Passive DNS: A Complete Primer

Posted on February 24, 2022

The Domain Name System (DNS) is essential for the operation of the Internet. It enables the assignment of hostnames to IP addresses: the numerical identifiers of network nodes (computers, cell phones, IoT devices, etc.). As for the detailed description of the Domain Name System we refer to our Domain Name System primer white paper.

The DNS is, by design, a distributed database. Domain names form a hierarchical system of domain levels, e.g., whoisxmlapi.com is in the top-level domain (TLD) .com, the second-level domain is whoisxmlapi.com; there can be many more levels. A **zone**, roughly speaking, is the segment of this system containing data for a given level, e.g., the domains and hosts directly below .com are described in .com's zone file in the form of various **DNS records**. The zone files are stored and queried in a distributed manner in the global Domain Name System in a network of **name servers**. They are maintained on the **authoritative name servers** by the entity responsible for the zone, the **zone administrator**. Notably, the zones are the smallest units of replication in the DNS.

Even though the DNS has a robust design and many measures were applied during its history to maintain its security and reliability, it still has vulnerabilities. Robustness, for instance, is ensured by the replication of data. However, as certain zones have grown huge, replication on the zone file basis has become difficult due to scalability issues, and it is not even possible in many cases without the cooperation of the zone administrator. In addition, zone administration failures and attacks against name servers, resulting in bad data or the unavailability of all the authoritative servers of the zone, and other types of attacks such as domain hijacking can all lead to the malfunctioning of the domain name system. This can, in turn, result in additional disruption of various services.

In addition, the Domain Name System has no memory. It provides data according to its current status, and it is not possible to find, e.g., a domain name assignment that existed, say, a year ago.

This is a serious limitation in many use cases, including recovery after a failure or cybersecurity investigations.

What is passive DNS?

So what is Passive DNS? It was the ingenious invention of Florian Weimer in 2005 to serve as a complementary data source to overcome many of the aforementioned issues. The key idea is to replicate DNS data partially by logging the DNS traffic on the Internet. It can be implemented without any collaboration with zone administrators or active communication with the Domain Name System, hence the name "passive" for the data. The idea was remarkably successful, and today passive DNS data is essential in many cybersecurity and other applications. In the present white paper we give a detailed overview of passive DNS and its applications.

Passive DNS fundamentals

In this section we describe how passive DNS data footprints are generated, what they contain, and how they can be obtained.

Generating of passive DNS data

As mentioned, passive DNS data come from monitoring network traffic, more precisely, name server requests. The logging of the traffic is carried out by the elements of passive DNS architecture called **sensors**. The placement of sensors is an important aspect of the design. It is common that bigger subnets, especially those with a demilitarized zone file inside, operate a **caching name server** to minimize the communication between external name servers and the nodes within the subnet. While a sensor between the clients and the caching name server is of moderate use, a **sensor** logging the caching nameserver's communication with the outside world is especially useful. The most practical information comes, of course, from the communication with authoritative name servers, which is likely to happen here. Also, such big caching name servers are likely to communicate with a diversity of authoritative name servers. Hence, such a sensor placement is even more efficient than placing a sensor close to an actual authoritative name

server unless an extended coverage of the given zone is needed.

According to the architecture description by Weimer, the data collected by the sensor are directed to analyzers that extract the relevant information from the data. The analyzers pass the data to the collector which maintains a single database of all the collected information. This database can then be queried with query processors.

Contents of passive DNS data

The primary goal is to collect data which are normally there in "A" records of the DNS: a domain name and an IP address that it resolves to. (In the case of IPv6, "AAAA" records are to be collected.) In addition to these records, any other record can be collected. Notably: MX (specifying name servers and their priorities), NS (specifying the authoritative name servers), TXT (containing miscellaneous other information), CNAME (aliases from one domain name to another), SOA (zone administrative information), or any other kinds of records can be stored. By an extensive observation of the DNS traffic, the contents of zone files are partially replicated in this way. One can never be sure about the coverage, i.e., how much of the information in the zone files has been collected. In spite of that, such a partial reconstruction of the zone files has significant benefits. It is created independently, without any active communication with the name servers. It is available even if the actual zone file is not revealed directly by the zone administrator. It reflects particular name resolutions, hence, it enables the detection of compromised records.

Essentially, DNS records contain an additional important field which is not there in normal DNS counterparts: a timestamp. This reflects when the record of the given form was last observed. (Some implementations contain an additional "first seen" timestamp that shows when the record was first observed.) This information opens an additional perspective: the possibility of analyzing DNS data in time.

Obtaining passive DNS data

Let us now turn our attention to actual passive DNS data: in what form and how can we get in hold of it.

Forms of passive DNS access

There can be two typical forms of passive DNS data access: passive DNS lookups and entire passive DNS databases.

For instance, if we are to find hostnames that did or do resolve to a given IP along with timestamps, the best option is just to query an existing database: a passive DNS lookup. Weimer's original approach contained a special WHOIS server for this purpose. Currently the popular way to obtain such data is through RESTful APIs.

If the goal is to run complex queries, it might be reasonable to obtain a whole passive DNS database. This requires storage and computing resources, but enables very efficient and highly customized queries as well as data mining applications. There is also flexibility in choosing the database management tool most suitable for the purpose.

Sources of passive DNS data

It is, of course, possible to set up a network of sensors and start collecting passive DNS data. As for the software application, a C implementation by Florian Weimer is available, or one can join the D4 project, "a large-scale distributed sensor network to monitor DDoS and other malicious activities relying on an open and collaborative project" including passive DNS collection, etc. Collecting passive DNS data is mainly an option for those who are in a good position to properly place the sensors and have enough resources.

Another, probably simpler option might be to get access to a passive DNS database. Many IT security providers compete in offering passive DNS services through APIs, and there are some data sources available for download, too. The databases have different contents and coverage. Let us go through the related products of WhoisXML API, Inc.

Passive DNS services of WhoisXML API

WhoisXML API offers a number of products related to passive DNS. These include:

DNS history database

It is WhoisXML API's passive DNS data download product. It is an extensive resource covering 4.2+ billion domains and subdomains, with historical data going as far back as 2008 and 1091 million+ DNS records added weekly. The data contain A, MX, NS, TXT, CNAME, and SOA records. The data come in standard csv format so they can easily be loaded into virtually any software, including relational or semistructured database management systems.

Reverse DNS lookup tool

An interactive web-based tool generates an intuitive and shareable report with all domain names connected to an IP address by looking up the IP address in a passive DNS database.

Reverse DNS API

The same lookups as the Reverse DNS lookup tool, carried out via a RESTful API for programmable processing and software integration. An extended variant is available in the Domain Research Suite, too, which is a part of a pivotable framework for IT investigations.

Subdomains lookup tool

An interactive web-based tool to find all the subdomains of a given domain. Recall that zone files of subdomains are really hard to obtain in many cases. Hence, to find lower-level domains below a domain is a very important use case of passive DNS data. This service offers a specialized query for the goal.

Subdomains lookup API

The RESTful API version of the Subdomains lookup tool.

Domains and Subdomains Discovery

An interactive tool, part of the Domain Research Suite. It enables to find domains and subdomains registered in a given period, containing a given substring. Its extra power is in the possibility of pivoting the search on the results to obtain additional data such as current and historical WHOIS data, website content categorization, etc.

The lookup tools can be tried out without registration. A free plan is available for the APIs too.

Passive DNS applications

In what follows we turn our attention to the main use cases of passive DNS, showcasing some of these using WhoisXML API services.

Subdomain search

A significant amount of malicious activity, especially phishing, uses misleading subdomain names. It is very easy to demonstrate this using the subdomain lookup tool or the API.

As an example, consider the free web hosting provider 000webhost[.]com, which has a questionable reputation: a large number of subdomains under 000webhostapp[.]com appear on PhishTank as valid, verified phishing sites. Why don't we send 000webhostapp.com to the subdomains lookup tool to check its subdomains?

The subdomain lookup API returns 10,000 subdomains at most. Our domain under investigation has more than this, but let's check what we get. Looking for domain names containing the word "pay". At the time of writing this blog (2022.02.18) the API's results contain 10 subdomains of the second-level domain containing the word "pay", and the following 5 of these have really suspicious names:

```
    1 htinstallpayments[.]000webhostapp[.]com
     2 payement-visio[.]000webhostapp[.]com
     3 paylinks[.]000webhostapp[.]com
->   4 transaction-securise-ue-pay[.]000webhostapp[.]com
     5  zellepaycustomerservice-help[.]000webhostapp[.]com
```

And indeed, the one marked one is a confirmed and online phish according to PhishTank at the time of writing. From the passive DNS data we also find that this domain has appeared as early as 2020.09.16. Meanwhile, a screenshot taken with WhoisXML API's screenshot lookup tool shows

that there is only a message at the page that the page is no longer available, so it has probably been removed since. PhishTank does not know about the other ones, yet we believe they deserve some attention.

Our example illustrates that analyzing subdomain lists can be very fruitful. The limitation to 10,000 subdomains can be sufficient for many applications but one can go further by purchasing the entire database and doing the search and, possibly more advanced text mining and other analytics locally.

Another interesting application of subdomains search can be the analysis of a domain by its owner or operator. Finding domains in the passive DNS history that have never been there in the zone file can reveal malicious intrusions to the system, but this leads us to the next use case.

DNS protection and recovery

The Domain Name System itself can benefit from passive DNS data. As mentioned before, the DNS can be a subject of many types of attacks. See our white paper on DNS attacks for a review. Assume, for instance, that both authoritative name servers of a domain get destroyed or compromised. (Being destroyed is an unlikely event as the two authoritative servers should be located at different places and different subnets, but this criterion is not always met.) If the zone operator had no fresh backup of the zone file, the question arises, how to recover it. Passive DNS is an answer: from the passive DNS data one can distill at least partially the lost DNS records.

Putting it another way around, assume that there is a DNS-based attack against the system, e.g., via DNS spoofing. If one has to detect or investigate such a case, passive DNS data is vital. It can, with some probability, enable investigators to recover the actual DNS communications with their whole history. This can indicate if compromised name resolutions take place or took place in the past, revealing their contents, e.g., the IPs where the communication was directed to.

Botnet detection

Botnet detection is a very important and frequently addressed subject of IT security research. It is indeed a central problem as botnets are the main infrastructure of attacks like DDoS attacks, spamming, phishing, or malware distribution. They consist of a "zombie army" of infected

computers, commanded by a central node, the Command & Control (CC) server, to take part in the malicious activity.

The formation of a botnet starts with the exploitation phase of distributing bots through malware that infects nodes on the Internet. Most frequently, the users or owners of these devices are not aware of the infection. The bots enter the rallying phase, where they establish their connection with the CC server. In this phase, the DNS plays a crucial role; the CC servers essentially hide themselves in the DNS behind a tremendous number of domain names and IP addresses. This is typically implemented with the use of numerous domains generated with a Domain Generation Algorithm (DGA) and fast-flux methods. Hence, one of the main directions in fighting botnets is to reveal them through their DNS activity, including domain registrations and DNS communication. A very good review of botnets and DNS-based methods to detect them can be found in a paper by Alieyan et al. (Neural Computing and Applications vol. 28, no. 7 (2017): 1541–58).

Passive DNS certainly became an enabler in botnet detection: as it contains information based on the actual DNS activity along with timestamps, it is very much suited for detecting anomalous DNS activity. The first proposals in this direction were NOTOS by Antonakakis et al. (in: USENIX security symposium, (2010): 273–290. and the EXPLOSURE system by Bilge et al. (ACM Transactions on Information and System Security vol. 16, no. 4. (2014)). Botnet detection has a very broad coverage in the literature and is a subject of still increasing research interest, mainly based on traffic observation and the application of Neural Networks. Passive DNS data is important in almost all such techniques.

Investigations

Passive DNS data is very useful also in IT security (and other) investigations. We illustrate it with two examples, albeit the possible applications are countless.

Find domains related to an IP address

IP addresses, especially IPv4 addresses can be used by many domains. In addition, especially when using them for malicious purposes, they do not have a PTR record in the DNS, that is, a direct DNS lookup of the IP will not return any information.

As an example, consider the IP 164[.]90[.]194[.]36, which took part in a brute-force attack that was conducted against a certain server on 2022.02.20. Even though it is probably an infected machine that is a part of a botnet, it would be interesting to see if it had any associated domain. As expected, a direct lookup will help us little with finding any of these:

```
nslookup 164.90.194.36
server can't find 36.194.90.164.in-addr.arpa: NXDOMAIN
```

If we go for WhoisXML API's passive-DNS-based reverse IP/DNS service instead, we get much farther. The query returns a number of results:

```
account[.]just[.]testing[.]thesyzhack[.]com
account[.]test[.]thesyzhack[.]com
hostmaster[.]thesyzhack[.]com
login[.]just[.]testing[.]thesyzhack[.]com
login[.]test[.]thesyzhack[.]com
outlook[.]just[.]testing[.]thesyzhack[.]com
outlook[.]test[.]thesyzhack[.]com
pentesting[.]syzhack[.]com
thesyzhack[.]com
```

The earliest appearance being the last but one on 4 January 2019, while the others (note the difference in the second-level domain name) appeared around 2021 March. The dates of the last observations start on 21 December 2021, and last till 18 January. This is a definite clue in such an investigation.

Just to make a few steps further: thesyzhack[.]com was created on 2021-01-03 according to its WHOIS data, allegedly by an individual from Dominica. A direct DNS lookup resolves the domain to 127.0.0.1, that is, the standard local loopback interface. All this suggests that this infrastructure is either owned or overtaken by some miscreant. In fact, the IP is listed on numerous blacklists.

Impersonating domains

Here we want to investigate various domains related to a brand. We go for a low-hanging fruit, "paypal" which is often targeted by various scams. We will search for domain names that contain "paypal" as a substring at any level. A possibility would be to go for the Domains and Subdomains Discovery tool of the Domain Research Suite. Here rather go for passive DNS data downloads, especially as we want to allow for misspellings in the search. For the latter, we want to do with a locally running tool.

For illustrative purposes, we download a recent full passive DNS database file, dns_database.2022-02-21.full.csv.gz from the database download service. Its size is 577 megabytes and it contains 83,318,406 records to date. (The weekly files are not independent, so the number of pDNS records offered is higher than this, but it is sufficient for our purposes.)

Next we search for domain names that have "paypal" up to 1 mistypings at any level. For those who want to reproduce the results, let us quote the command-line that does the job:

```
zcat ./dns_database.2022-02-21.full.csv.gz | cut -d"," -f1 \
|tre-agrep -1 paypal | sort -u > paypal_result_full_2022-02-21.txt
```

(In the csv file, the first record is the domain name. The tre-agrep utility is an open-source fuzzy text search tool.) Certainly one can envisage a broad variety of much more sophisticated ways of processing, but this simple example already illustrates the power of having a passive DNS database at hand.

The result set contains 453 results. Amongst these there are many which are almost surely trying to impersonate a valid PayPal page,

```
b[.]paypal[.]com[.]official-support[.]cloudns[.]asia
b[.]paypal[.]com-support-verificatio-account-webapps-ab724a[.]cloudns[.]asi
b[.]paypal[.]com[.]update-information-accounts[.]cloudns[.]asia
n1-login-update-paypal[.]usa[.]cc
```

and we do not have to go far for typosquatting instances either:

```
b[.]stats[.]paypa1[.]authsecr[.]com
b[.]stats[.]paypa1[.]sec2fauthl[.]com
b[.]stats[.]paypa1-ss[.]sec2fauthl[.]com
b[.]stats[.]paypa1-ss[.]sec-rdp[.]com
b[.]stats[.]paypai[.]com[.]iirzeoo[.]xyz
b[.]stats[.]paypal[.]amaozn[.]com[.]de
```

In addition to the names, we can also find out the timestamps and IP addresses. For instance, b[.]paypal[.]com-support-verificatio-account-webapps-ab724a[.]cloudns[.]asia was seen on Fri 07 Jan 2022 04:17:22 PM UTC, and its IP address was 149[.]202[.]249[.]203, a heavily shared IP.

Our examples illustrate also that impersonation of domains and typosquatting often goes on on the level of subdomains. Hence, passive DNS is an essential resource in detecting and fighting them.

Summary

We have provided a brief overview of passive DNS: its notion, technology, and availability. We have also showcased some of the state-of-the-art applications of passive DNS. Our list is certainly not complete. As IT security research and development uses passive DNS heavily, there are many more applications and also new ones are continuously arising.

WhoisXML API provides high quality passive DNS data in the form of downloadable databases and various kinds of lookups, from simple passive DNS lookups to more complex, specialized queries, and pivotable interactive investigative tools. Most of these have a limited free access to give them a try.