UTNHAEDO
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL HAEDO

Security Implications of Network Address Translators (NATs)

(draft-gont-behave-nat-security)

Fernando Gont
UTN/FRH

Pyda Srisuresh
Kazeon Systems, Inc.

73rd IETF meeting, November 16-21, 2008
Minneapolis, MN, USA

What motivated this document?

- Earlier this year, a number of vulnerabilities were found in popular DNS implementations
- In order to exploit these vulnerabilities, an attacker had to guess the four-tuple {source IP, Source port, Destination IP, Destination port}
- Some implementations were randomizing the ephemeral ports of their DNS requests, thus making it harder for an attacker to exploit these vulnerabilities
- Yet sometimes these systems were behind a NAT
  - The NAT would rewrite the source port of outgoing packets, using a global linear sequence
  - As a result, this was as bad as if the end-systems were not doing port randomization in the first place

Document overview

- Based on the aforementioned experience, we tried to analyze the security implications of NATs rewritting (or NOT rewriting!) each of the header fields of the involved protocols
- In many cases, there are **interoperability** implications if some header fields are **not** rewritten. Therefore, if they must be rewritten… why not do it in the right way?
- Some issues have been discussed in detail in this first version of the document:
  - Security implications arising from IP fragmentation
  - DHCP-configured NATs
  - Security implications of some header fields

Example of (not?) rewriting header fields (I)

- Source port
  - You don't: Potential of interoperability problems (collision of connection-id's)
  - You do it "wrong": Easier to predict future connection-id's
- TCP Sequence numbers
  - You don't: Potential of data corruption
  - You do it "wrong": Easy to predict future sequence numbers
- TCP timestamps
  - You don't: Potential of data corruption or connection failures
  - You do it "wrong": Easy to predict future values
- IP Identification
  - You don't: Potential of data corruption (collision of IP ID's), leaks out number of systems behind a NAT
  - You do it wrong: leaks information (e.g., packets transmitted)

Rewriting the source port

- RFC 5382 leaves this unspecified
- RFC 4787 states:
  - *A NAT MUST NOT have a "Port assignment" behavior of "Port overloading“*
  - It is RECOMMENDED that the port ranges (whether 0-1023 or 1024-65535) is respected
  - *Applications must, therefore, be able to deal with both port preservation and no port preservation.*
- Options:
  - Always randomize the source port?
  - Randomize the source port unless you are doing port preservation?

Feedback we've got so far…

- Much feedback from Dave Thaler, Dan Wing, and others.
- Rewriting the source port
  - There was some discussion on-list
  - Question: Does it still really make sense to do port preservation?
  - Possible outcome: Randomize the source port unless you are doing port preservation?
- Rewriting the TTL
  - Comment: May break traceroute!
  - Answer: How about rewriting the TTL when it is largen that, e.g., 50?
- We plan to publish a revision of this document any time soon

Moving forward

Should this document be adopted as a BEHAVE WG item?