Applying GREASE to TLS Extensibility
draft-davidben-tls-grease-00

Abstract

This document describes GREASE (Generate Random Extensions And Sustain Extensibility), a mechanism to prevent extensibility failures in the TLS ecosystem. It reserves a set of TLS protocol values that may be advertised by clients to ensure servers correctly handle unknown values.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 26, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of...
1. Introduction

The TLS protocol [RFC5246] includes several points of extensibility, including the list of cipher suites and the list of extensions. The values in these lists identify implementation capabilities. TLS follows a model where clients advertise capabilities and servers select them. It is required that servers ignore unknown values so that new capabilities may be introduced to the ecosystem while maintaining interoperability.

However, bugs may cause a server to reject unknown values. These broken servers will interoperate with existing clients, so the mistake may spread through the ecosystem unnoticed. Later, when new values are defined, updated clients will discover that the metaphorical joint in the protocol has rusted shut and that the new values cannot be deployed without interoperability failures.

To avoid this problem, this document reserves some currently unused values for clients to advertise at random. Correct server implementations will ignore these values and interoperate. Servers that do not tolerate unknown values will fail to interoperate with existing clients, revealing the mistake before it is widespread. This document reserves such values in the TLS cipher suite, extension, named group [RFC4492], and ALPN [RFC7301] registries.

In keeping with the rusted joint metaphor, this technique is named GREASE (Generate Random Extensions And Sustain Extensibility).
1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. GREASE Values

This document reserves a number of TLS protocol values, referred to as GREASE values. These values were allocated sparsely to discourage server implementations from conditioning on them. For convenience, they were also chosen so all types share a number scheme with a consistent pattern while avoiding collisions with any existing applicable registries in TLS.

The following values are reserved as GREASE cipher suite values:

{0x0A, 0x0A}
{0x1A, 0x1A}
{0x2A, 0x2A}
{0x3A, 0x3A}
{0x4A, 0x4A}
{0x5A, 0x5A}
{0x6A, 0x6A}
{0x7A, 0x7A}
{0x8A, 0x8A}
{0x9A, 0x9A}
{0xAA, 0xAA}
{0xBA, 0xBA}
{0xCA, 0xCA}
{0xDA, 0xDA}
{0xEA, 0xEA}
{0xFA, 0xFA}

The following values are reserved as both GREASE extension values and GREASE named group values:

2570 (0x0A0A)
6682 (0x1A1A)
10794 (0x2A2A)
14906 (0x3A3A)
19018 (0x4A4A)
23130 (0x5A5A)
27242 (0x6A6A)
31354 (0x7A7A)
35466 (0x8A8A)
39578 (0x9A9A)
43690 (0xA AAA)
47802 (0xBABA)
51914 (0xCACA)
56026 (0xDADA)
60138 (0xEAEA)
64250 (0xFAFA)

TODO: Depending on which of this or TLS 1.3 happens first, also reserve SignatureScheme values. (The same number scheme will work fine there too.)

Note that these correspond to the reserved cipher suites when treated as big-endian 16-bit integers.

Finally, this document reserves all ALPN identifiers beginning with the prefix "ignore/". This corresponds to the seven-octet prefix: 0x69, 0x67, 0x6e, 0x6f, 0x72, 0x65, 0x2f.

3. Client Behavior

When sending a ClientHello, a client which implements GREASE behaves as follows:

- A client MAY select one or more random GREASE cipher suite values and advertise them in the ClientHello.cipher_suites field.

- A client MAY select one or more random GREASE named group values and advertise them in the supported_groups extension, if sent.

- A client MAY select one or more random GREASE extension values and advertise them in the ClientHello.extensions field. The client MAY randomly vary the length and contents of the extension.

- A client MAY select one or more random GREASE ALPN identifiers and advertise them in the application_layer_protocol_negotiation extension, if sent.

Clients using GREASE values SHOULD randomly vary their position relative to other offered values. However, note that some server implementations do not interoperate correctly with a ClientHello whose final extension is empty. Thus clients SHOULD avoid using an empty GREASE extension as the final extension.

Clients MUST reject GREASE values when negotiated by the server. When processing a ServerHello containing a GREASE value in the ServerHello.cipher_suite or ServerHello.extensions fields, the client MUST fail the connection. When processing an ECParameters structure with a GREASE value in the ECParameter.namedcurve field, the client MUST fail the connection. (Note that this requires no special
processing on the client. Clients are already required to reject unknown values selected by the server.)

4. Server Behavior

Servers MUST NOT treat GREASE values differently from any unknown value. Servers MUST NOT negotiate any GREASE value when offered in a ClientHello. Servers MUST correctly ignore unknown values in a ClientHello and attempt to negotiate with one of the remaining parameters.

5. IANA Considerations

This document updates the TLS Cipher Suite Registry, available from <https://www.iana.org/assignments/tls-parameters>:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>DTLS-OK</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>{0x0A,0x0A}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0x1A,0x1A}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0x2A,0x2A}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0x3A,0x3A}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0x4A,0x4A}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0x5A,0x5A}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0x6A,0x6A}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0x7A,0x7A}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0x8A,0x8A}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0x9A,0x9A}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0xAA,0xAA}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0xBA,0xBA}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0xCA,0xCA}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0xDA,0xDA}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0xEA,0xEA}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>{0xFA,0xFA}</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
</tbody>
</table>

Additions to the TLS Cipher Suite Registry

This document updates the Supported Groups Registry, available from <https://www.iana.org/assignments/tls-parameters>: 
### Additions to the Supported Groups Registry

This document updates the ExtensionType Values registry, available from [https://www.iana.org/assignments/tls-extensiontype-values](https://www.iana.org/assignments/tls-extensiontype-values):

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>DTLS-OK</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>2570</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>6682</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>10794</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>14906</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>19018</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>23130</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>27242</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>31354</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>35466</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>39578</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>43690</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>47802</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>51914</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>56026</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>60138</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
<tr>
<td>64250</td>
<td>Reserved</td>
<td>Y</td>
<td>(this document)</td>
</tr>
</tbody>
</table>
TODO: How do I write IANA instructions to reserve all ALPN identifiers that begin with "ignore/"? Perhaps it would be better to reserve a concrete handful of identifiers instead.

6. Security Considerations

GREASE values may not be negotiated, so they do not directly impact the security of TLS connections.

Historically, when interoperability problems arise in deploying new TLS features, implementations have used a fallback retry on error with the feature disabled. This allows an active attacker to silently disable the new feature. By preventing a class of such interoperability problems, GREASE reduces the need for this kind of fallback.

7. Acknowledgements

The author would like to thank Adam Langley, Nick Harper, and Steven Valdez for their feedback and suggestions. In addition, the rusted joint metaphor is originally due to Adam Langley.

8. Normative References


