# Chapter 1: Introduction

## 1.1 Background Information

To understand the broad picture of the DNS we must go back and understand what the internet is and how it works. The world has arguably become smaller every day. There are many computing devices throughout the world, and the internet (Kurose & Ross, 2013) plays a huge role in connecting all these devices.

Before the Domain Name System (DNS), surfing the internet was tiresome due to lack of scalability. It goes back to 1969 when the Advanced Research Projects Agency Network (ARPANET) was established, also referred to as, ‘the internet’s grandfather’ (*ARPANET | Definition, Map, Cold War, First Message, & History | Britannica*, n.d.). It made use of a huge directory of websites and their corresponding internet protocol (IP) addresses which became harder to maintain by the Network Information Center (NIC).

In 1983, Paul Mockapetris proposed DNS architecture (Metz, n.d.). It automated the management of the internet names and addresses by spreading the duties among servers and allowed the internet to operate without NIC or any other single naming authority. The system ensured that every device on the network knew about every other device. When you typed a domain name, it turned into an IP where the packets should be routed. Every internet service including access to websites, email sent or received, streaming media, social media, and every other internet service, needs DNS to translate relatable domain names, to the IP address related to the service on the internet. For example, when searching the domain name “YouTube”, the DNS translates it to IP address, “172.217.170.174” to access the specific service. Evidently, for the internet’s proper functioning, the DNS plays a huge role. In the process of loading a webpage, four DNS servers are involved (*DNS Server Types | Cloudflare*, n.d.): Recursive resolvers, Root nameservers, Top Level Domain (TLD) nameservers, and authoritative nameservers.

Despite the benefits of the DNS architecture to the internet, security was not a primary consideration in its design. A big challenge remains of how to detect, prevent, and protect the DNS from cyber-attacks and attackers. With this limitation and the continuous technological advancements, it puts the DNS at a great risk to cyber-attacks. There are many vulnerabilities in the network and some common threats include DNS Denial-of-Service, DNS Spoofing, DNS Tunneling, DNS hijacking, and DNS Amplification. It is important to secure and put in measures to counter these cyberattacks on a DNS as it can be a prime target. DNS attacks are also frequently in conjunction with other cyber-attacks to distract from a bigger target (*DNS Security*, n.d.). Some good practices would be regular monitoring of DNS logs to identify irregular traffic patterns and suspicious queries, checking all domain requests against real-time risk scoring threat intelligence, and blocking access to malicious domain and services.

A DNS nameserver and a client are connected through the recursive resolver. The recursive resolver acts as a middleman between a client and a DNS nameserver. It responds to a DNS request from a web client by using cached information, or it will make three requests: one to a root nameserver, one to a TLD nameserver, and one to an authoritative nameserver. The recursive resolver subsequently provides a response to the client after getting a reply from the authoritative nameserver that includes the requested IP address. When it sends a query to an authoritative nameserver, the resolver has no mechanism to confirm the legitimacy of the response. It can only verify that a response seems to originate from the same IP address as the first query. This calls into question the reliability of the DNS answers, though. The focus of the research will be on DNS spoofing attacks.

A forged response to even one of the resolver's queries cannot be easily detected. By faking a response that appears to come from the authoritative server that a resolver initially queries, an attacker can easily assume the identity of that server and route users to potentially malicious websites without them being aware of it. The DNS information is cached by a resolver after receiving a response, which speeds up the request process. When a forged DNS response is sent and the recursive resolver accepts the response and stores it in the cache, the forged response poisons the cache. The recursive resolver then will proceed to return the fraudulent DNS data to other devices within the same DNS.

## 1.2 Problem Statement

DNS Authentication is very vital to cybersecurity, as it verifies DNS records and prevents attackers from redirecting traffic to malicious sites. As a result, the lack of a proper authentication process puts the DNS at a great risk to spoofing attacks among other attacks. Relying on the source IP address from which the resolver transmitted the initial query is not a reliable authentication method as a DNS packet's source IP address is simple to forge.

To curb the issue of lack of strong authentication in a DNS and ensure data integrity and protection from these attacks and attackers, DNS Security Extension (DNSSEC) and DNSCRYPT will be utilized. DNSSEC strengthens authentication in DNS using digital signatures based on public key cryptography. It adds **data origin authentication**, that ensures data came from the actual origin and **data integrity protection**, which ensures that data hasn’t been modified in transit since it was originally signed by the zone owner with the zone’s private key. DNSCRYPT **authenticates** and **encrypts** DNS traffic or the communication between the DNS client and DNS resolver. It uses cryptographic signatures to verify that responses originate from the chosen DNS resolver. It provides double-protection and prevents any eavesdropping on traffic in a network.

## 1.3 Objectives

### 1.3.1 General Objective

To ensure strong authentication and protection of the DNS from DNS Spoofing Attacks using The DNSSEC and DNSCRYPT.

### 1.3.2 Specific Objectives

1. To do an analysis on types of spoofing attacks.
2. To do an analysis on the consequences and impact of spoofing attacks on DNS.
3. To review existing tools and practices used to detect and prevent DNS spoofing attacks.
4. To implement DNSSEC and DNSCRYPT solutions to solve DNS spoofing attacks.
5. To test the performance of the proposed solutions.

## 1.4 Research Question

What is required to be able to strengthen the authentication, detect and prevent spoofing attacks on the DNS ?

1. What are the most common spoofing attacks that the DNS is vulnerable to?
2. What are the consequences and impact on the DNS in the case of these spoofing attacks?
3. What are the existing tools and practices used to detect and prevent DNS spoofing attacks?
4. What are the benefits of utilizing the proposed effective solutions to curb the gaps in detecting and preventing DNS spoofing attacks?
5. Do these proposed solutions curb the problem at hand?

## 1.5 Justification

The Domain Name system is one of the key components for the proper functioning of the internet. However, security was not a consideration in its design. In addition to the continuous technological advancements, it puts the DNS at a great risk to cyber-attacks. In this project, spoofing attacks is taken to consideration and this problem is solved by strengthening the authentication, ensuring data integrity and protection, by utilizing DNSSEC and DNSCRYPT. DNSSEC will strengthen authentication in DNS using digital signatures based on public key cryptography adding **data origin authentication** and **data integrity protection.** DNSCRYPT **authenticates** and **encrypts** DNS traffic between the DNS client and DNS resolver.

## 1.6 Scope and Delimitation

The research only solves the problem of spoofing attacks among other attacks that can occur on the DNS. Looking at the problem at hand, a simulation will be carried out that will mirror a real DNS spoofing attack that could occur on a DNS and together with the solutions utilized, it will all be focused on an Ubuntu Virtual Machine.

## 1.7 Limitations

When a DNS query is sent to a DNS server, there is no information about the DNS client that sent the DNS query. Because of this problem, DNS has become a favorite tool of attackers because the DNS server only sees the IP address that originated the query, which can occasionally be modified by hackers.

The DNSSEC system can be complicated to use because of configuration errors that could cause the DNS servers to lose their security advantages or collectively refuse access to a website.

Along with encryption for DNSCRYPT, DNSSEC utilizes digital signatures based on public key cryptography. Because it takes extra steps to verify DNS records, this increases processing time and network overhead and can slow DNS resolution times. Because signed DNS records are greater in size, this can result in more network traffic and longer download times.