page structure hash' signature and takes into account both the HTTP response headers and any changing values in the page bodies, like CSRF tokens. Therefore, it is the most suitable filter.

After this filter step, we can only apply selection methods that create higher chances of missed cases.

Filter	Count
Filter 5a: Filter 4b + Page hash	4297
Filter 5b: Filter 4b + Page structure hash	6113
Filter 5c: Filter 4b + Headers hash + page hash	2999
Filter 5d: Filter 4b + Headers hash + page structure hash	4423

Table 9: Impact of different HTTP response filtering methods

6.2.4 Filtering on hosting location

Next, we consider the impact of filtering on a new server's location.

- First, we assess the impact of filtering on the country a new server is located in. Previously, we determined that in 45/50 investigated hijacking cases the MitM server was located in a different country than the targeted server.
- Secondly, we assess the impact of filtering on new servers residing in high-risk autonomous systems. We define a high-risk autonomous system as an AS that has previously hosted infrastructure of one or multiple threat actors. From multiple public and private sources, we sourced 213 small low-reputation ASs and seven large legitimate ASs that have hosted malicious infrastructure of a wide range of adversaries.

As shown in table 10, none of these selection steps reduces the number of hits to less than a few hundred, though the assessed filters do introduce a false negative risk. We generate data sets that are filtered using the filters described in this section.

Filter	Count
Filter 5d + change of country	1363
Filter 5d + high-risk AS	813
Filter 5d + change of country + high-risk AS	343

Table 10: Impact of different location filtering methods

6.2.5 Filtering on Outlook portals

Finally, we assess whether filtering on Outlook webmail portals yields any results. We focus specifically on logon portals, like Outlook webmail environments, as these websites have a potentially higher chance of being targeted, especially if a threat actor wants to perform credential harvesting. Filtering on new Outlook servers, for existing domains, reduces the number of hits significantly. We do not directly implement this filter in our detection routine.

Filter	Count
Filter 5d + Outlook server	53
Filter 5d + Outlook server + change of country	3
Filter 5d + Outlook server + high-risk AS	0
Filter 5d + Outlook server + change of country + high-risk AS	0

Table 11: Impact of filtering on Outlook servers

6.3 Evaluation by hunting for new hijacking attacks

The final part of the research is an evaluation of the filtering system by searching in historic scan data from Rapid7 [13] for indicators of hijacking attacks. The search is limited to:

- Suspicious new Outlook servers for existing domains.
- Suspicious new servers for domains belonging to governmental organisations.

Filtering on new Outlook servers, for existing domains, that were either hosted in new countries or were located in high-risk autonomous systems yielded 233 suspicious hits. Of these hits, we manually selected the servers that were not hosted at large cloud hosting providers and only had one or two subject alternative names. We performed further manual analysis of these servers using RiskIQ [10] but did not identify any hijacked domains.

Filtering on domain names with a '.gov.*' top-level domain yielded three possible hijacking cases against different government organisations.

- Hit one was (inadvertently) picked up by the detection system after the targeted domain was restored. The domain belongs to a Middle Eastern national security agency. The website was always hosted in the home country of the organisation. For four months, the domain resolved to a server in a foreign country. Given the combination of this hosting pattern and the targeted organisation, it is likely, though not completely certain, that the domain was targeted in a previously undocumented DNS hijacking attack.
- Hit two was a previously undocumented MitM server that is almost certainly related to the attacks documented by Cisco Talos [4]. In at least one case, the server was equipped with a certificate that was issued to the adversary. With RiskIQ [10], we determined that the server also used four compromised certificates belonging to four different governmental organisations. No passive DNS data is available that directly confirms the hijacks. However, the presence of the certificates is a strong indicator the server was used as an MitM server in a hijacking attack.
- Hit three was an MitM server that was previously documented by CrowdStrike [2].

7 Discussion

Our research is impacted by false positives, the risk of false negatives and limited visibility.

7.1 False positives and negatives

Our filtering system has difficulties with differentiating between new legitimate and new malicious servers, which results in large amounts of false positives. Only by applying additional restrictive filtering steps, one can reduce the number of hits to a manageable amount. Additional rigorous filtering on its turn can result in false negatives.

7.2 Limited visibility

The research used internet-wide scans from Rapid7 [13] that are only performed every two weeks. In case attacker-controlled servers are not online at the moment of scanning, the filtering system cannot detect the hijacking attack. Additionally, the reliance on scan data for obtaining domain names is limited by whether websites are directly accessible on the web server's IP address. In the case of shared hosting servers, where Server Name Indication (SNI) is used, one cannot use our research method. Also, in case servers use wildcard certificates, it is impossible to derive existing subdomains which reduces visibility.

8 Conclusion

Our research primarily focused on determining whether DNS hijacking attacks can be detected indirectly by identifying the attacker-controlled (MitM) servers using internet-wide HTTPS scan data.

The analysis of previously-reported incidents shows that DNS hijacks have specific characteristics; In all investigated cases, the malicious servers were hosted in a different autonomous system then the targeted servers. In many cases, new browser-trusted DV certificates were issued. And in several incidents, the MitM servers exposed a fully proxied/cloned version of the targeted website to the internet.

Our constructed filtering method uses these characteristics to reduce the number of suspicious servers in the diff between two biweekly internet-wide HTTPS scans back to a few thousand. At several stages, we evaluated whether different filtering methods resulted in drastic improvements. This was not the case. Only after searching in the filtered data sets for targeted governmental domains, we identified artefacts of hijacking attacks against several organisations.

One can thus use internet-wide HTTPS scan data combined with very restrictive filtering to identify artefacts of DNS hijacks. Though, our current implementation is highly unpractical and requires more research to be deployable for real-world detection.

9 Future Work

Our filtering approach has potential, but yields too many false positives to be used as a standalone identification method. Only after filtering on domains belonging to high-profile targets, we identified artefacts of several likely hijacking attacks. Therefore, future research could be performed into the identification and classification of domain names that have a higher-than-average chance of being targeted in DNS hijacks. These can include domains belonging to governmental organisations, NGOs, think tanks, financial institutions, etc.

Additionally, internet-wide HTTPS scan data currently prevents us from observing websites hosted on shared-hosting environments. To bypass this issue, crawl data could be used instead. By crawling all domains instead of all IPs, one can possibly identify more incidents. Certificate Transparency logs or bulk passive DNS data can be used as a starting point.

Moreover, due to the short time frame of the research, we only focused on hijacking attacks in which both the MitM and the targeted servers have CA-issued certificates. This scope creates coverage gaps. Future implementations should work independently of the certificates used in the attacks.

10 Acknowledgements

Research access to scan data was provided by Rapid7.

References

[1] Erik de Jong and Frank Groenewegen. *Fox-IT hit by cyber attack*. Dec. 16, 2017. URL: https://www.fox-it.com/en/news/blog/fox-it-hit-by-cyber-attack/.

[2] Matt Dahl. *Widespread DNS Hijacking Activity Targets Multiple Sectors*. Jan. 25, 2019. URL: https://www.crowdstrike.com/blog/widespread-dns-hijacking-activity-targets-multiple-sectors/.

[3] Muks Hirani, Sarah Jones, and Ben Read. *Global DNS Hijacking Campaign: DNS Record Manipulation at Scale*. Jan. 10, 2019. URL: https://www.fireeye.com/blog/threat-research/2019/01/global-dns-hijacking-campaign-dns-record-manipulation-at-scale.html.

[4] Danny Adamitis et al. *DNS Hijacking Abuses Trust In Core Internet Service*. Apr. 17, 2019. URL: https://blog.talosintelligence.com/2019/04/seaturtle.html.

[5] Danny Adamitis and Paul Rascagneres. *Sea Turtle keeps on swimming, finds new victims, DNS hijacking techniques*. July 9, 2019. URL: https://blog.talosintelligence.com/2019/07/sea-turtle-keeps-on-swimming.html.

[6] Félix Aimé. *Tweet - Main issue of DNS hijacking monitoring*. Jan. 10, 2019. URL: https://twitter.com/felixaime/status/1084361261528293376.

[7] Félix Aimé. *Tweet - DNS change monitoring system*. Jan. 10, 2019. URL: https://twitter.com/felixaime/status/1083425671882440706.

[8] Benjamin Braun. *Investigating DNS Hijacking Through High Frequency Measurements*. UC San Diego, 2016. URL: https://escholarship.org/uc/item/8tm5c7r7.

[9] *crt.sh - CT Log search engine*. URL: https://crt.sh/.

[10] *RiskIQ PassiveTotal*. URL: https://www.riskiq.com/.

[11] *VirusTotal*. URL: https://www.virustotal.com/.

[12] *Shodan - Internet-devices search engine*. URL: https://www.shodan.io/.

[13] *Rapid7 Project Sonar*. URL: https://www.rapid7.com/research/project-sonar/.

[14] *MaxMind GeoLite2*. URL: https://dev.maxmind.com/geoip/geoip2/geolite2/.

[15] *ZCertificate - X509 certificate parser*. URL: https://github.com/zmap/zcertificate.

[16] *Selectolax - HTML5 parser*. URL: https://github.com/rushter/selectolax.

[17] *pyasn - IP to ASN lookup module*. URL: https://github.com/hadiasghari/pyasn.