DNS Pentesting

Sunny Neo
Santosh Ananthakrishnan

Recap

- DNS can be used as a communication channel - not just for name resolution
- DNS TXT record based XSS attack
- Building tools that use this channel for stealthy pentesting!

Our Work

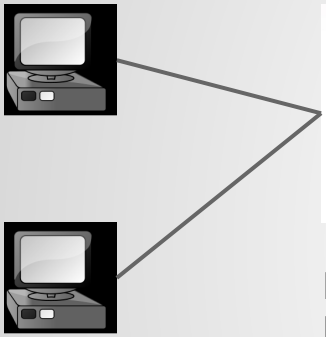
- UDP Hole Punching with DNS
- Staging exploit over DNS
- DNS performance as a channel

UDP Hole Punching

- A common technique for 2 clients to establish connection behind NAT

- A common misconception that clients behind NAT are not able to provide services without port forwarding

NAT “Security Feature”

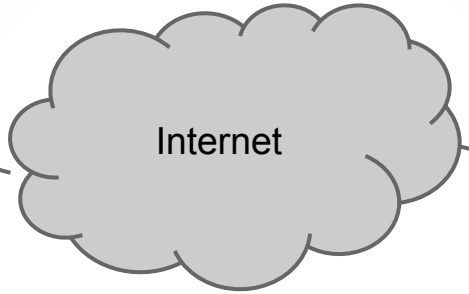


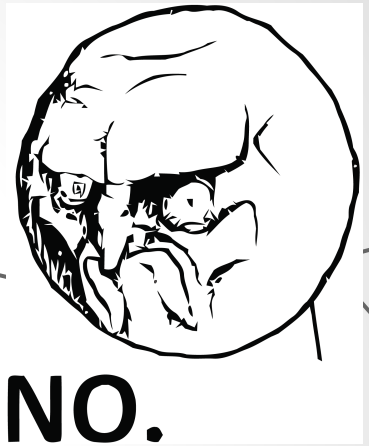
Client B

wants to connect to Client A Streaming Service via 199.20.199.10: 3000

NAT Router 1
Public IP: 10.20.99.11

Private IP: 192.168.1.1 /24





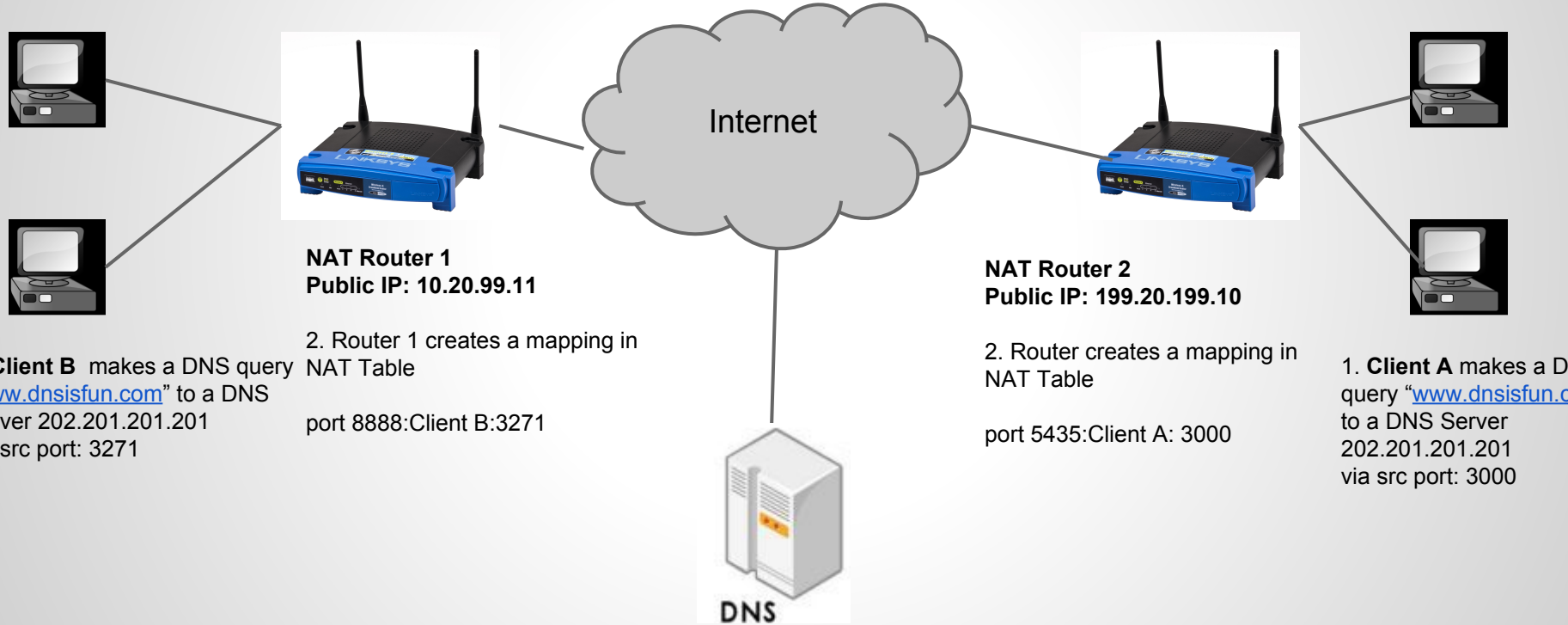
NAT Router 2
Public IP: 199.20.199.10

Private IP: 192.168.1.1/24

No port forwarding setup

Client A Running Streaming service on UDP port 3000

What if



1. **Client B** makes a DNS query "www.dnsisfun.com" to a DNS Server 202.201.201.201 via src port: 3271

NAT Router 1
Public IP: 10.20.99.11

2. Router 1 creates a mapping in NAT Table

port 8888:Client B:3271

NAT Router 2
Public IP: 199.20.199.10

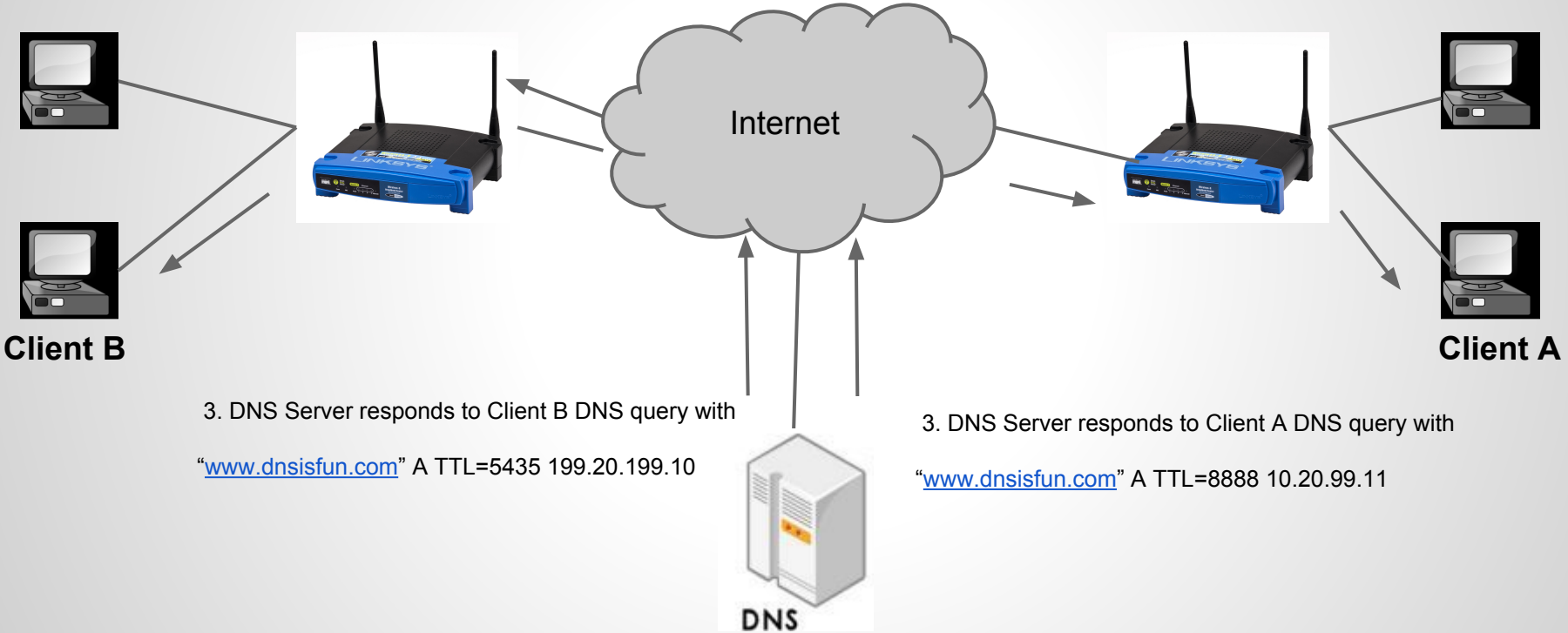
2. Router creates a mapping in NAT Table

port 5435:Client A: 3000

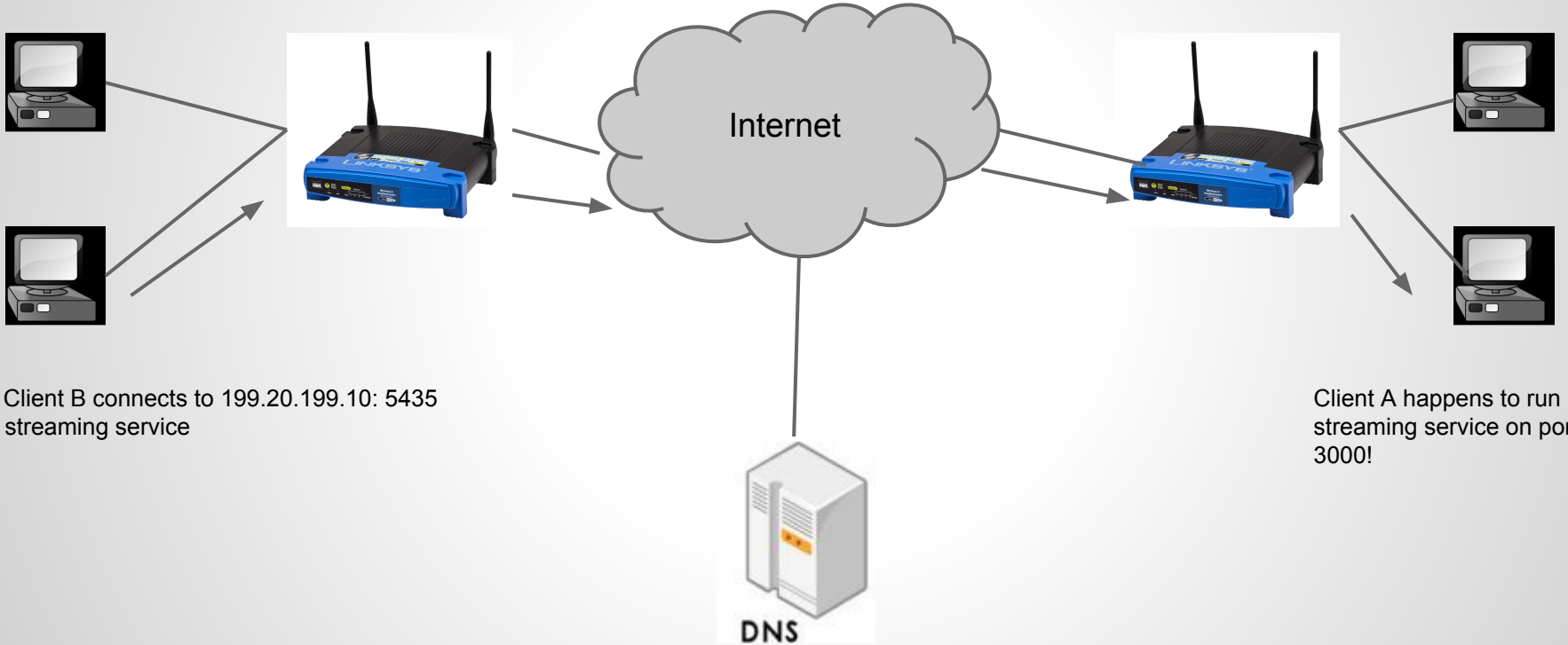
1. **Client A** makes a DNS query "www.dnsisfun.com" to a DNS Server 202.201.201.201 via src port: 3000

IP: 202.201.201.201

What if


Internet
Client B
Client A
3. DNS Server responds to Client B DNS query with
"www.dnsisfun.com" A TTL=5435 199.20.199.10
3. DNS Server responds to Client A DNS query with
"www.dnsisfun.com" A TTL=8888 10.20.99.11
DNS
IP: 202.201.201.201

What if



4. Client B connects to 199.20.199.10: 5435 for streaming service

Client A happens to run streaming service on port 3000!

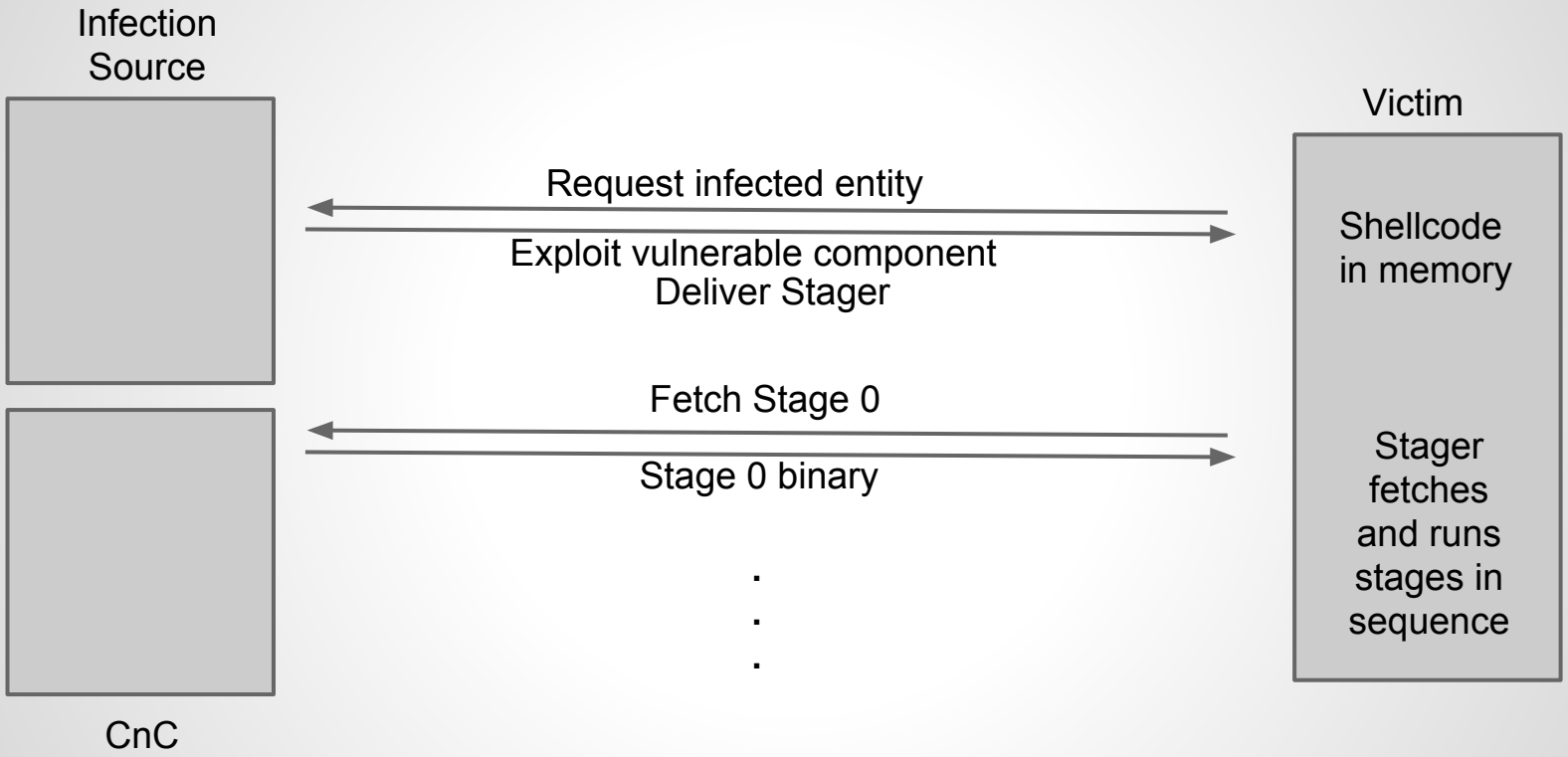
IP: 202.201.201.201

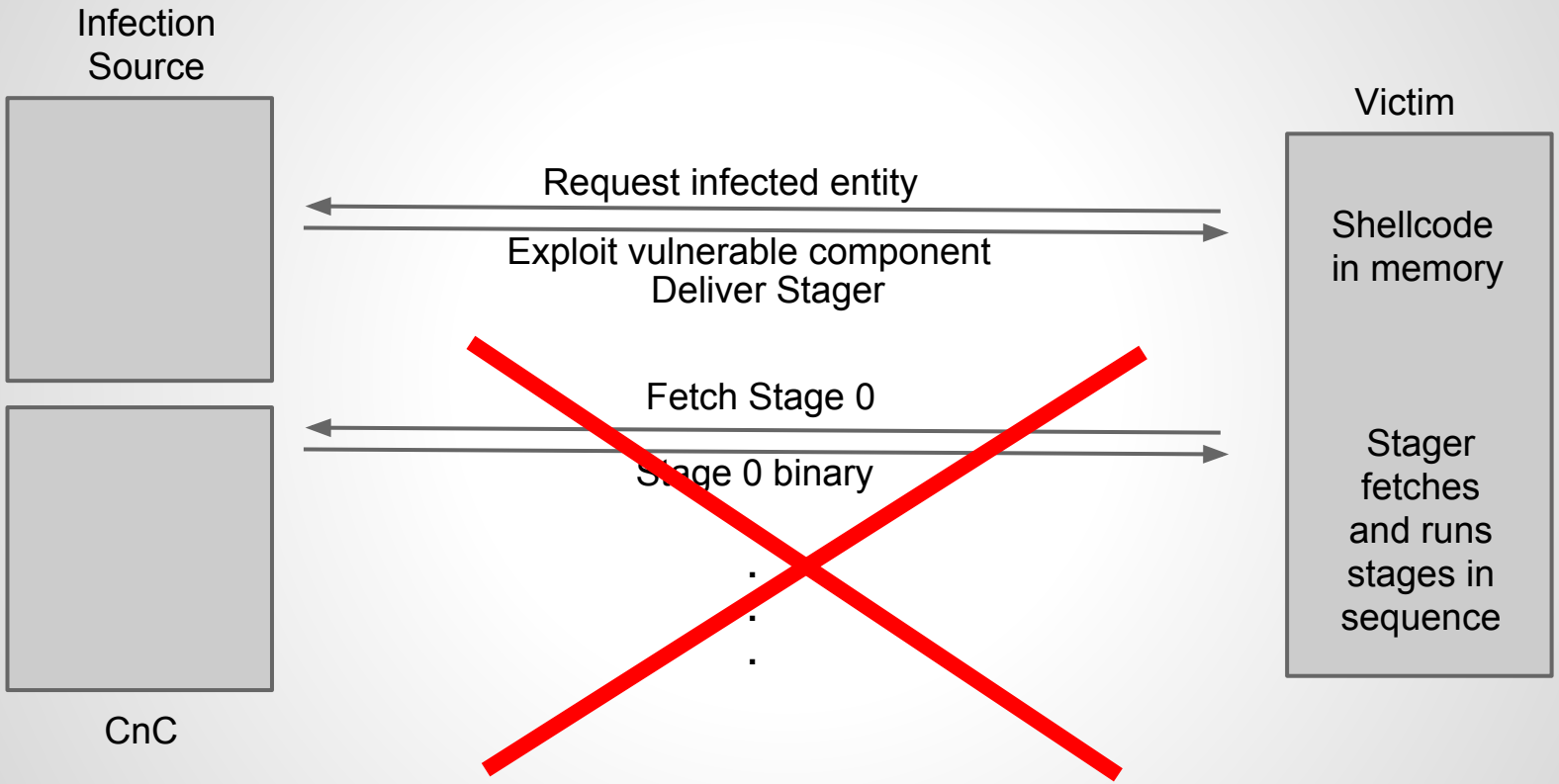
Why this works

- Legitimate DNS query and response
- Most NAT does not implement IP Address/Port Restriction for mapping
- Could this be used for P2P Malware?

Stagers

- Small exploits are typically *inline* - the payload is delivered with the exploit as shellcode
- Tools that abstract out the payload delivery channel from the payload itself are called stagers


Infection Source
Victim
Request infected entity
Exploit vulnerable component
Deliver Stager
Shellcode in memory
Fetch Stage 0
Stage 0 binary
Stager fetches and runs stages in sequence
.
.
.
CnC


Infection Source
CnC
Victim
Request infected entity
Exploit vulnerable component
Deliver Stager
Fetch Stage 0
Stage 0 binary
.
.
.
Shellcode in memory
Stager fetches and runs stages in sequence

Staging over DNS

- The stager stealthily polls an A-server controlled by the CNC
- When a stage is ready, the A-server responds with the number of chunks to fetch encoded in the last two bytes of the response IP address
- 141.232.01.10 = 266 chunks

Staging over DNS

- The stager then fetches these chunks at random, also over DNS
- <encoded req for chunk N>.evildom.com
- The response includes several resource records (RRs)
- Each RR corresponds to an encoded block of bytes

Staging over DNS

- Subsequent requests can be made far apart - days or more!
- Eventually re-assemble and execute payload
- Use dynamic DNS providers for even more indirection

DEMO

Potential Problems

- Analysis of DNS requests
  - CDNs and other abusive services have *huge* volume
  - The stager can slow down queries arbitrarily
- Slow Transfer Rate
  - This is obviously only for cases where there are no other channels, or when no direct TCP connections are to be made to the CnC, tradeoff.

Performance

- How good is DNS as a channel, really?

	Receiver				
	NJ 1	**NJ 2**	**LA**	**NLD**	**JPN**
NJ 1	-	2981/2981	2888/2889	2964/2964	3053/3054
NJ 2	3016/3016	-	3100/3101	2734/2735	3054/3054
LA	2901/2941	2932/2975	-	2938/2942	2712/2712
NLD	3038/3038	2771/2772	2724/2724	-	2791/2791
JPN	2551/2552	2886/2886	2836/2838	2887/2887	-

UDP Reliability (Source: http://openmymind.net/How-Unreliable-Is-UDP/)

References

[1]http://en.wikipedia.org/wiki/UDP_hole_punching

[2]http://resources.infosecinstitute.com/udp-hole-punching/