

# **Netzwerkemulation zur Untersuchung von Secure DNS**

An der Fakultät für Informationstechnik der  
Fachhochschule Dortmund im Studiengang

Digitale Technologien

eingereicht

## **Bachelorarbeit**

zur Erlangung des akademischen Grades des  
Bachelor of Science (B.Sc.)

vorgelegt von:	Ano N'dri Jean-Michel, Ekra
Matrikel-Nr.:	90364173
Erstgutachter:	Prof. Dr.-Ing. Ulf Niemeyer.
Zweitgutachter*in:	Prof. Dr. Frank Gustrau
Abgabedatum:	05.06.2024

**Ano N'dri Jean-Michel, Ekra**

## **Thema der Bachelorarbeit**

Netzwerkemulation zur Untersuchung von Secure DNS

## **Stichworte**

DNS, IT-Sicherheit, IT-Grundschutz, DNSSEC, DNSSEC Validierung, OSPF, EIGRP DoT, DoH, Bedrohungsanalyse, Risikoanalyse, DNS-Sicherheit, DNS-Cookies, PKI, CA

## **Kurzzusammenfassung**

Das Ziel dieser Forschungsarbeit ist es, DNSSEC zu untersuchen und in einer eigens entwickelten Emulationsumgebung zu demonstrieren. Dabei werden verschiedene Szenarien nachgebildet, um die Funktionsweise und Wirksamkeit von DNSSEC zu veranschaulichen. DNS-Abfragen und Antworten werden simuliert, um die Implementierung von DNSSEC in realitätsnahen Netzwerksituation zu analysieren. Anschließend wird DNSSEC gegenüber anderen Sicherheitsverfahren wie DNS over TLS, DNS over HTTPS und DANE abgegrenzt und bewertet.

## **Title of the Thesis**

Network Emulation for the Study of Secure DNS

## **Keywords**

DNS, IT-Security, DNSSEC, DNS over TLS, DNS over HTTPS, Cisco GNS3, BIND9, PKI, DNS Security, DNS Delegation, Threat analysis, MAC, MITM

## **Abstract**

The goal of this research work is to investigate DNSSEC and demonstrate it in a specially developed emulation environment. Various scenarios will be replicated to illustrate the functionality and effectiveness of DNSSEC. DNS queries and responses will be simulated to analyze the implementation of DNSSEC in realistic network situations. Subsequently, DNSSEC will be compared and evaluated against other security methods such as DNS over TLS, DNS over HTTPS, and DANE.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>I</b>
<b>Abbildungsverzeichnis .....</b>	<b>III</b>
<b>Tabellenverzeichnis .....</b>	<b>V</b>
<b>Abkürzungsverzeichnis .....</b>	<b>VI</b>
<b>1 Einleitung.....</b>	<b>1</b>
1.1 Motivation.....	1
1.2 Zielstellung.....	2
1.3 Aufbau der Arbeit .....	2
<b>2 Grundlagen.....</b>	<b>3</b>
2.1 IT-Sicherheit .....	3
2.2 DNS.....	4
2.3 DNSSEC .....	9
2.4 DANE.....	10
2.5 Unterschied zwischen OSPF vs. EIGRP Protokolle .....	11
2.6 OSPF Routing Konfiguration.....	12
<b>3 Kryptografische Grundlagen .....</b>	<b>20</b>
3.1 Message Authentication (MAC) .....	20
3.2 Asymmetrie Verschlüsselung.....	21
3.3 Public-Key-Infrastruktur (PKI).....	22
3.3.1 Komponente einer Public-Key- Infrastruktur .....	22
3.3.2 Public-Key .....	22
3.3.3 Privat-Key Kryptographie .....	24
3.3.4 Dignital Signature .....	25
3.3.5 Hashfunktion.....	26
3.3.6 X.509 Certificate.....	27
3.3.7 OpenSSL.....	28
<b>4 Was ist GNS3?.....</b>	<b>29</b>
<b>5 GNS3-basierte DNS-Implementierung/Erprobung .....</b>	<b>32</b>
5.1 DNS Architecture.....	32

5.2 DNS Forwarding/ Delegation Configuration .....	35
<b>6 Beispiel für einen erfolgreichen Angriff auf DNS .....</b>	<b>38</b>
<b>7 Beschreibung der DNSSEC Implementierung/Erprobung.....</b>	<b>41</b>
7.1 DNSSEC-Konfiguration .....	41
7.1.1 Key Signing Key (KSK) generieren .....	46
7.1.2 Zone Signing Key (ZSK) generieren .....	47
7.1.3 Signieren der Zonen selbst.....	48
7.1.4 DNSSEC in Bind9 aktivieren .....	49
7.1.5 DNSSEC Diagnostik und Tools .....	50
7.1.6 Testergebnis und DNSSEC-Validation .....	52
<b>8 Beispiel für erfolgreich abgewerteten Angriff auf DNSSEC .....</b>	<b>57</b>
<b>9 Schlussbetrachtung .....</b>	<b>60</b>
9.1 Diskussion .....	60
9.2 Limitationen .....	61
9.3 Fazit.....	61
9.4 Ausblick .....	62
<b>Literaturverzeichnis .....</b>	<b>63</b>
<b>Anhang.....</b>	<b>69</b>
<b>Erklärung .....</b>	<b>73</b>

# Abbildungsverzeichnis

- Abbildung 1: Hierarchische DNS-Struktur .....	6
- Abbildung 2: Rekursive und Iterative Abfragen bei der Namensauflösung.....	7
- Abbildung 3: OSPF-Topologie in GNS3 erstellt .....	13
- Abbildung 4: Testen der Konnektivität von R1 .....	14
- Abbildung 5: Testen der Konnektivität von R2.....	14
- Abbildung 6: Testen der Konnektivität von R3.....	15
- Abbildung 7: Aktivierung des OSPF-Routings auf R1.....	16
- Abbildung 8: Aktivierung des OSPF-Routings auf R2 .....	16
- Abbildung 9: Aktivierung des OSPF-Routings auf R3.....	16
- Abbildung 10: Überprüfung des OSPF-Routings.....	17
- Abbildung 11: OSPF-Routen anzeigen.....	18
- Abbildung 12: Verbindung Testen zwischen Endgeräte.....	19
- Abbildung 13: Grundlegendes Kryptografisches Konzept.....	27
- Abbildung 14: Erstellung von Container-Images.....	31
- Abbildung 15: Fügen Image zu GNS3 hinzu.....	32
- Abbildung 16: Docker Container Verfügbar in GNS3.....	32
- Abbildung 17: DNS-Topologie erstellt in GNS3.....	33
- Abbildung 18: DNS Forwarding.....	36
- Abbildung 19: DNS-Auflösung ns.pizza.com in Client Maschine.....	37
- Abbildung 20: DNS Zone Delegation ns.pizza.com.....	38
- Abbildung 21: MITM-Angriff auf DNS Request.....	40
- Abbildung 22: Analyse DNS Request mit Wireshark.....	41

- Abbildung 23: DNSSEC Chains of Trust.....	43
- Abbildung 24: DNSSEC Chains of Trust Traversal.....	45
- Abbildung 25: DNSSEC Key Signing Key generieren.....	47
- Abbildung 26: Zone Signing Key (ZSK) generieren.....	48
- Abbildung 27: DNSSEC Zonen Signieren.....	49
- Abbildung 28: DNSSEC Aktivierung.....	50
- Abbildung 29: Domain IP-Adresse mit nslookup.....	51
- Abbildung 30: nslookup im interaktiven Modus.....	52
- Abbildung 31: Dig Domain Name Auflösen.....	53
- Abbildung32: DNS Zone Transfer zwischen Register und Client.....	54
- Abbildung 33: DNSSEC Validation.....	54
- Abbildung 34: Dig Befehl +dnssec Option in GNS3.....	55
- Abbildung 35: Fehler bei DNSSEC Validierung.....	56
- Abbildung 36: DNSSEC Deaktivieren mit +cd Option.....	57
- Abbildung 37: Angriffe Detektion mit Hilfe von DNSSEC.....	59
- Abbildung 38: DNS Paket Capture mit Wireshark.....	60

## **Tabellenverzeichnis**

-	Tabelle 1: Beschreibung des Netzwerkelements.....	35
---	---	----

# Abkürzungsverzeichnis

ad	Authenticator Data
BSI	Bundesamt für Sicherheit und Informationstechnik
CA	Certificat Authority CA
CPS	Certification Practice Statement
CPU	Central Processing Unit
CRLs	Certificate Revokation Lists
DANE	DNS-based Authentication of Named Entities
DKIM	DomainKeys Identified Mail
DNS	Domain-Name-System
DNSSEC	Standard Domain Name System Security Extension
DS	Delegation Signer
EIGRP	Enhanced Interior Gateway Routing Protocol
FQDN	Fully Qualified Domain Name
GNS3	Graphical Network Simulator
HMAC	Keyed-Hash Message Authentication Codes
IANA	Internet Assigned Numbers Authority
ICANN	Internet Corporation for Assigned Names and Numbers
KSK	Key-Signing-Keys
MAC	Message Authentication Code



MITM	Man in the Middle
NIC	Network Information Center
OCSP	Online Certificate Status Protocol
OSPF	Open Shortest Path First
PKI	Public-key-Infrastruktur
PKIX	Public Key Infrastructure for X.509
RA	Registration Authority
RAM	Random Access Memory
RFC	Request for Comments
SLD	Second Level-Domain
SPF	Sender Policy Framework
TCP	Transmission Control Protocol
TLDs	Top-Level-Domains
TLSA	Transport Layer Security Authentication
UDP	User Datagramm Protocol
ZSK	Zone-Signing-Keys
DoT	DNS over TLS
DoH	DNS over HTTPS

# 1 Einleitung

## 1.1 Motivation

Das Internet ist in der letzten Jahrhunderthälfte zu einem grundlegenden und sehr wichtigen Bestandteil unserer Lebenswelt und unseres Alltags geworden. Es ist nahezu unmöglich alle Bereiche und Auswirkungen des Internets auf unser Leben aufzulisten.

In unserem geschäftlichen Alltag sind wir auf das Internet angewiesen um zu kommunizieren. Wir bestellen zunehmend Produkte online, angefangen von Elektronik über Kleidung bis hin zu Lebensmitteln, und finden sogar unseren Lebenspartner online. Das Internet ist zu einem zentralen Werkzeug für die Interaktion zwischen Bürgern und Regierung geworden, und die meisten Länder verlassen sich sogar so weit darauf, dass sie das Internet als eines der Hauptinstrumente für ihre demokratischen Prozesse wie Referenden und Wahlen nutzen.

In diesem Kontext gewinnt die Sicherheit des Internets, insbesondere im Hinblick auf das Domain-Name-System (DNS), kontinuierlich an Bedeutung. Diese enorme Erhöhung an Relevanz macht das DNS zu einem lukrativen Angriffsziel für Cyberattacken. Im jährlich veröffentlichten Global DNS Threat Report von 2020 wurde offenbart, dass 79 % der befragten Unternehmen in der ersten Hälfte des Jahres 2020 einen Angriff auf ihre DNS-Systeme registriert haben (1). Mit der ständig wachsenden Bedrohungslage im Bereich der Cybersecurity wird die Implementierung von Sicherheitsprotokollen, wie sicher DNSSEC (Standard Domain Name System Security Extension), zu einem unverzichtbaren Bestandteil des Schutzes digitaler Kommunikation.

1993 begannen die ersten Sicherheitsbedenken im Umgang mit DNS (2).

Anschließend wurde 1999 der Standard Domain Name System Security Extension (DNSSEC) eingeführt, um die DNS-Daten und Kommunikation sicherer zu machen (3). Die Netzwerkemulation ist eine gewichtete Kombination aus realer Technologie und Simulation. Sie wird genutzt, um Experimente mit Netzmodellen und realen Protokollimplementierungen durchzuführen. Dies erlaubt im Wesentlichen, eine kontrollierte Kommunikationsumgebung zu schaffen und ist von größtem Nutzen für die Herstellung von sicheren Kommunikationswegen, die im Alltag so dringend notwendig sind. Durch den Einsatz von Netzwerkemulation sind wir in der Lage, realistische Szenarien zu

erstellen und deren Auswirkungen auf die Sicherheit von DNS zu analysieren. (4) Dies ermöglicht uns Aussagen über mögliche Gefahren und Sicherheitslücken.?

## 1.2 Zielstellung

Ziel der Arbeit ist es, DNSSEC aufzuarbeiten und in einer aufzubauenden Emulationsumgebung zu demonstrieren. In einer vergleichenden Betrachtung soll anschließend DNSSEC gegenüber anderen Absicherungsverfahren – DNS Over TLS, DNS Over HTTPS, DANE – abgegrenzt und beurteilt werden(1), um seine Verwendung im Vergleich mit anderen Absicherungsverfahren abschließend zu evaluieren.

DNS wurde für diese Arbeit gewählt, da es zu den zentralen und essenziellen Diensten im Internet zählt, dessen Verfügbarkeit für den Großteil der Kommunikation unerlässlich ist. Die Analyse der Kommunikation dieses Dienstes bietet die Möglichkeit, potenzielle Verbesserungen aufzuzeigen. Zusätzlich zeichnet sich die Client-Server-Kommunikation durch ihre Einfachheit aus, da in der Regel lediglich einzelne UDP-Pakete für die Namensauflösung ausgetauscht werden. Diese Simplizität ermöglicht es, die Unterschiede der Kommunikationsmodelle klarer herauszuarbeiten.

## 1.3 Aufbau der Arbeit

Zu Beginn werden die essenziellen Prinzipien des Domain Name Systems (DNS) vermittelt, und es erfolgt eine Einführung in Kryptografische Konzepte, die erforderlich sind, um die Sicherheitserweiterungen des DNS zu verstehen. Danach wird ein Netzwerkaufbau für Tests erstellt. Zusätzlich werden Netzwerkszenarien definiert und untersucht. Hierbei liegt der Fokus auf einer realistischen Strukturierung des Netzwerks, die mehrere kooperierende DNS-Server auf verschiedenen Hierarchieebenen und mehreren Teilnetzen einschließt.

„Basierend auf Docker Container wird dieser Testaufbau realisiert, der die Emulation der zur Demonstration notwendigen Netzelemente und deren Vernetzung ermöglicht.

Zu diesem Zweck soll GNS3 verwendet werden, damit die Emulationsumgebung auf einfache Weise dokumentierbar ist und reproduzierbare Ergebnisse liefert. Zugleich kann sie leicht verändert auch auf andere Untersuchungsgegenstände ausgerichtet werden. Darüber hinaus werden mit der Emulationsumgebung, die zuvor definierten Netzwerkszenarien und Untersuchungsfälle aufgebaut und praktisch erprobt. Anschließend wird DNSSEC mit anderen Absicherungsverfahren verglichen und gegen diese abgegrenzt.

## 2 Grundlagen

### 2.1 IT-Sicherheit

IT-Sicherheit ist der Schutz von Informationen und insbesondere der Verarbeitung von Informationen. Die IT-Sicherheit soll die Manipulation von Daten und Systemen durch unbefugte Dritte verhindern. Dahinter verbirgt sich, dass sozio-technische Systeme, also Menschen und Technik, in Unternehmen/Organisationen und deren Daten vor Schäden und Bedrohungen geschützt werden. Dabei geht es nicht nur um Informationen und Daten, sondern auch um physische Rechenzentren oder Cloud-Dienste (2).

Claudia Eckert definierte in ihrem Werk „IT-Sicherheit“ ein IT-System als ein „geschlossenes oder offenes, dynamisches technisches System mit der Fähigkeit zur Speicherung und Verarbeitung von Informationen“[(6), S.3].

In den Rahmen der IT-Systeme stellen die Informationen und Daten die Werte dar, die mithilfe von IT-Sicherheitsmaßnahmen geschützt werden sollen. Hierbei werden die drei Schutzziele der Informationssicherheit berücksichtigt:

- Wahrung der (Informations-)Vertraulichkeit – Schutz vor unbefugter Erlangung von Informationen
- Gewährleistung der (Daten-) Integrität – Schutz vor unautorisiert Veränderung der Daten
- Sicherstellung der (System-)Verfügbarkeit – Schutz vor Beeinträchtigungen und Unterbrechungen

Diese Schutzziele zeigen uns die Anforderungen an IT-Systeme im Hinblick auf Sicherheit. Dies ermöglicht uns bei der Entwicklung neuer und bei der Evaluation bestehender Systeme diese Schutzziele einzusetzen und uns nach ihnen zu richten. Security Engineering bezeichnet einen strukturierten Ansatz zur Beurteilung der Sicherheit von IT-Systemen. Zunächst werden die zu schützenden Güter identifiziert und dann wird geprüft, welche Risiken für die ermittelten Werte bestehen.

Anschließend erfolgt die Klassifizierung des jeweiligen Risikos anhand von Eintrittswahrscheinlichkeiten und potenzieller Schadenshöhe(4).

Durch verschiedene Wege wird versucht die Risiken mit Hilfe von geeigneten Maßnahmen zu minimieren. Um das Sicherheitsniveau zu behalten, ist es erforderlich, diesen Prozess in regelmäßigen Intervallen zu wiederholen(3). Das Bundesamt für Sicherheit und Informationstechnik (BSI) hat die Methode „IT-Grundschutz“ entwickelt, um die Komplexität bei der Erreichung eines angemessenen Schutzniveaus zu reduzieren. Weiterhin gilt beim IT-Grundschutz, im Vergleich zum klassischen Security Engineering, dass auf Risikoanalyse verzichtet wird und von üblichen Bedrohungen pro System ausgegangen wird. Es wird auch empfohlen, Sicherheitsmaßnahmen entsprechend dem Schutzbedarf zu ergreifen.(5).

## 2.2 DNS

Das „Domain Name System“ ist ein wesentlicher Bestandteil der Internetinfrastruktur. Mit Ausnahme einiger spezifischer Anwendungen (z. B. Peer-to-Peer-Anwendungen usw.) basieren die meisten Internetdienste derzeit auf dem Arbeitsmodell, bei welchem vor den Kommunikationsaktivitäten einige DNS-Abfragen erfolgen. Die DNS kann entweder durch die Vorwärtsauflösung die IP-Adresse eines Hostnamens liefern oder durch die Rückwärtsauflösung den zugehörigen Hostnamen einer IP-Adresse ermitteln(6). DNS wird häufig als „Telefonbuch des Internets bezeichnet“(7). Paul Mockapetris formulierte erstmals im Jahr 1983 die Konzeption von DNS (Domain Name System) (6). Im Jahr 1987 wurde die Spezifikationen für das Domain Name System (DNS) in den Dokumenten „Request for Comments“ (RFC) 1034 und 1035 standardisiert(8,9).

DNS wird in der Form einer verteilten, hierarchischen Datenbank realisiert und kann grafisch als Baumstruktur mit Wurzel dargestellt werden. Die Struktur wird in Abbildung 1 in schematischer Form dargestellt. Die oberste Ebene, die in Gelb dargestellt ist, wird als Root-Zone bezeichnet. In der unteren Ebene befinden sich die Top-Level-Domains (kurz TLD, dargestellt in Blau). Die Top-Level-Domains (TLDs), die ihrerseits die Second-Level-Domains (kurz SLD, in Grün dargestellt) enthalten, werden umgangssprachlich oft einfach als Domains bezeichnet. Unterhalb einer SLD (Second-Level-Domain) können weitere Ebenen existieren, die als Sub-Domains (in Rot dargestellt) bezeichnet werden. Somit bildet eine Domain einen Teilbaum innerhalb der Gesamthierarchie. Ein Blatt dieses Baums repräsentiert einen Host (in Violet dargestellt). Der Name des Hosts wird als Fully Qualified Domain Name (FQDN) bezeichnet und entsteht durch die Verknüpfung

aller übergeordneten Ebenen bis zur Wurzel, wobei zwischen den Knoten jeweils Punkt steht(10–12). Der vollständige Name des Hosts „Users“ lautet demnach „users.FB10.fh-dortmund.de“. im FQDN-Format. Die Wurzelzone (Root-Zone) wird von 13 Nameservern verwaltet, die die Adressen von A.root-servers.net tragen.

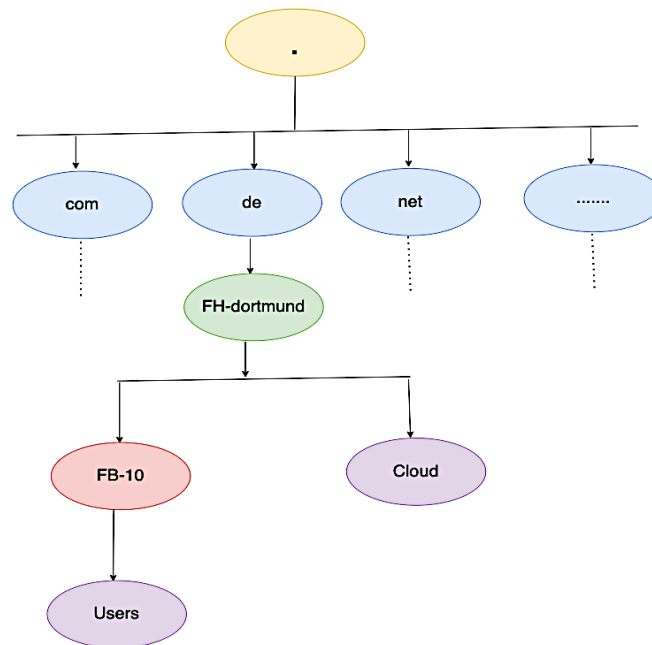


Abbildung 1: Hierarchische DNS-Struktur

Hinter den 13 Nameservern verbergen sich in Wirklichkeit hunderte von Servern, die über verschiedene Standorte weltweit verteilt sind. Diese Server werden mittels Anycast, einer Adressierungstechnik, bei der verschiedene Server dieselbe IP-Adresse teilen, angesprochen. Ein Nameserver ist ein DNS-Server, der die Zuständigkeit für eine spezifische Domain hat (13). Es besteht die Möglichkeit, dass ein Nameserver seine Verantwortung durch Delegation an andere Nameserver weitergibt. Die Internet Assigned Numbers Authority (IANA) verwaltet die Hauptdatenbank in der Verweise auf die maßgeblichen Nameserver jeder Top-Level-Domain (TLD) gespeichert sind. Jede TLD wird von einem Network Information Center (NIC) betrieben, dass für die Verwaltung des entsprechenden Namensraums und den Betrieb der Nameserver verantwortlich ist (11).

DNS ist als Client-Server-Anwendung konzipiert und wird daher im OSI-Referenzmodell der Anwendungsschicht zugeordnet. Zur Kommunikation zwischen Client und Server wird der Port 53 verwendet. In der Regel wird für den DNS-Dienst das

verbindungslose und unzuverlässige UDP als Transportprotokoll verwendet. Bei umfangreichen Antworten besteht jedoch auch die Möglichkeit, auf das verbindungsorientierte und zuverlässige TCP zurückzugreifen(14).

In Abbildung 2 wird kurz erläutert, wie rekursive und iterative Abfragen funktionieren und wie beide Prozesse verschiedene DNS-Probleme lösen.

Der Resolver schickt eine Anfrage für diesen Hostnamen an einen lokalen DNS-Server und verlangt, dass der Server eine Antwort liefert. Durch die DHCP-Option „Domain-Server“ (Option 6 in DHCP und Option 23 in DHCPv6) oder durch Ankündigungen rekursiver DNS-Server, DNS Server (RDNSS) für SLAAC, kann die IP-Adresse des lokalen DNS-Servers manuell angepasst werden. Daraufhin wird dieser DNS-Server versuchen, die Anfrage zu beantworten, indem er die folgenden Bereiche in einer bestimmten Reihenfolge, wie es in Abbildung 2 dargestellt, ansteuert.(15)

Er verweist auf diesen lokalen DNS-Server, an den der Resolver seine Anfrage als rekursiver Server sendet. „Rekursiv“ bedeutet, dass der Resolver DNS-Server bittet, die Antwort auf seine Anfrage zu ermitteln, falls er diese selbst nicht kennt. Aus der Perspektive des Resolvers stellt er eine Anfrage und erwartet eine Antwort(15).

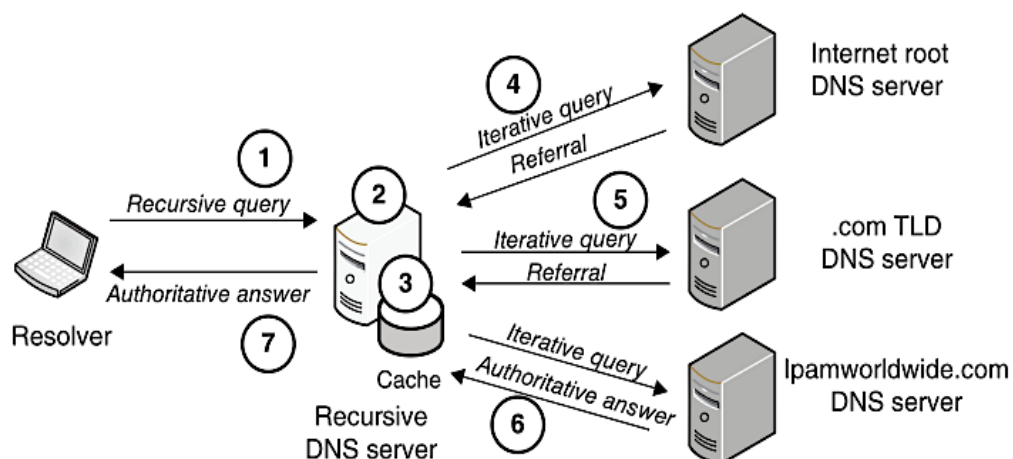


Abbildung 2: Rekursive und Iterative Abfragen bei der Namensauflösung(15)

Aus der Perspektive des rekursiven DNS-Servers kann es notwendig sein, mehrere Anfragen zu stellen, um die Antwort für den Resolver zu finden. Der rekursive Server fungiert als "Portal" des Resolvers in das globale Domain Name System. Der rekursive Server nimmt Anfragen direkt von Client-Resolvern entgegen und führt die folgenden Schritte aus, um die Antwort auf die Anfrage im Namen des Resolvers zu erhalten(15).

1. Der Resolver startet eine Anfrage an den rekursiven DNS-Server.
2. Zuerst durchsucht der rekursive Server seine konfigurierten Zonen Dateien.  
Das bedeutet, dass der DNS-Server normalerweise mit Konfigurations- und Resources Record Information ausgestattet ist, für die er autoritativ ist.
3. Wenn der rekursive Server nicht autoritativ für die angefragte Domain ist, überprüft er seinen Cache, um festzustellen, ob er kürzlich eine Antwort auf dieselbe oder eine ähnliche Anfrage von einem anderen DNS-Server erhalten hat. Befindet sich die Antwort für pc52.dev.ipamworldwide.com im Cache, antwortet der DNS-Server mit dieser nicht autoritativen Information an den Resolver und der Prozess endet. Obwohl dies keine autoritative Antwort ist, hat dies normalerweise keine größeren Konsequenzen. Der Server weist den Resolver jedoch in seiner Antwort darauf hin.
4. Wenn der rekursive DNS-Server die angefragten Informationen nicht in seinem Cache findet, wird er versuchen, diese von einem andern DNS-Server zu beziehen, der die gewünschten, Informationen hat.
5. Bei einer Abfrage an einen Server, der für die Domain Com. autoritativ ist, erhält man als Antwort eine Weiterleitung zum Name Server, der für ipamworldwide.com. autoritativ ist. Dies setzt sich so fort, bis der DNS-Server gefunden wird, der für die angefragten Informationen autoritativ ist.
6. Wenn die abgefragte Information auf einem autoritativen Server verfügbar ist, enthält die Antwort auf die Anfrage einen oder mehrere Resource Records mit dem abgefragten Namen, der abgefragten Klasse und dem Resource Record-Typ. Der rekursive Server aktualisiert in der Regel nicht nur seinen Cache mit der endgültigen Antwort auf die spezifische Anfrage, sondern auch mit weiteren Informationen, die er aus der Antwort und Weiterleitungsmeldungen erhält, die während des Prozesses übermittelt werden. Auf diese Weise speichert der rekursive Server die Domain Namen und IP-Adressen der Domains .com bzw. ipamworldwide.com ab.



7. Der rekursive DNS-Server hat das Ziel, dem Stub-Resolver die Antwort zu übermitteln oder anzuzeigen, dass keine Antwort vorliegt, und damit den Prozess zu beenden.

Eine iterative DNS-Abfrage ist eine Anfrage nach der IP-Adresse eines Domänennamens, die an einen Namensserver (DNS-Resolver) gesendet wird, der mit der relevantesten Antwort antwortet. Diese Antwort kann die IP-Adresse sein, wenn sie im Cache des DNS-Resolvers gespeichert ist. Andernfalls antwortet der DNS-Resolver mit den Daten eines anderen Nameservers. Wie der Begriff "iterativ" schon sagt, wird dieser Verweisungsprozess so lange fortgesetzt, bis der anfragende Server die passende DNS-Antwort erhält. Eine iterative DNS-Abfrage wird auch als "nicht rekursive DNS-Abfrage" bezeichnet, da die Nameserver dem anfragenden Server antworten, anstatt einen anderen Nameserver zu befragen(16).

Bei der Durchführung einer rekursiven DNS-Anfrage sendet der Client seine Anfrage an den relevanten DNS-Server weiter. Wie bei einer iterativen Anfrage überprüft der DNS-Server seine Datensätze und seinen Cache. Wenn er die FQDN auf eine IP-Adresse zurückführen kann, wird er dem Client entsprechend antworten. Wenn es nicht in der Lage ist, das FQDN zu lösen, ist es sehr nützlich; Statt nur mit einer „*I don't know*“ Nachricht zu antworten, sagt sie tatsächlich „*I don't know, aber hier ist die IP-Adresse des root-Domain-Servers. Ihr könnt euch in die richtige Richtung weisen*“. Die Anfrage wird vom Client an den root Domain Server gesendet, der dann den Client in Richtung des TLD-DNS-Servers lenkt. Der TLD-DNS-Server leitet die Anfrage wiederum auf den entsprechenden Domänen DNS Server weiter, der dem Client eine IP-Adresse zur Verfügung stellt oder ihn an den entsprechenden Subdomänen DNS Server weiterleitet(17).

## 2.3 DNSSEC

Das DNS ist ein wesentlicher Bestandteil der kritischen Internetinfrastruktur. Es ist unerlässlich, dass es ordnungsgemäß funktioniert, damit die Internetnutzer es nutzen können. Jedoch wurde das DNS eher für die Verfügbarkeit als die Sicherheit konzipiert. Es verfügt über keinen Authentifizierungsmechanismus und ist derzeit der wichtigste Internet-Angriffsvektor für die Cyberkriminalität. Aufgrund seiner weiten Verbreitung, seiner Flexibilität und seiner wichtigen Rolle im Internet ist das DNS ein äußerst wertvolles Ziel für Online-Datendiebstahl und andere Online-Angriffe(18).

Seit 1997 wird unter dem Namen DNSSEC (Domain Name System Security Extension) an Sicherheitserweiterungen für DNS gearbeitet. Die RFCs 4033 und 4034 sowie das Update von 2013 im RFC 6840 stellen seit 2005 Spezifikationen von DNSSEC dar. Um Angriffe wie DNS-Cache Poisoning zu verhindern, sind die Authentizität und die Integrität von DNS-Einträgen die Schutzziele von DNSSEC. Die Sicherstellung der Vertraulichkeit von DNS-Daten (z. B. durch die Verschlüsselung von IP-Adressen zum Schutz der Privatsphäre) oder die Absicherung gegen Denial-of-Service-Angriffe sind hingegen keine Ziele von DNSSEC. Um die Schutzziele zu erreichen, verwendet DNSSEC asymmetrische Kryptografie sowie kryptografische Hashfunktionen.

DNSSEC zielt darauf ab, DNS-Daten in ihrer Integrität und Authentizität zu schützen, nicht aber auf Vertraulichkeit. Dafür verwendet DNSSEC kryptographische Signaturen. Zwei Schlüsselpaartypen existieren: Key-Signing-Keys (KSK) und Zone-Signing-Keys (ZSK). Der private ZSK wird genutzt, um alle Ressourcen der eigenen Zone zu signieren. Dabei werden Ressourcen in Resource Record Sets zusammengefasst.

Bei Resource Record Sets handelt es sich um sämtliche Resource Records derselben Art. Der öffentliche ZSK wird vom privaten KSK signiert. Der öffentliche Teil des KSK ist der übergeordneten Zone zugänglich zu machen. Der Hash des öffentlichen KSK wird von der übergeordneten Zone mit ihrem eigenen ZSK signiert und zur Verfügung gestellt. Auf dieser entsteht eine Kette von Vertrauen bis in die Wurzel. Der Trust Anchor, welcher als öffentlicher Schlüssel des KSK der Wurzel fungiert, wird auch in den Endsystemen eingestellt(3).

## 2.4 DANE

DANE (DNS-based Authentication of Named Entities) ist die Möglichkeit, eine gesicherte DNS-Infrastruktur zu nutzen, um allgemeine überprüfbare Informationen für eine Mehrfaktorenprüfung zu speichern. Der TLSA-Datensatztyp (Transport Layer Security Authentication) ist heutzutage die am weitesten verbreitete Anwendung von DANE. Mit diesem Typ können Benutzer das PKIX-Zertifikat einer Website überprüfen, indem sie ihre DNS-Informationen abfragen. TLSA wird im RFC 6698 erklärt(19).

Das DANE-Protokoll wurde entwickelt, um DNS zu nutzen, um Zertifikate oder Schlüssel an Domäne Namen zu binden, indem TLSA-Ressourcen Datensätze (RRs) zu Zonen hinzugefügt werden. Durch den Abruf der TLSA-RR des Zertifikats und den Abgleich mit dieser wird die Verifizierung durchgeführt. DANE nutzt DNSSEC zur Gewährleistung der Authentizität und Integrität von DNS-Antworten. Neben TLS-Servern haben TLS-Clients auch die Möglichkeit, TLS RRs zu nutzen und DANE zu überprüfen. Dadurch ist eine gegenseitige Authentifizierung zwischen Clients und Servern möglich (20). DNSSEC wird von DANE verwendet, um die Integrität und Echtheit von DNS-Antworten zu gewährleisten(20).

DANE verbessert insgesamt die Sicherheit und Verlässlichkeit der Internetkommunikation, indem es eine dezentrale und zuverlässige Methode zur Authentifizierung von TLS/SSL-Zertifikaten bereitstellt.

DANE ist ein Protokoll, das nur funktioniert, wenn DNSSEC aktiv ist. DANE lässt den Browser den TLSA-Record nach einem öffentlichen Fingerabdruck eines Zertifikats überprüfen, dass der Benutzer als sicher markiert ist. Möglicherweise handelt es sich hierbei um das Zwischen-Zertifikat der CA, welches das Zertifikat auf dem Server ausgestellt hat, oder um den Fingerabdruck selbst. Ein TLSA-Record kann leicht online erstellt werden, indem man einen Generator wie den TLSA-Generator auf SSL-tools.net nutzt (21). Die DANE-Technologie hat sich auch erfolgreich in der Sicherung von TLS für E-Mails bewährt. Ein E-Mail-Client kann einen E-Mail-Server authentifizieren und die Kommunikation zwischen dem Client und dem Server verschlüsseln. Dieser Prozess funktioniert ähnlich wie bei HTTPS. Ein E-Mail-Client kann den TLSA-Eintrag der Mail-Domain abfragen, um Informationen über die öffentlichen Schlüssel und Zertifikate der Domain zu erhalten, die er zur Herstellung einer TLS Verbindung nutzen kann(15).

## 2.5 Unterschied zwischen OSPF vs. EIGRP Protokolle

EIGRP (Enhanced Interior Gateway Routing Protocol) ist hinsichtlich der Netzwerkkonvergenz, des Routing-Datenverkehrs und der Ethernet-Verzögerung effizienter. EIGRP verbessert die Netzwerkkonvergenz, reduziert den Routing-Protokollverkehr und reduziert die CPU und RAM-Auslastung im Vergleich zu RIP und dem OSPF (Open Shortest Path First) -Protokoll, besitzt die Eigenschaften von Distanzvektor- und Link-State-Protokollen. Im regulären Betrieb benötigt EIGRP kaum Netzwerkressourcen, da ausschließlich „Hallo“-Pakete übermittelt werden. Eine Änderung der Routing-Tabelle bewirkt eine verkürzte Konvergenzzeit und eine Verringerung der Bandbreitennutzung. EIGRP ist ein eigenes Cisco-Protokoll und kann deshalb nicht in einem Netzwerk verwendet werden, in dem ein Nicht-Cisco-Router vorhanden ist(22).

OSPF gehört zu der allgemeinen Kategorie von Routing-Protokollen, die Link State Protokolle genannt werden. OSPF ist ein Interior Gateway Protokoll, was bedeutet, dass es von allen Routern innerhalb desselben autonomen Systems verwendet wird, um Pakete innerhalb des AS weiterzuleiten. Basiert auf dem Dijkstra-Algorithmus für den kürzesten Weg. OSPF Router sammeln Verbindungsstatus-Informationen und verwenden den Algorithmus des kürzesten Weges zur Berechnung des optimalen Weges für die Weiterleitung von Datenpaketen(24).

OSPF wird in großen, heterogenen Internetnetzen eingesetzt. Das Routing-Protokoll unterteilt das Netz in kleinere Segmente, um dieses Problem zu lösen. OSPF unterstützt Mehrfachpfade, bei denen die kürzesten Routen für ein Gebiet schneller neu berechnet werden können, was zu einem minimalen Netzwerk-Routing-Verkehr führt(23).

Das OSPF-Protokoll verwaltet drei Tabellen die Routing-Tabelle, eine Tabelle zur Verfolgung direkt angeschlossener Nachbarn und eine Tabelle zur Verfolgung der Topologie und des Designs des gesamten Netzwerks, die sogenannte Link-State-Datenbank.

OSPF, ein offenes Standardprotokoll, ist in der Lage, große Netzwerke zu managen. Seine Nachteile liegen darin begründet, dass es im Gegensatz zu RIP und EIGRP einen komplexeren Algorithmus verwendet, beim Erstellen der Routing-Tabelle längere Konvergenzzeiten erfordert und somit zusätzlich Protokollverkehr verursacht. OSPF verbraucht eine große Bandbreite für das anfängliche Link State-Packet Flooding und erfordert zusätzliche Speicher und Verarbeitungsanforderungen in einem simulierten Netzwerk mittlerer Größe(22).

## 2.6 OSPF Routing Konfiguration

OSPF ist ein Routing Protokoll. In der Abbildung 3 wird erklärt, wie man das OSPF Routing Protokoll auf Cisco Router konfigurieren kann. Mit GNS3 Netzwerksimulationen werden die OSPF-Konfigurationsschritte erklärt.

Sie haben die Möglichkeit, sämtliche Simulationssoftware für die Konfiguration von OSPF zu nutzen. Die Schritte zur Konfiguration des OSPF sind auf allen Plattformen gleich. Die Konfigurationspraxis von OSPF umfasst die nachfolgenden Schritte(24).

- Erstellung Praxis Labs und IP-Konfiguration an alle Schnittstellen
- Testen der Konnektivität zwischen allen Schnittstellen
- Aktivieren des OSPF-Routings auf allen Routern
- Überprüfen und testen der OSPF-Konfiguration

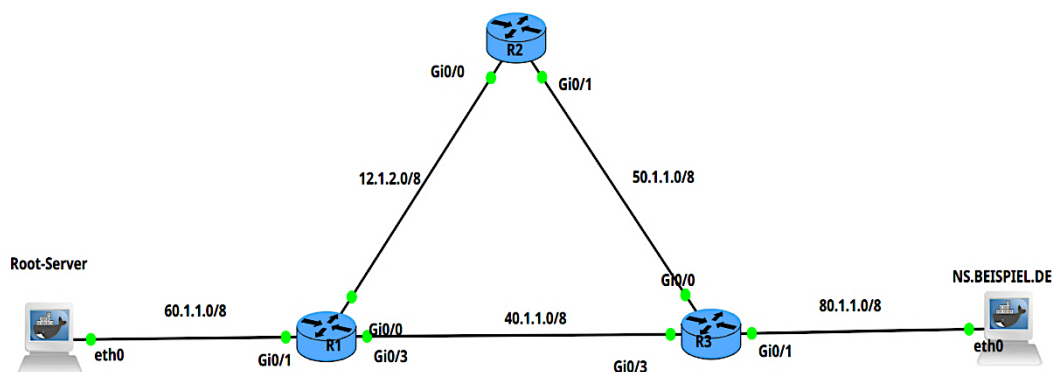
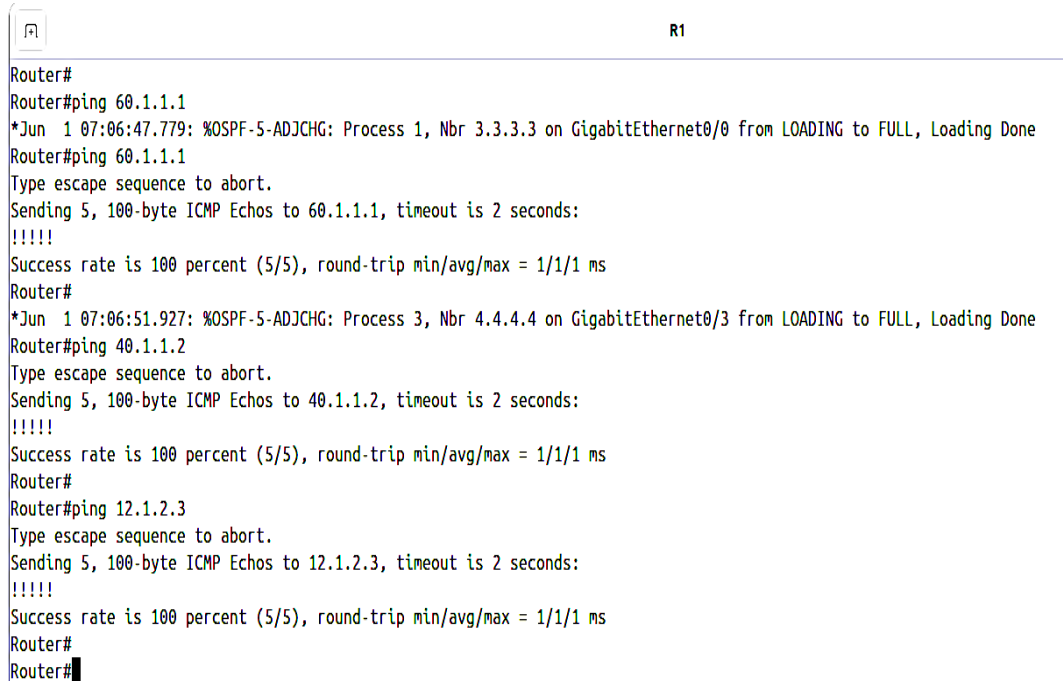


Abbildung 3: OSPF-Topologie in GNS3 erstellt

### - Prüfung der Konnektivität

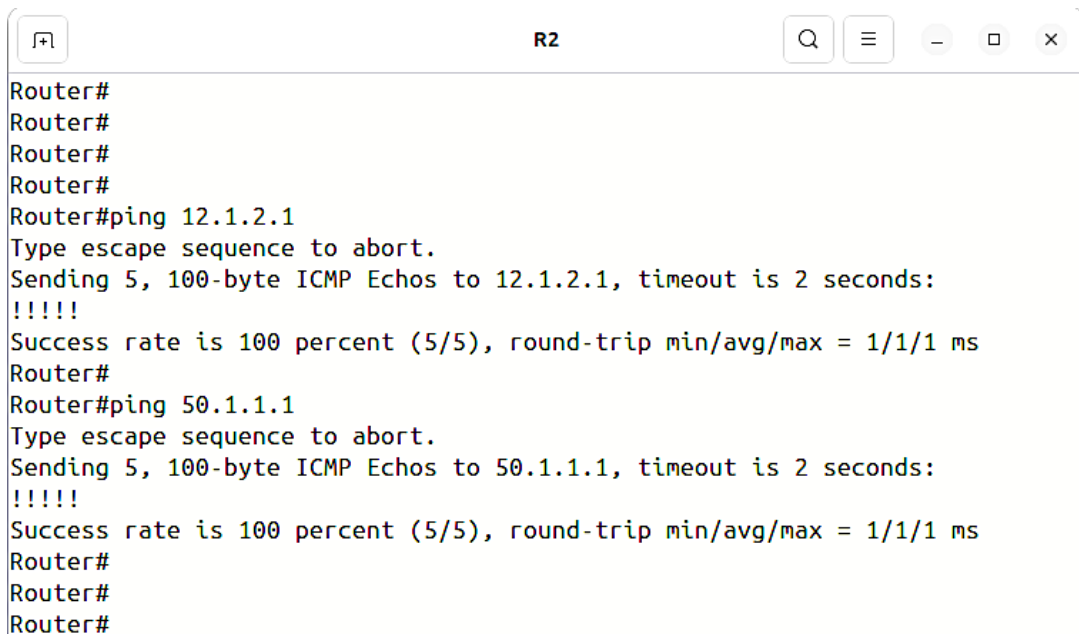
In den Abbildung 4,5 und 6 werden die Konnektivitäten mit dem ping Befehl überprüft und bestätigen dadurch, dass die Geräte miteinander verbunden sind.



R1

```
Router#
Router#ping 60.1.1.1
*Jun  1 07:06:47.779: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
Router#ping 60.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 60.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Router#
*Jun  1 07:06:51.927: %OSPF-5-ADJCHG: Process 3, Nbr 4.4.4.4 on GigabitEthernet0/3 from LOADING to FULL, Loading Done
Router#ping 40.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 40.1.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Router#
Router#ping 12.1.2.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.1.2.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Router#
Router#
```

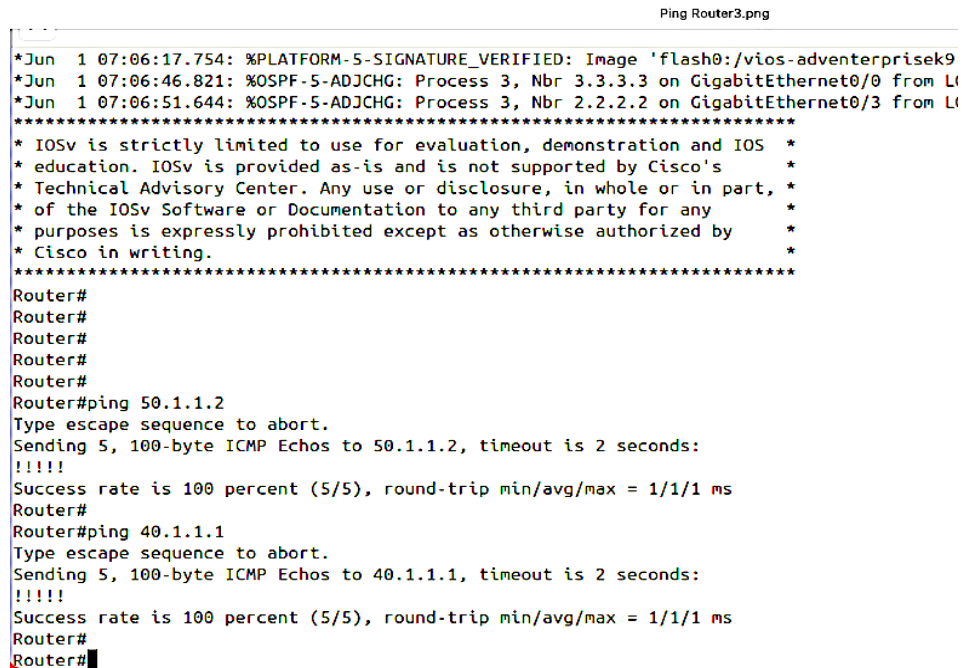
Abbildung 4: Testen der Konnektivität von R1



R2

```
Router#
Router#
Router#
Router#
Router#ping 12.1.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.1.2.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Router#
Router#ping 50.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 50.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Router#
Router#
Router#
```

Abbildung 5: Testen der Konnektivität von R2



```

Ping Router3.png
*Jun  1 07:06:17.754: %PLATFORM-5-SIGNATURE_VERIFIED: Image 'flash0:/vios-adventerprisek9
*Jun  1 07:06:46.821: %OSPF-5-ADJCHG: Process 3, Nbr 3.3.3.3 on GigabitEthernet0/0 from Li
*Jun  1 07:06:51.644: %OSPF-5-ADJCHG: Process 3, Nbr 2.2.2.2 on GigabitEthernet0/3 from Li
*****
* IOSv is strictly limited to use for evaluation, demonstration and IOS *
* education. IOSv is provided as-is and is not supported by Cisco's *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any *
* purposes is expressly prohibited except as otherwise authorized by *
* Cisco in writing. *
*****
Router#
Router#
Router#
Router#
Router#
Router#ping 50.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 50.1.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Router#
Router#ping 40.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 40.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Router#
Router#

```

Abbildung 6: Testen der Konnektivität von R3

OSPF verwendet Areas, um die von den Routern gemeinsam genutzten Routing Informationen zu begrenzen. Es unterteilt Routing Informationen in zwei Typen: detaillierte und zusammengefasste. Router tauschen detaillierte Informationen nur innerhalb desselben OSPF-Gebiets. Sie tauschen nur zusammengefasste Informationen, aber keine detaillierten Informationen aus. OSPF-Bereiche sind schnittstellenspezifisch. Die Schnittstellen eines Routers können in verschiedenen OSPF-Bereichen laufen. Gebiet 0 hat eine besondere Bedeutung. OSPF verwendet ihn als Backbone-Bereich. Alle OSPF-Gebiete müssen sich mit ihm verbinden. Es ist ein Pflichtgebiet(25).

### - OSPF Single Area Konfiguration

Bei der OSPF-Single-Area bleiben alle Router in einem einzigen Bereich. Die Area 0 umfasst eine Single-Area-OSPF Konfiguration(24). Area 0 ist eine spezielle OSPF-Region, die als Backbone dient und andere Bereiche miteinander verbindet.

## - OSPF Routing Aktivieren

In den Abbildung 7,8 und 9 wird gezeigt, wie man OSPF-Protokolle auf dem Routern R1, R2 und R3 konfigurieren kann.

```
Router1>enable
Router1#configure
Router1(config)#router
Router1(config-router)#network 60.1.1.1 0.255.255.255 area 0
Router1(config-router)#network 12.1.2.2 0.255.255.255 area 0
Router1(config-router)#network 40.1.1.1 0.255.255.255 area 0
Router1(config-router)#exit
```

Abbildung 7: Aktivierung des OSPF-Routings auf R1.

```
Router2>enable
Router2#configure
Router2(config)#router
Router2(config-router)#network 50.1.1.2 0.255.255.255 area 0
Router2(config-router)#network 12.1.2.2 0.255.255.255 area 0
Router2(config-router)#exit
Router2(config)#exit
Router2#
```

Abbildung 8: Aktivierung des OSPF-Routings auf R2.

```
Router3>enable
Router3#config
Router3(config)#router
Router3(config-router)#network 50.1.1.2 0.255.255.255 area 0
Router3(config-router)#network 40.1.1.1 0.255.255.255 area 0
Router3(config-router)#network 80.0.0.1 0.255.255.255 area 0
Router3(config-router)#exit
Router3(config)#exit
Router3#
```

Abbildung 9: Aktivierung des OSPF-Routings auf R3.



OSPF gibt nur Nachbarn Routing Informationen weiter. Mit dem Kommando „Show ip ospf neighbor“ überprüfen wir OSPF-Nachbarn. Abbildung 10 zeigt, wie man die OSPF-Routing-Protokolle auf Router R1 überprüfen kann.

```
Router# show ip ospf neighbor.
```

```
Neighbor ID Pri State Dead Time Address Interface
4.4.4.4 1 FULL/DR 00:00:34 40.1.1.1 GigabitEthernet0/3
3.3.3.3 1 FULL/DR 00:00:33 12.1.2.1 GigabitEthernet0/0
```

Abbildung 10: Überprüfung des OSPF-Routings

### **Neighbor ID**

Dieses Feld zeigt den Nachbarn RID an.

### **State**

Dieser Bereich zeigt den Konvergenz Zustand. Konvergenz wird durch sieben Zustände eines OSPF-Routers erreicht. Die Konvergenz des Routers mit dem im Nachbar-ID-Feld aufgeführten Router wird durch den Vollstatus in diesem Feld bescheinigt.

### **Interface**

In diesem Field wird die mit dem Nachbarn verknüpfte lokale Schnittstelle angezeigt.

### **Adresse**

Dieses Feld zeigt die IP-Adresse des Nachbarn.

### **Dead time**

Dieses Feld zeigt das tote Intervall.

Im „Show ip route“ ospf-Kommando werden alle OSPF-routen in der Routing-Tabelle aufgelistet. In der folgenden Abbildung 11 wird mit dem Befehl "Show ip route" die Liste aller OSPF-Router in diesem Netzwerk angezeigt.

```
Router1#
```

```
Router1#show ip ospf route
```

```
OSPF Router with ID (2.2.2.2) (Process ID 1)
```

```
Base Topology (MTID 0)
```

```
Area BACKBONE (0)
```

```
Intra-area Route List
```

- \* 40.1.1.0/24, Intra, cost 1, area 0, Connected  
via 40.1.1.2, GigabitEthernet0/3
- \* 12.1.2.0/24, Intra, cost 1, area 0, Connected  
via 12.1.2.3, GigabitEthernet0/0
- \* 50.1.1.0/24, Intra, cost 2, area 0  
via 12.1.2.1, GigabitEthernet0/0  
via 40.1.1.1, GigabitEthernet0/3
- \* 20.1.1.0/24, Intra, cost 2, area 0  
via 12.1.2.1, GigabitEthernet0/0
- \* 30.1.2.0/24, Intra, cost 2, area 0  
via 12.1.2.1, GigabitEthernet0/0
- \* 45.1.2.0/24, Intra, cost 2, area 0  
via 12.1.2.1, GigabitEthernet0/0
- \* 60.1.1.0/24, Intra, cost 1, area 0, Connected  
via 60.1.1.1, GigabitEthernet0/1

#### *Inter-area Route List*

#### *Intra-area Router Path List*

*I 4.4.4.4 [1] via 40.1.1.1, GigabitEthernet0/3, ABR, Area 0, SPF 4  
Area 1*

#### *Intra-area Route List*

- \* 192.168.0.0/24, Intra, cost 1, area 1, Connected  
via 192.168.0.1, GigabitEthernet0/2
- First Hop Forwarding Gateway Tree*
- 12.1.2.1 on GigabitEthernet0/0, count 4*
- 192.168.0.1 on GigabitEthernet0/2, count 1*
- 40.1.1.1 on GigabitEthernet0/3, count 3*
- 40.1.1.2 on GigabitEthernet0/3, count 1*
- Router1#*

Abbildung 11: OSPF-Routen anzeigen

### Backbone Area (Area 0)

Area Backbone (0) ist ein fundamentaler Bestandteil von OSPF-Routing-Domänen. In OSPF werden Bereiche als logische Gruppierungen von Routern und Netzwerken definiert, die dieselben Routing-Informationen teilen.

### Intra-area Route List

Intra-area Route List bezieht sich auf die Liste der Routen innerhalb eines OSPF-Bereichs, die durch das OSPF-Protokoll erlernt wurden und ausschließlich innerhalb dieses Bereichs gültig sind.

OSPF fügt nur den schnellsten Weg in die Routing Tabelle hinzu, wenn es mehrere Routen zu einer Destination gibt. Die Routing-Tabelle enthält alle Routen, die einen gleichen Preis haben. Er verwendet sie für die Load-Balance. So hat R2 beispielsweise zwei gleiche Streckenpreise für das Netzwerk 40.1.1.0/8. Zwei Strecken werden in die Routing-Tabelle aufgenommen(26).

Wie in Abbildung 12 gezeigt, wird die Verbindung zwischen den Endgeräten getestet, um die OSPF-Konfiguration auf allen Routern zu überprüfen. Hierbei wird Ping-Anfragen vom Alpin-Client an den Root-Server gesendet. Sobald eine Antwort erhalten wird, wird die OSPF-Konfiguration überprüft

```
Router#
Router# ping 20.1.1.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.1.1.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Router#
Router# traceroute 20.1.1.10
Type escape sequence to abort.
Tracing the route to 20.1.1.10
VRF info: (vrf in name/id, vrf out name/id)
 1 12.1.2.1 1 msec 1 msec 1 msec
 2 20.1.1.10 0 msec 0 msec 1 msec
Router#
```

Abbildung 12: Verbindung Testen zwischen Endgeräte

R1 ist das Standard-Gateway für den Alpin-Client Maschine. Es hat zwei Routen, um das Netzwerk 50.1.1.0/8 zu erreichen: direkte Route und via R2. Er leitet Datenpakete über die zweite Route weiter. Die erste Route hat eine serielle und eine Ethernet-Verbindung. Die zweite Route verfügt über drei Ethernet Verbindungen von 64. Eine 100-Mbits/s Ethernet Verbindung mit Standardbreite kostet 1. Die Gesamtkosten der zweiten Route sind 65 (64+1). OSPF wählt die Route aus, die die geringsten Kosten verursacht(26).

### 3 Kryptografische Grundlagen

#### 3.1 Message Authentication (MAC)

Ein Message Authentication Code (MAC) wird verwendet, um die Integrität von Daten zu gewährleisten und Authentizität der Kommunikationspartner zu bestätigen. Diese Codes finden oft Anwendung an Stellen, wo eine Übertragung von Daten in unsichere Netzwerke erforderlich ist und in vielen Sicherheitsprotokollen integriert ist(27).

Ein MAC umfasst eine Hashfunktion sowie einen weiteren geheimen Schlüssel, der von den Kommunikationspartnern bekannt sein sollte. Um sicherzustellen, dass Daten unverfälscht sind, können zwei Kommunikationspartner MACs verwenden, die neben der Nachricht im Klartext gesendet werden. Die geheimen Schlüssel und der Hash der verketteten Nachricht erzeugen diese Macs. Der Sender schickt dem Empfänger die Klartextnachricht sowie den MAC. Er ist in der Lage, mithilfe der Klartextnachricht und des geheimen Schlüssels einen MAC zu erzeugen und die beiden MACs miteinander abzugleichen. Um sicherzustellen, dass die Nachricht unverändert blieb und der andere Kommunikationspartner derjenige ist, der er vorgibt zu sein, sind sie identisch(28).

So kann jeder mit dem gemeinsamen Geheimnis vertraute Kommunikationspartner einen MAC überprüfen und einen neuen MAC für eine Nachricht erstellen. Daher ist ein MAC nicht in der Lage, die Verbindlichkeit zu gewährleisten, da bei mehr als zwei Kommunikationspartnern jeder die Nachricht und den dazugehörigen MAC erstellen könnte (28).

Keyed-Hash Message Authentication Codes (HMAC) stellen eine Ergänzung zu den MACs dar. Sie nutzen zwei Hashfunktionen zur Generierung. Hierbei erfolgt eine Aufteilung des geheimen Schlüssels in einen äußeren und einen inneren. Im Klartext wird der innere Schlüssel mit der Nachricht verknüpft, wodurch der Hash entsteht. Der entstandene Hash wird erneut mit dem äußeren Schlüssel verknüpft und dann wieder gehasht. Dadurch wird der letzte HMAC geschaffen (3).

MACs werden in vielen kryptographischen Protokollen und Anwendungen verwendet, wie z.B. in der sicheren Datenübertragung (SSL/TLS), in digitalen Signaturen und in der Nachrichtenauthentifizierung in verschiedenen Kommunikationsprotokollen.

## 3.2 Asymmetrie Verschlüsselung

Bei der asymmetrischen Verschlüsselung ist kein gemeinsames Geheimnis zwischen zwei Kommunikationspartnern erforderlich, im Unterschied zur symmetrischen Verschlüsselung. Stattdessen hat jeder Teilnehmer sowohl einen öffentlichen und als auch einen persönlichen Schlüssel, während der persönliche Schlüssel geheim gehalten wird. Der öffentliche Schlüssel erlaubt die Verschlüsselung einer Nachricht, deren Entschlüsselung nur mit dem persönlichen Schlüssel möglich ist (3). Dadurch wird gewährleistet, dass nur derjenige, der den privaten Schlüssel besitzt, die Nachricht entschlüsseln kann. Die asymmetrischen Methoden eignen sich auch zur Datensignierung. Beim Unterzeichnen von Nachrichten verwendet der Sender im Gegensatz zur Verschlüsselung seinen privaten Schlüssel zur Verschlüsselung. Durch das Entschlüsseln des öffentlichen Schlüssels des Senders kann der Empfänger prüfen, ob die Nachricht wirklich vom Sender stammt (3). Daher ist für die Entschlüsselung immer der Schlüssel erforderlich, der nicht zur Durchführung der Verschlüsselung verwendet wurde.

Asymmetrische Verschlüsselung basiert auf mathematischen Konzepten, wie Einwegfunktionen, die einfach zu berechnen sind, aber mit keinem effizienten Verfahren umkehrbar sind. Ein mögliches Beispiel für eine Einwegfunktion ist die Multiplikation zweier großer Primzahlen. Während die Berechnung des Produkts einfach ist, ist es äußerst schwierig, die beiden Primzahlen aus dem Produkt zu bestimmen. Für legitime Kommunikationspartner muss die Entschlüsselung jedoch effizient möglich sein. Daher werden Funktionen verwendet, die eine effiziente Entschlüsselung ermöglichen, wenn zusätzliche Informationen bekannt sind (Einwegfunktionen mit Falltür). Ohne diese zusätzlichen Informationen verhalten sich diese Funktionen wie normale Einwegfunktionen (3).

Das RSA-Verfahren dient als Musterbeispiel für eine solche Einwegfunktion mit Falltür. RSA stammt von den drei Wissenschaftlern Ronald Rivest, Adi Shamir und Leonard Adleman aus dem Jahr 1978 und wird heutzutage als praktischer Standard für die asymmetrische Verschlüsselung und Signierung betrachtet (4). Hier soll nicht auf die genaue Funktionsweise von RSA eingegangen werden. Diese kann in (3) nachgelesen werden. Asymmetrische Verfahren sind im Vergleich zu symmetrischen Verfahren deutlich langsamer und werden deshalb häufig zum Schlüsselaustausch eines symmetrischen Verfahrens eingesetzt(4).

### 3.3 Public-Key-Infrastruktur (PKI)

#### 3.3.1 Komponente einer Public-Key- Infrastruktur

CA (Certificat Authority) bestätigen Organisationen, Personen und Geräte durch die Ausgabe digitaler Zertifikate. Diese Zertifikate dienen dazu, Transaktionen zu entschlüsseln, Informationen zu schützen und eine sichere Kommunikation zu ermöglichen(29).

Die Registrierungsinstanz RA (Registration Authority) garantiert den Zusammenhang zwischen dem öffentlichen Schlüssel und den Identitäten und Merkmalen der Zertifikatsinhaber. Oft werden die CA- und RA Komponenten zu einem Trust Center kombiniert (3). Policy, die Vorschriften für die Ausgabe und Verwaltung von Policy-Zertifikaten des Trust Centers, sind in einer Zertifikatsrichtlinie, einem Certification Practice Statement (CPS), festgelegt. Eine solche Politik umfasst unter anderem rechtliche und finanzielle Bedingungen, die Art und Weise, wie Zertifikatsinhaber im Trust Center authentifiziert werden, sowie die Schutzkriterien des Trust Centers (29).

CRL-Zertifikate, die zurückgezogen wurden, werden in einer Sperrliste (CRL) verwaltet. Die CRL enthält die Nummern der Zertifikate, deren Gültigkeit vor Ablauf abgelaufen ist. Oftmals wird auch ein Protokoll zur Zertifikatsstatusabfrage als Alternative zur Sperrliste verwendet. Das OCSP, das Online Certificate Status Protocol, gilt als gängiges Protokoll dafür (29).

Verzeichnis: Zertifikate und Sperrlisten, die ausgestellt werden, werden in einem Verzeichnis gespeichert und bereitgestellt (29).

#### 3.3.2 Public-Key

Die Verwaltung des öffentlichen Schlüssels ist ein Problem in asymmetrischen Verfahrenen. Der öffentliche Schlüssel kann zunächst von einem Angreifer ersetzt werden, der den Austausch zwischen zwei Teilnehmern abfängt. Keiner der beiden Teilnehmer ist in der Lage festzustellen, ob der erhaltene öffentliche Schlüssel tatsächlich der wahre Schlüssel eines Angreifers ist. Das Ziel der Public-key-Infrastruktur (PKI) besteht darin, die Echtheit des öffentlichen Schlüssels zu gewährleisten (4).

Zertifikate, die von einer zuverlässigen Stelle ausgestellt werden, bilden den Kern einer PKI. Sie beglaubigen auch die Zuordnung eines öffentlichen Schlüssels an einen Inhaber. Für diesen Zweck wird die Option der Signierung mithilfe asymmetrischer Verfahren

genutzt. Ein Zertifikat umfasst nicht nur den öffentlichen Schlüssel, sondern auch weitere Angaben wie das Ablaufdatum und den Namen des Inhabers, sowie eine Unterschrift mit einem privaten Schlüssel (4).

Eine PKI kann mit unterschiedlichen Vertrauensmodellen betrieben werden. Jedoch ist das hierarchische Vertrauensmodell, das im Folgenden präsentiert wird, das am weitesten verbreitete Modell. Das hierarchische Vertrauensmodell ist eine Struktur von Zertifikaten, mit der eine Kette von Zertifizierungen aufgebaut wird (4).

Eine CA (Zertifizierungsstelle) verwaltet das Wurzelzertifikat einer Zertifizierungskette, das als Vertrauensanker (Trust Anchor) bekannt ist. Das Wurzel-Zertifikat enthält den CA Public Key und wird mit seinem privaten Schlüssel signiert.

Das Zertifikat muss den Endsystemen bekannt gemacht werden, weshalb die Wurzel eine Besonderheit darstellt. Von der Wurzel aus können zusätzliche Zertifikate signiert werden. Diese können wiederum zur Signierung weiterer Zertifikate verwendet werden(4).

Es besteht die Möglichkeit, Zertifikate zu gefährden, z.B. wenn der private Schlüssel des Schlüsselpaars entwendet wird. Die Aussteller von Zertifikaten führen daher Sperrlisten (CRL) aus. Die Sperrlisten enthalten die vor Ablauf ihres Geltungsbeginns gesperrten Zertifikate (4). Der private Schlüssel der Wurzel und der Knoten, die eine Kante zur Wurzel haben, sind aufgrund der hierarchischen Baumstruktur besonders gefährdet. Aus diesem Grund weisen sie einen hohen Schutzbedarf auf (28).

Zwei Benutzer können ihre öffentlichen Schlüssel und die dazugehörige Zertifikatskette austauschen, die alle Zertifikate bis zur Wurzel enthält, wenn sie miteinander kommunizieren wollen. Bei jedem Zertifikat sind der öffentliche Schlüssel und die Signatur enthalten. Diese wurden mithilfe des privaten Schlüssels des Zertifikats darüber erzeugt und können daher mit dem öffentlichen Schlüssel desselben Zertifikats überprüft werden. Jeder Kommunikationspartner kann die Vertrauenskette bis zur Wurzel durchlaufen und überprüfen, ob die Signatur tatsächlich vom übergeordneten Zertifikat stammt. Da die Endsysteme die Vertrauensanker konfiguriert haben und diesen vertrauen, verfügen sie bereits über den zugehörigen öffentlichen Schlüssel der Wurzel und können somit die gesamte Kette validieren(30).



### 3.3.3 Privat-Key Kryptographie

Ein privater Schlüssel, auch bekannt als geheimer Schlüssel, ist eine in der Kryptographie verwendete Variable, die mit einem Algorithmus zur Verschlüsselung und Entschlüsselung von Daten genutzt wird. Geheime Schlüssel sollten nur mit dem Erzeuger des Schlüssels oder mit Parteien geteilt werden, die die Daten entschlüsseln können. Private Schlüssel sind sowohl in der symmetrischen Kryptographie als auch in Krypto-Währungen von entscheidender Bedeutung (31).

Typischerweise handelt es sich bei einem privaten Schlüssel um eine lange, Random- oder pseudo-Random generierte Reihe von Bits, die nicht leicht zu erraten ist. Die Komplexität und Länge des privaten Schlüssels bestimmen, wie leicht ein Angreifer einen Brute-Force-Angriff durchführen kann. Dabei versuchen sie verschiedene Schlüssel auszuprobieren, bis die Richtige gefunden ist (31).

Private Key Encryption bietet verschiedene nützliche Funktionen. Dazu gehören folgende vier Vorteile (32):

1. Noch sicherer. Private Schlüssel mit größerer Länge und höherer Entropie bieten besseren Schutz gegen Brute-Force- und Wörterbuch-Angriffe.
2. Schneller. Symmetrische Schlüssel-Encryption ist aufgrund ihrer publik privaten Schlüssel-Pairs schneller im Computing als asymmetrische.
3. Am besten für die Encryption. Für die Verschlüsselung von Datenübertragungen verwenden die meisten kryptografischen Verfahren private Schlüssel-Encryption. Typischerweise nutzen sie einen öffentlichen Schlüsselalgorithmus, um geheime Schlüssel sicher zu teilen.
4. Secret key ciphers, ist ein Algorithmus zur Entschlüsselung von Daten. Hierfür werden diese in der Regel in eine von zwei Kategorien eingeteilt: stream ciphers, oder block Cipher. Bei einem block Cipher wird ein privater Schlüssel und ein Algorithmus zur gleichen Zeit auf einen Datensatz angewendet(33).

### 3.3.4 Digital Signature

Digitale Signaturen sind eine spezielle technische Umsetzung einer elektronischen Signatur (eSignature). Elektronische Signaturen werden für Workflow Prozesse genutzt, zum Beispiel zur Anwendung einer Signatur auf einem Dokument. Sie bestätigen nicht die Echtheit eines Dokuments, einer Datei oder einer Software. Doch eine digitale Signatur einer vertrauten dritten Partei, einer Zertifikatsbehörde, um die Identität des Benutzers zu überprüfen und diese an ein PKI-gestütztes digitales Zertifikat zu binden. Daher stellt eine digitale Signatur eine Garantie dar (34);

- **Herkunft** - wer hat das Dokument signiert
- **Zeit** - das Datum und die Zeit Signatur
- **Integrität** - Nachweis, dass das Dokument nicht verfälscht oder geändert wurde
- **Non-repudiation** - Der Sender kann nicht verneinen, dass er das Dokument als die kryptographischen Komponenten unterzeichnet hat, die ein digital signiertes Dokument nicht replizieren oder ändern können

Digitale Signaturen können an allen Stellen verwendet werden, an denen eine Unterzeichneridentität und eine Dokument Integrität gewährleistet werden müssen. Sie sind in vielen verschiedenen Umgebungen mit unbeschränkter geografischer Reichweite verwendbar. Viele andere Branchen, die auf Techniken zur Erkennung von Fälschungen oder Manipulationen angewiesen sind, verwenden sie, darunter Finanzdienstleister, Softwarevertrieb, Fertigung und Gesundheitswesen (34).

### 3.3.5 Hashfunktion

Hashes sind ausgefallene mathematische Berechnungen, die in vielen Bereichen der Datenverarbeitung verwendet werden.

Eine kryptografische Hashfunktion ist wie ein mathematischer Zauberspruch, der aus beliebigen Eingabedaten (wie einer Nachricht) einen festen Ausgabewert generiert, der als Hash oder Digest bekannt ist. Der Hash ist eine Art Fingerabdruck der ursprünglichen Nachricht, der in der Regel eine feste Größe hat, unabhängig von der Länge der Eingaben. Wenn man nur den Hashwert hat, ist es nahezu unmöglich, die ursprüngliche Nachricht zu rekonstruieren(35).

Die Abbildung 14 bezeichnet die Eingabe als Nachricht und die Ausgabe als Hashwert (35).

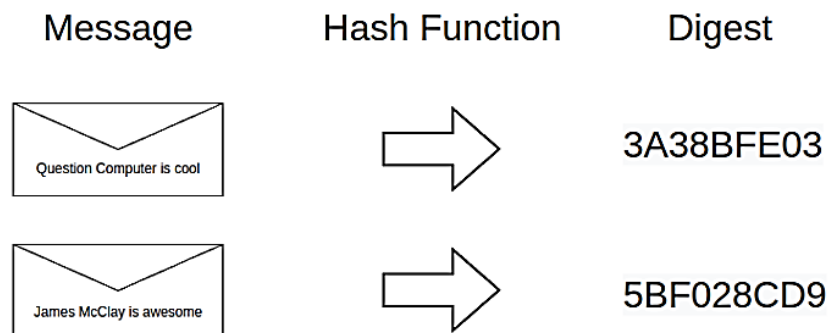


Abbildung 13: Grundlegendes Kryptografisches Konzept (35)

Hashes sind in der Regel länger als das obige Beispiel, und die Nachricht ist normalerweise nicht ein Selbstlob. Aber dieses Beispiel dient dazu, zu veranschaulichen, wie ein Hash funktioniert. Aus diesem Diagramm sollten hoffentlich zwei weitere Punkte deutlich werden(36):

- Eine Nachricht kann jede Art von digitaler Eingabe sein, z. B. eine Datei, ein Zertifikat, ein Zeichen, ein Bild, eine E-Mail, einfach alles.
- Digest ist binär, immer gleich, immer gleich lang und wird in der Regel in hexadezimaler Form dargestellt.

### 3.3.6 X.509 Certificate

X.509-Zertifikate sind entscheidend für das Funktionieren von TLS und TLS-basierten Protokolls, wie zum Beispiel HTTPS, bei denen Zertifikate verwendet werden, um die Identitäten von Websites zu belegen. So werden also Zertifikate in sicheren Messaging-Standards wie S/MIME, VPN-Lösungen wie OpenVPN, Smart Cards, Software-Signing usw. eingesetzt. X.509 -Zertifikate können also optional in IPSEC verwendet werden.

Ein X.509-Zertifikat ist eine Datenstruktur, die zur Identitätspräsentation und Identitätsprüfung verwendet wird. Das Zertifikat kann in einer Datei gespeichert oder über ein Netzwerk übertragen werden, also als Bestandteil eines sicheren Netzwerkprotokolls wie TLS. An X.509-Zertifikat bindet eine Identität mit einer digitalen Signatur an einen öffentlichen Schlüssel (37).

Jedes X.509-Zertifikat hat den entsprechenden privaten Schlüssel. Der private Schlüssel ist nicht im Zertifikat enthalten, entspricht aber dem öffentlichen Schlüssel, der im Zertifikat enthalten ist. So kann eine digital signierte Signatur, die mit dem privaten Schlüssel erstellt wird, mit dem öffentlichen Schlüssel aus dem Zertifikat überprüft werden. Dieser private Schlüssel wird oft als „Zertifikats-Privatschlüssel“ bezeichnet. Der Besitzer des Zertifikats benötigt den privaten Schlüssel des Zertifikats, um zu belegen, dass er das Zertifikat wirklich besitzt und es nicht nur kopiert hat. Dieser Nachweis kann durch die Unterzeichnung einiger Daten mit dem privaten Schlüssel durch den Eigentümer und die Überprüfung der Unterschrift mit dem öffentlichen Schlüssel aus dem Zertifikat erbracht werden (38).

In der Regel ist ein X.509-Zertifikat keine geheime Information und kann kostenlos verbreitet werden, ähnlich wie ein öffentlicher Schlüssel. Der private Schlüssel des Zertifikats ist im Gegensatz dazu eine geheime Information, die nur der Zertifikatsinhaber besitzen sollte. Denn wer auch immer den privaten Schlüssel besitzt, kann die Identität des Zertifikats Inhabers geltend machen und eine solche Identität kryptografisch nachweisen. Der Diebstahl des Privatschlüssels kann zu einem Identitätsdiebstahl führen. So könnte beispielsweise jemand den privaten Schlüssel des [www.openssl.org](http://www.openssl.org) Zertifikats stehlen und dadurch einen DNS Poisoning oder Man in the Middle (MITM)-Angriff ausführen, der eine falsch [www.openssl.org](http://www.openssl.org) Website erstellen würde, die die Zertifikatsprüfung überwinden soll. Es gibt verschiedene Möglichkeiten, die Identitätsverletzung zu imitieren, wie zum Beispiel Certificate Revocation Lists (CRLs) und Online Certificate Status Protocol (OCSP), aber ein Angreifer kann immer noch ein Zeitfenster für seine Angriffe haben, bevor das verletzte Zertifikat widerrufen wird. Ein weiteres Beispiel: Wenn der

private Schlüssel eines Zertifikats zum Signieren von E-Mail-Nachrichten verwendet wird, kann jeder der den privaten Schlüssel stiehlt, E-Mails im Namen des Zertifikatsinhabers signieren (37).

### 3.3.7 OpenSSL

OpenSSL ist ein Open-Source-Software Toolkit, das eine Kryptographie und SSL/TLS Bibliothek sowie Command-Line-Utilities umfasst, die die Bibliothek nutzen, um auf der Kommandozeile nützliche Funktionen wie die Generierung von Zertifikaten und X.509 Schlüsseln anzubieten. Die wesentliche Komponente von OpenSSL ist seine Bibliothek. Das bedeutet, dass OpenSSL vor allem für Software-Entwickler von Nutzen ist. Allerdings finden System Administratoren und IT-Spezialisten die Kommandozeilen-Utilities von OpenSSL sehr nützlich. OpenSSL war früher unter der BSD-style Lizenz lizenziert, aber ab Version 3.0 ist es unter der Apache 2.0 Lizenz lizenziert, also unter dem BSD-style. Mit dieser Lizenz kann OpenSSL sowohl in geschlossenen als auch in offenen Source-Anwendungen genutzt werden (39).

OpenSSL bietet eine Vielzahl von kryptographischen Algorithmen an, darunter Algorithmen für Symmetrie und Asymmetrie Encryption, digitale Signaturen, Nachrichtendienste und Schlüsselaustausch. OpenSSL bietet Unterstützung für X.509-Zertifikate, SSL, TLS und DTLS-Protokolle sowie für andere kryptografische Technologien, die weniger bekannt sind (39).

OpenSSL existiert schon eine Weile, und während seiner Entwicklung erhielt es Unterstützung für verschiedene Betriebssysteme. OpenSSL entstand ursprünglich für Betriebssysteme, die Unix ähneln. GNU/Linux, BSD, IBM AIX und MacOS sind nur einige der verschiedenen Varianten von Unix, die OpenSSL bis heute unterstützt. Beliebte Nicht-Unix-Betriebssysteme wie Microsoft Windows, mobile Betriebssysteme wie Android und iOS sowie sogar alte und exotische Betriebssysteme wie MS-DOS und VMS sind alle in OpenSSL enthalten. OpenSSL 3.0 ist eine Version mit großen Änderungen an der internen Architektur der Bibliothek. Die Architekturänderungen sind noch nicht abgeschlossen und werden in OpenSSL 4.0 fortgesetzt (39).

## 4 Was ist GNS3?

GNS3 (Graphical Network Simulator) ist eine Open-Software, die komplexe Netzwerke simuliert und dabei der Funktionsweise echter Netzwerke so nahe wie möglich kommt. Dies alles ist ohne spezielle Netzwerk Hardware wie Router und Switches möglich. Die Software bietet eine intuitive grafische Benutzeroberfläche zum Entwerfen und Konfigurieren von virtuellen Netzwerken. Sie läuft auf herkömmlicher PC-Hardware und kann unter verschiedenen Betriebssystemen wie Windows, Linux und MacOS-X eingesetzt werden. Um vollständige und genaue Simulationen zu ermöglichen, verwendet GNS3 die folgenden Emulatoren, um genau die gleichen Betriebssysteme wie in realen Netzwerken auszuführen:

- Dynamips, der bekannte Cisco IOS-Emulator.
- VirtualBox, auf dem Desktop- und Server-Betriebssysteme wie Juniper Junos.
- Qemu, ein generischer Open-Source-Maschinenemulator, auf dem Cisco ASA, PIX und IPS laufen.

GNS3 ist eine hervorragende Alternative oder Ergänzung zu echten Laboren für Netzwerkingenieure, Administratoren und Personen, die sich auf Zertifizierungen wie Cisco CCNA, CCNP und CCIE sowie Juniper JNCIA, und JNCIE vorbereiten. Auch Open-Source-Netzwerke werden unterstützt. Es kann auch verwendet werden, um mit Funktionen zu experimentieren oder um Konfigurationen zu überprüfen, die später auf echten Geräten eingesetzt werden sollen. Die Programme bieten interessante Funktionen, wie z.B. die Verbindung ihres virtuellen Netzwerks mit einem realen Netzwerk oder Erfassung von Datenpaketen mit Wireshark.

GNS3 gibt die Möglichkeit nicht nur an Online-Kursen zur Verwendung der Tools teilzunehmen, sondern auch Dutzende von Online-Tutorials, die zeigen, wie man verschiedene Aufgaben erledigen kann. Wie bei vielen Open-Source-Produkten gibt es bei GNS3 eine aktive Benutzergemeinschaft und auf der GNS3-Website stehen zahlreiche Dokumentationen und Support-Möglichkeiten zur Verfügung (1).

Wenn man sich für GNS3 entscheidet, kann die GNS3-VM entweder lokal auf dem PC mit einer Virtualisierungssoftware wie VMware, VirtualBox oder Hyper-V, oder aus der

Ferne auf einem Server mit VMware ESXi oder sogar in der Cloud ausführt werden (41). Für diese Abschlussarbeit wurde Linux Ubuntu als Betriebssystem verwendet und GNS3-Installer wurde direkt in dem Betriebssystem installiert (21). Darüber hinaus hat GNS3 eine sehr interessante Integration mit dem Docker. Es ermöglicht, vollständige Netzwerkadapter zu den Containern hinzuzufügen und einige praktische Tools zu kopieren, um die Befehlszeilenumgebung nutzbar zu machen. Bei dem Docker-Container handelt es sich nicht um ein komplettes Betriebssystem - vollständige Betriebssysteme sind für die gleichzeitige Ausführung vieler Prozesse ausgelegt. Anschließend wird geklärt, wie man Docker DNS bind9 Server in GNS3 installieren kann (42).

### **Installation:**

Erstellung von Verzeichnis „jamesbind“, wo man die Docker Datei schreiben und das Image erstellen kann:

```
ekra@ekra-OptiPlex-7020:~$  
ekra@ekra-OptiPlex-7020:~$  
ekra@ekra-OptiPlex-7020:~$ cd jamesbind  
ekra@ekra-OptiPlex-7020:~/jamesbind$  
ekra@ekra-OptiPlex-7020:~/jamesbind$  
ekra@ekra-OptiPlex-7020:~/jamesbind$ ls  
Dockerfile  
ekra@ekra-OptiPlex-7020:~/jamesbind$  
ekra@ekra-OptiPlex-7020:~/jamesbind$  
cat Dockerfile FROM internetsystemsconsortium/bind9:9.11  
ekra@ekra-OptiPlex-7020:~/jamesbind$
```

Abbildung 14: Erstellung von Container-Images

## Dockerfile

Dockerfile ist eine Textdatei, die eine Reihe von Anweisungen enthält, um eine Docker Image zu erstellen. Hier in Abbildung 15 kann man sehen, wie man der Docker Container in GNS3 erstellen kann.

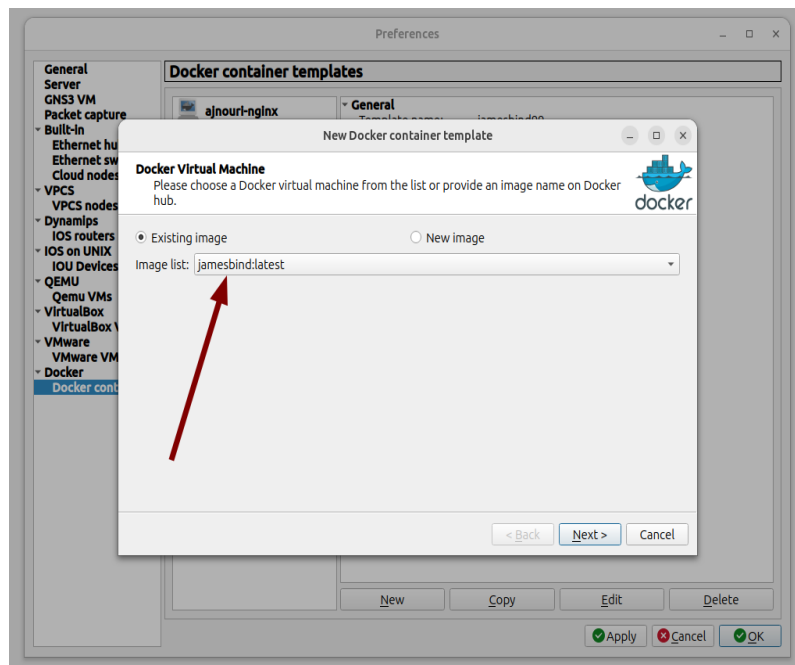


Abbildung 15: Fügen Image zu GNS3 hinzu

Im GNS3-Einstellungsfenster ist es möglich nun eine Docker Image zur Liste der verfügbaren Geräte hinzufügen.

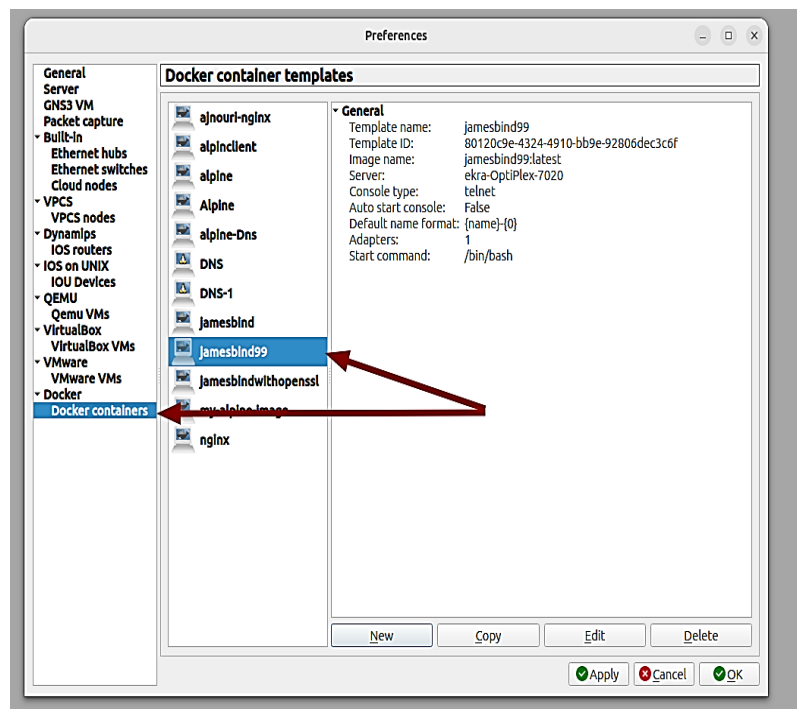


Abbildung 16: Docker Container Verfügbar in GNS3

Jetzt steht der Docker Container Bind9 in GNS3 zum Einsatz bereit.



## 5 GNS3-basierte DNS-Implementierung/Erprobung

### 5.1 DNS Architecture

In Abbildung 17 ist zu erkennen, dass in der DNS-Topologie drei Zonen (Grüne, Weiß und Violett) vorhanden sind. Damit diese drei Zonen miteinander interagieren können, müssen die drei Cisco IOS Router im Zentrum mit dem Routing Protokoll OSPF ausgestattet sein.

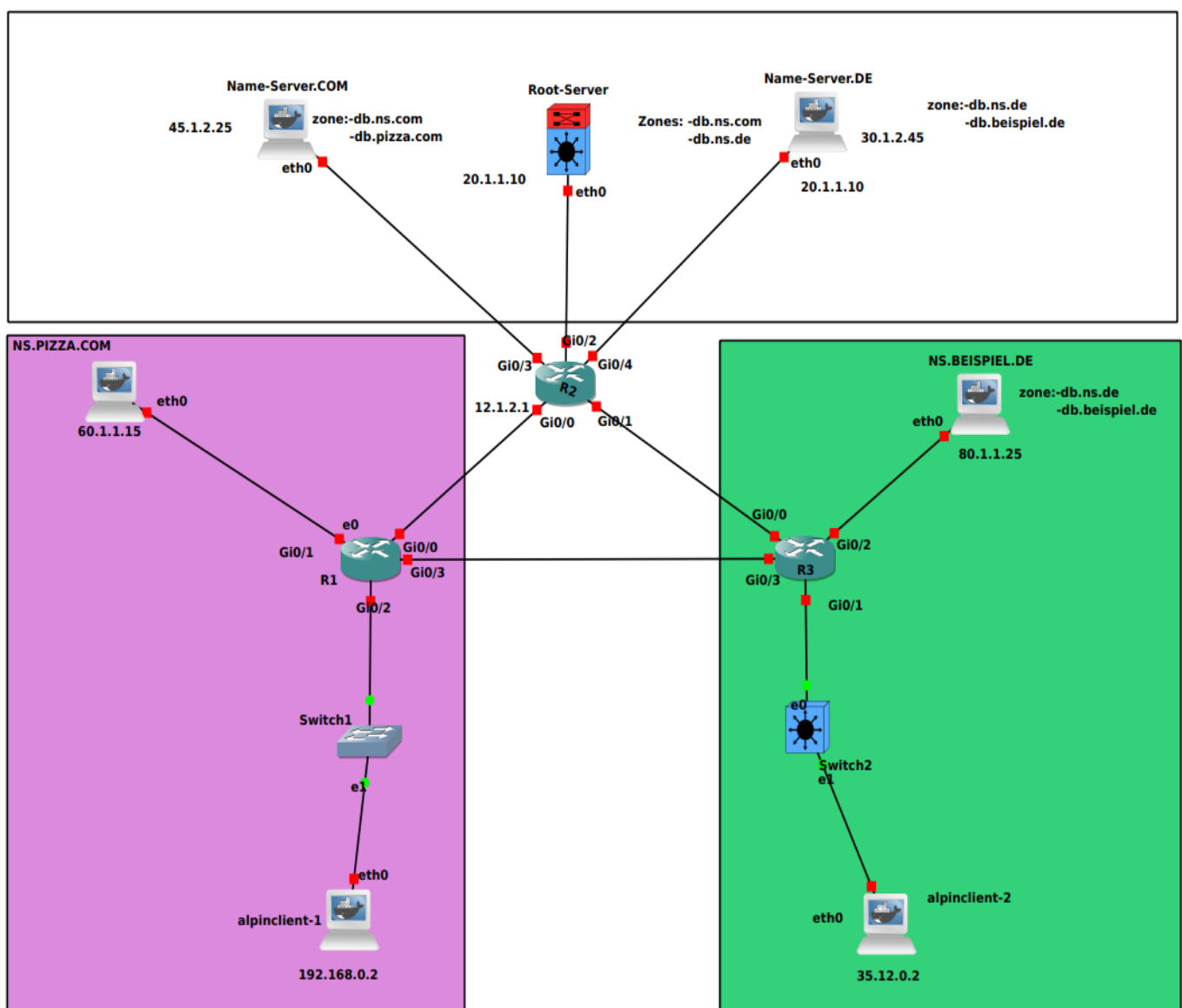


Abbildung 17: DNS-Topologie erstellt in GNS3

Die Konsole dient zur Konfiguration auf die gleiche Weise wie die physischen Router. Sobald OSPF konfiguriert ist, kann OSPF überprüft und mögliche Probleme identifiziert und gelöst werden. Unter Berücksichtigung der hierarchischen Regeln des DNS-Systems entstand die Struktur. Hier ist zu finden:

- 1 Root-Server
- 2 TLD (Top Level Domain: .com, .de)
- 2 Subdomains (ns.pizza.com und ns.beispiel.de)

Die DNS-Hierarchie ermöglicht es, dass DNS-Anfragen von einem Resolver (Client) durch das DNS-System geleitet werden, beginnend mit der Root-Zone, und sich dann zu den spezifischen Nameservern bewegen, die für die Autorität über die angeforderten Domains zuständig sind. Dieser hierarchische Ansatz erleichtert die effiziente und zuverlässige Auflösung von Domain Namen zu IP-Adressen und umgekehrt im gesamten Internet.

### **Netzwerkelement:**

Für die Arbeit an diesem Thesis Projekt wurden insgesamt 7 Docker Container (5 Name Server, 2 Docker Alpine Client) und 3 Cisco IOS verwendet. Die Rolle von jedem Netzwerkelement in dieser Topologie wird in Tabelle 2 beschrieben:

<u>5 Name Server:</u> - Root-Server, - NS.com - NS.de, - NS.pizza.com - NS.beispiel.de	<u>IP-Addressee:</u> - 20.1.1.10 - 45.1.2.25 - 30.1.2.45 - 60.1.1.15 - 80.1.1.25	Name Server basiert auf Docker Container Image Bind9. DNS wurde in Jeder NS konfiguriert. Domain Name sind alpinclient-1 und alpinclient-2 erreichbar (ping NS.pizza.com oder ping NS.beispiel.de). Die Original Key Verschlüsselung (bind.kez) befindet sich in Root-server.
<u>Alpine Docker:</u> - Alpineclient-1 - Alpineclient-2	<u>IP-Addressee:</u> - 192.168.0.2 - 35.12.0.2	In jedem Client Maschine ist es möglich zu überprüfen, ob DNS und DNSSEC erfolgreich konfiguriert wurde, mit Hilfe der Command

		(ping ns.pizza.com, ping ns.beispiel.de und dig +dnssec +multi +trusted-key=zsk.key).
<u>Router:</u> - R1, R2, R3		Provider core routers. Router verwenden OSPF.

Tabelle 1: Beschreibung des Netzwerkelements

Im Netzwerkelement wird beschrieben, welche Funktion ein bestimmtes Gerät in der endgültigen Topologie erfüllen kann und seine IP-Adresse wird angezeigt.

## 5.2 DNS Forwarding/ Delegation Configuration

DNS forwarding verbessert die Leistung, sorgt für einen Lastausgleich und macht Ihr Netzwerk widerstandsfähiger. Es bietet eine Möglichkeit, Namensräume oder Ressourcendatensätze, die nicht in der Zone eines lokalen DNS-Server sind, weiterzuleiten, um Namensanfragen sowohl innerhalb als auch außerhalb eines Netzwerks aufzulösen (43). DNS forwarding ist besonders nützlich, wenn Unternehmen und Einzelpersonen über sehr große Namenräume verfügen. Unternehmen, die zusammenarbeiten, können auch DNS forwarding verwenden, um den Namensraum des jeweils anderen aufzulösen, was die Namensauflösung beschleunigen kann, wenn eines der Unternehmen Probleme mit der Auflösung von Domains hat (44).

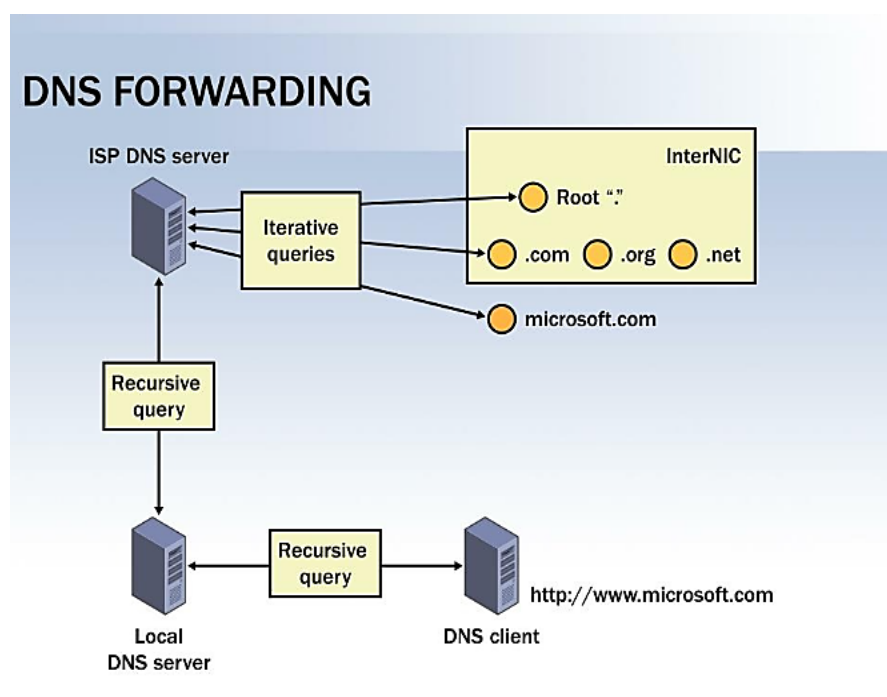


Abbildung 18: DNS Forwarding (45)

### DNS forwarding:

In dieser Arbeit werden alle Anfragen von der Alpineclient-1 Maschine direkt an den Root-server (IP: 20.1.1.10) weitergeleitet. In Root-server wurde die Zone (.com) in der Bind Konfiguration **Datei /etc/bind/zones/db.com** konfiguriert, was mir ermöglicht die Anfrage schnell und einfach zu analysieren und der Domain Name NS.pizza.com in Terminal von der Alpineclient-1 Maschine aufzulösen. Schließlich werden alle Bind Name Server mit dem Befehl „**named -g**“ gestartet. Dadurch wird es im Vordergrund

ausgeführt, mit Debug-Ausgabe, was sehr nützlich sein kann. Alternativ kann auch einfach „named“ ausgeführt werden und es wird im Hintergrund ausgeführt.

In Abbildung 19 und 20 ist zu erkennen, dass alle Nameserver (Root, ns.pizza.com, ns.beispiel.de) im Netzwerk erreichbar sind und die DNS-Konfiguration erfolgreich abgeschlossen wurde.

#### DNS Delegation in ns.pizza.com

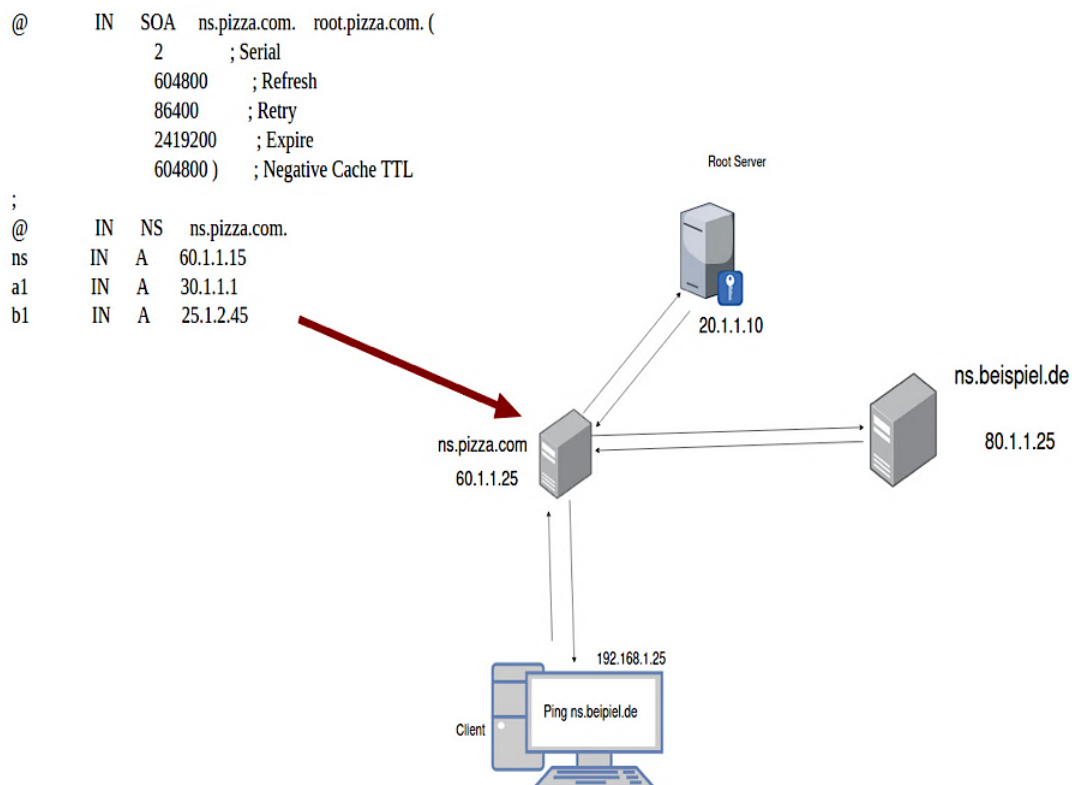


Abbildung 19: DNS-Auflösung ns.pizza.com in Client Maschine

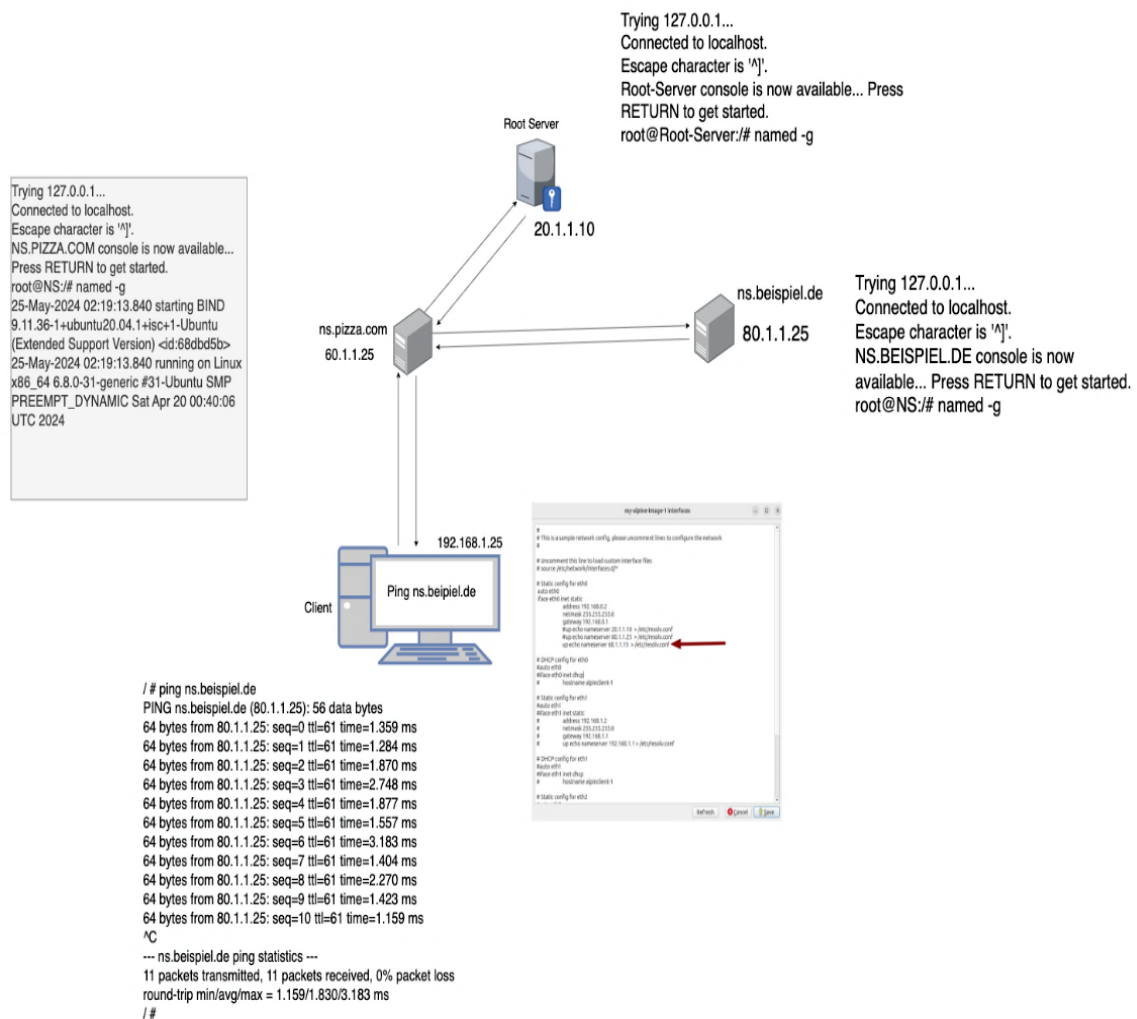


Abbildung 20: DNS Zone Delegation ns.pizza.com

Die DNS-Delegation ist ein wichtiger Aspekt bei der Verwaltung großer und komplexer DNS-Infrastrukturen. Sie ermöglicht es Organisationen, ihre DNS-Zonen in kleinere, besser zu verwaltende Teile aufzuteilen und die Zuständigkeit an verschiedene Gruppen oder Einzelpersonen zu delegieren. Die Delegation ist eine der Grundlagen des gesamten DNS-Systems, da sie die Aufteilung der Zuständigkeit für verschiedene Teile von Domänen ermöglicht und somit Flexibilität und andere Vorteile bietet. DNS-Delegation ist der Prozess, bei dem eine übergeordnete DNS-Zone den DNS-Resolver anzeigt, dass sie die Autorität für eine DNS-Unterzone (oder Child Zone) an eine Gruppe von DNS-Server delegiert hat. Dies ermöglicht es den DNS-Resolver, die Delegierten DNS-Server zu finden und nach den DNS-Einträgen der Unterzone zu fragen (46).

## 6 Beispiel für einen erfolgreichen Angriff auf DNS

Es gibt verschiedene Arten von DNS-Angriffen. Dieses Kapitel konzentriert sich auf die klassischen „Man-in-the-Middle“ (MITM) Angriffe zwischen der Alpine-Client Maschine und dem DNS-Server ns.pizza.com. Zunächst müssen wir definieren:

### Was ist eine MITM-Attacke und DNS-Spoofing?

Der MITM Angriff ist einer der bekanntesten Angriffe im Bereich der Computersicherheit und stellt eine der größten Sorgen für Sicherheitsexperten dar. MITM zielt auf die eigentlichen Daten, die zwischen den Endpunkten fließen, sowie auf die Vertraulichkeit und Integrität der Daten selbst ab (47).

Ein Black Hat Hacker, oft einfach als „Cracker“ bezeichnet, ist jemand, der illegal und ohne Erlaubnis in Computersysteme, Netzwerke oder Software eindringt, um Daten zu stehlen, zu manipulieren oder zu zerstören. Black Hat Hacker sind in der Regel auf kriminelle Aktivitäten ausgerichtet und können großen Schaden anrichten.

Spoofing ist eine Angriffsmethode, die Teil einer Klasse von Maskierungsangriffen ist und darauf abzielt, eine falsche Identität vorzutäuschen. Im Zusammenhang mit DNS bedeutet Spoofing, dass eine DNS-Antwort manipuliert wird, um einen Client auf einen anderen Dienst fehlzuleiten. Dadurch wird die Integrität der DNS-DATEN gefährdet und die Verfügbarkeit der Ressource eingeschränkt (48). Das Fehlen von DNSSEC setzt eine gesamte Infrastruktur verschiedenen Sicherheitsrisiken und Schwachstellen aus, was die Wahrscheinlichkeit erfolgreicher Angriffe erhöht und die Integrität, Vertraulichkeit und Verfügbarkeit der Dienste gefährdet.

Abbildung 21 verdeutlicht, dass Hacker durch das Fehlen der DNSSEC-Implementierung Netzwerkpakete abfragen und DNS-Antworten manipulieren können. Die Alpine-Client-Maschine verfügt nicht über Mechanismen zur Erkennung und zum Schutz von Datenmanipulationen. Daher fällt es dem Alpine-Client schwer, die vom Server empfangenen Daten zu authentifizieren und zu überprüfen.

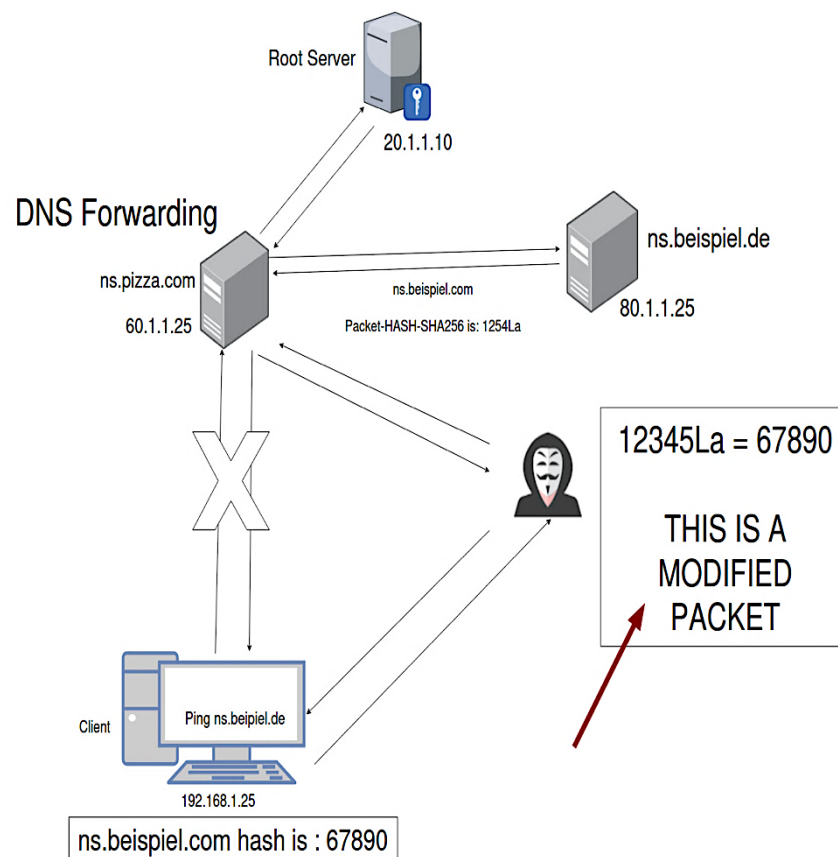


Abbildung 21: MITM-Angriff auf DNS Request

Abbildung 22 illustriert mithilfe von Wireshark-PCAP-Paketen die Fähigkeit eines Hackers, DNS-Anfragen zu lesen und DNS-Daten zu verändern. PCAP-Pakete sind in der Lage, Angriffe und Sicherheitsbedrohungen zu erkennen. Die Kontrolle des Netzwerkverkehrs ermöglicht es, verdächtige Tätigkeiten zu erkennen, Schwachstellen aufzudecken und Maßnahmen zur Sicherung der Netzwerksicherheit zu ergreifen. Hier ist es wichtig zu betonen, dass mit Hilfe von PCAP-Paketen ein Angreifer verschiedene Angriffe durchführen und sensible Informationen sammeln kann. Hier sind einige Möglichkeiten, wie ein Angreifer PCAP-Pakete verwenden kann: Passwortdiebstahl, Man-in-the-Middle-Angriffe, Informationssammlung, DNS Spoofing.



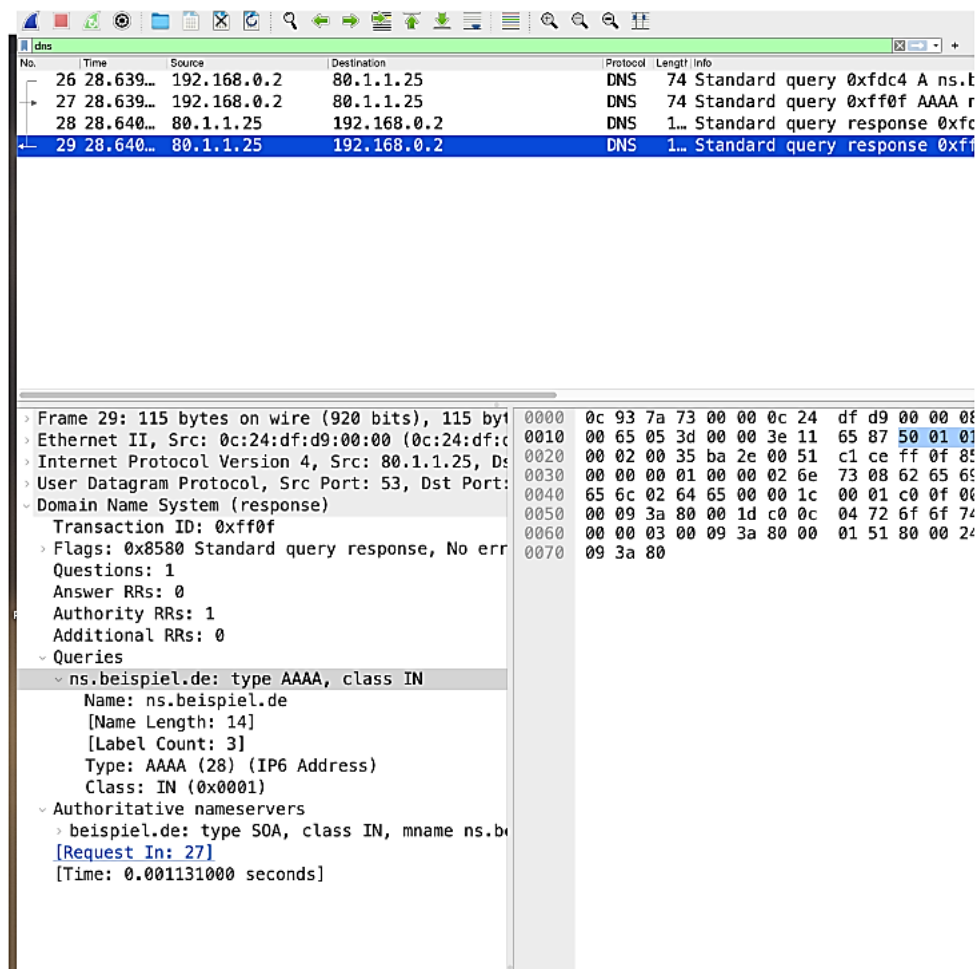


Abbildung 22: Analyse DNS Request mit Wireshark

Im Paketlisten-Panel von Wireshark entspricht jede Zeile im oberen Bereich einem einzelnen Netzwerkpaket, das erfasst wurde. Die Standardansicht zeigt die Zeit, zu der das Paket erfasst wurde (relativ zum Beginn der Aufzeichnung), sowie die Quell- und Ziel-IP-Adressen und das verwendete Protokoll.

Im Packet Details Panel von Wireshark befinden sich detaillierte Informationen zum ausgewählten Paket im oberen Fenster. Die „+“ Symbole ermöglichen es, verschiedene Ebenen von Details über jede Schicht der Informationen innerhalb des Pakets anzuzeigen. Im obigen Beispiel habe ich ein DNS-Antwortpaket ausgewählt. Der DNS-Antwortbereich (Anwendungsschicht) des Paketes wurde erweitert, um zu verdeutlichen, dass die ursprüngliche Anfrage eine DNS-Auflösung für **ns.beispiel.de** angefordert hat.

Die erhaltene Antwort gibt Auskunft darüber, dass die verfügbaren IP-Adressen, einschließlich 80.1.1.25, umfassen (49).

## 7 Beschreibung der DNSSEC Implementierung/Erprobung

### 7.1 DNSSEC-Konfiguration

Die Einrichtung von DNSSEC umfasst verschiedene Schritte zur Unterzeichnung von DNS-Zonen sowie zur Sicherstellung der Integrität der DNS-Daten. Bevor wir mit der vollständigen Konfiguration beginnen, ist es von Bedeutung zu betonen, dass eine Konfiguration von DNSSEC auf den Root-Servern nicht unbedingt notwendig ist, um eine Funktionalität im gesamten DNS-System sicherzustellen. DNSSEC wird nicht direkt auf den Root-Servern eingeführt, sondern auf den Nameservern, die für die Top-Level-Domains (TLDs) verantwortlich sind.

In unserem Fall findet die Konfiguration auf dem Nameserver **ns.beispiel.de** statt. Für jeweils jede zu sichernde Zone wird ein individueller Zonenschlüssel (Zone Signing Key) erstellt, der einen öffentlichen und einen privaten Schlüssel beinhaltet. Der öffentliche Teil des ZSK wird in der dazugehörigen Zonendatei als DNSKEY Resource Record hinterlegt.(50)

Anschließend werden mit dem privaten Schlüssel alle einzelnen RRs in der Zone digital signiert (51). Es wird auch eine syntaktisch identische Schlüssel Unterzeichnung KSK (Key Signing Key) und Zonensignatur (ZSK) erstellt.

Der private Key Signing Key (KSK) sollte eine Länge von 2048 Bit haben und ist für eine Dauer von 2 bis 4 Jahren gültig. Wie der Name andeutet, wird der private KSK verwendet, um andere Schlüssel zu signieren. Der private Zone Signing Key (ZSK) sollte mindestens 512 Bit lang sein und hat eine Gültigkeit von nur 1 bis 2 Monaten. Mit dem privaten ZSK signiert der Domain-Administrator alle DNS-Einträge in seiner Zonen-Datenbank (4).

Die Ursache dafür liegt darin begründet, dass die Zone keys (DNSKEY) sich häufiger verändern können.

Wie in Abbildung 23 dargestellt, würde in einer Vertrauenskette die übergeordnete Zone der untergeordneten Zone nicht mehr Vertrauen. In diesem Fall bleibt jedoch der Schlüssel zur Signierung (KSK) für einen längeren Zeitraum unverändert.

Deshalb wird die Kette des Vertrauens zwischen der Eltern- und der Kinderzone auch bei Veränderungen beibehalten. Andernfalls wäre es sehr ineffizient und fehleranfällig, wenn die Child-Zone fortlaufend neue Schlüssel an die Parent-Zone weitergeben müsste.

Am Ende erfolgt die Signatur der Zone mithilfe der DNSKEYs, während die Resource Records mithilfe der RRSIGs unterzeichnet werden (52).

DNSSEC verwendet asymmetrische Schlüsselpaaren – also Paare aus privaten und öffentlichen Schlüsseln. Dieses Zwei-Elemente-System wurde von Architekten der IETF (Internet Engineering Task Force) entwickelt. Ein Zone Signing Key (ZSK) schützt die einzelnen Ressourceneinträge (RRs) in einer Zonendatei und wird seinerseits durch den Key Signing Key (KSK) geschützt (siehe Abbildung 23)(53).

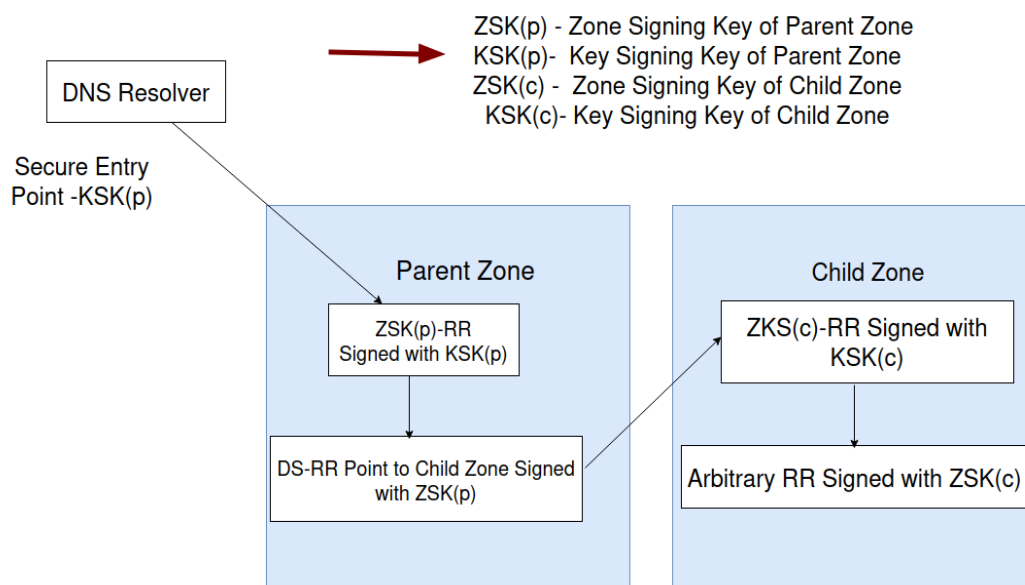


Abbildung 23: Chains of Trust DNSSEC

## DNS-Resolver

Ein DNS-Resolver ist ein entscheidender Bestandteil des Domain Name System (DNS), der die Aufgabe hat, Domain-Namen in IP-Adressen aufzulösen. Wenn ein Benutzer eine URL in einen Webbrowser eingibt, übersetzt der DNS-Resolver diese URL in eine IP-Adresse, die Computer verwenden, um miteinander zu kommunizieren.

## Parent Zone in DNSSEC

Die Parent Zone ist die übergeordnete Zone im DNS-Hierarchiebaum. Für eine gegebene Domain ist die Parent Zone die Zone, die die NS (Nameserver)-Einträge und die DS (Delegation Signer) -Einträge für diese Domain enthält. Zum Beispiel ist für die Domain example.com die Parent Zone .com.

### **Child Zone in DNSSEC**

Die Child Zone ist eine untergeordnete Zone, die von einer Parent Zone delegiert wird, und in DNSSEC muss sie bestimmte Sicherheitsmechanismen implementieren, um Teil der Vertrauenskette zu sein.

Mit DNSSEC wird eine Vertrauenskette aufgebaut, deren Ursprung der Root-KSK ist, der verwendet wird, um den ZSK der DNS-Root-Zone zu signieren. Wie in Abbildung 24 zu sehen ist. Der ZSK der Root-Zone dient zur Validierung der Top-Level-Domains (TLDs). Der Root-KSK muss in allen DNSSEC-fähigen Resolvern hinterlegt sein, damit die signierten Einträge entlang der Vertrauenskette überprüft werden können.

Darüber hinaus ist es wichtig zu erwähnen, dass der ZSK der Root-Zone von Verisign verwaltet wird. Der Root-KSK wird von der Internet Corporation for Assigned Names and Numbers (ICANN) administriert (1).

### **Der DNSSEC-Auflösungsprozess**

Wenn der rekursive Server so konfiguriert ist, dass er validiert und die Auflösungsdaten signiert sind, wird der rekursive Server die Validierung durchführen, sobald die Auflösungsdaten eingegangen sind. Somit beginnt der Validierungsprozess, nachdem der Auflösungsprozess abgeschlossen ist, obwohl während des Auflösungsprozesses bereits ein Großteil der für die Validierung erforderlichen Daten gesammelt werden kann(54).

Nehmen wir an, der rekursive Server ist mit dem Vertrauensanker der DNS-Root-Zone konfiguriert. Dieser Vertrauensanker ist der vertrauenswürdige Schlüssel, der die signierten Auflösungsdaten in der Vertrauenskette validiert. Eine andere Möglichkeit, dies zu betrachten, ist, dass wir anhand der Auflösungsdaten, die vom autoritativen DNS-Server in Bezug auf die Antwort auf die Anfrage erhalten wurden, die Vertrauenskette im DNS-Baum bis zur Root-Zone zurückverfolgen können. Abbildung 24 veranschaulicht diesen grundlegenden Prozess(54).

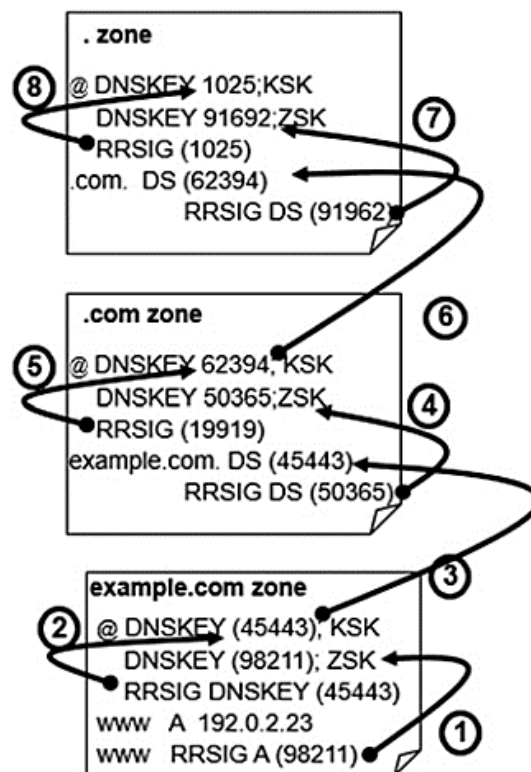


Abbildung 24: DNSSEC Chains of Trust Traversal(54)

Der rekursive Server validiert zunächst die Antwort auf die Anfrage, das A-Record, indem er die Antwort hasht und den ZSK anwendet, um sie mit der RRSIG-Signatur zu vergleichen. Wenn dies übereinstimmt, wie in Schritt 1 in Abbildung 24 gezeigt, wiederholt der rekursive Server diesen Prozess, um den ZSK und den KSK zu validieren, indem er die Signaturen auf dem DNSKEY-RRset überprüft, was Schritt 2 darstellt. Nach diesen beiden Validierungen kann mein rekursiver Server bestätigen, dass die Antwort auf die Anfrage vom Administrator von example.com signiert wurde und die Antwort wie vom Administrator veröffentlicht ist. Aber vertraue ich dem Schlüssel von example.com? Nein, dieser Schlüssel ist nicht als Vertrauensanker konfiguriert(54).

Der rekursive Server muss daher feststellen, ob `example.com` in der Vertrauenskette mit seiner übergeordneten Zone `.com` verbunden ist. Diese Vertrauensverknüpfung wird in DNS in Form eines Delegation Signer (DS) Ressourceneintrags veröffentlicht. Der DS-Eintrag enthält einen Hash des entsprechenden Key Signing Keys (KSK) der untergeordneten Zone, um den KSK der untergeordneten Zone zu authentifizieren. Schritt 3 unseres Prozesses überprüft, ob die übergeordnete Zone einen DS-Eintrag hat, der dem KSK von `example.com` entspricht. Schritt 4 validiert die Signatur des DS-Eintrags mit dem Zone Signing Key (ZSK) von `.com`, während Schritt 5 die Schlüssel von `.com` validiert. Durch Wiederholen dieser Schritte bis zur Root-Zone in den Schritten 6–8 gelangen wir zur Validierung der Signaturen für die Root-Zone bis zum KSK, das mit unserer Abfrageantwort für `www.example.com` verknüpft ist. Da wir dem KSK vertrauen, betrachten wir diese Antwort als sicher. Prozess überprüft, ob die übergeordnete Zone einen DS-Eintrag hat, der dem KSK von `example.com` entspricht. Schritt 4 validiert die Signatur des DS-Eintrags mit dem Zone Signing Key (ZSK) von `.com`, während Schritt 5 die Schlüssel von `.com` validiert. Durch Wiederholen dieser Schritte bis zur Root-Zone in den Schritten 6–8 gelangen wir zur Validierung der Signaturen für die Root-Zone bis zum KSK, das mit unserer Abfrageantwort für `www.example.com` verknüpft ist. Da wir dem KSK vertrauen, betrachten wir diese Antwort als sicher.

Obwohl Abbildung 24 das Abfragen "nach oben" im Domainbaum andeutet, werden die Signaturen, Schlüssel und DS-Einträge normalerweise vom rekursiven Server zwischengespeichert, um den Validierungsprozess zu beschleunigen(54).

### 7.1.1 Key Signing Key (KSK) generieren

Das Key Signing Key (KSK) wird erstellt. Das Tool `dnssec-keygen`, das Teil von Bind9 ist, ermöglicht es, den KSK zu generieren. Abbildung 25 zeigt und beschreibt die Befehle, die verwendet werden, um den KSK zu generieren:

```
mkdir -p /etc/bind/keys/
```

```
chown -R bind:bind /etc/bind/keys/
```

Innerhalb dieses neuen Verzeichnisses wird der KSK mit folgendem Befehl erzeugt:

```
dnssec-keygen -3 -a RSASHA512 -b 4096 -n ZONE -r /dev/urandom -f KSK example.tld
```

Befehlsbeschreibung:

*-3 aktiviert das gewünschte NSEC3*

*-a bestimmt den Typ der Signatur*

*-b gibt die gewünschte Blockgröße an*

*-n spezifiziert den Name typ wie ZONE, HOST, USER*

*-f speziell für KSK muss diese Flag gesetzt werden*

In dem Verzeichnis liegen jetzt zwei Dateien:

*Kexample.tld. +010+64413.key (DNSKEY Record),*

der dem Verifizieren der ZSKs dient und von höheren Zonen signiert wird.<sup>[1][SEP]</sup>

*Kexample.tld. +010+64413.private,*

Mit dem privaten Schlüssel zur Signatur des öffentlichen ZSK.

Der DNSKEY Record im öffentlichen Schlüssel sieht etwa so aus:

*Cat Kexample.tld. +010+64413.key*

This is a key-signing key, 46eyed 64413, for example.tld.

Created: 20200313175118 (Fri Mar 13 18:51:18 2020).

Publish: 20200313175118 (Fri Mar 13)

Abbildung 25: DNSSEC Key Signing Key generieren(50)

### 7.1.2 Zone Signing Key (ZSK) generieren

Der Zone Signing Key wird anschließend erzeugt, gefolgt von einer Anpassung der Dateiattribute (50). Wie in Abbildung 26 zu sehen ist, werden insgesamt zwei Dateien erzeugt: der private ZSK-Schlüssel und der öffentliche ZSK-Schlüssel.

```
Dnssec-keygen -3 -a NSEC3RSASHA1 -b 2048 -n ZONE -r /dev/urandom -R bind: bind /etc/bind/keys/
```

Jetzt liegen zwei weitere Dateien in dem Verzeichnis:

*Kexample.tld. +007+40385.key,*

der öffentliche Schlüssel (DNSKEY Record), der dem Verifizieren der DNS-Antworten dient.

*Kexample.tld. +007+40385.private,*

der private Schlüssel zum Signieren der Resource Records.

Der Inhalt dieses ZSK stellt sich in etwa so dar.

```
Cat Kexample.tld. +007+40385.key
```

```
This is a zone-signing key, 47eyed 40385, for example.tld.; Created: 20200313175211  
(Fri Mar 13 18:52:11 2020); Publish: 20200313175211 (Fri Mar 13 18:52:11 2020)  
Activate: 20200313175211 (Fri Mar 13 18:52:11 2020)  
example.tld. IN DNSKEY 256 3 7 AwEAAuXGNQMitXTRPKAkme9[...] Mqi3ZvO
```

Abbildung 26: Zone Signing Key (ZSK) generieren(50)



### 7.1.3 Signieren der Zonen selbst

Durch die Verwendung der \$INCLUDE-Direktive für die Signatur kann nun die Zone in der jeweiligen Zonendatei vorbereitet werden, indem der öffentliche Schlüssel hinzugefügt wird. Dies kann durch eine Schleife erfolgen (50). In Abbildung 27 wird mit dem Befehl „dig“ gezeigt, wie die DNSSEC Zonen signiert wurde.

Man darf nicht außer Acht lassen, dass nach jeder Änderung in der Zonendatei diese eine neue Signatur benötigt. Außerdem sind die signierten Zonen standardmäßig nur 30 Tage lang gültig. Mit dem Befehl „dnssec-signzone“ und der Option „-e YYYYMMDD-DHHMMSS“ kann ein eigenes Enddatum für die Gültigkeit der signierten Zonendatei festgelegt werden (50).



```
/ #
/ # dig ns.beispiel.de

; <<>> DiG 9.16.39 <<>> ns.beispiel.de
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45323
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 327e1fb4c7f72c4253fe15e5665ba920fd7502550b3e4c87 (good)
;; QUESTION SECTION:
;ns.beispiel.de.                IN      A

;; ANSWER SECTION:
ns.beispiel.de.                604800  IN      A      80.1.1.25

;; Query time: 8 msec
;; SERVER: 60.1.1.15#53(60.1.1.15)
;; WHEN: Sat Jun 01 23:05:04 UTC 2024
;; MSG SIZE rcvd: 87

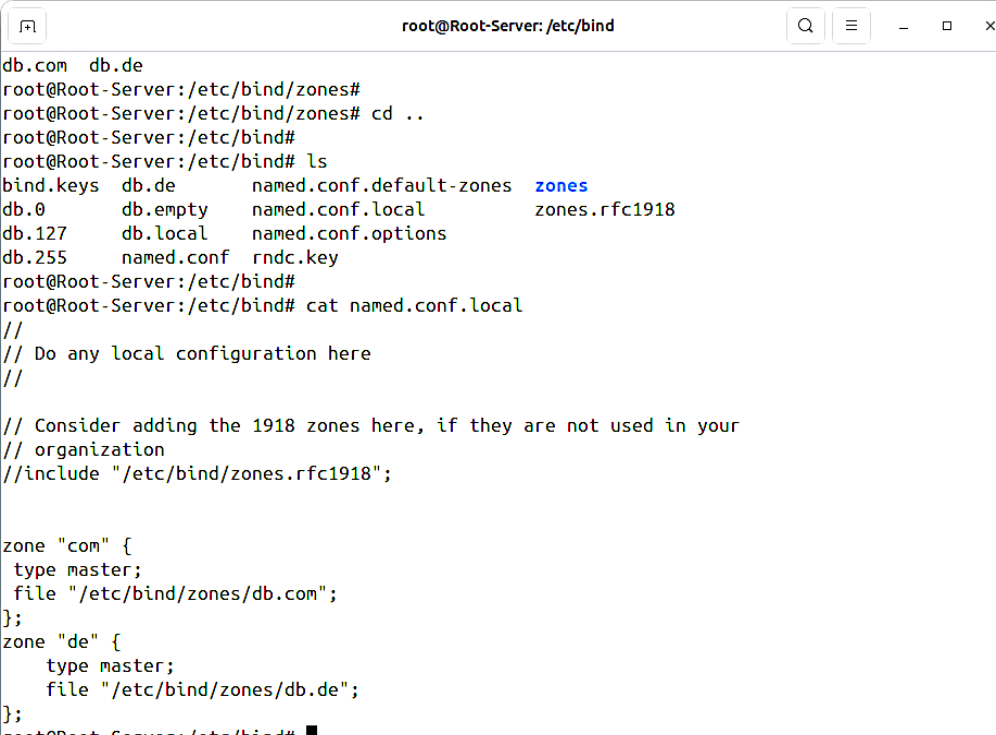
/ #
```

Abbildung 27: DNSSEC Zonen signieren

### 7.1.4 DNSSEC in Bind9 aktivieren

Man kann DNSSEC in den globalen Options von BIND9 aktivieren, indem man die folgende Datei in `/etc/bind/named.conf.options` einträgt. In Abbildung 28 ist zu beachten, dass der Befehl „`dnssec-validation yes`“ der einzige Befehl ist, der die DNSSEC-Validierung in Bind9 ermöglicht(50).

```
root@Root-Server:/etc/bind#  
root@Root-Server:/etc/bind# cd named.conf.local  
bash: cd: named.conf.local: Not a directory  
root@Root-Server:/etc/bind# cat named.conf.options  
options {  
    directory "/var/cache/bind" ;  
    listen-on { any; };  
    allow-query { any; };  
    dnssec-validation no;  
};  
root@Root-Server:/etc/bind#
```



```
root@Root-Server: /etc/bind  
db.com db.de  
root@Root-Server:/etc/bind/zones#  
root@Root-Server:/etc/bind/zones# cd ..  
root@Root-Server:/etc/bind#  
root@Root-Server:/etc/bind# ls  
bind.keys db.de named.conf.default-zones zones  
db.0 db.empty named.conf.local zones.rfc1918  
db.127 db.local named.conf.options  
db.255 named.conf rndc.key  
root@Root-Server:/etc/bind#  
root@Root-Server:/etc/bind# cat named.conf.local  
//  
// Do any local configuration here  
//  
  
// Consider adding the 1918 zones here, if they are not used in your  
// organization  
//include "/etc/bind/zones.rfc1918";  
  
zone "com" {  
    type master;  
    file "/etc/bind/zones/db.com";  
};  
zone "de" {  
    type master;  
    file "/etc/bind/zones/db.de";  
};  
root@Root-Server:/etc/bind#
```

Abbildung 28: DNSSEC Aktivierung

### 7.1.5 DNSSEC Diagnostik und Tools

In der Regel ermöglichen Applikation einen impliziten Zugriff auf das DNS. Dennoch ist es gelegentlich erforderlich, insbesondere zur Fehleranalyse, einen spezialisierten Client zu verwenden und explizit Fragen an das DNS zu stellen. Für diesen Zweck werden 2 wichtige DNS-Tools vorgestellt (55).

***Nslookup*** gehört zu den beliebtesten Programmen für die DNS-Abfrage und ist ein Bestandteil der Bind-Verteilung. Trotz der Nachricht, dass das ISC als „deprecated“ eingestuft wird, liefert es *nslookup* immer noch mit jeder neuen Bind-Distribution aus. Windows-Systeme können es auch auf den Kommandozeilen sofort verwenden. Der Befehl kann direkt auf einen zu lösenden Namen und einen Server zur Auflösung aufgerufen werden. Es beginnt ohne Streit in einem interaktiven Modus. Wenn unqualifizierte Namen angegeben werden, nutzt *nslookup* die Einstellungen aus der Datei `/etc/resolv.conf`, um die zu verwendenden Domainnamen zu ermitteln. Eine *nslookup*-Abfrage ist in Abbildung 29 zusehen (55)

```
ekra@Ekra-MacBook-Pro-8 ~ % nslookup google.com
Server:          9.9.9.9
Address:  9.9.9.9#53
```

```
Non-authoritative answer:
Name:    google.com
Address: 142.250.186
```

Abbildung 29: Domain IP-Adresse mit nslookup

Die beiden ersten Zeilen bezeichnen den genutzten Nameserver mit der Portnummer, während die dritte Zeile den gewünschten Namen und die letzte die gefundene Adresse anzeigt.

Darüber hinaus ist es möglich, andere RR-Typen (auch andere RR-Klassen, aber das ist heutzutage nicht mehr relevant) abzufragen, wenn man nslookup im interaktiven Mo-

us benutzt. Um den MX-RR von google.com zu bestimmen, geht man wie folgt vor(55):

```
ekra@Ekras-MacBook-Pro-8 ~ % nslookup google.com
```

```
Server:          9.9.9.9
```

```
Address:  9.9.9.9#53
```

```
Non-authoritative answer:
```

```
Name:      google.com
```

```
Address: 142.250.186.110
```

Abbildung 30: nslookup im interaktiven Modus

Das Kommando set ändert den Fragetyp für die gesamte Dauer der Sitzung. Wenn Adressen wieder aufgelöst werden sollen, müssen Sie mit set type=a erneut auf diese zurückschalten.

**Dig** ist ein Netzwerk-Tool, das DNS-Server um Informationen fragen kann. Es kann bei der Diagnose von Problemen mit Domain-Pointing sehr nützlich sein und ist eine gute Möglichkeit, zu überprüfen, ob die Konfiguration funktioniert (56). *Dig* ist ein flexibles Werkzeug für DNS-Abfragen. Der erste wesentliche Unterschied zwischen *Dig* und nslookup Tools besteht darin, dass dig unqualifizierte Domain Namen nicht ergänzt, sondern wörtlich nimmt (55). Hier in Abbildung 31 wird gezeigt, wie man Domain IP-Adresse mit dig Befehle erstellen kann.

```
; <<>> DiG 9.10.6 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26472
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.                IN      A
;; ANSWER SECTION:
google.com.                 256 IN    A      108.177.119.139
google.com.                 256 IN    A      108.177.119.138
```

```

google.com.      256  IN   A    108.177.119.102
google.com.      256  IN   A    108.177.119.101
google.com.      256  IN   A    108.177.119.113
google.com.      256  IN   A    108.177.119.100

```

```

;; Query time: 18 msec
;; SERVER: 9.9.9.9#53(9.9.
##

```

Abbildung 31: Dig Domain Name

### 7.1.6 Testergebnis und DNSSEC-Validation

Hier geht es allgemein um die Registrierung, des DNSSEC-Schlüssels. Nachdem man die DNSSEC Grundkonfiguration im Server abgeschlossen hat, soll man in der Regel DNSSEC bei dem eigenen Register aktivieren. Normalerweise geschieht dies über die Webportale der Registerarie, indem man den passenden DNSKEY RR für die Zone eingibt und zusätzlich den DS-Rekord hinzufügt. Anschließend wartet man darauf, dass die AXFR-Übertragung der Nameserver des Regiestars abgeschlossen ist und eine Abfrage durchgeführt wird.

AXFR bezieht sich auf das Protokoll, das bei einem DNS-Zonen-Transfer verwendet wird. Eine von Client initiierte Anfrage. Daher können Informationen auf dem primären DNS-Server bearbeitet und anschließend mit AXFR vom sekundären DNS-Server die gesamte Zone downloaden. In unserem Fall, in der Abbildung 32 ist der Register der primäre DNS-Server und der Client der sekundäre DNS-Server (57).

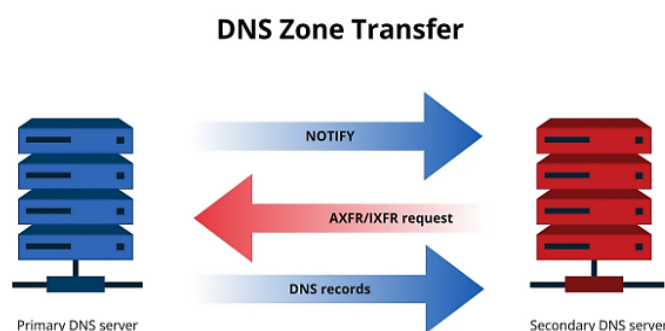


Abbildung 32: DNS Zone Transfer zwischen Register und Client

Der Domain Register ist in der Praxis derjenige, der DNSSEC aktiviert. Dieses Projekt wird in einem Virtual Network Emulation GNS3 durchgeführt. Daher ist geplant, einen Trust Point wie etwa eine Certificate Authority (CA) zwischen ns.pizza.com und der

Client Maschine einzurichten. Dieser Trust Point hilft dnssec bei der Validierung und dem Erhalt von Ad flag. Hundertprozentig bestätigt mit dem Ad flag, dass es DNSSEC erfolgreich umgesetzt wurde.

Wie aus der Abbildung 33 hervorgeht, erhält der Client von der Trust Point (CA) eine autorisierte und überprüfte Ausgabe. Ein Trust Point ist im Wesentlichen ein vertrauenswürdiger Ausgangspunkt für die Überprüfung der Authentizität und Integrität in einem kryptografischen System.

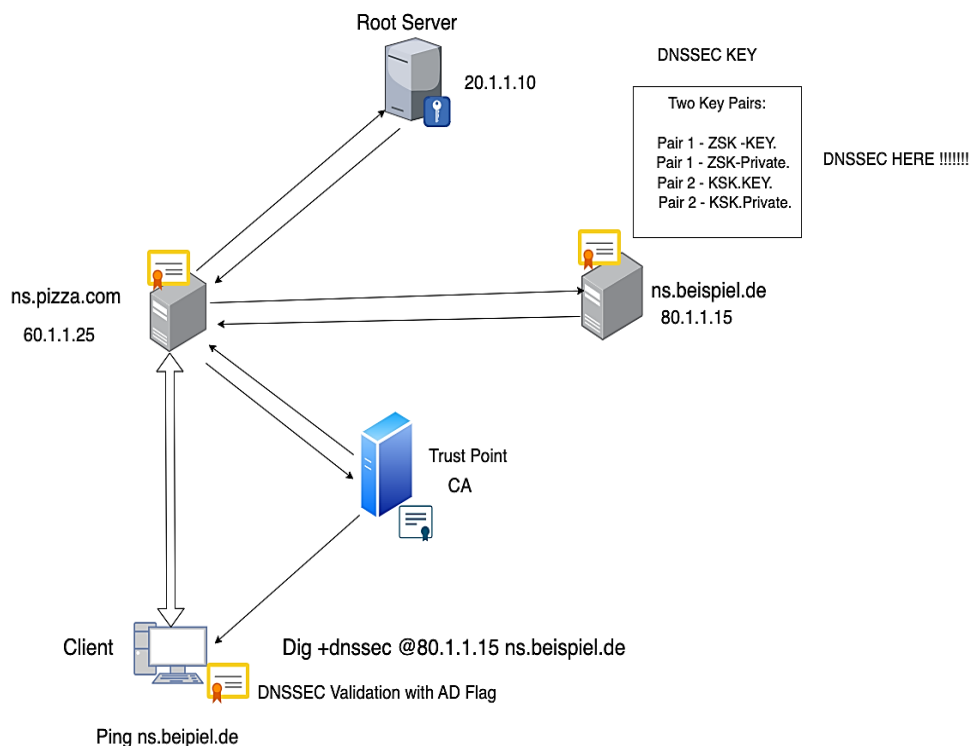
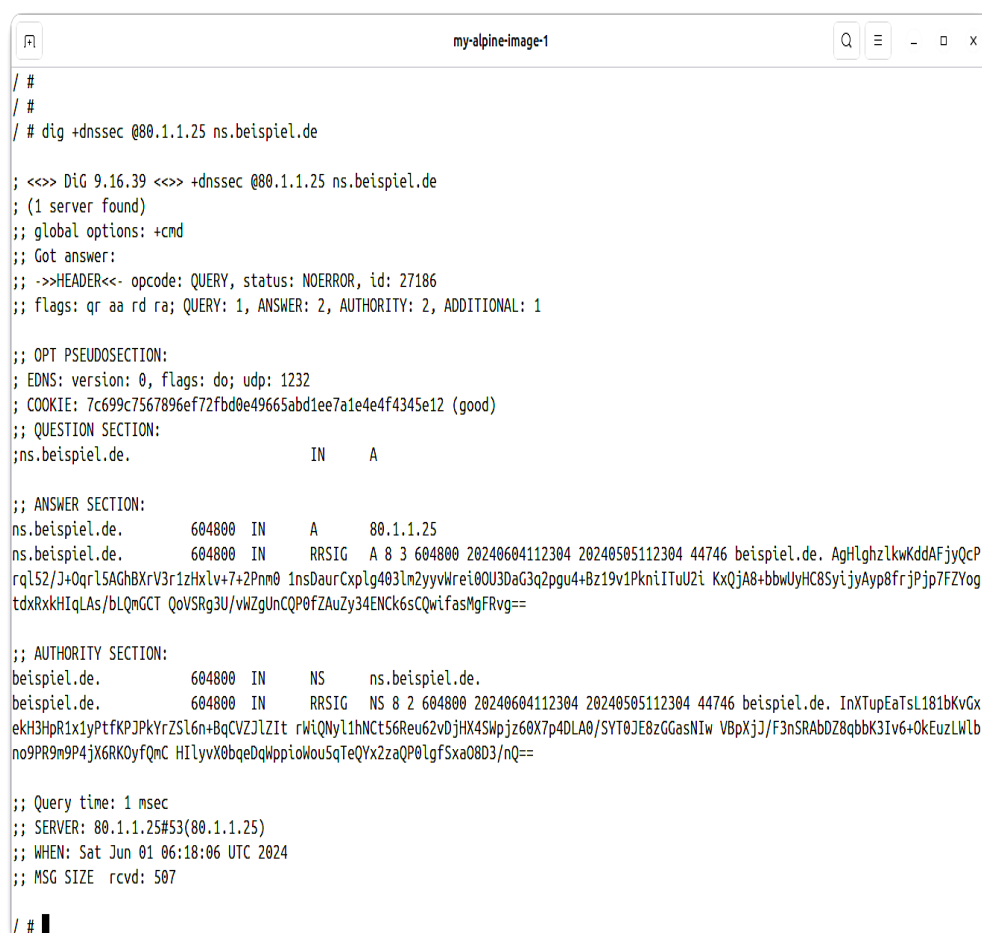


Abbildung 33: DNSSEC Validation

Das „ad“ (Authenticator Data) Flag ist ein Indikator im DNS-Protokoll, der anzeigt, dass die DNS-Antwort durch DNSSEC erfolgreich authentifiziert wurde. Wenn das „ad“ in einer DNS-Antwort gesetzt ist, bedeutet dies, dass der Resolver die Signaturen überprüft und festgestellt hat, dass die Antwort nicht verändert wurde und von einer vertrauenswürdigen Quelle stammt.

Ein DNS-Resolver, der DNSSEC unterstützt, prüft die digitalen Signaturen der DNS-Daten. Wenn die Überprüfung erfolgreich ist. Setzt er das „ad“ Flag in der Antwort. Das „ad“ Flag erhöht die Sicherheit, indem es den Empfängern der DNS-Antwort anzeigt, dass die Daten authentisch und nicht manipuliert sind. Hier ist ein Beispiel für eine *dig* Antwort mit einem „ad“ Flag im Terminal zu sehen:

Wenn das DNSSEC bei seinem Registrar bereits aktiviert wurde, ist es nun möglich den DNS-Server oder andere Nameserver abzufragen. In Abbildung 34 wird außerdem gezeigt, wie man die Option „+dnssec“ im Befehl „dig“ verwendet, um auch die RRSIG Records anzuzeigen (58):



```
/ #
/ #
/ # dig +dnssec @80.1.1.25 ns.beispiel.de

; <<>> DiG 9.16.39 <<>> +dnssec @80.1.1.25 ns.beispiel.de
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27186
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
; COOKIE: 7c699c7567896ef72fbd0e49665abd1ee7a1e4e4f4345e12 (good)
;; QUESTION SECTION:
;ns.beispiel.de.                IN      A

;; ANSWER SECTION:
ns.beispiel.de.                604800  IN      A      80.1.1.25
ns.beispiel.de.                604800  IN      RRSIG   A 8 3 604800 20240604112304 20240505112304 44746 beispiel.de. AghLghzlkWkddAFjyQcP
rql52/J+QqrLSAGhBXrV3r1zHxlv+7+2Pnm0 1nsDaurCxplg403lm2yyvWrei00U3DaG3q2pgu4+Bz19v1PkniITuUzi KxQjA8+bbwUyHC8SyijyAyp8frjPjp7FZYog
tdxRkxHIqLAs/bLQmGCT QoVSRg3U/vwZgUnCQP0fZauZy34ENck6sCQwifasMgFRvg==

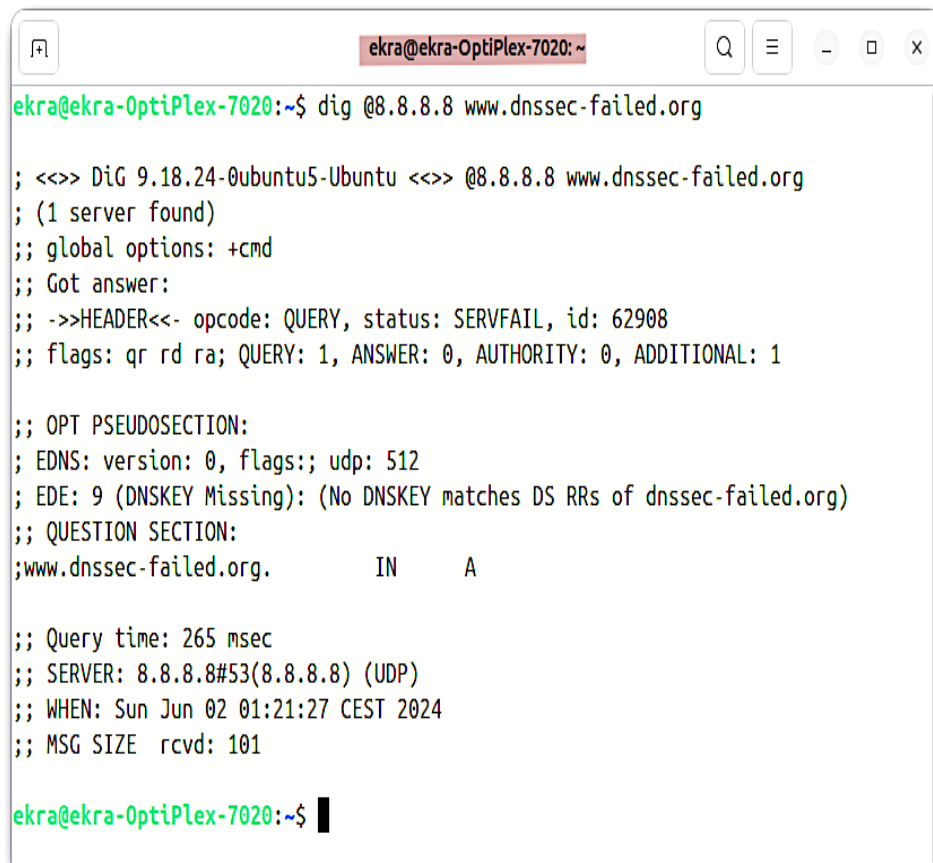
;; AUTHORITY SECTION:
beispiel.de.                  604800  IN      NS      ns.beispiel.de.
beispiel.de.                  604800  IN      RRSIG   NS 8 2 604800 20240604112304 20240505112304 44746 beispiel.de. InXtupEaTsL181bKvGx
ekH3HpR1x1yPtFKPJPKYrZSL6n+BqCVZJLZit rWiQNYl1hNct56Reu62vDjHX4SWpjz60X7p4DLA0/SYT0JE8zGGasNIw VBpXjJ/F3nSRABDZ8qbbK3Iv6+OkEuzLWlb
no9PR9n9P4jX6RK0yfQmC HIlyvX0bqeDqWppioWouSqTeQYx2zaQP0LgfSxa08D3/nQ==

;; Query time: 1 msec
;; SERVER: 80.1.1.25#53(80.1.1.25)
;; WHEN: Sat Jun 01 06:18:06 UTC 2024
;; MSG SIZE rcvd: 507

/ #
```

Abbildung 34: Dig Befehl +dnssec Option in GNS3

Falls es Probleme mit der Validierung gibt, wird ein DNSSEC-gestützter Resolver den gewünschten Rekord nicht zurückgeben. Stattdessen wird eine Antwort mit einem Fehlercode gesendet (59). Abbildung 35 zeigt, dass der Domainname "www.dnssec-failed.org" keine DNSSEC-Validierung unterstützt, was durch die Verwendung der Befehle „dig @8.8.8.8 www.dnssec-failed.org“ und dem Status SERVFAIL verdeutlicht wird.

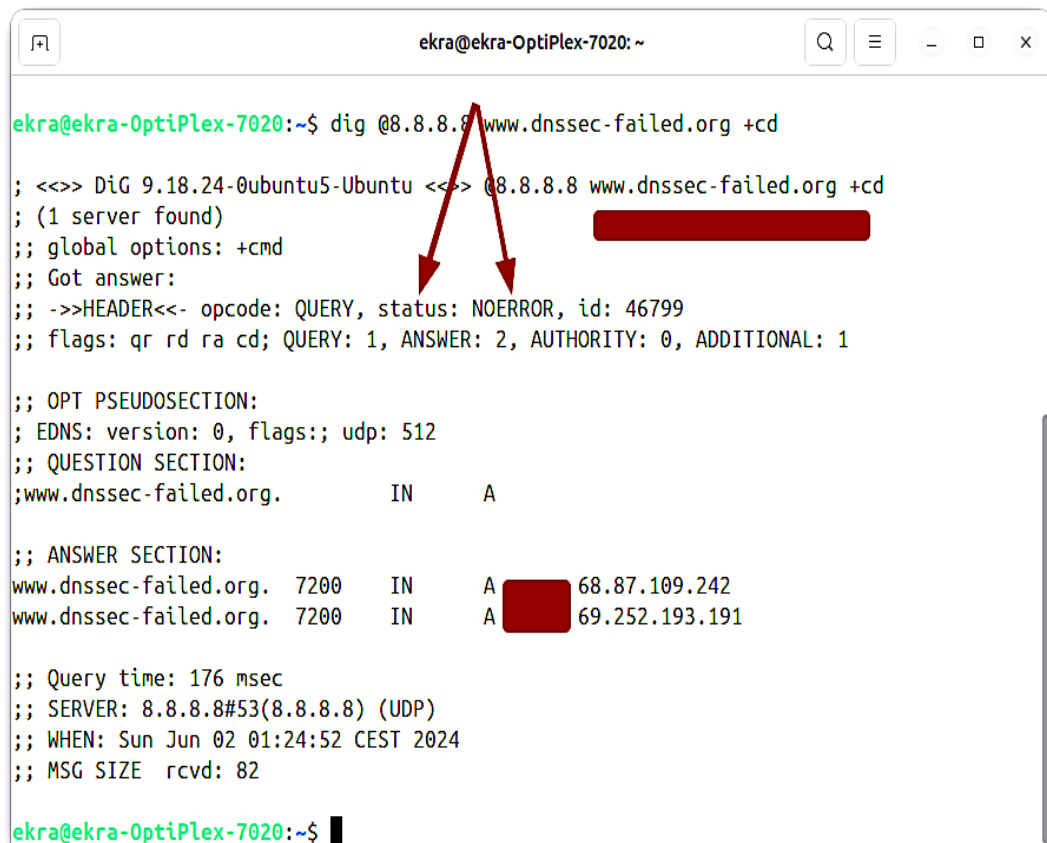


```
ekra@ekra-OptiPlex-7020: ~  
ekra@ekra-OptiPlex-7020:~$ dig @8.8.8.8 www.dnssec-failed.org  
  
; <<>> DiG 9.18.24-0ubuntu5-Ubuntu <<>> @8.8.8.8 www.dnssec-failed.org  
; (1 server found)  
;; global options: +cmd  
;; Got answer:  
;; ->HEADER<- opcode: QUERY, status: SERVFAIL, id: 62908  
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 512  
; EDE: 9 (DNSKEY Missing): (No DNSKEY matches DS RRs of dnssec-failed.org)  
;; QUESTION SECTION:  
;www.dnssec-failed.org.      IN      A  
  
;; Query time: 265 msec  
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)  
;; WHEN: Sun Jun 02 01:21:27 CEST 2024  
;; MSG SIZE rcvd: 101  
  
ekra@ekra-OptiPlex-7020:~$
```

Abbildung 35: Fehler bei DNSSEC Validierung



Wenn die Validierung auf dem vorgelagerten DNS-Resolver deaktivieren werden soll, kann in der Kopfzeile einer Abfrage das CD-Flag (Überprüfung deaktiviert) gesetzt werden. Dann sollte die Antwort mit den angeforderten Einträgen herauskommen, unabhängig vom DNSSEC-Status (59). In Abbildung 36 wird mit der Option "+cd" im dig-Befehl gezeigt, wie die DNSSEC-Validierung deaktiviert wurde.



```
ekra@ekra-OptiPlex-7020: ~  
ekra@ekra-OptiPlex-7020:~$ dig @8.8.8.8 www.dnssec-failed.org +cd  
  
; <<>> DiG 9.18.24-0ubuntu5-Ubuntu <<>> @8.8.8.8 www.dnssec-failed.org +cd  
; (1 server found)  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46799  
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags;; udp: 512  
;; QUESTION SECTION:  
;www.dnssec-failed.org.      IN      A  
  
;; ANSWER SECTION:  
www.dnssec-failed.org.  7200    IN      A      68.87.109.242  
www.dnssec-failed.org.  7200    IN      A      69.252.193.191  
  
;; Query time: 176 msec  
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)  
;; WHEN: Sun Jun 02 01:24:52 CEST 2024  
;; MSG SIZE rcvd: 82  
  
ekra@ekra-OptiPlex-7020:~$
```

Abbildung 36: DNSSEC Deaktivieren mit +cd Option

## 8 Beispiel für erfolgreich abgewerteten Angriff auf DNSSEC

Die Verfügbarkeit, Integrität oder Vertraulichkeit von Informationen kann durch ein Ereignis oder eine Situation beeinträchtigt werden. Eine Bedrohung zielt darauf ab, die Schwachstellen oder Verletzlichkeiten eines Systems zu nutzen, um die Schutzgüter zu gefährden (3). Dabei verursacht dies einen Schaden für den Inhaber oder Nutzer der Information (60).

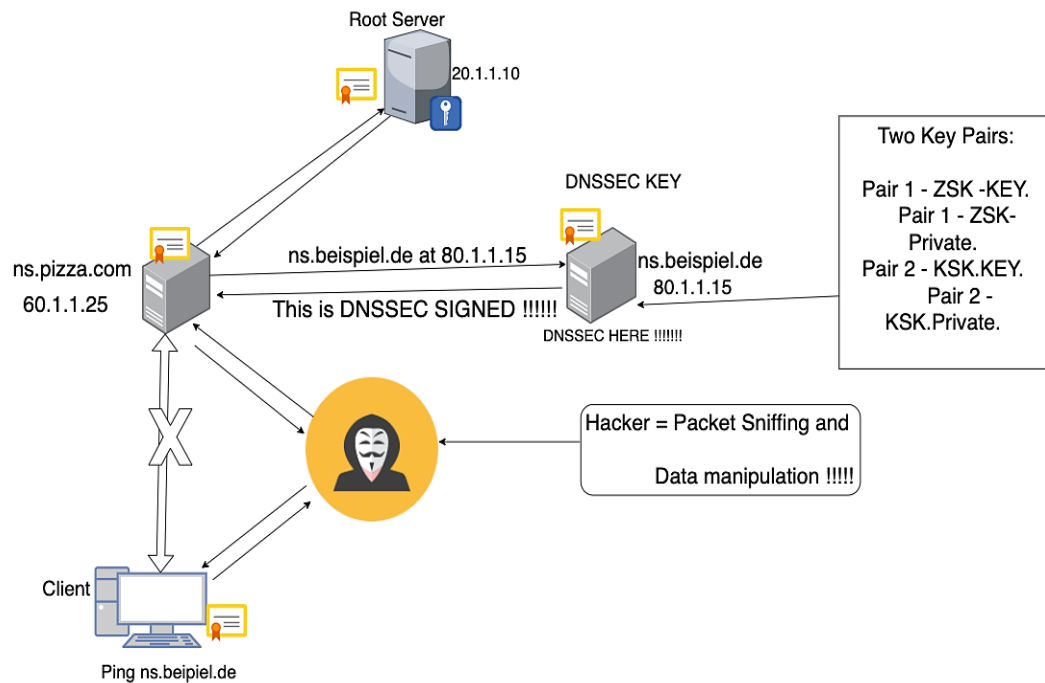
Die Implementierung von DNSSEC bietet einen wirksamen Schutz vor DNS-basierten Angriffen und trägt dazu bei, die Sicherheit und Zuverlässigkeit des DNS-System insgesamt zu verbessern. In diesem Zusammenhang ist anzumerken, dass seit der Entwicklung und erstmaliger Einführung von DNSSEC bereits einige Angriffe abgewert wurden. Ein Fall, in dem ein Angriff auf DNSSEC erfolgreich abgeschwächt wurde, ist der „Cache Poisoning“-Angriff: Um Benutzer auf bösartige Websites oder Server umzuleiten, versucht ein Angreifer bei diesem Angriff, gefälschte DNS-Einträge in dem Cache ein DNS-Resolver einzuschleusen.

Mit DNSSEC kann der Resolver mithilfe kryptografischer Signaturen die Echtheit von DNS-Antworten überprüfen. Der Resolver erkennt die Diskrepanz zwischen den Signaturen und den tatsächlichen DNS-Datensätzen und weist die betrügerischen Daten zurück, wenn ein Angreifer versucht, betrügerische Datensätze in den Cache zu stecken.

Im erfolgreichen Angriffsszenario in Kapitel 6 konnte der Angreifer ohne Kenntnis des Clients Pakete oder Daten zwischen dem autoritativen Nameserver „ns.beispiel.de“ und der Client-Maschine sniffen und manipulieren.

Nach der Einführung von DNSSEC könnte der Angreifer zwar weiterhin Pakete mitlesen oder manipulieren, aber die Client-Maschine könnte den Angriff oder die Manipulation dieses Mal erkennen und abwehren, wie in Abbildung 37 dargestellt wird. Die Daten oder Informationen werden vom Nameserver verschlüsselt (Signatur), was eine zusätzliche Erklärung dafür darstellt.

Sobald die Informationen oder Daten ankommen, hat der Client die Möglichkeit, die Verschlüsselung (Signatur) mit dem Original zu vergleichen, indem er verschiedene Befehle wie `dig`, `delv`, `nslookup` einsetzt. So kann man manipulierte Daten oder Informationen unmittelbar ablehnen oder blockieren.



As was previously established, the attacker may still be able to read and manipulate the file. However, by using DNSSEC implementation (ZSK and KSK), the client is able to compare the signed files (documents, etc.) that have been received with the original signature of ns.pizza.com. If the signature is incorrect, the file will be rejected or blocked right away.

Abbildung 37: Angriffe Detektion mit Hilfe von DNSSEC

Wireshark (Netzwerk-Analyzer) erlaubt eine Analyse der DNS-Pakets zwischen dem Client und ns.pizza.com. Wie man es in Abbildung 38 sehen kann, die RRSI-Signatur von ns.beispiel.de sowie die Signatur Gültigkeit (Anfang & Ende) sind im Paket Capture klar zu erkennen. Diese Anweisungen ermöglichen es dem Client, das entsendete Paket mit der ursprünglichen Signatur auf dem autoritativen Name Server (ns.beispiel.de) zu vergleichen. Falls dies nicht stimmt, sollte die Kommunikation unverzüglich abgebrochen oder die Datei zurückgewiesen werden.

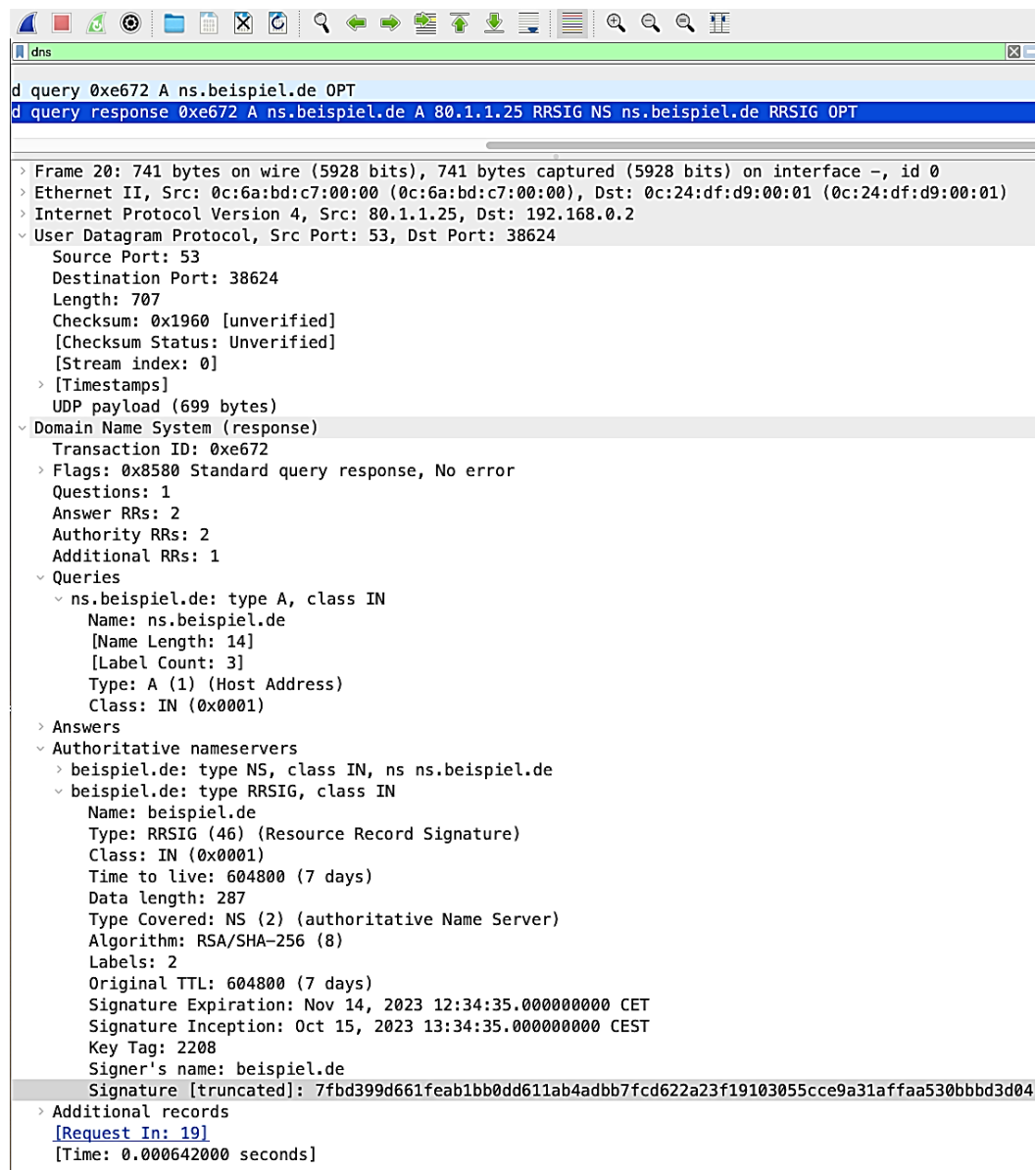


Abbildung 38: DNS Paket Capture mit Wireshark

## 9 Schlussbetrachtung

In diesem Kapitel werden die erarbeiteten Ergebnisse zusammengefasst und kritisch betrachtet. Außerdem wird ein Ausblick in die Zukunft von DNS und DNS-Sicherheit gegeben.

### 9.1 Diskussion

Im Rahmen dieser Arbeit hatte ich die Möglichkeit, DNSSEC zu analysieren und in einer Emulationsumgebung mit Docker-Containern verschiedene Angriffe auf DNS-Protokolle zu demonstrieren. Dabei konnten ich aufzeigen, welche Unterschiede bestehen, wenn DNSSEC implementiert ist und wenn nicht. DNSSEC bietet Schutz für Domain-Namen vor diversen DNS-Angriffen. Ein Beispiel hierfür ist Cache Poisoning, ein schwer abzuwehrender Angriff, der erheblichen Schaden anrichten kann. Richtig implementiert verhindert DNSSEC Cache Poisoning Angriffe, da Angreifer nicht länger autoritative Name-server imitieren können.

Diese Arbeit hat außerdem gezeigt, dass die Kommunikation zwischen der Client-Maschine (DNS Stub-Resolver) und dem rekursiven DNS-Resolver von einem Angreifer abgefangen, verändert oder unterbrochen werden kann. Die daraus gewonnenen Informationen können für weitere Angriffe verwendet werden.

Daher ist es besonders im Unternehmensumfeld wichtig, diesen Kommunikationskanal abzusichern, da DNS ein unverschlüsseltes Protokoll ist. Eine individuelle Risikoabschätzung für jeden Anwendungsfall ist hierbei unerlässlich.

Darüber hinaus können alle Angriffe und Sicherheits-Szenarien (wie DNSSEC) in einer Netzwerkemulationsumgebung wie GNS3 simuliert werden.

Durch die Nutzung von GNS3 können Netzwerkingenieure ihre Fähigkeiten erheblich verbessern, indem sie praktische Erfahrungen sammeln, Netzwerke kostengünstig simulieren, sich auf Zertifizierungen vorbereiten und kontinuierlich neue Technologien erlernen und testen.

Die initiale Phase, welche die konzeptuelle Untersuchung von DNS und seiner Sicherheit mithilfe von DNSSEC in einer Netzwerk-Emulationsumgebung (GNS3) umfasste, ist nun abgeschlossen. Zukünftige Arbeiten könnten die Einbindung von DoT oder DoH in Betracht ziehen.

## 9.2 Limitationen

Diese Arbeit ist nicht frei von Einschränkungen, da die implementierten Mechanismen in dieser Emulationsumgebung auf DNSSEC basieren. DNSSEC (Domain Name System Security Extension) nutzt sowohl die Authentifizierung als auch Integrität, um die Sicherheit des DNS (Domain Name System) zu verbessern. Daher sollte in zukünftigen Forschungen untersucht werden, inwieweit andere Sicherheitsmaßnahmen oder Sicherheitsprotokolle auf der Transportschicht eingesetzt werden können. Die Ergebnisse der abgeschwächten DNS-Angriffe könnten für weitere Forschungen mit anderen DNS-Sicherheitsmechanismen wie beispielsweise DNS over TLS oder DNS over HTTPS genutzt werden.

## 9.3 Fazit

Ziel dieser Arbeit ist es, DNSSEC aufzuarbeiten und in einer aufzubauenden Emulationsumgebung zu demonstrieren. In einer vergleichenden Betrachtung soll anschließend DNSSEC gegenüber anderen Absicherungsverfahren – DNS over TLS, DNS over HTTPS, DANE – abgegrenzt und beurteilt werden.

Es wurde aufgezeigt, dass DNSSEC einen Signaturprozess beinhaltet. Im Kern verwendet DNSSEC digitale Signaturen, um die Authentizität und Integrität von DNS-Daten sicherzustellen. Dieser Prozess besteht aus vier wichtigen Stufen: Signaturerstellung, Verteilung Öffentlicher Schlüssel, DNSSEC Validierung und Kette des Vertrauens.

Es wurde gezeigt, dass die DNS-Infrastruktur generell anfällig ist und für verschiedene Arten von Angriffen missbraucht werden kann. Somit kann die erste Forschungsfrage „Wie kann ein Angreifer die Sicherheit einer DNS-Infrastruktur gefährden“ als beantwortet angesehen werden.

Außerdem ist ersichtlich, dass DNSSEC allein keinen umfassenden Schutz bietet und eine komplexe Implementierung erfordert. Es wurden zusätzliche Herausforderungen und potenzielle Risiken identifiziert, wie beispielsweise die Sicherheit von Daten während ihrer Übertragung, um Manipulation oder das Abfangen von Paketen zu verhindern.

Deshalb ist es notwendig, DNS-Sicherheitserweiterungen einzusetzen, um einen effektiven Schutz zu gewährleisten.

Abschließend wurde demonstriert, dass DNSSEC ein entscheidendes Sicherheitsprotokoll ist, um die DNS-Infrastruktur vor Cyberangriffen zu schützen. Um eine umfassende

Sicherheit zu gewährleisten, könnten zusätzliche Sicherheitsprotokolle wie DoT (DNS over TLS), DoH (DNS over HTTPS) oder DANE implementiert werden. Alternativ könnte DNSSEC mit weiteren Sicherheitsmaßnahmen wie SPF (Sender Policy Framework) zur Bekämpfung von Spam oder DKIM (DomainKeys Identified Mail) zur Überprüfung und Authentifizierung von E-Mails kombiniert werden.

## 9.4 Ausblick

In dieser Studie wurde DNS als alleiniges System für die Namensauflösung untersucht. Dennoch gibt es Alternativen zum DNS, wie zum Beispiel Namecoin oder das GNU Name System (GNS), die in weiteren Forschungen berücksichtigt werden können. Namecoin ist eine Kryptowährung mit dem Ziel, eine von der ICANN unabhängige und dezentrale Namensauflösung anzubieten, wobei der Schutz der Privatsphäre der Nutzer im Vordergrund steht. Der Namensraum umfasst die inoffizielle TLD *.bit*, die nur über Namecoin erreichbar sind (61). Im Gegensatz dazu beruht GNS nicht auf einer hierarchischen Struktur, sondern auf einem gerichteten Graphen. Der Namensraum von GNS umfasst die inoffizielle TLD *.gnu* (62). Diese beiden Alternativen könnten im Zusammenhang mit den Ergebnissen dieser Arbeit betrachtet werden. Zum Beispiel könnte untersucht werden, ob diese Systeme denselben Bedrohungen ausgesetzt sind(48).

Darüber hinaus sollte für DNSSEC ein automatisierter Prozess zur regelmäßigen Erneuerung der Verschlüsselungsschlüssel KSK und ZSK entwickelt werden. Ohne einen solchen Prozess muss der IT-Administrator die Schlüssel alle zwei Wochen oder zwei Monate manuell erneuern. Ein ähnlicher Automatisierungsprozess wurde bereits für die Zertifikatserneuerung zwischen „Let's Encrypt“ und NGINX mithilfe der Certbot-Software erfolgreich implementiert und läuft seit mehreren Jahren. Die „Let's Encrypt“-Zertifikate sind 90 Tage gültig und werden regelmäßig durch einen Cron-Job überprüft und erneuert. Es ist jedoch wichtig, den DNSSEC-Validierungsprozess zu verbessern, da die dahinterliegenden kryptografischen Verfahren die Implementierung für IT-Spezialisten und Administratoren erschweren.

## Literaturverzeichnis

1. Aufgabenstellung für die Bachelorarbeit.pdf.
2. Hornetsecurity [Internet]. [zitiert 27. Dezember 2023]. What is IT Security? - Definition and measures! Verfügbar unter: <https://www.hornetsecurity.com/en/knowledge-base/it-security/>
3. Eckert C. IT-Sicherheit: Konzepte, Verfahren, Protokolle. 10. Auflage. München: De Gruyter Oldenburg; 2018. 1004 S. (De Gruyter studium).
4. Eckert C. IT-Sicherheit: Konzepte – Verfahren – Protokolle. In: IT-Sicherheit [Internet]. De Gruyter Oldenburg; 2018 [zitiert 19. April 2024]. Verfügbar unter: <https://www.degruyter.com/document/doi/10.1515/9783110563900/html>
5. Bundesanzeiger Verlag GmbH. Buchkatalog. 2018 [zitiert 1. Juni 2024]. IT-Grundschutz-Kompendium. Verfügbar unter: <https://www.buchkatalog.de/product/3000000714774>
6. MOCKAPETRIS, PV. Domain names - implementation and specification [Internet]. Internet Engineering Task Force; 1987 Nov [zitiert 12. Januar 2024]. Report No.: RFC 1035. Verfügbar unter: <https://datatracker.ietf.org/doc/rfc1035>
7. Meinel, Christoph: Meinels Web-Tutorial: Das Domain Name System - das Telefonbuch des Internets. – URL <https://www.spektrum.de/kolumne/das-domain-name-system-ist-das-telefonbuch-des-internets/1732674>. – Zugriffsdatum: 2024-01-21.
8. RFC Editor. Domain names - concepts and facilities [Internet]. Internet Engineering Task Force; 1987 Nov [zitiert 24. Januar 2024]. Report No.: RFC 1034. Verfügbar unter: <https://datatracker.ietf.org/doc/rfc1034>
9. Domain names - implementation and specification [Internet]. Internet Engineering Task Force; 1987 Nov [zitiert 24. Januar 2024]. Report No.: RFC 1035. Verfügbar unter: <https://datatracker.ietf.org/doc/rfc1035>
10. Hoffman PE, Sullivan A, Fujiwara K. DNS Terminology [Internet]. Internet Engineering Task Force; 2019 Jan [zitiert 24. Januar 2024]. Report No.: RFC 8499. Verfügbar unter: <https://datatracker.ietf.org/doc/rfc8499>
11. RFC Editor 920. Domain requirements [Internet]. Internet Engineering Task Force; 1984 Okt [zitiert 24. Januar 2024]. Report No.: RFC 920. Verfügbar unter: <https://datatracker.ietf.org/doc/rfc920>
12. Baun C. Bitübertragungsschicht. In: Computernetze kompakt: Eine an der Praxis orientierte Einführung für Studium und Berufspraxis [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2020. S. 45–94. verfügbar unter: [https://doi.org/10.1007/978-3-662-59897-9\\_5](https://doi.org/10.1007/978-3-662-59897-9_5)
13. Baun C. Computernetze kompakt: Eine an der Praxis orientierte Einführung für



- Studium und Berufspraxis [Internet]. Berlin, Heidelberg: Springer; 2020 [zitiert 24. Januar 2024]. (IT kompakt). Verfügbar unter: <http://link.springer.com/10.1007/978-3-662-59897-9>
14. Baun C. Computernetze kompakt: Eine an der Praxis orientierte Einführung für Studium und Berufspraxis [Internet]. Berlin, Heidelberg: Springer; 2020 [zitiert 24. Januar 2024]. (IT kompakt). Verfügbar unter: <http://link.springer.com/10.1007/978-3-662-59897-9>
  15. Rooney T, Dooley M. IP address management. Second edition. Hoboken, New Jersey: Wiley; 2021. (IEEE press series on networks and service management).
  16. Interactive Problems, Computer Networking: A Top-Down Approach [Internet]. [zitiert 24. Januar 2024]. Verfügbar unter: [https://gaia.cs.umass.edu/kur-ose\\_ross/interactive/dns\\_query.php](https://gaia.cs.umass.edu/kur-ose_ross/interactive/dns_query.php)
  17. Recursive DNS queries [Internet]. [Zitiert 18. Mai 2024]. Verfügbar unter: <https://learning.oreilly.com/library/view/networking-fundamentals/9781838643508/feb09b24-eec0-4b55-9321-334e58262923.xhtml>
  18. Southam M. DNSSEC: What it is and why it matters. Network Security [Internet]. 1. Mai 2014 [zitiert 22. April 2024];2014(5):12–5. verfügbar unter: <https://www.sciencedirect.com/science/article/pii/S1353485814700509>
  19. What is DANE? [Internet]. Infoblox. [zitiert 27. April 2024]. Verfügbar unter: <https://www.infoblox.com/dns-security-resource-center/dns-security-faq/what-is-dane/>
  20. Ayoub I, Berthaud-Müller G, Balakrichenan S, Khawam K, Ampeau B. The DNS to Reinforce the PKIX for IoT Backend Servers: Implementation and Evaluation. In: 14th IFIP Wireless and Mobile Networking Conference [Internet]. Tunis, Tunisia; 2022 [zitiert 27. April 2024]. Verfügbar unter: <https://hal.science/hal-03798669>
  21. B.V N. Networking4all Cybersecurity Specialist. [zitiert 17. Mai 2024]. Leading in cybersecurity solutions. Verfügbar unter: <https://www.networking4all.com/en/support/faq/what-is-dane-and-dnssec>
  22. Xu und Trajkovi - Performance Analysis of RIP, EIGRP, and OSPF using.pdf [Internet]. [zitiert 30. April 2024]. Verfügbar unter: [https://www.sfu.ca/~ljilja/papers/Opnetwork2011\\_xu\\_final.pdf](https://www.sfu.ca/~ljilja/papers/Opnetwork2011_xu_final.pdf)
  23. Scribd [Internet]. [zitiert 12. Mai 2024]. OSPF-Thesis | PDF | Routing | Router (Computing). Verfügbar unter: <https://www.scribd.com/document/89162506/OSPF-Thesis>
  24. Snapshot [Internet]. [zitiert 10. Mai 2024]. Verfügbar unter: <https://www.computernetworkingnotes.com/ccna-study-guide/ospf-configuration-step-by-step-guide.html>

25. Computer Networking Notes [Internet]. [zitiert 10. Mai 2024]. How to configure OSPF Routing Protocol. Verfügbar unter: <https://www.computernetworkingnotes.com/ccna-study-guide/ospf-configuration-step-by-step-guide.html>
26. Computer Networking Notes [Internet]. [zitiert 12. Mai 2024]. How to configure OSPF Routing Protocol. Verfügbar unter: <https://www.computernetworkingnotes.com/ccna-study-guide/ospf-configuration-step-by-step-guide.html>
27. Eckert C. IT-Sicherheit: Konzepte, Verfahren, Protokolle. 10. Auflage. München: De Gruyter Oldenburg; 2018. 1004 S. (De Gruyter studium).
28. Hüsgen - DNS-Sicherheit am Beispiel eines mittelständischen.pdf.
29. Full Text PDF [Internet]. [zitiert 19. April 2024]. Verfügbar unter: <https://www.degruyter.com/document/doi/10.1515/9783110563900-fm/pdf>
30. Hüsgen K. DNS-Sicherheit am Beispiel eines mittelständischen Softwareunternehmens [Internet] [Thesis]. Hochschule für Angewandte Wissenschaften Hamburg; 2024 [zitiert 1. Juni 2024]. Verfügbar unter: <https://reposit.haw-hamburg.de/handle/20.500.12738/15298>
31. Snapshot [Internet]. [zitiert 3. Mai 2024]. Verfügbar unter: <https://www.techtarget.com/searchsecurity/definition/private-key>
32. Security [Internet]. [zitiert 3. Mai 2024]. What is a private key? Verfügbar unter: <https://www.techtarget.com/searchsecurity/definition/private-key>
33. Pednekar - © UNIVERSITY OF MUMBAI.pdf [Internet]. [zitiert 3. Juni 2024]. Verfügbar unter: <http://mu.ac.in/wp-content/uploads/2022/05/information-security.pdf>
34. Snapshot [Internet]. [zitiert 7. Mai 2024]. Verfügbar unter: <https://utimaco.com/service/knowledge-base/digital-signing/digital-signature>
35. McClay J. What is a Cryptographic Hash? – Question Computer [Internet]. 2012 [zitiert 8. Mai 2024]. Verfügbar unter: <https://www.questioncomputer.com/what-is-a-cryptographic-hash/>
36. Snapshot [Internet]. [zitiert 8. Mai 2024]. Verfügbar unter: <https://www.questioncomputer.com/what-is-a-cryptographic-hash/>
37. Chapter 8: X.509 Certificates and PKI [Internet]. [Zitiert 8. Mai 2024]. Verfügbar unter: [https://learning.oreilly.com/library/view/demystifying-cryptography-with/9781800560345/B16767\\_08\\_Final\\_AM.xhtml](https://learning.oreilly.com/library/view/demystifying-cryptography-with/9781800560345/B16767_08_Final_AM.xhtml)
38. Snapshot [Internet]. [zitiert 8. Mai 2024]. Verfügbar unter: [https://learning.oreilly.com/library/view/demystifying-cryptography-with/9781800560345/B16767\\_08\\_Final\\_AM.xhtml#\\_idParaDest-159](https://learning.oreilly.com/library/view/demystifying-cryptography-with/9781800560345/B16767_08_Final_AM.xhtml#_idParaDest-159)
39. Snapshot [Internet]. [zitiert 8. Mai 2024]. Verfügbar unter:

- [https://learning.oreilly.com/library/view/demystifying-cryptography-with/9781800560345/B16767\\_01\\_Final\\_AM.xhtml#\\_idParaDest-18](https://learning.oreilly.com/library/view/demystifying-cryptography-with/9781800560345/B16767_01_Final_AM.xhtml#_idParaDest-18)
40. Maxim\_Fomin\_Thesis\_d20m06.pdf [Internet]. [zitiert 20. Februar 2024]. Verfügbar unter: [https://www.theseus.fi/bitstream/handle/10024/132478/Maxim\\_Fomin\\_Thesis\\_d20m06.pdf?sequence=1](https://www.theseus.fi/bitstream/handle/10024/132478/Maxim_Fomin_Thesis_d20m06.pdf?sequence=1)
  41. GitHub [Internet]. [zitiert 21. Februar 2024]. [gns3-docs/docs/getting-started/what-is-gns3.md](https://github.com/mother/gns3-docs/blob/master/docs/getting-started/what-is-gns3.md) at master · mother/gns3-docs. Verfügbar unter: <https://github.com/mother/gns3-docs/blob/master/docs/getting-started/what-is-gns3.md>
  42. McClay J. Add Dockerized Bind DNS Server to GNS3 – Question Computer [Internet]. 2012 [zitiert 25. Februar 2024]. Verfügbar unter: <https://www.questioncomputer.com/add-dockerized-bind-dns-server-to-gns3/>
  43. Alvarez AE. DNS Forwarding and Conditional Forwarding [Internet]. Tech Jobs Academy. 2016 [zitiert 5. März 2024]. Verfügbar unter: <https://medium.com/tech-jobs-academy/dns-forwarding-and-conditional-forwarding-f3118bc93984>
  44. Forward DNS [Internet]. Techopedia. 2012 [zitiert 5. März 2024]. Verfügbar unter: <https://www.techopedia.com/definition/15671/forward-dns>
  45. bethan. SlideServe. 2014 [zitiert 26. Mai 2024]. PPT - DESIGNING THE DNS STRUCTURE PowerPoint Presentation, free download - ID:5436702. Verfügbar unter: <https://www.slideserve.com/bethan/designing-the-dns-structure>
  46. DNS Delegation: Concepts and Best Practices [Internet]. [zitiert 8. März 2024]. Verfügbar unter: <https://www.catchpoint.com/dns-monitoring/dns-delegation>
  47. Conti M, Dragoni N, Lesyk V. A Survey of Man In The Middle Attacks. IEEE Communications Surveys & Tutorials [Internet]. 2016 [zitiert 26. März 2024];18(3):2027–51. verfügbar unter: <https://ieeexplore.ieee.org/abstract/document/7442758>
  48. Hüsken K. DNS-Sicherheit am Beispiel eines mittelständischen Softwareunternehmens.
  49. Saxena und Sharma - 2017 - Analysis of Network Traffic by using Packet Sniffi.pdf [Internet]. [zitiert 29. März 2024]. Verfügbar unter: [https://d1wqtxts1xzle7.cloudfront.net/107795340/V3I6-1369-libre.pdf?1700896791=&response-content-disposition=inline%3B+filename%3DAnalysis\\_of\\_Network\\_Traffic\\_by\\_Using\\_Pac.pdf&Expires=1711694990&Signature=YKxMmmM3hTegUdShu9F7bwSbY-SuCDrRuzqRWzhL1SyKIMZI9If-MiesqDfC8GsEMRjQRyZIlWzm2nMrGB9mvDM11~7Q85WT9RskFZFANr-PunWq5gabWXdAgs03r5jLNb2fZnWkUmfNqL1fm1BwlCO3yzNAY24IAZr-fUJ5jWnk5vNeyRhNTe0iNGfMVRisd1qXYCUFv-eYVPzwjBUAkx7XnDK7-WAGYaVAi8qqqEtTNa9Ylzei08dYeJWFAgT6FwRmj5Us~IBdX92KJHkaO-Bym8RR9IHJcS2fb6beUZXYTC9WXTdlZxb9dpYe6UGtCJ8Zrww3fEWpXUH](https://d1wqtxts1xzle7.cloudfront.net/107795340/V3I6-1369-libre.pdf?1700896791=&response-content-disposition=inline%3B+filename%3DAnalysis_of_Network_Traffic_by_Using_Pac.pdf&Expires=1711694990&Signature=YKxMmmM3hTegUdShu9F7bwSbY-SuCDrRuzqRWzhL1SyKIMZI9If-MiesqDfC8GsEMRjQRyZIlWzm2nMrGB9mvDM11~7Q85WT9RskFZFANr-PunWq5gabWXdAgs03r5jLNb2fZnWkUmfNqL1fm1BwlCO3yzNAY24IAZr-fUJ5jWnk5vNeyRhNTe0iNGfMVRisd1qXYCUFv-eYVPzwjBUAkx7XnDK7-WAGYaVAi8qqqEtTNa9Ylzei08dYeJWFAgT6FwRmj5Us~IBdX92KJHkaO-Bym8RR9IHJcS2fb6beUZXYTC9WXTdlZxb9dpYe6UGtCJ8Zrww3fEWpXUH)

--28NTHDhQ\_\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

50. LINUXMAKER [Internet]. 2024 [zitiert 29. Mai 2024]. Konfiguration von DNSSEC. Verfügbar unter: <https://www.linuxmaker.com/linux/bind9-mit-dnssec-absichern/konfiguration-von-dnssec.html>
51. Konfiguration von DNSSEC [Internet]. [zitiert 6. April 2024]. Verfügbar unter: <https://www.linuxmaker.com/linux/bind9-mit-dnssec-absichern/konfiguration-von-dnssec.html>
52. Konfiguration von DNSSEC [Internet]. [zitiert 6. April 2024]. Verfügbar unter: <https://www.linuxmaker.com/linux/bind9-mit-dnssec-absichern/konfiguration-von-dnssec.html>
53. Amberg E. Linux Magazine. [zitiert 27. Mai 2024]. Chain of Trust - Page: 1.3 » Linux Magazine. Verfügbar unter: <http://www.linux-magazine.com/Issues/2008/90/DNSSEC>
54. Dooley Michael, Timothy R. DNS Security Management [Internet]. 1. Aufl. Wiley; 2017 [zitiert 31. Mai 2024]. Verfügbar unter: <https://online-library.wiley.com/doi/book/10.1002/9781119328292>
55. Agouros K. DNS, DHCP [Internet]. [zitiert 4. Juni 2024]. München Open Source Pr. 2007: Open Source Pr.; 2007. 376 S. : Ill. (Root's reading). Verfügbar unter: [https://katalog.ulb.hhu.de/digitale-objekte-hbz/storage/2007/11/22/file\\_73/2190335.pdf](https://katalog.ulb.hhu.de/digitale-objekte-hbz/storage/2007/11/22/file_73/2190335.pdf)
56. How To Use Dig, Whois, & Ping on an Ubuntu VPS to Query DNS Data | DigitalOcean [Internet]. [zitiert 21. Mai 2024]. Verfügbar unter: <https://www.digitalocean.com/community/tutorials/how-to-use-dig-whois-ping-on-an-ubuntu-vps-to-query-dns-data>
57. Chouhan RS. How DNS Works — DNS Zone Transfer [Internet]. Medium. 2023 [zitiert 17. Mai 2024]. Verfügbar unter: <https://bot2root.medium.com/how-dns-works-dns-zone-transfer-a7ac53f5b21e>
58. Konfiguration von DNSSEC [Internet]. [zitiert 6. April 2024]. Verfügbar unter: <https://www.linuxmaker.com/linux/bind9-mit-dnssec-absichern/konfiguration-von-dnssec.html>
59. Mariusz H. CodiLime. 2024 [zitiert 22. Mai 2024]. Introduction to DNSSEC. Verfügbar unter: <https://codilime.com/blog/introduction-to-dnssec/>
60. IT\_Grundschatz\_Kompndium\_Edition2020.pdf [Internet]. [zitiert 19. April 2024]. Verfügbar unter: [https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschatz/Kompndium/IT\\_Grundschatz\\_Kompndium\\_Edition2020.pdf%3F\\_\\_blob%3DpublicationFile%26v%3D6](https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschatz/Kompndium/IT_Grundschatz_Kompndium_Edition2020.pdf%3F__blob%3DpublicationFile%26v%3D6)
61. Namecoin [Internet]. [zitiert 3. Juni 2024]. Verfügbar unter: <https://www.namecoin.org/>

62. GNUnet [Internet]. [zitiert 3. Juni 2024]. Verfügbar unter:  
<https://www.gnunet.org/de/gns.html>

## Anhang

### OSPF Routing Configuration

---

Router1>enable

Router1#configure terminal

Router1(config)#router ospf 1

Router1(config-router)#network 60.1.1.1 0.255.255.255 area 0

Router1(config-router)#network 12.1.2.2 0.255.255.255 area 0

Router1(config-router)#network 40.1.1.1 0.255.255.255 area 0

Router1(config-router)#exit

---

Router2>enable

Router2#configure terminal

Router2(config)#router ospf 2

Router2(config-router)#network 50.1.1.2 0.255.255.255 area 0

Router2(config-router)#network 12.1.2.2 0.255.255.255 area 0

Router2(config-router)#exit

Router2(config)#exit

Router2#

---

Router3>enable

Router3#config

Router3(config)#router ospf 3

Router3(config-router)#network 50.1.1.2 0.255.255.255 area 0

Router3(config-router)#network 40.1.1.1 0.255.255.255 area 0

Router3(config-router)#network 80.0.0.1 0.255.255.255 area 0

Router3(config-router)#exit

Router3(config)#exit

Router3#

**Konfiguration von DNSSEC auf BIND9**

```

mkdir -p /etc/bind/keys/
chown -R bind:bind /etc/bind/keys/
#-----

dnssec-keygen -3 -a RSASHA512 -b 4096 -n ZONE -r /dev/urandom -f KSK example.tld
Kexample.tld.+010+64413.key (DNSKEY Record),

Kexample.tld.+010+64413.private,
dem privaten Schlüssel zur Signatur des öffentlichen ZSK.
Der DNSKEY Record im öffentlichen Schlüssel sieht etwa so aus:
cat Kexample.tld.+010+64413.key
; This is a key-signing key, keyid 64413, for example.tld.
; Created: 20200313175118 (Fri Mar 13 18:51:18 2020)
; Publish: 20200313175118 (Fri Mar 13 18:51:18 2020)
; Activate: 20200313175118 (Fri Mar 13 18:51:18 2020)
example.tld. IN DNSKEY 257 3 10 AwE-
AAaoJiFk+Xj+9v[...]KTMN4QNqRm8l1RgnDq/2GvrxmSPXfGKyWm
fjvSJot4EX2i97ER
#-----
--
Zone Signing Key (ZSK) generieren
Den Zone Signing Key erzeugt man im Anschluß daran wie folgt, gefolgt von einer
Neuzuordnung der Dateiattribute.

dnssec-keygen -3 -a NSEC3RSASHA1 -b 2048 -n ZONE -r /dev/urandom example.tld
chown -R bind:bind /etc/bind/keys/
#-----
Kexample.tld.+007+40385.key,
der öffentliche Schlüssel (DNSKEY Record), der dem Verifizieren der DNS-Antworten
dient.
Kexample.tld.+007+40385.private,
#-----
cat Kexample.tld.+007+40385.key
; This is a zone-signing key, keyid 40385, for example.tld.
; Created: 20200313175211 (Fri Mar 13 18:52:11 2020)
; Publish: 20200313175211 (Fri Mar 13 18:52:11 2020)
; Activate: 20200313175211 (Fri Mar 13 18:52:11 2020)
example.tld. IN DNSKEY 256 3 7 AwEAAuXGNQMitXTRPKAkme9[...]Mqi3ZvO
0yn19+nxfb8=
#-----
for key in `ls Kexample.tld.*.key`
do
echo "$INCLUDE /etc/bind/keys/$key">> /var/cache/bind/tld/db.example.tld
done
oder einzeln pro Zonendatei mit einer vi-Funktion:
:r! ls /etc/bind/keys/Kexample.tld.*.key

```

Durch das Signieren wird gleichzeitig eine neue Zonendatei mit dem Suffix ".signed" erzeugt.

```
dnssec-signzone -3 `head -c 512 /dev/urandom | sha1sum | cut -b 1-16` -z -H 330 -K  
/etc/bind/keys -t -o example.tld  
/var/cache/bind/tld/db.example.tld
```

Verifying the zone using the following algorithms: NSEC3RSASHA1 RSASHA512.

Zone fully signed:

Algorithm: NSEC3RSASHA1: KSKs: 0 active, 0 stand-by, 0 revoked

ZSKs: 1 active, 0 stand-by, 0 revoked

Algorithm: RSASHA512: KSKs: 1 active, 0 stand-by, 0 revoked

ZSKs: 0 active, 0 stand-by, 0 revoked

/var/cache/bind/tld/db.example.tld.signed

Signatures generated: 26

Signatures retained: 0

Signatures dropped: 0

Signatures successfully verified: 0

Signatures unsuccessfully verified: 0

Signing time in seconds: 0.315

Signatures per second: 82.418

Runtime in seconds: 0.358

chown -R bind:bind /var/cache/bind/

#-----

dnssec-enable yes;

dnssec-validation yes;

dnssec-lookaside auto;

key-directory "/etc/bind/keys/";

-----

zone "example.tld" {

type master;

file "tld/db.example.tld";

allow-transfer {

"slaves";

};

notify yes;

};

und startet den BIND9 neu

systemctl restart bind9.service



**Dockerisierten Bind-Server zu GNS3 hinzufügen und DNS Konfiguration**

```
mkdir jamesbind
cd jamesbind
```

```
vi Dockerfile
```

```
FROM internetsystemsconsortium/bind9:9.11
RUN apt-get update
RUN apt-get install vim -y
```

```
docker build -t jamesbind .
```

```
options {
directory "/var/cache/bind";
listen-on { any; };
};
```

```
mkdir zones
cd zones
vi db.jamesmcclay.com
```

```
zone "jamesmcclay.com" {
type master;
file "/etc/bind/zones/db.jamesmcclay.com";
};
```

```
@ IN SOA ns.jamesmcclay.com. root.jamesmcclay.com. (
2 ; Serial
604800 ; Refresh
86400 ; Retry
2419200 ; Expire
604800 ) ; Negative Cache TTL
;
```

```
@ IN NS ns.jamesmcclay.com.
ns IN A 10.0.0.3
alpine1 IN A 10.0.0.1
alpine2 IN A 10.0.0.2
```

```
named -g
```

```
<...removed for brevity...>
```

```
26-Oct-2021 23:49:14.231 zone jamesmcclay.com/IN: loaded serial 2
26-Oct-2021 23:31:49.828 all zones loaded
26-Oct-2021 23:31:49.829 running
```

## Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder der Verfasserin/des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Anmerkung: In einigen Studiengängen findet sich die Erklärung unmittelbar hinter dem Deckblatt der Arbeit.

---

Ort, Datum

---

Unterschrift