CIA Lab Assignment: Domain Name System (1)

2.1 Validating the Download

Question 1. Why is it wise to use a signature to check your download?

Because it might happen that a certain installation package we download from a server was created or modified by a hacker who discovered a vulnerability and alter the installation packages stored on that server, or it might alter any of these files on route to us. That's why we should always verify the integrity and authenticity of each file or package we download in order to make sure that the files was created by trusted source and it was not altered by any hacker.

Source: 1- https://veracrypt.codeplex.com/wikipage?title=Digital%20Signatures

Question 2. Which signature is the best one to use? Why?

BIND these types of signature:

```
1-ASC
2-SHA1
3-SHA256
4-SHA512
```

Check its validity of ASC:

Using OpenPGP encryption and signing tool we use –verify option

```
kotaiba@bristol:~$ gpg --verify bind-9.10.6.tar.gz

gpg: directory `/home/kotaiba/.gnupg' created
gpg: new configuration file `/home/kotaiba/.gnupg/gpg.conf' created
gpg: WARNING: options in `/home/kotaiba/.gnupg/gpg.conf' are not yet active
during this run
gpg: keyring `/home/kotaiba/.gnupg/pubring.gpg' created
gpg: no valid OpenPGP data found.
gpg: the signature could not be verified.
Please remember that the signature file (.sig or .asc)
should be the first file given on the command line.


kotaiba@bristol:~$ gpg --verify bind-9.10.6.tar.gz.asc
gpg: assuming signed data in `bind-9.10.6.tar.gz'
gpg: Signature made Thu 27 Jul 2017 11:10:12 AM CEST using RSA key ID
5CF02E57
gpg: Can't check signature: public key not found
```

Before I can check the signature I have to get the public key of key ID 5CF02E57

```
kotaiba@bristol:~$ gpg --keyserver pgp.mit.edu --recv-keys 5CF02E57

gpg: keyring `/home/kotaiba/.gnupg/secring.gpg' created
gpg: requesting key 5CF02E57 from hkp server pgp.mit.edu
```

```
gpg: /home/kotaiba/.gnupg/trustdb.gpg: trustdb created
gpg: key 5CF02E57: public key "Internet Systems Consortium, Inc. (Signing
key, 2017-2018) <codesign@isc.org>" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
```

It gave me:

```
kotaiba@bristol:~$ gpg --verify bind-9.10.6.tar.gz.asc

gpg: assuming signed data in `bind-9.10.6.tar.gz'
gpg: Signature made Thu 27 Jul 2017 11:10:12 AM CEST using RSA key ID
5CF02E57
gpg: Good signature from "Internet Systems Consortium, Inc. (Signing key,
2017-2018) <codesign@isc.org>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the
owner.
Primary key fingerprint: BE0E 9748 B718 253A 28BB  89FF F1B1 1BF0 5CF0 2E57
```

The question now is which signature is the best one to use and Why?

BIND has 4 types of signatures: SHA1, SHA256, SHA512, and ASC. The first 3 are hashing algorithm which care only for the integrity of the file and verify that no one tempered of modified the data inside the file. However, what if an attacker got access to that server(that store the package of file) and put his own file and signature. In, this case we can't verify the owner of the file. On the other hand, ASC is PGP electronic signature. Instead of having a signature that an attacker may replace with his own. the PGP solves this issue. It support authentication and integrity checking. The former to determine whether it was actually uploaded to the server by the person or company that own this file and the latter is used to detect if someone modified it since it was completed. That's why the ASC is the best one.

AFTER FEEDBACK

To give you an exact and specific answer I think the best signature is SHA-512 because it has the largest search space if anyone wanted to do a brute force attack; however, SHA-256 is somehow secure since the search space is also completely infeasible and the algorithms are of the same family in addition to that it is more efficient. But SHA-1 is really a bad idea, since it is somehow already broken and its attacks are known.

Sources:

1- https://deepthought.isc.org/article/AA-00768/0/Getting-started-with-BIND-how-to-build-and-run-named-with-a-basic-recursive-configuration.html

2- https://www.linuxquestions.org/questions/linux-security-4/gpg-can%27t-check-signature-public-key-not-found-when-decrypting-file-4175494608/

3- https://www.isc.org/downloads/

4- https://learningnetwork.cisco.com/thread/63610

5- https://en.wikipedia.org/wiki/Pretty_Good_Privacy

Question 3. Why are caching-only name servers still useful?

First, what is caching-only servers ? A caching-only server saves data in a cache file until the data expires. Expiration occurs based on a “time-to-live” field attached to data received from another server. It answers data from its cache if it has the information, or requests it from authoritative servers if it does not.

Why it is still useful? because it speeds the performance of DNS resolution, improve reliability, and reduce DNS-related query traffic. In addition to that is they do not generate zone transfer network traffic because they do not contain any zones. It has only on disadvantage which is when the server is started it has no cached information and it takes time to build up this information as it service requests.

Source:

1- https://technet.microsoft.com/en-us/library/cc958964.aspx

2- http://osr507doc.xinuos.com/en/NetAdminG/dnsD.cacheserv.html

Question 4. Now that you know all the elements of the main configuration, create a simple named.conf or unbound.conf for a caching-only name server. Show the configuration file in your log.

```
#
# Example configuration file.
#
# See unbound.conf(5) man page, version 1.6.6.
#
# this is a comment.

#Use this to include other text into the file.
#include: "otherfile.conf"

# The server clause sets the main parameters.
server:
    # whitespace is not necessary, but looks cleaner.

    # verbosity number, 0 is least verbose. 1 is default.
    verbosity: 1

    # print statistics to the log (for every thread) every N seconds.
    # Set to "" or 0 to disable. Default is disabled.
    # statistics-interval: 0

    # enable shm for stats, default no.  if you enable also enable
    # statistics-interval, every time it also writes stats to the
    # shared memory segment keyed with shm-key.
    # shm-enable: no
```

```
    # shm for stats uses this key, and key+1 for the shared mem segment.
    # shm-key: 11777

    # enable cumulative statistics, without clearing them after printing.
    # statistics-cumulative: no

    # enable extended statistics (query types, answer codes, status)
    # printed from unbound-control. default off, because of speed.
    # extended-statistics: no

    # number of threads to create. 1 disables threading.
    # num-threads: 1

    # specify the interfaces to answer queries from by ip-address.
    # The default is to listen to localhost (127.0.0.1 and ::1).
    # specify 0.0.0.0 and ::0 to bind to all available interfaces.
    # specify every interface[@port] on a new 'interface:' labelled line.
    # The listen interfaces are not changed on reload, only on restart.
    # interface: 192.0.2.153
    # interface: 192.0.2.154
    # interface: 192.0.2.154@5003
    # interface: 2001:DB8::5

    # enable this feature to copy the source address of queries to reply.
    # Socket options are not supported on all platforms. experimental.
    # interface-automatic: no

    # port to answer queries from
    # port: 53

    # specify the interfaces to send outgoing queries to authoritative
    # server from by ip-address. If none, the default (all) interface
    # is used. Specify every interface on a 'outgoing-interface:' line.
    # outgoing-interface: 192.0.2.153
    # outgoing-interface: 2001:DB8::5
    # outgoing-interface: 2001:DB8::6

    # Specify a netblock to use remainder 64 bits as random bits for
    # upstream queries.  Uses freebind option (Linux).
    # outgoing-interface: 2001:DB8::/64
    # Also (Linux:) ip -6 addr add 2001:db8::/64 dev lo
    # And: ip -6 route add local 2001:db8::/64 dev lo
    # And set prefer-ip6: yes to use the ip6 randomness from a netblock.
    # Set this to yes to prefer ipv6 upstream servers over ipv4.
    # prefer-ip6: no

    # number of ports to allocate per thread, determines the size of the
    # port range that can be open simultaneously.  About double the
    # num-queries-per-thread, or, use as many as the OS will allow you.
    # outgoing-range: 4096
```

```
    # permit unbound to use this port number or port range for
    # making outgoing queries, using an outgoing interface.
    # outgoing-port-permit: 32768

    # deny unbound the use this of port number or port range for
    # making outgoing queries, using an outgoing interface.
    # Use this to make sure unbound does not grab a UDP port that some
    # other server on this computer needs. The default is to avoid
    # IANA-assigned port numbers.
    # If multiple outgoing-port-permit and outgoing-port-avoid options
    # are present, they are processed in order.
    # outgoing-port-avoid: "3200-3208"

    # number of outgoing simultaneous tcp buffers to hold per thread.
    # outgoing-num-tcp: 10

    # number of incoming simultaneous tcp buffers to hold per thread.
    # incoming-num-tcp: 10

    # buffer size for UDP port 53 incoming (SO_RCVBUF socket option).
    # 0 is system default.  Use 4m to catch query spikes for busy servers.
    # so-rcvbuf: 0

    # buffer size for UDP port 53 outgoing (SO_SNDBUF socket option).
    # 0 is system default.  Use 4m to handle spikes on very busy servers.
    # so-sndbuf: 0

    # use SO_REUSEPORT to distribute queries over threads.
    # so-reuseport: no

    # use IP_TRANSPARENT so the interface: addresses can be non-local
    # and you can config non-existing IPs that are going to work later on
    # (uses IP_BINDANY on FreeBSD).
    # ip-transparent: no

    # use IP_FREEBIND so the interface: addresses can be non-local
    # and you can bind to nonexisting IPs and interfaces that are down.
    # Linux only.  On Linux you also have ip-transparent that is similar.
    # ip-freebind: no

    # EDNS reassembly buffer to advertise to UDP peers (the actual buffer
    # is set with msg-buffer-size). 1472 can solve fragmentation (timeouts)
    # edns-buffer-size: 4096

    # Maximum UDP response size (not applied to TCP response).
    # Suggested values are 512 to 4096. Default is 4096. 65536 disables it.
    # max-udp-size: 4096

    # buffer size for handling DNS data. No messages larger than this
    # size can be sent or received, by UDP or TCP. In bytes.
    # msg-buffer-size: 65552
```

```
    # the amount of memory to use for the message cache.
    # plain value in bytes or you can append k, m or G. default is "4Mb".
    # msg-cache-size: 4m

    # the number of slabs to use for the message cache.
    # the number of slabs must be a power of 2.
    # more slabs reduce lock contention, but fragment memory usage.
    # msg-cache-slabs: 4

    # the number of queries that a thread gets to service.
    # num-queries-per-thread: 1024

    # if very busy, 50% queries run to completion, 50% get timeout in msec
    # jostle-timeout: 200

    # msec to wait before close of port on timeout UDP. 0 disables.
    # delay-close: 0

    # the amount of memory to use for the RRset cache.
    # plain value in bytes or you can append k, m or G. default is "4Mb".
    # rrset-cache-size: 4m

    # the number of slabs to use for the RRset cache.
    # the number of slabs must be a power of 2.
    # more slabs reduce lock contention, but fragment memory usage.
    # rrset-cache-slabs: 4

    # the time to live (TTL) value lower bound, in seconds. Default 0.
    # If more than an hour could easily give trouble due to stale data.
    # cache-min-ttl: 0

    # the time to live (TTL) value cap for RRsets and messages in the
    # cache. Items are not cached for longer. In seconds.
    # cache-max-ttl: 86400

    # the time to live (TTL) value cap for negative responses in the cache
    # cache-max-negative-ttl: 3600

    # the time to live (TTL) value for cached roundtrip times, lameness and
    # EDNS version information for hosts. In seconds.
    # infra-host-ttl: 900

    # minimum wait time for responses, increase if uplink is long. In msec.
    # infra-cache-min-rtt: 50

    # the number of slabs to use for the Infrastructure cache.
    # the number of slabs must be a power of 2.
    # more slabs reduce lock contention, but fragment memory usage.
    # infra-cache-slabs: 4

    # the maximum number of hosts that are cached (roundtrip, EDNS, lame).
```

```
    # infra-cache-numhosts: 10000

    # define a number of tags here, use with local-zone, access-control.
    # repeat the define-tag statement to add additional tags.
    # define-tag: "tag1 tag2 tag3"

    # Enable IPv4, "yes" or "no".
    # do-ip4: yes

    # Enable IPv6, "yes" or "no".
    # do-ip6: yes

    # Enable UDP, "yes" or "no".
    # do-udp: yes

    # Enable TCP, "yes" or "no".
    # do-tcp: yes

    # upstream connections use TCP only (and no UDP), "yes" or "no"
    # useful for tunneling scenarios, default no.
    # tcp-upstream: no

    # Maximum segment size (MSS) of TCP socket on which the server
    # responds to queries. Default is 0, system default MSS.
    # tcp-mss: 0

    # Maximum segment size (MSS) of TCP socket for outgoing queries.
    # Default is 0, system default MSS.
    # outgoing-tcp-mss: 0

    # Use systemd socket activation for UDP, TCP, and control sockets.
    # use-systemd: no

    # Detach from the terminal, run in background, "yes" or "no".
    # Set the value to "no" when unbound runs as systemd service.
    # do-daemonize: yes

    # control which clients are allowed to make (recursive) queries
    # to this server. Specify classless netblocks with /size and action.
    # By default everything is refused, except for localhost.
    # Choose deny (drop message), refuse (polite error reply),
    # allow (recursive ok), allow_snoop (recursive and nonrecursive ok)
    # deny_non_local (drop queries unless can be answered from local-data)
    # refuse_non_local (like deny_non_local but polite error reply).
    # access-control: 0.0.0.0/0 refuse
    # access-control: 127.0.0.0/8 allow
    # access-control: ::0/0 refuse
    # access-control: ::1 allow
    # access-control: ::ffff:127.0.0.1 allow

    # tag access-control with list of tags (in "" with spaces between)
```

```
    # Clients using this access control element use localzones that
    # are tagged with one of these tags.
    # access-control-tag: 192.0.2.0/24 "tag2 tag3"

    # set action for particular tag for given access control element
    # if you have multiple tag values, the tag used to lookup the action
    # is the first tag match between access-control-tag and local-zone-tag
    # where "first" comes from the order of the define-tag values.
    # access-control-tag-action: 192.0.2.0/24 tag3 refuse

    # set redirect data for particular tag for access control element
    # access-control-tag-data: 192.0.2.0/24 tag2 "A 127.0.0.1"

    # Set view for access control element
    # access-control-view: 192.0.2.0/24 viewname

    # if given, a chroot(2) is done to the given directory.
    # i.e. you can chroot to the working directory, for example,
    # for extra security, but make sure all files are in that directory.
    #
    # If chroot is enabled, you should pass the configfile (from the
    # commandline) as a full path from the original root. After the
    # chroot has been performed the now defunct portion of the config
    # file path is removed to be able to reread the config after a reload.
    #
    # All other file paths (working dir, logfile, roothints, and
    # key files) can be specified in several ways:
    #     o as an absolute path relative to the new root.
    #     o as a relative path to the working directory.
    #     o as an absolute path relative to the original root.
    # In the last case the path is adjusted to remove the unused portion.
    #
    # The pid file can be absolute and outside of the chroot, it is
    # written just prior to performing the chroot and dropping permissions.
    #
    # Additionally, unbound may need to access /dev/random (for entropy).
    # How to do this is specific to your OS.
    #
    # If you give "" no chroot is performed. The path must not end in a /.
    # chroot: "/usr/local/etc/unbound"

    # if given, user privileges are dropped (after binding port),
    # and the given username is assumed. Default is user "unbound".
    # If you give "" no privileges are dropped.
    # username: "unbound"

    # the working directory. The relative files in this config are
    # relative to this directory. If you give "" the working directory
    # is not changed.
    # If you give a server: directory: dir before include: file statements
    # then those includes can be relative to the working directory.
```

```
    # directory: "/usr/local/etc/unbound"

    # the log file, "" means log to stderr.
    # Use of this option sets use-syslog to "no".
    # logfile: ""

    # Log to syslog(3) if yes. The log facility LOG_DAEMON is used to
    # log to. If yes, it overrides the logfile.
    # use-syslog: yes

    # Log identity to report. if empty, defaults to the name of argv[0]
    # (usually "unbound").
    # log-identity: ""

    # print UTC timestamp in ascii to logfile, default is epoch in seconds.
    # log-time-ascii: no

    # print one line with time, IP, name, type, class for every query.
    # log-queries: no

    # print one line per reply, with time, IP, name, type, class, rcode,
    # timetoresolve, fromcache and responsesize.
    # log-replies: no

    # the pid file. Can be an absolute path outside of chroot/work dir.
    # pidfile: "/usr/local/etc/unbound/unbound.pid"

    # file to read root hints from.
    # get one from https://www.internic.net/domain/named.cache
    # root-hints: ""

    # enable to not answer id.server and hostname.bind queries.
    # hide-identity: no

    # enable to not answer version.server and version.bind queries.
    # hide-version: no
    # enable to not answer trustanchor.unbound queries.
    # hide-trustanchor: no

    # the identity to report. Leave "" or default to return hostname.
    # identity: ""

    # the version to report. Leave "" or default to return package version.
    # version: ""

    # the target fetch policy.
    # series of integers describing the policy per dependency depth.
    # The number of values in the list determines the maximum dependency
    # depth the recursor will pursue before giving up. Each integer means:
    #     -1 : fetch all targets opportunistically,
    #     0: fetch on demand,
```

```
#    positive value: fetch that many targets opportunistically.
# Enclose the list of numbers between quotes ("").
# target-fetch-policy: "3 2 1 0 0"

# Harden against very small EDNS buffer sizes.
# harden-short-bufsize: no

# Harden against unseemly large queries.
# harden-large-queries: no

# Harden against out of zone rrsets, to avoid spoofing attempts.
# harden-glue: yes

# Harden against receiving dnssec-stripped data. If you turn it
# off, failing to validate dnskey data for a trustanchor will
# trigger insecure mode for that zone (like without a trustanchor).
# Default on, which insists on dnssec data for trust-anchored zones.
# harden-dnssec-stripped: yes

# Harden against queries that fall under dnssec-signed nxdomain names.
# harden-below-nxdomain: no

# Harden the referral path by performing additional queries for
# infrastructure data.  Validates the replies (if possible).
# Default off, because the lookups burden the server.  Experimental
# implementation of draft-wijngaards-dnsext-resolver-side-mitigation.
# harden-referral-path: no

# Harden against algorithm downgrade when multiple algorithms are
# advertised in the DS record.  If no, allows the weakest algorithm
# to validate the zone.
# harden-algo-downgrade: no

# Sent minimum amount of information to upstream servers to enhance
# privacy. Only sent minimum required labels of the QNAME and set QTYPE
# to NS when possible.
# qname-minimisation: no

# QNAME minimisation in strict mode. Do not fall-back to sending full
# QNAME to potentially broken nameservers. A lot of domains will not be
# resolvable when this option in enabled.
# This option only has effect when qname-minimisation is enabled.
# qname-minimisation-strict: no

# Use 0x20-encoded random bits in the query to foil spoof attempts.
# This feature is an experimental implementation of draft dns-0x20.
# use-caps-for-id: no

# Domains (and domains in them) without support for dns-0x20 and
# the fallback fails because they keep sending different answers.
# caps-whitelist: "licdn.com"
```

```
    # caps-whitelist: "senderbase.org"

    # Enforce privacy of these addresses. Strips them away from answers.
    # It may cause DNSSEC validation to additionally mark it as bogus.
    # Protects against 'DNS Rebinding' (uses browser as network proxy).
    # Only 'private-domain' and 'local-data' names are allowed to have
    # these private addresses. No default.
    # private-address: 10.0.0.0/8
    # private-address: 172.16.0.0/12
    # private-address: 192.168.0.0/16
    # private-address: 169.254.0.0/16
    # private-address: fd00::/8
    # private-address: fe80::/10
    # private-address: ::ffff:0:0/96

    # Allow the domain (and its subdomains) to contain private addresses.
    # local-data statements are allowed to contain private addresses too.
    # private-domain: "example.com"

    # If nonzero, unwanted replies are not only reported in statistics,
    # but also a running total is kept per thread. If it reaches the
    # threshold, a warning is printed and a defensive action is taken,
    # the cache is cleared to flush potential poison out of it.
    # A suggested value is 10000000, the default is 0 (turned off).
    # unwanted-reply-threshold: 0

    # Do not query the following addresses. No DNS queries are sent there.
    # List one address per entry. List classless netblocks with /size,
    # do-not-query-address: 127.0.0.1/8
    # do-not-query-address: ::1

    # if yes, the above default do-not-query-address entries are present.
    # if no, localhost can be queried (for testing and debugging).
    # do-not-query-localhost: yes

    # if yes, perform prefetching of almost expired message cache entries.
    # prefetch: no

    # if yes, perform key lookups adjacent to normal lookups.
    # prefetch-key: no

    # if yes, Unbound rotates RRSet order in response.
    # rrset-roundrobin: no

    # if yes, Unbound doesn't insert authority/additional sections
    # into response messages when those sections are not required.
    # minimal-responses: no

    # true to disable DNSSEC lameness check in iterator.
    # disable-dnssec-lame-check: no
```

```
    # module configuration of the server. A string with identifiers
    # separated by spaces. Syntax: "[dns64] [validator] iterator"
    # module-config: "validator iterator"

    # File with trusted keys, kept uptodate using RFC5011 probes,
    # initial file like trust-anchor-file, then it stores metadata.
    # Use several entries, one per domain name, to track multiple zones.
    #
    # If you want to perform DNSSEC validation, run unbound-anchor before
    # you start unbound (i.e. in the system boot scripts).  And enable:
    # Please note usage of unbound-anchor root anchor is at your own risk
    # and under the terms of our LICENSE (see that file in the source).
    # auto-trust-anchor-file: "/usr/local/etc/unbound/root.key"

    # trust anchor signaling sends a RFC8145 key tag query after priming.
    # trust-anchor-signaling: no

    # File with DLV trusted keys. Same format as trust-anchor-file.
    # There can be only one DLV configured, it is trusted from root down.
    # DLV is going to be decommissioned.  Please do not use it any more.
    # dlv-anchor-file: "dlv.isc.org.key"

    # File with trusted keys for validation. Specify more than one file
    # with several entries, one file per entry.
    # Zone file format, with DS and DNSKEY entries.
    # Note this gets out of date, use auto-trust-anchor-file please.
    # trust-anchor-file: ""

    # Trusted key for validation. DS or DNSKEY. specify the RR on a
    # single line, surrounded by "". TTL is ignored. class is IN default.
    # Note this gets out of date, use auto-trust-anchor-file please.
    # (These examples are from August 2007 and may not be valid anymore).
    # trust-anchor: "nlnetlabs.nl. DNSKEY 257 3 5
AQPzzTWMz8qSWIQlfRnPckx2BiVmkVN6LPupO3mbz7FhLSnm26n6iG9N
Lby97Ji453aWZY3M5/xJBSOS2vWtco2t8C0+xeO1bc/d6ZTy32DHchpW
6rDH1vp86Ll+ha0tmwyy9QP7y2bVw5zSbFCrefk8qCUBgfHm9bHzMG1U BYtEIQ=="
    # trust-anchor: "jelte.nlnetlabs.nl. DS 42860 5 1
14D739EB566D2B1A5E216A0BA4D17FA9B038BE4A"

    # File with trusted keys for validation. Specify more than one file
    # with several entries, one file per entry. Like trust-anchor-file
    # but has a different file format. Format is BIND-9 style format,
    # the trusted-keys { name flag proto algo "key"; }; clauses are read.
    # you need external update procedures to track changes in keys.
    # trusted-keys-file: ""

    # Ignore chain of trust. Domain is treated as insecure.
    # domain-insecure: "example.com"

    # Override the date for validation with a specific fixed date.
    # Do not set this unless you are debugging signature inception
```

```
    # and expiration. "" or "0" turns the feature off. -1 ignores date.
    # val-override-date: ""

    # The time to live for bogus data, rrsets and messages. This avoids
    # some of the revalidation, until the time interval expires. in secs.
    # val-bogus-ttl: 60

    # The signature inception and expiration dates are allowed to be off
    # by 10% of the signature lifetime (expir-incep) from our local clock.
    # This leeway is capped with a minimum and a maximum.  In seconds.
    # val-sig-skew-min: 3600
    # val-sig-skew-max: 86400

    # Should additional section of secure message also be kept clean of
    # unsecure data. Useful to shield the users of this validator from
    # potential bogus data in the additional section. All unsigned data
    # in the additional section is removed from secure messages.
    # val-clean-additional: yes

    # Turn permissive mode on to permit bogus messages. Thus, messages
    # for which security checks failed will be returned to clients,
    # instead of SERVFAIL. It still performs the security checks, which
    # result in interesting log files and possibly the AD bit in
    # replies if the message is found secure. The default is off.
    # val-permissive-mode: no

    # Ignore the CD flag in incoming queries and refuse them bogus data.
    # Enable it if the only clients of unbound are legacy servers (w2008)
    # that set CD but cannot validate themselves.
    # ignore-cd-flag: no

    # Serve expired reponses from cache, with TTL 0 in the response,
    # and then attempt to fetch the data afresh.
    # serve-expired: no

    # Have the validator log failed validations for your diagnosis.
    # 0: off. 1: A line per failed user query. 2: With reason and bad IP.
    # val-log-level: 0

    # It is possible to configure NSEC3 maximum iteration counts per
    # keysize. Keep this table very short, as linear search is done.
    # A message with an NSEC3 with larger count is marked insecure.
    # List in ascending order the keysize and count values.
    # val-nsec3-keysize-iterations: "1024 150 2048 500 4096 2500"

    # instruct the auto-trust-anchor-file probing to add anchors after ttl.
    # add-holddown: 2592000 # 30 days

    # instruct the auto-trust-anchor-file probing to del anchors after ttl.
    # del-holddown: 2592000 # 30 days
```

```
    # auto-trust-anchor-file probing removes missing anchors after ttl.
    # If the value 0 is given, missing anchors are not removed.
    # keep-missing: 31622400 # 366 days

    # debug option that allows very small holddown times for key rollover,
    # otherwise the RFC mandates probe intervals must be at least 1 hour.
    # permit-small-holddown: no

    # the amount of memory to use for the key cache.
    # plain value in bytes or you can append k, m or G. default is "4Mb".
    # key-cache-size: 4m

    # the number of slabs to use for the key cache.
    # the number of slabs must be a power of 2.
    # more slabs reduce lock contention, but fragment memory usage.
    # key-cache-slabs: 4

    # the amount of memory to use for the negative cache (used for DLV).
    # plain value in bytes or you can append k, m or G. default is "1Mb".
    # neg-cache-size: 1m

    # By default, for a number of zones a small default 'nothing here'
    # reply is built-in.  Query traffic is thus blocked.  If you
    # wish to serve such zone you can unblock them by uncommenting one
    # of the nodefault statements below.
    # You may also have to use domain-insecure: zone to make DNSSEC work,
    # unless you have your own trust anchors for this zone.
    # local-zone: "localhost." nodefault
    # local-zone: "127.in-addr.arpa." nodefault
    # local-zone:
"1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa."
nodefault
    # local-zone: "onion." nodefault
    # local-zone: "test." nodefault
    # local-zone: "invalid." nodefault
    # local-zone: "10.in-addr.arpa." nodefault
    # local-zone: "16.172.in-addr.arpa." nodefault
    # local-zone: "17.172.in-addr.arpa." nodefault
    # local-zone: "18.172.in-addr.arpa." nodefault
    # local-zone: "19.172.in-addr.arpa." nodefault
    # local-zone: "20.172.in-addr.arpa." nodefault
    # local-zone: "21.172.in-addr.arpa." nodefault
    # local-zone: "22.172.in-addr.arpa." nodefault
    # local-zone: "23.172.in-addr.arpa." nodefault
    # local-zone: "24.172.in-addr.arpa." nodefault
    # local-zone: "25.172.in-addr.arpa." nodefault
    # local-zone: "26.172.in-addr.arpa." nodefault
    # local-zone: "27.172.in-addr.arpa." nodefault
    # local-zone: "28.172.in-addr.arpa." nodefault
    # local-zone: "29.172.in-addr.arpa." nodefault
    # local-zone: "30.172.in-addr.arpa." nodefault
```

```
    # local-zone: "31.172.in-addr.arpa." nodefault
    # local-zone: "168.192.in-addr.arpa." nodefault
    # local-zone: "0.in-addr.arpa." nodefault
    # local-zone: "254.169.in-addr.arpa." nodefault
    # local-zone: "2.0.192.in-addr.arpa." nodefault
    # local-zone: "100.51.198.in-addr.arpa." nodefault
    # local-zone: "113.0.203.in-addr.arpa." nodefault
    # local-zone: "255.255.255.255.in-addr.arpa." nodefault
    # local-zone:
"0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa."
nodefault
    # local-zone: "d.f.ip6.arpa." nodefault
    # local-zone: "8.e.f.ip6.arpa." nodefault
    # local-zone: "9.e.f.ip6.arpa." nodefault
    # local-zone: "a.e.f.ip6.arpa." nodefault
    # local-zone: "b.e.f.ip6.arpa." nodefault
    # local-zone: "8.b.d.0.1.0.0.2.ip6.arpa." nodefault
    # And for 64.100.in-addr.arpa. to 127.100.in-addr.arpa.

    # If unbound is running service for the local host then it is useful
    # to perform lan-wide lookups to the upstream, and unblock the
    # long list of local-zones above.  If this unbound is a dns server
    # for a network of computers, disabled is better and stops information
    # leakage of local lan information.
    # unblock-lan-zones: no

    # The insecure-lan-zones option disables validation for
    # these zones, as if they were all listed as domain-insecure.
    # insecure-lan-zones: no

    # a number of locally served zones can be configured.
    #     local-zone: <zone> <type>
    #     local-data: "<resource record string>"
    # o deny serves local data (if any), else, drops queries.
    # o refuse serves local data (if any), else, replies with error.
    # o static serves local data, else, nxdomain or nodata answer.
    # o transparent gives local data, but resolves normally for other names
    # o redirect serves the zone data for any subdomain in the zone.
    # o nodefault can be used to normally resolve AS112 zones.
    # o typetransparent resolves normally for other types and other names
    # o inform acts like transparent, but logs client IP address
    # o inform_deny drops queries and logs client IP address
    # o always_transparent, always_refuse, always_nxdomain, resolve in
    #   that way but ignore local data for that name.
    #
    # defaults are localhost address, reverse for 127.0.0.1 and ::1
    # and nxdomain for AS112 zones. If you configure one of these zones
    # the default content is omitted, or you can omit it with 'nodefault'.
    #
    # If you configure local-data without specifying local-zone, by
    # default a transparent local-zone is created for the data.
```

```
#
# You can add locally served data with
# local-zone: "local." static
# local-data: "mycomputer.local. IN A 192.0.2.51"
# local-data: 'mytext.local TXT "content of text record"'
#
# You can override certain queries with
# local-data: "adserver.example.com A 127.0.0.1"
#
# You can redirect a domain to a fixed address with
# (this makes example.com, www.example.com, etc, all go to 192.0.2.3)
# local-zone: "example.com" redirect
# local-data: "example.com A 192.0.2.3"
#
# Shorthand to make PTR records, "IPv4 name" or "IPv6 name".
# You can also add PTR records using local-data directly, but then
# you need to do the reverse notation yourself.
# local-data-ptr: "192.0.2.3 www.example.com"

# tag a localzone with a list of tag names (in "" with spaces between)
# local-zone-tag: "example.com" "tag2 tag3"

# add a netblock specific override to a localzone, with zone type
# local-zone-override: "example.com" 192.0.2.0/24 refuse

# service clients over SSL (on the TCP sockets), with plain DNS inside
# the SSL stream.  Give the certificate to use and private key.
# default is "" (disabled).  requires restart to take effect.
# ssl-service-key: "path/to/privatekeyfile.key"
# ssl-service-pem: "path/to/publiccertfile.pem"
# ssl-port: 853

# request upstream over SSL (with plain DNS inside the SSL stream).
# Default is no.  Can be turned on and off with unbound-control.
# ssl-upstream: no

# DNS64 prefix. Must be specified when DNS64 is use.
# Enable dns64 in module-config.  Used to synthesize IPv6 from IPv4.
# dns64-prefix: 64:ff9b::0/96

# ratelimit for uncached, new queries, this limits recursion effort.
# ratelimiting is experimental, and may help against randomqueryflood.
# if 0(default) it is disabled, otherwise state qps allowed per zone.
# ratelimit: 0

# ratelimits are tracked in a cache, size in bytes of cache (or k,m).
# ratelimit-size: 4m
# ratelimit cache slabs, reduces lock contention if equal to cpucount.
# ratelimit-slabs: 4

# 0 blocks when ratelimited, otherwise let 1/xth traffic through
```

```
    # ratelimit-factor: 10

    # override the ratelimit for a specific domain name.
    # give this setting multiple times to have multiple overrides.
    # ratelimit-for-domain: example.com 1000
    # override the ratelimits for all domains below a domain name
    # can give this multiple times, the name closest to the zone is used.
    # ratelimit-below-domain: com 1000

    # global query ratelimit for all ip addresses.
    # feature is experimental.
    # if 0(default) it is disabled, otherwise states qps allowed per ip
address
    # ip-ratelimit: 0

    # ip ratelimits are tracked in a cache, size in bytes of cache (or k,m).
    # ip-ratelimit-size: 4m
    # ip ratelimit cache slabs, reduces lock contention if equal to
cpucount.
    # ip-ratelimit-slabs: 4

    # 0 blocks when ip is ratelimited, otherwise let 1/xth traffic through
    # ip-ratelimit-factor: 10

    # Specific options for ipsecmod. unbound needs to be configured with
    # --enable-ipsecmod for these to take effect.
    #
    # Enable or disable ipsecmod (it still needs to be defined in
    # module-config above). Can be used when ipsecmod needs to be
    # enabled/disabled via remote-control(below).
    # ipsecmod-enabled: yes
    #
    # Path to executable external hook. It must be defined when ipsecmod is
    # listed in module-config (above).
    # ipsecmod-hook: "./my_executable"
    #
    # When enabled unbound will reply with SERVFAIL if the return value of
    # the ipsecmod-hook is not 0.
    # ipsecmod-strict: no
    #
    # Maximum time to live (TTL) for cached A/AAAA records with IPSECKEY.
    # ipsecmod-max-ttl: 3600
    #
    # Reply with A/AAAA even if the relevant IPSECKEY is bogus. Mainly used
for
    # testing.
    # ipsecmod-ignore-bogus: no
    #
    # Domains for which ipsecmod will be triggered. If not defined (default)
    # all domains are treated as being whitelisted.
    # ipsecmod-whitelist: "example.com"
```

```
    # ipsecmod-whitelist: "nlnetlabs.nl"


# Python config section. To enable:
# o use --with-pythonmodule to configure before compiling.
# o list python in the module-config string (above) to enable.
# o and give a python-script to run.
python:
    # Script file to load
    # python-script: "/usr/local/etc/unbound/ubmodule-tst.py"

# Remote control config section.
remote-control:
    # Enable remote control with unbound-control(8) here.
    # set up the keys and certificates with unbound-control-setup.
    # control-enable: no

    # Set to no and use an absolute path as control-interface to use
    # a unix local named pipe for unbound-control.
    # control-use-cert: yes

    # what interfaces are listened to for remote control.
    # give 0.0.0.0 and ::0 to listen to all interfaces.
    # control-interface: 127.0.0.1
    # control-interface: ::1

    # port number for remote control operations.
    # control-port: 8953

    # unbound server key file.
    # server-key-file: "/usr/local/etc/unbound/unbound_server.key"

    # unbound server certificate file.
    # server-cert-file: "/usr/local/etc/unbound/unbound_server.pem"

    # unbound-control key file.
    # control-key-file: "/usr/local/etc/unbound/unbound_control.key"

    # unbound-control certificate file.
    # control-cert-file: "/usr/local/etc/unbound/unbound_control.pem"

# Stub zones.
# Create entries like below, to make all queries for 'example.com' and
# 'example.org' go to the given list of nameservers. list zero or more
# nameservers by hostname or by ipaddress. If you set stub-prime to yes,
# the list is treated as priming hints (default is no).
# With stub-first yes, it attempts without the stub if it fails.
# Consider adding domain-insecure: name and local-zone: name nodefault
# to the server: section if the stub is a locally served zone.
# stub-zone:
#   name: "example.com"
```

```
#   stub-addr: 192.0.2.68
#   stub-prime: no
#   stub-first: no
#   stub-ssl-upstream: no
# stub-zone:
#   name: "example.org"
#   stub-host: ns.example.com.

# Forward zones
# Create entries like below, to make all queries for 'example.com' and
# 'example.org' go to the given list of servers. These servers have to
handle
# recursion to other nameservers. List zero or more nameservers by hostname
# or by ipaddress. Use an entry with name "." to forward all queries.
# If you enable forward-first, it attempts without the forward if it fails.
# forward-zone:
#   name: "example.com"
#   forward-addr: 192.0.2.68
#   forward-addr: 192.0.2.73@5355  # forward to port 5355.
#   forward-first: no
#   forward-ssl-upstream: no
# forward-zone:
#   name: "example.org"
#   forward-host: fwd.example.com

# Views
# Create named views. Name must be unique. Map views to requests using
# the access-control-view option. Views can contain zero or more local-zone
# and local-data options. Options from matching views will override global
# options. Global options will be used if no matching view is found.
# With view-first yes, it will try to answer using the global local-zone and
# local-data elements if there is no view specific match.
# view:
#   name: "viewname"
#   local-zone: "example.com" redirect
#   local-data: "example.com A 192.0.2.3"
#   local-data-ptr: "192.0.2.3 www.example.com"
#   view-first: no
# view:
#   name: "anotherview"
#   local-zone: "example.com" refuse

# DNSCrypt
# Caveats:
# 1. the keys/certs cannot be produced by unbound. You can use dnscrypt-
wrapper
#   for this:
https://github.com/cofyc/dnscrypt-wrapper/blob/master/README.md#usage
# 2. dnscrypt channel attaches to an interface. you MUST set interfaces to
#   listen on `dnscrypt-port` with the follo0wing snippet:
# server:
```

```
#     interface: 0.0.0.0@443
#     interface: ::0@443
#
# Finally, `dnscrypt` config has its own section.
# dnscrypt:
#     dnscrypt-enable: yes
#     dnscrypt-port: 443
#     dnscrypt-provider: 2.dnscrypt-cert.example.com.
#     dnscrypt-secret-key: /path/unbound-conf/keys1/1.key
#     dnscrypt-secret-key: /path/unbound-conf/keys2/1.key
#     dnscrypt-provider-cert: /path/unbound-conf/keys1/1.cert
#     dnscrypt-provider-cert: /path/unbound-conf/keys2/1.cert

# CacheDB
# Enable external backend DB as auxiliary cache.  Specify the backend name
# (default is "testframe", which has no use other than for debugging and
# testing) and backend-specific options.  The 'cachedb' module must be
# included in module-config.
# cachedb:
#     backend: "testframe"
#     # secret seed string to calculate hashed keys
#     secret-seed: "default"
```

Question 5. Why do the programs return a result value?

When I tried to use unbound-checkconf for the unbound.conf file I get this error

fatal error: user 'unbound' does not exist so I used this command to add user.

```
kotaiba@bristol:/usr/local/etc/unbound$ sudo useradd unbound -s
/sbin/nologin
```

So now I can check:

```
kotaiba@bristol:/usr/local/etc/unbound$ unbound-checkconf
unbound-checkconf: no errors in /usr/local/etc/unbound/unbound.conf
kotaiba@bristol:/usr/local/etc/unbound$ echo $?
0
```

It gave me no errors and return value of 0. if we chekc on unbound website it states the following “The unbound-checkconf program exits with status code 1 on error, 0 for a correct config file.”

AFTER FEEDBACK “Why does this happen?”:

Because, Based on Unbound-checkconf documentation of , the program will either return 0 or 1 if there are no errors, or any errors respectively. So the program give a result value because this value can be used by other programs/scripts depend on its result. In other case the value can represent the errors. To be more specific it is used to write code/script that use the error codes.

Source:

1- http://lost-and-found-narihiro.blogspot.nl/2011/01/how-to-compile-unbound-147-on-centos55.html

2- https://www.unbound.net/documentation/unbound-checkconf.html

Question 6. Show the changes you made to your configuration to allow remote control.

If we want to setup the remote control:

First, we run unbound-control-setup to generate the necessary TLS key file:

```
kotaiba@bristol:~$ unbound-control-setup
setup in directory /usr/local/etc/unbound
generating unbound_server.key
unbound_server.key: Permission denied
140262812071576:error:0200100D:system library:fopen:Permission
denied:bss_file.c:398:fopen('unbound_server.key','w')
140262812071576:error:20074002:BIO routines:FILE_CTRL:system
lib:bss_file.c:400:
/usr/local/sbin/unbound-control-setup fatal error: could not genrsa
kotaiba@bristol:~$ sudo unbound-control-setup
[sudo] password for kotaiba:
setup in directory /usr/local/etc/unbound
generating unbound_server.key
Generating RSA private key, 3072 bit long modulus
.......................++
.....................................................................................
.............................++
e is 65537 (0x10001)
generating unbound_control.key
Generating RSA private key, 3072 bit long modulus
.....................................................++
.....................................++
e is 65537 (0x10001)
create unbound_server.pem (self signed certificate)
create unbound_control.pem (signed client certificate)
Signature ok
subject=/CN=unbound-control
Getting CA Private Key
Setup success. Certificates created. Enable in unbound.conf file to use
```

Then, since we already specified a username of unbound to run a daemon we use this to generate the keys, so that the server is allowed to read the keys.

```
setup in directory /usr/local/etc/unbound
unbound_server.key exists
unbound_control.key exists
```

Last, add the following at the end of the config file.

```
# enable remote-control
remote-control: control-enable: yes
```

Sources:

1- https://unbound.net/documentation/howto_setup.html

Question 7. Show that remote control works.

```
kotaiba@bristol:/usr/local/etc/unbound$ sudo unbound-control status

version: 1.6.6
verbosity: 1
threads: 1
modules: 2 [ validator iterator ]
uptime: 6 seconds
options: control(ssl)
unbound (pid 9542) is running...
```

Question 8. What other commands/functions does rndc/unbound-control provide? Make your own list that describes the most important ones.

1- start: start the server.

2- stop: stop the server. The server daemon exits.

3- reload: Reload the server. This flushes the cache and reads the config file fresh.

4- status: display server status. Exit code 3 if not running

5- dump_cache: The contents of the cache is printed in a text format to stdout.

6- local_zone: Add new local zone with name and type.

7- local_zone_remove : Remove the local zone with the given name.

8- forward: Setup forwarding mode. Configures if the server should ask other upstream nameservers, should go to the internet root name-servers itself, or show the current config.

9- lookup: Print to stdout the name servers that would be used to look up the name specified.

10- set_option: Set the option to the given value without a reload. The cache is therefore not flushed.

Source:

1- https://unbound.net/documentation/unbound-control.html

Question 9. What do you need to put in resolv.conf (and/or other files) to permanently use your own name server?

First, lets check our resolv.conf file:

```
kotaiba@bristol:/etc/resolvconf$ cat /etc/resolv.conf

# Dynamic resolv.conf(5) file for glibc resolver(3) generated by
resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
```

```
nameserver 145.100.96.11
nameserver 145.100.96.22
search studlab.os3.nl
```

Second, I have to add these to the /etc/network/interfaces

```
# The primary network interface
dns-nameservers 127.0.0.1


#IPV6
dns-nameservers ::1
```

After that, it should look like this:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eno1
iface eno1 inet dhcp
dns-nameservers 127.0.0.1

#IPV6
iface eno1 inet6 static
address 2001:610:158:1046:145:100:104:163
netmask 64
gateway 2001:610:158:1046::1
dns-nameservers ::1
```

Then, restart the network:

```
kotaiba@bristol:/etc/resolvconf$ sudo service networking restart
```

Now, we check inside the resolv.conf file:

```
kotaiba@bristol:/etc/resolvconf$ cat /etc/resolv.conf

# Dynamic resolv.conf(5) file for glibc resolver(3) generated by
resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 145.100.96.11
nameserver 145.100.96.22
nameserver 127.0.0.1
```

```
search studlab.os3.nl
```

Note: to answer this question I just wanted to add it instead of overwrite the file it self.

AFTER THE FEEDBACK " You need to be above the system nameservers in order to resolve with your server. Please do this." IT is really easy to change the order of the nameserver and put "nameserver 127.0.0.1" as the first one.

so update it:

```
kotaiba@bristol:/etc/resolvconf$ cat /etc/resolv.conf

# Dynamic resolv.conf(5) file for glibc resolver(3) generated by
resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 127.0.0.1
nameserver 145.100.96.11
nameserver 145.100.96.22
search studlab.os3.nl
```

After the we need to update it:

```
kotaiba@bristol:~$ sudo resolvconf -u
```

Now, to check it:

```
kotaiba@bristol:~$ dig google.com

; <<>> DiG 9.10.6 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37935
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.com.            IN    A

;; ANSWER SECTION:
google.com.     300    IN    A    172.217.17.78

;; AUTHORITY SECTION:
google.com.     155711    IN    NS    ns2.google.com.
google.com.     155711    IN    NS    ns1.google.com.
google.com.     155711    IN    NS    ns3.google.com.
google.com.     155711    IN    NS    ns4.google.com.

;; ADDITIONAL SECTION:
ns1.google.com.     155711    IN    A    216.239.32.10
ns2.google.com.     155711    IN    A    216.239.34.10
```

```
ns3.google.com.     155711    IN    A    216.239.36.10
ns4.google.com.     155711    IN    A    216.239.38.10

;; Query time: 5 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Oct 26 22:49:23 CEST 2017
;; MSG SIZE  rcvd: 191
```

Question 10. Show the forward mapping zone file in your log.

1- nsd.conf file:

```
kotaiba@bristol:/usr/local/etc/nsd$ cat nsd.conf

server:
    ip-address: 127.0.0.1
    logfile: "log/unbound.log"
    root-hints: "/usr/local/etc/unbound/named.root"
    hide-identity: yes
    hide-version: yes
    do-not-query-localhost: no

remote-control:
    control-enable: yes
    server-key-file: "/usr/local/etc/unbound/unbound_server.key"
    server-cert-file: "/usr/local/etc/unbound/unbound_server.pem"
    control-key-file: "/usr/local/etc/unbound/unbound_control.key"
    control-cert-file: "/usr/local/etc/unbound/unbound_control.pem"

forward-zone:
    name: "bristol.prac.os3.nl"
    forward-addr: 145.100.96.11#53

    do-ip4: yes
    do-ip6: yes
    port: 53
    verbosity: 4
    username: nsd
    database: ""
    logfile: "/usr/local/etc/nsd/nsd.log"
    pidfile: "/var/run/nsd.pid"
    remote-control:
    control-enable: yes
    control-interface: ::1
    control-interface: 127.0.0.1

zone:
    name: "bristol.prac.os3.nl"
    zonefile: "bristol.prac.os3.nl.zone"
```

bristol.prac.os3.nl.zone file:

```
kotaiba@bristol:/usr/local/etc/nsd$ cat bristol.prac.os3.nl.zone
$ORIGIN bristol.prac.os3.nl.
$TTL 1800

@       IN      SOA     ns1.bristol.prac.os3.nl.
admin.bristol.prac.os3.nl. (
                        2017092401          ; serial number
                        3600                    ; refresh
                        900                     ; retry
                        1209600                 ; expire
                        1800                    ; ttl
                        )

    IN    NS    ns1.bristol.prac.os3.nl.
    IN    NS    ns2.bristol.prac.os3.nl.

ns1 IN    A    145.100.96.66
ns1 IN    AAAA    2001:610:158:960::66

ns2 IN    A    145.100.96.99
ns2 IN    AAAA    2001:610:158:960::99

@   IN    A    145.100.104.38
@   IN    AAAA    2001:610:158:1043:145:100:104:38

mail    IN    A    145.100.96.66
mail    IN    AAAA    2001:610:158:960::66

relay   IN    A    145.100.96.66
relay   IN    AAAA    2001:610:158:960::66

www IN    CNAME    bristol.prac.os3.nl.

@   IN    MX    10 mail.bristol.prac.os3.nl.
@   IN    MX    20 relay.bristol.prac.os3.nl.
```

Question 11. If Niels had not yet implemented the delegation, what information would you need to give him so that he can implement it?

1- A name server (NS) resource record to effect the delegation. 2- A host (A or AAAA) resource record is necessary to resolve the name of the server that is specified in the NS resource record to its IP address.

In this case:

```
kotaiba@bristol:~$ nslookup -type=mx os3.nl
Server:     145.100.96.11
Address:    145.100.96.11#53
```

```
Non-authoritative answer:
os3.nl  mail exchanger = 42 smtp.os3.nl.

Authoritative answers can be found from:
os3.nl  nameserver = ns2.os3.nl.
os3.nl  nameserver = ns1.os3.nl.
os3.nl  nameserver = ns1.zurich.surf.net.
smtp.os3.nl internet address = 145.100.96.119
smtp.os3.nl has AAAA address 2001:610:158:960::119
ns1.os3.nl  internet address = 145.100.96.66
ns1.os3.nl  has AAAA address 2001:610:158:960::66
ns1.zurich.surf.net internet address = 195.176.255.9
ns1.zurich.surf.net has AAAA address 2001:620:0:9::1103
ns2.os3.nl  internet address = 145.100.96.99
ns2.os3.nl  has AAAA address 2001:610:158:960::99
```

Source:

1- http://support.simpledns.com/kb/a196/how-to-delegate-a-sub-domain-to-other-dns-servers.aspx

2- https://technet.microsoft.com/en-us/library/cc771640

Question 12. What important requirement is not yet met for your subdomain?

1- There must be at least two NS records listed in a delegation, and the hosts must not resolve to the same IP address.

2- The name servers must answer DNS queries over both the UDP and TCP protocols on port 53. Tests will be conducted from multiple network locations to verify the name server is responding.

3- For name servers that have IP addresses listed as glue, the IP addresses must match the authoritative A and AAAA records for that host.

4- At the time of the listing request, there must be a DNSKEY that matches the DS record present in the child zone.

Source:

1- https://www.iana.org/help/nameserver-requirements

Oct 6 14:16:16 bristol nsd[11246]: xfrd: zone bristol.reims.prac.os3.nl received error code SERVER NOT AUTHORITATIVE FOR ZONE from 145.100.104.122