Internet Engineering Task Force (IETF)
Request for Comments: 5769
Category: Informational
ISSN: 2070-1721

R. Denis-Courmont
Nokia

April 2010

Test Vectors for Session Traversal Utilities for NAT (STUN)

Abstract

The Session Traversal Utilities for NAT (STUN) protocol defines several STUN attributes. The content of some of these -- FINGERPRINT, MESSAGE-INTEGRITY, and XOR-MAPPED-ADDRESS -- involve binary-logical operations (hashing, xor). This document provides test vectors for those attributes.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5769.

Table of Contents

1. Introduction

The Session Traversal Utilities for NAT (STUN)[RFC5389] protocol defines two different hashes that may be included in messages exchanged by peers implementing that protocol:

FINGERPRINT attribute: a 32-bit Cyclic Redundancy Check.

MESSAGE-INTEGRITY attribute: an HMAC-SHA1 [RFC2104] authentication code.

This document provides samples of properly formatted STUN messages including these hashes, for the sake of testing implementations of the STUN protocol.

2. Test Vectors

All included vectors are represented as a series of hexadecimal values in network byte order. Each pair of hexadecimal digits represents one byte.

Messages follow the Interactive Connectivity Establishment (ICE) Connectivity Checks use case of STUN (see [RFC5245]). These messages include FINGERPRINT, MESSAGE-INTEGRITY, and XOR-MAPPED-ADDRESS STUN attributes. These attributes are considered to be most prone to implementation errors. An additional message is provided to test STUN authentication with long-term credentials (which is not used by ICE).

In the following sample messages, two types of plain UTF-8 text attributes are included. The values of certain of these attributes were purposely sized to require padding. Non-ASCII characters are represented as <U+xxxx> where xxxx is the hexadecimal number of their Unicode code point.

In this document, ASCII white spaces (U+0020) are used for padding within the first three messages - this is arbitrary. Similarly, the last message uses nul bytes for padding. As per [RFC5389], padding bytes may take any value.

2.1. Sample Request

This request uses the following parameters:

Software name: "STUN test client" (without quotes)

Username: "evtj:h6vY" (without quotes)

Password: "VOkJxbRl1RmTxUk/WvJxBt" (without quotes)

```
       00 01 00 58     Request type and message length
       21 12 a4 42     Magic cookie
       b7 e7 a7 01  }
       bc 34 d6 86  }  Transaction ID
       fa 87 df ae  }
       80 22 00 10     SOFTWARE attribute header
       53 54 55 4e  }
       20 74 65 73  }  User-agent...
       74 20 63 6c  }  ...name
       69 65 6e 74  }
       00 24 00 04     PRIORITY attribute header
       6e 00 01 ff     ICE priority value
       80 29 00 08     ICE-CONTROLLED attribute header
       93 2f f9 b1  }  Pseudo-random tie breaker...
       51 26 3b 36  }   ...for ICE control
       00 06 00 09     USERNAME attribute header
       65 76 74 6a  }
       3a 68 36 76  }  Username (9 bytes) and padding (3 bytes)
       59 20 20 20  }
       00 08 00 14     MESSAGE-INTEGRITY attribute header
       9a ea a7 0c  }
       bf d8 cb 56  }
       78 1e f2 b5  }  HMAC-SHA1 fingerprint
       b2 d3 f2 49  }
       c1 b5 71 a2  }
       80 28 00 04     FINGERPRINT attribute header
       e5 7a 3b cf     CRC32 fingerprint
```

2.2. Sample IPv4 Response

This response uses the following parameter:

Password: "VOkJxbRl1RmTxUk/WvJxBt" (without quotes)

Software name: "test vector" (without quotes)

Mapped address: 192.0.2.1 port 32853

```
  01 01 00 3c     Response type and message length
  21 12 a4 42     Magic cookie
  b7 e7 a7 01  }
  bc 34 d6 86  }  Transaction ID
  fa 87 df ae  }
  80 22 00 0b     SOFTWARE attribute header
  74 65 73 74  }
  20 76 65 63  }  UTF-8 server name
  74 6f 72 20  }
  00 20 00 08     XOR-MAPPED-ADDRESS attribute header
  00 01 a1 47     Address family (IPv4) and xor'd mapped port number
  e1 12 a6 43     Xor'd mapped IPv4 address
  00 08 00 14     MESSAGE-INTEGRITY attribute header
  2b 91 f5 99  }
  fd 9e 90 c3  }
  8c 74 89 f9  }  HMAC-SHA1 fingerprint
  2a f9 ba 53  }
  f0 6b e7 d7  }
  80 28 00 04     FINGERPRINT attribute header
  c0 7d 4c 96     CRC32 fingerprint
```

2.3. Sample IPv6 Response

This response uses the following parameter:

Password: "VOkJxbRl1RmTxUk/WvJxBt" (without quotes)

Software name: "test vector" (without quotes)

Mapped address: 2001:db8:1234:5678:11:2233:4455:6677 port 32853

```
  01 01 00 48     Response type and message length
  21 12 a4 42     Magic cookie
  b7 e7 a7 01  }
  bc 34 d6 86  }  Transaction ID
  fa 87 df ae  }
  80 22 00 0b     SOFTWARE attribute header
  74 65 73 74  }
  20 76 65 63  }  UTF-8 server name
  74 6f 72 20  }
  00 20 00 14     XOR-MAPPED-ADDRESS attribute header
  00 02 a1 47     Address family (IPv6) and xor'd mapped port number
  01 13 a9 fa  }
  a5 d3 f1 79  }  Xor'd mapped IPv6 address
  bc 25 f4 b5  }
  be d2 b9 d9  }
  00 08 00 14     MESSAGE-INTEGRITY attribute header
  a3 82 95 4e  }
  4b e6 7b f1  }
  17 84 c9 7c  }  HMAC-SHA1 fingerprint
  82 92 c2 75  }
  bf e3 ed 41  }
  80 28 00 04     FINGERPRINT attribute header
  c8 fb 0b 4c     CRC32 fingerprint
```

2.4. Sample Request with Long-Term Authentication

This request uses the following parameters:

Username: "<U+30DE><U+30C8><U+30EA><U+30C3><U+30AF><U+30B9>" (without quotes) unaffected by SASLprep [RFC4013] processing

Password: "The<U+00AD>M<U+00AA>tr<U+2168>" and "TheMatrIX" (without quotes) respectively before and after SASLprep processing

Nonce: "f//499k954d6OL34oL9FSTvy64sA" (without quotes)

Realm: "example.org" (without quotes)

```
   00 01 00 60     Request type and message length
   21 12 a4 42     Magic cookie
   78 ad 34 33  }
   c6 ad 72 c0  }  Transaction ID
   29 da 41 2e  }
   00 06 00 12     USERNAME attribute header
   e3 83 9e e3  }
   83 88 e3 83  }
   aa e3 83 83  }  Username value (18 bytes) and padding (2 bytes)
   e3 82 af e3  }
   82 b9 00 00  }
   00 15 00 1c     NONCE attribute header
   66 2f 2f 34  }
   39 39 6b 39  }
   35 34 64 36  }
   4f 4c 33 34  }  Nonce value
   6f 4c 39 46  }
   53 54 76 79  }
   36 34 73 41  }
   00 14 00 0b     REALM attribute header
   65 78 61 6d  }
   70 6c 65 2e  }  Realm value (11 bytes) and padding (1 byte)
   6f 72 67 00  }
   00 08 00 14     MESSAGE-INTEGRITY attribute header
   f6 70 24 65  }
   6d d6 4a 3e  }
   02 b8 e0 71  }  HMAC-SHA1 fingerprint
   2e 85 c9 a2  }
   8c a8 96 66  }
```

3. Security Considerations

There are no security considerations.

4. Acknowledgments

The author would like to thank Marc Petit-Huguenin, Philip Matthews and Dan Wing for their inputs, and Brian Korver, Alfred E. Heggestad and Gustavo Garcia for their reviews.

5. References

5.1. Normative References

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

5.2. Informative References

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC4013] Zeilenga, K., "SASLprep: Stringprep Profile for User Names and Passwords", RFC 4013, February 2005.

Appendix A. Source Code for Test Vectors

```
   const unsigned char req[] =
     "\x00\x01\x00\x58"
     "\x21\x12\xa4\x42"
     "\xb7\xe7\xa7\x01\xbc\x34\xd6\x86\xfa\x87\xdf\xae"
     "\x80\x22\x00\x10"
       "STUN test client"
     "\x00\x24\x00\x04"
       "\x6e\x00\x01\xff"
     "\x80\x29\x00\x08"
       "\x93\x2f\xf9\xb1\x51\x26\x3b\x36"
     "\x00\x06\x00\x09"
       "\x65\x76\x74\x6a\x3a\x68\x36\x76\x59\x20\x20\x20"
     "\x00\x08\x00\x14"
       "\x9a\xea\xa7\x0c\xbf\xd8\xcb\x56\x78\x1e\xf2\xb5"
       "\xb2\xd3\xf2\x49\xc1\xb5\x71\xa2"
     "\x80\x28\x00\x04"
       "\xe5\x7a\x3b\xcf";
```

Request message

```
const unsigned char respv4[] =
  "\x01\x01\x00\x3c"
  "\x21\x12\xa4\x42"
  "\xb7\xe7\xa7\x01\xbc\x34\xd6\x86\xfa\x87\xdf\xae"
  "\x80\x22\x00\x0b"
    "\x74\x65\x73\x74\x20\x76\x65\x63\x74\x6f\x72\x20"
  "\x00\x20\x00\x08"
    "\x00\x01\xa1\x47\xe1\x12\xa6\x43"
  "\x00\x08\x00\x14"
    "\x2b\x91\xf5\x99\xfd\x9e\x90\xc3\x8c\x74\x89\xf9"
    "\x2a\xf9\xba\x53\xf0\x6b\xe7\xd7"
  "\x80\x28\x00\x04"
    "\xc0\x7d\x4c\x96";
```

IPv4 response message

```
const unsigned char respv6[] =
  "\x01\x01\x00\x48"
  "\x21\x12\xa4\x42"
  "\xb7\xe7\xa7\x01\xbc\x34\xd6\x86\xfa\x87\xdf\xae"
  "\x80\x22\x00\x0b"
    "\x74\x65\x73\x74\x20\x76\x65\x63\x74\x6f\x72\x20"
  "\x00\x20\x00\x14"
    "\x00\x02\xa1\x47"
    "\x01\x13\xa9\xfa\xa5\xd3\xf1\x79"
    "\xbc\x25\xf4\xb5\xbe\xd2\xb9\xd9"
  "\x00\x08\x00\x14"
    "\xa3\x82\x95\x4e\x4b\xe6\x7b\xf1\x17\x84\xc9\x7c"
    "\x82\x92\xc2\x75\xbf\xe3\xed\x41"
  "\x80\x28\x00\x04"
    "\xc8\xfb\x0b\x4c";
```

IPv6 response message

```
   const unsigned char reqltc[] =
     "\x00\x01\x00\x60"
     "\x21\x12\xa4\x42"
     "\x78\xad\x34\x33\xc6\xad\x72\xc0\x29\xda\x41\x2e"
     "\x00\x06\x00\x12"
       "\xe3\x83\x9e\xe3\x83\x88\xe3\x83\xaa\xe3\x83\x83"
       "\xe3\x82\xaf\xe3\x82\xb9\x00\x00"
     "\x00\x15\x00\x1c"
       "\x66\x2f\x2f\x34\x39\x39\x6b\x39\x35\x34\x64\x36"
       "\x4f\x4c\x33\x34\x6f\x4c\x39\x46\x53\x54\x76\x79"
       "\x36\x34\x73\x41"
     "\x00\x14\x00\x0b"
       "\x65\x78\x61\x6d\x70\x6c\x65\x2e\x6f\x72\x67\x00"
     "\x00\x08\x00\x14"
       "\xf6\x70\x24\x65\x6d\xd6\x4a\x3e\x02\xb8\xe0\x71"
       "\x2e\x85\xc9\xa2\x8c\xa8\x96\x66";
```

Request with long-term credentials

Author's Address

Remi Denis-Courmont
Nokia Corporation
P.O. Box 407
NOKIA GROUP 00045
FI

Phone: +358 50 487 6315
EMail: remi.denis-courmont@nokia.com