Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 6, 2017

D. Benjamin
Google

September 2, 2016

Applying GREASE to TLS Extensibility
draft-davidben-tls-grease-01

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 March 6, 2017.

Table of Contents

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:

{Values TBD}

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

{Values TBD}

[[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 corresponding extensions with varying length and contents in the ClientHello.extensions field.

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

Clients SHOULD balance diversity in GREASE advertisements with determinism. For example, a client which randomly varies GREASE value positions for each connection may only fail against a broken server with some probability. This risks the failure being masked by automatic retries. A client which positions GREASE values deterministically over a period of time (such as a single software release) stresses fewer cases but is more likely to detect bugs from those cases.

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.

Note that these requirements are restatements or corollaries of existing server requirements in TLS.

5. IANA Considerations

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

Value	Description	DTLS-OK	Reference
{TBD} {0x0A,0x0A}	Reserved	Y	(this document)
{TBD} {0x1A,0x1A}	Reserved	Y	(this document)
{TBD} {0x2A,0x2A}	Reserved	Y	(this document)
{TBD} {0x3A,0x3A}	Reserved	Y	(this document)
{TBD} {0x4A,0x4A}	Reserved	Y	(this document)
{TBD} {0x5A,0x5A}	Reserved	Y	(this document)
{TBD} {0x6A,0x6A}	Reserved	Y	(this document)
{TBD} {0x7A,0x7A}	Reserved	Y	(this document)
{TBD} {0x8A,0x8A}	Reserved	Y	(this document)
{TBD} {0x9A,0x9A}	Reserved	Y	(this document)
{TBD} {0xAA,0xAA}	Reserved	Y	(this document)
{TBD} {0xBA,0xBA}	Reserved	Y	(this document)
{TBD} {0xCA,0xCA}	Reserved	Y	(this document)
{TBD} {0xDA,0xDA}	Reserved	Y	(this document)
{TBD} {0xEA,0xEA}	Reserved	Y	(this document)
{TBD} {0xFA,0xFA}	Reserved	Y	(this document)

Additions to the TLS Cipher Suite Registry

The cipher suite numbers listed in the first column are numbers used for cipher suite interoperability testing and it's suggested that IANA use these values for assignment.

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

Value	Description	DTLS-OK	Reference
{TBD} 2570	Reserved	Y	(this document)
{TBD} 6682	Reserved	Y	(this document)
{TBD} 10794	Reserved	Y	(this document)
{TBD} 14906	Reserved	Y	(this document)
{TBD} 19018	Reserved	Y	(this document)
{TBD} 23130	Reserved	Y	(this document)
{TBD} 27242	Reserved	Y	(this document)
{TBD} 31354	Reserved	Y	(this document)
{TBD} 35466	Reserved	Y	(this document)
{TBD} 39578	Reserved	Y	(this document)
{TBD} 43690	Reserved	Y	(this document)
{TBD} 47802	Reserved	Y	(this document)
{TBD} 51914	Reserved	Y	(this document)
{TBD} 56026	Reserved	Y	(this document)
{TBD} 60138	Reserved	Y	(this document)
{TBD} 64250	Reserved	Y	(this document)

Additions to the Supported Groups Registry

The named group numbers listed in the first column are numbers used for cipher suite interoperability testing and it's suggested that IANA use these values for assignment.

This document updates the ExtensionType Values registry, available from <https://www.iana.org/assignments/tls-extensiontype-values>:

Value	Extension name	Reference
{TBD} 2570	Reserved	(this document)
{TBD} 6682	Reserved	(this document)
{TBD} 10794	Reserved	(this document)
{TBD} 14906	Reserved	(this document)
{TBD} 19018	Reserved	(this document)
{TBD} 23130	Reserved	(this document)
{TBD} 27242	Reserved	(this document)
{TBD} 31354	Reserved	(this document)
{TBD} 35466	Reserved	(this document)
{TBD} 39578	Reserved	(this document)
{TBD} 43690	Reserved	(this document)
{TBD} 47802	Reserved	(this document)
{TBD} 51914	Reserved	(this document)
{TBD} 56026	Reserved	(this document)
{TBD} 60138	Reserved	(this document)
{TBD} 64250	Reserved	(this document)

Additions to the ExtensionType Values registry

The extension numbers listed in the first column are numbers used for cipher suite interoperability testing and it's suggested that IANA use these values for assignment.

[[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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, DOI 10.17487/RFC4492, May 2006, <http://www.rfc-editor.org/info/rfc4492>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <http://www.rfc-editor.org/info/rfc7301>.

Author's Address

David Benjamin
Google
355 Main St
Cambridge, MA 02142
USA

Email: davidben@google.com