Classical internet applications
Lab session: DNS-1

Anatoly Tykushin Security and Network Engineering, a.tykushin@innopolis.ru

August 29, 2016

Contents

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

A hash value processed on the downloaded file is a way to make sure that the content is transferred well and has not been damaged during the download process (authenticity and integrity).

2 Which kind of signature is the best one to use? Why?

One quick and easy way to verify the integrity of a downloaded file is to use various checksum tools (e.g., md5sum, sha256sum, cksum) to compute and compare checksums (e.g., MD5, SHA or CRC). However, checksums are vulnerable to collision attacks, and also cannot be used to verify the authenticity (i.e., owner) of a file.

Second way is to use asymmetric cryptography. By encrypting a message's hash value with the sender's private key generates a digital signature of a message. Now anyone can verify the signature by decrypting it using sender's public key. If this reveals the correct hash value for the message, so it is the only possible sender since he is the only one to have the private key that fits public key. Now anyone can find out about the encrypted message's hash value.

To sum it up, using combination of asymmetric cryptography with hash algorithms is the best way to support message authentication and integrity checking. One of the best example is PGP (pretty good privacy).

3 Why are caching-only name servers still useful?

DNS cache servers fundamentally provide options described in table 1

Option	Value
Avaliabiity	Should only respond to queries that originate from a local user base
Types of query that it should answer	recursive queries
Design of a software implementation	Must maintain state because of recursive queries and an internal cache
Records that it should attempt to resolve	Should attempt to resolve any request

Table 1: DNS Cache servers fundemental providing abilities

Also DNS cache servers (e.g. Unbound) are:

- lightweight
- high performance
- adding of software diversity
- single-purpose server
- securable
- manageable and portable

4 Unbound configuration

Created configuration file of caching-only unbound dns-server is shown in the listing below.

```
server:
verbosity: 2 # sets verbosity level to show detailed information in logs
port: 53
interface: 188.130.155.38 # interface to listen

do-ip4: yes
do-ip6: no
do-udp: yes
do-tcp: yes
access-control: 188.130.155.0/24 allow
access-control: 127.0.0.0/8 allow
#access-control: 0.0.0.0/0 refuse
username: "unbound"
directory: "/usr/local/etc/unbound"

logfile: "dns.log"
use-syslog: no
hide-version: yes # do not to show information of unbound's version

remote-control:
control-enable: yes

forward-zone: #forward all queries to google
name: "."
  forward-addr: 8.8.8.8 #google.com

forward-zone: # forward queries to the zone "st5.os3.su" to the ip4 + port to the authoritative server
name: "st5.os3.su"
  forward-addr: 188.130.155.38@5355
```

5 Testing configuration

Why do the *unbound-checkconf* program returns a value? This program returns a value to use it in script writing. Because writing log to the screen of the terminal isn't enough for script, but may be useful for users.

6 Changes you made to configuration to allow remote control

To use remote control as a tool to send commands to the server during runtime. To enable this commands we have to do the following steps:

Step 1: ***unbound-control-setup*** to generate the necessary TLS key files (they are put in the default install directory).

Step 2: add this option to configuration file
remote-control:
control-enable: yes

7 What other commands/functions does unbound-control provide?

Unbound-control commands provide runtime editing unbound.conf file. So it's more likely to list commands provided by **unbound-control** it the table. Table 2 contains most frequently used commands.

Option	Description
start	Start the server. Simply execs unbound
stop	Stop the server. The server daemon exits.
reload	Reload the server. This flushes the cache and reads the config file fresh
verbosity	Change verbosity value for logging
local zone *name type*	Add new local zone with name and type
load cache	The contents of the cache is loaded from stdin
dump cache	The contents of the cache is printed in a text format to stdout
forward add [+i] *zone addr*	Add a new forward zone to running unbound
forward remove [+i] *zone*	Remove a forward zone from running unbound

Table 2: Unbound-control commands

8 What do you need to put in resolv.conf (and/orotherfiles) to use your own name server?

As already mentioned file ***resolv.conf*** is automatically generated by resolv-conf utility. To add a record to the ***resolv.conf*** file you have to write **nameserver** ***[IP address of NSD]*** in a ***/etc/resolvconf/resolv.conf.d/base*** file.

9 Forward mapping zone

Log of the forward zone is shown in the list below:

```
================================ NSD zone file ================================
===============================================================================


$ORIGIN st5.os3.su.
$TTL 1800
@       IN      SOA     st5.os3.su.      godofzone.st5.os3.su. (
                        2014070312        ; serial number (was 201407201)
                        3600                    ; refresh
                        900                     ; retry
                        1209600                 ; expire
                        1800                    ; ttl
                        )
; Name servers
```

```
                    IN      NS      sne1.st5.os3.su.
                    IN      NS      sne2.st5.os3.su.
; Mail exchange
                    MX 10 mail1
                    MX 20  thegod
; A records for name servers
sne2                 IN      A       188.130.155.38
sne1                 IN      A       188.130.155.36
sne3  IN   A    188.130.155.46
@                    IN      A       188.130.155.38

; Canonical names
numbaone  CNAME  sne1
godmail  CNAME  thegod.st5.os3.su.

; Additional A records
```

10 Delegation

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

I have to tell him this data:

domain: st5.os3.su

ip: 188.130.155.38

nameserver: ns1.os3.su ; the next name server points to ns1 in the os3.su zone above

11 Important requirements

What important requirement is not yet met for your subdomain?

In the lists below you can see what important requirements do we have for our subdomain. You need the following records in your hosts DNS:

- two Name Server (NS) records pointing to the authoritative name servers for your sub-domain
- Address (A) records for the sub-domain name servers

And you need to provide a pair of DNS name servers for your sub-domain. They need to serve the following records:

- a Start of Authority (SOA) record for the sub-domain
- two or more (NS) records
- (A) records for the sub-domain name servers

Most of these fields are pertinent only for name server maintenance operations. However, MINIMUM is used in all query operations that retrieve RRs from a zone. Whenever a RR is sent in a response to a query, the TTL field is set to the maximum of the TTL field from the RR and the MINIMUM field in the appropriate SOA. Thus MINIMUM is a lower bound on the TTL field for all RRs in a zone. Note that this use of MINIMUM should occur when the RRs are copied into the response and not when the zone is loaded from a master file or via a zone transfer. The reason for this provison is to allow future dynamic update facilities to change the SOA RR with known semantics.