THE CURSE OF NUMERIC IDENTIFIERS (in network protocols)

Iván Arce – Programa de Seguridad en TIC Fundación Dr. Manuel Sadosky

PROTOSEC 201.
April 2, 2016 - Buenos Aires, Argentina

Ministerio de
Ciencia, Tecnología
e Innovación Productiva
Presidencia de la Nación

fundación
SADOSKY
Investigación y Desarrollo en TIC

cessi
ArgenTIna

CICOMRA

The Security in ICT Program (STIC) at Fundación Sadosky

What is the Fundación Dr. Manuel Sadosky

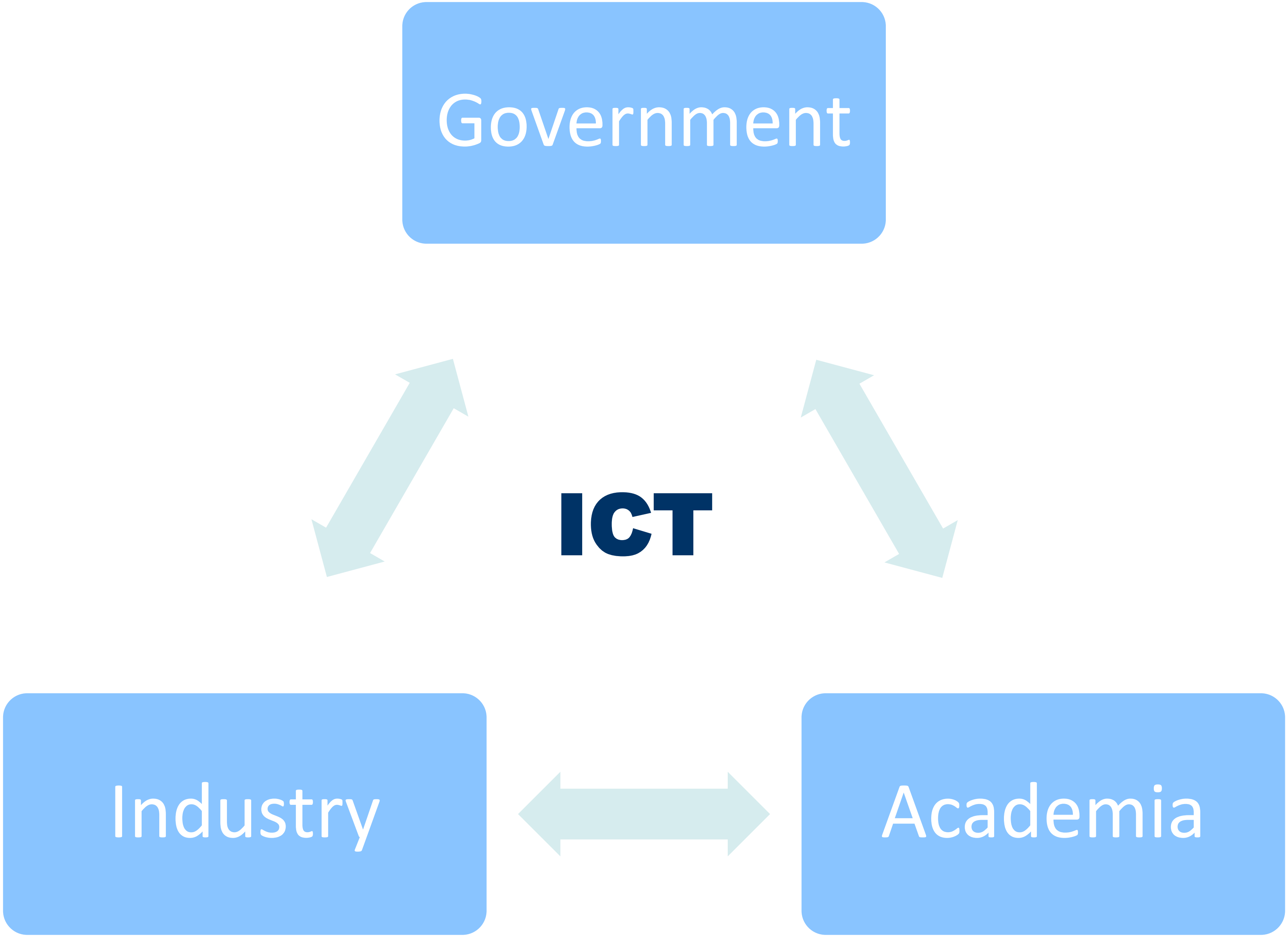
- The Sadosky Foundation is a mixed (public-private) institution whose goal is to promote a closer and stronger interaction between Academia, Government and Industry in all fields related to Information and Communications Technology

- It was formally created by the Argentine Government through a Presidential decree in July 2009, and started operations in April 2011

- It is named after Argentina's and Latin America's computer science pioneer Manuel Sadosky

Manuel Sadosky (1914-2005)
http://www.fundacionsadosky.org.ar/biografia-dr-manuel-sadosky/

What does Fundación Dr. Manuel Sadosky do?



Government

ICT

Industry

Academia

Our vision

“Information and Communication Technology as key enabler for an enterprising society that fosters and pursues knowledge creation, productive and sustainable innovation , economic competitiveness and the continued improvement of the quality of life of the Argentina population **minimizing the risk of technological dependency and vulnerability of our critical infrastructure**“

Enough of that!
Now, let’s talk about numbers

Why would we even want to do that?

- Common issue in several protocol specifications at various layers
- Common issue repeated in protocol specifications for over 30+ years
- [AFAIK] There is no cross-WG, security-oriented approach to address this problem
- Solutions & mitigations proposed or adopted in one protocol do not "travel" to other protocols

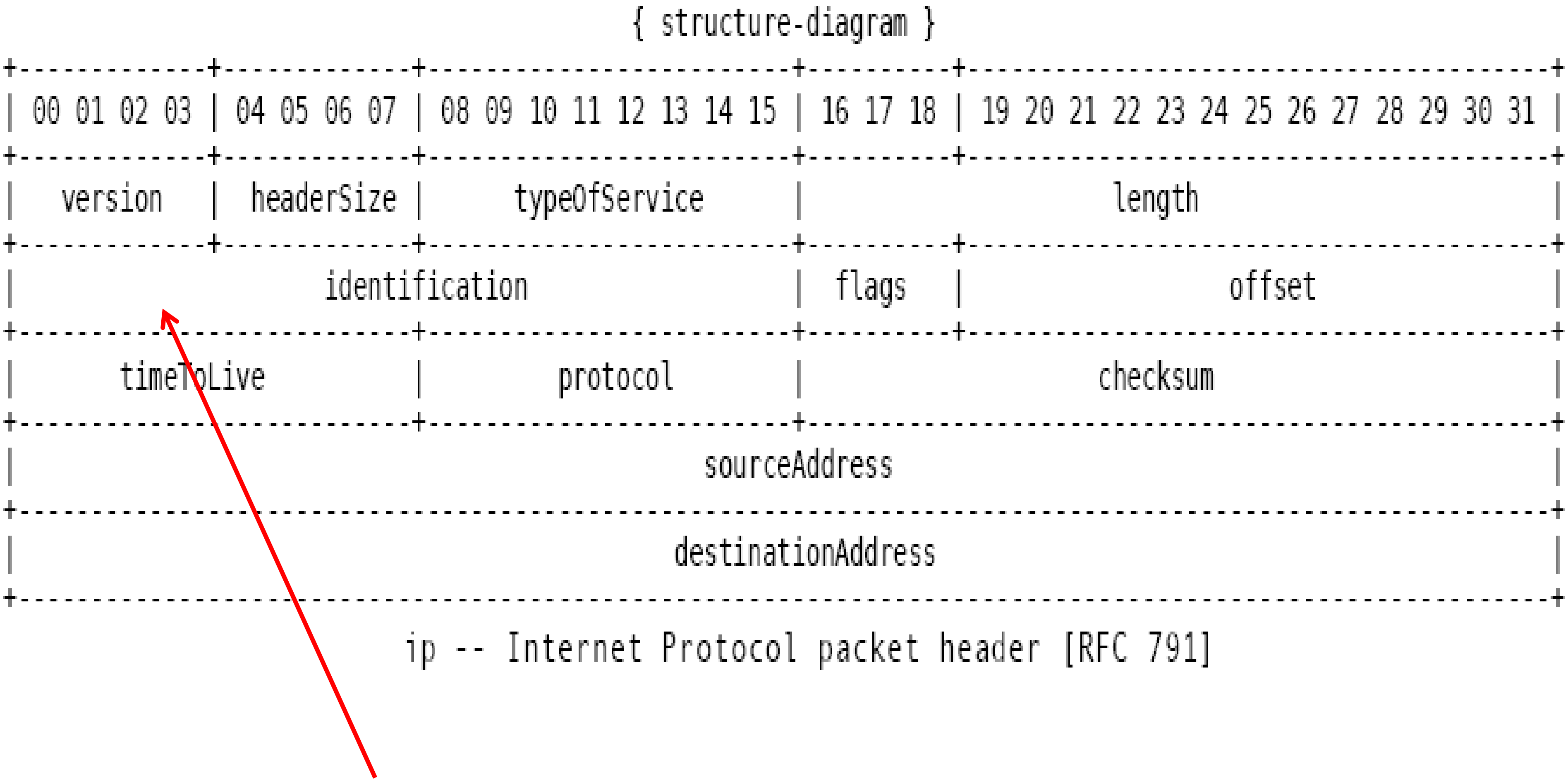
CAVEAT: this is Work in Progress

Hmmmkey… but what do you mean with “numeric IDs”?

- Identifier:

  A data object in a protocol specification that can be used to definitely distinguish a protocol object (a datagram, network interface, transport protocol endpoint, session, etc) from all other objects of the same type, in a given context. Identifiers are usually defined as a series of bits and represented using integer values. We note that different identifiers may have additional requirements or properties depending on their specific use in a protocol. We use the term "identifier" as a generic term to refer to any data object in a protocol specification that satisfies the identification property stated above.

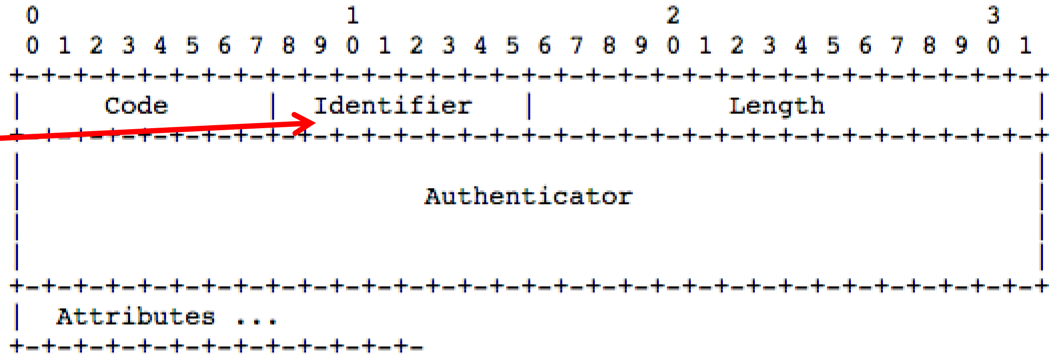
WTF? That sounds utterly complicated and academicshy



```
                              { structure-diagram }
+-------------+-------------+-------------------------+----------+-------------------------------------+
| 00 01 02 03 | 04 05 06 07 | 08 09 10 11 12 13 14 15 | 16 17 18 | 19 20 21 22 23 24 25 26 27 28 29 30 31 |
+-------------+-------------+-------------------------+----------+-------------------------------------+
|   version   | headerSize  |       typeOfService     |                 length                         |
+-------------+-------------+-------------------------+----------+-------------------------------------+
|                 identification                    |  flags   |                offset               |
+---------------------------+-------------------------+----------+-------------------------------------+
|        timeToLive         |         protocol        |                 checksum                       |
+---------------------------+-------------------------+------------------------------------------------+
|                                          sourceAddress                                              |
+-----------------------------------------------------------------------------------------------------+
|                                        destinationAddress                                           |
+-----------------------------------------------------------------------------------------------------+
                      ip -- Internet Protocol packet header [RFC 791]
```

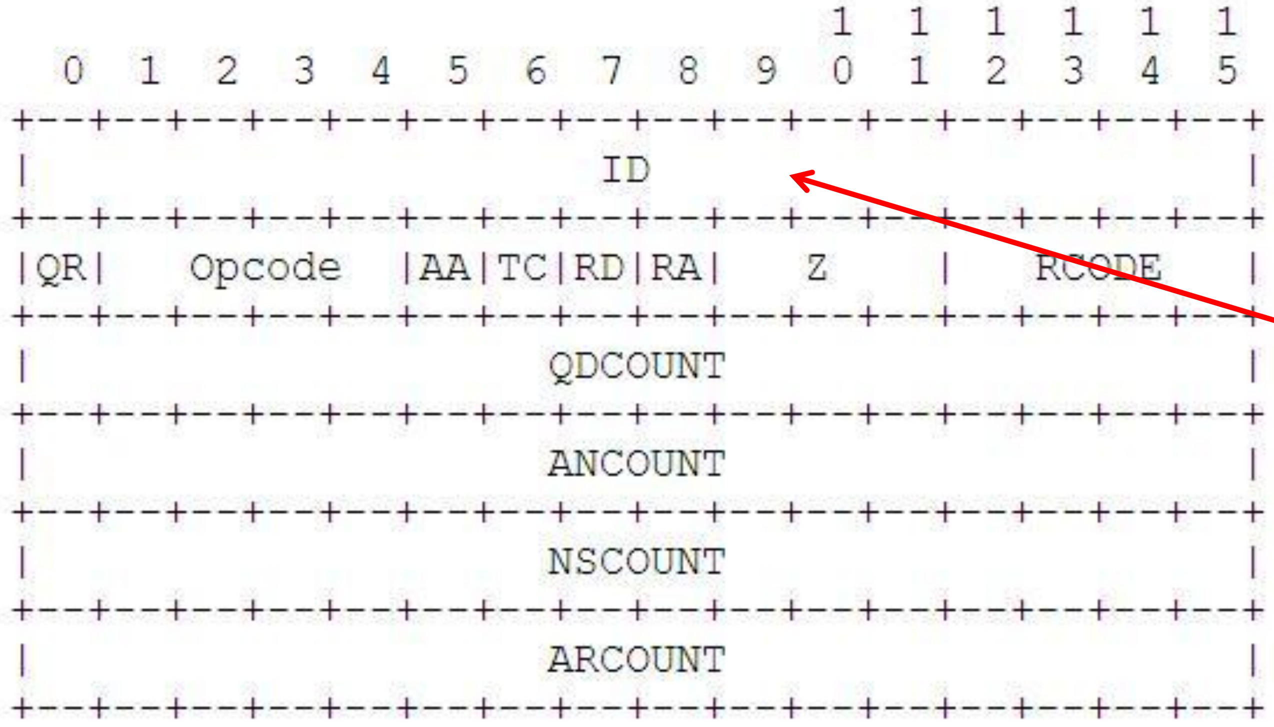
This guy here! IPv4 Identification field, RFC 791

Thank you, can I have another?



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Authenticator                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

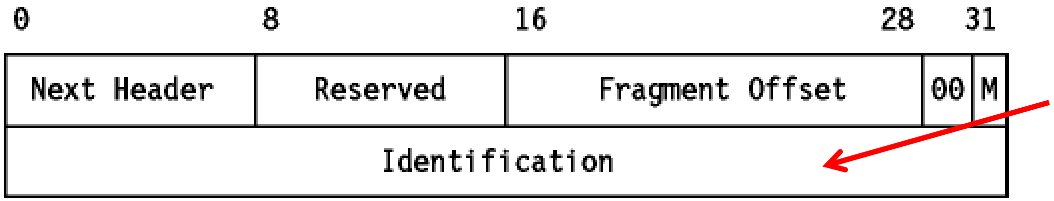
RADIUS request ID
(RFC 2865)



```
                              1  1  1  1  1  1
  0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                      ID                       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|QR|   Opcode  |AA|TC|RD|RA|   Z    |   RCODE   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    QDCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    ANCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    NSCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    ARCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

DNS query ID
(RFC 1035)

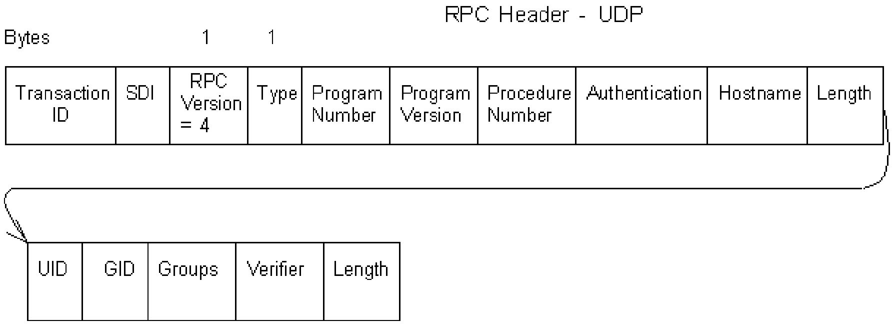
and there is more…



0	8	16	28	31
Next Header	Reserved	Fragment Offset	00	M
Identification				

IPv6 fragmentation header
RFC 2460 & 2460bis

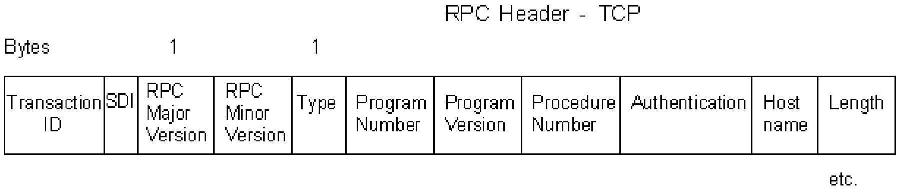
ONC RPC transaction ID
RFC 5531 (originally RFC 1831**)**



RPC Header - UDP

Bytes		1	1						
Transaction ID	SDI	RPC Version = 4	Type	Program Number	Program Version	Procedure Number	Authentication	Hostname	Length

UID	GID	Groups	Verifier	Length

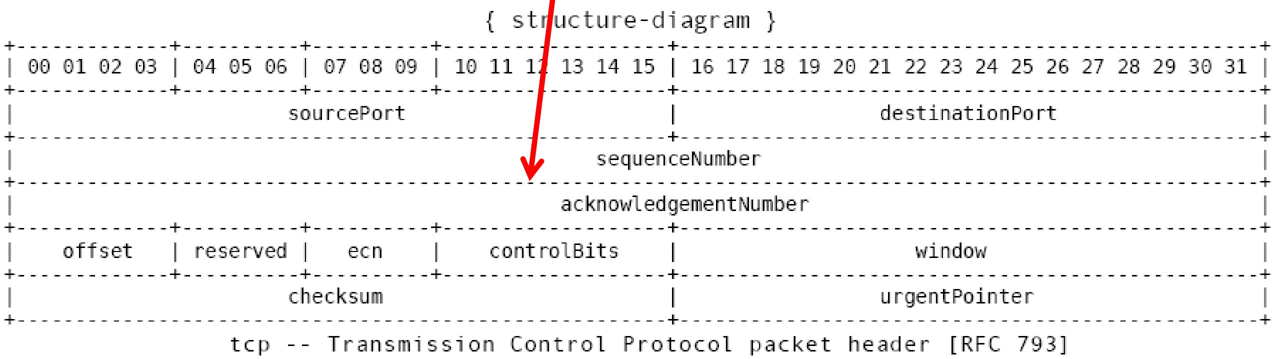


RPC Header - TCP

Bytes		1		1						
Transaction ID	SDI	RPC Major Version	RPC Minor Version	Type	Program Number	Program Version	Procedure Number	Authentication	Host name	Length

etc.

TCP Sequence number
RFC 783



```
                        { structure-diagram }
+-------------+----------+----------+----------+-----------------------------------------------+
| 00 01 02 03 | 04 05 06 | 07 08 09 | 10 11 12 13 14 15 | 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 |
+-------------+----------+----------+----------+-----------------------------------------------+
|                    sourcePort                |                destinationPort                |
+----------------------------------------------+-----------------------------------------------+
|                                       sequenceNumber                                         |
+----------------------------------------------------------------------------------------------+
|                                    acknowledgementNumber                                     |
+-------------+----------+----------+----------+-----------------------------------------------+
|   offset    | reserved |   ecn    | controlBits |                  window                     |
+-------------+----------+----------+----------+-----------------------------------------------+
|                    checksum                  |                 urgentPointer                 |
+----------------------------------------------+-----------------------------------------------+
              tcp -- Transmission Control Protocol packet header [RFC 793]
```

wait…WHAT?

Sequence numbers aren’t IDs…

They aren’t but they have *ID-like semantics* so…
…sometimes they are:

- TCP ISN
- RTP sequence number
- OSPF DD sequence number
- IKEv2 Message ID

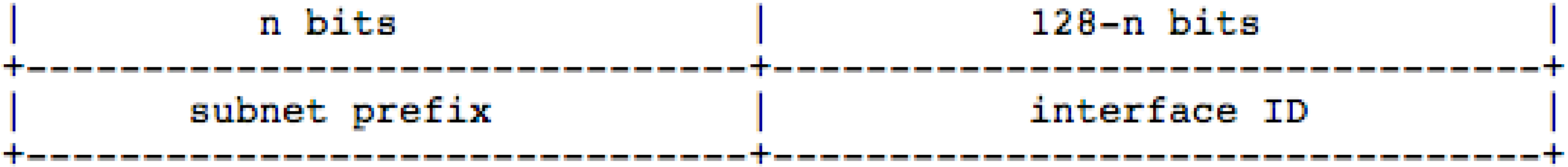
…

So, we settled on:
“can be used to definitely distinguish a protocol object from all other objects of the same type, in a given context. Identifiers are usually defined as a series of bits and represented using integer values”

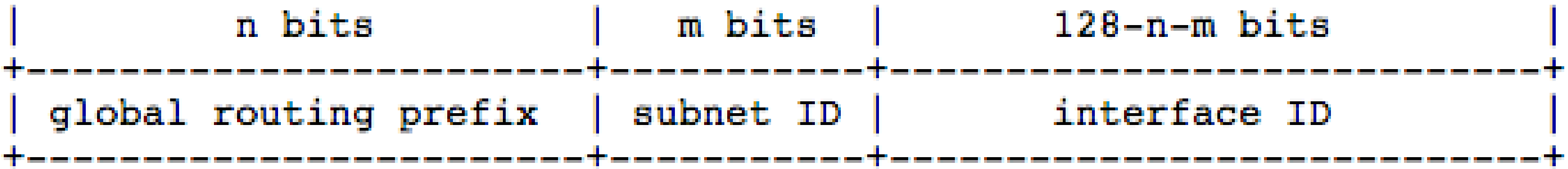
An initial *seqnum* identifies a byte in stream|packet in session

OK so Sequence Numbers may be IDs, what else?

IPv6 addresses: “*IP Version 6 Addressing Architecture*” - RFC 4291

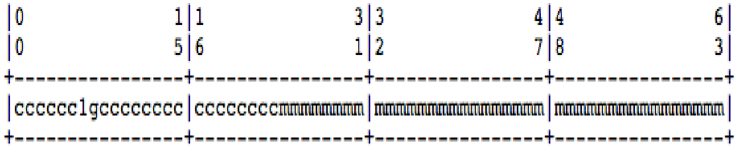


```
|          n bits               |           128-n bits            |
+-------------------------------+---------------------------------+
|       subnet prefix           |           interface ID          |
+-------------------------------+---------------------------------+
```



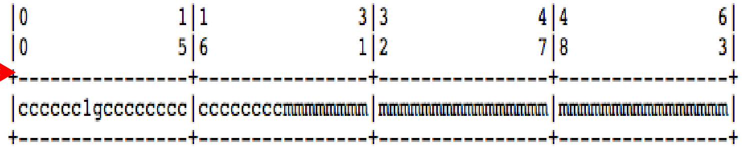
```
|         n bits          |   m bits   |       128-n-m bits         |
+-------------------------+------------+----------------------------+
| global routing prefix   | subnet ID  |       interface ID         |
+-------------------------+------------+----------------------------+
```

IEEE EUI-64 64 bit Identifier

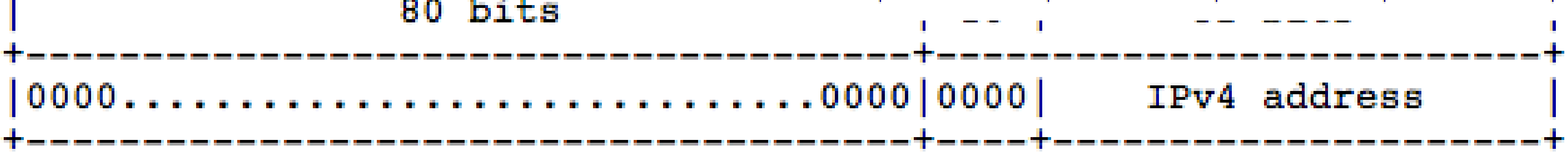


```
|0              1|1              3|3              4|4              6|
|0              5|6              1|2              7|8              3|
+----------------+----------------+----------------+----------------+
|cccccc1gcccccccc|ccccccccmmmmmmmm|mmmmmmmmmmmmmmmm|mmmmmmmmmmmmmmmm|
+----------------+----------------+----------------+----------------+
```

IEEE 802 48 bit MAC address



```
|0              1|1              3|3              4|4              6|
|0              5|6              1|2              7|8              3|
+----------------+----------------+----------------+----------------+
|cccccc1gcccccccc|ccccccccmmmmmmmm|mmmmmmmmmmmmmmmm|mmmmmmmmmmmmmmmm|
+----------------+----------------+----------------+----------------+
```



```
|                80 bits               | 16 |      32 bits        |
+--------------------------------------+--------------------------+
|0000..............................0000|0000|    IPv4 address     |
+--------------------------------------+----+---------------------+
```

Why does this guy’s rant matter at all?

There is no such thing as “*free semantic overload*”

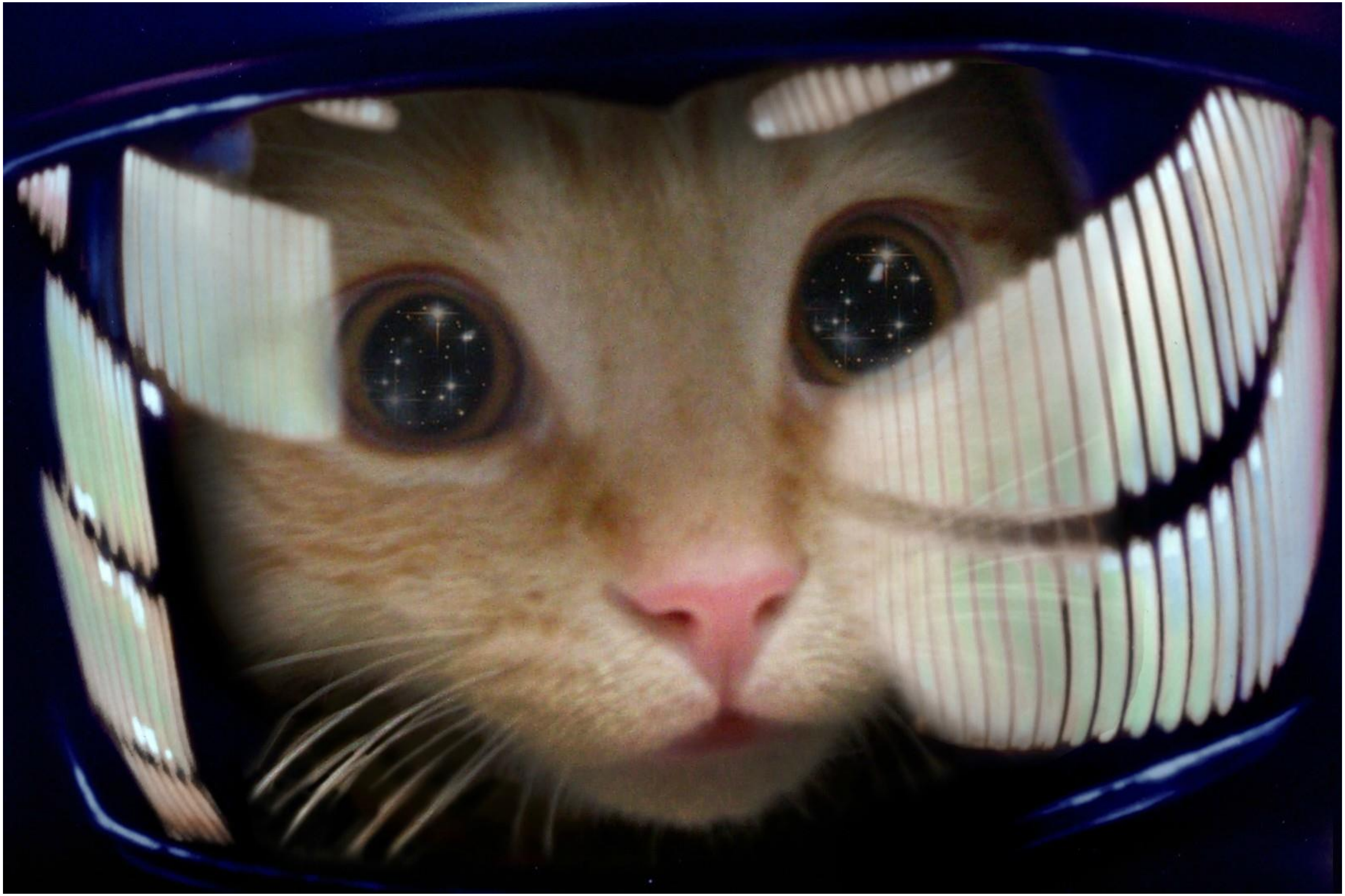
- ID -> SeqNum: Ordering relationship, wrap-around
- ID -> Address: Topological information, leakage
- Address -> ID: Context free tracking, leakage

Unique != !Predictable
!Predictable != !Collisionable

Depending on { transport | network } layer context:

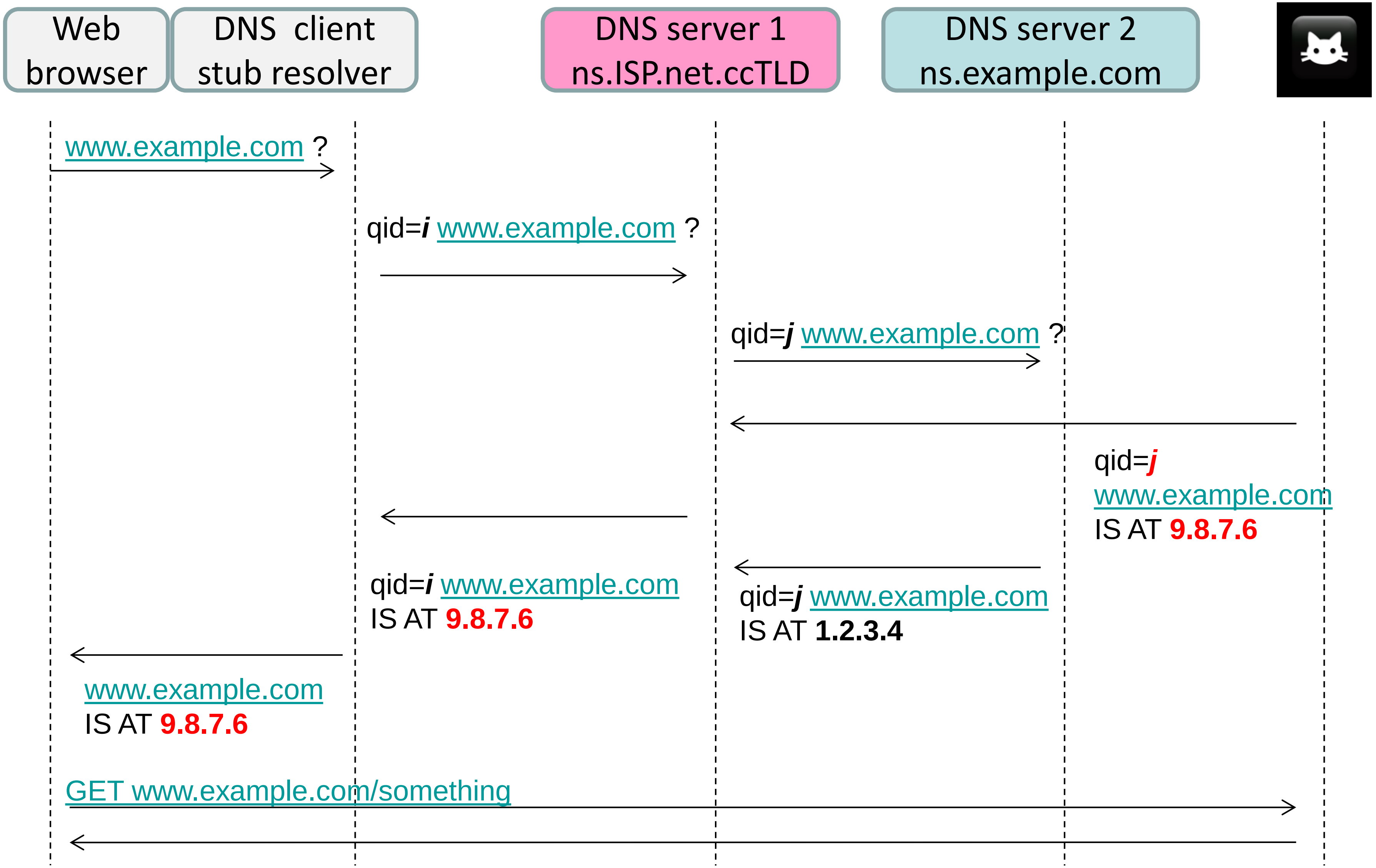
- Predictability -> DoS
- Spoofing + Predictability -> DoS, hijacking, evasion,
- Spoofing + Ordering relationship -> DoS, insertion, hijacking
- ANY semantic added –> leakage_risk++

Oh my God, it's full of NUMERIC IDENTIFIER FIELDS!!



Security & Privacy issues with numeric IDs

DNS cache poisoning



Web browser

DNS client stub resolver

DNS server 1 ns.ISP.net.ccTLD

DNS server 2 ns.example.com

www.example.com ?

qid=*i* www.example.com ?

qid=*j* www.example.com ?

qid=*j* www.example.com IS AT **9.8.7.6**

qid=*j* www.example.com IS AT **1.2.3.4**

qid=*i* www.example.com IS AT **9.8.7.6**

www.example.com IS AT **9.8.7.6**

GET www.example.com/something

404 Unauthorized:
WWW-Authenticate: BAD KITTY SAYS: YO! I AM EXAMPLE DOT COM GIIMME ALL UR SEKRETZ, MEOW !

Why did this happen?

DNS Query ID:

- was predictable (very)
- is only 16 bits wide
- source port is fixed (53)
- spoofing replies is possible (UDP)

How predictable?

```
qid = 0;
for each new query {
        set query.qid to qid;
        send query;
        qid = qid +1;
}
```

Chronology: DNS poisoning due to a predictable query ID

1987: Mockapetris P., RFC 1035 Domain Name Implementation and Specification
Query ID is a 16 bit integer. Use it to match <query, response>

1993 Schuba C. & Spafford E., *Addressing weaknesses in the DNS protocol*
Describes cache poisoning
Cache poisoning possible is QID predicted (see TCP ISN attack!).
On-path attacker with root on a nameserver can do it.

1995 Vixie P.,*DNS and BIND security issues*
Resolvers should match QID in response to sent request
UDP source port, Query ID predictable (16 bits)

1997 Arce I. & Kargieman E., *BIND vulnerabilities and solutions*
Query ID implementation: A global variable, initialized to 0, incremented by 1
Query ID is easily predicted, resolvers use fixed source port
Off-path attack is possible: Just send recursive query to resolver and get its qid
Fix: Randomize QID, randomize source port

Chronology: DNS poisoning due to a predictable query ID

2002 Sacramento V., *Vulnerability in the sending requests control of Bind versions 4 and 8 allows DNS spoofing*

Randomized Query ID not good enough

Birthday attack: spoof N request with diff src address, spoof matching reply

2007 Klein A. *BIND 9 DNS Cache Poisoning*

Attacker can force resolver to make many queries

Birthday attack still works.

BIND PRNG state can be recovered, Query ID easily predictable .

2008 Kaminsky D., *Black ops 2008-it's the end of the cache as we know it.*

Attacker can force resolver to make many queries

Birthday attack STILL works

Attacker spoofed replies set an authoritaive server for target domain.

Attacker always wins.

Patch: Randomize QID & source port (?!)

Fix: Use DNSSEC (meh)

Chronology: DNS poisoning due to a predictable query ID

2009 Hubert A, van Mook R., *RFC 5452 Measures for Making DNS More Resilient against Forged Answers*

2010 Economou N., *Windows SMTP Service DNS query Id vulnerabilities*

Windows STMP resolver used naïve Query ID generator (+1 sequence)

Does not match QID in reply to QID sent in request

23 years after [RFC 1035] said ”check query ID” a massively deployed resolver did not.

TCP Initial Sequence Numbers (ISN)

September 1981:

[RFC0793], suggests the use of a global 32-bit ISN generator, whose lower bit is incremented roughly every 4 microseconds. However, such an ISN generator makes it trivial to predict the ISN that a TCP will use for new connections, thus allowing a variety of attacks against TCP.

February 1985:

[Morris1985] was the first to describe how to exploit predictable TCP ISNs for forging TCP connections that could then be leveraged for trust relationship exploitation.

May 1989

[Bellovin1989] discussed the security implications of predictable ISNs (along with a range of other protocol-based vulnerabilities).

February 1995:

[Shimomura1995] reported a real-world exploitation of the attack described in 1985 (ten years before) in [Morris1985].

TCP Initial Sequence Numbers (ISN)

May 1996:

[RFC1948] was the first IETF effort, authored by Steven Bellovin, to address predictable TCP ISNs. The same concept specified in this document for TCP ISNs was later proposed for TCP ephemeral ports [RFC6056], TCP Timestamps, and eventually even IPv6 Interface Identifiers [RFC7217].

March 2001:

[Zalewski2001] provides a detailed analysis of statistical weaknesses in some ISN generators, and includes a survey of the algorithms in use by popular TCP implementations.

May 2001:

Vulnerability advisories [CERT2001] [USCERT2001] are released regarding statistical weaknesses in some ISN generators, affecting popular TCP/IP implementations.

.

TCP Initial Sequence Numbers (ISN)

March 2002:

[Zalewski2002] updates and complements [Zalewski2001]. It concludes that "while some vendors [...] reacted promptly and tested their solutions properly, many still either ignored the issue and never evaluated their implementations, or implemented a flawed solution that apparently was not tested using a known approach". [Zalewski2002].

February 2012:

[RFC6528], **27 years after Morris' original work** [Morris1985], formally updates [RFC0793] to mitigate predictable TCP ISNs.

August 2014:

[I-D.eddy-rfc793bis-04], the upcoming revision of the core TCP protocol specification, incorporates the algorithm specified in [RFC6528] as the recommended algorithm for TCP ISN generation

IPv4 / IPv6 Identification

December 1998:

[Sanfilippo1998a] finds that predictable IPv4 Identification values can be leveraged to count the number of packets sent by a target node. [Sanfilippo1998b] explains how to leverage the same vulnerability to implement a port-scanning technique known as dumb/idle scan. A tool that implements this attack is publicly released.

November 1999:

[Sanfilippo1999] discusses how to leverage predictable IPv4 Identification to uncover the rules of a number of firewalls.

November 1999:

[Bellovin2002] explains how the IPv4 Identification field can be exploited to count the number of systems behind a NAT.

December 2003:

[Zalewski2003] explains a technique to perform TCP data injection attack based on predictable IPv4 identification values which requires less effort than TCP injection attacks performed with bare TCP packets.

IPv4 / IPv6 Identification

November 2005:

[Silbersack2005] discusses shortcoming in a number of techniques to mitigate predictable IPv4 Identification values.

October 2007:

[Klein2007] describes a weakness in the pseudo random number generator (PRNG) in use for the generation of the IP Identification by a number of operating systems.

June 2011:

[Gont2011] describes how to perform idle scan attacks in IPv6.

November 2011:

Linux mitigates predictable IPv6 Identification values [RedHat2011] [SUSE2011] [Ubuntu2011].

December 2011:

[I-D.ietf-6man-predictable-fragment-id-08] describes the security implications of predictable IPv6 Identification values, and possible mitigations.

IPv4 / IPv6 Identification

May 2012:

[Gont2012] notes that some major IPv6 implementations still employ predictable IPv6 Identification values.

June 2015:

[I-D.ietf-6man-predictable-fragment-id-08] notes that some popular host and router implementations still employ predictable IPv6 Identification values.

January 2016:

[I-D.draft-ietf-6man-rfc2460bis.03] still suggests use of an algorithm that generates predictable IPv6 Identification values.

March 2016:

[I-D.draft-ietf-6man-rfc2460bis.04] now refers to [RFC7739] for selection of an algorithm to generate IPv6 identification values.

Problem addressed (sort of) 17 years after issues with IP identification values were first discussed

Why do you say “sort of”??

2460bis

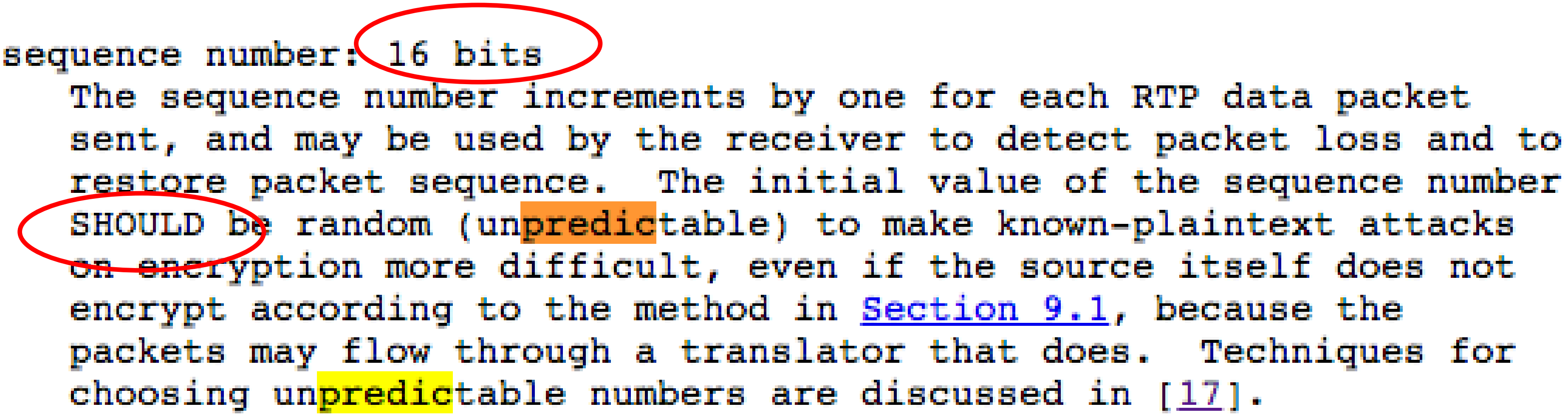
The Identification must be different than that of any other fragmented packet sent recently* with the same Source Address and Destination Address. If a Routing header Is present, the Destination Address of concern is that of the final destination.

> * "recently" means within the maximum likely lifetime of a packet, including transit time from source to destination and time spent awaiting reassembly with other fragments of the same packet. However, it is not required that a source node know the maximum packet lifetime. Rather, it is assumed that the requirement can be met by implementing an algorithm that results in a low identification reuse frequency. Examples of algorithms that can meet this requirement are described in [RFC7739].

Still NOT explicitly requiring IPv6 ID values to be NOT PREDICTABLE

See for example...

[RFC 3550] RTP: A Transport Protocol for Real-Time Applications



```
sequence number: 16 bits
   The sequence number increments by one for each RTP data packet
   sent, and may be used by the receiver to detect packet loss and to
   restore packet sequence.  The initial value of the sequence number
   SHOULD be random (unpredictable) to make known-plaintext attacks
   on encryption more difficult, even if the source itself does not
   encrypt according to the method in Section 9.1, because the
   packets may flow through a translator that does.  Techniques for
   choosing unpredictable numbers are discussed in [17].
```

Close, but no cigar

- Search space
- SHOULD vs MUST (!)

We need a cross-WG reference (BCP?)

Recommendations

Recommendations in the use of numeric IDS

- Semantics matter. Do I need…
  - ❑ A [global] unique ID ?
  - ❑ A sequence number ?
  - ❑ An address ?
  - ❑ A rubber ducky ?

- Think about the failure modes (interoperability)

The consequences of a failure to comply with the interoperability requirements of a given identifier. Severity considers the worst potential consequence of a failure, determined by the system damage and/or time lost to repair the failure. we define two types of failure severity: "soft" and "hard".

Interoperatbiity failure modes

- **Hard Failure**

  A hard failure is a non-recoverable condition in which a protocol does not operate in the prescribed manner or it operates with excessive degradation of service. For example, an established TCP connection that is aborted due to an error condition constitutes, from the point of view of the transport protocol, a hard failure, since it enters a state from which normal operation cannot be recovered.

- **Soft Failure**

  A soft failure is a recoverable condition in which a protocol does not operate in the prescribed manner but normal operation can be resumed automatically in a short period of time. For example, a simple packet-loss event that is subsequently recovered with a retransmission can be considered a soft failure.

Recommendations in the use of numeric IDS

- **Security & Privacy considerations**
  - **Search space, temporal scope**
  - **Transport layer && Spoofing**

    - Off-path attacks are real.

    Denial of Service

    Insertion

    Evasion

    - Replay

    - Information Leakage & tracking

- **Use The Random, Luke!**

  ...*with recommended algorithms only*…

Resources

- **Gont, F. and S. Bellovin, "Defending against Sequence Number Attacks", RFC 6528, DOI 10.17487/RFC6528, February 2012, http://www.rfc-editor.org/info/rfc6528**

See *“3. Proposed Initial Sequence Number Generation Algorithm”*

- **Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, http://www.rfc-editor.org/info/rfc7739**

See *“5. Algorithms for Selecting Fragment Identification Values”*

- **Gont, F. and Arce I., “Security and Privacy Implications of Numeric Identifiers Employed in Network Protocols”**

Work In Progress

Email: stic@fundacionsadosky.org.ar

GRACIAS!

References

[RFC1948] Bellovin, S., "Defending Against Sequence Number Attacks", RFC 1948, DOI 10.17487/RFC1948, May 1996, <http://www.rfc-editor.org/info/rfc1948>.

[Zalewski2001] Zalewski, M., "Strange Attractors and TCP/IP Sequence Number Analysis", 2001, <http://lcamtuf.coredump.cx/oldtcp/tcpseq.html>.

[Zalewski2002] Zalewski, M., "Strange Attractors and TCP/IP Sequence Number Analysis - One Year Later", 2001, <http://lcamtuf.coredump.cx/newtcp/>.

[Bellovin1989] Bellovin, S., "Security Problems in the TCP/IP Protocol Suite", Computer Communications Review, vol. 19, no. 2, pp. 32-48, 1989, <https://www.cs.columbia.edu/~smb/papers/ipext.pdf>.

[Joncheray1995] Joncheray, L., "A Simple Active Attack Against TCP", Proc. Fifth Usenix UNIX Security Symposium, 1995.

[Morris1985] Morris, R., "A Weakness in the 4.2BSD UNIX TCP/IP Software", CSTR 117, AT&T Bell Laboratories, Murray Hill, NJ, 1985, <https://pdos.csail.mit.edu/~rtm/papers/117.pdf>.

References

[Shimomura1995] Shimomura, T., "Technical details of the attack described by Markoff in NYT", Message posted in USENET's comp.security.misc newsgroup Message-ID:<3g5gkl$5j1@ariel.sdsc.edu>, 1995 <http://www.gont.com.ar/docs/post-shimomura-usenet.txt>.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport Protocol Port Randomization", BCP 156, RFC 6056, DOI 10.17487/RFC6056, January 2011 <http://www.rfc-editor.org/info/rfc6056>.

[RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, DOI 10.17487/RFC5927, July 2010, <http://www.rfc-editor.org/info/rfc5927>.

[RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <http://www.rfc-editor.org/info/rfc7739>.

[RFC4963] Heffner, J., Mathis, M., and B. Chandler, "IPv4 Reassembly Errors at High Data Rates", RFC 4963, DOI 10.17487/RFC4963, July 2007, <http://www.rfc-editor.org/info/rfc4963>.

References

[Bellovin2002] Bellovin, S., "A Technique for Counting NATted Hosts", IMW'02 Nov. 6-8, 2002, Marseille, France, 2002.

[Fyodor2004] Fyodor, , "Idle scanning and related IP ID games", 2004, <http://www.insecure.org/nmap/idlescan.html>.

[Sanfilippo1998a] Sanfilippo, S., "about the ip header id", Post to Bugtraq mailing-list, Mon Dec 14 1998, <http://seclists.org/bugtraq/1998/Dec/48>.

[Sanfilippo1998b] Sanfilippo, S., "Idle scan", Post to Bugtraq mailing-list, 1998, <http://www.kyuzz.org/antirez/papers/dumbscan.html>.

[Sanfilippo1999] Sanfilippo, S., "more ip id", Post to Bugtraq mailing-list, 1999, <http://www.kyuzz.org/antirez/papers/moreipid.html>.

[Silbersack2005] Silbersack, M., "Improving TCP/IP security through randomization without sacrificing interoperability", EuroBSDCon 2005 Conference, 2005, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.4542&rep=rep1&type=pdf>.

References

[Zalewski2003] Zalewski, M., "A new TCP/IP blind data injection technique?", 2003, <http://lcamtuf.coredump.cx/ipfrag.txt>.

[Klein2007] Klein, A., "OpenBSD DNS Cache Poisoning and Multiple O/S Predictable IP ID Vulnerability", 2007, <http://www.trusteer.com/files/OpenBSD_DNS_Cache_Poisoning_and_Multiple_OS_Predictable_IP_ID_Vulnerability.pdf>.

[I-D.ietf-6man-default-iids] Gont, F., Cooper, A., Thaler, D., and S. LIU, "Recommendation on Stable IPv6 Interface Identifiers", draft-ietf-6man-default-iids-09 (work in progress), January 2016.

[I-D.ietf-6man-ipv6-address-generation-privacy]
Cooper, A., Gont, F., and D. Thaler, "Privacy Considerations for IPv6 Address Generation Mechanisms", draft-ietf-6man-ipv6-address-generation-privacy-08 (work in progress), September 2015.