Public Service Announcement

• An announcement has been posted on Blackboard w.r.t. last week’s homework assignment

• The intention is for you to use PGP and/or S/MIME to secure (read: sign) e-mail messages

• You have until Wednesday 23:59:59 to complete the assignment
About SURFnet

• National Research and Education Network (NREN)

• Founded in 1986, incorporated 1988
  • 2013: 25 year anniversary!

• > 11000km dark-fibre network

• Shared ICT innovation centre

• > 160 connected institutions ± 1 million end users
A little bit about me

• M.Sc. in Computer Science from the University of Twente

• Worked in security for past 13 years (mostly crypto stuff)

• Ph.D. student (part-time) at DACS, UT

• Love scuba diving ;-)
Overview

• Part 1 (10:45h - 11:30h)
  – DNS: a brief history
  – Problems with classic DNS
  – DNSSEC: a theoretical introduction

• 11:30h: coffee break

• Part 2 (11:45h - 12:30h)
  – DNSSEC: deployment in practice; SURFnet’s setup
  – We don’t need no education
    A demonstration how DNS(SEC) can be abused
Background reading

You can download these from Blackboard
The Domain Name System was introduced in 1983 — the year I got my first computer.

The Internet was growing; using a "hosts" file no longer worked for the people running the net.

But the net was still small, people knew each other and security was not a design criterium.
Fast forward to the 90s and 00s

• The 1990s, those were the days :-)  
  – House music, “Purple” liberal cabinet in office, suave hair styles, my first web page and much more...  
  – But more importantly: first DNS cache poisoning attack by a guy called Eugene Kashpureff (1997)  
  – Standardisation process of DNSSEC starts in the IETF

• And onward into the new millenium!  
  – 2008: oops, a guy called Dan Kaminsky demonstrates that DNS cache poisoning is much, much worse then people thought  
  – 2010: DNSSEC takes off
### A deeper look at the DNS protocol

<table>
<thead>
<tr>
<th>IP</th>
<th>headers &amp; stuff</th>
</tr>
</thead>
<tbody>
<tr>
<td>src IP = 192.87.106.101 (ns1.surfnet.nl)</td>
<td></td>
</tr>
<tr>
<td>dst IP = 208.77.188.166 (<a href="http://www.example.com">www.example.com</a>)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UDP</th>
<th>src port = 53</th>
<th>dst port = 54321</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>headers &amp; stuff</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DNS</th>
<th>query ID = 1201</th>
<th>some flags</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#questions = 1</td>
<td>#answers = 1</td>
</tr>
<tr>
<td></td>
<td>#authority = 3</td>
<td>#additional = 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q?</th>
<th>AAAA record for <a href="http://www.surfnet.nl">www.surfnet.nl</a></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Aut:</th>
<th>surfnet.nl = ns1.surfnet.nl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aut:</td>
<td>surfnet.nl = ns2.surfnet.nl</td>
</tr>
<tr>
<td>Aut:</td>
<td>surfnet.nl = ns3.surfnet.nl</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Add.</td>
<td>ns2.surfnet.nl = 192.87.36.36 2001:610:3:200a:192:87:36:2</td>
</tr>
</tbody>
</table>
Kaminsky in action

**Attacker**

- `nonxist-abc.surfnet.nl A?` with `QID = x, src port = y`
- `nonxist-abc.surfnet.nl A?` with `QID = x - 2, port = y`

**Caching Resolver**

- `nonxist-abc.surfnet.nl A?` with `QID = x, src port = y`
- `QID = x - 1, port = y`
- `QID = x, port = y`

**Authoritative Name Server**

- `NXDOMAIN` with `QID = x, dst port = y`

**Match**

- This means the answer is accepted!
The “Kaminsky” packet

<table>
<thead>
<tr>
<th>IP</th>
<th>headers &amp; stuff</th>
</tr>
</thead>
<tbody>
<tr>
<td>src IP = 192.87.106.101 (ns1.surfnet.nl)</td>
<td></td>
</tr>
<tr>
<td>dst IP = spoofed; caching resolver’s IP</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UDP</th>
<th>headers &amp; stuff</th>
</tr>
</thead>
<tbody>
<tr>
<td>src port = 53</td>
<td>dst port = 54321</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DNS</th>
<th>headers &amp; stuff</th>
</tr>
</thead>
<tbody>
<tr>
<td>query ID = 1201</td>
<td>some flags</td>
</tr>
<tr>
<td>#questions = 1</td>
<td>#answers = 1</td>
</tr>
<tr>
<td>#authority = 3</td>
<td>#additional = 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q?</th>
<th>AAAA record for <a href="http://www.surfnet.nl">www.surfnet.nl</a></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Aut:</th>
<th>surfnet.nl = ns1.surfnet.nl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aut:</td>
<td>surfnet.nl = ns2.surfnet.nl</td>
</tr>
<tr>
<td>Aut:</td>
<td>surfnet.nl = ns3.surfnet.nl</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Add.</th>
<th>ns1.surfnet.nl = 195.99.147.113</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add.</td>
<td>ns2.surfnet.nl = 195.99.147.114</td>
</tr>
<tr>
<td>Add.</td>
<td>ns3.surfnet.nl = 195.99.147.115</td>
</tr>
</tbody>
</table>
Kaminsky in action

Attacker

Victim

Caching Resolver

Authoritative Name Server

voip.surfnet.nl A?

A = 195.99.147.125

SURFnet: we make innovation work
Response to Kaminsky’s work

• Immediate response
  – “Most massive patching effort on the Internet ever”
  – Randomise source port and query ID to make spoofing harder
  – Not really a solution; 32-bits is not a lot of entropy :-(

• Longer term response
  – Deploy DNSSEC
  – Often overlooked but just as important: implement measures to prevent packet spoofing! (BCP 38)
2013: source port and query ID random?

image courtesy of Pieter Lexis

University of Amsterdam
System & Network Engineering

SURFnet: we make innovation work
Herzberg-Shulman attack

- **Attacker**
- **Caching Resolver**
  - Spoofed fragment with spoofed auth. and additional
  - Contains (most of) the answer
- **Authoritative Name Server**
  - Contains auth. and additional section

surfnet.nl ANY?

```
Attacker
```

```
Caching Resolver
```

```
Authoritative Name Server
```
Herzberg-Shulman attack

- Presented at IETF 87 in Berlin this year
- Shows that DNS cache poisoning is even easier than thought
- Attack depends on guessing of 16-bit IPID value that is used to link fragments
  - IPID is highly predictable on some OS-es (passive port scan)
- And ironically, shows that DNSSEC makes this attack easier to perform :-S
Recap: DNS attack vectors

- Man in the middle
- Cache poisoning
- Data modification
- Data modification
- Spoofed updates
- Corrupt data
Bad news for the Internet :-(

source: http://lambicpeach.files.wordpress.com/2008/10/badnewspup.jpg
Good news!

DNSSEC
What is DNSSEC?

• DNSSEC was first devised in 1997

• We are at the third generation of the protocol
  - DNSSEC (ca. 2000)
  - DNSSECbis (2005)
  - NSEC3 (2008)

• Over 20 (!) active RFCs
  - That’s excluding the ‘normal’ DNS RFCs

• Protocol is mature
  - Changes are mainly new algorithms
What is DNSSEC?

- Digital Signatures guarantee authenticity of DNS records
  - Like a wax seal

- Resolvers validate the signatures and discard records with bogus signatures

- DNSSEC only provides authenticity
  - So no confidentiality
  - nor protection against DDoS-es
  - or typosquatting, phishing, etc.
Deployment status

• Widely deployed across the Internet
  – Major gTLDs like .com, .org, .net, etc.
  – over 33% of .nl domain names have DNSSEC enabled

Source: Internet Society (http://www.internetsociety.org/deploy360/dnssec/maps/)
• We measure the % of queries we can validate:
How does DNSSEC validation work?

trust anchor

root (.)

root zone

Root KSK public key

Root KSK private key

signs

Root ZSK public key

Root ZSK private key

nl DS record

contains

reference to

nl KSK public key

nl KSK private key

signs

nl zone

signed record for 'www.surfnet.nl'

surfnet

surfnet zone

surfnet.nl KSK public key

surfnet.nl KSK private key

signs

surfnet.nl ZSK public key

surfnet.nl ZSK private key

contains

surfnet.nl DS record

contains

reference to

surfnet.nl KSK public key

surfnet.nl KSK private key

signs

surfnet.nl ZSK public key

surfnet.nl ZSK private key

contains

signed record for 'www.surfnet.nl'
Example: a signed response

$ dig +dnssec www.surfnet.nl @192.87.106.106

... 

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6193
;; flags: qr rd ra ad QUERY: 1, ANSWER: 2, AUTHORITY: 6, ADDITIONAL: 13

... 

;; ANSWER SECTION:
www.surfnet.nl. 3541 IN A 145.0.2.10
www.surfnet.nl. 3541 IN RRSIG A 8 3 3600 20120209210255 20120202092011 65233 surfnet.nl. jBv79k4EvXt3bN6moWuY5Sr8KuUW4rDodzi03SMrbMgg9uBT7kdVRRzVweMF6vZBTxtaacefbMud41G...

SURFnet public resolver
Signature resource record
Data was validated successfully
Signature expiry date
Key identifier
Signature inception date
Signature data
What if validation fails?

• Some users will see this:
What if validation fails?

• But most users will see this:
Diagnosing problems

FOKKE & SUKKE
Quickly diagnose the problem

I think you’ve got diarrhea...
Diagnosing problems

• DNSSEC is a lot more complicated than plain old DNS (which is complex in its own right!)

• Troubleshooting validation failures is specialist work for trained experts

• We did a lot of this in the “early days”

• Lots of “big” failures (TLDs offline, big sites)

• Situation is much better now, we are slowly reaching technology maturity
Time for coffee or tea!

See you in 15 minutes!
Case study: DNSSEC signing at SURFnet

• SURFnet operates a managed DNS environment called ‘SURFdomeinen’

• We wanted to add DNSSEC

• Our goals:
  – To make it easy for our connected institutions to operate signed zones
  – To make it easy for ourselves to operate signed zones
SURFdomeinen

SURFnet: we make innovation work
Requirements

- DNSSEC should be a ‘box to tick’
  - Plain DNS is already considered complex by most users

- The integrity of zones should be guaranteed
  - SURFdomeinen should not be the weakest link to attack!

- Turning DNSSEC on or off should be quick
  - Ideally less than 1 hour

- Once DNSSEC is turned on, customers should not notice any difference
**Design: using HSMs**

- HSM = Hardware Security Module
- Secure and robust way to store DNSSEC key material
- We can never access the raw key material
- Role separation
- Standard API (PKCS #11)
- Disadvantage: expensive!
Security theatre?
Design: using OpenDNSSEC

- There are multiple options nowadays for software to use
  - Open source: BIND, PowerDNSSEC, OpenDNSSEC
  - Commercial: MS Windows 2012, InfoBlox, Xelerance, ...

- We participate in the OpenDNSSEC project
  - So it was a matter of “eating our own dogfood”

- Advantage of OpenDNSSEC: integrates well in an automated environment
Design: bump-in-the-wire
Design: network security

- **Colocation**
  - DNS VLAN
    - Authoritative DNS
    - OpenDNSSEC signer
  - HSM VLAN
    - Network HSM
    - SSL HSM

- **Network Architecture**
  - Internet
  - SURFnet WAN
  - Admin VLAN
    - firewall
    - SURFdomenen server

**Key Components**
- Authoritative DNS
- OpenDNSSEC signer
- Network HSM
- SSL HSM
- firewall
- local router
- SURFnet WAN
- SURFdomenen server
- Admin VLAN
This is what users see
Take-aways
or: why my job is so much fun ;-)  

• Setting up an environment like this is very complex
  – Staff with M.Sc. and Ph.D. degrees worked on this!

• Technically challenging, pioneering work
  – Extremely satisfying when it “just works”

• Valuable for our constituency and the wider Internet community
  – I travelled a lot the last couple of years to teach others
  – Proud to be on IETF list of people that made “DNSSEC” work :-)}
We don’t need no education!
or: using DNS to DDoS your school

Roland van Rijswijk - Deij
roland.vanrijswijk@surfnet.nl
DNS(SEC) amplification

1. 'bots' send queries with spoofed victim IP to open DNS resolvers.

2. Open DNS resolvers send responses to authoritative name servers.

3. Authoritative name servers send responses to victim.

Controlled by the attacker.

Traffic volume: N bits/s

Traffic volume: N x amp bits/s
Let’s ask comcast.net a question

$ tcpdump -n -v -i en0 host xxxx
...
11:00:19.411981 IP (... proto UDP (17), length 68)
...
11:00:19.430637 IP (... proto UDP (17), length 1500)
    xxxx.53 > yyyy.55023: 36075$ 3/6/29 comcast.net. MX ...
11:00:19.430640 IP (... length 1500)
    xxxx > yyyy: udp
11:00:19.430641 IP (... length 297)
    xxxx > yyyy: udp

Send: 68 bytes, recv: 3297 bytes, amp. ≈ 48.5x!
A small example

• Recent attack against Spamhaus infra we host:

Yes, it really says 38 Gigabits/s
Another example: Abuse of SURFnet name servers

*Outgoing traffic before filtering*

*Incoming traffic hardly increases*
Diving deeper: a case study

• In May 2012 we saw a large increase in the number of queries to our infrastructure

• 3 out of 4 auth. started showing large numbers of ANY queries for gigaport.nl (DNSSEC!)

• Not to the server in Switzerland (neutral?)

• Initially regular DNS queries (without EDNS0)
It showed in our statistics...
Profiling the attack: HAL 9000

• Queries use EDNS0
• With a very exotic maximum response size of 9000 bytes
• And we are not the only ones being abused...
  – 3 name servers of SURFnet customers
  – Also in the UK and across the globe
• Largest common denominators:
  – queries for DNSSEC-signed zones
  – all with EDNS0, buffer size 9000, DNSSEC OK = 0 (!)
  – fixed query ID, 25 subsequent queries, spoofed sender IP
“I’m sorry Dave,
I’m afraid I can’t do that”

• Hmm... this ain’t good, we gotta do something...

• Arbor TMS filter:
  \x08gigaport\x02nl\x00\x00\xff
  \x00\x01\x00\x00\x29\x23\x28\x00\x00\x00\x00\x00\x00\x00

• IP tables filter:
  $ python gen-dns-rule.py --qname gigaport.nl --qtype ANY --bufsize 9000
  0>>22&0x3C@20=0x08676967&&0>>22&0x3C@24=0x61706f72&&0>>22&0x3C@28=0x74026e6c&&0>>22&0x3C@32&0x00FFFFFF00=0x0000FF00&&0>>22&0x3C@36&0x00FFFFFF=0x00000029&&0>>22&0x3C@40&0x00FFFF0000=0x23280000
Clues in the packets

• IP packets contain a “time-to-live” field
  – Max. value is 255, is lowered on every hop (or every second)
  – If you know the starting value: good measure for “distance”

• All query packets in the attack had a TTL ≥ 250
  – That almost certainly means they set the initial value to 255, so the packets traversed at most 6 hops

• That is “very close” in network terms
  – E.g. route from ns1.surfnet.nl to www.xs4all.nl has 6 hops
So where is it really coming from?
Let’s drop a peering on the AMS-IX

Bingo :-)
Keep your friends close...

- We see this attack being used against our customers
- One of our customers appeared in a list of spoofed IP addresses for this particular attack
- And was the victim of a number of DDoS attacks
- The timing of the attacks was rather suggestive...

*Always during school hours!*

- Let’s have a look at what the school found
Let’s see what happens if...

- The external NAT IP address is changed
  - Will the attack follow?
- We look at the time lines
  - Comparing attack times against class schedule
- We ask teachers about suspicious behaviour
  - Are there signs that the culprit is among the students?
- Policy-Based Routing (PBR)
  - Giving a suspected class a different external IP address

*with thanks to “Graafschap College”!*
Ladies and gentlemen: we got him!

• Time to put the fear of God in someone :-)

CENSORED
Booters

ABOUT

Rage Booter is powered by quick, strong, and DDoS protected servers to guarantee uptime and stability. With the constantly growing usage of our service, we upgrade our servers regularly.

Rage Booter combines two powerful synthesized floods: Amplified UDP Reflection and Spoofed SYN, to give you the power that you need. Rage Booter utilizes both of these flood techniques to give your servers the most realistic DDoS attack.

DDOS Types:
* HTTP FLOOD
* HIGH HTTP FLOOD
* TCP FLOOD
* UDP FLOOD
* SYN FLOOD
And much more!

Contact: Skype: Or send me a private message
Payment: Paypal (Gift only)
Price: $4.50 USD / Hour
24 hours = $100 USD
And “shells”
What can we do about it?

• Make sure there are no open resolvers

• The only real solution: deploying BCP38
  – BCP38 = *ingress filtering*; only allow traffic onto a network with legitimate IP addresses (SURFnet does this network-wide)

• We actively monitor and filter the attacks

• DNS *rate limiting* is under heavy discussion
  – Patch available for BIND
  – Integrated in NSD 4 by NLnet Labs
  – But beware! Rate limiting may affect legitimate traffic!
Homework

• Read paper about DDoS potential in DNSSEC

• Answer 2 questions about a recommendation from the paper

• Due next week; more info on Blackboard
Special thanks to:

Pieter Lexis  (visualisation of DNS data)
Xander Jansen  (DDoS investigations)